LEGO Mindstorms EV3
BricxCC による C 言語プログラミング

（1）準備

（2）ソフトウエア起動方法

（3）モーターを動かす

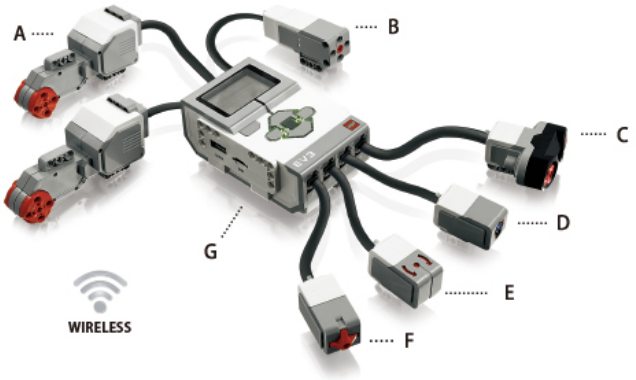
（4）センサー読み取り値の LCD 表示

（5）超音波センサーによる障害物回避

（6）EV3 本体でのプログラム起動

使用ソフトウエア：Bricx Command Center (BricxCC)

EV3 ロボットの制御を行うための、C 言語プログラミングツール。



A:サーボモーターL　　B:サーボモーターM　　C:超音波センサー
D:カラーセンサー　　E:ジャイロセンサー　　F:タッチセンサー
G:インテリジェントブロック EV3

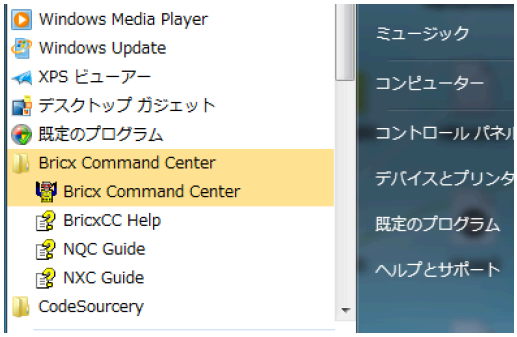
（1）準備

BricxCC プログラムファイルを保存するフォルダ（"BricxCCProject"など）を作成する。jissenPBL.h ファイルを以下よりダウンロードし、フォルダ内に入れる。

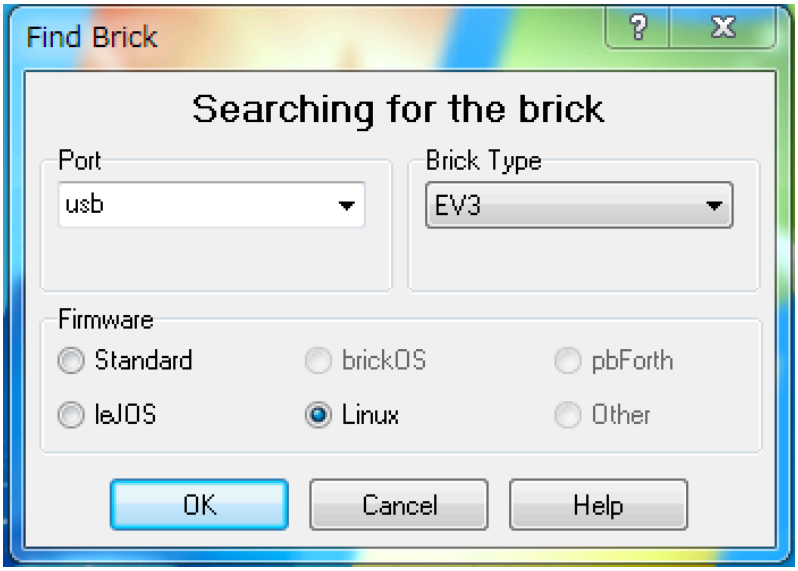
http://www.mns.kyutech.ac.jp/~hanazawa/education/downloads/

（2）ソフトウエア起動方法

デスクトップ上のショートカットまたはスタートメニューから、“Bricx Command Center（BricxCC)”を起動する。



最初に表示される接続設定ダイアログで、
port: usb、Brick Type: EV3、Firmware: Linx を設定



（3）モーターを動かす

プログラムの作成・実行には、同一フォルダの中に、ヘッダファイル jissenPBL.h、プログラムファイル（Program.c など）、プロジェクトファイル（Program.prj など）の３ファイルが必要である。

