CAB 202 Assignment

Name: Joseph Haddad
Student Number: n10535268
Submission Date: 30/05/2020

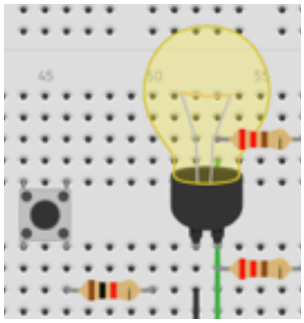
Table of Contents

Introduction

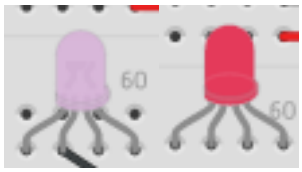
The following application has been designed with the influence of being able to provide a meaningful service, whether it be in the way of entertainment, home automation… Regarding the application that concerns this report, the design provides a fire alarm system while looking to address an indoor environment improvement system which is able to monitor and protect an environment such as your home or office so that both the users and the building can be safe. The application is driven by an Arduino using the c programming language. In which the functionality of the application is constrained to meeting 8 different criteria. As listed below it needs to feature:

1. Digital I/O – switch
2. Digital I/O – Interrupt based debouncing
3. Digital I/O – LED
4. Analog Input
5. Analog Output – PWM
6. Serial I/O
7. LCD
8. Timers

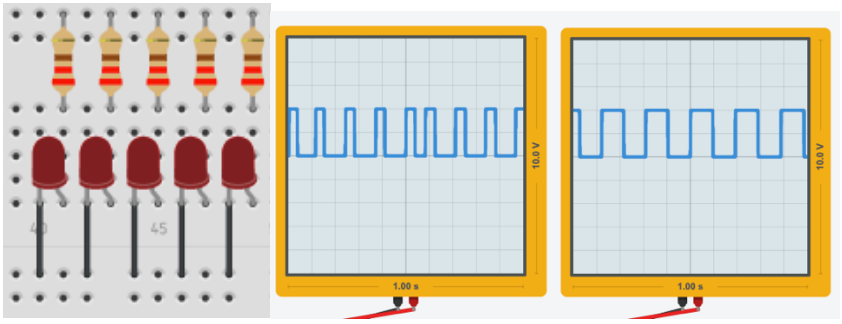
Of these features the application is able to effectively meet 8/8 of the criteria. Starting off with 1 and 2. The program features a button that is able to switch on and off the lights in the area. This button is debounced via an interrupt timer based system to ensure that when the button is pressed it doesn't accidently get pressed more than once. In terms of functionality this one button should be able to allow access to lights within the area it is connected to, or however the user would like it configured. It could light up an entire room or just one light.



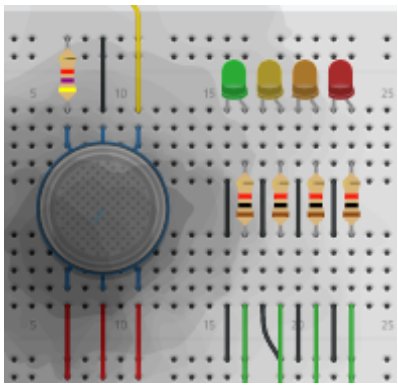
The next functionality the design conforms to are LED lights 3. They are located in several areas across the design to indicate different things. Each Arduino features an RGB led light running on its own timer to indicate if the application is in a running state or not, they both switch from red to blue creating a pink colour. These allow the application to meet criteria 8.



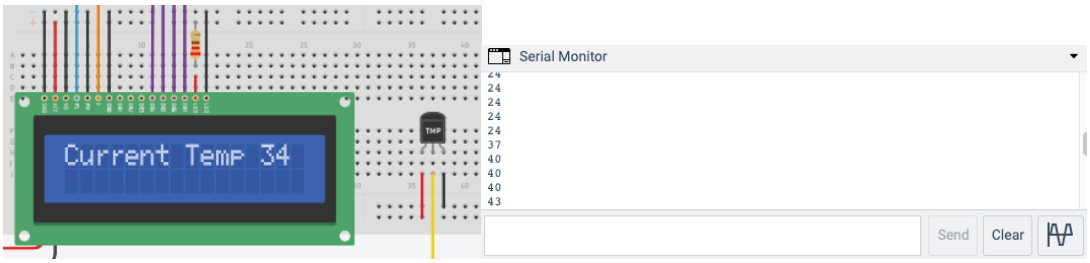
The next set of LED's featured in the design are designed to operate with the siren buzzer. They flash at a changing pulse rate. The PWM changes from approximately 8% - 59%. This alongside the siren in the event of a fire will flash lights at a changing rate that will help inform. Any users inside of the building that there is a fire. As the changing speed of the light should attract their eyes. This functionality also adds to constraint 5. The two oscilloscope screenshots on the right of the LED's pictured below, show the changing pulse width over time of the blinking speed. Initially is starts off small as shown in the left picture then gradually gets bigger like the right picture. This then in time will reset back to the first state and start over.



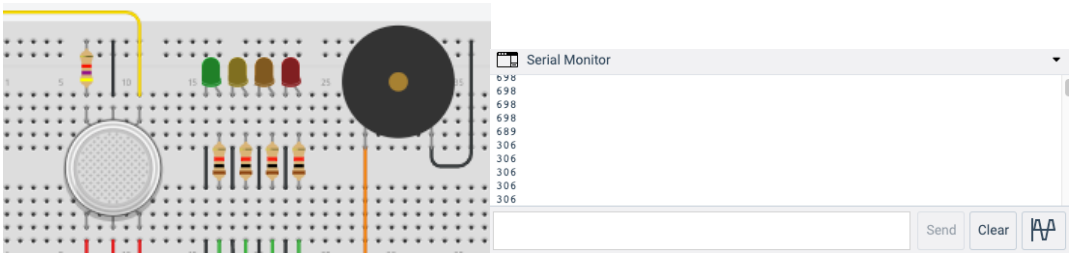
The last location with LED's is used to show how much gas/fire smoke there is in the area. The colours go from green to red. They light up from left to right progressively as the gas in the area increases. These help notify the user before a fire might occur if they should be concerned if their area has a high concentration of smoke.



In order to conform with the LCD requirement 7. A simple text display has been implemented. It receives data that has been read from the analog signal of the temperature sensor and displays it as the current temp in degrees C allowing the application to also meet constraint 4. This information can be used to adjust the ambient temperature in the room by sending the signal to an ac or to allow a user to check it and manually adjust and ac system if needed. Additionally the temperature sensor outputs its analog signal to the serial console. This is then converted to a readable temperature in Celsius and displayed to the LCD screen. Allowing it to also meet the criteria 6.

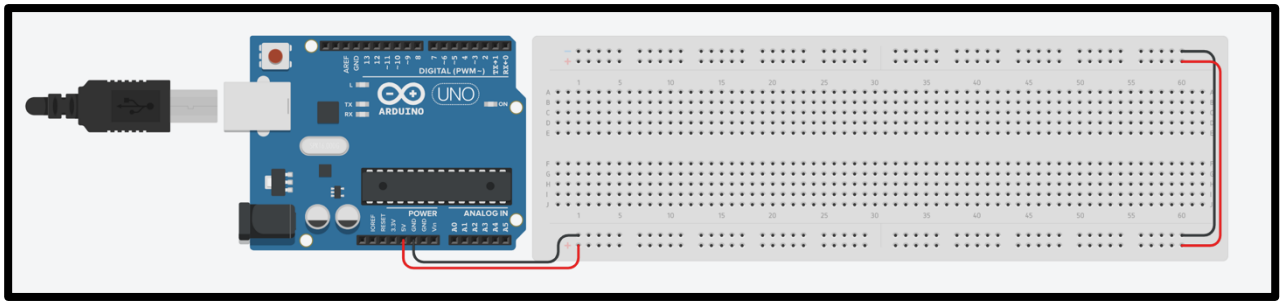


The final components to this application are the gas sensor and the buzzer. The gas sensor reads information on the amount of gas in the surrounding area its located in – (fumes from fire) which once it exceeds a certain threshold will trigger an interrupt that fires of the buzzer and flashes the lights indefinitely until the entire system is powered off and rest. This is to ensure nobody can turn it off accidentally in the event of a fire. These add to the analog read criteria and analog output of PWM as the buzzer needs a constant quick pulse for it to buzz at a desired frequency. The gas sensor allows the design to meet the serial I/O constraint 6. As it outputs to the serial console and reads the data to check the smoke.

