

ETロボコン向けTOPPERS活用セミナー

EV3 APIの解説

2015年6月20日(土)

名古屋大学 石川拓也

EV3RTの提供するEV3用API

- APIを提供するモジュール一覧
 - サーボモータ
 - 各種センサ
 - 超音波, ジャイロ, タッチ, カラー
 - LCD
 - ファイルシステム
 - シリアル送受信機能を含む
 - EV3本体機能
 - バッテリ, ボタン, LED, スピーカ
 - 拡張RTOS機能
 - ユーザ空間での周期ハンドラ

APIリファレンス

- EV3用C言語APIリファレンス
 - http://www.toppers.jp/ev3pf/EV3RT_C_API_Reference/index.html
 - こちらのリファレンスがC++APIに対するベース
- EV3用C++言語APIリファレンス
 - http://www.toppers.jp/ev3pf/EV3RT_CXX_API_Reference/index.html
 - モータやセンサをクラス化

モータ制御APIの例(1/3)

- 関数の引数で使用する列挙型

```
typedef enum {
    EV3_PORT_A = 0, //ポートA
    EV3_PORT_B = 1, //ポートB
    EV3_PORT_C = 2, //ポートC
    EV3_PORT_D = 3, //ポートD
    TNUM_MOTOR_PORT = 4 //モータポートの数
} motor_port_t; //モータポートを表す番号

typedef enum {
    NONE_MOTOR = 0, //モータ未接続
    MEDIUM_MOTOR, //サーボモータM
    LARGE_MOTOR, //サーボモータL
    UNREGULATED_MOTOR, //未調整モータ
    TNUM_MOTOR_TYPE //モータタイプの数
} motor_type_t; //サポートするモータタイプ
```

モータ制御APIの例(2/3)

- クラスMotor (namespace ev3api)
- Motorの公開関数
 - Motor (コンストラクタ)
 - ~Motor (デストラクタ)
 - reset
 - getCount
 - setCount
 - setPWM
 - setBrake

モータ制御APIの例(3/3)

• 使用例

```
/* モータオブジェクトの生成 */  
tailMotor = new Motor(PORT_A);  
/* モータ停止, 回転角度を0初期化 */  
tailMotor->reset();  
/* モータの回転角度を取得 */  
float pwm = (float)(angle - tailMotor->getCount()) *  
             P_GAIN;  
/* モータ回転速度(PWM)を設定 */  
tailMotor->setPWM(pwm);
```

- コンストラクタの引数は, モータを繋いでいるポートに対応するmotor_port_t, ブレーキモードのtrue/false(デフォルトはtrue), モータに対応するmotor_type_t(デフォルトはLARGE_MOTOR)
- getCountはreset直後はreset時からの回転角度を返すが, setCountを呼び出すことで, 回転角度のオフセットを設定可能

センサ取得APIの例(1/3)

- 関数の引数で使用する列挙型

```
typedef enum {
    EV3_PORT_1 = 0,          //ポート1
    EV3_PORT_2 = 1, //ポート2
    EV3_PORT_3 = 2, //ポート3
    EV3_PORT_4 = 3, //ポート4
    TNUM_SENSOR_PORT = 4 //センサポートの数
} sensor_port_t; //センサポートを表す番号

typedef enum {
    NONE_SENSOR = 0,          //センサ未接続
    ULTRASONIC_SENSOR,      //超音波センサ
    GYRO_SENSOR,           //ジャイロセンサ
    TOUCH_SENSOR, //タッチセンサ
    COLOR_SENSOR,         //カラーセンサ
    TNUM_SENSOR_TYPE //センサタイプの数
} sensor_type_t; //サポートするセンサタイプ
```

センサ取得APIの例(2/3)

- クラスGyroSensor (namespace ev3api)
 - クラスSensorの子クラス
- GyroSensorの公開関数
 - GyroSensor (コンストラクタ)
 - setOffset
 - reset
 - getAnglerVelocity

センサ取得APIの例(3/3)

• 使用例

```
/* ジャイロセンサオブジェクトの生成 */  
gyroSensor = new GyroSensor(PORT_4);  
/* ジャイロセンサ初期化 */  
gyroSensor->reset();  
/* ジャイロセンサのオフセット値を設定 */  
gyroSensor->setOffset(OFFSET);  
/* ジャイロセンサ値の取得 */  
gyro = gyroSensor->getAnglerVelocity();
```

- getAnglerVelocityは、ジャイロセンサの現在の値と、setOffsetで指定したオフセット値との差分を返す

ファイルシステムAPIの例(1/2)

- シリアル入出力用の列挙型

```
typedef enum {  
    EV3_SERIAL_DEFAULT // EV3RTコンソール用ポート  
    EV3_SERIAL_UART    // UARTポート (センサポート1)  
    EV3_SERIAL_BT      // Bluetooth仮想シリアルポート  
} serial_port_t; //シリアルポートを表す番号
```

- シリアル入出力用の関数

- ev3_bluetooth_is_connected
- ev3_serial_open_file
- 入出力自体はnewlib APIを使用できる
 - fprintf, fputc, ...
- 他にファイルストリーム入力用のAPIもある

ファイルシステムAPIの例(2/2)

- 使用例

```
/* Bluetooth(SPP)経由でのシリアルI/Oのopen */  
bt = ev3_serial_open_file(EV3_SERIAL_BT)  
/* ホストPCとBluetooth接続が確立されるまで待つ */  
while (!ev3_bluetooth_is_connected()) tslp_tsk(100);  
/* Bluetooth経由でのシリアル受信(1文字) */  
uint8_t c = fgetc(bt);  
/* Bluetooth経由でのシリアル出力 */  
fprintf(bt, "main task started.¥n");
```

拡張RTOS機能APIの例(1/2)

- ユーザ空間の周期ハンドラ生成用静的API
 - EV3_CRE_CYC(周期ハンドラID, { 属性, 引数, 周期ハンドラの関数名, 周期[ms], 初期位相[ms] });
 - 各パラメータの意味はCRE_CYCと同じ
- ユーザ空間の周期ハンドラ制御関数
 - ev3_sta_cyc
 - ev3_stp_cyc

拡張RTOS機能APIの例(2/2)

- 使用例

```
DOMAIN(TDOM_APP) {  
    EV3_CRE_CYC(TEST_EV3_CYC,  
    { TA_NULL, 0, test_ev3_cychdr, 250, 0 });  
}
```

app.cfg

```
void main_task(intptr_t exinf){  
    /* 周期ハンドラの開始 */  
    ev3_sta_cyc(TEST_EV3_CYC);  
}  
void test_ev3_cychdr(intptr_t exinf) {  
    ...  
}
```

app.c

250ms周期で
起動するようになる