

Wiiリモコンで操作するリモコンカーの製作

— USBデバイスドライバおよびBluetoothプロトコルスタックのH8マイコンへの移植 —

北陸ポリテクカレッジ
(北陸職業能力開発大学校)

人見 功治郎

1. はじめに

Wii リモコンで操作するリモコンカーを製作した。リモコンカーはUSBコントローラ内蔵のH8マイコンボードを搭載し、USB型Bluetoothアダプタを通して、Wiiリモコンから制御できる。Wiiリモコンで操作するために、H8マイコン用 μ ITRON 環境にLinuxカーネルのUSB デバイスドライバおよびBluetoothプロトコルスタックを移植した。以下、詳細について報告する。

2. Wiiリモコン

2.1 Wii

Wiiとは任天堂より2006年12月に発売された家庭用ゲーム機である。Wiiの特徴は、従来家庭用ゲーム機で採用されていたコントローラとは違い、3次元加速度センサを搭載した無線リモコン（「Wiiリモコン」と称する）を用いる点であり、これにより身振りなどのジェスチャでゲームを操作できる。Wiiリモコンは、Bluetoothによる無線通信でWii本体と接続する。Bluetooth自体は汎用的な規格のため、Wiiリモコンをパソコンと接続し、マウスの代わりとして利用することも可能である。

2.2 Bluetooth

Bluetooth規格は携帯電話・PHS、スマートフォン・PDAやPC等の携帯端末において無線通信を行う

ために策定された¹⁾。Bluetooth無線技術は、携帯デバイスや固定デバイスのケーブル接続技術の代わりとして機能する、近距離の通信技術である。Bluetooth仕様には、幅広い範囲のデバイスが相互に接続および通信できるように、統一構造が定義されている。図1にBluetoothの統一レイヤー構造を示す。WiiリモコンはBluetoothコントローラとBluetoothホストが一体となった機器であり、HID (Human Interface Device) プロファイルを備えた機器である²⁾。

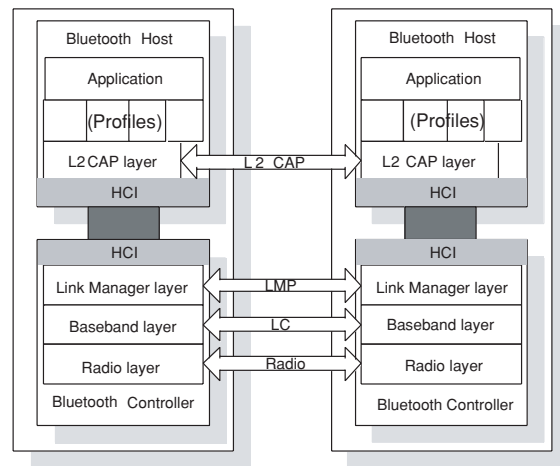


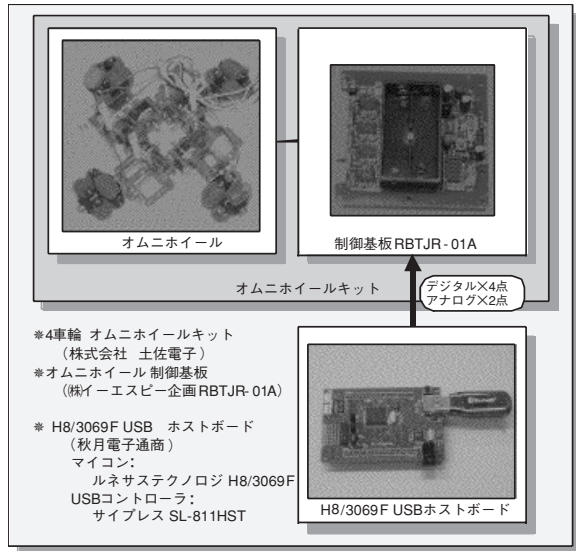
図1 Bluetoothのレイヤー構造

3. リモコンカー

3.1 ハードウェア構成

今回作成したリモコンカーは株式会社土佐電子の4車輪オムニホイールキットに秋月電子通商のH8/3069F USBホストボード（「H8ボード」と称する）を

接続したものである。これにUSBタイプのBluetoothアダプタを接続し、Wiiリモコンから操作できるようにする。ハードウェア構成を図2に示す。



オムニホイールキットには制御基板として(株)イーエスピー企画の制御基板RBTJR-01A(「Cロボ」と称する)が付属している。本基板には6本の入出力用インターフェースが備わっており、アナログあるいはデジタル信号を外部に入出力可能である。また、Cロボにはアナログ・デバイゼ社のマイクロコントローラADuC814ARUが搭載され、6本の入出力線は本コントローラに直結されている。ADuC814ARUにはプログラム用として8Kバイトのオンチップフラッシュメモリが搭載されており、ここにプログラムを焼き付けることにより、センサ入力に応じたオムニホイール制御が可能である。

