

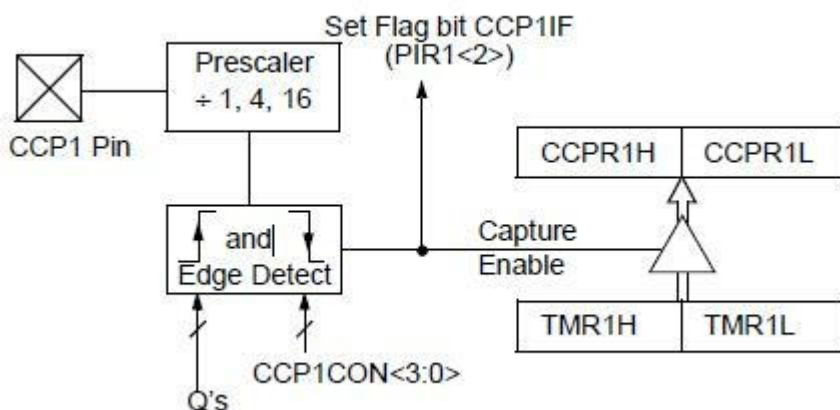
# パルス周期測定ユニット

## 概要

パルスの周期を測定するユニットです。心拍検知ユニットの出力(心拍パルス)の周期を測定するのに最適です。

## 動作原理

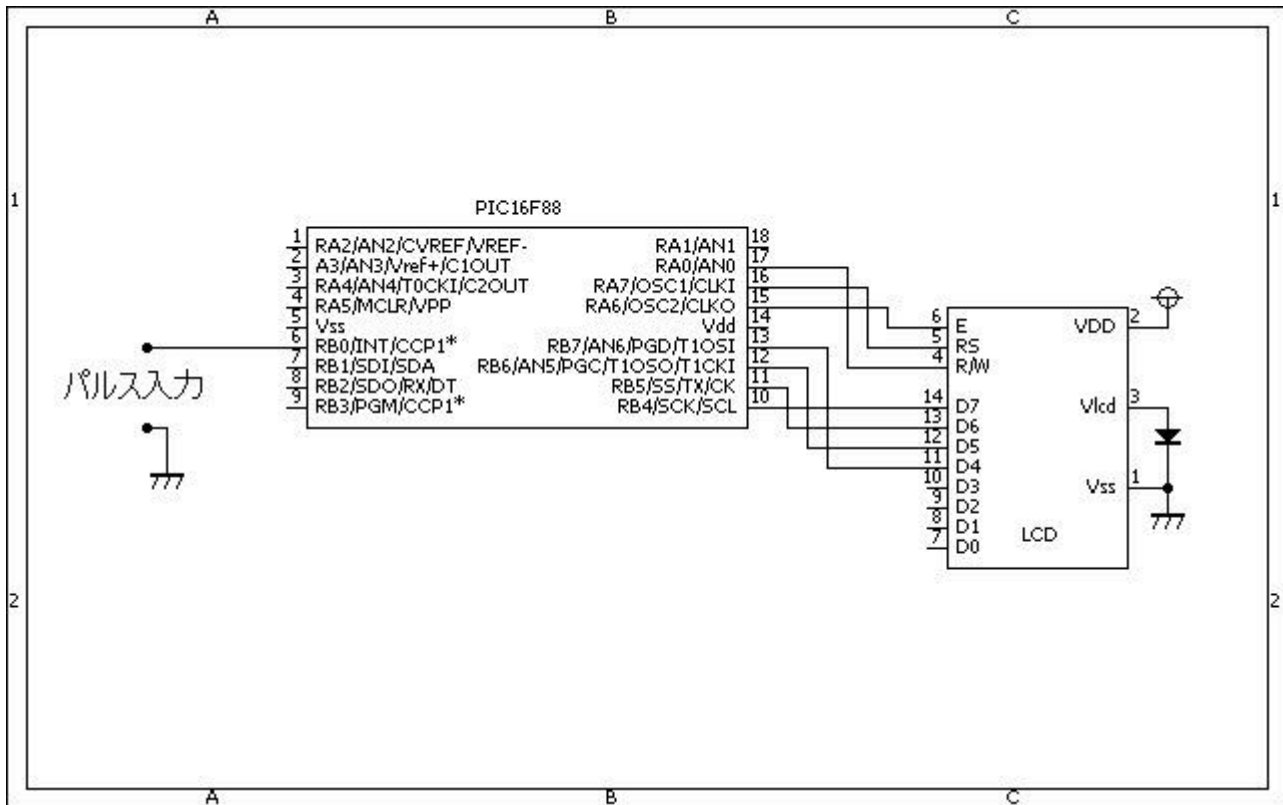
PIC16F88に装備されているCCPモジュールには外部信号の間隔を測定できるキャプチャモードがありますのでこれを使用し、パルスの立ち上がりから次のパルスの立ち上がりまでの時間を計測します。



