# **ECEN 215 – PRIN OF ELECTRICAL ENGR**

# **Fall 2018**

## **Lab 9: Introduction to Microcontrollers**



**Submitted by:**

|  |  |  |
| --- | --- | --- |
| **Student Name** | UIN: | **Section #** |
| Daniel Lopez | 426001591 | **516** |
| Adam Johnston | 516007336 |
| Jared Blunt | 524009660 |  |

**Date Performed: Feb 11th , 2019**

1. **Objective**

The objective of this lab was to install the Energia IDE for use with Texas Instrument microcontrollers and modify some existing code samples to familiarize ourselves with the C-based language that is compiled to these boards.

1. **Procedure**
2. We ran the example code “Blink” to make sure that the program was working along with the board we were given.
3. We modified the “Blink” program to make the board change the color of its lights.
4. We modified the “Blink2LED\_student.ino” program to use an integer counter to track iterations. For every iteration, we would turn on LED1 for 0.5s and off for 0.5s. Also for every 5th iteration, we turned on LED2 for 0.5s and off for 0.5s.
5. We modified the “Analog\_LED\_student.ino” program to use an integer counter to track PB1 and PB2 actuations (presses). For PB1 actuation, we increased x if x < 10, and for PB2 we decreased x if x > 0. Then we energized the LED to an intensity proportional to x (roughly between 0 and 255).
6. We modified the “Binary\_LED\_student.ino” program to use an integer counter to track the binary state of the 2-bit binary system LED1 represented the MSB (Most Significant Bit) and LED2 represents the LSB (Least Significant Bit). Then we used PB1 to move this binary system from state to state.
7. **Difficulties**
8. The program we used was very hard to work. We had to install several items to try and get the program to work, and nothing worked. Finally when we uninstalled and reinstalled everything the program started to work.
9. Keeping the wires inside the Analog Discovery.
10. **Results**

**Blink\_mod.ino**

|  |  |
| --- | --- |
|  |  |
|  | **// most launchpads have a red LED** |
|  | **#define LED RED\_LED** |
|  |  |
|  | **// the setup routine runs once when you press reset:** |
|  | **void setup() {** |
|  | **// initialize the digital pin as an output.** |
|  | **pinMode(LED, OUTPUT);** |
|  | **}** |
|  |  |
|  | **// the loop routine runs over and over again forever:** |
|  | **void loop() {** |
|  | **digitalWrite(LED, HIGH);** |
|  | **delay(500);** |
|  | **digitalWrite(LED, LOW);** |
|  | **delay(250);** |
|  | **}** |

**Blink2LED\_student.ino**

|  |  |
| --- | --- |
|  | **//see pins\_energia.h for more LED definitions** |
|  | **#define LED2 GREEN\_LED** |
|  |  |
|  | **int i = 0;** |
|  |  |
|  | **// the setup routine runs once when you press reset:** |
|  | **void setup() {** |
|  | **// initialize the digital pin as an output.** |
|  | **pinMode(LED1, OUTPUT);** |
|  | **pinMode(LED2, OUTPUT);** |
|  | **}** |
|  |  |
|  | **// the loop routine runs over and over again forever:** |
|  | **void loop() {** |
|  | **digitalWrite(LED1, HIGH);** |
|  | **delay(500);** |
|  | **if (i % 5 == 0)** |
|  | **{** |
|  | **digitalWrite(LED2, HIGH);** |
|  | **}** |
|  | **digitalWrite(LED1, LOW);** |
|  | **delay(500);** |
|  | **digitalWrite(LED2, LOW);** |
|  | **i++;** |
|  | **}** |

