Solar Garden Data Collection, Analysis, and LED Signage

Logan McInnis
Bryan Nearing
Tyler Thompson

Advisor and Sponsor: Dr. Brad Bazuin
ECE 4820 Electrical and Computer Engineering Design II
December 6th, 2019

Abstract

The Educational Solar Garden at WMU has been lacking both the real-time, continuous data analysis of the available weather and solar data collected and a means of displaying current system performance. This project has implemented a single server and database where solar and weather data is collected and new software performs solar data analytics. The resulting information is then directly displayed on an LED information sign to inform the public about the real-time performance of the Solar Garden. The outcome of this project greatly enhances the ability of WMU faculty and students to study the behavior of our educational solar arrays while also providing highly visible public information on their performance.

Table of Contents

Summary

The Solar Garden Data Collection, Analysis, and LED Signage project has a successful and efficient database. This database will store data collected from the solar garden and has protocols to collect lost data in case of network errors. There is a web interface that allows users to reprogram the sign through WiFi. The sign itself was sealed up with caulk and made the HDPE prevent water from entering the system.

Introduction

The Solar Garden Data Collection, Analysis, and LED Signage project will provide a data collection server for the WMU solar garden. This server will store and calculate information that can be used by students and faculty to perform data analysis on the solar array. The database will also communicate with an LED sign that will be programmed to display information and messages to the public. This information can be uploaded through an online interface. The database is meant to be used by all of WMU and the sign can be seen by all Parkview visitors.

Discussion

Overview

Background

Western Michigan University has a solar garden. This solar garden was built by Consumers Energy as part of a grant deal with Western Michigan University. They also provide a grant that the Electrical and Computer Engineering department can use for educational purposes as well as the development of solar-based projects. Dr. Bazuin proposed a project, for students of the Electrical and Computer Engineering department, that would build off of the WMU Solar Garden. The plan for this project is to help WMU students understand the performance of solar panels in Michigan. This project will also help with the research and development of solar-based projects in the area. The *Solar Garden Data Collection, Analysis, and LED Signage* project is a solar-based project that records data from Western Michigan University's solar garden into a database for future research. This data will also be displayed on an LED sign to show the solar array output.

Figure 1. LED Sign Logo

Need Statement

The purpose of the *Solar Garden Data Collection, Analysis, and LED Signage* project is to provide Western Michigan University with a tool to perform data analysis on the Solar Garden. This project should also give the visitors of WMU some insight into the Solar Garden. This project will be done by recording solar array data into a database and output this data onto an LED sign. This database is needed for students and faculty that want to perform analysis on the solar garden.

