Lab 1 is an introduction to the basics on how to use a Arduino as well as understanding code and writing code on our own. In this lab we learned about *void setup(), void loop(),* and *Bitwise Operators.* Learning how to set, reset and toggle values in each pin in PORT. I well talk about the lab circuits more in detail later but there are diagrams in the *Microcontrollers Laboratory Manual* on page 10 & 11 which gives more details. In Lab 1, circuit 1 we are told to download the file *lab1\_circuit1.ino* from Blackboard, which was written by Doctor Alyssa J. Pasquale, that file had a code that makes the LED to blink. In Lab 1, circuit 2 we are told to use the same code in circuit 1 and change it to toggle the LED on and off instead of turning the LED on and off. In Lab 1, circuit 3 we are told to write our own code for a traffic light on a main street and side street. One of the main or side street traffic light is either is green or yellow while the opposite street traffic light is red then the traffic lights switch places and loop over again.

In Lab 1, circuit 1 we where give the code on Blackboard, which was written by Doctor Alyssa J. Pasquale, which turn on and off the LED. In the *void setup()* we configured pin 7D as an output on the Arduino { DDRD = DDRD | 0b10000000;}. In the *void loop ()* our first line of code sets PORTD bit 7 by putting it through an OR gate to set the value to high { PORTD = PORTD | 0b10000000;}. The second line of code we put a delay for 1 second { \_delay\_ms(1000);}. The third line of code clears PORTD bit 7 by putting it through an AND gate to reset the value to low { PORTD = PORTD & 0b01111111; }. The fourth and final line of code we put a delay for 1 second { \_delay\_ms(1000);} then it restarts the *void loop()* all over again causing it to set to turn on and clears it to turn off the LED every second.

In Lab 1, circuit 2 we use the same code in circuit 1 and change it to toggle the LED on and off instead of turning the LED on and off. In the *void setup()* we configured pin 7D as an output on the Arduino { DDRD = DDRD | 0b10000000;}. In the *void loop ()* our first line of code toggles PORTD bit 7 by putting it through an XOR gate to toggles the value { PORTD = PORTD ^ 0b10000000;}. The second line of code we put a delay for 1 second { \_delay\_ms(1000);} then it restarts the *void loop()* all over again causing it to toggle on and off the LED every second. This circuit is much easier to code and uses less program memory then circuit 1 although you can’t tell the difference between circuit 1 and circuit 2 without looking at the code.

In Lab 1, circuit 3 we write our own code for a traffic light on a main street and side street. When one of either the main or side street lights are green on it will wait 5 seconds then the traffic light will turn green to yellow and stays yellow for 2 seconds while the opposite street is red. In the *void setup()* we configured D as an output on the Arduino { DDRD = DDRD | 0b11111100;}. The function of each in PORTD is D7= Main Red, D6= Main Yellow, D5= Main Green, D4= Side Red, D3= Side Yellow, D2= Side Green, D1 & D0= Don't Care. In the *void loop ()* our first line of code turns on Main street light green, turn on Side street light red, and turn off all others by putting it through an OR gate to set value D7 and D4 to high { PORTD = PORTD | 0b00110000;}. The second line of code we put a delay for 5 seconds { \_delay\_ms(5000);}. The third line of code turns on Main street yellow, turn on Side street red, and turn off all others by putting it through an XOR gate to toggle value D6 to high, and toggle value D7 low { PORTD = PORTD ^ 0b01100000; }. The fourth line of code we put a delay for 2 seconds { \_delay\_ms(2000);}. The fifth line of code turns on Main street light red, turn on Side street light green, and turn off all others by putting it through an XOR gate to toggle values D5 and D2 to high, and toggle value D6 and D4 to low { PORTD = PORTD ^ 0b11010100; }. The sixth line of code we put a delay for 5 seconds { \_delay\_ms(5000);}. The Seventh line of code turns on Main street light red, turn on Side street light yellow, and turn off all others by putting it through an XOR gate to toggle values D3 to high, and toggle value D2 to low { PORTD = PORTD ^ 0b00001100; }. The eighth line of code we put a delay for 2 seconds { \_delay\_ms(2000);}. The ninth line of code made to reset the traffic lights before loop { PORTD = PORTD ^ 0b10111000; }. The Tenth line of code we put a delay for 100 milliseconds { \_delay\_ms(100);} then it restarts the *void loop()* of the traffic light. The microcontroller is way easier than a purely hardware because when you purely hardware it requires a lot of different components and a lot of time. While the microcontroller requires less components and a lot less time. Arduino BEST THING IN LIFE!!!

The most challenging part of this lab was resetting the lights on the traffic light before it loops. When I was testing out the circuit 3 at home the circuit was looking all good until it looped. Before it looped it worked as planed but as soon as it looped the red light on the Main street and the green light on the Side street stayed on when it looped back to the beginning then turned off when I wanted it to turn on. So, I decided to put an XOR gate to toggle off the red light on the Main street and the green light on the Side street before it looped so that fixed the problem in with the traffic light. I know that you can fix it another by use an AND gate to clear all the pins would also fix the problem but I choice an XOR gate because I like XOR more and if I used the AND gate you would see the traffic light soddenly turn off all the lights for about 100 milliseconds so that is why I used the XOR gate instead.

In conclusion is lab taught that there are many ways to code something to do that exact same thing. Sometimes there are easier ways to code something to use less program memory and/or become faster. Also, microcontrollers are great tools to use to which makes life better because it least cost effective and least time consuming then purely hardwiring it by hand. So Arduino is life, Arduino is love.

