Design and Implementation of Smart Irrigation System by using Advanced Level Sensor.



SONARGAON UNIVERSITY (SU)

Submitted By

Md. Mosarof Hossain Somon	**ID: EEE2003021051**
Md. Ikbal Hossain	**ID: EEE2003021041**
Md. Nazmul Hassan	**ID: EEE2003021084**
Md. Mashiurrrahman	**ID: EEE2003021086**
Habibur Rahman	**ID: EEE2003021042**

Supervised By

Md. Rais Uddin Mollah
Assistant Prof. and Head of the department
Department of EEE

Department of Electrical & Electronic Engineering (EEE)
Sonargaon University (SU)
147/I, Panthapath,
Dhaka-1215, Bangladesh

Date of Submission: 10th May 2024

Declaration

It is declared hereby that this thesis paper or any part of it has not been submitted to anywhere else for the award of any degree.

..... ….....

Md. Mosarof Hossain Somon **Md. Ikbal Hossain**

..... ….....

Md. Nazmul Hassan **Md. Mashiurrrahman**

.....

Habibur Rahman

Under Supervision of

.....

Md. Rais Uddin Mollah
Assistant. Prof. and Head of the department
Department of EEE
Sonargaon University (SU)

Certification

This is to certify that this project entitled " **Design and Implementation of Smart Irrigation System by using Advanced Level Sensor**" is done by the following students under my direct supervision. This project work has been carried out by them in the laboratories of the Department of Electrical and Electronic Engineering under the Faculty of Engineering, Sonargaon University (SU) in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering.

Supervisor

.....

Md. Rais Uddin Mollah
Assistant Prof. and Head of the department
Department of EEE
Sonargaon University (SU)

ACKNOWLEDGEMENT

The report titled as on **" Design and Implementation of Smart Irrigation System by using Advanced Level Sensor**" has been prepared to fulfill the requirement of our practicum program. In the process of doing and preparing our practicum report, we would like to pay our gratitude to some persons for their enormous help and vast co-operation. At first, we would like to show our gratitude to the University authority to permit us to do our practicum. Specially, we would like to thank to our honorable teacher **Md. Rais Uddin Mollah** Asst. Prof. And Head Department of Electrical & Electronics Engineering, Sonargaon University, Dhaka, for his valuable and patient advice, sympathetic assistance, co-operation, contribution of new idea. Deep theoretical and hardware knowledge & keen interest of our supervisor in this field influenced us to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project.Finally, we would like to thanks again to the respected **Vice-Chancellor of SU, Professor Dr. Shamim Ara Hassan** also thanks to **Head Department of Electrical & Electronics Engineering, Assistant Professor Md. Rais Uddin Mollah** because they are designated such an environment for learning through which we got the opportunity to acquire knowledge under Bsc in EEE program, and that will be very helpful for our prospective career. We are, indeed, grateful to all those from whom we got sincere cooperation and help for the preparation of this report.

ABSTRACT

Agriculture is one of the biggest economic sectors in our country, many farmers and labor depend upon agriculture to develop and to increase the economy of our country. Water is extensively used in modern agriculture. Effective and economic use of water can enhance yield quality and quantity, thus ensuring food security for the ever-growing population around the globe. Excessive use of water to produce crops causes the water level of the ground to go deep down as a result shortage of pure drinking water is a big threat to us. To grow 1kg of paddy requires 2500 liters of water, but farmers of Bangladesh are using 3300 liters of water to produce 1kg of paddy which means 800 liters are just wasted. In India, the numbers are much higher, almost 5500 liters. In this smart irrigation project, we utilized modern sensor and IoT technology to efficiently manage the water in agriculture. The sensor collects 7 important soil parameters and provides the data to microcontroller. Based on sensor data and the crop type microcontroller determines the amount of water required for the crop and controls the water pump accordingly. All the data can be visualized from the LCD display and as well as from the remote dashboard almost in real time. The pump can also be operated manually using smartphone or web browser.

TABLE OF CONTENTS

Topics	Page No
ACKNOWLEDGEMENT	IV
ABSTRACT	V
CHAPTER NAME	VI
LIST OF FIGURES	IX
TABLES	11

CHAPTER I: INTRODUCTION

TOPICS		Page No.
1.1	Introduction	12
1.2	Purpose of the Study	13
1.3	Objectives	14
1.4	Project Organization	14

CHAPTER II: THEORY OF THE PROJECT

2.1	Introduction	15
2.2	Theory	15
2.2.1	Arduino Uno	16
2.2.2	Microcontroller ATMEGA328P	18
2.2.3	Design of solar power for smart irrigation project	18
2.2.3.1	Solar Panel	20
2.2.4	NPK 7 in 1 soil sensor	21
2.2.5	Water Pump	23
2.2.6	LCD Display 16x4	24
2.2.7	DC Power Supply	25
2..2.8	5V Single Channel Relay	26
2.2.9	ESP Module	26
2.2.10	12V Li-po Battery	27
2.2.11	DC Buck Converter	28
2.2.12	Connecting Wire	28

2.2.13	PCB Board	29
2.2.14	Screw Connection	29

CHAPTER III: DESIGN AND FABRICATION

3.1	Introduction	30
3.2	Block Diagram of the Project	30
3.3	Circuit Description	31
3.4	Overview of smart irrigation technology	33
3.5	Working Principle	33
3.5.1	7in1 Soil Sensor Section	34
3.5.2	Control Section	35
3.5.3	Mobile Apps Section	35
3.6	Image of Project	36
3.7	Flow Chart	37
3.8	Used Materials	38
3.8.1	Hardwar Equipment	38
3.8.2	Software Equipment	38
3.9	Conclusion	38

CHAPTER IV: RESULT & DISCUSSION

4.1	Introduction	39
4.2	Result	39
4.3	Advantages of smart irrigation	40
4.4	Disadvantages of smart irrigation	40
4.5	Discussion	40
4.6	Cost Estimation	41
4.7	Application	41
4.8	Conclusion	42

CHAPTER V: CONCLUSION

5.1	Conclusion	43
5.2	Future Work	43
References		44
Appendix		44-53

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO
2.2.1	Arduino UNO Pin Diagram	16
2.2.2	Microcontroller ATMEGA32P	18
2.2.3.1	Solar system	21
2.2.4	7 in 1 Soil Sensor	22
2.2.5	water Pump	23
2.2.6.1	LCD Display	24
2.2.6.2	LCD Pin Out of LCD	24
2.2.7.1	DC Power Supply	25
2.2.7.2	Output Waveforms of 220V/12V rectifier	25
2.2.8	Single Channel Relay Module	26
2.2.9.1	ESP Wi-Fi Module	27
2.2.9.2	ESP Wi-Fi Module with UNO	27
2.2.10	Li-po Battery	28
2.2.11	DC Buck Converter	28
2.2.12	Connecting Wire / Jumper Wires	29
2.2.13	PCB Board	29
2.2.14	Screw Connection	29
3.2	Block Diagram of the Project	30
3.3.1	Circuit Diagram of the Project-1	32
3.3.2	Circuit Diagram of the Project-2	32
3.4	Smart Irrigation Technology	33
3.5	Overall Design (ESP)	34
3.5.1	Arduino UNO	34
3.5.2	Soil Sensor With UNO	34

3.6	Real Image of the Project	36
3.7	Flow Chart of the Project	37
4.2	Output Result	39

LIST OF TABLE

Table.2.1	Technical specifications of Arduino UNO	67
Table.4.1	Cost Estimation	41

CHAPTER I

INTRODUCTION

1.1 Introduction

The advent of technology has spurred innovative solutions to address pressing global challenges, and agriculture is no exception. Traditional irrigation methods, while essential for crop production, often suffer from inefficiencies that result in water wastage, increased energy consumption, and diminished crop yields. In response to these challenges, the concept of Smart Irrigation Systems (SIS) has emerged as a promising solution to optimize water usage in agriculture while enhancing overall sustainability.

Smart Irrigation Systems leverage the power of Internet of Things (IoT) technology to revolutionize conventional irrigation practices. By integrating sensors, actuators, and communication networks, SIS enables real-time monitoring and precise control of irrigation parameters such as soil moisture levels, weather conditions, and crop water requirements. This data-driven approach allows farmers to make informed decisions and implement targeted irrigation strategies, thereby maximizing water efficiency and crop productivity.

The importance of optimizing water usage in agriculture cannot be overstated, particularly in the face of increasing water scarcity and climate variability. According to the Food and Agriculture Organization (FAO), agriculture accounts for approximately 70% of global freshwater withdrawals, making efficient water management a critical priority for sustainable food production and environmental conservation.

Traditional irrigation methods, such as flood irrigation or manual watering, often lead to overwatering or underwatering, both of which can have detrimental effects on crop health and yield. Overwatering not only wastes precious water resources but also leaches nutrients from the soil and promotes the growth of pests and diseases. Conversely, underwatering can lead to stunted growth, reduced yields, and even crop failure, particularly in arid or drought-prone regions. Smart Irrigation Systems offer a holistic approach to addressing these challenges by optimizing water usage based on real-time data and predictive analytics. By deploying sensors to monitor soil moisture levels, weather conditions, and crop water requirements, SIS can precisely regulate irrigation schedules, duration, and intensity, ensuring that crops receive the right amount of water at the right time.

Furthermore, Smart Irrigation Systems offer numerous benefits beyond water conservation. By minimizing water wastage and energy consumption, SIS can help reduce operational costs and improve farm profitability. Additionally, by maintaining optimal soil moisture levels, SIS can enhance soil health and fertility, leading to healthier crops and improved yields. Moreover, by mitigating the risk of water stress and drought, SIS can enhance crop resilience to climate variability and contribute to food security and livelihoods of farming communities.

In conclusion, Smart Irrigation Systems represent a transformative approach to agricultural water management, offering a sustainable solution to optimize water usage, enhance crop productivity, and promote environmental stewardship. By harnessing the power of IoT technology, SIS has the potential to revolutionize agriculture and contribute to a more sustainable and resilient food system for future generations.

1.2 Purpose of the Study

The purpose of studying Smart Irrigation Systems (SIS) is multifaceted and encompasses various objectives aimed at addressing critical challenges in agricultural water management and promoting sustainable agriculture. Some key purposes of studying SIS include:

Water Conservation: One of the primary purposes of studying SIS is to optimize water usage in agriculture and minimize water wastage. By leveraging IoT technology and data-driven approaches, SIS aims to precisely regulate irrigation schedules, duration, and intensity, ensuring that crops receive the right amount of water at the right time while avoiding overwatering or underwatering.

Enhanced Crop Productivity: Another purpose of studying SIS is to enhance crop productivity and yields through improved irrigation practices. By maintaining optimal soil moisture levels and providing crops with adequate water and nutrients, SIS can promote healthier plants, reduce water stress, and increase overall crop yields, contributing to food security and livelihoods of farming communities.

Resource Efficiency: Studying SIS enables the identification and implementation of resource-efficient irrigation practices that minimize energy consumption, fertilizer usage, and environmental impact. By optimizing water, energy, and nutrient inputs, SIS helps reduce production costs, improve farm profitability, and mitigate the environmental footprint of agriculture.

Environmental Sustainability: SIS contributes to environmental sustainability by reducing water pollution, soil erosion, and greenhouse gas emissions associated with conventional

irrigation practices. By promoting soil health, biodiversity, and ecosystem resilience, SIS helps preserve natural resources and protect fragile ecosystems, ensuring the long-term viability of agricultural landscapes.

Technology Adoption and Innovation: Studying SIS fosters the adoption of innovative technologies and practices in agriculture, driving technological advancements and fostering collaboration between researchers, farmers, and industry stakeholders.

In summary, the purpose of studying Smart Irrigation Systems is to optimize water usage, enhance crop productivity, promote resource efficiency, and foster environmental sustainability in agriculture. By leveraging advanced technologies and interdisciplinary approaches, SIS offers a holistic solution to the challenges of water scarcity, food security, and environmental degradation, contributing to a more resilient and sustainable food system for future generations.

1.3 Objectives

The main objectives of Smart Irrigation Systems include water conservation, enhanced crop productivity, resource efficiency, environmental sustainability, precision irrigation, climate resilience, technology integration, and economic viability. By addressing these objectives, SIS contributes to the promotion of sustainable agriculture and the achievement of global food security goals.

1.4 Project Organization

In this report, Chapter one covers introduction, Purpose of the Study, objectives. Chapter two represents theory, description of the arduino UNO, soil moisture sensors, solar & water pump, LCD display and other components. Chapter three describes and working principle of the project. In chapter four we have discussed result and discussion, cost estimation, future modification, conclusion.

CHAPTER II

THEORY OF THE PROJECT

2.1 Introduction

This chapter includes the total over view of the device. In this chapter we have followed-up the theory of Arduino Uno, LCD Display, Resistor, Diode (IN 4007), , Transistor, briefly describe of Microcontroller (ATMEGA 328P), Crystal, LCD display (16x4), Vero board, and other components. Here we can know that the total system overview of the projects. And we will also know that how the equipment are working with each other.

2.2 Theory

We've developed the system that can handle this critical switching situation perfectly. Our system has several soil, and temputere sensors to measure any temperature, nitrogen, phosphorus, potassium, and pH Suitable for all kinds of soil.

2.2.1 Arduino Uno

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language and the Arduino Software (IDE), based on Processing.

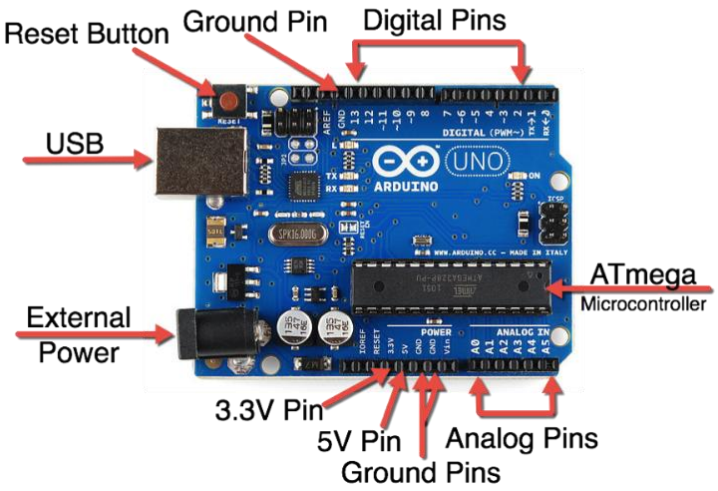


Figure 2.2.1. Arduino UNO Pin Diagram

Technical specifications:

Table 2.2.1: Technical specifications of Arduino UNO

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz

2.2.2 Microcontroller ATMEGA328P

The Atmel® picoPower® ATmega328/P is a low-power CMOS 8-bit microcontroller based on the AVR® enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega328/P achieves throughputs close to 1MIPS per MHz. This empowers system designer to optimize the device for power consumption versus processing speed.

The Atmel AVR® core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU),

allowing two independent registers to be accessed in a single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers. The ATmega328/P provides the following features: 32Kbytes of In-System Programmable Flash with Read-While-Write capabilities, 1Kbytes EEPROM, 2Kbytes SRAM, 23 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), three flexible Timer/Counters with compare modes and PWM, 1 serial programmable USARTs , 1 byte-oriented 2-wire Serial Interface (I2C), a 6- channel 10-bit ADC (8 channels in TQFP and QFN/MLF packages) , a programmable Watchdog Timer with internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main oscillator and the asynchronous timer continue to run. Atmel offers the QTouch® library for embedding capacitive touch buttons, sliders and wheels functionality into AVR microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully debounced reporting of touch keys and includes Adjacent Key Suppression® (AKS™) technology for unambiguous detection of key events. The easy-to-use QTouch Suite toolchain allows you to explore, develop and debug your own touch applications. The device is manufactured using Atmel’s high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The Boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega328/P is a powerful microcontroller that provides a highly flexible and cost effective solution to many

embedded control applications. The ATmega328/P is supported with a full suite of program and system development tools including: C Compilers, Macro Assemblers, Program Debugger/Simulators, In-Circuit Emulators, and Evaluation kits.

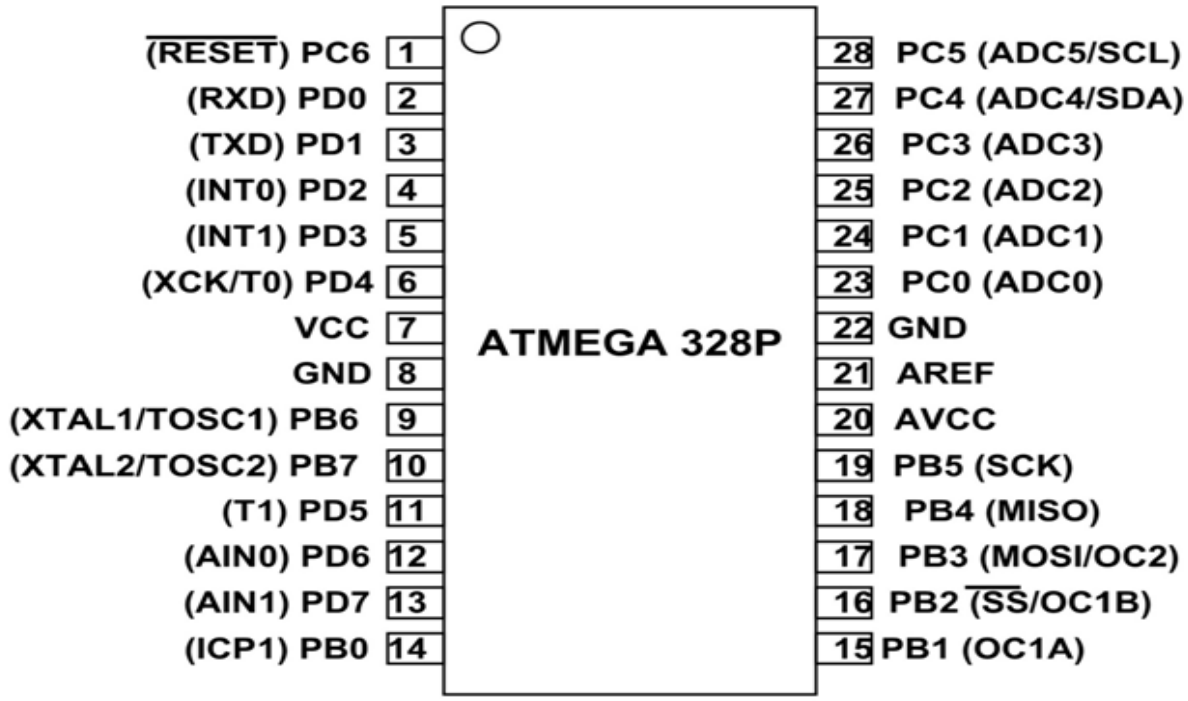


Figure 2.2.2: Pin Diagram Microcontroler

2.2.3 Design of solar power for smart irrigation project

A solar regulator must be able to handle the maximum current that can be produced by the solar panels. Reflected sunlight and specific temperature conditions can increase the output current of a solar panel by as much as 25% above its rated output current. The solar regulator must be sized to handle the increased current.

Step-1

The electrical load calculation for a project is given below.

Load	Watt
Submersible Pump	10
Microcontroller Unit	5
Sensor	3
Solar charger	10
Total	**28W**

Battery DC 12v
DC current required
P = V*I

Or, I = P/V

Or, = 28/12

Or, = 2.33A

Load is 28w and current required (I) = 2.33A

Step-2

How to Calculate Battery Size for Solar System?

Efficient battery capacity calculation is crucial for maximizing the benefits of a solar system. After understanding the factors that affect battery sizing, we can calculate the required battery capacity. To do this, consider the following aspects:

Here,

W= Total Load in watt
H= Backup Time in hour (8)
V= Battery Voltage volt

Formula= (W*H)/V
Or, = (28*8)/12
Or, = 18.66Ah
Battery standard value= 30 AH
So, Battery selection = 30AH

Battery charging current:

Step-3

We will calculate charging current for 30Ah battery. As we know that charging current should be 10% of the Ah rating battery.

Therefore,

Charging current for 30Ah battery
So, battery charging current is = 30*10% Ah
= 3A
Solar plate current = Battery charging current + Home load current
Or, = (3+2.33) A
Or, =5.33A

How to Calculate Solar Panel Requirements?

Step-4

Solar panels are assigned a power rating in watts, indicating the amount of electricity they can generate during a single hour of direct sunlight.

Formula = V*I
Or, = (12*5.33) Watt
Or, = 63.96W

28w Load = 63.96W solar panel required. 63.96w solar panel is not on the market, the solar panel below is available in the market.

10w, 20w, 40w, 50w, 65w

Finally we selected 65w Solar Panel. (Smart irrigation project)

Battery



30Ah

Solar Panel



65W

In conclusion, calculating the appropriate battery capacity for your solar system is essential for achieving energy independence and sustainability.

2.2.3.1 Solar Panel

A solar panel is a device that converts sunlight into electricity by using photovoltaic (PV) cells. PV cells are made of materials that produce excited electrons when exposed to light. The electrons flow through a circuit and produce direct current (DC) electricity, which can be used to power various devices or be stored in batteries. Solar panels are also known as solar cell panels, solar electric panels, or PV modules.

Some advantages of solar panels are that they use a renewable and clean source of energy, reduce greenhouse gas emissions, and lower electricity bills. Some disadvantages are that they depend on the availability and intensity of sunlight, require cleaning, and have high initial costs. Solar panels are widely used for residential, commercial, and industrial purposes, as well as in space, often together with batteries.

