

UNIVERSAL SOLAR PUMP CONTROLLER

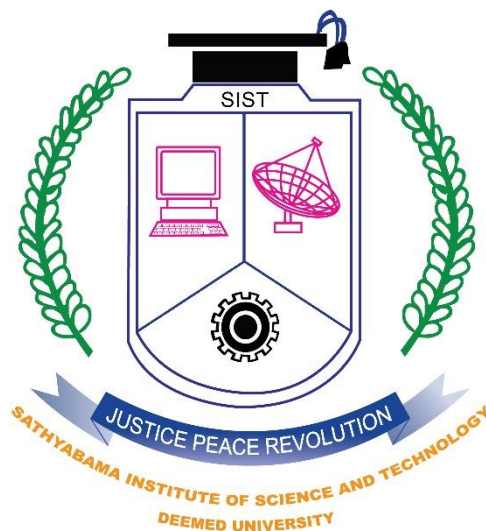
Submitted in partial fulfillment of the requirements for the award of

Bachelor of Engineering Degree in
Electrical and Electronics Engineering

by

SURYA V (Register No. 39140060)

KESHAV SRIDHAR (Register No. 39140024)



**DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **SURYA V** (Reg. No. 39140060) and **KESHAV SRIDHAR** (Reg. No. 39140024) who carried out the project entitled "**UNIVERSAL SOLAR PUMP CONTROLLER**" under our supervision from June 2022 to April 2023.

Pu
18/4/23
Dr.M.Pushpavalli
Internal Guide

Sivachidambar
18/4/23
Dr.V.SIVACHIDAMBARANATHAN
Head of the Department

Submitted for Viva voce Examination held on 26/04/2023

Name: *Dr. S. Ramesh*

Signature: *[Signature]*

Dr. V. Balaraman

[Signature]

Internal Examiner

External Examiner

DECLARATION

We **SURYA V** (Reg. No. 39140060) and **KESHAV SRIDHAR** (Reg. No. 39140024) hereby declare that the Project Report entitled "**UNIVERSAL SOLAR PUMP CONTROLLER**" done by us under the guidance of **Dr.M.Pushpavalli** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electrical and Electronics Engineering.

DATE: 18-04-2023

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATES

1. V. Surya

2. S. Keshav

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. Nandhita., Dean, School of Electrical and Electronics Engineering** and **Dr. V. Sivachidambaranathan, Head of the Department, Dept. of Electrical and Electronics Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr. M.Pushpavalli** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Electrical and Electronics Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

The farming system is rapidly evolving. Farming equipment, agricultural structures, and production facilities are all constantly being upgraded. There are many farming applications for panel solutions. The Solar PV pumping system, which was installed at great expense to the farmer and the government's exchequer, is currently only used for 150 days or about half of the year. The controller provided for installation of the solar pumping system should be able to carry out a number of other tasks for agricultural and other needs of a farmer in order to make the most of the solar photovoltaic system that generates the electricity throughout the year during sunny hours. The productivity of the agricultural industry and farmer income will both rise as a result. The purpose of this topic is to describe the various operating modes used by solar water pumping systems throughout the year.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGENO
	ABSTRACT	V
	LIST OF FIGURES	Viii
1	INTRODUCTION	
	1.1 GENERAL INTRODUCTION	1
2	LITERATURE SURVEY	
	2.1 LITERATURE INCEPTION OF SOLAR PUMP CONTROLLER	4
	2.2 MAJOR FINDINGS FROM LITERATURE SURVEY	4
	2.3 LITERATURE OVERVIEW ON SOLAR PUMP CONTROLLER	7
	2.4 PROBLEM IDENTIFICATION	8
	2.5 SCOPE OF THE PROJECT	9
	2.6 OBJECTIVE	9
	2.7 METHODOLOGY	9
	2.8 APPLICATION	10
	2.9 STANDARDS	10

3 PROJECT DESCRIPTION

3.1	EXISTING BLOCK DIAGRAM	12
3.2	DISADVANTAGE OF EXISTING SYSTEM	13
3.3	PROPOSED BLOCK DIAGRAM	13
3.4	ADVANTAGE OF PROPOSED SYSTEM	14
3.5	CONSTRAINTS	15
3.6	TRADEOFFS	16

4 SIMULATION ANALYSIS

4.1	SYSTEM DEVELOPMENT & TOOLS	18
4.2	SIMULATION RESULTS	18
4.2.1	Simulation results for ultrasonic sensor	19
4.2.2	Simulation results for soil moisture sensor	20

5 RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

5.1	COMPONENTS USED	22
5.1.1	Arduino uno	22
5.1.2	Node mcu	22
5.1.3	Soil moisture sensor	23
5.1.4	Ultrasonic sensor	24

	5.1.5	Servo motor	24
	5.1.6	DC Pump motor	25
	5.1.7	Relay module	26
	5.1.8	LCD display	26
	5.1.9	Solar panel	27
	5.2	ARDUINO UNO INTERFACING NODE MCU	28
	5.3	UBIDOTS	29
	5.3.1	Connect a node MCU ESP8266 to UBIDOTS	29
	5.4	PROPOSED SYSTEM FLOWCHART	31
	5.4.1	Arduino UNO PROCESS	31
	5.4.2	UBIDOTS PROCESS	33
	5.5	UBIDOTS RESULT	34
	5.6	HARDWARE MODEL	38
6	6.1	CONCLUSION	41
	6.2	FUTURE SCOPE	41
		REFERENCES	43
		APPENDIX SOURCE CODE	46-55

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
1.1	Solar Pump installation in India	2
1.2	State wise installed solar pumps	3
1.3	Existing Block Diagram	12
3.2	Proposed Block Diagram	13
4.1	Flowchart for simulation	18
4.2	Low Water Level Motor Turned ON	19
4.3	Water Tank Full Motor Turned OFF	20
4.4	Soil Dry Motor Turned ON	20
5.1	Arduino UNO	22
5.2	ESP8266 Node MCU Pinout	23
5.3	Soil Moisture Sensor	24
5.4	Ultrasonic Sensor	24
5.5	Servo Motor	25
5.6	5-12V DC Motor	25
5.7	Relay Module	26
5.8	16X2 LCD Display	27
5.9	Solar Panel	27
5.10	Arduino UNO Interfacing Node MCU	28
5.11	UBIDOTS Setup Code	30

5.12	Flow Chart for Arduino UNO	31
5.13	UBIDOTS Process Flowchart	33
5.14	Pump Switches	34
5.15	Data Page	35
5.16	Pump 1 Data	35
5.17	Pump 2 Data	36
5.18	Soil Moisture Sensor Data	36
5.19	Ultrasonic Sensor Data	37
5.20	Hardware Circuit Layout Diagram	38
5.21	Prototype of Universal Solar Pump Controller	39

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Energy is a fundamental requirement for a nation's expanding economic development. Today, many electrical operations are required to stop when the power supply is cut off. It is practically impossible to determine the precise amount of power that was supplied and how much of it was used to construct modern development. Due to increased agricultural and industrial production and the consumption of human labour, there is a large amount of power energy available. The higher a nation's per capita energy consumption, the higher its population's standard of living. There are many different forms of energy in nature and behaviour, but power energy is the most significant one.

Today's society is more reliant and reliable on the use of power energy, which has almost completely taken over our daily lives. The most plentiful source of energy in the entire world is solar energy. In addition to providing a solution to the current energy crisis, solar energy is also an environmentally friendly form of energy. Using solar energy effectively involves photovoltaic generation. The use of PV modules to power water heaters, street lights, and domestic loads has become very popular. Solar panel prices have been steadily falling, which promotes and supports their use across a variety of industries. Farmers in India's current energy crisis may find a suitable alternative in solar irrigation systems. This method of energy production is environmentally friendly and initially offers free energy.

Solar Irrigation Pump Installations in India

Slow progress on government's 2019 PM-KUSUM scheme

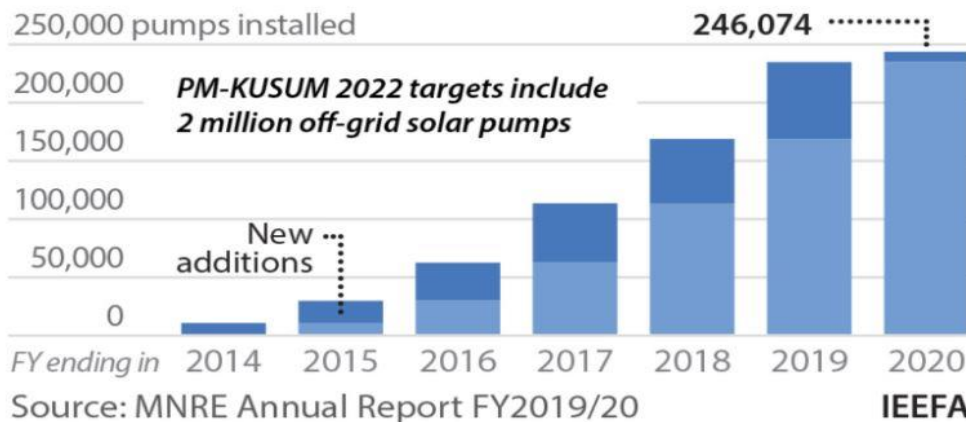


Fig 1.1 Solar Pump installations in India

The figure 1.1 displays year-by-year solar pump installation data for India. Solar systems, also known as solar systems, turn sunlight directly into electricity. Compared to PV thermal collectors used in solar water heaters, this is different. By generating electricity from the sun rather than burning fossil fuels, a solar photovoltaic system can help lower carbon emissions and your electricity bill. To help you harness the power of the sun for all kinds of commercial, industrial, and residential electricity applications, Wiktionary offers a variety of solar products. The company that generates and/or distributes electricity to consumers handles the majority of the power distribution. The power from various sources is distributed along the grid, which can extend up to 100 miles from the power plants to residences and commercial buildings. Due to overload, extreme weather, and maintenance or updates, this power system is not always dependable. Installing a power system enables you to generate your own power to run your entire home or business and may resolve any issues. with an abundance of utilities. The size and location of the power system, the type of module, the amount of sunlight available,

and the effectiveness of the electrical components used to convert solar energy into power usable by your home all affect how much power is generated.

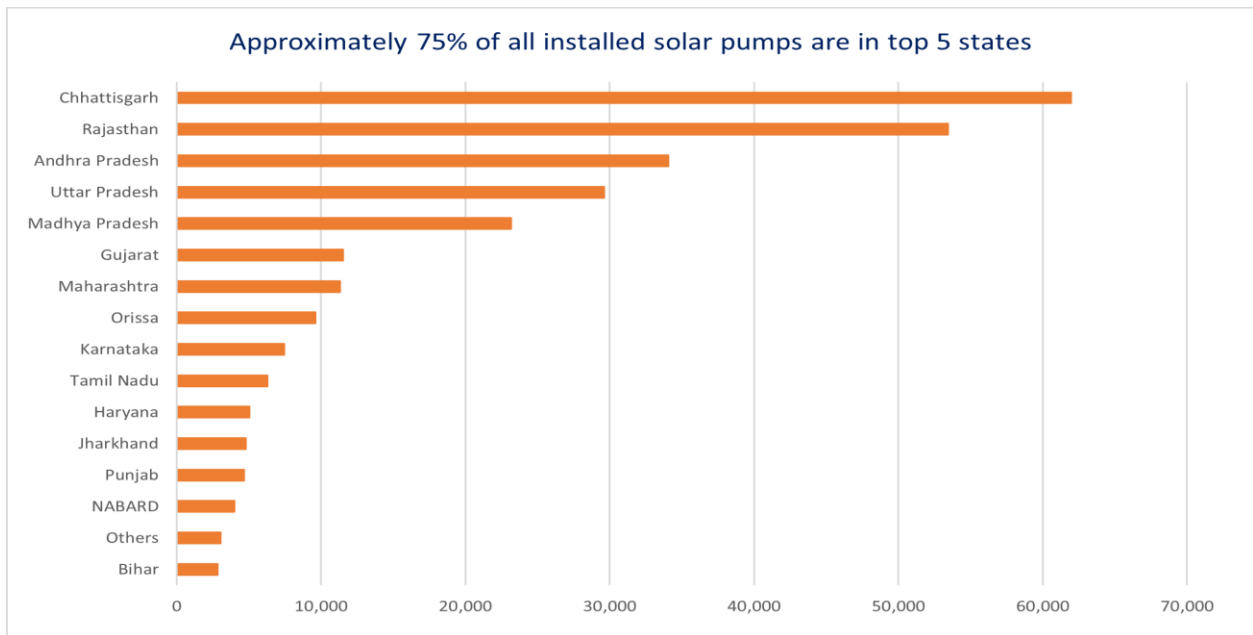


Fig 1.2 State Wise Installed Solar Pumps

Fig 1.2 displays state wise installed solar pumps in india. The primary goal of developing this model is to contribute to "AatmaNirbhar Bharat," which aims to provide farmers with numerous benefits and empower them to independently complete a variety of tasks using a single device, or "USPC." The agricultural sector must be taken into consideration and developed in many ways because it provides the majority of our nation's income. Without a doubt, the farmers are the backbone of our Indian economy, and it is our moral and social duty to help them. They are a great asset to all of us. This project will definitely be a game changer for our farmers & very soon it will revolutionize the agriculture sector. Also, by using this system we can use both pumps and rooftop system. This will be the money saver as well as economy booster system. In India 90% of farmer are still have their homes in their farm and with this system they can continue their farming as well as household appliances at the same timing.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE INCEPTION OF SOLAR PUMP CONTROLLER

The literature on the inception of solar pump controllers highlights the potential benefits of combining solar energy and IoT technology for efficient and sustainable water pumping systems in agriculture. IoT technology can enable remote monitoring and control of irrigation systems, while solar energy can provide a cost-effective and environmentally friendly power source.

