

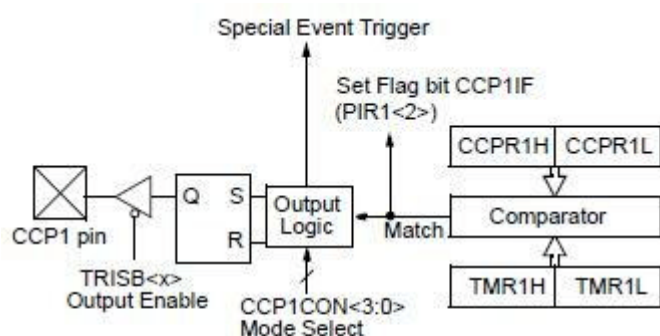
ストップウォッチ(8ビット版)

概要

前回製作した、ストップウォッチ(16ビット版)の8ビット版を製作してみました。機能、性能は同じです。プログラムの構造上は、16ビット版の方が少しだけですが簡単です。

動作原理

- クロックには20.0000MHzの高精度なクロックモジュールを使用しました。
- 正確な1msecを得る方法
CCPモジュールをコンペアモードで使用しました。
TMR1とPIR1がコンペアされTMR1がPIR1に達すると割り込みが発生します。プリスケール値は、1/1とします。
PIR1には、5000を設定します。
 $5000 = 1\text{msec} \div (1 \div (20\text{MHz} \div 4))$



Special event trigger will:

- RESET Timer1, but not set interrupt flag bit, TMR1IF (PIR1<0>)
- Set bit GO/DONE (ADCON0<2>) bit, which starts an A/D conversion

- ストップウォッチの開始
単純にポートを監視し、立下りで、即開始します。
- カウント
開始されると、先程の1msecの割り込み処理の中でカウント値をインクリメントします。
- ストップウォッチの停止
外部割り込み(立下り)で停止します。

