# EE2361 – mock exam - 7/3/2012 Exam time: 10 minutes

* **This exam is open book, open notes, and electronically submitted (both this word file and individual programs). No communication devices or online collaborations allowed.**
* **Use MPLAB to compile and test each program. Submit programs individually (only the .c and .asm files please) and name them as p1.c, p2.asm, etc.**

**NAME: Patrick Barrett ID:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1- (20 points) Write an assembly program to add numbers stored at address 0x00 and 0x01 and place the result in address 0x02.  
 **IN ADDITION TO UPLOADING YOUR FILE p1.asm, please copy/paste the .asm file’s contents in the box as well.**

|  |
| --- |
| **Did your program compile? How did you test it?** Yes, placed numbers in to 0x00, 0x01 |
| **Source code:**  include p18F4550.inc ;Include Rigister Names and Whatnot  start: movf 0x00, w  addwf 0x01, w  movwf 0x02  end |

2- (25 points) Write a program to turn on RD0, wait 1ms and turn off RD0. Use Timer 0 to generate the 1ms delay.

|  |
| --- |
| **Show your timer calculations here:** ((1ms / (250/3) ns)) = 12000 |
| **Did your program compile? How did you test it?** Yes, ran the stopwatch. |
| **Source code:** #include <p18f4550.h>  #pragma config PLLDIV=2, CPUDIV=OSC1\_PLL2, USBDIV=2, IESO=ON, WDT=OFF  #pragma config BOR=OFF, PWRT=ON, LVP=OFF, FOSC=HSPLL\_HS, FCMEN=OFF, VREGEN=OFF  #pragma config MCLRE=ON, STVREN=ON, LPT1OSC=ON, PBADEN=OFF  //######### Function Declarations ################  void high\_isr(void);  void low\_isr(void);  void display(unsigned char);  void displayNext(unsigned char);  void setup();  //######### Variables ################  unsigned char i;  //######### Interrupts ################  #pragma code high\_isr\_entry=8  void high\_isr\_entry(void){  \_asm goto high\_isr \_endasm  }  #pragma code low\_isr\_entry=0x18  void low\_isr\_entry(void){  \_asm goto low\_isr \_endasm  }  #pragma interrupt high\_isr  void high\_isr(void){  if(INTCONbits.TMR0IF == 1){  TMR0H = -12000 >> 8;  TMR0L = -12000;  INTCONbits.TMR0IF = 0;  PORTD ^= 1;  }  }  #pragma interruptlow low\_isr  void low\_isr(void){  }  //######### Functions ################  void setup(){  TRISB = 0x03; // set RB0, RB1 to input  TRISC = 0x00;  TRISD = 0x00;  INTCON2bits.TMR0IP = 1; //High Priority  T0CONbits.T0CS = 0; //Clock from interal clock = 0, Pin Count = 1  T0CONbits.PSA = 1; //Don't Use Prescaler  T0CONbits.T08BIT = 0; //Use Timer 0 as 16 Bit  INTCONbits.TMR0IE = 1; //Enable Interrupt  TMR0H = -12000 >> 8;  TMR0L = -12000;  T0CONbits.TMR0ON = 1; //Turn on Timer 0  INTCONbits.GIE = 1; // Enable Interrups Globally  //Setup Port:  PORTD = 0x00;  }  void main(void) {  setup();  while(1) {  }  } |