# Real Time Clock Based Multilevel Sun Tracking Solar Panel System

# Implementation and Verification

**A Thesis Submitted to the**
**Department of Electrical & Electronic Engineering of BRAC University**

In partial fulfillment of the requirements for the
Bachelor of Science degree in Electrical & Electronic Engineering

**By**

Siam Bin Khurshid
Tanmoy Debnath
Syed Fakir Nawaz
Syed Nafiz Imtiaz

**August 2016**

# DECLARATION

We do hereby declare that the thesis titled —"Real time clocked based multilevel solar system" submitted to the Department of Electrical and Electronic Engineering of BRAC University in fulfillment of the Bachelor of Science in Electrical and Electronic Engineering, is our original work and was not submitted elsewhere for the award of any other degree or any other publication.

Date:

Supervisor

Dr. Md. Mosaddequr Rahman

- Siam Bin Khurshid
  Student ID: 12121110
- Tanmoy Debnath
  Student ID: 12121136
- Syed Fakir Nawaz
  Student ID: 12121157
- Syed Nafiz Imtiaz
  Student ID: 12321056

# ACKNOWLWDGEMENT

We would like to extend our vivid gratefulness to our thesis supervisor Dr. MD. Mosaddequr Rahman, Professor of BRAC University who has supported us with his kind assistance and dedicated involvement in every step throughout the accomplishment of this paper and our project work. We are immensely grateful for his help, professionalism, precious guidance and encouragement throughout the course of this work. Moreover, we are thankful to our parents and friends for encouraging us.

# ABSTRACT

Solar energy has originated to be a feasible source of renewable energy over the past two or three decades as this has been used to power up our industrial and domestic applications and gain desired national production. Bangladesh, like some other third world countries, has higher demand of energy than production met by the power sector, even after considering only 70% of the population who are hitherto under electricity coverage of national grid. An inevitable consequence of this shortage is load-shedding, and people of urban areas are major sufferer of it because of their dependency on electricity for quotidian activities. Although solar energy has the promise and potential to solve the energy crisis of Bangladesh, the limited space for setting enough solar photovoltaic (PV) panels to meet the demand of city dwellers emerges as a constraint in implementing solar energy system in densely populated urban areas. In the proposed system, a modified structure equipped with a sliding mechanism holds three solar panels. The panels are stacked one above another in a rack to minimize the floor area, and track the sun using a sun tracking system to maximize power generation. This work focuses on the implementation of the proposed system and verification of its claims. The implementation has been carried out by constructing a physical structure equipped with servo motors controlled by a microcontroller to rotate the panels on a horizontal axis. The Microcontroller determines the appropriate position of the panels for maximum power absorption for a particular time using a real time clock and a set of equations. The data of current and voltages of each panel are collected using a data logger device developed for this purpose. The collected data are then analyzed, which show a high level of conformity to the theoretically calculated values. The Microcontroller determines the appropriate position of the panels for maximum power absorption for a particular time using a real time clock and a set of equations. The panels are supported by a metal rod fixed at a tilted angle equal to the latitude among north-south direction facing south (23°). The panels rotate throughout the day from sunrise to sunset through servo motors controlled by microcontroller. It allows rotation of the three panels on a fixed horizontal axis to track the sun at hourly intervals and maximize power generation. The entire position of the three panels is changed by the actuator after midday when the sun moves from east to west. The projected system will give a worthwhile solution to electricity problems in urban areas where space is inadequate.

# Table of Contents

## List of Figures

# CHAPTER 1

# Introduction

## 1.1 Motivation

Solar energy is considered to play an instrumental role in the infrastructure of a country as a distributing source. Country like Bangladesh which has added several natural grids to the national power grids but still that is not enough. In 2015, despite the requirement of 10000 MW on an average, the installed capacity has reached to 12071 MW of which only 75% is available. The consequence of this shortage is nothing but load shedding. People of urban areas are major sufferers of it because of their dependency on electricity for daily activities. Solar energy is something that has the promising potentiality to make our country getting rid of electrical energy crisis. It is shown by theoretical calculations that the system harnesses 18% to 64 % more energy based on season than that of conventional fixed panel system at the same time occupying 33% less space [1]. The proposed system is easy to implement, but has a potential to be of immense benefit for the dwellers of large cities like Dhaka, where not only electricity is short in supply but also availability of rooftop space to install solar panels is also very limited. The position of Bangladesh is between 20.30 - 26.38 degrees north and 88.04 - 92.44 degrees east which indicates an ideal location for solar energy consumption. In our country 4 to 6.5 KWh per square meter and maximum amount of radiation is available in summer. Therefore Bangladesh is the suitable place for installation and testing our system efficiently.

## 1.2 Project Outline

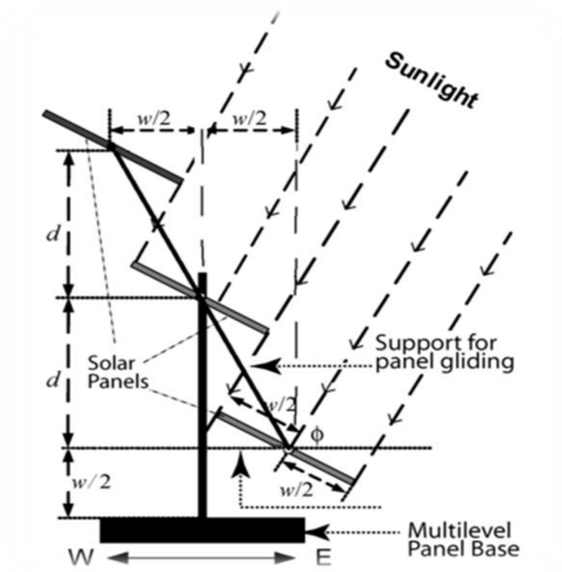



Figure 1.1: 2-D Physical structure of the rack of solar panels.

The system of this project is designed in such a way that automatically tracks sun from sunrise to sunset and converts it into solar energy. Each of the three panels is fitted with a servo motor that is controlled by the microcontroller in the Arduino. In our system, we have used both Arduino Mega and Arduino Uno microcontroller boards. Arduino Uno is based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. The Arduino Mega board is based on the ATmega1280. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. A simple side view of the proposed system is depicted in figure 1 where three identical panels are supported by a bar with a mechanism so that the panels can rotate on a horizontal axis. This bar along with the panels is mounted on another vertical bar fixed to a base in such a way that the former bar can rotate around a certain pivot. In order to maximize the average solar irradiance on the panels further throughout the year, the panels are tilted by an angle equal to the latitude along north-south direction facing south (not shown in Fig). Here we have used The Mini Maestro which is a highly versatile (and compact) servo controller and general-purpose I/O boards. Mini Maestro 12-Channel Servo Controller controls and provides power to the three servo motors. The channels can be organized as servo outputs making this servo controller well suited for high-precision animatronics, and built-in speed and acceleration control make it easy to achieve smooth, seamless movements without requiring the control source to constantly compute and stream intermediate position updates to the Mini Maestro. The panels will be reset to their original position at sunset for the following day along with the actuator.

The gliding bar faces east in the morning with an angle that the axis of top and bottom panels are away from each other in the horizontal plane by a distance equivalent to the width of a panel; three panels, thus, occupy the space of two in that plane. The gliding bar, thus, creates an angle of 67.5 degree. When the sun starts to move to west at noon, the panels need to be repositioned with the top panel moved to east and bottom panel to west, so that all the panels can experience maximum sun exposure. This is done by pushing lower part of the panel holding bar to west with the aid of an actuator by a certain angle so that the top panel takes place of the bottom panel in the horizontal plane and vice versa. The gliding process will allow the top and bottom panels to rotate throughout noon but the middle panel will stay vertically stationary from 1 hour before the midday till 1 hour after the midday so that shading over the bottom panel can be eradicated. According to the theory, the proposed distance between two consecutive panels is selected to be 1.5 times of the panel width .This will allow the top and bottom panels get the full sun exposure during noon, as shown in Fig. 2. The angular separation between the vertical bar and the mounted solar panel stack will be 67.5º at sunrise. The total height of the structure will be (1.5+1.5+0.5) or 3.5 time the panel width, as can be seen from Fig. 1.

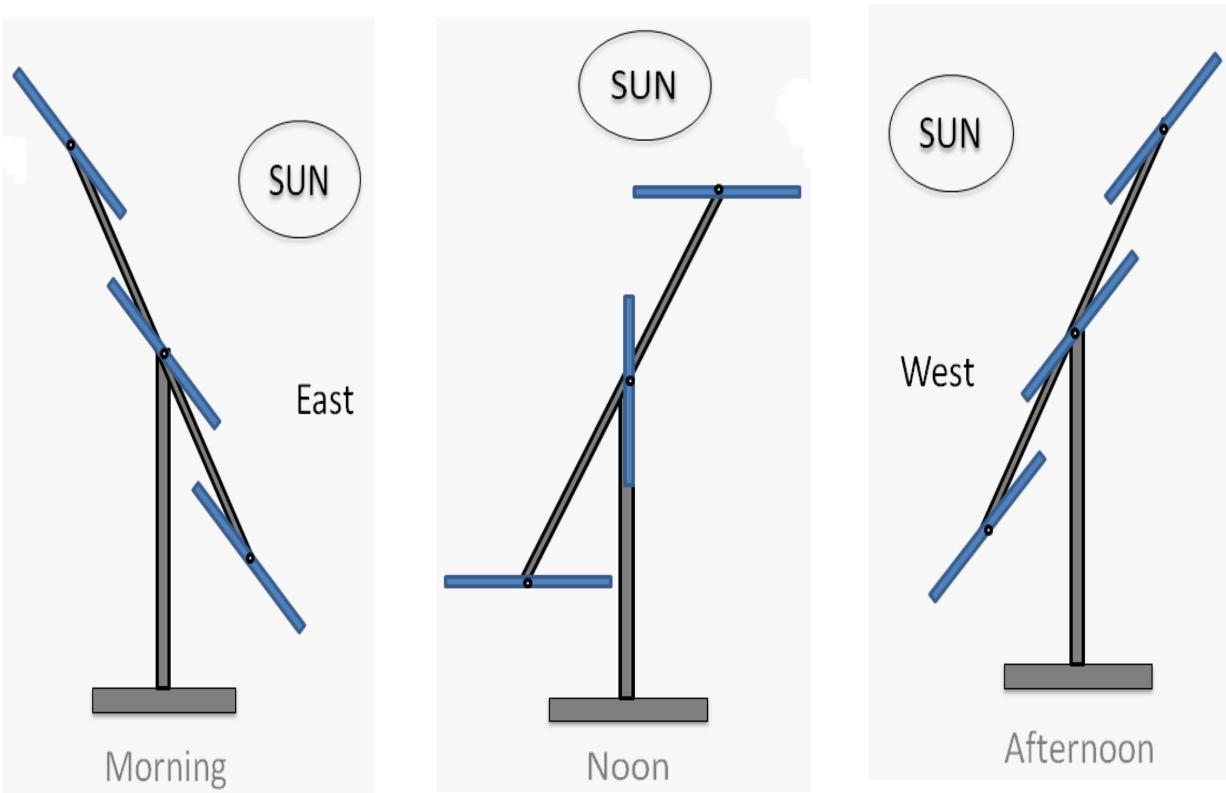



Figure 1.2: Positions of the solar panels throughout different times of the day.

## 1.3 Scope of the project

As it was already been proposed to construct a prototype multilevel solar panel and it has already been constructed. The next plan will be increasing the efficiency of panel as much as possible. For that we have to use the same component those we have used for constructing this solar panel and which includes Arduino, Mini Maestro (motor shield), RTC(Real Time Clock), Relay Break out board, 12V actuator and battery for power supply. To increase the efficiency, the panels that will be used further should generate much more power.

## 1.4 Project Objective

The consumption of energy is growing everyday as the technology is developing day by day. Hence adequate sources of electricity are necessary for ultimate living standard and well-being of a country. At present power generation in Bangladesh is not enough to meet the growing consumers demand. So it is impossible to guarantee an endless supply of electric power to all consumers throughout the country. The sun can deliver us with unlimited solar energy. It is big task and matter of wisdom, that how we can make the efficient use of solar energy. The sun is providing with solar energy every hour in same amount and the entire nation is using that. So we could use this huge amount of resources by employing solar photo-voltaic systems. But the conventional PV modules are less efficient and they require a lot of floor space. A developing country like Bangladesh which is facing the scarcity of electricity where only 62 percent people are somehow been sheltered under the shade of electricity and in the meantime where there is a big shortage of space. It is such a big questions that where and how we can implement the idea of setting a solar panel. So plenty space is a chief issue for establishing solar PV modules in cities where rooftop space is less and that is why our project and paper shows and depicts the way that how we can set up systems built on photovoltaic in a modern and scientific way.

# CHAPTER 2

## System Description

### 2.1 Introduction

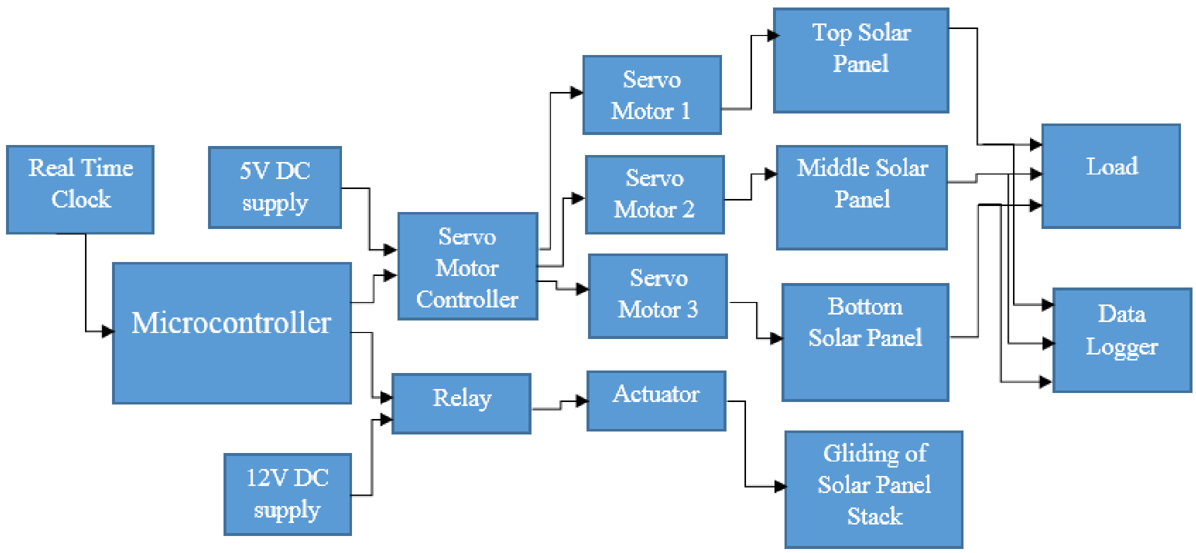
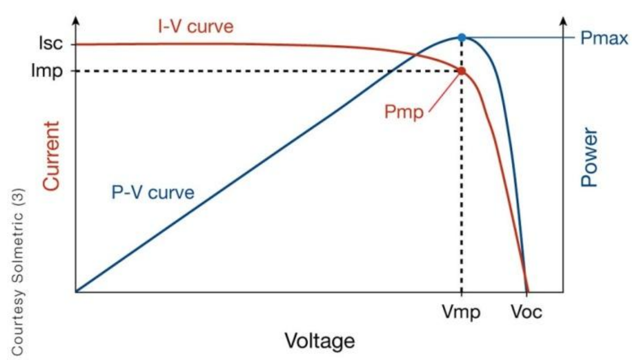



Figure 2.1: Block diagram of the system.

Fig. 3 shows the block diagram of the multilevel solar panels system. In the system, three solar panels mounted on stack transform energy from the sun into electrical energy. The rotation of the solar panels and the linear movement of the actuator are controlled by a control circuit that reads the time from the Real Time Clock as an input. A battery accumulates the electrical energy from where the load draws necessary power. The voltages and currents generated by the panels are also passed through the voltage and current sensors of a predesigned data logger for recording those values in a memory card. The paper targets at the design and execution of physical construction of the rack of the solar panels and control circuit.

## 2.2 Photovoltaic Technology

Something which is very much well known for being most plentiful and suitable source of energy is known as solar energy. The solar panel comprises of photovoltaic cells. PV acquires its name from the process of converting light (photons) to electricity (voltage), which is called the photovoltaic effect. “Photo” means light and “Voltaic” means producing electricity. So photo voltaic process is “making electricity directly from sunlight”. Generally the output of the power of a solar cell or photovoltaic cell is dependent on projected light on the cell. Time of the day, season, panel position and alignment are also the major issues behind the output power. The Current (I), Voltage (V) and the power (P) features of photovoltaic cell are shown in the figure below:




The normal I-V curve (red) and P-V curve (blue) shown here could represent any portion of a PV array—from a single cell, to a cell string or module, up through the array itself. The points making up this curve can be measured with a single connection and a single piece of equipment.

Figure 2.2: I vs. V and P vs. V curve.

## 2.3 Solar Panel

Solar panels are devices which converts light into electricity. They are called "solar" panels because most of the time, the most powerful source of light available is the Sun, called Sol by astronomers. Some scientists call them photovoltaic which means, basically, "light-electricity.” It is a form of photoelectric cell, defined as a device whose electrical characteristics—e.g. current, voltage, or resistance—differ when subjected to light. When the numbers of the photovoltaic cells are gathered together, it will create a solar module which will produce electric power from sunlight. A solar photovoltaic panel is in the combination of multiple cells in one plane. The efficiency of most solar panel is from 11-15 percent. The efficiency rating measures what percentage of sunlight striking a panel gets converted into electricity that can be used. The higher efficiency, the fewer surface area we'll need in our solar panels. Though the average percentage may be tiny, we can effortlessly fit a typical roof with enough power to cover our energy needs. There are a number of different types of panel and there are variations in properties among those panels as well. There are mostly three different categories of solar panel:

1. Monocrystalline Silicon Solar Cells
2. Thin-Film Solar Cells (TFSC) (Amorphous Silicon Solar Cells)
3. Polycrystalline Silicon solar cells

(a) (b) (c)

Figure 2.3: (a) Monocrystalline Silicon Solar Cells (b) Thin-Film Solar Cells (TFSC) (Amorphous Silicon Solar Cells) (c) Polycrystalline Silicon solar cells.

Mono crystalline photovoltaic electric solar energy panels have been the go-to choice for many years. They are among the ancient, most efficient and most reliable ways to yield electricity from the sun. Since cost efficiency is one of the major issues to be taken into consideration, so it does not make it appropriate for the system to be designed.

A thin film solar cell is a second generation solar cell that is produced by dropping one or more thin layers or thin film (TF) of photovoltaic material on a substrate, such as glass, plastic or metal. Since the output of electrical power is little, solar cells built using amorphous silicon have usually only been used for small-scale uses, example- calculators. But, new advances have made them more attractive for few large-scale uses too. By using amorphous silicon, mass production is easy but amorphous solar panels are usually not very beneficial for maximum residential sites. These Panels are very low-cost but at the same time, they also need enormous space and the efficiency of amorphous modules to produce sunlight to electricity is half polycrystalline and half mono crystalline .Therefore, this will be not a sensible decision to choose the amorphous silicon.

The current benefits that have been presented in polycrystalline solar panel have made it more alike in efficiency as we get in monocrystalline panels. Among all the advantages, it does have one advantage that is something extra ordinary, and that is, it is not affected by the shades of nearby buildings and shaded trees. Furthermore, we will acquire it in smaller rate and is much more efficient. So, for our usage, as we will attempt to use our panels in urban areas like Dhaka city, it is highly advised to use polycrystalline solar panel because of its efficiency.

## 2.4 Servo Motor-MG 995

Servo motors (or servos) are self-contained electric devices (see Figure 2.3 below) that rotate or drive parts of a machine with great accuracy. A Servo is a small device that uses a two wire DC motor, a gear train, a potentiometer, an integrated circuit, and an output shaft. Of the three wires that stick out from the motor casing, one is for power, one is for ground, and one is a control input line. The rating of servo is 6 volts and delivers 66.7 oz-in. of maximum torque at 70r/min. They are fitted with a servo mechanism for accurate control of angular position. The RC servo motors typically have a rotation limit from 90° to 180°. Few servos do have rotation limit of 360° or more. But servos do not rotate continually. Their rotation is limited in between the fixed angles. We are using three servo motors in our system which consequently rotates the upper, middle and bottom panel as time goes on. The servo motors are not directly connected to the arduino but to the pololu mini maestro 12. The pololu will run the three servo motors of our system.

**Features:**

- Voltage: 4.8-6.0 Volts
- Torque: 45.8/66.7 oz-in. (4.8/6.0V)
- Speed: 60/70 r/min (4.8/6.0V)
- Rotation: 360°
- Dual Ball Bearing
- 4 Plastic Gears + 1 Metal Gear
- 25T Spline
- Rest point adjustment


Figure 2.4: Servo Motor Mg 995.

Servo controllers have three wires- Brown, Red and Orange. Brown connects to the ground of the Arduino. Red connects to the 5V on the Arduino. Orange connects to the I/O pin on the Arduino, in this case digital pin 9. Jumper wires are used to connect between the female connectors on the servo cable and Arduino headers.

Figure 2.5: The way servo motors are attached with the solar panels in the system.

In this planned project servo motors are used to rotate the panel. Sun rises in the east and sets in the west. So our panels will start rotating from east to the west. At the initial period all the panels are oriented and faced towards east, it will then start rotating from east by 15 degree towards west and it will take total 12 steps to complete the whole rotation part.

## 2.5 Servo Controller (Pololu Mini Maestro)

The Mini Maestros or servo controllers are highly adaptable servo controllers and general-purpose I/O boards. They support three control methods:

1) USB for direct connection to a computer
2) TTL serial for use with embedded systems and

3) Internal scripting for self-contained, host controller-free applications. The channels can be configured as servo outputs for use with radio control (RC) servos or electronic speed controls (ESCs), as digital outputs, or as analog/digital inputs.

A free configuration and control program is available for Windows and Linux, making it simple to configure and test the device over USB, create sequences of servo movements for animatronics or walking robots, and write, step through, and run scripts stored in the servo controller. The Mini Maestros' 8 KB of internal script memory allows storage of up to approximately 3000 servo positions that can be automatically played back without any computer or external microcontroller connected.

