

簡易緯度/経度表示ユニット(GPS-52D)

概要

GPS(Global Positioning System)を利用して、自分が地球上の何処(緯度/経度)にいるのかを、表示させるユニットを製作しました。

動作原理

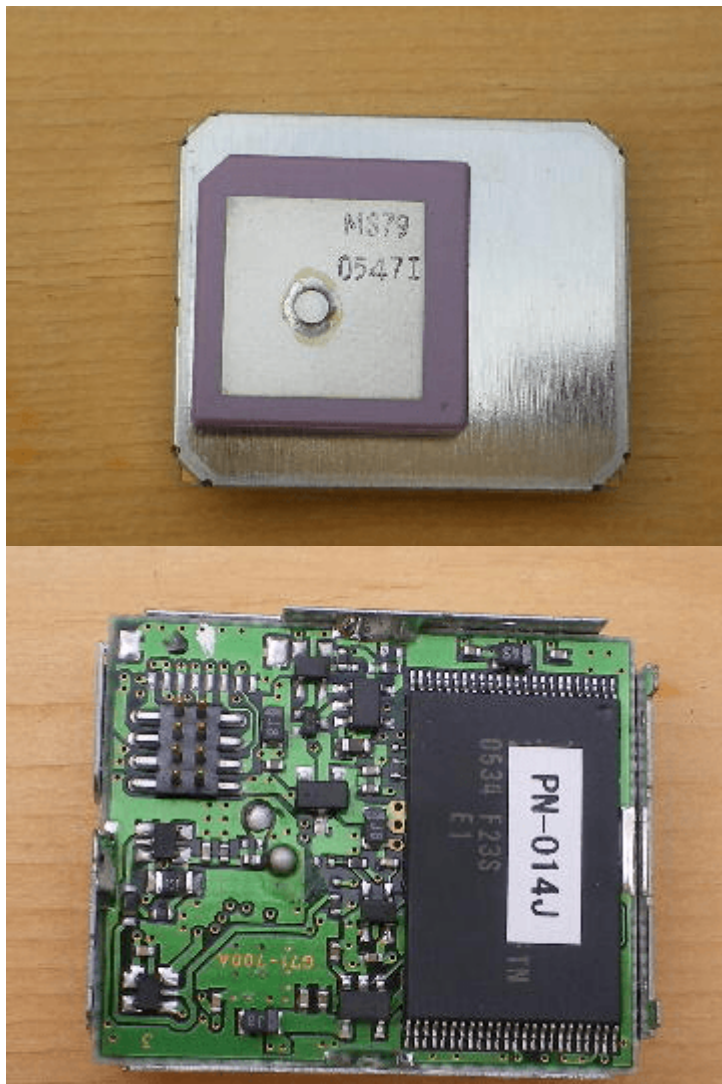
秋月電子で販売されている「GPSレシーバモジュールキット」に含まれるGPSモジュール(GPS-52D)をPICで制御し、経度と緯度をLCDに表示させるユニットです。

<GPSレシーバモジュールキットの特長>

- ポジション社製GPSモジュール(SiRFチップセット)とDサブ/3端子レギュレータ/ADM3202などの部品セットです。
- 緯度/経度/方位/高度/速度/時間などのGPS情報が得られます。
- GPSモジュール:GPS-52D(B)-014アンテナ一体型(電源3.3V)
- RS232C通信用回路部品PC測定ソフトが付属
- デフォルト通信スピード:9600bps
- 出力データ:NMEA-0183準拠
- 外形:25.8(W)x30.8(D)x9.7(H)mm出力端子:1.27mmピッチ8ピン(メスコネクタ付属)

<GPS-52Dの性能仕様>

- 更新周期
1秒(0秒~255秒まで設定変更可能)
- 精度
位置(15m:GPS測位 10m:DGPS測位)
- 初期測位時間
コールドスタート(標準感度製品:50秒、高感度製品:70秒、超高感度製品:60秒)
ウォームスタート(38秒)
ホットスタート(8秒)
- 測地系
東京 WGS-84
- 測位モード
測位不可/2次元測位/3次元測位自動切り替え



GPS-52Dが出力するNMEAは、米国海洋電子機器協会(National Marine Electronics Association)が定めた規格で、受信機とナビゲーション機器の通信に使用されるプロトコルです。中でもNMEA-0183は、GPS受信機とナビゲーション機器の間を、シリアルポートを利用して通信するための規格で、すべての文字がASCIIテキストの「センテンス」で送られます。各データ項目は、カンマ(,)で区切られており、行末にはCR/LFコードが含まれます。

