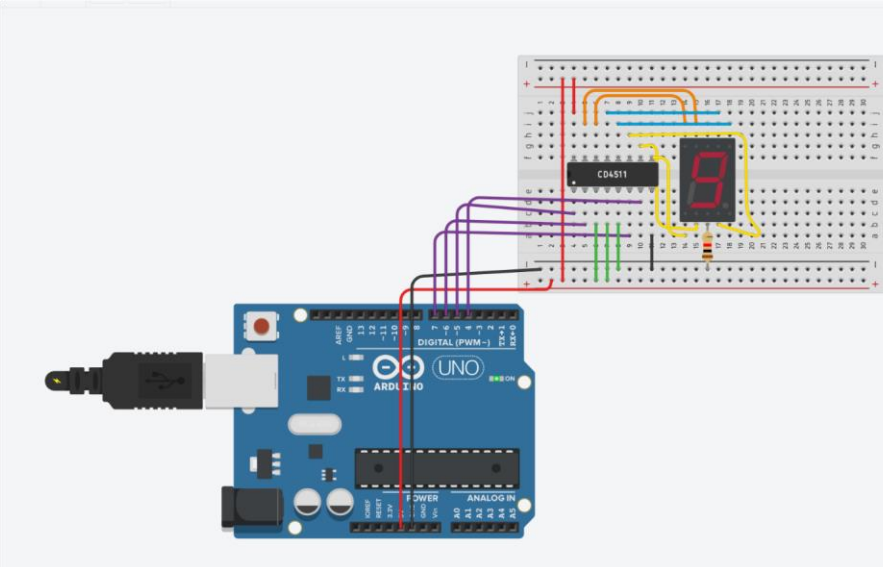
LAB REPORT #7

Name: Kenilkumar Patel, Aaryaman Sharma
Date: 4/20/2021
Screenshot and components



Segment Display: This can be used to display numbers by turning on or off 7 LEDS. Each LED is controlled by a separate digital input.
Segment Decoder: This further simplifies the number of pins to control the LEDs from 7 to 4. The decoder inputs the value for the 7 original pins with a circuit that maps the 4 inputs to the correct outputs.

Summary:
First I assemble the hardware. Then I copied over the code and did a test run. The 7 segment display showed a number signalling the hardware assembling works. Instead of controlling 7 pins, I only need to control 4 pins to produce numbers from 0 - 9. The 4 pins are the digital inputs 4 to 7 on the arduino, which can be turned on or off, and such combinations can create a binary number which will be converted to a decimal from 0-9. The goal to print 0 to 9 in a loop on the display. This can be done by knowing what binary corresponds to each decimal. Thus for every decimal, I must turn on or off each of the 4 pins to print the decimal.
Result:

I hard coded the pins needed to print out every decimal from 0 - 9 using the binary combinations of pins 4-7. The result worked as the display numbers from 0 to 9 and repeat again for every void loop().

Conclusion: I learned how the Segment display works and how the 7 LEDs can be controlled by the pins. I learned the numbers of pins can be reduced from 7 to 4 through the segment decoder as the 4 pins can act as a binary number by turning on or off. The big mistake I made was not turning every pin off again for every decimal in the for loop. This would cause a problem because if I turned on one pin, that pin would still stay on the next for loop. By resetting the pins to off every for loop, I would not have to worry about the next number being wrong. int bits[] = {4,5,6,7}; //bit zero is bits[0]

```
void setup()
{
        Serial.begin(9600); for(int i = 4; i<= 7; i++) {
                //initalize pins [4,7] pinMode(i,
        OUTPUT);
        }
}

void loop()
{

        for(int i = 0; i < 10; i++) {      //displaying numbers 0 to
    9 if (i == 0){ i = 0;}
    if (i == 1){ digitalWrite(bits[0],
     OUTPUT);} if (i == 2){
      digitalWrite(bits[0], INPUT);
     digitalWrite(bits[1], OUTPUT);}
    if (i == 3){
      digitalWrite(bits[0], OUTPUT);
     digitalWrite(bits[1], OUTPUT);}
    if (i == 4){
      digitalWrite(bits[0], INPUT);
     digitalWrite(bits[1], INPUT);
        digitalWrite(bits[2], OUTPUT);}
  if (i == 5){ digitalWrite(bits[0],
      OUTPUT);
     digitalWrite(bits[1], INPUT);
        digitalWrite(bits[2], OUTPUT);}
        if (i == 6){
      digitalWrite(bits[0], INPUT);
     digitalWrite(bits[1], OUTPUT);
    digitalWrite(bits[2], OUTPUT);} if (i
    == 7){
      digitalWrite(bits[0], OUTPUT);
     digitalWrite(bits[1], OUTPUT);
    digitalWrite(bits[2], OUTPUT);} if (i
    == 8){
      digitalWrite(bits[0], INPUT);
     digitalWrite(bits[1], INPUT);
        digitalWrite(bits[2], INPUT);
        digitalWrite(bits[3], OUTPUT);}
      if (i == 9){
         digitalWrite(bits[0], OUTPUT);
        digitalWrite(bits[1], INPUT);
        digitalWrite(bits[2], INPUT);
      digitalWrite(bits[3], OUTPUT);}


/* here you will need to add logic to figure out what bits to turn on and off
 * refer to the table in the instructions to know what bits to turn on and off*/

/* example of turning 0 bit on : digitalWrite(bits[0], HIGH);
 * example of turning 0 bit off: digitalWrite(bits[0], LOW ); */


                        delay(500); //1/2 a second delay so that we can see the numbers changing
      digitalWrite(bits[0], INPUT);
        digitalWrite(bits[1], INPUT);
            digitalWrite(bits[2], INPUT);
      digitalWrite(bits[3], INPUT);
            }
}
```