There are number of Pololu's servo controllers with different number of channels such as:

1. Micro Maestro 6

2. Mini Maestro 12-Channel USB Servo Controller

3. Mini Maestro 18-Channel USB Servo Controller

4. Mini Maestro 24-Channel USB Servo Controller

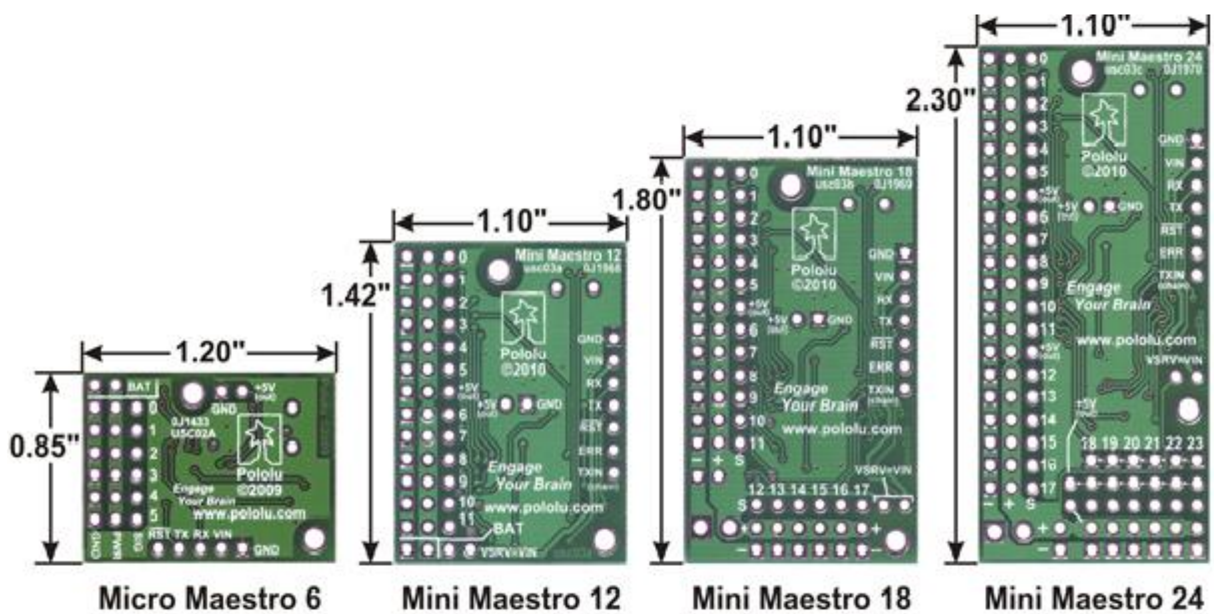



Figure 2.6: Different types of servo controller of Pololu.

For our system we have chosen Mini Maestro 12 channels USB controller which can run 12 servo motor at a time. In our system there are total 3 panels and we have used total three servo motors so we have used mini maestro 12. High current flow is needed to be supplied to run the servo which is provided by the mini maestro other if arduino was used to provide such high current it would damage.

If we use Mini Maestro, it will reduce the usage of wiring of the system and it will be much easier to make the programming code by putting the exact pin numbers in the program of that specific servo motor.

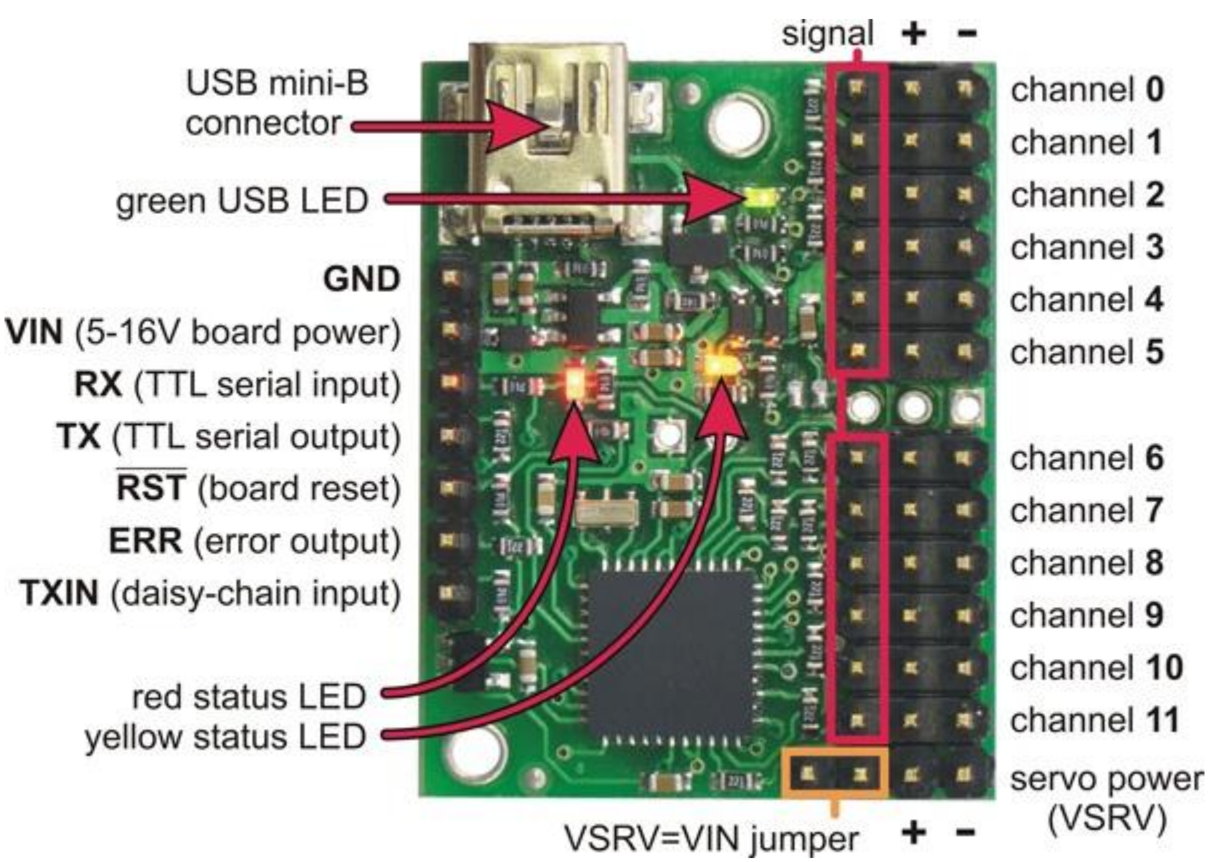



Figure 2.7: Mini Maestro 12-channel USB servo controller (fully assembled) labeled top view.

## 2.6 Arduino

For constructing electronics projects, we use an open-source platform which is known as "Arduino". It can be used to write and upload computer code to the physical board. The board consists of all the components that a miniature microcontroller requires to run. Arduino contains a physical programmable circuit board (often referred to as a microcontroller and a piece of software, or IDE (Integrated Development Environment) that runs on our computer. A very significant feature of the Arduino is the typical way that connectors are all visible, which permits the CPU board to be linked to different interchangeable add-on modules which is termed as shields. Certain shields communicate with the Arduino board directly over several pins, but numerous shields are separately addressable through an $I^2C$ serial bus, letting many shields to be arranged and used.

There are generally two types of Arduino for this sort of system:

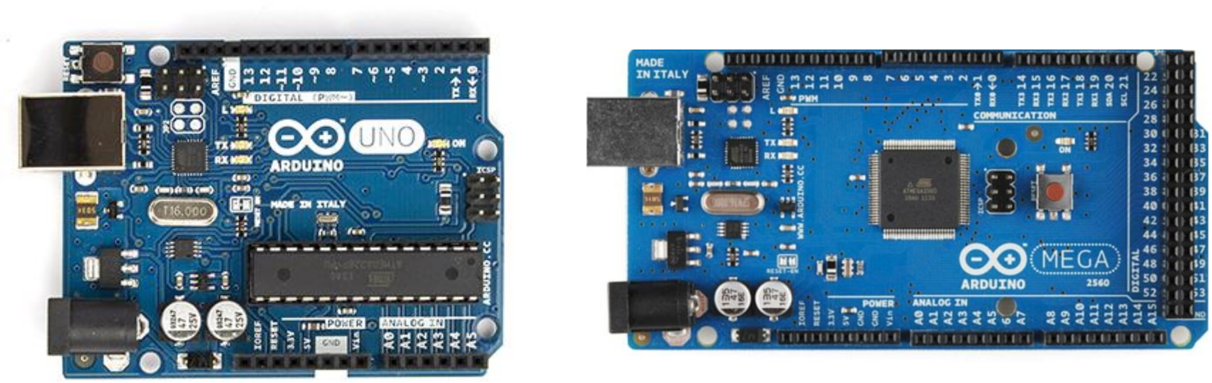
1. Arduino UNO
2. Arduino MEGA

### 2.6.1 Arduino Uno

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It simply consists of everything required to support the microcontroller. We can attach it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. In other words, it can be attached to a computer with a USB cable or powered it with an AC-to-DC adapter or a battery to get started.

### 2.6.2 Arduino Mega

The Arduino Mega is a microcontroller board based on the ATmega1280. It has total 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains each and everything that are needed to support the microcontroller; simply we need to connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.




(a) (b)

Figure 2.8: (a) Arduino Uno (b) Arduino Mega.

**Application Ideas**

- LED display/LCD controller
- Pulse width modulation driver
- Robot controller
- Data acquisition systems
- Alarm systems
- Programmable logic controllers
- Embedded Web servers
- Control Systems

At present, the Arduino hardware platform has the power and reset circuitry setup as well as circuitry to program and transfer data over USB. Moreover, the I/O pins of the microcontroller are usually already provided out to sockets/headers for very easy access (This can differ a little bit with the particular model). Arduino and 8051 are almost similar. This is because, one is a microcontroller and while the other one is an improvement platform. Arduino runs (quite originally) on Atmel's AVR microcontroller series. The main variations are the major variations in the architecture of the controllers. Thinking from the perspective of user, AVR is richer in peripheral. It consists of in-built ADC and numerous timer based peripherals. It is very easy to program when compared to microcontroller. Thinking from the software side, Arduino does offers numerous libraries making the microcontroller programming easier. The easiest of these are functions to regulate and read the I/O pins rather than having to use the bus/bit masks generally used to interface with the Atmega I/O (This is a slight problem). More suitable are factors such as being able to set I/O pins to PWM at a particular duty cycle using a single command or performing Serial communication. The highest benefit is to have the hardware platform set up already, especially the circumstance that it permits programming and serial communication over USB. This saves the problem of having to do PCB (whose value is more than an Arduino) or breadboarding. The main benefits are the speed as well as ease of development. The open source nature of Arduino has caused the accessibility of various “shields" (i.e. daughter cards) that promote advantages, such as, as Internet access, wireless networking, data logging and device control. Another big benefit is the Arduino IDE which permits software development on every main platform (Mac, PC, and Linux) with an easy-to-use subset of C/C++.

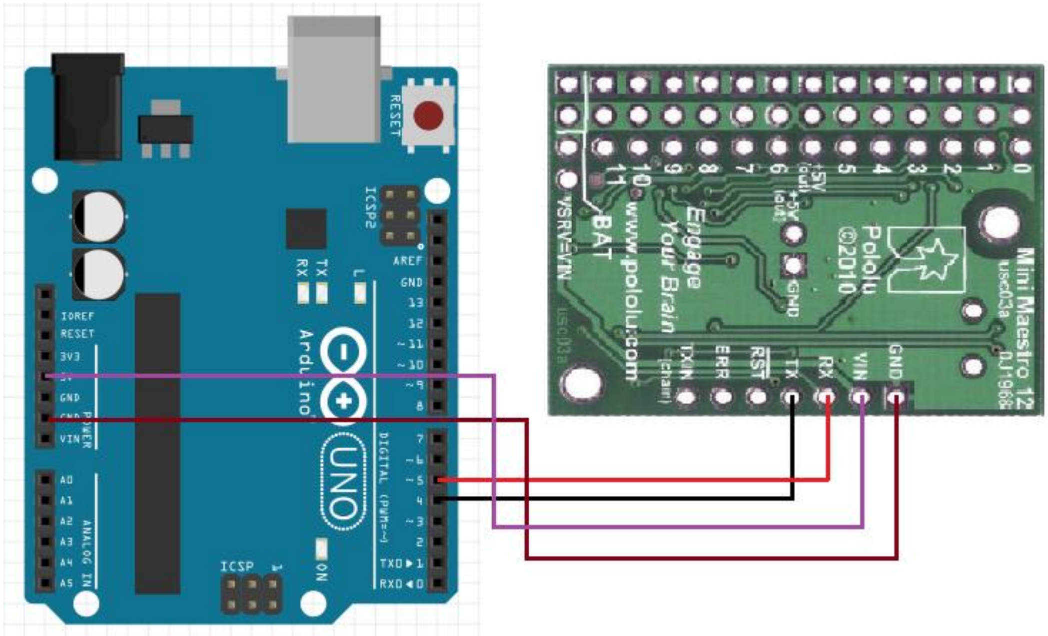


Figure 2.9: Connection of Arduino Uno with Pololu Mini Maestro 12.

## 2.7 Real Time Clock (DS3231)

The component which permits microcontroller to keep track of time even if it is reprogrammed, or if the power is lost, that component is known as RTC(Real Time Clock). The RTC is ideal for different purposes, such as-

1) Data logging
2) Clock-building
3) Time stamping
4) Timers and
5) Alarms etc.

The DS3231 is a low-cost, extremely accurate I²C real-time clock (RTC) with an integrated temperature-compensated crystal oscillator (TCXO) and crystal. The device incorporates a battery input, and maintains accurate timekeeping when main power to the device is interrupted. The integration of the crystal resonator enhances the long-term accuracy of the device as well as reduces the piece-part count in a manufacturing line. The DS3231 is available in commercial and industrial temperature ranges, and is offered in a 16-pin, 300-mil SO package.

The RTC maintains seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an active-low AM/PM indicator. Two programmable time-of-day alarms and a programmable square-wave output are provided. Address and data are transferred serially through an $I^2C$ bidirectional bus.

A precision temperature-compensated voltage reference and comparator circuit monitors the status of $V_{CC}$ to detect power failures, to provide a reset output, and to automatically switch to the backup supply when necessary. Additionally, the active-low RST pin is monitored as a pushbutton input for generating a µP reset [4].

**Key Features**

- Highly Accurate RTC Completely Manages All Timekeeping Functions
    - Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and Year, with Leap-Year Compensation Valid Up to 2100
    - Accuracy ±2ppm from 0°C to +40°C
    - Accuracy ±3.5ppm from -40°C to +85°C
    - Digital Temp Sensor Output: ±3°C Accuracy
    - Register for Aging Trim
    - Active-Low RST Output/Pushbutton Reset Debounce Input
    - Two Time-of-Day Alarms
    - Programmable Square-Wave Output Signal
- Simple Serial Interface Connects to Most Microcontrollers
    - Fast (400kHz) $I^2C$ Interface
- Battery-Backup Input for Continuous Timekeeping
    - Low Power Operation Extends Battery-Backup Run Time
    - 3.3V Operation
- Operating Temperature Ranges: Commercial (0°C to +70°C) and Industrial (-40°C to +85°C)


Figure 2.10: DS3231 (RTC).

DS3231 has been chosen because of its convenient characteristics compared to its nearest competitor DS1307. While using DS1307, the external temperature can affect the frequency of the oscillator circuit which drives the DS1307's internal counter. This will usually result with the clock being off by around five or so minutes per month. The DS3231 is much more accurate, as it has an internal oscillator which isn't affected by external factors – and thus is accurate down to a few minutes per year at the most [5].

## 2.7.1 Wiring It Up with the Arduino

- SCL pin of DS323 is to be plugged into A5 port of Arduino Uno
- The SDA pin plugs into the A4 port
- The VCC pin plugs into the 5V port of the Arduino
- GND plugs into the GND port.

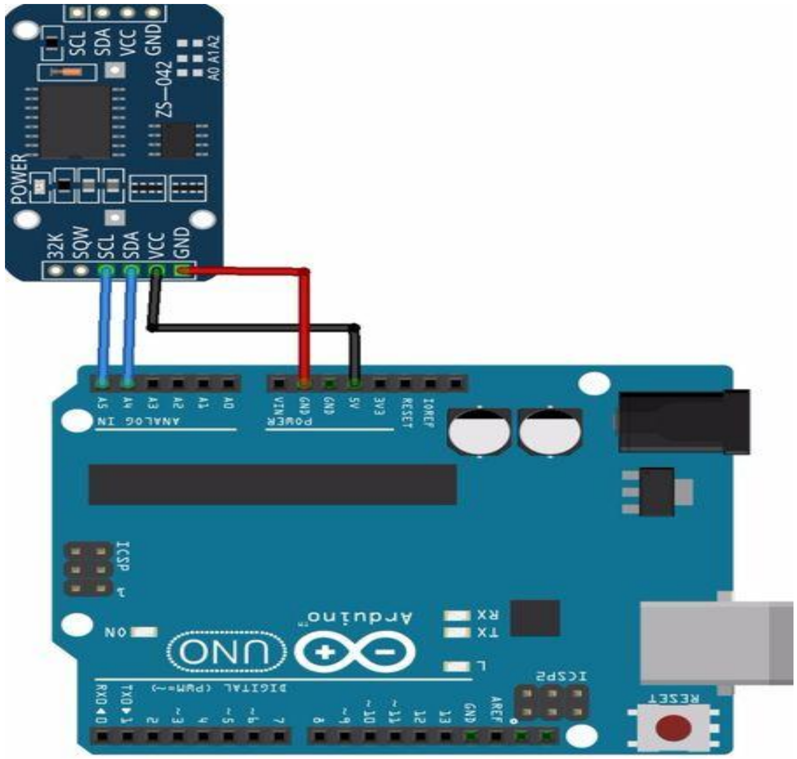



Figure 2.11: RTC DS 3231 connected to Arduino.

## 2.7.2 CR2032 Lithium Battery

Specifications of CR2032 battery is given below [6]:

| Item | Performance Characteristics |
|---|---|
| Nominal Voltage | 3 V |
| Nominal Capacity | 220 mAh |
| Continuous Standard Drain | 0.2 mA |
| Diameter | 20 mm |
| Height | 3.2 mm |
| Approx. Weight | 2.9 g |
| Operating temperature | -30 to 60 degree C |

## 2.8 Actuator

An actuator is one type of motor that plays an important role for controlling and moving a structure. It normally functions by the influence of an energy source, for example- electric current, hydraulic fluid pressure, or pneumatic pressure, and converting the input energy into motion. An actuator is the device by which a control system acts depending on the surrounding. The control system can be very simple. It can be software-based, a human, or any other input. There are different categories of actuators .Example- Mechanical actuators, Hydraulic actuators, Pneumatic actuators, Electro-mechanical and Linear motor etc. In our multilevel sun tracking solar panel, we have used linear actuator because of the fact that, motion is created in a straight line in linear actuator, compared to the circular motion of a conventional electric motor.

**Advantages and Disadvantages:**

| Actuator Type | Advantages | Disadvantages |
|---|---|---|
| Mechanical actuators | Inexpensive, repeatable, no need of power source, self-sufficient, similar behavior extending or retracting. | Operated manually.<br>No need of automation. |
| Hydraulic actuators | Possible to create very high forces. | It can leak under some circumstances.<br>For repeatability, there is need of position feedback.<br>There is need for external hydraulic pump. Few designs are suitable in compression only. |
| Linear motor | Very simple design. | Low to medium force. |

| | | |
|---|---|---|
| | Minimum moving parts. Achieving high speeds is possible.<br>Self-sufficient.<br>Similar behavior when extending or retracting. | |
| Electro-mechanical | Cheap, repeatable and operation can be automated. Self-contained. Identical behavior extending or retracting. DC or stepping motors. Position feedback possible. | Many moving parts prone to wear |
| Pneumatic actuators | Strong, light, simple and fast. | Precision position control impossible except at full stops |

The main role of the actuator in our system is to move our whole system back and forth in a day. Initially the actuator remains inactive from the morning to the noon. When it is noon the actuator becomes active and shifts the whole system to just opposite alignment compared to its initial position. Initially the panels remain in such a position to capture maximum sunlight so the actuator operation is not required here. At noon when the sun is just at the top of the system the panels do not receive much energy so the actuator needs to shift the system towards its opposite direction to get the full exposure of the sunlight to the panels.

Figure 2.12: An electronic actuator in our system.

The actuator in our system will be provided with power from a battery source rather than a power source because the power rating of the actuator is huge which is 12V compared to the power rating of other components used. It is seen that when the actuator is getting power supply from the power source the real time clock in our system gets reset. The real time clock gets its power from a lithium battery and is grounded to a common point where all the components are grounded. The real time clock gets reset because its power rating is small and when this huge power flows through it real time clock gets reset. When the actuator in active it creates magnetic flux around it so the interference between the magnetic flux and the electric flux also reset the real time clock and this interference also causes problems to other circuitry of the system so the entire actuator operation and circuitry is separated from the whole system.

## 2.9 Relay Breakout Board

A relay is an electrically operated switch. Current flowing through the coil of the relay creates a magnetic field which attracts a lever and changes the switch contacts. The coil current can be on or off so relays have two switch positions and they are double throw (changeover) switches. The relay's switch connections are usually labeled COM (POLE), NC and NO. The operating voltage of relay is 6v.

COM/POLE= Common, NC and NO always connect to this; it is the moving part of the switch.
NC = Normally Closed, COM/POLE is connected to this when the relay coil is not magnetized.
NO = Normally Open, COM/POLE is connected to this when the relay coil is MAGNETIZED and vice versa.

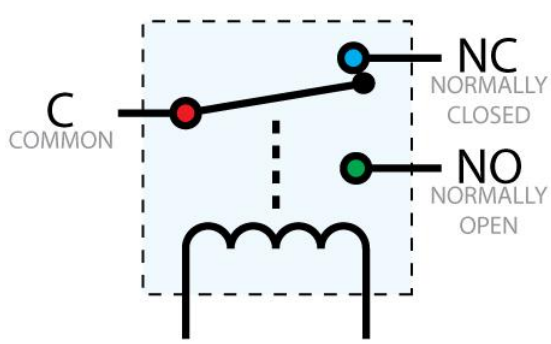



Figure 2.13: Relay and its symbols.

There are 5 Pins in a relay. Two pins A and B are two ends of a coil that are kept inside the relay. The coil is wound on a small rod that gets magnetized whenever current passes through it.COM/POLE is always connected to NC (Normally connected) pin. As current is passed through the coil A, B, the pole gets connected to NO (Normally Open) pin of the relay. In our system, we used a 2 relay module.

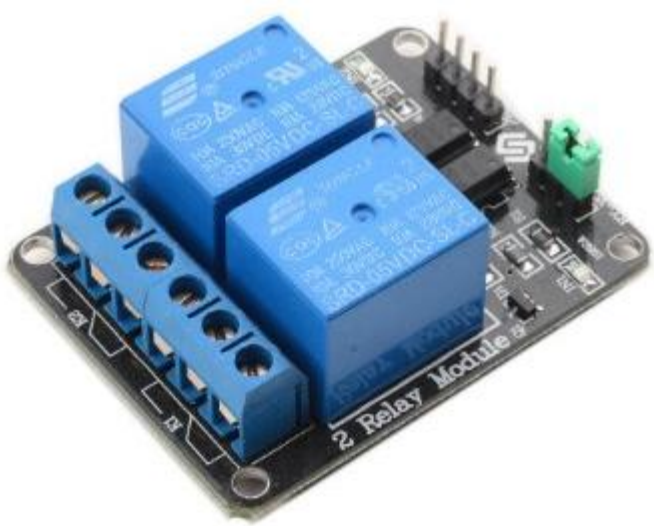



Figure 2.14: 2 Relay Module.

The two relays are used to toggle the positive and negative power to the actuator in order to determine the direction of piston travel. 12V power supply is given to the NC pin and the NO pin is grounded when the actuator is inactive. When NC pin is grounded and NO pin is given 12V the relay switch is pressed and the code reads the current position, and then determines which direction to drive the piston to reach the goal position. Once the goal position is reached, it shuts off the power to the motor.

## 2.10 Sun tracker

A solar tracker is a device that orients a payload toward the sun. Payloads are photovoltaic panels. There are different types of sun trackers. These are used for special purposes. Some of these tracking systems are discussed below:

### 2.10.1 Need of a Sun Tracker

Each day, the sun rises in the east, moves across the sky, and sets in the west. Whenever the sun is shining on us, it is sending energy in our direction. We can feel the heat from the sun, and we can see objects that are illuminated by the light from the sun as it moves across the sky. However, if we could get a solar cell to turn and look at the sun all day, then it would be receiving the maximum amount of sunlight possible and converting it into the more useful energy form electricity. If we are located in the tropics, we see that the sun appears to follow a path that is nearly directly overhead. However, for locations north or south of the tropics (e.g., latitudes greater than 23.5 degrees), the sun never reaches a position that is directly overhead. Instead, it follows a path across the southern or the northern part of the sky.

### 2.10.2 Solar Tracker Fundamentals

A solar tracker is a device that is used to align a single P.V module or an array of modules with the sun. Although trackers are not a necessary part of a P.V system, their implementation can dramatically improve a systems power output by keeping the sun in focus throughout the day. Efficiency is particularly improved in the morning and afternoon hours where a fixed panel will be facing well away from the sun’s rays. P.V modules are expensive and in most cases the cost of the modules themselves will outweigh the cost of the tracker system. Additionally a well-designed system which utilizes a tracker will need fewer panels due to increased efficiency, resulting in a reduction of initial implementation costs.

### 2.10.3 Different Types of Tracking

**Active tracker:**

These trackers use motors and gear trains in directing the tracker as commanded by a controller responding to the solar direction. Active two axis trackers are used to orient heliostats. Light sensing type of trackers has two photo sensors, photodiodes, configured differentially so that they output a null when receiving the same light flux. They are supposed to be flat and are 90 degrees apart. Since the motors consume energy, use them only when necessary. Now if the light is below some threshold level, there may not be enough power available for reorientation. Precaution should be taken so as not waste energy during cloudy days when there is not enough light.

**Passive tracker:**

This type of tracker works on the principle of creating a gas pressure by using a low BP compressed gas fluid. This fluid is then driven to one side or the other due to solar heating thus creating gas pressure. This causes the tracker to move in response to an imbalance. These kinds of trackers are provided with Viscous Dampers to prevent excessive motion in response to wind gusts and Shader or Reflectors to reflect the early morning light to start the panel tilting towards the sun.

**Open loop trackers:**

Open loop trackers determine the position of the sun using computer controlled algorithms or simple timing systems.

**Altitude/Azimuth Trackers:**

Use astronomical data or sun position algorithms to determine the position of the sun for any given time and location. Tracker location, date and time are used by a microcontroller to fix the position of the sun. Once the position has been calculated, the modules are moved using servo motors and the position is measured by encoders built into the tracker frame. There are many types of solar trackers, of varying costs, sophistication, and performance.

### 2.10.4 Types of Solar Tracker

There are many different types of solar tracker which can be grouped into single axis and double axis models.

**Single axis tracker:**

**Features**

- Track the sun from east to west using a single pivot point
- Increase solar yield up to 34%
- Simple, effective design
- Low maintenance
- Lower cost compared to dual axis
- Minimal points of failure

**Horizontal single axis tracker:**

Single axis horizontal trackers may be oriented by either passive or active mechanisms. In these, a long horizontal tube is supported on bearings mounted upon pylons or frames .The axis of the tube is on a north-south line. Panels are mounted upon the tube, and the tube will rotate on its axis to track the apparent motion of the sun through the day. Since these do not tilt toward the equator they are not especially effective during winter midday (unless located near the equator),but add a substantial of productivity during the spring and summer seasons when the solar path is high in the sky. These devices are less effective at higher latitudes .The principal advantage is the inherent robustness of the supporting structure and the simplicity of the mechanism. Since the panels are horizontal, they can be compactly placed on the axle tube without danger of self-shading and are also readily accessible for cleaning. For active mechanisms, a single control and motor may be used to actuate multiple rows of panels.

### Vertical single axis tracker:

A single axis tracker may be constructed that pivots only about a vertical axle, with the panels either vertical or at a fixed elevation angle. Such trackers are suitable for high latitudes, where the apparent solar path is not especially high, but which leads to long days in summer, with the sun travelling through a long arc.

### Polar single axis tracker:

This method is scientifically well known as the standard method of mounting a telescope support structure. The tilted single axis is aligned to the polar star. It is therefore called a polar aligned single axis tracker (PASAT). In this particular implementation of a tilted single axis tracker, the tilt angle is equal to the site latitude. This aligns the tracker axis of rotation with the earth's axis of rotation.

### Tilted single axis tracker:

All trackers with axes of rotation between horizontal and vertical are considered tilted single axis trackers. Tracker tilt angles are often limited to reduce the wind profile and decrease the elevated end height. Field layouts must consider shading to avoid unnecessary losses and to optimize land utilization. With backtracking, they can be packed without shading perpendicular to their axis of rotation at any density. However, the packing parallel to their axes of rotation is limited by the tilt angle and the latitude. Tilted single axis trackers typically have the face of the module oriented parallel to the axis of rotation. As a module tracks, it sweeps a cylinder that is rotationally symmetric around the axis of rotation.

### Dual axis tracker:

**Features**

- Track the sun from east to west and north to south using two pivot points
- Increase solar yield up to 37%
- Complex design-more motor and sensors
- Higher maintenance

- Higher cost due to additional parts and installation time
- Additional points of failure

**Tip-tilt dual axis tracker (TTDAT):**

A tip–tilt dual axis tracker is so-named because the panel array is mounted on the top of a pole. Normally the east-west movement is driven by rotating the array around the top of the pole. On top of the rotating bearing is a T- or H-shaped mechanism that provides vertical rotation of the panels and provides the main mounting points for the array. The posts at either end of the primary axis of rotation of a tip–tilt dual axis tracker can be shared between trackers to lower installation costs. Other such TTDAT trackers have a horizontal primary axis and a dependent orthogonal axis. The vertical azimuthal axis is fixed. This allows for great flexibility of the payload connection to the ground mounted equipment because there is no twisting of the cabling around the pole. Field layouts with tip–tilt dual axis trackers are very flexible. The simple geometry means that keeping the axes of rotation parallel to one another is all that is required for appropriately positioning the trackers with respect to one another. Normally the trackers would have to be positioned at fairly low density in order to avoid one tracker casting a shadow on others when the sun is low in the sky. Tip-tilt trackers can make up for this by tilting closer to horizontal to minimize up-sun shading and therefore maximize the total power being collected. The axes of rotation of many tip–tilt dual axis trackers are typically aligned either along a true north meridian or an east west line of latitude. Given the unique capabilities of the Tip-Tilt configuration and the appropriated controller totally automatic tracking is possible for use on portable platforms. The orientation of the tracker is of no importance and can be placed as needed.

# CHAPTER 3

## Calculations of System Size

### 3.1 Introduction

This part of the chapter explains the calculation of the entire solar system. Hence, calculations are now needed to be done in order to determine the parameters for a photovoltaic system to generate electricity. The system will require solar panel for the collection of sun energy and servo motors to run them. The torque speed of the motors will determine the angle and speed at which the panels will rotate. Furthermore, the total calculation of the energy used in the system also depends on each component's power utility.

### 3.2 Solar Panel Calculations

In our system we have selected polycrystalline semi flexible solar panel for its low price availability in the market and they are easy to work with.

| | |
|---|---|
| Type of Cell | Polycrystalline |
| Maximum Power ($P_{max}$) | 18 W |
| Maximum Power Voltage ($V_{mp}$) | 19.4 V |
| Maximum Power Current ($I_{mp}$) | 0.9 A |
| Open Circuit Voltage ($V_{oc}$) | 23.7 V |
| Short Circuit Current ($I_{sc}$) | 1.0 A |

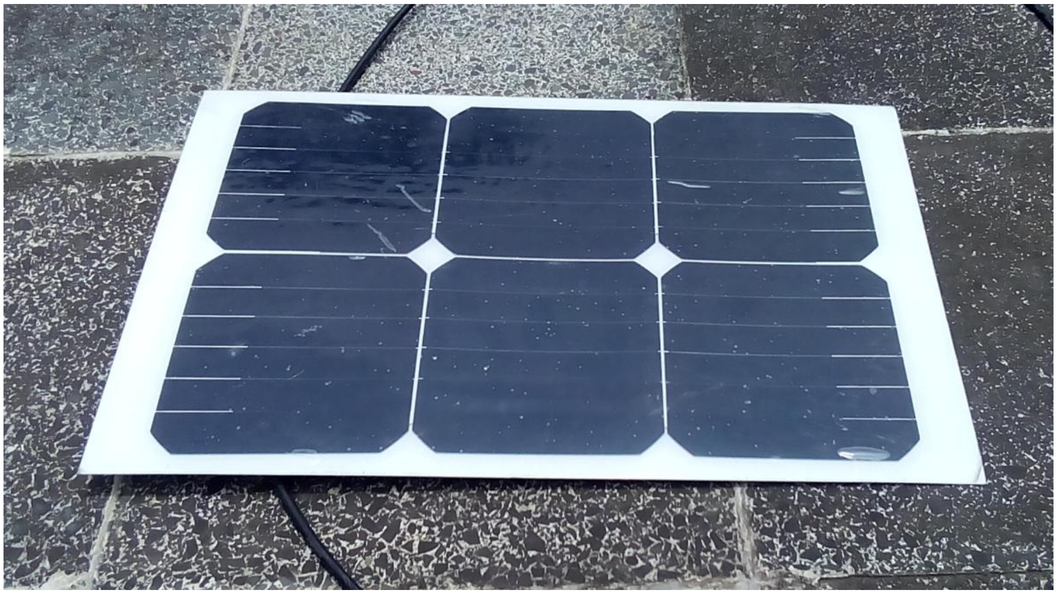


Figure3.1: 18W Solar Panel.

It is necessary to estimate the panel separation that will enable harnessing enough solar energy without making the system size too large. The distance between two panels is proportional to the time span of solar irradiance without shadow. If the distance is increased, longer time of no shadow will occur and vice versa. As horizontally two consecutive panels are separated by a fixed distance of half of the panel width, the sun angle at which shading of the lower panel begins will depend on the vertical separation between the panels. The greater is the vertical separation, the greater will be the sun angle and the higher will be the position of the sun in the sky for which shading of the lower panel will begin. That is the panels will get more sun exposure for a longer time with harnessing more solar energy if the panels separation is greater.

## 3.3 Calculating the distance between panels

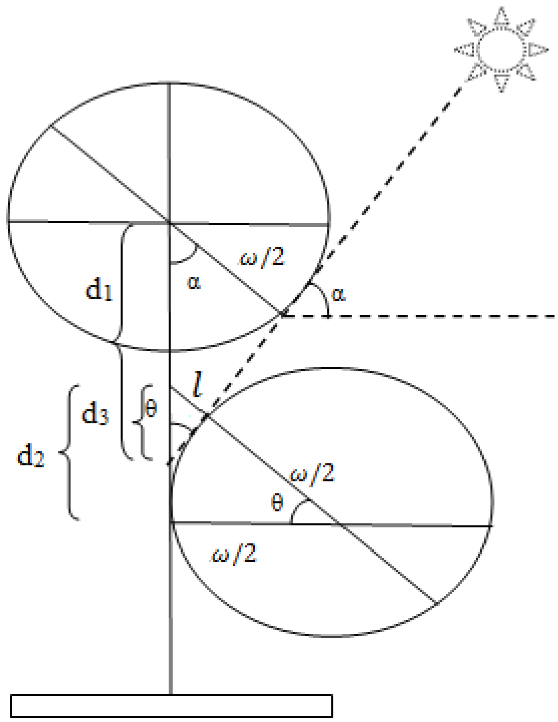



Figure 3.2: Finding distance between two panels.

But the increase in the panel separation will result the increase in the overall size and the cost of the system well. So that in our system we have come up with a detailed analysis where the panel's separation will not be too high but it will also be able to collect the high solar energy. Panel separation=d, is chosen to be 1.5 times of panel width in our design which corresponds to an angular position Φ=67.5º.With this panel separation, the total height of the structure will be (1.5+1.5+0.75) or 3.75 times the panel width. $d_3$

*w = width of the panel = 165mm*

*l = length of the panel = 225mm*

*An overall height of the structure = 3.75w = 3.75*165mm = 618.75cm*

$\alpha = altitude\ angle\ of\ sun\ w.r.t\ horizontal = 90° - \theta$

Now,

d1 = $\frac{w}{2sin\theta}$

d2 = $\frac{w \times tan\theta}{2}$

$(\frac{w}{2} + l) = \frac{d2}{sin\theta}$

$$\therefore l = \frac{d2}{sin\theta} - \frac{w}{2}$$

$$= \frac{wtan\theta}{2sin\theta}$$

$$= \frac{w}{2cos\theta} - \frac{w}{2}$$

$$= \frac{w(1 - cos\theta)}{2cos\theta}$$

d3 = $\frac{l}{sin\theta} = \frac{w(1-cos\theta)}{2sin\theta cos\theta}$

And finally we get,

Total distance between two panels, D = d1 + d2 - d3

$$= \frac{w}{2sin\theta} + \frac{w \times tan\theta}{2} - \frac{w(1-cos\theta)}{2sin\theta cos\theta}$$

It is estimated that with this panel separation, about 68% of total solar irradiation will fall on the middle panel. It can be noted here that if we want to increase energy collection from 68% to 75% for the middle panel, the minimum panel separation needed will be 2.5 times the panel width requiring the overall height to be (2.5+2.5+0.5)$w$ = 3674 mm, which is about 57% larger than the previous size for an increase of just 7% in incident energy.

## 3.4 Solar Angle Calculation:

Below is an overview of the angles involved in calculating the amount of solar radiation that a PV panel receives at any given time. The angle at which the sun hits a PV panel is the basis for understanding how to design the most efficient PV array for a specific location.

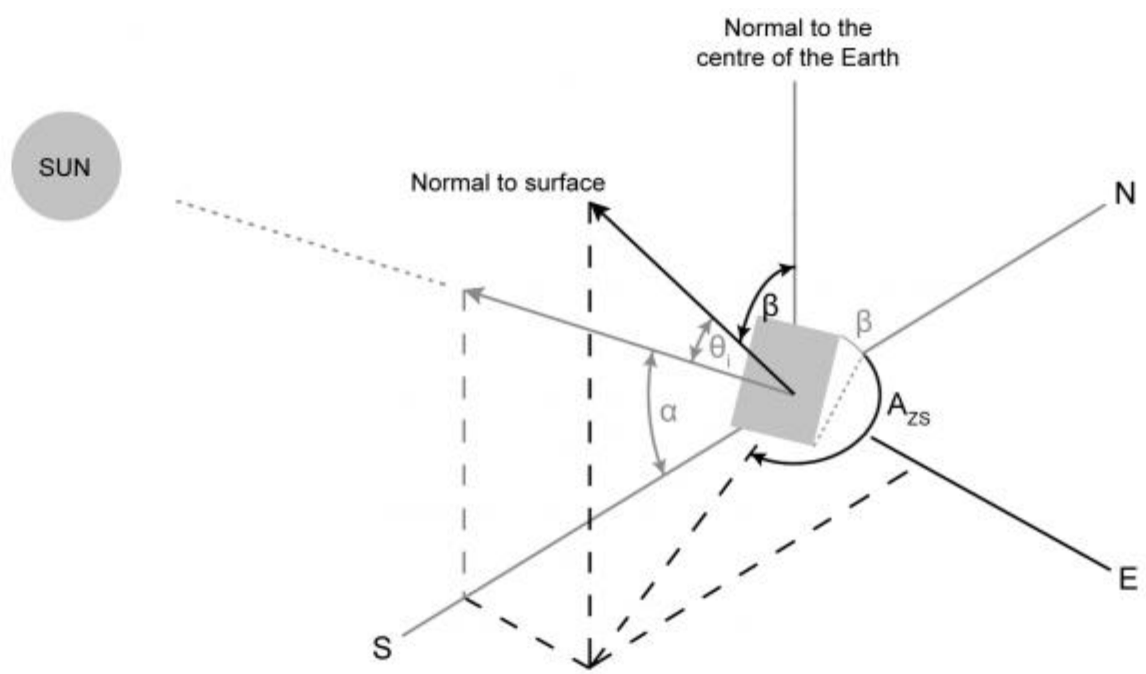



Figure 3.3: Solar Angles used in power calculations for PV panels.

**Solar Zenith Angle:** This is the angle between the line that points to the sun and the vertical — basically, this is just where the sun is in the sky. At sunrise and sunset this angle is 90º.

**Solar Altitude Angle, α:** This is the angle between the line that points to the sun and the horizontal. It is the complement of the zenith angle. At sunrise and sunset this angle is 0º.

**Solar Azimuth Angle:** This is the angle between the line that points to the sun and south. Angles to the east are negative. Angles to the west are positive. This angle is 0º at solar noon. It is probably close to -90º at sunrise and 90º at sunset, depending on the season. This angle is only measured in the horizontal plane; in other words, it neglects the height of the sun.

**Angle of Incidence, $\theta_i$:** This is the angle between the lines that points to the sun and the angle those points straight out of a PV panel (this is also called the line that is normal to the surface of the panel). This is the most important angle. Solar panels are the most efficient when pointing at the sun, so engineers want to minimize this angle at all times. To know this angle, you must know all of the angles listed and described next.

**Hour Angle:** This is based on the sun's angular displacement, east or west, of the local meridian (the line the local time zone is based on). The earth rotates 15º per hour so at 11am, the hour angle is -15º and at 1pm it is 15º.

**Surface Azimuth Angle:** This is the angle between the line that points straight out of a PV panel and south. It is only measured in the horizontal plane. Again, east is negative and west is positive. If a panel pointed directly south, this angle would be 0º.

**Collector Slope, β:** This is the angle between the plane of the solar collector and the horizontal. If a panel is lying flat, then it is 0º. As you tip it up, this angle increases. It does not matter which direction the panel faces.

**Declination, δ:** This is the angle between the line that points to the sun from the equator and the line that points straight out from the equator (at solar noon). North is positive and south is negative. This angle varies from 23.45 to -23.45 throughout the year, which is related to why we have seasons.

**Latitude:** This is the angle between a line that points from the center of the Earth to a location on the Earth's surface and a line that points from the center of the Earth to the equator. This can be easily found on a map.

## Angle Calculation :

**Declination angle calculation:**

The equation used to calculate declination angle in radians on any given day can be calculated by the following equation [3]:

$$\delta = 23.45 \frac{\pi}{180} \sin[2\pi(\frac{284 + n}{36.25})]$$

Where,

$\delta$ = declination angle (rads);

n = the day number (e.g. for 1st January n = 1)

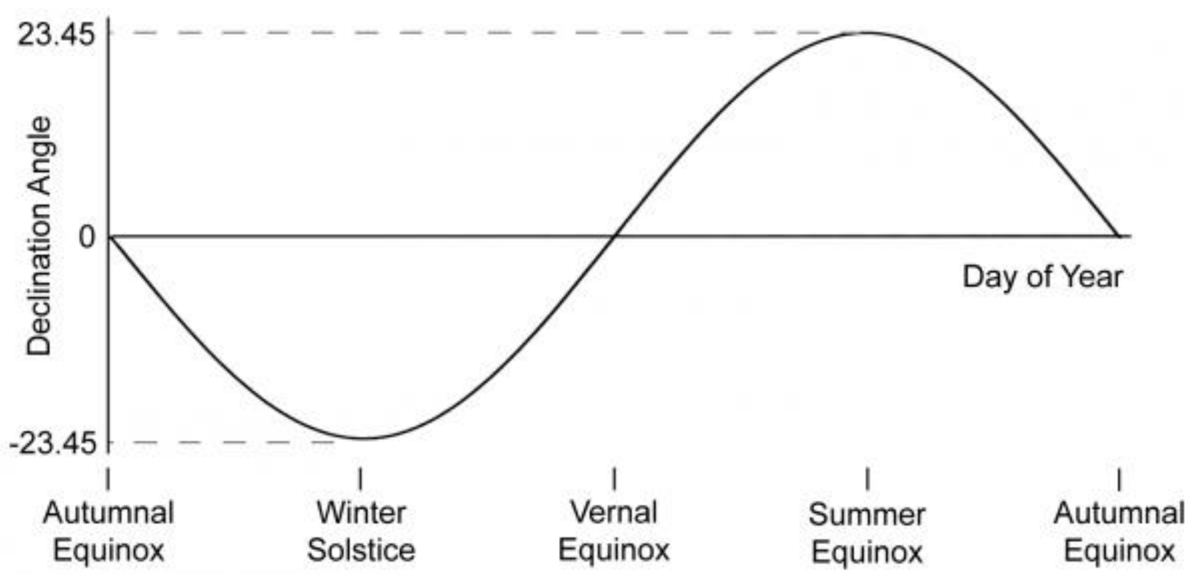



Figure 3.4: The variation in the declination angle throughout the year.

The declination angle is the same on the earth on a given day. Figure 18 shows the difference in the declination angle throughout a year. Because the period of the Earth's complete revolution around the Sun does not coincide exactly with the calendar year the declination varies slightly on the same day from year to year.

**The Hour Angle Calculation:**

The hour angle is positive during the morning, zero at noon and becomes negative increasingly as the afternoon advances. Two equations can be used to calculate hour angle when different angles are known [3]:

$$sin\,\omega = -\frac{\cos\alpha \sin Az}{\cos\delta}$$

$$sin\,\omega = \frac{\sin\alpha - \sin\delta \sin\varphi}{\cos\delta \cos\varphi}$$

Where,

$\omega$ = the hour angle

$\alpha$ = the altitude angle

$A_z$ = the solar azimuth angle

$\delta$ = the declination angle

$\varphi$ = observer's latitude

It is to be mentioned that the hour angle equals zero at the noon and since this angle changes by 15° every hour, it is very simple to calculate the hour angle at any time of the day. The hour angles at sunrise and sunset ($\omega_s$) are very important to be familiar with. They are basically of the same value. However, the sunrise angle is negative and the sunset angle is positive. Both can be calculated from:

$$\cos\omega_s = -\tan\varphi \tan\delta$$

**The Altitude Angle Calculation:**

The altitude angle can be calculated from the following equation [3]:

$$\sin\alpha = \sin\delta \sin\varphi + \cos\delta \cos\omega \cos\varphi$$

**The Azimuth Angle Calculation:**

The equation to calculate the azimuth angle is the following [3]:

$$\sin\alpha = \frac{\sin\omega\cos\delta}{\sin\theta z} = \frac{\sin\omega\cos\delta}{\cos\alpha}$$

The azimuth angle at sunrise ($A_{SR}$) can be calculated from:

$$\sin A_{SR} = -\sin\omega_S \cos\delta$$

**The Incidence Angle Calculation:**

The angle of incidence ($\theta_i$) of the Sun on a surface tilted at an angle from the horizontal ($\beta$) and with any surface azimuth angle ($A_{ZS}$) can be calculated from the following equation [3]:

$$\cos\theta_{i} = \sin\delta\sin\varphi\cos\beta + \sin\delta\cos\varphi\sin\beta\cos A_{ZS} + \cos\delta\cos\varphi\cos\beta\cos\omega$$
$$-\cos\delta\sin\varphi\sin\beta\cos A_{SZ}\cos\omega - \cos\delta\sin\beta\sin A_{ZS}\sin\omega$$

## 3.5 Finding the torque to rotate

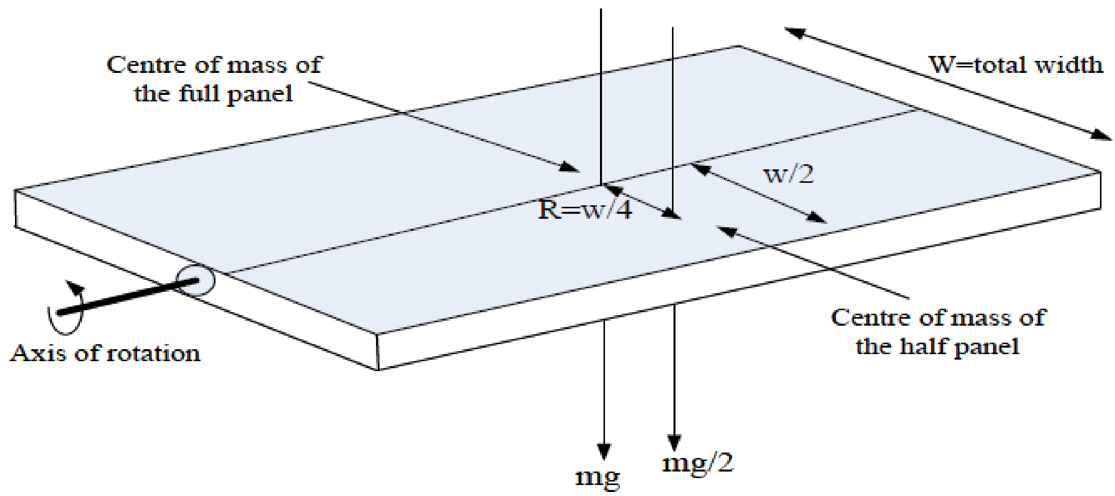



Figure 3.5: Finding the torque needed to rotate.

Mass of the panel, m = 0.3 kg

$$g = 9.8\ ms^{-2}$$

R = w/4 = 27.7 cm/4 = 6.925cm = 0.06925m

We know,

Torque, $\zeta = fR$

$= \frac{mgR}{2}$

$= (0.3\times 9.8 \times 0.06925)/2$

$= 0.1017975$ Nm

Now, we need find a suitable stepper motor to supply this amount of power. Therefore we need to calculate the power needed to supply this torque.

Let's consider a motor of $RPM = 60$

Power, $P = (\zeta\times 2\pi\times RPM)/60$ watt

$= (0.1017975\times 2\pi\times 60)/60$ watt

## 3.5 Calculation of Sunrise and Sunset Time

Microcontroller will take reading of the local time and date from a digital clock and check for the sunrise and sunset times. Sunrise time in actual solar time (AST) can be calculated from the sunrise angle $-\omega_s$ using the followings [7]:

$$AST = 720\text{ mins } -\omega_s\times(4\text{ mins/deg}) \qquad \ldots\ldots\ldots\ldots\ldots\ldots(1)$$

Sunrise time in AST can be converted in local standard time (LST) using the following relation [8],

$$LST = AST - (4\ mins)(LSTM - longitude) - ET \qquad \ldots\ldots\ldots\ldots\ldots\ldots(2)$$

**Local Standard Time Meridian (LSTM)**

The Local Standard Time Meridian (LSTM) is a reference meridian used for a particular time zone and is similar to the Prime Meridian, which is used for Greenwich Mean Time. The LSTM is illustrated below:

The (LSTM) is calculated according to the equation,

$$LSTM = 15°\left(\frac{longitude}{15}\right)_{\text{round to interger}} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(3)$$

**Equation of Time (ET)**

The equation of time (ET) (in minutes) is an empirical equation that corrects for the eccentricity of the Earth's orbit and the Earth's axial tilt.

$$ET = 9.87\sin(2D) - 7.53\cos(D) - 1.5\sin(D) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(4)$$

Where,

$$D = 360 \times \frac{n-80}{365}.degree \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(5)$$

Where n denotes the nth day of the year

Similarly, the LST of the sunset can be calculated with $\omega_s$ being the sunset angle.

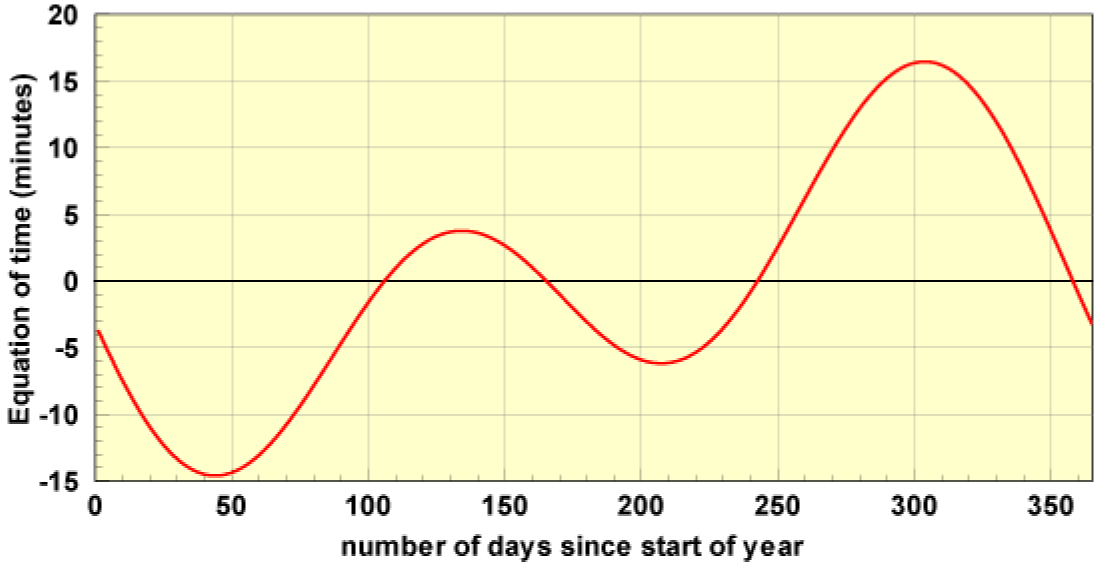



Figure 3.6: The time correction ET.

**Time Correction Factor (TC)**

The net Time Correction Factor (in minutes) accounts for the variation of the Local Solar Time (LST) within a given time zone due to the longitude variations within the time zone and also incorporates the ET above [2].

$$TC = 4(LSTM - longitude) + ET$$

The factor of 4 minutes comes from the fact that the Earth rotates 1° every 4 minutes.

**Local Solar Time (LST)**

The Local Solar Time (LST) can be found by using the previous two corrections to adjust the local time (LT) [2].

$$LST = LT + \frac{TC}{60}$$

# CHAPTER 4

## Hardware Implementation

### 4.1 Introduction

This chapter is dedicated to reflect how we used our research and design to build our prototype of the sun tracking system. After an intensive research on all the available techniques and components, to meet the requirement of the desired sun tracking system and to satisfy the prime objective of the project, suitable methods and components were selected and implemented through electrical circuits. In our system we have a control circuit controlled by the microcontroller in Arduino. The microcontroller sends signal to the polulu mini maestro calculating and comparing time with the help of RTC and the motor rotates the panel.

### 4.2 Mechanical Construction

While doing the mechanical construction a iron bar was connected vertically with the heavy weight iron base through gas welding, the stack that contains the solar panels is made of stainless steel (SS) bar and it is connected to the vertical bar through ball bearing so that the stack of solar panel can easily rotate along a horizontal arc throughout the day time with the operation of the linear actuator.

For connecting the solar panels with the SS bar stack, 3 holes have been made on the SS bar by drilling it and 3 iron rods were mounted on the SS bar through those holes with small ball bearing so that the iron rods can rotate easily around the horizontal axis. The iron rods are mounted on the main stack in such manner that they are bit tilted downwards and each of the rods contains two rectangular sized long iron plate in transverse manner so that the flexible solar panel can easily be attached to these iron rods. The end of the iron rod that remains on the other side of the ball bearing is connected with servo motor through the servo horns for a better grip on them for rotating through the whole day. Theoretically it has been proposed that the distance between two consecutive holes containing iron rods is 1.5 times of the width of the solar panel.

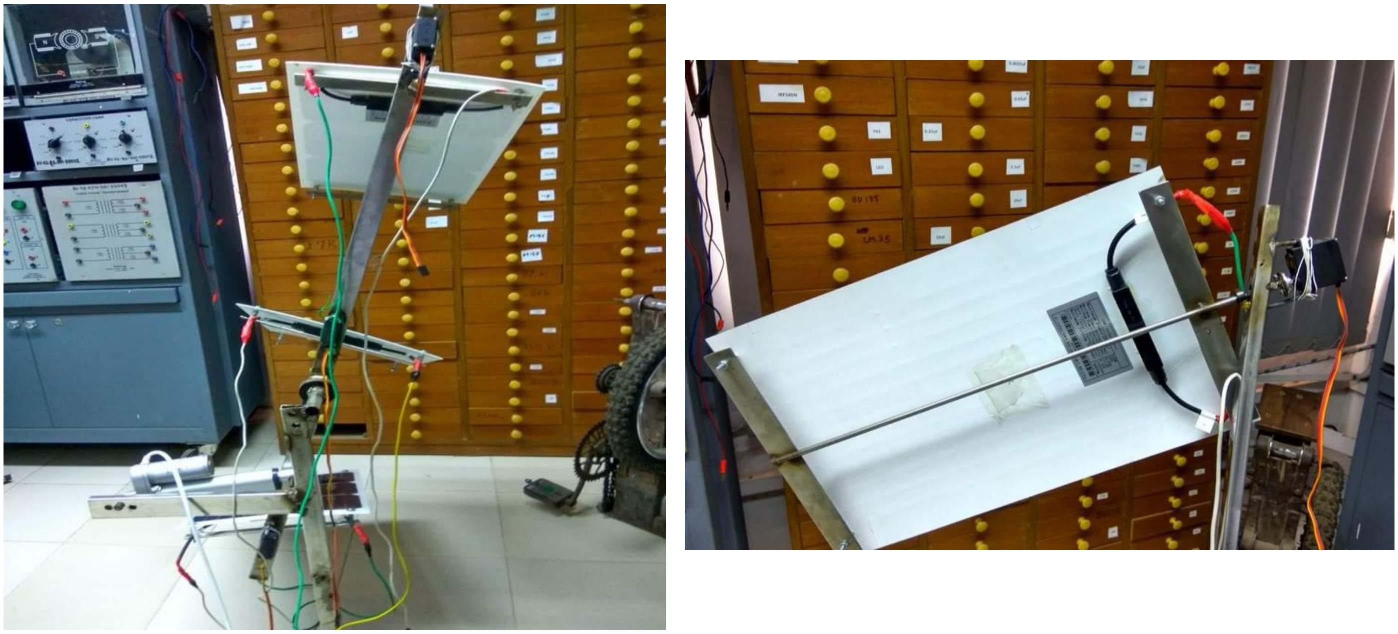


Figure 4.1: Mechanical alignment of our system.

## 4.3 Control Circuit Diagram

In our proposed system we have implemented our circuit by using one Arduino Mega, three MG995 Servo Motors, two Real Time Clock, one 12V Actuator, a 2 Relay Module , one 3V Lithium battery, one Pololu Mini Maestro 12 and some male to male and female to male connectors.

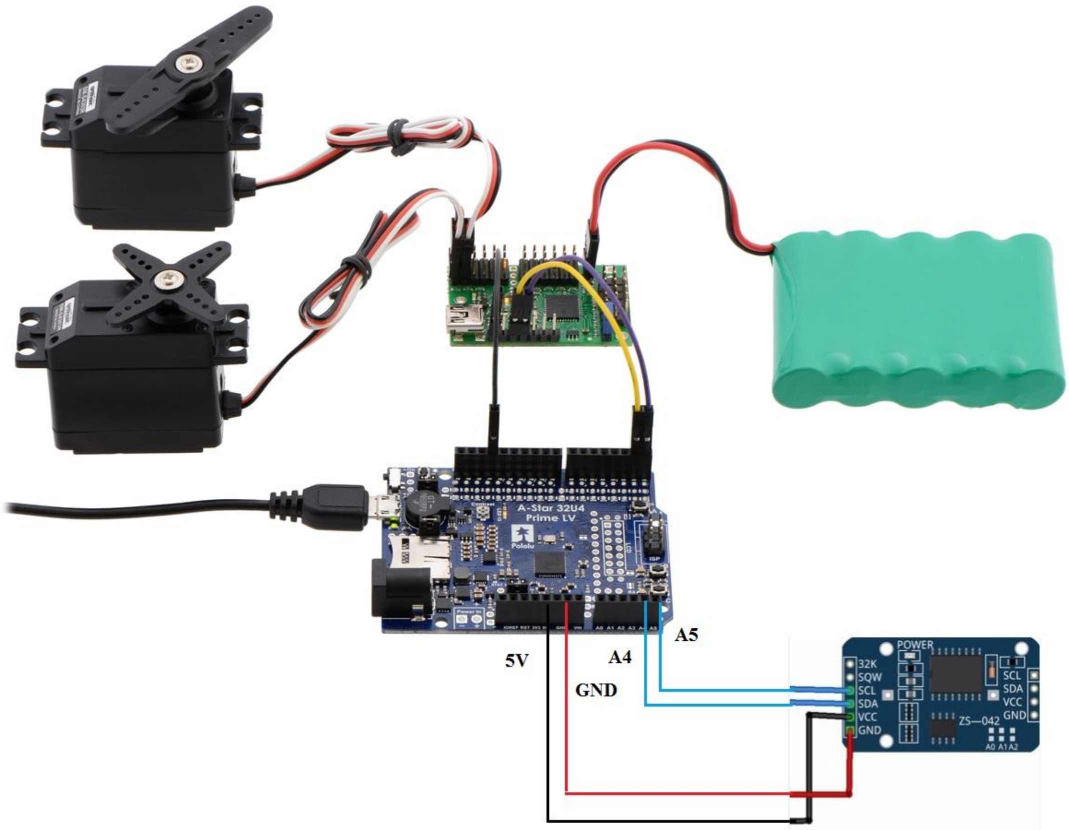



Figure 4.2: The control circuit diagram.

Arduino is connected with the PC or laptop via an USB cable. The Arduino gives supply to all the other components used in our circuitry design except the relays and the actuator. Arduino has a fixed 5V power supply which is supplied to the components. The pololu mini maestro12 is connected to Arduino in pin number 10 and 11 via its RxD and TxD pins respectively for transmission and reception of the signals. The real time clock is connected to the Arduino in pin SCL and ground through its pin number 5 and 6. The real time clock gets its power supply from a 3V lithium power supply.

Due to interference between the electric field and the magnetic field created by the actuator the circuitry of the actuator is fully separated from the above circuit. The result of this interference causes a huge problem in the whole system. Firstly, the real time clock gets reset; it has a very small power rating so it cannot tolerate the huge power of the actuator which is 12V so it gets reset. Secondly, the other components in the circuit with low power might get burn due to huge power supply. So the power is provided to the actuator externally via a battery.

## 4.4 States of the sun tracking solar panels throughout the daytime

The sun tracking process of this multilevel solar panel system works with the joint actions of servo motors rotating the solar panels and the actuator that moves the iron bar containing the panels in an horizontal arc. In this system 12 positions for the panels have been programmed for following the coordinates of the sun throughout the whole day as the microcontroller reads the time from the Real Time Clock and sends signals to the servo motor driver and to the relay of the actuator so that the panels can be aligned properly facing the sun with certain changes of angular position after predetermined regular time gap. These positions are distinguished from one another by the positive angles created by the panels with the vertical line due to their rotation and these positions have been defined as the states of the sun tracking solar panels in our project report.

The states have been described below:

**1. <u>State '1'</u>:** This state begins at 6:00 am in the morning when all the servo motors start working as they take the panels at an angular distance of +7.5° from the vertical line by rotating their shaft connected with the panels. At the beginning the linear displacement of the actuator shaft is kept 0mm as the iron stack carrying the solar panels remain tilted towards the east till 12pm.

**2. <u>State '2'</u>:** This state begins at 7:00am when the angular distances of the panels from the vertical line goes up to +22.5°.

**3. <u>State '3'</u>:** This state begins at 8:00am when the angular distances of the panels from the vertical line goes up to +37.5°.

4. **State '4':** This state begins at 9:00am when the angular distances of the panels from the vertical line goes up to +52.5°.

5. **State '5':** This state begins at 10:00 am when the angular distances of the panels from the vertical line go up to +67.5°.

6. **State '6':** This state begins at 11:00 am when the angular distances of the top and bottom panel increases up to +82.5° but the middle panel is taken to a position of -180 ° angular displacement with the vertical line and this position is maintained up to 12:00pm for avoiding it's shading over the bottom panel and again this middle panel cannot convert solar energy efficiently during this period due to the shading of top panel over it as the sun radiates energy from the angular position of approximately 90° with the horizon during this time span.

7. **State '7':** This state begins at 12:00 pm when the 10mm linear movement of actuator shaft tilts the iron stack carrying the solar panels towards the west. In this state, the angular displacement of top and bottom panels with the vertical line reaches +97.5° and the middle panel goes through a rotation of 180 ° comparing to its previous position for avoiding further shading.

8. **State'8':** This state begins at 1:00pm when all the panels get aligned at an angular displacement of +112.5° with the vertical line.

9. **State '9':** This state begins at 2:00pm when all the panels get aligned at an angular displacement of +127.5° with the vertical line.

10. **State'10':** This state begins at 3:00pm when all the panels get aligned at an angular displacement of +142.5° with the vertical line.

11. **<u>State'11'</u>:** This state begins at 4:00pm when all the panels get aligned at an angular displacement of +157.5° with the vertical line.

12. **<u>State'11'</u>:** This state begins at 5:00pm and continues till 6:00pm. In this state all the panels get aligned at an angular displacement of +172.5° with the vertical line. According to the seasonal daytime length of this country, this system will stop working at 6:00pm and it will again retain state '1' in the next morning.

**<u>Note</u>:** During each state the generalized angular displacement is +15°,which has been determined by the calculations of some parameters related to the motor driver for the convenience of the system.

## 4.5 Solar Panel Indoor Testing

We tested three panels indoor and provided the maximum illumination in order to calculate the short circuit current, open circuit voltage and maximum power for the three panels separately.

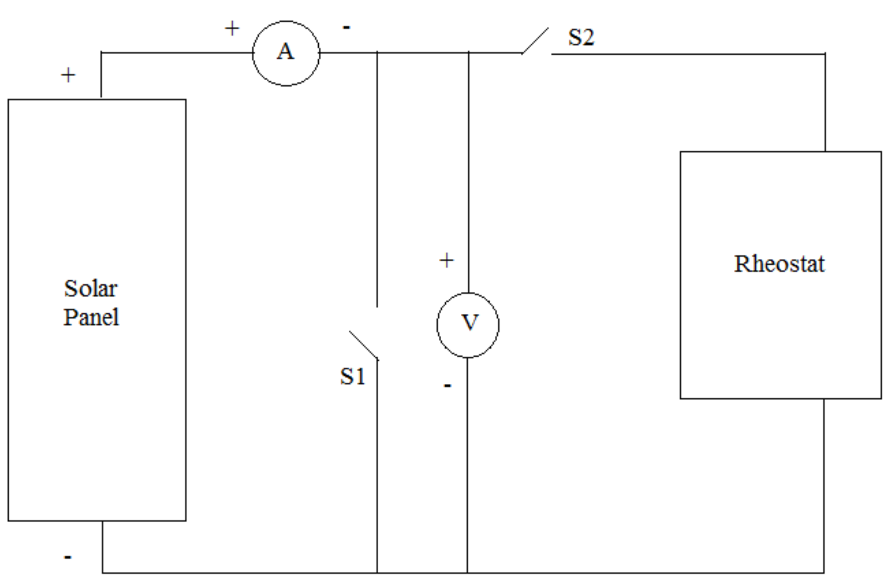



Figure 4.3: Connection diagram of Solar Panel for indoor testing.

We need to measure the maximum power point in a solar panel and the figure above shows how it was done. There are two switches S1 and S2. S1 is connected in parallel with voltmeter and S2 is connected in series with ammeter.

S1 is close and S2 is open to get the short circuit current, Isc.

S1 and S2 both are open to get the open circuit voltage, Voc.

S1 is open and S2 is close to get the current and terminal voltage of the solar panel.

The current is then varied by varying the value of the rheostat.

**For Panel 1:**

Open Circuit Voltage, Voc=20.1V

Short Circuit Current, Isc=210mA

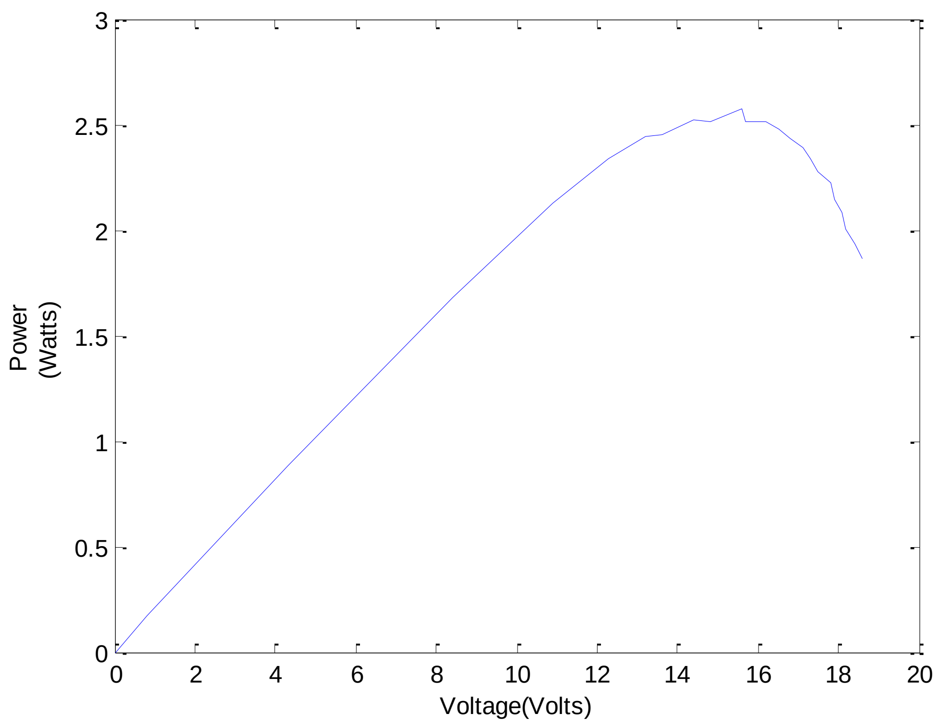



Figure 4.4: I-V curve for Panel 1 of indoor testing.

**For Panel2:**

Open Circuit Voltage, Voc=19.5V

Short Circuit Current, Isc=205mA

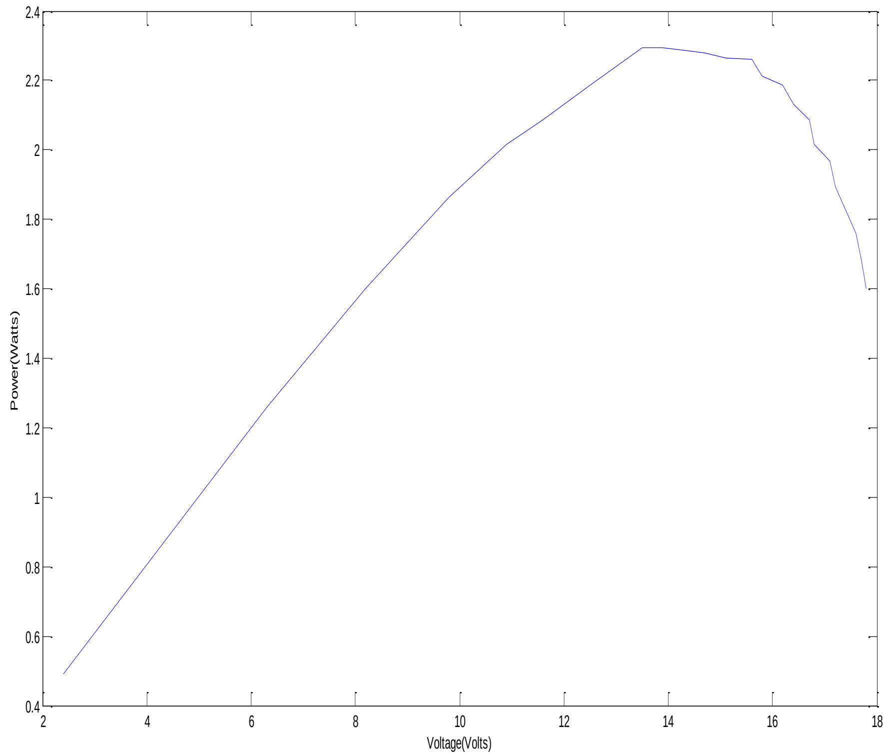



Figure 4.5: I-V curve for Panel 2 of indoor testing.

**For panel 3**: Open Circuit Voltage, Voc=20.5V

Short Circuit Current, Isc=205mA

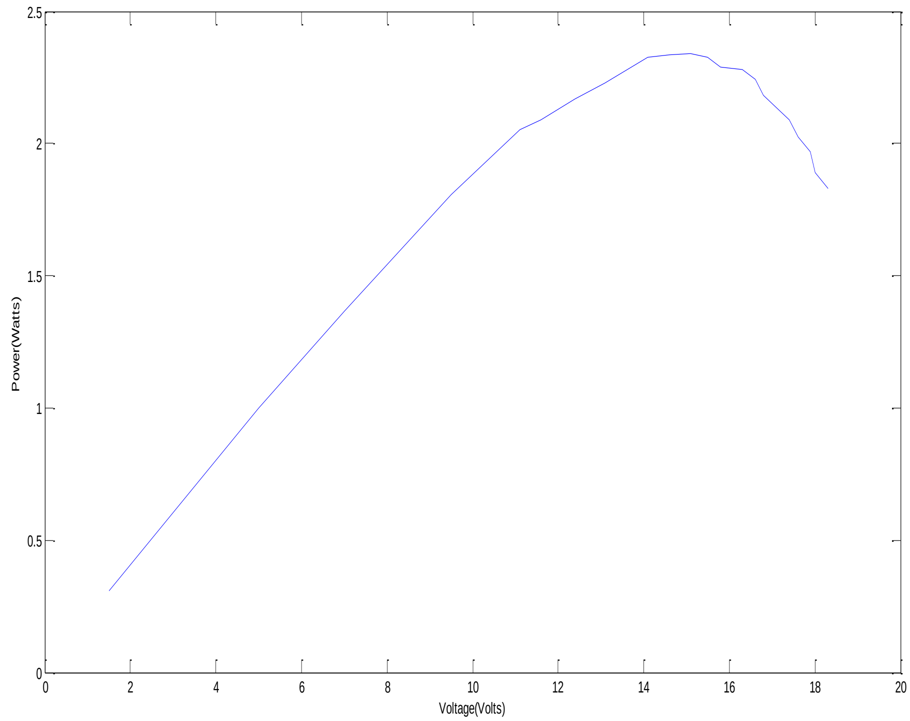



Figure 4.6: I-V curve for Panel 3 of indoor testing.

**For 20Watt Panel:** Open Circuit Voltage, Voc=20V

Short Circuit Current, Isc=280mA

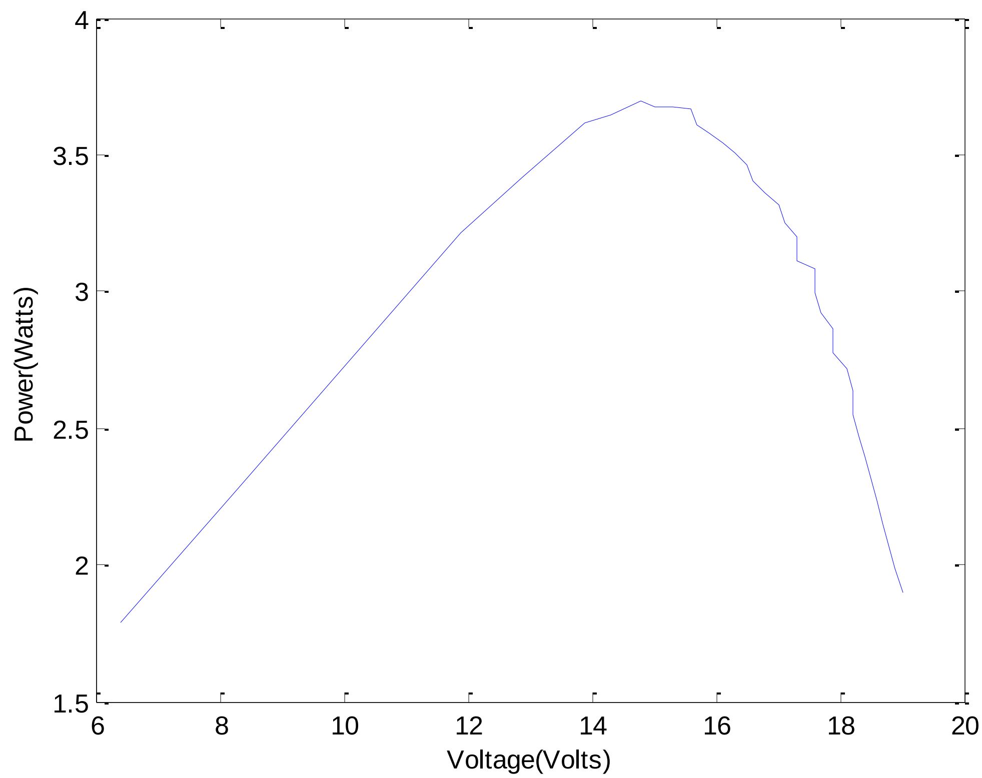



Figure 4.7: I-V curve for 20W Panel of indoor testing.

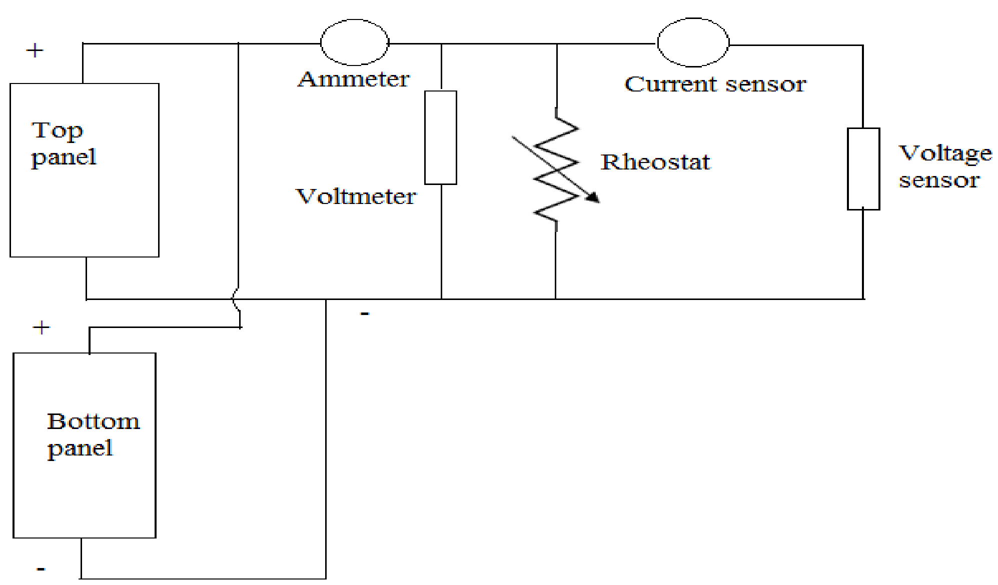



Figure 4.8: Arrangement of top and bottom panel in parallel with ammeter, voltmeter, rheostat, current sensor and voltage sensor.

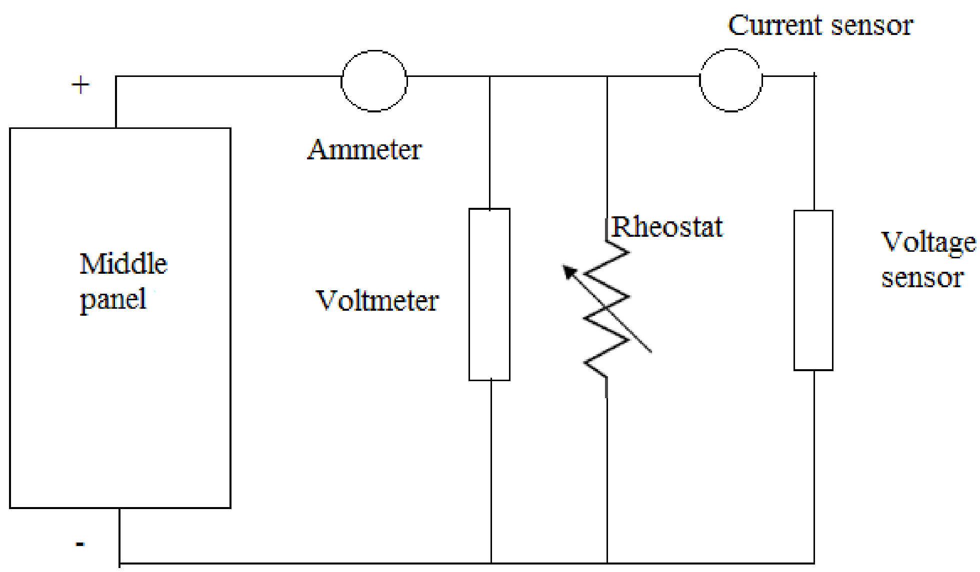



Figure 4.9: Arrangement of middle panel with ammeter, voltmeter, rheostat, current sensor and voltage sensor.

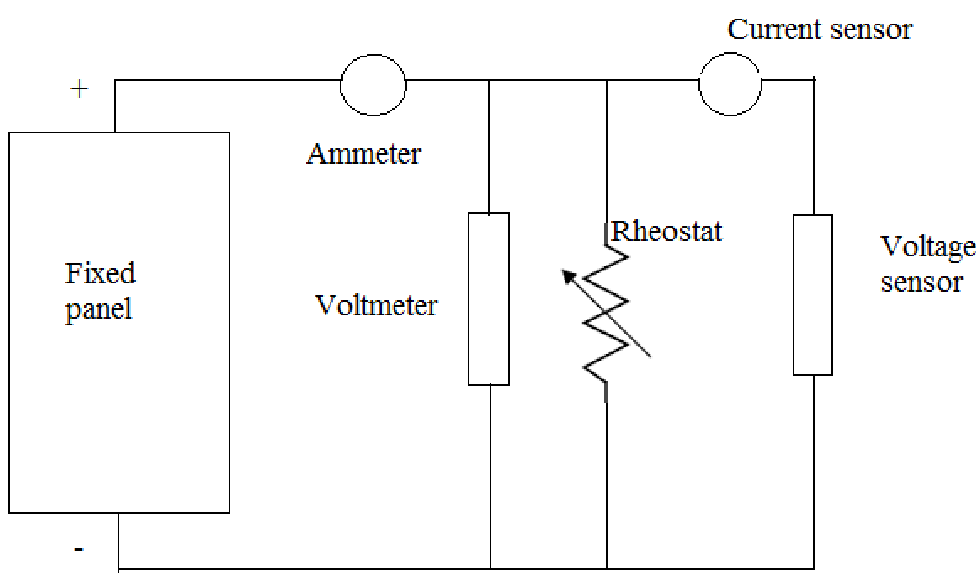



Figure 4.10: Arrangement of fixed panel with ammeter, voltmeter, rheostat, current sensor and voltage sensor.

Three panels-top, middle and bottom will be rotating. Since the top and bottom panel rotates in similar fashion, they were connected in parallel. There was a fixed panel with which we can compare the efficiency with respect to the rotating panels.

## 4.6 Data logger

A data logger is an electronic device that records data over time or in relation to location either with a built in instruments or sensors or via external instruments and sensors. Increasingly, but not entirely, they are based on a digital processor (or computer). In our project we used both current sensors and voltage sensors as parts of our data logger to record the currents and voltages generated by our multilevel automated solar panel system.

**System Description of the data logger:**

The figure below shows the system overview of the data logger, where we can see that the top and bottom solar panels are connected parallelly to the same current and voltage sensors as they remain in the same alignment throughout the day for tracking the sun. Again the middle panel is connected to the second pair of voltage and current sensors and the third pair of voltage and current sensors is used for reading a standard fixed panel. The microcontroller reads the data from the voltage and current sensors through its analog input pins and align them with the receiving time taken as input from the Real Time Clock. Then the microcontroller records the data in a memory card which has been connected with it through the digital output pins.

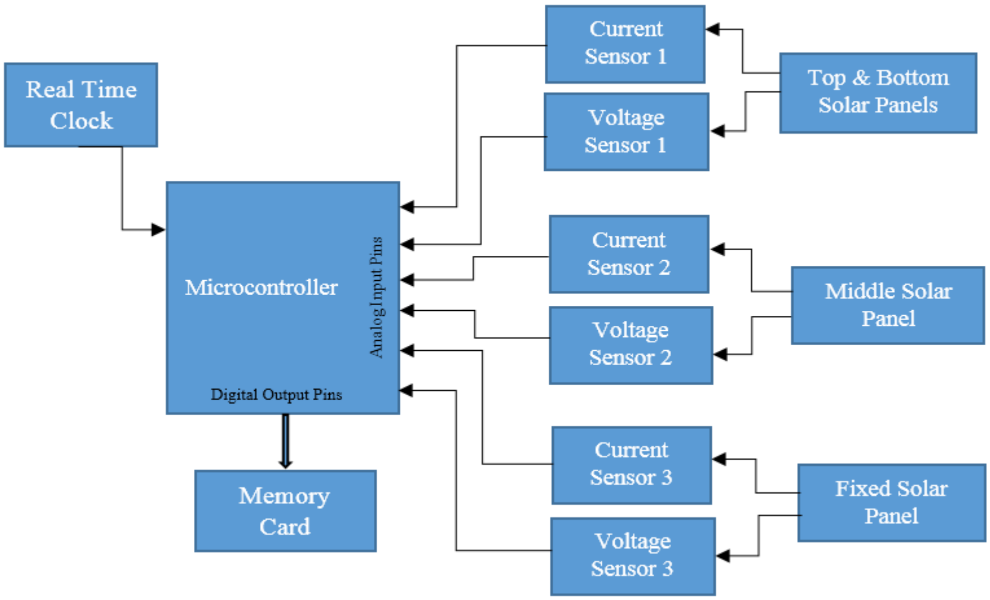



Figure 4.11: Block Diagram of Data Logger.

### 4.6.1 Current Sensor

A current sensor is a device that detects electric current (AC or DC) in a wire, and generates a signal proportional to it. The generated signal could be analog voltage or current or even digital output.

In this data logger "ACS714 Current Sensor" has been used for sensing the changing current which is a Hall effect-based linear current sensor that operates at 5V and has **an output sensitivity of 185 mV per Ampere** .The sensor requires a supply voltage of 4.5 V to 5.5 V to be connected across the Vcc and GND pins in the Figure 1. The ACS714 outputs an analog voltage which is linearly proportional to the input current. When Vcc is 5 V, this output voltage is centered at 2.5 V and changes by 185 mV per amp of input current, with positive current increasing the output voltage and negative current decreasing the output voltage [1].

A **Hall effect sensor** is a transducer that varies its output voltage in response to a magnetic field. Hall effect sensors are used for proximity switching, positioning, speed detection, and current sensing applications. In its simplest form, the sensor operates as an analog transducer, directly returning voltage.

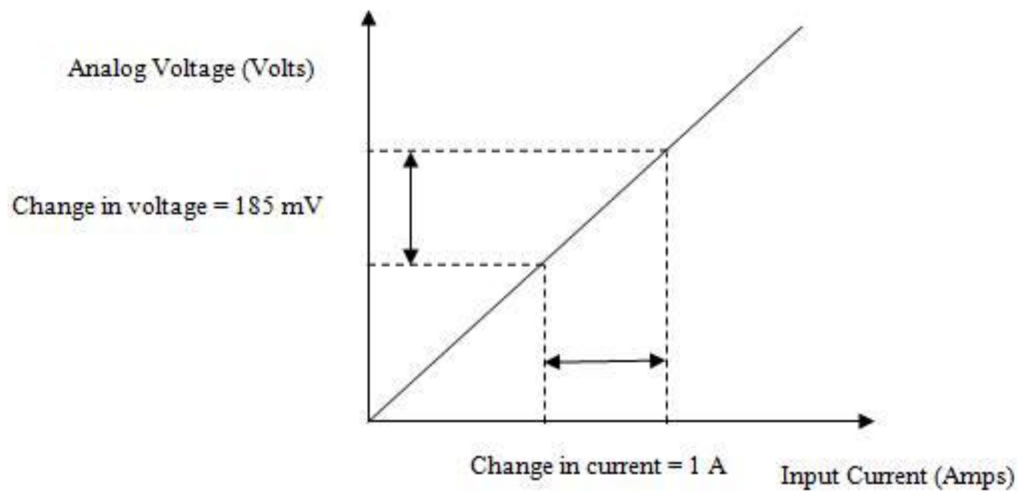



Figure 4.12: Relationship between analog voltage and input current (when input current is increasing).

The current sensor board features:

- Bidirectional measurement from -5 to 5 V (though the robust sensor IC can survive up to five times the over current condition).
- The sensor's conductive path internal resistance is typically 1.2 mΩ (just over 1/1000 of an ohm), and the PCB is made with 2-oz copper, so *very little* power is lost in the board.
- Use of a Hall effect sensor means the IC is able to electrically isolate the current path from the sensor’s electronics (up to 2.1 kV RMS). This means it can be used anywhere along the current path, and still offers good electrical isolation!
- 80 kHz bandwidth that can optionally be decreased by adding a capacitor across the board pins marked “filter”.
- High accuracy and reliability. With a typical total output error of 1.5% (at room temperature with factory calibration), an extremely stable output offset voltage, and almost zero magnetic hysteresis.
- Automotive grade operating temperature range of -40°C – 150°C [1].

**Note:** This current sensor should be used below 30V

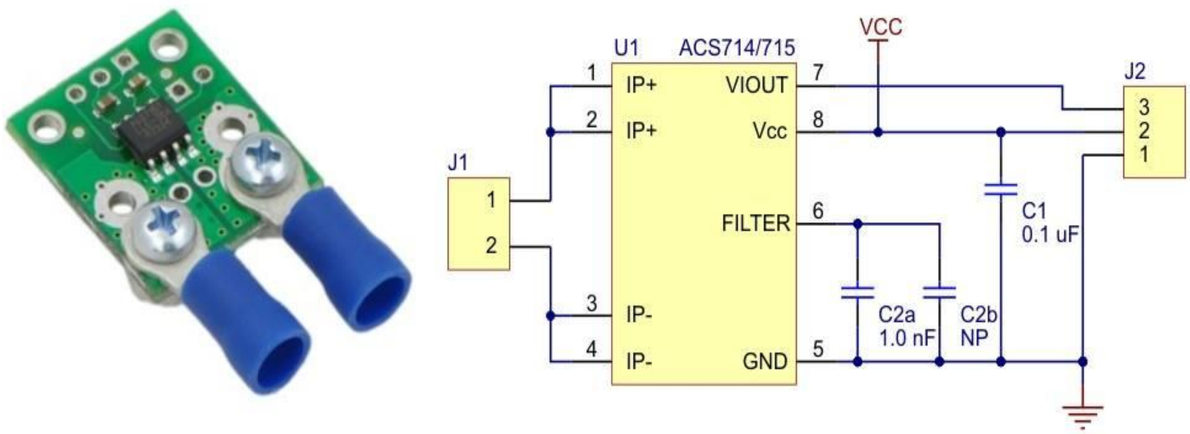



Figure 4.13: ACS 714 current sensor's pin diagram.

Vout pin from ACS 714 should be connected with Analog input (A0, A1, A2, A4 or A5). In Figure 3 Vout from ACS 714 is connected with Analog input A3 of Arduino Controller.

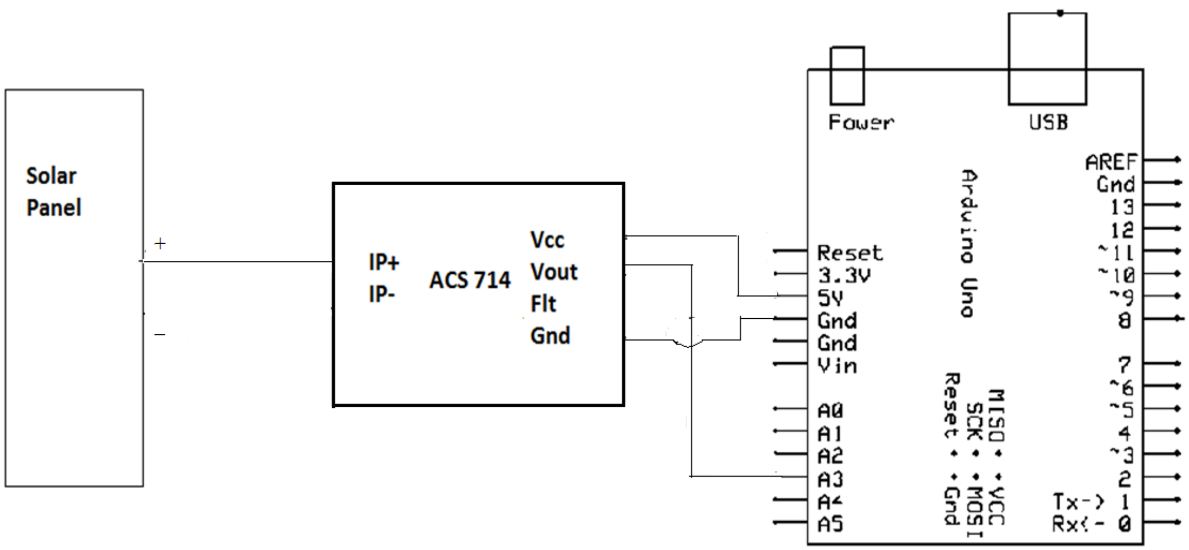



Figure 4.14: Connection of ACS 714 with Arduino Controller.

## 4.6.2 Voltage Sensor

A voltage sensor is electrical device that can read voltages in certain range and sends these values to the analog input pins of the system's micro controller. In this system, conventional voltage sensors of 25V, available in local markets has been used for sensing the continuous voltage change and record then through the analog pin in a memory. We used SEN-00101 as voltage sensor for our data logger.

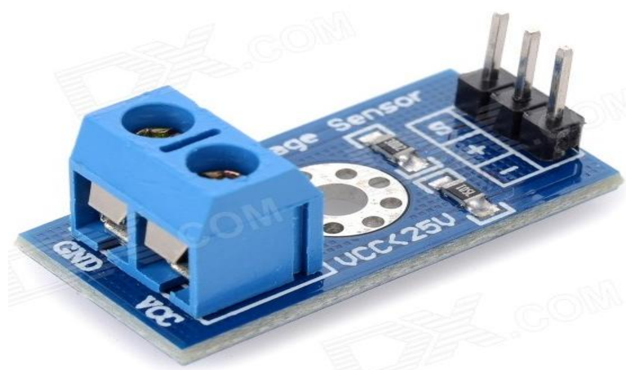



Figure 4.15: 25V voltage sensor.

The Vout pin from Voltage Sensor can be connected with Analog input (A0, A1, A2, A4 or A5). In Figure 5 Vout pin from Voltage Sensor has been connected with Analog input pin A1.

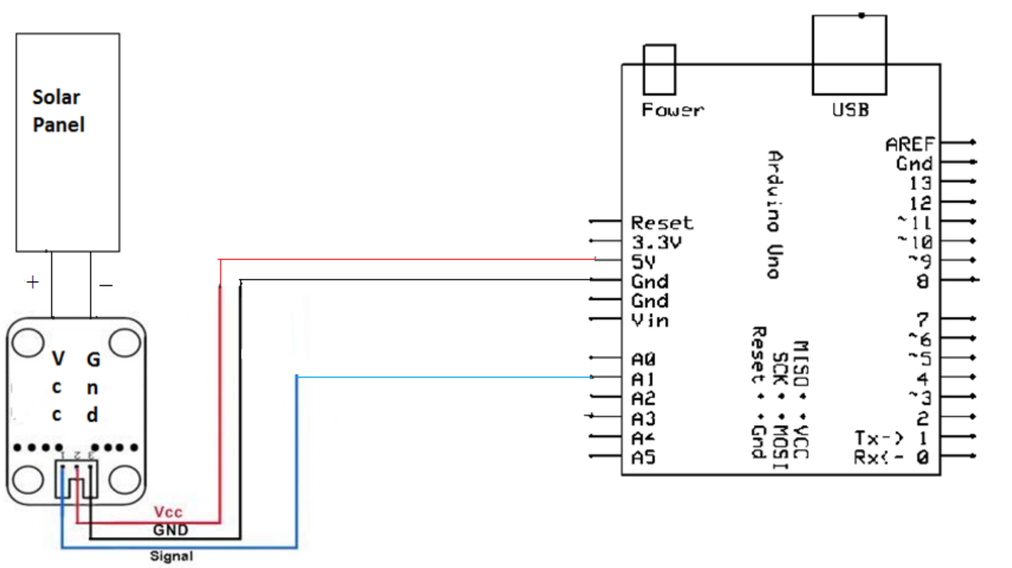



Figure 4.16: Connection between voltage sensor and Arduino Controller.

**Note:** We have used three current sensors and three voltage sensors. The top and bottom panel shares a common current sensor and voltage sensor. The other two current sensors and voltage sensors is for our middle panel and fixed panel.

**Equation for output reading of Voltage and Current sensors:**

I=0.028496*(v)-14.629

V=0.023647*(v)-1.0186

Here, v is the output parameter bits from arduino MCU ranging from 0 to 1023.

### 4.6.3 SD card slot for saving data

The card slot that has been used in our data logger is known as Arduino SD card slot, which has been shown in Figure 6.

Figure 4.17: Arduino SD card slot.

The communication between the microcontroller and the SD card uses Serial Peripheral Interface bus, which takes place on digital pins 11, 12, and 13 (on most Arduino boards) or 50, 51, and 52 (Arduino Mega). Additionally, another pin must be used to select the SD card. This can be the hardware SS pin - pin 10 (on most Arduino boards) or pin 53 (on the Mega) [2].

The connections have been shown in Figure 7, where we can see that Digital pins of MCU have been connected to the SDCS, MOSI, SCK and MISO pins of the SD card sold. The slot holds an 8GB micro SD card for saving our system's data.

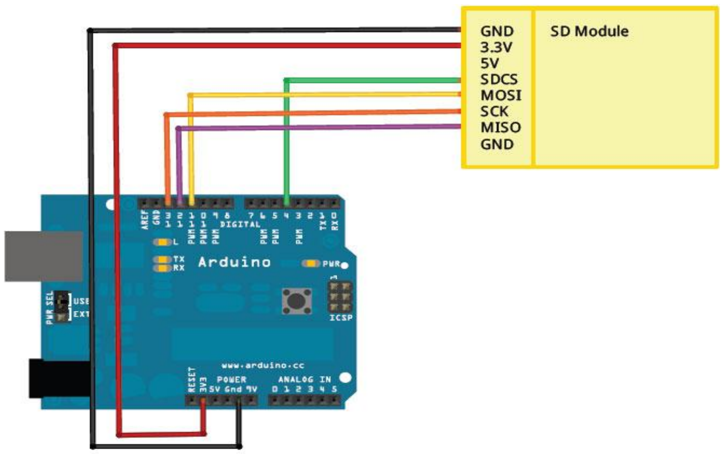



Figure 4.18: Connection between Arduino and SD card slot.

# CHAPTER 5

# Software Implementation

## 5.1 Software used for programming

In our proposed system we have used Arduino 1.6.4 version for programming. That is an open project that allows writing and debugging codes efficiently which is also immensely supported by vast library files. The library files assisted us in designing algorithms from a sketch books and example's directory. Verification and compilation of error is smooth and easily accessible. By opening Arduino Uno sketch book we can select specific board and upload programming codes.

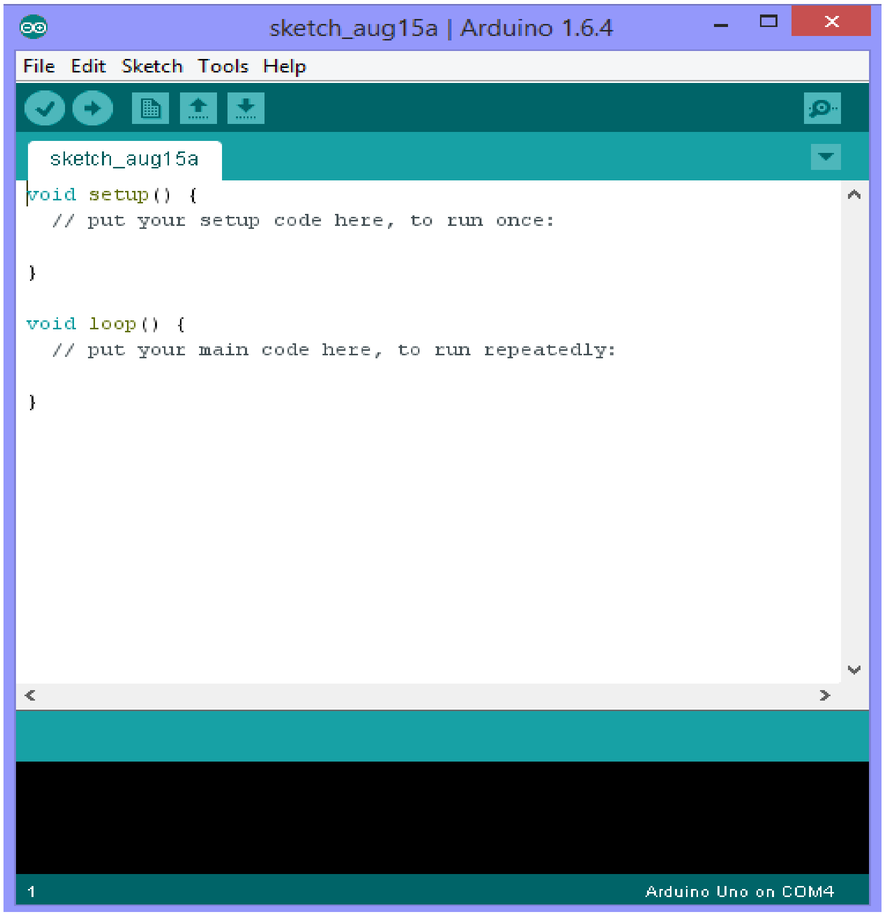



Figure 5.1: Empty sketch book of the Arduino 1.6.4 version.

## 5.2 Algorithm for coding

The flow chart in Figure 5.2 shows the algorithm used in the micro-controller programming to control the horizontal gliding and rotational motion of the panels to track the sun. The microcontroller reads the real time $t$ and $n$th day from a digital clock and determines the sunrise ($TSR$) and sunset ($TSS$) times using the set of Eq. (1)-(5) mentioned in section 3.5. It then calculates the time interval $\Delta T = (TSS - TSR)/12$ at which the panels should be rotated.

The panels are rotated at every instance $t = TSR + N.\Delta T$, where $N$ is number of instances the panels should be rotated and varies from 1 to 12.

When $t = T\varphi$, the time that relates to the maximum position of the sun in the sky until which lower panels get full sun exposure, the middle panel is turned in vertically upright position and kept disabled until $t = TSS - T\phi$ .

At $t = TSR + (TSS - TSR)/2$, i.e. at noon, the upper and lower panels are glided and realigned.
At $t = TSS$, the sunset time, the panels are reset to their original position and angle.

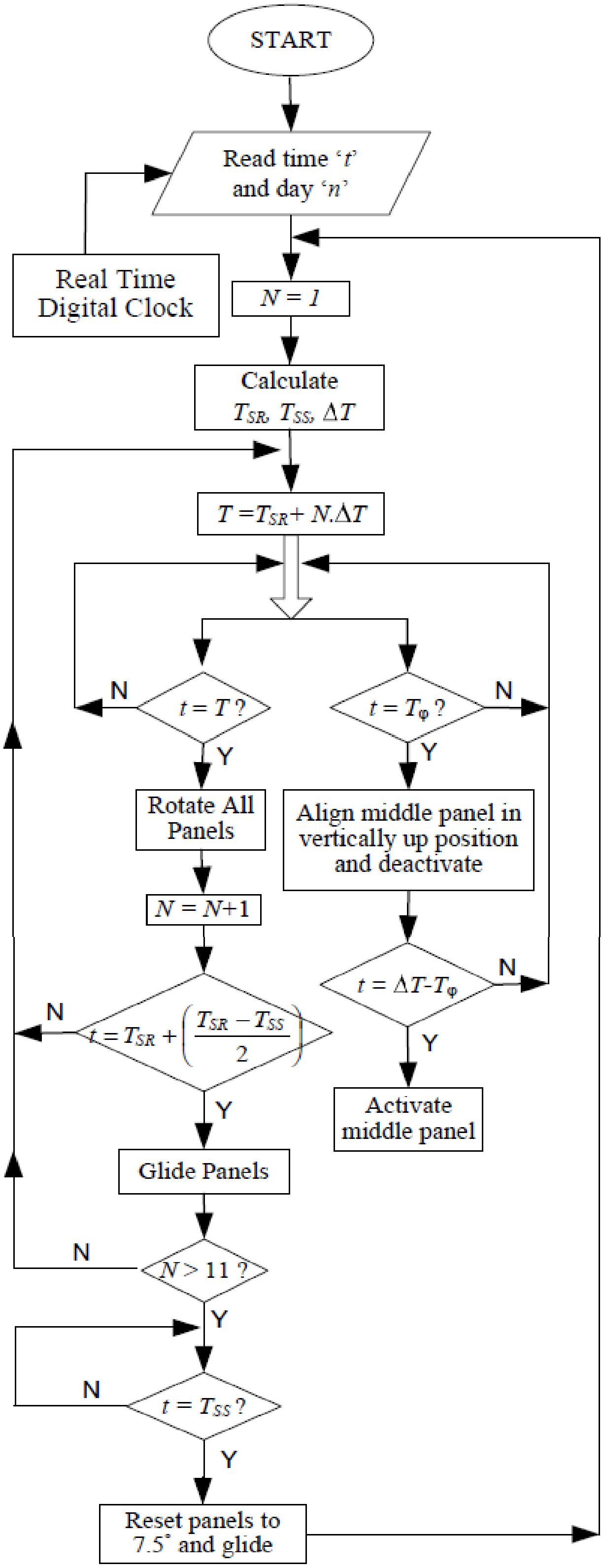



Figure 5.2: Flow chart of the algorithm used in the microcontroller programming to control the horizontal gliding and rotational motion of the panels to track the sun [1].

# CHAPTER 6

## Data Collection and Analysis

### 6.1 Introduction

After the mechanical construction of the multilevel sun tracking solar system, it was setup on a rooftop for data collection during this summer season. When we collected the data for several days of June, July and August, we could analyze the energy and power generation characteristics of the system during typical summer days.

### 6.2 Circuit Configuration of the System for Data Collection

For collecting data we connected the top and bottom panels parallelly as the rotational movement and alignment of these two panels remain similar throughout the day. For measuring currents and voltages we used conventional instruments like ammeter and voltmeter along with the current and voltage sensors of our data logger.

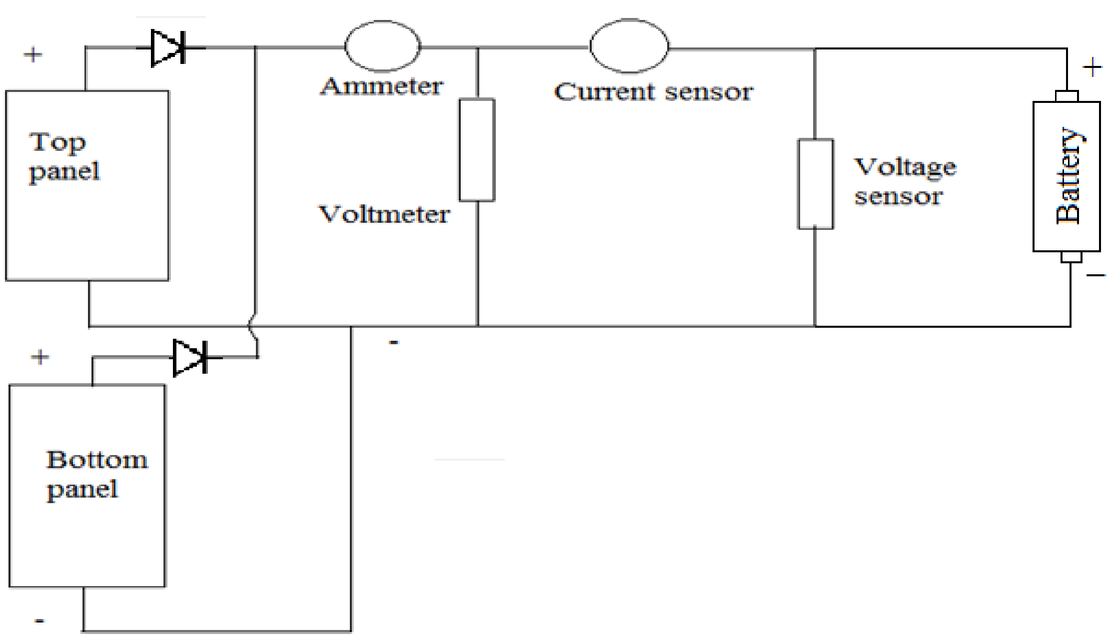



Figure 6.1: Connection diagram of top and bottom panel for data collection.

The middle panel has been connected with a single set of ammeter and voltmeter and a single pair of current and voltage sensors as it remains disabled through the midday. For comparing the system's data with the single level solar system, we used a fixed PV panel with same characteristics and size for generating same parameters.

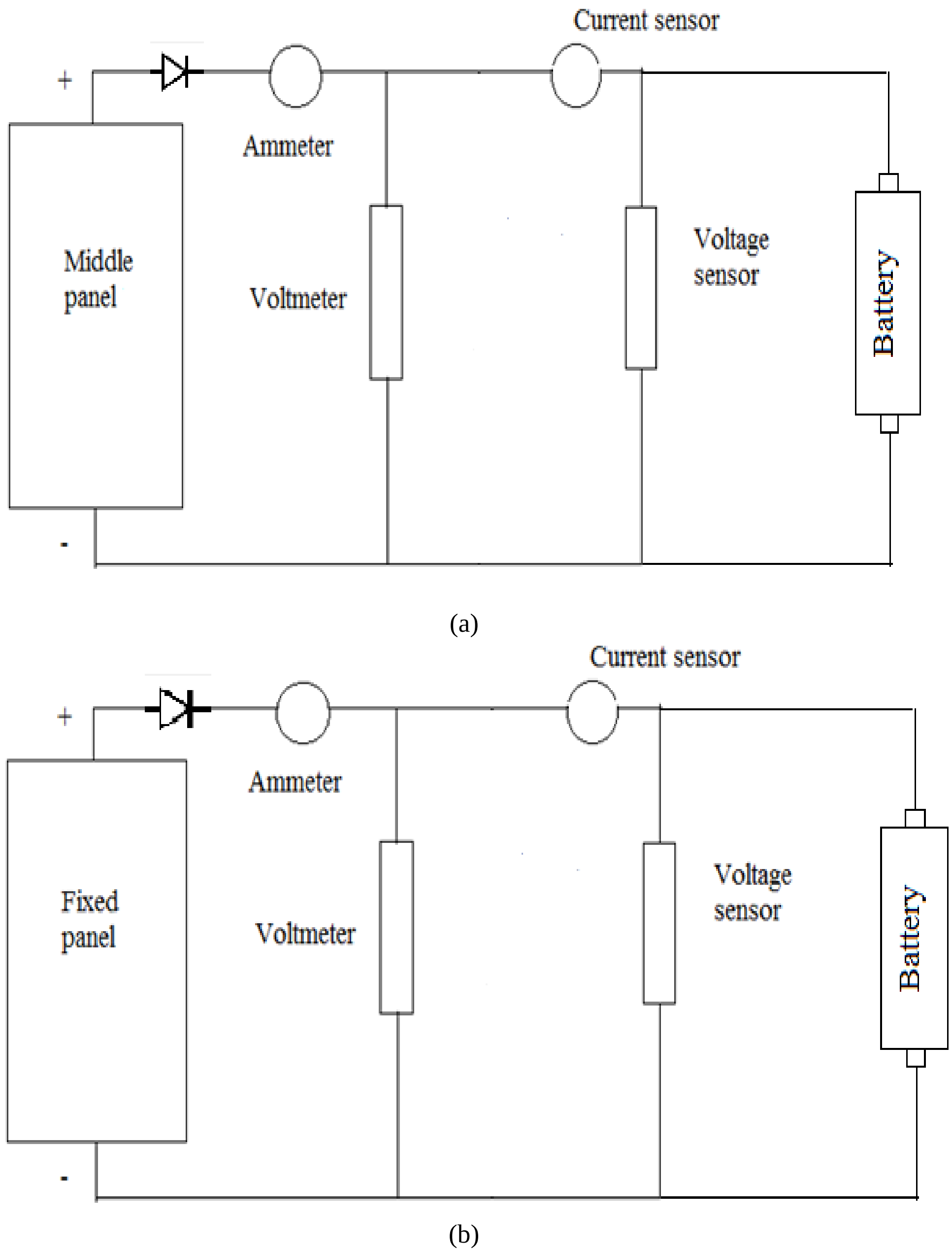



Figure 6.2: (a) Connection diagram of middle panel of multilevel sun tracking system

(b) Connection diagram of conventional fixed panel.

## 6.3 Data Analysis

### 6.3.1 Analysis of single day data

Data collected on July 14 and August 07 of 2016 have been used for analyzing the efficiency of our system. The whole day data of August 07 has been shown below when the temperature was 33° C and it was a moderate sunny day [9].

The parameters are named accordingly:

Top and Bottom Panel (Parallelly connected) Current= $I_1$

Top and Bottom Panel (Parallelly connected) Voltage= $V_1$

Middle Panel Current= $I_2$

Middle Panel Voltage= $V_2$

Voltage across the Battery connected to the multi-level sun tracking system= $V_4$

Power from top and bottom panel= $P_1$

Power from middle panel= $P_2$

Fixed Panel Current= $I_3$

Voltage across the Battery connected to the fixed panel= $V_3$

Fixed Panel Power= $P_3$

Data taken on August 07, 2016

| | $I_1$(A) | $V_1$(V) | $I_2$(A) | $V_2$(V) | $V_4$(V) | $P_1$(W) | $P_2$(W) | $I_3$(A) | $V_3$(V) | $P_3$(W) |
|---|---|---|---|---|---|---|---|---|---|---|
| 08.00am | 1.23 | 12.9 | 0.60 | 12.80 | 12.01 | 15.867 | 7.68 | 0.33 | 12.13 | 4.0029 |
| 08.15am | 1.36 | 12.7 | 0.62 | 12.72 | 12.04 | 17.272 | 7.8864 | 0.37 | 12.19 | 4.5103 |
| 08:30am | 1.43 | 12.7 | 0.63 | 12.65 | 12.07 | 18.161 | 7.9695 | 0.38 | 12.21 | 4.6398 |
| 08:45am | 1.47 | 12.8 | 0.64 | 12.64 | 12.09 | 18.816 | 8.0896 | 0.41 | 12.29 | 5.0389 |
| 09:00am | 1.52 | 12.4 | 0.73 | 12.51 | 12.12 | 18.848 | 9.1323 | 0.39 | 12.30 | 4.797 |
| 09:15am | 1.53 | 12.3 | 0.76 | 12.49 | 12.15 | 18.819 | 9.4924 | 0.40 | 12.33 | 4.932 |
| 09:30am | 1.51 | 12.1 | 0.72 | 12.53 | 12.18 | 18.271 | 9.0216 | 0.42 | 12.37 | 5.1954 |
| 09:45am | 1.49 | 12.2 | 0.74 | 12.51 | 12.22 | 18.178 | 9.2574 | 0.41 | 12.39 | 5.0799 |
| 10:00am | 1.42 | 12.0 | 0.71 | 12.54 | 12.25 | 17.04 | 8.9034 | 0.44 | 12.43 | 5.4692 |
| 10:15am | 1.37 | 12.2 | 0.67 | 12.64 | 12.27 | 16.714 | 8.4688 | 0.48 | 12.48 | 5.9904 |
| 10:30am | 1.39 | 12.3 | 0.71 | 12.61 | 12.29 | 17.097 | 8.9531 | 0.49 | 12.51 | 6.1299 |
| 10:45am | 1.40 | 12.5 | 0.69 | 12.63 | 12.32 | 17.5 | 8.7147 | 0.46 | 12.53 | 5.7638 |
| 11:00am | 1.42 | 12.4 | 0.20 | NT | 12.36 | 17.608 | 0 | 0.49 | 12.37 | 6.0613 |
| 11:15am | 1.37 | 12.6 | 0.20 | NT | 12.38 | 17.262 | 0 | 0.51 | 12.62 | 6.4362 |
| 11.30am | 1.39 | 12.5 | 0.20 | NT | 12.42 | 17.375 | 0 | 0.52 | 12.67 | 6.5884 |
| 11.45am | 1.38 | 12.3 | 0.20 | NT | 12.48 | 16.974 | 0 | 0.54 | 12.69 | 6.8526 |
| 12.00pm | 1.42 | 12.1 | 0.20 | NT | 12.53 | 17.182 | 0 | 0.57 | 12.71 | 7.2447 |
| 12.15pm | 1.43 | 12.0 | 0.20 | NT | 12.57 | 17.16 | 0 | 0.58 | 12.74 | 7.3892 |
| 12:30pm | 1.51 | 12.2 | 0.20 | NT | 12.60 | 18.422 | 0 | 0.61 | 12.78 | 7.7958 |
| 12:45pm | 1.54 | 12.1 | 0.20 | NT | 12.64 | 18.634 | 0 | 0.63 | 12.81 | 8.0703 |
| 01:00pm | 1.55 | 12.0 | 0.20 | NT | 12.69 | 18.6 | 0 | 0.68 | 12.86 | 8.7448 |
| 01:15pm | 1.54 | 12.3 | 0.79 | 12.59 | 12.73 | 18.942 | 9.9461 | 0.69 | 12.89 | 8.8941 |
| 01:30pm | 1.52 | 12.4 | 0.76 | 12.62 | 12.78 | 18.848 | 9.5912 | 0.71 | 12.92 | 9.1732 |
| 01:45pm | 1.47 | 12.7 | 0.72 | 12.65 | 12.83 | 18.669 | 9.108 | 0.68 | 12.94 | 8.7992 |
| 02:00pm | 1.48 | 12.6 | 0.73 | 12.63 | 12.87 | 18.648 | 9.2199 | 0.69 | 12.97 | 8.9493 |
| 02:15pm | 1.43 | 12.8 | 0.76 | 12.64 | 12.89 | 18.304 | 9.6064 | 0.65 | 13.02 | 8.463 |
| 02:30pm | 1.41 | 12.9 | 0.71 | 12.69 | 12.93 | 18.189 | 9.0099 | 0.63 | 13.08 | 8.2404 |
| 02:45pm | 1.40 | 13.1 | 0.72 | 12.67 | 12.97 | 18.34 | 9.1224 | 0.61 | 13.12 | 8.0032 |
| 03:00pm | 1.38 | 13.31 | 0.69 | 12.70 | 13.03 | 18.3678 | 8.763 | 0.58 | 13.16 | 7.6328 |
| 03:15pm | 1.35 | 13.33 | 0.68 | 12.72 | 13.08 | 17.9955 | 8.6496 | 0.57 | 13.21 | 7.5297 |
| 03:30pm | 1.33 | 13.35 | 0.67 | 12.75 | 13.12 | 17.7555 | 8.5425 | 0.53 | 13.26 | 7.0278 |
| 03:45pm | 1.29 | 13.37 | 0.64 | 12.79 | 13.17 | 17.2473 | 8.1856 | 0.49 | 13.31 | 6.5219 |
| 04:00pm | 1.26 | 13.40 | 0.62 | 12.80 | 13.22 | 16.884 | 7.936 | 0.48 | 13.34 | 6.4032 |
| 04:15pm | 1.19 | 13.41 | 0.59 | 12.83 | 13.26 | 15.9579 | 7.5697 | 0.44 | 13.39 | 5.8916 |
| 04:30pm | 1.11 | 13.43 | 0.58 | 12.86 | 13.29 | 14.9073 | 7.4588 | 0.41 | 13.40 | 5.494 |
| 04:45pm | 1.04 | 13.47 | 0.54 | 12.91 | 13.34 | 14.0088 | 6.9714 | 0.39 | 13.43 | 5.2377 |
| 05:00pm | 0.96 | 13.48 | 0.54 | 12.96 | 13.39 | 12.9408 | 6.9984 | 0.32 | 13.44 | 4.3008 |
| 05:15pm | 0.91 | 13.51 | 0.51 | 13.05 | 13.42 | 12.2941 | 6.6555 | 0.31 | 13.46 | 4.1726 |
| 05:30pm | 0.86 | 13.53 | 0.51 | 13.05 | 13.47 | 11.6358 | 6.6555 | 0.29 | 13.47 | 3.9063 |
| 05:45pm | 0.86 | 13.53 | 0.47 | 13.09 | 13.52 | 11.6358 | 6.1523 | 0.28 | 13.48 | 3.7744 |
| 06:00pm | 0.82 | 13.59 | 0.42 | 13.10 | 13. 54 | 11.1438 | 5.502 | 0.26 | 13.49 | 3.5074 |

**NT**: When the middle panel was inactive, we did not measure the voltage across it.

Now, the analysis about the difference of energy generation between the multilevel sun tracker and the conventional fixed panel has been stated below.

