

# GPS時計(調整不要)

## 概要

今回のタイトルは、電波時計の間違いではありません。以前に、緯度と経度を表示する、簡易緯度/経度表示ユニット(GPS-52D)を製作しましたが、これを応用して、調整が不要な『GPS時計』を製作しました。

## 動作原理

簡易緯度/経度表示ユニット(GPS-52D)を、次の3項目について修正しました。

- ハードウェアは、高速に処理するために、内臓クロック8MHzを外付けクロック20MHzに変更しました。
- ソフトウェアは、GPS-52からのデータを受信する処理の構造を、大幅に見直(拡張性大)しました。
- GGAセンテンスに含まれる、測位時刻(UTC/協定世界時)から日本標準時(JST)を求めました。

日本標準時(JST)は、独立行政法人である情報通信研究機構の原子時計で求めた協定世界時(UTC)を、9時間進めることにより求めることができます[(JST=UTC+9時間)]

※UTC(Coordinated Universal Time[協定世界時]) ※JST(Japan Standard Time[日本標準時])

<GGAセンテンスを利用した[UTC]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 = 平均海水面からのアンテナ高度(m) 46.93, M = WGS-84楕円体から平均海水面の高度差(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

## 回路図



```

#define      LF                0x0A

#define      HEADER_SIZE      8
#define      DATA_SIZE       12

//*****
*

unsigned  short  flg, len;
unsigned  short  hd[HEADER_SIZE], utc[DATA_SIZE],
latitude[DATA_SIZE], longitude[DATA_SIZE];
unsigned  short  quality[4], satellites[4];

/*
Format:
$GPGGA,<1>,<2>,<3>,<4>,<5>,<6>,<7>,<8>,<9>,M,<10>,M,<11>,<12>,*<13><CR>
<LF>
Example:
$GPGGA,104549.04,2447.2038,N,12100.4990,E,1,06,01.7,00078.8,M,0016.3,M,
,*5C<CR><LF>
*/

void  interrupt()
{
    unsigned  short  rd;
    //
    LED = ON;
    while (Usart_Data_Ready()) {
        rd = Usart_Read();
        if (rd == LF) {
            len = 0;
            continue;
        }
        //
        switch (flg) {
        case 0:
            hd[len] = rd;
            len++;
            if (len == 7) {
                len = 0;
                if (strncmp(hd, "$GPGGA,", 7) == 0) {
                    flg = 1;
                }
            }
            break;
        case 1:          //UTC(Coordinated Universal Time 協定世界時)、日本
            標準時(Japan Standard Time, JST)+9時間の時差
            if (rd == ',') {
                utc[len] = 0x00;
                len = 0;
                flg = 2;
            }
        }
    }
}