<GPS-52Dが出力するデータ(文字列)>

```
$GSU-50 : Position Co.,Ltd.2003 $Firmware Checksum: 4d50 $TOW: 0 $WK: 1491 $POS: 6378137 0 0
$CLK: 96000 $CHNL:12 $Baud rate: 9600 System clock: 12.277MHz $HW Type: S2AM $Asic Version:
0x23 $Clock Source: GPSClk $Internal Beacon: None $PSRF150,1,*12
```

```
$GPGGA,235948.000,3600.0000,N,13600.0000,E,0,00,99.9,00000.0,M,0000.0,M,000.0,0000*4A
$GPGSA,A,1,,,,,,,,,99.9,99.9,99.9*09
$GPRMC,235948.000,V,3600.0000,N,13600.0000,E,9999.99,999.99,020808,,*2F
$GPVTG,999.99,T,,M,9999.99,N,9999.99,K*59 $GPZDA,235949.000,02,08,2008,,*56
```

以降<\$GPGGA~\$GPZDA>までが、約1秒周期で繰り返されます。

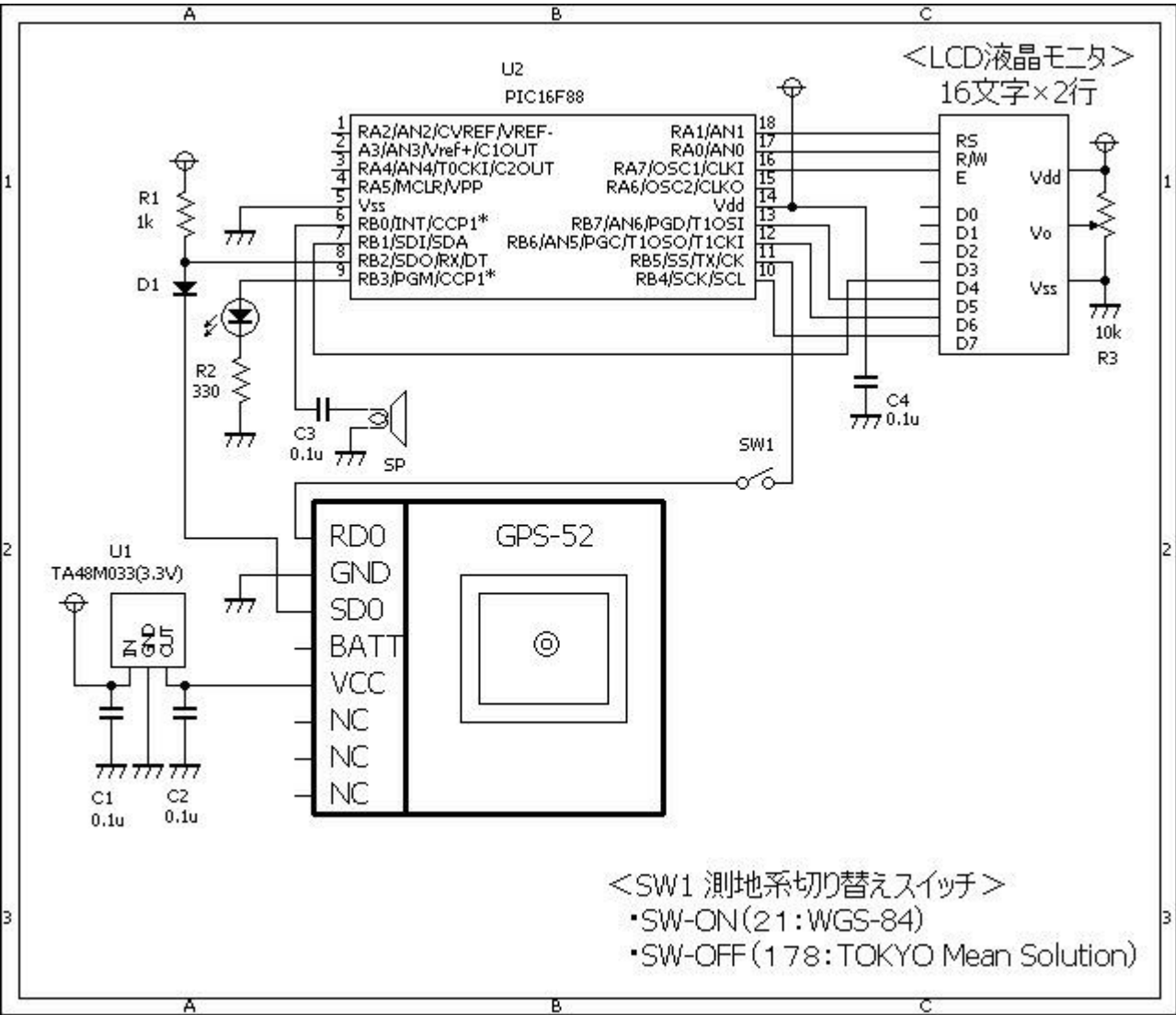
<GGAセンテンスを利用した、経度、緯度の取得>

```
例:$GPGGA,123519.00,4807.038247,N,01131.324523,E,1,08,0.9,545.42,M,46.93,M,5.0,1012*42
123519.00 =測位時刻(UTC)12:35:19.00 **4807.038247,N =緯度48度07.038247分(北緯) **
**01131.324523,E =経度11度31.324523分(東経) ** 1 =GPSのクオリティ; 0 = 受信不能, 1 = 単独測位, 2 =
```

DGPS 08 =受信衛星数 0.9=水平測位誤差 545.42, M =平均海面からのアンテナ高度(m) 46.93, M =WGS-84楕円体から平均海面の高度差(m) 5.0 =DGPSデータのエイジ(秒) 1012 =DGPS基準局のID *42 =チェックサム

<測地系の設定> GPS-52Dは、デフォルトでは、測地系設定値が、[178:TOKYO Mean Solution]になっています。このままでは、世界測地系[21:WGS-84]の経緯度で表わすと、北西方向へ約450mずれてしまうので、起動時に、[21:WGS-84]に変更します[SW1で切り替え可能]

回路図



ソースコード

[nmea0183.c](#)

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

【簡易モニタ】

*/

//*****
*

```
#define      LED      PORTB.F3
#define      CR        0x0D
#define      LF        0x0A
```

//*****
*

```
void  interrupt()
{
    if (INTCON.T0IF == 1) {
        INTCON.T0IF = 0;
    }
    if (PIR1.TMR1IF == 1) {
        LED = ~LED;
        PIR1.TMR1IF = 0;
    }
}
```

//*****
*

```
void  Pwm_Change_DutyEx(unsigned int duty_ratio)
{
    CCP1L = duty_ratio >> 2;
    CCP1CON.F6 = duty_ratio & 0b00000001;
    CCP1CON.F7 = (duty_ratio & 0b00000010) >> 1;
}
```

//*****
*

```
void  Uart_Write_String(char *buf)
{
    static  int    len, i;
    len = strlen(buf);
    for (i = 0; i < len; i++) {
        Uart_Write(buf[i]);
    }
}
```

//*****
*

```
void  GetFieldData(char* msg, short number, char* result)
```

```

{
    short    cnt1, cnt2, pnt1, pnt2, len;
    //
    len = StrLen(msg);
    cnt2 = 0;
    pnt1 = 0;
    pnt2 = 0;
    for (cnt1 = 0; cnt1 < len; cnt1++) {
        if (msg[cnt1] == ',') {
            pnt1 = pnt2;
            pnt2 = cnt1 + 1;
            cnt2++;
            if (cnt2 == number)
                break;
        }
    }
    strncpy(result, msg + pnt1, pnt2 - 1 - pnt1);
    result[pnt2 - pnt1] = 0x00;
}
/*
01234567890123456789012345678901234567890123456789012345678901234567890
1234567890123456789
$GPGGA,235948.000,3600.0000,N,13600.0000,E,0,00,99.9,00000.0,M,0000.0,M
,000.0,0000*4A
*/
//*****
*

void    main()
{
    static    char    rd, buf[50], tmp[20], cnt, cnt2, mode;
    //
    OSCCON = 0b01110000;    // クロックは8Mhz
    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 = 0b01111100;
    TRISB = 0b00000100;
    OPTION_REG = 0b10000111;
    PIE1.TMR1IE = 1;
    PIR1.TMR1IF = 0;
    T1CON = 0b00110001;
    INTCON = 0b01100000;
    //
    Pwm_Init(3000);    // 3Khz
    Pwm_Change_DutyEx(1024 / 2);
    //
    Lcd_Custom_Config(&PORTB, 4, 6, 7, 1, &PORTA, 1, 0, 7);
    TRISA = 0b01111100;
    Lcd_Custom_Cmd(LCD_CURSOR_OFF);
    for (cnt = 0; cnt < 5; cnt++) {
        Lcd_Custom_Out(1, 1, "NMEA0183 Monitor");
    }
}