**Binary\_LED\_student.ino**

|  |  |
| --- | --- |
|  | **//see pins\_energia.h for more LED definitions** |
|  | **#define LED2 GREEN\_LED** |
|  | **int PB1 = PUSH1; // the number of the pushbutton pin** |
|  | **int PB2 = PUSH2; // the number of the pushbutton pin** |
|  | **int x = 0; //x is (the number of PB1 press/release cycles)%4** |
|  | **int x1 = 0; //x1 is the current state of PB1** |
|  | **int x2 = 0; //x2 is the current state of PB2** |
|  |  |
|  |  |
|  | **// the setup routine runs once when you press reset:** |
|  | **void setup() {** |
|  | **// initialize the digital pin as an output.** |
|  | **pinMode(LED1, OUTPUT);** |
|  | **pinMode(LED2, OUTPUT);** |
|  | **pinMode(PB1, INPUT\_PULLUP);** |
|  | **pinMode(PB2, INPUT\_PULLUP);** |
|  | **Serial.begin(9600); // opens serial port, data rate = 9600 bps** |
|  | **delay(10);** |
|  | **}** |
|  |  |
|  | **// the loop routine runs over and over indefinitely:** |
|  | **void loop()** |
|  | **{** |
|  | **while (digitalRead(PB1)); // pause program until PB1 is unpressed** |
|  |  |
|  | **if (x == 0)** |
|  | **{** |
|  | **digitalWrite(LED1, LOW);** |
|  | **digitalWrite(LED2, LOW);** |
|  | **}** |
|  |  |
|  | **else if (x == 1)** |
|  | **{** |
|  | **digitalWrite(LED1, LOW);** |
|  | **digitalWrite(LED2, HIGH);** |
|  | **}** |
|  |  |
|  | **else if (x == 2)** |
|  | **{** |
|  | **digitalWrite(LED1, HIGH);** |
|  | **digitalWrite(LED2, LOW);** |
|  | **}** |
|  |  |
|  | **else if (x == 3)** |
|  | **{** |
|  | **digitalWrite(LED1, HIGH);** |
|  | **digitalWrite(LED2, HIGH);** |
|  | **}** |
|  |  |
|  | **Serial.print("x = ");** |
|  | **Serial.println(x); // print as an ASCII-encoded decimal** |
|  |  |
|  | **x = x + 1; // increment x** |
|  | **x = x % 4; // compute the next 2-bit Binary state** |
|  |  |
|  | **while (!digitalRead(PB1)); // pause program until PB1 is pressed** |
|  |  |
|  | **}** |

**Analog\_LED\_student.ino**

|  |  |
| --- | --- |
|  | **// most launchpads have a red LED** |
|  | **#define RED RED\_LED** |
|  |  |
|  | **//see pins\_energia.h for more LED definitions** |
|  | **//#define LED2 GREEN\_LED** |
|  | **int x = 0; //x is the interation counter** |
|  | **int PB1 = PUSH1; // the number of the pushbutton pin** |
|  | **int PB2 = PUSH2;** |
|  |  |
|  |  |
|  | **// the setup routine runs once when you press reset:** |
|  | **void setup() {** |
|  | **// initialize the digital pin as an output.** |
|  | **pinMode(RED, OUTPUT);** |
|  | **// pinMode(LED2, OUTPUT);** |
|  | **pinMode(PB1, INPUT\_PULLUP);** |
|  | **pinMode(PB2, INPUT\_PULLUP);** |
|  |  |
|  | **Serial.begin(9600); // opens serial port, data rate = 9600 bps** |
|  | **delay(10);** |
|  | **}** |
|  |  |
|  | **// the loop routine runs over and over again forever:** |
|  | **void loop()** |
|  | **{** |
|  |  |
|  | **if (digitalRead(PB1) && x < 10)** |
|  | **{** |
|  | **x++;** |
|  | **}** |
|  | **if ((digitalRead(PB2)) && x > 0)** |
|  | **{** |
|  | **x--;** |
|  | **}** |
|  | **analogWrite(RED, 25 \* x);** |
|  | **Serial.println(x); // print as an ASCII-encoded decimal** |
|  | **delay(500); // wait for a 0.5 second** |
|  | **}** |

1. **Conclusion**

Overall the main difficulty we had was installing all the proper drivers for the Energia IDE to be compatible with our version of the Launchpad. Once this was done, we found the syntax to be identical to the C-based language used for Arduino boards, making the coding aspect of the lab rather simple. If we were to change anything in the code, it would be the debouncing of the button in the Analog\_LED\_student.ino file to be less than half a second to make the dimming of the LED more responsive.