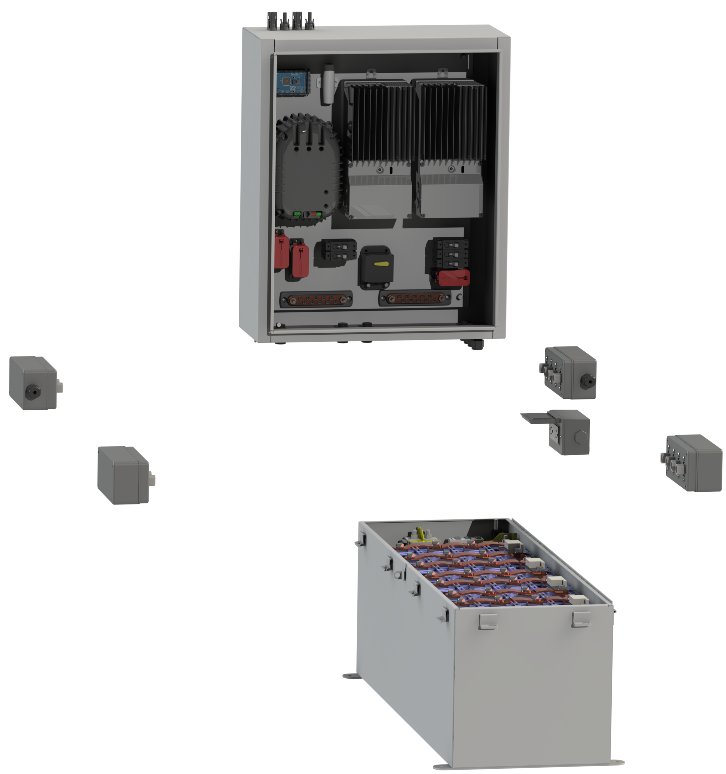
Arboretum Micro-Grid Documentation and Operation Manual

Retrofitting, redesigning, and optimizing the existing greenhouse solar microgrid at the UCSC Arboretum

Isaak Cherdak, Kurt Ringer, Michael Delorio

June 2017



Contents

1 Introduction

1.1 Our Project

The University of California, Santa Cruz's Arboretum has a very neglected micro-grid with practically non-existent management. This has enabled reconfiguration by those who don't understand the intricacies of their actions, and hence encourages increasing complexity of the system. The grid itself is in a terrible state of disarray, featuring unfortunate developments such as random, hanging live wires, batteries more than 5 years expired, inadequate power for more crucial items, and plenty of broken equipment with obfuscated wiring, that still sit in the same housing as the working equipment. We are the team tasked with amending this poor facility's design once and for all, with great advancements such as a completely new design, fool-proof safety, and guidelines for future expansion. We hope this document elaborates on the dire need to make changes to the grid as well as documents what changes we recommend and plan to implement.

1.2 The Team

1.2.1 Isaak Cherdak

Student at University of California, Santa Cruz majoring in Computer Engineering and Computer Science. Was in charge of the design of Sensor Node versions 2 and 3, as well as the data visualization system for the grid.

1.2.2 Kurt Ringer

Student at University of California, Santa Cruz majoring in Computer Engineering. Was in charge of solar and power scaling of the system. Additionally, worked on system design and integration.

1.2.3 Michael Delorio

Student at University of California, Santa Cruz majoring in Robotics Engineering. Was in charge of the electrical and mechanical design of the microgrid, creation of CAD documentation for the system, and general documentation for the team.

1.3 This Document

Due to the fact that we are continuing to work on and expand this project beyond Senior Design, please understand that this is a living document. Sections which are left blank are for planned future work.

2 Previous Micro-grid

2.1 The Previous System

The previous microgid, Figure 1, is comprised of solar panels for power production. The power they produce is controlled through Morningstar solar charge controllers. These charge controllers communicate through a Morningstar MeterHub. The charge controllers are used to charge a parallel array of lead acid 12V batteries. One load is two 12V DC ventilation fans. These are switched through a contactor which is controlled by a Morningstar Relay Driver. The Relay Driver receives data from the Meter Hub on battery SOC to decide when to switch the fans on or off. The other load is a 400W Inverter which outputs 120VAC at 60Hz. This powers various loads for the aquaponics system such as air pumps and water pumps. This inverter isn't controlled from a contactor and is manually switched on and off by the users.

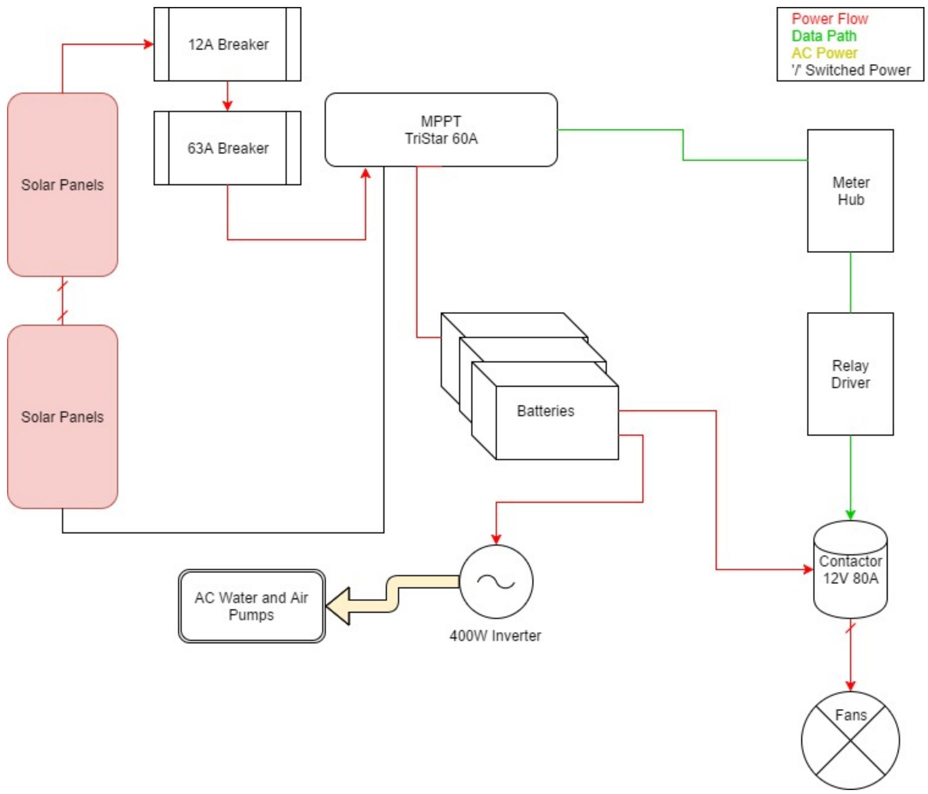


Figure 1: Current System Block Diagram

2.2 Solar Panels

These solar panels, branded LUMO and produced by Soliculture, have non-continuous photovoltaic cells which allow light to pass through the clear glass panels so that plants can still harvest a majority of the light. Additionally they act as replacements for the glass roof panels of the greenhouse.

2.2.1 Panel Characterization

To characterize the panels we switched the panels off from all loads and disconnected the wires. Using a digital multimeter ,we then measured the voltage across an individual panel to give voltage open circuit (V_{oc}) and measured the current, using the multimeter's ammeter, through the panels under no load to give amperage short circuit (I_{sc}). The wattage figure, though approximate and not considering voltage drop and panel resistance is calculated by multiplying the two figures, $W = IV$. The actual value is approximately 70% of the calculated value based on an approximation given by an Industry reference, Glenn Alers.

Company: Soliculture

Panel Dimensions: 64" X 22"

Mounting: Panels slot into slides on roof and are held in place by an aluminium strap on the face that meets the wall.

Older Panels: Qty 22

$\mathbf{V}_{oc}$	$\mathbf{I}_{sc}$	**Power**
29.9V	0.22A	6.6W

Table 1: Measurements taken on a cloudy/rainy day

$\mathbf{V}_{oc}$	$\mathbf{I}_{sc}$	**Power**
31.3V	0.81A (West Side) 0.50A (East Side)	25.4W

Table 2: Measurements taken on a average sunny day

New Panels: Qty 10

$\mathbf{V}_{oc}$	$\mathbf{I}_{sc}$	**Power**
6.75V	2.4A	16.2W

Table 3: Measurements taken on a cloudy/rainy day

$\mathbf{V}_{oc}$	$\mathbf{I}_{sc}$	**Power**
6.84V	4.3A West Side	29.4W

Table 4: Measurements taken on a average sunny day

2.2.2 Panel Specs from Manufacture

From the data in Table 5 we can calculate the maximum power point(M_{pp}) $M_{pp} = V_{mp} * I_{mp}$ of the panels, which is the maximum power the panels can produce under ideal conditions. This data was provided by a contact from the panel manufacturers. This gives $\mathbf{M}_{pp}$ **= 68W** for the average panel performance.

$\mathbf{V}_{oc}$	$\mathbf{I}_{sc}$	$\mathbf{V}_{mp}$	$\mathbf{I}_{mp}$
8.00V	12.30A	6.30V	10.80A

Table 5: New panel data from manufacture

2.3 Batteries

The batteries used in the old system are Deep Cycle Sealed AGM Lead Acid Batteries. These batteries are rated for one year of use under a deep cycle application, such as the solar microgrid which they are used in. The Tristar Solar Charge Controllers which are being used in the old system are designed to work directly with lead acid batteries with integrated charging profiles. To achieve the total storage capacity for the system, the 12V lead acid batteries are in a solely parallel configuration, maintaining the total systems nominal voltage of 12V. This holds true in all of the various battery containers.

To characterize the state of charge (SOC) of the batteries the users are measuring the V_{oc} across a single battery. A reading of 13.2V corresponds to fully charged, and a reading of 12.4V corresponds to 40% SOC, or a highly depleted battery. At this value the batteries are considered deeply cycled and discharge should be stopped to protect their lifespan. These values hold true with most any style of lead acid battery, with different chemistries affecting the size and capacity of the battery. Though a stable and resilient option, lead acid capacities are severely affected by use after their lifetime, and tend to be less energy dense than most lithium-ion solutions.

2.4 Critical Analysis

The previous system is about five years old, and does not meet the current user's energy needs. It is not user friendly to use or maintain, nor is it well documented. In its current state it is arguably unsafe, having multiple examples of disrepair. There are loose wires, both simply disconnected and live, as well as disabled or bypassed system protection contactors and breakers. The inverter, which is currently the main load, is being switched on and off by the user based off of manual measurements, which in cases of human error can destroy the batteries. Additionally, the system does not meet the user's power needs, even operating at its full capacity. A primary issue is the usage of lead-acid batteries which are long past their expected expiration, reducing their marketed capacity. Finally, all mounting for the components leaves the system exposed to moisture, causing corrosion, and will only worsen when more aquaponics and aeroponics systems are added.

3 Solar Calculations

Integral to designing an islanded solar microgrid system such as this is assuring the system will meet the power demands of the user. To do this one requires studying averaged solar irradiance patterns for sizing the number of panels to achieve the required average energy production for the average worst month. This will ensure that the system will operate ideally in the least opportune months. Additionally, the batteries should be sized to give an operational autonomy margin which is suitable for the application.

The user provided a poor power budget, which failed to consider major loads such as ventilation fans. Additionally, we are unsure of the number of solar panels which we can obtain from the manufacturer, as these are custom dimensions. Due to these limitations, we have decided to approach the design from a standpoint which is somewhat ignorant of power demands. Because we currently possess 10 new panels, and are assured we can get 10 additional new panels through a partnership with the manufacturer, as well as 20 panels nearly filling the roof of a single greenhouse, we are scaling the system with units of 20 panels. This will allow for a single installation to serve a whole greenhouse with only 2 solar charge controllers, a large improvement compared to the previous set of 5. If a larger system is desired, the entire 20 panel microgrid system can be duplicated onto an additional greenhouse, and connected in parallel to the preexisting system.

Month	Solar Radiation ($kWh / m^2 / day$)
January	3.48
February	4.47
March	5.17
April	6.12
May	6.49
June	6.72
July	7.24
August	6.83
September	6.62
October	5.30
November	3.72
December	3.22
Annual	5.45

Figure 2: PV Watts Solar data for Arboretum site

Based on the solar radiation data in Figure 2 for January $3.48kWh/m^2$, and the solar panel specifications of 68.04W described in subsubsection 2.2.2, we can calculate the average daily output of the system. We are assuming the maximum power point figure comes from standard test conditions (STC) with $1kWh/m^2$ of irradiance.

$$\frac{68.04W * 3.48kWh/m^2}{1kW/m^2} = 236.78Wh/panel \tag{1}$$

$$236.78Wh/panel * 20panels = 4.74kWh/day \tag{2}$$

This gives the average amount of energy our 20 panels will produce per average day in January with 100% efficiency. Combining our charge controller efficiency of 94%, our charging efficiency of 99%, and our non-tracking loss of 30%[4], this gives our efficiency.

$$94\% * 99\% * 70\% = 63\% \tag{3}$$

$$4.74kWh/day * 63\% = 2.96kWh/day \tag{4}$$

Equation 4 Gives our final average January daily energy production of **2.96kWh/day** from the 20 panels, split 10 to each side of the greenhouse.

4 Proposed Micro-grid Implementation

The proposed system to implement in the arboretum greenhouses is a modular, sealed system, designed to function with low maintenance. We also intend to provide data on the usage and status of the microgrid both locally and remotely. The system will utilize lithium iron phosphate batteries in order to maximize the installation's energy density, in addition to creating a safe system with high longevity. The primary advantage to this chemistry is its stability, with an average of 2000 full cycles in its lifetime, or approximately 5 years if abusing the system with deep cycling. The main disadvantage of this chemistry, as compared to lead acid, is the need of a battery management system (BMS) to regulate the charging of the cells. The accumulator will be external to the main controller housing to provide ease of installation and flexibility in placement and replacement.

