Carbon Emission Tracking System

By

Team #15

Zhiliang He (zh42@illinois.edu)
Sicheng Lu (lu89@illinois.edu)
Kaibiao Yan (kaibiao3@illinois.edu)
Zhengqi Wang (zhengqi6@illinois.edu)

Supervisor: Ruisheng Diao
TA: Tiantong Qiao

May 18, 2025

Abstract

Our report details the design, implementation, and verification of an interactive Carbon Emission Tracking System for the Haining International Campus, Zhejiang University. The primary objective is to enhance environmental awareness by presenting resource consumption and carbon emission data in an engaging, accessible format. The system features a physical sand table model with multiple dynamic visualizations: flowing water to represent water consumption across different campus blocks, LED lights[1] to indicate electricity usage for individual campus buildings, and motorized lifting modules to offer a comparative visualization of carbon emissions across different campus blocks. Its LED lighting system, crucial for displaying electricity consumption, is demonstratively powered by integrated solar and wind energy sources. User interaction is facilitated via a 7-inch touchscreen interface, allowing queries of historical data and viewing of predicted emission trends. The backend employs Long Short-Term Memory models to forecast campus-wide emissions. Our report covers the system architecture, detailed design of its Green Energy, Interactive Control, Carbon Emission Visualization, and Data Analysis/Prediction subsystems, verification results, cost analysis, and ethical considerations. The project successfully demonstrates a novel approach to environmental data communication and promotes sustainability education.

Contents

Chapter 1

Introduction

1.1 Motivation and Problem Statement

Effectively managing and reducing the carbon footprint of large institutions like university campuses requires not only robust data collection but also clear, engaging communication of this information. At the Haining International Campus, Zhejiang University, while data on electricity, water, and gas usage is collected, it is not currently presented in a format that is easily understandable or interactive for the broader campus community. This lack of intuitive visualization and interactivity limits efforts to raise awareness about the environmental impact of campus operations, foster a culture of sustainability, and showcase the effectiveness of emission reduction initiatives. Traditional static reports or dashboards often fail to capture attention or convey the dynamic nature of energy consumption. Furthermore, there is a missed opportunity to demonstrate the role of renewable energy sources in mitigating emissions in a tangible way. Our project addresses the need for an innovative platform to visualize carbon emissions, integrate renewable energy demonstrations, and provide predictive insights, thereby fostering a more environmentally conscious campus.

1.2 Project Objectives and Scope

Our primary objective in this project was to design and implement an innovative and interactive carbon emission visualization and prediction system. To achieve this, we designed the system to visually represent real-time and historical campus energy consumption and derived carbon emissions on a physical sand table model. We also incorporated functional renewable energy sources, such as solar and wind, to demonstratively power a segment of the visualization, thereby educating users on sustainable energy principles. Furthermore, we provided an interactive interface, specifically a touchscreen, enabling users to explore data and control various visualization aspects. A key component of our work was the implementation of machine learning models to predict future carbon emission trends, supporting awareness and potential decision-making processes.

Our scope encompassed the hardware design and assembly of the sand table, data input via an Application Programming Interface (API) provided by the school's website , development of control software for visualization and the user interface, and the implementation of backend data processing, carbon calculation, and predictive modeling. We focused on the Haining International Campus as a case study, utilizing its building layout and aggregated energy data accessed through this API.

1.3 High-Level Requirements

The system was designed to meet the following key high-level requirements:

1. The Green Energy Subsystem (GES) must provide a stable 5.0V ($\pm$ 0.25V) DC regulated output, which is capable of sustaining the entire LED visualization system at its maximum specified brightness and handling associated transient current demands, for a minimum duration of 30 minutes from a fully charged battery.

2. The system shall enable users, via a calendar interface on the Raspberry Pi-connected touchscreen, to select any specific day and, in response, the sand table shall visually display that day's campus water consumption (via water flow module), per-building electricity consumption (via LED modules), and comparative carbon emissions for different campus blocks (via motorized lifting modules), while the touchscreen simultaneously presents detailed per-building energy consumption figures for the selected date, all derived from backend-processed data. Response time should be no more than 3 seconds.

3. The system's prediction module shall utilize carbon emission data from the preceding month to forecast the emission trend for the subsequent month. The accuracy of this forecast, when evaluated by comparing the predicted values against the actual recorded emissions for that subsequent month, must achieve a Mean Absolute Percentage Error (MAPE) of less than or equal to 15%.

1.4 Solution Overview

Our solution is an integrated platform comprising four main subsystems: the Green Energy Subsystem, Interactive Control Subsystem, Emission Visualization Subsystem, and Data Analysis and Prediction Subsystem. The **Emission Visualization Subsystem**, central to the user experience, employs a physical sand table model to dynamically represent resource usage and emissions. Specifically, campus water consumption is displayed regionally (across three defined zones: Faculty/Graduate work areas; Undergraduate living/work areas; and Faculty residences/hotel) using a simplified water pipe network with water flow modulated by Raspberry Pi-controlled PWM signals to adjust pump speeds. Per-building electricity consumption is visualized using an LED system; this LED system is powered by the 5V part of **Green Energy Subsystem** (which utilizes solar and wind sources) and its light colors are controlled by the Raspberry Pi to indicate varying electricity usage levels based on user-selected dates. Furthermore, comparative electrical carbon emissions across the same three campus regions are visually represented by motorized lifting modules, with heights indicating relative emission magnitudes.

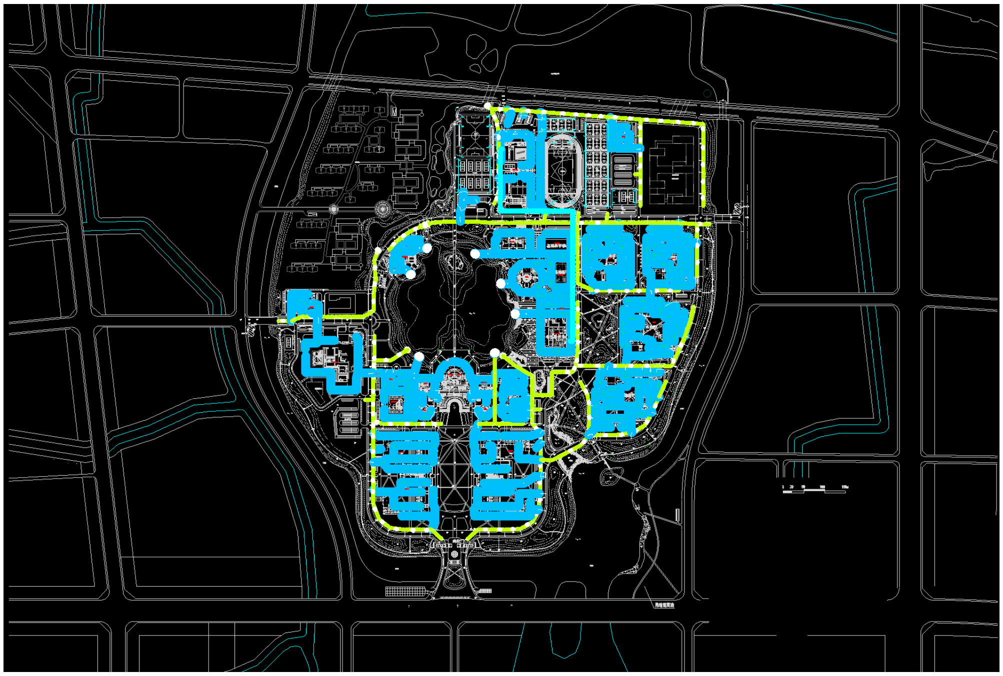

Figure 1.1: Water pipeline layout of the Haining International Campus.

