

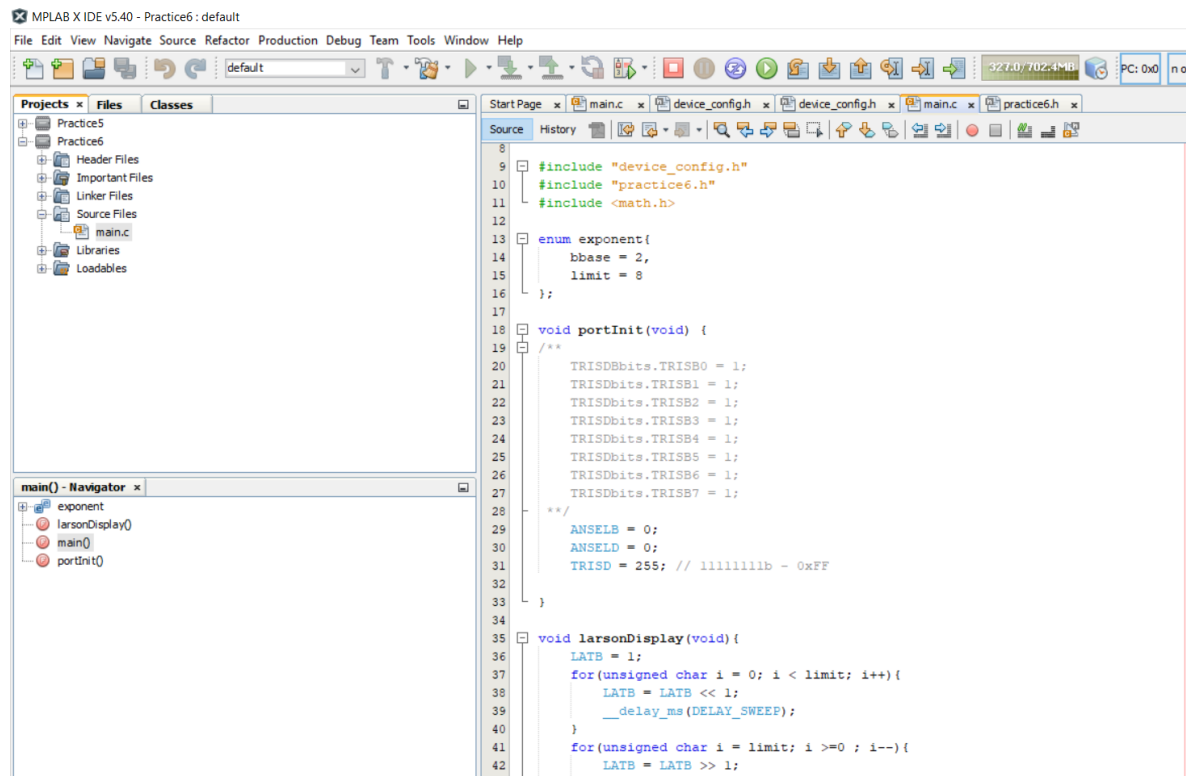
Jesus Ramiro Garza Hernandez

A01410925

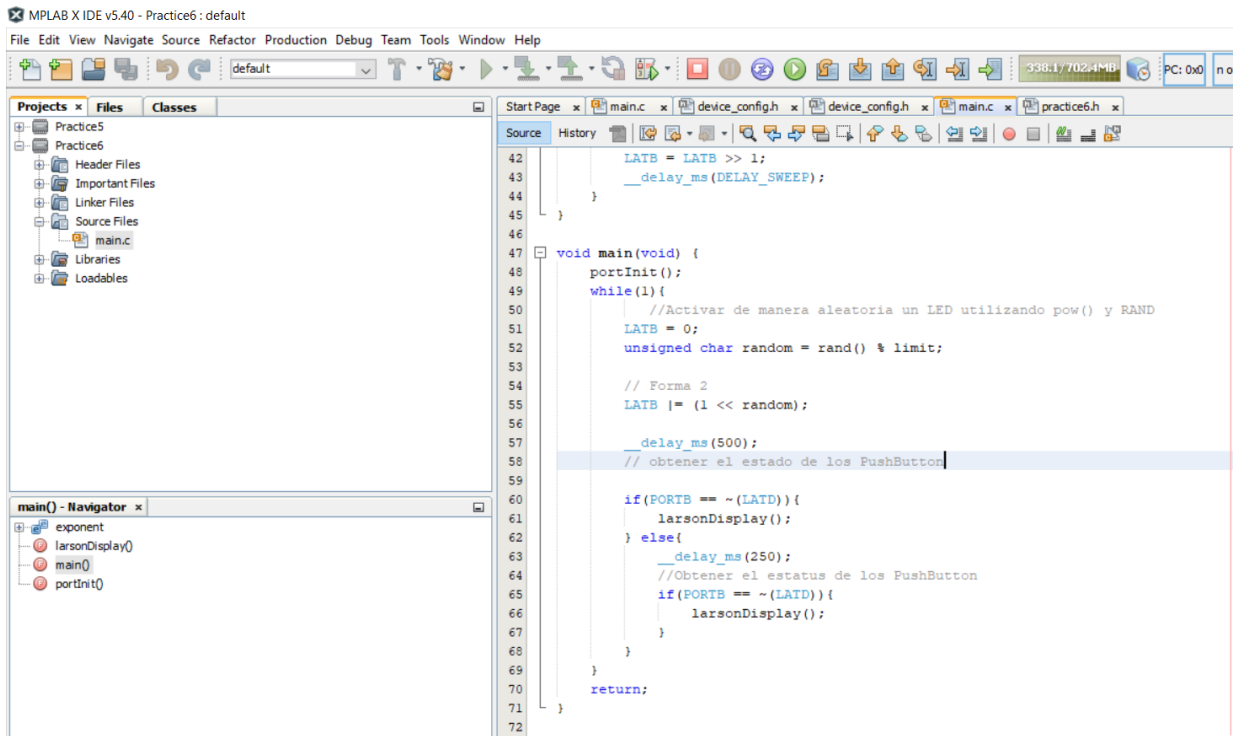
Reporte Práctica 6

Para esta práctica se realizó un programa de Mplab que simulaba ser un “whack a mole” el famoso juego en arcade en donde topes salen de unos agujeros y el jugador tiene que golpearlos a tiempo para hacer que se vuelvan a meter y ganar puntos. En este caso los “topes” fueron simulados con LEDs que se encendían y apagaban aleatoriamente y la forma de “golpearlos” era presionando algunos botones.

El código de la práctica fue el siguiente:



```
8
9 #include "device_config.h"
10 #include "practice6.h"
11 #include <math.h>
12
13 enum exponent{
14     bbase = 2,
15     limit = 8
16 };
17
18 void portInit(void) {
19     /**
20      *
21      * TRISDbits.TRISB0 = 1;
22      * TRISDbits.TRISB1 = 1;
23      * TRISDbits.TRISB2 = 1;
24      * TRISDbits.TRISB3 = 1;
25      * TRISDbits.TRISB4 = 1;
26      * TRISDbits.TRISB5 = 1;
27      * TRISDbits.TRISB6 = 1;
28      * TRISDbits.TRISB7 = 1;
29      *
30      */
31     ANSELB = 0;
32     ANSEL = 0;
33     TRISD = 255; // 11111111b - 0xFF
34 }
35
36 void laronDisplay(void){
37     LATB = 1;
38     for(unsigned char i = 0; i < limit; i++){
39         LATB = LATB << 1;
40         __delay_ms(Delay_Sweep);
41     }
42     for(unsigned char i = limit; i >= 0 ; i--){
43         LATB = LATB >> 1;
44     }
45 }
```



En el Main, teníamos las operaciones necesarias para encender y apagar LEDs de forma aleatoria por medio segundo, siempre preguntando por el estado del push button en caso de que el jugador lograra presionarlo a tiempo, si este fuera el caso, la funcion que entraba en juego era la que tenemos arriba llamada “larsenDisplay”.

Desafortunadamente debido a que no teníamos la licencia de Proteus disponible, no pudimos simular el código correctamente, sin embargo si pudimos probar que no hubiera errores de código ya que este compiló perfectamente como se puede ver a continuación:

