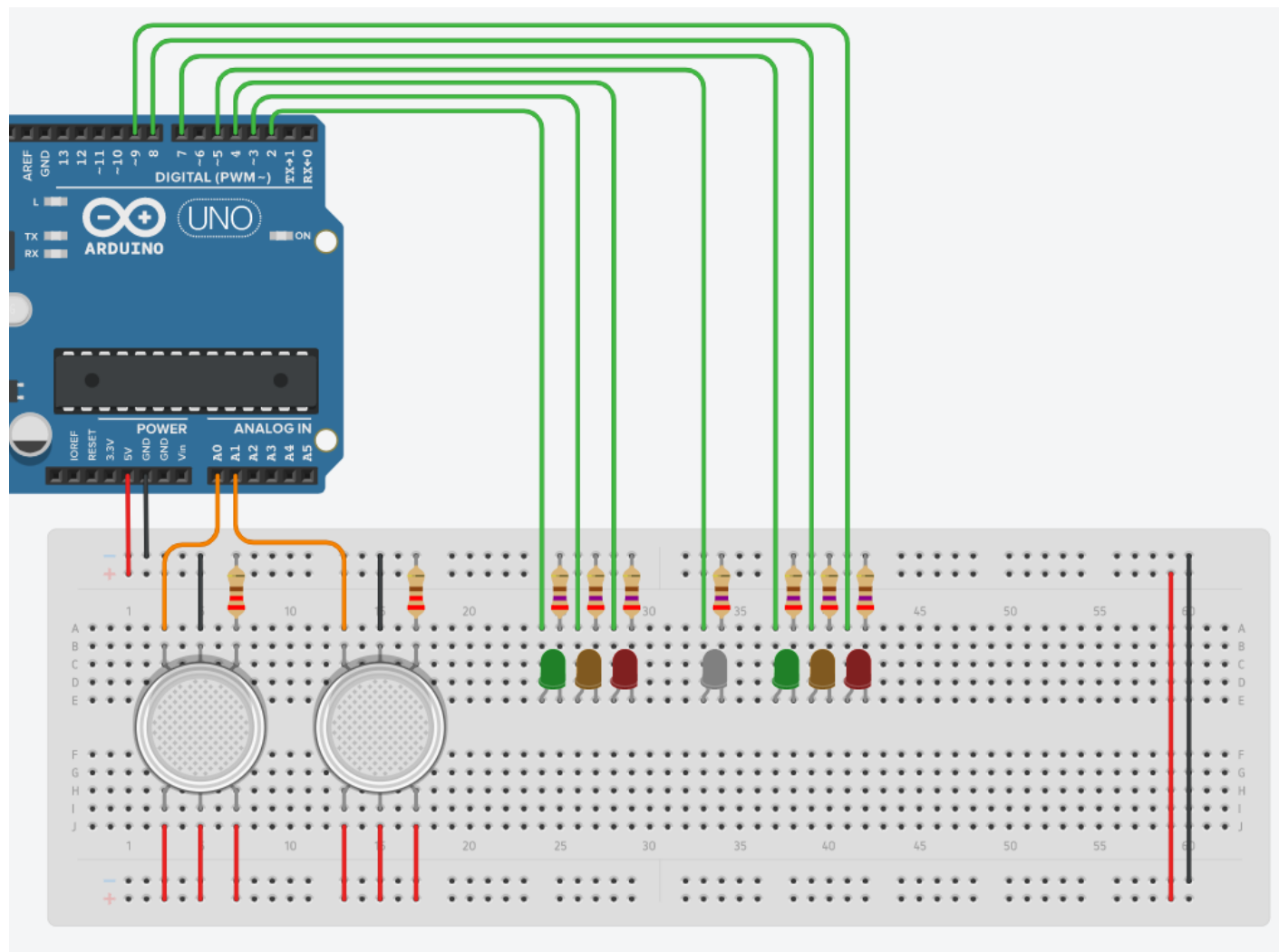


LVL	Criteria
R	
1	
2	
3	
4	<p>"build and wire"[3]</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> circuit is correct, routed cleanly and easy to follow[1½]</li> <li><input type="checkbox"/> all full voltage wire red and all gnd wires black</li> <li><input type="checkbox"/> signal wire colours chosen to allow easier tracing of circuit[½]</li> </ul> <p>tinkerCAD[2]</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> all components mounted on breadboard and do not block view of other components[½]</li> <li><input type="checkbox"/> wires horizontal or vertical only with 90 degree bends[½]</li> <li><input type="checkbox"/> wires do not cross in front or behind other components or component terminals and do not run on top of one another[½]</li> <li><input type="checkbox"/> wires and component do not share the same hole on the breadboard and wires do not cross when possible[½]</li> </ul> <p>in person[2]</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> all full voltage and gnd wires are solid core, flat to breadboard, horizontal or vertical with 90 degree bends</li> <li><input type="checkbox"/> solid core wires stripped 6-8mm[½]</li> <li><input type="checkbox"/> no bare wire visible [½]</li> </ul> <p>"programming"[3]</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> final "test your understanding" complete and working correctly[1½]</li> <li><input type="checkbox"/> code commenting is accurate and complete (including title)[½]</li> <li><input type="checkbox"/> program structure and spacing is logical and demonstrates organization[½]</li> <li><input type="checkbox"/> code text submission is courier new font and is coloured to allow easier identification of comments[½]</li> </ul> <p>"inspection questions"[1]</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> demonstrates full understanding of circuit and interfacing concepts in conversation with teacher</li> </ul>
4+	<p>"enhancements"[1]</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> minimized number and length of wires and wire crossings[½]</li> <li><input type="checkbox"/> circuit enhancement complete and working correctly[½]</li> </ul>

```
Build image:
```



code:

```
/*
Names: Siddarth & Mostafa
Dates: Apr, 19, 2022
Description: Code for interfacing unit lab 3

Uses input from two gas sensors to indicate the amount of gas detected and light 3
different leds each denoting a certain level (red - worse, orange - mid, green -
good). Additionally, when both gas sensors read greater than 50 a white led turns
on.
*/

// initializes int variables for leds of the first sensor
int greenPin1 = 2;
int orangePin1 = 3;
int redPin1 = 4;

// initializes int variables for leds of the second sensor
int greenPin2 = 7;
int orangePin2 = 8;
int redPin2 = 9;

// initializes the variable for the white led
int whiteLed = 5;

// initializes an int multidimensional array (matrix) to store the different led
variables
// grouped in sub arrays based on which sensor they belong to