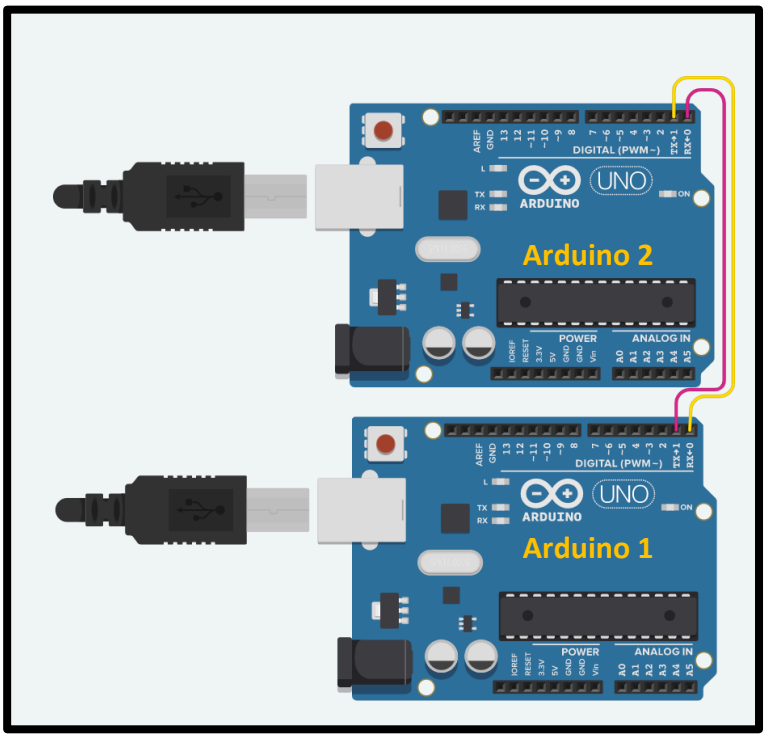


Wiring Instructions

- **Initial Setup**
  - Before starting the complete wiring setup, ensure to have an adequate amount of wires of various lengths. The setup pictures in the schematic diagram blow used around 32 long wires and 40 shorter wires. Additionally, it is recommended that two long breadboards are used. One for each Arduino. Ensure not to keep the breadboard connections separate from each other.
  - Some important wires to initialise are 5V and ground. For each Arduino connect 5V pin to the positive power rail on the breadboard and connect ground to the negative power rail as shown in the diagram below.



- **Connecting Serial Communications**
  - Once the Arduinos are in a stable position. Directly connect the TX pin from Arduino 1 to the RX pin of Arduino 2. Next get another wire and connect the RX pin or Arduino 1 to the TX pin of Arduino 2. This is demonstrated in the picture below.

