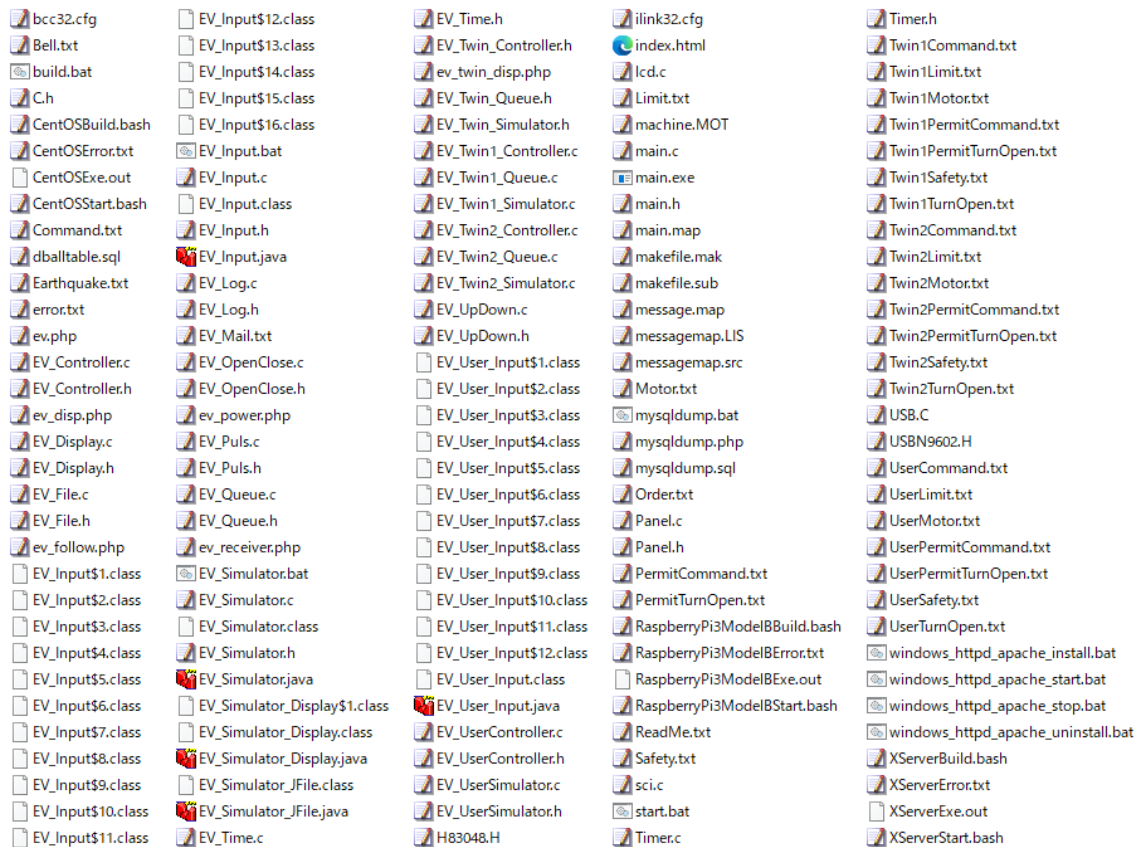


発明の巻

タイトル： センサーからアプリ、アプリからモーターまで、入力から出力までの電気設計

今回はシステム設計について考えてみたいと思います。こんな設計はどうでしょうか？ センサーの中身は、短絡・絶縁でスイッチングしています。A 接点と B 接点の 2 種類があります。~~センサーから伸びる 2 本の銅線は電源電圧と基準電圧につながって、プルアップ方式は短絡のとき電源電圧、絶縁のとき基準電圧、プルダウン方式は短絡のとき基準電圧、絶縁のとき電源電圧になり、電圧信号として取り扱えます。~~(回顧録平成 29 年 1 月 27 日： センサーから伸びる 2 本の銅線は電源電圧と基準電圧につながって、プルアップ方式は解放のとき電源電圧、短絡のとき信号電圧、プルダウン方式は解放のとき基準電圧、短絡のとき信号電圧になり、電圧信号として取り扱えます。) 集積点に集まってきた電圧信号はパラレル(並列)→シリアル(直列)変換をします。並列信号はシフトレジスタに入力され、直列ワード(シリアルデータ)になります。シフトレジスタはクロック信号で駆動されます。~~数箇所の集積点から集まってきたシリアルデータはトークンで順番に CPU に割り込みます。トークンはリングカウンタで構成されます。シリアルデータ用クロックは CPU の外で作り、CPU にも入力します。割り込みを起点にクロックでタイミングを測りながらシリアルデータを入力します。入力値は割り込み処理ルーチンの記述でポインタに格納します。アプリケーションはポインタを参照して処理します。~~(回顧録平成 29 年 1 月 28 日： 数箇所の集積点から集まってきたシリアルデータはトークンで順番に CPU に取り込まれます。トークンはリングカウンタで構成されます。シリアルデータ用クロックは、マイコンの出力ポインタアドレスから出力して、リングカウンタに送り込み、リングカウンタから、トークン権をもっているシリアルデータを、クロックのタイミングで、マイコンの入力ポインタアドレスに取得します。入力値は、疑似スレッドの、シリアル通信クロック取り扱い用スレッドの記述で、構造体の変数に格納します。アプリケーションは構造体の変数を参照して処理します。) 処理結果はあて先付きシリアルデータとして出力します。デコーダで行き先が振り分けられます。目的の信号はリレーコイルで電力線に変換されます。電力線が ON のとき電源(200V 等)からモーターに電力が供給されモーターが回転します。私はこのように設計しますが、皆さんもっと良い方法があれば教えて下さい。

C 言語の疑似スレッド最新版(ライセンスフリー) の サポートパック (330,000円) (内10%消費税30,000円)



スレッドとは、コンピュータープログラミング上の、並列処理の機能です。スレッドをサポートしているプログラミング言語と、スレッドをサポートしていないプログラミング言語があります。高度な機能のスレッドをわざとサポートしない言語があるのは、パソコン以外の環境で実行するために、低レベルマイコン対応のプログラミング言語でいるから(例えばC 言語)で

発明アルゴリズムに疑似スレッドと名付けました。本物に偽物と名付けました。特許申請はしないで、無料公開をいたしました。発明という手段で人類としてのノルマを達成し、また、生きた証が発生いたしました。 [REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

タイトル(title):

C 言語の疑似スレッド

サブタイトル(subtitle):

C 言語の偽物のスレッド

The thread at the imitation of the C language.

似ているが独創的な別物

The resembling but original singleton.

C 言語によるスレッドという概念の模倣

Copying a concept,

the thread, by the C language.

解説

C言語のプロジェクト Thread について、

このプロジェクトは予告なくデバッグ目的で更新されることがあります。

このプロジェクトは、お客様が改造して、よりお客様に使いやすいプログラムにするために、提供されるサンプルプログラムであり、お客様の好みに完成させてください。

=====
このプロジェクトは、4通りの方法でコンパイル 実行 できます。

1つ目の方法：

Windows版 Borland BCC C/C++ のダウンロードとインストール

bcc コンパイラー で検索します

C++Builder のホームページを開きます

C++ Compiler 5.5 / Turbo Debugger (日本語) (コンパイラのみで軽い)

(テキストエディタと組み合わせて使用します)(フリーソフト)

(ユーザー登録が必要)(メールアドレスが必要)をダウンロードして

解凍します

freecommandlinetools2.exeをインストールします

freeturbodebugger.exeをインストールします

makefile.mak build.bat は

複数のファイルを1個のプロジェクトとしてコンパイルするための

ファイルです

error.txt はコンパイルエラーを表示するファイルです

main.exe をダブルクリックすると BCC実行ソフト (このソフト) が
起動します

Javaのダウンロードとインストール

JDK で検索します

Java をダウンロードしてインストールして下さい

jcpad で検索します

Java開発環境 jcpad をダウンロードしてインストールして下さい

start.bat をダブルクリックすると BCC実行ソフト (このソフト) と
JavaSwing が起動して、 2階建てエレベーター と 3階建てエレベーター
が同時起動します

ブラウザで起動する場合

ev_power.php の exec main.exe の 3択に 注意してください

2つ目の方法：

Windows版 AKI-H8 3052F USB開発セット の購入

メーカー：ルネサスエレクトロニクスさん

販売者：秋月電子通商さん

通販コード：K-00182

商品名：AKI-H8 3052F USB開発セット

商品価格：税込7,000円(2018年3月6日現在)

AKI-H8 3052F USB開発セット で検索します

AKI-H8 3052F USB開発セット を 秋月電子通商さんの通販サイトで購入するか、それとも ご自宅の最寄りの電子パーツ専門店で注文購入します

AC/DCコンバータ(ACアダプター) と、 RS-232Cケーブル と、
延長USBコード を、ご自宅の最寄りの電子パーツ専門店で注文購入
します

AC/DCコンバータ の 電流電圧 は お店の人に聞いてください
RS-232C は、パソコンに端子がある必要があり、また、パソコンと
AKI-H8基板 の両方の コネクタ の オス端子・メス端子 を確認して
購入してください

延長USBコード も オス端子・メス端子 を確認してください

カラー短絡ソケット6mm 青 2228CG-BU で検索します
カラー短絡ソケット6mm 青 2228CG-BU のような 短絡コネクタ を、
ご自宅の最寄りの電子パーツ専門店で 2個 注文購入します
(100円 か 200円 支払うと何個かまとめて購入できます)

USB開発セット のマニュアルに従い、 usbフォルダ の main.c を
コンパイルして、 H8WriteTurbo をインストールして、 AKI-H8基板
への書き込み時に、 短絡ソケット(短絡コネクタ) を使用して、
マニュアルに従い モード を調節して、 usbフォルダ の usbtest.MOT
を AKI-H8基板 へ書き込み、走らせてみます
押しボタンを押すと、sw1 sw2 sw3 sw4 などと 液晶パネル に
表示されれば大丈夫です。

C_Threadフォルダ の中の、 Thread_Workフォルダ の中の、 main.c
を、コンパイルします(build.bat を、編集で中身を確認後、

ダブルクリックします)

C_Threadフォルダの中の、Thread_Workフォルダの中の、machine.MOT
を、モードを調節しながら、AKI-H8基板へ書き込み、走らせてみます
液晶パネル・押しボタン・LEDなどが機能していれば、成功です

messagemap.src makefile.sub build.bat は

複数のファイルを1個のプロジェクトとしてコンパイルするための
ファイルです

Puls.h Puls.c の記述により、

PortB bit 0 1 2 3 4 5 (AKI-H8-CN1 pin 16 17 18 19 20) から、
エレベーターのモーター指令信号の出力(OUTPUT)があります

error.txt はコンパイルエラーを表示するファイルです

3つ目の方法：

LINUX CentOS6 GCC の利用

LINUX の GCC を使用するときには、文字コードを utf8 にして
改行コードを <LF>のみ(UNIX) にして名前を付けて保存してください

データベース MySQL を使用しておりますので、dballtable.sql
を解析して、MySQL の準備をしてから、挑戦してみてください

MySQL をインストールしたら、所定の場所に、<mysql.h> があると

思います

エレベーター EV の、非常停止メール通報 について、

work@localhost.localdomain

work@127.0.0.1 のところを、

適当な 送信元 ・ 送信先 にしてください

CentOSBuild.bash は

複数のファイルを1個のプロジェクトとしてコンパイルするための

ファイルです

CentOSError.txt はコンパイルエラーを表示するファイルです

LINUX CentOS6 で CentOSStart.bash をダブルクリックすると、

GCC実行ソフト CentOSExe.out (このソフト) が起動します

私は、CentOS にしましたが、CentOSBuild.bash CentOSStart.bash

が読める方は、お好きな LINUX で挑戦してみてください

4つ目の方法：

LINUX Raspberry Pi 3 Model B (ラズベリーパイ) Raspbian GCC の利用

LINUX の GCC を使用するときには、文字コードを utf8 にして

改行コードを <LF>のみ(UNIX) にして名前を付けて保存してください

データベース MySQL を使用しておりますので、dballtable.sql
を解析して、MySQL の準備をしてから、挑戦してみてください

以下のコードを試して、<mysql.h> をインストールしてください

```
sudo apt-get install libmariadbclient-dev
```

エレベーター EV の、非常停止メール通報 について、

LINUX のシステムコールの mail コマンドと、

LINUX のプロセス分割 fork() を使いました

info@hidemine.ciao.jp のところを、

適当な送信先にしてください

RaspberryPi3ModelBBuild.bash は

複数のファイルを1個のプロジェクトとしてコンパイルするための

ファイルです

RaspberryPi3ModelBError.txt はコンパイルエラーを表示するファイルです

LINUX Raspbian で RaspberryPi3ModelBStart.bash

をダブルクリックすると、GCC実行ソフト RaspberryPi3ModelBExe.out

(このソフト) が起動します

Puls.h Puls.c の記述により、

GPIO 16 17 18 19 20 (pin 36 11 12 35 38) から、

エレベーターのモーター指令信号の出力(OUTPUT) があります

ブラウザで起動する場合

ev_power.php の exec sudo . . . Exe.out の3択に 注意してください

=====
messagemap.src について

リセットベクトルの 転送先ラベルが `_start` になっています
ずっと下の方の `_start` の ラベル から処理を開始して、
`jsr @_main` で C言語の関数 `main` を呼び出しています
C言語の関数 `main` は、 `void main(void);` という形で、
`main.c` に記述があります
その後、 `int_error: rte` で `rte` (`return`と同じ意味) で終了しています

リセットベクトルに続く1番から60番までの 割り込みベクトルについて、
使用しない 割り込みベクトルは ラベル `int_error` に転送されます

`;26 OVI0 _INT_OVI0: .DATA.L _ITU_OVI_0 ;タイマ0割り込み`
で、 タイマ0割り込み は、 ラベル `_ITU_OVI_0` に転送されます
ラベル `_ITU_OVI_0` から開始して、 スタック 退避をして、
`jsr @_InterruptITU0` で、 C言語の関数 `InterruptITU0`
を呼び出しています
C言語の関数 `InterruptITU0` は `void InterruptITU0(void);`
という形で、 `Timer.h` `Timer.c` に記述があります
戻ってくると、再び スタック を戻して、 `rte` です

ファイルの先頭に、

```
.IMPORT _main
```

```
.IMPORT _InterruptITU0
```

という記述があり、 C言語の関数 を参照しています

```
.EXPORT _EnableInterrupt,_DisableInterrupt
```

```
_EnableInterrupt: andc.b #H'3f,ccr rts
```

`_DisableInterrupt: orc.b #H'c0,ccr rts`

で、C言語から

`_EnableInterrupt` (割り込み許可)

`_DisableInterrupt` (割り込み禁止)

を呼び出せるようにしています

C言語の `Panel.h` に 外部参照プロトタイプ宣言 があります

```
extern void EnableInterrupt(void);
```

```
extern void DisableInterrupt(void);
```

C言語からの呼び出し名は、

```
EnableInterrupt();
```

```
DisableInterrupt();
```

です

=====

main.c の 関数 `Run` の ID==31 を見てください

Thread Ready GO! で開始して競馬のコースが8コースあります

Thread Ready GO! There are 8 cources on a race.

ゴールまで14歩です

There are 14 cells to a GOAL.

<1>コースは'r'ボタンが鞭で<2>コースは'l'ボタンが鞭です

For the <1> course, You click a 'R' button.

For the <2> cource, You click a 'L' button.

スレッドを使用しています

Thread `*th[8];` でオブジェクト宣言しています

`th[i] = new_Thread(i + 1);` で初期値設定しています

この2行は Java で次と同じ意味です

```
Thread th[] = new Thread[8];
```

```
th[i] = new Thread(i + 1);
```

```
void Repaint(void)
```

```
{
```

```
    ...
```

```
}
```

```
void Run(Thread *This)
```

```
{
```

```
    ...
```

```
}
```

```
void Init(Thread *This)
```

```
{
```

```
    ...
```

```
}
```

```
void Destroy(Thread *This)
```

```
{
```

```
    ...
```

```
}
```

はそれぞれ Java で次と同じ意味です

```
public void paint(Graphics g)
```

```
{
```

```
    ...
```

```
}
```

```
public void run()
```

```
{
```

```
    ...
```



```
}  
  
public void init()  
{  
  
    ...  
  
}  
  
public void destroy()  
{  
  
    ...  
  
}
```

`delete_(th[i]);` でオブジェクトを消去しています

この1行は C++ で次と同じ意味です

```
delete th[i];
```

スレッド20が走り始めたら、

0以外の数字キーを押してみてください

その数字に20を加えた番号のスレッドが、キーを押す度毎に、

起動・消去を繰り返します

20を含めて、全部スレッドが消去されると、終了です

これらのスレッドに関する仕様は Timer.c に記述しました

=====

2階建エレベーターEVについて

使用方法

(※重要)開始前にチェックしてください。

※Command.txt の中身が N になっていたら、q に替えてください。

※Order.txt の中身が q になっていたら、N__ に替えてください。

EV_Simulator にエレベーターが表示されます

EV_Controller にエレベーターの動作が表示されます

EV_Input の使用方法

u キーを押すとエレベーターが2階に上昇して扉が開きます

d キーを押すとエレベーターが1階に下降して扉が開きます

o キーを押すと扉が開きます

c キーを押すと扉が閉じます

s キーを押すと籠が非常停止します

r キーを押すと籠が非常停止から復帰します

Y キーを押すとエレベーターが2階に上昇して扉が開きます

H キーを押すと2階で扉が閉じます

y キーを押すとエレベーターが1階に下降して扉が開きます

h キーを押すと1階で扉が閉じます

籠が無い階で H h キーを押しても籠は動作しません

籠が無い階で Y y キーを押したとき籠の扉が開いていると、

籠は動作しません

開いた状態の扉は一定時間後自動で閉じます

EV_Time.h に #define OPENTIMEOUT 10 と書いてあるので 10秒 です

閉まりかけの時に開く動作をするキーを押すと扉が反転して開きます

動作説明

全体の動作説明

モーターの情報は Motor.txt にあります

エレベーター塔内のリミットスイッチの情報は Limit.txt にあります

EV_Simulator はエレベーターの次の位置を出力していて Safety.txt
Motor.txt Limit.txt を採取して Safety.txt Limit.txt に書き込んで
エレベーターの画面表示もしています

EV_Controller はエレベータを制御していて Command.txt Limit.txt
を採取して PermitCommand.txt Motor.txt に書き込んでいます

EV_Controllerの動作説明

エレベーターには現在位置情報(Limit.txt)があります

最簡形の2階建ての場合通常系には5個の位置状態があります

下の階の停止状態

下の階の低速区域

中間の高速区域

上の階の低速区域

上の階の停止状態

5個の区域の境界に合計4個のセンサーがあります

4個のセンサーがエレベーターの現在位置を取得しています

4個のセンサーからの信号はメンバ変数(Positionクラスの

*p_UnderSlow *p_UnderStop *p_UpperSlow *p_UpperStop)

に読み込みます

昇りのメソッド(UpMotorクラスのOnUpMotor)と降りのメソッド
(DownMotorクラスのOnDownMotor)で使って

モーターに出力(Motor.txtに出力)します

全く同じ様にドアも通常系で4個のセンサーがありドアの開閉ではエレベーターの昇降と全く同じクラス構造です

後はインスタンス(Position P UpMotor UPMT DownMotor DNMT Door DR OpenMotor OPMT CloseMotor CLMT)

を宣言して仕様に合わせてメソッドを呼び出すだけでO.K.です

終了方法

エレベーターが通常停止しているときに q キーを押します

メンテナンス

異常終了した場合、終了後、Thread_Work フォルダの次のファイルをチェックしてください

Safety.txt

Safety.txt を開いて r にして上書き保存してください

r は通常動作を意味します

s は非常停止を意味します

h は復帰を意味します

Y はスターデルタのスター起動を意味します

Command.txt

Command.txt を開いて q にして上書き保存してください

q は終了を意味します

u は上昇を意味します

d は下降を意味します

o は開を意味します

c は閉を意味します

Y は上階呼びを意味します

y は下階呼びを意味します

H は上階閉を意味します

h は下階閉を意味します

N は信号無しを意味します

PermitCommand.txt

PermitCommand.txt を開いて c にして上書き保存してください

N は命令入力禁止を意味します

c は命令入力許可を意味します

PermitTurnOpen.txt

PermitTurnOpen.txtを開いて N にして上書き保存してください

N は反転開信号入力禁止を意味します

o は反転開信号入力許可を意味します

Motor.txt

Motor.txt を開いて s にして上書き保存してください

s はモーター停止を意味します

j はモーター上昇回転開始を意味します

u はモーター低速上昇回転を意味します

U はモーター高速上昇回転を意味します

k はモーター下降回転開始を意味します

d はモーター低速下降回転を意味します

D はモーター高速下降回転を意味します

h はモーター開回転開始を意味します

o はモーター低速開回転を意味します

O はモーター高速開回転を意味します

t はモーター閉回転開始を意味します

c はモーター低速閉回転を意味します

C はモーター高速閉回転を意味します

Limit.txt

Limit.txt を開いて ynnnyynn にして上書き保存してください

ynnnnyynn は籠が下階停止状態で扉が閉停止状態を意味します

ynnnnyynn は籠が下階停止状態で扉が閉低速区域を意味します

ynnnnyynn は籠が下階停止状態で扉が中間高速区域を意味します

ynnnnyynn は籠が下階停止状態で扉が開低速区域を意味します

ynnnnyynn は籠が下階停止状態で扉が開停止状態を意味します

nnyynnnyynn は籠が下階低速区域で扉が閉停止状態を意味します

nnyynnnyynn は籠が中間高速区域で扉が閉停止状態を意味します

nnyynnnyynn は籠が上階低速区域で扉が閉停止状態を意味します

nnyynnnyynn は籠が上階停止状態で扉が閉停止状態を意味します

nnyynnnyynn は籠が上階停止状態で扉が閉低速区域を意味します

nnyynnnyynn は籠が上階停止状態で扉が中間高速区域を意味します

nnyynnnyynn は籠が上階停止状態で扉が開低速区域を意味します

nnyynnnyynn は籠が上階停止状態で扉が開停止状態を意味します

=====

改造について

マイコン基板とオペアンプ基板と電動機とは、共通地線ムリなので、

フォトカプラでリレーしてください。

EV_Puls.c, EV_Puls.h は、シフトレジスタを、想定しています。

H8 3052F マイコンの書き込みは、RS-232Cケーブルです。

パソコンにRS232C端子がなければ、USB-RS232C変換ケーブルを、探してください。

パソコン-->USB端子-->書き込みケーブル-->RS-232C端子
-->H8 3052F マイコン基板-->フォトカプラ-->シフトレジスタ
-->オペアンプ-->(リレー)-->電動機

失礼いたしました。シフトレジスタはD-FFと共に。

パソコン-->USB端子-->書き込みケーブル-->RS-232C端子
-->H8 3052F マイコン基板-->フォトカプラ-->シュミットトリガ
-->シフトレジスタ-->D-FF-->オペアンプ-->(リレー)-->電動機

共通地線がダメなのは、複数のコンセントの場合でした。

1つのバッテリーから、複数の基板に電源を供給するなら、共通地線が大丈夫かもしれません。

電源アンペア数は実験しないとわかりません。

[1.5V電池を4(?)本直列]×[電池を3(?)列並列]=[電池12(?)本]

はいかがですか？

電池が何本必要か実験しないとわかりません。

電流計を電池のプラス極に直列につないでみるとか。

パソコン-->USB端子-->書き込みケーブル-->RS-232C端子
-->H8 3052F マイコン基板-->シフトレジスタ-->D-FF-->オペアンプ
-->直流モーター

申し遅れましたが、

EV_Puls.c の、関数OnPulsの、
EV_AddressDataSetに入力している8bitの数値は、
上位4bitがアドレスで、下位4bitがモーターのデータなのですが、
テキトーな値です。相手の回路に合わせて自由に決めてください。

main.c の、関数Runの、ID==43の、
110ms毎にパルスを送っているところですが、
パルスをLED上で見えるだけのための、110msです。
D-FFに届くまでに、1周2秒かかり、話にならないので、
11msぐらいにしてみると、いかがでしょうか。

ロボットを真っ直ぐ進めるためには、
D-FFのあとのオペアンプとセンサーの回路を、
フィードバックループにしたいところですが、
スミマセン、未設計です。制御工学でしょうか。

センサーのつなぐ相手はLimit信号です。
現在EV_SimulartorがLimit信号に入力していますが、
H8基板の外のセンサーからLimit信号に取り込む逆向きパルスが必要です。

EV_Controllerに代えてCar_Controllerを新規作成ですね。

H8基板の外のセンサーからLimit信号に取り込む逆向きパルスの受信端は、
EV_InputならぬCar_Inputを、新規作成してください。

プログラムを増やすと反応速度に影響するので、
EVを削除しながらCarに転生してください。

逆向きパルスのクロックは、H8側から出して、
H8側からタイミングを管理します。

EV_Input.c の、GetChar関数で、取り込みができるようになったら、
取り込み端子の1つを、リモコンにしませんか？

Google先生に、「部品」「38kHz」と聞いてみてくださいね。

そういえばH8基板のLCDパネルに文字列は映りましたか？

裏のツマミを回すと、文字が浮かび上がります。

もしもルネサスエディタの HEW で H8-3052F を編集するなら、
途中で聞かれるところで 16文字×2行=32Byte を選択してください。
マッピングのところで、makefile.sub の内容に合わせてください。
ルネサスIDEの HEW は、お試し無料版の場合でも ev9 動きます。

ハンドコンパイルの build.bat は、ev9 システムの手製の開発環境です。
0からの手作り感なら build.bat を選択です。

ev9 で Raspberry Pi でエレベーターが勝手に動く現象は、
ev_receiver.php と ev_follow.php を削除したら解決します。

参照ライブラリ

```

/*****
/*      H8/3048F Include File                               */
/*****

struct st_sam {
    void          *MAR;          /* MAR          */
    unsigned int  ETCR;         /* ETCR         */
    unsigned char IOAR;         /* IOAR         */
    unsigned char DTCR;         /* DTCR         */
};

struct st_fam {
    void          *MARA;         /* MARA         */
    unsigned int  ETCRA;        /* ETCRA        */
    unsigned char IOARA;        /* IOAR         */
    unsigned char DTCRA;        /* DTCRA        */
    void          *MARB;         /* MARB         */
    unsigned int  ETCRB;        /* ETCRB        */
    unsigned char IOARB;        /* IOAR         */
    unsigned char DTCRB;        /* DTCRB        */
};

struct st_itu {
    unsigned char TSTR;         /* TSTR         */
    unsigned char TSNC;         /* TSNC         */
    unsigned char TMDR;         /* TMDR         */
    unsigned char TFCR;         /* TFCR         */
    char          wk[44];       /*              */
    unsigned char TOER;         /* TOER         */
    unsigned char TOCR;         /* TOCR         */
};

struct st_itu0 {
    unsigned char TCR;          /* TCR          */
    unsigned char TIOR;         /* TIOR         */
    unsigned char TIER;         /* TIER         */
    unsigned char TSR;          /* TSR          */
    unsigned int  TCNT;         /* TCNT         */
    unsigned int  GRA;          /* GRA          */
    unsigned int  GRB;          /* GRB          */
};

struct st_itu3 {
    unsigned char TCR;          /* TCR          */
    unsigned char TIOR;         /* TIOR         */
    unsigned char TIER;         /* TIER         */
    unsigned char TSR;          /* TSR          */
    unsigned int  TCNT;         /* TCNT         */
    unsigned int  GRA;          /* GRA          */
    unsigned int  GRB;          /* GRB          */
    unsigned int  BRA;          /* BRA          */
    unsigned int  BRB;          /* BRB          */
    char          wk[2];        /*              */
};

struct st_tpc {
    unsigned char TPMR;         /* TPMR         */
    unsigned char TPCR;         /* TPCR         */
};

```

```

unsigned char  NDERB;      /* NDERB      */
unsigned char  NDERA;      /* NDERA      */
unsigned char  NDRB1;      /* NDRB (H'A4) */
unsigned char  NDRA1;      /* NDRA (H'A5) */
unsigned char  NDRB2;      /* NDRB (H'A6) */
unsigned char  NDRA2;      /* NDRA (H'A7) */
};

struct st_rfshc {          /* struct RFSHC */
    unsigned char  RFSHCR;    /* RFSHCR      */
    unsigned char  RTMCSR;    /* RTMCSR      */
    unsigned char  RTCNT;     /* RTCNT       */
    unsigned char  RTCOR;     /* RTCOR       */
};

struct st_sci {           /* struct SCI  */
    unsigned char  SMR;       /* SMR         */
    unsigned char  BRR;       /* BRR         */
    unsigned char  SCR;       /* SCR         */
    unsigned char  TDR;       /* TDR         */
    unsigned char  SSR;       /* SSR         */
    unsigned char  RDR;       /* RDR         */
    char           wk[2];     /*             */
};

struct st_p1 {           /* struct P1   */
    unsigned char  DDR;       /* P1DDR       */
    char           wk;        /*             */
    unsigned char  DR;        /* P1DR        */
};

struct st_p2 {           /* struct P2   */
    unsigned char  DDR;       /* P2DDR       */
    char           wk1;       /*             */
    unsigned char  DR;        /* P2DR        */
    char           wk2[20];   /*             */
    unsigned char  PCR;       /* P2PCR       */
};

struct st_p4 {           /* struct P4   */
    unsigned char  DDR;       /* P4DDR       */
    char           wk1;       /*             */
    unsigned char  DR;        /* P4DR        */
    char           wk2[18];   /*             */
    unsigned char  PCR;       /* P4PCR       */
};

struct st_p5 {           /* struct P5   */
    unsigned char  DDR;       /* P5DDR       */
    char           wk1;       /*             */
    unsigned char  DR;        /* P5DR        */
    char           wk2[16];   /*             */
    unsigned char  PCR;       /* P5PCR       */
};

struct st_p6 {           /* struct P6   */
    unsigned char  DDR;       /* P6DDR       */

```

```

char          wk;          /* */
unsigned char DR;          /* P6DR */
};

struct st_p7 {              /* struct P7 */
    unsigned char DR;      /* P7DR */
};

struct st_p8 {              /* struct P8 */
    unsigned char DDR;     /* P8DDR */
    char          wk;      /* */
    unsigned char DR;      /* P8DR */
};

struct st_p9 {              /* struct P9 */
    unsigned char DDR;     /* P9DDR */
    char          wk;      /* */
    unsigned char DR;      /* P9DR */
};

struct st_da {              /* struct D/A */
    unsigned char STCR;    /* DASTCR */
    char          wk[127]; /* */
    unsigned char DR0;     /* DADR0 */
    unsigned char DR1;     /* DADR1 */
    unsigned char CR;      /* DACR */
};

struct st_ad {              /* struct A/D */
    unsigned int  DRA;     /* ADDRA */
    unsigned int  DRB;     /* ADDR B */
    unsigned int  DRC;     /* ADDR C */
    unsigned int  DRD;     /* ADDR D */
    unsigned char CSR;     /* ADCSR */
    unsigned char CR;      /* ADCR */
};

struct st_bsc {             /* struct BSC */
    unsigned char CSCR;    /* CSCR */
    char          wk1[140]; /* */
    unsigned char ABWCR;   /* ABWCR */
    unsigned char ASTCR;   /* ASTCR */
    unsigned char WCR;     /* WCR */
    unsigned char WCER;    /* WCER */
    char          wk2[3];  /* */
    unsigned char BRCR;    /* BRCR */
};

struct st_intc {           /* struct INTC */
    unsigned char ISCR;    /* ISCR */
    unsigned char IER;     /* IER */
    unsigned char ISR;     /* ISR */
    char          wk;      /* */
    unsigned char IPRA;    /* IPRA */
    unsigned char IPRB;    /* IPRB */
};

```

```

#define DMAC0A (*(volatile struct st_sam *)0xFFFF20) /* DMAC 0A Addr */
#define DMAC0B (*(volatile struct st_sam *)0xFFFF28) /* DMAC 0B Addr */
#define DMAC1A (*(volatile struct st_sam *)0xFFFF30) /* DMAC 1A Addr */
#define DMAC1B (*(volatile struct st_sam *)0xFFFF38) /* DMAC 1B Addr */
#define DMAC0 (*(volatile struct st_fam *)0xFFFF20) /* DMAC 0 Addr */
#define DMAC1 (*(volatile struct st_fam *)0xFFFF30) /* DMAC 1 Addr */
#define ITU (*(volatile struct st_itu *)0xFFFF60) /* ITU Address*/
#define ITU0 (*(volatile struct st_itu0 *)0xFFFF64) /* ITU0 Address*/
#define ITU1 (*(volatile struct st_itu0 *)0xFFFF6E) /* ITU1 Address*/
#define ITU2 (*(volatile struct st_itu0 *)0xFFFF78) /* ITU2 Address*/
#define ITU3 (*(volatile struct st_itu3 *)0xFFFF82) /* ITU3 Address*/
#define ITU4 (*(volatile struct st_itu3 *)0xFFFF92) /* ITU4 Address*/
#define TPC (*(volatile struct st_tpc *)0xFFFFA0) /* TPC Address*/
#define RFSHC (*(volatile struct st_rfshc *)0xFFFFAC) /* RFSHC Address*/
#define SCI0 (*(volatile struct st_sci *)0xFFFFB0) /* SCI0 Address*/
#define SCI1 (*(volatile struct st_sci *)0xFFFFB8) /* SCI1 Address*/
#define P1 (*(volatile struct st_p1 *)0xFFFFC0) /* P1 Address*/
#define P2 (*(volatile struct st_p2 *)0xFFFFC1) /* P2 Address*/
#define P3 (*(volatile struct st_p1 *)0xFFFFC4) /* P3 Address*/
#define P4 (*(volatile struct st_p4 *)0xFFFFC5) /* P4 Address*/
#define P5 (*(volatile struct st_p5 *)0xFFFFC8) /* P5 Address*/
#define P6 (*(volatile struct st_p6 *)0xFFFFC9) /* P6 Address*/
#define P7 (*(volatile struct st_p7 *)0xFFFFCE) /* P7 Address*/
#define P8 (*(volatile struct st_p8 *)0xFFFFCD) /* P8 Address*/
#define P9 (*(volatile struct st_p9 *)0xFFFFD0) /* P9 Address*/
#define PA (*(volatile struct st_p1 *)0xFFFFD1) /* PA Address*/
#define PB (*(volatile struct st_p1 *)0xFFFFD4) /* PB Address*/
#define DA (*(volatile struct st_da *)0xFFFF5C) /* D/A Address*/
#define AD (*(volatile struct st_ad *)0xFFFFE0) /* A/D Address*/
#define BSC (*(volatile struct st_bsc *)0xFFFF5F) /* BSC Address*/
#define FLMCR (*(volatile unsigned char *)0xFFFF40) /* FLMCR Address*/
#define EBR1 (*(volatile unsigned char *)0xFFFF42) /* EBR1 Address*/
#define EBR2 (*(volatile unsigned char *)0xFFFF43) /* EBR2 Address*/
#define RAMCR (*(volatile unsigned char *)0xFFFF48) /* RAMCR Address*/
#define DIVCR (*(volatile unsigned char *)0xFFFF5D) /* DIVCR Address*/
#define MSTCR (*(volatile unsigned char *)0xFFFF5E) /* MSTCR Address*/
#define MDCR (*(volatile unsigned char *)0xFFFFF1) /* MDCR Address*/
#define SYSCR (*(volatile unsigned char *)0xFFFFF2) /* SYSCR Address*/
#define INTC (*(volatile struct st_intc *)0xFFFFF4) /* INTC Address*/
#define st_itu1 st_itu0 /* Change Struct ITU1 */
#define st_itu2 st_itu0 /* Change Struct ITU2 */
#define st_itu4 st_itu3 /* Change Struct ITU4 */
#define st_p3 st_p1 /* Change Struct P3->P1 */
#define st_pa st_p1 /* Change Struct PA->P1 */
#define st_pb st_p1 /* Change Struct PB->P1 */

```

```
/*=====
```

```
====
```

N9604 Address

```
=====
```

```
==*/
```

```
#define USB9602R      (*(volatile unsigned char *)0x400003)
```

```
#define USB9602D      (*(volatile unsigned char *)0x400001)
```

```
/*=====
```

```
====
```

N9604 Define

```
=====
```

```
==*/
```

```
#define USB_CLKDIV      0x04      /* CLKOUT = 48MHz/4 = 12MHz */
```

```
/* USB1.0リクエスト */
```

```
#define USB_GET_STATUS      0
```

```
#define USB_CLEAR_FEATURE   1
```

```
#define USB_SET_FEATURE     3
```

```
#define USB_SET_ADDRESS     5
```

```
#define USB_GET_DESCRIPTOR   6
```

```
#define USB_SET_DESCRIPTOR   7
```

```
#define USB_GET_CONFIGURATION 8
```

```
#define USB_SET_CONFIGURATION 9
```

```
#define USB_GET_INTERFACE    10
```

```
#define USB_SET_INTERFACE    11
```

```
#define USB_SYNCH_FRAME     12
```

/* ディスクリプタ名 */

```
#define USB_DEVICE 1
#define USB_CONFIGURATION 2
#define USB_XSTRING 3
#define USB_INTERFACE 4
#define USB_ENDPOINT 5
#define USB_HID 0x21
#define USB_HIDREPORT 0x22
#define USB_HIDPHYSICAL 0x23
```

/* HIDリクエスト */

```
#define USB_GET_REPORT 0x01
#define USB_GET_IDLE 0x02
#define USB_GET_PROTOCOL 0x03
#define USB_SET_REPORT 0x09
#define USB_SET_IDLE 0x0A
#define USB_SET_PROTOCOL 0x0B
```

/*=====

====

N9604 Register

=====

==*/

```
#define USB_MCNTL 0x00 /*Main control register */
#define USB_CCONF 0x01 /*Clk.config.register */
#define USB_TCR 0x02 /*Xcvr config.register */
#define USB_RID 0x03 /*Rev.ID register */
```



```

#define USB_FAR          0x04 /*Funcaddressregister */
#define USB_NFSR        0x05 /*Nodefunc st register */
#define USB_MAEV        0x06 /*Mainevent register */
#define USB_MAMSK       0x07 /*Mainmask register */
#define USB_ALTEV       0x08 /*Alt.event register */
#define USB_ALTMSK      0x09 /*ALT mask register */
#define USB_TXEV        0x0A /*TX event register */
#define USB_TXMSK       0x0B /*TX mask register */
#define USB_RXEV        0x0C /*RX event register */
#define USB_RXMSK       0x0D /*RX mask register */
#define USB_NAKEV       0x0E /*NAK event register */
#define USB_NAKMSK      0x0F /*NAK mask register */
#define USB_FWEV        0x10 /*FIFO warning register */
#define USB_FWMSK       0x11 /*FIFO warning mask */
#define USB_FNH         0x12 /*Frame nbr hi register */
#define USB_FNL         0x13 /*Frame nbr lo register */
#define USB_DMACNTRL    0x14 /*DMA control register */

#define USB_EPC0        0x20 /*Endpoint0 register */
#define USB_TXD0        0x21 /*TX data register0 */
#define USB_TXS0        0x22 /*TX status register0 */
#define USB_TXC0        0x23 /*TX commandregister0 */

#define USB_RXD0        0x25 /*RX data register0 */
#define USB_RXS0        0x26 /*RX status register0 */
#define USB_RXC0        0x27 /*RX commandregister0 */

#define USB_EPC1        0x28 /*Endpoint1 register */
#define USB_TXD1        0x29 /*TX data register1 */

```

```

#define USB_TXS1      0x2A /*TX status register1 */
#define USB_TXC1      0x2B /*TX commandregister1 */

#define USB_EPC2      0x2C /*Endpoint2 register */
#define USB_RXD1      0x2D /*RX data register1 */
#define USB_RXS1      0x2E /*RX status register1 */
#define USB_RXC1      0x2F /*RX commandregister1 */

#define USB_EPC3      0x30 /*Endpoint3 register */
#define USB_TXD2      0x31 /*TX data register2 */
#define USB_TXS2      0x32 /*TX status register2 */
#define USB_TXC2      0x33 /*TX commandregister2 */

#define USB_EPC4      0x34 /*Endpoint4 register */
#define USB_RXD2      0x35 /*RX data register2 */
#define USB_RXS2      0x36 /*RX status register2 */
#define USB_RXC2      0x37 /*RX commandregister2 */

#define USB_EPC5      0x38 /*Endpoint5 register */
#define USB_TXD3      0x39 /*TX data register3 */
#define USB_TXS3      0x3A /*TX status register3 */
#define USB_TXC3      0x3B /*TX commandregister3 */

#define USB_EPC6      0x3C /*Endpoint6 register */
#define USB_RXD3      0x3D /*RX data register3 */
#define USB_RXS3      0x3E /*RX status register3 */
#define USB_RXC3      0x3F /*RX commandregister3 */

```

```
/*----- MCNTRL bits -----*/
```

```

#define USB_SRST          0x01 /*softwarereset */
#define USB_DBG          0x02 /*debugmode */
#define USB_VGE          0x04 /*voltage regulator enable*/
#define USB_NAT          0x08 /*node attached */
#define USB_INT_DIS      0x00 /*interrupts disabled */
#define USB_INT_L_O      0x40 /*act lo ints, open drain*/
#define USB_INT_H_P      0x80 /*act hi ints, push pull */
#define USB_INT_L_P      0xC0 /*act lo ints, push pull */

/*----- FAR bits-----*/
#define USB_AD_EN        0x80 /*addressable */

/*----- NFSR bits-----*/
#define USB_RST_ST       0x00 /*reset state */
#define USB_RSM_ST       0x01 /*resume state */
#define USB_OPR_ST       0x02 /*operational state */
#define USB_SUS_ST       0x03 /*suspend state */

/*----- MAEV, MAMSK bits-----*/
#define USB_WARN         0x01 /*warning bit has been set*/
#define USB_ALT          0x02 /*alternate event */
#define USB_TX_EV        0x04 /*transmit event */
#define USB_FRAME        0x08 /*SOF packet received */
#define USB_NAK          0x10 /*NAK event */
#define USB_ULD          0x20 /*unlock locked detected */
#define USB_RX_EV        0x40 /*receive event */
#define USB_INTR_E       0x80 /*master interrupt enable */

```

```
/*----- ALTEV,ALTMSKbits -----*/
```

```
#define USB_EOP          0x08 /*end of packet */
#define USB_SD3         0x10 /*3 ms suspend */
#define USB_SD5         0x20 /*5 ms suspend */
#define USB_RESET_A     0x40 /*reset detected */
#define USB_RESUME_A    0x80 /*resumedetected */
```

```
/*----- TXEV, TXMSKbits -----*/
```

```
#define USB_TXFIFO0     0x01 /*TX_DONE,FIFO 0 */
#define USB_TXFIFO1     0x02 /*TX_DONE,FIFO 1 */
#define USB_TXFIFO2     0x04 /*TX_DONE,FIFO 2 */
#define USB_TXFIFO3     0x08 /*TX_DONE,FIFO 3 */
#define USB_TXUDRN0     0x10 /*TX_URUN,FIFO 0 */
#define USB_TXUDRN1     0x20 /*TX_URUN,FIFO 1 */
#define USB_TXUDRN2     0x40 /*TX_URUN,FIFO 2 */
#define USB_TXUDRN3     0x80 /*TX_URUN,FIFO 3 */
```

```
/*----- RXEV, RXMSKbits -----*/
```

```
#define USB_RXFIFO0     0x01 /*RX_DONE,FIFO 0 */
#define USB_RXFIFO1     0x02 /*RX_DONE,FIFO 1 */
#define USB_RXFIFO2     0x04 /*RX_DONE,FIFO 2 */
#define USB_RXFIFO3     0x08 /*RX_DONE,FIFO 3 */
#define USB_RXOVRN0     0x10 /*RX_OVRN,FIFO 0 */
#define USB_RXOVRN1     0x20 /*RX_OVRN,FIFO 1 */
#define USB_RXOVRN2     0x40 /*RX_OVRN,FIFO 2 */
#define USB_RXOVRN3     0x80 /*RX_OVRN,FIFO 3 */
```

```
/*----- NAKEV, NAKMSKbits -----*/
```

```

#define USB_NAK_I0      0x01 /*IN NAK, FIFO 0 */
#define USB_NAK_I1      0x02 /*IN NAK, FIFO 1 */
#define USB_NAK_I2      0x04 /*IN NAK, FIFO 2 */
#define USB_NAK_I3      0x08 /*IN NAK, FIFO 3 */
#define USB_NAK_O0      0x10 /*OUT NAK, FIFO 0 */
#define USB_NAK_O1      0x20 /*OUT NAK, FIFO 1 */
#define USB_NAK_O2      0x40 /*OUT NAK, FIFO 2 */
#define USB_NAK_O3      0x80 /*OUT NAK, FIFO 3 */

```

```
/*----- EPCX bits -----*/
```

```

#define USB_EP_EN      0x10 /*enablesendpt. (1-6) */
#define USB_ISO        0x20 /*set for isochr.(1-6) */
#define USB_DEF        0x40 /*force def. adr (0 only)*/
#define USB_STALL      0x80 /*force stall handshakes */

```

```
/*----- TXCx bits -----*/
```

```

#define USB_TX_EN      0x01 /*transmitenable */
#define USB_TX_LAST    0x02 /*last data in FIFO */
#define USB_TX_TOGL    0x04 /*specifiesPID used */
#define USB_FLUSH      0x08 /*flushesall FIFO data */
#define USB_IGNIOS     0x80 /* */

```

```
/*----- TXSx bits -----*/
```

```

#define USB_TX_DONE    0x20 /*transmitdone */
#define USB_ACK_STAT   0x40 /*ack status of xmission */

```

```
/*----- RXCx bits -----*/
```

```

#define USB_RX_EN      0x01 /*receive enable */
#define USB_IGN_OUT    0x02 /*ignoreout tokens */

```

```
#define    USB_IGN_SETUP    0x04 /*ignore setup tokens */

/*----- RXS0 bits -----*/

#define    USB_RX_LAST      0x10 /*indicates RCOUNT valid */
#define    USB_RX_TOGL     0x20 /*last pkt was DATA1 PID */
#define    USB_SETUP_RX    0x40 /*setup packet received */
#define    USB_RX_ERR      0x80 /*last packet had an error*/
```

```
/*
```

```
USB N9604 コントロール
```

```
(C)2002 C.I.M
```

```
*/
```

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include "usbn9602.h"
```

```
#include "h83048.h"
```

```
extern void PrintSCI(const char *fmt, ...); /* sci.c */
```

```
static void RegisterSet();
```

```
static void ResetUSB();
```

```
static void WakeupUSB();
```

```
static void rx0();
```

```
static void rx1();
```

```
static void tx0();
```

```
static void tx1();
```

```
static void nako0();
```

```
static void nako1();
```

```
static void naki0();
```

```
static void naki1();
```

```
static void clrfeature();
```

```

static void setfeature();

static void getdescriptor();

static void send_desc_sub(void *ptr,int size);

static void send_desc();

static void getstatus();

static void setconfiguration();

static void SetStallUSB(int adr);

static void ClearStallUSB(int adr);

static void FlushRXC(int no);

static void FlushTXC(int no);

static void TxToggle(int no);

static void WriteUSB(int adr,int data);

static unsigned char ReadUSB(int adr);

static int ReadUSBBurst(int adr,int adr2,char *buff,int cnt);

static int WriteUSBBurst(int adr,int adr2,char *buff,int cnt);

/*-----*/

static int SendTX1();

/*-----*/

int get_inbufflen(void);

void init_usbbuff(void);

int write_inbuff(char *p,int size);

int get_outbufflen(void);

int write_buff(char *p,int size);

int read_outbuff(char *p,int size);

/*-----*/

```



```

static unsigned char usbevent;          /* USB割り込みイベント */
static unsigned char SETADDR;          /* アドレスセット */
static unsigned char configno;        /* コンフィグレーションNO*/
static unsigned char usbbuff[64];     /* 読み込みバッファ */
static unsigned char rx1buff[64];
static unsigned char rx2buff[64];
static unsigned char STALLD;          /* ECPの状態 */
static unsigned char DATA0_1;        /* USB_TXTGLのフラグ */
static char senddesc;                 /* 1 = ディスクリプタ送信中 */
static int desc_size;                 /* ディスクリプタ送信サイズ*/
static char *desc_ptr;                /* ディスクリプタポインタ */

```

```

static const unsigned char epctbl[8] =
{USB_EPC0,USB_EPC1,USB_EPC2,USB_EPC3,USB_EPC4,USB_EPC5,USB_EPC6,USB_EPC0};

```

```

static int txcreg[4] = {USB_TXC0,USB_TXC1,USB_TXC2,USB_TXC3};

```

```

static int rxcreg[4] = {USB_RXC0,USB_RXC1,USB_RXC2,USB_RXC3};

```

```

/*-----*/

```

```

/*-----*/

```

```

static const unsigned char dev_desc[] = {
    0x12,          /* length of this desc. */
    0x01,          /* デバイス・ディスクリプタ1 */
    0x00,0x01,    /* USB Version 1.0 */
    0x00,          /* device class クラス無し */
    0x00,          /* device subclass */
    0x00,          /* device protocol */
    0x08,          /* EP0の最大パッケージサイズ */

```

```

0xfe,0xff,          /* vendorID サンプルなのでとりあえず */
0x10,0x00,         /* productID */
0x01,0x00,         /* revisionID */
0x01,              /* index of manuf. string */
0x01,              /* index of prod. string */
0x02,              /* index of ser. # string */
0x01               /* bNumConfigurations */
};

/* コンフィグレーションディスクリプタ */
static const unsigned char cfg_desc[] = {
    0x09,           /* length of this desc. */
    0x02,           /* コンフィグレーション・ディスクリプタ */
    9+9+7*3,       /* インターフェース/エンドポイントディスクリプタ等の合計長 CFG + IF +
EP*3 */
    0x00,           /*
*/
    0x01,           /* インターフェース数1 */
    0x01,           /* コンフィグレーションは1 */
    0x00,           /* index of config.string */
    0xc0,           /* attr.: self powered D6=自己電源 */
    100,           /* ;max power (100 mA) */
};

/*static const unsigned char if_desc[] = {*/
    0x09,           /* length of this desc. */
    0x04,           /* INTERFACEdescriptor */
    0x00,           /* interfacenumber */
    0x00,           /* alternatesetting */
    0x03,           /* # of (non 0) endpoints */
    0x00,           /* interfaceclass */
};

```

```

0x00,          /* interfacesubclass          */
0x00,          /* interfaceprotocol          */
0x03,          /* index of intf. string      */
/*},*/
/*static const unsigned char endp_desc[] = {*/
/* pipe 0 */
7,             /* length of this desc.      */
5,             /* ENDPOINT descriptor       */
0x81,          /* address (IN)              */
0x02,          /* attributes (BULK)         */
0x40,0x00,     /* max packet size (64)      */
255,          /* interval(ms)              */
/* pipe 1 */
7,             /* length of this desc.      */
5,             /* ENDPOINT descriptor       */
0x02,          /* address (OUT)             */
0x02,          /* attributes (BULK)         */
0x40,0x00,     /* max packet size (64)      */
255,          /* interval(ms)              */

/* pipe 2 (not use) */
7,             /* length of this desc.      */
5,             /* ENDPOINT descriptor       */
0x83,          /* address (IN)              */
0x02,          /* attributes (BULK)         */
0x40,0x00,     /* max packet size (64)      */
255,          /* interval(ms)              */
};

```

```
static const char lang_data[] = {
    4,3,9,4      /* LANGID(English)      */
};
```

```
static const char mfg_str[] = {
    18,3,
    'U',0,'S',0,'B',0,' ',0,'T',0,'E',0,'S',0,'T',0,
};
```

```
static const char nbr_str[] = {
    8,3,
    '1',0,'.',0,'0',0,
};
```

```
static const char int_str[] = {
    34,3,
    'U',0,'S',0,'B',0,' ',0,
    'T',0,'E',0,'S',0,'T',0,' ',0,'P',0,'R',0,'O',0,'G',0,'R',0,'A',0,'M',0,
};
```

```
static void wait(int c)
{
    int    i,j;
    for(j=0;j<c;j++)
    {
        for(i=0;i<0x682;i++)
```

```

    {
    }
}

/*-----*/

/* USB初期化 */
void InitUSB()
{
    init_usbbuff();

    ResetUSB();

    RegisterSet();

    WakeupUSB();

/*    PrintSCI(" REV = %d¥n",ReadUSB(USB_RID)); */

/*    PrintSCI(" CLOCK = %02X¥n",ReadUSB(USB_CCONF)); */

}

static void RegisterSet()
{
    STALLD = 0;

    senddesc = 0;

    DATA0_1 = 0;

    SETADDR = 0;

    WriteUSB(USB_FAR,USB_AD_EN+0);          /* アドレス初期化    */
    WriteUSB(USB_EPC0,USB_EP_EN);          /* EP0をイネーブル  */
    WriteUSB(USB_NAKMSK,USB_NAK_O0);       /* NAK MASKをセット*/
    WriteUSB(USB_TXMSK,USB_TXFIFO0+USB_TXFIFO1+USB_TXFIFO2+USB_TXFIFO3); /* TX MASK

```

をセツト*/

```
WriteUSB(USB_RXMSK,USB_RXFIFO0+USB_RXFIFO1+USB_RXFIFO2+USB_RXFIFO3); /* RX MASK
```

をセツト*/

```
WriteUSB(USB_ALTMSK,USB_SD3+USB_RESET_A); /* ALT MASKをセツト*/
```

```
WriteUSB(USB_MAMSK,USB_INTR_E+USB_RX_EV+USB_NAK+USB_TX_EV+USB_ALT); /*
```

MAIN MASKをセツト*/

```
FlushTXC(0);
```

```
FlushRXC(1);
```

```
FlushTXC(1);
```

```
WriteUSB(USB_TXC1,0);
```

```
WriteUSB(USB_RXC1,0);
```

```
WriteUSB(USB_RXC0,USB_RX_EN); /* RX0をイネ-ブル */
```

}

static void ResetUSB()

{

```
WriteUSB(USB_MCNTRL,USB_SRST+USB_VGE); /* USBリセット 3.3V供給 */
```

```
wait(100); /* 100msec*/
```

```
WriteUSB(USB_MCNTRL,USB_INT_L_P+USB_VGE); /* 割込みはactive low push pull */
```

```
WriteUSB(USB_CCONF,USB_CLKDIM); /* 48MHz/4 = 12MHz*/
```

}

static void WakeupUSB()

{

```
WriteUSB(USB_NFSR,USB_OPR_ST); /* 動作可にする*/
```

```
WriteUSB(USB_MCNTRL,USB_INT_L_P+USB_NAT+USB_VGE); /* USBのノ-トを動作可にする*/
```

}

/* USBポートデータ表示 */

```
/* ※リードすると、ステータスが変わるレジスタもあるので注意 */
```

```
void DispUSBPort()
```

```
{
```

```
    int    i,j;
```

```
    PrintSCI("00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F¥n");
```

```
    for(j=0;j<4;j++)
```

```
    {
```

```
        for(i=0;i<16;i++)
```

```
            PrintSCI("%02X",ReadUSB(i+j*16));
```

```
            PrintSCI("¥n");
```

```
    }
```

```
}
```

```
/*-----*/
```

```
/* USB割り込み */
```

```
#ifdef __GNUC__
```

```
void usb_int() __attribute__((interrupt_handler));
```

```
#endif
```

```
void usb_int()
```

```
{
```

```
    unsigned char  nakevent,rxevent,txevent,altevent;
```

```
    char  reg;
```

```
    usbevent = ReadUSB(USB_MAEV);
```

```
    if( usbevent & USB_NAK )
```

```
    {
```

```

nakevent = ReadUSB(USB_NAKEV);
if( nakevent& USB_NAK_O0)
{
    nako0();
}
if( nakevent& USB_NAK_O1)
{
    nako1();
}
else if( nakevent & USB_NAK_I0 )
{
    naki0();
}
else if( nakevent & USB_NAK_I1 )
{
    naki1();
}
}
else if( usbevent & USB_RX_EV )
{
    rxevent = ReadUSB(USB_RXEV);
    if( rxevent& USB_RXFIFO0)
    {
        rx0();
    }
    else if( rxevent & USB_RXFIFO1 )
    {
        rx1();
    }
}

```



```

}
else if( usbevent & USB_TX_EV )
{
    txevent = ReadUSB(USB_TXEV);
    if( txevent & USB_TXFIFO0 )
    {
        tx0();
    }
    else if( txevent & USB_TXFIFO1 )
    {
        tx1();
    }
}
else if( usbevent & USB_ALT )
{
    altevent = ReadUSB(USB_ALTEV);
    if( altevent & USB_RESET_A )
    {
        /* リセット */
        RegisterSet();
        WakeupUSB();
    }
    else if( altevent & USB_SD3 )
    {
        /* サスペンド */
        WriteUSB(USB_ALTMSK, USB_RESUME_A + USB_RESET_A); /* ALTMSKをセット */
        WriteUSB(USB_NFSR, USB_SUS_ST); /* ノードをサスペンド */
    }
}

```

```
else if( altevent & USB_RESUME_A )
```

```
{
```

```
    /* リジューム */
```

```
    WriteUSB(USB_ALTMSK,USB_SD3+USB_RESET_A);    /* ALTMSKをセット    */
```

```
    WriteUSB(USB_NFSR,USB_OPR_ST);                /* ノードを動作可能にする*/
```

```
}
```

```
}
```

```
}
```

```
/*=====
```

```
=====
```

RXイベントの処理

```
=====
```

```
=====*/
```

```
/* RX0(system) */
```

```
/*
```

リクエストコードの取得

0 byte

D7 ... データ方向 0=ホスト->デバイス, 1=デバイス->ホスト

D6-D5 ... タイプ

0:標準, 1:クラス, 2:ベンダ, 3:予約

D4-D0 ... 受信側

0:デバイス, 1:インターフェイス, 2:エンドポイント, 3:その他

1 byte

特定のリクエスト

2 byte

value

2 byte

index

2 byte

length

*/

static void rx0()

{

unsigned char rxstat;

rxstat = ReadUSB(USB_RXS0);

if(rxstat & USB_SETUP_RX)

{

ReadUSBBurst(USB_RXD0,USB_RXS0,(char*)usbbuff,8);

FlushRXC(0);

FlushTXC(0);

ClearStallUSB(USB_EPC0);

if((usbbuff[0]& 0x60) == 0)

{

/* 標準リクエスト*/

switch(usbbuff[1])

{

case USB_CLEAR_FEATURE

clrfeature();

break;

case USB_GET_CONFIGURATION

WriteUSB(USB_TXD0,configno);

break;

case USB_GET_DESCRIPTOR

getdescriptor();

break;

case USB_GET_STATUS:

```

        getstatus());
        break;
case USB_GET_INTERFACE
    WriteUSB(USB_TXD0,0);
    break;
case USB_SET_ADDRESS:
    WriteUSB(USB_EPC0,USB_DEF);
    SETADDR= usbbuff[2];USB_AD_EN;
    WriteUSB(USB_FAR,SETADDR);
    break;
case USB_SET_CONFIGURATION
    setconfiguration();
    break;
case USB_SET_FEATURE:
    setfeature();
    break;
case USB_SET_INTERFACE
    if( usbbuff[2]!= 0 )
        SetStallUSB(USB_EPC0);
    break;
default:
    /* 未定義 */
    SetStallUSB(USB_EPC0);
    break;
}
}
else if( (usbbuff[0]&0x60) == 0x20 )
{
    /* クラスリクエスト */

```

```

        SetStallUSB(USB_EPC0);
    }
    else if( (usbbuff[0]&0x60) == 0x40 )
    {
        /* ベンダリクエスト*/
        SetStallUSB(USB_EPC0);
    }
    else
    {
        /* 未定義*/
        SetStallUSB(USB_EPC0);
    }
    DATA0_1 |= 1; /* SETUPなのでデータの有無に関係無くDATA1として送信*/
    TxToggle(0);
}
else
{
    if( senddesc )
    {
        senddesc= 0;
    }
    FlushTXC(0);
    WriteUSB(USB_RXC0,USB_RX_EN);
}
}

/*-----*/
/* RX1 受信*/
static void rx1()

```

```

{
    int          cnt;
    unsigned char rxstat;
    rxstat = ReadUSB(USB_RXS1);          /* RX1ステータス取得 */
    if( !(rxstat & (USB_SETUP_RX|USB_RX_ERR)) )          /* SETUP,ERRORパケットでない */
    {
        /* ホストからデータの受信*/
        cnt = ReadUSBBurst(USB_RXD1,USB_RXS1,(char*)rx1buff,64); /* FIFOからデータ取得 */
        write_inbuff((char*)rx1buff,cnt);          /* リングバッファに書き込み*/
    }
    FlushRXC(1);          /* バッファをフラッシュ*/
    WriteUSB(USB_RXC1,USB_RX_EN); /* 受信可に設定 */
}

/*-----*/
/* RX2 受信(not use) */
static void rx2()
{
    unsigned char rxstat;
    rxstat = ReadUSB(USB_RXS2);
}

/*=====
=====
TXイベントの処理
=====
=====*/
/* TX0 送信終了 */
static void tx0()

```

```

{
    unsigned char  txstat;
    txstat = ReadUSB(USB_TXS0);
    if( (txstat & USB_ACK_STAT) && (txstat & USB_TX_DONE) )
    {
        /* ok */
        FlushTXC(0);
        if( senddesc )
        {
            send_desc();
            TxToggle(0);          /* TX0送信可 */
        }
        else
        {
            WriteUSB(USB_RXC0,USB_RX_EN); /* RX0受信可 */
        }
    }
    else
    {
        /* error? */
    }
}

/*-----*/
/* TX1送信終了 */
static void tx1()
{
    unsigned char  txstat;

```

```

txstat = ReadUSB(USB_TXS1);
if( (txstat & USB_ACK_STAT) && (txstat & USB_TX_DONE) )
{
    /*
        送信終了後に次の送信データを送信するようにする
        送信データが無い場合、HOSTには0バイトで送る
    */
    SendTX1();
}
else
{
    /*
        送ったサイズより小さい読み込みが行われた場合、こちらにくる場合がある
    */
}
}

/*=====
=====
                                NAKイベントの処理
=====
=====*/
/*
    NAKイベントは、エラーが発生した場合など再送信する場合処理する
    NAK0しかENABLEにしていないので、それ以外は処理無し
*/
static void nako0()
{
}

```



```
static void nako1()
```

```
{  
}
```

```
static void naki0()
```

```
{  
}
```

```
static void naki1()
```

```
{  
}
```

```
/*=====
```

```
=====
```

標準リクエストの処理

```
=====*/
```

```
/* 選択機器 */
```

```
static void clrfeature()
```

```
{
```

```
    if( (usbbuff[0] & 3) == 2 )
```

```
    {
```

```
        /* エンドポイントデータ*/
```

```
        if((usbbuff[3]&7)!= 0 )
```

```
            ClearStallUSB(epctbl[usbbuff[3]&7]);
```

```
            STALLD &= -1 ^ (1<<(usbbuff[3]&7));
```

```
    }
```

```
}
```

```
static void setfeature()
```

```

{
    if( (usbbuff[0]&3)== 2 )      /* ENDPOINT*/
    {
        /* エンドポイントデータ*/
        if((usbbuff[3]&7)!= 0 )
            SetStallUSB(epctbl[usbbuff[3]&7]);
        STALLD |= (1<<(usbbuff[3]&7));
    }
}

/*-----*/

/* ディスクリプタを返す */
static void getdescriptor()
{
    DATA0_1 &= 0xfe;
    switch( usbbuff[3] )
    {
        case  USB_DEVICE:
            send_desc_sub((void*)dev_desc,dev_desc[0]);
            break;
        case  USB_CONFIGURATION:
            {
                send_desc_sub((void *)cfg_desc,cfg_desc[2]);
                break;
            }
        case  USB_XSTRING:
            {
                switch(usbbuff[2])
                {

```

```

        case 0:
            send_desc_sub((void *)lang_data,lang_data[0]);
            break;
        case 1:
            send_desc_sub((void*)mfg_str,mfg_str[0]);
            break;
        case 2:
            send_desc_sub((void*)nbr_str,nbr_str[0]);
            break;
        case 3:
            send_desc_sub((void*)int_str,int_str[0]);
            break;
    }
    break;
}
default:
{
}
}
}

static void send_desc_sub(void *ptr,int size)
{
    desc_size = (usbbuff[7] << 8) + usbbuff[6];
    if( desc_size > size ) desc_size = size;    /* 受信要求バッファ以上はデータを送らない */
    desc_ptr = ptr;
    senddesc= 1;                               /* ディスクリプタ送信中フラグを立てる*/
    send_desc();
}

```

```
}
```

```
static void send_desc()
```

```
{
```

```
    int    sz;
```

```
    sz = 8;
```

```
    if( desc_size == 0 ) return;
```

```
    if( desc_size <= 8 ) sz = desc_size;
```

```
    sz = WriteUSBurst(USB_TXD0,USB_TXS0,desc_ptr,sz);
```

```
    desc_size -= sz;
```

```
    desc_ptr += sz;
```

```
    if( desc_size == 0 ) senddesc = 0;
```

```
}
```

```
/*-----*/
```

```
/* ステータス */
```

```
static void getstatus()
```

```
{
```

```
    int    data,ep;
```

```
    data = usbbuff[0]&3;
```

```
    if( (data == 0) || (data == 1) )          /* DEVICE,INTERFACE*/
```

```
    {
```

```
        WriteUSB(USB_TXD0,0);
```

```
        WriteUSB(USB_TXD0,0);
```

```
    }
```

```
    else if( data==2 )                        /* エンドポイント*/
```

```
    {
```

```
        ep = usbbuff[3]&7;
```

```
        /* epのSTALL状態を送信*/
```

```
        if( STALLD & (1<<ep) ) WriteUSB(USB_TXD0,1);
```

```

else                WriteUSB(USB_TXD0,0);
}
else
{
    WriteUSB(USB_TXD0,0);
}
}

/*-----*/

static void setconfiguration()
{
    configno = usbbuff[2];
    if( configno== 0 )
    {
        WriteUSB(USB_EPC1,0);           /* EPC1を使用不可*/
        WriteUSB(USB_EPC2,0);           /* EPC2を使用不可*/
        WriteUSB(USB_EPC3,0);           /* EPC3を使用不可*/
        WriteUSB(USB_EPC4,0);           /* EPC4を使用不可*/
        WriteUSB(USB_EPC5,0);           /* EPC5を使用不可*/
        WriteUSB(USB_EPC6,0);           /* EPC6を使用不可*/
    }
    else
    {
        STALLD = 0;
        FlushTXC(1);
        WriteUSB(USB_EPC1,USB_EP_EN+01);           /* EPC1をアドレス1としてイネ-ブル*/
        WriteUSB(USB_TXC1,USB_TX_EN|USB_TX_LAST); /* TX1送信可 */

        FlushRXC(1);
    }
}

```

```
WriteUSB(USB_EPC2,USB_EP_EN+02);          /* EPC2をアドレス2としてイネ-ブル*/
```

```
WriteUSB(USB_RXC1,USB_RX_EN);            /* RX1受信可*/
```

```
/*
```

USB_TX_LASTを立てると、READ時ストールしなくなる。

ただし、データを送る前の最初のREADは0byteになります。

host側は複数のリクエストを同時発行できないので、

リクエストがストールするのはまずいの回避。

```
*/
```

```
}
```

```
}
```

```
/*=====
```

```
=====
```

汎用ルーチン

```
=====
```

```
=====*/
```

```
/* STALLのセットとクリア */
```

```
static void SetStallUSB(int adr)
```

```
{
```

```
    WriteUSB(adr,ReadUSB(adr) | 0x80);
```

```
}
```

```
static void ClearStallUSB(int adr)
```

```
{
```

```
    WriteUSB(adr,ReadUSB(adr)&0x7f);
```

```
}
```

```
/* FIFOのフラッシュ */
```

```
static void FlushRXC(int no)
```

```

{
    WriteUSB(rxcreg[no],USB_FLUSH);
}

```

```

static void FlushTXC(int no)

```

```

{
    int    d;
    d = ReadUSB(txcreg[no]);
    d |= USB_FLUSH;
    WriteUSB(txcreg[no],d);
}

```

```

/* 送信終了フラグセット */

```

```

/* reg = USB_TXC0~6 */

```

```

static void TxToggle(int no)

```

```

{
    unsigned char d;
    d = USB_TX_EN;
    if( DATA0_1 & (1<<no) ) d |= USB_TX_TOGL;
    else                d &= ~USB_TX_TOGL;
    d |= USB_TX_LAST;
    WriteUSB(txcreg[no],d);
    DATA0_1 ^= (1<<no);
}

```

```

/*-----*/

```

```

/* USBのアドレスから読み込み */

```

```

static unsigned char ReadUSB(int adr)

```

```

{
    USB9602R = (unsigned char)adr;
    return( USB9602D );
}

/* USBのアドレスへ書き込み */
static void WriteUSB(int adr,int data)
{
    USB9602R = (unsigned char)adr;
    USB9602D = (unsigned char)data;
}

/* バーストト転送 */
static int ReadUSBBurst(int adr,int adr2,char *buff,int cnt)
{
    int    i;
    int    rcnt;
    USB9602R = (unsigned char)adr;
    for(rcnt=0,i=0;i<cnt;i++)
    {
        if( (ReadUSB(adr2)& 0xf) == 0 )    break;

        USB9602R = (unsigned char)adr;
        *buff = USB9602D;
        buff++;
        rcnt++;
    }
    return(rcnt);
}

```



```
static int WriteUSBurst(int adr,int adr2,char *buff,int cnt)
```

```
{
```

```
    int    i,scnt;
```

```
    for(scnt=0,i=0;i<cnt;i++)
```

```
    {
```

```
        if( (ReadUSB(adr2)& 0x1f) == 0 )    break;
```

```
        USB9602R = (unsigned char)adr;
```

```
        USB9602D= *buff;
```

```
        buff++;
```

```
        scnt++;
```

```
    }
```

```
    return(scnt);
```

```
}
```

```
/*=====
```

```
=====
```

サンプルプログラム

```
=====
```

```
=====*/
```

```
/* TX1送信ルーチン */
```

```
/*
```

USBから一方的に送信できないため、今回は、

TX1送信終了時にバッファ(outbuff)にあるデータを送信します。

よって、HOSTからは定期的にREADを行う。

それ以外のタイミングではEPC1のFIFOバッファのサイズに注意する。

```
*/
```

```
static int SendTX1()
```

```

{
    int      c,cnt,sz,i;

    cnt = 0;
    sz = read_outbuff((char*)rx2buff,64);          /* FIFOは最大64byte */
    FlushTXC(1);                                  /* 送信バッファをフラッシュ*/
    if( sz != 0 )
    {
        cnt = WriteUSBurst(USB_TXD1,USB_TXS1,(char*)rx2buff,sz); /* バースト転送 */
    }
    TxToggle(1);                                  /* 送信終了処理*/
    return(cnt );                                 /* 送信データ数を返すsz==cntのはず*/
}

```

```

/*-----*/

```

```

/*

```

送受信バッファ

inbuffがHOSTから送られてきたデータのバッファ

outbuffがHOSTへ送るデータ用

このサンプルではリングバッファを超えた分は捨てられます。

今回は、256バイト確保しています。バッファがあふれないように

メイン側で処理してください。

```

*/

```

```

#define      USBBUFFLEN  256                      /* バッファのサイズ*/
static int      inpos,inlen;                      /* 入力バッファ位置、サイズ*/
static int      outpos,outlen;                   /* 出力バッファ位置、サイズ*/
static char     inbuff[USBBUFFLEN];             /* 入力リングバッファ */
static char     outbuff[USBBUFFLEN];           /* 出力リングバッファ */

```

```

/*-----*/
/*          バッファの初期化          */
/*-----*/

void init_usbbuff()
{
    inpos = inlen = 0;
    outpos = outlen = 0;
}

/*-----*/
/*          HOSTからの受信バッファへ書き込み          */
/*          char    *p    バッファポインタ          */
/*          int     size  書き込みサイズ          */
/*          戻り値          書き込んだサイズ          */
/*-----*/

int write_inbuff(char *p,int size)
{
    int    i;
    INTC.IER&= (-1^0x20);          /* IRQ5 Disable*/
    for(i=0;i<size;i++)
    {
        if( inlen >= USBBUFFLEN )    break;
        inbuff[inpos] = *p;
        inpos = (inpos + 1)%USBBUFFLEN;
        inlen++;
        p++;
    }
    INTC.IER|= 0x20;          /* IRQ5 Enable*/
}

```

```

    return(i);
}

/*-----*/
/*          送信バッファへ書き込み          */
/* char    *p      バッファポインタ          */
/* int     size   書き込みサイズ            */
/* 戻り値          書き込んだサイズ          */
/*-----*/

int write_buff(char *p,int size)
{
    int    i;

    INTC.IER&= (-1^0x20);          /* IRQ5 Disable*/

    for(i=0;i<size;i++)
    {
        if( outlen >= USBBUFFLEN ) break;

        outbuff[outpos%USBBUFFLEN] = *p;
        outpos = (outpos + 1)%USBBUFFLEN;
        outlen++;

        p++;
    }

    INTC.IER|= 0x20;              /* IRQ5 Enable*/

    return(i);
}

/*-----*/
/*          送信バッファから読み込み          */
/* char    *p      バッファポインタ          */
/* int     size   バッファ最大サイズ        */
/* 戻り値          読み込んだサイズ          */

```

```

/*-----*/
int read_outbuff(char *p,int size)
{
    int i;
    INTC.IER&= (-1^0x20);          /* IRQ5 Disable*/
    for(i=0;outlen>0;i++)
    {
        if( i >= size ) break;
        p[i] = outbuff[ (USBUFFLEN+outpos-outlen)%USBUFFLEN ];
        outlen--;
    }
    INTC.IER|= 0x20;              /* IRQ5 Enable*/
    return(i);
}

/*-----*/
/*          受信バッファから読み込み          */
/* char      *p      バッファポインタ          */
/* int       size   バッファ最大サイズ        */
/* 戻り値     読み込んだサイズ                */
/*-----*/
int read_buff(char *p,int size)
{
    int i;
    INTC.IER&= (-1^0x20);          /* IRQ5 Disable*/
    for(i=0;inlen>0;i++)
    {
        if( i >= size ) break;
        p[i] = inbuff[ (USBUFFLEN+inpos-inlen)%USBUFFLEN ];
        inlen--;
    }
}

```

```

}

INTC.IER|= 0x20;          /* IRQ5 Enable */

return(i);

}

/*-----*/
/*          受信バッファのサイズ取得          */
/*-----*/

int get_inbufflen()
{
    return( inlen%USBUFFLEN );
}

/*-----*/
/*          送信バッファのサイズ取得          */
/*-----*/

int get_outbufflen()
{
    return( outlen%USBUFFLEN );
}

```

```
/*
```

SCI処理

(C)2002 C.I.M

```
*/
```

```
#include <stdio.h>
#include <stdarg.h>
#include "h83048.h"
```

```
static char buff[80];          /* 文字列展開用バッファ(必要なら増やす) */
```

```
/*=====
=====
```

SCI初期化

```
-----
9600bpsパリティ無しSTOP1
=====
=====*/
```

```
void InitSCI()
```

```
{
```

```
    int    i;
    SCI1.SCR= 0;
    SCI1.SMR= 0;          /* パリティ無しSTOP1          */
    SCI1.BRR= 80;        /* 9600bps  3052          */
    for(i=0;i<280;i++) {} /* wait                  */
    SCI1.SCR= 0x30;      /* TE = 1 , RE = 1      */
```

```
i = SCI1.SSR;
```

```
SCI1.SSR= 0x80;          /* Clear Error Flag (TDRE=1) */
```

```
}
```

```
/*=====
```

```
=====
```

SCI出力

```
-----
```

```
=====
```

```
=====*/
```

```
void PutSCI(char c)
```

```
{
```

```
    unsigned char i;
```

```
    while(1 )
```

```
    {
```

```
        i = SCI1.SSR;
```

```
        if(i & 0x80)    break;
```

```
    }
```

```
    SCI1.TDR= c;
```

```
    SCI1.SSR = i&0x7f;
```

```
}
```

```
/*=====
```

```
=====
```

SCI入力

```
-----
```

データを受信するまで待ちつづけます。

```
=====
```

```
=====*/
```



```
char GetSCI()
{
    unsigned char i;
    char c;
    while(1)
    {
        i = SCI1.SSR;
        if(i & 0x40) break;
    }
    c = SCI1.RDR;
    SCI1.SSR = i&0xbf;
    return(c);
}
```

/*=====

=====

SCI入力データチェック

SCIにデータがあるかチェックします。

戻り値 1 = データあり、0 = データなし

=====

=====*/

```
int ScanSCI()
{
    if( SCI1.SSR & 0x40 ) return(1);
    return(0);
}
```

/*=====

=====

SCI文字列出力

書式はprintf()と同じです。バッファは80文字分しか取っていないので、必要ならば、増やしてください。

=====

=====*/

```
void PrintSCI(char *fmt, ...)
```

```
{
```

```
    int    i;
```

```
    va_list arg;
```

```
    va_start(arg, fmt);
```

```
    *buff = '¥0';
```

```
    vsprintf(buff,fmt,arg);
```

```
    va_end(arg);
```

```
    for(i=0;;i++)
```

```
    {
```

```
        if( buff[i]== 0 )    break;
```

```
        if( buff[i] == '¥n' ) PutSCI('¥r');    /* 改行コードは2バイトにして送信 */
```

```
        PutSCI(buff[i]);
```

```
    }
```

```
}
```

```
/*
```

```
    LCD处理
```

```
    (C)2002 C.I.M
```

```
*/
```

```
#include <stdio.h>
```

```
#include <stdarg.h>
```

```
#include "h83048.h"
```

```
void ClearLCD();
```

```
/* PortB (write) b0..3 = LCD(LED) out , bit4 = LCD RS , bit7 = LCD E */
```

```
#define LCD_RS      0x10
```

```
#define LCD_E      0x80
```

```
#define LCDMASK    0x60
```

```
static void wait(int c)
```

```
{
    int    i,j;
    for(j=0;j<c;j++)
    {
        for(i=0;i<0x682;i++) {}
    }
}
```

```

/*=====
=====

LCD BYTE 出力

=====
=====*/

/*
    今回の基板は4bit接続なので、下位4bitのみ出力
*/

static void LCDOut8(short rs,short code)
{
    int    stat;
    int    pb;

    pb = PB.DR;                /* 現在のPBDを退避*/

    if( rs )    stat = (pb & LCDMASK) | LCD_RS;
    else        stat = (pb & LCDMASK);
    PB.DR = code | stat | LCD_E;
    PB.DR = code | stat;
    PB.DR = pb;                /* 元のPBDに復帰*/
    wait(4);
}

/*=====
=====

LCD BYTE 出力(4bit)

=====
=====*/

void LCDOut4(int rs,int code)

```

```

{
    int    stat;
    int    pb;
    char  lb,hb;

    pb = PB.DR;                /* 現在のPBDを退避*/

    if( rs )    stat = (pb & LCDMASK) | LCD_RS;
    else        stat = (pb & LCDMASK);

    hb = ((code>>4)&0xf) | stat;
    lb = (code&0xf) | stat;

    PB.DR = hb | LCD_E;
    PB.DR = hb;

    PB.DR = lb | LCD_E;
    PB.DR = lb;

    PB.DR = pb;                /* 元のPBDに復帰*/
    wait(4);
}

```

```

/*=====
=====

```

LCD コントロール

LCD初期化、表示、クリア

=====

```
=====*/
```

```
void InitLCD()
```

```
{
```

```
    int    i;
```

```
    wait(15);
```

```
    for(i=0;i<3;i++)
```

```
    {
```

```
        LCDOut8(0,0x3);
```

```
    }
```

```
    LCDOut8(0,0x2);
```

```
    LCDOut4(0,0x28);    /* bit4=8/4bit , bit3=1/2line , bit2=large/small */
```

```
    LCDOut4(0,0x10);    /* bit3=Display/Cursor , bit2=Right/Left */
```

```
    LCDOut4(0,0x0e);    /* bit2=display , bit1=cursor , bit0=blink */
```

```
    LCDOut4(0,0x06);
```

```
    LCDOut4(0,0x01);    /* クリア */
```

```
    LCDOut4(0,0x02);    /* カーソルホーム */
```

```
}
```

```
/*=====
```

```
=====
```

LCD クリア

LCD初期化、表示、クリア

```
====*/
```

```
void ClearLCD()
```

```
{  
    LCDOut4(0,0x01);    /* クリア */  
    LCDOut4(0,0x02);    /* カーソルホーム */  
}
```

```
/*=====
```

```
=====
```

LCDキャラクタ表示

'¥n','¥r','¥f'はLCDクリア処理を行います。

```
====*/
```

```
void PutLCD(char c)
```

```
{  
    if( c == '¥f' ) ClearLCD();  
    else if( c == '¥n' ) ClearLCD();  
    else if( c == '¥r' ) ClearLCD();  
    else LCDOut4(1,c);  
}
```

```
/*=====
```

```
=====
```

LCDカーソル移動

x = 0~15

y = 0,1

```
=====  
====*/  
void LocateLCD(int x,int y)  
{  
    LCDOut4(0,0x80 + y*0x40 + x);  
}
```

```
/*=====  
=====
```

LCD文字列表示

パラメータはprintf()と同じです。64文字を超えないように設定してください。
'¥f'はLCDクリア、'¥n'は改行。

```
=====  
====*/  
void PrintLCD(char *fmt,...)  
{  
    int    i;  
    static char buff[64];  
    va_list arg;  
    va_start(arg, fmt);  
    *buff = '¥0';  
    vsprintf(buff,fmt,arg);  
    va_end(arg);  
    for(i=0;;i++)  
    {  
        if( buff[i]== 0 )          break;  
        else if( buff[i]== '¥n' )  LocateLCD(0,1); /* 改行 */  
    }
```



```
else if( buff[i]== '¥r') LCDOut4(0,0x2); /* カーソルホーム */
else if( buff[i]== '¥f') ClearLCD(); /* LCDクリア */
else LCDOut4(1,buff[i]); /* データ出力 */
}
}
```

本文

```

/* C.h */

#define USE_THREAD
/* #define USE_BCC */
/* #define USE_LINUX */
/* #define USE_CENTOS */
/* #define USE_RASPBIAN */
#ifdef USE_CENTOS
#define USE_LINUX
#endif
#ifdef USE_RASPBIAN
#define USE_LINUX
#endif
#ifdef USE_XSERVER
#define USE_BCC
#endif
#ifdef USE_LINUX
#define USE_BCC
#endif
#ifdef USE_BCC
#define TWIN
#endif
#include <stdio.h> /* printf() */
#include <string.h> /* strcmp(), strlen() */
#include <stdlib.h> /* calloc(), free(), rand() */
#ifndef USE_BCC
/* AKI-H8 3052F USB */
#include "h83048.h"
#define SLEEP_PER_SEC 16000.0
#define NOTUSE_FILES
#else
/* time_t, tm, time(), clock(), CLOCKS_PER_SEC */
#include <time.h>
#ifdef USE_LINUX
#include <termios.h> /* kbit() */
#include <unistd.h> /* kbit(), mail, fork() */
#include <fcntl.h> /* kbit() */
/* mail バークレー・ソケット */
#define _OE_SOCKETS
#include <sys/types.h> /* mail */
#include <sys/socket.h> /* mail */
#include <netinet/in.h> /* mail */
#include <netdb.h> /* mail */
#include <arpa/inet.h> /* mail バークレー・ソケット */
#ifdef USE_CENTOS
#include <mysql.h> /* LINUX CentOS GCC MySQL */
#endif
#ifdef USE_RASPBIAN
#include <mysql/mysql.h> /* LINUX Raspbian GCC MySQL */
#endif
#ifdef USE_RASPBIAN
#include <wiringPi.h> /* LINUX Raspberry Pi 3 Model B I/O */
#endif
#else
#ifndef USE_XSERVER
#include <conio.h> /* kbit(), getche() */
#endif
#endif
#define SLEEP_PER_SEC 100000000.0
#endif

```

```
#ifdef USE_LINUX
#define CLEAR system("clear")
#else
#ifdef USE_XSERVER
#define CLEAR system("clear")
#else
#define CLEAR system("cls")
#endif
#endif
#define OK 0
#define NG 1
#define ONE_MORE_TIME 2
#define ON 1
#define OFF 0
```

```
/* Panel.h */
```

```
#ifndef USE_BCC
```

```
/*=====
```

```
外部参照
```

```
=====*/
```

```
/* messagemap.src 内に定義 */
```

```
extern void EnableInterrupt(void);
```

```
extern void DisableInterrupt(void);
```

```
/* lcd.c */
```

```
extern void InitLCD(void);
```

```
extern void PrintLCD(char *fmt,...);
```

```
extern void PutLCD(char c);
```

```
/* usb.c */
```

```
extern void InitUSB(void);
```

```
extern void DispUSBPort(void);
```

```
/* バッファ処理 */
```

```
extern int get_inbufflen(void);
```

```
extern int write_buff(char *p,int size);
```

```
extern int read_buff(char *p,int size);
```

```
/* sci.c */
```

```
extern void InitSCI(void);
```

```
extern void PrintSCI(char *fmt, ...);
```

```
extern int ScanSCI(void);
```

```
extern char GetSCI(void);
```

```
/* main.c内定義 */
```

```
void H8init(void);
```

```
int SetLED(int no,int onoff);
```

```
int GetSW(int no);

#endif

/* 表示を表す列挙体宣言 */

enum PrintF

{

    Panel,

    ClsPnl,

    InputCommand,

    Monitor

};

/* 画面クリア */

void Clear(void);

/* 表示を表す関数のプロトタイプ宣言 */

void PrintF(int mode, char *str);

#ifdef USE_LINUX

int kbhit(void);

#endif

#ifdef USE_XSERVER

int getche(void);

int kbhit(void);

#endif
```

```
/* Panel.c */
```

```
#include "C.h"
```

```
#include "Panel.h"
```

```
/* 画面クリア */
```

```
void Clear(void)
```

```
{
```

```
#ifndef USE_BCC
```

```
    static char buff[64];
```

```
    sprintf(buff,"%f");
```

```
    PrintLCD(buff);
```

```
#else
```

```
    CLEAR;
```

```
#endif
```

```
}
```

```
/* 表示のための関数 */
```

```
void PrintF(int mode, char *str)
```

```
{
```

```
#ifndef USE_BCC
```

```
    static char buff[64];
```

```
#endif
```

```
    switch(mode)
```

```
    {
```

```
        case ClsPnl:
```

```
#ifndef USE_BCC
```

```
            Clear();
```

```

if(strcmp(str, "\n\r0") == 0)
{
    sprintf(str, "%s", "\n\rf");
}

sprintf(buff,"%s", str);
PrintSCI("%s",buff);
PrintLCD(str);
#endif

break;

case Panel:

#ifndef USE_BCC

if(strcmp(str, "\n\r0") == 0)
{
    sprintf(str, "%s", "\n\rf");
}

sprintf(buff,"%s", str);
PrintSCI("%s",buff);
PrintLCD(str);

#else

printf("%s", str);

#endif

break;

case InputCommand:

#ifdef USE_BCC

printf("%s", str);

#endif

break;

case Monitor:

#ifndef USE_BCC

```



```

sprintf(buff,"%s", str);
PrintSCI("%s",buff);
write_buff(buff,strlen(buff)+1);
#else
    printf("%s", str);
#endif
break;
default:
break;
}
return;
}

#ifdef USE_LINUX
int kbhit(void)
{
    struct termios oldt, newt;

    int ch;

    int oldf;

    tcgetattr(STDIN_FILENO, &oldt);
    newt = oldt;
    newt.c_lflag &= ~(ICANON | ECHO);
    tcsetattr(STDIN_FILENO, TCSANOW, &newt);
    oldf = fcntl(STDIN_FILENO, F_GETFL, 0);
    fcntl(STDIN_FILENO, F_SETFL, oldf | O_NONBLOCK);

    ch = getchar();

```

```
tcsetattr(STDIN_FILENO, TCSANOW, &oldt);
```

```
fcntl(STDIN_FILENO, F_SETFL, oldf);
```

```
if (ch != EOF) {
```

```
    ungetc(ch, stdin);
```

```
    return 1;
```

```
}
```

```
return 0;
```

```
}
```

```
#endif
```

```
#ifdef USE_XSERVER
```

```
int getche(void)
```

```
{
```

```
    return 0;
```

```
}
```

```
int kbhit(void)
```

```
{
```

```
    return 0;
```

```
}
```

```
#endif
```

```
/* Timer.h */
```

```
#ifndef Panel_h
```

```
#define Panel_h
```

```
#include "Panel.h"
```

```
#endif
```

```
/* 疑似スレッド定義 */
```

```
#ifdef USE_THREAD
```

```
/* 疑似スレッドに使用する定数の宣言 */
```

```
#define INITCLOCKNO -1
```

```
#define STOPCLOCKNO -2
```

```
/* 構造体宣言 */
```

```
typedef struct tag_Thread
```

```
{
```

```
    /* 疑似スレッドID*/
```

```
    int ID;
```

```
    /* 指定開始時*/
```

```
    double preClock;
```

```
    /* woviClockがpreClockからsetClock秒増えたらRunを呼ぶ*/
```

```
    double setClock;
```

```
    /* Runが呼ばれた回数を調べるために使用(countUpNextRunが呼ばれた回数)*/
```

```
    long count;
```

```
    /* List機能*/
```

```
    struct tag_Thread *previous;
```

```
    struct tag_Thread *next;
```

```
}Thread;
```

```

/* 疑似メソッドとwovi用関数のプロトタイプ宣言 */
/* 宣言の順番は以下の通り */

#endif

double getClock(void);

void SleepMSec(long ms); /* ミリ秒待ち関数 */

#ifdef USE_THREAD

void nextRun(Thread *This, long ms);

void countUpNextRun(Thread *This, long ms);

void Run(Thread *This); /* main.cで内容を定義します */

void Init(Thread *This); /* main.cで内容を定義します */

void Destroy(Thread *This); /* main.cで内容を定義します */

Thread *new_Thread(int id);

void delete_(Thread *This);

void Start(Thread *This);

void Stop(Thread *This);

int Thread_checkAllDelete(void);

int Thread_checkStayAnother(void);

Thread *Thread_getThread(int id);

Thread *Thread_Start(int id);

void Thread_Toggle(int id);

/* タイマ関数 */

void woviRun(void); /* Runを呼ぶタイミング */

void wovilnit(void); /* タイマ初期化関数 */

#endifdef USE_BCC

void InitITU(void); /* タイマ割り込み用 */

void InterruptITU0(void); /* タイマ割り込み用 */

#endif

#ifdef USE_LINUX

```

```
void wovi(double threadPerSec); /* タイマ関数 */  
  
#else  
  
#ifdef USE_XSERVER  
  
void wovi(double threadPerSec); /* タイマ関数 */  
  
#else  
  
void wovi(void); /* タイマ関数 */  
  
#endif  
  
#endif  
  
void initWOVI(void); /* タイマ初期化関数 */  
  
#endif  
  
#ifdef USE_BCC  
  
void PrintCurrentTime(void); /* 現在日時表示 */  
  
void myDateTime(long *mydate, long *mytime); /* 現在日時取得 */  
  
#endif
```

```
/* Timer.c */

#include "C.h"
#include "Timer.h"

/* 時間を表す外部変数宣言 */
double woviClock;

/* 疑似スレッド定義 */
#ifdef USE_THREAD
/* wovi用疑似インスタンス宣言 */
Thread woviThreadFirst;
Thread woviThreadLast;
#endif

/* 時刻取得 */
double getClock(void)
{
    return woviClock;
}

/* ミリ秒待ち関数 */
void SleepMSec(long ms)
{
#ifdef USE_BCC
    double start;
    double set;
    double end;
#endif
}
```

```

start = getClock();
set = ((double) ms) / 1000;
end = start;
while(end < start + set)
{
    end = getClock();
}
#else
double cnt;
double set;
cnt = 0.0;
set = ((double) ms) / 1000;
while(cnt < set)
{
    cnt += 1.0 / SLEEP_PER_SEC;
}
#endif
return;
}

#ifdef USE_THREAD
/* スレッドのvoid Sleep(int ms)の代用 */
void nextRun(Thread *This, long ms)
{
    This->preClock = getClock();
    This->setClock = (((double) ms) / 1000);
    return;
}

```

```
/* スレッドのvoid Sleep(int ms)の代用 */  
void countUpNextRun(Thread *This, long ms)  
{  
    nextRun(This, ms);  
    This->count++;  
}
```

```
#ifndef USE_BCC
```

```
Thread th101;
```

```
Thread th102;
```

```
Thread th111;
```

```
Thread th112;
```

```
Thread th113;
```

```
Thread th114;
```

```
Thread th119;
```

```
Thread th120;
```

```
Thread th121;
```

```
Thread th122;
```

```
Thread th123;
```

```
Thread th130;
```

```
Thread th131;
```

```
Thread th141;
```

```
Thread th142;
```

```
Thread th143;
```

```
Thread th144;
```

```
#endif
```

```
/* スレッドのコンストラクタの代用 */
```

```
Thread *new_Thread(int id)
```



```
{  
  
    Thread *List;  
    Thread *new_List;  
    List = &woviThreadFirst;  
    while(List->next->next != NULL)  
    {  
        List = List->next;  
    }  
  
#ifndef USE_BCC  
    if(id == 1)  
    {  
        new_List = &th101;  
    }  
    else if(id == 2)  
    {  
        new_List = &th102;  
    }  
    else if(id == 11)  
    {  
        new_List = &th111;  
    }  
    else if(id == 12)  
    {  
        new_List = &th112;  
    }  
    else if(id == 13)  
    {  
        new_List = &th113;  
    }  
}
```

```
else if(id == 14)
{
    new_List = &th114;
}
else if(id == 19)
{
    new_List = &th119;
}
else if(id == 20)
{
    new_List = &th120;
}
else if(id == 21)
{
    new_List = &th121;
}
else if(id == 22)
{
    new_List = &th122;
}
else if(id == 23)
{
    new_List = &th123;
}
else if(id == 30)
{
    new_List = &th130;
}
```

```
else if(id == 31)
{
    new_List = &th131;
}
else if(id == 41)
{
    new_List = &th141;
}
else if(id == 42)
{
    new_List = &th142;
}
else if(id == 43)
{
    new_List = &th143;
}
else if(id == 44)
{
    new_List = &th144;
}
#endif

#ifdef USE_BCC
    new_List = (Thread *)calloc(1, sizeof(Thread));
#endif

if(new_List == NULL)
{
    Printf(PANEL, "%n");
    Printf(PANEL, "calloc failed");
    return NULL;
}
```

```

}

new_List->previous = List;

new_List->next = List->next;

new_List->next->previous = new_List;

List->next = new_List;

new_List->preClock = INITCLOCKNO;

new_List->setClock = 0;

new_List->ID = id;

new_List->count = 0;

/* スレッドのvoid init(void)の代用 */

Init(new_List);

return new_List;

}

/* スレッドのデストラクタの代用 */

void delete_(Thread *This)

{

    if(This != NULL)

    {

        Destroy(This);

        This->previous->next = This->next;

        This->next->previous = This->previous;

#ifdef USE_BCC

        free(This);

#endif

        This = NULL;

    }

    return;

}

```

```
/* スレッドのvoid start(void)の代用 */
```

```
void Start(Thread *This)
```

```
{  
    woviClock = getClock();  
    This->preClock = woviClock;  
    return;  
}
```

```
/* スレッドのvoid stop(void)の代用 */
```

```
void Stop(Thread *This)
```

```
{  
    This->preClock = STOPCLOCKNO;  
    return;  
}
```

```
int Thread_checkAllDelete(void)
```

```
{  
    if(woviThreadFirst.next->next == NULL)  
    {  
        return OK;  
    }  
    else  
    {  
        return NG;  
    }  
}
```

```
int Thread_checkStayAnother(void)
```

```
{  
    Thread *checkthread = woviThreadFirst.next->next;  
    int i = 0;  
    while(checkthread != NULL)  
    {  
        checkthread = checkthread->next;  
        i = i + 1;  
    }  
    return i;  
}
```

```
Thread *Thread_getThread(int id)  
{  
    Thread *th;  
  
    if(woviThreadFirst.next->next == NULL)  
    {  
        return NULL;  
    }  
    else  
    {  
        th = woviThreadFirst.next;  
        do  
        {  
            if(th->ID == id)  
            {  
                return th;  
            }  
        }  
    }  
}
```

```

        else
        {
            th = th->next;
        }
    }while(th->next != NULL);
}
return NULL;
}

```

```

Thread *Thread_Start(int id)

```

```

{
    Thread *th;

    th = Thread_getThread(id);
    if(th == NULL)
    {
        th = new_Thread(id);
        Start(th);
    }
    else if(th->preClock == STOPCLOCKNO)
    {
        Start(th);
    }
    return th;
}

```

```

void Thread_Toggle(int id)

```

```

{
    Thread *th;

```

```

th = Thread_getThread(id);
if(th == NULL)
{
    th = new_Thread(id);
    Start(th);
}
else if(th->preClock == STOPCLOCKNO)
{
    Start(th);
}
else
{
    delete_(th);
}
return;
}

```

/ タイマ関数 */*

/ Runを呼ぶタイミング */*

void woviRun(void)

```

{
    Thread *List;
    Thread *next_List;
    List = &woviThreadFirst;
    List = List->next;
    while(List->next != NULL)
    {
        next_List= List->next;
    }
}

```



```

if((List->preClock != INITCLOCKNO) && (List->preClock != STOPCLOCKNO))
{
    if(woviClock >= List->preClock + List->setClock)
    {
        List->preClock= woviClock;
        /* スレッドのvoidrun(void)の代用*/
        Run(List);
    }
}
List = next_List;
}
return;
}

```

/* タイマ初期化関数 */

/* 関数main の冒頭で、 */

/* スレッド構造体リストの両端を初期化します。 */

void wovilnit(void)

```

{
    woviThreadFirst.previous = NULL;
    woviThreadFirst.next = &woviThreadLast;
    woviThreadLast.previous = &woviThreadFirst;
    woviThreadLast.next = NULL;
    return;
}

```

#ifndef USE_BCC

/* タイマ割り込み用 */

/* 関数main の冒頭で、 */

```

/* タイマ割り込みの専用設定をして、*/
/* タイマ0割り込みを立ち上げます。*/
void InitITU(void)
{
    ITU.TSTR = 0x01; /* timer 0 enable */
    ITU.TSNC = 0;
    ITU.TMDR = 0x0;
    ITU.TFCR = 0x0;
    ITU.TOER = 0x0;
    ITU.TOCR = 0xff;
    ITU0.TCR = 0x00; /* 分周なし */
    ITU0.TIOR = 0x88;
    ITU0.TIER = 0x04; /* オーバーフロー割り込み許可 */
    /* AKI-H8 3052F USB の演算速度は 25MHz なので、*/
    /* 25Kカウントすると、1ms です。*/
    ITU0.TCNT = 0xffff - 25000; /* 1 msec interval */
    ITU0.GRA = 0;
    ITU0.GRB = 0;
}

/* タイマ割り込み用 */
/* 周りの関数が関数mainから呼び出されているのに、*/
/* この関数だけは、messagemap.srcのタイマ0割り込みから*/
/* 直接呼び出されています。*/
void InterruptITU0(void)
{
    ITU0.TSR &= 0xfb;

    /* AKI-H8 3052F USB の演算速度は 25MHz なので、*/

```

```

/* 25Kカウント すると、1ms です。 */
ITU0.TCNT = 0xffff - 25000; /* 1 msec interval */

/* woviClock はシステムリセット時からの秒数時計です。 */
woviClock += 0.001;

return;

}

#endif

/* タイマ関数 */
#ifdef USE_LINUX
void wovi(double threadPerSec)
#else
#ifdef USE_XSERVER
void wovi(double threadPerSec)
#else
void wovi(void)
#endif
#endif
#endif
{
#ifdef USE_BCC
    /* woviClock は秒数時計 */
#ifdef USE_LINUX
    /* threadPerSec で受け取る数値が1に近づく程、 */
    /* スピードが上がります。 */
    /* threadPerSec で受け取る数値が大きくなる程、 */
    /* スピード下がります。 */
    /* タイマ割り込み を使用できない時に */
    /* threadPerSec を使用します。 */
    woviClock += 1.0 / threadPerSec;

```

```

#else

#ifdef USE_XSERVER

    woviClock += 1.0 / threadPerSec;

#else

    /* もしくは、 <time.h> の clock() を使用します。 */
    woviClock = (double) (clock() / CLOCKS_PER_SEC);

#endif

#endif

#endif

    woviRun(); /* スレッドのためのRunを呼ぶタイミング */
    return;
}

/* タイマ初期化関数 */
void initWOVI(void)
{
    /* 関数main の冒頭で、 */
    /* 秒数時計woviClock を */
    /* 0.0秒 で初期化します。 */
    woviClock = 0.0;
    /* タイマ割り込み用 */
#ifdef USE_BCC
    /* タイマ0割り込み を立ち上げます。 */
    InitITU();
    /* asmfile.src の 割り込み許可ラベル を呼びます。 */
    EnableInterrupt();
#endif

    /* スレッド構造体リスト の 両端 を初期化します。 */
    wovilnit();
}

```

```

    return;
}

#endif

#ifdef USE_BCC

/* 現在日時表示 */

void PrintCurrentTime(void)
{
    time_t timer;

    struct tm *t_st;

    /* 現在時刻の取得*/

    time(&timer);

    /* 現在時刻を構造体に変換 */

    t_st = localtime(&timer);

    printf("%d", t_st->tm_year+1900);
    if(t_st->tm_mon+1 < 10)
    {
        printf("0%d", t_st->tm_mon+1);
    }
    else
    {
        printf("%d", t_st->tm_mon+1);
    }
    if(t_st->tm_mday < 10)
    {
        printf("0%d", t_st->tm_mday);
    }
}

```

```
}  
else  
{  
    printf("%d", t_st->tm_mday);  
}  
printf(" ");  
if(t_st->tm_hour < 10)  
{  
    printf("0%d", t_st->tm_hour);  
}  
else  
{  
    printf("%d", t_st->tm_hour);  
}  
if(t_st->tm_min < 10)  
{  
    printf("0%d", t_st->tm_min);  
}  
else  
{  
    printf("%d", t_st->tm_min);  
}  
if(t_st->tm_sec < 10)  
{  
    printf("0%d", t_st->tm_sec);  
}  
else  
{
```

```

        printf("%d", t_st->tm_sec);
    }

    return;
}

void myDateTime(long *mydate, long *mytime)
{
    time_t timer;
    struct tm *t_st;

    /* 現在時刻の取得 */
    time(&timer);

    /* 現在時刻を構造体に変換 */
    t_st = localtime(&timer);

    *mydate = t_st->tm_year + 1900;
    *mydate *= 100;
    *mydate += (t_st->tm_mon + 1);
    *mydate *= 100;
    *mydate += t_st->tm_mday;

    *mytime = t_st->tm_hour;
    *mytime *= 100;
    *mytime += t_st->tm_min;
    *mytime *= 100;
    *mytime += t_st->tm_sec;

    return;
}

```

```
}
```

```
#endif
```



```
/* EV_Time.h */
```

```
#ifndef Panel_h
```

```
#define Panel_h
```

```
#include "Panel.h"
```

```
#endif
```

```
#ifndef Timer_h
```

```
#define Timer_h
```

```
#include "Timer.h"
```

```
#endif
```

```
#define OPENTIMEOUT 10
```

```
#define POSITIONTIMEOUT 10
```

```
#define DOORTIMEOUT 10
```

```
/*=====
```

```
時間を表す構造体
```

```
=====*/
```

```
struct EV_Time{
```

```
    double TimeTemp;
```

```
    double *p_TimeTemp;
```

```
    int Permit;
```

```
    int *p_Permit;
```

```
    int tmpTimeSafety;
```

```
    Thread *th;
```

```
};
```

```
/*=====
時間を表すプロトタイプ宣言
=====*/

void EV_Time(struct EV_Time *This, Thread *th);
void SetCurrentTime(struct EV_Time *This);
int GetCurrentTime(struct EV_Time *This);
void WaitSecond(struct EV_Time *This, int num_Second);
void SetPermit(struct EV_Time *This, int P);
int GetPermit(struct EV_Time *This);
void Checkfmove(int *p_check, int *p_fmove, int tmp);
void Wait_ms(struct EV_Time *This, int num);
```