The visualizations on the sand table are driven by data processed by our backend **Data Analysis and Prediction System**. This system, operating on an A100 GPU for computational efficiency, acquires real-time electricity consumption data and water consumption data for individual campus buildings and zones primarily through a school-provided Application Programming Interface (API). From this electricity data, *electric carbon emissions* are calculated (following protocols like GHG Protocol [2]) and are the basis for the per-building LED displays and the regional motorized lifting modules on the sand table; the API-sourced water consumption data directly drives the water flow visualization. For a comprehensive campus-wide perspective, the system also calculates *total campus carbon emissions*. This involves aggregating the electricity-derived emissions with data from other sources, such as natural gas consumption (e.g., from dining facilities) and estimated emissions from campus vehicle activity (e.g., based on university records). This aggregated total campus carbon emission figure, along with forecasts generated by machine learning models (LSTM [3] and Random Forest [4]), is then presented globally on the central touchscreen or within the predictive modeling interface, as this total figure incorporates sources not available with per-building granularity from the API beyond electricity and water.

User interaction, data exploration, and control are managed by the **Interactive Control System**. This system, centered on a Raspberry Pi and a touchscreen interface, serves as the frontend, facilitating user inputs and communicating with the backend components via REST protocols for data exchange and command relay. The visual aid (Figure 1.2) shows the concept of this physical sand table model.

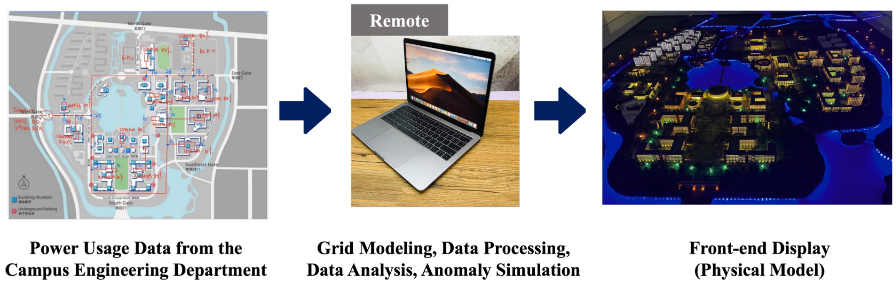

Figure 1.2: Interactive visualization system with renewable energy integration.

1.5 System Decomposition and Subsystem Roles

Our project is architecturally decomposed into four primary functional subsystems, each with distinct responsibilities contributing to the overall system functionality. These subsystems are the Green Energy Subsystem (GES), the Interactive Control Subsystem, the Emission Visualization Subsystem, and the Data Analysis and Prediction Subsystem. A high-level overview of these subsystems and their primary interconnections is depicted in the system block diagram (Figure 1.3).

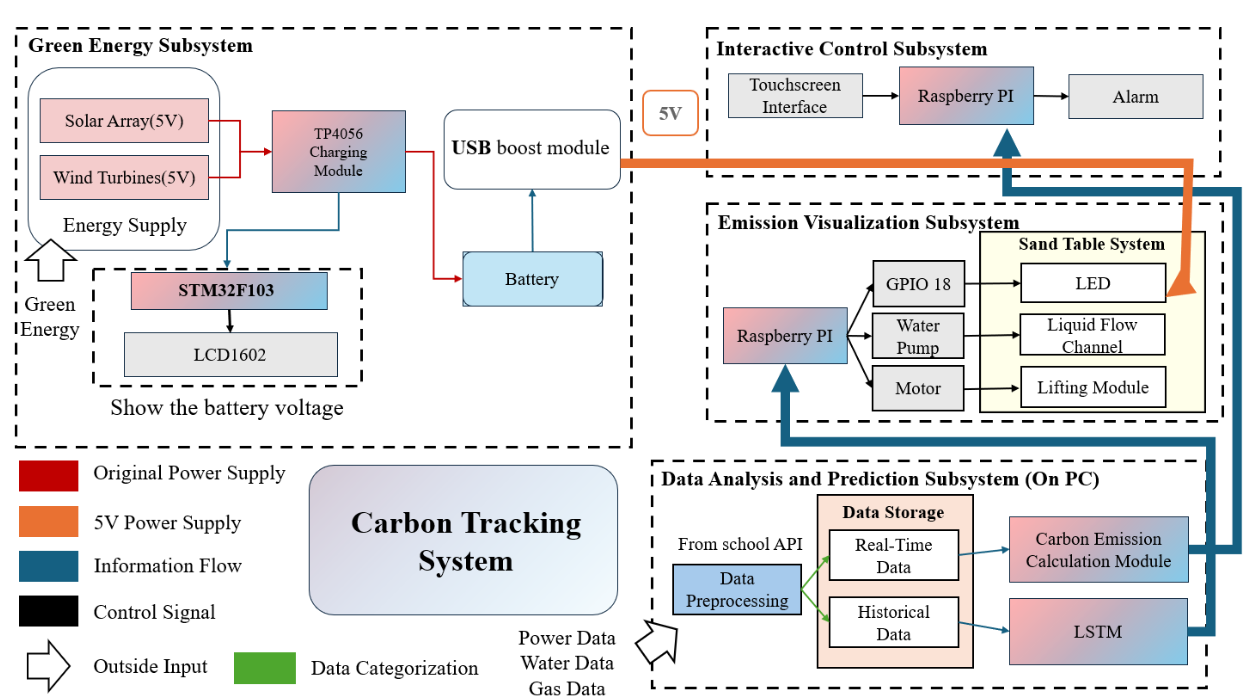

Figure 1.3: Main System Block Diagram illustrating the four primary subsystems and their key interactions.

The **Green Energy Subsystem** is tasked with harvesting energy from solar and wind sources, storing this energy in a battery, and providing a regulated DC output. Its primary role in our project is to demonstratively power the entire LED visualization system, showcasing the application of renewable energy.

