AIR QUALITY MONITOR PROJECT

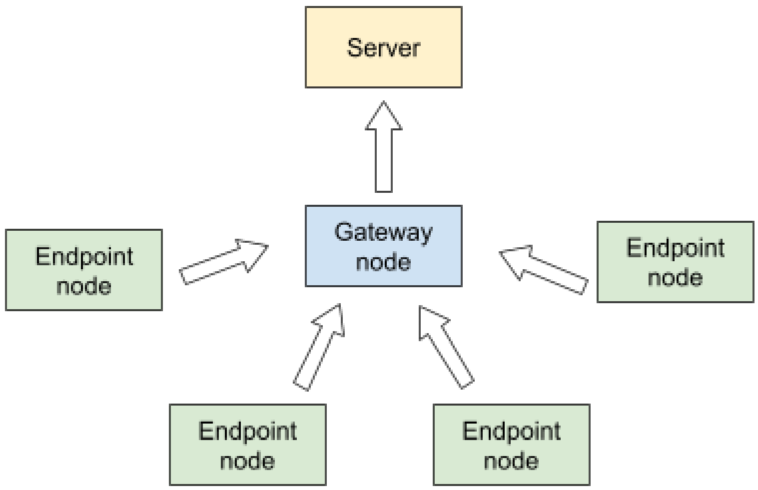
Hardware Guide

Table of Contents

1. Introduction
2. Parts
    a. Master Parts List
    b. Budgeting with Parts
3. Setup
    a. LoRa Transmitters
    b. Network Endpoint
    c. Sensors
        i. MQs
        ii. PM2.5
        iii. DHT22
        iv. NO_2 & SO_2
4. Air Pollutants - Which Ones Matter?

1. Introduction

The air quality monitor network setup consists of a **gateway node** and multiple **endpoint nodes**. The gateway node is responsible for receiving sensor data from the endpoint nodes and sending the data to a designated server. The various sensors are installed only on the endpoint nodes.



The network is established using LoRa (long-range) radio communication. Each node is installed with a LoRa module. There is no communication between the endpoint nodes; each endpoint node communicates directly to the gateway node. In perfect condition, the physical range of LoRa transmission can extend up to 10km, but the effective range depends significantly on the line of sight.

2. Parts

Check your project kit and ensure all of these parts are accounted for. Use this table as a reference for any parts mentioned in the setup instructions for this project. Each monitor will need one of the mentioned parts unless otherwise specified. Prices mentioned and sources for parts are subject to change. Keep in mind that any sensor can be removed and other sensors can be added to the monitors at your discretion.

Note About Resistors

Our parts list recommends buying 1k and 2k resistors, but if you already have different resistors you can still make it work. The resistors are needed to build a voltage divider that divides the voltage going to the LoRa from the Arduino. The formula for a voltage divider is Vout = Vin * (R2 / (R1 + R2)). The Vin in our case is 5V, and the operating voltage of the LoRA module is ~3.3V. So R2 / (R1 + R2) = ⅔ => R1/R2 = ½.

So as long as R1/R2 = ½, we achieve the desired dividing effect. For example, while prototyping this project we had a one hundred resistor with two one hundred resistors, simplifying to the ratio of ½. A 1k with a 2k resistor accomplishes this in two parts.

Master Parts List

Part Name	Cost Per Unit	Description	Source
DC01 Infrared PM2.5 Sensor	$9.99	Detects Particulate Matter Pollution	Link
Ximimark MQ-135	$8.99	Detects carbon	Link

Sensor	(for 3)	dioxide and other gas pollutants	
MQ-7 Sensor	$6.70	Detects Carbon Monoxide	Link
Huaban MQ-3 Sensor	$7.99	Detects alcohol gas	Link
REYAX RYLR998 LoRa Transmitter	$12.60	Transmits data between sensors	Link
WWZMDiB DHT11 Temp. & Humidity Sensor	$5.99 (for 3)	Measures temperature and humidity	Link
ARDUINO UNO R4 Wifi	$27.99	The computer behind each sensor, the Hub Monitor connects directly to the database through Wifi	Link
Wires	$7 for 120	Used to connect Arduino components	Link
1k Resistor	$4 for 40	Used to lower voltage for LoRa transmitter	Link
2k Resistor	$4 for 40	Used to lower voltage for LoRa transmitter	Link
UART USB to Serial Converter	$7.39	Used to set up each LoRa transmitter	Link
Arduino Breadboard	$7.59 for 3	Breadboard for laying out circuits	Link

Budgeting with Parts

This project requires a one-time order of some parts and a per-monitor order of other parts. Our parts list is optimized for a 3-monitor setup with 1 gateway node and 2 endpoint nodes collecting air quality data. You are always welcome to source your parts from another source or in different quantities. This project is also very modular; If any sensor is too expensive for your budget or is not important for what specific pollutants you are trying to monitor you can always not include them, the wiring and code should be easy to add/remove/modify.

One-Time Purchase Items (within reason, may need multiple orders if many monitors are needed):

Part Name	Cost Per Unit	Description	Source
Male to Female Wires	$7 for 120	Used to connect Arduino components	Link
1k Resistor	$4 for 40	Used to lower voltage for LoRa transmitter	Link
2k Resistor	$4 for 40	Used to lower voltage for LoRa transmitter	Link
UART USB to Serial Converter	$7.39	Used to set up each LoRa transmitter	Link
Arduino Breadboard	$8 for 6	Breadboard for laying out circuits, one per monitor	Link
TOTAL	$30.39		

Must Purchase for each Gateway Monitor without Sensors

Part Name	Cost Per Unit	Description	Source

REYAX RYLR998 LoRa Transmitter	$12.60	Transmits data between sensors	Link
ARDUINO UNO R4 Wifi	$27.99	The computer behind each sensor, the Hub Monitor connects directly to the database through Wifi	Link
TOTAL	$40.59		

Must Purchase for each Endpoint Monitor (With Sensors) (3-packs of sensors can be used for 3 different monitors)

Part Name	Cost Per Unit	Description	Source
DC01 Infrared PM2.5 Sensor	$9.99	Detects Particulate Matter Pollution	Link
Ximimark MQ-135 Sensor	$8.99 (for 3)	Detects carbon dioxide and other gas pollutants	Link
MQ-7 Sensor	$6.70	Detects Carbon Monoxide	Link
Huaban MQ-3 Sensor	$7.99	Detects alcohol gas	Link
REYAX RYLR998 LoRa Transmitter	$12.60	Transmits data between sensors	Link
WWZMDiB DHT11 Temp. & Humidity Sensor	$5.99 (for 3)	Measures temperature and humidity	Link
ARDUINO UNO R4 Wifi	$27.99	The computer behind each sensor, the Hub Monitor connects directly to the database through Wifi	Link

Muzuzzi Solar Panel	$15.99	Supplies Power to the endpoint node	Link
Dr. Prepare Battery Pack	16.99	Stores power for endpoint node	Link
TOTAL	$103.19 / each monitor (dividing price of 3-packs by 3)		

3. Setup

This section is a tutorial for setting up the various parts of the air quality monitor

