# Citizen Air Quality Sensor

Joshua Egwuatu, Kaleb Irwin, Cameron Smith, Alejandro Moore, Marvo Odds

Computer and Electrical Engineering Department at Tennessee Technological University

115 W. Tenth St., Cookeville, TN 38505

Jiegwuatu42@tntech.edu, kjirwin42@tntech.edu, casmith51@tntech.edu, aamoore42@tntech.edu, modds42@tntech.edu

## I. Introduction

### A. Problem Definition

The goal of this project is a replicable device that accepts and operates multiple air quality sensors independently with a selection of interchangeable sensor options for local data collection. Air quality is vital to maintaining life on Earth [1]. Poor air quality can result in a slew of health problems such as heart disease and lung cancer [2]. In the modern age, humanity pushes the levels of pollution in the world to unprecedented levels as compared to the pre-industrial era [3]. This pollution affects the very air that all living things rely on to survive. This perspective on the negative repercussions of polluting the air and Earth is what led to this team's proposition of a device that allows everyday citizens to see what pollutants blight their communities.

There are already services that exist to alleviate this problem, however, the services do not fulfill the solution to a high degree. The biggest of these services would be the Air Quality Index (AQI) provided by the United States Environmental Protection Agency [4]. The AQI takes in data from sensors placed systematically all across the country and in highly populated areas [5]. This data is then aggregated together to fill in any regions that were not explicitly sampled from a sensor directly [5]. The issue derives from that last sentence. Because it is not reasonable or cost-effective for these services to place sensors everywhere, the readings that are received from these sensors may not be accurate in any given place. For a community wanting a detailed view into the air quality of their chosen area, they would require more concrete sources of data. It is here where the team's proposed device comes in to provide an easy-to-use, solar-powered, easily replicable, and low-cost modular system that uses multiple sensors to gather data on different pollutants.

This document is important in the design process because it will present general solutions based upon the constraints and requirements previously stated in the project proposal. The document will also give a description of each subsystem and how each works. Giving a high-level look at the solutions and proposed diagram will allow for the team to break down the overall problem. In doing this, the idea is to solve the problem in smaller and easier to comprehend steps.

### B. Constraints

List of shall statements (format [SS#]) that are referred later:

1. Shall have a selection of tested sensors to cover a range of pollutants, particulates, air composites, and other qualities. Common qualities to measure are carbon monoxide, lead, nitrogen oxides, ozone, particulate matter, and sulfur dioxide. Other qualities may be selected in the future, but these are definite qualities to look for. These sensors should work independently from each other.
2. Shall be easily replicable. The device and additional components shall be mostly, if not entirely, off-the-shelf purchases. This should lower the technical knowledge required to construct the device tremendously. The software should be available for free online on GitHub. Software licenses will be included depending on which software is chosen and used.
3. Shall be relatively cost-effective. The device, additional components, and software should remain under \$300. The cost of sensors is not taken into account as the sensors are used on a case-by-case basis.
4. Shall be protected against the environment. The standard to be followed for water resistance will be IPX4, protecting from all directions against the rain. The standard to be followed for dust and small objects will be IP3X which states protection against hazardous parts, thick wires, or solids of sizes greater than 2.5 mm. The device will operate in a temperature range of 10°F to 90°F. This covers a wide and likely temperature range seen in the U.S. The device will also be staked into the ground and resist wind speeds up to 74 mph.
5. Shall be modular. In this instance, modularity refers to the ability to change to a multitude of previously tested and compatible sensors without the need to change any or at most 2 select settings. The change of

sensor should be seamless and handled by the board and software.

