



Ver1.1 Mar,11 2001

1. //c用マウスの製作

今では、手に入れることが不可能に近くなったApple//c用のマウスの製作方法です。製作は各自の責任において行なって下さいね。

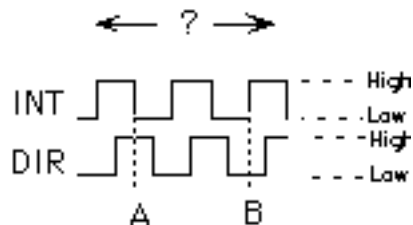
1.1. どのようにマウスの移動を検出しているのか

早く作りたい人には済みませんが、製作に入る前に簡単に、マウスの仕組みを考えておきたいと思います。

動いている方向を決める方法は、ジョイスティックなどでは、動かしたい方向に倒している間その方向のスイッチが入っていてその時間移動するようにしていますし、AppleII等のアナログジョイスティックでは、ボリュームが入っていてそのレベルを読み取っています。では、マウスはどのようにその移動を検出しているのでしょうか？

マウスを動かしている間、移動方向に対してそのスイッチが入るようにするのも手かもしれませんが、ただジョイスティックの仕組みだと斜めの移動が1方向しか検出できないので、なめらかな動きは期待できません。アナログ・ジョイスティックの仕組みだと、なめらかに動きます。(マウスの発明者、ダグラスエンゲルバートが初めに作ったマウスは、アナログ・ジョイスティックと同じように、ボリュームで移動量を検出していたときいています。)

実際のマウスはどうしているかを考える前に、



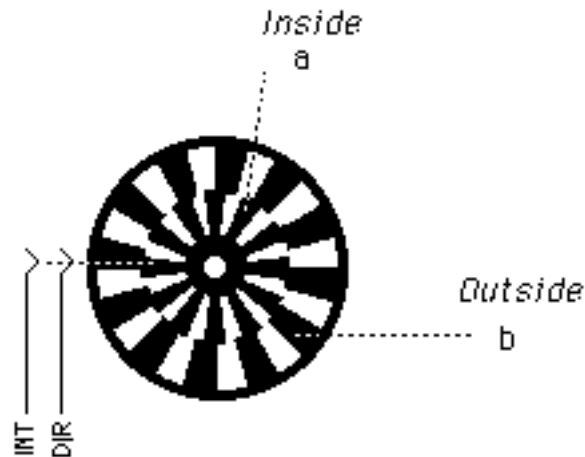
のように、位相の90度ずれた、波形について考えてみてください。INTの信号の立ちさがりの時、DIRのレベルはどうなっているのでしょうか？

Highと答える人が多いかもしれませんが、A点をみてそう思ったかも知れません。それは、多分右の方が時刻が遅いと思ったからでしょう。ただ意地悪なのですが、時間の向きは決まっていないのです。時間軸が逆の場合、つまり左の方が時刻が遅い場合は、INTの立ち下がりの際のDIRのレベルはB点のようにLowになります。

時間軸の向きを動かす方向に置き換えて考えれば、INTの立ち下がりの時のDIRのレベルで、動く向きが分かるはずです。また、マウスの動いた量はそのINTの立ち下がりが来た数で決める事が出来ます。

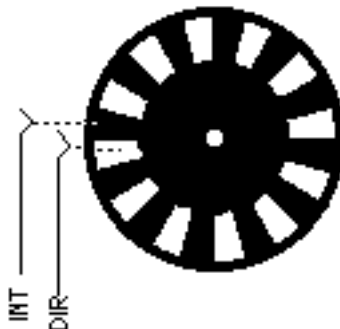
2. マウスのメカニズム

あとは、INTとDIRのような信号を発生するメカニズムを考えればいいのです。一見難しそうですが、答えが分かれば簡単です。このような、メカニズムを考える頭のいい人がいるものですね。