Let,

Energy produced by parallelly connected top and bottom panel= $E_1$

Energy produced by middle panel= $E_2$

Time interval= 15 minutes = 900 seconds

| Time | $E_1$(KJ) | $E_2$(KJ) | $E_s$=$E_1$+$E_2$(KJ) |
|---|---|---|---|
| 08.00am | 14.2803 | 6.912 | 21.1923 |
| 08.15am | 15.5448 | 7.09776 | 22.64256 |
| 08:30am | 16.3449 | 7.17255 | 23.51745 |
| 08:45am | 16.9344 | 7.28064 | 24.21504 |
| 09:00am | 16.9632 | 8.21907 | 25.18227 |
| 09:15am | 16.9371 | 8.54316 | 25.48026 |
| 09:30am | 16.4439 | 8.11944 | 24.56334 |
| 09:45am | 16.3602 | 8.33166 | 24.69186 |
| 10:00am | 15.336 | 8.01306 | 23.34906 |
| 10:15am | 15.0426 | 7.62192 | 22.66452 |
| 10:30am | 15.3873 | 8.05779 | 23.44509 |
| 10:45am | 15.75 | 7.84323 | 23.59323 |
| 11:00am | 15.8472 | 0 | 15.8472 |
| 11:15am | 15.5358 | 0 | 15.5358 |
| 11.30am | 15.6375 | 0 | 15.6375 |
| 11.45am | 15.2766 | 0 | 15.2766 |
| 12.00pm | 15.4638 | 0 | 15.4638 |
| 12.15pm | 15.444 | 0 | 15.444 |
| 12:30pm | 16.5798 | 0 | 16.5798 |
| 12:45pm | 16.7706 | 0 | 16.7706 |
| 01:00pm | 16.74 | 0 | 16.74 |
| 01:15pm | 17.0478 | 8.95149 | 25.99929 |
| 01:30pm | 16.9632 | 8.63208 | 25.59528 |
| 01:45pm | 16.8021 | 8.1972 | 24.9993 |
| 02:00pm | 16.7832 | 8.29791 | 25.08111 |
| 02:15pm | 16.4736 | 8.64576 | 25.11936 |
| 02:30pm | 16.3701 | 8.10891 | 24.47901 |
| 02:45pm | 16.506 | 8.21016 | 24.71616 |
| 03:00pm | 16.53102 | 7.8867 | 24.41772 |

| | | | |
|---|---|---|---|
| 03:15pm | 16.19595 | 7.78464 | 23.98059 |
| 03:30pm | 15.97995 | 7.68825 | 23.6682 |
| 03:45pm | 15.52257 | 7.36704 | 22.88961 |
| 04:00pm | 15.1956 | 7.1424 | 22.338 |
| 04:15pm | 14.36211 | 6.81273 | 21.17484 |
| 04:30pm | 13.41657 | 6.71292 | 20.12949 |
| 04:45pm | 12.60792 | 6.27426 | 18.88218 |
| 05:00pm | 11.64672 | 6.29856 | 17.94528 |
| 05:15pm | 11.06469 | 5.98995 | 17.05464 |
| 05:30pm | 10.47222 | 5.98995 | 16.46217 |
| 05:45pm | 10.47222 | 5.53707 | 16.00929 |
| 06:00pm | 10.02942 | 4.9518 | 14.98122 |
| | | Total Energy | 863.755 KJ |

Let,

Energy produced by single fixed panel= $E_3$

| Time | $E_3$(KJ) | $3\times E_3$(KJ) |
|---|---|---|
| 08.00am | 3.60261 | 10.80783 |
| 08.15am | 4.05927 | 12.17781 |
| 08:30am | 4.17582 | 12.52746 |
| 08:45am | 4.53501 | 13.60503 |
| 09:00am | 4.3173 | 12.9519 |
| 09:15am | 4.4388 | 13.3164 |
| 09:30am | 4.67586 | 14.02758 |
| 09:45am | 4.57191 | 13.71573 |
| 10:00am | 4.92228 | 14.76684 |
| 10:15am | 5.39136 | 16.17408 |
| 10:30am | 5.51691 | 16.55073 |
| 10:45am | 5.18742 | 15.56226 |
| 11:00am | 5.45517 | 16.36551 |
| 11:15am | 5.79258 | 17.37774 |
| 11.30am | 5.92956 | 17.78868 |
| 11.45am | 6.16734 | 18.50202 |
| 12.00pm | 6.52023 | 19.56069 |
| 12.15pm | 6.65028 | 19.95084 |
| 12:30pm | 7.01622 | 21.04866 |

| | | |
|---|---|---|
| 12:45pm | 7.26327 | 21.78981 |
| 01:00pm | 7.87032 | 23.61096 |
| 01:15pm | 8.00469 | 24.01407 |
| 01:30pm | 8.25588 | 24.76764 |
| 01:45pm | 7.91928 | 23.75784 |
| 02:00pm | 8.05437 | 24.16311 |
| 02:15pm | 7.6167 | 22.8501 |
| 02:30pm | 7.41636 | 22.24908 |
| 02:45pm | 7.20288 | 21.60864 |
| 03:00pm | 6.86952 | 20.60856 |
| 03:15pm | 6.77673 | 20.33019 |
| 03:30pm | 6.32502 | 18.97506 |
| 03:45pm | 5.86971 | 17.60913 |
| 04:00pm | 5.76288 | 17.28864 |
| 04:15pm | 5.30244 | 15.90732 |
| 04:30pm | 4.9446 | 14.8338 |
| 04:45pm | 4.71393 | 14.14179 |
| 05:00pm | 3.87072 | 11.61216 |
| 05:15pm | 3.75534 | 11.26602 |
| 05:30pm | 3.51567 | 10.54701 |
| 05:45pm | 3.39696 | 10.19088 |
| 06:00pm | 3.15666 | 9.46998 |
| Total energy of 1 fixed panel | 232.7899 KJ | |
| Total energy of similar 3 fixed panels | | 698.37 KJ |

**Percentage of energy difference in each span of 15 minutes:**

| Time | (Es-3E3)/3E3 | [(Es-3E3)/3E3]*100% |
|---|---|---|
| 08.00am | 0.960828 | 96.08284 |
| 08.15am | 0.859329 | 85.93294 |
| 08:30am | 0.877272 | 87.7272 |
| 08:45am | 0.779859 | 77.98594 |
| 09:00am | 0.944292 | 94.42916 |
| 09:15am | 0.91345 | 91.34496 |
| 09:30am | 0.751075 | 75.10747 |
| 09:45am | 0.800259 | 80.02585 |
| 10:00am | 0.581182 | 58.11819 |
| 10:15am | 0.401287 | 40.12865 |
| 10:30am | 0.416559 | 41.65593 |
| 10:45am | 0.516054 | 51.60542 |
| 11:00am | -0.03167 | -3.16709 |
| 11:15am | -0.10599 | -10.5994 |

| | | |
|---|---|---|
| 11.30am | -0.12093 | -12.093 |
| 11.45am | -0.17433 | -17.4328 |
| 12.00pm | -0.20945 | -20.9445 |
| 12.15pm | -0.2259 | -22.5897 |
| 12:30pm | -0.21231 | -21.2311 |
| 12:45pm | -0.23035 | -23.0347 |
| 01:00pm | -0.29101 | -29.1007 |
| 01:15pm | 0.082669 | 8.266904 |
| 01:30pm | 0.033416 | 3.341618 |
| 01:45pm | 0.052255 | 5.225475 |
| 02:00pm | 0.037992 | 3.79918 |
| 02:15pm | 0.099311 | 9.931073 |
| 02:30pm | 0.100226 | 10.02257 |
| 02:45pm | 0.143809 | 14.38091 |
| 03:00pm | 0.184834 | 18.48339 |
| 03:15pm | 0.179556 | 17.95556 |
| 03:30pm | 0.247332 | 24.7332 |
| 03:45pm | 0.299872 | 29.98717 |
| 04:00pm | 0.292062 | 29.20623 |
| 04:15pm | 0.331138 | 33.11381 |
| 04:30pm | 0.357002 | 35.70016 |
| 04:45pm | 0.335204 | 33.52044 |
| 05:00pm | 0.545387 | 54.53869 |
| 05:15pm | 0.513812 | 51.38123 |
| 05:30pm | 0.560838 | 56.08376 |
| 05:45pm | 0.570943 | 57.09428 |
| 06:00pm | 0.58197 | 58.19696 |

Total energy difference with the conventional 3 fixed panels in a day = (863.755-698.37) KJ

= 165.385 KJ

Total percentage difference = 23.68 %

So, we can state that after implementation of the proposed multilevel sun tracking system, more energy can be produced compared the conventional single level fixed panels.

### 6.3.2 Comparing the performance of the system using graphs

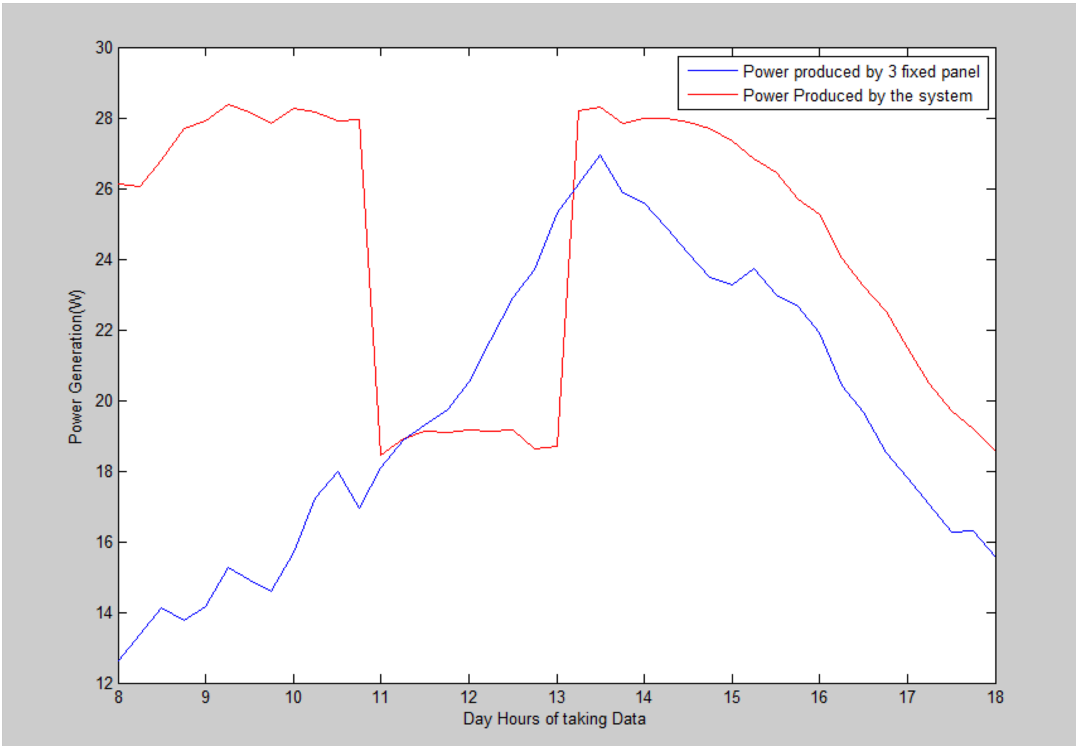



Figure 6.3: Amount of power generated from 8:00am-6:00pm on 14th July, 2016.

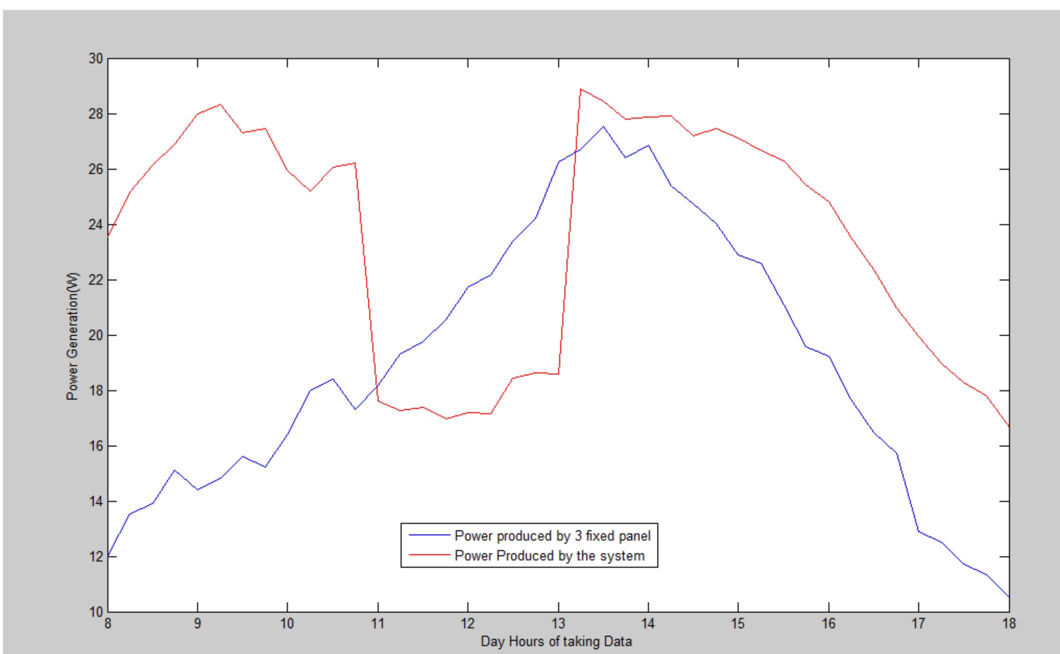



Figure 6.4: Amount of power generated from 8:00am-6:00pm on 7th August, 2016.

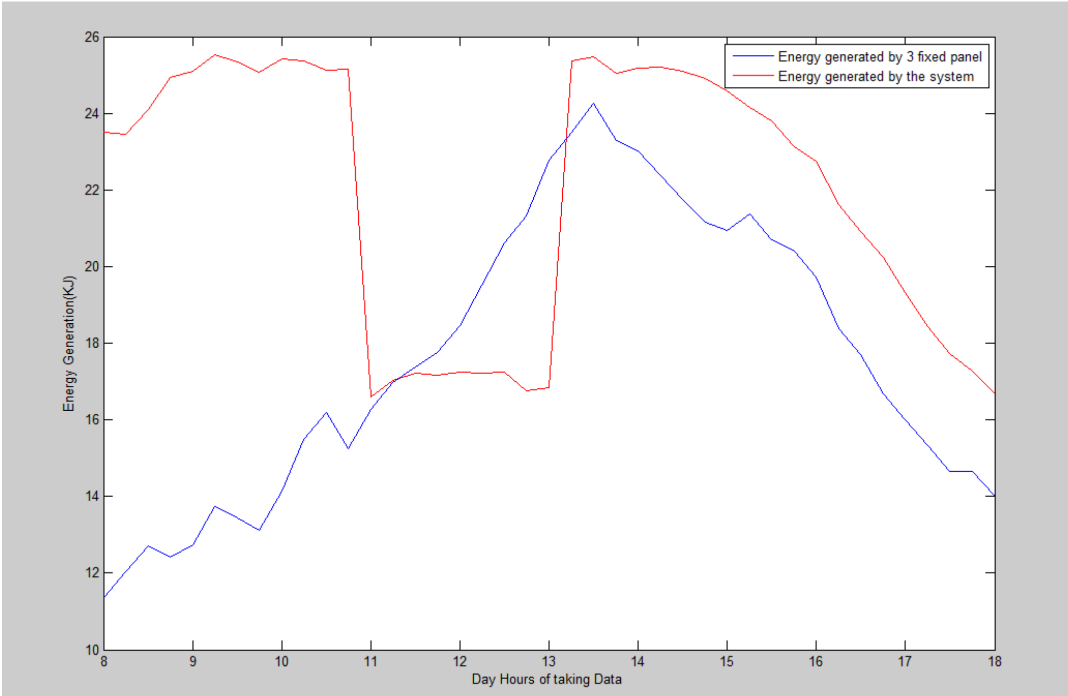



Figure 6.5: Amount of energy generated from 8:00am-6:00pm on 14th July, 2016.

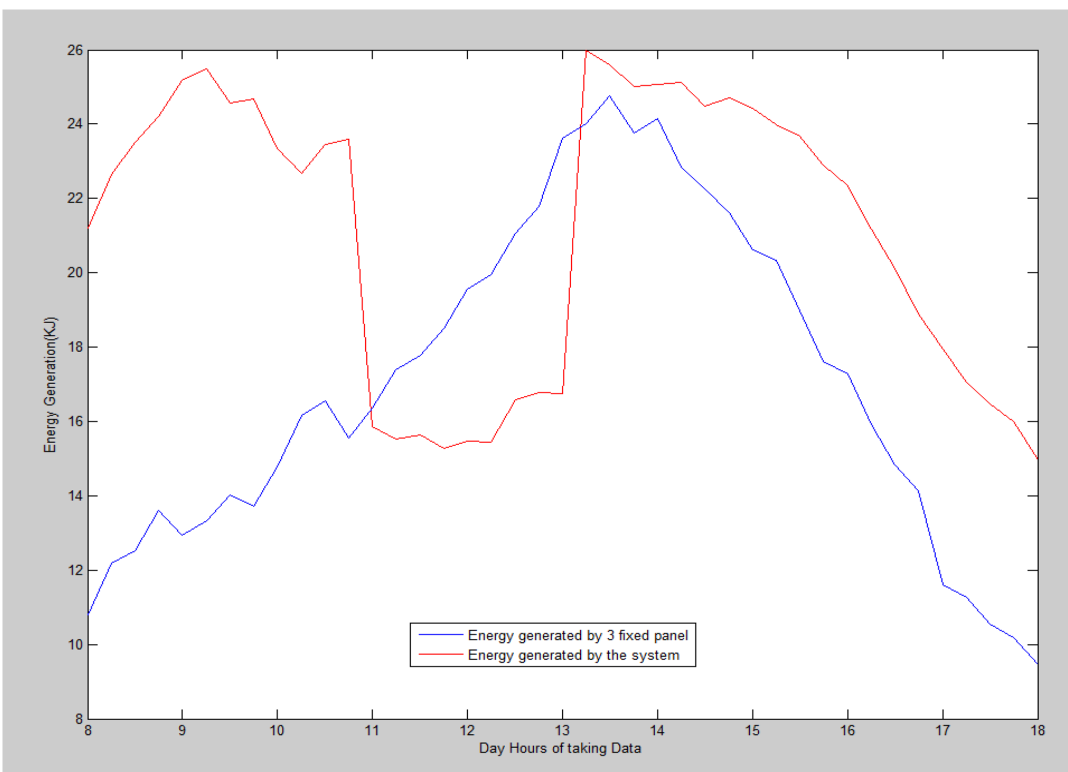



Figure 6.6: Amount of energy generated from 8:00am-6:00pm on 7th August, 2016.

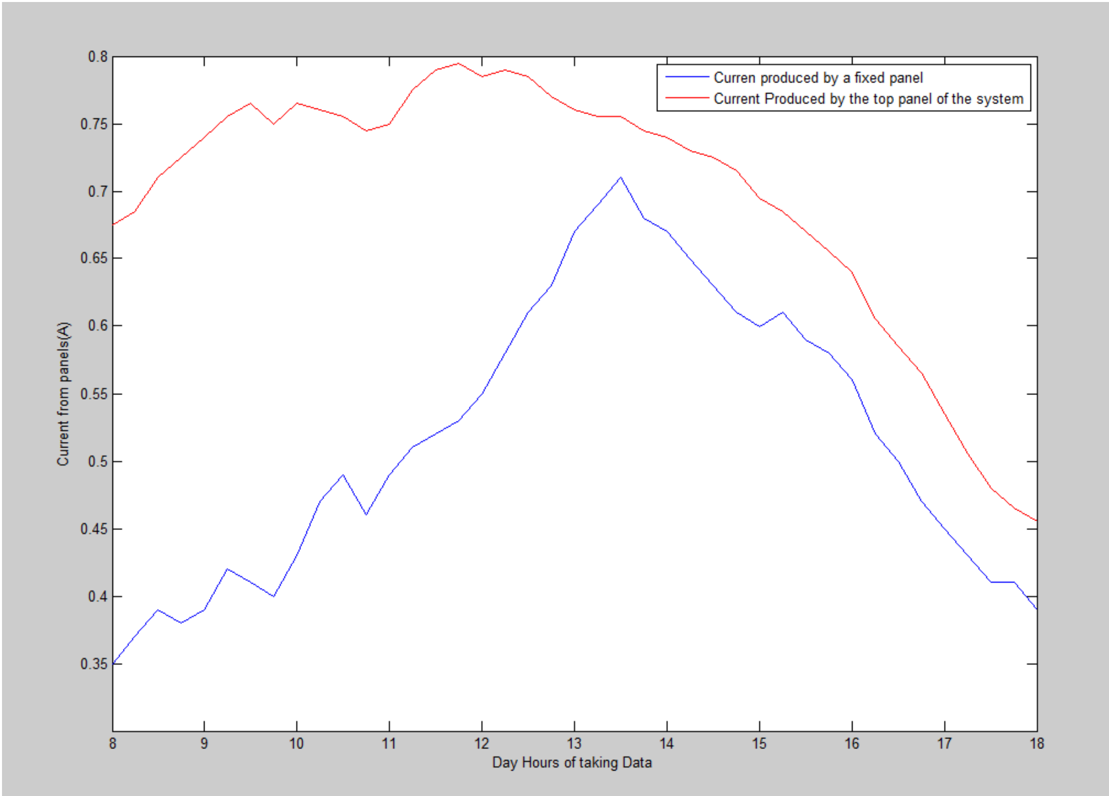



Figure 6.7: Comparison of currents generated by the top PV panel and the fixed PV panel from 8:00am-6:00pm on 14th July, 2016.

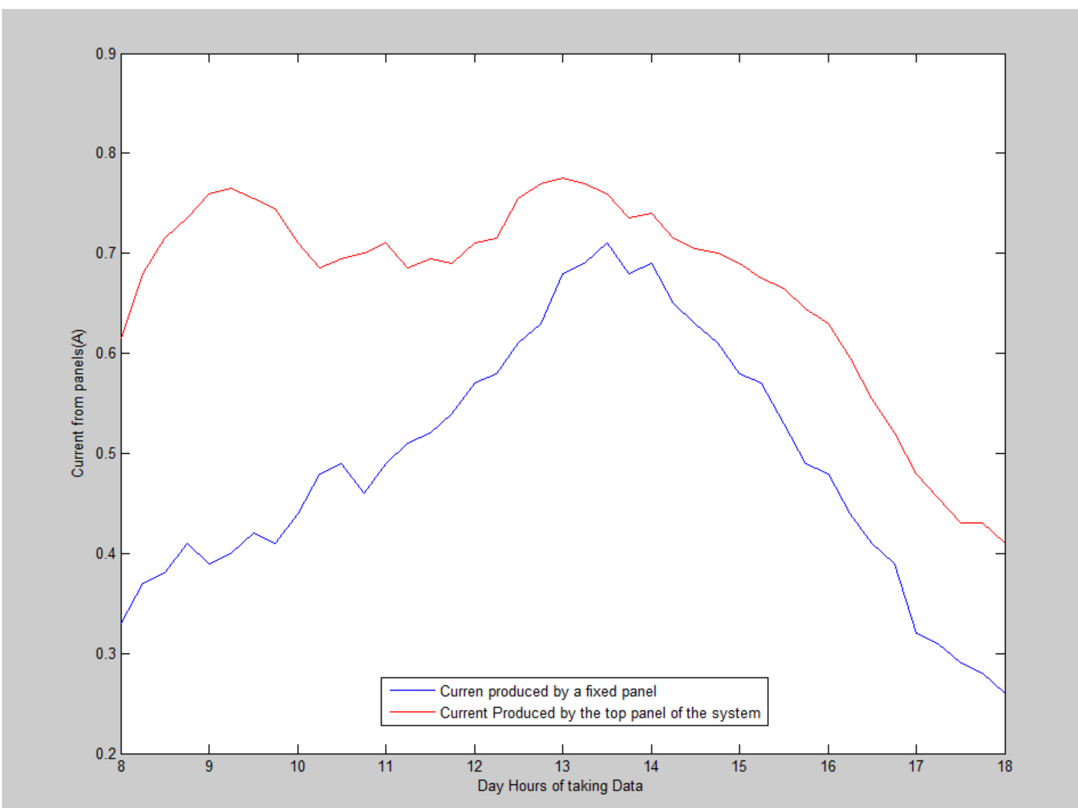



Figure 6.8: Comparison of currents generated by the top PV panel and the fixed PV panel from 8:00am-6:00pm on 7th August, 2016.

From the analysis of the graphs stated above, it can be stated that if the RTC based multilevel sun tracking system is implemented and maintained properly, it could work efficiently in a country like Bangladesh.

# CONCLUSION

The proposed system contains 3V PV solar panel stacked above one another on a single iron bar with a fixed distance in between each of them to minimize the floor space and maximize the energy accumulation from solar radiation as the panels can accurately track the sun being controlled by servo motors. Data analysis part shows that 18-24% more solar energy can be collected with the proposed system compared to the conventional single level fixed panels of same size. The proposed system with three panels can be operated within the space of two fixed panels of single level system as the tracking and sliding mechanisms of solar panels and the solar panel stack will make it more efficient. The solar power sun tracking system is more efficient because solar radiation can be an infinite source of renewable energy in our country as it is situated near the equator. If the idea of this sun tracking system can be implemented, the system will greatly reduce the electricity problem of our country. An automated microcontroller based solar tracker system with three levels of solar panel has been developed that uses a set of equations to calculate the sunrise and sunset times and send signals to the servo motors to rotate the panels by a fixed angle at pre-determined time intervals. Instead of continuous tracking, the developed system rotates the panels only 12 times a day, in a step of 15° in each rotation. As the motors are operated only for a small fraction of the total time for rotating the panels, the energy consumed by the motors will be only a small fraction of the energy collected by the system.

**Future Work:**

1) Specially designed solar concentrator can be used to further enhance the output of the system.

2) This system can be an off grid solution if we use transparent glass-made large houses to keep hundreds of these sun trackers and control all the sun trackers using a central server supported by global weather update and GPS system. This configuration can supply electrical energy to the national or local grid also.

# Reference

[1] Moniruzzaman, H. M., Patwary, M., & Rahman, M. (2014, May). A three level solar panel system with an automated solar tracker. In *Developments in Renewable Energy Technology (ICDRET), 2014 3rd International Conference on the* (pp. 1-7). IEEE.

[2] Solar Time | PVEducation. Retrieved August 03, 2016, from http://www.pveducation.org/pvcdrom/2-properties-sunlight/solar-time.

[3] Part 3: Calculating Solar Angles|ITACA. Retrieved August 04, 2016, from http://www.itacanet.org/the-sun-as-a-source-of-energy/part-3-calculating-solar-angles/.

[4] DS3231 Extremely Accurate $I^2C$-Integrated RTC/TCXO/Crystal – Maxim. Retrieved August 04, 2016, from https://www.maximintegrated.com/en/products/digital/real-time-clocks/DS3231.html/tb_tab0.

[5] Tutorial – Using DS1307 and DS3231 Real-Time Clock Modules with Arduino. Retrieved August 05, 2016, from http://tronixstuff.com/2014/12/01/tutorial-using-ds1307-and-ds3231-real-time-clock-modules-with-arduino/.

[6] CR2032 : Coin type lithium batteries (CR series) | Industrial devices & Solutions | Panasonic. Retrieved August 06, 2016, from http://industrial.panasonic.com/ww/products/batteries/primary-batteries/lithium-batteries/coin-type-lithium-batteries-cr-series/CR2032.

[7] A.W. Culp, Principles of Energy Conversion, 2nd ed., McGraw-Hill, 1991, pp. 98-107.

[8] Christopher Gronbeck, SunAngle, [Online]. Available: http://www.susdesign.com/sunangle/.

[9] Dhaka August Weather 2016 – AccuWeather Forecast for Dhaka Bangladesh. Retrieved August 10, 2016, from http://www.accuweather.com/en/bd/dhaka/28143/august-weather/28143.

# Appendix

## Codes used in Arduino:

### Code for the system running:

```
#include <Wire.h>
#include <Time.h>
#include <DS1307RTC.h>
#include <PololuMaestro.h>
#include <SoftwareSerial.h>

SoftwareSerial maestroSerial(5, 4);

MiniMaestro maestro(maestroSerial);
//These variables are used to scroll LCD, change
//the value of sn to adjust the scroll level
int sn = 14,si=0,sl=0;

//tm stores time and date information for RTC
tmElements_t tm;


//seconds,minute,hour,day,month,year
int s,mi,h,d,mo,y;

//day Number
int N;
int actuator1 = 8;
int actuator2 = 9;

double panPos = 0;
 int newPos = 0;
int actst = 0;

//SRT = Sun rise time, SST= Sun set time
double SRT, SST;
double longitude = 90.35;
double lattitude = 23.7;
const float pi = 3.14159265359;
void setup() {
  pinMode(actuator1, OUTPUT);
  pinMode(actuator2, OUTPUT);
  digitalWrite(actuator1, LOW);
  digitalWrite(actuator2, LOW);
  actst = 0;
  delay(15000);
  digitalWrite(actuator1, HIGH);
  digitalWrite(actuator2, LOW);
  Serial.begin(9600);
  while (!Serial) ; // wait for serial
  delay(200);
  maestroSerial.begin(9600);
  maestro.setSpeed(0, 2);
```

```
  maestro.setSpeed(3, 2);
  maestro.setSpeed(6, 2);
  maestro.setAcceleration(0, 127);
  maestro.setAcceleration(3, 127);
  maestro.setAcceleration(6, 127);
}
void loop() {
  if (RTC.read(tm)) {
    updateTime();
    numberOfDay();
    sunriseTime();
    sunsetTime();
    rotatePanels();
  } else {
    if (RTC.chipPresent()) {
      Serial.println("The DS1307 is stopped.  Please run the SetTime");
      Serial.println("example to initialize the time and begin running.");
      Serial.println();
    } else {
      Serial.println("DS1307 read error!  Please check the circuitry.");
      Serial.println();
    }
    delay(9000);
  }
  delay(1000);
}

void print2digits(int number) {
  if (number >= 0 && number < 10) {
    Serial.write('0');
  }
  Serial.print(number);
}

void updateTime(){
  s = tm.Second;
  mi = tm.Minute;
  h = tm.Hour;
  d = tm.Day;
  mo = tm.Month;
  y = tmYearToCalendar(tm.Year);
  print2digits(h);
  Serial.print(":");
  print2digits(mi);
  Serial.println();
}
void numberOfDay(){
  int N1 = floor(275 * mo / 9);
  int N2 = floor((mo + 9) / 12);
  int N3 = (1 + floor((y - 4 * floor(y / 4) + 2) / 3));
  N = N1 - (N2 * N3) + d - 30;
}
void sunriseTime(){
  double del = 23.45*sin(360*pi/180/365*(N+284));
```

```
  double Ws = acos(-tan(lattitude*pi/180)*tan(del*pi/180))*180/pi;
  double AST = 720-Ws*4;
  double LSTM = 15*round(longitude/15);
  double D = 360*pi/180*(N-80)/365;
  double ET = 9.87*sin(2*D)-7.53*cos(D)-1.5*sin(D);
  SRT= AST-4*(LSTM-90.25)-ET;
  //SRT = 13*60+55;
}
void sunsetTime(){
  double del = 23.45*sin(360*pi/180/365*(N+284));
  double Ws = -acos(-tan(lattitude*pi/180)*tan(del*pi/180))*180/pi;
  double AST = 720-Ws*4;
  double LSTM = 15*round(longitude/15);
  double D = 360*pi/180*(N-80)/365;
  double ET = 9.87*sin(2*D)-7.53*cos(D)-1.5*sin(D);
  SST= AST-4*(LSTM-90.25)-ET;
  //SST = 14*60+19;
}

int stat[] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

void rotatePanels(){
  double delT = (SST-SRT)/12;
  double currentTime = h*60+mi;
  if(currentTime-SRT>=0){
    if (currentTime == SRT+5.5*delT){
      panPos = 5.5;
    }else if (currentTime == SRT+7.5*delT){
      panPos = 7.5;
    }else{
      panPos = (currentTime-SRT)/delT+1;
    }
    if(panPos>12){
      panPos = 0;
    }
  }else{
    panPos = 0;
  }
  Serial.print("State Number: ");
  Serial.println(panPos);
  if (panPos<=5){
    newPos = (int) panPos;
  }else if(panPos == 5.5){
    Serial.println("Now 5.5");
    newPos = 6;
  }else if(panPos <=7){
    newPos = (int) panPos + 1;
  }else if(panPos == 7.5){
    newPos = 9;
  }else {
    newPos = (int) panPos + 2;
  }
  if(!stat[newPos]){
    reposPanels(newPos);
```

```
  }
  if (newPos>14){
    newPos = 0;
  }
}

void reposPanels(int panPos){
  double top, bottom, middle;
  switch (panPos) {
    case 0:
      digitalWrite(actuator1, LOW);
      digitalWrite(actuator2, LOW);
      delay(15000);
      digitalWrite(actuator1, HIGH);
      digitalWrite(actuator2, LOW);
      actst = 1;
      Serial.println("no state");
      top = 848*4;
      middle = 2224*4;
      bottom = 2304*4;
      break;
    case 1:
      top = 848*4;
      middle = 2224*4;
      bottom = 2304*4;
      if(actst == 0){
        digitalWrite(actuator1, LOW);
        digitalWrite(actuator2, LOW);

        delay(15000);
      }
      actst = 1;
      digitalWrite(actuator1, HIGH);
      digitalWrite(actuator2, HIGH);

      delay(5500);
      digitalWrite(actuator1, HIGH);
      digitalWrite(actuator2, LOW);

      Serial.println("state 1");
      break;
    case 2:
      if (actst == 0){
        digitalWrite(actuator1, LOW);
        digitalWrite(actuator2, LOW);
        delay(15000);
        digitalWrite(actuator1, HIGH);
        digitalWrite(actuator2, HIGH);
        delay(5500);
      }

      digitalWrite(actuator1, HIGH);
      digitalWrite(actuator2, LOW);
      actst = 1;
```

```
    top = 955.5*4;
    middle = 2043.25*4;
    bottom = 2162.25*4;
    Serial.println("state 2");
    break;
  case 3:
    digitalWrite(actuator1, LOW);
    digitalWrite(actuator2, LOW);
    delay(15000);
    digitalWrite(actuator1, HIGH);
    digitalWrite(actuator2, LOW);
    actst = 1;
    top = 883.75*4;
    middle = 2101*4;
    bottom = 2196.25*4;
    Serial.println("state 3");
    break;
  case 4:
  if (actst == 0){
      digitalWrite(actuator1, LOW);
      digitalWrite(actuator2, LOW);
      delay(15000);
      digitalWrite(actuator1, HIGH);
      digitalWrite(actuator2, HIGH);
    }
    actst = 1;
    top = 1079.00*4;
    middle = 1952.5*4;
    bottom = 2032.5*4;
    Serial.println("state 4");
    break;
  case 5:
    if (actst == 0){
      digitalWrite(actuator1, LOW);
      digitalWrite(actuator2, LOW);
      delay(15000);
      digitalWrite(actuator1, HIGH);
      digitalWrite(actuator2, HIGH);
    }
    actst = 1;
    top = 1247*4;
    middle = 1762.25*4;
    bottom = 1977*4;
    Serial.println("state 5");
    break;
  case 6:
      if (actst == 0){
      digitalWrite(actuator1, LOW);
      digitalWrite(actuator2, LOW);
      delay(15000);
      digitalWrite(actuator1, HIGH);
      digitalWrite(actuator2, HIGH);
    }
    actst = 1;
```

```
   top = 1247*4;
  middle = 624*4;
  bottom = 1977*4;
  Serial.println("state 5.5");
  break;
case 7:
 if (actst == 0){
    digitalWrite(actuator1, LOW);
    digitalWrite(actuator2, LOW);
    delay(15000);
    digitalWrite(actuator1, HIGH);
    digitalWrite(actuator2, HIGH);
  }
  actst = 1;
  top = 1396.75*4;
  middle = 624*4;
  bottom = 1848.5*4;
  Serial.println("state 6");
  break;
case 8:
  digitalWrite(actuator1, HIGH);
  digitalWrite(actuator2, HIGH);
  delay(15000);
  digitalWrite(actuator1, LOW);
  digitalWrite(actuator2, HIGH);
  actst = 1;
  top = 1984*4;
  middle = 2224*4;
  bottom =1220*4;
  Serial.println("state 7");
  break;
case 9:
  top = 1984*4;
  middle = 1113.75*4;
  bottom = 1220*4;
  Serial.println("state 7.5");
  break;
case 10:
   top = 2077.25*4;
  middle = 1113.75*4;
  bottom = 1118*4;
  Serial.println("state 8");
  break;
case 11:
  top = 2350.5*4;
  middle = 919.25*4;
  bottom = 1004*4;
  Serial.println("state 9");
  break;
case 12:
  top = 2437*4;
  middle = 802.75*4;
  bottom = 865.5*4;
  Serial.println("state 10");
```

```
    break;
  case 13:
    if (actst == 0){
      digitalWrite(actuator1, HIGH);
      digitalWrite(actuator2, HIGH);
      delay(15000);
      digitalWrite(actuator1, LOW);
      digitalWrite(actuator2, LOW);
      delay(5000);
    }
    actst = 1;
    digitalWrite(actuator1, HIGH);
    digitalWrite(actuator2, LOW);

    top = 2349.25*4;
    middle = 802.75*4;
    bottom =1006.5*4;
    Serial.println("state 11");
    break;
  case 14:
    if (actst == 0){
      digitalWrite(actuator1, HIGH);
      digitalWrite(actuator2, HIGH);
      delay(15000);
      digitalWrite(actuator1, HIGH);
      digitalWrite(actuator2, LOW);
    }
    actst = 1;
    digitalWrite(actuator1, LOW);
    digitalWrite(actuator2, LOW);
    delay(5000);
    digitalWrite(actuator1, HIGH);
    digitalWrite(actuator2, LOW);

    top = 2528*4;
    middle = 624*4;
    bottom = 768*4;
    Serial.println("state 12");
    break;
}
stat[newPos] = 1;
if (!newPos){
  stat[12] = 0;
}else{
  stat[newPos-1] = 0;
}
maestro.setSpeed(0, 2);
maestro.setSpeed(3, 2);
maestro.setSpeed(6, 2);
maestro.setAcceleration(0, 2);
maestro.setAcceleration(3, 2);
maestro.setAcceleration(6, 2);
maestro.setTarget(0, top);
maestro.setTarget(3, middle);
```

```
  maestro.setTarget(6, bottom);
  delay(2000);
  maestro.setTarget(0, 0);
  maestro.setTarget(3, 0);
  maestro.setTarget(6, 0);
  Serial.print("Top Panel: ");
  Serial.println(round(0.029532*(top)-90.407));
  Serial.print("Middle Panel: ");
  Serial.println(round(0.029532*(middle)-90.407));
  Serial.print("Bottom Panel: ");
  Serial.println(round(0.029532*(bottom)-90.407));
}
```

**Code for data logger:**

```
#include <SPI.h>
#include <SD.h>
#include <Wire.h>
#include <Time.h>
#include <DS1307RTC.h>
File myFile;

int currentSensor1 = A0;
int currentVal1;
int voltageSensor1 = A1;
int voltageVal1;
int currentSensor2 = A2;
int currentVal2;
int voltageSensor2 = A3;
int voltageVal2;
int currentSensor3 = A4;
int currentVal3;
int voltageSensor3 = A5;
int voltageVal3;
int changeFile = A6;
double sumV1, sumV2, sumV3, sumI1, sumI2, sumI3;
int c;
double pc1, pc2, pc3, pv1, pv2, pv3;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial.println("Current sensing started!");
  Serial.print("Initializing SD card...");

  if (!SD.begin(53)) {
    Serial.println("initialization failed!");
    return;
  }
  Serial.println("initialization done.");
  sumV1 = 0;
  sumV2 = 0;
  sumV3 = 0;
```

```
  sumI1 = 0;
  sumI2 = 0;
  sumI3 = 0;
  c = 0;
  pc1 = 0;
  pv1 = 0;
  pc2 = 0;
  pv2 = 0;
  pc3 = 0;
  pv3 = 0;
}

void loop() {
  tmElements_t tm;

  // put your main code here, to run repeatedly:
  while(c<200){
    currentVal1 = analogRead(currentSensor1);
    voltageVal1 = analogRead(voltageSensor1);
    currentVal2 = analogRead(currentSensor2);
    voltageVal2 = analogRead(voltageSensor2);
    currentVal3 = analogRead(currentSensor3);
    voltageVal3 = analogRead(voltageSensor3);

    sumI1 += currentVal1;
    sumV1 += voltageVal1;
    sumI2 += currentVal2;
    sumV2 += voltageVal2;
    sumI3 += currentVal3;
    sumV3 += voltageVal3;
    c++;
    delay(10);
  }

  double result = round(sumI1/c);
  double current1 = floor((0.028496*result - 14.629)*1000);
  result = round(sumV1/c);
  double voltage1 = 0.023647*result-1.0186;

  result = round(sumI2/c);
  double current2 = floor((0.026189*result - 13.463)*1000);
  result = round(sumV2/c);
  double voltage2 = 0.056733*result-1.7727;

  result = round(sumI3/c);
  double current3 = floor((0.029002*result - 14.601)*1000);
  result = round(sumV3/c);
  double voltage3 = 0.055008*result-1.5791;
  if ((pc1!=0 && abs(current1-pc1)>0.02)|| (pv1!=0 && abs(voltage1-
pv1)>0.05)||
      (pc2!=0 && abs(current2-pc2)>0.02)|| (pv2!=0 && abs(voltage2-
pv2)>0.05)||
      (pc3!=0 && abs(current3-pc3)>0.02)|| (pv3!=0 && abs(voltage3-
pv3)>0.05)){
```

```
    //Serial.println("Broken");
  }else{
    RTC.read(tm);
    Serial.print(tm.Hour);
    Serial.print(":");
    Serial.print(tm.Minute);
    Serial.print("     ");
    Serial.print("c1: ");
    Serial.print(current1);
    Serial.print("     ");
    Serial.print("v1: ");
    Serial.print(voltage1);
    Serial.print("     ");
    Serial.print("c2: ");
    Serial.print(current2);
    Serial.print("     ");
    Serial.print("v2: ");
    Serial.print(voltage2);
    Serial.print("     ");
    Serial.print("c3: ");
    Serial.print(current3);
    Serial.print("     ");
    Serial.print("v3: ");
    Serial.println(voltage3);
    myFile = SD.open("data.txt", FILE_WRITE);
    RTC.read(tm);
    if(myFile){
      myFile.print(tm.Hour);
      myFile.print(":");
      myFile.print(tm.Minute);
      myFile.print("    ");
      myFile.print("c1: ");
      myFile.print(current1);
      myFile.print("    ");
      myFile.print("v1: ");
      myFile.print(voltage1);
      myFile.print("    ");
      myFile.print("c2: ");
      myFile.print(current2);
      myFile.print("    ");
      myFile.print("v2: ");
      myFile.print(voltage2);
      myFile.print("    ");
      myFile.print("c3: ");
      myFile.print(current3);
      myFile.print("    ");
      myFile.print("v3: ");
      myFile.println(voltage3);
    }
    myFile.close();
  }

  c = 0;
  sumI1 = 0;
```

```
  sumV1 = 0;
  sumI2 = 0;
  sumV2 = 0;
  sumI3 = 0;
  sumV3 = 0;

  pc1 = current1;
  pv1 = voltage1;

  pc2 = current2;
  pv2 = voltage2;

  pc3 = current3;
  pv3 = voltage3;
}
```