```
/* EV_Time.c */
```

```
#include "C.h"
```

```
#include "EV_Time.h"
```

```
/*=====
```

```
時間を表す関数
```

```
=====*/
```

```
void EV_Time(struct EV_Time *This, Thread *th)
```

```
{
```

```
    This->TimeTemp = 0;
```

```
    This->p_TimeTemp = &This->TimeTemp;
```

```
    SetCurrentTime(This);
```

```
    This->p_Permit = &This->Permit;
```

```
    This->Permit = OFF;
```

```
    This->tmpTimeSafety = 0;
```

```
    This->th = th;
```

```
    /* 戻る */
```

```
    return;
```

```
}
```

```
void SetCurrentTime(struct EV_Time *This)
```

```
{
```

```
    This->p_TimeTemp = &This->TimeTemp;
```

```
    *This->p_TimeTemp = getClock();
```

```
    /* 戻る */
```

```

    return;
}

int GetCurrentTime(struct EV_Time *This)
{
    This->p_TimeTemp = &This->TimeTemp;
    return (int) (getClock() - *This->p_TimeTemp);
}

void WaitSecond(struct EV_Time *This, int num_Second)
{
    nextRun(This->th, (num_Second * 1000));
    /* 戻る */
    return;
}

void SetPermit(struct EV_Time *This, int P)
{
    This->p_Permit = &This->Permit;
    if(P == ON) This->Permit = ON;
    else if(P == OFF) This->Permit = OFF;

    /* 戻る */
    return;
}

int GetPermit(struct EV_Time *This)
{
    This->p_Permit = &This->Permit;

```

```

    return This->Permit;
}

void Checkfmove(int *p_check, int *p_fmove, int tmp)
{
    if(*p_check != tmp){
        *p_fmove= OFF;
        *p_check= tmp;
    }

    /* 戻る */
    return;
}

void Wait_ms(struct EV_Time *This, int num){

    nextRun(This->th, num);

    /* 戻る */
    return;
}

```

```
/* EV_File.h */
```

```
#ifndef Panel_h
```

```
#define Panel_h
```

```
#include "Panel.h"
```

```
#endif
```

```
#define LIMIT 13
```

```
/*=====
   ファイルを表す構造体
   =====*/
```

```
#ifdef NOTUSE_FILES
```

```
typedef struct tag_Handle_EV_Status
```

```
{
```

```
    char safety;
```

```
    char *p_limit;
```

```
    char limit[9];
```

```
    char motor;
```

```
    char command;
```

```
    char permitcommand;
```

```
    char permitturnopen;
```

```
}Handle_EV_Status;
```

```
typedef struct tag_EV_Status
```

```
{
```

```
    Handle_EV_Status *p_status;
```

```
}EV_Status;
```

```

#endif

/* ファイルストリーム */

struct EV_File
{
    FILE *fp;
};

/*=====
   ファイルを表すプロトタイプ宣言
   =====*/

#ifdef NOTUSE_FILES
void new_EV_Status(EV_Status *This);
#endif

void EV_File(struct EV_File *This);
int Write(struct EV_File *This, char *filename, char ch);
int WriteString(struct EV_File *This, char *filename, char *str);
int Read(struct EV_File *This, char *filename, char *p_ch);
int ReadString(struct EV_File *This, char *filename, char *str, int strlength);
int PermitCommand_Read(struct EV_File *This, char *p_PermitCommand);
int PermitCommand_Write(struct EV_File *This, char PermitCommand);
int Command_Read(struct EV_File *This, char *p_Command);
int Command_Write(struct EV_File *This, char Command);
int PermitTurnOpen_Read(struct EV_File *This, char *p_PermitTurnOpen);
int PermitTurnOpen_Write(struct EV_File *This, char PermitTurnOpen);
void Motor_Write(struct EV_File *This, char Motor);
char Motor_Read(struct EV_File *This);
void Limit_Read(struct EV_File *This, char *str);

```

```
/* EV_File.c */
```

```
#include "C.h"
```

```
#include "EV_File.h"
```

```
/*=====
```

```
ファイル不使用時大域オブジェクト宣言
```

```
=====*/
```

```
#ifdef NOTUSE_FILES
```

```
Handle_EV_Status status;
```

```
#endif
```

```
/*=====
```

```
ファイルを表す関数
```

```
=====*/
```

```
#ifdef NOTUSE_FILES
```

```
void new_EV_Status(EV_Status *This)
```

```
{
```

```
    This->p_status = &status;
```

```
    status.safety = 'r';
```

```
    status.p_limit = &status.limit[0];
```

```
    strcpy(status.p_limit, "yynnyynn¥0");
```

```
    status.motor = 's';
```

```
    status.command = 'N';
```

```
    status.permitcommand = 'N';
```

```
    status.permitturnopen = 'N';
```

```
}
```

```
#endif
```



```

void EV_File(struct EV_File *This)
{
    /* 初期値 */
    This->fp = NULL;

    /* 戻る */
    return;
}

#ifdef NOTUSE_FILES
int Write(struct EV_File *This, char *filename, char ch)
{
    switch(filename[0])
    {
        case 'S':
            status.safety= ch;
            break;
        case 'M':
            status.motor= ch;
            break;
        case 'C':
            status.command= ch;
            break;
        case 'P':
            switch(filename[6])
            {
                case 'C':
                    status.permitcommand= ch;

```

```

        break;
    case 'T':
        status.permitturnopen = ch;
        break;
    default:
        break;
}
break;
default:
    break;
}
return OK;
}
#else
int Write(struct EV_File *This, char *filename, char ch)
{
    int Ret = OK;
    if((This->fp = fopen(filename, "w")) == NULL){
        Ret = NG;
    }
    else if(fputc((int) ch, This->fp) == ch){
        fclose(This->fp);
        Ret = OK;
    }
    else{
        fclose(This->fp);
        Ret = NG;
    }
    return Ret;
}

```

```

}

#endif

#ifdef NOTUSE_FILES

int WriteString(struct EV_File *This, char *filename, char *str)
{
    switch(filename[0])
    {
        case 'L':
            status.p_limit = &status.limit[0];
            strcpy(status.p_limit, str);
            break;
        default:
            break;
    }
    return OK;
}

#else

/* 文字列書き込み */

int WriteString(struct EV_File *This, char *filename, char *str)
{
    int Ret = OK;
    if((This->fp = fopen(filename, "w")) == NULL){
        Ret = NG;
    }

    /* ¥nは追記されない */
    else if(fputs(str, This->fp) >= 0){
        fclose(This->fp);
        /* 書き込み成功 */
    }
}

```

```

    Ret = OK;
}
else{
    fclose(This->fp);
    /* 書き込み失敗 */
    Ret = NG;
}
return Ret;
}
#endif

#ifdef NOTUSE_FILES
int Read(struct EV_File *This, char *filename, char *p_ch)
{
    switch(filename[0])
    {
    case 'S':
        *p_ch = status.safety;
        break;
    case 'M':
        *p_ch = status.motor;
        break;
    case 'C':
        *p_ch = status.command;
        break;
    case 'P':
        switch(filename[6])
        {

```

```

    case 'C':
        *p_ch = status.permitcommand;
        break;
    case 'T':
        *p_ch = status.permitturnopen;
        break;
    default:
        break;
}
break;
default:
    break;
}
return OK;
}

#else

int Read(struct EV_File *This, char *filename, char *p_ch)
{
    int Ret = OK;
    if((This->fp = fopen(filename, "r")) == NULL){
        Ret = NG;
    }
    else if((*p_ch = fgetc(This->fp)) == EOF){
        fclose(This->fp);
        Ret = ONE_MORE_TIME;
    }
    else if(*p_ch == '¥n'){
        fclose(This->fp);
        Ret = ONE_MORE_TIME;
    }
}

```

```

}
else if(*p_ch == 'N'){
    fclose(This->fp);
    Ret = ONE_MORE_TIME;
}
else{
    fclose(This->fp);
    Ret = OK;
}
return Ret;
}
#endif

#ifdef NOTUSE_FILES

int ReadString(struct EV_File *This, char *filename, char *str, int strlength)
{
    switch(filename[0])
    {
        case 'L':
            status.p_limit = &status.limit[0];
            strcpy(str, status.p_limit);
            break;
        default:
            break;
    }
    return OK;
}

#else

int ReadString(struct EV_File *This, char *filename, char *str, int strlength)

```

```

{
    int Ret = OK;
    if((This->fp = fopen(filename, "r")) == NULL){
        Ret = NG;
    }
    else if(fgets(str, strlen, This->fp) == NULL){
        fclose(This->fp);
        Ret = ONE_MORE_TIME;
    }
    else if(str[0] == '\n'){
        fclose(This->fp);
        Ret = ONE_MORE_TIME;
    }
    else if(str[0] == 'N'){
        fclose(This->fp);
        Ret = ONE_MORE_TIME;
    }
    else{
        fclose(This->fp);
        Ret = OK;
    }
    return Ret;
}
#endif

```

```

int PermitCommand_Read(struct EV_File *This, char *p_PermitCommand)
{
    int Ret = OK;
    switch(Read(This, "PermitCommand.txt", p_PermitCommand)){

```

```
case NG:
    Printf(ClsPnl, "%nReading Error");
    Ret = NG;
    break;
```

```
case ONE_MORE_TIME:
    Ret = ONE_MORE_TIME;
    break;
```

```
default:
    Ret = OK;
    break;
```

```
}
```

```
return Ret;
```

```
}
```

```
int PermitCommand_Write(struct EV_File *This, char PermitCommand)
```

```
{
```

```
int Ret = OK;
```

```
switch(Write(This, "PermitCommand.txt%0", PermitCommand)){
```

```
case NG:
```

```
    Printf(ClsPnl, "%nWriting Error");
```

```
    Ret = NG;
```

```
    break;
```

```
case OK:
```

```
    Ret = OK;
```

```
    break;
```

```
default:
```

```
    Ret = NG;
```

```
    break;
```



```

    }
    return Ret;
}

int Command_Read(struct EV_File *This, char *p_Command)
{
    int Ret = OK;
    switch(Read(This, "Command.txt¥0", p_Command)){
    case NG:
        Printf(ClsPnl, "¥nReading Error");
        Ret = NG;
        break;
    case ONE_MORE_TIME:
        Ret = ONE_MORE_TIME;
        break;
    default:
        Ret = OK;
        break;
    }
    return Ret;
}

```

```

int Command_Write(struct EV_File *This, char Command)
{
    int Ret = OK;
    switch(Write(This, "Command.txt¥0", Command)){
    case NG:
        Printf(ClsPnl, "¥nWriting Error");
        Ret = NG;
    }
}

```

```

        break;
case OK:
    Ret = OK;
    break;
default:
    Ret = NG;
    break;
}
return Ret;
}

```

```

int PermitTurnOpen_Read(struct EV_File *This, char *p_PermitTurnOpen)
{
    int Ret = OK;
    switch(Read(This, "PermitTurnOpen.txt¥0", p_PermitTurnOpen)){
case NG:
    Printf(ClsPnl, "¥nReading Error");
    Ret = NG;
    break;
case ONE_MORE_TIME:
    Ret = ONE_MORE_TIME;
    break;
default:
    Ret = OK;
    break;
}
return Ret;
}

```

```

int PermitTurnOpen_Write(struct EV_File *This, char PermitTurnOpen)
{
    int Ret = OK;
    switch(Write(This, "PermitTurnOpen.txt¥0", PermitTurnOpen)){
    case NG:
        Printf(ClsPnl, "¥nWriting Error");
        Ret = NG;
        break;
    case OK:
        Ret = OK;
        break;
    default:
        Ret = NG;
        break;
    }
    return Ret;
}

```

```

void Motor_Write(struct EV_File *This, char Motor)
{
    switch(Write(This, "Motor.txt¥0", Motor)){
    case NG:
        Printf(ClsPnl, "¥nWriting Error");
        break;
    case OK:
        return;
        break;
    default:
        break;
    }
}

```

```

}

/* 戻る */
return;
}

char Motor_Read(struct EV_File *This)
{
    char ch;
    char *p_ch;
    ch = '\0';
    p_ch = &ch;

    switch(Read(This, "Motor.txt\0", p_ch)){
    case NG:
        break;
    case ONE_MORE_TIME:
        return '\0';
        break;
    case OK:
        return ch;
        break;
    default:
        break;
    }
    return '\0';
}

```

```
void Limit_Read(struct EV_File *This, char *str)
{
    switch(ReadString(This, "Limit.txt¥0", str, 9)){
    case NG:
        Printf(ClsPnl, "¥nReading Error");
        break;
    case ONE_MORE_TIME:
        return;
        break;
    default:
        return;
        break;
    }

    /* 戻る */
    return;
}
```

```
/* EV_UpDown.h */
```

```
#ifndef Panel_h
```

```
#define Panel_h
```

```
#include "Panel.h"
```

```
#endif
```

```
#ifndef Timer_h
```

```
#define Timer_h
```

```
#include "Timer.h"
```

```
#endif
```

```
#ifndef EV_Time_h
```

```
#define EV_Time_h
```

```
#include "EV_Time.h"
```

```
#endif
```

```
#ifndef EV_File_h
```

```
#define EV_File_h
```

```
#include "EV_File.h"
```

```
#endif
```

```
/*=====
```

```
上昇下降を表す構造体
```

```
=====*/
```

```
struct Position
```

```
{
```

```
    int m_UNSL;
```

```

int m_UNST;
int m_UPSL;
int m_UPST;
/* 下降減速位置 */
int *p_UnderSlow;
/* 下降停止位置 */
int *p_UnderStop;
/* 上昇減速位置 */
int *p_UpperSlow;
/* 上昇停止位置 */
int *p_UpperStop;
/* Sleep用 */
int fstop;
int *p_fstop;
int fmove;
int *p_fmove;
};

```

```

/* 上昇 */

```

```

struct UpMotor

```

```

{
    struct EV_File SF;
    struct EV_File MF;
};

```

```

/* 下降 */

```

```

struct DownMotor

```

```

{
    struct EV_File SF;

```

```

    struct EV_File MF;
};

/* エレベーターの位置仮想ログ */
struct WaitPositionChangeLog
{
    struct EV_File LF;
    char strLimit[9];
    char *p_strLimit;
    /* Sleep用 */
    int tmp;
};

/*=====
  上昇下降を表すプロトタイプ宣言
  =====*/

void Position(struct Position *This);
/* 上昇 */
void UpMotor(struct UpMotor *This);
void OnUpMotor(struct UpMotor *This, struct Position *P, char *p_Safety);
/* 下降 */
void DownMotor(struct DownMotor *This);
void OnDownMotor(struct DownMotor *This, struct Position *P, char *p_Safety);
/* 経過時間 */
void WaitPositionChangeLog(struct WaitPositionChangeLog *This);
void OnInitWaitPositionChangeLog(struct WaitPositionChangeLog *This, struct Position *P);
void OnWaitUpPositionChangeLog(struct WaitPositionChangeLog *This, struct Position *P);
void OnWaitDownPositionChangeLog(struct WaitPositionChangeLog *This, struct Position *P);
/* 上昇 */

```



```
void Up(struct Position *P, struct UpMotor UPMT, struct WaitPositionChangeLog WPCL, char  
*p_Safety);
```

```
/* 下降 */
```

```
void Down(struct Position *P, struct DownMotor DNMT, struct WaitPositionChangeLog WPCL, char  
*p_Safety);
```

```
/* EV_UpDown.c */
```

```
#include "C.h"
```

```
#include "EV_UpDown.h"
```

```
/*=====
```

```
上昇下降を表す関数
```

```
=====*/
```

```
/*
```

```
 * Position
```

```
*/
```

```
void Position(struct Position *This)
```

```
{
```

```
    /* 初期値 */
```

```
    This->m_UNSL = OFF;
```

```
    This->m_UNST = OFF;
```

```
    This->m_UPSL = OFF;
```

```
    This->m_UPST = OFF;
```

```
    This->p_UnderSlow = &This->m_UNSL;
```

```
    This->p_UnderStop = &This->m_UNST;
```

```
    This->p_UpperSlow = &This->m_UPSL;
```

```
    This->p_UpperStop = &This->m_UPST;
```

```
    /* Sleep用 */
```

```
    This->fstop = OFF;
```

```
    This->p_fstop = &This->fstop;
```

```
    This->fmove = OFF;
```

```
    This->p_fmove = &This->fmove;
```

```

    /* 戻る */
    return;
}

/*
 * UpMotor
 */
void UpMotor(struct UpMotor *This)
{
    EV_File(&This->SF);
    EV_File(&This->MF);

    /* 戻る */
    return;
}

/*
 * OnUpMotor
 */
/* 上昇 */
void OnUpMotor(struct UpMotor *This, struct Position *P, char *p_Safety)
{
    if(*P->p_UpperStop == ON){
        /* Sleep用 */
        if(*P->p_fstop == OFF){
            *P->p_fstop = ON;
            /* 現在実行中の命令を外部に報告 */
            Motor_Write(&This->MF, 's');
            Printf(ClsPnl, "STOP");
        }
    }
}

```

```

}
if(*p_Safety== 'Y'){
    *p_Safety= 'r';
    Write(&This->SF, "Safety.txt¥0", 'r');
}
}
else if(*P->p_UpperSlow == ON){
    /* Sleep用 */
    *P->p_fstop= OFF;
    if(*P->p_fmove== OFF){
        *P->p_fmove= ON;
        /* 現在実行中の命令を外部に報告*/
        Motor_Write(&This>MF, 'u');
        Printf(ClsPnl,"UP");
    }
    else if(*p_Safety == 'Y'){
        Motor_Write(&This>MF, 'j');
        *p_Safety= 'r';
        Write(&This->SF, "Safety.txt¥0", 'r');
    }
}
}
else if(*P->p_UnderSlow == OFF){
    /* Sleep用 */
    *P->p_fstop= OFF;
    if(*P->p_fmove== OFF){
        *P->p_fmove= ON;
        /* 現在実行中の命令を外部に報告*/
        Motor_Write(&This>MF, 'U');
        Printf(ClsPnl,"UP Speedy");
    }
}
}

```

```

}

else if(*p_Safety == 'Y'){
    Motor_Write(&This>MF, 'j');
    *p_Safety= 'r';
    Write(&This->SF, "Safety.txt¥0", 'r');
}

}

else if(*P->p_UnderStop == OFF){
    /* Sleep用 */
    *P->p_fstop= OFF;
    if(*P->p_fmove== OFF){
        *P->p_fmove= ON;
        /* 現在実行中の命令を外部に報告*/
        Motor_Write(&This>MF, 'u');
        Printf(ClsPnl,"UP");
    }
    else if(*p_Safety == 'Y'){
        Motor_Write(&This>MF, 'j');
        *p_Safety= 'r';
        Write(&This->SF, "Safety.txt¥0", 'r');
    }
}

else if(*P->p_UnderStop == ON){
    /* Sleep用 */
    *P->p_fstop= OFF;
    if(*P->p_fmove== OFF){
        *P->p_fmove= ON;
        /* 現在実行中の命令を外部に報告*/
        Motor_Write(&This>MF, 'j');

```

```

        Printf(ClsPnl,"UP Start");
    }
    if(*p_Safety== 'Y'){
        *p_Safety= 'r';
        Write(&This->SF, "Safety.txt¥0", 'r');
    }
}

```

```

/* 戻る */

```

```

return;

```

```

}

```

```

/*

```

```

* DownMotor

```

```

*/

```

```

void DownMotor(struct DownMotor *This)

```

```

{

```

```

    EV_File(&This->SF);

```

```

    EV_File(&This->MF);

```

```

/* 戻る */

```

```

return;

```

```

}

```

```

/*

```

```

* OnDownMotor

```

```

*/

```

```

/* 下降 */

```

```

void OnDownMotor(struct DownMotor *This, struct Position *P, char *p_Safety)
{
    if(*P->p_UnderStop == ON){
        /* Sleep用 */
        if(*P->p_fstop == OFF){
            *P->p_fstop = ON;
            /* 現在実行中の命令を外部に報告*/
            Motor_Write(&This>MF, 's');
            Printf(ClsPnl, "STOP");
        }
        if(*p_Safety == 'Y'){
            *p_Safety = 'r';
            Write(&This->SF, "Safety.txt¥0", 'r');
        }
    }
    else if(*P->p_UnderSlow == ON){
        /* Sleep用 */
        *P->p_fstop = OFF;
        if(*P->p_fmove == OFF){
            *P->p_fmove = ON;
            /* 現在実行中の命令を外部に報告*/
            Motor_Write(&This>MF, 'd');
            Printf(ClsPnl, "DOWN");
        }
        else if(*p_Safety == 'Y'){
            Motor_Write(&This>MF, 'k');
            *p_Safety = 'r';
            Write(&This->SF, "Safety.txt¥0", 'r');
        }
    }
}

```

```

}
else if(*P->p_UpperSlow == OFF){
    /* Sleep用 */
    *P->p_fstop= OFF;
    if(*P->p_fmove== OFF){
        *P->p_fmove= ON;
        /* 現在実行中の命令を外部に報告*/
        Motor_Write(&This>MF, 'D');
        Printf(ClsPnl,"DOWN Speedy");
    }
    else if(*p_Safety == 'Y'){
        Motor_Write(&This>MF, 'k');
        *p_Safety= 'r';
        Write(&This->SF, "Safety.txt¥0", 'r');
    }
}
else if(*P->p_UpperStop == OFF){
    /* Sleep用 */
    *P->p_fstop= OFF;
    if(*P->p_fmove== OFF){
        *P->p_fmove= ON;
        /* 現在実行中の命令を外部に報告*/
        Motor_Write(&This>MF, 'd');
        Printf(ClsPnl,"DOWN");
    }
    else if(*p_Safety == 'Y'){
        Motor_Write(&This>MF, 'k');
        *p_Safety= 'r';
        Write(&This->SF, "Safety.txt¥0", 'r');
    }
}

```



```

    }
}
else if(*P->p_UpperStop == ON){
    /* Sleep用 */
    *P->p_fstop= OFF;
    if(*P->p_fmove== OFF){
        *P->p_fmove= ON;
        /* 現在実行中の命令を外部に報告*/
        Motor_Write(&This>MF, 'k');
        Printf(ClsPnl,"DOWN Start");
    }
    if(*p_Safety== 'Y'){
        *p_Safety= 'r';
        Write(&This->SF, "Safety.txt¥0", 'r');
    }
}

/* 戻る */
return;
}

/*
 * WaitPositionChangeLog
 */
/* エレベーターの位置仮想ログ */
void WaitPositionChangeLog(struct WaitPositionChangeLog *This)
{
    This->p_strLimit = &This->strLimit[0];
}

```

```

    /* 戻る */
    return;
}

/*
 * OnInitWaitPositionChangeLog
 */
/* エレベーターの位置仮想ログ */
void OnInitWaitPositionChangeLog(struct WaitPositionChangeLog *This, struct Position *P)
{
    /* センサ初期値 */
    /* リミットスイッチの読み込み */
    This->p_strLimit = &This->strLimit[0];
    Limit_Read(&This->LF, This->p_strLimit);
    This->tmp = ((This->strLimit[0] == 'y') ? ON : OFF);
    Checkfmove(P->p_UnderStop, P->p_fmove, This->tmp);
    This->tmp = ((This->strLimit[1] == 'y') ? ON : OFF);
    Checkfmove(P->p_UnderSlow, P->p_fmove, This->tmp);
    This->tmp = ((This->strLimit[2] == 'y') ? ON : OFF);
    Checkfmove(P->p_UpperSlow, P->p_fmove, This->tmp);
    This->tmp = ((This->strLimit[3] == 'y') ? ON : OFF);
    Checkfmove(P->p_UpperStop, P->p_fmove, This->tmp);

    /* 戻る */
    return;
}

/*

```

```
* OnWaitUpPositionChangeLog
```

```
*/
```

```
/* エレベーターの位置仮想ログ */
```

```
void OnWaitUpPositionChangeLog(struct WaitPositionChangeLog *This, struct Position *P)
```

```
{
```

```
    /* リミットスイッチの読み込み */
```

```
    This->p_strLimit = &This->strLimit[0];
```

```
    Limit_Read(&This->LF, This->p_strLimit);
```

```
    This->tmp = (This->strLimit[0] == 'y') ? ON : OFF;
```

```
    Checkmove(P->p_UnderStop, P->p_fmove, This->tmp);
```

```
    This->tmp = (This->strLimit[1] == 'y') ? ON : OFF;
```

```
    Checkmove(P->p_UnderSlow, P->p_fmove, This->tmp);
```

```
    This->tmp = (This->strLimit[2] == 'y') ? ON : OFF;
```

```
    Checkmove(P->p_UpperSlow, P->p_fmove, This->tmp);
```

```
    This->tmp = (This->strLimit[3] == 'y') ? ON : OFF;
```

```
    Checkmove(P->p_UpperStop, P->p_fmove, This->tmp);
```

```
    /* 戻る */
```

```
    return;
```

```
}
```

```
/*
```

```
* OnWaitDownPositionChangeLog
```

```
*/
```

```
/* エレベーターの位置仮想ログ */
```

```
void OnWaitDownPositionChangeLog(struct WaitPositionChangeLog *This, struct Position *P)
```

```
{
```

```
    /* リミットスイッチの読み込み */
```

```
    This->p_strLimit = &This->strLimit[0];
```

```

Limit_Read(&This->LF, This->p_strLimit);
This->tmp = (This->strLimit[0] == 'y') ? ON : OFF;
Checkfmove(P->p_UnderStop, P->p_fmove, This->tmp);
This->tmp = (This->strLimit[1] == 'y') ? ON : OFF;
Checkfmove(P->p_UnderSlow, P->p_fmove, This->tmp);
This->tmp = (This->strLimit[2] == 'y') ? ON : OFF;
Checkfmove(P->p_UpperSlow, P->p_fmove, This->tmp);
This->tmp = (This->strLimit[3] == 'y') ? ON : OFF;
Checkfmove(P->p_UpperStop, P->p_fmove, This->tmp);

```

```

/* 戻る */

```

```

return;

```

```

}

```

```

/*

```

```

 * Up

```

```

 */

```

```

/* 上昇 */

```

```

void Up(struct Position *P, struct UpMotor UPMT, struct WaitPositionChangeLog WPCL, char
*p_Safety)

```

```

{

```

```

/* 到着まで */

```

```

if(*P->p_UpperStop == OFF){

```

```

    /* エレベーターの位置仮想ログ */

```

```

    OnWaitUpPositionChangeLog(&WPCL, P);

```

```

/* 上昇 */

```

```

    OnUpMotor(&UPMT, P, p_Safety);

```

```
/* エレベーターの位置仮想ログ*/
```

```
OnWaitUpPositionChangeLog(&WPCL, P);
```

```
}
```

```
/* 戻る */
```

```
return;
```

```
}
```

```
/*
```

```
* Down
```

```
*/
```

```
/* 下降 */
```

```
void Down(struct Position *P, struct DownMotor DNMT, struct WaitPositionChangeLog WPCL, char  
*p_Safety)
```

```
{
```

```
/* 到着まで */
```

```
if(*P->p_UnderStop == OFF){
```

```
/* エレベーターの位置仮想ログ*/
```

```
OnWaitDownPositionChangeLog(&WPCL, P);
```

```
/* 下降 */
```

```
OnDownMotor(&DNMT, P, p_Safety);
```

```
/* エレベーターの位置仮想ログ*/
```

```
OnWaitDownPositionChangeLog(&WPCL, P);
```

```
}
```

```
/* 戻る */
```

```
return;
```

}

```
/* EV_OpenClose.h */
```

```
#ifndef Panel_h
```

```
#define Panel_h
```

```
#include "Panel.h"
```

```
#endif
```

```
#ifndef Timer_h
```

```
#define Timer_h
```

```
#include "Timer.h"
```

```
#endif
```

```
#ifndef EV_Time_h
```

```
#define EV_Time_h
```

```
#include "EV_Time.h"
```

```
#endif
```

```
#ifndef EV_File_h
```

```
#define EV_File_h
```

```
#include "EV_File.h"
```

```
#endif
```

```
#ifndef EV_UpDown_h
```

```
#define EV_UpDown_h
```

```
#include "EV_UpDown.h"
```

```
#endif
```

```
/*=====
```

開閉を表す構造体

=====*/

struct Door

```
{
    int m_CLSL;
    int m_CLST;
    int m_OPST;
    int m_OPST;
    /* 閉減速位置 */
    int *p_CloserSlow;
    /* 閉停止位置 */
    int *p_CloserStop;
    /* 開減速位置 */
    int *p_OpennerSlow;
    /* 開停止位置 */
    int *p_OpennerStop;
    /* Sleep用 */
    int fstop;
    int *p_fstop;
    int fmove;
    int *p_fmove;
};
```

/* 開 */

struct OpenMotor

```
{
    struct EV_File SF;
    struct EV_File MF;
};
```



```

/* 閉 */

struct CloseMotor
{
    struct EV_File SF;
    struct EV_File MF;
};

/* エレベーターの位置仮想ログ */
struct WaitDoorChangeLog
{
    struct EV_File TOF;
    char chTurnOpen;
    char *p_chTurnOpen;
    struct EV_File LF;
    char strLimit[9];
    char *p_strLimit;
    /* Sleep用 */
    int tmp;
};

/*=====
  開閉を表すプロトタイプ宣言
=====*/

void Door(struct Door *This);

/* 開 */

void OpenMotor(struct OpenMotor *This);
void OnOpenMotor(struct OpenMotor *This, struct Door *DR, char *p_Safety);

/* 閉 */

```

```
void CloseMotor(struct CloseMotor *This);

void OnCloseMotor(struct CloseMotor *This, struct Door *DR, char *p_Safety);

/* 經過時間 */

void WaitDoorChangeLog(struct WaitDoorChangeLog *This);

void OnInitWaitDoorChangeLog(struct WaitDoorChangeLog *This, struct Door *DR);

void OnWaitOpenDoorChangeLog(struct WaitDoorChangeLog *This, struct Door *DR);

void OnWaitCloseDoorChangeLog(struct WaitDoorChangeLog *This, struct Door *DR);

/* 開 */

void Open(struct Door *DR, struct OpenMotor OPMT, struct WaitDoorChangeLog WDCL, char
*p_Safety);

/* 閉 */

void Close(struct Door *DR, struct CloseMotor CLMT, struct WaitDoorChangeLog WDCL, char
*p_Safety);
```

```
/* EV_OpenClose.c */
```

```
#include "C.h"
```

```
#include "EV_OpenClose.h"
```

```
/*=====
```

```
  開閉を表す関数
```

```
=====*/
```

```
/*
```

```
 * Door
```

```
*/
```

```
void Door(struct Door *This)
```

```
{
```

```
    /* 初期値 */
```

```
    This->m_CLSL = OFF;
```

```
    This->m_CLST = OFF;
```

```
    This->m_OPST = OFF;
```

```
    This->m_OPST = OFF;
```

```
    This->p_CloserSlow = &This->m_CLSL;
```

```
    This->p_CloserStop = &This->m_CLST;
```

```
    This->p_OpennerSlow = &This->m_OPST;
```

```
    This->p_OpennerStop = &This->m_OPST;
```

```
    /* Sleep用 */
```

```
    This->fstop = OFF;
```

```
    This->p_fstop = &This->fstop;
```

```
    This->fmove = OFF;
```

```
    This->p_fmove = &This->fmove;
```

```

    /* 戻る */
    return;
}

/*
 * OpenMotor
 */
void OpenMotor(struct OpenMotor *This)
{
    EV_File(&This->SF);
    EV_File(&This->MF);

    /* 戻る */
    return;
}

/*
 * OnOpenMotor
 */
/* 開 */
void OnOpenMotor(struct OpenMotor *This, struct Door *DR, char *p_Safety)
{
    if(*DR->p_OpennerStop == ON){
        /* Sleep用 */
        if(*DR->p_fstop == OFF){
            *DR->p_fstop = ON;
            /* 現在実行中の命令を外部に報告 */
            Motor_Write(&This->MF, 's');
            Printf(ClsPnl, "STOP");
        }
    }
}

```

```

}
if(*p_Safety== 'Y'){
    *p_Safety= 'r';
    Write(&This->SF, "Safety.txt¥0", 'r');
}
}
else if(*DR->p_OpennerSlow == ON){
    /* Sleep用 */
    *DR->p_fstop= OFF;
    if(*DR->p_fmove == OFF){
        *DR->p_fmove= ON;
        /* 現在実行中の命令を外部に報告*/
        Motor_Write(&This>MF, 'o');
        Printf(ClsPnl,"OPEN");
    }
    else if(*p_Safety == 'Y'){
        Motor_Write(&This>MF, 'h');
        *p_Safety= 'r';
        Write(&This->SF, "Safety.txt¥0", 'r');
    }
}
}
else if(*DR->p_CloserSlow == OFF){
    /* Sleep用 */
    *DR->p_fstop= OFF;
    if(*DR->p_fmove == OFF){
        *DR->p_fmove= ON;
        /* 現在実行中の命令を外部に報告*/
        Motor_Write(&This>MF, 'O');
        Printf(ClsPnl,"OPEN Speedy");
    }
}
}

```

```

}

else if(*p_Safety == 'Y'){
    Motor_Write(&This>MF, 'h');
    *p_Safety= 'r';
    Write(&This->SF, "Safety.txt¥0", 'r');
}

}

else if(*DR->p_CloserStop == OFF){
    /* Sleep用 */
    *DR->p_fstop= OFF;
    if(*DR->p_fmove == OFF){
        *DR->p_fmove= ON;
        /* 現在実行中の命令を外部に報告*/
        Motor_Write(&This>MF, 'o');
        Printf(ClsPnl,"OPEN");
    }
    else if(*p_Safety == 'Y'){
        Motor_Write(&This>MF, 'h');
        *p_Safety= 'r';
        Write(&This->SF, "Safety.txt¥0", 'r');
    }
}

else if(*DR->p_CloserStop == ON){
    /* Sleep用 */
    *DR->p_fstop= OFF;
    if(*DR->p_fmove == OFF){
        *DR->p_fmove= ON;
        /* 現在実行中の命令を外部に報告*/
        Motor_Write(&This>MF, 'h');

```

```

        Printf(ClsPnl,"OPEN Start");
    }
    if(*p_Safety== 'Y'){
        *p_Safety= 'r';
        Write(&This->SF, "Safety.txt¥0", 'r');
    }
}

/* 戻る */
return;
}

/*
 * CloseMotor
 */
void CloseMotor(struct CloseMotor *This)
{
    EV_File(&This->SF);
    EV_File(&This->MF);

    /* 戻る */
    return;
}

/*
 * OnCloseMotor
 */
/* 閉 */

```

```

void OnCloseMotor(struct CloseMotor *This, struct Door *DR, char *p_Safety)
{
    if(*DR->p_CloserStop == ON){
        /* Sleep用 */
        if(*DR->p_fstop== OFF){
            *DR->p_fstop= ON;
            /* 現在実行中の命令を外部に報告*/
            Motor_Write(&This>MF, 's');
            Printf(ClsPnl,"STOP");
        }
        if(*p_Safety== 'Y'){
            *p_Safety= 'r';
            Write(&This->SF, "Safety.txt¥0", 'r');
        }
    }
    else if(*DR->p_CloserSlow == ON){
        /* Sleep用 */
        *DR->p_fstop= OFF;
        if(*DR->p_fmmove == OFF){
            *DR->p_fmmove= ON;
            /* 現在実行中の命令を外部に報告*/
            Motor_Write(&This>MF, 'c');
            Printf(ClsPnl,"CLOSE");
        }
        else if(*p_Safety == 'Y'){
            Motor_Write(&This>MF, 't');
            *p_Safety= 'r';
            Write(&This->SF, "Safety.txt¥0", 'r');
        }
    }
}

```



```

}
else if(*DR->p_OpennerSlow == OFF){
    /* Sleep用 */
    *DR->p_fstop= OFF;
    if(*DR->p_fmove == OFF){
        *DR->p_fmove= ON;
        /* 現在実行中の命令を外部に報告*/
        Motor_Write(&This>MF, 'C');
        Printf(ClsPnl, "CLOSE Speedy");
    }
    else if(*p_Safety == 'Y'){
        Motor_Write(&This>MF, 't');
        *p_Safety= 'r';
        Write(&This->SF, "Safety.txt¥0", 'r');
    }
}
else if(*DR->p_OpennerStop == OFF){
    /* Sleep用 */
    *DR->p_fstop= OFF;
    if(*DR->p_fmove == OFF){
        *DR->p_fmove= ON;
        /* 現在実行中の命令を外部に報告*/
        Motor_Write(&This>MF, 'c');
        Printf(ClsPnl,"CLOSE");
    }
    else if(*p_Safety == 'Y'){
        Motor_Write(&This>MF, 't');
        *p_Safety= 'r';
        Write(&This->SF, "Safety.txt¥0", 'r');
    }
}

```

```

    }
}
else if(*DR->p_OpennerStop == ON){
    /* Sleep用 */
    *DR->p_fstop= OFF;
    if(*DR->p_fmove == OFF){
        *DR->p_fmove= ON;
        /* 現在実行中の命令を外部に報告*/
        Motor_Write(&This>MF, 't');
        Printf(ClsPnl,"CLOSE Start");
    }
    if(*p_Safety== 'Y'){
        *p_Safety= 'r';
        Write(&This->SF, "Safety.txt¥0", 'r');
    }
}

/* 戻る */
return;
}

/*
 * WaitDoorChangeLog
 */
/* エレベーターの位置仮想ログ */
void WaitDoorChangeLog(struct WaitDoorChangeLog *This)
{
    This->p_chTurnOpen = &This->chTurnOpen;
    This->p_strLimit = &This->strLimit[0];
}

```

```

    /* 戻る */
    return;
}

/*
 * OnInitWaitDoorChangeLog
 */
/* エレベーターの位置仮想ログ */
void OnInitWaitDoorChangeLog(struct WaitDoorChangeLog *This, struct Door *DR)
{
    /* センサ初期値 */
    /* リミットスイッチの読み込み */
    This->p_strLimit = &This->strLimit[0];
    Limit_Read(&This->LF, This->p_strLimit);
    This->tmp = (This->strLimit[4] == 'y') ? ON : OFF;
    Checkmove(DR->p_CloserStop, DR->p_fmove, This->tmp);
    This->tmp = (This->strLimit[5] == 'y') ? ON : OFF;
    Checkmove(DR->p_CloserSlow, DR->p_fmove, This->tmp);
    This->tmp = (This->strLimit[6] == 'y') ? ON : OFF;
    Checkmove(DR->p_OpennerSlow, DR->p_fmove, This->tmp);
    This->tmp = (This->strLimit[7] == 'y') ? ON : OFF;
    Checkmove(DR->p_OpennerStop, DR->p_fmove, This->tmp);

    /* 戻る */
    return;
}

```

```

/*
 * OnWaitOpenDoorChangeLog
 */
/* エレベーターの位置仮想ログ */
void OnWaitOpenDoorChangeLog(struct WaitDoorChangeLog *This, struct Door *DR)
{
    /* リミットスイッチの読み込み */
    This->p_strLimit = &This->strLimit[0];
    Limit_Read(&This->LF, This->p_strLimit);
    This->tmp = (This->strLimit[4] == 'y') ? ON : OFF;
    Checkfmove(DR->p_CloserStop, DR->p_fmove, This->tmp);
    This->tmp = (This->strLimit[5] == 'y') ? ON : OFF;
    Checkfmove(DR->p_CloserSlow, DR->p_fmove, This->tmp);
    This->tmp = (This->strLimit[6] == 'y') ? ON : OFF;
    Checkfmove(DR->p_OpennerSlow, DR->p_fmove, This->tmp);
    This->tmp = (This->strLimit[7] == 'y') ? ON : OFF;
    Checkfmove(DR->p_OpennerStop, DR->p_fmove, This->tmp);

    /* 戻る */
    return;
}

/*
 * OnWaitCloseDoorChangeLog
 */
/* エレベーターの位置仮想ログ */
void OnWaitCloseDoorChangeLog(struct WaitDoorChangeLog *This, struct Door *DR)
{
    /* リミットスイッチの読み込み */

```

```

This->p_strLimit = &This->strLimit[0];
Limit_Read(&This->LF, This->p_strLimit);
This->tmp = (This->strLimit[4] == 'y') ? ON : OFF;
Checkfmove(DR->p_CloserStop, DR->p_fmove, This->tmp);
This->tmp = (This->strLimit[5] == 'y') ? ON : OFF;
Checkfmove(DR->p_CloserSlow, DR->p_fmove, This->tmp);
This->tmp = (This->strLimit[6] == 'y') ? ON : OFF;
Checkfmove(DR->p_OpennerSlow, DR->p_fmove, This->tmp);
This->tmp = (This->strLimit[7] == 'y') ? ON : OFF;
Checkfmove(DR->p_OpennerStop, DR->p_fmove, This->tmp);

```

```

/* 戻る */

```

```

return;

```

```

}

```

```

/*

```

```

* Open

```

```

*/

```

```

/* 開 */

```

```

void Open(struct Door *DR, struct OpenMotor OPMT, struct WaitDoorChangeLog WDCL, char
*p_Safety)

```

```

{

```

```

/* 到着まで */

```

```

if(*DR->p_OpennerStop == OFF)

```

```

{

```

```

/* エレベーターの位置仮想ログ */

```

```

OnWaitOpenDoorChangeLog(&WDCL, DR);

```

```

/* 開 */

```

```
OnOpenMotor(&OPMT, DR, p_Safety);
```

```
/* エレベーターの位置仮想ログ*/
```

```
OnWaitOpenDoorChangeLog(&WDCL, DR);
```

```
}
```

```
/* 戻る */
```

```
return;
```

```
}
```

```
/*
```

```
* Close
```

```
*/
```

```
/* 閉 */
```

```
void Close(struct Door *DR, struct CloseMotor CLMT, struct WaitDoorChangeLog WDCL, char *p_Safety)
```

```
{
```

```
/* 到着まで*/
```

```
if(*DR->p_CloserStop == OFF)
```

```
{
```

```
/* エレベーターの位置仮想ログ*/
```

```
OnWaitCloseDoorChangeLog(&WDCL, DR);
```

```
/* 閉 */
```

```
OnCloseMotor(&CLMT, DR, p_Safety);
```

```
/* エレベーターの位置仮想ログ*/
```

```
OnWaitCloseDoorChangeLog(&WDCL, DR);
```

```
}
```

```
/* 戻る */
```

```
return;
```

```
}
```

```
/* EV_Display.h */
```

```
#ifndef Panel_h
```

```
#define Panel_h
```

```
#include "Panel.h"
```

```
#endif
```

```
#ifndef Timer_h
```

```
#define Timer_h
```

```
#include "Timer.h"
```

```
#endif
```

```
#ifndef EV_Time_h
```

```
#define EV_Time_h
```

```
#include "EV_Time.h"
```

```
#endif
```

```
#ifndef EV_File_h
```

```
#define EV_File_h
```

```
#include "EV_File.h"
```

```
#endif
```

```
/*=====
```

```
シミュレータを表す関数のプロトタイプ宣言
```

```
=====*/
```

```
void DisplInput(void);
```

```
void Disp(char ch, char str[9]);
```



```
/* EV_Display.c */
```

```
#include "C.h"
```

```
#include "EV_Display.h"
```

```
void DisplInput(void)
```

```
{
```

```
    /* 入力指示 */
```

```
    Printf(InputCommand, "%nUP = 'u', DOWN = 'd', OPEN = 'o', CLOSE = 'c'");
```

```
    Printf(InputCommand, "%nEMERGENCY = 's', RECOVERY = 'r'");
```

```
    Printf(InputCommand, "%n1st Floor CALL = 'y', 2nd Floor CALL = 'Y'");
```

```
    Printf(InputCommand, "%n1st Floor CLOSE = 'h', 2nd Floor CLOSE = 'H'");
```

```
    Printf(InputCommand, "%nQUIT = 'q'");
```

```
    Printf(InputCommand, "%nCOMMAND>");
```

```
}
```

```
/*
```

```
 * 表示関数
```

```
*/
```

```
void Disp(char ch, char str[9])
```

```
{
```

```
#ifdef USE_BCC
```

```
    int i;
```

```
    /* 画面クリア */
```

```
    CLEAR;
```

```
    if((((ch == 'o') || (ch == 'h')) && (str[3] == 'y') && (str[7] == 'y'))
```

```
        || ((ch == 't') && (str[3] == 'y') && (str[7] == 'n') && (str[6] == 'y'))
```

```
        || ((ch == 's') && (str[3] == 'y') && (str[7] == 'y')))
```

```

{
    for(i = 0; i < 4; i++)
    {
        Printf(Monitor,"¥n0000    0000");
    }
    for(i = 4; i < 12; i++)
    {
        Printf(Monitor,"¥n        ");
    }
}
else if((((ch == 'O') || (ch == 'h')) && (str[3] == 'y') && (str[7] == 'n') && (str[6] == 'y'))
    || (((ch == 'c') || (ch == 't')) && (str[3] == 'y') && (str[7] == 'n') && (str[6] == 'n') && (str[5] ==
'n') && (str[4] == 'n'))
    || ((ch == 's') && (str[3] == 'y') && (str[7] == 'n') && (str[6] == 'y')))
{
    for(i = 0; i < 4; i++)
    {
        Printf(Monitor,"¥n 0000    0000 ");
    }
    for(i = 4; i < 12; i++)
    {
        Printf(Monitor,"¥n        ");
    }
}
else if((ch == 's') && (str[3] == 'y') && (str[7] == 'n') && (str[6] == 'n') && (str[5] == 'n') && (str[4]
== 'n'))
{
    for(i = 0; i < 4; i++)
    {

```

```

        Printf(Monitor,"¥n 0000 0000 ");
    }
    for(i = 4; i < 12; i++)
    {
        Printf(Monitor,"¥n      ");
    }
}
else if((((ch == 'o') || (ch == 'h')) && (str[3] == 'y') && (str[7] == 'n') && (str[6] == 'n') && (str[5] ==
'n') && (str[4] == 'n'))
    || (((ch == 'C') || (ch == 't')) && (str[3] == 'y') && (str[5] == 'y') && (str[4] == 'n'))
    || ((ch == 's') && (str[3] == 'y') && (str[5] == 'y') && (str[4] == 'n')))
{
    for(i = 0; i < 4; i++)
    {
        Printf(Monitor,"¥n 0000 0000 ");
    }
    for(i = 4; i < 12; i++)
    {
        Printf(Monitor,"¥n      ");
    }
}
else if((((ch == 'u') || (ch == 'j')) && (str[3] == 'y') && (str[4] == 'y'))
    || ((ch == 'k') && (str[3] == 'n') && (str[2] == 'y') && (str[4] == 'y'))
    || ((ch == 'h') && (str[3] == 'y') && (str[5] == 'y') && (str[4] == 'n'))
    || (((ch == 'c') || (ch == 't')) && (str[3] == 'y') && (str[4] == 'y'))
    || ((ch == 's') && (str[3] == 'y') && (str[4] == 'y')))
{
    for(i = 0; i < 4; i++)
    {

```

```

        Printf(Monitor,"%n 00000000 ");
    }
    for(i = 4; i < 12; i++)
    {
        Printf(Monitor,"%n      ");
    }
}
else if((((ch == 'o') || (ch == 'h')) && (str[3] == 'n') && (str[2] == 'y') && (str[7] == 'y'))
    || ((ch == 't') && (str[3] == 'n') && (str[2] == 'y') && (str[7] == 'n') && (str[6] == 'y'))
    || ((ch == 's') && (str[3] == 'n') && (str[2] == 'y') && (str[7] == 'y')))
{
    for(i = 0; i < 1; i++)
    {
        Printf(Monitor,"%n      ");
    }
    for(i = 1; i < 5; i++)
    {
        Printf(Monitor,"%n0000  0000");
    }
    for(i = 5; i < 12; i++)
    {
        Printf(Monitor,"%n      ");
    }
}
else if((((ch == 'O') || (ch == 'h')) && (str[3] == 'n') && (str[2] == 'y') && (str[7] == 'n') && (str[6] ==
'y'))
    || (((ch == 'c') || (ch == 't')) && (str[3] == 'n') && (str[2] == 'y') && (str[7] == 'n') && (str[6] ==
'n') && (str[5] == 'n') && (str[4] == 'n'))
    || ((ch == 's') && (str[3] == 'n') && (str[2] == 'y') && (str[7] == 'n') && (str[6] == 'y')))

```

```

{
    for(i = 0; i < 1; i++)
    {
        Printf(Monitor,"¥n      ");
    }
    for(i = 1; i < 5; i++)
    {
        Printf(Monitor,"¥n 0000  0000 ");
    }
    for(i = 5; i < 12; i++)
    {
        Printf(Monitor,"¥n      ");
    }
}
else if((ch == 's') && (str[3] == 'n') && (str[2] == 'y') && (str[7] == 'n') && (str[6] == 'n') && (str[5]
== 'n') && (str[4] == 'n'))
{
    for(i = 0; i < 1; i++)
    {
        Printf(Monitor,"¥n      ");
    }
    for(i = 1; i < 5; i++)
    {
        Printf(Monitor,"¥n 0000  0000 ");
    }
    for(i = 5; i < 12; i++)
    {
        Printf(Monitor,"¥n      ");
    }
}

```

```

    }
}
else if((((ch == 'o') || (ch == 'h')) && (str[3] == 'n') && (str[2] == 'y') && (str[7] == 'n') && (str[6] ==
'n') && (str[5] == 'n') && (str[4] == 'n'))
    || (((ch == 'C') || (ch == 't')) && (str[3] == 'n') && (str[2] == 'y') && (str[5] == 'y') && (str[4] ==
'n'))
    || ((ch == 's') && (str[3] == 'n') && (str[2] == 'y') && (str[5] == 'y') && (str[4] == 'n'))))
{
    for(i = 0; i < 1; i++)
    {
        Printf(Monitor, "%n      ");
    }
    for(i = 1; i < 5; i++)
    {
        Printf(Monitor, "%n 0000 0000 ");
    }
    for(i = 5; i < 12; i++)
    {
        Printf(Monitor, "%n      ");
    }
}
else if((((ch == 'U') || (ch == 'j')) && (str[3] == 'n') && (str[2] == 'y') && (str[4] == 'y'))
    || (((ch == 'd') || (ch == 'k')) && (str[3] == 'n') && (str[2] == 'n') && (str[1] == 'n') && (str[0] ==
'n') && (str[4] == 'y'))
    || ((ch == 'h') && (str[3] == 'n') && (str[2] == 'y') && (str[5] == 'y') && (str[4] == 'n'))
    || (((ch == 'c') || (ch == 't')) && (str[3] == 'n') && (str[2] == 'y') && (str[4] == 'y'))
    || ((ch == 's') && (str[3] == 'n') && (str[2] == 'y') && (str[4] == 'y'))))
{
    for(i = 0; i < 2; i++)

```

```

    {
        Printf(Monitor,"%n      ");
    }
    for(i = 2; i < 6; i++)
    {
        Printf(Monitor,"%n 00000000 ");
    }
    for(i = 6; i < 12; i++)
    {
        Printf(Monitor,"%n      ");
    }
}
else if((((ch == 'o') || (ch == 'h')) && (str[3] == 'n') && (str[2] == 'n') && (str[1] == 'n') && (str[0] ==
'n') && (str[7] == 'y'))
    || ((ch == 't') && (str[3] == 'n') && (str[2] == 'n') && (str[1] == 'n') && (str[0] == 'n') && (str[7]
== 'n') && (str[6] == 'y'))
    || ((ch == 's') && (str[3] == 'n') && (str[2] == 'n') && (str[1] == 'n') && (str[0] == 'n') && (str[7]
== 'y'))))
{
    for(i = 0; i < 4; i++)
    {
        Printf(Monitor,"%n      ");
    }
    for(i = 4; i < 8; i++)
    {
        Printf(Monitor,"%n0000  0000");
    }
    for(i = 8; i < 12; i++)
    {

```

```

        Printf(Monitor,"%n      ");
    }
}
else if((((ch == 'O') || (ch == 'h')) && (str[3] == 'n') && (str[2] == 'n') && (str[1] == 'n') && (str[0] ==
'n') && (str[7] == 'n') && (str[6] == 'y'))
    || (((ch == 'c') || (ch == 't')) && (str[3] == 'n') && (str[2] == 'n') && (str[1] == 'n') && (str[0] ==
'n') && (str[7] == 'n') && (str[6] == 'n') && (str[5] == 'n') && (str[4] == 'n'))
    || ((ch == 's') && (str[3] == 'n') && (str[2] == 'n') && (str[1] == 'n') && (str[0] == 'n') && (str[7]
== 'n') && (str[6] == 'y')))
{
    for(i = 0; i < 4; i++)
    {
        Printf(Monitor,"%n      ");
    }
    for(i = 4; i < 8; i++)
    {
        Printf(Monitor,"%n 0000  0000 ");
    }
    for(i = 8; i < 12; i++)
    {
        Printf(Monitor,"%n      ");
    }
}
else if((ch == 's') && (str[3] == 'n') && (str[2] == 'n') && (str[1] == 'n') && (str[0] == 'n') && (str[7]
== 'n') && (str[6] == 'n') && (str[5] == 'n') && (str[4] == 'n'))
{
    for(i = 0; i < 4; i++)
    {
        Printf(Monitor,"%n      ");
    }
}

```



```

}
for(i = 4; i < 8; i++)
{
    Printf(Monitor,"¥n 0000 0000 ");
}
for(i = 8; i < 12; i++)
{
    Printf(Monitor,"¥n      ");
}
}
else if((((ch == 'o') || (ch == 'h')) && (str[3] == 'n') && (str[2] == 'n') && (str[1] == 'n') && (str[0] ==
'n') && (str[7] == 'n') && (str[6] == 'n') && (str[5] == 'n') && (str[4] == 'n'))
    || (((ch == 'C') || (ch == 't')) && (str[3] == 'n') && (str[2] == 'n') && (str[1] == 'n') && (str[0] ==
'n') && (str[5] == 'y') && (str[4] == 'n'))
    || ((ch == 's') && (str[3] == 'n') && (str[2] == 'n') && (str[1] == 'n') && (str[0] == 'n') && (str[5]
== 'y') && (str[4] == 'n')))
{
    for(i = 0; i < 4; i++)
    {
        Printf(Monitor,"¥n      ");
    }
    for(i = 4; i < 8; i++)
    {
        Printf(Monitor,"¥n 0000 0000 ");
    }
    for(i = 8; i < 12; i++)
    {
        Printf(Monitor,"¥n      ");
    }
}

```

```

    }
}
else if(((ch == 'h') && (str[3] == 'n') && (str[2] == 'n') && (str[1] == 'n') && (str[0] == 'n') && (str[5]
== 'y') && (str[4] == 'n'))
    || (((ch == 'c') || (ch == 't')) && (str[3] == 'n') && (str[2] == 'n') && (str[1] == 'n') && (str[0] ==
'n') && (str[4] == 'y'))
    || ((ch == 's') && (str[3] == 'n') && (str[2] == 'n') && (str[1] == 'n') && (str[0] == 'n') && (str[4]
== 'y'))))
{
    for(i = 0; i < 4; i++)
    {
        Printf(Monitor, "%n      ");
    }
    for(i = 4; i < 8; i++)
    {
        Printf(Monitor, "%n 00000000 ");
    }
    for(i = 8; i < 12; i++)
    {
        Printf(Monitor, "%n      ");
    }
}
else if((((ch == 'o') || (ch == 'h')) && (str[1] == 'y') && (str[0] == 'n') && (str[7] == 'y'))
    || ((ch == 't') && (str[1] == 'y') && (str[0] == 'n') && (str[7] == 'n') && (str[6] == 'y'))
    || ((ch == 's') && (str[1] == 'y') && (str[0] == 'n') && (str[7] == 'y'))))
{
    for(i = 0; i < 6; i++)
    {
        Printf(Monitor, "%n      ");
    }
}

```

```

}
for(i = 6; i < 10; i++)
{
    Printf(Monitor, "%n0000 0000");
}
for(i = 10; i < 12; i++)
{
    Printf(Monitor, "%n ");
}
}
else if((((ch == 'O') || (ch == 'h')) && (str[1] == 'y') && (str[0] == 'n') && (str[7] == 'n') && (str[6] ==
'y'))
    || (((ch == 'c') || (ch == 't')) && (str[1] == 'y') && (str[0] == 'n') && (str[7] == 'n') && (str[6] ==
'n') && (str[5] == 'n') && (str[4] == 'n'))
    || ((ch == 's') && (str[1] == 'y') && (str[0] == 'n') && (str[7] == 'n') && (str[6] == 'y'))))
{
    for(i = 0; i < 6; i++)
    {
        Printf(Monitor, "%n ");
    }
    for(i = 6; i < 10; i++)
    {
        Printf(Monitor, "%n 0000 0000 ");
    }
    for(i = 10; i < 12; i++)
    {
        Printf(Monitor, "%n ");
    }
}
}

```

```

else if((ch == 's') && (str[1] == 'y') && (str[0] == 'n') && (str[7] == 'n') && (str[6] == 'n') && (str[5]
== 'n') && (str[4] == 'n'))
{
    for(i = 0; i < 6; i++)
    {
        Printf(Monitor,"¥n      ");
    }
    for(i = 6; i < 10; i++)
    {
        Printf(Monitor,"¥n 0000 0000 ");
    }
    for(i = 10; i < 12; i++)
    {
        Printf(Monitor,"¥n      ");
    }
}
else if((((ch == 'o') || (ch == 'h')) && (str[1] == 'y') && (str[0] == 'n') && (str[7] == 'n') && (str[6] ==
'n') && (str[5] == 'n') && (str[4] == 'n'))
    || (((ch == 'C') || (ch == 't')) && (str[1] == 'y') && (str[0] == 'n') && (str[5] == 'y') && (str[4] ==
'n'))
    || ((ch == 's') && (str[1] == 'y') && (str[0] == 'n') && (str[5] == 'y') && (str[4] == 'n'))))
{
    for(i = 0; i < 6; i++)
    {
        Printf(Monitor,"¥n      ");
    }
    for(i = 6; i < 10; i++)
    {
        Printf(Monitor,"¥n 0000 0000 ");
    }
}

```

```

}
for(i = 10; i < 12; i++)
{
    Printf(Monitor, "%n    ");
}
}
else if((((ch == 'u') || (ch == 'j')) && (str[3] == 'n') && (str[2] == 'n') && (str[1] == 'n') && (str[0] ==
'n') && (str[4] == 'y'))
    || (((ch == 'D') || (ch == 'k')) && (str[1] == 'y') && (str[0] == 'n') && (str[4] == 'y'))
    || ((ch == 'h') && (str[1] == 'y') && (str[0] == 'n') && (str[5] == 'y') && (str[4] == 'n'))
    || (((ch == 'c') || (ch == 't')) && (str[1] == 'y') && (str[0] == 'n') && (str[4] == 'y'))
    || ((ch == 's') && (str[1] == 'y') && (str[0] == 'n') && (str[4] == 'y')))
{
    for(i = 0; i < 6; i++)
    {
        Printf(Monitor, "%n    ");
    }
    for(i = 6; i < 10; i++)
    {
        Printf(Monitor, "%n 00000000 ");
    }
    for(i = 10; i < 12; i++)
    {
        Printf(Monitor, "%n    ");
    }
}
else if((((ch == 'o') || (ch == 'h')) && (str[0] == 'y') && (str[7] == 'y'))
    || ((ch == 't') && (str[0] == 'y') && (str[7] == 'n') && (str[6] == 'y'))

```

```

    || ((ch == 's') && (str[0] == 'y') && (str[7] == 'y')))
{
    for(i = 0; i < 8; i++)
    {
        Printf(Monitor,"%n      ");
    }
    for(i = 8; i < 12; i++)
    {
        Printf(Monitor,"%n0000    0000");
    }
}

else if((((ch == 'O') || (ch == 'h')) && (str[0] == 'y') && (str[7] == 'n') && (str[6] == 'y'))
    || (((ch == 'c') || (ch == 't')) && (str[0] == 'y') && (str[7] == 'n') && (str[6] == 'n') && (str[5] ==
'n') && (str[4] == 'n'))
    || ((ch == 's') && (str[0] == 'y') && (str[7] == 'n') && (str[6] == 'y')))
{
    for(i = 0; i < 8; i++)
    {
        Printf(Monitor,"%n      ");
    }
    for(i = 8; i < 12; i++)
    {
        Printf(Monitor,"%n 0000    0000 ");
    }
}

else if((ch == 's') && (str[0] == 'y') && (str[7] == 'n') && (str[6] == 'n') && (str[5] == 'n') && (str[4]
== 'n'))
{
    for(i = 0; i < 8; i++)

```

```

{
    Printf(Monitor,"%n      ");
}
for(i = 8; i < 12; i++)
{
    Printf(Monitor,"%n 0000 0000 ");
}
}
else if((((ch == 'o') || (ch == 'h')) && (str[0] == 'y') && (str[7] == 'n') && (str[6] == 'n') && (str[5] ==
'n') && (str[4] == 'n'))
    || (((ch == 'C') || (ch == 't')) && (str[0] == 'y') && (str[5] == 'y') && (str[4] == 'n'))
    || ((ch == 's') && (str[0] == 'y') && (str[5] == 'y') && (str[4] == 'n')))
{
    for(i = 0; i < 8; i++)
    {
        Printf(Monitor,"%n      ");
    }
    for(i = 8; i < 12; i++)
    {
        Printf(Monitor,"%n 0000 0000 ");
    }
}
else if(((ch == 'j') && (str[1] == 'y') && (str[0] == 'n') && (str[4] == 'y'))
    || (((ch == 'd') || (ch == 'k')) && (str[0] == 'y') && (str[4] == 'y'))
    || ((ch == 'h') && (str[0] == 'y') && (str[5] == 'y') && (str[4] == 'n'))
    || (((ch == 'c') || (ch == 't')) && (str[0] == 'y') && (str[4] == 'y'))
    || ((ch == 's') && (str[0] == 'y') && (str[4] == 'y')))
{
    for(i = 0; i < 8; i++)

```

```
{
    Printf(Monitor,"%n    ");
}
for(i = 8; i < 12; i++)
{
    Printf(Monitor,"%n 00000000 ");
}
}
#endif
/* 入力指示 */
DisplInput();
return;
}
```



```
/* EV_Input.h */
```

```
#ifndef Panel_h
```

```
#define Panel_h
```

```
#include "Panel.h"
```

```
#endif
```

```
#ifndef Timer_h
```

```
#define Timer_h
```

```
#include "Timer.h"
```

```
#endif
```

```
#ifndef EV_Time_h
```

```
#define EV_Time_h
```

```
#include "EV_Time.h"
```

```
#endif
```

```
#ifndef EV_File_h
```

```
#define EV_File_h
```

```
#include "EV_File.h"
```

```
#endif
```

```
/*=====
```

```
  入力を表す関数のプロトタイプ宣言
```

```
=====*/
```

```
char GetChar(char Ret);
```

```
/*=====
```

入力を表す構造体

```
=====*/  
struct EV_Input  
{  
    /* 安全 */  
    char Safety;  
    char *p_Safety;  
    char strCommand[4];  
    char *p_strCommand;  
    char strUserCommand[4];  
    char *p_strUserCommand;  
    char PermitCommand;  
    char *p_PermitCommand;  
    char PermitTurnOpen;  
    char *p_PermitTurnOpen;  
    char ch;  
    char *p_ch;  
    char str[9];  
    char *p_str;  
    char userstr[13];  
    char *p_userstr;  
#ifdef TWIN  
    char Twin1Safety;  
    char *p_Twin1Safety;  
    char strTwin1Command[4];  
    char *p_strTwin1Command;  
    char Twin1PermitCommand;  
    char *p_Twin1PermitCommand;  
    char Twin1PermitTurnOpen;
```

```

char *p_Twin1PermitTurnOpen;
char Twin1str[13];
char *p_Twin1str;
char Twin2Safety;
char *p_Twin2Safety;
char strTwin2Command[4];
char *p_strTwin2Command;
char Twin2PermitCommand;
char *p_Twin2PermitCommand;
char Twin2PermitTurnOpen;
char *p_Twin2PermitTurnOpen;
char Twin2str[13];
char *p_Twin2str;

#endif

/* ファイルストリーム */
struct EV_File SF;
struct EV_File CF;
struct EV_File PCF;
struct EV_File TOF;
struct EV_File PTOF;
struct EV_File LF;
struct EV_File MF;

};

/*=====
  入力を表すコンストラクタとメソッドのプロトタイプ宣言
=====*/

void EV_Input(struct EV_Input *This);
void OnInput(struct EV_Input *This, Thread *th);

```



```
/* EV_Input.c */
```

```
#include "C.h"
```

```
#include "EV_Input.h"
```

```
/*=====
```

```
    入力を表す関数
```

```
=====*/
```

```
char GetChar(char Ret)
```

```
{
```

```
#ifndef USE_BCC
```

```
    char sw[4];
```

```
    int i;
```

```
    int j;
```

```
    /* スイッチワーク初期化 */
```

```
    sw[0] = sw[1] = sw[2] = sw[3] = 0;
```

```
    /* ボタンが押された時にLCD,SCI,USBにメッセージを送る */
```

```
    for(j=0;j<4;j++)
```

```
    {
```

```
        i = GetSW(j);
```

```
        if( ((sw[j]^1)& i) ) /* sw = off->onで条件成立*/
```

```
        {
```

```
            switch(j){
```

```
            case 0:
```

```
                Ret = 'u'; /* sw0 Up 2階で開く */
```

```
                break;
```

```
            case 1:
```

```
                Ret = 'd'; /* sw1 Down 1階で開く */
```

```

        break;
    case 2:
        Ret = 'c'; /* sw2 Close 閉じる */
        break;
    case 3:
        Ret = 'q'; /* sw3 Quit 終了 */
        break;
    default:
        break;
    }
}

#else

#ifdef USE_LINUX
    if(kbhit())
    {
        Ret = (char)getchar();
    }
#else

#ifdef USE_MSVS2005
    if(_kbhit())
    {
        Ret = (char)_getche();
    }
#else
    if(kbhit())
    {
        Ret = (char)getche();
    }
}

```

```

#endif

#endif

#endif

    return Ret;
}

/*=====
  入力を表すコンストラクタとメソッド
=====*/

void EV_Input(struct EV_Input *This)
{
    /* 初期化 */
    This->p_strCommand = &This->strCommand[0];
    This->strCommand[3] = '\0';
    This->p_strUserCommand = &This->strUserCommand[0];
    This->strUserCommand[3] = '\0';
    This->p_ch = &This->ch;
    This->p_str = &This->str[0];
    This->str[8] = '\0';
    This->p_userstr = &This->userstr[0];
    This->userstr[12] = '\0';
    /* 安全初期化 */
    This->p_Safety = &This->Safety;
    /* 入力許可初期化 */
    This->p_PermitCommand = &This->PermitCommand;
    /* 反転開許可初期化 */
    This->p_PermitTurnOpen = &This->PermitTurnOpen;
#ifdef TWIN
    This->p_strTwin1Command = &This->strTwin1Command[0];

```

```

This->strTwin1Command[3] = '¥0';

This->p_Twin1str = &This->Twin1str[0];

This->Twin1str[12] = '¥0';

/* 安全初期化 */

This->p_Twin1Safety = &This->Twin1Safety;

/* 入力許可初期化 */

This->p_Twin1PermitCommand = &This->Twin1PermitCommand;

/* 反転開許可初期化 */

This->p_Twin1PermitTurnOpen = &This->Twin1PermitTurnOpen;

This->p_strTwin2Command = &This->strTwin2Command[0];

This->strTwin2Command[3] = '¥0';

This->p_Twin2str = &This->Twin2str[0];

This->Twin2str[12] = '¥0';

/* 安全初期化 */

This->p_Twin2Safety = &This->Twin2Safety;

/* 入力許可初期化 */

This->p_Twin2PermitCommand = &This->Twin2PermitCommand;

/* 反転開許可初期化 */

This->p_Twin2PermitTurnOpen = &This->Twin2PermitTurnOpen;

#endif

```

```

EV_File(&This->SF);

EV_File(&This->CF);

EV_File(&This->PCF);

EV_File(&This->TOF);

EV_File(&This->PTOF);

EV_File(&This->MF);

```

```

/* 安全入力 */

```



```

if(Read(&This->SF, "Safety.txt¥0", This->p_Safety) == NG) return;
/* 入力許可 */
if(PermitCommand_Read(&This->PCF, This->p_PermitCommand) == NG) return;
/* 反転開許可 */
if(PermitTurnOpen_Read(&This->PTOF, This->p_PermitTurnOpen) == NG) return;
/* モーター命令解読 */
if(Read(&This->MF, "Motor.txt¥0", This->p_ch) == NG) return;
/* リミットスイッチの前状態読み込み */
ReadString(&This->LF, "Limit.txt¥0", This->p_str, 9);
}

void OnInput(struct EV_Input *This, Thread *th)
{
#ifdef USE_BCC
    Thread *th46;
#endif

    if(ReadString(&This->CF, "Order.txt¥0", This->p_strCommand, 4) == NG) return;

    if((strcmp(This->p_strCommand, "N__¥0") == 0) || (strcmp(This->p_strCommand, "_NN¥0") ==
0)){
        This->strCommand[0] = GetChar('N');
        This->strCommand[1] = '_';
        This->strCommand[2] = '_';
        This->strCommand[3] = '¥0';
    }else{
        This->strCommand[3] = '¥0';
    }
}

```

```

/* 安全入力 */
if(Read(&This->SF, "Safety.txt¥0", This->p_Safety) == NG) return;

/* 入力許可 */
if(PermitCommand_Read(&This->PCF, This->p_PermitCommand) == NG) return;

/* 反転開許可 */
if(PermitTurnOpen_Read(&This->PTOF, This->p_PermitTurnOpen) == NG) return;

/* モーター命令解読 */
if(Read(&This->MF, "Motor.txt¥0", This->p_ch) == NG) return;

/* リミットスイッチの前状態読み込み */
ReadString(&This->LF, "Limit.txt¥0", This->p_str, 9);

switch(This->strCommand[0]){
    case 'N':
    case 'q':
        break;
    case '_':
        if(This->strCommand[1]== 'N'){
            break;
        }
    default:
#ifdef USE_BCC
        th46 = Thread_Start(46);
        th46->count = 0;
        nextRun(th46,0);
#endif
        break;
}

```

```

if((This->strCommand[0] == '_' ) && (This->strCommand[1] != '_')){
#ifdef USE_BCC
    if(ReadString(&This->CF, "UserCommand.txt¥0", This->p_strUserCommand, 4) == NG) return;
    if(Read(&This->PCF, "UserPermitCommand.txt¥0", This->p_PermitCommand) == NG) return;
    if(Read(&This->PTOF, "UserPermitTurnOpen.txt¥0", This->p_PermitTurnOpen) == NG) return;
    if(ReadString(&This->LF, "UserLimit.txt¥0", This->p_userstr, 13) == NG) return;
    if(This->PermitCommand == 'c'){
        if(This->strUserCommand[1] == 'e'){
            Write(&This->CF, "Earthquake.txt¥0", 'e');
            WriteString(&This->CF, "UserCommand.txt¥0", "01¥0");
        }else{
            WriteString(&This->CF, "UserCommand.txt¥0", &This->strUserCommand[1]);
        }
        // Write(&This->PCF, "UserPermitCommand.txt¥0", 'N');
    }
    if(This->PermitTurnOpen == 'o'){
        if((This->userstr[0] == 'y') && (This->userstr[7] != 'y')){
            switch(This->strUserCommand[1]){
            case '1':
            case 'e':
                Write(&This->TOF, "UserTurnOpen.txt¥0";o");
                break;
            default:
                break;
            }
            switch(This->strUserCommand[2]){
            case 'o':
            case '1':
            case 'L':

```

```

        Write(&This>TOF, "UserTurnOpen.txt¥0";o');
        break;
    default:
        break;
}
}else if((This->userstr[9] == 'u') && (This->userstr[10] == 'd') && (This->userstr[7] != 'y')){
    switch(This>strUserCommand[1]){
    case '2':
        Write(&This>TOF, "UserTurnOpen.txt¥0";o');
        break;
    default:
        break;
    }
    switch(This>strUserCommand[2]){
    case 'o':
    case '2':
    case 'L':
        Write(&This>TOF, "UserTurnOpen.txt¥0";o');
        break;
    default:
        break;
    }
}else if((This->userstr[3] == 'y') && (This->userstr[7] != 'y')){
    switch(This>strUserCommand[1]){
    case '3':
        Write(&This>TOF, "UserTurnOpen.txt¥0";o');
        break;
    default:
        break;

```

```
    }  
    switch(This->strUserCommand[2]){  
    case 'o':  
    case '3':  
    case 'L':  
        Write(&This->TOF, "UserTurnOpen.txt¥0", 'o');  
        break;  
    default:  
        break;  
    }  
}  
}
```

```
#endif
```

```
#ifdef TWIN
```

```
    if(ReadString(&This->CF, "Twin1Command.txt¥0", This->p_strTwin1Command, 4) == NG)
```

```
return;
```

```
    if(Read(&This->PCF, "Twin1PermitCommand.txt¥0", This->p_Twin1PermitCommand) == NG)
```

```
return;
```

```
    if(Read(&This->PTOF, "Twin1PermitTurnOpen.txt¥0", This->p_Twin1PermitTurnOpen) == NG)
```

```
return;
```

```
    if(ReadString(&This->LF, "Twin1Limit.txt¥0", This->p_Twin1str, 13) == NG) return;
```

```
    if(This->Twin1PermitCommand == 'c'){
```

```
        if(This->strTwin1Command[1] == 'e'){
```

```
            Write(&This->CF, "Earthquake.txt¥0", 'e');
```

```
            WriteString(&This->CF, "Twin1Command.txt¥0", "01¥0");
```

```
        }else{
```

```
            WriteString(&This->CF, "Twin1Command.txt¥0", &This->strTwin1Command[1]);
```

```
        }
```

```

    // Write(&This->PCF, "Twin1PermitCommand.txt¥0", 'N');
}
if(This->Twin1PermitTurnOpen == 'o'){
    if((This->Twin1str[0] == 'y') && (This->Twin1str[7] != 'y')){
        switch(This->strTwin1Command[1]){
            case '1':
            case 'e':
                Write(&This>TOF, "Twin1TurnOpen.txt¥0", 'o');
                break;
            default:
                break;
        }
        switch(This->strTwin1Command[2]){
            case 'o':
            case '1':
            case 'L':
                Write(&This>TOF, "Twin1TurnOpen.txt¥0", 'o');
                break;
            default:
                break;
        }
    }else if((This->Twin1str[9] == 'u') && (This->Twin1str[10] == 'd') && (This->Twin1str[7] !=
'y')){
        switch(This->strTwin1Command[1]){
            case '2':
                Write(&This>TOF, "Twin1TurnOpen.txt¥0", 'o');
                break;
            default:
                break;
        }
    }
}

```

```

    }
    switch(This->strTwin1Command[2]){
    case 'o':
    case '2':
    case 'L':
        Write(&This->TOF, "Twin1TurnOpen.txt¥0",o');
        break;
    default:
        break;
    }
}else if((This->Twin1str[3] == 'y') && (This->Twin1str[7] != 'y')){
    switch(This->strTwin1Command[1]){
    case '3':
        Write(&This->TOF, "Twin1TurnOpen.txt¥0",o');
        break;
    default:
        break;
    }
    switch(This->strTwin1Command[2]){
    case 'o':
    case '3':
    case 'L':
        Write(&This->TOF, "Twin1TurnOpen.txt¥0",o');
        break;
    default:
        break;
    }
}
}
}

```

```
if(ReadString(&This->CF, "Twin2Command.txt¥0", This->p_strTwin2Command, 4) == NG)
```

```
return;
```

```
if(Read(&This->PCF, "Twin2PermitCommand.txt¥0", This->p_Twin2PermitCommand) == NG)
```

```
return;
```

```
if(Read(&This->PTOF, "Twin2PermitTurnOpen.txt¥0", This->p_Twin2PermitTurnOpen) == NG)
```

```
return;
```

```
if(ReadString(&This->LF, "Twin2Limit.txt¥0", This->p_Twin2str, 13) == NG) return;
```

```
if(This->Twin2PermitCommand == 'c'){
```

```
    if(This->strTwin2Command[1] == 'e'){
```

```
        Write(&This->CF, "Earthquake.txt¥0", 'e');
```

```
        WriteString(&This->CF, "Twin2Command.txt¥0", "01¥0");
```

```
    }else{
```

```
        WriteString(&This->CF, "Twin2Command.txt¥0", &This->strTwin2Command[1]);
```

```
    }
```

```
    // Write(&This->PCF, "Twin2PermitCommand.txt¥0", 'N');
```

```
}
```

```
if(This->Twin2PermitTurnOpen == 'o'){
```

```
    if((This->Twin2str[0] == 'y') && (This->Twin2str[7] != 'y')){
```

```
        switch(This->strTwin2Command[1]){
```

```
            case '1':
```

```
            case 'e':
```

```
                Write(&This->TOF, "Twin2TurnOpen.txt¥0", 'o');
```

```
                break;
```

```
            default:
```

```
                break;
```

```
        }
```

```
        switch(This->strTwin2Command[2]){
```

```
            case 'o':
```

```
            case '1':
```



```

    case 'L':
        Write(&This>TOF, "Twin2TurnOpen.txt¥0",o');
        break;
    default:
        break;
}
}else if((This->Twin2str[9] == 'u') && (This->Twin2str[10] == 'd') && (This->Twin2str[7] !=
'y')){

    switch(This>strTwin2Command[1]){
    case '2':
        Write(&This>TOF, "Twin2TurnOpen.txt¥0",o');
        break;
    default:
        break;
    }
    switch(This>strTwin2Command[2]){
    case 'o':
    case '2':
    case 'L':
        Write(&This>TOF, "Twin2TurnOpen.txt¥0",o');
        break;
    default:
        break;
    }
}else if((This->Twin2str[3] == 'y') && (This->Twin2str[7] != 'y')){
    switch(This>strTwin2Command[1]){
    case '3':
        Write(&This>TOF, "Twin2TurnOpen.txt¥0",o');

```

```

        break;
    default:
        break;
}
switch(This->strTwin2Command[2]){
    case 'o':
    case '3':
    case 'L':
        Write(&This->TOF, "Twin2TurnOpen.txt¥0", 'o');
        break;
    default:
        break;
}
}
}
#endif
}else{
    switch(This->strCommand[0]){
    case 's':
        /* 命令入力 */
        Write(&This->SF, "Safety.txt¥0", 's');
        Motor_Write(&This->MF, 's');
        This->strCommand[0]= 'N';
        Command_Write(&This->CF, 'N');
        PermitCommand_Write(&This->PCF, 'N');
        break;
    case 'r':
        /* 命令入力 */
        Write(&This->SF, "Safety.txt¥0", 'h');

```

```

This->strCommand[0]= 'N';

Command_Write(&This->CF, 'N');

PermitCommand_Write(&This->PCF, 'c');

break;

case 'q':

    /* 命令入力 */

    Command_Write(&This->CF, 'q');

    Printf(ClsPnl,"QUIT");

    delete_(th);

    break;

default:

    if(This->PermitCommand== 'c'){

        switch(This->strCommand[0]){

            case 'o':

                if(This->str[4]!= 'y'){

                    PermitTurnOpen_Write(&This->PTOF, 'o');

                }

                if(This->Safety== 'h'){

                    /* 命令入力 */

                    This->Safety= 'Y';

                    Write(&This->SF, "Safety.txt¥0", 'Y');

                    if((This->ch == 's') && (This->Safety== 'Y') && (This->str[7]== 'n')){

                        Motor_Write(&This->MF, 'h');

                    }

                }

            }

            /* 命令入力 */

            Command_Write(&This->CF, 'o');

            PermitCommand_Write(&This->PCF, 'N');

            break;

```

```

case 'c':
    if(This->str[4] != 'y'){
        PermitTurnOpen_Write(&This>PTOF, 'o');
    }
    if(This->Safety == 'h'){
        /* 命令入力 */
        This->Safety = 'Y';
        Write(&This>SF, "Safety.txt¥0", 'Y');
        if((This->ch == 's') && (This->Safety == 'Y') && (This->str[4] == 'n')){
            Motor_Write(&This>MF, 't');
        }
    }
    /* 命令入力 */
    Command_Write(&This>CF, 'c');
    PermitCommand_Write(&This>PCF, 'N');
    break;
case 'u':
    if(This->str[4] != 'y'){
        PermitTurnOpen_Write(&This>PTOF, 'o');
    }
    if(This->Safety == 'h'){
        /* 命令入力 */
        This->Safety = 'Y';
        Write(&This>SF, "Safety.txt¥0", 'Y');
        if((This->ch == 's') && (This->Safety == 'Y') && (This->str[0] == 'y') && (This->str[4] == 'n')){
            Motor_Write(&This>MF, 't');
        }
        else if((This->ch == 's') && (This->Safety == 'Y') && (This->str[3] == 'n')){

```

```

        Motor_Write(&This>MF, 'j');
    }
    else if((This->ch == 's') && (This->Safety == 'Y') && (This->str[3] == 'y') &&
(This->str[7] == 'n')){
        Motor_Write(&This>MF, 'h');
    }
}
/* 命令入力 */
Command_Write(&This>CF, 'u');
PermitCommand_Write(&This>PCF, 'N');
break;
case 'd':
    if(This->str[4] != 'y'){
        PermitTurnOpen_Write(&This>PTOF, 'o');
    }
    if(This->Safety == 'h'){
        /* 命令入力 */
        This->Safety = 'Y';
        Write(&This>SF, "Safety.txt¥0", 'Y');
        if((This->ch == 's') && (This->Safety == 'Y') && (This->str[3] == 'y') && (This-
>str[4] == 'n')){
            Motor_Write(&This>MF, 't');
        }
        else if((This->ch == 's') && (This->Safety == 'Y') && (This->str[0] == 'n')){
            Motor_Write(&This>MF, 'k');
        }
        else if((This->ch == 's') && (This->Safety == 'Y') && (This->str[0] == 'y') &&
(This->str[7] == 'n')){

```

```

        Motor_Write(&This>MF, 'h');
    }
}
/* 命令入力 */
Command_Write(&This>CF, 'd');
PermitCommand_Write(&This>PCF, 'N');
break;
case 'y':
    if((This->str[0] != 'y') && (This->str[4] != 'y')){
        Printf(Panel, "¥nA Basket isn't 1st Floor");
    }
    else{
        if(This->Safety == 'h'){
            /* 命令入力 */
            This->Safety = 'Y';
            Write(&This>SF, "Safety.txt¥0", 'Y');
            if((This->ch == 's') && (This->Safety == 'Y') && (This->str[0] == 'n')){
                Motor_Write(&This>MF, 'k');
            }
            else if((This->ch == 's') && (This->Safety == 'Y') && (This->str[0] == 'y')
&& (This->str[7] == 'n')){
                Motor_Write(&This>MF, 'h');
            }
        }
        /* 命令入力 */
        Command_Write(&This>CF, 'y');
        PermitCommand_Write(&This>PCF, 'N');
    }
    break;

```

```

case 'Y':
    if((This->str[3] != 'y') && (This->str[4] != 'y')){
        Printf(Panel, "¥nA Basket isn't 2nd Floor");
    }
    else{
        if(This->Safety == 'h'){
            /* 命令入力 */
            This->Safety = 'Y';
            Write(&This>SF, "Safety.txt¥0", 'Y');
            if((This->ch == 's') && (This->Safety == 'Y') && (This->str[3] == 'n')){
                Motor_Write(&This>MF, 'j');
            }
            else if((This->ch == 's') && (This->Safety == 'Y') && (This->str[3] == 'y')
&& (This->str[7] == 'n')){
                Motor_Write(&This>MF, 'h');
            }
        }
        /* 命令入力 */
        Command_Write(&This>CF, 'Y');
        PermitCommand_Write(&This>PCF, 'N');
    }
    break;
case 'h':
    if(This->str[0] != 'y'){
        Printf(Panel, "¥nA Basket isn't 1st Floor");
    }
    else{
        if(This->str[4] != 'y'){
            PermitTurnOpen_Write(&This>PTOF, 'o');

```

```

    }

    if(This->Safety == 'h'){
        /* 命令入力 */
        This->Safety = 'Y';
        Write(&This>SF, "Safety.txt¥0", 'Y');
        if((This->ch == 's') && (This->Safety == 'Y') && (This->str[4] == 'n')){
            Motor_Write(&This>MF, 't');
        }
    }

    /* 命令入力 */
    Command_Write(&This>CF, 'h');
    PermitCommand_Write(&This>PCF, 'N');
}

break;

case 'H':
    if(This->str[3] != 'y'){
        Printf(Pannel, "¥nA Basket isn't 2nd Floor");
    }

    else{
        if(This->str[4] != 'y'){
            PermitTurnOpen_Write(&This>PTOF, 'o');
        }

        if(This->Safety == 'h'){
            /* 命令入力 */
            This->Safety = 'Y';
            Write(&This>SF, "Safety.txt¥0", 'Y');
            if((This->ch == 's') && (This->Safety == 'Y') && (This->str[4] == 'n')){
                Motor_Write(&This>MF, 't');
            }
        }
    }
}

```



```

    }

    /* 命令入力 */
    Command_Write(&This>CF, 'H');
    PermitCommand_Write(&This>PCF, 'N');
}

break;

default:
    break;
}
}

else if(This->PermitTurnOpen == 'o'){
    if((This->str[0] == 'y') && (This->str[7] != 'y')){
        switch(This->strCommand[0]){
            case 'o':
            case 'd':
            case 'y':
                if(This->Safety == 'h'){
                    /* 命令入力 */
                    This->Safety = 'Y';
                    Write(&This>SF, "Safety.txt¥0", 'Y');
                    if((This->ch == 's') && (This->Safety == 'Y') && (This->str[0] == 'y') &&
(This->str[7] == 'n')){
                        Motor_Write(&This>MF, 'h');
                    }
                }
            /* 命令入力 */
            Command_Write(&This>CF, This->strCommand[0]);
            PermitTurnOpen_Write(&This>PTOF, 'N');

```

```

        break;
    default:
        break;
    }
}
else if((This->str[3]== 'y') && (This->str[7]!= 'y')){
    switch(This->strCommand[0]){
    case 'o':
    case 'u':
    case 'Y':
        if(This->Safety== 'h'){
            /* 命令入力 */
            This->Safety= 'Y';
            Write(&This>SF, "Safety.txt¥0", 'Y');
            if((This->ch == 's') && (This->Safety== 'Y') && (This->str[3]== 'y') &&
(This->str[7] == 'n')){
                Motor_Write(&This>MF, 'h');
            }
        }
        /* 命令入力 */
        Command_Write(&This>CF, This->strCommand[0]);
        PermitTurnOpen_Write(&This>PTOF, 'N');
        break;
    default:
        break;
    }
}
}
break;

```

```
}
```

```
}
```

```
WriteString(&This->CF, "Order.txt¥0", "N__¥0");
```

```
return;
```

```
}
```

```
/* EV_Controller.h */
```

```
#ifndef Panel_h
```

```
#define Panel_h
```

```
#include "Panel.h"
```

```
#endif
```

```
#ifndef Timer_h
```

```
#define Timer_h
```

```
#include "Timer.h"
```

```
#endif
```

```
#ifndef EV_Time_h
```

```
#define EV_Time_h
```

```
#include "EV_Time.h"
```

```
#endif
```

```
#ifndef EV_File_h
```

```
#define EV_File_h
```

```
#include "EV_File.h"
```

```
#endif
```

```
#ifndef EV_UpDown_h
```

```
#define EV_UpDown_h
```

```
#include "EV_UpDown.h"
```

```
#endif
```

```
#ifndef EV_OpenClose_h
```

```
#define EV_OpenClose_h
```

```
#include "EV_OpenClose.h"
```

```
#endif
```

```
/*=====
```

```
制御盤を表す構造体宣言
```

```
=====*/
```

```
struct EV_Controller
```

```
{
```

```
/* エレベーターの現在位置オブジェクトの宣言 */
```

```
struct Position P;
```

```
struct Position *p_P;
```

```
/* 上昇オブジェクトの宣言 */
```

```
struct UpMotor UPMT;
```

```
/* 下降オブジェクトの宣言 */
```

```
struct DownMotor DNMT;
```

```
/* エレベーターの位置仮想ログ */
```

```
struct WaitPositionChangeLog WPCL;
```

```
/* エレベーターの現在位置オブジェクトの宣言 */
```

```
struct Door DR;
```

```
struct Door *p_DR;
```

```
/* 開オブジェクトの宣言 */
```

```
struct OpenMotor OPMT;
```

```
/* 閉オブジェクトの宣言 */
```

```
struct CloseMotor CLMT;
```

```
/* エレベーターの位置仮想ログ */
```

```
struct WaitDoorChangeLog WDCL;
```

```
/* 経過時間テンポラリ */
```

```
struct EV_Time T;
```

```
/* 安全 */
```

```
char Safety;
```

```
char *p_Safety;
```

```
/* Limit */
```

```
char str[9];
```

```
char *p_str;
```

```
/* 命令 */
```

```
char Command;
```

```
char *p_Command;
```

```
char PermitCommand;
```

```
char *p_PermitCommand;
```

```
char PermitTurnOpen;
```

```
char *p_PermitTurnOpen;
```

```
/* ファイルストリーム */
```

```
struct EV_File SF;
```

```
struct EV_File LF;
```

```
struct EV_File CF;
```

```
struct EV_File PCF;
```

```
struct EV_File PTOF;
```

```
struct EV_File MF;
```

```
};
```

```
/*=====
制御を表すコンストラクタとメソッドのプロトタイプ宣言
=====*/
void EV_Controller(struct EV_Controller *This, Thread *th);
void OnController(struct EV_Controller *This, Thread *th);
```

```
/* EV_Controller.c */
```

```
#include "C.h"
```

```
#include "EV_Controller.h"
```

```
/*=====
```

```
制御関数
```

```
=====*/
```

```
void EV_Controller(struct EV_Controller *This, Thread *th)
```

```
{
```

```
    /* 初期化 */
```

```
    This->p_P = &This->P;
```

```
    Position(This->p_P);
```

```
    UpMotor(&This->UPMT);
```

```
    DownMotor(&This->DNMT);
```

```
    WaitPositionChangeLog(&This->WPCL);
```

```
    This->p_DR = &This->DR;
```

```
    Door(This->p_DR);
```

```
    OpenMotor(&This->OPMT);
```

```
    CloseMotor(&This->CLMT);
```

```
    WaitDoorChangeLog(&This->WDCL);
```

```
    EV_Time(&This->T, th);
```

```
    EV_File(&This->SF);
```

```
    EV_File(&This->LF);
```

```
    EV_File(&This->CF);
```

```
    EV_File(&This->PCF);
```

```
    EV_File(&This->PTOF);
```

```
    EV_File(&This->MF);
```



```

/* 安全初期化 */
This->p_Safety = &This->Safety;

/* Limit初期化 */
This->p_str = &This->str[0];

/* 命令初期化 */
This->p_Command = &This->Command;
This->p_PermitCommand = &This->PermitCommand;
This->p_PermitTurnOpen = &This->PermitTurnOpen;

/* モーター停止命令 */
Motor_Write(&This->MF, 's');

/* エレベーターの位置仮想ログ */
/* 初期値 */
OnInitWaitPositionChangeLog(&This->WPCL, This->p_P);

/* エレベーターの位置仮想ログ */
/* 初期値 */
OnInitWaitDoorChangeLog(&This->WDCL, This->p_DR);

/* リミットスイッチの前状態読み込み */
ReadString(&This->LF, "Limit.txt¥0", This->p_str, 9);

/* 命令初期化 */
if(Command_Write(&This->CF, 'N') == NG) return;
if(PermitCommand_Write(&This->PCF, 'c') == NG) return;
if(PermitTurnOpen_Write(&This->PTOF, 'N') == NG) return;
}

```

```

/*
 * 主制御関数
 */
void OnController(struct EV_Controller *This, Thread *th)
{
    /* 安全入力 */
    if(Read(&This->SF, "Safety.txt¥0", This->p_Safety) == NG) return;
    /* リミットスイッチの前状態読み込み */
    ReadString(&This->LF, "Limit.txt¥0", This->p_str, 9);
    /* 命令入力 */
    if(Command_Read(&This->CF, This->p_Command) == NG) return;
    /* 入力許可 */
    if(PermitCommand_Read(&This->PCF, This->p_PermitCommand) == NG) return;
    /* 反転開許可 */
    if(PermitTurnOpen_Read(&This->PTOF, This->p_PermitTurnOpen) == NG) return;

    switch(This->Command){
        /* 終了命令ならば */
        case 'q':
            Motor_Write(&This->MF, 's');
            delete_(th);
            delete_(Thread_getThread(47));
#ifdef TWIN
            delete_(Thread_getThread(49));
#endif
            break;
        /* 非常停止命令ならば */
        case 's':
            SetPermit(&This->T, OFF);
    }
}

```

```

        break;

/* 復帰命令ならば*/
case 'r':

    break;

/* 上階呼命令ならば */
case 'Y':

    if((*This->p_P->p_UnderStop == ON) && (*This->p_DR->p_CloserStop == OFF)){

        break;

    }

/* 上昇命令ならば*/
case 'u':

    if(*This->p_P->p_UpperStop == ON){

        /* 開完了時*/

        if(*This->p_DR->p_OpennerStop == ON){

            PermitCommand_Write(&This>PCF, 'c');

            SetPermit(&This>T, ON);

            Command_Write(&This>CF, 'N');

            Clear();

            Printf(Panel,"Hello EV    ");

            break;

        }

        else{

            SetPermit(&This>T, OFF);

            SetCurrentTime(&This>T);

            /* 開 */

            Open(This->p_DR,This->OPMT,This->WDCL, This->p_Safety);

        }

    }

}

else if(*This->p_DR->p_CloserStop == ON){

```

```

    /* 閉完了時 */
    SetPermit(&This>T, OFF);
    SetCurrentTime(&This>T);
    PermitTurnOpen_Write(&This->PTOF, 'N');

    /* 上昇 */
    Up(This->p_P, This->UPMT, This->WPCL, This->p_Safety);
}
else if(*This->p_DR->p_CloserStop == OFF){
    SetPermit(&This>T, OFF);
    SetCurrentTime(&This>T);

    /* 閉 */
    Close(This->p_DR, This->CLMT, This->WDCL, This->p_Safety);
}
else{
    SetPermit(&This>T, OFF);
    SetCurrentTime(&This>T);
}
break;
/* 下階呼命令ならば */
case 'y':
    if((*This->p_P->p_UpperStop == ON) && (*This->p_DR->p_CloserStop == OFF)){
        break;
    }
/* 下降命令ならば */
case 'd':
    if(*This->p_P->p_UnderStop == ON){
        /* 開完了時 */
        if(*This->p_DR->p_OpennerStop == ON){

```

```

        PermitCommand_Write(&This->PCF, 'c');
        SetPermit(&This->T, ON);
        Command_Write(&This->CF, 'N');
        Clear();
        Printf(Pannel,"Hello EV    ");
        break;
    }
    else{
        SetPermit(&This->T, OFF);
        SetCurrentTime(&This->T);
        /* 開 */
        Open(This->p_DR,This->OPMT,This->WDCL, This->p_Safety);
    }
}
else if(*This->p_DR->p_CloserStop == ON){
    /* 閉完了時 */
    SetPermit(&This->T, OFF);
    SetCurrentTime(&This->T);
    PermitTurnOpen_Write(&This->PTOF, 'N');
    /* 下降 */
    Down(This->p_P,This->DNMT,This->WPCL, This->p_Safety);
}
else if(*This->p_DR->p_CloserStop == OFF){
    SetPermit(&This->T, OFF);
    SetCurrentTime(&This->T);
    /* 閉 */
    Close(This->p_DR,This->CLMT,This->WDCL, This->p_Safety);
}
else{

```

```

        SetPermit(&This>T, OFF);
        SetCurrentTime(&This>T);
    }
    break;
/* 開命令ならば*/
case 'o':
    /* 開完了時*/
    if(*This->p_DR->p_OpennerStop == ON){
        PermitCommand_Write(&This->PCF, 'c');
        SetPermit(&This>T, ON);
        Command_Write(&This->CF, 'N');
        Clear();
        Printf(Pannel,"Hello EV  ");
        break;
    }
    else{
        SetPermit(&This>T, OFF);
        SetCurrentTime(&This>T);
        /* 開 */
        Open(This->p_DR,This->OPMT,This->WDCL, This->p_Safety);
    }
    break;
/* 閉命令ならば*/
case 'c':
    SetPermit(&This->T, OFF);
    SetCurrentTime(&This->T);
    /* 閉完了時*/
    if(*This->p_DR->p_CloserStop == ON){
        PermitTurnOpen_Write(&This->PTOF, 'N');

```

```

    Command_Write(&This->CF, 'N');
    PermitCommand_Write(&This->PCF, 'c');
    Clear();
    Printf(Panel,"Hello EV    ");
    break;
}
else{
    /* 閉 */
    Close(This->p_DR,This->CLMT,This->WDCL, This->p_Safety);
}
break;
/* 閉命令ならば*/
case 'H':
    if(*This->p_P->p_UpperStop == ON){
        SetPermit(&This>T, OFF);
        SetCurrentTime(&This>T);
        /* 閉完了時 */
        if(*This->p_DR->p_CloserStop == ON){
            Command_Write(&This>CF, 'N');
            PermitTurnOpen_Write(&This->PTOF, 'N');
            PermitCommand_Write(&This>PCF, 'c');
            Clear();
            Printf(Panel,"Hello EV    ");
            break;
        }
    }
    else{
        /* 閉 */
        Close(This->p_DR,This->CLMT,This->WDCL, This->p_Safety);
    }
}

```

```

}

break;
/* 閉命令ならば*/
case 'h':
    if(*This->p_P->p_UnderStop == ON){
        SetPermit(&This>T, OFF);
        SetCurrentTime(&This>T);
        /* 閉完了時*/
        if(*This->p_DR->p_CloserStop == ON){
            Command_Write(&This>CF, 'N');
            PermitTurnOpen_Write(&This->PTOF, 'N');
            PermitCommand_Write(&This>PCF, 'c');
            Clear();
            Printf(Pannel,"Hello EV   ");
            break;
        }
        else{
            /* 閉*/
            Close(This->p_DR,This->CLMT,This->WDCL, This->p_Safety);
        }
    }
    break;
default:
    break;
}

if((GetCurrentTime(&This->T) >= OPENTIMEOUT) && (*This->p_DR->p_OpennerStop == ON) &&
(GetPermit(&This->T) == ON)){

```



```
SetPermit(&This->T, OFF);  
SetCurrentTime(&This->T);  
PermitTurnOpen_Write(&This->PTOF, 'o');  
PermitCommand_Write(&This->PCF, 'N');  
/* 閉 */  
Command_Write(&This->CF, 'c');  
}  
return;  
}
```

```
/* EV_Puls.h */
```

```
#ifndef Panel_h
```

```
#define Panel_h
```

```
#include "Panel.h"
```

```
#endif
```

```
#ifndef Timer_h
```

```
#define Timer_h
```

```
#include "Timer.h"
```

```
#endif
```

```
#ifndef EV_Time_h
```

```
#define EV_Time_h
```

```
#include "EV_Time.h"
```

```
#endif
```

```
#ifndef EV_File_h
```

```
#define EV_File_h
```

```
#include "EV_File.h"
```

```
#endif
```

```
/* Raspberry Pi 3 Model B I/O */
```

```
#ifdef USE_RASPBIAN
```

```
#define GPIO16 16
```

```
#define GPIO17 17
```

```
#define GPIO18 18
```

```
#define GPIO19 19
```

```
#define GPIO20 20
```

```
#endif
```

```
/*=====
```

```
送信を表す構造体
```

```
=====*/
```

```
struct EV_Puls
```

```
{
```

```
    char ch;
```

```
    char *p_ch;
```

```
    char Command;
```

```
    char *p_Command;
```

```
    /* ファイルストリーム */
```

```
    struct EV_File CF;
```

```
    struct EV_File MF;
```

```
};
```

```
/*=====
```

```
送信を表す関数のプロトタイプ宣言
```

```
=====*/
```

```
void EV_Set(int addressDataSet, int dataSet, int addressClockSet,
```

```
int clockSet);
```

```
void EV_EnableSet(void);
```

```
int EV_AddressSet(Thread *th, int base, int address);
```

```
int EV_DataSet(Thread *th, int base, int data);
```

```
void EV_AddressDataSet(struct EV_Puls *puls, Thread *th,
```

```
int address_1, int address_2, int address_3, int address_4,
```

```
int data_1, int data_2, int data_3, int data_4);
```

```
/*=====
送信を表すコンストラクタとメソッドのプロトタイプ宣言
=====*/
void EV_Puls(struct EV_Puls *This, Thread *th);
void OnPuls(struct EV_Puls *This, Thread *th);
```

```
/* EV_Puls.c */
```

```
#include "C.h"
```

```
#include "EV_Puls.h"
```

```
void EV_Set(int addressDataSet, int dataSet, int addressClockSet,  
int clockSet){
```

```
#ifndef USE_BCC
```

```
    /* address data set */
```

```
    if(addressDataSet == 0){
```

```
        PB.DR &= 0xfe;
```

```
    }
```

```
    else if(addressDataSet == 1){
```

```
        PB.DR |= 0x01;
```

```
    }
```

```
    /* data set */
```

```
    if(dataSet == 0){
```

```
        PB.DR &= 0xfd;
```

```
    }
```

```
    else if(dataSet == 1){
```

```
        PB.DR |= 0x02;
```

```
    }
```

```
    /* address clock set */
```

```
    if(addressClockSet == 0){
```

```
        PB.DR &= 0xfb;
```

```
    }
```

```
    else if(addressClockSet == 1){
```

```
        PB.DR |= 0x04;
```

```

}

/* clock set */
if(clockSet == 0){
    PB.DR &= 0xf7;
}
else if(clockSet == 1){
    PB.DR |= 0x08;
}
/* disable set */
PB.DR &= 0xef;

#endif

#ifdef USE_RASPBIAN
    /* address data set */
    digitalWrite(GPIO16, addressDataSet);
    /* data set */
    digitalWrite(GPIO17, dataSet);
    /* address clock set */
    digitalWrite(GPIO18, addressClockSet);
    /* clock set */
    digitalWrite(GPIO19, clockSet);
    /* disable set */
    digitalWrite(GPIO20, 0);
#endif

    return;
}

void EV_EnableSet(void){
#ifdef USE_BCC
    /* address data set */

```

```

PB.DR &= 0xfe;

/* data set */

PB.DR &= 0xfd;

/* address clock set */

PB.DR &= 0xfb;

/* clock set */

PB.DR &= 0xf7;

/* enable set */

PB.DR |= 0x10;

#endif

#ifdef USE_RASPBIAN

/* address data set */

digitalWrite(GPIO16, 0);

/* data set */

digitalWrite(GPIO17, 0);

/* address clock set */

digitalWrite(GPIO18, 0);

/* clock set */

digitalWrite(GPIO19, 0);

/* enable set */

digitalWrite(GPIO20, 1);

#endif

return;

}

int EV_AddressSet(Thread *th, int base, int address){

int Ret;

Ret = NG;

if(th->count == base){

```

```

    EV_Set(address,0, 0, 0);
    th->count++;
    Ret = OK;
}
else if(th->count == base + 1){
    EV_Set(address,0, 1, 0);
    th->count++;
    Ret = OK;
}
return Ret;
}

```

```

int EV_DataSet(Thread *th, int base, int data){
    int Ret;
    Ret = NG;
    if(th->count == base){
        EV_Set(0,data,0, 0);
        th->count++;
        Ret = OK;
    }
    else if(th->count == base + 1){
        EV_Set(0,data,0, 1);
        th->count++;
        Ret = OK;
    }
    return Ret;
}

```