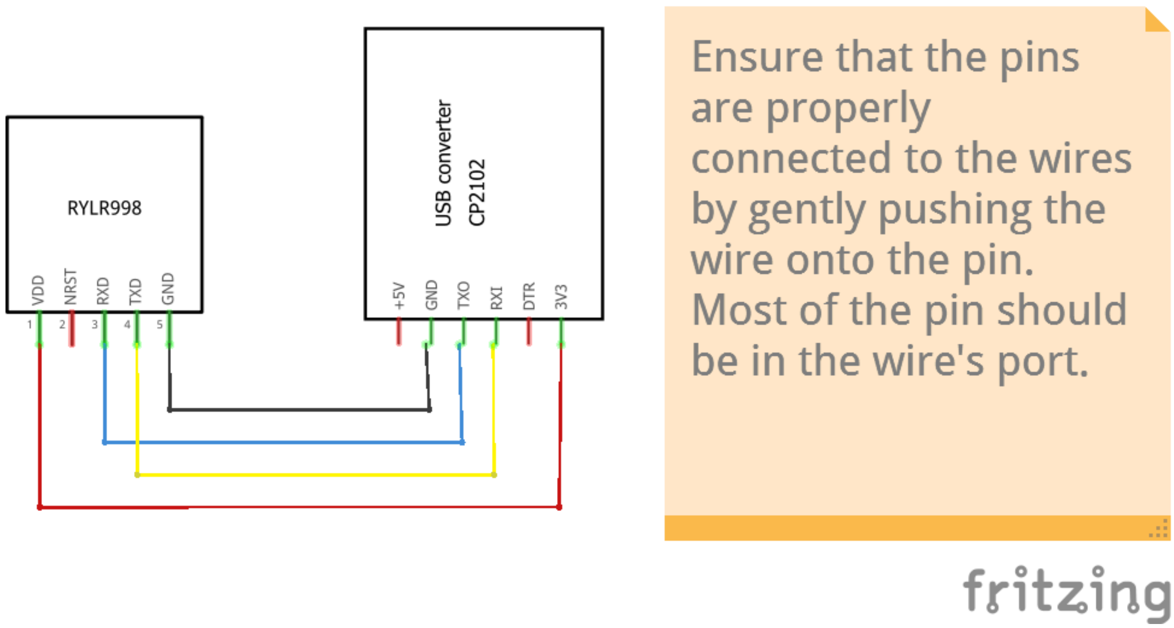
Lora Transmitters

In this section, we will describe how to set up the LoRa transmitters so that your air quality monitors will be able to communicate with each other over radio transmissions.

In order to set up the Reyax RYLR998 transmitters, you will need:

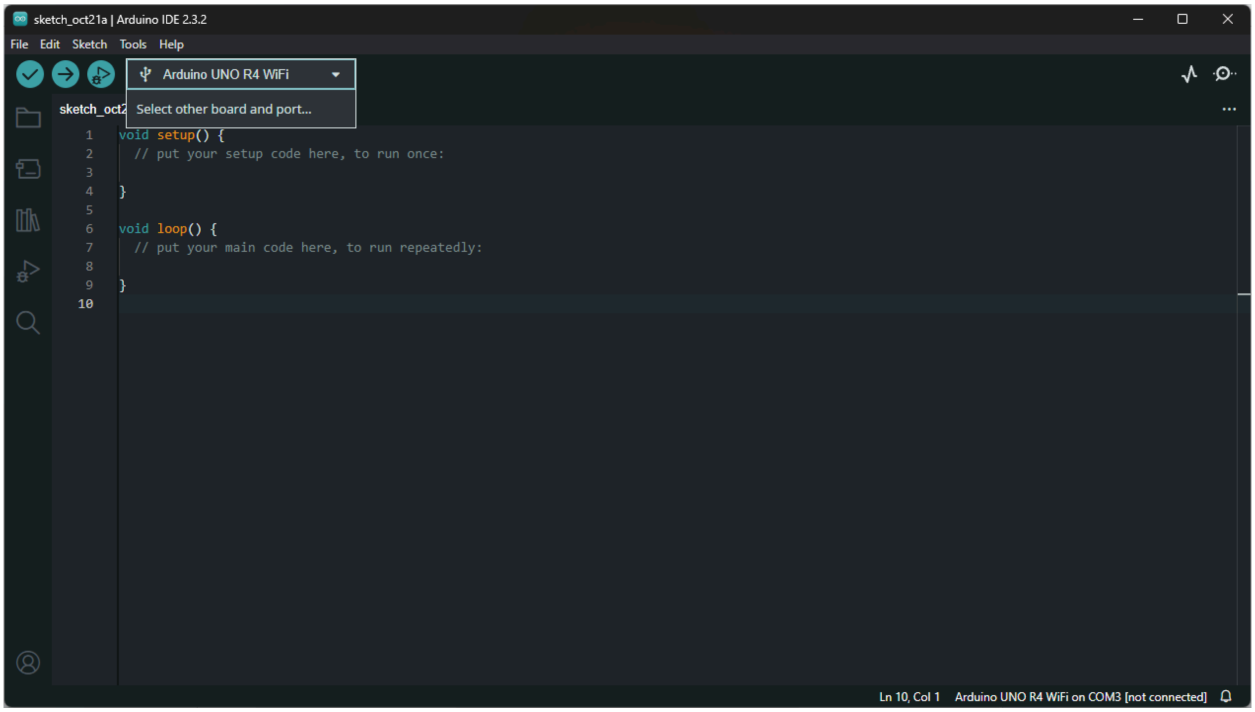
- 1 UART USB to Serial Converter
- 1 Reyax RYLR998 LoRa Transmitters
- 1 Computer with the Arduino IDE application installed (see their website for download options)
- 1 1k Resistor
- 1 2k Resistor

(Note the size of resistor). Once you have these parts, the first step will be to connect the UART bridge to the Reyax LoRa. Here is a diagram showing which pins on each component must be connected to each other.

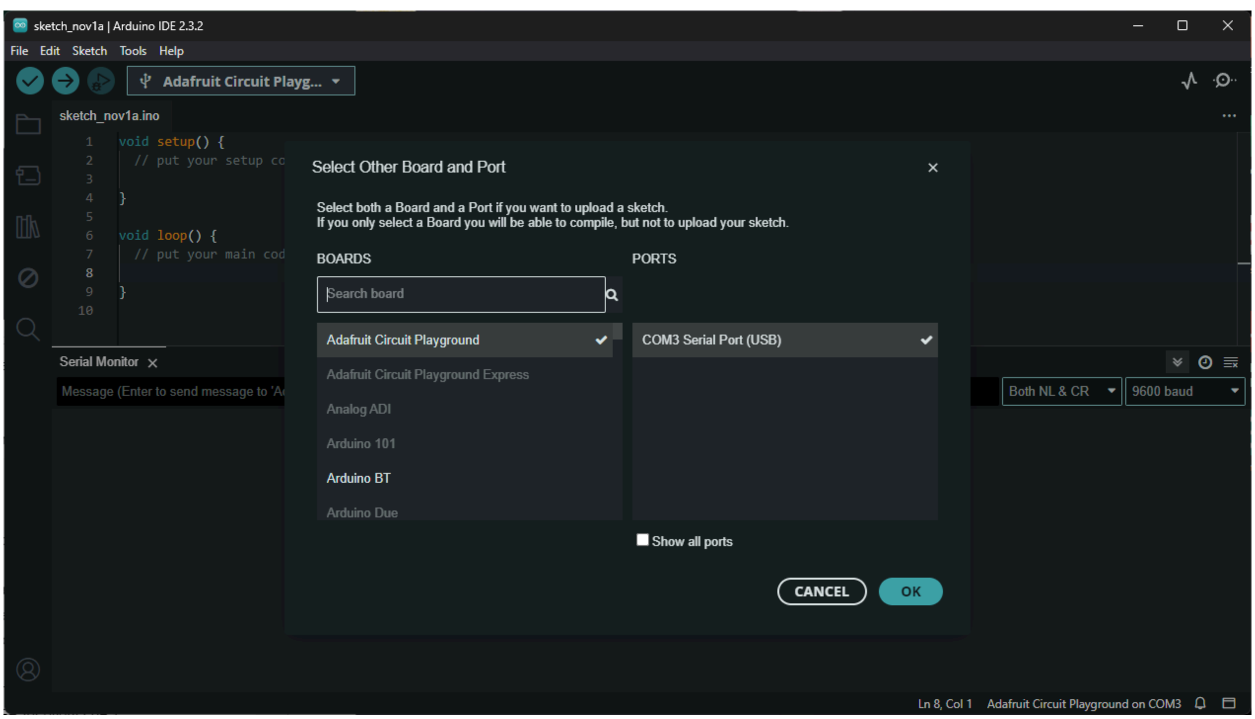


When you are sure that the pins are properly connected to each, you must now download the appropriate driver for the UART from the Silicon Labs website. After installing this driver, you can plug the UART's USB into your computer, and your computer should recognize it as a UART USB to Serial Bridge (in Windows' Device Manager or Mac's System Profiler). If this is not the case, then verify that you properly installed the correct driver or try to reinstall it.

