# Protocol Design for "Air Pollution Monitoring System Based on IoT using Actual Hardware"


NAMAL INSTITUTE

COMPUTER NETWORKS

Muhammad Sajid Hameed Khan

**2018-UET-NML-CS-13**

# Table of Contents

## Abstract

Air pollution is biggest problem in the whole world. In the modern Era, pollution is going on its peak due to vehicles, industries, over population, urbanization which is very dangerous for humans. Air pollution monitoring system based on IoT is used to monitor the environment using a webserver over the internet. This system will trigger an emergency alarm when level of poisonous gases like NOx, CO2 and smoke goes beyond a certain level.


## 1. Introduction

Air pollution is the biggest problem of every nation, whether it is developed or developing. Health problems have been growing at faster rate especially in urban areas of developing countries where industrialization and growing number of vehicles leads to release of lot of gaseous pollutants.

Air pollution monitoring system is application in IoT which uses actual hardware to monitor the environment. These days, IoT has a big scope and usage in daily life. Air pollution monitoring system will measure the level of poisonous gases like NOx, CO2 and smoke. This system will use sensor to measure the level of above-mentioned gases. The System will make an emergency alarm if the level of these gases crosses a certain level.

## 2. Components

Air pollution monitoring system will be consisted of following components:

I. Arduino
II. MQ135 sensor
III. Emergency Alarm
IV. Wi-Fi module
V. Client
VI. Server

## 3. Sensor

Sensor is major component of our system. We will be using the MQ135 sensor. This sensor can measure the level of CO2, NOx and smoke. MQ135 sensor gives the output in the form voltages levels.

## 4. Arduino Microcontrollers

### a) What is Arduino?

Arduino Microcontrollers are open-source electronics platform, on the base of easy usage, hardware and software. Arduino boars are able to read inputs like light on a sensor, temperature of room, level of poisonous gases. After reading the input, it converts it into an output like making an emergency alarm. We can make our Arduino board do any task which we want to be accomplished by sending a set of instructions microcontrollers on the board. For that purpose, we use Arduino programming language (which is based on wiring) and the Arduino Software or IDE based on processing.

### b) How to connect Arduino with sensor?

Sensor measures the level of gases and sends data to Arduino. For that purpose, first we have to connect sensor and Arduino through Wi-Fi so that they can work together. And then have to code on Arduino Software for which task we want to perform.

