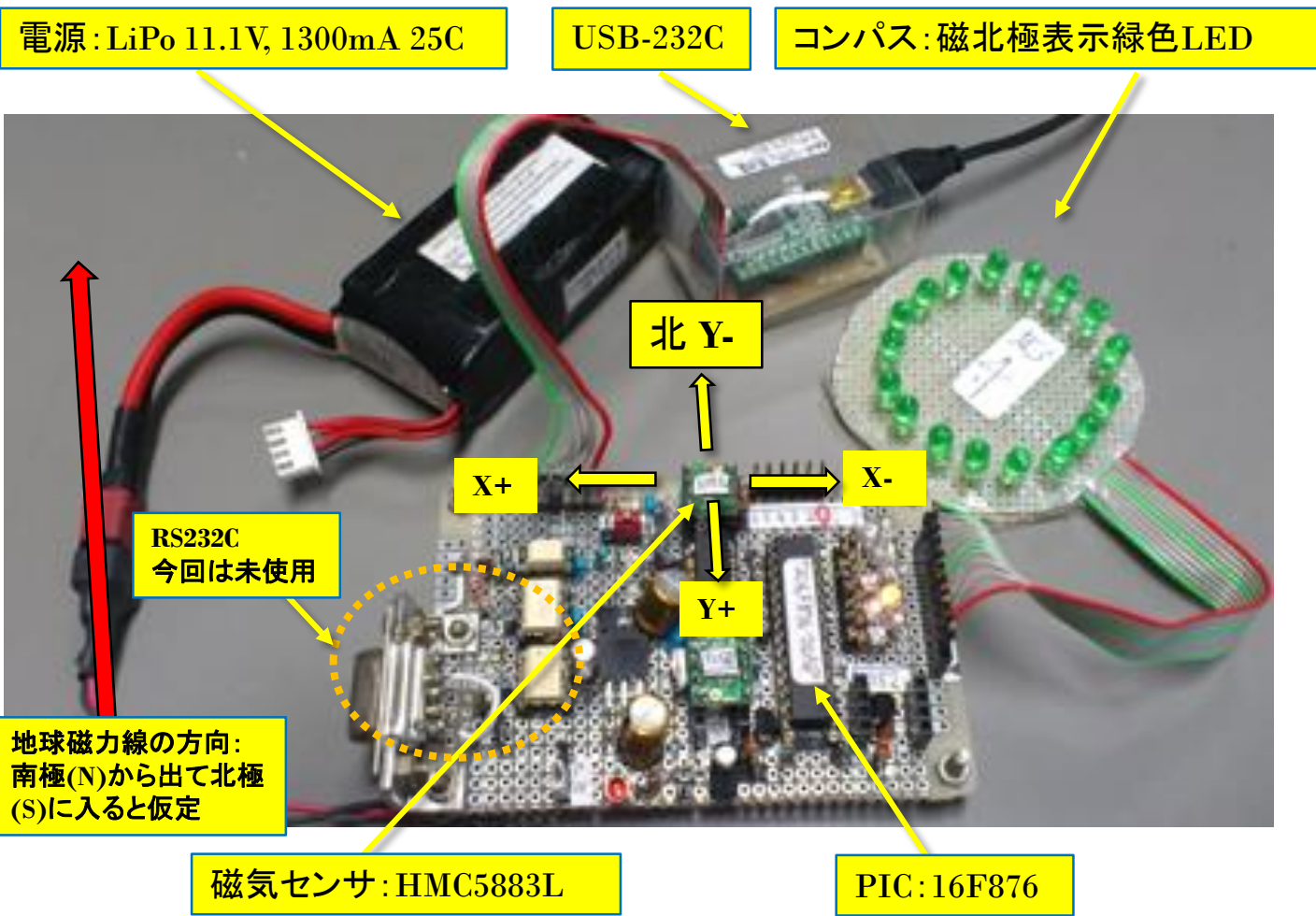


磁気センサを用いた磁気コンパスの試作 ~ データ送信の無線化まで

2014.12.04 技術交流会にて紹介
2014.11.xx 技術交流開始



```
COM4:19200baud - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) 漢字コード(K) ヘルプ(H)
00089, 00412, -00376, 00167
Hai This is test of HMC5883L Magnetic Compass 2014.11.28
Now Let's Start !
XX YY Z degree= 57.3*atan(y../x..)
00254, 00146, -00389, 00119
```

