Bluetoothデバイスプログラミング教材開発の取り組み

東北ポリテクカレッジ　生産情報システム技術科　**櫻木　伸英**
（東北職業能力開発大学校）

1．はじめに

近年産業界では組込み機器の需要が高まり，これに対応できる技術者の不足が問題化している。組込み機器のソフトウェア製作にはプログラミングの理論だけではなく，対象となるハードウェアの理解やハードウェアに特化したコーディングの知識も必要である。産業界でも組込み系ソフトウェア技術者は他分野のソフトウェア技術者より平均年齢が高く，人材育成が急がれる。

既存のカリキュラムは従来からの学問体系や産業体系により電子系，情報系と縦割りにされており，この範疇で短時間に組込み技術者を育成するのは難しい。

本稿では学生にとって興味を引く身近なデバイスを使って計測･制御のプログラミング実習教材を作成し，新しい技術習得の機会を増やす試みについて報告する。

2．Wiiコントローラの特徴

Wiiは2006年末にNintendoより発売された家庭用ゲーム機であり，本体とコントローラ間の接続を無線化し，コントローラに3軸の加速度センサを搭載するなど今までの家庭用ゲーム機にない特徴を持っている。コントローラの特徴は次のとおりである（表1）。

表1　コントローラの特徴

- 寸法 148mm×36.2mm×30.8mm
- 重量 87g
- 単三アルカリ電池2個
- bluetooth2.0
  （Broadcom社製 BCM2042）
- ボタン12個
- 加速度センサ
  （AnalogDevices社製ADXL330）
- 赤外線CMOSセンサ
- インジケータLED4個
- 小型スピーカ
- 振動機能
- 拡張インターフェース
- 内部EPROM

Wiiのコントローラはこれらの機能を持ったBluetoothHIDデバイスとして単独で利用することもできる[1)]。今回はPCとコントローラをbluetoothで接続し，ボタンと3軸加速度センサを使って計測･制御用のプログラミング教材を作成した。

Wiiコントローラのパネルを上面にして，水平に置いたときの軸は次のようになる（図1）。

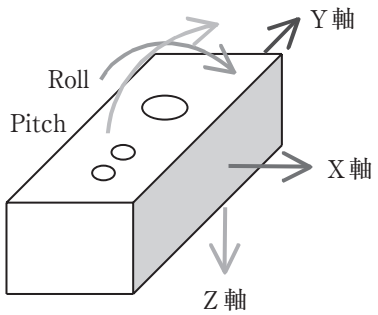

図1　コントローラと3軸

加速度は１軸につきunsigned char型の1byteで得られ，0Gの時80hとなる。今回使用したコントローラで重力（1G）の値およびコントローラを強く振ったとき計測される最大値および最小値は表２のとおりである。

表２　３軸の各計測値

	X軸	Y軸	Z軸	平均
正方向１G	9 eh	a 0 h	9 ch	9 eh
逆方向１G	6 ah	6 bh	6 ah	6 ah
実測最大値	f 6 h	f 6 h	f 4 h	f 5 h
実測最小値	01h	01h	01h	01h

この計測値より各軸の正負の１Gの値から分解能は１/26Gと推測できる。

$$(9\,eh - 6\,ah) / 2 = 1\,ah = (26)_{10} \cdots\cdots (式1)$$

データシート上ではセンサの動作保障は±3Gであるが，実際には4G以上計測できた。

３軸で計測される加速度をそれぞれXg，Yg，Zgとすると，コントローラ静止時のpitch角およびroll角は次の式で求められる。

$$pitch = \arcsin(Xg/1G) \cdots\cdots\cdots (式2)$$
$$roll = \arcsin(Yg/1G) \cdots\cdots\cdots (式3)$$

またコントローラにかかる加速度は次式で求められる。

$$acc = sqrt(Xg^2 + Yg^2 + Zg^2) \cdots\cdots (式4)$$

コントローラ静止時はpitch角，roll角によらず加速度は1Gである。

３．デバイスプログラミング

今回はコントローラから得られるボタンの値および３軸加速度センサの値を取得し，表示するプログラムを作成した。製作環境は次のとおりである（表３）。

コントローラとの通信は，bluezライブラリを使う。また描画はキャラクタベースでcursesライブラリを使った。

表３　主な製作環境

ホストPC	AT-互換機
Bluetoothアダプタ	ELECOM BT-UD1
OS	Linux/Debian4.0r0
kernel	kernel-2.6.18
コンパイラ	gcc-4.1.2
ライブラリ	bluez-2.0, ncurses5

コントローラの接続手順は下記のとおりである[2)]（表４）。

表４　コントローラ接続手順

1. bluetoothアダプタへの接続
2. bluetoothデバイスのサーチ
3. 接続対象デバイスに対してPSM17をcontrol channel，PSM19をinterrupt channelとしてsocket作成
4. connect後にreadおよびwriteで送受信

最初にPCに接続したbluetoothアダプタを取得し，接続する。アダプタが１つとは限らないのと，接続していても使えない場合があることを考慮しなければならない。接続後にbluetoothデバイスのサーチを行い，通信範囲内に応答するデバイス一覧を取得する。ここまで実行した画面が図２である。

