XYZ加速度(ガル)表示ユニット

概要

日本及びその周辺では、人間が感じない小さな地震まで含めると1年に10万個以上、1日平均300個以上の 地震が発生しているそうです。(気象庁で震源を決める事が出来た地震の数より)

地震の揺れの強さを表すのに用いる加速度の単位は、ガル(Gal)です。 1ガルは、毎秒1cmの割合で速度が増す事(加速度)を示しています。 地震の揺れは、地面に水平に縦、横(南北、東西方向)と上下方向の3方向(XYZ)で解析されます。

そこで今回は、この3方向の加速度を検出する安価なセンサーを入手しましたので、早速、ガルを表示する ユニットを製作してみました。

<震度の階級:地震と加速度の目安>

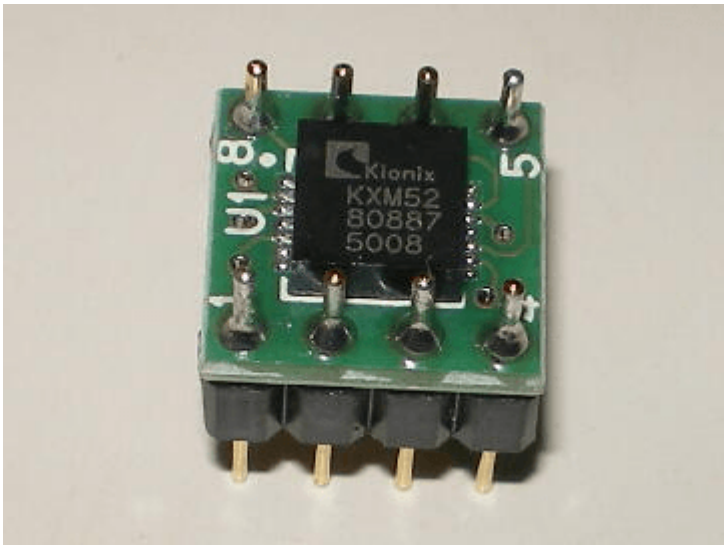
階級	説明	相当加速度
震度0	人体には感じない。地震計に記録される程度。	0~0.8ガル
震度1	静止している人や、特に注意深い人のみが感じる。	0.8~2.5ガル
震度2	大勢の人が感じる程度、戸障子がわずかに動く。	2.5~8.0ガル
震度3	家屋が揺れ電灯等の吊下げものは相当揺れる。	8.0~25ガル
震度4	家屋の揺れは相当激しく花瓶などは倒れ多くの人は戸外に飛び出す。	25~80ガル
震度5	壁に亀裂が走り、煙突/石垣等が破損する程度。	80~250ガル
震度6	家屋の倒壊は30%以下で多くの人は立っていることができない。	250~400ガル
震度7	家屋の倒壊は30%以上で山崩れ/地割れ/断層を生ずる。	400ガル以上

