LTI Microgrid

Final Report



**Submitted to:** Professor John Kennedy

**Due by:** December 19th, 2012

**Team Members:** **Hardware Team Software Team**

Steve Schulz Erik Mendoza

Angel Rodriguez Alvin Del Rosario

Leo Ulman Michael Bautista

Nick Apuzzo Martin Padilla

Javier Vizcaya

Jonathan Gibbs



Table of Contents

[**Abstract 1**](#_Toc343688944)

[**Problem Description 1**](#_Toc343688945)

[**Hardware Design 2**](#_Toc343688946)

[**Solar Panel Integration 2**](#_Toc343688947)

[**Battery Array Integration 3**](#_Toc343688948)

[**User Interaction Interface 4**](#_Toc343688949)

[Figure 1: User interaction PCB 6](#_Toc343688950)

[**User Interaction Microcontroller 7**](#_Toc343688951)

[**FT232 UART Communications 8**](#_Toc343688952)

[**Coulomb Counter 9**](#_Toc343688953)

[Table 1 – Amplifier’s Output 10](#_Toc343688954)

[Table 2 – Amplifier with Potentiometer Output 11](#_Toc343688955)

[Table 3 – Dual Amplifier with Potentiometer Output 11](#_Toc343688956)

[**Voltage Sensor for Coulomb Counter 11**](#_Toc343688957)

[**Tree Design 12**](#_Toc343688958)

[**Software Design 14**](#_Toc343688959)

[**System Communications 15**](#_Toc343688960)

[Hamachi 16](#_Toc343688961)

[Server/Client-Based Communication 16](#_Toc343688962)

[**Island Sub-System 17**](#_Toc343688963)

[Project Hermes 17](#_Toc343688964)

[Light-O-Rama 18](#_Toc343688965)

[AutoIt 19](#_Toc343688966)

[**Base Station Sub-System 21**](#_Toc343688967)

[Figure 2 – Base Station Sub-System Block Diagram 21](#_Toc343688968)

[LTI Data Listener 21](#_Toc343688969)

[Project Prometheus (Solar Panel Maintenance Utility) 22](#_Toc343688970)

[**Website Sub-System 23**](#_Toc343688971)

[Figure 3 – Website Sub-System Block Diagram 23](#_Toc343688972)

[LTI Independent Light Controller 23](#_Toc343688973)

[User Interaction Website 24](#_Toc343688974)

[**Testing and Troubleshooting 26**](#_Toc343688975)

[**Software Testing and Troubleshooting 26**](#_Toc343688976)

[**Hardware Testing and Troubleshooting 27**](#_Toc343688977)

[**Budget Analysis 27**](#_Toc343688978)

[**Conclusions and Recommendations 28**](#_Toc343688979)

[**Appendices 30**](#_Toc343688980)

[**Mega Tree Design 30**](#_Toc343688981)

[**Star Topper Design 31**](#_Toc343688982)

[**Coulomb Counter Design 32**](#_Toc343688983)

[**Bill of Materials 33**](#_Toc343688984)

[**LTI MICROGRID Load Calculations 35**](#_Toc343688985)

[**LTI MICROGRID Consumption Calculations 35**](#_Toc343688986)

[**Incentive Calculator – CSI Standard PV 36**](#_Toc343688987)

[**GC2-XHD Charging Recommendations 37**](#_Toc343688988)

[**GC2-XHD Voltage to State of Charge Table 37**](#_Toc343688989)

**Gantt Chart………………………………………………………………………………………………………………………….……….38**

Abstract

SDG&E has sponsored LTI Microgrid, a collection of senior design students at San Diego State University, and given us the task of creating an independent, solar powered, self-sustaining Microgrid on an island located in Santee Lakes. The system that we are designing for the island will harness renewable solar energy and convert it to power up a Christmas light display on the island.

This system will also be capable of communicating with both a sub system at the university as well as a website in which users will be able to interact with and change the light display. In addition to user interaction, the light show will also move in rhythm to a predetermined set list of Christmas music that will be locally broadcast over FM radio.

While the light display will likely be this projects main attraction, its most important aspect is to educate and raise the awareness of both young engineers and members of the community to the importance and functionality of renewable energy.

Problem Description

To power holiday lights display on Santee Lakes Island, we use 2 solar panels with 4 115AH 12V Deep Cycle Batteries. These batteries will provide power for the laptop on the island and Light-O-Rama (LoR) using DC to AC Power Inverter to provide control of the C9 LED lights. The MorningStar 60A Tristar charge controller will monitor the charge rate and level of the solar units’ battery. This data will be obtained by the laptop via RS232 to USB communication and utilize the Verizon hotspot service to populate the OSI-PI Database located at SDSU.

The Solar Panel Maintenance Utility will use allocated data for an administrative display in order to monitor and process energy generation and consumption. As this design relies heavily on solar energy, battery and weather forecast data will be integrated to determine the quality of the light show of any given day to ensure a four hour event. The light display will be in conjunction with TX-01S FM Transmitter that will broadcast music synchronized to LoR. In order to provide user-friendly interaction, independent lights will be featured in a similar manner with the exception that viewers will be given control of the system to manipulate its lighting display as they wish via website.

Hardware Design

Starting at the solar panels, there are two 235 watt sharp solar panels set in parallel to optimize the production of current. From there the generated power travels to the charge controller, which not only monitors and protects the charging process of the batteries, but provides us with valuable information in regards to that process. Some of that information includes: enclosure temperature, charging amperage, charging voltage, total amp hours produced and total kilowatts hours produced.

Our 12V battery system consists of four 115AH marine deep cycle marine batteries connected in parallel, creating one large 352AH battery at 20 hours. Connected to our battery our team designed a coulomb counter, which provides real time consumption data. That allows us to monitor the consumption of our light show, cross reference the charging data from the charge controller and allow us to determine what levels we are at in terms of energy produced to energy consumed.

From our coulomb counter the power is distributed to the user interaction control lights and inverter which converts the DC voltage to 120V AC. The majority of the system relies on the converted power, including: the Island Laptop, FM Transmitter, and Light-O-Rama.

Solar Panel Integration

In order to produce sustainable power for our system, we decided to implement the use of photovoltaic power generation. The choice of photovoltaic generation was chosen over other types of sustainable energy due to the budget, typical weather history and environmental impact factors. Using our load calculations we determined the necessary generated power needed for our system to be run consistently and without interruption. In order to calculate the amount of energy our proposed system would produce, we used the “CSI-EPBB” calculator (<http://www.csi-epbb.com/default.aspx>).

This program accurately calculates the monthly kilowatt hour production by factoring the solar panel PVUsa Test Conditions (PTC) rating, solar panel and inverter efficiency, and solar panel azimuth and tilt angle. The program also accounts for the location of the photovoltaic generation to determine for the annual change of the suns path in the sky. With this calculator we determined that our proposed system would produce at least 52 kilowatt hours for the month of December, which was over our conservative load calculations.

Our solar panel final design consisted of two 235 watt Sharp NU-U235F1 solar panels connected in parallel to optimize the production of current. This parallel connection made the two panels act as a single 470 watt solar panel and to accommodate this in our CSI-EPBB calculator we had to use a single panel with a PTC rating similar to the addition of the PTC ratings for both individual panels.

Battery Array Integration

In order to determine the correct about of amp hours needed to supply energy for our light show we developed a load calculation table using Microsoft Excel. This table allowed us to implement calculations and add our loads together, bringing them to a common metric where we could determine the correct size and amount of batteries needed to hold enough power for our system. During the calculation we used “worse case scenarios” to guarantee our systems success. For instance, the load calculations account for the lights to be turned on consistently during the entire show when in reality the lights are periodically flashing.

We decided to connect the 12 volt batteries in parallel to maintain a 12 volt system to limit the design complications of the user interaction control lights and coulomb counter while minimizing the cost of the inverter. When connecting the batteries in parallel, each unit combines and turns into a single battery array. We chose deep cycle marine batteries because their chemical composition is robust enough to handle the consistent charge and discharge our system experiences. The size and manufacture of a single battery was mainly driven by budget and therefore we specified 115 amp hour Kirkland Signature Deep Cycle Marine batteries. Our battery array final design consisted of four 115 amp hour deep cycle marine batteries connected in parallel creating a single 352 amp hour battery at 20 hours.

User Interaction Interface

In addition to our main lighting element on the MegaTree, we devised a second element that would allow spectators to directly interact with the holiday light display. We specifically developed the user-interaction portion to allow show viewers to directly manipulate key lighting elements strategically placed throughout the holiday display. This was done to add an element of uniqueness to our display in which we believed would create more of a memorable experience for the observer.

To implement our design idea we developed an embedded control system comprised of the following main components:

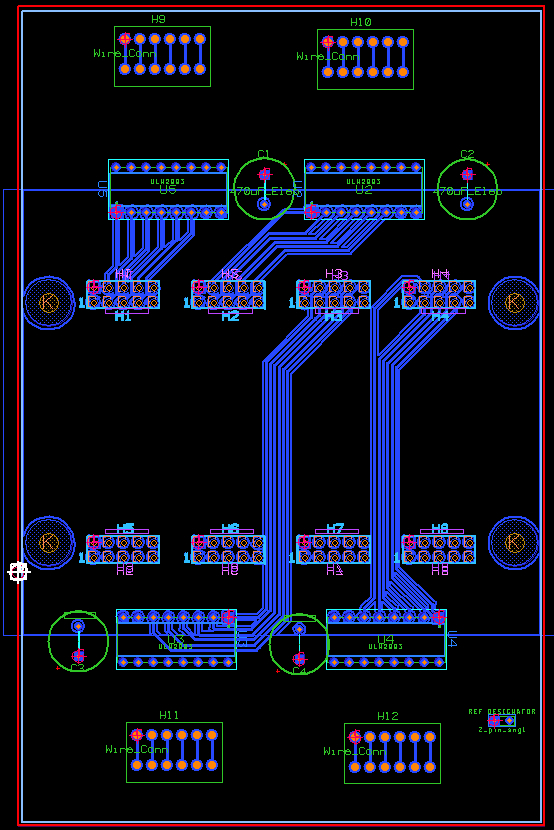
* PIC Microcontroller PIC16F887
* ULN2003 Integrated Circuit (IC)
* Printed Circuit Board (PCB)
* 48-Port Private Branch Exchange (PBX) Patch Panel
* DC-DC Regulated Power Supply
* Universal Asynchronous Receiver/Transmitter (UART)

The microcontroller we selected for this task was the PIC16F887, which was installed on a Logic Root PCB designed by our instructor John Kennedy. The reason we selected this board was for the large number of outputs pins (24 from each PIC).

The datasheet on the microcontroller revealed that output pins are limited to approximately 25mA of current, which would not be strong enough to directly drive the additional interactive lighting elements. To solve this issue we selected the ULN2003 IC from Texas Instruments. The chip acts essentially as a high speed/high power switch. Among the input and output of the ULN2003 sits seven independent NPN Darlington transistor pairs. Each one of the Darlington pairs performs as a single transistor capable of supporting a high current gain that is capable of withstanding a sustained 500mA current drain and a maximum 50V making it a perfect choice to drive multiple LED lights.

When the ULN2003 receives a high input signal or a logic 1 from the microcontroller the corresponding output pin is driven to ground allowing current to flow and thereby turning on the LED light. Conversely, as a low input signal or logic 0 is received from the microcontroller the corresponding output pin develops a high input impedance thereby preventing current flow and turning off the LED light. The default state of the ULN2003 is set to high input impedance or off, and remains in that state until a signal is received from the microcontroller.

We designed and developed a PCB using the design program Mentor Graphics to directly mate with the header pins from the Logic Root PCB. The layout of the PCB contained four of the ULN2003 ICs mapped to the output pins of the microcontroller, four 470uF capacitors designed to help reduce unwanted residual DC periodic variation known as ripple, and power connections for both input and output power. The design layout is shown below, in Figure 1.



### Figure 1: User interaction PCB

Due to the proximity of the display in relation to a spectator’s vantage point we wanted to create lighting element that would be very eye-catching. Our research led us to select mini work-lights/flashlights from Harbor Freight Tools.

Testing revealed that each work-light consumed approximately 100 to 150mW of current at 5V. Multiplying an average of 125mW by the maximum number of LED lights used (48) would give a total current draw of 6A or 30 watts if all lights were to remain in a constant on state. The overall power consumption was calculated to be 30 Watts \* 1 Hour = 30 Wh or 0.03 kWh. The combination of low power consumption, extremely brilliant and reflective output, and inexpensive price point made these LED lights an excellent choice.

To power the interactive lights we selected a high power buck-boost DC-DC converter from mini-box.com. What made this power supply ideal was the ability to accept a range of inputs from 6-34V DC and provide a selectable regulated output ranging from 5-24V and up to 15A max current draw. We experimented with a variety of voltages ranging from 5-9V and as we increased the voltage we observed a dramatic increase in brightness of the LEDs but at an expected cost of more current consumed. We ultimately selected to power the LEDs with 5V to keep power consumption levels to a minimum.

Power from the PCB was distributed to the LED work lights through a PBX phone patch panel. The PBX patch panel was selected as a versatile and cost effective means of delivering power to the LED lights. RJ-11 cables were created in custom lengths and bridged the gap between the work lights and the corresponding PBX panel port.

User Interaction Microcontroller

We were asked to implement a system capable of allowing for user interaction from a website. We could transmit data via TCP/IP communication to a terminal communicating to our USART board that would then transmit the necessary output to be read by a receiving USART device. So we had the task of choosing a device capable of implementing USART protocol and also capable of being programmable to turn on and off digital outputs as required of our patch panel design.

The microcontroller we used to accomplish this task was the PIC16F887, and these were then, along with the USART board, integrated into our independent light controller. Our final design included 6 different tree light shows and 6 different present light shows. Our 6 different programmed lights shows included: on, off, twinkle, chase, shift, and random. They were also programmed so that the default action would turn the lights off in order to conserve energy from our batteries.

To implement our different light shows it was important that we create an appropriate manner from which to call each individual function as the appropriate input was read. The USART receive interrupt flag would be set to 1 whenever a character was sensed as being inputted and the microcontroller would constantly wait for this flag to be set. When the microcontroller sensed the flag as being 1, it would read the character from the RCREG containing an 8-bit character that would then be passed through a switch case function.

We also had the challenge of creating a second light show to run from the main tree light show. These were the present light show that would also have the same options as the main tree show; specifically, the presents should also be able to turn on, off, twinkle, chase, and random.

Ultimately, we found a better solution to our problem. We found that we could call the present function to run for a specific period of time and have the default action for the main tree lights to only be turned off. In this way, instead of writing 36 different light shows for each specific main tree light show, we only had to write an additional 6; more importantly, this was also seen as a more aesthetics solution since turning off the main tree lights as the presents turned on created a pleasant effect, which ultimately was the design goal.

FT232 UART Communications

Our independent light controller needed a way to receive a signal from our laptop, so we used an FT232 USART device to create this connection. Using a C# terminal, we opened up the COM port necessary for transmitting data. Our USART board received a character from our laptop which was then transmitted to both of our microcontrollers simultaneously. So part of the design also included that the microcontroller include a USART interface capable of reading the input received and executing the appropriate light show given the user input.

The USART interfacing was necessary for both our coulomb counter as well as our independent light controller. The USART breakout board that we used for both design ran on 5V which was convenient for our PCB design; it was easier to supply our circuit with 5V DC along with our USART. Within our design, we did not implement the use of the dynamic link libraries for our designs since we felt it would add unnecessary complexity to our solution. The most important aspect of our USART interface was the opening of a COM port to communicate to our laptop computer. Using a virtual created COM port from a USB plug-in allowed for easier implementation of our design using our microcontrollers from which our USART interface was solely concerned with.

It was also of large importance that the COM port from the USART would read or write from was correctly addressed from the terminal program. The terminal program was written in C# using visual studios and created an interface to which from both read or write to the USART as necessary. This was convenient since it allowed for manipulation of data as necessary for our design either as the data was coming in or out of our USART. This overall design allowed for versatility in our design when it came from the need to communicate freely from our microcontroller or to them in any way we saw fit. In this way, a lot of applications could have been possible.

From our C# terminal we also had to write the correct baud rate for communication, and in this case it was a 9600 baud rate. This baud rate not only had to be set-up correctly in the terminal but it most also be set up correctly inside of the microcontroller. The microcontrollers must also be initiated in the correct way for communication to occur properly.

Coulomb Counter

The main objective of this project is to monitor and provide enough energy to the light displays for four hours. We designed a coulomb counter device that monitored the amount of coulomb leaving the battery. This information allowed us to determine the state of charge of the battery by comparing it with the battery's rated kilowatt hours or with the coulomb going into the battery.

First, we calculated the total amount of current that the loads are using. The rated maximum current of the microgrid is 30 amperes. We used three 0. 01 Ω, 4 watts Ohm shunt resistor to sense the current leaving the battery. From ohm's law, we determined the total power dissipation between the shunt resistor and its voltage drop at 30 A.

V= IR = (30 A) (0.003333 Ω) = **0.1 volts** P= VI = (0.1 V) (30A) = **3 W**

The reason we used three shunt resistors in parallel to each other is to lower the power dissipation between the shunt resistors. It was important not to dissipate over the shunt resistor's rated power dissipation and also not to take much energy from the system. We also used a MPC602 dual differential amplifier and a PIC12F1822 microcontroller. The Coulomb Counter Figure in the Appendices illustrates the schematic of the coulomb counter.

The voltage between the shunt resistors is the voltage input of the differential amplifier. The chosen resistor’s are R2=R3=330 k and R1=R4=100 K. The amplifier amplifies the shunt resistor's voltage drop. Its output is scaled between 0 to 5 volts so that it does not go over the microcontroller's power supply. The table below shows the first measurements of the coulomb counter.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| I (A) | 0.08 | 1 | 2 | 3 | 4 | 5 | 6 |
| Vout (mV) | 4.87 | 5.15 | 7.31 | 10.32 | 13.29 | 16.29 | 19.32 |

### Table 1 – Amplifier’s Output

As you can see, as the current increases the amplifier's voltage has an increment of 3 mV. But at lower current, between 0-1 A, it does not have an increment of 3 mV. To improve this, we added a potentiometer of 1 K Ω, R5=33 K, and R6= 1 K so that R4 will sense 20mV instead of ground. These are the improved measurements:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| I | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| mV | 23.36 | 26.34 | 29.8 | 32.8 | 35.8 | 38.8 | 41.8 |

### Table 2 – Amplifier with Potentiometer Output

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| I | 0.09 | 1.03 | 2 | 3 | 4 | 5 | 6 |
| V | 1.03 | 1.17 | 1.31 | 1.46 | 1.62 | 1.77 | 1.92 |

### Table 3 – Dual Amplifier with Potentiometer Output

The added potentiometer and resistors improved the amplifier's voltage output to have an increment of 3 mV. Another amplifier is used to linearly amplify the first amplifier's output. The chosen R6=8K Ω and R7=400K ohm. The amplifier has a gain of 50 v/v. Table.3, shows the measurements of the dual amplifier.

The second amplifer's output voltage is inputted into the microcontroller's digital-to-analog converter. Then, the analog values are sent to the island's laptop via UART. The laptop calculates the state of charge and sends this information to SDSU's PI server. The brightness of the light show depends on the state of charge. The SDSU laptop will send the laptop a command stating the light show's brightness.

Voltage Sensor for Coulomb Counter

For our Coulomb counter microcontroller, we used the PIC12F1822 to be programmed as a voltage sensor device. We also had to program it to output a character corresponding to the voltage that was sensed by the microcontroller. So, the microcontroller would read a voltage from the analog to digital converter result register and would compare that to a look-up table that would output the correct voltage value out from its USART output.

The most important task for this to function was to appropriate the correct registers as the outputs for the transmission of the character. We used the USART capabilities of the PIC12F1822 by appropriating the appropriate register to use the transmission protocol. More specifically we chose to have RA2 as our transmission pin as specified by the data sheet. Furthermore, we had to select an analog input from which the operational amplifiers voltage would come in and we chose AN0 for that purpose. The resulting value of the analog to digital conversion is stored in the ADRES register which is a percentage value of 5V. For example, if the ADRESS value read as 512, the resulting voltage value would be calculated as (512/1024)\*5 or 2.5V.

Furthermore, we created an appropriate look up table designating the appropriate input voltage value from the resulting ADRES register to the appropriate output character that would be read as a current value from the receiving laptop. The value of the current was calculated via ohm’s law appropriately from the design of the coulomb counter circuit. In this specific case, we had a resistance value of .0033 ohms using three shunt resistors in parallel whose values equaled .01 ohms. It is also important to note that the microcontroller operates on the TIMER0 interrupt function which is called every time the interrupt flag detects an overflow.

This overflow occurs every 65ms as we selected via appropriately setting our option register as our prescaler and our internal frequency Fosc for 4Mhz. Since we also set our registers to allow for interrupts, everytime our TIMER0 interrupt flag is set, our interrupt service routine will function. Within this routing is contained our look up table from which the appropriate voltage character is output every 65ms, or on every interrupt call.

Tree Design

The request for proposal given to us stated that we needed to create a lighting display on an island in the Santee Lakes Reserve Park. Our initial thought was to hang a certain amount of C-9 colored LED strands on a 40-foot eucalyptus tree that is sitting already on the island in order to make it seem as a Christmas tree. We later projected that it was going to be nearly impossible to hang our desired light strands on the tree itself due to the fact that it had a numerous amount of offset branches, the proximity it had to the water would have made the light display a hazardous task to complete and ultimately would not look appealing for our client.

We then tried to brainstorm and concluded we would try to build our own tree display next to the eucalyptus tree. Immediately we decided that the tree would have to follow certain requirements in order for our idea to be possible. First of all, it would have to be at least as tall as the eucalyptus tree next to it in order to be mostly visible for our desired audience. Secondly we would have to provide a steady enough base for our tree to prevent from any possible collapses. As we went along with this idea, we discovered we were heading straight for a dead end, that is, the city of Santee required us to have a permit to be able to excavate on the island. If we had taken this route, it would have taken us many more weeks to even obtain a digging permit and finish our project. Ultimately the board agreed and we opted to get a 5 by 5 feet steel platform on which our pole would stand. The final design of our MegaTree can be found in the Appendices.

Our process for the pole setup started with using fishing line. We had our pole flat on the ground and we measured double the height of the tree with the fishing line. After we had the necessary lines hanging from our top plate we then raised the pole. We situated the base were we saw it would be the best fit according to the flattest ground on the island and the total diameter the tree would need. We then started to raise each one of the light strands by tying them to the corresponding fishing line we had previously designated for them. The whole set up of the tree took about 3 weeks to conclude.

The work-lights we used came with 2 AA batteries, which were integrated to provide the necessary power to the light. We took upon the task of completely disassemble all 40 of them. This came out to be a tedious task since we had to unscrew every light, take out the batteries, carefully desoldering the battery leads off the circuit board and soldering instead leads from “rj-11” (telephone cable) pigtails. The pigtails were of crucial importance since these were the inputs of cables coming from a patch panel that controlled the lights by using the programmed PIC16F887 microcontroller.

Software Design

The Software portion of this project consists of three different sub-systems that work together in order to satisfy the project requirements:

* **Island Sub-System**
  + Project Hermes
  + LTI Lightshow Override
  + LTI Independent Light Listener
* **Base Station Sub-System**
  + Project Prometheus (Solar Panel Maintenance Utility)
  + LTI Data Listener
* **Website Sub-System**
  + Project Aether (User Interactive Website)
  + LTI Independent Light Controller

The Island Sub-System will obtain Metric Information from the Charge Controller and Coulomb Counter that the Hardware Team has put together. The main purpose of the Island Sub-System is to transmit this data to the Base Station Sub-System, where it is then stored into the OSI PI Database. Another functionality of the Island Sub-System is to receive instructions that originate from either Project Prometheus, or the User Interactive Website in order to manipulate the light show.

In order to communicate the Charge Controller and Coulomb Counter Metric Information, TCP/IP Communication was used utilizing a Server/Client-based setup. A Virtual Private Network program called ‘Hamachi’ was used in order to ensure that none of the data was blocked by the SDSU Firewall.

After receiving the Metric Information from the Island Sub-System, Project Prometheus (Solar Panel Maintenance Utility) utilizes the OSI PI SDK (Software Development Kit) in order to maintain and store the data inside the OSI PI database. After ensuring that the data is properly allocated, values seen on Project Prometheus will change accordingly so that an Administrative User can decide which Light Show should be used for that night.

The Website Sub-System also utilizes the OSI PI SDK in order to read values from the OSI PI Database. In addition to Metric Information, boolean values are also stored into the database which correspond to specific features on the User Interactive Website. Through various pages on the website, users are able to manipulate some of the independent lights on the tree.

Both Project Prometheus and the User Interactive Website Sub-Systems send Light Control data and overrides back to the Island Sub-System. Priorities are set in place so that the Administrative User will always have Top-Level Control over the system. Project Hermes, which is being run on the Island Sub-System accepts requests for the light show, and acts accordingly.

System Communications

One of the big challenges the Software Team faced during the design and implementation phases of the project was getting around the SDSU Firewall. The Firewall at SDSU is set up in order to prevent attacks from people outside of the network, and is generally considered a must for network security anywhere. Unfortunately, this meant that the Firewall was also blocking communications coming in from the Island Sub-System, and the Base Station Sub-System was unable to receive any of the Metric Information.

Hamachi

In order to solve this problem, and also reach one of our milestones, we utilized a Virtual Private Network program called ‘Hamachi’ in order to create a “Wireless LAN” between the Island Sub-System and Base Station Sub-Systems. In essence, rather than utilizing the actual IP addresses registered to our laptops, we used Hamachi-based IP addresses that were only used in the Private Network.

Not only did this solve our issues for communications, but it also had the added advantage of providing us with extra security. In Hamachi, we are able to create our own networks and also manage its users. Only people within the network are included inside the communications system, which meant that any other incoming transmissions from outside IPs are essentially blocked. Hamachi-based IP addresses are also static, meaning we won’t have to constantly designate new IP addresses in our programs.

Server/Client-Based Communication

Once the Virtual Private Network solution was found, we utilized C# .NET libraries for Socket Communication in order to establish our connections. In order to receive data from someone, you require a Server. If you wish to transmit data, you require a Client. Both the Island and Base Station Sub-Systems have Servers and Clients for communications running in order to facilitate the functionality of our system.

The way the Server works is that a TcpListener object is created in a background thread. This thread then accepts communications from any IP address, but only through a specific port specified in the code. Once a message is received, a TcpClient object is received, which reads the data contained in a NetworkStream object. This object contains the data received from another user.

As for the Client, a TcpClient is also used, but primarily for connecting to a specific server endpoint. Once a connection has been established, the Client program then writes into the NetworkStream, which is then read by the Server program.

Island Sub-System

Project Hermes

Project Hermes (PH), hosted on the "Island" laptop, is the mainframe of the LTI system. PH was programmed in C# to handle several functionalities that are:

* Read/parse charge controller data
* Transmit charge controller Data
* Activate light show based on system's state of charge and timing
* Override light show display from Administrator's command

**Read/Parse Charge Controller Data:** MSView, a state and logger software for the TriStar Charge Controller, was set to record battery data every minute to provide a granular representation of battery information. Prior to reading and parsing data, PH first copies the original log file into a separate directory. The reason for this is that MSView keeps the current recording log file opened, unable to be processed. This is repeated every time new information is saved. PH obtains this copied log file and process it every minute to deal with new incoming information into a readable format for both local and administrative access. The log file is parsed through the use of substrings and indices to easily allocate any desired battery information. Once the data is successfully parsed it is stored into a list awaiting command to be transmitted to the SDSU laptop.

**Transmit Charge Controller Data:** Once the log file has been successfully parsed, a timer of 60 seconds is initiated. Since the MSView records battery information every minute, a 60 second timer to transmit new parsed information was an ideal choice. Once the time limit expires, the data stored in the list is transmitted to the SDSU laptop through TCP/IP Communication and usage of the mobile hotspot.

**Activate Light Show Based On System's State of Charge and Timing:** Every night throughout December 2012 from December 8th, the system is automated to activate a light show from 5PM-9PM. Of course, this portion of the system is widely based on the battery array's state of charge. Plan A, was to obtain the state of charge from the coulomb counter and charge controller. Plan B, was to use the inaccurate state of charge table provided by the battery manufacturer. Due to time constraints, the team decided to proceed with Plan B. When the state of charge is at 100%, 80%, and 60%, a script will execute to run the appropriate light intensity light show. For system reliability, if the battery array is ever at a critical level, justified to be 40%, a script will execute to disable the light show to conserve power.

**Override light show display from Administrator's Command:** PH was made to be interactive. Using the same TCP/IP Communication, the Administrator accessing the SDSU Laptop can override the light show on command. That is from 5PM-9PM, the Administrator can transmit a command to the "Island" Laptop to execute selected light show intensity. Once a command is transmitted, a script based on the command input will execute.

Light-O-Rama

With all the technical and necessary components to build a system of renewable energy, the energy stored in the battery array will produce a 4 hour light show through the use of the Light-O-Rama. Light-O-Rama (LoR) is a mega controller with 16 output channels that served as our animator of light shows. The LoR can controller lights through several different functionalities such as: blink, fade, twinkle, and on. Additionally, the functionalities can be combined to provide a dynamic light display.

In the design phase, the team unanimously voted to have the C-9 LED lights to be in conjunction with music. With the Light-O-Rama, we managed to program 19 unique sequences, an hour worth, to dance to the beat of the broadcasted music.

We programmed the sequences through the use of the "Tap Wizard" provided in the Light-O-Rama Showtime Suite Software. That is, we listen to the music and "tap," press any key, when we hear a desired musical accent we wish to emphasize. At the end of the sequence the collected "taps" are applied on the timing grid of the selected song. We then choose any light functionality we wish to display on the 16 channel output. This process was repeated to build 19 unique sequences. Additionally, the 19 sequences were modified to run the same show but at both 80% and 60% light intensity for power consumption purposes.

In order to see how well our sequences "dance" to the beat of music, the LoR Visualizer allowed us to virtually animate the designed sequence. This feature allowed us to see our programmed light show prior to implementation. This process was *repeated thoroughly* to make sure that functionalities of the light show were on cue with the music.

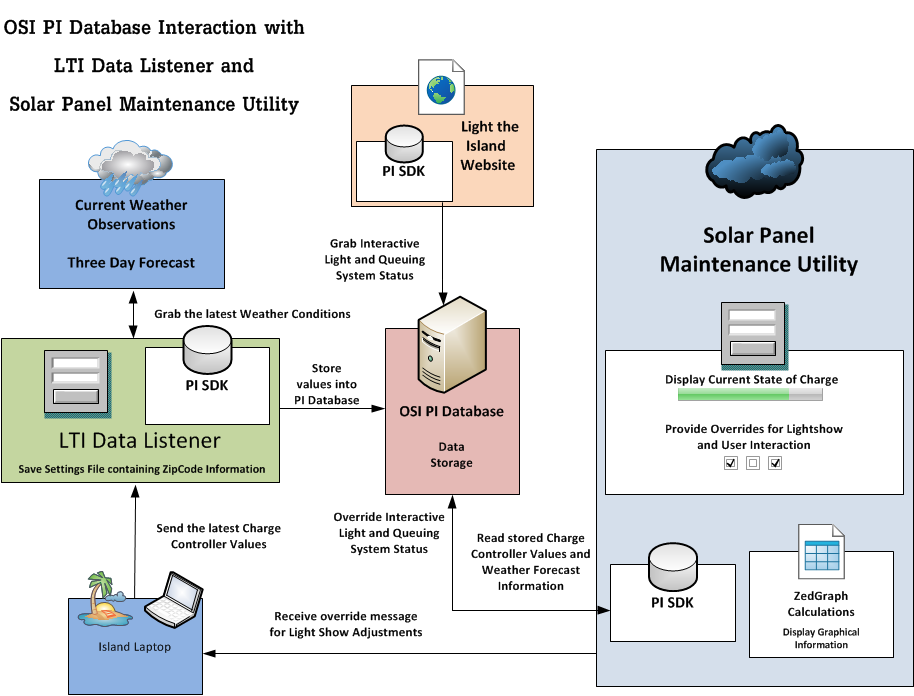
AutoIt

The script functionalities mentioned above were built using AutoIt scripting. With all the components left on an island, AutoIt served as the AI of the system for it to run independently. Scripts were written for the following descriptions.

**Charge Controller Script:** MSView records battery information by creating a log file. Since the system is left alone on an island, MSView will periodically record on the same log file throughout December. To counter this, a script was written to activate every new day, 12:00 AM, to stop the current log file and record a new one saved in the format of MM-DD-YYYY. The purpose of this was so that the administrator can easily allocate the battery performance of the system of any given day as opposed to reading one heavily congested file.

**Light-O-Rama Scripts:** Five Scripts were written specifically for the LoR. Three for executing 100%, 80%, 60% light intensity shows and two for *Stop the Show* and *Stop the Show Gracefully.* If a light show is executed, the script activates the *Show On Demand* and outputs the necessary light show intensity. When the show is about to end the *Stop the Show Gracefully* script is activated to end the show by allowing the last sequence to play its course. *Stop the Show* script, however, stops the light show on demand immediately regardless of how far along the current playing sequence is.

Base Station Sub-System



### Figure 2 – Base Station Sub-System Block Diagram

The Base Station Sub-System shown in Figure 2 consists of two different programs that work together in order to accept and parse incoming Charge Controller and Coulomb Counter information. These two programs are the LTI Data Listener and Project Prometheus. This Sub-System is also able to transmit lightshow overrides back to the Island Sub-System in order to manipulate the power intensity of the nightly show.

LTI Data Listener

The LTI Data Listener is the reception point for the Metric Information, and acts as a server listening on Port 81. The message format that it expects is a long character array that can be parsed through substrings and separated easily. When a message is finally received, the information is further parsed for easy insertion into the OSI PI Database.

In addition to receiving information, the LTI Data Listener proactively grabs Weather Information for an area specified by the user via zipcode. This feature essentially means that the entire system is capable of being relocated to various locations throughout the United States. The weather information is grabbed through the use of an API developed by a company named “WunderGround,” and allows Project Prometheus to act not only as an Administrative Override tool, but also as a planning utility.

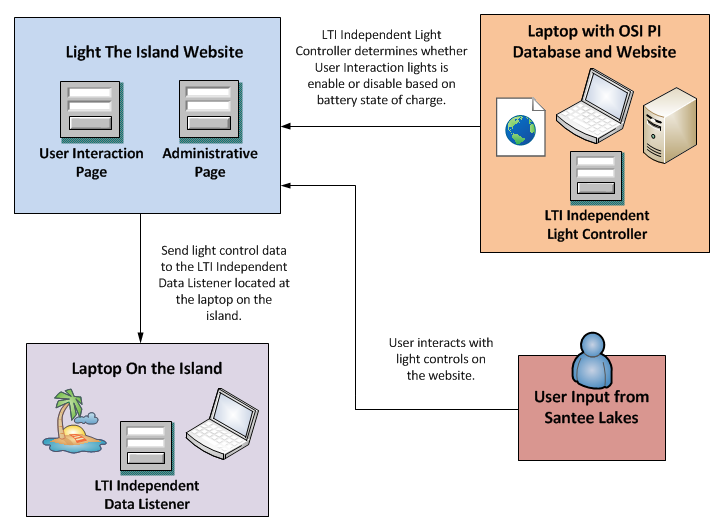
Project Prometheus (Solar Panel Maintenance Utility)

Project Prometheus acts as a window for Administrative Users to view the information lying in the OSI PI Database. Whereas the LTI Data Listener’s purpose was to store incoming data, Project Prometheus’ purpose is primarily based on information retrieval. Utilizing the OSI PI SDK just like the LTI Data Listener, all values stored in the database is easily viewable via Project Prometheus through various displays.

The main display features a status bar for the State of Charge on the battery array system located on the Island Sub-System, and the time the latest values were received. Additionally, there are four buttons located on the left side of the display which enables the Administrative User to override specific portions of the light show for that night. Metric Information that has been received, as well as weather forecasts are easily accessible through these displays.

On the lower right portion of the main display, there is a graphical representation of the incoming data from the Island Sub-System. Radio buttons on the lower left allows the Administrative User to adjust the graph to their liking. Overall, the main purpose of Project Prometheus is to act as an Administrative Override for the system, as well as a Planning Tool for further light shows.

Website Sub-System



### Figure 3 – Website Sub-System Block Diagram

LTI Independent Light Controller

The LTI Independent Light Controller will enable or disable the independent lights on the island based on the state of charge of the batteries. This program first checks if the state of charge is over 60% between 4:59 to 5:00 PM. If the battery state of charge is over 60%, the LTI Independent Light Controller will enable the independent lights on the island and allow users to use the light controls on the User Interaction Page. However, the independent lights will remain off if the battery state of charge is below 60%. Between 8:59 to 9:00 PM, this program will disable the independent lights until the next show.

User Interaction Website

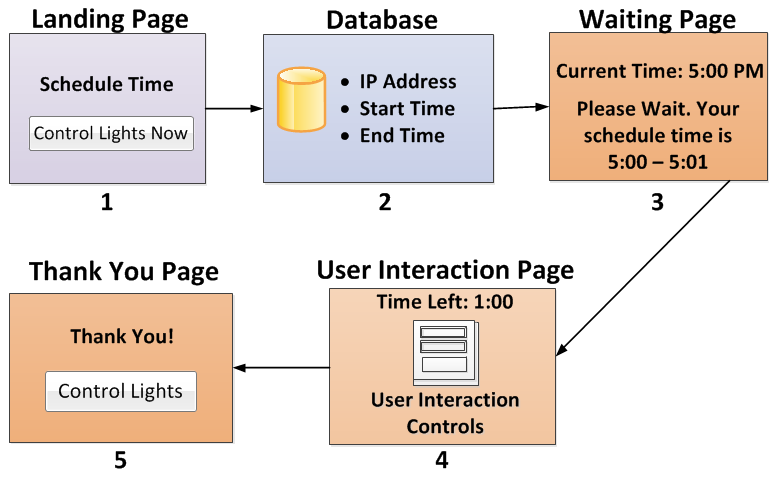
The website will be hosted on the laptop in SDSU and will allow users to interact with the lights on the presents or the mini work lights on the tree. Users will be able to access our website by scanning a QR (Quick Response) code or entering ltinteraction.sdsu.edu using their mobile devices.

Website Queuing System

To accommodate all the users using the User Interaction page, we implemented a queuing system to allow only one user at a time to control the independent lights on the User Interaction website.

The queuing system starts when the user enters the Landing Page. When the user presses the button “Control Lights Now,” the website will first store the user’s ip address into the database and assign the user a start and end time to use the light controls. The website assigns these times by comparing the current time to each user’s end time in the database. If the current time is greater than the latest end time, this means no user is using the light controls at this time. This will make the user’s start time the current time and their end time one minute from the current time. However if one of the user’s end time in the database is greater than the current time, the website will make the user’s start time the latest end time in the database and their end time one minute from the latest end time. When the user obtains their schedule time, they will be redirected to the Waiting Page. This page will redirect the user to the User Interaction Page when their schedule time matches the current time. If it is not met, then the user will have to wait.

In the User Interaction Page, the user can control the lights on the presents or the mini work lights on the tree for one minute. When the user presses a light display button on this page, a character value representing the light display will be transmitted to the laptop on the island using TCP communication. The user’s input will be sent to the microcontroller which will output the user’s light display. Once the time reaches zero or the user decides to end their session, they will be redirected to the Thank You Page. This page would give the user the option to control the lights again or exit out from the User Interaction Page.



#### Figure 4 – Queueing System Block Diagram

*This is a diagram of the queuing system.*

*The user first starts at the Landing Page and ends in the Thank You Page.*

LTI Independent Light Listener

When the server starts, the Independent Light Listener will continuously listen for the user’s input from the User Interaction Page through port 82. Once it receives a character value, this program will send the data to the microcontroller to output the user’s light display. In addition, this program allows the person to clear the textbox by pressing the button “Clear”.

Testing and Troubleshooting

Software Testing and Troubleshooting

**Communication through SDSU Firewall for Charge Controller Data:** In order to communicate the charge controller data, TCP/IP communication will be used with a Server/Client-based setup between the laptop on the island and the laptop in SDSU. We set up a virtual private network between the two laptops using Hamachi. This program ensures none of the charge controller data will be blocked by the SDSU Firewall. Once we establish communication, we were able to populate the OSI Pi database with the charge controller data.

**User Interaction Communication with the Microcontroller:** One requirement of the website is user interaction with the independent lights on the island. In order to test this functionality, we set up a User Interaction Page which hosts a number of controls to display different light patterns. Utilizing the TCP communication piece, input from the user will be transmitted back to the laptop on the island. To verify if this test works properly, we made sure the microcontroller will output the user’s light display from the User Interaction Page.

**Solar Panel Maintenance Utility Overrides:** In order to manage the battery consumption of our system, we tested the overrides of the Solar Panel Maintenance Utility against the power consumption scripts. The Solar Panel Maintenance Utility will send a message to the laptop on the island to initiate a script. This script ensures the Light- O-Rama runs on the suggested power.

Hardware Testing and Troubleshooting

During the testing and troubleshooting phase of our project our team experienced some problems. Most of the issues occurred once we integrated all of the parts of the system and began testing. One of the most significant issues we ran into was with the first inverter we selected for this project. The inverter had a manufacture input voltage setting that was set too high, meaning the inverter would shut off once the batteries reached a certain level (around 12.3 V).

Furthermore, once the inverter shut off and the voltage level was brought back within the optimal range the inverter would not turn back on automatically. Also, the charge controller’s charging algorithm was initially set too low, meaning the batteries were not being charged at the correct level. Working with the battery manufacturer we gathered the appropriate charging algorithm levels, enabling our system to charge correctly.

Another issue that our team ran into was the ventilation of the battery array container. At first we did not plan for any ventilation, but during our first set of stress tests we noticed an acidic gaseous smell, which quickly led to a ventilation design. During the printed circuit board design of our interaction controls board we ran into grounding issues because we did not solider to connect both grounding nodes on opposite sides of our board.

Budget Analysis

Our client SDG&E has given us $3,500 to spend for this project. Due to numerous design adjustments and the change of the scope of this project, we were forced to spend an additional amount and beyond the $3,500 that was provided to us. This change included the addition of materials and labor for the construction of the Mega Tree, additional LED lighting to support the Mega Tree, and three gift boxes on the island. Including all of these additions, we spent in total of $5,177.31, or $1,677.31 over the budget.

Looking at our expenses, our biggest expense was the purchase of LED lighting at nearly 38% of our total budget. This includes rope lights that lit up the star topper, string lights that lit up the mega tree and work lights that supported the user interaction display. Our next biggest expense was the DMX controller which is approximately 10% of our total budget. Tied at 9% we have the mega tree rough materials, as well as our deep cycled battery array. Another 8% of the budget was used to purchase a laptop, and the remaining 26% of the budget was used for items such as, hardware supplies, charge controller, storage boxes, software, cables & connectors, FM transmitter, PVC piping, and other small ticket items. For more detail of the budget and specific items purchased, you can find the pie chart as well as the material list very helpful.

In addition to our expenses, we received many supplies and services for free or at a discounted rate. This included, two free solar panels from Sullivan Solar, free material and labor for the tree topper as well as the tree base, free Mi-Fi and tree trimming by Santee Lakes, and discounted metal piping for our tree. In conclusion to our budget, thank you to our Electrical and Computer Engineering department, we were able to cover this additional cost, and complete this project with its high expectations.

Conclusions and Recommendations

This project was challenging but rewarding as well. There were many unexpected challenges faced and even some of the obstacles that we anticipated were more daunting than we originally anticipated. As a group we were able to overcome all of the problems that we were confronted with and successfully light up the island on the evening of the deadline. Although the minimum requirement for this project was met, there are adjustments that could be made in order that future endeavors of this type have a smoother, more organized design process which would leave enough time to ensure that the project not only meets, but exceeds all requirements and expectations.

Carefully researching and anticipating any possible permitting restraints prior to the design phase of a project like this, is an absolute necessity. Because we did not account for this step in our planning process at all, we changed our structural design multiple times resulting in weeks of wasted time. The time spent dealing with city council and designing structural components of our project that never came into existence could have been spent in other areas of our design, testing, and verification.

Another area in which our design could have been more efficient was the setup of our battery array. More research into the benefits of an array in series as opposed to our parallel design would have been useful and would likely have convinced us to use the higher voltage, more efficient series array setup.

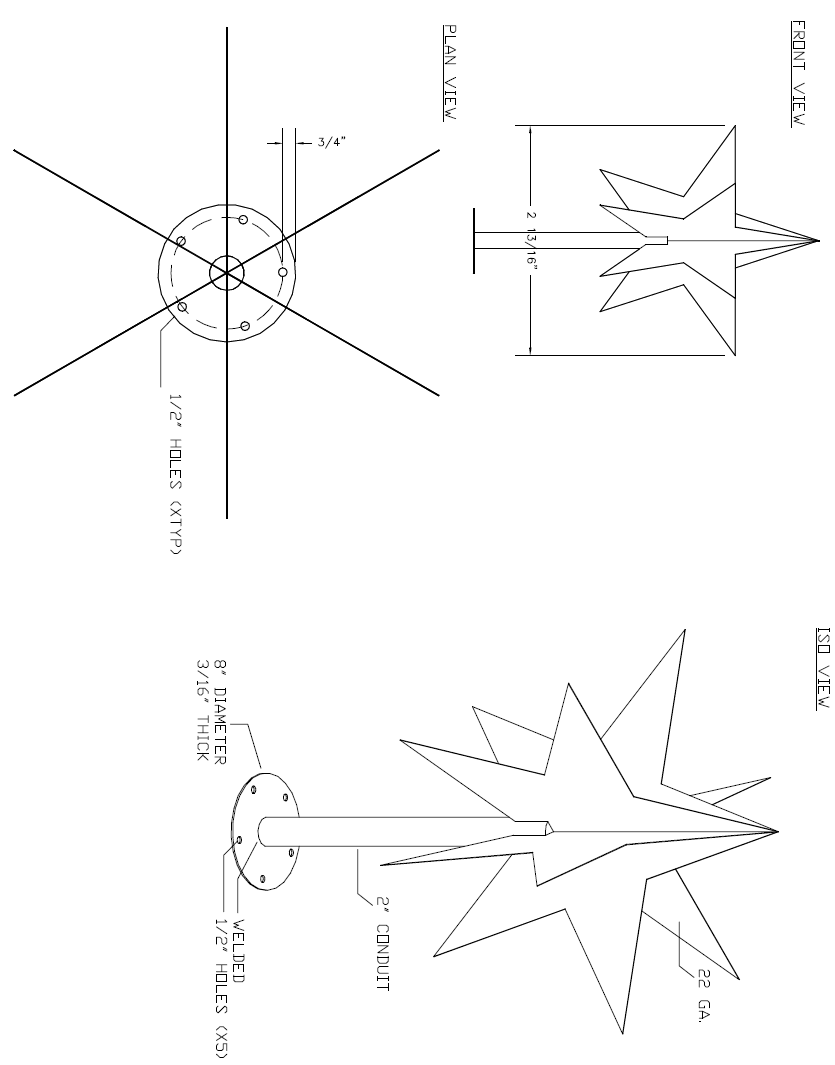
Although there were short comings in this project, there were successes as well. Our overall success, however, is determined by the subjective views of both our sponsors and the professor grading this report. Was it enough that at the highly anticipated moment of our deadline the lights came on but other, less visible, aspects of our design were not running? If the overwhelmingly positive and vocal response from audience members outweighed the 24 hour exceeding of our deadline to implement the remaining portions of our design, then I think this project could be considered a success.

I believe that all of the members of this project have taken something very positive from this experience away. We’ve all gained some real world engineering experience and we completed a challenging, and at times stressful, project working with a group who at the beginning of this project were, for the most part, complete strangers.

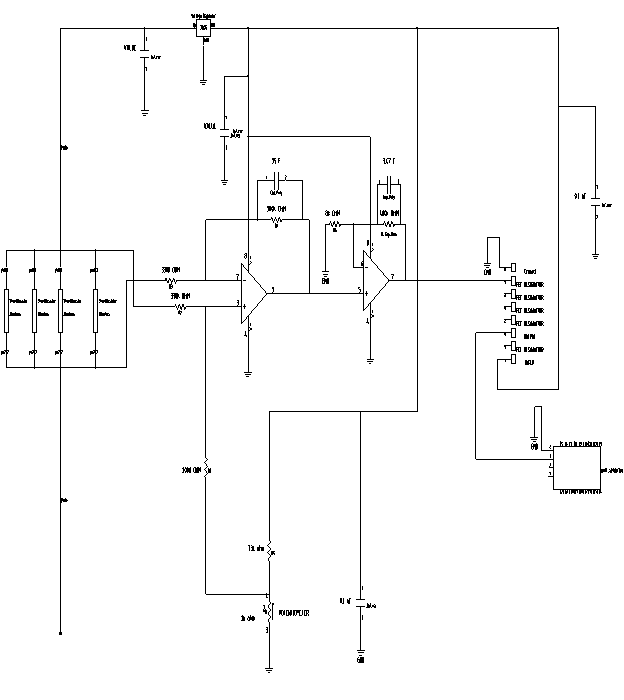
Appendices



## Mega Tree Design



## Star Topper Design



## Coulomb Counter Design

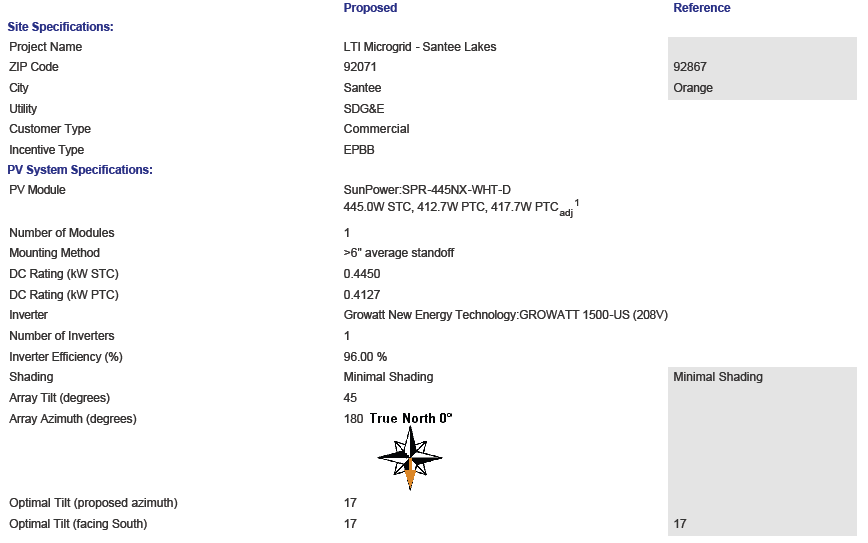
## Bill of Materials

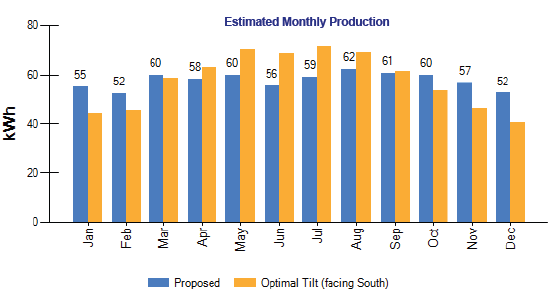
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **PO #** | **Item Description** |  |  | **Shipping** | **Quantity** | **Cost Per Unit** | **Total** |
| **1** | Pro Series LOR1602W 16 Channel Starter Package. Started | | |  | 1 | $ 387.80 | $ 387.80 |
|  | package includes: LOR1602W, 25' Cat5 Cable, S3 License, | | |  |  |  |  |
|  | USB485 adapter, USB Cable with filters. | | |  |  |  |  |
| **2** | MorningStar TS-60 | MorningStar 60A TriStar Charge Controller | |  | 1 | $ 172.50 | $ 172.50 |
| **3** | Kirkland - 115AH 12V Deep Cycle Battery | | |  | 4 | $ 85.00 | $ 340.00 |
| **4** | HLLY 1.5W TX-01S Power adjustable FM Transmitter | | |  | 1 | $ 188.00 | $ 188.00 |
| **5** | Power Bright 12 Volt DC to AC 1100 Watt Power Inverter | | |  | 1 | $ 80.99 | $ 80.99 |
| **6** | TOSHIBA Satellite C855-S5231 Notebook Intel Pentium | | |  | 1 | $ 329.99 | $ 329.99 |
|  | B970(2.3GHz) 15.6" 4GB Memory DDR3 1333 320GB | | |  |  |  | - |
|  | HDD 5400rpm DVD Super Multi Intel HD Graphics | | |  |  |  | - |
| **7** | Power Bright 12 Volt DC to AC 1100 Watt Power Inverter | | | $ 10.68 | 1 | $ 85.56 | $ 85.56 |
| **8** | High- voltage high- current Darlington transister arrays | | |  | 10 | $ 0.50 | $ 5.00 |
| **9** | GE C-9 LED Lights |  |  |  | 80 | $ 16.49 | $ 1,319.20 |
|  | Deep Cycled Batteries | |  |  | 4 | $ 83.49 | $ 333.96 |
|  | Core Charge Per Battery | |  |  | 4 | $ 9.00 | $ 36.00 |
| **10** | Rubbermaid 31 gallon Roughneck Dark Indigo Metallic storage box | | |  | 1 | $ 25.25 | $ 25.25 |
| **11** | Gordon 27 Led Portable LED Worklight/Flashlight | | |  | 30 | $ 2.49 | $ 74.70 |
|  | 73 Piece Eye Bolt Kit | |  |  | 2 | $ 5.97 | $ 11.94 |
| **12** | Breakout Board for FT232RL USB to Serial | | |  | 1 | $ 14.95 | $ 14.95 |
| **14** | 20' X 2 1/2" ID Aluminum Schedule 40 Pipe (2.875 OD X .203 wall) | | | $ 13.05 | 1 | $ 169.60 | $ 169.60 |
|  | 20' X 2" ID Aluminum Schedule 40 Pipe (2.375 OD X .154 wall) | | |  | 1 | $ 106.60 | $ 106.60 |
| **15** | Step-Down Voltage Regulator D24V6AHV | | |  | 1 | $ 9.95 | $ 9.95 |
| **17** | 4 Wire, UL, 26AWG, Stranded, Silver - 1000ft Phone Cable | | | $ 13.25 | 1 | $ 46.80 | $ 46.80 |
|  | RJ11 6P4C Plug Flat Stranded 50Pcs/Bag RJ11 Connectors | | |  | 2 | $ 1.47 | $ 2.94 |
|  | 50ft 14AWG Power Extension Cord Cable for indoor and outdoor | | |  | 1 | $ 23.87 | $ 23.87 |
|  | 10ft 16AWG Power Extension Cord Cable | | |  | 15 | $ 3.67 | $ 55.05 |
| **18** | Software Upgrade : Basic to Advanced (+90.00) | | |  | 1 | $ 90.00 | $ 90.00 |
| **19** | 1/2 in. x 10 ft. PVC Sch. 40 Plain-End Pipe | | |  | 17 | $ 1.68 | $ 28.56 |
|  | 1/2 in. PVC Slip x Slip x Slip Tee (10-Pack) | | |  | 2 | $ 1.98 | $ 3.96 |
| **20** | 1/2" 3-way Elbow PVC Fitting Connector | | | $ 25.46 | 16 | $ 1.40 | $ 22.40 |
|  | Set of 12 Large 10" Galvanized Steel Tent Peg | | |  | 7 | $ 11.98 | $ 83.86 |
|  | Stake Nail with T-Top - Heavy-Duty 8mm Diameter | | |  |  |  | - |
|  | Joy Fish Monofilament Fishing Line 100 lb test,1.1 mm, 430 yards | | |  | 3 | $ 12.50 | $ 37.50 |
| **21** | Stanley Pro Mobile Tool Chest (12 gallon) | | |  | 1 | $ 49.97 | $ 49.97 |
|  | Dimensions: 24 and (3/16)" x 14 and (3/4)" X 16 and (1/2)" | | |  |  |  | - |
| **22** | 0.100" (2.54 mm) Female Header: 2x5-Pin, Straight | | |  | 20 | $ 0.62 | $ 12.40 |
|  | 0.100" (2.54 mm) Breakaway Male Header: 2x40-Pin, Straight | | |  | 4 | $ 1.95 | $ 7.80 |
|  | Screwless Terminal Block: 6-Pin, 0.1" Pitch, Side Entry (2-Pack) | | |  | 4 | $ 2.45 | $ 9.80 |
| **23** | DCDC-USB, Intelligent DC-DC converter with USB interface | | |  | 1 | $ 58.49 | $ 58.49 |
|  | Convert any Voltage (6-34VDC) into ANY Voltage (5-24VDC) | | |  |  |  | - |
| **24** | Logic Root PCB |  |  |  | 2 | $ 20.00 | $ 40.00 |
| **26** | DCDC USB POWER BOX | |  |  | 1 | $ 6.95 | $ 6.95 |
| **27** | LASCO 1/2-in dia. 90-Degree PVD Sch 40 Side Outlet Elbow | | |  | 8 | $ 1.18 | $ 9.44 |
| **28** | Joy Fish Monofilament Fishing Line 100 lb test,1.1mm,430yds | | | $ 9.09 | 2 | $ 12.50 | $ 25.00 |
| **29** | 10 ft. x 12 ft. Heavy-Duty Tarp | | |  | 1 | $ 24.98 | $ 24.98 |
| **30** | Hampton Bay 12-Piece Rope Light Mounting Kit | | |  | 1 | $ 3.96 | $ 3.96 |
|  | Hampton Bay 48 ft. Incandescent Clear Rope Light Kit | | |  | 1 | $ 36.48 | $ 36.48 |
|  | Hampton Bay 108-Light 13-1/2 ft. Clear LED Rope Light | | |  | 1 | $ 32.97 | $ 32.97 |
| **31** | Gordon 27 Led Portable LED Worklight/Flashlight | | |  | 20 | $ 2.49 | $ 49.80 |
| **32** | PVC pipes and supporting supplies | | |  | 1 | $ 70.00 | $ 70.00 |
| **33** | Miscellaneous items | |  |  | 1 | $ 60.00 | $ 60.00 |
| **34** | LED rope lights/ decorative items for the gift boxes | | |  | 1 | $ 110.00 | $ 110.00 |

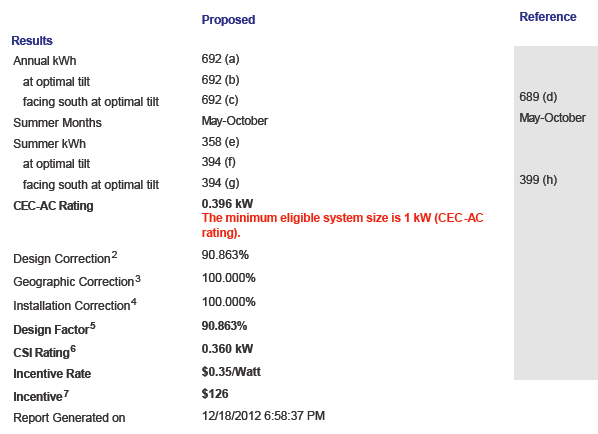
|  |  |
| --- | --- |
| **Sub-Total** | **$ 4,694.97** |
| **Sales Tax 8.75%** | **$ 410.81** |
| **Shipping** | **$ 71.53** |
| **Total** | **$ 5,177.31** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| LTI MICROGRID Load Calculations | | | | | | | |
|  |  |  |  |  |  |  |  |
| Equipment | Qty | Watts | Hours | kWH\*\*\* | System Voltage | Amps\*\*\* | Amp Hours\*\*\* |
| Light O Rama - 1602\* | 1 | 8 | 4 | 0.032 | 110 | 0.0727 | 0.291 |
| Laptops\*\* | 1 | 6 | 24 | 0.144 | 110 | 0.0545 | 1.309 |
| Energy Star C-9 LED Lights\* | 81 | 3.2 | 4 | 1.0368 | 110 | 2.3564 | 9.425 |
| FM Transmitter\*\* | 1 | 4 | 24 | 0.096 | 110 | 0.0364 | 0.873 |
| LED Mini Work Light\* | 48 | 0.625 | 4 | 0.12 | 5 | 6.0000 | 24.000 |
| LED String Lights\* | 1 | 23 | 4 | 0.092 | 120 | 0.1917 | 0.767 |
| **Total** | | 44.825 |  | 1.5208 |  | 8.7117 | 36.665 |
| Maximum Daily kWH Used\*\*\* | | | | 1.5208 |  |  |  |
| Maximum Monthly kWH Used\*\*\* | | | | 45.624 | **< 52** | **GOOD** |  |
| \*Calculated for a 4 hour usage. | |  |  |  |  |  |  |
| \*\*Calculated for 24 hour usage. | |  |  |  |  |  |  |
| \*\*\*Calculation not including flashing lights | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| LTI MICROGRID Consumption Calculations | | | | | | | |
|
| Solar Panel | Qty | Voc (V) | Vpm (V) | Isc (A) | Ipm (A) | Efficiency (%) | PTC (W) |
| Sharp NU-235F1 | 2 | 37 | 30 | 8.6 | 7.84 | 14.4 | 211.7 |
| Total |  | 37 | 30 | 17.2 | 290.08 | 14.4 | 423.4 |
| SunPower:SPR-445NX | 1 | 90.5 | 76.7 | 6.21 | 5.8 | 14.1 | 412.7 |
|  |  |  |  |  |  |  |  |
| |  | | --- | |  | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

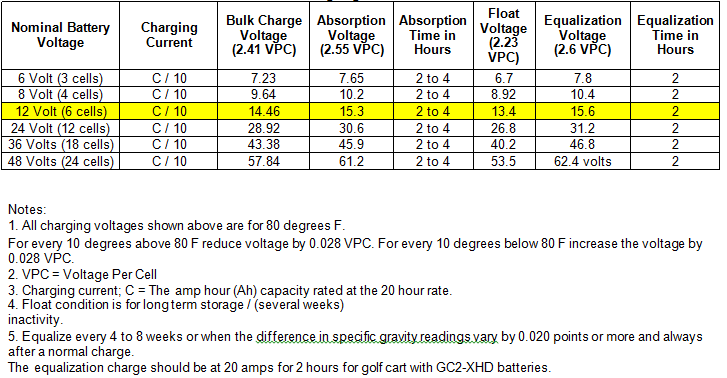
## Incentive Calculator – CSI Standard PV







## GC2-XHD Charging Recommendations



## GC2-XHD Voltage to State of Charge Table

