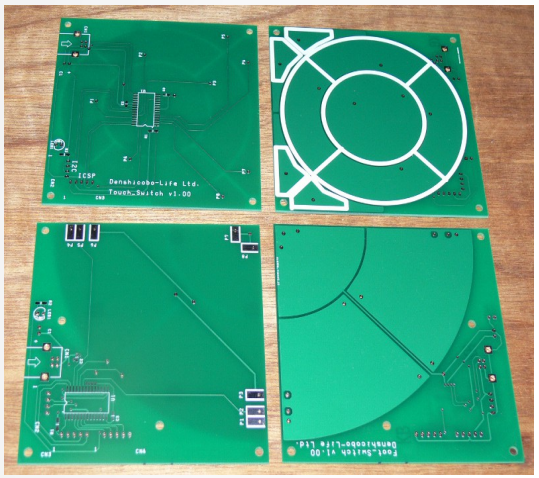
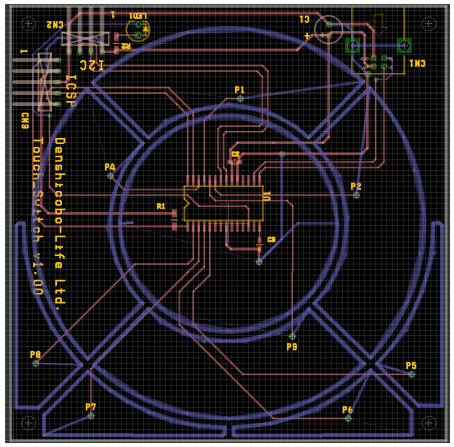
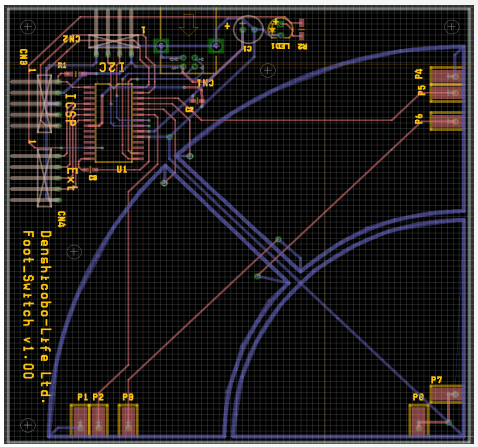
Foot_Switchの開発







XYステージの二次元位置情報を操作する I2Cインターフェースのフット・パネルという、当初の構想を変更して、I2CとUSBを備えたCTMUマウスの開発に着手した。

二次元位置情報の操作パネルとしては使えるがマウスとしての操作性には課題が残る。==>何か改善する策が必要

JAVA　Robotクラスの限界

PICはI2Cを介してCTMUの計測値をRaspberry piに送り、Raspberry piのJAVAプログラムからRobotクラスを呼び出してマウス・ポインタを操作していた。

Robotクラスの利用は簡便だったが、マウスの操作性を大きく損なう原因になった。（Robot関数を呼び出すと0.1秒かかる）

開発上の課題となった、マウスの操作性を改善するため、先送りしてきたUSBマウス機能を組み込むことにした。

以前USBシリアルを組み込んだ経験があるので、USBマウスの組み込みも問題なく行くものと思っていた・・・

USBマウスが動かない！

MPLABX＋XC8（V1.21)でMLA（v2013-06-15）のUSBライブラリをPIC18F25K50用に修正したものを組み込んでみたが、Unkownデバイスになって動かない。