動作原理

XYZの3方向の加速度を検出するセンサーには、カイオニクス社ローノイズ3軸加速度センサ“KXM52“を使用します。

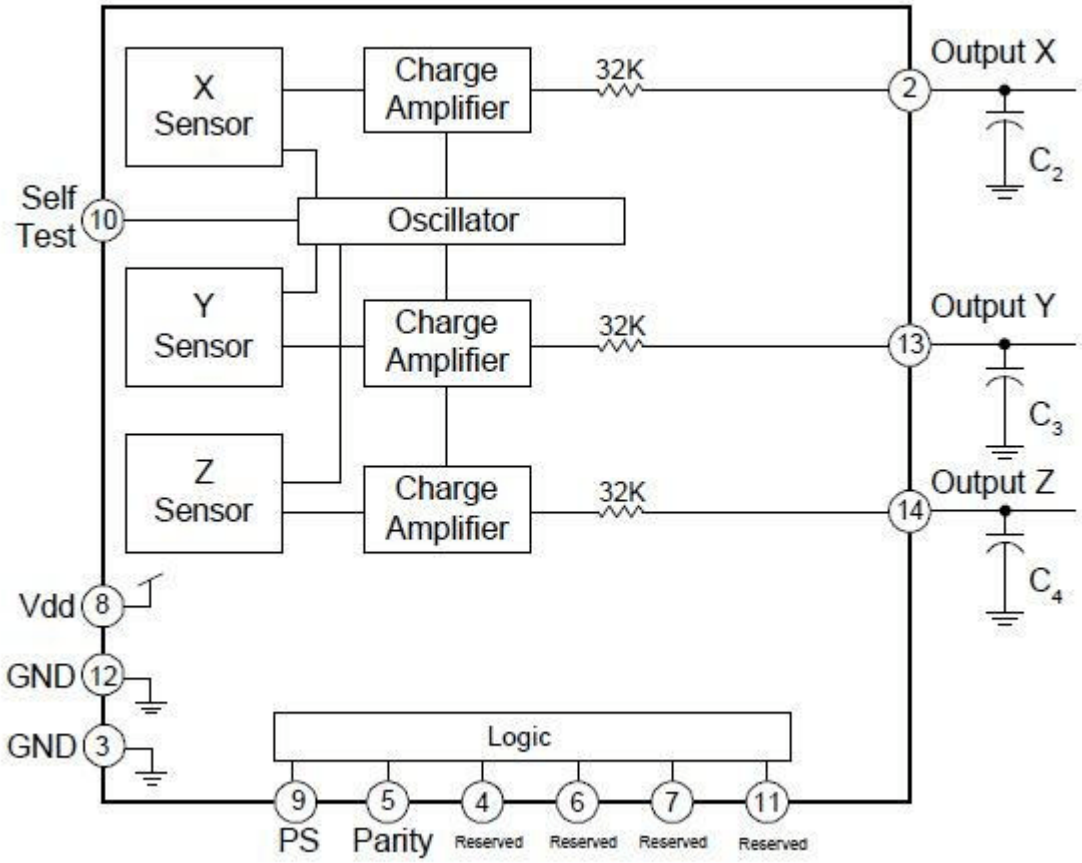
<KXM52の仕様>

- 測定レンジ(2Gal)
- 感度(1000mV/Gal電源5V時)
- オフセット(0Gal出力=2500mV電源5V時)
- 電源電圧(2.7V~5.5V)



<KXM52の概観>

<KXM52のブロックダイアグラム>



<感度**とオフセット**> KXM52**は、**使用**する**電源電圧**により、**感度(出力振幅)、**オフセット**(0Gal出力)**が異なります。**

- 感度(V/g)□Vdd÷5
- **オフセット**(V)□Vdd÷2

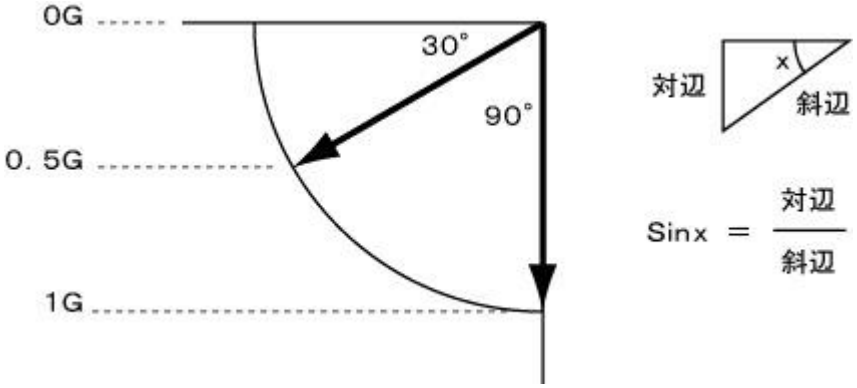
電源電圧	**感度**	**オフセット**
5.5V	1100mV	2.75V
5V	1000mV	2.5V
4V	800mV	2V

電源電圧	感度	オフセット
3.3V	660mV	1.65V
3V	600mV	1.5V
2.7V	540mV	1.35V

使用する電源電圧によって、この表の値を単純に適用するばよいのですが、少し工夫して自動的に求めるよう にしました。

■ オフセット(V)の求め方 KXM52を、“0G”つまり傾斜の無い所に設置し、その時の電圧を取り込んでオフセット値とします。

※参考 地球の重力は垂直方向に1Gです。加速度センサを傾けるとこの重力に反応しますので90 °傾ける と1Gの力が生じます。傾きと重力の関係は次の図のようになります。