H8ボードには2MBのDRAMとサイプレス社USBコントローラSL811HSTが搭載されている。SL811HST自身はUSBホスト/ターゲットのどちらとしても利用できるが、本ボードではホストモードでのみの利用しかできない。H8/3069Fにはアナログ出力が2チャンネルしか用意されていないため、これら両基板を表1のように接続して使用する。

またリモコンカーを電池で稼働させるため、H8ボードの電源として、9[V]乾電池を用いた三端子レギュレータ電源回路を自作した。

表1 Cロボ, H8ボード間の入出力仕様

Cロボ (ADuC814ARU)	方向	H8ボード (H8/3069F)	種別
P1.2	←	DA0	アナログ
P1.3	←	DA1	アナログ
P1.4	←	P40	デジタル
P1.5	←	P41	デジタル
P1.6	←	P42	デジタル
P1.7	←	P43	デジタル

3.2 ソフトウェア構成

今回制作したソフトウェアは、Cロボ上のオムニホイール制御プログラムとH8ボード上の制御プログラムの2つである。今回、表1の信号のうち、4つのデジタル信号を車輪の回転方向指示に、2つのアナログ信号を車輪の回転速度に使用する。アナログ信号が2つであるため、対角線上に配置された車輪を同じ速度で回転させる。

おのおののプログラム開発には、プラットフォームとしてFedora Core Linux 7を使用し、共にC言語でクロス開発を行った。開発用ツールチェーンを表2に示す。

表2 開発ツールチェーン

基板 (CPU)	ツールチェーン
Cロボ (ADuC814ARU)	sdcc-2.7.0
H8ボード (H8/3069F)	binutils-2.16.0
	gcc-3.4.6
	newlib-1.13.0

3.2.1 Cロボ用プログラム

Cロボ用プログラムは、H8ボードからの指示に従って4つの車輪を制御する。オムニホイールに使用されているマブチモータ株式会社 RE-260はDCモータであるため、PWM制御によりモータ速度を制御する。今回は1.45[ms]の周期でデューティ比0[%]~95[%]の範囲で制御する。本プログラムはOSを用いず、タイマー割り込みルーチンとメインルーチンとの単純な構成とした。

3.2.2 H8ボード用プログラム

H8ボード用プログラムは、Wiiリモコンとの接

続・切断処理と、Wiiリモコンの状態に応じたオムニホイール制御を行う。本プログラムには μ ITRON 4.0準拠リアルタイムカーネルであるTOPPERS/JSP 1.4.3を利用した。なお、今回のH8ボードはTOPPERSプロジェクトではサポートされていない。しかし、サポートされている秋月電子通商製のその他ターゲットボードとCPUクロック値が異なるだけであり、この点を修正し、利用した。またTOPPERSのドキュメントで言及されている環境に比べ、今回使用したツールチェインはバージョンが新しい。これはLinuxのソースファイルを移植する際、アセンブルできない擬似命令が使用されていたためである。

H8ボード用プログラムの状態遷移図を図3に示す。接続待ち状態でWiiリモコンの①②ボタンを同時に押すと接続する。接続が完了すると、十字カーソルモードでは十字カーソルで、加速度センサ操作モードではWiiリモコンの傾きに応じてオムニホイールを移動させる。また、どちらのモードでも①あるいは②ボタンを押すことでその場で回転する。