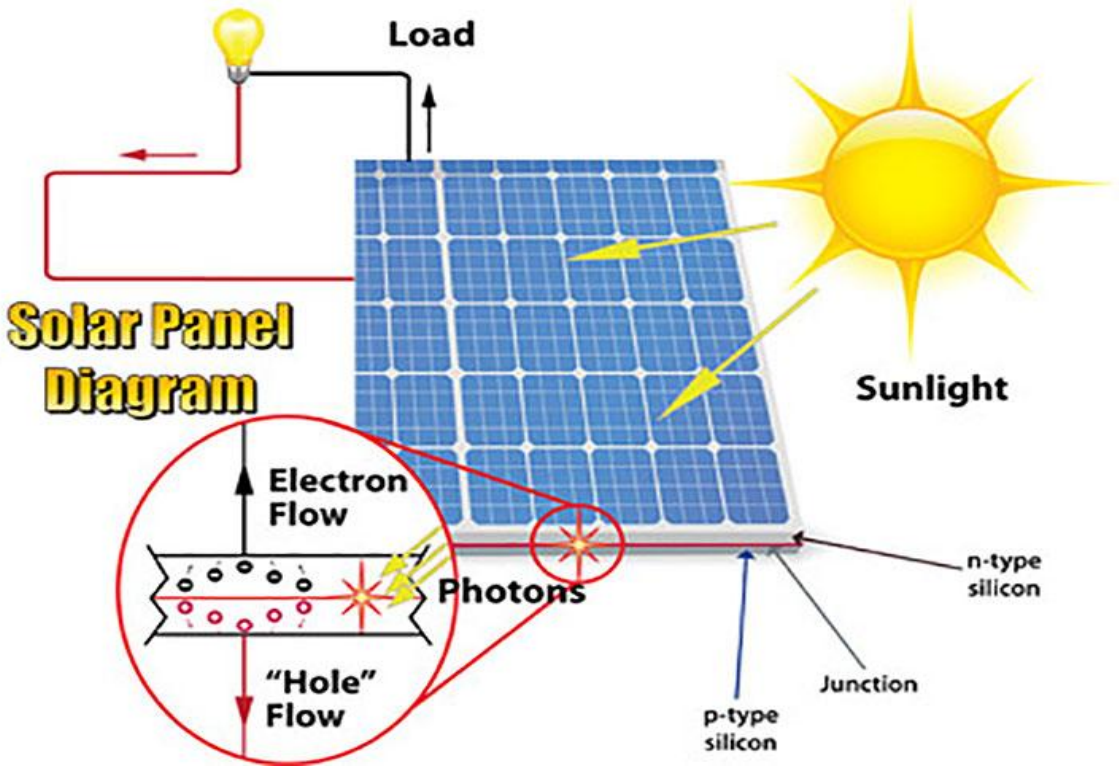


Figure 2.2.3.1: Solar Panel

2.2.4 NPK 7 in 1 soil Sensor

The soil npk sensor is suitable for detecting the content of nitrogen, phosphorus, and potassium, temperature in the soil, and pH judging the fertility of the soil. thereby facilitating the systematic evaluation of the soil condition. Can be buried in the soil for a long time, resistant to long-term electrolysis, corrosion resistance, vacuum potting, and completely waterproof. Soil npk sensors are widely used in soil nitrogen, phosphorus and potassium detection, precision agriculture, forestry, soil research, geological prospecting, plant cultivation and other fields.



Figure 2.2.4: 7 in 1 soil sensor

Specifications:

- Power supply: DC 4.5-30V
- Maximum power consumption: 0.5W (24V DC power supply)
- Working temperature: -20℃~+60℃
- Soil temperature: Range -40~80℃, Accuracy= ±0.5℃, Resolution=0.1℃
- Soil moisture: Range -0-100%,Accuracy= ±2% within 0-50%, Resolution=0.1%
- Core chip temperature resistance: 85℃
- Protection grade: IP68
- Probe material: Anti-corrosion special electrode
- Sealing material: Black flame-retardant epoxy resin
- The default cable length: 2 meters, the cable length can be customized as required
- Dimensions: 45*15*123mm
- Output signal: RS485

Advantages of 7in1 Soil Sensor

- High Accuracy, fast testing, can reach within 2% accuracy, increase the stability.
- Compared with traditional laboratory measurement, this producr has low cost, few steps, no re-agent, and no limited monitoring times.
- Insert or bury this sensor into the soil can measure all data details.

2.2.5 Water Pump

This DC 12V Mini Micro Submersible Water Pump is a low-cost, small-size Submersible Pump Motor that can be operated from 12V power supply. It can take up to 10 liters per hour with a very low current consumption of 220mA. Just connect the tube pipe to the motor outlet, submerge it in water, and power it. a mini water pump is a tiny or micro-DC water pump usually used to pressurize, circulate, or pump water for different uses. The DC type can have a rating of 3V, 5V, 6V, 12V, or 24V, and you can power it using a battery or solar.



Figure 2.2.5: Mini Water Pump

Specifications:

- DC Voltage: 12V
- Working current: 130-220mA
- Power: 0.4-1.5W
- Maximum lift: 40-110cm / 15.75"-43.4"
- Flow rate: 80-120L/H
- Outside diameter of water outlet: approx. 7.5mm / 0.3"
- Inside diameter of water outlet: approx. 4.7mm / 0.18"
- Diameter: approx. 24mm / 0.95"
- Length: approx. 45mm / 1.8"
- Height: approx. 33mm / 1.30"
- Wire length: about 15-20cm (red: " + ", black(white): " - ")
- Material: engineering plastic
- Driving mode: brushless DC design, magnetic driving
- Continuous working life of 500 hours

2.2.6 LCD Display 16x4

A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals. Liquid crystals do not emit light directly, instead using a backlight or reflector to produce images in colour or monochrome. LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images with low information content, which can be displayed or hidden, such as preset words, digits, and 7-segment displays, as in a digital clock. They use the same basic technology, except that arbitrary images are made up of a large number of small pixels, while other displays have larger elements.

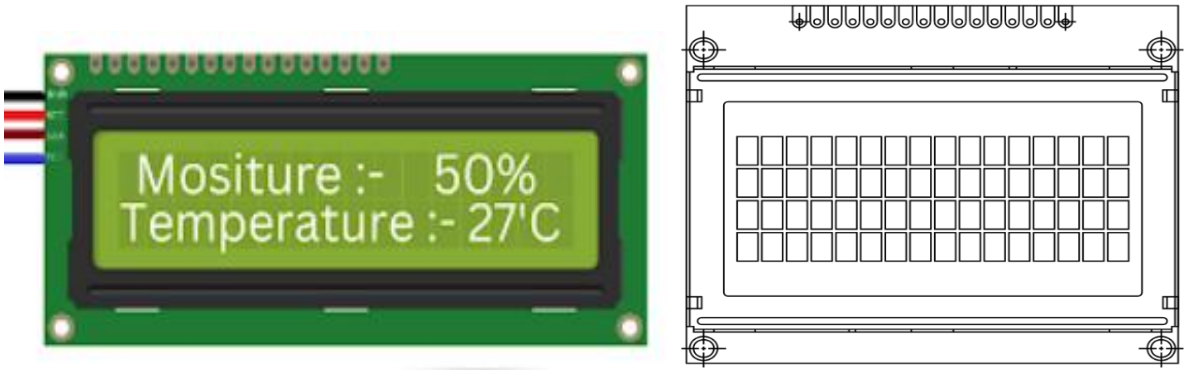


Figure 2.2.6.1: LCD Display

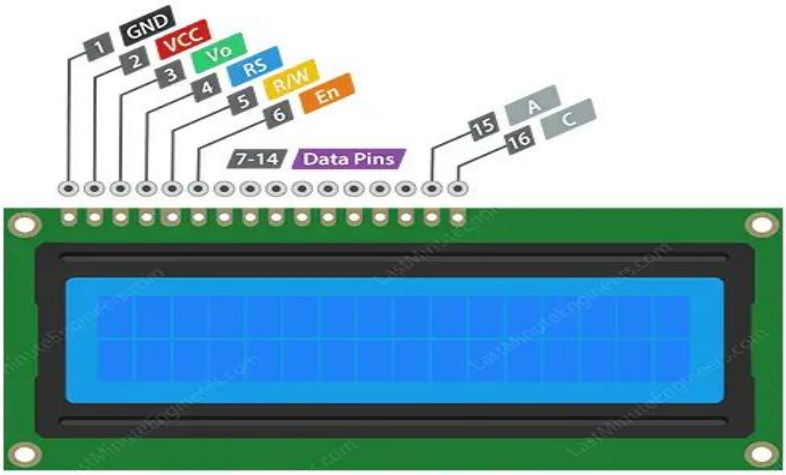


Figure 2.2.6.2: LCD Display pinout

2.2.7 DC Power supply (Rectifier)

A bridge rectifier circuit is a common part of the electronic power supplies. Many electronic circuits require rectified DC power supply for powering the various electronic basic components from available AC mains supply. We can find this rectifier in a wide variety of electronic AC power devices like home appliances, motor controllers, modulation process, welding applications, etc.

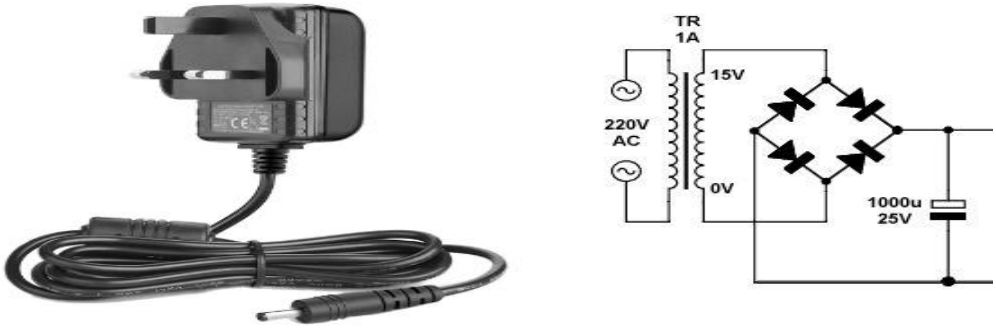


Figure 2.2.7.1: dc power supply

The bridge rectifier circuit diagram consists of various stages of devices like transformer, Diode Bridge, filtering and regulators. The first stage of the circuit is a transformer which is a step-down type that changes the amplitude of the input voltage. Most of the electronic projects uses 230/12V transformer to step-down the AC mains 230V to 12V AC supply. Next stage is a diode-bridge rectifier which uses four or more diodes depending on the type of bridge rectifier. Choosing a particular diode or any other switching device for a corresponding rectifier needs some considerations of the device like Peak Inverse Voltage (PIV), forward current If, voltage ratings, etc. It is responsible for producing unidirectional or DC current at the load by conducting a set of diodes for every half cycle of the input signal.

