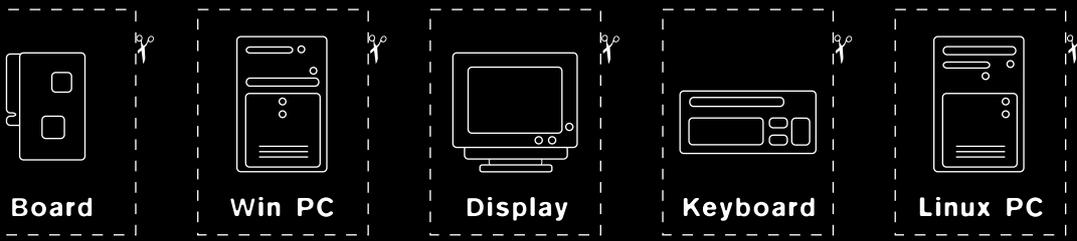




Special Linuxハードウェア工作教室

# USB-IOキットを使う



有限会社ハンブルソフト  
成松宏

## はじめに

パソコンの周辺機器を自作するとき、パソコンとどのように接続するかは大きな問題です。シリアルポートやプリンタポートはお手軽ですが、「レガシーデバイス」といわれ、いつまで使えるか分かりません。代わりに出現したUSBは、便利な周辺装置がどんどん増えてきました。

USBには、シリアルポートやプリンタポートと比べて次のような利点があります。

- ・USBから電源(0.5A)取得することができる
- ・ハブを使用することで最大127台まで接続可能(ハブも含む)
- ・転送速度が速い

そしてUSBの部品も、秋月電子(記事末RESOURCE[1]を参照)などで入手できるようになってきましたし、ここで1度、USBに挑戦してみようというのが本記事の趣旨です。今回は、モルフィー企画[2]の「USB-IOキット」を使用して、USBで接続する周辺機器を作ってみます。

## モルフィー企画のUSB-IO

モルフィー企画のUSB-IOは、Cypressの「CY7C37001A」というUSBコントローラを使って単純なデジタルI/Oを作るキットです(写真1、図1)。USB-IOは、1500円(送料込み)で頒布されています。申込方法はモルフィー企画のWebページ[3]をご覧ください。

図1の回路図を見ると分かるように、USB-IOは、ほとんど

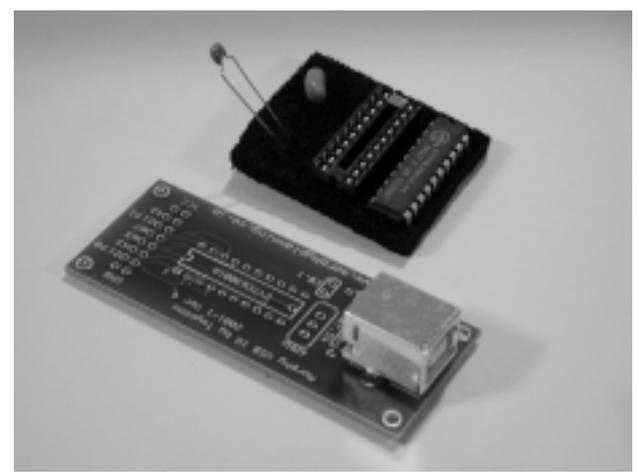
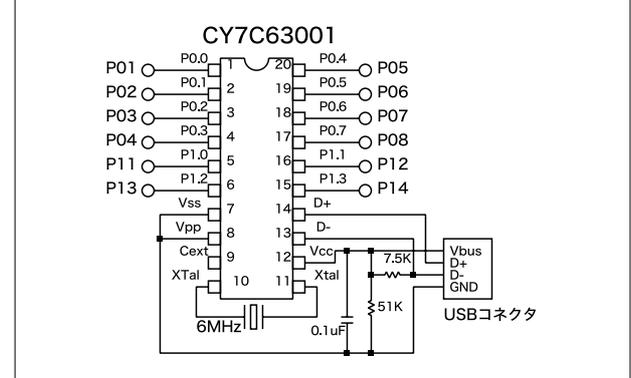


