Software Requirement Specification

1. **Product/Project Overview**

In order to formulate a solution to this burning global issue we've come up with an idea that inculcates techniques of IoT/IoE (Internet of Things) and Data Analytics to predict and prevent air pollution substantially. The project aims at developing a device that will monitor the toxicity of the vehicular emission and assist the owner in getting the vehicle serviced timely to keep the emission levels low. Another objective of the project work is to analyze the air pollution levels at different geographical locations and to derive certain useful insights out of it that would help to reduce overall air pollution levels to a considerable extent.

1. **External Interface Requirements**

**2.1 User Interfaces**

User communication with the system is limited to one of the outcomes of the project which is "Pollution based Routing". An android app will be built that takes user registration and provides an interface for the user to enter the source and destination of his travel and using Google Maps the appropriate route prioritized on pollution levels will be shown and accordingly can be used for navigation.

**2.2 Hardware Interfaces**

The Hardware interfaces include the following:

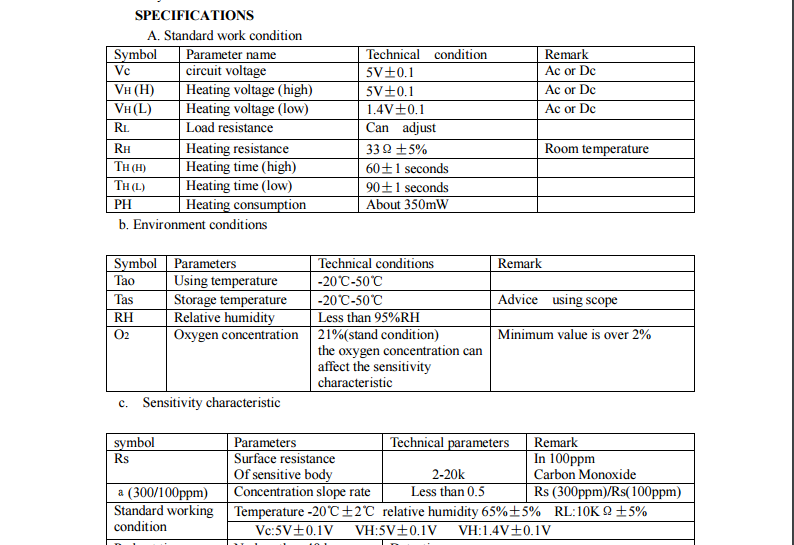
1. Arduino Uno 3:

|  |  |
| --- | --- |
| Microcontroller | [ATmega328P](http://www.atmel.com/Images/doc8161.pdf) |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limit) | 6-20V |
| Digital I/O Pins | 14 (of which 6 provide PWM output) |
| PWM Digital I/O Pins | 6 |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 20 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 32 KB (ATmega328P) of which 0.5 KB used by bootloader |
| SRAM | 2 KB (ATmega328P) |
| EEPROM | 1 KB (ATmega328P) |
| Clock Speed | 16 MHz |
| Length | 68.6 mm |
| Width | 53.4 mm |
| Weight | 25 g |
|  |  |

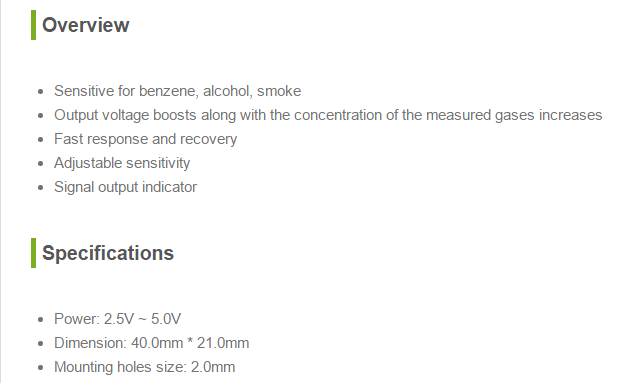
2. ESP8266 WiFi Module:

* + - * + 32-bit [RISC](https://en.wikipedia.org/wiki/Reduced_instruction_set_computing) CPU: [Tensilica](https://en.wikipedia.org/wiki/Tensilica" \o "Tensilica) Xtensa LX106 running at 80 MHz
        + 64 KiB of instruction RAM, 96 KiB of data RAM
        + External QSPI flash - 512 KiB to 4 MiB (up to 16MiB is supported)
        + [IEEE 802.11](https://en.wikipedia.org/wiki/IEEE_802.11) b/g/n [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi)
        + Integrated [TR switch](https://en.wikipedia.org/wiki/Duplexer#Transmit-receive_switch), [balun](https://en.wikipedia.org/wiki/Balun" \o "Balun), [LNA](https://en.wikipedia.org/wiki/Low-noise_amplifier), [power amplifier](https://en.wikipedia.org/wiki/RF_power_amplifier) and [matching network](https://en.wikipedia.org/wiki/Matching_network)
        + [WEP](https://en.wikipedia.org/wiki/Wired_Equivalent_Privacy) or [WPA/WPA2](https://en.wikipedia.org/wiki/Wi-Fi_Protected_Access) authentication, or open networks
        + 16 [GPIO](https://en.wikipedia.org/wiki/General-purpose_input/output) pins
        + [SPI](https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus), [I²C](https://en.wikipedia.org/wiki/I%C2%B2C),[I²S](https://en.wikipedia.org/wiki/I%C2%B2S) interfaces with DMA (sharing pins with GPIO)
        + [UART](https://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter) on dedicated pins, plus a transmit-only UART can be enabled on GPIO2.

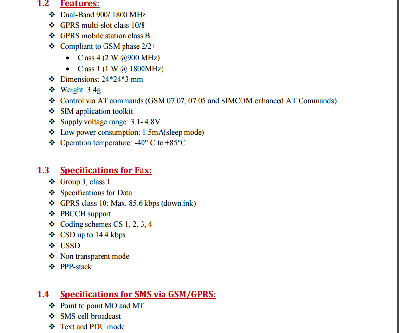
3. MQ-7 Gas Sensor:



4. MQ-135 Gas Sensor



5. GSM Board



**2.3 Software Interfaces**

* + 1. Arduino Platform:

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.  
This software can be used with any Arduino board.

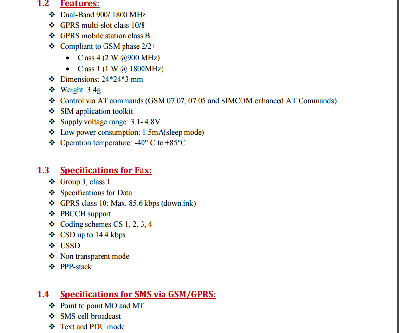
**2.4 Communication Interfaces**

1. ESP8266 WiFi Module:

* + - * + 32-bit [RISC](https://en.wikipedia.org/wiki/Reduced_instruction_set_computing) CPU: [Tensilica](https://en.wikipedia.org/wiki/Tensilica" \o "Tensilica) Xtensa LX106 running at 80 MHz
        + 64 KiB of instruction RAM, 96 KiB of data RAM
        + External QSPI flash - 512 KiB to 4 MiB (up to 16MiB is supported)
        + [IEEE 802.11](https://en.wikipedia.org/wiki/IEEE_802.11) b/g/n [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi)
        + Integrated [TR switch](https://en.wikipedia.org/wiki/Duplexer#Transmit-receive_switch), [balun](https://en.wikipedia.org/wiki/Balun" \o "Balun), [LNA](https://en.wikipedia.org/wiki/Low-noise_amplifier), [power amplifier](https://en.wikipedia.org/wiki/RF_power_amplifier) and [matching network](https://en.wikipedia.org/wiki/Matching_network)
        + [WEP](https://en.wikipedia.org/wiki/Wired_Equivalent_Privacy) or [WPA/WPA2](https://en.wikipedia.org/wiki/Wi-Fi_Protected_Access) authentication, or open networks
        + 16 [GPIO](https://en.wikipedia.org/wiki/General-purpose_input/output) pins
        + [SPI](https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus), [I²C](https://en.wikipedia.org/wiki/I%C2%B2C),[I²S](https://en.wikipedia.org/wiki/I%C2%B2S) interfaces with DMA (sharing pins with GPIO)
        + [UART](https://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter) on dedicated pins, plus a transmit-only UART can be enabled on GPIO2.

2. GSM Board

.



**3. Functional Requirements**

1. **Sensing the gaseous value for CO2 from the Exhaust emission**

The MQ7gas sensor senses the presence of CO2 from the emission and determines the level of content of CO2 gas in the emission.

1. **Sensing the gaseous value for NO2 from the Exhaust emission**

The MQ135 gas sensor senses the presence of NO2 from the emission and determines the level of content of NO2 gas in the emission.

1. **Sending the data to the main server using Wi-Fi**

Once the values are sensed from the emission, it has to be sent to the main server for processing on the value collected and hence requires a Wi-Fi Module.

1. **Processing the data obtained from sensors**

Once the data has been received from the sensors by the board and it has sent the data to the server, it has to be processed further. Based on the data received from the sensor, if the values cross the threshold for those gaseous levels, a first warning message will be sent about the condition of emission and the vehicle service period will be intimated to the owner.

