Midterm: Multimeter with Pulse-meter (Group Project)

Due Date: Fri 11 Dec 2020 at 4pm on D2L Dropbox

Assignment:

Using the Microcontroller and the driver functions developed so far (ADCs, IO control, Serial Displays, timers etc), you will design a multi-meter capable of measuring voltage, resistance, **digital pulse signals** and displaying the measured values on the Computer terminal. The App will use the push buttons (PBs) connected to the input ports RA2, RA4 and RB4 as shown in the schematic in the lecture slide 'HW and IO Control.pdf'. PB1, PB2 and PB3 represent push buttons connected to ports RA2, RA4 and RB4 respectively. **The app should be able to measure resistance of a resistor connected to port pin16/RB13/AN11, measure any voltage applied to AN5/pin 8/ RA3 as follows and measure amplitude and frequency of any pulse signal applied on pin 15/AN12/RB12**.

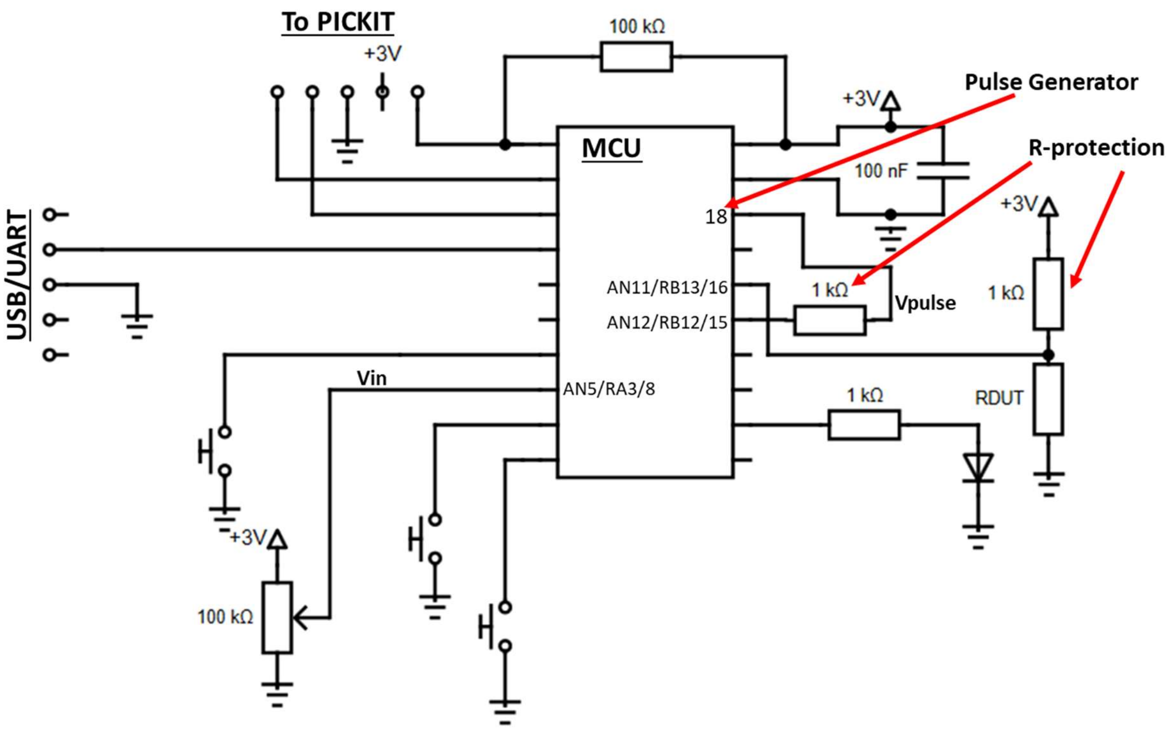
User input(s)	**Output(s)**
if PB1 is pressed	MCU is in voltmeter mode and should measure any voltage applied to AN5/pin 8/ RA3 in volts. This value should be displayed in volts on a single line of the PC terminal as follows: **VOLTMETER Voltage = ______ V**
if PB2 is pressed	MCU is in ohmmeter mode and should measure the resistance of any resistor connected to pin16/RB13/AN11 in ohms. This value should be displayed in ohms on a single line of the PC terminal as follows: **OHMMETER Resistance = _______ Ω** Please use the circuit connection shown below for pin16/RB13/AN11 to ensure there is no damage to the programmer and/or MCU. R_{DUT} is the resistance to be measured by your MCU/software
if PB3 is pressed **(Midterm project)**	MCU is in pulse meter mode, and should measure the amplitude and frequency of the input pulse coming into pin 15/AN12/RB12. This mode should work with any clock speed of the MCU. These values should be displayed on a single line of the PC terminal as follows: **PULSEMETER Freq = ___kHz, Amplitude = ____V**