- パルスは、CCP1ピンより入力されます。
- パルスは、Prescalerを通して分周されます。今回は分周を1とします。
- エッジは立ち上がりエッジとします。
- タイマ1(TMR1)のクロックは内部クロック(今回は1MHz)を利用します。
- タイマ1のプリスケアラは、8とします。  
従って、キャプチャの最小単位は、0.032msecとなります。  
 $0.000032\text{sec} = (1 / 1000000\text{Hz}) * 4 * 8$   
また、キャプチャの最大は、2.09712secとなります。  
 $2097.12\text{msec} = 0.032\text{msec} * 65535$
- キャプチャされるとCCP1Fが“1”になるので、これを監視します。
- CCP1Fが“1”になった時に、その時点のタイマ1の計数値がCCPR1H/CCPR1Lにセットされます。
- CCPR1H/CCPR1Lを読み込んで0.032msecを掛けるとパルス周期が求められます。
- パルス周期より、60秒間のパルス数が求められます。  
 $60000\text{msec} \div \text{パルス周期}$

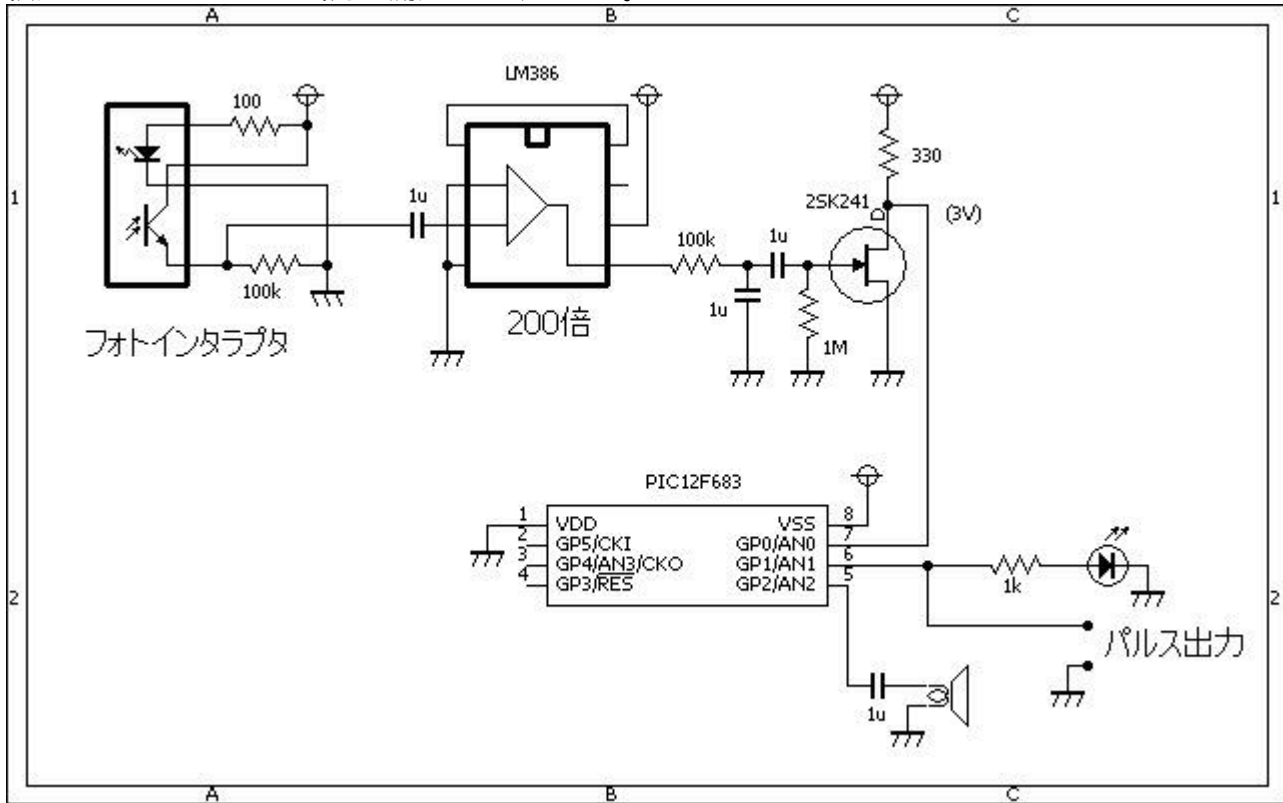
## 回路図

PICとLCDだけの単純な回路です。



心拍

検知ユニットのパルス出力を測定してみました。



## ソースコード

[PulseMeasure.c](#)

```
//*****
```

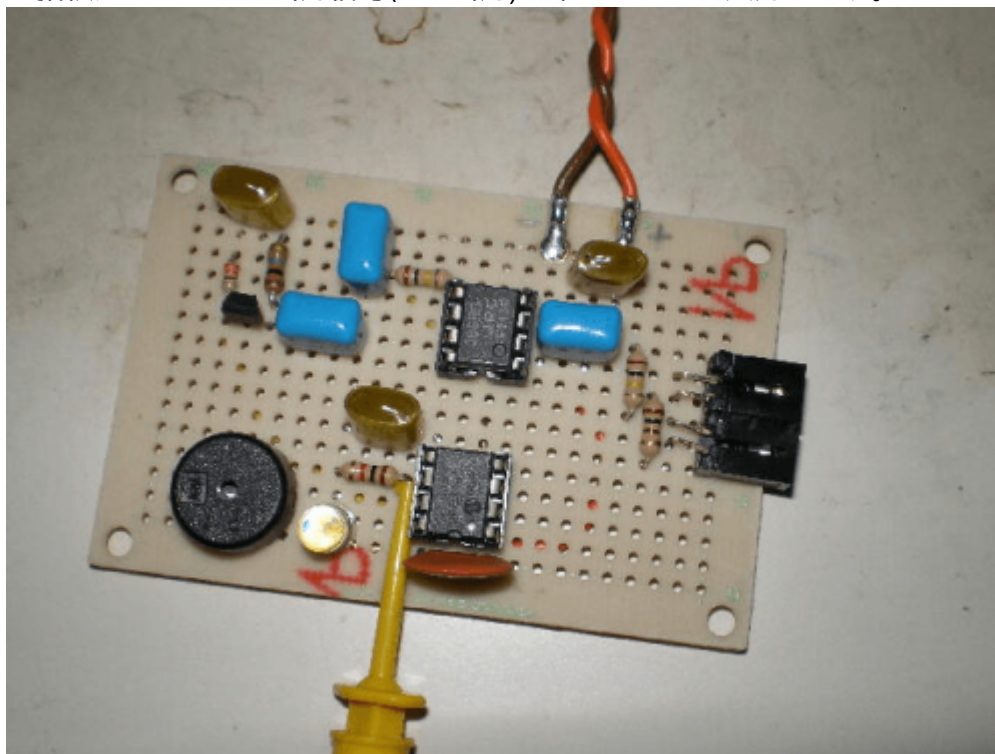
```
*  
  
void main()  
{  
    unsigned long dat;  
    unsigned int dat2;  
    unsigned char buf[20];  
    //  
    OSCCON = 0b01000000; // クロックは1Mhz  
    CMCON = 0b00000111; // コンパレータは使用しない。  
    ANSEL = 0b00000000; // A/D変換は使用しない。  
    TRISA = 0b00111110;  
    TRISB = 0b00001111;  
    //  
    T1CON.TMR1CS = 0;  
    T1CON.T1CKPS0 = 1;  
    T1CON.T1CKPS1 = 1;  
    T1CON.TMR1ON = 0;  
    TMR1H = 0;  
    TMR1L = 0;  
    PIE1.TMR2IE = 0;  
    PIR1.TMR2IF = 0;  
    //  
