Team 2 Documentation

Air Quality Tester

Table of Contents

[Introduction 1](#_Toc462049716)

Materials 2

[Instructions 3-4](#_Toc462049718)

[Conclusion 5](#_Toc462049719)

[References 5](#_Toc462049720)

# Introduction

This project is simply a device that will help you identify the quality in the air. More specifically it will identify if there is a harmful smoke in the air as well as give general information about the air like the temperature and humidity. This device is meant to be implemented as more of an indoor product since it deals with smoke which is a more general gas.

# 

# Materials

-Arduino Uno

-MQ9 & MQ2 gas sensor

-DHT11 sensor

-combination of female and male wires

-16x2 LCD Shield

\*it’s important to note that we are using a Optrex C-51505 as other common LCD shields can have

Different pinouts\*

-1 Led light

-10k potentiometer

-2 220 Ohms resistors

- Pins for LCD shield (to solder for wiring connections)

-5V power source

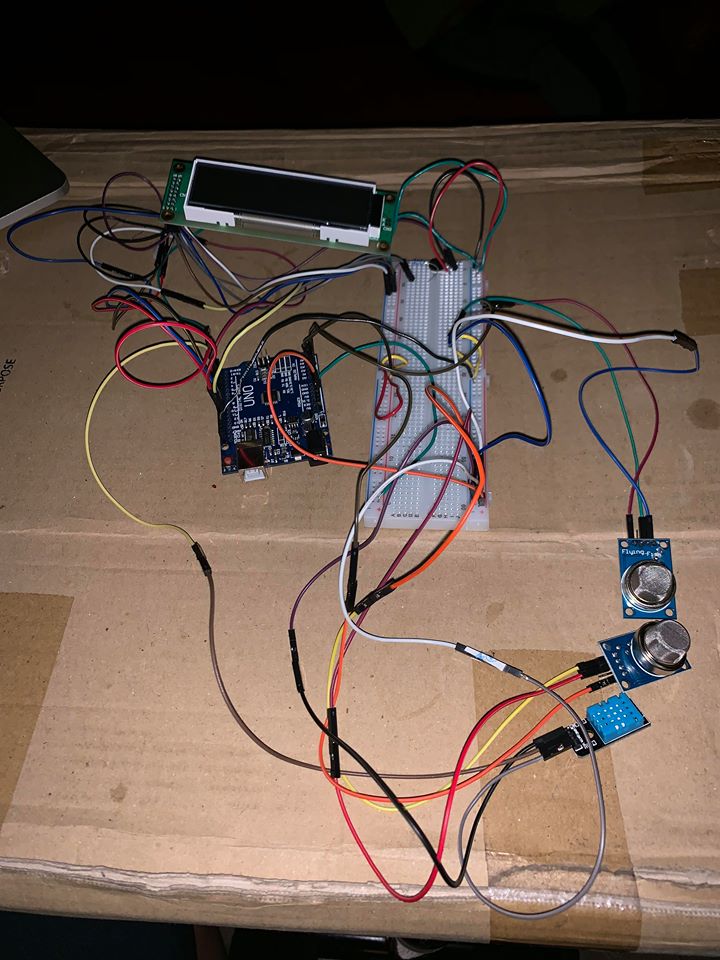
# 

# Instructions

Instructions will be split into software and hardware

Hardware

Here is a picture of the entire design. We have 2 gas sensors that have 4 pin connectors. We have 3 wires connected each of the sensors. One for ground, power, and the last for the data connection which is connected via the analog ports. The DHT11 or humidity and temperature sensor has 3 pins which are for the same 3 connections but instead its data pin is connected to the digital ports of the Arduino. For the LED light, we have each one connected with the negative terminal (shorter leg) connected to the resistor which is then grounded. The other leg is then connected to the digital port terminal for the connection to the Arduino.



The LCD screen is a little more complicated in how its wired. Below is a picture of the connections.

|  |  |
| --- | --- |
| https://www.arduino.cc/en/uploads/Tutorial/LCD_Base_bb_Schem.png | https://scontent-lax3-1.xx.fbcdn.net/v/t1.15752-9/46811005_257150128287100_6335806180190846976_n.png?_nc_cat=101&_nc_ht=scontent-lax3-1.xx&oh=38a5e0e16746386c3e5da72327dc947d&oe=5CA74DEF |
| This is to provide a reference to know which port values such as R/W, RS, etc. are connected to what part of the Arduino.  \*note again the pins of the LCD screen are assigned differently as we are using a different LCD screen. This is simply for connection reference\* | This here is the correct pinout for the LCD screen we will be using. You can follow this pinout with the help of the connection design to connect your LCD screen |

For the connections, we have pin 1 and 2 be the anode and cathode terminals connected to provide the backlight of the LCD. We also use a 220-ohm resistor here to provide the full value of the backlight. Pins 3,4,5 provide us the visibility of the text which is controlled by the potentiometer. Pin 7 is grounded while pins 7 and 8 are connected to the digital ports of the Arduino to apply the first part of the connection that will display. The last four pins which are 13-16 are also connected to the digital ports which are the final connections that will allow us to display our desired test.

Software

We used the SimpleDHT library to read input from the DHT11 sensor, which measures temperature and humidity. With these values, we were able to calculate the heat index, which determines the level of comfort people face depending on the temperature and humidity of their environment. We used the MQ2 library to find the average smoke level from multiple inputs from the MQ2 sensor. Finally, we used the MQ9 library to find the average carbon monoxide level from multiple inputs from the MQ9 sensor. After determining heat index, smoke level, and carbon monoxide level, we determined the different thresholds for each value. We then set the frequency of the LED to represent the severity of each value. The higher the frequency of the LED, the more severe the value was determined to be.

Output: For each value that was calculated, the result is outputted to the LCD shield and the frequency of the LED changes for each value based on its severity.

# Conclusion

If we had more time, we would try to create an app where we could receive/manage these inputs. Also, because getting additional parts for this option would cost more money. Additionally, if we had the resources, we would like to make the measuring portion of the sensor be in real time instead of with the delay we face. This would make the idea even more practical for an actual home use as it is intended.

If you had more time, energy, funding, or resources, what would you change or improve upon? What was your greatest obstacle and proudest moment?

# References

The basic idea of our design/project came from here:

<https://www.instructables.com/id/Air-Pollution-Detector/>

We used the code here for our MQ-2 sensor to get the correct output in the desired units ppm:

<https://www.instructables.com/id/How-to-Detect-Concentration-of-Gas-by-Using-MQ2-Se/>