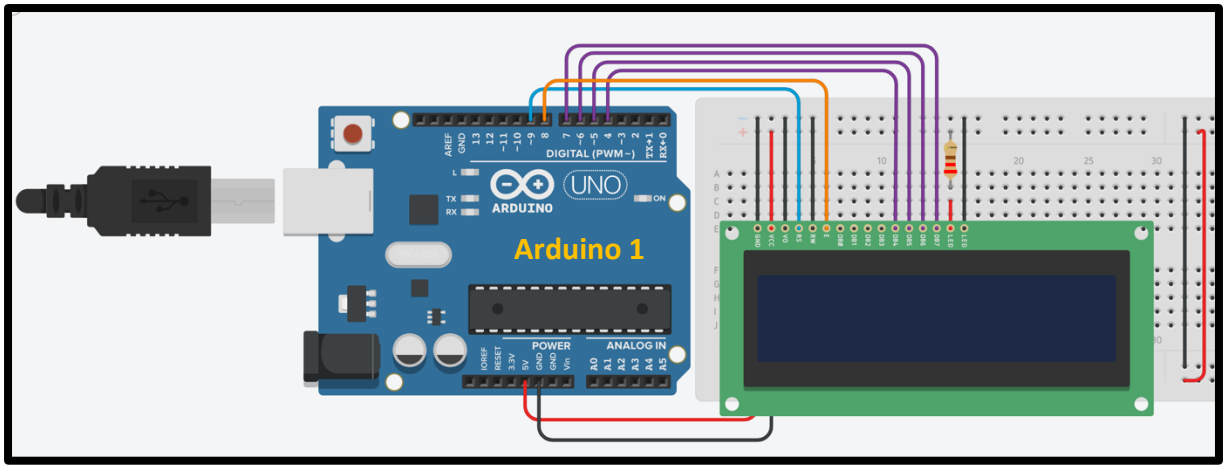


- **Connecting LCD**
    - To connect the LCD to Arduino 1. Place the LCD on the breadboard. Ensure that the breadboard has 5V and ground connections from the Arduino to all the power rails. Now connect a wire from the ground rail of the breadboard to the GND, V0, RW and LED Cathode pins of the LCD. Next connect the VCC pin of the LCD to the positive power rail of the breadboard.

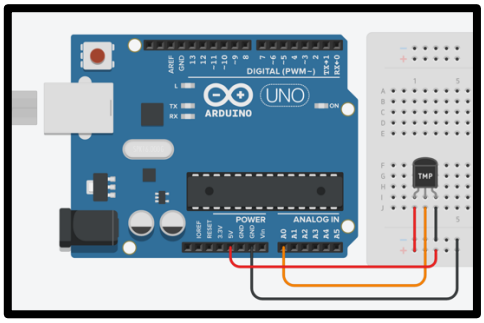
    - After that as shown in blue in the diagram below. From pin 9 on the Arduino connect a wire to the RS pin of the LCD. Then as shown in orange. Starting from pin 8 on the Arduino connect a wire that goes to pin E on the LCD.

    - The LED anode requires a R1 resistor – (220Ω). Place this resistor from the positive rail of the breadboard to the corresponding rail on the breadboard that contains the LED Anode pin of the LCD.

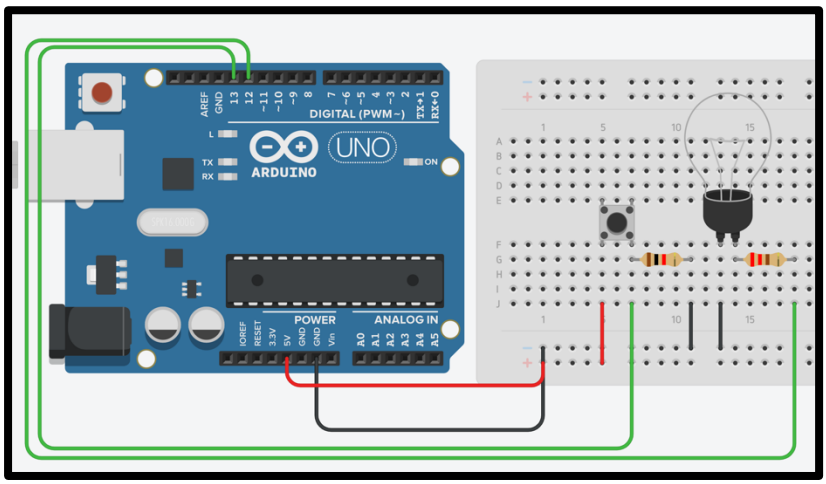
    - Lastly, from pins 7,6,5,4 of the Arduino, connect a wire that goes to pins DB7, DB6, DB5, DB4 of the LCD respectively. These are shown in purple below.