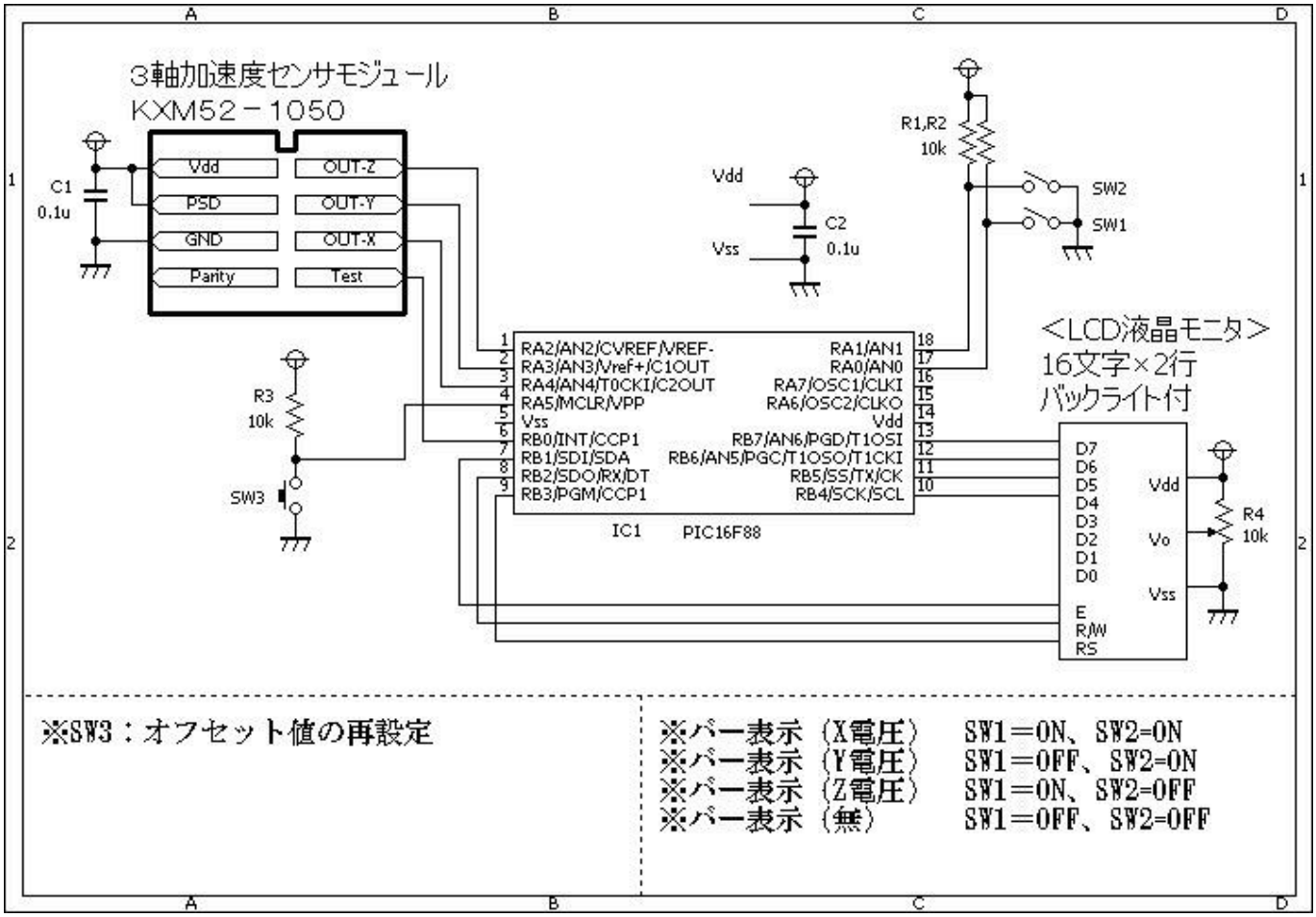
■ 感度(V/g)の求め方 KXM52には、セルフテスト(SelfTest)の端子が出ています。この端子をVddに接続すると現状値に 1Gが加えられた値が出力されます。 例えば、電源電圧=5V□現状値=2.5Vであれば、セルフテスト端子をVddに接続すると□3.5Vに なります。

```
感度=セルフテスト時の値-現状値
```

■ 加速度(Gal)の求め方 感度とオフセットと現状値より、次式で加速度を求めることが出来ます。

```
加速度=(絶対値(現状値-オフセット値))÷感度
```