TeraTerm を用いて測定された磁束密度を表示する。この値より、磁北極 (S極)からのセンサの回転角 (偏角) を計算する。磁束密度の単位はガウス*の仕様となっている。磁気測定操作と偏角計算操作はPIC16F876が行う。計算式は各象限によって異なる。計算は、偏角 (0~359度) に応じ、これを8等分し、対応するLEDを点灯させる。点灯箇所は常に磁北極とした。これにより、実験者は磁北極を認識できる (コンパス機能)。なお、コンパスは2次元(x, y)表示とした。そのため、x, y成分のデータのみを使用する。

東京で使用する時は、センサのy軸を“北”に向け、北方向yをおよそ49度 (伏角) 下に傾ける。その時yは最大となり最大磁束密度を得る。yの値に着目して、センサyを上下に振らせればおよその伏角を推定できる。ちなみに磁石式コンパスは磁石の“南”におもりをつけることによって物理的平衡を得る。一方、東京の磁北極と真北とのずれ(磁気偏角)はおよそ7°となる。なお偏角値は右回りの値とし、常に+の極性を示す。測定サイクルはおおよそ10Hzとなった。

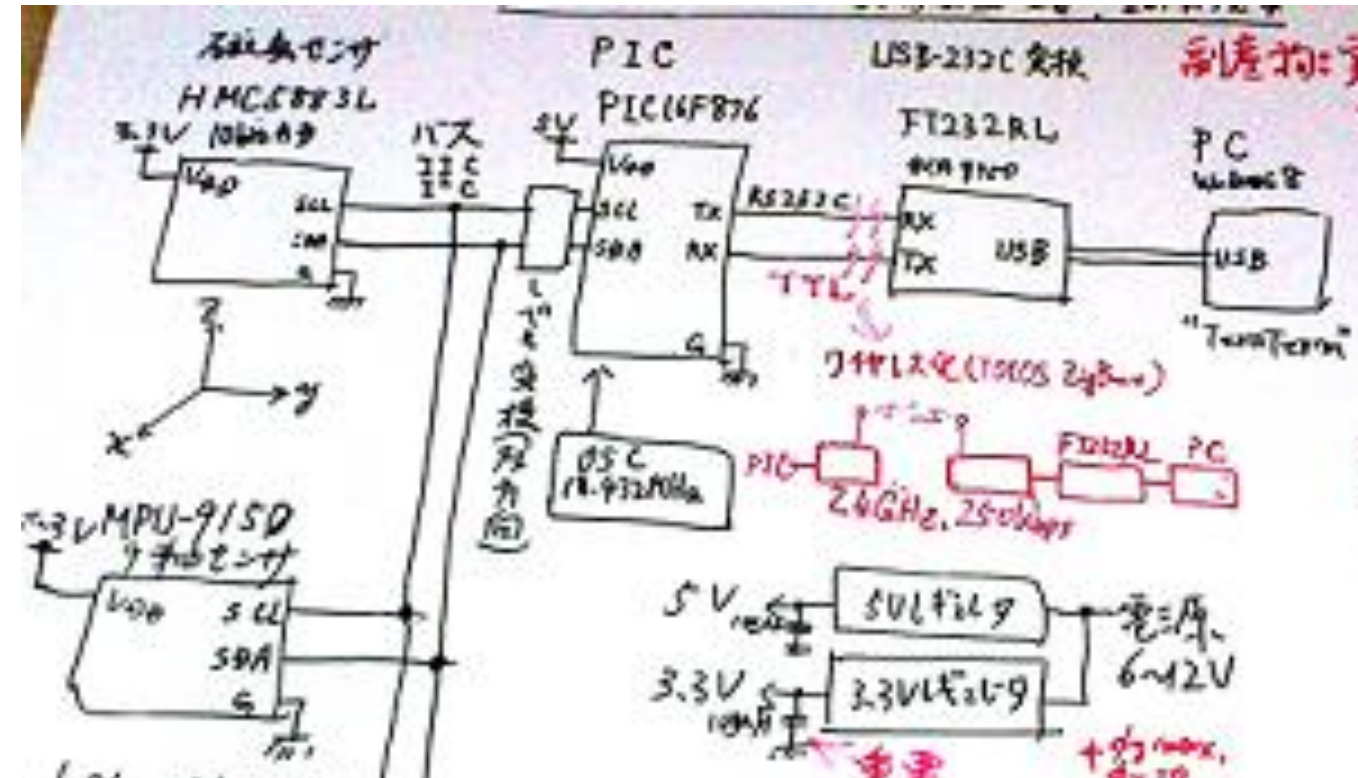
* [G] CGS系による磁束密度の単位。 SI単位系では、テスラ[T]。測定レンジを考慮すれば、『ガウスメータ』となる。