    CCP1CON.CCP1M3 = 0;  
    CCP1CON.CCP1M2 = 1;  
    CCP1CON.CCP1M1 = 0;  
    CCP1CON.CCP1M0 = 1;  
    CCPR1H = 0;  
    CCPR1L = 0;  
    PIE1.CCP1IE = 0;  
    PIR1.CCP1IF = 0;  
    //  
    Lcd_Custom_Config(&PORTB, 4, 5, 6, 7, &PORTA, 0, 7, 6);  
    Lcd_Custom_Cmd(LCD_CURSOR_OFF);  
    Lcd_Custom_Out(1, 1, "Pulse Measure v1");  
    Lcd_Custom_Out(2, 1, "JF3SFB{^_^}chan!");  
    Delay_ms(1000);  
    Lcd_Custom_Cmd(LCD_CLEAR);  
    //  
    T1CON.TMR1ON = 1;  
    while (1) {  
        asm { // キャプチャフラグ確認(asm) 高速化!!  
            loop_001:  
                btfss PIR1, 2  
                goto loop_001  
            }  
        //  
        // while (PIR1.CCP1IF == 0) // キャプチャフラグ確認(c)  
        // ;  
        //  
        T1CON.TMR1ON = 0; // タイマーオフ
```

```
TMR1H = 0;
TMR1L = 0;
T1CON.TMR1ON = 1;           // タイマーオン
PIR1.CCP1IF = 0;
//
dat = CCPR1H << 8;
dat |= CCPR1L;
//
dat2 = (double)dat * 0.032;           // msec変換 0.032 =
(1 / 1000000Hz) * 4 * 8 * 1000
IntToStr(dat2, buf);
Lcd_Custom_Out(1, 1, &buf[2]);
Lcd_Custom_Out_Cp("msec");
//
dat2 = 60000.0 / (double)dat2;       // 脈拍数変換(1分間)
IntToStr(dat2, buf);
Lcd_Custom_Out(2, 1, &buf[2]);
Lcd_Custom_Out_Cp("カイ " );
}
}

//*****
*
```