写真1 モルフィー企画のUSB-IOキット

図1 USB-IOの回路図



CypressのCY7C37001AというICだけでできています。CY7C37001Aは、USBインターフェイスを内蔵したマイコンで、プログラムを内蔵のEEPROMに書き込むことができま

リスト1 USB-I/Oを接続したときの/var/log/messages(ドライバがない場合)

```
Apr 13 21:39:29 hslpc2 kernel: usb.c: USB new device connect, assigned device number 3
Apr 13 21:39:29 hslpc2 kernel: Manufacturer: Morphy Planning
Apr 13 21:39:29 hslpc2 kernel: Product: generic USB IO ver.2
Apr 13 21:39:29 hslpc2 kernel: usb.c: This device is not recognized by any installed USB driver.
Apr 13 21:39:29 hslpc2 kernel: Length = 18
Apr 13 21:39:29 hslpc2 kernel: DescriptorType = 01
Apr 13 21:39:29 hslpc2 kernel: USB version = 1.10
Apr 13 21:39:29 hslpc2 kernel: Vendor:Product = 0bfe:1003
```

す。このEEPROMへのプログラム書き込み機は、およそ1万円ほどで購入可能です。プログラム次第で、USBマウスやキーボードのような低速USBデバイスを作ることができます。またUSB-I/Oでは、デジタル入出力を行うプログラムを書き込み済のCY7C37001Aが付いてきますし、立派なプリント基板もついてきますので、半田づけさえできれば、誰でも簡単に製作できます。

USB-I/Oが完成したら動作の確認です。Windows環境であれば、モルフィー企画が公開しているVisual Basicのプログラム([4])を使って動作の確認ができます。

USBをサポートするLinuxのシステムにUSB-I/Oを接続すると、/var/log/messageにデバイスドライバがないというメッセージが出力されます(リスト1)。これでUSB-I/Oが認識されていることや、ベンダコードが「0bfe」でプロダクトコードが「1003」であることが分かります。

しかし、ドライバがないのでは何もできないので、以下では、ドライバを作成することにします。

## デバイスドライバの作成

デバイスドライバの作り方については「LINUXデバイスドライバ([5])」という本をお勧めします。私は、この本のせいでFreeBSDからLinuxに転びました(笑)。LinuxのUSBドライバの作り方については、英文ですが「Programming Guide for Linux USB Device Drivers([6])」が参考になります。このドキュメントに必要な情報はすべて載っていました。USB自体については、これも英文ですが、「USB COMPLETE([7])」がまとまっていて便利です。私は、ドライバ製作途中の「はまったとき」に大体読んでしまいました(笑)。分かってしまえば、(当然ですが)USBのドライバ作成は簡単でした。

ドライバの仕様を決めるに当たっては、以下に挙げる点を考慮しました。

表1 ioctlによるモード設定

設定項目	設定内容
入力/出力モード	バイナリ、2進テキスト(デフォルト)、16進テキスト
入出力先ポート設定	ポート0、ポート1、ポート0/1連続(12bit出力)

- ・複数のUSB-I/Oにも対応したい
- ・ポート0とポート1に別のデバイス(マイナー番号)を割り当て、書き込みと読み出しをできるようにするのが素直であろう
- ・テキストモードでのI/Oをサポートすればシェルからも簡単に使用できて便利
- ・USB機器はメジャー番号が固定され、マイナー番号も16個しか使用できない

テキストモードでのI/Oについてちょっと説明します。

USB-I/Oのポート0を/dev/usbio0に、ポート1を/dev/usbio1にそれぞれ割り当てたとします。ポート0は8bitですから、ポート0の1回の読み書きを/dev/usbio0/の1byteのデータの読み書きで行うことができます。例えば、ポート0の8つの端子の値がすべて「1」であれば、/dev/usbio0から8つのbitの値がすべて「1」である値、つまり16進数でいう「0xFF」という値が読み出されます。これは、プログラムで操作する際には使いやすいのですが、コマンドラインやシェルスクリプトからコントロールするには面倒です。

そこでUSB-I/Oへの読み書きを文字列に変換して行うモードを用意します。先ほどの例の値を2進数の文字列に変換すると「11111111」となります。これに、区切り文字として改行(文字コードは0x0A)を付加したデータを出力することにします。このようにすると、1回/dev/usbio1から9bytes読み出す必要がありますが、catコマンドやheadコマンドで状態を読み出すことができます。特に、「head -1」とすると、1回だけデータを読み出すことができ便利です。