まずプログラムファイルを作成する。File メニュー　→　New で新しいプログラムを作成し、以下のプログラムを入力する。

```
#include "jissenPBL.h"

int main()
{
  OutputInit();

  OnFwdEx(OUT_AB,70,0); //AB Port Motor Start Power 70
  Wait(1000);   //Wait 1000msec
  Off(OUT_AB); //AB Port Motor Stop
}
```

File メニュー　→　Save でプログラムファイルを保存する。ファイルダイアログが表示されたら、ファイルの名を Motor.c とし、保存場所に、「準備」で作成した BricxCC 用フォルダを指定して、「保存」ボタンをクリックする。

次に、プロジェクトファイルを作成する。File メニュー → New で新しいプログラムを作成し、以下を入力する。

C:\BricxCC\lms_api\ev3_button.c
C:\BricxCC\lms_api\ev3_command.c
C:\BricxCC\lms_api\ev3_lcd.c
C:\BricxCC\lms_api\ev3_output.c
C:\BricxCC\lms_api\ev3_sound.c
C:\BricxCC\lms_api\ev3_timer.c

File メニュー → Save でプログラムファイルを保存する。ファイルダイアログが表示されたら、ファイルの名を Motor.prj とし、保存場所に「準備」で作成した BricxCC 用フォルダを指定して、「保存」ボタンをクリックする。

A および B ポートにモーターを接続する。

Motor.c ファイルを表示した状態で、Compile→Compile でプログラムがコンパイルされ、実行ファイルが作成される。さらに、Compile→Download and Run で、プログラムが EV3 にダウンロードされ、実行される。Motor.c プログラムは、A および B ポートに接続されたモーターを 1 秒間回転させる。

（注 1）プログラムを修正した場合は、必ず File メニュ → Save で保存をしてからコンパイルを行う。保存しないと、変更は反映されない。

（注 2）コンパイル時にエラーダイアログが表示されたら、View メニュー → Show Code/Error|Warning Listing を選択し、エラー内容を表示する。

＜項目解説＞
#include "jissenPBL.h"　必要なヘッダファイル読み込み

int main()　メイン関数（C 言語テキスト参照）

OutputInit();　出力ポートの初期化

Wait(Time_ms);　指定時間待つ
　Time_ms：ミリ秒での時間指定

OnFwdEx(Port,Power,Reset); モーターの正方向回転
OnRevEx(Port,Power,Reset); モーターの逆方向回転
Off(Port); モーター停止

　Port：駆動するモーターのポート（A～C）指定
　#define OUT_A　　0x01 /*!< Output port A */
　#define OUT_B　　0x02 /*!< Output port B */
　#define OUT_C　　0x04 /*!< Output port C */
　#define OUT_D　　0x08 /*!< Output port D */
　#define OUT_AB　 0x03 /*!< Output ports A and B */
　#define OUT_AC　 0x05 /*!< Output ports A and C */
　#define OUT_AD　 0x09 /*!< Output ports A and D */
　#define OUT_BC　 0x06 /*!< Output ports B and C */
　#define OUT_BD　 0x0a /*!< Output ports B and D */
　#define OUT_CD　 0x0c /*!< Output ports C and D */
　#define OUT_ABC　0x07 /*!< Output ports A, B, and C */
　#define OUT_BCD　0x0e /*!< Output ports B, C, and D */
　#define OUT_ABCD 0x0f /*!< Output ports A, B, C, and D */
　#define OUT_ALL　0x0f /*!< All output ports (A-D) */

Power：モーターの駆動力指定（0〜100)、負値で逆回転

Reset：モーターの回転カウンタのリセット制御
#define RESET_NONE 0x00 /*!< No counters will be reset */

#define RESET_COUNT 0x08 /*!< Reset the internal tachometer counter */

#define RESET_BLOCK_COUNT 0x20 /*!< Reset the block tachometer counter */

#define RESET_ROTATION_COUNT 0x40 /*!< Reset the rotation counter */

#define RESET_BLOCKANDTACHO 0x28 /*!< Reset both the internal counter and the block counter */