System Diagram

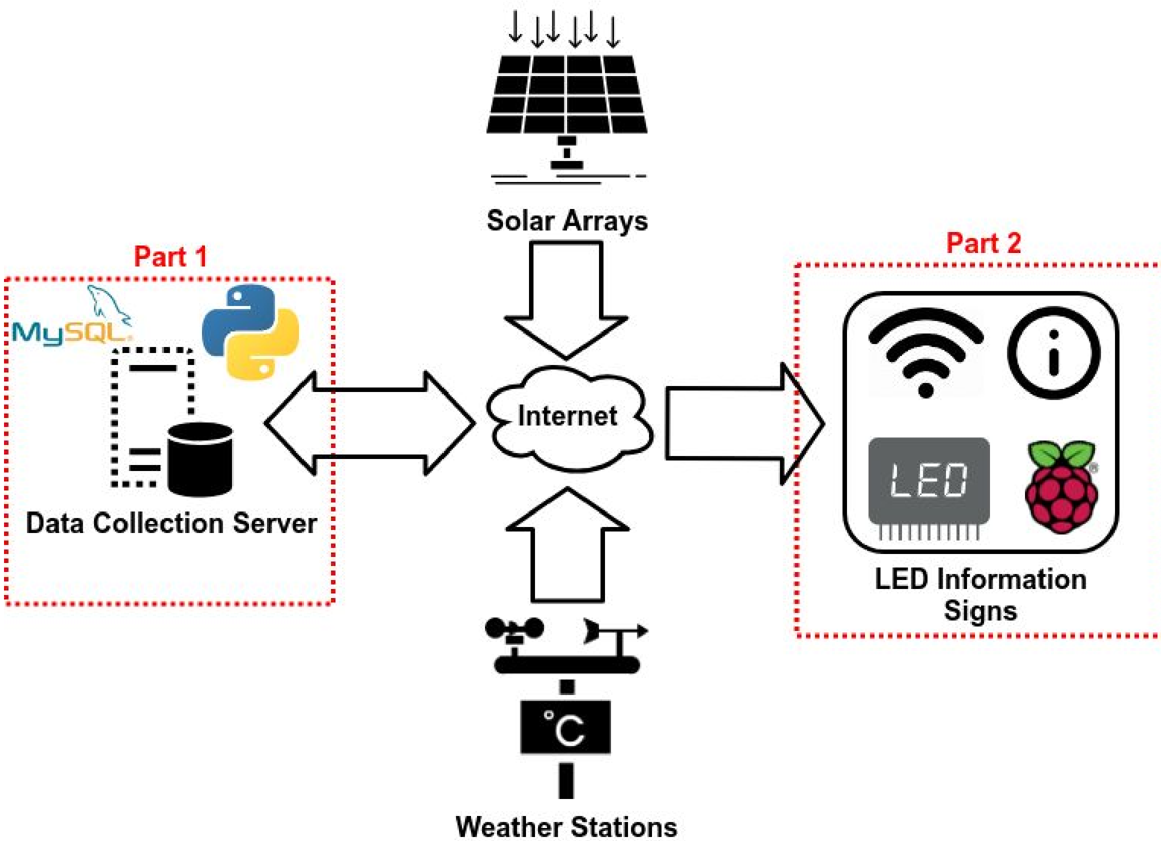


Figure 2. Block Diagram

Specifications

Physical Characteristics

LED Signage

- *The sign needs to be clearly visible from the road next to the solar panel*s: We plan on mounting the first sign outside next to the Solar Garden. Its placement and size need to be such that it can be seen and easily read by people driving by.
- *Weatherproof*: Everything about the exterior of the sign needs to be able to withstand the elements for at least 10 years.
- *Reproducible*: The entire construction of the sign can be reproduced easily and placed at different locations.
- *The sign needs to run off of a regular NEMA outlet which is weather-resistant*: A power cable will need to be run to the sign-in some fashion. The power cable will connect to the sign with a connector that is rated for outside weather.
- *Uses LEDs*: How information is displayed to the public will be through the form of an LED display.

Functional Characteristics

Data Collection Server

- *The database needs to collect as much useful information as possible*: Any information that is available from a data source will be downloaded into the database. Some of this information will not be useful for displaying on a sign but will be useful to research and other projects that need data from the Solar Garden.

- *Additional data sources can be added at a later time*: The data server is set up in a way that allows for additional data sources (solar arrays and weather stations) to be added at a later date. New data sources will be able to be added to locations so that new information can be automatically displayed on the signs.

LED Signage

- *Reconfigurable*: Information that is displayed on the sign should be relevant to the location where it is placed. There will be a way to configure the sign's location so that it can be moved or multiple signs placed at different locations.
- *The sign needs to be able to communicate over a Wi-Fi network*: The controller for the sign will communicate with the database and download information through a Wi-Fi connection.

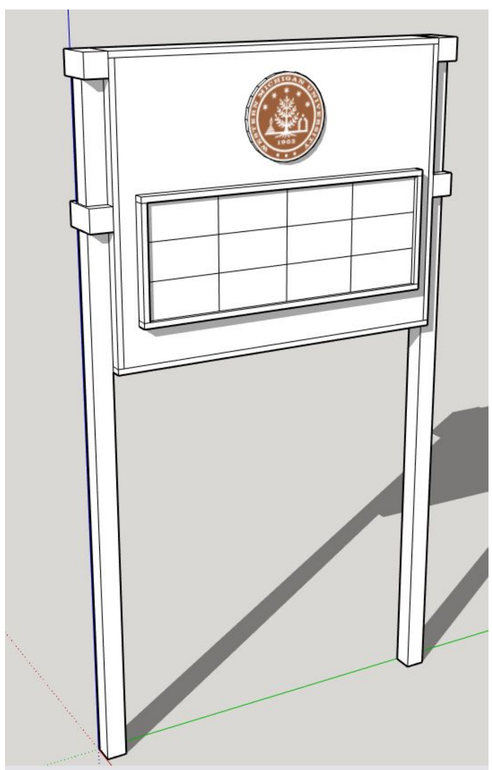


Figure 3. Proposed Sign Design

Design and Implementation

Data Collection System

In order to retrieve data from the solar arrays and weather stations (data sources), a custom Python program has been written. The data from each source is downloaded through a HyperText Transfer Protocol (HTTP) request and then loaded into the database using the MySQL protocol. Python has pre-built libraries for interfacing with HTTP requests and MySQL database connections. The program is designed to require two functions to be written for each data source. One to download the raw data, and one to format the data so that it can be written to the database. This design will allow us to handle each data source generically and added more data sources on the fly easily. This custom program is running on an Ubuntu Server 18.04 virtual server setup just for the project and is hosted by the CAE Center.