```

```
    } else {
        utc[len] = rd;
        len++;
    }
    break;
case 2:
    if (rd == ',') {
        latitude[len] = 0x00;
        len = 0;
        flg = 3;
    } else {
        latitude[len] = rd;
        len++;
    }
    break;
case 3:
    if (rd == ',') {
        len = 0;
        flg = 4;
    }
    break;
case 4:
    if (rd == ',') {
        longitude[len] = 0x00;
        len = 0;
        flg = 5;
    } else {
        longitude[len] = rd;
        len++;
    }
    break;
case 5:
    if (rd == ',') {
        len = 0;
        flg = 6;
    }
    break;
case 6:
    if (rd == ',') {
        quality[len] = 0x00;
        len = 0;
        flg = 7;
    } else {
        quality[len] = rd;
        len++;
    }
    break;
case 7:
    if (rd == ',') {
        satellites[len] = 0x00;
```

```
        len = 0;
        flg = 8;
    } else {
        satellites[len] = rd;
        len++;
    }
    break;
}
}
LED = OFF;
} //~

//*****
*

void Usart_Write_String(char *str)
{
    unsigned short i;

    i = 0;
    while (str[i] != 0x00) {
        USART_Write(str[i]);
        i++;
    }
} //~

//*****
*

unsigned short tmp[12], buf[12];

void secConv(char* msg) // dddmm.mmmm→ddd.mmssss やまおたく形式)
{
    char *p;
    float f;
    int i;
    //
    p = strchr(msg, '.');
    f = atof(p);
    f = f * 6000.0;
    WordToStr(f, tmp);
    if (tmp[1] == ' ')
        tmp[1] = '0';
    if (tmp[2] == ' ')
        tmp[2] = '0';
    if (tmp[3] == ' ')
        tmp[3] = '0';
    *(p) = *(p - 1);
    *(p - 1) = *(p - 2);
    *(p - 2) = '.';
    *(p + 1) = 0x00;
}
```

```
    strcat(p + 1, &tmp[1]);
}

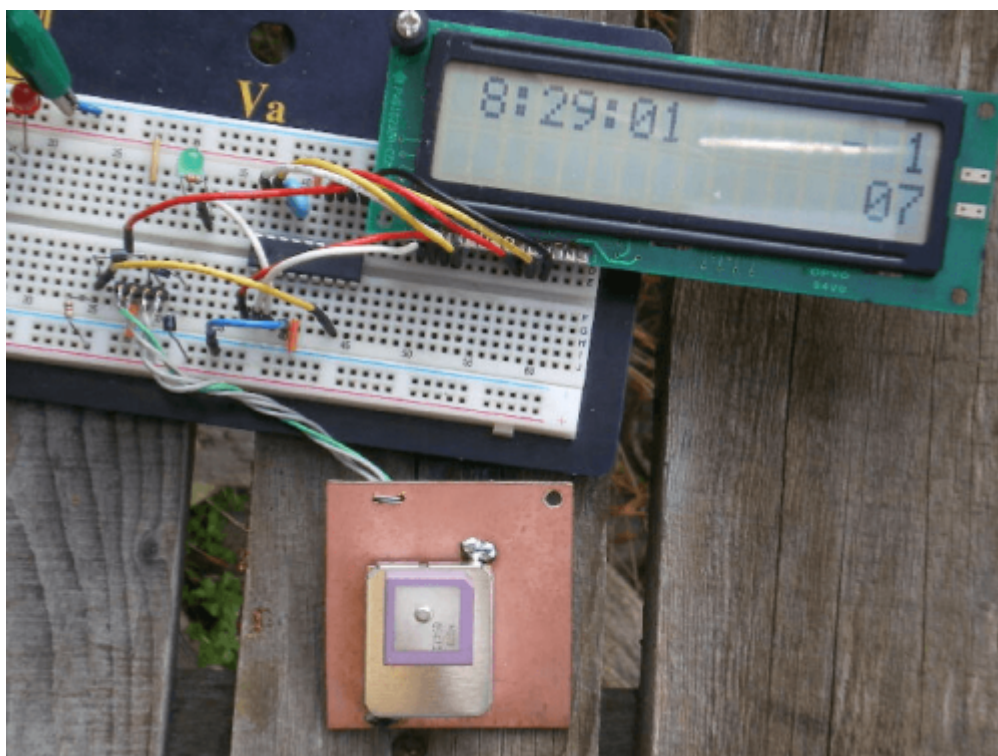
//*****
*

void main()
{
    short mode, i;
    //
//    OSCCON = 0b01110000; // クロックは8Mhz
CMCON = 0b00000111; // コンパレータは使用しない。
ANSEL = 0b00000000; // □□□変換は使用しない。
// ポートを初期化する。
//ポートの設定
TRISA = 0b11100000;
TRISB = 0b00000100;
//変数の初期化
flg = 0;
len = 0;
mode = 1;
//□□□の初期化
Lcd_Custom_Config(&PORTA,3,2,1,0,&PORTB,4,6,7);
Lcd_Custom_Cmd(LCD_CURSOR_OFF);
Lcd_Custom_Cmd(LCD_CLEAR);
//□□□□□の初期化
Usart_Init(9600);
// 割り込みを許可する。
PIE1.RCIE = 1;
PIR1.RCIF = 0;
INTCON.PEIE = 1;
INTCON.GIE = 1;
//
Lcd_Custom_Out(1, 1, "WGS-84 set!");
while (1) {
    if (flg != 8)
        continue;
    flg = 0;
    Usart_Write_String("$PSRF106,21*0F\r\n"); //測地系□WGS-84に設定
    break;
}
Lcd_Custom_Cmd(LCD_CLEAR);
//
while (1) {
    // $GPGGA データを受信していれば、内容を表示する。
    if (flg == 8) {
        if ((SW == 1) && (mode == 0)) {
            mode = 1;
            Lcd_Custom_Cmd(LCD_CLEAR);
        }
    }
}
```

```
    if ((SW == 0) && (mode == 1)) {
        mode = 0;
        Lcd_Custom_Cmd(LCD_CLEAR);
    }
    //
    Lcd_Custom_Chr(1, 14, 0xFF);
    //
    if (mode == 1) {
        secConv(latitude);
        Lcd_Custom_Out(1, 1, "N= ");
        Lcd_Custom_Out(1, 4, latitude);
        //
        secConv(longitude);
        Lcd_Custom_Out(2, 1, "E=");
        Lcd_Custom_Out(2, 3, longitude);
    } else {
        tmp[8] = 0x00;
        tmp[7] = utc[5];
        tmp[6] = utc[4];
        tmp[5] = ':';
        tmp[4] = utc[3];
        tmp[3] = utc[2];
        tmp[2] = ':';
        utc[2] = 0x00;
        i = atoi(utc) + 9;
        i = i < 24 ? i : i - 24;
        ByteToStr(i, buf);
        tmp[1] = buf[2];
        tmp[0] = buf[1];
        Lcd_Custom_Out(1, 1, tmp);
    }
    //
    Lcd_Custom_Out(1, 16, quality);
    //
    Lcd_Custom_Out(2, 15, satelllites);
    //
    Lcd_Custom_Chr(1, 14, '-');
    //
    len = 0;
    flg = 0;
}
}
} //~!

//*****
*
```

