

# Rules for ECE3301L Lab

## 1) Commenting

- a) Use `/**` to start a comment line
- b) Place a general comment line for each routine describing what is the function of the routine
- c) Add a comment line on each relevant line of code. Make sure that all the comment lines are lined up under a fixed column and this column should be lined throughout the entire program (if possible). See the sample program below.

## 2) Variable definitions

The definition of the variables must be done right at the beginning of the routine.

## 3) Tabbing

- a) Always tab in right below a `{`. I prefer that this `{` should be on the start of the next line.
- b) Each `}` should line up with the associated `{`
- c) Don't tab when a variable name is defined. See the location on the variable `'result'` below.

## 4) Prototyping

To make sure that the program can be properly compiled, it is a good practice to prototype each function when they are first defined. Once the function has been completely coded, you will need to copy the name of that function and paste it at the beginning of the program. If the function has parameters, then keep the list of those parameters (inside the parentheses) preserved with the types of each parameter while the name of each parameter can be omitted.

## 5) Program Start Code

To make sure that your program can properly run, each code must have the following lines at the start of the program:

```
#include <p18f4321.h>
#include <stdio.h>
#include <math.h>
#include <usart.h>
```

```
#pragma config OSC = INTIO2
#pragma config WDT=OFF
#pragma config LVP=OFF
#pragma config BOR =OFF
```

## 6) Uart.h program linked

Make sure that the 'Link in Peripheral Library' option is set in the linker section. See instructions in Lab #1.

## 7) Program properly organized in sub folders for each lab

**As described in Lab#1, each sub part within a lab should be developed using a separated sub-folder of that lab's folder. Make sure to follow the same procedure for every lab.**

Here is a typical code that your program should have after the comments are added:

```
void Init_ADC(void); // Start of prototyping area
unsigned int Get_Full_ADC(void); //
void Flash_LED(unsigned int); //

unsigned int Get_Full_ADC(void)
{
int result;
    ADCON0bits.GO=1; // Start Conversion
    while(ADCON0bits.DONE==1); // Wait for conversion to be completed (DONE=0)
    result = (ADRESH * 0x100) + ADRESL; // Combine result of upper byte and lower byte into
    return result; // return the most significant 8- bits of the result.
}

void Init_ADC(void)
{
    ADCON0=0x01; // select channel AN0, and turn on the ADDC subsystem
    ADCON1=0x0E; // set pin 2 as analog signal, VDD-VSS as reference voltage
                // and right justify the result
    ADCON2=0xA9; // Set the bit conversion time (TAD) and acquisition time
}

void Flash_LED(unsigned int ADC_result)
{
unsigned int counter1, counter2;

    LATB = 0x0A;
    for (counter2=delay; counter2>0; --counter2)
    {
        for (counter1=ADC_result ; counter1>0; -- counter1);
    }

    LATB = 0x05
    for (counter2=delay; counter2>0; --counter2)
```

```
{  
    for (counter1=ADC_result ; counter1>0; -- counter1);  
}
```