回路図



ソースコード

acceleration.c

```
//********************************************************************************
*
/*
□□□□□□加速度（ガル）表示器）＞
*/
//********************************************************************************
*

unsigned   long    measurement(unsigned short channel)
{
    static    unsigned    int        cnt;
    static    unsigned    long    ad;
    //
    ad = 0;
    for (cnt = 0; cnt < 10; cnt++) {
        ad += Adc_Read(channel);
    }
    return (ad / 10);
}
```

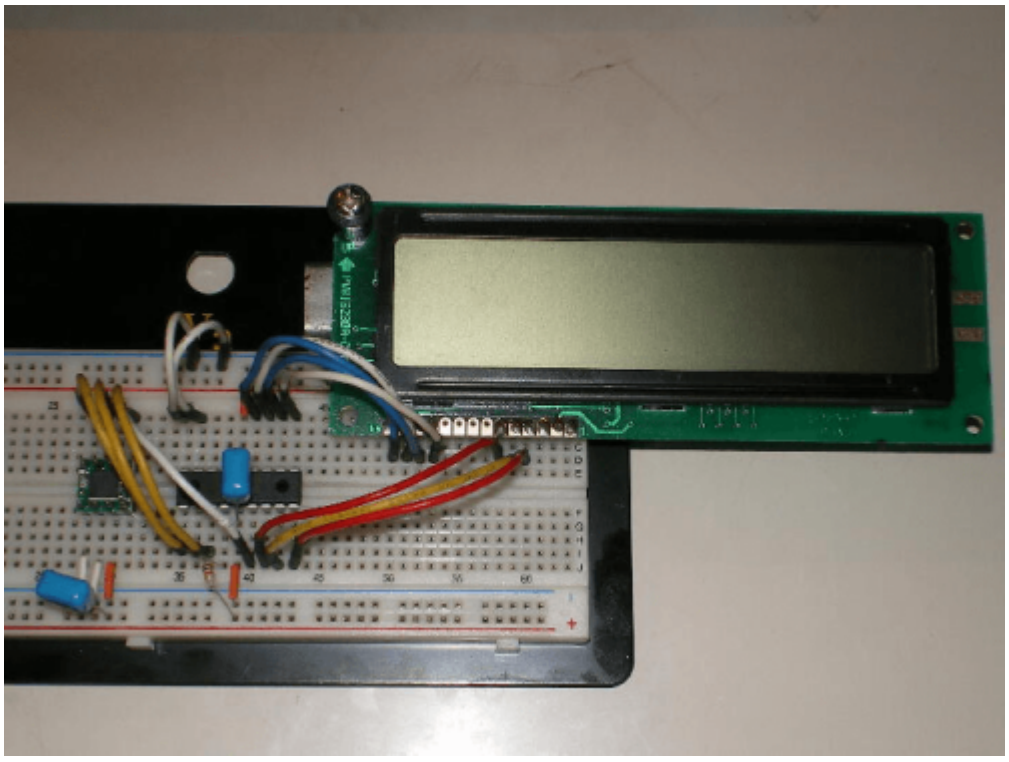
```
//*********************************************************************
*

const char character1[] = {16,16,16,16,16,16,16,16};
const char character2[] = {24,24,24,24,24,24,24,24};
const char character3[] = {28,28,28,28,28,28,28,28};
const char character4[] = {30,30,30,30,30,30,30,30};
const char character5[] = {31,31,31,31,31,31,31,31};

void RegistCustomChar()
{
    char i;
    //
    LCD_Cmd(64);
    for (i = 0; i<=7; i++) {
        LCD_Chr_Cp(character1[i]);
    }
    for (i = 0; i<=7; i++) {
        LCD_Chr_Cp(character2[i]);
    }
    for (i = 0; i<=7; i++) {
        LCD_Chr_Cp(character3[i]);
    }
    for (i = 0; i<=7; i++) {
        LCD_Chr_Cp(character4[i]);
    }
    for (i = 0; i<=7; i++) {
        LCD_Chr_Cp(character5[i]);
    }
    LCD_Cmd(LCD_RETURN_HOME);
}

//*********************************************************************
*

void   BarDisp(char index, unsigned int dat)
{
    short    i, j, k, cnt;
    //
    i = (dat * 10) / 128;
    j = i / 5;
    k = i - (j * 5);
    //
    if (index == 1)
        Lcd_Cmd(LCD_FIRST_ROW);
    else
        Lcd_Cmd(LCD_SECOND_ROW);
    //
    for (cnt = 1; cnt <= j; cnt++) {
        Lcd_Chr_Cp(4);
```