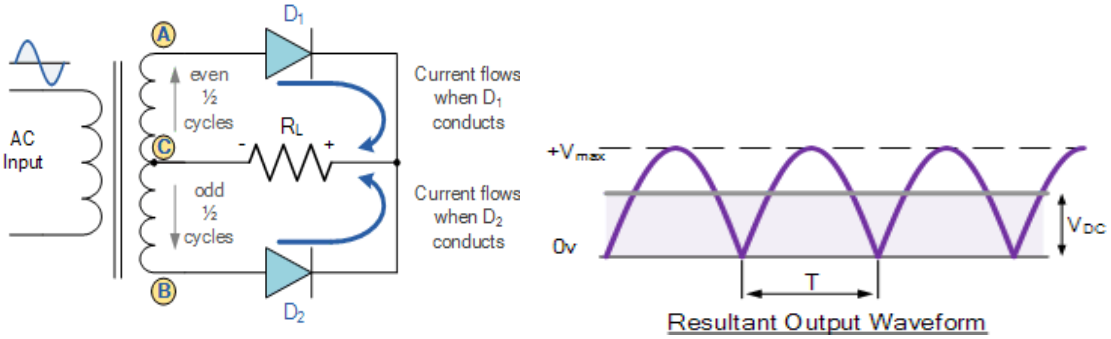


Figure 2.2.7.2: Output Waveforms of 220V/12V rectifier

2.2.8 5V Single-Channel Relay Module

Relay is an electromechanical device that uses an electric current to open or close the contacts of a switch. The single-channel relay module is much more than just a plain relay, it comprises of components that make switching and connection easier and act as indicators to show if the module is powered and if the relay is active or not.

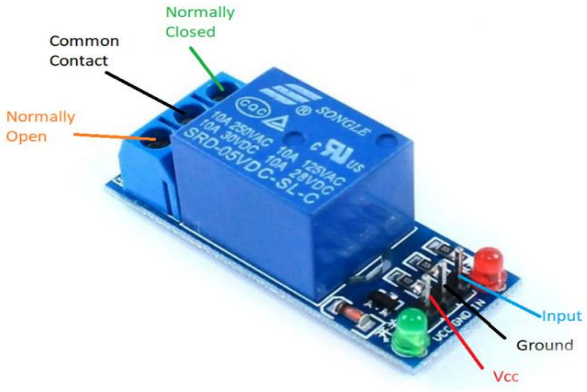


Figure 2.2.8: Single-Channel Relay Module

Single-Channel Relay Module Specifications

- Supply voltage – 3.75V to 6V
- Quiescent current: 2mA
- Current when the relay is active: ~70mA
- Relay maximum contact voltage – 250VAC or 30VDC
- Relay maximum current – 10A

2.2.9 ESP Wifi Module

The D1 module is a mini WIFI board based on ESP-8266EX. This WLAN board contains 11 digital input / output pins, all pins have interrupt / pwm / I2C / 1-wire and a micro-USB connection. The D1 mini development board is compatible with Arduino WeMos and can be programmed in the for Arduino IDE.

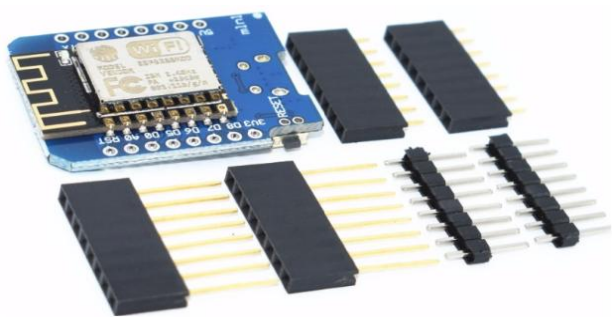


Figure 2.2.9.1: ESP Wifi Module

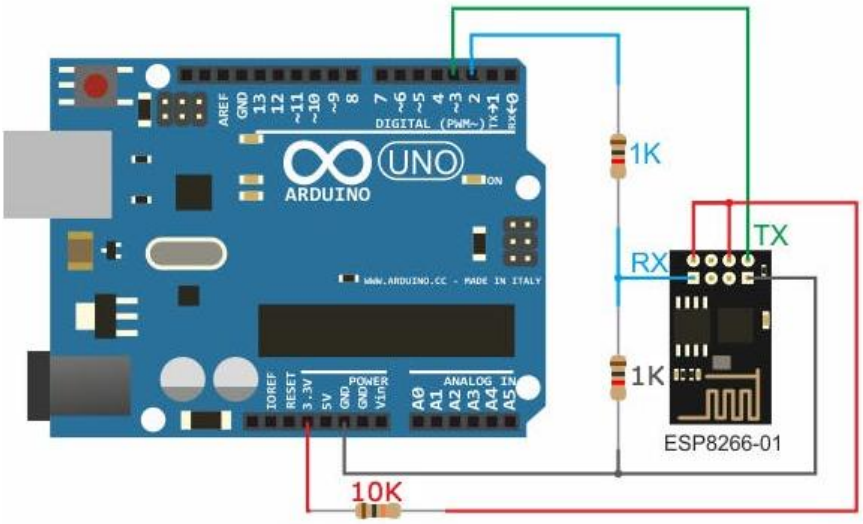


Figure 2.2.9.2: ESP Wifi Module with arduino

ESP Wifi Module Specifications

- Model: ESP8266-12E.
- Wireless Standard: 802.11 b/g/n.
- Frequency range: 2.4 GHz - 2.5 GHz (2400M-2483.5M)
- Wi-Fi mode: Station / SoftAP / SoftAP+station.
- Stack: Integrated TCP/IP.
- Output power: 19.5dBm in 802.11b mode.
- Data interface: UART / HSPI / I2C / I2S / Ir.
- Remote Control GPIO / PWM.

2.2.10 12v Li-po Battary

A lithium-polymer battery (LiPo) is a rechargeable battery that, in the case of true LiPo, uses solid polymer for the electrolyte and lithium for one of the electrodes. Commercially available LiPo are hybrids: gel polymer or liquid electrolyte in a pouch format, more accurately termed a lithium ion polymer battery.



Figure 2.2.10: Lipo battary

Lipo Battary Specifications

- Weight: 64g.
- Size: 135mm x 45mm x 5mm.
- Charge voltage 4.2v, nominal voltage 3.7V, cutoff at 3.0 volts.
- Recommended charge current 600mA.
- Charge current Max 1500mA.
- Recommended discharge current 1500mA.
- Discharge current Max 3000mA.
- Operating temperature: 0 to 55°C during charging, -25 to 60°C during discharge.

2.2.11 DC Buck Converter

A buck converter or step-down converter is a DC-to-DC converter which decreases voltage, while increasing current, from its input (supply) to its output (load).



Figure 2.2.11: DC Buck Converter

2.2.12 Connection Wire/ Jumper wires

Jumper wires are used to connect different electronic components on a breadboard or prototype board. They are essential for creating electrical connections between various pins, helping to establish a circuit for testing or prototyping.



Figure 2.2.12: Jumper wires

2.2.13 PCB Board

A printed circuit board is a rigid structure that contains electrical circuitry made up of embedded metal surfaces called traces and larger areas of metal called planes. Components are soldered to the board onto metal pads, which are connected to the board circuitry**.**

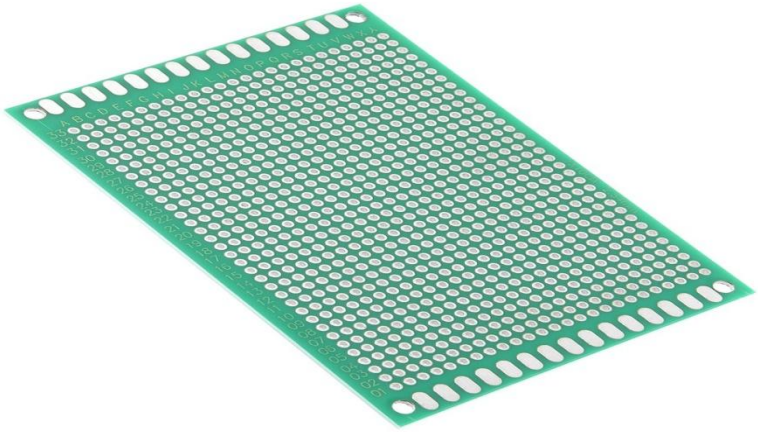


Figure 2.2.13: Pcb board

2.2.14 Scrow Connection

A screw terminal is a type of electrical connection where a wire is held by the tightening of a screw.



Figure 2.2.14: screw terminal

CHAPTER III

DESING & FEBRICATION

3.1 Introduction

In this chapter fully cover with discuss design and fabrication of this project. Here we will discuss about developed block diagram and briefly describe about the circuit description and also learn about working principle. Total project flow chart is also available in this chapter.

3.2 Block Diagram

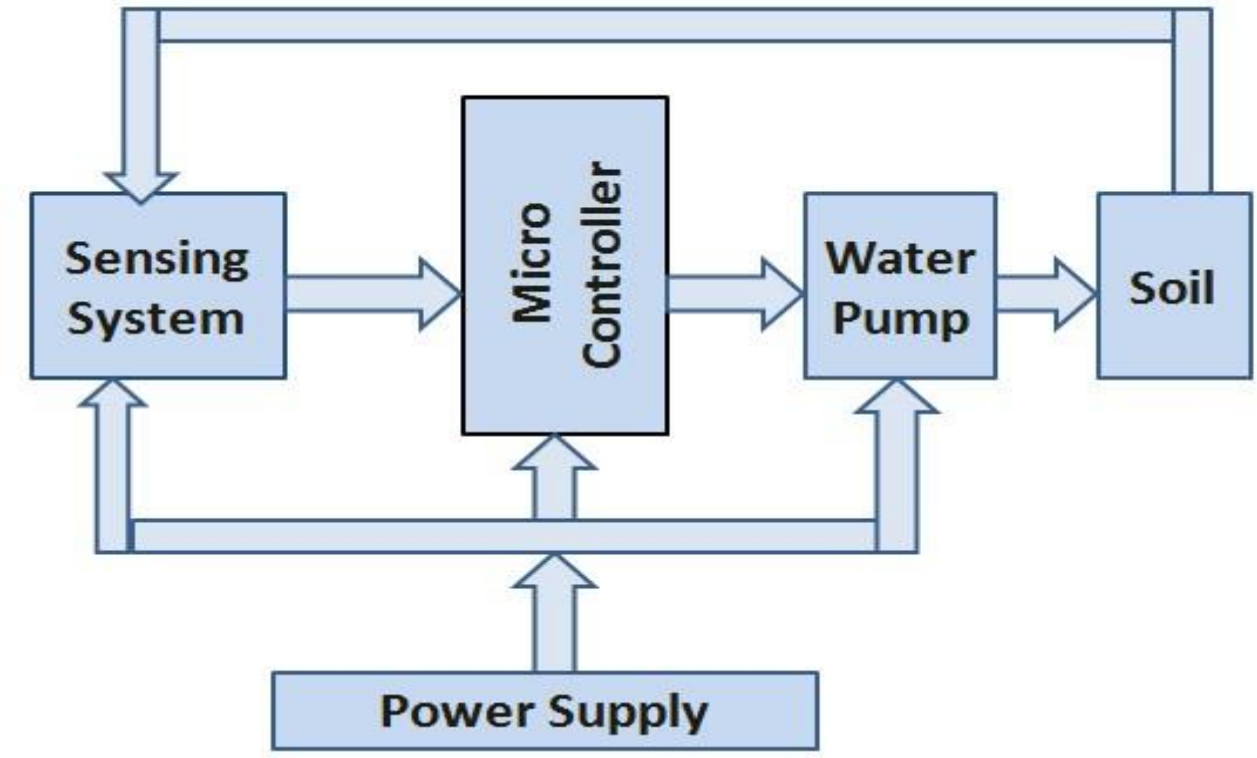


Figure 3.2: Block diagram of project

Here's the block diagram of our smart irrigation system. We can clearly describe the working process through this block diagram. Firstly we have two energy sources to supply electricity. One is the Solar and the other is the battary. Generally Solar is the first priority of supplying the electricity. Our control system continuously monitors the voltage and the current of both

sources. If any unwanted situation occurs such as under voltage of a sources, or power failure of any of those sources, our system immediately switches the supply line to another sources (battary) that has no problem and vice versa. If both of the sources lose their voltage to the minimum threshold voltage, it combines the two sources. A voltage stabilizer keeps the output load voltage always stable as the loads need. All processes, electrical conductivity, temperature, nitrogen, phosphorus, potassium, and pH is shown in a (control circuit) LCD display.