上記のような、内円と外円で位相の90度ずれたパターンを用いればいいのです。実際の回路としては、黒いところを単純に電気的接点にしてもいいですし、最近の主流のように、白いところをくり貫いて、フォト・ディテクター等で検出しても構いません。

もしくは、位相が90度ずれるような場所にその検出するものを配置するのです。



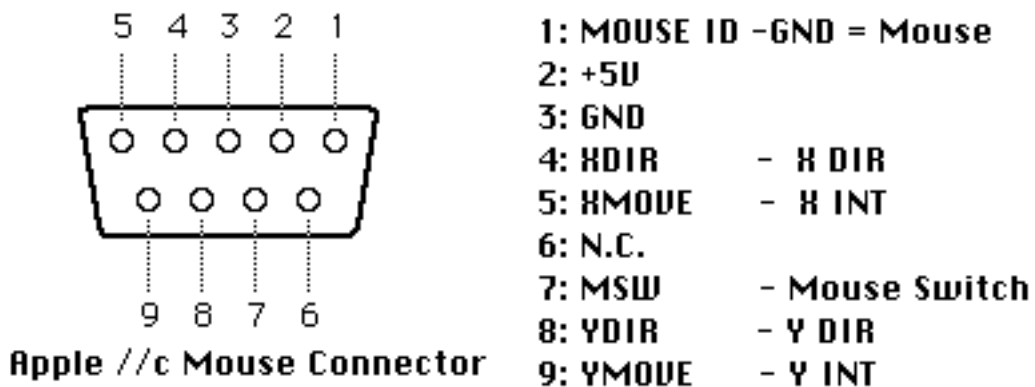
要するに、INT とDIRでそれらを検出して、90度位相がずれた信号が二つ取り出せれば良いのです。これで、このホイールの廻り方によって所望している信号が得られます。

この信号をそのままマウスポートに直結したものが、IIcマウスですし、PC9801シリーズのバスマウスです。ただし、これら二つはピン配置は違いますので互換性は有りません。現在、標準的なシリアルマウスもUSBマウスも検出部分のメカニズムは変わらないので、シリアル信号に変換する前のところで改造すれば使えるでしょう(きっと)。今回は、PC9801のバスマウスを利用します。