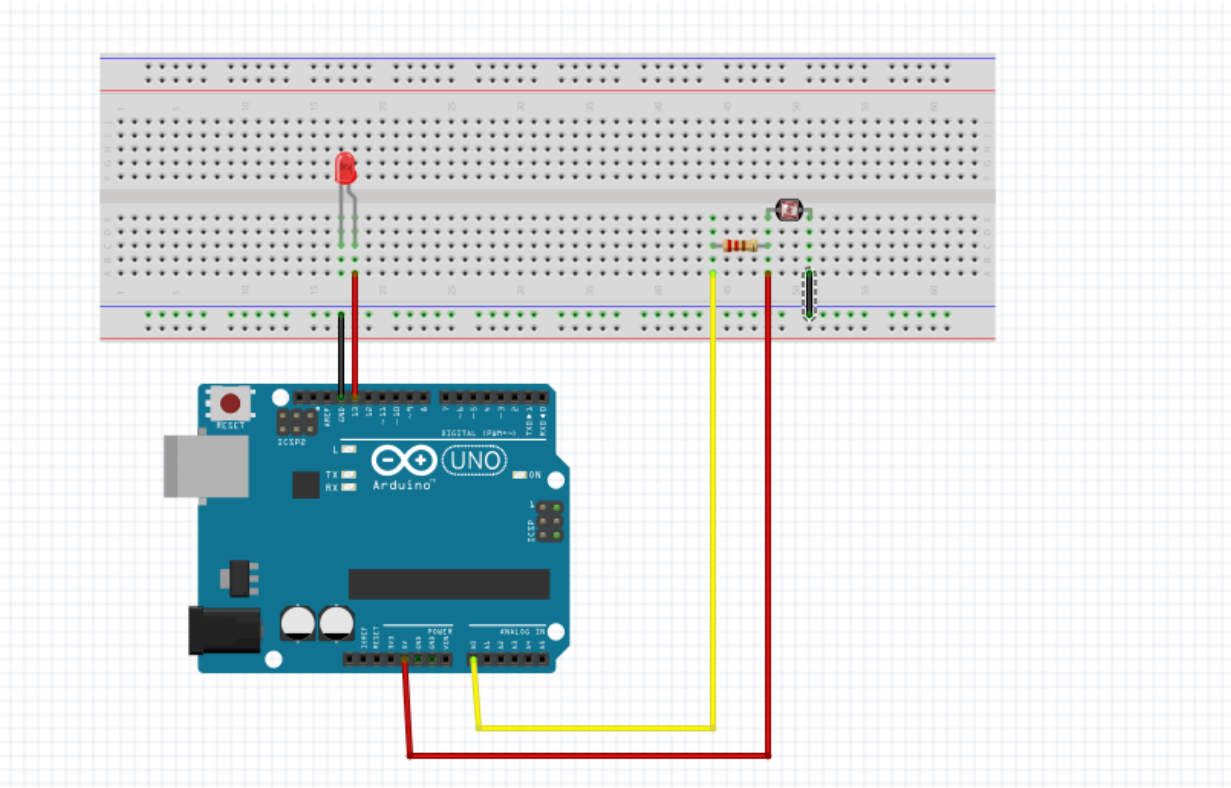


**Figure 1: How to connect Arduino?**

### c) Why Arduino?

There are some points due to which we use Arduino device.

**a. Cheap**

Arduino devices are cheap as compared to other microcontrollers.

**b. Open Source**

Arduino software is an open-source tool.

c. **Simple programming environment**
The Arduino Software is easy to use for novice users and it is also enough for the use of advanced users. As it is programming on programming environment that's why teacher and students can be easily familiar with how Arduino IDE works.

d. **Cross Platform**
The Arduino Software is a cross platform because it runs on Linux, Mac OS and Windows while most of the microcontrollers can only be run on Windows.

## 5. Application Functional Requirements

There some functional requirements of my application, which are given below:

I. Sensors detect the level of NOx, CO2 and smoke.
II. Arduino has to be connected with the sensor.
III. Arduino will send the data to Cloud Server, in which database will store the data.
IV. Data will be sent to application (client).
V. On the basis of level of gases, result will be sent back to the server.
VI. Result will be sent to Arduino device, and emergency alarm will start ringing if the level of gases is crossing certain threshold.
VII. Communication among the components will be sent through the protocols and in a specific format.

## 6. Application Non-functional Requirements

There are some non-functional requirements of my application, which re given below:

**I. Performance**
Our application should have a good performance. Performance involves response time, throughput, utilization.

**II. Availability**
Our application should provide continuous availability so that users can have ease to use whenever they want to use.

**III. Reliability**

Major requirement that our application should be reliable in data delivery.

**IV. Maintainability**

Maintainability is one major requirement that our application should be maintainable so that from documentation everyone can maintain the application in future.

**V. Scalability**

Scalability is the quality of any system to tackle the growing amount of work by adding resources to the system.

**VI. Serviceability**

Serviceability is the ease with which a deployed system can be maintained. Serviceability can include tasks like monitoring the system, adding and removing the users from the system and repairing the problems which arise in the system.

**VII. Usability**

Usability is the degree to which a software can be used by specific users to accomplished objectives with effectiveness, efficiency and satisfaction in a quantified context of use.