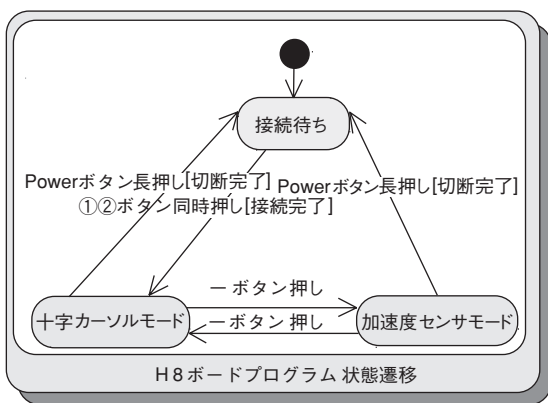


図3 H8ボード用プログラム状態遷移図

4. Wiiリモコン制御プログラム

H8ボード用プログラムの詳細を述べる。

今回使用するサイプレス社SL811HS USBコントローラにはLinux用ドライバが提供されている。このため、その上位層を構成するプログラムについてもLinux環境のものを利用する。実はLinux 2.6系列のカーネルは、H8/300HシリーズCPUを一部サポートしているため、すべてをLinuxで構築する選択肢もある。

しかし、最終的なプログラムサイズがH8/3069FのFlash ROMに収まらないと予想されたので、 μ ITRON環境にLinuxデバイスドライバを移植する構成とした。

Linuxのデバイスドライバはレイヤー構造となっている。プログラムの開発に当たっては、これらレイヤー全層を一気に移植できないため、USBコントローラに近いレイヤーから移植した。その際、そのつど適当なアプリケーションを作成し、動作を確認しながら行った。移植に際し、注意した点は次のとおりである。

① コンパイル段階

プログラムに対する変更は、`#if`あるいは`#ifdef`ディレクティブを使用して行い、極力元に戻せる配慮をしながら行う。

プログラム内で使用しているヘッダファイルや記号定数を解決するため、必要なものをカーネルソースツリーあるいは開発システムのヘッダファイル群からコピーして使用する。ヘッダファイルは入れ子構造になっているものが多いので、必要なヘッダファイルはすべてコピーする。またCPU依存のヘッダファイルについては2.6系列の`include/asm-h8300`ディレクトリのものを利用する。

Linuxではカーネルの特定の機能を有効あるいは無効にしたい場合、対応する記号定数を定義あるいは未定義にすることによって切り替える。これは通常、カーネル作成の際に行うコンフィギュレーション作業を通して生成する。そこで今回は、

```
make ARCH=h8300 menuconfig
```

により設定ファイルを生成し、これを記号定数の形式に変更して、使用する。使用に際しては、これら定義のほか、全体にかかわる設定をヘッダファイルに記述し、すべてのソースファイルからインクルードするようにした。

② リンク段階

未定義シンボルを解決するため、必要なソースファイルをソースツリーに組み入れる。しかしやみくもにコピーするとLinux全体をコピーすることになる

ため、どうしても必要な関数呼び出しと判断されるもの以外をコメントアウトする。さらに、必要なものについてもコピーは最小限にとどめ、コピーしないものについては自作や他の関数への置き換えで対応する。

例えば、malloc関数については、newlibのものを利用する。このためリンク時に `-lc` オプションを渡し、インストールしたnewlibを利用するように指示する。ただし、newlibのmallocをうまく動作させるためには、

- `_sbrk`または`_sbrk_r`
- `__malloc_lock`
- `__malloc_unlock`

といった関数を自作する必要がある。このため、TOPPERSに付属する`systask/newlibrt.c`を参考にして、上記関数をリンクする。

`printk`や`kmalloc`といった関数については、今回リンク時に解決させるのではなく、ソースファイルあるいはソースファイルからインクルードするファイルに次のような`#define`文を定義して問題を回避した。

```
#define printk (arg...) ¥ [次行へ]
        syslog ( LOG_NOTICE, ## arg)
#define kmalloc ( a , b ) malloc (( a ))
#define kfree ( a ) free (( a ))
```

③ デバック段階

確認用アプリケーションを作成できたら、実際にボード上でテストする。これにはルネサステクノロジ社が無償で提供しているモニタプログラム³⁾を利用する。モニタを利用してアプリケーションをDRAM領域に転送し、実際にDRAM領域上で実行させてデバックを行う。

4.1 USB制御プログラム

今回使用したUSBドライバはサイプレス社のWebから無償で利用できるSL811HS Linux Host Driverリリース1.0 (「SL811ドライバ」と称する)である。本ドライバはAccelent社IDPボードをターゲットに、Linux 2.4.4カーネル用として作成されている。しか

し、後述するBluetoothドライバはLinux 2.4.18カーネルのものを使用しているため、今回カーネルのソースコードとして、Linux 2.4.18を採用した。また、SL811ドライバで提供されている仮想ハブ機能は省略した。