#define RESET_ALL 0x68 /*!< Reset all tachometer counters */

（４）センサー読み取り値の LCD 表示

接続

ポート１：タッチセンサー：
ポート２：ジャイロセンサー：
ポート３：カラーセンサー：
ポート４：超音波センサー：

　センサーの値を読み取り、EV3 本体の LCD に表示するプログラムを作成する。File メニュー　→　New で新しいプログラムを作成し、以下のプログラムを入力する。

```
#include "jissenPBL.h"

int main()
{
	int i;
	int sensorTouch1, sensorTouch2;
	int sensorGyro, sensorColor, sensorUSonic;
	char sensval[64];

	LcdInit();
	LcdSelectFont(2);
	LcdRefresh();

	LcdText( 1, 0, 104, "Start!");
	Wait(1000);

	initSensor();
	setSensorPort(CH_1,TOUCH, 0);
```

```
    setSensorPort(CH_2,GYRO, 0);
    setSensorPort(CH_3,COLOR, 0);
    setSensorPort(CH_4,USONIC, 0);

    startSensor();

    for (i = 0; i < 1000; i++) {
      sensorTouch1 = getSensor(CH_1);
      sensorGyro = getSensor(CH_2);
      sensorColor = getSensor(CH_3);
      sensorUSonic = getSensor(CH_4);

      sprintf(sensval, "Sensor %d", i);
      LcdScroll(20);
      LcdText( 1, 0, 104, sensval);

      sprintf(sensval,"T:%d                G:%d",sensorTouch1,
sensorGyro);
      LcdScroll(20);
      LcdText( 1, 0, 104, sensval);

      sprintf(sensval,"C:%d          U:%d",          sensorColor,
sensorUSonic);
      LcdScroll(20);
      LcdText( 1, 0, 104, sensval);

      if (sensorTouch1) break;
      Wait(500);
    }
    closeSensor();
}
```

ファイル名を Sensor.c とし、BricxCC 用フォルダに保存する。
Motor.prj を複製し、ファイル名を Sensor.prj に変更する。

Compile および Download and Run で実行する。

プログラムの動作：
　実行すると、EV3 本体 LDC パネルの１行目に繰り返し処理の回数、２行目にタッチセンサーとジャイロセンサーの値、３行目にカラーセンサーと超音波センサーの値が表示される。値更新によって表示が上にスクロールしていく。プログラムを途中で終了するには、タッチセンサーを押す。
　タッチセンサー：起動時の値は０、タッチセンサーを押すと値が１になり、プログラムが終了する。
ジャイロセンサー：センサーを回転させると値が変化する。
カラーセンサー：反射率モードのため、センサーを白い紙などに近づけると値が大きくなり、黒い紙などに近づけると値が小さくなる。
超音波センサー：センサーの前に手をかざすと値が変化する。

＜項目解説＞
int i；　繰り返し処理用の変数
int sensorTouch1, sensorTouch2;　タッチセンサー値格納用変数
int sensorGyro, sensorColor, sensorUSonic,;
ジャイロセンサー、超音波センサー、カラーセンサー値格納用変数
char sensval[64];　EV3 本体 LCD 表示テキスト格納用配列変数

LcdInit();　EV3 本体 LCD 初期化
LcdSelectFont(2);　LCD フォントの大きさ設定（1:普通、2:大きい）
LcdRefresh();　LCD 画面クリア
LcdScroll(20);　LCD を上方へ 20 ピクセルスクロール

LcdText(style, x, y, text); LCD に文字を表示
　style:1=普通、2=白黒反転
　x, y:表示する xy 座標
　text:表示内容
LcdText(1, 0, 104, "Start!");　LCD の最下部に"Start!"と表示

initSensor();　センサー初期化

setSensorPort(port, sensor, mode);　センサーの種類と接続ポートを設定
　port:CH_1～4 で接続ポートを指定

　sensor：センサーの種類を指定
　　TOUHC：タッチセンサー
　　GYRO：ジャイロセンサー
　　COLOR：カラーセンサー
　　USONIC：超音波センサー
　　NXT： Mindstorms NXT 用センサー（タッチ、カラー、音声）
　　HTACC：HiTechnic 製加速度センサー
　　（XYZ 軸独立のため、特別なプログラミングが必要）
　　HiTechnic 製の他のセンサー（ジャイロなど）は定義されていませんが、使用可能です

