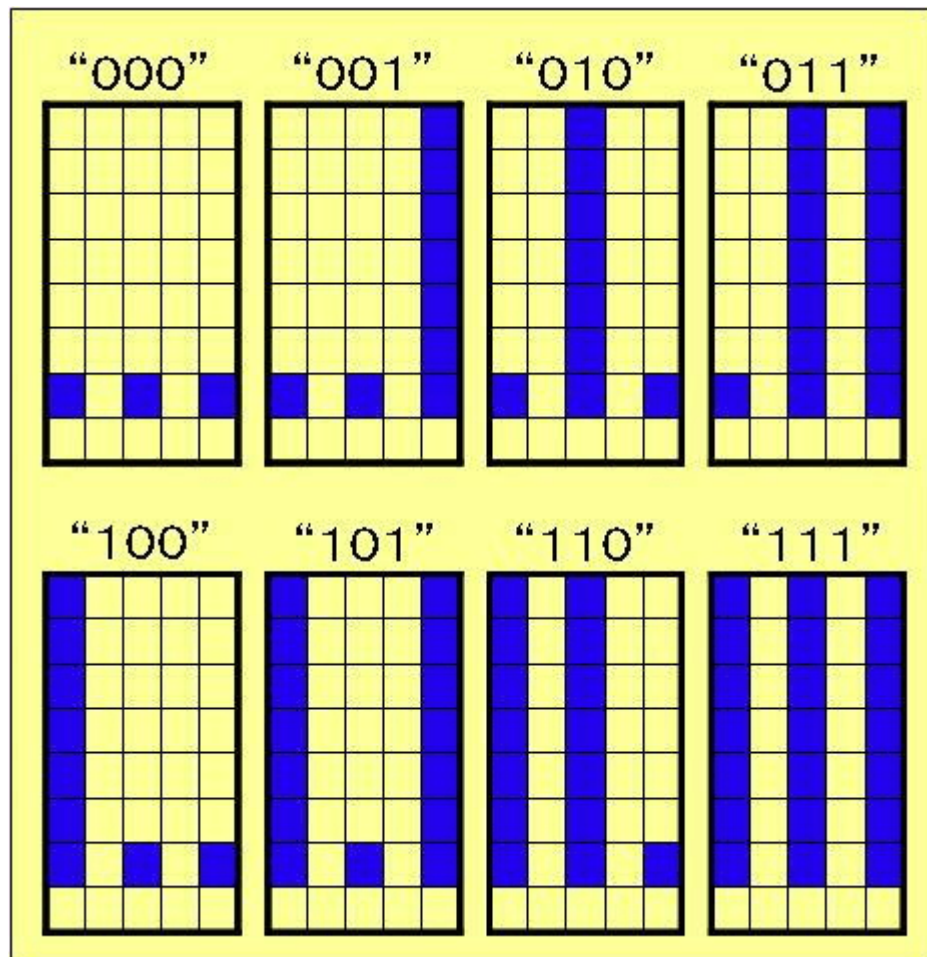
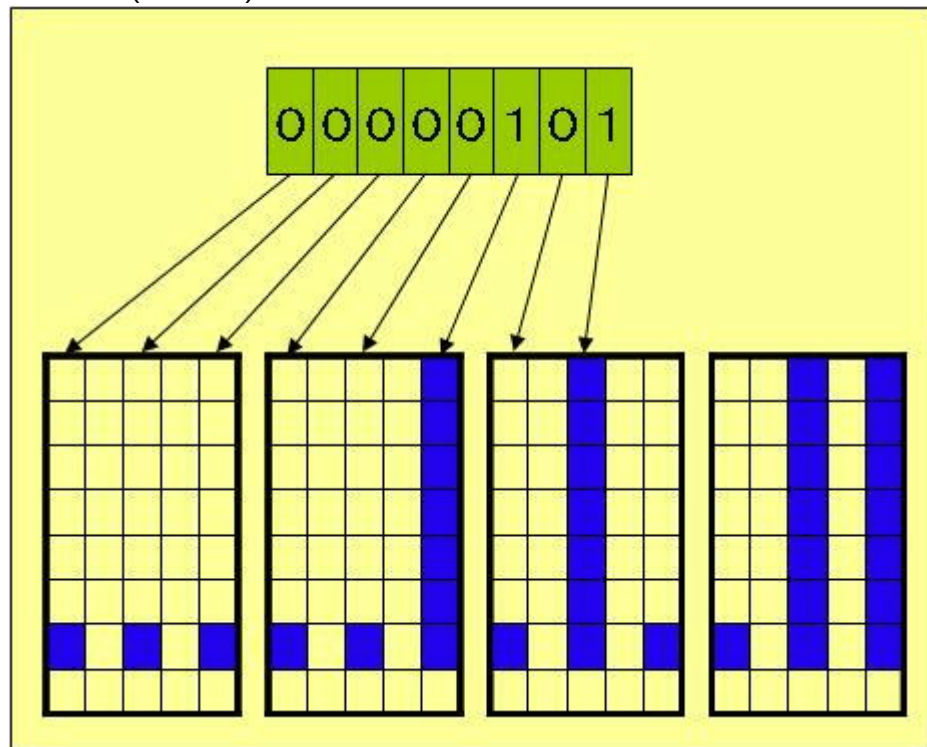