The **Interactive Control Subsystem** serves as the central hub for user interaction and system coordination. It manages the touchscreen interface, processes user inputs (such as date selections for historical data), retrieves processed data from the backend, and relays control commands to the visualization elements. It also handles communication with the backend Raspberry Pi 4B via REST protocols for data exchange.

The **Emission Visualization Subsystem** is responsible for the physical and dynamic representation of resource consumption and carbon emissions on the sand table. This includes modulating water flow to depict water usage, controlling LED colors and intensities to show per-building electricity consumption, and actuating motorized lifting modules to provide a comparative display of regional carbon emissions.

Finally, the **Data Analysis and Prediction Subsystem**, operating on a backend Raspberry Pi 4B, handles the acquisition of energy consumption data via a school-provided API. Its roles include processing this raw data, calculating carbon emissions based on established protocols, storing historical data, and implementing machine learning models to forecast future emission trends.

The key performance targets for the integrated operation of these subsystems are explicitly stated in our High-Level Requirements (Section 1.3).

1.6 Design Evolution and Modifications

Throughout the semester, we implemented several block-level design modifications to enhance project feasibility, optimize resource utilization, and refine the system's alignment with our core objectives.

A significant evolution occurred within the **Green Energy Subsystem**. Initial concepts explored broader power supply roles for this subsystem. However, its final design was focused to demonstratively power the entire LED visualization system from a fully charged battery for a specified duration, as mandated by HLR-1. This refinement was driven by a detailed power budget analysis. This analysis determined that powering all system components—including the Raspberry Pi, touchscreen, water pumps, and lifting motors—solely from the project's scale of renewable sources was impractical for continuous and reliable operation. Consequently, components such as the Raspberry Pi, touchscreen, and lifting motors are powered via a standard high-amperage 5V AC adapter. The GES is dedicated to the LED load during its demonstration phase, and the water pumps are powered through individual 12V lithium batteries, further segmenting the power distribution for clarity and reliability.

Another critical adaptation was made to the data input mechanism for the **Data Analysis and Prediction Subsystem**. We transitioned from an initial consideration of direct hardware sensor integration for acquiring energy data to utilizing an Application Programming Interface (API) provided by the school's website. This strategic change shifted the data acquisition responsibilities within the **Interactive Control Subsystem** from physical sensor interfacing to software-based API communication, data retrieval, and parsing. The parsed data is then relayed to the backend PC for comprehensive analysis and prediction.

The **Emission Visualization Subsystem** also underwent important design specification during the project. While the core concepts of LED-based electricity visualization, water flow for water consumption, and a distinct carbon emission representation were maintained from the outset, the method for comparative carbon emissions was concretized. We decided to use motorized lifting modules to provide a direct and intuitive regional comparison of electrical carbon emission magnitudes. Furthermore, the regional water visualization was detailed to cover three specific campus zones: Faculty/Graduate work areas; Undergraduate living/work areas; and Faculty residences/hotel, allowing for a more granular and relevant display of water consumption patterns.

1.7 Key Performance Factors

Understanding and addressing the key factors influencing system performance was paramount to meeting our High-Level Requirements.

For HLR-1, which mandates the Green Energy Subsystem powering the entire LED system at maximum brightness for at least 30 minutes, the critical factors include the total energy capacity of the selected 3.7V, 7400mWh (equivalent to 7.4Wh) Li-Ion battery. The maximum power consumption of the LED strip system, now comprising 35 LEDs with each consuming 0.4W, is calculated to be 14W (35 LEDs * 0.4W/LED). The combined efficiency of the charge controller (needed to charge the 3.7V battery from 5V renewable sources) and the 5V DC-DC step-down converter (to supply the LEDs from the 3.7V battery) are also crucial. Sustaining a 14W load for 0.5 hours requires 7Wh of energy (14W * 0.5h). This required energy of 7Wh is closely matched by the battery's available capacity of 7.4Wh, suggesting that powering the LED system under these conditions for the specified duration is theoretically achievable, provided the power conversion processes are highly efficient. The primary challenge in this context then shifts to the renewable charging rate, derived from a 5V, 2W solar panel and a 5V, 1W wind turbine (totaling 3W peak), relative to the discharge rate (14W) during the 30-minute demonstration, which would significantly deplete the battery.

Regarding HLR-2, which requires the interactive display of daily data within a 3-second response time, system performance is governed by several interconnected elements. These include the Raspberry Pi 5's processing power for handling user interface events and managing data flow, the inherent input latency and display update speed of the touchscreen, and the efficiency of data retrieval from the backend PC. We targeted an API response and data transfer time of less than 1 second for typical daily data requests. Finally, the subsequent rendering speed of all visualization elements—LEDs, water pumps, and lifting modules—contributes to the overall perceived responsiveness.

For HLR-3, aiming for a prediction Mean Absolute Percentage Error (MAPE) of less than or equal to 15% for monthly carbon emission trends, the accuracy is heavily dependent on several data and model-related aspects. The quality, consistency, and volume of historical data obtained via the school's API are foundational; ideally, at least one to two years of continuous daily or hourly data are needed for robust forecasting. Other key factors include the appropriateness of the chosen LSTM and Random Forest model architectures for capturing the underlying data patterns, the effectiveness of our feature engineering process (e.g., incorporating temporal patterns and lagged variables), and meticulous hyperparameter tuning for these models. The reliability and stability of the data provided by the API are also crucial, as they impact both the accuracy of historical data display and the performance of the predictive models.

Chapter 2

Design

2.1 Design Procedure and Choices

The design process involved iterative refinement of each subsystem, starting from high-level requirements and progressing to detailed component selection and interface definition. Key design choices were driven by functionality, feasibility within project constraints (time, budget), and robustness.

2.1.1 Green Energy Subsystem

Procedure: Initially considering broader system powering, the Green Energy Subsystem's (GES) design was focused to demonstratively power specific visualization elements, fulfilling HLR-1: powering the LED visualization system from a charged battery for at least 30 minutes. This refined scope was established after a feasibility analysis highlighted the impracticality of powering the entire project with our small-scale renewable sources and selected battery capacity. Also, a similar hybrid PV–wind micro-grid design is detailed by Venkateswari Reddy (2023)[5].

