

/\*\*\*\*\*

Battery operated simple frequency counter  
PIC12F1829 with 1602 display via 4bits I/F  
By nobcha all right reserved

Ver 1.0 09/02/2011 PIC12F1822  
Ver 2.0 09/11/2011 PIC12F1822  
16MHz & Prescaler read out  
40MHz capable  
Ver 3.0 08/29/2011 PIC16F1827  
Ver 4.0 11/04/2012 For MGS  
Ver 5.0 10/03/2014 for 1829 intosc  
Ver 5.1 10/03/2014 for 1829 xtal

PIC12F1829 + LCD + charge pumping minus volt  
RC2-5:SC1602は4ビットモードとし、RC2-5で接続  
RA2:T0CKI:F-in  
RA3:Reset\_sw  
RC6: Charge pump source:CCP4  
RB6:Heart beat LED  
RC0:Toggle Prescaler:CNT DISABLE  
RB5: not used SDA2 MSSP2  
RB4: not used UP\_sw  
RC7: not used Down\_sw  
RB7: not used SCL2 MSSP2  
RA1:LCD EN bit (enable)はRA1に接続  
RA0:LCD RS bit(RS)はRA0に接続

SC1602 pin connection via 4bit mode  
#1 Vdd=5V  
#2 Vss=GND  
#3 LCD contrast center of 2k VOL  
#4 RS RA0  
#5 R/W GND  
#6 EN RA1  
  
#11-14 DATA RC2-5

TMR0 is counter with 1/32 prescaler switched by RC0  
TMR1 is gate time controller as set (65536-40000) for10ms

XTALOSC 16MHz

Development Circumstance  
MPLAB IDE V8.86 HiTECH C V9.83

Counter data is put on to RA2 which is TMR0 input.  
TMR1 is worked for 10mS gate. TMR0 has 1:32 prescaler.  
Overflow of TMR0 is counted on count\_1.  
Get prescaler counts by padding later

\*\*\*\*\*/

#define \_XTAL\_FREQ 16000000

```

#include <htc.h>
#include <stdio.h>
#include "lcd.h"

#define TOGGLE_PRE LATC0
#define FIN_PORT TRISA2
#define F_GATE TRISC0
#define HEART_BEAT LATB6

_CONFIG(
    FOSC_HS & WDTE_OFF & PWRTE_ON & MCLR_ON & CP_OFF
    & CPD_OFF & BOREN_OFF & CLKOUTEN_OFF & IESO_OFF & FCMEN_OFF
);

_CONFIG(
    WRT_OFF & PLLEN_OFF & STVREN_ON & LVP_OFF
);

unsigned char timeup, pre_pad, tmr0_value, count_1 ;
long read_data;
int time;
// prototyping
void cnt_setup(void);
unsigned char get_pres(char);          // delay routine for 12.8MHz XCO
void time_delay( char );
void interrupt cnt_int(void);

void main(){
    unsigned char i,j=0, zero_sup,disp_data ;
    long decimal;

    LATA=0b10000000;          // LATA RESET
    TRISA = 0b00111100;      // LATA INPUT RA2:T0CKI,RA3:MCLR,RA4,5:CLOCK
    LATB=0b00000000;        // LATB RESET
    TRISB = 0b10110000;      // LATB INPUT RB4:SW2,RB5:SDA,RB7:SCL
    LATC=0b00000000;        // LATC RESET
    TRISC = 0b10000011;      // LATC INPUT RC1:SHT,RC7:SW3
    ANSELA = 0;              // PORTA all digital
    ANSELB = 0;              // PORTB all digital
    ANSELC = 0;              // PORTC all digital

    OSCCON = 0b01111000;    // OSTS 16MHz INTCLK

    OPTION_REG = 0b10100100; // INTEDG 0,TOCS T0CKI 1,T0SE0,PSA TIMER0,1/32

    INTCON=0;                // INT off
    T1CON = 0;                // Timer1 off
    PIR1 = 0b00000000;      // ADIF 0,RCIF 0,TXIF 0,SSPIF 0,CCP1IF 0,TMR2IF
0,TMR1IF 0

    //CCP4 PWM initializing (83.8KHz on RB3 @16MHz)
    CCP4CON = 0b00001100;    // use PWM mode
    CCPR4L = 0x0c;          // duty is 50%
    CCPTMRS = 0;            // Select TMR2

