Grand Plan Outline

Author: William Kerr

# Introduction:

Goals:

My goal is to create a solar-powered greenhouse. The requirements of the greenhouse are to store plants at a reasonable temperature, store a monitoring system inside, and power said monitoring system with rooftop solar panels providing power.

UCSC has 24 solar panels that provide 1.5kWhr of power per day. The power will be delivered properly using a MPPT Solar Charge Controller, provided by Morningstar. Each charge controller can only take 150V, and the total amount of solar panels we have sums up to 196V. Therefore, we will have two charge controllers working in tandem to charge the battery pack. These charge controllers can charge the same battery pack without interference, according to Morningstar. Both charge controllers will have a remote temperature sensor connected to it in order to monitor the amount of charge being delivered to the batteries.

The Charge Controller will deliver power to a battery pack. The battery pack cannot stabilize itself, so the battery pack will be controlled by a Battery Management System, provided by EMUS. That Battery Management system will control the charging of the Batteries using variables such as the voltage, current, temperature, and charge of each individual cell, and modify things such as the battery balancing rate accordingly. Each individual battery has its own board attached to it, provided by EMUS. It can monitor the current drawn from the battery, the voltage of the battery, and the temperature of the battery. The Battery Management system will calculate how much charge the battery has based on its own internal logic system. If the batteries get too hot, the battery management system will turn on a fan that cools off the batteries. If the batteries get too cold, the battery management system will turn on a heater that will heat up the batteries.

The battery pack will power the internal computer system inside the greenhouse and will also power an AC outlet somewhere in the greenhouse. The battery power will be conerted from DC to AC power using a power converter provided by Morningstar. It’s called the Suresine inverter.

The internal computer system consists of a Raspberry Pi main computer, located next to the Charge Controllers (for convenience), the 2G GSM shield attached to it, the faculty sensors (all will be I2C), the student sensors (will have various protocols), the Raspberry Pi slave computers, and an external server somewhere at UCSC.

The faculty sensors will consist of light sensors, and Humidity + Temperature combination sensors. Since I can choose which faculty sensors I will give to the client, I will choose only I2C sensors. I will connect all faculty sensors to the Main Raspberry Pi Computer. Therefore, I will design a daughterboard that only accepts I2C inputs (that are detachable), and will attach directly to the Raspberry Pi board.

The main Raspberry Pi Computer is very powerful for a tiny computer. It can run a Linux system designed for it, called Raspbian. I will install a touch screen display to it, so that if somebody needs to debug the computer, they can see what they’re doing. I will also have a case for it to protect the computer from the elements. The main computer’s tasks will consist of: Reading the sensor values from the faculty sensors (using the I2C protocol), receiving a student sensor value package from the slave computers from around the greenhouse over the Bluetooth protocol, reading and storing the values generated from the Charge Controller’s internal PLC, reading and storing the values generated from the Battery Management System’s internal logic system, compiling all of that data, and then sending it over a 2G TCP connection directly to a server somewhere at UCSC.

The student sensors can consist of almost anything. Here are some possibilities: Water temperature sensors, Soil moisture sensors, light sensors, humidity and temperature combination sensors, pressure sensors, etc. Since it is impossible to know what protocol the students will like more, I will incorporate the most popular protocols, including I2C, SPI, UART, and 1-wire. I will incorporate a lot of UART ports, so that way if their sensor uses a strange protocol not listed here, they can make a bridge from that protocol to UART and it will work.

The slave Raspberry pi Computers will also be powerful, albeit smaller than the main computers. They will be capable of interacting with the Bluetooth protocol. They will also be capable of reading sensor values over a various number of protocols. Their main tasks will be to gather data from every student sensor they are connected to. Then, they will compile that sensor data and send it to the main Raspberry Pi computer with the Bluetooth protocol.

The Main Raspberry Pi computer will be sending a packet of data to a UCSC server somewhere. Once that server receives the data, it will store it in a SQL-style database. That server will also be connected to the internet. If a user connects to this server, the server will serve a website back to the user. The user can then use that website to send a request to the server asking for student sensor data, faculty sensor data, Battery Management data, or Charge Controller data. The user can add a format to their request, such as a graph over time, instantaneous data, data from the last time the computer has polled, data from a specific point in time,

If the user requests a graph over time, they will need to specify what type of data they are requesting, Y-Axis units (i.e. V, mA, etc.), X-axis data (i.e. units of time), and range and scale for both X and Y axis for each dataset they request. Optional: a caption, a title, names for the X and Y axis. The server will return a webpage containing their graph, and some graph metadata: how many data points are returned, what the scale and range are, and a legend if they requested multiple data sets.

Key:

Goal

Solar Panels

Charge Controller+Battery Management

Battery Pack + Load Delivery

Internal Computer System

Faculty Sensors

Main Microcontrollers

Student Sensors

Slave Microcontrollers

Website

Request Format

# Sources:

<https://www.morningstarcorp.com/parallel-charging-using-multiple-controllers-separate-pv-arrays/>