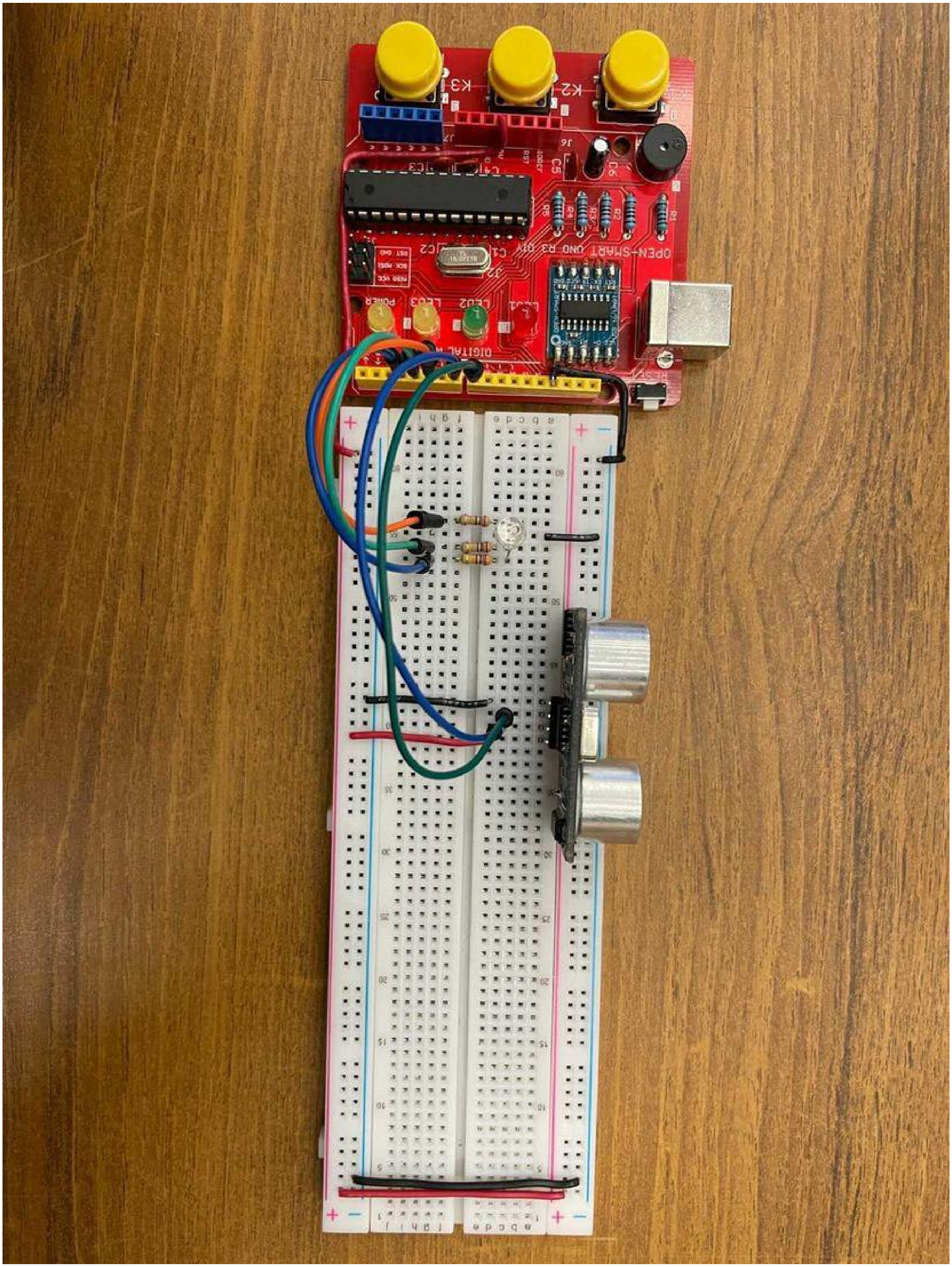
Interfacing Lab 5 *\/10* *names:* ***Siddharth & Mostafa***

LVL	Criteria
R	
1	
2	
3	
4	"build and wire"[3] □ circuit is correct, routed cleanly and easy to follow[1½] □ all full voltage wire red and all gnd wires black □ signal wire colours chosen to allow easier tracing of circuit[½] tinkerCAD[2] □ all components mounted on breadboard and do not block view of other components[½] □ wires horizontal or vertical only with 90 degree bends[½] □ wires do not cross in front or behind other components or component terminals and do not run on top of one another[½] □ wires and component do not share the same hole on the breadboard and wires do not cross when possible[½] in person[2] □ all full voltage and gnd wires are solid core, flat to breadboard, horizontal or vertical with 90 degree bends □ solid core wires stripped 6-8mm[½] □ no bare wire visible [½] “programming”[3] □ final “test your understanding” complete