Several studies have proposed IoT-based solar pump controllers for irrigation systems, such as the work by Jilani et al. (2020) that proposed a cloud-based solar pump controller for precision agriculture. Other studies have investigated the use of machine learning algorithms to improve the efficiency of solar pump controllers, such as the work by Bansal et al. (2021) that proposed a reinforcement learning-based controller for a solar-powered water pumping system.

Overall, the literature on the inception of solar pump controllers in IoT emphasizes the potential benefits of combining solar energy and IoT technology for sustainable and efficient water pumping systems in agriculture. The integration of sensors and machine learning algorithms can further improve the efficiency and effectiveness of these systems.

2.2 MAJOR FINDINGS FROM LITERATURE SURVEY

RinaeSinugo et al. (2021) proposed a “Solar-Powered Smart Irrigation System”. Irrigation systems has been using different methods of operation such as using manual operating system and automatic operating system. In manual irrigation the person operating the system need to next to the system, so it can be manually operated, and automatic irrigation operates using a mobile application and it can be operated away from the system using either Bluetooth or Wi-Fi module.

Asseel Noori Talib, Kamran Hafeez&et al. (2022) describes a "Design and Implementation of New Battery Monitoring System for Photovoltaic Application". The three stages of the PV monitoring system: data collecting, processing, data display, and storage. Data is collected from a variety of sensors, voltage, current, temperature, humidity, and radiation are all measured and then communicated to the next level via a wired or wireless system.

Mohd. Kashif et al. (2021) proposed "Adaptive Hybrid Generalized Integrator Based SMO for Solar PV Array Fed Encoderless PMSM Driven Water Pump". The encoder influences reliability and cost of permanent magnet synchronous motor (PMSM) operated solar water pump (WP). It is even sensitive to electromagnetic noise and temperature, which thereby reduces its accuracy.

Ujjwal Kumar Kalla et al. (2021) proposed a "Design and Analysis of Solar PV Fed IMD". This paper presents a water-pumping system using solar photovoltaic Arrays. The system consists of PV array, DC-DC boost converter, voltage source inverter, 3- ϕ induction motor drive (IMD) and centrifugal pump. The modified space vector pulse width modulation (MSVPWM) method for control of inrush current and drive's commutation uses lower number of sensors.

Jalu A. Prakosa et al. (2021) proposed Development of Simple Method for Quality Testing of PT100 Sensors Due to Temperature Coefficient of Resistance Measurement. The platinum-100 (PT100) is the one type of temperature sensor which is widely used not only in industry but also in Calibration Laboratory, especially in room condition. To assure the measurement accuracy, it is necessary to have a simple alternative method to test the quality of a PT100 resistance thermometer.

A.R. Al-Ali et al. (2020) proposed a "IoT-Solar Energy Powered Smart Farm Irrigation System". As the Internet of things (IoT) technology is evolving, distributed solar energy resources can be operated, monitored, and controlled remotely. The design of an IoT based solar energy system for smart irrigation is essential for regions around the world, which face water scarcity and power shortage.

AK. Singh et al. (2020) proposed a Multifunctional Solar PV and Grid Based On-Board Converter for Electric Vehicles. This work deals with the development of a multifunctional power electronic converter (PEC) utilizing dual power sources (grid and solar photovoltaic (PV)) for charging phenomenon of plug-in electric vehicles (PEVs). The developed configuration accomplished all modes of vehicles (charging, propulsion (PP) and regenerative braking (RB)).

C.A.Bolu et al. (2019) proposed a "Solar Powered Microcontroller-based Automated Irrigation System with Moisture Sensors". In traditional irrigation approach where water is irrigated on the land without adequate control measures, crops sometimes undergo increased stress with disparities in the soil moisture which consequently reduce the crop performance and output. In this paper, we come up with a prototype to increase crop yield while considering adequate agricultural water management and labour reduction, by adequate control measures in the irrigation process.

López-Vargas et al. (2019) proposed a "Low-Cost Datalogger Intended for Remote Monitoring of Solar Photovoltaic Standalone Systems Based on Arduino". A low-cost datalogger intended for standalone photovoltaic (PV) systems in developing countries was designed by employing open source software and hardware. The low-cost datalogger is capable of measuring electrical and meteorological parameters meeting the accuracy requirements established by the International Electro-technical Commission (IEC) standard for the PV monitoring systems.

Samara et al. (2019) proposed a "Intelligent Real-Time Photovoltaic Panel Monitoring System Using Artificial Neural Networks". Photovoltaic panels system is becoming a popular choice as an alternative source of energy. To harness reliable energy efficiently, the photovoltaic panels system must remain in its best condition.

G. Esakkimuthu et al. (2022) proposed a "PV Based Drive Control System and MPPT Technique under various Irradiation conditions". Recent day's Non- conventional Energy source plays an important role on the agriculture sector. A variety of MPPT methods and control strategies are available to estimate water pumping system efficiency under various atmospheric conditions.

Anshul Varshney et al. (2021) proposed a "An Intelligent Grid Integrated Solar PV Array Fed RSM Drive-Based Water Pumping System". This article deals with an intelligent grid integrated solar photovoltaic (PV) powered water pumping system driven by three-phase reluctance synchronous motor (RSM) drive. A voltage source inverter is used for feeding the dc power to a three-phase RSM, from a solar PV array.

Rashmi Rai et al. (2020) proposed a "Reactive Power Based MRAS for Speed Estimation of Solar Fed Induction Motor With Improved Feedback Linearization for Water Pumping". This article presents an efficient method for control of a solar water pumping system consisting of an induction motor drive (IMD) with a model reference adaptive system (MRAS) based adaptive mechanism of speed estimation. An improved perturb and observe control technique is implemented to achieve maximum power point tracking.

Saurabh Shukla et al. (2020) proposed a "Improved power quality converter for three-phase grid-interfaced PV array fed reduced current sensor-based induction motor drive for water pumping". This paper presents a photovoltaic (PV)–grid integrated system feeding an induction motor-driven water pumping system. This system is used for feeding the induction motor-driven water pump primarily and feeding the utility if the water pumping is not required.

2.3 LITERATURE OVERVIEW ON SOLAR PUMP CONTROLLER

The literature on solar pump controllers is of particular relevance to this survey, as Al-Ali et al. (2020) proposed an IoT-based smart farm irrigation system that utilizes an inception solar pump controller. The literature emphasizes the benefits of using IoT technology for remote monitoring and control of the irrigation system, which can be accessed from a mobile device. The system is designed to optimize the use of solar energy through the inception solar pump controller and the use of solar panels for power, resulting in reduced costs and increased sustainability.

The literature highlights the importance of water quality monitoring, which is facilitated through the integration of sensors into the irrigation system. Additionally, the system includes an automated scheduling algorithm that determines the optimal time and

duration for irrigation based on weather data and soil moisture levels. This reduces water consumption and improves crop yield, making the system more sustainable and cost-effective. Overall, the literature on solar pump controllers in IoT emphasizes the benefits of using this technology for smart irrigation systems, such as improved efficiency, reduced costs, and increased sustainability

2.4 PROBLEM IDENTIFICATION

Here are some potential problem identification points for the listed surveys:

Adaptive Hybrid Generalized Integrator Based SMO for Solar PV Array Fed Encoderless PMSM Driven Water Pump: One potential problem could be the reliability and accuracy of the encoderless PMSM motor. Additionally, the impact of electromagnetic noise and temperature on the system's accuracy may be a concern.

Design and Analysis of Solar PV Fed IMD: A potential problem could be the accuracy and reliability of the system's control methods, particularly during periods of varying atmospheric conditions. Additionally, the use of lower sensor numbers may impact the overall effectiveness of the control system.

Simple Method for Quality Testing of PT100 Sensors Due to Temperature Coefficient of Resistance Measurement: A potential problem could be the accuracy of the alternative testing method proposed and its ability to accurately assess the quality of PT100 sensors.

Multifunctional Solar PV and Grid Based On-Board Converter for Electric Vehicles: A potential problem could be the effectiveness of the dual power sources for charging electric vehicles, particularly during periods of limited sunlight or fluctuating grid power.

An Intelligent Grid Integrated Solar PV Array Fed RSM Drive-Based Water Pumping System: A potential problem could be the reliability and effectiveness of the RSM drive and the overall control system during periods of limited sunlight or weather-related disruptions.

Reactive Power Based MRAS for Speed Estimation of Solar Fed Induction Motor With Improved Feedback Linearization for Water Pumping: A potential problem could be the

accuracy and reliability of the MRAS-based adaptive mechanism of speed estimation, particularly during periods of limited sunlight or weather-related disruptions.

2.5 SCOPE OF THE PROJECT

The Solar PV pumping system deployed at huge cost to the farmer and the exchequer for the Government is currently utilised only for half of the days in a year (around 150 days per year) on an average. In order to optimally utilize the solar photovoltaic system that generates the electricity throughout the year during sunshine hours, the controller supplied for installation of solar pumping system should be able to perform several other tasks for agricultural and other needs of a farmer. This will increase the productivity of agriculture sector and income of farmer. With the use of USPC

Throughout the year, we will successfully use a number of operational modes simultaneously

2.6 OBJECTIVE

The objectives of the proposed work are as follows

- To increase the productivity of agriculture sector.
- To increase the income of farmer.
- To reduce the electricity used for irrigation.

2.7 METHODOLOGY

The system is comprised of sections namely: manual irrigation, automatic irrigation, hardware testing, of which can be accessed by using the menu options located in the Control box.

2.8 APPLICATION

The USPC with SPV modules and structure can be used for agrarian applications such as

- Water pumping
- Apple grading and polishing system, wheat (grain) flour grinding machine.
- Aata/chakki, cutter/chaff, deep-fridger / cold storage, blower fan for cleaning of grains, heating loads.
- Standard voltage three phase motor/equipment of capacity not more than the capacity of Solar PV pumping system.

2.9 STANDARDS

For this project, the following IEEE standards are relevant:

IEEE 1547: This standard provides guidelines for interconnecting distributed energy resources (DERs) with the electric power system. This includes solar pump systems that may be connected to the grid, and it covers safety, performance, and reliability requirements.

IEEE 802.15.4: This standard specifies the physical and media access control (MAC) layer protocols for low-rate wireless personal area networks (LR-WPANs). This can be used for wireless communication between components of the solar pump system, such as sensors and the controller.

IEEE 802.11: This standard specifies the protocols for wireless local area networks (WLANs), commonly known as Wi-Fi. This can be used to connect the solar pump controller to a network for remote monitoring and control.

IEEE 2030.5: This standard provides a communication protocol for integrating DERs, energy storage systems, and electric vehicles with the power grid. It can be used to ensure that the solar pump system operates in coordination with the electric power system, and to enable demand response and other grid services.

IEEE 1888: This standard provides a common data format and communication protocol for IoT devices and systems. It can be used to ensure that the data exchanged between components of the solar pump system is standardized and interoperable.

CHAPTER 3

PROJECT DESCRIPTION

3.1 EXISTING BLOCK DIAGRAM AND EXPLANATION

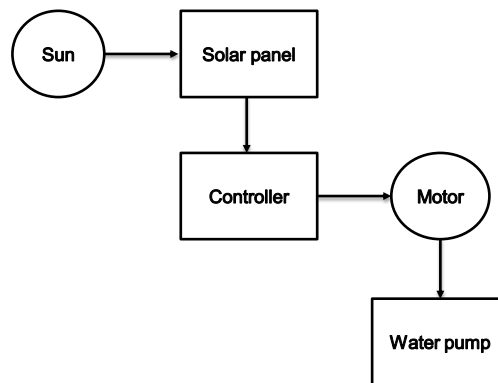


Fig 3.1 Block diagram

The block diagram of the existing system is shown in figure 3.1. The first step in the working procedure of a Existing system is to convert the DC electricity generated by the solar panels into AC (alternating current) power. This is necessary because most pumps are designed to operate on AC electricity. The controller uses an inverter to convert the DC power to AC power, which can then be used to power the pump.

Once the power has been converted to AC, the controller begins to monitor the power output of the solar panels. It does this by measuring the voltage and current produced by the panels and comparing them to the requirements of the pump. If the voltage or current is too high or too low, the controller adjusts the power output to ensure that the pump operates at maximum efficiency.

Finally, the controller monitors the battery that stores excess power generated by the solar panels. If the battery is low on power, the controller adjusts the power output to ensure that the battery is charged. If the battery is fully charged, the controller may shut off the power output to prevent overcharging.

3.2 DISADVANTAGE OF EXISTING SYSTEM

- The main drawback is that there will only be one operating mode.
- The existing system does not support remote access. It is only physically accessible.
- The system does not record or provide information on the starting and stopping times of the motor, or any related data.