初めて動かす自作基板==>回路設計ミスの可能性

==>パターン切れの可能性

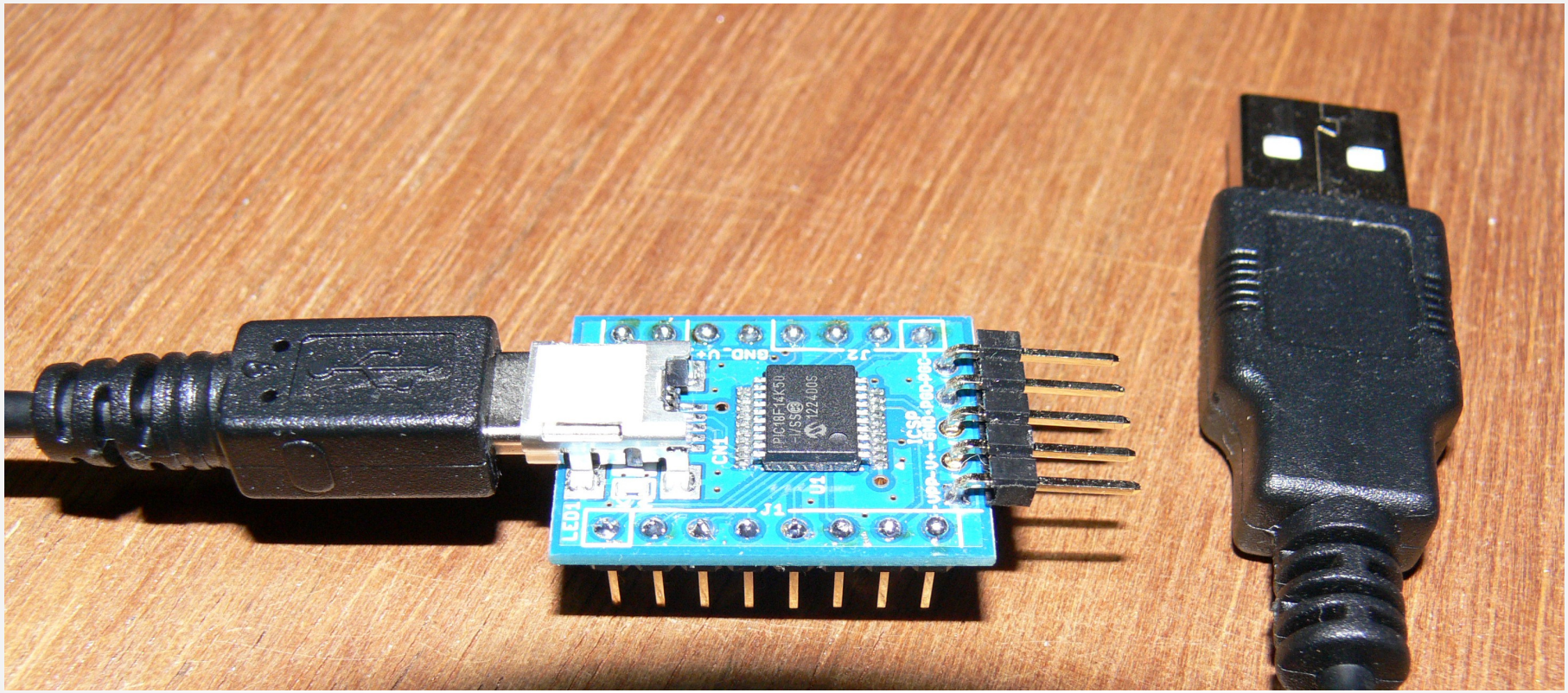
初めて使うPIC18F25K50のACT機能==>設定ミスの可能性

修正して使うUSBライブラリ==>修正ミスの可能性

原因はハードなのか？ソフトなのか？

切り分けが出来ず、途方にくれて年を越した（2013年暮れ）

USBリファレンスを入手した



秋月電子のUSB対応超小型マイコンボード

PIC18F14K50を使用しているので、MLAの修正が殆ど要らない。

（C18用からXC8用への変更が必要）

やっぱり動かない

秋月電子のUSB対応超小型マイコンボードでも同じ状況（Unkownデバイス）になる。==>有力な手掛かり！

ハードは問題ない（筈）

MLAも問題ない（筈）

それでも動かないのは、C18用からXC8用への修正にミスがある（に違いない）

不具合原因はこれ

原因はusb_hal_pic18.hにある以下の記述

```
//----- Defintions for BDT address ----------------

#if defined(__18CXX)

    #if defined(__18F14K50)

        #define USB_BDT_ADDRESS 0x200

    #endif

#define BDT_BASE_ADDR_TAG    __attribute__ ((aligned (512)))

#define CTRL_TRF_SETUP_ADDR_TAG

#define CTRL_TRF_DATA_ADDR_TAG

#endif
```

関連するUSB_DEVICE.Cの記述

```
/** USB FIXED LOCATION VARIABLES ***********************************/
#if defined(__18CXX)
    #pragma udata USB_BDT=USB_BDT_ADDRESS   <== XC8はこれを無視する
#endif

volatile BDT_ENTRY BDT[BDT_NUM_ENTRIES] BDT_BASE_ADDR_TAG;
volatile CTRL_TRF_SETUP SetupPkt CTRL_TRF_SETUP_ADDR_TAG;
volatile BYTE CtrlTrfData[USB_EP0_BUFF_SIZE] CTRL_TRF_DATA_ADDR_TAG;
```