Database

The database was set up on the virtual Ubuntu server using MySQL. The design of the database was done using MySQL Workbench, which is a tool for creating and modifying databases. MySQL Workbench allowed us to create a database template (schema) and plan out where the data will be stored and built-in functions of the database. Each data source has an associated table and stored procedure that handles updating the corresponding table. These stored procedures are called by the program when updating data for a data source and will be able to keep the program from making duplicate entries. Each data source table also has a trigger set up that updates a separate table with the latest timestamp of the last piece of information collected. Each data source has a different time interval between data points, so it is important to know where the database left off on each entry. This way the program will know the time range in which it needs to pull data. The implication of this setup is that we are able to turn the server off for maintenance and when it comes back on, it catches the database with the data it missed while it was down. Additionally, using stored procedures to update tables allows us to do calculations, such as peak values, at the same time we update the data. The calculations are done on the database level which is much more efficient than doing it on the program level.

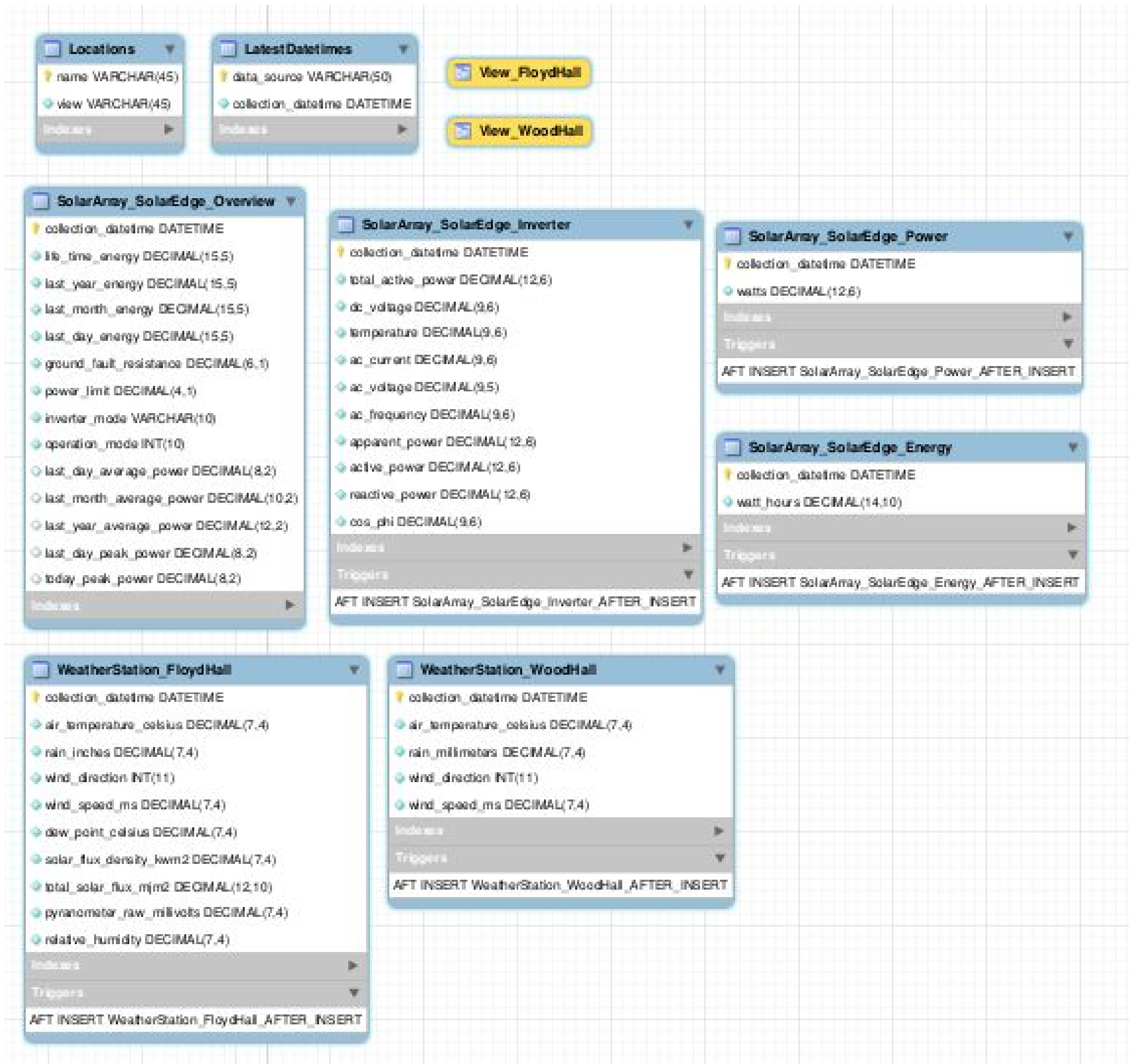


Figure 4. MySQL UML

LED Web Interface

The core of the controller that is driving the LED modules is a Raspberry Pi. On the Raspberry Pi, two processes are running, a web interface, and an LED controller. The web interface hosts a web page that allows the user to make modifications to a configuration file, while the LED controller reads the configuration file and drives the LED modules. The web interface is written in Python using a library call Flask. Flask is a lightweight web framework that is well suited for small applications that need to run on small systems. The web page allows the user to configure text, images, and collected data to be shown on the LED display. Data from the database is synchronized to the configuration file allowing the user to choose the data points they want to display and how they are formatted on the LED display all from the web interface.