3.3 Circuit Description

The following image represents the connections among all the components. The brain of the system is an Arduino UNO microcontroller board. It collects the data from 7 in 1 soil sensor and procsses the data. After procession te data, it displays the infromation to an lcd display. An ESP8266 WiFi module is attached to the system, Arduino sends the data to the wifi module and the modules upload the data to Ubidots cloud server. Data can be remotely monitored from remote location through graphical dashboard.

The wifi module also helps to read the weather information from weather server and based on the sensor reading and weather data Arduino controls the water pump. The whole system is powered from a solar.

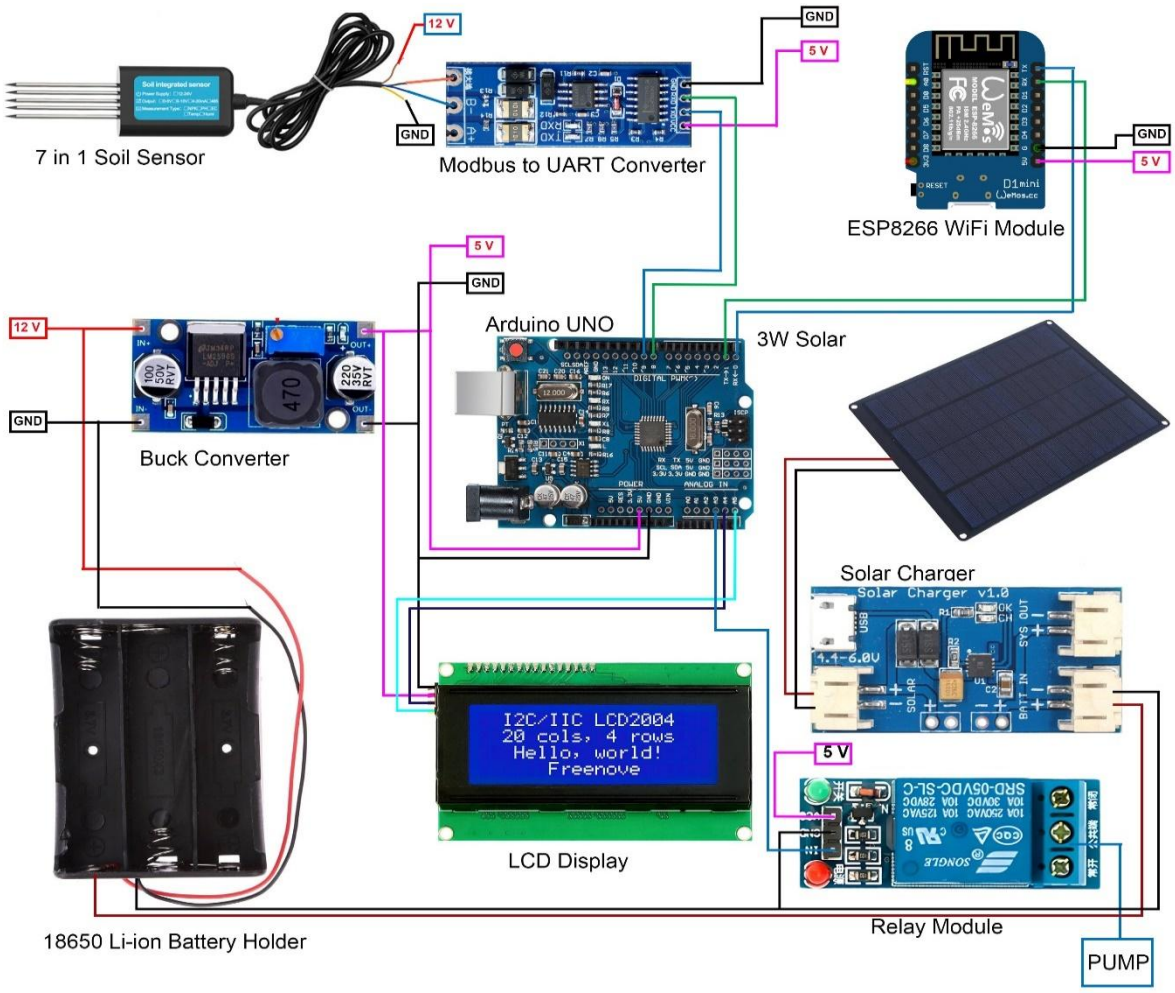


Figure 3.3.1: Connection diagram of the Project-1

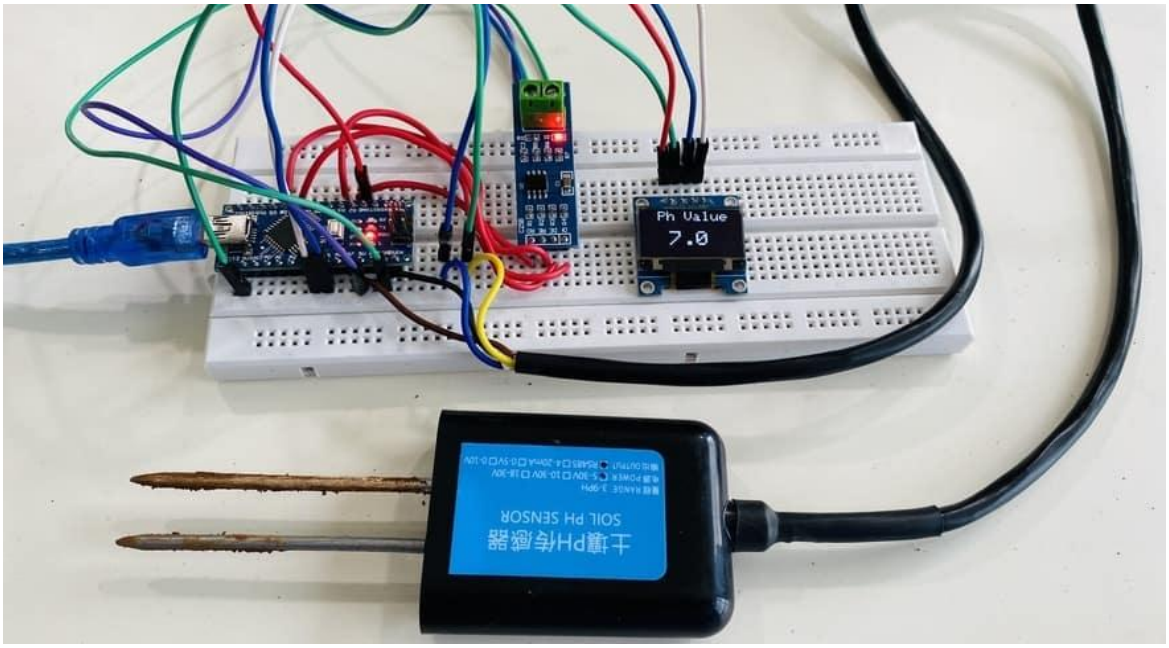


Figure.3.3.2: Circuit diagram of the Project-2

3.4 Overview of Smart Irrigation System

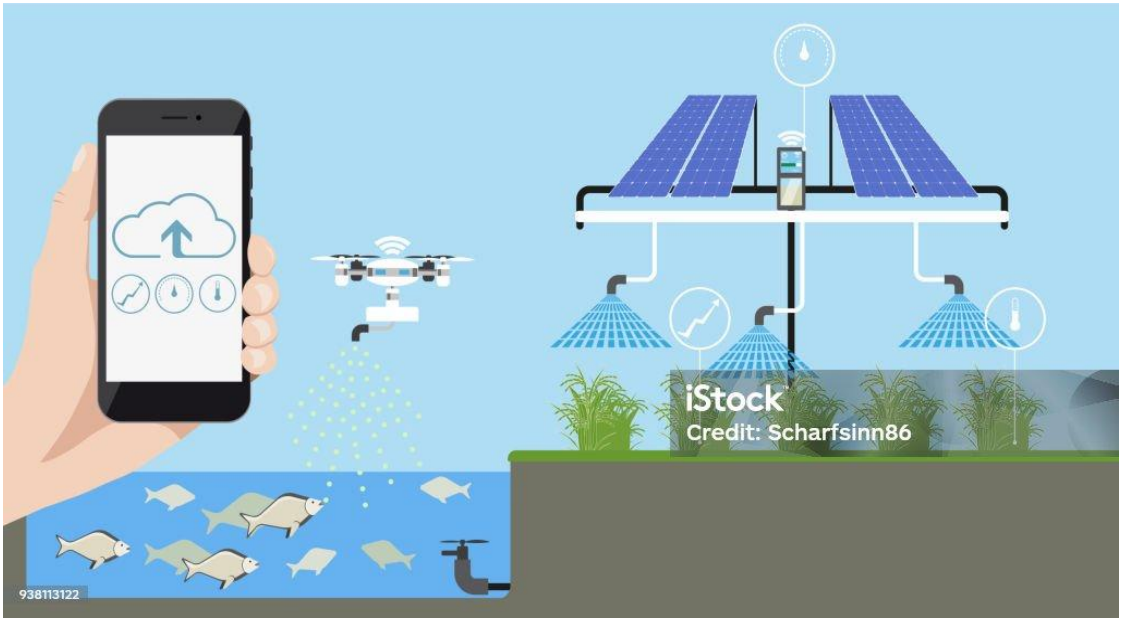


Figure 3.4: Smart irrigationTechnology

(Use an image from Google) In the block diagram above, after providing electricity, the ESP8266 Wi-Fi module receives the instructions to turn on the motor via the app or manually using a switch. A comprehensive and self-contained Wi-Fi network solution that can transport software programmes is the ESP8266 Wi-Fi module. A relay will be used to turn on the motor in accordance with the programme that was burned into the Wi-Fi module. A relay is an electromagnetic component that connects two electrically isolated circuits magnetically. The ESP8266 Wi-Fi module receives input from the soil moisture sensor. The farm's moisture level was detected using the soil moisture sensor. Once it reaches the soil, it determines the amount of moisture there is.