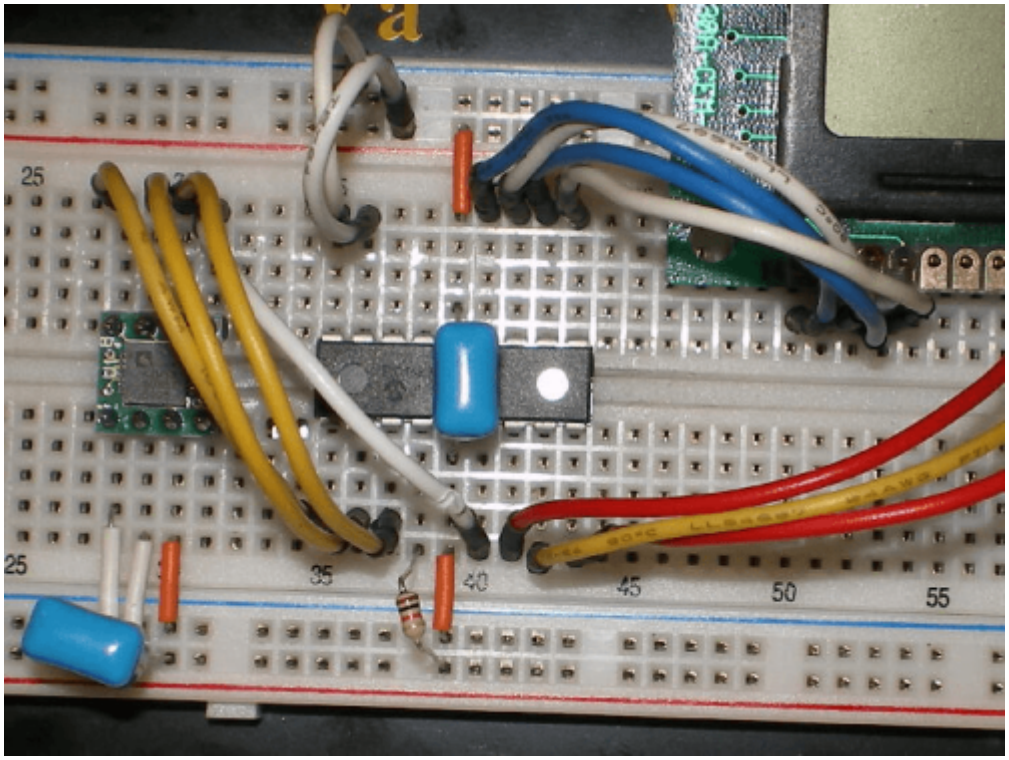
```
    }
    Lcd_Chr_Cp(k);
    for (cnt++; cnt <= 16; cnt++) {
        Lcd_Chr_Cp(' ');
    }
}

//*********************************************************************
*

double offset_x, offset_y, offset_z, offset_1g;

void   initKXM52()
{
    PORTB.F0 = 0;
    Delay_ms(100);
    //
    offset_z = (double)measurement(2) * 4.8828125;
    //
    offset_y = (double)measurement(3) * 4.8828125;
    //
    offset_x = (double)measurement(4) * 4.8828125;
    //
    PORTB.F0 = 1;
    Delay_ms(100);
    //
    offset_1g = measurement(4) * 4.8828125;
    offset_1g = offset_1g - offset_x;
    //
    PORTB.F0 = 0;
    Delay_ms(100);
}

//*********************************************************************
*

void main()
{
    static    unsigned    short    cnt, buf[16];
    static    unsigned    long    tmp_x, tmp_y, tmp_z, tmp;
    static    double        ad;
    //
    OSCCON = 0b01110000;    //クロックを8Mhzに設定する。
    TRISA  = 0b11111111;
    TRISB  = 0b11111110;
    //
    ANSEL  = 0b00011100;
    //
    Lcd_Config(&PORTB,3,1,2,7,6,5,4);
    RegistCustomChar();
```

```
    Lcd_Cmd(LCD_CURSOR_OFF);
    Lcd_Cmd(LCD_CLEAR);
    for (cnt = 0; cnt < 16; cnt++) {
        Lcd_Chr(1, cnt + 1, 0xFF);
        Delay_ms(50);
    }
    for (cnt = 0; cnt < 16; cnt++) {
        Lcd_Chr(2, cnt + 1, 0xFF);
        Delay_ms(50);
    }
    Delay_ms(1000);
    Lcd_Cmd(LCD_CLEAR);
    //
    initKXM52();
    //
    while (1) {
        // Z,Y,X
        tmp_z = measurement(2);
        tmp_y = measurement(3);
        tmp_x = measurement(4);
        // Z
        ad = (double)tmp_z * 4.8828125;
        ad = ad - offset_z;
        ad = (((double)labs(ad)) * 1000) / offset_1g;
        tmp = ad / 10;
        if ((ad - (tmp * 10)) >= 5)
            tmp++;
        WordToStr(tmp * 10, buf);
        Lcd_Out(1, 1, &buf[1]);
        // Y
        ad = (double)tmp_y * 4.8828125;
        ad = ad - offset_y;
        ad = (((double)labs(ad)) * 1000) / offset_1g;
        tmp = ad / 10;
        if ((ad - (tmp * 10)) >= 5)
            tmp++;
        WordToStr(tmp * 10, buf);
        Lcd_Out(1, 6, &buf[1]);
        // X
        ad = (double)tmp_x * 4.8828125;
        ad = ad - offset_x;
        ad = (((double)labs(ad)) * 1000) / offset_1g;
        tmp = ad / 10;
        if ((ad - (tmp * 10)) >= 5)
            tmp++;
        WordToStr(tmp * 10, buf);
        Lcd_Out(1, 11, &buf[1]);
        //
        switch (PORTA & 0x03) {
        case 0:
            Lcd_Chr(1, 16, 'X');
```