LED Modules

The LED display on the sign is comprised of multiple P10 LED modules. These modules have a pixel area of $10\,mm^2$ and are 16 x 32 pixels in dimension. There are 3 rows and 4 columns of LED modules for a total of 12 modules. Each module is sealed to the sign with a rubber gasket and 12 screws. Figure 5 shows what one of the P10 modules looks like.

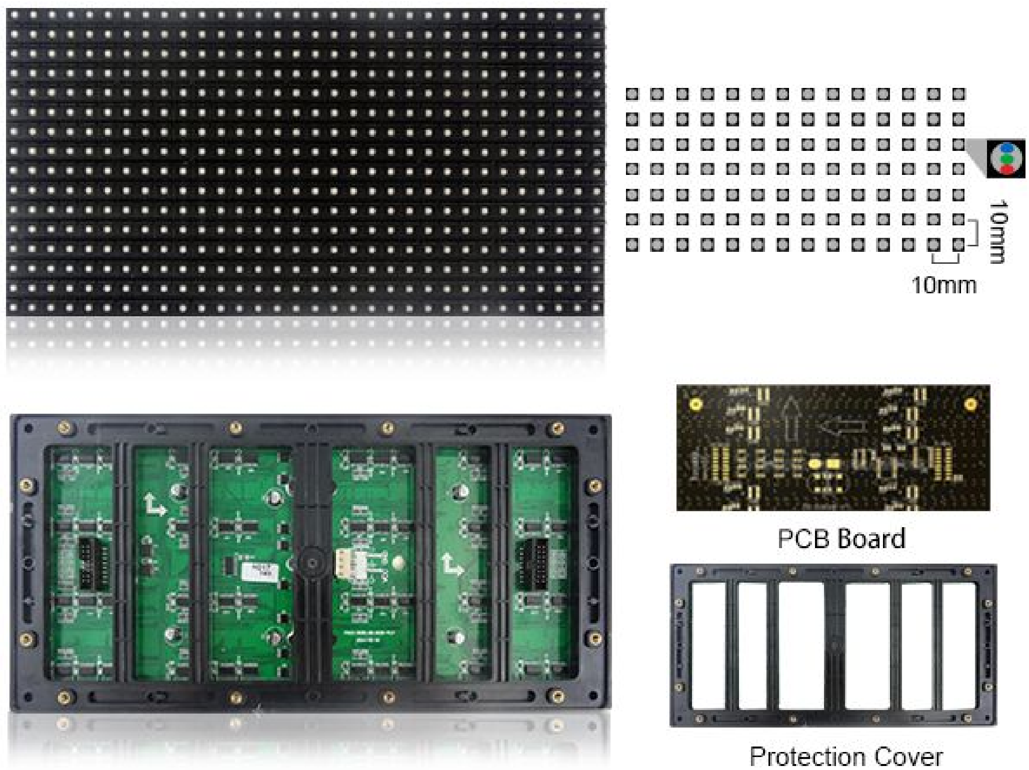


Figure 5. P10 LED Module

LED Controller

The controller that runs the sign is built using a Raspberry Pi 3 Model B+ microcontroller. The Raspberry Pi drives each row of the LED modules independently with buffers between the Raspberry Pi's GPIO pins and the LED module's shift registers.

Figure 6. Raspberry Pi 3 B+

When the sign doesn't have a connection to the internet, it defaults to displaying the current time and date, as well as the most recently acquired data from the database. In order for the controller to keep accurate time while not on the internet, a Real-Time Clock (RTC) is attached to the Raspberry Pi through an I2C interface. We are using a DS3231 chip with a small lithium-ion battery attached directly that will keep the time when the sign is off [7]. One version of the DS3231 can be seen in Figure 7.

Figure 7. DS3231 Real Time Clock

The controller of the sign communicates with the data collection server using the wifi network already in place at WMU in order for it to meet the communication specification. WMU's wifi network supports many different standards, but we are using the 802.11n standard. One of our standards is the ability to connect to the sign over a wifi connection. The 802.11n standard is the wifi standard that will allow us to fulfill this specification. This standard is widely used by devices today and supports multiple frequency bands and auto-negotiation. These features allow the controller to have a more reliable connection over a longer distance. We have achieved this by using a Raspberry Pi 3 microcontroller for the sign controller which already implements the 802.11n standard in its internal wifi chip.

Figure 8. WiFi Adapter 600mbps

Because a specification of the sign is that it needs to be waterproof, the controller is encased in a weatherproof housing. The NEMA Type 4 enclosure can withstand rain, sleet, snow, splashing water and hose directed water (see Figure 9). This provides protection from the elements as well as from the maintenance crew that maintains the grass areas. Additionally, the enclosure is plastic so it does not degrade the wifi signal of the controller.