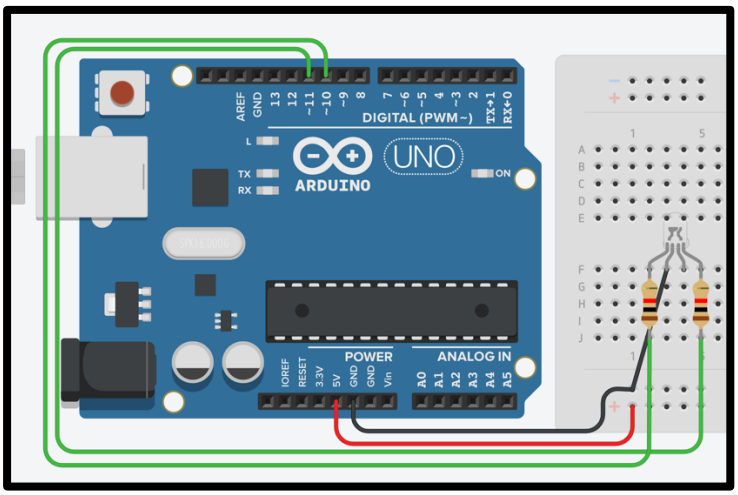
- **Temperature Sensor**
    - To connect the temperature sensor. From the power rail of the breadboard connect a positive wire to the left pin of the sensor and connect a ground pin to the right pin of the sensor. Then from the Arduino pin A0, connect a wire the middle pin of the senor.



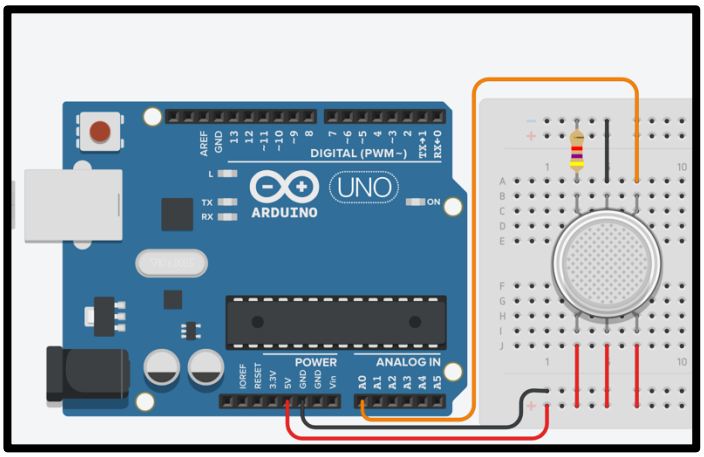
- **Button and Lightbulb**
    - From pin 13 of Arduino 1 connect a wire to the right side of the button. Additionally, horizontally connect a 1k resistor from the right side of the button to ground. Lastly, from the positive rail of the breadboard connect 5V power to the left side of the button.
    - To connect the light bulb, connect the left terminal to the grounding rail of the bread board. Then from the right terminal connect a horizontal resistor, to this horizontal resistor connect a wire that goes from pin 12 of the Arduino to the resistor.



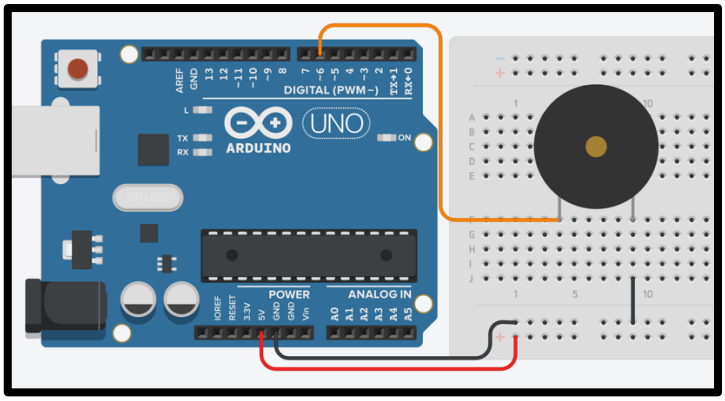
- **Connecting RGB LED**
    - Connect ground to the left middle pin of the LED from the ground rail. For Arduino 1 connect a 220Ω resistor to the leftmost and right most pins of the led. From pin 10 connect a wire to the LED's leftmost resistor and from pin 11 connect a wire to the LED's rightmost resistor. This same process is repeated for Arduino 2 except connect pin 13 to the leftmost resistor and pin 12 to the rightmost resistor.



- **Gas Sensor**
    - When wiring up the gas sensor place the senor in the middle of the bread board so that the top 3 pins are not connected to the bottom 3 pins. From the 5V rail connect each of the bottom rails to 5V. These are shown by the red wires in the diagram below. Then for the top half, connect the left pin to ground using a 4.7k resistor, connect the middle pin straight to ground using a wire and for the right pin. Connect this using a wire to the A0 pin in the Arduino. These connections are also shown in the below diagram.