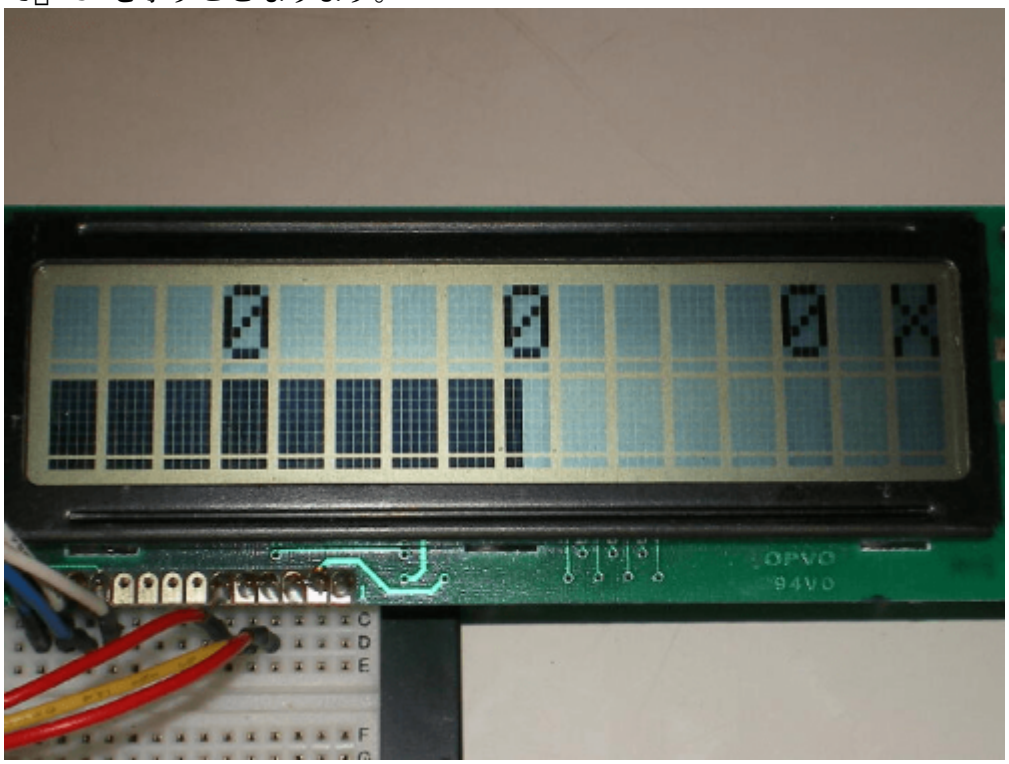
```
            BarDisp(2, tmp_x);
            break;
        case 1:
            Lcd_Chr(1, 16, 'Y');
            BarDisp(2, tmp_y);
            break;
        case 2:
            Lcd_Chr(1, 16, 'Z');
            BarDisp(2, tmp_z);
            break;
        case 3:
            Lcd_Chr(1, 16, ' ');
