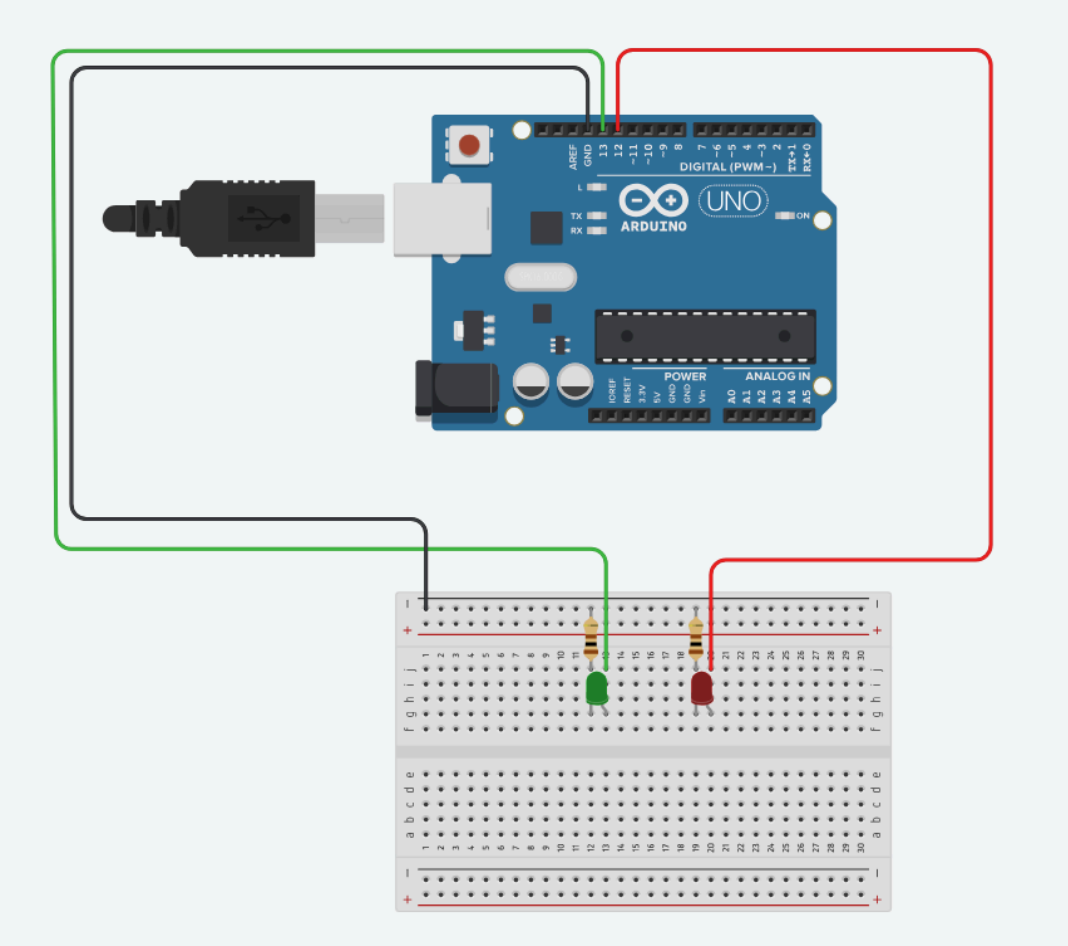
Class:	CPE301L Digital Systems Architecture and Design 1001		Semester:	Spring 2025
Points		Document author:	Narek Kalikian	
		Author's email:	kalikn1@unlv.nevada.edu	
		Document topic:	Postlab 1	
Instructor's comments:				

1. Introduction / Theory of Operation

In this lab, I used Tinkercad to simulate various Arduino projects. Each experiment required using different hardware components on a breadboard connected to an Arduino Uno. Additionally, I worked on some basic assembly programming.