3.3 PROPOSED BLOCK DIAGRAM AND EXPLANATION

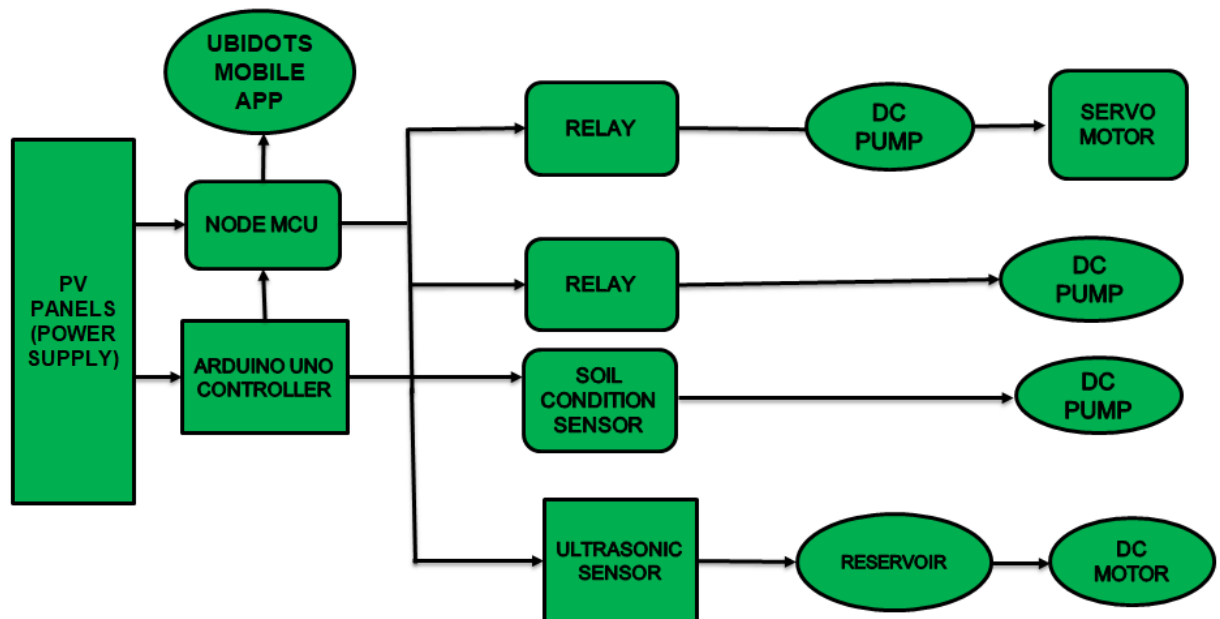


Fig 3.2 Block diagram

The block diagram of the proposed system is shown in figure 3.2. This system converts high voltage, direct current from a solar array into highly controlled, single or three-phase pulse width modulated. Alternative current to run standard DC water pump

motors. Automatic shutdown protects the pump and motor from damage during neglected operation. In the morning, the USPC begins supervising power from the solar array. When power is supplied, the system is turned on. The Arduino provides power to the entire system, including the LCD display. The Arduino detects the soil moisture value, and if it's below 50, the DC pump motor turns on. If the value is above 100, the motor turns off. The soil moisture value is displayed on the LCD. The ultrasonic sensor value is also detected by the Arduino. If the value is above 10, the DC pump motor turns on. If it's below 10, the motor turns off. The Arduino stores the soil moisture and ultrasonic sensor data and sends it to the NodeMCU. The NodeMCU sends the data to Ubidots, where the data can be viewed on a mobile app.

The mobile app has two switches, Pump 1 and Pump 2, that can be turned on or off. When Pump 1 is turned on, the NodeMCU receives the data from Ubidots and turns on the relay. The relay then turns on DC Pump 1 motor and servomotor. When Pump 1 is turned off, the motors turn off.

When Pump 2 is turned on, the NodeMCU receives the data from Ubidots and turns on the relay. The relay then turns on DC Pump 2 motor. When Pump 2 is turned off, the motor turns off. The Ubidots mobile app displays the on and off times for Pump 1 and Pump 2. This system allows for remote control of the motors and monitoring of the soil moisture and ultrasonic sensor data from anywhere in the world.

3.4. ADVANTAGE OF PROPOSED SYSTEM

- Remotely accessible

we can physically control the motors anywhere by using the ubidotsapp. additionally, a review of the motor's starting time and power consumption will be provided, both numerically and graphically.

➤ Multiple operational modes

Having multiple operational modes allows the user to choose the most appropriate mode for their specific needs, and optimizes the performance and efficiency of the system

➤ Very low operating cost

The main advantage of the solar water pump is that it uses sunlight at no cost. just as solar water pumps reduce dependence on electricity or diesel, once installed, there is no recurrence cost of electricity or fuel..

➤ Simple and highly reliable

Since there is less dependence on electricity and fuel, power cuts, low voltage, and single phase problems such as solar water pumps do not affect the work.

➤ Environmentally friendly

other water pumps (diesel or propane-based) create noise and air pollution, while solar water pumps are environmentally friendly because they do not require any fuel to operate.

➤ Financially beneficial

solar water pumps can be a blessing for indian farmers. the solar water pumps can also be a source of income for the farmers.

3.5. CONSTRAINTS

The Universal Solar Pump Controller may have various constraints, including:

Power constraints: Since the Solar Pump Controller is designed to operate on solar power; it may face constraints due to fluctuations in solar radiation and availability of sunlight. This could impact the efficiency and reliability of the device.

Network Connectivity: The IoT-based Universal Solar Pump Controller requires a network connection to operate efficiently. However, if the network connection is poor or unavailable, the device may not function correctly.

Security: The device must be secure from cyber threats and hacking attempts to ensure its reliable and safe operation. The IoT devices are known for their vulnerabilities and can become a potential target for cyber-attacks.

Maintenance and Support: The device must have proper maintenance and support systems to ensure its reliable operation. It must also have a user-friendly interface to make it easy for users to operate and troubleshoot any issues that may arise.

Environmental Constraints: The solar pump controller project needs to be made to withstand extremes in temperature, humidity, and other environmental conditions in order to function well in isolated and rural areas.

3.6. TRADEOFFS

The universal solar pump controller project may involve several tradeoffs that need to be considered. Some of the potential tradeoffs are:

Cost vs. Performance: One of the primary tradeoffs in this project could be the cost of the solar pump controller versus its performance. Higher performance controllers with more features and capabilities will typically cost more than basic controllers. Thus, there may be a need to balance the cost with the desired level of performance.

Efficiency vs. Reliability: Another potential tradeoff in this project could be the efficiency of the solar pump controller versus its reliability. More efficient controllers may

be more complex and therefore less reliable than simpler, less efficient controllers. Thus, there may be a need to balance efficiency with reliability.

Flexibility vs. Compatibility: The universal aspect of the solar pump controller may require a tradeoff between flexibility and compatibility. A more flexible controller that can work with a wider range of solar panels and pumps may be less compatible with specific types of pumps or panels, while a less flexible controller may be more compatible with specific devices.

Features vs. Complexity: The solar pump controller may have various features that can be added, such as monitoring and remote control capabilities. However, adding more features may increase the complexity of the controller, which could make it more difficult to use and maintain.

Design vs. Manufacturing: There may be a tradeoffs between the design of the controller and its ease of manufacturing. A more complex design may be more difficult to manufacture, leading to higher production costs, while a simpler design may be easier to manufacture but may not meet all the desired performance requirements.

Overall, the tradeoffs will depend on the specific requirements and goals of the project, as well as the resources available to implement it. Careful consideration of the potential tradeoffs can help to ensure that the solar pump controller meets the desired performance requirements while remaining within budget and feasible to manufacture.

CHAPTER 4

SIMULATION ANALYSIS

4.1 SYSTEM DEVELOPMENT & TOOLS

Arduino IDE is an open-source software development environment that is used to program microcontrollers, specifically Arduino boards. It is a platform that is easy to use, and is aimed at both beginners and experienced programmers.

Proteus Professional is a comprehensive EDA tool that includes a schematic capture module, a PCB layout module, and a simulation module. The schematic capture module allows users to create and edit electronic schematics, while the PCB layout module enables users to design printed circuit boards (PCBs) with ease. The simulation module allows users to test and validate their circuit designs before they are fabricated, saving both time and money.

4.2 SIMULATION RESULTS

4.2.1 Flowchart for Simulation

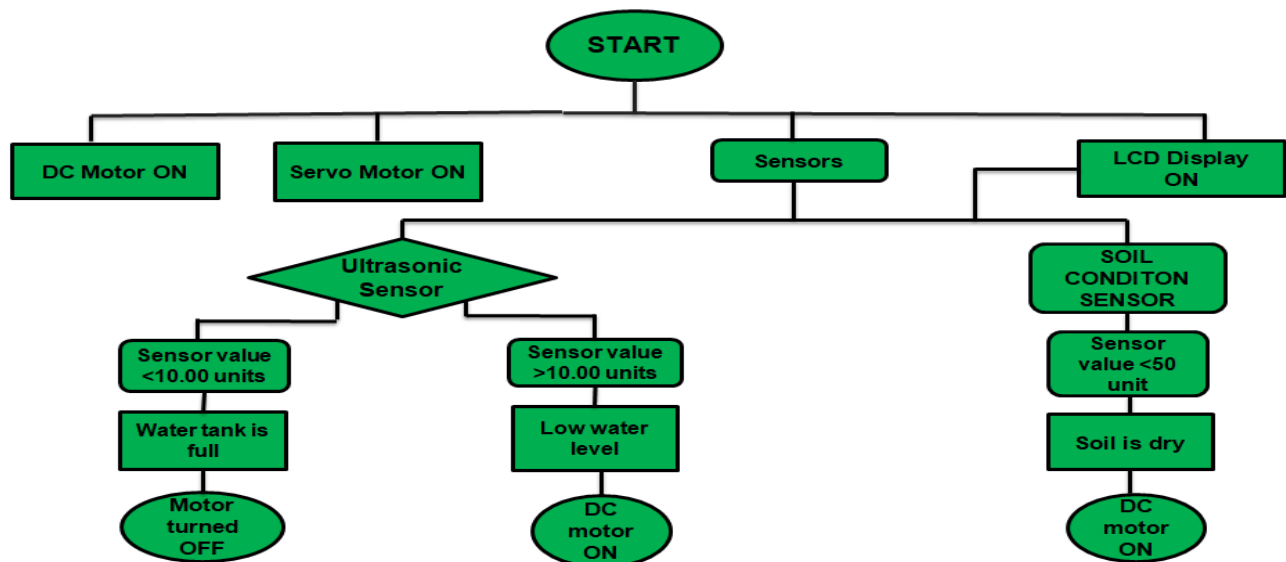


Fig 4.1 Flowchart for simulation

Fig 4.1 displays flow chart for Arduino ide code. Upon receiving power supply, the system initiates its operation. The Arduino board serves as the power source for all components, including the LCD screen. Next, the servo motor also on the Arduino board measures the soil moisture level, and if it falls below 50, it activates the DC pump motor. Conversely, if the level rises above 100, the motor is deactivated. The soil moisture level is presented on the LCD screen for easy monitoring.

In addition to the soil moisture level, the Arduino board also reads data from the ultrasonic sensor. If the recorded value is higher than 10, the DC pump motor turns on, while it is deactivated if the value falls below 10.

4.2.2 Simulation results for ultrasonic sensor

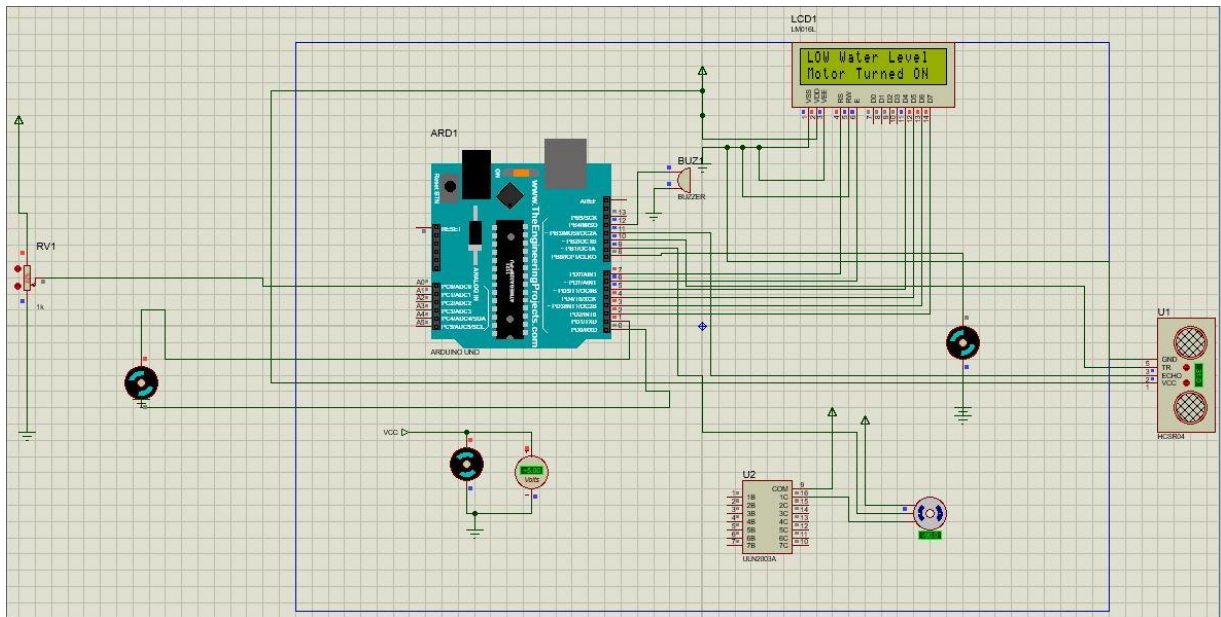


Fig 4.2 Low Water Level Motor Turned ON

This fig 4.2 represents when the Ultrasonic Sensor value is greater than 10 values, it indicates that the reservoir's water level is low, and the motor is turned on.