```

void EV_AddressDataSet(struct EV_Puls *puls, Thread *th,

```



```

int address_1, int address_2, int address_3, int address_4,
int data_1, int data_2, int data_3, int data_4){
    if(EV_AddressSet(th, 0, address_1) == OK);
    else if(EV_AddressSet(th, 2, address_2) == OK);
    else if(EV_AddressSet(th, 4, address_3) == OK);
    else if(EV_AddressSet(th, 6, address_4) == OK);
    else if(EV_DataSet(th, 8, data_1) == OK);
    else if(EV_DataSet(th, 10, data_2) == OK);
    else if(EV_DataSet(th, 12, data_3) == OK);
    else if(EV_DataSet(th, 14, data_4) == OK);
    else{
        EV_EnableSet();
        th->count = 0;
        switch(puls->Command){
            case 'q':
#endif USE_BCC
                /* address data set */
                PB.DR &= 0xfe;
                /* data set */
                PB.DR &= 0xfd;
                /* address clock set */
                PB.DR &= 0xfb;
                /* clock set */
                PB.DR &= 0xf7;
                /* disable set */
                PB.DR &= 0xef;
#endif
#endif USE_RASPBIAN

```

```

        /* address data set */
        digitalWrite(GPIO160);
        /* data set */
        digitalWrite(GPIO170);
        /* address clock set */
        digitalWrite(GPIO180);
        /* clock set */
        digitalWrite(GPIO190);
        /* disable set */
        digitalWrite(GPIO200);
#endif

        delete_(th);
        break;
    default:
        break;
    }
}
return;
}

void EV_Puls(struct EV_Puls *This, Thread *th){
    This->p_ch = &This->ch;
    This->p_Command = &This->Command;
    EV_File(&This->MF);
    EV_File(&This->CF);
#ifdef USE_BCC
    PB.DDR = 0xff; /* bit7..0 out */
    PB.DR |= 0xff;
#endif
}

```

```

#ifdef USE_RASPBIAN
    if (wiringPiSetupGpio() == -1) exit(NG);
    pinMode(GPIO16, OUTPUT);
    pinMode(GPIO17, OUTPUT);
    pinMode(GPIO18, OUTPUT);
    pinMode(GPIO19, OUTPUT);
    pinMode(GPIO20, OUTPUT);
#endif

    return;
}

void OnPuls(struct EV_Puls *This, Thread *th){
    if(th->count == 0){
        /* 命令入力 */
        Command_Read(&This->CF, This->p_Command);
        /* モーター命令解読 */
        Read(&This->MF, "Motor.txt¥0", This->p_ch);
    }
    switch(This->ch){
    case 's':
        EV_AddressDataSet(This, th, 0, 1, 1, 1, 0, 0, 0, 0);
        break;
    case 'j':
        EV_AddressDataSet(This, th, 0, 1, 1, 1, 0, 0, 0, 1);
        break;
    case 'u':
        EV_AddressDataSet(This, th, 0, 1, 1, 1, 0, 0, 1, 0);
        break;
    case 'U':

```

```
    EV_AddressDataSet(This, th, 0, 1, 1, 1, 0, 0, 1, 1);  
    break;  
case 'k':  
    EV_AddressDataSet(This, th, 0, 1, 1, 1, 0, 1, 0, 0);  
    break;  
case 'd':  
    EV_AddressDataSet(This, th, 0, 1, 1, 1, 0, 1, 0, 1);  
    break;  
case 'D':  
    EV_AddressDataSet(This, th, 0, 1, 1, 1, 0, 1, 1, 0);  
    break;  
case 'h':  
    EV_AddressDataSet(This, th, 0, 1, 1, 1, 0, 1, 1, 1);  
    break;  
case 'o':  
    EV_AddressDataSet(This, th, 0, 1, 1, 1, 1, 0, 0, 0);  
    break;  
case 'O':  
    EV_AddressDataSet(This, th, 0, 1, 1, 1, 1, 0, 0, 1);  
    break;  
case 't':  
    EV_AddressDataSet(This, th, 0, 1, 1, 1, 1, 0, 1, 0);  
    break;  
case 'c':  
    EV_AddressDataSet(This, th, 0, 1, 1, 1, 1, 0, 1, 1);  
    break;  
case 'C':  
    EV_AddressDataSet(This, th, 0, 1, 1, 1, 1, 1, 0, 0);  
    break;
```

default:

break;

}

return;

}

```
/* EV_Simulator.h */
```

```
#ifndef Panel_h
```

```
#define Panel_h
```

```
#include "Panel.h"
```

```
#endif
```

```
#ifndef Timer_h
```

```
#define Timer_h
```

```
#include "Timer.h"
```

```
#endif
```

```
#ifndef EV_Time_h
```

```
#define EV_Time_h
```

```
#include "EV_Time.h"
```

```
#endif
```

```
#ifndef EV_File_h
```

```
#define EV_File_h
```

```
#include "EV_File.h"
```

```
#endif
```

```
#ifndef EV_Display_h
```

```
#define EV_Display_h
```

```
#include "EV_Display.h"
```

```
#endif
```

```
/*=====
```

シミュレータを表す構造体

```
=====*/  
struct EV_Simulator  
{  
    char ch;  
    char *p_ch;  
    char ch2;  
    char *p_ch2;  
    char ch3;  
    char *p_ch3;  
    char str[9];  
    char *p_str;  
  
    /* 時間管理 */  
    struct EV_Time T;  
  
    /* ファイルストリーム */  
    struct EV_File SF;  
    struct EV_File CF;  
    struct EV_File MF;  
    struct EV_File LF;  
};  
  
/*=====*/  
シミュレータを表すコンストラクタとメソッドのプロトタイプ宣言  
=====*/  
void EV_Simulator(struct EV_Simulator *This, Thread *th);  
void OnSimulator(struct EV_Simulator *This, Thread *th);
```

```
/* EV_Simulator.c */
```

```
#include "C.h"
```

```
#include "EV_Simulator.h"
```

```
/*=====
```

```
シミュレータ関数
```

```
=====*/
```

```
void EV_Simulator(struct EV_Simulator *This, Thread *th)
```

```
{
```

```
    /* 初期化 */
```

```
    EV_Time(&This->T, th);
```

```
    SetCurrentTime(&This->T);
```

```
    This->p_ch = &This->ch;
```

```
    This->p_ch2 = &This->ch2;
```

```
    This->p_ch3 = &This->ch3;
```

```
    This->p_str = &This->str[0];
```

```
    This->str[8] = '¥0';
```

```
    EV_File(&This->SF);
```

```
    EV_File(&This->CF);
```

```
    EV_File(&This->MF);
```

```
    EV_File(&This->LF);
```

```
    /* モーター命令解読 */
```

```
    if(Read(&This->MF, "Motor.txt¥0", This->p_ch) == NG) return;
```

```
    /* リミットスイッチの前状態読み込み */
```

```
    if(ReadString(&This->LF, "Limit.txt¥0", This->p_str, 9) == NG) return;
```

```
    /* 籠表示 */
```



```

Disp(This->ch, This->str);
}

void OnSimulator(struct EV_Simulator *This, Thread *th)
{
    /* 終了条件 */
    Read(&This->CF, "Command.txt¥0", This->p_ch3);
    if(This->ch3 == 'q'){
        Clear();
        delete_(th);
        return;
    }

    /* モーター命令解読 */
    Read(&This->MF, "Motor.txt¥0", This->p_ch);

    /* リミットスイッチの前状態読み込み */
    ReadString(&This->LF, "Limit.txt¥0", This->p_str, 9);

    /* 停止条件 */
    Read(&This->SF, "Safety.txt¥0", This->p_ch2);
    if(This->ch2 == 's'){
        Write(&This->CF, "Command.txt¥0", 'N');
        /* 籠表示 */
        Disp(This->ch, This->str);
        return;
    }

    /* リミットスイッチの新状態作成 */

```

```

if(This->ch == 'u'){
    if(This->str[0] == 'y');
    else if(This->str[1] == 'y') This->str[1] = 'n';
    else if(This->str[2] == 'n');
    else if(This->str[3] == 'n') This->str[3] = 'y';
}
else if(This->ch == 'U'){
    if(This->str[0] == 'y');
    else if(This->str[1] == 'y');
    else if(This->str[2] == 'n') This->str[2] = 'y';
    else if(This->str[3] == 'n');
}
else if(This->ch == 'j'){
    if(This->str[0] == 'y') This->str[0] = 'n';
    else if(This->str[1] == 'y') This->str[1] = 'n';
    else if(This->str[2] == 'n') This->str[2] = 'y';
    else if(This->str[3] == 'n') This->str[3] = 'y';
}
else if(This->ch == 'd'){
    if(This->str[3] == 'y');
    else if(This->str[2] == 'y') This->str[2] = 'n';
    else if(This->str[1] == 'n');
    else if(This->str[0] == 'n') This->str[0] = 'y';
}
else if(This->ch == 'D'){
    if(This->str[3] == 'y');
    else if(This->str[2] == 'y');
    else if(This->str[1] == 'n') This->str[1] = 'y';
    else if(This->str[0] == 'n');
}

```

```

}
else if(This->ch == 'k'){
    if(This->str[3] == 'y') This->str[3] = 'n';
    else if(This->str[2] == 'y') This->str[2] = 'n';
    else if(This->str[1] == 'n') This->str[1] = 'y';
    else if(This->str[0] == 'n') This->str[0] = 'y';
}
else if(This->ch == 'o'){
    if(This->str[4] == 'y');
    else if(This->str[5] == 'y') This->str[5] = 'n';
    else if(This->str[6] == 'n');
    else if(This->str[7] == 'n') This->str[7] = 'y';
}
else if(This->ch == 'O'){
    if(This->str[4] == 'y');
    else if(This->str[5] == 'y');
    else if(This->str[6] == 'n') This->str[6] = 'y';
    else if(This->str[7] == 'n');
}
else if(This->ch == 'h'){
    if(This->str[4] == 'y') This->str[4] = 'n';
    else if(This->str[5] == 'y') This->str[5] = 'n';
    else if(This->str[6] == 'n') This->str[6] = 'y';
    else if(This->str[7] == 'n') This->str[7] = 'y';
}
else if(This->ch == 'c'){
    if(This->str[7] == 'y');
    else if(This->str[6] == 'y') This->str[6] = 'n';
    else if(This->str[5] == 'n');
}

```

```

        else if(This->str[4] == 'n') This->str[4] = 'y';
    }
else if(This->ch == 'C'){
    if(This->str[7] == 'y');
    else if(This->str[6] == 'y');
    else if(This->str[5] == 'n') This->str[5] = 'y';
    else if(This->str[4] == 'n');
}
else if(This->ch == 't'){
    if(This->str[7] == 'y') This->str[7] = 'n';
    else if(This->str[6] == 'y') This->str[6] = 'n';
    else if(This->str[5] == 'n') This->str[5] = 'y';
    else if(This->str[4] == 'n') This->str[4] = 'y';
}

/* リミットスイッチの新状態書き込み */
WriteString(&This->LF, "Limit.txt¥0", This->str);

/* 籠表示 */
Disp(This->ch, This->str);

return;
}

```

```
/* EV_Log.h */
```

```
#ifndef Panel_h
```

```
#define Panel_h
```

```
#include "Panel.h"
```

```
#endif
```

```
#ifndef Timer_h
```

```
#define Timer_h
```

```
#include "Timer.h"
```

```
#endif
```

```
#ifndef EV_Time_h
```

```
#define EV_Time_h
```

```
#include "EV_Time.h"
```

```
#endif
```

```
#ifndef EV_File_h
```

```
#define EV_File_h
```

```
#include "EV_File.h"
```

```
#endif
```

```
/*=====
```

```
記録を表す構造体
```

```
=====*/
```

```
struct EV_Log{
```

```
    MYSQL *conn;
```

```
    MYSQL_RES *resp;
```

```

MYSQL_ROW row;
char str_sql1[255];
char str_sql2[255];
long mydate;
long mytime;
char ch_safety;
int i_safety;
char str_limit[9];
char ch_command;
char ch_permitcommand;
char ch_permitturnopen;
char ch_motor;
char *p_ch_safety;
char *p_str_limit;
char *p_ch_command;
char *p_ch_permitcommand;
char *p_ch_permitturnopen;
char *p_ch_motor;
struct EV_File SF;
struct EV_File LF;
struct EV_File CF;
struct EV_File PCF;
struct EV_File PTOF;
struct EV_File MF;
};

/*=====
記録を表すプロトタイプ宣言
=====*/

```

```
void EV_Log(struct EV_Log *This); /* database接続 */  
void delete_EV_Log(struct EV_Log *This); /* database切断 */  
void OnLog(struct EV_Log *This, Thread *th); /* database記録 */
```

```
/* EV_Log.c */

#include "C.h"

#include "EV_Log.h"

#define DBSERVER "localhost"

#define DBID "pi"

#define DBPASSWORD "raspberry"

#define DBNAME "ev001"

#define DBPORTNO 3306

#define SERVER_NAME "127.0.0.1"

#define ERR -1

#define BUFSIZE 1024

#ifdef USE_CENTOS

/* 送信ユーザー名(Linux Login ID) */

#define MAIL_FROM_HOST "work"

/* 送信サーバー名(自身 IP 127.0.0.1) */

#define MAIL_FROM_SERVER "localhost.localdomain"

/* 送信名 */

#define MAIL_FROM_HOST_AT_SERVER "work@localhost.localdomain"

/* 受信名(送信先 E-mail address) */

#define RCPT_TO_HOST_AT_SERVER "work@localhost.localdomain"

#endif

/* 送信メール内容ファイル */

#define SEND_DATA "./EV_Mail.txt"
```



```

char *command_name;

int socket(int domain, int type, int protocol);

/* 非常停止メッセージのメール通報のプロトタイプ */

void emergency(void);

/* database接続 */

void EV_Log(struct EV_Log *This)
{
    This->conn = NULL;

    This->resp = NULL;

    This->p_ch_safety = &This->ch_safety;

    This->p_str_limit = &This->str_limit[0];

    This->p_ch_command = &This->ch_command;

    This->p_ch_permitcommand = &This->ch_permitcommand;

    This->p_ch_permitturnopen = &This->ch_permitturnopen;

    This->p_ch_motor = &This->ch_motor;

    EV_File(&This->SF);

    EV_File(&This->LF);

    EV_File(&This->CF);

    EV_File(&This->PCF);

    EV_File(&This->PTOF);

    EV_File(&This->MF);

    memset(&This->str_sql2[0], 0x00, sizeof(This->str_sql2));

    /* mysql接続 */

    This->conn = mysql_init(NULL);

    if(!mysql_real_connect(This->conn, DBSERVER, DBID, DBPASSWORD, DBNAME, DBPORTNO, NULL,
0))
    {

```

```

    /* error */

    printf("%s¥n", mysql_error(This->conn));

    exit(NG);

}

return;

}

/* databaset切断 */

void delete_EV_Log(struct EV_Log *This)

{

    mysql_close(This->conn);

    return;

}

/* database記録 */

void OnLog(struct EV_Log *This, Thread *th)

{

    This->i_safety = 0;

    snprintf(This->str_sql1, sizeof(This->str_sql1), "select * from s_log order by lg_ymd desc, lg_hms
desc limit 1;" );

    mysql_query(This->conn , This->str_sql1);

    This->resp = mysql_use_result(This->conn);

    if((This->row = mysql_fetch_row(This->resp)) != NULL )

    {

        /* 直前のSafety(安全)のDatabase記録を採取 */

        This->i_safety = (int) This->row[2][0];

        /* Debug */

        /* printf("¥n%d %d : %s" , atoi(This->row[0]), atoi(This->row[1]), This->row[2]); */

```

```

}

mysql_free_result(This->resp);

myDateTime(&This->mydate, &This->mytime);
Read(&This->SF, "Safety.txt¥0", This->p_ch_safety);
ReadString(&This->LF, "Limit.txt¥0", This->p_str_limit, 9);
Read(&This->CF, "Command.txt¥0", This->p_ch_command);
Read(&This->PCF, "PermitCommand.txt¥0", This->p_ch_permitcommand);
Read(&This->PTOF, "PermitTurnOpen.txt¥0", This->p_ch_permitturnopen);
Read(&This->MF, "Motor.txt¥0", This->p_ch_motor);

sprintf(This->str_sql2, "insert into s_log(lg_ymd, lg_hms, lg_safety, lg_limit, lg_command,
lg_permitcommand, lg_permitturnopen, lg_motor) values ( %ld, %ld, '%c', '%s', '%c', '%c', '%c', '%c');",
This->mydate, This->mytime, This->ch_safety, This->p_str_limit, This->ch_command, This-
>ch_permitcommand, This->ch_permitturnopen, This->ch_motor);

/* Debug */
/* printf("¥n¥s¥n", This->str_sql2); */

mysql_query(This->conn, This->str_sql2);

/* 直前のSafety(安全)の記録がr(Run 運転)かどうか */
if(This->i_safety - ((int) 'r') == 0)
{
    /* 現在のSafety(安全)の記録がs(Stop 非常停止)かどうか */
    if(This->ch_safety == 's')
    {
        /* 非常停止メッセージのメール通報*/
        emergency();
    }
}

```

```

    }
}

if(This->ch_command == 'q'){
    delete_(th);
}
return;
}

#ifdef USE_CENTOS
/* 非常停止メッセージのメール通報 */
void emergency(void)
{
    /* ソケットディスクリプタ */
    int sds;
    FILE *fp;
    /*サーバプロセスのソケットアドレス情報 */
    struct sockaddr_in server;
    /* ホスト情報 */
    struct hostent *hp;
    char buf[BUFSIZE];
    int i_buf;
    char ch_buf;

    /* 画面クリア */
    CLEAR;

    /* ソケットの作成 */

```

```

if((sds = socket(PF_INET, SOCK_STREAM, 0)) == ERR)
{
    perror("client: socket");
    exit(NG);
}

/* サーバの情報を格納 */
memset((void*)&server, 0, sizeof(server));

server.sin_family = PF_INET;
server.sin_addr.s_addr = inet_addr(SERVER_NAME);

/* ポート番号25 */
server.sin_port = htons(25);

printf("%d¥r¥n", server.sin_family);

/* IPアドレスの設定 */
hp = gethostbyname(MAIL_FROM_SERVER);
memcpy(&(server.sin_addr), hp->h_addr_list[0], hp->h_length);

/* サーバに接続 */
if(connect(sds, (struct sockaddr *)&server, sizeof(server))==ERR)
{
    perror("client: connect");
    exit(NG);
}

read(sds, buf, sizeof(buf));

printf("%s¥r¥n", buf);

/* HELOの送信 */
strcpy(buf, "HELO ");
strcat(buf, MAIL_FROM_HOST);

```

```
strcat(buf, "¥r¥n¥0");  
printf("%s", buf);  
write(sds, buf, strlen(buf));  
memset(&buf[0], 0x00, sizeof(buf));  
read(sds, buf, sizeof(buf));  
printf("%s", buf);
```

```
/* MAIL FROMの送信 */
```

```
strcpy(buf, "MAIL FROM:");  
strcat(buf, MAIL_FROM_HOST);  
strcat(buf, "¥r¥n¥0");  
printf("%s", buf);  
write(sds, buf, strlen(buf));  
memset(&buf[0], 0x00, sizeof(buf));  
read(sds, buf, sizeof(buf));  
printf("%s", buf);
```

```
/* RCPT TOの送信 */
```

```
strcpy(buf, "RCPT TO:");  
strcat(buf, RCPT_TO_HOST_AT_SERVER);  
strcat(buf, "¥r¥n¥0");  
printf("%s", buf);  
write(sds, buf, strlen(buf));  
memset(&buf[0], 0x00, sizeof(buf));  
read(sds, buf, sizeof(buf));  
printf("%s", buf);
```

```
/* DATA の送信 */
```

```
strcpy(buf, "DATA¥r¥n¥0");
```

```

printf("%s", buf);
write(sds, buf, strlen(buf));
memset(&buf[0], 0x00, sizeof(buf));
read(sds, buf, sizeof(buf));
printf("%s", buf);
i_buf = 0;
if((fp = fopen(SEND_DATA, "r")) != NULL)
{
    strcpy(buf, "");
    while((ch_buf = fgetc(fp)) != EOF)
    {
        buf[i_buf++] = ch_buf;
    }
    fclose(fp);
    strcat(buf, "¥r¥n.¥r¥n¥0");
    /* 非常停止メッセージ通報*/
    printf("%s", buf);
    write(sds, buf, strlen(buf));
    memset(&buf[0], 0x00, sizeof(buf));
    read(sds, buf, sizeof(buf));
    printf("%s", buf);
}

strcpy(buf, "QUIT¥r¥n¥0");
printf("%s", buf);
write(sds, buf, strlen(buf));
memset(&buf[0], 0x00, sizeof(buf));
read(sds, buf, sizeof(buf));
printf("%s", buf);

```

```

    return;
}
#endif

#ifdef USE_RASPBIAN
/* 非常停止メッセージのメール通報 */
void emergency(void)
{
    /* プロセス分割用 */
    pid_t p;

    p = fork(); /* プロセス分割 */
    if(p == 0) /* 子プロセス */
    {
        /* 非常停止メッセージ通報 */
        system("echo An EV is stopped emergency! | mail -s ¥\"This is from Raspberry Pi.¥\"
info@hidemine.ciao.jp");
        printf("An EV is stopped emergency!");

        exit(OK); /* 子プロセス終了 */
    }

    return;
}
#endif

```



```

// EV_Queue.h

#ifndef Panel_h
#define Panel_h
#include "Panel.h"
#endif

#ifndef Timer_h
#define Timer_h
#include "Timer.h"
#endif

#ifndef EV_Time_h
#define EV_Time_h
#include "EV_Time.h"
#endif

#ifndef EV_File_h
#define EV_File_h
#include "EV_File.h"
#endif

struct EV_Queue{
    struct EV_Time T;
    struct EV_File F;
    char strQueue[33];
    char *p_strQueue;
    char strLimit[LIMIT];
    char *p_strLimit;
    char chMotor;
    char *p_chMotor;
    char *p_UserCommand;
    struct EV_Time *p_tOpenTO;
    char UD;
};

void EV_Q_Init(struct EV_Queue *This, Thread *th);
void EV_Q_Write(struct EV_Queue *This, char *filename, char *p_str, char mode);
int EV_Q_Command_Read(struct EV_Queue *This, char *p_Command, Thread *th);
int EV_Q_CheckTurnOpen(struct EV_Queue *This);
int EV_Q_Read(struct EV_Queue *This, char *p_check);
int EV_Q_Check(struct EV_Queue *This, char *c0, char *c1, char *c2, char *c3, char *c4, char *c5, char *c6,
char *c7, char *c8, char *c9, char *c10, char *c11);
int EV_Q_Flow(struct EV_Queue *This);
void EV_Q_Motor(struct EV_Queue *This, struct EV_Time *p_OpenTO);
void EV_Q_OTOUChange(struct EV_Queue *This, struct EV_Time *p_OpenTO, Thread *th);

```

```

// EV_Queue.c

#include "C.h"
#include "EV_Queue.h"

void EV_Q_Init(struct EV_Queue *This, Thread *th){
    int i;
    EV_Time(&This->T, th);
    EV_File(&This->F);
    This->p_strQueue = &This->strQueue[0];
    for(i = 0; i < 34; i++){
        This->strQueue[i]= 'N';
    }
    This->strQueue[34] = ' ';
    // uYdyocHhqe0o0c0102031U2D2U3D2y2h0L 
    // NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN 
    This->UD = 'U';
    return;
}

void EV_Q_Write(struct EV_Queue *This, char *filename, char *p_str, char mode)
{
    This->p_strLimit = &This->strLimit[0];
    ReadString(&This->F, "UserLimit.txt", This->p_strLimit, LIMIT);
    This->p_strQueue = &This->strQueue[0];
    if(filename[0] == 'C'){ // Command.txt
        switch(p_str[0]){
            case 'u':
                if(mode == 'w'){

```

```
        This->strQueue[0]= 'u';
    }
    else if(mode== 'r'){
        This->strQueue[0]= 'N';
    }
    break;
case 'Y':
    if(mode== 'w'){
        This->strQueue[1]= 'Y';
    }
    else if(mode== 'r'){
        This->strQueue[1]= 'N';
    }
    break;
case 'd':
    if(mode== 'w'){
        This->strQueue[2]= 'd';
    }
    else if(mode== 'r'){
        This->strQueue[2]= 'N';
    }
    break;
case 'y':
    if(mode== 'w'){
        This->strQueue[3]= 'y';
    }
    else if(mode== 'r'){
        This->strQueue[3]= 'N';
    }
}
```

```
        break;
case 'o':
    if(mode == 'w'){
        This->strQueue[4] = 'o';
    }
    else if(mode == 'r'){
        This->strQueue[4] = 'N';
    }
    break;
case 'c':
    if(mode == 'w'){
        This->strQueue[5] = 'c';
    }
    else if(mode == 'r'){
        This->strQueue[5] = 'N';
    }
    break;
case 'H':
    if(mode == 'w'){
        This->strQueue[6] = 'H';
    }
    else if(mode == 'r'){
        This->strQueue[6] = 'N';
    }
    break;
case 'h':
    if(mode == 'w'){
        This->strQueue[7] = 'h';
    }
}
```

```

else if(mode== 'r'){
    This->strQueue[7]= 'N';
}
break;
case 'q':
    if(mode== 'w'){
        This->strQueue[8]= 'q';
    }
    else if(mode== 'r'){
        This->strQueue[8]= 'N';
    }
    break;
case 'e':
    if(mode== 'w'){
        This->strQueue[9]= 'e';
    }
    else if(mode== 'r'){
        This->strQueue[9]= 'N';
    }
    break;
default:
    break;
}
}
else if(filename[0] == 'U'){ // UserCommand.txt
    switch(p_str[1]){
    case 'o':
        if(mode== 'w'){
            This->strQueue[10]= '0';

```

```

        This->strQueue[11]= 'o';
    }
    else if(mode== 'r'){
        This->strQueue[10]= 'N';
        This->strQueue[11]= 'N';
    }
    break;
case 'c':
    if(mode== 'w'){
        This->strQueue[12]= '0';
        This->strQueue[13]= 'c';
    }
    else if(mode== 'r'){
        This->strQueue[12]= 'N';
        This->strQueue[13]= 'N';
    }
    break;
case '1':
    if(mode== 'w'){
        if((This->strQueue[14] == '0') && (This->strQueue[15] == '1')){
            This->strQueue[14]= 'N';
            This->strQueue[15]= 'N';
            // printf("1階行きキャンセル\n");
        }
        else{
            This->strQueue[14]= '0';
            This->strQueue[15]= '1';
            // printf("1階行きON\n");
        }
    }

```

```

        }
    }
    else if(mode == 'r'){
        This->strQueue[14] = 'N';
        This->strQueue[15] = 'N';
    }
    break;
case '2':
    if(mode == 'w'){
        if((This->strQueue[16] == '0') && (This->strQueue[17] == '2')){
            This->strQueue[16] = 'N';
            This->strQueue[17] = 'N';
            // printf("2階行きキャンセル\n");
        }
        else{
            This->strQueue[16] = '0';
            This->strQueue[17] = '2';
            // printf("2階行きON\n");
        }
    }
    else if(mode == 'r'){
        This->strQueue[16] = 'N';
        This->strQueue[17] = 'N';
    }
    break;
case '3':
    if(mode == 'w'){
        if((This->strQueue[18] == '0') && (This->strQueue[19] == '3')){

```

```

        This->strQueue[18]= 'N';
        This->strQueue[19]= 'N';
        // printf("3階行きキャンセル¥n");
    }
    else{
        This->strQueue[18]= '0';
        This->strQueue[19]= '3';
        // printf("3階行きON¥n");
    }
}
else if(mode== 'r'){
    This->strQueue[18]= 'N';
    This->strQueue[19]= 'N';
}
break;
case 'U':
    if(mode== 'w'){
        if(p_str[0]== '1'){
            if((This->strQueue[20]== '1') && (This->strQueue[21]== 'U')){
                This->strQueue[20]= 'N';
                This->strQueue[21]= 'N';
                // printf("1階から上行きキャンセル¥n");
            }
            else{
                This->strQueue[20]= '1';
                This->strQueue[21]= 'U';
                // printf("1階から上行きON¥n");
            }
        }
    }
}

```



```

else if(p_str[0]== '2'){
    if((This->strQueue[24]== '2') && (This->strQueue[25]== 'U')){
        This->strQueue[24]= 'N';
        This->strQueue[25]= 'N';
        // printf("2階から上行きキャンセル¥n");
    }
    else{
        This->strQueue[24]= '2';
        This->strQueue[25]= 'U';
        // printf("2階から上行きON¥n");
    }
}
}

else if(mode== 'r'){
    if(p_str[0]== '1'){
        This->strQueue[20]= 'N';
        This->strQueue[21]= 'N';
    }
    else if(p_str[0]== '2'){
        This->strQueue[24]= 'N';
        This->strQueue[25]= 'N';
    }
}

break;

case 'D':
    if(mode== 'w'){
        if(p_str[0]== '2'){
            if((This->strQueue[22]== '2') && (This->strQueue[23]== 'D')){
                This->strQueue[22]= 'N';
            }
        }
    }
}

```

```

        This->strQueue[23]= 'N';
        // printf("2階から下行きキャンセル¥n");
    }
    else{
        This->strQueue[22]= '2';
        This->strQueue[23]= 'D';
        // printf("2階から下行きON¥n");
    }
}
else if(p_str[0]== '3'){
    if((This->strQueue[26]== '3') && (This->strQueue[27]== 'D')){
        This->strQueue[26]= 'N';
        This->strQueue[27]= 'N';
        // printf("3階から下行きキャンセル¥n");
    }
    else{
        This->strQueue[26]= '3';
        This->strQueue[27]= 'D';
        // printf("3階から下行きON¥n");
    }
    This->strQueue[26]= '3';
    This->strQueue[27]= 'D';
}
}
else if(mode== 'r'){
    if(p_str[0]== '2'){
        This->strQueue[22]= 'N';
        This->strQueue[23]= 'N';
    }
}

```

```

    }
    else if(p_str[0]== '3'){
        This->strQueue[26]= 'N';
        This->strQueue[27]= 'N';
    }
}
break;
case 'y':
    if(mode== 'w'){
        if(p_str[0]== '2'){
            This->strQueue[28]= '2';
            This->strQueue[29]= 'y';
        }
    }
    else if(mode== 'r'){
        if(p_str[0]== '2'){
            This->strQueue[28]= 'N';
            This->strQueue[29]= 'N';
        }
    }
    break;
case 'h':
    if(mode== 'w'){
        if(p_str[0]== '2'){
            This->strQueue[30]= '2';
            This->strQueue[31]= 'h';
        }
    }
    else if(mode== 'r'){

```

```

        if(p_str[0]== '2'){
            This->strQueue[30]= 'N';
            This->strQueue[31]= 'N';
        }
    }
    break;
case 'L':
    if(mode== 'w'){
        if((This->strQueue[32] == '0') && (This->strQueue[33] == 'L')){
            This->strQueue[32]= 'N';
            This->strQueue[33]= 'N';
            // printf("開延長キャンセル¥n");
        }
        else if((This->p_strLimit[6] == 'y') && (This->p_strLimit[7] == 'y'))
        {
            This->strQueue[32]= '0';
            This->strQueue[33]= 'L';
            // printf("開延長ON¥n");
        }
    }
    else if(mode== 'r'){
        This->strQueue[32]= 'N';
        This->strQueue[33]= 'N';
    }
    break;
default:
    break;
}
}

```

```
if(This->strQueue[0] == 'N' &&  
    This->strQueue[1] == 'N' &&  
    This->strQueue[2] == 'N' &&  
    This->strQueue[3] == 'N' &&  
    This->strQueue[4] == 'N' &&  
    This->strQueue[5] == 'N' &&  
    This->strQueue[6] == 'N' &&  
    This->strQueue[7] == 'N' &&  
    This->strQueue[8] == 'N' &&  
    This->strQueue[9] == 'N' &&  
    This->strQueue[10] == 'N' &&  
    This->strQueue[11] == 'N' &&  
    This->strQueue[12] == 'N' &&  
    This->strQueue[13] == 'N' &&  
    This->strQueue[14] == 'N' &&  
    This->strQueue[15] == 'N' &&  
    This->strQueue[16] == 'N' &&  
    This->strQueue[17] == 'N' &&  
    This->strQueue[18] == 'N' &&  
    This->strQueue[19] == 'N' &&  
    This->strQueue[20] == 'N' &&  
    This->strQueue[21] == 'N' &&  
    This->strQueue[22] == 'N' &&  
    This->strQueue[23] == 'N' &&  
    This->strQueue[24] == 'N' &&  
    This->strQueue[25] == 'N' &&  
    This->strQueue[26] == 'N' &&  
    This->strQueue[27] == 'N' &&  
    This->strQueue[28] == 'N' &&
```

```

    This->strQueue[29] == 'N' &&
    This->strQueue[30] == 'N' &&
    This->strQueue[31] == 'N' &&
    This->strQueue[32] == 'N' &&
    This->strQueue[33] == 'N')
{
    if((This->p_strLimit[6] == 'y') && (This->p_strLimit[7] == 'y'))
    {
        Write(&This->F, "UserMotor.txt", 's');
    }
    else if((This->strQueue[32] == '0') && (This->strQueue[33] == 'L')){
        This->strQueue[32]= 'N';
        This->strQueue[33]= 'N';
    }
}
return;
}

int EV_Q_Command_Read(struct EV_Queue *This, char *p_Command, Thread *th){
    char ch;
    char *p_ch;
    char str[3];
    char *p_str;

    This->p_strQueue = &This->strQueue[0];
    ch = 'N';
    p_ch = &ch;
    str[0] = 'N';
    str[1] = 'N';

```

```

str[2] = '\n';

p_str = &str[0];

ReadString(&This->F, "UserCommand.txt", p_str, 3);

// write queue
EV_Q_Write(This, "UserCommand.txt", p_str, 'w');

// remove UserCommand.txt
WriteString(&This->F, "UserCommand.txt", "NN¥0");

*p_Command = ch;
if((ch == 'u') || (ch == 'Y') || (ch == 'd') || (ch == 'y')
|| (ch == 'o') || (ch == 'c') || (ch == 'H') || (ch == 'h')
|| (ch == 'e')){
    // 命令入力許可終了
    //Write(&This->F, "UserPermitCommand.txt", 'N');

// 命令信号保持
    // WaitSecond(&This->T, 1);

    // 消命令
    // Write(&This->F, "Command.txt", 'N');

// remove queue
EV_Q_Init(This, th);

return OK;
}

```

```
else if(ch == 'q'){
    // 命令入力許可終了
    Write(&This->F, "UserPermitCommand.txt", 'N');
```

```
// 命令信号保持
```

```
    // WaitSecond(&This->T, 3);
```

```
    // 消命令
```

```
    // Write(&This->F, "Command.txt", 'N');
```

```
    // WriteString(&This->F, "UserCommand.txt", "NN¥0");
```

```
    // remove queue
```

```
    EV_Q_Init(This, th);
```

```
    return OK;
```

```
}
```

```
return ONE_MORE_TIME;
```

```
}
```

```
int EV_Q_CheckTurnOpen(struct EV_Queue *This){
```

```
    int Ret;
```

```
    char strTurnOpen[3];
```

```
    char *p_strTurnOpen;
```

```
    char PermitTurnOpen;
```

```
    strTurnOpen[0] = 'N';
```

```
    strTurnOpen[0] = 'N';
```

```
    strTurnOpen[0] = '¥0';
```

```
    PermitTurnOpen = 'N';
```

```
    p_strTurnOpen = &strTurnOpen[0];
```

```
    ReadString(&This->F, "UserTurnOpen.txt", p_strTurnOpen, 3);
```



```

// Wait_ms(&This->T, 200);

Read(&This->F, "UserPermitTurnOpen.txt", &PermitTurnOpen);

// Wait_ms(&This->T, 200);

if(PermitTurnOpen == 'N'){
    Ret = ONE_MORE_TIME;
}

else{
    // 閉中斷時
    if(p_strTurnOpen[0] == 'N'){
        Ret = ONE_MORE_TIME;
    }

    else if((p_strTurnOpen[0] == '0') && (p_strTurnOpen[1] == 'o')){
        Write(&This->F, "UserMotor.txt", 'h');
        EV_Q_Write(This, "UserCommand.txt", "0c¥0", 'r');
        This->p_UserCommand[0]= '0';
        This->p_UserCommand[1]= 'o';
        // printf("反転開します¥n");
        WriteString(&This->F, "UserCommand.txt", "0o¥0");
        Write(&This->F, "UserPermitCommand.txt", 'c');
        WriteString(&This->F, "UserTurnOpen.txt", "NN¥0");
        Write(&This->F, "UserPermitTurnOpen.txt", 'N');
        // WaitSecond(&This>T, 1);
        Ret = NG;
    }

    else if((p_strTurnOpen[0] == '0') && (p_strTurnOpen[1] == 'L')){
        Write(&This->F, "UserMotor.txt", 'h');
        EV_Q_Write(This, "UserCommand.txt", "0c¥0", 'r');
        This->p_UserCommand[0]= '0';
        This->p_UserCommand[1]= 'o';
    }
}

```

```

// printf("反転開します¥n");
WriteString(&This->F, "UserCommand.txt", "0L¥0");
Write(&This->F, "UserPermitCommand.txt", 'c');
WriteString(&This->F, "UserTurnOpen.txt", "NN¥0");
Write(&This->F, "UserPermitTurnOpen.txt", 'N');
// WaitSecond(&This>T, 1);
Ret = NG;
}
else if((This->p_strLimit[0] == 'y') && ((p_strTurnOpen[0] == '1') || (p_strTurnOpen[1] ==
'1'))){
Write(&This->F, "UserMotor.txt", 'h');
EV_Q_Write(This, "UserCommand.txt", "0c¥0", 'r');
This->p_UserCommand[0]= '0';
This->p_UserCommand[1]= 'o';
// printf("反転開します¥n");
WriteString(&This->F, "UserCommand.txt", "0o¥0");
Write(&This->F, "UserPermitCommand.txt", 'c');
WriteString(&This->F, "UserTurnOpen.txt", "NN¥0");
Write(&This->F, "UserPermitTurnOpen.txt", 'N');
// WaitSecond(&This>T, 1);
Ret = NG;
}
else if((This->p_strLimit[9] == 'u') && (This->p_strLimit[10] == 'd') && ((p_strTurnOpen[0] ==
'2') || (p_strTurnOpen[1] == '2'))){
Write(&This->F, "UserMotor.txt", 'h');
EV_Q_Write(This, "UserCommand.txt", "0c¥0", 'r');
EV_Q_Write(This, "UserCommand.txt", "2h¥0", 'r');
This->p_UserCommand[0]= '0';

```

```

This->p_UserCommand[1]='o';
// printf("反転開します¥n");
WriteString(&This->F, "UserCommand.txt", "0o¥0");
Write(&This->F, "UserPermitCommand.txt", 'c');
WriteString(&This->F, "UserTurnOpen.txt", "NN¥0");
Write(&This->F, "UserPermitTurnOpen.txt", 'N');
// WaitSecond(&This>T, 1);
Ret = NG;
}
else if((This->p_strLimit[3] == 'y') && ((p_strTurnOpen[0] == '3') || (p_strTurnOpen[1] ==
'3'))){
Write(&This->F, "UserMotor.txt", 'h');
EV_Q_Write(This, "UserCommand.txt", "0c¥0", 'r');
This->p_UserCommand[0]='0';
This->p_UserCommand[1]='o';
// printf("反転開します¥n");
WriteString(&This->F, "UserCommand.txt", "0o¥0");
Write(&This->F, "UserPermitCommand.txt", 'c');
WriteString(&This->F, "UserTurnOpen.txt", "NN¥0");
Write(&This->F, "UserPermitTurnOpen.txt", 'N');
// WaitSecond(&This>T, 1);
Ret = NG;
}
else{
Ret = ONE_MORE_TIME;
}
}
return Ret;
}

```

```

int EV_Q_Read(struct EV_Queue *This, char *p_check){
    int Ret;
    This->p_strQueue = &This->strQueue[0];
    Ret = EV_Q_CheckTurnOpen(This);
    if(Ret == NG) return NG;
    switch(p_check[1]){
    case 'o':
        if(This->strQueue[11] == 'o'){
            This->p_UserCommand[0] = '0';
            This->p_UserCommand[1] = 'o';
            Ret = OK;
        }
        else{
            This->p_UserCommand[0] = 'N';
            This->p_UserCommand[1] = 'N';
            Ret = ONE_MORE_TIME;
        }
        break;
    case 'c':
        if(This->strQueue[13] == 'c'){
            This->p_UserCommand[0] = '0';
            This->p_UserCommand[1] = 'c';
            Ret = OK;
        }
        else{
            This->p_UserCommand[0] = 'N';
            This->p_UserCommand[1] = 'N';
            Ret = ONE_MORE_TIME;
        }
    }
}

```

```
break;
case '1':
    if(This->strQueue[15] == '1'){
        This->p_UserCommand[0]= '0';
        This->p_UserCommand[1]= '1';
        Ret = OK;
    }
    else{
        This->p_UserCommand[0]= 'N';
        This->p_UserCommand[1]= 'N';
        Ret = ONE_MORE_TIME;
    }
    break;
case '2':
    if(This->strQueue[17] == '2'){
        This->p_UserCommand[0]= '0';
        This->p_UserCommand[1]= '2';
        Ret = OK;
    }
    else{
        This->p_UserCommand[0]= 'N';
        This->p_UserCommand[1]= 'N';
        Ret = ONE_MORE_TIME;
    }
    break;
case '3':
    if(This->strQueue[19] == '3'){
        This->p_UserCommand[0]= '0';
        This->p_UserCommand[1]= '3';
```

```

        Ret = OK;
    }
    else{
        This->p_UserCommand[0]= 'N';
        This->p_UserCommand[1]= 'N';
        Ret = ONE_MORE_TIME;
    }
    break;
case 'U':
    if(p_check[0]== '1'){
        if(This->strQueue[21]== 'U'){
            This->p_UserCommand[0]= '1';
            This->p_UserCommand[1]= 'U';
            Ret = OK;
        }
        else{
            This->p_UserCommand[0]= 'N';
            This->p_UserCommand[1]= 'N';
            Ret = ONE_MORE_TIME;
        }
    }
    if(p_check[0]== '2'){
        if(This->strQueue[25]== 'U'){
            This->p_UserCommand[0]= '2';
            This->p_UserCommand[1]= 'U';
            Ret = OK;
        }
        else{

```

```

        This->p_UserCommand[0]= 'N';
        This->p_UserCommand[1]= 'N';
        Ret = ONE_MORE_TIME;
    }
}
break;
case 'D':
    if(p_check[0]== '2'){
        if(This->strQueue[23]== 'D'){
            This->p_UserCommand[0]= '2';
            This->p_UserCommand[1]= 'D';
            Ret = OK;
        }
        else{
            This->p_UserCommand[0]= 'N';
            This->p_UserCommand[1]= 'N';
            Ret = ONE_MORE_TIME;
        }
    }
    if(p_check[0]== '3'){
        if(This->strQueue[27]== 'D'){
            This->p_UserCommand[0]= '3';
            This->p_UserCommand[1]= 'D';
            Ret = OK;
        }
        else{
            This->p_UserCommand[0]= 'N';
            This->p_UserCommand[1]= 'N';
            Ret = ONE_MORE_TIME;
        }
    }
}

```

```

        }
    }
    break;
case 'y':
    if(p_check[0]== '2'){
        if(This->strQueue[29]== 'y'){
            This->p_UserCommand[0]= '2';
            This->p_UserCommand[1]= 'y';
            Ret = OK;
        }
        else{
            This->p_UserCommand[0]= 'N';
            This->p_UserCommand[1]= 'N';
            Ret = ONE_MORE_TIME;
        }
    }
    break;
case 'h':
    if(p_check[0]== '2'){
        if(This->strQueue[31]== 'h'){
            This->p_UserCommand[0]= '2';
            This->p_UserCommand[1]= 'h';
            Ret = OK;
        }
        else{
            This->p_UserCommand[0]= 'N';
            This->p_UserCommand[1]= 'N';
            Ret = ONE_MORE_TIME;
        }
    }

```



```

    }
    break;
case 'L':
    if(This->strQueue[33] == 'L'){
        This->p_UserCommand[32] = '0';
        This->p_UserCommand[33]= 'L';
        Ret = OK;
    }
    else{
        This->p_UserCommand[32]= 'N';
        This->p_UserCommand[33]= 'N';
        Ret = ONE_MORE_TIME;
    }
    break;
default:
    This->p_UserCommand[0] = 'N';
    This->p_UserCommand[1] = 'N';
    Ret = ONE_MORE_TIME;
    break;
}
return Ret;
}

int EV_Q_Check(struct EV_Queue *This, char *c0, char *c1, char *c2, char *c3, char *c4, char *c5, char
*c6, char *c7, char *c8, char *c9, char *c10, char *c11){
    int Ret;
    Ret = EV_Q_Read(This, c0);
    if(Ret == OK) return OK;
    if(Ret == NG) return NG;
    Ret = EV_Q_Read(This, c1);

```

```
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Q_Read(This, c2);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Q_Read(This, c3);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Q_Read(This, c4);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Q_Read(This, c5);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Q_Read(This, c6);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Q_Read(This, c7);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Q_Read(This, c8);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Q_Read(This, c9);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Q_Read(This, c10);
if(Ret == OK) return OK;
```

```

if(Ret == NG) return NG;

Ret = EV_Q_Read(This, c11);

return Ret;

}

int EV_Q_Flow(struct EV_Queue *This){

    int Ret;

    This->p_strLimit = &This->strLimit[0];

    This->p_chMotor = &This->chMotor;

    ReadString(&This->F, "UserLimit.txt", This->p_strLimit, LIMIT);

    // Wait_ms(&This->T, 200);

    Read(&This->F, "UserMotor.txt", This->p_chMotor);

    // Wait_ms(&This->T, 200);

    if((This->chMotor == 'j') || (This->chMotor == 'u') || (This->chMotor == 'U')){

        //    remove 0o,0c

        EV_Q_Write(This, "UserCommand.txt", "0o¥0", 'r');

        EV_Q_Write(This, "UserCommand.txt", "0c¥0", 'r');

        if((This->strLimit[9] == 'u') && (This->strLimit[10] == 'd') && ((This->chMotor == 'j') || (This-
>chMotor == 'u'))){

            //    02,2y,2U,03,3D,2D,01,1U

            Ret = EV_Q_Check(This,

"02¥0","2y¥0","2U¥0","03¥0","3D¥0","2D¥0","01¥0","1U¥0","NN¥0","NN¥0","NN¥0","NN¥0");

        }

        else if(This->strLimit[9] == 'u'){

            //    03,3D,02,2D,2y,01,1U,2U

            Ret = EV_Q_Check(This,

"03¥0","3D¥0","02¥0","2D¥0","2y¥0","01¥0","1U¥0","2U¥0","NN¥0","NN¥0","NN¥0","NN¥0");

        }

        else{

            //    02,2U,2y,03,3D,2D,01,1U

```

```

        Ret = EV_Q_Check(This,
"02¥0","2U¥0","2y¥0","03¥0","3D¥0","2D¥0","01¥0","1U¥0","NN¥0","NN¥0","NN¥0","NN¥0");
    }
}
else if((This->chMotor == 'k') || (This->chMotor == 'd') || (This->chMotor == 'D')){
    // remove 0o,0c
    EV_Q_Write(This, "UserCommand.txt", "0o¥0", 'r');
    EV_Q_Write(This, "UserCommand.txt", "0c¥0", 'r');
    if((This->strLimit[9] == 'u') && (This->strLimit[10] == 'd') && ((This->chMotor == 'k') || (This-
>chMotor == 'd'))){
        // 02,2y,2D,01,1U,2U,03,3D
        Ret = EV_Q_Check(This,
"02¥0","2y¥0","2D¥0","01¥0","1U¥0","2U¥0","03¥0","3D¥0","NN¥0","NN¥0","NN¥0","NN¥0");
    }
    else if(This->strLimit[10] == 'd'){
        // 01,1U,02,2U,2y,03,3D,2D
        Ret = EV_Q_Check(This,
"01¥0","1U¥0","02¥0","2U¥0","2y¥0","03¥0","3D¥0","2D¥0","NN¥0","NN¥0","NN¥0","NN¥0");
    }
    else{
        // 02,2D,2y,01,1U,2U,03,3D
        Ret = EV_Q_Check(This,
"02¥0","2D¥0","2y¥0","01¥0","1U¥0","2U¥0","03¥0","3D¥0","NN¥0","NN¥0","NN¥0","NN¥0");
    }
}
else if((This->chMotor == 'h') || (This->chMotor == 'o') || (This->chMotor == 'O')){
    if(This->strLimit[0] == 'y'){
        // 0L,0o,01,1U,0c,02,2U,03,3D,2D
        Ret = EV_Q_Check(This,

```

```

"0L¥0","0o¥0","01¥0","1U¥0","0c¥0","02¥0","2U¥0","03¥0","3D¥0","2D¥0","NN¥0","NN¥0");
    }
    else if((This->strLimit[9] == 'u') && (This->strLimit[10] == 'd')){
        //    0L,0o,02,2D/2U,2y,0c,01/03,1U/3D
        Ret = EV_Q_Check(This,
"0L¥0","0o¥0","02¥0","2D¥0","2U¥0","2y¥0","0c¥0","01¥0","03¥0","1U¥0","3D¥0","NN¥0");
    }
    else if(This->strLimit[3] == 'y'){
        //    0L,0o,03,3D,0c,02,2D,01,1U,2U
        Ret = EV_Q_Check(This,
"0L¥0","0o¥0","03¥0","3D¥0","0c¥0","02¥0","2D¥0","01¥0","1U¥0","2U¥0","NN¥0","NN¥0");
    }
}
else if((This->chMotor == 't') || (This->chMotor == 'c') || (This->chMotor == 'C')){
    if(This->strLimit[0] == 'y'){
        //    0L,0o,01,1U,0c,02,2U,03,3D,2D
        Ret = EV_Q_Check(This,
"0L¥0","0o¥0","01¥0","1U¥0","0c¥0","02¥0","2U¥0","03¥0","3D¥0","2D¥0","NN¥0","NN¥0");
    }
    else if((This->strLimit[9] == 'u') && (This->strLimit[10] == 'd')){
        //    0L,0o,02,2D/2U,2y,0c,2h,01/03,1U/3D
        //    Ret = EV_Q_Check(This,
"0L¥0","0o¥0","02¥0","2D¥0","2U¥0","2y¥0","0c¥0","2h¥0","01¥0","03¥0","1U¥0","3D¥0");
        if(This->UD == 'U')
        {
            //    0L,0o,02,2U,2y,0c,2h,03,3D,2D,01,1U
            Ret = EV_Q_Check(This,
"0L¥0","0o¥0","02¥0","2U¥0","2y¥0","0c¥0","2h¥0","03¥0","3D¥0","2D¥0","01¥0","1U¥0");
        }
    }
}

```

```

else if(This->UD == 'D')
{
    // 0L,0o,02,2D,2y,0c,2h,01,1U,2U,03,3D
    Ret = EV_Q_Check(This,
"0L¥0","0o¥0","02¥0","2D¥0","2y¥0","0c¥0","2h¥0","01¥0","1U¥0","2U¥0","03¥0","3D¥0");
    }
}

else if(This->strLimit[3] == 'y'){
    // 0L,0o,03,3D,0c,02,2D,01,1U,2U
    Ret = EV_Q_Check(This,
"0L¥0","0o¥0","03¥0","3D¥0","0c¥0","02¥0","2D¥0","01¥0","1U¥0","2U¥0","NN¥0","NN¥0");
    }
}

else if(This->chMotor == 's'){
    if(This->strLimit[0] == 'y'){
        if(This->strLimit[7]== 'y'){ // 1開
            // remove 0o,01,1U
            EV_Q_Write(This, "UserCommand.txt", "0o¥0", 'r');
            EV_Q_Write(This, "UserCommand.txt", "01¥0", 'r');
            EV_Q_Write(This, "UserCommand.txt", "1U¥0", 'r');
            // 0L,0c,02,2U,03,3D,2D
            // Ret = EV_Q_Check(This,
"0L¥0","0c¥0","02¥0","2U¥0","03¥0","3D¥0","2D¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0");
            // 0L,0c
            Ret = EV_Q_Check(This,
"0L¥0","0c¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0");
        }
    }
    else if(This->strLimit[4]== 'y'){ // 1閉

```

```

// remove 0c
EV_Q_Write(This, "UserCommand.txt", "0c%0", 'r');
// 0L,0o,01,1U,02,2U,2y,03,3D,2D
Ret = EV_Q_Check(This,
"0L%0", "0o%0", "01%0", "1U%0", "02%0", "2U%0", "2y%0", "03%0", "3D%0", "2D%0", "NN%0", "NN%0");
}
else{
// 0L,0o,01,1U,0c,02,2U,03,3D,2D
Ret = EV_Q_Check(This,
"0L%0", "0o%0", "01%0", "1U%0", "0c%0", "02%0", "2U%0", "03%0", "3D%0", "2D%0", "NN%0", "NN%0");
}
}
else if((This->strLimit[9] == 'u') && (This->strLimit[10] == 'd'))
{
if(This->strLimit[7]== 'y'){ // 2開
// remove 0o,02,2U,2D
EV_Q_Write(This, "UserCommand.txt", "0o%0", 'r');
EV_Q_Write(This, "UserCommand.txt", "02%0", 'r');
if(This->UD == 'U')
{
EV_Q_Write(This, "UserCommand.txt", "2U%0", 'r');
// 0L,0o,02,2U,2y,0c,2h,03,3D
Ret = EV_Q_Check(This,
"0L%0", "0o%0", "02%0", "2U%0", "2y%0", "0c%0", "2h%0", "03%0", "3D%0", "NN%0", "NN%0", "NN%0");
if(Ret == ONE_MORE_TIME)
{
EV_Q_Write(This, "UserCommand.txt", "2D%0", 'r');
}
}
}
}
}

```

```

else if(This->UD == 'D')
{
    EV_Q_Write(This,"UserCommand.txt";2D¥0", 'r');
    // 0L,0o,02,2D,2y,0c,2h,01,1U
    Ret = EV_Q_Check(This,
"0L¥0", "0o¥0", "02¥0", "2D¥0", "2y¥0", "0c¥0", "2h¥0", "01¥0", "1U¥0", "NN¥0", "NN¥0", "NN¥0");
    if(Ret== ONE_MORE_TIME)
    {
        EV_Q_Write(This,"UserCommand.txt";2U¥0", 'r');
    }
}
EV_Q_Write(This, "UserCommand.txt", "2y¥0", 'r');
// 0L,0c,2h,01/03,1U/3D
// Ret = EV_Q_Check(This,
"0L¥0", "0c¥0", "2h¥0", "01¥0", "03¥0", "1U¥0", "3D¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0");
// 0L,0c
Ret = EV_Q_Check(This,
"0L¥0", "0c¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0");
}
else if(This->strLimit[4]== 'y'){ // 2閉
    // remove 0c,2h
    EV_Q_Write(This, "UserCommand.txt", "0c¥0", 'r');
    EV_Q_Write(This, "UserCommand.txt", "2h¥0", 'r');
    // 0L,0o,02,2D/2U,2y,01/03,1U/3D
    // Ret = EV_Q_Check(This,
"0L¥0", "0o¥0", "02¥0", "2D¥0", "2U¥0", "2y¥0", "01¥0", "03¥0", "1U¥0", "3D¥0", "NN¥0", "NN¥0");
    if(This->UD == 'U')
    {
        // 0L,0o,02,2U,2y,03,3D,2D,01,1U

```



```

        Ret = EV_Q_Check(This,
"0L¥0","0o¥0","02¥0","2U¥0","2y¥0","03¥0","3D¥0","2D¥0","01¥0","1U¥0","NN¥0","NN¥0");
    }
    else if(This->UD == 'D')
    {
        // 0L,0o,02,2D,2y,01,1U,2U,03,3D
        Ret = EV_Q_Check(This,
"0L¥0","0o¥0","02¥0","2D¥0","2y¥0","01¥0","1U¥0","2U¥0","03¥0","3D¥0","NN¥0","NN¥0");
    }
}
else{
    // 0L,0o,02,2D/2U,2y,0c,2h,01/03,1U/3D
    Ret = EV_Q_Check(This,
"0L¥0","0o¥0","02¥0","2D¥0","2U¥0","2y¥0","0c¥0","2h¥0","01¥0","03¥0","1U¥0","3D¥0");
}
}
else if(This->strLimit[3] == 'y'){
    if(This->strLimit[7] == 'y'){ // 3開
        // remove 0o,03,3D
        EV_Q_Write(This, "UserCommand.txt", "0o¥0", 'r');
        EV_Q_Write(This, "UserCommand.txt", "03¥0", 'r');
        EV_Q_Write(This, "UserCommand.txt", "3D¥0", 'r');
        // 0L,0c,02,2D,01,1U,2U
        // Ret = EV_Q_Check(This,
"0L¥0","0c¥0","02¥0","2D¥0","01¥0","1U¥0","2U¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0");
        // 0L,0c
        Ret = EV_Q_Check(This,
"0L¥0","0c¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0");
    }
}

```

```

else if(This->strLimit[4]== 'y'){ // 3閉
    // remove 0c
    EV_Q_Write(This, "UserCommand.txt", "0c¥0", 'r');
    // 0L,0o,03,3D,02,2D,2y,01,1U,2U
    Ret = EV_Q_Check(This,
"0L¥0", "0o¥0", "03¥0", "3D¥0", "02¥0", "2D¥0", "2y¥0", "01¥0", "1U¥0", "2U¥0", "NN¥0", "NN¥0");
    }
else{
    // 0L,0o,03,3D,0c,02,2D,01,1U,2U
    Ret = EV_Q_Check(This,
"0L¥0", "0o¥0", "03¥0", "3D¥0", "0c¥0", "02¥0", "2D¥0", "01¥0", "1U¥0", "2U¥0", "NN¥0", "NN¥0");
    }
}

else if(This->strLimit[9] == 'd'){
    // remove 0o,0c
    EV_Q_Write(This, "UserCommand.txt", "0o¥0", 'r');
    EV_Q_Write(This, "UserCommand.txt", "0c¥0", 'r');
    // 0L,01/02,1U/2U,2y,03,3D,2D
    Ret = EV_Q_Check(This,
"0L¥0", "01¥0", "02¥0", "1U¥0", "2U¥0", "2y¥0", "03¥0", "3D¥0", "2D¥0", "NN¥0", "NN¥0", "NN¥0");
    }

else if(This->strLimit[10] == 'u'){
    // remove 0o,0c
    EV_Q_Write(This, "UserCommand.txt", "0o¥0", 'r');
    EV_Q_Write(This, "UserCommand.txt", "0c¥0", 'r');
    // 0L,03/02,3D/2D,2y,01,1U,2U
    Ret = EV_Q_Check(This,
"0L¥0", "03¥0", "02¥0", "3D¥0", "2D¥0", "2y¥0", "01¥0", "1U¥0", "2U¥0", "NN¥0", "NN¥0", "NN¥0");

```

```

    }
}
return Ret;
}

void EV_Q_Motor(struct EV_Queue *This, struct EV_Time *p_OpenTO){
    int EV_Q_FlowRet;
    char UserCommand[3];
    UserCommand[0] = 'N';
    UserCommand[1] = 'N';
    UserCommand[2] = '¥0';
    This->p_UserCommand = &UserCommand[0];
    This->p_strLimit = &This->strLimit[0];
    This->p_chMotor = &This->chMotor;
    ReadString(&This->F, "UserLimit.txt", This->p_strLimit, LIMIT);
    Read(&This->F, "UserMotor.txt", This->p_chMotor);
    This->p_tOpenTO = p_OpenTO;
    EV_Q_FlowRet = EV_Q_Flow(This);
    if(EV_Q_FlowRet == OK){
        if(This->strLimit[0] == 'y'){
            if(This->strLimit[4] == 'y'){ // 1閉
                This->UD = 'U';
                switch(UserCommand[1]){
                    case 'U':
                        if(UserCommand[0] == '1'){
                            Write(&This->F, "UserMotor.txt";s');
                            // printf("STOPします¥n");
                            // WaitSecond(&This->T, 1);
                            Write(&This->F, "UserMotor.txt";h');
                            // printf("開きます¥n");

```

```

        // WaitSecond(&This>T, 1);
    }
    else if(UserCommand[0]== '2'){
        Write(&This>F, "UserMotor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "UserPermitTurnOpen.txt",N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "UserMotor.txt",'j');
        // printf("上昇します¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'D':
    if(UserCommand[0]== '2'){
        Write(&This>F, "UserMotor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "UserPermitTurnOpen.txt",N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "UserMotor.txt",'j');
        // printf("上昇します¥n");
        // WaitSecond(&This>T, 1);
    }
    else if(UserCommand[0]== '3'){
        Write(&This>F, "UserMotor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "UserPermitTurnOpen.txt",N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "UserMotor.txt",'j');
        // printf("上昇します¥n");

```

```

        // WaitSecond(&This>T, 1);
    }
    break;
case 'y':
    if(UserCommand[0]== '2'){
        Write(&This>F, "UserMotor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "UserPermitTurnOpen.txt",N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "UserMotor.txt",'j');
        // printf("上昇します¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'o':
case 'L':
case '1':
    Write(&This>F, "UserMotor.txt",'s');
    // printf("STOPします¥n");
    // WaitSecond(&This>T, 1);
    Write(&This>F, "UserMotor.txt",'h');
    // printf("開きます¥n");
    // WaitSecond(&This>T, 1);
    break;
case '2':
case '3':
    Write(&This>F, "UserMotor.txt",'s');
    // printf("STOPします¥n");
    Write(&This>F, "UserPermitTurnOpen.txt",N');

```

```

    // WaitSecond(&This>T, 1);
    Write(&This>F, "UserMotor.txt", 'j');
    // printf("上昇します¥n");
    // WaitSecond(&This>T, 1);
    break;
case 'c':
    Write(&This>F, "UserMotor.txt", 's');
    // printf("STOPします¥n");
    Write(&This>F, "UserPermitTurnOpen.txt", 'N');
    // WaitSecond(&This>T, 1);
    break;
default:
    break;
}
}
else if(This->strLimit[7]== 'y'){ // 1開
    switch(UserCommand[1]){
    case 'U':
        if(UserCommand[0]== '1'){
            Write(&This>F, "UserMotor.txt", 's');
            // printf("STOPします¥n");
            // 現在開
            // printf("開いています¥n");
            SetCurrentTime(p_OpenTO);
            SetPermit(p_OpenTOON);
            // WaitSecond(&This>T, 1);
        }
        else if(UserCommand[0]== '2'){

```

```

        Write(&This>F, "UserMotor.txt",'t');
        // printf("閉じます¥n");
        Write(&This>F, "UserPermitTurnOpen.txt"ρ');
        SetPermit(p_OpenTOOFF);
        // WaitSecond(&This>T, 1);
    }
    break;
case 'D':
    if(UserCommand[0]== '2'){
        Write(&This>F, "UserMotor.txt",'t');
        // printf("閉じます¥n");
        Write(&This>F, "UserPermitTurnOpen.txt"ρ');
        SetPermit(p_OpenTOOFF);
        // WaitSecond(&This>T, 1);
    }
    else if(UserCommand[0]== '3'){
        Write(&This>F, "UserMotor.txt",'t');
        // printf("閉じます¥n");
        Write(&This>F, "UserPermitTurnOpen.txt"ρ');
        SetPermit(p_OpenTOOFF);
        // WaitSecond(&This>T, 1);
    }
    break;
case 'o':
case 'L':
case '1':
    Write(&This>F, "UserMotor.txt",'s');
    // printf("STOPします¥n");
    // 現在開

```

```

        // printf("開いています¥n");
        SetCurrentTime(p_OpenTO);
        SetPermit(p_OpenTOON);
        // WaitSecond(&This>T, 1);
        break;
case 'c':
case '2':
case '3':
        Write(&This>F, "UserMotor.txt",'t');
        // printf("閉じます¥n");
        Write(&This>F, "UserPermitTurnOpen.txt",'o');
        SetPermit(p_OpenTOOFF);
        // WaitSecond(&This>T, 1);
        break;
default:
        break;
}
}
else if((This->strLimit[5] == 'y') || (This->strLimit[6] == 'y')){
    switch(UserCommand[1]){
case 'U':
        if(UserCommand[0] == '1'){
            Write(&This>F, "UserMotor.txt",'o');
            // printf("OPENします¥n");
            // WaitSecond(&This>T, 1);
        }
        else{
            Write(&This>F, "UserMotor.txt",'c');
            // printf("CLOSEします¥n");

```



```

        Write(&This>F, "UserPermitTurnOpen.txt"ρ');
        // WaitSecond(&This>T, 1);
    }
    break;
case 'o':
case 'L':
case '1':
    Write(&This>F, "UserMotor.txt";'o');
    // printf("OPENします¥n");
    // WaitSecond(&This>T, 1);
    break;
default:
    Write(&This>F, "UserMotor.txt";'c');
    // printf("CLOSEします¥n");
    Write(&This>F, "UserPermitTurnOpen.txt"ρ');
    // WaitSecond(&This>T, 1);
    break;
}
}
else{
    switch(UserCommand[1]){
    case 'U':
        if(UserCommand[0]== '1'){
            Write(&This>F, "UserMotor.txt";'O');
            // printf("高速OPENします¥n");
            // WaitSecond(&This>T, 1);
        }
        else{
            Write(&This>F, "UserMotor.txt";'C');

```

```

        // printf("高速CLOSEします¥n");
        Write(&This>F, "UserPermitTurnOpen.txt"ρ');
        // WaitSecond(&This>T, 1);
    }
    break;
case 'o':
case 'L':
case '1':
    Write(&This>F, "UserMotor.txt";'O');
    // printf("高速OPENします¥n");
    // WaitSecond(&This>T, 1);
    break;
default:
    Write(&This>F, "UserMotor.txt";'C');
    // printf("高速CLOSEします¥n");
    Write(&This>F, "UserPermitTurnOpen.txt"ρ');
    // WaitSecond(&This>T, 1);
    break;
}
}
}
else if((This->strLimit[9] == 'u') && (This->strLimit[10] == 'd')){
    if(This->strLimit[4]== 'y'){ // 2閉
        switch(UserCommand[1]){
        case 'U':
            if(UserCommand[0]== '2'){
                Write(&This>F, "UserMotor.txt";'s');
                // printf("STOPします¥n");
            }
        }
    }
}

```

```

        // WaitSecond(&This>T, 1);

        Write(&This>F, "UserMotor.txt",'h');

        // printf("開きます¥n");

        // WaitSecond(&This>T, 1);
    }

    else if(UserCommand[0]== '1'){
        This->UD = 'D';

        if(This->chMotor== 'D'){
            Write(&This>F, "UserMotor.txt",'D');

            // printf("高速DOWNします¥n");

            // WaitSecond(&This>T, 1);
        }

        else{
            Write(&This>F, "UserMotor.txt",'s');

            // printf("STOPします¥n");

            Write(&This>F, "UserPermitTurnOpen.txt",'N');

            // WaitSecond(&This>T, 1);

            Write(&This>F, "UserMotor.txt",'k');

            // printf("下降します¥n");

            // WaitSecond(&This>T, 1);
        }
    }

    break;

case 'D':

    if(UserCommand[0]== '2'){
        Write(&This>F, "UserMotor.txt",'s');

        // printf("STOPします¥n");

        // WaitSecond(&This>T, 1);

        Write(&This>F, "UserMotor.txt",'h');
    }

```

```

        // printf("開きます¥n");
        // WaitSecond(&This>T, 1);
    }
else if(UserCommand[0]== '3'){
    This->UD = 'U';
    if(This->chMotor== 'U'){
        Write(&This>F, "UserMotor.txt",'U');
        // printf("高速UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    else{
        Write(&This>F, "UserMotor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "UserPermitTurnOpen.txt",'N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "UserMotor.txt",'j');
        // printf("上昇します¥n");
        // WaitSecond(&This>T, 1);
    }
}
break;
case 'y':
    if(UserCommand[0]== '2'){
        Write(&This>F, "UserMotor.txt",'s');
        // printf("STOPします¥n");
        // WaitSecond(&This>T, 1);
        Write(&This>F, "UserMotor.txt",'h');
        // printf("開きます¥n");
        // WaitSecond(&This>T, 1);

```

```

    }
    break;
case 'o':
case 'L':
case '2':
    Write(&This>F, "UserMotor.txt",'s');
    // printf("STOPします¥n");
    // WaitSecond(&This>T, 1);
    Write(&This>F, "UserMotor.txt",'h');
    // printf("開きます¥n");
    // WaitSecond(&This>T, 1);
    break;
case '1':
    This->UD = 'D';
    if(This->chMotor== 'D'){
        Write(&This>F, "UserMotor.txt",'D');
        // printf("高速DOWNします¥n");
        // WaitSecond(&This>T, 1);
    }
    else{
        Write(&This>F, "UserMotor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "UserPermitTurnOpen.txt",'N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "UserMotor.txt",'k');
        // printf("下降します¥n");
        // WaitSecond(&This>T, 1);
    }
    break;

```

```

case '3':
    This->UD = 'U';
    if(This->chMotor=='U'){
        Write(&This>F, "UserMotor.txt",'U');
        // printf("高速UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    else{
        Write(&This>F, "UserMotor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "UserPermitTurnOpen.txt",'N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "UserMotor.txt",'j');
        // printf("上昇します¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'c':
    Write(&This>F, "UserMotor.txt",'s');
    // printf("STOPします¥n");
    Write(&This>F, "UserPermitTurnOpen.txt",'N');
    // WaitSecond(&This>T, 1);
    break;
case 'h':
    if(UserCommand[0]=='2'){
        Write(&This>F, "UserMotor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "UserPermitTurnOpen.txt",'N');
    }

```

```

        // WaitSecond(&This>T, 1);
    }
    break;
default:
    break;
}
}
else if(This->strLimit[7]== 'y'){ // 2開
    switch(UserCommand[1]){
    case 'U':
    case 'D':
        if(UserCommand[0]== '2'){
            Write(&This>F, "UserMotor.txt",'s');
            // printf("STOPします¥n");
            // 現在開
            // printf("開いています¥n");
            SetCurrentTime(p_OpenTO);
            SetPermit(p_OpenTOON);
            // WaitSecond(&This>T, 1);
        }
        else{
            Write(&This>F, "UserMotor.txt",'t');
            // printf("閉じます¥n");
            Write(&This>F, "UserPermitTurnOpen.txt"'\0');
            SetPermit(p_OpenTOOFF);
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'y':

```

```

if(UserCommand[0]== '2'){
    Write(&This>F, "UserMotor.txt",'s');
    // printf("STOPします¥n");
    // 現在開
    // printf("開いています¥n");
    SetCurrentTime(p_OpenTO);
    SetPermit(p_OpenTOON);
    // WaitSecond(&This>T, 1);
}
break;

case 'o':
case 'L':
case '2':
    Write(&This>F, "UserMotor.txt",'s');
    // printf("STOPします¥n");
    // 現在開
    // printf("開いています¥n");
    SetCurrentTime(p_OpenTO);
    SetPermit(p_OpenTOON);
    // WaitSecond(&This>T, 1);
    break;

case 'c':
case '1':
case '3':
    Write(&This>F, "UserMotor.txt",'t');
    // printf("閉じます¥n");
    Write(&This>F, "UserPermitTurnOpen.txt"'\0');
    SetPermit(p_OpenTOOFF);
    // WaitSecond(&This>T, 1);

```



```

        break;
    case 'h':
        if(UserCommand[0]== '2'){
            Write(&This>F, "UserMotor.txt",'t');
            // printf("閉じます¥n");
            Write(&This>F, "UserPermitTurnOpen.txt"'\o');
            SetPermit(p_OpenTOOFF);
            // WaitSecond(&This>T, 1);
        }
        break;
    default:
        break;
}
}

else if((This->strLimit[5] == 'y') || (This->strLimit[6] == 'y')){
    switch(UserCommand[1]){
    case 'U':
    case 'D':
        if(UserCommand[0]== '2'){
            Write(&This>F, "UserMotor.txt",'o');
            // printf("OPENします¥n");
            // WaitSecond(&This>T, 1);
        }
        else{
            Write(&This>F, "UserMotor.txt",'c');
            // printf("CLOSEします¥n");
            Write(&This>F, "UserPermitTurnOpen.txt"'\o');
            // WaitSecond(&This>T, 1);
        }
    }
}

```

```

        break;
    case 'y':
        if(UserCommand[0]== '2'){
            Write(&This>F, "UserMotor.txt",'o');
            // printf("OPENします¥n");
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'o':
    case 'L':
    case '2':
        Write(&This>F, "UserMotor.txt",'o');
        // printf("OPENします¥n");
        // WaitSecond(&This>T, 1);
        break;
    default:
        Write(&This>F, "UserMotor.txt",'c');
        // printf("CLOSEします¥n");
        Write(&This>F, "UserPermitTurnOpen.txt"o');
        // WaitSecond(&This>T, 1);
        break;
    }
}
else{
    switch(UserCommand[1]){
    case 'U':
    case 'D':
        if(UserCommand[0]== '2'){

```

```

        Write(&This>F, "UserMotor.txt",'O');
        // printf("高速OPENします¥n");
        // WaitSecond(&This>T, 1);
    }
    else{
        Write(&This>F, "UserMotor.txt",'C');
        // printf("高速CLOSEします¥n");
        Write(&This>F, "UserPermitTurnOpen.txt"ρ');
        // WaitSecond(&This>T, 1);
    }
    break;
case 'y':
    if(UserCommand[0]== '2'){
        Write(&This>F, "UserMotor.txt",'O');
        // printf("高速OPENします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'o':
case 'L':
case '2':
    Write(&This>F, "UserMotor.txt",'O');
    // printf("高速OPENします¥n");
    // WaitSecond(&This>T, 1);
    break;
default:
    Write(&This>F, "UserMotor.txt",'C');
    // printf("高速CLOSEします¥n");
    Write(&This>F, "UserPermitTurnOpen.txt"ρ');

```


case 'D':

```
    if(UserCommand[0]== '2'){
        Write(&This>F, "UserMotor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "UserPermitTurnOpen.txt",'\N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "UserMotor.txt",'k');
        // printf("下降します¥n");
        // WaitSecond(&This>T, 1);
    }
    else if(UserCommand[0]== '3'){
        Write(&This>F, "UserMotor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "UserPermitTurnOpen.txt",'\N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "UserMotor.txt",'h');
        // printf("開きます¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
```

case 'y':

```
    if(UserCommand[0]== '2'){
        Write(&This>F, "UserMotor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "UserPermitTurnOpen.txt",'\N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "UserMotor.txt",'k');
        // printf("下降します¥n");
        // WaitSecond(&This>T, 1);
```

```

    }
    break;
case 'o':
case 'L':
case '3':
    Write(&This>F, "UserMotor.txt",'s');
    // printf("STOPします¥n");
    Write(&This>F, "UserPermitTurnOpen.txt",N');
    // WaitSecond(&This>T, 1);
    Write(&This>F, "UserMotor.txt",'h');
    // printf("開きます¥n");
    // WaitSecond(&This>T, 1);
    break;
case '1':
case '2':
    Write(&This>F, "UserMotor.txt",'s');
    // printf("STOPします¥n");
    Write(&This>F, "UserPermitTurnOpen.txt",N');
    // WaitSecond(&This>T, 1);
    Write(&This>F, "UserMotor.txt",'k');
    // printf("下降します¥n");
    // WaitSecond(&This>T, 1);
    break;
case 'c':
    Write(&This>F, "UserMotor.txt",'s');
    // printf("STOPします¥n");
    Write(&This>F, "UserPermitTurnOpen.txt",o');
    // WaitSecond(&This>T, 1);

```

```

        break;
    default:
        break;
    }
}
else if(This->strLimit[7]== 'y'){ // 3開
    switch(UserCommand[1]){
    case 'U':
        if(UserCommand[0]== '1'){
            Write(&This>F, "UserMotor.txt",'t');
            // printf("閉じます¥n");
            Write(&This>F, "UserPermitTurnOpen.txt"ρ');
            SetPermit(p_OpenTOOFF);
            // WaitSecond(&This>T, 1);
        }
        else if(UserCommand[0]== '2'){
            Write(&This>F, "UserMotor.txt",'t');
            // printf("閉じます¥n");
            Write(&This>F, "UserPermitTurnOpen.txt"ρ');
            SetPermit(p_OpenTOOFF);
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'D':
        if(UserCommand[0]== '2'){
            Write(&This>F, "UserMotor.txt",'t');
            // printf("閉じます¥n");
            Write(&This>F, "UserPermitTurnOpen.txt"ρ');
            SetPermit(p_OpenTOOFF);

```

```

        // WaitSecond(&This>T, 1);
    }
    else if(UserCommand[0]== '3'){
        Write(&This>F, "UserMotor.txt",'s');
        // printf("STOPします¥n");
        //   現在開
        // printf("開いています¥n");
        SetCurrentTime(p_OpenTO);
        SetPermit(p_OpenTOON);
        // WaitSecond(&This>T, 1);
    }
    break;
case 'o':
case 'L':
case '3':
    Write(&This>F, "UserMotor.txt",'s');
    // printf("STOPします¥n");
    //   現在開
    // printf("開いています¥n");
    SetCurrentTime(p_OpenTO);
    SetPermit(p_OpenTOON);
    // WaitSecond(&This>T, 1);
    break;
case 'c':
case '1':
case '2':
    Write(&This>F, "UserMotor.txt",'t');
    // printf("閉じます¥n");
    Write(&This>F, "UserPermitTurnOpen.txt"ο');

```



```

        SetPermit(p_OpenTOOFF);
        // WaitSecond(&This>T, 1);
        break;
default:
        break;
}
}
else if((This->strLimit[5] == 'y') || (This->strLimit[6] == 'y')){
    switch(UserCommand[1]){
case 'D':
        if(UserCommand[0] == '3'){
            Write(&This>F, "UserMotor.txt",'o');
            // printf("OPENします¥n");
            // WaitSecond(&This>T, 1);
        }
        else{
            Write(&This>F, "UserMotor.txt",'c');
            // printf("CLOSEします¥n");
            Write(&This>F, "UserPermitTurnOpen.txt"ρ');
            // WaitSecond(&This>T, 1);
        }
        break;
case 'o':
case 'L':
case '3':
        Write(&This>F, "UserMotor.txt",'o');
        // printf("OPENします¥n");
        // WaitSecond(&This>T, 1);
        break;

```

```

default:
    Write(&This>F, "UserMotor.txt",'c');
    // printf("CLOSEします¥n");
    Write(&This>F, "UserPermitTurnOpen.txt"'\0');
    // WaitSecond(&This>T, 1);
    break;
}
}
else{
    switch(UserCommand[1]){
    case 'D':
        if(UserCommand[0]== '3'){
            Write(&This>F, "UserMotor.txt",'O');
            // printf("高速OPENします¥n");
            // WaitSecond(&This>T, 1);
        }
        else{
            Write(&This>F, "UserMotor.txt",'C');
            // printf("高速CLOSEします¥n");
            Write(&This>F, "UserPermitTurnOpen.txt"'\0');
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'o':
    case 'L':
    case '3':
        Write(&This>F, "UserMotor.txt",'O');
        // printf("高速OPENします¥n");

```

```

        // WaitSecond(&This>T, 1);
        break;
    default:
        Write(&This>F, "UserMotor.txt",'C');
        // printf("高速CLOSEします¥n");
        Write(&This>F, "UserPermitTurnOpen.txt",'o');
        // WaitSecond(&This>T, 1);
        break;
    }
}
}

else if(This->strLimit[1] == 'y'){
    switch(UserCommand[1]){
    case 'U':
        if(UserCommand[0]== '1'){
            Write(&This>F, "UserMotor.txt",'d');
            // printf("DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
        else if(UserCommand[0]== '2'){
            Write(&This>F, "UserMotor.txt",'u');
            // printf("UPします¥n");
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'D':
        if(UserCommand[0]== '2'){
            Write(&This>F, "UserMotor.txt",'u');
            // printf("UPします¥n");

```

```

        // WaitSecond(&This>T, 1);
    }
    else if(UserCommand[0]== '3'){
        Write(&This>F, "UserMotor.txt",'u');
        // printf("UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'y':
    if(UserCommand[0]== '2'){
        Write(&This>F, "UserMotor.txt",'u');
        // printf("UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case '1':
    Write(&This>F, "UserMotor.txt",'d');
    // printf("DOWNします¥n");
    // WaitSecond(&This>T, 1);
    break;
case '2':
case '3':
    Write(&This>F, "UserMotor.txt",'u');
    // printf("UPします¥n");
    // WaitSecond(&This>T, 1);
    break;
default:
    break;
}

```

```

}

else if(This->strLimit[8] == 'd'){
    switch(UserCommand[1]){
    case 'U':
        if(UserCommand[0] == '1'){
            Write(&This>F, "UserMotor.txt",'D');
            // printf("高速DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
        else if(UserCommand[0] == '2'){
            Write(&This>F, "UserMotor.txt",'U');
            // printf("高速UPします¥n");
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'D':
        if(UserCommand[0] == '2'){
            Write(&This>F, "UserMotor.txt",'U');
            // printf("高速UPします¥n");
            // WaitSecond(&This>T, 1);
        }
        else if(UserCommand[0] == '3'){
            Write(&This>F, "UserMotor.txt",'U');
            // printf("高速UPします¥n");
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'y':
        if(UserCommand[0] == '2'){

```

```

        Write(&This>F, "UserMotor.txt",'U');
        // printf("高速UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case '1':
    Write(&This>F, "UserMotor.txt",'D');
    // printf("高速DOWNします¥n");
    // WaitSecond(&This>T, 1);
    break;
case '2':
case '3':
    Write(&This>F, "UserMotor.txt",'U');
    // printf("高速UPします¥n");
    // WaitSecond(&This>T, 1);
    break;
default:
    break;
}
}
else if(This->strLimit[9] == 'd'){
    switch(UserCommand[1]){
    case 'U':
        if(UserCommand[0] == '1'){
            Write(&This>F, "UserMotor.txt",'d');
            // printf("DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
    }
}

```

```

else if(UserCommand[0]== '2'){
    Write(&This>F, "UserMotor.txt",'u');
    // printf("UPします¥n");
    // WaitSecond(&This>T, 1);
}
break;
case 'D':
    if(UserCommand[0]== '2'){
        Write(&This>F, "UserMotor.txt",'u');
        // printf("UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    else if(UserCommand[0]== '3'){
        Write(&This>F, "UserMotor.txt",'U');
        // printf("高速UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'y':
    if(UserCommand[0]== '2'){
        Write(&This>F, "UserMotor.txt",'u');
        // printf("UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case '1':
    Write(&This>F, "UserMotor.txt",'d');
    // printf("DOWNします¥n");
    // WaitSecond(&This>T, 1);

```

```

        break;
    case '2':
        Write(&This>F, "UserMotor.txt",'u');
        // printf("UPします¥n");
        // WaitSecond(&This>T, 1);
        break;
    case '3':
        Write(&This>F, "UserMotor.txt",'U');
        // printf("高速UPします¥n");
        // WaitSecond(&This>T, 1);
        break;
    default:
        break;
}
}
else if((This->strLimit[10] == 'u') && (This->strLimit[11] == 'd')){
    switch(UserCommand[1]){
    case 'U':
        if(UserCommand[0] == '1'){
            Write(&This>F, "UserMotor.txt",'D');
            // printf("高速DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
        else if(UserCommand[0] == '2'){
            Write(&This>F, "UserMotor.txt",'d');
            // printf("DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
        break;

```



```

case 'D':
    if(UserCommand[0]== '2'){
        Write(&This>F, "UserMotor.txt",'d');
        // printf("DOWNします¥n");
        // WaitSecond(&This>T, 1);
    }
    else if(UserCommand[0]== '3'){
        Write(&This>F, "UserMotor.txt",'u');
        // printf("UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'y':
    if(UserCommand[0]== '2'){
        Write(&This>F, "UserMotor.txt",'d');
        // printf("DOWNします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case '1':
    Write(&This>F, "UserMotor.txt",'D');
    // printf("高速DOWNします¥n");
    // WaitSecond(&This>T, 1);
    break;
case '2':
    Write(&This>F, "UserMotor.txt",'d');
    // printf("DOWNします¥n");
    // WaitSecond(&This>T, 1);
    break;

```



```

    }

    else if(UserCommand[0]== '3'){

        Write(&This>F, "UserMotor.txt",'U');

        // printf("高速UPします¥n");

        // WaitSecond(&This>T, 1);

    }

    break;

case 'y':

    if(UserCommand[0]== '2'){

        Write(&This>F, "UserMotor.txt",'D');

        // printf("高速DOWNします¥n");

        // WaitSecond(&This>T, 1);

    }

    break;

case '1':

case '2':

    Write(&This>F, "UserMotor.txt",'D');

    // printf("高速DOWNします¥n");

    // WaitSecond(&This>T, 1);

    break;

case '3':

    Write(&This>F, "UserMotor.txt",'U');

    // printf("高速UPします¥n");

    // WaitSecond(&This>T, 1);

    break;

default:

    break;

}

}

```

```

else if(This->strLimit[2] == 'y'){
    switch(UserCommand[1]){
    case 'U':
        if(UserCommand[0]== '1'){
            Write(&This>F, "UserMotor.txt",'d');
            // printf("DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
        else if(UserCommand[0]== '2'){
            Write(&This>F, "UserMotor.txt",'d');
            // printf("DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'D':
        if(UserCommand[0]== '2'){
            Write(&This>F, "UserMotor.txt",'d');
            // printf("DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
        else if(UserCommand[0]== '3'){
            Write(&This>F, "UserMotor.txt",'u');
            // printf("UPします¥n");
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'y':
        if(UserCommand[0]== '2'){
            Write(&This>F, "UserMotor.txt",'d');

```

```

        // printf("DOWNします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case '1':
case '2':
    Write(&This>F, "UserMotor.txt",'d');
    // printf("DOWNします¥n");
    // WaitSecond(&This>T, 1);
    break;
case '3':
    Write(&This>F, "UserMotor.txt",'u');
    // printf("UPします¥n");
    // WaitSecond(&This>T, 1);
    break;
default:
    break;
}
}
return;
}

void EV_Q_OTOUChange(struct EV_Queue *This, struct EV_Time *p_OpenTO, Thread *th)
{
    char *p_cmd;
    char cmd;
    p_cmd = &cmd;
    cmd = 'N';

```

```

This->p_tOpenTO = p_OpenTO;
This->p_strLimit = &This->strLimit[0];
ReadString(&This->F, "UserLimit.txt", This->p_strLimit, LIMIT);
if((This->UD == 'U') && (This->p_strLimit[9] == 'u') && (This->p_strLimit[10] == 'd'))
{
    // 0L,0o,02,2U,2y,0c,2h,03,3D
    if(EV_Q_Check(This,
"0L    ", "0o    ", "02    ", "2U    ", "2y    ", "NN    ", "NN    ", "NN    ", "NN    ", "NN    ", "NN    ", "NN    ") == OK)
    {
        WriteString(&This->F, "UserCommand.txt", "0o    ");
        EV_Q_Command_Read(This, p_cmd, th);
        EV_Q_CheckTurnOpen(This);
        SetCurrentTime(This->p_tOpenTO);
        SetPermit(This->p_tOpenTO, ON);
    }
    else if(EV_Q_Check(This,
"0c    ", "2h    ", "03    ", "3D    ", "NN    ", "NN    ", "NN    ", "NN    ", "NN    ", "NN    ", "NN    ", "NN    ") ==
ONE_MORE_TIME)
    {
        if(EV_Q_Check(This,
"2D    ", "01    ", "1U    ", "NN    ", "NN    ", "NN    ", "NN    ", "NN    ", "NN    ", "NN    ", "NN    ", "NN    ") == OK)
        {
            WriteString(&This->F, "UserCommand.txt", "0c    ");
            EV_Q_Command_Read(This, p_cmd, th);
            EV_Q_Motor(This, p_OpenTO);
            This->UD = 'D';
            EV_Q_Write(This, "UserCommand.txt", "2U    ", 'r');
        }
    }
}

```

```

else
{
    WriteString(&This->F, "UserCommand.txt", "0c¥0");
    EV_Q_Command_Read(Thisp_cmd, th);
    EV_Q_Motor(This, p_OpenTO);
}
}
else
{
    WriteString(&This->F, "UserCommand.txt", "0c¥0");
    EV_Q_Command_Read(This, p_cmd, th);
    EV_Q_Motor(This, p_OpenTO);
}
}
else if((This->UD == 'D') && (This->p_strLimit[9] == 'u') && (This->p_strLimit[10] == 'd'))
{
    // 0L,0o,02,2D,2y,0c,2h,01,1U
    if(EV_Q_Check(This,
"0L¥0", "0o¥0", "02¥0", "2D¥0", "2y¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0") == OK)
    {
        WriteString(&This->F, "UserCommand.txt", "0o¥0");
        EV_Q_Command_Read(This, p_cmd, th);
        EV_Q_CheckTurnOpen(This);
        SetCurrentTime(This->p_tOpenTO);
        SetPermit(This->p_tOpenTO, ON);
    }
    else if(EV_Q_Check(This,
"0c¥0", "2h¥0", "01¥0", "1U¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0") ==
ONE_MORE_TIME)

```

```

{
    if(EV_Q_Check(This,
"2U 0","03 0","3D 0","NN 0","NN 0","NN 0","NN 0","NN 0","NN 0","NN 0","NN 0") == OK)
    {
        WriteString(&This->F, "UserCommand.txt", "0c 0");
        EV_Q_Command_Read(Thisp_cmd, th);
        EV_Q_Motor(This, p_OpenTO);
        This->UD = 'U';
        EV_Q_Write(This, "UserCommand.txt", "2D 0", 'r');
    }
    else
    {
        WriteString(&This->F, "UserCommand.txt", "0c 0");
        EV_Q_Command_Read(Thisp_cmd, th);
        EV_Q_Motor(This, p_OpenTO);
    }
}
else
{
    WriteString(&This->F, "UserCommand.txt", "0c 0");
    EV_Q_Command_Read(This, p_cmd, th);
    EV_Q_Motor(This, p_OpenTO);
}
}
else
{
    WriteString(&This->F, "UserCommand.txt", "0c 0");
    EV_Q_Command_Read(This, p_cmd, th);
    EV_Q_Motor(This, p_OpenTO);
}
}
else
{
    WriteString(&This->F, "UserCommand.txt", "0c 0");
    EV_Q_Command_Read(This, p_cmd, th);
    EV_Q_Motor(This, p_OpenTO);
}
}

```



```
}
```

```
return;
```

```
}
```

```
// EV_Twin_Queue.h
```

```
#ifndef Panel_h  
#define Panel_h  
#include "Panel.h"  
#endif
```

```
#ifndef Timer_h  
#define Timer_h  
#include "Timer.h"  
#endif
```

```
#ifndef EV_Time_h  
#define EV_Time_h  
#include "EV_Time.h"  
#endif
```

```
#ifndef EV_File_h  
#define EV_File_h  
#include "EV_File.h"  
#endif
```

```
struct EV_Twin_Queue{  
    struct EV_Time T;  
    struct EV_File F;  
    char strQueue[33];  
    char *p_strQueue;  
    char strLimit[LIMIT];  
    char *p_strLimit;  
    char chMotor;  
    char *p_chMotor;  
    char *p_TwinCommand;  
    struct EV_Time *p_tOpenTO;  
    char UD;  
};
```

```
void EV_Twin1_Q_Init(struct EV_Twin_Queue *This, Thread *th);  
void EV_Twin1_Q_Write(struct EV_Twin_Queue *This, char *filename, char *p_str, char mode);  
int EV_Twin1_Q_Command_Read(struct EV_Twin_Queue *This, char *p_Command, Thread *th);  
int EV_Twin1_Q_CheckTurnOpen(struct EV_Twin_Queue *This);  
int EV_Twin1_Q_Read(struct EV_Twin_Queue *This, char *p_check);  
int EV_Twin1_Q_Check(struct EV_Twin_Queue *This, char *c0, char *c1, char *c2, char *c3, char *c4, char *c5,  
char *c6, char *c7, char *c8, char *c9, char *c10, char *c11);  
int EV_Twin1_Q_Flow(struct EV_Twin_Queue *This);  
void EV_Twin1_Q_Motor(struct EV_Twin_Queue *This, struct EV_Time *p_OpenTO);  
void EV_Twin1_Q_OTOUChange(struct EV_Twin_Queue *This, struct EV_Time *p_OpenTO, Thread *th);
```

```
void EV_Twin2_Q_Init(struct EV_Twin_Queue *This, Thread *th);  
void EV_Twin2_Q_Write(struct EV_Twin_Queue *This, char *filename, char *p_str, char mode);  
int EV_Twin2_Q_Command_Read(struct EV_Twin_Queue *This, char *p_Command, Thread *th);  
int EV_Twin2_Q_CheckTurnOpen(struct EV_Twin_Queue *This);  
int EV_Twin2_Q_Read(struct EV_Twin_Queue *This, char *p_check);  
int EV_Twin2_Q_Check(struct EV_Twin_Queue *This, char *c0, char *c1, char *c2, char *c3, char *c4, char *c5,  
char *c6, char *c7, char *c8, char *c9, char *c10, char *c11);  
int EV_Twin2_Q_Flow(struct EV_Twin_Queue *This);  
void EV_Twin2_Q_Motor(struct EV_Twin_Queue *This, struct EV_Time *p_OpenTO);  
void EV_Twin2_Q_OTOUChange(struct EV_Twin_Queue *This, struct EV_Time *p_OpenTO, Thread *th);
```

```

// EV_Twin1_Queue.c

#include "C.h"
#include "EV_Twin_Queue.h"

void EV_Twin1_Q_Init(struct EV_Twin_Queue *This, Thread *th){
    int i;
    EV_Time(&This->T, th);
    EV_File(&This->F);
    This->p_strQueue = &This->strQueue[0];
    for(i = 0; i < 34; i++){
        This->strQueue[i]= 'N';
    }
    This->strQueue[34] = ' 0';
    // uYdyocHhqe0o0c0102031U2D2U3D2y2h0L 0
    // NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN 0
    This->UD = 'U';
    return;
}

void EV_Twin1_Q_Write(struct EV_Twin_Queue *This, char *filename, char *p_str, char mode)
{
    This->p_strLimit = &This->strLimit[0];
    ReadString(&This->F, "Twin1Limit.txt", This->p_strLimit, LIMIT);
    This->p_strQueue = &This->strQueue[0];
    if(filename[0] == 'C'){ // Command.txt
        switch(p_str[0]){
            case 'u':
                if(mode == 'w'){

```

```
        This->strQueue[0]= 'u';
    }
    else if(mode== 'r'){
        This->strQueue[0]= 'N';
    }
    break;
case 'Y':
    if(mode== 'w'){
        This->strQueue[1]= 'Y';
    }
    else if(mode== 'r'){
        This->strQueue[1]= 'N';
    }
    break;
case 'd':
    if(mode== 'w'){
        This->strQueue[2]= 'd';
    }
    else if(mode== 'r'){
        This->strQueue[2]= 'N';
    }
    break;
case 'y':
    if(mode== 'w'){
        This->strQueue[3]= 'y';
    }
    else if(mode== 'r'){
        This->strQueue[3]= 'N';
    }
}
```

```
        break;
case 'o':
    if(mode == 'w'){
        This->strQueue[4] = 'o';
    }
    else if(mode == 'r'){
        This->strQueue[4] = 'N';
    }
    break;
case 'c':
    if(mode == 'w'){
        This->strQueue[5] = 'c';
    }
    else if(mode == 'r'){
        This->strQueue[5] = 'N';
    }
    break;
case 'H':
    if(mode == 'w'){
        This->strQueue[6] = 'H';
    }
    else if(mode == 'r'){
        This->strQueue[6] = 'N';
    }
    break;
case 'h':
    if(mode == 'w'){
        This->strQueue[7] = 'h';
    }
}
```

```

        else if(mode== 'r'){
            This->strQueue[7]= 'N';
        }
        break;
case 'q':
    if(mode== 'w'){
        This->strQueue[8]= 'q';
    }
    else if(mode== 'r'){
        This->strQueue[8]= 'N';
    }
    break;
case 'e':
    if(mode== 'w'){
        This->strQueue[9]= 'e';
    }
    else if(mode== 'r'){
        This->strQueue[9]= 'N';
    }
    break;
default:
    break;
}
}

else if((filename[0] == 'T') && (filename[4] == '1')){ // Twin1Command.txt
    switch(p_str[1]){
    case 'o':
        if(mode== 'w'){
            This->strQueue[10]= '0';

```

```

        This->strQueue[11]= 'o';
    }
    else if(mode== 'r'){
        This->strQueue[10]= 'N';
        This->strQueue[11]= 'N';
    }
    break;
case 'c':
    if(mode== 'w'){
        This->strQueue[12]= '0';
        This->strQueue[13]= 'c';
    }
    else if(mode== 'r'){
        This->strQueue[12]= 'N';
        This->strQueue[13]= 'N';
    }
    break;
case '1':
    if(mode== 'w'){
        if((This->strQueue[14] == '0') && (This->strQueue[15] == '1')){
            This->strQueue[14]= 'N';
            This->strQueue[15]= 'N';
            // printf("1階行きキャンセル\n");
        }
        else{
            This->strQueue[14]= '0';
            This->strQueue[15]= '1';
            // printf("1階行きON\n");
        }
    }

```

```

    }
}
else if(mode == 'r'){
    This->strQueue[14] = 'N';
    This->strQueue[15] = 'N';
}
break;
case '2':
    if(mode == 'w'){
        if((This->strQueue[16] == '0') && (This->strQueue[17] == '2')){
            This->strQueue[16] = 'N';
            This->strQueue[17] = 'N';
            // printf("2階行きキャンセル\n");
        }
        else{
            This->strQueue[16] = '0';
            This->strQueue[17] = '2';
            // printf("2階行きON\n");
        }
    }
    else if(mode == 'r'){
        This->strQueue[16] = 'N';
        This->strQueue[17] = 'N';
    }
    break;
case '3':
    if(mode == 'w'){
        if((This->strQueue[18] == '0') && (This->strQueue[19] == '3')){

```



```

        This->strQueue[18]= 'N';
        This->strQueue[19]= 'N';
        // printf("3階行きキャンセル¥n");
    }
    else{
        This->strQueue[18]= '0';
        This->strQueue[19]= '3';
        // printf("3階行きON¥n");
    }
}
else if(mode== 'r'){
    This->strQueue[18]= 'N';
    This->strQueue[19]= 'N';
}
break;
case 'U':
    if(mode== 'w'){
        if(p_str[0]== '1'){
            if((This->strQueue[20]== '1') && (This->strQueue[21]== 'U')){
                This->strQueue[20]= 'N';
                This->strQueue[21]= 'N';
                // printf("1階から上行きキャンセル¥n");
            }
            else{
                This->strQueue[20]= '1';
                This->strQueue[21]= 'U';
                // printf("1階から上行きON¥n");
            }
        }
    }
}

```

```

else if(p_str[0]== '2'){
    if((This->strQueue[24]== '2') && (This->strQueue[25]== 'U')){
        This->strQueue[24]= 'N';
        This->strQueue[25]= 'N';
        // printf("2階から上行きキャンセル¥n");
    }
    else{
        This->strQueue[24]= '2';
        This->strQueue[25]= 'U';
        // printf("2階から上行きON¥n");
    }
}
}

else if(mode== 'r'){
    if(p_str[0]== '1'){
        This->strQueue[20]= 'N';
        This->strQueue[21]= 'N';
    }
    else if(p_str[0]== '2'){
        This->strQueue[24]= 'N';
        This->strQueue[25]= 'N';
    }
}

break;

case 'D':
    if(mode== 'w'){
        if(p_str[0]== '2'){
            if((This->strQueue[22]== '2') && (This->strQueue[23]== 'D')){
                This->strQueue[22]= 'N';
            }
        }
    }
}

```

```

        This->strQueue[23]= 'N';
        // printf("2階から下行きキャンセル¥n");
    }
    else{
        This->strQueue[22]= '2';
        This->strQueue[23]= 'D';
        // printf("2階から下行きON¥n");
    }
}
else if(p_str[0]== '3'){
    if((This->strQueue[26]== '3') && (This->strQueue[27]== 'D')){
        This->strQueue[26]= 'N';
        This->strQueue[27]= 'N';
        // printf("3階から下行きキャンセル¥n");
    }
    else{
        This->strQueue[26]= '3';
        This->strQueue[27]= 'D';
        // printf("3階から下行きON¥n");
    }
    This->strQueue[26]= '3';
    This->strQueue[27]= 'D';
}
}
else if(mode== 'r'){
    if(p_str[0]== '2'){
        This->strQueue[22]= 'N';
        This->strQueue[23]= 'N';
    }
}

```

```

    }
    else if(p_str[0]== '3'){
        This->strQueue[26]= 'N';
        This->strQueue[27]= 'N';
    }
}
break;
case 'y':
    if(mode== 'w'){
        if(p_str[0]== '2'){
            This->strQueue[28]= '2';
            This->strQueue[29]= 'y';
        }
    }
    else if(mode== 'r'){
        if(p_str[0]== '2'){
            This->strQueue[28]= 'N';
            This->strQueue[29]= 'N';
        }
    }
    break;
case 'h':
    if(mode== 'w'){
        if(p_str[0]== '2'){
            This->strQueue[30]= '2';
            This->strQueue[31]= 'h';
        }
    }
    else if(mode== 'r'){

```

```

        if(p_str[0]== '2'){
            This->strQueue[30]= 'N';
            This->strQueue[31]= 'N';
        }
    }
    break;
case 'L':
    if(mode== 'w'){
        if((This->strQueue[32] == '0') && (This->strQueue[33] == 'L')){
            This->strQueue[32]= 'N';
            This->strQueue[33]= 'N';
            // printf("開延長キャンセル¥n");
        }
        else if((This->p_strLimit[6] == 'y') && (This->p_strLimit[7] == 'y'))
        {
            This->strQueue[32]= '0';
            This->strQueue[33]= 'L';
            // printf("開延長ON¥n");
        }
    }
    else if(mode== 'r'){
        This->strQueue[32]= 'N';
        This->strQueue[33]= 'N';
    }
    break;
default:
    break;
}
}

```

```
if(This->strQueue[0] == 'N' &&  
    This->strQueue[1] == 'N' &&  
    This->strQueue[2] == 'N' &&  
    This->strQueue[3] == 'N' &&  
    This->strQueue[4] == 'N' &&  
    This->strQueue[5] == 'N' &&  
    This->strQueue[6] == 'N' &&  
    This->strQueue[7] == 'N' &&  
    This->strQueue[8] == 'N' &&  
    This->strQueue[9] == 'N' &&  
    This->strQueue[10] == 'N' &&  
    This->strQueue[11] == 'N' &&  
    This->strQueue[12] == 'N' &&  
    This->strQueue[13] == 'N' &&  
    This->strQueue[14] == 'N' &&  
    This->strQueue[15] == 'N' &&  
    This->strQueue[16] == 'N' &&  
    This->strQueue[17] == 'N' &&  
    This->strQueue[18] == 'N' &&  
    This->strQueue[19] == 'N' &&  
    This->strQueue[20] == 'N' &&  
    This->strQueue[21] == 'N' &&  
    This->strQueue[22] == 'N' &&  
    This->strQueue[23] == 'N' &&  
    This->strQueue[24] == 'N' &&  
    This->strQueue[25] == 'N' &&  
    This->strQueue[26] == 'N' &&  
    This->strQueue[27] == 'N' &&  
    This->strQueue[28] == 'N' &&
```