<1バイト(8ビット)とLCDの関係>

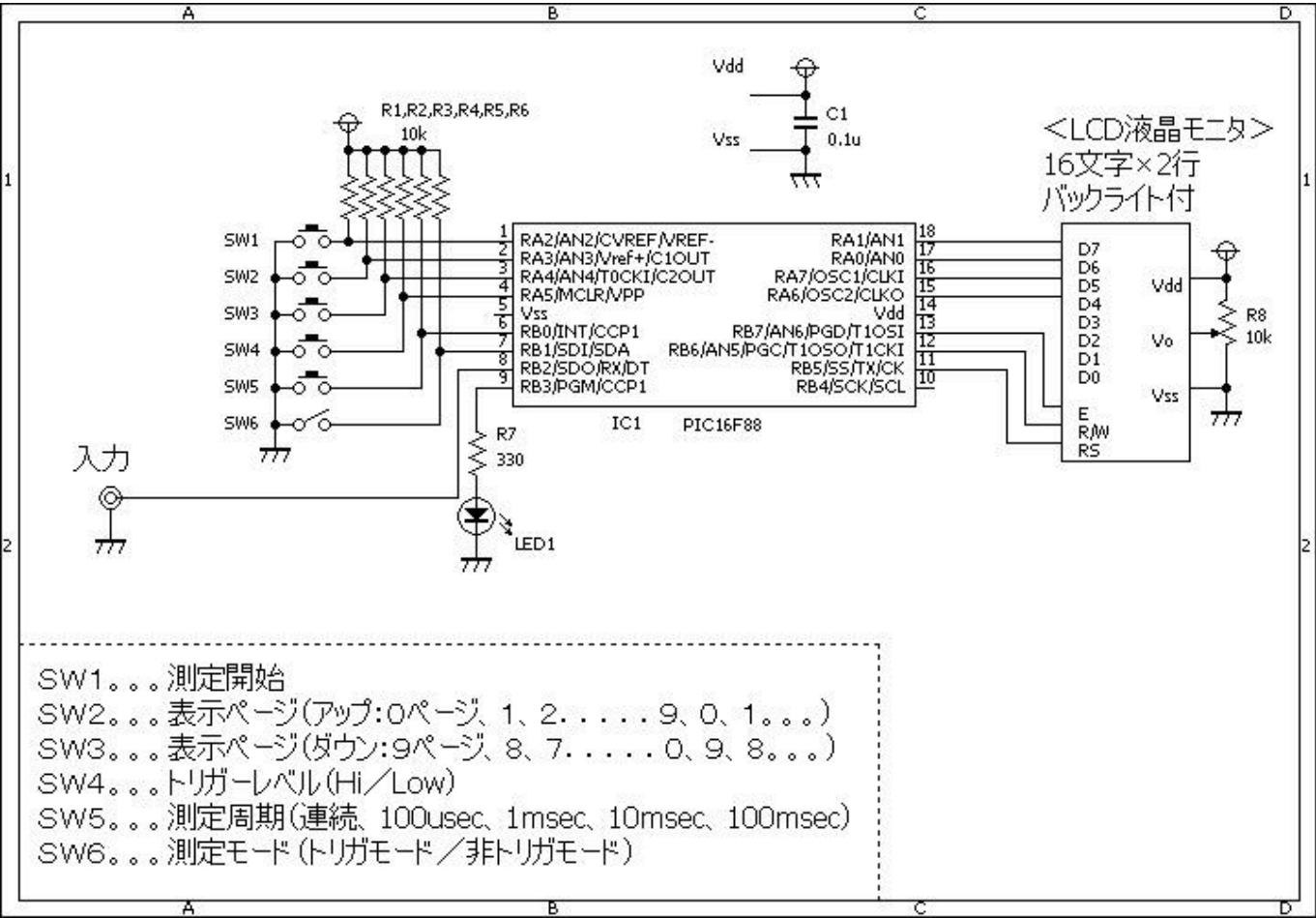


<測定周期と時間の関係>

測定周期	測定時間
連続	-

測定周期	測定時間
100usec	40msec
1msec	480msec
10msec	4.8sec
100msec	48sec

回路図



ソースコード

bitScope.c

```
//*****
*
/*
<ビットスコープ>
*/
//*****
*

#define      SW_START      PORTA.F2
#define      SW_UP        PORTA.F3
```

```
#define      SW_DOWN          PORTA.F4
#define      SW_TRIGGER      PORTA.F5
#define      SW_CYCLE        PORTB.F0
#define      SW_MODE         PORTB.F1

#define      SIGNAL          PORTB.F2

#define      LED              PORTB.F3

//*****
*

const char character1[] = {0,0,0,0,0,0,21,0};
const char character2[] = {1,1,1,1,1,1,21,0};
const char character3[] = {4,4,4,4,4,4,21,0};
const char character4[] = {5,5,5,5,5,5,21,0};
const char character5[] = {16,16,16,16,16,16,21,0};
const char character6[] = {17,17,17,17,17,17,21,0};
const char character7[] = {20,20,20,20,20,20,21,0};
const char character8[] = {21,21,21,21,21,21,21,0};

void RegistCustomChar()
{
    short    i;
    //
    Lcd_Custom_Cmd(64);
    for (i = 0; i <= 7; i++) {
        Lcd_Custom_Chr_Cp(character1[i]);
    }
    for (i = 0; i <= 7; i++) {
        Lcd_Custom_Chr_Cp(character2[i]);
    }
    for (i = 0; i <= 7; i++) {
        Lcd_Custom_Chr_Cp(character3[i]);
    }
    for (i = 0; i <= 7; i++) {
        Lcd_Custom_Chr_Cp(character4[i]);