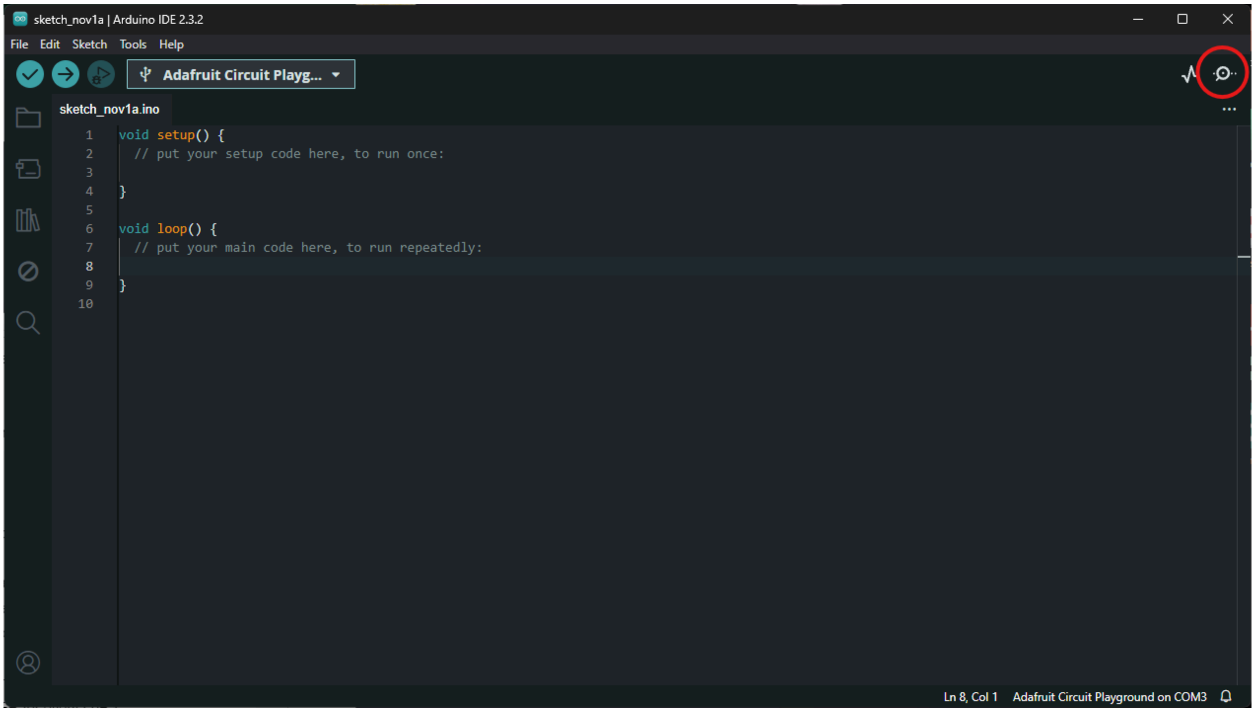
Now, open the Arduino IDE application on your computer, and once it opens click the menu labeled "Select other board and port…"



Here you should see a selection of boards and ports. With the UART and LoRa plugged in, you should see one COM Serial Port in the port section. Choose this COM port, and select "Adafruit Circuit Playground."



Now that the LoRa is selected, we must open the serial monitor. Simply do this by selecting the Serial Monitor option on the top right as shown here:



With the LoRa and UART selected and the serial monitor open, we must program certain parameters into the LoRa so it can function properly once installed on the AQ monitor. We will do this through commands called AT commands. There are four parameters to program: network id, network address, radio frequency, and baud rate. These define what network the transmitter can send messages on, at what frequency, and what speed.

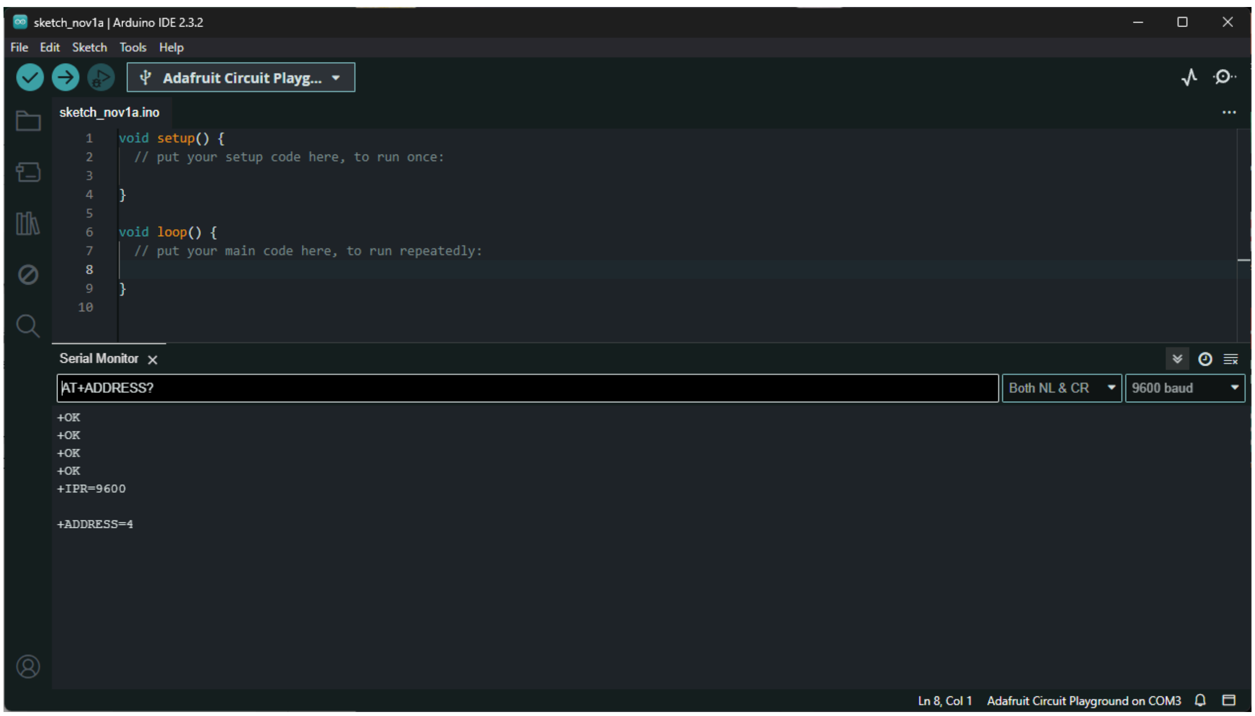
To set the network id, type “AT+NETWORKID=” into the serial monitor and pick a number between 1 and 16. Note that this will be the SAME NUMBER for every LoRa transmitter, as they can only communicate with transmitters in the same network.

Next, use “AT+ADDRESS=” and a number between 0 and 65,535. This is the address for the specific transmitter on the network, and is how they differentiate each other on the network. It is our recommendation to make the gateway’s address 1 and the monitors’ addresses start at 2 and onward. This is for the sake of easy identification.

The radio frequency is called the Band in the AT commands, and this is the frequency (in Hertz) that the transmitters communicate on. Since radio is a regulated form of communication in the United States there is a frequency designated for LoRa, 915MHz. In order to set the Band, type “AT+BAND=915000000”.

Finally, we must set the UART Baud rate, or IPR, which defines the speed that transmissions are sent. If the transmitters are running on two different bauds they will not be able to understand what was sent to them. The accepted standard for the RYLR998 is 9600, so it is set through the command “AT+IPR=9600”.