LinuxにおけるUSBドライバの概要を図4に示す。

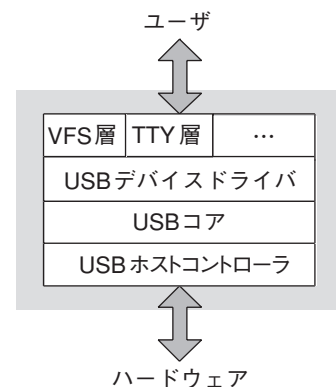


図4 USBドライバの概要

図4中のUSBデバイスドライバが、URB (USB Request block :USB要求ブロック)をUSBコアに送信し、USB機器との通信を開始する。USBコアはURBを処理する際、USBホストコントローラに対し次の関数ポインタに登録された関数を呼び出す。

```
struct usb_operations {
    int (*allocate) (struct usb_device*);
    int (*deallocate) (struct usb_device*);
    int (*get_frame_number)
        (struct usb_device *usb_dev);
    int (*submit_urb) (struct urb* purb);
    int (*unlink_urb) (struct urb* purb);
};
```

図5 USB操作関数群

このためUSBホストコントローラに相当するSL811ドライバでは、ハードウェアの初期化といった機能に加え、上記関数ポインタに登録される関数を提供している。また、USB機器からのデータ転送完了は割り込みで伝えられるため、SL811ドライバでは割り込みハンドラも提供され、URBに登録されているコールバック関数は割り込みハンドラ内から呼び出される。

例えばUSBデバイスの記述子情報を読み出すには、コントロール転送⁴⁾を使用してGET_DESCRIPTORリクエストを発行すればよい。これを実現するには、USBコアのusb_get_descriptor関数を使用するが、この関数から順に次のような関数が呼び出される。

```
usb.c::usb_get_descriptor
|usb.c::usb_control_msg
||usb.c::usb_internal_control_msg
|||usb.c::usb_start_wait_urb
||||usb.c::usb_submit_urb
|||||hc_simple::hci_submit_urb
||||||hc_simple::hcs_urb_queue
|||||||hc_simple::qu_queue_urb
||||||||hc_simple::qu_queue_active_urb
|||||||||hc_simple::hc_start_int
|||||semaphore.c::wai_sem
```

図6 GET_DESCRIPTORの処理

図6の手順中usb_submit_urb関数内で関数ポインタsubmit_urbに登録された関数(hci_submit_urb)が呼び出される。そして、実際の転送作業はhc_start_int関数で割り込みを許可した後が始まる。実際の転送は時間がかかるため、μITRONのセマフォを用いて転送が終了するまで、タスクを停止させる。タスクの再開は、割り込みを通して転送が終わったのち、URBに登録されたコールバック関数内からisig_sem関数を用いて行う。

4.2 Bluetooth制御プログラム

今回のH8ボードでは図1にあるHCI間がUSBでの接続となる。また、WiiリモコンはHIDとして動作するように作られているため、ドライバ全体のレイヤー構造は図7のようになる。

今回使用したBluetoothプロトコルスタックはBluez 2.25である。これにはLinux 2.4.18に対するパッチファイルとユーティリティルーチンが含まれている。これらのうち必要と思われる部分をUSBドライバ同様、移植する。なお、Linuxにおける通信アプリケーションはソケットインターフェースを利用して作成するのが一般的である。これに倣いL2CAPドライバもソケットインターフェースを提供している。

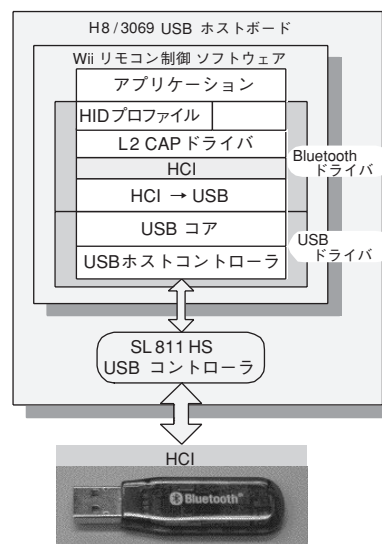


図7 Wiiリモコン制御プログラムのレイヤー

このためL2CAPドライバを移植する際にはLinuxのソケット関連プログラムも合わせて移植する。

4.3 Wiiリモコン制御プログラム

Wiiリモコン制御プログラムは、USBドライバBluetoothプロトコルスタックに加え、Wiiリモコン用ライブラリlibwiimote-0.4⁵⁾を移植したうえで、これらを利用するアプリケーションとして作成した。