## 動作確認



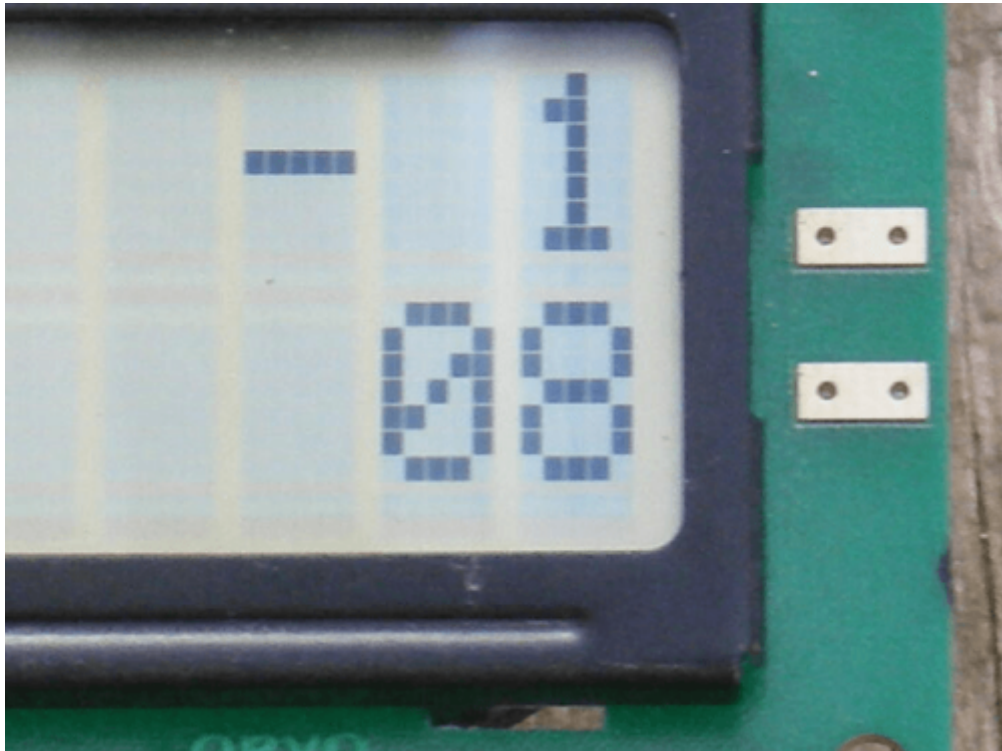
GPS-52Dの下部には、銅テープやアルミ箔でGND層を形成することが推奨されていたので、手持ちの銅板に半田付けしました。