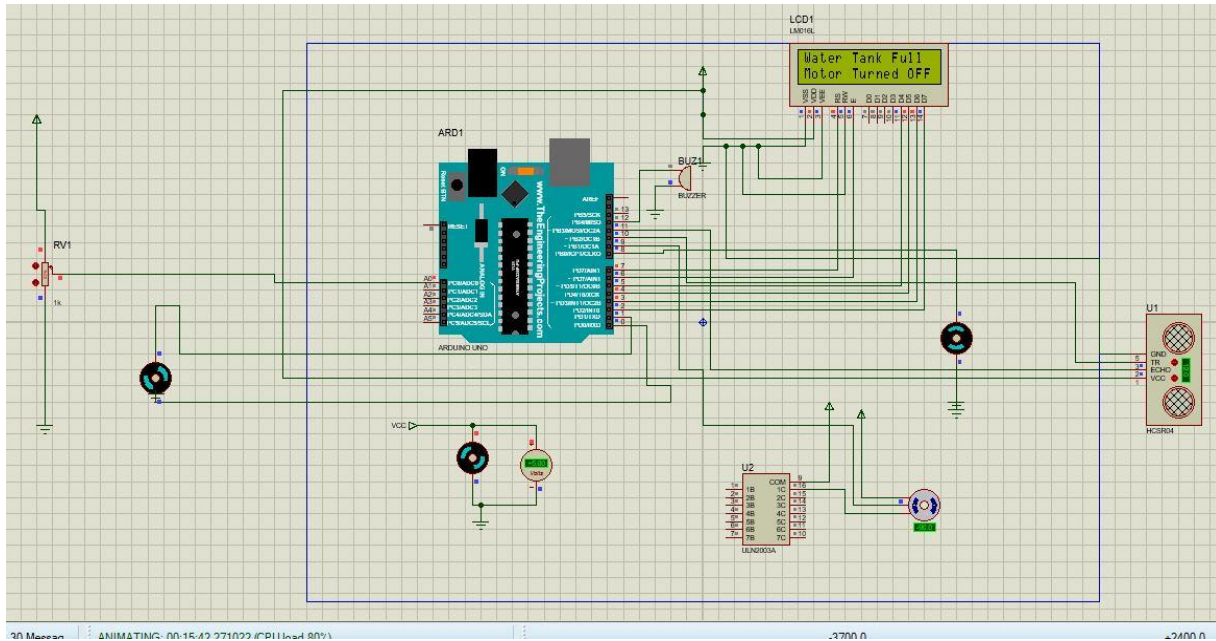


Fig 4.3 Water Tank Full Motor Turned OFF

This fig 4.3 represents when the ultrasonic sensor value is less than 10 values, it means the water level in the reservoir is full, and the motor is turn.

4.2.3 Simulation results for soil moisture sensor

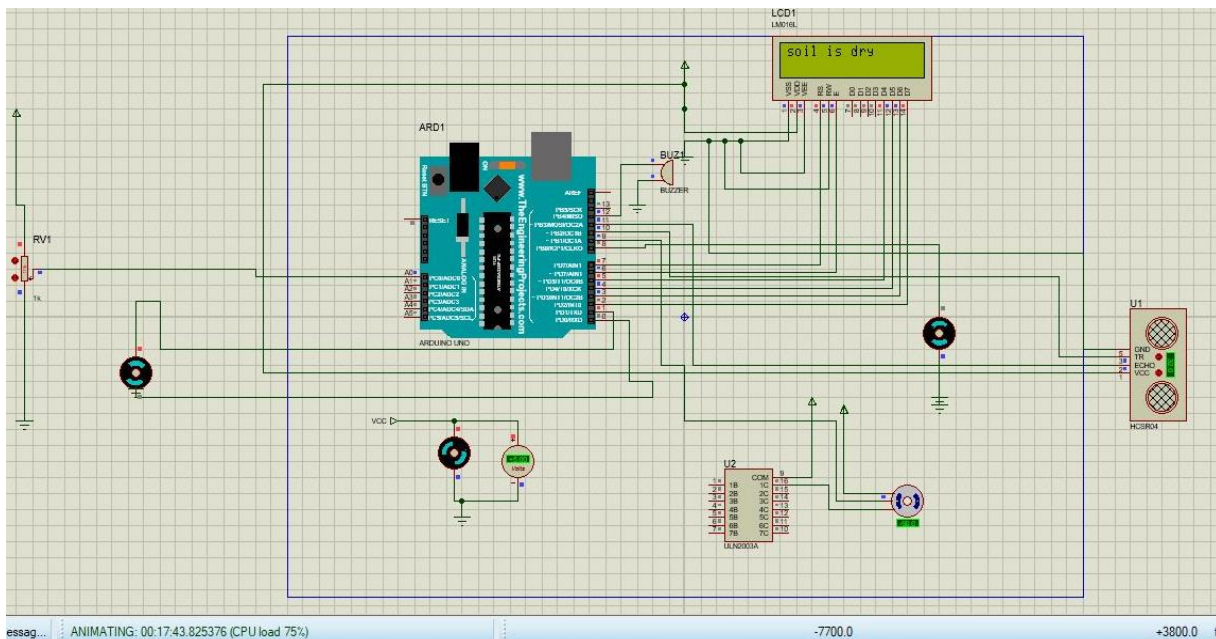


Fig 4.4 Soil Dry Motor Turned ON

This fig 4.4 represents when the soil moisture sensor value is less than 50 values, it means the soil is dry, and the motor is turned on. When the soil moisture sensor value exceeds 100 values, it means the soil is wet, and the motor is turned off.

CHAPTER - 5

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

5.1 COMPONENTS USED:

5.1.1 Arduino UNO

Arduino Uno is a single-board microcontroller intended to make the application of interactive objects or environments more accessible. The board is based on the ATmega328. Current models feature a USB interface, a power jack, 6 analog input pins, a 16MHz ceramic resonator, an ICSP Header and a reset button.

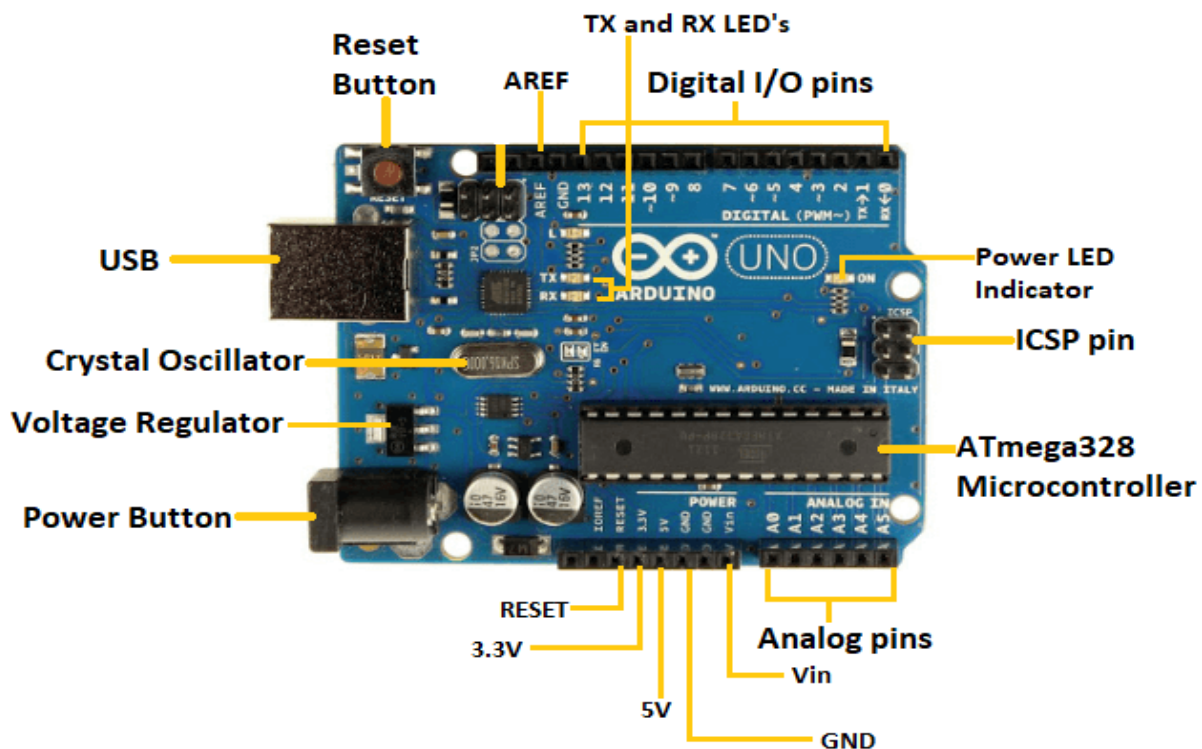


Fig 5.1 Arduino UNO

5.1.2 Node MCU

"NodeMCU" naturally alludes to the firmware instead of the dev units. The firmware utilizes the Lua scripting language. It depends on the eLua project and based on the

Espressif Non-OS SDK for ESP8266. It utilizes many open source projects, for example, Lua-cjson and spiffs. LUA based natural firmware for Expressif ESP8622 Wi-Fi SoC, similarly as an open source gear board that despite the \$3 ESP8266 Wi-Fi modules consolidates a CP2102 TTL to USB chip for programming and exploring, is breadboard-obliging, and can essentially be constrained through its small USB port.

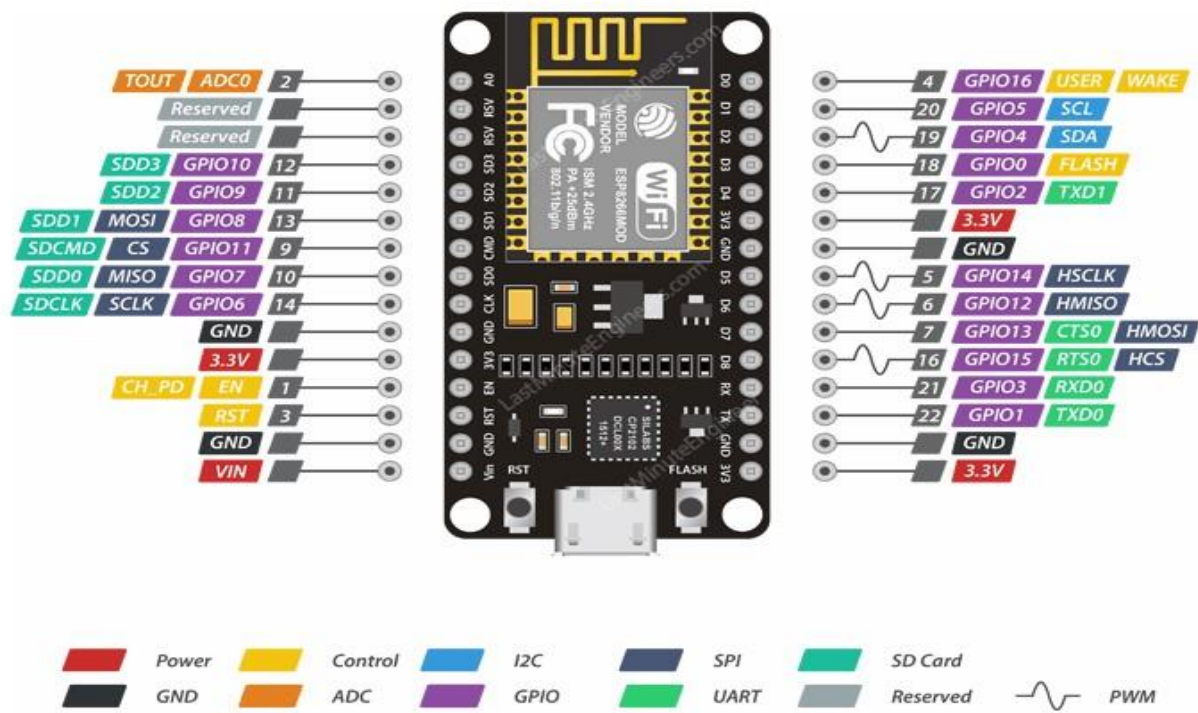


Fig 5.2 ESP8266 Node MCU Pinout

5.1.3 Soil Moisture Sensor

A soil moisture sensor is an electronic device that measures the amount of moisture present in soil. These sensors typically consist of two metal probes that are inserted into the soil to a certain depth. One probe sends a small electrical signal into the soil, while the other measures the resistance of the soil to the signal.



Fig 5.3 Soil Moisture Sensor

5.1.4 Ultrasonic Sensor

An ultrasonic sensor is a device that uses sound waves to detect the presence or absence of objects. It works by emitting high-frequency sound waves that bounce off of objects and return to the sensor. The sensor measures the time it takes for the sound waves to return, and uses this information to calculate the distance between the sensor and the object. Ultrasonic sensors are commonly used in a variety of applications, such as robotics, industrial automation, and automotive parking assistance systems.



Fig 5.4 Ultrasonic Sensor

5.1.5 Servo Motor

A servo motor is a type of motor that is commonly used in robotics and other applications where precise control of movement is required. Servo motors are designed

to rotate to a specific angle and hold that position until instructed to move to a new position. They work by receiving a signal from a control device, such as a microcontroller, which tells the motor where to move and how fast to move. The motor then adjusts its position and speed to match the signal, allowing for precise control of movement.



Fig 5.5 Servo Motor

5.1.6 DC Pump Motor

A DC pump motor is a type of electric motor that is used to power various types of pumps. These motors are designed to convert electrical energy into mechanical energy, which is used to rotate the pump impeller and move fluid through a system. DC pump motors are commonly used in a variety of applications, such as water pumps for irrigation systems, aquarium pumps, and fuel transfer pumps.



Fig 5.6 5-12VDCMotor

5.1.7 Relay Module

A relay module is an electronic device that is used to control high voltage and high current circuits using low voltage and low current signals. A relay module typically consists of a small control circuit and one or more relays, which are switches that are activated by an electrical signal. When the control circuit receives a signal, it activates the relay, which then switches the high voltage or high current circuit on or off.



Fig 5.7 Relay Module

5.1.8 LCD Display

LCD stands for liquid crystal display. They come in many sizes 8x1, 8x2, 10x2, 16x1, 16x2, 16x4, 20x2, 20x4, 24x2, 30x2, 32x2, 40x2 etc. Many multinational companies like Philips Hitachi Panasonic make their own special kind of LCD'S to be used in their products. All the LCD'S performs the same functions (display characters numbers special characters ASCII characters etc). Their programming is also same and they all have same 14 pins (0-13) or 16 pins (0 to 15). Alphanumeric displays are used in a wide range of applications, including palmtop computers, word processors, photocopiers, point of sale terminals, medical instruments, cellular phones, etc.