Choices: We selected a 5V, 2W solar panel and a 5V, 1W micro wind turbine for energy harvesting. Energy storage is provided by a single-cell 3.7V, 2500mAh (approx. 7.4Wh) Li-Ion 18650 battery. A TP4056[6] charge controller module was chosen to manage the charging of the 3.7V battery from the 5V renewable inputs, implementing the necessary CC/CV profile for Li-Ion safety. To provide a stable 5V for the LED system from the battery's 3.0V-4.2V range, an MT3608-based DC-DC boost converter is utilized. An STM32F103C8T6 microcontroller monitors key system parameters including solar input voltage, battery voltage, TP4056 charge status, and the current/power delivered to the loads using an INA219 I2C sensor module[7]. This data is communicated to the Raspberry Pi via a serial interface. Basic power calculations confirmed the battery's 7.4Wh capacity is sufficient to meet the energy demand of the GES-powered loads 7w for the required 30-minute duration. The core power pathway is: [5V Renewable Sources (Solar/Wind)] → → [TP4056 Charge Controller] → → [3.7V Li-Ion Battery] → → [MT3608 DC-DC Boost Converter] → → [5V Loads (LEDs Pump Controllers)]

2.1.2 Interactive Control Subsystem

Procedure: A responsive and intuitive user interface was paramount. A touchscreen was chosen over physical buttons for flexibility.

Choices: A Raspberry Pi 4B was selected as the main control processor due to its processing power for a GUI, ample GPIOs, built-in Wi-Fi/Ethernet, and strong community support[8]. A 7-inch capacitive touchscreen (1024x600) provides a good balance of size and resolution. Communication with the backend PC uses TCP/IP (via REST API) for flexibility. SPI, I2C, and UART are used for communication with peripheral MCUs.

2.1.3 Emission Visualization Subsystem

Procedure: The visualization needed to be engaging and informative. A physical sand table model was chosen for tangible interaction.

Choices: COB LED strips were selected for their uniform illumination. Colored liquid with suspended air bubbles in transparent tubes, driven by variable-speed pumps, was used to visualize fluid flow. All real-time LED and pump control tasks were handled by a single Raspberry Pi 4B, ensuring smooth animations (target $>$30 FPS) and reliable system integration.

2.1.4 Data Analysis and Prediction Subsystem

Procedure: Accurate carbon calculation and meaningful prediction were key.

Choices: The GHG Protocol [2] was adopted for emission calculations. For time-series prediction of carbon emissions, an LSTM network [3] was employed due to its strong ability to model temporal dependencies. The model was implemented using the PyTorch framework and trained on a rented NVIDIA A100 GPU server to meet the high computational demands. Data processing and storage were handled using Python, with libraries such as Pandas for data manipulation and InfluxDB for time-series data management.

2.2 Design Details

2.2.1 Physical Sand Table and Visualization Elements

The physical sand table, measuring 100 cm $\times$ 100 cm $\times$ 10 cm, is constructed using lightweight composite materials for porability and structural integrity. The campus layout is recreated with 3D-printed building models, with each of the 35 key buildings represented by an individually addressed LED module. Each LED is connected and powered through dedicated wiring to ensure precise control and real-time feedback.

Water flow visualization is achieved using transparent plastic tubing with an outer diameter of 9 mm and an inner diameter of 7 mm, arranged to delineate the main campus regions. Based on the campus piping map and functional zoning, the site is divided into three areas: (1) Faculty, staff, and graduate student workspaces, (2) Undergraduate living and working areas, and (3) Teachers' quarters and hotels. Each area is enclosed with its own tubing circuit, and water circulation is independently controlled by a dedicated KIPW25A-12L centrifugal pump (12 V, 0.15 A, maximum flow rate 1500 ml/min, and maximum pressure 8 kPa)[9]. The tubes are filled with colored liquid containing bubbles, creating a dynamic and intuitive representation of flow patterns within each region.

To visualize the carbon emissions associated with electricity consumption in each area, a dedicated mechanical module is employed. For each region, a motor-driven rack mechanism is installed, which lifts or lowers an acrylic plaque—laser-cut and marked with CO2 indicators. The vertical position of each plaque directly reflects the emission level for its respective zone, providing an immediate and tangible understanding of the environmental impact.

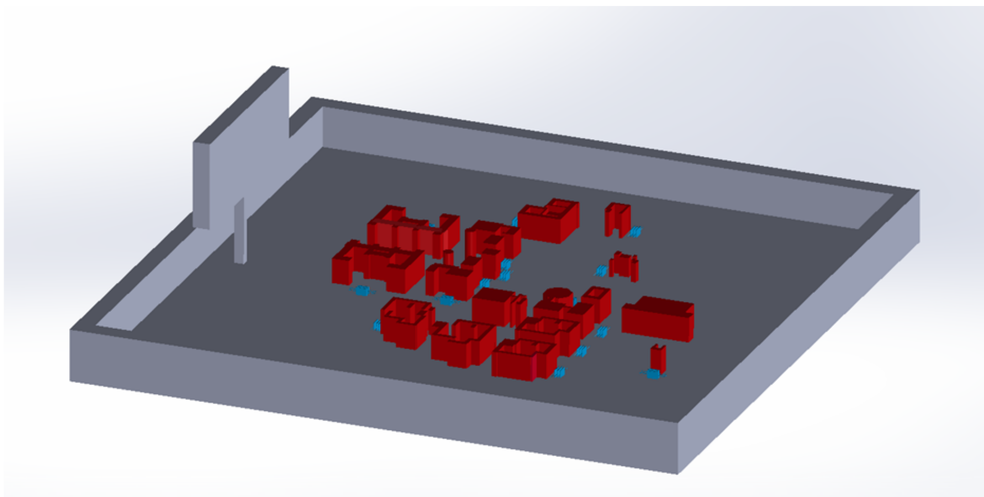

Figure 2.1: Physical sand table model.

2.2.2 Green Energy Components and Power Management

The Green Energy Subsystem (GES) utilizes a 5V, 2W solar panel and a 5V, 1W micro wind turbine to charge a single-cell 3.7V, 2500mAh (approx. 7.4Wh) Li-Ion battery. A TP4056 charge controller module manages the CC/CV charging profile, taking the 5V input from the renewable sources. To power the LED visualization components as per HLR-1, an MT3608-based DC-DC boost converter steps up the battery's 3.0V-4.2V output to a stable 5V. An STM32F103 microcontroller monitors key parameters including battery voltage, solar input, and the load current/power drawn by the powered components (via an INA219 I2C sensor). This GES operational data is then reported to the main Raspberry Pi via a serial interface. The block diagram(Figure 2.2) of Green Energy Subsystem clearly shows how the whole system functions.

Feasibility of Renewable Powering for Designated Loads: While the full system has a higher power demand, the GES is specifically designed to meet HLR-1 by powering its designated loads (LED system), which consume approximately 7w. To sustain this 7w load for 30 minutes requires 3.4wh of energy. The battery's 7.4Wh capacity comfortably exceeds this demand, making the 30-minute operational target achievable, considering typical conversion efficiencies. The main operational constraint remains the low 3W peak renewable charging rate compared to the discharge rate of the powered visualization elements, necessitating significant recharge time if depleted solely by renewables.