## 7. Application Structure

The Structure of my application is based on Client- Server model. My application will have some component in it, these components are shown below:

a) Sensors
b) Alarm
c) Arduino Device
d) Server
e) Database
f) Application
g) Area to be Monitored

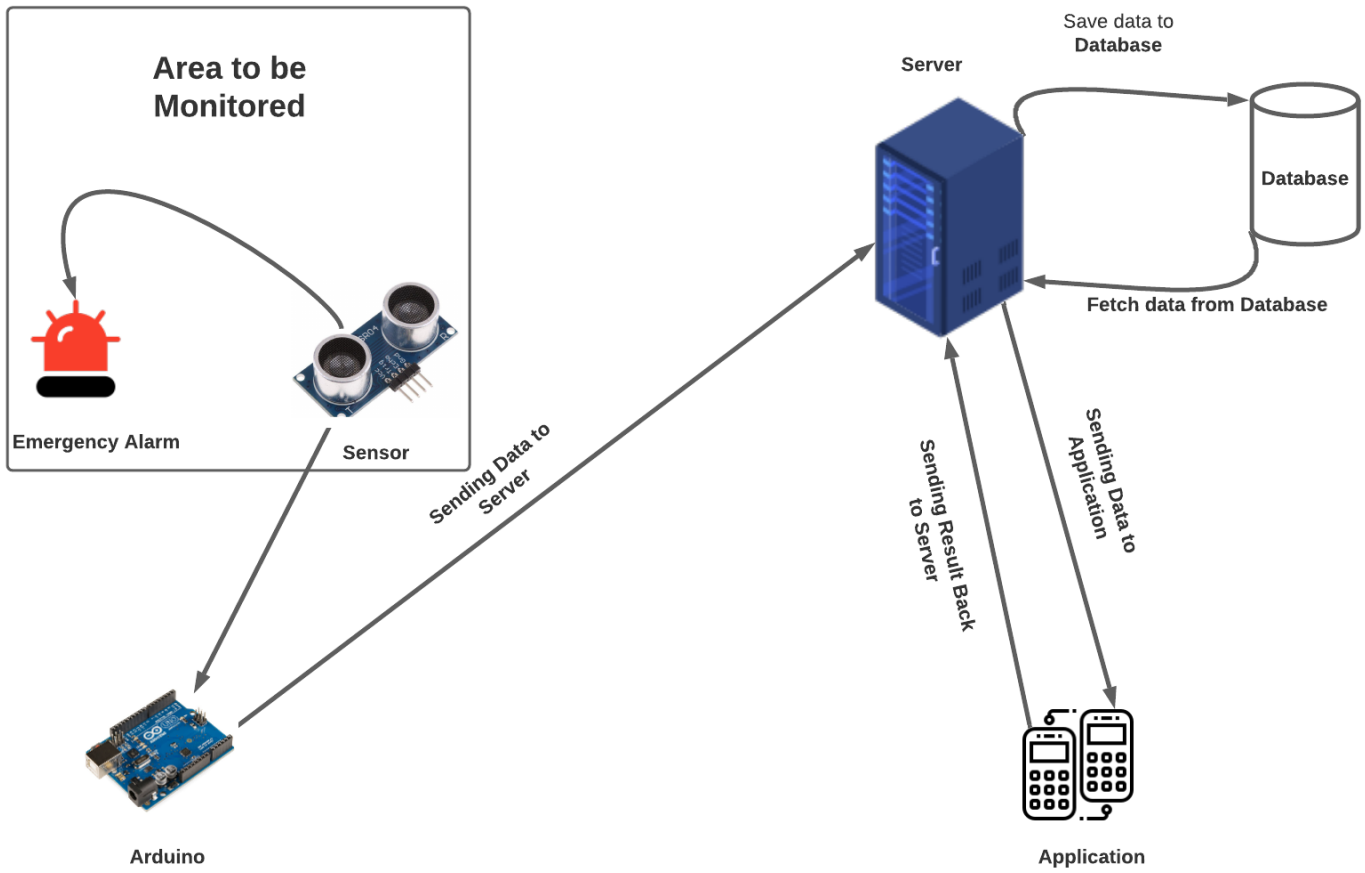


**Figure 2: Structure of Air Pollution Monitoring System**

In this structure of application, sensors will be placed in the area which is to be monitored. Sensors will be measuring the level of gases in that area. After every one-minute data, which sensors will be measuring, will send forward to the Arduino device. Sensors and Arduino device will be connected with each other. Arduino will send this data to server. The server will save the data to a database. Server will first send the data to application and the it will save that data to database. The application will check the level of gases