If you want to check any of these to make sure that the command actually went through, you simply type the command ending with a ?, for example “AT+ADDRESS?” will print out



what the address of the LoRa is.

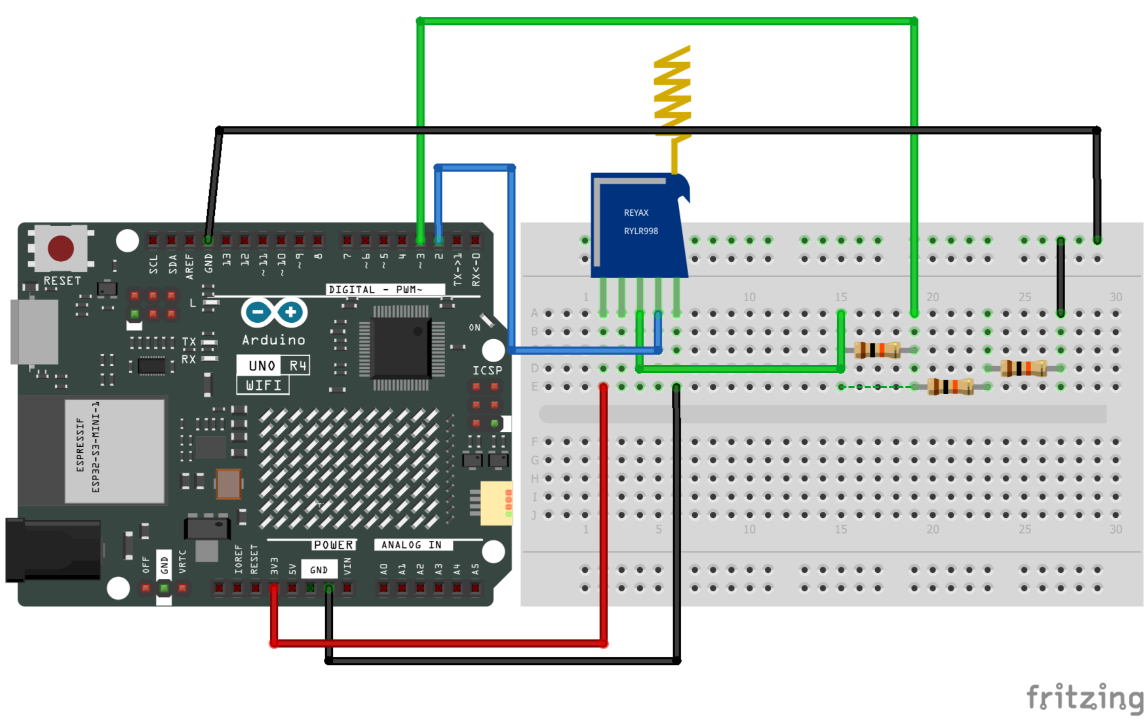
Gateway Node

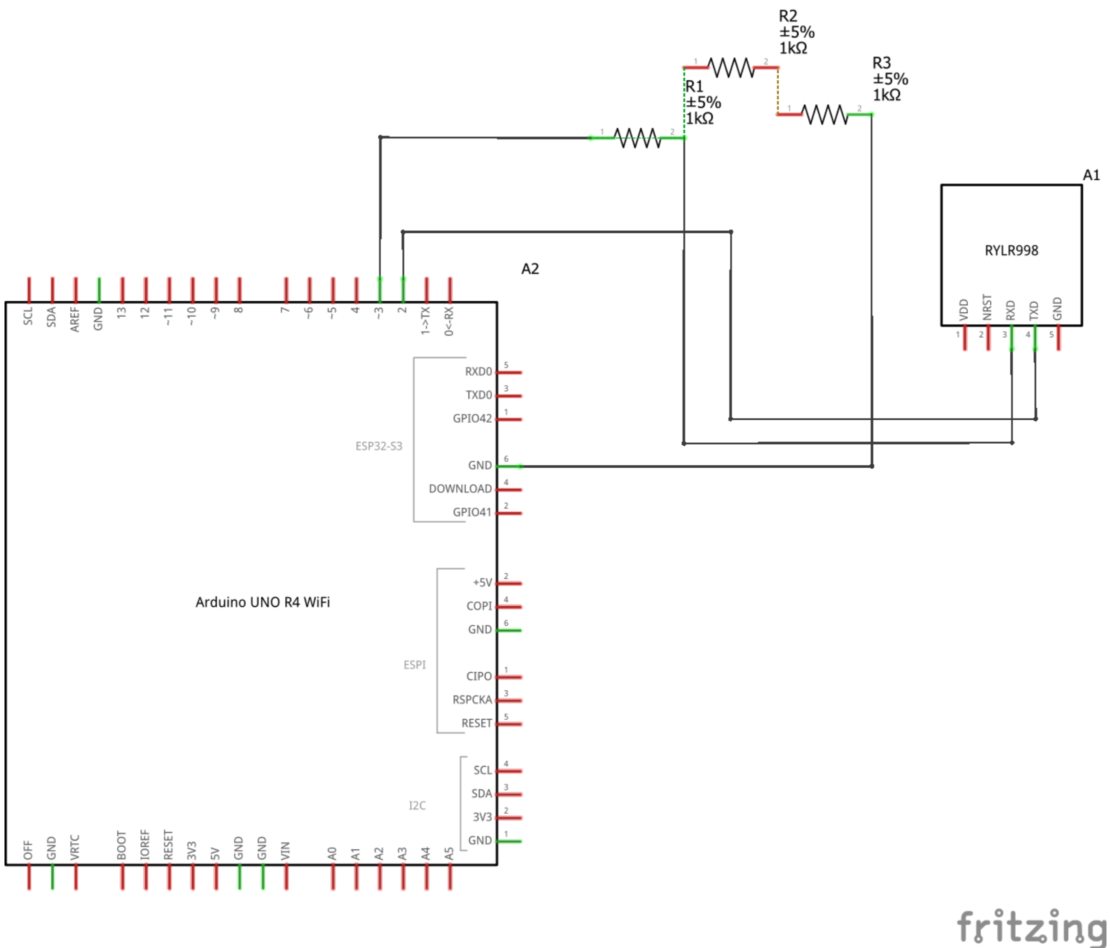
In this section, we will show you how to build an endpoint that is capable of communicating with other endpoints via the LoRa transmitters we set up in the previous section. This version of the endpoint is also the final version of the Gateway endpoint that you will need at least one of to have a functioning network of air quality monitors.

This endpoint will have 4 parts: the Arduino motherboard, a breadboard, a voltage divider, and the LoRa transmitter. The voltage divider is included because of the power output difference between the Arduino and LoRa. The Arduino runs on 5V, meaning every pin on the board runs on a logic of 5V, where data (or highs) on the pin run at 5 Volts. By contrast, the LoRa transmitters run on a logic of 3.3V, and if you were to just plug them into each other without the voltage divider you would fry the LoRa transmitter. Thus we use a voltage divider to bring the logic down from 5V to 3.3V which makes it possible to connect the two components together.

If this diagram is not helping you and you'd like to watch a video, please click this link or visit https://youtu.be/SLLJUxW4hDA.

To ensure that you have correctly built the gateway as shown in the above video, compare your work to the diagrams below. These will show which pins of the arduino are connected to the breadboard and transmitter.