1. **Second warning message to be sent as a reminder to the owner**

Once in the first message the owner is intimated about the period for servicing of his vehicle. Again a second reminder is sent to the owner if he still hasn’t got his vehicle serviced. This reminder is only sent if the owner has still not got his vehicle serviced even after first notification, which is generally sent in after half the time period allotted to him for getting his vehicle serviced.

1. **Choking mechanism**

Even after the second intimation, if the owner doesn’t responds and doesn’t get his vehicle serviced, our mechanism will choke the fuel pipe and hence until and unless he services his vehicle, he won’t be able to use it.

1. **Ranking the various areas of the city based on the air pollution levels. Graphical representations for various parts of the city based on various parameters should be provided.**

Based on the contamination level in the air for a particular region, the region will be provided with a rank which will eventually be used in deciding the amount of time provided to the vehicle owners when their vehicle’s emissions are exceeding threshold. Various graphical representations for the same are to be provided based on various parameters like locality or type of area etc.

1. **Predictions should be provided based on the current levels of the pollution and certain precautionary measures to be suggested.**

Based on the available data, certain predictions will be made informing people about the adverse effects of the rising air pollution. These predictions will be basically about what all can happen if the same contamination rate of air continues for certain period of time. Along with such predictions, precautionary measures can be suggested to the general public and to spread the awareness about it. We call it “Tip of the week” and "Tip of the day". The former will alert the people about uneven increase of the pollutants in the atmosphere of the area over a week and provides air pollution preventive measures. The latter is a daily notification sent to all the registered vehicle users which contains information about the highly polluted area and asking the residents/passers of'/from that area not to use their vehicles unless there's necessity.

1. **Routing of vehicles based on Air pollution at different locations.**

The route which a vehicle takes to reach a destination from a source is usually decided based on the distance, traffic etc but the project aims at considering another important parameter: Air pollution levels, to route. The vehicles can take different routes from the usual so as to account as least as possible for the air pollution in the city.

**4. Software System Attributes**

**4.1 Reliability**

* This system is available all time and mainly the reliability of the system depends on the sensors and the board. As the lifespan of board and sensors are more, the reliability factor is also more.

**4.2 Availability**

* It has to be available all the time 24x7 , provided users have Internet to access the web page and also for sensors to push data to the server.

**4.3 Security**

* The catalogue is made available only to the admin and the users. A separate

Login is made available to the user through which he can access information about his vehicle’s air pollution level and also about various preventive measures, which requires authentication. Also the routing service will be accessed by him on that particular app. Thus the data is made secure by making it available only to the concerned people. Any modification to the database shall be synchronized and be done only by the database admin.

**4.4 Portability**

* The system developed is portable, as small and light components with wireless connections are used .It can run over different platforms.

**4.5 Maintainability**

* This system is easily maintainable as it has fewer components and its design is not complicated. As the lifespan of sensors and the board is high, so it requires less maintenance. It is also well documented thus making it easy to use.

**4.6 Performance**

* It depends on computer hardware specifications like RAM, GPU etc. Since we are handling huge data, it should be processed in parallel. The system shall give responses in 1 second after collecting the emitted gases using sensors. The user-interface screen shall respond within 5 seconds. The System supports 1 vehicle at a time.

5. **Performance Requirements**

* If any excess of emission of those gases are detected, the owner should get the intimation of the excess levels and time period for servicing his vehicle within next 1 minute.
* Every morning by 8 a.m., “Tip of the day” should be sent to the owners.
* Second intimation should be sent to the owner in the halfway of the time period for servicing (Only if he has not got his bike serviced).
* The data analysis of the air pollution data has to be closely monitored and every month, the threshold should be updated.
* Routing should be highly responsive and suggest best routes possible based on air pollution levels.

6. **Database Requirement**

* A table has to be maintained about the air pollution level and has to be continuously updated as and when data is received from authoritative bodies.
* There are two separate tables i.e. one will contain owner-specific data(owner’s personal information and vehicular emission levels and servicing records.) and other will have data about the city’s pollution level and will be updated at regular intervals.

**7. Design Constraints**

1. **Hard drive space:** The amount of space an application needs for storage and execution purpose is major design limitation as Arduino Uno 3 uses a Flash Memory for storage.
2. **Application memory Usage:** The amount of memory space needed for an application to run is an important limitation in the design as Arduino Uno 3 has only 256MB RAM and it needs to effectively utilize it.
3. **Budget:** The amount of money that can be spent in the overall development of the project restricts the design of the system.
4. **Application Quality:** If the quality of the application is high then it means that the design is highly effective. Thus the quality of the application is an important deciding factor in the design of an application.

8. **Other Requirements(if any)**

Knowledge of Web programming, Python programming, R programming and other Data Anaytics Techniques, C programming with Arduino and connection of sensors to Arduino Uno 3 Board.