以上Wiiリモコン制御プログラムを作成するに当たり、Linuxカーネル、サイプレス社ドライバ、Bluez、libwiimote等より利用したファイルの一部を図8に示す。

5. 動作状況

完成したプログラムサイズは165kバイトであり、H8のFlash ROMに焼き付けることができた。Wiiリモコンの接続から、十字カーソルを使った操作、加速度センサを使った操作、共に良好に動作している。また、ポリテクビジョンの催物として2台の機体を展示し、半日程度来場者に実際に操作してもらった。ねじのゆるみに伴う機体の故障は発生したが、ソフトウェアに起因する目だったトラブルは発生しなかった。

- USBドライバ (サイプレス社ドライバより, ホストコントローラ)
drivers/usb/hc_simple.c
drivers/usb/hc_sl811.c
- USBドライバ (Linuxカーネルより, USBコア)
drivers/usb/usb-debug.c
drivers/usb/usb.c
- Bluetoothプロトコルスタック (BlueZ)
drivers/bluetooth/hci_usb.c
net/bluetooth/hci_core.c
net/bluetooth/hci_sock.c
net/bluetooth/hci_event.c
net/bluetooth/l2cap.c
net/bluetooth/hci_conn.c
net/bluetooth/lib.c
net/bluetooth/af_bluetooth.c
- ソケット関連 (Linuxカーネル)
net/socket.c
net/core/iovec.c
net/core/datagram.c
net/core/sock.c
net/core/scm.c
net/core/skbuff.c
- Wiiリモコン制御 (libwiimote)
net/wiimote/wiimote_io.c
net/wiimote/wiimote_link.c

図8 移植したファイル (一部)

6. おわりに

WiiリモコンはBluetoothを利用したHIDデバイスという位置づけであるため、パソコンで利用すること

も可能であり、実績も多く公開されている。しかし、H8マイコンなどを使用した組み込みシステムの操作デバイスとして使用された実績は見当たらない。そこで、今回Linuxなどさまざまなオープンソースのプログラムを活用し、Wiiリモコンで動作するリモコンカーを製作し、良好に動作することが確認できた。

また、筆者が指導した北陸職業能力開発大学校平成19年度開発課題「付加機能をもった電動車いすの開発」では、付加機能としてWiiリモコンでの電動車いす操作を実装した。本報告における開発はこの車椅子の開発に先立って行ったものであり、この成果をフィードバックする形で学生の開発を指導した。さらに今回のリモコンカー製作に当たっては、USBプログラミングなどさまざまなプログラミング要素が含まれているため、プログラムの一部を抜き出すことでさまざまな教材として応用可能であると考え

<参考文献>

- 1) <http://japanese.bluetooth.com/bluetooth/>
- 2) <http://www.wiili.org/index.php/Wiimote>
- 3) http://japan.renesas.com/media/support/seminar/sample_program/seminar_sample_downh83h/300hhew.exe
- 4) トランジスタ技術編集部編：「オリジナルUSB機器の設計と製作」, CQ出版株式会社, 2005年
- 5) http://download.cypress.com.edgesuite.net/design_resources/software_and_drivers/contents/sl811hs_linux_host_driver_v1_0_13.zip

新刊案内

好評発売中!



地域の未来は
こうして拓く!

執筆者

伊藤 実
金 明中
清水希容子
永久寿夫
西澤正樹

地域における雇用創造

～未来を拓く地域再生のための処方箋～

地方自治体による人材育成、資源活用、情報発信への支援、連携
——地域が主体的に推進する活性化策の具体例を紹介

● A5判 / 292ページ / ソフトカバー

● 2008年6月発行

● 定価2,000円(本体1,905円)

ISBN978-4-87563-245-0

発行 ● 財団法人 雇用開発センター

発売 ● 社団法人 雇用問題研究会

■ご購入・お申し込みはこちら **社団法人 雇用問題研究会** <http://www.koyoerc.or.jp>

〒104-0033 東京都中央区新川1-16-14 電話 03-3523-5181 (代表) FAX 03-3523-5187