You can test that this endpoint is functioning through a very simple sketch on the Arduino IDE. Create a new sketch and copy this information into the file. Then save it under whatever name you'd like.

```
/* Network Test Code by Spencer Hagan Nov 11, 2024*/
#include <SoftwareSerial.h>

// String for the data we receive from LoRa transmissions
String incomingString;

// LoRa
SoftwareSerial lora(2,3); // RX, TX pin numbers on arduino board.

void setup() {
  Serial.begin(9600);  // Start serial communication
```

```
 Serial1.begin(9600);
 lora.begin(9600);   // Start LoRa communication on 9600 Baud
 lora.setTimeout(500); // Max time LoRa will allow for a transmission

 delay(1000);  // Small delay to allow the serial monitor to initialize
 Serial.println("Waiting for transmissions...");
}

void loop() {
 Serial.println(".");
 delay(1000);
 Serial.println(".");
 delay(1000);
 Serial.println(".");

 //checks if anything has been received yet
   if(lora.available()) {
    Serial.println("DATA RECEIVED!!");
    //read in the data
    incomingString = lora.readString();
    Serial.println(incomingString);
   }
 delay(1000);
}
```

After this, plug the endpoint into your computer using a usb or usb-c to usb-c cable and click the button on the top right of the IDE shaped like an arrow and named "UPLOAD." Once this finishes uploading the sketch to the endpoint, plug in a spare LoRa transmitter with the UART bridge (see LoRa Transmitters section) and open both on separate serial monitors. If you send an AT+SEND command from the LoRa and UART, you should see the endpoint print out the received data.

Sensors

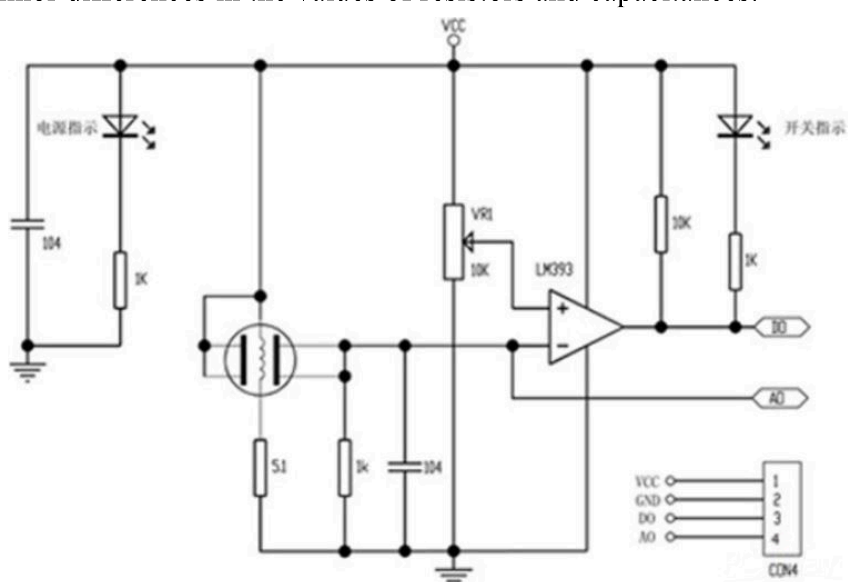
Many sensors are involved in assembling the air quality monitor and each sensor is set up differently. In this section, we describe the process of setting up each sensor at the software and hardware level. The **datasheet** for each sensor is available as a collection on our github repository.

MQ sensors

Three different MQ sensors ie. MQ3, MQ7, and MQ 135 are used in the air quality monitor. Two major components of the MQ sensors are: a heating coil that keeps the sensor at optimal temperature and a chemi-resistive element (Tin Dioxide - SnO_2) that functions as the sensing element which changes its resistance upon the onset of different concentrations of the target gasses. The sensor outputs an analog voltage signal that varies with respect to the variation in the sensing resistance ie. Higher concentrations of a target gas reduce the sensing element's resistance, resulting in a higher analog voltage signal. All of the MQ sensors in our list also come with a small PCB that enables a digital output with an adjustable threshold. Because of the heating coil, the MQ sensors consume a lot more power when compared to digital sensors. The operating voltage of the MQ sensors are 5V.

Hardware details

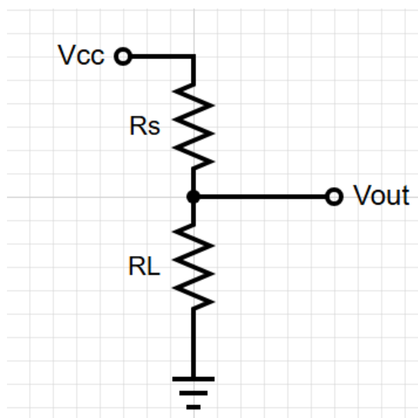
The following schematic applies to MQ3, MQ7, and MQ135 listed above for purchase, with possible minor differences in the values of resistors and capacitances:



This diagram includes two indicator LEDs, where the LED on the left is the power indicator light and the other LED is an indicator light for the digital output signal, which turns on when the LM393 op-amp outputs LOW, ie. the inverting signal is higher than the non-inverting signal. A potentiometer is set up as an adjustable voltage divider that serves as a variable reference voltage for the non-inverting signal of the op-amp. The inverting signal of the opamp is the analog voltage signal that comes from the heating-sensing element.

Since our goal is to measure the concentration of the gasses and not whether a certain gas concentration crosses a predefined threshold, the digital output signal is of no use to us. The question now is, given an analog voltage signal, how to convert that signal to a meaningful concentration value? To do this, we must make use of the datasheet of each of the sensors and some simple circuit theory.

We can visualize the sensing element part of the circuit as a voltage divider:



where Rs is the chemiresistor/sensing element, RL is the loading resistance, Vcc is the input voltage, and Vout is the output analog signal. Note

$$V_{out} = V_{cc} \frac{R_L}{R_L + R_S} \tag{1}$$

then given R_L, V_{out}, and V_{cc}, we can find R_S. Doing some algebra, we can get:

$$R_S = R_L\left(\frac{V_{cc}}{V_{out}} - 1\right) \tag{2}$$

Now you may wonder, how does finding R_S help us find the concentration? We must find the clue by looking at the datasheet. In the datasheet for the MQ sensor, we are given this graph:

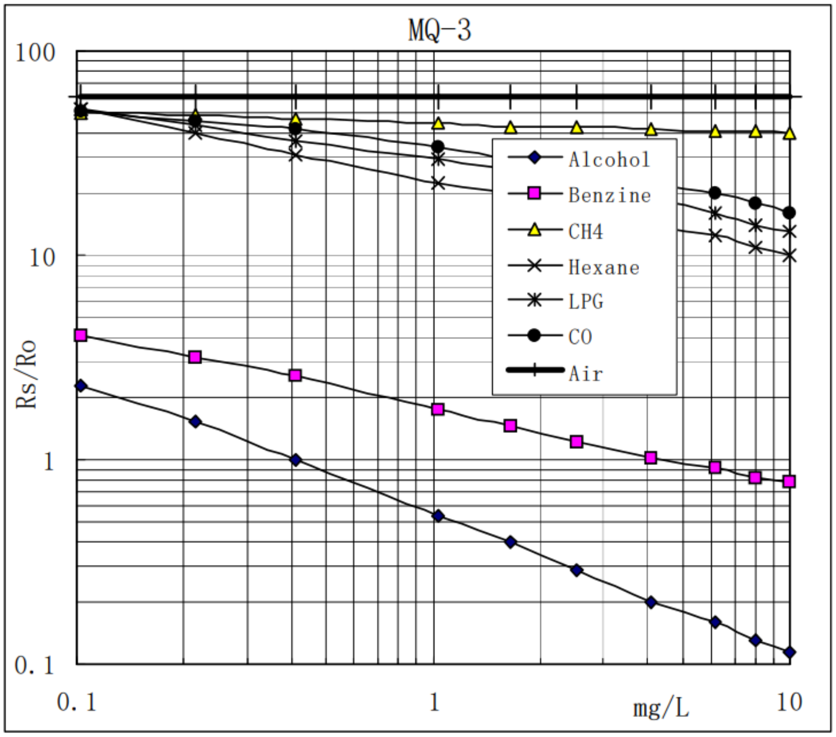


Fig.3 is shows the typical sensitivity characteristics of the MQ-3 for several gases.
in their: Temp: 20℃、
Humidity: 65%、
O_2 concentration 21%
RL=200k Ω
Ro: sensor resistance at 0.4mg/L of Alcohol in the clean air.
Rs:sensor resistance at various concentrations of gases.