図２　取得したデバイス一覧

使用するコントローラを選択後，control channelとinterrupt channelに接続するsocketを２つ作成しconnectする。

control channel はコントローラの制御に使い，interrupt channelはコントローラから加速度センサやボタンの情報取得に使う。

図３はconnect後の実行画面である。

図３　プログラム実行画面

画面上半分はコントローラで検出された加速度を棒グラフで，押されたボタンを反転色で表示する。画面下半分はリアルタイムで取得したコントローラからの出力をraw data のままスクロール表示している。

表示にcursesライブラリを使うことによりプログラムが短くなり，学習の本質を見失わないように工夫した。

コントローラはデフォルト動作としてボタンに変化があった場合，または加速度センサの値に変化があった場合のみデータを送信する。送信サイクルは最大100Hzである。受信データは7byteで1，2byte目が固定，3，4byte目がボタン情報，5，6，7byte目がそれぞれX軸，Y軸，Z軸の加速度の値である。

各ボタンは表５の2byteの値を受信データの3，4byte目にマスクすることにより得ることができる。

例えば受信したデータがunsigned char str[７]に代入されているとすると，str[３]&0x08でAボタンの状態をフラグ判定できる。

２つ目のプログラムは３軸の値について横軸を時間軸としてキャラクタベースでグラフをリアルタイム表示する（図４）。

表５　ボタンのマスク値

button	value		button	value
2	0x0001		←	0x0100
1	0x0002		→	0x0200
B	0x0004		↓	0x0400
A	0x0008		↑	0x0800
—	0x0010		+	0x1000
Home	0x0080		Power	

図４　プログラム実行画面２

このプログラムは描画速度を＋と－のボタンで調整できるようにした。

４．実習への導入

従来は十分な能力を持ったハードウェアとリアルタイムOSを用いてアプリケーションソフトウェアを製作していた場合でも，技術発展により経済的側面から必要最低限のハードウェア，擬似的なリアルタイムOSおよびアプリケーションソフトウェアのすべてを設計・製作する方法も選択できるようになった。

またLinuxもkernel-2.6よりプリエンプションがmsecオーダで実現されており，割込み間隔がシビアではないリアルタイムシステムへの利用が試みられている。

計測・制御関係の実習は当初VxWorks等の完全なリアルタイムシステムに重点を置いていた。しかし機器更新のコスト的な問題もあり，ここ数年の技術進歩に合わせてLinux環境への切換えを進めている。本教材もその１つとなる。

技術要素としては，bluezライブラリやcursesライ

ブラリについての解説が必要である。しかしネットワークインターフェースとしてTCP/IPのsocketプログラミングを事前に学習しているため，同じsocketを利用しているbluezライブラリの概念が短時間で理解できる。またシリアルポートを使った非同期入出力の課題でも，事前にcursesライブラリを使ってコンソール画面の設計や入力および出力の分割表示を行っている。

以上より本教材を利用するに当たっては，ある程度の予備知識のある状態で導入できるため，半日程度の解説とプログラミング実習で実施が可能である。

今回利用したbluezのソースコード[3]は読みやすく，教材としても利用できた。例えば図5のhci_inquiry()関数の実行時にSegmentation faultでプログラムが停止する場合があるという相談を受けた。慣れているエンジニアであればポインタの不正操作が原因で対処方法も理解できるが，学生の場合理由を論理的にわかりやすく説明しなければ理解してもらえない。

hci_inquiry（）関数は通信可能範囲内にいるデバイスの検出を行い，引数にはサーチ時間（len），検出最大デバイス数（nrsp），個々のデバイスの情報を格納する構造体の配列へのポインタ（**ii）などがある。

この関数の実行時に仮引数iiには，あらかじめ確保した構造体inquiry_infoの配列へのポインタを渡しそれを関数内で利用するか，ポインタへのエントリだけ作成し，それをiiに渡してこの関数内で動的に確保するかのどちらかが選択できる。前述の相談は後者の方法をとっており，初期化していないポインタを関数へ渡したためエラーが発生した。仮引数iiへ渡すポインタは明示的にNULLで初期化しなければ，意図したとおりに実行されないことがソースコードから理解できる。

```
int hci_inquiry(int dev_id, int len, int nrsp,
                const uint8_t *lap,
                inquiry_info **ii, long flags)
{
        :
        :

if (!err) {
  int size = sizeof(inquiry_info) * ir->num_rsp;

 if (!*ii)
     *ii = (void *) malloc(size);

     if (*ii) {
       memcpy((void *) *ii, buf + sizeof(*ir), size);
       err = ir->num_rsp;
     } else
       err = -1;
 }

 free(buf);
 return err;
}
```

図5　hci_inquiry（）関数抜粋

上記のような例より，一般的に利用するだけで読むことがないライブラリのソースコードも，技術者育成には良い教材になると考える。

5．まとめ

bluetooth機器は広く使われるようになってきたものの，プログラミングに関するドキュメントはまだ少ない。最新の技術を教材として提供することで，開発課題等への応用が期待できると考えている。

＜参考文献＞

1）Wiimote
http://www.wiili.org/Wiimote

2）An Introduction to Bluetooth programming in GNU/Linux
http://people.csail.mit.edu/albert/bluez-intro/

3）bluez-libs-2.10 Documentation
http://www-md.e-technik.uni-rostock.de/ma/hm27/uebung_hnp_2004/documentation/bluez-libs-2.10-html/