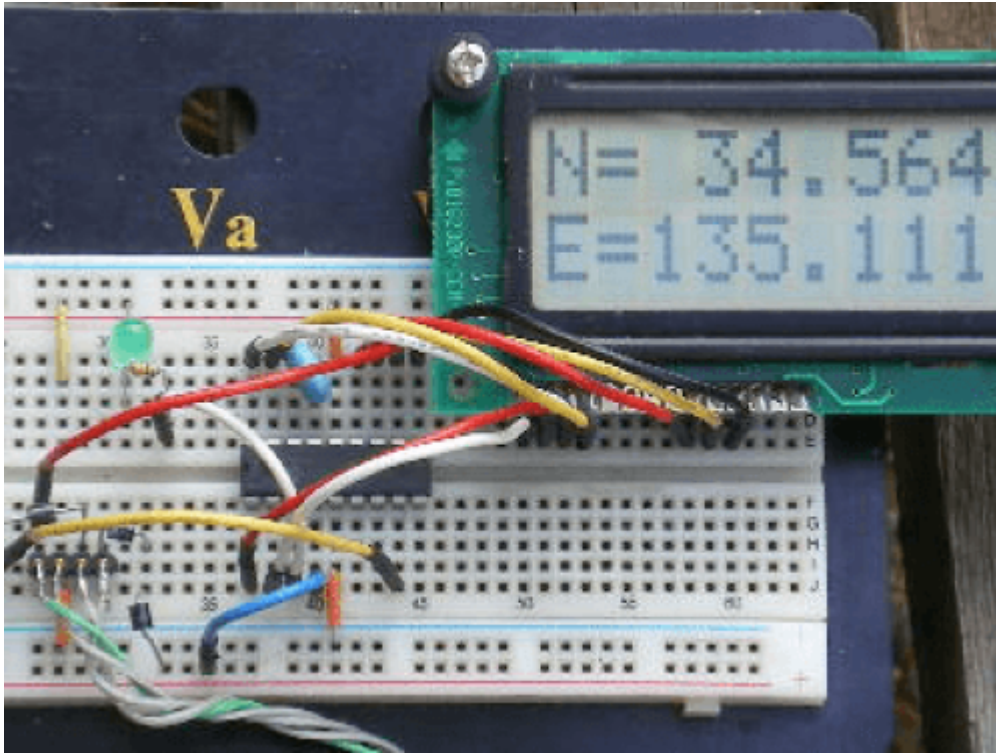
時刻を表示させました。

GPSのクオリティと受信衛星数を表示させました。“-”は、状態を見るために参考として表示させて



いるだけです。

緯度と経度を表示させました。表示フォーマットは、『やまおたく形式/ddd.mmssss』です。左側:アイドリング時 右側:アイドリング後



如何ですか? 手持ちの電波

時計とも比較しましたが、殆ど誤差は無いようですね。{ 😊 }! GPSモジュールを使用するので、電波時計を製作するよりは容易なのは。。。

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:pic16f88:102&rev=1588204765>

Last update: 2025/10/17 14:27

