Brian Sheridan

Lab 1: Flashing Lights

Performed on 4/09/15

Submitted on 4/16/15

Objective:

The objective of the lab was to learn how to create a project in Code Blocks for Arduino or the official Arduino IDE and to use the GPIO pins on the Arduino microcontroller to control a 3-color light emitting diode. The Arduino's libraries allowed for easy control to turn on and off the LEDs.

Introduction:

The Arduino Uno microcontroller is an Atmel ATmega328 integrated onto a developer board to allow for easy prototyping with the chips many pins. The ATmeaga328 has 6 analog inputs, 14 digital inputs/outputs along with 3.3v, 5v and ground pins. Arduino also has a integrated development environment (IDE) for quick prototyping and coding which allows one to quickly design and test a project.

A three LED shield was used with the Arduino to investigate how control statements and flow works. The LED was powered by the 5v pin and also contacted to 3 analog pins, A0, A1 and A2. Through the use of the digitalWrite library, the code was able to selectively turn on each LED through setting the pin to High or Low.

The development environment of choice for the lab was Code Blocks for Arduino or the official Arduino IDE. These two programs syntax highlighting of the code to indicate possible errors before the code is compiled. Once the code is finished and compiled, the Arduino is programmed over a serial connection through a USB cable.

The Arduino uses two control loops to run, setup() and loop(). Setup() contains all the initialization in which the code needs to run to begin the program. Loop() is a never ending loop that continuously runs through the code within it till the micrcontoller is turned off.

Procedure:

Part 1 of the lab consisted of setting up the development environment and running a HelloWorld program. The development environment used was the offical Arduino IDE. A new project was created through “create Project” tile. It was given the name “Hello World”. The IDE automatically generated a base code to begin the project.This code blinked LED 13 on and off every second while printing “Hello World!” on the serial promp. The example included with the Arduino IDE creates the following code:

Hello World code:

```
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// 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);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
  Serial.println("Hello World!");
}
```

part 2 of the lab worked on controlling the color of the RGB LED. The 3 pins were connected to analog A0, A1, and A2 which allowed us to change the color through control the power on each of the pins. Red was connected to Pin A0, Green to A1 and Blue to A2. The code was written to turn on each color individually, wait for 1 second and then proceed to the next color. Since this code was done in the loop() it continues forever. The RGB code is below:

RGB Flasher Code:

```
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(A0, OUTPUT);
  pinMode(A1, OUTPUT);
  pinMode(A2, OUTPUT);
  Serial.begin(9600);
}


// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(A0, LOW);
  digitalWrite(A1, HIGH);
  digitalWrite(A2, HIGH);
  delay(1000);
  digitalWrite(A0, HIGH);
  digitalWrite(A1, LOW);
  digitalWrite(A2, HIGH);
  delay(1000);
  digitalWrite(A0, HIGH);
  digitalWrite(A1, HIGH);
  digitalWrite(A2, LOW);
  delay(1000);
}
```

Part 3 of the lab was to turn the LED on in the same order as a traffic light, add a specified delay between each color change and loop indefinetly. There was a 30 second hold on the green, followed by a 15 second yellow, and finally a 30 second red light. This would then loop and continue forever. The code for the Trafflic light is below:

TrafficLight Code:

```
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(A0, OUTPUT);
  pinMode(A1, OUTPUT);
  pinMode(A2, OUTPUT);
  Serial.begin(9600);
}


// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(A0, LOW);
  digitalWrite(A1, LOW);
  digitalWrite(A2, LOW);

  digitalWrite(A1, LOW);
  digitalWrite(A0, HIGH);
  digitalWrite(A2, HIGH);
  delay(30000);
  digitalWrite(A0, LOW);
  digitalWrite(A1, LOW);
  digitalWrite(A2, HIGH);
  delay(15000);
  digitalWrite(A0, LOW);
  digitalWrite(A1, HIGH);
  digitalWrite(A2, HIGH);
  delay(30000);
}
```

Results:

Part 1 of the had a create a simple helloWorld application that lit up LED 13 on and off every second. It also printed “Hello World” to the serial terminal every second. It guided us through how to setup the script in the IDE and how to compile and load the code onto the microcontroller. The code has use change the delay statement in the code from 1000 microseconds to 250 microseconds. This caused the LED to blink much faster, at 1/4th of the rate of the previous code. It also the Hello World to be printed 4 times faster than the previous example.

Part 2 of the lab worked exactly as detailed in the lab instructions. The light intially blinked green as A1 and A2 were driven High. A one second delay would be called, followed by setting A0 high and A1 Low and A2 High. This changed the LED to blue. Another one second delay would be called and A0 would be set high, A1 High and A2 Low. This produced the red color in the LED. Another one second delay was called and the program began to loop again.

Part 3 of the lab was a bit more difficult. It had the same setup as the previous lab. The assignment

wanted us to simply change the color and length between each color change. The first color assigmnet was the same as the previous lab and had A1 and A2 high to produce green. A delay for 30000 microseconds was called to hold the green light for 30 seconds. Then A2 was set to high and the rest to low to create yellow. This part was a bit tricker as it recreated a bit of trial and error to correctly determine which combination of outputs created yellow on the LED. A delay for 15000 microseconds was then called to hold the light yellow for 15 seconds. Finally A1 and A2 were set to high and A0 to low to create Red, the same as before in the previous lab. Finally another delay was added to hold the light for 30 seconds. This then repeated forever as the code was executed in the loop() function.

Conclusion:

Overall this lab has showed how to use the basic features of the Arduino IDE and program the Arduino Uno microcontroller. It also has shown how to selectively turn on and off specific GPIO pins in order to achieve a specific result, and how to use the delay command to force the program to wait. Overall this lab has achieved those goals and I have learned how to program the Arduino in basic Arduino C and effectively utilize the GPIO to display specified colors on our tri-color LED shields.