Figure 9. Plastic NEMA Type 4 Electrical Enclosure

Power Supply

The power supply we used was a Mean Well 200W power supply. This power supply is capable of outputting 28 Amps at 5 Volts in the current configuration. When we were selecting this power supply, we were using the assumption that each LED panel uses about 2 Amps at full load. With each panel using 2 Amps and our sign having 12 panels, that is about 24 Amps at full load. We also budgeted about 2 Amps for our Raspberry Pi LED controller so that put us under our 28 Amp budget by 2 Amps, which we feel like is plenty of room for our LED sign to operate safely.

Figure 10. Mean Well 200W AC to DC

Sign Construction

The materials that our sign uses are primarily High-Density Polyethylene (HDPE), pressure-treated 2x4's, and screws and bolts to hold it together along with clear caulk. The sign consists of a wooden frame in the center of the sign that everything is connected to. The wooden frame has eight individually cut HDPE sheets that are screwed to the wood in multiple places. Inside the frame is wiring that is connected to both the LED panels and the NEMA box on the outside of the sign. To seal the sign, caulk is placed on the inside edges and the seams on the outside so that there are virtually no spaces where water or debris could fit into the sign. The LED panels have a rubber gasket that when tightened to the HDPE sheet creates a waterproof seal around the LED. We believe that we have taken appropriate steps to stop water from getting into the sign and in some cases, we have shrink-wrapped wiring and wiring harnesses to make sure that even if water gets into the sign, it should be able to correctly function anyways.

Performance Testing and Analysis

Testing Data Collection System

Initial testing of the data collection system was done by testing each individual data source and analyzing the output. Some sources have special cases that needed handling, such as the Solar Edge Solar Array. SolarEdge has a maximum time frame we can pull data for, so if the collection program needs to pull data for a large time frame, it does it incrementally. We then were able to start pushing the collected data into the database. Viewing the tables after an insert confirmed the functionality of the stored procedures and triggers that were set up. Final testing was done by taking the server offline for several days and allowing it to catch itself up when it turned back on. We were also able to prove that the server can be offline for two months and still catch itself up by modifying the database to make the collection program think it had been offline for two months.

Testing Database

Testing the database was done in parallel with testing the data collection program. Watching the state of tables while inserting new data allowed us to confirm the operation of the built-in functions we put in the database. Stored procedures and triggers were mainly tested this way. Once the initial data was in the database, we could then test the views for reading key information back. Since the views are how the LED controller interfaces to the database, it is important to test that they are generating accurate and well-formatted data. A view can be tested just by running a SELECT statement.

Testing LED Web Interface

Since the web interface does not interface directly to the LED modules, testing and development were initially done on a separate system than the LED controller. After each iteration of the code, the configuration file is verified by hand. When we got to a point of completion, the configuration file is then tested with the LED controller program to verify what the web interface is writing to the configuration file is readable by the LED controller. As the project started to take physical shape, continual testing was done with the web interface to ensure the interface is easy to use and the LED display is updating correctly with changes made by the user.

Testing LED Modules

The LED modules were tested using a small prototype which was representative of the final product. We used all of the same equipment that we would be using for the full-scale sign which also helped with our verification. To test the modules, we started with displaying text. This text was used to make decisions on the font sizes that we would want to use. The prototype also displayed some ghosting of the text on the sign so improvements to how we interface with the driver were coded so that we could better display information. After testing the text, we tested the ability to display images. When we displayed images, we discovered some noise from the images which was unrelated to the code we had written. This noise was eliminated by using a hamming code error correction filter which filters all of the noise from the images we were displaying. After we tested images and text, we were confident that the panels could display anything that we would need to display in order to convey the correct information to anyone looking at the sign.