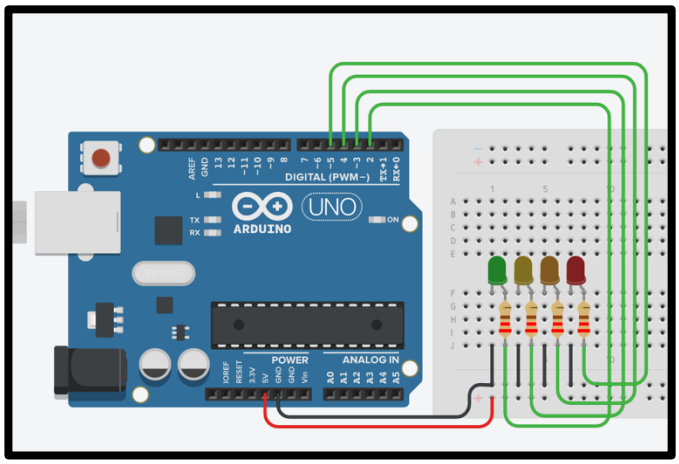


- **PIEZO**
    - To wire up the PIEZO buzzer. Connect the left side of the PIEZO to pin 6 on the Arduino, and connect the right side straight to ground.

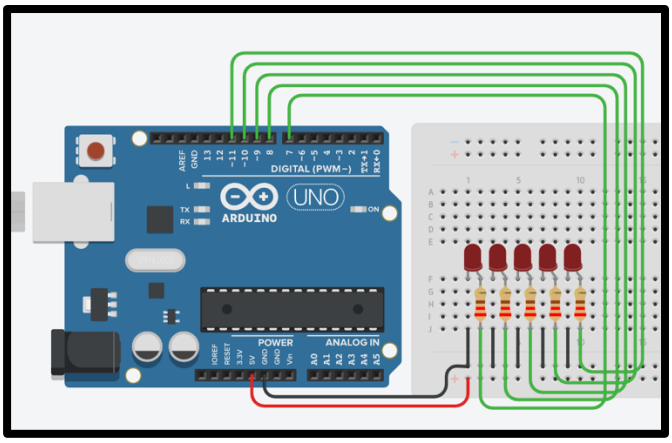


- **Green Yellow Orange Red LED**

  Place each led on the breadboard going left to right putting green then yellow, orange and red. Make sure that each led has its cathode and anode in its own separate individual breadboard hole. This is shown in the picture below. To wire them up connect each LED's cathode to ground and connect a 220 ohm resistor to each of the anodes. Once the resistors are connected from the corresponding anodes connect a wires going from the to pins 2,3,4,5 – (Green, Yellow, Orange and Red) on the Arduino.



- **Flashing LED**
  - Place all 5 red LED's right next to each other having each anode and cathode in their own holes in the breadboard. This is demonstrated in the breadboard below. Connect each LED's cathode to ground and connect a 220ohm resistor to the anode of each LED. Next connect from each resistor a wire's going to pin's – (7,8,9,10,11) on the arduino.