```

    This->strQueue[29] == 'N' &&
    This->strQueue[30] == 'N' &&
    This->strQueue[31] == 'N' &&
    This->strQueue[32] == 'N' &&
    This->strQueue[33] == 'N')
{
    if((This->p_strLimit[6] == 'y') && (This->p_strLimit[7] == 'y'))
    {
        Write(&This->F, "Twin1Motor.txt", 's');
    }
    else if((This->strQueue[32] == '0') && (This->strQueue[33] == 'L')){
        This->strQueue[32]= 'N';
        This->strQueue[33]= 'N';
    }
}
return;
}

int EV_Twin1_Q_Command_Read(struct EV_Twin_Queue *This, char *p_Command, Thread *th){
    char ch;
    char *p_ch;
    char str[3];
    char *p_str;

    This->p_strQueue = &This->strQueue[0];
    ch = 'N';
    p_ch = &ch;
    str[0] = 'N';
    str[1] = 'N';

```

```

str[2] = '¥n';

p_str = &str[0];

ReadString(&This->F, "Twin1Command.txt", p_str, 3);

// write queue
EV_Twin1_Q_Write(This, "Twin1Command.txt", p_str, 'w');

// remove TwinCommand.txt
WriteString(&This->F, "Twin1Command.txt", "NN¥0");

*p_Command = ch;
if((ch == 'u') || (ch == 'Y') || (ch == 'd') || (ch == 'y')
|| (ch == 'o') || (ch == 'c') || (ch == 'H') || (ch == 'h')
|| (ch == 'e')){
    // 命令入力許可終了
    //Write(&This->F, "Twin1PermitCommand.txt", 'N');

// 命令信号保持
    // WaitSecond(&This->T, 1);

    // 消命令
    // Write(&This->F, "Command.txt", 'N');

// remove queue
EV_Twin1_Q_Init(This, th);

return OK;
}

```



```
else if(ch == 'q'){
    // 命令入力許可終了
    Write(&This->F, "Twin1PermitCommand.txt", 'N');
```

```
// 命令信号保持
```

```
    // WaitSecond(&This->T, 3);
```

```
    // 消命令
```

```
    // Write(&This->F, "Command.txt", 'N');
```

```
    // WriteString(&This->F, "Twin1Command.txt", "NN¥0");
```

```
    // remove queue
```

```
    EV_Twin1_Q_Init(This, th);
```

```
    return OK;
```

```
}
```

```
return ONE_MORE_TIME;
```

```
}
```

```
int EV_Twin1_Q_CheckTurnOpen(struct EV_Twin_Queue *This){
```

```
    int Ret;
```

```
    char strTurnOpen[3];
```

```
    char *p_strTurnOpen;
```

```
    char PermitTurnOpen;
```

```
    strTurnOpen[0] = 'N';
```

```
    strTurnOpen[0] = 'N';
```

```
    strTurnOpen[0] = '¥0';
```

```
    PermitTurnOpen = 'N';
```

```
    p_strTurnOpen = &strTurnOpen[0];
```

```
    ReadString(&This->F, "Twin1TurnOpen.txt", p_strTurnOpen, 3);
```

```

// Wait_ms(&This->T, 200);

Read(&This->F, "Twin1PermitTurnOpen.txt", &PermitTurnOpen);

// Wait_ms(&This->T, 200);

if(PermitTurnOpen == 'N'){
    Ret = ONE_MORE_TIME;
}

else{
    // 閉中斷時
    if(p_strTurnOpen[0] == 'N'){
        Ret = ONE_MORE_TIME;
    }

    else if((p_strTurnOpen[0] == '0') && (p_strTurnOpen[1] == 'o')){
        Write(&This->F, "Twin1Motor.txt", 'h');
        EV_Twin1_Q_Write(This, "Twin1Command.txt", "0c¥0", 'r');
        This->p_TwinCommand[0]= '0';
        This->p_TwinCommand[1]= 'o';
        // printf("反転開します¥n");
        WriteString(&This->F, "Twin1Command.txt", "0o¥0");
        Write(&This->F, "Twin1PermitCommand.txt", 'c');
        WriteString(&This->F, "Twin1TurnOpen.txt", "NN¥0");
        Write(&This->F, "Twin1PermitTurnOpen.txt", 'N');
        // WaitSecond(&This>T, 1);
        Ret = NG;
    }

    else if((p_strTurnOpen[0] == '0') && (p_strTurnOpen[1] == 'L')){
        Write(&This->F, "Twin1Motor.txt", 'h');
        EV_Twin1_Q_Write(This, "Twin1Command.txt", "0c¥0", 'r');
        This->p_TwinCommand[0]= '0';
        This->p_TwinCommand[1]= 'o';
    }
}

```

```

// printf("反転開します¥n");
WriteString(&This->F, "Twin1Command.txt", "0L¥0");
Write(&This->F, "Twin1PermitCommand.txt", 'c');
WriteString(&This->F, "Twin1TurnOpen.txt", "NN¥0");
Write(&This->F, "Twin1PermitTurnOpen.txt", 'N');
// WaitSecond(&This>T, 1);
Ret = NG;
}
else if((This->p_strLimit[0] == 'y') && ((p_strTurnOpen[0] == '1') || (p_strTurnOpen[1] ==
'1'))){
Write(&This->F, "Twin1Motor.txt", 'h');
EV_Twin1_Q_Write(This, "Twin1Command.txt", "0c¥0", 'r');
This->p_TwinCommand[0]= '0';
This->p_TwinCommand[1]= 'o';
// printf("反転開します¥n");
WriteString(&This->F, "Twin1Command.txt", "0o¥0");
Write(&This->F, "Twin1PermitCommand.txt", 'c');
WriteString(&This->F, "Twin1TurnOpen.txt", "NN¥0");
Write(&This->F, "Twin1PermitTurnOpen.txt", 'N');
// WaitSecond(&This>T, 1);
Ret = NG;
}
else if((This->p_strLimit[9] == 'u') && (This->p_strLimit[10] == 'd') && ((p_strTurnOpen[0] ==
'2') || (p_strTurnOpen[1] == '2'))){
Write(&This->F, "Twin1Motor.txt", 'h');
EV_Twin1_Q_Write(This, "Twin1Command.txt", "0c¥0", 'r');
EV_Twin1_Q_Write(This, "Twin1Command.txt", "2h¥0", 'r');
This->p_TwinCommand[0]= '0';

```

```

This->p_TwinCommand[1]= 'o';
// printf("反転開します¥n");
WriteString(&This->F, "Twin1Command.txt", "0o¥0");
Write(&This->F, "Twin1PermitCommand.txt", 'c');
WriteString(&This->F, "Twin1TurnOpen.txt", "NN¥0");
Write(&This->F, "Twin1PermitTurnOpen.txt", 'N');
// WaitSecond(&This>T, 1);
Ret = NG;
}
else if((This->p_strLimit[3] == 'y') && ((p_strTurnOpen[0] == '3') || (p_strTurnOpen[1] ==
'3'))){
Write(&This->F, "Twin1Motor.txt", 'h');
EV_Twin1_Q_Write(This, "Twin1Command.txt", "0c¥0", 'r');
This->p_TwinCommand[0]= '0';
This->p_TwinCommand[1]= 'o';
// printf("反転開します¥n");
WriteString(&This->F, "Twin1Command.txt", "0o¥0");
Write(&This->F, "Twin1PermitCommand.txt", 'c');
WriteString(&This->F, "Twin1TurnOpen.txt", "NN¥0");
Write(&This->F, "Twin1PermitTurnOpen.txt", 'N');
// WaitSecond(&This>T, 1);
Ret = NG;
}
else{
Ret = ONE_MORE_TIME;
}
}
return Ret;
}

```

```

int EV_Twin1_Q_Read(struct EV_Twin_Queue *This, char *p_check){
    int Ret;
    This->p_strQueue = &This->strQueue[0];
    Ret = EV_Twin1_Q_CheckTurnOpen(This);
    if(Ret == NG) return NG;
    switch(p_check[1]){
    case 'o':
        if(This->strQueue[11] == 'o'){
            This->p_TwinCommand[0]= '0';
            This->p_TwinCommand[1]= 'o';
            Ret = OK;
        }
        else{
            This->p_TwinCommand[0]= 'N';
            This->p_TwinCommand[1]= 'N';
            Ret = ONE_MORE_TIME;
        }
        break;
    case 'c':
        if(This->strQueue[13] == 'c'){
            This->p_TwinCommand[0]= '0';
            This->p_TwinCommand[1]= 'c';
            Ret = OK;
        }
        else{
            This->p_TwinCommand[0]= 'N';
            This->p_TwinCommand[1]= 'N';
            Ret = ONE_MORE_TIME;
        }
    }
}

```

```
break;
case '1':
    if(This->strQueue[15] == '1'){
        This->p_TwinCommand[0]= '0';
        This->p_TwinCommand[1]= '1';
        Ret = OK;
    }
    else{
        This->p_TwinCommand[0]= 'N';
        This->p_TwinCommand[1]= 'N';
        Ret = ONE_MORE_TIME;
    }
    break;
case '2':
    if(This->strQueue[17] == '2'){
        This->p_TwinCommand[0]= '0';
        This->p_TwinCommand[1]= '2';
        Ret = OK;
    }
    else{
        This->p_TwinCommand[0]= 'N';
        This->p_TwinCommand[1]= 'N';
        Ret = ONE_MORE_TIME;
    }
    break;
case '3':
    if(This->strQueue[19] == '3'){
        This->p_TwinCommand[0]= '0';
        This->p_TwinCommand[1]= '3';
```

```

        Ret = OK;
    }
    else{
        This->p_TwinCommand[0]= 'N';
        This->p_TwinCommand[1]= 'N';
        Ret = ONE_MORE_TIME;
    }
    break;
case 'U':
    if(p_check[0]== '1'){
        if(This->strQueue[21]== 'U'){
            This->p_TwinCommand[0]= '1';
            This->p_TwinCommand[1]= 'U';
            Ret = OK;
        }
        else{
            This->p_TwinCommand[0]= 'N';
            This->p_TwinCommand[1]= 'N';
            Ret = ONE_MORE_TIME;
        }
    }
    if(p_check[0]== '2'){
        if(This->strQueue[25]== 'U'){
            This->p_TwinCommand[0]= '2';
            This->p_TwinCommand[1]= 'U';
            Ret = OK;
        }
        else{

```

```

        This->p_TwinCommand[0]= 'N';
        This->p_TwinCommand[1]= 'N';
        Ret = ONE_MORE_TIME;
    }
}
break;
case 'D':
    if(p_check[0]== '2'){
        if(This->strQueue[23]== 'D'){
            This->p_TwinCommand[0]= '2';
            This->p_TwinCommand[1]= 'D';
            Ret = OK;
        }
        else{
            This->p_TwinCommand[0]= 'N';
            This->p_TwinCommand[1]= 'N';
            Ret = ONE_MORE_TIME;
        }
    }
    if(p_check[0]== '3'){
        if(This->strQueue[27]== 'D'){
            This->p_TwinCommand[0]= '3';
            This->p_TwinCommand[1]= 'D';
            Ret = OK;
        }
        else{
            This->p_TwinCommand[0]= 'N';
            This->p_TwinCommand[1]= 'N';
            Ret = ONE_MORE_TIME;
        }
    }
}

```



```

        }
    }
    break;
case 'y':
    if(p_check[0]== '2'){
        if(This->strQueue[29]== 'y'){
            This->p_TwinCommand[0]= '2';
            This->p_TwinCommand[1]= 'y';
            Ret = OK;
        }
        else{
            This->p_TwinCommand[0]= 'N';
            This->p_TwinCommand[1]= 'N';
            Ret = ONE_MORE_TIME;
        }
    }
    break;
case 'h':
    if(p_check[0]== '2'){
        if(This->strQueue[31]== 'h'){
            This->p_TwinCommand[0]= '2';
            This->p_TwinCommand[1]= 'h';
            Ret = OK;
        }
        else{
            This->p_TwinCommand[0]= 'N';
            This->p_TwinCommand[1]= 'N';
            Ret = ONE_MORE_TIME;
        }
    }

```

```

    }
    break;
case 'L':
    if(This->strQueue[33] == 'L'){
        This->p_TwinCommand[32]= '0';
        This->p_TwinCommand[33]= 'L';
        Ret = OK;
    }
    else{
        This->p_TwinCommand[32]= 'N';
        This->p_TwinCommand[33]= 'N';
        Ret = ONE_MORE_TIME;
    }
    break;
default:
    This->p_TwinCommand[0] = 'N';
    This->p_TwinCommand[1] = 'N';
    Ret = ONE_MORE_TIME;
    break;
}
return Ret;
}

int EV_Twin1_Q_Check(struct EV_Twin_Queue *This, char *c0, char *c1, char *c2, char *c3, char *c4,
char *c5, char *c6, char *c7, char *c8, char *c9, char *c10, char *c11){
    int Ret;
    Ret = EV_Twin1_Q_Read(This, c0);
    if(Ret == OK) return OK;
    if(Ret == NG) return NG;
    Ret = EV_Twin1_Q_Read(This, c1);

```

```
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Twin1_Q_Read(This, c2);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Twin1_Q_Read(This, c3);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Twin1_Q_Read(This, c4);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Twin1_Q_Read(This, c5);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Twin1_Q_Read(This, c6);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Twin1_Q_Read(This, c7);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Twin1_Q_Read(This, c8);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Twin1_Q_Read(This, c9);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Twin1_Q_Read(This, c10);
if(Ret == OK) return OK;
```

```

if(Ret == NG) return NG;

Ret = EV_Twin1_Q_Read(This, c11);

return Ret;

}

int EV_Twin1_Q_Flow(struct EV_Twin_Queue *This){

    int Ret;

    This->p_strLimit = &This->strLimit[0];

    This->p_chMotor = &This->chMotor;

    ReadString(&This->F, "Twin1Limit.txt", This->p_strLimit, LIMIT);

    // Wait_ms(&This->T, 200);

    Read(&This->F, "Twin1Motor.txt", This->p_chMotor);

    // Wait_ms(&This->T, 200);

    if((This->chMotor == 'j') || (This->chMotor == 'u') || (This->chMotor == 'U')){

        //    remove 0o,0c

        EV_Twin1_Q_Write(This, "Twin1Command.txt", "0o¥0", 'r');

        EV_Twin1_Q_Write(This, "Twin1Command.txt", "0c¥0", 'r');

        if((This->strLimit[9] == 'u') && (This->strLimit[10] == 'd') && ((This->chMotor == 'j') || (This-
>chMotor == 'u'))){

            //    02,2y,2U,03,3D,2D,01,1U

            Ret = EV_Twin1_Q_Check(This,

"02¥0","2y¥0","2U¥0","03¥0","3D¥0","2D¥0","01¥0","1U¥0","NN¥0","NN¥0","NN¥0","NN¥0");

        }

        else if(This->strLimit[9] == 'u'){

            //    03,3D,02,2D,2y,01,1U,2U

            Ret = EV_Twin1_Q_Check(This,

"03¥0","3D¥0","02¥0","2D¥0","2y¥0","01¥0","1U¥0","2U¥0","NN¥0","NN¥0","NN¥0","NN¥0");

        }

        else{

            //    02,2U,2y,03,3D,2D,01,1U

```

```

        Ret = EV_Twin1_Q_Check(This,
"02¥0","2U¥0","2y¥0","03¥0","3D¥0","2D¥0","01¥0","1U¥0","NN¥0","NN¥0","NN¥0","NN¥0");
    }
}
else if((This->chMotor == 'k') || (This->chMotor == 'd') || (This->chMotor == 'D')){
    // remove 0o,0c
    EV_Twin1_Q_Write(This, "Twin1Command.txt", "0o¥0", 'r');
    EV_Twin1_Q_Write(This, "Twin1Command.txt", "0c¥0", 'r');
    if((This->strLimit[9] == 'u') && (This->strLimit[10] == 'd') && ((This->chMotor == 'k') || (This-
>chMotor == 'd'))){
        // 02,2y,2D,01,1U,2U,03,3D
        Ret = EV_Twin1_Q_Check(This,
"02¥0","2y¥0","2D¥0","01¥0","1U¥0","2U¥0","03¥0","3D¥0","NN¥0","NN¥0","NN¥0","NN¥0");
    }
    else if(This->strLimit[10] == 'd'){
        // 01,1U,02,2U,2y,03,3D,2D
        Ret = EV_Twin1_Q_Check(This,
"01¥0","1U¥0","02¥0","2U¥0","2y¥0","03¥0","3D¥0","2D¥0","NN¥0","NN¥0","NN¥0","NN¥0");
    }
    else{
        // 02,2D,2y,01,1U,2U,03,3D
        Ret = EV_Twin1_Q_Check(This,
"02¥0","2D¥0","2y¥0","01¥0","1U¥0","2U¥0","03¥0","3D¥0","NN¥0","NN¥0","NN¥0","NN¥0");
    }
}
else if((This->chMotor == 'h') || (This->chMotor == 'o') || (This->chMotor == 'O')){
    if(This->strLimit[0] == 'y'){
        // 0L,0o,01,1U,0c,02,2U,03,3D,2D
        Ret = EV_Twin1_Q_Check(This,

```

```

"0L¥0","0o¥0","01¥0","1U¥0","0c¥0","02¥0","2U¥0","03¥0","3D¥0","2D¥0","NN¥0","NN¥0");
    }
    else if((This->strLimit[9] == 'u') && (This->strLimit[10] == 'd')){
        //    0L,0o,02,2D/2U,2y,0c,01/03,1U/3D
        Ret = EV_Twin1_Q_Check(This,
"0L¥0","0o¥0","02¥0","2D¥0","2U¥0","2y¥0","0c¥0","01¥0","03¥0","1U¥0","3D¥0","NN¥0");
    }
    else if(This->strLimit[3] == 'y'){
        //    0L,0o,03,3D,0c,02,2D,01,1U,2U
        Ret = EV_Twin1_Q_Check(This,
"0L¥0","0o¥0","03¥0","3D¥0","0c¥0","02¥0","2D¥0","01¥0","1U¥0","2U¥0","NN¥0","NN¥0");
    }
}
else if((This->chMotor == 't') || (This->chMotor == 'c') || (This->chMotor == 'C')){
    if(This->strLimit[0] == 'y'){
        //    0L,0o,01,1U,0c,02,2U,03,3D,2D
        Ret = EV_Twin1_Q_Check(This,
"0L¥0","0o¥0","01¥0","1U¥0","0c¥0","02¥0","2U¥0","03¥0","3D¥0","2D¥0","NN¥0","NN¥0");
    }
    else if((This->strLimit[9] == 'u') && (This->strLimit[10] == 'd')){
        //    0L,0o,02,2D/2U,2y,0c,2h,01/03,1U/3D
        //    Ret = EV_Twin1_Q_Check(This,
"0L¥0","0o¥0","02¥0","2D¥0","2U¥0","2y¥0","0c¥0","2h¥0","01¥0","03¥0","1U¥0","3D¥0");
        if(This->UD == 'U')
        {
            //    0L,0o,02,2U,2y,0c,2h,03,3D,2D,01,1U
            Ret = EV_Twin1_Q_Check(This,
"0L¥0","0o¥0","02¥0","2U¥0","2y¥0","0c¥0","2h¥0","03¥0","3D¥0","2D¥0","01¥0","1U¥0");
        }
    }
}

```

```

else if(This->UD == 'D')
{
    // 0L,0o,02,2D,2y,0c,2h,01,1U,2U,03,3D
    Ret = EV_Twin1_Q_Check(This,
"0L¥0","0o¥0","02¥0","2D¥0","2y¥0","0c¥0","2h¥0","01¥0","1U¥0","2U¥0","03¥0","3D¥0");
}
}

else if(This->strLimit[3] == 'y'){
    // 0L,0o,03,3D,0c,02,2D,01,1U,2U
    Ret = EV_Twin1_Q_Check(This,
"0L¥0","0o¥0","03¥0","3D¥0","0c¥0","02¥0","2D¥0","01¥0","1U¥0","2U¥0","NN¥0","NN¥0");
}
}

else if(This->chMotor == 's'){
    if(This->strLimit[0] == 'y'){
        if(This->strLimit[7]== 'y'){ // 1開
            // remove 0o,01,1U
            EV_Twin1_Q_Write(This, "Twin1Command.txt", "0o¥0", 'r');
            EV_Twin1_Q_Write(This, "Twin1Command.txt", "01¥0", 'r');
            EV_Twin1_Q_Write(This, "Twin1Command.txt", "1U¥0", 'r');
            // 0L,0c,02,2U,03,3D,2D
            // Ret = EV_Twin1_Q_Check(This,
"0L¥0","0c¥0","02¥0","2U¥0","03¥0","3D¥0","2D¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0");
            // 0L,0c
            Ret = EV_Twin1_Q_Check(This,
"0L¥0","0c¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0");
        }
        else if(This->strLimit[4]== 'y'){ // 1閉

```

```

// remove 0c
EV_Twin1_Q_Write(This, "Twin1Command.txt", "0c ", 'r');
// 0L,0o,01,1U,02,2U,2y,03,3D,2D
Ret = EV_Twin1_Q_Check(This,
"0L ", "0o ", "01 ", "1U ", "02 ", "2U ", "2y ", "03 ", "3D ", "2D ", "NN ", "NN ");
}
else{
// 0L,0o,01,1U,0c,02,2U,03,3D,2D
Ret = EV_Twin1_Q_Check(This,
"0L ", "0o ", "01 ", "1U ", "0c ", "02 ", "2U ", "03 ", "3D ", "2D ", "NN ", "NN ");
}
}
else if((This->strLimit[9] == 'u') && (This->strLimit[10] == 'd'))
{
if(This->strLimit[7] == 'y'){ // 2開
// remove 0o,02,2U,2D
EV_Twin1_Q_Write(This, "Twin1Command.txt", "0o ", 'r');
EV_Twin1_Q_Write(This, "Twin1Command.txt", "02 ", 'r');
if(This->UD == 'U')
{
EV_Twin1_Q_Write(This, "Twin1Command.txt", "2U ", 'r');
// 0L,0o,02,2U,2y,0c,2h,03,3D
Ret = EV_Twin1_Q_Check(This,
"0L ", "0o ", "02 ", "2U ", "2y ", "0c ", "2h ", "03 ", "3D ", "NN ", "NN ", "NN ");
if(Ret == ONE_MORE_TIME)
{
EV_Twin1_Q_Write(This, "Twin1Command.txt", "2D ", 'r');
}
}
}
}
}

```



```

else if(This->UD == 'D')
{
    EV_Twin1_Q_Write(This, "Twin1Command.txt", "2D¥0", 'r');
    // 0L,0o,02,2D,2y,0c,2h,01,1U
    Ret = EV_Twin1_Q_Check(This,
"0L¥0", "0o¥0", "02¥0", "2D¥0", "2y¥0", "0c¥0", "2h¥0", "01¥0", "1U¥0", "NN¥0", "NN¥0", "NN¥0");
    if(Ret == ONE_MORE_TIME)
    {
        EV_Twin1_Q_Write(This, "Twin1Command.txt", "2U¥0", 'r');
    }
}
EV_Twin1_Q_Write(This, "Twin1Command.txt", "2y¥0", 'r');
// 0L,0c,2h,01/03,1U/3D
// Ret = EV_Twin1_Q_Check(This,
"0L¥0", "0c¥0", "2h¥0", "01¥0", "03¥0", "1U¥0", "3D¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0");
// 0L,0c
Ret = EV_Twin1_Q_Check(This,
"0L¥0", "0c¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0");
}
else if(This->strLimit[4] == 'y'){ // 2閉
    // remove 0c,2h
    EV_Twin1_Q_Write(This, "Twin1Command.txt", "0c¥0", 'r');
    EV_Twin1_Q_Write(This, "Twin1Command.txt", "2h¥0", 'r');
    // 0L,0o,02,2D/2U,2y,01/03,1U/3D
    // Ret = EV_Twin1_Q_Check(This,
"0L¥0", "0o¥0", "02¥0", "2D¥0", "2U¥0", "2y¥0", "01¥0", "03¥0", "1U¥0", "3D¥0", "NN¥0", "NN¥0");
    if(This->UD == 'U')
    {
        // 0L,0o,02,2U,2y,03,3D,2D,01,1U

```

```

        Ret = EV_Twin1_Q_Check(This,
"0L¥0","0o¥0","02¥0","2U¥0","2y¥0","03¥0","3D¥0","2D¥0","01¥0","1U¥0","NN¥0","NN¥0");
    }
    else if(This->UD == 'D')
    {
        // 0L,0o,02,2D,2y,01,1U,2U,03,3D
        Ret = EV_Twin1_Q_Check(This,
"0L¥0","0o¥0","02¥0","2D¥0","2y¥0","01¥0","1U¥0","2U¥0","03¥0","3D¥0","NN¥0","NN¥0");
    }
}
else{
    // 0L,0o,02,2D/2U,2y,0c,2h,01/03,1U/3D
    Ret = EV_Twin1_Q_Check(This,
"0L¥0","0o¥0","02¥0","2D¥0","2U¥0","2y¥0","0c¥0","2h¥0","01¥0","03¥0","1U¥0","3D¥0");
}
}
else if(This->strLimit[3] == 'y'){
    if(This->strLimit[7] == 'y'){ // 3開
        // remove 0o,03,3D
        EV_Twin1_Q_Write(This, "Twin1Command.txt", "0o¥0", 'r');
        EV_Twin1_Q_Write(This, "Twin1Command.txt", "03¥0", 'r');
        EV_Twin1_Q_Write(This, "Twin1Command.txt", "3D¥0", 'r');
        // 0L,0c,02,2D,01,1U,2U
        // Ret = EV_Twin1_Q_Check(This,
"0L¥0","0c¥0","02¥0","2D¥0","01¥0","1U¥0","2U¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0");
        // 0L,0c
        Ret = EV_Twin1_Q_Check(This,
"0L¥0","0c¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0");
    }
}

```

```

else if(This->strLimit[4]== 'y'){ // 3閉
    // remove 0c
    EV_Twin1_Q_Write(This, "Twin1Command.txt", "0c ", 'r');
    // 0L,0o,03,3D,02,2D,2y,01,1U,2U
    Ret = EV_Twin1_Q_Check(This,
"0L ", "0o ", "03 ", "3D ", "02 ", "2D ", "2y ", "01 ", "1U ", "2U ", "NN ", "NN ");
    }
else{
    // 0L,0o,03,3D,0c,02,2D,01,1U,2U
    Ret = EV_Twin1_Q_Check(This,
"0L ", "0o ", "03 ", "3D ", "0c ", "02 ", "2D ", "01 ", "1U ", "2U ", "NN ", "NN ");
    }
}
else if(This->strLimit[9] == 'd'){
    // remove 0o,0c
    EV_Twin1_Q_Write(This, "Twin1Command.txt", "0o ", 'r');
    EV_Twin1_Q_Write(This, "Twin1Command.txt", "0c ", 'r');
    // 0L,01/02,1U/2U,2y,03,3D,2D
    Ret = EV_Twin1_Q_Check(This,
"0L ", "01 ", "02 ", "1U ", "2U ", "2y ", "03 ", "3D ", "2D ", "NN ", "NN ", "NN ");
    }
else if(This->strLimit[10] == 'u'){
    // remove 0o,0c
    EV_Twin1_Q_Write(This, "Twin1Command.txt", "0o ", 'r');
    EV_Twin1_Q_Write(This, "Twin1Command.txt", "0c ", 'r');
    // 0L,03/02,3D/2D,2y,01,1U,2U
    Ret = EV_Twin1_Q_Check(This,
"0L ", "03 ", "02 ", "3D ", "2D ", "2y ", "01 ", "1U ", "2U ", "NN ", "NN ", "NN ");

```

```

    }
}
return Ret;
}

void EV_Twin1_Q_Motor(struct EV_Twin_Queue *This, struct EV_Time *p_OpenTO){
    int EV_Twin1_Q_FlowRet;
    char TwinCommand[3];
    TwinCommand[0] = 'N';
    TwinCommand[1] = 'N';
    TwinCommand[2] = '¥0';
    This->p_TwinCommand = &TwinCommand[0];
    This->p_strLimit = &This->strLimit[0];
    This->p_chMotor = &This->chMotor;
    ReadString(&This->F, "Twin1Limit.txt", This->p_strLimit, LIMIT);
    Read(&This->F, "Twin1Motor.txt", This->p_chMotor);
    This->p_tOpenTO = p_OpenTO;
    EV_Twin1_Q_FlowRet = EV_Twin1_Q_Flow(This);
    if(EV_Twin1_Q_FlowRet == OK){
        if(This->strLimit[0] == 'y'){
            if(This->strLimit[4] == 'y'){ // 1閉
                This->UD = 'U';
                switch(TwinCommand[1]){
                    case 'U':
                        if(TwinCommand[0] == '1'){
                            Write(&This->F, "Twin1Motor.txt",'s');
                            // printf("STOPします¥n");
                            // WaitSecond(&This->T, 1);
                            Write(&This->F, "Twin1Motor.txt",'h');
                            // printf("開きます¥n");

```

```

        // WaitSecond(&This>T, 1);
    }
    else if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin1Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin1PermitTurnOpen.txt"¥N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin1Motor.txt",'j');
        // printf("上昇します¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'D':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin1Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin1PermitTurnOpen.txt"¥N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin1Motor.txt",'j');
        // printf("上昇します¥n");
        // WaitSecond(&This>T, 1);
    }
    else if(TwinCommand[0]== '3'){
        Write(&This>F, "Twin1Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin1PermitTurnOpen.txt"¥N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin1Motor.txt",'j');
        // printf("上昇します¥n");

```

```

        // WaitSecond(&This>T, 1);
    }
    break;
case 'y':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin1Motor.txt";s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin1PermitTurnOpen.txt"N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin1Motor.txt";j');
        // printf("上昇します¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'o':
case 'L':
case '1':
    Write(&This>F, "Twin1Motor.txt";s');
    // printf("STOPします¥n");
    // WaitSecond(&This>T, 1);
    Write(&This>F, "Twin1Motor.txt";h');
    // printf("開きます¥n");
    // WaitSecond(&This>T, 1);
    break;
case '2':
case '3':
    Write(&This>F, "Twin1Motor.txt";s');
    // printf("STOPします¥n");
    Write(&This>F, "Twin1PermitTurnOpen.txt"N');

```

```

        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin1Motor.txt",'j');
        // printf("上昇します¥n");
        // WaitSecond(&This>T, 1);
        break;
    case 'c':
        Write(&This>F, "Twin1Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin1PermitTurnOpen.txt",'N');
        // WaitSecond(&This>T, 1);
        break;
    default:
        break;
    }
}
else if(This->strLimit[7]== 'y'){ // 1開
    switch(TwinCommand[1]){
    case 'U':
        if(TwinCommand[0]== '1'){
            Write(&This>F, "Twin1Motor.txt",'s');
            // printf("STOPします¥n");
            // 現在開
            // printf("開いています¥n");
            SetCurrentTime(p_OpenTO);
            SetPermit(p_OpenTOON);
            // WaitSecond(&This>T, 1);
        }
        else if(TwinCommand[0]== '2'){

```

```

        Write(&This>F, "Twin1Motor.txt",'t');
        // printf("閉じます¥n");
        Write(&This>F, "Twin1PermitTurnOpen.txt",'o');
        SetPermit(p_OpenTOOFF);
        // WaitSecond(&This>T, 1);
    }
    break;
case 'D':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin1Motor.txt",'t');
        // printf("閉じます¥n");
        Write(&This>F, "Twin1PermitTurnOpen.txt",'o');
        SetPermit(p_OpenTOOFF);
        // WaitSecond(&This>T, 1);
    }
    else if(TwinCommand[0]== '3'){
        Write(&This>F, "Twin1Motor.txt",'t');
        // printf("閉じます¥n");
        Write(&This>F, "Twin1PermitTurnOpen.txt",'o');
        SetPermit(p_OpenTOOFF);
        // WaitSecond(&This>T, 1);
    }
    break;
case 'o':
case 'L':
case '1':
    Write(&This>F, "Twin1Motor.txt",'s');
    // printf("STOPします¥n");
    // 現在開

```



```

        // printf("開いています¥n");
        SetCurrentTime(p_OpenTO);
        SetPermit(p_OpenTOON);
        // WaitSecond(&This>T, 1);
        break;
case 'c':
case '2':
case '3':
        Write(&This>F, "Twin1Motor.txt";'t');
        // printf("閉じます¥n");
        Write(&This>F, "Twin1PermitTurnOpen.txt";'o');
        SetPermit(p_OpenTOOFF);
        // WaitSecond(&This>T, 1);
        break;
default:
        break;
}
}
else if((This->strLimit[5] == 'y') || (This->strLimit[6] == 'y')){
    switch(TwinCommand[1]){
case 'U':
        if(TwinCommand[0]== '1'){
            Write(&This>F, "Twin1Motor.txt";'o');
            // printf("OPENします¥n");
            // WaitSecond(&This>T, 1);
        }
        else{
            Write(&This>F, "Twin1Motor.txt";'c');
            // printf("CLOSEします¥n");

```

```

        Write(&This>F, "Twin1PermitTurnOpen.txt"ρ');
        // WaitSecond(&This>T, 1);
    }
    break;
case 'o':
case 'L':
case '1':
    Write(&This>F, "Twin1Motor.txt"ο');
    // printf("OPENします¥n");
    // WaitSecond(&This>T, 1);
    break;
default:
    Write(&This>F, "Twin1Motor.txt"ϒ');
    // printf("CLOSEします¥n");
    Write(&This>F, "Twin1PermitTurnOpen.txt"ρ');
    // WaitSecond(&This>T, 1);
    break;
}
}
else{
    switch(TwinCommand[1]){
    case 'U':
        if(TwinCommand[0]== '1'){
            Write(&This>F, "Twin1Motor.txt"ο');
            // printf("高速OPENします¥n");
            // WaitSecond(&This>T, 1);
        }
        else{
            Write(&This>F, "Twin1Motor.txt"ϒ');

```

```

        // printf("高速CLOSEします¥n");
        Write(&This>F, "Twin1PermitTurnOpen.txt"ρ');
        // WaitSecond(&This>T, 1);
    }
    break;
case 'o':
case 'L':
case '1':
    Write(&This>F, "Twin1Motor.txt";O');
    // printf("高速OPENします¥n");
    // WaitSecond(&This>T, 1);
    break;
default:
    Write(&This>F, "Twin1Motor.txt";C');
    // printf("高速CLOSEします¥n");
    Write(&This>F, "Twin1PermitTurnOpen.txt"ρ');
    // WaitSecond(&This>T, 1);
    break;
}
}
}
else if((This->strLimit[9] == 'u') && (This->strLimit[10] == 'd')){
    if(This->strLimit[4]== 'y'){ // 2閉
        switch(TwinCommand[1]){
        case 'U':
            if(TwinCommand[0]== '2'){
                Write(&This>F, "Twin1Motor.txt";s');
                // printf("STOPします¥n");
            }
        }
    }
}

```

```

        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin1Motor.txt",'h');
        // printf("開きます¥n");
        // WaitSecond(&This>T, 1);
    }
else if(TwinCommand[0]== '1'){
    This->UD = 'D';
    if(This->chMotor== 'D'){
        Write(&This>F, "Twin1Motor.txt",'D');
        // printf("高速DOWNします¥n");
        // WaitSecond(&This>T, 1);
    }
    else{
        Write(&This>F, "Twin1Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin1PermitTurnOpen.txt"'\N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin1Motor.txt",'k');
        // printf("下降します¥n");
        // WaitSecond(&This>T, 1);
    }
}
break;
case 'D':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin1Motor.txt",'s');
        // printf("STOPします¥n");
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin1Motor.txt",'h');

```

```

        // printf("開きます¥n");
        // WaitSecond(&This>T, 1);
    }
else if(TwinCommand[0]== '3'){
    This->UD = 'U';
    if(This->chMotor== 'U'){
        Write(&This>F, "Twin1Motor.txt",'U');
        // printf("高速UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    else{
        Write(&This>F, "Twin1Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin1PermitTurnOpen.txt",'N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin1Motor.txt",'j');
        // printf("上昇します¥n");
        // WaitSecond(&This>T, 1);
    }
}
break;
case 'y':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin1Motor.txt",'s');
        // printf("STOPします¥n");
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin1Motor.txt",'h');
        // printf("開きます¥n");
        // WaitSecond(&This>T, 1);

```

```

    }
    break;
case 'o':
case 'L':
case '2':
    Write(&This>F, "Twin1Motor.txt",'s');
    // printf("STOPします¥n");
    // WaitSecond(&This>T, 1);
    Write(&This>F, "Twin1Motor.txt",'h');
    // printf("開きます¥n");
    // WaitSecond(&This>T, 1);
    break;
case '1':
    This->UD = 'D';
    if(This->chMotor== 'D'){
        Write(&This>F, "Twin1Motor.txt",'D');
        // printf("高速DOWNします¥n");
        // WaitSecond(&This>T, 1);
    }
    else{
        Write(&This>F, "Twin1Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin1PermitTurnOpen.txt",'N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin1Motor.txt",'k');
        // printf("下降します¥n");
        // WaitSecond(&This>T, 1);
    }
    break;

```

```

case '3':
    This->UD = 'U';
    if(This->chMotor=='U'){
        Write(&This>F, "Twin1Motor.txt",'U');
        // printf("高速UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    else{
        Write(&This>F, "Twin1Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin1PermitTurnOpen.txt",'N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin1Motor.txt",'j');
        // printf("上昇します¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'c':
    Write(&This>F, "Twin1Motor.txt",'s');
    // printf("STOPします¥n");
    Write(&This>F, "Twin1PermitTurnOpen.txt",'N');
    // WaitSecond(&This>T, 1);
    break;
case 'h':
    if(TwinCommand[0]!='2'){
        Write(&This>F, "Twin1Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin1PermitTurnOpen.txt",'N');
    }
}

```

```

        // WaitSecond(&This>T, 1);
    }
    break;
default:
    break;
}
}
else if(This->strLimit[7]== 'y'){ // 2開
    switch(TwinCommand[1]){
    case 'U':
    case 'D':
        if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin1Motor.txt",'s');
            // printf("STOPします¥n");
            // 現在開
            // printf("開いています¥n");
            SetCurrentTime(p_OpenTO);
            SetPermit(p_OpenTOON);
            // WaitSecond(&This>T, 1);
        }
        else{
            Write(&This>F, "Twin1Motor.txt",'t');
            // printf("閉じます¥n");
            Write(&This>F, "Twin1PermitTurnOpen.txt",'o');
            SetPermit(p_OpenTOOFF);
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'y':

```



```

if(TwinCommand[0]== '2'){
    Write(&This>F, "Twin1Motor.txt",'s');
    // printf("STOPします¥n");
    //   現在開
    // printf("開いています¥n");
    SetCurrentTime(p_OpenTO);
    SetPermit(p_OpenTOON);
    // WaitSecond(&This>T, 1);
}
break;

case 'o':
case 'L':
case '2':
    Write(&This>F, "Twin1Motor.txt",'s');
    // printf("STOPします¥n");
    //   現在開
    // printf("開いています¥n");
    SetCurrentTime(p_OpenTO);
    SetPermit(p_OpenTOON);
    // WaitSecond(&This>T, 1);
    break;

case 'c':
case '1':
case '3':
    Write(&This>F, "Twin1Motor.txt",'t');
    // printf("閉じます¥n");
    Write(&This>F, "Twin1PermitTurnOpen.txt",'o');
    SetPermit(p_OpenTOOFF);
    // WaitSecond(&This>T, 1);

```

```

        break;
    case 'h':
        if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin1Motor.txt",'t');
            // printf("閉じます¥n");
            Write(&This>F, "Twin1PermitTurnOpen.txt"ρ');
            SetPermit(p_OpenTOOFF);
            // WaitSecond(&This>T, 1);
        }
        break;
    default:
        break;
}
}

else if((This->strLimit[5] == 'y') || (This->strLimit[6] == 'y')){
    switch(TwinCommand[1]){
    case 'U':
    case 'D':
        if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin1Motor.txt",'o');
            // printf("OPENします¥n");
            // WaitSecond(&This>T, 1);
        }
        else{
            Write(&This>F, "Twin1Motor.txt",'c');
            // printf("CLOSEします¥n");
            Write(&This>F, "Twin1PermitTurnOpen.txt"ρ');
            // WaitSecond(&This>T, 1);
        }
    }
}

```

```

        break;
    case 'y':
        if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin1Motor.txt",'o');
            // printf("OPENします¥n");
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'o':
    case 'L':
    case '2':
        Write(&This>F, "Twin1Motor.txt",'o');
        // printf("OPENします¥n");
        // WaitSecond(&This>T, 1);
        break;
    default:
        Write(&This>F, "Twin1Motor.txt",'c');
        // printf("CLOSEします¥n");
        Write(&This>F, "Twin1PermitTurnOpen.txt",'o');
        // WaitSecond(&This>T, 1);
        break;
    }
}
else{
    switch(TwinCommand[1]){
    case 'U':
    case 'D':
        if(TwinCommand[0]== '2'){

```

```

        Write(&This>F, "Twin1Motor.txt",'O');
        // printf("高速OPENします¥n");
        // WaitSecond(&This>T, 1);
    }
    else{
        Write(&This>F, "Twin1Motor.txt",'C');
        // printf("高速CLOSEします¥n");
        Write(&This>F, "Twin1PermitTurnOpen.txt",'o');
        // WaitSecond(&This>T, 1);
    }
    break;
case 'y':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin1Motor.txt",'O');
        // printf("高速OPENします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'o':
case 'L':
case '2':
    Write(&This>F, "Twin1Motor.txt",'O');
    // printf("高速OPENします¥n");
    // WaitSecond(&This>T, 1);
    break;
default:
    Write(&This>F, "Twin1Motor.txt",'C');
    // printf("高速CLOSEします¥n");
    Write(&This>F, "Twin1PermitTurnOpen.txt",'o');

```

```

        // WaitSecond(&This>T, 1);
        break;
    }
}
}
else if(This->strLimit[3] == 'y'){
    if(This->strLimit[4] == 'y'){ // 3閉
        This->UD = 'D';
        switch(TwinCommand[1]){
        case 'U':
            if(TwinCommand[0] == '1'){
                Write(&This>F, "Twin1Motor.txt",'s');
                // printf("STOPします¥n");
                Write(&This>F, "Twin1PermitTurnOpen.txt"¥N");
                // WaitSecond(&This>T, 1);
                Write(&This>F, "Twin1Motor.txt",'k');
                // printf("下降します¥n");
                // WaitSecond(&This>T, 1);
            }
            else if(TwinCommand[0] == '2'){
                Write(&This>F, "Twin1Motor.txt",'s');
                // printf("STOPします¥n");
                Write(&This>F, "Twin1PermitTurnOpen.txt"¥N");
                // WaitSecond(&This>T, 1);
                Write(&This>F, "Twin1Motor.txt",'k');
                // printf("下降します¥n");
                // WaitSecond(&This>T, 1);
            }
        }
        break;
    }
}
}
}

```

case 'D':

```
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin1Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin1PermitTurnOpen.txt"'\N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin1Motor.txt",'k');
        // printf("下降します¥n");
        // WaitSecond(&This>T, 1);
    }
    else if(TwinCommand[0]== '3'){
        Write(&This>F, "Twin1Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin1PermitTurnOpen.txt"'\N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin1Motor.txt",'h');
        // printf("開きます¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
```

case 'y':

```
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin1Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin1PermitTurnOpen.txt"'\N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin1Motor.txt",'k');
        // printf("下降します¥n");
        // WaitSecond(&This>T, 1);
```

```

    }
    break;
case 'o':
case 'L':
case '3':
    Write(&This>F, "Twin1Motor.txt",'s');
    // printf("STOPします¥n");
    Write(&This>F, "Twin1PermitTurnOpen.txt"'\N');
    // WaitSecond(&This>T, 1);
    Write(&This>F, "Twin1Motor.txt",'h');
    // printf("開きます¥n");
    // WaitSecond(&This>T, 1);
    break;
case '1':
case '2':
    Write(&This>F, "Twin1Motor.txt",'s');
    // printf("STOPします¥n");
    Write(&This>F, "Twin1PermitTurnOpen.txt"'\N');
    // WaitSecond(&This>T, 1);
    Write(&This>F, "Twin1Motor.txt",'k');
    // printf("下降します¥n");
    // WaitSecond(&This>T, 1);
    break;
case 'c':
    Write(&This>F, "Twin1Motor.txt",'s');
    // printf("STOPします¥n");
    Write(&This>F, "Twin1PermitTurnOpen.txt"'\rho');
    // WaitSecond(&This>T, 1);

```

```

        break;
    default:
        break;
    }
}
else if(This->strLimit[7]== 'y'){ // 3開
    switch(TwinCommand[1]){
    case 'U':
        if(TwinCommand[0]== '1'){
            Write(&This>F, "Twin1Motor.txt",'t');
            // printf("閉じます¥n");
            Write(&This>F, "Twin1PermitTurnOpen.txt"ρ');
            SetPermit(p_OpenTOOFF);
            // WaitSecond(&This>T, 1);
        }
        else if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin1Motor.txt",'t');
            // printf("閉じます¥n");
            Write(&This>F, "Twin1PermitTurnOpen.txt"ρ');
            SetPermit(p_OpenTOOFF);
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'D':
        if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin1Motor.txt",'t');
            // printf("閉じます¥n");
            Write(&This>F, "Twin1PermitTurnOpen.txt"ρ');
            SetPermit(p_OpenTOOFF);

```



```

        // WaitSecond(&This>T, 1);
    }

    else if(TwinCommand[0]== '3'){
        Write(&This>F, "Twin1Motor.txt",'s');
        // printf("STOPします¥n");
        //   現在開
        // printf("開いています¥n");
        SetCurrentTime(p_OpenTO);
        SetPermit(p_OpenTOON);
        // WaitSecond(&This>T, 1);
    }

    break;

case 'o':
case 'L':
case '3':
    Write(&This>F, "Twin1Motor.txt",'s');
    // printf("STOPします¥n");
    //   現在開
    // printf("開いています¥n");
    SetCurrentTime(p_OpenTO);
    SetPermit(p_OpenTOON);
    // WaitSecond(&This>T, 1);
    break;

case 'c':
case '1':
case '2':
    Write(&This>F, "Twin1Motor.txt",'t');
    // printf("閉じます¥n");
    Write(&This>F, "Twin1PermitTurnOpen.txt",'p');

```

```

        SetPermit(p_OpenTOOFF);
        // WaitSecond(&This>T, 1);
        break;
default:
        break;
}
}
else if((This->strLimit[5] == 'y') || (This->strLimit[6] == 'y')){
    switch(TwinCommand[1]){
    case 'D':
        if(TwinCommand[0] == '3'){
            Write(&This>F, "Twin1Motor.txt",'o');
            // printf("OPENします¥n");
            // WaitSecond(&This>T, 1);
        }
        else{
            Write(&This>F, "Twin1Motor.txt",'c');
            // printf("CLOSEします¥n");
            Write(&This>F, "Twin1PermitTurnOpen.txt",'o');
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'o':
    case 'L':
    case '3':
        Write(&This>F, "Twin1Motor.txt",'o');
        // printf("OPENします¥n");
        // WaitSecond(&This>T, 1);
        break;

```

```

default:
    Write(&This>F, "Twin1Motor.txt";c');
    // printf("CLOSEします¥n");
    Write(&This>F, "Twin1PermitTurnOpen.txt";o');
    // WaitSecond(&This>T, 1);
    break;
}
}
else{
    switch(TwinCommand[1]){
    case 'D':
        if(TwinCommand[0]≠= '3'){
            Write(&This>F, "Twin1Motor.txt";O');
            // printf("高速OPENします¥n");
            // WaitSecond(&This>T, 1);
        }
        else{
            Write(&This>F, "Twin1Motor.txt";C');
            // printf("高速CLOSEします¥n");
            Write(&This>F, "Twin1PermitTurnOpen.txt";o');
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'o':
    case 'L':
    case '3':
        Write(&This>F, "Twin1Motor.txt";O');
        // printf("高速OPENします¥n");

```

```

        // WaitSecond(&This>T, 1);
        break;
default:
    Write(&This>F, "Twin1Motor.txt";'C');
    // printf("高速CLOSEします¥n");
    Write(&This>F, "Twin1PermitTurnOpen.txt";'o');
    // WaitSecond(&This>T, 1);
    break;
    }
}
}
else if(This->strLimit[1] == 'y'){
    switch(TwinCommand[1]){
    case 'U':
        if(TwinCommand[0]== '1'){
            Write(&This>F, "Twin1Motor.txt";'d');
            // printf("DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
        else if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin1Motor.txt";'u');
            // printf("UPします¥n");
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'D':
        if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin1Motor.txt";'u');
            // printf("UPします¥n");

```

```

        // WaitSecond(&This>T, 1);
    }
    else if(TwinCommand[0]== '3'){
        Write(&This>F, "Twin1Motor.txt",'u');
        // printf("UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'y':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin1Motor.txt",'u');
        // printf("UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case '1':
    Write(&This>F, "Twin1Motor.txt",'d');
    // printf("DOWNします¥n");
    // WaitSecond(&This>T, 1);
    break;
case '2':
case '3':
    Write(&This>F, "Twin1Motor.txt",'u');
    // printf("UPします¥n");
    // WaitSecond(&This>T, 1);
    break;
default:
    break;
}

```

```

}

else if(This->strLimit[8] == 'd'){
    switch(TwinCommand[1]){
    case 'U':
        if(TwinCommand[0] == '1'){
            Write(&This>F, "Twin1Motor.txt";'D');
            // printf("高速DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
        else if(TwinCommand[0] == '2'){
            Write(&This>F, "Twin1Motor.txt";'U');
            // printf("高速UPします¥n");
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'D':
        if(TwinCommand[0] == '2'){
            Write(&This>F, "Twin1Motor.txt";'U');
            // printf("高速UPします¥n");
            // WaitSecond(&This>T, 1);
        }
        else if(TwinCommand[0] == '3'){
            Write(&This>F, "Twin1Motor.txt";'U');
            // printf("高速UPします¥n");
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'y':
        if(TwinCommand[0] == '2'){

```

```

        Write(&This>F, "Twin1Motor.txt";'U');
        // printf("高速UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case '1':
    Write(&This>F, "Twin1Motor.txt";'D');
    // printf("高速DOWNします¥n");
    // WaitSecond(&This>T, 1);
    break;
case '2':
case '3':
    Write(&This>F, "Twin1Motor.txt";'U');
    // printf("高速UPします¥n");
    // WaitSecond(&This>T, 1);
    break;
default:
    break;
}
}
else if(This->strLimit[9] == 'd'){
    switch(TwinCommand[1]){
    case 'U':
        if(TwinCommand[0]== '1'){
            Write(&This>F, "Twin1Motor.txt";'d');
            // printf("DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
    }
}

```

```

else if(TwinCommand[0]== '2'){
    Write(&This>F, "Twin1Motor.txt",'u');
    // printf("UPします¥n");
    // WaitSecond(&This>T, 1);
}
break;
case 'D':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin1Motor.txt",'u');
        // printf("UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    else if(TwinCommand[0]== '3'){
        Write(&This>F, "Twin1Motor.txt",'U');
        // printf("高速UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'y':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin1Motor.txt",'u');
        // printf("UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case '1':
    Write(&This>F, "Twin1Motor.txt",'d');
    // printf("DOWNします¥n");
    // WaitSecond(&This>T, 1);

```



```

        break;
    case '2':
        Write(&This>F, "Twin1Motor.txt",'u');
        // printf("UPします¥n");
        // WaitSecond(&This>T, 1);
        break;
    case '3':
        Write(&This>F, "Twin1Motor.txt",'U');
        // printf("高速UPします¥n");
        // WaitSecond(&This>T, 1);
        break;
    default:
        break;
}
}
else if((This->strLimit[10] == 'u') && (This->strLimit[11] == 'd')){
    switch(TwinCommand[1]){
    case 'U':
        if(TwinCommand[0]== '1'){
            Write(&This>F, "Twin1Motor.txt",'D');
            // printf("高速DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
        else if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin1Motor.txt",'d');
            // printf("DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
        break;

```

```

case 'D':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin1Motor.txt",'d');
        // printf("DOWNします¥n");
        // WaitSecond(&This>T, 1);
    }
    else if(TwinCommand[0]== '3'){
        Write(&This>F, "Twin1Motor.txt",'u');
        // printf("UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'y':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin1Motor.txt",'d');
        // printf("DOWNします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case '1':
    Write(&This>F, "Twin1Motor.txt",'D');
    // printf("高速DOWNします¥n");
    // WaitSecond(&This>T, 1);
    break;
case '2':
    Write(&This>F, "Twin1Motor.txt",'d');
    // printf("DOWNします¥n");
    // WaitSecond(&This>T, 1);
    break;

```

```

case '3':
    Write(&This>F, "Twin1Motor.txt",'u');
    // printf("UPします¥n");
    // WaitSecond(&This>T, 1);
    break;
default:
    break;
}
}
else if((This->strLimit[11] == 'u') && (This->strLimit[2] == 'n')){
    switch(TwinCommand[1]){
case 'U':
    if(TwinCommand[0]== '1'){
        Write(&This>F, "Twin1Motor.txt",'D');
        // printf("高速DOWNします¥n");
        // WaitSecond(&This>T, 1);
    }
    else if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin1Motor.txt",'D');
        // printf("高速DOWNします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'D':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin1Motor.txt",'D');
        // printf("高速DOWNします¥n");
        // WaitSecond(&This>T, 1);
    }
}
}
}
}

```

```

    }

    else if(TwinCommand[0]== '3'){

        Write(&This>F, "Twin1Motor.txt",'U');

        // printf("高速UPします¥n");

        // WaitSecond(&This>T, 1);

    }

    break;

case 'y':

    if(TwinCommand[0]== '2'){

        Write(&This>F, "Twin1Motor.txt",'D');

        // printf("高速DOWNします¥n");

        // WaitSecond(&This>T, 1);

    }

    break;

case '1':

case '2':

    Write(&This>F, "Twin1Motor.txt",'D');

    // printf("高速DOWNします¥n");

    // WaitSecond(&This>T, 1);

    break;

case '3':

    Write(&This>F, "Twin1Motor.txt",'U');

    // printf("高速UPします¥n");

    // WaitSecond(&This>T, 1);

    break;

default:

    break;

}

}

```

```

else if(This->strLimit[2] == 'y'){
    switch(TwinCommand[1]){
    case 'U':
        if(TwinCommand[0]== '1'){
            Write(&This>F, "Twin1Motor.txt",'d');
            // printf("DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
        else if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin1Motor.txt",'d');
            // printf("DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'D':
        if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin1Motor.txt",'d');
            // printf("DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
        else if(TwinCommand[0]== '3'){
            Write(&This>F, "Twin1Motor.txt",'u');
            // printf("UPします¥n");
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'y':
        if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin1Motor.txt",'d');

```

```

        // printf("DOWNします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case '1':
case '2':
    Write(&This>F, "Twin1Motor.txt",'d');
    // printf("DOWNします¥n");
    // WaitSecond(&This>T, 1);
    break;
case '3':
    Write(&This>F, "Twin1Motor.txt",'u');
    // printf("UPします¥n");
    // WaitSecond(&This>T, 1);
    break;
default:
    break;
}
}
return;
}

void EV_Twin1_Q_OTOUDChenge(struct EV_Twin_Queue *This, struct EV_Time *p_OpenTO, Thread
*th)
{
    char *p_cmd;
    char cmd;
    p_cmd = &cmd;

```

```

cmd = 'N';
This->p_tOpenTO = p_OpenTO;
This->p_strLimit = &This->strLimit[0];
ReadString(&This->F, "Twin1Limit.txt", This->p_strLimit, LIMIT);
if((This->UD == 'U') && (This->p_strLimit[9] == 'u') && (This->p_strLimit[10] == 'd'))
{
    // 0L,0o,02,2U,2y,0c,2h,03,3D
    if(EV_Twin1_Q_Check(This,
"0L", "0o", "02", "2U", "2y", "NN", "NN", "NN", "NN", "NN", "NN", "NN") == OK)
    {
        WriteString(&This->F, "Twin1Command.txt", "0o");
        EV_Twin1_Q_Command_Read(This, p_cmd, th);
        EV_Twin1_Q_CheckTurnOpen(This);
        SetCurrentTime(This->p_tOpenTO);
        SetPermit(This->p_tOpenTO, ON);
    }
    else if(EV_Twin1_Q_Check(This,
"0c", "2h", "03", "3D", "NN", "NN", "NN", "NN", "NN", "NN", "NN", "NN") ==
ONE_MORE_TIME)
    {
        if(EV_Twin1_Q_Check(This,
"2D", "01", "1U", "NN", "NN", "NN", "NN", "NN", "NN", "NN", "NN") == OK)
        {
            WriteString(&This->F, "Twin1Command.txt", "0c");
            EV_Twin1_Q_Command_Read(This, p_cmd, th);
            EV_Twin1_Q_Motor(This->p_OpenTO);
            This->UD = 'D';
            EV_Twin1_Q_Write(This, "Twin1Command.txt", "2U", 'r');

```

```

    }

    else
    {
        WriteString(&This->F, "Twin1Command.txt", "0c¥0");
        EV_Twin1_Q_Command_Read(This, p_cmd, th);
        EV_Twin1_Q_Motor(Thisp_OpenTO);
    }
}

else
{
    WriteString(&This->F, "Twin1Command.txt", "0c¥0");
    EV_Twin1_Q_Command_Read(This, p_cmd, th);
    EV_Twin1_Q_Motor(This, p_OpenTO);
}
}

else if((This->UD == 'D') && (This->p_strLimit[9] == 'u') && (This->p_strLimit[10] == 'd'))
{
    // 0L,0o,02,2D,2y,0c,2h,01,1U
    if(EV_Twin1_Q_Check(This,
"0L¥0", "0o¥0", "02¥0", "2D¥0", "2y¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0") == OK)
    {
        WriteString(&This->F, "Twin1Command.txt", "0o¥0");
        EV_Twin1_Q_Command_Read(This, p_cmd, th);
        EV_Twin1_Q_CheckTurnOpen(This);
        SetCurrentTime(This->p_tOpenTO);
        SetPermit(This->p_tOpenTO, ON);
    }

    else if(EV_Twin1_Q_Check(This,
"0c¥0", "2h¥0", "01¥0", "1U¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0") ==

```


ONE_MORE_TIME)

{

if(EV_Twin1_Q_Check(This,

"2U¥0","03¥0","3D¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0") == OK)

{

WriteString(&This->F, "Twin1Command.txt", "0c¥0");

EV_Twin1_Q_Command_Read(This, p_cmd, th);

EV_Twin1_Q_Motor(Thisp_OpenTO);

This->UD = 'U';

EV_Twin1_Q_Write(This, "Twin1Command.txt", "2D¥0", 'r');

}

else

{

WriteString(&This->F, "Twin1Command.txt", "0c¥0");

EV_Twin1_Q_Command_Read(This, p_cmd, th);

EV_Twin1_Q_Motor(Thisp_OpenTO);

}

}

else

{

WriteString(&This->F, "Twin1Command.txt", "0c¥0");

EV_Twin1_Q_Command_Read(This, p_cmd, th);

EV_Twin1_Q_Motor(This, p_OpenTO);

}

}

else

{

WriteString(&This->F, "Twin1Command.txt", "0c¥0");

EV_Twin1_Q_Command_Read(This, p_cmd, th);

```
EV_Twin1_Q_Motor(This, p_OpenTO);
```

```
}
```

```
return;
```

```
}
```

```

// EV_Twin2_Queue.c

#include "C.h"
#include "EV_Twin_Queue.h"

void EV_Twin2_Q_Init(struct EV_Twin_Queue *This, Thread *th){
    int i;
    EV_Time(&This->T, th);
    EV_File(&This->F);
    This->p_strQueue = &This->strQueue[0];
    for(i = 0; i < 34; i++){
        This->strQueue[i]= 'N';
    }
    This->strQueue[34] = ' ';
    // uYdyocHhqe0o0c0102031U2D2U3D2y2h0L 
    // NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN 
    This->UD = 'U';
    return;
}

void EV_Twin2_Q_Write(struct EV_Twin_Queue *This, char *filename, char *p_str, char mode)
{
    This->p_strLimit = &This->strLimit[0];
    ReadString(&This->F, "Twin2Limit.txt", This->p_strLimit, LIMIT);
    This->p_strQueue = &This->strQueue[0];
    if(filename[0] == 'C'){ // Command.txt
        switch(p_str[0]){
            case 'u':
                if(mode == 'w'){

```

```
        This->strQueue[0]= 'u';
    }
    else if(mode== 'r'){
        This->strQueue[0]= 'N';
    }
    break;
case 'Y':
    if(mode== 'w'){
        This->strQueue[1]= 'Y';
    }
    else if(mode== 'r'){
        This->strQueue[1]= 'N';
    }
    break;
case 'd':
    if(mode== 'w'){
        This->strQueue[2]= 'd';
    }
    else if(mode== 'r'){
        This->strQueue[2]= 'N';
    }
    break;
case 'y':
    if(mode== 'w'){
        This->strQueue[3]= 'y';
    }
    else if(mode== 'r'){
        This->strQueue[3]= 'N';
    }
}
```

```
        break;
case 'o':
    if(mode == 'w'){
        This->strQueue[4] = 'o';
    }
    else if(mode == 'r'){
        This->strQueue[4] = 'N';
    }
    break;
case 'c':
    if(mode == 'w'){
        This->strQueue[5] = 'c';
    }
    else if(mode == 'r'){
        This->strQueue[5] = 'N';
    }
    break;
case 'H':
    if(mode == 'w'){
        This->strQueue[6] = 'H';
    }
    else if(mode == 'r'){
        This->strQueue[6] = 'N';
    }
    break;
case 'h':
    if(mode == 'w'){
        This->strQueue[7] = 'h';
    }
}
```

```

        else if(mode == 'r'){
            This->strQueue[7] = 'N';
        }
        break;
    case 'q':
        if(mode == 'w'){
            This->strQueue[8] = 'q';
        }
        else if(mode == 'r'){
            This->strQueue[8] = 'N';
        }
        break;
    case 'e':
        if(mode == 'w'){
            This->strQueue[9] = 'e';
        }
        else if(mode == 'r'){
            This->strQueue[9] = 'N';
        }
        break;
    default:
        break;
}
}

else if((filename[0] == 'T') && (filename[4] == '2')){ // Twin2Command.txt
    switch(p_str[1]){
    case 'o':
        if(mode == 'w'){
            This->strQueue[10] = '0';

```

```

        This->strQueue[11]= 'o';
    }
    else if(mode== 'r'){
        This->strQueue[10]= 'N';
        This->strQueue[11]= 'N';
    }
    break;
case 'c':
    if(mode== 'w'){
        This->strQueue[12]= '0';
        This->strQueue[13]= 'c';
    }
    else if(mode== 'r'){
        This->strQueue[12]= 'N';
        This->strQueue[13]= 'N';
    }
    break;
case '1':
    if(mode== 'w'){
        if((This->strQueue[14] == '0') && (This->strQueue[15] == '1')){
            This->strQueue[14]= 'N';
            This->strQueue[15]= 'N';
            // printf("1階行きキャンセル\n");
        }
        else{
            This->strQueue[14]= '0';
            This->strQueue[15]= '1';
            // printf("1階行きON\n");
        }
    }

```

```

    }
}
else if(mode == 'r'){
    This->strQueue[14] = 'N';
    This->strQueue[15] = 'N';
}
break;
case '2':
    if(mode == 'w'){
        if((This->strQueue[16] == '0') && (This->strQueue[17] == '2')){
            This->strQueue[16] = 'N';
            This->strQueue[17] = 'N';
            // printf("2階行きキャンセル\n");
        }
        else{
            This->strQueue[16] = '0';
            This->strQueue[17] = '2';
            // printf("2階行きON\n");
        }
    }
    else if(mode == 'r'){
        This->strQueue[16] = 'N';
        This->strQueue[17] = 'N';
    }
    break;
case '3':
    if(mode == 'w'){
        if((This->strQueue[18] == '0') && (This->strQueue[19] == '3')){

```



```

        This->strQueue[18]= 'N';
        This->strQueue[19]= 'N';
        // printf("3階行きキャンセル¥n");
    }
    else{
        This->strQueue[18]= '0';
        This->strQueue[19]= '3';
        // printf("3階行きON¥n");
    }
}
else if(mode== 'r'){
    This->strQueue[18]= 'N';
    This->strQueue[19]= 'N';
}
break;
case 'U':
    if(mode== 'w'){
        if(p_str[0]== '1'){
            if((This->strQueue[20]== '1') && (This->strQueue[21]== 'U')){
                This->strQueue[20]= 'N';
                This->strQueue[21]= 'N';
                // printf("1階から上行きキャンセル¥n");
            }
            else{
                This->strQueue[20]= '1';
                This->strQueue[21]= 'U';
                // printf("1階から上行きON¥n");
            }
        }
    }
}

```

```

else if(p_str[0]== '2'){
    if((This->strQueue[24]== '2') && (This->strQueue[25]== 'U')){
        This->strQueue[24]= 'N';
        This->strQueue[25]= 'N';
        // printf("2階から上行きキャンセル¥n");
    }
    else{
        This->strQueue[24]= '2';
        This->strQueue[25]= 'U';
        // printf("2階から上行きON¥n");
    }
}
}

else if(mode== 'r'){
    if(p_str[0]== '1'){
        This->strQueue[20]= 'N';
        This->strQueue[21]= 'N';
    }
    else if(p_str[0]== '2'){
        This->strQueue[24]= 'N';
        This->strQueue[25]= 'N';
    }
}

break;

case 'D':
    if(mode== 'w'){
        if(p_str[0]== '2'){
            if((This->strQueue[22]== '2') && (This->strQueue[23]== 'D')){
                This->strQueue[22]= 'N';
            }
        }
    }
}

```

```

        This->strQueue[23]= 'N';
        // printf("2階から下行きキャンセル¥n");
    }
    else{
        This->strQueue[22]= '2';
        This->strQueue[23]= 'D';
        // printf("2階から下行きON¥n");
    }
}
else if(p_str[0]== '3'){
    if((This->strQueue[26]== '3') && (This->strQueue[27]== 'D')){
        This->strQueue[26]= 'N';
        This->strQueue[27]= 'N';
        // printf("3階から下行きキャンセル¥n");
    }
    else{
        This->strQueue[26]= '3';
        This->strQueue[27]= 'D';
        // printf("3階から下行きON¥n");
    }
    This->strQueue[26]= '3';
    This->strQueue[27]= 'D';
}
}
else if(mode== 'r'){
    if(p_str[0]== '2'){
        This->strQueue[22]= 'N';
        This->strQueue[23]= 'N';
    }
}

```

```

    }
    else if(p_str[0]== '3'){
        This->strQueue[26]= 'N';
        This->strQueue[27]= 'N';
    }
}
break;
case 'y':
    if(mode== 'w'){
        if(p_str[0]== '2'){
            This->strQueue[28]= '2';
            This->strQueue[29]= 'y';
        }
    }
    else if(mode== 'r'){
        if(p_str[0]== '2'){
            This->strQueue[28]= 'N';
            This->strQueue[29]= 'N';
        }
    }
    break;
case 'h':
    if(mode== 'w'){
        if(p_str[0]== '2'){
            This->strQueue[30]= '2';
            This->strQueue[31]= 'h';
        }
    }
    else if(mode== 'r'){

```

```

        if(p_str[0]== '2'){
            This->strQueue[30]= 'N';
            This->strQueue[31]= 'N';
        }
    }
    break;
case 'L':
    if(mode== 'w'){
        if((This->strQueue[32] == '0') && (This->strQueue[33] == 'L')){
            This->strQueue[32]= 'N';
            This->strQueue[33]= 'N';
            // printf("開延長キャンセル¥n");
        }
        else if((This->p_strLimit[6] == 'y') && (This->p_strLimit[7] == 'y'))
        {
            This->strQueue[32]= '0';
            This->strQueue[33]= 'L';
            // printf("開延長ON¥n");
        }
    }
    else if(mode== 'r'){
        This->strQueue[32]= 'N';
        This->strQueue[33]= 'N';
    }
    break;
default:
    break;
}
}
}

```

```
if(This->strQueue[0] == 'N' &&  
    This->strQueue[1] == 'N' &&  
    This->strQueue[2] == 'N' &&  
    This->strQueue[3] == 'N' &&  
    This->strQueue[4] == 'N' &&  
    This->strQueue[5] == 'N' &&  
    This->strQueue[6] == 'N' &&  
    This->strQueue[7] == 'N' &&  
    This->strQueue[8] == 'N' &&  
    This->strQueue[9] == 'N' &&  
    This->strQueue[10] == 'N' &&  
    This->strQueue[11] == 'N' &&  
    This->strQueue[12] == 'N' &&  
    This->strQueue[13] == 'N' &&  
    This->strQueue[14] == 'N' &&  
    This->strQueue[15] == 'N' &&  
    This->strQueue[16] == 'N' &&  
    This->strQueue[17] == 'N' &&  
    This->strQueue[18] == 'N' &&  
    This->strQueue[19] == 'N' &&  
    This->strQueue[20] == 'N' &&  
    This->strQueue[21] == 'N' &&  
    This->strQueue[22] == 'N' &&  
    This->strQueue[23] == 'N' &&  
    This->strQueue[24] == 'N' &&  
    This->strQueue[25] == 'N' &&  
    This->strQueue[26] == 'N' &&  
    This->strQueue[27] == 'N' &&  
    This->strQueue[28] == 'N' &&
```

```

    This->strQueue[29] == 'N' &&
    This->strQueue[30] == 'N' &&
    This->strQueue[31] == 'N' &&
    This->strQueue[32] == 'N' &&
    This->strQueue[33] == 'N')
{
    if((This->p_strLimit[6] == 'y') && (This->p_strLimit[7] == 'y'))
    {
        Write(&This->F, "Twin2Motor.txt", 's');
    }
    else if((This->strQueue[32] == '0') && (This->strQueue[33] == 'L')){
        This->strQueue[32]= 'N';
        This->strQueue[33]= 'N';
    }
}
return;
}

int EV_Twin2_Q_Command_Read(struct EV_Twin_Queue *This, char *p_Command, Thread *th){
    char ch;
    char *p_ch;
    char str[3];
    char *p_str;

    This->p_strQueue = &This->strQueue[0];
    ch = 'N';
    p_ch = &ch;
    str[0] = 'N';
    str[1] = 'N';

```

```

str[2] = '¥n';

p_str = &str[0];

ReadString(&This->F, "Twin2Command.txt", p_str, 3);

// write queue
EV_Twin2_Q_Write(This, "Twin2Command.txt", p_str, 'w');

// remove TwinCommand.txt
WriteString(&This->F, "Twin2Command.txt", "NN¥0");

*p_Command = ch;
if((ch == 'u') || (ch == 'Y') || (ch == 'd') || (ch == 'y')
|| (ch == 'o') || (ch == 'c') || (ch == 'H') || (ch == 'h')
|| (ch == 'e')){
    // 命令入力許可終了
    //Write(&This->F, "Twin2PermitCommand.txt", 'N');

// 命令信号保持
    // WaitSecond(&This->T, 1);

    // 消命令
    // Write(&This->F, "Command.txt", 'N');

// remove queue
EV_Twin2_Q_Init(This, th);

return OK;
}

```



```
else if(ch == 'q'){
    // 命令入力許可終了
    Write(&This->F, "Twin2PermitCommand.txt", 'N');
```

```
// 命令信号保持
```

```
    // WaitSecond(&This->T, 3);
```

```
    // 消命令
```

```
    // Write(&This->F, "Command.txt", 'N');
```

```
    // WriteString(&This->F, "Twin2Command.txt", "NN¥0");
```

```
    // remove queue
```

```
    EV_Twin2_Q_Init(This, th);
```

```
    return OK;
```

```
}
```

```
return ONE_MORE_TIME;
```

```
}
```

```
int EV_Twin2_Q_CheckTurnOpen(struct EV_Twin_Queue *This){
```

```
    int Ret;
```

```
    char strTurnOpen[3];
```

```
    char *p_strTurnOpen;
```

```
    char PermitTurnOpen;
```

```
    strTurnOpen[0] = 'N';
```

```
    strTurnOpen[0] = 'N';
```

```
    strTurnOpen[0] = '¥0';
```

```
    PermitTurnOpen = 'N';
```

```
    p_strTurnOpen = &strTurnOpen[0];
```

```
    ReadString(&This->F, "Twin2TurnOpen.txt", p_strTurnOpen, 3);
```

```

// Wait_ms(&This->T, 200);

Read(&This->F, "Twin2PermitTurnOpen.txt", &PermitTurnOpen);

// Wait_ms(&This->T, 200);

if(PermitTurnOpen == 'N'){
    Ret = ONE_MORE_TIME;
}

else{
    // 閉中斷時
    if(p_strTurnOpen[0] == 'N'){
        Ret = ONE_MORE_TIME;
    }
    else if((p_strTurnOpen[0] == '0') && (p_strTurnOpen[1] == 'o')){
        Write(&This->F, "Twin2Motor.txt", 'h');
        EV_Twin2_Q_Write(This, "Twin2Command.txt", "0c¥0", 'r');
        This->p_TwinCommand[0]= '0';
        This->p_TwinCommand[1]= 'o';
        // printf("反転開します¥n");
        WriteString(&This->F, "Twin2Command.txt", "0o¥0");
        Write(&This->F, "Twin2PermitCommand.txt", 'c');
        WriteString(&This->F, "Twin2TurnOpen.txt", "NN¥0");
        Write(&This->F, "Twin2PermitTurnOpen.txt", 'N');
        // WaitSecond(&This>T, 1);
        Ret = NG;
    }
    else if((p_strTurnOpen[0] == '0') && (p_strTurnOpen[1] == 'L')){
        Write(&This->F, "Twin2Motor.txt", 'h');
        EV_Twin2_Q_Write(This, "Twin2Command.txt", "0c¥0", 'r');
        This->p_TwinCommand[0]= '0';
        This->p_TwinCommand[1]= 'o';
    }
}

```

```

// printf("反転開します¥n");
WriteString(&This->F, "Twin2Command.txt", "0L¥0");
Write(&This->F, "Twin2PermitCommand.txt", 'c');
WriteString(&This->F, "Twin2TurnOpen.txt", "NN¥0");
Write(&This->F, "Twin2PermitTurnOpen.txt", 'N');
// WaitSecond(&This>T, 1);
Ret = NG;
}
else if((This->p_strLimit[0] == 'y') && ((p_strTurnOpen[0] == '1') || (p_strTurnOpen[1] ==
'1'))){
Write(&This->F, "Twin2Motor.txt", 'h');
EV_Twin2_Q_Write(This, "Twin2Command.txt", "0c¥0", 'r');
This->p_TwinCommand[0]= '0';
This->p_TwinCommand[1]= 'o';
// printf("反転開します¥n");
WriteString(&This->F, "Twin2Command.txt", "0o¥0");
Write(&This->F, "Twin2PermitCommand.txt", 'c');
WriteString(&This->F, "Twin2TurnOpen.txt", "NN¥0");
Write(&This->F, "Twin2PermitTurnOpen.txt", 'N');
// WaitSecond(&This>T, 1);
Ret = NG;
}
else if((This->p_strLimit[9] == 'u') && (This->p_strLimit[10] == 'd') && ((p_strTurnOpen[0] ==
'2') || (p_strTurnOpen[1] == '2'))){
Write(&This->F, "Twin2Motor.txt", 'h');
EV_Twin2_Q_Write(This, "Twin2Command.txt", "0c¥0", 'r');
EV_Twin2_Q_Write(This, "Twin2Command.txt", "2h¥0", 'r');
This->p_TwinCommand[0]= '0';

```

```

This->p_TwinCommand[1]= 'o';
// printf("反転開します¥n");
WriteString(&This->F, "Twin2Command.txt", "0o¥0");
Write(&This->F, "Twin2PermitCommand.txt", 'c');
WriteString(&This->F, "Twin2TurnOpen.txt", "NN¥0");
Write(&This->F, "Twin2PermitTurnOpen.txt", 'N');
// WaitSecond(&This>T, 1);

Ret = NG;
}
else if((This->p_strLimit[3] == 'y') && ((p_strTurnOpen[0] == '3') || (p_strTurnOpen[1] ==
'3'))){

Write(&This->F, "Twin2Motor.txt", 'h');
EV_Twin2_Q_Write(This, "Twin2Command.txt", "0c¥0", 'r');
This->p_TwinCommand[0]= '0';
This->p_TwinCommand[1]= 'o';
// printf("反転開します¥n");
WriteString(&This->F, "Twin2Command.txt", "0o¥0");
Write(&This->F, "Twin2PermitCommand.txt", 'c');
WriteString(&This->F, "Twin2TurnOpen.txt", "NN¥0");
Write(&This->F, "Twin2PermitTurnOpen.txt", 'N');
// WaitSecond(&This>T, 1);

Ret = NG;
}
else{
Ret = ONE_MORE_TIME;
}
}
return Ret;
}

```

```

int EV_Twin2_Q_Read(struct EV_Twin_Queue *This, char *p_check){
    int Ret;
    This->p_strQueue = &This->strQueue[0];
    Ret = EV_Twin2_Q_CheckTurnOpen(This);
    if(Ret == NG) return NG;
    switch(p_check[1]){
    case 'o':
        if(This->strQueue[11] == 'o'){
            This->p_TwinCommand[0]= '0';
            This->p_TwinCommand[1]= 'o';
            Ret = OK;
        }
        else{
            This->p_TwinCommand[0]= 'N';
            This->p_TwinCommand[1]= 'N';
            Ret = ONE_MORE_TIME;
        }
        break;
    case 'c':
        if(This->strQueue[13] == 'c'){
            This->p_TwinCommand[0]= '0';
            This->p_TwinCommand[1]= 'c';
            Ret = OK;
        }
        else{
            This->p_TwinCommand[0]= 'N';
            This->p_TwinCommand[1]= 'N';
            Ret = ONE_MORE_TIME;
        }
    }
}

```

```

    break;
case '1':
    if(This->strQueue[15] == '1'){
        This->p_TwinCommand[0]= '0';
        This->p_TwinCommand[1]= '1';
        Ret = OK;
    }
    else{
        This->p_TwinCommand[0]= 'N';
        This->p_TwinCommand[1]= 'N';
        Ret = ONE_MORE_TIME;
    }
    break;
case '2':
    if(This->strQueue[17] == '2'){
        This->p_TwinCommand[0]= '0';
        This->p_TwinCommand[1]= '2';
        Ret = OK;
    }
    else{
        This->p_TwinCommand[0]= 'N';
        This->p_TwinCommand[1]= 'N';
        Ret = ONE_MORE_TIME;
    }
    break;
case '3':
    if(This->strQueue[19] == '3'){
        This->p_TwinCommand[0]= '0';
        This->p_TwinCommand[1]= '3';

```

```

        Ret = OK;
    }
    else{
        This->p_TwinCommand[0]= 'N';
        This->p_TwinCommand[1]= 'N';
        Ret = ONE_MORE_TIME;
    }
    break;
case 'U':
    if(p_check[0]== '1'){
        if(This->strQueue[21]== 'U'){
            This->p_TwinCommand[0]= '1';
            This->p_TwinCommand[1]= 'U';
            Ret = OK;
        }
        else{
            This->p_TwinCommand[0]= 'N';
            This->p_TwinCommand[1]= 'N';
            Ret = ONE_MORE_TIME;
        }
    }
    if(p_check[0]== '2'){
        if(This->strQueue[25]== 'U'){
            This->p_TwinCommand[0]= '2';
            This->p_TwinCommand[1]= 'U';
            Ret = OK;
        }
        else{

```

```

        This->p_TwinCommand[0]= 'N';
        This->p_TwinCommand[1]= 'N';
        Ret = ONE_MORE_TIME;
    }
}
break;
case 'D':
    if(p_check[0]== '2'){
        if(This->strQueue[23]== 'D'){
            This->p_TwinCommand[0]= '2';
            This->p_TwinCommand[1]= 'D';
            Ret = OK;
        }
        else{
            This->p_TwinCommand[0]= 'N';
            This->p_TwinCommand[1]= 'N';
            Ret = ONE_MORE_TIME;
        }
    }
    if(p_check[0]== '3'){
        if(This->strQueue[27]== 'D'){
            This->p_TwinCommand[0]= '3';
            This->p_TwinCommand[1]= 'D';
            Ret = OK;
        }
        else{
            This->p_TwinCommand[0]= 'N';
            This->p_TwinCommand[1]= 'N';
            Ret = ONE_MORE_TIME;
        }
    }
}

```



```

        }
    }
    break;
case 'y':
    if(p_check[0]== '2'){
        if(This->strQueue[29]== 'y'){
            This->p_TwinCommand[0]= '2';
            This->p_TwinCommand[1]= 'y';
            Ret = OK;
        }
        else{
            This->p_TwinCommand[0]= 'N';
            This->p_TwinCommand[1]= 'N';
            Ret = ONE_MORE_TIME;
        }
    }
    break;
case 'h':
    if(p_check[0]== '2'){
        if(This->strQueue[31]== 'h'){
            This->p_TwinCommand[0]= '2';
            This->p_TwinCommand[1]= 'h';
            Ret = OK;
        }
        else{
            This->p_TwinCommand[0]= 'N';
            This->p_TwinCommand[1]= 'N';
            Ret = ONE_MORE_TIME;
        }
    }

```

```

    }
    break;
case 'L':
    if(This->strQueue[33] == 'L'){
        This->p_TwinCommand[32]= '0';
        This->p_TwinCommand[33]= 'L';
        Ret = OK;
    }
    else{
        This->p_TwinCommand[32]= 'N';
        This->p_TwinCommand[33]= 'N';
        Ret = ONE_MORE_TIME;
    }
    break;
default:
    This->p_TwinCommand[0] = 'N';
    This->p_TwinCommand[1] = 'N';
    Ret = ONE_MORE_TIME;
    break;
}
return Ret;
}

int EV_Twin2_Q_Check(struct EV_Twin_Queue *This, char *c0, char *c1, char *c2, char *c3, char *c4,
char *c5, char *c6, char *c7, char *c8, char *c9, char *c10, char *c11){
    int Ret;
    Ret = EV_Twin2_Q_Read(This, c0);
    if(Ret == OK) return OK;
    if(Ret == NG) return NG;
    Ret = EV_Twin2_Q_Read(This, c1);

```

```
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Twin2_Q_Read(This, c2);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Twin2_Q_Read(This, c3);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Twin2_Q_Read(This, c4);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Twin2_Q_Read(This, c5);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Twin2_Q_Read(This, c6);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Twin2_Q_Read(This, c7);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Twin2_Q_Read(This, c8);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Twin2_Q_Read(This, c9);
if(Ret == OK) return OK;
if(Ret == NG) return NG;
Ret = EV_Twin2_Q_Read(This, c10);
if(Ret == OK) return OK;
```

```

if(Ret == NG) return NG;

Ret = EV_Twin2_Q_Read(This, c11);

return Ret;

}

int EV_Twin2_Q_Flow(struct EV_Twin_Queue *This){

    int Ret;

    This->p_strLimit = &This->strLimit[0];

    This->p_chMotor = &This->chMotor;

    ReadString(&This->F, "Twin2Limit.txt", This->p_strLimit, LIMIT);

    // Wait_ms(&This->T, 200);

    Read(&This->F, "Twin2Motor.txt", This->p_chMotor);

    // Wait_ms(&This->T, 200);

    if((This->chMotor == 'j') || (This->chMotor == 'u') || (This->chMotor == 'U')){

        //    remove 0o,0c

        EV_Twin2_Q_Write(This, "Twin2Command.txt", "0o¥0", 'r');

        EV_Twin2_Q_Write(This, "Twin2Command.txt", "0c¥0", 'r');

        if((This->strLimit[9] == 'u') && (This->strLimit[10] == 'd') && ((This->chMotor == 'j') || (This->chMotor == 'u'))){

            //    02,2y,2U,03,3D,2D,01,1U

            Ret = EV_Twin2_Q_Check(This,

"02¥0", "2y¥0", "2U¥0", "03¥0", "3D¥0", "2D¥0", "01¥0", "1U¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0");

        }

        else if(This->strLimit[9] == 'u'){

            //    03,3D,02,2D,2y,01,1U,2U

            Ret = EV_Twin2_Q_Check(This,

"03¥0", "3D¥0", "02¥0", "2D¥0", "2y¥0", "01¥0", "1U¥0", "2U¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0");

        }

        else{

            //    02,2U,2y,03,3D,2D,01,1U

```

```

        Ret = EV_Twin2_Q_Check(This,
"02¥0","2U¥0","2y¥0","03¥0","3D¥0","2D¥0","01¥0","1U¥0","NN¥0","NN¥0","NN¥0","NN¥0");
    }
}
else if((This->chMotor == 'k') || (This->chMotor == 'd') || (This->chMotor == 'D')){
    // remove 0o,0c
    EV_Twin2_Q_Write(This, "Twin2Command.txt", "0o¥0", 'r');
    EV_Twin2_Q_Write(This, "Twin2Command.txt", "0c¥0", 'r');
    if((This->strLimit[9] == 'u') && (This->strLimit[10] == 'd') && ((This->chMotor == 'k') || (This-
>chMotor == 'd'))){
        // 02,2y,2D,01,1U,2U,03,3D
        Ret = EV_Twin2_Q_Check(This,
"02¥0","2y¥0","2D¥0","01¥0","1U¥0","2U¥0","03¥0","3D¥0","NN¥0","NN¥0","NN¥0","NN¥0");
    }
    else if(This->strLimit[10] == 'd'){
        // 01,1U,02,2U,2y,03,3D,2D
        Ret = EV_Twin2_Q_Check(This,
"01¥0","1U¥0","02¥0","2U¥0","2y¥0","03¥0","3D¥0","2D¥0","NN¥0","NN¥0","NN¥0","NN¥0");
    }
    else{
        // 02,2D,2y,01,1U,2U,03,3D
        Ret = EV_Twin2_Q_Check(This,
"02¥0","2D¥0","2y¥0","01¥0","1U¥0","2U¥0","03¥0","3D¥0","NN¥0","NN¥0","NN¥0","NN¥0");
    }
}
else if((This->chMotor == 'h') || (This->chMotor == 'o') || (This->chMotor == 'O')){
    if(This->strLimit[0] == 'y'){
        // 0L,0o,01,1U,0c,02,2U,03,3D,2D
        Ret = EV_Twin2_Q_Check(This,

```

```

"0L¥0","0o¥0","01¥0","1U¥0","0c¥0","02¥0","2U¥0","03¥0","3D¥0","2D¥0","NN¥0","NN¥0");
    }
    else if((This->strLimit[9] == 'u') && (This->strLimit[10] == 'd')){
        // 0L,0o,02,2D/2U,2y,0c,01/03,1U/3D
        Ret = EV_Twin2_Q_Check(This,
"0L¥0","0o¥0","02¥0","2D¥0","2U¥0","2y¥0","0c¥0","01¥0","03¥0","1U¥0","3D¥0","NN¥0");
    }
    else if(This->strLimit[3] == 'y'){
        // 0L,0o,03,3D,0c,02,2D,01,1U,2U
        Ret = EV_Twin2_Q_Check(This,
"0L¥0","0o¥0","03¥0","3D¥0","0c¥0","02¥0","2D¥0","01¥0","1U¥0","2U¥0","NN¥0","NN¥0");
    }
}
else if((This->chMotor == 't') || (This->chMotor == 'c') || (This->chMotor == 'C')){
    if(This->strLimit[0] == 'y'){
        // 0L,0o,01,1U,0c,02,2U,03,3D,2D
        Ret = EV_Twin2_Q_Check(This,
"0L¥0","0o¥0","01¥0","1U¥0","0c¥0","02¥0","2U¥0","03¥0","3D¥0","2D¥0","NN¥0","NN¥0");
    }
    else if((This->strLimit[9] == 'u') && (This->strLimit[10] == 'd')){
        // 0L,0o,02,2D/2U,2y,0c,2h,01/03,1U/3D
        // Ret = EV_Twin2_Q_Check(This,
"0L¥0","0o¥0","02¥0","2D¥0","2U¥0","2y¥0","0c¥0","2h¥0","01¥0","03¥0","1U¥0","3D¥0");
        if(This->UD == 'U')
        {
            // 0L,0o,02,2U,2y,0c,2h,03,3D,2D,01,1U
            Ret = EV_Twin2_Q_Check(This,
"0L¥0","0o¥0","02¥0","2U¥0","2y¥0","0c¥0","2h¥0","03¥0","3D¥0","2D¥0","01¥0","1U¥0");
        }
    }
}

```

```

else if(This->UD == 'D')
{
    // 0L,0o,02,2D,2y,0c,2h,01,1U,2U,03,3D
    Ret = EV_Twin2_Q_Check(This,
"0L¥0","0o¥0","02¥0","2D¥0","2y¥0","0c¥0","2h¥0","01¥0","1U¥0","2U¥0","03¥0","3D¥0");
}
}

else if(This->strLimit[3] == 'y'){
    // 0L,0o,03,3D,0c,02,2D,01,1U,2U
    Ret = EV_Twin2_Q_Check(This,
"0L¥0","0o¥0","03¥0","3D¥0","0c¥0","02¥0","2D¥0","01¥0","1U¥0","2U¥0","NN¥0","NN¥0");
}
}

else if(This->chMotor == 's'){
    if(This->strLimit[0] == 'y'){
        if(This->strLimit[7] == 'y'){ // 1開
            // remove 0o,01,1U
            EV_Twin2_Q_Write(This, "Twin2Command.txt", "0o¥0", 'r');
            EV_Twin2_Q_Write(This, "Twin2Command.txt", "01¥0", 'r');
            EV_Twin2_Q_Write(This, "Twin2Command.txt", "1U¥0", 'r');
            // 0L,0c,02,2U,03,3D,2D
            // Ret = EV_Twin2_Q_Check(This,
"0L¥0","0c¥0","02¥0","2U¥0","03¥0","3D¥0","2D¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0");
            // 0L,0c
            Ret = EV_Twin2_Q_Check(This,
"0L¥0","0c¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0");
        }
        else if(This->strLimit[4] == 'y'){ // 1閉

```

```

// remove 0c
EV_Twin2_Q_Write(This, "Twin2Command.txt", "0c ", 'r');
// 0L,0o,01,1U,02,2U,2y,03,3D,2D
Ret = EV_Twin2_Q_Check(This,
"0L ", "0o ", "01 ", "1U ", "02 ", "2U ", "2y ", "03 ", "3D ", "2D ", "NN ", "NN ");
}
else{
// 0L,0o,01,1U,0c,02,2U,03,3D,2D
Ret = EV_Twin2_Q_Check(This,
"0L ", "0o ", "01 ", "1U ", "0c ", "02 ", "2U ", "03 ", "3D ", "2D ", "NN ", "NN ");
}
}
else if((This->strLimit[9] == 'u') && (This->strLimit[10] == 'd'))
{
if(This->strLimit[7] == 'y'){ // 2開
// remove 0o,02,2U,2D
EV_Twin2_Q_Write(This, "Twin2Command.txt", "0o ", 'r');
EV_Twin2_Q_Write(This, "Twin2Command.txt", "02 ", 'r');
if(This->UD == 'U')
{
EV_Twin2_Q_Write(This, "Twin2Command.txt", "2U ", 'r');
// 0L,0o,02,2U,2y,0c,2h,03,3D
Ret = EV_Twin2_Q_Check(This,
"0L ", "0o ", "02 ", "2U ", "2y ", "0c ", "2h ", "03 ", "3D ", "NN ", "NN ", "NN ");
if(Ret == ONE_MORE_TIME)
{
EV_Twin2_Q_Write(This, "Twin2Command.txt", "2D ", 'r');
}
}
}
}
}

```



```

else if(This->UD == 'D')
{
    EV_Twin2_Q_Write(This, "Twin2Command.txt", "2D¥0", 'r');
    // 0L,0o,02,2D,2y,0c,2h,01,1U
    Ret = EV_Twin2_Q_Check(This,
"0L¥0", "0o¥0", "02¥0", "2D¥0", "2y¥0", "0c¥0", "2h¥0", "01¥0", "1U¥0", "NN¥0", "NN¥0", "NN¥0");
    if(Ret == ONE_MORE_TIME)
    {
        EV_Twin2_Q_Write(This, "Twin2Command.txt", "2U¥0", 'r');
    }
}
EV_Twin2_Q_Write(This, "Twin2Command.txt", "2y¥0", 'r');
// 0L,0c,2h,01/03,1U/3D
// Ret = EV_Twin2_Q_Check(This,
"0L¥0", "0c¥0", "2h¥0", "01¥0", "03¥0", "1U¥0", "3D¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0");
// 0L,0c
Ret = EV_Twin2_Q_Check(This,
"0L¥0", "0c¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0");
}
else if(This->strLimit[4] == 'y'){ // 2閉
    // remove 0c,2h
    EV_Twin2_Q_Write(This, "Twin2Command.txt", "0c¥0", 'r');
    EV_Twin2_Q_Write(This, "Twin2Command.txt", "2h¥0", 'r');
    // 0L,0o,02,2D/2U,2y,01/03,1U/3D
    // Ret = EV_Twin2_Q_Check(This,
"0L¥0", "0o¥0", "02¥0", "2D¥0", "2U¥0", "2y¥0", "01¥0", "03¥0", "1U¥0", "3D¥0", "NN¥0", "NN¥0");
    if(This->UD == 'U')
    {
        // 0L,0o,02,2U,2y,03,3D,2D,01,1U

```

```

        Ret = EV_Twin2_Q_Check(This,
"0L¥0","0o¥0","02¥0","2U¥0","2y¥0","03¥0","3D¥0","2D¥0","01¥0","1U¥0","NN¥0","NN¥0");
    }
    else if(This->UD == 'D')
    {
        // 0L,0o,02,2D,2y,01,1U,2U,03,3D
        Ret = EV_Twin2_Q_Check(This,
"0L¥0","0o¥0","02¥0","2D¥0","2y¥0","01¥0","1U¥0","2U¥0","03¥0","3D¥0","NN¥0","NN¥0");
    }
}
else{
    // 0L,0o,02,2D/2U,2y,0c,2h,01/03,1U/3D
    Ret = EV_Twin2_Q_Check(This,
"0L¥0","0o¥0","02¥0","2D¥0","2U¥0","2y¥0","0c¥0","2h¥0","01¥0","03¥0","1U¥0","3D¥0");
}
}
else if(This->strLimit[3] == 'y'){
    if(This->strLimit[7] == 'y'){ // 3開
        // remove 0o,03,3D
        EV_Twin2_Q_Write(This, "Twin2Command.txt", "0o¥0", 'r');
        EV_Twin2_Q_Write(This, "Twin2Command.txt", "03¥0", 'r');
        EV_Twin2_Q_Write(This, "Twin2Command.txt", "3D¥0", 'r');
        // 0L,0c,02,2D,01,1U,2U
        // Ret = EV_Twin2_Q_Check(This,
"0L¥0","0c¥0","02¥0","2D¥0","01¥0","1U¥0","2U¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0");
        // 0L,0c
        Ret = EV_Twin2_Q_Check(This,
"0L¥0","0c¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0");
    }
}

```

```

else if(This->strLimit[4]== 'y'){ // 3閉
    // remove 0c
    EV_Twin2_Q_Write(This, "Twin2Command.txt", "0c ", 'r');
    // 0L,0o,03,3D,02,2D,2y,01,1U,2U
    Ret = EV_Twin2_Q_Check(This,
"0L ", "0o ", "03 ", "3D ", "02 ", "2D ", "2y ", "01 ", "1U ", "2U ", "NN ", "NN ");
    }
else{
    // 0L,0o,03,3D,0c,02,2D,01,1U,2U
    Ret = EV_Twin2_Q_Check(This,
"0L ", "0o ", "03 ", "3D ", "0c ", "02 ", "2D ", "01 ", "1U ", "2U ", "NN ", "NN ");
    }
}
else if(This->strLimit[9] == 'd'){
    // remove 0o,0c
    EV_Twin2_Q_Write(This, "Twin2Command.txt", "0o ", 'r');
    EV_Twin2_Q_Write(This, "Twin2Command.txt", "0c ", 'r');
    // 0L,01/02,1U/2U,2y,03,3D,2D
    Ret = EV_Twin2_Q_Check(This,
"0L ", "01 ", "02 ", "1U ", "2U ", "2y ", "03 ", "3D ", "2D ", "NN ", "NN ", "NN ");
    }
else if(This->strLimit[10] == 'u'){
    // remove 0o,0c
    EV_Twin2_Q_Write(This, "Twin2Command.txt", "0o ", 'r');
    EV_Twin2_Q_Write(This, "Twin2Command.txt", "0c ", 'r');
    // 0L,03/02,3D/2D,2y,01,1U,2U
    Ret = EV_Twin2_Q_Check(This,
"0L ", "03 ", "02 ", "3D ", "2D ", "2y ", "01 ", "1U ", "2U ", "NN ", "NN ", "NN ");

```

```

    }
}
return Ret;
}

void EV_Twin2_Q_Motor(struct EV_Twin_Queue *This, struct EV_Time *p_OpenTO){
    int EV_Twin2_Q_FlowRet;
    char TwinCommand[3];
    TwinCommand[0] = 'N';
    TwinCommand[1] = 'N';
    TwinCommand[2] = '¥0';
    This->p_TwinCommand = &TwinCommand[0];
    This->p_strLimit = &This->strLimit[0];
    This->p_chMotor = &This->chMotor;
    ReadString(&This->F, "Twin2Limit.txt", This->p_strLimit, LIMIT);
    Read(&This->F, "Twin2Motor.txt", This->p_chMotor);
    This->p_tOpenTO = p_OpenTO;
    EV_Twin2_Q_FlowRet = EV_Twin2_Q_Flow(This);
    if(EV_Twin2_Q_FlowRet == OK){
        if(This->strLimit[0] == 'y'){
            if(This->strLimit[4] == 'y'){ // 1閉
                This->UD = 'U';
                switch(TwinCommand[1]){
                    case 'U':
                        if(TwinCommand[0] == '1'){
                            Write(&This->F, "Twin2Motor.txt",'s');
                            // printf("STOPします¥n");
                            // WaitSecond(&This->T, 1);
                            Write(&This->F, "Twin2Motor.txt",'h');
                            // printf("開きます¥n");

```

```

        // WaitSecond(&This>T, 1);
    }
    else if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin2Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin2PermitTurnOpen.txt"¥N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin2Motor.txt",'j');
        // printf("上昇します¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'D':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin2Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin2PermitTurnOpen.txt"¥N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin2Motor.txt",'j');
        // printf("上昇します¥n");
        // WaitSecond(&This>T, 1);
    }
    else if(TwinCommand[0]== '3'){
        Write(&This>F, "Twin2Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin2PermitTurnOpen.txt"¥N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin2Motor.txt",'j');
        // printf("上昇します¥n");

```

```

        // WaitSecond(&This>T, 1);
    }
    break;
case 'y':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin2Motor.txt";s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin2PermitTurnOpen.txt"N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin2Motor.txt";j');
        // printf("上昇します¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'o':
case 'L':
case '1':
    Write(&This>F, "Twin2Motor.txt";s');
    // printf("STOPします¥n");
    // WaitSecond(&This>T, 1);
    Write(&This>F, "Twin2Motor.txt";h');
    // printf("開きます¥n");
    // WaitSecond(&This>T, 1);
    break;
case '2':
case '3':
    Write(&This>F, "Twin2Motor.txt";s');
    // printf("STOPします¥n");
    Write(&This>F, "Twin2PermitTurnOpen.txt"N');

```

```

    // WaitSecond(&This>T, 1);
    Write(&This>F, "Twin2Motor.txt",'j');
    // printf("上昇します¥n");
    // WaitSecond(&This>T, 1);
    break;
case 'c':
    Write(&This>F, "Twin2Motor.txt",'s');
    // printf("STOPします¥n");
    Write(&This>F, "Twin2PermitTurnOpen.txt",'N');
    // WaitSecond(&This>T, 1);
    break;
default:
    break;
}
}
else if(This->strLimit[7]== 'y'){ // 1開
    switch(TwinCommand[1]){
    case 'U':
        if(TwinCommand[0]== '1'){
            Write(&This>F, "Twin2Motor.txt",'s');
            // printf("STOPします¥n");
            // 現在開
            // printf("開いています¥n");
            SetCurrentTime(p_OpenTO);
            SetPermit(p_OpenTOON);
            // WaitSecond(&This>T, 1);
        }
        else if(TwinCommand[0]== '2'){

```

```

        Write(&This>F, "Twin2Motor.txt",'t');
        // printf("閉じます¥n");
        Write(&This>F, "Twin2PermitTurnOpen.txt",'o');
        SetPermit(p_OpenTOOFF);
        // WaitSecond(&This>T, 1);
    }
    break;
case 'D':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin2Motor.txt",'t');
        // printf("閉じます¥n");
        Write(&This>F, "Twin2PermitTurnOpen.txt",'o');
        SetPermit(p_OpenTOOFF);
        // WaitSecond(&This>T, 1);
    }
    else if(TwinCommand[0]== '3'){
        Write(&This>F, "Twin2Motor.txt",'t');
        // printf("閉じます¥n");
        Write(&This>F, "Twin2PermitTurnOpen.txt",'o');
        SetPermit(p_OpenTOOFF);
        // WaitSecond(&This>T, 1);
    }
    break;
case 'o':
case 'L':
case '1':
    Write(&This>F, "Twin2Motor.txt",'s');
    // printf("STOPします¥n");
    //   現在開

```



```

        // printf("開いています¥n");
        SetCurrentTime(p_OpenTO);
        SetPermit(p_OpenTOON);
        // WaitSecond(&This>T, 1);
        break;
case 'c':
case '2':
case '3':
        Write(&This>F, "Twin2Motor.txt";'t');
        // printf("閉じます¥n");
        Write(&This>F, "Twin2PermitTurnOpen.txt";'o');
        SetPermit(p_OpenTOOFF);
        // WaitSecond(&This>T, 1);
        break;
default:
        break;
}
}
else if((This->strLimit[5] == 'y') || (This->strLimit[6] == 'y')){
    switch(TwinCommand[1]){
case 'U':
        if(TwinCommand[0]== '1'){
            Write(&This>F, "Twin2Motor.txt";'o');
            // printf("OPENします¥n");
            // WaitSecond(&This>T, 1);
        }
        else{
            Write(&This>F, "Twin2Motor.txt";'c');
            // printf("CLOSEします¥n");

```

```

        Write(&This>F, "Twin2PermitTurnOpen.txt"ρ');
        // WaitSecond(&This>T, 1);
    }
    break;
case 'o':
case 'L':
case '1':
    Write(&This>F, "Twin2Motor.txt"ο');
    // printf("OPENします¥n");
    // WaitSecond(&This>T, 1);
    break;
default:
    Write(&This>F, "Twin2Motor.txt"ο');
    // printf("CLOSEします¥n");
    Write(&This>F, "Twin2PermitTurnOpen.txt"ρ');
    // WaitSecond(&This>T, 1);
    break;
}
}
else{
    switch(TwinCommand[1]){
    case 'U':
        if(TwinCommand[0]== '1'){
            Write(&This>F, "Twin2Motor.txt"ο');
            // printf("高速OPENします¥n");
            // WaitSecond(&This>T, 1);
        }
        else{
            Write(&This>F, "Twin2Motor.txt"ο');

```

```

        // printf("高速CLOSEします¥n");
        Write(&This>F, "Twin2PermitTurnOpen.txt"ρ');
        // WaitSecond(&This>T, 1);
    }
    break;
case 'o':
case 'L':
case '1':
    Write(&This>F, "Twin2Motor.txt";O');
    // printf("高速OPENします¥n");
    // WaitSecond(&This>T, 1);
    break;
default:
    Write(&This>F, "Twin2Motor.txt";C');
    // printf("高速CLOSEします¥n");
    Write(&This>F, "Twin2PermitTurnOpen.txt"ρ');
    // WaitSecond(&This>T, 1);
    break;
}
}
}
else if((This->strLimit[9] == 'u') && (This->strLimit[10] == 'd')){
    if(This->strLimit[4]== 'y'){ // 2閉
        switch(TwinCommand[1]){
        case 'U':
            if(TwinCommand[0]== '2'){
                Write(&This>F, "Twin2Motor.txt";s');
                // printf("STOPします¥n");
            }
        }
    }
}

```