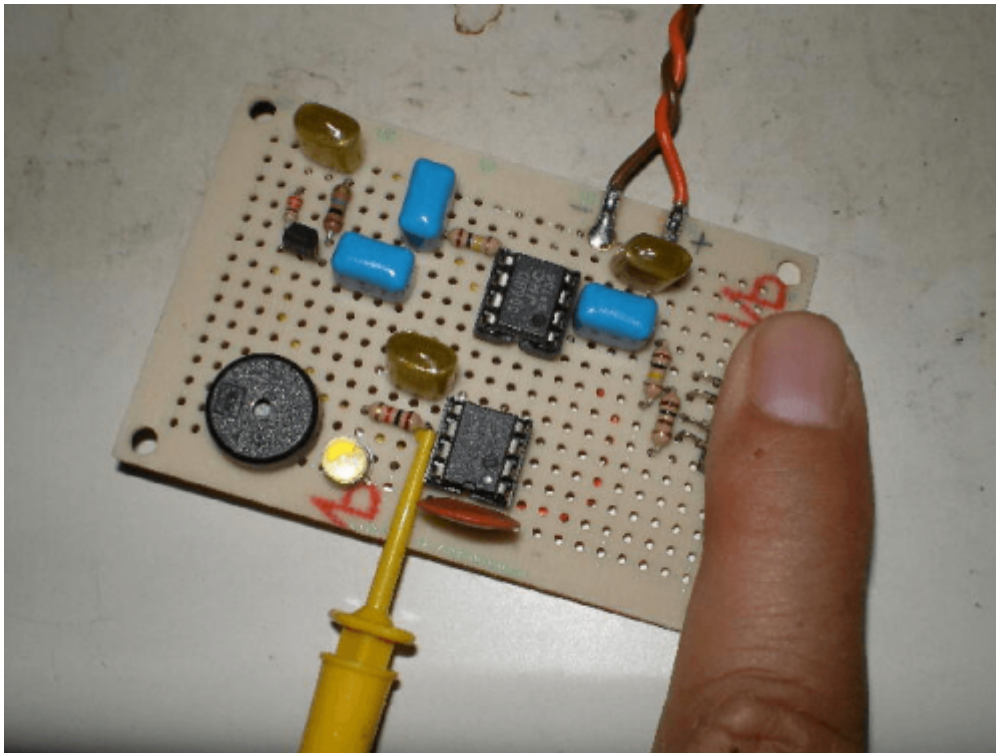
## 動作確認

心拍検知ユニットの出力信号(LED出力)を本ユニットへ入力します。

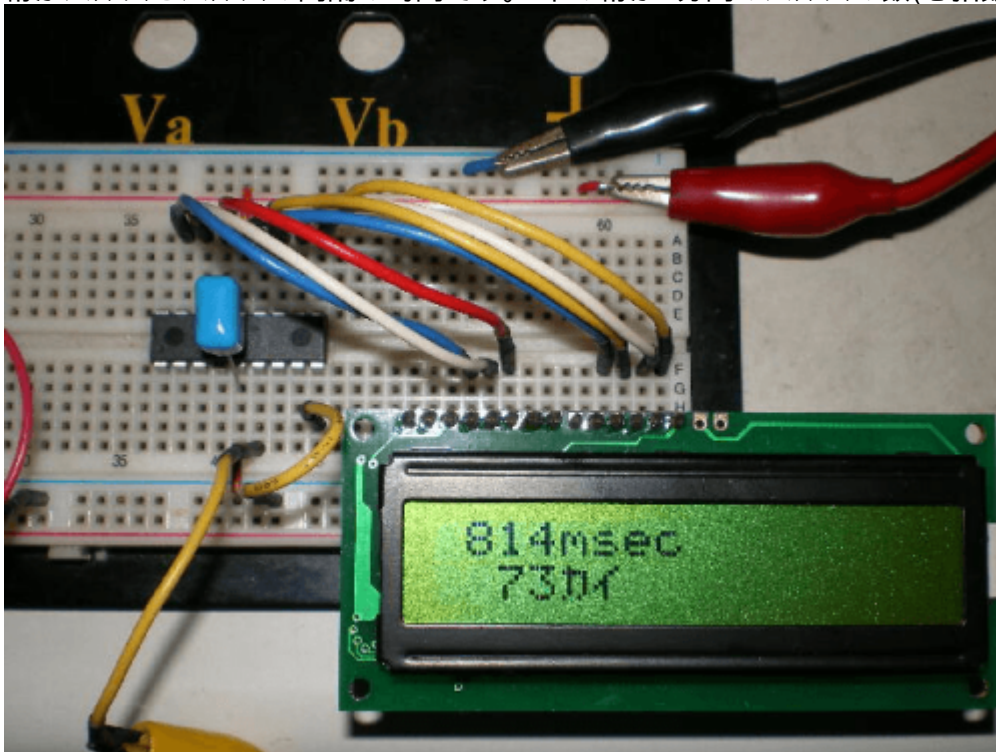


センサー部分(フォトイン

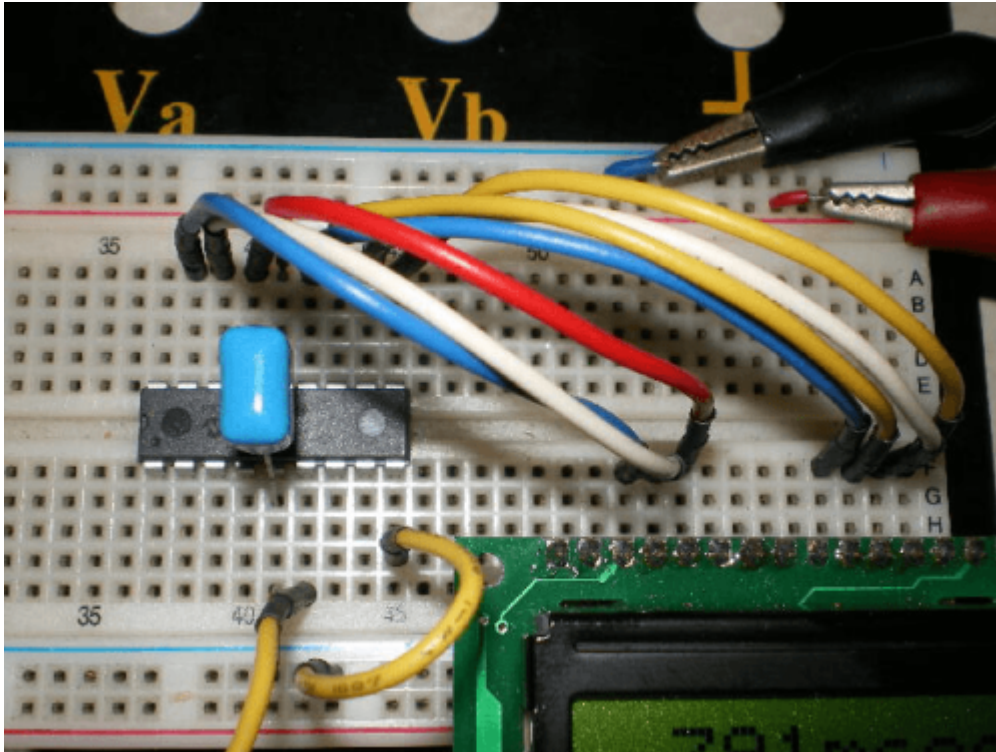
タラプタ)へ指を当てます。



計測した結果です。上の桁がパルスとパルスの間隔の時間です。下の桁が1分間のパルスの数(心拍数)です。



PICとLCDの接続は7本(データ信号4本、制御信号3本)だけです。



如何ですか、とても簡単な回路でパルスの周期を計測することが出来ましたね。但、今回は基準クロックにPIC内臓クロックを使用しましたので、あまり精度は高くありません。出来れば外部に精度の高い1MHzのクロックモジュールを使って頂ければ宜しいかと。。。

From:  
<http://www.deepsky.jp/wiki/> - うごくといいな

Permanent link:  
<http://www.deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic16f88:55&rev=1588146618>

Last update: 2025/10/17 14:28