　mode:センサーの動作モードを指定
　　～センサーの動作モード～
　　タッチセンサー：モードなし
　　ジャイロセンサー：0=回転角度、1=角速度
　　カラーセンサー：0=反射率、1=周囲の明るさ、２=色
　　超音波センサー：0=距離（cm)、1=距離（inch)、2=超音波

setSensorPort(CH_1,TOUCH, 0);ポート１をタッチセンサーに
setSensorPort(CH_2,GYRO, 0);ポート２をジャイロセンサーに
setSensorPort(CH_3,COLOR, 0);ポート３をカラーセンサーに
setSensorPort(CH_4,USONIC, 0);ポート４を超音波センサーに

startSensor();　センサー動作開始

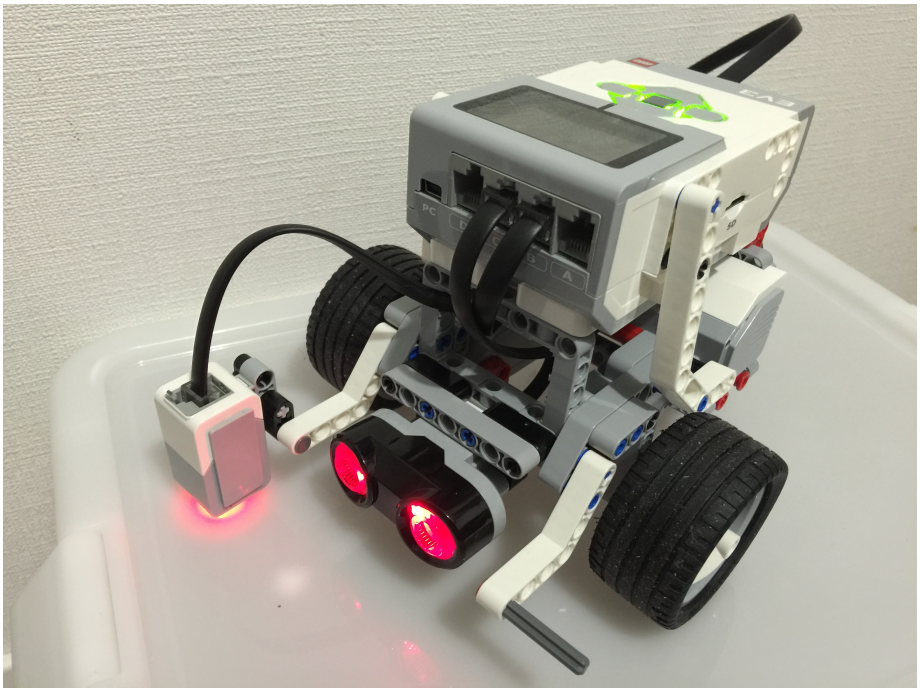
getSensor(port);　センサーの値を読み取る
port:CH_1～４で指定
sensorTouch1 = getSensor(CH_1); ポート１（タッチセンサー）のセンサー値を読み取り、sensorTouch1 変数に格納

closeSensor();　センサー動作終了

（５）超音波センサーによる障害物回避

接続

モーター：ポート A、B
タッチセンサー：ポート１
超音波センサー：ポート４



超音波センサーとカラーセンサーを取り付けた
トレーニングロボット

　タッチセンサー、超音波センサーを取り付けたトレーニングロボットを使用し、障害物の手前で進行方向を変える動きをプログラムする。File メニュー　→　New で新しいプログラムを作成し、Sensor.c の内容をコピーする。次ページのプログラムコードを、下から５行目の if (sensorTouch1) break;の下に挿入する。下から４行目の Wait(500);を削除する。上から１０行目の initSensor();の前に OutputInit();を加える。

```
if (sensorUsonic >300) {
   OnFwdEx(OUT_AB,30,0); //AB Port Motor Start
   Wait(200);   //Wait 1000msec
   Off(OUT_AB); //AB Port Motor Stop
}   else {
   OnRevEx(OUT_B,30,0); //B Port Motor Reverse
   OnFwdEx(OUT_A,30,0); //A Port Motor Start
   Wait(1000);   //Wait 1000msec
   Off(OUT_AB); //AB Port Motor Stop
}
```