メモ:
地磁気の本質は回転電流である。ならば、地磁気の変化を測定し、傾向を把握することでマントル対流と関連付けて地震を予知できるのではないかと

磁気センサの極性:
各軸でマイナスからプラス方向に向かう磁力線の時、正の値を得る。上図の状態、Xはゼロ、Yは負の最大値となり、センサを少し右 (時計回り) に回転した場合は、X, Yともに、負の値を得る。計算プログラムは、極性変換の操作を行ってから atan() の計算をする。予め、センサの取り付け回転位置を180度変更すればその必要はない。

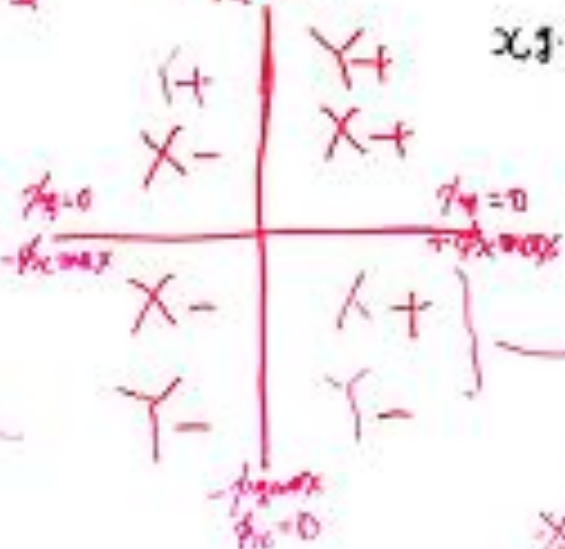
注意:この情報によって第三者が、何らかの損害を受けてもこの情報の提供者は、いかなる損害賠償責任も負わない。
秋月電子 ⇒ 秋月電子 様 (敬称略)

プログラムソースはPWMなど、余計なコードを含む。コンパイラはCCS C を用いた。参考:
http://www.s-yamaga.jp/nanimono/chikyuu/chi_jiki-01.htm
<http://www.pasco.co.jp/recommend/word/word046/>

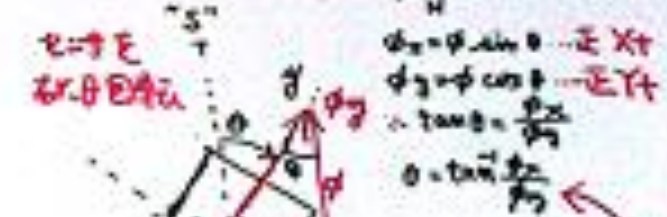
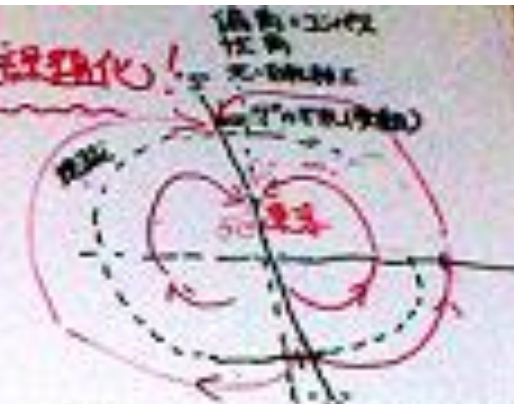


Clear 3bit: 5bit
 Uw 3 = 16bit
 M 3 = 13bit
 9s

センサの出力
 よび限界E
 判別可能



副産物: 実験の自動化!