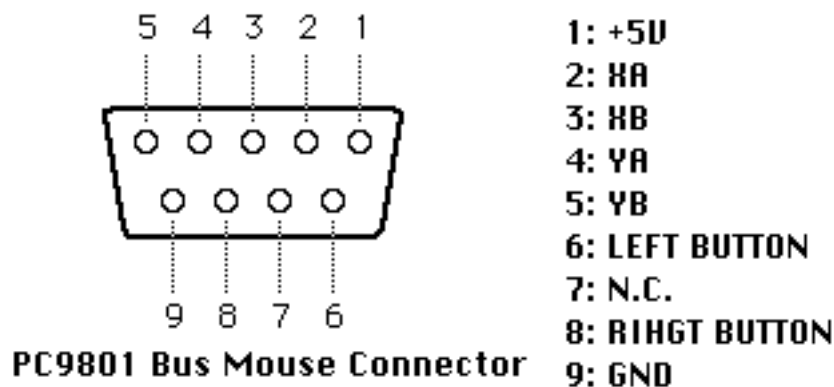
3. バスマウスの流用

3.1. ピンアサイン

まず流用するに当たって、それぞれのピンアサインです。



IIcでは、1pinの論理で、ジョイスティックが繋がっているか、マウスが繋がっているのかを検出しています。マウスの場合、1pinはGNDです。

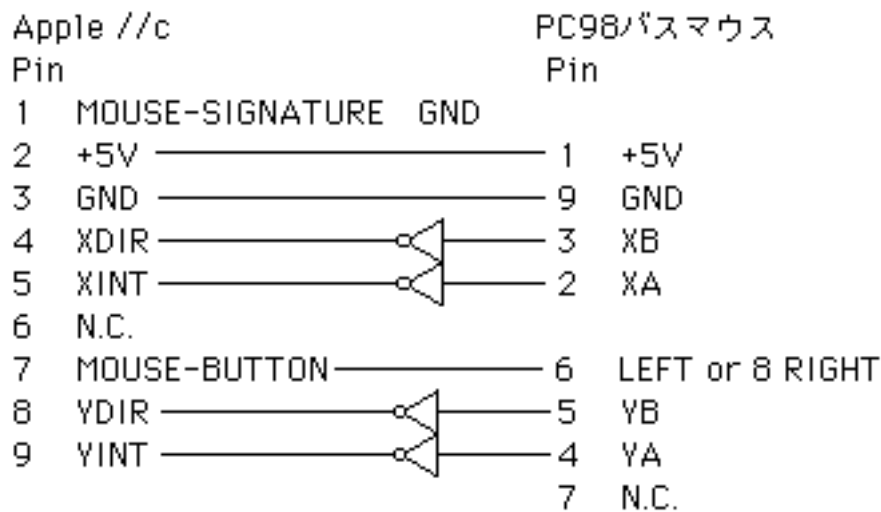


XA, XB, YA, YBと書いてありますが、AとBのどちらがINTに相当し、どちらがDIRに相

当するののかの確かな資料は手元に有りません。昔みたバスマウスの取り扱い説明書のおぼろげな記憶では、AがINTで、BがDIRに相当だったというふうに憶えています。

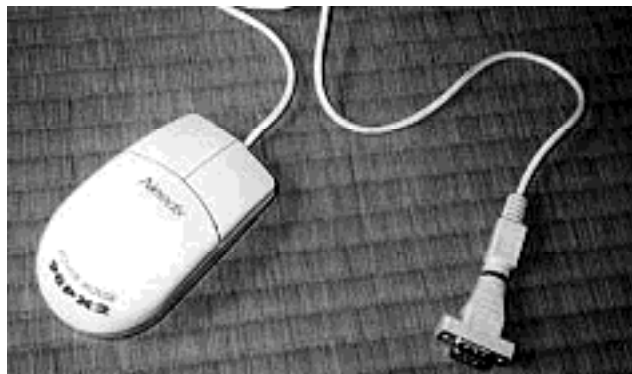
3.2. 変換回路

そこで、以下のようにバスマウスの出力をIICに結線すれば上手く動きます。インバータは、信号のバッファです。直接繋ぐとどうも上手く動きません。インバータを入れてありますが、INTとDIRに両方入れるので、それらの信号位相関係は崩れませんので問題にはなりません。マウスのボタン信号は、RightとLeft、二つともつないでも構いませんし、一つだけでも好きな風に接続してください。

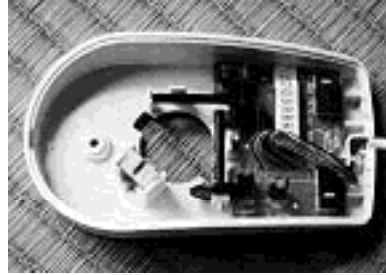


もし、動きが逆になるようでしたら、DIRとINTを入れ換えてみてください。

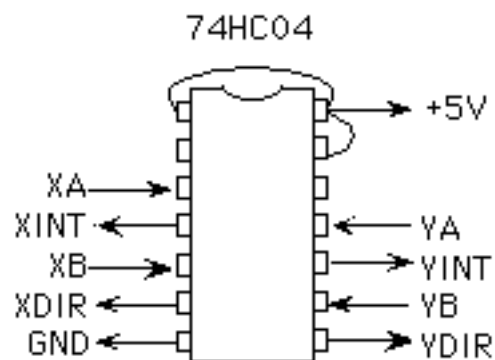
では、以下のマウスを調理します。運の悪いことに、PC98バスマウスでは9ピンコネクタの7ピンに配線が繋がっていません。そこでコネクタは切り落として、別途9ピンDSUBコネクタを付ける必要があります。きり落とす前に、どの色の線が9ピンDSUBのコネクタに繋がっているかを確認して、どの信号になるか把握しておいてください。