usb_hal_pic18.hの修正

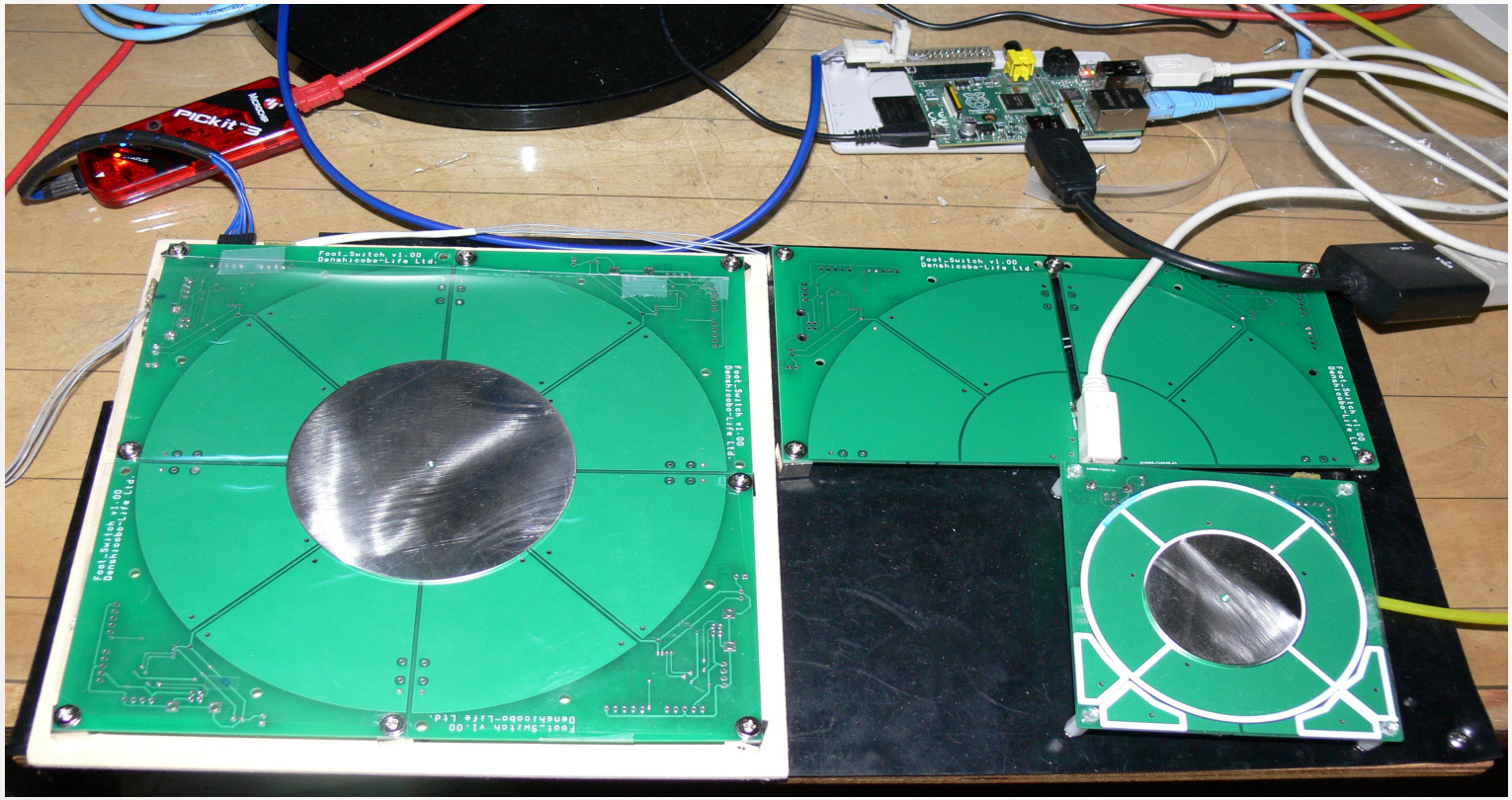
```
#if defined(__18CXX)
    #if defined(__18F14K50)
        #define USB_BDT_ADDRESS 0x200
        #define CTRL_TRF_SETUP_ADDR 0x230
        #define CTRL_TRF_DATA_ADDR 0x238
    #else
        #define USB_BDT_ADDRESS 0x400
        #define CTRL_TRF_SETUP_ADDR 0x430
        #define CTRL_TRF_DATA_ADDR 0x438
    #endif
#endif
```

```
#ifdef __XC8

#define BDT_BASE_ADDR_TAG
@USB_BDT_ADDRESS

#define CTRL_TRF_SETUP_ADDR_TAG
@CTRL_TRF_SETUP_ADDR

#define CTRL_TRF_DATA_ADDR_TAG
@CTRL_TRF_DATA_ADDR

#else

#define BDT_BASE_ADDR_TAG   __attribute__
((aligned (512)))

#define CTRL_TRF_SETUP_ADDR_TAG

#define CTRL_TRF_DATA_ADDR_TAG

#endif
```