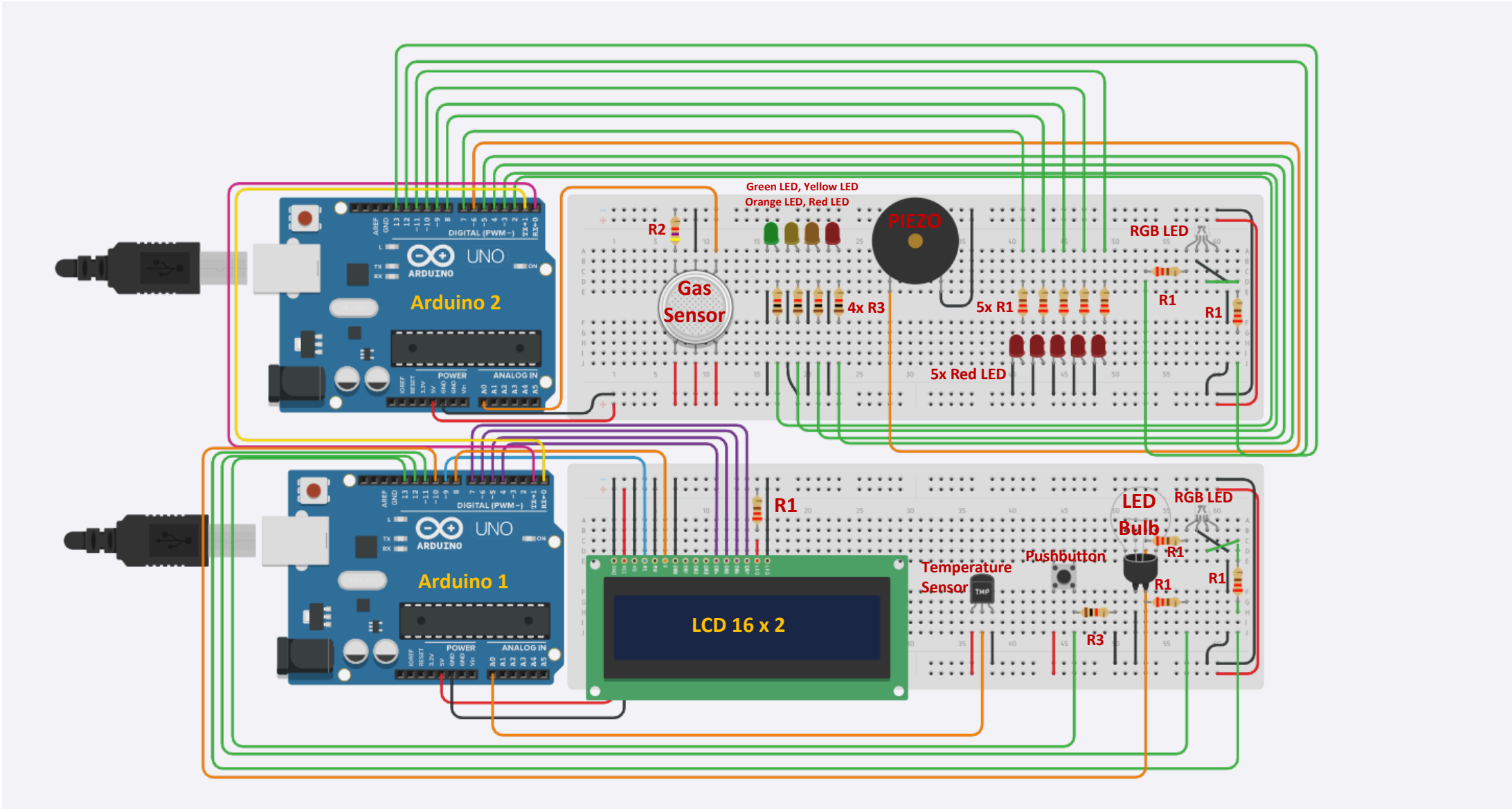


Fully Labelled Schematic Diagram

Bill of Materials

Name	Quantity	Component
Arduino	2	Arduino Uno R3
LCD Display	1	LCD 16 x 2
Resistors – R1	11	220Ω Resistor
Temperature Sensor	1	Temperature Sensor [TMP36]
Resistor – R2	1	4.7kΩ Resistor
Green LED	1	Green LED
Yellow LED	1	Yellow LED
Orange LED	1	Orange LED
Red LED	6	Red LED
Resistor – R3	5	1kΩ Resistor
PIEZO Buzzer	1	Piezo
Light Bulb Button	1	Pushbutton
LED Light Bulb	1	Light Bulb
Operating LED	2	LED RGB
Gas Sensor	1	Gas Sensor

Schematic Diagram



Tinker CAD Circuit Link

https://www.tinkercad.com/things/jATLjF7vYZG-n10535268-joseph-haddad-cab-202/editel?sharecode=BTVoDZeGLMEvXqhcS6sGKtJMXQfzSsk0siQvnuzDh3M

Video Presentation

https://www.youtube.com/watch?v=F5MXXeq45rI&t=12s