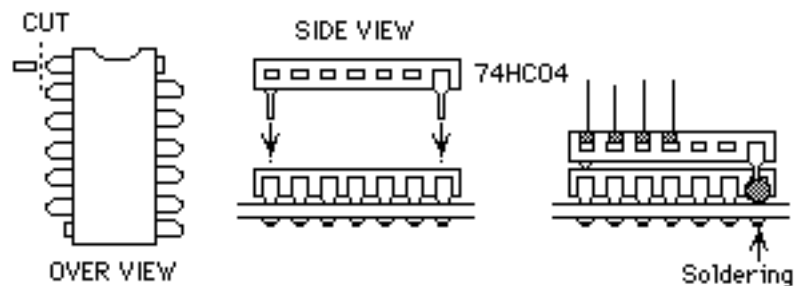
利用したマウスの概観ともとの中身の写真です。



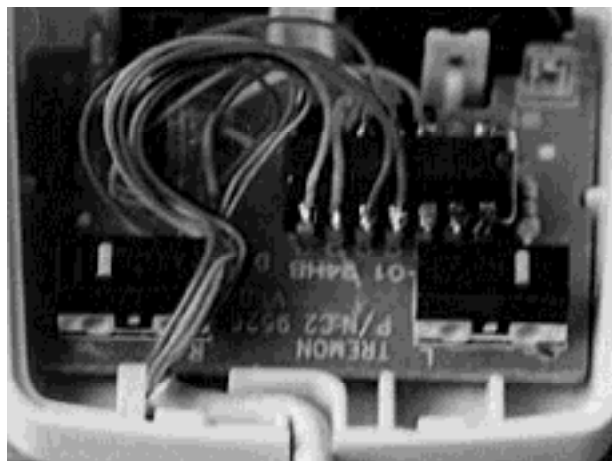
インバータのIC(74HC04)を上からみた配線図です。もう、TTLは手に入らないでしょうから、CMOSの74HCシリーズを使いました。CMOSなので、とりあえず空き入力ピンの処理をしておきました。空きピンは、GNDに繋ぐことが多いですが、今回は近いので+5Vに繋がります。



内蔵するときのコツですが、運良く中に入っていたICと74HC04の電源ピンが同じ位置なので、そのまま、子亀の様にもともとあるICの上に74HC04を載せます。電源ピン以外のICの足は横に開いて、先端のみ切り落とします。その部分に、配線をするのが楽です。電源ピンはそのまま、もとあるICの電源ピンに被せてはんだづけです。あとは、9ピンDSUBの配線を帳じりが合うように配線してください。



