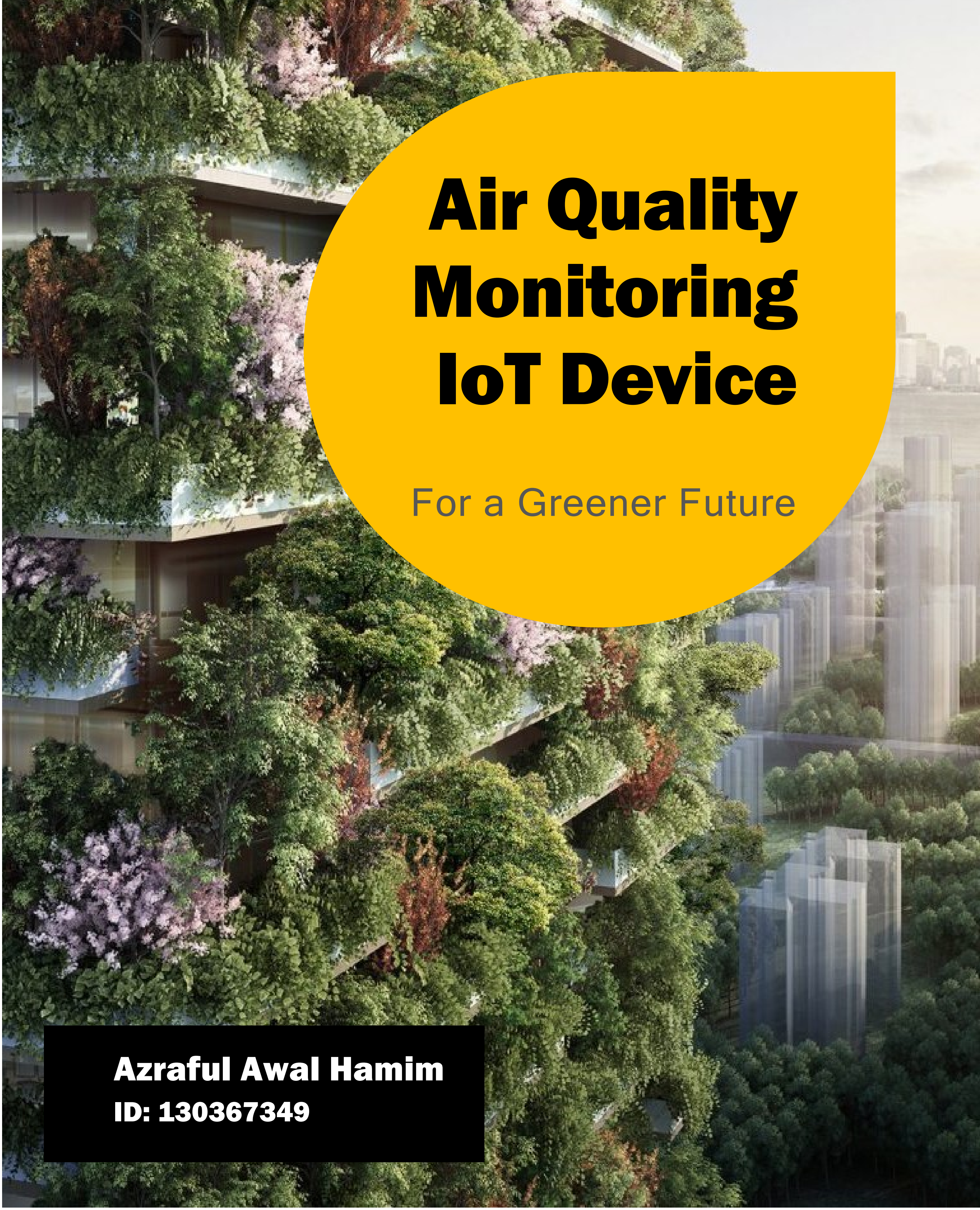


Air Quality Monitoring IoT Device

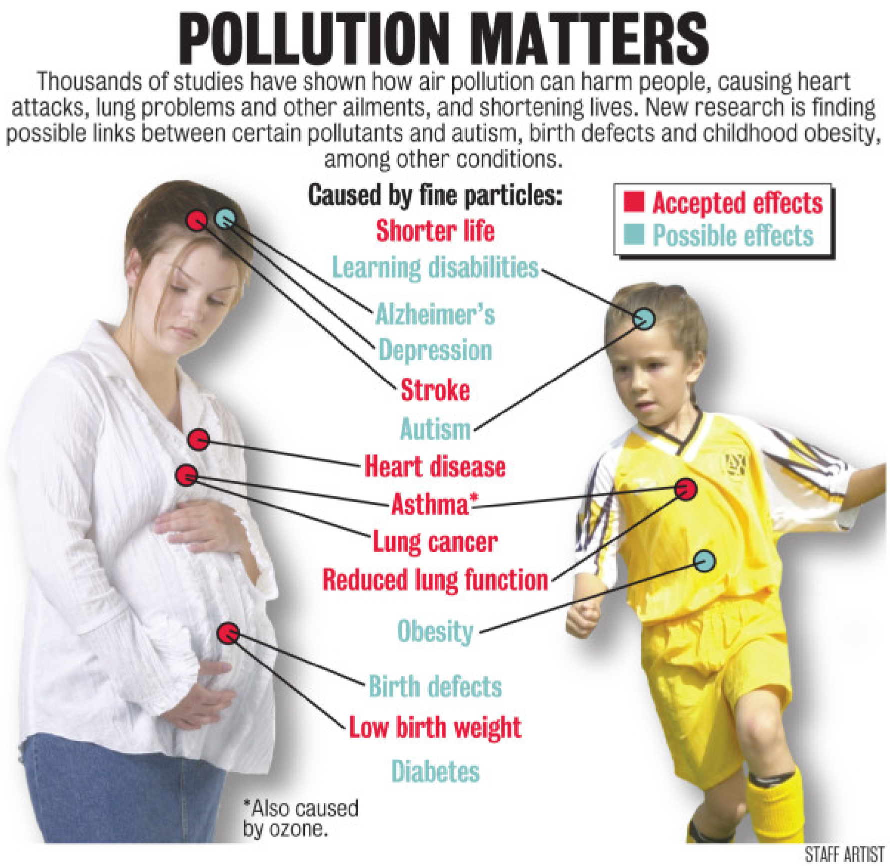
For a Greener Future

Azraful Awal Hamim
ID: 130367349

Introduction

Since the advent of the industrial revolution in the 18th century, air pollution has become a major environmental concern. Air pollution is the introduction of chemicals, particulate matter, or biological materials that cause harm or discomfort to humans or other living organisms, or damages the natural environment, into the atmosphere. The atmosphere is a complex, dynamic natural gaseous system that is essential to support life on planet Earth. Stratospheric ozone depletion due to air pollution has long been recognized as a threat to human health as well as to the Earth's ecosystems. (Mask, 2018)

Air pollution can cause various hazardous effects on life, which is briefly described by the following image:

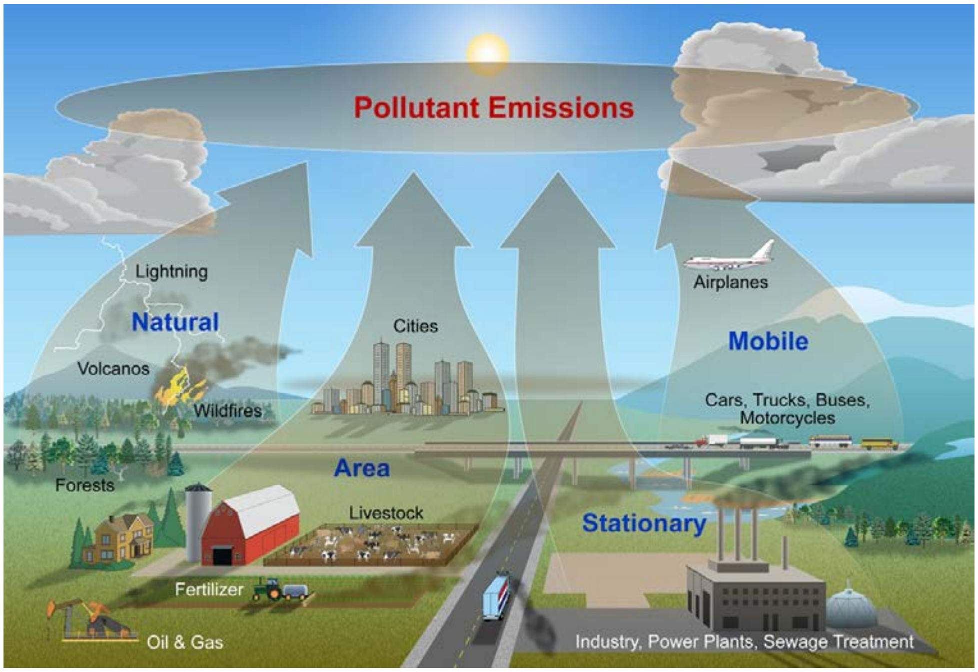


(FujiTak, 2015)

An air pollutant is known as a substance in the air that can cause harm to humans and the environment. Pollutants can be in the form of solid particles, liquid droplets, or gases. In addition, they may be natural or man-made. Pollutants can be classified as either primary or secondary. Usually, primary pollutants are substances directly emitted from a process, such as ash from a volcanic eruption, the carbon monoxide gas from a motor vehicle exhaust or sulfur dioxide released from factories.

Secondary pollutants are not emitted directly. Rather, they form in the air when primary pollutants react or interact. An important example of a secondary pollutant is ground level ozone – one of the many secondary pollutants that make up photochemical smog. Note that some pollutants may be both primary and secondary: that is, they are both emitted directly and formed from other primary pollutants.

There are many sources contributing to air pollution, which is illustrated in the following image:



(Mask, 2018)

A lack of ventilation indoors concentrates air pollution where people often spend the majority of their time. Radon (Rn) gas, a carcinogen, is exuded from the Earth in certain locations and trapped inside houses. Building materials including carpeting and plywood emit formaldehyde (H2CO) gas. Paint and solvents give off volatile organic compounds (VOCs) as they dry. Lead paint can degenerate into dust and be inhaled. Intentional air pollution is introduced with the use of air fresheners, incense, and other scented items. Controlled wood fires in stoves and fireplaces can add significant amounts of smoke particulates into the air, inside and out. Indoor pollution fatalities may be caused by using pesticides and other chemical sprays indoors without proper ventilation.

Carbon monoxide (CO) poisoning and fatalities are often caused by faulty vents and chimneys, or by the burning of charcoal indoors. Chronic carbon monoxide poisoning can result even from poorly adjusted pilot lights. Traps are built into all domestic plumbing to keep sewer gas, hydrogen sulfide, out of interiors. Clothing emits tetrachloroethylene, or other dry cleaning fluids, for days after dry cleaning.

Though its use has now been banned in many countries, the extensive use of asbestos in industrial and domestic environments in the past has left a potentially very dangerous material in many localities. Asbestosis is a chronic inflammatory medical condition affecting the tissue of the lungs. It occurs after long-term, heavy exposure to asbestos from asbestos-containing materials in structures. Sufferers have severe dyspnea (shortness of breath) and are at an increased risk regarding several different types of lung cancer. As clear explanations are not always stressed in non-technical literature, care should be taken to distinguish between several forms of relevant diseases. According to the World Health Organisation (WHO), these may defined as; asbestosis, lung cancer, and mesothelioma (generally a very rare form of cancer, when more widespread it is almost always associated with prolonged exposure to asbestos).

Biological sources of air pollution are also found indoors, as gases and airborne particulates. Pets produce dander, people produce dust from minute skin flakes and decomposed hair, dust mites in bedding, carpeting and furniture produce enzymes and micrometre-sized fecal droppings, inhabitants emit methane, mold forms in walls and generates mycotoxins and spores, air conditioning systems can incubate Legionnaires' disease and mold, and houseplants, soil and surrounding gardens can produce pollen, dust, and mold. Indoors, the lack of air circulation allows these airborne pollutants to accumulate more than they would otherwise occur in nature.

(Mask, 2018)

Use Cases

Air Quality Monitoring usage should be emphasized not only in households but also commercial buildings and even in public spaces. The usage of Air Quality Monitoring devices should be enforced especially in industrial areas, where the impact on pollution may be the most. For example, industry-heavy countries such as China and India, contain the most polluted environments. Resultantly, the residents would be intensely victimized unless they are always alert and aware of their invisible surroundings.