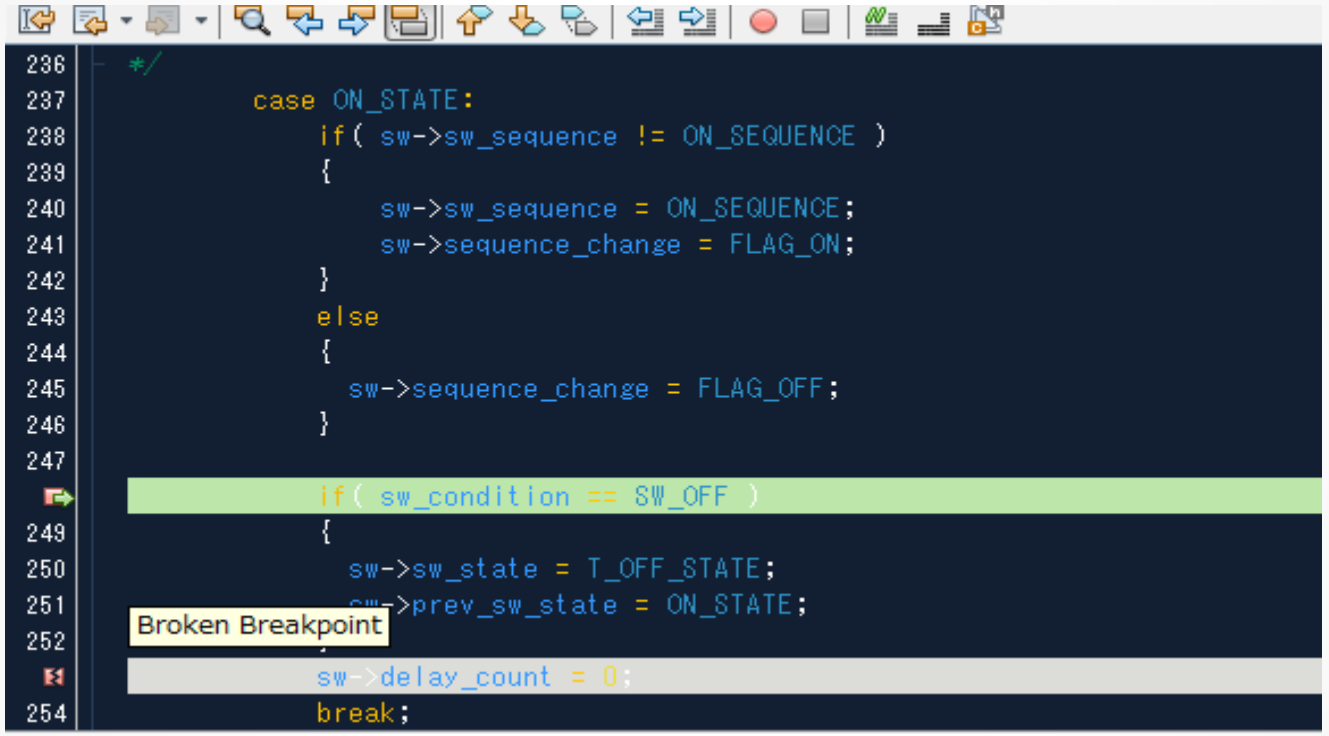
1．4mのケーブルでI2C接続

二つのマウスをおよそ1．4mのケーブルを介してRaspberry PiとI2Cで接続した。

通信エラーの発生は0．1％以下でチェック・サムを使って、エラー・データを排除している。



"Broken Breakpoint"って何？



直前のif文でブレークするが、STEP実行するとifブロックの下のブレーク・ポイントには行かず、別のcase文の中に入る。
253行に”Broken Breakpoint”と表示されている。

対応するアセンブラ行がない

アセンブル・リストを調べると、ソース・コードに対応するアセンブラ行が無い。

```
＜アセンブル・リスト抜粋＞
l9245:            <== ここに来る(4)
            ;sw_operation.c: 158: }
            ;sw_operation.c: 159: sw->delay_count = 0;
    movff       update_sw_common@sw,fsr2l
    movff       update_sw_common@sw+1,fsr2h
    movlw       0
    movwf       indf2,c
            ;sw_operation.c: 160: break;
    goto  l9311
 L9:                <==ここを経由して(3)
    movwf       indf2,c
    goto  l9245
         ;sw_operation.c: 248: if( sw_condition == 0 )
    tstfsz update_sw_common@sw_condition& (0+255),b
    goto  l9245  <== ifブロックに入らない時はここで分岐(1)
         ;sw_operation.c: 249: {
         ;sw_operation.c: 250: sw->sw_state = 1;
         ;sw_operation.c: 251: sw->prev_sw_state = 5;
    movlw       5
    goto  L9     <== ifブロックを抜ける時はここで分岐(2)
l9305:
```

対策

以下のコードを挿入し、そこにBreakPointを設定すれば良い。

```
asm("nop");
```

コード・デバッグの最中は、この最適化処理を止めにしたいのだが、その方法が判らない。

ソース・ファイルごとに最適化処理の要否を選択できれば良いのだが・・・

PIC32MZの開発準備