

```

/*****
* LCD(I2C)の応用プログラム
* I2C インターフェースで接続
* HI-TECH C を使用
* 数取り器の試作 2017.10.16 by Takehiko Inoue
*****/

#include <htc.h>
#include "lcd_i2c_lib.h"

/***** コンフィギュレーションの設定 *****/
__CONFIG(INTIO & WDTDIS & PWRTEN & BOREN & MCLREN
        & UNPROTECT & IESODIS & FCMDIS);

void itostring(char digit, unsigned int data, char *buffer);
unsigned char Msg[17] = "Counter = xxxxx ";
unsigned char Msg1[17] = "Memory = xxxxx ";

/***** メイン関数 *****/
void main(void)
{
    unsigned int Count, icon, Memory, Memoryh;
    unsigned char Flag,Memo;

    OSCCON = 0x70; // Set to 8MHz
    TRISIO = 0x38; // GP3,4,5 入力設定、ほか出力設定
    ANSEL = 0x00; // デジタルに設定
    CMCON0 = 0x07; // コンパレータオフ
    GPPU = 0; //プルアップ有効
    WPU = 0x30; //GP4,5 プルアップ
    WREN = 1; //EEPROM 書き込み開始可能
    /* 液晶表示器の初期化 */
    lcd_init();
    icon = 0;
    Flag = 1;
    Count = 0;
    Memo = 0;
    GPIO2 = 0;

```

```

lcd_cmd(0x80);
    lcd_str("Let's Count All!");
lcd_cmd(0xC0);
    lcd_str("2017 by T.Inoue");
/**** メインループ *****/
while(1)
{

if(GPIO5 == 0){
    GPIO2 ^= 1;
    lcd_cmd(0x80);
    itostring(5, Count++, Msg+10);
    lcd_str(Msg);
    if(Flag == 1)icon = 4;
    lcd_icon(icon, Flag);
    delay_100ms(5);
}

if(GPIO4 == 0){
    if(Memo == 0){
        GPIO2 ^= 1;
        Memory = Count-1;
        Memoryh = Memory;
        Memoryh /=256;
        Memory %=256;
        eeprom_write(0,Memory);
        eeprom_write(1,Memoryh);
        delay_100ms(1);
        icon = 5;
        lcd_icon(icon, Flag);
        Memo = 1;
        Flag = 1;
    }
    else{
        GPIO2 ^= 1;
        lcd_cmd(0xC0);

        itostring(5, eeprom_read(1)*256+eeprom_read(0) , Msg1+10);
    }
}
}

```

```

        delay_100ms(1);
        lcd_str(Msg1);

        icon = 6;
        lcd_icon(icon, Flag);
        delay_100ms(5);
        Memo = 0;
    }

}

}

}

/*****
* int 整数から ASCII 文字に変換
*****/
void itostring(char digit, unsigned int data, char *buffer)
{
    char i;

    buffer += digit;                // 最後の数字位置
    for(i=digit; i>0; i--) {        // 変換は下位から上位へ
        buffer--;                    // ポインタ-1
        *buffer = (data % 10) + '0'; // ASCII へ
        data = data / 10;            // 次の桁へ
    }
}

```