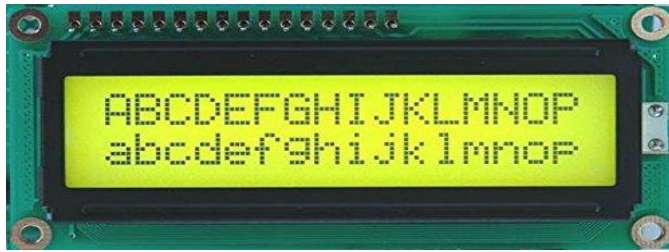


Fig 5.8 16X2 LCD Display

5.1.9 Solar Panel

Solar energy begins with the sun. Solar panels (also known as "PV panels") are used to convert light from the sun, which is composed of particles of energy called "photons", into electricity that can be used to power electrical loads.

A photovoltaic system typically includes a panel or an array of solar modules, a solar inverter, and sometimes a battery and/or solar tracker and interconnection wiring. A single solar module can produce only a limited amount of power, most installations contain multiple modules. The majority of modules use wafer-based crystalline silicon cells or thin-film cells based on cadmium telluride or silicon.



Fig 5.9 Solar panel

5.2 Arduino UNO Interfacing Node MCU

To send data from an Arduino Uno to a NodeMCU, you can use a serial communication protocol. First, connect the Arduino Uno's TX (transmit) pin to the NodeMCU's RX (receive) pin, and the Arduino Uno's RX (receive) pin to the NodeMCU's TX (transmit) pin. Then, in the Arduino IDE, write a program that reads the data you want to send and sends it over the UART serial port. For example, you can read the value from a sensor connected to an analog pin and send it over the serial port using the `Serial.println()` function.

In the NodeMCU, write a program that reads the data sent over the serial port and does something with it. For example, you can use the `Serial.available()` function to check if there is data available on the serial port, and then use the `Serial.parseInt()` function to read an integer value. You can then use this value to control other devices connected to the NodeMCU, or send it over the internet using WiFi. Make sure that the baud rate used by the Arduino Uno and NodeMCU match, and that you have selected the correct serial ports in both the Arduino IDE and the NodeMCU programming environment.

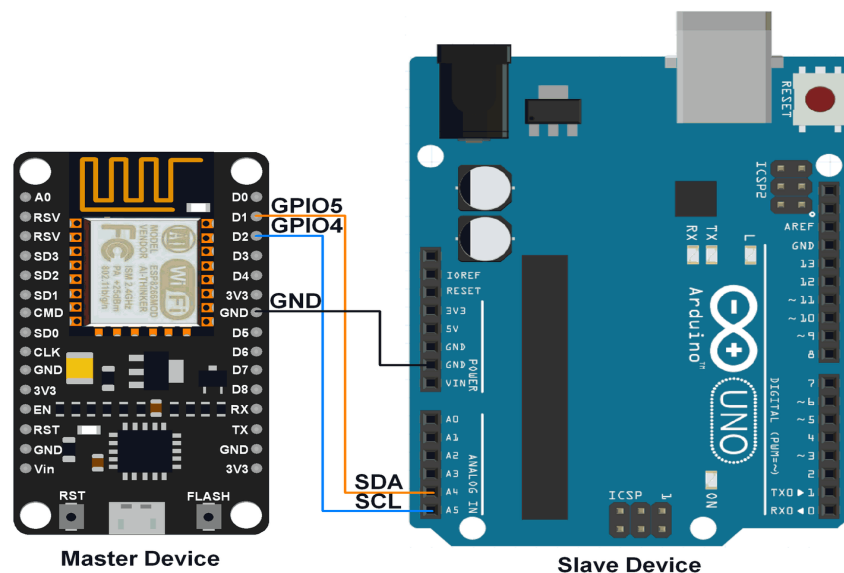


Fig 5.10 Arduino UNO Interfacing NODEMCU

5.3 UBIDOTS

- Ubidots is an Internet of Things (IoT) application builder with data analytics and visualization.
- We can physically control the motors anywhere by using the ubidots app.
- Additionally, a review of the motor's starting time and power consumption will be provided, both numerically and graphically.

5.3.1 Connect a Node MCU ESP8266 to UBIDOTS

- To be able to work with the NodeMCU ESP8266 platform in the Arduino IDE, you will need to install the ESP8266 platform using the preconfigured Arduino Board Manager. If you are not familiar with adding a board with the Arduino IDE.
- With the ESP8266 platform installed, select the ESP8266 device you are working with. In the case, we are working with a “NodeMCU 1.0(ESP12E module)”. To select your board from the Arduino IDE, select Tools > Board “NodeMCU 1.0(ESP12E module)”.
- Download and install the Ubidots library.
- With the following example, you will be able to simulate random readings taken from the NodeMCUESP8266 to Ubidots.
- To begin posting values to Ubidots, open the Arduino IDE and paste the sample code below. Once you have pasted the code, be sure to assign the following parameters:
- SSID (WiFi Name) & Password of the available network connection.
- Ubidots TOKEN.

```

/*****
 * Include Libraries
 *****/

#include "Ubidots.h"

/*****
 * Define Instances and Constants
 *****/

const char* UBIDOTS_TOKEN = "..."; // Put here your Ubidots TOKEN
const char* WIFI_SSID = "..."; // Put here your Wi-Fi SSID
const char* WIFI_PASS = "..."; // Put here your Wi-Fi password

Ubidots ubidots(UBIDOTS_TOKEN, UBI_HTTP);

/*****
 * Auxiliar Functions
 *****/

// Put here your auxiliar functions

/*****
 * Main Functions
 *****/

void setup() {

    Serial.begin(115200);
    ubidots.wifiConnect(WIFI_SSID, WIFI_PASS);
}

```

Fig 5.11 UBIDOTS Setup Code

- Verify your code within the Arduino IDE. To do this, in the top left corner of our Arduino IDE you will see the "Check Mark" icon; press it to verify your code.
- Upload the code into your "NodeMCU 1.0(ESP12E module)". To do this, choose the "right-arrow" icon beside the "check mark" icon.
- To verify the connectivity of the device and the data sent, open the serial monitor by selecting the "magnifying glass" icon in the top right corner of the Arduino IDE to see the connectivity logs.
- Confirm your data in Ubidots. Now you should see the posted data in your Ubidots account.

With the following sample code you will be able to get a value from Ubidots to start controlling any asset you desire.

1. To begin getting values from Ubidots, open the Arduino IDE and paste the sample code below. Once you have pasted the code, be sure to assign the following parameters:

- **SSID (WiFi Name) & Password** of the available network connection.
- **Ubidots TOKEN.**
- **Device Label** of the device which contains the variable to want to GET.
- **Variable Label** of the variable you want to GET

2. Verify & Upload the code into the board following the same steps provided in the POST step above.
3. To verify the connectivity of the device and the data which is being received, open the serial monitor by selecting the "magnifying glass" icon in the top right corner of the Arduino IDE to see the connectivity logs.
4. In the serial monitor, you will be able to see the last value received in Ubidots of the variable specified in the firmware.

5.4 PROPOSED SYSTEM FLOWCHART

5.4.1 ARDUINO UNO PROCESS

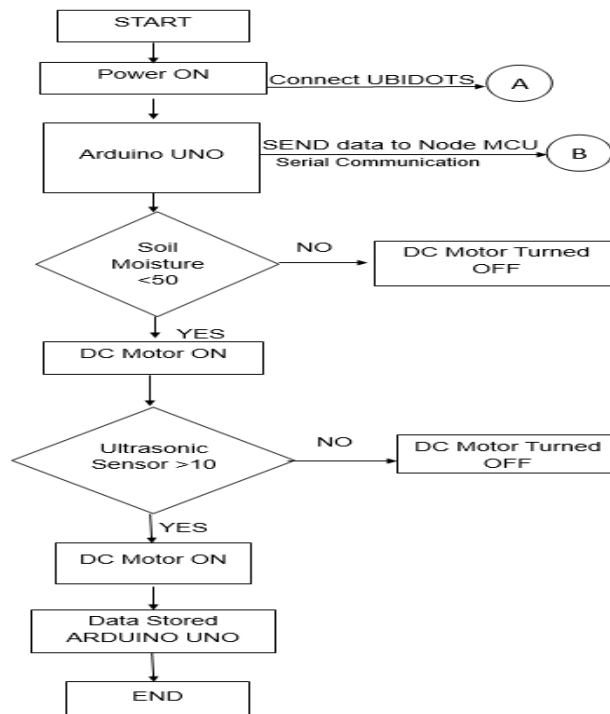


Fig 5.12 Flow Chart for Arduino UNO

Fig 5.12 displays flow char for Arduino UNO process. Upon receiving power, the system is activated, with the Arduino serving as the power source for all components, including the LCD display. The Arduino is programmed to detect the moisture level in the soil, and if it falls below 50, the DC pump motor is activated to supply water to the plant. Conversely, if the moisture level exceeds 100, the motor is turned off to prevent overwatering. The LCD display shows the current soil moisture level for easy monitoring.

Additionally, the Arduino uses an ultrasonic sensor to measure the distance between the sensor and the water level in the reservoir. If the distance is above 10, indicating that there is sufficient water, the DC pump motor is turned on to water the plant. If the distance falls below 10, the motor is turned off to prevent water shortage.

The Arduino records the data from the soil moisture sensor and the ultrasonic sensor and transmits it to the NodeMCU for further processing. The NodeMCU then sends the data to Ubidots, where it is stored and made accessible via a mobile application for remote monitoring of the plant's growth and health.

5.4.2 UBIDOTS PROCESS

The mobile application contains two switches, namely Pump 1 and Pump 2, which can be toggled on or off by the user. Upon activation of Pump 1, the NodeMCU receives data from Ubidots, prompting the relay to activate. Subsequently, the DC Pump 1 motor and servomotor are also turned on. The motors will be turned off upon deactivation of Pump 1.

Similarly, upon activation of Pump 2, the NodeMCU retrieves data from Ubidots, which in turn triggers the relay to turn on DC Pump 2 motor. The motor will subsequently be turned off upon deactivation of Pump 2. The mobile application displays the on and off times for both pumps, providing real-time monitoring of the system's activity

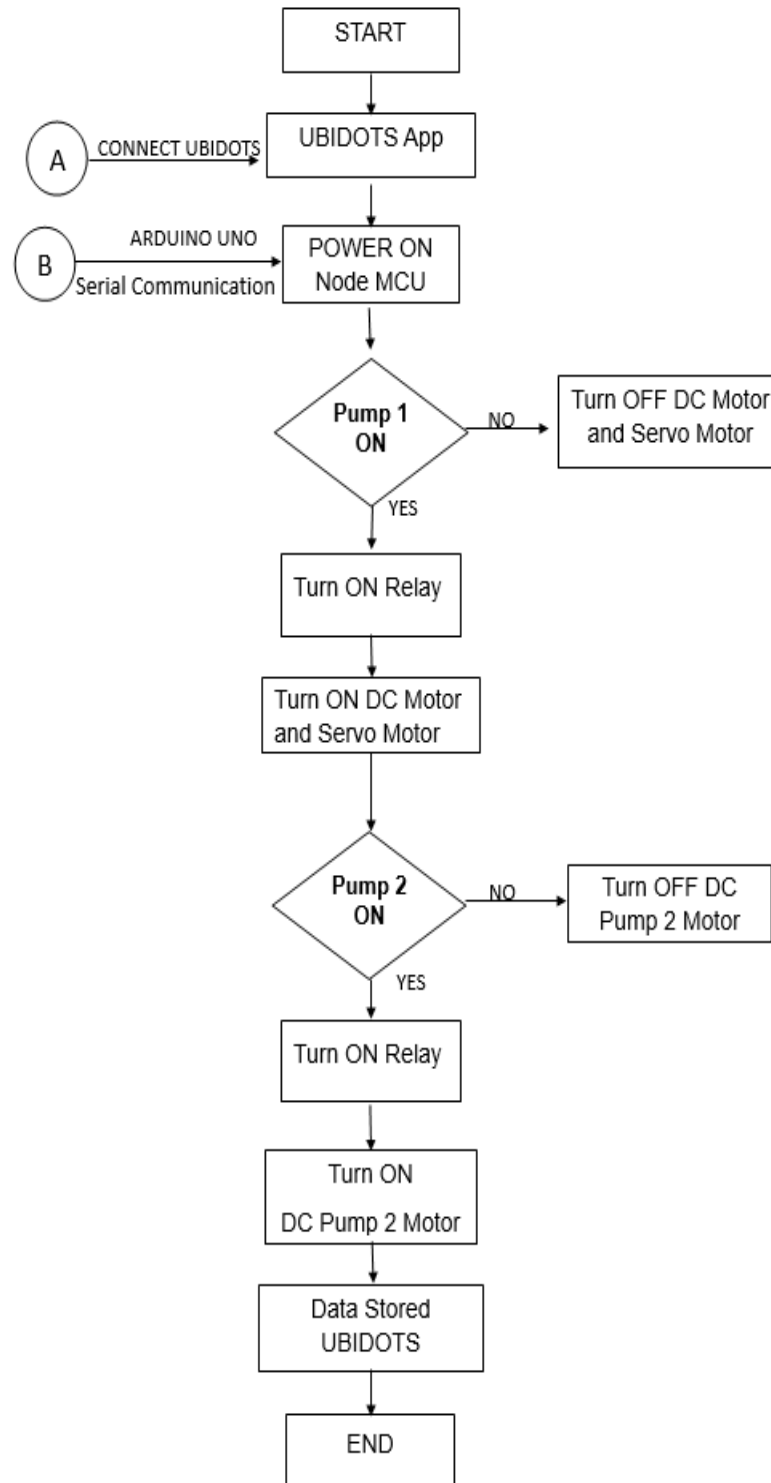


Fig 5.13 UBIDOTS Process Flowchart

Fig 5.13 displays UBIDOTS process flowchart. Overall, this system enables remote control of the motors and facilitates the monitoring of soil moisture and ultrasonic sensor data from any location worldwide.

5.5 UBIDOTS RESULTS

These are the results we got from the ubidots app. We attached a screenshot of them. For this result, we wrote code using the ubidots library and we successfully got our results.