## ドライバの仕様

実際のデバイスドライバの仕様は、以下のようになります。

- (1)ベンダコード「0x0BFE」、プロダクトコード「0x1000~0x10ff」のデバイスをUSB-I/Oと見なす
- (2)1つのデバイス当たり、2つのマイナー番号を使用し、ポート0とポート1に割り当てる。最大で8台使用可能(デフォルトでメジャー番号180、マイナー番号32~47を使用)
- (3)read、write、ioctlをサポート(表1)

## 実行例1 Linux版USBドライバのmake

```
$ tar xovfz musbio-1.00.tar.gz
$ cd musbio-1.00
$ make
cc -D__KERNEL__ -DMODULE -DDEBUG -O -Wall -c \
    morphy-usbio.c -o morphy-usbio.o
cc -o usbioc -O -Wall usbioc.c
```

## 実行例2

```
$ su
Password:
# make devices
/bin/mknod /dev/usbio c 180 32
/bin/mknod /dev/usbio0 c 180 32
/bin/mknod /dev/usbio1 c 180 33
/bin/mknod /dev/usbio2 c 180 34
/bin/mknod /dev/usbio3 c 180 35
```

- (4) `ioctl`でモードを設定する。モードは、オープン中のファイルおよびデフォルトの両方を設定できる。モードの設定は `ioctl`で行う。モード設定コマンド「`usbioc`」を同梱
- (5) その他の `ioctl`として、「USB-IO version {プロダクトID}」の取得、パルス出力コマンドに対応

## ドライバーのインストール

まず、USBをサポートするLinuxシステムを準備します。私は、カーネル2.2.18にUSBバックポートパッチを当ててカーネルを再構築しました。最近のディストリビューションであれば、最初からUSBをサポートしているものも多いと思いますので、それらを使用するのも簡単でしょう。カーネル2.0を同2.2に変更して使用する場合には、カーネル以外に `modutils` などもバージョンアップする必要があるらしいので注意してください。USBバックポートパッチや、カーネルそのものについての情報は、Linux USB ProjectのWebページ[8]をご覧ください。

USB-IOのLinuxドライバは、筆者のWebページで公開しています[9] (本誌付録CD-ROMにも収録)。アーカイブを展開後、`make`すれば、`morphy-usbio.o`というファイルができます(実行例1)。マイナー番号を変更するには、`morphy-usbio.h`というファイルにある「`MUSBIO_MINOR`」の値を変更してください。なおこの値は16の倍数でなければいけません。

次に、「`make devices`」で、`/dev/usbio`などのデバイスファイルを作っておきます(実行例2)。

## 実行例3

```
# insmod morphy-usbio.o
# lsmod
Module                Size  Used by
morphy-usbio          3056   0 (unused)
.....
# tail /var/log/messages
.....
kernel: morphy-usbio.c: morphy-usbio: init_module
kernel: usb.c: registered new driver morphy-usbio
```

## ドライバの使い方

このUSBデバイスドライバを使用するには、`insmod`コマンドでモジュールを組み込む必要があります(実行例3)。デバイスドライバが正常に組み込まれると、`lsmod`コマンドで確認できますし、`/var/log/messages`にメッセージが出力されます。

では、さっそくUSB-IOをPCに接続してみましょ。システムログ(`/var/log/messages`)にはリスト2のようなメッセージが出力されているはず。この状態でUSB-IOに対する入出力を実行することができます。デフォルトでは、バイナリテキストモードになっていますので、`echo`コマンドで出力、`head`コマンドで読み出しを行うことができます。

以下の例では、ポート0の1番目、3番目、5番目、7番目と、ポート1の1番目、4番目の各bitに1を、他のbitに0を出力しています。

```
$ echo 01010101 > /dev/usbio0
$ echo 1001 > /dev/usbio1
```