An Air Quality Monitoring IoT Device would solve the problem of residents in pollution-heavy areas to become critically aware of their air quality and take precautionary measurements to avoid sustainable health concerns. The device would be able to measure simple or complex parameters about the quality of air around them in real-time. It could even have potential for the analysis of the collected data over periods of time, to monitor improvements or deterioration in the air quality.

This device can be very helpful for end-users at their homes and also commercial businesses to monitor their surroundings more efficiently and in real-time.

System Specifications

In order to build an Air Quality Monitoring (Internet of Things) Device, we need the following physical electronic hardware components and assemble them together:

- Arduino Uno Microcontroller
- Base Shield V2
- Grove Temperature & Humidity Sensor
- Grove Air Quality Sensor
- SNS-MQ135 Gas Sensor

On the other hand, we need **Arduino IDE** to program the microcontrollers so that they can process the valuable input data from the sensors and produce useful output visualisation of the various parameters, intended for the users to become more aware of their environments.

Yes, it indeed only requires 5 physical electronic components for its assembly. In the next section, it is highlighted how a simple and cost-effective design it is, yet fulfilling its purpose efficiently.

Design & Architecture

The architecture of such an Internet of Things (IoT) Device can be quite cheap and simple. In other cases, it could also be very complex, if we wish to measure other complex data parameters and connect it to the cloud for further analysis. However, my design for the Air Quality Monitoring Device is aimed to be as simple and cost-effective as possible, while still fulfilling its basic purpose effectively and efficiently.

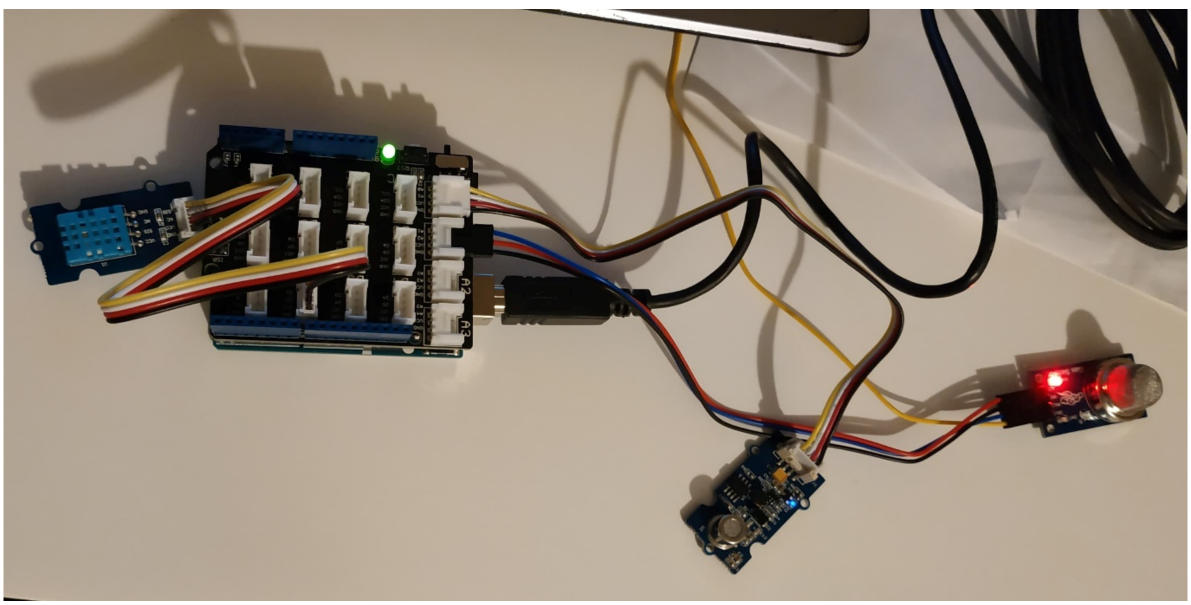
The following table gives a breakdown of the costs of all the components required to assemble such a device:

Component	Cost (GBP)
Arduino Uno Microcontroller	18.90
Base Shield V2	8.65
Grove Temperature & Humidity Sensor	5.28
Grove Air Quality Sensor	8.48
SNS-MQ135 Gas Sensor	5.35
Complete Device	**46.66**

Hence, it is possible to build a simple Air Quality Monitoring device in less than £50, which can efficiently measure data for the most important air quality deterministic factors such as temperature, humidity, harmful gases, dust particles, etc. However, this design is aimed at a simplified DIY (do it yourself) model, which can be assembled by anyone at home without the necessity of having engineering or coding skills. In theory, the design can be even cheaper if we were to use a more specific processing unit instead of a multi-purpose microcontroller. But, the cost of £47 for an Air Quality Monitoring device which can measure the most crucial data from the air within its simplicity, is quite a cost-effective design by itself.

In my simple design, there are no requirements of bulky or multitudes of tangled wires within the circuit. In fact, we do not even require the usual breadboard, battery or external memory devices. We only need a processing unit, like Arduino Uno which powers and programs the sensors. Furthermore, we need a Base Shield which is compatible to the microcontroller (in this case, Arduino), that can be a connecting hub for all the sensors.

After the assembling of all the electronic components, the device will have a neat getup as follows:



The design of this device setup can be further improved by making it portable. In order to do that, we simply need to treat it as a mobile device that needs an external power supply. Because the Arduino Uno is connected via a simple USB port, any smartphone power bank can simply do the trick of powering this device. It will still be able to collect all the data and if synced with a smartphone app, can also display the variable data with visualisation.

System Analysis and Evaluation

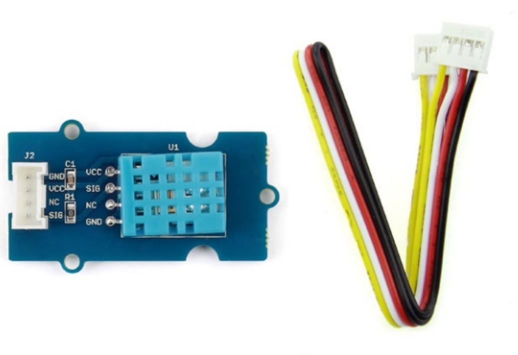
Each component of the whole device has a specific purpose. For example, the multiple sensors gather information on various environmental input parameters. The microcontroller acts as a processing unit to display the output useful information.

Grove Temperature & Humidity Sensor

This Temperature & Humidity sensor provides a pre-calibrated digital output. A unique capacitive sensor element measures relative humidity and the temperature is measured by a negative temperature coefficient (NTC) thermistor. It has excellent reliability and long term stability. Please note that this sensor will not work for temperatures below 0 degree.

The features of this sensor are as follows:

- Relative Humidity and temperature measurement
- Full range temperature compensation Calibrated
- Digital signal
- Long term stability
- Long transmission distance(>20m)
- Low power consumption
- Voltage: 3.3V to 5V
- Current: 1.3mA to 2.1mA
- Temperature Range: 0 - 50 ℃



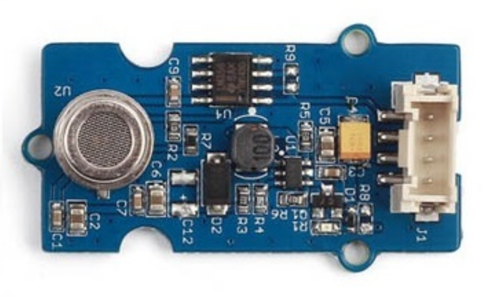
(Grove - Temperature&Humidity Sensor, n.d.)

Grove Air Quality Sensor

This sensor is designed for comprehensive monitor over indoor air condition. It's responsive to a wide scope of harmful gases, as carbon monoxide, alcohol, acetone, thinner, formaldehyde and so on. Due to the measuring mechanism, this sensor can't output specific data to describe target gases' concentrations quantitatively. But it's still competent enough to be used in applications that require only qualitative results, like auto refresher sprayers and auto air cycling systems.

The following are the features of the sensor:

- Responsive to a wide scope of target gases
- Cost efficient
- Durable
- Compatible with 5V and 3.3V



(Grove - Air Quality Sensor v1.3, 2016)

SNS-MQ135 Gas Sensor

Some features are as follows:

- Detects NH3, NOx, alcohol, benzene, smoke, CO2, etc.
- Operating voltage 5V
- Operating current 40mA
- Both digital and analog output
- Popular sensor - a lot of examples available for different platforms
- Port explanation: AO for analog output, DO for digital output, GND for ground, VCC for 5V voltage input

(SNS-MQ135, n.d.)

Arduino Uno

The Arduino UNO is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable.

The technical specifications of Arduino Uno are as follows:

- Microcontroller: Microchip ATmega328P
- Operating Voltage: 5 Volt
- Input Voltage: 7 to 20 Volts
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by bootloader
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- Length: 68.6 mm
- Width: 53.4 mm
- Weight: 25 g

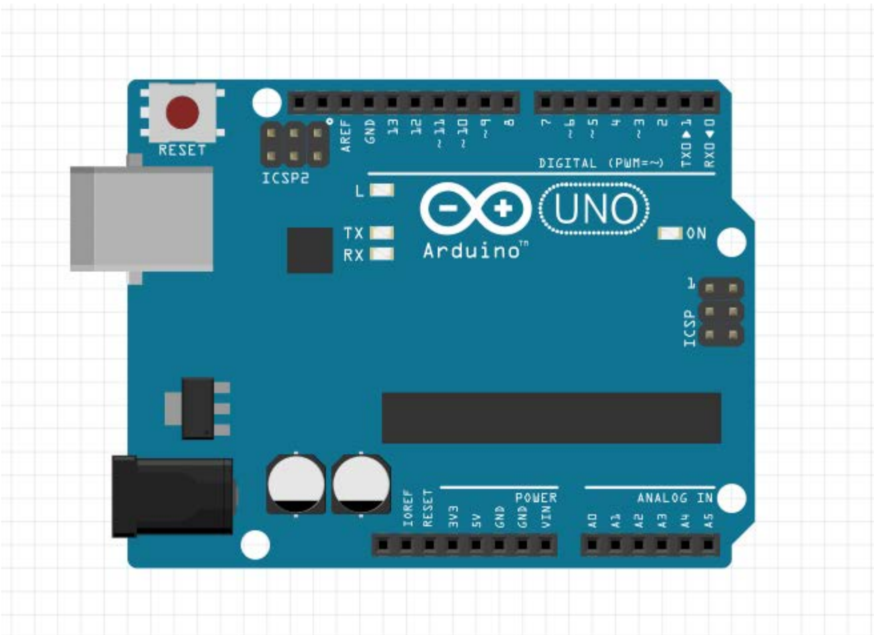
The general Pin functions are as follows:

- **LED**: There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **VIN**: The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V**: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.
- **3V3**: A 3.3 volts supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND**: Ground pins.
- **IOREF**: This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.
- **Reset**: Typically used to add a reset button to shields which block the one on the board.

In addition, some pins have specialised functions:

- **Serial**: pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts**: pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

- **PWM** (Pulse Width Modulation) 3, 5, 6, 9, 10, and 11 Can provide 8-bit PWM output with the analogWrite() function.
- **SPI** (Serial Peripheral Interface): 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- **TWI** (Two Wire Interface): A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.
- **AREF** (Analog REFerence): Reference voltage for the analog inputs.