that whether level of gases is beyond the certain limit or not. And application will check through some algorithm and send back the response to Server. Server will now save data (result) into database. Now if the level of gases is beyond the certain limit, then application will trigger emergency alarm otherwise alarm remain quiet.

## 8. Pattern of Communication

There are three patterns of communication.

a) Client – Server communication
b) Peer-to-Peer communication
c) Hierarchical communication

We are using Client-Server communication in our protocol. In this pattern of communication one party initiates the communication, and other responds.

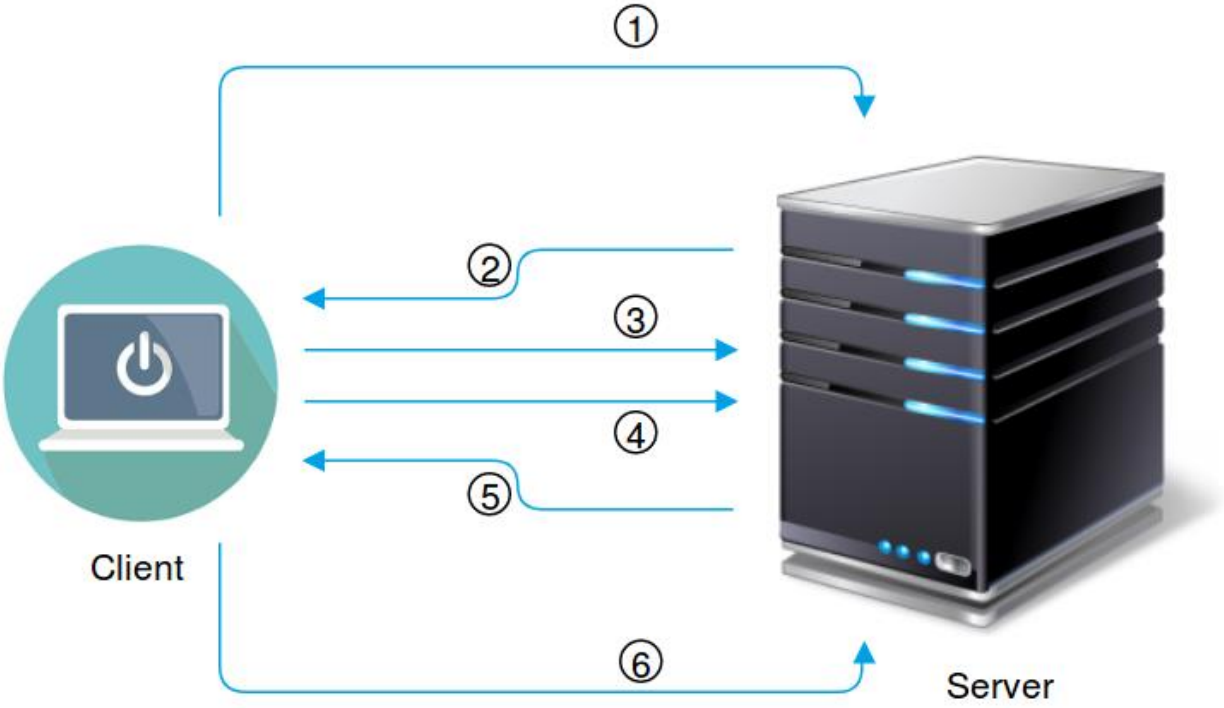


**Figure 3: Client-Server Communication**

## 9. Pattern of Data Transmission

In our application, pattern of data is one to one because when server is receiving data sending the data, only one task happens at a time.

## 10. The Protocol

### A. Message Type

There are two categories of messages in our application which are given below:

a) Command messages
b) Data transfer messages

In **command messages**, we client can request for the data. These messages are orders to any other component. For example, to establish a connection with server client requests and then server responds to the request of client.
In **data transfer messages,** we can transfer the data between different components like storing data into database and fetching data from database, passing data from server to client, etc. That means, we can transfer data among all component through data transfer messages.

### B. Message Format

There are two formats of messages.

**a) Text-Oriented Messages**

Text-oriented messages are readable character strings. There some advantages and disadvantages of text-oriented messages.

**Advantages:**

These messages are readable, flexible, easy to understand and monitor. They can be extended easily. They are easy to test.

**Disadvantages:**

These messages are human readable. That's why, easy to read by unauthorized person if messages are without encryption. These messages may become complex, and also hard to parse in code.

**b) Binary Messages**

Binary messages are blocks of structured data. These messages have also their own advantages and disadvantages.

**Advantages:**

Binary protocols are best way of structuring the data. They are suitable for large and complex data. Binary messages are very small as compared to text-oriented messages.

**Disadvantages:**

They are hard to read and test. To use this type of message, we should have a sound knowledge of data representation on hosts and network.

We will use both formats in our application protocol. When application (client) will request for the connection to server, server will respond to that request. Request will be in textual format. And data will be sent to component which is requesting data. And the data, which will be sent among the different components of Air Pollution Monitoring System, will be encrypted in binary format for security purposes, which will be decrypted later.

### C. Message Syntax

Message syntax describes the grammar of messages that how are they formed or how they will look like. When Client request for the connection, that connection request will contain IP address and the port of the server. The message contains section, header and body. The header section of the message contains message type, method type, format, status, content length. All these things make the message. And the body section will be empty if it is command message. In case of data transfer message, body section will be containing the data objects.

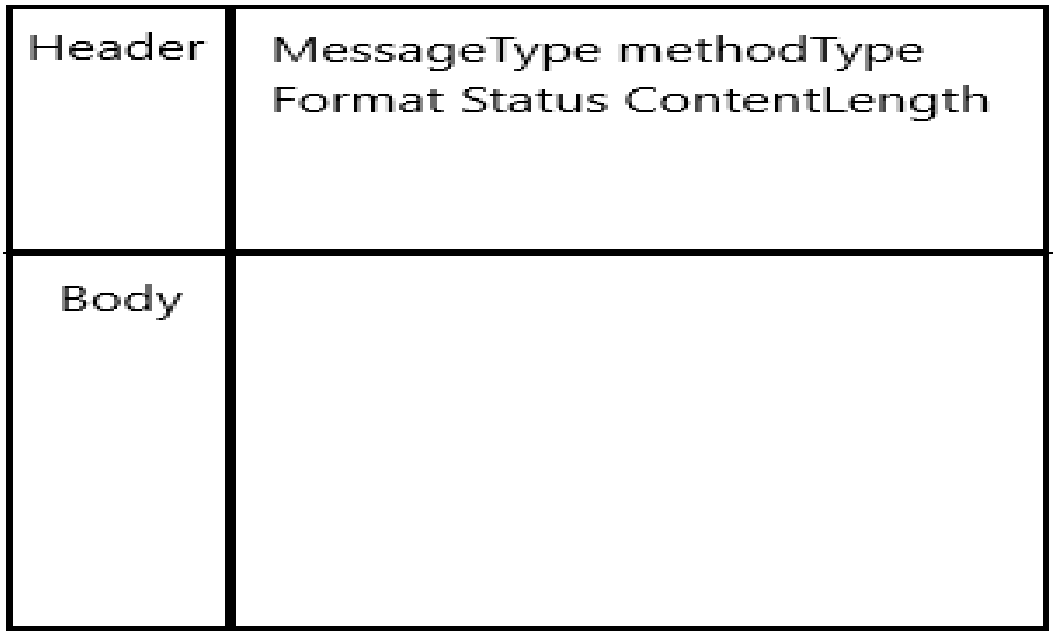


**Figure 4: Command Messages**

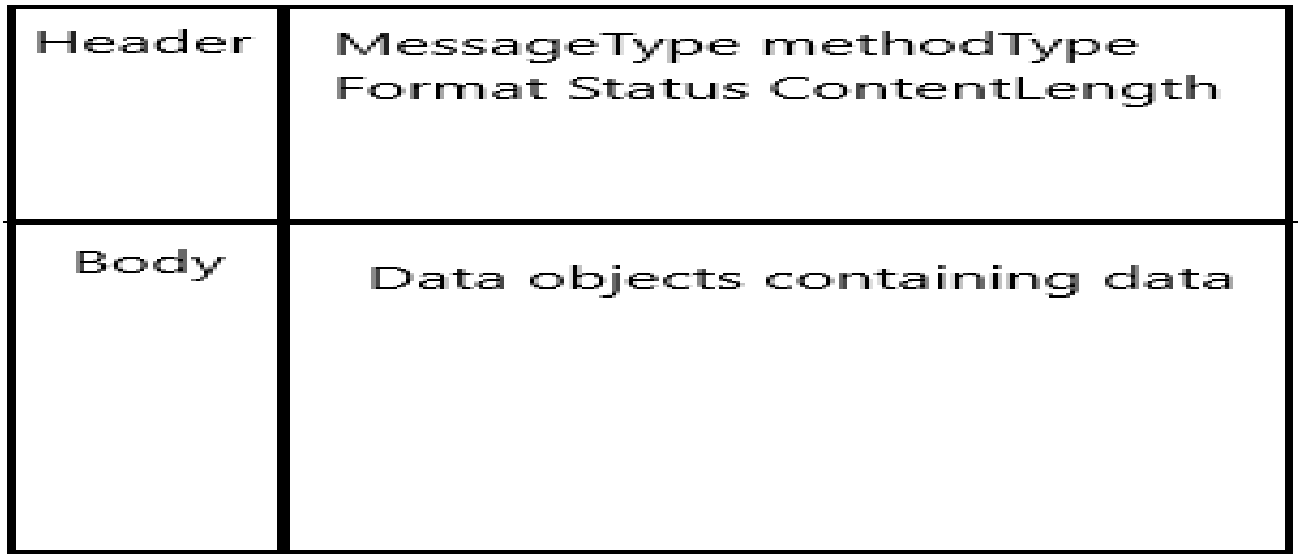


**Figure 5: Data Transfer Messages**

### D. Message Semantic

In my application, I will use two type of messages, command messages and data transfer messages. Command messages will contain empty bodies because they only give commands. Data messages will contain data in the body. We will use get and post methods. When we want to fetch some resources from server or any other component, get method will be used. And for sending data to server or any other component, we will use post method.

Format describes that whether it's binary message or textual message. Status shows the any failure, success and error messages. Encryption of the message will be done in binary numbers. And content length represents the length of data in body section.

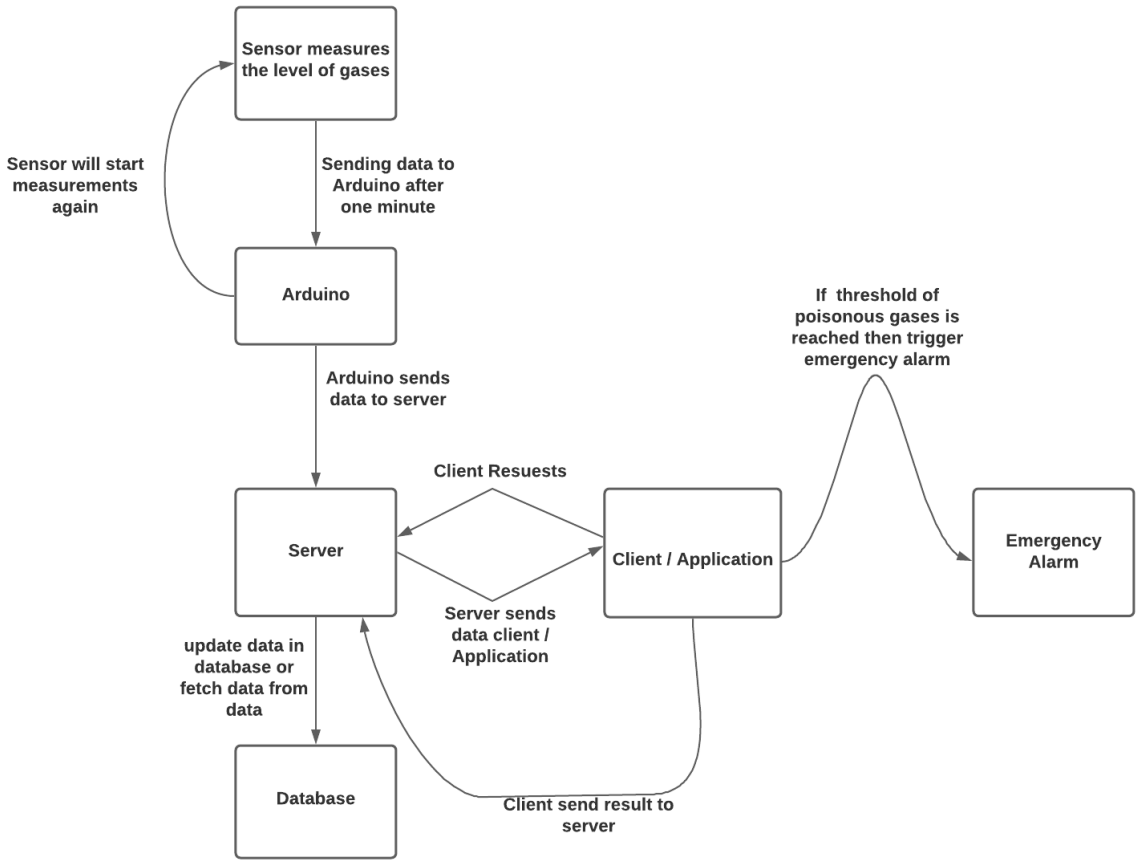


**Figure 6: Control Flow of Air Pollution Monitoring System**

## 11. Addition Information for the Developer

Here is some additional information for the developer which may help him to develop the Air pollution monitoring system.

- To code for this application, classes and algorithms for classification should be used.
- Developer should use the text editor for better experience vi, gedit, Notepad++, VSCode and sublime.
- IDEs are software for building applications that combines common developer tools into a single graphical user interface (GUI). Examples of IDEs include IntelliJ, and Visual Studio, NetBeans, Eclipse.
- Developer can use the database of own choice. SQL language should be useful for making our own database.
- Developer should use programming languages like python, C++ and Java.

## 12. Reference


- https://miro.medium.com/max/1500/1*9TkbU2pgT5QFpseeThO12g.png
- https://create.arduino.cc/projecthub/JANAK13/using-sensors-with-arduino-eab1ec


## 13. Appendix A – Glossary

**IoT:** Internet of Things

**IDE:** Integrated Development Environment

**NOx:** Nitrogen Oxide

**CO2:** Carbon di Oxide

**Threshold:** Specific limit for quantity of poisonous gases

**Throughput:** In general terms, throughput is the rate of production or the rate at which something is processed. When used in the context of communication networks, such as Ethernet or packet radio, throughput or network throughput is the rate of successful message delivery over a communication channel.

**Utilization:** "Utilization" is the percentage of a network's bandwidth that is currently being consumed by network traffic.