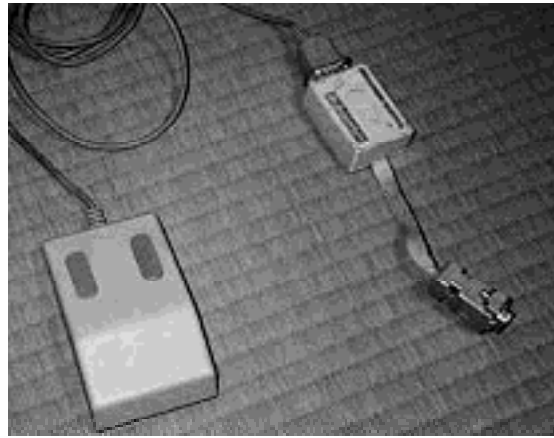
実配線の写真は以下ようになります。配線が出ているICが、追加したICです。



今回作ったマウスの完成図です。もともとあったロゴも、9ピンコネクタも白く塗ってあります。とくに改造してあるようには見えなと思います。



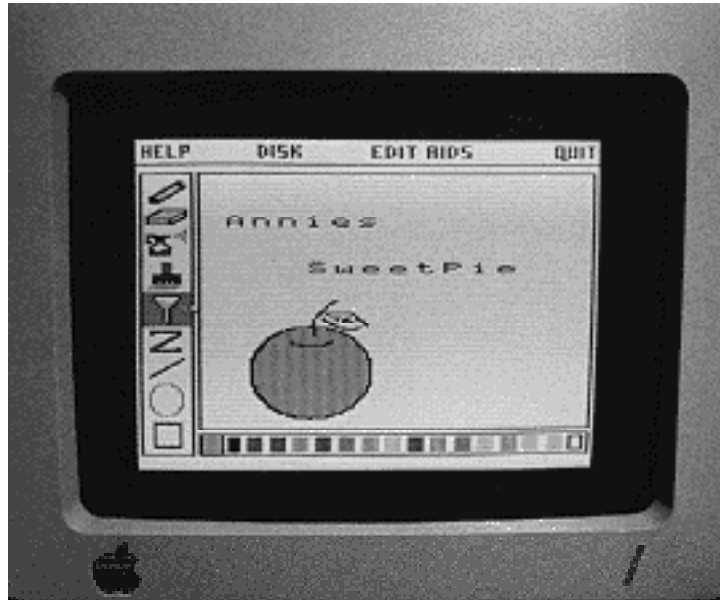
下の写真は10年近く前に作ったものです。当時は、マウスが高かったの変換ボックスを用いて使いました。



今回製作したマウスを使って画を描いているところです。最近手に入るソフトといえば、ByteWorks社のByte Paintぐらいでしょうか？



描いた画の拡大図です。//cモニターでは白黒表示になりますが、普通のテレビに出せば、カラーで表示できます。



4. //cマウステストプログラム

現在、手に入る//cマウスの使えるソフトが少ないので、マウスを使うテストプログラムを書いてみます。

4.1. MOUSEのインターフェイス

Original //cとLaser128のMOUSE・インターフェイスはPORT4 (\$C400~)にあります。メモリー拡張バージョンの//cと//c Plusは、\$C700~だそうです。MOUSE・インターフェイスがあることは、以下のアドレスが

\$Cn0C = \$20

\$CnFB = \$D6

であることでわかります。(n=PORT番号)。この二つで十分かと思いますが、AppleIIのFAQには、

\$Cn05 = \$38

\$Cn07 = \$18

\$Cn0B = \$01

も付け加えられています。

4.2. MOUSEのポートのI/O

詳しい、I/Oのアドレスマップは以下のようになっています。nは、もちろんポート番号です。

```

$Cn05 = $38 Pascal IDバイト
$Cn07 = $18 Pascal IDバイト
$Cn0B = $01 ファームウェアの属性
$Cn0C = $20 2=X-Yポインティングデバイス,0=IDコード
$Cn0D 初期化ルーチン(実装無し)
$Cn0E 読み出しルーチン(実装無し)
$Cn0F 書き込みルーチン(実装無し)
$Cn10 ステータスルーチン(実装無し)
$Cn11 = $00 オプションルーチンが続く
$Cn12 SETMOUSE マウスモードの設定。
$Cn13 SERVEMOUSE マウスの割り込みを受ける。
$Cn14 READMOUSE マウス位置の読み出し。
$Cn15 CLEARMOUSE マウスの位置を0にクリアする(デルタモード用)。
$Cn16 POSMOUSE マウスの位置をユーザーが定義したところへ設定する。
$Cn17 CLAMPMOUSE マウスのウィンドウの移動範囲を指定する。
$Cn18 HOMEMOUSE マウスの範囲指定された中の位置版左上にもってくる。
$Cn19 INITMOUSE マウスの移動範囲を初期値に戻す。マウスの位置は、0,0に設定する。
$CnFB = $D6 マウスIDバイト

```

でも今回は、上のようなルーチンを直接使うことはしません。

4.3. MOUSEをジョイスティック代わりに使う

AppleIIの個々のルーチンは非常によくできています。今回も上手くそれを利用して、マウスが上手く出来たかどうかのテストプログラムを簡単に作ってしまいます。

```

10 HGR : HCOLOR = 3
20 H PLOT PDL(0),PDL(1)
30 GOTO 20

