ペン型オシロスコープ(もどき)の作り方

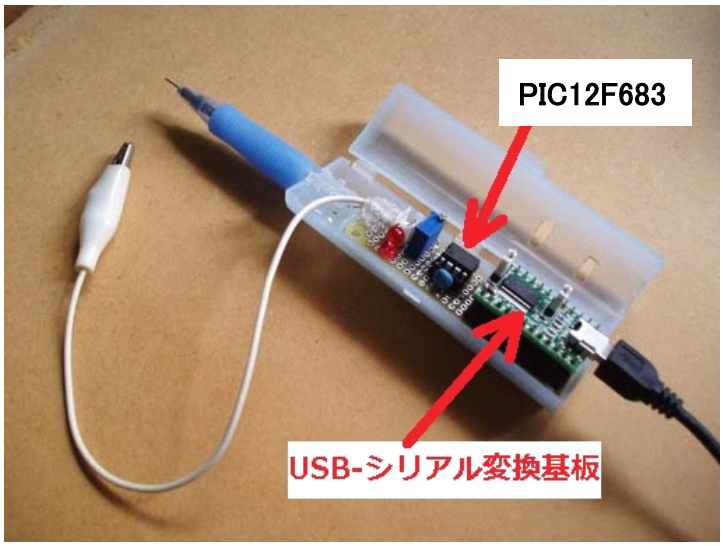
本書は、PC接続タイプの簡易「ペン型オシロスコープ」を自作する方のための解説書です。

開発時間、経費を極力おさえたため、通常の電子回路やファームウェアの作成方法と異なることがあります。動作不具合、故障などは保証いたしません。また、本機を接続、ソフトウェアを使用したことによるパソコンの故障等の一切の責務は当方にはありません。自己責任にてご利用ください。

と、お決まりの文句はここまで。

概略仕様:

・本体サイズ	直径 25、全長 150
・基板サイズ	幅 20、長さ 70
・チャンネル	1ch
・サンプリング	最大 40kHz
・最大入力	12.5V
・A/D 分解能	12.5V/256
・電源	USB
・精度	調整次第



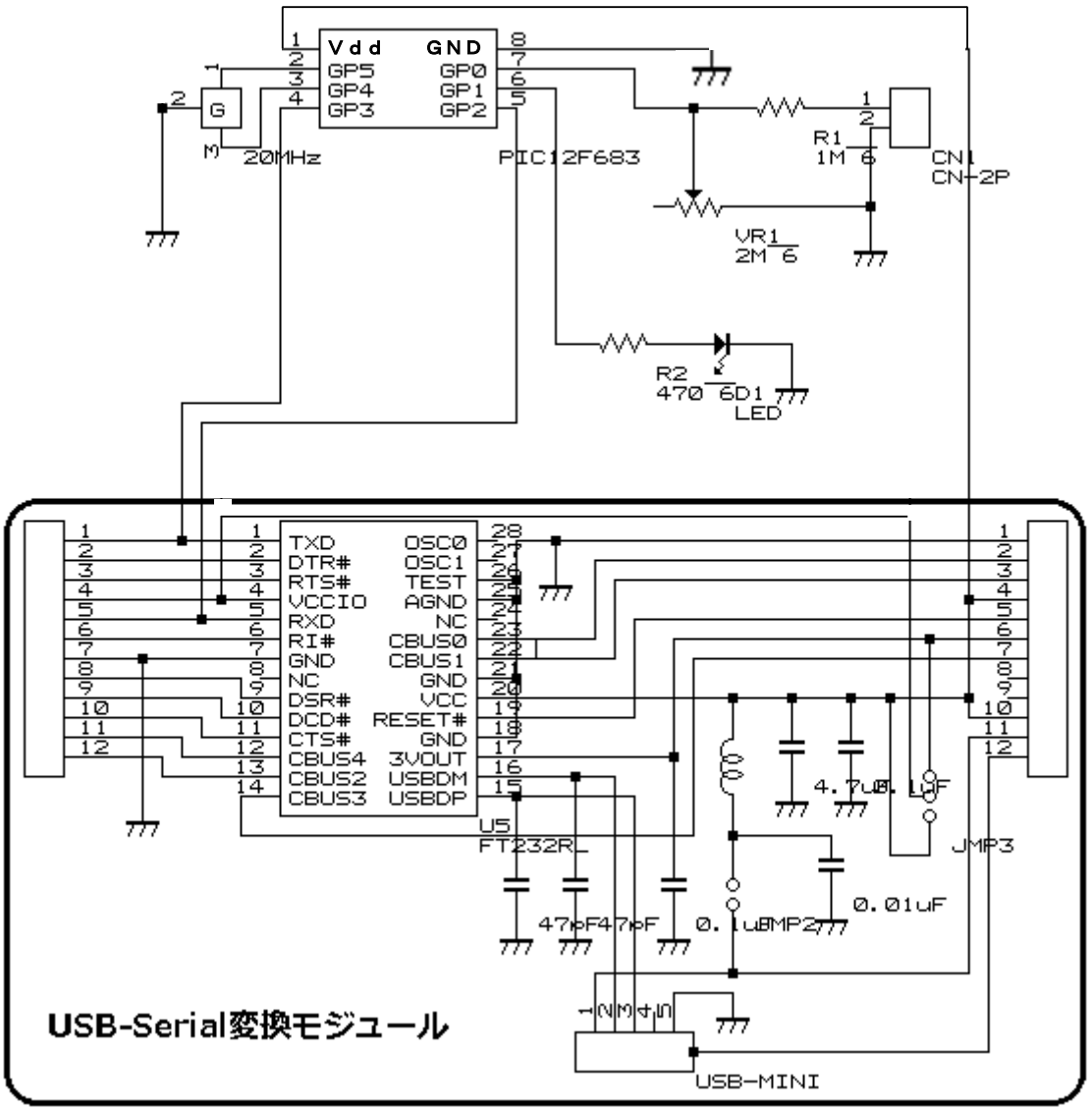
用意するもの:

・PICライター(秋月電子製、マイクロチップ社製、自作など)
・半田ごて、半田、リード線などの電子工作道具
・パソコン

使用する部品:(すべて秋月電子で購入可能)

・USB-シリアル変換モジュール(950 円)
・上記用の 24pin ピンソケット(数十円　大きめのものをカットして使う)
・PIC12F683(150 円)
　機能を増やしたい方は(PIC16F88)などお使いください(A/D、TMR1、CCP があればOK)
・上記用のピンソケット(数十円)
・20MHz のセラミック発振子(40 円)
・2MΩの半固定抵抗(50 円)
・1MΩの抵抗(秋月電子で 100 本 100 円)
・LED(何でもいいです)
・470Ωの抵抗(LED の電流制限用)
・ユニバーサル基板(2.54 ピッチ　20×70 もあれば十分です　カットして使いましょう)
・ワニロクリップ、ジャンパー端子など必要に応じて
・適当なサイズのケース(100 均あたりで探す)
・要らなくなったボールペンの先

回路:



回路図と呼べるようなものではありません。

オシレータは外部(セラミック発振子)なので、GP4 と GP5 は使えません。

四角の枠内はモジュールなので、ここから VDD、GND、TX、RX の4本をつなぐだけです。

測定端子(GP0)は 1MΩの抵抗と、2MΩの半固定抵抗で分圧しています。

Vout=Vin＊(VR1/(R1＋VR1))の関係から

最大入力 12.5V を 5V に変換するので VR1 は 0.67MΩ前後に調整します。

※大きな電圧を測定したい場合は VR1 を下げれば可能ですが、ソフトウェアは未対応です。

※USB-シリアル変換モジュールは、FT232RL という IC を使用しています。この IC を使用するには
メーカから専用のドライバを PC にインストールする必要があります。

※初回のみ、PC に接続する前にドライバをインストールしてください。
詳しくはモジュールに付属の資料を参照してください。

ファームウェア：(PIC アセンブラ)

・使用するデバイス　　PIC12F683

・使用するモジュール　A/D コンバータ、TIMER１、CCP、割込み

・ピン設定

GPIO, 0	A/D 入力
GPIO, 1	インジケータ（LED）
GPIO, 2	シリアル出力（TX）
GPIO, 3	シリアル入力（RX）
GPIO, 4/5	外部オシレータ

◇設定項目（※印はイニシャル設定）

A/D コンバータ

CMCON	00000111	(BANK0)コンパレータオフ※
ANSEL	00100001	(BANK1)Fosc/32、AN0 設定※
ADCON0	00000001	(BANK0)A/D コン Enable

TIMER1/CCP

PIE1	CCP1IE	(BANK1)CCP 割込み Enable※
PIR1	CCP1IF	(BANK0)CCP 割込みフラグ
CCP1CON	00001011	(BANK0)コンペアモード※
T1CON	TMR1ON	(BANK0)タイマースタート

割込み

IOC	00000100	(BANK1)GPIO 2 を割込み設定※
INTCON	GPIE	(BANK0)IO 割込み Enable※
INTCON	GPIF	(BANK0)IO 割込みフラグ
INTCON	PEIE	(BANK0)周辺割込み Enable※
INTCON	GIE	(BANK0)割込み Enable

◇解説

CCP レジスタに設定値を入れて TIMER1 のカウントが一致したときに割込みが発生するようにしています。
この割込みは、サンプリング周期に使用しています。
GPIO 2 は、シリアル受信時に発生します。PC からのスタート、ストップ、設定を常に監視します。