```

```

// TMR2 initilizing
T2CON = 0b00000100;          // POSTSCALE 1:1 ,TMR2 ON , PRESCALE 1:1
PR2 = 0x17;                  // resolution is 6.5bit mode
TMR2ON = 1;                  // TMR2 start
HEART_BEAT = 1;              // LED on
lcd_init();
_delay_ms(200);
_delay_ms(200);

HEART_BEAT = 0;              // LED on

lcd_goto(0);                  // select first line
lcd_puts("SIMPLE COUNTER");

lcd_goto(0x40);               // select second line
_delay_ms(100);
HEART_BEAT = ! RB6;          // LED on
_delay_ms(200);
HEART_BEAT = ! RB6;          // LED on

cnt_setup();                  // counter initial set up

while(1){
    GIE=1;                    // int enable

    if(timeup){               // wait for gate time up
        timeup=0;
        GIE=0;                // int disable

        read_data = 8192*(long)(count_1);

        read_data=((long)(TMR0-1)*32)+(long)(get_pres(32))+read_data ; // original

        i=0;
        decimal = 100000;
        zero_sup=1;           // Zero suppres flag

        while(i<6){           // 5digit
            disp_data = ((char)((read_data/decimal)%10)) | 0x30;
            // Get digit data

            if((disp_data==0x30)&zero_sup & i<4 ){ // i==0 ->change for khz
                disp_data=0x20; // zero suppres
            }
            else{ zero_sup=0;} // zero suppress release //

            lcd_putch(disp_data ); // Display digit
            decimal = decimal / 10; // 10->1
            i++;
            if(i==5){lcd_putch( 0x2e);} // Display Decimal Point
        }
        lcd_puts("kHz@10mS"); // khz
        _delay_us(100);
    }
}

```

```

        HEART_BEAT = ! RB6;           // LED on/off;
        _delay_ms(200);
        _delay_ms(200);
        lcd_goto(0x40);               // select second line
        cnt_setup();                  // Gate open Counter start
    }
}
}

// Function
void cnt_setup(void){
    GIE=0;
    TMR0 = 0 ;                       // TMR0 clear
    TMR1L = 0;                       // Clear Low Byte of TMR1
                                        // 16MHz 62.5nS*4 250nS 40000 count
    TMR1H = 99;                      // Set 177*256 + 224 -8(INT DELAY)
    TMR1L = 186;                     // Set (99*256+192-8) =25536-8=65536-40000-8

    TOIF=0;                          // TMR0 flag off
    TMR1IF=0;                         // TMR1 flag off
    TMR1IE=1;                         // TMR1 INT ENABLE
    TOIE=1;                          // TMR0 INT ENABLE
    PIE1=0b00000001;                // ADIE 0,RCIE 0,TXIE 0,SSPIE 0,CCP1IE 0,TMR2IE
0,TMR1IE 1

    timeup=0;                        // Reset timeup flag
    count_1=0;                       // Reset overtime flag

    INTCON=0b01100000;              // GIE 0,PEIE 1,T0IE 1,INTE 0,GPIE 0,T0IF 0,INTF
0,GPIF 0

    T1CON = 0b00000001;             // T1RUN 1,T1CKPS 00,T1OSCDIS,T1SYNC 1,TMR1CS
0,TMR1ON 1
    F_GATE = 1;                     // RC0 INPUT enable TRISCO
    FIN_PORT = 1;                   // RA2 INPUT enable
}

// getting prescaler value by padding
unsigned char get_pres(char pres){ // LB5 disable LA4 input , pres is prescaler setting
    pre_pad=0;                      // reset variant
    tmr0_value=TMR0;                // save TMR0 value
    F_GATE=0;                       // RC0 gate open
    while (TMR0==tmr0_value){      // prescaler count up ?
        TOGGLE_PRE =1;             // toggle
        TOGGLE_PRE = 0;
        pre_pad++;                 // prescaler padding
    }
    F_GATE=1;                       // RC0 gate closed
    return (pres - pre_pad);        // return with prescaler value
}

```

```

// time delay for 12.8MHz oscillator adopted
void time_delay( char value){          // 1uS for 12.8MHz, 0.65uS for 16MHz
    char m;
    for (m=value;m>0;m--){           // value must be under 127
        asm("nop");
        asm("nop");
        asm("nop");
        asm("nop");
    }
}

void interrupt cnt_int(void){
    GIE=0;
    if(TOIF){
        count_1++;                    // Timer 0 overflow occurred
        TOIF=0;
        GIE=1;
    }
    if(TMR1IF){
        F_GATE=0;                      // Stop count disable

        TMR1ON=0;
        timeup=1;                      // Gate time over
        TMR1IF =0;
    }
}

```