    }
    for (i = 0; i <= 7; i++) {
        Lcd_Custom_Chr_Cp(character5[i]);
    }
    for (i = 0; i <= 7; i++) {
        Lcd_Custom_Chr_Cp(character6[i]);
    }
    for (i = 0; i <= 7; i++) {
        Lcd_Custom_Chr_Cp(character7[i]);
    }
    for (i = 0; i <= 7; i++) {
        Lcd_Custom_Chr_Cp(character8[i]);
    }
}
```

```
Lcd_Custom_Cmd(LCD_RETURN_HOME);
}

//*****
*

static unsigned short cycleTime;

void cycleTimeProc()
{
    if (cycleTime == 0)
        return;
    //
    switch (cycleTime) {
    case 1:
        Delay_us(100);
        break;
    case 2:
        Delay_us(1000);
        break;
    case 3:
        Delay_ms(10);
        break;
    case 4:
        Delay_ms(100);
        break;
    }
}

//*****
*

static unsigned short data[60];

void measurement()
{
    static short i, j, k;
    //
    for (i = 0; i < 60; i++) {
        k = 0x00;
        for (j = 0; j < 8; j++) {
            k |= SIGNAL;
            if (j < 7)
                k = k << 1;
            cycleTimeProc();
        }
        data[i] = k;
    }
}

//*****
```

```
*

void display(short page)
{
    static unsigned short i, j, k, tmp, dat[6];
    //
    Lcd_Custom_Chr(2, 6, page + '0');
    //
    dat[0] = data[(page * 6) + 0];
    dat[1] = data[(page * 6) + 1];
    dat[2] = data[(page * 6) + 2];
    dat[3] = data[(page * 6) + 3];
