#include <xc.h>

#include <stdint.h>

#include <stdlib.h>

#include <stdio.h>

#include "configBits.h"

#include "lcd.h"

//BEGIN CONFIG

//#pragma config OSC = INTIO67

#pragma config WDT = OFF

//END CONFIG

//#define \_XTAL\_FREQ 1000000ul //Set clock FRQ to 1 MHz

#define LEDPin LATDbits.LATD7 //Define LEDPin as Port D PIN 0

#define LEDTRIS TRISDbits.TRISD7 //Define LEDTris as TRISD Pin 0

#define OSC\_CLOCKS\_PER\_INSTRUCTION\_CYCLE (4ul)

#define TIMER0\_PRESCALE\_COUNT (64ul)

#define TIMER0\_COUNTS\_PER\_HALF\_SECOND (65536ul - (((\_XTAL\_FREQ / OSC\_CLOCKS\_PER\_INSTRUCTION\_CYCLE)/TIMER0\_PRESCALE\_COUNT)/2ul))

int count = 0;

int cnt =0;

//END DEFINE

//REVIEW DBL CHECK

//void CLK\_INIT( void ){

// OSCCONbits.IRCF = 0b100; //1 MHz

//}

void TIMER\_INIT( void ){

T0CONbits.TMR0ON = 0; //stop TIMER0 counter

T0CONbits.T08BIT = 0; //select 16-bit mode

T0CONbits.T0CS = 0; //select the internal clock to drive timer0

T0CONbits.PSA = 0; //use prescaler for TIMER0

T0CONbits.T0PS = 0b101; //assign the 1:64 pre-scaler to TIMER0

TMR0H = TIMER0\_COUNTS\_PER\_HALF\_SECOND >> 8;

TMR0L = TIMER0\_COUNTS\_PER\_HALF\_SECOND;

T0CONbits.TMR0ON = 1; //enable TIMER0

INTCONbits.TMR0IE = 1; //Enable the TIMER0 overflow interrupt

//INTERUPT ENABLE

INTCONbits.GIE = 1; //GLOBAL INTERUPT ENABLE

INTCONbits.PEIE = 1; //PERIPHERAL INTERUPT ENABLE; TIMER0 = PERIPHERAL

}

void main( void ){

TRISD = 0x00; //Define output or Input

LATD = 0x00;

initLCD();

// CLK\_INIT();

TIMER\_INIT();

count++;

while (1){

}

}

//INTERRUPT ISR

void \_\_interrupt() SW\_ISR( void ){ //WEIRD SYNTAX BECAUSE OF PIC

INTCONbits.TMR0IF = 0; //CLEAR FLAG SO IT CAN BE TRIGGERED AGAIN

TMR0H = TIMER0\_COUNTS\_PER\_HALF\_SECOND >> 8;

TMR0L = TIMER0\_COUNTS\_PER\_HALF\_SECOND;

LEDPin ^= 1u; //Toggle LED

cnt++;

lcd\_clear();

printf("time");

lcd\_set\_ddram\_addr(LCD\_LINE2\_ADDR);

if (cnt%2 == 0){

printf("%d",cnt/2);

}

}