We must first notice that this is a log-log plot (Google "log-log" plot if you don't know how to read a log-log plot). The x-axis gives the concentration in mg/L; the y-axis gives the value of Rs/Ro, where Rs is the same Rs we found above, and Ro is a constant value of the sensor resistance at 0.4 mg/L of alcohol in the clean air, i.e., Rs/Ro = 1 at 0.4 mg/L.

Notice that we are given Rs/Ro in clean air, represented by the horizontal black solid line at the top of the graph. In this case, Rs/Ro = 60, for MQ-3. By putting the monitor in fresh air, we can use equation (2) to find the value of Rs. Then Ro can easily be calculated:

$Ro = Rs/60$, where Rs is the sensing resistance in fresh air.

Ro is a constant value that is only accurately measured after the sensor is preheated for at least 24 hours. For every Rs measurement we make, we can compute a value of Rs/Ro. Now notice we can find equations for the sensitivity curves. These curves are approximately linear, and since they are on a log-log scale, we can write:

$log\,(y) = m\,log\,(x) + b$, where $y = Rs/Ro$; $x = \rho$, the concentration.

(The logarithms are base-10) We can find the weights *m* and *b* by using two reference points on the sensitivity curve. Here, I'll use the points (0.4, 1) and (4, 0.2). Then

$$m = \frac{log(y1) - log(y2)}{log(x1) - log(x2)} \tag{3}$$

$$= \frac{log(1) - log(0.2)}{log(0.4) - log(4)} = -0.7.$$

Then

$$b = log(y) - m\,log(x) = log(1) - (-0.7)\,log\,(0.4) = -0.28.$$

We can now solve for x:

$$log\,(y) = m\,log\,(x) + b$$

$$y = 10^{m\,log(x) + b} = x^{m} \cdot 10^{b}$$

$$\Rightarrow x^{m} = y \cdot 10^{-b} \Rightarrow x = (y \cdot 10^{-b})^{1/m} = (y \cdot 10^{0.28})^{-1/0.7}.$$

Alternatively, we can do

$$log\,(y) = m\,log\,(x) + b$$

$$log\,(x) = (\,log\,(y) - b)/m$$

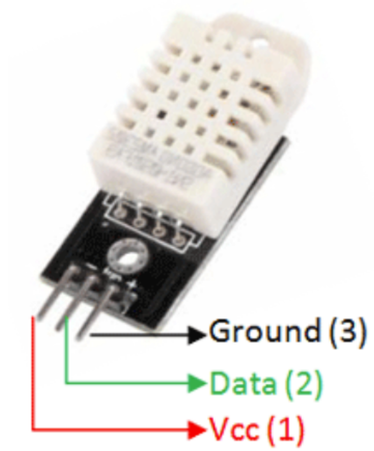
$$x = 10^{(log\,y - b)/m} = 10^{-(log\,y + 0.28)/0.7}.$$

These two expressions are equivalent. The first expression is used in the Arduino sketch. So now, given every measurement (analog signal), we can find Rs/Ro, and then use one of the above two equations to find x, the concentration of the target gas.

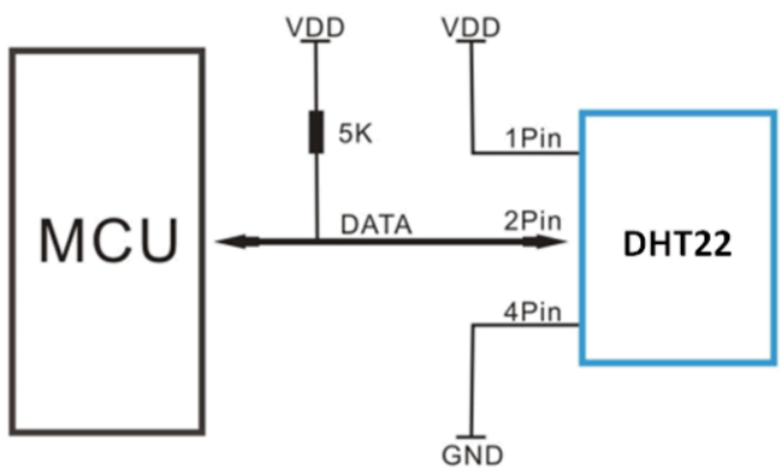
For accurate reading, the above method must be conducted after the sensor's preheated for at least 24 hours.

DHT22

The DHT22 sensor is very easy to set up. It consists of 3 pins (GND, VCC, DATA).



The operating voltage should be 3.3-6 V DC. DHT22 outputs calibrated digital signals that make data-reading a lot easier the the MQs. Single-bus data is used for communication between the DHT22 and a microcontroller (MCU), the arduino, in this case. The details of the data format used in the communication can be found in the DHT22 datasheet on github. The pull-up resistor in the diagram below is part of the DHT22 circuit so you don't need to worry about it.



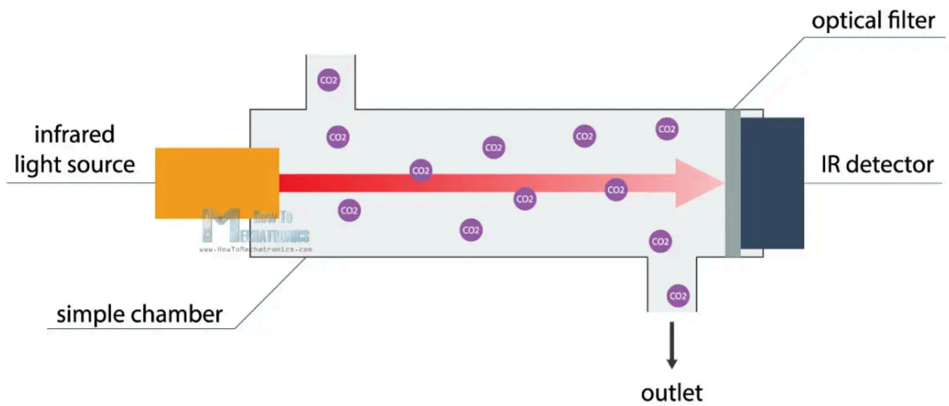
There are many premade libraries for DHT22 and we use one of the libraries for convenience. We decided to go with the *DHT22* library. The data pin on the DHT22 can be connected to any I/O pin on the arduino. The general usage of the library is described below:

```
#include <DHT22.h>
#define DHTpin 5
DHT22 dht22(DHTpin);

void setup(){
  Serial.begin(9600);
}

void DHTread() {
  float t = dht22.getTemperature();
  float h = dht22.getHumidity();

  if (dht22.getLastError() != dht22.OK) {
    Serial.print("last error :");
    Serial.println(dht22.getLastError());
  }

  Serial.println(t,h);
}

void loop() {
  DHTread();
}
```

Refer to the DHT22 library for more details.

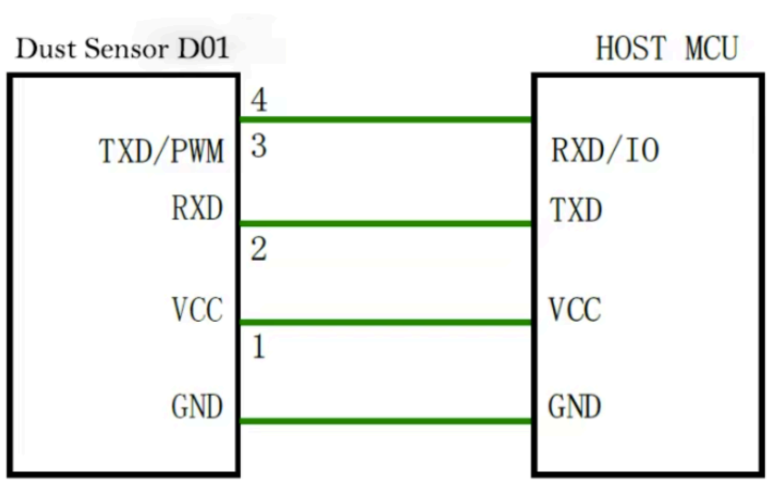
DC-01 infrared pm2.5 sensor

The DC-01 sensor detects particulate matter using an infrared LED and an IR detector. The following picture demonstrates an infrared CO2 detector, but the working principle is identical.