```

        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin2Motor.txt",'h');
        // printf("開きます¥n");
        // WaitSecond(&This>T, 1);
    }
else if(TwinCommand[0]== '1'){
    This->UD = 'D';
    if(This->chMotor== 'D'){
        Write(&This>F, "Twin2Motor.txt",'D');
        // printf("高速DOWNします¥n");
        // WaitSecond(&This>T, 1);
    }
    else{
        Write(&This>F, "Twin2Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin2PermitTurnOpen.txt" 'N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin2Motor.txt",'k');
        // printf("下降します¥n");
        // WaitSecond(&This>T, 1);
    }
}
break;
case 'D':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin2Motor.txt",'s');
        // printf("STOPします¥n");
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin2Motor.txt",'h');

```

```

        // printf("開きます¥n");
        // WaitSecond(&This>T, 1);
    }
else if(TwinCommand[0]== '3'){
    This->UD = 'U';
    if(This->chMotor== 'U'){
        Write(&This>F, "Twin2Motor.txt",'U');
        // printf("高速UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    else{
        Write(&This>F, "Twin2Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin2PermitTurnOpen.txt",'N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin2Motor.txt",'j');
        // printf("上昇します¥n");
        // WaitSecond(&This>T, 1);
    }
}
break;
case 'y':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin2Motor.txt",'s');
        // printf("STOPします¥n");
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin2Motor.txt",'h');
        // printf("開きます¥n");
        // WaitSecond(&This>T, 1);

```

```

    }
    break;
case 'o':
case 'L':
case '2':
    Write(&This>F, "Twin2Motor.txt",'s');
    // printf("STOPします¥n");
    // WaitSecond(&This>T, 1);
    Write(&This>F, "Twin2Motor.txt",'h');
    // printf("開きます¥n");
    // WaitSecond(&This>T, 1);
    break;
case '1':
    This->UD = 'D';
    if(This->chMotor== 'D'){
        Write(&This>F, "Twin2Motor.txt",'D');
        // printf("高速DOWNします¥n");
        // WaitSecond(&This>T, 1);
    }
    else{
        Write(&This>F, "Twin2Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin2PermitTurnOpen.txt",'N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin2Motor.txt",'k');
        // printf("下降します¥n");
        // WaitSecond(&This>T, 1);
    }
    break;

```

```

case '3':
    This->UD = 'U';
    if(This->chMotor=='U'){
        Write(&This>F, "Twin2Motor.txt",'U');
        // printf("高速UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    else{
        Write(&This>F, "Twin2Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin2PermitTurnOpen.txt",'N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin2Motor.txt",'j');
        // printf("上昇します¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'c':
    Write(&This>F, "Twin2Motor.txt",'s');
    // printf("STOPします¥n");
    Write(&This>F, "Twin2PermitTurnOpen.txt",'N');
    // WaitSecond(&This>T, 1);
    break;
case 'h':
    if(TwinCommand[0]!='2'){
        Write(&This>F, "Twin2Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin2PermitTurnOpen.txt",'N');
    }

```

```

        // WaitSecond(&This>T, 1);
    }
    break;
default:
    break;
}
}
else if(This->strLimit[7]== 'y'){ // 2開
    switch(TwinCommand[1]){
    case 'U':
    case 'D':
        if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin2Motor.txt",'s');
            // printf("STOPします¥n");
            // 現在開
            // printf("開いています¥n");
            SetCurrentTime(p_OpenTO);
            SetPermit(p_OpenTOON);
            // WaitSecond(&This>T, 1);
        }
        else{
            Write(&This>F, "Twin2Motor.txt",'t');
            // printf("閉じます¥n");
            Write(&This>F, "Twin2PermitTurnOpen.txt",'o');
            SetPermit(p_OpenTOOFF);
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'y':

```



```

if(TwinCommand[0]== '2'){
    Write(&This>F, "Twin2Motor.txt",'s');
    // printf("STOPします¥n");
    // 現在開
    // printf("開いています¥n");
    SetCurrentTime(p_OpenTO);
    SetPermit(p_OpenTOON);
    // WaitSecond(&This>T, 1);
}
break;

case 'o':
case 'L':
case '2':
    Write(&This>F, "Twin2Motor.txt",'s');
    // printf("STOPします¥n");
    // 現在開
    // printf("開いています¥n");
    SetCurrentTime(p_OpenTO);
    SetPermit(p_OpenTOON);
    // WaitSecond(&This>T, 1);
    break;

case 'c':
case '1':
case '3':
    Write(&This>F, "Twin2Motor.txt",'t');
    // printf("閉じます¥n");
    Write(&This>F, "Twin2PermitTurnOpen.txt",'p');
    SetPermit(p_OpenTOOFF);
    // WaitSecond(&This>T, 1);

```

```

        break;
    case 'h':
        if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin2Motor.txt",'t');
            // printf("閉じます¥n");
            Write(&This>F, "Twin2PermitTurnOpen.txt",'o');
            SetPermit(p_OpenTOOFF);
            // WaitSecond(&This>T, 1);
        }
        break;
    default:
        break;
}
}

else if((This->strLimit[5] == 'y') || (This->strLimit[6] == 'y')){
    switch(TwinCommand[1]){
    case 'U':
    case 'D':
        if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin2Motor.txt",'o');
            // printf("OPENします¥n");
            // WaitSecond(&This>T, 1);
        }
        else{
            Write(&This>F, "Twin2Motor.txt",'c');
            // printf("CLOSEします¥n");
            Write(&This>F, "Twin2PermitTurnOpen.txt",'o');
            // WaitSecond(&This>T, 1);
        }
    }
}

```

```

        break;
    case 'y':
        if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin2Motor.txt",'o');
            // printf("OPENします¥n");
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'o':
    case 'L':
    case '2':
        Write(&This>F, "Twin2Motor.txt",'o');
        // printf("OPENします¥n");
        // WaitSecond(&This>T, 1);
        break;
    default:
        Write(&This>F, "Twin2Motor.txt",'c');
        // printf("CLOSEします¥n");
        Write(&This>F, "Twin2PermitTurnOpen.txt",'o');
        // WaitSecond(&This>T, 1);
        break;
    }
}
else{
    switch(TwinCommand[1]){
    case 'U':
    case 'D':
        if(TwinCommand[0]== '2'){

```

```

        Write(&This>F, "Twin2Motor.txt",'O');
        // printf("高速OPENします¥n");
        // WaitSecond(&This>T, 1);
    }
else{
    Write(&This>F, "Twin2Motor.txt",'C');
    // printf("高速CLOSEします¥n");
    Write(&This>F, "Twin2PermitTurnOpen.txt",'o');
    // WaitSecond(&This>T, 1);
}
break;
case 'y':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin2Motor.txt",'O');
        // printf("高速OPENします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'o':
case 'L':
case '2':
    Write(&This>F, "Twin2Motor.txt",'O');
    // printf("高速OPENします¥n");
    // WaitSecond(&This>T, 1);
    break;
default:
    Write(&This>F, "Twin2Motor.txt",'C');
    // printf("高速CLOSEします¥n");
    Write(&This>F, "Twin2PermitTurnOpen.txt",'o');

```

```

        // WaitSecond(&This>T, 1);
        break;
    }
}
}
else if(This->strLimit[3] == 'y'){
    if(This->strLimit[4] == 'y'){ // 3閉
        This->UD = 'D';
        switch(TwinCommand[1]){
        case 'U':
            if(TwinCommand[0] == '1'){
                Write(&This>F, "Twin2Motor.txt",'s');
                // printf("STOPします¥n");
                Write(&This>F, "Twin2PermitTurnOpen.txt"¥N);
                // WaitSecond(&This>T, 1);
                Write(&This>F, "Twin2Motor.txt",'k');
                // printf("下降します¥n");
                // WaitSecond(&This>T, 1);
            }
            else if(TwinCommand[0] == '2'){
                Write(&This>F, "Twin2Motor.txt",'s');
                // printf("STOPします¥n");
                Write(&This>F, "Twin2PermitTurnOpen.txt"¥N);
                // WaitSecond(&This>T, 1);
                Write(&This>F, "Twin2Motor.txt",'k');
                // printf("下降します¥n");
                // WaitSecond(&This>T, 1);
            }
        }
        break;
    }
}

```

case 'D':

```
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin2Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin2PermitTurnOpen.txt"'\N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin2Motor.txt",'k');
        // printf("下降します¥n");
        // WaitSecond(&This>T, 1);
    }
    else if(TwinCommand[0]== '3'){
        Write(&This>F, "Twin2Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin2PermitTurnOpen.txt"'\N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin2Motor.txt",'h');
        // printf("開きます¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
```

case 'y':

```
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin2Motor.txt",'s');
        // printf("STOPします¥n");
        Write(&This>F, "Twin2PermitTurnOpen.txt"'\N');
        // WaitSecond(&This>T, 1);
        Write(&This>F, "Twin2Motor.txt",'k');
        // printf("下降します¥n");
        // WaitSecond(&This>T, 1);
```

```

    }
    break;
case 'o':
case 'L':
case '3':
    Write(&This>F, "Twin2Motor.txt",'s');
    // printf("STOPします¥n");
    Write(&This>F, "Twin2PermitTurnOpen.txt"'\N');
    // WaitSecond(&This>T, 1);
    Write(&This>F, "Twin2Motor.txt",'h');
    // printf("開きます¥n");
    // WaitSecond(&This>T, 1);
    break;
case '1':
case '2':
    Write(&This>F, "Twin2Motor.txt",'s');
    // printf("STOPします¥n");
    Write(&This>F, "Twin2PermitTurnOpen.txt"'\N');
    // WaitSecond(&This>T, 1);
    Write(&This>F, "Twin2Motor.txt",'k');
    // printf("下降します¥n");
    // WaitSecond(&This>T, 1);
    break;
case 'c':
    Write(&This>F, "Twin2Motor.txt",'s');
    // printf("STOPします¥n");
    Write(&This>F, "Twin2PermitTurnOpen.txt"'\rho');
    // WaitSecond(&This>T, 1);

```

```

        break;
default:
        break;
}
}
else if(This->strLimit[7]== 'y'){ // 3開
    switch(TwinCommand[1]){
    case 'U':
        if(TwinCommand[0]== '1'){
            Write(&This>F, "Twin2Motor.txt",'t');
            // printf("閉じます¥n");
            Write(&This>F, "Twin2PermitTurnOpen.txt"ρ');
            SetPermit(p_OpenTOOFF);
            // WaitSecond(&This>T, 1);
        }
        else if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin2Motor.txt",'t');
            // printf("閉じます¥n");
            Write(&This>F, "Twin2PermitTurnOpen.txt"ρ');
            SetPermit(p_OpenTOOFF);
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'D':
        if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin2Motor.txt",'t');
            // printf("閉じます¥n");
            Write(&This>F, "Twin2PermitTurnOpen.txt"ρ');
            SetPermit(p_OpenTOOFF);

```



```

        // WaitSecond(&This>T, 1);
    }
    else if(TwinCommand[0]== '3'){
        Write(&This>F, "Twin2Motor.txt",'s');
        // printf("STOPします¥n");
        //   現在開
        // printf("開いています¥n");
        SetCurrentTime(p_OpenTO);
        SetPermit(p_OpenTOON);
        // WaitSecond(&This>T, 1);
    }
    break;
case 'o':
case 'L':
case '3':
    Write(&This>F, "Twin2Motor.txt",'s');
    // printf("STOPします¥n");
    //   現在開
    // printf("開いています¥n");
    SetCurrentTime(p_OpenTO);
    SetPermit(p_OpenTOON);
    // WaitSecond(&This>T, 1);
    break;
case 'c':
case '1':
case '2':
    Write(&This>F, "Twin2Motor.txt",'t');
    // printf("閉じます¥n");
    Write(&This>F, "Twin2PermitTurnOpen.txt",'p');

```

```

        SetPermit(p_OpenTOOFF);
        // WaitSecond(&This>T, 1);
        break;
default:
        break;
}
}
else if((This->strLimit[5] == 'y') || (This->strLimit[6] == 'y')){
    switch(TwinCommand[1]){
case 'D':
        if(TwinCommand[0] == '3'){
            Write(&This>F, "Twin2Motor.txt",'o');
            // printf("OPENします¥n");
            // WaitSecond(&This>T, 1);
        }
        else{
            Write(&This>F, "Twin2Motor.txt",'c');
            // printf("CLOSEします¥n");
            Write(&This>F, "Twin2PermitTurnOpen.txt",'o');
            // WaitSecond(&This>T, 1);
        }
        break;
case 'o':
case 'L':
case '3':
        Write(&This>F, "Twin2Motor.txt",'o');
        // printf("OPENします¥n");
        // WaitSecond(&This>T, 1);
        break;

```

```

default:
    Write(&This>F, "Twin2Motor.txt",'c');
    // printf("CLOSEします¥n");
    Write(&This>F, "Twin2PermitTurnOpen.txt",'o');
    // WaitSecond(&This>T, 1);
    break;
}
}
else{
    switch(TwinCommand[1]){
    case 'D':
        if(TwinCommand[0]≠= '3'){
            Write(&This>F, "Twin2Motor.txt",'O');
            // printf("高速OPENします¥n");
            // WaitSecond(&This>T, 1);
        }
        else{
            Write(&This>F, "Twin2Motor.txt",'C');
            // printf("高速CLOSEします¥n");
            Write(&This>F, "Twin2PermitTurnOpen.txt",'o');
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'o':
    case 'L':
    case '3':
        Write(&This>F, "Twin2Motor.txt",'O');
        // printf("高速OPENします¥n");

```

```

        // WaitSecond(&This>T, 1);
        break;
default:
    Write(&This>F, "Twin2Motor.txt";'C');
    // printf("高速CLOSEします¥n");
    Write(&This>F, "Twin2PermitTurnOpen.txt";'o');
    // WaitSecond(&This>T, 1);
    break;
    }
}
}
else if(This->strLimit[1] == 'y'){
    switch(TwinCommand[1]){
    case 'U':
        if(TwinCommand[0]== '1'){
            Write(&This>F, "Twin2Motor.txt";'d');
            // printf("DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
        else if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin2Motor.txt";'u');
            // printf("UPします¥n");
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'D':
        if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin2Motor.txt";'u');
            // printf("UPします¥n");

```

```

        // WaitSecond(&This>T, 1);
    }
    else if(TwinCommand[0]== '3'){
        Write(&This>F, "Twin2Motor.txt",'u');
        // printf("UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'y':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin2Motor.txt",'u');
        // printf("UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case '1':
    Write(&This>F, "Twin2Motor.txt",'d');
    // printf("DOWNします¥n");
    // WaitSecond(&This>T, 1);
    break;
case '2':
case '3':
    Write(&This>F, "Twin2Motor.txt",'u');
    // printf("UPします¥n");
    // WaitSecond(&This>T, 1);
    break;
default:
    break;
}

```

```

}

else if(This->strLimit[8] == 'd'){
    switch(TwinCommand[1]){
    case 'U':
        if(TwinCommand[0] == '1'){
            Write(&This>F, "Twin2Motor.txt";'D');
            // printf("高速DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
        else if(TwinCommand[0] == '2'){
            Write(&This>F, "Twin2Motor.txt";'U');
            // printf("高速UPします¥n");
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'D':
        if(TwinCommand[0] == '2'){
            Write(&This>F, "Twin2Motor.txt";'U');
            // printf("高速UPします¥n");
            // WaitSecond(&This>T, 1);
        }
        else if(TwinCommand[0] == '3'){
            Write(&This>F, "Twin2Motor.txt";'U');
            // printf("高速UPします¥n");
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'y':
        if(TwinCommand[0] == '2'){

```

```

        Write(&This>F, "Twin2Motor.txt";'U');
        // printf("高速UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case '1':
    Write(&This>F, "Twin2Motor.txt";'D');
    // printf("高速DOWNします¥n");
    // WaitSecond(&This>T, 1);
    break;
case '2':
case '3':
    Write(&This>F, "Twin2Motor.txt";'U');
    // printf("高速UPします¥n");
    // WaitSecond(&This>T, 1);
    break;
default:
    break;
}
}
else if(This->strLimit[9] == 'd'){
    switch(TwinCommand[1]){
    case 'U':
        if(TwinCommand[0]== '1'){
            Write(&This>F, "Twin2Motor.txt";'d');
            // printf("DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
    }
}

```

```

else if(TwinCommand[0]== '2'){
    Write(&This>F, "Twin2Motor.txt",'u');
    // printf("UPします¥n");
    // WaitSecond(&This>T, 1);
}
break;
case 'D':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin2Motor.txt",'u');
        // printf("UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    else if(TwinCommand[0]== '3'){
        Write(&This>F, "Twin2Motor.txt",'U');
        // printf("高速UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'y':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin2Motor.txt",'u');
        // printf("UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case '1':
    Write(&This>F, "Twin2Motor.txt",'d');
    // printf("DOWNします¥n");
    // WaitSecond(&This>T, 1);

```



```

        break;
case '2':
    Write(&This>F, "Twin2Motor.txt",'u');
    // printf("UPします¥n");
    // WaitSecond(&This>T, 1);
    break;
case '3':
    Write(&This>F, "Twin2Motor.txt",'U');
    // printf("高速UPします¥n");
    // WaitSecond(&This>T, 1);
    break;
default:
    break;
}
}
else if((This->strLimit[10] == 'u') && (This->strLimit[11] == 'd')){
    switch(TwinCommand[1]){
case 'U':
    if(TwinCommand[0]== '1'){
        Write(&This>F, "Twin2Motor.txt",'D');
        // printf("高速DOWNします¥n");
        // WaitSecond(&This>T, 1);
    }
    else if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin2Motor.txt",'d');
        // printf("DOWNします¥n");
        // WaitSecond(&This>T, 1);
    }
    }
    break;

```

```

case 'D':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin2Motor.txt",'d');
        // printf("DOWNします¥n");
        // WaitSecond(&This>T, 1);
    }
    else if(TwinCommand[0]== '3'){
        Write(&This>F, "Twin2Motor.txt",'u');
        // printf("UPします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case 'y':
    if(TwinCommand[0]== '2'){
        Write(&This>F, "Twin2Motor.txt",'d');
        // printf("DOWNします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case '1':
    Write(&This>F, "Twin2Motor.txt",'D');
    // printf("高速DOWNします¥n");
    // WaitSecond(&This>T, 1);
    break;
case '2':
    Write(&This>F, "Twin2Motor.txt",'d');
    // printf("DOWNします¥n");
    // WaitSecond(&This>T, 1);
    break;

```

```

case '3':
    Write(&This>F, "Twin2Motor.txt",'u');
    // printf("UPします¥n");
    // WaitSecond(&This>T, 1);
    break;
default:
    break;
}
}
else if((This->strLimit[11] == 'u') && (This->strLimit[2] == 'n')){
    switch(TwinCommand[1]){
    case 'U':
        if(TwinCommand[0]== '1'){
            Write(&This>F, "Twin2Motor.txt",'D');
            // printf("高速DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
        else if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin2Motor.txt",'D');
            // printf("高速DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'D':
        if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin2Motor.txt",'D');
            // printf("高速DOWNします¥n");
            // WaitSecond(&This>T, 1);

```

```

    }

    else if(TwinCommand[0]== '3'){

        Write(&This>F, "Twin2Motor.txt",'U');

        // printf("高速UPします¥n");

        // WaitSecond(&This>T, 1);

    }

    break;

case 'y':

    if(TwinCommand[0]== '2'){

        Write(&This>F, "Twin2Motor.txt",'D');

        // printf("高速DOWNします¥n");

        // WaitSecond(&This>T, 1);

    }

    break;

case '1':

case '2':

    Write(&This>F, "Twin2Motor.txt",'D');

    // printf("高速DOWNします¥n");

    // WaitSecond(&This>T, 1);

    break;

case '3':

    Write(&This>F, "Twin2Motor.txt",'U');

    // printf("高速UPします¥n");

    // WaitSecond(&This>T, 1);

    break;

default:

    break;

}

}

```

```

else if(This->strLimit[2] == 'y'){
    switch(TwinCommand[1]){
    case 'U':
        if(TwinCommand[0]== '1'){
            Write(&This>F, "Twin2Motor.txt",'d');
            // printf("DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
        else if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin2Motor.txt",'d');
            // printf("DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'D':
        if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin2Motor.txt",'d');
            // printf("DOWNします¥n");
            // WaitSecond(&This>T, 1);
        }
        else if(TwinCommand[0]== '3'){
            Write(&This>F, "Twin2Motor.txt",'u');
            // printf("UPします¥n");
            // WaitSecond(&This>T, 1);
        }
        break;
    case 'y':
        if(TwinCommand[0]== '2'){
            Write(&This>F, "Twin2Motor.txt",'d');

```

```

        // printf("DOWNします¥n");
        // WaitSecond(&This>T, 1);
    }
    break;
case '1':
case '2':
    Write(&This>F, "Twin2Motor.txt", 'd');
    // printf("DOWNします¥n");
    // WaitSecond(&This>T, 1);
    break;
case '3':
    Write(&This>F, "Twin2Motor.txt", 'u');
    // printf("UPします¥n");
    // WaitSecond(&This>T, 1);
    break;
default:
    break;
}
}
return;
}

void EV_Twin2_Q_OTOUDChenge(struct EV_Twin_Queue *This, struct EV_Time *p_OpenTO, Thread
*th)
{
    char *p_cmd;
    char cmd;
    p_cmd = &cmd;

```

```

cmd = 'N';
This->p_tOpenTO = p_OpenTO;
This->p_strLimit = &This->strLimit[0];
ReadString(&This->F, "Twin2Limit.txt", This->p_strLimit, LIMIT);
if((This->UD == 'U') && (This->p_strLimit[9] == 'u') && (This->p_strLimit[10] == 'd'))
{
    // 0L,0o,02,2U,2y,0c,2h,03,3D
    if(EV_Twin2_Q_Check(This,
"0L", "0o", "02", "2U", "2y", "NN", "NN", "NN", "NN", "NN", "NN", "NN", "NN") == OK)
    {
        WriteString(&This->F, "Twin2Command.txt", "0o");
        EV_Twin2_Q_Command_Read(This, p_cmd, th);
        EV_Twin2_Q_CheckTurnOpen(This);
        SetCurrentTime(This->p_tOpenTO);
        SetPermit(This->p_tOpenTO, ON);
    }
    else if(EV_Twin2_Q_Check(This,
"0c", "2h", "03", "3D", "NN", "NN", "NN", "NN", "NN", "NN", "NN", "NN", "NN") ==
ONE_MORE_TIME)
    {
        if(EV_Twin2_Q_Check(This,
"2D", "01", "1U", "NN", "NN", "NN", "NN", "NN", "NN", "NN", "NN", "NN") == OK)
        {
            WriteString(&This->F, "Twin2Command.txt", "0c");
            EV_Twin2_Q_Command_Read(This, p_cmd, th);
            EV_Twin2_Q_Motor(This->p_OpenTO);
            This->UD = 'D';
            EV_Twin2_Q_Write(This, "Twin2Command.txt", "2U", 'r');

```

```

    }
else
{
    WriteString(&This->F, "Twin2Command.txt", "0c¥0");
    EV_Twin2_Q_Command_Read(This, p_cmd, th);
    EV_Twin2_Q_Motor(Thisp_OpenTO);
}
}
else
{
    WriteString(&This->F, "Twin2Command.txt", "0c¥0");
    EV_Twin2_Q_Command_Read(This, p_cmd, th);
    EV_Twin2_Q_Motor(This, p_OpenTO);
}
}
else if((This->UD == 'D') && (This->p_strLimit[9] == 'u') && (This->p_strLimit[10] == 'd'))
{
    // 0L,0o,02,2D,2y,0c,2h,01,1U
    if(EV_Twin2_Q_Check(This,
"0L¥0", "0o¥0", "02¥0", "2D¥0", "2y¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0") == OK)
    {
        WriteString(&This->F, "Twin2Command.txt", "0o¥0");
        EV_Twin2_Q_Command_Read(This, p_cmd, th);
        EV_Twin2_Q_CheckTurnOpen(This);
        SetCurrentTime(This->p_tOpenTO);
        SetPermit(This->p_tOpenTO, ON);
    }
    else if(EV_Twin2_Q_Check(This,
"0c¥0", "2h¥0", "01¥0", "1U¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0", "NN¥0") ==

```



```
ONE_MORE_TIME)
```

```
{
```

```
    if(EV_Twin2_Q_Check(This,
```

```
"2U¥0","03¥0","3D¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0","NN¥0") == OK)
```

```
    {
```

```
        WriteString(&This->F, "Twin2Command.txt", "0c¥0");
```

```
        EV_Twin2_Q_Command_Read(This, p_cmd, th);
```

```
        EV_Twin2_Q_Motor(Thisp_OpenTO);
```

```
        This->UD = 'U';
```

```
        EV_Twin2_Q_Write(This, "Twin2Command.txt", "2D¥0", 'r');
```

```
    }
```

```
else
```

```
{
```

```
    WriteString(&This->F, "Twin2Command.txt", "0c¥0");
```

```
    EV_Twin2_Q_Command_Read(This, p_cmd, th);
```

```
    EV_Twin2_Q_Motor(Thisp_OpenTO);
```

```
}
```

```
}
```

```
else
```

```
{
```

```
    WriteString(&This->F, "Twin2Command.txt", "0c¥0");
```

```
    EV_Twin2_Q_Command_Read(This, p_cmd, th);
```

```
    EV_Twin2_Q_Motor(This, p_OpenTO);
```

```
}
```

```
}
```

```
else
```

```
{
```

```
    WriteString(&This->F, "Twin2Command.txt", "0c¥0");
```

```
    EV_Twin2_Q_Command_Read(This, p_cmd, th);
```

```
EV_Twin2_Q_Motor(This, p_OpenTO);
```

```
}
```

```
return;
```

```
}
```

```
// EV_UserController.h
```

```
#ifndef Panel_h
```

```
#define Panel_h
```

```
#include "Panel.h"
```

```
#endif
```

```
#ifndef Timer_h
```

```
#define Timer_h
```

```
#include "Timer.h"
```

```
#endif
```

```
#ifndef EV_Time_h
```

```
#define EV_Time_h
```

```
#include "EV_Time.h"
```

```
#endif
```

```
#ifndef EV_File_h
```

```
#define EV_File_h
```

```
#include "EV_File.h"
```

```
#endif
```

```
#ifndef EV_Queue_h
```

```
#define EV_Queue_h
```

```
#include "EV_Queue.h"
```

```
#endif
```

```
struct EV_UserController{
```

```
// 経過時間テンポラリ
struct EV_Time T;
struct EV_Time OpenTO;
// 命令
char Command;
char *p_Command;
struct EV_Queue Q;
// ファイルストリーム
struct EV_File F;
// 地震用
char ech;
char *p_ech;
// 閉失敗時の反転開
int TurnOpen;
int *p_TurnOpen;
};

void EV_UC_Init(struct EV_UserController *This, Thread *th);
void EV_UC_On(struct EV_UserController *This, Thread *th);
```

// EV_UserController.c : メイン プロジェクト ファイルです。

```
#include "C.h"
```

```
#include "EV_UserController.h"
```

```
void EV_UC_Init(struct EV_UserController *This, Thread *th){
```

```
    EV_Time(&This->T, th);
```

```
    EV_Time(&This->OpenTO, th);
```

```
    // 命令初期化
```

```
    This->p_Command = &This->Command;
```

```
    // Queueテーブルフォーマット
```

```
    EV_Q_Init(&This->Q, th);
```

```
    EV_File(&This->F);
```

```
    // 地震用
```

```
    This->p_ech = &This->ech;
```

```
    // 閉失敗時の反転開初期化
```

```
    This->p_TurnOpen = &This->TurnOpen;
```

```
    *This->p_TurnOpen = OFF;
```

```
    // モーター停止命令
```

```
    Write(&This->F, "UserMotor.txt", 's');
```

```
    // 地震検知初期化
```

```
    Write(&This->F, "Earthquake.txt", 'N');
```

```
    // ファイル初期化
```

```
    Write(&This->F, "UserPermitCommand.txt", 'N');
```

```
    Command_Write(&This->F, 'N');
```

```
    WriteString(&This->F, "UserCommand.txt", "NN¥0");
```

```
    Write(&This->F, "UserPermitTurnOpen.txt", 'N');
```

```
    Write(&This->F, "UserTurnOpen.txt", 'N');
```

```

// 命令入力許可
Write(&This->F, "UserPermitCommand.txt", 'c');

return;
}

void EV_UC_On(struct EV_UserController *This, Thread *th){
    // 地震用
    Read(&This->F, "Earthquake.txt", This->p_ech);
    if(This->ech == 'e'){
        // printf("地震により扉開きます¥n");
        WriteString(&This->F, "UserCommand.txt", "01¥0");
        Write(&This->F, "Earthquake.txt", 'N');
    }
    // 命令入力
    switch(EV_Q_Command_Read(&This->Q, This->p_Command, th)){
    case NG:
        return;
    case ONE_MORE_TIME:
        if((GetPermit(&This->OpenTO) == ON)
            && (GetCurrentTime(&This->OpenTO) > OPENTIMEOUT)){
            // printf("開タイムアウト¥n");
            // Command= 'c';
            SetPermit(&This->OpenTO, OFF);
            EV_Q_OTOUChange(&This->Q, &This->OpenTO, th);
        }
        else
        {
            EV_Q_Motor(&This->Q, &This->OpenTO);
        }
    }
}

```

```
        break;
default:
        break;
}
return;
}
```

```
// EV_Twin_Controller.h
```

```
#ifndef Panel_h
```

```
#define Panel_h
```

```
#include "Panel.h"
```

```
#endif
```

```
#ifndef Timer_h
```

```
#define Timer_h
```

```
#include "Timer.h"
```

```
#endif
```

```
#ifndef EV_Time_h
```

```
#define EV_Time_h
```

```
#include "EV_Time.h"
```

```
#endif
```

```
#ifndef EV_File_h
```

```
#define EV_File_h
```

```
#include "EV_File.h"
```

```
#endif
```

```
#ifndef EV_Twin_Queue_h
```

```
#define EV_Twin_Queue_h
```

```
#include "EV_Twin_Queue.h"
```

```
#endif
```

```
struct EV_Twin_Controller{
```



```

// 経過時間テンポラリ
struct EV_Time T;

struct EV_Time OpenTO;

// 命令
char Command;

char *p_Command;

struct EV_Twin_Queue Q;

// ファイルストリーム
struct EV_File F;

// 地震用
char ech;

char *p_ech;

// 閉失敗時の反転開
int TurnOpen;

int *p_TurnOpen;

};

void EV_T1C_Init(struct EV_Twin_Controller *This, Thread *th);
void EV_T1C_On(struct EV_Twin_Controller *This, Thread *th);

void EV_T2C_Init(struct EV_Twin_Controller *This, Thread *th);
void EV_T2C_On(struct EV_Twin_Controller *This, Thread *th);

```

// EV_Twin_Controller.c : メイン プロジェクト ファイルです。

```
#include "C.h"
```

```
#include "EV_Twin_Controller.h"
```

```
void EV_T1C_Init(struct EV_Twin_Controller *This, Thread *th){
```

```
    EV_Time(&This->T, th);
```

```
    EV_Time(&This->OpenTO, th);
```

```
    // 命令初期化
```

```
    This->p_Command = &This->Command;
```

```
    // Queueテーブルフォーマット
```

```
    EV_Twin1_Q_Init(&This->Q, th);
```

```
    EV_File(&This->F);
```

```
    // 地震用
```

```
    This->p_ech = &This->ech;
```

```
    // 閉失敗時の反転開初期化
```

```
    This->p_TurnOpen = &This->TurnOpen;
```

```
    *This->p_TurnOpen = OFF;
```

```
    // モーター停止命令
```

```
    Write(&This->F, "Twin1Motor.txt", 's');
```

```
    // 地震検知初期化
```

```
    Write(&This->F, "Earthquake.txt", 'N');
```

```
    // ファイル初期化
```

```
    Write(&This->F, "Twin1PermitCommand.txt", 'N');
```

```
    Command_Write(&This->F, 'N');
```

```
    WriteString(&This->F, "Twin1Command.txt", "NN¥0");
```

```
    Write(&This->F, "Twin1PermitTurnOpen.txt", 'N');
```

```
    Write(&This->F, "Twin1TurnOpen.txt", 'N');
```

```

// 命令入力許可
Write(&This->F, "Twin1PermitCommand.txt", 'c');

return;
}

void EV_T1C_On(struct EV_Twin_Controller *This, Thread *th){
    // 地震用
    Read(&This->F, "Earthquake.txt", This->p_ech);
    if(This->ech == 'e'){
        // printf("地震により扉開きます¥n");
        WriteString(&This->F, "Twin1Command.txt", "01¥0");
        Write(&This->F, "Earthquake.txt", 'N');
    }
    // 命令入力
    switch(EV_Twin1_Q_Command_Read(&This->Q, This->p_Command, th)){
    case NG:
        return;
    case ONE_MORE_TIME:
        if((GetPermit(&This->OpenTO) == ON)
            && (GetCurrentTime(&This->OpenTO) > OPENTIMEOUT)){
            // printf("開タイムアウト¥n");
            // Command= 'c';
            SetPermit(&This->OpenTO, OFF);
            EV_Twin1_Q_OTOUDChenge(&This->Q, &This->OpenTO, th);
        }
        else
        {
            EV_Twin1_Q_Motor(&This->Q, &This->OpenTO);
        }
    }
}

```

```
break;
```

```
default:
```

```
break;
```

```
}
```

```
return;
```

```
}
```

// EV_Twin_Controller.c : メイン プロジェクト ファイルです。

```
#include "C.h"
```

```
#include "EV_Twin_Controller.h"
```

```
void EV_T2C_Init(struct EV_Twin_Controller *This, Thread *th){
```

```
    EV_Time(&This->T, th);
```

```
    EV_Time(&This->OpenTO, th);
```

```
    // 命令初期化
```

```
    This->p_Command = &This->Command;
```

```
    // Queueテーブルフォーマット
```

```
    EV_Twin2_Q_Init(&This->Q, th);
```

```
    EV_File(&This->F);
```

```
    // 地震用
```

```
    This->p_ech = &This->ech;
```

```
    // 閉失敗時の反転開初期化
```

```
    This->p_TurnOpen = &This->TurnOpen;
```

```
    *This->p_TurnOpen = OFF;
```

```
    // モーター停止命令
```

```
    Write(&This->F, "Twin2Motor.txt", 's');
```

```
    // 地震検知初期化
```

```
    Write(&This->F, "Earthquake.txt", 'N');
```

```
    // ファイル初期化
```

```
    Write(&This->F, "Twin2PermitCommand.txt", 'N');
```

```
    Command_Write(&This->F, 'N');
```

```
    WriteString(&This->F, "Twin2Command.txt", "NN¥0");
```

```
    Write(&This->F, "Twin2PermitTurnOpen.txt", 'N');
```

```
    Write(&This->F, "Twin2TurnOpen.txt", 'N');
```

```

// 命令入力許可
Write(&This->F, "Twin2PermitCommand.txt", 'c');

return;
}

void EV_T2C_On(struct EV_Twin_Controller *This, Thread *th){
    // 地震用
    Read(&This->F, "Earthquake.txt", This->p_ech);
    if(This->ech == 'e'){
        // printf("地震により扉開きます¥n");
        WriteString(&This->F, "Twin2Command.txt", "01¥0");
        Write(&This->F, "Earthquake.txt", 'N');
    }
    // 命令入力
    switch(EV_Twin2_Q_Command_Read(&This->Q, This->p_Command, th)){
    case NG:
        return;
    case ONE_MORE_TIME:
        if((GetPermit(&This->OpenTO) == ON)
            && (GetCurrentTime(&This->OpenTO) > OPENTIMEOUT)){
            // printf("開タイムアウト¥n");
            // Command= 'c';
            SetPermit(&This->OpenTO, OFF);
            EV_Twin2_Q_OTOUDChenge(&This->Q, &This->OpenTO, th);
        }
        else
        {
            EV_Twin2_Q_Motor(&This->Q, &This->OpenTO);
        }
    }
}

```

```
break;
```

```
default:
```

```
break;
```

```
}
```

```
return;
```

```
}
```

```
/* EV_UserSimulator.h */
```

```
#ifndef Panel_h
```

```
#define Panel_h
```

```
#include "Panel.h"
```

```
#endif
```

```
#ifndef Timer_h
```

```
#define Timer_h
```

```
#include "Timer.h"
```

```
#endif
```

```
#ifndef EV_Time_h
```

```
#define EV_Time_h
```

```
#include "EV_Time.h"
```

```
#endif
```

```
#ifndef EV_File_h
```

```
#define EV_File_h
```

```
#include "EV_File.h"
```

```
#endif
```

```
/*=====
```

```
シミュレータを表す構造体
```

```
=====*/
```

```
struct EV_UserSimulator
```

```
{
```

```
    char ch;
```



```
char *p_ch;  
char ch2;  
char *p_ch2;  
char ch3;  
char *p_ch3;  
char str[13];  
char *p_str;
```

```
/* 時間管理 */
```

```
struct EV_Time T;
```

```
/* ファイルストリーム */
```

```
struct EV_File SF;
```

```
struct EV_File CF;
```

```
struct EV_File MF;
```

```
struct EV_File LF;
```

```
};
```

```
/*=====
```

```
シミュレータを表すコンストラクタとメソッドのプロトタイプ宣言
```

```
=====*/
```

```
void EV_US_Init(struct EV_UserSimulator *This, Thread *th);
```

```
void EV_US_On(struct EV_UserSimulator *This, Thread *th);
```

```

/* EV_UserSimulator.c */

#include "C.h"
#include "EV_UserSimulator.h"

/*=====
シミュレータ関数
=====*/

void EV_US_Init(struct EV_UserSimulator *This, Thread *th)
{
    /* 初期化 */
    EV_Time(&This->T, th);
    SetCurrentTime(&This->T);
    This->p_ch = &This->ch;
    This->p_ch2 = &This->ch2;
    This->p_ch3 = &This->ch3;
    This->p_str = &This->str[0];
    This->str[12] = '¥0';
    EV_File(&This->SF);
    EV_File(&This->CF);
    EV_File(&This->MF);
    EV_File(&This->LF);

    /* モーター命令解読 */
    if(Read(&This->MF, "UserMotor.txt¥0", This->p_ch) == NG) return;
    /* リミットスイッチの前状態読み込み */
    if(ReadString(&This->LF, "UserLimit.txt¥0", This->p_str, 13) == NG) return;
}

```

```

void EV_US_On(struct EV_UserSimulator *This, Thread *th)
{
    /* 終了条件 */
    Read(&This->CF, "Command.txt¥0", This->p_ch3);
    if(This->ch3 == 'q'){
        Clear();
        delete_(th);
        return;
    }

    /* モーター命令解釈 */
    Read(&This->MF, "UserMotor.txt¥0", This->p_ch);

    /* リミットスイッチの前状態読み込み */
    ReadString(&This->LF, "UserLimit.txt¥0", This->p_str, 13);

    /* 停止条件 */
    Read(&This->SF, "UserSafety.txt¥0", This->p_ch2);
    if(This->ch2 == 's'){
        Write(&This->CF, "UserCommand.txt¥0", 'N');
        return;
    }

    switch(This->ch)
    {
    case 't':
        if(strncmp(This->p_str, "yynnnnyddddd¥0", 13) == 0){
            strncpy(This->p_str, "yynnnnyndddd¥0", 13);

```

```

        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnnnyuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnnyuudd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnyynnyuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyynnyuuuu¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    break;
case 'c':
    if(strncmp(This->p_str, "yynnnnyndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnnyndddd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "yynnnynndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnynndddd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnnnyuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnnyuudd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnnyynuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnyynuudd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnyynnyuuuu¥0", 13) == 0){

```

```

        strncpy(This->p_str, "nnyynnnnuuuu¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnyynnnnuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyyyynnuuuu¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    break;
case 'C':
    if(strncmp(This->p_str, "yynnnnnndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnynndddd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnnnnnuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnynnuudd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnyynnnnuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyynnnnuuuu¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    break;
case 'j':
    if(strncmp(This->p_str, "yynnyynndddd¥0", 13) == 0){
        strncpy(This->p_str, "nynnyynndddd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuudd¥0", 13);

```

```

        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    break;
case 'u':
    if(strncmp(This->p_str, "nynnyynndddd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynndddd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuddd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuddd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuuud¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuuuu¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnyynnuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyyyynnuuuu¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    break;
case 'U':
    if(strncmp(This->p_str, "nnnnyynndddd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuddd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuddd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuddd¥0", 13);

```

```

        nextRun(th,667);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuud¥0", 13);
        nextRun(th,667);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuud¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuuu¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnnyynnuuu¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    break;
case 'k':
    if(strncmp(This->p_str, "nnyyyynnuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnnyynnuuu¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuddd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    break;
case 'd':
    if(strncmp(This->p_str, "nnnyynnuuu¥0", 13) == 0){

```

```

        strncpy(This->p_str, "nnnnyynnuuuu¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    if(strncmp(This->p_str, "nnnnyynnuuud¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuudd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    if(strncmp(This->p_str, "nnnnyynnuddd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynndddd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nynnnyyndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnyyndddd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    break;
case 'D':
    if(strncmp(This->p_str, "nnnnyynnuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuuud¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuuud¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuudd¥0", 13);
        nextRun(th,667);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuddd¥0", 13);
        nextRun(th,667);
    }

```



```

        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuddd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynndddd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynndddd¥0", 13) == 0){
        strncpy(This->p_str, "nynnyynndddd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    break;
case 'h':
    if(strncmp(This->p_str, "nnyyyynnuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyynynnuuuu¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuddd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnyynnuddd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "yynnyynndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnynndddd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnyynynnuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyynynnuuuu¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnyynnnnuuuu¥0", 13) == 0){

```

```

        strncpy(This->p_str, "nnyynnyuuuu¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
else if(strncmp(This->p_str, "nnyynnyuuuu¥0", 13) == 0){
    strncpy(This->p_str, "nnyynnyuuuu¥0", 13);
    WriteString(&This->LF, "UserLimit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "nnnnnynnuudd¥0", 13) == 0){
    strncpy(This->p_str, "nnnnnynnuudd¥0", 13);
    WriteString(&This->LF, "UserLimit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "nnnnnnnuudd¥0", 13) == 0){
    strncpy(This->p_str, "nnnnnnyuudd¥0", 13);
    WriteString(&This->LF, "UserLimit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "nnnnnynnuudd¥0", 13) == 0){
    strncpy(This->p_str, "nnnnnnyuudd¥0", 13);
    WriteString(&This->LF, "UserLimit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "yynnnynndddd¥0", 13) == 0){
    strncpy(This->p_str, "yynnnynndddd¥0", 13);
    WriteString(&This->LF, "UserLimit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "yynnnnnndddd¥0", 13) == 0){
    strncpy(This->p_str, "yynnnnyndddd¥0", 13);
    WriteString(&This->LF, "UserLimit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "yynnnnyndddd¥0", 13) == 0){

```

```

        strncpy(This->p_str, "yynnnnyyddd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    break;
case 'o':
    if(strncmp(This->p_str, "nnyynnnuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyynnnuuuu¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnyynnyuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyynnyuuuu¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnnyuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnyuudd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnnyuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnyuudd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "yynnnynndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnynndddd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "yynnnnyndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnnyndddd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
}

```

```
    break;
case 'O':
    if(strncmp(This->p_str, "nnyynnnnuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyynnyuuuu¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnnnnuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnnyuudd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "yynnnnnndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnnyndddd¥0", 13);
        WriteString(&This->LF, "UserLimit.txt¥0", This->str);
    }
    break;
default:
    break;
}
return;
}
```

```
/* EV_Twin_Simulator.h */
```

```
#ifndef Panel_h
```

```
#define Panel_h
```

```
#include "Panel.h"
```

```
#endif
```

```
#ifndef Timer_h
```

```
#define Timer_h
```

```
#include "Timer.h"
```

```
#endif
```

```
#ifndef EV_Time_h
```

```
#define EV_Time_h
```

```
#include "EV_Time.h"
```

```
#endif
```

```
#ifndef EV_File_h
```

```
#define EV_File_h
```

```
#include "EV_File.h"
```

```
#endif
```

```
/*=====
```

```
シミュレータを表す構造体
```

```
=====*/
```

```
struct EV_Twin_Simulator
```

```
{
```

```
    char ch;
```

```
char *p_ch;  
char ch2;  
char *p_ch2;  
char ch3;  
char *p_ch3;  
char str[13];  
char *p_str;
```

```
/* 時間管理 */
```

```
struct EV_Time T;
```

```
/* ファイルストリーム */
```

```
struct EV_File SF;
```

```
struct EV_File CF;
```

```
struct EV_File MF;
```

```
struct EV_File LF;
```

```
};
```

```
/*=====
```

```
シミュレータを表すコンストラクタとメソッドのプロトタイプ宣言
```

```
=====*/
```

```
void EV_T1S_Init(struct EV_Twin_Simulator *This, Thread *th);
```

```
void EV_T1S_On(struct EV_Twin_Simulator *This, Thread *th);
```

```
void EV_T2S_Init(struct EV_Twin_Simulator *This, Thread *th);
```

```
void EV_T2S_On(struct EV_Twin_Simulator *This, Thread *th);
```

```
/* EV_Twin1_Simulator.c */
```

```
#include "C.h"
```

```
#include "EV_Twin_Simulator.h"
```

```
/*=====
```

```
シミュレータ関数
```

```
=====*/
```

```
void EV_T1S_Init(struct EV_Twin_Simulator *This, Thread *th)
```

```
{
```

```
    /* 初期化 */
```

```
    EV_Time(&This->T, th);
```

```
    SetCurrentTime(&This->T);
```

```
    This->p_ch = &This->ch;
```

```
    This->p_ch2 = &This->ch2;
```

```
    This->p_ch3 = &This->ch3;
```

```
    This->p_str = &This->str[0];
```

```
    This->str[12] = '¥0';
```

```
    EV_File(&This->SF);
```

```
    EV_File(&This->CF);
```

```
    EV_File(&This->MF);
```

```
    EV_File(&This->LF);
```

```
    /* モーター命令解読 */
```

```
    if(Read(&This->MF, "Twin1Motor.txt¥0", This->p_ch) == NG) return;
```

```
    /* リミットスイッチの前状態読み込み */
```

```
    if(ReadString(&This->LF, "Twin1Limit.txt¥0", This->p_str, 13) == NG) return;
```

```
}
```

```

void EV_T1S_On(struct EV_Twin_Simulator *This, Thread *th)
{
    /* 終了条件 */
    Read(&This->CF, "Command.txt¥0", This->p_ch3);
    if(This->ch3 == 'q'){
        Clear();
        delete_(th);
        return;
    }

    /* モーター命令解釈 */
    Read(&This->MF, "Twin1Motor.txt¥0", This->p_ch);

    /* リミットスイッチの前状態読み込み */
    ReadString(&This->LF, "Twin1Limit.txt¥0", This->p_str, 13);

    /* 停止条件 */
    Read(&This->SF, "Twin1Safety.txt¥0", This->p_ch2);
    if(This->ch2 == 's'){
        Write(&This->CF, "Twin1Command.txt¥0", 'N');
        return;
    }

    switch(This->ch)
    {
    case 't':
        if(strncmp(This->p_str, "yynnnnyydddd¥0", 13) == 0){
            strncpy(This->p_str, "yynnnnyndddd¥0", 13);

```



```

        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnnnyuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnnyuudd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnyynnyuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyynnyuuuu¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    break;
case 'c':
    if(strncmp(This->p_str, "yynnnnyndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnnyndddd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "yynnnynndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnynndddd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnnnyuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnnyuudd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnnyynnuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnyynnuudd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnyynnyuuuu¥0", 13) == 0){

```

```

        strncpy(This->p_str, "nnyynnnuuuu¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
else if(strncmp(This->p_str, "nnyynnnuuuu¥0", 13) == 0){
    strncpy(This->p_str, "nnyyyynnnuuuu¥0", 13);
    WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
}
break;
case 'C':
    if(strncmp(This->p_str, "yynnnnnndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnynndddd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
else if(strncmp(This->p_str, "nnnnnnnnuudd¥0", 13) == 0){
    strncpy(This->p_str, "nnnnnynnuudd¥0", 13);
    WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "nnyynnnuuuu¥0", 13) == 0){
    strncpy(This->p_str, "nnyynnnuuuu¥0", 13);
    WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
}
break;
case 'j':
    if(strncmp(This->p_str, "yynnyynndddd¥0", 13) == 0){
        strncpy(This->p_str, "nynnyynndddd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
else if(strncmp(This->p_str, "nnnnyynnuudd¥0", 13) == 0){
    strncpy(This->p_str, "nnnnyynnuudd¥0", 13);

```

```

        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    break;
case 'u':
    if(strncmp(This->p_str, "nynnyynndddd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynndddd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuddd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuddd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuuud¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuuuu¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnyynnuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyyyynnuuuu¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    break;
case 'U':
    if(strncmp(This->p_str, "nnnnyynndddd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuddd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuddd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuddd¥0", 13);

```

```

        nextRun(th,667);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
else if(strncmp(This->p_str, "nnnnyynnuudd¥0", 13) == 0){
    strncpy(This->p_str, "nnnnyynnuud¥0", 13);
    nextRun(th,667);
    WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "nnnnyynnuud¥0", 13) == 0){
    strncpy(This->p_str, "nnnnyynnuuu¥0", 13);
    WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "nnnnyynnuuu¥0", 13) == 0){
    strncpy(This->p_str, "nnnyynnuuu¥0", 13);
    WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
}
break;
case 'k':
    if(strncmp(This->p_str, "nnyyyynnuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnnyynnuuu¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
else if(strncmp(This->p_str, "nnnnyynnuudd¥0", 13) == 0){
    strncpy(This->p_str, "nnnnyynnuddd¥0", 13);
    WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
}
break;
case 'd':
    if(strncmp(This->p_str, "nnnyynnuuu¥0", 13) == 0){

```

```

        strncpy(This->p_str, "nnnnyynnuuuu¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    if(strncmp(This->p_str, "nnnnyynnuuud¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuudd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    if(strncmp(This->p_str, "nnnnyynnuddd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynndddd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nynnnyndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnyndddd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    break;
case 'D':
    if(strncmp(This->p_str, "nnnnyynnuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuuud¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuuud¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuudd¥0", 13);
        nextRun(th,667);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuddd¥0", 13);
        nextRun(th,667);
    }

```

```

        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuddd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynndddd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynndddd¥0", 13) == 0){
        strncpy(This->p_str, "nynnyynndddd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    break;
case 'h':
    if(strncmp(This->p_str, "nnyyyynnuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyynynnuuuu¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnyynnuudd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "yynnyynndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnynndddd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnyynynnuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyynynnuuuu¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnyynnnnuuuu¥0", 13) == 0){

```

```

    strncpy(This->p_str, "nnyynnyuuuu¥0", 13);
    WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "nnyynnyuuuu¥0", 13) == 0){
    strncpy(This->p_str, "nnyynnyuuuu¥0", 13);
    WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "nnnnnynnuudd¥0", 13) == 0){
    strncpy(This->p_str, "nnnnnynnuudd¥0", 13);
    WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "nnnnnnnuudd¥0", 13) == 0){
    strncpy(This->p_str, "nnnnnnnyuudd¥0", 13);
    WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "nnnnnynnuudd¥0", 13) == 0){
    strncpy(This->p_str, "nnnnnnyuudd¥0", 13);
    WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "yynnnynndddd¥0", 13) == 0){
    strncpy(This->p_str, "yynnnynndddd¥0", 13);
    WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "yynnnnnndddd¥0", 13) == 0){
    strncpy(This->p_str, "yynnnnyyddd¥0", 13);
    WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "yynnnnyndddd¥0", 13) == 0){

```

```

        strncpy(This->p_str, "yynnnnyyddd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    break;
case 'o':
    if(strncmp(This->p_str, "nnyynnnuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyynnnuuuu¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnyynnyuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyynnyuuuu¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnnnyuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnnyuudd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnnnyyudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnnyyudd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "yynnnynndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnynndddd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "yynnnnyndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnnyndddd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
}

```



```

    break;
case 'O':
    if(strncmp(This->p_str, "nnyynnnnuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyynnyuuuu¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnnnnuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnnyuudd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "yynnnnnndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnnyndddd¥0", 13);
        WriteString(&This->LF, "Twin1Limit.txt¥0", This->str);
    }
    break;
default:
    break;
}
return;
}

```

```
/* EV_Twin2_Simulator.c */
```

```
#include "C.h"
```

```
#include "EV_Twin_Simulator.h"
```

```
/*=====
```

```
シミュレータ関数
```

```
=====*/
```

```
void EV_T2S_Init(struct EV_Twin_Simulator *This, Thread *th)
```

```
{
```

```
    /* 初期化 */
```

```
    EV_Time(&This->T, th);
```

```
    SetCurrentTime(&This->T);
```

```
    This->p_ch = &This->ch;
```

```
    This->p_ch2 = &This->ch2;
```

```
    This->p_ch3 = &This->ch3;
```

```
    This->p_str = &This->str[0];
```

```
    This->str[12] = '¥0';
```

```
    EV_File(&This->SF);
```

```
    EV_File(&This->CF);
```

```
    EV_File(&This->MF);
```

```
    EV_File(&This->LF);
```

```
    /* モーター命令解読 */
```

```
    if(Read(&This->MF, "Twin2Motor.txt¥0", This->p_ch) == NG) return;
```

```
    /* リミットスイッチの前状態読み込み */
```

```
    if(ReadString(&This->LF, "Twin2Limit.txt¥0", This->p_str, 13) == NG) return;
```

```
}
```

```

void EV_T2S_On(struct EV_Twin_Simulator *This, Thread *th)
{
    /* 終了条件 */
    Read(&This->CF, "Command.txt¥0", This->p_ch3);
    if(This->ch3 == 'q'){
        Clear();
        delete_(th);
        return;
    }

    /* モーター命令解釈 */
    Read(&This->MF, "Twin2Motor.txt¥0", This->p_ch);

    /* リミットスイッチの前状態読み込み */
    ReadString(&This->LF, "Twin2Limit.txt¥0", This->p_str, 13);

    /* 停止条件 */
    Read(&This->SF, "Twin2Safety.txt¥0", This->p_ch2);
    if(This->ch2 == 's'){
        Write(&This->CF, "Twin2Command.txt¥0", 'N');
        return;
    }

    switch(This->ch)
    {
    case 't':
        if(strncmp(This->p_str, "yynnnnyddddd¥0", 13) == 0){
            strncpy(This->p_str, "yynnnnyndddd¥0", 13);

```

```

        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnnnyuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnnyuudd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnyynnyuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyynnyuuuu¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    break;
case 'c':
    if(strncmp(This->p_str, "yynnnnyndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnnyndddd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "yynnnynndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnynndddd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnnnyuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnnyuudd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnnyynnuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnyynnuudd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnyynnyuuuu¥0", 13) == 0){

```

```

        strncpy(This->p_str, "nnyynnnuuuu¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
else if(strncmp(This->p_str, "nnyynnnuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyyyynnnuuuu¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    break;
case 'C':
    if(strncmp(This->p_str, "yynnnnnndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnynndddd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
else if(strncmp(This->p_str, "nnnnnnnnuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnynnuudd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
else if(strncmp(This->p_str, "nnyynnnuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyynnnuuuu¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    break;
case 'j':
    if(strncmp(This->p_str, "yynnyynndddd¥0", 13) == 0){
        strncpy(This->p_str, "nynnyynndddd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
else if(strncmp(This->p_str, "nnnnyynnuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuudd¥0", 13);

```

```

        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    break;
case 'u':
    if(strncmp(This->p_str, "nynnyynndddd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynndddd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuddd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuddd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuuud¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuuuu¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnyynnuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyyyynnuuuu¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    break;
case 'U':
    if(strncmp(This->p_str, "nnnnyynndddd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuddd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuddd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuddd¥0", 13);

```

```

        nextRun(th,667);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuud¥0", 13);
        nextRun(th,667);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuud¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuuu¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnnyynnuuu¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    break;
case 'k':
    if(strncmp(This->p_str, "nnyyyynnuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnnyynnuuu¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuddd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    break;
case 'd':
    if(strncmp(This->p_str, "nnnyynnuuu¥0", 13) == 0){

```

```

        strncpy(This->p_str, "nnnnyynnuuuu¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    if(strncmp(This->p_str, "nnnnyynnuuud¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuudd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    if(strncmp(This->p_str, "nnnnyynnuddd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynndddd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nynnyynndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnyynndddd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    break;
case 'D':
    if(strncmp(This->p_str, "nnnnyynnuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuuud¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuuud¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuudd¥0", 13);
        nextRun(th,667);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynnuddd¥0", 13);
        nextRun(th,667);
    }

```



```

        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuddd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnyynndddd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynndddd¥0", 13) == 0){
        strncpy(This->p_str, "nynnyynndddd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    break;
case 'h':
    if(strncmp(This->p_str, "nnyyyynnuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyynynnuuuu¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnyynnuddd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnyynnuddd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "yynnyynndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnynndddd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnyynynnuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyynynnuuuu¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnyynnnnuuuu¥0", 13) == 0){

```

```

    strncpy(This->p_str, "nnyynnyuuuu¥0", 13);
    WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "nnyynnyuuuu¥0", 13) == 0){
    strncpy(This->p_str, "nnyynnyuuuu¥0", 13);
    WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "nnnnnnyuudd¥0", 13) == 0){
    strncpy(This->p_str, "nnnnnnyuudd¥0", 13);
    WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "nnnnnnnyuudd¥0", 13) == 0){
    strncpy(This->p_str, "nnnnnnnyuudd¥0", 13);
    WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "nnnnnnnyuudd¥0", 13) == 0){
    strncpy(This->p_str, "nnnnnnnyuudd¥0", 13);
    WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "yynnnynndddd¥0", 13) == 0){
    strncpy(This->p_str, "yynnnynndddd¥0", 13);
    WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "yynnnnnndddd¥0", 13) == 0){
    strncpy(This->p_str, "yynnnnyyddd¥0", 13);
    WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
}
else if(strncmp(This->p_str, "yynnnnyndddd¥0", 13) == 0){

```

```

        strncpy(This->p_str, "yynnnnyyddd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    break;
case 'o':
    if(strncmp(This->p_str, "nnyynnnuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyynnnuuuu¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnyynnyuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyynnyuuuu¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnnynuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnynuudd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnnyyuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnyyuudd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "yynnnynndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnynndddd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "yynnnnyndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnnyndddd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
}

```

```
    break;
case 'O':
    if(strncmp(This->p_str, "nnyynnnnuuuu¥0", 13) == 0){
        strncpy(This->p_str, "nnyynnyuuuu¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "nnnnnnnuudd¥0", 13) == 0){
        strncpy(This->p_str, "nnnnnnyuudd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    else if(strncmp(This->p_str, "yynnnnnndddd¥0", 13) == 0){
        strncpy(This->p_str, "yynnnnyndddd¥0", 13);
        WriteString(&This->LF, "Twin2Limit.txt¥0", This->str);
    }
    break;
default:
    break;
}
return;
}
```

```
/* main.h */

#ifndef Panel_h
#define Panel_h
#include "Panel.h"
#endif

#ifndef Timer_h
#define Timer_h
#include "Timer.h"
#endif

#ifndef EV_Time_h
#define EV_Time_h
#include "EV_Time.h"
#endif

#ifndef EV_File_h
#define EV_File_h
#include "EV_File.h"
#endif

#ifndef EV_UpDown_h
#define EV_UpDown_h
#include "EV_UpDown.h"
#endif

#ifndef EV_OpenClose_h
#define EV_OpenClose_h
#include "EV_OpenClose.h"
#endif

#ifndef EV_Display_h
#define EV_Display_h
#include "EV_Display.h"
#endif

#ifndef EV_Input_h
#define EV_Input_h
#include "EV_Input.h"
#endif

#ifndef EV_Controller_h
#define EV_Controller_h
#include "EV_Controller.h"
#endif

#ifndef EV_Puls_h
#define EV_Puls_h
#include "EV_Puls.h"
#endif

#ifndef EV_Simulator_h
#define EV_Simulator_h
#include "EV_Simulator.h"
#endif
```

```
#ifndef USE_LINUX
#ifndef USE_XSERVER
#ifndef EV_Log_h
#define EV_Log_h
#include "EV_Log.h"
#endif
#endif
#endif
```

```
#ifndef USE_BCC
#ifndef EV_Queue_h
#define EV_Queue_h
#include "EV_Queue.h"
#endif
#endif
```

```
#ifndef TWIN
#ifndef EV_Twin_Queue_h
#define EV_Twin_Queue_h
#include "EV_Twin_Queue.h"
#endif
#endif
```

```
#ifndef USE_BCC
#ifndef EV_UserController_h
#define EV_UserController_h
#include "EV_UserController.h"
#endif
#endif
```

```
#ifndef TWIN
#ifndef EV_Twin_Controller_h
#define EV_Twin_Controller_h
#include "EV_Twin_Controller.h"
#endif
#endif
```

```
#ifndef USE_BCC
#ifndef EV_UserSimulator_h
#define EV_UserSimulator_h
#include "EV_UserSimulator.h"
#endif
#endif
```

```
#ifndef TWIN
#ifndef EV_Twin_Simulator_h
#define EV_Twin_Simulator_h
#include "EV_Twin_Simulator.h"
#endif
#endif
```

```
#ifndef USE_THREAD
typedef struct tag_Count
{
#ifndef USE_BCC
    int cnt[2];
#else
```

```
    int cnt[8];  
#endif  
}Count;  
#endif  
  
int main(void);
```

```
/* main.c */

#include "C.h"
#include "main.h"

#ifdef USE_THREAD
Count Cnt;
int i_cnt, j_cnt;
#else
/* 擬似スレッドの擬似インスタンス宣言 */
Thread* th[2];
/* 擬似スレッドの擬似インスタンス宣言 */
Thread* th[8];
#endif
/* 擬似スレッドの擬似インスタンス宣言 */
Thread *th1[4];
Thread *th19;
Thread *th20;
Thread *th41;
Thread *th42;
Thread *th43;
Thread *th44;
#ifdef USE_LINUX
#ifndef USE_XSERVER
Thread *th45;
#endif
#endif
#endif
```



```
#ifdef USE_BCC
```

```
Thread *th46;
```

```
Thread *th47;
```

```
Thread *th48;
```

```
#endif
```

```
#ifdef TWIN
```

```
Thread *th49;
```

```
Thread *th50;
```

```
#endif
```

```
#endif
```

```
#ifdef NOTUSE_FILES
```

```
EV_Status s;
```

```
#endif
```

```
/*=====
  入力オブジェクト宣言
  =====*/
```

```
struct EV_Input in;
```

```
/*=====
  制御オブジェクト宣言
  =====*/
```

```
struct EV_Controller cntrl;
```

```
/*=====
  送信オブジェクト宣言
  =====*/
```

```
struct EV_Puls puls;
```

```

/*=====
 シミュレータオブジェクト宣言
=====*/

struct EV_Simulator simu;

#ifdef USE_LINUX
#ifndef USE_XSERVER
/*=====
 記録オブジェクト宣言
=====*/

struct EV_Log lg;

#endif
#endif

#ifdef USE_BCC
/*=====
 3階建てオブジェクト宣言
=====*/

struct EV_UserController uc;
struct EV_UserSimulator us;
#endif

#ifdef TWIN
struct EV_Twin_Controller t1c;
struct EV_Twin_Simulator t1s;
struct EV_Twin_Controller t2c;
struct EV_Twin_Simulator t2s;
#endif

int main(void)

```

```

{
#ifdef USE_BCC
    char sw[4];

    int i;

    int j;

    int f;

    int cnt;

    static char buff[64];

#endif

#ifdef USE_THREAD
    Thread *th30;

    Thread *th31;

#endif

#ifdef USE_BCC

    for(i=0;i<0x7fff;i++) {}

    H8init(); /* H8 レジスタ初期化 */

    InitSCI(); /* SCI1初期化(serial) */

    InitLCD(); /* LCD初期化 */

    /* LED OFF */

    SetLED(0,0);

    SetLED(1,0);

    SetLED(2,0);

    SetLED(3,0);

    /*-----*/

    /* USB初期化 */

    InitUSB();

    INTC.ISCR &= (-1^0x20); /* IRQ_5 センソコントロ-ル Active Low */

```

```

INTC.IER |= 0x20; /* IRQ5 Enable */

/*-----*/

EnableInterrupt(); /* 割り込み許可 ccr */

f = 0;

PrintSCI("CPU MODE %02X\n",MDCR); /* MODE 6 */
PrintLCD("%fReady!3052"); /* %fはLCDクリアに利用 */

/* スイッチワーク初期化 */
sw[0] = sw[1] = sw[2] = sw[3] = 0;

#else

    printf("%nHello BCC");

#endif

#ifdef NOTUSE_FILES

    /* ファイル初期化 */
    new_EV_Status(&s);

#endif

#ifdef USE_THREAD

    /* タイマー初期化 */
    initWOVI();

    /* 2秒待機 */
    SleepMSec(2000);

    /* LEDTEST */
    th30 = new_Thread(30);
    th31 = new_Thread(31);
    Start(th30);
    Start(th31);
    for(;;)

```

```

{
    /* タイマー呼び出し*/
#ifdef USE_LINUX
#ifdef USE_CENTOS
    wovi(15000000.0);
#endif
#ifdef USE_RASPBIAN
    wovi(1000000.0);
#endif
#else
#ifdef USE_XSERVER
    wovi(15000000.0);
#else
    wovi();
#endif
#endif
    if(Thread_checkAllDelete() == OK)
    {
        break;
    }
}
#endif
    Clear();
    Printf(Panel,"NEXT    ");
#ifdef USE_BCC
#ifdef USE_LINUX
#ifdef USE_XSERVER
    /* system("start start.bat"); */
#endif
#endif
#endif

```

```

#endif

#endif

    /* 2秒待機 */
    SleepMSec(2000);

    Clear();

#ifdef USE_BCC

    for(;;)
    {
        /*-----*/
        /* ボタンが押された時にLCD,SCI,USBにメッセージを送る */
        for(j=0;j<4;j++)
        {
            i = GetSW(j);
            if( ((sw[j]^1)& i) ) /* sw = off->onで条件成立 */
            {
                SetLED(j,1);/* LED押した瞬間点灯 */
                sprintf(buff,"sw%u",j+1);
                PrintSCI("%s¥n",buff);
                /* NULL(0x00)まで送信 */
                write_buff(buff,strlen(buff)+1);
                PrintLCD(buff);
            }
            else SetLED(j,0);
            sw[j] = i;
        }

        /*-----*/
        /* HOSTからのシリアル入力をLCD,USBに送る */
        if(ScanSCI()) /* SCIに受信データあり? */

```

```

{
    i = GetSCI();/* シリアル入力*/
    PutLCD(i);/* LCD出力*/
    buff[0] = i;
    write_buff(buff,1);/* USB出力*/
}

```

```

/*-----*/

```

```

/* USBからデータを受信した場合、そのままHOSTへリダイレクトする */

```

```

if(get_inbufflen()) /* 受信データあり? */

```

```

{
    /* データ取得(buffサイズは64byteまで)*/
    cnt = read_buff(buff,64);
    PrintLCD("%f");/* LCDクリア*/
    PrintLCD(buff);/* LCDへ表示*/
    PrintSCI(buff);/* シリアル出力*/
    write_buff(buff,cnt); /* USBへリダイレクト*/
}

```

```

/*-----*/

```

```

/* 動作確認のため点滅*/

```

```

SetLED(3,f);

```

```

f ^= 1;

```

```

for(i=0;i<10000;i++) {} /* 適当にウエイト*/

```

```

}

```

```

#else

```

```

printf("END");

```

```

/* 5秒待機*/

```

```

SleepMSec(5000);

```

```

    return OK;
#endif
}

#ifdef USE_THREAD
/*
 * 擬似スレッドの擬似メソッド関数
 */
/* public void paint(Graphics g)の代用 */
void Repaint(void)
{
#ifdef USE_BCC
    int i;
#else
    int i,j;
#endif
    Clear();
#ifdef USE_BCC
    for(i = 0; i < Cnt.cnt[0]; i++)
    {
        Printf(Panel," ");
    }
    Printf(Panel, "<1>");
    Printf(Panel, "¥n");
    for(i = 0; i < Cnt.cnt[1]; i++)
    {
        Printf(Panel," ");
    }

```



```

    Printf(Panel, "<2>");
#else
    for(i = 0; i < 8; i++)
    {
        for(j = 0; j < Cnt.cnt[i]; j++)
        {
            printf("");
        }
        printf("<%d>", i + 1);
        for(j = 0; j < 13 - Cnt.cnt[i]; j++)
        {
            printf("");
        }
        printf("|");
        printf("¥n");
    }
#endif

    return;
}

/*
 * 疑似スレッドの疑似メソッド関数
 */
/* スレッドのpublic void run()の代用 */
void Run(Thread *This)
{
    int i;

    Thread *th1;

#ifdef USE_BCC

```

```

    char key = '¥0';

#else

    int j;

    char sw[4];

    /* スイッチワーク初期化 */

    sw[0] = sw[1] = sw[2] = sw[3] = 0;

#endif

    if(This->ID == 1)
    {
        Repaint();
    }

#ifdef USE_BCC

    Cnt.cnt[0]++;

    nextRun(This, (((rand() % 9) + 10) * 100));

#else

    if(kbhit())
    {

#ifdef USE_LINUX

        key = (char)getchar();

#else

        key = (char)getche();

#endif

    }

    if(key == 'r')
    {
        Cnt.cnt[0]++;
    }

    nextRun(This, (((rand() % 9) + 10) * 15));

    while(kbhit())
    {

```

```

#ifdef USE_LINUX
    key = (char) getchar();
#else
    key = (char) getche();
#endif
}
#endif
}
else if(This->ID == 2)
{
    Repaint();
#ifdef USE_BCC
    Cnt.cnt[1]++;
    nextRun(This, (((rand() % 9) + 10) * 100));
#else
    if(kbhit())
    {
#ifdef USE_LINUX
        key = (char) getchar();
#else
        key = (char) getche();
#endif
    }
    if(key == 'l')
    {
        Cnt.cnt[1]++;
    }
    nextRun(This, (((rand() % 9) + 10) * 15));
    while(kbhit())

```

```

    {
#ifdef USE_LINUX
        key = (char) getchar();
#else
        key = (char) getche();
#endif
    }
#endif
}

#ifdef USE_BCC
    else if(((This->ID) >= 3) && ((This->ID) <= 8))
    {
        Repaint();
        Cnt.cnt[(This->ID) - 1]++;
        nextRun(This,(((rand()% 9) + 10) * 40));
    }
#endif

    else if(This->ID == 11)
    {
        if(This->count == 1)
        {
            Clear();
            Printf(PANEL,"<1>1st  ");
            countUpNextRun(This,(1900 * 1));
        }
        else if(This->count == 2)
        {
            Clear();

```

```

        Printf(Panel,"<1>2nd");
        Printf(Panel,"<1>Stop ");
        Stop(This);
    }
else if(This->count == 3)
{
    Clear();
    Printf(Panel,"<1>3rd ");
    countUpNextRun(This(1500 * 1));
}
else if(This->count == 4)
{
    Clear();
    Printf(Panel,"<1>Stop ");
    Stop(This);
}
}
else if(This->ID == 12)
{
    if(This->count == 1)
    {
        Clear();
        Printf(Panel,"<2>1st ");
        countUpNextRun(This(1700 * 2));
    }
else if(This->count == 2)
{
    Clear();
    Printf(Panel,"<2>2nd ");

```

```

        countUpNextRun(This(1700 * 2));
    }
    else if(This->count == 3)
    {
        Clear();
        Printf(Panel,"<2>3rd    ");
        countUpNextRun(This(1700 * 2));
    }
    else
    {
        Clear();
        Printf(Panel,"<2>Stop");
        Stop(This);
        delete_(This);
    }
}
else if(This->ID == 13)
{
    if(This->count == 1)
    {
        Clear();
        Printf(Panel,"<3>1st    ");
        countUpNextRun(This(1700 * 3));
    }
    else if(This->count == 2)
    {
        Clear();
        Printf(Panel,"<3>2nd    ");
        countUpNextRun(This(1700 * 3));
    }
}

```

```

}
else if(This->count == 3)
{
    Clear();
    Printf(Panel,"<3>3rd    ");
    countUpNextRun(This(1700 * 3));
}
else
{
    Clear();
    Printf(Panel,"<3>Stop");
    Stop(This);
    delete_(This);
}
}
else if(This->ID == 14)
{
    if(This->count == 1)
    {
        Clear();
        Printf(Panel,"<4>1st    ");
        countUpNextRun(This(1500 * 4));
    }
    else if(This->count == 2)
    {
        Clear();
        Printf(Panel,"<4>2nd");
        countUpNextRun(This(1500 * 4));
    }
}

```

```

        Printf(Panel,"<1>Start ");
        th11 = Thread_Start(11);
        countUpNextRun(th11(1500 * 1));
    }
    else if(This->count == 3)
    {
        Clear();
        Printf(Panel,"<4>3rd ");
        countUpNextRun(This(1500 * 4));
    }
    else if(This->count == 4)
    {
        th11 = Thread_getThread(11);
        if(th11 != NULL)
        {
            delete_(th11);
        }

        Printf(Panel,"<4>Sto");
        Stop(This);
        delete_(This);
    }
}
else if(This->ID == 19)
{
#ifdef USE_LINUX
    nextRun(This, 4000);
#else

```



```

        nextRun(This,1);
#endif

#ifndef USE_BCC
    /* ボタンが押された時にLCD,SCI,USBにメッセージを送る */
    for(j=0;j<4;j++)
    {
        i = GetSW(j);
        if( ((sw[j]^1)& i) ) /* sw = off->onで条件成立*/
        {
            Thread_Toggle(j+ 20);
            nextRun(This,1000);
        }
    }
#else
    key = '¥0';
    key = GetChar(key);

    if(key == '1')
    {
        Thread_Toggle(21);
    }
    else if(key == '2')
    {
        Thread_Toggle(22);
    }
    else if(key == '3')
    {
        Thread_Toggle(23);
    }
}

```

```
else if(key == '4')
{
    Thread_Toggle(24);
}
else if(key == '5')
{
    Thread_Toggle(25);
}
else if(key == '6')
{
    Thread_Toggle(26);
}
else if(key == '7')
{
    Thread_Toggle(27);
}
else if(key == '8')
{
    Thread_Toggle(28);
}
else if(key == '9')
{
    Thread_Toggle(29);
}
else if(key == '0')
{
    Thread_Toggle(20);
}
```

#endif

```

}
else if(This->ID == 20)
{
    Printf(Panel,"0");
    countUpNextRun(This, 2000);
}
else if(This->ID == 21)
{
    Printf(Panel,"1");
    countUpNextRun(This, 2000);
}
else if(This->ID == 22)
{
    Printf(Panel,"2");
    countUpNextRun(This, 2000);
}
else if(This->ID == 23)
{
    Printf(Panel,"3");
    countUpNextRun(This, 2000);
}
#ifdef USE_BCC
else if((This->ID >= 24) && (This->ID <= 29))
{
    printf("%d", This->ID - 20);
    nextRun(This, 2000);
}
#endif
#endif USE_BCC

```

```

else if(This->ID == 30)
{
    if(This->count == 0)
    {
        This->count++;
        PB.DR &= 0x0e;
        nextRun(This,1000);
    }
    else if(This->count == 1)
    {
        This->count--;
        PB.DR |= 0x01;
        nextRun(This,1000);
    }
}
}

#endif

else if(This->ID == 31)
{
    if(This->count == 0)
    {
        /* 第1部分 */

#ifdef USE_BCC
        printf("\nThread Ready GO! There are 8 cources on a race.");
        printf("\nThereare 14 cells to a GOAL.");
        printf("\nFor the <1> cource, You click a 'R' button.");
        printf("\nFor the <2> cource, You click a 'L' button.");
#endif

#endif
}
}

```

```

        countUpNextRun(This0);

#else

        /* 5秒待機 */
        countUpNextRun(This5000);

#endif

    }

    else if(This->count == 1)
    {
        /* 擬似スレッド開始 */
        Printf(Pannel,"¥n");
        Printf(Pannel, "Thread Ready GO!");
        /* 2秒待機 */
        countUpNextRun(This2000);
    }

    else if(This->count == 2)
    {

#ifdef USE_BCC

        /* 擬似スレッドの擬似インスタンス初期化 */
        for(i = 0; i < 2; i++)
        {
            th[i] = new_Thread(i+ 1);
        }

#else

        /* 擬似スレッドの擬似インスタンス初期化 */
        for(i = 0; i < 8; i++)
        {
            th[i] = new_Thread(i+ 1);
        }

#endif

#endif

```

```

        countUpNextRun(This,1);
    }
    else if(This->count == 3)
    {
#ifdef USE_BCC
        if(Cnt.cnt[0] >= 13)
        {
            i_cnt = 1;
            This->count++;
        }
        else if(Cnt.cnt[1] >= 13)
        {
            i_cnt = 2;
            This->count++;
        }
#else
        i_cnt = Cnt.cnt[0];
        j_cnt = 0;
        for(i = 1; i < 8; i++)
        {
            if(i_cnt < Cnt.cnt[i])
            {
                i_cnt = Cnt.cnt[i];
                j_cnt = i;
            }
        }
        if(i_cnt >= 13) This->count++;
#endif
        nextRun(This,1);
    }
}

```

```

    }

    else if(This->count == 4)
    {
        Clear();
        if(i_cnt == 1)
        {
            Printf(Panel,"GOAL!<1>WON  ");
        }
        else if(i_cnt == 2)
        {
            Printf(Panel,"GOAL!<2>WON  ");
        }
#ifdef USE_BCC
        else
        {
            printf("GOAL!¥n<%d>WON"(j_cnt + 1));
        }
#endif
#ifdef USE_BCC
        delete_(th[0]);
        delete_(th[1]);
#else
        for(i = 0; i < 8; i++)
        {
            delete_(th[i]);
        }
#endif
#ifdef USE_LINUX
        This->count = 15;

```

```

#else
    This->count++;
#endif

    /* 2秒待機 */
    nextRun(This,2000);
}
else if(This->count == 5)
{
    Clear();
    Printf(Panel,"NEXT ");
    /* 2秒待機 */
    countUpNextRun(This,2000);
}
else if(This->count == 6)
{
    Clear();
    /* 第2部分 */
    Printf(Panel,"CountUp ");
    /* 2秒待機 */
    countUpNextRun(This,2000);
}
else if(This->count == 7)
{
    /* 疑似スレッド開始 */
    Clear();
    /* 疑似スレッドの疑似インスタンス初期化 */
    for(i = 0; i < 4; i++)
    {

```



```

        th1[i] = new_Thread(i+ 11);
    }
    countUpNextRun(This,1);
}
else if(This->count == 8)
{
    if(Thread_checkStayAnother() == 2)
    {
        This->count++;
    }
    nextRun(This,1);
}
else if(This->count == 9)
{
    /* 2秒待機 */
    countUpNextRun(This,2000);
}
else if(This->count == 10)
{
    Clear();
    Printf(Pannel,"NEXT ");
    /* 2秒待機 */
    countUpNextRun(This,2000);
}
else if(This->count == 11)
{
    Clear();
    /* 第3部分 */
    Printf(Pannel,"Toggle ");
}

```

```

        /* 2秒待機 */
        countUpNextRun(This,2000);
    }
else if(This->count == 12)
{
    th19 = new_Thread(19);
    Start(th19);
    th20 = new_Thread(20);
    Start(th20);
    countUpNextRun(This,1);
}
else if(This->count == 13)
{
    if(Thread_checkStayAnother() == 3)
    {
        delete_(th19);
        This->count++;
    }
    nextRun(This,1);
}
else if(This->count == 14)
{
    countUpNextRun(This,2000);
}
else if(This->count == 15)
{
    Clear();
    Printf(Pannel,"NEXT    ");
    /* 2秒待機 */

```

```

        countUpNextRun(This2000);
    }
else if(This->count == 16)
{
    Clear();
    /* 第4部分 */
    Printf(Pannel,"Hello EV    ");
    /* 2秒待機 */
    countUpNextRun(This2000);
}
else if(This->count == 17)
{
    /* 擬似スレッドの擬似インスタンス初期化*/
    th41 = new_Thread(41);
    th42 = new_Thread(42);
    th43 = new_Thread(43);
    th44 = new_Thread(44);

#ifdef USE_LINUX
#ifdef USE_XSERVER
    th45 = new_Thread(45);
#endif
#endif

#ifdef USE_BCC
    th47 = new_Thread(47);
    th48 = new_Thread(48);
#endif

#ifdef TWIN
    th49 = new_Thread(49);
    th50 = new_Thread(50);

```

```
#endif

    delete_(This);

    /* LEDTEST*/

    delete_(Thread_getThread(30));
}
}
else if(This->ID == 41)
{
    nextRun(This,250);
    OnInput(&in,This);
}
else if(This->ID == 42)
{
    nextRun(This, 2000);
    OnController(&cntrl, This);
}
else if(This->ID == 43)
{
    nextRun(This,110);
    OnPuls(&puls, This);
}
else if(This->ID == 44)
{
    nextRun(This, 2000);
    OnSimulator(&simu, This);
}

#ifdef USE_LINUX
```

```
#ifndef USE_XSERVER

    else if(This->ID == 45)
    {
        nextRun(This, 2000);
        OnLog(&lg,This);
    }

#endif

#endif

#ifdef USE_BCC

    else if(This->ID == 46)
    {
        if(This->count == 0)
        {
            countUpNextRun(This, 180000);
        }
        else if(This->count == 1)
        {
            delete_(This);
        }
    }

    else if(This->ID == 47)
    {
        nextRun(This, 2000);
        EV_UC_On(&uc, This);
    }

    else if(This->ID == 48)
    {
        nextRun(This, 2000);
        EV_US_On(&us, This);
    }
}
```

```

    }
#endif

#ifdef TWIN
    else if(This->ID == 49)
    {
        nextRun(This, 2000);
        EV_T1C_On(&t1c, This);
        EV_T2C_On(&t2c, This);
    }
    else if(This->ID == 50)
    {
        nextRun(This, 2000);
        EV_T1S_On(&t1s, This);
        EV_T2S_On(&t2s, This);
    }
#endif

    return;
}

/* スレッドのコンストラクタのpublic void init()の代用 */
void Init(Thread *This)
{
    if(This->ID == 1)
    {
        Cnt.cnt[0]= 0;
        nextRun(This, (((rand() % 9) + 10) * 30));
    }
    else if(This->ID == 2)
    {

```

```

    Cnt.cnt[1]= 0;
    nextRun(This, (((rand() % 9) + 10) * 30));
}
else if((This->ID >= 3) && (This->ID <= 8))
{
    Cnt.cnt[(This->ID) - 1] = 0;
    nextRun(This, (((rand() % 9) + 10) * 200));
}
else if(This->ID == 11)
{
    Printf(Panel, "<1>Init");
    countUpNextRun(This, (1500 * 1));
}
else if(This->ID == 12)
{
    Printf(Panel, "<2>Init ");
    countUpNextRun(This, (1500 * 2));
}
else if(This->ID == 13)
{
    Printf(Panel, "¥n");
    Printf(Panel, "<3>Init");
    countUpNextRun(This, (1500 * 3));
}
else if(This->ID == 14)
{
    Printf(Panel, "<4>Init ");
    countUpNextRun(This, (1500 * 4));
}

```

```

else if(This->ID == 20)
{
    Clear();
    Printf(Panel,"<0>Init    ");
    Printf(Panel, "¥n");
    countUpNextRun(This, 2000);
}
else if(This->ID == 21)
{
    Clear();
    Printf(Panel,"<1>Init    ");
    Printf(Panel, "¥n");
    countUpNextRun(This, 2000);
}
else if(This->ID == 22)
{
    Clear();
    Printf(Panel,"<2>Init    ");
    Printf(Panel, "¥n");
    countUpNextRun(This, 2000);
}
else if(This->ID == 23)
{
    Clear();
    Printf(Panel,"<3>Init    ");
    Printf(Panel, "¥n");
    countUpNextRun(This, 2000);
}

```



```

#ifdef USE_BCC

    else if((This->ID >= 24) && (This->ID <= 29))
    {
        printf("¥n<%d>Init¥n", This->ID - 20);
        nextRun(This,2000);
    }

#endif

#ifndef USE_BCC

    else if(This->ID == 30)
    {
        PB.DDR = 0xff; /* bit7..0 out */
        PB.DR |= 0xff;
    }

#endif

    else if(This->ID == 31)
    {
        This->count = 17;
        nextRun(This,0);
    }

    else if(This->ID == 41)
    {
        nextRun(This,100);
        EV_Input(&in);
    }

    else if(This->ID == 42)
    {
        nextRun(This, 1600);
        EV_Controller(&cntrl, This);
    }

```

```

else if(This->ID == 43)
{
    nextRun(This,110);
    EV_Puls(&puls, This);
}
else if(This->ID == 44)
{
    nextRun(This,600);
    EV_Simulator(&simu, This);
}
#ifdef USE_LINUX
#ifndef USE_XSERVER
else if(This->ID == 45)
{
    nextRun(This,1);
    EV_Log(&lg);
}
#endif
#endif
#ifdef USE_BCC
else if(This->ID == 46)
{
    nextRun(This,0);
}
else if(This->ID == 47)
{
    nextRun(This, 1600);
    EV_UC_Init(&uc, This);
}

```

```

else if(This->ID == 48)
{
    nextRun(This,600);
    EV_US_Init(&us, This);
}
#endif

#ifdef TWIN

else if(This->ID == 49)
{
    nextRun(This, 1100);
    EV_T1C_Init(&t1c, This);
    EV_T2C_Init(&t2c, This);
}

else if(This->ID == 50)
{
    nextRun(This,100);
    EV_T1S_Init(&t1s, This);
    EV_T2S_Init(&t2s, This);
}
#endif

return;
}

/* スレッドのデストラクタの代用 */
void Destroy(Thread *This)
{
#ifdef USE_BCC
    struct EV_File cf;
#endif
#endif

```

```

if(This->ID == 11)
{
    Clear();
    Printf(Panel, "<1>Destroy");
}
else if(This->ID == 12)
{
    Printf(Panel, "<2>Destro");
}
else if(This->ID == 13)
{
    Printf(Panel, "<3>Destro");
}
else if(This->ID == 14)
{
    Printf(Panel, "¥n");
    Printf(Panel, "<4>Destroy  ");
}
if(This->ID == 20)
{
    Clear();
    Printf(Panel, "<0>Destroy  ");
    Printf(Panel, "¥n");
}
else if(This->ID == 21)
{
    Clear();
    Printf(Panel, "<1>Destroy  ");
}

```

```

        Printf(Panel, "¥n");
    }
    else if(This->ID == 22)
    {
        Clear();
        Printf(Panel, "<2>Destroy ");
        Printf(Panel, "¥n");
    }
    else if(This->ID == 23)
    {
        Clear();
        Printf(Panel, "<3>Destroy ");
        Printf(Panel, "¥n");
    }
#ifdef USE_BCC
    else if((This->ID >= 24) && (This->ID <= 29))
    {
        printf("¥n<%d>Destroy¥n", This->ID - 20);
    }
#endif
#ifdef USE_LINUX
#ifdef USE_XSERVER
    else if(This->ID == 45)
    {
        delete_EV_Log(&lg);
    }
#endif
#endif
#endif
#ifdef USE_BCC

```

```

else if(This->ID == 46)
{
    EV_File(&cf);
    Write(&cf, "Order.txt¥0", 'q');
}
#endif

return;
}
#endif

#ifndef USE_BCC
/*=====
                                LEDコントロール
-----

int SetLED(intno,int onoff)

int    no        LEDナンバー0~3
int    onoff     0=OFF,1=ON
戻り値          以前のLEDの状態 (0=OFF,else=ON)

LEDをコントロールします。
=====*/

int SetLED(int no,int onoff)
{
    int f;
    f = PB.DR&(1<<no);
    if( onoff == 0 ) PB.DR |= (1<<no); /* off (1) */
    else PB.DR &= 0xff^(1<<no); /* on (0) */
    return(f);
}

```

```
}
```

```
/*=====
```

SW状態取得

```
-----
```

```
int GetSW(intno)
```

```
int    no        SWナンバー 0~3  
戻り値        SWの状態(0=OFF,else=ON)
```

SWの状態を取得します。

```
=====*/
```

```
int GetSW(int no)
```

```
{  
    return( ((PA.DR&(1<<no))?0:1) );  
}
```

```
/*=====
```

H8初期化

```
-----
```

BUSモードや、ポートの初期化

P1	bit1	BUS	USB A0
P3		BUS	USB D7..0
P6	bit4	BUS	USB RD
P6	bit5	BUS	USB WR
P8	bit2	BUS	USB CS
P9	bit5	BUS	USB INT(IRQ5)
P9	bit3	BUS	RS232C

P9 bit1	BUS	RS232C
PA bit0..3	IN	SW0..3
PB bit0..3	OUT	LED0..3 LCD DB4..7
PB bit4	OUT	LCD RS
PB bit7	OUT	LCD E

=====*/

```

void H8init()
{
    BSC.ABWCR = 0x06; /* 8bit BUS MODE */

    P1.DDR = 0xff; /* all OUT */
    P2.DDR = 0xff; /* all OUT */
    P2.PCR = 0x00; /* Pull up off */
    P5.DDR = 0xff; /* all OUT */
    P5.PCR = 0x00; /* Pull up off */
    P6.DDR = 0xff; /* all OUT */
    P9.DDR = 0xdf; /* Bit5 IN */
    P8.DDR = 0xff; /* all OUT */
    PA.DDR = 0xf0; /* bit7..4 out , bit3..0 in */
    PB.DDR = 0xff; /* bit7..0 out */
}

#endif

```


実行環境

```
-I"c:\borland\Bcc55\include"  
-L"c:\borland\Bcc55\lib"
```

-L"c:\borland\Bcc55\Lib"

; messagemap.src

.CPU 300HA

.SECTION V, CODE, LOCATE=H'000000

; C言語の関数 を参照

.IMPORT _main ; C言語の関数main を参照

.IMPORT _usb_int ; C言語の関数usb_int を参照

.IMPORT _InterruptITU0 ; C言語の関数InterruptITU0 を参照

;-----

; リセットベクトルの 転送先ラベルが _start になっています

; リセットベクトル

.DATA.L _start

;-----

; リセットベクトル に続く1番から60番までの 割り込みベクトル

; について、使用しない 割り込みベクトルは ラベルint_error

; に転送されます

; 割り込みベクタ

; 1 Reserved

_INT_Reserved1: .DATA.L int_error

; 2 Reserved

_INT_Reserved2: .DATA.L int_error

; 3 Reserved

_INT_Reserved3: .DATA.L int_error

; 4 Reserved

_INT_Reserved4: .DATA.L int_error

; 5 Reserved

_INT_Reserved5: .DATA.L int_error

; 6 Reserved

_INT_Reserved6: .DATA.L int_error

; 7 NMI

_INT_NMI: .DATA.L int_error

; 8 TRAP

_INT_TRAP1: .DATA.L int_error

; 9 TRAP

_INT_TRAP2: .DATA.L int_error

; 10 TRAP

_INT_TRAP3: .DATA.L int_error

; 11 TRAP

_INT_TRAP4: .DATA.L int_error

; 12 IRQ0

IRQ0: .DATA.L int_error

; 13 IRQ1

IRQ1: .DATA.L int_error

; 14 IRQ2

IRQ2: .DATA.L int_error

; 15 IRQ3

IRQ3: .DATA.L int_error

; 16 IRQ4

IRQ4: .DATA.L int_error

; 17 IRQ5

IRQ5: .DATA.L usb_interrupt ; USB割り込み

; 18 Reserved

_INT_Reserved18: .DATA.L int_error

; 19 Reserved

_INT_Reserved19: .DATA.L int_error
; 20 WOVI
_INT_WOVI: .DATA.L int_error
; 21 CMI
_INT_CMI: .DATA.L int_error
; 22 Reserved
_INT_Reserved22: .DATA.L int_error
; 23 Reserved
_INT_Reserved23: .DATA.L int_error
; 24 IMIA0
_INT_IMIA0: .DATA.L int_error
; 25 IMIB0
_INT_IMIB0: .DATA.L int_error
; タイマ0割り込み は、 ラベル_ITU_OVI_0 に転送されます
; 26 OVI0
_INT_OVI0: .DATA.L ITU_OVI_0 ; タイマ0割り込み
; 27 Reserved
_INT_Reserved27: .DATA.L int_error
; 28 IMIA1
_INT_IMIA1: .DATA.L int_error
; 29 IMIB1
_INT_IMIB1: .DATA.L int_error
; 30 OVI1
_INT_OVI1: .DATA.L int_error
; 31 Reserved
_INT_Reserved31: .DATA.L int_error
; 32 IMIA2
_INT_IMIA2: .DATA.L int_error
; 33 IMIB2

_INT_IMIB2: .DATA.L int_error

; 34 OVI2

_INT_OVI2: .DATA.L int_error

; 35 Reserved

_INT_Reserved35: .DATA.L int_error

; 36 IMIA3

_INT_IMIA3: .DATA.L int_error

; 37 IMIB3

_INT_IMIB3: .DATA.L int_error

; 38 OVI3

_INT_OVI3: .DATA.L int_error

; 39 Reserved

_INT_Reserved39: .DATA.L int_error

; 40 IMIA4

_INT_IMIA4: .DATA.L int_error

; 41 IMIB4

_INT_IMIB4: .DATA.L int_error

; 42 OVI4

_INT_OVI4: .DATA.L int_error

; 43 Reserved

_INT_Reserved43: .DATA.L int_error

; 44 DEND0A

_INT_DEND0A: .DATA.L int_error

; 45 DEND0B

_INT_DEND0B: .DATA.L int_error

; 46 DEND1A

_INT_DEND1A: .DATA.L int_error

; 47 DEND1B

_INT_DEND1B: .DATA.L int_error

; 48 Reserved

_INT_Reserved48: .DATA.L int_error

; 49 Reserved

_INT_Reserved49: .DATA.L int_error

; 50 Reserved

_INT_Reserved50: .DATA.L int_error

; 51 Reserved

_INT_Reserved51: .DATA.L int_error

; 52 ERI0

_INT_ERI0: .DATA.L int_error

; 53 RXI0

_INT_RXI0: .DATA.L int_error

; 54 TXI0

_INT_TXI0: .DATA.L int_error

; 55 TEI0

_INT_TEI0: .DATA.L int_error

; 56 ERI1

_INT_ERI1: .DATA.L int_error

; 57 RXI1

_INT_RXI1: .DATA.L int_error

; 58 TXI1

_INT_TXI1: .DATA.L int_error

; 59 TEI1

_INT_TEI1: .DATA.L int_error

; 60 ADI

_INT_ADI: .DATA.L int_error

;-----


```
.SECTION P, CODE, ALIGN=2
```

```
; _start のラベルから処理を開始
```

```
; リセットベクトルの転送先
```

```
_start:
```

```
mov.l #H'0FFFF10, er7
```

```
; 初期化付きデータを使用する場合、RAMに転送する
```

```
; message.MAP のメモリアドレス使用状況を見る
```

```
; メモリアドレスが重複するとコンパイルエラーになる
```

```
; makefile.sub も D(99C0), C(9A00) 等必要があれば合わせる
```

```
mov.l #H'99C0, er0 ; 転送元(99C0)
```

```
mov.l #H'0FFE000, er1 ; 転送先
```

```
mov.l #DATA_END, er2 ; 転送終了
```

```
init_loop:
```

```
cmp.l er1, er2
```

```
beq init_end
```

```
mov.b @er0+, r3l
```

```
mov.b r3l, @er1
```

```
inc.l #1, er1
```

```
bra init_loop
```

```
init_end:
```

```
; C言語の関数main を呼び出しています
```

```
; C言語の関数main は、void main(void); という形で、
```

```
; main.c に記述があります
```

```
jsr @_main
```

```
; 割り込み未使用
```

```
int_error:
```

```
; rte (returnと同じ意味) で終了
```

```
rte
```

```
;-----  
; USB割り込み からの転送先  
usb_interrupt:  
; スタック 退避  
push.l er0  
push.l er1  
push.l er2  
push.l er3  
push.l er4  
push.l er5  
push.l er6  
; C言語の関数usb_int を呼び出しています  
jsr @_usb_int  
; スタック 戻  
pop.l er6  
pop.l er5  
pop.l er4  
pop.l er3  
pop.l er2  
pop.l er1  
pop.l er0  
; 終了  
rte  
  
;-----  
; タイマ0割り込み からの転送先  
_ITU_OVI_0:  
; スタック 退避  
push.l er0
```

push.l er1

push.l er2

push.l er3

push.l er4

push.l er5

push.l er6

; C言語の関数InterruptITU0 を呼び出しています

; C言語の関数InterruptITU0 は void InterruptITU0(void);

; という形で、 Timer.h Timer.c に記述があります

jsr @_InterruptITU0

; スタック 戻

pop.l er6

pop.l er5

pop.l er4

pop.l er3

pop.l er2

pop.l er1

pop.l er0

; 終了

rte

; C言語から

; _EnableInterrupt (割り込み許可)

; _DisableInterrupt (割り込み禁止)

; を呼び出せるようにしています

; C言語の Panel.h に 外部参照プロトタイプ宣言 があります

; extern void EnableInterrupt(void);

; extern void DisableInterrupt(void);

```
; C言語からの呼び出し名は、
; EnableInterrupt();
; DisableInterrupt();
; です
; 割り込み許可、禁止ルーチン
.EXPORT _EnableInterrupt,_DisableInterrupt
_EnableInterrupt:
andc.b #H'3f,ccr
rts
_DisableInterrupt:
orc.b #H'c0,ccr
rts

;-----
.SECTION D,DATA

.SECTION B,DATA
DATA_END: .RES.W 1

.END
```

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;

public class EV_Input
{
    public static void main(String[] args)
    {
        EV_User_Input ui = new EV_User_Input();

        // フレームを作成してレイアウトをセット
        JFrame myFrame = new JFrame("EV_Input");
        myFrame.getContentPane().setLayout(new GridLayout(2, 2));
        // ボタンを作成
        JButton btns = new JButton("停");
        JButton btnr = new JButton("複");
        JButton btne = new JButton("震");
        JButton btnq = new JButton("切");
        // フォントを作成
        btns.setFont(new Font("DialogInput",Font.PLAIN,24));
        btnr.setFont(new Font("DialogInput",Font.PLAIN,24));
        btne.setFont(new Font("DialogInput",Font.PLAIN,24));
        btnq.setFont(new Font("DialogInput",Font.PLAIN,24));
        // ボタンがクリックされた時のイベントを定義
        btns.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent ae)

```

```

    {
        try{
            PrintWriterpw = new PrintWriter(newBufferedWriter(new
FileWriter("UserSafety.txt")));
            pw.println("s");
            pw.close();
        }catch(IOExceptionioe){}
    }
});
btnr.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent ae)
    {
        try{
            PrintWriterpw = new PrintWriter(newBufferedWriter(new
FileWriter("UserSafety.txt"));
            pw.println("Y");
            pw.close();
        }catch(IOExceptionioe){}
    }
});
btnq.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent ae)
    {
        Stringstr = "";
        try{
            BufferedReaderbr = new BufferedReader(new
FileReader("UserPermitCommand.txt"));

```

```

        str = br.readLine();
        br.close();
    }catch(IOExceptionioe){}
    if(str.equals("c"))
    {
        try{
            PrintWriterpw = new PrintWriter(newBufferedWriter(new
FileWriter("Order.txt")));
            pw.println("q");
            pw.close();
        }catch(IOExceptionioe){}
        System.exit(0);
    }
    }
});
btne.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent ae)
    {
        try{
            PrintWriterpw = new PrintWriter(newBufferedWriter(new
FileWriter("Earthquake.txt"));
            pw.println("e");
            pw.close();
        }catch(IOExceptionioe){}
    }
});
// フレームにパネルをセット
myFrame.getContentPane().add(btns);

```

```

myFrame.getContentPane().add(btnr);
myFrame.getContentPane().add(btne);
myFrame.getContentPane().add(btnq);
// フレーム(ウィンドウ)が閉じる際の処理を定義
myFrame.addWindowListener(new WindowAdapter()
{
    public void windowClosing(WindowEvent e)
    {
        String str = "";
        try{
            BufferedReaderbr = new BufferedReader(new
FileReader("UserPermitCommand.txt"));
            str = br.readLine();
            br.close();
        }catch(IOExceptionioe){}
        if(str.equals("c"))
        {
            try{
                PrintWriterpw = new PrintWriter(newBufferedWriter(new
FileWriter("Order.txt")));
                pw.println("q");
                pw.close();
            }catch(IOExceptionioe){}
        }
        System.exit(0);
    }
});
// Look & Feelの設定
try{

```



```
// WindowsスタイルのLookAndFeelに設定
UIManager.setLookAndFeel(
    "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
// 設定を反映させる
SwingUtilities.updateComponentTreeUI(myFrame);
}
// エラー処理ブロック
catch(Exceptione)
{
}
// サイズを200×200に設定してウィンドウを表示
myFrame.setSize(200, 200);
myFrame.setVisible(true);
}
}
```

```

import javax.swing.*;

import java.awt.*;

import java.awt.event.*;

import java.io.*;

import java.util.*;

public class EV_User_Input
{
    // ボタンを作成
    JButton btn3 = new JButton("3");
    JButton btn2 = new JButton("2");
    JButton btn1 = new JButton("1");
    JButton btnc = new JButton("開");
    JButton btnc = new JButton("閉");
    JButton btnc = new JButton("呼");
    JButton btn3D = new JButton("3▽");
    JButton btn2U = new JButton("2△");
    JButton btn2D = new JButton("2▽");
    JButton btn1U = new JButton("1△");
    JButton btnLONG = new JButton("延");

    public EV_User_Input()
    {
        // フレームを作成してレイアウトをセット
        JFrame myFrame_User = new JFrame("EV_Basket");
        myFrame_User.getContentPane().setLayout(new GridLayout(6, 2));

        // フォントを作成
        btn3.setFont(new Font("DialogInput",Font.PLAIN,24));
    }
}

```

```
btn2.setFont(new Font("DialogInput",Font.PLAIN,24));
btn1.setFont(new Font("DialogInput",Font.PLAIN,24));
btno.setFont(new Font("DialogInput",Font.PLAIN,24));
btnc.setFont(new Font("DialogInput",Font.PLAIN,24));
btnc.setFont(new Font("DialogInput",Font.PLAIN,24));
btn3D.setFont(new Font("DialogInput",Font.PLAIN,24));
btn2U.setFont(new Font("DialogInput",Font.PLAIN,24));
btn2D.setFont(new Font("DialogInput",Font.PLAIN,24));
btn1U.setFont(new Font("DialogInput",Font.PLAIN,24));
btnLONG.setFont(new Font("DialogInput",Font.PLAIN,24));
```

```
// ボタンがクリックされた時のイベントを定義
```

```
btn3D.addActionListener(new ActionListener()
```

```
{
```

```
    public void actionPerformed(ActionEvent ae)
```

```
    {
```

```
        String str = "";
```

```
        try{
```

```
            BufferedReaderbr = new BufferedReader(new
```

```
FileReader("UserPermitCommand.txt"));
```

```
            str = br.readLine();
```

```
            br.close();
```

```
        }catch(IOExceptionioe){}
```

```
        if(str.equals("c"))
```

```
        {
```

```
            try{
```

```
                PrintWriterpw = new PrintWriter(newBufferedWriter(new
```

```
FileWriter("UserCommand.txt"));
```

```
                pw.println("3D¥0");
```

```
                pw.close();
```

```

        }catch(IOExceptionioe){}
    }
    try{
        BufferedReaderbr = new BufferedReader(new
FileReader("UserPermitTurnOpen.txt"));
        str = br.readLine();
        br.close();
    }catch(IOExceptionioe){}
    if(str.equals("o"))
    {
        try{
            PrintWriterpw = new PrintWriter(newBufferedWriter(new
FileWriter("UserTurnOpen.txt")));
            pw.println("3D¥0");
            pw.close();
        }catch(IOExceptionioe){}
    }
}
});
btn2U.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent ae)
    {
        String str = "";
        try{
            BufferedReaderbr = new BufferedReader(new
FileReader("UserPermitCommand.txt"));
            str = br.readLine();
            br.close();