回路図



```
//*****
*
/*
    <ストップウォッチ>
    ■ 機能概要
        開始と停止スイッチによるストップウォッチの機能を提供する。
        精度は、□□□□とする。
    □■□□□□の設定
        □LVP_OFF
        □□□MCLR_OFF
        □□□WDT_OFF
        □□□EXTCLK
    */
//*****
*

#define          LED          PORTA.F2

#define          START_SW     PORTB.F1
#define          STOP_SW      PORTB.F0

//*****
*
```

```
static unsigned long cnt;
static unsigned char flag;

void interrupt()
{
    //□□□□□□の割り込み処理
    if (PIR1.CCP1IF == 1) {
        PIR1.CCP1IF = 0;
        //
        if (flag == 1) {
            cnt++;
        }
    }
    // 停止スイッチの割り込み処理
    if (INTCON.INTF == 1) {
        INTCON.INTF = 0;
        //
        flag = 2;    // count stop!
        LED = 1;    // LED off!
    }
}

//*****
*

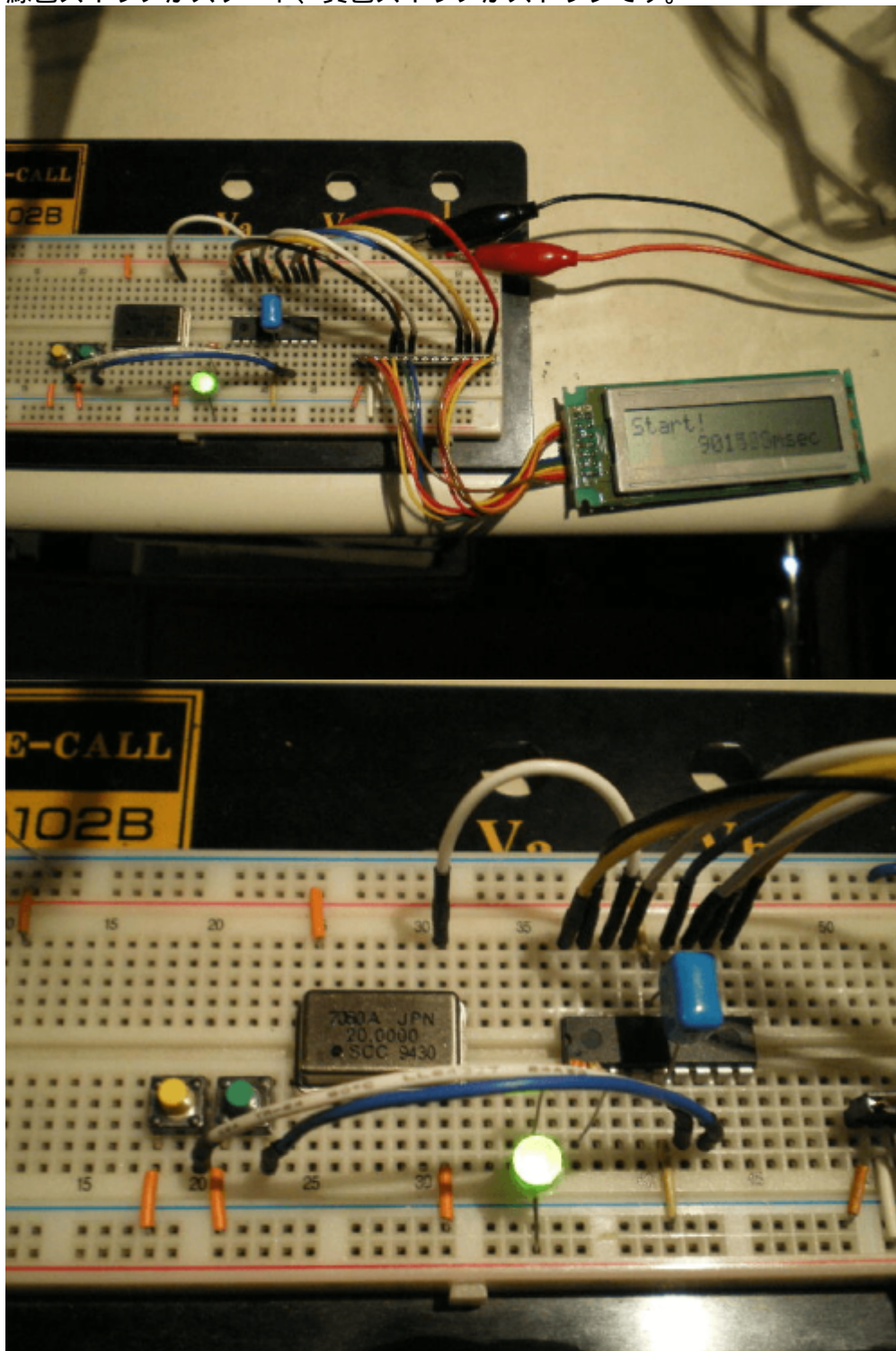
void main()
{
    static unsigned char buf[20];
    // アナログの設定
    ANSEL = 0b00000000;    // 使用しない。
    // ポートの設定
    TRISA = 0b10111000;
    TRISB = 0b00001111;
    OPTION_REG.F7 = 0;    // PORTBをプルアップする。
    // 入力割り込みの設定
    INTCON.INTE = 1;
    INTCON.INTF = 0;
    OPTION_REG.INTEDG = 0;
    // CCPの設定
    PIE1.CCP1IE = 1;
    PIR1.CCP1IF = 0;
    CCP1CON = 0b00001011;
    CCPR1L = 0x88;    // 0.001sec...(1÷20000000)*4*5000
    CCPR1H = 0x13;
    // TIMER1の設定
    PIE1.TMR1IE = 0;
    PIR1.TMR1IF = 0;
    TMR1L = 0;
    TMR1H = 0;
    T1CON.T1CKPS0 = 0;
    T1CON.T1CKPS1 = 0;
```

```
T1CON.TMR1ON = 1;
// 変数の初期化
TMR1L = 0;
TMR1H = 0;
// □□□□液晶モニタ)の初期化
Lcd_Custom_Config(&PORTB,4,5,6,7,&PORTA,1,0,6);
Lcd_Custom_Cmd(LCD_CURSOR_OFF);
Lcd_Custom_Out(1, 1, "StopWatch V2");
Delay_ms(1000);
Lcd_Custom_Cmd(LCD_CLEAR);
// 割り込み(全体)の設定
INTCON.PEIE = 1;
INTCON.GIE = 1;
//
LED = 1;          // LED off!
flag = 0;
cnt = 0;
//
while(1) {
    // 開始を判断する。
    if ((flag != 1) && (START_SW == 0)) {
        cnt = 0;
        flag = 1;    // count start!
        LED = 0;     // LED on!
        Lcd_Custom_Out(1, 1, "Start!");
    }
    // 停止を判断する。
    if (flag == 2) {
        Lcd_Custom_Out(1, 1, "Stop! ");
        flag = 0;
        //
        LongToStr(cnt, buf);
        Lcd_Custom_Out(2, 1, buf);
        Lcd_Custom_Out(2, 12, "msec");
    }
    // 開始中はカウント値を表示する。
    if (flag == 1) {
        LongToStr(cnt, buf);
        Lcd_Custom_Out(2, 1, buf);
        Lcd_Custom_Out(2, 12, "msec");
    }
}

//*****
*
```

動作確認

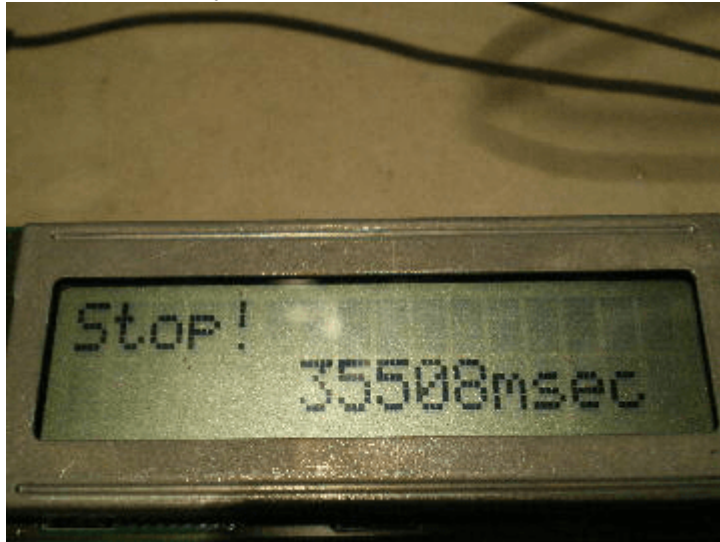
緑色スイッチがスタート、黄色スイッチがストップです。



左側:計測しているところです。秒以下は目では見えないです。 右側:停止したところです□1msecの単位



まで分かります。



著作権表示 **copyright notice**

このページは稲崎様の閉鎖したHPのコピーで、著作権は稲崎様にあります。[詳細](#) This page is a copy of Mr. Inasaki's closed website, and the copyright is held by him.[Details](#)

From:

<http://deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic16f88:81>

Last update: **2020/05/01 17:58**