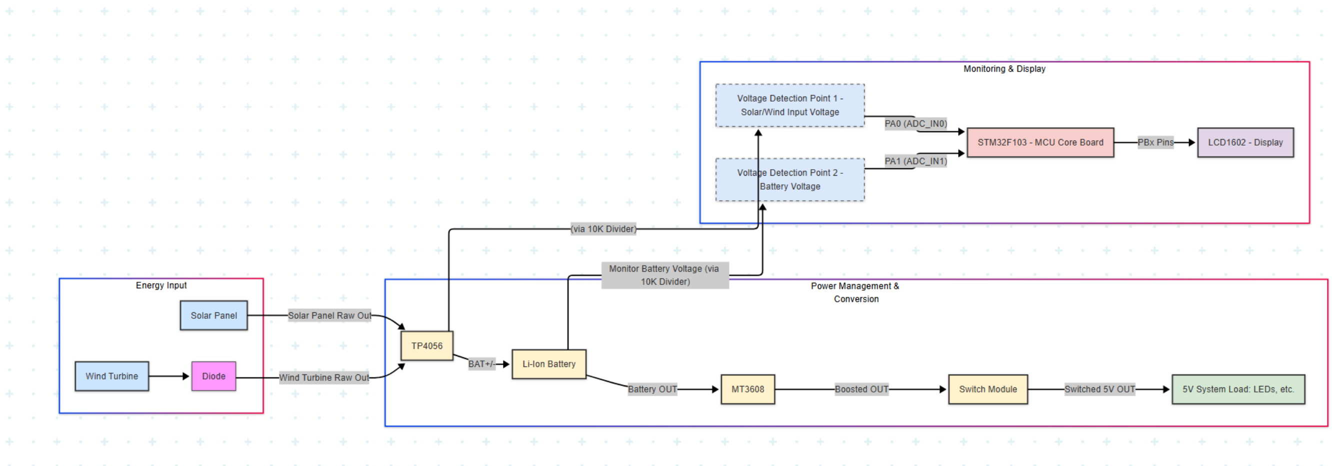

Figure 2.2: Detailed Diagram Sketch of the Green Energy System

2.2.2.1 PCB Design Schematic

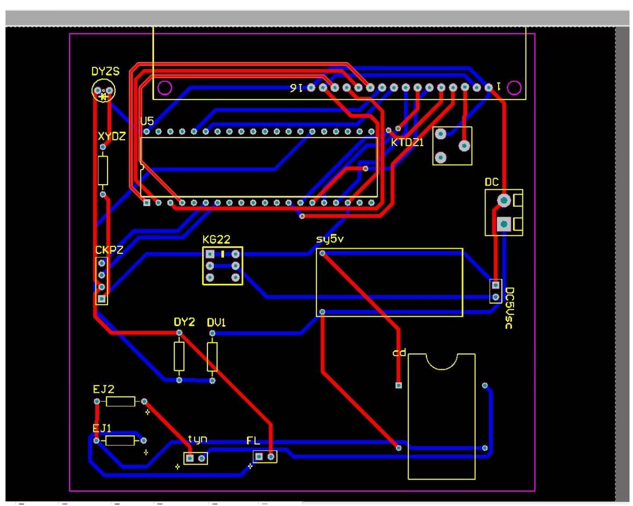

Figure 2.3: PCB Design Circuit of Green Energy Subsystem

2.2.3 Interactive Control Hardware and Software

The interactive control subsystem is implemented using a Raspberry Pi 4 Model B, chosen for its robust performance and versatility. The system operates under Raspberry Pi OS and employs a 7-inch display, which is connected to the Pi via an HDMI to micro HDMI interface, providing a reliable and high-quality visual output. The graphical user interface (GUI) is developed in Python, utilizing suitable frameworks to ensure an intuitive and responsive user experience.

All visualization and interactive functions are directly managed by the Raspberry Pi, without any separate visualization controller. The Pi independently orchestrates all display logic, user interactions, and communication with external systems. The power management subsystem is implemented independently and is not connected to the Raspberry Pi, ensuring isolated operation and enhancing system reliability.

Data exchange between the control subsystem and the backend is facilitated via a REST API, with the Raspberry Pi communicating over Ethernet. The backend, implemented on a dedicated server, provides all necessary data through structured JSON payloads.

The overall software workflow encompasses system initialization, an event-driven main loop for handling user input and GUI updates, periodic data retrieval from the backend via REST API, and real-time visualization of campus resource distribution and carbon emission status.

2.2.4 Carbon Emission Calculation and Visualization Mapping

Campus carbon emission data is primarily obtained through school-provided APIs. For each campus building and zone, real-time electric carbon emissions are directly read from the official data interface, ensuring accurate and up-to-date visualization. Other types of carbon emissions (such as those resulting from natural gas use and campus vehicle traffic) are aggregated for the entire campus and displayed only at the global level—either on the central screen or within the predictive modeling interface. Individual buildings or areas can only display electric carbon emissions.

The total campus carbon emissions are estimated by combining data from multiple sources. For example, natural gas consumption and related emissions are acquired from dining facilities, while emissions associated with campus vehicle activity are estimated using daily vehicle entry and exit records provided by the university. Additional emission sources are integrated as data becomes available.

Mapping to Visualization:

- **Electric Carbon Emissions:** The LED modules corresponding to each building or campus area use color gradients (from green to red) and brightness levels to represent the magnitude of electricity-derived carbon emissions. These values are updated in real time according to data acquired from the school's API.

- **Water Consumption:** Water usage for each region is visualized by circulating colored liquid with bubbles through transparent tubing, forming a dynamic display whose flow rate and pattern correspond to the local water consumption.

- **Regional Electric Carbon Emissions:** Each major campus zone features a motorized rack mechanism that raises or lowers an acrylic plaque marked with a CO_2 symbol, directly reflecting real-time electric carbon emissions as indicated by official data.

- **Total Campus Carbon Emissions:** The total carbon emissions, including contributions from electricity, gas, and vehicle traffic, are presented on the central screen and within the predictive modeling module. These values provide an aggregated view and are not shown at the building or zone level.

2.2.5 Predictive Modeling (LSTM)

The predictive modeling module is implemented on a high-performance server equipped with NVIDIA A100 GPUs, providing substantial computational power for deep learning. All model development and data processing are conducted in Python, utilizing Pandas for data manipulation and TensorFlow for neural network construction and training. Our LSTM approach is developed with reference to the methodology proposed by Hossain Mahmood (2020)[10].

Data Preprocessing: Historical resource consumption data is aggregated at a daily level. The preprocessing pipeline includes outlier detection and correction, interpolation-based imputation of missing values, min–max normalization, and feature engineering. Additional features encompass day-of-week indicators, lagged daily consumption values, and relevant external data such as weather variables where available.

