

簡易雷距離計

概要

私が住んでいるところでは、夏場には、よく雷が発生します。そこで落雷地点までの距離を測定できないものかと考えました。

昔話 在住する三田市には、桑原 欣勝寺という雷除けで有名なお寺があります。昔々、雷の子供が井戸に落ち、「助けてくれ」というのを、和尚は日頃から雷に迷惑をかけられているので、二度と桑原に雷を落とさないよう約束させて帰した。これ以降、「くわばらくわばら欣勝寺」といえば雷が落ちなくなったという伝承があるそうです。

動作原理

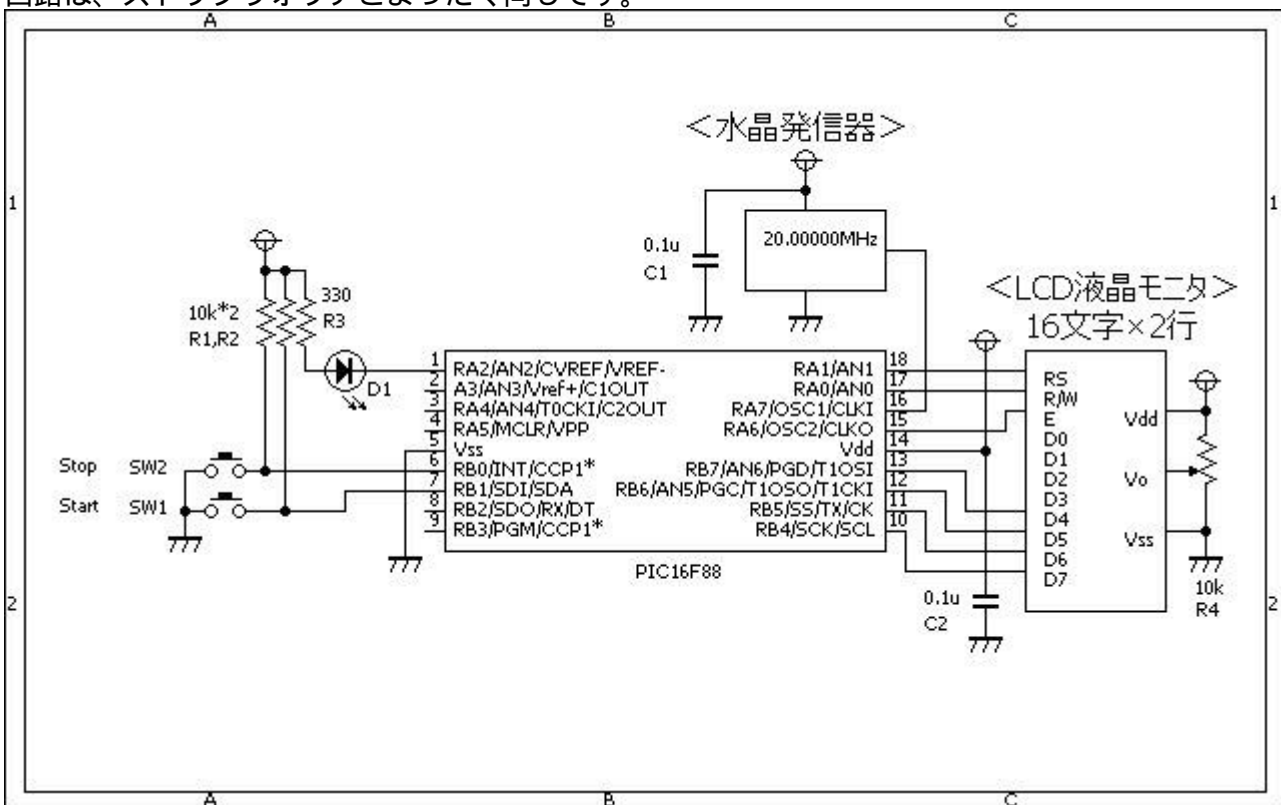
基本的には、以前に製作した、ストップウォッチの応用です。つまり、落雷で雷が光ってから、雷鳴が聞こえるまでの時間を測定(手動操作)し、その結果に音速を掛け算し、距離を表示します。

国際標準大気(ISA:International Standard Atmosphere)では、海面上で気温15 での音速は、約 340 m/s (= 1225 km/h) となります。これは、1 気圧中の音速の計算方法 $331.5\text{m/sec} + (0.61 \times \text{摂氏温度})$ で求められます。