Additional info:

Implement the above controller using the hardware kit and your code, which will be designed using basic ANSI C commands; and driver functions provided. **Use of polling instead of interrupts will lose points.**

Circuit connections:

Use the following the circuit connections to implement the voltmeter and ohmmeter. Ensure the use of the current-limiting 1 k-ohms resistors (Rprotection) for protection of the IO ports and PICKIT programmer in the event the port are incorrectly configured as inputs or outputs as shown below. R-DUT, Vin and Vpulse are the resistance, analog input voltage and voltage pulse to be measured by your software in the 3 modes determined by the push button presses.



Pre-processor Directives:

Pin 8 is one of those exceptional pins that are multiplexed to the external clock oscillator port in addition to an analog and digital IO. Using the above preprocessor commands will turn off any clock-input related feature on pin 8 thereby avoiding any conflicts on pin 8 while it is being used with the ADC.

```
// CLOCK CONTROL
#pragma config IESO = OFF    // 2 Speed Startup disabled
#pragma config FNOSC = FRC  // Start up CLK = 8 MHz
#pragma config FCKSM = CSECMD // Clock switching is enabled, clock monitor off
#pragma config SOSCSEL = SOSCLP // Secondary oscillator for Low Power Operation
#pragma config POSCFREQ = MS  //Primary Oscillator/External clk freq betwn
#pragma config OSCIOFNC = ON  //CLKO output disabled on pin 8, use as IO.
#pragma config POSCMOD = NONE  // Primary oscillator mode is disabled
```

Pulse Generation:

You will use the MCU itself to emulate a pulse generator on pin 18 of the MCU. This generated pulse is fed into AN12/RB12/pin15 on which your code will implement a pulse meter. To generate the pulse place the following code in your main() function before your code enters the infinite while(1) loop. To generate pulses of different frequencies, you will modify the clock frequency and RODIV bits in the code below as shown under deliverables.

```
    // Change Clock
      NewClk(8); // 8 for 8 MHz; 500 for 500 kHz; 32 for 32 kHz

    //Clock output on REFO/RB15 - PULSE GEN Testing purposes only
     TRISBbits.TRISB15 = 0;  // Set RB15 as output for REFO
     REFOCONbits.ROSSLP = 1; // Ref oscillator is disabled in sleep
     REFOCONbits.ROSEL = 0; // Output base clk showing clock switching
     REFOCONbits.RODIV = 0b0011;
     REFOCONbits.ROEN = 1; // Ref oscillator is enabled
```

Function names: Students can use any convention when naming functions or organizing code. A state diagram is required as part of your submission. Use microcontroller-specific register and bit names wherever applicable in the state diagram.

Display instructions: All displays on the PC terminal window should be on a single line. Note that display functions carried out at 32 kHz (300 Baud) can affect timer delays. Your code should account for such delays when producing delays specified in the table above.

Interrupts: Interrupt ISR names are provided in the lecture slides. As specified in lecture, IO (CN interrupts) are triggered on rising and falling edges and due to any debounce effects of the push buttons. A debounced switch will result in several hi to lo and lo to hi fluctuations at the Microcontroller input before stabilizing to a steady and fixed voltage when the switch is pressed. Your code should filter out any such effects.

Deliverables:

This is a group project. Each group should upload the following onto their respective group D2L-Dropbox folder created:

1. **Zipped up file of the MPLAB project**. MPLAB projects can be zipped up by right clicking on the project and selecting package (See screenshot below). The zipped project is saved in the same project folder created by user. Make sure your driver code is commented properly.
2. **A single pdf document** showing the following:
   a. Names and UCIDs of all students in the group at the top of the document
   b. A State diagram showing the working of your code. Use microcontroller-specific register and bit names wherever application in the state diagram.
   c. The mathematical formulas used in determining the voltage and resistance being measured and displayed on the PC terminal.
   d. The maximum and minimum limits of the resistance and voltage that your app project can measure
   e. <u>The mathematical formulas used in determining the pulse frequency being measured and displayed on the PC terminal.</u>
   f. List of tasks performed by each group member
3. **Link to your video demo** uploaded on youtube, Vimeo or similar video hosting website along with the zipped up project. Include the link under description while uploading the zipped up project. Dropbox or Google or OneDrive links are allowed as well but ensure that videos are in .mp4 or .mov format. Videos uploaded in any other format will lose points. Video demo should be a single recording and show the following
   a. UCID card of one group member placed in front of the computer with MPLAB and/or hardware running
   b. Demo of the code and hardware operation showing all states. Use the potentiometer to generate different voltages and resistances on pins 8 and 16 of the MCU in voltmeter and ohmmeter modes. Show your app measuring the voltage and resistance on the PC terminal in real time as the potentiometer knob is being turned.
   c. <u>Demo of the code and hardware operation showing pulse measurement operation for a few different pulse frequencies fed into AN12/RB12/15. Try out your code for the following values of clock and RODIV bits. MCU will have to be reprogrammed for each of the pulse settings below.</u>

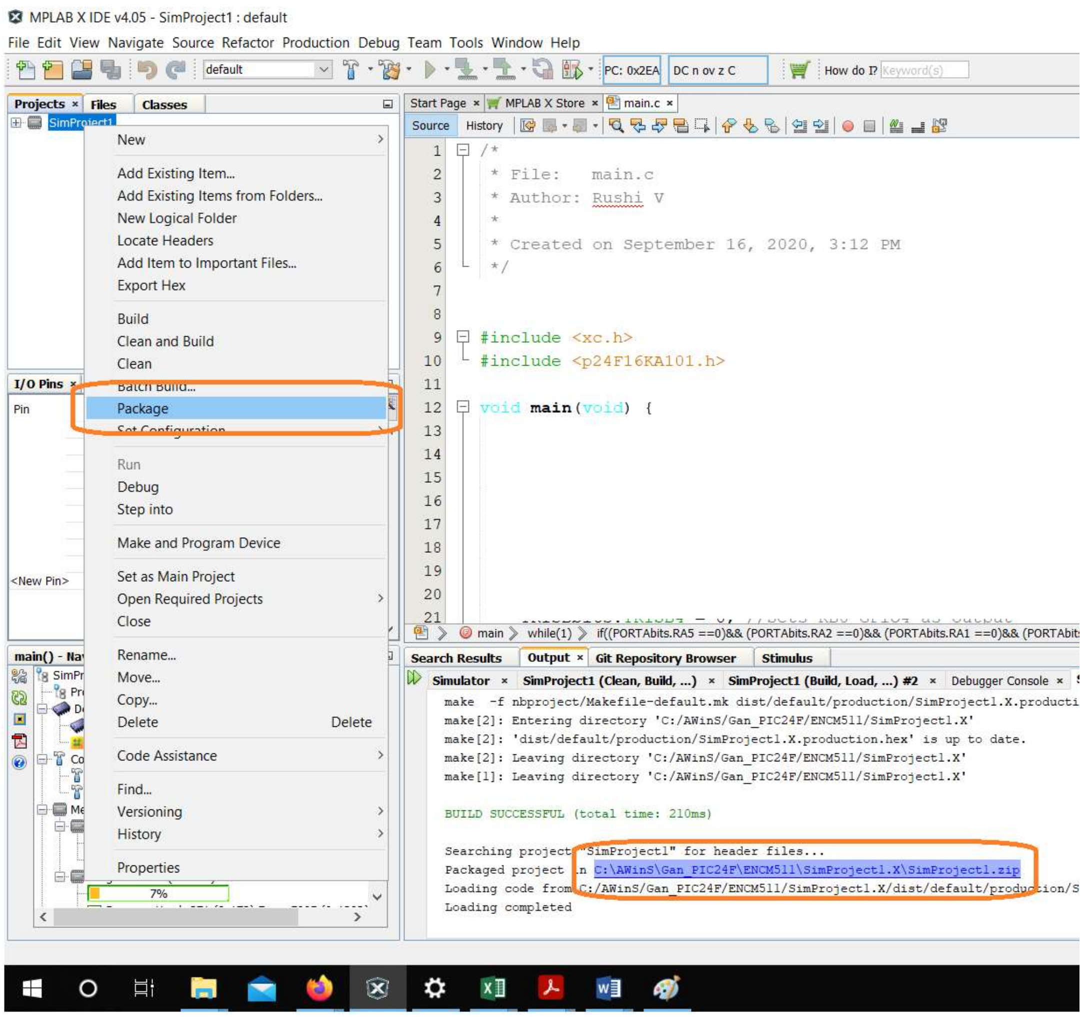
```
 ///////////////PULSE TEST SETTING 1
 // Change Clock
   NewClk(8); // 8 for 8 MHz; 500 for 500 kHz; 32 for 32 kHz

 //Clock output on REFO/RB15 - PULSE GEN Testing purposes only
  TRISBbits.TRISB15 = 0;  // Set RB15 as output for REFO
  REFOCONbits.ROSSLP = 1; // Ref oscillator is disabled in sleep
  REFOCONbits.ROSEL = 0; // Output base clk showing clock switching
  REFOCONbits.RODIV = 0b1001;
  REFOCONbits.ROEN = 1; // Ref oscillator is enabled


///////////////PULSE TEST SETTING 2
```

```
    // Change Clock
      NewClk(8); // 8 for 8 MHz; 500 for 500 kHz; 32 for 32 kHz

    //Clock output on REFO/RB15 - PULSE GEN Testing purposes only
     TRISBbits.TRISB15 = 0;  // Set RB15 as output for REFO
     REFOCONbits.ROSSLP = 1; // Ref oscillator is disabled in sleep
     REFOCONbits.ROSEL = 0; // Output base clk showing clock switching
     REFOCONbits.RODIV = 0b1111;
     REFOCONbits.ROEN = 1; // Ref oscillator is enabled


   ///////////////PULSE TEST SETTING 3
    // Change Clock
      NewClk(500); // 8 for 8 MHz; 500 for 500 kHz; 32 for 32 kHz

    //Clock output on REFO/RB15 - PULSE GEN Testing purposes only
     TRISBbits.TRISB15 = 0;  // Set RB15 as output for REFO
     REFOCONbits.ROSSLP = 1; // Ref oscillator is disabled in sleep
     REFOCONbits.ROSEL = 0; // Output base clk showing clock switching
     REFOCONbits.RODIV = 0b1001;
     REFOCONbits.ROEN = 1; // Ref oscillator is enabled


   ///////////////PULSE TEST SETTING 4
    // Change Clock
      NewClk(500); // 8 for 8 MHz; 500 for 500 kHz; 32 for 32 kHz

    //Clock output on REFO/RB15 - PULSE GEN Testing purposes only
     TRISBbits.TRISB15 = 0;  // Set RB15 as output for REFO
     REFOCONbits.ROSSLP = 1; // Ref oscillator is disabled in sleep
     REFOCONbits.ROSEL = 0; // Output base clk showing clock switching
     REFOCONbits.RODIV = 0b0111;
     REFOCONbits.ROEN = 1; // Ref oscillator is enabled
```