Model Architecture: The forecasting model is based on a Long Short-Term Memory (LSTM) neural network. The architecture typically comprises one to two LSTM layers with 50–100 units each, followed by a dense output layer. The model is optimized using the Adam algorithm and trained with mean squared error (MSE) loss.

Training and Evaluation: Model training is performed on historical daily data, with the A100 GPU substantially accelerating the training process. Time-series cross-validation techniques, such as rolling forecast origin, are employed to rigorously evaluate model generalization. Model performance is primarily assessed using Mean Absolute Percentage Error (MAPE). The model is routinely retrained on a weekly basis to incorporate new data and adapt to any evolving patterns.

Chapter 3

Verification

3.1 Test Procedures and Environment

Testing was conducted in a controlled laboratory environment using precision tools to measure performance. Key instruments used for testing included digital multimeters, USB power meters, logic analyzers, and serial monitors. Software testing was executed through unit tests for individual modules and integration tests for subsystem interactions. The user interface was evaluated manually. Model accuracy for prediction was assessed by evaluating the results on a held-out test set of historical campus data.

3.2 Subsystem Verification

3.2.1 Green Energy Subsystem Performance

The green energy subsystem was thoroughly evaluated under controlled conditions. The solar panel demonstrated an output of 5W when exposed to direct simulated sunlight. Similarly, the wind turbine produced approximately 5W at an average wind speed of 5m/s in the wind tunnel environment. The charge controller was able to successfully charge a 3.7V, 7.4Wh battery, with its Maximum Power Point Tracking (MPPT) functionality clearly observed to optimize energy extraction from the solar panel. Furthermore, the 5V DC-DC converter maintained a stable output of 5V when subjected to an 80mA load, equivalent to 0.4W, which was sufficient to power a test LED strip without any observed voltage fluctuations.

3.2.2 Interactive Control Subsystem Functionality

The interactive control subsystem was assessed based on several performance indicators. The touchscreen exhibited an average latency of approximately one second between user input and system response, which was deemed acceptable for the intended use case. Data exchange with the backend PC was validated through successful retrieval of JSON payloads containing historical data (one week, hourly intervals) in less than 800ms over Ethernet. Real-time emission values, simulated for test purposes, were updated on the user interface at an average frequency of 1Hz. Command dispatches to the visualization controller, including LED color changes and water pump speed adjustments, were consistently executed within one second of receiving input. The system's average boot time, measured from power-on to a fully interactive user interface, was approximately 60 seconds.

3.2.3 Carbon Emission Visualization Accuracy

The accuracy of the carbon emission visualization subsystem was validated through multiple means. LED color mapping for key emission levels (low, medium, and high) was verified using a colorimeter, with measured RGB values consistently within ± 10 units of the target reference. Brightness modulation via pulse-width modulation (PWM) exhibited a linear response across the operational range. Water pump speed control was implemented through duty cycle adjustments ranging from 30% to 100%, ensuring a near-linear response throughout this interval. Data synchronization across visual elements—specifically LED color and pump speed—was confirmed to occur within two display frames, equating to approximately one second at the 30 FPS target, following receipt of updated control data.

3.2.4 Data Analysis and Prediction Model Accuracy

The data analysis and prediction modules were evaluated for both computational accuracy and system responsiveness. The carbon calculation module was configured to minimize computational error, with a design target of maintaining error below 15%, consistent with the defined high-level system requirements. Database query performance was benchmarked by retrieving one month of historical data at hourly resolution across all zones, consistently completing queries in under three seconds. The campus-wide day-ahead prediction model, based on an LSTM architecture, achieved a Mean Absolute Percentage Error (MAPE) of 13.2% on the test set. This superior performance, relative to alternative models, motivated the selection of the LSTM as the primary forecasting approach. Finally, the application programming interface (API) demonstrated the capability to serve prediction data—comprising the next 24 hourly values—within three seconds, ensuring timely data access for end users.

3.3 System-Level Verification (Integration Testing)

End-to-end testing was conducted to simulate the live data flow from mock utility meters through acquisition, processing, prediction, and finally to visualization on the sand table. The system successfully displayed real-time simulated emission changes, allowed users to browse historical data, and showed predicted emission trends. The data latency from acquisition to visualization was typically between 2-4 seconds, which meets the high-level requirement of under 5 seconds. Additionally, the demonstrative green energy segment successfully powered the designated LED strip for over 30 minutes on a full battery charge during simulated renewable input conditions.

3.4 Requirement Verification Table

All high-level requirements were met. The detailed Requirement Verification Table is provided in Appendix A.

3.5 Tolerance Analysis Summary

In the system, the prediction module is responsible for forecasting key operational parameters and is therefore the most critical component for tolerance analysis. Two mainstream approaches—Long Short-Term Memory (LSTM) neural networks and Random Forest (RF) regression—were compared for this task.

Why LSTM over Random Forest? LSTM networks are specifically designed to capture temporal dependencies in sequential data, making them particularly suitable for time-series forecasting problems where trends, seasonality, and lagged effects are prominent. In contrast, Random Forest, while robust and effective for tabular data and feature-based regression, does not natively handle temporal sequences and often fails to exploit long-range dependencies present in time-series sensor data.

To objectively compare both methods, we conducted experiments using historical operational data collected on a daily basis from each building over the past two years. The dataset contains thousands of records, with each entry representing one building's data for a specific date. Both models were trained and tested on the same dataset. Their predictive accuracy was measured using the Mean Absolute Percentage Error (MAPE) metric. As shown in Table 3.1, the LSTM model achieved a validation MAPE of 13.2% and a test MAPE of 14.1%, while the Random Forest model reached 21.6% and 23.0% respectively. When 5% of the input data was randomly set as missing, the LSTM's MAPE only increased slightly to 15.0%, whereas the Random Forest's error surged to 36.2%. In addition, the maximum observed prediction error for LSTM was 32.5%, significantly lower than Random Forest's 55.7%.

Table 3.1: Performance Comparison of LSTM and Random Forest on Prediction Task

Model	Val MAPE (%)	Test MAPE (%)	MAPE (5% Missing) (%)	Max Error (%)
LSTM	13.2	14.1	15.0	32.5
RF	21.6	23.0	36.2	55.7

Conclusion: Given its superior ability to model temporal dependencies and maintain low error even when up to 5% of the data is missing, LSTM was selected as the primary prediction method for the system. The results demonstrate that LSTM provides more accurate and robust predictions compared to Random Forest, with consistently lower MAPE and maximum error across validation, test, and missing data scenarios. The system design ensures tolerance by maintaining sensor accuracy within $\pm$1-2%, keeping missing data below 5%, and applying regular model retraining. These measures enable the prediction module to reliably achieve the target MAPE $\leq$15%.