割込みが発生すると INTCON,GIE（割込み ENABLE）は LOW になるので、その割込みが TIMER1 か GPIO かを判断して対応処理をした後、割込みフラグをクリアしてから再度 INTCON,GIE を ENABLE にします。
※割込みのタイミング処理が重要

```
TMR1_SET
        MOVF    T1H,W    ;Sampling Late
        MOVWF   CCPR1H
        MOVF    T1L,W
        MOVWF   CCPR1L
        CLRF    TMR1H
        CLRF    TMR1L
        BSF     T1CON,TMR1ON
```

```
        ORG             0
        GOTO    INIT

        ORG             4
        BTFSS   RX              ;Recieve?
        GOTO    RCV_MODE
        BTFSC   PIR1,CCP1IF     ;Timer1?
        GOTO    AD_MODE
                RETFIE
```

◇シリアル通信解説

通常、PIC マイコンでのシリアル通信は USART などマイコン内で処理しますが、ボーレートが高くできないので、プログラムで処理しています。

最高 460800bps（厳密には 454545bps）を実現するためには 20MHz のクロックで 11 カウントで 1bit を処理しなければならないので LOOP も使っていません。（C言語では絶対に無理です）

※私の PC では通信設定を 460800pbs としても問題ないようです。PC によってはエラーになるかも。
（連続通信は無く、一定時間ごとの char 通信のため？）

PC からスタートコード（"S"）を受信するとマイコンは一定時間ごとに A/D 値を送信し続けます。
これは、エンドコード（"E"）を受信するまで続きます。
※PC 側がオーバーフローしようがお構いなしです。
したがって、PC 側のアプリケーションで一定時間ごとにシリアル受信バッファをクリアする必要があります。

通信プログラム

```
DATA_SND
        BCF     TX              ;Start Bit
        NOP
        NOP
        CALL    BITWAIT0
DATA_BIT
        BTFSS   TXREG,0
        BCF     TX
        BTFSC   TXREG,0
        BSF     TX
        CALL    BITWAIT0
        BTFSS   TXREG,1
        BCF     TX
        BTFSC   TXREG,1
        BSF     TX
        CALL    BITWAIT0
         ・
         ・
         ・
        BTFSS   TXREG,7
        BCF     TX
        BTFSC   TXREG,7
        BSF     TX
        CALL    BITWAIT0
        BSF     TX              ;Stop Bit
        RETURN

BITWAIT0
        NOP
        NOP
        NOP
        RETURN
```

```
RCV_MODE
        CLRF    RXREG
        CALL    BITWAIT0
RCV
        BTFSC   RX      ;Data Bit0
        BSF     RXREG,0
        NOP
        NOP
        CALL    BITWAIT0
        BTFSC   RX      ;Data Bit1
        BSF     RXREG,1
        NOP
        NOP
        CALL    BITWAIT0
         ・
         ・
         ・
        BTFSC   RX      ;Data Bit7
        BSF     RXREG,7
        CALL    BITWAIT0
COMPARE
        MOVF    RXREG,W
        SUBLW   'S'     ;Start Code
        BTFSC   STATUS,Z
        GOTO    TMR1_SET
        MOVF    RXREG,W
        SUBLW   'E'     ;End Code
        BTFSC   STATUS,Z
        GOTO    TMR1_END
        GOTO    START
```

以上のように、難しい処理や特殊なことは何もやっていません。その方が確実な通信ができるからです。
他に良い方法があればご自由に改造してください。

PC アプリケーション：(Visual Basic)

※このアプリケーションはインストールする必要はありません。任意の場所にファイルを置いて
実行するだけで起動します。

ペン型オシロスコープを PC に接続した状態で pen-oscillo.exe を実行します。
正常に起動すると左下に「PORT OPEN」が表示されます。
左上の COMPORT はご使用の PC によって違う番号が表示される場合があります。
※PORT OPEN が表示されない場合、COMPORT のドロップダウンメニューから他のポートを選択してください。
他のポートでも PORT OPEN にならない場合、正常に接続されていないことが考えられますので、
USB を接続しなおしてからアプリケーションを再起動してください。

右上の START ボタンを押すとデータが表示されます。STOP ボタンで停止。

機能説明

- Interval　　オーバーフローしないようにデータを破棄する間隔（通常はそのまま）
- Sampling　　横軸の間隔を設定します。(動作中は操作不可)
- Range　　縦軸の値を設定します。(動作中に操作可能)
- Trig　　赤い横軸がトリガー値になります。
　　※トリガー機能は不安定です。
- Single　　トリガーが働いたときに動作を停止します。
- エッジ選択　　立上り／立下がりを選択します。
- 波形画面上をクリックすると下部ステータスバーにクリック位置の数値を表示します。
- PORT OPEN/CLOSE の文字をクリックすると切り替わります。
- ウィンドウ幅は大きく出来ます。