```

「おいおい、これはジョイスティックのプログラムと変わらないじゃないか」と言う人もいるかと思いますが、これをおまじないの後にRUNさせるとマウスが動いた分だけ画面に点が描かれます。そのおまじないは、

- 1.PR#4 と入力してRETRUN。
- 2.Control-Aと入力してRETRUN(画面には見えませんが)。
- 3.PR#0と入力してRETURN。

このあと、RUNすれば良いのです。とっても簡単でしょ。マウスのポートが4でない場合は、PR#4の所を変えて下さい。メモリ拡張バージョン//cや//c+はPR#7となります。これでマウスが上手く動いているかどうかは確認できると思います。もし、マウスの動きと逆

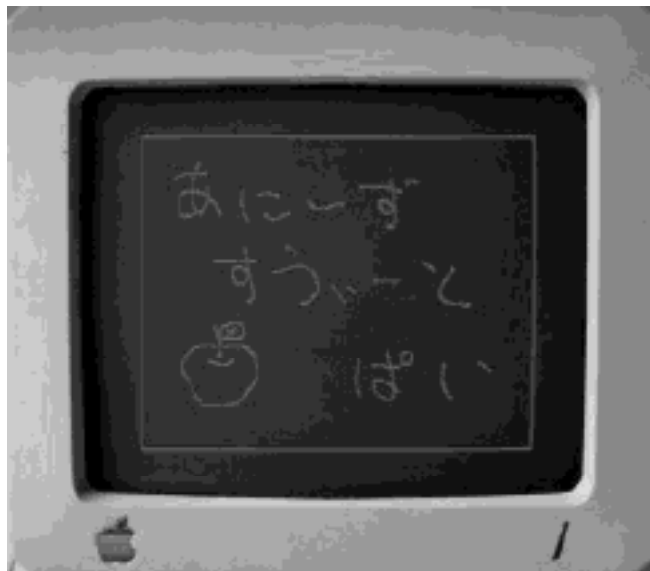
に画面上の線が動いたりした場合は、マウスの中の配線を入れ換えてください。

4.4. MOUSE で画を描きましょう

もう少し手を加えて、マウスで画を描きましょう。Apple Soft BASICのShape機能を使って、カーソルも表示させて、オープンアップルキーをおせば、点を消去できるようにもしました。ついでに、プログラムの中でマウスをジョイスティックの代わりに使うように設定しました。30の行がそれです。また、PDLの値をそのまま使っているの横は255までしか動けません。

```
10 X = 0 : Y = 0 : SCALE = 1 : POKE 232,0 : POKE 233,3
20 FOR A = 768 TO 777 : READ B : POKE A,B:NEXT
30 PRINT CHR$(4) "PR#4" : PRINT CHR$(1) : PRINT CHR$(4) "PR#0"
40 HGR2 : HCOLOR = 3
50 HPLOT 0,0 : HPLT TO 279,0 : HPLT TO 279,191 : HPLT TO 0,191 : HPLT TO 0,0
60 X = PDL(0) : Y = PDL(1) : IF Y > 191 THEN Y =191
70 IF PEEK(49251) = 0 THEN HPLT X,Y
80 IF PEEK(49249) > 127 THEN HCOLOR = 0 : HPLT X,Y : HCOLOR = 3
90 XDRAW 1 AT X,Y : XDRAW 1 AT X,Y
100 GOTO 50
110 DATA 1,0,4,0,51,45,36,63,54,0
```

上のプログラムを用いて描いたものです。



時には、AppleIIと戯れてみてください。では、

参考文献&SITE

LASER128 Tecnnical Reference Manual

Apple//c Tecnnical Reference Second Edition

Apple II FAQ

Byte Works 社<http://www.byteworks.org/>

編集履歴

Ver1.0 :HTMLの形でWebで公開(Aug 1999)。

Ver1.1:PDF化、プログラムのコードにゴミが入っていたのでそれを削除(Mar 11,2001)。