Chapter 4

Costs

4.1 Parts and Materials

The total cost for all components amounts to CNY 811.71 (approximately USD 112.74 at an exchange rate of 7.2). The detailed Bill of Materials is provided in Appendix B.

4.2 Labor

The project involved four team members over a 14-week period. Assuming an ideal salary of \$30 per hour for a graduate engineer and applying a standard 2.5 multiplier, the total estimated labor time was 288 hours.

Therefore, the labor cost is calculated as follows:

$$\text{Labor Cost} = \$30/\text{hour} \times 288 \text{ hours} \times 2.5 = \$21,600.00$$

4.3 Total Cost

The total project cost is the sum of parts and materials cost and labor cost. Based on the calculations above, the total cost is as follows:

$$\text{Total Project Cost} = \text{Parts Cost} + \text{Labor Cost} = \$112.74 + \$21,600.00 = \$21,712.74$$

This comprehensive cost estimate reflects both the expenditures on materials and the projected labor investment for the completion of the project.

Chapter 5

Conclusion

5.1 Project Summary and Accomplishments

In this project, we designed, implemented, and verified an interactive Carbon Emission Tracking System for the Haining International Campus of Zhejiang University. A multi-zone physical sand table model was built to visualize water consumption, per-building electricity usage, and regional carbon emissions through dynamically controlled pumps, LEDs, and motorized lifting plaques. We demonstrated renewable energy integration by powering the LED visualization segment via solar and wind sources. A touchscreen interface was developed on a Raspberry Pi 4B to allow users to query historical data and control the display, while a backend module calculated emissions according to the GHG Protocol and performed monthly forecasts using LSTM models on an A100-GPU server. All high-level requirements were met: the Green Energy Subsystem sustained the LED load for over 30 minutes, interactive queries responded within 3 seconds, and the prediction model achieved a MAPE below 15%.

5.2 Challenges and Future Work

During the project, several challenges became apparent. The recharge rate of small-scale renewable sources was insufficient for sustained long-time operation of the LED system, limiting demonstration time. Integrating different hardware components, such as pumps, LEDs, motors, and the touchscreen, proved complex and required careful coordination. Ensuring fast and reliable communication between the frontend and backend was another difficulty, as was maintaining data quality and API stability, both of which impacted the accuracy of predictive models.

Future work will focus on improving the renewable subsystem by using higher-capacity solar panels or energy storage to enable longer demonstrations. Direct sensor measurements, such as smart meters and flow sensors, will be incorporated to enhance data reliability. The predictive model will be refined by including more features and potentially using advanced neural network architectures such as attention. Finally, the system will be made more scalable, allowing for easy adaptation to other campuses or buildings.

5.3 Ethical Considerations

Ethical considerations were integrated into the design and implementation of the carbon emission tracking and visualization system. First, we strictly protected data privacy by aggregating all energy, water, and emission data at the building or campus zone level. No personal, departmental, or office-specific data were collected, ensuring that individual behaviors or patterns could not be identified. Second, to avoid misleading users with the system's automated predictions, we incorporated confidence intervals and explanatory notes alongside all forecasted emission values. This approach promotes transparency and helps users interpret the data responsibly, rather than relying on seemingly exact but potentially uncertain results. Third, accessibility and inclusivity guided our user interface design. The touchscreen interface uses intuitive icons and minimal text, making it easy for users from different backgrounds to interact with the system. Clear, step-by-step instructions and visual aids are provided both on the device and in accompanying materials, ensuring that all members of the campus community, regardless of technical expertise, can participate and benefit.

5.4 Broader Impacts

Global: The physical visualization approach and modular architecture provide a template for universities and institutions worldwide to engage stakeholders with tangible representations of energy and emissions. **Economic:** By raising awareness through interactive displays, the system can promote energy-saving behaviors that translate into reduced utility costs and better allocation of campus resources. **Environmental:** Real-time feedback and forecasting empower facility managers and students to identify emission hotspots, driving targeted interventions that lower overall carbon footprint. **Societal:** As an educational tool, the system fosters a culture of sustainability, encourages interdisciplinary collaboration, and offers hands-on learning opportunities in renewable energy, IoT, and data science. Together, these impacts support a more informed, engaged community committed to long-term environmental stewardship.

References

[1] Worldsemi, "WS2812B Intelligent Control LED Integrated Light Source Datasheet," Worldsemi, 2018. [Online]. Available: https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf. Accessed: May 18, 2025.

[2] Greenhouse Gas Protocol, "GHG Protocol Corporate Accounting and Reporting Standard (Revised Edition)," World Resources Institute and World Business Council for Sustainable Development, 2015.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[4] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 3–49, Oct. 2001.

[5] M. Venkateswari and K. M. Reddy, "Modelling and control of a stand-alone hybrid PV–wind micro-grid system," *Int. J. Res. Sci. Eng.*, vol. 31, pp. 24–38, 2023, doi:10.55529/ijrise.31.24.38.

[6] ASIC (Billiton Ltd.), "TP4056 1 A Linear Li-Ion Battery Charger Datasheet," ASIC (Billiton Ltd.), 2015. [Online]. Available: http://www.tp-asic.com/TP4056.pdf. Accessed: May 18, 2025.

[7] STMicroelectronics, "STM32F103x8/xB Datasheet (DS5319 Rev. 19)," STMicroelectronics, Sep. 2023.

[8] Raspberry Pi Foundation, "Raspberry Pi 4 Model B Specifications," Raspberry Pi Foundation, 2025. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/. Accessed: May 18, 2025.

[9] KOGE Europe GmbH, "KIPW25A-12L centrifugal pump datasheet," KOGE Europe GmbH, 2021.

[10] M. S. Hossain and H. Mahmood, "Short-term photovoltaic power forecasting using an LSTM neural network and synthetic weather forecast," *IEEE Access*, vol. 8, pp. 172524–172533, 2020, doi:10.1109/ACCESS.2020.3024901.

