ASEE'S VIRTUAL CONFERENCE
At Home with Engineering Education
JUNE 22 - 26, 2020 #ASEEVC
Paper ID #31473

Prototype Automated Solar Tracking with Power Generation System

Dr. Saeed Sean Monemi, California State Polytechnic University Pomona

Dr. Sean Monemi is a Professor of Electrical & Computer Engineering and Director of Smart Grid Laboratory at Cal Poly Pomona. He has many years of industrial experience and practiced engineering profession as an electrical engineer for Litton Industries. His professional experience includes working as a senior engineer and supervisor in one of the largest power company, Tennessee Valley Authority, in the areas of Substation Deign and Metering.

Prototype Automated Solar Tracking with Power Generation System

Abstract — The purpose of this project was to design and build a Solar Tracking System from an electrical and mechanical perspective. The tracking system is equipped with automated battery charging circuit and switching capabilities for multiple batteries. The system can run a fully autonomous mode using photoresistor to track the sun ensuring optimal coverage. The popular ESP8266 Wi-Fi Module is interfaced with an Arduino Mega using its communications protocols for wireless control and monitoring of the system.

I. Introduction

In our everyday lives we constantly hear the chatter government and activists have in which they criticize the usage of fossil fuels because it is destroying the environment. Despite this, as of 2015, according to the Institute for Energy Research, solar energy constituted to only 5% of the total renewable energy consumptions, which in turn constitutes 13.4% of the energy domestically produced. However, according to the US State Department of Energy since 2008, US solar installations have grown seventeen-fold. Therefore, there seems to be a trend in solar energy production. Solar energy development seems to be walking in a direction where it could potentially become a fundamental source of energy in the US.[8]

II. Design And Implementation

A. Solar Tracking

The design of the solar panel structure went through various phases. Each change comes with what materials that could be gathered. The original plan discussion first started with a tripod or a quad-pod design. Either design quickly became arduous in terms of balancing. With the legs having to be made the same and attached at an angle, construction would be more difficult. This is when a square base became the ideal design. The structure itself will be well balanced and can sustain a substantial amount of weight.

For the solar panel to have the most movement, deciding on having one pole holding up the panel from the base was the best idea. Having multiple support poles would obscure the panel's movement when it rotates to locate the sun. Also, having one pole will be more than enough to hold up the weight of the panel.

With the tripod structure out of the question, the square base design is what was decided on planning on the axis movement. To achieve our goal we had to design a mechanical system with the capability of rotating. Two main ideas were narrowed down about the panel's movement: one with base rotation and the other with panel rotation, which subsequently became the final design. Base rotation required more gears and parts to accomplish a full 360° spin and would be allowed to face all parts of the visible sky above. The minimum materials required for this are chains, bearings, gears, and a motor. Evidently, the solar panel only needed to face the southern side of the hemisphere. In Pomona, CA, the angle of the sun reaches the highest during the summer solstice at 80° and lowest during the winter solstice at 32° above the horizon. Given this information, limiting the panel's 360° movement would not affect how the solar cells' function of gathering the most direct sunlight. With the panel only rotation, the base itself is stationary and must be facing in the general direction of the South Pole. [7]

A cross support pole would swivel from the pole to handle one rotation. The other rotation will be done by using hinges attached to the panel. These rotations will be further explained in the Movement section. To handle the rotations necessary to move the panel in the proper direction, using actuators is the best decision. It was a simple idea that did not need any other materials. They are to push and pull the panel bi-directionally. Placement of the actuators at both ends was a crucial decision. One actuator is placed on the support pole and the cross support. The other actuator is placed onto the cross support and the back of the panel. The solar panel will be attached to the cross support via hinges. Doing so, the panel will be able to look up and down.

The control system is designed to stop the panel once the sun was perpendicular to the panel. However, with the rotating pole it would be too hard to achieve this. Having a freely rotating pole would mean that we would have to deal with overshoot and damping of an oscillating pole. In order to control the oscillations it would require to run the motor constantly, which is not energy efficient. The software is designed with a stopping function so no excessive will be used.

The System Overview is a layout of the Solar Panel's structure (see Figure 1). The tracking aspect of the panel is handled by a separate Arduino Mega that interfaces the light sensors with the actuators. A control signal is sent from the other Arduino that handles the charging and switching aspect of the tracker. Power is supplied to both microcontrollers and the H-bridge after the load switching circuit. The two 12V lead-acid batteries are wired with the charging and switching circuits. They are wired to an inverter as well which supplies power to the Smart Grid. As the solar cells are positioned to receive maximum exposure to sunlight, the cells charge the batteries and is converted from direct current to alternating current via the inverter.

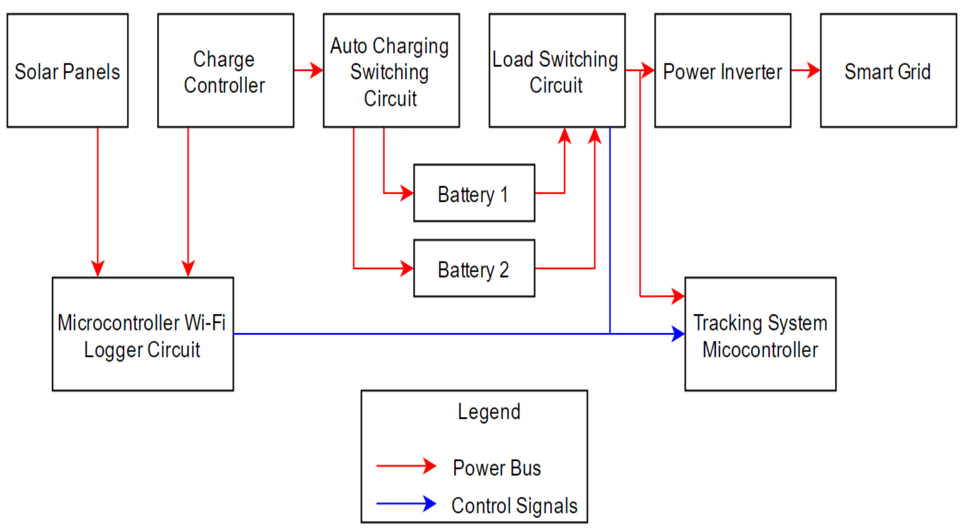

Figure 1. System Overview.

1) Structure

The final structure is made out of ¾" aluminum square tubing that was sourced from a scrap yard (see Figure 2). This material was cut and assembled using one of the team member's metal shop. All of the aluminum tubes were secured together with various hardware parts such as nuts, bolts and washers for ease of removal. The actual panel base is made of plywood sheets with an aluminum border and support beams on the back for support to prevent flex. Once the plywood sheet was painted, the solar cells were glued on in series and wires pulled down to the base. Aluminum was the best choice for the structure due to its lightweight but tensile strength.

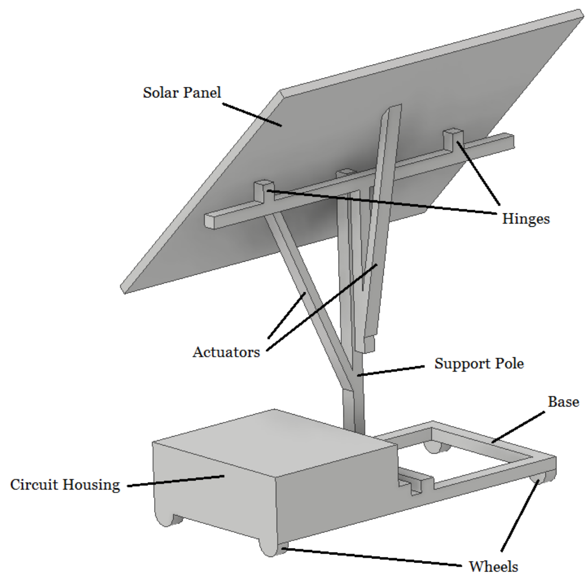

Figure 2. Final Design of the structure.

Once the structure was completed, cosmetics improvements were made such as adding wheels for ease of transport, and spray paint for uniform color. All wires and cables were also hidden as much as possible to provide a clean look to the product. All wires were also soldered and shrink-wrapped to prevent shorting.

Two actuators were used for the rotation of the panel (two axis) due to the simplicity of the push-pull nature of the actuators. Upon looking at various other solar panel tracker in industry as well as hobby projects, many of them used actuators as the method of movement. Originally, a chain and gear was considered with a mechanical stop, but this method of rotation proved to be too many moving and exposed parts. While this would have allowed us to specifically customize how fast the structure rotated, we opted to purchase two 12V linear actuators instead.

2) The Panel

The entire panel is built around a 42" x 38" aluminum frame that was scrapped from an already made solar panel from before. Four pieces of ⅛" plywood are measured and cut to fit within the frame. The solar cells used are Sunpower Maxeon Mono Solar Cells. They are 3.55W and measure 5"x5". Thirty solar cells are placed in series in a 6x5 pattern (see Figure 3).

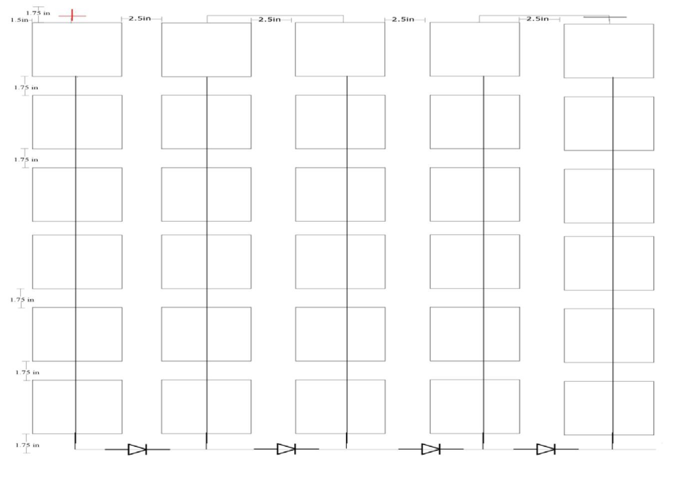

Figure 3. Each solar cell is separated by 2.5” in each column and 1.75” in between each row.

3) Panel Movement

There are three directions and their axis rotations: X, Y, and Z are roll, pitch, and yaw, respectively (see Figure 4). Only two of the three rotations are needed for following the sun through the visible sky. In this case, the panel’s movement only deals with the pitch and roll rotations. Having the yaw rotation would be excessive and not necessary.

Each of the two actuators are able to push the panel in their respective roll and pitch rotation. Pitch deals with the horizontal rotation when panel needs to find the exact height of the sun. In replacement of yaw, roll with the help of pitch will locate the sun if it is either to the left or right of the panel.

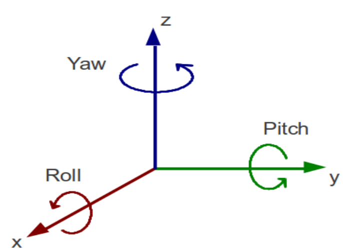

Figure 4: These angles determine the orientation and therefore the direction the panel will take.

The actuators do have their limits. They can only extend up to 305mm (~1 ft.). When the panel is facing forward (South), the lowest is ~22° above the horizon. Going the opposite way, the panel can reach straight above (zenith point). This helps the panel to face direct left and direct right. Facing left (East), the lowest point the actuator can retract is ~23° above the horizon. Facing right (West), the lowest point the actuator can extend is ~44° above the horizon.

4) Sensors

The light sensors that are used called photoresistors (see Figure 3). A photoresistor is a light controlled variable resistor. The sensors are GL5516 photoresistors; they have an approximate 12% variation. What that means is that a photo resistor receiving the same amount of light might be 1 kΩ and other resistor might be 1.2 kΩ. The resistance of the photoresistor decreases as the intensity of the sunlight onto the photoresistor increases. Four of them are used with wall barriers in between them to read four separate values (see Figure 4). [6]

Without the walls, the photoresistors would all read the same and would have no sense in where the sun is located. The walls allow the sun to cast a shadow on top the resistors if the sun is not perpendicular to the solar panel. The actuators will move the panel until all four photoresistors read about the same values and then stop, therefore the panel will then be perpendicular to the sun to gather the most direct sunlight. To further explain, the photoresistors will never read the exact same values as each other unless they were made exactly the same and placed onto the panels exactly alike, which is a very tough feat to accomplish.

Each sensor is attached onto an Arduino Mega in series (see Figure 5). By doing this, the Arduino Mega will be able to read each sensor through voltage division. This will be further explained in the Coding.

Figure 5. Small photoresistors that have a resistance of 10K ohms.

5) Coding

The microcontroller selected was Arduino Omega. The reason was that it was economically convenient, it is easy to program, and to interface with the many inputs and outputs that we were expecting to use.

The algorithm is succinct. The panel moves in the direction of the sensor receiving the most light. The algorithm is not greatly impacted by the 12% error of the sensors. The reason is that it is based on what sensors are receiving shade. Shade produces more than a 12% change. This means that the shaded photoresistors are always going to have a greater resistance than the non-shaded resistor. The movement stops until the microcontroller reads a difference of the sensors smaller than 20%. What this algorithm produces is that the panel will stop moving until the four sensors are close to being perpendicular to the sun or light source; or when none of them have a significant amount of shade.

In order to compare the amount of light received by the four sensors, a circuit involving the four sensors was created (see Figure 6). The circuit contained four sensors connected in series. At one end of the series was the Arduino's ground and at the other end was the Arduino's 5 volts power supply. The Arduino code measured and calculated the voltage dissipated by every sensor. In such a circuit, voltage and resistance are directly related. Higher voltage means a directly proportional higher resistance. By doing these measurements, it was possible to compare the resistances of the four sensors, and thus to determine, which corner of the cross section was receiving more light.

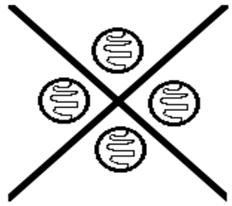

Figure 6. Top view of the photoresistors with the walls in between each of them

This image displays the circuit created with the sensors. The wires sticking out were connected to the Arduino inputs indicated by the labels. The Arduino measured the voltage at those locations. The voltage dissipated by Sensor1 is obtained by just measuring AnalogRead0. However to calculate the voltage dissipated by Sensor2, Sensor3 and Sensor4, it is necessary to make the Arduino perform subtraction (see Figure 7).. For example, to calculate the Voltage dissipated by Sensor 4 it is necessary to subtract AnalogRead2 from V_{dd} (which is 5V, or 1022 as measured by the Arduino).

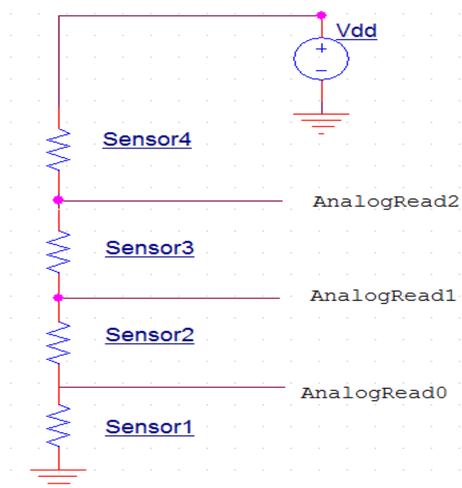

Figure 7. Circuit schematic of photoresistors connecting in series.

The Arduino's power and voltage limits made it impossible for the device to directly control the Actuators used to move the panel. Thus a L298N chip was also used to control the motion of the actuators. Such chip contains two H Bridges. An H Bridge is a device where transistors are arranged in a way to make it possible to control the direction of the motors, or to cut the power off the motors (i.e. stop the motors), by opening and closing MOSFET transistors. At the same time, the bridge may be powered by a high power source and is controlled by a low voltage, low current source. Thus, the Arduino's maximum 5V of its digital outputs were enough to open and close those transistors to control the direction and motion of the motors, while at the same time the energy driving the motors came from an outside power supply.

The L298N chip contains two H-Bridges (see Figure 8). The parts indicated by number 3, 6, 7, and 12 were not used. Part numbers 3, 7, and 12 are fuses. A motor's terminals can be connected to number 1 and 2. Another motor can also be connected to terminals 14 and 13. The power supply that will drive the motor is connected to part number 4, and ground is connected to part number 5. The power driving the motors was also used to power the Arduino. Thus, part number 4 and 5 were connected to the microcontroller inputs V_{in} and ground respectively. The chip inputs indicated by 8, 9, 10, and 11 are used to control the direction and motion of the motors. They are digital inputs where 0 V corresponds to a logical 0.5 V correspond to a logical 1. Thus, this chip is ideal for the Arduino.

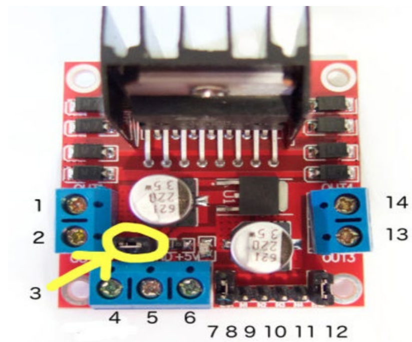

Figure 8. L298N chip with two H-Bridges.

Part numbers 8 and 9 control the motor connected to 1 and 2. Part numbers 10 and 11 control the motor connected to 14 and 13. A combination of a logical 10 moves the motor in one direction; a combination of 01 moves the motor in the opposite direction; and a combination of 00 cuts off the current to the motors (i.e. stalls the motors).

B. Charging

A charge controller, or charge regulator is a voltage and/or current regulator to keep batteries from overcharging. It regulates the voltage and current coming from the solar panels going to the battery. Ordinary charge controller will often work with high voltage panels if the maximum input voltage of the charge controller is not exceeded. However, a lot of power will be lost, from 20% to 60% of what your panel rating. The only way to get full power out of high voltage grid tie solar panels is to use a MPPT controller. A MPPT, or maximum power point tracker is an electronic DC to DC converter that optimizes the match between the solar array (PV panels), and the battery bank or utility grid. They convert a higher voltage DC output from solar panels (and a few wind generators) down to the lower voltage needed to charge batteries. [3]

For this project, the team originally wanted to use Linear Technology LT8490 as the MPPT charge controller. But because of the size of the chip was too small, and the chip has 52 pins, the team decided that the LT8490 was too hard for this project. Then, the team decided to use Texas Instrument bq24650. The bq24650 was the chip recommended by Texas Instruments because it is made especially for MPPT solar panel charge controller. The bq24650 provides input voltage regulation, which reduces charge current when input voltage falls below a programmed level. The bq24650 (see Figure 9) has 5V-28V input volar panel, and can be charged at any current level. [1]

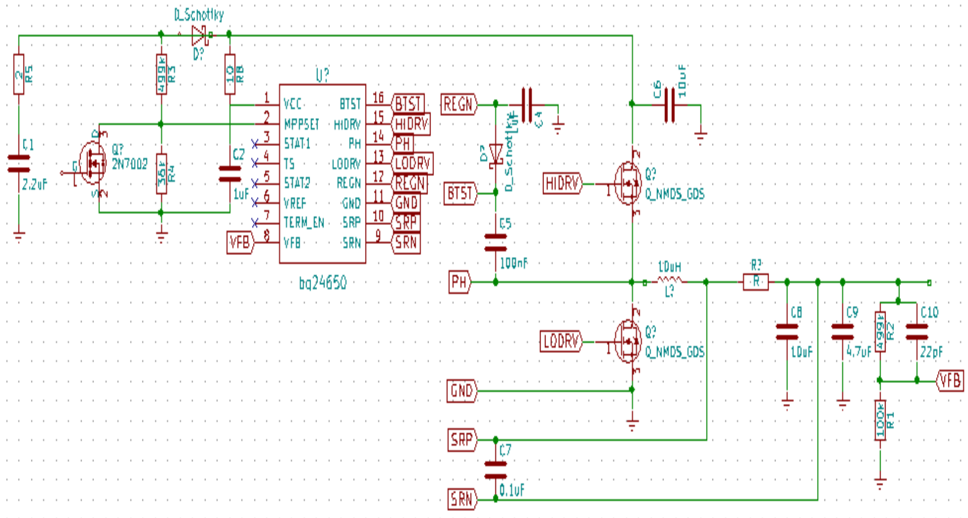

Figure 9. Schematic for charge controller using bq24650

C. Battery charging Switching

The solar panel produces power continuously. To avoid any loss of power because of a low load, it is optimal to store the power in batteries. Because we are using two batteries, we decided to have a battery switch, so that when the first battery is fully charged, it will automatically switch to charge the second one. It was decided to use the LTC4412 chip as a power pass controller. The LTC4412 controls an external P-channel MOSFET to create a near ideal diode function for power switchover or load sharing. This chip has the input voltage between 2.4-4.8 V, which is the range of our input voltage. [2]

Whichever battery has the lower voltage will receive the charging current until both battery voltages are equal, then both will be charged. When both are charged simultaneously, the higher capacity battery will get proportionally higher current from the charger. Figure 10 shows configuration used for this part of the project.

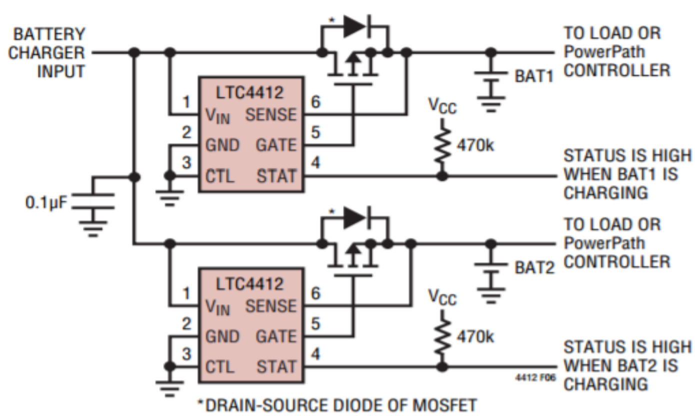

Figure 10. Schematic for battery switching

D. Battery Load Switching

Batteries are connected to smart grid and operated via a switching circuit. The circuit is operated through a microcontroller with the aid of a resistor voltage divider, monitors each battery and performing the switching. Back-to-back MOSFETs are used so that the drain-source diode will not power the load when the MOSFET is turned off. See Figure 11.

When the input from the microcontroller is low, Battery 1 supplies power to the load regardless of Battery 2 voltage. When CTL is switched high, Battery 2 will power the load whether or not it is higher or lower than Battery 1 voltage. Once Battery 2 is on, Battery 1 can be removed and Battery 1 will continue to power the grid. Only when Battery 1 voltage is higher than Battery 2 voltage will taking CTL low switch back to Battery 1, otherwise Battery 2 stays connected. At a minimum, COUT capacitance must be sized to hold up Voltage for grid until the transition between the sets of MOSFETs is complete. Sufficient capacitance on the load and low or no capacitance on input voltage were used to help ensure this. BJTS were also used to drive the power MOSFETs.

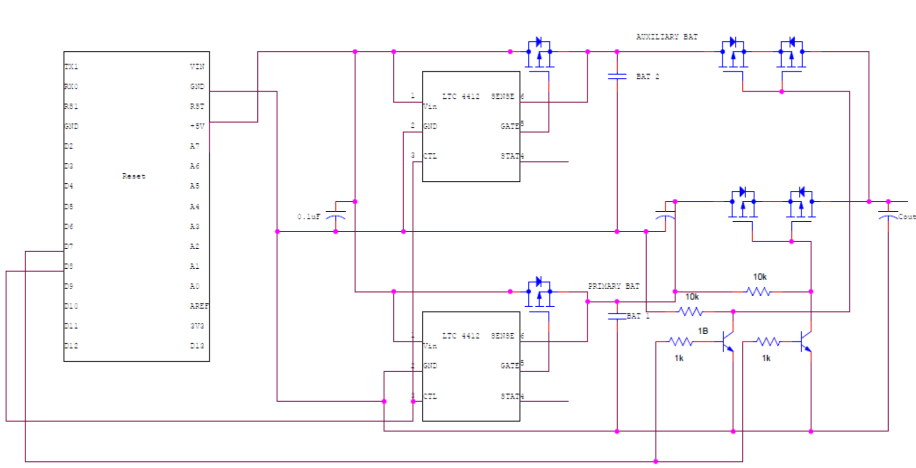

Figure 11. Complete schematic of switching circuit.

E. Inverter

It has been observed that the solar panel utilizes energy that it transduces from sunlight into two batteries. The batteries are charged simultaneously and a switching circuit is used to charge the second battery when the first battery becomes full. Eventually both batteries become fully charged and excess energy is to be sent to either the residential load of the smart grid or back to the utility company. Due to AC energy being advantageous to DC energy when transmitting power over transmission lines it is necessary to use an inverter to convert the DC energy that is being generated from the solar power into AC energy which will be transmitted to the residential load of the smart grid as well as the utility company. Since the team will need an inverter to transfer the DC voltage from the panel to the prototype grid, the team would like to understand more on how the inverter works. After some research, it was decided that the best inverter circuit to utilize for the solar panel is a multilevel inverter through using independent dc sources. See Figure 12.

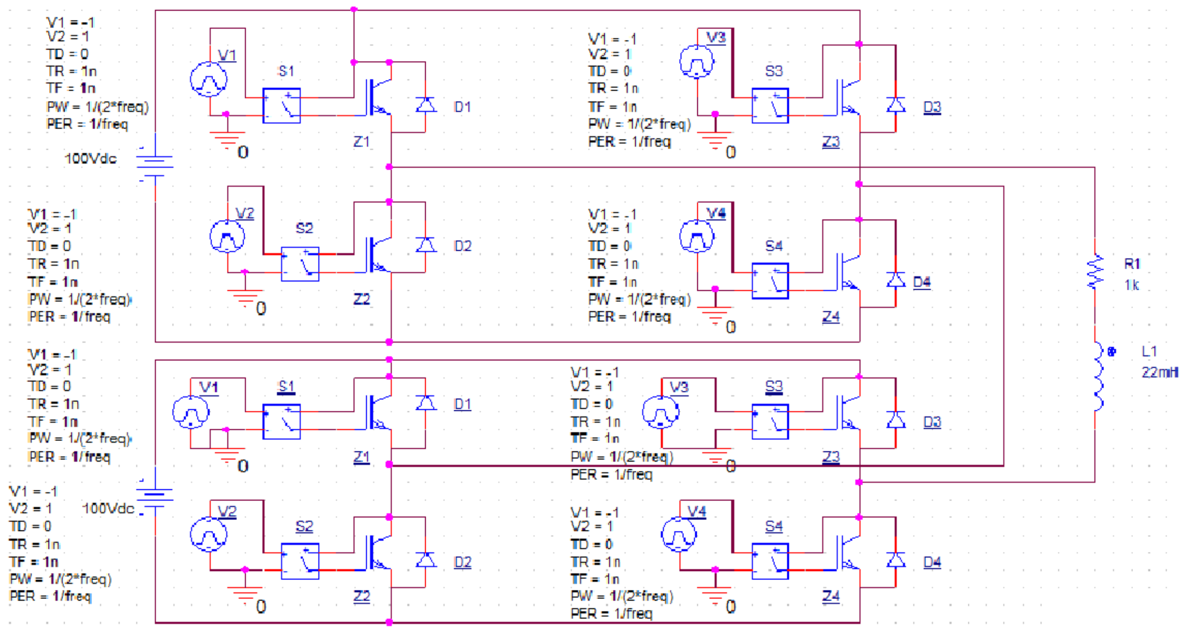

Figure 12. Schematic of prototype inverter

F. *Wi-Fi Implementation*

We wanted to make our system be connected to the internet and move away from a physical control. The ESP8266 is a low-cost Wi-Fi capable chip with a microcontroller unit. This chip was used due to its low cost and versatility. The code written and uploaded to the ESP8266 via the Arduino IDE gave the ability to monitor the outputs of the solar panel, and the HTML webpage gave the ability to wirelessly adjust the angles of inclination of the panel through the actuators. Along with the ESP8266 the INA219 breakout board was used for power-monitoring (high side voltage, DC current) over I2C.

III. Testing and Performance of The System

A. Solar Tracking

1) Structure

With the two actuators first attached to the structure, they are extended to their maximum and retracted to their minimum to see if any obstructions were in the way. The T-frame that is held on with a single bolt onto the main support pole has to be securely and tightly attached. Any looseness will result in the structure hitting itself. This was then repeated for when the panel is attached. The main bolt that swivels the panel with the support pole is durable enough to withstand the entire weight of the panel. Natural occurrences, such as wind, are also taken into account.

The actuator that controls the roll rotation has its limitations. When it is retracted, the angel above the eastern horizon is about 23º. This is for when the sun rises. When the actuator is extended and the panel is facing the opposite side, the angel above the western horizon is 44º.

The only problem with this is tracking the sun during sunsets. Most of the energy will be during the peak hours in the day. Early sunrise and late sunsets will not output as much energy from the solar cells compared to when the sun is higher in the sky.

2) Coding

The code to control the motors went through various trial and errors. Driving power to the actuators was the easy part. Reading the sensors was a challenge. Since each photoresistor was reading differently with the same amount of light, the error could not be at the minimum. Each actuator is connected to a pair of sensors. To stop the actuator, both sensors are to be read equally. With the sensors reading in intervals of 1 second, it is very difficult to have both sensors read the same values. The only solution is to have the difference below a certain threshold. If the threshold is too low, the actuators will never stop. If threshold was too high, the actuator would stop too early and the panel would not be perpendicular to the sun.

B. Charging

After the bq24650 was connected to all necessary components, the circuit was tested and it was charging the battery. The problem was that the battery was charging at a very small current. Instead of charging at a minimal 2 Amp, it was only charge at 18 mA. The speculation why the circuit didn't work properly was either because the bq24650 was burned or the MOSFET didn't work properly. It was difficult to reflow the QFN chip. Since the team didn't have enough time to order a new bq24650 to test the circuit, the team came up with a plan B. See Figure 13.

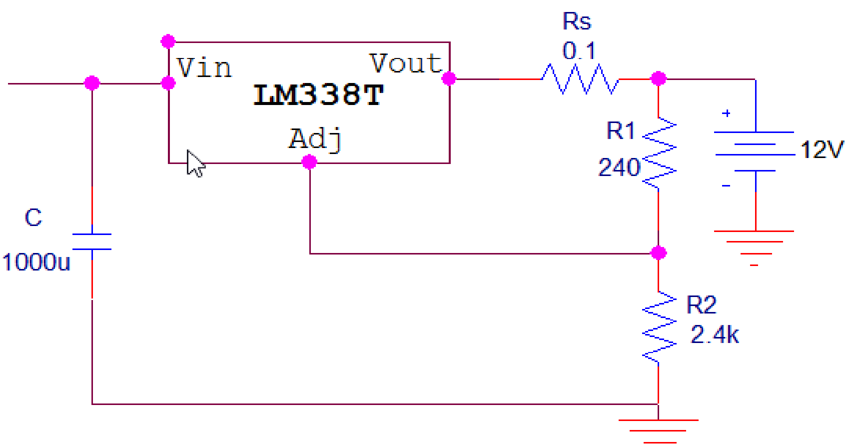

Figure 13. Final schematic for charge controller

The team decided to change the charge controller to a simpler circuit with the amount of time left. It was decided to use the LM338 with a maximum input-output voltage differential of 40 V and a 5 A output current (see Figure 14). The output of the LM338 was connected with a 0.1 Ω sense resistor to sets the output impedance of the charger, and then it's also connected to the battery. The use of sense resistor allows low charging rates with fully charged battery. The input of the LM338 was connected to a 1000uF to filter out input transients. After the LM338 was connected to all necessary components, the circuit was tested and it worked. The circuit was able to draw 3A under subnormal condition (when it was cloudy outside). [5]

Figure 14. Final design of charge controller

C. Switching

After putting the circuit together, unfortunately, when testing the battery load switching circuit, the team noticed we have a current greater than 10 Amp to the inverter. At such a current we needed to put heat sinks to each of the MOSFET. Although they are rated greater than 10 Amps and had adequate heats sinks the MOSFETs gave too much of a voltage drop to the inverter because it to prop the under voltage alarm. Therefore, our final design had to incorporate two relays. Each relay is rated for 10 Amps. It was decided to double up relays to split the current. Battery charging switching circuit was tested and it is working properly providing charge to the lowest voltage battery or both when equal voltage. See Figure 15.

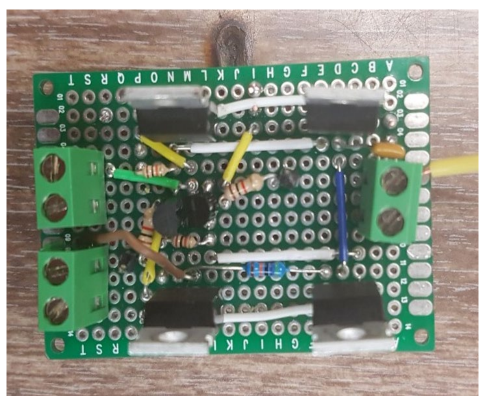

Figure 15. Prototype for battery switching

D. Inverter

The multilevel inverter utilizes two H-bridge circuits with an independent dc source being used for each h bridge. The circuit utilizes a total of 8 IGBTs each utilizes a feedback diode for each IGBT to ensure that the IGBT is able to output a current value. The circuit utilizes an RL load with a resistance value of 1K and an inductance value of 22m in order to produce the voltage output of the inverter in AC.

The switching scheme that was chosen to utilize the inverter is a PWM square wave that has an amplitude of 1V and a frequency of 60 Hz. We chose to utilize an Arduino microcontroller in order to implement the PWM for the circuit. The microcontroller chip that was utilized had 32 pins and pins D3 to D10 was what controlled the PWM for the 8 IGBTs. PIN D0 of the microcontroller defined the battery voltage, PIN D1 defined the heat sink, PIN D2 defined enable power which is high voltage power on switch, PIN D11 defined the fanOnInd which is the indicator if the fan is on or not, PIN D12 defined the batteryGoodInd which indicates if a proper battery is being utilized, and PIN D13 defined the HVON which indicated high voltage on blinks. The microcontroller was programmed for the PWM to be implemented simultaneously for each IGBT in periods of 8.6 ms where each IGBT will be turned on for only 8.6ms and then turned off and will repeat this process throughout the duration the inverter produces an output voltage in AC.

When testing was performed, the circuit came to be a success in terms of converting the dc input into a dc output. Oscilloscope was used to measure the output of the inverter circuit. It was the RL load of the inverter circuit that was measured in order to demonstrate what the output was. The response came out to be good but one tradeoff was that the output voltage was very small being in the range 500 to 750 mV. While the schematic that was chosen was successful in converting from DC to AC it was not successful in producing a large enough voltage to be able to be utilized effectively by the residential load of the solar city.

This design of the inverter is successful though in sending excess energy not stored in the batteries of the solar power to the power utility company. Because it is of utmost importance to be able to have the residential loads of the solar city to be powered by the excess energy of the solar panel, the team had to buy an inverter off the internet in order to accomplish that necessity. Even though the final design wasn't used by the team for the solar power, the team learned much about how inverters work is crucial and beneficial to being able to communicate with people observing this senior project detail and insight into the inverter aspect of the solar panel.

E. Wi-Fi Implementation

The design of our system makes this circuit the system ON since we control the battery switching. There for the Wi-Fi circuit need to always be power by either bat1 or bat1. As a result we need to put switch to turn this main board on. Afterwards since the ESP8266 only takes regulated power input we made use of L805CV to provide us with 5V. In order for the ESP8266 to send control signals to the Arduino MEGA we needed a level shifter. The ESP8266 uses only 3.3V logic while the Arduino uses 5V logic. This data collected was viewed within the Arduino IDE in Serial COM. With the Wi-Fi capabilities of the ESP8266, which was connect to the INA219 to take advantage of I2C, the data was sent to the MATLAB API- ThingSpeak. The data collected was viewed with respect to time as seen in Figure 1, where the DC current in mA for a test load was collected. Due to restrictions imposed by the API and budget limitations all the data points seen in Figure 1 are spaced 15 seconds apart. [4]

The programing of the ESP8266 was rather simple as there are multiple libraries for the Wi-Fi communication protocol side. We need to add html code that creates the GUI for manual control and switching. For this we used the Web-Server library to have a page created on the local network. Also we used the adafruit library for the INA219 since it was readily available. Since, we needed to have different current rating than the one provided from stock in conjunction to modifying the sense resistor on the PCB. To modify the program in INA219 chip ourselves we change the include INA219.h file. For the manual control of the solar panel, we used a truth for the control of each linear actuators. Using the ESP8266 to create the control signals to the MEGA for each of the possible combinations from the table 1 below.

B0	B1	B2	Action
0	1	1	STOP
0	1	0	Gray In
1	0	0	Gray Out
1	0	0	Black In

Table 1. Truth Table of the actuators' manual control.

IV. Learning and Outcome Assessment

A. *Wi-Fi Implementation*

Now that the system is fully implemented future teams can add more sensor easily by adding to the I2C bus one feature that is possible is more current sensors for accurate charge information to each of the batteries. If team decide to build another panel with Wi-Fi, our team would recommend looking to Wireless Sensor Networks, which are used in the power industry for monitor of individual plants. See Figure 16.

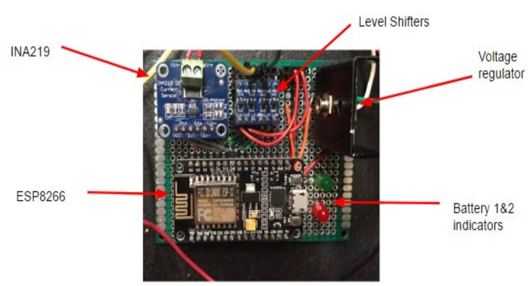

Figure 16. Final Circuit of Wi-Fi implementation solution

B. *Assessment and Automated Tracking*

Our system had some relevant mechanical limitations (see Figure 17). It was not able to travel the whole upper half-sphere (travel 180º in any direction from any horizon to the opposite horizon). This results in the user having to make a decision in how to set up the system, making the system not 100% automatic. An improvement to this project would be to expand the surface area that the tracking device can cover. A suggestion for this improvement would be to replace one of the actuators with a rotating base. However, this would result in a more complicated mechanical system, and possibly, a more complicated control system.

Future teams could also improve the precision of the tracking. The sensors we used (photoresistors) were significantly inaccurate. As a result, sometimes the tracking may have a difficult time finding the optimum position. Also, our system relies purely on information from the photoresistors. The microcontroller does not know the inclination of the panel. Thus, including additional sensors such as gyroscopes would allow more accuracy.

Finally, in this age of the smartphone, it would be a strong improvement to be able to control the movement of the system by using a mobile application.

Figure 17. Completed Photo of the solar panel and structure.

V. Educating students and broader impact

The team designed and built a prototype solar tracking system from an electrical and mechanical perspective in order to demonstrate the concept of solar tracking and power generation. The prototype has automated battery charging circuit and switching capabilities for multiple batteries. The system was able to run in fully autonomous mode using photo-resistor to track the sun ensuring optimal coverage. The popular ESP8266 Wi-Fi Module was interfaced with an Arduino Mega using its communications protocols for wireless control and monitoring of the system.

Global warming is a serious issue that society has to aggressively tackle. This team decided to provide a proposition on how to take some actions against global warming. The design provides a simple structure that can produce and store solar energy efficiently. In a more lucrative tone, the simplicity of our design also provides a mean to obtain profits by farming solar energy.

VI. Acknowledgment

We would like to thank Ganpat Patel for the sponsorship of this project. In addition, we extend our special thanks to our advisor Dr. Sean Monemi.

VII. References

[1] *Bq24650 Synchronous Switch-Mode Battery Charge Controller for Solar Power With Maximum Power Point Tracking*. N.p.: Texas Instruments, n.d. Pdf.

[2] Corporation, Linear Technology. LTC4412 - Low Loss PowerPath Controller in ThinSOT (n.d.): n. pag. Web.

[3] "Everything You Need to Know About the Basics of Solar Charge Controllers." Northern Arizona Wind & Sun, n.d. Web. 16 May 2017.

[4] Jayakumar, Magesh. *Nodemcu Dev Kit Using Arduino IDE: Get Started with ESP8266*. N.p.: n.p., n.d. 31 Dec. 2015. Web. 27 May 2017.

[5] *LM138 and LM338 5-Amp Adjustable Regulators*. N.p.: Texas Instuments, n.d. Pdf.

[6] "PhotoResistor Type 5516 5-10K." *YourDuino*. N.p., n.d. Web. 27 May 2017.

[7] "Solar Angle Calculator." *Solar Electricity Handbook*. N.p., n.d. Web. 27 May 2017.

[8] "Solar Energy in the United States." *Solar Energy in the United States | Department of Energy*. U.S. Department of Energy, n.d. Web. 27 May 2017.

[9] Hart, Daniel W. *Power Electronics*. New York: McGraw-Hill, 2011. Print.