今回は、摂氏温度を20 と固定値を使用しました。

回路図

回路は、ストップウォッチとまったく同じです。



ソースコード

ThunderRange.c

```
//*****
*
/*
< 雷距離計 >
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。
対象を音速とした時の距離を表示する。
□■□□□□の設定
□LVP_OFF
□□MCLR_OFF
□□WDT_OFF
□□EXTCLK
*/
//*****
*

#define          LED          PORTA.F2

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

#define          ON          0
#define          OFF         1

#define          IDLE       0
#define          START      1
#define          STOP       2

//*****
*

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 = STOP;    // count stop!
    }
}

//*****
*

void display(unsigned long cnt)
{
    static unsigned char buf[20];
    //
    LongToStr(cnt, buf);
    Lcd_Custom_Out(2, 2, buf);
    Lcd_Custom_Out(2, 13, "msec");
    //
    LongToStr(cnt * 0.3437, buf);    // 0.3437 = (331.5 + (0.61 * 20))
/ 1000
    Lcd_Custom_Out(1, 8, &buf[6]);
    Lcd_Custom_Out(1, 13, "m");
}

void main()
{
    // 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
```

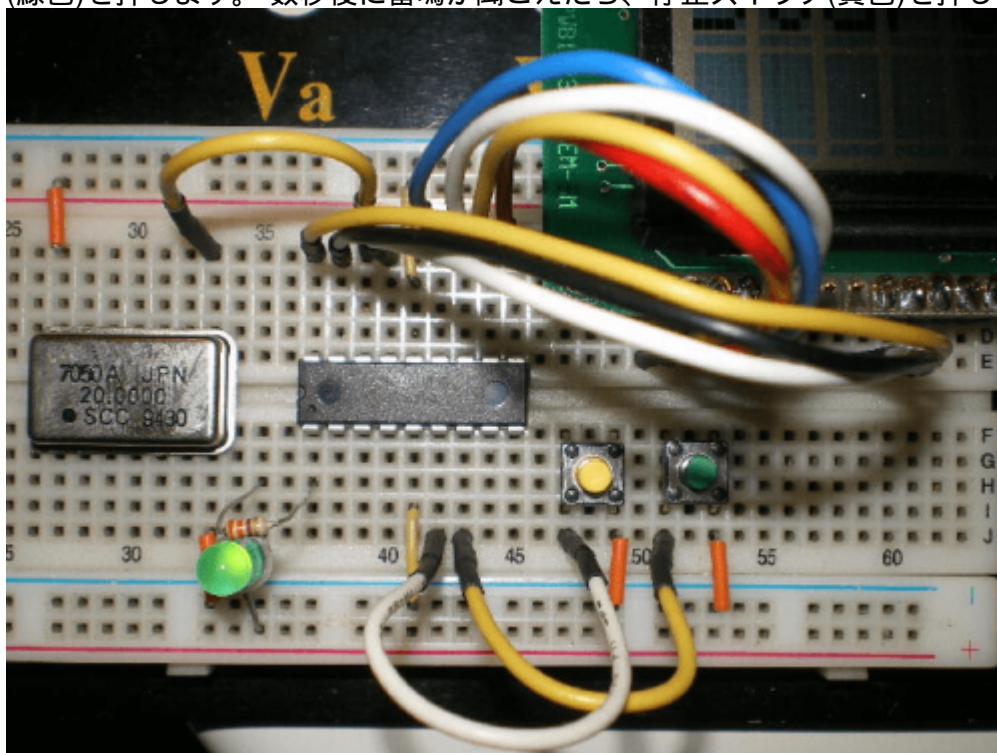
```
TMR1L = 0;
TMR1H = 0;
// □□□□液晶モニタ)の初期化
Lcd_Custom_Config(&PORTB,4,5,6,7,&PORTA,1,0,6);
Lcd_Custom_Cmd(LCD_CURSOR_OFF);
Lcd_Custom_Out(1, 1, "ThunderRange V1");
Delay_ms(1000);
Lcd_Custom_Cmd(LCD_CLEAR);
// 割り込み(全体)の設定
INTCON.PEIE = 1;
INTCON.GIE = 1;
//
LED = OFF;          // LED off!
flag = IDLE;
cnt = 0;
//
while(1) {
    // 開始を判断する。
    if ((flag != START) && (START_SW == 0)) {
        cnt = 0;
        flag = START;    // count start!
        LED = ON;        // LED on!
        Lcd_Custom_Out(1, 1, "Start!");
    }
    // 停止を判断する。
    if (flag == STOP) {
        flag = IDLE;
        LED = OFF;      // LED off!
        Lcd_Custom_Out(1, 1, "Stop! ");
        //
        display(cnt);
    }
    // 開始中はカウント値を表示する。
    if (flag == START) {
        display(cnt);
    }
}
}

//*****
*
```

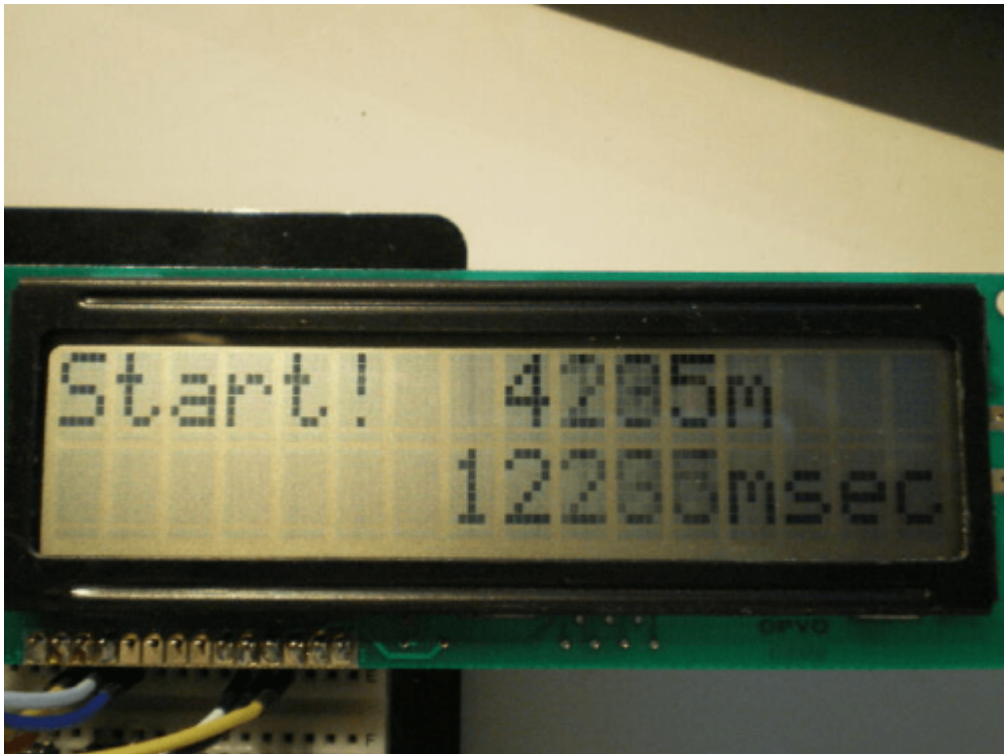
動作原理



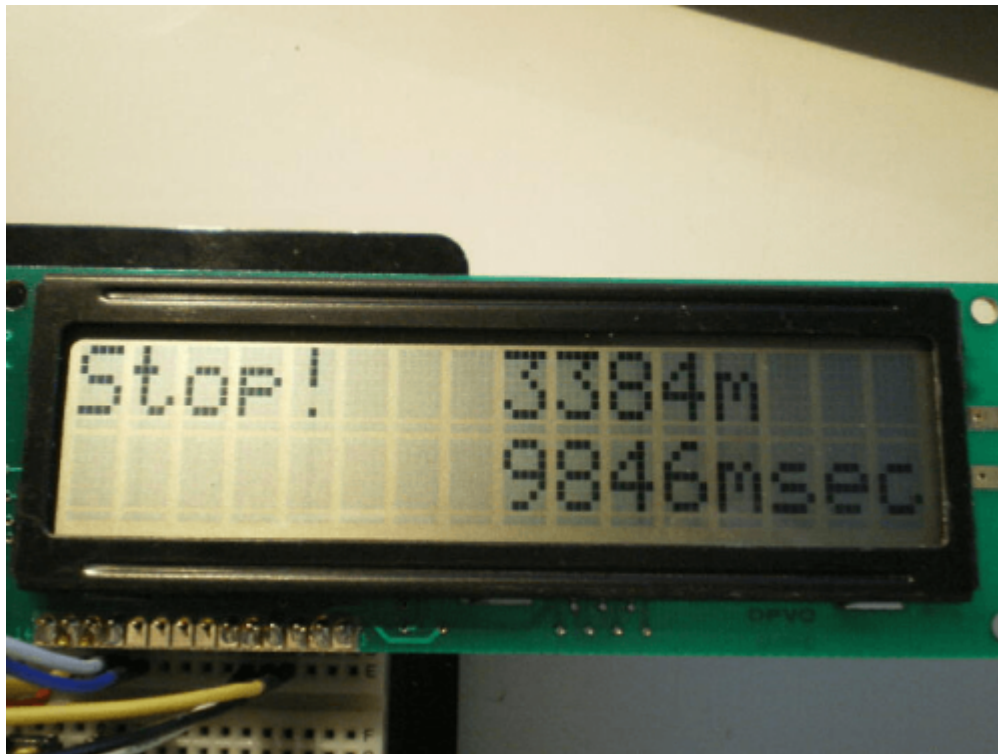
雷が光ったら、開始スイッチ(緑色)を押します。数秒後に雷鳴が聞こえたら、停止スイッチ(黄色)を押します。



測定時は、LEDが点灯し、LCDに経過時間と落雷地点までの距離が表示されます。高速に数値が変化しますので、目視での判



断は難しいです。停止するとLEDが消灯し、LCDに経過時間と落雷地点までの距離が表示されます。この例では、落雷してから、雷鳴が聞こえるまでの時間は、約9.8秒で、落雷地点までの距離は、約3.4kmです。



如何ですか? 摂氏温度や湿度を考慮すれば、更に精度を上げることが出来ます。

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:pic16f88:89&rev=1588163361>

Last update: 2025/10/17 14:28