int ledPins[][3] = {{greenPin1, orangePin1, redPin1},{greenPin2, orangePin2,
redPin2}};

// initializes both sensor pins
int sensorPin1 = 0;
int sensorPin2 = 1;

// initializes the variables used to store the value read from the sensors
int sensorValue1, sensorValue2;

void setup(){ //Setup Code (runs once)
```

```

// Sets the LED pins to output
// For loop to iterate through the array of ledPins
for (int i = 0; i < sizeof(ledPins)/sizeof(int); i++)
{
    // For loop to iterate through the subarray of a set of Led'
    for (int j = 0; j < sizeof(ledPins[i])/sizeof(int); j++)
    {
        // Sets the current selected LED pin to output
        pinMode(ledPins[i][j], OUTPUT);
    }
}

// Sets the white led pin to output
pinMode(whiteLed, OUTPUT);
}

void loop(){ //Loop Code (runs repeatedly)
    sensorValue1 = analogRead(sensorPin1); // Reads value for first sensor
    sensorValue2 = analogRead(sensorPin2); // Reads value for second sensor

    int sensorValues[2] = {sensorValue1, sensorValue2}; // Places values into an
array

    // If both sensors read 50 the white led is on
    if (sensorValue1 > 50 && sensorValue2 > 50)
    {
        digitalWrite(whiteLed, HIGH);
    }
    else // else it is turned off
    {
        digitalWrite(whiteLed, LOW);
    }

    //Iterates through the numbers 0 and 1, switching between first sensor and
second sensor
    for (int j = 0; j < 2; j++)
    {
        // if j = 0 this sets the currentSensorValue to the value of the first
sensor
        // if its 1, its the value of the second sensor
        int currentSensorValue = sensorValues[j];

```

```

// initializes an int array of size 3
int currentLeds[3];
// For loop to set the elements of the array currentLeds to that of
// the sensor currently being checked
for (int i = 0; i < sizeof(ledPins[j])/sizeof(int); i++)
{
    currentLeds[i] = ledPins[j][i];
}

// if the current value for the sensor is greater than 40 light up red pin
if (currentSensorValue > 40)
{
    // Goes through all the current leds (those of the current sensor)
    // and checks if it is the red led pin of the first or second sensor
    // if it is, the red led is set on
    for (int i = 0; i < sizeof(currentLeds)/sizeof(int); i++)
    {
        if (currentLeds[i] == redPin1 || currentLeds[i] == redPin2)
        {
            digitalWrite(currentLeds[i], HIGH);
        }
        else
        {
            digitalWrite(currentLeds[i], LOW);
        }
    }
}

// else if the current value is less than 40 and greater than 25 light
// the orange pin
else if (currentSensorValue > 25){
    // Goes through all the current leds (those of the current sensor)
    // and checks if it is the orange led pin of the first or second sensor
    // if it is, the orange led is set on
    for (int i = 0; i < sizeof(currentLeds)/sizeof(int); i++)
    {
        if (currentLeds[i] == orangePin1 || currentLeds[i] == orangePin2)
        {
            digitalWrite(currentLeds[i], HIGH);
        }
        else
        {

```