6. Shall utilize sensors that are interchangeable. The sensors will have multiple kinds of Input/Output (I/O) ports, so a list will be made that acknowledges all I/O ports of sensors to be approved and tested. The I/O ports that are shared among a group of sensors should be able to operate seamlessly if swapped out from the same type of I/O port for another sensor in the group using that I/O port. Sensors that do not share this I/O port will have a port of its own grouping that has this same functionality within that group.
7. Shall be able to operate with at least 3 sensors sampling independently of each other. Sensors will take samples at set intervals and be read and stored on the device periodically. Samples may be received and outputted one at a time.
8. Shall have easily accessible data. The device should be able to take data from storage and send it to a website to be read by the user remotely. p If the user wishes, he/she may also read data from the device directly. Formatting will be basic and leave room for improvement for future development.
9. Shall have a set of security measures to protect the device and data. The data shall be encrypted with an undisclosed algorithm. This is during transmission encryption and encryption during the storage of the data on the board. The chassis will be made of a material to be reasonably impact resistant at the standard of IK08. This standard rates the device at 5 Joules of impact. A global positioning system will also be added to aid to reclaiming during loss or theft.
10. Shall have an independent power source. The device should be able to be planted somewhere and operate with an onboard power supply. This power supply may include two major components, a solar cell pack and or a rechargeable battery pack. The battery pack should be able to maintain the device for at least 24 hours without recharging. If the solar cell pack is included, the solar cell should allow for charging of the battery pack and prolong the life expectancy.

### C. *Ethics and Considerations*

This design team will follow National Electrical Code (NEC) and National Fire Protection Agency (NFPA 70) procedures. The team will not break any laws established by the United States of America Government. All patents, licenses, and copyrights will be used under legal fair use with credit when appropriate. This device will be designed to prevent risk and harm to the users, communities, and environment. The team will follow IEEE code of Ethics and common electrical concepts/standards: resistor color codes, design schematics, communication between devices, etc.

Broader considerations are the potential effects that the device may have. Given the nature of the device, the only harm that may be caused by the device would be flammable ignition. The risk of wildfire will be minimized during design. The design does not include any hazardous materials save the optional battery. If the battery were to get wet or damaged, there could be a chance or fire. The design will be made to resist water and small impacts to help account for this. There are also considerations to further improvement on the project made by future teams. This project could allow citizens to see if the air quality in their homes, work places, and schools is safe for adults and children alike. If other groups see the work and use it as a foundation for new devices, that would show enough evidence to consider this project successful and impactful.

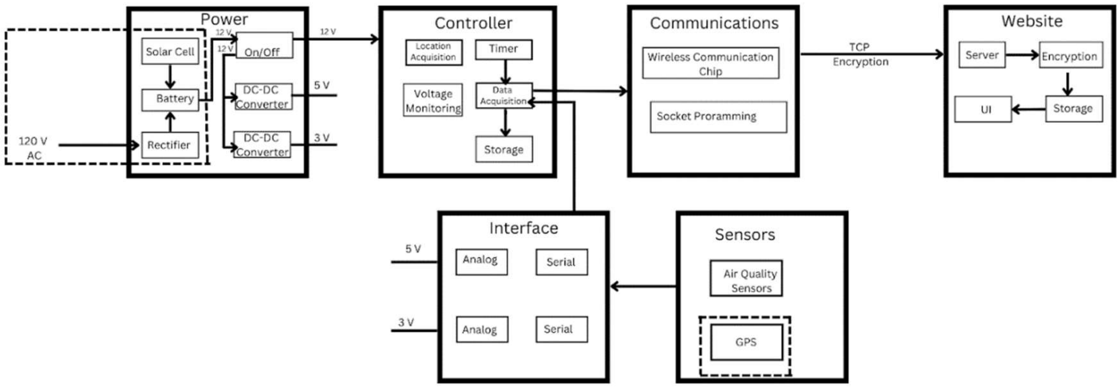


*Figure 1. Block Diagram*

## II. POWER SYSTEMS

The power system will be designed with modularity in mind. The user will be able to select from three different configurations depending on their operating requirements and budget [SS11]. The first option will power the device using 120 volts straight from the wall. The second configuration will use a solar panel and a battery for continuous operation. The third layout will use both the solar panel and the AC input in order to charge the battery that will provide power to the rest of the system. Regardless of the chosen configuration, the output will be sent to the microcontroller and DC-DC converters through an on/off switch. The converters will be used to power various sensors with different voltage requirements [SS7].

