Solar Panel And Inverter Monitoring System With Mobile Application

Group Members

MUHAMMAD HASEEB	**200718**
SYED MUHAMMAD USAMA BIN AQIL	**200758**
RAO UMER RASHID	**200776**

BACHELOR ELECTRICAL ENGINEERING (ELECTRONICS) (2020-2024)

Project Supervisor

DR. MUHAMMAD ZAHEER

Designation of Supervisor

ASSISTANT PROFESSOR

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

FACULTY OF ENGINEERING

AIR UNIVERSITY, ISLAMABAD

Solar Panel And Inverter Monitoring System With Mobile Application

Final Year Project

(SESSION 2020-2024)

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Solar Panel And Inverter Monitoring System With Mobile Application

Submitted by:

MUHAMMAD HASEEB	**200718**
SYED MUHAMMAD USAMA BIN AQIL	**200758**
RAO UMER RASHID	**200776**

Project Supervisor

____

Dr. Muhammad Zaheer

Assistant Professor

Head of Department

____

Dr.Ashiq Hussain

Head Of Department

Acknowledgement

Praise be to the Most Merciful, Allah, and Lord of all worlds who bestowed us with the knowledge and gave us the wisdom. We are grateful to our supervisor, Engr. Dr. Muhammad Zaheer, for his advice and insights, as well as his assistance and direction. We are extremely fortunate to have such supportive parents; without their help, none of this would have been possible, and we would not be here. We value our jury's opinions and ideas at each level since they are critical to the effective completion of this study. We would like to thank all of the faculty members for their assistance and support during this project, especially when we shared concerns, we had with them. We would like to thank all of our friends for their encouragement and support during this effort. We'd like to thank the Lab Engineers for their encouragement, which has paved the road for us to learn, plan, and execution the project.

Abstract

This project is a Solar Panel and Inverter Monitoring System alongside a mobile application, this also includes is a designed inverter. The collected data undergoes processing and is displayed on a mobile app, allowing users to engage in real-time monitoring and remote-control functions. The monitoring system effortlessly incorporates hardware components and sensors to gather crucial data from solar panels and the crafted inverter. The application is a user-friendly interface. The inverter design contributes to improved system efficiency. By combining effective monitoring of the inverter, this project plays a role in promoting sustainable solar energy demand. This provides users with practical understanding for proper energy management and the ease of use of renewable energy.

Table of Contents

List of Figures

Chapter 1
Introduction

1.1 Brief Background

For our FYP we have chosen a solar energy related project. The name of our project is Solar Panel and Inverter Monitoring System with Mobile Application. As the name states we will be monitoring the data from both the solar panel system and the inverter and integrate this data with a custom mobile application. For our project we will not only be designing the mobile application ourselves, but we will also design the inverter as well as the charge controller.

1.2 Motivation

We chose this as our Final Year Project as it was related to solar energy, since it is one of the most abundant resource available to mankind and still it is not being used to its fullest potential. This is due to the lack of awareness in the society regarding the benefit of Solar Power. Inverters these days that are available in the market are extremely costly and out of reach for the common man. Overall mankind has contributed a lot towards climate change. Opting for solar energy reduces the overall carbon footprint and makes the environment much cleaner and safer. There has also been an exponential increase in the overall population of the country and our industrial sector have also seen a lot of growth over the past years due to which the demand for electric energy is at an all-time high and is not being fulfilled due to scarce resources, solar is a more viable option to shift to. The new photovoltaic technology seen in today's solar panels is much more efficient and cost effective. Looking at the ever-increasing trend of the country's population the increasing demand for electricity is not possible to be met and other options are to be considered to fulfill the increasing demand for energy.

1.3 Problem Statement

Pakistan is facing an energy crisis and is unable to provide its inhabitants with sufficient amounts of electricity. The problems that were observed in the society which led us to choosing Solar energy for our FYP were that there was a huge energy crisis being faced not only throughout the country but worldwide. Due to increasing demand and finite supply of nonrenewable resources, we are faced with an exponential increase in demand of the already depleting resources.

The more our resources are becoming scarce the higher the cost of using them. Electricity is one of the most basic needs in today's world. Therefore, the increasing cost of electricity is making it difficult for people to afford hence the population is slowly being deprived of electricity. The increased cost of electricity and its low supply has led to frequent power outages. This has significantly reduced the standard of living of the population.

Areas of Pakistan that have yet not been fully developed or are underdeveloped lack the structures to facilitate them with electricity. Due to which the health and security of people living in such areas is at risk as their quality of life is being affected.

There is lack of awareness when it comes to solar energy and its benefits or what solar energy itself is. Even though solar energy being the most abundant and easily accessible renewable source of energy people are still not opting for it.

The industrial sector of the country is also suffering greatly due to the frequent power outages and in turn this affects the economic growth of the country causing us a huge loss for the present and future generations.

1.4 Objectives

Keeping the problem statement our objectives are as follow:

- Designing a small-scale model of a solar-powered inverter for the purposes of testing our monitoring system's application.
- Self-designed Inverter tested with load.
- Data Collection from Solar Panel and inverter through sensors.
- Communicate the data to cloud storage through a module.
- Display Collected data on an Mobile Application.
- Storing data on the cloud for future reference and analysis.
- Alert notification on any abnormal behavior.

Chapter 2
Literature Review

2.1 Solar Panels

The Earth has an abundance of sunlight leading to extensive solar power potential which is close to 173,000 TW(Terawatts) that is sufficiently more than the whole populations power consumption. Due to expanding economy and growing population there is an increase in demand for energy. The use of renewable energy is being considered in recent times over other sources as non-renewable resources can be harmful for our environment. Solar energy provides an optimal solution to this situation. [1]

Solar energy with a PV system is the most promising renewable energy nowadays. It is free from carbon emissions making it environmentally friendly. It is also readily available and due to the abundance in nature there is no cost for the source. [2]

2.1.1 Working of a Solar Panel?

Solar cells, which are smaller components, make up solar panels. The semiconductor silicon, which is of the most prevalent element on Earth, it is used in the most of the solar cells today. Crystalline silicon is placed in cases of conductive layers within a solar cell. Four strong bonds bind each silicon atom to its neighbours, holding the electrons in place and preventing the flow of current. Two distinct layers of silicon are used in solar cells. Extra electrons are present in N-type silicon, and extra holes, or places for electrons, are present in P-type silicon. Electrons have the ability to move across the PN Junction, leaving a positive charge on one side and producing a negative charge on the other, where the two types of silicon meet. One way to think about light is as a stream of tiny particles known as photons that are released from the sun. One of these photons can break an electron from its link in the silicon cell by striking it with enough energy, creating a hole. Now that the positively charged hole and negatively charged electron are free to move, the electric field at the PN Junction will only let them to travel in one direction. Whereas the hole is drawn to the ”P” side, the electron is drawn to the ”N” side. At the top of the cell, thin metal fingers gather the mobile electrons, which then pass via an external circuit to power an electrical device, such as a lightbulb.

2.1.2 Types of Solar Panels

The most commonly available solar panels can be fit in one of the three types. Monocrystalline, thin film and Polycrystalline. Monocrystalline solar panels, better known as ”single crystal panels”, they are made from a single pure silicon crystal cut into numerous wafers. Since they are made from pure silicon they can be identified by their deep black color. The use of pure silicon makes these type of solar panels more space efficient and longest lasting their efficiency is near to 20% and they are optimized for commercial use because of their high lifetime value.

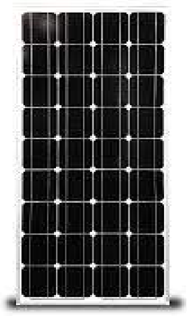

Figure 2.1: MonoCrytalline Panels

Polycrystalline panels are made from various types of silicon crystals instead of one. The silicon fragments are melted at high temperatures and poured into a square shaped mold making this type of panel affordable since there is hardly any wastage. Although this type of solar panel is more sensitive to high temperatures and has a lower expected lifespan and has comparatively lower efficiency. The efficiency rate of these solar panels is around 15%.

Figure 2.2: PolyCrytalline Panels

Thin Film solar panels are identified by very fine layers to the point they are flexible. Each panel does not require a frame backing making them lighter and easier to install. These are relatively produced at low cost and are easy to produce and they are very flexible. The efficiency rate of these penalties 7-10% and their lifespan is the smallest compared to the other solar panels.

Figure 2.3: MonoCrytalline Panels

2.2 Arduino Nano

The Arduino Nano is a powerful Micro Controller Board in the realm electronics. The Nano is powered by the ATmega328P micro controller, a powerful and efficient chip that enables it to be used in projects in numerous situation especially to produce a PWM, record and display data.

Due to its compact size. Measuring at just 45mm × 18mm, it exhibits an impressive array of functions, making it ideal for projects where space is very limited however, the Nano doesn't compromise on performance. It comes packaged 16MHz clock speed and 32KB of flash memory, it can handle a diverse range of tasks, from interfacing with sensors and process data.

Versatility is another important trait of the Arduino Nano. Equipped with an array of digital and analog pins. 14 digital pins and 8 analog pins. The Nano has ample ports for connecting peripherals and expanding functionality.

Ease of use is a major characteristic of the Nano. Its onboard USB interface simplifies the process of programming and debugging, allowing the user to upload code and monitor serial output with ease. The Arduino IDE is used to interface with the board.

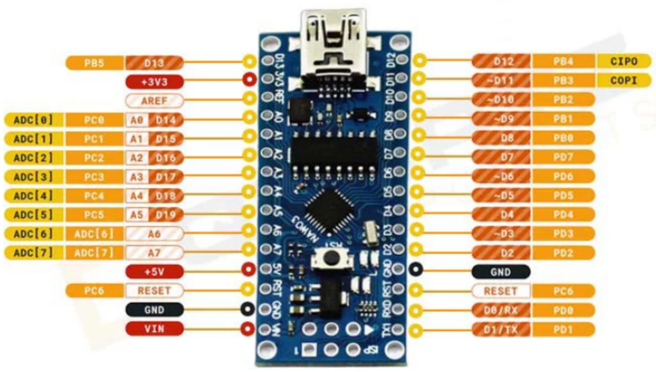

Figure 2.4: Arduino Nano Pin-OUT

The Arduino Nano's affordability is another reason that contributes to its widespread usage. Despite its modest price tag, the Arduino Nano doesn't lack any features. In addition to its core functionality, the Nano offers a range of capabilities that enhance versatility. These include built-in support for pulse-width modulation (PWM), which enables smooth control of motors and LEDs, as well as hardware and software serial communication for interfacing with external devices such as GPS modules and wireless transceivers. The Nano can be powered via USB or an external power source, providing flexibility in terms of deployment and power management.

The Arduino Nano's potential uses can be further expanded by its ability to be seamlessly integrated with other hardware platforms and software tools, in addition to its standalone capabilities. The possibilities are only limited by your imagination, whether you're using cloud services to analyse and visualize data collected by your Nano-based sensor network or integrating with a Raspberry Pi to create a potent IoT gateway.

2.3 Charge Controller

Various charge controllers are used to convert Solar Power to the required form of Energy. Among various methods of charge controllers, the most popular charge controller is PWM based and MPPT based. As we know that the power generated by the Solar Panel is solely dependent on the sunlight which itself is never constant from sunrise to sunset. Due to this variation in solar energy because of dependence on the intensity of the sunlight that strike the panel, a controller is required that manages the charging of the backup battery. Depending on the energy stored inside the battery it requires a specific charging voltage and current. This current varies from maximum input current to zero ampere and the voltage ranges from 12 volt to 14 volts [3]. The needs of the battery and the incoming solar energy are completely different. This difference requires a charge controller so that the incoming solar energy is supplied as per the need of the battery. [4] Because of this variation, a regulator must make sure that solar power is delivered in a way that meets the needs of the battery. When the battery backup is at its lowest, the controller tries to maximize solar absorption and reduce extra energy as the battery gets closer to its maximum capacity. Once the battery is fully charged, the controller refrains from charging it further and instead monitors the battery voltage at this point.

Different types of charge controllers are available in the market and they're working depends on different concepts the most frequently used charge controllers are MPPT based solar charge controller and PWM based charge controller.

The battery voltage is controlled by the PWM solar charge controller. It uses an oscillating circuit in which the battery voltage determines the pulse width. If the battery voltage drops below the pulse width, all the solar energy that enters is directed toward battery storage. As the battery storage voltage rises, the pulse width of the PWM decreases. Likewise, when the battery backup is full, the pulse width reduces to a minimum, limiting solar energy storage. During this phase, the controller solely monitors the battery voltage. [5].

This controller offers a significant benefit by enhancing battery longevity, thereby improving battery life. As mentioned earlier, another type of charge controller is the MPPT charge controller, which stands for Maximum Power Point Tracking. Initially, this controller decreases input current while boosting input voltage, and conversely, during peak hours, it increases input current while reducing input voltage.

2.3.1 PWM Charge Controller

A commonly employed charge controller for storing energy from solar panels to batteries is known as the Pulse Width Modulation (PWM) charge controller. In this concept, the solar energy passes through a switching circuit so that energy can be stored within the battery. An oscillator regulates the switching circuit and adjusts the pulse width which is based on the energy stored in the battery. Figure 2.4 illustrates a block diagram of the PWM charge controller.

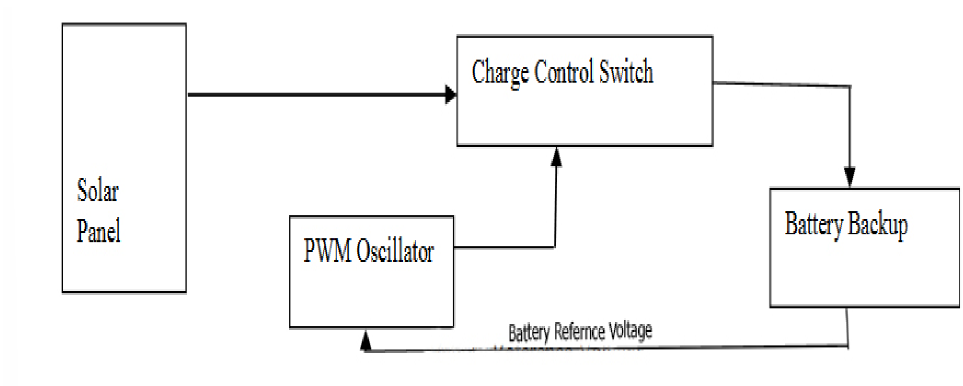

Figure 2.5: Block Model of Solar PWM charge controller
[3]

The pulse width of low state will be a single spike and the pulse width of high state will be large enough if the energy in the battery backup is depleting. A model to show the nature of pulses during heavy charging is shown on figure 2.5.

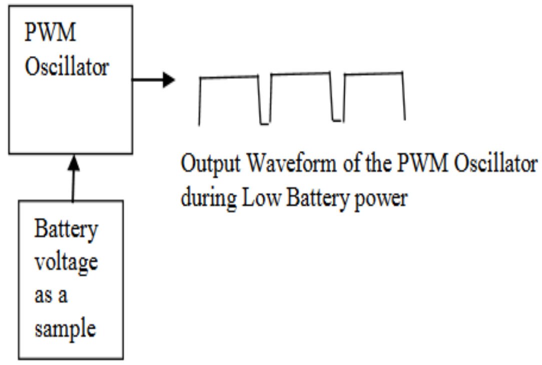

Figure 2.6: The output of PWM oscillator during low battery storage
[3]

The battery is connected to the output of the PWM oscillator through a switching circuit The on state of the pulse width will reduce and the off stage will increase once the battery is getting filled. Figure 2.6 shows the nature of the pulse once the battery is 50% charged.

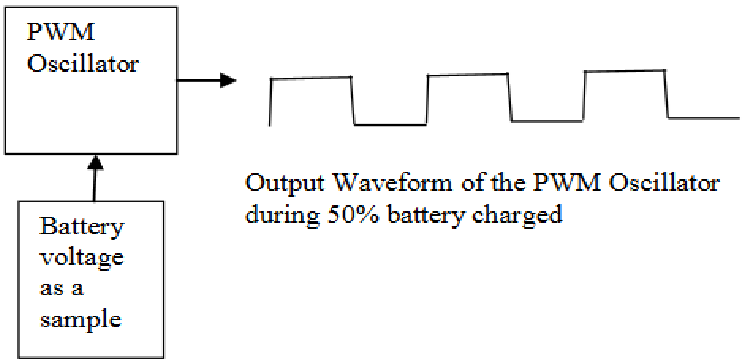

Figure 2.7: The output of PWM oscillator during 50% battery storage
[3]

Most of the time the pulse will be in a low sate only and there will only be a spike in high state this happens when the battery storage is full, and the spike is only there so that you can just sense the battery backup level. The pulse width of the oscillator during the fully charged battery state is shown in figure 2.7.

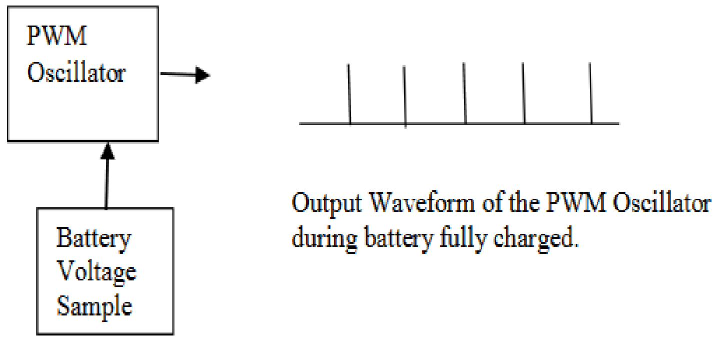

Figure 2.8: The output of the PWM oscillator during fully charged battery [3]

It should be noted that the PWM controller is not a DC to DC transformer.It is a switch that connects the battery to the solar panel PWM system has the following advantages [6]:

- Higher charging efficiency
- Longer battery life
- Reduced battery over heating
- Minimizes stress on the battery
- Ability to de-sulfate a battery

2.3.2 MPPT Charge Controller

Nowadays an advanced solar charge controller available called Maximum Power Point Tracking (MPPT). It is more expensive and considered more sophisticated than its predecessors. The most important and distinguishable quality in this solar charge controller is its efficiency at lower temperatures. It is said to be 30 to 40% more efficient at lower temperatures.

The maximum power is taken from the solar panel by either increasing the voltage and decreasing the current or vice versa. The required voltage for a battery backup is usually 12V -14V for a typical 12V battery. Similarly, a 12V solar panel will vary its output voltage from 0V – 15V. Any solar output voltage outside the required range of the battery will be wasted. The MPPT technology converts the output voltage of the solar panel to this range required for the battery. So, in this case any voltage from 0V-15V will be converted to 14V by changing the current to ensure that, during all conditions, the battery backup is charging considering the dependence of the output voltage of solar panel on the sunlight. Figure 2.8 shows a conceptual model of the controller.

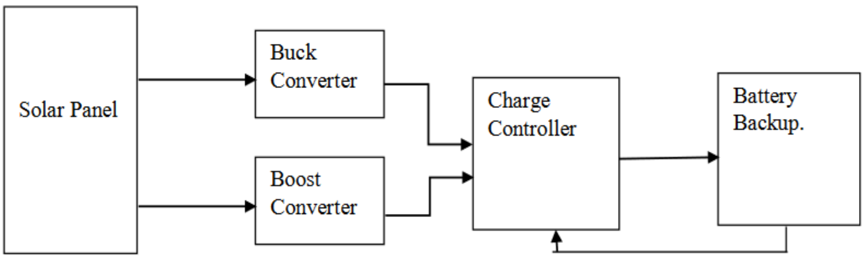

Figure 2.9: Block Model of MPPT charge controller

[3]

2.3.3 Comparision

A MPPT solar charge controller should be used if maximizing the charging capacity is the only factor considered. However there are many other factors including cost ,load and size of a particular solar panel system that should be included before choosing the suitable controller.

- MPPT controllers are typically more expensive but more efficient.
- The PWM solar charge controllers are unable to utilize the access voltage because the Pulse Width Modulation technology charges at the same voltage as the battery.
- MPPT controller is far better in colder conditions as it is able to capture the excess voltage.

- PWM is typically recommended for use in smaller systems such as micro grids installed in houses.

2.4 Inverters

In simpler terms, inverter takes direct current power as input and converts it into alternating current. The main purpose of an inverter is to be able to run those electronic devices and appliances that required ac power. They became necessary in situations where only dc power is available such as solar panels and batteries. The inverter receives DC power at its input and then through electrical switching with the help of MOSFETs/IBJTs converts this DC power into AC power.

2.4.1 Applications

Inverters play a very crucial role in off grid power or ”localized” power systems. Many rural areas are often disconnected to the electric grid. They use energy from sources like batteries and solar panels to run their daily household appliances. Similarly, even in urban areas many households use uninterruptible power supply (UPS) systems. When the regular AC power is disrupted from the grid or disconnected due to any sudden or known fault, an inverter is used to run the household appliances which are AC power. Electrical vehicles are becoming a huge part in the modes of Transportation. The Electrical vehicles use AC motors to drive. But the main source of energy in these vehicles is from dc batteries, hence an inverter is used.

2.4.2 Common Topologies

There are two main types of converters when it comes to converting a dc signal to an ac signal, full- bridge and half-bridge. A bridge circuit can be called an electric circuit that connects two nodes by providing a path between them. The Wheatstone Ridge is a very common type of bridge circuit which is studies widely in electronics. In power converters, the build circuit are used to alter the voltage or current in any electrical circuit. An electronic circuit converts DC current to AC current is called a bridge converter. Half bridge converter uses two switches rather than four. DC power sources are connected to both positive and negative side of the switches. When one switch is on the path that the current follows is through the positive end of our power supply through the load and when the other switches on the current flow is again going backwards towards the negative side of the power source. The main disadvantage of this type of converter is that its output is half the power source.

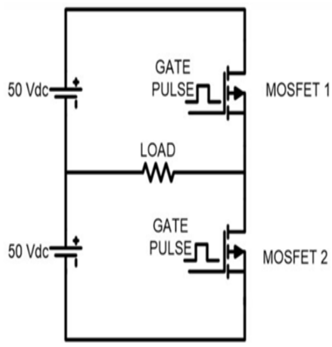

Figure 2.10: Half Bridge Inverter

In reference to the desired frequency at the output side, the time for which the mosfet is off and on is calculated so that the gate pulses are generated. For half the cycle, Mosfet one is triggered and for the rest of the cycle Mosfet 2 is triggered. After half cycle, the current will change direction through the load, hence producing an Ac output.

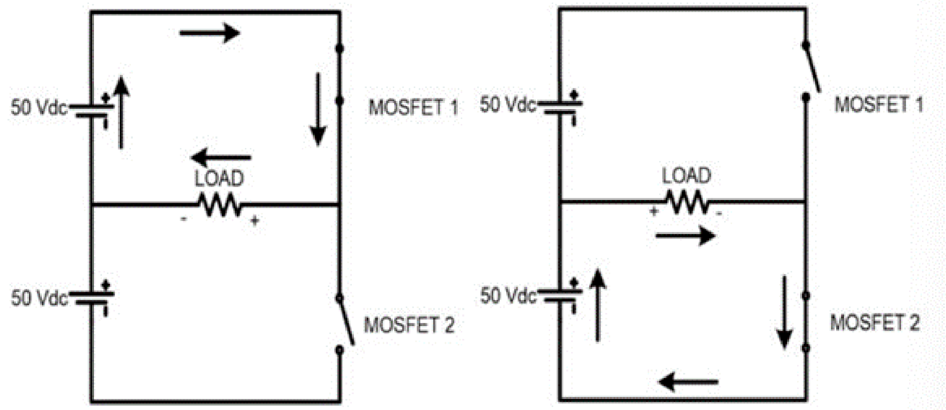

Figure 2.11: Half Bridge Inverter Working

Basic and most noticeable differentiation between the commonly used half bridge and our full bridge is the combination of four switches. For the common half bridge, the maximum peak voltage comes out to be almost half of that of the DC supply voltage we have. But for the working of the full bridge inverter the maximum peak Voltage value is equal to the value of DC supply voltage.

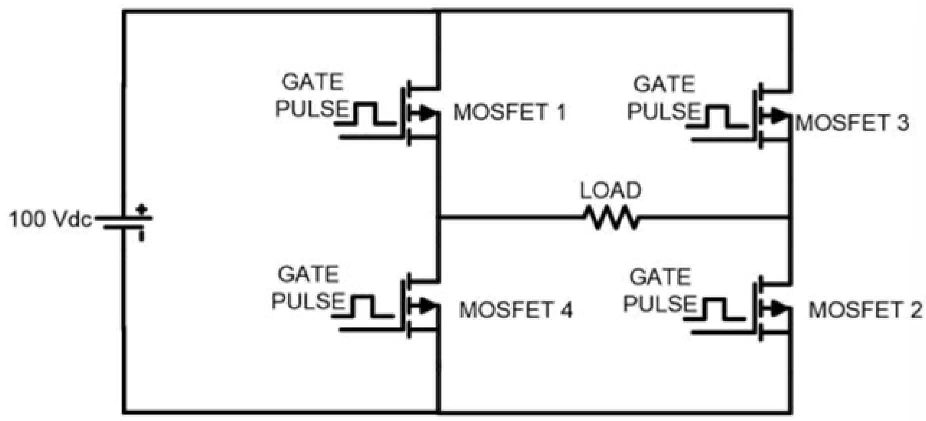

Figure 2.12: Full Bridge Inverter

The switch is 1 and 2 are operating at the same time and so is the case for the switches 3 and 4. Switches 1 and 4 can never operate simultaneously or else short circuit will happen. In the first half cycle both MOSFETs are turned on and electric current flows as it is observed in the figure 2.11. During the time period it is seen that the current flow is observed to be going from left to right.

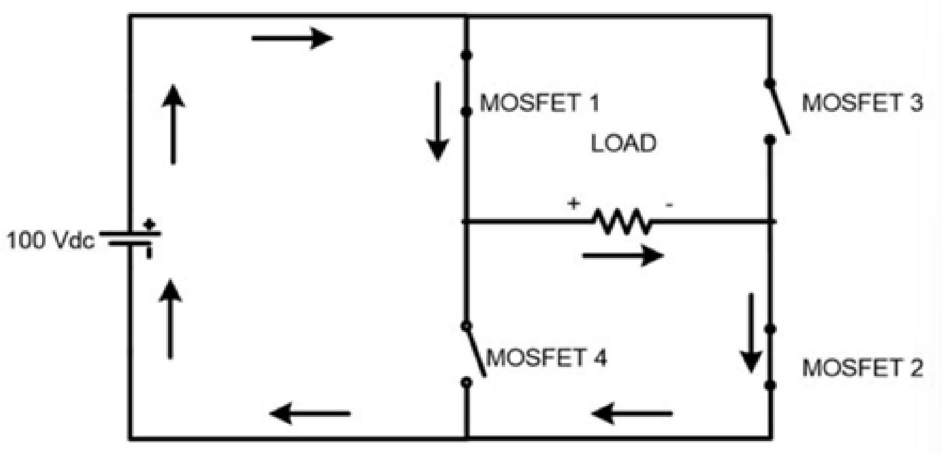

Figure 2.13: Full Bridge Inverter Working

For the next half cycle, Mosfets 3 and 4 will be switched on. and current will change its direction through the load. It is important to note that the peak load voltage in this case will be equal to the DC voltage obtained from the supply.

The three main types of output waveforms generated by inverters are square, modified sine wave and pure sine wave. Depending on the PWM applied to the Mosfets we can achieve different waveforms at the load. If the switching between Mosfets is done without any delay, we get a square wave across the load. You can use a microcontroller or multi vibrator like 555 Timer IC or IC4047 to create the PWM signals for the switching with the frequency of 50 or 60 Hertz. Square waveform was used by old inverters. Some inverters use the modified square wave.

In these inverters there is a pause between the change of polarities. In figure 2.12 Switches S1 and S2 turns on. Then all the switches turn off. Then S3 and S4 turn on. Again, all the switches turn off and the cycle repeats. This is achieved by reducing our pwm and in turn the duty cycle of the signal. This. Modified square wave is more similar to the sine wave than the square wave. Square Wave contains harmonics, if those harmonics were to be removed we would obtain a pure sine wave.

2.5 Communication

2.5.1 ESP32

The ESP32 is a widely used microcontroller that is used in the field of embedded system and Iot based applications. It is developed by Espressif systems. ESP32 comes equipped with the dual core CPU which can be easily sped up till upto 80 to 240 Mhz. This is a great number of compiling power for a very considerably small chip. And also the esp comes equipped with a ULP. The use for this slower processor is to perform less complicated calculations when the faster dual core CPU is not being used. The ESP32 comes also equipped with 512Kb of on chip SRAM memory being completely used for program instructions. Also it gives support in the external memory calculations and towards the dependency of the board as it uses 4-8Mb.This results in our understanding giving us a meaning that it is also good for our more complicated reasons such as integrating it with camera, understanding speech, also the streaming of data from the web and further on. But the main distinguishable quality of this chip is that it has built in Wi-Fi and Bluetooth so there is no need for additional radio modules like the ones typically used on Arduino boards. There is also a 12-bit ADC that is there to calculate external voltage. There are also 10 pressure sensors to detect capacitive touches. It also has up to 34 programmable GPIO pins.

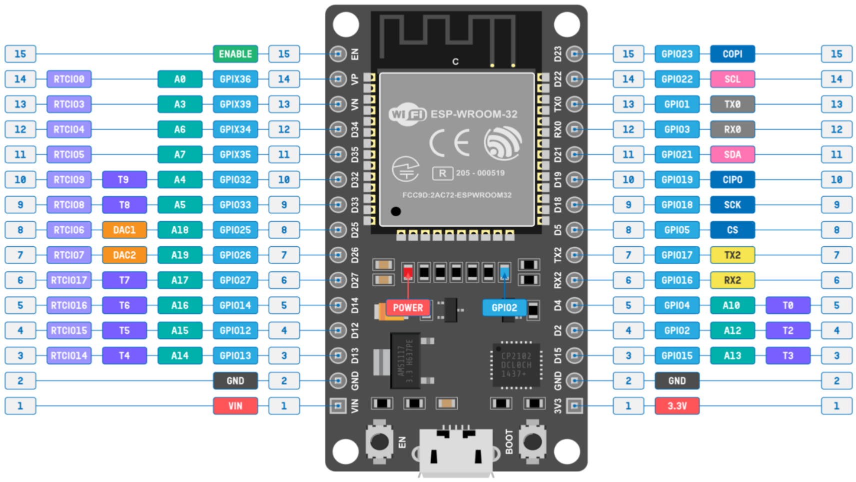

Figure 2.14: ESP32

One of the most common applications of the ESP32 microcontroller Is in smart home automation can be used to monitor devices such as lights, security cameras etc. The Wifi connectivity of the microcontroller allows these devices to be monitored or accessed through a mobile application or a web interface. In summary, the wireless connectivity, dual-core processing and a handful set of peripherals makes this microcontroller as a robust choice for Iota applications

2.5.2 Cloud storage

Communication through the Internet is becoming vital in recent times. Considering that a web based online interactive application should be capable enough to share text, images or any sort of documents in a much more convenient way with least minimum delay. In which our Firebase is the most common web-based site which provides a Real-time database and cloud services which enables the users to create various applications without any hiccups. [7].

There are different types of cloud storage systems some are very specific in the items that are stored on them, some store messages emails or digital pictures. While others save various different types of user data at the same time other available cloud storage systems present are not so large operations whereas others are done at a larger scale to the point physical equipment can take up a whole warehouse worth of space these facilities are called data centers.Cloud storage refers to the saving of data on an offsite storing system that is maintained by a 3rd party. This is an alternative to storing data on your local storage the Internet provides a connection between your device and the respective cloud storage [8]

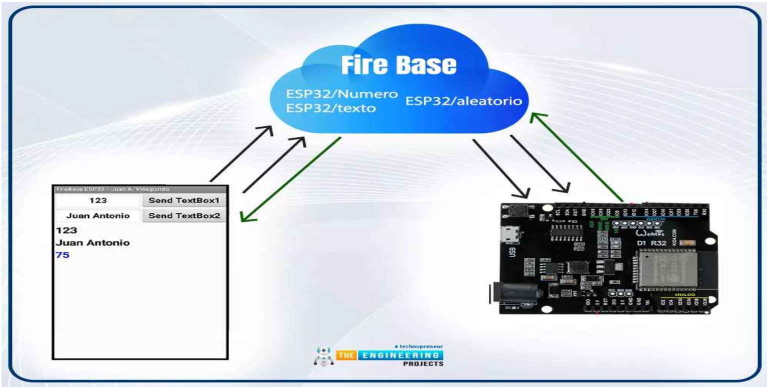

Figure 2.15: Cloud Interface Example

2.6 Mobile Application

Mobile applications play a very important part in Internet of things (IoT) based systems. They provide a platform which serves as a user interface and data visualization platform. Falling or some of the important parts that mobile application play in IoT based systems. Mobile applications are perhaps the most convenient way to access real time data from various Sensors Which are constantly feeding data to any communication module. These parameters can be extracted and viewed from a cloud server or in real time. Mobile application could also inform us about critical change in status with the help of alerts and push notifications in the case of emergencies or malfunctions. Devices can also be controlled remotely from a mobile application.

Representation of data in different forms is another feature that can be used in iot based applications with help of a mobile application. Data collected from connected devices can be stored in cloud and then displayed on the user's device which can be further used for additional analysis such as graphs and charts.

2.6.1 Kodular

With the help of the free platform Kodular, anyone may become a creator without needing to know Java or how to code. Creating an app using Kodular is similar to figuring out a puzzle made of blocks,you use creative language blocks, or code blocks. Connecting appropriate blocks is the first step in creating an app.If a function isn't compatible, you look for another block. With its simple and easy-to-use UI, Kodular is the fastest alternative for creating apps. It offers a wide range of design possibilities without the need for professional advice, even for complex applications. It does have certain restrictions and limitations.

With its drag-and-drop interface, Kodular makes app creation more accessible to a wider range of users, including non-programmers. It reduces development time and costs by allowing users to quickly test and optimize app concepts without requiring substantial code. With support for both iOS and Android, Kodular enables a wider audience reach. Its large user base also encourages resource accessibility, idea sharing, and knowledge exchange.

Chapter 3
Methodology

3.1 System Block Diagram

In the following sections we will explain the inner workings of our project with the help of block diagrams explain the basic method of our approach to completing a system that does all of the tasks mentioned in the objectives of our project.

3.2 Block Diagram

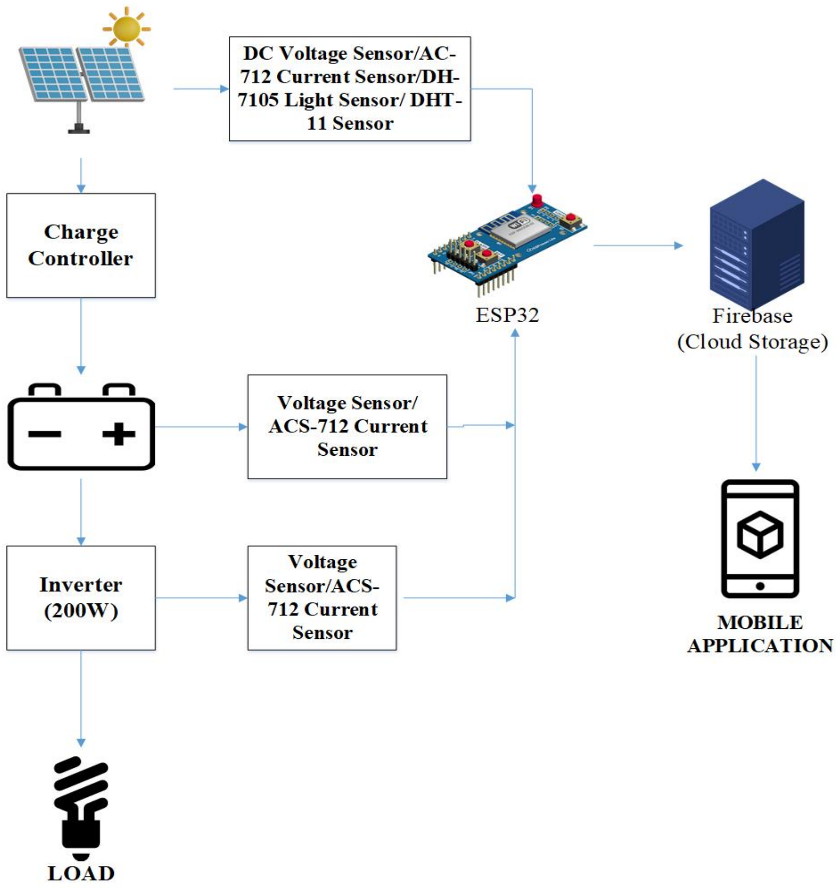

Figure 3.1: Block Diagram

3.2.1 Solar Panel

Photo Voltic Panels (PV Panels) better known as Solar Panel is a device that receives and converts Light from the sun that contains energy partials called photons which are made into a usable form of electricity. Photo Voltic Panels are comprised of several individual solar cells that are themselves made up of layers of silicon phosphorus and boron, so panels absorb the photons by doing this they initiate an electric current this results in energy generated from the photons striking the surface of the photovoltaic panels this allows electrons to be knocked out of the atomic orbit and released into the electric field generated by the solar cell. This flow of electron then induces DC or Direct current.
The panel we will use generates approximately a 30 Watts of power. That power is used to charge our systems battery after it is passed through a solar charge controller.

3.2.2 Sensing Module

1. **ACS712 Current Sensor**:
   This is a Hall Effect Current sensor. Hall Effect will convert the magnetic field created by the current that flows through this sensor to voltage that is directly proportional to it and make this sensor able to generate voltage linear with the current that flows through this it. It has three types according to the maximum range of current measurement . We will be using ACS712ELCTR-05B-T which is used where maximum current is 5 Amperes. According to the data sheet, the sensor will give 185mV for every Ampere that flows through it. The output voltage at 0A is 0.5 x VCC, which is 2.5 volts. The pin out of this sensor is very simple. It has three pins, Vcc, out and ground. The VCC is connected to the 5 Volt of the ESP32 whereas “out” is connected to the analog input pin of ESP32 that is chosen by the user.

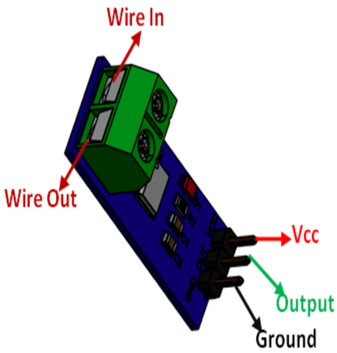

Figure 3.2: ACS712 Current Sensor

The ACS712 Provides an analog output voltage that changes according to the current that is being measured so calibration is required. Which involves measuring known current from a dummy load or power supply and adjusting the scaling factor in the code to ensure accuracy

2. **DHT11 Temperature Sensor**:
DHT11 is a digital sensor that is very low-cost sensor, and it is used for sensing temperature and humidity. It can be easily connected with microcontrollers such as ESP32 , Arduino, Raspberry pi etc. and can measure temperature and humidity instantaneously. The DHT11 sensor has four pins, and the DHT11 sensor module has three pins. The Vcc pin is the power supply that can take up to 3.5 to 5.5 volts. One pin is allocated for ground that can be provided by the microcontroller and the third is the data pin that outputs temperature and humidity via serial data communication.
DHT11 sensor has a capacitive humidity sensing element and a thermistor for sensing temperature. The capacitor has two electrodes with a moisture holding substrate that acts as a dielectric between the electrodes. When the value of the capacitor changes, it corresponds to the change in humidity value.

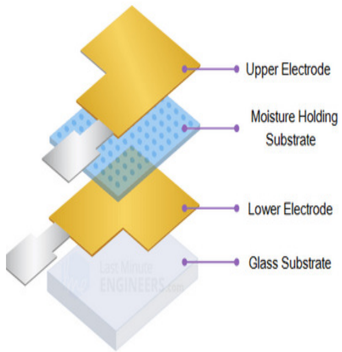

Figure 3.3: Humdity Component DHT11

A negative temperature coefficient thermistor is used to measure temperature in the sensor. With the increasing temperature, the value of the resistance decreases. It is usually made up of semiconductor ceramics or polymers so that even for the smallest change in temperature ,a larger resistance value is obtained. The temperature range of the sensor is 0-50 degrees Celsius, and the humidity range is from 20 -90%. A few substitutes for this sensor are the DHT22 sensor and AM2302 sensor.

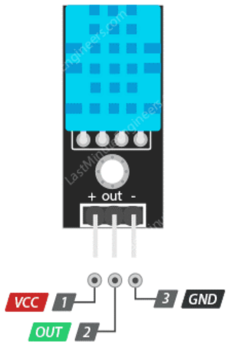

Figure 3.4: DHT11 Module

3. **BH1750 Light Sensor**:

   BH1750 is a sensor that is used to measure digital ambient light. It uses 12C communication protocol and is easy to interface with a microcontroller. A photodiode is used for the purpose of sensing. When the light falls on the PN junction of the photodiode ,holes are created in the depletion region. Due to this, electricity is produced in the photo diode. The intensity of light is proportional to this electricity that is produced. An Op-Amp is then used to change this electricity into voltage.

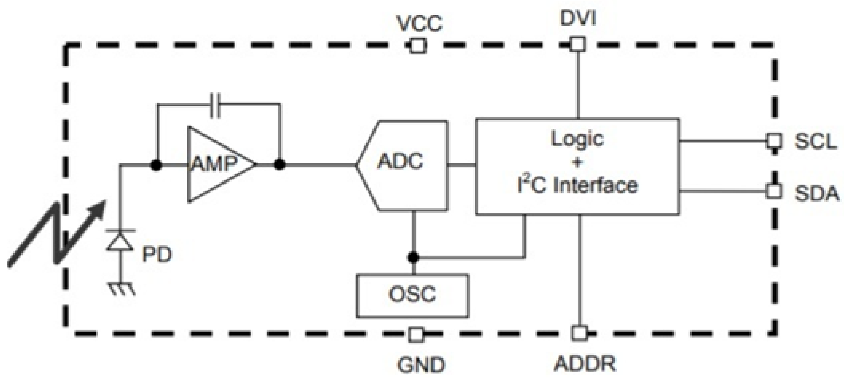

Figure 3.5: Internals of BH1750

An analog to digital converter is used to convert the voltage values powered by the Op-Amp into digital values. The sensor can work on any voltage between 2.4 to 3.6 volts. It consumes less than 0.2 mA. It can measure up to 65536 lx units . Some of the ICs that can be used as an alternative to BH1750 are TSL2561, VEML6035.

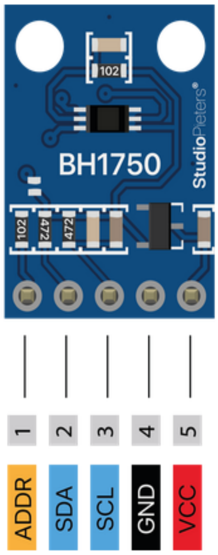

Figure 3.6: BH1750 Module

4. **Voltage Sensor**:

No specific sensor is used in our case to measure dc voltage. You can use a simple circuit consisting of two resistors connected in series. The voltage to be measured is applied across the two resistors, and the Arduino measures the voltage at the junction between them(Voltage Divider). Select two resistors R1 and R2 such that they create a voltage division ratio suitable for the voltage range you want to measure. The output voltage (Vout) can be calculated using the voltage divider formula.

Connect the resistors in series. Apply the voltage to be measured across R1 and R2. Connect the junction between the two resistors to one of the analog input pins of the micro controller. One thing to keep in mind is that most micro controllers cannot measure more than 5 volts at its analog input. So, the voltage divider must be designed in a way that when maximum voltage is applied the output at the divider junction is less than 5v. Use the analogRead() function to read the voltage at the analog pin and then calibrate it with a known voltage source.

To measure ac voltage we will be using at 230Vto 6V ac transformer. Once the voltage is step downed,a rectifier IC b125C1000 Is used to convert AC voltage to

DC voltage and the output is then given to the voltage divider, as discussed above, to measure the voltage.

3.2.3 Code Flow Explanation

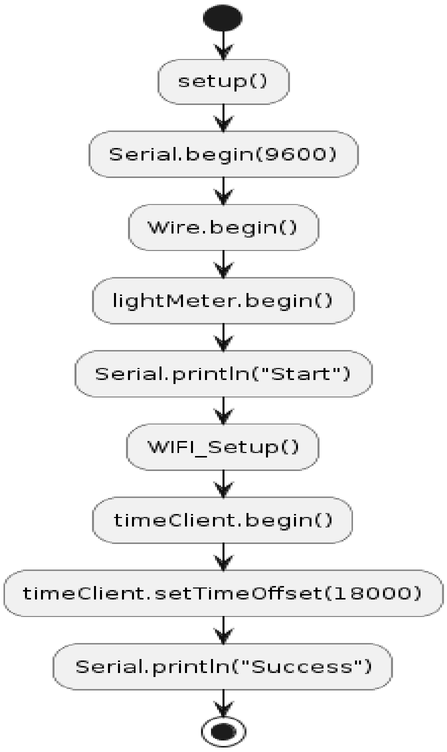

Figure 3.7: Setup Function

The setup process for the microcontroller system begins with initializing the serial communication at a baud rate of 9600 for debugging purposes. It then initiates the I2C communication protocol and the light meter sensor, likely used for measuring light intensity. The system prints ”Start” via serial communication to indicate the initial setup phase. Next, it sets up the Wi-Fi connection through a dedicated function, followed by initializing a time client to obtain the current time from an NTP server. The time client is configured with a 5-hour offset for the appropriate time zone. After successfully completing these steps, it prints ”Success” to confirm the setup’s completion, ensuring that the system is ready for operation.

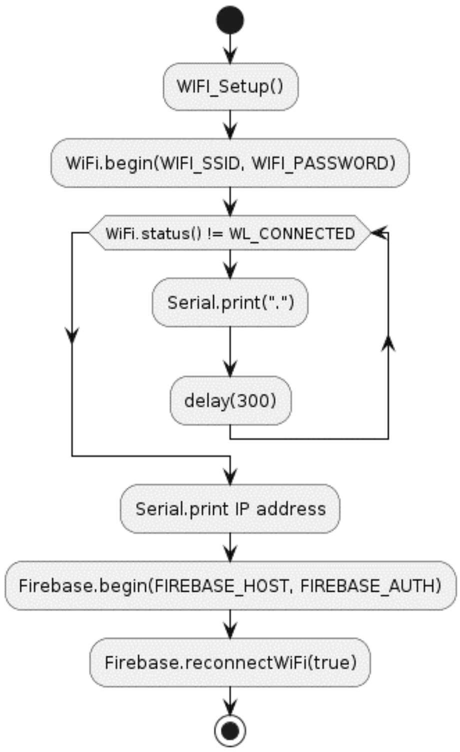

Figure 3.8: Wifi Function

This diagram depicts the setup process for connecting to a Wi-Fi network and initializing communication with a Firebase server. Initially, the system calls a function, presumably for configuring Wi-Fi settings. Following this, it attempts to connect to the Wi-Fi network using the provided SSID and password by the user. While the connection status is not yet established, it iterates through a loop, printing dots to indicate ongoing connection attempts and introducing a delay between attempts. Once the connection is successful, it prints the device's IP address. The communication with the Firebase server is then established using the provided host and login information. Finally, it sets up Firebase to automatically reestablish a WiFi connection in a situation that it is lost. After completing these setup stages, the process ends, verifying that the device is prepared to communicate with the Firebase server and transmit and receive data via Wi-Fi.

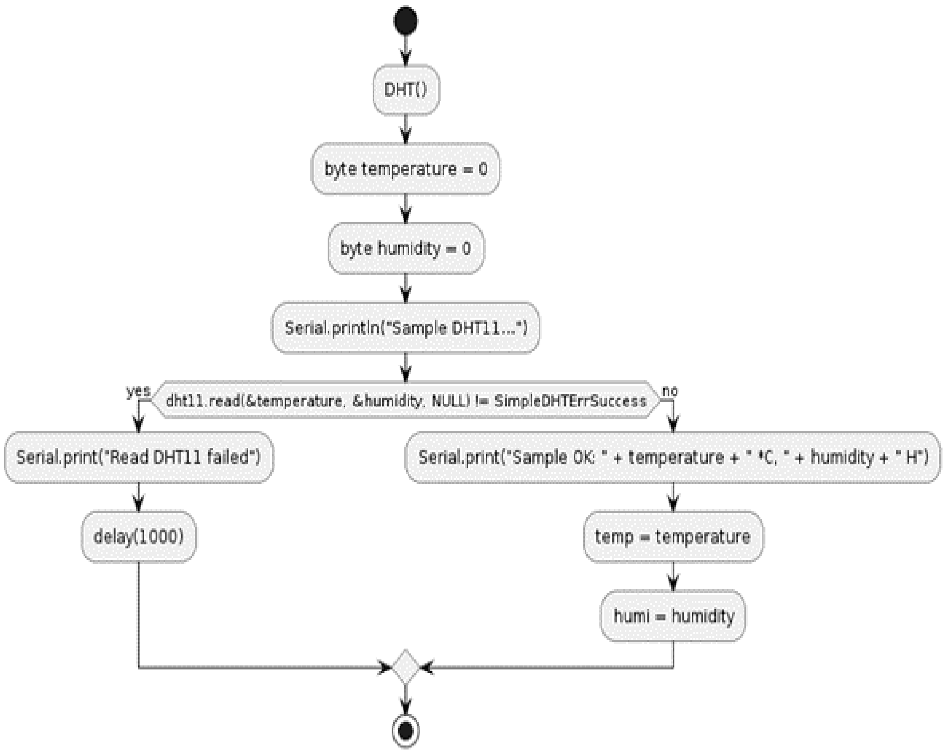

Figure 3.9: DHT Sensor Function

The sequence diagram illustrates the process of sampling temperature and humidity data from a DHT11 sensor. Initially, the DHT11 sensor is set up to start communication and get ready for data retrieval. Two variables, temperature and humidity, are then declared and initialized to store the collected readings. After then, a message indicating the start of DHT11 sampling is printed over the serial connection. It then makes an attempt to read data from the sensor. In the event that the reading failed to be successful, an error notification is displayed and a delay is added before trying again.A message with the temperature and humidity readings is printed if the reading is successful.These values are stored to the variables ”humi” and ”temp” to be used later.

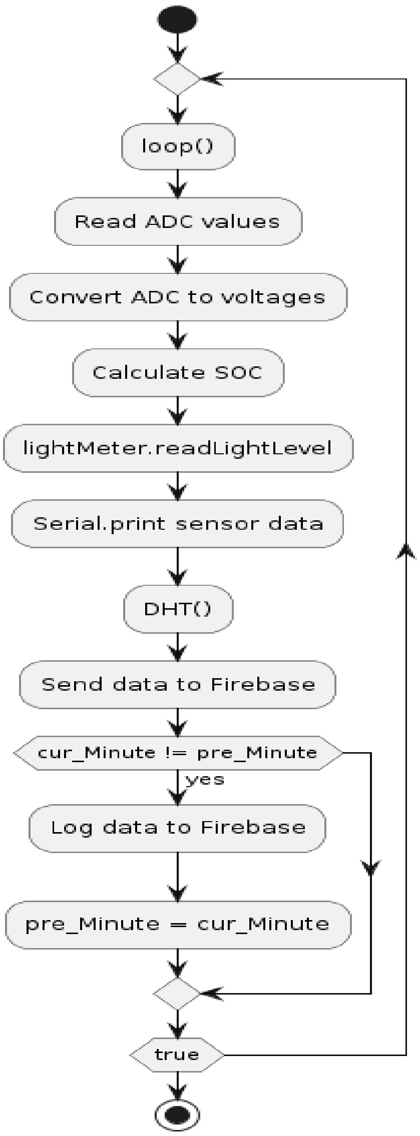

Figure 3.10: Firebase Logging Loop

The system runs in loops, gathering information from various detectors on a regular basis. This information is then stored for later use by the program. Inside these loops, the system performs several functions. It gets readings from a light sensor, estimates battery health(SOC), and converts raw sensor outputs into voltage readings. It also collects temperature and humidity data from the DHT 11 sensor. The collected data is both displayed in real-time and stored in variables to be used later by the application.. Additionally, temperature and humidity data are acquired from a DHT sensor. The collected sensor data is printed over a serial connection for real-time monitoring and then sent to a Firebase database for storage and further analysis. Furthermore, the system logs the data to Firebase at each minute interval, ensuring periodic data capture. This loop continues indefinitely, ensuring continuous monitoring and logging of sensor data.

3.2.4 Charge Controller

1. **PWM Basic Working**:
   PWM (pulse width modulation) is a technique in which the signal is modulated and the current flow between the solar panel and the battery is controlled by varying the duty cycle of the signal. Common variations of the PWM technique are 25%, 50% and 75% duty cycles which means the this is the percentage amount of time during which the signal is allowed to pass and for the rest percentage the signal is blocked. For example, the 25% duty cycle means that the for 25% of the signals time its current is allowed to flow and the remaining 75% of the signal is blocked. The main motivation behind using such a method to charge the battery rather than directly charging it is to maximize the efficiency of the overall charging of the battery and to reduce as much overheating as possible to increase the overall life of the battery.

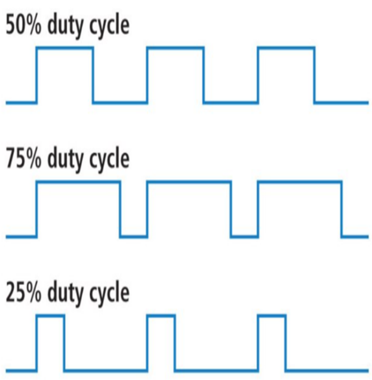

Figure 3.11: PWM Duty Cycles

2. **Arduino Nano for PWM**:

Arduino board is a better option as compared to micro-controllers as its much more user friendly and easy to code and gives the user a much more precise controller over the percentage modulation that is to be done on the signal. Arduino boards are also easily integrated with other sensors. In our circuit the Arduino board is being used to control the duty cycle of the signal for optimum battery charging and this is done through the code that we have fed into the board which is constantly measuring the voltage that is being received for the solar panel and the battery and then adjusts its PWM accordingly. Another benefit of using the Arduino board is its integration with our current sensor (ACS 712) and the LCD that we have used for the display of the information the board is receiving.

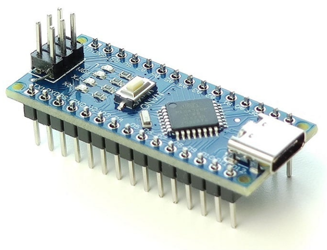

Figure 3.12: Arduino Nano

3. **Buck Converter:**
   The buck converter is used to prevent the high voltages of the solar panels from harming the battery health. Solar panels often produce a higher voltage than needed to charge the battery, so we use a buck converter to reduce the DC voltage to a suitable value.
   The buck converter works by changing the switching frequency to reduce the power transmission. Duty cycle is varied by using a variable resister and a transistor. The inductor and capacitor are used in this reduction of voltage as in the first cycle the inductor is charged and in the second cycle the inductor is discharged, and the capacitor is charged which is then discharged by a load resister.

This charging and discharging between 2 cycles are done by the help of the diode.

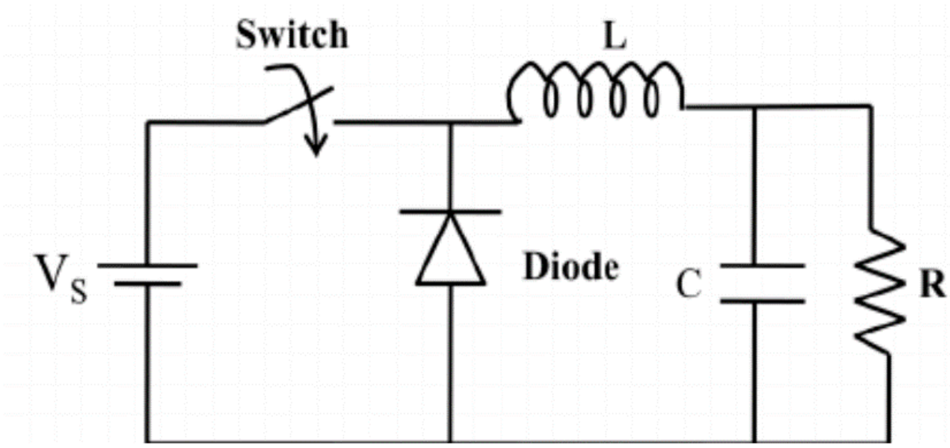

Figure 3.13: Buck Converter

4. **LCD Display**:

   The lcd is being used to display the voltages being sensed of the solar panel and the battery. It will also display the amount of charge the battery has.

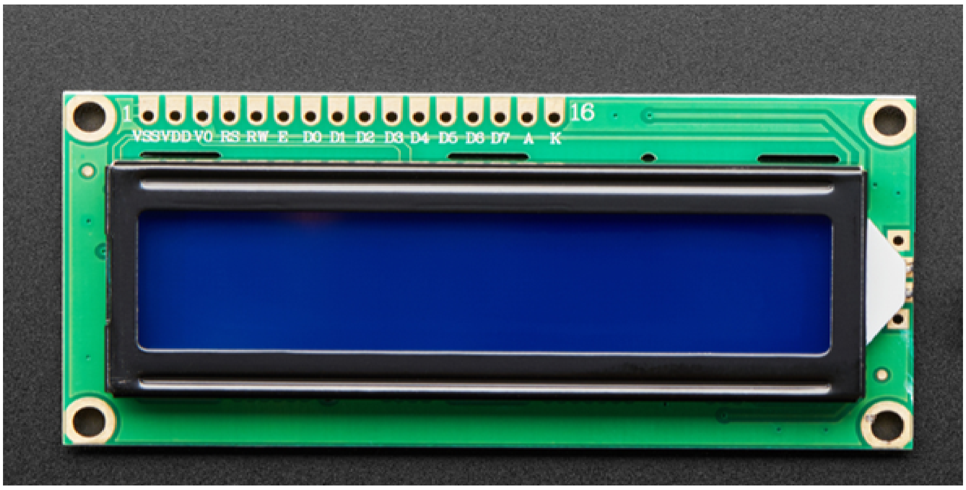

Figure 3.14: LCD

5. **Code**:

   The first step in the code section is to import the libraries required for LCD functioning and I2C connectivity. It initializes the PWM pin, which controls the buck converter, and the variables that measure the voltages of the solar panel and batteries. For PWM control, predetermined duty cycle values are set. Backlight activation is done along with LCD initialization and setup.

The code then sets up pins for PWM output and voltage input, centers the LCD cursor, and shows the first few messages. It displays the voltages from the solar panel and batteries on the LCD after calibrating and reading them. A variety of charging scenarios are carried out with corresponding PWM modifications based on the voltages of the batteries and solar panels.

In the first case, charging is stopped and a ”Battery Charged” message with a 100% charge indication is displayed if solar voltage exceeds battery voltage and battery voltage is over 13V.

In the following cases, battery voltage thresholds determine the different PWM levels that are used for charging, along with LCD feedback. Depending on the condition of the battery, PWM changes assure effective charging; stronger PWM speeds up charging for lower battery voltages. A ”Battery Dead” message is displayed if the battery voltage drops below 10V, suggesting that it needs to be replaced or recharged.

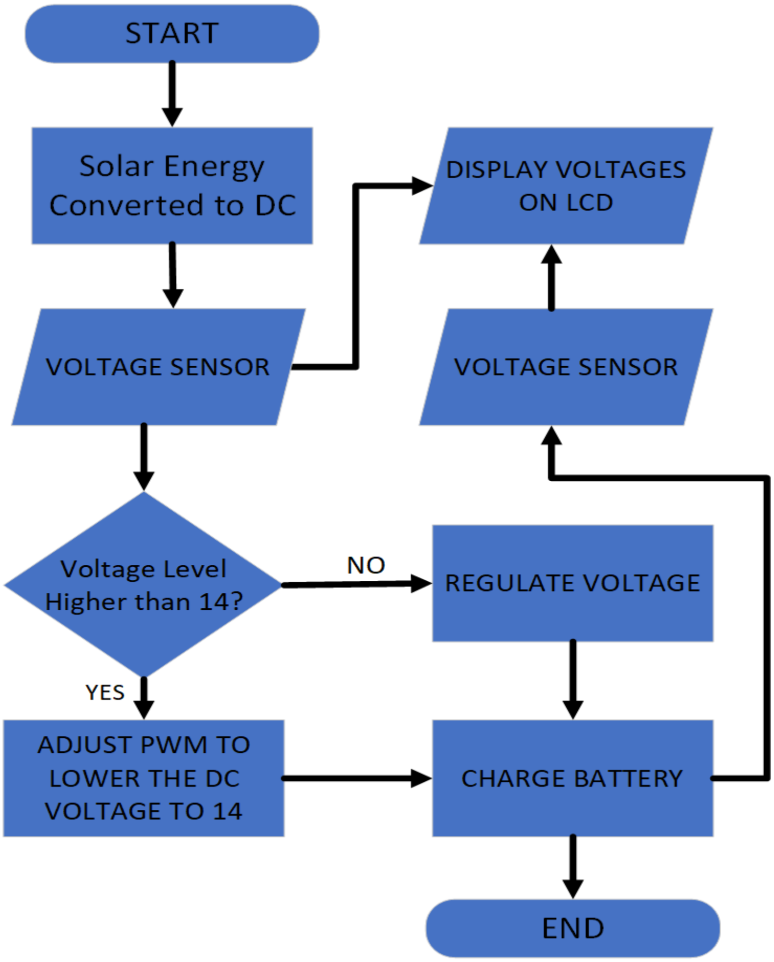

Figure 3.15: Charge Controller Code Flow Diagram

In the first case, charging is stopped and a ”Battery Charged” message with a 100% charge indication is displayed if solar voltage exceeds battery voltage and battery voltage is over 13V.

In the following cases, battery voltage thresholds determine the different PWM levels that are used for charging, along with LCD feedback. Depending on the condition of the battery, PWM changes assure effective charging; stronger PWM speeds up charging for lower battery voltages. A ”Battery Dead” message is displayed if the battery voltage drops below 10V, suggesting that it needs to be replaced or recharged.

3.2.5 ESP32

ESP32 is the main communication device. It is collecting the required parameters from respected sensors and send the resulting parameters to the cloud-based server over wireless connection. These parameters include voltage and current readings, battery temperature and voltages and Inverter voltages. Further the data coming from ESP32 is uploaded to Cloud using the Wi-Fi module embedded on esp32.

3.2.6 Firebase(Cloud Storage)

The ESP32 will be used to upload the received data for example, the current and voltage from the respective sensors and upload them to Firebase is a mobile and web app development Platform which was founded by Google it provides a real time database that means it can update values in real time it also provides a secure and easy to implement authentication system. Firebase can also be used for cloud storage.

3.2.7 Inverter

This article talks about the construction of a 220V AC inverter that can run a load of up-to 200 Watts using the IC CD4047 as the oscillator furthermore comprising of BJTs, MOSFETs in PushPull Configuration, Transformer and 12V DC 7Ah Battery.

1. IC CD4047: A flexible CMOS integrated circuit, the CD4047 produces accurate timing signals for a range of uses. It may generate a steady square wave output at a frequency set by external components when set up as an astable multivibrator. A 50 Hz waveform, which is frequently used in power systems and some kinds of electronic equipment, can be produced using the CD4047 by carefully choosing the resistors and capacitors in the circuit. For applications needing a steady 50 Hz waveform, like frequency converters in industrial settings or inverter circuits for AC power generation, its affordability, dependability, and simplicity make it a popular option.

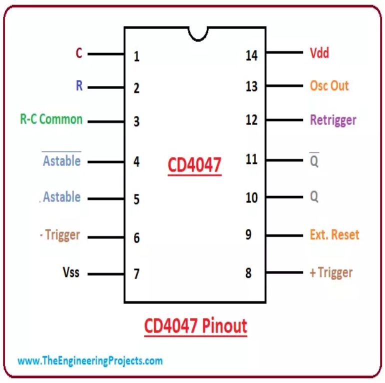

Figure 3.16: CD4047 Pinout

The purpose of this IC in our case is to generate a Pulse width Modulation (PWM) wave that is used to drive 6 of the mosfets on opposite ends at one time. 2 mosfets at a time run at 50Hz. each is out of phase from the other phase to avoid short circuiting.

The Mosfet Q1 and Q2 are turned on in the first cycle, Mosfet Q3 and Q4 are turned on in the second cycle, and then Q5 and Q6 is turned in the final cycle. The square wave is of 50 Hz drives all the mosfet. This is done out of phase. the 50 Hz wave is then sent to the base of the BJT. Whenever, the base receives a 5v input. It allows the BJT to flow the 12v at its collector to its emitter. this gives us a 12 volts 50 Hz square wave which is used to bias the gates of the mosfet.

2. Push-Pull Configuration: A MOSFET-based push-pull inverter is a kind of power inverter that uses MOSFETs organized in a push-pull arrangement to convert DC input into AC output. This setup is frequently utilized in many different applications, such as motor drives, UPSs, and renewable energy systems. Two MOSFETs are used in this arrangement, each of which functions as a switch to alternately push and pull current through the load, which is typically a transformer. These MOSFETs are turned on and off in a complimentary way by the inverter during operation, creating an AC waveform across the load. The output waveform's frequency is determined by the MOSFETs' switching frequency, and careful design makes sure that it matches the intended frequency, such as the typical 50 or 60

Hz for mains electricity.

The efficiency of a push-pull inverter based on MOSFETs is one of its main advantages. MOSFETs are renowned for having low on-state resistance and fast switching speeds, which minimize power losses while operating. In applications like solar or wind power systems, where energy conservation is of the utmost importance, this efficiency is crucial for optimizing the overall performance of the system by maximizing the conversion efficiency from DC to AC. Furthermore, in contrast to other inverter topologies, the push-pull topology naturally yields both positive and negative half-cycles of the output waveform, producing a symmetrical AC output with lower harmonic distortion.

Push-pull inverters based on MOSFET technology provide flexibility and scalability in addition to efficiency and dependability. By changing the quantity and rating of MOSFETs and other circuit components, they are easily scalable to meet varying power requirements. They can be used in a variety of settings, from high-power industrial systems to small-scale electronics, thanks to their scalability.

For a variety of applications, a push-pull inverter with a MOSFET base is a reliable and effective way to convert DC power to AC power. Designers and engineers looking to create power conversion systems that fulfill strict performance criteria while minimizing energy consumption and maximizing system reliability frequently choose it because of its high efficiency, dependability, and scalability.

3.2.8 App Development

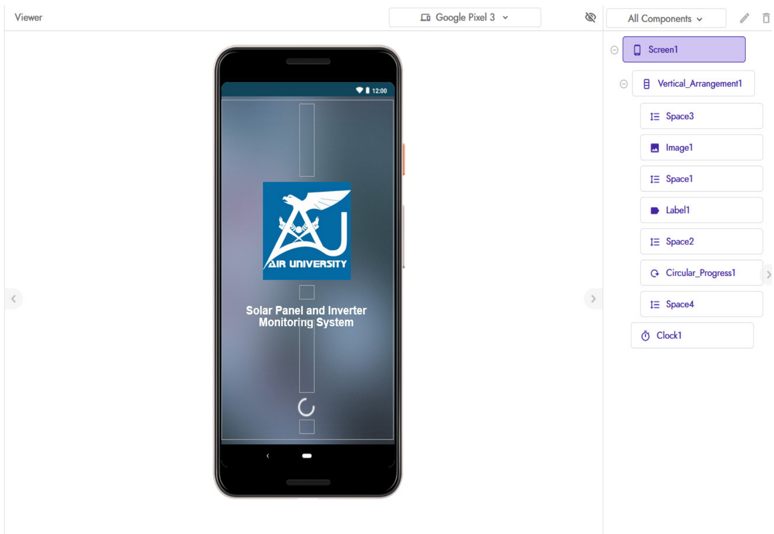

Figure 3.17: Splash Screen Front-end

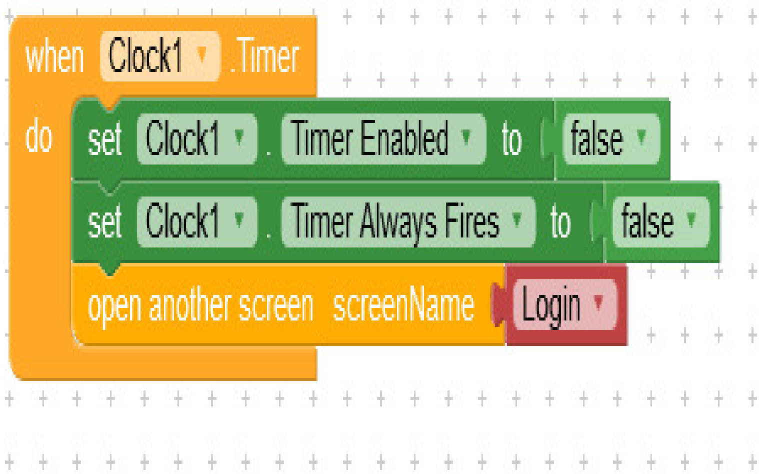

Figure 3.18: Splash Screen Back-end

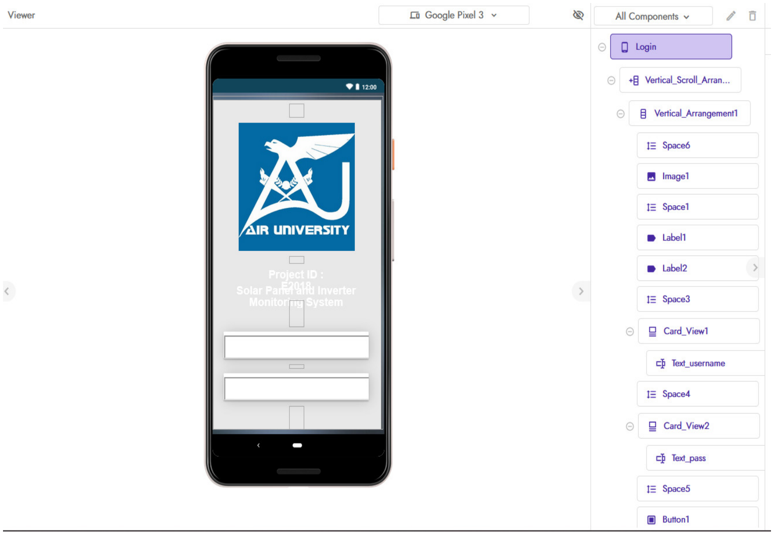

Figure 3.19: Log-In Front-end

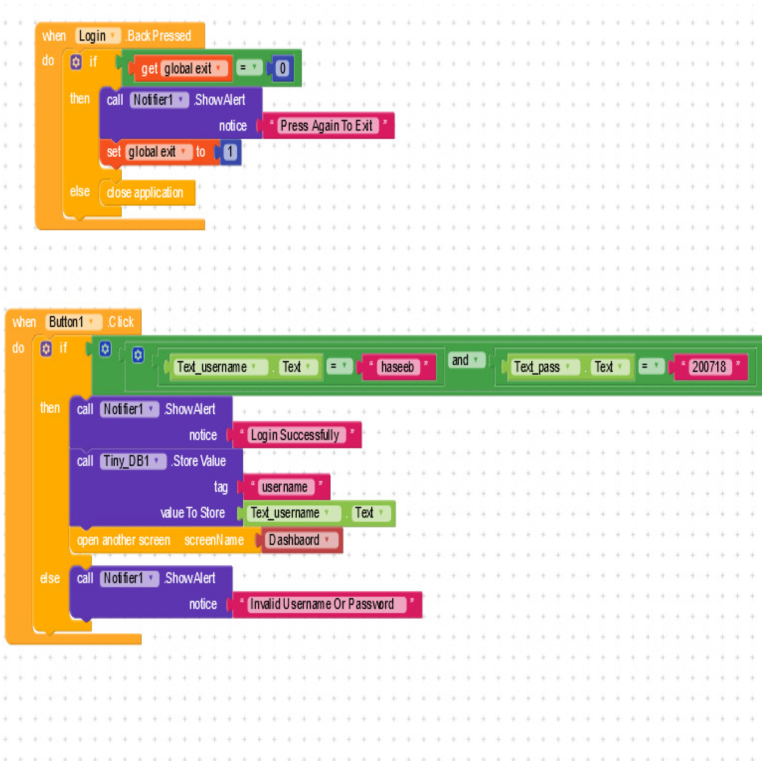

Figure 3.20: Log-In Back-end

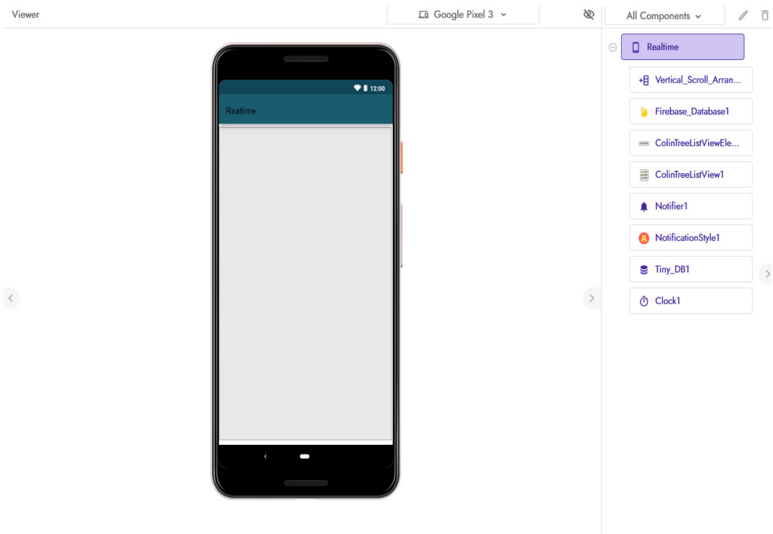

Figure 3.21: Real-Time Front-end

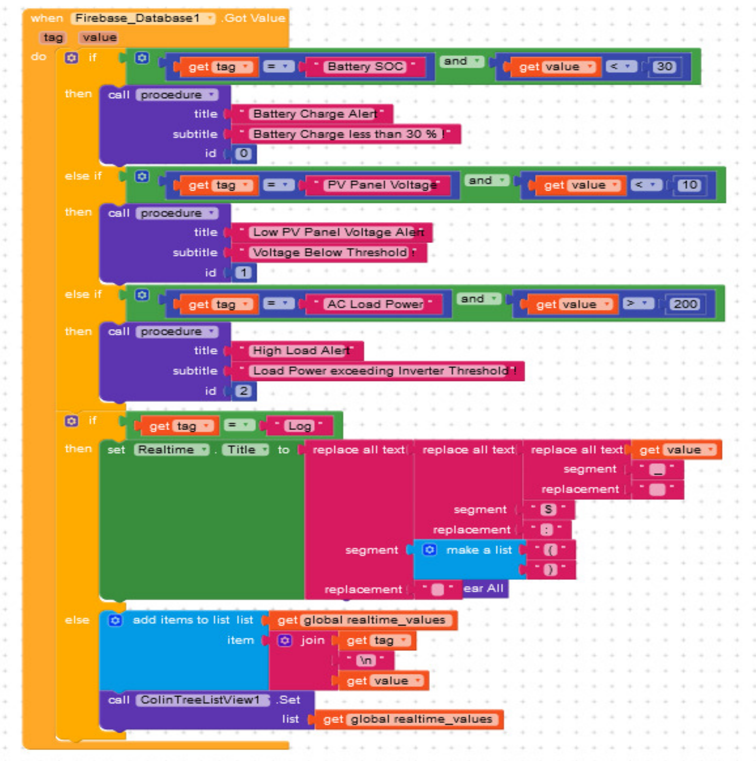

Figure 3.22: Called Alert Function

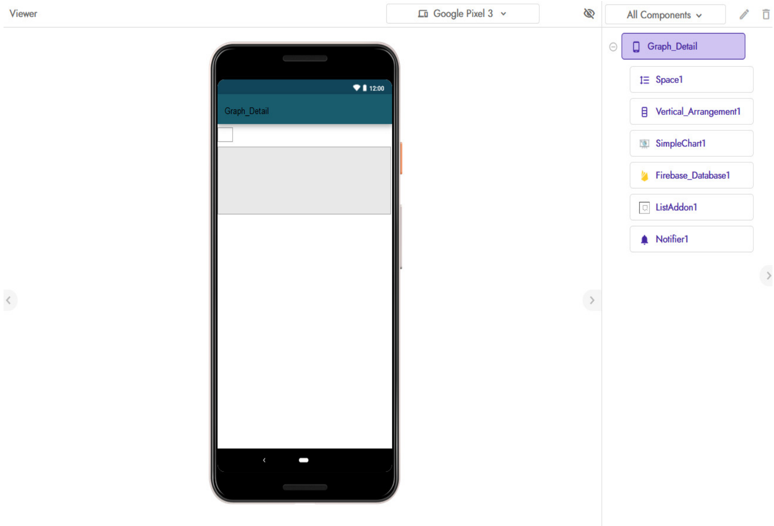

Figure 3.23: Graphical Display Front-End

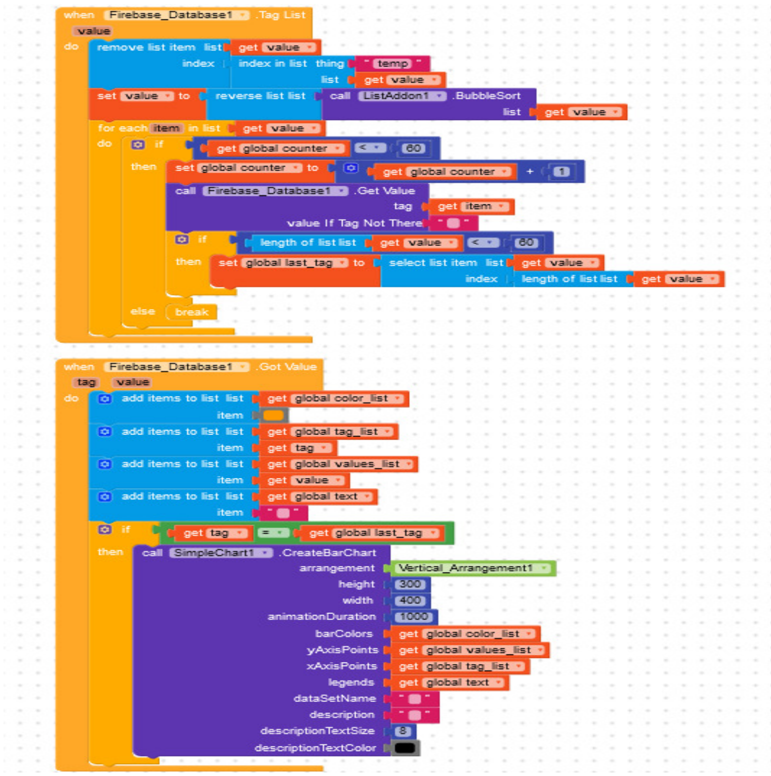

Figure 3.24: Graphical Display Back-End

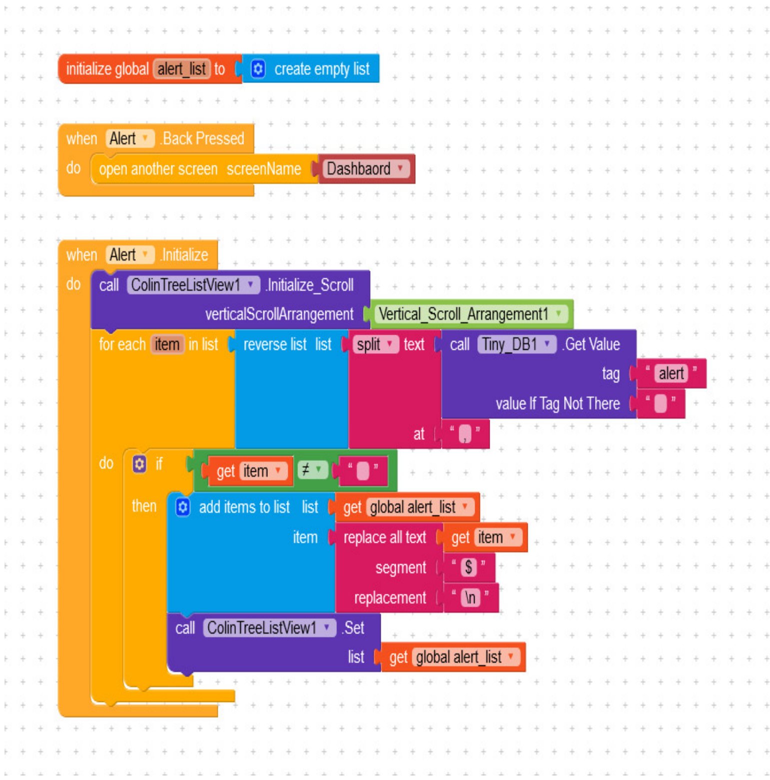

Figure 3.25: Alert Screen Back-End

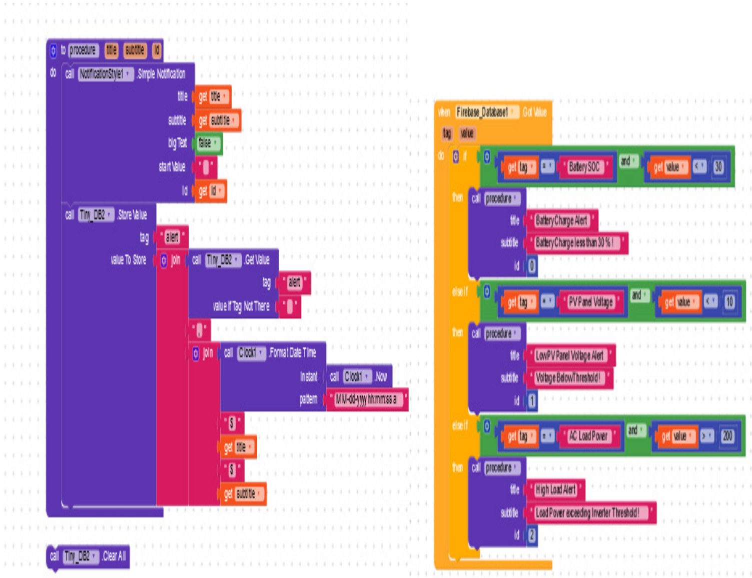

Figure 3.26: Alert Function

Chapter 4

Results

4.1 Designed Circuit

Complete Hardware circuit is displayed in the following image, it includes, PV panel, Charge Controller, Batter, Inverter, Sensing Module. The following system is capable of running a 200 Watt AC load.

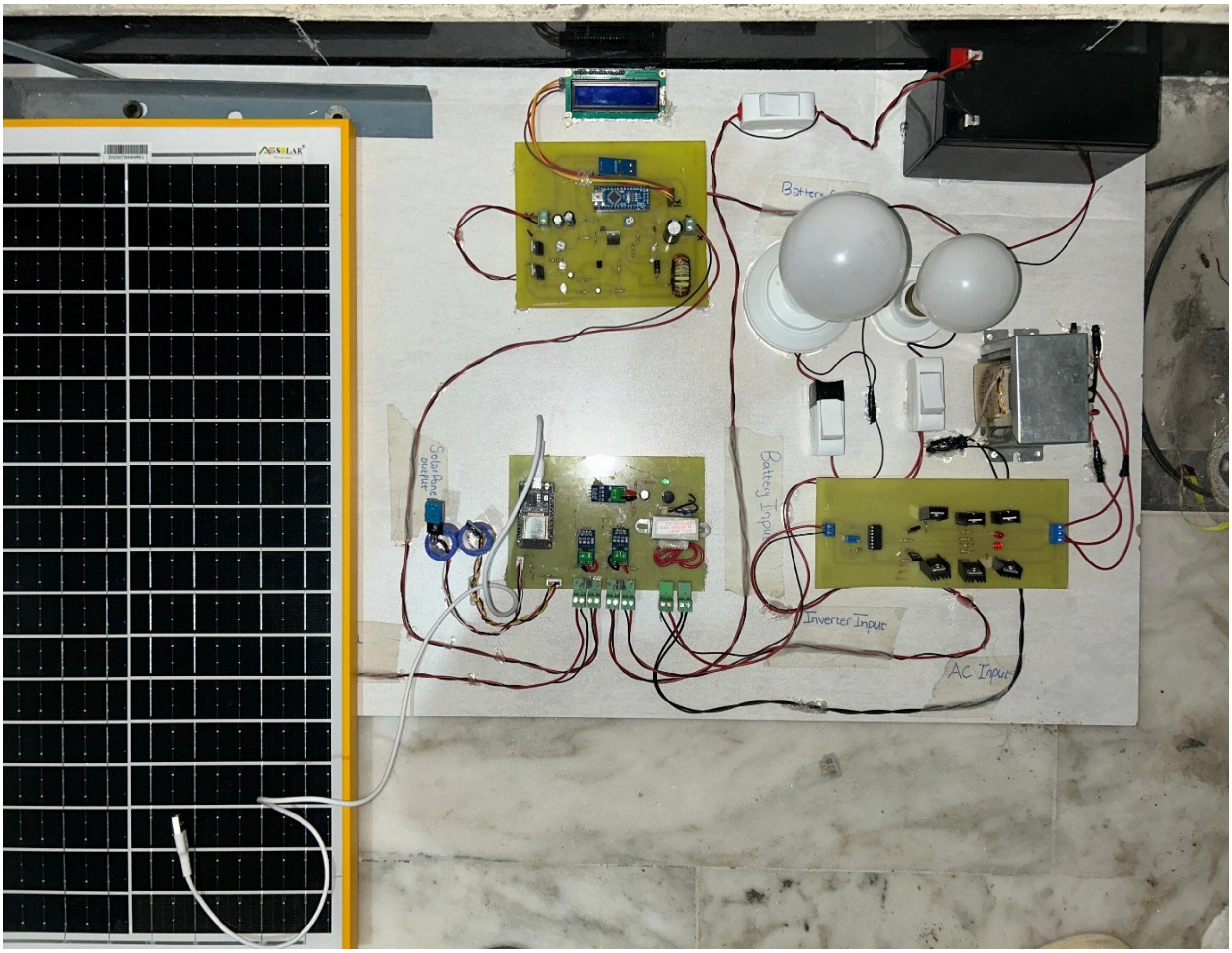

Figure 4.1: Complete Hardware

4.1.1 Charge Controller

Our Charge control includes a buck converter and lcd display the Arduino is used to provide pwm to the buck circuit and also interfaces with the lcd.The Arduino is also responsible for logging and displaying the voltage of the battery and the panel.This is done through a voltage divider circuit with this combination we can assure the battery recieves sufficient amount of voltage by reducing the excess voltage from the solar panel to 14 volts which is optimum for the battery's charging state.

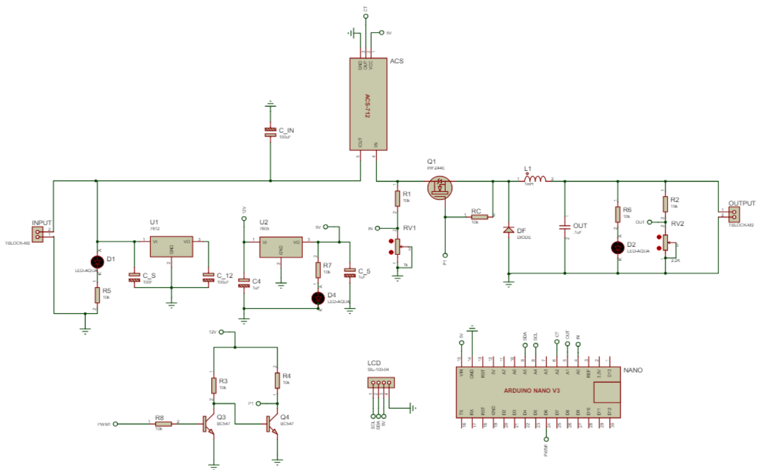

Figure 4.2: Charge Controller Circuit

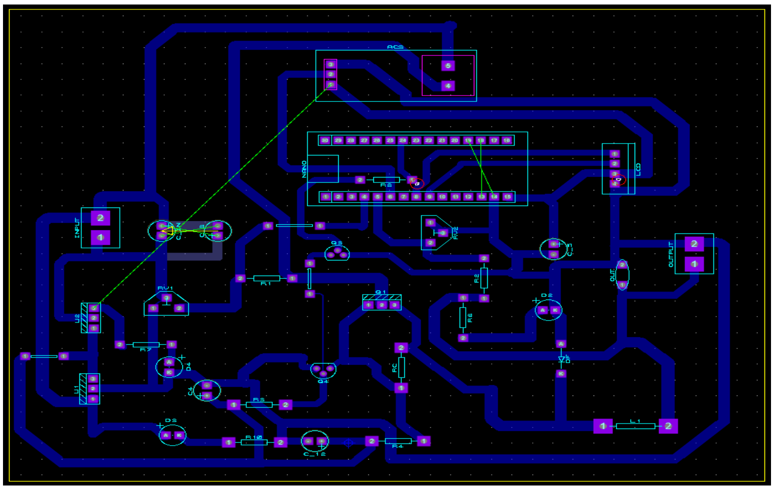

Figure 4.3: PCB Design

4.1.2 Inverter

An essential component of ”localized” or off-grid electricity systems is the inverter. A lot of rural places are frequently cut off from the electrical grid. Their everyday home appliances are powered by energy sourced from solar panels and batteries. Similar to this, even in cities Uninterruptible power supply (UPS) systems are widely used in homes. Home appliances that depend on AC electricity are powered by an inverter in the event that regular AC power is interrupted or disconnected from the grid for any reason, known or unknown. The use of electrical cars in transportation is growing significantly. AC motors are used in electrical vehicles to power them. However, since dc batteries are the primary energy source in these cars, an inverter is required.

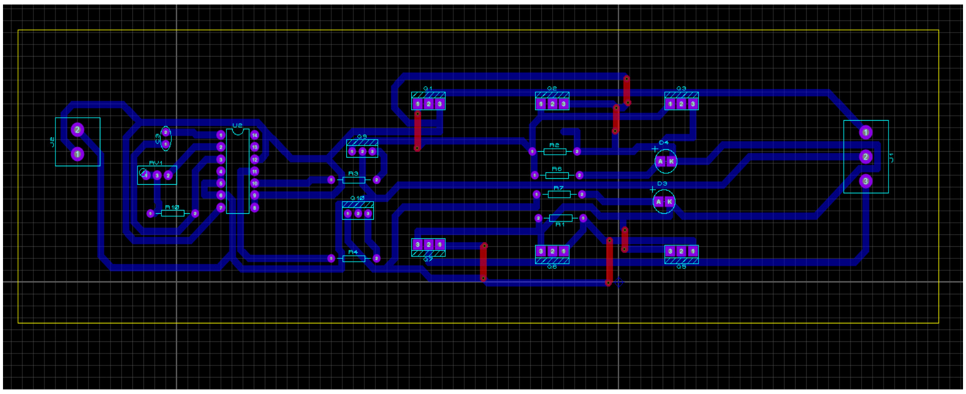

Figure 4.4: Inverter PCB

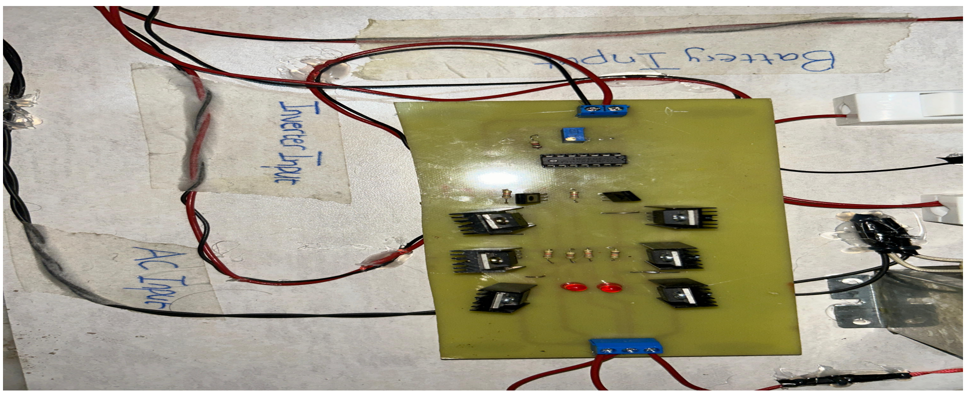

Figure 4.5: Inverter Circuit

We've used Mosfet IRF 250 a fixed pwm of 50HZ in a combination of voltage driver circuit to create an inverter. this inverter is a push pull inverter that can generate upto 200 watts of power that can be used to power home appliances. A Square wave is generated of 12 volts. this 12 volt square wave can be stepped up to 220v using a power transformer.

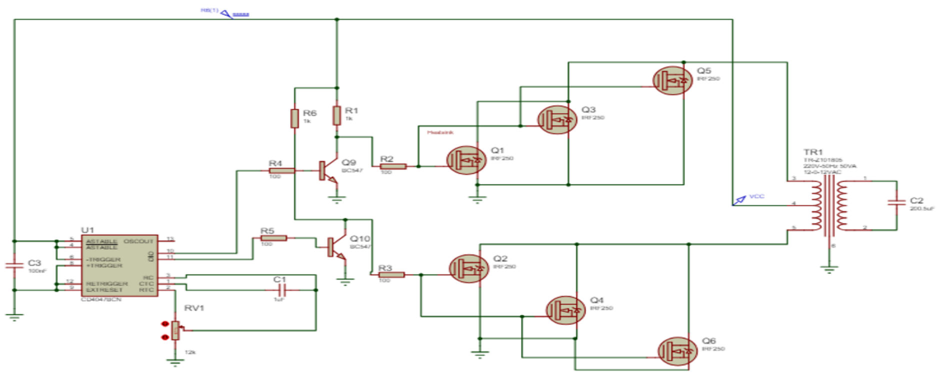

Figure 4.6: Inverter Circuit

4.1.3 *Sensing Module*

There are 3 Voltage Dividers used in this circuit .The output taken from these dividers is fed to the analog pin of the esp32.V1 is used to measure the PV Panel Output Voltage.V2 is used to measure the Battery Voltage.

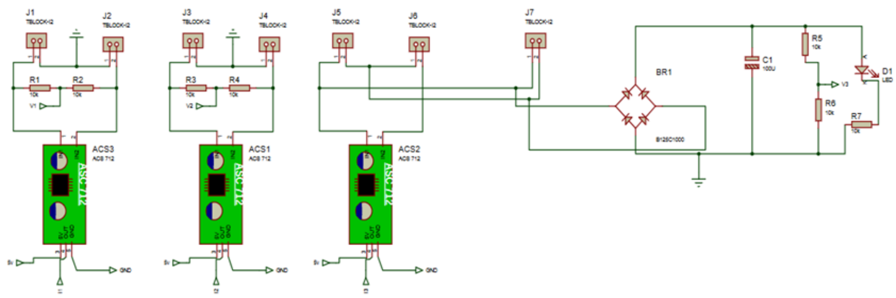

Figure 4.7: Sensing Module Circuit

V3 is the AC voltage which is actually measure in the same way as v1 and v2 by rectifying it and converting to dc thorough a step down transformer and a rectifier IC.

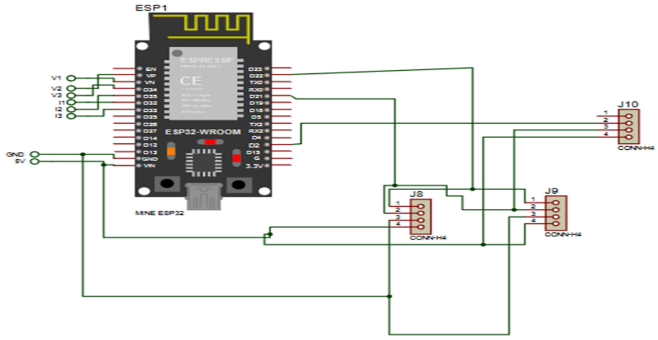

Figure 4.8: ESP32 Interfacing

Similarly three ACS712 are connected to Pins 25,32 and 33 which measures Solar Panel Current , Battery Current and Ac Load current Respectively. BH1750 Light sensor and DHT11 Temp/Humidty sensor is also connected to the ESP32.

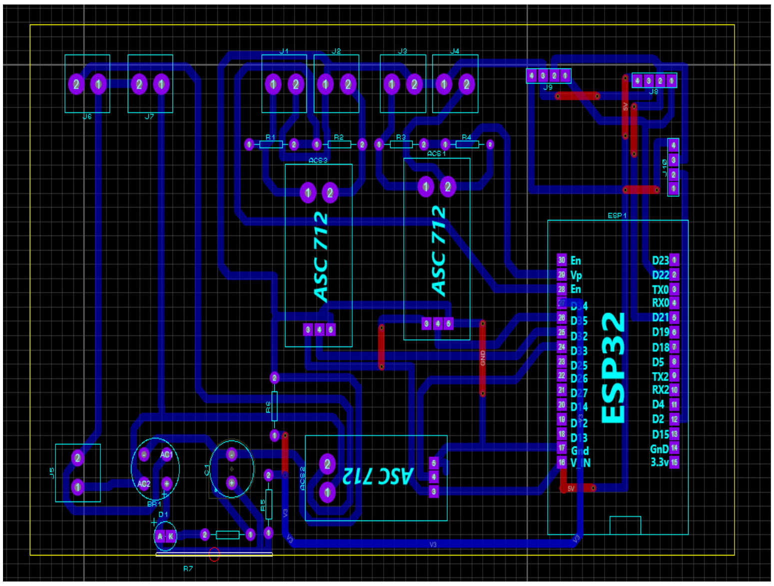

Figure 4.9: Sensing Module Circuit

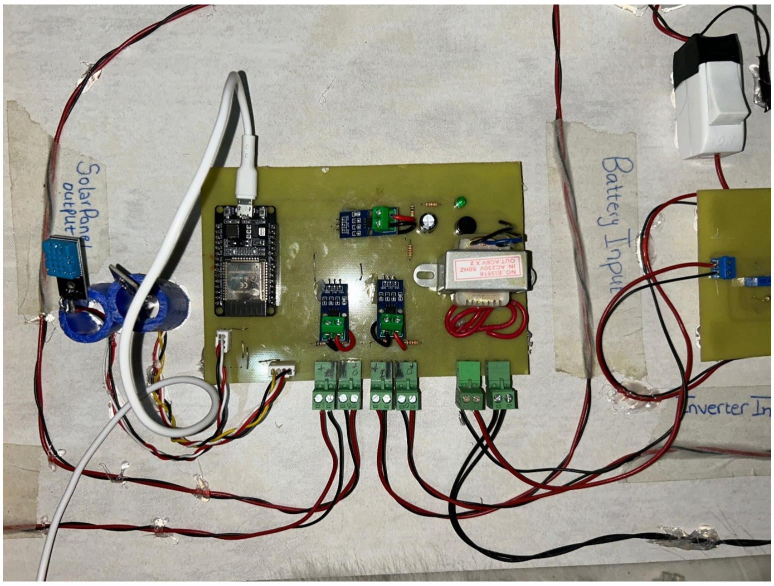

Figure 4.10: Sensing Module Circuit

4.1.4 Fire-Base (Cloud Storage)

The ESP32 is coded in a way as soon it is given power it searches for the predefined wifi network that the user has provided.It will stay in a loop until it is given access to it.As soon as wifi access is granted it establishes a connection with firebase and starts storing data to the cloud.

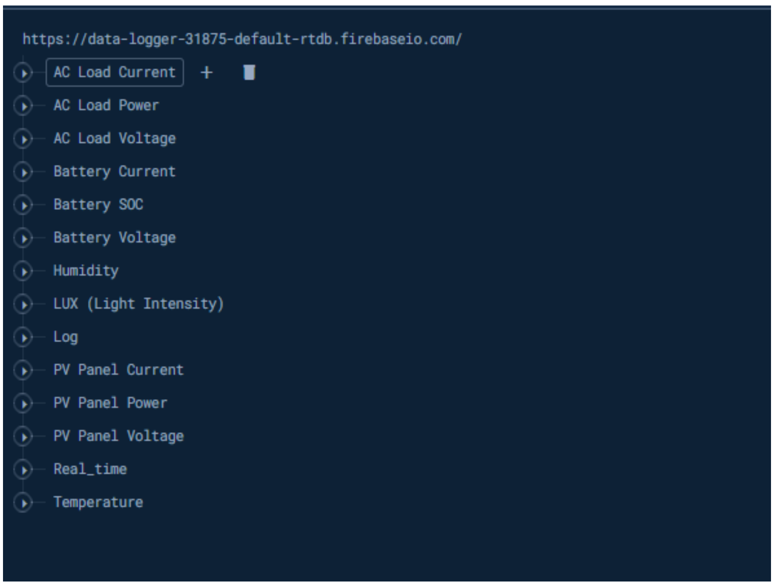

Figure 4.11: FireBase Log

The Above figure shows the database created once ESP32 is turned on. The “Real_time” entry in the data base has all the parameters inside it and they change instantaneously with the behavior of the system.

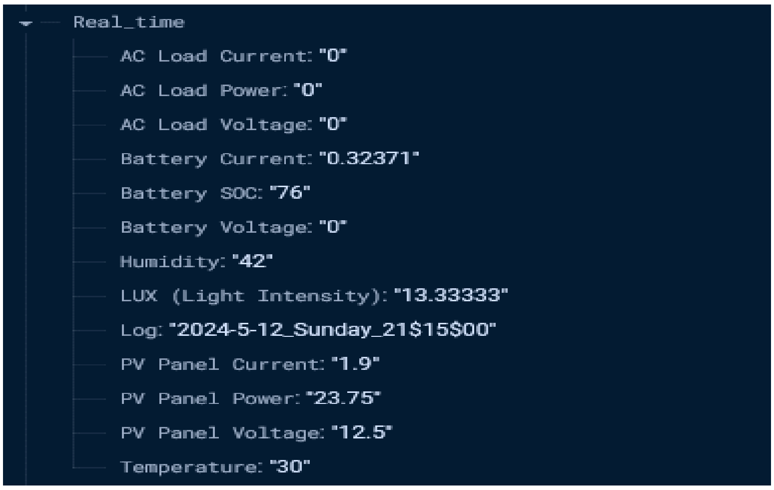

Figure 4.12: Real Time Log

The “log” entry is storing time of each entry which is stored in the other sections.We have coded in a way that it adds a new value in the log every 1 minute.These timestamps are stored in the variable log.

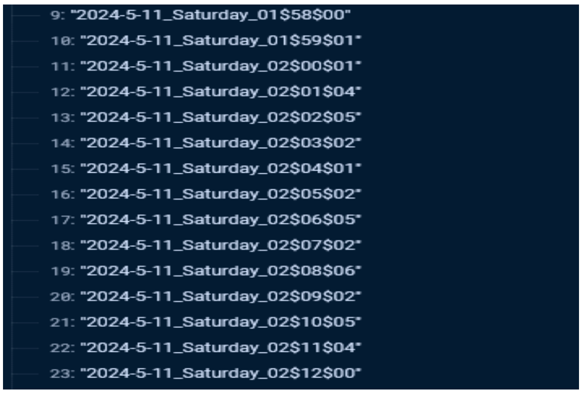

Figure 4.13: Date Log

So now the data is successful being sent to the cloud.The database is ready and the mobile application can now easily fetch any value from any entry as per the requirements and programming that is done.

4.1.5 Mobile Application

1. The login page:

   Before the Login Screen there is a three second animation.The login Page is a simple page with predefined usernames and password.Project title and ID is displayed along with the University Logo.

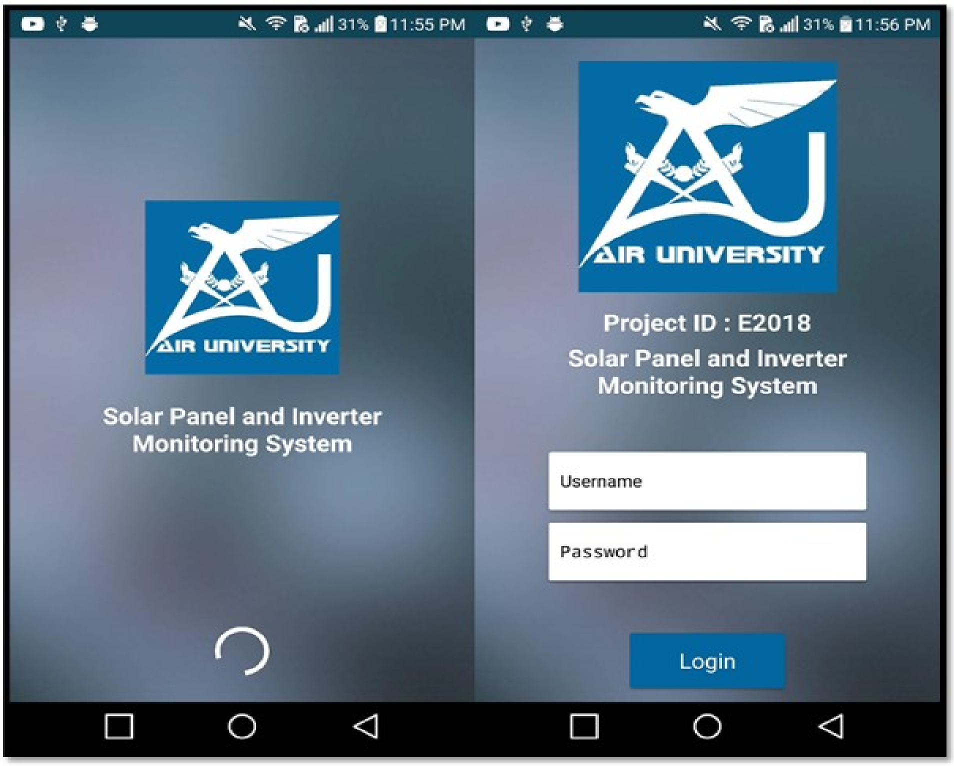

Figure 4.14: Log-in Page

2. Dashboard:

   Once correct username and password is entered.The user is directed to the dashboard.The dashboard displayed four options to the user about the next screen that the user wants to go.The four screens are Real Time , log , Graph and alerts. On the top the username is displayed and a button for logout is there so that if the user wants to exit he can use that.

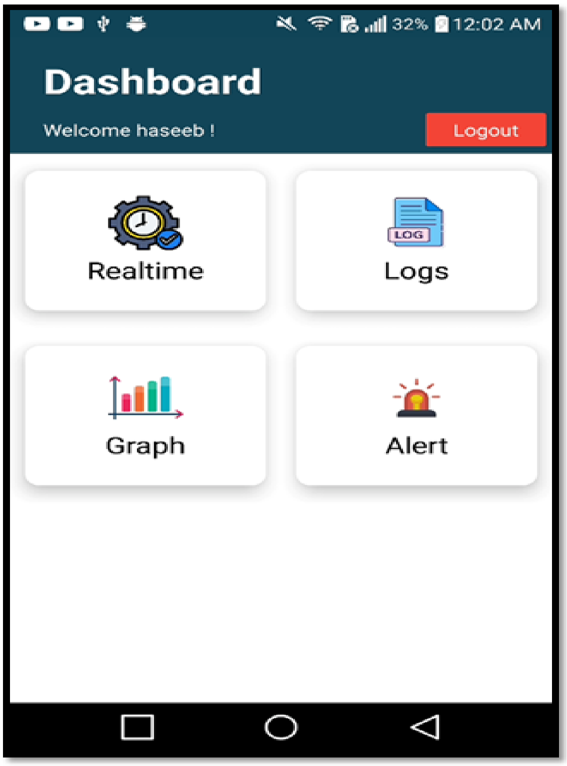

Figure 4.15: Log-in Page

3. Real-Time Data:

   The Real time screen shows the all the variables that are sensed using the ESP32.This screen fetches the values from the “Real_Time” entry in the database which was explained above.On the top of the screen .The day , date and time is displayed which is also present in “Real_time”. As soon as the ESP32 is turned on it starts to send real time data to firebase and it is simultaneusly updated in this screen of the application.

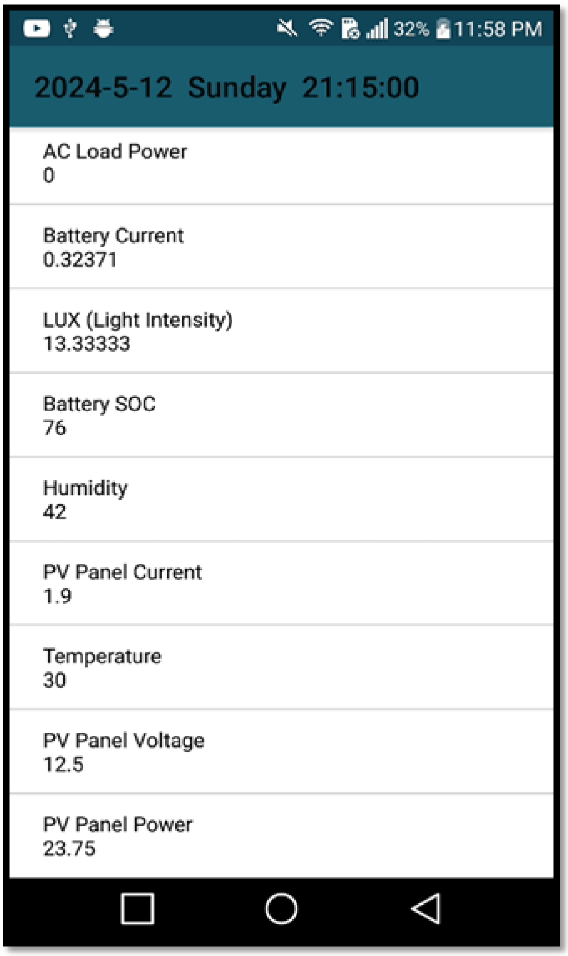

Figure 4.16: Real-Time Data Display

4. The Log Screen:

   Once the user selects log on the dashboard.It is directed to the log screen which itself is a prerequisite to another screen.The log shows the last ten values of any variable that the user wants to check.The user selects the variable and then it is directed to another screen where the values along with their day and time.If the user stays on this screen for too long and now wants to see the updated last ten values. The user will simply swipe on the right or left side of the screen to go back and click that variable again.Now the updated values will be shown to the user.

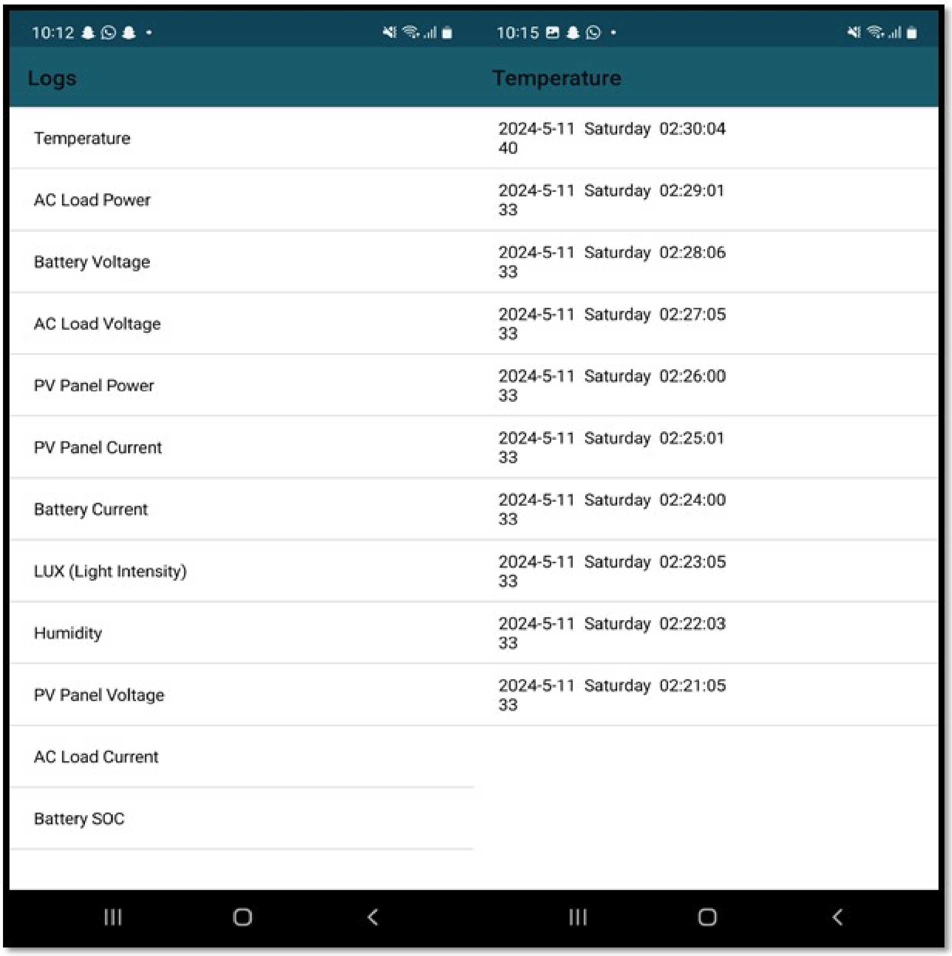

Figure 4.17: Real-Time Data Display

5. Graph:

   The graph screen initially show four options as we will be generating four graphs.These include temperature, Battery State of charge, PV Panel Voltage and Ac load Power.Once the user selects which graph is required ,it directs to another screen where the graph is shown.The graph is a bar graph.It searches in the database under the log entries.It extracts the last 60 values from the log and their time and generates the graph.If the values are less than sixty indicating that the system is turned on not too long ago and the graph is generated of this values only.

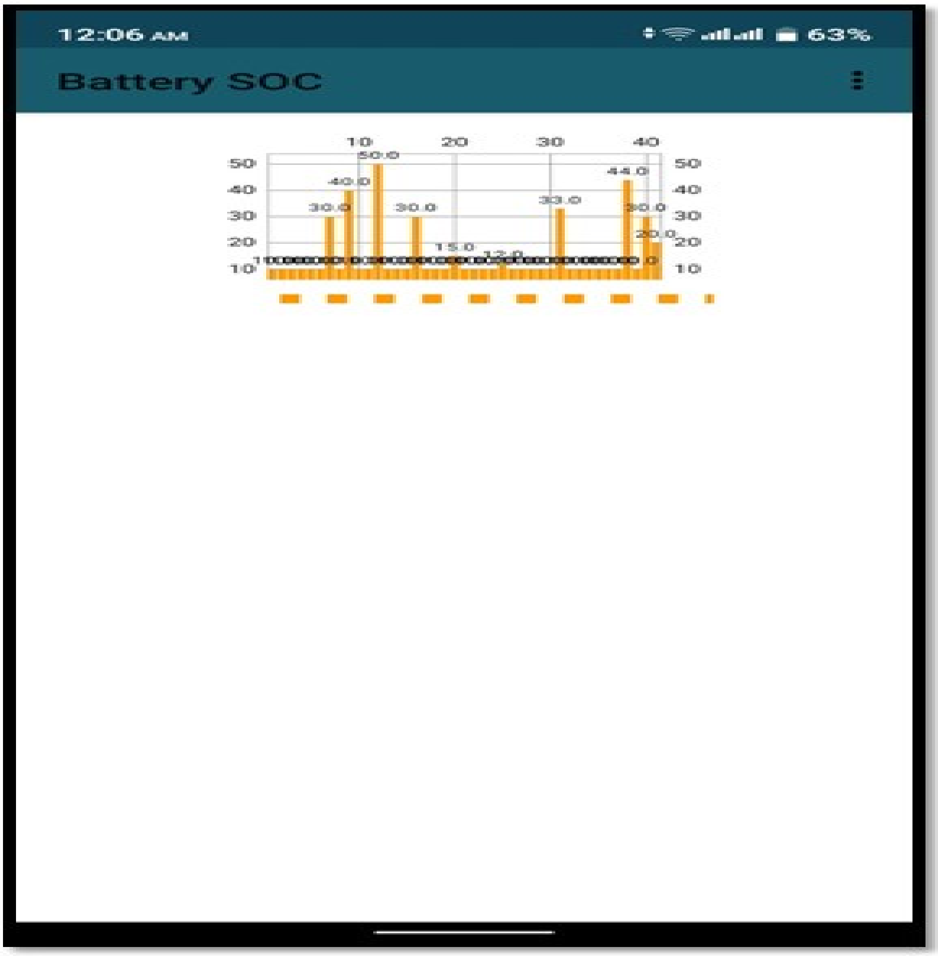

Figure 4.18: Graphical Display

6. Alerts:

   The Alert screen shows a history of alerts that were given on the application.It includes the exact day and time of when the alert was generated.There are three alerts that can be given to the user .The check on variables for there alerts are done from the”Real_time” database.The following are the conditions for the alerts.

   If SOC < 30

   ”Battery Charge Alert:Battery Charge less than 30 %!”

   If PV Panel Voltage <10

   ”Low PV Panel Voltage Alert: Voltage Below Threshold!”

If AC Load Power 200

”High Load Alert: Load Power exceeding Inverter Threshold!”

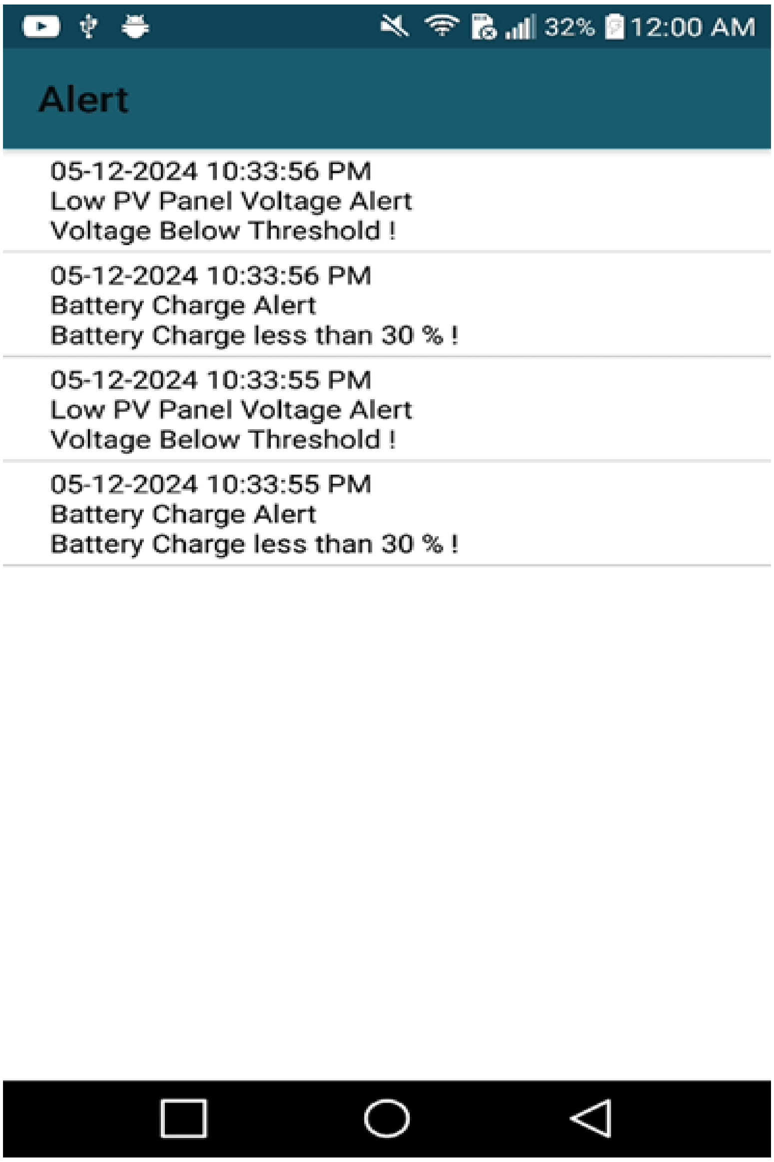

Figure 4.19: Alert Page

Chapter 5
UN Sustainable Development Goals

Several United Nations Sustainable Development Goals (SDGs) are in accordance with our project. Our project specifically helps achieve the following goals:

5.0.1 Goal 7 : Affordable and Clean Energy

Target 7.1 : By 2030, ensure universal access to affordable, reliable and modern energy services.

Target 7.2 :By 2030, increase substantially the share of renewable energy in the global energy mix.

The objective of this project is to enhance the use of sustainable and environmentally friendly sources of energy. This project aligns with the United Nations' Sustainable Development Goals such as the seventh goal of Affordable and Clean Energy through the utilization of technologies in solar power production and monitoring. It reduces many other hazardous emissions such as carbon dioxide and thus helps to make the environment cleaner and healthier by harnessing the energy of the sun. Solar energy usage has the potential to reduce greenhouse gas emissions in a way that is much better than the fossil fuels. This project supports sustainable resource management ensuring that the energy needs are met without compromising future generations' ability to meet theirs. The project's focus on solar energy directly supports the goal of increasing the share of renewable energy in the global energy mix. By improving the efficiency and reliability of solar power systems the project encourages more households and businesses to adopt solar energy.

5.0.2 Goal 11 : Sustainable Cities and Communities

Target 11.1 : By 2030, ensure access for all to adequate, safe and affordable housing and basic services and upgrade slums.

By using solar power as the main source of energy our project gives users a stable, efficient and cost-efficient means for electricity. Our project is extremely useful and vital for rural underdeveloped locations where the demand for electricity is high and the cost of electricity is unbearable also the use of grid electricity if available in rural areas is harmful for the environmental. Our solar powered energy project is not only beneficial for the underdeveloped rural areas but also the project provides the users with a very cheap and stable energy source and in turn increasing the standard of living of the common man.
With our solar powered system our project also contains a monitoring system through a mobile application that allow users to constantly check the systems energy parameters in real-time. This allows the user to keep check and make sure the system is providing optimum power output. There are also alerts that the user will be given in case of any ambiguity in the system which makes the dependency of the users with the grid electricity much lower. The combination of real time monitoring and alerts system allows the user to make sure the system is at its maximum potential.
For the rural underdeveloped areas where the access to electricity is not at all present our project for such areas is extremely vital and important. If our project in its entirety with the hardware and the software application is able to give the users of such rural areas with a stable source of electricity which in turn increases the standard of living of these people as they are now able use common necessities that every human should have.

5.0.3 Goal 13 : Climate Action

Target 13.2 : Integrate climate change measures into national policies, strategies and planning.

Through our project and the mobile application, the user can easily understand the peak production times and overall efficiency of solar installations, governments can design policies that promote the optimal use of solar energy.By the help of the mobile application the data on energy consumption through the help of graphs helps in planning resource allocation, ensuring that areas with high solar potential receive the necessary support and infrastructure.Through our mobile application users can educated and shown output data of the solar system briefing users of the benefits of solar energy and the importance of monitoring energy systems, fostering a culture of sustainability. This includes targets achieved in favor of renewable energy adoption and greenhouse gas emission reductions.By contributing to SDG 13 (Climate Action) and other related goals our project aims to provide an approach to a sustainable development while also integrating environmental and economical dimensions.

Chapter 6
Conclusion

We have successfully designed a fully functional hardware system which is monitored by a mobile application.Our system is integrated with various sensors which help to monitor the correct working of the individual separate parts of the circuit eg. sensor module, Charge controller.All values obtained by these sensors are calibrated using the code of the micro controllers to ensure accuracy.These values are then sent to Cloud storage(Firebase) where both real time monitoring and log based entries.Our mobile Application then displays these parameters with the help of graphs as well.Any abnormal condition that are predefined by us also gives an alert to the user which can be helpful in diagnostics.

References

[1] M. Hosenuzzaman, N. A. Rahim, J. Selvaraj, M. Hasanuzzaman, A. A. Malek, and A. Nahar, “Global prospects, progress, policies, and environmental impact of solar photovoltaic power generation,” *Renewable and sustainable energy reviews*, vol. 41, pp. 284–297, 2015.

[2] E. Kabir, P. Kumar, S. Kumar, A. A. Adelodun, and K.-H. Kim, “Solar energy: Potential and future prospects,” *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 894–900, 2018.

[3] P. S. Acharya and P. Aithal, “A comparative study of mppt and pwm solar charge controllers and their integrated system,” in *Journal of Physics: Conference Series*, vol. 1712, no. 1. IOP Publishing, 2020, p. 012023.

[4] V. Wallies Thounaojam and A. Balekundri, “Design and development of microcontroller based solar charge controller,” *International Journal of Emerging Technology and Advanced Engineering*, vol. 4, pp. 510–513, 2014.

[5] M. R. Rezoug, R. Chenni, and D. Taibi, “A new approach for optimizing management of a real time solar charger using the firebase platform under android,” *Journal of Low Power Electronics and Applications*, vol. 9, no. 3, p. 23, 2019.

[6] A. K. Singh, A. K. Agrawal, S. Vohra, S. S. Thakur, and G. Patel, “Solar charge controller,” *Int. J. Acad. Res. Dev*, vol. 2, no. 6, pp. 994–1001, 2017.

[7] P. Chougale, V. Yadav, A. Gaikwad, and B. Vidyapeeth, “Firebase-overview and usage,” *International Research Journal of Modernization in Engineering Technology and Science*, vol. 3, no. 12, pp. 1178–1183, 2021.

[8] J. Wu, L. Ping, X. Ge, Y. Wang, and J. Fu, “Cloud storage as the infrastructure of cloud computing,” in *2010 International conference on intelligent computing and cognitive informatics*. IEEE, 2010, pp. 380–383.

Appendix A: Charge Controller Code

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

float Vin=A0;             /// SOLAR VOLTAGE
float Vbat=A1;          // Battery Volt
 float x ;                      // for battery
  float y ;                    // for solar


 int pwm=6;

   #define FULL 255
  #define HALF 128
  #define QUARTER 64
  #define OFF 00

void setup()
{
  lcd.begin(16,2);
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  pinMode(Vin,INPUT);
  pinMode(Vbat,INPUT);

   pinMode(pwm,OUTPUT);
lcd.setCursor(0, 0);
  lcd.print("Charge");
  lcd.setCursor(0, 1);
  lcd.print("Controller");
   delay(2000);
   lcd.clear();

}

void loop()

{


     x= analogRead(Vin);

    float battery_v = x*0.061 ;      /// read solar voltage
     lcd.setCursor(0,0);
     lcd.print("Battery(v)=");
      lcd.setCursor(11,0);
     lcd.print(battery_v);

         // Read the battery voltage i;e Vbat
     y= analogRead(Vbat) ;


     float solar_in = y*0.0163;    // read battery voltage

    lcd.setCursor(0,1);
     lcd.print("Solar(v)=");
     lcd.setCursor(9,1);
     lcd.print(solar_in);
        delay(1000);
```

```
lcd.clear();
delay(1000);


 if  ((solar_in>battery_v)  &&( battery_v>13.00))
 {
  analogWrite(pwm,0);

    lcd.setCursor(0,0);
    lcd.print("Battery Charged");
     lcd.setCursor(0,1);
    lcd.print("Charged 100%");

 }



 if  ((solar_in>battery_v)  && (battery_v>12.75 and battery_v<=13))
 {
  analogWrite(pwm,125);

    lcd.setCursor(0,0);
    lcd.print("Battery Charging");
     lcd.setCursor(0,1);
    lcd.print("Charged 90%");

 }

   if  ((solar_in>battery_v)  && (battery_v>12.50 and battery_v<=12.75))
    {
  analogWrite(pwm,180);

    lcd.setCursor(0,0);
    lcd.print("Battery Charging");
           lcd.setCursor(0,1);
    lcd.print("Charged 80%");
    }

         if  ((solar_in>battery_v)  && (battery_v>12.30 and battery_v<=12.50))
    {
  analogWrite(pwm,190);

      lcd.setCursor(0,0);
    lcd.print("Battery Charging");
           lcd.setCursor(0,1);
    lcd.print("Charged 70%");
    }


if  ((solar_in>battery_v)  && (battery_v>12.00 and battery_v<=12.30))
    {
  analogWrite(pwm,200);


    lcd.setCursor(0,0);
    lcd.print("Battery Charging");
    lcd.print("Charged 50%");
    }

if  ((solar_in>battery_v)  && (battery_v>11.95 and battery_v<=12.00))
    {
  analogWrite(pwm,220);

    lcd.setCursor(0,0);
    lcd.print("Battery Charging");
    lcd.print("Charged 40%");
    }
```

```
  if  ((solar_in>battery_v)  && (battery_v>11.4 and battery_v<=11.95))
     {
   analogWrite(pwm,255);

   lcd.setCursor(0,0);
    lcd.print("Battery Charging");
    lcd.setCursor(0,1);
     lcd.print("Charged 20%");
     }

  if  ((solar_in>battery_v)  && (battery_v>10.5 and battery_v<=11.4))
     {
   analogWrite(pwm,255);

     lcd.setCursor(0,0);
     lcd.print("Battery Charging");
     lcd.setCursor(0,1);
     lcd.print("Charged 10%");
     }
     if ((solar_in>battery_v)  && (battery_v>10 and battery_v<=10.5))
     {
   analogWrite(pwm,255);

     lcd.setCursor(0,0);
     lcd.print("Battery Charging");
      lcd.setCursor(0,1);
     lcd.print("Charged 10%");
     }

    if  ((solar_in<battery_v))        //    solar is  low
     {
   analogWrite(pwm,OFF);

         lcd.setCursor(0,0);
     lcd.print("SOLAR LOW");
     }


       if  ((battery_v<10))        //    solar is  low
     {
   analogWrite(pwm,OFF);

         lcd.setCursor(0,0);
     lcd.print("Battery is dead");
     }



delay(1000);
lcd.clear();
delay(1000);



  }
```

Appendix B: ESP32 Code

```
#include <BH1750.h>
#include <Wire.h>


BH1750 lightMeter;
#include <SimpleDHT.h>
int pinDHT11 = 4;
SimpleDHT11 dht11(pinDHT11);
#include <FirebaseESP32.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include "WiFi.h"
#define WIFI_SSID "haseeb"
#define WIFI_PASSWORD "200718"
#define FIREBASE_HOST "https://data-logger-31875-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "seMbzJgT54r9KiWlw8wRe2tVzp31mluOQy03gORY"
const char* ntpServer = "time.google.com";
FirebaseData fbdo;
FirebaseData getdata1;
String formattedTime;
 String currentDate ;
     int cur_Minute =0;
    int pre_Minute =0;
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org");

//Week Days
String weekDays[7]={"Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday"};

//Month names
String months[12]={"January", "February", "March", "April", "May",
"June", "July", "August", "September", "October", "November", "December"};

float temp = 0;
float humi = 0;

void setup() {
  Serial.begin(9600);
    Wire.begin();

  lightMeter.begin();

  Serial.println(F("BH1750 Test begin"));
  Serial.println("Start");
  WIFI_Setup();
    timeClient.begin();
  // Set offset time in seconds to adjust for your timezone, for example:
  // GMT +1 = 3600
  // GMT +8 = 28800
  // GMT -1 = -3600
  // GMT 0 = 0
  timeClient.setTimeOffset(18000);

  Serial.println("Connected with success");

}

int entryCount = 0;


void loop() {
```

```
  float v1=0;
  float v2=0;
  float v3=0;

  float i1=0;
  float i2=0;
  float i3=0;

  float adc1=analogRead(36);
  float adc2=analogRead(39);
  float adc3=analogRead(34);
  for(int a=0;a<1000;a++)
  {
  float adc4=analogRead(35);if(adc4>i1){i1=adc4;}
  float adc5=analogRead(32);if(adc5>i2){i2=adc5;}
  float adc6=analogRead(33);if(adc6>i3){i3=adc6;}
  }

  i1=abs(2791-i1);
  i1=i1/272.00;


  i2=abs(2831-i2);
  i2=i2/271.85;


  i3=abs(2805-i3);
  i3=i3/737.8;


  v1=adc1/111.65;
  v2=adc2/115.55;
  v3=adc3/5.766;

  float soc=(18*v2)-152.2;
if(soc>100){soc=100;}
if(soc<10){soc=10;}
  float lux = lightMeter.readLightLevel();


   Serial.print(" ");Serial.print(v1);
   Serial.print(" ");Serial.print(v2);
   Serial.print(" ");Serial.print(v3);
   Serial.print(" ");Serial.print(i1);
   Serial.print(" ");Serial.print(i2);
   Serial.print(" ");Serial.print(i3);
   Serial.print(" ");Serial.print(soc);
   Serial.print(" "); Serial.println(lux);


   time1();
   DHT();


   Firebase.setString(fbdo,"Real_time/Log",currentDate);

   Firebase.setString(fbdo,"Real_time/PV Panel Voltage",v1);
   Firebase.setString(fbdo,"Real_time/PV Panel Current",i1);
   Firebase.setString(fbdo,"Real_time/PV Panel Power",(i1*v1));

   Firebase.setString(fbdo,"Real_time/Battery Voltage",v2);
   Firebase.setString(fbdo,"Real_time/Battery Current",i2);
    Firebase.setString(fbdo,"Real_time/Battery SOC",soc);

   Firebase.setString(fbdo,"Real_time/AC Load Voltage",v3);
   Firebase.setString(fbdo,"Real_time/AC Load Current",i3);
   Firebase.setString(fbdo,"Real_time/AC Load Power",(i3*v3));


   Firebase.setString(fbdo,"Real_time/LUX (Light Intensity)",lux);
   Firebase.setString(fbdo,"Real_time/Temperature",temp);
   Firebase.setString(fbdo,"Real_time/Humidity",humi);
```

```
void time1() {
  timeClient.update();
  time_t epochTime = timeClient.getEpochTime();
   formattedTime = timeClient.getFormattedTime();
   formattedTime.replace(":", "$");
  Serial.print("Formatted Time: ");
  Serial.println(formattedTime);
  String weekDay = weekDays[timeClient.getDay()];
  Serial.print("Week Day: ");
  Serial.println(weekDay);
  struct tm *ptm = gmtime ((time_t *)&epochTime);
  int monthDay = ptm->tm_mday;
  int currentMonth = ptm->tm_mon+1;
  String currentMonthName = months[currentMonth-1];
  int currentYear = ptm->tm_year+1900;
   currentDate = String(currentYear) + "-" + String(currentMonth) + "-" +
   String(monthDay) + "_" + String(weekDay)+"_";
 // currentDate+="  ";
  currentDate+=formattedTime ;
  formattedTime.replace(":", "$");

  Serial.print("Current date: ");
  Serial.println(currentDate);
  Serial.println("");
     cur_Minute = timeClient.getMinutes();

  }

void WIFI_Setup(void)
{
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.reconnectWiFi(true);
}

void DHT(void) {
byte temperature = 0;
byte humidity = 0;

// start working...
  Serial.println("=================================");
  Serial.println("Sample DHT11...");

  // read without samples.
  int err = SimpleDHTErrSuccess;
  if ((err = dht11.read(&temperature, &humidity, NULL)) != SimpleDHTErrSuccess) {
    Serial.print("Read DHT11 failed, err="); Serial.print(SimpleDHTErrCode(err));
    Serial.print(","); Serial.println(SimpleDHTErrDuration(err)); delay(1000);
    return;
  }

  Serial.print("Sample OK: ");
  Serial.print((int)temperature); Serial.print(" *C, ");
  Serial.print((int)humidity); Serial.println(" H");

temp=temperature;
humi=humidity;
}
```

122122-1162472859

ORIGINALITY REPORT

15%	8%	7%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	P. Sridhar Acharya, P. S. Aithal. "A Comparative Study of MPPT and PWM Solar Charge Controllers and their Integrated System", Journal of Physics: Conference Series, 2020 Publication	3%
2	www.academicjournal.in Internet Source	1%
3	Iustin Constantin, Alexandru Dinu. "Solution for Efficiently Charging of a Camper Van Battery", 2023 8th International Symposium on Electrical and Electronics Engineering (ISEEE), 2023 Publication	1%
4	miliohm.com Internet Source	<1%
5	suspace.su.edu.bd Internet Source	<1%
6	Submitted to University of Cape Town Student Paper	<1%

7	5dok.net Internet Source	<1 %
8	Submitted to Istanbul Aydin University Student Paper	<1 %
9	researchspace.csir.co.za Internet Source	<1 %
10	Submitted to Manchester Metropolitan University Student Paper	<1 %
11	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1 %
12	www.testmagzine.biz Internet Source	<1 %
13	www.unep.org Internet Source	<1 %
14	Submitted to University of Warwick Student Paper	<1 %
15	Submitted to University of Maryland, University College Student Paper	<1 %
16	downloads.hindawi.com Internet Source	<1 %
17	journal.maranatha.edu Internet Source	<1 %

18	dspace.uii.ac.id Internet Source	<1 %
19	Submitted to Engineers Australia Student Paper	<1 %
20	Submitted to Higher Education Commission Pakistan Student Paper	<1 %
21	www.mdpi.com Internet Source	<1 %
22	www.ganapathymani.com Internet Source	<1 %
23	Submitted to University of Liverpool Student Paper	<1 %
24	Submitted to East Forsyth High School Student Paper	<1 %
25	Submitted to University of Leeds Student Paper	<1 %
26	Submitted to Nelson Mandela Metropolitan University Student Paper	<1 %
27	Submitted to Olabisi Onabanjo University Student Paper	<1 %
28	Submitted to Texas A&M International University Student Paper	<1 %

29	Submitted to Oran Park Anglican College Student Paper	<1 %
30	Submitted to St Stephen's College Student Paper	<1 %
31	Submitted to Texas State University- San Marcos Student Paper	<1 %
32	Submitted to Vienna International School Student Paper	<1 %
33	docplayer.net Internet Source	<1 %
34	ungc-production.s3.us-west-2.amazonaws.com Internet Source	<1 %
35	courses.ece.illinois.edu Internet Source	<1 %
36	Submitted to Riverside College Halton Student Paper	<1 %
37	Submitted to University of Newcastle Student Paper	<1 %
38	Submitted to Vet Nurse Plus Student Paper	<1 %
39	www4.enphase.com Internet Source	<1 %

40	www.coursehero.com Internet Source	<1%
41	Submitted to University College Technology Sarawak Student Paper	<1%
42	Submitted to University Tun Hussein Onn Malaysia Student Paper	<1%
43	Submitted to University of Northumbria at Newcastle Student Paper	<1%
44	Submitted to University of Ulster Student Paper	<1%
45	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%
46	cf.sembcorp.com Internet Source	<1%
47	www.ijamtes.org Internet Source	<1%
48	Submitted to UCSI University Student Paper	<1%
49	Submitted to University of Southern California Student Paper	<1%
50	greenhomeinstitute.org	

Internet Source

<1%

51	hyclassproject.com Internet Source	<1%
52	www.diva-portal.org Internet Source	<1%
53	Ton Duc Thang University Publication	<1%
54	ujcontent.uj.ac.za Internet Source	<1%
55	www2.mdpi.com Internet Source	<1%
56	Chapelsky, Chris Terry. "A novel two-switch, multi-level class-D inverter and its implementation as an audio power amplifier", Proquest, 20111003 Publication	<1%
57	Kumar Sachin, Vivek Prakash Yadav, Aashiwal Nirmal Mukesh, Praveen Kumar Sharma, L Solanki. "Design and Development of PWM based Solar Hybrid Charge Controller", 2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 2021 Publication	<1%

58	L. Ashok Kumar, S. Albert Alexander, Madhuvanthani Rajendran. "Charge controls and maximum power point tracking", Elsevier BV, 2021 Publication	<1%
59	vdocument.in Internet Source	<1%
60	www.proshareng.com Internet Source	<1%
61	Khushal Verma, Savita Chauhan, Jay Singh, Nagendra Kumar. "Bidirectional Visitor Counter using Microcontroller with Sanitization and Thermal Screening", Journal of Physics: Conference Series, 2021 Publication	<1%
62	Μπούρας, Δημήτριος | Bouras, Dimitris. "Quantitative Modelling for the Assessment of Organizations Sustainable Development with the Aim of Formulating Appropriate Decision-Making Policies", University of Piraeus (Greece), 2023 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		