3.5 Working Principle

The system is a combination of hardware and software components. The hardware part consists of embedded system and software is the webpage designed using Ubidots. The webpage is hosted online and consists of a database in which readings from sensors are inserted using the hardware.

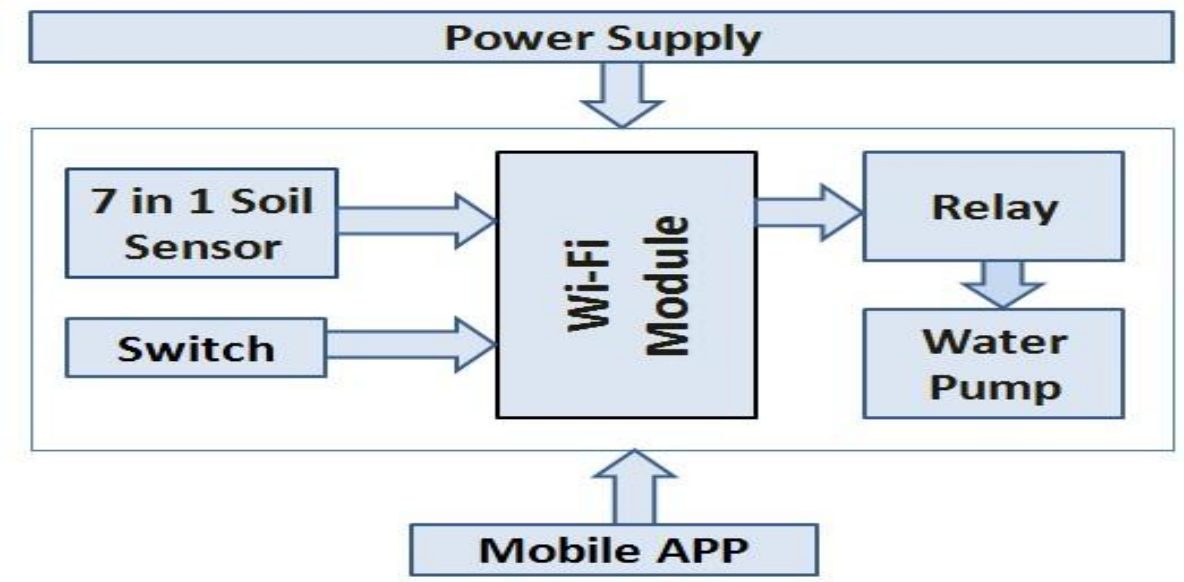


Figure 3.5: Overall Design (ESP)

3.5.1 7in1 Soil Sensor Section

NPK sensors work based on various technologies, but one common approach is using ion-selective electrodes. These electrodes are designed to selectively detect the ions of interest (N, P, and K). When these electrodes come into contact with the soil, they generate electrical signals proportional to the concentration of the specific ion present.soil NPK sensor actually measures the electrical conductivity of the soil. The manufacturer multiplies the measured conductivity value by a corresponding factor (based on the conventional soil content of nitrogen, phosphorus and potassium) to arrive at a value for the nitrogen, phosphorus and potassium content.Due to the different soils and environments on site, such sensors cannot accurately measure the actual nitrogen, phosphorus and potassium content of the soil on site, but give an empirical, theoretical value..

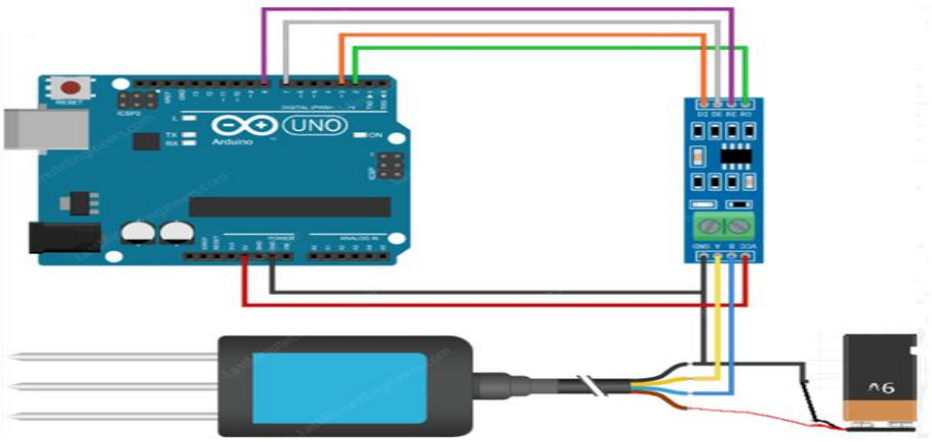


Figure 3.5.1 : 7in1 Soil Sensor with Uno

3.5.2 Control Section

Information from the sensors is transmitted to the arduino board. The arduino board consists of microcontroller ATMEGA328P which is responsible for controlling the switching on/off of the motor on which water sprinklers can be attached. Sensor values from arduino are transmitted to the ESP module. The ESP8266 wifi module is low cost standalone wireless transceiver that can be used for end-point IoT developments. ESP8266 wifi module enables internet connectivity to embedded applications. It uses TCP/UDP communication protocol to connect with the server/client. To communicate with the ESP8266 wifi module, microcontroller needs to use set of AT commands. The microcontroller communicates with ESP8266-01 wifi module using UART having specified Baud rate (Default 115200).

3.5.3 Mobile Apps Section

This section comprises of a webpage which displays the current water sprinkler status i.e. on or off and a button which redirects the user to a thing speak page which graphically depicts the sensor values.

3.6 Image of the Project

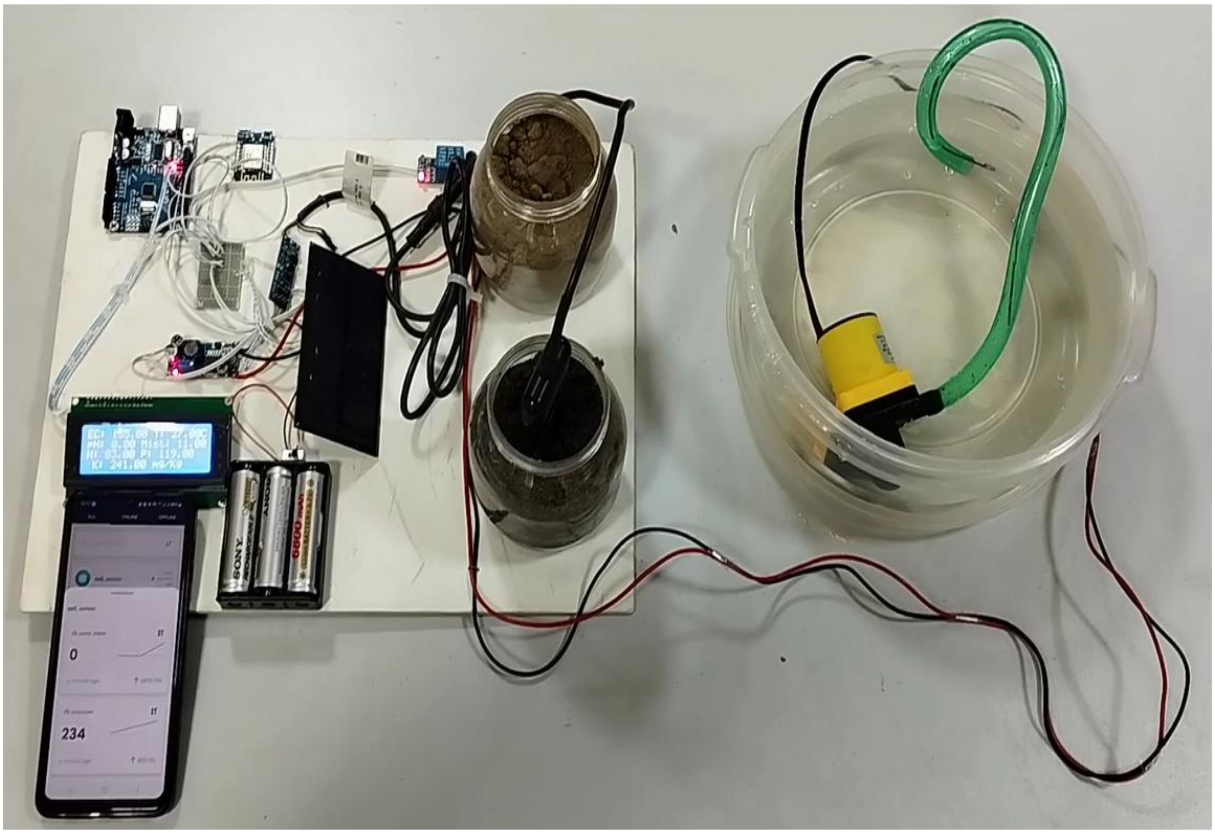


Figure 3.6: Real image of the project

The picture above is the actual demo project we've made. Now let's take a look at the step by step working process. After powering up everything these two sources namely Solar (on the left) and battery (on the right) .The output Power goes to the sensor, control circuit and pump.

3.7 Flow Chart

The following flow diagram shows how the full system works. The algorithm is developed in the system using C programming language. For cloud storage Ubidots cloud server is used and the system uses MQTT protocol for uploading data to cloud. In this flowchart we can easily determine the working process of our system. A diagrammatic illustration of a solution model for a specific problem is shown in Figure 4.3. In many different industries, flowcharts are used for process analysis, design, documentation, and programme management.