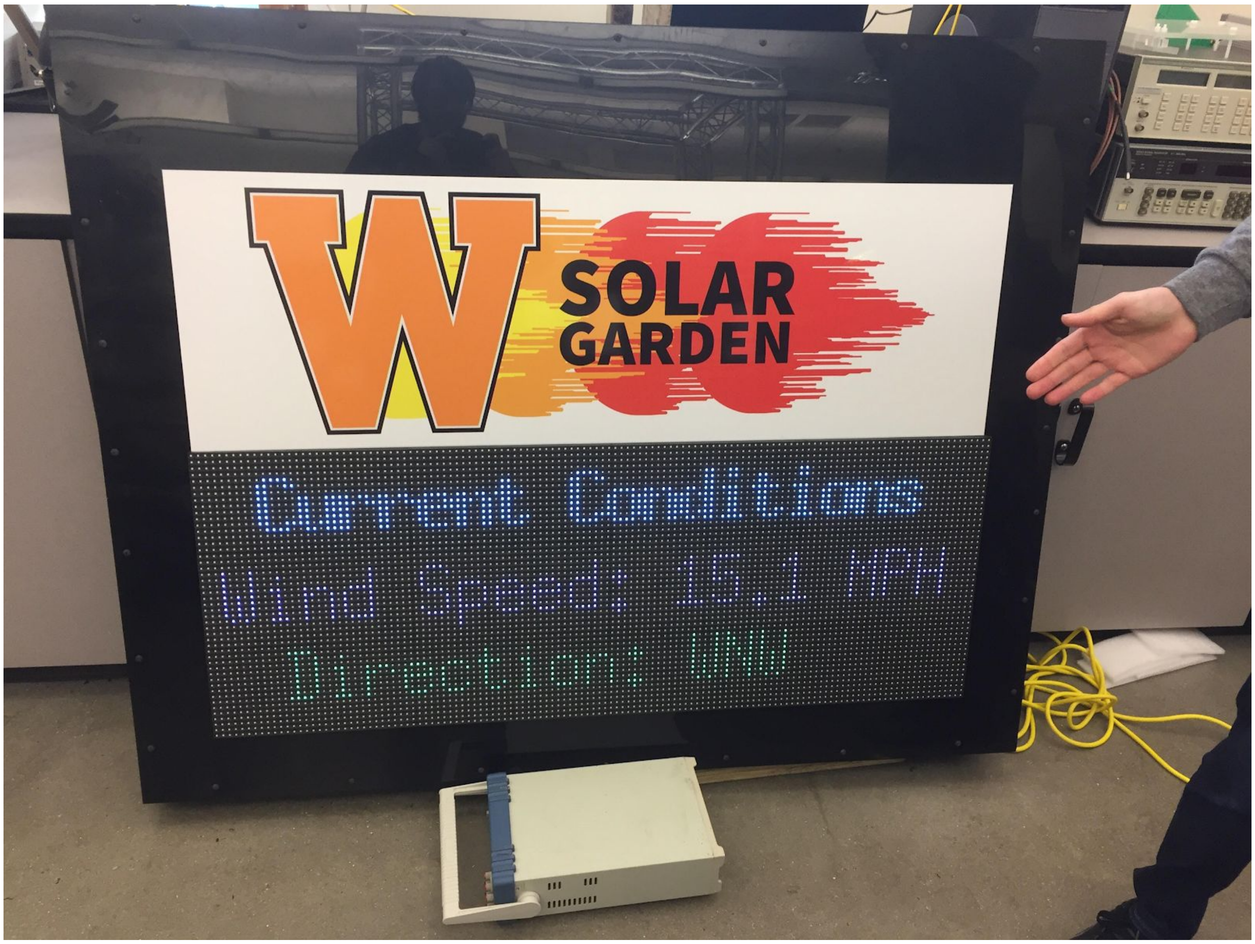


Figure 11. LED Sign

Testing LED Controller

After the LED controller was fully soldered and put together, we needed to make sure that every part of the controller was properly connected to the correct pins and also that each male connector on the board was outputting the correct signals for each pin. Another task was to see how much noise was being added along with each connection. These criteria were verified using an oscilloscope and digital multimeter. The oscilloscope sent pulse width modulated signals down each connection and the noise was measured. The noise was found to be within acceptable levels, although we may use capacitors in the near future to reduce the noise levels. The digital multimeter was used to measure voltage levels in the circuit. It was also used to test connections between the inputs and outputs. These voltage and connectivity tests all came back ok so we moved forward with our now verified PCB board.