The choice to use the DC-DC converters and power the sensors from the source instead of the microcontroller was made due to current (Ampage) considerations. Depending on the user's selection of sensors, operating them simultaneously may result in higher current draw than the selected microcontroller can supply. Each configuration will be tested with LTSpice using the available data given on relevant datasheets. The team will look at the output voltages as well as the current capabilities of all components and ensure all values remain in specifications. The input and output of each component will also be tested experimentally in order to double check the simulation before connecting the entire system.

## III. MICROCONTROLLER SYSTEM

The microcontroller such as an Arduino Mega REV 3 with 54 digital input/output pins, 16 analog inputs and will be powered by a battery. Using a microcontroller like the Mega allows for more sensors to be added and accounts for types of sensors that might take up more pins [SS5]. The timer will set a determined time interval for the Arduino to wait before acquiring and sending data gathered from the sensors to account for unstable data given by the sensors due to warmup and cooldown times. The data acquisition system will receive input from the interface and output data to two places: the storage within the arduino and to the website database [SS9]. Location acquisition will set up the Arduino to be able to interface with an outside GPS module and allow the user to see the location of the data readings. The voltage monitoring system will allow the user to see the charge status of the battery [SS10].

The constraints for the microcontroller consists of; it needs to have enough pins to be able to support multiple sensors and the data acquired by the microcontroller needs to be easily accessible [SS8].

## IV. INTERFACE AND PORT CONTROLS

The interface is a specified area on the chassis that is available to the user for their sensor selection. The targeted area will be able to house at least 3 different sensors at a time and will send the received data to the microcontroller [SS7]. It is important that the interface be able to take in multiple types of sensors that detect a specified pollutant within the list of common qualities [SS1].

When the device turns on, the input ports for the sensors will be activated with various voltages coming from the power subsystem once plugged in by the user. From research, there are several sensors that require slightly different amounts of input voltage. Therefore, using the DC-DC converters from the power block, the team will connect wires to the input ports within the interface.

The constraints for the interface will be implemented into the design process. Sensors that do not meet the specified input voltage range at the interface ports will not be approved for the user to integrate for production. The sensors to be tested and verified will output to the microcontroller analog or serial data that is detected. Any sensor that does not output either analog or serial shall not be used or tested within the design process. The GPS within the sensor subsystem is in actuality a sensor but will not be connected to the interface. The specified sensor shall be connected to the microcontroller and also will behave as a modular design. Thus, the user has the option to install the GPS or not and the device will function similarly.

## V. COMMUNICATIONS

The microcontroller board, by default, will not be able to access the Internet. Before the data can be sent to and stored on the server, it must travel to an attached communication chip that allows for the data to traverse the Internet. This chip will connect to the microcontroller and behave as an antenna of sorts. There are several manufactured chips that could fulfill this role. For example, Arduino provides what they call a "GSM Shield" that has guaranteed compatibility with the Arduino Mega REV 3 [6].

### *A. Wireless Communication Chip*

A chip, such as the GSM Shield (Global System for Mobile communication), features a SIM (Subscriber Identity Module) card that can be assigned an identifying set of credentials and location used by cell towers [6]. A cellular service provider must assign the SIM card information to allow access to the cell towers [6]. With a connection established, the microcontroller will be able to send and receive data as if it were a common cell phone. This enables a wireless connection and a method for sending data to and from the server [SS8]. With modern phones, the cell towers are easily able to carry information to servers hosting websites [7]. This will be perfect for the purposes of this project, but this does come with a third-party service fee that varies depending on the cellular service provider that is chosen.

### *B. Socket Programming*

Socket programming refers to a method of communication where a dedicated port is agreed upon between two machines for the purposes of data transferal [8]. When two machines are attempting to communicate, there must be some framework or ground level that can be used. Socket programming fulfills this by giving the two machines a near guaranteed way to listen and talk to each other once the agreement is made. "Agreement" refers to the final decision on which port is available for other machines to utilize. These ports are to be used only by one other machine at a time and cannot be accessed by other machines without knowing key information about the connection [8] [SS9]. Modern day devices come with many ports and will more than satisfy the needs of the microcontroller [8]. This project will use this method as it is quite common and helps in the aim to maintain the "easy-to-use" nature of the design [SS2]. There are many articles, guides, and documents regarding socket programming available to the public.