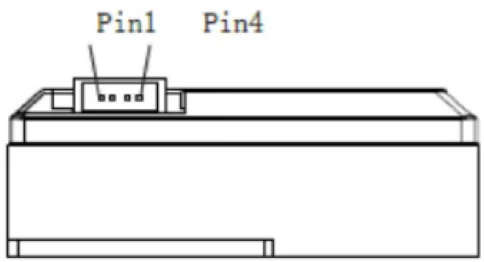


The sensor has a hollow chamber with two openings at the top and bottom to allow maximal airflow. The infrared light from the LED gets deflected by these particles; the amount of light captured by the IR detector depends on how much light is deflected i.e. how many particulates are in the air. **Note**: No well-organized datasheet for the DC-01 is available currently, though this is subject to change. However, the following documentation should include all the relevant information needed for setting up the DC-01 sensor.

The pinout of the sensor's application circuit is shown below:



The pin diagram and a table describing the pin parameters:



Pin diagram

Pin number	Pin name definition	Pin function description	Pin electrical characteristics
Pin 1	GND	power negative	no reverse protection
Pin 2	VCC	power positive(+5V)	
Pin 3	RXD	RXD pin of the module UART interface	TTL level @ 5V
Pin 4	TXD	TXA pin of the module UART interface	open-circuit output, internal pull-up resistor connected to the power positive

Pin table

The DC-01 uses UART for the communication between the sensor and MCU. The **UART configuration** is listed below.

- Baud rate: 9600bps; Check bit: None; Stop bit: 1bit; Data bit: 8bits
- A frame of serial output data includes 4 bytes, and the data format is show in this table:

Characteristic Byte	Byte 1	Byte 2	Check Byte
0xA5	DATAH	DATAL	SUM

Where,

- DATAH is the high 7 bits of concentration value
- DATAL is the low 7 bits of concentration value
- SUM is the low 7 bits of the raw sum of the three bytes before the check byte
- Serial Data conversion formula: Concentration value = DATAH * 128 + DATAL
    - Note this is the concentration for PM 10. The PM 2.5 dust concentration value can be obtained by calibrating the PM 10 concentration value with a K value coefficient based on the TSI instrument's photometric method. It is generally recommended to use 0.4.
    - We tested our PM 2.5 sensor with an existing Purple Air Monitor on campus and determined a calibration constant of 1/3.
    - Therefore, PM 2.5 concentration value = (DATAH * 128 + DATAL)/3

Setting up the Endpoint Node

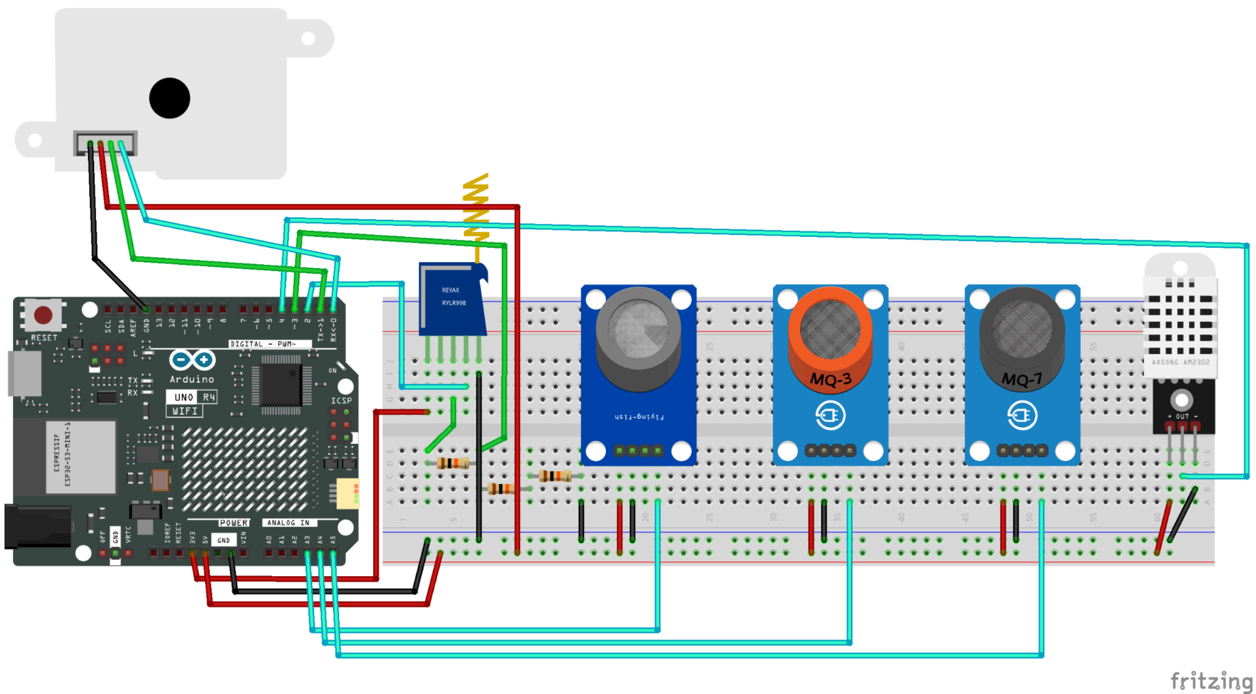
In this section, we will show you how to build an endpoint node. These nodes are what include the sensors to actually monitor the air quality. Since the base design of this node is identical to the gateway node, we will start building from the finished node. So, start by building another gateway node, and then come back to this point to build the endpoint.

Besides the parts of the gateway, there are multiple sensors that you can add to the endpoint. They are not all required, and we will tell you what needs changed later on to accommodate what sensors you do and don't use, but here is the complete list of parts you would need if you have everything installed:

- 1x MQ-3 sensor
- 1x MQ-7 sensor
- 1x MQ-135 sensor
- 1x DHT22 Temp. & Humidity sensor
- 1x DC01 PM-2.5 sensor

Once you have the parts you want, please follow [this video](https://youtu.be/0irOUgVkDII) that will guide you through building an endpoint node or visit https://youtu.be/0irOUgVkDII.

To ensure that you have correctly built the endpoint as shown in the above video, compare your work to the diagram below. These will show which pins of the arduino are connected to the breadboard and transmitter.



Arduino
UNO R4
WIFI
DIGITAL ~ PWM~
POWER
ANALOG IN
ICSP
RESET
ESPRESSIF ESP32-S3-MINI-1
REYAX
RYLR998
MQ-3
MQ-7
fritzing

4. Air Pollutants - Which Ones Matter?

As previously explained in the parts section the sensors attached to each air quality monitor are not particularly set and different sensors can be added or removed in order to detect different pollutants in different environments. This section will look at the core air pollutants we tested with and compare their importance weighed against the space they take up and the cost of their associated sensors. Keep in mind that if an air pollutant is not included in this list it merely means we haven't tested it and you are virtually guaranteed to find a sensor compatible with these air quality monitors for any pollutant you'd like to track.