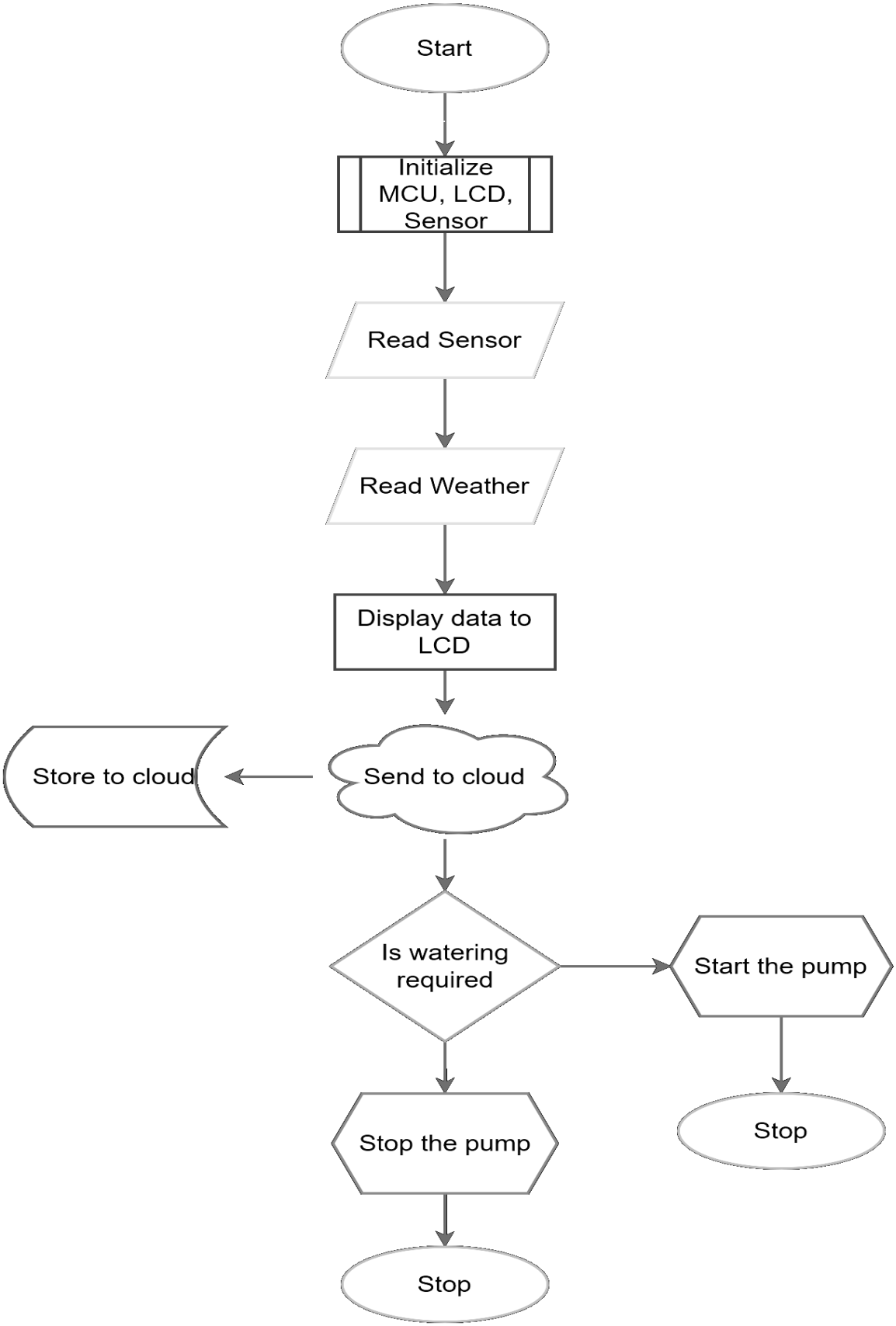


Figure 3.7: Flow chart of the project

3.8 Used of Materials

The Hardware and Software utilised in this project are explained as follows:

3.8.1 Hardware Equipment

1.	7 in 1 Soil Sensor	- 1 pcs
2.	Relay Module	- 2 pcs
3.	Buck Converter	- 1 pcs
4.	Wifi Module	- 1 pcs
5.	Solar Panel	l- 1 pcs
6.	Bread board	-1pcs
7.	Lipo Battary 12v	- 1 pcs
8.	LCD16x4	-1 pcs
9.	DC power supply	- 1 pcs
10.	Water pump	- 1 pcs
11.	Switchs	- 3 pcs
12.	PCB Board	- 2 pcs
13.	Wires	As needed
14.	Connector	- 2 pcs
15.	Glue Gun Stick	- 2 pcs

3.8.2 Software Equipment

1. Arduino UNO
2. Blynk app/ Ubidots App
3. ESP Module

3.9 Conclusion

A system to monitor moisture levels in the soil was designed and the project provided an opportunity to study the existing systems, along with their features and drawbacks. The proposed system can be used to switch on/off the water sprinkler according to soil moisture levels thereby automating the process of irrigation which is one of the most time consuming activities in farming. Agriculture is one of the most water-consuming activities. The system uses information from soil moisture sensors to irrigate soil which helps to prevent over irrigation or under irrigation of soil thereby avoiding crop damage. The farm owner can monitor the process online through a website. Through this project it can be concluded that there can be considerable development in farming with the use of IOT and automation. Thus, the system is a potential solution to the problems faced in the existing manual and cumbersome process of irrigation by enabling efficient utilization of water resources.

Chapter IV

RESULT & DISCUSSION

4.1 Introduction

This chapter contains the results obtained and discussion about the project. We have also covered discussions about advantages, disadvantages and limitation of current version of the protection system.

4.2 Result

The values obtained through sensors enable the system to switch the sprinkler on and off. A farmer can remotely monitor the irrigation process on the farm. Hence, the system contributed in making a smart farm. Table 2 depicts the readings of the two YL-69 soil moisture sensors taken over a period of one hour It can be observed from the graphs that moisture content of over irrigated soil takes longer period to reduce by 10% as compared to the moisture content of soil which is initially 79% moist. Thus, the curve for sensor A is a continuously decreasing curve as the moisture reduces as time passes. For sensor B the moisture value initially is 00%. The readings depict that sensor was in an over irrigated soil at the beginning. In our system possibility of over irrigation is avoided by turning the water sprinklers off as soon as the sensor reading is 100%

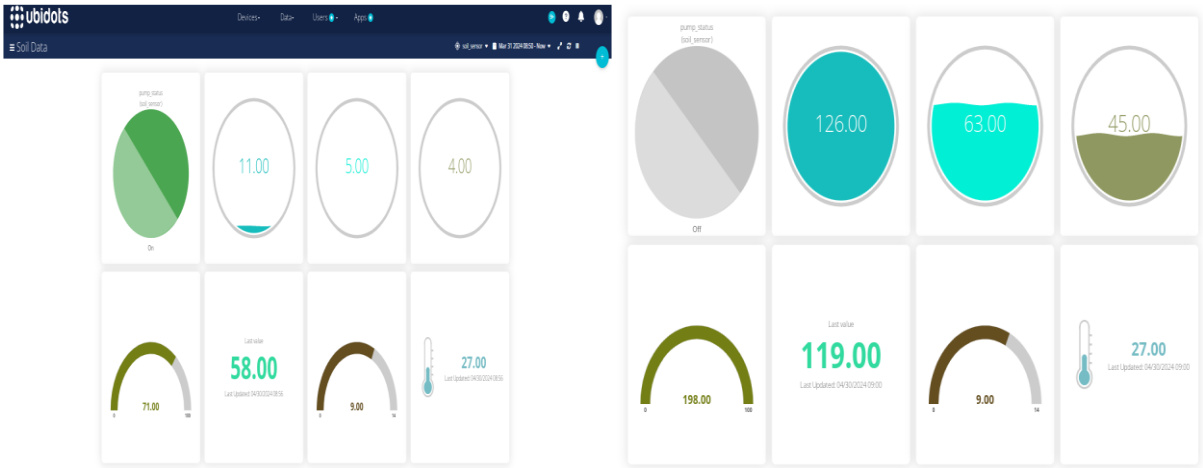


Figure 4.2.: Output result (Apps)

4.3 Advantags of Smart Irrigation System

The advantages of Smart Irrigation Systems (SIS) are manifold, offering numerous benefits to farmers, the environment, and society as a whole. Some key advantages of SIS include:

- Reduced labour.
- Timely irrigation — plants being watered when needed.
- Management of higher flow rates.
- Accurate cut-off of water compared to manual checking.
- Reduced runoff of water and nutrients.
- Reduced costs for vehicles used to check irrigation.

4.4 Disadvantages of Smart Irrigation System

- Smart watering system is a bit expensive.
- Depending on the size of your property, you will need more systems. of course, saving on water bills will lead to less cost.
- If you want to use this system for pipe watering, it's better to fix it under the ground before planting.
- This system work under the condition of light.

4.5 Discussion

- The results of the "Soil Moisture Sensor using Arduino and Automatic Irrigation using DC Pump" project showed that the system was able to effectively measure soil moisture levels and respond by activating the DC pump to water the plants.
- The soil moisture sensor was able to accurately measure the moisture content of the soil and transmit this information to the Arduino board. Based on pre-defined moisture level thresholds, the Arduino board was able to activate the DC pump to irrigate the plants when the soil was too dry.
- The system was able to operate continuously for an extended period of time without any issues, indicating that itwas a reliable and efficient method for monitoring and maintaining plant hydration levels.
- The automatic irrigation using a DC pump resulted in better plant growth and health compared to manual watering. It was able to provide the plants with the correct amount of water at the right time, preventing under or over-watering.

❖ Overall, the project demonstrated the potential of using technology to automate plant irrigation and ensure optimal growth and health. It can be a useful tool for farmers, gardeners, and even in residential settings to maintain green spaces efficiently

4.6 Cost Estimation

Table 4.5: Cost Estimation

SI	Particulars	Specification	Qty.
1	Arduino Uno	ATMEGA328P	1
2	Microcontroller	ATMEGA328P	2
3	NPK 7 in 1 Soil sensor	RS485	1
4	LCD Display	16x4	2
5	DC Power Supply	5V 2A Adapter	1
6	Solar Panel	6 watts	1
7	Wi-Fi Module	ESP8266	1
8	Water Pump	12vDC (submersible)	1
9	Relay Module	Single channel	1
10	DC Buck Converter	LM2596DC-DC	1
11	PCB Board	Chine	1
12	Vero Board	Local	1
13	Glue Stick	Local	3
14	wire	As per need	20ft
15	Resistor	Various	20
16	Soldering lead	N/A	As per need
17	Others	N/A	As per need

4.7 Applications of Project

- Depending on the crop and kind of soil, the system must be able to control every last drop and determine the actual amount of water needed. This is a highly useful feature that enables waste-free water conservation .
- Both small and large fields can be used with this technique
- A trustworthy and effective method for monitoring environmental parameters;
- Easy and convenient garden maintenance.

4.8 Conclusion

With the help of a soil moisture sensor and a Esp8266 Wi-Fi module, the "Design and Implementation of smart irrigation system by using advanced level sensor" project optimises the use of water in agricultural fields without the need for human intervention from farmers. The motor is turned on and off in accordance with commands from an Android app. Only the necessary amount of water is provided to each crop in this method, and the Android app displays the field's present condition.

Chapter V

CONCLUSION

5.1 Conclusion

In this study, a Design and Implementation of smart irrigation system by using advanced level sensor Using Arduino was developed. Themicrocontroller circuit has been made with few components and the circuit is highlyreliable. This circuit consisted of Arduino UNO, relay, soil moisture sensor, motor (12 v) and battery. This system uses information from soil moisture sensors to irrigate the soil,which helps prevent excessive irrigation and insufficient irrigation. This system works asa potential solution to the problems faced in manual irrigation in the plant. This intelligentirrigation system was designed for flood saturation and irrigation. Both operations were performed on clayey and sandy soil in a pot. From the result, for the saturation,This study revealed that the sensor-based irrigation system required less water and less time than the manual irrigationsystem. Increased irrigation efficiency. The sensor-based irrigation system is the mostreliable system and has worked automatically without help. It makes the system moreefficient and convenient. The main limitation of this study is that it is tested on a smallscale; further studies are needed for largescale agriculture.

