Lab 2 is a practicing of the code as while as learning how to use new code. The new code we learned in Lab 2 code like Boolean AND, Boolean OR, Boolean NOT, Datatypes: int, Variable Scope, Comparison Operators, Digital Inputs, Analog Inputs, Conditional Statements; If/Else, while and do/while. I well talk about the lab circuits more in detail later but there are diagrams in the *Microcontrollers Laboratory Manual* on page 26, 27, & 28 which gives more details. In Lab 2, circuit 1 we were give the code on Blackboard, which was written by Doctor Alyssa J. Pasquale, which turns on and off an LED using a pushbutton. In Lab 2, circuit 2 we wrote our own code where we choose two logic functions, we choose OR and XOR, then create a programmable logic gate that toggles between logical gate functions. Also in Lab 2, circuit 2 the if statement well have a different number at the end of the word “if statement” to indicate what if statement I am talking about. Example : “if the if statement(#) is true” or “if the if statement(#) is false”. In Lab 2, circuit 3 we were give part of the code on Blackboard, which was written by Doctor Alyssa J. Pasquale, and which we modified the code to blink the LED depending on the analog value of a potentiometer.

In Lab 2, circuit 1 we were give the code on Blackboard, which was written by Doctor Alyssa J. Pasquale, which turns on and off an LED using a pushbutton. In the *void setup()* we set DDRB bit 5 which causes pin D13 on the Arduino to be an output { DDRB = DDRB | 0b00100000;}. In the *void loop ()* our first line of code makes D8 an input that reads the value from the pushbutton { unsigned char buttonState = PINB & 0b00000001; }. Our second line of code is the beginning of an If/Else conditional statements if the pushbutton is pressed it is true { if (buttonState == 1) {} }. Our third line of code turn on the LED { PORTB = PORTB | 0b00100000; }. Our fourth line of code is the ending of the If/Else conditional statements if the pushbutton isn’t pressed it is false { else {} }. Our fifth line of code turned off the LED { PORTB = PORTB & 0b11011111; }.

In Lab 2, circuit 2 we wrote our own code where we choose two logic functions, we choose OR and XOR, then create a programmable logic gate that toggles between logical gate functions. In the *void setup()* we set DDRB bit 5 which causes pin D13 on the Arduino to be an output { DDRB = DDRB | 0b00100000;}. In the *void loop ()* our first line of code reads the value from the dipswitch { unsigned char ToggleState = PINB & 0b00000100; }. Our second line of code is a togglestate which makes it an OR gate if the if statement is true but if the if statement(1) is false it turns the circuit into a XOR gate { if (ToggleState == 4) { }. If the if statement (1) is true our third line of code makes D8 & D9 an input that reads the value from the pushbutton { unsigned char buttonState = PINB & 0b00000011; }. If the if statement (1) is true our third line of code is another if statement (2) for the OR logic gate function where if the buttonstate is great than zero it is true or if the buttonstate is equal to zero the if statement (2) is false { if (buttonState > 0) {} }. If the if statement (2) is true our fourth line of code turns on the LED { PORTB = PORTB | 0b00100000; }. If the if statement (2) is false our fifth line of code turns off the LED { PORTB = PORTB & 0b11011111; }. If the if statement (1) is false our sixth line of code makes D8 & D9 an input that reads the value from the pushbutton { unsigned char buttonState = PINB & 0b00000011; }. Our seventh line of code is another if statement (3) for the XOR logic gate function where if the buttonstate is equal to 1 it is true but if it is something else it would be false { if (buttonState == 1) {} }. If the if statement (3) is true our eghth line of code turns on the LED { PORTB = PORTB | 0b00100000; }. If the if statement (3) is false our ninth line of code is another if statement (4) for the XOR logic gate function where if the buttonstate is equal to 2 it is true but if it is something else it would be false { else if (buttonState == 2) {} }. If the if statement (4) is true our tenth line of code turns on the LED { PORTB = PORTB | 0b00100000; }. If the if statement (4) is false our eleventh line of code turns off the LED { PORTB = PORTB & 0b11011111; }.

In Lab 2, circuit 3 we were give part of the code on Blackboard, which was written by Doctor Alyssa J. Pasquale, and which we modified the code to blink the LED depending on the analog value of a potentiometer. In the *void setup()* our first line of code initialize ADCSRA with prescaler of 128 { ADCSRA = 0xEF; }. Our Second line of code initialize ADMUX with reference voltage of AVcc { ADMUX = 0x40; }. Our third line of code initialize ADCSRB into free-running mode { ADCSRB = 0x00; }. Our fourth line of code configure digital pin D3 as an output { DDRB = DDRB | 0x04; }. Our fifth line of code configure analog pin A0 as an input { Configure your anloge pin as an input }. Before the *void loop ()* we assigned the result equal to 0 as a volatile unsigned 16-bit { volatile unsigned int result = 0;

}. In the *void loop ()* our first line of code is to toggle the LED on and off on pin D3 { PORTD = PORTD ^ 0x04; }. Our second line of code we assigned the i equal to 0 as an unsigned 16-bit { unsigned int i=0; }. Our third line of code if while is true it will follow the rest of the command { while(i < result){} }. Our fourth line of code is to wait 1000 unseconds (1 millisecond) { \_delay\_us(1000); }. Our fifth line of code is to force i to a high value { i=i+1; }. The *ISR()* this function automatically happens whenever the ADC has completed a loop { ISR(ADC\_vect) {} }. In the *ISR()* our first line of code assigns ADCL as the least significant unsigned 8-bit { unsigned char LSByte = ADCL; }. Our second line of code assigns ADCH as the most significant unsigned 8-bit { unsigned char MSByte = ADCH; }. Our third line of code assigns the result depending on the value of ADLAR, my value was low, { result = ADCL + (ADCH << 8); }. If my ADLAR value was high the equation would have been { result = (ADCL >> 6) + (ADCH << 2); }.

There weren’t many challenges in Lab 2, circuit 1 and circuit 2, I felt like i understood what to do, but in circuit 3 I didn’t understand and/or got confused. First thing I thought was that the code was complete as well as we just needed to download it the Arduino and wire it up. By looking closely inside the microcontrollers laboratory manual, microcontrollers book, and pictures of the teachers notes in class I found my answers. Like in what the equation for the “result”. I found the answer in one of the picture of the teacher notes.

In conclusion lab 2 I got a better grasp on the code as well as great practice for both new and old codes I learned. I learnt about how to convert an analog input as a digital input. I had did get confused at the end of circuit 3 but I learnt from it as well as got better at it. Greatly improving my skills.