            BarDisp(2, 0);
            break;
        }
        //
        if (PORTA.F5 == 0)
            initKXM52();
    }
}

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

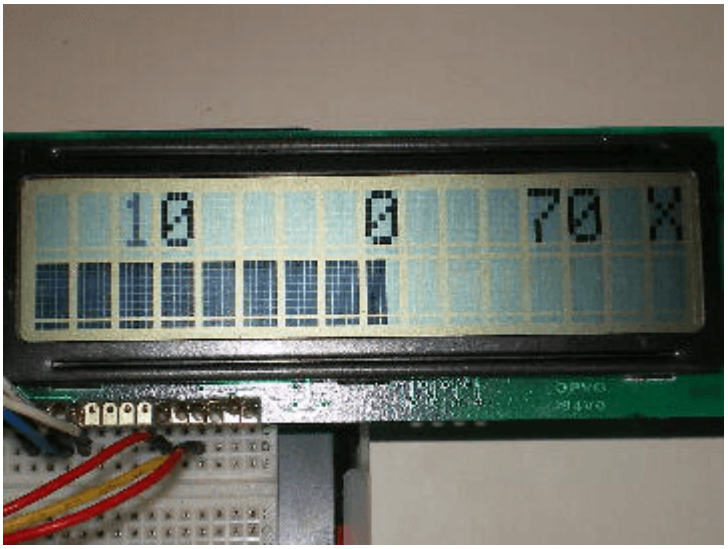
動作確認



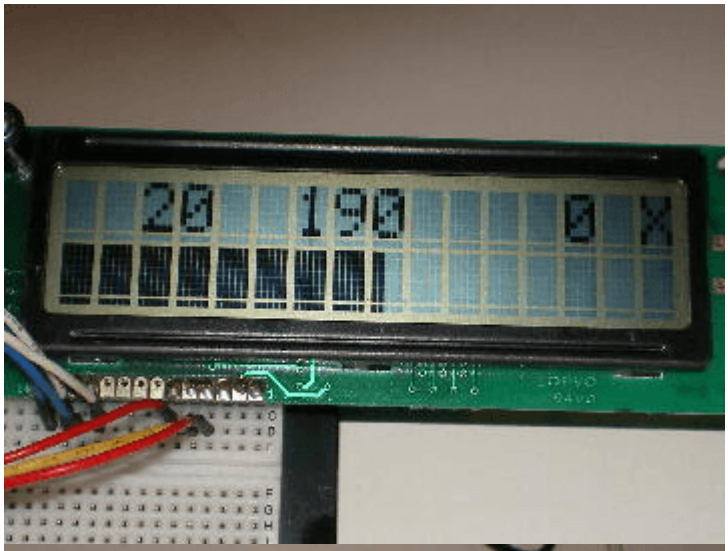


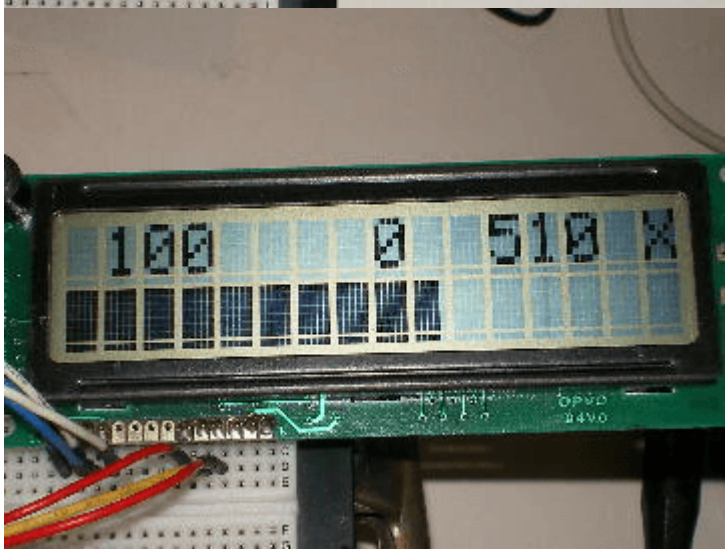