```

```

        }catch(IOExceptionioe){}

        if(str.equals("c"))
        {
            try{
                PrintWriterpw = new PrintWriter(newBufferedWriter(new
FileWriter("UserCommand.txt")));
                pw.println("2U¥0");
                pw.close();
            }catch(IOExceptionioe){}
        }
        try{
            BufferedReaderbr = new BufferedReader(new
FileReader("UserPermitTurnOpen.txt"));
            str = br.readLine();
            br.close();
        }catch(IOExceptionioe){}
        if(str.equals("o"))
        {
            try{
                PrintWriterpw = new PrintWriter(newBufferedWriter(new
FileWriter("UserTurnOpen.txt"));
                pw.println("2U¥0");
                pw.close();
            }catch(IOExceptionioe){}
        }
    }
});

btn2D.addActionListener(new ActionListener()
{

```

```

public void actionPerformed(ActionEvent ae)
{
    String str = "";
    try{
        BufferedReader br = new BufferedReader(new
FileReader("UserPermitCommand.txt"));
        str = br.readLine();
        br.close();
    }catch(IOException ioe){}
    if(str.equals("c"))
    {
        try{
            PrintWriter pw = new PrintWriter(newBufferedWriter(new
FileWriter("UserCommand.txt")));
            pw.println("2D¥0");
            pw.close();
        }catch(IOException ioe){}
    }
    try{
        BufferedReader br = new BufferedReader(new
FileReader("UserPermitTurnOpen.txt"));
        str = br.readLine();
        br.close();
    }catch(IOException ioe){}
    if(str.equals("o"))
    {
        try{
            PrintWriter pw = new PrintWriter(newBufferedWriter(new

```

```

FileWriter("UserTurnOpen.txt"));

        pw.println("2D¥0");

        pw.close();

    }catch(IOExceptionioe){}

    }

}

});

btn1U.addActionListener(new ActionListener()

{

    public void actionPerformed(ActionEvent ae)

    {

        String str = "";

        try{

            BufferedReaderbr = new BufferedReader(new

FileReader("UserPermitCommand.txt"));

            str = br.readLine();

            br.close();

        }catch(IOExceptionioe){}

        if(str.equals("c"))

        {

            try{

                PrintWriterpw = new PrintWriter(newBufferedWriter(new

FileWriter("UserCommand.txt")));

                pw.println("1U¥0");

                pw.close();

            }catch(IOExceptionioe){}

        }

        try{

            BufferedReaderbr = new BufferedReader(new

```

```

FileReader("UserPermitTurnOpen.txt"));
        str = br.readLine();
        br.close();
    }catch(IOExceptionioe){}
    if(str.equals("o"))
    {
        try{
            PrintWriterpw = new PrintWriter(newBufferedWriter(new
FileWriter("UserTurnOpen.txt")));
                pw.println("1U¥0");
                pw.close();
            }catch(IOExceptionioe){}
        }
    }
});
btn3.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent ae)
    {
        Stringstr = "";
        try{
            BufferedReaderbr = new BufferedReader(new
FileReader("UserPermitCommand.txt"));
                str = br.readLine();
                br.close();
            }catch(IOExceptionioe){}
            if(str.equals("c"))
            {
                try{

```



```

        PrintWriterpw = new PrintWriter(newBufferedWriter(new
FileWriter("UserCommand.txt")));
        pw.println("03¥0");
        pw.close();
    }catch(IOExceptionioe){}
    }
    try{
        BufferedReaderbr = new BufferedReader(new
FileReader("UserPermitTurnOpen.txt"));
        str = br.readLine();
        br.close();
    }catch(IOExceptionioe){}
    if(str.equals("o"))
    {
        try{
            PrintWriterpw = new PrintWriter(newBufferedWriter(new
FileWriter("UserTurnOpen.txt"));
            pw.println("03¥0");
            pw.close();
        }catch(IOExceptionioe){}
    }
    }
});
btn2.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent ae)
    {
        Stringstr = "";
        try{

```

```

        BufferedReaderbr = new BufferedReader(new
FileReader("UserPermitCommand.txt"));
        str = br.readLine();
        br.close();
    }catch(IOExceptionioe){}
    if(str.equals("c"))
    {
        try{
            PrintWriterpw = new PrintWriter(newBufferedWriter(new
FileWriter("UserCommand.txt")));
            pw.println("02¥0");
            pw.close();
        }catch(IOExceptionioe){}
    }
    try{
        BufferedReaderbr = new BufferedReader(new
FileReader("UserPermitTurnOpen.txt"));
        str = br.readLine();
        br.close();
    }catch(IOExceptionioe){}
    if(str.equals("o"))
    {
        try{
            PrintWriterpw = new PrintWriter(newBufferedWriter(new
FileWriter("UserTurnOpen.txt"));
            pw.println("02¥0");
            pw.close();
        }catch(IOExceptionioe){}
    }

```

```

        }
    }
});
btn1.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent ae)
    {
        String str = "";
        try{
            BufferedReader br = new BufferedReader(new
FileReader("UserPermitCommand.txt"));
            str = br.readLine();
            br.close();
        }catch(IOException ioe){}
        if(str.equals("c"))
        {
            try{
                PrintWriter pw = new PrintWriter(newBufferedWriter(new
FileWriter("UserCommand.txt")));
                pw.println("01¥0");
                pw.close();
            }catch(IOException ioe){}
        }
        try{
            BufferedReader br = new BufferedReader(new
FileReader("UserPermitTurnOpen.txt"));
            str = br.readLine();
            br.close();
        }catch(IOException ioe){}
    }
}

```

```

        if(str.equals("o"))
        {
            try{
                PrintWriterpw = new PrintWriter(newBufferedWriter(new
FileWriter("UserTurnOpen.txt")));
                pw.println("01¥0");
                pw.close();
            }catch(IOExceptionioe){}
        }
    };

    btno.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent ae)
        {
            Stringstr = "";
            try{
                BufferedReaderbr = new BufferedReader(new
FileReader("UserPermitCommand.txt"));
                str = br.readLine();
                br.close();
            }catch(IOExceptionioe){}

            if(str.equals("c"))
            {
                try{
                    PrintWriterpw = new PrintWriter(newBufferedWriter(new
FileWriter("UserCommand.txt")));
                    pw.println("0o¥0");
                    pw.close();

```

```

        }catch(IOExceptionioe){}
    }
    try{
        BufferedReaderbr = new BufferedReader(new
FileReader("UserPermitTurnOpen.txt"));
        str = br.readLine();
        br.close();
    }catch(IOExceptionioe){}
    if(str.equals("o"))
    {
        try{
            PrintWriterpw = new PrintWriter(newBufferedWriter(new
FileWriter("UserTurnOpen.txt")));
            pw.println("0o¥0");
            pw.close();
        }catch(IOExceptionioe){}
    }
}
});
btnc.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent ae)
    {
        Stringstr = "";
        try{
            BufferedReaderbr = new BufferedReader(new
FileReader("UserPermitCommand.txt"));
            str = br.readLine();
            br.close();

```

```

        }catch(IOExceptionioe){}
        if(str.equals("c"))
        {
            try{
                PrintWriterpw = new PrintWriter(newBufferedWriter(new
FileWriter("UserCommand.txt")));
                pw.println("0c¥0");
                pw.close();
            }catch(IOExceptionioe){}
        }
    }
});
btnb.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent ae)
    {
        try{
            java.awt.Toolkit.getDefaultToolkit().beep();
            PrintWriterpw1 = new PrintWriter(newBufferedWriter(new
FileWriter("Bell.txt"));
            pw1.println("b");
            pw1.close();
            PrintWriterpw2 = new PrintWriter(newBufferedWriter(new
FileWriter("Bell.txt"));
            Thread.sleep(10000);
            pw2.println("N");
            pw2.close();
        }catch(Exceptione){}
    }
}

```

```

    }
});
btnLONG.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent ae)
    {
        String str = "";
        try{
            BufferedReader br = new BufferedReader(new
FileReader("UserPermitCommand.txt"));
            str = br.readLine();
            br.close();
        }catch(IOException ioe){}
        if(str.equals("c"))
        {
            try{
                PrintWriter pw = new PrintWriter(newBufferedWriter(new
FileWriter("UserCommand.txt")));
                pw.println("0L¥0");
                pw.close();
            }catch(IOException ioe){}
        }
        try{
            BufferedReader br = new BufferedReader(new
FileReader("UserPermitTurnOpen.txt"));
            str = br.readLine();
            br.close();
        }catch(IOException ioe){}
        if(str.equals("o"))

```

```

        {
            try{
                PrintWriterpw = new PrintWriter(newBufferedWriter(new
FileWriter("UserTurnOpen.txt")));
                pw.println("0L¥0");
                pw.close();
            }catch(IOExceptionioe){}
        }
    }
});
// フレームにパネルをセット
myFrame_User.getContentPane().add(btn3D);
myFrame_User.getContentPane().add(btnb);
myFrame_User.getContentPane().add(btn2U);
myFrame_User.getContentPane().add(btn3);
myFrame_User.getContentPane().add(btn2D);
myFrame_User.getContentPane().add(btn2);
myFrame_User.getContentPane().add(btn1U);
myFrame_User.getContentPane().add(btn1);
myFrame_User.getContentPane().add(btno);
myFrame_User.getContentPane().add(btnc);
myFrame_User.getContentPane().add(btnLONG);
// フレーム(ウィンドウ)が閉じる際の処理を定義
myFrame_User.addWindowListener(new WindowAdapter()
{
    public void windowClosing(WindowEvent e)
    {
        String str = "";
        try{

```



```

        BufferedReader br = new BufferedReader(new
FileReader("UserPermitCommand.txt"));

        str = br.readLine();

        br.close();

    } catch (IOException ioe) {}

    if (str.equals("c"))
    {

        try {

            PrintWriter pw = new PrintWriter(new BufferedWriter(new
FileWriter("Order.txt")));

            pw.println("q");

            pw.close();

        } catch (IOException ioe) {}

    }

    System.exit(0);

}

});

// Look & Feelの設定
try {

    // WindowsスタイルのLookAndFeelに設定
    UIManager.setLookAndFeel(

        "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");

    // 設定を反映させる
    SwingUtilities.updateComponentTreeUI(myFrame_User);

}

// エラー処理ブロック
catch (Exception e)

{

}

```

```
// サイズを200×500に設定してウィンドウを表示
```

```
myFrame_User.setSize(200, 500);
```

```
myFrame_User.setVisible(true);
```

```
}
```

```
}
```

```
import java.io.*;
```

```
public class EV_Simulator_JFile
```

```
{
```

```
    public synchronized static String Read(String Message, String FileName)
```

```
    {
```

```
        try{
```

```
            BufferedReader br = new BufferedReader(new FileReader(FileName));
```

```
            Message = br.readLine();
```

```
            br.close();
```

```
        }
```

```
        catch(IOException e)
```

```
        {
```

```
            System.out.println("読み込み失敗");
```

```
        }
```

```
        catch(NullPointerException e)
```

```
        {
```

```
            System.out.println("JFile Read String " + FileName + " 読み込み失敗");
```

```
        }
```

```
        return Message;
```

```
    }
```

```
    public synchronized static void Write(String Message, String FileName)
```

```
    {
```

```
        try{
```

```
            PrintWriter pw = new PrintWriter(new BufferedWriter(new FileWriter(FileName)));
```

```
            pw.print(Message);
```

```

        pw.close();
    }
    catch(IOException e)
    {
        System.out.println("書き込み失敗");
    }
}

public synchronized static char Read(char Message, String FileName)
{
    try{
        BufferedReader br = new BufferedReader(new FileReader(FileName));
        String str = br.readLine();
        if(!str.equals("")){
            Message = str.charAt(0);
        }
        br.close();
    }
    catch(IOException e)
    {
        System.out.println("読み込み失敗");
    }
    catch(NullPointerException e)
    {
        System.out.println("JFile Read Char " + FileName + " 読み込み失敗");
    }

    return Message;
}

public synchronized static void Write(char Message, String FileName)

```

```
{
    try{
        PrintWriter pw = new PrintWriter(new BufferedWriter(new FileWriter(fileName)));
        String str1 = "";
        StringBuffer sb = new StringBuffer(str1);
        sb = sb.insert(0,Message);
        String str2 = sb.toString();
        pw.print(str2);
        pw.close();
    }
    catch(IOException e)
    {
        System.out.println("書き込み失敗");
    }
}

public static void debug(String str)
{
    System.out.println("Debug " + str);
}
}
```

```
import javax.swing.*;
import java.awt.Graphics;
import java.awt.Color;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class EV_Simulator_Display extends JPanel implements Runnable
{
    int L_x;
    int R_x;
    int y;
    int width;
    int height;
    char Safety;
    char Motor;
    String Limit;
    char PermitTurnOpen;
    char TurnOpen;

    public EV_Simulator_Display()
    {
        width = 100;
        height = 100;
        Safety();
        Limit();
        PermitTurnOpen = 'N';
    }
}
```

```
TurnOpen= 'N';
```

```
addMouseListener(new MouseAdapter()
```

```
{
```

```
    // ○ void型マウスのクリックに反応させる
```

```
    public void mouseClicked(MouseEvent e)
```

```
    {
```

```
        if((e.getX()> L_x + 60) && (e.getX()< R_x)
```

```
            && (e.getY()> y) && (e.getY()< y + 120))
```

```
        {
```

```
            //ReadPermitTurnOpen();
```

```
            PermitTurnOpen = EV_Simulator_JFile.Read(PermitTurnOpen,
```

```
"UserPermitTurnOpen.txt");
```

```
            if(PermitTurnOpen== 'o')
```

```
            {
```

```
                //TurnOpen= 'o';
```

```
                //WriteTurnOpen();
```

```
                EV_Simulator_JFile.Write("0o¥0", "UserTurnOpen.txt");
```

```
            }
```

```
        }
```

```
        PermitTurnOpen= 'N';
```

```
        TurnOpen= 'N';
```

```
    }
```

```
});
```

```
}
```

```
private void Safety()
```

```
{
```

```
    try{
```

```

while(true)
{
    ReadSafety();
    if(Safety== 'r') break;
    else if(Safety== 'Y'){
        Thread.sleep(1000);
        ReadSafety();
        if(Safety== 'Y'){
            Safety= 'r';
            WriteSafety();
            break;
        }
    }
}
}
catch(InterruptedException e){}
}

private void Limit()
{
    try{
        ReadMotor();
        ReadLimit();
        switch(Motor)
        {
            case 's':
                if(Limit.equals("yynnyynndddd")){
                    L_x = 60;
                    R_x = 120;
                    y = 480;

```



```
        repaint();
        Thread.sleep(1000);
    }
else if(Limit.equals("yynnnnyydddd")){
    L_x = 0;
    R_x = 180;
    y = 480;
    repaint();
    Thread.sleep(1000);
}
else if(Limit.equals("nnnnyynnuudd")){
    L_x = 60;
    R_x = 120;
    y = 240;
    repaint();
    Thread.sleep(1000);
}
else if(Limit.equals("nnnnnnyyuudd")){
    L_x = 0;
    R_x = 180;
    y = 240;
    repaint();
    Thread.sleep(1000);
}
else if(Limit.equals("nnyyyynnuuuu")){
    L_x = 60;
    R_x = 120;
    y = 0;
    repaint();
```

```

        Thread.sleep(1000);
    }
    else if(Limit.equals("nnyynnyuuuu")){
        L_x = 0;
        R_x = 180;
        y = 0;
        repaint();
        Thread.sleep(1000);
    }
    break;
case 't':
    if(Limit.equals("yynnnnyddddd")){
        //Limit= "yynnnnyddddd";
        //WriteLimit();
        Thread.sleep(1000);
    }
    else if(Limit.equals("nnnnnnyuudd")){
        //Limit= "nnnnnnyuudd";
        //WriteLimit();
        Thread.sleep(1000);
    }
    else if(Limit.equals("nnyynnyuuuu")){
        //Limit= "nnyynnyuuuu";
        //WriteLimit();
        Thread.sleep(1000);
    }
    else{
        Thread.sleep(1000);
    }

```

```

    }
    break;
case 'c':
    if(Limit.equals("yynnnnyndddd")){
        //Limit= "yynnnnnndddd";
        //WriteLimit();
        for(int j = 0; j < 15; j++){
            L_x = j;
            R_x = 180 - j;
            y = 480;
            repaint();
            Thread.sleep(66);
        }
    }
    else if(Limit.equals("yynnyyndddd")){
        //Limit= "yynnyyndddd";
        //WriteLimit();
        for(int j = 45; j < 60; j++){
            L_x = j;
            R_x = 180 - j;
            y = 480;
            repaint();
            Thread.sleep(66);
        }
    }
    else if(Limit.equals("nnnnnnynuudd")){
        //Limit= "nnnnnnnuudd";
        //WriteLimit();
        for(int j = 0; j < 15; j++){

```

```

        L_x = j;
        R_x = 180 - j;
        y = 240;
        repaint();
        Thread.sleep(66);
    }
}

else if(Limit.equals("nnnnnyynnuudd")){
    //Limit= "nnnnnyynnuudd";
    //WriteLimit();
    for(int j = 45; j < 60; j++){
        L_x = j;
        R_x = 180 - j;
        y = 240;
        repaint();
        Thread.sleep(66);
    }
}

else if(Limit.equals("nnyynnnynuuuu")){
    //Limit= "nnyynnnynuuuu";
    //WriteLimit();
    for(int j = 0; j < 15; j++){
        L_x = j;
        R_x = 180 - j;
        y = 0;
        repaint();
        Thread.sleep(66);
    }
}
}

```

```

else if(Limit.equals("nnyynnnuuuu")){
    //Limit= "nnyyyynnnuuuu";
    //WriteLimit();
    for(int j = 45; j < 60; j++){
        L_x = j;
        R_x = 180 - j;
        y = 0;
        repaint();
        Thread.sleep(66);
    }
}
else{
    Thread.sleep(1000);
}
break;
case 'C':
    if(Limit.equals("yynnnnnndddd")){
        //Limit= "yynnnynndddd";
        //WriteLimit();
        for(int j = 15; j < 45; j++){
            L_x = j;
            R_x = 180 - j;
            y = 480;
            repaint();
            Thread.sleep(33);
        }
    }
    else if(Limit.equals("nnnnnnnnuudd")){
        //Limit= "nnnnnynnuudd";

```

```

        //WriteLimit();
        for(int j = 15; j < 45; j++){
            L_x = j;
            R_x = 180 - j;
            y = 240;
            repaint();
            Thread.sleep(33);
        }
    }
    else if(Limit.equals("nnyynnnuuuu")){
        //Limit= "nnyynnnuuuu";
        //WriteLimit();
        for(int j = 15; j < 45; j++){
            L_x = j;
            R_x = 180 - j;
            y = 0;
            repaint();
            Thread.sleep(33);
        }
    }
    else{
        Thread.sleep(1000);
    }
    break;
case 'j':
    if(Limit.equals("yynnyynndddd")){
        //Limit= "nynnyynndddd";
        //WriteLimit();

```

```

        Thread.sleep(1000);
    }
    else if(Limit.equals("nnnnyynnuudd")){
        //Limit= "nnnnyynnuudd";
        //WriteLimit();
        Thread.sleep(1000);
    }
    else{
        Thread.sleep(1000);
    }
    break;
case 'u':
    if(Limit.equals("nynnyynndddd")){
        //Limit= "nnnnyynndddd";
        //WriteLimit();
        for(int i = 480; i > 420; i--){
            L_x = 60;
            R_x = 120;
            y = i;
            repaint();
            Thread.sleep(16);
        }
    }
    else if(Limit.equals("nnnnyynnuudd")){
        //Limit= "nnnnyynnuudd";
        //WriteLimit();
        for(int i = 300; i > 240; i--){
            L_x = 60;
            R_x = 120;

```

```

        y = i;
        repaint();
        Thread.sleep(16);
    }
}
else if(Limit.equals("nnnnyynnuud")){
    //Limit= "nnnnyynnuuuu";
    //WriteLimit();
    for(int i = 240; i > 180; i--){
        L_x = 60;
        R_x = 120;
        y = i;
        repaint();
        Thread.sleep(16);
    }
}
else if(Limit.equals("nnynyynnuuuu")){
    //Limit= "nnyyyynnuuuu";
    //WriteLimit();
    for(int i = 60; i > 0; i--){
        L_x = 60;
        R_x = 120;
        y = i;
        repaint();
        Thread.sleep(16);
    }
}
else{
    Thread.sleep(1000);
}

```



```

    }
    break;
case 'U':
    if(Limit.equals("nnnnyynnndddd")){
        //Limit= "nnnnyynnuddd";
        //WriteLimit();
        for(int i = 420; i > 300; i--){
            L_x = 60;
            R_x = 120;
            y = i;
            repaint();
            Thread.sleep(8);
        }
    }
    else if(Limit.equals("nnnnyynnuddd")){
        //Limit= "nnnnyynnuddd";
        //WriteLimit();
        for(int i = 300; i > 240; i--){
            L_x = 60;
            R_x = 120;
            y = i;
            repaint();
            Thread.sleep(8);
        }
    }
    else if(Limit.equals("nnnnyynnudd")){
        //Limit= "nnnnyynnudd";
        //WriteLimit();
    }
}

```

```

else if(Limit.equals("nnnnyynnuud")){
    //Limit= "nnnnyynnuuuu";
    //WriteLimit();
    for(int i = 240; i > 180; i--){
        L_x = 60;
        R_x = 120;
        y = i;
        repaint();
        Thread.sleep(8);
    }
}
else if(Limit.equals("nnnnyynnuuuu")){
    //Limit= "nnnyynnuuuu";
    //WriteLimit();
    for(int i = 180; i > 60; i--){
        L_x = 60;
        R_x = 120;
        y = i;
        repaint();
        Thread.sleep(8);
    }
}
else{
    Thread.sleep(1000);
}
break;
case 'k':
    if(Limit.equals("nnyyyynnuuuu")){

```

```

        //Limit= "nnnyynnuuuu";
        //WriteLimit();
        Thread.sleep(1000);
    }
    else if(Limit.equals("nnnnyynnuudd")){
        //Limit= "nnnnyynnuudd";
        //WriteLimit();
        Thread.sleep(1000);
    }
    else{
        Thread.sleep(1000);
    }
    break;
case 'd':
    if(Limit.equals("nnnyynnuuuu")){
        //Limit= "nnnnyynnuuuu";
        //WriteLimit();
        for(int i = 0; i < 60; i++){
            L_x = 60;
            R_x = 120;
            y = i;
            repaint();
            Thread.sleep(16);
        }
    }
    if(Limit.equals("nnnnyynnuudd")){
        //Limit= "nnnnyynnuudd";
        //WriteLimit();
        for(int i = 180; i < 240; i++){

```

```

        L_x = 60;
        R_x = 120;
        y = i;
        repaint();
        Thread.sleep(16);
    }
}

if(Limit.equals("nnnnyynnuddd")){
    //Limit= "nnnnyynnndddd";
    //WriteLimit();
    for(int i = 240; i < 300; i++){
        L_x = 60;
        R_x = 120;
        y = i;
        repaint();
        Thread.sleep(16);
    }
}

else if(Limit.equals("nynnnyynnndddd")){
    //Limit= "yynnnyynnndddd";
    //WriteLimit();
    for(int i = 420; i < 480; i++){
        L_x = 60;
        R_x = 120;
        y = i;
        repaint();
        Thread.sleep(16);
    }
}
}

```

```

else{
    Thread.sleep(1000);
}
break;
case 'D':
    if(Limit.equals("nnnnyynnuuuu")){
        //Limit= "nnnnyynnuuud";
        //WriteLimit();
        for(int i = 60; i < 180; i++){
            L_x = 60;
            R_x = 120;
            y = i;
            repaint();
            Thread.sleep(8);
        }
    }
    else if(Limit.equals("nnnnyynnuuud")){
        //Limit= "nnnnyynnuudd";
        //WriteLimit();
        for(int i = 180; i < 240; i++){
            L_x = 60;
            R_x = 120;
            y = i;
            repaint();
            Thread.sleep(8);
        }
    }
    else if(Limit.equals("nnnnyynnuudd")){
        //Limit= "nnnnyynnuddd";

```

```

        //WriteLimit();
    }
    else if(Limit.equals("nnnnyynnuddd")){
        //Limit= "nnnnyynndddd";
        //WriteLimit();
        for(int i = 240; i < 300; i++){
            L_x = 60;
            R_x = 120;
            y = i;
            repaint();
            Thread.sleep(8);
        }
    }
    else if(Limit.equals("nnnnyynndddd")){
        //Limit= "nynnyynndddd";
        //WriteLimit();
        for(int i = 300; i < 420; i++){
            L_x = 60;
            R_x = 120;
            y = i;
            repaint();
            Thread.sleep(8);
        }
    }
    else{
        Thread.sleep(1000);
    }
    break;

```

case 'h':

```
if(Limit.equals("nnyyyynnuuuu")){
    //Limit= "nnyynnuuuu";
    //WriteLimit();
    Thread.sleep(1000);
}
else if(Limit.equals("nnnnyynnuudd")){
    //Limit= "nnnnynnuudd";
    //WriteLimit();
    Thread.sleep(1000);
}
else if(Limit.equals("yynnyynndddd")){
    //Limit= "yynnnynndddd";
    //WriteLimit();
    Thread.sleep(1000);
}
else if(Limit.equals("nnyynnuuuu")){
    //Limit= "nnyynnuuuu";
    //WriteLimit();
    for(int j = 15; j < 45; j++){
        L_x = 60 - j;
        R_x = 120 + j;
        y = 0;
        repaint();
        Thread.sleep(33);
    }
}
else if(Limit.equals("nnyynnnnuuuu")){
    //Limit= "nnyynnyuuuu";
```

```

//WriteLimit();
for(int j = 45; j < 60; j++){
    L_x = 60 - j;
    R_x = 120 + j;
    y = 0;
    repaint();
    Thread.sleep(66);
}
}
else if(Limit.equals("nnyynnyuuuu")){
    //Limit= "nnyynnyuuuu";
    //WriteLimit();
    Thread.sleep(1000);
}
else if(Limit.equals("nnnnnnyuuudd")){
    //Limit= "nnnnnnyuuudd";
    //WriteLimit();
    for(int j = 15; j < 45; j++){
        L_x = 60 - j;
        R_x = 120 + j;
        y = 240;
        repaint();
        Thread.sleep(33);
    }
}
else if(Limit.equals("nnnnnnnyuuudd")){
    //Limit= "nnnnnnnyuuudd";
    //WriteLimit();
    for(int j = 45; j < 60; j++){

```



```

        L_x = 60 - j;
        R_x = 120 + j;
        y = 240;
        repaint();
        Thread.sleep(66);
    }
}
else if(Limit.equals("nnnnnnyuudd")){
    //Limit= "nnnnnnyuudd";
    //WriteLimit();
    Thread.sleep(1000);
}
else if(Limit.equals("yynnnynndddd")){
    //Limit= "yynnnynndddd";
    //WriteLimit();
    for(intj = 15; j < 45; j++){
        L_x = 60 - j;
        R_x = 120 + j;
        y = 480;
        repaint();
        Thread.sleep(33);
    }
}
else if(Limit.equals("yynnnnnyddddd")){
    //Limit= "yynnnnnyddddd";
    //WriteLimit();
    for(intj = 45; j < 60; j++){
        L_x = 60 - j;
        R_x = 120 + j;

```

```

        y = 480;
        repaint();
        Thread.sleep(66);
    }
}
else if(Limit.equals("yynnnnyndddd")){
    //Limit= "yynnnnyyddd";
    //WriteLimit();
    Thread.sleep(1000);
}
else{
    Thread.sleep(1000);
}
break;
case 'o':
    if(Limit.equals("nnyynnnuuuu")){
        //Limit= "nnyynnnuuuu";
        //WriteLimit();
        for(int j = 0; j < 15; j++){
            L_x = 60 - j;
            R_x = 120 + j;
            y = 0;
            repaint();
            Thread.sleep(66);
        }
    }
    else if(Limit.equals("nnyynnyuuuu")){
        //Limit= "nnyynnyuuuu";

```

```

//WriteLimit();
for(intj = 45;j < 60;j++){
    L_x = 60 - j;
    R_x = 120 + j;
    y = 0;
    repaint();
    Thread.sleep(66);
}
}
else if(Limit.equals("nnnnnynnuudd")){
    //Limit= "nnnnnnnuudd";
    //WriteLimit();
    for(intj = 0;j < 15;j++){
        L_x = 60 - j;
        R_x = 120 + j;
        y = 240;
        repaint();
        Thread.sleep(66);
    }
}
else if(Limit.equals("nnnnnynnuudd")){
    //Limit= "nnnnnnyyuudd";
    //WriteLimit();
    for(intj = 45;j < 60;j++){
        L_x = 60 - j;
        R_x = 120 + j;
        y = 240;
        repaint();
        Thread.sleep(66);
    }
}

```

```

        }
    }
    else if(Limit.equals("yynnnynndddd")){
        //Limit= "yynnnnnndddd";
        //WriteLimit();
        for(intj = 0;j < 15;j++){
            L_x = 60 - j;
            R_x = 120 + j;
            y = 480;
            repaint();
            Thread.sleep(66);
        }
    }
    else if(Limit.equals("yynnnnyndddd")){
        //Limit= "yynnnnyyddddd";
        //WriteLimit();
        for(intj = 45;j < 60;j++){
            L_x = 60 - j;
            R_x = 120 + j;
            y = 480;
            repaint();
            Thread.sleep(66);
        }
    }
    else{
        Thread.sleep(1000);
    }
    break;
case 'O':

```

```

if(Limit.equals("nnyynnnuuuu")){
    //Limit= "nnyynnyuuuu";
    //WriteLimit();
    for(intj = 15;j < 45;j++){
        L_x = 60 - j;
        R_x = 120 + j;
        y = 0;
        repaint();
        Thread.sleep(33);
    }
}

else if(Limit.equals("nnnnnnnuudd")){
    //Limit= "nnnnnnyuudd";
    //WriteLimit();
    for(intj = 15;j < 45;j++){
        L_x = 60 - j;
        R_x = 120 + j;
        y = 240;
        repaint();
        Thread.sleep(33);
    }
}

else if(Limit.equals("yynnnnnndddd")){
    //Limit= "yynnnnyndddd";
    //WriteLimit();
    for(intj = 15;j < 45;j++){
        L_x = 60 - j;
        R_x = 120 + j;
        y = 480;
    }
}

```

```

        repaint();
        Thread.sleep(33);
    }
}
else{
    Thread.sleep(1000);
}
break;
default:
    Thread.sleep(1000);
    break;
}
}
catch(InterruptedException e){}
}
public void run()
{
    while(true)
    {
        Safety();
        Limit();
    }
}
public void paintComponent(Graphics g)
{
    g.setColor(Color.green);
    g.fillRect(0, 0, width, height);
    g.setColor(Color.lightGray);

```

```

    g.fillRect(L_x + 60, y, R_x - L_x - 60, 120);
    g.setColor(Color.yellow);
    g.fillRect(L_x, y, 60, 120);
    g.fillRect(R_x, y, 60, 120);
    g.setColor(Color.black);
    g.drawRect(L_x + 60, y, R_x - L_x - 60, 120);
    g.drawRect(L_x, y, 60, 120);
    g.drawRect(R_x, y, 60, 120);
}

public void setmySize(int w, int h)
{
    if(width > 0 && width < 1000)
    {
        width = w;
    }
    if(height > 0 && height < 500)
    {
        height = h;
    }
}

public void debug(int i)
{
    System.out.println("Debug " + i);
}

private void ReadSafety()
{
    Safety = EV_Simulator_JFile.Read(Safety, "UserSafety.txt");
}

private void WriteSafety()

```

```

{
    EV_Simulator_JFile.Write(Safety, "UserSafety.txt");
}

private void ReadMotor()
{
    Motor = EV_Simulator_JFile.Read(Motor, "UserMotor.txt");
}

private void ReadLimit()
{
    Limit = EV_Simulator_JFile.Read(Limit, "UserLimit.txt");
}

private void WriteLimit()
{
    EV_Simulator_JFile.Write(Limit, "UserLimit.txt");
}

private void ReadPermitTurnOpen()
{
    PermitTurnOpen = EV_Simulator_JFile.Read(PermitTurnOpen, "UserPermitTurnOpen.txt");
}

private void WriteTurnOpen()
{
    EV_Simulator_JFile.Write(TurnOpen, "UserTurnOpen.txt");
    EV_Simulator_JFile.Write("0o¥0", "UserTurnOpen.txt");
}

public void Wait_ms(int ms)
{
    try{
        Thread.sleep(ms);
    }
}

```



```
catch(InterruptedException e){}
```

```
}
```

```
}
```

```

import java.awt.*;
import javax.swing.*;

public class EV_Simulator
{
    public static void main(String args[])
    {
        EV_Simulator_JFile.Write("N__\n", "Order.txt");
        EV_Simulator_JFile.Write("N\n", "Command.txt");
        int width = 260;
        int height = 650;
        JFrame myFrame = new JFrame("EV_Simulator");
        EV_Simulator_Display d = new EV_Simulator_Display();
        myFrame.getContentPane().add(d);
        myFrame.setSize(width, height);
        myFrame.setVisible(true);
        Thread th = new Thread(d);
        d.setmySize(width, height);
        th.start();

        char Command = '\0';
        while(true)
        {
            Command = EV_Simulator_JFile.Read(Command, "Command.txt");
            if(Command == 'q') break;
            d.Wait_ms(2000);
        }
        EV_Simulator_JFile.Write("N__\n", "Order.txt");
        System.exit(0);
    }
    public static void debug(String str)
    {
        System.out.println("Debug " + str);
    }
}

```

drop database if exists ev001;

create database ev001;

use ev001;

GRANT SELECT ON ev001.* TO pi@localhost IDENTIFIED BY 'raspberry';

GRANT UPDATE ON ev001.* TO pi@localhost IDENTIFIED BY 'raspberry';

GRANT INSERT ON ev001.* TO pi@localhost IDENTIFIED BY 'raspberry';

GRANT DELETE ON ev001.* TO pi@localhost IDENTIFIED BY 'raspberry';

GRANT SELECT ON ev001.* TO pi@'192.168.115.%' IDENTIFIED BY 'raspberry';

GRANT UPDATE ON ev001.* TO pi@'192.168.115.%' IDENTIFIED BY 'raspberry';

GRANT INSERT ON ev001.* TO pi@'192.168.115.%' IDENTIFIED BY 'raspberry';

GRANT DELETE ON ev001.* TO pi@'192.168.115.%' IDENTIFIED BY 'raspberry';

GRANT SELECT ON ev001.* TO pi@'192.168.43.%' IDENTIFIED BY 'raspberry';

GRANT UPDATE ON ev001.* TO pi@'192.168.43.%' IDENTIFIED BY 'raspberry';

GRANT INSERT ON ev001.* TO pi@'192.168.43.%' IDENTIFIED BY 'raspberry';

GRANT DELETE ON ev001.* TO pi@'192.168.43.%' IDENTIFIED BY 'raspberry';

GRANT SELECT ON LAA0034274-ev001.* TO LAA0034274@'mysql144.phy.lolipop.lan' IDENTIFIED BY 'raspberry';

GRANT UPDATE ON LAA0034274-ev001.* TO LAA0034274@'mysql144.phy.lolipop.lan' IDENTIFIED BY 'raspberry';

GRANT INSERT ON LAA0034274-ev001.* TO LAA0034274@'mysql144.phy.lolipop.lan' IDENTIFIED BY 'raspberry';

GRANT DELETE ON LAA0034274-ev001.* TO LAA0034274@'mysql144.phy.lolipop.lan' IDENTIFIED BY 'raspberry';

```
GRANT SELECT ON hidemine_ev001.* TO hidemine_pi@'mysql10007.xserver.jp' IDENTIFIED BY 'raspberrypi';
```

```
GRANT UPDATE ON hidemine_ev001.* TO hidemine_pi@'mysql10007.xserver.jp' IDENTIFIED BY 'raspberrypi';
```

```
GRANT INSERT ON hidemine_ev001.* TO hidemine_pi@'mysql10007.xserver.jp' IDENTIFIED BY 'raspberrypi';
```

```
GRANT DELETE ON hidemine_ev001.* TO hidemine_pi@'mysql10007.xserver.jp' IDENTIFIED BY 'raspberrypi';
```

```
DROP TABLE IF EXISTS s_log;
```

```
CREATE TABLE s_log (
```

```
lg_ymd decimal(8) not null comment '日付',
```

```
lg_hms decimal(6) not null comment '時刻',
```

```
lg_safety varchar(1) not null comment '安全スイッチ',
```

```
lg_limit varchar(9) not null comment 'リミットスイッチ',
```

```
lg_command varchar(1) not null comment '命令入力',
```

```
lg_permitcommand varchar(1) not null comment '命令許可',
```

```
lg_permitturnopen varchar(1) not null comment '反転開許可',
```

```
lg_motor varchar(1) not null comment 'モーター出力',
```

```
PRIMARY KEY(lg_ymd,lg_hms)
```

```
)
```

```
ENGINE = INNODB
```

```
Default Charset = UTF8
```

```
COLLATE = UTF8_BIN
```

```
COMMENT='ログ'
```

```
;
```

```
DROP TABLE IF EXISTS s_info;
```

```
CREATE TABLE s_info (  
no decimal(1) not null comment '行番号',  
ch_order varchar(1) not null comment '命令入力',  
ch_motor varchar(1) not null comment 'モーター出力',  
str_limit varchar(9) not null comment 'リミット出力',  
PRIMARY KEY(no)  
)  
  
ENGINE = INNODB  
  
Default Charset = UTF8  
  
COLLATE = UTF8_BIN  
  
COMMENT='ログ'  
;
```

```
insert into s_info (no, ch_order, ch_motor, str_limit) values (1, 'N', 's', "yynnyynn¥0");
```

An EV is stopped emergency!

```

<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
</head>
<body>
<?php
if(($fp = @fopen("Command.txt", "r")) == false){
}
else{
    if(($tmpChC = fgets($fp)) == true){
        $chC = str_split($tmpChC);
    }
    fclose($fp);
}
if(($fp = @fopen("Order.txt", "r")) == false){
}
else{
    if(($tmpChO = fgets($fp)) == true){
        $chO = str_split($tmpChO);
    }
    fclose($fp);
}
if((strcmp($chC[0], 'q') == 0) && (strcmp($chO[0], 'q') != 0) && (strcmp($chO[0], 'N') != 0) &&
(strcmp($chO[1], 'N') != 0)){
    //exec('sudo /home/pi/ダウンロード/ev9/RaspberryPi3ModelBExe.out'); // for RaspberryPi
    //exec('./XServerExe.out'); // for XServer
    exec('main.exe'); // for Windows
}
?>
<form method="get" action="./ev_power.php" name="myform">
<script type="text/javascript">
<!--
setInterval("document.myform.submit()",2000);
//-->
</script>
</form>
</body>
</html>

```

```

<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">

<?php
/* 表示を表す列挙体宣言 */
const Panel = 1;
const ClsPnl = 2;
const InputCommand = 3;
const Monitor = 4;

function myPrintF($mode, $str)
{
    switch($mode)
    {
        case ClsPnl:
            break;
        case Panel:
            ?><br><?php
            echo $str;
            break;
        case InputCommand:
            ?><br><?php
            echo $str;
            break;
        case Monitor:
            ?><br><?php
            echo $str;
            break;
        default:
            break;
    }
    return;
}

/* EV_Display.c */

function WriteFL($chO)
{
    if(($fp = @fopen("Order.txt", "w")) == false){
    }
    else if(fwrite($fp, $chO)){
        fclose($fp);
    }
    else{
        fclose($fp);
    }
}

function DispInput()
{
    /* 入力指示 */
    myPrintF(InputCommand, "\nUP = 'u', DOWN = 'd', OPEN = 'o', CLOSE = 'c'");
    myPrintF(InputCommand, "\nEMERGENCY = 's', RECOVERY = 'r'");
    myPrintF(InputCommand, "\n1st Floor CALL = 'y', 2nd Floor CALL = 'Y'");
    myPrintF(InputCommand, "\n1st Floor CLOSE = 'h', 2nd Floor CLOSE = 'H'");
    myPrintF(InputCommand, "\nQUIT = 'q'");
    myPrintF(InputCommand, "\nCOMMAND>");
}

```



```

function DispE($width_ueM, $width_hidariM, $width_hidariK, $width_nakaH, $width_migiK,
$width_migiM, $width_shitaM, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH, $color_migiK,
$color_migiM, $color_shitaM){
?>
<table width="400px" height="500px" style="border-collapse:collapse"><tr><td>
<table width="200px" height="500px" style="border-collapse:collapse">
<tr><td colspan="5" height="200px" width="200px" bgcolor="#FFFFFF"></td></tr>
<tr><td colspan="5" height="<?=$width_ueM?>px" width="200px" bgcolor="<?=$color_ueM?
>"></td></tr>
<tr>
<td height="100px" width="<?=$width_hidariM?>px" bgcolor="<?=$color_hidariM?>"></td>
<td height="100px" width="<?=$width_hidariK?>px" bgcolor="<?=$color_hidariK?>"></td>
<?php if($width_nakaH != 0){ ?>
<td height="100px" width="<?=$width_nakaH?>px" bgcolor="<?=$color_nakaH?>">
    <?php
        $chOrder = 'N';
        if(isset($_GET['order'])){
            $chOrder = $_GET['order'];
            switch($chOrder){
                case "":
                    WriteFL('o__');
                    break;
                case 'N':
                default:
                    break;
            }
        }
    ?>
    <form method="get" action="./ev_disp.php">
        <input type="submit" style="width:100%;height:90px;padding:0px;" name="order"
value="">
        </input>
    </form>
</td>
<?php } ?>
<td height="100px" width="<?=$width_migiK?>px" bgcolor="<?=$color_migiK?>"></td>
<td height="100px" width="<?=$width_migiM?>px" bgcolor="<?=$color_migiM?>"></td>
</tr>
<tr><td colspan="5" height="<?=$width_shitaM?>px" width="200px" bgcolor="<?=$color_shitaM?
>"></td></tr>
</table>
</td>
<?php
}

```

```

function UserDispE($user_width_ueM, $user_width_hidariM, $user_width_hidariK, $user_width_nakaH,
$user_width_migiK, $user_width_migiM, $user_width_shitaM, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM){
?>
<td>
<table width="200px" height="500px" style="border-collapse:collapse">
<tr><td colspan="5" height="<?=$user_width_ueM?>px" width="200px" bgcolor="<?
=$user_color_ueM?>"></td></tr>
<tr>
<td height="100px" width="<?=$user_width_hidariM?>px" bgcolor="<?=$user_color_hidariM?
>"></td>
<td height="100px" width="<?=$user_width_hidariK?>px" bgcolor="<?=$user_color_hidariK?
>"></td>
<?php if($user_width_nakaH != 0){ ?>
<td height="100px" width="<?=$user_width_nakaH?>px" bgcolor="<?=$user_color_nakaH?>">

```

```

<?php
    $chUserOrder = 'N';
    if(isset($_GET['userorder'])){
        $chUserOrder = $_GET['userorder'];
        switch($chUserOrder){
            case "":
                WriteFL('_0o');
                break;
            case 'N':
            default:
                break;
        }
    }
?>
<form method="get" action="./ev_disp.php">
    <input type="submit" style="width:100%;height:90px;padding:0px;" name="userorder"
value="">
    </input>
</form>
</td>
<?php } ?>
<td height="100px" width="<?=$user_width_migiK?>px" bgcolor="<?=$user_color_migiK?>"></td>
<td height="100px" width="<?=$user_width_migiM?>px" bgcolor="<?=$user_color_migiM?>"></td>
</tr>
<tr><td colspan="5" height="<?=$user_width_shitaM?>px" width="200px" bgcolor="<?
=$user_color_shitaM?>"></td></tr>
</table>
</td></tr></table>
<?php
}

```

```

/*
 * 表示関数
 */
function Disp()
{
    if(($fp = @fopen("Command.txt", "r")) == false){
    }
    else{
        if(($tmpChC = fgets($fp)) == true){
            $chC = str_split($tmpChC);
        }
        fclose($fp);
    }
    if(($fp = @fopen("Motor.txt", "r")) == false){
    }
    else if($chM = fgetc($fp)){
        fclose($fp);
    }
    else{
        fclose($fp);
    }
    if(($fp = @fopen("Limit.txt", "r")) == false){
    }
    else{
        if(($tmpstr = fgets($fp)) == true){
            $str = str_split($tmpstr);
        }
        fclose($fp);
    }
    if(($user_fp = @fopen("UserMotor.txt", "r")) == false){

```

```

}
else if($userChM = fgetc($user_fp)){
    fclose($user_fp);
}
else{
    fclose($user_fp);
}
if(($user_fp = @fopen("UserLimit.txt", "r")) == false){
}
else{
    if(($usertmpstr = fgets($user_fp)) == true){
        $user_str = str_split($usertmpstr);
    }
    fclose($user_fp);
}
if(strcmp($chC[0], 'q') == 0){
    $color_ueM = 'gray';
    $color_hidariM = 'gray';
    $color_hidariK = 'lightgray';
    $color_nakaH = 'gray';
    $color_migiK = 'lightgray';
    $color_migiM = 'gray';
    $color_shitaM = 'gray';
    $user_color_ueM = 'gray';
    $user_color_hidariM = 'gray';
    $user_color_hidariK = 'lightgray';
    $user_color_nakaH = 'gray';
    $user_color_migiK = 'lightgray';
    $user_color_migiM = 'gray';
    $user_color_shitaM = 'gray';
}
else{
    $color_ueM = 'green';
    $color_hidariM = 'green';
    $color_hidariK = 'yellow';
    $color_nakaH = 'lightgray';
    $color_migiK = 'yellow';
    $color_migiM = 'green';
    $color_shitaM = 'green';
    $user_color_ueM = '#F0DDB6';
    $user_color_hidariM = '#F0DDB6';
    $user_color_hidariK = '#330000';
    $user_color_nakaH = '#F0DDB6';
    $user_color_migiK = '#330000';
    $user_color_migiM = '#F0DDB6';
    $user_color_shitaM = '#F0DDB6';
}
echo $chM;
?><!--<br>//--><?php
//var_dump($str);
echo $userChM;
?><!--<br>//--><?php
//var_dump($user_str);
if(($str[3] == 'y') && ($str[7] == 'y'))
{
    DispE(0, 0, 50, 100, 50, 0, 200, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[3] == 'y') && ($str[7] == 'n') && ($str[6] == 'y'))
{
    DispE(0, 6, 50, 88, 50, 6, 200, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}

```

```

}
else if(($str[3] == 'y') && ($str[7] == 'n') && ($str[6] == 'n') && ($str[5] == 'n') && ($str[4] ==
'n'))
{
    DispE(0, 25, 50, 50, 50, 25, 200, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[3] == 'y') && ($str[5] == 'y') && ($str[4] == 'n'))
{
    DispE(0, 44, 50, 12, 50, 44, 200, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[3] == 'y') && ($str[4] == 'y'))
{
    DispE(0, 50, 50, 0, 50, 50, 200, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[3] == 'n') && ($str[2] == 'y') && ($str[7] == 'y'))
{
    DispE(25, 0, 50, 100, 50, 0, 175, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[3] == 'n') && ($str[2] == 'y') && ($str[7] == 'n') && ($str[6] == 'y'))
{
    DispE(25, 6, 50, 88, 50, 6, 175, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[3] == 'n') && ($str[2] == 'y') && ($str[7] == 'n') && ($str[6] == 'n') && ($str[5] ==
'n') && ($str[4] == 'n'))
{
    DispE(25, 25, 50, 50, 50, 25, 175, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[3] == 'n') && ($str[2] == 'y') && ($str[5] == 'y') && ($str[4] == 'n'))
{
    DispE(25, 44, 50, 12, 50, 44, 175, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[3] == 'n') && ($str[2] == 'y') && ($str[4] == 'y'))
{
    DispE(25, 50, 50, 0, 50, 50, 175, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[3] == 'n') && ($str[2] == 'n') && ($str[1] == 'n') && ($str[0] == 'n') && ($str[7] ==
'y'))
{
    DispE(100, 0, 50, 100, 50, 0, 100, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[3] == 'n') && ($str[2] == 'n') && ($str[1] == 'n') && ($str[0] == 'n') && ($str[7] ==
'n') && ($str[6] == 'y'))
{
    DispE(100, 6, 50, 88, 50, 6, 100, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[3] == 'n') && ($str[2] == 'n') && ($str[1] == 'n') && ($str[0] == 'n') && ($str[7] ==
'n') && ($str[6] == 'n') && ($str[5] == 'n') && ($str[4] == 'n'))
{
    DispE(100, 25, 50, 50, 50, 25, 100, $color_ueM, $color_hidariM, $color_hidariK,
$color_nakaH, $color_migiK, $color_migiM, $color_shitaM);
}
}

```

```

else if(($str[3] == 'n') && ($str[2] == 'n') && ($str[1] == 'n') && ($str[0] == 'n') && ($str[5] ==
'y') && ($str[4] == 'n'))
{
    DispE(100, 44, 50, 12, 50, 44, 100, $color_ueM, $color_hidariM, $color_hidariK,
$color_nakaH, $color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[3] == 'n') && ($str[2] == 'n') && ($str[1] == 'n') && ($str[0] == 'n') && ($str[4] ==
'y'))
{
    DispE(100, 50, 50, 0, 50, 50, 100, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[1] == 'y') && ($str[0] == 'n') && ($str[7] == 'y'))
{
    DispE(175, 0, 50, 100, 50, 0, 25, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[1] == 'y') && ($str[0] == 'n') && ($str[7] == 'n') && ($str[6] == 'y'))
{
    DispE(175, 6, 50, 88, 50, 6, 25, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[1] == 'y') && ($str[0] == 'n') && ($str[7] == 'n') && ($str[6] == 'n') && ($str[5] ==
'n') && ($str[4] == 'n'))
{
    DispE(175, 25, 50, 50, 50, 25, 25, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[1] == 'y') && ($str[0] == 'n') && ($str[5] == 'y') && ($str[4] == 'n'))
{
    DispE(175, 44, 50, 12, 50, 44, 25, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[1] == 'y') && ($str[0] == 'n') && ($str[4] == 'y'))
{
    DispE(175, 50, 50, 0, 50, 50, 25, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[0] == 'y') && ($str[7] == 'y'))
{
    DispE(200, 0, 50, 100, 50, 0, 0, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[0] == 'y') && ($str[7] == 'n') && ($str[6] == 'y'))
{
    DispE(200, 6, 50, 88, 50, 6, 0, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[0] == 'y') && ($str[7] == 'n') && ($str[6] == 'n') && ($str[5] == 'n') && ($str[4] ==
'n'))
{
    DispE(200, 25, 50, 50, 50, 25, 0, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[0] == 'y') && ($str[5] == 'y') && ($str[4] == 'n'))
{
    DispE(200, 44, 50, 12, 50, 44, 0, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
else if(($str[0] == 'y') && ($str[4] == 'y'))
{

```

```

DispE(200, 50, 50, 0, 50, 50, 0, $color_ueM, $color_hidariM, $color_hidariK, $color_nakaH,
$color_migiK, $color_migiM, $color_shitaM);
}
if(($user_str[3] == 'y') && ($user_str[7] == 'y'))
{
    UserDispE(0, 0, 50, 100, 50, 0, 400, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[3] == 'y') && ($user_str[7] == 'n') && ($user_str[6] == 'y'))
{
    UserDispE(0, 6, 50, 88, 50, 6, 400, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[3] == 'y') && ($user_str[7] == 'n') && ($user_str[6] == 'n') && ($user_str[5]
== 'n') && ($user_str[4] == 'n'))
{
    UserDispE(0, 25, 50, 50, 50, 25, 400, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[3] == 'y') && ($user_str[5] == 'y') && ($user_str[4] == 'n'))
{
    UserDispE(0, 44, 50, 12, 50, 44, 400, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[3] == 'y') && ($user_str[4] == 'y'))
{
    UserDispE(0, 50, 50, 0, 50, 50, 400, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[3] == 'n') && ($user_str[2] == 'y') && ($user_str[7] == 'y'))
{
    UserDispE(25, 0, 50, 100, 50, 0, 375, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[3] == 'n') && ($user_str[2] == 'y') && ($user_str[7] == 'n') && ($user_str[6]
== 'y'))
{
    UserDispE(25, 6, 50, 88, 50, 6, 375, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[3] == 'n') && ($user_str[2] == 'y') && ($user_str[7] == 'n') && ($user_str[6]
== 'n') && ($user_str[5] == 'n') && ($user_str[4] == 'n'))
{
    UserDispE(25, 25, 50, 50, 50, 25, 375, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[3] == 'n') && ($user_str[2] == 'y') && ($user_str[5] == 'y') && ($user_str[4]
== 'n'))
{
    UserDispE(25, 44, 50, 12, 50, 44, 375, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[3] == 'n') && ($user_str[2] == 'y') && ($user_str[4] == 'y'))
{
    UserDispE(25, 50, 50, 0, 50, 50, 375, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[3] == 'n') && ($user_str[2] == 'n') && ($user_str[11] == 'u') && ($user_str[10]
== 'u') && ($user_str[7] == 'y'))
{
    UserDispE(100, 0, 50, 100, 50, 0, 300, $user_color_ueM, $user_color_hidariM,

```

```

$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[3] == 'n') && ($user_str[2] == 'n') && ($user_str[11] == 'u') && ($user_str[10]
== 'u') && ($user_str[7] == 'n') && ($user_str[6] == 'y'))
    {
        UserDispE(100, 6, 50, 88, 50, 6, 300, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[3] == 'n') && ($user_str[2] == 'n') && ($user_str[11] == 'u') && ($user_str[10]
== 'u') && ($user_str[7] == 'n') && ($user_str[6] == 'n') && ($user_str[5] == 'n') && ($user_str[4]
== 'n'))
    {
        UserDispE(100, 25, 50, 50, 50, 25, 300, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[3] == 'n') && ($user_str[2] == 'n') && ($user_str[11] == 'u') && ($user_str[10]
== 'u') && ($user_str[5] == 'y') && ($user_str[4] == 'n'))
    {
        UserDispE(100, 44, 50, 12, 50, 44, 300, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[3] == 'n') && ($user_str[2] == 'n') && ($user_str[11] == 'u') && ($user_str[10]
== 'u') && ($user_str[4] == 'y'))
    {
        UserDispE(100, 50, 50, 0, 50, 50, 300, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[11] == 'd') && ($user_str[10] == 'u') && ($user_str[7] == 'y'))
    {
        UserDispE(175, 0, 50, 100, 50, 0, 225, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[11] == 'd') && ($user_str[10] == 'u') && ($user_str[7] == 'n') && ($user_str[6]
== 'y'))
    {
        UserDispE(175, 6, 50, 88, 50, 6, 225, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[11] == 'd') && ($user_str[10] == 'u') && ($user_str[7] == 'n') && ($user_str[6]
== 'n') && ($user_str[5] == 'n') && ($user_str[4] == 'n'))
    {
        UserDispE(175, 25, 50, 50, 50, 25, 225, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[11] == 'd') && ($user_str[10] == 'u') && ($user_str[5] == 'y') && ($user_str[4]
== 'n'))
    {
        UserDispE(175, 44, 50, 12, 50, 44, 225, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[11] == 'd') && ($user_str[10] == 'u') && ($user_str[4] == 'y'))
    {
        UserDispE(175, 50, 50, 0, 50, 50, 225, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[10] == 'd') && ($user_str[9] == 'u') && ($user_str[7] == 'y'))
    {
        UserDispE(200, 0, 50, 100, 50, 0, 200, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[10] == 'd') && ($user_str[9] == 'u') && ($user_str[7] == 'n') && ($user_str[6]

```

```

== 'y'))
{
    UserDispE(200, 6, 50, 88, 50, 6, 200, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[10] == 'd') && ($user_str[9] == 'u') && ($user_str[7] == 'n') && ($user_str[6]
== 'n') && ($user_str[5] == 'n') && ($user_str[4] == 'n'))
{
    UserDispE(200, 25, 50, 50, 50, 25, 200, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[10] == 'd') && ($user_str[9] == 'u') && ($user_str[5] == 'y') && ($user_str[4]
== 'n'))
{
    UserDispE(200, 44, 50, 12, 50, 44, 200, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[10] == 'd') && ($user_str[9] == 'u') && ($user_str[4] == 'y'))
{
    UserDispE(200, 50, 50, 0, 50, 50, 200, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[9] == 'd') && ($user_str[8] == 'u') && ($user_str[7] == 'y'))
{
    UserDispE(225, 0, 50, 100, 50, 0, 175, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[9] == 'd') && ($user_str[8] == 'u') && ($user_str[7] == 'n') && ($user_str[6]
== 'y'))
{
    UserDispE(225, 6, 50, 88, 50, 6, 175, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[9] == 'd') && ($user_str[8] == 'u') && ($user_str[7] == 'n') && ($user_str[6]
== 'n') && ($user_str[5] == 'n') && ($user_str[4] == 'n'))
{
    UserDispE(225, 25, 50, 50, 50, 25, 175, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[9] == 'd') && ($user_str[8] == 'u') && ($user_str[5] == 'y') && ($user_str[4]
== 'n'))
{
    UserDispE(225, 44, 50, 12, 50, 44, 175, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[9] == 'd') && ($user_str[8] == 'u') && ($user_str[4] == 'y'))
{
    UserDispE(225, 50, 50, 0, 50, 50, 175, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[1] == 'n') && ($user_str[0] == 'n') && ($user_str[9] == 'd') && ($user_str[8]
== 'd') && ($user_str[7] == 'y'))
{
    UserDispE(300, 0, 50, 100, 50, 0, 100, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[1] == 'n') && ($user_str[0] == 'n') && ($user_str[9] == 'd') && ($user_str[8]
== 'd') && ($user_str[7] == 'n') && ($user_str[6] == 'y'))
{
    UserDispE(300, 6, 50, 88, 50, 6, 100, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}

```



```

    }
    else if(($user_str[1] == 'n') && ($user_str[0] == 'n') && ($user_str[9] == 'd') && ($user_str[8]
== 'd') && ($user_str[7] == 'n') && ($user_str[6] == 'n') && ($user_str[5] == 'n') && ($user_str[4]
== 'n'))
    {
        UserDispE(300, 25, 50, 50, 50, 25, 100, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[1] == 'n') && ($user_str[0] == 'n') && ($user_str[9] == 'd') && ($user_str[8]
== 'd') && ($user_str[5] == 'y') && ($user_str[4] == 'n'))
    {
        UserDispE(300, 44, 50, 12, 50, 44, 100, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[1] == 'n') && ($user_str[0] == 'n') && ($user_str[9] == 'd') && ($user_str[8]
== 'd') && ($user_str[4] == 'y'))
    {
        UserDispE(300, 50, 50, 0, 50, 50, 100, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[1] == 'y') && ($user_str[0] == 'n') && ($user_str[7] == 'y'))
    {
        UserDispE(375, 0, 50, 100, 50, 0, 25, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[1] == 'y') && ($user_str[0] == 'n') && ($user_str[7] == 'n') && ($user_str[6]
== 'y'))
    {
        UserDispE(375, 6, 50, 88, 50, 6, 25, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[1] == 'y') && ($user_str[0] == 'n') && ($user_str[7] == 'n') && ($user_str[6]
== 'n') && ($user_str[5] == 'n') && ($user_str[4] == 'n'))
    {
        UserDispE(375, 25, 50, 50, 50, 25, 25, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[1] == 'y') && ($user_str[0] == 'n') && ($user_str[5] == 'y') && ($user_str[4]
== 'n'))
    {
        UserDispE(375, 44, 50, 12, 50, 44, 25, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[1] == 'y') && ($user_str[0] == 'n') && ($user_str[4] == 'y'))
    {
        UserDispE(375, 50, 50, 0, 50, 50, 25, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[0] == 'y') && ($user_str[7] == 'y'))
    {
        UserDispE(400, 0, 50, 100, 50, 0, 0, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[0] == 'y') && ($user_str[7] == 'n') && ($user_str[6] == 'y'))
    {
        UserDispE(400, 6, 50, 88, 50, 6, 0, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[0] == 'y') && ($user_str[7] == 'n') && ($user_str[6] == 'n') && ($user_str[5]
== 'n') && ($user_str[4] == 'n'))
    {

```

```

        UserDispE(400, 25, 50, 50, 50, 25, 0, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[0] == 'y') && ($user_str[5] == 'y') && ($user_str[4] == 'n'))
    {
        UserDispE(400, 44, 50, 12, 50, 44, 0, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[0] == 'y') && ($user_str[4] == 'y'))
    {
        UserDispE(400, 50, 50, 0, 50, 50, 0, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    /* 入力指示 */
    /* DispInput(); */
    return;
}

```

```

?>
</head>
<body>
<?php
Disp();
?>
<form method="get" action="./ev_disp.php" name="myform">
<script type="text/javascript">
<!--
setInterval("document.myform.submit()",2000);
//-->
</script>
</form>
</body>
</html>

```

```

<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">

<?php
/* 表示を表す列挙体宣言 */
const Panel = 1;
const ClsPnl = 2;
const InputCommand = 3;
const Monitor = 4;

function myPrintF($mode, $str)
{
    switch($mode)
    {
        case ClsPnl:
            break;
        case Panel:
            ?><br><?php
            echo $str;
            break;
        case InputCommand:
            ?><br><?php
            echo $str;
            break;
        case Monitor:
            ?><br><?php
            echo $str;
            break;
        default:
            break;
    }
    return;
}

```

```

/* EV_Display.c */

```

```

function WriteFL($chO)
{
    if(($fp = @fopen("Order.txt", "w")) == false){
    }
    else if(fwrite($fp, $chO)){
        fclose($fp);
    }
    else{
        fclose($fp);
    }
}

function WriteTwin1FL($ch1)
{
    if(($fp = @fopen("Twin1Command.txt", "w")) == false){
    }
    else if(fwrite($fp, $ch1)){
        fclose($fp);
    }
    else{
        fclose($fp);
    }
}

```

```

function WriteTwin2FL($ch2)
{
    if(($fp = @fopen("Twin2Command.txt", "w")) == false){
    }
    else if(fwrite($fp, $ch2)){
        fclose($fp);
    }
    else{
        fclose($fp);
    }
}

function DispInput()
{
    /* 入力指示 */
    myPrintf(InputCommand, "\nUP = 'u', DOWN = 'd', OPEN = 'o', CLOSE = 'c'");
    myPrintf(InputCommand, "\nEMERGENCY = 's', RECOVERY = 'r'");
    myPrintf(InputCommand, "\n1st Floor CALL = 'y', 2nd Floor CALL = 'Y'");
    myPrintf(InputCommand, "\n1st Floor CLOSE = 'h', 2nd Floor CLOSE = 'H'");
    myPrintf(InputCommand, "\nQUIT = 'q'");
    myPrintf(InputCommand, "\nCOMMAND>");
}

function UserDispE($TwinInt, $user_width_ueM, $user_width_hidariM, $user_width_hidariK,
$user_width_nakaH, $user_width_migiK, $user_width_migiM, $user_width_shitaM, $user_color_ueM,
$user_color_hidariM, $user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM,
$user_color_shitaM){
if($TwinInt == 1){
?>
<table width="200px" height="500px" style="border-collapse:collapse">
<tr><td colspan="5" height="<?=$user_width_ueM?>px" width="200px" bgcolor="<?
=$user_color_ueM?>"></td></tr>
<tr>
<td height="100px" width="<?=$user_width_hidariM?>px" bgcolor="<?=$user_color_hidariM?
>"></td>
<td height="100px" width="<?=$user_width_hidariK?>px" bgcolor="<?=$user_color_hidariK?
>"></td>
<?php if($user_width_nakaH != 0){ ?>
<td height="100px" width="<?=$user_width_nakaH?>px" bgcolor="<?=$user_color_nakaH?>">
<?php
    $chUser1Order = 'N';
    if(isset($_GET['user1order'])){
        $chUser1Order = $_GET['user1order'];
        switch($chUser1Order){
            case "":
                WriteTwin1FL('_0o');
                break;
            case 'N':
            default:
                break;
        }
    }
?>
<form method="get" action="/ev_twin_disp.php">
    <input type="submit" style="width:100%;height:90px;padding:0px;" name="user1order"
value="">
    </input>
</form>
</td>
<?php } ?>
<td height="100px" width="<?=$user_width_migiK?>px" bgcolor="<?=$user_color_migiK?>"></td>

```

```

<td height="100px" width="<?=$user_width_migiM?>px" bgcolor="<?=$user_color_migiM?>"></td>
</tr>
<tr><td colspan="5" height="<?=$user_width_shitaM?>px" width="200px" bgcolor="<?
=$user_color_shitaM?>"></td></tr>
</table>
<?php
}else if($TwinInt == 2){
?>
<table width="200px" height="500px" style="border-collapse:collapse">
<tr><td colspan="5" height="<?=$user_width_ueM?>px" width="200px" bgcolor="<?
=$user_color_ueM?>"></td></tr>
<tr>
<td height="100px" width="<?=$user_width_hidariM?>px" bgcolor="<?=$user_color_hidariM?
>"></td>
<td height="100px" width="<?=$user_width_hidariK?>px" bgcolor="<?=$user_color_hidariK?
>"></td>
<?php if($user_width_nakaH != 0){ ?>
<td height="100px" width="<?=$user_width_nakaH?>px" bgcolor="<?=$user_color_nakaH?>">
<?php
    $chUser2Order = 'N';
    if(isset($_GET['user2order'])){
        $chUser2Order = $_GET['user2order'];
        switch($chUser2Order){
            case "":
                WriteTwin2FL('_0o');
                break;
            case 'N':
            default:
                break;
        }
    }
?>
<form method="get" action="./ev_twin_disp.php">
    <input type="submit" style="width:100%;height:90px;padding:0px;" name="user2order"
value="">
    </input>
</form>
</td>
<?php } ?>
<td height="100px" width="<?=$user_width_migiK?>px" bgcolor="<?=$user_color_migiK?>"></td>
<td height="100px" width="<?=$user_width_migiM?>px" bgcolor="<?=$user_color_migiM?>"></td>
</tr>
<tr><td colspan="5" height="<?=$user_width_shitaM?>px" width="200px" bgcolor="<?
=$user_color_shitaM?>"></td></tr>
</table>
<?php
}
}
/*
 * 表示関数
 */
function Disp($chC, $chTC, $TwinInt, $userChM, $user_str)
{
    if(strcmp($chC[0], 'q') == 0){
        $user_color_ueM = 'gray';
        $user_color_hidariM = 'gray';
        $user_color_hidariK = 'lightgray';
        $user_color_nakaH = 'gray';
        $user_color_migiK = 'lightgray';
        $user_color_migiM = 'gray';

```

```

        $user_color_shitaM = 'gray';
    }else{
        $user_color_ueM = '#F0DDB6';
        $user_color_hidariM = '#F0DDB6';
        $user_color_hidariK = '#330000';
        $user_color_nakaH = '#F0DDB6';
        $user_color_migiK = '#330000';
        $user_color_migiM = '#F0DDB6';
        $user_color_shitaM = '#F0DDB6';
    }
    echo $chTC[0] . $chTC[1];
    ?><br><?php
    echo $userChM;
    ?><br><?php
    //var_dump($user_str);
    if(($user_str[3] == 'y') && ($user_str[7] == 'y'))
    {
        UserDispE($TwinInt, 0, 0, 50, 100, 50, 0, 400, $user_color_ueM, $user_color_hidariM,
        $user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[3] == 'y') && ($user_str[7] == 'n') && ($user_str[6] == 'y'))
    {
        UserDispE($TwinInt, 0, 6, 50, 88, 50, 6, 400, $user_color_ueM, $user_color_hidariM,
        $user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[3] == 'y') && ($user_str[7] == 'n') && ($user_str[6] == 'n') && ($user_str[5]
    == 'n') && ($user_str[4] == 'n'))
    {
        UserDispE($TwinInt, 0, 25, 50, 50, 50, 25, 400, $user_color_ueM, $user_color_hidariM,
        $user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[3] == 'y') && ($user_str[5] == 'y') && ($user_str[4] == 'n'))
    {
        UserDispE($TwinInt, 0, 44, 50, 12, 50, 44, 400, $user_color_ueM, $user_color_hidariM,
        $user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[3] == 'y') && ($user_str[4] == 'y'))
    {
        UserDispE($TwinInt, 0, 50, 50, 0, 50, 50, 400, $user_color_ueM, $user_color_hidariM,
        $user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[3] == 'n') && ($user_str[2] == 'y') && ($user_str[7] == 'y'))
    {
        UserDispE($TwinInt, 25, 0, 50, 100, 50, 0, 375, $user_color_ueM, $user_color_hidariM,
        $user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[3] == 'n') && ($user_str[2] == 'y') && ($user_str[7] == 'n') && ($user_str[6]
    == 'y'))
    {
        UserDispE($TwinInt, 25, 6, 50, 88, 50, 6, 375, $user_color_ueM, $user_color_hidariM,
        $user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[3] == 'n') && ($user_str[2] == 'y') && ($user_str[7] == 'n') && ($user_str[6]
    == 'n') && ($user_str[5] == 'n') && ($user_str[4] == 'n'))
    {
        UserDispE($TwinInt, 25, 25, 50, 50, 50, 25, 375, $user_color_ueM, $user_color_hidariM,
        $user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[3] == 'n') && ($user_str[2] == 'y') && ($user_str[5] == 'y') && ($user_str[4]
    == 'n'))
    {

```

```

    UserDispE($TwinInt, 25, 44, 50, 12, 50, 44, 375, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[3] == 'n') && ($user_str[2] == 'y') && ($user_str[4] == 'y'))
{
    UserDispE($TwinInt, 25, 50, 50, 0, 50, 50, 375, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[3] == 'n') && ($user_str[2] == 'n') && ($user_str[11] == 'u') && ($user_str[10]
== 'u') && ($user_str[7] == 'y'))
{
    UserDispE($TwinInt, 100, 0, 50, 100, 50, 0, 300, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[3] == 'n') && ($user_str[2] == 'n') && ($user_str[11] == 'u') && ($user_str[10]
== 'u') && ($user_str[7] == 'n') && ($user_str[6] == 'y'))
{
    UserDispE($TwinInt, 100, 6, 50, 88, 50, 6, 300, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[3] == 'n') && ($user_str[2] == 'n') && ($user_str[11] == 'u') && ($user_str[10]
== 'u') && ($user_str[7] == 'n') && ($user_str[6] == 'n') && ($user_str[5] == 'n') && ($user_str[4]
== 'n'))
{
    UserDispE($TwinInt, 100, 25, 50, 50, 50, 25, 300, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[3] == 'n') && ($user_str[2] == 'n') && ($user_str[11] == 'u') && ($user_str[10]
== 'u') && ($user_str[5] == 'y') && ($user_str[4] == 'n'))
{
    UserDispE($TwinInt, 100, 44, 50, 12, 50, 44, 300, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[3] == 'n') && ($user_str[2] == 'n') && ($user_str[11] == 'u') && ($user_str[10]
== 'u') && ($user_str[4] == 'y'))
{
    UserDispE($TwinInt, 100, 50, 50, 0, 50, 50, 300, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[11] == 'd') && ($user_str[10] == 'u') && ($user_str[7] == 'y'))
{
    UserDispE($TwinInt, 175, 0, 50, 100, 50, 0, 225, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[11] == 'd') && ($user_str[10] == 'u') && ($user_str[7] == 'n') && ($user_str[6]
== 'y'))
{
    UserDispE($TwinInt, 175, 6, 50, 88, 50, 6, 225, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[11] == 'd') && ($user_str[10] == 'u') && ($user_str[7] == 'n') && ($user_str[6]
== 'n') && ($user_str[5] == 'n') && ($user_str[4] == 'n'))
{
    UserDispE($TwinInt, 175, 25, 50, 50, 50, 25, 225, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[11] == 'd') && ($user_str[10] == 'u') && ($user_str[5] == 'y') && ($user_str[4]
== 'n'))
{
    UserDispE($TwinInt, 175, 44, 50, 12, 50, 44, 225, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}

```

```

}
else if(($user_str[11] == 'd') && ($user_str[10] == 'u') && ($user_str[4] == 'y'))
{
    UserDispE($TwinInt, 175, 50, 50, 0, 50, 50, 225, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[10] == 'd') && ($user_str[9] == 'u') && ($user_str[7] == 'y'))
{
    UserDispE($TwinInt, 200, 0, 50, 100, 50, 0, 200, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[10] == 'd') && ($user_str[9] == 'u') && ($user_str[7] == 'n') && ($user_str[6]
== 'y'))
{
    UserDispE($TwinInt, 200, 6, 50, 88, 50, 6, 200, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[10] == 'd') && ($user_str[9] == 'u') && ($user_str[7] == 'n') && ($user_str[6]
== 'n') && ($user_str[5] == 'n') && ($user_str[4] == 'n'))
{
    UserDispE($TwinInt, 200, 25, 50, 50, 50, 25, 200, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[10] == 'd') && ($user_str[9] == 'u') && ($user_str[5] == 'y') && ($user_str[4]
== 'n'))
{
    UserDispE($TwinInt, 200, 44, 50, 12, 50, 44, 200, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[10] == 'd') && ($user_str[9] == 'u') && ($user_str[4] == 'y'))
{
    UserDispE($TwinInt, 200, 50, 50, 0, 50, 50, 200, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[9] == 'd') && ($user_str[8] == 'u') && ($user_str[7] == 'y'))
{
    UserDispE($TwinInt, 225, 0, 50, 100, 50, 0, 175, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[9] == 'd') && ($user_str[8] == 'u') && ($user_str[7] == 'n') && ($user_str[6]
== 'y'))
{
    UserDispE($TwinInt, 225, 6, 50, 88, 50, 6, 175, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[9] == 'd') && ($user_str[8] == 'u') && ($user_str[7] == 'n') && ($user_str[6]
== 'n') && ($user_str[5] == 'n') && ($user_str[4] == 'n'))
{
    UserDispE($TwinInt, 225, 25, 50, 50, 50, 25, 175, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[9] == 'd') && ($user_str[8] == 'u') && ($user_str[5] == 'y') && ($user_str[4]
== 'n'))
{
    UserDispE($TwinInt, 225, 44, 50, 12, 50, 44, 175, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}
else if(($user_str[9] == 'd') && ($user_str[8] == 'u') && ($user_str[4] == 'y'))
{
    UserDispE($TwinInt, 225, 50, 50, 0, 50, 50, 175, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
}

```



```

    }
    else if(($user_str[1] == 'n') && ($user_str[0] == 'n') && ($user_str[9] == 'd') && ($user_str[8]
== 'd') && ($user_str[7] == 'y'))
    {
        UserDispE($TwinInt, 300, 0, 50, 100, 50, 0, 100, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[1] == 'n') && ($user_str[0] == 'n') && ($user_str[9] == 'd') && ($user_str[8]
== 'd') && ($user_str[7] == 'n') && ($user_str[6] == 'y'))
    {
        UserDispE($TwinInt, 300, 6, 50, 88, 50, 6, 100, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[1] == 'n') && ($user_str[0] == 'n') && ($user_str[9] == 'd') && ($user_str[8]
== 'd') && ($user_str[7] == 'n') && ($user_str[6] == 'n') && ($user_str[5] == 'n') && ($user_str[4]
== 'n'))
    {
        UserDispE($TwinInt, 300, 25, 50, 50, 50, 25, 100, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[1] == 'n') && ($user_str[0] == 'n') && ($user_str[9] == 'd') && ($user_str[8]
== 'd') && ($user_str[5] == 'y') && ($user_str[4] == 'n'))
    {
        UserDispE($TwinInt, 300, 44, 50, 12, 50, 44, 100, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[1] == 'n') && ($user_str[0] == 'n') && ($user_str[9] == 'd') && ($user_str[8]
== 'd') && ($user_str[4] == 'y'))
    {
        UserDispE($TwinInt, 300, 50, 50, 0, 50, 50, 100, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[1] == 'y') && ($user_str[0] == 'n') && ($user_str[7] == 'y'))
    {
        UserDispE($TwinInt, 375, 0, 50, 100, 50, 0, 25, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[1] == 'y') && ($user_str[0] == 'n') && ($user_str[7] == 'n') && ($user_str[6]
== 'y'))
    {
        UserDispE($TwinInt, 375, 6, 50, 88, 50, 6, 25, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[1] == 'y') && ($user_str[0] == 'n') && ($user_str[7] == 'n') && ($user_str[6]
== 'n') && ($user_str[5] == 'n') && ($user_str[4] == 'n'))
    {
        UserDispE($TwinInt, 375, 25, 50, 50, 50, 25, 25, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[1] == 'y') && ($user_str[0] == 'n') && ($user_str[5] == 'y') && ($user_str[4]
== 'n'))
    {
        UserDispE($TwinInt, 375, 44, 50, 12, 50, 44, 25, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[1] == 'y') && ($user_str[0] == 'n') && ($user_str[4] == 'y'))
    {
        UserDispE($TwinInt, 375, 50, 50, 0, 50, 50, 25, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[0] == 'y') && ($user_str[7] == 'y'))

```

```

    {
        UserDispE($TwinInt, 400, 0, 50, 100, 50, 0, 0, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[0] == 'y') && ($user_str[7] == 'n') && ($user_str[6] == 'y'))
    {
        UserDispE($TwinInt, 400, 6, 50, 88, 50, 6, 0, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[0] == 'y') && ($user_str[7] == 'n') && ($user_str[6] == 'n') && ($user_str[5]
== 'n') && ($user_str[4] == 'n'))
    {
        UserDispE($TwinInt, 400, 25, 50, 50, 50, 25, 0, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[0] == 'y') && ($user_str[5] == 'y') && ($user_str[4] == 'n'))
    {
        UserDispE($TwinInt, 400, 44, 50, 12, 50, 44, 0, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    else if(($user_str[0] == 'y') && ($user_str[4] == 'y'))
    {
        UserDispE($TwinInt, 400, 50, 50, 0, 50, 50, 0, $user_color_ueM, $user_color_hidariM,
$user_color_hidariK, $user_color_nakaH, $user_color_migiK, $user_color_migiM, $user_color_shitaM);
    }
    /* 入力指示 */
    /* DispInput(); */
    return;
}

```

```

function Twin1_Read()
{
    if(($fp = @fopen("Command.txt", "r")) == false){
    }
    else{
        if(($tmpChC = fgets($fp)) == true){
            $chC = str_split($tmpChC);
        }
        fclose($fp);
    }
    if(($fp = @fopen("Twin1Command.txt", "r")) == false){
    }
    else{
        if(($tmpChTC = fgets($fp)) == true){
            $chTC = str_split($tmpChTC);
        }
        fclose($fp);
    }
    if(($user_fp = @fopen("Twin1Motor.txt", "r")) == false){
    }
    else if($userChM = fgetc($user_fp)){
        fclose($user_fp);
    }
    else{
        fclose($user_fp);
    }
    if(($user_fp = @fopen("Twin1Limit.txt", "r")) == false){
    }
    else{
        if(($usertmpstr = fgets($user_fp)) == true){
            $user_str = str_split($usertmpstr);

```

```

    }
    fclose($user_fp);
}
Disp($chC, $chTC, 1, $userChM, $user_str);
}

```

```

function Twin2_Read()
{
    if(($fp = @fopen("Command.txt", "r")) == false){
    }
    else{
        if(($tmpChC = fgets($fp)) == true){
            $chC = str_split($tmpChC);
        }
        fclose($fp);
    }
    if(($fp = @fopen("Twin2Command.txt", "r")) == false){
    }
    else{
        if(($tmpChTC = fgets($fp)) == true){
            $chTC = str_split($tmpChTC);
        }
        fclose($fp);
    }
    if(($user_fp = @fopen("Twin2Motor.txt", "r")) == false){
    }
    else if($userChM = fgetc($user_fp)){
        fclose($user_fp);
    }
    else{
        fclose($user_fp);
    }
    if(($user_fp = @fopen("Twin2Limit.txt", "r")) == false){
    }
    else{
        if(($usertmpstr = fgets($user_fp)) == true){
            $user_str = str_split($usertmpstr);
        }
        fclose($user_fp);
    }
    Disp($chC, $chTC, 2, $userChM, $user_str);
}

```

```

?>
</head>
<body>
<table width="400px" height="500px" style="border-collapse:collapse"><tr><td><?php
Twin1_Read();?></td><td><?php Twin2_Read();?></td></tr></table>

<form method="get" action="./ev_twin_disp.php" name="myform">
<script type="text/javascript">
<!--
setInterval("document.myform.submit()",2000);
//-->
</script>
</form>
</body>
</html>

```

```

<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">

<?php
function WriteFL($chO)
{
    if(($fp = @fopen("Order.txt", "w")) == false){
    }
    else if(fwrite($fp, $chO)){
        fclose($fp);
    }
    else{
        fclose($fp);
    }
}
function WriteUserFL($chO)
{
    if(($fp = @fopen("UserCommand.txt", "w")) == false){
    }
    else if(fwrite($fp, $chO)){
        fclose($fp);
    }
    else{
        fclose($fp);
    }
}
function WriteTwin1FL($ch1)
{
    if(($fp = @fopen("Twin1Command.txt", "w")) == false){
    }
    else if(fwrite($fp, $ch1)){
        fclose($fp);
    }
    else{
        fclose($fp);
    }
}
function WriteTwin2FL($ch2)
{
    if(($fp = @fopen("Twin2Command.txt", "w")) == false){
    }
    else if(fwrite($fp, $ch2)){
        fclose($fp);
    }
    else{
        fclose($fp);
    }
}
?>

```

```

<style>
input[type=radio] {
display: none; /* ラジオボタンを非表示にする */
}
.label {
/* display: block; ブロックレベル要素化する */
display: flex;
float: left; /* 要素の左寄せ・回り込を指定する */

```

```

margin: 5px; /* ボックス外側の余白を指定する */
width: 60px; /* ボックスの横幅を指定する */
height: 60px; /* ボックスの高さを指定する */
padding-left: 5px; /* ボックス内左側の余白を指定する */
padding-right: 5px; /* ボックス内御右側の余白を指定する */
color: #b20000; /* フォントの色を指定 */
text-align: center; /* テキストのセンタリングを指定する */
align-items: center; /* テキストの縦センタリングを指定する */
justify-content: center; /* テキストの横センタリングを指定する */
line-height: 25px; /* 行の高さを指定する */
cursor: pointer; /* マウスカーソルの形(リンクカーソル)を指定する */
border: 2px solid #006DD9; /* ボックスの境界線を実線で指定する */
border-radius: 5px; /* 角丸を指定する */
}
</style>

```

```
</head>
```

```
<body>
```

```
<table height="570px" width="730px"><tr>
```

```
<td width="150px">
```

```
<?php
```

```
$chCommand = 'N';
```

```
$chOrder = 'N';
```

```
if(isset($_GET['command'])){
```

```
    $chCommand = $_GET['command'];
```

```
}
```

```
if(isset($_GET['order'])){
```

```
    $chOrder = $_GET['order'];
```

```
    switch($chOrder){
```

```
        case 's':
```

```
        case 'r':
```

```
        case 'u':
```

```
        case 'd':
```

```
        case 'o':
```

```
        case 'c':
```

```
        case 'Y':
```

```
        case 'y':
```

```
        case 'H':
```

```
        case 'h':
```

```
        case 'q':
```

```
            WriteFL($chOrder.'__');
```

```
            break;
```

```
        case 'N':
```

```
        default:
```

```
            break;
```

```
    }
```

```
}
```

```
if(($fp = @fopen("Command.txt", "r")) == false){
```

```
}
```

```
else if($chC = fgetc($fp)){
```

```
    fclose($fp);
```

```
}
```

```
else{
```

```
    fclose($fp);
```

```
}
```

```
$userChCommand = '_NN';
```

```
$Twin1ChCommand = '_NN';
```

```
$Twin2ChCommand = '_NN';
```

```
$userChOrder = '_NN';
```

```
$TwinChangeCh = 1;
```

```

if(isset($_GET['usercommand'])){
    $userChCommand = $_GET['usercommand'];
}
if(isset($_GET['TwinChange'])){
    $TwinChangeCh = $_GET['TwinChange'];
}
if(isset($_GET['userorder'])){
    $userChOrder = $_GET['userorder'];
    $UCO = $userChOrder[0] . $userChOrder[1] . $userChOrder[2] . '\0';
    switch($userChOrder){
        case ' e _':
            WriteFL($UCO);
            WriteUserFL($UCO);
            break;
        case ' e f':
            WriteFL($UCO);
            WriteTwin1FL($UCO);
            WriteTwin2FL($UCO);
            break;
        case ' _03_':
        case ' _02_':
        case ' _01_':
        case ' _0o_':
        case ' _0c_':
        case ' _0L_':
            WriteFL($UCO);
            WriteUserFL($UCO);
            break;
        case ' _03l':
        case ' _02l':
        case ' _01l':
        case ' _0ol':
        case ' _0cl':
        case ' _0Ll':
            WriteFL($UCO);
            WriteTwin1FL($UCO);
            break;
        case ' _03r':
        case ' _02r':
        case ' _01r':
        case ' _0or':
        case ' _0cr':
        case ' _0Lr':
            WriteFL($UCO);
            WriteTwin2FL($UCO);
            break;
        case ' _3D_':
        case ' _2U_':
        case ' _2D_':
        case ' _1U_':
            WriteFL($UCO);
            WriteUserFL($UCO);
            break;
        case ' _3Df':
        case ' _2Uf':
        case ' _2Df':
        case ' _1Uf':
            WriteFL($UCO);
            if($TwinChangeCh == 1){
                WriteTwin1FL($UCO);
                $TwinChangeCh = 2;
            }
    }
}

```

```

        }else if($TwinChangeCh == 2){
            WriteTwin2FL($UCO);
            $TwinChangeCh = 1;
        }
        break;
    case '_NN_':
    case '_NNf':
    default:
        break;
}
}
}
if(($user_fp = @fopen("UserCommand.txt", "r")) == false){
}
else if($userChC = fgets($user_fp, 4)){
    fclose($user_fp);
}
else{
    fclose($user_fp);
}
?>
<h2>
<form method="get" action="./ev.php" name="myform">
<table height="500px" width="150px">
<tr><td><input type="hidden" name="command" value="<?=$chC?>">
<input type="radio" name="order" value="u" onclick="document.myform.submit();" id="u">
<label for="u" class="label">2階<br>^</label></td>
<td><input type="radio" name="order" value="d" onclick="document.myform.submit();" id="d">
<label for="d" class="label">1階<br>^</label></td></tr>
<tr><td><input type="radio" name="order" value="o" onclick="document.myform.submit();" id="o">
<label for="o" class="label">開</label></td>
<td><input type="radio" name="order" value="c" onclick="document.myform.submit();" id="c">
<label for="c" class="label">閉</label></td></tr>
<tr><td><input type="radio" name="order" value="Y" onclick="document.myform.submit();" id="Y">
<label for="Y" class="label">2階<br>呼</label></td>
<td><input type="radio" name="order" value="H" onclick="document.myform.submit();" id="H">
<label for="H" class="label">2階<br>閉</label></td></tr>
<tr><td><input type="radio" name="order" value="y" onclick="document.myform.submit();" id="y">
<label for="y" class="label">1階<br>呼</label></td>
<td><input type="radio" name="order" value="h" onclick="document.myform.submit();" id="h">
<label for="h" class="label">1階<br>閉</label></td></tr>
<tr><td><input type="radio" name="order" value="s" onclick="document.myform.submit();" id="s">
<label for="s" class="label">非常<br>停止</label></td>
<td><input type="radio" name="order" value="r" onclick="document.myform.submit();" id="r">
<label for="r" class="label">復歸</label></td></tr>
<tr><td><input type="radio" name="order" value="q" onclick="document.myform.submit();" id="q">
<label for="q" class="label">切</label></td>
<td><input type="radio" name="order" value="N" onclick="document.myform.submit();" checked
id="N">
<label for="N" class="label">無<br>信号</label></td></tr>
</table>
</form>
</h2>
<h3 align="center">LIFT</h3>
</td>
<td width="500px">
<iframe height="0px" width="0px" src="./ev_power.php"></iframe>
<iframe height="560px" width="430px" src="./ev_disp.php"></iframe><br>
</td>
<td width="150px">
<h2>

```

```

<form method="get" action="./ev.php" name="usermyform">
<table height="500px" width="150px"><tr>
<td>
<input type="hidden" name="usercommand" value="<?=$userChC?>">
<input type="radio" name="userorder" value="_03_" onclick="document.usermyform.submit();"
id="_03_">
<label for="_03_" class="label">3階<br>^</label></td>
<td><input type="radio" name="userorder" value="_3D_" onclick="document.usermyform.submit();"
id="_3D_">
<label for="_3D_" class="label">3階<br>▽</label></td>
</tr><tr>
<td><input type="radio" name="userorder" value="_02_" onclick="document.usermyform.submit();"
id="_02_">
<label for="_02_" class="label">2階<br>^</label></td>
<td><input type="radio" name="userorder" value="_2U_" onclick="document.usermyform.submit();"
id="_2U_">
<label for="_2U_" class="label">2階<br>△</label></td>
</tr><tr>
<td><input type="radio" name="userorder" value="_01_" onclick="document.usermyform.submit();"
id="_01_">
<label for="_01_" class="label">1階<br>^</label></td>
<td><input type="radio" name="userorder" value="_2D_" onclick="document.usermyform.submit();"
id="_2D_">
<label for="_2D_" class="label">2階<br>▽</label></td>
</tr><tr>
<td><input type="radio" name="userorder" value="_0o_" onclick="document.usermyform.submit();"
id="_0o_">
<label for="_0o_" class="label">開</label></td>
<td><input type="radio" name="userorder" value="_1U_" onclick="document.usermyform.submit();"
id="_1U_">
<label for="_1U_" class="label">1階<br>△</label></td>
</tr><tr>
<td><input type="radio" name="userorder" value="_0c_" onclick="document.usermyform.submit();"
id="_0c_">
<label for="_0c_" class="label">閉</label></td>
<td><input type="radio" name="userorder" value="_e_" onclick="document.usermyform.submit();"
id="_e_">
<label for="_e_" class="label">地震</label></td>
</tr><tr>
<td><input type="radio" name="userorder" value="_0L_" onclick="document.usermyform.submit();"
id="_0L_">
<label for="_0L_" class="label">開<br>延長</label></td>
<td><input type="radio" name="userorder" value="_NN_" onclick="document.usermyform.submit();"
checked id="_NN_">
<label for="_NN_" class="label">無<br>信号</label></td>
</tr>
</table>
</form>
</h2>
<h3 align="center">ELEVATOR</h3>
</td>
</tr><tr>
<form method="get" action="./ev.php" name="twinmyform">
<td width="75px">
<h2>
<table height="500px" width="75px"><tr>
<td>
<input type="radio" name="userorder" value="_03l" onclick="document.twinmyform.submit();"
id="_03l">
<label for="_03l" class="label">3階<br>L^</label></td>

```



```

</tr><tr>
<td><input type="radio" name="userorder" value="_02l" onclick="document.twinmyform.submit();"
id="_02l">
<label for="_02l" class="label">2階<br>L^</label></td>
</tr><tr>
<td><input type="radio" name="userorder" value="_01l" onclick="document.twinmyform.submit();"
id="_01l">
<label for="_01l" class="label">1階<br>L^</label></td>
</tr><tr>
<td><input type="radio" name="userorder" value="_00l" onclick="document.twinmyform.submit();"
id="_00l">
<label for="_00l" class="label">開L</label></td>
</tr><tr>
<td><input type="radio" name="userorder" value="_0cl" onclick="document.twinmyform.submit();"
id="_0cl">
<label for="_0cl" class="label">閉L</label></td>
</tr><tr>
<td><input type="radio" name="userorder" value="_0Ll" onclick="document.twinmyform.submit();"
id="_0Ll">
<label for="_0Ll" class="label">開L<br>延長</label></td>
</tr>
</table>
</h2>
<h3 align="center">LEFT</h3>
</td>
<td width="500px">
<iframe height="580px" width="430px" src="./ev_twin_disp.php"></iframe><br>
</td>
<td width="150px">
<h2>
<table height="500px" width="150px"><tr>
<td>
<input type="hidden" name="TwinChange" value="<?=$TwinChangeCh?>">
<input type="radio" name="userorder" value="_3Df" onclick="document.twinmyform.submit();"
id="_3Df">
<label for="_3Df" class="label">3階<br>▽</label></td>
<td><input type="radio" name="userorder" value="_03r" onclick="document.twinmyform.submit();"
id="_03r">
<label for="_03r" class="label">3階<br>R^</label></td>
</tr><tr>
<td><input type="radio" name="userorder" value="_2Uf" onclick="document.twinmyform.submit();"
id="_2Uf">
<label for="_2Uf" class="label">2階<br>△</label></td>
<td><input type="radio" name="userorder" value="_02r" onclick="document.twinmyform.submit();"
id="_02r">
<label for="_02r" class="label">2階<br>R^</label></td>
</tr><tr>
<td><input type="radio" name="userorder" value="_2Df" onclick="document.twinmyform.submit();"
id="_2Df">
<label for="_2Df" class="label">2階<br>▽</label></td>
<td><input type="radio" name="userorder" value="_01r" onclick="document.twinmyform.submit();"
id="_01r">
<label for="_01r" class="label">1階<br>R^</label></td>
</tr><tr>
<td><input type="radio" name="userorder" value="_1Uf" onclick="document.twinmyform.submit();"
id="_1Uf">
<label for="_1Uf" class="label">1階<br>△</label></td>
<td><input type="radio" name="userorder" value="_00r" onclick="document.twinmyform.submit();"
id="_00r">
<label for="_00r" class="label">開R</label></td>

```

```
</tr><tr>
<td><input type="radio" name="userorder" value="_e_f" onclick="document.twinmyform.submit();"
id="_e_f">
<label for="_e_f" class="label">地震</label></td>
<td><input type="radio" name="userorder" value="_0cr" onclick="document.twinmyform.submit();"
id="_0cr">
<label for="_0cr" class="label">閉R</label></td>
</tr><tr>
<td><input type="radio" name="userorder" value="_NNf" onclick="document.twinmyform.submit();"
checked id="_NNf">
<label for="_NNf" class="label">無<br>信号</label></td>
<td><input type="radio" name="userorder" value="_0Lr" onclick="document.twinmyform.submit();"
id="_0Lr">
<label for="_0Lr" class="label">開R<br>延長</label></td>
</tr>
</table>
</h2>
<h3 align="center">RIGHT</h3>
</td>
</form>
</tr></table>
</body>
</html>
```

```

<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">

<?php
function Receiver()
{
    $strOrder = "N__";
    if(($fp = fopen("http://hidemine.xsrv.jp/hidemine/ev9/Order.txt", "r")) == false){
    }
    else if($strOrder = fgets($fp)){
        fclose($fp);
    }
    else{
        fclose($fp);
    }
    if(($strOrder != "N__") && ($strOrder != "_NN")){
        echo $strOrder;
        if(($fp = fopen("./Order.txt", "w")) == false){ // for Linux
        //if(($fp = fopen("./Order.txt", "wb")) == false){ // for Windows
        }
        else if(fwrite($fp, $strOrder)){
            fclose($fp);
        }
        else{
            fclose($fp);
        }
    }
    return;
}
?>
</head>
<body>
<?php
Receiver();
?>
<form method="get" action="./ev_receiver.php" name="myform">
<script type="text/javascript">
<!--
setInterval("document.myform.submit()",2000);
//-->
</script>
</form>
</body>
</html>

```

```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
</head>
<body>
<iframe height="0px" width="0px" src="./ev_power.php"></iframe><br>
<iframe src="./ev_receiver.php"></iframe><br>
<iframe height="560px" width="430px" src="./ev_disp.php"></iframe>
<iframe height="630px" width="820px" src="http://hidemine.xsrv.jp/hidemine/ev9/ev.php" title="ev9"></iframe>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title>mysqldump.php</title>
</head>
<body>
<p>mysqldump.php</p>
<?php
exec('mysqldump ev001 -u pi -praspberry > /home/pi/ダウンロード/ev9/mysqldump.sql');
exec('sudo chmod 777 /home/pi/ダウンロード/ev9/mysqldump.sql');
?>
<script>
<!--
window.opener=self;
window.close();
//-->
</script>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="ja">

<head>
  <meta charset="UTF-8">
  <title>ev9</title>
</head>
<body>
<iframe height="1260px" width="820px" src="./ev.php" title="ev9"></iframe><br>
</body>
</html>
```

```
OUTPUT machine.abs
PRINT message.map
INPUT messagemap, main, EV_Simulator, EV_Puls, EV_Controller, EV_Input, EV_Display, EV_OpenClose,
EV_UpDown, EV_File, EV_Time, Timer, Panel, sci, lcd, usb
LIB c:\h8\akic\c38hab
START R(0FFE000), P(200), D(99C0), C(9A00)
ROM (D, R)
EXIT
```

```

# makefile.mak
CC = bcc32
LL = ilink32
INCLUDE = -I"C:\borland\bcc55\Include"
LIB = -L"C:\borland\bcc55\Lib"
CFLAGS = -O2 -w -tWC -D"USE_BCC"
LFLAGS = /Tpe
TARGET = main.exe
OBJS = Panel.obj Timer.obj EV_Time.obj EV_File.obj EV_UpDown.obj EV_OpenClose.obj EV_Display.obj
EV_Input.obj EV_Controller.obj EV_Puls.obj EV_Simulator.obj EV_Queue.obj EV_Twin1_Queue.obj
EV_Twin2_Queue.obj EV_UserController.obj EV_Twin1_Controller.obj EV_Twin2_Controller.obj
EV_UserSimulator.obj EV_Twin1_Simulator.obj EV_Twin2_Simulator.obj main.obj
$(TARGET): $(OBJS)
    $(LL) $(LFLAGS) $(LIB) \
    $(OBJS) c0x32.obj,$(TARGET),,cw32.lib import32.lib
main.obj : main.c main.h EV_Twin_Simulator.h EV_UserSimulator.h EV_Twin_Controller.h EV_UserController.h
EV_Twin_Queue.h EV_Queue.h EV_Simulator.h EV_Puls.h EV_Controller.h EV_Input.h EV_Display.h
EV_OpenClose.h EV_UpDown.h EV_File.h EV_Time.h Timer.h Panel.h C.h
    $(CC) $(CFLAGS) -c main.c
EV_Twin2_Simulator.obj : EV_Twin2_Simulator.c EV_Twin_Simulator.h EV_File.h EV_Time.h Timer.h Panel.h
C.h
    $(CC) $(CFLAGS) -c EV_Twin2_Simulator.c
EV_Twin1_Simulator.obj : EV_Twin1_Simulator.c EV_Twin_Simulator.h EV_File.h EV_Time.h Timer.h Panel.h
C.h
    $(CC) $(CFLAGS) -c EV_Twin1_Simulator.c
EV_UserSimulator.obj : EV_UserSimulator.c EV_UserSimulator.h EV_File.h EV_Time.h Timer.h Panel.h C.h
    $(CC) $(CFLAGS) -c EV_UserSimulator.c
EV_Twin2_Controller.obj : EV_Twin2_Controller.c EV_Twin_Controller.h EV_Queue.h EV_File.h EV_Time.h
Timer.h Panel.h C.h
    $(CC) $(CFLAGS) -c EV_Twin2_Controller.c
EV_Twin1_Controller.obj : EV_Twin1_Controller.c EV_Twin_Controller.h EV_Queue.h EV_File.h EV_Time.h
Timer.h Panel.h C.h
    $(CC) $(CFLAGS) -c EV_Twin1_Controller.c
EV_UserController.obj : EV_UserController.c EV_UserController.h EV_Queue.h EV_File.h EV_Time.h Timer.h
Panel.h C.h
    $(CC) $(CFLAGS) -c EV_UserController.c
EV_Twin2_Queue.obj : EV_Twin2_Queue.c EV_Twin_Queue.h EV_File.h EV_Time.h Timer.h Panel.h C.h
    $(CC) $(CFLAGS) -c EV_Twin2_Queue.c
EV_Twin1_Queue.obj : EV_Twin1_Queue.c EV_Twin_Queue.h EV_File.h EV_Time.h Timer.h Panel.h C.h
    $(CC) $(CFLAGS) -c EV_Twin1_Queue.c
EV_Queue.obj : EV_Queue.c EV_Queue.h EV_File.h EV_Time.h Timer.h Panel.h C.h
    $(CC) $(CFLAGS) -c EV_Queue.c
EV_Simulator.obj : EV_Simulator.c EV_Simulator.h EV_Display.h EV_File.h EV_Time.h Timer.h Panel.h C.h
    $(CC) $(CFLAGS) -c EV_Simulator.c
EV_Puls.obj : EV_Puls.c EV_Puls.h EV_File.h EV_Time.h Timer.h Panel.h C.h
    $(CC) $(CFLAGS) -c EV_Puls.c
EV_Controller.obj : EV_Controller.c EV_Controller.h EV_OpenClose.h EV_UpDown.h EV_File.h EV_Time.h
Timer.h Panel.h C.h
    $(CC) $(CFLAGS) -c EV_Controller.c
EV_Input.obj : EV_Input.c EV_Input.h EV_File.h EV_Time.h Timer.h Panel.h C.h
    $(CC) $(CFLAGS) -c EV_Input.c
EV_Display.obj : EV_Display.c EV_Display.h EV_File.h EV_Time.h Timer.h Panel.h C.h
    $(CC) $(CFLAGS) -c EV_Display.c
EV_OpenClose.obj : EV_OpenClose.c EV_OpenClose.h EV_UpDown.h EV_File.h EV_Time.h Timer.h Panel.h
C.h
    $(CC) $(CFLAGS) -c EV_OpenClose.c
EV_UpDown.obj : EV_UpDown.c EV_UpDown.h EV_File.h EV_Time.h Timer.h Panel.h C.h

```



```
$(CC) $(CFLAGS) -c EV_UpDown.c
EV_File.obj : EV_File.c EV_File.h Panel.h C.h
$(CC) $(CFLAGS) -c EV_File.c
EV_Time.obj : EV_Time.c EV_Time.h Timer.h Panel.h C.h
$(CC) $(CFLAGS) -c EV_Time.c
Timer.obj : Timer.c Timer.h Panel.h C.h
$(CC) $(CFLAGS) -c Timer.c
Panel.obj : Panel.c Panel.h C.h
$(CC) $(CFLAGS) -c Panel.c
```

clean:

```
del *.obj
del main.tds
del main.ilc
del main.ild
del main.ilf
del main.ils
```

```

@rem build.bat
C:
set bccDir="C:\borland\bcc55\Bin"
set akih8asmDir="c:\h8\akiasm"
set akih8cDir="c:\h8\akic"
set path=%bccDir%;%path%
set path=%asih8cDir%;%asih8asmDir%;%path%
set CurrentDir="%~dp0"
cd %CurrentDir%
del error.txt
make -f makefile.mak >> error.txt
make -f makefile.mak clean >> error.txt
cc38h.exe -cpu=300ha -include=%asih8cDir% usb.c >> error.txt
cc38h.exe -cpu=300ha -include=%asih8cDir% sci.c >> error.txt
cc38h.exe -cpu=300ha -include=%asih8cDir% lcd.c >> error.txt
cc38h.exe -cpu=300ha -include=%asih8cDir% Panel.c >> error.txt
cc38h.exe -cpu=300ha -include=%asih8cDir% Timer.c >> error.txt
cc38h.exe -cpu=300ha -include=%asih8cDir% EV_Time.c >> error.txt
cc38h.exe -cpu=300ha -include=%asih8cDir% EV_File.c >> error.txt
cc38h.exe -cpu=300ha -include=%asih8cDir% EV_UpDown.c >> error.txt
cc38h.exe -cpu=300ha -include=%asih8cDir% EV_OpenClose.c >> error.txt
cc38h.exe -cpu=300ha -include=%asih8cDir% EV_Display.c >> error.txt
cc38h.exe -cpu=300ha -include=%asih8cDir% EV_Input.c >> error.txt
cc38h.exe -cpu=300ha -include=%asih8cDir% EV_Controller.c >> error.txt
cc38h.exe -cpu=300ha -include=%asih8cDir% EV_Puls.c >> error.txt
cc38h.exe -cpu=300ha -include=%asih8cDir% EV_Simulator.c >> error.txt
cc38h.exe -cpu=300ha -include=%asih8cDir% main.c >> error.txt
a38h.exe messagemap.src >> error.txt
l38h.exe -subcommand=makefile.sub >> error.txt
c38h.exe machine.abs >> error.txt
del %CurrentDir%*.obj >> error.txt
del %CurrentDir%machine.abs >> error.txt
@rem set path="C:\Program Files\Java\jdk1.8.0_111\bin\";%path%
@rem javac EV_Simulator_JFile.java
@rem javac EV_Simulator_Display.java
@rem javac EV_Simulator.java
@rem javac EV_User_Input.java
@rem javac EV_Input.java
error.txt
exit