```

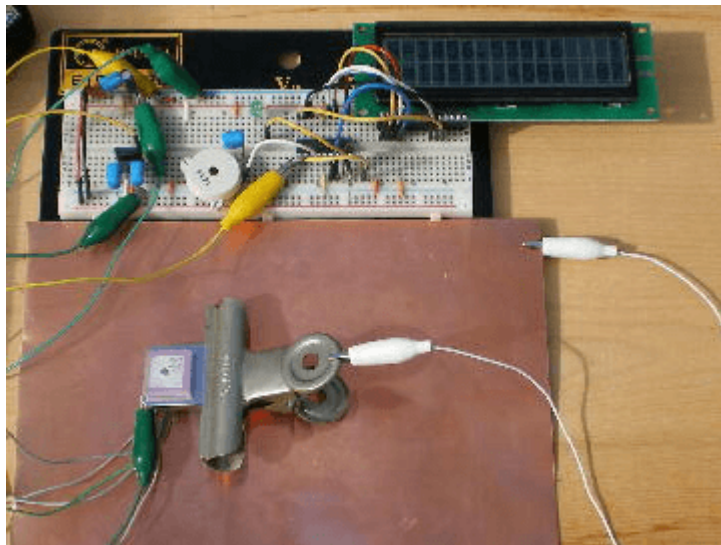
```
Pwm_Start();
Delay_ms(300);
Pwm_Stop();
Lcd_Custom_Cmd(LCD_CLEAR);
Delay_ms(300);
}
//
Usart_Init(9600);
//
INTCON.GIE = 1;    // これ以降の処理で割り込みを許可する。
//
Pwm_Start();
Delay_ms(300);
Pwm_Stop();
//
cnt = 0;
mode = 0;
while (1) {
    if (Usart_Data_Ready() == 0)
        continue;
    rd = Usart_Read();
    buf[cnt] = rd;
    if (cnt < 49)
        cnt++;
    buf[cnt] = 0x00;
    //
    if (rd == LF) {
        if (strncmp(buf, "$GPGGA", 6) == 0) {
            Pwm_Start();
            Delay_ms(100);
            Pwm_Stop();
            //
            GetFieldData(buf, 3, tmp);    //緯度の表示
            Lcd_Custom_Out(1, 1, "N=");
            Lcd_Custom_Out(1, 3, tmp);
            //
            GetFieldData(buf, 5, tmp);    //経度の表示
            Lcd_Custom_Out(2, 1, "E=");
            Lcd_Custom_Out(2, 3, tmp);
            //
            GetFieldData(buf, 7, tmp);    //GPSクオリティの表示
            Lcd_Custom_Out(1, 16, tmp);
            //
            GetFieldData(buf, 8, tmp);    //受信衛星数
            Lcd_Custom_Out(2, 15, tmp);
        }
        cnt = 0;
        if (mode == 0) {
            Usart_Write_String("$PSRF106,21*0F\r\n");    //測地
```

系WGS-84に設定

```
mode = 1;  
}  
}  
}  
}  
}  
  
//*****  
*
```

動作確認

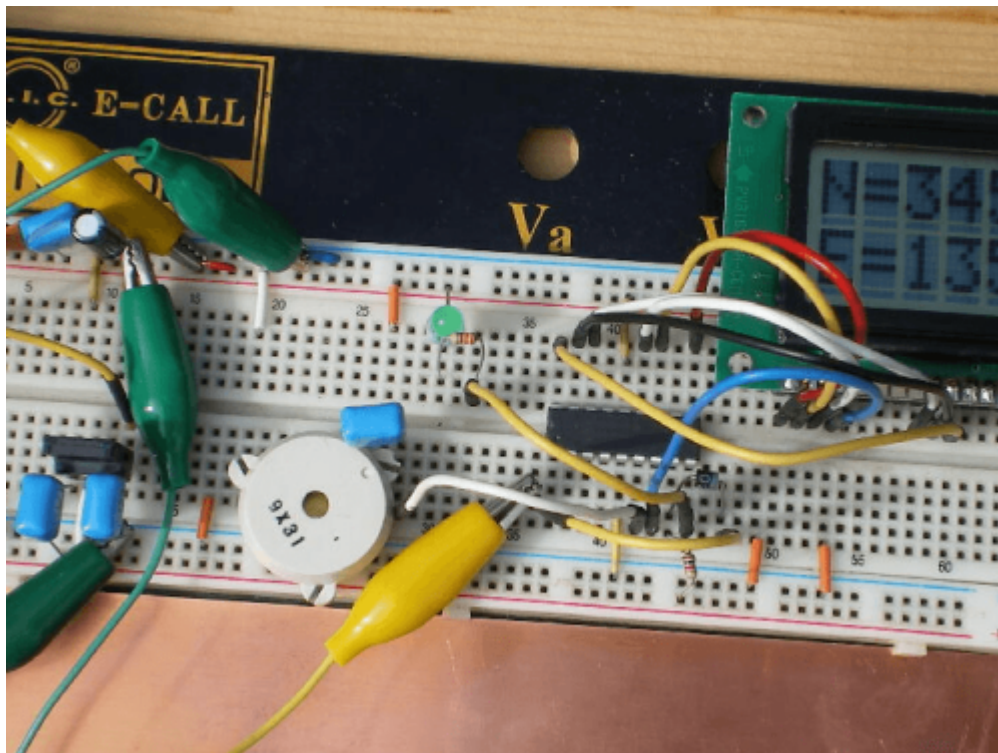
GPS-52Dの下部には、銅テープやアルミ箔でGND層を形成することが推奨されていたので、手持ち



