ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Final Report Draft for ECE 445 Vehicular Edge Computing System

---

Team #38

SHAOHUA SUN
(shaohua6@illinois.edu)
YE YANG
(yeyang3@illinois.edu)
MINGJUN WEI
(mingjun9@illinois.edu)
YINJIE RUAN
(yinjier2@illinois.edu)

TA: Guozheng He

May 10, 2024

Contents

1 Introduction

1.1 Background

Academia and industry alike are increasingly intrigued by the intelligent transportation system. Managing the expanding data load while aiming for quick response times necessitates the advancement of edge computing systems[1].

1.2 Problems

1.2.1 Energy Consumption

Efficiency in energy usage remains a paramount concern in computing servers. The majority of computing devices consume considerable energy for both computational processes and cooling. Traditional cooling mechanisms, such as fans or air-conditioners, can alone consume upwards of 40% of the total energy required within a data center. Moreover, the rapid advancement of Artificial Intelligence (AI) forebodes a surge in electrical energy demands in the forthcoming years. Elon Musk has even forecasted the possibility of AI exhausting electricity resources as early as next year[2]. Consequently, urgent measures are imperative to mitigate energy consumption in electronic devices. This project aims to explore avenues to conserve energy in edge computing systems, potentially enhancing the overall energy utilization efficiency. For example, identifying effective strategies to minimize server cooling requirements could significantly contribute to energy savings.[3].

1.2.2 Immobility of edge servers

The primary concept behind deploying edge computing systems is to position them at fixed locations near the end user, thereby covering a specific service area. However, this approach faces challenges related to low utilization rates for various reasons. Firstly, the duration during which a car is served by a particular edge server is limited due to difficulties in reliably transmitting large volumes of data, hindering the server's ability to undertake time-intensive tasks such as deep learning problems. Consequently, the server may either fail to provide the intended service, resulting in idle periods, or opt to transfer the data to the next server as the vehicle approaches, adding complexity to the process.

1.3 Solutions

1.3.1 Propose

Given the aforementioned challenge, our objective is to install the edge server directly onboard a vehicle, such as a car. We suggest that leveraging the vehicle's movement can significantly enhance server cooling through exposure to wind. Additionally, by adopting a dynamic service coverage approach, the average utilization rate of a server is expected to increase statistically. This is due to the presence of server stations that are compelled to remain in fixed locations to serve areas with lower vehicle flow rates.

1.3.2 Scheme

With the server now onboard, our focus shifts to designing a protective enclosure with a ventilated structure to ensure stability, weather resistance, and efficient wind cooling. To enhance server performance and conserve energy, we aim to develop a control module capable of dynamically adjusting CPU usage based on cooling efficiency. Additionally, since the server cannot be accessed via wired connections, we intend to integrate a wireless communication module to enable connectivity with nearby base stations. Furthermore, we seek to integrate edge computing with cloud computing to some extent, facilitating data transmission between edge and cloud servers.

1.4 Visual Aid

1.5 High-level requirements list

1. Energy saving:

The cooling system is capable of reducing electricity consumption by 40% when compared to traditional temperature-based cooling methods, such as using only fans or air-conditioners.

2. Cooling effectiveness:

Irrespective of the car's speed, the peak server temperature can be reliably maintained below 70°C at all times.

3. Computational performance improvement:

Utilizing wind cooling enables dynamic adjustment of computing intensity, enhancing overall server utilization and performance. The onboard vehicle server can handle over 10% more workload compared to servers deployed indoors.

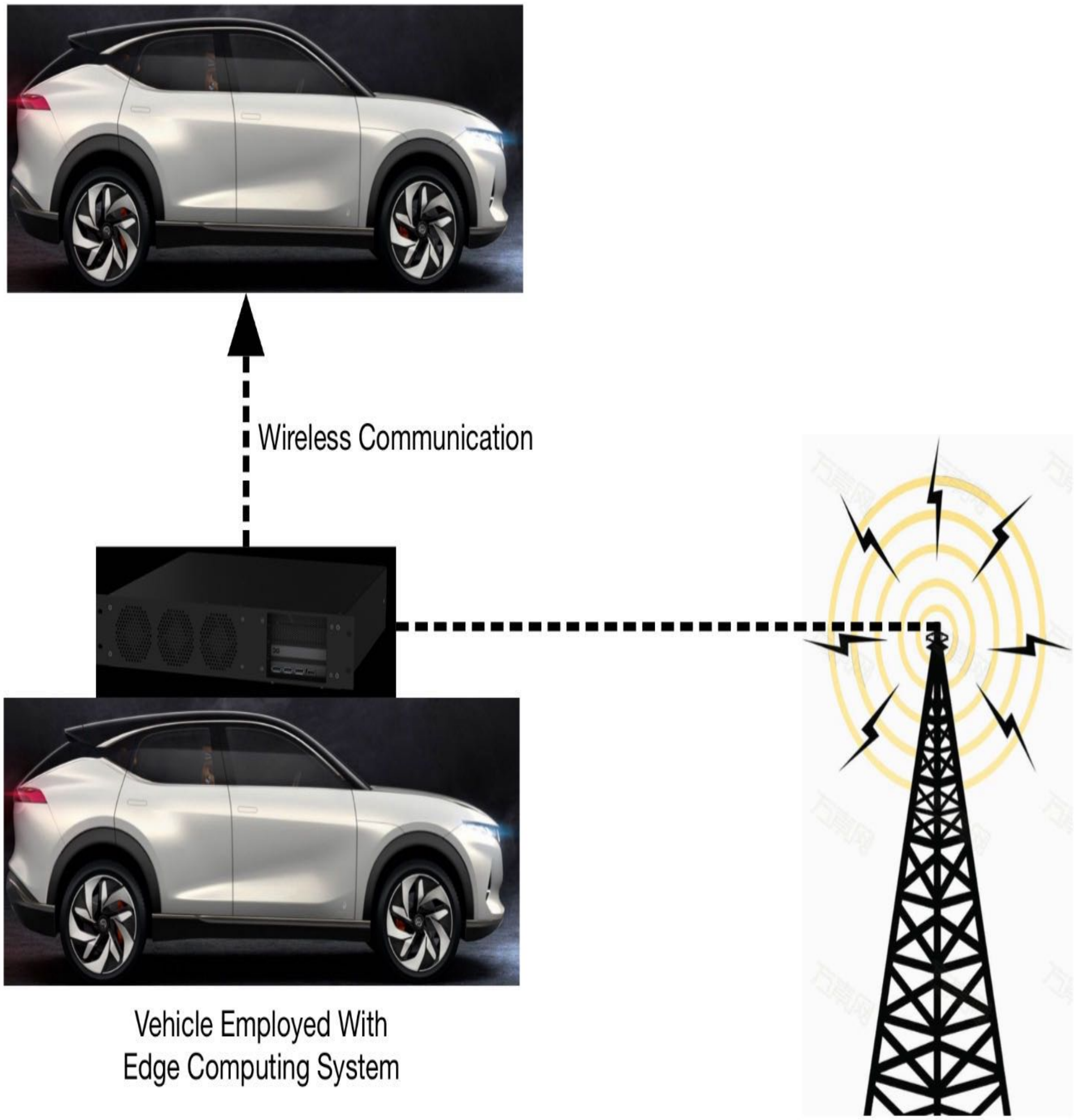

Figure 1: Visual Aid

2 Design

2.1 Block Diagram

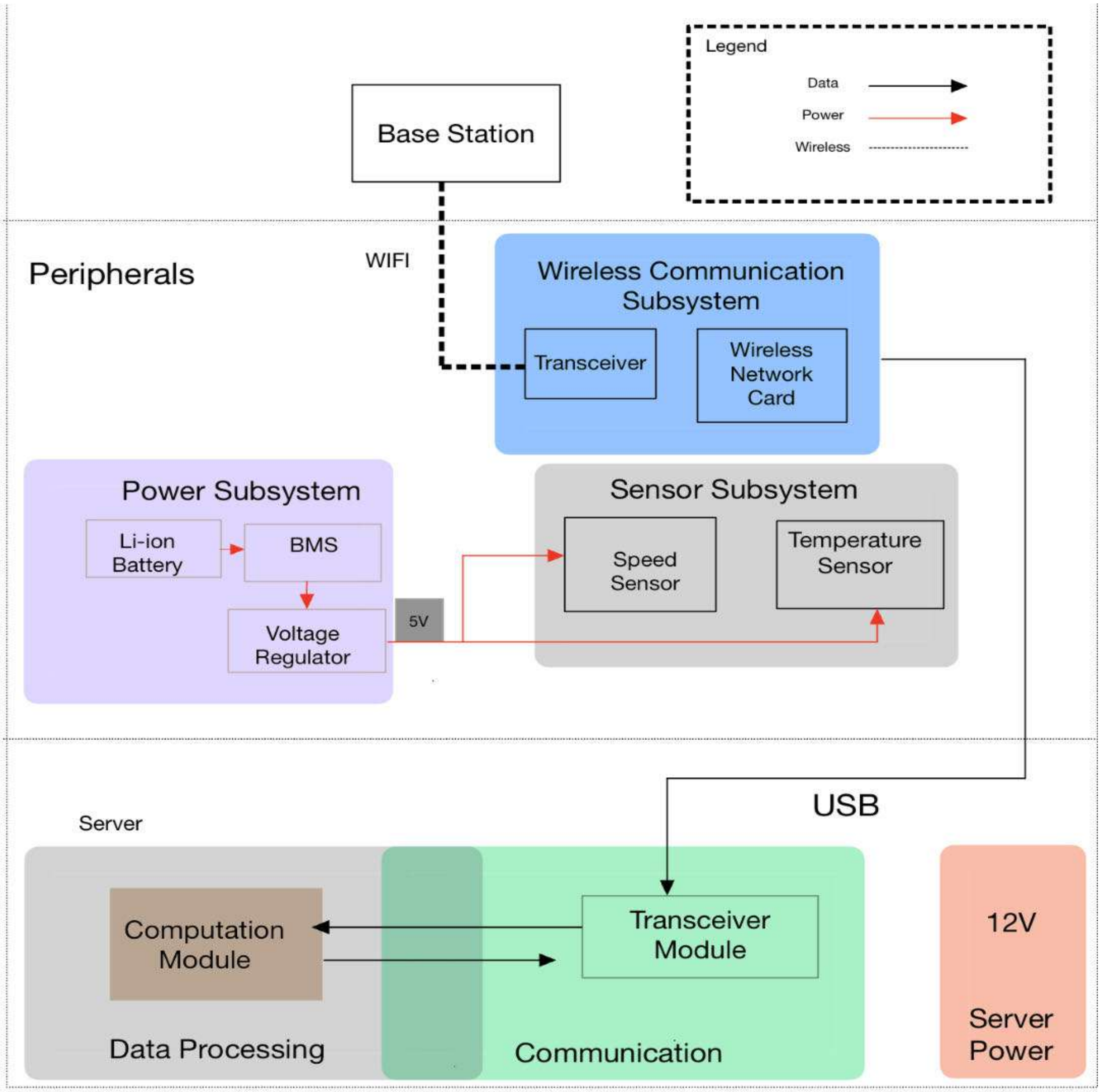

Figure 2: Block Diagram

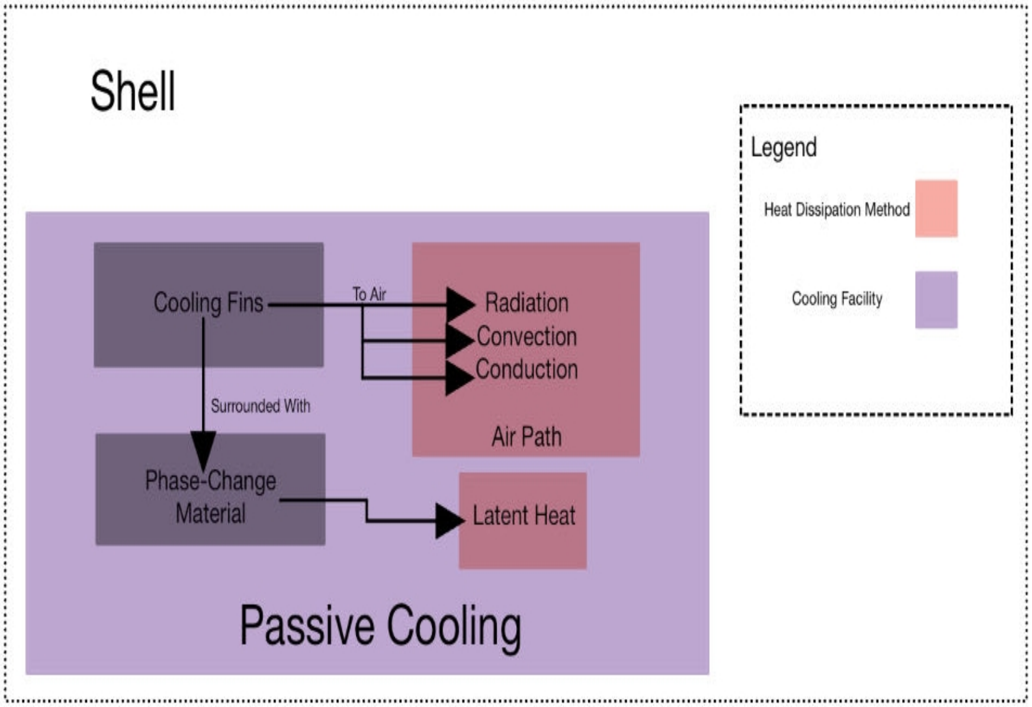

Figure 3: Block Diagram

2.2 Physical Diagram

The physical representation of the protective and functional enclosure designed for our server is depicted in Figures 4 and 5. These figures provide a clear justification for the design block diagram presented in Figure 3. In Figure 4, Aluminum 6061 is selected as the material for the enclosure, while Figure 5 illustrates the wiring layout, showcasing the dimensions and internal structures of the enclosure. It is evident that three wind tunnels are incorporated as part of the air cooling subsystem. The server is positioned at the center within a container of adjustable size and surrounded by fins. This design allows servers of varying sizes to be accommodated within the shelter. The fins are sloped downward to prevent water accumulation on the surface. The front of the enclosure is shaped outward to enhance airflow into the wind tunnel, resulting in a significant funneling effect and increased convection heat transfer rate between the fins and the air.

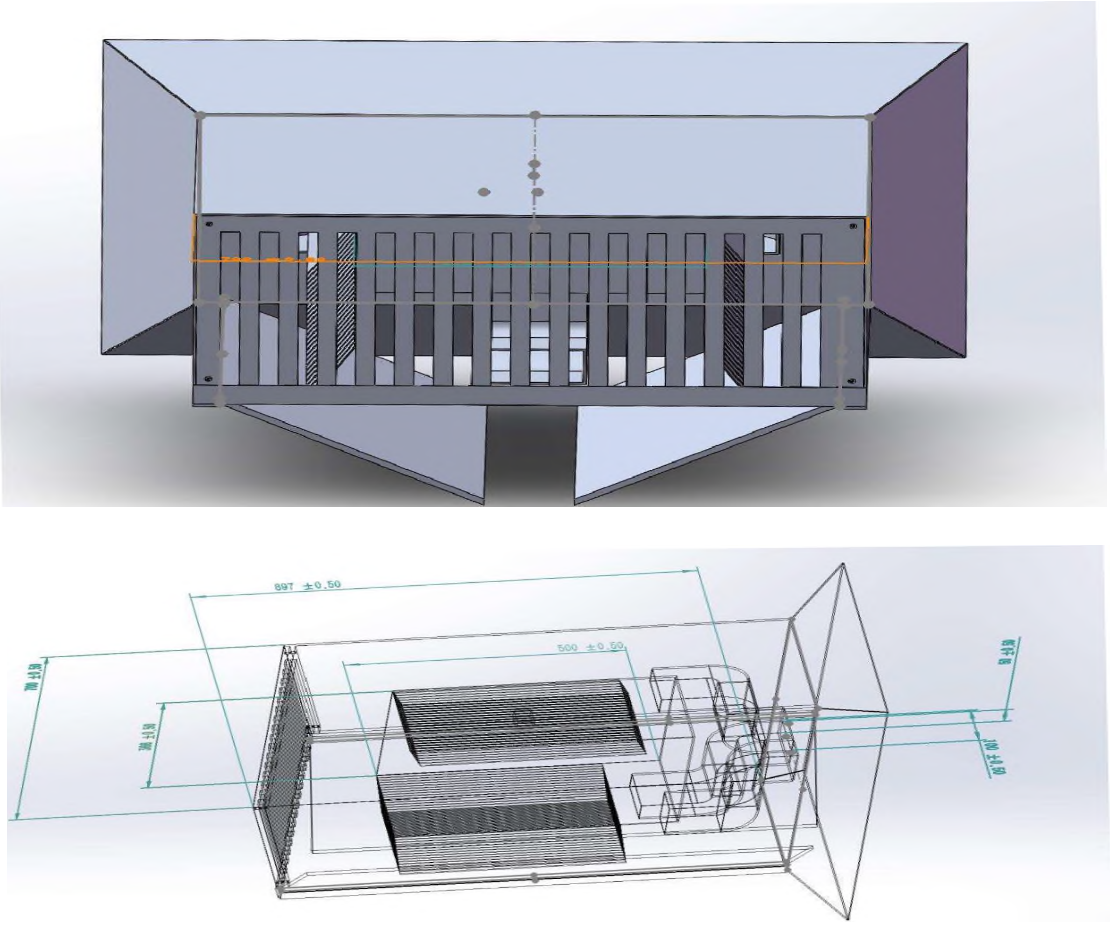

Figure 4: Physical Diagram

2.3 Subsystems

Our design is divided into three main systems, which are the server, the shelter of the server, and the peripherals.

2.3.1 Server

This section encompasses the software aspect of the server, including data processing, communication, and power provision, tailored for practical use in edge computing.

Configuration:
- We will configure Ubuntu on the server and deploy Docker based on Ubuntu, a widely adopted approach in edge computing. The server will run deep learning algorithms to operate at a specified intensity.

Experiment participation:
- Under these circumstances, we will position the server in front of a laboratory wind tunnel outlet to assess the effectiveness of wind cooling in maintaining safe temperatures for the server.

Communication:
- We will enhance the server's wireless communication capabilities to enable access to end users via the Internet. To achieve this, we will install a driver for a USB wireless network card and configure Ubuntu with integrated software tools such as "nmcli," "iw," and "wpa-supplicant" for Internet connectivity.

Data Processing:
- To simulate real-world scenarios for a mobile server, we will execute traditional deep learning tasks on the server to mimic the data processing it would typically undergo, such as decision-making for self-driving. We will dynamically adjust CPU computational intensity and fan operation based on CPU temperature. Additionally, we will develop code blocks to control and manage hardware resource utilization and set temperature thresholds for CPU operation, with relevant modifications made to documents in the "/etc" directory in Linux. Throughout the experiments, we will primarily monitor CPU temperature, CPU utilization statistics, and embedded fan speeds.

Power:
- In practical usage, the server will be accompanied by a battery on the vehicle to supply power.

2.3.2 Shelter

This block serves as the enclosure for the server, providing three main functionalities.

• Firstly, it serves the basic purpose of securely mounting and stabilizing the server on the vehicle.

• Secondly, it acts as a cooling system utilizing two methods illustrated in the physical diagrams (Figures 4 and 5). The first method involves a double-deck shelter comprising an outer and inner layer. The inner layer holds the server and incorporates cooling fins and phase-change material, facilitating heat dissipation through radiation, convection, conduction, and latent heat. The outer layer shields the server from direct sunlight, maintaining a stable temperature environment within the shelter. The second cooling method employs three air tunnels, strategically positioned to collect wind at the shelter's front, optimizing airflow for increased convection heat exchange among the cooling fins.

• Thirdly, the shelter is waterproof and provides complete protection for the server during rainy conditions, as depicted in the physical diagram. Additionally, the shelter design incorporates tilted devices such as fins and two disconnected boards to prevent water accumulation.

• Furthermore, the inner shelter's size is adjustable to accommodate servers of varying sizes.

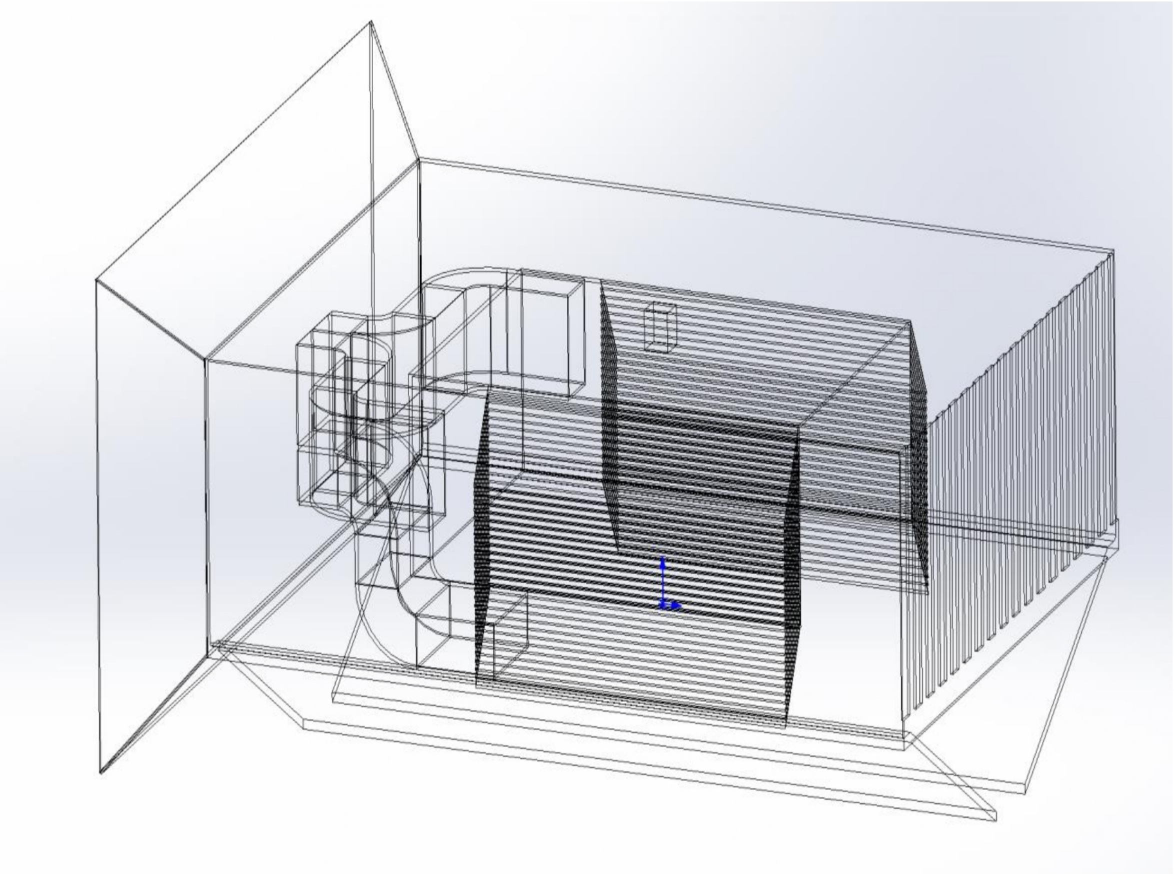

Figure 5: CAD Model for Double-Deck Shelter

In manufacturing, the server shell should be made by different materials and assembled separately.

This is because the main goal of the design is to achieve a perfect balance between cooling and energy-consumption. Since the design is mostly sealed, the water-proof should not be a significant element in choosing materials. From the famous kinematic energy equation:

$$K = \frac{1}{2}mv^2 \tag{1}$$

We know that a lighter material will consume less energy. Therefore, a first principle is that the overall material should be light enough.

To maximize cooling effect, considering the famous heat transfer function:

$$q = -kA\frac{dT}{dx} \tag{2}$$

Hence, the heat conduction properties of the material play a crucial role in heat dissipation. I opted for Al6061 for the inner shell material due to its lightweight nature and exceptionally high heat conduction rate. However, the selection of material for the outer shell requires careful consideration. Considering ease of manufacturing, both the front and air tunnel should be crafted from alloys, thus Al6061 is chosen. Nevertheless, the bulk of the outer shell should not consist of alloys. In warmer climates, the outer shell material should have low heat conductivity to maintain a cooler inner temperature environment, while in colder climates, heat conductivity is less critical as overheating is less likely. Therefore, the majority of the outer shell is made from plastic. Through meticulous selection, POM is utilized for the remaining outer shell material, as it is sufficiently robust to support the server's weight.

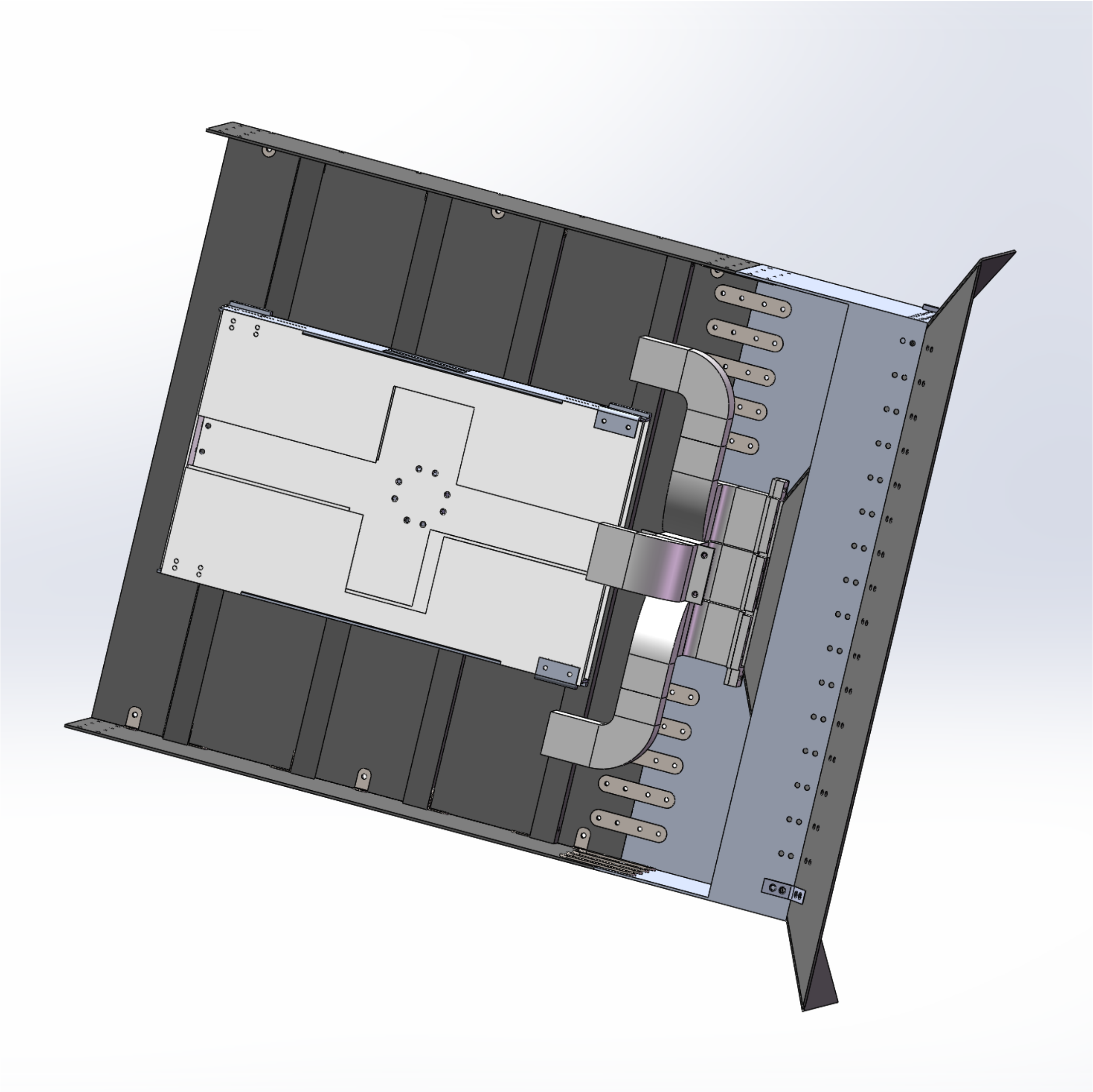

Figure 6: Server Shell in Reality

2.3.3 USB Wireless Network Card

We employ the AX300 high-gain USB wireless network card, manufactured by ”MERCURY,” for establishing wireless connectivity with the base station via nearby WiFi or hotspot, ensuring reliable accessibility through specific protocols. Equipped with a high-gain transceiver antenna, it facilitates stable data transmission and reception. This network card seamlessly connects to the server via USB protocol, requiring only the installation of the company-provided driver software on the Linux system for optimal functionality.

Moving on to its core functionality, the wireless network card operates within the Link Layer of the OSI network model, serving as a bridge between the host and physical hardware through the NIC (network interface controller). Essentially, it acts as a ”gateway,” assigning IP and MAC addresses to the server and processing incoming and outgoing data in ”Frame” format via the transceiver antenna. Employing protocols like CDMA ensures data privacy and enhances performance.

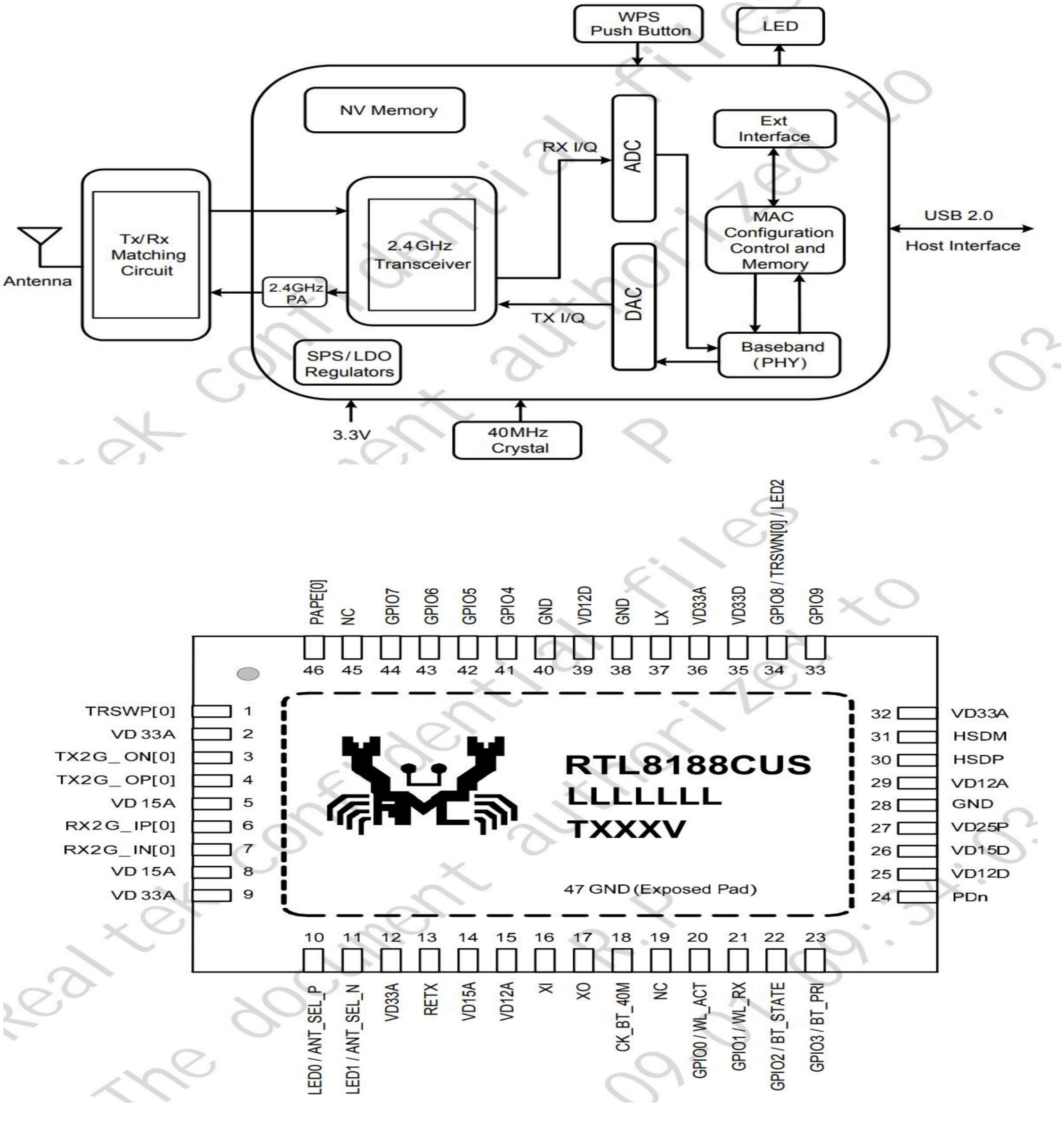

Figure 7: The block diagram of USB Wireless Network Card

2.3.4 The sensor subsystem

- A temperature sensor.

Input: Li-ion battery.
Output: The environment temperature [°C].
We incorporate the DS18B20 temperature sensor to monitor specific areas on the server's body. By working in conjunction with a sensor attached to the CPU, it provides comprehensive temperature data, compensating for potential uneven cooling effects from the wind cooling system. Below is the circuit diagram for the DS18B20 sensor.[4].

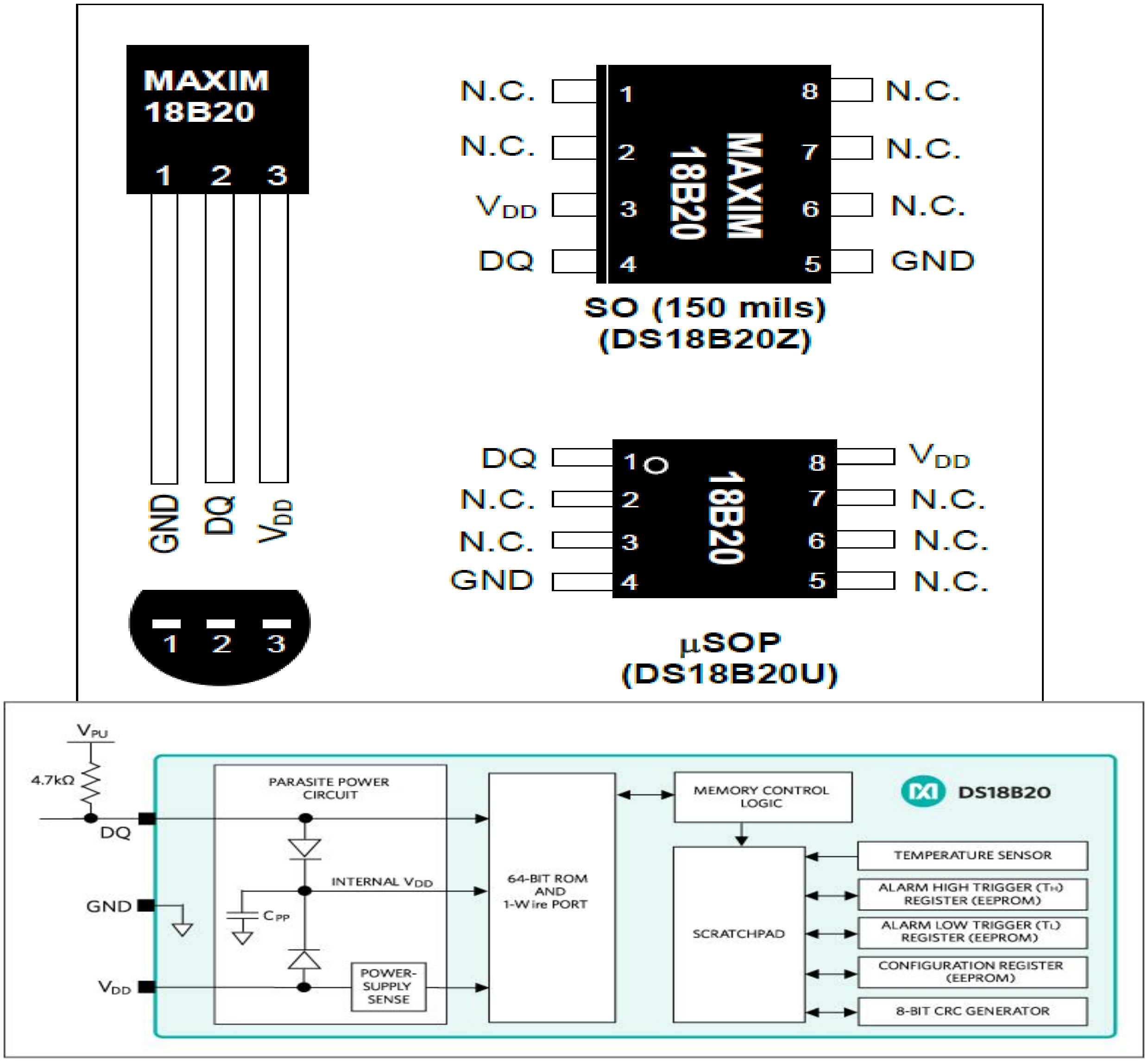

Figure 8: Temperature Sensor

- A wind speed sensor.

Input: Li-ion battery.
Output: wind velocity [m/s].
We have incorporated the SM5388M wind speed sensor into our setup. Its purpose is to address the discrepancy between the smaller cross-sectional area of the wind tunnel outlet in the lab and the actual air flux required in practice. Initially, we will expand the outlet to accommodate the shelter and utilize the wind speed sensor to measure the incoming wind speed at the shelter. The recorded data will be utilized for subsequent quantitative analysis. Below is a photograph of our wind speed sensor along with its basic schematic diagram (in Chinese).

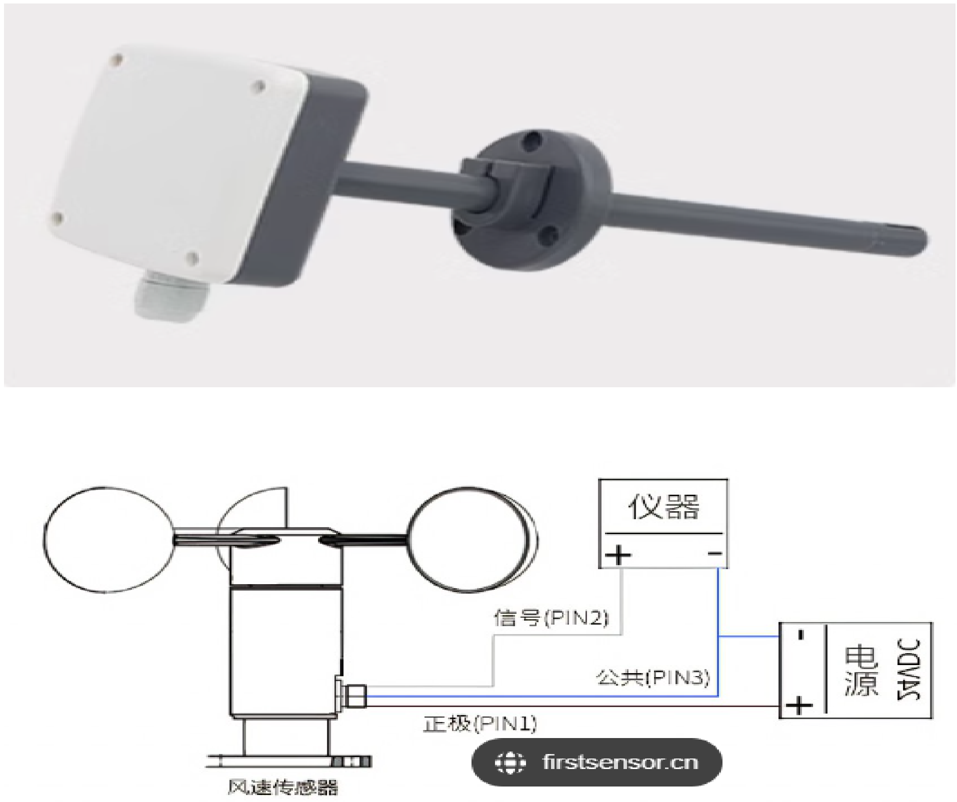

Figure 9: Wind Speed Sensor

2.3.5 The power subsystem

The power subsystem is required to provide continuous power supply to the sensors at specific voltage levels such as 3.3V or 5V, depending on the requirements of each sensor. Below are the block diagram illustrating the power subsystem and the inner structure diagram of the voltage regulator (in Chinese). The primary function of the voltage regulator is to ensure a stable output voltage, adapting to dynamic loads. This is achieved through the utilization of several NPN transistors within the regulator circuitry.

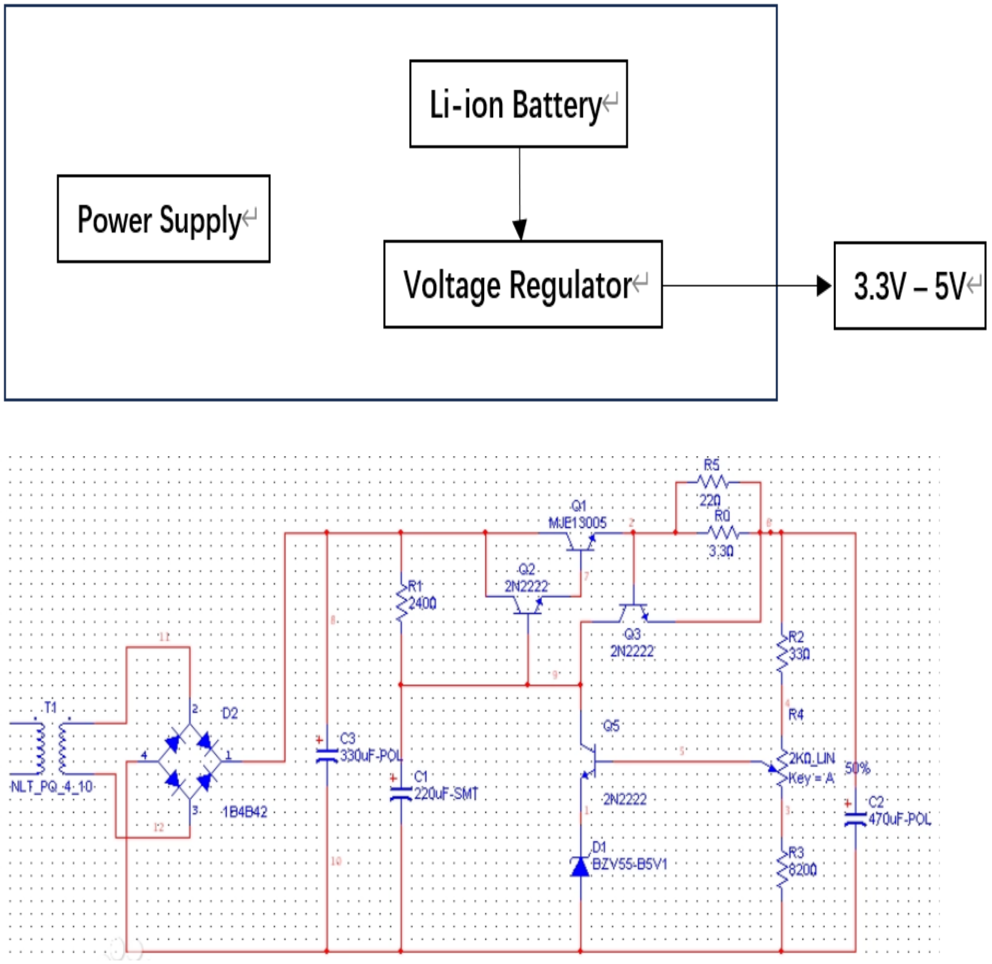

Figure 10: Li-ion Battery and the structure of regulator

2.4 Requirements and Verifications

2.4.1 Shelter

Table 1: Shelter

Requirement	**Verification**
1) The shelter must withstand rainfall conditions to protect the server from moisture while the vehicle is in motion on the highway.	1) Conduct wind tunnel experiments to evaluate the shelter's waterproof capability. Utilize a spray to simulate rainfall with varying vapor concentrations. If the shelter can prevent the server from becoming moistened by large water sprays and wind, it meets the waterproof requirement.
2) Traditionally, air-conditioning systems are necessary to cool servers operating at full load due to insufficient cooling provided by embedded fans. However, our project mandates effective server cooling solely through wind circulation within the shelter's outdoor air-path. The shelter must prevent server overheating, especially when the vehicle is moving slowly (resulting in low wind speeds).	2) Employ a wind tunnel laboratory to assess the shelter's cooling effectiveness. Operate the server at full load to generate heat and activate the wind generator to simulate airflow through the shelter. Record CPU temperature data over time. For comparison, repeat the process with the wind generator turned off. If the CPU temperature remains stable at a certain value during the first case, it confirms an effective cooling system.

Notes: Further description of the experiments and success measurement is shown in the following pages.

2.4.2 Server

Table 2: Server

Requirement	**Verification**
1) The server has been properly configured.	1) Execute a deep learning code block to verify if it completes its task as expected.
2) The server can establish bidirectional data communication over the Internet.	2) Transmit the required dataset to the server for the deep learning problem and verify the correct reception of data by the server.
3) The server effectively maintains its CPU temperature below 70°C at all times.	3) Conduct tests in the wind tunnel laboratory, varying wind speeds from 0 to 30 m/s to simulate different car movement conditions. At lower wind speeds, the server will automatically halt its operations if its temperature reaches 70°C; at higher wind speeds, the server should remain below 70°C due to effective cooling.

Notes: Further description of the experiments and success measurement is shown in the next page.

For specific experiments, we intend to carry out waterproof experiments and wind tunnel experiments.

1) Waterproof experiments will be conducted in a wind tunnel.
- Initially, position four humidometers inside the container at its four corners, and set the wind tunnel to a specified wind speed.
- Next, simulate rainfall by spraying water into the air tunnel.
- After a designated time period, record the initial and final humidity values displayed on the humidometers.

The effectiveness of waterproofing will be assessed based on two criteria: the maximum final humidity (MFH) among the four humidometers, and the maximum difference in humidity (MDH) among them. By plotting diagrams correlating wind speed/spray rate with MFH/MDH, we can determine the waterproofing effectiveness based on the slope of the graphs.

2) Cooling experiments will be conducted in a wind tunnel
- Execute deep learning programs on the server and record the CPU temperature curve over time. Ensure that if the temperature exceeds 70°C, the server will automatically suspend computing.
- Apply the same computing load and activate the wind tunnel to allow airflow. Monitor whether the CPU temperature stabilizes at a specific value below 70°C. Adjust the wind speed to ascertain the range of average wind speeds that maintain the server's temperature below 70°C. Record these findings.

If, under the highest possible computing load, the server's temperature stabilizes below the alarm threshold with a natural wind speed (approximately 25 km/s in typical traffic conditions), the cooling effectiveness of the structure is confirmed. (If wind speed is low, the speed sensor should provide feedback to the CPU to reduce the computing load).

Note: Due to the server's susceptibility, we have an alternative plan involving the use of a heat generation bar with constant heating power to simulate the server's operation. In this experiment, we will conduct two heating simulation tests: one without wind and one with wind in the wind tunnel. We will record the time taken for the bar to heat from room temperature to a predetermined higher temperature in both tests and compare the results.

2.4.3 Sensors

Table 3: Sensors

Requirement	**Verification**
1) The speed sensor must accurately measure wind speed within a 5% margin of error within the wind tunnels of the shelter.	1) Validate the effectiveness and accuracy of the speed sensor in a wind tunnel setting. Position the speed sensor at the tunnel outlet and compare the measured speed to the wind tunnel's indicated speed. If the difference falls within a 5% margin of error, the speed sensor is deemed reliable.
2) The temperature sensor must exhibit immediate responsiveness to temperature fluctuations, as prompt action is required to record transient temperature values, given the sensitivity of electronic components in the server to temperature changes.	2) Ahead of the laboratory experiments, conduct a comparison with a mercury thermometer to ensure alignment of room temperature readings. Subsequently, immerse the temperature sensor in ice water and verify if the panel temperature registers within ±1°C of 0 Celsius. This assessment confirms the accuracy of the temperature sensor. Simultaneously, record the sensor's response time from room temperature to freezing point. This data may aid in calibrating results obtained during physical experiments.

2.5 Tolerance Analysis

For the heat transfer, due to the container′s complex shape and 3-tunnel combining effect, the final temperature distribution is done by Fluent. Here we roughly calculate the rate of convective heat loss.

With:

- T_{air} is the ambient air temperature,
- T_{c} is the surface temperature of the server,
- L_{c} is the horizontal length of the server which is seen as a flat plate,
- L is the vertical length of the plate,
- u_{air} is the wind speed,
- nu is the kinematic viscosity of the air,
- k is the heat conductivity of the air,
- Pr is the property of the air,
- Re is Reynolds number.

Known: $T_{\text{air}} = 298\text{K}$, $T_{\text{c}} = 348\text{K}$, $L_c = 1\text{m}$, $L = 0.6\text{m}$, $u_{\text{air}} = 10\text{m/s}$

Analysis: $\overline{T} = \frac{T_c + T_{\text{air}}}{2} = 323\text{K}$, at 323K, $\nu = 18.9 \times 10^{-6}\text{m}^2/\text{s}$,

$k = 28.9 \times 10^{-3}\text{W/mK}$, $Pr = 0.707$,

$Re = \frac{u_{\text{air}} L}{\nu} = \frac{10 \times 2}{18.9 \times 10^{-6}} = 1.058 \times 10^6 > 5 \times 10^5$, turbulent flow

Use correlation $Nu_L = 0.037 Re^{4/5} - 871 Pr^{1/3}$

$\frac{hL_c}{k} = Nu_L$

$\Rightarrow h = \frac{Nu_L \cdot k}{L_c}$

$\Rightarrow h = \frac{(0.037 \times (1.058 \times 10^6)^{4/5} - 871) \times 0.707^{1/3} \times k}{L_c}$

$\Rightarrow h = \frac{(0.037 \times (1.058 \times 10^6)^{4/5} - 871) \times 0.707^{1/3} \times 28.9 \times 10^{-3}}{1}$

$\Rightarrow h = 13.87\text{W/m}^2\text{K}$ is the avg HT coefficient

$q_{\text{known}} = hA(T_{\text{air}} - T_c)$

$= 13.87 \times 2 \times 1 \times (348 - 298)$

$= 416.1\text{W} > 300\text{W}$

The critical surface area is 0.433m^2, with this area, the heat transfer rate is 300W. Therefore, as long as the effective contact area between the air and aluminum is bigger than 0.433m^2, the temperature rise is desirable.

We can perform a basic thermal analysis to estimate the airflow required to prevent overheating.

Assumptions:

- The server generates a uniform heat load, which we will assume is 300 W (a typical value for a small server).

- The shell is made of a thermally conductive material (like aluminum).

- The ambient temperature is 25°C.

- The target maximum internal temperature is 75°C to ensure electronic components operate within safe temperatures.

Heat Transfer Calculation:

Using the formula $Q = \dot{m} \cdot c_p \cdot \Delta T$, where:

- Q is the heat power (Watts),
- $\dot{m}$ is the mass flow rate of air (kg/s),
- c_p is the specific heat capacity of air (approximately 1005 J/(kg·K) at room temperature),
- ΔT is the temperature difference (K).

We need to solve for $\dot{m}$, given that $Q = 300$ W and $\Delta T = 50$ K (from $25°C$ to $75°C$):

$$\dot{m} = \frac{Q}{c_p \cdot \Delta T}$$

$$\dot{m} = \frac{300\,\text{W}}{1005\,\text{J/kg} \cdot \text{K} \times 50\,\text{K}}$$

$$\dot{m} = \frac{300\,\text{W}}{50250\,\text{J/kg}}$$

The volumetric flow rate of air required is:

$$\dot{m} = .00597\,\text{kg/s}$$

We can also convert this mass flow rate to a volumetric flow rate using the density of air (which is approximately $1.225\,\text{kg/m}^3$ at sea level and at $25°C$):

$$\dot{V} = \dot{m} \cdot \frac{1}{\rho}$$

$$\dot{V} = .00487\,\text{m}^3/texts$$

The calculation is based solely on ideal conditions. However, during actual server operation, the required airflow rate may vary. We will utilize experimental data for subsequent mathematical analysis.

The primary functions of the shelter encompass waterproofing and cooling. Waterproofing is guaranteed through the shelter's appropriate shape, which effectively guides raindrops to drainage channels, irrespective of manufacturing precision, as gravity prevents water infiltration.

2.6 Cost and Schedule

2.6.1 Cost Analysis

Labor:
According to online sources, the average annual salary for an ECE student from Illinois is $87,769, with an average hourly wage of $36. Assuming each student in this project requires 160 hours, labor costs can be estimated using the following equation:

$$\$36 \times 4 \times 160 = \$23040$$

Parts:

Table 4: Cost of Parts

Parts	**Cost(CNY)**
1) Aluminum Shelter and manufacturing (Manufacturer: ”A 90” Hardware Manufacturer)	¥3500
2) Air Speed Sensor (Part: SN-3009TH-FS; Manufacturer: Puruisenshe)	¥120
3) Server (Part: 1U Black; Manufacturer: Hankong)	¥10800
4) Arduino Board (Part: UNO R3; Manufacturer: Youchuangxiang)	¥161.7
5) Temperature Sensor (Part: DS18B20; Manufacturer: Risym)	¥120
6) PCB Three-proof Paint (Manufacturer:Qijialin)	¥15

The sum of costs would be ¥14716.7

2.6.2 Schedule

Team Member Date	Shaohua Sun	Ye Yang	Mingjun Wei	Yinjie Ruan
Mar. 25 – Mar. 31	Build CAD model of the two-layer server shelter shell	Finish Pretreatment part in simulation-SCDC	Connect USB wireless card to the WiFi in Ubuntu	Install USB wireless driver in Ubuntu
Apr. 1 – Apr. 7	Build CAD model of three branches of vent pipes	Mesh Generation for the server shelter	Understand the kinds of running modes of CPU and learn to draw the CPU temperature curve when running at high speed	Get familiar with how the Deep Learning codes are running
Apr. 8 – Apr. 14	Contact manufacturer for the server shelter	Use Fluent solver to iterate to generate nephogram	Set the threshold temperature in Ubuntu so that the CPU will stop even if the cooling system is absent	Search relevant papers and make summary for recent research background
Apr. 15 – Apr. 21	Modify the CAD model for better manufacturing	Iterate design to assist shelter design improvement	Try to do further work to build a better data processing and storage platform in server to model in practical use	Understand why the Docker and Container is useful in edge computing and try to deploy it in Ubuntu
Apr. 22 – Apr. 28	Prepare heat generation bar to substitute the server in experiment and assemble parts of the manufactured components	Finish the heat transfer and temperature simulation of the entire server shelter	Buy temperature sensor and wind speed sensor and make it into the server system	Build a website for the project and find ways to transfer data to the website
Apr. 29 – May 5	Assemble heat pipes and conduct wind tunnel experiment	Think about waterproof experiment and conduct Wind Tunnel Experiment	Let the server run in full speed and conduct wind tunnel experiment	Prepare the materials for lab and conduct wind tunnel experiment

Figure 11: Schedule

3 Ethics and Safety

We acknowledge the paramount importance of ensuring both team safety throughout the project and the safe practical application of our project's design. We have identified several potential safety concerns that require significant attention:

• During experiments, we prioritize safety by ensuring the wind generator's functionality and removing any debris inside the wind tunnel that could pose harm if blown. Additionally, we maintain a clean environment and incorporate sound suppressors into the wind tunnel. Throughout the experiment, we maintain a safe distance from the wind tunnel. Post-experiment, we thoroughly clean the laboratory to ensure a proper conclusion.

• In commercial applications, as our server is affixed to a vehicle, we prioritize ensuring the reliability of the fixation mechanism. We take measures to ensure that the edge computing system does not compromise the vehicle's safe operation or emergency handling. Recognizing that the edge server processes data from other vehicles, we employ robust methods, including encryption algorithms, to safeguard data and protect user privacy.

Throughout the entirety of the project, we adhere to the IEEE code of ethics. We commit to upholding the highest standards of integrity, responsible behavior, and ethical conduct in all professional activities. We treat all individuals fairly and respectfully, refrain from engaging in harassment or discrimination, and strive to prevent harm to others. Additionally, we endeavor to uphold this code among our colleagues and collaborators [5].

References

[1] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang. “”Vehicular Edge Computing and Networking: A Survey”.” (2020), [Online]. Available: https://doi.org/10.1007/s11036-020-01624-1 (visited on 07/25/2020).

[2] L. Blain. “”Elon Musk: AI will run out of electricity and transformers in 2025”.” (2024), [Online]. Available: https://newatlas.com/technology/elon-musk-ai/ (visited on 03/01/2024).

[3] X. Zhang, T. Lindberg, N. Xiong, V. Vyatkin, and A. Mousavi. “”Cooling Energy Consumption Investigation of Data Center IT Room with Vertical Placed Server”.” (2017), [Online]. Available: https://doi.org/10.1016/j.egypro.2017.03.581 (visited on 05/01/2024).

[4] AnalogDevices. “”DS18B20”.” (), [Online]. Available: https://www.analog.com/cn/products/DS18B20.html (visited on 04/14/2020).

[5] IEEE. “”IEEE Code of Ethics”.” (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 02/08/2020).