また次の例では、`head`コマンドでポートの値を読み出しています。値を読み出すと、ポートは入力モードになり、出力された値はクリアされてしまいますので注意が必要です。

```
$ head -1 /dev/usbio0
10001111
$ head -1 /dev/usbio1
1111
```

```
kernel: usb.c: USB new device connect, assigned device number 3
kernel: Manufacturer: Morphy Planning
kernel: Product: generic USB IO ver.2
kernel: morphy-usbio.c: musbio_probe(0bfe:1003) ifnum=0
kernel: morphy-usbio.c: morphy-usbio found.
kernel: morphy-usbio.c: dev 0 assigned.
```

## 実行例4 usbiocコマンド

```

$ ./usbioc
Usage:usbioc /dev/usbio[0-4] cmd
cmd:
  bin          set binary mode
  bintext      set bin text mode
  hextext      set hex text mode
$ ./usbioc /dev/usbio hextext
$ head -1 /dev/usbio0
008f

```

ポートのモードを切り替えるために、usbiocというコマンドがついています。オプションを付けずに起動すると、このコマンドの簡単な説明が表示されます(実行例4)。

## USB-AC100Vアダプタ

このUSB-IOを秋月電子のSSR(Solid State Relay)キットと組み合わせ、USBからAC 100Vのオン/オフをコントロールする「USB-AC100Vアダプタ」を作ってみましょう(写真2)。

図2に回路図を示します。USB-IOとSSRキットを組み合わせただけですので、とても簡単です。AC 100Vを使用するので、気休めでヒューズも入れてあります。SSRキット付属の説明書によると、トライアックには12Aまで流せるそうですが、電圧降下が1.5Vあるので、12A時には $12A \times 1.5V = 18W$ となり、かなりの発熱量になります。今回はトライアックをアルミのケースに固定し(写真3)、放熱できるようにしました。



写真2 USB-AC100Vアダプタ

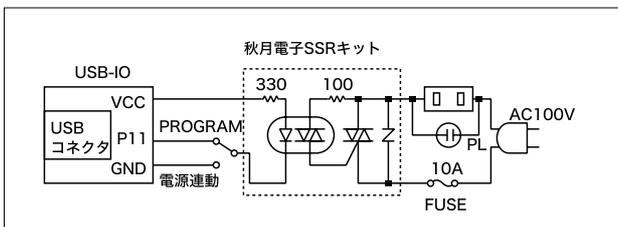


図2 USB-AC100Vアダプタ回路図

## 部品の入手と工作

ここで使用する部品は入手性のいいものを選んでいまして、地方でも簡単に手に入ると思います。部品表を表2に示します。

ケースは、放熱も考えて、タカチの「YM-130」というアルミのものを使用しました。このシリーズは安価で見ためも良いし、熊本でもいろんなサイズのもので手に入るのに気に入っています。ケース加工は、ドリルとハンドリーマー、ハンドニブラー、ヤスリでグリグリと行いました。

AC 100Vとの接続には「シャーシコネクタ」というものを使いました。これはパソコンなどのAC 100Vのケーブルに合うコネクタで、ケース加工は多少面倒になるので、ACプラグ付きのケーブルを使うのも良いと思います。私の場合は、何しろパソコンのケーブルはたくさん溢れていますし、使用しないときの収納が便利なのでよく使っています。

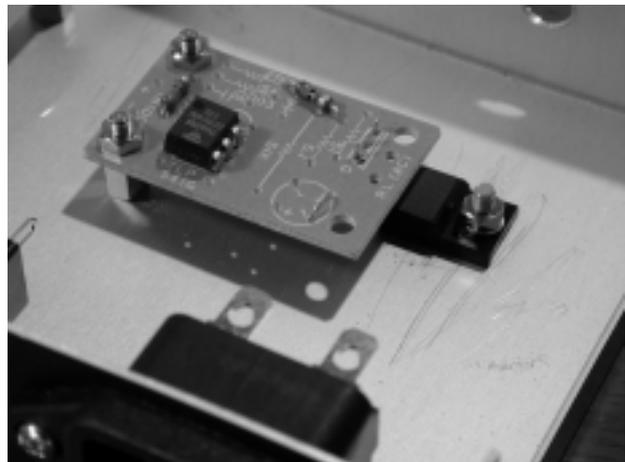


写真3 トライアックの取り付け方

表2 USB-AC100Vアダプタの部品表

品名・型名・仕様	数量	参考単価(円)	購入先
USB-IOキット	1	1500円	モルフィー企画
SSRキット	1	300円	秋月電子
ケース(YM-130)	1	570円	サトー電気など
スライドスイッチ(小3P)	1	40円	サトー電気など
ネオンランプ(100V用R入り)	1	240円	サトー電気など
ヒューズケース(リード付き)	1	80円	サトー電気など
シャーシプラグ	1	250円	サトー電気など
AC100Vソケット	1	100円	サトー電気など

サトー電気 <http://www2.cyberoz.net/city/satodenk/>

## 使ってみる

今回はAC 100Vを使っているので、配線ミスがあると即PC破壊になってしまいます。実験をする前には、テスターなどを使ってAC 100V関係の配線とそれ以外が完全に絶縁されていることを十分に確認してください。

確認が済んだらAC 100Vを接続し、いよいよUSBケーブルでPCに接続します。この状態でスライドスイッチを電源連動にするとネオンランプが付き、PROGRAM側にするとネオンランプが消えるはず。AC 100Vのソケットに電気スタンドなどをつなげば、点滅できるはず。

確認ができたなら、スライドスイッチをPROGRAM側にセットし、Linuxにusbio.oを組み込んで、echoコマンドでAC 100Vのオン/オフができるかどうか確かめます。SSRはUSB-IOのポート1の0ビットに接続されていますので、以下のコマンドを実行します。

```
$ echo 0 > /dev/usbio1 # 電源オン
$ echo 1 > /dev/usbio1 # 電源オフ
```

これで完成です(写真4)。

## 応用

このUSBIO-AC100Vの使用方法はいろいろ考えられます。例えば、スピーカーやプリンタ、モニタを使用時だけオンにすることができます。勇気があれば?、他のパソコンの電源もコントロールできるかもしれません。そういうまじめな使い方以外にも、いろいろふざけた用途もあるかもしれません。メールが来ると電灯が付くとか、テキストファイルを入力するとモールス符号に変換し電灯を点滅させるとか.....(電



写真4 AC 100Vに接続し実験中

球がすぐ切れてしまいそうだなあ)。

## 実はPC連動コンセント

実はこのUSB-AC100Vアダプタは、トランジスタ技術の記事([10])を読み、PC連動コンセントが欲しくなって作りました。

日ごろはスイッチを電源連動にしておいて、PC連動コンセントとして使い、たまにふざけた使い方で人を驚かす、というあたりが楽しいかもしれません。

## 今後

今回は、USB-IOを利用してお手軽にUSB機器を製作してみました。Linuxのドライバは思ったよりも簡単に作ることができました。今後はコントローラ側のプログラミングにも挑戦して、もう少し複雑なことができるものも作れるようになっていたいと思います。

## R E S O U R C E

- [ 1 ] 秋月電子  
<http://www.akizuki.ne.jp/>
- [ 2 ] モルフィー企画  
<http://www.morphyplanning.co.jp/>
- [ 3 ] USB-IOの製品情報  
<http://www.morphyplanning.co.jp/Products/USBIO/>
- [ 4 ] USB-IO用ファームウェア + Visual Basicサンプルプログラム  
<http://www.morphyplanning.co.jp/Products/USBIO/usbhidio.zip>
- [ 5 ] LINUXデバイスドライバ  
Alessandro Rubini著、山崎康宏、山崎邦子 共訳、  
オライリージャパン刊、ISBN4-9009-0073-7
- [ 6 ] Programming Guide for Linux USB Device Drivers  
<http://usb.in.tum.de/usbdoc/>
- [ 7 ] USB COMPLETE  
Jan Axelson著、Lakeview Research社発行、ISBN0-9650-8193-1
- [ 8 ] Linux USB Project  
<http://www.linux-usb.org/>
- [ 9 ] 筆者が開発したUSB-IOのドライバ  
[http://www.narimatsu.net/nari\\_page/musbio.html](http://www.narimatsu.net/nari_page/musbio.html)
- [ 10 ] トランジスタ技術2000年9月号 p.304  
由良義一氏「パソコン用シンクロ・タップの製作」