Specifications	**Lead Acid**	**NiCd**	**NiMH**	**Li-ion[1]** Cobalt	Manganese	Phosphate
Specific energy (Wh/kg)	30–50	45–80	60–120	150–250	100–150	90–120
Internal resistance	Very Low	Very low	Low	Moderate	Low	Very low
Cycle life[2] (80% DoD)	200–300	1,000[3]	300–500[3]	500–1,000	500–1,000	1,000–2,000
Charge time[4]	8–16h	1–2h	2–4h	2–4h	1–2h	1–2h
Overcharge tolerance	High	Moderate	Low	Low. No trickle charge		
Self-discharge/ month (room temp)	5%	20%[5]	30%[5]	<5% Protection circuit consumes 3%/month		
Cell voltage (nominal)	2V	1.2V[6]	1.2V[6]	3.6V[7]	3.7V[7]	3.2–3.3V
Charge cutoff voltage (V/cell)	2.40 Float 2.25	Full charge detection by voltage signature		4.20 typical Some go to higher V		3.60
Discharge cutoff voltage (V/cell, 1C)	1.75V	1.00V		2.50–3.00V		2.50V
Peak load current Best result	5C[8] 0.2C	20C 1C	5C 0.5C	2C <1C	>30C <10C	>30C <10C
Charge temperature	–20 to 50°C (–4 to 122°F)	0 to 45°C (32 to 113°F)		0 to 45°C[9] (32 to 113°F)		
Discharge temperature	–20 to 50°C (–4 to °F)	–20 to 65°C (–4 to 49°F)		–20 to 60°C (–4 to 140°F)		
Maintenance requirement	3–6 months[10] (toping chg.)	Full discharge every 90 days when in full use		Maintenance-free		
Safety requirements	Thermally stable	Thermally stable, fuse protection		Protection circuit mandatory[11]		
In use since	Late 1800s	1950	1990	1991	1996	1999
Toxicity	Very high	Very high	Low	Low		
Cost	Low	Moderate		High[12]		

Figure 3: Battery Characteristics

The main electrical panel will house most of the electronics of the system, in addition to our sensor node module, designed to facilitate the data acquisition of the microgrid and greenhouse. Finally, the power distribution system consists of four DC nodes housing four 12 V plug ports, and an AC node housing two 120V AC outlets. The line diagram for the system can be seen below, in Figure 4.

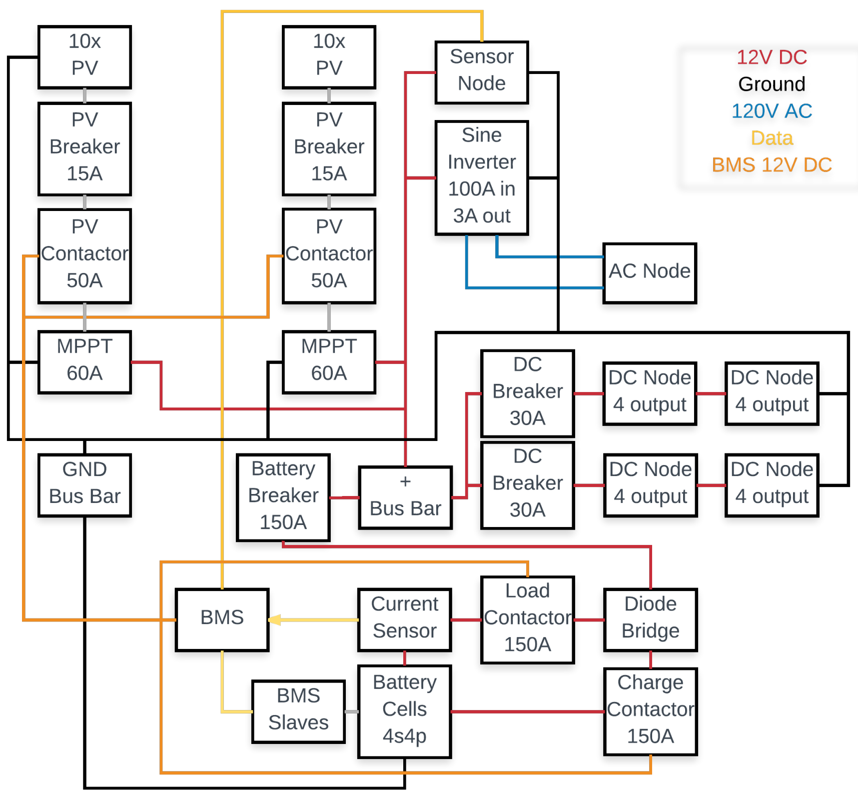


Figure 4: Line drawing depicting the full microgrid.

4.1 Solar System

We have decided to design the system assuming we have 20 or 40 new panels. If we were to have 20 panels in series we would get a V_{oc} of 160V which is above the

Tristar MPPT max of 150V therefore we will run 10 panels in series, requiring 2 charge controllers per 20 panels. This makes our maximum charge currents per charge controller about 60A, as seen in Figure 5. However, this set maximum charge current will depend on the capacity of the current accumulator, with a current of about 50A total throughput for each set of batteries installed in parallel, discussed further in the system limitations section. The above charging current calculation is based on the efficiency of the charge controller, battery charging efficiency, and input power.

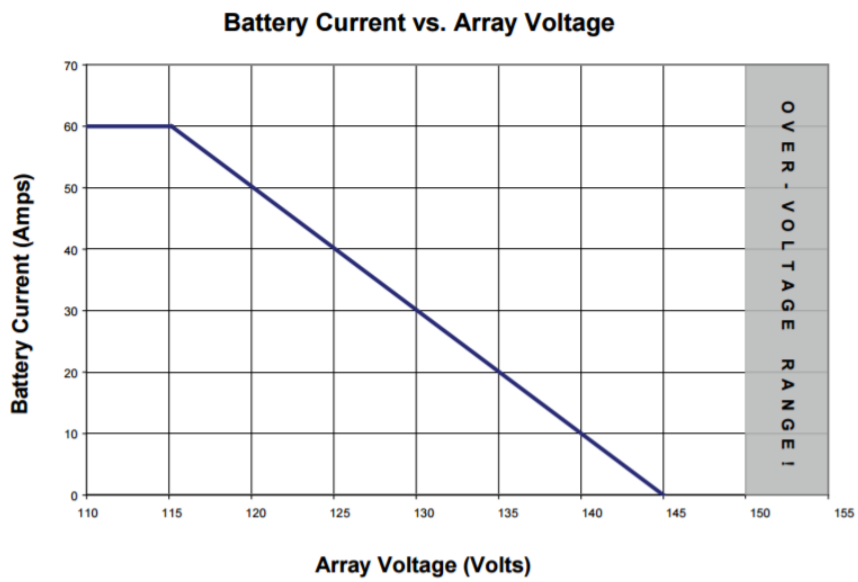


Figure 5: Battery Current vs. Array Voltage

We will size the batteries approximate minimum specification per unit of 20 panels. Based on a 3kWh daily output we would want to shoot for approximately 250Ah batteries, justified by the calculation $\frac{3,000Wh}{12V} = 250Ah$.

4.2 Control Panel and Solar Charge Controllers

The main panel of the system houses the Morningstar Tristar MPPTs and Suresine inverter, in addition to the custom Sensor Node data acquisition system, and acts as a hub for interconnecting the system's components. The housing for the main control panel is a NEMA 4 rated electrical panel, ensuring moisture protection, with wall-mounting provisions for ease of access and cooling. The majority of the space is taken by the aforementioned Morningstar equipment, making sure to arrange the components according to the airflow guidelines noted in the MPPTs documentation, as seen in Figure 6.

Figure 6: Main panel containing MPPTs, sine inverter, and wiring accessories.

Below this equipment is the breakers and contactors for the photovoltaics, wired into the PV contacts on the MPPT. The other end of this circuit is a set of panel mount PV connectors. Additionally, another set of breakers is mounted below the MPPTs to provide protection for the distribution system, with cable glands on the bottom of the housing to wire this system, which can be seen in Figure 7.

Above the MPPTs, the Arduino stack-up for the Sensor Node data acquisition system is mounted, connected to a panel mount antenna to facilitate 2G cellular data for remote data visualization. All systems are wired to 2 large bus bars, for the common power and ground of the system, using a 150A breaker to protect the battery itself. Finally, the Morningstar Suresine inverter will also be fused on the input and output of the module, with an additional control switch mounted to the bottom of the housing to toggle the AC power. Proper care will also be taken to establish an earth ground for this AC system.

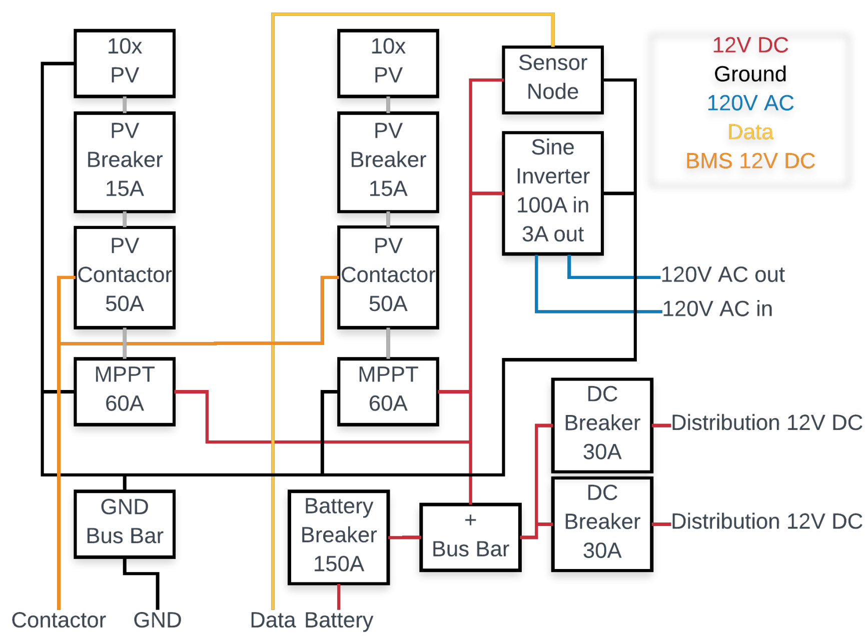


Figure 7: Main Panel Diagram

4.3 Accumulator

The accumulator for this microgrid consists of an energy storage solution, BMS to manage the system, and hardware with which the BMS manages the system. The housing for the accumulator is a NEMA 4 rated electrical trough, allowing for a ground mounted accumulator protected from moisture ingress. Within this enclosure, a set of acrylic holders for the battery cells are arranged at the bottom of the housing. These serve to support and align the battery cells for wiring in the modular system. An overview of this system can be seen in Figure 8.

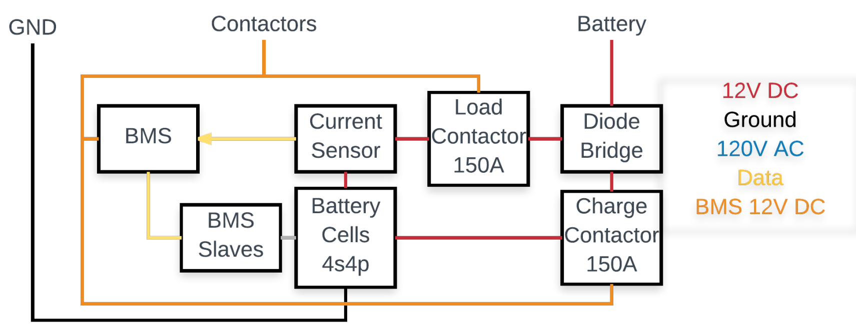


Figure 8: Accumulator Diagram

The accompanying electronics for the system are based off of e-Lithion's, a BMS manufacturer's, charge-discharge isolation circuit, as seen in Figure 9. This circuit uses a series diode bridge to isolate the voltage on the positive lead of the battery as charging and discharging paths. These paths are individually controlled using high amperage contactors, allowing for granular control of the charging and discharging of the accumulator. Additionally, a contactor is included on the photovoltaics of the system, to more efficiently control the charging of the system. These contactors are driven by configurable outputs of the BMS. Additionally, these contactors will be able to shut down the system in the case of a fault, such as a severe transient. In the case of our accumulator, the EMUS BMS was chosen, as it offered superior packaging and pricing compared to the aforementioned e-Lithion. The style BMS chosen is a distributed system, with slave boards on all series cells, allowing for ease of replacement and maintenance, while minimizing the size of the master unit. These slaves are connected to each of the series cells, tapping both terminals of the cell to determine its voltage, and allowing it to shunt voltage across it in the case of a full cell, in order to balance the series cells of the accumulator. The BMS also uses a transformer-style current sensor to determine the current of the battery, for more accurate state of charge estimates. The CAN communication lines for this BMS are sent out of the box through a cable gland, in addition to the contactor control line, to the main panel.

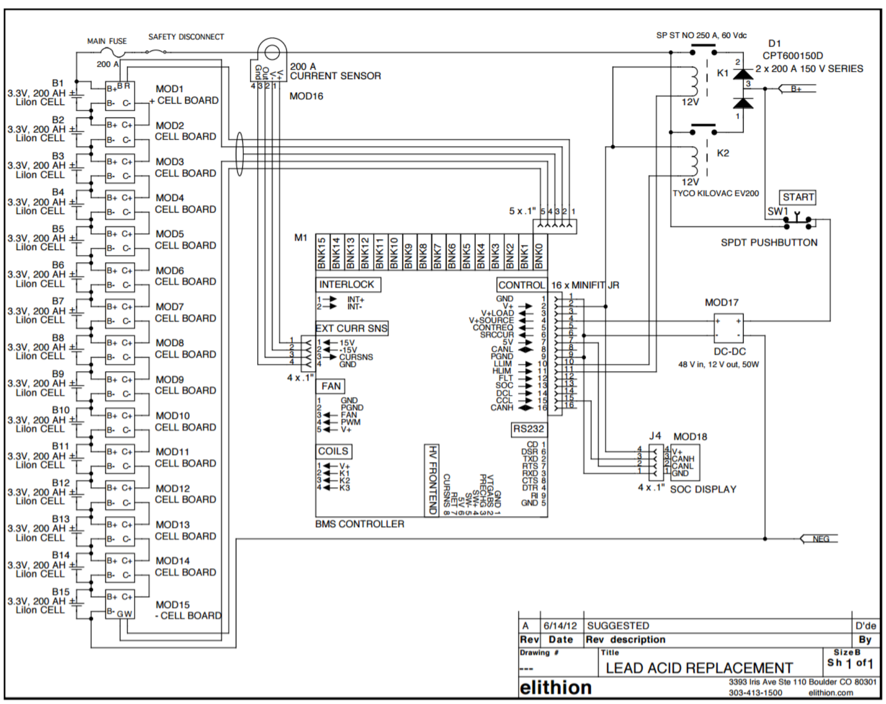


Figure 9: BMS accessory circuit. Notable is the load separation method using a diode bridge, featured in the top right

The general battery charging solution is not amperage regulated, switching the battery charging circuit off to allow the cells to balance. Though not the quickest method, this provides simple and robust integration with the charge controller. The BMS will switch the charge contactors off when a sufficient charge has been met, while the predefined curve on the charge controller will be a simple constant amperage and voltage cutoff. In the case of low charge, the BMS will disconnect the load path of the battery while maintaining a connection to the charge controllers and the PV elements, thus charging the depleted system.