KXM52を、平坦な所に設置し、SW3を押下し感度とオフセットを求めます LCDの1行目は、左側から“Z方向““Y方向”“X方向 “ 、バー表示のタイトル(“X”“Y““Z”“ “) 表示単位は、mGalです LCDの2行目は、各方向(“X”“Y““Z”)の電圧のバー表示(0~5V) この画像では、バー表示がほぼ中央なので2.5Vを示すことなります。

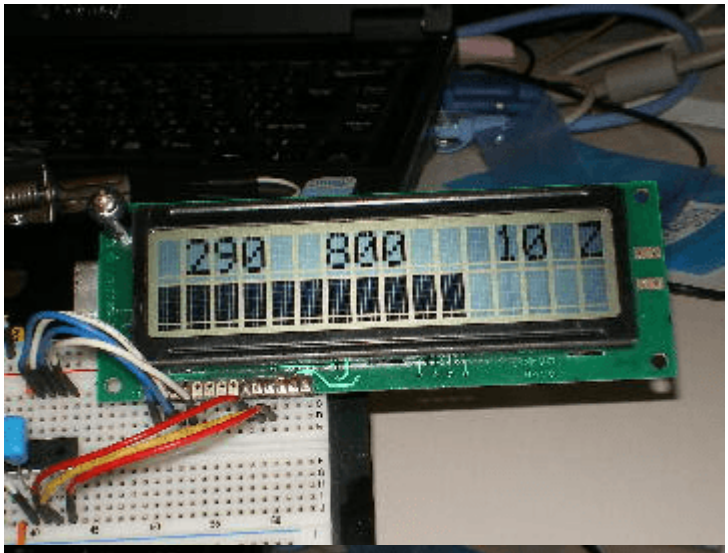


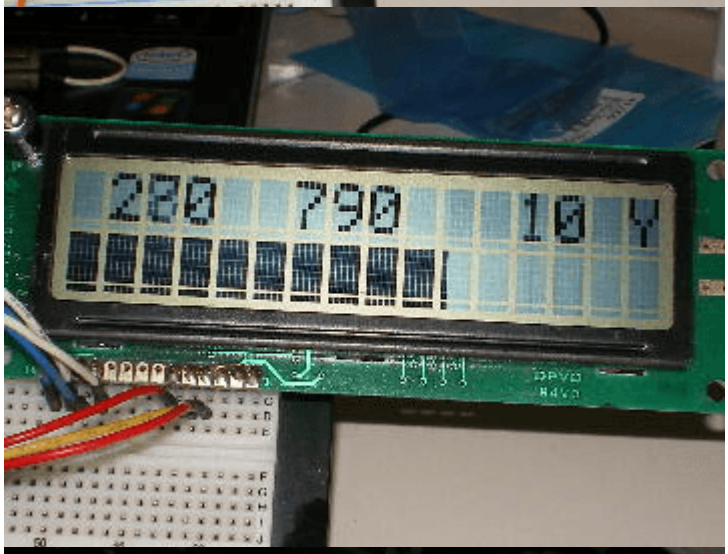


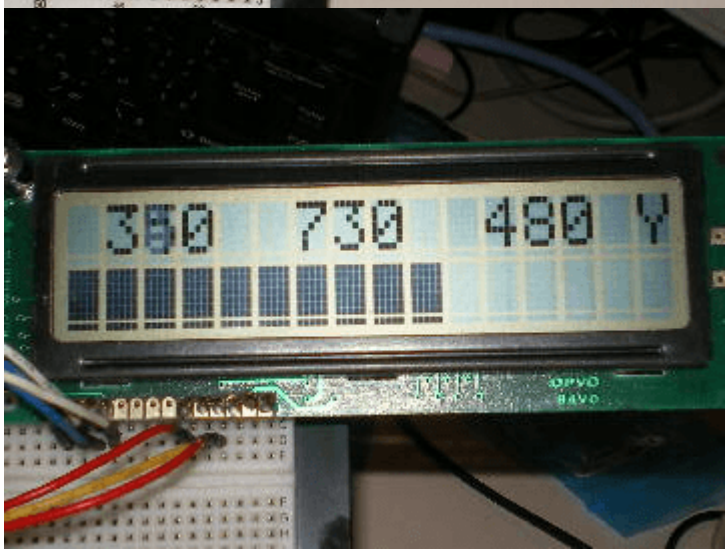
KXM52を、いろいろな方向に傾けてみました。











如何ですか? 本ユニットを応用すると、地震の震度表示や記録等に使えそうですね!{ }!