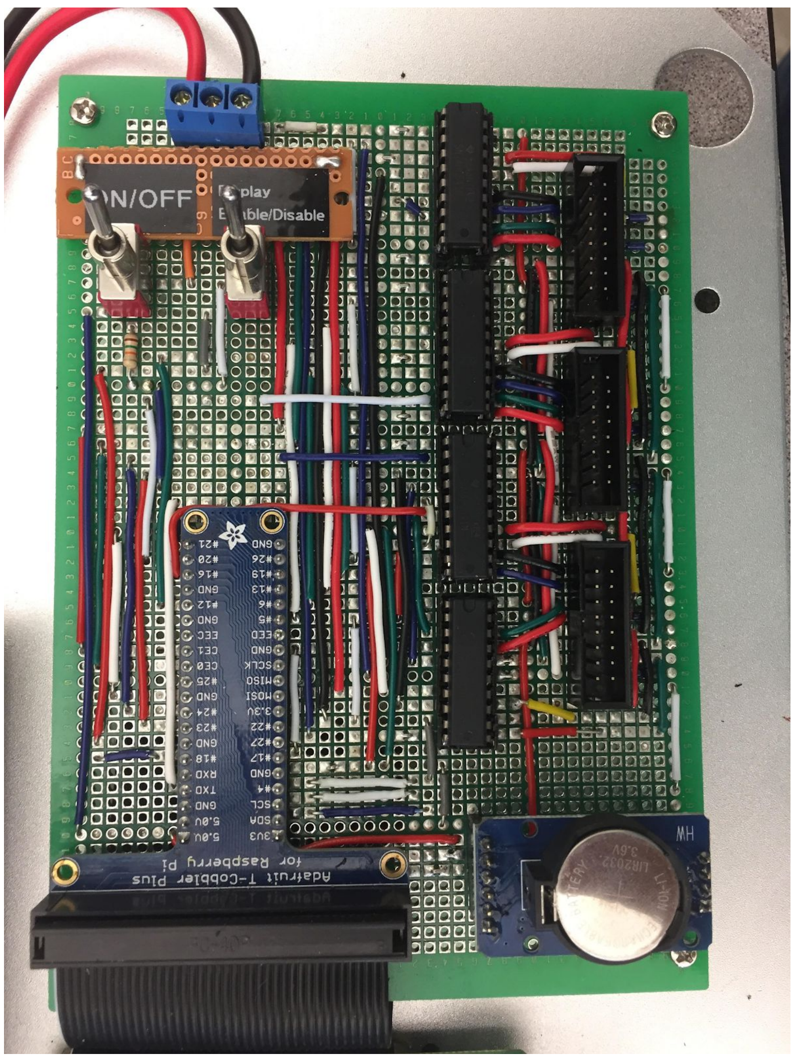


Figure 12. LED Controller

Testing Power Supply

When we were testing the power supply, we had to find out what was the most intensive workload that we could give it. It turns out that because the LED technically has 3 LEDs, a red, green, and blue LED, The most power-intensive point of the LED's operation is when all three LEDs are on at the same time with full power, which is when it displays the color white. Knowing this, we decided to test our power supply by setting all twelve panels of our sign to the color white at full brightness to see if we could breach our power limit but after testing multiple times at various lengths of time, we never had a power supply failure or interruption. We believe at this time, barring some sort of large electrical spike, the power supply is good enough for operation in most if not all situations that the sign would reasonably operate in.