```

@rem httpd apache install for Windows

C:

cd c:\Apache24\bin

@rem C\server\php フォルダを作成してください

@rem C\server\php フォルダに ev9 プロジェクトフォルダを入れてください

@rem C\server\php\ev9\ev_power.php の exec を for Windows に 選択してください

@rem C\server\php\ev9\ev_receiver.php の fopen を for Windows に 選択してください

@rem Apache24 を ダウンロードして 解凍して Cドライブ の直下に配置してください

@rem php を ダウンロードして 解凍して Cドライブ の直下に配置してください

@rem install 実行前に あらかじめ c:\Apache24\conf\httpd.conf

@rem を編集しておいてください。

@rem サーバーの port は ServerName localhost:52080 です

@rem サーバーの root は DocumentRoot "c:/server/php" です

@rem 次の4行を末行に追加してください

@rem LoadModule php7_module "c:/php/php7apache2_4.dll"

@rem AddHandler application/x-httpd-php .php

@rem # configure the path to php.ini

@rem PHPIniDir "c:/php"

httpd -k install -n "Apache24php" -f "c:\Apache24\conf\httpd.conf"

```
@rem httpd apache uninstall for Windows
```

```
c:
```

```
cd c:\Apache24\bin
```

```
httpd -k uninstall -n "Apache24php"
```

@rem httpd apache start for Windows

C:
cd c:\Apache24\bin

@rem C\server\php\ev9\ev_power.php の exec を for Windows に 選択してください
@rem C\server\php\ev9\ev_receiver.php の fopen を for Windows に 選択してください

@rem ApacheのWebページを更新して公開
@rem start httpd.exe -k reload
net start "Apache24php"

@rem Windows の ev を起動する場合
@rem start http://localhost:52080/ev9/

@rem Xserver の ev を Windows の ev で follow する場合
@rem start http://localhost:52080/ev9/ev_follow.php

@rem httpd apache stop for Windows

c:

cd c:\Apache24\bin

net stop "Apache24php"

```
@rem mysqldump.bat
start http://192.168.115.200/ev9/mysqldump.php
ping 127.0.0.1
ping 127.0.0.1
ping 127.0.0.1
ping 127.0.0.1
ping 127.0.0.1
ping 127.0.0.1
bitsadmin.exe /TRANSFER html dl http://192.168.115.200/ev9/mysqldump.sql C:\server\php\ev9\mysqldump.sql
```

```
set path="C:\Program Files\Java\jre1.8.0_121\bin\";%path%  
java EV_Input  
exit
```



```
set path="C:\Program Files\Java\jre1.8.0_121\bin\";%path%  
java EV_Simulator  
exit
```

```
@rem start.bat
C:
set workpath="%~dp0"
@rem set path="C:\Program Files\Java\jdk1.8.0_111\bin\";%path%
cd %workpath%
start "" "EV_Simulator.bat"
start "" "EV_Input.bat"
start "" "main.exe"
exit
```

```
#!/bin/bash
cd `dirname $0`
find $PWD -name CentOSBuild.bash
rm ./CentOSError.txt
gcc -D"USE_CENTOS" -Wall -o CentOSExe.out main.c EV_UserSimulator.c EV_UserController.c EV_Queue.c
EV_Log.c EV_Simulator.c EV_Puls.c EV_Controller.c EV_Input.c EV_Display.c EV_OpenClose.c
EV_UpDown.c EV_File.c EV_Time.c Timer.c Panel.c -I/usr/include/mysql -L/usr/lib/mysql -lmysqlclient
&>>./CentOSError.txt
exit
```

```
#!/bin/bash
cd `dirname $0`
find $PWD -name CentOSStart.bash
./CentOSExe.out
exit
```

```
#!/bin/bash
cd `dirname $0`
find $PWD -name RaspberryPi3ModelBBuild.bash
rm ./RaspberryPi3ModelBError.txt
gcc -D"USE_RASPBIAN" -Wall -o RaspberryPi3ModelBExe.out main.c EV_UserSimulator.c
EV_UserController.c EV_Queue.c EV_Log.c EV_Simulator.c EV_Puls.c EV_Controller.c EV_Input.c
EV_Display.c EV_OpenClose.c EV_UpDown.c EV_File.c EV_Time.c Timer.c Panel.c -I/usr/local/include -
L/usr/local/lib -lwiringPi -lmariadbclient &>>./RaspberryPi3ModelBError.txt
cd ..
sudo chmod -R 777 ./ev9
cd ./ev9
exit
```

```
#!/bin/bash
cd `dirname $0`
find $PWD -name RaspberryPi3ModelBStart.bash
./RaspberryPi3ModelBExe.out
exit
```

```
#!/bin/bash
cd `dirname $0`
find $PWD -name XServerBuild.bash
rm ./XServerError.txt
gcc -D"USE_XSERVER" -Wall -o XServerExe.out main.c EV_Twin2_Simulator.c EV_Twin1_Simulator.c
EV_UserSimulator.c EV_Twin2_Controller.c EV_Twin1_Controller.c EV_UserController.c EV_Twin2_Queue.c
EV_Twin1_Queue.c EV_Queue.c EV_Simulator.c EV_Puls.c EV_Controller.c EV_Input.c EV_Display.c
EV_OpenClose.c EV_UpDown.c EV_File.c EV_Time.c Timer.c Panel.c &>>./XServerError.txt
exit
```

```
#!/bin/bash
cd `dirname $0`
find $PWD -name XServerStart.bash
./XServerExe.out
exit
```


実行環境状態ファイル

yynnyynn

yynnyynndddd

yynnyynndddd

yynnyynndddd

NN

NN

NN

N

N_

出力ファイル

Start	Length	Name	Class
0001:00401000	000016234H	_TEXT	CODE
0002:00418000	000005CE8H	_DATA	DATA
0003:0041DCE8	000000CD0H	_BSS	BSS
0004:00000000	0000000A4H	_TLS	TLS

LINK COMMAND LINE

LNK -subcommand=makefile.sub

LINK SUBCOMMANDS

OUTPUT machine.abs
 PRINT message.map
 INPUT messagemap, main, EV_Simulator, EV_Puls, EV_Controller, EV_Input, EV_Display, EV_OpenClose,
 EV_UpDown, EV_File, EV_Time, Timer, Panel, sci, lcd, usb
 LIB c:\h8\akic\c38hab
 START R(0FFE000), P(200), D(99C0), C(9A00)
 ROM (D, R)
 EXIT

*** LINKAGE EDITOR LINK MAP LIST ***

SECTION NAME	START	END	LENGTH
	UNIT NAME		MODULE NAME

ATTRIBUTE : CODE NOSHR

V	H'00000000	- H'000000F3	H'000000F4
		messagemap	messagemap

* TOTAL ADDRESS * H'00000000 - H'000000F3 H'000000F4

ATTRIBUTE : CODE NOSHR

P	H'00000200	- H'000002AF	H'000000B0
		messagemap	messagemap
	H'000002B0	- H'000012DB	H'0000102C
		main	main
	H'000012DC	- H'0000179B	H'000004C0
		EV_Simulator	EV_Simulator
	H'0000179C	- H'00001B17	H'0000037C
		EV_Puls	EV_Puls
	H'00001B18	- H'0000249F	H'00000988
		EV_Controller	EV_Controller
	H'000024A0	- H'00002EE9	H'00000A4A
		EV_Input	EV_Input
	H'00002EEA	- H'00002F3B	H'00000052
		EV_Display	EV_Display


```

H'00002F3C - H'000037A3 H'00000868
              EV_OpenClose          EV_OpenClose
H'000037A4 - H'00003FEB H'00000848
              EV_UpDown             EV_UpDown
H'00003FEC - H'000043DF H'000003F4
              EV_File                EV_File
H'000043E0 - H'00004537 H'00000158
              EV_Time                EV_Time
H'00004538 - H'00004A9F H'00000568
              Timer                  Timer
H'00004AA0 - H'00004BF5 H'00000156
              Panel                  Panel
H'00004BF6 - H'00004CC7 H'000000D2
              sci                    sci
H'00004CC8 - H'00004F05 H'0000023E
              lcd                    lcd
H'00004F06 - H'000059F7 H'00000AF2
              usb                    usb
H'000059F8 - H'00005A31 H'0000003A
              rand                   rand
H'00005A32 - H'00005A8F H'0000005E
              sprintf                sprintf
H'00005A90 - H'00005AB9 H'0000002A
              strcmp                 strcmp

```

H8/300H LINKAGE EDITOR (Evaluation software) Ver.1.0 PAGE : 2

*** LINKAGE EDITOR LINK MAP LIST ***

SECTION NAME	START - END	LENGTH
	UNIT NAME	MODULE NAME

ATTRIBUTE : CODE NOSHR

P	H'00005ABA - H'00005AD5	H'0000001C
	strcpy	strcpy
	H'00005AD6 - H'00005AF1	H'0000001C
	strlen	strlen
	H'00005AF2 - H'00005B21	H'00000030
	vsprintf	vsprintf
	H'00005B22 - H'00005DFB	H'000002DA
	addd3	addd3
	H'00005DFC - H'0000602D	H'00000232
	divd3	divd3
	H'0000602E - H'000060A7	H'0000007A
	dtol3	dtol3
	H'000060A8 - H'000060B3	H'0000000C
	eqd3	eqd3
	H'000060B4 - H'000060C3	H'00000010
	ged3	ged3
	H'000060C4 - H'000060D3	H'00000010
	ltd3	ltd3
	H'000060D4 - H'00006119	H'00000046

H'0000611A	-	ltod3 H'00006137	ltod3 H'0000001E
		mv83	mv83
H'00006138	-	H'0000615F	H'00000028
		mvn3	mvn3
H'00006160	-	H'0000616D	H'0000000E
		ned3	ned3
H'0000616E	-	H'0000618F	H'00000022
		spregld3	spregld3
H'00006190	-	H'000061B7	H'00000028
		spregsv3	spregsv3
H'000061B8	-	H'00007F65	H'00001DAE
		_fmtout	_fmtout
H'00007F66	-	H'00008021	H'000000BC
		cmpd3	cmpd3
H'00008022	-	H'00008047	H'00000026
		divl3	divl3
H'00008048	-	H'00008067	H'00000020
		mull3	mull3
H'00008068	-	H'0000844D	H'000003E6
		_dti	_dti
H'0000844E	-	H'000085F1	H'000001A4
		_its	_its
H'000085F2	-	H'0000864B	H'0000005A
		memcpy	memcpy
H'0000864C	-	H'00008687	H'0000003C
		divul3	divul3

H8/300H LINKAGE EDITOR (Evaluation software) Ver.1.0 PAGE : 3

*** LINKAGE EDITOR LINK MAP LIST ***

SECTION NAME	START	-	END	LENGTH	MODULE NAME
	UNIT NAME				

ATTRIBUTE : CODE NOSHR

P	H'00008688	-	H'000086AF	H'00000028	
			_allzero	_allzero	
	H'000086B0	-	H'000087A7	H'000000F8	
			_calcnpw	_calcnpw	
	H'000087A8	-	H'0000884B	H'000000A4	
			_log10	_log10	
	H'0000884C	-	H'000088C3	H'00000078	
			_lsfts	_lsfts	
	H'000088C4	-	H'000088F1	H'0000002E	
			_pow5	_pow5	
	H'000088F2	-	H'0000896B	H'0000007A	
			_rsfts	_rsfts	
	H'0000896C	-	H'00008A17	H'000000AC	
			_sub	_sub	
	H'00008A18	-	H'00008ABB	H'000000A4	
			_unpack	_unpack	

```

H'00008ABC - H'00008AF9 H'0000003E
             memcmp                      memcmp
H'00008AFA - H'00008B81 H'00000088
             _mult64                    _mult64
H'00008B82 - H'00008CE3 H'00000162
             _power                      _power
H'00008CE4 - H'00008DCD H'000000EA
             _rnd                        _rnd
H'00008DCE - H'00008E69 H'0000009C
             _setsbit                    _setsbit
H'00008E6A - H'00008F6F H'00000106
             frexp                       frexp
H'00008F70 - H'000090A5 H'00000136
             modf                        modf
H'000090A6 - H'000090C7 H'00000022
             dslc3                       dslc3
H'000090C8 - H'000090E9 H'00000022
             dsruc3                      dsruc3
H'000090EA - H'0000911F H'00000036
             itod3                       itod3
H'00009120 - H'0000940D H'000002EE
             muld3                        muld3
H'0000940E - H'0000945F H'00000052
             _duchek                     _duchek
H'00009460 - H'000094B1 H'00000052
             _lsft                       _lsft
H'000094B2 - H'0000963F H'0000018E
             _mult                       _mult
H'00009640 - H'000096DB H'0000009C
             _pow10                      _pow10

```

H8/300H LINKAGE EDITOR (Evaluation software) Ver.1.0 PAGE : 4

*** LINKAGE EDITOR LINK MAP LIST ***

SECTION NAME	START	END	LENGTH
	UNIT NAME	MODULE NAME	

ATTRIBUTE : CODE NOSHR

P	H'000096DC	- H'00009723	H'00000048
		_add	_add
	H'00009724	- H'00009753	H'00000030
		memset	memset

* TOTAL ADDRESS * H'00000200 - H'00009753 H'00009554

ATTRIBUTE : DATA NOSHR ROM

D	H'000099C0	- H'000099C0	H'00000000
		messagemap	messagemap
	H'000099C0	- H'000099CF	H'00000010

usb usb

* TOTAL ADDRESS * H'000099C0 - H'000099CF H'00000010

ATTRIBUTE : DATA NOSHR

```

C          H'00009A00 - H'00009C84 H'00000285
              main                main
H'00009C86 - H'00009CB4 H'0000002F
              EV_Simulator        EV_Simulator
H'00009CB6 - H'00009CC0 H'0000000B
              EV_Puls              EV_Puls
H'00009CC2 - H'00009CE9 H'00000028
              EV_Controller        EV_Controller
H'00009CEA - H'00009D59 H'00000070
              EV_Input             EV_Input
H'00009D5A - H'00009E19 H'000000C0
              EV_Display           EV_Display
H'00009E1A - H'00009E65 H'0000004C
              EV_OpenClose        EV_OpenClose
H'00009E66 - H'00009EA8 H'00000043
              EV_UpDown           EV_UpDown
H'00009EAA - H'00009F1B H'00000072
              EV_File              EV_File
H'00009F1C - H'00009F23 H'00000008
              EV_Time              EV_Time
H'00009F24 - H'00009F5B H'00000038
              Timer                Timer
H'00009F5C - H'00009F66 H'0000000B
              Panel                Panel
H'00009F68 - H'0000A021 H'000000BA
              usb                  usb

```

H8/300H LINKAGE EDITOR (Evaluation software) Ver.1.0 PAGE : 5

*** LINKAGE EDITOR LINK MAP LIST ***

SECTION NAME	START - END	LENGTH
	UNIT NAME	MODULE NAME

ATTRIBUTE : DATA NOSHR

```

C          H'0000A022 - H'0000A029 H'00000008
              _fmtout              _fmtout
H'0000A02A - H'0000A129 H'00000100
              _ctype                _ctype
H'0000A12A - H'0000A1B1 H'00000088
              _its                    _its
H'0000A1B2 - H'0000A1B9 H'00000008
              _log10                 _log10
H'0000A1BA - H'0000A299 H'000000E0
              _pow5                  _pow5

```

```

H'0000A29A - H'0000A39D H'00000104
           _power                _power
H'0000A39E - H'0000A3A5 H'00000008
           frexp                  frexp
H'0000A3A6 - H'0000A3AD H'00000008
           modf                   modf

```

* TOTAL ADDRESS * H'00009A00 - H'0000A3AD H'000009AE

ATTRIBUTE : DATA NOSHR RAM

```

R           H'00FFE000 - H'00FFE000 H'00000000
           messagemap                messagemap
H'00FFE000 - H'00FFE00F H'00000010
           usb                        usb

```

* TOTAL ADDRESS * H'00FFE000 - H'00FFE00F H'00000010

ATTRIBUTE : DATA NOSHR

```

B           H'00FFE010 - H'00FFE011 H'00000002
           messagemap                messagemap
H'00FFE012 - H'00FFE245 H'00000234
           main                      main
H'00FFE246 - H'00FFE259 H'00000014
           EV_File                   EV_File
H'00FFE25A - H'00FFE49B H'00000242
           Timer                     Timer
H'00FFE49C - H'00FFE51B H'00000080
           Panel                     Panel
H'00FFE51C - H'00FFE56B H'00000050
           sci                       sci
H'00FFE56C - H'00FFE5AB H'00000040
           lcd                        lcd

```

H8/300H LINKAGE EDITOR (Evaluation software) Ver.1.0 PAGE : 6

*** LINKAGE EDITOR LINK MAP LIST ***

SECTION NAME	START	END	LENGTH	MODULE NAME
	UNIT NAME	UNIT NAME		

ATTRIBUTE : DATA NOSHR

```

B           H'00FFE5AC - H'00FFE87F H'000002D4
           usb                        usb
H'00FFE880 - H'00FFE8BB H'0000003C
           _fmtout                   _fmtout
H'00FFE8BC - H'00FFE8BF H'00000004
           _rnext                    _rnext
H'00FFE8C0 - H'00FFE8C1 H'00000002

```

_errno

_errno

* TOTAL ADDRESS * H'00FFE010 - H'00FFE8C1 H'000008B2

H8/300H LINKAGE EDITOR (Evaluation software) Ver.1.0 PAGE : 1

*** LINKAGE EDITOR EXTERNALLY DEFINED SYMBOLS LIST ***

SYMBOL NAME	ADDR	TYPE
\$ADDD\$3	H'00005BEC	DAT
\$CMPD\$3	H'00007F66	DAT
\$DIVD\$3	H'00005E96	DAT
\$DIVL\$3	H'00008022	DAT
\$DIVUL\$3	H'0000864C	DAT
\$DSL\$3	H'000090A6	DAT
\$DSRUC\$3	H'000090C8	DAT
\$DTOL\$3	H'0000602E	DAT
\$EQD\$3	H'000060A8	DAT
\$GED\$3	H'000060B4	DAT
\$ITOD\$3	H'000090EA	DAT
\$LTD\$3	H'000060C4	DAT
\$LTOD\$3	H'000060D4	DAT
\$MULD\$3	H'000091D6	DAT
\$MULL\$3	H'00008048	DAT
\$MV8\$3	H'0000611A	DAT
\$MVN\$3	H'00006138	DAT
\$NED\$3	H'00006160	DAT
\$SUBD\$3	H'00005BBC	DAT
\$sp_regld\$3	H'0000616E	DAT
\$sp_regsv\$3	H'00006190	DAT
_Checkfmove	H'000044FA	ENT
_Clear	H'00004AA0	ENT
_ClearLCD	H'00004DE6	ENT
_Close	H'0000375C	ENT
_CloseMotor	H'00003210	ENT
_Cnt	H'00FFE012	DAT
_Command_Read	H'000041F6	ENT
_Command_Write	H'00004242	ENT
_Destroy	H'00001146	ENT
_DisableInterrupt	H'000002AC	DAT
_Disp	H'00002F3A	ENT
_DispInput	H'00002EEA	ENT
_DispUSBPort	H'00005016	ENT
_Door	H'00002F3C	ENT
_Down	H'00003FA4	ENT
_DownMotor	H'00003A78	ENT
_EV_AddressDataSet	H'000018EC	ENT
_EV_AddressSet	H'00001808	ENT
_EV_Controller	H'00001B18	ENT
_EV_DataSet	H'0000187A	ENT
_EV_EnableSet	H'000017F2	ENT
_EV_File	H'00004040	ENT
_EV_Input	H'0000250C	ENT
_EV_Puls	H'000019BC	ENT

_EV_Set	H'0000179C	ENT
_EV_Simulator	H'000012DC	ENT
_EV_Time	H'000043E0	ENT
_EnableInterrupt	H'000002A8	DAT
_GetChar	H'000024A0	ENT
_GetCurrentTime	H'00004442	ENT

H8/300H LINKAGE EDITOR (Evaluation software) Ver.1.0 PAGE : 2

*** LINKAGE EDITOR EXTERNALLY DEFINED SYMBOLS LIST ***

SYMBOL NAME	ADDR	TYPE
_GetPermit	H'000044DC	ENT
_GetSCI	H'00004C26	ENT
_GetSW	H'0000128A	ENT
_H8init	H'000012B0	ENT
_Init	H'00000E8E	ENT
_InitITU	H'00004A24	ENT
_InitLCD	H'00004D76	ENT
_InitSCI	H'00004BF6	ENT
_InitUSB	H'00004F1C	ENT
_InterruptITU0	H'00004A5A	ENT
_LCDOut4	H'00004D1E	ENT
_Limit_Read	H'0000439C	ENT
_LocateLCD	H'00004E26	ENT
_Motor_Read	H'0000434E	ENT
_Motor_Write	H'00004316	ENT
_OnCloseMotor	H'0000322C	ENT
_OnController	H'00001CB8	ENT
_OnDownMotor	H'00003A94	ENT
_OnInitWaitDoorChangeLog	H'00003498	ENT
_OnInitWaitPositionChangeLog	H'00003CF2	ENT
_OnInput	H'00002678	ENT
_OnOpenMotor	H'00002FCC	ENT
_OnPuls	H'000019FA	ENT
_OnSimulator	H'000013C6	ENT
_OnUpMotor	H'00003834	ENT
_OnWaitCloseDoorChangeLog	H'00003640	ENT
_OnWaitDownPositionChangeLog	H'00003E8E	ENT
_OnWaitOpenDoorChangeLog	H'0000356C	ENT
_OnWaitUpPositionChangeLog	H'00003DC0	ENT
_Open	H'00003714	ENT
_OpenMotor	H'00002FB0	ENT
_PermitCommand_Read	H'00004166	ENT
_PermitCommand_Write	H'000041B2	ENT
_PermitTurnOpen_Read	H'00004286	ENT
_PermitTurnOpen_Write	H'000042D2	ENT
_Position	H'000037A4	ENT
_PrintF	H'00004AC8	ENT
_PrintLCD	H'00004E4A	ENT
_PrintSCI	H'00004C46	ENT
_PutLCD	H'00004DFA	ENT
_PutSCI	H'00004C16	ENT
_Read	H'000040D4	ENT

_ReadString	H'0000412E	ENT
_Repaint	H'000004C0	ENT
_Run	H'0000052C	ENT
_ScanSCI	H'00004C36	ENT
_SetCurrentTime	H'00004424	ENT
_SetLED	H'0000123C	ENT
_SetPermit	H'000044A8	ENT
_SleepMSec	H'0000454A	ENT
_Start	H'00004810	ENT

H8/300H LINKAGE EDITOR (Evaluation software) Ver.1.0 PAGE : 3

*** LINKAGE EDITOR EXTERNALLY DEFINED SYMBOLS LIST ***

SYMBOL NAME	ADDR	TYPE
_Stop	H'0000483E	ENT
_Thread_Start	H'000048D6	ENT
_Thread_Toggle	H'00004914	ENT
_Thread_checkAllDelete	H'0000485C	ENT
_Thread_checkStayAnother	H'00004876	ENT
_Thread_getThread	H'0000489C	ENT
_Up	H'00003F5C	ENT
_UpMotor	H'00003818	ENT
_WaitDoorChangeLog	H'00003470	ENT
_WaitPositionChangeLog	H'00003CD8	ENT
_WaitSecond	H'00004486	ENT
_Wait_ms	H'0000451C	ENT
_Write	H'0000404A	ENT
_WriteString	H'0000409C	ENT
__add	H'000096DC	ENT
__allzero	H'00008688	ENT
__calcpw	H'000086B0	ENT
__ctype	H'0000A02A	DAT
__dti	H'00008068	ENT
__duchek	H'0000940E	ENT
__errno	H'00FFE8C0	DAT
__fmtout	H'000061B8	ENT
__its	H'0000844E	ENT
__log10	H'000087A8	ENT
__lsft	H'00009460	ENT
__lsfts	H'0000884C	ENT
__mult	H'000094B2	ENT
__mult64	H'00008AFA	ENT
__pow10	H'00009640	ENT
__pow5	H'000088C4	ENT
__power	H'00008B82	ENT
__rnd	H'00008CE4	ENT
__rnext	H'00FFE8BC	DAT
__rsfts	H'000088F2	ENT
__setsbit	H'00008DCE	ENT
__sub	H'0000896C	ENT
__unpack	H'00008A18	ENT
__cntrl	H'00FFE0B2	DAT
_countUpNextRun	H'00004620	ENT

delete	H'000047D8	ENT
_frexp	H'00008E6A	ENT
_getClock	H'00004538	ENT
_get_inbufflen	H'000059D0	ENT
_get_outbufflen	H'000059E4	ENT
_i_cnt	H'00FFE016	DAT
_in	H'00FFE04E	DAT
_initWOVI	H'00004A86	ENT
_init_usbbuff	H'00005834	ENT
_j_cnt	H'00FFE018	DAT
_main	H'000002B0	ENT
_memcpy	H'00008ABC	ENT

H8/300H LINKAGE EDITOR (Evaluation software) Ver.1.0 PAGE : 4

*** LINKAGE EDITOR EXTERNALLY DEFINED SYMBOLS LIST ***

SYMBOL NAME	ADDR	TYPE
_memcpy	H'000085F2	ENT
_memset	H'00009724	ENT
_modf	H'00008F70	ENT
_new_EV_Status	H'00003FEC	ENT
_new_Thread	H'00004640	ENT
_nextRun	H'000045DC	ENT
_puls	H'00FFE1AA	DAT
_rand	H'000059F8	ENT
_read_buff	H'00005972	ENT
_read_outbuff	H'00005914	ENT
_s	H'00FFE04A	DAT
_simu	H'00FFE1BE	DAT
_sprintf	H'00005A32	ENT
_status	H'00FFE246	DAT
_strcmp	H'00005A90	ENT
_strcpy	H'00005ABA	ENT
_strlen	H'00005AD6	ENT
_th	H'00FFE01A	DAT
_th1	H'00FFE022	DAT
_th101	H'00FFE29E	DAT
_th102	H'00FFE2BC	DAT
_th111	H'00FFE2DA	DAT
_th112	H'00FFE2F8	DAT
_th113	H'00FFE316	DAT
_th114	H'00FFE334	DAT
_th119	H'00FFE352	DAT
_th120	H'00FFE370	DAT
_th121	H'00FFE38E	DAT
_th122	H'00FFE3AC	DAT
_th123	H'00FFE3CA	DAT
_th130	H'00FFE3E8	DAT
_th131	H'00FFE406	DAT
_th141	H'00FFE424	DAT
_th142	H'00FFE442	DAT
_th143	H'00FFE460	DAT
_th144	H'00FFE47E	DAT

_th19	H'00FFE032	DAT
_th20	H'00FFE036	DAT
_th41	H'00FFE03A	DAT
_th42	H'00FFE03E	DAT
_th43	H'00FFE042	DAT
_th44	H'00FFE046	DAT
_usb_int	H'0000507A	ENT
_vsprintf	H'00005AF2	ENT
_wovi	H'00004A82	ENT
_woviClock	H'00FFE25A	DAT
_woviInit	H'000049F2	ENT
_woviRun	H'00004958	ENT
_woviThreadFirst	H'00FFE262	DAT
_woviThreadLast	H'00FFE280	DAT
_write_buff	H'000058AE	ENT

H8/300H LINKAGE EDITOR (Evaluation software) Ver.1.0 PAGE : 5

*** LINKAGE EDITOR EXTERNALLY DEFINED SYMBOLS LIST ***

SYMBOL NAME	ADDR	TYPE
_write_inbuff	H'00005852	ENT

S00E00006D616368696E65204D4F540D
S1130000000022A0000022A0000022A67
S1130010000022A0000022A0000022A2D
S1130020000022A0000022A0000022A1D
S10B0030000022A0000022A6D
S1130038000022A0000022A0000022C03
S1130048000022A0000022A0000022AF5
S1130058000022A0000022A0000022AE5
S10B0068000022A0000022AF5
S1130070000022A0000022A0000022ACD
S1130080000022A0000022A0000022ABD
S1130090000022A0000022A0000022AAD
S10B00A0000022A0000022AFD
S11300A8000022A0000022A0000022A95
S11300B8000022A0000022A0000022A85
S11300C8000022A0000022A0000022A75
S10B00D8000022A0000022AC5
S11300E0000022A0000022A0000022A5D
S10700F0000022ADD
S11302007A0700FFFF107A0000099C07A0100FF0F
S1130210E0007A0200FFE0101F92587000086C0B98
S1130220689B0B7140F25E0002B0567001006DF0E6
S113023001006DF101006DF201006DF301006DF439
S113024001006DF501006DF65E00506801006D76EA
S113025001006D7501006D7401006D7301006D7215
S113026001006D7101006D70567001006DF00100A9
S11302706DF101006DF201006DF301006DF40100F9
S11302806DF501006DF65E004A4801006D760100D0
S11302906D7501006D7401006D7301006D720100D5
S11302A06D7101006D705670063F547004C0547038
S11302B05E00617E7A370000000A790B0001193372
S11302C00FF47A0600FFE1FE19550B5579257FFFE0
S11302D04DF85C000FDC5E004BE45E004D640D38AE
S11302E00D305C000F580D380DB05C000F500D3809
S11302F0790000025C000F460D38790000035C00B2
S11303000F3C5E004F0A7FF472507FF570505E0021
S113031002A80D3228F117506DF07A0000009A0000
S113032001006DF05E004C340B970B877A000000E0
S11303309A0F01006DF05E004E380B9718886EC857
S113034000036EC800026EC8000168C87A0000FF8F
S1130350E04A5E003FDA5E004A747A00000007D08C
S11303605E0045387900001E5E00462E01006FF0E6
S113037000047900001F5E00462E0F8501006F7098
S113038000045E0047FE0FD05E0047FE5E004A7029
S10403905E0B
S113039100484A0D0046F45E004A8E7A0100009A35
S11303A11B0D305E004AB67A00000007D05E00459F
S11303B1385E004A8E19550D505C000ECE0D0D0DA1
S11303C15117F10AC16819D90117D1660147540DB3
S11303D1B80D505C000E660D5009B06DF07A000047
S11303E1009A2C01006DF00FE05E005A200B970B71
S11303F18701006DF67A0000009A3101006DF05E0D
S1130401004C340B970B970FE05E005AC409B00DF3
S1130411010FE05E00589C01006DF65E004E380B43
S11304219740080D380D505C000E120DD017F50FD3
S1130431C10AD168980B557925000458D0FF785E1D
S1130441004C240D0047165E004C1417D00D055EB9

S1130451004DE868ED0DB10FE05E00589C5E005958
S1130461BE0D004738790100400FE05E0059600D71
S1060471057A0105
S113047400009A3501006DF15E004E380B970100C0
S11304846DF65E004E380B9701006DF65E004C343A
S11304940B970D510FE05E00589C0D287900000363
S11304A45C000D960D20795000010D0219550B5572
S11304B4792527104DF85A0003B654705E00617E07
S11304C47A0500FFE01219665E004A8E19EE400EAB
S11304D47A0100009A370D605E004AB60B5E6950DC
S11304E41D0E4DEC7A0100009A390D605E004AB688
S11304F47A0100009A3D0D605E004AB619EE400E83
S11305047A0100009A370D605E004AB60B5E6F50A5
S10B051400021D0E4DEA7A01FD
S113051C00009A3F0D605E004AB65E00615C547049
S113052C5E00617E7A370000000A7A030000000146
S113053C7A04000007D019550F8618886EF800034B
S113054C6EF800026EF8000168F86960792000010A
S113055C46365C00FF5E7A0000FFE01269010B5126
S113056C69815E0059E617F07902000901D0532026
S113057C7918000A79000064528017F00F810FE09C
S113058C5E0045CA5A000E8869607920000246361F
S113059C5C00FF207A0000FFE01469010B516981B4
S11305AC5E0059E617F07902000901D0532079183F
S11305BC000A79000064528017F00F810FE05E008F
S11305CC45CA5A000E8869607920000B586000B642
S11305DC01006F6000127A200000000146205E00CB
S11305EC4A8E7A0100009A430D505E004AB67A0196
S11305FC0000076C0FE05E00460E5A000E880100E7
S109060C6F6000127A206A
S10A06120000000246265E12
S1130619004A8E7A0100009A540D505E004AB67A58
S11306290100009A5B0D505E004AB60FE05E004878
S11306392C5A000E8801006F6000127A2000000016
S113064903461E5E004A8E7A0100009A660D505ECB
S1130659004AB67A0100005DC0FE05E00460E4051
S11306692401006F6000127A200000000446165E20
S1130679004A8E7A0100009A770D505E004AB60F40
S1130689E05E00482C5A000E8869607920000C58F6
S11306996000A801006F6000127A20000000014683
S10A06A9205E004A8E7A0176
S11306B000009A880D505E004AB67A0100000D488A
S11306C00FE05E00460E5A000E8801006F600012B4
S11306D07A200000000246205E004A8E7A01000064
S11306E09A990D505E004AB67A0100000D480FE05A
S11306F05E00460E5A000E8801006F6000127A20D9
S113070000000003461E5E004A8E7A0100009AAA8A
S11307100D505E004AB67A0100000D480FE05E00FE
S1130720460E401C5E004A8E7A0100009ABB0D50B3
S11007305E004AB60FE05E00482C0FE05E4D
S113073D0047C65A000E8869607920000D58600085
S113074DA801006F6000127A200000000146205EB0
S113075D004A8E7A0100009AC30D505E004AB67AA4
S113076D01000013EC0FE05E00460E5A000E8801E7
S113077D006F6000127A200000000246205E004ADE
S113078D8E7A0100009AD40D505E004AB67A0100AC
S113079D0013EC0FE05E00460E5A000E8801006F49

S11307AD6000127A2000000003461E5E004A8E7A16
S11307BD0100009AE50D505E004AB67A0100001360
S11207CDEC0FE05E00460E401C5E004A8E7A0180
S11307DC00009AF60D505E004AB60FE05E00482CFE
S11307EC0FE05E0047C65A000E8869607920000E40
S11307FC586000E401006F6000127A2000000001D1
S113080C46205E004A8E7A0100009AFE0D505E006F
S113081C4AB67A01000017700FE05E00460E5A00CC
S113082C0E8801006F6000127A200000000246401F
S113083C5E004A8E7A0100009B0F0D505E004AB693
S113084C7A01000017700FE05E00460E7A0100007B
S113085C9B160D505E004AB67900000B5E0048C42F
S10C086C0F867A01000005DC5E31
S113087500460E5A000E8801006F6000127A2000B0
S1130885000003461E5E004A8E7A0100009B210D7F
S1130895505E004AB67A01000017700FE05E00460D
S11308A50E403801006F6000127A200000000446F4
S11308B52A7900000B5E00488A0F8247060FA05E67
S11308C50047C67A0100009B320D505E004AB60F01
S11308D5E05E00482C0FE05E0047C65A000E8869AB
S11308E5607920001346440FB10FE05E0045CA1935
S11308F5DD0DD05C0009900DD117F10FF20A926856
S11309052ADA0117D2660247160DD0791000145E54
S11309150049027A01000003E80FE05E0045CA0BB7
S11309255D792D00044DCA5A000E8869607920004F
S10809351446187A01CD
S113093A00009B390D505E004AB60FC10FE05E00FE
S113094A460E5A000E8869607920001546187A0106
S113095A00009B3B0D505E004AB60FC10FE05E00DC
S113096A460E5A000E8869607920001646187A01E5
S113097A00009B3D0D505E004AB60FC10FE05E00BA
S113098A460E5A000E8869607920001746187A01C4
S113099A00009B3F0D505E004AB60FC10FE05E0098
S11309AA460E5A000E8869607920001E465E0100D1
S11309BA6F600012462401006F6000120B70010081
S11309CA6FE0001228D6E80E38D67A01000003E851
S10609DA0FE05ECA
S11309DD0045CA5A000E8801006F6000127A20008C
S11309ED0000015860049401006F6000121B700138
S11309FD006FE000127FD670007A01000003E80F4C
S1130A0DE05E0045CA5A000E8869607920001F58C0
S1130A1D6003E601006F600012460C1A910FE05E51
S1130A2D00460E5A000E8801006F6000127A2000F6
S1130A3D00000146247A0100009A3D0D505E004AE4
S1130A4DB67A0100009B410D505E004AB60FC10FEF
S1130A5DE05E00460E5A000E8801006F6000127AA8
S1130A6D2000000002463019DD0DD00B505E00460C
S1130A7D2E0DD217F210321032010078A06BA000A8
S1130A8DFFFE01A0B5D792D00024DDE0FB10FE05E15
S1130A9D00460E5A000E0201006F6000127A20000C
S10A0AAD00000346566B2015
S1130AB400FFE0127920000D4D1A790000016BA0AC
S1130AC400FFE01601006F6000120B7001006FE07D
S1130AD4001240246B2000FFE0147920000D4D1810
S1130AE4790000026BA000FFE01601006F600012A2
S1130AF40B7001006FE000120FB10FE05E0045CAF6
S1130B045A000E0201006F6000127A2000000004F4

S1130B1446685E004A8E6B2000FFE01679200001D0
S1130B24460E7A0100009B520D505E004AB64018EF
S1130B346B2000FFE01679200002460C7A010000C6
S1130B449B630D505E004AB601006B2000FFE01A60
S10B0B545E0047C601006B209F
S1130B5C00FFE01E5E0047C601006F6000120B70C1
S1130B6C01006FE000120FC10FE05E0045CA5A008E
S1130B7C0E0201006F6000127A2000000005461C73
S1130B8C5E004A8E7A0100009A1B0D505E004AB635
S1130B9C0FC10FE05E00460E5A000E0201006F609B
S1130BAC00127A2000000006461C5E004A8E7A0171
S1130BBC00009B740D505E004AB60FC10FE05E003F
S1130BCC460E5A000E0201006F6000127A200000DC
S1130BDC000746365E004A8E19DD0DD07910000BE6
S1130BEC5E00462E0DD217F210321032010078A09F
S1050BFC6BA0E9
S1130BFE00FFE0220B5D792D00044DDC0FB10FE0F9
S1130C0E5E00460E5A000E0201006F6000127A203B
S1130C1E000000846245E00486479200002460E58
S1130C2E01006F6000120B7001006FE000120FB134
S1130C3E0FE05E0045CA5A000E0201006F600012FB
S1130C4E7A2000000009460C0FC10FE05E00460E2D
S1130C5E5A000E0201006F6000127A200000000A93
S1130C6E461C5E004A8E7A0100009A1B0D505E00F0
S1130C7E4AB60FC10FE05E00460E5A000E02010087
S1130C8E6F6000127A200000000B461C5E004A8E35
S1130C9E7A0100009B850D505E004AB60FC10FE02E
S1040CAE5EE4
S1130CAF00460E5A000E0201006F6000127A2000F8
S1130CBF00000C4634790000135E00462E01006BD2
S1130CCFA000FFE0325E0047FE790000145E00468D
S1130CDF2E01006BA000FFE0365E0047FE0FB10F41
S1130CEFE05E00460E5A000E0201006F6000127A9A
S1130CFF200000000D46305E004864792000034653
S1130D0F1A01006B2000FFE0325E0047C601006F3F
S1130D1F6000120B7001006FE000120FB10FE05E65
S1130D2F0045CA5A000E0201006F6000127A2000BC
S1110D3F00000E460C0FC10FE05E00460E5A78
S1130D4D000E0201006F6000127A200000000F46B2
S1130D5D1C5E004A8E7A0100009A1B0D505E004AFC
S1130D6DB60FC10FE05E00460E5A000E0201006F72
S1130D7D6000127A2000000010461A5E004A8E7A37
S1130D8D0100009B960D505E004AB60FC10FE05E49
S1130D9D00460E406001006F6000127A20000000D3
S1130DAD114652790000295E00462E01006BA0000A
S1130DBDFFE03A7900002A5E00462E01006BA00089
S1130DCDFFE03E7900002B5E00462E01006BA00074
S10B0DDDFFE0427900002C5EE7
S1130DE500462E01006BA000FFE0460FE05E0047C2
S1130DF5C67900001E5E00488A5E0047C65A000E8B
S1130E0588696079200029461A7A01000000640F79
S1130E15E05E0045CA0FE17A0000FFE04E5E002662
S1130E2566406069607920002A461A7A010000004D
S1130E35640FE05E0045CA0FE17A0000FFE0AA5E99
S1130E45001CBA403E69607920002B461A7A0100DE
S1130E5500006E0FE05E0045CA0FE17A0000FFE176
S1130E65A25E0019FC401C69607920002C46140F12

S1130E75C10FE05E0045CA0FE17A0000FFE1B65EEF
S1130E850013C87A170000000A5E00615C54705EA7
S1130E9500617E0F86790400097A0500FFE0126977
S1130EA560792000014626190069D05E0059E617CE
S1130EB5F001D053407918000A7900001E528017BB
S1130EC5F00F810FE05E0045CA5A0011426960794F
S1130ED5200002462819006FD000025E0059E6176C
S1130EE5F001D053407918000A7900001E5280178B
S1130EF5F00F810FE05E0045CA5A001142696C7913
S1130F052C00034D36792C00084E3069601B5017B1
S1130F15F010300A85190069D05E0059E617F00113
S1130F25D053407918000A790000C8528017F00F92
S1130F35810FE05E0045CA5A0011427A0400004A57
S1130F45B6195569607920000B461A7A0100009B92
S1130F55A70D505D407A01000005DC0FE05E0046F9
S1120F650E5A00114269607920000C461A7A0176
S1130F7400009BAF0D505D407A0100000BB80FE0F9
S1130F845E00460E5A00114269607920000D462422
S1130F947A0100009A3D0D505D407A0100009BB92F
S1130FA40D505D407A01000011940FE05E00460E7F
S1130FB45A00114269607920000E461A7A01000032
S1130FC49BC10D505D407A01000017700FE05E0075
S1130FD4460E5A0011427A03000007D06960792053
S1130FE4001446245E004A8E7A0100009BCB0D5008
S1130FF45D407A0100009A3D0D505D400FB10FE052
S11310045E00460E5A001142696079200015462499
S10410145E7A
S1131015004A8E7A0100009BDC0D505D407A010089
S1131025009A3D0D505D400FB10FE05E00460E5A2C
S113103500114269607920001646245E004A8E7AC3
S11310450100009BED0D505D407A0100009A3D0DB6
S1131055505D400FB10FE05E00460E5A0011426924
S1131065607920001746245E004A8E7A0100009BB2
S1131075FE0D505D407A0100009A3D0D505D400F15
S1131085B10FE05E00460E5A0011426960792000F7
S10D10951E460AF8FF38D438D65A75
S113109F00114269607920001F46187A0000000092
S11310AF1101006FE000121A910FE05E0045CA5A5A
S11310BF00114269607920002946187A0100000067
S11310CF640FE05E0045CA7A0000FFE04E5E002524
S11310DF0E406069607920002A461A7A01000000E9
S11310EF640FE05E0045CA0FE17A0000FFE0AA5EDD
S11310FF001B1A403E69607920002B461A7A0100C3
S113110F00006E0FE05E0045CA0FE17A0000FFE1B9
S113111FA25E0019BE401C69607920002C46140F93
S113112FB10FE05E0045CA0FE17A0000FFE1B65E42
S112113F0012DE5E00615C54705E00617E7A0414
S113114E00004AB619660F8569507920000B4610C8
S113115E5E004A8E7A0100009C0F0D605D40404494
S113116E69507920000C460C7A0100009C1A0D6020
S113117E5D40403069507920000D460C7A01000025
S113118E9C240D605D40401C69507920000E46146E
S113119E7A0100009A3D0D605D407A0100009C2E9D
S11311AE0D605D40695079200014461A5E004A8E28
S11311BE7A0100009C3F0D605D407A0100009A3D6C
S11311CE0D605D404064695079200015461A5E003B
S11311DE4A8E7A0100009C500D605D407A0100003A

S11311EE9A3D0D605D404042695079200016461AC3
S11311FE5E004A8E7A0100009C610D605D407A01AB
S113120E00009A3D0D605D40402069557925001719
S113121E46185E004A8E7A0100009C730D605D4095
S113122E7A0100009A3D0D605D405E00615C547072
S113123E6DF66DF50D067909000128D617500D953B
S113124E0CE91A094B04101540F866050D88461271
S113125E1A0E4B04101940F80D9029D6148939D65D
S113126E40141A0E4B04101940F8795900FF0D90D3
S113127E29D6168939D60D506D756D7654706DF667
S113128E0D0628D31750790100011A0E4B041011C5
S113129E40F86610470419114004790100010D103E
S11312AE6D765470F80638ECF8FF38C038C11888DC
S11312BE38D8F8FF38C8188838DBF8FF38C9F8DF94
S11312CE38D0F8FF38CDF8F038D1F8FF38D4547051
S1139A00435055204D4F444520253032580A000C11
S1139A1052656164792133303532004E4558542004
S1139A20202020202020202020202000737725754F
S1139A300025730A000C0020003C313E000A003C64
S1139A40323E003C313E31737420202020202000
S1139A50202020003C313E326E64003C313E537482
S1139A606F70202020003C313E3372642020202080
S1139A70202020202020003C313E53746F70202092
S1139A80202020202020003C323E3173742020EF
S1139A9020202020202020003C323E326E6420F3
S1139AA02020202020202020003C323E337264DE
S1139AB0202020202020202020003C323E5374F0
S1139AC06F70003C333E3173742020202020200F
S1139AD0202020003C333E326E642020202020B2
S1139AE020202020003C333E337264202020209D
S1099AF0202020202000CD
S1139AF63C333E53746F70003C343E317374202004
S1139B062020202020202020003C343E326E64009A
S1139B163C313E53746172742020003C343E3372F0
S1139B2664202020202020202020003C343E5387
S1139B36746F00300031003200330054687265617F
S1139B466420526561647920474F2100474F414C99
S1139B56213C313E574F4E202020202000474F41C5
S1139B664C213C323E574F4E202020202000436F8D
S1139B76756E745570202020202020202000544C
S1139B866F67676C65202020202020202020007E
S1139B9648656C6C6F2045562020202020200D
S1139BA6003C313E496E6974003C323E496E69742D
S1139BB62020003C333E496E6974003C343E496EB6
S1139BC669742020003C303E496E6974202020B1
S1139BD62020202020003C313E496E69742020203D
S1099BE6202020202020B6
S1139BEC003C323E496E69742020202020202026
S1139BFC20003C333E496E697420202020202015
S1139C0C2020003C313E44657374726F79003C3202
S1139C1C3E44657374726F003C333E4465737472D7
S1139C2C6F003C343E44657374726F792020209E
S1139C3C2020003C303E44657374726F792020E1
S1139C4C202020003C313E44657374726F7920D0
S1139C5C20202020003C323E44657374726F7920BF
S1139C6C202020202020003C333E44657374726F07
S10C9C7C79202020202020200083

S11312DE5E00617E0F860F957A00000000200AE003
S11312EE0FD15E0043CE7A00000000200AE05E00BC
S11312FE441201006FE600027A00000000060AE0C5
S113130E01006FE000087A000000000C0AE0010003
S113131E6FE0000E7A00000000120AE001006FE099
S113132E001C18886EE8001A7A050000402E7A0019
S113133E000000380AE05D507A000000003C0AE02D
S113134E5D507A00000000400AE05D507A00000014
S113135E00440AE05D5001006F60000201006DF071
S113136E7A00000000400AE07A0100009C865E00CD
S113137E40C20B9779200001473A790000096DF0BE
S113138E01006F60001C01006DF07A000000004444
S113139E0AE07A0100009C915E00411C0B970B87BB
S11313AE79200001470E7A01000000120AE16868F5
S11313BE5E002F285E00615C54705E00617E0F86B6
S10913CE0F9501006F60A2
S11313D4000E01006DF07A000000003C0AE07A017F
S11313E400009C9C5E0040C20B976E68000CA871C1
S11313F4460E5E004A8E0FD05E0047C65A00179809
S113140401006F60000201006DF07A0000000040EB
S11314140AE07A0100009C865E0040C20B977900C3
S113142400096DF001006F60001C01006DF07A008B
S10B1434000000440AE07A0104
S113143C00009C915E00411C0B970B8701006F60B1
S113144C000801006DF07A00000000380AE07A0110
S113145C00009CA95E0040C20B976E680006A8733F
S113146C461CF84E6DF07A000000003C0AE07A014D
S113147C00009C9C5E0040380B875A00178A6868F2
S113148CA87546366E680012A879587002D06E683B
S113149C0013A879460AF86E6EE800135A00176A0F
S11314AC6E680014A86E470E6E680015A86E46068B
S11314BCF8796EE800155A00176A6868A855462E25
S11314CC6E680012A879587002946E680013A8799C
S11314DC5870028A6E680014A86E4608F8796EE894
S11314EC001440066E680015A86E5A00176A6868E7
S11314FCA86A46466E680012A879460AF86E6EE82A
S113150C00125A00176A6E680013A879460AF86E1F
S113151C6EE800135A00176A6E680014A86E46082A
S109152CF8796EE80014DB
S1131532400E6E680015A86E4606F8796EE800152F
S11315425A00176A6868A86446366E680015A87957
S1131552587002146E680014A879460AF86E6EE891
S113156200145A00176A6E680013A86E470E6E685D
S11015720012A86E4606F8796EE800125AC2
S113157F00176A6868A844462E6E680015A8795844
S113158F7001D86E680014A879587001CE6E680088
S113159F13A86E4608F8796EE8001340066E6800CC
S11315AF12A86E5A00176A6868A86B46466E6800E1
S11315BF15A879460AF86E6EE800155A00176A6E79
S11315CF680014A879460AF86E6EE800145A0017DB
S11315DF6A6E680013A86E4608F8796EE800134028
S11315EF0E6E680012A86E4606F8796EE800125A5E
S11315FF00176A6868A86F46366E680016A8795890
S113160F7001586E680017A879460AF86E6EE800E5
S113161F175A00176A6E680018A86E470E6E680097
S113162F19A86E4606F8796EE800195A00176A680A
S113163F68A84F462E6E680016A8795870011C6E65

S113164F680017A879587001126E680018A86E46C3
S113165F08F8796EE8001840066E680019A86E5AEC
S109166F00176A6868A879
S11316756846466E680016A879460AF86E6EE80055
S1131685165A00176A6E680017A879460AF86E6E2F
S1131695E800175A00176A6E680018A86E4608F81E
S11316A5796EE80018400E6E680019A86E4606F8B4
S11316B5796EE800195A00176A6868A86346346E9C
S11316C5680019A8795870009C6E680018A87946B7
S11316D50AF86E6EE800185A00176A6E680017A8B4
S11316E56E470E6E680016A86E4606F8796EE8001A
S11316F51640726868A84346286E680019A879479A
S1131705646E680018A879475C6E680017A86E4672
S113171508F8796EE8001740066E680016A86E4053
S1131725446868A874463E6E680019A8794608F8A7
S11317356E6EE80019402E6E680018A8794608F801
S11317456E6EE80018401E6E680017A86E4608F80E
S1131755796EE80017400E6E680016A86E4606F807
S1081765796EE8001697
S113176A7A0000000120AE001006DF07A0000001E
S10E177A00440AE07A0100009C915E2D
S113178500408A0B977A01000000120AE168685E3F
S10C1795002F285E00615C547012
S1139C864D6F746F722E74787400004C696D69742D
S1139C962E7478740000436F6D6D616E642E747854
S1129CA67400005361666574792E74787400003E
S113179E0D0046067FD67200400A792000014604EA
S11317AE7FD670000D8846067FD67210400A7928C0
S11317BE000146047FD670100D1146067FD67220A7
S11317CE400A7921000146047FD670200D99460602
S11317DE7FD67230400A7929000146047FD67030D5
S11317EE7FD6724054707FD672007FD672107FD62A
S11317FE72207FD672307FD6704054705E00617E49
S113180E0F850D1C0D94790E000119660DC017F08E
S113181E01006F5100121F81461E0D690D610D6887
S113182E0D405C00FF6A01006F5000120B70010047
S113183E6FD000120D6E402E0B5C0DC017F0010021
S113184E6F5100121F81461E0D69790100010D684B
S113185E0D405C00FF3A01006F5000120B70010047
S113186E6FD000120D6E0DE05E00615C54705E0071
S113187E617E0F850D1C0D94790E000119660DC046
S109188E17F001006F5189
S113189400121F81461E0D690D610D480D605C0029
S11318A4FEF801006F5000120B7001006FD000129C
S11318B40D6E402E0B5C0DC017F001006F5100122A
S11318C41F81461E790900010D610D480D605C00FE
S11318D4FEC801006F5000120B7001006FD000129C
S11318E40D6E0DE05E00615C54705E00617E0F85D9
S11318F40F966F79001819110FE05C00FF080D00B3
S1131904587000B06F79001A790100020FE05C008F
S1131914FEF40D005870009C6F79001C79010004DB
S11319240FE05C00FEE00D00587000886F79001E24
S1131934790100060FE05C00FECC0D0047766F7959
S11319440020790100080FE05C00FF2C0D004764C0
S11319546F7900227901000A0FE05C00FF1A0D0081
S113196447526F7900247901000C0FE05C00FF08F3
S11319740D0047406F7900267901000E0FE05C00EB

S1091984FEF60D00472EE4
S113198A5C00FE661A8001006FE000126E580006C2
S113199AA871461A7FD672007FD672107FD672203C
S11319AA7FD672307FD672400FE05E0047C65E0074
S11319BA615C547001006DF60F8601006FE6000248
S11319CA7A00000000060AE001006FE000087A00CE
S11319DA000000100AE05E00402E7A000000000CAE
S11319EA0AE05E00402EF8FF38D438D601006D763F
S11319FA54705E00617E0F850F9401006F400012E0
S1131A0A46307A000000000C0AD001006F5100082A
S1131A1A5E0041E401006F50000201006DF07A009C
S1131A2A000000100AD07A0100009CB65E0040C292
S1131A3A0B9719EE790600016858A843587000AC51
S1131A4AA8444758A84F4770A8554738A863477C06
S1131A5AA8644740A868474CA86A471CA86B472C48
S1131A6AA86F474AA873470CA8744756A875470E28
S1091A7A5A001B146DFE6F
S1131A8040026DF66DFE400A6DFE6DF640046DF684
S1131A906DF66DFE401E6DFE6DFE6DF640166DF625
S1131AA06DFE6DF6400E6DFE6DF66DF640066DF63D
S1131AB06DF66DF66DFE40266DFE6DFE6DFE6DF6E8
S1131AC0401C6DF66DFE6DFE6DF640126DFE6DF6FB
S1131AD06DFE6DF640086DF66DF66DFE6DF66DF6F6
S1131AE06DF66DF66DFE0FC10FD05C00FE007A1728
S1131AF000000010401E6DFE6DFE6DF66DF66DF676
S1131B006DF66DF66DFE0FC10FD05C00FDE07A1728
S10D1B10000000105E00615C5470D9
S10E9CB64D6F746F722E747874000001
S1131B1A5E00617E0F860F9501006FE600240100C7
S1131B2A6F6000245E0037927A00000000280AE002
S1131B3A5E0038067A00000000300AE05E003A666A
S1131B4A7A00000000380AE05E003CC67A00000012
S1131B5A004C0AE001006FE000705E002F2A7A0051
S1131B6A000000740AE05E002F9E7A000000007CE9
S1131B7A0AE05E0031FE7A00000000840AE05E009B
S1131B8A345E7A00000000A20AE00FD15E0043CE61
S1131B9A7A050000402E7A00000000E00AE05D505A
S1131BAA7A00000000E40AE05D507A00000000E8D1
S1131BBA0AE05D507A00000000EC0AE05D507A000A
S1131BCA000000F00AE05D507A00000000F40AE029
S1131BDA5D507A00000000BA0AE001006FE000BC21
S1131BEA7A00000000C00AE001006FE000CA7A0030
S1131BFA000000CE0AE001006FE000D07A00000086
S1071C0A00D40AE015
S1131C0E01006FE000D67A00000000DA0AE001005E
S1131C1E6FE000DC7A00000000F40AE0F9735E0066
S1131C2E43047A00000000380AE001006F610024CB
S1131C3E5E003CE07A00000000840AE001006F6160
S1131C4E00705E003486790000096DF001006F604C
S1131C5E00CA01006DF07A00000000E40AE07A0188
S1131C6E00009CC25E00411C0B970B877A0000009C
S1131C7E00E80AE0F94E5E00423079200001472663
S1131C8E7A00000000EC0AE0F9635E0041A07920BF
S1131C9E000147127A00000000F00AE0F94E5E00E0
S1131CAE42C0792000015E00615C54705E00617E6B
S1131CBE790D0001F463FC4E19550F860F93010045
S1131CCE6F6000BC01006DF07A00000000E00AE0D6

S1051CDE7A0186
S1131CE000009CCD5E0040C20B9779200001587024
S1131CF007AA790000096DF001006F6000CA0100B6
S1131D006DF07A00000000E40AE07A0100009CC252
S1131D105E00411C0B970B877A00000000E80AE085
S1131D2001006F6100D05E0041E47920000158702A
S1131D30076A7A00000000EC0AE001006F6100D638
S1131D405E00415479200001587007507A0000006A
S1131D5000F00AE001006F6100DC5E00427479204C
S1131D600001587007366E6800CEA8485870051CED
S1131D70A859476EA86358700464A8645870022E6B
S1131D80A868587005D0A86F587003A4A8714718A5
S1131D90A8725870068AA8734732A875476AA8794B
S1131DA0587001E45A0024207A00000000F40AE08D
S1131DB0F9735E0043040FB05E0047C67900002F3D
S1131DC05E00488A5E0047C65A0024207A0000005D
S1091DD000A20AE00D5120
S1131DD65E0044965A0024205A00242001006F60B6
S1131DE6002401006F00000C6900792000014612EF
S1131DF601006F60007001006F00000C69015870EC
S1131E06061801006F60002401006F0000146900CA
S1131E1679200001464E01006F60007001006F00DB
S1131E2600146900792000015860034A7A00000013
S1131E3600EC0AE00C495E0041A07A00000000A213
S1131E460AE00DD15E0044967A00000000E80AE03D
S1061E560CC95E53
S1131E590042305E004A8E7A0100009CD95A002361
S1131E69CE01006F60007001006F00000C690079FA
S1131E7920000146767A00000000A20AE00D515EB7
S1131E890044967A00000000A20AE05E0044127A38
S1131E9900000000F00AE00CC95E0042C07A0000AD
S1131EA90000380AE07A37000000140FF17A0200C3
S1131EB90000145E0061267A00000000280AE07A17
S1131EC937000000080FF17A02000000085E006184
S1131ED92601006F6100BC01006F6000245E003FB2
S1131EE94A7A170000001C5A00242001006F600081
S1131EF97001006F00000C690146667A000000005A
S1131F09A20AE00D515E0044967A00000000A20A7D
S1131F19E05E0044127A00000000840AE07A370088
S1131F2900001E0FF17A020000001E5E0061267A8E
S1131F39000000007C0AE07A37000000080FF17AFC
S1091F4902000000085E27
S1131F4F00612601006F6100BC01006F6000705ECD
S1131F5F00374A7A1700000026401A7A0000000063
S1131F6FA20AE00D515E0044967A00000000A20A17
S1091F7FE05E0044125A6B
S1131F8500242001006F60002401006F0000146924
S1131F950079200001461201006F60007001006F97
S1131FA500000C69015870047201006F6000240180
S1131FB5006F00000C690079200001465001006F95
S1131FC560007001006F000014690079200001466C
S1131FD5387A00000000EC0AE00C495E0041A07A63
S1131FE500000000A20AE00DD15E0044967A0000CD
S1131FF50000E80AE00CC95E0042305E004A8E7AB2
S11320050100009CD95A0023CE5A00217C01006FA0
S113201560007001006F00000C6900792000014623
S1132025767A00000000A20AE00D515E0044967A1C

S11320350000000A20AE05E0044127A00000000DE
S1132045F00AE00CC95E0042C07A00000000380ABD
S1132055E07A37000000140FF17A02000000145EE5
S11320650061267A00000000300AE07A370000009C
S1082075080FF17A02DF
S113207A000000085E00612601006F6100BC0100D8
S113208A6F6000245E003F927A170000001C5A001A
S113209A242001006F60007001006F00000C6901C9
S11320AA46667A00000000A20AE00D515E004496DB
S11320BA7A00000000A20AE05E0044127A000000DF
S11320CA00840AE07A370000001E0FF17A0200004A
S11320DA001E5E0061267A000000007C0AE07A375F
S11020EA000000080FF17A02000000085EFC
S11320F700612601006F6100BC01006F6000705E24
S113210700374A7A1700000026401A7A00000000B9
S1132117A20AE00D515E0044967A00000000A20A6D
S1132127E05E0044125A00242001006F6000700132
S1132137006F00001469007920000146387A000017
S11321470000EC0AE00C495E0041A07A00000000A1
S1132157A20AE00DD15E0044967A00000000E80A67
S1132167E00CC95E0042305E004A8E7A0100009C93
S1132177D95A0023CE7A00000000A20AE00D515E6F
S11321870044967A00000000A20AE05E0044127A37
S113219700000000840AE07A370000001E0FF17A7E
S11321A702000001E5E0061267A00000000740A28
S11321B7E07A37000000080FF17A02000000085E9A
S11321C700612601006F6100BC01006F6000705E53
S11321D70037025A00234C7A00000000A20AE00DE0
S10821E7515E00449667
S10C21EC7A00000000A20AE05E83
S11321F500441201006F60007001006F00000C695C
S1132205007920000146387A00000000F00AE00C4E
S1132215C95E0042C07A00000000E80AE00CC95E0E
S11322250042307A00000000EC0AE00C495E0041F0
S1132235A05E004A8E7A0100009CD95A0023CE7A0B
S113224500000000840AE07A370000001E0FF17ACF
S1132255020000001E5E0061267A000000007C0A71
S1132265E07A37000000080FF17A02000000085EEB
S113227500612601006F6100BC01006F6000705EA4
S113228500374A5A00234C01006F60002401006F98
S11322950000146900792000015860017E7A00006E
S11322A50000A20AE00D515E0044967A000000008A
S11322B5A20AE05E00441201006F60007001006F26
S11322C500000C69007920000146387A00000000FF
S11322D5E80AE00CC95E0042307A00000000F00A0B
S10722E5E00CC95EDF
S11322E90042C07A00000000EC0AE00C495E00419C
S10F22F9A05E004A8E7A0100009CD95AB6
S11323050023CE7A00000000840AE07A370000003B
S11323151E0FF17A020000001E5E0061267A00009E
S113232500007C0AE07A37000000080FF17A02000A
S11323350000085E00612601006F6100BC01006FAB
S11323456000705E00374A7A17000000265A0024A1
S11323552001006F60002401006F00000C69007903
S1132365200001586000B47A00000000A20AE00DC5
S1132375515E0044967A00000000A20AE05E004424
S11323851201006F60007001006F00000C69007995

S1132395200001463C7A00000000E80AE00CC95E13
S11323A50042307A00000000F00AE00CC95E0042EA
S11323B5C07A00000000EC0AE00C495E0041A05E13
S11323C5004A8E7A0100009CD90D505E004AB64042
S11323D54A7A00000000840AE07A370000001E0FE5
S11323E5F17A020000001E5E0061267A00000000FB
S10823F57C0AE07A37C9
S11323FA000000080FF17A02000000085E0061265F
S113240A01006F6100BC01006F6000705E00374A13
S113241A7A17000000267A00000000A20AE05E0094
S113242A44307920000A4D6A01006F600070010090
S113243A6F00001469007920000146567A000000F3
S113244A00A20AE05E0044CA7920000146447A00E9
S113245A000000A20AE00D515E0044967A000000D3
S108246A00A20AE05E80
S113246F0044127A00000000F00AE0F96F5E0042A8
S113247FC07A00000000EC0AE00CC95E0041A07AAC
S113248F00000000E80AE00C495E0042305E006184
S106249F5C547017
S1139CC24C696D69742E747874000053616665740F
S1139CD2792E747874000048656C6C6F20455620A9
S10B9CE2202020202020200097
S11324A25E00617E1B970FF40C8D18886EC80003C3
S11324B26EC800026EC8000168C819660D605E002E
S11324C2128C0D0E0D6117F10AC16819D90117D1CA
S11324D2660147260D600C004620A800470EA8019E
S11324E2470EA802470EA803470E400EFD75400A89
S11324F2FD644006FD634002FD710B56792600041C
S11325024DBA0CD80B975E00615C54705E00617E1D
S11325120F867A00000000060AE001006FE0000A5D
S113252218886EE800097A000000001A0AE0010028
S11325326FE0001C7A00000000200AE001006FE057
S1132542002A18886EE800287A000000002E0AE0AC
S113255201006FE0003C18886EE8003A01006FE664
S113256200027A000000000E0AE001006FE0001092
S11325727A00000000140AE001006FE000167A05F9
S11325820000402E7A00000000400AE05D507A000D
S1092592000000440AE012
S11325985D507A00000000480AE05D507A000000B0
S11325A8004C0AE05D507A00000000500AE05D50DC
S11325B87A00000000580AE05D5001006F600002D5
S11325C801006DF07A00000000400AE07A01000083
S11325D89CEA5E0040C20B9779200001477A7A0093
S11325E8000000480AE001006F6100105E004154DA
S11325F87920000147627A00000000500AE00100D8
S11326086F6100165E00427479200001474A010099
S11326186F60001C01006DF07A00000000580AE0AA
S11326287A0100009CF65E0040C20B9779200001F6
S11326384726790000096DF001006F60002A010048
S11326486DF07A00000000540AE07A0100009D0151
S11326585E00411C0B970B875E00615C54705E0043
S1132668617EF474FC6FF568FD4E0F860F92790056
S113267800046DF001006F60000A01006DF07A003C
S1092688000000440AE01B
S113268E7A0100009D0C5E00411C0B970B8779208D
S113269E00015870082E7A0100009D1701006F602B
S10626AE000A5EBE

S11326B1005A7E0D0047147A0100009D1C01006F32
S11326C160000A5E005A7E0D0046140CD85C00FDC2
S11326D1D06EE80006F85F6EE800076EE8000818A0
S11326E1886EE8000901006F60000201006DF07A55
S11326F10000000400AE07A0100009CEA5E00400D
S1132701C20B9779200001587007C67A00000000B8
S1132711480AE001006F6100105E00415479200016
S113272101587007AC7A00000000500AE001006F05
S11327316100165E0042747920000158700792010E
S1132741006F60001C01006DF07A00000000580A60
S1132751E07A0100009CF65E0040C20B97792000ED
S1132761015870076C790000096DF001006F60007A
S11327712A01006DF07A00000000540AE07A01009A
S1132781009D015E00411C0B970B876E680006A834
S11327914E4712A85F4708A871470A400840066ED2
S10827A1680007A84ECB
S11327A67A01000000060AE16818A85F460A6E1857
S11327B60001A85F586006F66E680006A871587097
S11327C6008EA872474CA873586000AAF8736DF080
S11327D67A00000000400AE07A0100009CEA5E00ED
S11327E640380B877A00000000580AE0F9735E0050
S11327F64304F84E6EE800067A00000000440AE03F
S1132806F94E5E0042307A00000000480AE0F94EB5
S113281640360C586DF07A00000000400AE07A0159
S113282600009CEA5E0040380B87F84E6EE800060F
S11328367A00000000440AE0F94E5E0042307A0056
S1102846000000480AE0F9635E0041A05A5B
S1132853002EB47A00000000440AE0F9715E0042DE
S1132863307A0100009D21790000015E004AB60F12
S1132873A05E0047C65A002EB46E68000EA86358C4
S11328836004F219DD6E680006A84858700454A862
S11328935958700322A86358700088A864587001BC
S11328A3B0A868587003B0A86F4710A87558700094
S11328B3DAA879587002605A002EB46E680024A80F
S11328C379470E7A00000000500AE00CC95E00420B
S11328D3C06868A868463EF85968E86DF07A000056
S11328E30000400AE07A0100009CEA5E0040380BD6
S11328F3876E68001AA873461C6868A85946166E43
S1132903680027A86E460E7A00000000580AE00C00
S1132913595E0043047A00000000440AE00CC95ADC
S1132923002CD06E680024A879470E7A00000000BB
S1132933500AE00CC95E0042C06868A868463EF8C6
S10829435968E86DF086
S11329487A00000000400AE07A0100009CEA5E0079
S113295840380B876E68001AA873461C6868A85924
S113296846166E680024A86E460E7A0000000058CA
S11329780AE00C495E0043047A00000000440AE0C0
S1132988F9635A002CD06E680024A879470E7A00A0
S10C2998000000500AE00CC95EC6
S11329A10042C06868A8685860009AF85968E86DE1
S11329B1F07A00000000400AE07A0100009CEA5E20
S11329C10040380B876E68001AA87346266868A80A
S11329D15946206E680020A87946186E680024A81D
S11329E16E46107A00000000580AE00C495E00436D
S11329F10440526E68001AA873461E6868A85946B7
S1132A01186E680023A86E46107A00000000580A69
S1132A11E0F96A5E004304402C6E68001AA873460D

S1132A21246868A859461E6E680023A87946166E65
S1132A31680027A86E460E7A00000000580AE00CD1
S1132A41595E0043047A00000000440AE0F9755A14
S1132A51002CD06E680024A879470E7A000000008C
S1132A61500AE00CC95E0042C06868A8685860005B
S1132A719AF85968E86DF07A00000000400AE07A9C
S1132A810100009CEA5E0040380B876E68001AA8BB
S1092A917346266868A8E5
S1132A975946206E680023A87946186E680024A853
S1132AA76E46107A00000000580AE00C495E0043A6
S1132AB70440526E68001AA873461E6868A85946F0
S1132AC7186E680020A86E46107A00000000580AA6
S1132AD7E0F96B5E004304402C6E68001AA8734646
S1132AE7246868A859461E6E680020A87946166EA2
S1132AF7680027A86E460E7A00000000580AE00C0B
S1132B07595E0043047A00000000440AE0F9645A5E
S1132B17002CD07A01000000200AE16818A8794741
S1132B27126E180004A879470A7A0100009D265AF5
S1132B37002CF46868A868466CF85968E86DF07A61
S1132B470000000400AE07A0100009CEA5E0040B2
S1132B57380B876E68001AA873461E6868A859461B
S1132B67186E680020A86E46107A00000000580A05
S1072B77E0F96B5EB5
S1132B7B004304402C6E68001AA87346246868A8A7
S1132B8B59461E6E680020A87946166E680027A862
S1132B9B6E460E7A00000000580AE00C595E0043A3
S1132BAB047A00000000440AE0F9795A002CD07A29
S1132BBB01000000200AE16E180003A87947126E8A
S1132BCB180004A879470A7A0100009D405A002C8B
S1132BDBF46868A868466CF85968E86DF07A0000E9
S1132BEB0000400AE07A0100009CEA5E0040380BCB
S1132BFB876E68001AA873461E6868A85946186E34
S1132C0B680023A86E46107A00000000580AE0F90A
S1132C1B6A5E004304402C6E68001AA8734624684E
S1132C2B68A859461E6E680023A87946166E68007D
S1132C3B27A86E460E7A00000000580AE00C595E76
S1132C4B0043047A00000000440AE0F95940766E11
S1132C5B680020A879470A7A0100009D265A002CA8
S1092C6BF46E680024A8CA
S1132C7179470E7A00000000500AE00CC95E004259
S1132C81C06868A868463EF85968E86DF07A0000A4
S1132C910000400AE07A0100009CEA5E0040380B24
S1132CA1876E68001AA873461C6868A85946166E91
S1132CB1680024A86E460E7A00000000580AE00C52
S1132CC1495E0043047A00000000440AE00C595EA7
S1132CD10042307A00000000480AE00CD95E00414E
S1052CE1A05AF4
S1132CE3002EB46E680023A87947107A0100009D73
S1132CF3400DD05E004AB65A002EB46E680024A875
S1132D0379470E7A00000000500AE00CC95E0042C6
S1132D13C06868A868463EF85968E86DF07A000011
S1132D230000400AE07A0100009CEA5E0040380B91
S1132D33876E68001AA873461C6868A85946166EFE
S1132D43680024A86E460E7A00000000580AE00CBF
S1132D53495E0043047A00000000440AE0F9485E38
S1132D630042307A00000000480AE00CD95E0041BB
S1132D73A05A002EB46E680014A86F586001327A0B

S1132D830100000200AE16818A8795860008C6EDE
S1132D93180007A879587000826E680006A8644774
S1132DA30CA86F4708A87947045A002EB46868A88B
S1132DB3684646F85968E86DF07A0000000400A57
S1132DC3E07A0100009CEA5E0040380B876E6800DE
S1082DD31AA873462459
S1132DD86868A859461E6E680020A87946166E686A
S1132DE80027A86E460E7A0000000580AE00C5926
S1132DF85E0043047A0000000440AE06E6900069E
S1132E085E0042307A0000000500AE00CD95E00F0
S1062E1842C05A58
S1132E1B002EB47A0100000200AE16E180003A80B
S1132E2B79586000846E180007A879477C6E680098
S1132E3B06A859470AA86F4706A8754702406A6850
S1132E4B68A8684646F85968E86DF07A00000000F8
S1132E5B400AE07A0100009CEA5E0040380B876E63
S1132E6B68001AA87346246868A859461E6E680042
S1132E7B23A87946166E680027A86E460E7A0000C3
S1132E8B0000580AE00C595E0043047A000000006E
S1132E9B440AE06E6900065E0042307A00000000CF
S1132EAB500AE00CD95E0042C07A0000009D170166
S1132EBB006DF07A00000000440AE07A0100009DE7
S1102ECB0C5E00408A0B975E00615C547042
S1139CEA5361666574792E74787400004D6F746FCE
S1139CFA722E74787400004C696D69742E747874CA
S1139D0A00004F726465722E74787400004E5F5FB0
S1139D1A00005F4E4E000051554954000A4120424B
S1139D2A61736B65742069736E27742031737420B1
S1139D3A466C6F6F72000A41204261736B6574202F
S1139D4A69736E277420326E6420466C6F6F7200DB
S1132ED85E00617E7A0500004AB6790600027A012F
S1132EE800009D5A0D605D507A0100009D890D60B8
S1132EF85D507A0100009DAA0D605D507A010000C3
S1132F089DD60D605D507A0100009E040D605D50F2
S1132F187A0100009E100D605D505E00615C547084
S1052F2840AEB6
S1139D5A0A5550203D202775272C20444F574E2063
S1139D6A3D202764272C204F50454E203D20276F46
S1139D7A272C20434C4F5345203D20276327000AB5
S1139D8A454D455247454E4359203D202773272CBD
S1139D9A205245434F56455259203D2027722700EA
S1139DAA0A31737420466C6F6F722043414C4C2006
S1139DBA3D202779272C20326E6420466C6F6F7200
S1139DCA2043414C4C203D20275927000A31737404
S1139DDA20466C6F6F7220434C4F5345203D20271A
S1139DEA68272C20326E6420466C6F6F7220434CB6
S1139DFA4F5345203D20274827000A5155495420EF
S1139E0A3D20277127000A434F4D4D414E443E00E2
S1132F2A01006DF60F86190069E06FE000026FE099
S1132F3A00046FE0000601006FE600087A00000053
S1132F4A00020AE001006FE0000C7A0000000004AE
S1132F5A0AE001006FE000107A00000000060AE0B0
S1132F6A01006FE0001419006FE000187A000000F6
S1132F7A00180AE001006FE0001A19006FE0001E52
S1132F8A7A000000001E0AE001006FE00020010041
S1132F9A6D76547001006DF60F865E00402E7A003E
S1132FAA000000040AE05E00402E01006D765470B2

S1132FBA5E00617E0F850F9601006F740018010091
S1132FCA6F600014690079200001465201006F60A6
S1132FDA001A6901462801006F66001A7900000188
S1132FEA69E07A00000000040AD0F9735E00430422
S1132FFA7A0100009E1A790000015E004AB6684809
S113300AA859586001E8F87268C86DF07A0100009F
S108301A9E1F0FD05EB4
S113301F0040380B875A0031F8FB6801006F6000DE
S113302F10690079200001466E01006F60001A19C4
S113303F11698101006F6000206901462C01006F47
S113304F6600207900000169E07A00000000040A9D
S113305FD0F96F5E0043047A0100009E2B790000C4
S113306F015E004AB65A0031F86848A85958600102
S113307F787A00000000040AD00CB95E004304F80C
S113308F7268C86DF07A0100009E1F0FD05E00407A
S113309F380B875A0031F801006F60000869014649
S11330AF6E01006F60001A1911698101006F6000D2
S11330BF206901462C01006F660020790000016929
S10F30CFE07A00000000040AD0F94F5E14
S11330DB0043047A0100009E30790000015E004A30
S11330EBB65A0031F86848A859586001007A0000B5
S11330FB0000040AD00CB95E004304F87268C86D73
S113310BF07A0100009E1F0FD05E0040380B875AE8
S113311B0031F801006F60000C6901466C01006F10
S113312B60001A1911698101006F60002069014663
S113313B2C01006F6600207900000169E07A000022
S113314B0000040AD0F96F5E0043047A0100009E6D
S113315B2B790000015E004AB65A0031F86848A883
S113316B59586000887A00000000040AD00CB95E3D
S113317B004304F87268C86DF07A0100009E1F0FBC
S113318BD05E0040380B87406401006F60000C6910
S113319B0079200001465601006F60001A1911696E
S11331AB8101006F6000206901462801006F6600F2
S11331BB207900000169E07A00000000040AD00CBA
S10831CBB95E0043049E
S10531D07A017F
S11331D200009E3C790000015E004AB66848A85987
S11331E24614F87268C86DF07A0100009E1F0FD072
S11331F25E0040380B875E00615C547001006DF61F
S11332020F865E00402E7A00000000040AE05E0092
S1133212402E01006D7654705E00617E0F850F961D
S113322201006F74001801006F60000C69007920BF
S11332320001465201006F60001A6901462801002D
S11332426F66001A7900000169E07A000000000449
S11332520AD0F9735E0043047A0100009E1A7900D2
S113326200015E004AB66848A859586001E8F8723E
S113327268C86DF07A0100009E1F0FD05E004038CF
S11332820B875A003458FB7401006F600008690011
S113329279200001466E01006F60001A19116981DD
S11332A201006F6000206901462C01006F66002057
S11332B27900000169E07A00000000040AD0F96392
S10932C25E0043047A01E3
S11032C800009E47790000015E004AB65ADF
S11332D50034586848A859586001787A00000000FE
S11332E5040AD00CB95E004304F87268C86DF07A1D
S11332F50100009E1F0FD05E0040380B875A003433
S11333055801006F6000106901466E01006F60008F

S11333151A1911698101006F6000206901462C01AA
S1133325006F6600207900000169E07A0000000063
S1133335040AD0F9435E0043047A0100009E4D79E7
S11333450000015E004AB65A0034586848A8595827
S11333556001007A00000000040AD00CB95E004346
S113336504F87268C86DF07A0100009E1F0FD05EE5
S11333750040380B875A00345801006F6000146908
S113338501466C01006F60001A1911698101006F14
S11333956000206901462C01006F6600207900005A
S11333A50169E07A00000000040AD0F9635E004376
S11333B5047A0100009E47790000015E004AB65A6F
S11333C50034586848A859586000887A00000000FE
S11333D5040AD00CB95E004304F87268C86DF07A2C
S11333E50100009E1F0FD05E0040380B874064012B
S11333F5006F600014690079200001465601006FD3
S113340560001A1911698101006F60002069014686
S11334152801006F6600207900000169E07A000049
S11334250000040AD00CB95E0043047A0100009E33
S11334355A790000015E004AB66848A8594614F84F
S11334457268C86DF07A0100009E1F0FD05E0040C0
S1133455380B875E00615C547001006DF60F867A48
S113346500000000040AE001006FE000067A000096
S113347500000E0AE001006FE0001801006D7654AC
S1133485705E00617E0F860F957A000000000E0ABC
S1133495E001006FE000187A000000000A0AE0016D
S11334A5006F6100185E00438A6E680012A87946B2
S10934B50679000001404E
S11334BB0219006FE0001C7A04000044E86F6000FF
S11334CB1C6DF001006F51002001006F50000C5D6B
S11334DB400B876E680013A87946067900000140FC
S11334EB0219006FE0001C6DF001006F5100200109
S11334FB006F5000085D400B876E680014A8794677
S113350B0679000001400219006FE0001C6DF00109
S113351B006F51002001006F5000105D400B876E50
S113352B680015A879460679000001400219006F5F
S113353BE0001C6F66001C6DF601006F510020014B
S113354B006F5000145D400B875E00615C54705E2E
S113355B00617E0F860F957A000000000E0AE001D2
S113356B006FE000187A000000000A0AE001006F08
S113357B6100185E00438A6E680012A879460679CB
S113358B000001400219006FE0001C7A04000044A4
S113359BE86F60001C6DF001006F51002001006F9C
S10835AB50000C5D401F
S11335B00B876E680013A879460679000001400264
S11335C019006FE0001C6DF001006F510020010035
S11335D06F5000085D400B876E680014A87946069B
S11335E079000001400219006FE0001C6DF001003A
S11335F06F51002001006F5000105D400B876E6813
S11336000015A879460679000001400219006FE011
S1133610001C6F66001C6DF601006F510020010055
S11236206F5000145D400B875E00615C54705E59
S113362F00617E0F860F957A000000000E0AE001FD
S113363F006FE000187A000000000A0AE001006F33
S113364F6100185E00438A6E680012A879460679F6
S113365F000001400219006FE0001C7A04000044CF
S113366FE86F60001C6DF001006F51002001006FC7
S113367F50000C5D400B876E680013A879460679DE

S113368F000001400219006FE0001C6DF001006F94
S113369F51002001006F5000085D400B876E6800DA
S11336AF14A879460679000001400219006FE00063
S11336BF1C6DF001006F51002001006F5000105D71
S11336CF400B876E680015A8794606790000014004
S11336DF0219006FE0001C6F66001C6DF601006F8E
S11336EF51002001006F5000145D400B875E006195
S11336FF5C54705E00617E0F860F9501006F600052
S113370F14690146307A00000000200AF00FE15CD3
S106371F00FE386E
S113372201006DF57A000000001C0AF00FE15C0055
S1133732F8860B977A00000000200AF00FE15C0084
S1133742FE165E00615C54705E00617E0F860F950B
S113375201006F60000C690146307A00000000200E
S11337620AF00FE15C00FEC401006DF57A0000006F
S1133772001C0AF00FE15C00FA9E0B977A0000002E
S113378200200AF00FE15C00FEA25E00615C54704F
S1139E1A53544F50005361666574792E74787400F5
S1139E2A004F50454E004F50454E205370656564B0
S1139E3A79004F50454E20537461727400434C4F5E
S1139E4A534500434C4F534520537065656479006D
S10F9E5A434C4F53452053746172740055
S113379201006DF60F86190069E06FE000026FE029
S11337A200046FE0000601006FE600087A000000E3
S11337B200020AE001006FE0000C7A0000000043E
S11337C20AE001006FE000107A00000000060AE040
S11337D201006FE0001419006FE000187A00000086
S11337E200180AE001006FE0001A19006FE0001EE2
S11337F27A000000001E0AE001006FE000200100D1
S11338026D76547001006DF60F865E00402E7A00CD
S1133812000000040AE05E00402E01006D76547041
S11338225E00617E0F850F9601006F740018010020
S11338326F600014690079200001465201006F6035
S1133842001A6901462801006F66001A7900000117
S113385269E07A00000000040AD0F9735E004304B1
S11338627A0100009E66790000015E004AB668484C
S1133872A859586001E8F87268C86DF07A0100002F
S10838829E6B0FD05EF8
S11338870040380B875A003A60FB6A01006F6000FB
S113389710690079200001466E01006F60001A1954
S11338A711698101006F6000206901462C01006FD7
S11338B76600207900000169E07A00000000040A2D
S11338C7D0F9755E0043047A0100009E7779000002
S11338D7015E004AB65A003A606848A85958600121
S11338E7787A00000000040AD00CB95E004304F89C
S11338F77268C86DF07A0100009E6B0FD05E0040BE
S1133907380B875A003A6001006F60000869014667
S11339176E01006F60001A1911698101006F600061
S1133927206901462C01006F6600207900000169B8
S10F3937E07A00000000040AD0F9555E9D
S11339430043047A0100009E7A790000015E004A75
S1133953B65A003A606848A859586001007A0000D3
S11339630000040AD00CB95E004304F87268C86D02
S1133973F07A0100009E6B0FD05E0040380B875A2C
S1133983003A6001006F60000C6901466C01006F2F
S113399360001A1911698101006F600020690146F3
S11339A32C01006F6600207900000169E07A0000B2

S11339B30000040AD0F9755E0043047A0100009EF7
S11339C377790000015E004AB65A003A606848A856
S11339D359586000887A00000000040AD00CB95ECD
S11339E3004304F87268C86DF07A0100009E6B0F00
S11339F3D05E0040380B87406401006F60000C69A0
S1133A030079200001465601006F60001A191169FD
S1133A138101006F6000206901462801006F660081
S1133A23207900000169E07A00000000040AD00C49
S1083A33B95E0043042D
S1053A387A010E
S1133A3A00009E84790000015E004AB66848A859CE
S1133A4A4614F87268C86DF07A0100009E6B0FD0B5
S1133A5A5E0040380B875E00615C547001006DF6AE
S1133A6A0F865E00402E7A00000000040AE05E0022
S1133A7A402E01006D7654705E00617E0F850F96AD
S1133A8A01006F74001801006F60000C690079204F
S1133A9A0001465201006F60001A690146280100BD
S1133AAA6F66001A7900000169E07A0000000004D9
S1133ABA0AD0F9735E0043047A0100009E66790016
S1133ACA00015E004AB66848A859586001E8F872CE
S1133ADA68C86DF07A0100009E6B0FD05E00403813
S1133AEA0B875A003CC0FB6B01006F60000869003A
S1133AFA79200001466E01006F60001A191169816D
S1133B0A01006F6000206901462C01006F660020E6
S1133B1A7900000169E07A00000000040AD0F96420
S1093B2A5E0043047A0172
S1103B3000009E8D790000015E004AB65A28
S1133B3D003CC06848A859586001787A000000001D
S1133B4D040AD00CB95E004304F87268C86DF07AAC
S1133B5D0100009E6B0FD05E0040380B875A003C6E
S1133B6DC001006F6000106901466E01006F6000B7
S1133B7D1A1911698101006F6000206901462C013A
S1133B8D006F6600207900000169E07A00000000F3
S1133B9D040AD0F9445E0043047A0100009E927931
S1133BAD0000015E004AB65A003CC06848A8595847
S1133BBD6001007A00000000040AD00CB95E0043D6
S1133BCD04F87268C86DF07A0100009E6B0FD05E29
S1133BDD0040380B875A003CC001006F6000146928
S1133BED01466C01006F60001A1911698101006FA4
S1133BFD6000206901462C01006F660020790000EA
S1133C0D0169E07A00000000040AD0F9645E004304
S1133C1D047A0100009E8D790000015E004AB65AB8
S1133C2D003CC06848A859586000887A000000001D
S1133C3D040AD00CB95E004304F87268C86DF07ABB
S1133C4D0100009E6B0FD05E0040380B874064016E
S1133C5D006F600014690079200001465601006F62
S1133C6D60001A1911698101006F60002069014616
S1133C7D2801006F6600207900000169E07A0000D9
S1133C8D0000040AD00CB95E0043047A0100009EC3
S1133C9D9E790000015E004AB66848A8594614F89B
S1133CAD7268C86DF07A0100009E6B0FD05E004004
S1133CBD380B875E00615C547001006DF60F867AD8
S1133CCD00000000040AE001006FE0000E01006D2A
S1133CDD7654705E00617E0F860F957A00000000AA
S1133CED040AE001006FE0000E01006F61000E0F8A
S1133CFDE05E00438A6E680004A8794606790000E9
S1133D0D01400219006FE000127A04000044E86FCD

S1083D1D6000126DF0CF
S1133D2201006F51002001006F50000C5D400B87B2
S1133D326E680005A8794606790000014002190061
S1133D426FE000126DF001006F51002001006F500F
S1133D5200085D400B876E680006A8794606790065
S1133D620001400219006FE000126DF001006F5173
S1133D72002001006F5000105D400B876E68000742
S1133D82A879460679000001400219006FE000128B
S1133D926F6600126DF601006F51002001006F5033
S1133DA200145D400B875E00615C54705E00617EAF
S1133DB20F860F957A00000000040AE001006FE00D
S1133DC2000E01006F61000E0FE05E00438A6E6811
S1133DD20004A879460679000001400219006FE049
S1133DE200127A04000044E86F6000126DF00100D3
S1133DF26F51002001006F50000C5D400B876E680D
S1133E020005A879460679000001400219006FE017
S1073E1200126DF03A
S1133E1601006F51002001006F5000085D400B87C1
S1133E266E680006A879460679000001400219006B
S1133E366FE000126DF001006F51002001006F501A
S1133E4600105D400B876E680007A8794606790067
S1133E560001400219006FE000126F6600126DF652
S1133E6601006F51002001006F5000145D400B8765
S10A3E765E00615C54705E05
S1133E7D00617E0F860F957A00000000040AE001B1
S1133E8D006FE0000E01006F61000E0FE05E004356
S1133E9D8A6E680004A8794606790000014002196C
S1133EAD006FE000127A04000044E86F6000126DA9
S1133EBDF001006F51002001006F50000C5D400BAD
S1133ECD876E680005A8794606790000014002193E
S1133EDD006FE000126DF001006F51002001006FC3
S1133EED5000085D400B876E680006A87946067979
S1133EFD000001400219006FE000126DF001006F28
S1133F0D51002001006F5000105D400B876E68005B
S1133F1D07A879460679000001400219006FE000F9
S1133F2D126F6600126DF601006F51002001006FD4
S1133F3D5000145D400B875E00615C54705E006140
S1133F4D7E0F860F9501006F600014690146307A6C
S1133F5D00000000200AF00FE15C00FE4401006D3B
S1063F6DF57A00DF
S1133F700000001C0AF00FE15C00F8A60B977A0022
S1133F80000000200AF00FE15C00FE225E00615C8D
S1133F9054705E00617E0F860F9501006F60000C08
S1133FA0690146307A00000000200AF00FE15C004E
S1133FB0FECA01006DF57A000000001C0AF00FE153
S1133FC05C00FABE0B977A00000000200AF00FE1B4
S10D3FD05C00FEA85E00615C547003
S1139E6653544F50005361666574792E74787400A9
S1139E760055500055502053706565647900555060
S1139E8620537461727400444F574E00444F574E2B
S1139E962053706565647900444F574E20537461AF
S1069EA6727400D0
S1133FDA01006DF60F867A0000FFE23E010069E0F8
S1133FEA7A0600FFE23EF87268E87A0000000006EB
S1133FFA0AE001006FE000027A0100009EAA0100B4
S113400A6F6000025E005AA8F8736EE8000FF84E5C
S113401A6EE800106EE80011F84E6EE80012010017

S113402A6D7654700F811A800100699054705E0096
S113403A617E0F946E7D00197A0600FFE23E68499D
S113404AA9434716A94D470CA9504714A953462813
S113405A68ED40246EED000F401E6EED001040180F
S113406A6E480006A8434706A8544708400A6EED5F
S113407A001140046EED001219005E00615C547079
S113408A01006DF60F966869A94C46247A0600FF6B
S113409AE23E7A00000000060AE001006FE0000237
S11340AA01006F71000801006F6000025E005AA8E8
S11340BA190001006D7654705E00617E0F94010051
S10940CA6F7500187A0671
S11340D000FFE23E6849A9434716A94D470CA95082
S11340E04716A953462E686E400A6E6E000F4004B1
S11340F06E6E001068DE401C6E480006A84347063B
S1134100A854470A400E6E6E001168DE40066E6EBC
S1134110001268DE19005E00615C547001006DF6E8
S11341200F966869A94C46247A0600FFE23E7A009E
S1134130000000060AE001006FE0000201006F6169
S1134140000201006F7000085E005AA81900010008
S11341506D7654705E00617E0F850F9601006DF6DB
S11341607A0100009EB40FD05C00FF560B970C0041
S11341704624A8014706A8024716401A7A01000000
S11341809EC7790000015E004AB679060001400827
S113419079060002400219660D605E00615C54708E
S10441A05EBD
S11341A100617E0F850C9E6DF67A0100009EB40FAF
S11341B1D05C00FE820B870C00461CA8004714A8A4
S11341C10146147A0100009ED6790000015E004A7F
S11341D1B6400419664004790600010D605E006172
S11341E15C54705E00617E0F850F9601006DF67A57
S11341F10100009EE50FD05C00FEC60B970C004644
S113420124A8014706A8024716401A7A0100009E16
S1134211C7790000015E004AB679060001400879BA
S1134221060002400219660D605E00615C54705E17
S113423100617E0F850C9E6DF67A0100009EE50FED
S1134241D05C00FDF20B870C00461CA8004714A8A4
S11342510146147A0100009ED6790000015E004AEE
S1134261B6400419664004790600010D605E0061E1
S11342715C54705E00617E0F850F9601006DF67AC6
S10442810138
S113428200009EF20FD05C00FE360B970C00462412
S1134292A8014706A8024716401A7A0100009EC7E2
S11342A2790000015E004AB67906000140087906EA
S11342B20002400219660D605E00615C54705E008C
S11342C2617E0F850C9E6DF67A0100009EF20FD07F
S11342D25C00FD620B870C00461CA8004714A80172
S11342E246147A0100009ED6790000015E004AB6A8
S11342F2400419664004790600010D605E00615CAA
S113430254705E00617E0F850C9E6DF67A0100008B
S11343129F060FD05C00FD1E0B870C004616A800FB
S11343224712A801460E7A0100009ED679000001C9
S11343325E004AB65E00615C54705E00617E1B875C
S11343420F8518886EF800017A06000000010AF64C
S113435201006DF67A0100009F060FD05C00FD603C
S11343620B970C004618A800470EA8014710A80295
S1094372470C400A40085D
S110437840066E780001400218880B875E36

S113438500615C54705E00617E0F850F96790000B5
S1134395096DF001006DF67A0100009F110FD05CE5
S11343A500FD740B970B870C004618A8014706A858
S11343B5024710400E7A0100009EC7790000015E96
S10C43C5004AB65E00615C54700D
S1139EAA79796E6E79796E6E00005065726D697498
S1139EBA436F6D6D616E642E74787400000A526587
S1139ECA6164696E67204572726F72000A5772691C
S1139EDA74696E67204572726F7200436F6D6D61AC
S1139EEA6E642E74787400005065726D69745475CB
S1139EFA726E4F70656E2E74787400004D6F746FB6
S1139F0A722E74787400004C696D69742E747874B7
S1059F1A000042
S11343CE5E00617E0F860F957A0000009F1C0FE141
S11343DE5E00610801006FE600080FE055267A00C3
S11343EE0000000C0AE001006FE0000E19006FE000
S11343FE000C19006FE0001201006FE500145E005F
S113440E615C54705E00617E0F8601006FE60008EA
S113441E0FE201006DF25E0045260B975E00615CB4
S113442E54705E00617E7A37000000100F86010023
S113443E6FE600087A02000000080AF201006DF22E
S113444E5E0045260B9701006F6600080FA10FE271
S113445E0FF05E005BAA5E00601C7A17000000106E
S113446E5E00615C54705E00617E0F850D16790EE1
S113447E03E852E617F60FE101006F5000145E00D9
S113448E45CA5E00615C547001006DF60F867A00BA
S113449E0000000C0AE001006FE0000E792100011C
S11344AE460A790000016FE0000C400A0D11460622
S10944BE19006FE0000C81
S11344C401006D76547001006DF60F867A000000CA
S11344D4000C0AE001006FE0000E6F60000C0100A5
S11344E46D7654705E00617E0F860F9569606F71FF
S11344F400181D10470A190069D06F70001869E08D
S11345045E00615C54705E00617E0F850D1617F6C4
S11345140FE101006F5000145E0045CA5E00615C48
S10545245470CE
S10B9F1C000000000000000003A
S11345267A0000FFE25201006F7100045E00610829
S113453654705E00617E7A37000000200F867A028F
S1134546000000180AF201006DF255D40B970FE133
S11345560FF05E0060C20F817A0200009F347A007A
S1134566000000100AF05E005E847A000000001866
S11345760AF07A01000000080AF15E0061084010A3
S11345867A02000000080AF201006DF255920B97B9
S11345967A01000000180AF17A02000000100AF2FC
S11345A60FF05E005BDA0F817A00000000080AF064
S11345B65E0060B20D0046C87A17000000205E0058
S11345C6615C54705E00617E1B971B970F860F9587
S11345D67A02000000020AE201006DF25C00FF406D
S11345E60B970FD10FF05E0060C20F817A020000B5
S11345F69F347A000000000A0AE05E005E840B978F
S11346060B975E00615C54705E00617E0F860F95AA
S10746160FE055B0A9
S113461A01006F6000120B7001006FE000125E0070
S113462A615C54705E00617E0D057A0400FFE25AF4
S113463A400601006F44001A01006F40001A01008E
S111464A6F01001A46EC79250001460A7A0634

S113465800FFE2965A00474C79250002460A7A067B
S113466800FFE2B45A00474C7925000B460A7A0644
S113467800FFE2D25A00474C7925000C460A7A0615
S113468800FFE2F05A00474C7925000D460A7A06E6
S113469800FFE30E5A00474C7925000E460A7A06B6
S11346A800FFE32C5A00474C79250013460A7A0683
S10846B800FFE34A5A74
S11346BD00474C79250014460A7A0600FFE3685A31
S11346CD00474C7925001546087A0600FFE386401E
S11346DD6E7925001646087A0600FFE3A44060793B
S11346ED25001746087A0600FFE3C24052792500DC
S11346FD1E46087A0600FFE3E040447925001F4675
S113470D087A0600FFE3FE40367925002946087A2C
S113471D0600FFE41C40287925002A46087A060086
S113472DFFE43A401A7925002B46087A0600FFE488
S113473D58400C7925002C46067A0600FFE4760FC7
S113474DE6461C7A0100009F2419005E004AB67AE2
S104475D0157
S113475E00009F2619005E004AB61A8040540100DD
S113476E6FE4001601006F40001A01006FE0001A9B
S113477E01006F86001601006FC6001A7A00000052
S113478E9F3C7A01000000020AE15E0061087A0094
S113479E00009F447A010000000A0AE15E006108EE
S11347AE69E51A8001006FE000120FE05E000E94BF
S11347BE0FE05E00615C547001006DF60F86472AB0
S11347CE0FE05E00114801006F60001601006F617B
S11347DE001A01006F81001A01006F60001A0100B8
S11347EE6F61001601006F81001601006D76547023
S11347FE5E00617E0F867A0200FFE25201006DF2C7
S113480E5C00FD140B977A0000FFE2527A01000060
S113481E00020AE15E0061085E00615C54700100F3
S113482E6DF60F867A0000009F4C7A01000000029D
S113483E0AE15E00610801006D76547001006B2081
S107484E00FFE2740E
S113485201006F01001A4604190054707900000127
S1094862547001006B21FC
S113486800FFE27401006F11001A1988400C01005F
S11348786F11001A0D800B500D080F9146F00D8033
S113488854706DF50D0501006B2000FFE274010003
S11348986F01001A472001006B2100FFE2746910C1
S11348A81D5046040F90401001006F11001A0100BB
S11348B86F10001A46E81A806D7554705E00617EA9
S11348C80D0555BE0F86460E0D505C00FD580F862C
S11348D85C00FF22401C7A00000000020AE07A0113
S11348E800009F4C5E0060960D0047060FE05C00D9
S11348F8FF040FE05E00615C54705E00617E0D058D
S113490855800F86460E0D505C00FD1A0F865C001D
S1134918FEE440247A00000000020AE07A01000065
S11349289F4C5E0060960D0047080FE05C00FEC6D2
S113493840060FE05C00FE865E00615C54705E001A
S1134948617E1B971B977A0500FFE25A01006F5699
S1074958001A40728C
S113495C01006F65001A7A00000000020AE07A0178
S113496C00009F3C5E00614E0D0047547A0000002E
S109497C00020AE07A01CB
S113498200009F4C5E00614E0D00473E7A0100001D
S113499200020AE17A020000000A0AE20FF05E0056

S11349A25BDA0F817A0000FFE2525E0060A20D0023
S11349B247187A0000FFE2527A01000000020AE17E
S11349C25E0061080FE05E00052C0FD601006F60E8
S11349D2001A46860B970B975E00615C54701A802F
S11349E201006BA000FFE2707A0000FFE278010091
S11349F26BA000FFE2747A0000FFE25A01006BA091
S10D4A0200FFE28E1A8001006BA092
S1134A0C00FFE2925470F8013860188838613862FC
S1134A1C188838633890F8FF389118883864F88808
S1134A2C3865F804386679009E576B80FF68190067
S1134A3C6B80FF6A19006B80FF6C547001006DF280
S1134A4C7F67722079009E576B80FF687A0100FFA5
S1134A5CE2527A0200009F540F905E005BDA010071
S1134A6C6D7254705A0049467A0000009F447A01D3
S1134A7C00FFE2525E006108558C5E0002A85A00EA
S1054A8C49E0FC
S1139F240A0063616C6C6F63206661696C6564002D
S1139F34408F400000000000BFF0000000000005C
S1139F440000000000000000C00000000000004A
S10B9F543F50624DD2F1A9FC5C
S1134A8E7A0000009F5C01006DF07A0000FFE49451
S1134A9E5E005A200B977A0000FFE49401006DF03C
S1134AAE5E004E380B9754705E00617E7A0500FFF0
S1134ABEE4D40D040F960C4458600114AC004768FF
S1134ACEAC014710AC0258700106AC03587000C21B
S1134ADE5A004BDE55AA7A0100009F5E0FE05E007E
S1134AEE5A7E0D00461E7A0000009F6401006DF091
S1134AFE7A0000009F6101006DF00FE05E005A2006
S1134B0E0B970B9701006DF67A0000009F61010071
S1134B1E6DF00FD05E005A200B970B9701006DF5C9
S10D4B2E7A0000009F6140507A01F5
S1134B3800009F5E0FE05E005A7E0D00461E7A005D
S1134B4800009F6401006DF07A0000009F6101007E
S1134B586DF00FE05E005A200B970B9701006DF67E
S1134B687A0000009F6101006DF00FD05E005A20AB
S1134B780B970B9701006DF57A0000009F61010008
S1134B886DF05E004C340B970B9701006DF65E00D9
S1134B984E380B974040403E01006DF67A00000006
S1134BA89F6101006DF00FD05E005A200B970B97A1
S1134BB801006DF57A0000009F6101006DF05E0051
S1134BC84C340B970B970FD05E005AC40B500D0152
S10A4BD80FD05E00589C5E44
S1084BDF00615C54704D
S10E9F5C0C000A00002573000A0C0033
S1134BE4188838BA38B8F85038B919000B507920F6
S1134BF401184DF8F83038BA28BCF88038BC547022
S1134C040C8820BC737047FA38BBE07F30BC547007
S1134C1420BC736047FA29BDE0BF30BC0C985470C4
S1134C247EBC736047067900000154701900547008
S1134C345E00617E7A0600FFE5147A050000000435
S1134C447A00000000180AF00AD07308470E7A00AD
S1134C54000000180AF00AD00B70400A7A00000022
S1134C6400180AF00AD00F85189968E901006DF05D
S1134C7401006F71001C0FE05E005AE00B97195599
S1134C840D5017F00AE0680947220D5017F00AE0A7
S1134C946808A80A4606F80D5C00FF640D5017F077
S1134CA40AE068085C00FF580B5540D45E00615C61