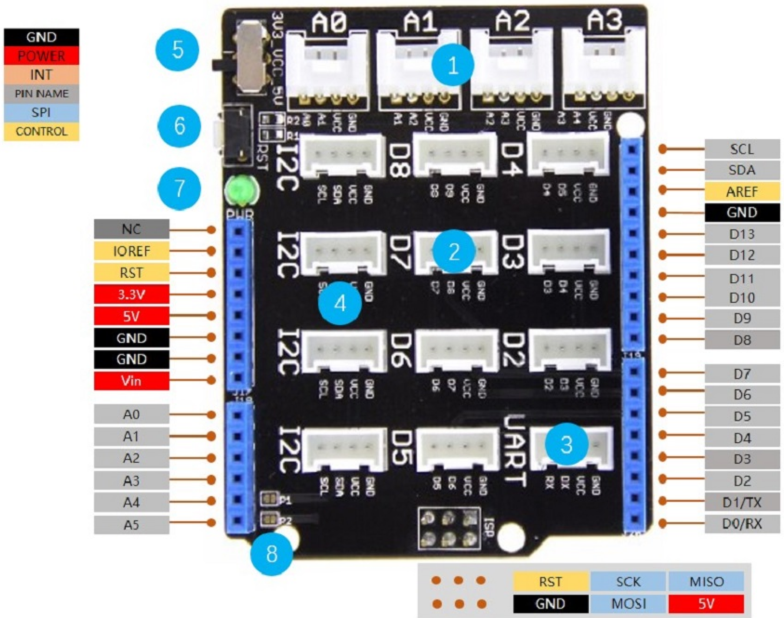


(Arduino Uno Rev3, 2018)

Base Shield V2

Arduino Uno is the most popular Arduino board so far, however it is sometimes frustrating when your project requires a lot of sensors or Leds and your jumper wires are in a mess. The purpose of creating the Base Shield is to help you get rid of bread board and jumper wires. With the rich grove connectors on the base board, you can add all the grove modules to the Arduino Uno conveniently! The pinout of Base Shield V2 is the same as Arduino Uno R3.

The following diagram illustrate the various pins and ports on the Base Shield V2 which sits on top of an Arduino Uno perfectly, before various sensors are connected to it:



- **Analog Ports**: include 4 anlog ports, A0, A1, A2 and A3.
- **Digital Ports**: include 7 digital ports, D2, D3, D4, D5, D6, D7 and D8.
- **UART Port**: 1 UART port.
- **I2C Ports**: 4 I2C ports.
- **Power Switch**: when using Arduino UNO with Base Shield v2, please turn the switch to 5v position; While using Seeeduino Arch with Base Shield v2, please turn the switch to 3.3v.
- **Reset Buton**: reset the arduino board.
- **PWR LED**: the Green LED turns on when power on.
- **P1, P2**： please solder the pads P1 and P2, if use Base Shield v2 with Seeeduino V3.
- **Dimension**: 2.1 * 2.7 inch

(Base Shield V2, 2014)

Implementation

Once the device hardware components are all set up, we are ready to implement the code for the operation of the device. The Arduino programming language is merely a set of C or C++ functions called from the code sketch. The compiler compiles the code which sets up the pins connected to the sensors, gets the sensor readings and displays them on a platform of choice. To keep things simple and just testing out the device, I am using the Arduino IDE Serial Monitor for the data readings display.

The following code implemented in Arduino IDE completes the initialisation, setup and operation of the sensors via the Arduino Uno microcontroller:

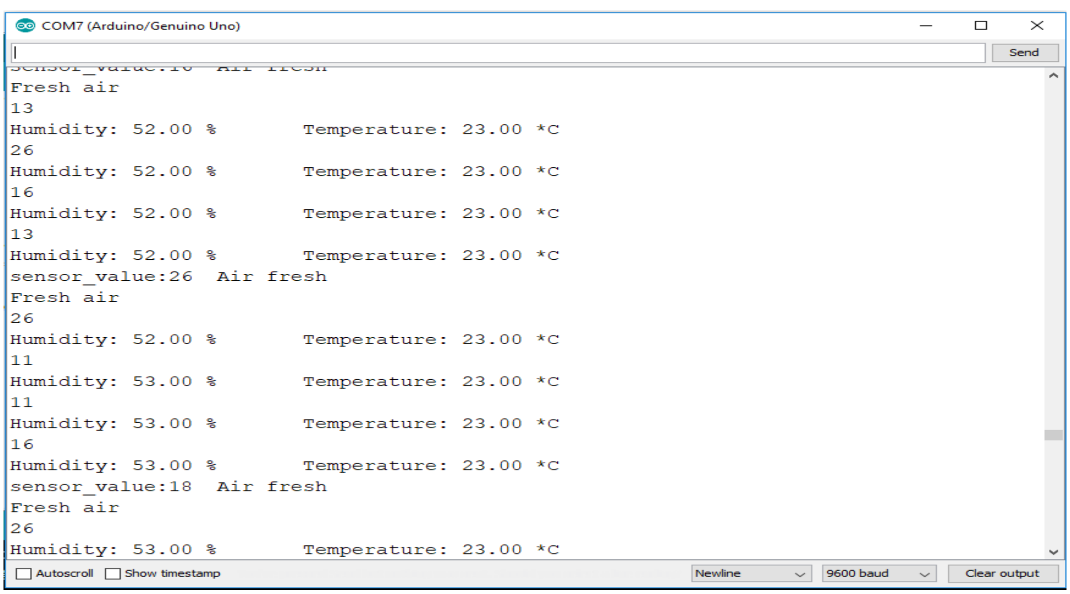
```
// Air Quality Monitor IoT Device by Azraful Awal Hamim

#include"AirQuality.h"
#include"Arduino.h"
AirQuality airqualitysensor;
int current_quality =-1;
#include "DHT.h"
#define DHTPIN 2     // what pin we're connected to
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
int sensorValue;

void setup()
{
   Serial.begin(9600);
   airqualitysensor.init(A0);
   Serial.println("DHTxx test!");
   dht.begin();
}
void loop()
{
    sensorValue = analogRead(0); // reads analog input pin 0
    Serial.println(sensorValue, DEC); // prints the value read
   // Reading temperature or humidity takes about 250 milliseconds!
   // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
   float h = dht.readHumidity();
   float t = dht.readTemperature();
   // check if returns are valid, if they are NaN (not a number) then something went wrong!
if (isnan(t) || isnan(h))
```

```
{
        Serial.println("Failed to read from DHT");
    }
    else
    {
        Serial.print("Humidity: ");
        Serial.print(h);
        Serial.print(" %\t");
        Serial.print("Temperature: ");
        Serial.print(t);
        Serial.println(" *C");
    }
    current_quality=airqualitysensor.slope();
    if (current_quality >= 0)// if a valid data returned.
    {
        if (current_quality==0)
            Serial.println("High pollution! Force signal active");
        else if (current_quality==1)
            Serial.println("High pollution!");
        else if (current_quality==2)
            Serial.println("Low pollution!");
        else if (current_quality ==3)
            Serial.println("Fresh air");
    }
  }
```

```
ISR(TIMER2_OVF_vect)
{
   if(airqualitysensor.counter==122)//set 2 seconds as a detected duty
   {
      airqualitysensor.last_vol=airqualitysensor.first_vol;
      airqualitysensor.first_vol=analogRead(A0);
      airqualitysensor.counter=0;
      airqualitysensor.timer_index=1;
  PORTB=PORTB^0x20;
   }
   else
   {
     airqualitysensor.counter++;
   }
}
```

From the 3 different sensors, the output readings of the sensors should show Humidity (percentage), Temperature (Degrees Celsius), Gas Rating (Integer), Air Quality Rating (Integer and Remark).

The output result of the sensor measurement data in the Arduino IDE Serial Monitor is as follows:



References

Arduino Uno Rev3. (2018). Retrieved from Arduino.cc: https://store.arduino.cc/arduino-uno-rev3

Base Shield V2. (2014, March). Retrieved from Seeed Studio: http://wiki.seeedstudio.com/Base_Shield_V2/

FujiTak. (2015, September 11). *WTS: HEPA or ULPA Standard Air Purifier View Single Post*. Retrieved from Hardwarezone: https://forums.hardwarezone.com.sg/97210827-post12.html

Grove - Air Quality Sensor v1.3. (2016, May). Retrieved from Seeed Studio: http://wiki.seeedstudio.com/Grove-Air_Quality_Sensor_v1.3/#version

Grove - Temperature&Humidity Sensor. (n.d.). Retrieved from Seeed Studio: http://wiki.seeedstudio.com/Grove-TemperatureAndHumidity_Sensor/

Mairs, J. (2017, February 22). *Stefano Boeri reveals plans for tree-covered towers in Nanjing*. Retrieved from Dezeen: https://www.dezeen.com/2017/02/22/stefano-boeri-nanjing-vertical-forest-plant-covered-towers-skyscraper-china/

Mask, S. (2018 , April 26). *The Basics About Air Pollution: What Is It?* Retrieved from Pollution Air Mask: https://www.pollutionairmask.com/the-basics-about-air-pollution-what-is-it/

SNS-MQ135. (n.d.). Retrieved from Olimex: https://www.olimex.com/Products/Components/Sensors/SNS-MQ135/