# RV College of Engineering

## Experiential Learning Report

## Project-Based Learning

**2024-25**

## Title of the Project

## Scalable Agri-Assistance System

## Students

| USN | Name |
|---|---|
| RVCE24BCS017 | Pramath J |
| RVCE24BCS054 | Samaraditya PH |
| RVCE24BME055 | Tejas N Naik |
| RVCE24BAS060 | Aditya Mallanagouda Yargal |

## Mentor

| Name, Designation & Department |
|---|
| Dr Bhuvaneswara Babu T<br>Physics Department |

## 1. Introduction

- Our project focuses on developing a Scalable Agri-Assistance system that uses sensors and real-time (by leveraging sensors) data to optimize irrigation and **minimize water wastage**.
- This is crucial in addressing challenges like **water scarcity, soil degradation/erosion, and inefficiency in resource management**, especially for small-holder farmers.
- Inefficient irrigation wastes 40% of water and pesticide overuse contaminates 30% of groundwater. This project offers a cost-effective solution to **optimize water use** and **prevent soil degradation**, crucial for sustainable farming.
- Over-irrigation and excessive pesticide use not only waste these valuable resources but also degrade soil health, leading to ***lower yields*** and ***higher long-term costs.***

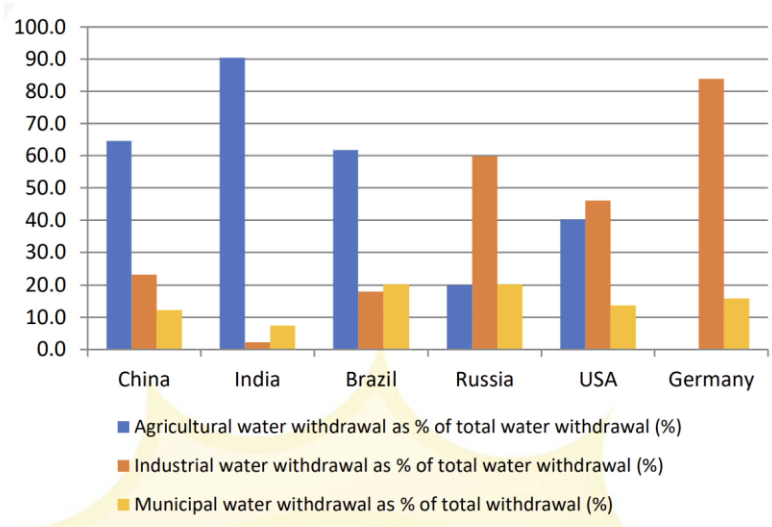
## 2. Problem Definition

- In India, the overuse of water for irrigation is a persistent issue, primarily due to the lack of technological involvement in farming.
- Most farmers rely on traditional methods, such as **flooding entire fields**, which results in significant water wastage.
- These practices are often based on **"intuition"** rather than real-time data, leading to excessive irrigation even when crops may not require it.
- This not only depletes precious water resources but also contributes to **waterlogging**, **reduced soil fertility**, and **inefficient energy usage** for water pumps.
- *With groundwater levels depleting at an alarming rate in many regions, the need for technological intervention in irrigation practices has become critical to ensure

sustainable water management and improve agricultural productivity.*

## Relevance in Real-Life

- Wheat is one of the most widely grown crops in India and it is often over-irrigated. It typically requires **4 to 6 irrigations during its growth** cycle, but farmers apply water more frequently than necessary, assuming it will increase yields.
- Over-irrigation in wheat fields across Punjab and Haryana depletes groundwater by over **1 meter per year**, threatening long-term water sustainability. These regions account for over 80% of India's wheat production, making this issue a critical concern.
- Despite China being the world's largest producer of many agricultural products, including rice, wheat; which require a lot of irrigation, the **water withdrawal for agriculture in China is significantly lesser than India which highlights the concern of over-irrigation in India.**




| Parameter | China | India |
|---|---|---|
| Agricultural water withdrawal (10^9 m3/yr) | 358 | 688 |
| Agricultural water withdrawal as % of total water withdrawal (%) | 64.61 | 90.41 |
| Total water withdrawal per capita (m3/inhab/yr) | 409.9 | 621.4 |
| Water used per Agricultural Produce in 1000M3/US$ | 0.49 | 2.27 |

- Annual precipitation in India is estimated to be around **4000 BCM**, and the usable resource potential is **1870 BCM**.

- India uses almost **700 BCM** of water for agriculture every year, whereas China uses about **361 BCM**.

- It is expected to rise to **1200 BCM** by the year 2050 if this continues.

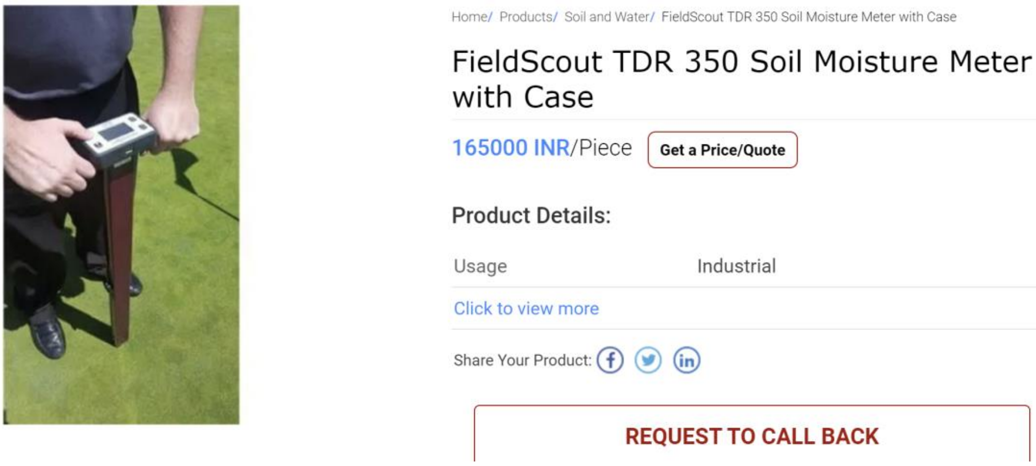
- This highlights the efficient water usage policies of China and the inefficient water usage in India.

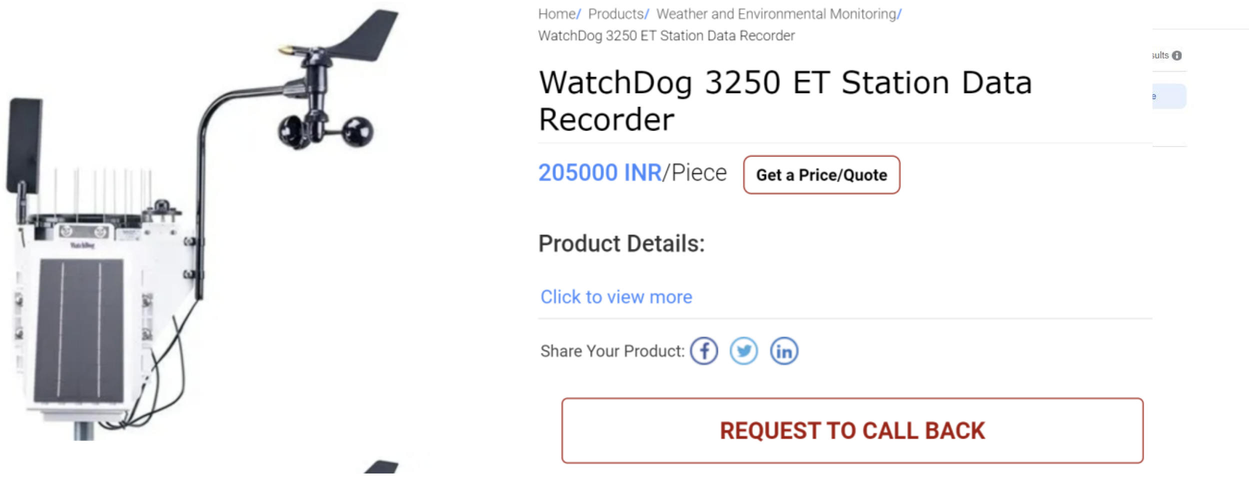
| Water Footprint in $M^3/MT$[1] | | | |
|---|---|---|---|
| **Crops** | India | | Global |
| **Wheat** | 1654 | > | 1334 |
| **Rice** | 2850 | > | 2291 |
| **Sugarcane** | 159 | | 175 |

| Virtual Water Use for Crops in $M^3/Tonne$[2] | | | | |
|---|---|---|---|---|
| **Crops** | India | | U.S. | China |
| **Rice** | 4254 | > | 1903 | 1972 |
| **Wheat** | 1654 | > | 849 | 690 |
| **Corn** | 1937 | > | 489 | 801 |
| **Soya beans** | 4124 | > | 1869 | 2617 |
| **Sugarcane** | 159 | > | 103 | 117 |
| **Cottonseed** | 8264 | > | 2535 | 1419 |
| **Roast coffee** | 14500 | > | 5790 | 7488 |

## Previous Attempts to Solve

- Western countries have addressed this issue using **expensive**, **high-tech** equipment, but these solutions are not **practical** or **affordable** for Indian farmers.
- Indian farmers need cost-effective, easy-to-use technologies that fit their unique challenges and **limited resources**.



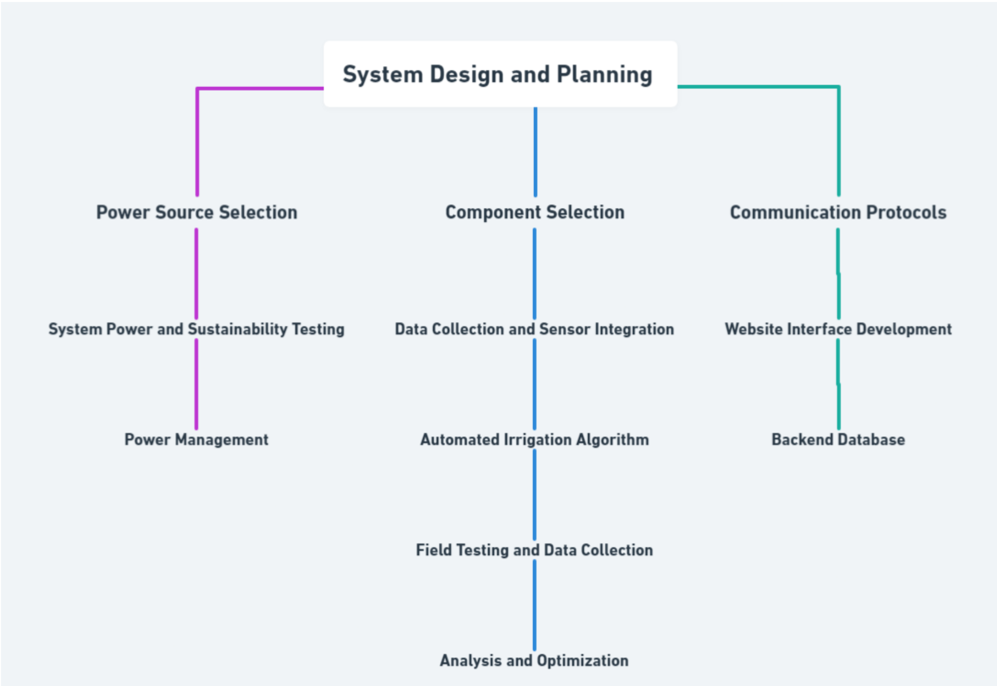



# 3. Objectives

- **Main Objective**: To Save water by using only the required amount of water.

- **Specific Objectives**:
    - **Cost-Effective Solutions**: Develop a smart irrigation system that utilizes affordable components and solar power, optimizing water usage while minimizing costs for scalable agricultural applications.
    - **Enhance Water Efficiency**: Implement an irrigation system that optimizes water usage based on soil moisture levels, reducing waste and conserving resources.
    - **Remote Monitoring and Control**: Create a user-friendly web interface that allows users to monitor soil moisture levels and control irrigation remotely via the ESP32.
    - **Sustainability**: Promote sustainable farming practices by minimizing water consumption and reliance on external power sources, thereby supporting eco-friendly agriculture.

# 4. Methodology




## Timeline

**Week (1 to 2) - Phase 1: Planning and Design**

- Finalize project **concept** and **objectives**.
- Design system architecture, sensor selection, and microcontroller setup.
- Research on **communication protocols**, **power supply options**, and **battery life**.

**Week (3 to 4) - Phase 1: Prototype Development**

- Begin initial assembly of hardware components.
- Integrate **sensors** and **microcontrollers** for basic functionality.
- Set up initial communication protocol and basic data transmission.

**Week (5 to 6) - Phase 2: Prototype Testing and Demonstration**

- Test **real-time data transmission** and control of irrigation **via relay**.
- Demonstrate the working of the prototype, focusing on **scalability** and **connectivity**.
- Troubleshoot and optimize the system for better **performance** and **reliability**.

**Week (7 to 8) - Phase 2: Scalability and Portability Testing**

- Finalize testing of the prototype for connectivity between modules and scalability.
- Demonstrate the ability to control water flow via remote nodes.

**Week (9 to 10) - Phase 3: Field Testing and Refinements**

- Begin extensive field testing with soil moisture sensors and irrigation control.
- Implement solar panels to make the modules self-sustaining.
- Design custom 3D-printed casings for modules to protect components.

**Week (11 to 12) - Phase 3: Final Testing and Adjustments**

- Conduct final tests to evaluate system performance in various conditions.
- Fine-tune the user interface and system outputs for usability.
- Make final design improvements based on feedback from field testing.
- Prepare final documentation and presentation for project review.

- **System Design and Planning**
    - **Objective:** To develop a low-cost, automated irrigation system using the ESP32 microcontroller, soil moisture sensors, and a website interface.
    - **Power Source Selection:** Design a **solar-powered system with battery backup** to ensure continuous operation. Solar panel and battery specifications chosen based on power requirements for the ESP32, sensors, and potential expansion to additional nodes.
    - **Component Selection:**
        - **ESP32 Microcontroller**: Selected for its low cost, Wi-Fi capability, and ease of integration with sensors.
        - **Soil Moisture Sensors**: Measure real-time soil moisture to control irrigation.
        - **Relay Module**: Interface with the irrigation pump to enable or disable water flow based on soil moisture data.
- **Communication Protocols:** To determine protocols for ESP32 communication with the website to send real-time data and receive manual control inputs.
- **Data Collection and Sensor Integration**
    - **Moisture Sensor Calibration**: Test and calibrate moisture sensors in various soil types and moisture levels to ensure accuracy.
    - **ESP32 Programming**: Code the ESP32 to:

    - Continuously monitor moisture sensor data.
    - Transmit data to a remote database for real-time monitoring on the website.
    - Activate the relay to trigger irrigation when moisture levels fall below a set threshold.
  - **Data Transmission**: Configure the ESP32 to transmit sensor readings to the cloud via Wi-Fi, ensuring reliable connectivity to the website.

- **Website Interface Development:**
  - **Real-Time Monitoring Dashboard**:
    - Display current moisture levels, system status, and irrigation history.
    - Use a simple, user-friendly interface optimized for desktop and mobile viewing.
    - Add controls for users to activate or deactivate the irrigation system manually, overriding automated responses if necessary.
- **Backend Database**: Store real-time and historical sensor data for trend analysis and potential future expansion to predictive irrigation models.
- **Automated Irrigation Algorithm**
  - **Threshold-Based Control**: Set threshold moisture levels that, when crossed, automatically activate irrigation.
  - **Testing and Optimization**:

- Adjust thresholds based on field conditions, crop type, and soil properties to ensure optimal water usage.
- Fine-tune the timing and duration of irrigation to prevent waterlogging or excessive dryness.

➢ **System Power and Sustainability Testing**

- **Solar and Battery Optimization**: Size the solar panel and battery based on ESP32 and sensor energy consumption to ensure continuous operation, even during low-sunlight days.

➢ **Field Testing and Data Collection**

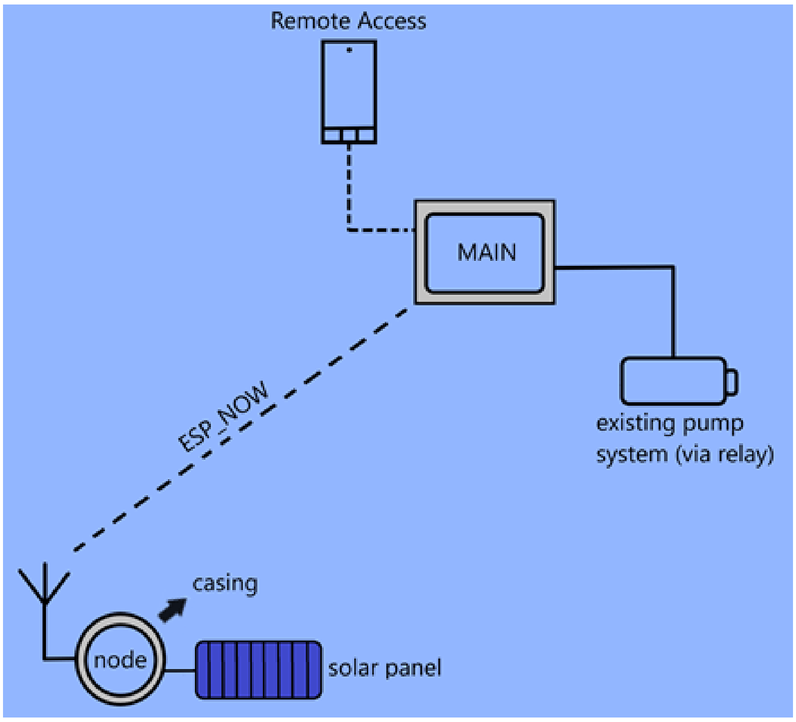
- **Initial Testing** : Test the system in controlled conditions to validate moisture readings, irrigation accuracy, and website communication.
- **Real-World Deployment** : Install the system in a field environment to test response accuracy under varying moisture levels, network stability, and battery performance.

# 5. Project Execution

## 5.1 Planning and Design

The project began with an extensive planning and brainstorming phase to identify the specific challenges faced by Indian farmers, particularly the **issue of water overuse** due **to lack of technology in irrigation practices**. The team discussed potential solutions and decided to design a solar-powered agricultural assistance system using IoT-enabled devices.
Key activities during this phase included:

- Brainstorming sessions to finalize the project scope and objectives.
- **Sketching initial design drafts** for the sensor nodes, main module, and energy management system.
- **Choosing components** like soil moisture sensors, ESP32 microcontrollers, HW-736 charging modules, XL6009 step-up boosters, and AMS1117 voltage regulators based on functionality, affordability, and availability.
- **Developing a communication plan** using the ESP-NOW protocol to enable efficient data transfer between nodes and the main module.
- **Preparing a project timeline** to ensure the systematic execution of tasks.



## 5.2 Implementation

The implementation phase involved building and testing the prototype, starting with smaller modules and gradually integrating them into the complete system.
Key steps included:

1. **Soil Moisture Sensor Deployment**:
   - Sensors were calibrated to provide accurate readings under various soil conditions.
   - Four sensor nodes equipped with ESP32 modules were designed for field data collection.
2. **Energy Management System**:
   - Solar panels were connected to HW-736 modules to charge two 3.7V Li-ion batteries in parallel.
   - XL6009 step-up boosters and AMS1117 voltage regulators were used to maintain stable voltage for powering the ESP32 modules.
3. **Prototype Development**:
   - Sensor nodes and the main module were assembled and tested independently before integration.
   - The main module, equipped with a SIM-enabled ESP32, was configured for remote data transmission to a mobile device.
4. **Software Development**:
   - The ESP32 firmware was programmed using C to handle soil moisture data collection, transmission via ESP-NOW, and motor control.
   - A mobile interface was designed to allow remote monitoring and control of irrigation systems.
5. **System Testing and Debugging**:
   - Various scenarios were tested, including different soil moisture levels, battery performance under varying solar input, and data transmission across distances.
   - Challenges, such as voltage mismatches and power management, were resolved through iterative debugging and component replacement.

# 6. Tools and Techniques Used

## Hardware

- **18650** Battery (3.7v) [power source]
- **HW736** [solar charger for Li-Ion batteries]
- **Esp32** [microcontroller]
- **AMS1117** (3.3v) [Voltage regulator]
- Capacitive moisture sensor ( with 662k voltage regulator and **TLC555C** OR **TLC555I** chip )
- **XL6009** [Buck booster upto 30v]
- **Veroboard** [Soldering Board]
- **Soldering Iron**
- **Soldering flux** [to remove oxidation layer]
- 2000-count **digital multimeter** [voltmeter, ammeter]
- Solid core wires [forming connections on veroboard]
- 6V **polycrystalline** solar panel [charging power source for battery]
- *Bread board,Motor driver (replicating 240v relay in existing system), Water pump and Plastic pipe*-*for demonstration only*

## Software

- **Arduino IDE**
- ESP32 Windows Driver [ **CP210x** ]
- ESP32 Extension [ **Espressif** ]
- **VSC**
- **Canva** [for diagram]
- **SolidWorks2020** [3d encasing design]
- **UltiMaker Cura** [3D printing]
- **KiCAD** [circuit design]

| ESP32 | ARDUINO and other microcontrollers around same price |
|---|---|
| **Built-In Connectivity** - Comes with built-in Wi-Fi and Bluetooth, allowing seamless wireless communication without the need for additional modules | Most other microcontrollers (e.g., Arduino Uno, STM32, ATmega series) do not have built-in wireless capabilities, requiring external Wi-Fi or Bluetooth modules. |
| **Power Efficiency and Advanced Sleep Modes** - Equipped with various low-power/deep sleep modes, that significantly reduce power consumption, making it ideal for | While some microcontrollers support low-power modes, the ESP32's deep sleep functionality is more advanced and allows greater flexibility. |

| | |
|---|---|
| **battery** and **solar-powered applications** | |
| **Cost-Effectiveness for Projects** - ESP32 is very **affordable**, making it suitable for large scale deployments and cost-sensitive applications. | Some microcontrollers, especially those requiring additional connectivity modules, can become more **expensive** and **less practical** for low-cost IoT solutions. |
| **High I/O Count and Versatile Sensor Compatibility** - Has a high number of GPIO pins and supports multiple communication protocols (e.g., I2C, SPI, ADC, PWM), making it easy to connect a wide range of sensors and peripherals**.** | Alternatives may have fewer I/O pins and limited support for certain interfaces, restricting the variety of sensors and devices that can be used in the system. |
| **Scalability and Compatibility with IoT Protocols -** Supports a range of IoT protocols, including **MQTT** and **HTTP**, allowing easy integration with cloud platforms and mobile applications. | Limited protocol support may restrict the ability to easily integrate with cloud-based or IoT systems, hindering scalability and real-time data sharing. |

## Capacitive Moisture Sensor

**Durability**: Capacitive sensors resist corrosion and withstand varying soil conditions, ensuring reliable long-term use in agriculture.

- **Accuracy**: They provide stable and precise soil moisture readings, enabling efficient irrigation and resource management.
- **Cost-Effectiveness**: While initially pricier than resistive sensors, their durability and accuracy reduce replacement costs, offering better long-term value.

**Reliability**: maintain consistent performance with minimal maintenance, ideal for long-term applications.

- **Resistive Sensors**: Prone to corrosion, requiring frequent calibration and maintenance, making them unsuitable for long-term agricultural use despite low cost.
- **Gypsum Blocks**: Degrade over time, have slow response rates, and are sensitive to soil salinity, limiting their reliability and practicality in diverse agricultural conditions.
- **HH2 Moisture Meter**: Highly accurate but expensive, bulky, and complex, making it impractical for small-holder farmers seeking cost-effective, user-friendly solutions.

# 7. Results and Discussion

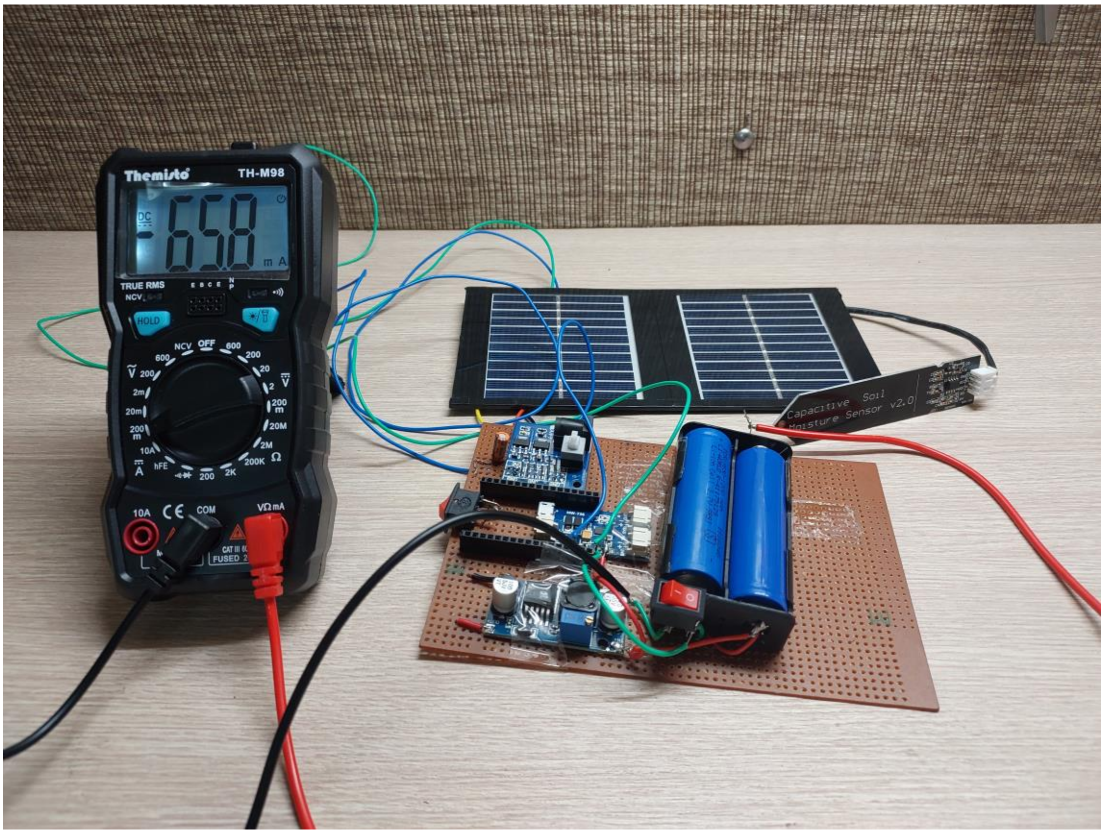
## Sustainability Testing (1)

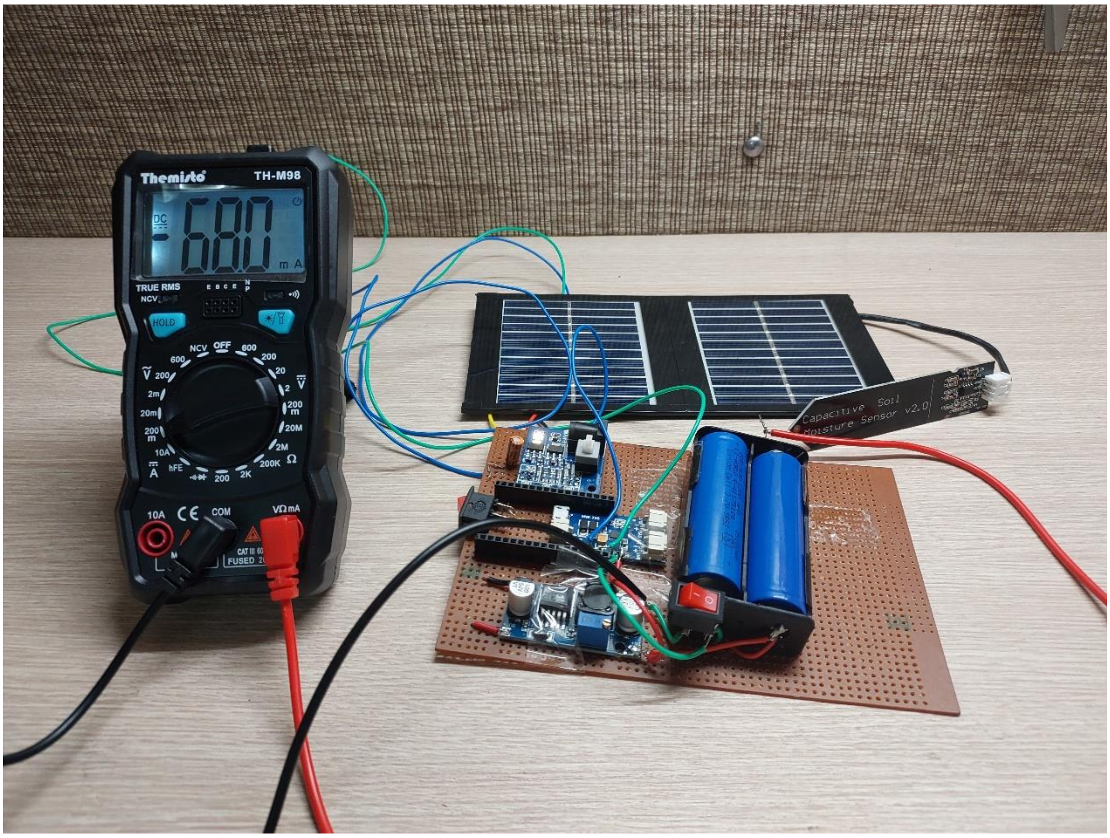
- **The system was tested for sustainability by connecting a multimeter in Ammeter mode in series with the circuit.**
- A **3.7v Li-ion** Battery when fully charge has a terminal potential of **4.2V**
- So a **1200mAh** battery of 3.7v has **(1200 x 4.2)/1000** = **5.04 WattHours** of energy

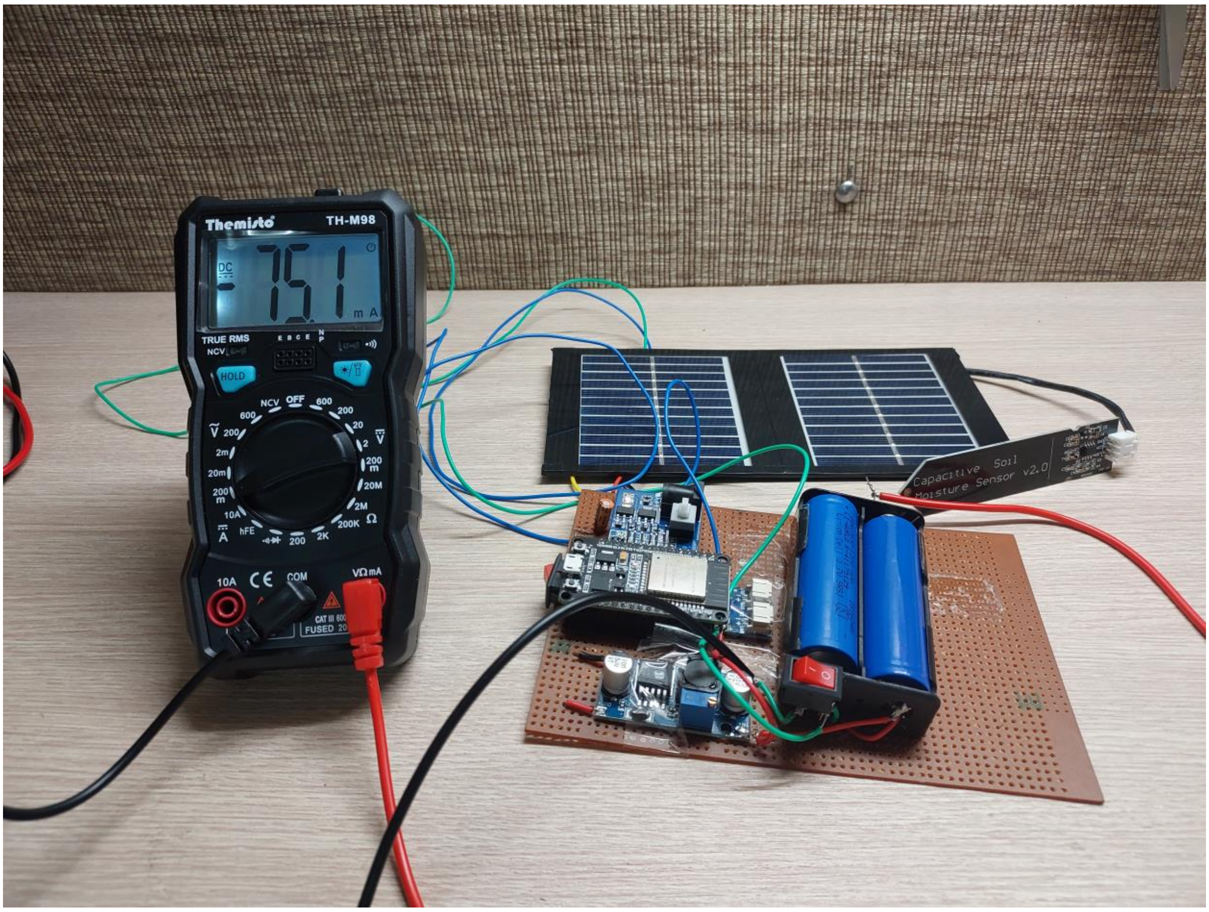
| Test Case | Power draw (V x I) |
|---|---|
| Only **XL6009** (buck booster) | **243.46mW** |
| **XL6009+AMS1117** (3.3v regulator | **251.6mW** |
| **XL6009+AMS1117+ESP32 (WIFI ON)** | **278.24mW** |

- Here, **two 1200mAh** batteries are used in parallel combination to double the energy storage capacity, making the total energy stored at full charge approximately **~10 WattHours**
- Hence, the System will run on max power draw mode (**278.24mW**) for (**10/278.24mW) = ~ 36 Hours**
- The System can last for **1.5 Days** without any power source for charging which ***replicates a situation where sunlight is not bright/intense enough to fully charge the battery as planned.***

## *Images:*


Themisto
TH-M98
658
Capacitive Soil Moisture Sensor v2.0


Themisto
TH-M98
680
Capacitive Soil Moisture Sensor v2.0




## Sustainability Testing (2)

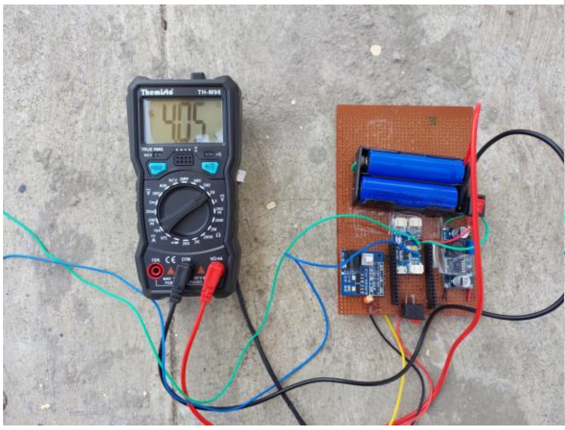
- **The system was tested for sustainability by connecting a multimeter in Voltmeter mode in Parallel config with the circuit.**
- The System's ability of **self sustainability** was tested by replicating real world conditions for solar charging of Battery.
- It was found that the *medium level bright sunlight* (sunlight through clouds) was more than enough to charge the battery which was observed as an increase in the voltage of the battery when the **system was in max power draw mode**.

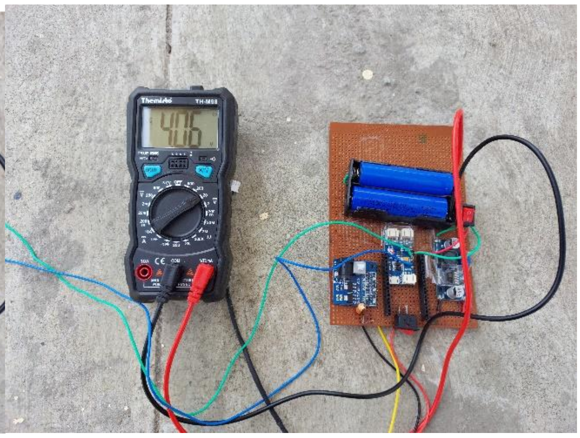
- It was also found that the intensity of sunlight from **7am to 5pm** was enough to charge the batteries.
- The charging was tested with an initial battery voltage of **4.05V** which close to the max voltage of the battery and charging it when the capacity is closer to the full capacity is harder and requires more power**, simulating the "worst case scenario".**

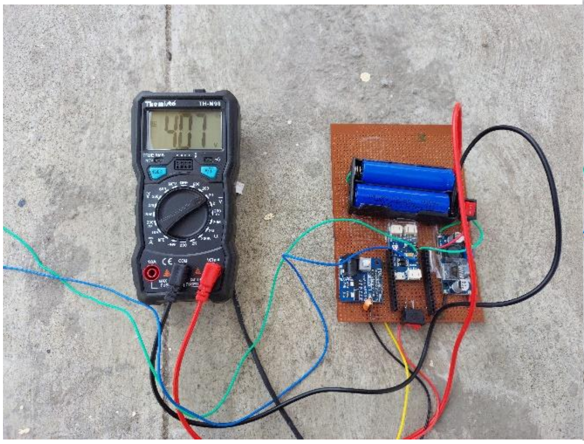
| Time | Voltage (V) |
|---|---|
| 12:25PM | 4.05 |
| 12:30PM | 4.06 |
| 12:34PM | 4.07 |
| 12:38PM | 4.08 |
| 12:42PM | 4.09 |
| 12:46PM | 4.10 |
| 12:55PM | 4.12 |
| 1:00PM | 4.13 |
| 1:10PM | 4.18 |

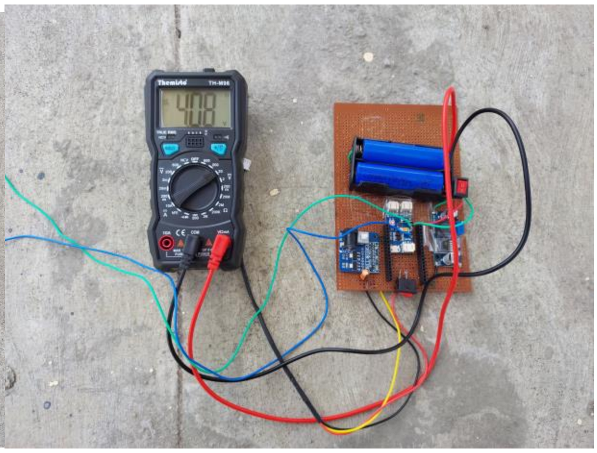
- The Two batteries were fully charged from an inital voltage of **4.05V to 4.18V (~4.2V) in less than an Hour.**
- This System can be charged under low sunlight as well and will last for 1.5 days without any sort of charging which is the worst case scenario and is unlikely to happen; as even low sunlight can charge it.

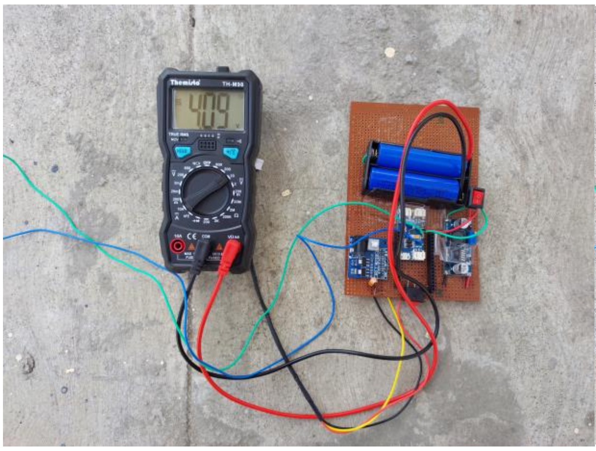
## *Images:*

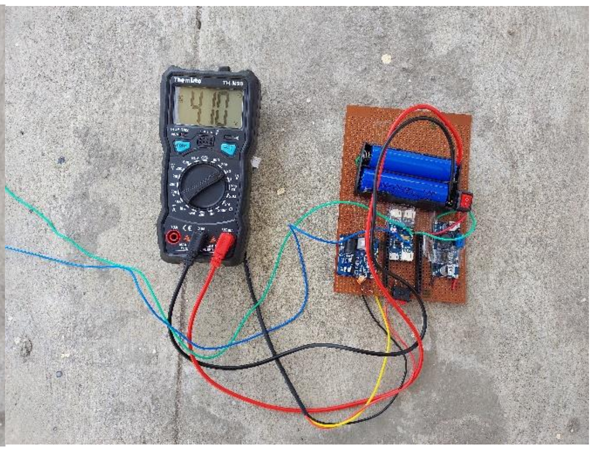


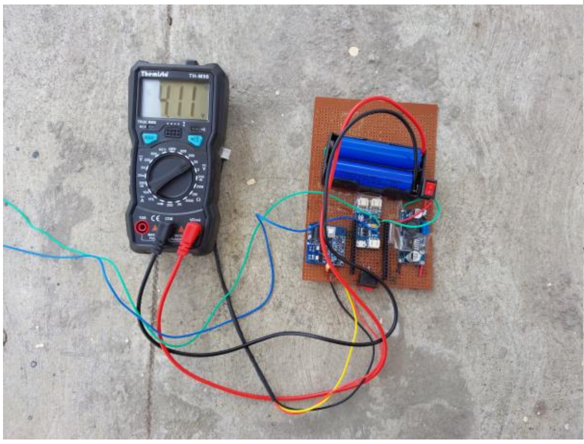
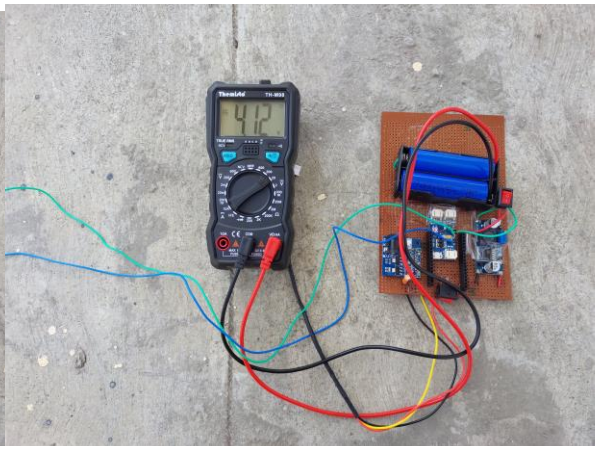
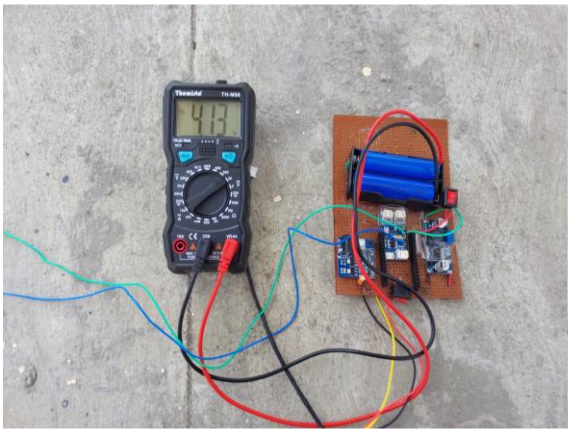
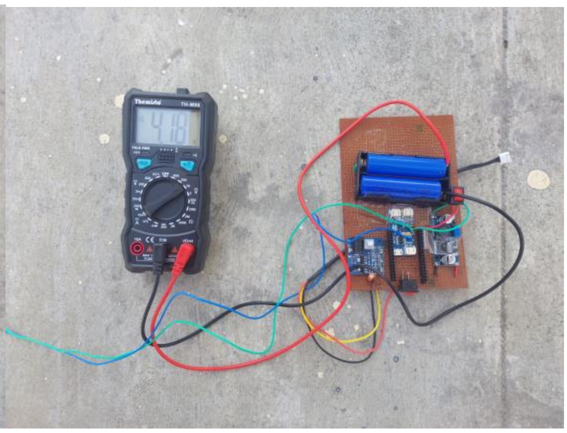
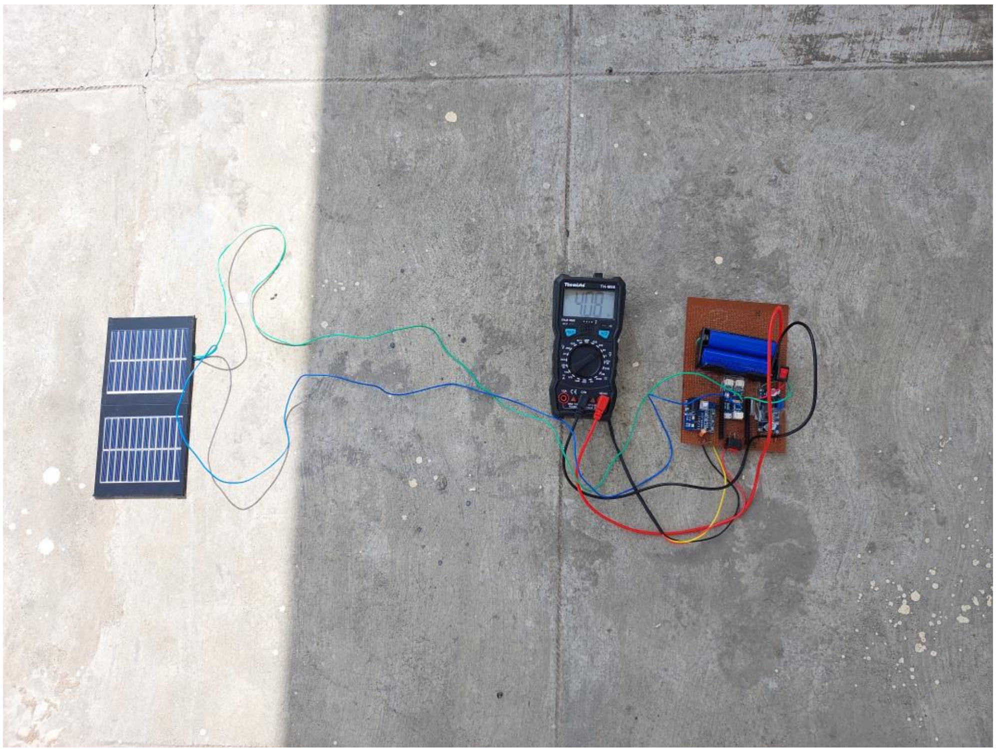










## Field Testing

- The moisture sensor gives an output of range **(1000,4000)** where **1000 is completely wet** (dipped in water) and **4000 is completely dry** (in contact with nothing).
- Optimal soil wetness was tested for different types of soils and their respective moisture sensor readings were noted down.

| Soil type | Optimal wetness reading |
| --- | --- |
| **Alluvial** | **1500** |
| **Black** | **1800** |
| **Laterite** | **1300** |

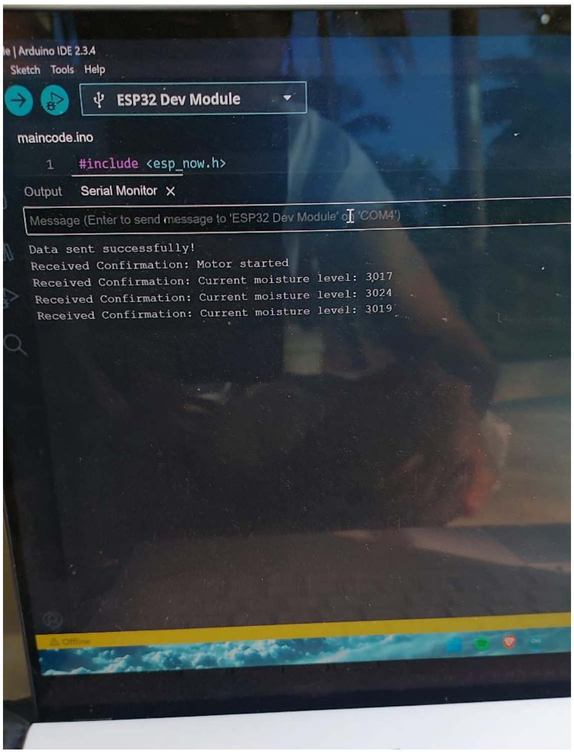
- Higher value for Black soil is due to its excellent water retention properties.
- Lower value for laterite soil is due to its bad water retention properties.

# Final Results

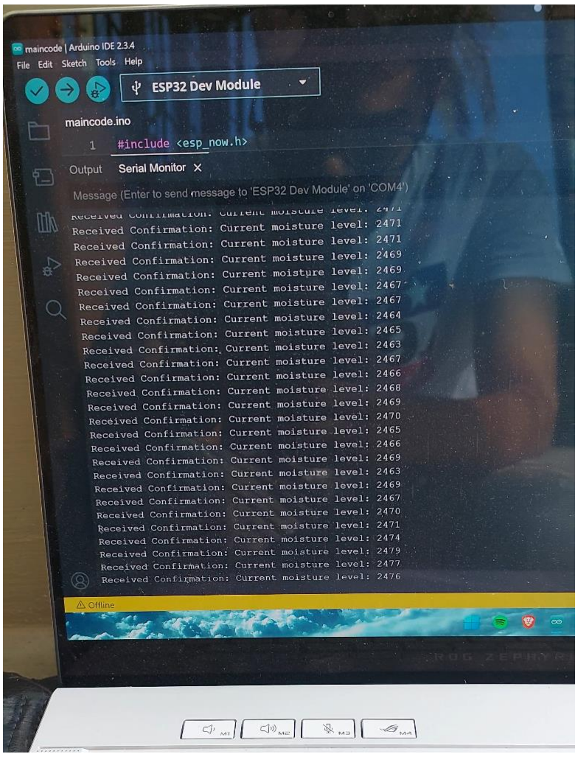
- System was turned on at **2:00PM** and the main module initiated the turning on of the motor pump for drip irrigation.
- The Nodal module was placed at the **corner of the field** and the motor pump system was at the **center of the field.**
- The field was of **2 Acres** and the distance between the main and nodal module was calculated using pythagoras theorem : **root((90)^2+(90)^2) = ~125m**
- At this distance, **ESP_NOW** works perfectly and the data was transmitted via WiFi to main module.
- System was automatically turned off at **4:00PM** and the main module initiated the turning on of the motor pump for drip irrigation.
- When farmers around the same field were surveyed it was found that they usually turn the irrigation on for for 4-5 hours at a time, no water the soil wetness status.
- In contrast to the traditional system, Automated system was only ON for 2 hours based on the situation of the field status when tested
- Based on this test result, the pump was ON for about **half** the time as the conventional system, implying the

potential to save about **half as much as water** used traditionally.

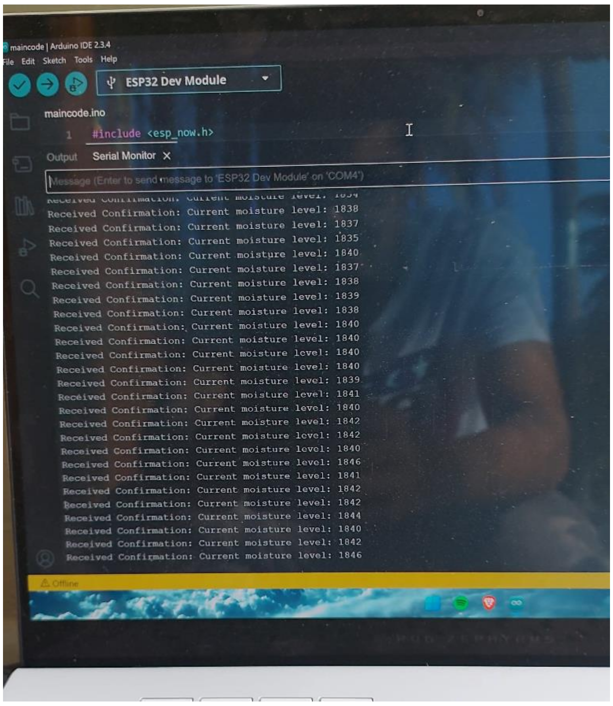
- This value may vary more or less based on the soil wetness status on a daily basis.
- For example : The field used for testing consists of **70** coconut trees in a **2 Acre** land, where each tree requires about **100L** of water per day.
- This usage can be reduced to **50-60L** by using an automated system leading to savings of around **30L(reduced number due to hot summer days) per tree** per day.
- So per year, **upto (30x70x365) = 766,000** Litres of water can be saved



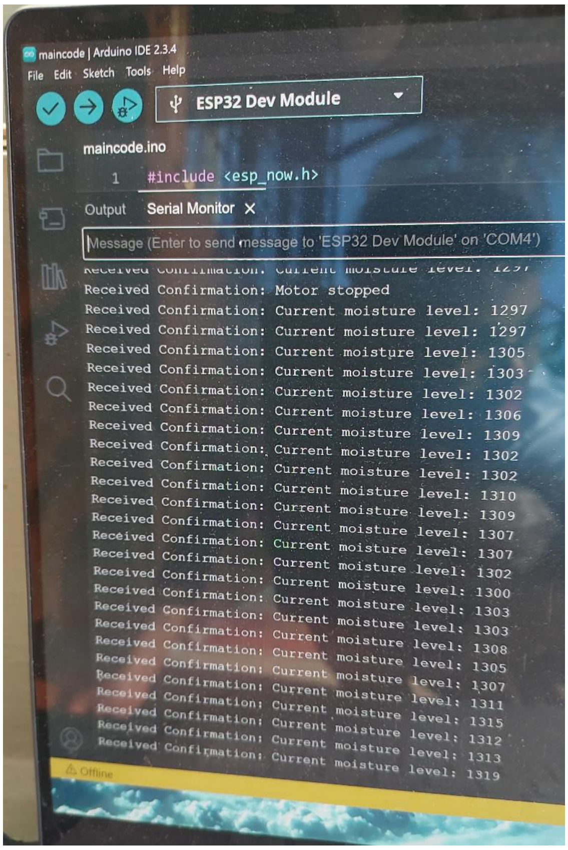

**2:00PM**




**2:45PM**



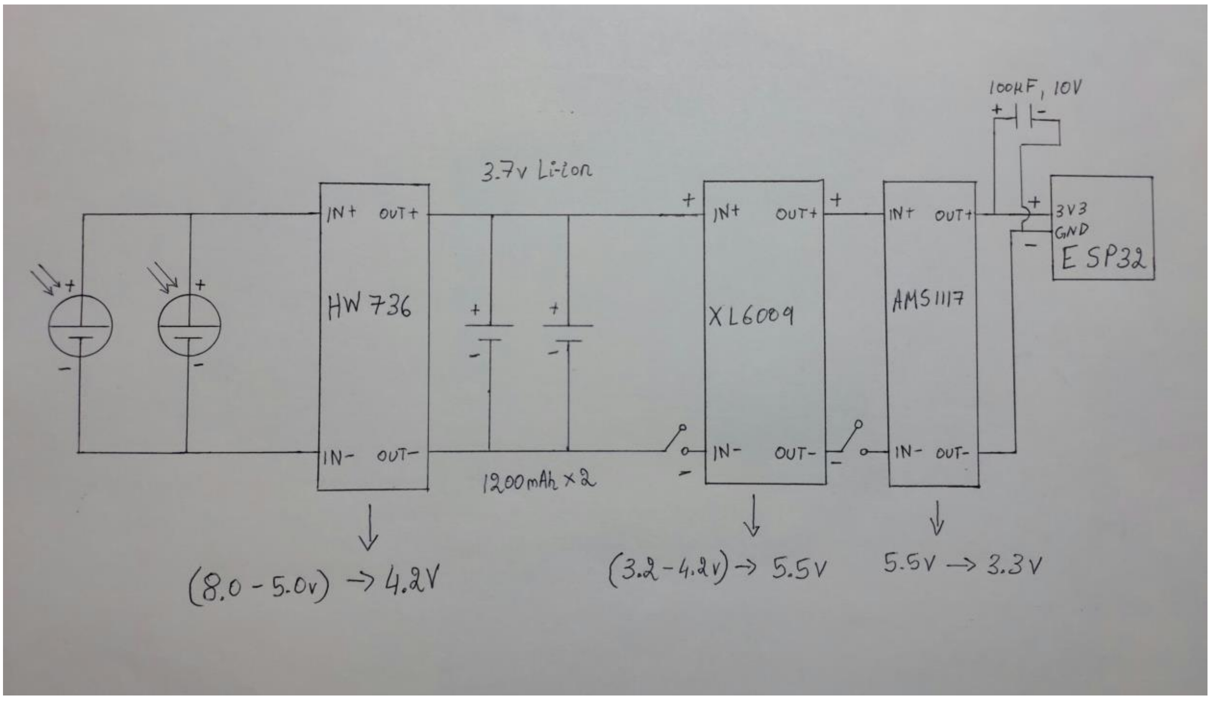

**3:30PM**

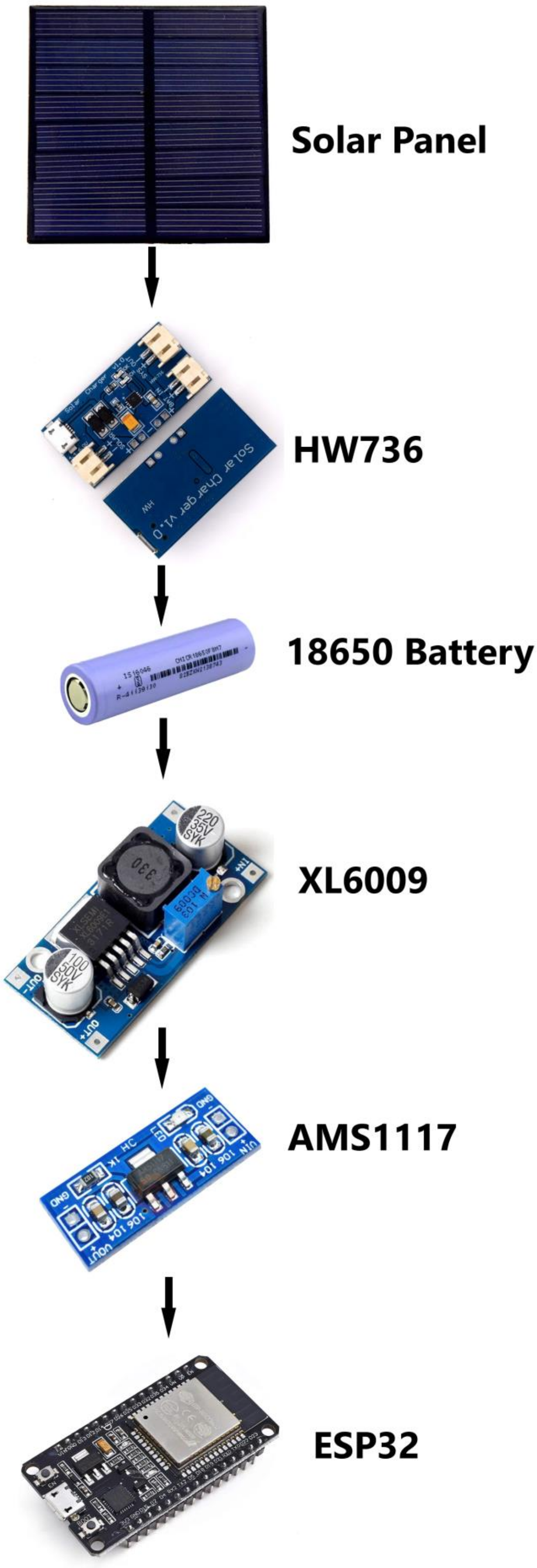



**4:00PM**

## 8. Prototype (Hardware/Software)

- **Main Module**: Designed and developed a functional prototype using ESP32 for data Transmission/Recieving.
- **Nodal Module**: Developed and integrated a node that collects soil moisture, which transmits to the main module.
- **Water Control System**: Implemented a relay-based water pump system to demonstrate
  - Irrigation control in **response to real-time data**.
  - Ability to **implement and improvise** based on any irrigation setup.




Solar Panel
HW736
18650 Battery
XL6009
AMS1117
ESP32

# Prototype Description

## Specifications:

- **Power Source**
    - **6V** solar panel connected to an **HW-736** module for charging.
    - **Dual 3.7V Li-ion batteries** connected in parallel for stable energy storage.
    - **XL6009 step-up booster** to raise voltage to **5.5V**, followed by an **AMS1117 3.3V regulator** for powering the ESP32 module.
- **Communication**:
    - **Node Modules**: ESP32 devices at soil moisture sensor locations communicate with the central ESP32 (main module) using ESP-NOW protocol.
    - **Remote Control**: Main module transmits data and allows motor control via cellular connectivity
- **Controller**:
    - The main ESP32 module is configured to control irrigation motors based on soil moisture levels and remote user inputs.

## Features and Functionality

- Real-time monitoring of soil moisture levels.
- Automatic and manual irrigation control via mobile.
- Efficient power management using solar charging and step-up/step-down voltage regulation.
- Communication between nodes and the central module using ESP-NOW for seamless data transfer.

## Development Process

- **Design and Component Selection**:
  - Components like the HW-736, XL6009, AMS1117, and Li-ion batteries were selected after extensive research on compatibility and efficiency.
  - ESP32 modules were programmed for ESP-NOW communication and integration with sensors.

- **Implementation**:
  - The solar-powered system was assembled, ensuring proper connections and protection with Schottky diodes.
  - Soil moisture sensors were calibrated for accurate readings.
  - Node and main modules were programmed to handle data collection, communication, and motor control.

- **Challenges and Solutions**:
  - **Voltage inconsistencies** from the solar panel.

    **Solution**: Implemented a voltage regulator and step-up booster for consistent output.

  - **Optimizing ESP-NOW communication** for long-range.

    **Solution** : Fine-tuned the firmware and placement of modules for better signal strength.

  - **High power draw from batteries by components**

    **Solution :** Research on similar components with same functionality and was done and components with significantly less power draw were utilized

## Testing and Validation

- **Power Management**:
  - The solar charging system was tested under various sunlight conditions to ensure consistent charging.
  - The battery runtime was monitored to confirm compatibility with the ESP32's power consumption.
- **Sensor Accuracy**:
  - Soil moisture sensors were tested in different soil types and moisture levels for calibration and reliability.
- **Communication**:
  - ESP-NOW protocol was tested for packet delivery rates and signal reliability across various distances in an open field.
- **Irrigation System**:
  - The motor control system was tested for responsiveness to both automatic and manual commands.

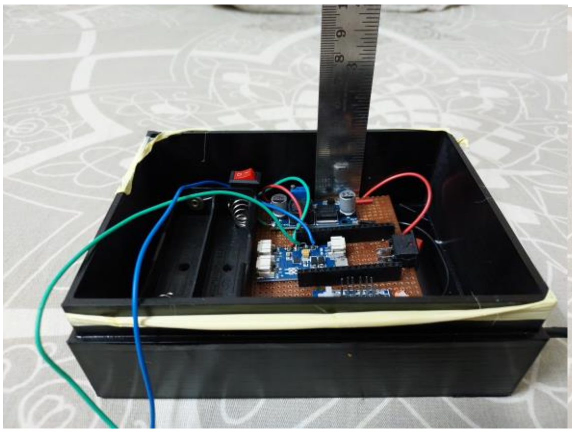
**The Final prototype was developed on a Veroboard/Prototype board with the help of solder. The whole setup was encased in a 3D printed box designed with solidworks and printed using PLA.**

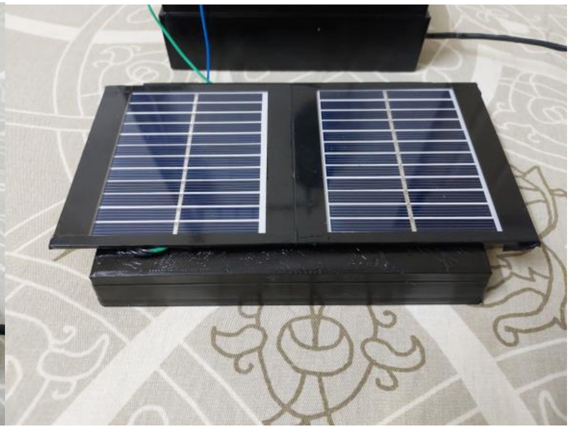
**Since PLA is unstable and not water proof, it was coated with multiple layers of black acyrlic spray paint and the lid interface was wrapped with teflon tape.**

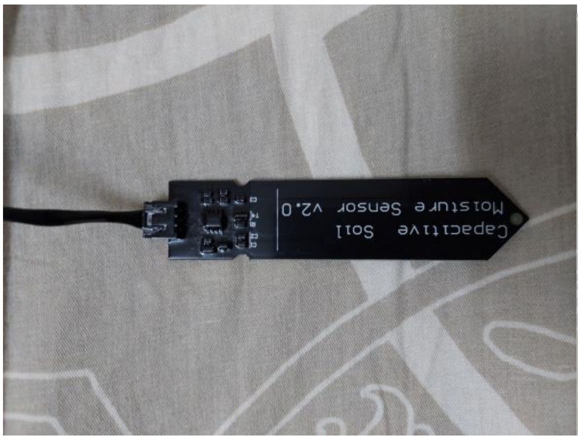
**The electronic components on the moisture sensor were made waterproof by acyrlic spray paint and the wires were covered with heatshrink.**

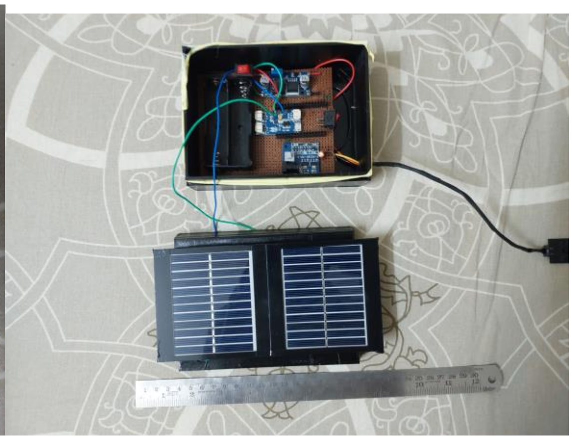
# Final Prototype









# 9. Conclusion

- The project focused on addressing the issue of **excessive water usage in Indian agriculture**, primarily caused by the **lack of accessible, technology-based solutions.**
- The main objective was to develop an **affordable**, **solar-powered agricultural assistance system** capable of monitoring soil moisture levels, improving irrigation efficiency, and providing farmers with practical insights.
- The system combined a soil moisture sensor, ESP32 modules, and a solar-powered energy management system. Using the ESP-NOW protocol, it enabled communication between sensor nodes and the main module, allowing real-time soil moisture monitoring and remote control of irrigation motors through a mobile device.
- **Testing results confirmed the potential for significant water savings through the system**. By irrigating only when the soil moisture levels dropped below the optimal threshold, unnecessary water usage was avoided, demonstrating the viability of technology-driven farming practices.
- The project has the potential to contribute substantially to sustainable farming practices, resource conservation, and improved agricultural productivity.

## Personal Reflection

As the team leader for this project, the journey has been both challenging and rewarding. Taking on the responsibility of overseeing the project's progress while ensuring that every team member's contributions were valued and aligned with our objectives was a significant learning experience.

This project introduced me to the world of IOT devices and micrcontrollers and about its working principles and mechanisms. It also put to use my logical reasoning and thinking for the coding and software part of the project and tested my skills with different problem aspects. I learnt the basics of C programming through this as the microcontroller needed to be programmed exclusively in C.

One of the most impactful aspects of this experience was learning to navigate unexpected hurdles, for example the voltage regulator for the ESP32 had a working voltage above that of the batteries we used, so I had to improvise and use a buck booster to increase the voltage and add another battery to cope with the additional power draw.

Collaborating with the team to overcome these challenges reinforced my ability to think critically and adapt under pressure.

Leading this project has given me a sense of fulfillment, knowing that our work has the potential to make a tangible difference in the lives of farmers. Beyond the technical aspects, it has strengthened my leadership, communication, and project management skills, which I will carry forward in my professional journey.

## 10. QR Code of Demonstration Video