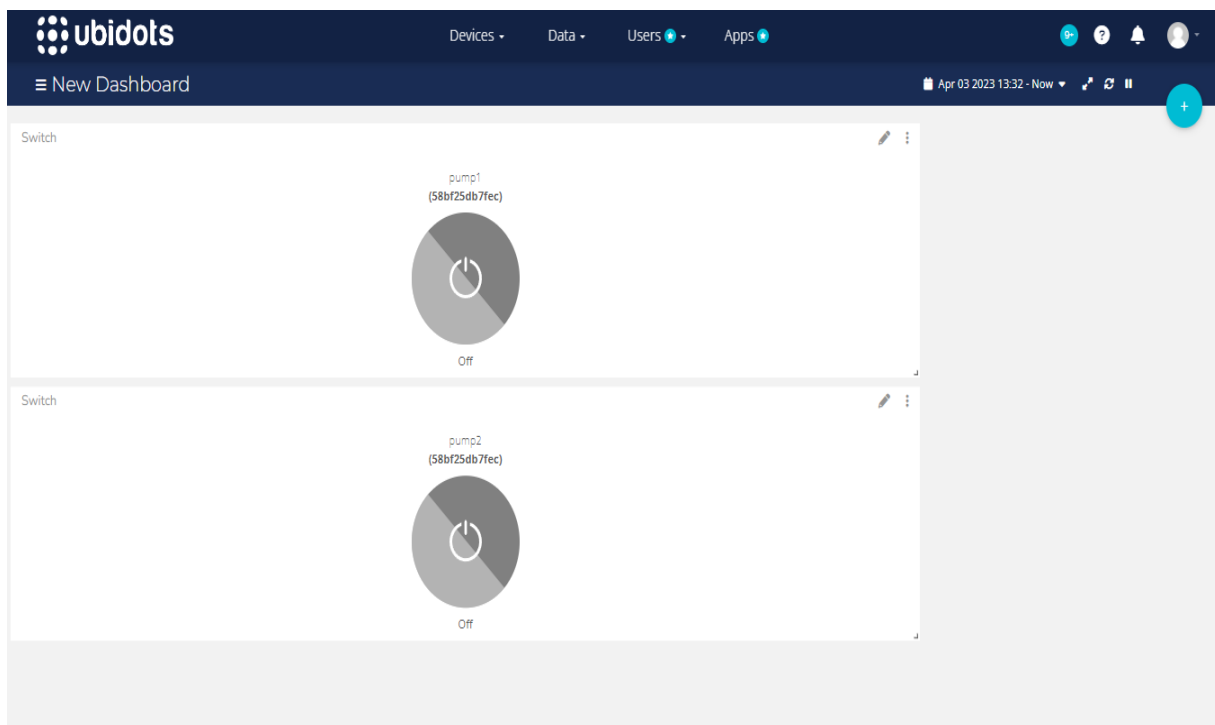


Fig 5.14 Pump Switches

Fig 5.14 displays UBIDOTS Pump switches, Pump 1 and Pump 2. Both the switches can be turned ON and OFF simultaneously. We can turn on and off the motors anywhere in the world with the use of the ubidots app.

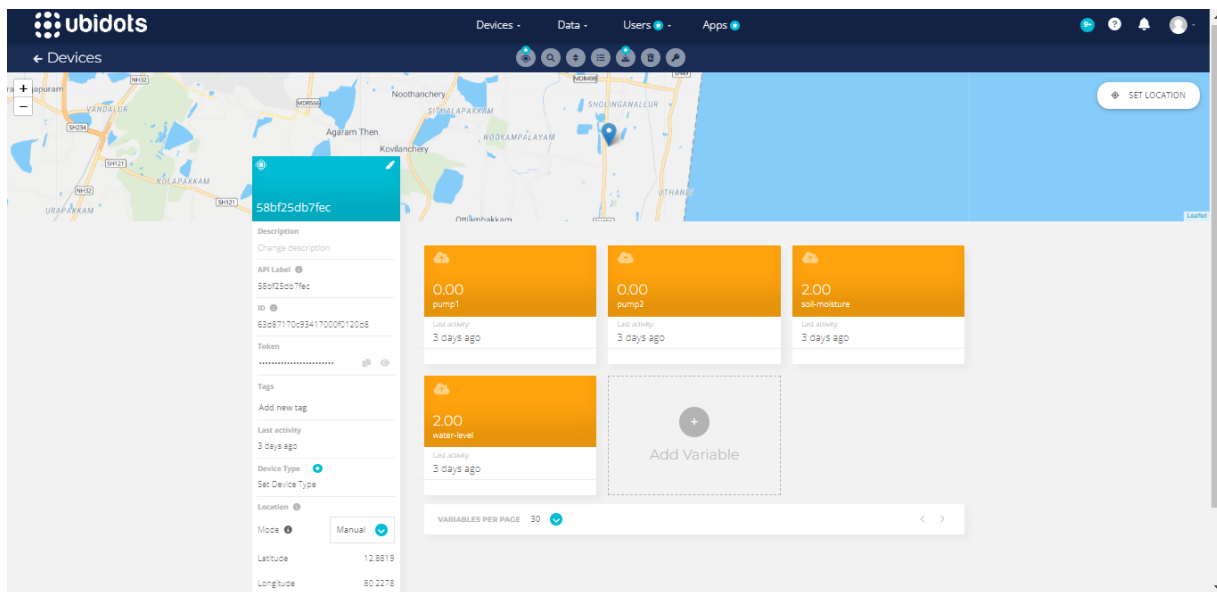


Fig 5.15 Data Page

Fig 5.15 displays UBIDOTS Data page, which has four types of devices, Pump 1, Pump 2, Soil moisture sensor and Ultrasonic sensor. The data of all the devices can be viewed both Numerically and Graphically. we can see the motors data anywhere in the world with the use of ubidots app..

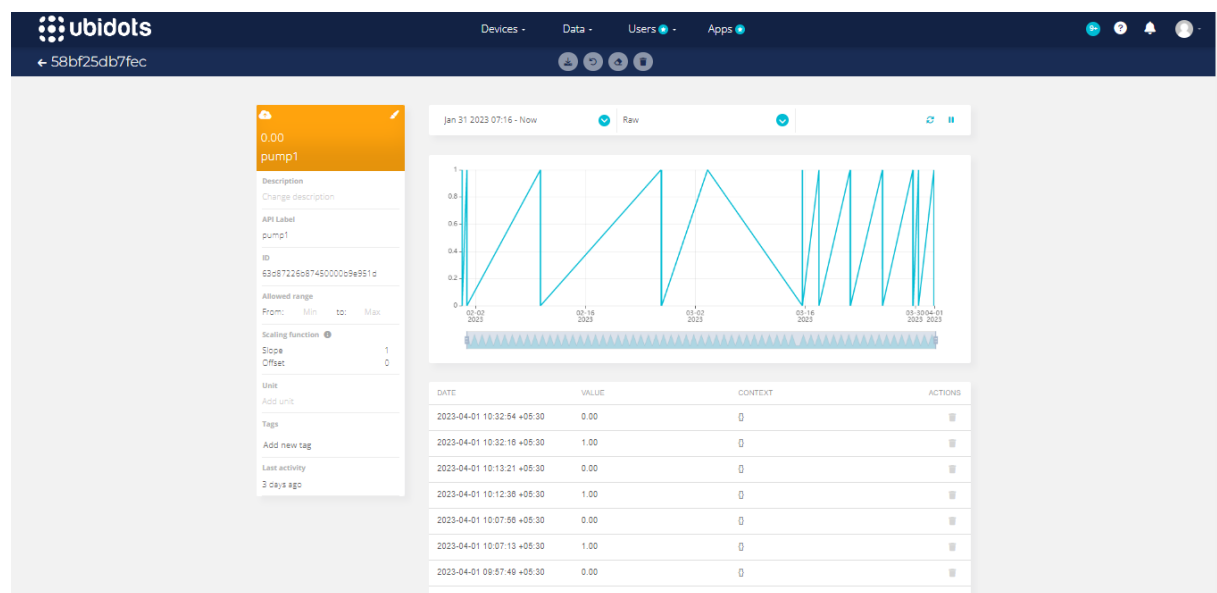


Fig 5.16 Pump 1 Data

This figure 5.16 displays Pump 1 ON and OFF time, both Numerically and Graphically in UBIDOTS App. we can see the pump 1 motor data anywhere in the world with the use of ubidots app..

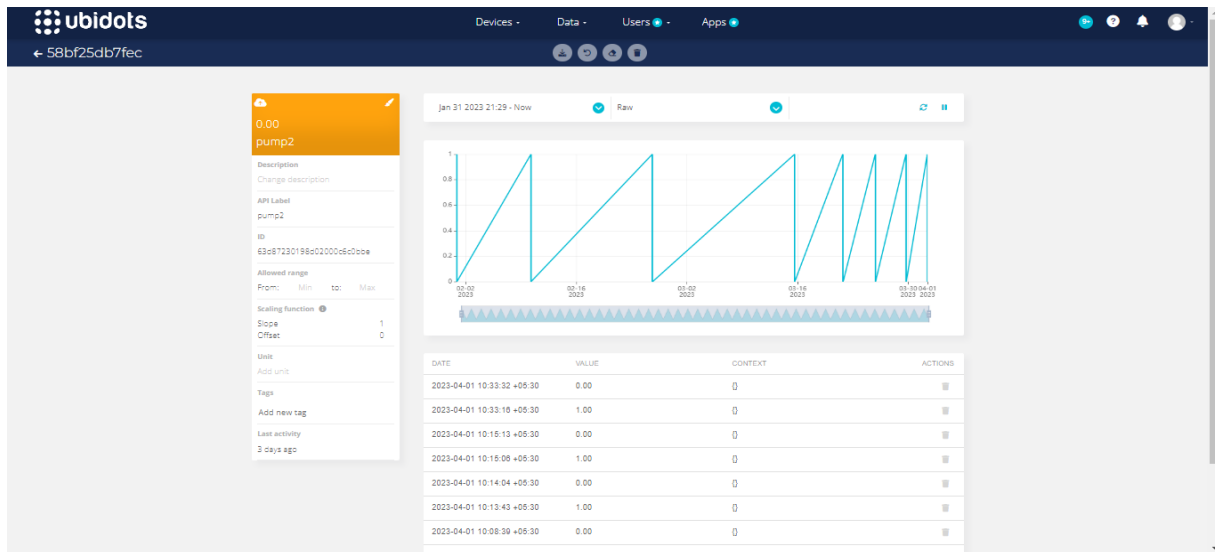


Fig 5.17 Pump 2 Data

This figure 5.17 displays Pump 2 ON and OFF time, both Numerically and Graphically in UBIDOTS App. we can see the pump 2 motor data anywhere in the world with the use of ubidots app.

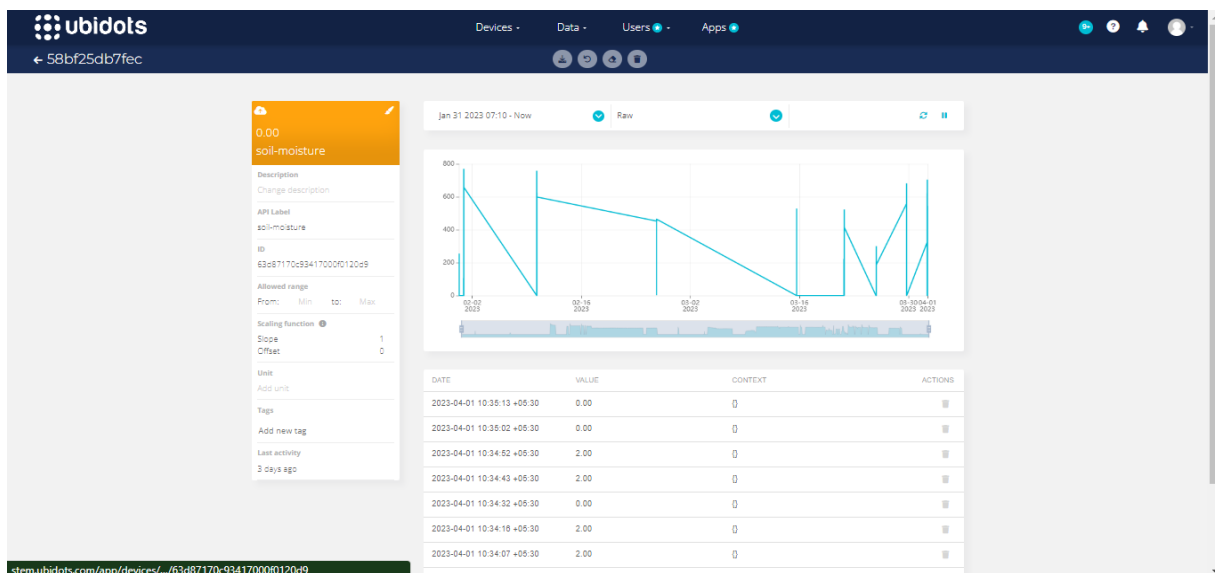


Fig 5.18 Soil Moisture Sensor Data

This figure 5.18 displays Soil Moisture Sensor Data, both Numerically and Graphically in UBIDOTS App with the help of Node MCU sent Data. we can see the soil moisture data anywhere in the world with the use of ubidots app.