2. Description of Experiments

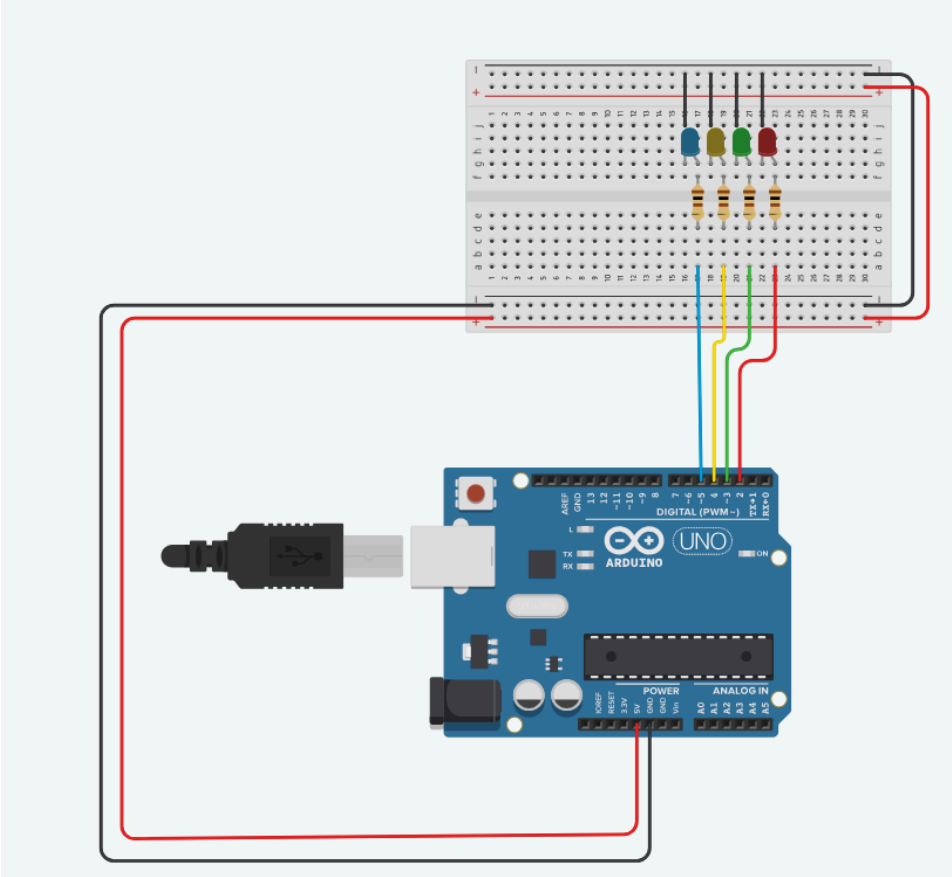
Experiment 1 - Circuit:



Experiment 1 - Code:

```
void setup()
{
 pinMode(13, OUTPUT);
 pinMode(12, OUTPUT);
}
void loop()
{
 // First LED ON
 digitalWrite(13, HIGH);
 // Second LED OFF
 digitalWrite(12, LOW);
 delay(1000); // Wait for 1000 milliseconds
 // First LED OFF
 digitalWrite(13, LOW);
 // Second LED ON
 digitalWrite(12, HIGH);
 delay(1000); // Wait for 1000 milliseconds
}
```

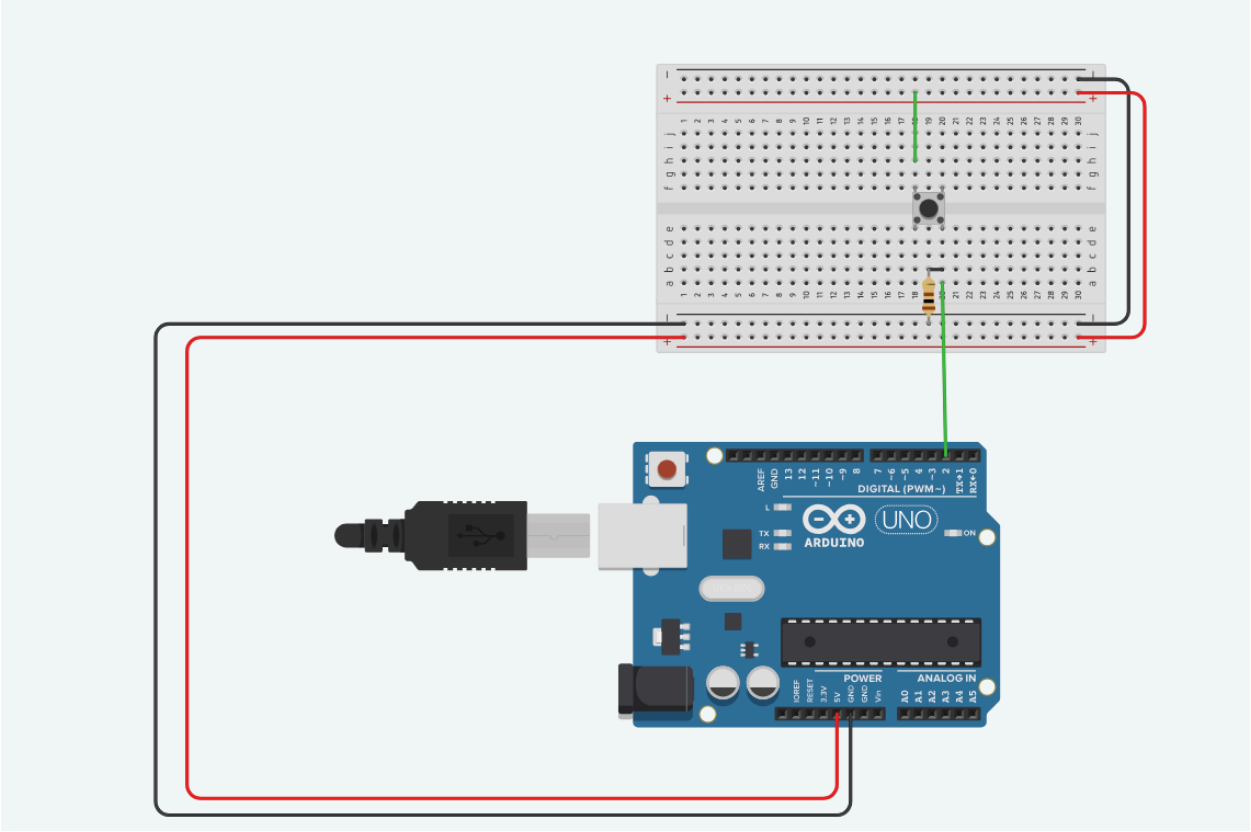
Experiment 2 - Circuit:



Experiment 2 - Code:

```
const int LED[] = {2, 3, 4, 5}; // LED pins
// Global variable
int pos = 0;

void setup() {
    for (int i = 0; i < 4; i++) {
        pinMode(LED[i], OUTPUT);
    }
}

void loop() {
    // All LEDs OFF
    for (int i = 0; i < 4; i++) {
        digitalWrite(LED[i], LOW);
    }

    // LEDs ON
    switch (pos) {
        case 0: digitalWrite(LED[0], HIGH);
        break;
        case 1: digitalWrite(LED[1], HIGH);
        break;
        case 2: digitalWrite(LED[2], HIGH);
        break;
        case 3: digitalWrite(LED[3], HIGH);
        break;
    }

    delay(1000); // 1000 ms wait

    // Increment pos using mod
    pos = (pos + 1) % 4;
}
```

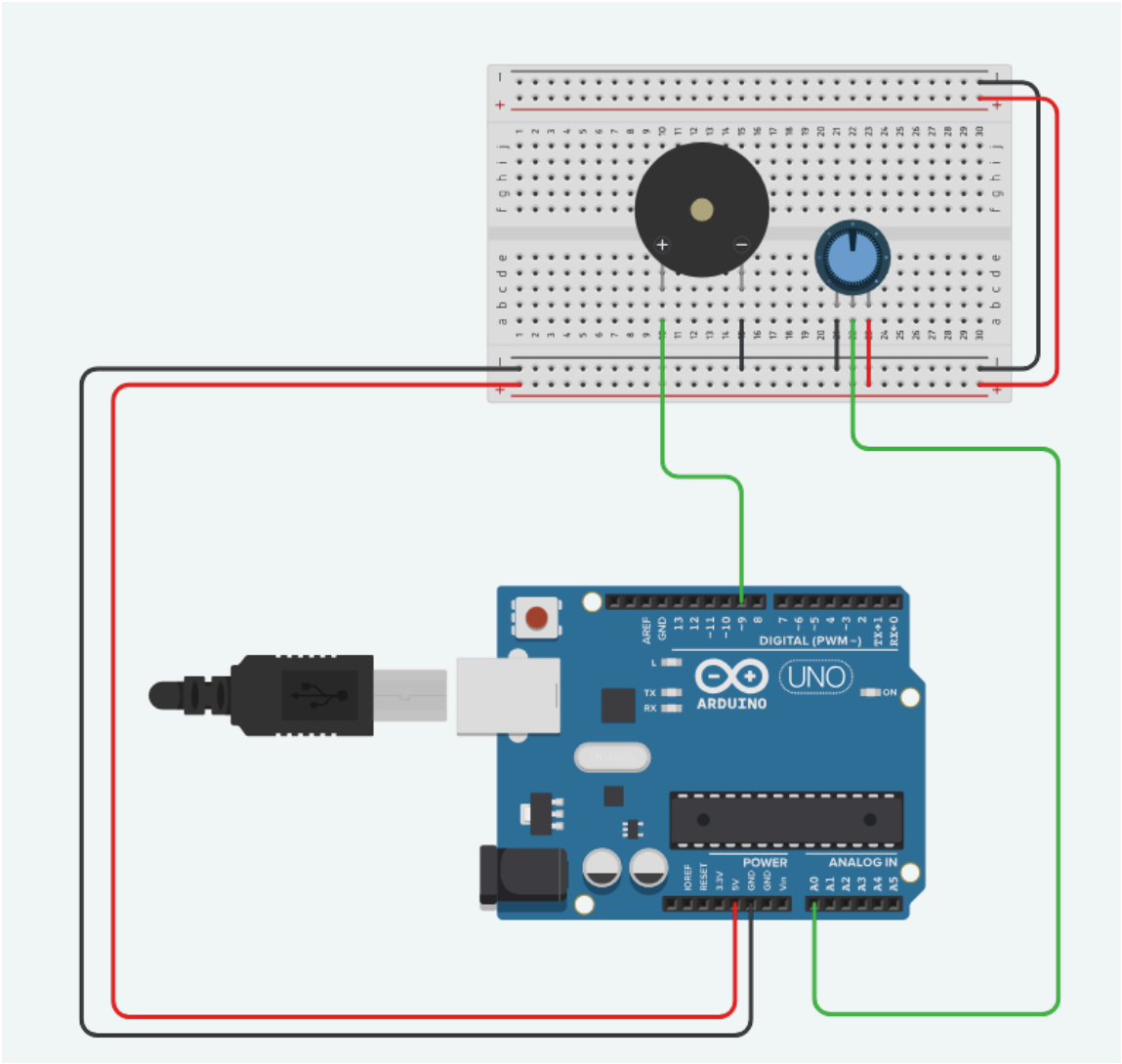
Experiment 3 - Circuit:



Experiment 3 - Code:

```
const int buttonPin = 2;  // Button pin
int buttonState = 0;      // Variable for button on/off

void setup() {
  pinMode(buttonPin, INPUT);
  Serial.begin(9600);          // 9600 baud
}

void loop() {
  buttonState = digitalRead(buttonPin);  // Button state

  if (buttonState == HIGH) {  // Check if button pressed
    Serial.println("Button pressed");  // Print message if button pres
  }

  delay(100);  // Delay
}
```

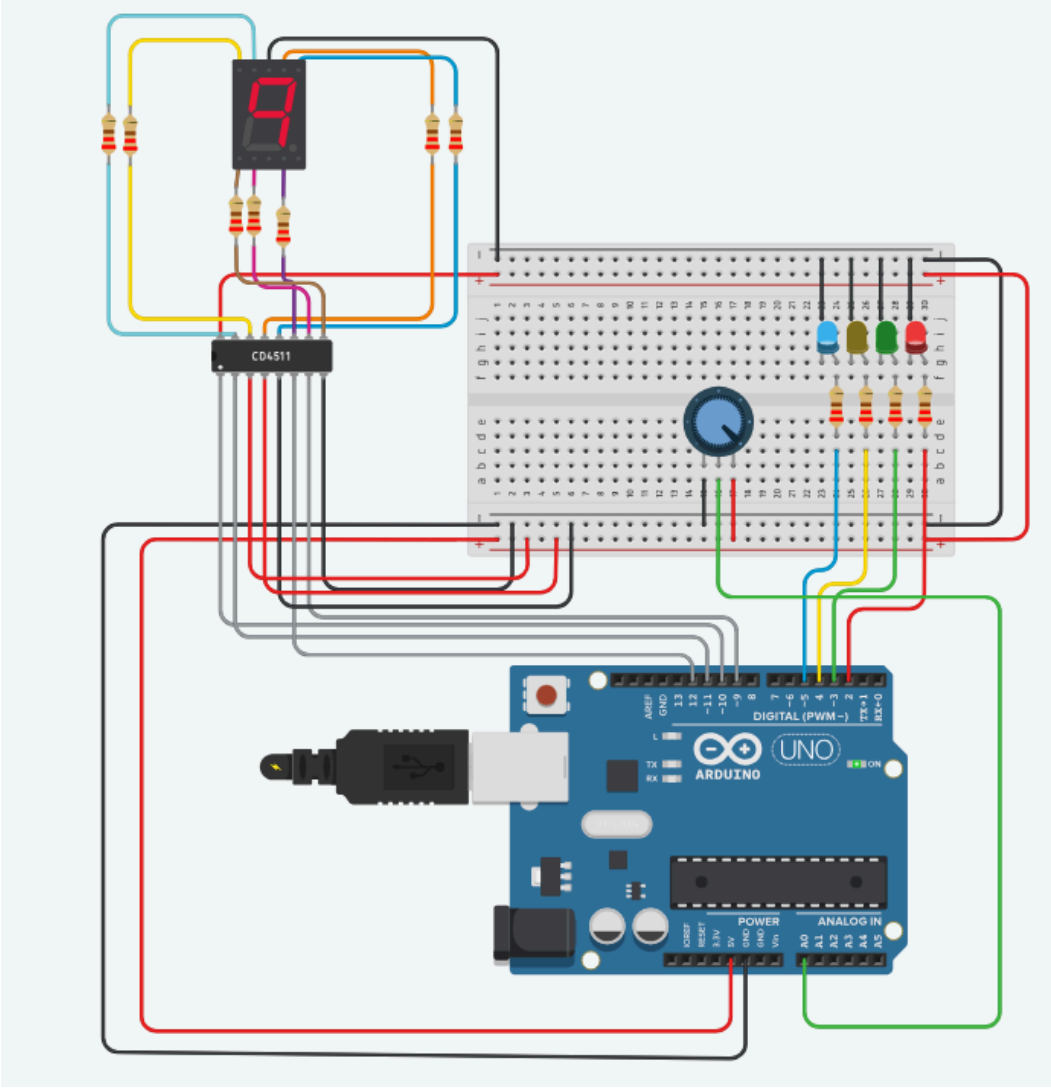
Experiment 4 - Circuit:



Experiment 4 - Code:

```
const int potPin = A0;  // Potentiometer pin
const int buzzerPin = 9;  // Buzzer pin
int prevPotValue = -1;  // Previous pot val variable

void setup() {
  Serial.begin(9600);  // Serial baud 9600
  pinMode(buzzerPin, OUTPUT);
}

void loop() {
  int potValue = analogRead(potPin);  // Read pot val

  // If pot val changes, display msg in serial monitor
  if (potValue != prevPotValue) {
    Serial.print("Potentiometer Value: ");
    Serial.println(potValue);
    prevPotValue = potValue;  // Update previous pot val
  }

  // Map analog range to readable potentiometer values
  int buzzerFreq = map(potValue, 0, 1023, 200, 2000);
  tone(buzzerPin, buzzerFreq);  // Drive buzzer using pot val

  delay(100);  // Delay
}
```

Experiment 5 - Circuit:



Experiment 5 - Code:

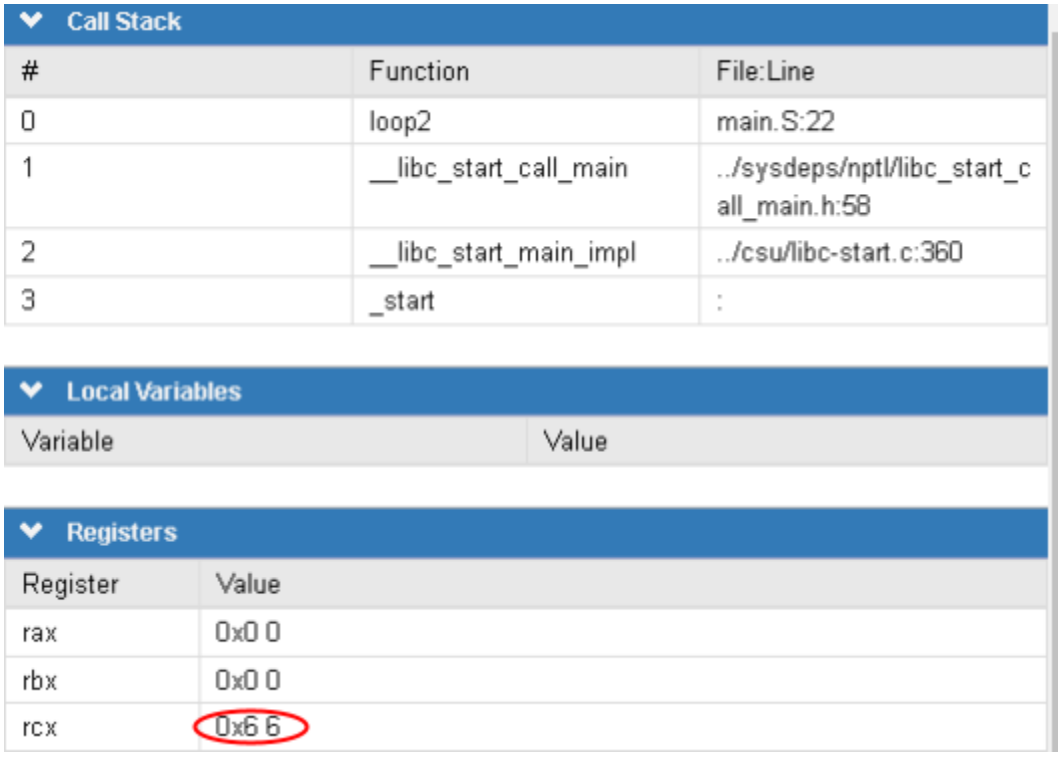
```
int potPin = A0;
int digit = 0;
int A = 9, B = 10, C = 11, D = 12;  // CD4511 Pins
int LED0 = 2, LED1 = 3, LED2 = 4, LED3 = 5; // LEDs

void setup() {
    pinMode(A, OUTPUT);
    pinMode(B, OUTPUT);
    pinMode(C, OUTPUT);
    pinMode(D, OUTPUT);
    pinMode(LED0, OUTPUT);
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(LED3, OUTPUT);
    Serial.begin(9600);
}

void control7SEG(int num) {
    // Use bitRead()
    digitalWrite(A, bitRead(num, 0)); // LSB
    digitalWrite(B, bitRead(num, 1));
    digitalWrite(C, bitRead(num, 2));
    digitalWrite(D, bitRead(num, 3)); // MSB

    digitalWrite(LED0, bitRead(num, 0));
    digitalWrite(LED1, bitRead(num, 1));
    digitalWrite(LED2, bitRead(num, 2));
    digitalWrite(LED3, bitRead(num, 3));
}


void loop() {
    int potValue = analogRead(potPin); // Potentiometer value
    digit = map(potValue, 0, 1023, 0, 9); // Map pot val
    Serial.print("Digit: ");
    Serial.println(digit);
    control7SEG(digit); // Display on 7 seg
    delay(100);
}
```

Experiment 6 - Code:

main.S

```
.data
.text
.global main
main:
    xor %rax, %rax          # RAX = 0