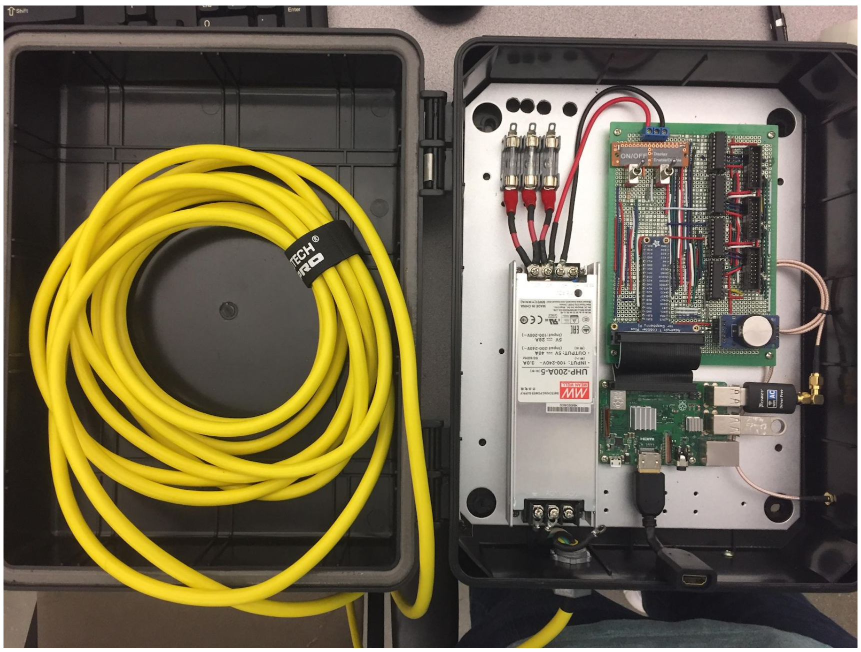


Figure 13. Controller and Power Supply

Conclusion

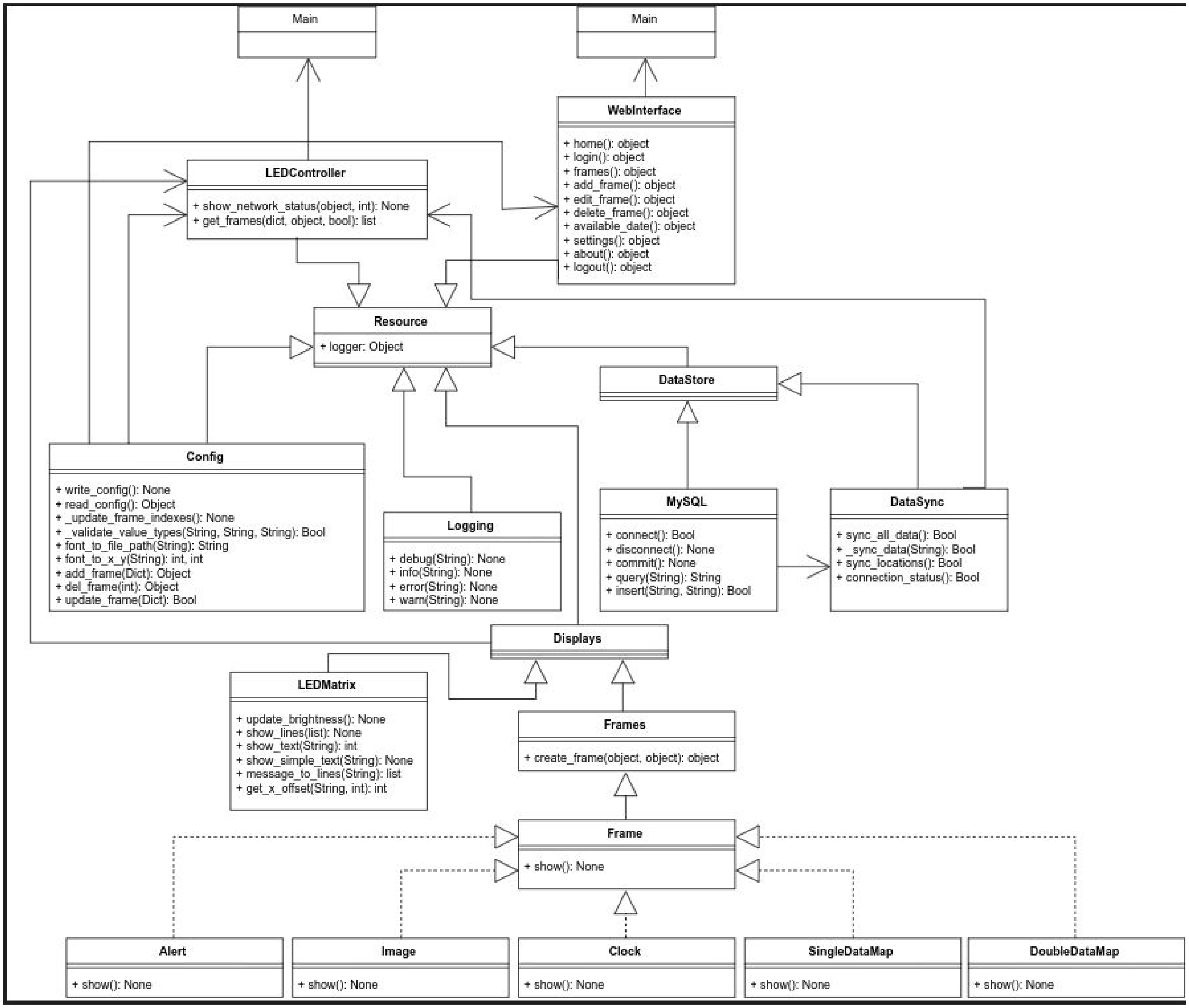
The project was successfully completed and implemented. The finished database provides solar garden data to anyone who gets access from the ECE department and the LED sign displays the database information to all Parkview visitors. This project met all the requirements that were stated in the proposal with one minor exception. This exception was the ability to mount the LED sign outside. Due to the school’s policy, signposts cannot be made without approval. This process is lengthy and must be complete by Dr.Bazuin. The sign is still made easy to mount on signposts so when Dr.Bazuin decides that he wants it outside, it will be easy to install. This project also finished well within our $1277.10 budget.

Recommendations

Although we believe the sign to meet all of the specifications outlined in our project proposal, there are a few ways we believe that the sign can be improved. First, we would change the sign materials. Although the HDPE works well and the caulking should keep the sign watertight, we believe a switch to aluminum sheeting and scaffolding would increase the integrity of these waterproof seals. The aluminum would also give us a more exact shape and structure than the plastic sheeting did. With aluminum, it would require welding to make sure all of the panels are in the correct place. In the initial design, we believed that welding was out of our skill range and would lead to delays in the project but after taking a step back these concerns were not as large as initially believed. The second change we would make would be to design the database differently. Currently, if any of the data sources go offline, the sign will stop displaying information because it believes the data is too old to display. In a redesign of the database, it would handle data sources independently so that we could possibly lose access to a data source and keep running. Another improvement we would make is that we would go with a more streamlined PCB board that has a much smaller footprint than the prototype board we bought earlier in the process. This would give us much more room in the NEMA box and would possibly allow us to buy a smaller box, saving money in the process. In regards to the NEMA box, the box isn’t able to fully open. This is due to the mounting that the sign has currently. In a revision, we would first think about how to mount the sign earlier in the design process. This would do two things for us. First, it would make our mounting design much more elegant. Second, it would give the NEMA box the room it fully needs. Although the NEMA box is not inaccessible in its current form, these changes would allow everything on the sign to work together in tandem, not against each other like it is currently.

Appendix

Appendix A - LED Signage Program UML



Appendix B - Instruction Manual