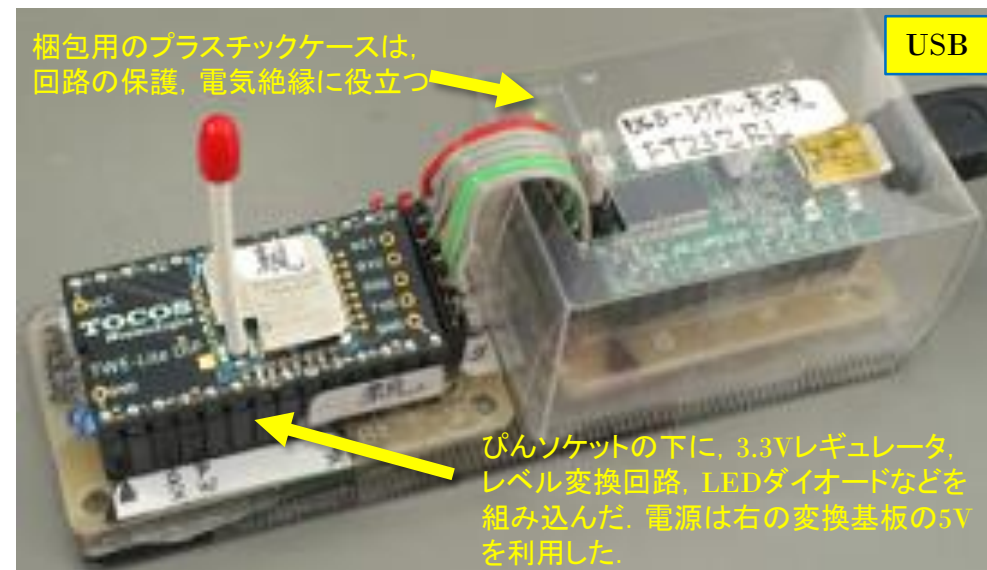


* 各象限の式が異なる

みにくくて、すみません



磁気(磁束密度)の計測と磁気偏角計算・データ送信部

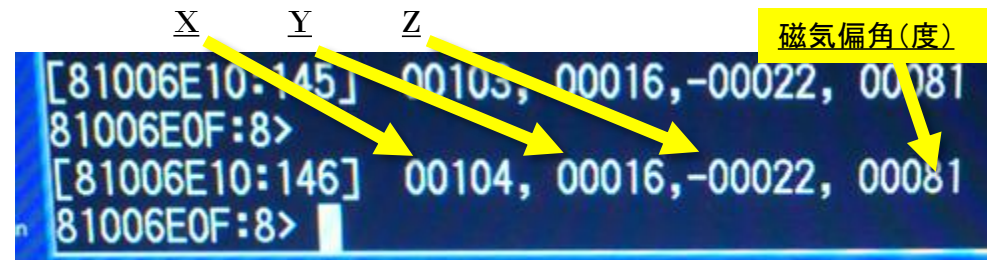


梱包用のプラスチックケースは、回路の保護、電気絶縁に役立つ

USB

ピンソケットの下に、3.3Vレギュレータ、レベル変換回路、LEDダイオードなどを組み込んだ。電源は右の変換基板の5Vを利用した。

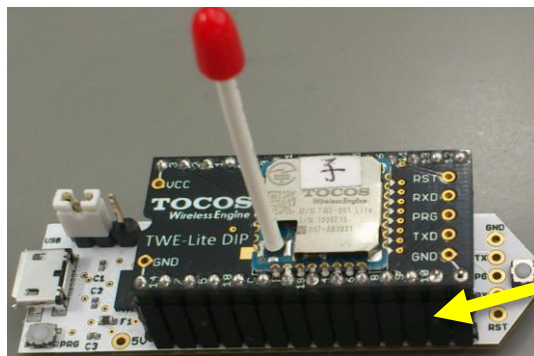
データ受信部



X Y Z 磁気偏角(度)

```
[81006E10:145] 00103, 00016, -00022, 00081
81006E0F:8>
[81006E10:146] 00104, 00016, -00022, 00081
81006E0F:8>
```

TOCOS:Cチャットモード(透過モードを検討中) 受信&表示



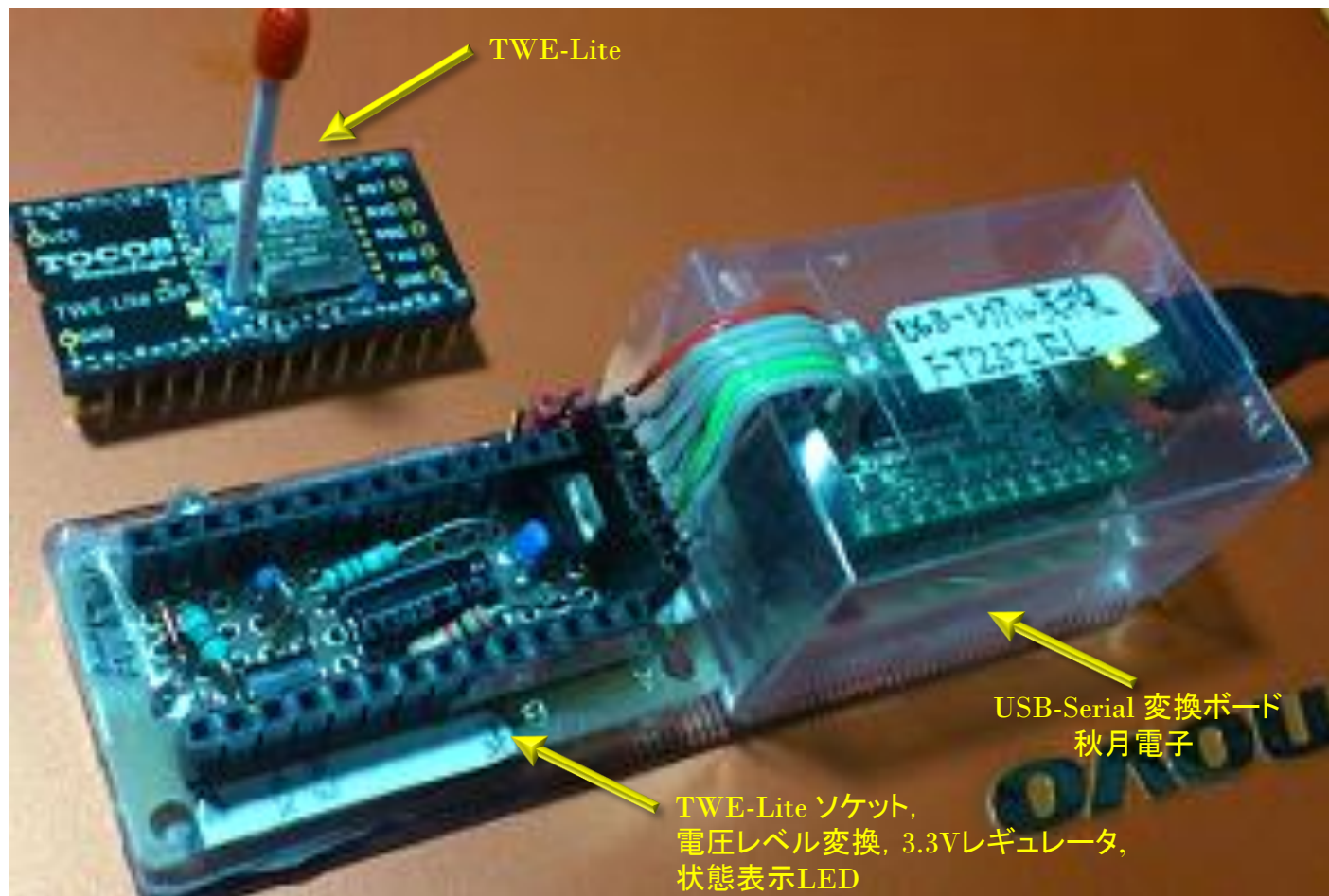
秋月電子より入手したピンソケットは、任意のピン数でカット可能であり、軽い挿入タッチで接触も良く、本体のピンを曲げてしまうことも少ない。このように背の高いタイプを用いると、本体の下に、若干の回路を組み込める。

トワイライタ購入により、右上のF232RL変換ボードは**不要**となった。

トワイライタによる TOCOS:TWE-Lite DIPのプログラム書き換え

展望:

無線化により、人が立ち入れないような危険な場所でも安全に計測が可能となる。磁気センサーを放射線センサーに換装すれば、遠隔地から線量をモニタできる。ただし、計測・送信部が放射線からの影響を受けない、あるいは非常に少ない条件が必要となる。TWE-Liteは、中継機能もあるので、数キロメートル離れた場所からのモニタが可能であろう。さらにPCをネットに接続すれば世界中から可能となる。また安価に実現できることも特徴と言える。ただし、バッテリーの電気量や設置の問題は残る。



TWE-Lite 通信ボード（自作）：トワイライタがあれば**不要**

磁気偏角のLCDへの数値表示

2014.12.30

(未完成)



AQM0802A-RN-GBW

<http://www.geocities.jp/zattouka/GarageHouse/micon/MPLAB/12F1822/LCD/LCD.htm>

<http://kingyonull.blogspot.jp/2013/06/i2caqm0802a-rn-gbw.html>

<http://akizukidenshi.com/download/ds/xiamen/AQM0802.pdf>

磁気コンパスの プログラム (データ送信の無線化 を行っても改変不要)

```
#include<16F876.h>
#include<math.h>
#define HS,NOWDT,NOPROTECT,PUT,BROWNOUT,NOLVP
#define use delay(clock=18432000)
#define use i2c(MASTER,SDA=PIN_C4,SCL=PIN_C3,FAST,NOFORCE_SW)
#define use RS232(PARITY=N, BAUD=19200,XMIT=PIN_C6, RCV=PIN_C7)
```

```
// i2c slave addresses
#define HMC5883L_WRT_ADDR 0x3C
#define HMC5883L_READ_ADDR 0x3D
```

```
// Register addresses
#define HMC5883L_CFG_A_REG 0x00
#define HMC5883L_CFG_B_REG 0x01
#define HMC5883L_MODE_REG 0x02
#define HMC5883L_X_MSB_REG 0x03
```

```
// mtsw 21014.11.11
#define HMC5883L_STR 0x09 //Status Register A
#define HMC5883L_IRA 0x0A //Identification Register A = 0x48
#define HMC5883L_IRB 0x0B //Identification Register B = 0x34
#define HMC5883L_IRC 0x0C //Identification Register C = 0x33
```

```
void beepi(int x){
    while(x>0){
        output_high(PIN_B6);
        output_high(PIN_B7);
        delay_ms(10);
        output_low(PIN_B6);
        output_low(PIN_B7);
        delay_ms(10);
        x--;
    }
}
```

```
void rot(int cnt){ // Like Night Rider
    long PW ;
    int8 posi;
    int speed;
```

```
    speed = 40;
    while ( 0 < cnt){ // Rotate light.
        posi = 0x01;
        while(0<posi){
            output_b(posi); // ON
            delay_ms(speed);
            posi <<= 1;
        }
        posi = 0x08;
        while(0<posi){
            output_b(posi); // ON
            delay_ms(speed);
            posi >>= 1;
        }
        cnt--;
    }
}
```

```
void pwm(int cnt){ // Fuwa Fuwa --- !
    long PW ;
    int speed;

    speed = 110;
    while ( 0 < cnt){ // Grows light.
        PW = 0;
        while(PW<10000){
            output_b(0xff); // ON
            delay_us(PW);
            output_b(0x00); // OFF
            delay_us(10000-PW);
            PW = PW + speed;
        }
    }
```

```
//-----
// Low level routines for HMC5883L
//-----
void hmc5883l_write_reg(int8 reg, int8 data){
    i2c_start();
    i2c_write(HMC5883L_WRT_ADDR);
    i2c_write(reg);
    i2c_write(data);
    i2c_stop();
}
```

```
//-----
int8 hmc5883l_read_reg(int8 reg){
    int8 retval;
    i2c_start();
    i2c_write(HMC5883L_WRT_ADDR);
    i2c_write(reg);
    i2c_stop(); // mtsw
    i2c_start();
    i2c_write(HMC5883L_READ_ADDR);
    retval = i2c_read(0);
    i2c_stop();
    return(retval);
}
```

```
//-----
typedef struct{
    int16 x;
    int16 y;
    int16 z;
}hmc5883l_result;
```

```
// This global structure holds the values read
// from the HMC5883L x,y,z registers.
```

```
hmc5883l_result compass = {0,0,0};
```

```
//-----
void hmc5883l_read_data(void){
    int8 x_lsb;
    int8 x_msb;
    int8 y_lsb;
    int8 y_msb;
    int8 z_lsb;
    int8 z_msb;
    int8 str;

    i2c_start();
    i2c_write(HMC5883L_WRT_ADDR);
    i2c_write(HMC5883L_X_MSB_REG); // Point to X-msb register = 0x03
```

```
    delay_ms(70); // mtsw
    i2c_start();
    i2c_write(HMC5883L_READ_ADDR);
```

```
    x_msb = i2c_read();
    x_lsb = i2c_read();
```

```
    z_msb = i2c_read();
    z_lsb = i2c_read();
```

```
    y_msb = i2c_read();
    y_lsb = i2c_read(0);
```

```
    i2c_stop();
```

```
// Combine high and low bytes into 16-bit values.
compass.x = make16(x_msb, x_lsb);
compass.y = make16(y_msb, y_lsb);
compass.z = make16(z_msb, z_lsb);
}
```

```
int disp_val(int16 x){ // Disp Digits With Sign
    int16 d1,d2,d3,d4,d5;
```

```
    if( x == 0xFFFF){
        printf("O FFFF");
        return(-1);
    }
    else{
        if( x & 0x8000){ // Minus ?
            printf("-");
            x = ( x ^ 0xFFFF)+1;
        }
        else{
            printf(" ");
        }
        d5 = x % 10 + 0x30;
        x /= 10;
        d4 = x % 10 + 0x30;
        x /= 10;
        d3 = x % 10 + 0x30;
        x /= 10;
        d2 = x % 10 + 0x30;
        x /= 10;
        d1 = x % 10 + 0x30;
        printf("%1c%1c%1c%1c%1c",d1,d2,d3,d4,d5);
    }
    return(0);
}
```

```

// =====
void main(){
double M, fai;
int16 degree, x, y, z;
double xx, yy, zz;

delay_ms(1000); // A power-on init delay may be needed
pwm(3);
rot(5);
beepi(10);

printf("\r\nHai This is test of HMC5883L Magnetic Compass 2014.11.28");
printf("\r\nNow Let's Start ! Mitsuo Takahashi U.E.C Tokyou");
printf("\r\nXX YY Z degree= 57.3*atan(y../x..)");
delay_ms(1000);
// continuous measurements.
while(1) {
hmc5883l_write_reg(HMC5883L_CFG_A_REG, 0x70); //0x70 CCS=0x70
hmc5883l_write_reg(HMC5883L_CFG_B_REG, 0x80); //0xA0 CCS=0xA0
hmc5883l_write_reg(HMC5883L_MODE_REG, 0x00); //0x00 CCS=0x00
hmc5883l_read_data();

x = (compass.x ^ 0xFFFF); // Reverse Sign
y = (compass.y ^ 0xFFFF); // Reverse Sign
z = compass.z;

xx = x;
yy = y;
if( x & 0x8000 ){
x *= -1;
xx = x;
if( y & 0x8000){
y *= -1;
yy = y;
degree = 57.3*ATAN(yy/xx) + 90;
}
else{
degree = 57.3*ATAN(xx/yy);
}
}
else{
if( x > 0 ){
if( y & 0x8000){
y *= -1;
yy = y;
degree = 57.3*ATAN(xx/yy) + 180;
}
else{
degree = 57.3*ATAN(yy/xx) + 270;
}
}
}
}

printf("\r\n", disp_val(xx);
printf(", ", disp_val(yy);
printf(", ", disp_val(z);
printf(", ", disp_val(degree);

```

```

if( ( 340 < degree ) || ( degree <= 25 ) ){
output_b(0x01);
}
if( ( 25 < degree ) && ( degree <= 70 ) ){
output_b(0x02);
}
if( ( 70 < degree ) && ( degree <= 115 ) ){
output_b(0x04);
}
if( ( 115 < degree ) && ( degree <= 160 ) ){
output_b(0x08);
}
if( ( 160 < degree ) && ( degree <= 205 ) ){
output_b(0x10);
}
if( ( 205 < degree ) && ( degree <= 250 ) ){
output_b(0x20);
}
if( ( 250 < degree ) && ( degree <= 295 ) ){
output_b(0x40);
}
if( ( 295 < degree ) && ( degree <= 340 ) ){
output_b(0x80);
}
}
}
}

```