    xor %rbx, %rbx          # RAX = 0
    xor %rcx, %rcx          # RCX = 0

    add $3, %rax            # RAX = RAX + 3

loop1:
    add $2, %rbx            # RBX = RBX + 2

loop2:
    inc %rcx                # RCX + 1
    dec %rbx                # RBX - 1
    jnz loop2               # Keep looping if RBX != 0

    dec %rax                # RAX - 1
    jnz loop1               # Keep looping if RAX != 0

    ret                     # End
```

Experiment 6 - Debug Mode Result in RCX Register:



Call Stack

#	Function	File:Line
0	loop2	main.S:22
1	__libc_start_call_main	../sysdeps/nptl/libc_start_call_main.h:58
2	__libc_start_main_impl	../csu/libc-start.c:360
3	_start	:

Local Variables

Variable	Value

Registers

Register	Value
rax	0x0 0
rbx	0x0 0
rcx	0x6 6

*I set the outside loop controller, RAX, to 3 and the inside loop controller, RBX, to 2. Per the lab manual, the RCX register contains the value of RAX * RBX after running the debugger (which is 6 in the case of my code).*

3. Questions and Answers

Question 1 - Arduino Digital vs Analog: Analog pins read values based on voltage levels from 0 volts to 5 volts. These values are mapped to analog values from 0 to 1023. Digital pins have two voltage levels: 0 volts (LOW) and 5 volts (HIGH). This is good for the triggering of relays. Digital pins can be used to output power to components and also be used as inputs to read data from peripheral devices.

Question 2 - Bitwise Operations: A bitwise operation is an operation that works on the individual bits of a binary number. This allows you to perform logical operations bitwise between two operands and apply boolean operators by converting decimal numbers to binary.

4. Conclusions

In this lab, I was able to refresh my skills in programming and wiring Arduino projects while also learning more about programming in assembly. Simulating the projects virtually on Tinkercad is a good place to start before working with real Arduino boards.