S1054CB4547037
S1134CB61911400C19880B58792806824DF80B51A7
S1134CC61D014DF0547001006DF50D090D8D28D6AB
S1134CD617500D010D99470C0D1979690060794933
S1134CE6001040060D19796900600DD50D90148DDD
S1134CF60CD8C88038D63DD639D67900000455B0CD
S1134D0601006D7554706DF56DF40D090D8528D68A
S1134D1617500D010D99470C0D19796900607949F2
S1134D26001040060D19796900600D541194119411
S1134D3611941194EC0F0D90148CED0F148D0CC877
S1134D46C88038D63CD60CD8C88038D63DD639D696
S1134D56790000045C00FF586D746D755470010092
S1134D666DF619EE7900000F5C00FF4419667908A9
S1134D7600030DE05C00FF4E0B56792600034DEE53
S1134D86790800020DE05C00FF3C196679080028EB
S1134D960D605C00FF70790800100D605C00FF6613
S1094DA67908000E0D6008
S1134DAC5C00FF5C790800060D605C00FF5279081B
S1134DBC00010D605C00FF48790800020D605C0087
S1134DCCFF3E01006D7654707908000119005C00F8
S1134DDCFF2E7908000219005A004D0C6DF60C8E4B
S1134DECAE0C460455E2401CAE0A460455DA401498
S1134DFCAE0D460455D2400C17D60D687900000150
S1134E0C5C00FEFC6D76547001006DF60D0E0D8684
S1134E1C0D6079080040528009E0791000800D087C
S1134E2C19005C00FEDA01006D7654705E00617E41
S1134E3C790C000119447A0600FFE5647A05000039
S1134E4C00047A00000000180AF00AD07308470E19
S1134E5C7A00000000180AF00AD00B70400A7A009E
S1134E6C000000180AF00AD00F85189968E90100B0
S1134E7C6DF001006F71001C0FE05E005AE00B97A0
S1134E8C19550D5017F00AE0680947560D5017F0E5
S1094E9C0AE06808A80A01
S1134EA2460A0DC80D405C00FF68403C0D5017F0E8
S1134EB20AE06808A80D460C790800020D405C0060
S1134EC2FE4840240D5017F00AE06808A80C460675
S1134ED25C00FEFE40120D5017F00AE0680817D07E
S1134EE20D080DC05C00FE220B5540A05E00615C04
S1054EF25470F7
S1134EF41911400C19880B58792806824DF80B5167
S1134F041D014DF054705C0009145C0000B4550499
S1134F145A004FEE01006DF618886AA800FFE66791
S1134F246AA800FFE66918886AA800FFE6686AA809
S1134F3400FFE5A579080080790000045C000808F7
S1134F4479080010790000205C0007FC7906000F43
S1134F54790800100D605C0007EE0D687900000B02
S1134F645C0007E40D687900000D5C0007DA79083A
S1134F740050790000095C0007CE790800D6790057
S1134F8400075C0007C219660D605C000710790E08
S1134F9400010DE05C0006E60DE05C0007000D680F
S1134FA47900002B5C0007A00D687900002F5C00DA
S1134FB407960DE8790000275C00078C01006D76E5
S1134FC454707908000519005C00077C79000064BB
S1134FD45C00FF1C790800C419005C00076A7908A7
S1094FE400037900000147
S1134FEA5A00574C79080002790000055C00075201
S1134FFA790800CC19005A00574C5E00617E7A0486

S113500A0000A01A7A0000009FE901006DF05E001B
S113501A4C340B9719EE19660DE5790D001052D52C
S113502A0D6009505C00070C17506DF001006DF418
S113503A5E004C340B970B870B56792600104DE014
S113504A7A000000A02001006DF05E004C340B973B
S113505A0B5E792E00044DBE5E00615C5470010044
S107506A6DF67A065C
S113506E00FFE5A4790000065C0006C468E87C60D6
S113507E734047367900000E5C0006B40C8E7348FD
S113508E47045C0002C8735E47085C0002C25A0004
S113509E515C730E47085C0002B85A00515C731ED4
S11350AE47045C0002AE5A00515C7C607360472279
S11350BE7900000C5C0006780C8E730847085C00C0
S11350CE00925A00515C731E587000825C0001C638
S11350DE407C7C607320471E7900000A5C000650FA
S11350EE0C8E730847065C00020A4062731E475E0D
S11350FE5C00023E40587C607310475279000008F2
S113510E5C00062C0C8E7368470A5C00FDFC5C0089
S113511EFECE403A734E471A790800C07900000953
S113512E5C00061A79080003790000055C00060E80
S113513E401C737E471879080050790000095C0003
S113514E05FC79080002790000055C0005F00100FA
S108515E6D7654705E44
S113516300617E19DD790000265C0005CE0C8E7389
S113517368587001047A0600FFE5A7790000086DFB
S1135183F00FE179080026790000255C0005CC0BBC
S1135193870DD05C0004E80DD05C0005027905009F
S11351A3200D505C0004C06868E8605860009C6E82
S11351B3680001473CA801471EA8034772A8054797
S11351C33EA8064726A8084716A809475CA80A4726
S11351D326A80B4760406C5C0001865A00526A6A3A
S11351E32800FFE5A617500D08400E5C000224407B
S11351F3765C00035440700DD87900002140247974
S11352030800400D505C0005406E6E0002CE806ABC
S1135213AE00FFE5A56A2800FFE5A517500D087941
S11352230000045C000522403E5C00038E40385CB2
S113523300018840326E680002472C0D505C000465
S11352430E40240D505C000406401C6868E860A807
S10852532046080D5088
S11352585C0003F6400C686EEE60AE400D505C00D7
S113526803E87A0000FFE6687D0070000DD05C005B
S1135278045440226A2800FFE669470818886AA888
S113528800FFE6690DD05C0004107908000179007D
S10A529800275C0004AE5E79
S113529F00615C54705E00617E7900002E5C000437
S11352AF8E0C8EE8C017504626790000406DF07AB9
S11352BF0500FFE5E70FD17908002E7900002D5C7B
S11352CF00048C0B870D060D010FD05C000562796E
S11352DF0000015C00039C790800017900002F5C3A
S11352EF00045A5E00615C5470790000365C000460
S11352FF3E54706DF6790000225C0004320C8E73FD
S113530F68472A7358472619005C0003866A2800EA
S113531FFFE669470C5C0001C819005C0003A0405D
S113532F0C79080001790000275C0004106D765496
S113533F706DF67900002A5C0003F40C8E736847D6
S113534F08735847045C0004826D7654705470548C
S113535F70547054705E00617E7A0600FFE5A76893

S113536F68E803A80246426E680003E80747186E11
S113537F690003E9071751177178106A2800009F16
S108538F6817505C00EB
S113539402D46E680003E8071750790100011A0864
S11353A44B04101140F817117A0000FFE667680AEE
S11353B4169A688A5E00615C54705E00617E7A06A8
S11353C400FFE5A76868E803A80246406E68000387
S11353D4E80747186E690003E90717511771781036
S11353E46A2800009F6817505C0002626E6800031D
S11353F4E8071750790100011A084B04101140F80B
S11354047A0000FFE667680A149A688A5E00615CA2
S111541454707A0000FFE6687D0072006A287B
S113542200FFE5AAA801470AA8024716A8034722D4
S113543254706A2900009F7017517A0000009F7010
S113544240686A2900009F8417517A0000009F82F6
S113545240586A2800FFE5A9470EA801471AA80287
S11354624726A803473254706A2900009FA917D11F
S11354727A0000009FA940326A2900009FAD17D12C
S11354827A0000009FAD40226A2900009FBBF17D116
S10D54927A0000009FBBF40126A2950
S113549C00009FC717D17A0000009FC755025470B4
S11354AC5E00617E0F840D187A0600FFE66A6A2897
S11354BC00FFE5AE17500C8018886A2900FFE5AD94
S11354CC1751091069E01D804F0269E801006BA4B4
S11354DC00FFE66CF8016AA800FFE66955065E005A
S11354EC615C54705E00617E7A0600FFE66A7905A2
S11354FC0008696047446960792000084E026965B9
S113550C7A0400FFE66C6DF501006941790800220D
S113551C790000215C0002780B870D056961190184
S113552C69E117F0010069410A81010069C16960F1
S113553C460818886AA800FFE6695E00615C54702F
S104554C5EFD
S113554D00617E6A2800FFE5A7E80317500D0519D2
S113555DEE790600210D0047067925000146100F4F
S113556DE05C0001DA0DE80D605C0001D2403C798E
S113557D250002462E6A2800FFE5AAE80717500DFD
S113558D05790100011A084B04101140F86A28002F
S113559DFFE6671750660147067908000140060DBF
S11355ADE840020DE80D605C0001945E00615C54FF
S11355BD706DF66A2800FFE5A96AA800FFE5A64607
S11355CD3C19660D68790000285C0001720D68793D
S11355DD00002C5C0001680D68790000305C00014F
S11355ED5E0D68790000345C0001540D687900008C
S11355FD385C00014A0D687900003C404018886A08
S113560DA800FFE667790600010D605C000086794E
S113561D080011790000285C000124790800037942
S113562D00002B5C0001180D60554A7908001279B2
S108563D00002C5C00DD
S113564201080D687900002F5C0000FE6D7654702E
S11356526DF60D065C0000E4C88017500D080D605E
S11356625C0000E66D7654706DF60D065C0000CCAE
S1135672E87F17500D080D605C0000CE6D76547004
S113568201006DF60D0617F61036790800087860EA
S11356926B2000FFE0085C0000B001006D765470DF
S11356A25E00617E0D0617F610367A0500FFE000F4
S11356B20AE569505C00008417500D06703E0D68C0
S11356C269505C0000845E00615C54705E00617E20

S11356D21B971B977A0500FFE6680D0EFE01790002
S11356E200010DE11A094B04101040F868591751D3
S11356F266104704702E4002722E701E17560DE07C
S113570217F0103001006FF000040D6801006F7193
S1135712000478106B2000FFE000010069F1552AB4
S1135722790000011B5E4B04101040F8685915897B
S109573268D90B970B97E9
S10457385E0F
S113573900615C54706AA8004000036A28004000B5
S11357490154700D016AA9004000030D806AA80085
S113575940000154705E00617E7A03004000030F2C
S1135769840F9568BC19EE196640180DC055C6E833
S11357790F471868BC6A280040000168D80B750BED
S11357895E0B566F7000181D064DE00DE05E00615B
S11357995C54705E00617E0D0C0D830F956F740070
S11357A91819EE1966401E0D30558AE81F471A0D60
S11357B9C06AA80040000368586AA8004000010BAA
S11357C9750B5E0B561D464DDE0DE05E00615C54A4
S11357D97001006DF619EE7A0000FFE627790100E2
S11357E9405C0001140D06790000015C00FEAA0D5E
S11357F96647166DF67A0100FFE6277908002A79CC
S1135809000029558E0B870D0E790000015C00FEFF
S1135819B40DE001006D76547019006BA000FFE62A
S1065829726BA0FC
S113582C00FFE67019006BA000FFE6766BA000FF8B
S113583CE67454705E00617E7A0400FFE6727A05AA
S113584C00FFE6700F830D1E7FF5725019994030DF
S113585C6940792001004C2C695017F068397800A5
S113586C6AA900FFE67869500B5017F07901010023
S113587C01D0531069D869400B5069C00B730B5995
S10E588C1DE94DCC7FF570500D905EC0
S113589700615C54705E00617E790E01007A04003A
S11358A7FFE6767A0500FFE6740F830D127FF57224
S11358B750199940346940792001004C32695017D7
S11358C7F001D053E00D8017F0683978006AA9001A
S11358D7FFE77869500B5017F001D053E069D86997
S11358E7400B5069C00B730B590D201D094DC67F23
S11358F7F570500D905E00615C54705E00617E7AB6
S11359070500FFE6760F840D1E7FF572501966407A
S1135917381DE64C386B2000FFE67469511910797E
S113592710010017F07901010001D053100D801702
S1135937F00D6117F10AC178006A2800FFE778685C
S11359479869501B5069D00B5669504EC47FF57048
S1135957500D605E00615C54705E00617E7A0500E5
S1135967FFE6720F840D1E7FF57250196640381DCE
S1135977E64C386B2000FFE6706951191079100166
S10859870017F0790197
S113598C010001D053100D8017F00D6117F10AC1FE
S113599C78006A2800FFE678689869501B5069D034
S11059AC0B5669504EC47FF570500D605EC0
S11359B900615C54706B2000FFE67217F0790101F6
S11359C90001D053100D8054706B2000FFE6761749
S11059D9F07901010001D053100D805470CE
S1139F6820282C3034383C2012010001000000085E
S1139F78FEFF10000100010102010902270001018F
S1139F8800C0640904000003000000030705810200
S1139F984000FF070502024000FF07058302400057

S1139FA8FF04030904120355005300420020005420
S1139FB800450053005400080331002E00300022EE
S1139FC80355005300420020005400450053005439
S1139FD8002000500052004F004700520041004D3E
S1049FE80075
S11399C00023002B0033003B0027002F0037003F0C
S1139FE930302030312030322030332030342030AB
S1139FF93520303620303720303820303920304171
S113A0092030422030432030442030452030460A56
S10CA019002530325820000A0032
S11359E601006DF67A0600FFE8B4010069607A01EA
S11359F641C64E6D5E0080367A10000030390100D4
S1135A0669E06960198817707A01000080005E00FA
S10D5A1680100D1001006D7654702E
S1135A205E00617E0F857A06000000047A000000A4
S1135A3000180AF00AE07308470E7A000000001805
S1135A400AF00AE00B70400A7A00000000180AF01E
S1135A500AE00F8601006DF001006F70001C010069
S1135A606DF001006DF51A91790000015E0061A6E9
S1115A707A170000000C0D065E00615C547096
S1135A7E5E00617E0F860F9540040B760B75686989
S1135A8E68581C8946040C9946F068681750685D7F
S10D5A9E175519505E00615C547047
S1135AA85E00617E0F840FC60F950FE00B766C596D
S10F5AB8688946F60FC05E00615C547004
S1135AC45E00617E0F851AE640020B760FD00B75DC
S10F5AD4680946F60FE05E00615C547048
S1135AE05E00617E0F850F9601006F710018010043
S1135AF06DF101006DF601006DF51A917900000159
S1135B005E0061A67A170000000C5E00615C5470B1
S1135B100FC446120FD5460E792107FF46120FB365
S1135B204604156E4A0A1AC47A050000000818666E
S1135B305A005DA40D9946120FC4461A0FD5461696
S1135B400FB3461E16E6580002660FA246360FB381
S1135B5046325800025A0FA246140FB3461058009B
S1135B60024E0CE60D190FA40FB55A005DB01035A7
S1135B701234792C00104C0C1B5910351234792C2B
S1135B8000104DF410331232792A00104C0C1B51C3
S1135B9010331232792A00104DF4580000CA1AC487
S1135BA01AD5186619995800020601006DF2010012
S1135BB06DF301006F23000401006DF301006923FD
S1135BC0795B800001006DF30FF2550E0B970B9775
S1135BD001006D7301006D7254706DF601006DF577
S1135BE001006DF401006DF301006DF201006DF130
S1135BF001006DF0010069140100692510341035AE
S1095C001FD442184510F9
S1135C0601006F14000401006F2500041FD444062D
S1135C160F940FA10FC26816682E0FA00100691416
S1135C2601006F1500040100690201006F030004FF
S1135C36796C000F796A000F691911191119111975
S1135C461119796907FF69011111111111111147
S1135C56796107FF792907FF5870FEAE0D11587059
S1135C66FECC10351234103512341035123410337D
S1135C7612321033123210331232794C0080794AC1
S1135C8600800D901910474C792000364E3E79203E
S1135C9600204D0E793000200FB34702700A0FA380
S1135CA61AA2792000104D14793000100D380DB367

S1135CB60D2B0DA219AA0D884702700B0C884F14E1
S1135CC6113213334402700B1B5040F01AA27A03AD
S1135CD600000010C6815E84B2A0AB544020B7450
S1135CE60AA4792C0100585000801134133544025C
S1075CF6700D0B59C6
S1135CFA792907FF5860006E1AC41AD5580000A6FE
S1135D0A1FA446061FB55870FE8A1AB544087A348A
S1135D1A0000000145061AA4450440121AA41735C7
S1135D2A17347A150000000144020B740CE60FC401
S1135D3A46080FD41AD57919FFE00DCC460C0D4C41
S1135D4A0DD40D5D19557919FFF0792C0100450819
S1135D5A113413350B5940F2792C00804C08103555
S1135D6A12341B5940F20D994E10113413350D9903
S1135D7A4A08113413350B5940F4732D471C0CD8B8
S1135D8AE80B47167A150000000844020B74792CB5
S1135D9A01004D06113413350B59113413351134DF
S1135DAA133511341335796C000F101910191019A2
S1135DBA1019649C101C1006131C01006D7001005D
S1135DCA698401006F85000401006D7101006D7221
S1135DDA01006D7301006D7401006D756D76547069
S1135DEA01F0640158600080792407FF5860006C51
S1135DFA5800007401F064235860006C40520F8508
S1135E0A01F06415460E0D44464601F064234640EC
S1135E1A5800005410311230792800104C0C1B5CC6
S1135E2A10311230792800104DF4580000BA0FA52A
S1135E3A01F06435472610331232792A00104C0CCC
S1135E4A1B5410331232792A00104DF45800009E65
S1135E5A1AC4100E131C1AD55800018E790E07FFA7
S1135E6A1AC41AD5580001621AC418EE790E07FF2C
S1135E7A19DD790500085800015001006DF601008B
S1135E8A6DF501006DF401006DF301006DF201007F
S1135E9A6DF101006DF0681E6826156E691C111CF0
S1135EAA111C111C111C796C07FF6924111411149C
S1135EBA11141114796407FF01006D1001006911AF
S1135ECA7968000F01006F23000401006922796ACF
S1095EDA000F792C07FF05
S1135EE05870FF06792407FF5870FF120DCC5870C5
S1135EF0FF160D445870FF40194C791C03FF0DCE5B
S1135F00792EFFCC58D0FF5279480010794A0010FF
S1135F101AC47A0500000100103312321031123016
S1135F20441C0AB144087A100000000145040AA089
S1135F3040040AA004011235123444E0401C1AB193
S1135F4044087A300000000145041AA040041AA056
S1135F5004011235DD01123444C2730D46080AB13F
S1135F6044020B700AA001F064014702700D792C02
S1135F7000804C06103512341B5E792E07FF58C083
S1135F80FEE40DEE4E2E113413354402700D0DEE6A
S1135F904A040B5E40F0732D47180CD8E80B4712E8
S1135FA07A1500000008440A0B74792C00804D0216
S1135FB00B5E4020732D471C0CD8E80B47167A154F
S1135FC00000000844020B74792C01004D061134C3
S1095FD013350B5E1134D2
S1135FD613351134133511341335796C000F101E34
S1135FE6101E101E101E64EC101C100E131C010054
S1135FF66D700100698401006F85000401006D71F5
S113600601006D7201006D7301006D7401006D7501
S109601601006D765470D9

S113601C01006DF201006DF101006F01000401003C
S113602C69000D821112111211121112796207FFFC
S113603C0D8A7968000F793203FF4B4A792200539A
S113604C4E44794800107912FFEC4B227922002040
S113605C4C0C0D224F1E103112301B5240F40F907A
S113606C7912FFE00D224F0C10301B5240F6113009
S113607C0B524BFA796A8000470217B001006D711D
S10D608C01006D7254701A8040F297
S10F60965E007F54A801470219005470FB
S11360A25E007F540C884E0419005470F80154703A
S11360B25E007F540C884604F80154701900547032
S11360C201006DF201006DF11AA20F9147224A06F7
S11360D27902080017B17912041F1B52103144FAD6
S11360E21031121210311212103112121031121217
S11360F2698201006F8100026F8A000601006D71DF
S109610201006D725470F0
S113610801006DF2010069020100699201006F024A
S1116118000401006F92000401006D725470C8
S11361266DF301006DF201006DF101006DF06C0B72
S1136136689B0B011B7246F601006D7001006D71C1
S10B614601006D726D735470CA
S111614E5E007F541A08470479000001547064
S113615C01006F76001401006D7201006FF20010E4
S113616C01006D7201006D7301006D7401006D759A
S105617C54705A
S113617E01006DF501006DF401006DF301006DF288
S113618E01006F72001001006DF201006FF6001432
S10B619E01006F72000454704C
S11361A65E00617E7A37000000B27A0300FFE88C56
S11361B67A0400FFE87C0D0201006FF100AA0100DA
S11361C66F7600CE01006F7500D20D00466C01009C
S11361D66F7000CA460E790004526BA000FFE8B840
S11361E65A00627801006F7000CA6E080010E80753
S11361F617D0460C790005146BA000FFE8B840726F
S113620601006F7000CA6E080010732846107308E9
S1136216470C790105166BA100FFE8B8405401004D
S11362266F7000CA6E0800107368464601006F70EF
S113623600AA01006BA000FFE8B00D2017F00100D3
S11362466BA000FFE87801006F7000CA01006BA025
S113625600FFE87E1A8001006BA000FFE882188821
S113626668C8F8306EF800885C001CBA0D005870D8
S11362760A1C7900FFF5A006CB66868A8255860A7
S113628609CC0B76188868C80FF001006BA000FFD5
S1076296E8900FF08A
S113629A01006BA000FFE8941A8001006BA000FFC5
S10962AAE89801006BA05F
S11362B000FFE89C1A8001006BA000FFE8A001002A
S11362C06BA000FFE8A41A8001006BA000FFE8A800
S11362D040306868A8204718A8234720A82B470AFE
S11362E0A82D461C7D40701040167D407030401034
S11362F07C407330460A7D40704040047D407020EE
S11363000B766868A82D47CAA82B47C6A82047C2A2
S1136310A82347BEF9206AA900FFE8866869A93067
S1136320460AF9306AA900FFE8860B761A80010055
S11363306BA000FFE8886868A82A586000800FD027
S11363400BF001006FF000A07308470A01006F70A3
S113635000A00B70400601006F7000A00F851BF0BA

S113636001006FF000967308470A01006F700096F2
S11363701B70400601006F700096690017F0010062
S11363806BA000FFE8884C167D40701001006B2065
S113639000FFE88817B001006BA000FFE888010048
S10963A06B2000FFE888FA
S11363A67A20000002004F0E7D4070007900051828
S11363B66BA000FFE8B80B76686817D0796000FF1A
S11363C617F078006A280000A02A73285870008600
S11363D61A8001006BA000FFE8884064686817D044
S11363E67930003017F001006FF000A07A01000049
S11063F602001A810F907A010000000A5E78
S113640300801001006B2100FFE8881F904D2401D9
S1136413006B2000FFE8887A010000000A5E008019
S11364233601006F7100A00A9001006BA000FFE822
S113643388400E7D407000790005186BA000FFE8CB
S1136443B80B76686817D0796000FF17F078006A95
S1136453280000A02A732846867A00FFFFFFF0166
S11364630069B06868A82E586001000B766868A8B5
S11364732A466C0FD00BF001006FF00096730847A8
S11364830A01006F7000960B70400601006F7000E5
S1136493960F851BF001006FF000A07308470A01F4
S11364A3006F7000A01B70400601006F7000A069AD
S11364B30017F0010069B04C0A7A00FFFFFFF01E8
S11364C30069B0010069307A20000002004F0E7D9D
S11364D3407000790005186BA000FFE8B80B7668DD
S11364E36817D0796000FF17F078006A280000A0CE
S10864F32A732847761F
S11364F81A80010069B04058686817D079300030B5
S113650817F001006FF000A07A01000002001A8161
S11365180F907A010000000A5E00801001006931C3
S11365281F904D1C010069307A010000000A5E00CB
S1136538803601006F7100A00A90010069B0400E17
S11365487D407000790005186BA000FFE8B80B7652
S1136558686817D0796000FF17F078006A28000090
S1136568A02A732846927A000000002001006BA03D
S113657800FFE8AC6868A8684708A86C4704A84CFB
S11365884610686817D017F001006BA000FFE8AC4D
S11365980B766868A825587004A4A8455870021299
S11365A8A8475870020CA8584742A863587002E4D9
S11365B8A8644738A865587001F8A866587001F2AE
S11365C8A867587001ECA8694722A86E5870043070
S11365D8A86F4718A87058700398A873587002F8E2
S10965E8A8754708A8781E
S10D65EE47045A006A8201006B2083
S11365F800FFE8AC7A200000006C46440FD00B90F3
S113660801006FF000AA7308470A01006F7000AA1F
S11366180B70400601006F7000AA0F851B900100E4
S11366286FF000967308470A01006F7000961B709D
S1136638400601006F700096010069025A00678ADC
S113664801006B2000FFE8AC7A2000000068586066
S1136658009A6868A875470CA8584708A87847049B
S1136668A86F46420FD00BF001006FF000AA730821
S1136678470A01006F7000AA0B70400601006F7093
S113668800AA0F851BF001006FF000967308470AF4
S113669801006F7000961B70400601006F70009632
S11366A86900177040400FD00BF001006FF000969F
S11366B87308470A01006F7000960B7040060100CB

S11366C86F7000960F851BF001006FF000AA730826
S11366D8470A01006F7000AA1B70400601006F7023
S10966E800AA690017F08F
S11366EE0F825A00678A6868A875470CA85847082E
S11366FEA8784704A86F46420FD00BF001006FF045
S113670E00967308470A01006F7000960B704006DF
S113671E01006F7000960F851BF001006FF000AA49
S113672E7308470A01006F7000AA1B704006010030
S113673E6F7000AA6900177040400FD00BF0010074
S113674E6FF000AA7308470A01006F7000AA0B705E
S113675E400601006F7000AA0F851BF001006FF059
S113676E00967308470A01006F7000961B7040066F
S113677E01006F700096690017F00F8201006930F7
S113678E7A20FFFFFFF460A7A0000000001010096
S113679E69B0686817D06DF07A01000000020AF143
S11367AE0FA05C00050E0B875A0069F601006B20E3
S11367BE00FFE8AC7A200000004C46420FD00B904D
S11367CE0B9001006FF000AA7308470A01006F7067
S10967DE00AA0B70400647
S11367E401006F7000AA0F851B901B9001006FF0CE
S11367F400967308470A01006F7000961B70404AA5
S113680401006F70009640420FD00B900B90010073
S11368146FF000AA7308470A01006F7000AA0B7097
S1136824400601006F7000AA0F851B901B900100A6
S11368346FF000967308470A01006F7100961B718D
S1136844400601006F7100960F907A0100000080EA
S11368540AF15E006108010069307A20FFFFFFF3F
S1136864460A7A0000000006010069B0686917D17E
S113687401006F70008401006DF001006F700084EB
S113688401006DF07A00000000080AF05C0007863E
S11368940B970B975A0069F60FD00BF001006FF0BA
S11368A400AA7308470A01006F7000AA0B70400620
S11368B401006F7000AA0F851BF001006FF00096B2
S11368C47308470A01006F7000961B7040060100AD
S10968D46F7000966E08D0
S11368DA00015A006A480FD00B9001006FF000AA1A
S11368EA7308470A01006F7000AA0B704006010083
S11368FA6F7000AA0F851B9001006FF00096730852
S113690A470A01006F7000961B70400601006F7002
S113691A0096010069000F825E005AC401006FF0FD
S113692A00AA010069317A21FFFFFFF4712010024
S113693A69311F814C0A0100693001006FF000AA16
S113694A0FF001006BA000FFE8941A8001006BA00E
S113695A00FFE8A001006B2000FFE88801006F71C7
S110696A00AA1A9001006BA000FFE8A85AD4
S1136977006A900FD00B9001006FF00096730847E1
S11369870A01006F7000960B70400601006F7000DC
S1136997960F851B9001006FF000AA7308470A0141
S11369A7006F7000AA1B70400601006F7000AA01F8
S11369B70069007A6000FFFFFFF01006FF000960196
S11369C70069307A20FFFFFFF460A7A00000000C4
S11369D701010069B0790000786DF07A01000000C9
S11369E7020AF101006F7000985C0002CE0B870F5B
S11369F7F00F825E005AC401006FF000AA5A006AC2
S1136A07900FD00B9001006FF000A07308470A01A5
S1136A17006F7000A00B70400601006F7000A00F9D
S1136A27851B900F81730847060F901B7040020F59

S1136A3790010069006B2100FFE8846981404AF8EF
S1136A472568F80FF00F827A000000000101006F3C
S1136A57F000AA0FF101006BA100FFE8941A91015E
S1066A67006BA11D
S1136A6A00FFE8A001006B2100FFE8881A810100FA
S1136A7A6BA100FFE8A8400E790005186BA000FF80
S1136A8AE8B87D4070006868A86E587001B86A2833
S1136A9A00FFE87C7308586001AC7C4073104636EB
S1136AAA01006B2000FFE8A84F2C40207A01000068
S1136ABA00017A0000FFE8865C0013B87A0000FF41
S1136ACAE8A8010069011B710100698101006B20BB
S1136ADA00FFE8A84ED601006B2000FFE89C46247D
S1136AEA01006B2000FFE8A0461A01006B2000FF9B
S1136AFAE8A4461001006F7100AA0FA05C0013748A
S1136B0A5A006C1401006B2000FFE8900FA11A9041
S1136B1A01006FF0009C01006FF000A04F300100EC
S1136B2A6F71009C0FA05C00134A40227A00000098
S1136B3A00880AF07A01000000015C0013367A002B
S1136B4A00FFE89C010069011B71010069810100D2
S1136B5A6B2000FFE89C4ED401006B2000FFE894F1
S1136B6A01006B2100FFE8901A9001006FF0009C6E
S1136B7A01006F7100A00A8101006FF100A00F806C
S1136B8A4F3601006F71009C01006B2000FFE890F3
S1136B9A5C0012E040227A00000000880AF07A01C1
S1136BAA000000015C0012CC7A0000FFE8A001009B
S1136BBA69011B710100698101006B2000FFE8A0D4
S1096BCA4ED401006F71BF
S1116BD000AA01006F7000A01A8101006B2063
S1136BDE00FFE8945C00129840227A0000000088BF
S1136BEE0AF07A01000000015C0012847A0000FFB3
S1136BFEE8A4010069011B710100698101006B208A
S1136C0E00FFE8A44ED47C407310473601006B207E
S1136C1E00FFE8A84F2C40207A01000000017A0003
S1136C2E00FFE8865C0012487A0000FFE8A8010026
S1136C3E69011B710100698101006B2000FFE8A847
S1136C4E4ED60B76404001006FF600A6400E0100B3
S1136C5E6F7000A60B7001006FF000A601006F703D
S1136C6E00A668086EF8009547086E780095A8256B
S1136C7E46DC01006F7100A61AE10FE05C0011F013
S1136C8E01006F7600A6686847145C0012900D0031
S1136C9E460C6A2800FFE87C73085870F5D45C0034
S1136CAE125E6B2000FFE8847A17000000B25E00CC
S1136CBE615C54705E00617E7A37000000247A05B1
S1096CCE000000040AF5BA
S1136CD401006FF0001801006FF1002001006FF153
S1136CE4001C18AA689A6F72003C0C22463CAA58EE
S1136CF4472CAA644712AA69470EAA6F4712AA75BA
S1136D044706AA78471840227A000000000A400682
S1136D147A000000000801006FF00014400C7A00B0
S1136D240000001001006FF000146F70003C792024
S1136D340064470679200069461201006F70001849
S1136D444C0A01006F74001817B4400601006F74F5
S1136D5400181AE61AB30FC4467401006B2000FF2F
S1136D64E88C476AF83068D87A060000000140626C
S1136D740FD00AE00F82010069F00FC001006F71A8
S1136D8400145E00863A0100697068890FA06808E0
S1136D94A8094E0C0FD00AE0680989306889401E9F

S1136DA46F70003C79200078460A0FD00AE0680926
S1136DB4895740080FD00AE06809893768890FC0EA
S1096DC401006F710014D1
S1116DCA5E00863A0F840B760FC4469E6A283D
S1136DD800FFE87C732847626F70003C0C00465A3A
S1136DE8A858471EA86F4706A8784716404C0100C5
S1136DF86F70001847440FE00B760AD0F9306889A2
S1136E08403801006F700018473001006F70002090
S1136E180B7001006FF000201B70F93068890100C6
S1136E286F7000200B7001006FF000201B706E79EB
S1136E38003D68897A03000000020FB00AE00F835F
S1136E4801006FF600146F70003C79200064470658
S1136E5879200069467201006B2000FFE88C460820
S1136E6801006F700018476001006F7000184D42F1
S1136E786A2800FFE87C733847160B7301006F70AC
S1136E8800200B7001006FF000201B70F92B4036B7
S1136E986A2800FFE87C7348472E0B7301006F7064
S1136EA800200B7001006FF000201B70F920688927
S1136EB840160B7301006F7000200B7001006FF018
S1096EC800201B70F92DF0
S1136ECE68897A0600FFE89401006F70001C680C55
S1136EDEAC2B4708AC2D4704AC20460E01006F7057
S1136EEE001C0B70010069E040466A2800FFE87C35
S1136EFE732847326F70003C0C004634A858470A7B
S1136F0EA86F4714A8784702402601006F70001C33
S1136F1E0BF0010069E0401801006F70001C0B704C
S1136F2E010069E0400A01006F70001C010069E076
S1136F3E7A0400FFE88C010069401FB04F14010072
S1136F4E6946010069441AB401006BA400FFE8A06E
S1136F5E400C0FB61A8001006BA000FFE8A07A0464
S1136F6E00FFE8A801006F70001C680BAB2B4708ED
S1136F7EAB2D4704AB20463801006B2000FFE88899
S1136F8E1FE04F2C6A2800FFE886A8304622010036
S10D6F9E6B2000FFE8881AE07A0177
S1136FA800FFE8A0010069120A82010069921A80B1
S1136FB8010069C0401E01006B2000FFE8881FE044
S1136FC84F0C01006B2000FFE8881AE040021A808A
S1136FD8010069C001006F7600141B76401A010096
S1136FE86F7000200B7001006FF000201B700FE121
S1136FF81B760AD1681968890FE64CE201006F70A5
S11370080020189968897A17000000245E00615CE3
S113701854705E00617E7A37000000647A0300FFD3
S1137028E87C7A04000000040AF47A0500FFE88C7F
S11370380F866FF1005001006FF6005CF83068C8E6
S11370487A000000007C0AF07A010000A0225E00AA
S1137058614E0D00587000B87A000000002F0AF046
S113706801006DF07A000000002E0AF001006DF0B7
S113707801006F70008801006DF001006F700088D7
S113708801006DF07A01000000320AF17A00000075
S109709800115E008056AA
S113709E7A17000000100D024752188868E80100A5
S11370AE695001006B2100FFE8881F904F06010015
S11370BE6950400801006B2000FFE88801006BA0B7
S11370CE00FFE8A87A0600FFE8860D207920FFFF6F
S11370DE470C79200001460CF82B5A007E06F82D3A
S10470EE5A44
S11370EF007E06F82A68E85A007E087A000000003E

S11370FF1101006DF00FC10B717A00000000260A19
S113710FF05E00843C0B97400CF8016EF8002F18CB
S113711F886EC800017A000000000101006FF000C3
S113712F60400E01006F7000600B7001006FF00084
S113713F6001006F7000600AC06808A83047E4015F
S113714F006F7000607A20000000014F7A01006F1A
S113715F7000600AC05E005AC40F8201006F700096
S113716F601B7001006F71002A0A8101006FF1002B
S113717F2A7A000000000101006FF000584030012F
S113718F006F7000600AC001006F7100580AC16878
S113719F08689801006F7000580B7001006FF000C2
S11371AF5801006F7000600B7001006FF0006001F9
S11371BF006F7000580FA11F904FC401006F700034
S11371CF580AC0189968890FC00B7001006FF0003F
S10871DF445E005AC4E8
S11371E401006FF000601A8001006FF00058010085
S11371F46F7000607A01000000025E0080100100DD
S11372046FF0004001006F70004401006F71006073
S11372140A901B7001006FF0003C404E01006F7038
S1137224004401006F7100580A9001006FF0004C94
S113723468086EF8005701006F71003C01006F721B
S113724400581AA101006FF1004801006F72004C4D
S1137254681968A901006F710048689801006F718B
S113726400580B7101006FF1005801006F70005852
S113727401006F7100401F904DA201006F70006008
S1137284461AF8306EC800017A00000000010100BC
S11372946FF000601A8001006FF0002A6F700050D5
S11372A479200067470679200047463C7D307050BB
S11372B401006F70002A01006F7100600A901B7057
S11372C47A20FFFFF4D0C01006B2100FFE88CCB
S10972D41F904F0C6F70C8
S11372DA00501BD06FF000504008790000666FF031
S11372EA00506E78002FA80146246838E8186EF813
S11372FA0057A8084706A810470A401A0FE00B765A
S113730AF92B40100FE00B76F920688940080FE04B
S113731A0B76F92D68896F700050792000665860E2
S113732A077201006F70002A58D001861A80401A2A
S113733A0FE00B7601006F7100580AC10B716819CF
S113734A688901006F7000580B7001006FF00058D4
S113735A01006F7100601F904DD61A800F820100E1
S113736A6BA600FFE89401006F70002A01006BA06E
S113737A00FFE8A07C307350460A01006B2000FF2F
S113738AE88C4E067C307320470E0FE00B76F92EFD
S113739A68890FA00B700F827C307350470C7C30C6
S11373AA735047247C307320471E01006BA600FFED
S11373BAE8980100695001006BA000FFE8A40100EE
S10973CA69500FA10A81C6
S11373D00F9201006F70002A0FA10A9001006F71D4
S11373E000600A9001006FF0006001006F71005CA3
S11373F06819A92B4708A92D4704A9204652010063
S11374006B2000FFE88801006F7100601F904F4000
S11374106A2800FFE886A830463601006F70005CDA
S10974200B7001006BA0DC
S113742600FFE89001006B2000FFE88801006F7100
S113743600601A9001006BA000FFE89C1A8001000F
S11374466BA000FFE8A85A007E0401006B2000FF32
S1137456E88801006F7100601F904F4C01006B209C

S113746600FFE88801006F7100601A9001006BA0AD
S113747600FFE8A86E78002FA80146186E78002F43
S1137486A801586009786E780057A8084706A8101F
S11374965860096A7A0000FFE8A8010069011B71B8
S11374A6010069815A007E041A8001006BA000FF67
S11374B6E8A85A007E0401006F70006001006F7136
S11374C6002A0A8101006FF1002A010069520AA10C
S11374D601006FF100521F814C120F914D0E0100F6
S11374E66F7100520B710FC05C00092201006F70AF
S11374F6002A58F0028A6848A83046147A00000029
S1137506000101006FF0004C01006FF00052402CA7
S10975161A8001006FF072
S113751C004C1A8001006FF0005201006F70002ABA
S113752C0B7001006FF0002A01006F7000600B708C
S113753C01006FF000601A8001006FF00058403AB0
S113754C0FE00B7601006F71004C0AC16819688952
S113755C01006F70002A1B7001006FF0002A0100FC
S113756C6F7000580B7001006FF0005801006F70C2
S113757C004C0B7001006FF0004C01006F70002A7F
S113758C4EBE0FE00B76F92E688940240FE00B7684
S113759C01006F71004C0B7101006FF1004C1B71FA
S11375AC0AC168196889010069501B70010069D010
S11375BC01006F7000580B7001006FF00058010050
S11375CC6F70004C01006F7100521A9001006F71C3
S11375DC00601F904C0A01006B2000FFE88C4EAC3E
S11375EC7C307350470C7C307350472E7C307320A7
S11375FC47280100695001006F7100580A8101008E
S107760C6FF10058BF
S107761001006BA661
S113761400FFE8940100695001006BA000FFE8A09B
S113762440226E68FFFA830461A40101B76010003
S11376346F7000581B7001006FF000586E68FFFFF5
S1137644A83047E87C307320464A6E68FFFA82EB3
S1137654464201006B2000FFE8A047067C307350CC
S113766447321B7601006F70005801006B2100FF45
S1137674E8A01A901B7001006FF000581A800100F3
S11376846BA000FFE8A001006F70005C01006BA019
S113769400FFE89401006F70005C6808A82B47089A
S11376A4A82D4704A820465001006B2000FFE8885A
S11376B401006F7100581F904F3E6A2800FFE8864F
S11376C4A830463401006F70005C0B7001006BA09E
S11376D400FFE89001006B2000FFE88801006F7150
S11376E400581A9001006BA000FFE89C1A80010067
S11376F46BA000FFE8A8406201006B2000FFE8884C
S109770401006F71005843
S10B770A1F904F4601006B20A4
S113771200FFE88801006F7100581A9001006BA006
S113772200FFE8A86E78002FA80146146E78002F98
S1137732A80146286E780057A8084704A810461CDB
S11377427A0000FFE8A8010069011B710100698149
S1137752400A1A8001006BA000FFE8A801006B2019
S113776200FFE89401006F71005C1F90586006925D
S113777201006B2000FFE89001006BA000FFE8947A
S11377825A007E046848A83046147A000000001BB
S113779201006FF0004C01006FF00052402A1A8082
S11377A201006FF0004C01006FF0005201006F7096
S11377B2002A0B7001006FF0002A01006F70006055

S11377C20B7001006FF0006001006F70002A4F140C
S11377D27A000000000101006FF0004C0FE00B760D
S11377E2684940060FE00B76F93068897A00000099
S11377F2000101006FF000580FE10B76FA2E689A30
S109780201006F70005845
S11378080B7001006FF0005801006F70002A58C018
S1137818009E01006BA600FFE89401006F70002A28
S113782817B0010069511F814F2001006F70002AB2
S113783817B001006BA000FFE8A001006F70002AD9
S113784801006F7100581A81401801006950010046
S11378586BA000FFE8A00100695001006F71005898
S11378680A8101006FF1005801006F70002A0100BE
S113787869510A81010069D140360FE00B76010096
S11378886F71004C0AC168196889010069501B703F
S1137898010069D001006F7000580B7001006FF090
S11378A8005801006F70004C0B7001006FF0004C22
S11378B801006F70004C01006F7100521A900100B3
S11378C86F7100601F904C0A01006B2000FFE88C69
S11378D84EA87C307350470C7C30735047347C304F
S11378E87320472E010069504F4A01006BA600FF21
S10978F8E898010069504D
S10778FE01006BA077
S113790200FFE8A40100695001006F7100580A8169
S113791201006FF1005840226E68FFFA830461A3B
S113792240101B7601006F7000581B7001006FF04E
S113793200586E68FFFA83047E87C30732046701A
S11379426E68FFFA82E466801006B2000FFE8A0C7
S1137952460A01006B2000FFE8A447067C307350FF
S1137962474E1B7601006F70005801006B2100FF28
S1137972E8A01A9001006B2100FFE8A41A901B7083
S113798201006FF000581A8001006BA000FFE8A409
S113799201006BA000FFE8A001006F70005C010012
S11379A26BA000FFE8941A8001006BA000FFE89827
S11379B201006F70005C6808A82B4708A82D4704D4
S11379C2A820465001006B2000FFE88801006F7178
S11379D200581F904F3E6A2800FFE886A8304634BD
S11379E201006F70005C0B7001006BA000FFE89058
S10779F201006B2002
S11379F600FFE88801006F7100581A9001006BA020
S1137A0600FFE89C1A8001006BA000FFE8A8406213
S1137A1601006B2000FFE88801006F7100581F907A
S1137A264F4601006B2000FFE88801006F71005884
S1137A361A9001006BA000FFE8A86E78002FA8013A
S1137A4646146E78002FA80146286E780057A808BA
S1137A564704A810461C7A0000FFE8A80100690144
S1137A661B7101006981400A1A8001006BA000FFA7
S1137A76E8A801006B2000FFE89401006F71005C29
S1137A861F90461001006B2000FFE89001006BA0D9
S1137A9600FFE8945A007E04010069500B70010050
S1137AA66F7100601F904C0C010069510BF10FC000
S1137AB65C00035A6848A830460E7A0000000001AD
S1137AC601006FF0004840241A8001006FF000485F
S1137AD601006F70002A1B7001006FF0002A01007D
S1137AE66F7000600B7001006FF0006001006F7033
S1097AF6006001006F7146
S1137AFC00481A9001006F72002A0A8201006FF28B
S1137B0C002A0FE00B760AC1681968897A00000015

S1137B1C000101006FF000580FE10B76FA2E689A02
S1137B2C01006F7000580B7001006FF000580100DA
S1137B3C6F700048402E0FE00B7601006F71004C04
S1137B4C0AC168196889010069501B70010069D06A
S1137B5C01006F7000580B7001006FF000580100AA
S1137B6C6F70004C0B7001006FF0004C01006F71D3
S1137B7C00601F904E0A01006B2000FFE88C4EB68C
S1137B8C7C307350470C7C307350472E7C30732001
S1137B9C472801006BA600FFE8940100695001001F
S1057BAC6BA0C9
S1137BAE00FFE8A00100695001006F7100580A81BF
S1137BBE01006FF1005840226E68FFFA830461A8D
S1137BCE40101B7601006F7000581B7001006FF0A0
S1137BDE00586E68FFFA83047E87C307320464A92
S1137BEE6E68FFFA82E464201006B2000FFE8A03F
S1137BFE47067C30735047321B7601006F70005876
S1137C0E01006B2100FFE8A01A901B7001006FF0BA
S1137C1E005801006F70005C01006BA000FFE89438
S1137C2E1A8001006BA000FFE8A06F70005079204E
S1137C3E006546080FE00B76F96540060FE00B76FC
S1137C4EF945688901006F7000580B7001006FF0E1
S1137C5E005801006F70002A4D0A0FE00B76F92BC6
S1137C6E688940160FE00B76F92D688901006F7055
S1137C7E002A17B001006FF0002A01006F70005840
S1137C8E0B7001006FF0005801006F70002A7A200C
S1097C9E000000644D0E1E
S1137CA40BF601006F7000580B800B7040140FE04B
S1137CB40B76F9306889F83068E801006F70005872
S1137CC40BF001006FF000580FE00B7001006FF030
S1137CD4003040360FE01B76010069F001006F703D
S1137CE4002A7A010000000A5E0080108930010036
S1137CF46970688901006F70002A7A010000000A24
S1137D045E00801001006FF0002A01006F70002AEA
S1137D144EC201006F76003001006F70005C68088A
S1137D24A82B4708A82D4704A820465001006B2020
S1137D3400FFE88801006F7100581F904F3E6A28C6
S1137D4400FFE886A830463401006F70005C0B70B6
S1137D5401006BA000FFE89001006B2000FFE8889E
S1137D6401006F7100581A9001006BA000FFE89C9A
S1137D741A8001006BA000FFE8A8406201006B2099
S1137D8400FFE88801006F7100581F904F460100FF
S1057D946B205F
S1137D9600FFE88801006F7100581A9001006BA07C
S1137DA600FFE8A86E78002FA80146146E78002F0E
S1137DB6A80146286E780057A8084704A810461C51
S1137DC67A0000FFE8A8010069011B7101006981BF
S1137DD6400A1A8001006BA000FFE8A801006B208F
S1137DE600FFE89401006F71005C1F9046100100CC
S1137DF66B2000FFE89001006BA000FFE894188851
S1137E0668E87A17000000645E00615C54705E00E7
S1137E16617E0F850F960FD00AE06808A8344F528B
S1137E260FD00AE0F93068891B760FD00AE068099B
S1137E368901688940200FD00AE06808A8394F14E1
S1137E460FD10AE1681888F668986E18FFFF880153
S1137E566E98FFFF1B760FE64EDC6E580001A839BD
S1137E664F106E58000188F66ED8000168588801D5
S1137E7668D85E00615C54705E00617E1B977A036E

S1097E8600FFE87E7A060E
S1137E8C00FFE882010069F00F9501006B2100FFF0
S1137E9CE8787A2100000001462601006DF50F8178
S1137EAC010069305E0085E00B97010069300AD050
S1137EBC010069B0010069600AD0010069E0403A31
S1137ECC19444030010069700B70010069F01B709C
S10F7EDC680817D00100693101006B2217
S1137EE800FFE8B05D207920FFFF471201006960B9
S1137EF80B70010069E00B5417F41FD44DCA0B979C
S1137F085E00615C547001006B2000FFE8787A2002
S1137F1800000001460C01006B2000FFE87E189961
S1137F286889547001006B2000FFE8787A2000000C
S1137F38000146041900547001006B2000FFE87E1D
S10F7F486E090010E94017D10D105470B1
S10BA02200000000000000000000033
S1137F5401006DF501006DF401006DF301006DF294
S1137F6401006DF101006D120100691301006F013D
S1137F740004010069000D840D8C796C7FF0792C69
S1137F847FF047540DAC796C7FF0792C7FF0475622
S1137F940D8C65AC4B5E1AB144087A3000000001C5
S1137FA445141AA0451001F06410475A0C444B0AB7
S1137FB479000002400C0C444BF61900400479008C
S1137FC4FFFF01006D7101006D7201006D7301000B
S1137FD46D7401006D7554700F85796D000F01F098
S1137FE4641546DA409E0FA5796D000F01F06435E0
S1137FF446CC409C01F064207A607FFFFFFFF46ACCF
S10F800401F0643146A67900000140B68B
S11380106DF20D820C2A4A0217B00D994A04D280E0
S113802017B15E00863A0C224A0217B00CAA4A0224
S109803017B16D725470DC
S113803601006DF301006DF20D8252120D9352038E
S113804652100928093801006D7201006D735470CE
S113A02A20202020202020202020606060602020E3
S113A03A2020202020202020202020202020202013
S113A04A48101010101010101010101010101010CB
S113A05A8484848484848484848484841010101010106B
S113A06A10818181818181818181818181818181C4
S113A07A01010101010101010101010101010101078
S113A08A1082828282828282020202020202020295
S113A09A02020202020202020202020210101010203D
S113A0AA00000000000000000000000000000000A3
S113A0BA0000000000000000000000000000000093
S113A0CA0000000000000000000000000000000083
S113A0DA0000000000000000000000000000000073
S113A0EA0000000000000000000000000000000063
S113A0FA0000000000000000000000000000000053
S113A10A0000000000000000000000000000000042
S109A11A00000000000000000000000000003C
S10DA12000000000000000000000000000032
S11380565E00617E7A370000003C7A0400000000C63
S11380660AF47A05000000140AF501006FF00038DF
S113807601006FF1003401006F73005C7A020000A7
S1138086000801006FF2002C19226FF2002A7A060B
S1138096000000540AF6686AEA7F46620FE00B7036
S11380A67A01000000075E0086760D0047501A80AD
S11380B6401A01006F70003401006F7100380A9096
S11380C61899688901006F7000380B7001006FF012

S11380D600387A20000000084DD81A80010069B0E4
S11380E67C607370470A01006F700060F9FF4008F7
S11380F601006F700060F90168895A00842E7A00C6
S11381060000001C0AF001006DF07A01000000284F
S11381160AF101006F70005C01006DF001006F70E1
S1138126005C01006DF001006F70006C5E008A0652
S11381367A170000000C01006F7000606808A8FF42
S1078146460C7A0066
S113814A000000540AF07D0072706F7000247920D9
S113815A07FF464A6E680001A8F0463A0FE00BF0A3
S113816A7A01000000065E0086760D0047280100AA
S113817A6F7000606808A8014608790000015A0078
S113818A843001006F7000606808A8FF4610790008
S113819AFFFF5A008430790000025A0084306F705E
S11381AA0024793003FF6FF000244D1E01006F7025
S11381BA005801006DF001006F70005801006DF066
S11381CA5E0087960B970B97401E01006F7000584D
S11381DA01006DF001006F70005801006DF05E0040
S11381EA87960B970B971B500D0618886EF800237A
S11381FA0B560D6017F001006F7100381A900100D9
S113820A69B0010069F10FD1010069705E0088B29B
S113821A01006F70003801006F71002C5E0080103E
S113822A01006FF000301AE67A000000000801002E
S109823A6F7100301A9081
S11382400F82401401006F7000300AE00AD00FD192
S11382500AE1680868980B760FA01F864DE6400A6E
S11382600FD00AE0189968890B767A260000000877
S11382704DEE6F7000326F78002E52806F71003AAE
S113828019016DF17A01000000080FD05E00883AF1
S11382900B8701006F7600381B760FC10FE05E007D
S11382A088B20FE001006F71002C5E0080100100A6
S11382B06FF000301AE67A000000000801006F71C9
S11382C000301A900F82401401006F7000300AE0F2
S11382D00AC00FC10AE1680868980B760FA01F86D1
S11382E04DE6400A0FC00AE0189968890B767A2692
S11382F0000000084DEE6F7000326F78002E528040
S11383006F71003A19011B516DF17A0100000008E9
S11383100FC05E00883A0B8701006F7000340100C4
S11383206DF00100693101006DF17A010000002454
S10983300AF16F70002C3E
S11383365E00869E0B970B977A06000000040AF6EA
S11383467A000000000801006DF001006F7100382B
S11383560FE05E0085E00B977A000000000801003D
S10A83666DF00FD10FE05E83
S113836D008AAA0B970D004F12790000016FF000E0
S113837D2A010069300B705A0084267A0000000030
S113838D0801006DF001006F7100380FE05E00858C
S113839DE00B977A000000000801006DF00FD10F7C
S11383ADE05E008AAA0B970D004632010069300B7F
S11383BD70010069B001006F70003401006DF001B0
S11383CD00693301006DF37A01000000240AF16F97
S11383DD70002C5E00869E0B970B9740447A00002D
S11383ED00000801006DF001006F7100380FE05EB1
S11383FD0085E00B977A000000000801006DF00F77
S113840DC10FE05E008AAA0B970D004C146F70002C
S113841D2A460E010069301B70010069B05A0083B2
S112842D1819007A170000003C5E00615C547060

S113843C5E00617E7A370000002C0FF30F8601007B
S113844C6FF100107A000000000101006FF00018BA
S113845C01006F7500440A95188868D87A000000EB
S113846C000801006DF00FE17A000000000C0AF027
S113847C5E0085E00B9701006F7000445A0085CCB9
S113848C0FA00B707A01000000055E0080100B70CA
S113849C10307A06000000081A867A0000000008E3
S11384AC1AE001006DF00FA11031103110317A1167
S11384BC0000A12A0AE10FB00AE05E0085E00B97E9
S11384CC7A000000000801006FF000281A800100F8
S11384DC6FF0002001006FF0001C7A04000000080C
S11384EC1AE401006FF400145A00858E01006F70BA
S11384FC001401006DF00FB10AE17A000000000CCA
S113850C0AF00AE05E008AAA0B970D004D3C0100AD
S113851C6DF40FB00AE001006DF07A010000001059
S109852C0AF10AE11A80C6
S11385325E00895A0B970B9717F001006FF0001832
S113854201006F70002801006F7100200A81010091
S11385526FF1002001006F7000287A010000000211
S11385625E00801001006FF000284730790000019F
S11385726DF00FB00AE00FC15E0088E00B870100C7
S11385826F70001C0B7001006FF0001C01006F7014
S1138592001C7A200000000458D0FF5A01006F70BB
S11385A20018461C6E78002388306CD84004F830DB
S11385B268D81B7501006F7000101F8544F04012CC
S11385C26E78002388306CD80FA01B700F8258C0BE
S11185D2FEB87A170000002C5E00615C547046
S113A12A00000000000000080000000000000050CA
S113A13A000000000000003200000000000001F4090
S113A14A0000000000001388000000000000C350008
S113A15A00000000007A12000000000004C4B400EA
S113A16A000000002FAF080000000001DCD65000F9
S113A17A00000012A05F2000000000BA43B74000AD
S113A18A00000746A5288000000048C27395000016
S113A19A0002D79883D20000001C6BF52634000016
S10BA1AA011C37937E0800003D
S11385E05E00617E0F850F9401006F7600181FC532
S11385F047401FC544200FD30FC21AC440120FB007
S11386000B730FA10B710F921B71681968890B749F
S11386101FE445EA401C0FD30AE30AE40FC21AC45D
S1138620400C0FA01B700F8268086CB80B741FE41A
S10D863045F00FD05E00615C54704A
S113863A01006DF20D9946100D82177253120DA89F
S113864A53100D810D28401E0F920D811771790861
S113865A0010121012311AA144020AA11B5846F241
S10F866A12101710177001006D7254708D
S11386765E00617E0F840F951AE6400E0FC00AE076
S1138686680947041900400A0B761FD64DEE790098
S10B869600015E00615C5470F9
S113869E5E00617E7A370000000C7A05000000014F
S11386AE0AF50D0C0F967904000801006DF57A0199
S11386BE0000000E0AF101006F70002817B05E0073
S11386CE8B700B9701006DF66DFC7A0000000010A5
S11386DE0AF00FD15E008AE80B970B87790C003FE7
S11386EE6F70000A190C0DC017F001D053400D0E18
S11386FE7900004219C017F001006DF07A010000F5
S113870E00090FD05E008DBC0B977906000840144C

S113871E0D6019E017F00AD00D6117F10AD1680840
S113872E68981B561DE64CE8400C0D6017F00AD0F6
S113873E189968891B560D664CF00DE05240190CC2
S113874E6DFC7A0100000090FD05E0088E00B87F4
S113875E7900003F6FF0000A7A01000000400FD04D
S113876E5E008CD27A000000000801006DF00FD17C
S113877E01006F70002C5E0085E00B977A170000E6
S109878E000C5E00615CBB
S105879454701C
S113879601006DF27A370000001A7A000000001813
S11387A60AF001006F71002601006DF101006F717F
S11387B6002601006DF17A01000000080AF10100AC
S11387C66DF15E008E587A170000000C6F71001869
S11387D60FF05E0090D80F817A020000A1B27A00F2
S11387E6000000080AF05E0091C47A000000001041
S11387F60AF001006F71000C01006DF101006F7149
S1138806000C01006DF17A01000000080AF1010075
S11388166DF15E008F5E7A170000000C7A0000008F
S113882600100AF05E00601C7A170000001A0100AF
S10788366D72547098
S10BA1B23FD34395810624DD30
S113883A5E00617E1B971B87010069F00F946F79B5
S113884A001E790D0008199D4754790100FF0DD2C6
S113885A1A0A4B04101140F80C9B18DD1B740FC63F
S113886A4028010069740AE468450CB816850FC0EC
S113887A0D915E009094684814D868C80DD01A0800
S113888A4B04110540F80C5D1B760FE64CD40CDD46
S113889A4706790100014002191117F10F900B875E
S10B88AA0B975E00615C547042
S11388B25E00617E0F860F957A00000000080100BA
S11388C26DF01036103610367A010000A1BA0AE1B3
S11188D20FD05E0085E00B975E00615C547072
S113A1BA00000000000000001000000000000058C
S113A1CA00000000000000001900000000000007DEC
S113A1DA000000000000000271000000000000C35BE
S113A1EA00000000000003D09000000000001312DBD
S113A1FA000000000005F5E10000000001DCD6528
S113A20A00000000009502F90000000002E90EDDDB
S113A21A000000000E8D4A510000000048C27395E9
S113A22A000000016BCC41E9000000071AFD498DCB
S113A23A0000002386F26FC1000000B1A2BC2EC544
S113A24A000003782DACE9D900001158E460913D70
S113A25A000056BC75E2D6310001B1AE4D6E2EF543
S113A26A000878678326EAC9002A5A058FC295ED42
S113A27A00D3C21BCECCEDA10422CA8B0A00A425AB
S113A28A14ADF4B7320334B96765C793FA10079D5F
S11388E05E00617E7A370000000A0F83010069F1A0
S11388F06F790022790C0008199C4752FAFF0DC0CA
S11389001A084B04110A40F80CA218CC1AE64026A8
S11389100FB50AE568540C2816840FD00D915E003C
S113892090B6685814C868D80DC01A084B041004D0
S113893040F80C4C0B76010069701F864DD20CCCAD
S11389404706790100014002191117F10F907A17B8
S10D89500000000A5E00615C547031
S113895A5E00617E7A370000000C01006FF00004AC
S113896A0F9601006F7500280AD61B7601006F73F4
S113897A00240AD31B737A02000000011AC440447C

S113898A6868175068391751191017F01AC001008F
S113899A69F04C14010069717A11000001000100A9
S11389AA69F1790000014002190017F00F840100F0
S11389BA697047041A800F826E78000368E81B7592
S11389CA1B761B730FD546B87A2400000001460EA6
S11389DA01006F70000446067900FFFF40120FA0E2
S11389EA7A200000000146041900400479000001BE
S10F89FA7A170000000C5E00615C5470F2
S1138A065E00617E7A03000000070F820F95010066
S1138A166F7600207A04000000180AF46941796130
S1138A2680007921800046067901FFFF4004790121
S1138A3600010FA06889694079607FF01190119059
S1138A461190119069D07A000000000701006DF0C3
S1138A560B740FC10FE05E0085E00B9779000003EE
S1138A666DF00FB10FE05E00883A0B8769504F0631
S1138A767D607070402869500B5069D07D607270BC
S1138A864016790000016DF00FB10FE05E00883AE1
S1138A960B8769501B5069D07C60737047E45E0096
S1078AA6615C547048
S1138AAA5E00617E0F860F9301006F7400180FE555
S1138ABA0FC44604190040201AE6400A6C586C3960
S1138ACA1C9846060B761FC645F21B756858175045
S1118ADA1B73683B175319305E00615C5470C8
S1138AE85E00617E7A37000000100FF60F850F9441
S1138AF869506F710028091069D001006DF60100F3
S1138B086F71002E0FC05E0094A00B977C6073708A
S1138B18470869500B5069D0400C7A0100000010D7
S1138B280FE05E00944E7A000000004201006DF0F1
S1138B387A01000000100FE05E008DBC0B976E6891
S1138B480008E8E06EE800087A0000000009010068
S1138B586DF00FE10FC05E0085E00B977A170000F8
S10B8B6800105E00615C547013
S1138B705E00617E7A37000000187A050000000964
S1138B800AF50F840F920FC44D087A030000000109
S1138B9040147A03FFFFFFFFF0FC07A01FFFFFFFFFBF
S1138BA05E0080360F840FC07A010000001B5E0058
S1138BB080100F860FC07A010000001B5E0080103A
S1138BC00F947A2300000001462E7A00000000086B
S1138BD001006DF00FE11031103110317A110000F6
S1138BE0A29A0FD05E0085E00B970FE0103078005B
S1138BF06B210000A36A40320FC446021B767A0041
S1138C000000000801006DF00FE110311031103148
S1138C107A110000A3020FD05E0085E00B970FE0EE
S1138C20103078006B210000A3846FF100120FC491
S1138C3047747A23FFFFFFFFF460A7A000000001BF8
S1138C401AC00F8401006F70003001006DF00FA196
S1138C500FC05E00962E0B970FE646087A2300009E
S1078C600001476263
S1138C6401006F70003001006DF00FA069006DF01A
S1138C747A00000000180AF00FD15E008AE80B970F
S1138C840B877A01000000400FD05E008CD279205C
S1138C940001460E6F7000120B506FF000127D50EE
S1138CA470700FA06F71001269817A0000000008D0
S1138CB401006DF00FD101006F7000345E0085E098
S1118CC40B977A17000000185E00615C547075
S113A29A800000000000000000CECB8F27F4200F3A85
S113A2AAA70C3C40A64E6C5286F0AC99B4E8DAFD92

S113A2BADA01EE641A708DEAB01AE745B101E9E4EE
S113A2CA8E41ADE9FBEBEC27DE5D3EF282A242E812B
S113A2DAB9A74A0637CE2EE195F83D0A1FB69CD98F
S113A2EAF24A01A73CF2DCD0C3B8358109E84F072B
S113A2FA9E19DB92B4E31BA99E74D1B791E07E4801
S113A30AC428D05AA4751E4CF2D56790AB41C2A497
S113A31A964E858C91BA2655BA121A4650E4DDEC4C
S113A32AE65829B3046B0AFA8E938662882AF53FA4
S113A33AB080392CC4349DEDDA7F5BF59096684979
S113A34A873E4F75E2224E68A76C582338ED2623C1
S113A35ACF42894A5DCE35EB8049A4AC0C5811AE85
S113A36A0000005900B3010D016601C0021A02730D
S113A37A02CD0327038003DA0434FFA6FF4CFEF25F
S109A38AFE99FE3FFDE514
S111A390FD8CFD32FCD8FC7FFC25FBCBFB7261
S1138CD25E00617E7A370000000C0F840F957A06DE
S1138CE2000000081AB30FD00FE15E0080100AC023
S1138CF20F820FD00FE15E008010790000801A0905
S1138D024B04119040F86EF8000511086EF8000B41
S1138D1211086EF800041B75010069F50FD00FE10D
S1138D225E0080100AC00F85010069700FE15E00CA
S1138D328010790600801A094B04119640F80FA09F
S1138D4268086E790005169847280FA068066E78A2
S1138D52000B166846086E780004166847087A0303
S1138D6200000001400C685816E847067A03000029
S1138D7200017A2300000001463240140CE8175028
S1138D8217106859168968D9100E46041B75FE011F
S1138D92685816E847041FC544E21FC545086858CA
S1138DA214E868D8400679000001400219007A17D6
S10D8DB20000000C5E00615C5470C9
S1138DBC5E00617E1B971B970F82010069F17A039A
S1138DCC0000000801006F7000200FB15E008010DE
S1138DDC0FA60A8601006F7000200FB15E00801091
S1138DEC790400801A094B04119440F8686816C87A
S1138DFC46500CCD1855400414D5110D0CDD46F816
S1138E0C686816584708686814C868E840340FA0A7
S1138E1C010069710A90010069F001006F70002074
S1138E2C0FB15E0080100FA50A85400C68684708D7
S1138E3C685814C868D8400A0B76010069701F86FD
S10F8E4C45EA0B970B975E00615C5470C5
S1138E585E00617E0F857A040000000701006F70D1
S1138E68002001006DF001006F70002001006DF01B
S1138E785E0093FC0B970B970D0647487926000174
S1138E88460A790004B86BA000FFE8B87926000207
S1138E98460A7900044C6BA000FFE8B819667A000B
S1138EA80000001C0AF00D6117F10A90F9FF6889A8
S1138EB80B56792600084DE67A000000001C0AF0DC
S1138EC85A008F4E7A000000001C0AF07A01000055
S1138ED8A39E5E0060960D00470C190069D07A00C6
S1138EE80000A39E40607A060000001C0AF669632E
S1138EF81113111311131113796307FF793303FE48
S1138F0869D36950791003FE462869500B5069D01C
S1138F1869607960800F69E00FE30B73400E0FC13E
S1138F280FB05E00944E69501B5069D06960734856
S1138F3847EC69607960800F79403FE069E07A0027
S1098F480000001C0AF00A
S1138F4E01006F7100185E0061085E00615C547071

S10BA39E00000000000000000B4
S1138F5E5E00617E0F8601006F70002001006DF0D0
S1138F6E01006F70002001006DF05E0093FC0B9703
S1138F7E0B970D05474079250001460A790004B881
S1138F8E6BA000FFE8B879250002460A7900044C6D
S1138F9E6BA000FFE8B819667A000000001C0AF007
S1138FAE0D6117F10A90F9FF68890B5679260008AF
S1138FBE4DE65A00907C7A050000001C0AF56955AF
S1138FCE1115111511151115796507FF793503FF64
S1138FDE0D5C4C0E7A000000A3A60FE15E00610843
S1138FEE40607A000000001C0AF00FE15E00610889
S1138FFE792C00344C4C0FE50B950BF579090034A5
S113900E19C90D9017F07901001001D053100D09F5
S113901E400A0FD01BF5191169811B590D994EF298
S113902E7900003419C017F07901001001D05310E4
S113903E7909FFFF1B584B04101940F869506690CD
S107904E69D07A0167
S11390520000001C0AF10FE20F905E005BAA0FE012
S11390627A010000A3A65E0060B20D00470C7A00ED
S11390720000001C0AF07D0070707A000000001CE2
S11390820AF001006F7100185E0061085E00615C06
S1059092547015
S10BA3A6000000000000000000AC
S113909401006DF20D1A4F14792A00084D04FA00E9
S11390A44008680A100A1B5A4EFA688A01006D7256
S10590B45470F3
S11390B601006DF20D1A4F14792A00084D04FA00C7
S11390C64008680A110A1B5A4EFA688A01006D7233
S10590D65470D1
S11390D801006DF119990D11471A4A06790908001B
S11390E817917919040F1B59101144FA10311031D3
S11390F810311031010069811A9101006F81000458
S109910801006D715470BB
S113910E0F8501F064155860009A792407FF471400
S113911E0D44586000820FA501F064355860007845
S113912E580000800FA501F0643558600076406842
S113913E0FA501F06435466C0DCC465C0F8501F02E
S113914E64154654405E0F8501F06415473C10319B
S113915E1230792800104C0C1B5C10311230792818
S113916E00104DF4580000C40FA501F06435471AE2
S113917E10331232792A00104C0C1B541033123256
S113918E792A00104DF4580000A81AA21AB3687D6C
S113919E100D131A58000228790107FF1AA21AB3E9
S11391AE580001FA790107FF1AA27A03000000089A
S11391BE68FA580001E801006DF601006DF5010033
S11391CE6DF401006DF301006DF201006DF101000C
S11391DE6DF07A3700000018691D692565D569F5AC
S11391EE691C111C111C111C111C796C07FF6924BD
S10991FE111411141114F9
S11392041114796407FF01006D1001006911796875
S1139214000F01006F23000401006922796A000F23
S1139224792C07FF5870FEE2792407FF5870FF0A70
S11392340DCC5870FF1A0D445870FF36094C793C15
S113924403FF792C07FF58C0FF58792CFFCB58D064
S1139254FF426FFC000279480010794A00100100B4
S11392646FF000040F9601006FF2000801006FF322
S1139274000C0D1552356FF500160D34529452B18E

S11392840A946511990009D44406791C00019900D4
S11392946FF400140DC50D1D18990D0452340AC53D
S11392A499000DE452B40AC599000D6452240AC509
S11392B499006FF500120DD50D1D18990D845234C4
S11392C40AC50D0452B40AC599000DE452240AC513
S11392D499000D6452A40AC599006FF500100DD5C9
S11392E40D1D18990DB352830AB50D0452240AC5F2
S10992F499000DE452A4F1
S11392FA0AC59900528209D24404791A0001091A4B
S113930A6F74000852040AC26F7400085284094A2F
S113931A0D5B6F73001001006F7400126F7D0016EE
S113932A19556F71000211321333133413350DA01B
S113933A79600100471211321333133413350B5179
S113934A792107FF5870FE5401F064454702700BF8
S113935A0D114E2E113213334402700B0D114A04B0
S113936A0B5140F0732B47180CB8E80B47127A13CA
S113937A00000008440A0B72792A00804D020B513F
S113938A4020732B471C0CB8E80B47167A130000CE
S113939A000844020B72792A01004D061132133375
S11393AA0B51113213331132133311321333796AD6
S11393BA000F1011101110111011641A101A687885
S11393CA1008131A7A170000001801006D700100C3
S11393DA698201006F83000401006D7101006D72DF
S10793EA01006D739B
S11193EE01006D7401006D7501006D76547001
S11393FC01006DF57A01000000080AF16818E87F96
S113940CA87F460A0B716818E8F0A8F04704190006
S113941C402A6818F01050006898460A0B711AD548
S113942C400E681847067900000140100B710B754C
S113943C7A25000000064DEA7900000201006D75E3
S105944C547057
S113944E5E00617E0F840F931AD51B730FB64028EF
S113945E0FC30AE30FB26838E880175017700F83F3
S113946E0FA06809100968890FD547080FC00AE0D5
S113947E7D0070000FB51B760FE64CD40FD5470653
S113948E790100014002191117F10F905E00615C22
S105949E547005
S11394A05E00617E7A37000000387A050000000410
S11394B00AF50F840F967A000000001001006DF08A
S11394C0191101006F7000545E0097120B970FE3A0
S11394D00B930BF37A160000000701006FF60034BC
S11394E0790600030FC00B900BF001006FF000280A
S11394F07A140000000701006FF4002C5A00961C38
S11395006838460C01006F7000346809587000FA1F
S11395107A000000001001006DF019110FD05E00F9
S113952097120B9701006F70002801006FF0003055
S113953001006F74002C790E0003407C01006F70F2
S113954000301A916809FA0810311A0A4EFA1A8083
S1139550684801F064101A916839FA0810311A0A40
S11395604EFA01006F720034010069F01A80682816
S113957001F06401010069705E00803601006FF044
S113958000247A000000000401006DF07A0100005D
S109959000280AF10DE2C0
S1139596096217F210320AD20FA05E0096CA0B9721
S11395A61B5E01006F7000301BF001006FF000308E
S11395B61BF40DEE4C807926000346247A00000046
S11395C6000A01006DF00D6417F40FC110310AD1C2

S11395D601006F7000540AC00AC05E0085E0402295
S11395E67A000000000A01006DF00D6417F40FC144
S11395F610310AD101006F7000540AC00AC05E0020
S113960696CA0B971B561BF301006F7000341BF0B1
S113961601006FF000340D6658C0FEDE7A170000B5
S10B962600385E00615C547022
S113962E5E00617E7A370000000C0F860F940D608A
S113963E7910003F69C00FF10FE05E0088B27A0027
S113964E0000000801006DF0191101006F70002871
S113965E5E0097120B971A800F8219550FF6401062
S113966E0B760FA00B700F8269407930000869C02A
S113967E686847ECFB804004110B0B55686816B8FD
S113968E47F66DF57A03000000080FA01A830FB199
S113969E0FE05E00883A0B876940195069C00100DC
S11396AE6DF30FE101006F7000285E0085E00B97EC
S10F96BE7A170000000C5E00615C547021
S11396CA5E00617E0F860F9401006F7500180AD63B
S11396DA1B760AD41B740FC319CC4022686817502F
S11396EA68391751091009C00D0468E817F479009D
S11396FA010001D053040D4C1B751B761B730FD548
S10B970A46DA5E00615C547055
S11397125E00617E1B870F8469F101006F73001A7B
S11397220FC51AE6400C0FD00B756E7900016889DC
S11397320B761FB645F00FC00B875E00615C547059
S9030000FD

PAGE 1

PROGRAM NAME =

```

1          1 ;messagemap.src
2          2
3          3 .CPU 300HA
4 000000   4 .SECTIONV,CODE,LOCATE=H'000000
5          5
6          6 ;C言語の関数を参照
7          7 .IMPORT _main;C言語の関数mainを参照
8          8 .IMPORT _usb_int;C言語の関数usb_intを参照
9          9 .IMPORT _InterruptITU0;C言語の関数InterruptITU0を参照
10         10
11         11 ;-----
12         12 ;リセットベクトルの転送先ラベルが_startになっています
13         13 ;リセットベクトル
14 000000 00000000   14 .DATA.L _start
15         15
16         16 ;-----
17         17 ;リセットベクトルに続く1番から60番までの割り込みベクトル
18         18 ;について、使用しない割り込みベクトルはラベルint_error
19         19 ;に転送されます
20         20 ;割り込みベクタ
21         21 ;1 Reserved
22 000004 00000000   22 _INT_Reserved1: .DATA.L int_error
23         23 ;2 Reserved
24 000008 00000000   24 _INT_Reserved2: .DATA.L int_error

```

25	25	; 3 Reserved
26	00000C 00000000	26 _INT_Reserved3: .DATA.L int_error
27	27	; 4 Reserved
28	000010 00000000	28 _INT_Reserved4: .DATA.L int_error
29	29	; 5 Reserved
30	000014 00000000	30 _INT_Reserved5: .DATA.L int_error
31	31	; 6 Reserved
32	000018 00000000	32 _INT_Reserved6: .DATA.L int_error
33	33	; 7 NMI
34	00001C 00000000	34 _INT_NMI: .DATA.L int_error
35	35	; 8 TRAP
36	000020 00000000	36 _INT_TRAP1: .DATA.L int_error
37	37	; 9 TRAP
38	000024 00000000	38 _INT_TRAP2: .DATA.L int_error
39	39	; 10 TRAP
40	000028 00000000	40 _INT_TRAP3: .DATA.L int_error
41	41	; 11 TRAP
42	00002C 00000000	42 _INT_TRAP4: .DATA.L int_error
43	43	; 12 IRQ0
44	000030 00000000	44 IRQ0: .DATA.L int_error
45	45	; 13 IRQ1
46	000034 00000000	46 IRQ1: .DATA.L int_error
47	47	; 14 IRQ2
48	000038 00000000	48 IRQ2: .DATA.L int_error
49	49	; 15 IRQ3
50	00003C 00000000	50 IRQ3: .DATA.L int_error
51	51	; 16 IRQ4
52	000040 00000000	52 IRQ4: .DATA.L int_error
53	53	; 17 IRQ5

54 000044 00000000 54 IRQ5: .DATA.L usb_interrupt ; USB割り込み
55 55 ; 18 Reserved
56 000048 00000000 56 _INT_Reserved18: .DATA.L int_error
57 57 ; 19 Reserved

*** H8/300H ASSEMBLER (Evaluation software) Ver.1.0 *** 12/28/24 19:15:28

PAGE 2

PROGRAM NAME =

58 00004C 00000000 58 _INT_Reserved19: .DATA.L int_error
59 59 ; 20 WOVI
60 000050 00000000 60 _INT_WOVI: .DATA.L int_error
61 61 ; 21 CMI
62 000054 00000000 62 _INT_CMI: .DATA.L int_error
63 63 ; 22 Reserved
64 000058 00000000 64 _INT_Reserved22: .DATA.L int_error
65 65 ; 23 Reserved
66 00005C 00000000 66 _INT_Reserved23: .DATA.L int_error
67 67 ; 24 IMIA0
68 000060 00000000 68 _INT_IMIA0: .DATA.L int_error
69 69 ; 25 IMIB0
70 000064 00000000 70 _INT_IMIB0: .DATA.L int_error
71 71 ; タイマ0割り込みは、ラベル_ITU_OVI_0に転送されます
72 72 ; 26 OVI0
73 000068 00000000 73 _INT_OVI0: .DATA.L _ITU_OVI_0 ; タイマ0割り込み
74 74 ; 27 Reserved
75 00006C 00000000 75 _INT_Reserved27: .DATA.L int_error
76 76 ; 28 IMIA1
77 000070 00000000 77 _INT_IMIA1: .DATA.L int_error
78 78 ; 29 IMIB1

79	000074	00000000	79	_INT_IMIB1: .DATA.L int_error
80			80	;30 OVI1
81	000078	00000000	81	_INT_OVI1: .DATA.L int_error
82			82	;31 Reserved
83	00007C	00000000	83	_INT_Reserved31: .DATA.L int_error
84			84	;32 IMIA2
85	000080	00000000	85	_INT_IMIA2: .DATA.L int_error
86			86	;33 IMIB2
87	000084	00000000	87	_INT_IMIB2: .DATA.L int_error
88			88	;34 OVI2
89	000088	00000000	89	_INT_OVI2: .DATA.L int_error
90			90	;35 Reserved
91	00008C	00000000	91	_INT_Reserved35: .DATA.L int_error
92			92	;36 IMIA3
93	000090	00000000	93	_INT_IMIA3: .DATA.L int_error
94			94	;37 IMIB3
95	000094	00000000	95	_INT_IMIB3: .DATA.L int_error
96			96	;38 OVI3
97	000098	00000000	97	_INT_OVI3: .DATA.L int_error
98			98	;39 Reserved
99	00009C	00000000	99	_INT_Reserved39: .DATA.L int_error
100			100	;40 IMIA4
101	0000A0	00000000	101	_INT_IMIA4: .DATA.L int_error
102			102	;41 IMIB4
103	0000A4	00000000	103	_INT_IMIB4: .DATA.L int_error
104			104	;42 OVI4
105	0000A8	00000000	105	_INT_OVI4: .DATA.L int_error
106			106	;43 Reserved
107	0000AC	00000000	107	_INT_Reserved43: .DATA.L int_error

```

108          108 ;44 DEND0A
109 0000B0 00000000      109  _INT_DEND0A: .DATA.L int_error
110          110 ;45 DEND0B
111 0000B4 00000000      111  _INT_DEND0B: .DATA.L int_error
112          112 ;46 DEND1A
113 0000B8 00000000      113  _INT_DEND1A: .DATA.L int_error
114          114 ;47 DEND1B

```

*** H8/300H ASSEMBLER (Evaluation software) Ver.1.0 *** 12/28/24 19:15:28

PAGE 3

PROGRAM NAME =

```

115 0000BC 00000000      115  _INT_DEND1B: .DATA.L int_error
116          116 ;48 Reserved
117 0000C0 00000000      117  _INT_Reserved48: .DATA.L int_error
118          118 ;49 Reserved
119 0000C4 00000000      119  _INT_Reserved49: .DATA.L int_error
120          120 ;50 Reserved
121 0000C8 00000000      121  _INT_Reserved50: .DATA.L int_error
122          122 ;51 Reserved
123 0000CC 00000000      123  _INT_Reserved51: .DATA.L int_error
124          124 ;52 ERI0
125 0000D0 00000000      125  _INT_ERI0: .DATA.L int_error
126          126 ;53 RXI0
127 0000D4 00000000      127  _INT_RXI0: .DATA.L int_error
128          128 ;54 TXI0
129 0000D8 00000000      129  _INT_TXI0: .DATA.L int_error
130          130 ;55 TEI0
131 0000DC 00000000      131  _INT_TEI0: .DATA.L int_error

```

```

132          132 ;56 ERI1
133 0000E0 00000000      133  _INT_ERI1: .DATA.L int_error
134          134 ;57 RXI1
135 0000E4 00000000      135  _INT_RXI1: .DATA.L int_error
136          136 ;58 TXI1
137 0000E8 00000000      137  _INT_TXI1: .DATA.L int_error
138          138 ;59 TEI1
139 0000EC 00000000      139  _INT_TEI1: .DATA.L int_error
140          140 ;60 ADI
141 0000F0 00000000      141  _INT_ADI: .DATA.L int_error
142          142
143          143 ;-----
144 000000      144  .SECTIONP, CODE, ALIGN=2
145          145 ;_startのラベルから処理を開始
146          146 ;リセットベクトルの転送先
147 000000      147  _start:
148 000000 7A0700FFFF10    148  mov.l #H'0FFFF10, er7
149          149 ;初期化付きデータを使用する場合、RAMに転送する
150          150 ;message.MAPのメモリアドレス使用状況を見る
151          151 ;メモリアドレスが重複するとコンパイルエラーになる
152          152 ;makefile.sub も D(99C0), C(9A00) 等必要があれば合わせる
153 000006 7A00000099C0    153  mov.l #H'99C0, er0 ; 転送元(99C0)
154 00000C 7A0100FFE000    154  mov.l #H'0FFE000, er1 ; 転送先
155 000012 7A0200000000    155  mov.l #DATA_END, er2 ; 転送終了
156 000018          156  init_loop:
157 000018 1F92          157  cmp.l er1, er2
158 00001A 58700008          158  beq init_end
159 00001E 6C0B          159  mov.b @er0+, r3l
160 000020 689B          160  mov.br 3l, @er1

```



```

161 0000220B71      161  inc.l#1,er1
162 00002440F2      162  bra init_loop
163 000026          163  init_end:
164              164  ;C言語の関数mainを呼び出しています
165              165  ;C言語の関数mainは、void main(void);という形で、
166              166  ;main.cに記述があります
167 000026 5E000000    167  jsr @_main
168              168  ;割り込み未使用
169 00002A          169  int_error:
170              170  ;rte (returnと同じ意味)で終了
171 00002A 5670      171  rte

```

*** H8/300H ASSEMBLER (Evaluation software) Ver.1.0 *** 12/28/24 19:15:28

PAGE 4

PROGRAM NAME =

```

172              172
173              173  ;-----
174              174  ;USB割り込みからの転送先
175 00002C          175  usb_interrupt:
176              176  ;スタック退避
177 00002C 01006DF0    177  push.l er0
178 000030 01006DF1    178  push.l er1
179 000034 01006DF2    179  push.l er2
180 000038 01006DF3    180  push.l er3
181 00003C 01006DF4    181  push.l er4
182 000040 01006DF5    182  push.l er5
183 000044 01006DF6    183  push.l er6
184              184  ;C言語の関数usb_intを呼び出しています
185 000048 5E000000    185  jsr @_usb_int

```

```

186          186 ;スタック戻
187 00004C 01006D76      187  pop.l er6
188 000050 01006D75      188  pop.l er5
189 000054 01006D74      189  pop.l er4
190 000058 01006D73      190  pop.l er3
191 00005C 01006D72      191  pop.l er2
192 000060 01006D71      192  pop.l er1
193 000064 01006D70      193  pop.l er0
194          194 ;終了
195 000068 5670          195  rte
196          196
197          197 ;-----
198          198 ;タイマ0割り込みからの転送先
199 00006A          199  _ITU_OVI_0:
200          200 ;スタック 退避
201 00006A 01006DF0      201  push.l er0
202 00006E 01006DF1      202  push.l er1
203 000072 01006DF2      203  push.l er2
204 000076 01006DF3      204  push.l er3
205 00007A 01006DF4      205  push.l er4
206 00007E 01006DF5      206  push.l er5
207 000082 01006DF6      207  push.l er6
208          208 ;C言語の関数InterruptITU0を呼び出しています
209          209 ;C言語の関数InterruptITU0 は void InterruptITU0(void);
210          210 ;という形で、Timer.hTimer.cに記述があります
211 000086 5E000000      211  jsr @_InterruptITU0
212          212 ;スタック戻
213 00008A 01006D76      213  pop.l er6
214 00008E 01006D75      214  pop.l er5

```

```

215 000092 01006D74      215  pop.l er4
216 000096 01006D73      216  pop.l er3
217 00009A 01006D72      217  pop.l er2
218 00009E 01006D71      218  pop.l er1
219 0000A2 01006D70      219  pop.l er0
220          220  ;終了
221 0000A6 5670          221  rte
222          222
223          223  ;-----
224          224  ;C言語から
225          225  ;_EnableInterrupt(割り込み許可)
226          226  ;_DisableInterrupt(割り込み禁止)
227          227  ;を呼び出せるようにしています
228          228  ;C言語の Panel.hに 外部参照プロトタイプ宣言があります

```

*** H8/300H ASSEMBLER (Evaluation software) Ver.1.0 *** 12/28/24 19:15:28

PAGE 5

PROGRAM NAME =

```

229          229  ;extern void EnableInterrupt(void);
230          230  ;extern void DisableInterrupt(void);
231          231  ;C言語からの呼び出し名は、
232          232  ;EnableInterrupt();
233          233  ;DisableInterrupt();
234          234  ;です
235          235  ;割り込み許可、禁止ルーチン
236          236  .EXPORT _EnableInterrupt,_DisableInterrupt
237 0000A8          237  _EnableInterrupt:
238 0000A8 063F      238  andc.b #H'3f,ccr

```

```

239 0000AA 5470      239  rts
240 0000AC          240  _DisableInterrupt:
241 0000AC 04C0     241  orc.b#H'c0,ccr
242 0000AE 5470     242  rts
243                243
244                244  ;-----
245 000000          245  .SECTIOND,DATA
246                246
247 000000          247  .SECTIONB,DATA
248 000000 00000002  248  DATA_END: .RES.W 1
249                249
250                250  .END

```

*****TOTAL ERRORS 0

*****TOTAL WARNINGS 0

*** H8/300H ASSEMBLER (Evaluation software) Ver.1.0 *** 12/28/24 19:15:28

PAGE 6

*** CROSS REFERENCE LIST

NAME	SECTION	ATTR	VALUE	SEQUENCE
B	B	SCT	00000000	247*
D	D	SCT	00000000	245*
DATA_END	B		00000000	155 248*
IRQ0	V		00000030	44*
IRQ1	V		00000034	46*
IRQ2	V		00000038	48*
IRQ3	V		0000003C	50*
IRQ4	V		00000040	52*

IRQ5	V	00000044	54*
P	P	SCT 00000000	144*
V	V	SCT 00000000	4*
_DisableInterrupt	P	EXPT 000000AC	236 240*
_EnableInterrupt	P	EXPT 000000A8	236 237*
_INT_ADI	V	000000F0	141*
_INT_CMI	V	00000054	62*
_INT_DEND0A	V	000000B0	109*
_INT_DEND0B	V	000000B4	111*
_INT_DEND1A	V	000000B8	113*
_INT_DEND1B	V	000000BC	115*
_INT_ERI0	V	000000D0	125*
_INT_ERI1	V	000000E0	133*
_INT_IMIA0	V	00000060	68*
_INT_IMIA1	V	00000070	77*
_INT_IMIA2	V	00000080	85*
_INT_IMIA3	V	00000090	93*
_INT_IMIA4	V	000000A0	101*
_INT_IMIB0	V	00000064	70*
_INT_IMIB1	V	00000074	79*
_INT_IMIB2	V	00000084	87*
_INT_IMIB3	V	00000094	95*
_INT_IMIB4	V	000000A4	103*
_INT_NMI	V	0000001C	34*
_INT_OVI0	V	00000068	73*
_INT_OVI1	V	00000078	81*
_INT_OVI2	V	00000088	89*
_INT_OVI3	V	00000098	97*
_INT_OVI4	V	000000A8	105*

_INT_RXI0	V	000000D4	127*
_INT_RXI1	V	000000E4	135*
_INT_Reserved1	V	00000004	22*
_INT_Reserved18	V	00000048	56*
_INT_Reserved19	V	0000004C	58*
_INT_Reserved2	V	00000008	24*
_INT_Reserved22	V	00000058	64*
_INT_Reserved23	V	0000005C	66*
_INT_Reserved27	V	0000006C	75*
_INT_Reserved3	V	0000000C	26*
_INT_Reserved31	V	0000007C	83*
_INT_Reserved35	V	0000008C	91*
_INT_Reserved39	V	0000009C	99*
_INT_Reserved4	V	00000010	28*
_INT_Reserved43	V	000000AC	107*
_INT_Reserved48	V	000000C0	117*
_INT_Reserved49	V	000000C4	119*

*** H8/300H ASSEMBLER (Evaluation software) Ver.1.0 *** 12/28/24 19:15:28

PAGE 7

*** CROSS REFERENCE LIST

NAME	SECTION	ATTR	VALUE	SEQUENCE
_INT_Reserved5	V		00000014	30*
_INT_Reserved50	V		000000C8	121*
_INT_Reserved51	V		000000CC	123*
_INT_Reserved6	V		00000018	32*
_INT_TEI0	V		000000DC	131*

```

_INT_TEI1          V      000000EC 139*
_INT_TRAP1        V      00000020 36*
_INT_TRAP2        V      00000024 38*
_INT_TRAP3        V      00000028 40*
_INT_TRAP4        V      0000002C 42*
_INT_TXI0         V      000000D8 129*
_INT_TXI1         V      000000E8 137*
_INT_WOVI         V      00000050 60*
_ITU_OVI_0        P      0000006A 73 199*
_InterruptITU0    IMPT 00000000 9 211
_main             IMPT 00000000 7 167
_start           P      00000000 14 147*
_usb_int         IMPT 00000000 8 185
init_end         P      00000026 158 163*
init_loop        P      00000018 156* 162
int_error        P      0000002A 22 24 26 28 30 32 34 36 38 40 42 44
                  46 48 50 52 56 58 60 62 64 66 68 70
                  75 77 79 81 83 85 87 89 91 93 95 97
                  99 101 103 105 107 109 111 113 115 117 119 121
                  123 125 127 129 131 133 135 137 139 141 169*
usb_interrupt     P      0000002C 54 175*

```

*** H8/300H ASSEMBLER (Evaluation software) Ver.1.0 *** 12/28/24 19:15:28

PAGE 8

*** SECTION DATA LIST

SECTION	ATTRIBUTE	SIZE	START
---------	-----------	------	-------

V	ABS-CODE	00000F4	000000
P	REL-CODE	00000B0	
D	REL-DATA	0000000	
B	REL-DATA	0000002	

MAKE Version 5.2 Copyright (c) 1987, 2000 Borland

```
bcc32 -O2 -w -tWC -D"USE_BCC" -c Panel.c
```

Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland

Panel.c:

```
bcc32 -O2 -w -tWC -D"USE_BCC" -c Timer.c
```

Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland

Timer.c:

警告 W8004 Timer.c 198: 'This' に代入した値は使われていない(関数 delete_)

```
bcc32 -O2 -w -tWC -D"USE_BCC" -c EV_Time.c
```

Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland

EV_Time.c:

```
bcc32 -O2 -w -tWC -D"USE_BCC" -c EV_File.c
```

Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland

EV_File.c:

警告 W8004 EV_File.c 74: 'Ret' に代入した値は使われていない(関数 Write)

警告 W8004 EV_File.c 108: 'Ret' に代入した値は使われていない(関数 WriteString)

警告 W8071 EV_File.c 166: 変換によって有効桁が失われる(関数 Read)

警告 W8004 EV_File.c 162: 'Ret' に代入した値は使われていない(関数 Read)

警告 W8004 EV_File.c 203: 'Ret' に代入した値は使われていない(関数 ReadString)

警告 W8004 EV_File.c 229: 'Ret' に代入した値は使われていない(関数 PermitCommand_Read)

警告 W8004 EV_File.c 247: 'Ret' に代入した値は使われていない(関数 PermitCommand_Write)

警告 W8004 EV_File.c 265: 'Ret' に代入した値は使われていない(関数 Command_Read)

警告 W8004 EV_File.c 283: 'Ret' に代入した値は使われていない(関数 Command_Write)

警告 W8004 EV_File.c 301: 'Ret' に代入した値は使われていない(関数 PermitTurnOpen_Read)

警告 W8004 EV_File.c 319: 'Ret' に代入した値は使われていない(関数 PermitTurnOpen_Write)

警告 W8066 EV_File.c 343: 実行されないコード(関数 Motor_Write)

警告 W8066 EV_File.c 364: 実行されないコード(関数 Motor_Read)

警告 W8066 EV_File.c 367: 実行されないコード(関数 Motor_Read)

警告 W8066 EV_File.c 382: 実行されないコード(関数 Limit_Read)

警告 W8066 EV_File.c 385: 実行されないコード(関数 Limit_Read)

bcc32 -O2 -w -tWC -D"USE_BCC" -c EV_UpDown.c

Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland

EV_UpDown.c:

bcc32 -O2 -w -tWC -D"USE_BCC" -c EV_OpenClose.c

Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland

EV_OpenClose.c:

bcc32 -O2 -w -tWC -D"USE_BCC" -c EV_Display.c

Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland

EV_Display.c:

bcc32 -O2 -w -tWC -D"USE_BCC" -c EV_Input.c

Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland

EV_Input.c:

bcc32 -O2 -w -tWC -D"USE_BCC" -c EV_Controller.c

Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland

EV_Controller.c:

bcc32 -O2 -w -tWC -D"USE_BCC" -c EV_Puls.c

Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland

EV_Puls.c:

警告 W8057 EV_Puls.c 53: パラメータ 'addressDataSet' は一度も使用されない(関数 EV_Set)

警告 W8057 EV_Puls.c 53: パラメータ 'dataSet' は一度も使用されない(関数 EV_Set)

警告 W8057 EV_Puls.c 53: パラメータ 'addressClockSet' は一度も使用されない(関数 EV_Set)

警告 W8057 EV_Puls.c 53: パラメータ 'clockSet' は一度も使用されない(関数 EV_Set)

警告 W8057 EV_Puls.c 182: パラメータ 'th' は一度も使用されない(関数 EV_Puls)

bcc32 -O2 -w -tWC -D"USE_BCC" -c EV_Simulator.c

Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland

EV_Simulator.c:

警告 W8019 EV_Simulator.c 68: コードは効果を持たない(関数 OnSimulator)

警告 W8019 EV_Simulator.c 86: コードは効果を持たない(関数 OnSimulator)

警告 W8019 EV_Simulator.c 104: コードは効果を持たない(関数 OnSimulator)

警告 W8019 EV_Simulator.c 122: コードは効果を持たない(関数 OnSimulator)

bcc32 -O2 -w -tWC -D"USE_BCC" -c EV_Queue.c

Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland

EV_Queue.c:

警告 W8004 EV_Queue.c 420: 'p_ch' に代入した値は使われていない(関数 EV_Q_Command_Read)

bcc32 -O2 -w -tWC -D"USE_BCC" -c EV_Twin1_Queue.c

Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland

EV_Twin1_Queue.c:

警告 W8004 EV_Twin1_Queue.c 420: 'p_ch' に代入した値は使われていない(関数

EV_Twin1_Q_Command_Read)

bcc32 -O2 -w -tWC -D"USE_BCC" -c EV_Twin2_Queue.c

Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland

EV_Twin2_Queue.c:

警告 W8004 EV_Twin2_Queue.c 420: 'p_ch' に代入した値は使われていない(関数

EV_Twin2_Q_Command_Read)

bcc32 -O2 -w -tWC -D"USE_BCC" -c EV_UserController.c

Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland

EV_UserController.c:

bcc32 -O2 -w -tWC -D"USE_BCC" -c EV_Twin1_Controller.c

Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland

EV_Twin1_Controller.c:

bcc32 -O2 -w -tWC -D"USE_BCC" -c EV_Twin2_Controller.c

Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland

EV_Twin2_Controller.c:

bcc32 -O2 -w -tWC -D"USE_BCC" -c EV_UserSimulator.c

Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland

EV_UserSimulator.c:

```
bcc32 -O2 -w -tWC -D"USE_BCC" -c EV_Twin1_Simulator.c
```

Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland

EV_Twin1_Simulator.c:

```
bcc32 -O2 -w -tWC -D"USE_BCC" -c EV_Twin2_Simulator.c
```

Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland

EV_Twin2_Simulator.c:

```
bcc32 -O2 -w -tWC -D"USE_BCC" -c main.c
```

Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland

main.c:

警告 W8004 main.c 350: 'key' に代入した値は使われていない(関数 Run)

```
ilink32 /Tpe -L"C:¥borland¥bcc55¥Lib" Panel.obj Timer.obj EV_Time.obj EV_File.obj
```

```
EV_UpDown.obj EV_OpenClose.obj EV_Display.obj EV_Input.obj EV_Controller.obj EV_Puls.obj
```

```
EV_Simulator.obj EV_Queue.obj EV_Twin1_Queue.obj EV_Twin2_Queue.obj EV_UserController.obj
```

```
EV_Twin1_Controller.obj EV_Twin2_Controller.obj EV_UserSimulator.obj EV_Twin1_Simulator.obj
```

```
EV_Twin2_Simulator.obj main.obj c0x32.obj,main.exe,,cw32.lib import32.lib
```

Turbo Incremental Link 5.00 Copyright (c) 1997, 2000 Borland

MAKE Version 5.2 Copyright (c) 1987, 2000 Borland

```
del *.obj
```

```
del main.tds
```

```
del main.ilc
```

```
del main.ild
```

```
del main.ilf
```

```
del main.ils
```

H8/300H ASSEMBLER (Evaluation software) Ver.1.0

```
*****TOTAL ERRORS    0
```

```
*****TOTAL WARNINGS  0
```

H8/300H LINKAGE EDITOR (Evaluation software) Ver.1.0

: OUTPUT machine.abs

: PRINT message.map

: INPUT messagemap, main, EV_Simulator, EV_Puls, EV_Controller, EV_Input, EV_Display,
EV_OpenClose, EV_UpDown, EV_File, EV_Time, Timer, Panel, sci, lcd, usb

: LIB c:\h8\akic\c38\hab

: START R(0FFE000), P(200), D(99C0), C(9A00)

: ROM (D, R)

: EXIT

LINKAGE EDITOR COMPLETED

H8/300H OBJECT CONVERTER (Evaluation software) Ver.1.0

OBJECT CONVERTER COMPLETED

EV_Queue.c: In function `EV_Q_Command_Read`:

EV_Queue.c:365:8: warning: variable `p_ch` set but not used [-Wunused-but-set-variable]

```
char *p_ch;  
    ^~~~
```

```
EV_Twin2_Queue.c: 関数 `EV_Twin2_Q_Command_Read' 内:  
EV_Twin2_Queue.c:365:8: 警告: 変数 `p_ch' が設定されましたが使用されていません [-Wunused-but-  
set-variable]  
    char *p_ch;  
        ^  
EV_Twin1_Queue.c: 関数 `EV_Twin1_Q_Command_Read' 内:  
EV_Twin1_Queue.c:365:8: 警告: 変数 `p_ch' が設定されましたが使用されていません [-Wunused-but-  
set-variable]  
    char *p_ch;  
        ^  
EV_Queue.c: 関数 `EV_Q_Command_Read' 内:  
EV_Queue.c:365:8: 警告: 変数 `p_ch' が設定されましたが使用されていません [-Wunused-but-set-  
variable]  
    char *p_ch;  
        ^
```



```

-- MySQL dump 10.16 Distrib 10.1.23-MariaDB, for debian-linux-gnueabi (armv7l)
--
-- Host: localhost Database: ev001
-----
-- Server version 10.1.23-MariaDB-9+deb9u1

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `s_info`
--

DROP TABLE IF EXISTS `s_info`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `s_info` (
  `no` decimal(1,0) NOT NULL COMMENT '行番号',
  `ch_command` varchar(1) COLLATE utf8_bin NOT NULL COMMENT '命令入力',
  PRIMARY KEY (`no`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='ログ';
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `s_info`
--

LOCK TABLES `s_info` WRITE;
/*!40000 ALTER TABLE `s_info` DISABLE KEYS */;
INSERT INTO `s_info` VALUES (1,'N');
/*!40000 ALTER TABLE `s_info` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `s_log`
--

DROP TABLE IF EXISTS `s_log`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `s_log` (
  `lg_ymd` decimal(8,0) NOT NULL COMMENT '日付',
  `lg_hms` decimal(6,0) NOT NULL COMMENT '時刻',
  `lg_safety` varchar(1) COLLATE utf8_bin NOT NULL COMMENT '安全スイッチ',
  `lg_limit` varchar(9) COLLATE utf8_bin NOT NULL COMMENT 'リミットスイッチ',
  `lg_command` varchar(1) COLLATE utf8_bin NOT NULL COMMENT '命令入力',
  `lg_permitcommand` varchar(1) COLLATE utf8_bin NOT NULL COMMENT '命令許可',
  `lg_permitturnopen` varchar(1) COLLATE utf8_bin NOT NULL COMMENT '反転開許可',
  `lg_motor` varchar(1) COLLATE utf8_bin NOT NULL COMMENT 'モーター出力',
  PRIMARY KEY (`lg_ymd`,`lg_hms`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='';

```



```
0211108,180816,'r','yynnyynn','N','c','N','s'),(20211108,180817,'r','yynnyynn','N','c','N','s'),(20211108,180818,'r','yynnyynn','N','c','N','s'),(20211108,180819,'r','yynnyynn','N','c','N','s'),(20211108,180820,'r','yynnyynn','N','c','N','s'),(20211108,180821,'r','yynnyynn','N','c','N','s'),(20211108,180822,'r','yynnyynn','N','c','N','s'),(20211108,180823,'r','yynnyynn','N','c','N','s'),(20211108,180824,'r','yynnyynn','N','c','N','s'),(20211108,180825,'r','yynnyynn','N','c','N','s'),(20211108,180826,'r','yynnyynn','N','c','N','s'),(20211108,180827,'r','yynnyynn','N','c','N','s'),(20211108,180828,'r','yynnyynn','N','c','N','s'),(20211108,180829,'r','yynnyynn','N','c','N','s'),(20211108,180830,'r','yynnyynn','N','c','N','s'),(20211108,180831,'r','yynnyynn','N','c','N','s'),(20211108,180832,'r','yynnyynn','N','c','N','s'),(20211108,180833,'r','yynnyynn','N','c','N','s'),(20211108,180834,'r','yynnyynn','N','c','N','s'),(20211108,180835,'r','yynnyynn','N','c','N','s'),(20211108,180836,'r','yynnyynn','N','c','N','s'),(20211108,180837,'r','yynnyynn','q','c','N','s');
/*!40000 ALTER TABLE `s_log` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

-- Dump completed on 2021-11-08 19:36:04

著者：

████████████████████

████

████