and working correctly[1½] □ code commenting is accurate and complete (including title)[½] □ program structure and spacing is logical and demonstrates organization[½] □ code text submission is courier new font and is coloured to allow easier identification of comments[½] “inspection questions”[1] □ demonstrates full understanding of circuit and interfacing concepts in conversation with teacher
4+	“enhancements”[1] □ minimized number and length of wires and wire crossings[½] □ circuit enhancement complete and working correctly[½]

Build image:



code:

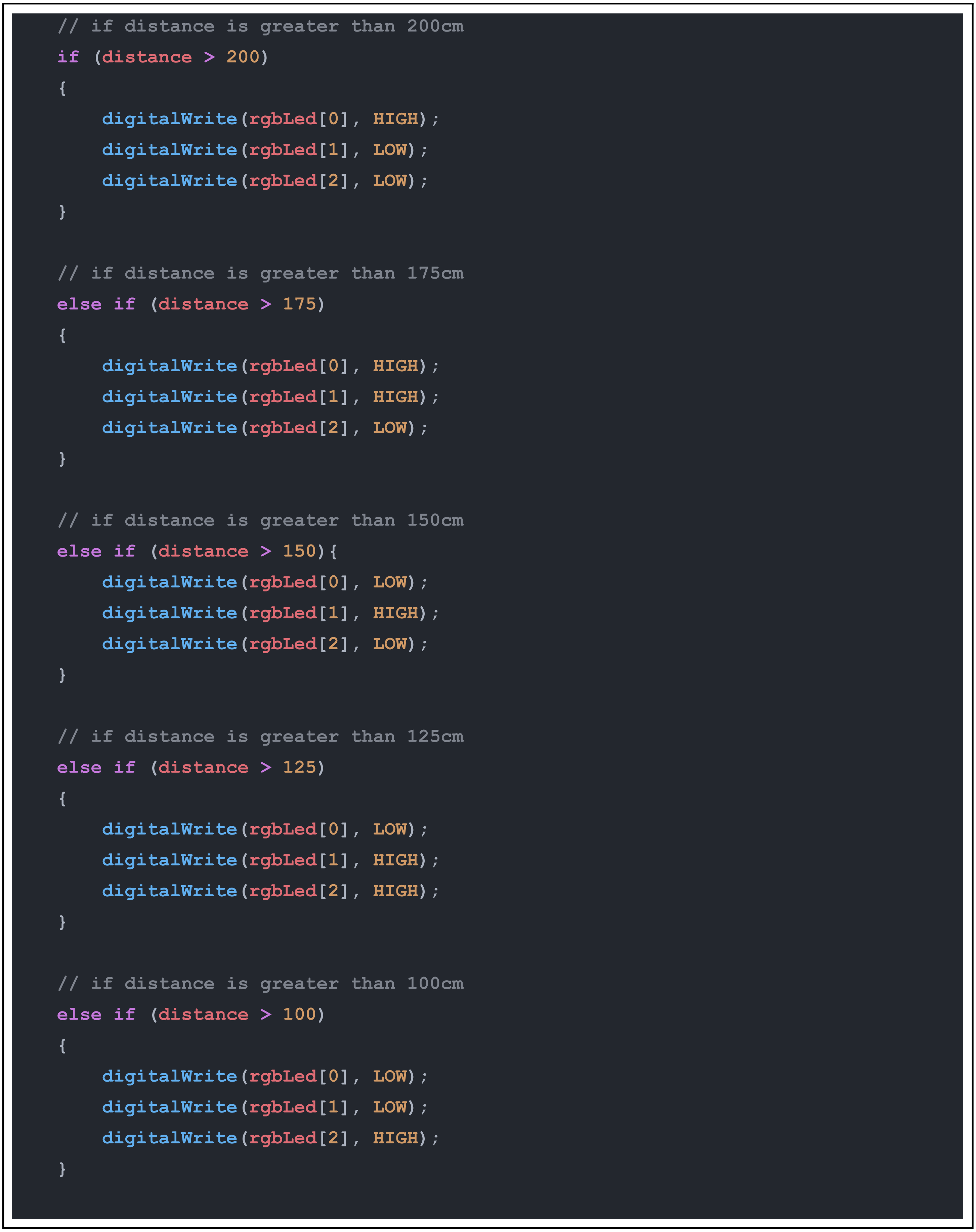
```
/*
Names: Siddarth & Mostafa
Dates: May, 2, 2022
Description: Code for interfacing lab 5 - US Sensor
*/

// declare variables for pins
int trig = 7;
int echo = 8;
int rgbLed[] = {5,4,3};

float distance, duration;

void setup()
{
    for(int i = 3; i < 6; i++)
    {
        pinMode(i,OUTPUT); // Sets RGB LED pins as OUTPUT pins
    }