Grading rubric: (Total = 30 points)

- Correct setup and use of pulsemeter - 16 points
- Correct integration of pulsemeter with app 2 – 3 points
- PDF report = 2 points
- Proper video and code upload format including commenting of code = 2 points
- Group participation = 2 points
- Participation in all online polls and surveys in class = 5 points

ENCM 511 – F2020 (Vyas)



```
/*
 * File:   main.c
 * Author: Rushi V
 *
 * Created on September 16, 2020, 3:12 PM
 */


#include <xc.h>
#include <p24F16KA101.h>

void main(void) {
```

```
make  -f nbproject/Makefile-default.mk dist/default/production/SimProject1.X.producti
make[2]: Entering directory 'C:/AWinS/Gan_PIC24F/ENCM511/SimProject1.X'
make[2]: 'dist/default/production/SimProject1.X.production.hex' is up to date.
make[2]: Leaving directory 'C:/AWinS/Gan_PIC24F/ENCM511/SimProject1.X'
make[1]: Leaving directory 'C:/AWinS/Gan_PIC24F/ENCM511/SimProject1.X'

BUILD SUCCESSFUL (total time: 210ms)

Searching project "SimProject1" for header files...
Packaged project in C:\AWinS\Gan_PIC24F\ENCM511\SimProject1.X\SimProject1.zip
Loading code from C:/AWinS/Gan_PIC24F/ENCM511/SimProject1.X/dist/default/production/S
Loading completed
```