

```

683_simp_cnt_lcd_main
*****
Battery operated simple frequency counter
1602 display by 4040B counter with PIC16F683
By nobcha all right reserved

```

Ver 0.1 08/25/2010 Ver0.2 02/15/2011  
 Ver 2.0 08/20/2011

PIC16F683 + LCD via TC4040B + minus volt  
 PIN Assign  
 #7 GPIO0:pulse out for 4040  
 #6 GPIO1:EN  
 #5 GPIO2/ToCK1:counter input  
 #2 GPIO4/OSC2/CLK0 for FOSC/4  
 ->charge pumping negative voltage generated

TMR0 is counter with 1/2 prescaler  
 TMR1 is gate time controller as set (65536-10000)  
 10mS Time gate 2MHz max count

4040B pin  
 #10 clock in  
 #11 rst (70uS delayed #10 high state )  
 #9 seg-a  
 #7 seg-b  
 #5 seg-c  
 #4 seg-d  
 #6 seg-e  
 #13 seg-f  
 #12 seg-g  
 #14 seg-dp

CLK INT 4MHz

Development Circumstance  
 MPLAB IDE V8.73a HiTECH C V9.82

Counter data is put on to GPIO2 which is TMR0 input.  
 TMR1 is worked for 10mS gate. TMR0 has 1:2 prescaler.  
 Overflow of TMR0 is counted on count\_1.

```

*****
#define _LEGACY_HEADERS
#define _XTAL_FREQ 4000000

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

__CONFIG(BORDIS & UNPROTECT & PWRSEN & WDTON & MCLREN & INTCLK);

unsigned char timeup ;
long read_data, count_1;

void cnt_setup(void){
    TRISIO = 0b00101000;           // GPIO2 disable
    TMR0 = 0;                     // TMR0 clear
    TMR1L = 0;                    // Clear Low Byte of TMR1
    TMR1H = 216;                  // Set 216*256 + 240
    TMR1L = 240;                  // Set (216*256) =55536=65536-10000

    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, SSP1IE 0, CCP1IE 0, TMR2IE 0, TMR1IE 1

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

    INTCON=0b01100000;           // GIE 0, PEIE 1, TOIE 1, INTE 0, GPIE 0, TOIF 0, INTF 0, GPIF 0

    T1CON = 0b00000001;          // T1RUN 1, T1CKPS 00, T1OSCDIS, T1SYNC 1, TMR1CS 0, TMR1ON 1
    TRISIO=0b00101100;           // GPIO2 INPUT enable
}
```

```

}                               683_simp_cnt_lcd_main

void interrupt cnt_int(void) {
    GIE=0;
    if(T0IF) {
        count_1++;           // Timer 0 overflow occurred
        T0IF=0;
        GIE=1;
    }
    if(TMR1IF) {
        TRIS10=0b00101000;   // Stop count RA4 disable
        timeup=1;             // Gate time over
        TMR1IF =0;
    }
}

void main() {
    unsigned char i, zero_sup, disp_data ;
    short decimal;

    GPIO = 0b00000000;          // PORT clear
    ANSEL = 0;                  // GPIO all digital
    TRIS10=0b00101000;         // io0:clk, io1:en, io2:data in, io3:reset, io:fosc out
    OSCCON = 0b01101000;       // OSTS 4MHz INTCLK
    OPTION = 0b00110000;       // PORTB pullup, INTEDG 0, TOCS TOCKI 1, ToSE1, PSA TIMER0, 1/4
    INTCON=0;                  // INT off
    T1CON = 0;                  // Timer1 off
    PIR1 = 0b00000000;         // ADIF 0, RCIF 0, TXIF 0, SSP1IF 0, CCP1IF 0, TMR2IF 0, TMR1IF 0
    __delay_ms(200);           // power on timing
    __delay_ms(200);
    __delay_ms(200);

    lcd_init();

    lcd_goto(0);               // select first line
    lcd_puts("Frequency Gate");

    lcd_goto(0x40);            // select second line
    lcd_puts("10mS");

    lcd_goto(0x40);            // select second line
    _deTay(200);               // counter initial set up
    cnt_setup();
    _deTay(10);

    while(1) {
        GIE=1;                  // int enable
        if(timeup){              // wait for gate time up
            timeup=0;
            GIE=0;                // int disable
            if(count_1>100){lcd_puts("OF ");} // more than 64 could not be covered short
            else{lcd_puts(" ");}
            read_data=((long)(TMR0)+(count_1)*256)*2; // (tmr0+(overflow
            count_1*256))*prescaler
            i=0;
            decimal = 10000;
            zero_sup=1;             // Zero suppress flag
            while(i<5){           // 5digit
                disp_data = ((char)((read_data/decimal)%10)) | 0x30; // Get digit data
                if((disp_data==0x30)&zero_sup & i!=3 ){
                    disp_data=0x20; // zero suppress
                }
                else{ zero_sup=0;} // zero suppress release
            }
        }
    }
}

```

```
683_simp_cnt_lcd_main
lcd_putchar(disp_data);           // Display digit
decimal = decimal / 10;           // 10->1
i++;
if(i==4){lcd_putchar( 0x2e);}    // Display DP
lcd_puts("kHz");
for(i=126;i>0;i++){
    _delay(100);
}
lcd_goto(0x40);                  // select second line
cnt_setup();                      // Gate open Counter start
}
}
```