ファイル名を Turn.c とし、BricxCC 用フォルダに保存する。
Motor.prj を複製し、ファイル名を Turn.prj に変更する。
Compile および Download and Run で実行する。

if (sensorUsonic >300) 以下により、障害物までの距離が 30cm より大きいければ直進する。else 以下により、障害物までの距離が 30cm 以下の場合はポート A のモーターを正回転、ポート B のモータを逆回転させ、進行方向を変える。

タッチセンサーを押すことで、プログラムは停止する。

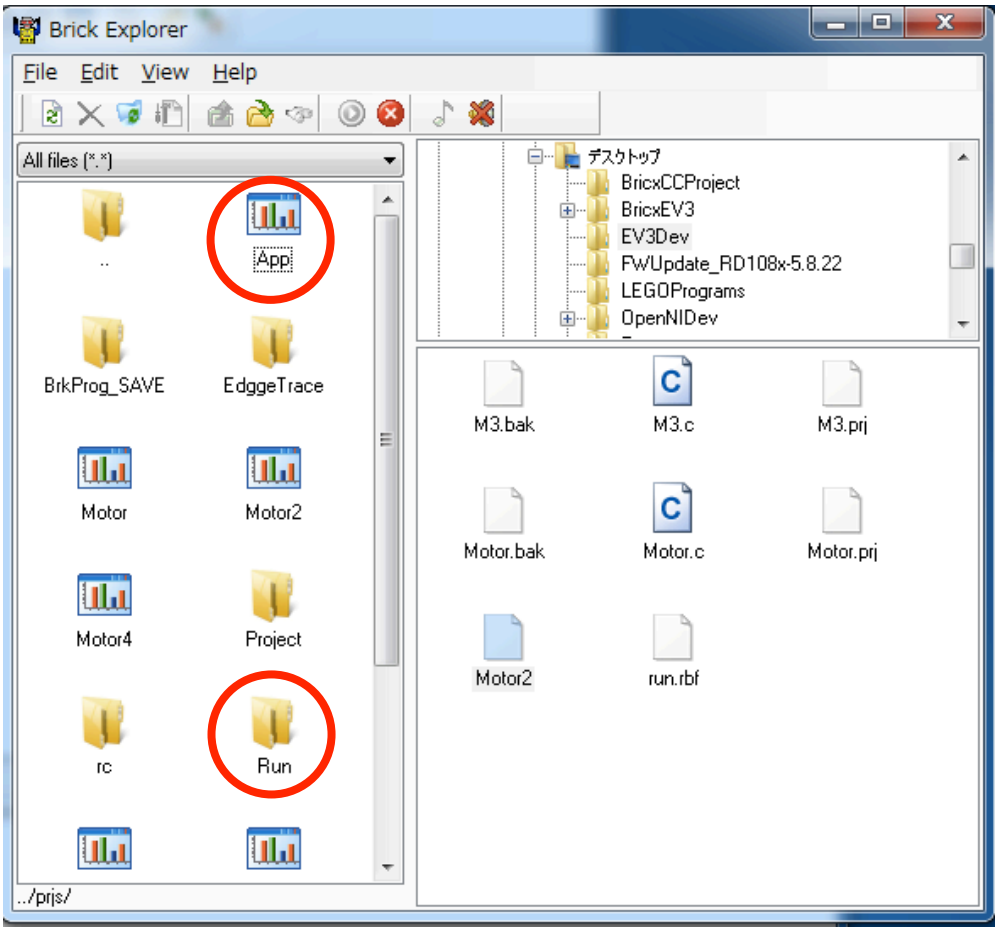
（6）EV3 本体でのプログラム起動

　BricXCC で作成したプログラムは、EV3 本体の LCD に表示されないため、本体でプログラムを選択し、実行することができない。作成したプログラムを本体の LCD メニューから起動するため、ランチャー（プログラム起動のためのプログラム）をインストールする。

　まず、run.rbf ファイルを以下からダウンロードする。

http://www.mns.kyutech.ac.jp/~hanazawa/education/downloads/

BricxCC の Tool メニュー → Explorer で Explorer ウィンドウを表示し、最初に表示される場所（/home/root/lms2012/prjs/App）に、Run フォルダを作成する。作成した Run フォルダの中に run.rbf をコピーする。実行したいプログラムの名前を App に変更する。



EV3 本体 LCD のファイルメニューから Run フォルダ内の run プログラムを選択し、実行すると、App プログラムが起動する。

作成：２０１５年６月
花沢明俊（九州工業大学）