

# ETロボコンシミュレータ

## 2020

## 開発の覚書

2020年7月19日(日)

ETロボコン中四国地区実行委員会



# はじめに

国内で年初から感染が拡大したコロナウイルスの蔓延により、ETロボコンはその開催形態を大きく変えることになり、リアルな走行体から、シミュレータ上の走行体での競技となりました。

ただ、このシミュレータは開発途上であり、まだその開発方法は不明な部分が多く、バグ等で不安定な部分があり実行委員も参加者に指導できる状態ではありません。

今後、シミュレータの仕様が大きく変わってくることも考えられます。

ここに書いたのは、「シミュレータを動かしてみた」程度で、動かす時の最低限の注意事項とデバッグ手法についてのメモ書きです。公式の解説が公開された場合は、そちらを参照して下さい。

# サンプルプログラムのある場所について

make sample で実行されているのは、以下のディレクトリの中のソースファイルです。

```
C:\Users\[ユーザ名]\AppData\Local\Packages\
CanonicalGroupLimited.Ubuntu18.04onWindows_79rhkp1fndgsc\
LocalState\rootfs\home\[ユーザ名]\etrobo\hrp3\sdk\workspace\
sample_c4\app.c
```

初心者の方は、このsample\_c4/app.cファイルをベースにして、プログラムを始めてみて下さい。

## プロジェクトディレクトリを指定してコンパイルとシミュレータを実行する

プロジェクトディレクトリを指定してコンパイルとシミュレータを実行するには、以下のようにプロジェクト名を指定してmakeして下さい。

```
$ make app=LapRace sim up  
$ make app=sample_c4 sim up
```

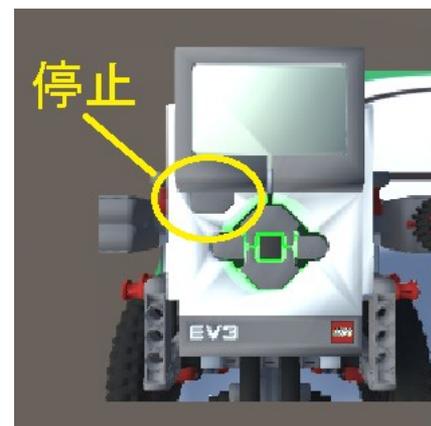
なお、workspace内のディレクトリやファイルは、wsl内で動作しているubuntu上から操作して下さい。Windowsから操作すると、パーミッション等がおかしくなり、ファイルを開けないなどの問題が発生します。

## 走行開始

シミュレータを起動後、走行フィールドが表示されたら、スペースキーを押して下さい。走行体が走行を開始します。

## 走行停止

走行体を停止するには、画面右に表示されている走行体画面のバックボタンをクリックして下さい。



[終了(ESC)]ボタンではありません。これは、シミュレータの終了です。シミュレータだけを終了すると、ログがうまく残せないなどの不具合が発生する事があります。

### ログの表示

sample\_c4/app.cファイルの中に、以下のような関数が書かれています。  
これは、コンソールにメッセージを表示するものです。

```
_log("HackEV sample_c4");
```

この関数をデバッグやログの記録に使用すると良いでしょう。  
ただし、表示できるのは、文字列だけです。変数を直接表示することはできません。変数の内容は、文字列に変換してから表示して下さい。

## 走行体のログのとり方(タイムスタンプ用の関数)

ここからは、ログのとり方について解説します。

sample\_c4/app.cの中に、以下のような関数を定義して下さい。

タイムスタンプを記録するために、現在時刻を取得します。

// 現在時刻を取得する。offsetで、測定開始時刻の補正を行う。

```
long getTime(long offset)
{
    SYSTIM tim;
    get_tim(&tim);
    return ((long)tim - offset);
}
```

## 走行体のログのとり方(変数と配列の定義)

sample\_c4/app.cのメインタスクの先頭部分に以下の変数と配列を追加して下さい。

```
/* メインタスク */  
void main_task(intptr_t unused)  
{  
    long starttime;           // 追加 走行開始時刻  
    char str[256];           // 追加 文字列表示用バッファ  
    uint8_t reflect;         // 追加 カラーセンサ値  
    int32_t motorCountL;     // 追加 左モータの動き  
    int32_t motorCountR;     // 追加 右モータの動き
```

## 走行体のログのとり方(走行開始時刻の記録)

sample\_c4/app.cの走行のメインループ直前に、以下のコードを追記して下さい。スタート時刻を記録しておくためのものです。

```
starttime = getTime(0L); // スタート時刻を記録しておく。
```

```
/**  
 * Main loop  
 */  
while(1)  
{  
    .....  
    .....  
}
```

## 走行体のログのとり方(走行ログの出力)

sample\_c4/app.cの走行のメインループ内に、以下のログを出力するコードを追記して下さい。場所は、各自で工夫して下さい。

```
/**
 * Main loop
 */
While(1) {
    tslp_tsk(4 * 1000U); /* 4msec周期起動 */
    reflect = ev3_color_sensor_get_reflect(color_sensor); //カラーセンサの値取得
    motorCountL = ev3_motor_get_counts(left_motor); //モータの角位置取得
    motorCountR = ev3_motor_get_counts(right_motor); //モータの角位置取得
    sprintf(str, "%ld,%d,%d,%d", getTime(starttime),
                (int)reflect, (int)motorCountL, (int)motorCountR);
    _log(str); // コンソールに表示する。
}
```

## 走行体のログのとり方(実行結果)

ここまでのソースコードの改良を行って実行すると、コンソールには、以下のように、記録した時刻とその時のセンサー等の値が表示されます。  
この記録を元にして、デバッグや走行のチューニングに活用してください。

```
1880686,26,523,631  
1888740,26,526,635  
1900823,26,528,639  
1908863,25,531,642  
1920988,25,533,646  
1941111,26,538,652  
1949151,26,541,656  
1961276,26,543,659  
1969316,27,548,666  
1981399,27,548,666  
1989439,27,553,674  
2001564,27,553,674  
2009604,26,555,677  
2021687,26,557,681  
2029727,27,560,684
```

## 走行体のログのとり方(ログファイルの記録[リダイレクト])

前ページのように、単に表示するだけでは、必要な値を探し出して見ることは大変です。以下のようにリダイレクトを利用して、表示内容をファイルに落とすといろいろと応用ができて便利です。CSV形式になっているので、表計算ソフトで読み込む事も可能です。

```
$ make app=LapRace sim up > logfile.csv
```