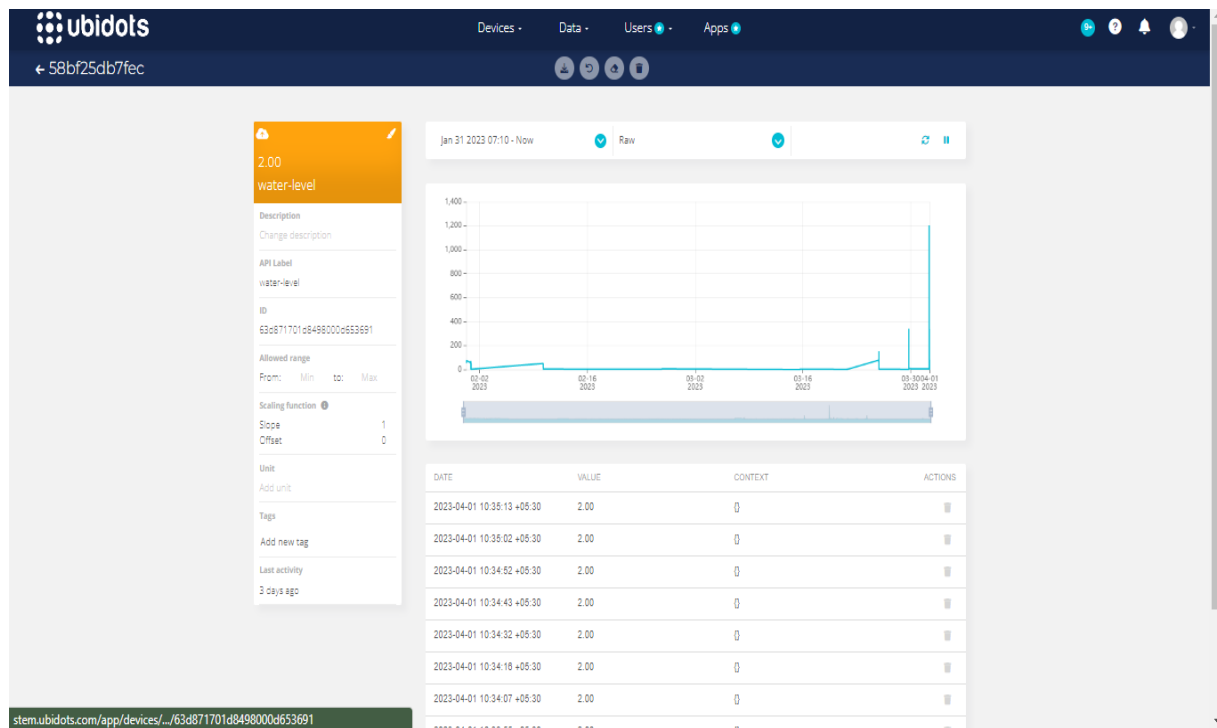


Fig 5.19 Ultrasonic Sensor Data

This figure 5.19 displays Ultrasonic Sensor Data, both Numerically and Graphically in UBIDOTS App with the help of Node MCU sent Data. we can see the ultrasonicsensor data anywhere in the world with the use of ubidots app.

5.6 HARDWARE MODEL

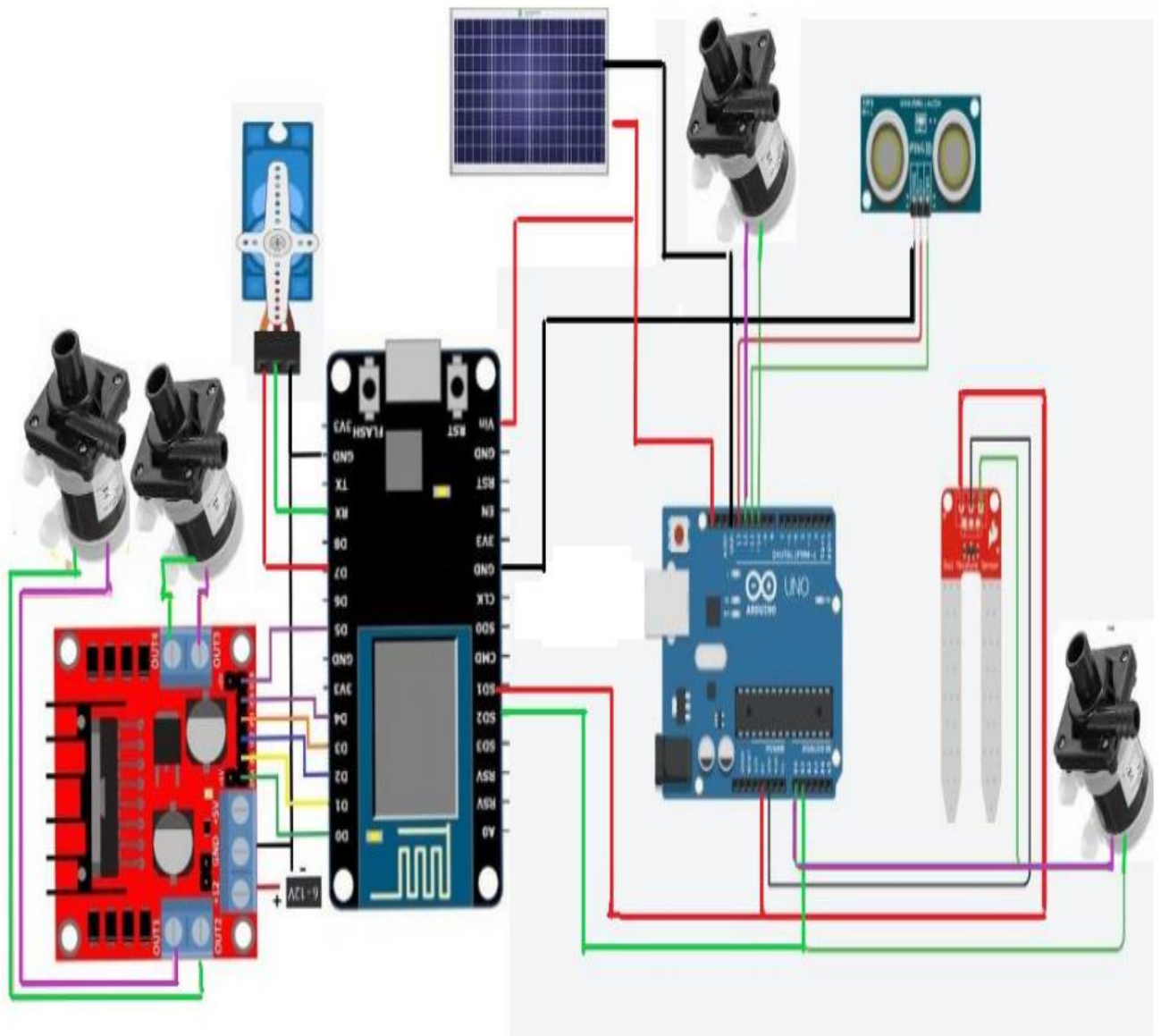


Fig 5.20 Hardware Circuit Layout Diagram

Fig 5.20 displays Hardware Circuit Layout Diagram and Fig 5.21 displays Prototype of Universal Solar Pump Controller. This prototype has a variety of components, including a Solar panel, Relay, LM298 Motor driver, Node MCU, Servo motor, Arduino UNO controller, Soil moisture sensor, Ultrasonic sensor and DC motors.

Through the successful implementation of USPC, we are now able to control the motors manually and automatically. As a result, the Ubidots app allows us to view the motor data numerically and graphically. Therefore, the project has fulfilled its objective.

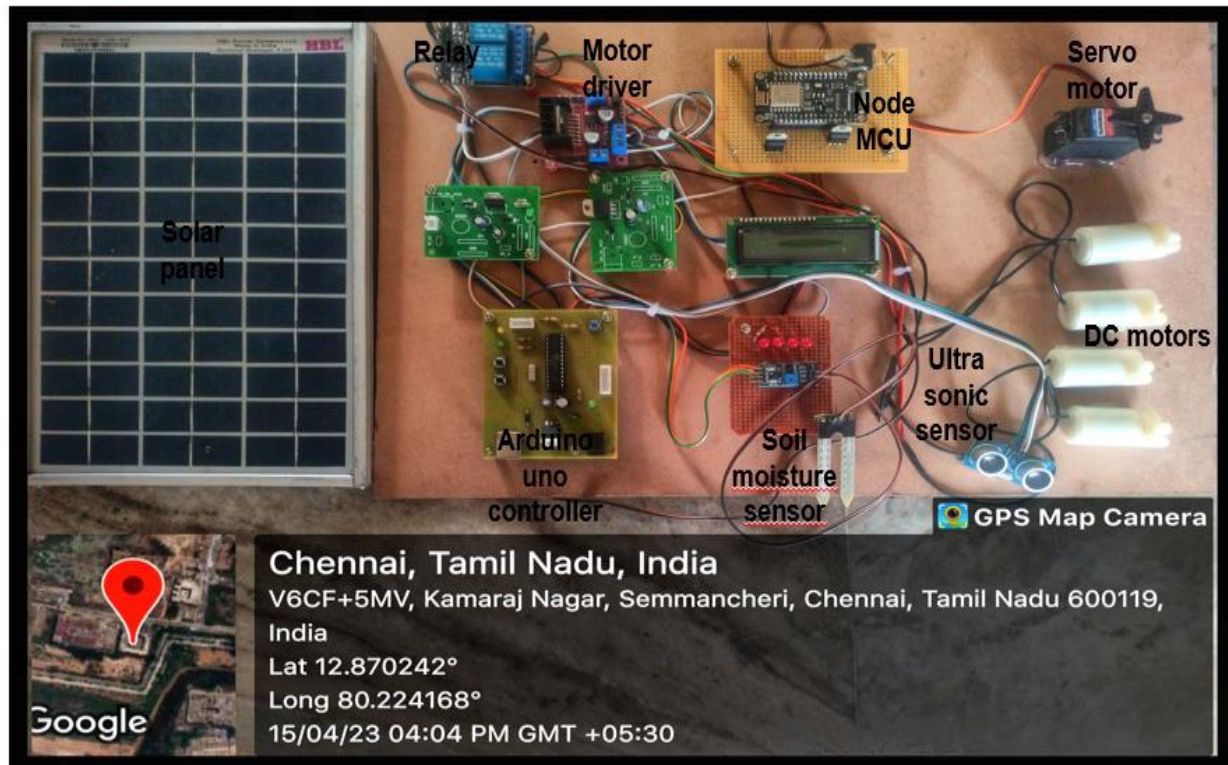


Fig 5.21 Prototype of Universal Solar Pump Controller

Overall, the universal solar pump controller is a cost-effective solution for off-grid applications, especially in areas where there is no access to electricity or where electricity supply is unreliable.

For this project, we used the following IEEE standards :

IEEE 1547: This standard provides guidelines for interconnecting distributed energy resources (DERs) with the electric power system. This includes solar pump systems that may be connected to the grid, and it covers safety, performance, and reliability requirements.

IEEE 802.15.4: This standard specifies the physical and media access control (MAC) layer protocols for low-rate wireless personal area networks (LR-WPANs). This can be used for wireless communication between components of the solar pump system, such as sensors and the controller.

IEEE 802.11: This standard specifies the protocols for wireless local area networks (WLANs), commonly known as Wi-Fi. This can be used to connect the solar pump controller to a network for remote monitoring and control.

IEEE 2030.5: This standard provides a communication protocol for integrating DERs, energy storage systems, and electric vehicles with the power grid. It can be used to ensure that the solar pump system operates in coordination with the electric power system, and to enable demand response and other grid services.

IEEE 1888: This standard provides a common data format and communication protocol for IoT devices and systems. It can be used to ensure that the data exchanged between components of the solar pump system is standardized and interoperable.

CHAPTER – 6

CONCLUSION

6.1. CONCLUSION

A universal solar pump controller (USPC) was conceptualized, designed, and a working prototype was constructed to validate the concept. Water and energy conservation during irrigation was the main focus of the USPC's development. Results of testing the USPC showed that farmers' water usage would be reduced significantly by 25.2%, and their electricity costs would be significantly reduced by 57.8%. Several experiments were carried out to test the system's efficacy, and the findings were satisfactory.

In the context of a real farm, it is necessary to employ more soil moisture sensors to cover the entire soil area effectively. To further enhance the system's efficiency, future improvements could focus on increasing the solar panel size and battery bank to optimize the system's performance on days that are cloudy and sunny. The existing solar panel could only be used from three angles, but this restriction can be removed by using a solar panel that can track the sun's direction automatically.

The mobile app currently stores information about the sensors and automates the watering process. Future iterations of the app could include rainfall detection and IoT capabilities for storing the data. This would enhance the irrigation system's automation, allowing farmers to have real-time access to crucial data that would help them make informed decisions.

6.2 FUTURE SCOPE

Better energy storage solutions: The efficiency of solar panels is increasing, but energy storage remains a significant challenge. Future developments in battery technology can help to improve the reliability of solar pumps by providing consistent power even during cloudy days or at night.

Improved durability: Solar pump controllers need to be able to withstand harsh environmental conditions, such as extreme temperatures, humidity, and dust. Advancements in material science can lead to more durable and robust controllers that can operate in harsh environments for longer periods.

Enhanced automation capabilities: With the use of advanced algorithms and machine learning, solar pump controllers can be made more intelligent and autonomous. This can help to optimize pump performance, reduce energy consumption, and prolong the lifespan of the system.