## VI. WEB AND WIRELESS CONNECTIONS

Wireless acquisition of the data adds an undeniably large boost to the "quality of life" when using the device. Having to physically go back to each device that was deployed for the data would be very tedious as opposed to remotely gathering the data and moving the devices when ready. The components of the "Web" block include: server, storage, user interface (UI), and encryption.

### *A. Server*

The server's job is to accept data from the device and store it for remote access from the user. To do this, the server will need a listening port that waits for a signal from the devices. The server will take in the data and then encrypt it. The reason for encrypting the data is for security purposes [SS9]. The Internet is free to everyone, so the threat of hackers or cyber-attacks are always present [9]. The encrypted data is then placed into storage on whatever machine is hosting the server. This is said as the option for privately hosting a server would be available in the future. For users who do not opt for a private server, a default server will be available [SS3].

The default server has two possible solutions: a website hosted by a third party service or a locally hosted server machine. Hosting the server from a third party would be easier and allow for quick setup. Doing this solution trades the cost of buying and maintaining a local machine for rent fees. Writing and building a server from scratch would allow for total control of the server's protocols, be cheaper in the long run, and secure the future of the project as it would always be available whereas a third party could stop providing the service. This option would cost more upfront and potentially require maintenance from time to time. As a proof of concept, a simple server may be written and demonstrated, however, long-term solutions will be one of the two previously mentioned solutions.

## B. *Storage*

The data will need to be stored on the server host machine. This means that it will be a requirement for the server host machine to possess adequate storage space for the server code to store all of the collected data. In terms of hardware, hard disk drives and solid-state drives would work optimally

## C. *User Interface*

The UI will be what the user sees when attempting to read the data that was collected from the device. It will be bare bones. The purpose of the UI on the server is simply to get the data into the hands of the user in an easy manner. The data should already be formatted in a way that makes sense, so the user will only need to see what is being presented and do whatever is wished with it. The format will likely be a similar setup to Google Sheets or Microsoft Excel with generic names for subjects. It will be up to the user to interpret the data. The UI itself will be a portion of code within the server that provides a palatable visual for the user to overview the data.

\

## REFERENCES


[1] Kim Rutledge, Melissa McDaniel, Santani Teng, Hilary Hall, Tara Ramroop, Erin Sprout, Jeff Hunt, Diane Boudreau, Hilary Costa, National Geographic, "Air Pollution", https://education.nationalgeographic.org/resource/air-pollution. (Accessed on 9-21-22).

[2] Minnesota Pollution Control Agency, "Air Quality and Health," https://www.pca.state.mn.us/air-water-land-climate/air-quality-and-health. (Accessed on 9-21-22).

[3] United States Environmental Protection Agency, "History Of Air Pollution," https://www.epa.gov/air-research/history-air-pollution. (Accessed on 10-16-22).

[4] Air Quality Life Index, Constructive Inc, "Pollution Facts", https://aqli.epic.uchicago.edu/pollution-facts/. (Accessed on 9-22-22).

[5] AirNow, Home of the U.S. Air Quality Index, "Air Quality Index (AQI) Basics", https://www.airnow.gov/aqi/aqi-basics/. (Accessed on 9-21-22).

[6] Arduino, Electronics and Software Manufacturers, "Getting Started with the Arduino GSM Shield," https://docs.arduino.cc/retired/getting-started-guides/ArduinoGSMShield. Revised on 10-5-12. (Accessed on 10-25-22).

[7] Batt, Ashutosh, Electrical Engineer and Author, "How Internet Works on Mobile Devices," https://www.engineersgarage.com/how-internet-works-on-mobile-devices/. (Accessed on 10-25-22).

[8] International Business Machine Corporation (IBM), "Socket Programming," https://www.ibm.com/docs/en/i/7.1?topic=communications-socket-programming. (Accessed on 9-19-25).

[9] DeVry University, "A Constant Threat: The Impact of Cyber Hacks," https://www.devry.edu/blog/threat-cyber-security-hacking-infographic.html. (Accessed on 9-19-22)

DESIGN PHASE 1 TIMELINE: