Air Quality Monitoring System Based on Sensor Networks

ECE445 Design Document

Team member: Yuhui Xu(yuhuixu2) Yuhang Ren(yuhangr2) Ke Liu(keliu5)

Advisor: Simon Hu

1. **Introduction**
   1. **Problem and Solution Overview:**

Even though we have lived for about 4 years in Haining campus, we are still not used to some strange smell that we can sometimes sense in campus, especially when it is in spring. So, it is very necessary to monitor the air quality in our campus and have an alarm if there are air pollutants. And these monitoring data can also be used by meteorologists or civil engineers to analyze weather pattern in this area. However, it is not so easy to get a precise measurement and prediction on the air quality, due to the quite large area of our campus and lots of environmental factors that may cause negative effects on the measurement

In general, we are going to design an air quality monitoring system that can analyze environment information and calculate optimized places to set up sensors. Once we finish setting up sensors, our system can obtain air quality data in several corners of our campus, and automatically create an air quality map that covers the whole campus based on airflow models. And our students can find this map in the website and get information about the real-time air quality.

**1.2 High-level Requirements List:**

* In the outdoor, the self-designed device can measure the concentration of the following matter: PM1.0, PM2.5, PM10, CO, NO2, H2, and NH3, as well as basic environment information such as temperature and humidity. The values would be compared with the values measured by a more professional device located in the laboratory in Haining Campus.
* This device should run continuously for at least 7 days without changing its battery and we can see the battery storage from our website.
* Users can use our airflow simulation framework to decide the optimized locations of setting sensors if he knows basic information in the area that need to be measured. And he can also get rea-time air condition map and future prediction once he sets up the sensors
* The website can show the real-time air quality map and the data in the past.

1. **Design**
   1. **Block Diagram**

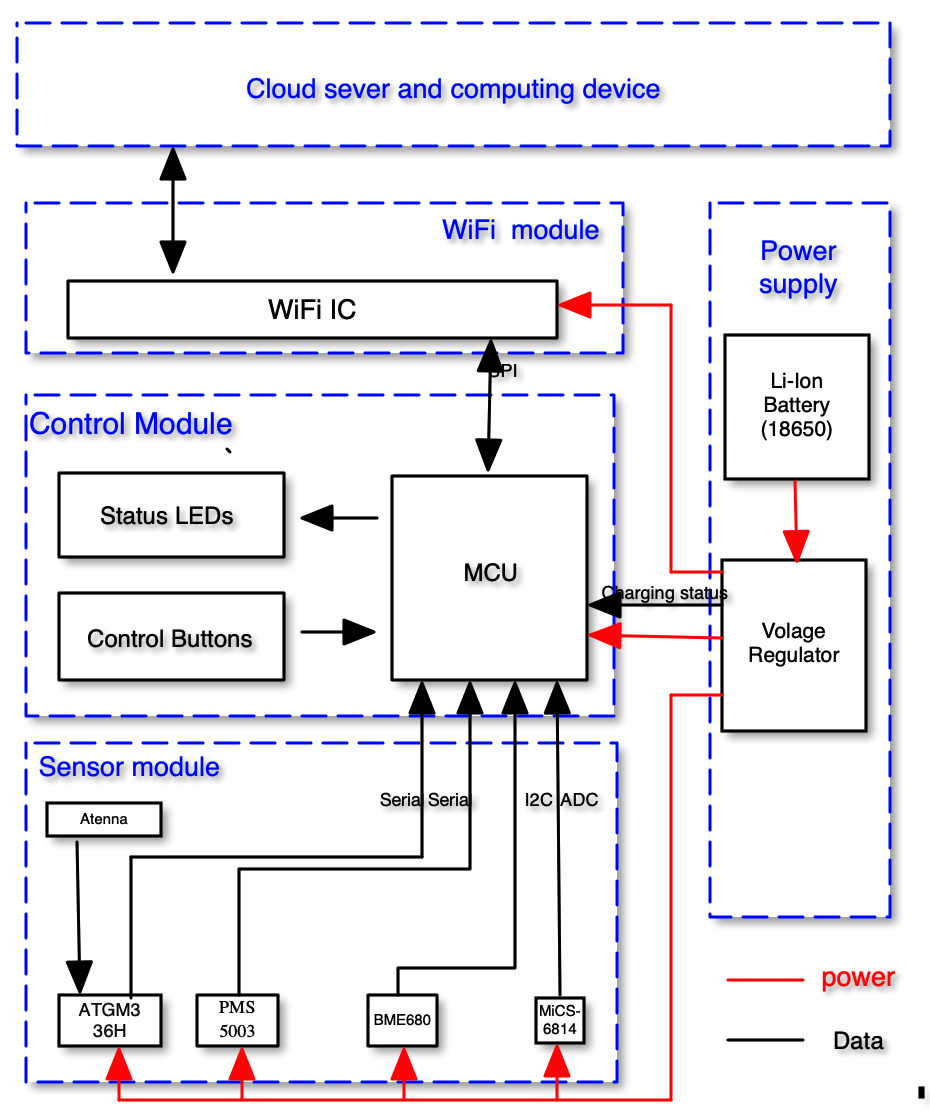


Figure 1 block diagram

* 1. **Physical Design (General Sketch)**

**图片包含 游戏机, 体育, 画, 桌子

描述已自动生成**

Figure 2 Outside View

**图片包含 游戏机

描述已自动生成**

Figure 3 Inside View

* 1. **Subsystem**
     1. **Power Supply Module**

Electrical devices in the circuit depend on power supplies. This includes a power source (18650 batteries) and a voltage regulator to create a 5V source as well as the 3.3V pin provided by MCU. This module would be connected with other module by wires.

|  |  |
| --- | --- |
| **Requirement** | **Verification** |
| The Voltage regulator can provide 3.3V energy source for MCU for more than 7 days. | 1. Charge the battery to full status 2. Keep our device running and check its battery status continuously until 7 days |
| Temperatures of components must be sustained below 125°C | 1. Running our device with all sensors are working. 2. Measure the temperature with IR thermometer to ensure all components do not reach temperatures greater than 125°C |

Table 1: RV for power supply module

* + 1. **Sensor Module**

This module contains all sensors that generate air quality data as well as other necessary data for our system, e.g. temperature, humidity, pressure and location. All these sensors would be connected with our MCU. The detailed pin assignment can be found in Figure 5, schematics. The interfaces that our sensors use are:

|  |  |
| --- | --- |
| Sensor | Interface |
| ATGM336H | Serial |
| PMS5003 | Serial |
| BME680 | I2C |
| MiCS6814 | Analog |

Table 2: Interfaces of sensors

|  |  |
| --- | --- |
| **Requirement** | **Verification** |
| All these sensors should get correct results within 15% accuracy | 1. Measure data in three different place, three times for each places 2. Use air quality egg (provided by Prof. Hu, we use it as reference) to get reference data in the same three places. 3. Compare the result from our sensors and the result from air quality egg. |
| All these sensors should get correct result with the temperature range of [-10°C, 40°C] | 1. Use heater and refrigerator to generate environment with temperature at about -10°C, and 40°C 2. Use our device and air quality egg to obtain data 3. Compare data from these two devices. |
| All these sensors should measure data in the frequency we specify (we assume once per hour but this frequency can be determined based on real case) | 1. Change the measure frequency to once per hour, twice per hour and once every two hours. 2. Check whether our device would upload data as the frequency we specify. |

Table 3: RV for sensor module

* + 1. **Cloud server and Computing device**

We use InfluxDB to manage and store data in our cloud server. The cloud computing device would analyze air quality data as well as environmental information so that it would automatically generate an air quality map.

The description of map generating algorithm is as follows: we would assume air can only flow through those space without tall building if necessary, which means air majorly flow on the road and street. Then we can easily monitor air quality on the whole street by using interpolation method if we had measured data of both ends of the street. In the first step, we would consider the cross point of streets as our optimized place.

A flow chart can be found in the following:

手机屏幕截图

描述已自动生成

Figure 4: algorithm flow chart

|  |  |
| --- | --- |
| **Requirement** | **Verification** |
| Algorithm-selected detection points generally provide air quality maps with higher accuracy than random points | 1. Set up devices in the output points and generate air quality map, called original map. 2. Each time change location of one random point and generate air quality maps. Repeat for three times. Call these three maps random maps. 3. Compare the average accuracy(accuracy is defined as the difference between the data generated by air quality map and the data measured by air quality egg) of original map and random maps. The average accuracy of original map should not be lower than the random maps. |
| Generate air quality map and for most of location, the data from our air quality map should be 20% accuracy within the reference result | 1. Randomly choose 10 sample points that do not belong to those points that have measuring devices. 2. Use air quality egg to measure air quality data in these 10 sample points. 3. Compare the data from our generated air quality map with the data from air quality map. The difference should be within 20% |
| Can view and download data from our server | Output sensor data from local devices, and verify if the sever shows the same data in real time |

Table 4: RV for software module

* 1. **Device schematics**

****

Figure 5: Schematics

* 1. **Tolerance Analysis**

The first thing we need to guarantee in this project is that our battery can provide enough energy for our device to work for at least 7 days. In this part, we would use calculation to confirm this requirement can be met based on the hardware we choose. The general idea is that we would let our device working at the frequency we specify. For example, we may want to update our map every hour, which means that our devices should send air quality data to our cloud server once per hour. Then in each hour, we would let our device be in the working mode for 1 minutes, which, by our measurement, should be enough for all sensor finish their work. And then it would be in silence mode for the last 59 minutes and wait until next cycle.

To achieve this, we firstly check data sheet of components. For sensor PMS5003, BME680 and Esp01, they all have built-in working mode and silence mode, so we can directly get its working mode current and silence mode current. For sensor MiCS-6814 and ATGM336H, they would be working only if we supply enough voltage to them, so we would just supply voltage to them for one minute each cycle. For the silence mode, they consume almost no energy. As for our stm32, from its datasheet [1], we choose its working mode with all peripherals working as our working mode and its sleeping mode with all peripherals sleeping as our silence mode. A summary about all component can be viewed in table3.

|  |  |  |
| --- | --- | --- |
| Component name | Working mode current | Silence mode current |
| PMS5003 | **≤100mA** | **≤200uA** |
| BME680 Humidity sensor | **≤450uA** | **≤1uA** |
| BME680 Pressure sensor | **≤849uA** | **≤1uA** |
| BME680 Temperature sensor | **350uA** | **≤1uA** |
| BME680 Gas sensor | **12 mA** | **≤1uA** |
| Esp01 | **≤170mA** | **≤15mA** |
| MiCS-6814 | **88mA** | **N/A** |
| ATGM336H | **≤25mA** | **N/A** |
| STM32 | **≤50mA** | **≤7.5mA** |

Table5: power consumption summary

Then if we specify our measurement frequency to be once per hour, we can get our total energy consumption:

If we use two 18650 batteries with 3350mAH for each, the total time we can use are:

As a result, we show our hardware design can achieve the goal that the device can run for at least 7 days without changing its battery. Here, all current we use are maximum current. But in real case, it should be smaller than the data we present in the above table, which means our hardware can definitely tolerate some small variance about the average value we use to calculate.

And, if we need to update our map in a higher frequency, we need to increase the number of battery or decrease the energy consumption of some components. As we can see in Table, Esp01 and PMS5003 consume a lot of energy.

1. **Cost and Schedule**
   1. **Cost Analysis**
      1. **Unit cost for measure device:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Description** | **value** | **Number** | **Unit price** |
| **Stm32F103C8T6** | MCU | N/A | 1 | ¥11.3 |
| **ESP01** | Wi-Fi module | N/A | 1 | ¥16.95 |
| **ATGM336H** | GPS module | N/A | 1 | ¥39.30 |
| **PMS5003** | PM1,PM2.5,PM10 | N/A | 1 | ¥70 |
| **BME680** | T,P,humidity | N/A | 1 | ¥55 |
| **MiCS-6814** | CO,NO2,NH3 | N/A | 1 | ¥149.6 |
| **Molex 1.25mm 8p connector** | N/A | N/A | 1 | ¥1.08 |
| **18650 battery** | N/A | 3350mAH | 2 | ¥45.9 |
| **Battery box** | N/A | N/A | 1 | ¥2.67 |
| **Resistor** | N/A | 56K | 3 | ¥0.02 |

Table4: device component cost

* + 1. **Labor cost**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Hourly Rate** | **Total Hours Invested** | **Total** |
| **Yuhui Xu** | ¥150 | 100 | ¥15000 |
| **Yuhang Ren** | ¥150 | 100 | ¥15000 |
| **Ke Liu** | ¥150 | 100 | ¥15000 |

Table6: labor cost

* + 1. **Total cost:**

From our design, our system should consist of multiple measuring device that can cover the whole area. And the number of device would be determined by users and the environment, so here we just donate it as n.

**437.76\*n+45000**

* 1. **Schedule**

|  |  |  |
| --- | --- | --- |
| **Week** | **Goals** | **Responsibility** |
| **3.30** | Sensors function testing | Yuhui Xu |
| First-version PCB drawing | Yuhang Ren |
| Document Design writing | Ke Liu |
| **4.6** | Hardware assembling | Yuhui Xu |
| PCB revise | Yuhang Ren |
| Hardware testing | Ke Liu |
| **4.13** | Display webpage design | Yuhui Xu |
| Test case writing | Yuhang Ren |
| Cloud sever preparation | Ke Liu |
| **4.20** | Map-generation algorithm code writing | Yuhui Xu |
| Environmental information gather | Yuhang Ren |
| Sever connection test | Ke Liu |
| **4.27** | Map-generation algorithm code writing II | Yuhui Xu |
| PCB re-design if needed | Yuhang Ren |
| Map-generation algorithm code testing | Ke Liu |
| **5.4** | Optimized-location algorithm code writing | Yuhui Xu |
| PCB shop contact and produce PCB | Yuhang Ren |
| Optimized-location algorithm code testing | Ke Liu |
| **5.11** | Bug fixing and demo preparation | All |
| **5.18** | Demo Week | All |

Table 7: Weekly schedule

1. **Ethics and Safety**

There are several potential safety and Ethics hazards in our project. Here we would like to discuss about power supply safety hazards, environmental safety hazards as well as code of ethics.

Firstly, for power supply safety hazards, we need to understand that safety is not absolute or intrinsic[2]. We can only reduce the probability of a safety event, minimize the level/severity of that event and limit the consequences of the event. For Lithium-ion hazard, it usually happen due to overcharging, overtemperature and mechanical abuse. For overcharging, it would require very high voltage and this can be easily prevented by using our particular battery chargers and battery regulator. These two module would convert the voltage to appropriate one. For overtemperature, we have several temperature sensors in our device. We would build in an algorithm to stop power supply once we detect extremely high temperature, and also set a message to our users. And for mechanical abuse, it is hard to detect this automatically, so we would pay attention to the battery when we are installing the devices.

Then for environment safety hazards, it means that our device may be damaged by rain, snow, dust because it is a outdoor device. However, the sensors should also be exposed to air so that they can measure air quality data in the air. In this case, we would design an IP55 shell or box to hold our PCB and also leave some holes so that our sensors can be exposed to air.

For code of ethics, we promise we would following IEEE Code of Ethics[3]. We promise “to be honest and realistic in stating claims or estimates based on available data.” We promise “to improve the understanding of technology; its appropriate application, and potential consequences”

1. **Reference:**

[1]. STMICROELECTRONICS. “STM32F103C8T6 Datasheet”, 2017. [Online] Available: <https://html.alldatasheet.com/html-pdf/201596/STMICROELECTRONICS/STM32F103C8T6/1950/1/STM32F103C8T6.html> [Accessed: 01-04-2020]

[2] Jim McDowall. “A Guide to Lithium-Ion Battery Safety”.2014. [online] Available: <https://cmte.ieee.org/pes-essb/wp-content/uploads/sites/43/2016/06/2015-WM-PN-A-Guide-to-Lithium-ion-safety-Jim-McDowall.pdf>[Accessed :01-04-2020]

[3]Ieee.org. "IEEE Code of Ethics", 2016. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html> [Accessed: 01-04-2020]