REFERENCES

1. Apoorva Sharma, Tripurari Nath Gupta, Mahiraj Singh Rawat, "Grid Connected Solar PV fed Constant Power Water Pumping System", 2021 International Conference on Intelligent Technologies (CONIT).
2. A.R. Al-Ali, A.A Nabulsi a, S. Mukhopadhyay, M.S. Awal, S. Fernandes, K. Ailabouni, IoT-Solar Energy Powered Smart Farm Irrigation system, *Journal of Electronic Science and Technology*), vol. 1, no. 100017, pp. 1 - 14, 2020.
3. A K Singh, K. A Chinmaya and M. Badom, "Solar PV and GridBased Isolated Converter for Plug-in Electric Vehicles," in *IET PowerElectronics*, vol. 12, no. 14, pp. 3707-3715, 27 11 2019.
4. A K. Singh, M. Badoni and Y N Tatte, "A Multifunctional Solar PVand Gnd Based On-Board Converter for Electric Vehicles," in *IEEE Transactions on Vehicular Technology*, vol. 69, no 4, April 2020.
5. Anshul Varshney, Utkarsh Sharma, Bhim Singh, "An Intelligent Grid Integrated Solar PV Array Fed RSM Drive-Based Water Pumping System", *IEEE Transactions on Industry Applications*, vol.57, no.2, pp.1818-1829, 2021.
6. Anjanee Kumar Mishra, Bhim Singh, "An Improved Control Technique for Grid Interactive 4-Phase SRM Driven Solar Powered WPS Using Three-Level Boost Converter", *IEEE Transactions on Industrial Informatics*, vol.17, no.1, pp.290-299, 2021.
7. C. A. Bolu, J. Azeta, F. Alele, E.O. Daranijo, P. Onyeubani, A.A. Abioye, Solar Powered Microcontroller-based Automated Irrigation System with Moisture Sensors, 2019.
8. G. Esakkimuthu, K. Rajesh, A. Ramkumar, K. Kathiravan, "PV Based Drive Control System and MPPT Technique under various Irradiation conditions - A Study", 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), pp.525-532, 2022.
9. López-Vargas, A., Fuentes, M., García, M. V., & Muñoz-Rodríguez, F. J. (2019). Low-Cost Datalogger Intended for Remote Monitoring of Solar Photovoltaic Standalone Systems Based on Arduino™. *IEEE Sensors Journal*, 19(11), 4308-4320.

10. Mohd. Kashif, Shadab Murshid, Bhim Singh, "Adaptive Hybrid Generalized Integrator Based SMO for Solar PV Array Fed Encoderless PMSM Driven Water Pump", IEEE Transactions on Sustainable Energy, vol.12, no.3, pp.1651-1661, 2021.
11. Ogidan, O. K., Onile, A. E., & Adegboro, O. G. (2019). Smart Irrigation System: A Water Management Procedure. Agricultural Sciences, 10(01), 25.
12. P. Santiya, J. Monah, G. Lakshmitha, T. Akshaya, "Smart Irrigation System Using Arduino and android," in International Journal of Engineering Research and Technology (IJERT), Namakkal, 2019.
13. R Kumar and B. Singh, "Solar PV powered-sensorless BLDC motordriven water pump," in IET Renewable Power Generation, vol. 13,no. 3, pp. 389-398. 252 2019.
14. Rashmi Rai, Saurabh Shukla, Bhim Singh, "Reactive Power Based MRAS for Speed Estimation of Solar Fed Induction Motor With Improved Feedback Linearization for Water Pumping", IEEE Transactions on Industrial Informatics, vol.16, no.7, pp.4714-4725, 2020.
15. Samara, S., & Natsheh, E. (2019). Intelligent Real-Time Photovoltaic Panel Monitoring System Using Artificial Neural Networks. IEEE Access, 7, 50287-50299.
16. Singh, R. S. S., Nurdin, M. I. B., &Chiew, W. Y. (2019) Raspberry Pi Zero Wireless Monitoring System for Analyzing Solar Photovoltaic Panel.
17. S. Shukla and B. Singh, "Reduced Current Sensor Based Solar PV Fed Motion Sensorless Induction Motor Drive for Water Pumping." in IEEE Transactions on Industrial Informatics, vol. 15, no. 7, pp.. 3973-3986, July 2019.
18. S. Murshid and B. Singh, "Analysis and Control of Weak Grid Interfaced Autonomous Solar Water Pumping System for Industrial and Commercial Applications," in IEEE Transactions on Industry Applications, vol. 55, no. 6, pp. 7207-7218, Nov-Dec 2019.
19. Sheshadri Shekhar Rauth, Banshidhari Samanta, "A Grid-Connected Solar Photovoltaic Hybrid System for Reliable Power and Water Supply in Modern Irrigation Application", Advances in Smart Grid Automation and Industry 4.0, vol.693, pp.493, 2021.

20. Ujjwal Kumar Kalla, Nikhil Bhati, Kanchan Chariya, Irfan Qureshi, "Design and Analysis of Solar PV Fed IMD Water - Pumping System", 2021 International Conference on Sustainable Energy and Future Electric Transportation (SEFET).

APPENDIX SOURCE CODE

PROGRAM FOR ARDUINO IDE SOFTWARE

```
#include <LiquidCrystal.h>
#include <Servo.h>
int sensorPin = A0; // input pin for the potentiometer
int digitalValue = 0; // variable to store the value coming from the sensor
#define trigger 10
#define echo 11
#define motor0 0
#define motor1 1
#define motor2 8
#define buzzer 12
Servo myservo; // create servo object to control a servo
int potpin = 0; // analog pin used to connect the potentiometer
int val; // variable to read the value from the analog pin
LiquidCrystal lcd(7,6,5,4,3,2);
float t=0,d=0;
int temp=0;
int pos = 0;
void setup()
{
  int sensorPin=0;
  myservo.attach(9);
  lcd.begin(16,2);
  pinMode(trigger,OUTPUT);
  pinMode(echo,INPUT);
  pinMode(motor1, OUTPUT);
  pinMode(motor0,OUTPUT);
  pinMode(motor2,OUTPUT);
  pinMode(buzzer, OUTPUT);
  lcd.print(" Water Level ");
  lcd.setCursor(0,1);
  lcd.print(" Indicator ");
  delay(1000);
}
void loop()
{
  int sensorvalue = analogRead(sensorPin); // read the value from the analog channel
  Serial.print("soillevel = ");
  Serial.println(sensorvalue); //print digital value on serial monitor
  delay(1000);
  lcd.clear();
  digitalWrite(trigger,LOW);
  delayMicroseconds(2);
```

```

digitalWrite(trigger,HIGH);
delayMicroseconds(10);
digitalWrite(trigger,LOW);
delayMicroseconds(2);
t=pulseIn(echo,HIGH);
d=t*340/20000;
lcd.clear();
lcd.print("Water Space In ");
lcd.setCursor(0,1);
lcd.print("Tank is: ");
lcd.print(d);
lcd.print("Cm");
delay(1000);
int pos;
for (pos = 0; pos <= 180; pos += 1) // goes from 0 degrees to 180 degrees
  // in steps of 1 degree
{
  myservo.write(pos);          // tell servo to go to position in variable 'pos'
  delay(15);                   // waits 15ms for the servo to reach the position
}
for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
  myservo.write(pos);          // tell servo to go to position in variable 'pos'
  delay(15);                   // waits 15ms for the servo to reach the position
}
if(d<10 && temp==0)
{
  digitalWrite(motor0, LOW);
  digitalWrite(motor1,LOW);
  delay(1000);
  digitalWrite(buzzer, HIGH);
  lcd.clear();
  lcd.print("Water Tank Full ");
  lcd.setCursor(0,1);
  lcd.print("Motor Turned OFF");
  delay(1000);
  digitalWrite(buzzer, LOW);
  delay(1000);
  temp=1;
}

else if(d>10)
{
  digitalWrite(motor1,HIGH);
  digitalWrite(motor0,LOW);
  lcd.clear();
  lcd.print("LOW Water Level");
}

```

```

    lcd.setCursor(0,1);
    lcd.print("Motor Turned ON");
    delay(1000);
    temp=0;
    lcd.print("soillevel");
    lcd.print(sensorvalue);
    if(sensorvalue<50)
    {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("soil is dry");
        digitalWrite(motor2,HIGH);
    }
}
}
}

```

PROGRAM FOR HARDWARE

```

#include <SoftwareSerial.h>

SoftwareSerial mySerial(16, 17);

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

#include <Servo.h>

Servo myservo;

const int trigPin = 8;

const int echoPin = 7;

int i=0;

int pos = 0;

void setup() {

```

```
Serial.begin(115200);

mySerial.begin(115200);

myservo.attach(4);

pinMode(5,OUTPUT);

pinMode(6,OUTPUT);

pinMode(2,OUTPUT);

pinMode(3,OUTPUT);

pinMode(9,INPUT);

pinMode(10,INPUT);

digitalWrite(5,HIGH);

digitalWrite(6,HIGH);

digitalWrite(2,LOW);

digitalWrite(3,LOW);

delay(1000);

digitalWrite(5,LOW);

digitalWrite(6,LOW);

digitalWrite(2,HIGH);

digitalWrite(3,HIGH);

pinMode(trigPin,OUTPUT);

pinMode(echoPin,INPUT);

lcd.begin();
```

```

lcd.backlight();

lcd.setCursor(0,0);

lcd.print("Universal");

lcd.setCursor(0,1);

lcd.print("MOTOR");

for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree

    myservo.write(pos);          // tell servo to go to position in variable 'pos'

    delay(15);                  // waits 15 ms for the servo to reach the position
}

for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees

    myservo.write(pos);          // tell servo to go to position in variable 'pos'

    delay(15);                  // waits 15 ms for the servo to reach the position
}
}

void loop() {

    int moist = analogRead(A0);

    moist = map(moist,0,1023,1023,0);

    int pump1 = digitalRead(9);

    int pump2 = digitalRead(10);

```

```

if (moist < 50)

{

digitalWrite(5,HIGH);

for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees

    // in steps of 1 degree

    myservo.write(pos);          // tell servo to go to position in variable 'pos'

    delay(15);                  // waits 15 ms for the servo to reach the position

}

for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees

    myservo.write(pos);          // tell servo to go to position in variable 'pos'

    delay(15);                  // waits 15 ms for the servo to reach the position

}

}

else if (moist > 100)

{

digitalWrite(5,LOW);

}

if (pump1)

{

digitalWrite(2,LOW);

}

```

```

else

{
    digitalWrite(2,HIGH);
}

if (pump2)

{
    digitalWrite(3,LOW);
}

else

{
    digitalWrite(3,HIGH);
}

long duration, cm, cm1;

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

cm = microsecondsToCentimeters(duration);

if (cm > 50)

```

```

{
    digitalWrite(6,HIGH);

    for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees

        // in steps of 1 degree

        myservo.write(pos);          // tell servo to go to position in variable 'pos'

        delay(15);                  // waits 15 ms for the servo to reach the position
    }

    for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees

        myservo.write(pos);          // tell servo to go to position in variable 'pos'

        delay(15);                  // waits 15 ms for the servo to reach the position
    }
}

else if (cm < 10)
{
    digitalWrite(6,LOW);
}

lcd.clear();

lcd.setCursor(0,0);lcd.print("SM:");lcd.setCursor(3,0);lcd.print(moist);

lcd.setCursor(10,0);lcd.print("L:");lcd.setCursor(12,0);lcd.print(cm);

lcd.setCursor(0,1);lcd.print("P1:");lcd.setCursor(3,1);lcd.print(pump1);

lcd.setCursor(7,1);lcd.print("P2:");lcd.setCursor(10,1);lcd.print(pump2);

```

```

mySerial.print(cm);mySerial.print(",");

mySerial.print(moist);mySerial.println(",");

delay(5000);

}

long microsecondsToCentimeters(long microseconds) {

    return microseconds / 29 / 2;

}

```

UBIDOTS PROGRAM

```

#include <SoftwareSerial.h>
SoftwareSerial mySerial(14, 12);
#include "Ubidots.h"
const char* UBIDOTS_TOKEN = "BBFF-zGUrAf0VrLMJDhTal2etBzZlxT4k4I"; // Put
here your Ubidots TOKEN
const char* WIFI_SSID = "universal";    // Put here your Wi-Fi SSID
const char* WIFI_PASS = "12345678";    // Put here your Wi-Fi password
const char* DEVICE_LABEL_TO_RETRIEVE_VALUES_FROM = "58bf25db7fec"; //
Replace with your device label
const char* VARIABLE_LABEL_TO_RETRIEVE_VALUES_FROM1 = "pump1";
const char* VARIABLE_LABEL_TO_RETRIEVE_VALUES_FROM2 = "pump2";
Ubidots ubidots(UBIDOTS_TOKEN, UBI_HTTP);
int firstVal, secondVal, thirdVal, fourthVal,fifthVal;
int j=0;
int b=0;
void setup() {
    Serial.begin(115200);
    mySerial.begin(115200);
    ubidots.wifiConnect(WIFI_SSID, WIFI_PASS);
    pinMode(D0,OUTPUT);
    pinMode(D1,OUTPUT);
    digitalWrite(D0,LOW);
    digitalWrite(D1,LOW);

}

void loop() {
    if (mySerial.available() > 0)
    {

```

```

    int value1 = ubidots.get(DEVICE_LABEL_TO_RETRIEVE_VALUES_FROM,
    VARIABLE_LABEL_TO_RETRIEVE_VALUES_FROM1);
    int value2 = ubidots.get(DEVICE_LABEL_TO_RETRIEVE_VALUES_FROM,
    VARIABLE_LABEL_TO_RETRIEVE_VALUES_FROM2);

    // Evaluates the results obtained
    if (value1 != ERROR_VALUE || value2 != ERROR_VALUE) {
        Serial.print("Value1:");Serial.print(value1);
        Serial.print("\t Value2:");Serial.println(value2);
        digitalWrite(D0,value1);
        digitalWrite(D1,value2);
    }
    String a = mySerial.readString();
    Serial.println(a);

    for (int i = 0; i < a.length(); i++)
    {
        if (a.substring(i, i+1) == ",")
        {
            j++;
            if (j==1)
            {
                firstVal = a.substring(0, i).toInt();
                Serial.print("first : ");
                Serial.println(firstVal);
                ubidots.add("Water level", firstVal);
                b=i;
            }
            else if (j==2)
            {
                secondVal = a.substring(b+1, i).toInt();
                Serial.print("Second : ");
                Serial.println(secondVal);
                ubidots.add("Soil Moisture", secondVal);
                b=0;
                j=0;
            }
        }
    }

    bool bufferSent = false;
    bufferSent = ubidots.send(); // Will send data to a device label that matches the
device Id

    if (bufferSent) {
        Serial.println("Values sent by the device");    } }}

```