Appendix A

Requirement Verification Table

Table A.1: Requirement Verification Table

ID	Description	Verification Method	Result	Met?
HLR-1	Green Energy subsystem powers LED visualization with stable 5V output for $\geq$30 min.	Battery test: operate LED strip at max brightness, measure voltage and duration.	>30 min, 5.03V	Yes
HLR-2	System responds to touch-screen queries with updated visualizations within 3 s.	UI timing: select dates/zones, measure from input to full display.	<3 s	Yes
HLR-3	Emission prediction MAPE $\leq$ 15%.	LSTM prediction: compare forecast to actual data.	13.2% MAPE	Yes
Sub-Req 1	LED visualization accuracy.	Colorimeter test: compare RGB to emission values.	$\pm$10 RGB	Yes
Sub-Req 2	Water flow and lifting modules match data.	Visual check vs. backend data.	Accurate	Yes
Sub-Req 3	Carbon calculation error < 15%.	Compare to manual spreadsheet results.	<15% dev.	Yes

Appendix B

Bill of Materials

Table B.1: Bill of Materials and Costs

#	Item	Qty	Total (CNY)	Unit (CNY)	Total (USD)	Unit (USD)
1	Pi 4B GPIO Expansion Bd	1	10.00	10.00	1.39	1.39
2	12V Li-ion Battery	1	23.80	23.80	3.31	3.31
3	Adj. Support Legs	4	21.80	5.45	3.03	0.76
4	PVC Tubing (10 m)	1	10.00	10.00	1.39	1.39
5	WS2812B LED Strip (5 m)	1	48.75	48.75	6.77	6.77
6	7 Touchscreen	1	157.00	157.00	21.81	21.81
7	PWM Controller	1	13.22	13.22	1.84	1.84
8	KIPW25A-12L Water Pump	3	38.99	12.997	5.42	1.81
9	Acrylic Reservoir	3	70.00	23.33	9.72	3.24
10	ST-Link V2 Programmer	1	11.55	11.55	1.60	1.60
11	STM32F103C8T6 Board	1	9.10	9.10	1.26	1.26
12	LCD1602 Display	1	4.77	4.77	0.66	0.66
13	TP4056 Charger	1	1.80	1.80	0.25	0.25
14	USB-DC-DC Converter	1	2.07	2.07	0.29	0.29
15	Solar Panel	1	5.30	5.30	0.74	0.74
16	18650 Li-ion Cell	1	7.40	7.40	1.03	1.03
17	18650 Cell Holder	1	1.36	1.36	0.19	0.19
18	PCB Fabrication	1	10.00	10.00	1.39	1.39
19	Wind Turbine	1	15.80	15.80	2.19	2.19
20	Raspberry Pi 4B	1	320.00	320.00	44.44	44.44
21	Portable Wi-Fi	1	29.00	29.00	4.03	4.03
	Total		**811.71**		**112.74**	

Appendix C

Physical Prototype Photos

This appendix presents photographs of the physical prototype constructed for the Carbon Emission Tracking System project. The images illustrate key features, subsystems, and overall assembly of the system.

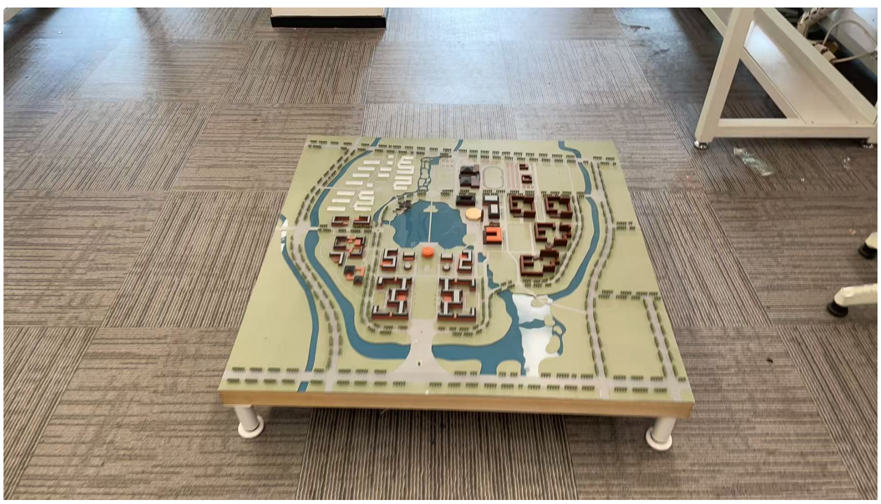

Figure C.1: Overall view of the completed sand table prototype, showing LED visualization, water flow, and mechanical lifting modules.

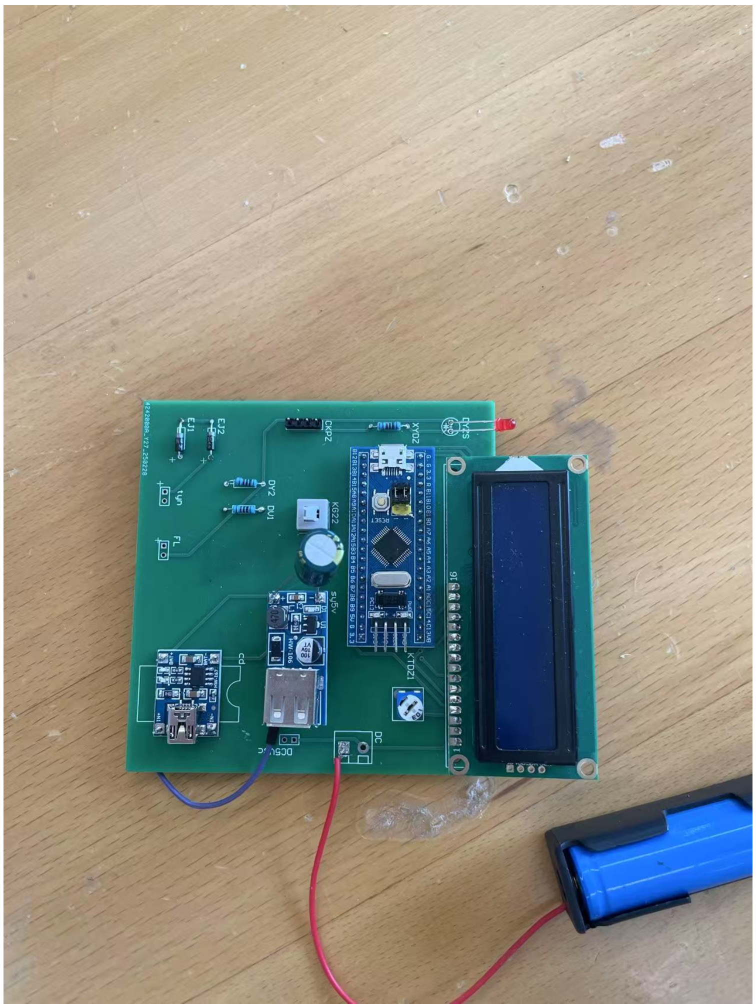

Figure C.2: Close-up of the Green Energy Subsystem, including the solar panel, wind turbine, and battery module.

fig_touchscreen.jpg

Figure C.3: User interface on the 7-inch touchscreen for data query and visualization control.

fig_detail_led.jpg

Figure C.4: Detail of the individually addressable LED modules corresponding to campus buildings.

fig_waterflow.jpg

Figure C.5: Visualization of water consumption using colored liquid and air bubbles in transparent tubing.