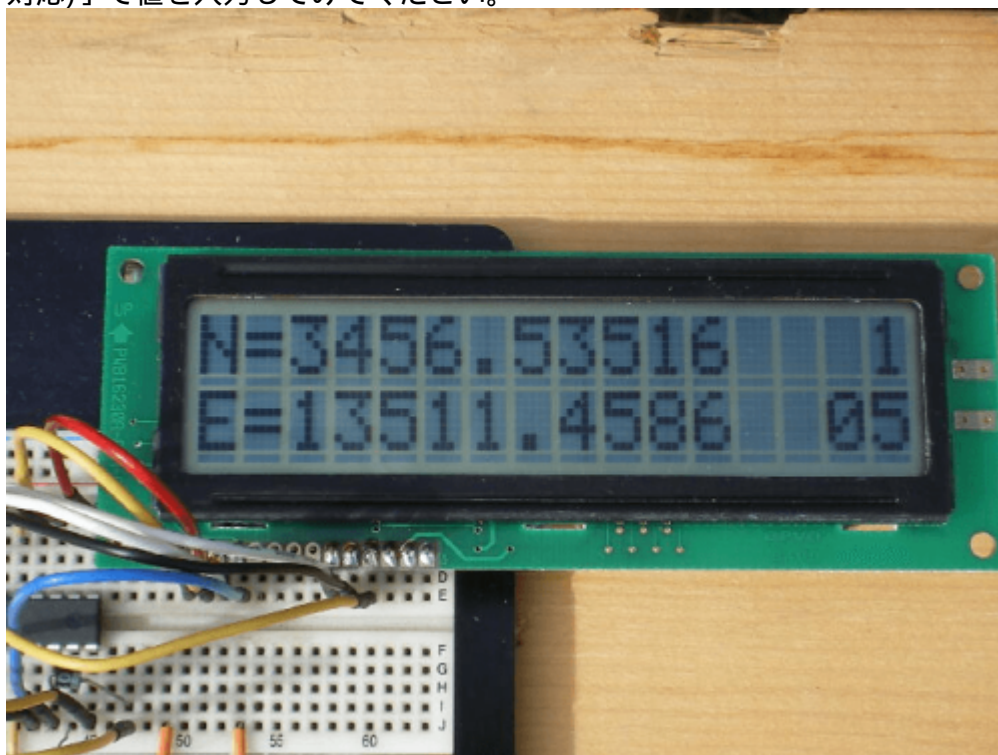
の銅板 を使いました。



左側から順に、3.3Vの3端子レギュレータ、圧電スピーカ、LED、PIC16F88、LCD周りの結線です。



左上:緯度 左下:経度 右上:
受信クオリティ 右下:受信衛星数(最高8を示しました) 表示された値が正しいかどうかを確認する方法の
一つとして、住所やランドマーク名から経度、緯度を検索可能な「[Geocoding](#)(世界測地系(WGS84)に
対応)」で値を入力してみてください。



如何ですか? これをコンパ
クトに組み立て、持ち歩き可能にすれば、自分の歩行軌跡がわかりますね。{😊}!

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

Permanent link:
<http://deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic16f88:90&rev=1588163899>

Last update: 2025/10/17 14:28