    dat[4] = data[(page * 6) + 4];
    dat[5] = data[(page * 6) + 5];
    //
    Lcd_Custom_Cmd(LCD_RETURN_HOME);
    tmp = 0x00;
    k = 0;
    for (i = 0; i < 6; i++) {
        for (j = 0; j < 8; j++) {
            tmp |= (dat[i] & 0b10000000) != 0 ? 1 : 0;
            k++;
            if (k == 3) {
                Lcd_Custom_Chr_Cp(tmp);
                tmp = 0x00;
                k = 0;
            }
            dat[i] = dat[i] << 1;
            tmp = tmp << 1;
        }
    }
}

//*****
*

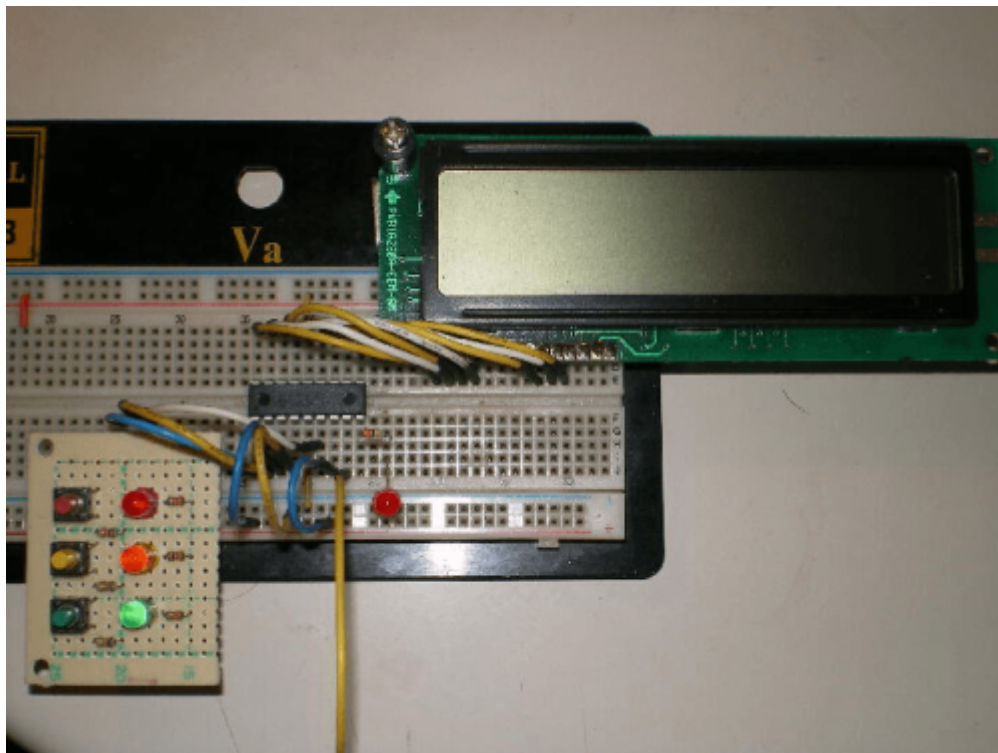
void main()
{
    static short cnt, page;
    //
    OSCCON = 0b01110000; // クロックを8Mhzに設定する。
    ANSEL = 0b00000000; // □□□変換は使用しない。
    //ポートの設定
    TRISA = 0b11111111;
    TRISB = 0b11110111;
    //
    Lcd_Custom_Config(&PORTA, 1, 0, 7, 6, &PORTB, 5, 6, 7);
    Lcd_Custom_Cmd(LCD_CURSOR_OFF);
    Lcd_Custom_Cmd(LCD_CLEAR);
    RegistCustomChar();
}
```

```
//
Lcd_Custom_Out(1, 1, "BitScope V1");
Delay_ms(1000);
Lcd_Custom_Cmd(LCD_CLEAR);
//
for (cnt = 0; cnt < 60; cnt++) {
    data[cnt] = 0x00;
}
page = 0;
LED = 0;
cycleTime = 0;
//
Lcd_Custom_Out(2, 1, "page=0");
Lcd_Custom_Out(2, 10, " 0usec");
//
while (1) {
    //測定
    if (SW_START == 0) {
        while (Button(&PORTA, 2, 1, 1) == 0)
            ;
        //
        LED = 1;
        if (SW_MODE == 0) { //トリガーモード切替
            while (SIGNAL != SW_TRIGGER)
                ;
        }
        measurement();
        LED = 0;
        //
        page = 0;
        display(page);
    }
    //ページ表示 (ページアップ)
    if (SW_UP == 0) {
        while (Button(&PORTA, 3, 1, 1) == 0)
            ;
        //
        if (page < 9)
            page++;
        else
            page = 0;
        display(page);
    }
    //ページ表示 (ページダウン)
    if (SW_DOWN == 0) {
        while (Button(&PORTA, 4, 1, 1) == 0)
            ;
        //
        if (page > 0)
            page--;
        else
            page = 9;
    }
}
```

```
        page = 9;
        display(page);
    }
    //測定周期切替
    if (SW_CYCLE == 0) {
        while (Button(&PORTB, 0, 1, 1) == 0)
            ;
        //
        if (cycleTime < 4)
            cycleTime++;
        else
            cycleTime = 0;
        //
        switch (cycleTime) {
        case 0:
            Lcd_Custom_Out(2, 10, " 0usec");
            break;
        case 1:
            Lcd_Custom_Out(2, 10, "100usec");
            break;
        case 2:
            Lcd_Custom_Out(2, 10, " 1msec");
            break;
        case 3:
            Lcd_Custom_Out(2, 10, " 10msec");
            break;
        case 4:
            Lcd_Custom_Out(2, 10, "100msec");
            break;
        }
    }
}

//*****
*
```

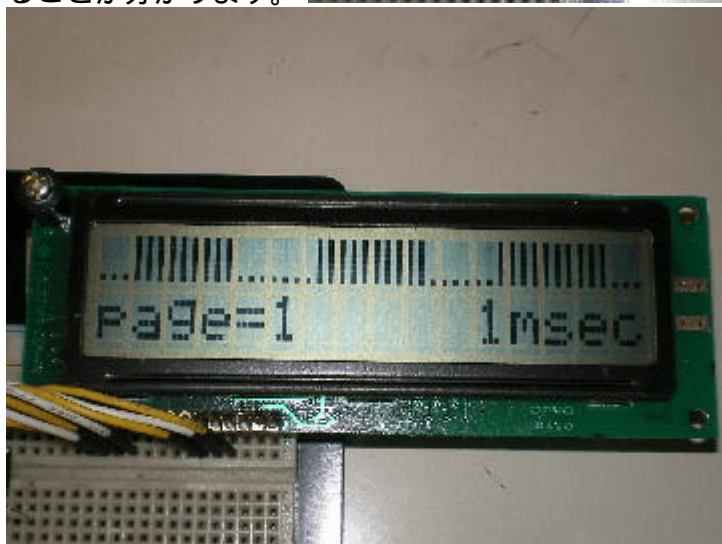
動作確認

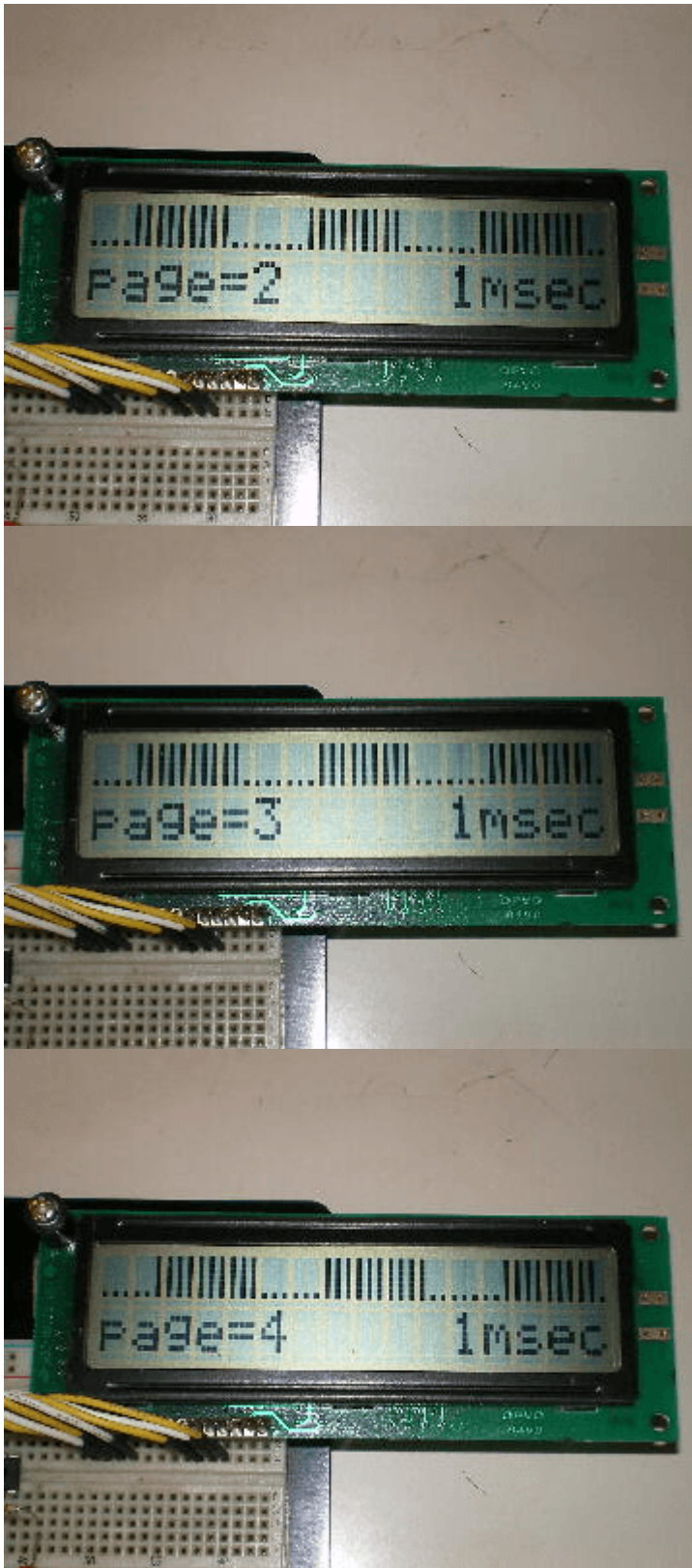


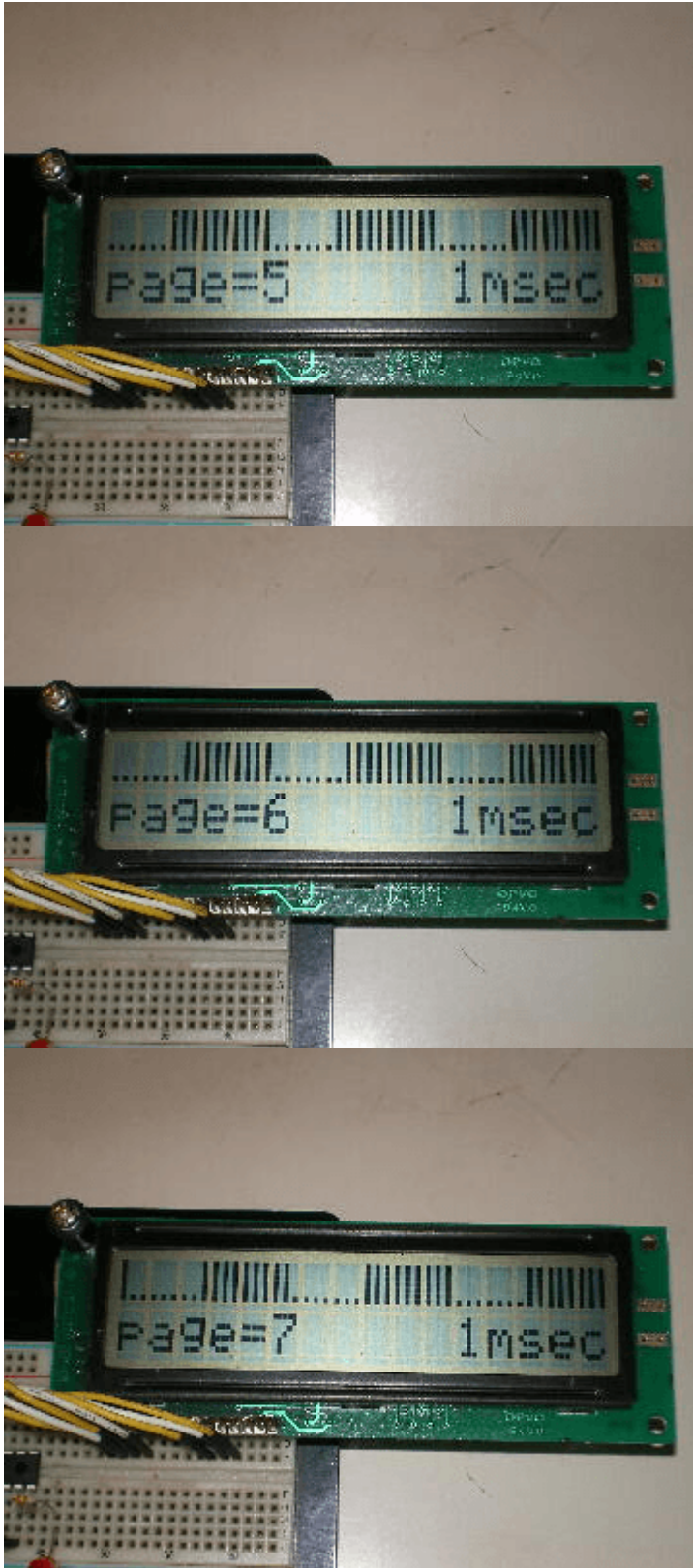
入力をオープン状態(60Hzのノイズ)で、測定周期1msecで測定してみました。その時の、ページ0-9までの測定結果です□ HiとLowの線を数えると、16-17本あるので□60Hz(17msec)をちゃんと測定してい

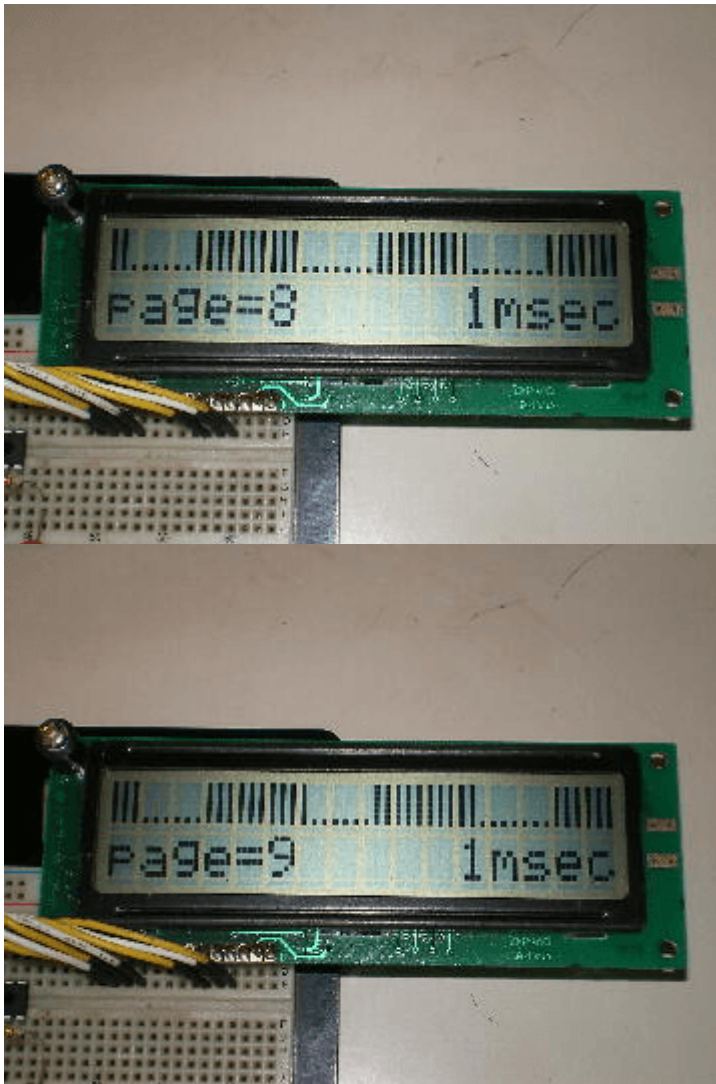


ることが分かります。





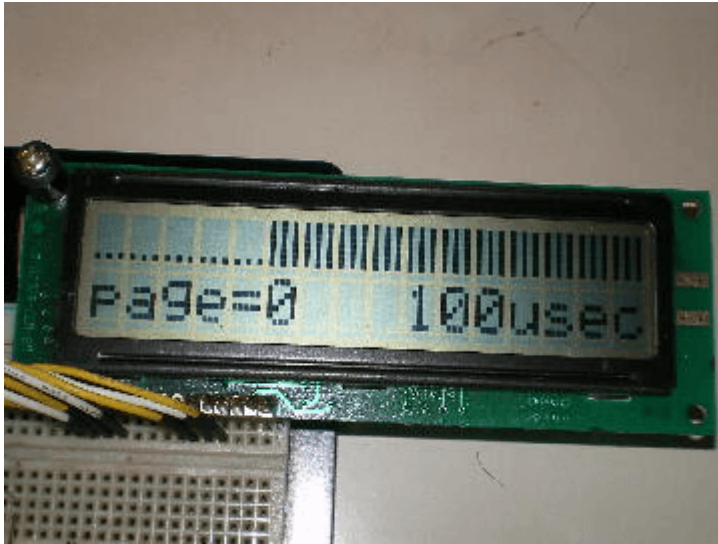




左側:10msec周期で測定してみました□(1msecよりも粗くなります) 右側:100usec周期で測定してみま



した□(1msecよりも細くなります)



From:

<http://deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic16f88:118&rev=1588211086>

Last update: 2025/10/17 14:27

