Design and implementation of a controllable DC load for an off-grid PV system

By

Dana Baddawi & Saif Sayegh & Hussam Baidas & Hassan Al- Halawani

Supervised by

Ibrahim Abu Shumais

Submitted in partial fulfillment of the requirements for the degree of
BACHELOR OF SCIENCE
in

Electrical Power and Energy Engineering
at
PRINCESS SUMAYA UNIVERSITY FOR TECHNOLOGY

Amman, Jordan
Fall 2021

This is to certify that I have examined
this copy of an engineering documentation by

Dana Baddawi & Saif Al-Sayegh & Husam Baidas & Hassan Halawani

And have found that it is complete and satisfactory in all respect,
And that any and all revisions required by the final Examining Committee have been made

Dr. Ibrahim Abu Shumais

Acknowledgments

First and foremost, we want to thank our advisor, Dr. Ibrahim Abu Shumais, for his unwavering support of our BSc. studies and research, as well as his patience, encouragement, passion, and vast expertise. His advice was important during the research and writing of this project. For our BSc. studies, we could not have asked for a greater advisor and mentor.

Abstract

Solar energy has a huge impact on our lives, it has the potential to generate huge amounts of power and provide us with electricity, not only in grid connected areas, but off-grid locations as well. Throughout this project, we designed a microcontroller to control a DC load, connected to the off-grid PV system in Princess Sumaya University for Technology's Renewable Energy Lab. It is intended to control the load in 10% steps, follow all safety protocols and effectively monitor the PV system's output power. This design also enables us to ensure that the battery is appropriately discharged to avoid over and undercharging the batteries to prevent any damage. It is designed using an Arduino Mega 2560 R3 and coded to be fully controllable.

Table of Contents

1 Introduction

1.1 Overview

Access to power provides great benefits to development as well as economic welfare. In Jordan, where the country is reliant on importing oil due to the lack of it in the country, can benefit from the development of PV systems. When taking the geographical location of

Jordan into consideration, the benefits of the high solar radiation can be reaped since it’s located in the earth-sun area that has high potential of solar energy.

Off-grid PV systems have the advantage of being independent from the grid which aids in providing power using renewable energy generated from solar radiation without being connected to the grid. However, battery storage is necessary for its implementation.

Off-grid PV systems can be connected to AC or DC loads depending on the purpose of the connection. Controllable DC loads, on the other hand, serve as a method of testing powersupplies. For instance, in our case, controllable loads can be used to make sure that the solar system is able to deliver its fully designed potential and protect the batteries. We needed this system to be connected to the off-grid system to discharge the batteries since it is not connected to a load. Discharging the batteries is necessary since overcharging and undercharging a battery causes damage to the batteries and might lead to a shorter battery life.

1.2 Objective:

The goal is to design a controllable DC load that is connected to the off-grid PV system installed in the Renewable Energy Lab at Princess Sumaya University for Technology. The load must not exceed 500W and be fully controlled by the lab user while abiding by the CE safety standard 2001/95/CE. The Off-grid PV system in the lab isn’t connected to any discharging source. Therefore, our system will be used to discharge the battery and will be turned off to charge it once again.

1.3 Principle of Operation of Off-grid Systems

The essential principles of all solar power systems are all the same. The photovoltaic (PV) effect is used by solar panels to convert solar energy or sunlight into DC power. The DC power can then be stored in a battery or converted to AC power by a solar inverter for use in household appliances. Excess solar energy can be delivered into the power grid for credits or stored in a variety of storge systems, depends on the type of installation.

Off grid systems consist of four main components:

- Solar panels
- Batteries
- Inverters
- Charge controller

The main task of solar panels is to generate electricity (DC power) from sunlight. Solar panels consist of several PV cells connected together. PV cells are made up of two slices of semiconducting material, usually silicon. They work by allowing photons to knock electrons free from the atoms, generating a DC current. In off grid systems, batteries can act as storage components, where additional power can be stored, which can be helpful during cloudy days.

Charge controllers, connected between PV panels and batteries, control the charging process by regulating the current and voltage supplied by the solar panels. Charge controllers are programmed to disconnect the batteries when they are fully charged to protect the batteries from overcharging. In addition, charge controllers feed a DC load whenever it exists at the same time of controlling.

Some household appliances such as TVs, laptops, and radios operate on AC power. In order to be able to power such devices, power has to be converted from DC power (supplied by the panels) to AC power.

1.4 Design Requirements

1. The designed systems should be connected to the off-grid PV system safely.
2. The system size should not exceed 500 W.
3. The system should be fully controlled by the lab user with the following functionalities:
    a. Display the load status (connectivity, current, power).
    b. Ability to turn-on and off the load at specific time.
    c. Control the load in 10% step.
4. Protection against load short circuit.
5. Disconnect the load automatically if battery bank is discharged.
6. Suitable for 24 or 48 Vdc input voltage level.
7. PSIM tools shall be used to simulate the system.
8. The total cost of the system should not exceed 500 JOD.
9. The system should fit in the renewable energy lab in Princess Sumaya University for Technology.

1.5 Analysis of Design Constraints

1) For our design to be as safe as possible, we used relays on each part of our load (our load was split into 10 equal parts of LED strips) to protect the already working PV system and our load.

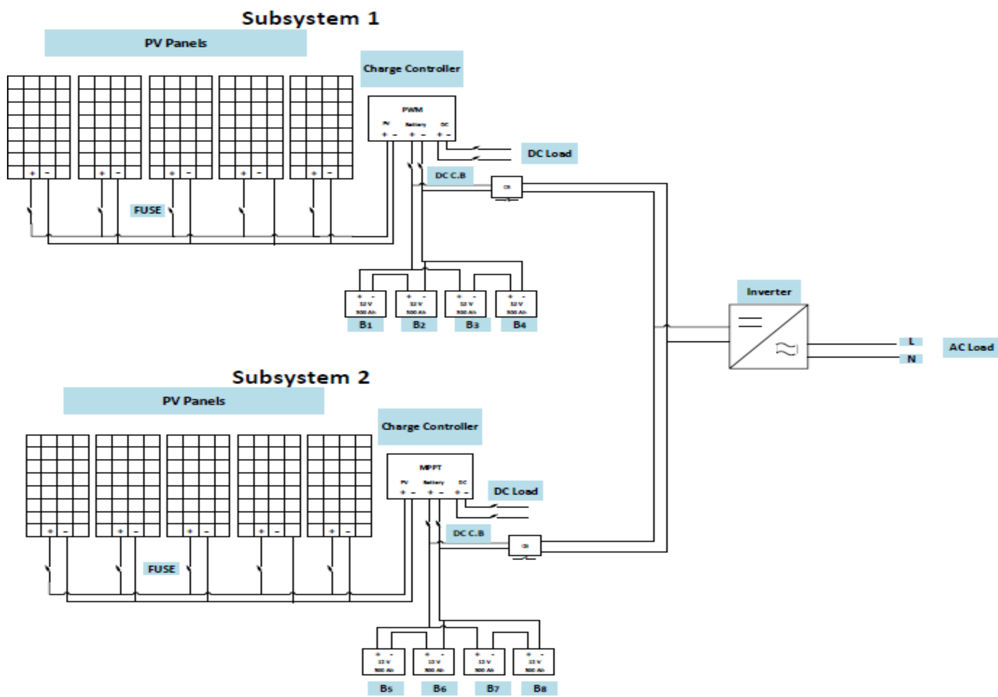


Figure 1-1 PSUT PV System

- This PV system's maximum power output is 2500W. Our load will be connected to the MPPT (we chose to use the MPPT since it has a higher efficiency than the PWM) which is half of the system producing 1250W. However, we will only be connecting a 350W controllable load using a PIC controller that controls how much of the load will be turned on at each time (the load will be divided into 10% steps of 35W for each LED) and turning the load ON / OFF at any time while displaying the current and power consumed.

2) Disconnect the load automatically if battery bank is discharged. connecting the current sensor to the battery's wires and connecting the current sensor to the microcontroller and placed a relay which is connected to the microcontroller. checking the value of the current sensor using TIMR0 interrupt every 1m second and computing the power if the current was high then the power is high too then the microcontroller going to give the relay a signal to connect the load if the current was low then the microcontroller going to send a signal to relay to disconnect the load

3) For the system cost to not exceed 500Jds, we decided to use 24V LED work light 35W LED flood spread beam for CAT articulated trucks as our load since it was the cheapest and most efficient option and small enough to fit in our renewable energy lab.

1.6 Engineering Standards

Referring to Directive 2014/35/EU of The European Parliament and of The Council of 26 February 2014, the principal elements of the safety objectives for electrical equipment designed for use within certain voltage limits are as follows:

1. General conditions

(a) The essential characteristics, the recognition and observance of which will ensure that electrical equipment will be used safely and in applications for which it was made, shall be marked on the electrical equipment, or, if this is not possible, on an accompanying document.

(b) The electrical equipment, together with its component parts, shall be made in such a way as to ensure that it can be safely and properly assembled and connected.

(c) The electrical equipment shall be so designed and manufactured as to ensure that protection against the hazards set out in points 2 and 3 is assured, providing that the equipment is used in applications for which it was made and is adequately maintained.

2. Protection against hazards arising from the electrical equipment.

Measures of a technical nature shall be laid down in accordance with point 1, in order to ensure that:

(a) Persons and domestic animals are adequately protected against the danger of physical injury or other harm which might be caused by direct or indirect contact.

(b) Temperatures, arcs, or radiation which would cause a danger, are not produced.

(c) Persons, domestic animals, and property are adequately protected against non-electrical dangers caused by the electrical equipment which are revealed by experience.

(d) The insulation is suitable for foreseeable conditions.

3. Protection against hazards which may be caused by external influences on the electrical equipment.

Technical measures shall be laid down in accordance with point 1, in order to ensure that the electrical equipment:

(a) Meets the expected mechanical requirements in such a way that persons, domestic animals, and property are not endangered.

(b) Is resistant to non-mechanical influences in expected environmental conditions, in such a way that persons, domestic animals and property are not endangered.

(c) Does not endanger persons, domestic animals, and property in foreseeable conditions of overload.

The standards set above were applied to the project and described in detail in section 3.7 “Application of Engineering Standards”.

2 Literature Review

2.1 Sizing and Design of an off-grid system

-The process of sizing any off-grid system is as follows:

1) **Determining the power consumption (the demand).**

This can be done by counting the number of devices that will be operated in each day, specifying the power of every single device in watts and determining the number of operating hours for each device in one day by using the following equation:

$$\boldsymbol{daily\ consumption = device\ power \times number\ of\ devices\ (of\ the\ same\ kind) \times operting\ time} \qquad (1)$$

-To get the total energy needed from the system, add the energy daily consumption Wh/day needed for all appliances Some factors cause energy losses in the system. Some of these factors are:

a) **An increase in the temperature of the PV panels.**

b) **Accumulation of dust on the PV panels.**

c) **The ideal conditions that the PV panels were designed in**

d) **The power consumption by the charge controller and inverter**

e) **The power consumption by batteries due to the charging and discharging processes.**

Due to the factors listed before, multiplying the total energy daily consumption by 1.3, which is 30%more from the total energy daily consumption, will give the total watt-hour per day which must be provided by the panels

2) **Determining the size and number of the PV panels needed**.

There are two methods to determine the size and number of the PV panels which are:

a) **Panel Generation Factor (PGF) method:**

- The PGF is a variable factor that is affected by the climate of the location where the PV panels are found. This factor can be obtained from tables belong to .

For example :

- When estimating the size of the PV cells, the total watt-peak rating of the cells needs to be calculated by dividing the total watt-hour/day needed from the PV modules by the PGF.

-The estimated number of PV panels needed can be found by the following equation:

$$number\ of\ panels = \frac{power\ of\ PV\ panels}{P\ Max\ of\ panels} \tag{2}$$

; Where the watt-peak per cell can be shown in Figure 2-1.

Figure 2-1: panel’s datasheet

b) Peak Sun Hour (PSH) method:

- The power delivered from the PV panels can be calculated using the following equation:

$$power\ of\ PV\ panels = \frac{daily\ consumptoin}{PHS} \tag{3}$$

Note that: $1 peak\ sun\ hour = 1000wh/m^2$ and the average solar radiation in Jordan is between 4 to 8 Kwh/m^2.

- The radiation levels in Jordan can be shown in Figure 2-2 below:

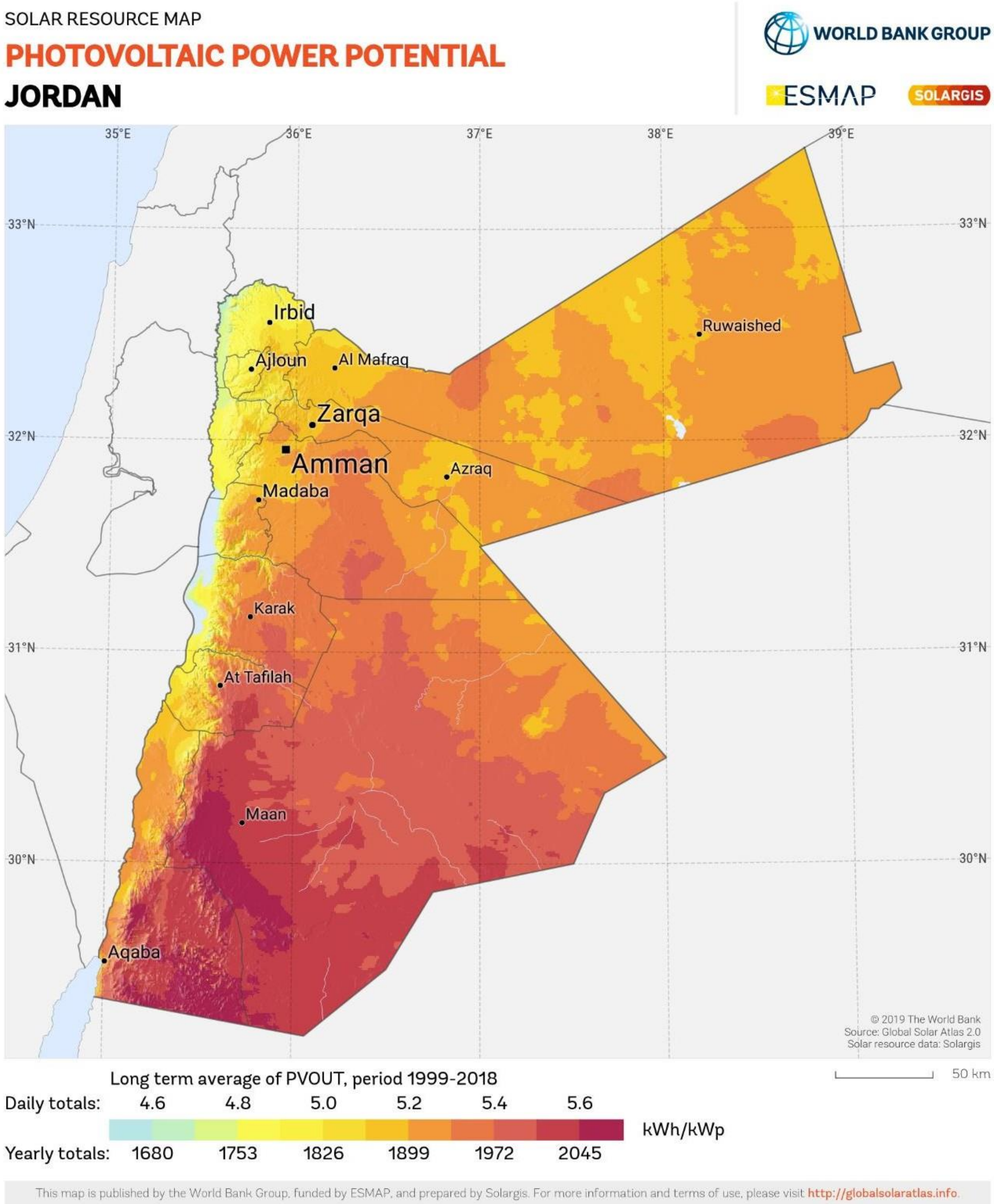

SOLAR RESOURCE MAP
PHOTOVOLTAIC POWER POTENTIAL
JORDAN
WORLD BANK GROUP
ESMAP
SOLARGIS
35°E
36°E
37°E
38°E
39°E
33°N
32°N
31°N
30°N
Irbid
Ajloun
Al Mafraq
Zarqa
Amman
Azraq
Madaba
Ruwaished
Karak
At Tafilah
Maan
Aqaba
© 2019 The World Bank
Source: Global Solar Atlas 2.0
Solar resource data: Solargis
50 km
Long term average of PVOUT, period 1999-2018
Daily totals: 4.6 4.8 5.0 5.2 5.4 5.6
kWh/kWp
Yearly totals: 1680 1753 1826 1899 1972 2045
This map is published by the World Bank Group, funded by ESMAP, and prepared by Solargis. For more information and terms of use, please visit http://globalsolaratlas.info.

Figure 2-2: Solar radiation in Jordan (Solargis, 2018)

3) Determining the number of batteries needed for the off-grid PV system.

- This can be done by the following steps:

a) Calculating the **current needed** by the system in every hour per day.

- This can be done by dividing the energy consumed by the system over the voltage of the system.

$$total\ load\ (\frac{Ah}{day}) = \frac{daily\ consmption\ (\frac{wh}{day})}{system's\ voltage\ (V)} \tag{4}$$

b) Calculating the **depth of discharging**. This value can be obtained by using the following equation or by assuming it to be 80%:

$$Depth\ of\ discharging = \frac{\text{capacity discharged from a fully charged battery}}{\text{nominal capacity of the battery}} \tag{5}$$

c) Finding the day of autonomy, which is the number of days that the battery can supply the sit's loads without any support from generation sources. In Jordan, there are two days of autonomy.

d) Calculating the **usable storage**. This value can be found using the following equation:

$$usable\ storge(Ah) = days\ of\ autonomy\ (day) \times total\ load\ (\frac{Ah}{day}) \tag{6}$$

e) Finding the **storage capacity of the batteries**. This value can be found from the datasheet of the batteries or can be calculated using the following equation:

$$storge\ capacity(Ah) = \frac{usable\ storage\ (Ah)}{depth\ of\ discharging} \tag{7}$$

f) Calculating the number of batteries needed by using the following equation:

$$number\ of\ batteries = \frac{capacity\ of\ the\ system\ (Ah)}{storge\ capacity\ of\ the\ battery\ (Ah)} \tag{8}$$

4) Determining a suitable charge controller for the system.

- To determine the current rating of the charge controller, the short circuit current of the panels used must be found and then multiplying it by a safety factor of 1.3. Then by using the following equation, the rated current can be found:

$$rated\ current = total\ short\ circuit\ current \times number\ of\ panels \times 1.3\ (safety\ factor) \tag{9}$$

- The voltage rating of the charge controller must be the same as the voltage of the system.

5) Determining a suitable inverter for the system.

- This can be done by calculating the rated power of the inverter needed. The rated power must also be multiplied by a safety factor of 1.3. This is shown in the following equation:

$$rated\ power = power\ consumption\ by\ the\ load(W) \times 1.3 \tag{10}$$

2.2 Microcontrollers

We live in an era of information revolution, with computers of incredible power at our disposal. Computers have infiltrated every aspect of human life. For high-powered applications, some are designed to be as powerful as feasible without regard for cost. In both industry and academia are geared for use at home and in the office, and are less powerful, less expensive but still useful. This is a sort of computer that is built into a device to give its functionality. Microcontrollers are the name for this type of product.

Microcontrollers are basically made from MOSFET (metal-oxide-semiconductor field-effect transistor). The MOS integrated-circuit, an integrated circuit chip built using M-O-S-F-E-Ts, first appeared. M-O-S chips had improved transistor concentration while also lowering productioncosts. Large-scale- integration (L-S-I) with lots of transistors on a single M-O-S chip became possible as M-O-S devices became more advanced.

So, the microcontroller is a miniature computer built on a single M-O-S integrated-circuit (I-C) chip.

It has one or more C-P-Us (processing cores), as well as memory (RAM plus ROM) and programmable input//output peripherals, all of which communicate with one another by translating data from its I//O peripherals using its processor to do that. The microcontroller's temporal data is saved in its data memory, where the process core restores(fetches) it, decodes(solves), executes (performs) the instructions that are stored in its program memory and then return a value by applying the incoming data, then it contacts its I//O peripherals and takes the action needed. The processor is the device's brain intelligence. It explains, interacts and responds to the instructions set which controls the microcontroller. This involves fundamental arithmetic, logic, and input//output (I//O) operations. Microcontrollers have two types of memory:

1. **Program memory**. This is where the CPU holds instructions it executes, without any kind of power source this type of memory stores the data inside it. This type is called non-volatile.
2. **Data memory.** With a certain type of source, this memory stores data temporarily while instructions are being executed. This type is called Volatile.

The processor core's input and output devices serve as an interaction with the outside world. Transferring information to the CPU after obtaining it from outside world in form of binary data is the input ports' task. The microcontroller's CPU takes this information and sends the desired commands to output devices that perform activities outside of the microcontroller.

While the CPU, I//O peripherals, and memory are the most significant microcontroller's elements, other elements are commonly supported. Incorporating elements that interact with the memory and CPU are referred to as I//O peripherals. Peripherals can include a wide range of supporting elements, like such as communicating with sensors. The microcontroller has a feature called ADC ("analog to digital converter") that allows the signal to be converted from analog to digital and vice versa using another feature called DAC ("digital to analog converter").

2.3 Off-grid System Components

2.3.1 PV arrays

- A PV system uses PV panels that consist of PV cells that collect solar energy and converts it to DC electricity. The reflection of sunlight creates an electric field across the photovoltaic system resulting in a flow of electricity.

- A PV array, on the other hand, is a combination of several PV Panels connected together. The Figure 2-3 shows the array's structure.

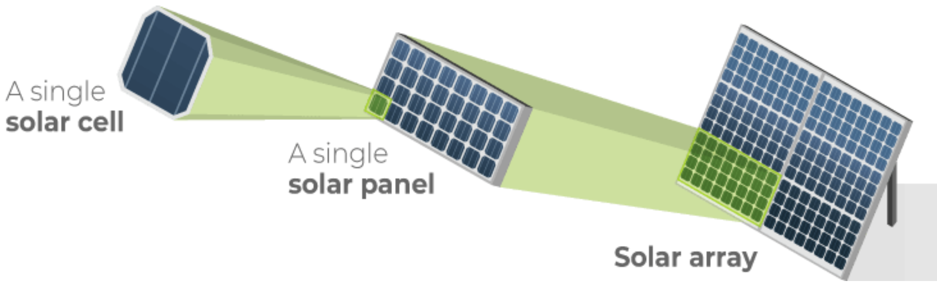


Figure 2-3: Solar arrays

- When PV cells are connected together, they make up a solar panel. Solar panels work by allowing photons, or particles of light, to knock electrons free from atoms, generating a flow of electricity.

- PV cells are made up of two slices of semiconducting material, usually silicon. In order for the cells to establish an electric field, silicon is doped with another material to give each slice of the cell a negative or a positive charge. Usually, phosphorous is added to the top layer of silicon, which adds extra electrons (n-doped), giving a negative charge to that layer. Boron is added to the lower layer which results in fewer electrons (p-doped), or a positive charge. This all adds up to an electric field at the junction between the silicon layers. Then, when a photon of sunlight knocks an electron free, the electric field will push that electron out of the silicon junction.

- Metal plates are also placed on the sides of the cells in order to collect the electrons and transfer them into the wires.

- Most of the solar cells are made from silicon wafers either single-crystalline or multi-crystalline. Crystalline silicon has an ordered crystal structure, with each atom ideally lying in a pre-determined position.

- Chemically pure silicon is called Intrinsic Silicon. Silicon has four valence electrons which are held by covalent bonds with the valence electrons of four adjacent silicon atoms. These electrons do not conduct electricity. However, when the temperature rises some valence electrons break out of the covalent bonds leaving a vacancy in the bond which is called a "hole".

-In order to generate electricity, doping atoms are deliberately introduced into the crystal lattice. These atoms have one electron more (phosphorus) or one electron less (boron) than silicon in their outermost electron shell. Hence, they result in impurity atoms in the crystal lattice. Figure 2-4 shows the doping process.

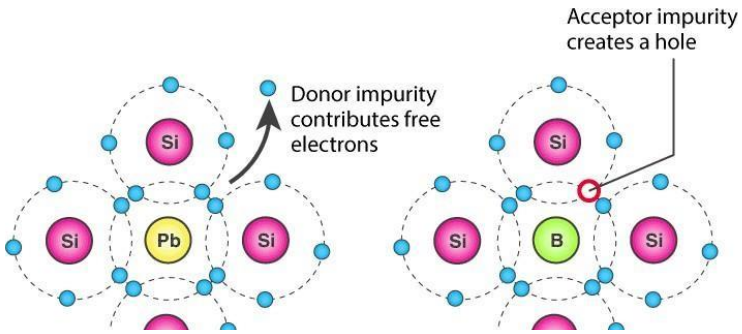


Figure 2-4: n and p-doped silicon

n - type p - type

- In the case of phosphorus doping (n-doped), there is a surplus electron for every phosphorus atom in the lattice. This electron can move freely in the crystal and hence transport an electric charge.

- With boron doping (p-doped), there is a hole (missing bonding electron) for every boron atom in the lattice. Electrons from neighboring silicon atoms can fill this hole, creating a new hole somewhere else. The conduction method based on doping atoms is known as extrinsic conduction.

2.3.2 Working Principle of PV cell

- The exposure of the semi-conductor to sunlight leads to the absorption of photons by the electrons. This energy absorbed breaks electron bonds; the electrons are released then pulled through the electrical field into the n-region.

- The holes that are formed migrate into the p-region. The diffusion of charge carriers to the electrical contacts causes a voltage to be present at the solar cell.

-In an unloaded state, the open circuit voltage (Voc) arises at the solar cell. If the electrical circuit is closed, current flows. Some electrons do not reach the contacts and recombine instead.

- The energy difference between the top of the valence (outer electron) band and the bottom of the conduction (free electron flow) band is called the “Band-gap”.

- In the valence band electrons are tightly held in their orbits by the nuclear forces of a single atom. In the conduction band, electrons have enough energy to move around freely and are not tied to any one atom.

- A semiconducting material is a material that has a small band gap, which behaves as insulators at absolute zero, but allow excitation of electrons into their conduction bands (at temperatures below their melting point).

-In order for an electron to make the leap from the valence band to the conduction band, it requires a boost of "band gap" energy where it can gain enough energy by absorbing a photon (light) with enough band gap energy to separate an electron pair. Figure 2-4 shows how the electricity generated in PV panels

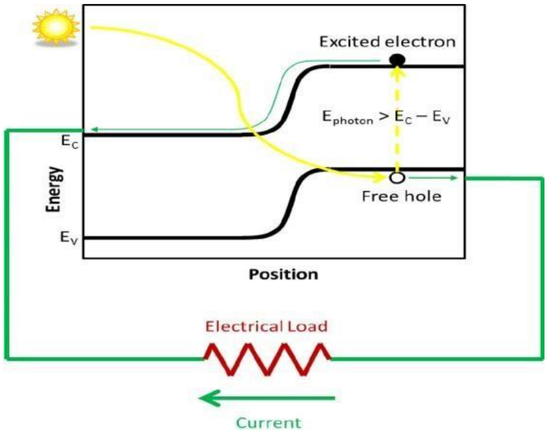


Figure 2-5: Electricity generation in a PV system

2.3.3 Charge controller

Charge controllers regulate the current and voltage supplied from PV arrays to the batteries and from batteries and to the load. A charge controller's fundamental role is to manage charging the battery. It prevents overcharging and controls the rate of the current and the voltage supplied.

Charge controllers prevent current from flowing in the opposite direction (into the PV array instead of the batteries). This happens as a result of an increase in the voltage across the batteries up to the point where: $V_{Baterries} > V_{Panels}$. Therefore, one of the roles of charge controllers is to limit current flow to one direction, from the panels to the batteries.

There are two main types of charge controllers:

1) Pulse Width Modulation (PWM):
PWM charge controllers pulse the power supplied to the battery in order to perform distinctive charging stages.
PWM charge controllers provide a nominal voltage of solar panels equal to the nominal voltage of the batteries. Therefore, if the panel has a nominal voltage of 12V, the connected batteries must have a nominal voltage of 12V. PWM charge controllers have the advantage of being the cheaper type of charge controllers.

2) Maximum Power Point Tracking (MPPT):

MPPT charge controllers are more sophisticated and more expensive than other types. A MPPT tracks the output power of the solar array and adjusts it accordingly in order to maximize efficiency.
MPPT charge controllers have a 30% higher efficiency than PWM charge controllers.

Since conditions vary such as solar irradiance, temperature, and load, Maximum power point tracking (MPPT) charge controllers keep PV systems operating at the peak power point of the PV panel. To achieve that an algorithm is developed and implemented in PV inverters in order to maximize the power generated by the PV system. MPPT algorithms are typically used in the controller designs for PV systems and account for a variety of factors that might affect the power generated

There are several types of algorithms two of them are:

1) Perturbation and observation (P&O):

This algorithm introduces a perturbation to the voltage to ensure maximum power point to the PV. The PV output power is periodically measured and compared with the previous power. The perturbation continues if the output power increases, otherwise the perturbation is reversed since the goal is to achieve maximum power. The voltage across the battery and the PV is measured when the MPPT is connected between them in order to determine whether the battery is fully charged or not. To prevent the battery from overcharging, the MPPT stops charging if the battery is full. This MPPT algorithm is simple, easy to implement, and low cost with high accuracy. Figure 2-6 shown the algorithmflow chart.

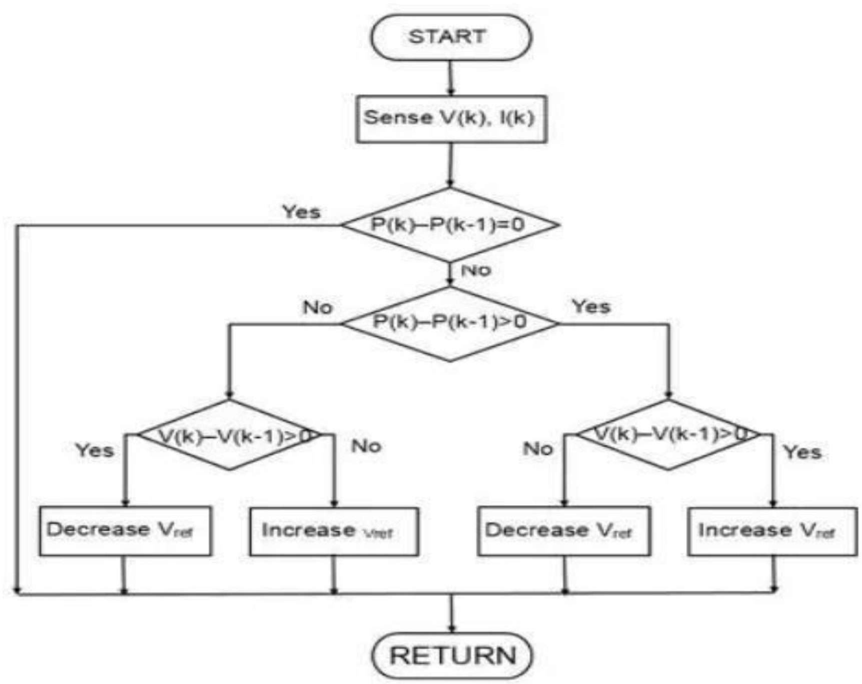


Figure 2-6 P&O Algorithm Flowchart (Matlab)

2) Incremental Conductance:

This algorithm depends on the instantaneous $\frac{I}{V}$ and incremental conductance ($\frac{dI}{dV}$). Both are compared and depending on the value, the voltage is varied to achieve optimum operating

power and depending on the relationship between the incremental conductance and the instantaneous conductance, the operating of the PV module in the P-V curve can be determined. As opposed to P&O algorithm, the voltage remains constant once the maximum power is reached, this algorithm is also better at finding the MPP during sudden changes in irradiance conditions, but it needs a complex controller. A detailed flow chart is shown in Figure 2-7.

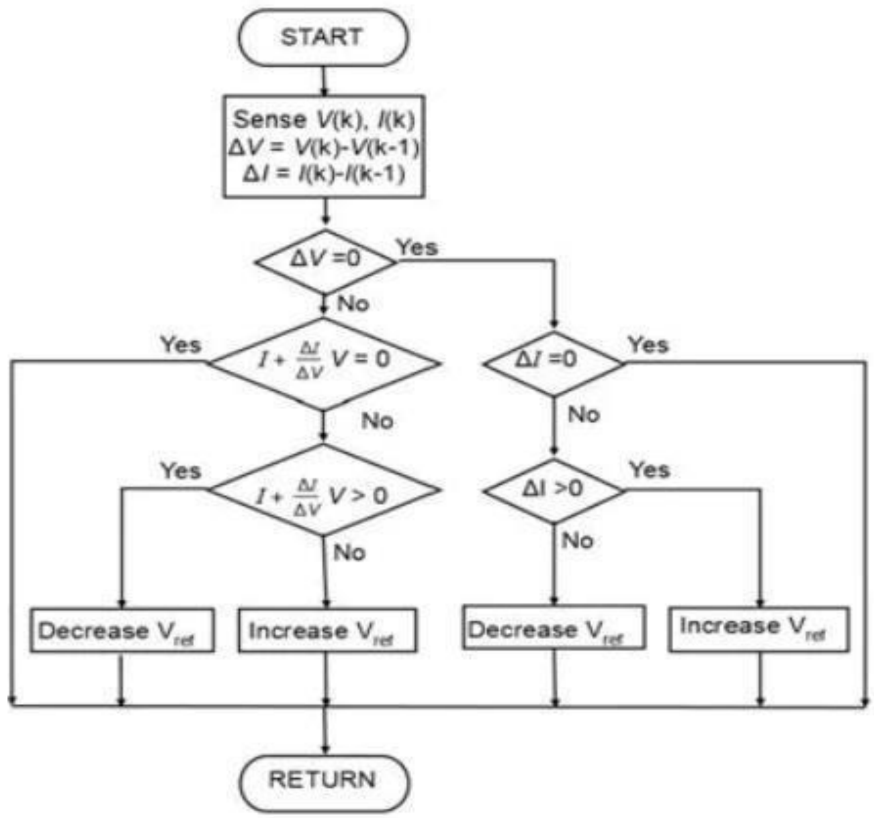


Figure 2-7 Incremental Conductance Algorithm Flowchart
(MATLAB)

2.3.4 The Battery

- A battery: is a device that consist of two metal layers (for example zinc and copper) separated bya paper or cloth socked in salt water. The working principle of a battery depends on the oxidation and reduction processes, as zinc oxidizes (loses electrons), which is in turn gained by the ions in water. These two processes produce hydrogen gas and energy.

- Note that this oxidation-reduction cycle creates a flow of electrons between two substances and ifa lightbulb for example has been connected this process will provide the lightbulb with power.

- Batteries have a finite supply of metal and once most of it has oxidized the battery “dies”. A temporary solution for this problem are the rechargeable batteries which converts the oxidation-reduction process to a reversible process. This can be achieved by allowing electrons to flow back in the opposite direction to regenerate the metal making more electrons available for oxidation thenext time if they are needed.

- But even rechargeable batteries don’t last forever over the time, the repetition of this processcauses imperfections and irregularities in the metal’s surface that prevents it from oxidizing properly, the electrons become no longer available to flow through a circuit and thus the battery “dies”.

- There are two main types of batteries used in solar systems:

1) lithium batteries.

2) lead-acid batteries.

Some of the benefits of using Lithium batteries over lead-acid batteries are:

1) Lithium batteries have ten times the cycle life of lead-acid while also being cheaper than lead-acid batteries.

2) The capacity of lithium batteries is around 100% and unlike the lead-acid batteries, the charging rate stays constant with the discharging rate. Whereas in lead-acid batteries, the capacity percentage is higher than in lithium where it can reach 120% in some types such as the FLA and AGM but it decreases with the discharging rate.

3) Lithium batteries are four times faster in charging than lead-acid and even at high temperatures the lithium batteries still have twice the life cycle of lead-acid batteries that are stored at room temperature.

4) For battery installation it rejected to install lead-acid in an inverted position and that’s to prevent any potential issues from happening on the other hand, its accepted to install lithium upside down or at any side. Regarding the weight lithium battery, on average is 55% lighter than lead-acid.

2.3.5 Fuses

It’s a protection or a safety device that gets rid of excess current in order to protect the components of the circuit. Its functionality is centered around knowing and sensing the short circuit current in addition to opening the circuit or interrupting the current to prevent any damage.to fuses have no effect on the circuit when operating under normal conditions.

Fuses are made up of a strip which is constructed of silver, copper, aluminum or tin, this strip is placed between two conductive joints in a non-combustible case.

In comparison to circuit wires, fuses have the following properties:

1) A strip with a low melting point.
2) Lower strip thickness.
3) Lower resistivity. Principle of Operation of Fuses:

When an overcurrent flows through the circuit, the conductive joints in the fuses and the low resistivity of the strip allows the current passing through to access the strip. The overcurrent heats up the strip until it reaches it melting point, since the strip’s thickness is relatively small, it will be torn due to heat and interrupt the current. The glass or plastic case surrounding the fuses prevents damage to the surrounding components.

Cartridge fuses:

There are two types of cartridge fuses:

1) Low rupturing capacity:
   This type of cartridge fuses is divided into two types, the Ferrule-contact, and Diazed Screw-type. There two types are further divided into two categories, the Quick Response and theDelayed Action fuses.
2) High rupturing capacity.

2.3.6 Molded Case Circuit Breakers (MCCB)

-Molded Case Circuit Breakers (MCCBs): are electrical protection devices that protectsan electrical circuit from high amounts of current.

- If a fault occurs in an electrical circuit, a huge flow of current will pass through thecircuit. This current is called “a fault current”.

- There are three main types of fault currents:

1) Overload current: This fault current is usually caused by using way more extension cords which places a higher current demand on the circuit than it is designed to withstand. This current can be high enough to melt the wires of acircuit
or damage its components.

Figure 2-8: MCCB

- A MCCB can protect against this overload current by its contacts that are made of two types of metals. When an overload current passes through a MCCB, its contacts' temperature will increase causing these metal contacts to expand at different rates and bend away from each other until they reach a certain temperature that we call the "trip value" where these contacts expand far enough to physically interrupting the circuit by pushing the trip bar and unlatching the contacts.

2) Short circuit currents: When the voltage across a solar cell becomes zero a very high current passes through the circuit which is called the short circuit current.

-A MCCB can protect against a short circuit current instantaneously. A MCCB contains a solenoid coil. Under normal conditions, when a current passes through this solenoid an electromagnetic field is generated. However, under normal conditions this electromagnetic field is negligible, and nothing happens. However, when a short circuit current passes through this solenoid, a very strong electromagnetic field is generated that is strong enough to attract the trip bars and thus opening the contacts.

3) Inrush current: This current usually appears in devices such as motors. This high current only lasts for a very small duration of time. Usually, the MCCB is set with a small delay to ignore this inrush current since it only lasts for a short duration of time.

2.3.7 Relays

In order to drive low-voltage DC loads transistors -as-a-switch is used but that will not be sufficient to drive a high-voltage DC and AC loads so the ultimate solution for this is to use relays.

Relays are electromechanical switches that operates like a normal switch but it is controlled electrically using a coil.

In its simplest form, relays consist of three contacts:

1) The NC (normal closed contact)
2) The NO (normal opened contact)
3) The Common Contact. It is connected to a spring and beneath it there is a coil which is connected to a low voltage source, around 5 volts or 12 volts.

In order to control the relay using a microcontroller (PIC) that is connected to a DC load, using a transistor and a freewheeling diode is important. One of the load's terminals will be connected to the MPPT 15 volts output and the other terminal to the typically open contact, the common contact will be connected to MPPT's positive terminal of 15-volt output and the coil will be connected to the microcontroller (PIC) through a transistor connected to either a 12V or 5V power source.

In order to trigger a relay, the microcontroller has to send a binary signal (1) to the relay through a transistor, this will create a magnetic field around the coil, making the relay an electromagnet. This causes the relay to close and conduct current.

A current sensor is used to determine whether there is a fault (overload current). The microcontroller receives a signal from the current sensor, identifying the fault. The microcontroller then sends zero binary signal to the relay through the transistor so that channel becomes non-conductive. When this happens, the electromagnet loses its power and spring closes, triggering the relay and causing it to act as an open switch. Therefore, no current is passing through the relay to the load and a fault is avoided.

2.3.8 Controllable DC load

DC controllable loads or electronic loads are instruments that apply voltage and sink current. They are most commonly used by power supply, solar, wind, or battery manufacturers. These loads are commonly used by manufacturers to test the compliance to quality of the product by rapidly increasing and decreasing the load.

They ensure that a product is working under safety conditions. They are commonly used to test batteries to determine their lifetime.

For instance, Chroma of the leaders in manufacturing electronic loads. One of their products (Benchtop DC electronic Load – 63000) designed to test switching power supplies, with 350 W maximum rated power, suitable for testing lower powered devices. Figure 2-9 shows the controllable DC load.

Figure 2-9: Chroma 63000 Series Electronic DC Load

This device can also be used for battery discharge testing with:

- Three discharge modes: CC, CR and CP.
- Set cut off voltage and time (1~100,000 sec.) to stop loading and ensuring the battery is not damaged due to over discharge
- Measures the battery discharge power (WH, AH) and total discharge time
- Applies to super capacitor for discharge time testing and other related applications

2.4 Similar Projects

2.4.1 First project

A study was conducted in Udayana University in Bali, Indonesia, where they designed an
Arduino-based Data Acquisition Device for Monitoring Solar PV System Parameters, showing the importance of integrating automatic-sensor based data acquisition in PV systems to provide quick response and accuracy in real-time in comparison to manual measurement in terms of monitoring the PV system's performance and storing data.

A correct reading can be hard to obtain when PV systems are constantly operating for long periods of time. This happens as a result of changing environmental conditions, when using traditional data collection methods such as multimeters and thermocouples. Automatic sensor based data acquisition (DAQ) provides a solution for this problem since manual measurement isn't as effective as quick response in real time.
monitoring the PV system performance and storing the data become possible.
Monitoring the PV system is essential since it helps to keep the system safe, stable, and provide information about the provided energy, energy potential, and operating temperature of different of faults that might occur. This reduces cost by avoiding faults in the system.

The proposed design for this study was intended to be a low cost DAQ for monitoring the PV system's parameters and analyzing its performance. The designed DAQ was equipped with the ATmega 2560 chip microcontroller of the Arduino Mega board with real-time data recording, low energy consumption, and utilizing open-source software.

Components of the DAQ:

- An Arduino Mega board with At mega 2560 chip microcontroller is used for a core signal processing. This mega board is equipped with 54 digital I/O, 16 analog input and 256 kB flash memory
- A SD card shield for data storage with a real-time clock (RTC) DS1307 on board.
- A FZ0430 model (range: 0.02445-25VDC, accuracy: ±0.00489V) voltage sensor.
- A ACS712ELC-30A (range: ±30A based on the output of 66mV/A) current sensor.
- A DS18B20 model digital temperature sensor is used to measure temperature in the range from -55 ºC to 125 ºC and accuracy of ±0.125 ºC.
- A DHT22 model (range: 0-100% of humidity, -40ºC -80ºC of temperature; accuracy: ±2% of humidity, ±0.5 ºC of temperature) digital temperature and humidity sensor.

Design of the DAQ

Current sensors are connected to pins A0, A2, and A6 with three voltage sensors connected to pins A1, A3, and A7 through a voltage divider consisting of 30KΩ and 7.5KΩ.

A digital temperature and humidity sensor is connected to pin D2, while two temperature sensors are connected to a pullout resistor of 10 KΩ, connected to pin D3.

The data readings are stored on the Arduino Mega board that is equipped with a Atmega 2560 chip microcontroller and displayed on an LCD (20x4).

The SD card stores the acquired data at a specific time on a CSV file by using a serial peripheral interface (SPI) connection.

A power supply of 9V is used for powering the Arduino board.

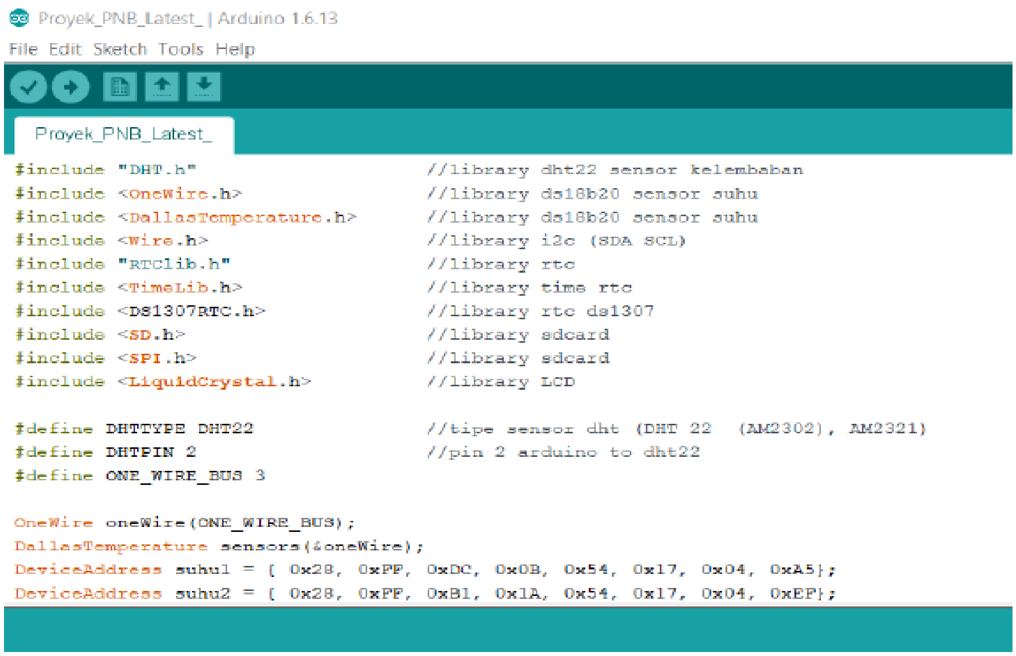


Figure 2-10: Code Uploaded to the Ardunio Mega board

The code is then written and uploaded to the Arduino board through a USB cable. The code used to implement their design is shown in Figure 2-10.

The DAQ is installed to a PV panel of 50Wp, a battery of 12V/24Ah, an SCC of 20A, and a DC load of 15W LED lamp.

The voltage, temperature, humidity, and current are the parameters being measured. Current and voltage sensors are used to measure the output of the panel, and the DC load.

The PV system was experimentally tested at Politeknik Negeri Bali in Indonesia from 8 am to 5 pm local time. The physical construction of the DAQ is shown in Figure 2-11:

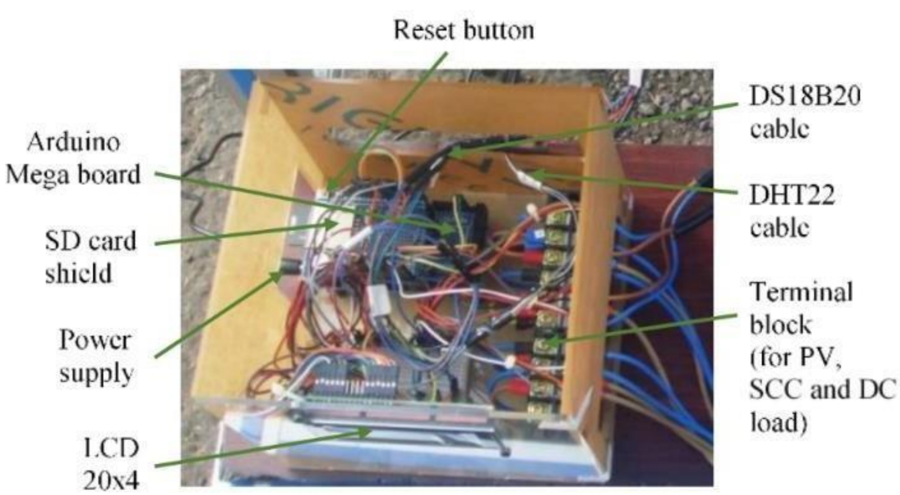


Figure 2-11: DAQ construction

2) Home Energy and Power Management and Control System.

Another study was conducted in University of Ghana, Legon, where a home energy and power management and control system were designed. This designed system allows users to monitor their demand response while also regulating and managing this response using a mobile application. Just like our project, these students are using a microcontroller as the main power management system. However, by using a raspberry pi which is a web server for the local network they were able to use a Wi-Fi connection to communicate between the mobile application and the power management system and Bluetooth to provide communication between the controller and the Raspberry Pi. Their goal was to make it easier and more flexible to control and regulate this home system to easily show how much of the cost is being saved.

Home energy management is an important skill to learn (such as turning off certain electrical devices, delaying using power such as washing clothes using a washing machine until peak hours to save more). Home energy management started with the "AMI" (advanced metering infrastructure) where this device helped the demand side to manage and efficiently save energy which means the user was able to respond to help improve the efficiency of power generation and supply. Recently, with the advancement of technology, home energy management solutions are now more intelligent and flexible than ever before. Many solutions for the user- side have appeared such as smart energy saving devices, newly made energy management systems which automate the user response for better load management.

The idea of this project was to design a home energy management and control system that is controlled using a mobile application to allow the users of this program to monitor, manage, and control their consumption and power usage. The system operated using voltage and current sensors that sent data to the microcontroller to analyze this data, then by receiving a control signal from the microcontroller, appliance usage can be controlled using electronic relays that receive the control signals from the microcontroller.

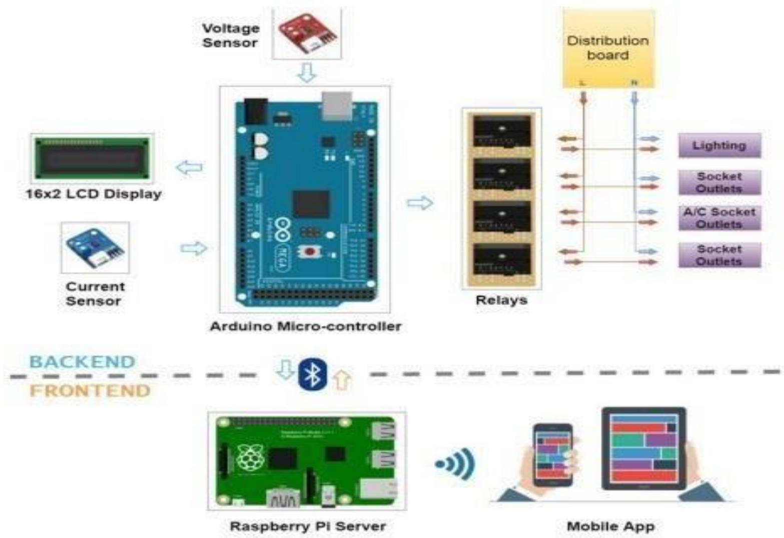


Figure 2-12: Desired system.

2.4.2 Second project

System Design and Development

1) Hardware system design:

a) Voltage and current sensors was designed to receive single-phase voltage supply of 230Vrms with ±10% tolerance that was scaled down to a range of 0-5V and sent to the microcontroller.

b) To automatically control the connected appliances, relay circuits were designed that were directly connected to the microcontroller.
Figure 2-13 shows the system's circuit.

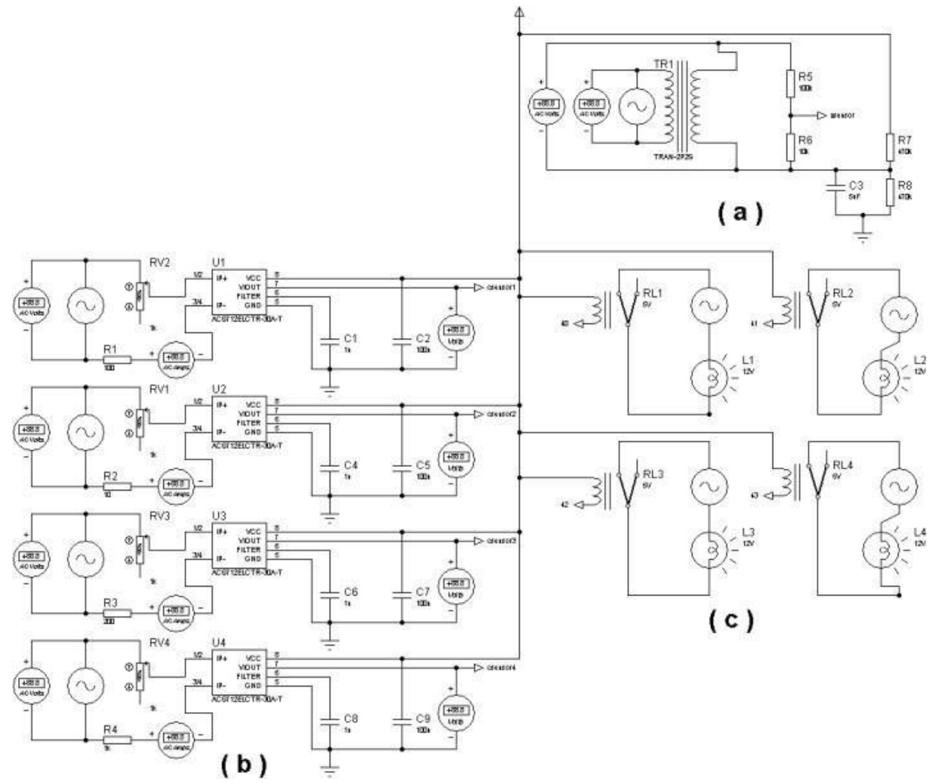


Figure 2-13: Designed circuit of the system

where the system labelled (a) is a voltage sensor circuit, (b) the current sensor circuit and (c) is the relay circuit for control.

- The desired system worked as provided in the following flow chart Figure 2-14:

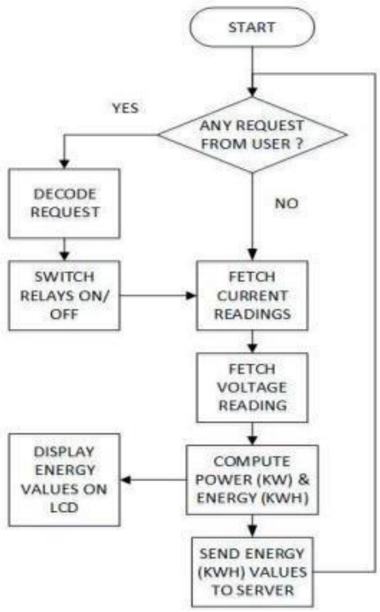


Figure 2-14: The desired system flow chart

2) Software system design:

Using the Android platform, the mobile application was designed to have five different interfaces for the user. These five interfaces were:

1) **Power monitor:** Responsible to view real time and historic data.

2) **Power control:** Responsible to control when to turn on and off the switches using toggle switches so that the user can control which appliances to turn on or off using the relays.

3) **Power scheduling:** Create or cancel user made schedules so that the user can control which times certain appliances are being used and this schedule continues to operate until the schedule is cancelled by the user.

4) **Power usage plan:** Create, edit, activate or delete plans where the user can control certain appliances and link them to the power supply such as a solar generator.

5) **Price manager:** Allows the user to update electricity prices so that the energy management system can compute the cost of the energy consumed at home.

- The Raspberry pi server's job is to save all the data it received where this data will be used to display old or real-time energy and power consumption data. It also provides asynchronous access to the various services mentioned above.
Figure 2-15 shows the software explanation

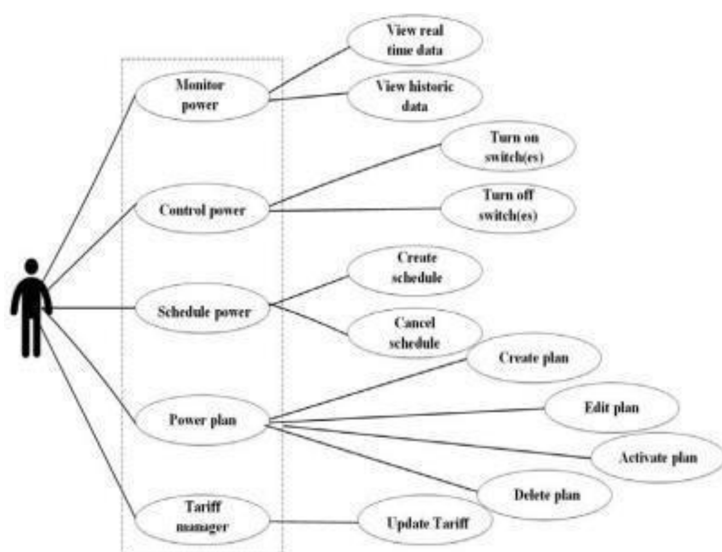


Figure 2-15 : Software explanation

Implementation:

- A simulation of the design and the model of the microcontroller was created using Proteus. The model used to distribute electricity to the different parts of the home was a four-way distribution board.

1) To measure different current values on the designed circuit, each current sensor was connected to a different variable resistor and the variable resistors' values were designed to be different from each other and fed to four current sensors.

2) To show the system's ability to measure voltages, the voltage sensors were connected to a nominal supply of 230Vrms.

3) To show the system's switching functionality, it was simulated to drive lamps which represented different loads and the control relays were toggled ON/OFF using virtual serial interface.

- A prototype device made to make sure the system is working properly; the prototype is shown below in Figure 2-16:

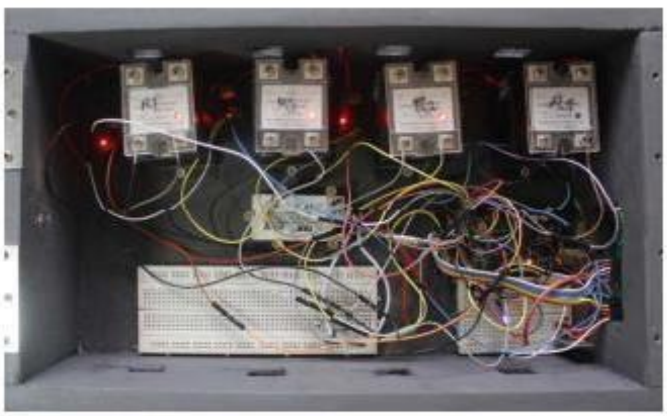


Figure 2-16: System prototype

- And the experimental setup to test the prototype above was shown in figure 2-17:

Figure 2-17 Testing setup

Which included several different loads to make sure the system works for usual home loads. These loads and their ratings were included in the table 2-18 below:

	Load types	Power ratings
1	Light bulbs (incandescent)	60W, 80W, 100W
2	Electric (steam) iron	2400W
3	Standing fan	70W
4	Water heater/Kettle	2000W
5	Electric blender	350W
6	Refrigerator	200W
7	Microwave oven	1200W
8	Hand dryer	1800W

Table 2-18: Load types and ratings

- The application was tested using a Samsung Galaxy S5 phone, the android version on this phone at the time was Android 5.0. Another phone was used to test the application, the Samsung Galaxy Note 3 phone was chosen, the android version on this phone at the time was Android 4.4. A figure was included to show the real time mobile interface information on power and energy consumption and the energy cost as the appliances or loads were increased or decreased on each circuit from the distribution board. Figure 2-19 shows the interfacing

Figure 2-19 Mobile application interface

- Finally, pictures of the mobile interface for scheduling scheme and the interface for the power plan schedule were also provided in Figure 2-20 and Figure 2-21:

Figure 2-19: Mobile interface for scheduling scheme

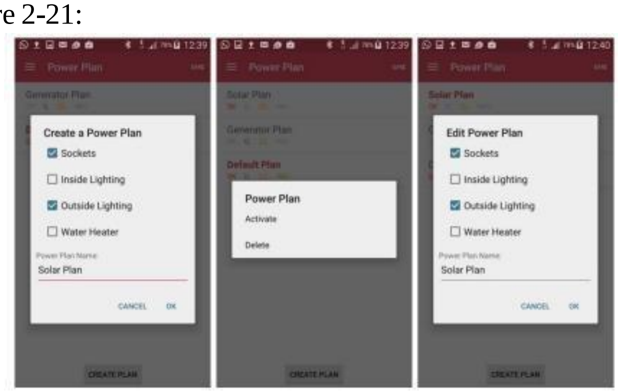


Figure 2-20: Mobile interface for the power plan schedule

3 Design

3.1 Different Designs Approaches/choices and developed hardware

1) The first load considered was a LED strip of around 2.5m. It required 1.5A and a 24V DC for each strip which means that each strip consumed around 35W, dividing our load into 10 strips meant that we could divide the 350W required load into 10 steps of 35W. This choice was not finalized because of the length of each LED strip even though it was cheaper than the LED car lights.

2) The second load considered was a highly lumen LED light that would be installed outside the renewable lab to light up the walkway during the night. However, this choice was also not finalized because of the high cost of this LED light that would make the cost exceed the required limit.

3) The third load considered was using five 24V LED work light 35W LED flood spread beam for CAT articulated trucks as the first half of our load and five resistors that can handle the current that will pass through them (1.5A) without their temperature increasing to an unsafe amount. This design not only helps us reduce the cost of our system but also from a practical side, five LED lights are enough to light up either the renewable lab or the walkway outside the renewable lab without taking up too much space.

4) For the relays and PIC controller, a single power supply was to be used to power both the controller and the relays. However, even though this will save some costs, to avoid any risk of the relays producing a very high current back to the controller it was decided to use a separate power supply for the relay and a separate power supply for the PIC controller.

5) The whole idea was to use light sensor tied on the solar panel to sense the light if there is a light then the load going to take power form panels else it's going to take it from battery the Idea has been changed to use voltage sensor with certain circuit, connecting this sensor on the battery, to reduce the cost and to guarantee that the sensor going to measure high values the Idea has been changed to use current sensor which tied on the wires that come out from the batteries to measure the current that comes out from the batteries if it was high the relay will close its contacts and feeding the load else the relay will disconnected.

6) Multiple loads were taken into consideration, first the LED flood lights. The initial design included connection of ten 35 W LED floodlights. However, it was difficult to find LEDs with that rating in Jordan. Therefore, we went with lower rated LED floodlights with 20 W each.

7) Due to the issues faced in finding a 35 W rated LED, we took into consideration LED car lights. Two options were tested, one of them didn't have desired heat tolerance and failed, the other was working properly, drawing around 24 W. We eventually scrapped this idea because the lights were more expensive, needed a frame (covering to be installed to it), and didn't provide us with a much higher power rating when compared to the cheaper LED floodlights.

8) We also took into consideration the connection of a mix of loads (LEDs and Resistors) That would've helped us calibrate the power needed to be drawn and would've given us less light since we're connecting half the number of lights. The issue was that a high-power resistor with a heat sink was unavailable in Jordan and ordering it online would have been very expensive.

9) PIC VS Arduino:

In our initial design, we chose the PIC 16F877A as the microcontroller to be used in our system which was going to be programmed using C language. This microcontroller was swapped for the Arduino Uno for the following reasons:

a. The PIC isn't equipped with enough RAM to handle our code due to its length. Whereas the Arduino has a bigger RAM, which made it more compatible with our system. We ended up usingonly 23% of the Aruino's RAM.

b. The addresses overlapped in the PIC's ROM due to a lot of data being added to the microcontroller. After switching to the Arduino, only 5% of the ROM was used.

c. The PIC isn't as safe in terms of components protection as the Arduino is. The PIC has the ability to return current into the relays which can cause damage to the system as opposed to theArduino which has better protection.

d. The Arduino is a modern, open-source device while microcontroller an old fashion device and the resources on it are scarce

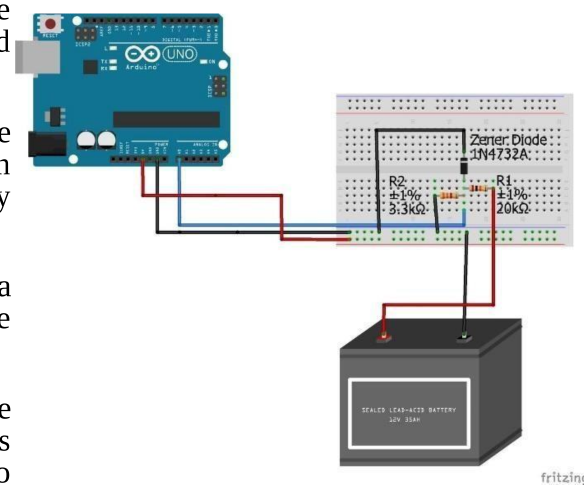


Figure 3-1 Voltage Divider Circuit

10) Voltage Divider:

Initially, we built a voltage divider to compute or measure the battery's voltage, this circuit consists of zener diode, two resistors, Arduino UNO and batteryconnected as Figure (3-1).

What happens here is that the voltage divider receives the voltage from the battery and scales it up from 0 to 4.2 V by multiplying it by $\frac{3.3}{20}$ to be able to insert it into the Arduino UNO. The code then returns the value as it was from 0V up to 30V, that's through multiplying the small-scale voltage by inverted voltage divider which is $\frac{20}{3.3}$.

Note that the Zener diode's job is to protect the Arduino from excess voltage it limits the voltage value up to 4.2 V. Also note that the maximum voltage that arrived at voltage divider is 26V, but we assumed it to be 30 V to give the circuit reliability. The option to measure the battery's voltage was no longer used because, when we connect the current sensor to measure the battery's current the voltage divider become inaccurate, the battery's current is distributed between the 3.3 k Ω resistor and the current sensor.

11) <u>Current Sensor (ACS712)</u>

We built a small circuit to measure the sensor's current or the battery's current, this circuit consists of battery or power supply, a constant resistor equals to 10.4 Ohms, Arduino UNO, voltmeter and CSA712 current sensor connected as following see Figure (3-2)

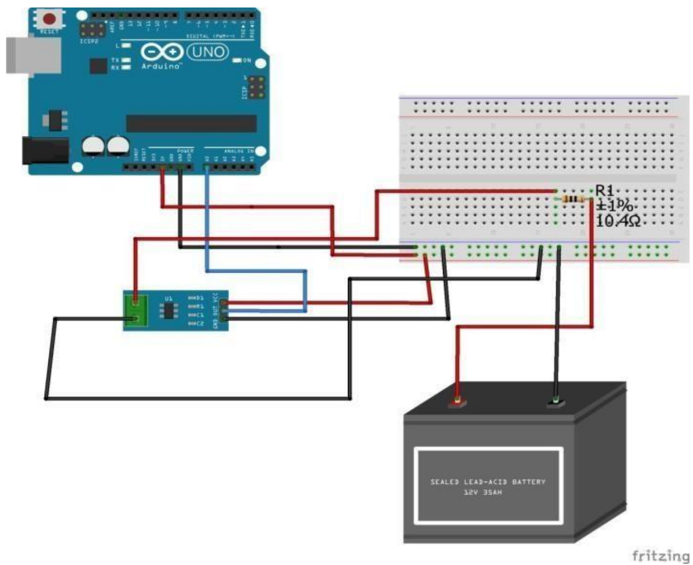


Figure 3-3 Current Measurement Circuit Using ACS712

To measure this current (battery's current) we built up a linear relation between current sensor and the output voltage of the sensor, through applying different values of voltages as an input from power supply, with constant resistor we could compute the current that enters the sensor, and by observing the output voltage of the sensor through the voltmeter we could find the relation.

The relation analytically :

Vin	I	Vo
0V	0A	510mV
0.5V	0.043A	522mV
1.5V	0.144A	533mV
2.5V	0.240A	544mV
3.5V	0.336A	554mV

$$slope = \frac{554m - 510m}{0.3365 - 0} = 0.13075$$

$$y - yo = s(x - xo)$$

$$y - 510m = 0.13075(x - 0)$$

$$y = 0.13075x + 510m$$

$$y \text{ is refered to } I \text{ (current)}$$

$$x \text{ is referd to } Vo \text{ (voltage)}$$

So, $Icurrent = 0.13075Vo + 510m$

We insert this equation in the code to compute the value of batteries current, note that we are observing the values of the output voltage from the serial communication program which is built in the Arduino UNO and beside this program we uses the voltmeter to double check the accuracy of output voltage values.
But this equation no longer works and that's because two reasons:

1. This current sensor read the values of current for a linear resistor (constant load) And what we found is that our loads were not linear, the current passes through them was changing dramatically when we apply on them a voltage.

For example:

(26 Volts – 24 Volts) →(gives)→ 30 Ω
(24 Volts – 20 Volt) → (gives) → 3Ω
(20 Volts – 10volts) → (gives) → 7Ω

2. This sensor is not accurate and has bad resolution it has a resolution of 113m with accuracy of 10 bits

12) Current. Voltage, Power Sensor (INA219)

We did some research and found another accurate current sensor which is INA219 Bi-directional DC current/power monitor sensor (see the data sheet in appendix G).
Connecting current sensor in parallel with battery or batteries and in series with a one load of our connected loads as its shown in Figure (3-4).

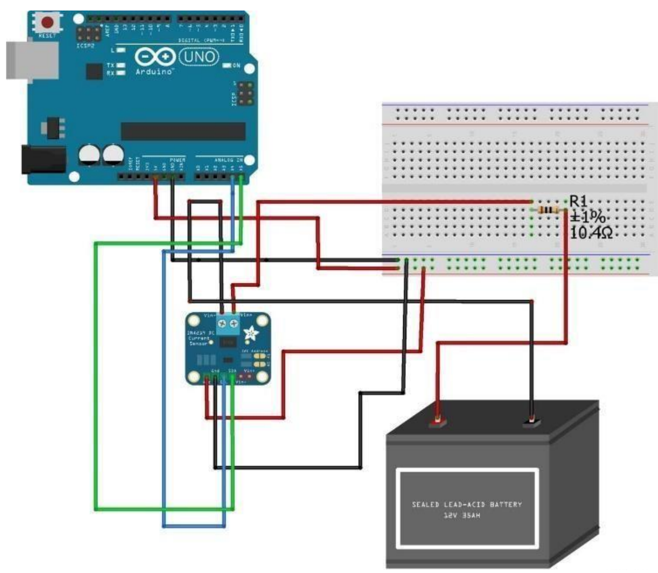


Figure 3-4 Current Measurement Circuit Using INA219

This sensor has the following specifications:

1. It has current resolution of 100µ A which means that this current sensor changes its value with ±100µA steps which is acceptable.
2. It has a minimum current of 100µA and maximum current of 3.2 A
   note that the value of the maximum current of this sensor is close to or high than the maximum current that flows through the loads which is at 30Ω equals to $\frac{26\,V}{30\Omega} = 0.8\ A$,

   and less than the minimum current that flow through the loads which is at 7Ω equals to $10/7 = 1.4\ A$, in other words the current that flow through the loads or the sensor's current is always in the range.

13) One of the options considered was to place a wall-mounted load, with the lights visible as shown in Figure 3-5 below. However, the lights shun too bright and where uncomfortable andunsafe for the user's eyes. Therefore, we chose to add four legs to the prototype, add plastic covering with heat tolerance, and place it on a surface (desk) near the batteries.

Figure 3-5 Wall-mounted Prototype Without Cover.

3.2 Developed Design

- In this project the off-grid system has been designed, and its ready to use, so the procedure of sizing will change a little bit, inversely walking on steps and equations will lead us to know the period allowed to discharge the batteries. There are two ways to calculate the discharging period:

1) When the load is connected to 4 batteries through the MPPT charge controller. Two batteries are connected in series and then parallel with the rest of the batteries. Each battery gives 300Ah, 15A, and 12, which equates to two batteries each providing 24 V, 15A, and 300Ah connected in parallel as this off-grid system

The power generated by the batteries is:

$$Power = Voltage \times current$$

$$Power = 24 \times (15)$$

$$Power = 360\,W$$

- According to the power calculated above, the allowed energy consumption of the loads is found by:

$$allowed\ energy\ consumption = 360 \times 5h(PSH)$$

$$allowed\ energy\ consumption = 1800Wh$$

- Dividing the allowed energy consumption by 1.2 to get rid of the energy consumed by the system that has been added on the energy consumed by loads:

$$allowed\ energy\ consumed\ by\ loads = \frac{1800}{1.2} = 1500\,Wh$$

Now, to connect 10 DC loads each one consumes 36 W the allowed discharging period equals to

$$number\ of\ operation\ hour = \frac{allowed\ energy\ consumed\ by\ loads}{number\ of\ laods \times power\ of\ loads}$$

$$number\ of\ operation\ hour = \frac{1500}{10 \times 36} \approx 4\ hours$$

2) Another way to calculate the discharging period is to take the discharging current of the batteries, the ampere-hour, and the depth of discharging (these values selected using datasheet of batteries) and by using the following equations, the discharging period can be found:

$$the\ time\ of\ discharging\ (discharging\ period) = \frac{depth\ of\ discharging\ \times amper - hour}{current\ of\ discharging}$$

$$discharging\ period = \frac{1 \times 300 \times 0.25\,Ah}{15} = 5\ hours$$

Note that each silicon cell only puts out half a volt so, this project needs 72 cells to produce 24 volts to install such a PV array to the system it must be taken into account few points to ensure that thesystem produces the most energy, since it is exposed to the most intense sunlight for the longestlength of time. when the sun's rays reach the solar panel's surface perpendicularly, the maximum electricity is captured, to ensure that this will happen the PV panels must be directed to true southin northern hemisphere and vise verses, the PV panels must be directed to true north in southern hemisphere.

The solar panel tilt angle from the horizontal surface should be decided according to the latitude of the location and other factors.

Another point to consider is how to connect solar panels in series or in parallel and that's depends on the quantity of energy that needed, for example this project has 5 panels connected in parallel every one of them gives 250W so

$$250 \times 5 = 1250W$$

and finally specifying the area needed to install the PV-panels

$$number\ of\ panels = \frac{power\ of\ system}{Pmax\ of\ the\ panel}$$

$$area\ nedded = 2(meter\ shadows) \times numberof\ panels \times area\ of\ each\ panel$$

3.3 Off-grid System Simulation

- The following circuit is built on PSIM software to study the behavior of the PV system installed with the load connected to it. The circuit wasn't simulated with the microcontroller designed for this system. It is used to study the output power, cell current, cell voltage, and the power consumed when certain loads are disconnected from the circuit which will be controlled with the designed controller.

'

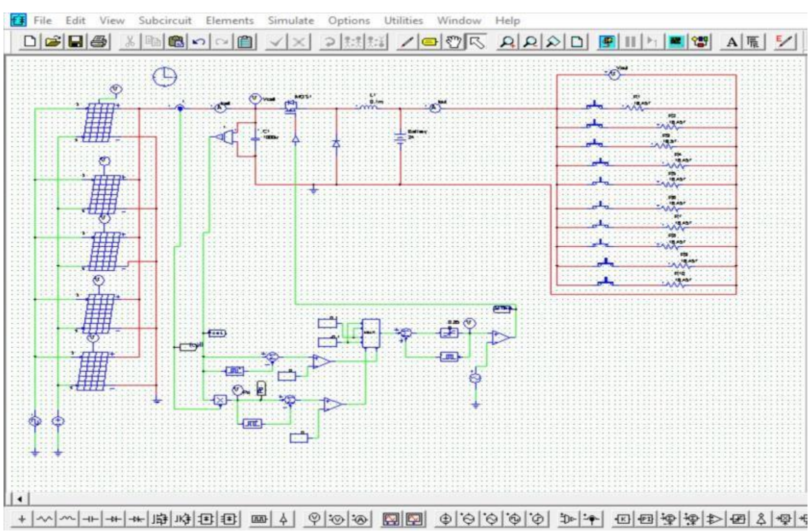


Figure 3-6 Circuit of The System on PSIM Software

According to the installed off-grid PV system in the university, five Suntech STP250-20/wfw panels are connected in parallel to the MPPT. The specifications of these panels were inserted to the PV module (physical model) in PSIM software to be able to simulate the system, in addition to acquiring the P-V and I-V curves shown below for a single PV panel.

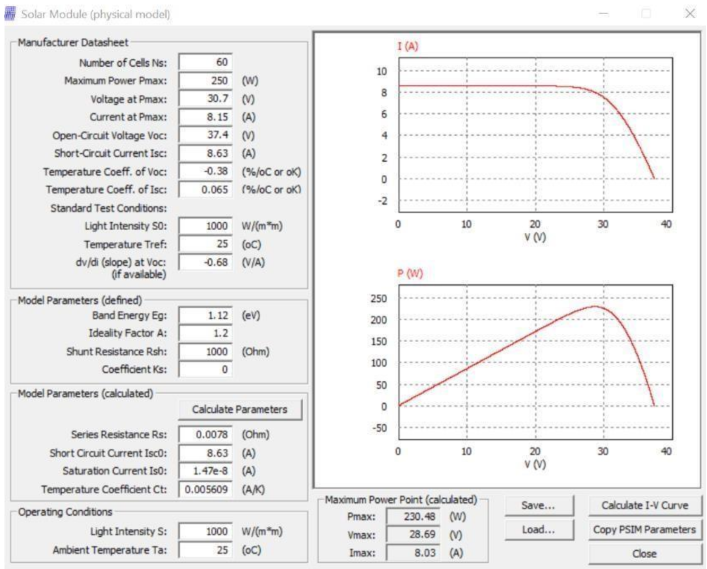


Figure 3-7: Specifications of Each Panel

3.4 Design of the MPPT

The P&O was implemented for this MPPT by following the P&O flow chart in Figure 2-7 and Figure 2-6:

Typically, the P&O flow chart controls V_{ref} however, as shown in the flowchart in Figure 3, the duty ratio is what is being controlled. Hence, there is no need to implement a PI controller in our simulation.

As shown in the circuit in Figure 1, the cell current and cell voltage are sensed by current and voltage sensors,

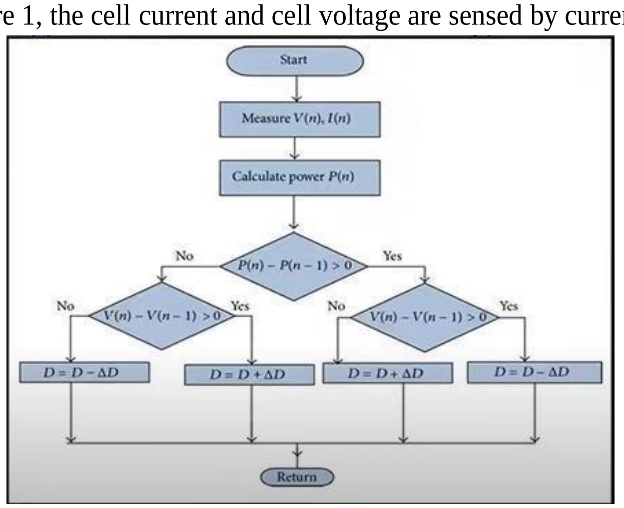
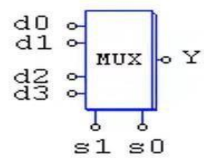


Figure 3-8 MPPT Algorithm Flowchart Used in the Simulation

they are then connected to a multiplier in order to calculate the actual output power of the PV array (step 2 in

The purpose of a unit time delay is to store the instantaneous value of a signal for a single simulation step. Therefore, two “Unit Time Delay” blocks were used to store the instantaneous values of V_{cell} and P_o. A “Summer (+/-)” block is then connected to each to subtract the stored values of V_{cell} and P_o with their instantaneous values to find $\Delta V\ and\ \Delta P$.

The comparator and “constant” blocks were used to find out whether the value of P_o is zero or not (step 3 in the flowchart). The “constant” block is set to zero, so if $\Delta P = 0$, the comparator’s output is 1. Otherwise, the output of the comparator is 0. Same procedure is applied to find whether $\Delta V = 0$ or not (step 4 in the flow chart).
In order to decrease or increase the duty ratio, a 4-input Multiplexer is connected to the comparators since there are four possible outcomes for the duty ratio as shown in the last step of the flow chart. The Multiplexer has 4 inputs and two control lines as shown in Figure 3-9 below.



The truth table is:

s1	s0	Y
0	0	d0
0	1	d1
1	0	d2
1	1	d3

Figure 3-9 Multiplexer Truth Table

where the truth table is also available. The comparators are then connected to the control lines with the power line connected to s0 and the voltage line to s1.
Two “Constant” blocks were connected to the 4 input signals. The values of those blocks were chosen to be 0.1 and -0.1, these are the values of $+\Delta D\ and\ -\ \Delta D$. Increasing these values would lead to a decrease in the maximum power tracking time and an increase in oscillations in steady state. However, the values 0.1 and -0.1 were found to be suitable for our design. “Constant” block (0.1) was connected to ports d2 and d1, while “Constant” block (-0.1) was connected to ports d3 and d0.
A “Summer (+\+)” block is the connected to the output of the multiplexer to add the value of ΔD to the new duty ratio, then connected to a limiter to limit the duty ratio between 0 and 0.85 which is connected to a “Unit Time Delay” block to store the instantaneous value of the duty ratio.
A triangular source to generate a PWM signal with 2500Hz frequency (frequency of the boost controller) is introduced to the controller, where it’s connected to the negative terminal of a comparator while the positive terminal is connected to the limiter introduced before. The comparator’s output is a PWM signal. Therefore, it is sent to the MOSFET’s ON/OFF controller.

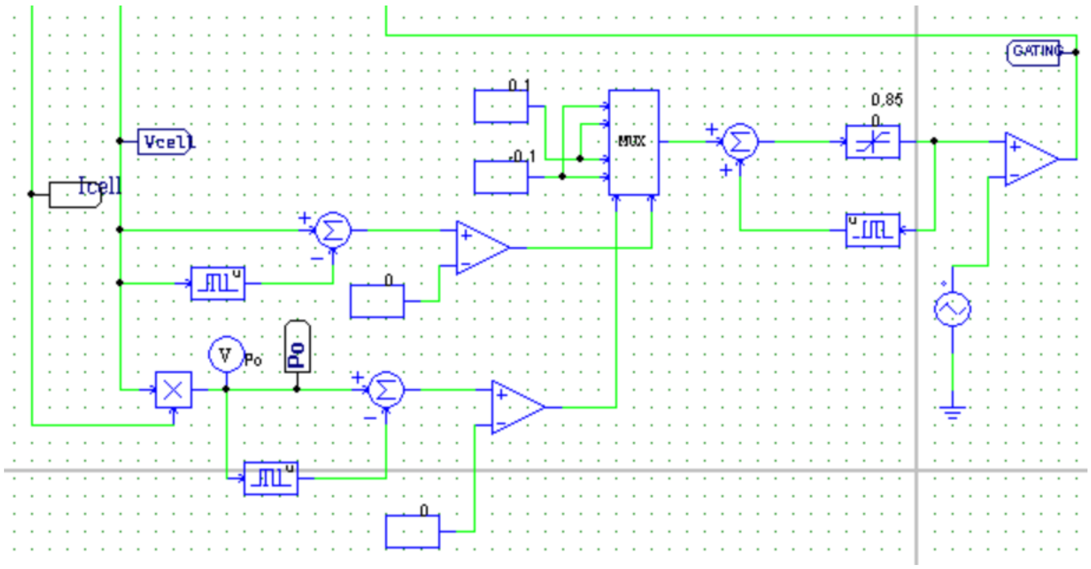


Figure 3-10: MPPT circuit

The simulation is then conducted to check whether the MPPT is tracking the power correctly:

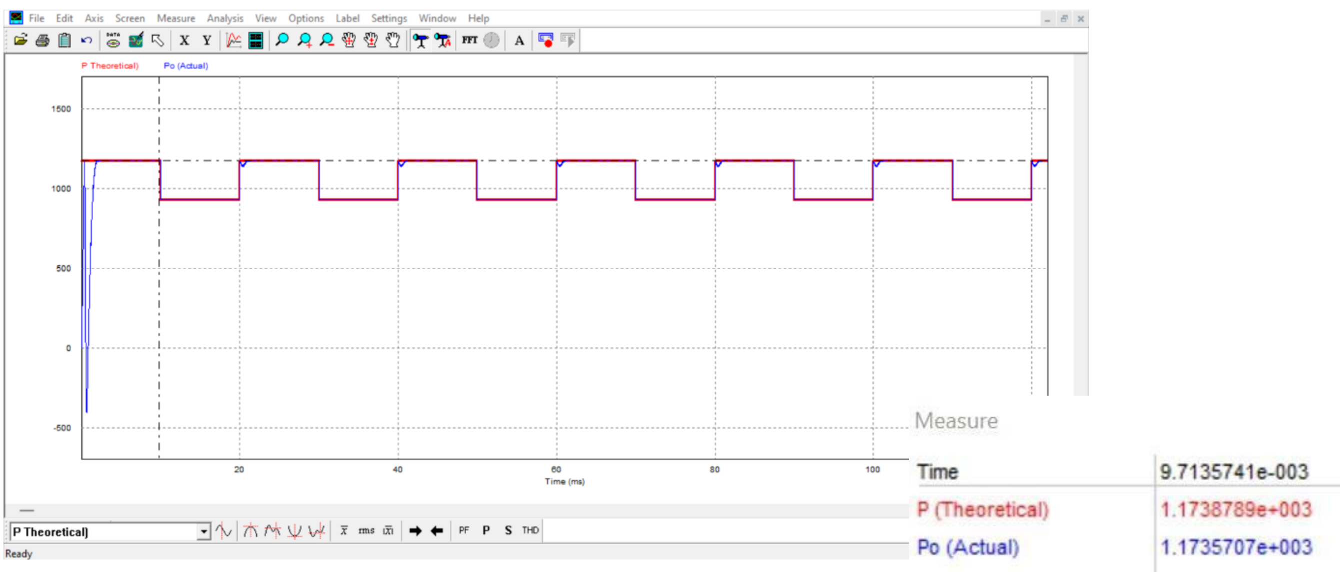


Figure 3-11 Actual vs Theoretical Power

As shown in Figure 3-10 above, there is a slight difference in between the theoretical and actual power, meaningthat the MPPT is able to track and control the power supplied by the solar array.

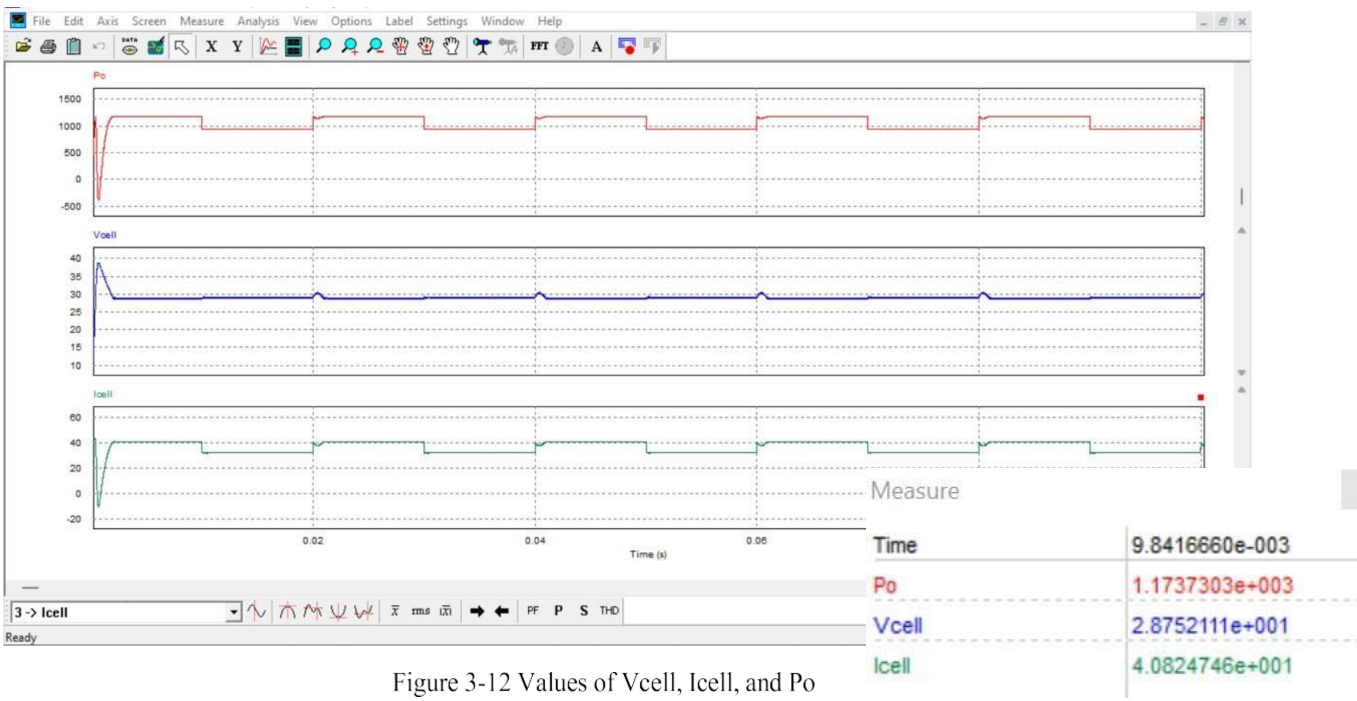


Figure 3-12 Values of Vcell, Icell, and Po

The values found are very close to the rated values of the solar panels, the voltage remained the same since the five panels are connected in parallel. However, the current increased since for the same reason I_{max} of each panel was found to be 8.15 A, meaning that a parallel connection with 5 panels would add up to 40.75 A. This value is considered to be very close to the value acquired from simulation (40.8247 A)

3.5 The Load

Initial Design:
As mentioned previously, the load will be controlled in 10% steps, meaning ten LED lights are going to be connected to the 24V battery in parallel with the battery. These lights are represented as resistors in the simulation, each connected to a switch, representing the relays to be controlled by the controller in the system. Since the system shouldn't exceed 500W, we chose to control the system in 35W steps for each bulb, meaning that the total size of the load will be 350W. Since the voltage supplied from the battery is $24V_{DC}$, The DC output current is:

$$I_{out} = \frac{P}{V} = \frac{350}{24} = 14.583\ A$$

And in order to have 10 resistors with 35 W each,

$$R = \frac{24^2}{35} = 16.457\ \Omega$$

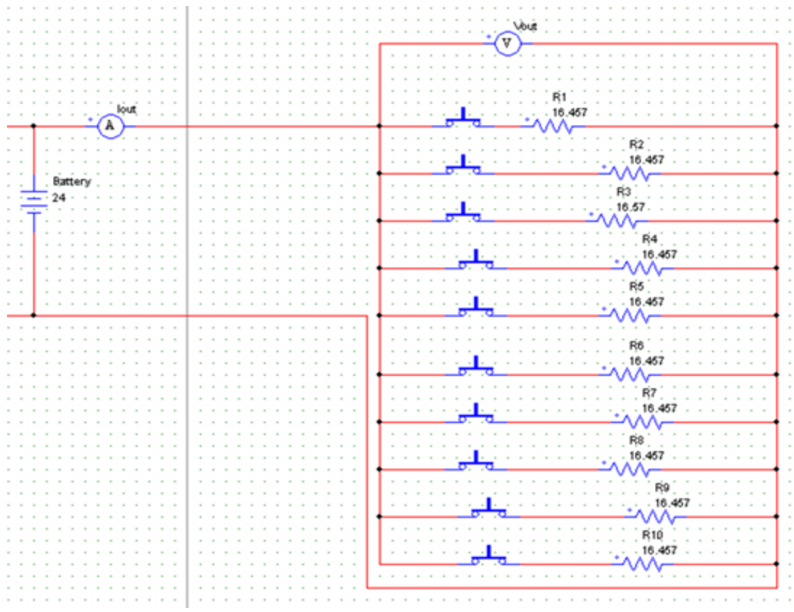


Figure 3-14 The Initial Load Design

As shown in Figure 3-14, each resistor is equal to 16.457Ω, connected to a switch to control the power flow.

When all switched are on, the following is obtained from running the simulation:

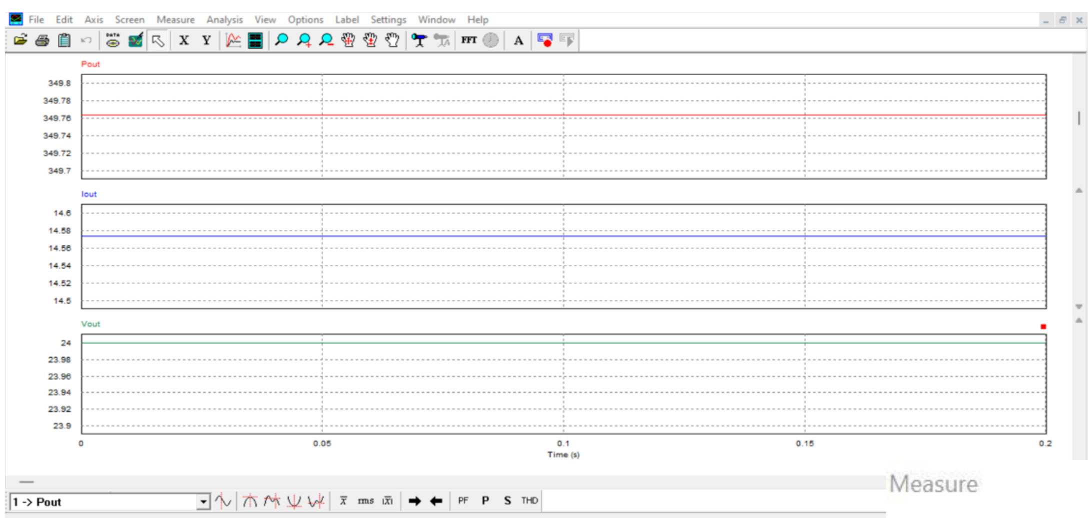


Figure 3-15: Output Values with all switches ON

The practical values were found to be extremely close to the theoretical values, where the output power was ≈ 350W with all switches turned on (closed).

However, when only half of the switches are switched on:

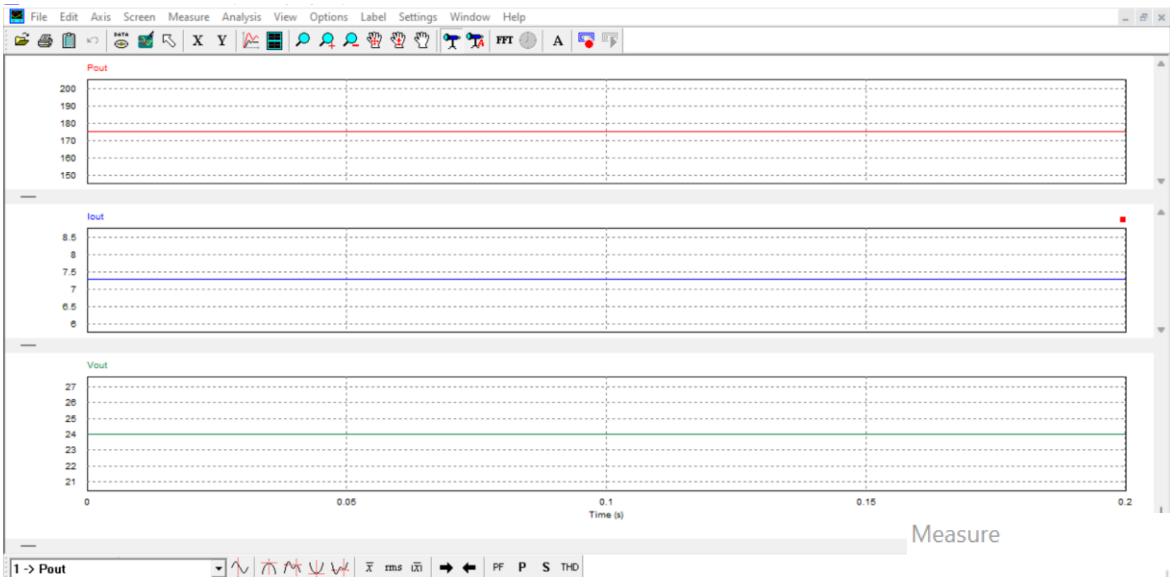


Figure 3-16: Output Values with Half of the Switches Open

-It can be concluded that when half of the switches were turned off, that the power and current reduced to half of their values, meaning that with every light (resistor in case of the simulation) that we turn off, we reduce the output power by 35W, reducing the current as well.

This design was switched for a load with lower power due to the unavailability of a 35 W load in the market. A different option was to connect the new, lower rated LEDs to a resistor to increase its power consumption, but the resistor had to have a high tolerance with a heat sink, which is also difficult to find on the market and costly.

Therefore, the final design included the following:

Ten LED flood lights, rated at 20 W each, drawing 0.8 A of current when connected to 26.2 V (battery voltage at maximum capacity), meaning that the full load would draw 200 W and 8 A of current.
The simulation was rerun to show the actual results obtained.

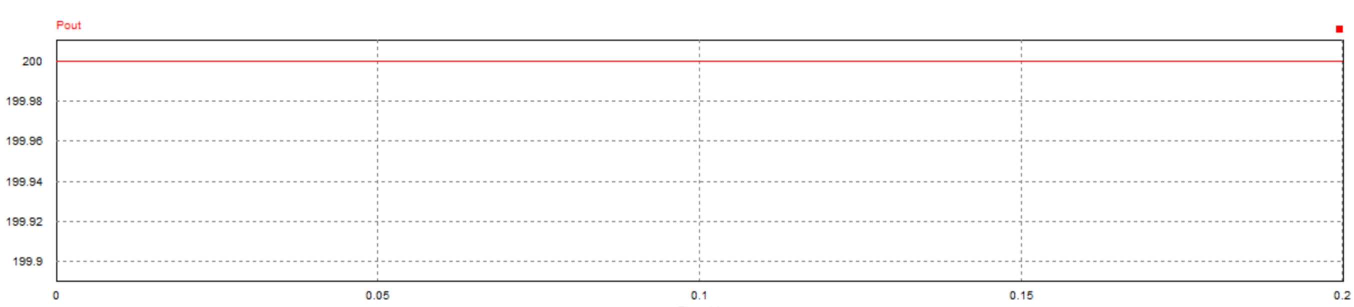


Figure 3-17 Output power using new flood lights

To choose the LEDs in the simulation, the same method used earlier was implemented, referring to the LEDs in terms of resistance. The LEDs were referred to as 28.8 Ω each and when full load is on, draw 200 W of power.

The lights used are shown in Figure (3-18).

Figure 3-18 Flood Lights Used

3.6 State of Charge Calculation

As discussed in previous chapters, the batteries installed in the Renewable Energy Lab are EverExceed Deep cycle AGM DP-12300. These batteries are rated at 12 V, 300 AH @ 20 HR Rate to 1.75 VPC. We contacted the manufacturer to get more information regarding the open circuit voltage of the battery to calculate the SOC curve. The following table (1) was sent by the manufacturer and used to do our calculations:

Table 3-1 Voc Vs SOC

Voc (V)	SOC (%)
26.2	100
25.8	75
25.4	50
25	25
24.4	0

Table (3-1) above was used to obtain the curve shown in Figure (3-19) below, and from the curve we were able to find an equation for the state of charge of the battery at rest

(SOC at Rest= 56.352*$V_{battery}$-1379.1).

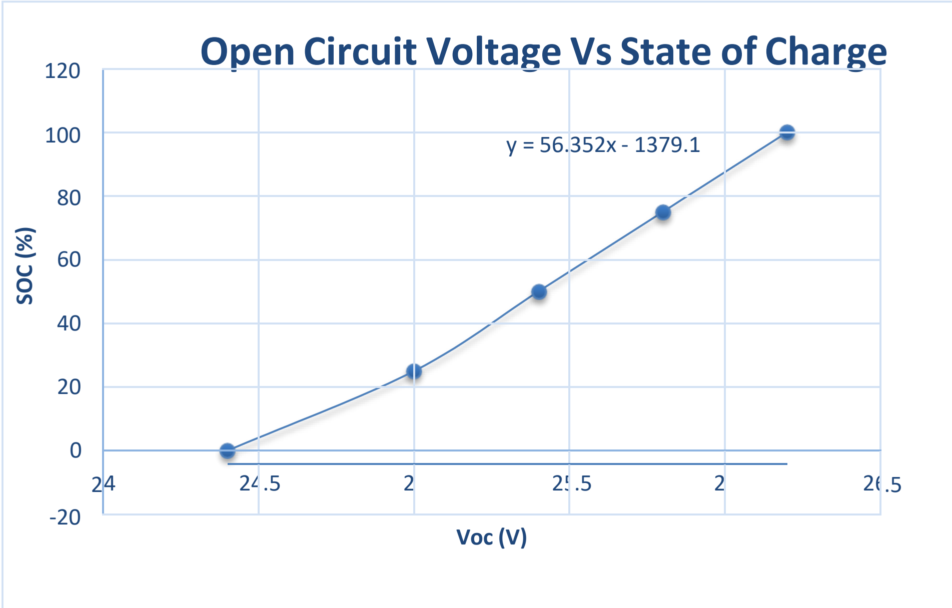


Figure 3-19 Open Circuit Voltage Vs State of Charge

We referred to table (3-2) below for some useful information to find the SOC.

Constant Current Discharging Ratings - Amperes Per Cell @ 20°C (68°F)											
End Point Volts/Cell	1.5hr	2hr	3hr	4hr	5hr	8hr	10hr	12hr	20hr	24hr	100hr
1.85	118	94.1	66.9	52.5	43.7	30.3	25.0	21.3	14.0	11.9	3.09
1.80	125	100	70.7	55.7	46.4	31.4	26.1	22.2	14.5	12.3	3.15
1.75	128	103	72.8	57.5	47.2	32.4	27.0	23.0	15.0	12.8	3.30

To calculate the energy (Wh) available in the battery, we must follow the equation below:

$$Wh = V_{battery} * I_{@\frac{c}{20}} * hours$$

The SOC at rest is useful to measure the available battery capacity.
To measure the available capacity, we used the following equation:

$$Available\ Capacity = \frac{SOC\ at\ rest}{100 * Wh}$$

The current @ c/20 can be obtained from the table, where for a 1.75 VPC battery, the current is 15 A for 20hr. Since we have four batteries, two of them connected in parallel, the current is multiplied by two, giving us 30 A.

The discharge time depends on the available capacity, battery voltage, and the current consumed by the load (depends on how many LED floodlights are turned on. Therefore, the following equation is used to calculate the discharge time:

$$Discharge\ time = \frac{Available\ Capacity}{V_{Battery} * I_{Load}}$$

The load time was updated each minute.

The energy (Wh) consumed by the load = $V_{Battery} * I_{Load} * Load\ Time$

The capacity after consumption = Available Capacity (Wh) – Energy Consumed by the Load (Wh)

$$State\ of\ Charge\ of\ The\ Battery = \frac{Energy\ Consumed\ by\ the\ Load}{100 * Wh}$$

This series of equations used to find the state of charge was used to program our system and make it possible to detect how much charge is left in the battery and display it on the 10-segment display.

3.7 Microcontroller Design

Initial Design:

3. Processor 16F877A

The Microcontroller PIC16f877a has been one of the industry's most well-known micro-controllers. This micro - controller is very easy to use, and its programming is also quite simple. Because it employs FLASH memory technology, one of the great benefits is that it could be write-erase so many more times as needed. It features a total of 40 pins, with 33 pins dedicated to input and output. Many pic micro - controller applications use the PIC16F877A. PIC16F877A is also widely used in digitized electronic circuitry. The PIC16f877a may be found in a wide range of devices. Smart sensors, safety devices, home automation, and a variety of industrial instruments all use it. It also has an EEPROM, which allows it to store some information permanently, such as transmitter codes and receiver frequencies, as well as other associated data. This controller has a minimal cost and is simple to operate. It's adaptable, allowing it to be utilized in fields where micro-controllers haven't been employed previously, such as micro-processor apps and timer functions. It comes with a smaller set of 35 instructions. It can run at a frequency of up to 20MHz. Between 4.2 and 5.5 volts is the operational voltage. If it has been given more over 5.5 volts, it might be irreversibly destroyed. It lacks an internal oscillator, as do other PIC18F46K22 and PIC18F4550 chips. Each PORT may sink or source a maximum current of roughly 100mA. As a result, each GPIO pin on the PIC16F877A has a 10-mile ampere current limit. And within this project, it is the brain that in charge of everything...

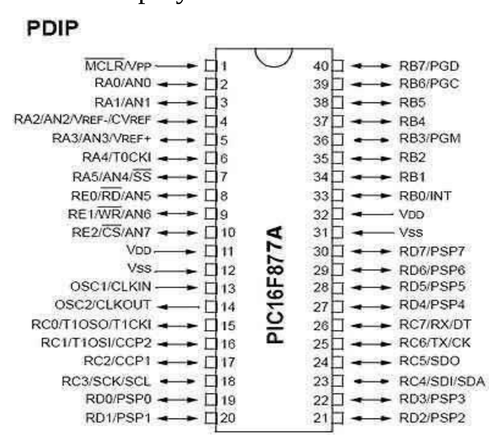


Figure 3-20 Processor 16F877A

4. Voltage divider

A small circuit that transfers the voltage from the batteries to the processor so that its measurement is from 0 to 5 volts so that the processor can read it

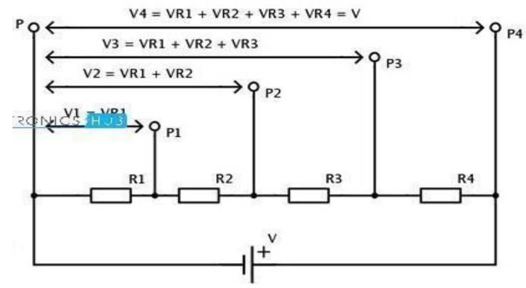


Figure 3-21 Voltage Divider

3 ACS712 20A CURRENT SENSOR

it's a ACS712 linear current sensor, which is based on the Hall Effect. For both Ac / Dc signals, the sensor provides exact current measurement. The gadget can survive up to five times under overcurrent circumstances due to the robust conductive material as copper plus signal traces. The ACS712 generates a linearly varying analog output voltage signal in response to detected current. VCC is 5VDC, and there are a few of filter capacitors required. Keep in mind that, while the ACS712 is certified for 2.1kV isolator, the PCB it's mounted on isn't suitable for that voltage. In this project, it is responsible for reading the consumed current and sending it to the processor.

Features:

- Proportional output voltage to AC || DC current
- Precision-trimmed at the factory
- Extremely stable offset voltage output
- Magnetic hysteresis is almost non-existent.
- Radiometric output as a result of the supplied voltage
- Blue in color
- Composition: PCB + metal
- Sensor chipset ACS712ELC-20A
- Requires a 5V power supply
- Power indicator on board
- Measures a current of -20+20A, with a simulated output of 100mV/A.
- While on current got measured, the output voltage was VCC/2.

Figure 3-22 ACS712 20A CURRENT SENSOR

- Because the ACS712 is founded on the Hall detection principle, it is important to prevent the magnetic field effect when utilizing it.

3.1.1.1 Tri-state-buffer

A tri-state buffer is works similar to a buffer, but it has an extra "enable" input that determines whether or not the main input is transmitted to the output. The tri-state buffer works like a conventional buffer if a "enable" incoming signal is true. The tri-state buffer transmits an infinite resistance (or hi-Z) signal if the "enable" incoming signal is false, thus disconnecting its output from the system. It oversees transmitting current from the panels to the batteries or from the batteries to the loads in this project.

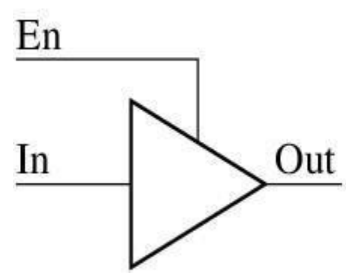


Figure 3-23 Tri-state Buffer

3.1.1.2 *20X4 WHITE ON BLUE CHARACTER LCD*

This is a 4 rows x 20 columns L-C-D screen with white text on a BLUE backdrop and backlight that is controlled by an HD44780. Because this LCD screen uses a parallel interface, you'll use 7 G-P-I-O pins for 4-bit mode and 11 G-P-I-O pins for 8-bit mode to connect to it. It is in charge of showing all of the data that the user requires in this project.

Specifications:

- Dimensions of the module (H x W x T): 60mm x 98mm x 14mm
- 40mm x 98mm black metal bezel (Hx W)
- •25.2mm x 76mm viewing area (H x W)

Features:

- High contrast and a wide viewing angle are two of the features.
- Built-in LCD controller that is equivalent to the industry standard HD44780
- LED lighting with a voltage of +5V DC
- There is no need for an additional power source for the backlight.
- 4-bit or 8-bit in parallel interface support
- Show a four-line, twenty-character display
- Use a 5V DC power supply.
- Male header with 16 spots available for free

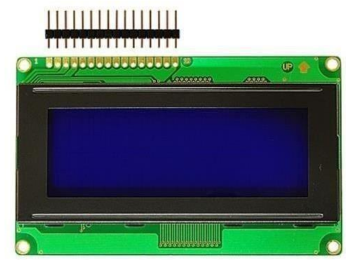


Figure 3-24 20X4 WHITE ON BLUE CHARACTER LCD

4 *TMLM 04225 Traco power supply*

The switching power supply of the TMLM Series provide the maximum energy capacity in a fully encased module that can be attached directly to PCBs. These modules are appropriate for all space-critical uses in industrial and commercial electronic devices because of this capability. The product has received international safety clearances, allowing it to be sold in all markets across the world. These Power Supplies have a high level of reliability due to SMD technology and great efficiency.

Figure 3-25 TMLM 04225 Traco

Features:

- PCB attaching AC/DC power module
- Maximum power density
- Plastic case that is completely encased
- Input range: 90–264 VAC, 47–440 Hz
- High productivity
- EMI is compliant with EN 55022, class B, and FCC, level B.
- Low noise and ripple
- Overload and short circuit protection
- 3-year warranty on the product

7. KEYPAD 4X4

- 500-ohm contact resistance
- 100M insulation resistance ()
- 150-200N KEY OPERATING FORCE
- 1 rebound time (MS)
- 100-million-year life span (times)
- temperature of operation 60 (C)

Figure 3-26 KEYPAD 4X4

The Electronic Features

- 35V (DC)100mA1W circuit rating
- Contact resistance: ten thousandths of a millionth of a millionth of a millionth of a
- (Varies depending on lead lengths, which are not the same as those of the material used)
- 100M x 100V insulation resistance
- Dielectric Strength: 250VRms (1min at 50-60Hz)
- Electric shock jitter: 5 milliseconds
- Tactile type life span: one million times

3.5.1 Developed Design:

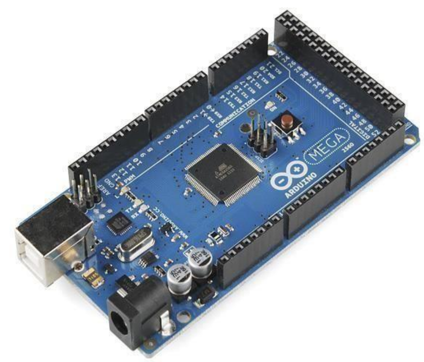


Figure 3-27 Arduino Mega 2560 R3

1. Arduino Mega 2560 R3:

The PIC microcontroller was swapped for an Arduino Mega 2560 R3 due to an issue faced during testing where we found out that the PIC didn’t have enough program memory and SRAM to be able to implement our code, in addition to that, a relatively large current was returned from the relays to the PIC while testing which damaged the PIC. Another reason we switched from the PIC to the Arduino is that it’s a newer technology, which motivated us to use.

The Traco Power supply was no longer needed after the introduction of the Arduino Mega 2560 R3 since it has a power input embedded in it that can be wired to a power socket immediately.

Features:

- ATmega2560 microcontroller
- Input voltage - 7-12V
- 54 Digital I/O Pins (14 PWM outputs)
- 16 Analog Inputs
- 256k Flash Memory
- 16Mhz Clock Speed

2. Relay Module 5V DC 2-Channel:

Five of these two-channel relays were used to control the ten loads, they control which lights switch in and which switch off, also the relays are connected to the Arduino Mega 2560, where it is programmed to shut the system down whenever the battery 20% charged.

Technical parameters:

- Voltage version:5V
- Static current:4mA
- Working current:135mA
- Trigger voltage:2-5V
- Trigger current:2mA

Figure 3-28 Relay Module 5V DC 2-Channel

3. LOC110 Single Linear Optocoupler

We didn't end up using the ACS712 20A current due to some issues faced during testing, where it was difficult to calculate the voltage of the batteries as it is constantly changing during charging and discharging. Therefore, we couldn't gat an accurate voltage reading. We also tried a sensor cable of measuring the current, the voltage, and the power (INA219 I2C bi-directional DC current/power monitor sensor). This sensor was functioning properly and giving us accurate readings, but the issue was that it had to be connected to a multi-ground, the ground of the sensor had to be connected to the ground of the power source we were testing on. While this may work for a system powered by a power source where the negative terminal acts as a ground for the system, it won't work when connected to a battery, as the battery's negative terminal isn't the same as the ground and might carry some current. Therefore, we opted for the Single Linear Optocoupler to be able to effectively measure the voltage and isolate the sensor. That way, a multi-ground isn't needed.

Figure 3-29 LOC110 Single Linear Optocoupler

The following circuit was built to implement the intended design:

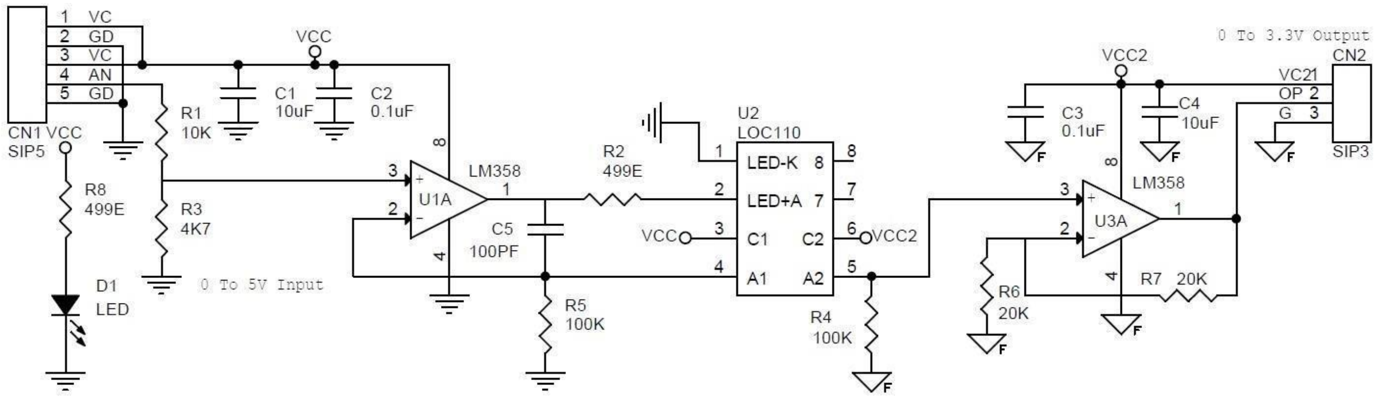


Figure 3-30 Schematic of Optically Isolated Analog Input Module for Arduino

4. SERIAL BLUETOOTH 4.0 BLE MODULE

The Bluetooth module was added later in the design to be able to control the system using a smartphone app.

Figure 3-31 SERIAL BLUETOOTH 4.0 BLE MODULE

5. Bluetooth Module HC-05

Two modules were used for the following reasons:

a. To communicate between the PC and the Arduino, where it becomes possible to control the system using the PC instead of using the keypad and LCD screen that were to be used in the initial design. It is also used for plotting the data (graphs).

b. It is used to store the data for it to be used in case machine learning was implemented in the future.

Specifications:

- Bluetooth protocol: Bluetooth Specification v2.0+EDR
- Frequency: 2.4GHz ISM band
- Modulation: GFSK (Gaussian Frequency Shift Keying)
- Emission power: ≤4dBmClass 2
- Sensitivity: ≤-84dBm at 0.1% BER
- Speed: Asynchronous: 2.1Mbps (Max) / 160 kbps
  Synchronous: 1Mbps/1Mbps
- Security: Authentication and encryption
- Profiles: Bluetooth serial port
- Power supply: +3.3VDC 50mA
- Working temperature: -20 ~ +75 Centigrade

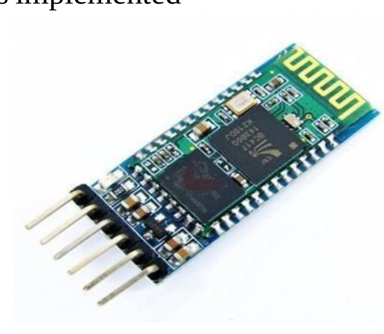


Figure 3-32 Bluetooth Module HC-05

- Dimensions: 15.2x35.7x5.6mm

6. <u>10-Segment Display CS-V1105</u>

The 10-segment display was connected to the Arduino Mega 2560 to display the battery percentage. For instance, the battery has 30% left in it, three bars light up on the 10-Segment Display.

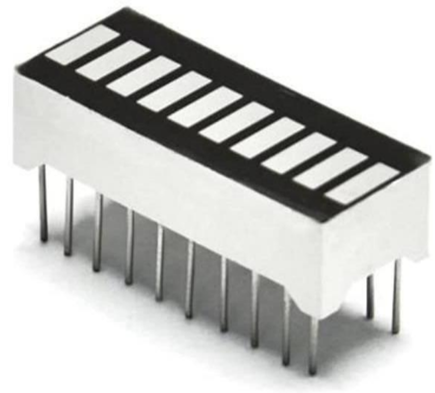


Figure 3-33 10-Segment Display CS-V1105

Specifications:

- Forward Current (Per Segment): 25mA
- Max Reverse Voltage: 5v
- 10 segment bars
- Color: Super Bright Red
- Industrial standard size
- Low power consumption
- Categorized for luminous intensity.

7. <u>DC Fuses:</u>

Fuses were added between each relay and load for added protection to the loads. The fuses are rated at 1 A and are designed to ensure that the system is further protected.

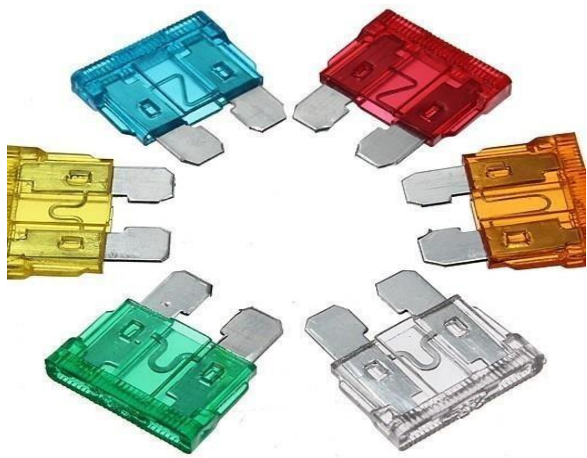


Figure 3-34 1 A DC Fuses

3.8 System Wiring Diagram

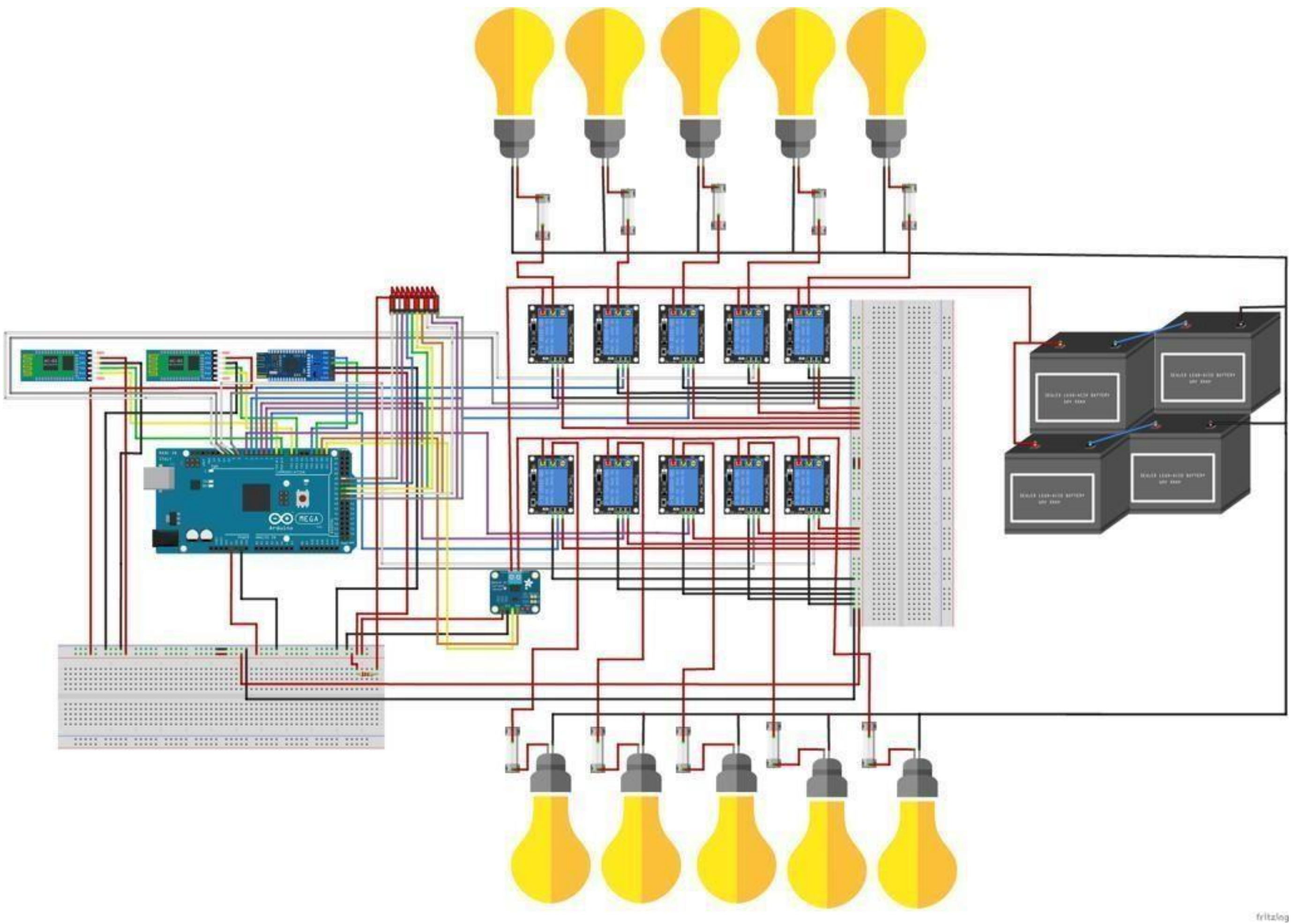


Figure 3-35 System Wiring Diagram

- <u>State Machine:</u>

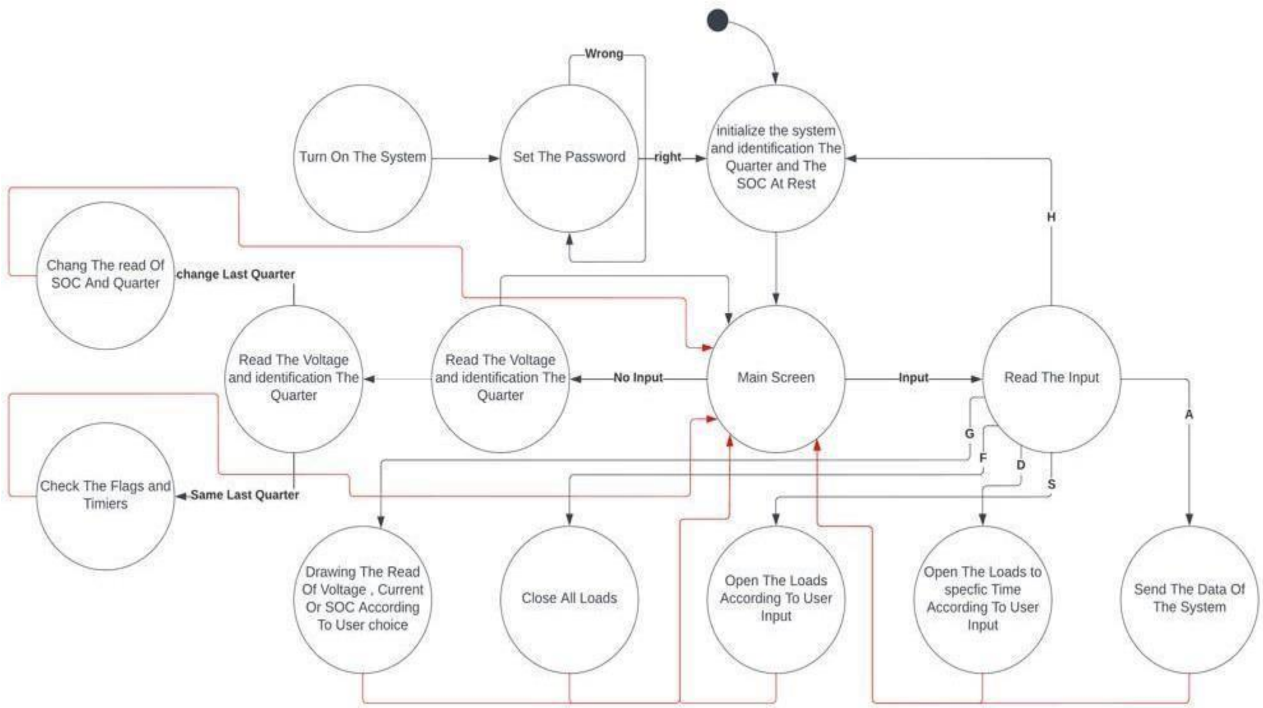


Figure 3-36 State Machine of The System

- <u>The Flowchart of The System</u>

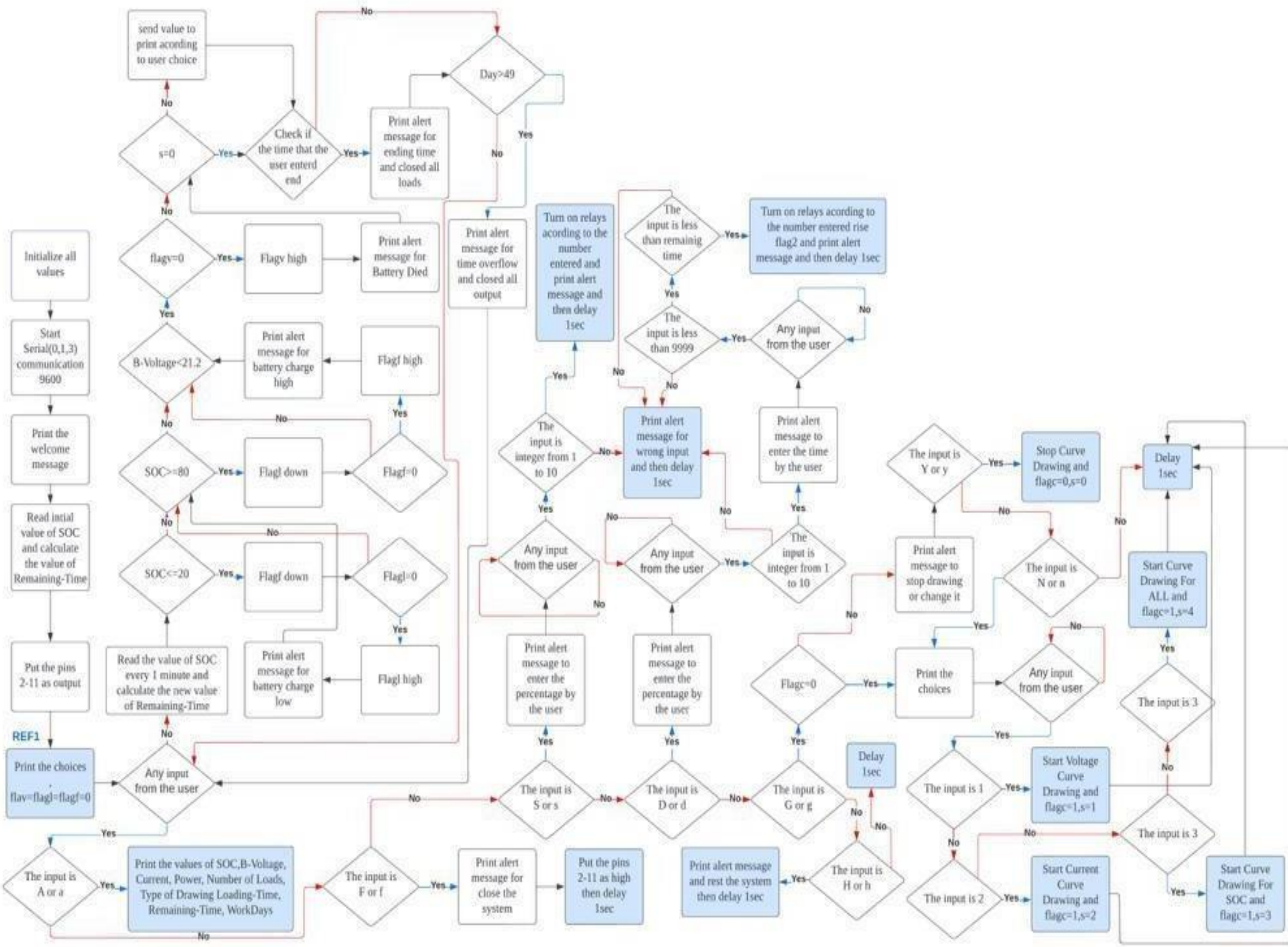


Figure 3-37 The Flowchart of The System

3.9 Developed Software

Turning on the system by the engineers (Us), or in other words giving the system the needed power, this will make the system at first Initializing all values and beginning to monitor the sensorafter we include the sensor's library. See appendix (C) the code appendix color:

Second starting the serial communication (9600 bit per second), one for drawing and programming, one for mobile connection or communication, and the other for saving data.

Note that the serial communication happens through connecting a Bluetooth module (4.0 BLE) "not programming it" to a mobile application called (Arduino Bluetooth controller HM-10) and connecting a Bluetooth module (HC-05) to Putty program for saving data and connecting another Bluetooth (HC-05) to the Arduino and IDE software directly for programming the systemand drawing info as curves.

So, the user types or enters some certain data through mobile application, the Bluetooth module send or transmit this data to the system basically to the Arduino (the brain of the system), the Arduino do some data processing and sending or transmit the processed data to the Putty program and IDE program through Bluetooth module to saving and drawing the processed data.see appendix (C) color:

Third, printing the welcoming message and ask the user for a password which is "PSUT" if the user type it correctly, then the user could enter and control the whole system. see appendix (C) color:

The system will start to calculate initial value for the SOC (state of charge) and the discharge remaining time. the first value , which is the SOC , is calculated through an equation that was reached by inserting the values of the open circuit voltage that belongs to the batteries used in the system and inserting the corresponding percentage of SOC in the excel program and calculate through this program the linear equation that combined between the voltage and the SOC through curve fitting method, note after we find the equation we insert it in the code also note that the opencircuit voltage is the sensor's voltage . see appendix (C) color:

Regarding the second value which is discharge remaining time, we need to extract someinformation from battery's data sheet like:

1. Current C rate, note and according to the datasheet we found that the capacity of our used 12- V battery equal to 300Ah at 20 Hr (hour rate) to 1.75 VPC (voltage per cell) and we connected 12- V four batteries every two of them in series and then connecting the series 24-V batteries in parallelwith each other so, we found that the current discharging rating equal to $15 \times 2 = 30A$

2. The number of discharging hours, which is how long it takes for battery to die and according to thedatasheet it equals to 20 hours.

And collecting the other needed information from the system like:

1. The sensor's voltage
2. The SOC at rest (the initial SOC) that we mentioned earlierand computing or calculating using excel program like:

    a. Wh (battery's Watts per hour)
    b. The available capacity which is the amount of energy stored in the batteries while the system working or turned-on

To calculate the discharge remaining time and insert it in the code. See appendix (C) color: ,
Fourth, the system will start to divide the batteries into four levels or quarters:
0% - 25% which equals to (24.4V-25V) 25% - 50% which equals to (25V-25.4V) 50% - 75% which equals to (25.4V-25.8V)
75% - 100% which equal to (25.8V- above 25.8)

And saving the number of levels into two integers on of these integers represent for the present level (z) and the other one represents for previous level (v), this division will happen every small interval of time, exactly before the do statement and in it, and in the upcoming texts the reader going to understand why this division have been made. see appendix (C) color:

Fifth, the system going to print out on the mobile screen the initial state of charge (SOC) and turning off the relays by first making them as an output then sending ones to the pins that theyinstalled at. see appendix (C) color:

Sixth, the system then needs to show the options that the user needs to choose like:A- For system's information
S - for load controlling

D - load controlling with specific intervalF – for turning of the load
G – for showing the information as curvesH – for resetting the system

See appendix (C) color :

And waiting the user to type or enter one of these characters, while the system is waiting for aninput and in the do while statement specifically in do statement the system start to collect the sensor's voltage and current which is the battery's voltage and current, and the division that previously mentioned , happens again in the do statement but this time saving the number of levelor quarter in one integer which is (z) and then the system starts to comparing the previous levelwith the present level if they were equal the system going to compute the value of SOC and remaining time depending on the new capacity which is the changed battery's capacity . see appendix (C) color:

On the other hand, if they are not equal the system going to compute a new SOC and new remaining time for the new quarter. see appendix (C) color:

then the system going to send the collected information (SOC, voltage, current) of the system to Putty program (a program to receive save data we mentioned before). see appendix (C) color:

the system will check on lots of conditions in the "do" statement:

1. The resultant SOC will be compared to the minimum percentage of charging which is 20% and with maximum percentage of charging which is 80%, if the resultant SOC equals to or less than 20% the relays will turn off by itself and if SOC equal to 80%.
2. The sensor's voltage if it was lower than $1.75\ VPC\ (voltage\ per\ cell) \times 6\ cells \times 2(two\ batteries\ in\ series) = 21.2\ V$ the system will send a warning message conclude that the battery died, and the system will turn off the loads by itself.
3. Drawing orders, the system will check if there is any order for drawing and that's accordingto (s) value if it was not equal to zero and equals to (1 or 2 or 3 or 4) this will make the system sends data to the IDE software to draw the needed piece of information in form of curve.
4. Entered time, the system will check if the time that user entered is end if so the system byitself going to turn-off the loads and print a warning message that the time is end.
5. Number of overflow days, the system will check the number of days if they are equal to 50to make an overflow
   See appendix (C) color:
   Note that all these conditions will be checked by the system in very short time because every line of code needs 2 ns (for execution, so the conditions will be checked before the user even think in entering an input. Also note that this process will stay until the user inserts an input)

6. If the user types or enters any choice that we previously mentioned (A, S, D, F, G, H) thesystem going to work as follows:

7. The system going to make the flags (flag f, flag l, flag v) equal to Zero, the reason behindthis is to make the system return to check the conditions again and again

8. If the user types or enters “A” or “a” the system going to send the collected and processedinformation to the mobile

   The sent information is:

- SOC: that we are calculated and processed it
- The present Battery’s Quarter
- Battery’s voltage
- Current: that flows in the system
- Power: that the load consumes, or the battery gives
- Loads: how many loads are on in percentage
- Loading-time: how long has the system been running in minutes
- Remaining time: how long it takes to reach 20% of its stored energy
- Workdays: how long has the system been running in days

   Note that when the user chooses “A” or “a” the system will check the potters drawing if it will draw current or voltage or SOC or all of them together, according to the value of “s” that the user entered in “G” choice.

   Note also if there is no choice entered by the user regarding “s” value nothing going to happen,there will be no curve to draw. See appendix (C) color:

9. If the user chooses “F” or “f” the system will turn-off, the relays going to open and theloads going to extinguished and the system going to print “system is closed”.
   see appendix (C) color:

10. If the user choose “S” or “s” he will be given a choice from 10% up to 100% which is the number of loads that the user wants to turn-on , if the user types or enters a wrong percentthe system will warn him, and then the system going to give the user about 5 seconds for choosing process if this process exceeded the 5 seconds the system will return to the calculation section (the do statement section) and start to calculate again . note that when the system turns-on the loads it will turn off the others See appendix (C) color:

11. If the user chooses “D” or “d” which is controlling the loads with certain interval.The system will give the user two choices:

- One for entering the number of loads from 10%-100%, if the user enters another valuethe system will not take it
- The other one for entering the duration of turning-on from 0 up to 4-digits of minutes,and the user has the maximum limit of the 4-digits reaches to 9999 minutes.
  Note here the duration must be less than remaining time to make sure that there is enough energy in the system. After the duration end the system will turn-off the loads.And the system will warn the user if he enters a wrong duration, see appendix (C) color:

12. If the user chooses “G” or “g” which is drawing the information of the system as curves, the system will check if the curves were drawn previously or nothing has been drawn through flag c.
    If nothing has been drawn, the system will give the user the drawing choices, either to drawcurrent or voltage or SOC or all of them together on the same graph, if one of the curves

have been drawn, the system gives the user the choice either to stop the drawing or changingthe curve.

- Stopping: the curve the value of “s” become zero and the system will make the c flagzero and the curve will stop running
- Changing: the system will display the choices again to the user and make the c flag zeroand depending on the user’s choice the drawing will start. see appendix color:

13. If the user chooses “H” or “h” the system will turn off the loads and do an initializationprocess for all values and start to calculate the SOC and the remaining time just like the beginning of demonstration. see appendix color:

3.10 Application of Engineering Standards:

Referring to Directive 2014/35/EU of The European Parliament and of The Council of 26 February 2014, the following items were used to ensure that the system is safe for use:

a. The system was equipped with a plastic covering as a barrier to obstruct the light shun by the LEDs to protect the user’s eyes from the light. (Shown in Figure 4-2b).

b. Fuses were added after each relay to further protect the loads from short circuit current.

c. The controller is installed underneath the LED lights as shown in Figure 4-1. Therefore, the lower part of the system needed to be covered to ensure that all wires are far from reach and safely hidden as shown in Figure 3-22 below.

d. Non-conductive material was used to design the base and covering of the system for additional safety. (Wood and Plastic)

e. System was designed to be controlled wirelessly through PC or smartphone, protecting control devices from any faults that might occur. The wireless connection also eliminates the need for physical contact with the system.

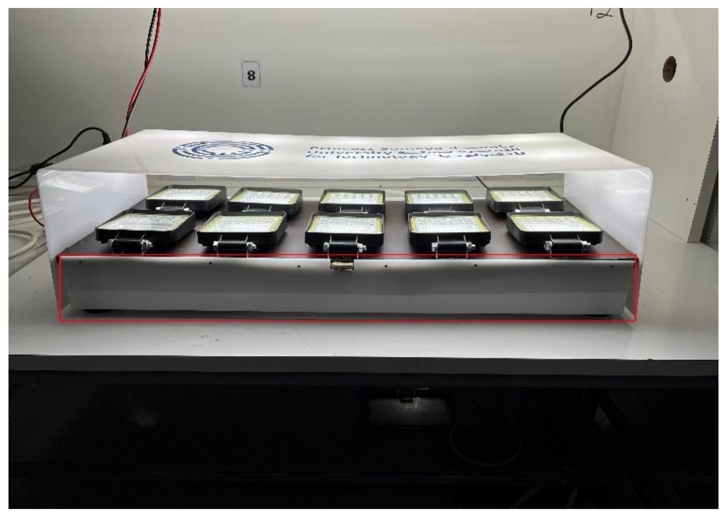


Figure 3-38 Wire Covering

f. Wires used in the system were carefully chosen to account for any temperature and derating factors that might affect the wires. Therefore, a (2x 2.5mm) Cu/PVC/PVC cable was used to connect the batteries to the system since it holds the highest current and 1mm CU/PVC/PVC wire was used to connect the lower current components (between each relay and load).

4 Results

4.1 Prototype Setup:

4.1.1 Developed Hardware:

After following the design restraints mentioned earlier in Chapter 3, the system was put together and tested using a DC power supply, which was used to represent the battery supplying our system with power. Figure (4-1) below, shows the physical connection of thecontrol system.

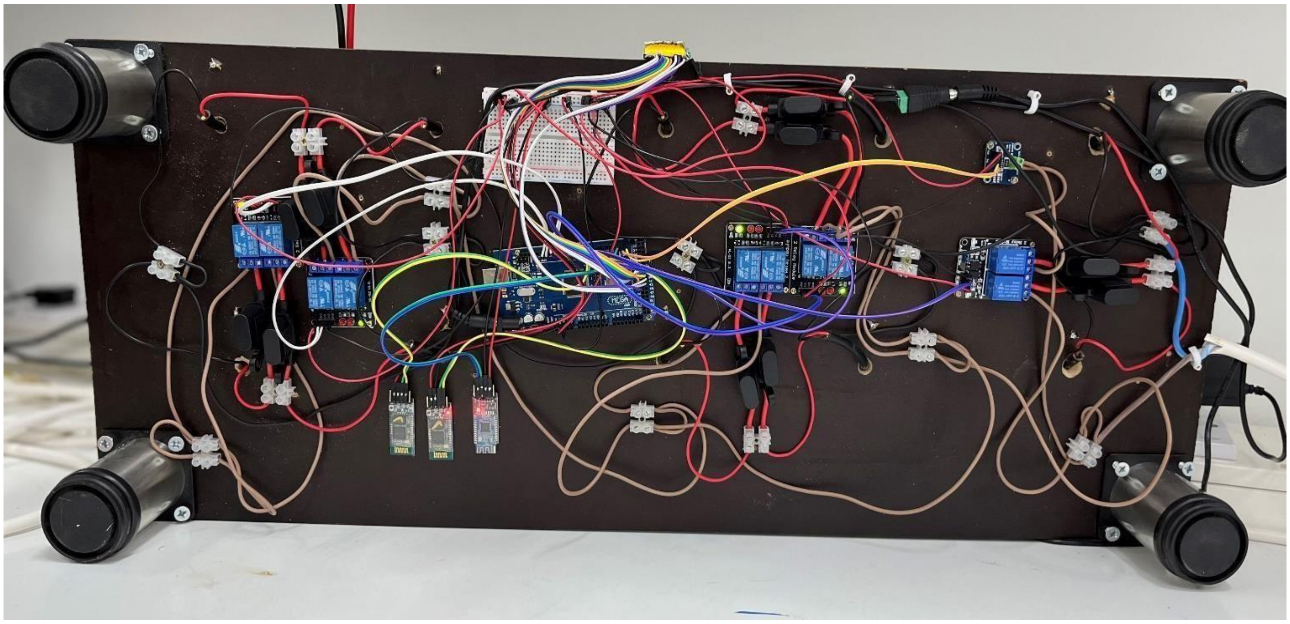


Figure 4-1 Controller Setup (Bottom View of Prototype)

Figures (4-2a), (4-2b), and (4-2c) demonstrate the final design of the prototype. The ten LED lights were drilled onto a piece of wood with the controller connected underneath it. A plasticcovering with the university's logo was used to protect the user's eyes from the intensity of the lights while still partially visible for observation. The 10-segment SOC indicator can be observed at the bottom of the prototype with a magnifying lens to make it easier to read it as shown in figure (4-2a) and (4-2d).

Figure (4-2d) 10-segment SOC with magnifyinglens

Figure (4-2a) Prototype Design (The indicator shows the 10-segment SOC display)

Figure (4-2b) Prototype Design (Top View)

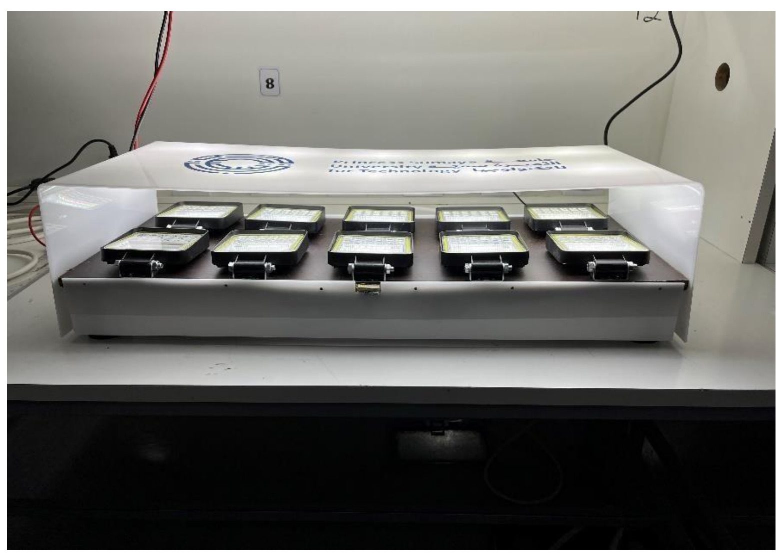


Figure (4-2c) Prototype Design (Side View)

4.1.2 Software:

The Arduino Mega was programmed to perform the tasks necessary. The interface to control the system was both available on PC and a smartphone app, both using wirelessconnections.
Below are figures showing the code and the interface on both PC and the smartphone app.

- **Smartphone Application [Arduino Bluetooth Controller (HM-10)]:**

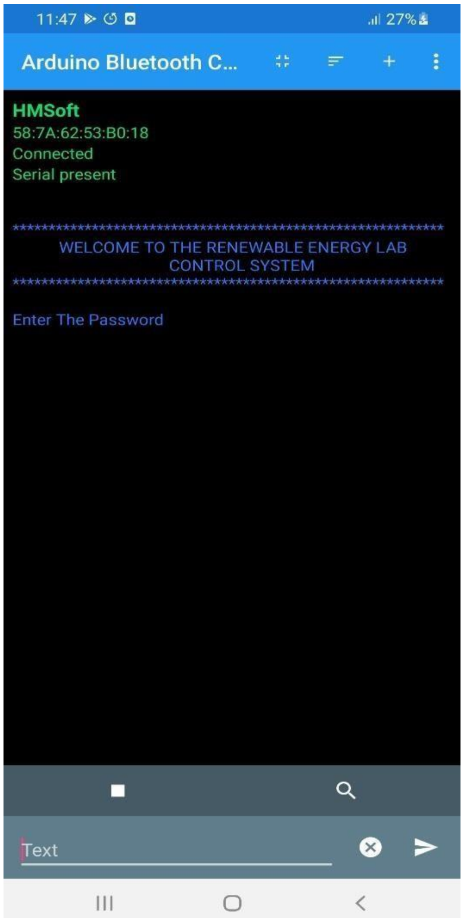


Figure (4-3a) Home screen where password has to be entered

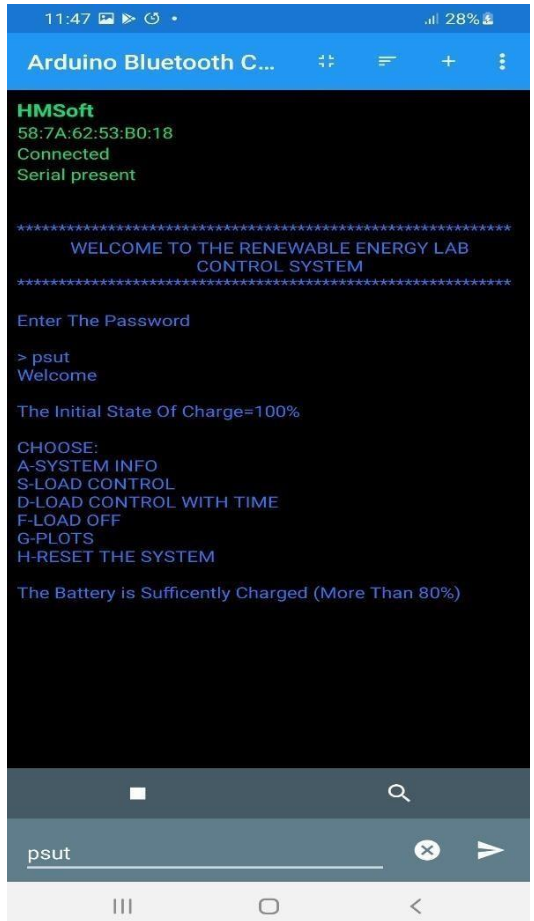


Figure (4-3b) After entering password

Figure (4-3b) shows the interface after entering the password. The following six options to choose from pop-up:

A- **System info:** This option provides the user with the system information as shown in Figure (4-3c).

After choosing option A to print the system information, the user is provided with the current state of charge of the battery, its voltage, current being drawn, power consumption, and how many loads are on (percentage of loads on). This option also provides the user with the elapsed time and the remaining time before the system is shut down, in addition to how many days the system has been on for.

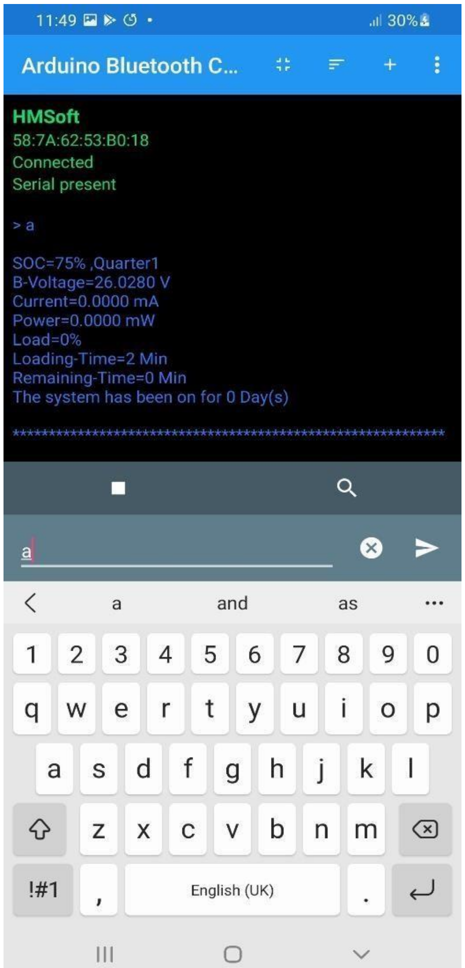


Figure (4-3c) System Information

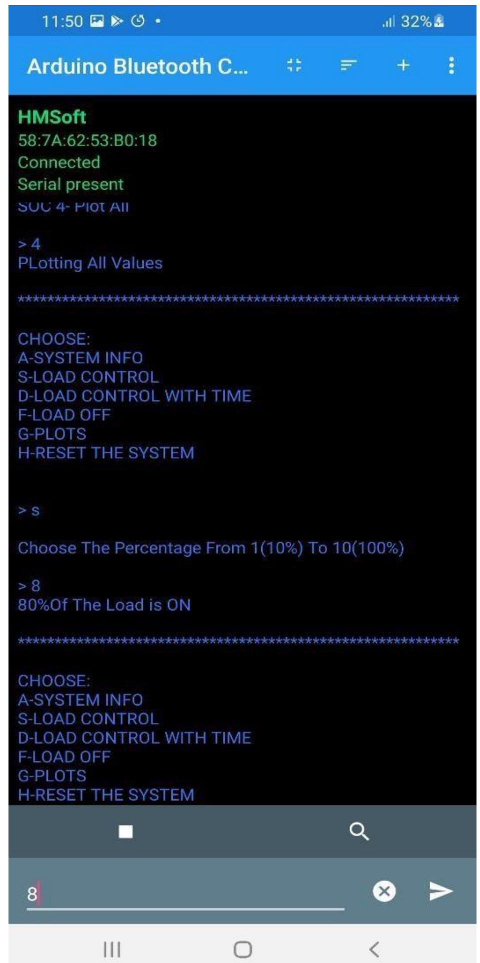


Figure (4-3d) Load Control

S- **Load Control:** Displayed in Figure (4-3d), the load control option gives the user the ability to control the percentage of the loads that are to be switched on. For instance, in Figure (4-3d), the user chose the value 8, meaning that 8 out of 10 LED floodlights are to be turned on. Meaning that 80% of the load is ON.

D- Load Control with Time: This option enables the user to switch on the desired percentage of the load for a certain time. In Figure (4-3e), the user chose to turn on 20% of the load for two minutes. A message on the screen informs the user that the load will turn OFF automatically after 2 minutes.

H- Reset the System: used to reset the system when desired.

Note that the system calculates the SOC every minute, if the battery is low (SOC <20%), a message pops-up and the system automatically shut down the load to protect the batteries from undercharging and prevent any damage to the battery. It also displays a message whenever the battery is sufficiently charged (80% or more).

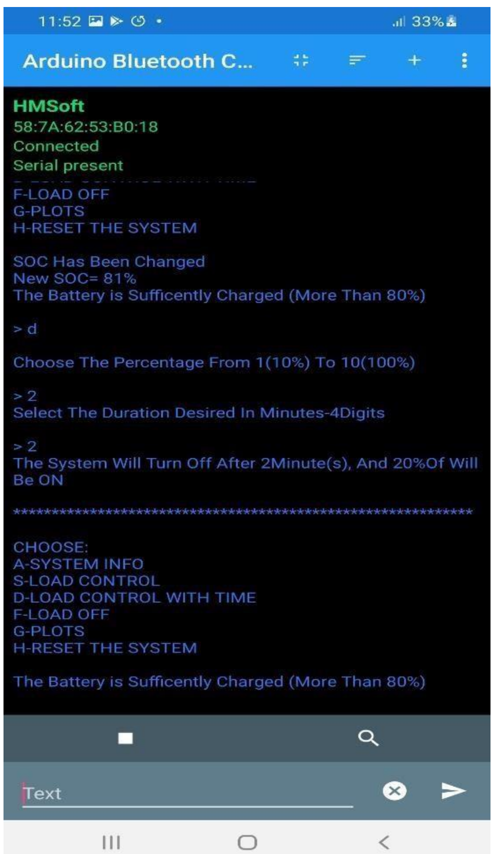


Figure (4-3e) Load Control with Time

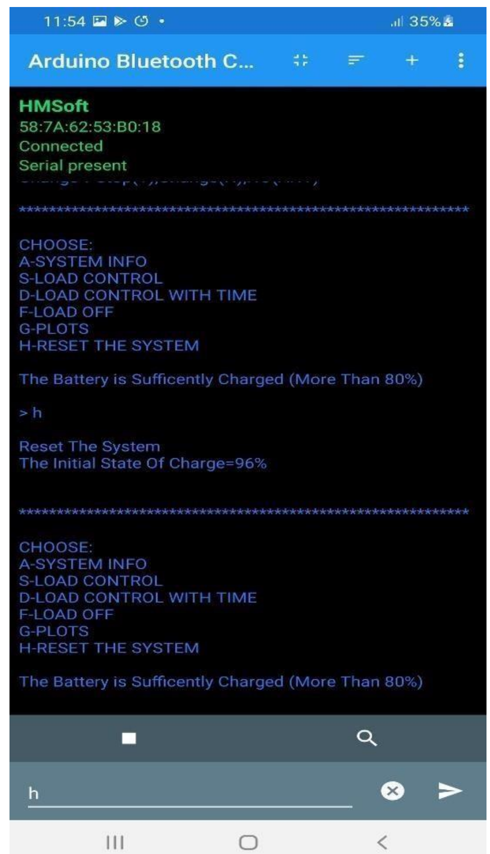


Figure (4-3f) Reset the System

<u>G- Plots:</u> Provides the user with system information in the form of plots. After choosing this option, the user can choose to plot the following:

1. Voltage
2. Current
3. State of Charge
4. Power
5. All values mentioned above

Once chosen, the plots are generated on the renewable energy lab's PC for observation as shown in Figures (4-4a,b, and c). The user chose the 5th option which is to plot all values.

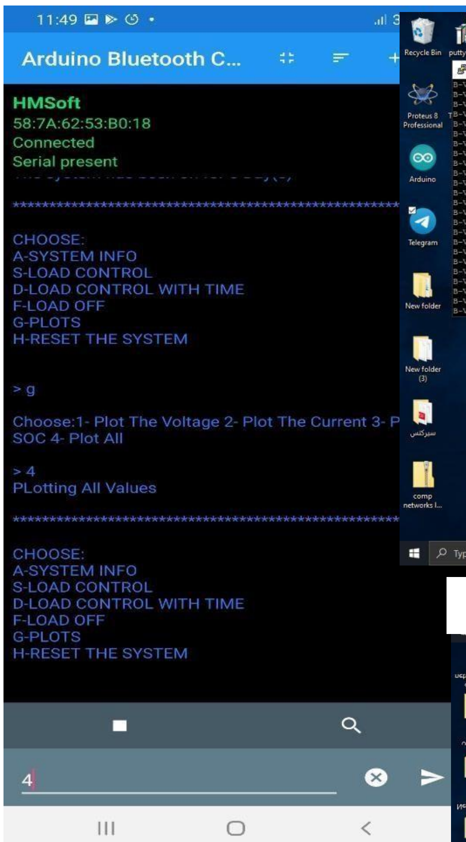


Figure (4-3g) Plots

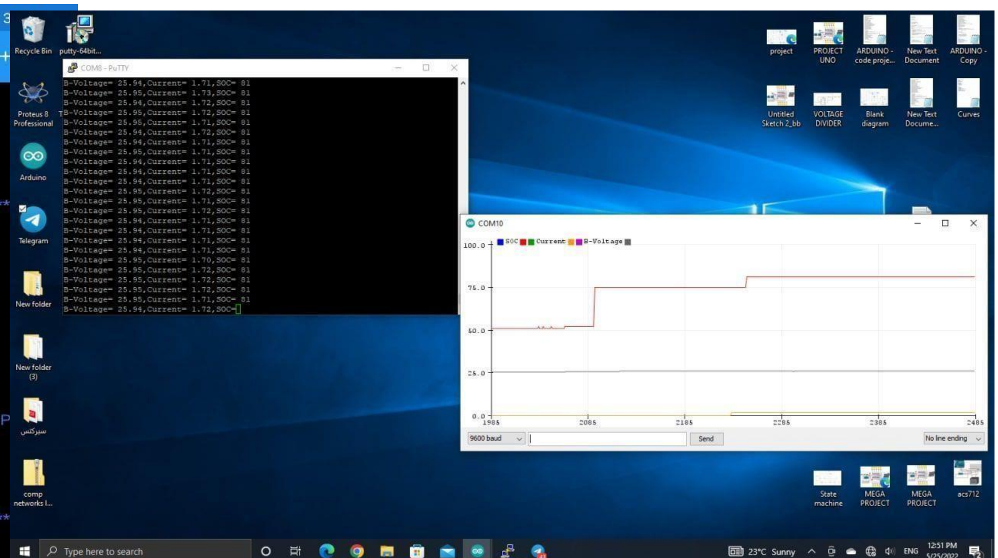


Figure (4-4a) Plots with loads turned on

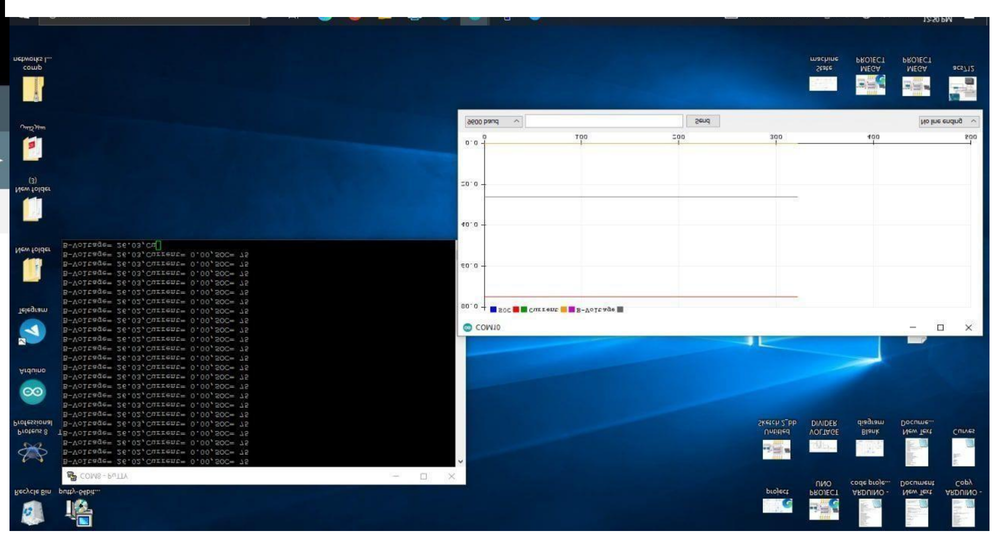


Figure (4-4b) Plots without any load

4.2 Experimental Setup

For the purpose of testing the system, we used a DC power supply to test the system instead of the batteries in order to examine any issues that might surface, the following connection (shown in Figure 4-7) illustrates the experiment setup. Both the PC and smartphone in Figure 4-7 can be used to control the system, both were tested and were functioning properly, we used both of them for the sake of the experiment. The multimeter shown in the figure was used to measure the voltage supplied by the DC source more accurately.

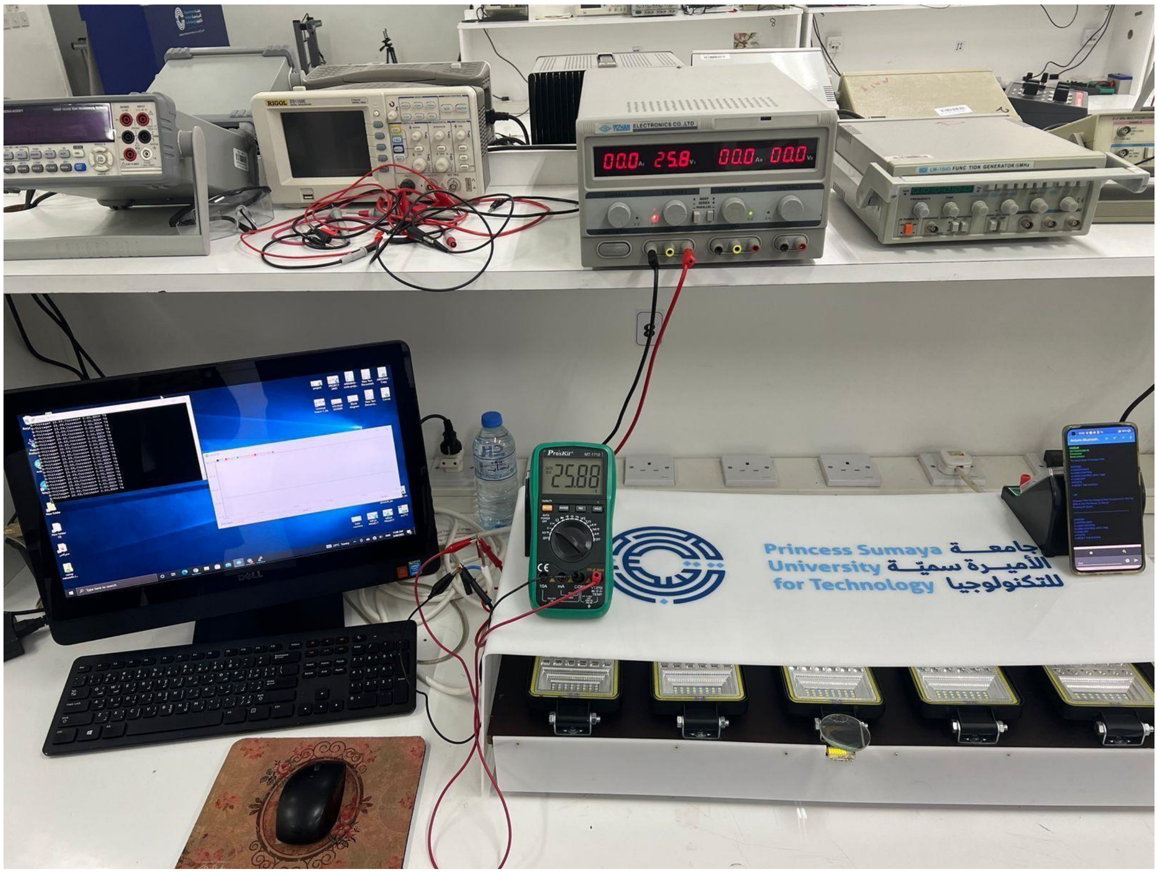


Figure 4-7 Experiment Setup

We chose all percentages of the loads that can turned on for this experiment and observed the values displayed to ensure system integrity. We will discuss on of the options, which is when the load is 50% on, supplied by 25.88 V.

4.3 Experiment Results

After conducting the experiment discussed in section 4.2, we got the following results:

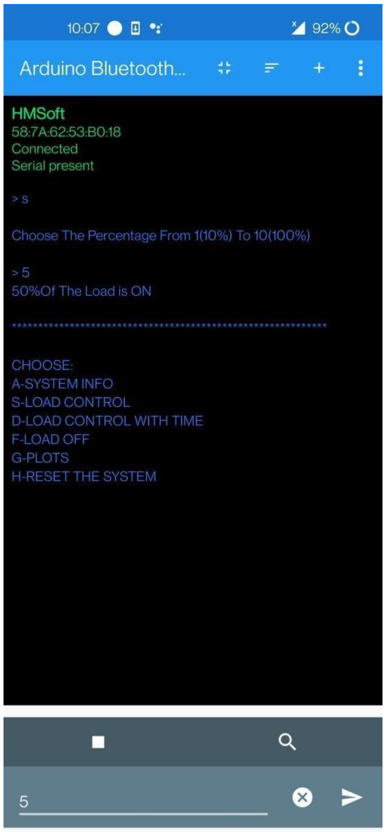


Figure 4-8 System Interface Showing 50% of the Load On.

Figure (4-8a) 5 LED Floodlights ON out of 10

Figure (4-8b) 5 LED Flood Lights ON out of 10

The user chose the load control option on the smartphone interface with 50% of the load on from Figure (4-7), that can be validated from Figures (4-8a) and (4-8b) where the system shows 50% of the load ON.

The 10-Segment has five bars illuminated, meaning that the battery is at around 50%.

4.4 Validation of Design Requirements Within the Realistic Constraints

	Parameter	Required	Achieved
Requirements	Connection to Off-grid PV System	Safe connection to off-grid system	✓
	System Size	Should not exceed 500 W	Approx. 235 W
	Functionalities	Display load status (Connectivity, Current, Power)	✓
		Ability to turn-on and off the load at specific time.	✓
		Control the load in 10% step	✓
	Protection	Against load short circuit	✓
	Input voltage level	24 V and 48 V	24 V
	Disconnection	If battery bank is discharged	✓
	Simulation	PSIM tools	✓
	Placement	Should fit in renewable energy lab	✓
Economic constraints	Price	Should not exceed 500 Jds	348.5 Jds

5 Conclusion

This project aimed to develop a system that is capable of controlling a DC load, connected to the off-grid PV system at Princess Sumaya University's Renewable Energy lab. Based on the quantitative and qualitative analysis obtained from the designed system, it can be concluded that the system is able to monitor the state of charge of the batteries, control the load in 10% step, display all system information, and disconnect the load when the battery is discharged to 20%, meeting all requirements set. Throughout our research, we discussed the importance of renewable energy, solar energy specifically, especially in Jordan and the means of generating electricity through off-grid PV systems, where generation isn't restricted by the grid. The research and development of the control system made it clear that the development of such control systems that provide battery protection, system information, and control over the loads is important for implementation of off-grid PV systems to safely generate electricity to households and buildings in rural areas. Therefore, by analyzing the behavior of the off-grid system and its batteries, this thesis has shown the importance of controllable loads. This project clearly illustrates how the initial design differs from the developed one, pointing out the importance of trial and error from an engineering perspective. Several reasons for the change in design were mentioned in earlier chapters, some were financial, others restricted our design, some were changed as means to protect the system such as controlling the system wirelessly through a computer or smartphone instead of a wired connection using a cable. Based on these conclusions, future studies could consider the implementation of machine learning on the developed system where an algorithm could be applied to it by using the data stored in the designed system. The algorithm can aid in providing the system with more information and control it in more complex ways.

Appendix (C) :

```
//millis() OVER FLOW AFTER 50 DAY.

//Instruction In Matlab take 62.5 ns
#include <Wire.h>//Library of cabelung
#include <INA219.h>//Library of the sensor
INA219 monitor;//Sensor for voltage and current
int inByte; // Stores incoming command
int j=0; //For Relays
int s=0;// For Drawing
int v=0; //For Battarey Quatar

int z=0;// For Battarey Quatar follow
int Delay=5000; // delay time
long int n=0; // Time work enterd
long int m=0;//Time compare
long int rt=0; //Remmaing time
long int day=0;//For millis over flow

long int sr=0;//Time for Calculate The SOC
unsigned long DelayTimeOut;//Timer For Input
double sensorVoltage = 0;//Reading
double sensorCurrent = 0;//Reading

double WH=0; //The Energy In The Batterey Per Hour
double WHLoad=0; //Energy When Using Load
double AvaliableCap=0;//Energey in the Batterey
```

```
double NewCap=0;//Energey diference
int SOC=0;//State of charge
int SOC1=0;//State of charge

int dflag=0;//For millis over flow

bool flag2=0;//Flag for timer with user
bool flagl=0;//Flag for Battarey low
bool flagf=0;//Flag for battarey full
bool flagc=0;//Flag for drawing the curves
bool flagv=0;//Flag for batterey reading
void setup()
{

  Serial.begin(9600);//serial for drawing and programming
  Serial1.begin(9600);//serial for conect with mobile
  Serial3.begin(9600);//serial for saving data


  monitor.begin();//define ina219 sensor
  day=86400000;//number of mille seconds in one day
  sr=60000;//number of mille seconds in one minute
  Serial1.println("");
  Serial1.println("********************************************");
  Serial1.println("WELCOME TO RENEWABLE LAB CONTROLLING SYSTEM");
  Serial1.println("*******************************************");
  Serial1.println("");
  Serial1.println("Enter The Password");
  password();
```

```
  sensorVoltage=(monitor.busVoltage()+(-j*monitor.shuntVoltage())+ monitor.busVoltage())/2;//read
the voltage

  SOC=(sensorVoltage*56.352)-1379.1;//calculate the SOC
  WH=sensorVoltage*20*28.2;//calculate watt per hour for battery at rest
  AvaliableCap=(SOC*WH)/100;//calculate avilable capacity in the battery
  rt=AvaliableCap/(sensorVoltage*28.2);//calculate the Remaining Time at rest
  if(rt<0){rt=0;}
  if(sensorVoltage>=25.8){z=v=1;}

  else if( sensorVoltage>=25.4 && sensorVoltage<25.8){z=v=2;}
  else if(sensorVoltage>=25 && sensorVoltage<25.4){z=v=3;}
  else if( sensorVoltage>=24.4 && sensorVoltage< 25){z=v=4;}
  Serial1.print("The Initial State Of Charge=");
  if(SOC>100){SOC=100;}
  if(SOC<0){SOC=0;}

  Serial1.print(SOC);

  Serial1.println("%");

  Serial1.println("");

  for (int i=2;i<12;i++){//put the pins in output mode and in high so the relays will be off
                pinMode(i, OUTPUT);
                digitalWrite(i, HIGH);



               }

}


void loop()

{
```

```
inByte=0;
Serial1.println("CHOOSE:");
Serial1.println("A-SYSTEM INFO");
Serial1.println("S-LOAD CONTROL");
Serial1.println("D-LOAD CONTROL WITH TIME");

Serial1.println("F-LOAD OFF ");
Serial1.println("G-Show Values On Curves");
Serial1.println("H-Rest The System");
Serial1.println("");
do{

   sensorVoltage=(monitor.busVoltage()+(-j*monitor.shuntVoltage())+ monitor.busVoltage())/2;
   sensorCurrent=monitor.shuntCurrent()*j * -1;
   if(sensorVoltage>=25.8){z=1;}

   else if( sensorVoltage>=25.4 && sensorVoltage<25.8){z=2;}
   else if(sensorVoltage>=25 && sensorVoltage<25.4){z=3;}
   else if( sensorVoltage>=24.4 && sensorVoltage< 25){z=4;}
   if(z==v){
        if(sr<=millis()){ //timer for calculate SOC, WH, Capacity and Remaining Time every 1 minute
            sr=sr+60000;
            WHLoad=(sensorVoltage*sensorCurrent)*0.01667;
            NewCap=AvaliableCap-WHLoad;


            if(NewCap!=AvaliableCap){



                        abs(NewCap);
```

```
                SOC=(sensorVoltage*56.352)-1379.1;
                AvaliableCap=(SOC*WH)/100;
                NewCap=AvaliableCap-WHLoad;
                AvaliableCap=NewCap;
                WH=sensorVoltage*20*28.2;
                SOC=(NewCap/WH)*100;
                if(SOC>100){SOC=100;}
                if(SOC<0){SOC=0;}


                rt=AvaliableCap/(sensorVoltage*sensorCurrent);
                Serial1.println("SOC Has Been Changed");
                Serial1.print("New SOC= ");
                Serial1.print(SOC);
                Serial1.println("%");


               }

      }

   }



else {

   delay(500);
    int k=v;
    SOC=(sensorVoltage*56.352)-1379.1;//calculate the SOC
    WH=sensorVoltage*20*28.2;//calculate watt per hour for battery at rest
    AvaliableCap=(SOC*WH)/100;//calculate avilable capacity in the battery
```

```
rt=AvaliableCap/(sensorVoltage*28.2);//calculate the Remaining Time at rest
if(rt<0){rt=0;}
if(sensorVoltage>=25.8){z=v=1;}

else if( sensorVoltage>=25.4 && sensorVoltage<25.8){z=v=2;}
else if(sensorVoltage>=25 && sensorVoltage<25.4){z=v=3;}
else if( sensorVoltage>=24.4 && sensorVoltage< 25){z=v=4;}
Serial1.print("The Quartar Change from ");
Serial1.print(k);
Serial1.print(" To ");
Serial1.println(z);
if(SOC>100){SOC=100;}
if(SOC<0){SOC=0;}


}



{

Serial3.print("B-Voltage= ");//send the values for save
Serial3.print(sensorVoltage);
Serial3.print(",");
Serial3.print("Current= ");
Serial3.print(sensorCurrent);
Serial3.print(",");
Serial3.print("SOC= ");
Serial3.println(SOC);
}
```

```
if(SOC<=20){//check SOC

            flagf=0;
            if(flagl==0){
                flagl=1;

                Serial1.println("The Battery Less Than 20%");
                delay(1000);
                for (int i=2;i<12;i++){



                    digitalWrite(i, HIGH);



                    }

                j=0;

                }

}



  else if(SOC>=80){//check SOC
            flagl=0;
            if(flagf==0){
                flagf=1;

                Serial1.println("The Battery More Than 80%");
                delay(1000);
              }

  }
```

```
if(sensorVoltage<21.2){ //check battery if it is low
        if(flagv==0){
            flagv=1;

            Serial1.println("The Battery Died");
            delay(1000);
            for (int i=2;i<12;i++){


                digitalWrite(i, HIGH);



            }

            j=0;



        }

    }



if(s!=0){//check if there is any order for drawing



    if(s==1){

        Serial.print("B-Voltage");
        Serial.print(",");
        Serial.println(sensorVoltage);
    }

    else if (s==2){

        Serial.print("Current");
```

```
              Serial.print(",");
              Serial.println(sensorCurrent);
             }

  else if (s==3){

              Serial.print("SOC");
              Serial.print(",");
              Serial.println(SOC);
             }

  else if (s==4){

              Serial.print("SOC");
              Serial.print(",");
              Serial.print(SOC);
              Serial.print(",");
              Serial.print("Current");
              Serial.print(",");
              Serial.print(sensorCurrent);
              Serial.print(",");
              Serial.print("B-Voltage");
              Serial.print(",");
              Serial.println(sensorVoltage);



             }

}
```

```
if(flag2==1){//check if the time that user enterd end or not
        if(millis()>= m){
                flag2=0;

                for (int i=2;i<12;i++){



                        digitalWrite(i, HIGH);



                        }

                Serial1.println("Time End");



                }

        }

if (day<=millis()){//check for days

        dflag++;
        day=day+86400000;
        if(dflag>49){
                n=0;
                m=0;
                sr=60000;
                day=86400000;

                for (int i=2;i<12;i++){

                        digitalWrite(i, HIGH);

                        }

                Serial1.println("Time Over Flow");
```

```
                    }

            }

    }while(!(Serial1.available()>0));//waitting any input
      flagf=0;
      flagl=0;
      flagv=0;
      Serial1.println("");
      inByte = Serial1.read();
    if (inByte == 'A'|| inByte =='a'){ //SYSTEM INFO
                    Serial1.print("SOC=");
                    if(SOC>100){SOC=100;}
                    if(SOC<0){SOC=0;}
                    Serial1.print(SOC);
                    Serial1.print("% ,Quartar");
                    Serial1.println(z);


                    Serial1.print("B-Voltage=");
                    Serial1.print(sensorVoltage, 4);
                    Serial1.println(" V");


                    Serial1.print("Current=");
                    sensorCurrent=sensorCurrent*1000;
                    Serial1.print(sensorCurrent, 4);
                    Serial1.println(" mA");
```

```
  Serial1.print("Power=");
  Serial1.print(sensorCurrent*sensorVoltage, 4);
  Serial1.println(" mW");


  Serial1.print("Load=");
  Serial1.print(j*10);
  Serial1.println("%");


  if(s==1){Serial1.println("The Plotter Drawing The Voltage");}
  if(s==2){Serial1.println("The Plotter Drawing The Current");}
  if(s==3){Serial1.println("The Plotter Drawing The SOC");}
  if(s==4){Serial1.println("The Plotter Drawing ALL");}


  Serial1.print("Loading-Time=");
  Serial1.print(millis() / 60000);
  Serial1.println(" Min");


  Serial1.print("Remaining-Time=");
  Serial1.print(rt*60);
  Serial1.println(" Min");


  Serial1.print("WorkDays=");
  Serial1.print( dflag);
  Serial1.println(" Day");
}
```

```
else if (inByte == 'F'||inByte == 'f'){ //LOAD OFF

        Serial1.println("System Is Closed");

        for (int i=2;i<12;i++){//put all relay on off mode


                digitalWrite(i, HIGH);


              }

        j=0;

       }


else if (inByte == 'S'||inByte == 's'){ //LOAD CONTROL

        Serial1.println("Choose The Percentage From 1(10%) To 10(100%)");
        DelayTimeOut = millis() + Delay;
        while(!(Serial1.available() > 0)&& (millis() < DelayTimeOut));
        inByte=Serial1.parseInt();
        if( inByte>=1 && inByte<=10){

                j= inByte;

                Serial1.print("The Load Open ");
                Serial1.print((inByte)*10);
                Serial1.println("%");


                for (int i=2;i<inByte+2;i++){//turn on the relays according to user input
                        digitalWrite(i, LOW);
```

```
}

for (int i=inByte+2;i<12;i++){//turn off the other relys

        digitalWrite(i, HIGH);

    }

}

else{

  Serial1.println("Wrong Input Percentage");

 }

}

else if (inByte == 'D'||inByte == 'd'){ //LOAD CONTROL WITH TIME
        Serial1.println("Choose The Percentage From 1(10%) To 10(100%)");
        DelayTimeOut = millis() + Delay;
        while(!(Serial1.available() > 0)&& (millis() < DelayTimeOut));
        inByte=Serial1.parseInt();
        if(inByte>=1 && inByte<=10){

              Serial1.println("Select The Duration You Want In Minutes-4Digits");
```

```
 DelayTimeOut = millis() + Delay;

 while(!(Serial1.available() > 0)&& (millis() < DelayTimeOut));
n=Serial1.parseInt();
if(n<=9999){

      if(n<rt){

                j= inByte;
                m=n*60000+ millis();
                flag2=1;
                Serial1.print("The Load Open ");
                Serial1.print((inByte)*10);
                Serial1.println("%");


                for (int i=2;i<inByte+2;i++){

                                          digitalWrite(i, LOW);



                }



                for (int i=inByte+2;i<12;i++){



                                                           digitalWrite(i, HIGH);



                                          }



                Serial1.print("The System Will Work For ");
                Serial1.print(n);
```

```
                                        Serial1.print("Min,And The Load Will Be ");
                                        Serial1.print((inByte)*10);
                                        Serial1.println("%");
                                      }

                                     else{

                                        Serial1.println("The Energy Is Not Enough With Time You
Entered");

                                        }

                                    }



                              else{

                                Serial1.println("Wrong Input Time");

                                }

                              }

                  else{

                    Serial1.println("Wrong Input Percentage");

                    }

}

 else if (inByte == 'G'|| inByte =='g'){ //Show Values On Curves
  if (flagc==0){
        Serial1.println("Choose:1-Drawing Voltage 2-Drawing Current 3-Drawing SOC 4-All");
        DelayTimeOut = millis() + Delay;
        while(!(Serial1.available() > 0)&& (millis() < DelayTimeOut));
        inByte=Serial1.parseInt();
        if (inByte == 1){

              s=1;
```

```
                        Serial1.println("Start Voltage Curve Drawing");
                        flagc=1;
                        }




                else if (inByte == 2){

                        s=2;

                        Serial1.println("Start Current Curve Drawing");
                        flagc=1;
                        }



                else if (inByte == 3){

                        s=3;

                        Serial1.println("Start SOC Curve Drawing");
                        flagc=1;
                        }

                else if (inByte == 4){

                        s=4;

                        Serial1.println("Start Curve Drawing for all");
                        flagc=1;
                        }



                }

     else if (flagc=1){

                Serial1.println("The Curve Is Being Drawn Do You Want To Stop Or Change ?
Stop(Y),Change(N),NO(ANY)");
```

```
DelayTimeOut = millis() + Delay;

while(!(Serial1.available() > 0)&& (millis() < DelayTimeOut));
inByte = Serial1.read();
if (inByte == 'Y'||inByte == 'y'){

                    s=0;

                    Serial1.println("Curve Drawing Stopped");
                    flagc=0;
                    }



if (inByte == 'N'||inByte == 'n'){

                    Serial1.println("Choose:1-Drawing Voltage 2-Drawing Current 3-Drawing
SOC 4-All");

DelayTimeOut = millis() + Delay;

while(!(Serial1.available() > 0)&& (millis() < DelayTimeOut));
inByte=Serial1.parseInt();
if (inByte == 1){

          s=1;

          Serial1.println("Start Voltage Curve Drawing");
          flagc=1;
          }

else if (inByte == 2){

          s=2;

          Serial1.println("Start Current Curve Drawing");
          flagc=1;
          }
```

```
else if (inByte == 3){

        s=3;

        Serial1.println("Start SOC Curve Drawing");
        flagc=1;
        }

else if (inByte == 4){

        s=4;

        Serial1.println("Start Curve Drawing for all");
        flagc=1;
        }

                }

  }

    }

else if (inByte == 'H'||inByte == 'h'){ //Rest The System
        Serial1.println("Rest The System");
        for (int i=2;i<12;i++){


                digitalWrite(i, HIGH);


                }

        inByte; // Stores incoming command
        j=0;
        n=0;
        m=0;
        rt=0;
```

```
s=0;

sensorVoltage = 0;

sensorCurrent = 0;
WH=0;
WHLoad=0;
AvaliableCap=0;
NewCap=0;
SOC=0;
dflag=0;
flag2=0;
flagl=0;
flagf=0;
flagc=0;
flagv=0;
delay(500);
sensorVoltage=(monitor.busVoltage()+(-j*monitor.shuntVoltage())+
monitor.busVoltage())/2;

SOC=(sensorVoltage*56.352)-1379.1;
WH=sensorVoltage*20*28.2;
AvaliableCap=(SOC*WH)/100;
rt=AvaliableCap/(sensorVoltage*28.2);
if(rt<0){rt=0;}
Serial1.print("The Initial State Of Charge=");
if(SOC>100){SOC=100;}
if(SOC<0){SOC=0;}

Serial1.print(SOC);
```

```
                        Serial1.println("%");
                        Serial1.println("");
                                        }

 Serial1.println("");
 delay(1000);
 Serial1.println("");
}

void password(){

             String pass;
             bool loop =1;
             while(loop){
             while(!(Serial1.available()>0));
             pass=Serial1.readString();
             if(pass=="psut"){
                            loop=0;
                            Serial1.println("Welcome");
                            Serial1.println("");
                           }

 else{

                 Serial1.println("Wrong Password");

                }

                      }

}
```

References

[1] "Panel generation factor - Wikipedia." https://en.wikipedia.org/wiki/Panel_generation_factor.

[2] "Sizing an off-grid photovoltaic system | Semantic Scholar." https://www.semanticscholar.org/paper/Sizing-an-off-grid-photovoltaic-system-Moumouni-Gdeisat/9ff174e87870601b30b3508dd521af22ea68d070.

[3] "(PDF) Solar radiation map of Jordan governorates." https://www.researchgate.net/publication/326832115_Solar_radiation_map_of_Jordan_governorates.

[4] "3: Solar Insolation Map of the World | Download Scientific ... " https://researchgate.net/figure/Solar-Insolation-Map-of-the-World_fig6_322134423.

[5] "EdExcel / OCR GCSEs and AS/A Levels – School teaching and ...." http://bedford-computing.co.uk/learning/wp-content/uploads/2016/04/Designing-Embedded-Systems-with-PIC-Microcontrollers_V2.pdf.

[6] Renewable and Efficient Electric Power System by Gilbert M. Masters 2nd Edition.
[7]"Microcontroller - Wikipedia." https://en.wikipedia.org/wiki/Microcontroller.
[8] "DESIGN AND IMPLEMENTATION OF A SOLAR CHARGE CONTROLLER..... " https://www.researchgate.net/publication/303683238_DESIGN_AND_IMPLEMENTATION_OF_A_SOLAR_CHARGE_CONTROLLER_WITH_VARIABLE_OUTPUT.

[9] "'HVLJQ DQG &RQVWUXFWLRQ RI D +LJK 3RZHU 3URIHVVLRQDO 6RODU .... " https://www.duet.ac.bd/wp-content/uploads/2019/09/Thesis_062201.pdf.

[10] "The Complete Guide to Lithium vs Lead Acid ... - Power Sonic." https://www.power-sonic.com/blog/lithium-vs-lead-acid-batteries/.

[11] "Moulded Case Circuit Breakers Explained - GSES." 11 Oct. 2017, https://www.gses.com.au/molded-case-circuit-breakers/.

[12] "What is a Relay? Definition, Working Principle and .... " https://circuitglobe.com/relay.html.

[13] Matlab. n.d. *MPPT Algorithm*. [online] Available at: <https://www.mathworks.com/solutions/power-electronics-control/mppt-algorithm.html#:~:text=Maximum%20power%20point%20tracking%20(MPPT,changing%20solar%20irradiance%2C%20temperature%2C%20and> [Accessed 20 December 2021].

https://www.electronicshub.org/what-is-relay-and-how-it-works/#:~:text=Relay%20works%20on%20the%20principle,a%20magnetic%20field%20around%20it.&text=In%20the%20relay%20Copper%20coil,attracting%20the%20contact%20as%20shown.
Relay information

List of Duties:

	Software	hardware	Documentation
Hassan Halawani	•	•	
Hussam baidas		•	•
Saif sayegh		•	•
Dana Baddawi	•		•