Particulate Matter - PM 2.5

Particulate Matter is perhaps the first category of air pollutant people think of when discussing air quality. It can be described in different sizes, the most popular being PM 2.5 which refers to fine inhalable particles with a size of 2.5 micrometers and smaller. They often consist of particles produced by combustion, organic compounds, and metals. PM 10 refers to slightly large inhalable particles with a size of 10 micrometers and smaller. They often consist of dust, pollen, and mold. Most particulate matter typically comes from human development such as construction sites, unpaved roads, fields, smokestacks, fires, etc. They are more of an outdoor pollutant and their tracking is an absolute essential on any good air quality monitor. PM 2.5 particles are particularly harmful to health as it is quite easy for them to deposit themselves deeply into the lungs and bloodstream of those who inhale them. We strongly encourage anyone doing this project to include a good PM 2.5 sensor like the one we included in our parts list for $10. They are usually larger than the other sensors and maybe a bit more expensive in terms of price and energy consumption, but their inclusion is strongly recommended.

Carbon Dioxide

Carbon dioxide is an air pollutant commonly connected to global warming, an issue most would consider can only be monitored at a larger scope than this project. This is a reasonable connection but it must also be noted that carbon dioxide can have negative health effects in local areas in large concentrations. Carbon dioxide is most often emitted from the burning of fossil fuels, common examples include the burning of coal for energy and the burning of gas in cars for fuel. Carbon Dioxide is an important pollutant to monitor and one that most people are more likely to understand. The sensor we used to monitor carbon dioxide is the MQ-135 sensor.

Carbon dioxide sensors are not expensive and are certainly worth having on your air quality monitor, particularly if you want to monitor the pollutants coming from Traffic in your area. It is not absolutely essential if your objective is to detect pollutants that are a dire threat to human health as it is often associated with more environmental concerns rather than health concerns.

Carbon Monoxide

Carbon monoxide is notorious for being a dangerous gas, particularly indoors. It is infamous for being colorless, odorless, tasteless, and deadly. It has many sources of emissions, including many household objects such as furnaces, generators, grills, and anything that burns anything for fuel. It can also come from cars when gas is burned for fuel. Carbon monoxide poisoning affects the brain and the heart the most, and some of its symptoms include headache, weakness, dizziness, nausea, loss of muscle control, and loss of consciousness. Carbon monoxide is the most dangerous when it is emitted in an area with poor ventilation. For this reason we strongly recommend including a sensor for it if you plan on monitoring the air quality of an indoor space. We currently use the MQ-7 sensor for carbon monoxide detection. It is also important to recognize that carbon monoxide is not just harmful indoors and for this reason we recommend you detect it even if your monitor(s) will be stationed outside. As mentioned earlier, carbon monoxide is emitted in large quantities from vehicles. If there is enough traffic and the airflow in a particular area is somewhat lacking, concentrations of carbon monoxide can build up and be particularly dangerous for those with heart problems.

Alcohol

Alcohol is a pollutant not many think of, but it can cause some environmental and health issues. Ethanol is considered a volatile organic compound (VOC) and when released into the atmosphere it can contribute to the formation of ozone pollution, one of the key components of smog. Both Ethanol and methanol are known for having negative consequences on human health, particularly in closed environments. Ethanol is a known carcinogen and is well researched given its use in alcoholic beverages. Both Ethanol and methanol are known to cause eye irritation in high enough concentrations in the air. Some sources of alcohol as a pollutant are the ethanol found in gasoline, industrial solvents and adhesives, ethanol released in the brewing process, and from exhalation, particularly after alcoholic beverages are consumed. The health effects of being exposed to alcohol in the air and the environmental effects are a lot less severe than some of the other pollutants, so we do not strongly recommend including a sensor for it unless you are placing your monitor near a potent source of alcohol emissions such as a gas station, brewery, or anywhere with an abundant amount of industrial chemicals. We are using the MQ-3 sensor for alcohol detection. The effects of inhaling alcohol are currently being researched more and so more findings on the harm of alcohol emissions may come out soon.

Nitrogen Dioxide

Nitrogen dioxide is a potentially harmful gas for both human health and the environment. It is known for irritating human respiratory systems, aggravating respiratory diseases such as asthma, increasing susceptibility to respiratory infections, and when mixed with other chemicals in the care can form particulate matter and ozone, both of which are also harmful to respiratory health. Nitrogen Dioxide is also very uniquely harmful to the environment as it is the main component of the creation of acid rain, an environmental event that is very harmful to sensitive ecosystems such as lakes and forests. Nitrate particles can also make the air hazy, reducing visibility. Nitrogen Dioxide is emitted from the burning of fuel, some common sources including cars and power plants. You may notice that we haven't included a nitrogen dioxide sensor in our default air quality monitor. This is for two reasons that are important to consider. The first is that the cheapest nitrogen dioxide sensor we could find on the market is $20, considerably expensive for a project that considers budget a top priority. The second reason is that the sensor is not set up to be used on a breadboard and requires soldering for installation. This isn't too difficult but we aim to make this project as accessible as possible for people of all hardware skill levels. If your aim is to monitor one of the gases responsible for acid rain or you are more focused on the environmental effects of air pollutants we certainly recommend a nitrogen dioxide sensor, assuming it fits your budget and skill capacity.

Sulfur Dioxide

Sulfur dioxide is similar in many ways to nitrogen dioxide in its harmful effects and source of emission. It can make breathing more difficult and can be especially harmful to those with asthma. It can mix with other chemicals in the air to form particulate matter, another pollutant harmful to human health. Similar to nitrogen dioxide, sulfur dioxide is a core component of acid rain, a harmful threat to sensitive ecosystems. In addition it can also harm trees and plants by damaging and decreasing the growth of foliage. Sulfur dioxide can also react with other atmospheric compounds to produce haze, reducing visibility. Deposition of sulfur dioxide can damage materials such as stone, meaning it can have cultural consequences from the degradation of important statues and monuments. Sulfur dioxide is emitted from the burning of fossil fuels, usually coming from power plants and industrial facilities. It can also come from any industrial process involving the burning of fuel with a high sulfur content, and can even come from volcanoes. We recommend sulfur dioxide for the same reasons as nitrogen dioxide, it is an important pollutant to detect when focused on environmental issues. It also would make more sense for a monitor deployed in an area with heavy industrial activity as opposed to an area exposed to common vehicle traffic. The sensor for sulfur dioxide has the same downsides as the one we found for nitrogen dioxide; The price is $20 and requires soldering to be attached to a monitor. As technology develops the price for these sensors may come down making these sensors more feasible.

Temperature & Humidity

If you are only looking to measure pollutants with the cheapest possible monitor configuration you don't necessarily need to measure temperature and humidity. However the sensor is very cheap and having more information never hurts, so we recommend you include a temperature and humidity sensor. The one we are using is the DHT11 sensor.

5. Works Cited

Basic Information about NO2 | US EPA. (2024, July 16). US EPA. https://www.epa.gov/no2-pollution/basic-information-about-no2#:~:text=Nitrogen%20Dioxide%20(NO2)%20is,larger%20group%20of%20nitrogen%20oxides.

Brewery/Winery/Distillery | Department of Environmental Conservation. (n.d.). https://dec.vermont.gov/air-quality/permits/source-categories/breweries#:~:text=While%20ethanol%20is%20the%20desired,main%20components%20of%20%E2%80%9Csmog%E2%80%9D.

Carbon Monoxide Poisoning Basics. (2024, April 17). Carbon Monoxide Poisoning. https://www.cdc.gov/carbon-monoxide/about/index.html

Lee, J. (2022, February 11). *Everything to know about carbon dioxide (CO2)*. Breeze Technologies. https://www.breeze-technologies.de/blog/carbon-dioxide-co2/#:~:text=Is%20carbon%20dioxide%20an%20air,and%20methane%20(CH4).

Particulate Matter (PM) Basics | US EPA. (2024, June 20). US EPA. https://www.epa.gov/pm-pollution/particulate-matter-pm-basics

Sulfur Dioxide Basics | US EPA. (2024, January 31). US EPA. https://www.epa.gov/so2-pollution/sulfur-dioxide-basics

Nazaroff, W. N., & Weschler, C. W. (2024). Methanol and ethanol in indoor environments. *ScienceDirect*, *1*(4), 100049. https://www.sciencedirect.com/science/article/pii/S2950362024000468