    pinMode(echo, INPUT); // Sets the echo pin as INPUT
    pinMode(trig, OUTPUT); // Sets the trigger pin as OUTPUT
}

void loop()
{
    /************ Start US Measurement Section ***********/
    digitalWrite(trig, LOW);
    delayMicroseconds(2);
    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig, LOW);
    duration = pulseIn(echo, HIGH);
    distance = duration * 0.034/2;
    /************** End US Measurement Section ***********/
```



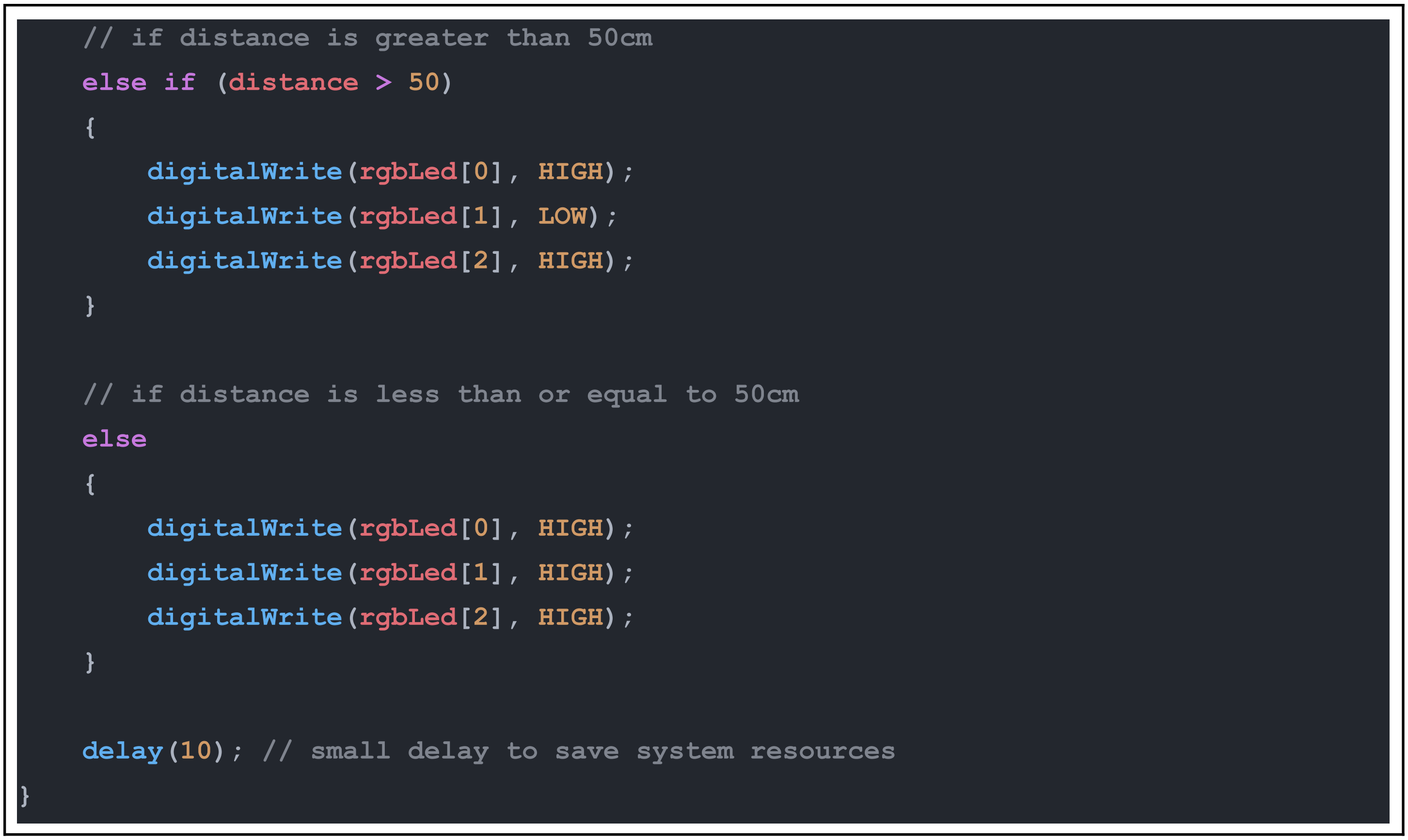
```
    // if distance is greater than 200cm
    if (distance > 200)
    {
        digitalWrite(rgbLed[0], HIGH);
        digitalWrite(rgbLed[1], LOW);
        digitalWrite(rgbLed[2], LOW);
    }

    // if distance is greater than 175cm
    else if (distance > 175)
    {
        digitalWrite(rgbLed[0], HIGH);
        digitalWrite(rgbLed[1], HIGH);
        digitalWrite(rgbLed[2], LOW);
    }

    // if distance is greater than 150cm
    else if (distance > 150){
        digitalWrite(rgbLed[0], LOW);
        digitalWrite(rgbLed[1], HIGH);
        digitalWrite(rgbLed[2], LOW);
    }

    // if distance is greater than 125cm
    else if (distance > 125)
    {
        digitalWrite(rgbLed[0], LOW);
        digitalWrite(rgbLed[1], HIGH);
        digitalWrite(rgbLed[2], HIGH);
    }

    // if distance is greater than 100cm
    else if (distance > 100)
    {
        digitalWrite(rgbLed[0], LOW);
        digitalWrite(rgbLed[1], LOW);
        digitalWrite(rgbLed[2], HIGH);
    }
```



```
    // if distance is greater than 50cm
    else if (distance > 50)
    {
        digitalWrite(rgbLed[0], HIGH);
        digitalWrite(rgbLed[1], LOW);
        digitalWrite(rgbLed[2], HIGH);
    }

    // if distance is less than or equal to 50cm
    else
    {
        digitalWrite(rgbLed[0], HIGH);
        digitalWrite(rgbLed[1], HIGH);
        digitalWrite(rgbLed[2], HIGH);
    }

    delay(10); // small delay to save system resources
}
```