5.2 Future Work

In the future, a wireless component like a wireless soil moisture sensor can take the place of a wired network. RFID technology can also be used to create wireless systems. At each phase, we can add an electric solenoid valve that can be automatically controlled by an Arduino board or by an ESP8266 Wi-Fi module.

References

[1] https://www.scribd.com/document/654552350/smart-irrigation-system-maheen

[2] file:///C:/Users/Administrator/Desktop/6Smart-irrigation-system-using-arduino-with-solar-power-IJERTV9IS050088.pdf

[3] https://www.irjet.net/archives/V10/i7/IRJET-V10I715.pdf

Appendix

Code for Ardiuno :

```
#include <AltSoftSerial.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4);  // set the LCD address to 0x27 for a 16 chars and 2 line display
int pump_status = 0;
int pump = 12;
float humidity, tempe, phs, moistur, nitrosa, phosphorus, potass;
const byte temp[] = {0x01,0x03, 0x00, 0x13, 0x00, 0x01, 0x75, 0xCF};
const byte mois[]  = {0x01,0x03,0x00,0x12,0x00,0x01,0x24,0x0F};
const byte econ[] = {0x01,0x03, 0x00, 0x15, 0x00, 0x01, 0x95, 0xCE};
const byte ph[] = {0x01,0x03, 0x00, 0x06, 0x00, 0x01, 0x64, 0x0B};
const byte nitro[] = { 0x01, 0x03, 0x00, 0x1E, 0x00, 0x01, 0xE4, 0x0C };
const byte phos[] = { 0x01, 0x03, 0x00, 0x1F, 0x00, 0x01, 0xb5, 0xCC };
const byte pota[] = { 0x01, 0x03, 0x00, 0x20, 0x00, 0x01, 0x85, 0xC0 };
byte values[11];
String msg = "";
AltSoftSerial mod;
float envhumidity = 0.0, envtemperature = 0.0, soil_ph = 0.0, soil_mois = 0.0, soil_temp = 0.0;
byte val1 = 0, val2 = 0, val3 = 0, val4 = 0,val5 = 0, val6 = 0, val7 = 0;
```

```
void setup() {
  lcd.init();
  lcd.init();
  //Welcome message
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("Welcome to...");
  lcd.setCursor(2,1);
  lcd.print("Smart Irrigation");
  lcd.setCursor(6,2);
  lcd.print("Project");
  lcd.setCursor(0,3);
  lcd.print("By: Mosarof Hossain");
  pinMode(pump, OUTPUT);
  //pump is off
  digitalWrite(pump, HIGH);
   Serial.begin(9600);
  mod.begin(9600);
  delay(3000);
}
void loop() {
  val1 = moisture();
  //if(val1>100) val1 = 100;
  soil_mois = val1;
  soil_temp = random(26,28);//temperature()/10.0;
  val3 = econduc();
  int conduct = val3;
  val4 = phydrogen()/10;
  soil_ph = val4; if(soil_ph>9) soil_ph = soil_ph - 5;
  val5 = nitrogen();
  int nitros = val5;
  val6 = phosphorous();
  int phosp = val6;
```

```
  val7 = potassium();
  int potas = val7;
  humidity = conduct, tempe = soil_temp, phs = soil_ph, moistur = soil_mois, nitrosa = nitros,
phosphorus = phosp, potass = potas;

  if(potas<100){
    digitalWrite(pump,LOW);
    pump_status = 1;
    delay(1000);
    }
  else if(potas>100){
    digitalWrite(pump,HIGH);
    pump_status = 0;
    delay(1000);
    }
  display_soil();
  msg += soil_mois;
  msg += ",";
  msg += soil_temp;
  msg += ",";
  msg += conduct;
  msg += ",";
  msg += soil_ph;
  msg += ",";
  msg += nitros;
  msg += ",";
  msg += phosp;
  msg += ",";
  msg += potas;
  msg += ",";
  msg += pump_status;
  msg += "\n";
  Serial.print(msg);
```

```
  msg = "";
  /*
  delay(1000);
  Serial.print("Moisture: ");Serial.print(soil_mois);Serial.println(" %");
  delay(1000);
  Serial.print("Temperature: ");Serial.print(soil_temp);Serial.println(" C");
  delay(1000);
  Serial.print("EC: ");Serial.print(val3);Serial.println(" us/cm");
  delay(1000);
  Serial.print("ph: ");Serial.print(soil_ph);Serial.println(" ph");
  delay(1000);
  Serial.print("Nitrogen: "); Serial.print(val5);Serial.println(" mg/kg");
  delay(1000);
  Serial.print("Phosphorous: ");Serial.print(val6);Serial.println(" mg/kg");
  delay(1000);
  Serial.print("Potassium: ");Serial.print(val7);Serial.println(" mg/kg");
  Serial.println();
  */
  delay(5000);
}
byte moisture() {
  mod.flushInput();
  delay(1);
  for (uint8_t i = 0; i < sizeof(mois); i++) mod.write(mois[i]);
  mod.flush();
  delay(500);
  for (byte i = 0; i < 7; i++) {
    values[i] = mod.read();
  }
  return values[4];
}
byte temperature() {
  mod.flushInput();
```

```
  delay(1);
  for (uint8_t i = 0; i < sizeof(temp); i++) mod.write(temp[i]);
  mod.flush();
  delay(500);
  for (byte i = 0; i < 7; i++) {
    values[i] = mod.read();
  }
  return values[3]<<8|values[4];
}
byte econduc() {
  mod.flushInput();
  delay(1);
  for (uint8_t i = 0; i < sizeof(econ); i++) mod.write(econ[i]);
  mod.flush();
  delay(500);
  for (byte i = 0; i < 7; i++) {
    values[i] = mod.read();
  }
  return values[4];
}
byte phydrogen() {
  mod.flushInput();
  delay(1);
  for (uint8_t i = 0; i < sizeof(ph); i++) mod.write(ph[i]);
  mod.flush();
  delay(500);
  for (byte i = 0; i < 7; i++) {
    values[i] = mod.read();
  }
  return values[4];
}
byte nitrogen() {
  mod.flushInput();
```

```
  delay(1);
  for (uint8_t i = 0; i < sizeof(nitro); i++) mod.write(nitro[i]);
  mod.flush();
  delay(500);
  for (byte i = 0; i < 7; i++) {
    values[i] = mod.read();
  }
  return values[4];
}
byte phosphorous() {
  mod.flushInput();
  delay(1);
  for (uint8_t i = 0; i < sizeof(phos); i++) mod.write(phos[i]);
  mod.flush();
  delay(500);
  for (byte i = 0; i < 7; i++) {
    values[i] = mod.read();
  }
  return values[4];
}
byte potassium() {
  mod.flushInput();
  delay(1);
  for (uint8_t i = 0; i < sizeof(pota); i++) mod.write(pota[i]);
  mod.flush();
  delay(500);
  for (byte i = 0; i < 7; i++) {
    values[i] = mod.read();
  }
  return values[4];
}
void display_soil(){
   lcd.clear();
```

```
  lcd.setCursor(0,0);
  lcd.print("EC: ");
  lcd.print(humidity);
  lcd.print(" ");
  lcd.print("T: ");
  lcd.print(tempe);
  lcd.print("C");
    lcd.setCursor(0,1);
  lcd.print("pH: ");
  lcd.print(phs);
  lcd.print(" ");
  lcd.print("Mist: ");
  lcd.print(moistur);
  lcd.setCursor(0,2);
  lcd.print("N: ");
  lcd.print(nitrosa);
  lcd.print(" ");
  lcd.print("P: ");
  lcd.print(phosphorus);
  lcd.setCursor(0,3);
  lcd.print(" ");
  lcd.print("K: ");
  lcd.print(potass);
  lcd.print(" mg/Kg");
  delay(1000);
```

Code for ESP:

```
#define BLYNK_PRINT Serial
/* Fill in information from Blynk Device Info here */
#define BLYNK_TEMPLATE_ID           "TMPL6TKuSuiYP"
#define BLYNK_TEMPLATE_NAME "IoT Smart Grid"
#define BLYNK_AUTH_TOKEN "DvgjV5EVRipuusYsxt0OglSuHMk508-4"
```

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Demark";
char pass[] = "721035za";
String inputString = "";         // a string to hold incoming data
boolean stringComplete = false;
long timer = 0;
// This function will be called every time Slider Widget
// in Blynk app writes values to the Virtual Pin 1
BLYNK_WRITE(V1)
{
  int pinValue = param.asInt(); // assigning incoming value from pin V1 to a variable
  if(pinValue==1){
     Serial.println("relay1on\n");
     }
  else if(pinValue==0){
     Serial.println("relay1off\n");
     }
}
BLYNK_WRITE(V2)
{
  int pinValue = param.asInt(); // assigning incoming value from pin V1 to a variable
  if(pinValue==1){
     Serial.println("relay2on\n");
     }
  else if(pinValue==0){
     Serial.println("relay2off\n");
     }
}
void setup()
{
```

```
  // Debug console
  Serial.begin(115200);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  inputString.reserve(200);
}
void loop()
{
  Blynk.run();
  if (stringComplete) {
     int i1 = inputString.indexOf(',');
     int i2 = inputString.indexOf(',', i1+1);
     int i3 = inputString.indexOf(',', i2+1);
     int i4 = inputString.indexOf(',', i3+1);
     int i5 = inputString.indexOf(',', i4+1);
     int i6 = inputString.indexOf(',', i5+1);
     int i7 = inputString.indexOf(',', i6+1);
     String voltage = inputString.substring(0, i1);
     String current = inputString.substring(i1 + 1, i2);
     String active = inputString.substring(i2 + 1, i3);
     String frequency = inputString.substring(i3 + 1, i4);
     String pf = inputString.substring(i4 + 1, i5);
     String energy = inputString.substring(i5 + 1, i6);
     String apprant = inputString.substring(i6 + 1, i7);
     String reactive = inputString.substring(i7 + 1);
     /*
     Serial.println(voltage);
     Serial.println(current);
     Serial.println(active);
     Serial.println(frequency);
     Serial.println(pf);
     Serial.println(energy);
     Serial.println(apprant);
     Serial.println(reactive);
```

```
    */
    double volt = voltage.toDouble();
    double amp = current.toDouble();
    double watt = active.toDouble();
    double hartz = frequency.toDouble();
    double pfactor = pf.toDouble();
    //Blynk.virtualWrite(V3, volt);
    Blynk.virtualWrite(V4, amp);
    Blynk.virtualWrite(V5, watt);
    //Blynk.virtualWrite(V6, hartz);
    Blynk.virtualWrite(V7, pfactor);
    // clear the string:
    inputString = "";
    stringComplete = false;
  }

}
void serialEvent() {
  while (Serial.available()) {
    // get the new byte:
    char inChar = (char)Serial.read();
    // if the incoming character is a newline, set a flag
    // so the main loop can do something about it:
    if (inChar == '\n') {
      stringComplete = true;
    }
    else
    // add it to the inputString:
      inputString += inChar;
  }
}
```