The batteries chosen for the accumulator are 180Ah lithium iron phosphate and from CALB, a reputable battery manufacturer that specializes in producing lithium iron phosphate batteries. The module size of 180 Ah was chosen with respect to the average daily power consumption of the system, with each series set of 4 modules providing 2.3KWh of energy. With a full bank of 4 parallel sets, this system should have enough energy storage for almost 3 days of autonomous run time. These batteries are bussed in series using a single set of custom bus bars. In addition, commercially available bus bars are used to parallel the strings, allowing for easy expansion in the future. These batteries and bussing solutions can be seen in Figure 10.

4.4 Power Distribution System

The purpose of the integrated power distribution system is to provide a simple and robust method of distributing both 12V DC and 120V AC power. This is accomplished through a node based DC distribution for the left and right sides of each greenhouse, with each circuit able to provide up to 30A of current. These strings are wired to reset-able breakers for each string, allowing for individual control of the strings for maintenance. These strings consist of wash-down enclosures wired together with cable glands to maintain wash-down

Figure 10: Accumulator and accessory electronics

safety, as seen in Figure 11. These wash-down enclosures will house 4 2-pin connectors for 12V DC connections, in addition to a 6 circuit terminal block to accommodate wiring the panel-mount connectors. The 2 strand cable will be routed through the cable glands, connecting the terminal blocks of the nodes to the main panel. DC nodes can be daisy chained, so long as the breaker's 30A rating and conductor's 25A rating is respected on each string.

The AC node of the distribution system is simply a GFCI outlet on an AC line from the main panel, shown in Figure 12. This is powered by the Morningstar Sure-Sine inverter, rated for up to 300W. This device is fused on both input and output for protection of the device. The AC line is routed through a cable gland into a weatherproof outlet housing, with a weatherproofing lid to cover the outlets. This will prevent accidental moisture ingress when not in

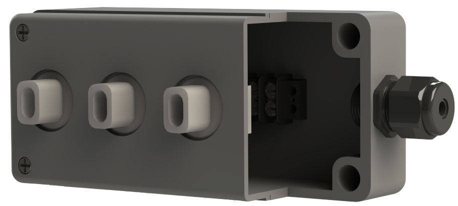


Figure 11: DC distribution node

use. Additionally, this AC line is toggled with a switch on the main panel, for safety and power reduction. A diagram of the distribution system can be seen in Figure 13.

Figure 12: AC distribution node

4.5 System Limitations

Though this system is expandable, there are limitations to its scalability. When expanding the panel count in series for each of the charge controllers, there is still the limitation of 150V input. This leads to a limit of about 18 of the LUMO panels in series. Note that operating at the limitations of the hardware will have slight effects on the efficiency of the system. The other limitation on

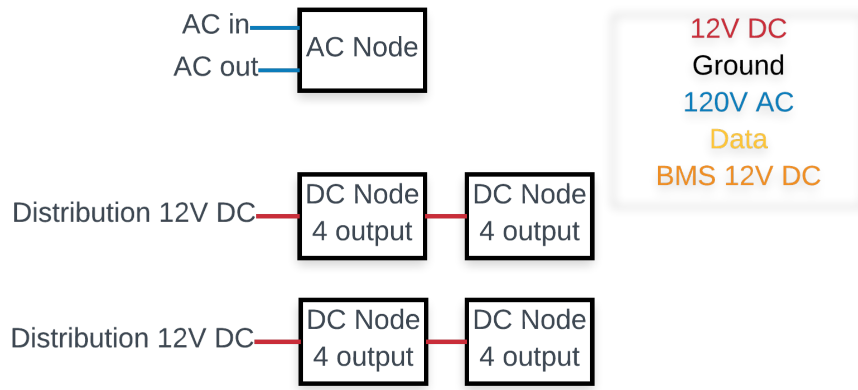


Figure 13: Distribution System Diagram

the expansion of the system is the parallelization of installations. The battery ideally operates under a 0.3C charge rate, indicating a charge amperage of 0.3*Ah. This limits the charge amperage to 54A per parallel set of cells installed in the accumulator. When parallelizing the installations, and setting up the accumulators in general, this rule must be used when configuring the charge controllers, dividing the charge rate by amount of controllers in the system. Note that this rate can be increased, but is discouraged. When expanding the solar panel count for each controller, care must be taken to not exceed the 150V max for each string. Additionally, the maximum charge current will be limited to the rate indicated in Figure 5.

5 Assembly

5.1 Control Panel

5.1.1 Internal Component Mounting

To mount the internal components to the back panel we began by printing a 1:1 scale drill template see subsubsection 12.2.1. This was affixed to the back panel. First the location of each drill hole was center punched, referencing the cross hatches on the drill template. The holes were then drilled to the appropriate size for the proper fastener for each component. Each hole was then deburred. Each component was fastened with a machine screw, with a washer and nylon lock nut on the back-side.

An issue which we encountered was one mounting hole being to close to the lip of the back panel. This interfered with the washer and lock nut. This was solved by flipping the machine screw to come through the back side so that the lock nut and washer were on the front side.

5.1.2 External Component Mounting

To mount the external components such as the cable glands, PV connectors, and auxiliary AC switch we began by printing a 1:1 scale drill template see subsubsection 12.2.1. This was affixed to the appropriate wall, referencing a corner. Then the location of each drill hole was center punched, referencing the cross hatches on the drill template. Each hole was pilot drilled then drilled to the appropriate size using a step-drill. The final hole was deburred to size and chamfered using a countersink tool. Finally each component was mounted using their included hardware.

An issue we had was bending the wall of the container near one hole. This was likely caused because we used a large twist drill instead of the step drill on that hole. We plan to solve this by sandwiching the hole with a large bolt and fender washers to clamp the wall back to flat.

5.1.3 Component Wiring

When wiring the control panel, care was taken with correctly sizing the conductors used. Crimped ring terminals were used to connect to most components, ensuring that all conductors had the proper bend radii. The exceptions to this are the Morningstar components, large screw terminals, the photovoltaic connectors, a crimped terminal, and the AC control switch which utilized flag terminals.

5.2 Accumulator

5.2.1 Component Mounting

To mount the components we began by printing a 1:1 scale drill template see subsubsection 12.2.1. This was affixed to the appropriate wall, referencing a corner. Then the location of each drill hole was center punched, referencing the cross hatches on the drill template. Each hole was pilot drilled then drilled to the appropriate size using a step-drill. The final hole was deburred to size and chamfered using a countersink tool. Finally each component was mounted using either their included hardware or with a machine screw and a nylon lock nut.

5.2.2 Battery Support

The battery supports were fabricated using the laser cutter in the fab-lab on campus. Firstly a layout was made in Solidworks and exported. We used a 1/8" acrylic sheet which was 48"x36". We then set up and cut out the support structure pieces using the laser cutter. The support structure was assembled within the accumulator container and was glued together using a clear adhesive.

An initial issue we encountered was working with the exported layout within the software for the laser cutter. After trial and error we found a .DXF file-type to work best. Another issue we encountered is that the sheet didn't quite fit inside the bed of the laser cutter. To solve this issue we cut the sheet down to 46.5"x36" and the layout was changed appropriately. Additionally in some areas the laser did not cut all the way through. We used a razor blade to finish the cut in these cases.

Note, we have not yet finished this entirely, changes and new issues encountered will be updated.

5.2.3 Component Wiring

A similar approach to the control panel was used to wire the accumulator, using ring terminals to facilitate power connections between components. The battery bus bars were fabricated out of copper bar, simply drilling and cutting the raw stock. Purpose built bus bars are used for the parallel battery connections. All conductors on the batteries are secured with split ring washers and M8 fasteners. Additionally, when installing the conductors, the BMS cell boards are installed, and wired to the BMS according to their documentation.

5.3 Power Distribution System

5.3.1 DC Distribution Node

Firstly the terminal blocks were attached to the center of the back-side of their wash-down enclosure using double-sided foam tape. To mount the cable glands we marked and center punched the desired location. This was then pilot drilled then drilled to the appropriate size using a step drill. The final hole was deburred to size. Finally they were mounted using the included hardware.

To mount the panel mount connectors to the front panel we printed a 1:1

scale drill template see subsubsection 12.2.1. This was affixed to the front panel, referencing a corner. Then the location of each drill hole was center punched, referencing the cross hatches on the drill template. Each hole was pilot drilled then drilled to the appropriate size using a hole saw. The final hole was deburred. Finally each component was mounted using the appropriate machine screw.

To wire the connectors to the power system, wires were prepared to connect the panel mount connectors to the terminal block, crimping the contact for the connector to one end, and a ring terminal to the other. These assemblies were then installed in the connectors themselves, and fastened to the terminal block using the ring terminals.

An issue which we encountered was the hole saw arbor not holding the pilot drill true. This caused an out-of-round hole. This may result in improper sealing so the front panels may have to be replaced.

5.3.2 AC Distribution Node

Since this was a purpose built enclosure designed for an application like ours this was a simple matter of assembly. First, the AC GFCI receptacle was installed using the supplied hardware. Then, the cable gland was screwed into the existing threaded hole in the enclosure, using the provided plugs to seal the remaining holes.

5.4 Installation

5.4.1 Mounting

5.4.2 Wiring

6 Sensor Node

6.1 The Information Infrastructure

Sensor Node will function as a component of the Information Infrastructure of this grid. The purpose of an Information Infrastructure is to facilitate the efficient and secure flow of data. It is responsible for communicating with sensors, actuating devices, etc, as well as providing this data securely and in real time to grid workers.

6.2 What is Sensor Node and why is it useful?

The current version of Sensor Node is a system written by Isaak Cherdak. The System's purpose was originally to be used in a distributed system of sensors communicating with a database but it actually has many more applications. In this case we are using it as a single node since we will only need one device to run all of the sensors. Regardless, we have plans to design a data visualization system so as to be able to conveniently get information from the grid in real time. In addition, Sensor Node is designed for two-way communication since its second version for things such as actuators (which we simply will be using LEDs with for the Arboretum Micro-grid) so the user interface providing data visualization will ideally provide this functionality as well. Note that all versions of Sensor Node were written for the Arduino AtMega 2560.

This section will be discussing the third version of Sensor Node. The first version of Sensor Node was created by Sargis Yonan and simply allowed continuous data output for one kind of sensor at a hard-coded frequency (usually 1 Hz). The first version is currently in a branch called "legacy" but will eventually be moved to "v1". The second version was created by Isaak Cherdak and allowed for communication with multiple devices but their locations were predefined and although it was designed with performance in mind, it wasn't designed very well to allow for easy integration with new kinds of devices. The second version is in a branch called "legacy_v2" and will eventually be moved to "v2". Version 3 was also written by Isaak Cherdak and improved on many of the limitations of version 2. In version 3 it is possible to easily add code to interface with new devices, add and remove devices *during run-time*, and easily see which devices are currently installed. This is accomplished by standardizing the kinds of values that each device, or as the system calls it, **module**, must support. One of the biggest core design choices was to categorize every device as having four kinds of commands: **init**, **read**, **write**, and **destroy**. Refer to subsubsection 6.8.2 for possible additions to this. Currently the master branch points to the third version. Sensor Node version 3 reached its first minimum-spec, complete state by the end of winter 2017 which was appropriately labeled v3.0.0. Sometime in spring 2017, Sensor Node was upgraded to version 3.1.0 which is described in subsubsection 6.6.6. Finally, note that when discussing something about the library, unless otherwise specified assume that it is in regards to Sensor Node v3.1.0.

6.3 Currently Implemented Modules

Some Modules already have been implemented and integrated into Sensor Node, further demonstrating the effectiveness of the library's simple design. Note that the wiring of these devices will be illustrated in subsection 6.4.

6.3.1 Actuator

The Actuator module was created by copying the format of the default module library and simply replacing the code as appropriate within the four standard functions. A quick comparison of module.c and actuator.c can demonstrate how simple this was to do. In addition, the actuator module is a very simple concept: a device that can be turned on or off. In our case we used LEDs as the physical devices that operated as actuators. This will be useful if we want to have a panel of LEDs to display the status of various systems inside a facility. Finally, this is also an example of a library that particularly requires two way communications between the data visualization system. For more information on this, please refer to subsubsection 6.8.1

6.3.2 Temp_Sensor

The Temp_Sensor module was created by taking its code from Sensor Node v2.0.0 and adding support for the init and read functions. It should be noted that this library was not implemented with best practices and will likely be rewritten in the near future. It contains an unnecessary length of code for the provided functionality that more than doubled the size of the binary through inclusion into Sensor Node v3.0.0. The origins and original author of the code are currently unknown but we know that it was present during Sensor Node v1. We are using the DS18B20 Temperature Sensor currently with this module.

6.3.3 Humidity_Sensor

The Humidity Sensor module was created from only using the module standard as a template and adafruit's arduino library code as a guideline. This was the first I2C sensor that was implemented in the system. The module is based off the AM2315 sensor which has an integrated DS18B20 sensor connected. When data is requested, it provides both a humidity percentage reading and a temperature reading. This sensor also has configurable integration time and gain. Currently, the defaults provided by adafruit are being used for the aforementioned parameters. Credit to adafruit for the general algorithm of communicating and requesting data from the sensor.

6.3.4 Light_Sensor

The Light Sensor module was created using the humidity sensor module as a template since the respective light sensor, the TSL 2591, is an I2C sensor as well. Similarly to the humidity sensor, the light sensor also has a configurable integration time and gain. Finally, credit to adafruit for the general algorithm of communication and requesting data from the sensor.

6.3.5 Fona

We are going to need Sensor Node to communicate over the web to receive commands and send data to a backend server. We have decided to move away from hot spot and wifi connections for a number of reasons. The major reasons were that we would have to worry less about bandwidth limitation if the data is exclusively used by the device, we wouldn't have to worry about powering an entire hot spot, and we can pay less since the data rates needed solely for communications with a backend server will be meager. Thus we will be looking into the use of Arduino shields that can communicate with the board directly. We considered the Adafruit FONA 3g initially but seeing as it currently doesn't have the libraries for TCP/IP and HTML, we will have to go with the Adafruit FONA 808. In the future, once the libraries for the FONA 3g are completed it would be best if someone added code to utilize it instead. The major concern is that the FONA 808 only uses 2g but at least it costs 50 dollars as opposed to the FONA 3g's price tag of 80 dollars. Example code is provided for the FONA

808 to do anything from making calls, texting, and connecting to the web.

This module is possibly the most important one due to some of the changes for Sensor Node that it will demand. The most notable change is that Sensor Node needs to be able to recognize a module as the main communications channel as opposed to the default of using UART0 to communicate through USB. Refer to subsubsection 6.8.4 for some potential future changes. The Fona module was created using the module standard as a template. It is untested and has been written without the opportunity of testing operation. The module should work with the Adafruit Fona 808 but also should work with other versions such as the 800. It is designed to set the Fona to transparent mode which basically forwards data from the uart to the network and vice versa automatically. This mode needs to be tested on the Fona through a serial terminal directly before trying to pipe Sensor Node's communications through this module. Furthermore, the module sets up the Fona to act as a TCP server which means it will wait for the backend server to request data before doing anything.

6.3.6 CAN_Bus

This module was created based off the module standard and uses the MCP 2561 CAN transceiver. This module requires two pins. The first pin is TX and the second is RX. To understand how to create a new device with multiple pins, please refer to subsubsection 6.5.2. TX allows the CAN bus to be set to dominant or idle by sending a 1 or 0 respectively. TX should only be momentarily set as the CAN transceiver will automatically disable dominant mode if held for longer than 125 ms. RX reflects the state of the CAN bus for dominant and idle by responding with a 1 or 0 respectively. Finally this module is untested, mainly because this isn't the final intended version. This module should be updated and renamed to BMS_Comm once it implements a set of requests for necessary data from the BMS.

6.3.7 Current_Sensor

This module was created based off the module standard and is intended to be used with HASS 200-s. The module makes use of the on-board ADC and scales the result accordingly. This module is currently untested.

6.4 Physical Wiring

Note that this wiring diagram is "loose" in the sense that we don't specify a required location for every device to be wired to. This is obviously because Sensor Node is modular and allows a user to change these locations during runtime. we will however complement the diagram with a set of commands needed to create the devices on the running instance. Also note that this is the layout that we used upon completion of v3.1.0 and this may change in later versions depending on the number of additional modules supported. For more information on how to use these commands please take a look at subsection 6.5. The illustration of the wiring can be found in Figure 14. The commands to configure Sensor Node to use this setup can be found in Table 6.

c TEMP_SENSOR $WLOC
c ACTUATOR $XLOC
c ACTUATOR $YLOC
c ACTUATOR $ZLOC
c HUMIDITY_SENSOR PD0 PD1
c LIGHT_SENSOR PD0 PD1

Table 6: Commands to create the devices in Figure 14. Note that PD0 and PD1 are the on-board locations of SCL and SDA respectively

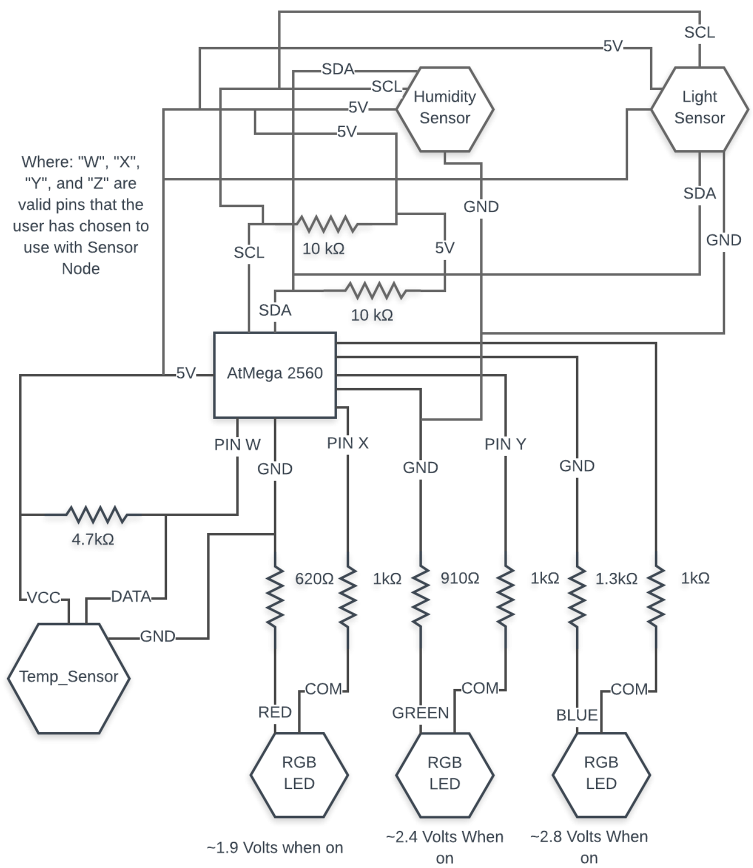


Figure 14: Sensor Node Wiring

6.5 Usage

For more details on how the system and commands work, please visit subsection 6.6. For more details on the usage of specific modules, please visit subsection 6.3

6.5.1 Key

Expect to see the terms in Table 7 throughout the Usage section.

[ ]	white-space
%PORT	PXN [PXN]*7
%DEV_STR	string describing the device type
%I	device index
%WRITE_STR	the command to send to write

Table 7: Key Terms for Usage Section

6.5.2 Create

This command creates a new device of type %DEV_STR with locations given by %PORT where for every location: 'P' has no meaning, X is the letter of the register (IE: 'A' in "PA0"), and N is the bit of the register we would like to use (IE: '0' in "PA0"). Note that it is possible to specify up to 8 pins for a module but at least one is required. After creation, "init" is called on the new device. Note that the index the device is placed on upon creation can later be determined with the "map" command described in subsubsection 6.5.8. The "create" command's syntax changed as a result of the new changes in Sensor Node v3.1.0. Please refer to subsubsection 6.6.6 for general information about the changes.

Syntax: **c**[ ]%DEV_STR[ ]%PORT

6.5.3 Initialize

This command can be overwritten by a module. By default it will print a generic string saying that "init" was called on device %I of type %DEV_STR. In short, it will call the "init" command on device %I and print an informative string.

Syntax: **i**%I

6.5.4 Read

This command can be overwritten by a module. By default it will print a generic string saying that ”read” was called on device %I of type %DEV_STR. In short, it will call the ”read” command on device %I and print an informative string.
Syntax: **r**%I

6.5.5 Write

This command can be overwritten by a module. By default it will print a generic string saying that ”write” was called on device %I of type %DEV_STR with string %WRITE_STR. In short, it will call the ”write” command on device %I and print an informative string.
Syntax: **w**%I[ ]%WRITE_STR

6.5.6 Destroy

This command can be overwritten by a module. By default it will print a generic string saying that ”destroy” was called on device %I of type %DEV_STR. In short, it will call the ”destroy” command on device %I and print an informative string.
Syntax: **d**%I

6.5.7 Kill

This command will remove device %I from the current instance's configuration. Before doing so, it also calls destroy on device %I.
Syntax: **k**%I

6.5.8 Map

This command prints information about the configuration of the current instance. Currently, this information simply shows which types of devices are located on which indices as well as the available device types. In the future it will also display the on-board location(s), and possibly an additional custom string that each device may provide for additional identification or status purposes.

Syntax: **m**

6.6 Documentation

6.6.1 Overview

This section will talk about details that the general user may not be interested in, but a programmer seeking to utilize our libraries or expand them will need to know. We will be designing with backwards compatibility in mind for at least version 3.0.0. As of Sensor Node v3.0.0 a device operating the firmware allows a user to do 7 fundamental operations. In subsubsection 6.6.6 however, changes are described that clearly break the backwards compatibility in exchange for important functionality. Three of the operations available on Sensor Node do not change in functionality no matter which modules are installed: ”map” (’m’) which displays the current system configuration, ”create” (’c’) which adds a new device at a specified group of pins to the running instance, and ”kill” (’k’) which removes a device from the running instance. Then there are four commands that are standardized and hence differ between different types of devices: ”init” (’i’) which should initialize a device, ”read” (’r’) which should request data from a device, ”write” (’w’) which should send a string to a device, and ”destroy” (’d’) which should set the pins the device was running on to the state that they were in before initializing the device. Note that if overwriting one of these four standardized commands in a module, intend for the module to print information using the UART library. Note that this is done in accordance with the changes in subsubsection 6.6.6. Finally, Sensor Node comes with three libraries that must be included: a UART library, a parser, and the module standard library. Note that some devices require additional libraries like the Two Wire Interface.

6.6.2 UART library

The UART library is made up of uart.c and uart.h. The library provides an interface with the on-board uart which currently allows communications through a serial terminal. It provides the same functions a developer would normally expect when programming using IO like printf() and gets(). This library also features two interrupts: one always triggers when data is received while the other is enabled only when there is data to be transmitted and triggers when the data register is empty.

6.6.3 Parser library

The parser encapsulates all of the arguments that we would expect from a command. It doesn't print anything directly either (this may change to avoid passing of entire strings) instead the core library prints messages related to the return of the parser. The parser will return with the command parameter being set to NULL in the scenario that it detects an error. It doesn't handle all errors: some error handling like checking for valid index or invalid string types are considered in the core library since the parser doesn't have access to the information necessary to check those things. When the parser successfully returns, it will have a character saved depending on the command that was correctly executed. In the case of 'i'/'r'/'d'/'k' it simply expects a number and will set the device_index to %I. In the case of 'w' it sets device_index to the %I and ret_str to the %WRITE_STR. In the case of 'c', it sets the ret_str to %DEV_STR, address_index to (X - 'A'), and reg_bit to (N - '0'). Note that the aforementioned functionality has slightly changed in accordance to subsubsection 6.6.6. Finally, In the case of 'm', it simply returns. Unless specifically defined for a specific case, developers should assume that the value for a given parameter of a Parser object is undefined. To see the expected syntax for Sensor Node, please visit subsection 6.5.

6.6.4 Module library

This library standardizes the parameters that are expected for a given module for Sensor Node. When creating a new module, the developer should override every parameter of the module struct. For example, overriding type_num is very important because, otherwise when a device of that type is created, it will be the default type which is -1, and hence complain about not existing, or crash the system. For more information check out subsubsection 6.6.5. It should be noted that some things are likely to be changed such as the way the information for pins is saved (this particular concern is addressed in the multiple pin fix described in subsubsection 6.6.6). The standard set by this library is what allows us to make simplified assumptions about how we can use every library. Also note that printing should happen inside modules. The functions pointed to by a module struct should not pass strings to the core library as doing so takes a large toll on system RAM. Furthermore, if a string is truly constant (as opposed to a non-constant string being cast to a constant) then it should be saved in program memory by using the PSTR() macro as well as printed using the uartx_puts_P() function (where x reflects the UART number).

6.6.5 Core library

This library is the control system of Sensor Node. It keeps all of the maps such as: type index to creation function, type index to type string, device index to device, etc. It also is in charge of receiving data from the UART and requesting specific information from that data through the parser. Once it receives the data, it acts upon one of the 7 commands. For general information on these commands refer to subsubsection 6.6.1 or for usage refer to subsection 6.5. A block diagram demonstrating the interconnection of the high level system architecture can be found in Figure 15

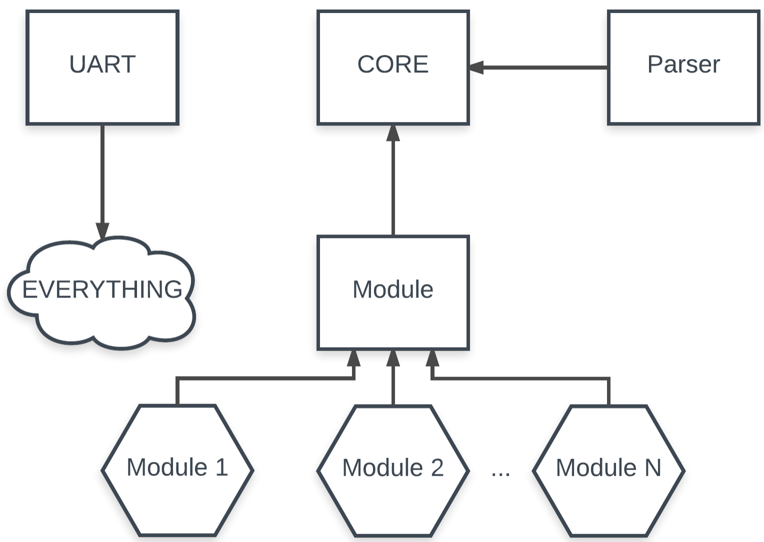


Figure 15: Sensor Node Architecture

6.6.6 Changes of Version 3.1.0

The recent changes to the system were big enough to warrant an increase in the version number. This was because all modules at this point had to be rewritten and a good chunk of the program was restructured. The change involved two main things: allowing for multiple pins per module, and making use of program memory for constants while reducing the number of strings being passed around. The former point was to address a potential downfall and issue of the V3.0.0.

The latter however, was in response to running out of RAM because the old method of returning strings in the standard functions resulted in over-use of system RAM. As a result of this fix, a lot more of the system is carefully written to optimize for using the Program Memory to store string constants whereas non-constant strings are usually printed immediately instead of passed around. It is now important to make sure that Program Memory is being used for all constant strings and that no strings are passed around which are both constant or not in Program memory.

6.7 Issues

6.7.1 Interrupt Interference

This is more of a potential issue that hasn't been considered and has yet to cause any problems. The point is that there may be modules or general additions to Sensor Node that utilize additional interrupts. This shouldn't necessarily be encouraged but it certainly *shouldn't be discouraged*. Although the design has been considerate for an embedded environment, it hasn't been considering possible problems that may arise with larger numbers of interrupts. Fundamentally, the key thing to be considerate of is atomic operation (not yet actualized) and limiting the amount done during an interrupt service routine (actualized).

6.7.2 Inconsistent depending on devices

Unfortunately, Sensor Node is designed to busy wait until any given operation is done. In theory this didn't seem like a problem but in reality, some devices have integration times of around 500 ms which is quite a lot. If enough sensors of this type (there are currently about 3 like this running on the system) are being used at once, the transmission times of packets won't be consistent. The planned fix is described in subsubsection 6.8.4.

6.8 Future Work

Sensor Node is always being improved upon but we have very concrete plans for our next steps that we expect to complete by the end of summer 2017. The plans will be extended even more at that time for the final version of this report.

6.8.1 Data Visualization System

We plan to create our own data visualization system. We have two major concerns: that the system be dynamic and scalable so as to match the devices running for any given instance as well as to be full duplex (allow reading data from as well as writing data to Sensor Node modules). Because this is a tall order, few existing systems provide this functionality so it is a better time investment to make our own than to modify and integrate with an existing system. Currently, the Data Visualization System is by far the largest concern in terms of expected time and scale for the Information Infrastructure of our grid.

6.8.2 Option standard command

A command for specifying options such as re-defining integration and delay times on the light and humidity sensors was being considered. The idea will however be scrapped and merged into the functionality of the write command since the write command is versatile and already passes a string so it can be overridden to expect a string to configure something or to actuate something.

6.8.3 RS232 module

The module will work with RS232 primarily for communicating with the morningstar systems already in the grid.

6.8.4 Sensor Node version 3.2.0

Sensor Node will be further revamped, with the use of work queues so that packets get sent out in a timely manner. The current idea is to replace every standard function with two functions. The first will be called and return a number for the expected minimum wait time before the second should be called. The second function will be put on the work queue with a saved completion time based off the provided minimum time. The system will continuously be polling the time from the last packet transmission to figure out exactly when a function needs to be run and hence will be able to run various sensors without having to block for busy waiting. As a result, a packet can be sent out every longest delay of all requested sensor data. This version is also rumored to be considered for implementation on the AtMega 3560. Finally, this version will use a communication module system: allowing a module to specify the communications operation, hence overriding the default of communication over the serial terminal.

6.8.5 Ideal Deliverable of Modules

In the ideal case, all of our future work would be completed by the end of Summer 2017. This is illustrated in Figure 16

7 Power Budget

7.1 Client Power Budget

We were supplied a power budget by the user which indicated **1.80kWh/day** total daily usage on average. However, on further analysis of their power budget we found numerous issues. For example, the power budget makes the assumption that an item which is used once or twice a month can have it's load averaged

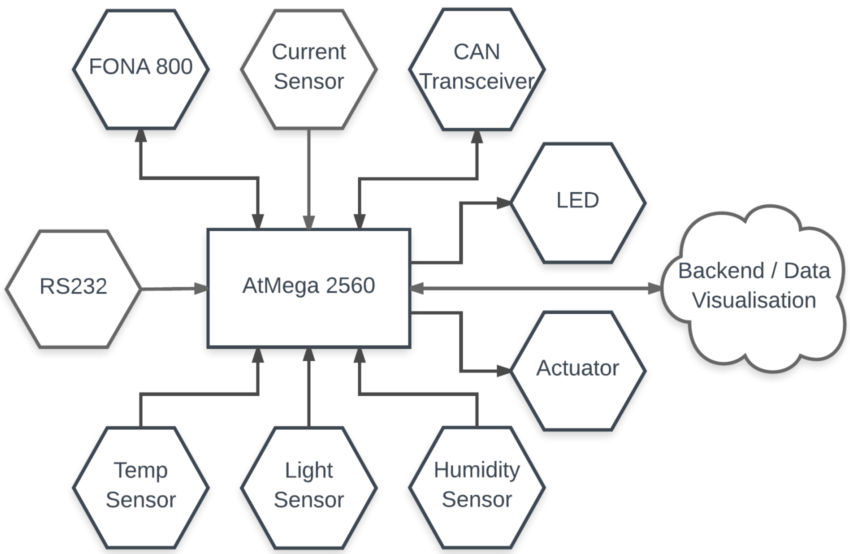


Figure 16: Sensor Node Deliverable

out and distributed throughout the month. Operating under this assumption is a discouraged practice, as on those days where the load needs to be operated, the system might not be able to support the additional strain. This is combated with a thorough power budget, and by considering safety factors when scaling systems. Additionally, being that there are two greenhouses it is unclear if this power budget is for one greenhouse, or if the loads are to be distributed through both.

Another shortcoming is the assumption that certain components do not consume power, or that the power that they consume is negligible. An example of this is a 12V digital timer. In the data sheet for the timer chosen by the user, it specifies a 2W power consumption. Additionally, as these are relays which operate on an inductive coil, the coil will consume additional power when the relay is switched closed. Another item which they assume consumes no power is a 12V battery backup air pump. This will always have some level of vampiric

Table 8: Power Budget

Equipment	Qty	Watt	Duration	Frequency of Use	Total Usage/ day
Air pump	1	35 W	24 hr	daily, all day	840Wh/day
LED Supplemental Lighting	4	10 W	18 hr	daily, dusk to dawn	720Wh/day for all lighting
Water Pump	2	5 W	45 min	4x/ day = 3hr	30Wh/day for both pumps
Utility transfer Pump - Rainwater Harvest	1	60-75W	15 min	up to twice/ month	~2.5Wh/day
Utility transfer Pump - Water Refill	1	60-75W	3-5 min	once/ week	~2.5Wh/day
Fish Feeder	1	N/A	15 min	4x/day = 1 hr	TBA (50wh/day?)
Digital Timer	4	N/A	24 hr	daily, all day	N/A
AquaBaclup BatteryON with Oxygen Diffuser	1	N/A	TBA	upon system failure	N/A
				Total Projected:	1.65KWh/ day
Additional Equipment (projects)					
Utility Transfer Pump - Hydroponics beds	1	60-75watt	30min	4x/day	120-150Wh/day
				Total Projected:	1.80KWh/ day
Additional Equipment (greenhouse)					
Fans	2	20watt	24 hr	daily, all day	1000Wh/day
				Total Projected:	2.80KWh/ day

energy losses, in addition to having considerable power draw when charging.

Finally, a major issue in the power budget is that it is missing the 2 ventilation fans which each greenhouse uses to circulate air. In the current microgrid configuration, these fans run continuously, as long as there is sufficient power in the accumulators. At an average of 20W each, these fans would consume $20W * 2fans * 24hours = .96kWh/day$. This is a considerable 53% increase in the power budget, which will be accounted for in our modified power budget, as seen below in Table 8.

7.2 Power Optimization

A major shortcoming of the user provided power budget is the inefficient and over-specified hardware components for their aquaponics system. We will make recommendations on alternative components, to maintain the system's overall efficiency. The suggestions made are to address components that outperform their assigned task, and have a significant power consumption relative to their use.

7.2.1 Air Pumps

The users specified a 12V DC 35W 68L/Min(2.4CFM) by HSH-Flo found here on Amazon. The important specification to consider for this equipment is maintaining proper airflow for the size of the tank, while minimizing power consumption. On the Amazon page where the pump is found there is also an 18W version of that series of air pump. Notably, the airflow is about half of the pump specified, providing a cost saving alternative.

The website aquacave.com, which is a retailer and information resource for aquariums and aquaponics. Here, we were able to compare a variety of air pumps, and find what level of airflow is suitable for the aquaponics system using the holding tanks in the greenhouses. The main basis of comparison was capacity of tanks which pump were being used on in gallons, with an approximation of the tank size being about 700 gallons.

Our recommended pump is the Eco Plus 1 air pump which is **18W** power consumption, and outputs 38L/Min. The air pump runs continuously and would result in a 408Wh/day reduction in power draw, 25% relative to the user supplied power budget. Not only does this supply the necessary flow at a reduced power consumption, the construction of the product is more robust than the user's original choice.

7.2.2 Ventilation Fans

The current fans in the greenhouse are Snapfans found on snapfan.com. There is one 20" fan and one 16" fan per greenhouse, acting as ventilation for the interior. Currently, only one of the fans is operating properly, with the other working intermittently. This suggests that the failing motor is seizing, operating at stall torque when it does, resulting in large, unaccounted losses.

The two front runners for ventilation fans are the existing Snapfans, and fans made by sunshineworks. Ultimately, the fans by the latter were ruled out, as they do not operate well on 12V DC and are less efficient than the Snapfans. Moreover, the Snapfans would be a simple replacement, as the current system is already designed with them in mind.

We would recommend we replace the current Snapfans with newer model Snapfans. An alternative option is replacing the motors in each fan. Replacement components and manuals can be found on snapfan.com. Additionally, we have added provisions to limit the time during which the fans run, with considerations for time of day, temperature, and battery level. This is achieved using the fan contactor in the charge controller panel, and control by sensor node. Moreover, the newer Snapfan's power consumption is **34W** total per greenhouse (one 20" fan and one 16" fan).

	(IN H2O)	(CFM)	RPM	AMPS	WATTS	CFM/WATT
24VDC	0.0	1720	1504	2.60	63	27.2
	0.05	1500	1447	2.72	65	23
	0.10	1310	1436	2.83	68	19.2
18VDC	0.0	1380	1211	1.73	31	44.5
	0.05	1110	1162	1.85	33	33.7
	0.10	890	1166	1.91	34	26.0
12VDC	0.0	980	876	0.98	12	84.7
	0.05	630	844	1.06	13	50.3
	0.08	260	841	1.15	14	19.0

(a) 16" DC Snapfan power consumption

	(IN H2O)	(CFM)	RPM	AMPS	WATTS	CFM/WATT
24VDC	0.0	2800	1445	4.13	101	27.7
	0.05	2500	1415	4.33	105	24.0
	0.10	2200	1391	4.45	108	20.5
18VDC	0.0	2200	1151	2.67	48	45.4
	0.05	1900	1121	2.86	51	36.3
	0.10	1500	1107	2.9	52	28.6
12VDC	0.0	1600	843	1.47	18	87.4
	0.05	1100	814	1.62	20	55.8
	0.10	500	823	1.55	19	28.8

(b) 20" DC Snapfan power consumption

7.2.3 Water Pumps

The pumps specified by the user seem to be well suited to the system, with both performance and power specifications within reason for the utility. The 60-75W is given by the pumps 1/12 HP rating, with it operating at 12V. Additionally, this is an intermittent load, operating only a handful of times per day, reducing the necessity to source a more efficient pump. Moreover, the pump meets the users requested capability of maintaining acceptable flow when raising water up to 5'. According to the manufacturer, a discharge height of 10' reduces the flow from 450GPH to 390GPH, more than sufficient for the utility. Finally, the pump is manufactured by a reputable company which provides documentation, manuals, and replacement parts.

7.3 Delivered Power Capabilities

The microgrid is scaled to best meet industry standards relative to the prospective usage which we've estimated for, within our budgetary and time constraints. The explanations of the specification given in Table 9 can be found at section 3, subsection 4.1, subsection 4.3, subsection 4.4, and in the SureSine Manual.

Table 9: Current Power Capabilities

Average Worse Case Output	2.98kWh/day
Accumulator Capacity	2.3kWh
Estimated Autonomy	1 day
Maximum Power Generation	1.44kW
Maximum Battery Discharge	1.92kW
Battery Lifetime	5 years
Maximum DC String Current	30A
Maximum AC Current	3A

8 Performance

The performance of the system is yet to be characterized. Thorough characterization and performance analysis will be performed on the system once the implementation is completed and data can be gathered through sensor node.

8.1 Current Performance

todo

8.2 Future Optimizations

todo

9 Project Budget

Table 10: Total Budget

BUDGET	
Controller Panel	\$1,908.87
Accumulator	\$1,884.04
Distribution System	\$335.47
Electrical	\$291.92
Total	\$4,420.30

Above is the budget for the installation at the time of finalization. Throughout the development of the installation, the target budget was \$5,000 for the purchased system components, with the itemized budget amounting to \$4,420.30. In addition to these costs, tools and consumables are noted in the miscellaneous section, amounting to \$XXXX. Components specifically sourced will be detailed in the appendix. The costs documented in this report are based off of supplier packing quantity, and are lessened when considering the amount actually utilized in the installation. The current ledger and price breakdown can be found in our arboretum microgrid budget. This document will be updated for any design amendments to ensure accuracy as the project progresses.

9.1 Controller Panel

Table 11: Controller Panel

Description	Comment	Price	Quantity	$1,908.87
Wall-Mount Type 4 Enclosure, 24x20x8		$217.87	1	$217.87
Tristar MPPT		$496.00	2	$992.00
Sure Sine Inverter		$211.05	1	$211.05
Panel Mount PV Connectors		$5.68	2	$11.36
PV Contactor	PN: P115		3	$0.00
Panel Mount Breaker	3x 15A, 2x 30A	$30.36	5	$151.80
Battery Breaker	1x 150A	$58.74	1	$58.74
Bus Bars		$20.49	2	$40.98
Cable Gland		$3.24	8	$25.92
100A DC AGU Fuse	100A AGU Fuse	$5.98	1	$5.98
#8 ring terminal 10-12awg	Bus Bar PV GND/distribution	$11.27	1	$11.27
#10 ring terminal 10awg	Solar contactor power	$11.28	1	$11.28
#6 ring terminal 16awg	Solar contactor control	$8.30	1	$8.30
1/4" ring terminals 4awg	Power connections	$9.20	1	$9.20
1/4" ring terminals 10-12awg	Breaker connections	$13.29	1	$13.29
#8 ring terminal 16awg	Bus Bar Low Power	$7.90	1	$7.90
AC switch		$11.04	1	$11.04
DB9 connector		$9.95	2	$19.90
Standoff		$0.70	8	$5.60
Breaker mount plate	1 ft	$1.18	1	$1.18
Current Sensor	200A	$22.77	3	$68.31
Current Sensor Connector		$0.46	3	$1.38
Current Sensor Connector Terminal		$0.16	12	$1.92
10g wire	PV	$2.26	10	$22.60

9.2 Accumulator

Table 12: Accumulator

Description	Comment	Price	Quantity	$1,884.04
Wiring Trough Type 4 Hinged Cover, 12x12x36		$244.78	1	$244.78
CALB 180 Ah SE Series LiFePo4 Battery		$214.00	4	$856.00
Accumulator Contactors	PN: GX32		2	$0.00
Diode Bridge		$38.77	1	$38.77
Cable Gland		$3.24	3	$9.72
Series Bus Bars	Copper Stock	$13.18	1	$13.18
Parallel Bus Bar	PN: CA180V71	$3.50	16	$56.00
EMUS BMS		335.68	1	$335.68
EMUS Isolators		12.82	1	$12.82
EMUS Cell Board		38.89	2	$77.78
EMUS Current Sensor		114.11	1	$114.11
Acrylic for Cell Holder	1/8" sheet	50.00	1	$50.00
Acrylic Cement			1	$0.00
Battery Bolt		$8.94	2	$17.88
Battery Split Washer		$8.59	1	$8.59
4awg Wire		$2.37	12	$28.44
Data Wire		$0.92	8	$7.36
3/8 ring terminals	Power	$9.33	1	$9.33
Diode Bridge mount	1ft	$3.60	1	$3.60

9.3 Distribution System

Table 13: Distribution System

Description	Comment	Price	Quantity	$335.47
Terminal Block		$3.00	4	$12.00
Terminal Block Jumper	3 Circuit Jumper for 3/8"	$2.08	8	$16.64
Washdown Enclosure		$14.93	4	$59.72
Cable Gland		$3.24	7	$22.68
Panel Mount Fastener		$5.70	1	$5.70
GFCI Outlet	Color: White	$21.73	1	$21.73
Outlet Box		$4.86	1	$4.86
Outlet Cover		$6.44	1	$6.44
12g 2 strand wire	DC	$1.31	58	$75.98
12g 3 strand	AC	$1.45	8	$11.60
16g 1 strand wire	25ft red, 25ft black	$6.98	2	$13.96
DC Panel Mount Connectors		$3.09	16	$49.44
DC Plug		$0.78	16	$12.48
DC Female Terminal		$0.31	32	$9.92
DC Male Terminal		$0.23	32	$7.36
DC Panel Mount Wedgelock		$0.15	16	$2.40
DC Plug Wedgelock		$0.16	16	$2.56

9.4 Electrical

Table 14: Electrical

Description	Comment	Price	Quantity	$291.92
LED Voltmeter	User requested	$2.10	1	$2.10
Arduino Shield PCB	Bay Area Circuits	$30.00	1	$30.00
FONA 808		$49.95	1	$49.95
SMA to uFL		$3.95	1	$3.95
SMA Antenna		$4.95	1	$4.95
SIM 2G		$9.00	1	$9.00
500 mAh Lipoly battery		$7.95	1	$7.95
AtMega 2560		$45.95	1	$45.95
DS18B20 Temperature Sensor		$9.95	4	$39.80
AM2315 Humidity Sensor		$29.95	1	$29.95
TSL2591 Light Sensor		$6.95	1	$6.95
CAN Transceiver		$0.93	1	$0.93
5V reg		$2.89	1	$2.89
Relay		$1.79	2	$3.58
4x4.7k Resistor Network		$0.55	1	$0.55
3x10k Resistor Network		$0.54	1	$0.54
Capacitor		$0.92	2	$1.84
Diode		$0.45	2	$0.90
2 Screw Terminal		$1.53	2	$3.06
4 Screw Terminal		$3.07	1	$3.07
8 Screw Terminal		$5.74	1	$5.74
9 Screw Terminal		$6.49	1	$6.49
12 Screw Terminal		$8.43	2	$16.86
Header Kit		$14.92	1	$14.92

9.5 Miscellaneous

Will contain miscellaneous purchases once final installation is complete.

todo

10 Standard Operating Procedure

The following are the standard operating procedures for the subsystems of the microgrid. These can be used as reference when using, maintaining, and expanding the system.

10.1 Power System

todo

10.1.1 Battery Troubleshooting

todo

10.1.2 Connecting to a DC Node

todo

10.1.3 Expanding the Accumulator

todo

10.1.4 Checking Breakers

todo

10.2 Information Infrastructure

todo

10.2.1 Wiring of devices to AtMega 2560

todo

10.2.2 Adding Sensor Node module libraries

todo

10.2.3 Creating and removing devices for Sensor Node

todo

10.2.4 Using the data visualization system

todo

10.3 Constructing the Installation

todo

10.3.1 Control Panel

todo

10.3.2 Accumulator

todo

10.3.3 Power Distribution System

todo

10.3.4 Installation

todo

11 Conclusion

With the retrofit of the Arboretum greenhouse's microgrid, we hope to alleviate all maintenance issues experienced by the users, facilitating a more efficient use of the greenhouses. Additionally, the installation of this system will also allow for expansion of the uses of the area, by more effectively using the harvested energy and introducing an energy surplus to the system. The system was designed with component replacement and maintenance in mind, which will allow for repair of features, rather than their removal. However, this modularity and documentation introduces the opportunity for future students to update and upgrade the system, beginning with the expansion and replication of the original system. With these advancements in mind, we hope to facilitate use of the system by both biology and engineering students as an example and test bench for future research.

12 Appendix

12.1 Electrical Schematics

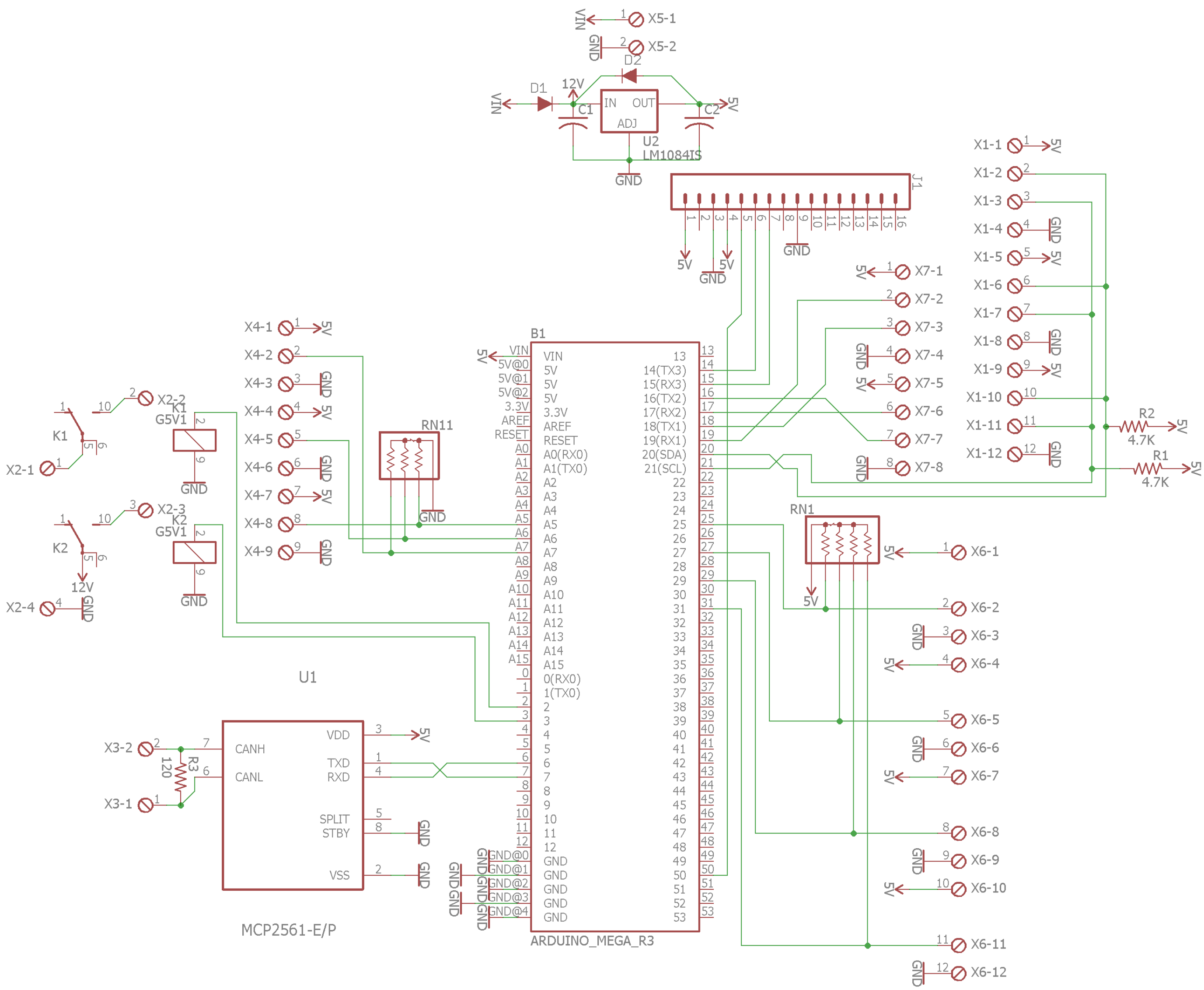


Figure 18: Sensor Node Shield Schematic

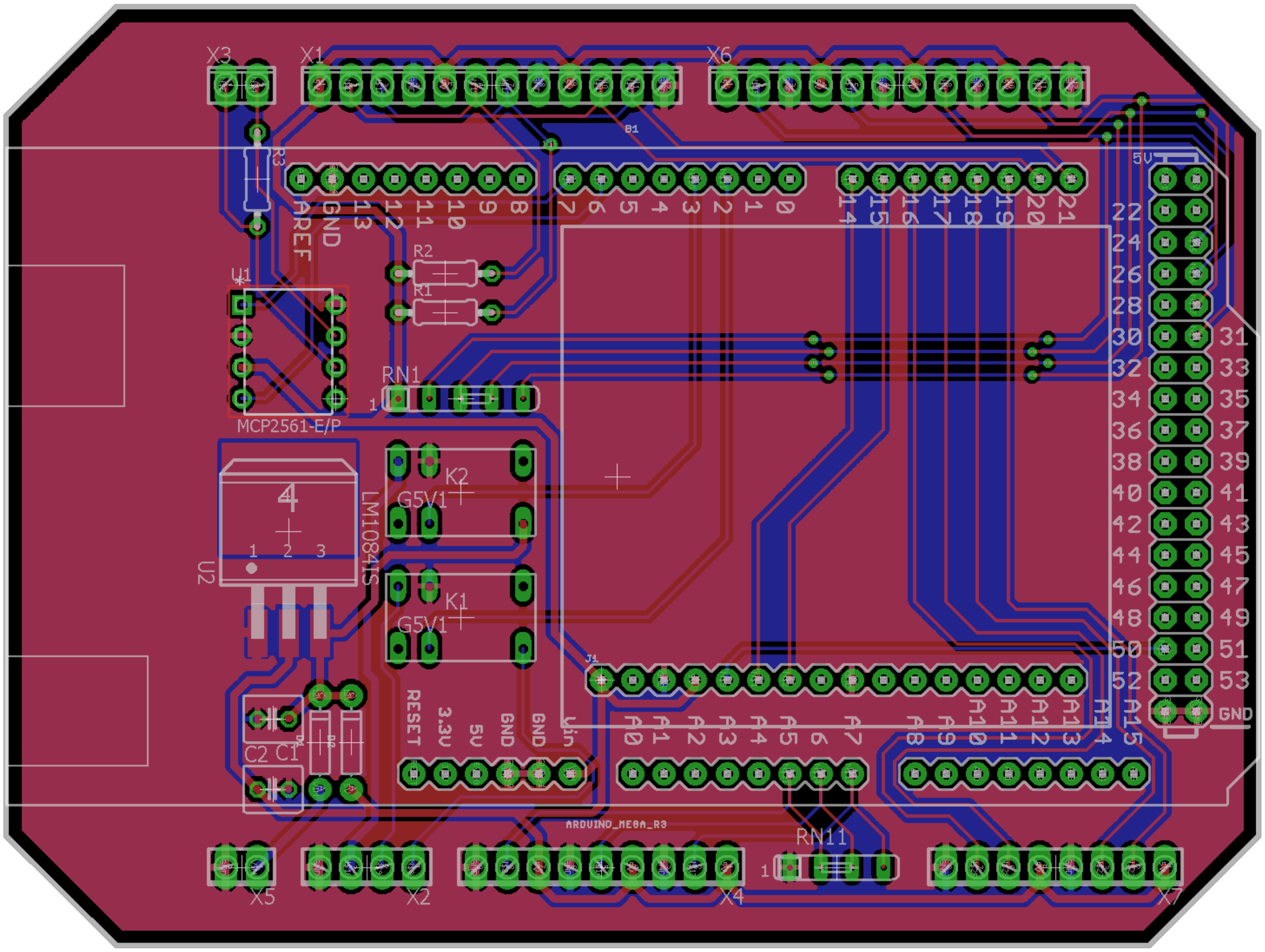
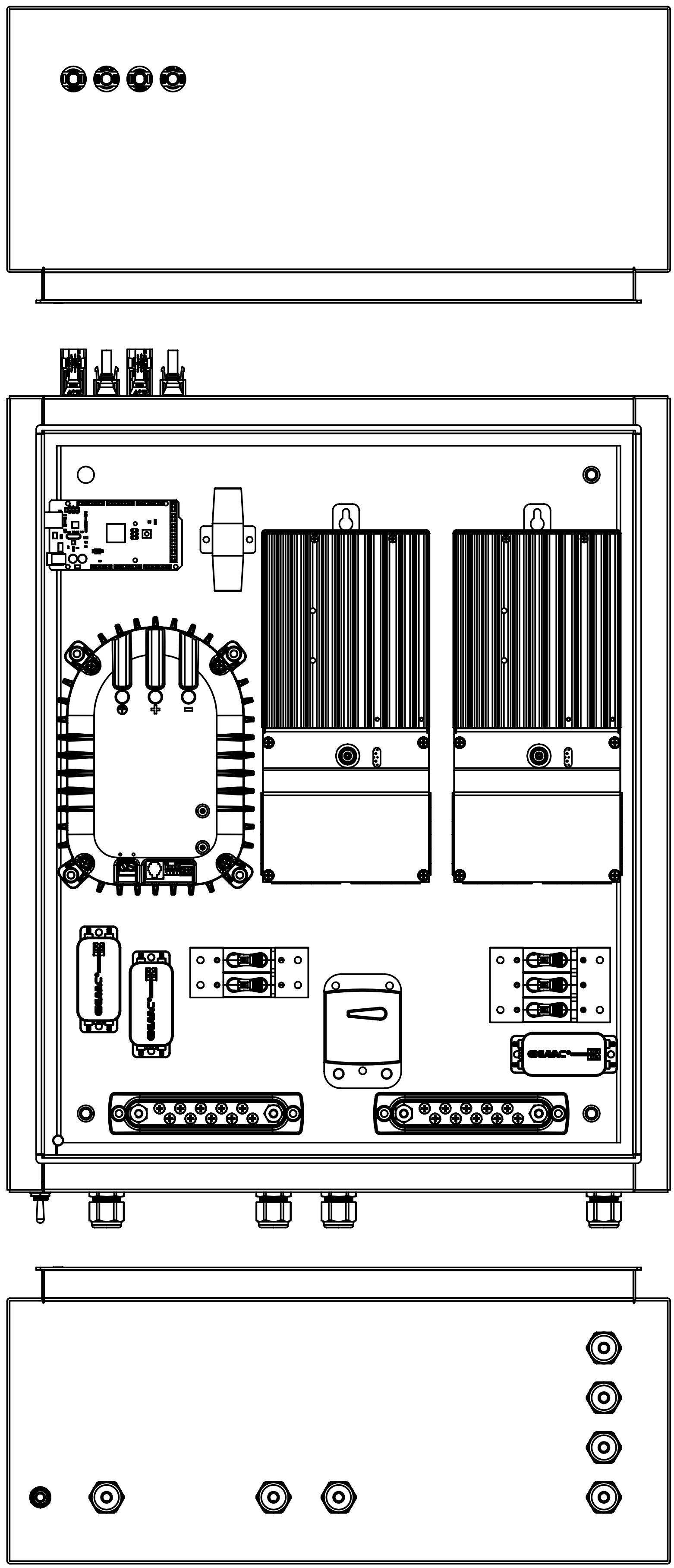


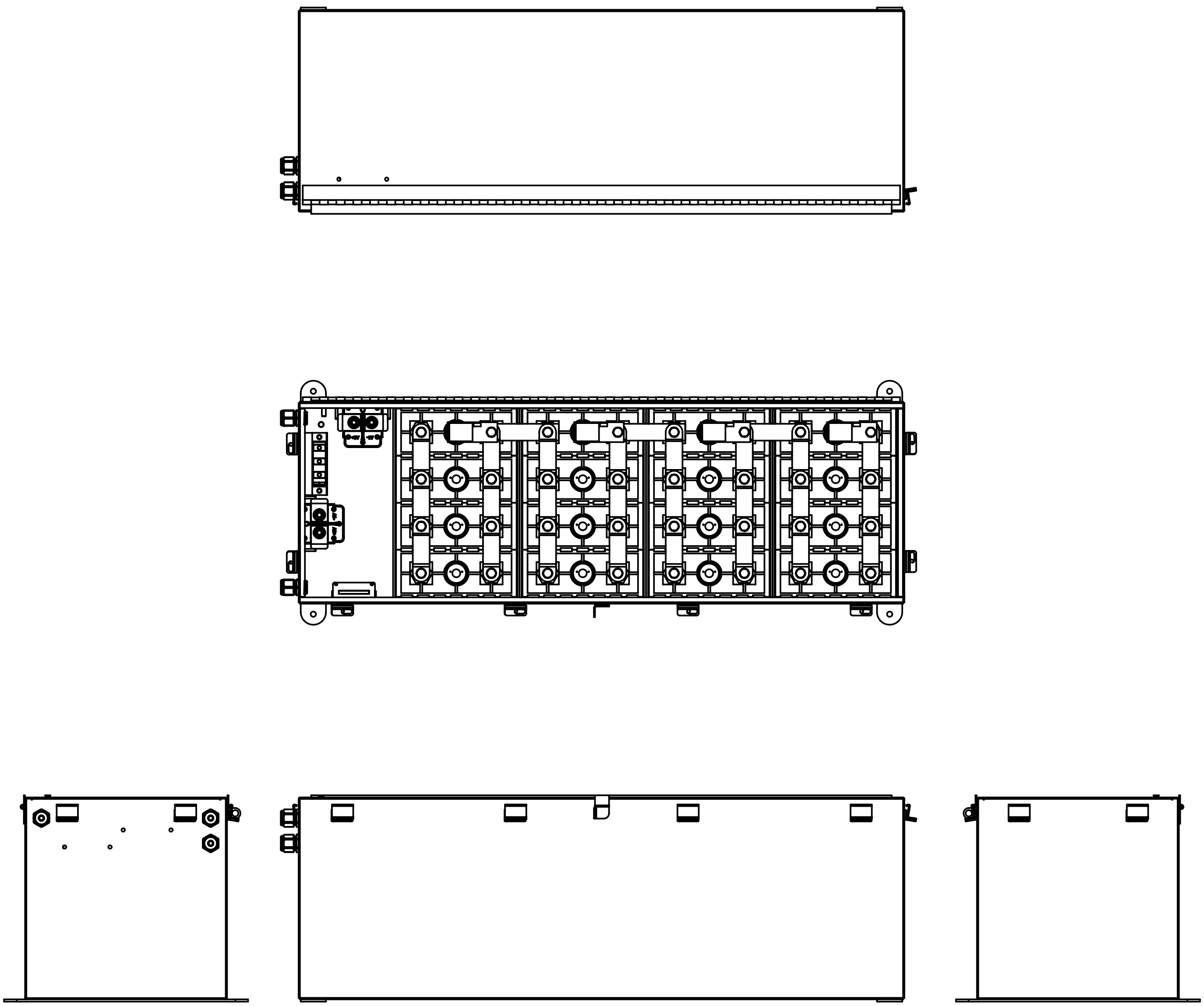
Figure 19: Sensor Node Shield Layout

12.2 Mechanical Drawings

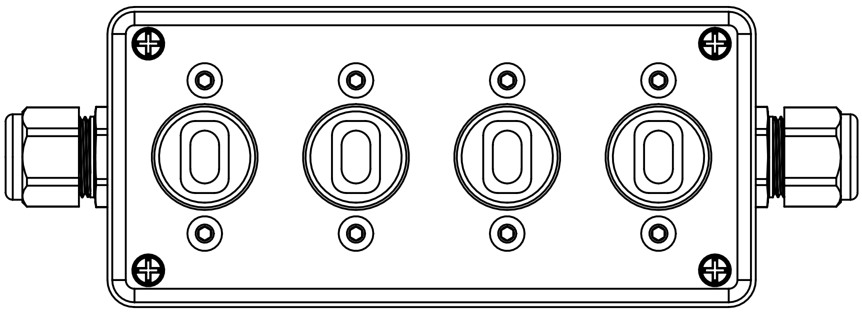


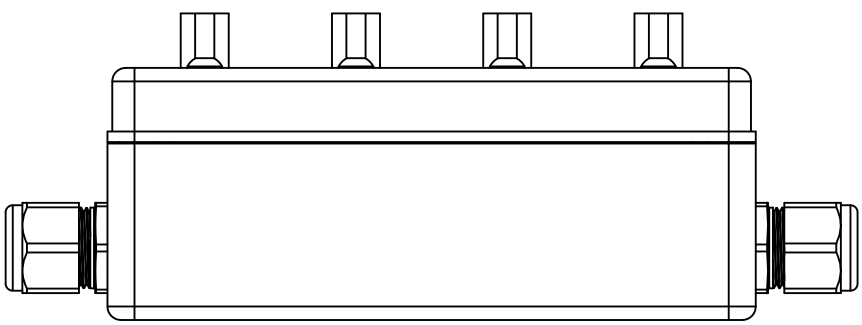
SOLIDWORKS Educational Product. For Instructional Use Only

Arboretum Microgrid			
Controller Panel Layout			
SIZE A	DWG. NO. Controller Panel		REV. 1
SCALE:1:5	Qty:	SHEET 1 OF 1	



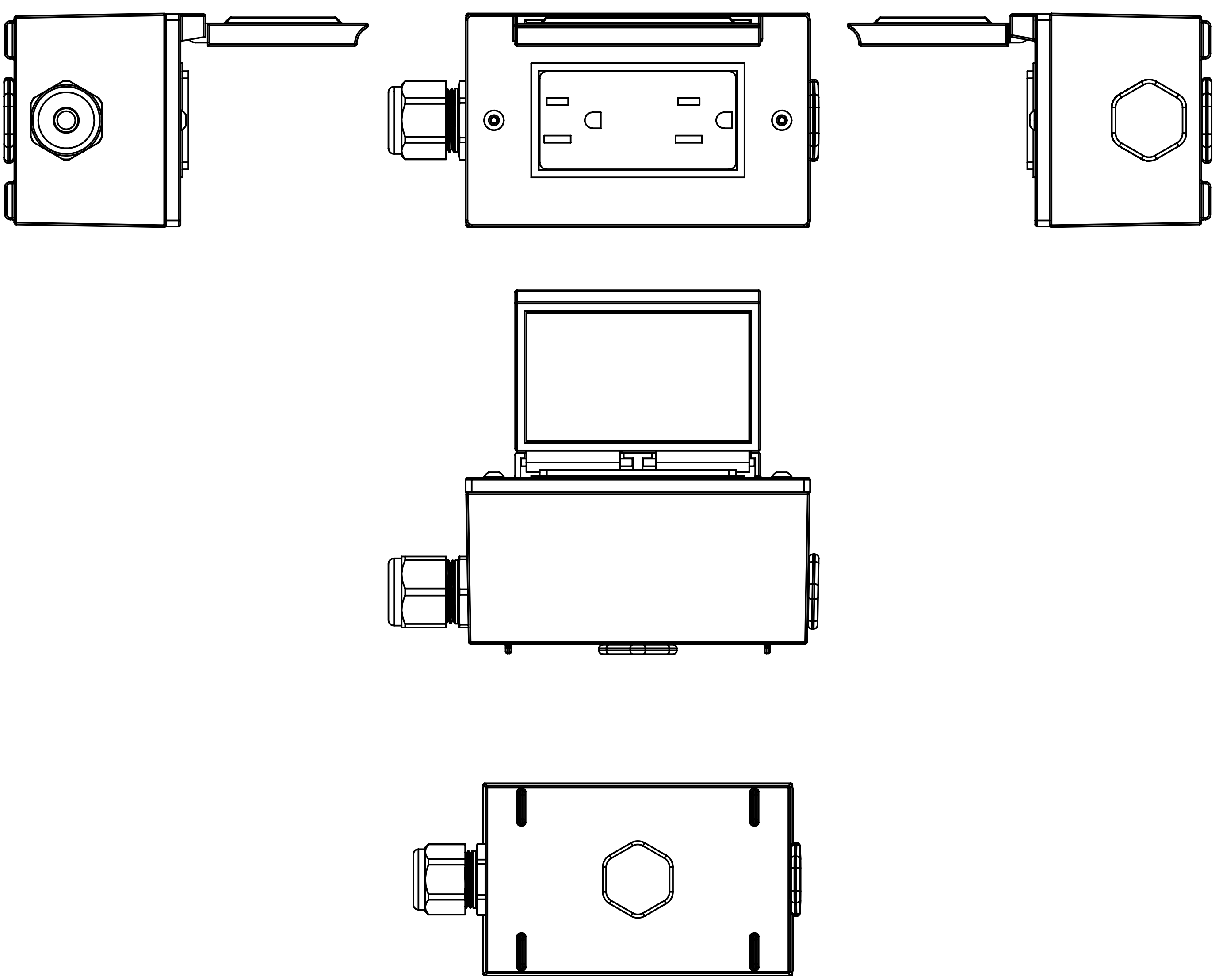
Arboretum Microgrid	
Accumulator Layout	
SIZE A	DWG. NO. Accumulator	REV. 1
SCALE:1:10	Qty: 1	SHEET 1 OF 1





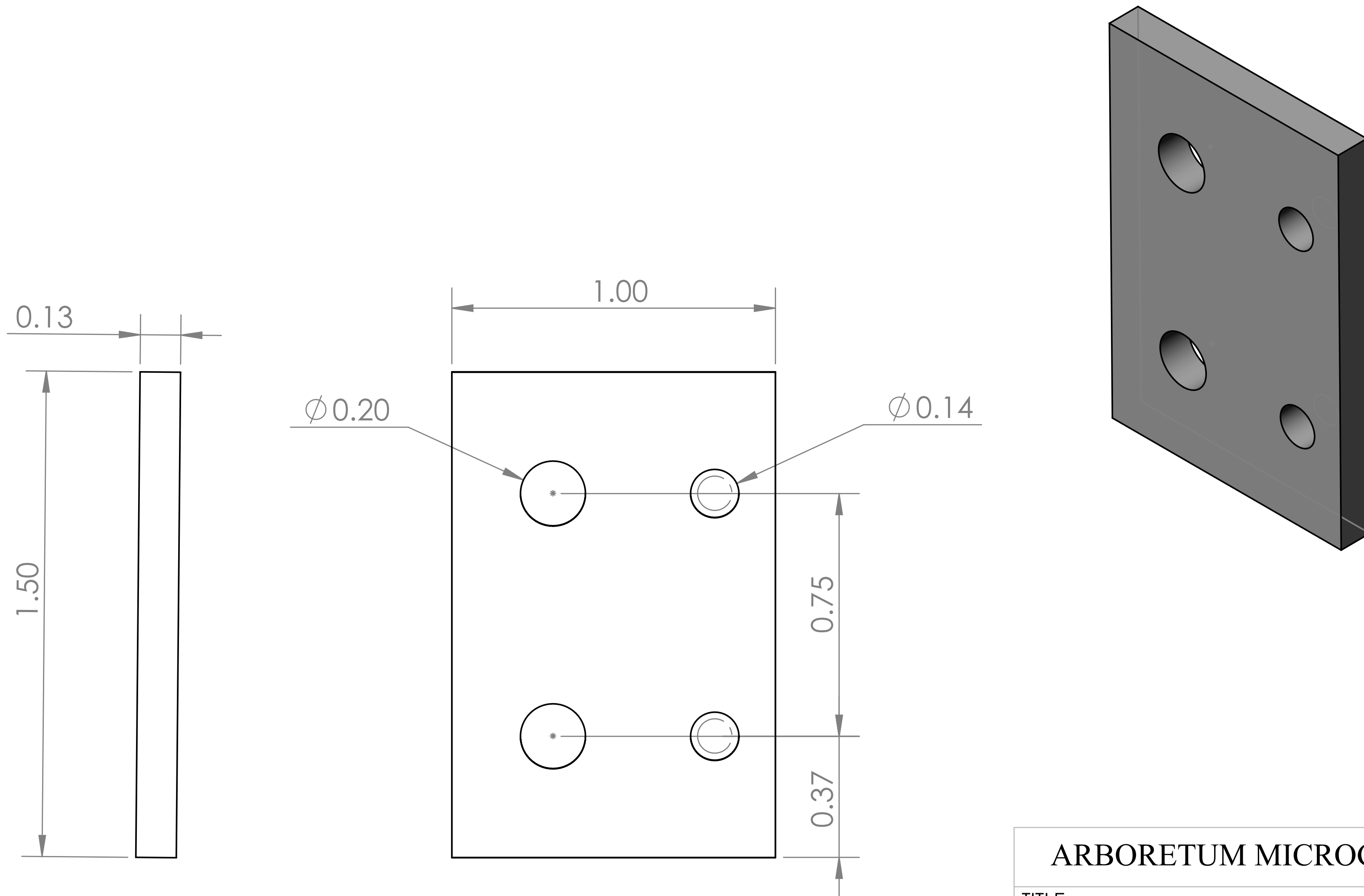
SOLIDWORKS Educational Product. For Instructional Use Only

Arboretum Microgrid			
DC Node Layout			
SIZE A	DWG. NO. DC Node		REV. 1
SCALE:1:2	Qty: 4	SHEET 1 OF 1	



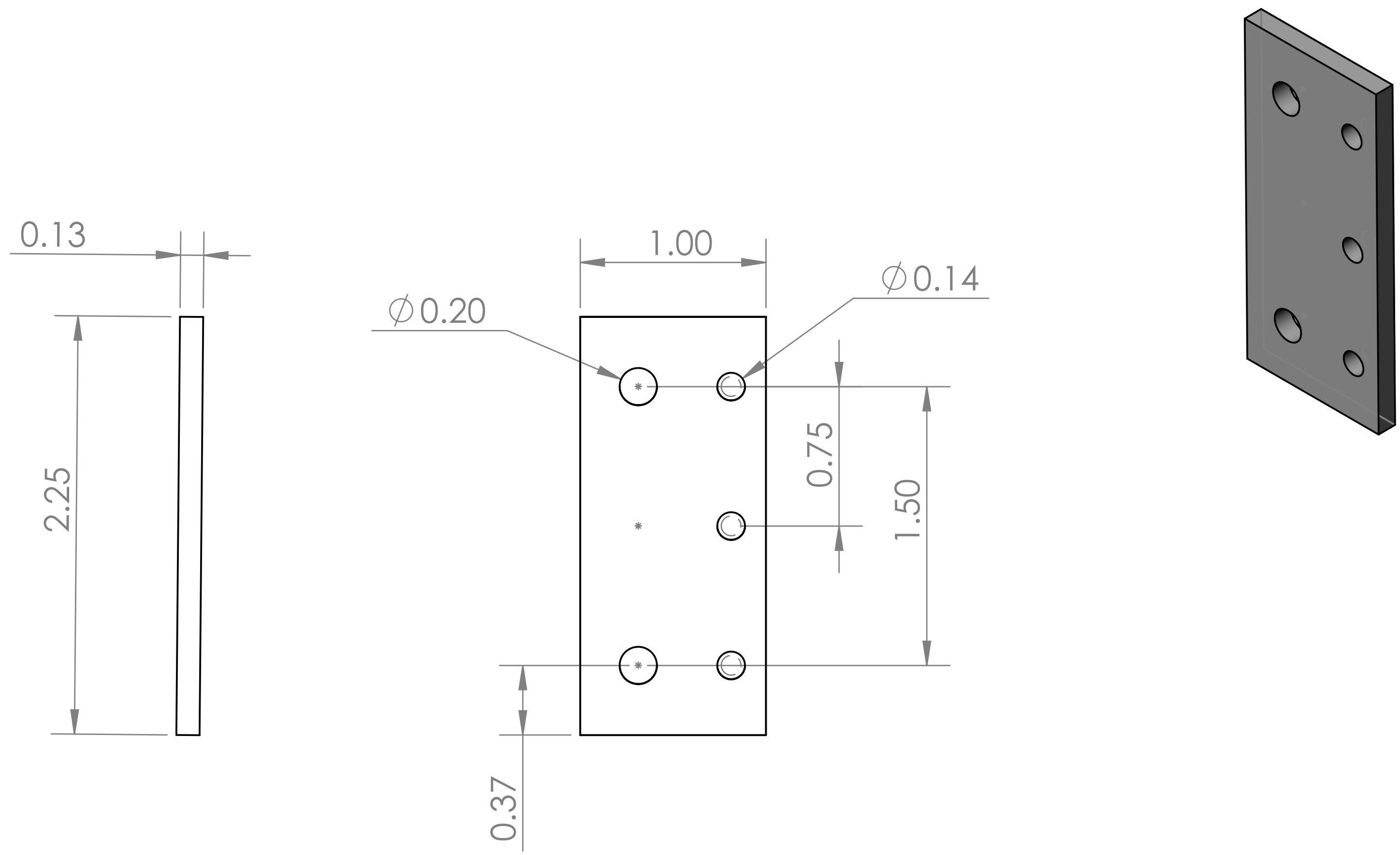
SOLIDWORKS Educational Product. For Instructional Use Only

Arboretum Microgrid		
AC Node Layout		
SIZE A	DWG. NO. AC Node	REV. 1
SCALE:1:2.25	Qty: 1	SHEET 1 OF 1



MATERIAL: ALUMINUM BAR
BREAK ALL EDGES REMOVE ALL BURRS

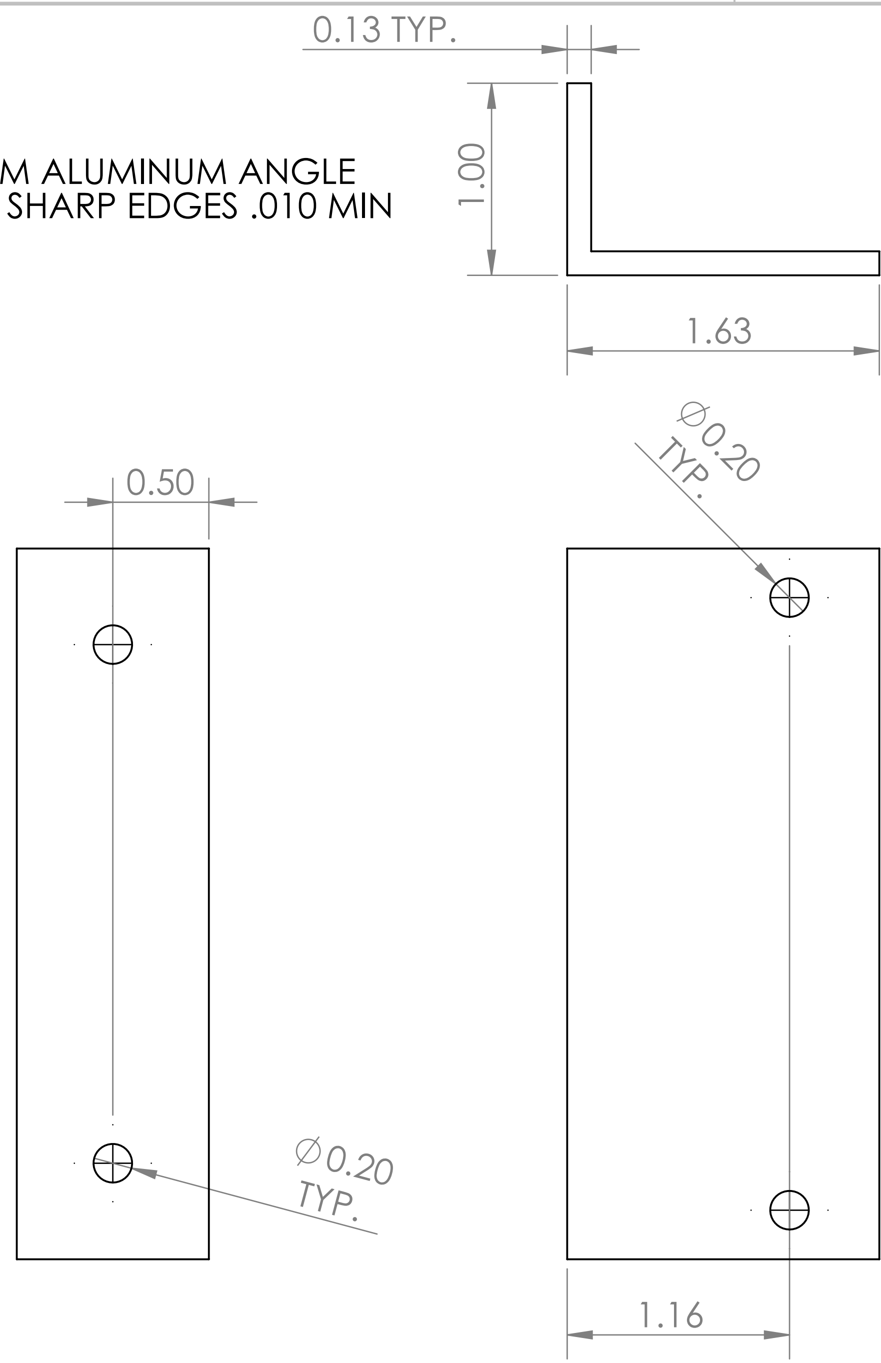
ARBORETUM MICROGRID	
TITLE: 2 BREAKER MOUNT	
SIZE A	DWG. NO.	REV 1
SCALE: 1:1	DATE: 6/20/17	SHEET 1 OF 1

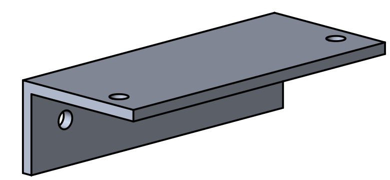


MATERIAL: ALUMINUM BAR
BREAK ALL EDGES REMOVE ALL BURRS

ARBORETUM MICROGRID		
TITLE: 3 BREAKER MOUNT		
SIZE A	DWG. NO.	REV 1
SCALE: 1:1	DATE: 6/20/17	SHEET 1 OF 1

MAKE FROM ALUMINUM ANGLE
BREAK ALL SHARP EDGES .010 MIN

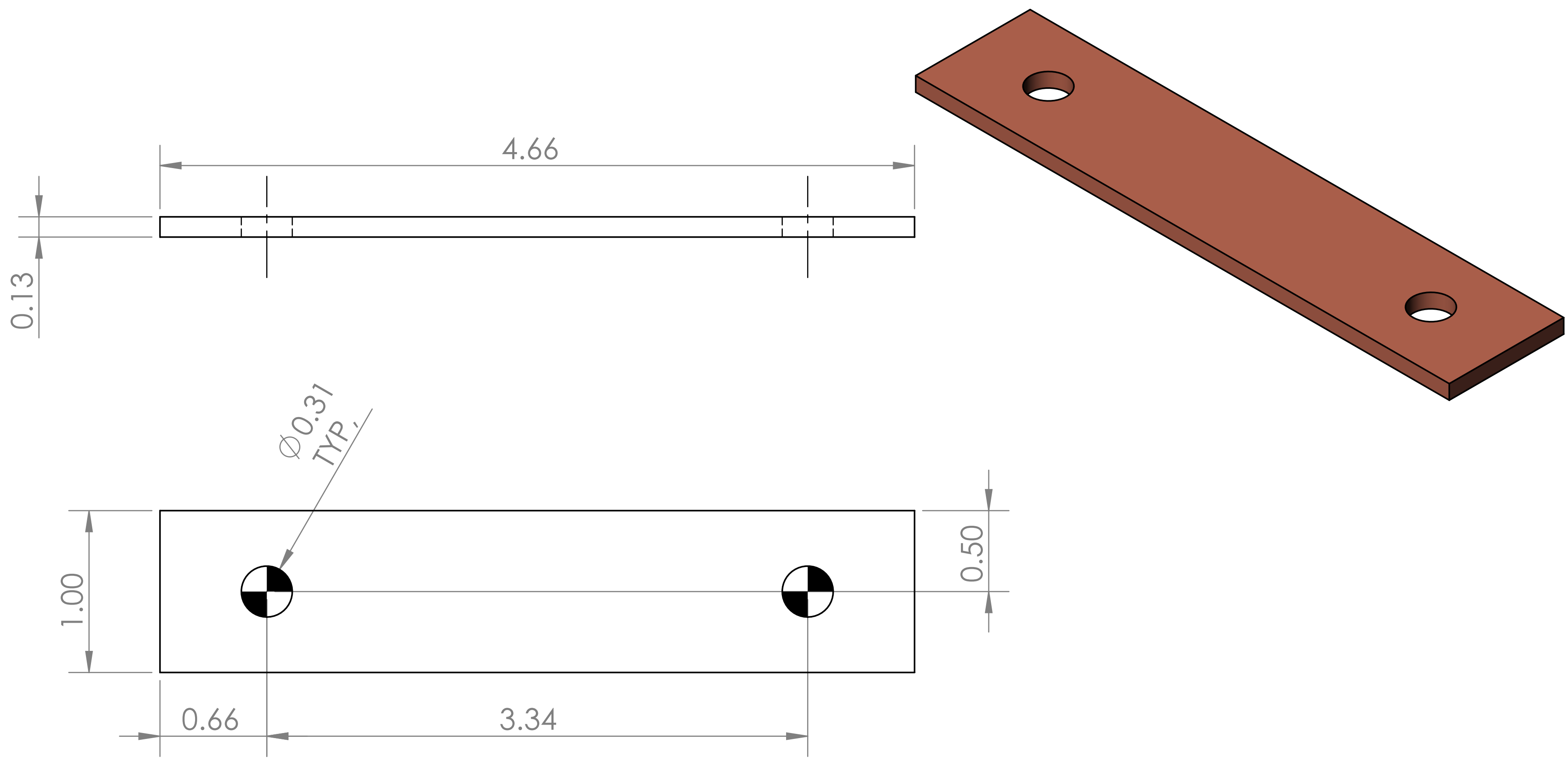




SCALE 1:2

SOLIDWORKS Educational Product. For Instructional Use Only

ARBORETUM MICROGRID		
TITLE: Diode Bridge Bracket		
SIZE A	DWG. NO.	REV 1
SCALE: 1:1	DATE: 6/20/17	SHEET 1 OF 1



MAKE FROM COPPER FLAT BAR
BREAK ALL EDGES REMOVE ALL BURRS

ARBORETUM MICROGRID		
TITLE: Series Bus Bar		
SIZE A	DWG. NO.	REV 1
SCALE: 1:1	DATE: 6/20/17	SHEET 1 OF 1

12.2.1 Templates

Will include links to installation templates for manufacturing. Will be referenced in assembly process and SOP. Will be hosted on drupal.

1.

12.3 Resources

12.3.1 Manuals

Solar Charge Controller: TriStar MPPT

Inverter: SureSine 300

BMS: Emus BMS

Batteries: CALB 180Ah

12.3.2 Data Sheets

Will include links to data sheets for components. Will be hosted on drupal.

12.4 Useful Links

The Sensor Node library can be found here: SensorNode Library

Solar Panel Manufacture: Soliculture

Solar Charge Controller and Inverter Manufacture: Morningstar

BMS Manufacture: Elektromotus

References

[1] T. L. Gibson and N. A. Kelly, "*Solar photovoltaic charging of lithium-ion batteries*", in Vehicle Power and Propulsion Conference. 2009. IEEE, 20 October 2009

[2] M. Shahbazi. B. Kazemtabrizi. and C. Dent, "*Coordinated control of DC voltage magnitudes and state of charges in a cluster of DC microgrids*", in PES Innovative Smart Grid Technologies Conference Europe. 2016. IEEE, 9 October 2016

[3] Tan Ma. M. H. Cintuglu. and O. A. Mohammed, "*Control of a hybrid AC/DC microgrid involving energy storage and pulsed loads*", in Transactions on Industry Applications(Volume: 53. Issue 1). Jan.-Feb 2017. IEEE, 27 September 2016

[4] C. R. Landau, "*Optimum Tilt of Solar Panels*", Web. http://www.solarpaneltilt.com/, Mar. 2017