PROJECT REPORT

ON

**IOT BASED SOLAR IRRIGATION SYSTEM**

Submitted In Partial Fulfilment of the requirement for the award of

Degree of **Bachelor of Technology**

In

**Instrumentation And Control Engineering**

By

**AKHIL PAL (1901100220005),**

**AZMAN KHAN (1901100220019),**

Under the Guidance Of

Mr. NITESH KUMAR JAISWAL

(HOD, Department of Instrumentation and Control Engineering)

A picture containing sketch, drawing, text, symbol

Description automatically generated

**DEPARTMENTOF INSTRUMENTATION AND CONTROLENGINEERING**

### **Institute of Engineering and Rural Technology,**

### **Prayagraj**

### Affiliated to

**Dr. A.P.J. Abdul Kalam Technical University, Lucknow**

**(MAY-2023**

# CERTIFICATE

This is certified that the project entitled "**IOT Based Solar Irrigation System**" submitted by Akhil Pal (1901100220005), Azman Khan (1901100220019),of Instrumentation and Control Engineering in B.Tech. has been completed successfully.

This is partial fulfillment of the requirements of B. Tech. in **Instrumentation and Control Engineering** under **Dr. A.P.J Abdul Kalam Technical University, Lucknow U.P.**

I wish her/him success in all future endeavors.

**Mr. NITESH KUMAR JAISWAL**

(HOD, Department of Instrumentation and Control Engineering)

#### ACKNOWLEDGEMENT

We would like to express my deep and sincere gratitude to Mr. Sunil Kumar, Assistant Professor Department of Instrumentation and Control Engineering for his unflagging support and continuous encouragement throughout the project work. Without his guidance and persistent help this report would not have been possible.

We are highly in debated to our head of department **Mr. Nitesh Kumar Jaiswal** for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express our gratitude to our parents & members of (**IERT, PRAYAGRAJ**) for their kind co-operation and encouragement which helped us incompletion of this project.

Our thanks and appreciation also goes to our college developing the project and people who have willingly helped us out with their abilities.

|  |  |
| --- | --- |
| **Akhil Pal** | **Azman Khan** |

# ABSTRACT

The project is all about innovative IoT-based solar irrigation system that merges IoT technology and solar energy to revolutionize traditional irrigation methods. The system enables real-time monitoring and control of irrigation processes, optimizing water utilization, energy efficiency, and crop yields.

The system comprises solar panels, sensors, actuators, communication devices, and a centralized control system. Solar panels harness renewable solar energy, while sensors measure crucial environmental factors like soil moisture, temperature, and ambient light. This data is wirelessly transmitted to the central control system, which employs advanced algorithms to make informed decisions on irrigation scheduling and resource allocation.

By integrating solar energy and IoT connectivity, the IoT-based solar irrigation system represents a sustainable approach to agriculture. It promotes efficient water management, reduces energy consumption, and enhances crop productivity. This abstract is highlight the system's potential to revolutionize irrigation practices and contribute to the advancement of sustainable agricultural methods.

**LIST OF FIGURE**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Description** | **Page No.** |
| Figure 3.1 | Arduino uno | 15 |
| Figure 3.2 | Servo motor | 19 |
| Figure 3.3 | Solenoid valve | 20 |
| Figure 3.4 | Labeling of Solenoid valve | 21 |
| Figure 3.5 | Capacitive soil moisture sensor | 23 |
| Figure 3.7 | Pump | 24 |
| Figure 3.8 | Relay module | 25 |
| Figure 3.9 | Solar panel | 27 |
| Figure 3.10 | LDR | 29 |
| Figure 3.11 | Wi-fi module | 30 |
| Figure 3.12 | ESP8266-01Module Pin out and Description | 32 |
| Figure 3.15 | Battery control PCB | 33 |
| Figure 3.3.1 | USB Connection to Arduino and laptop | 34 |
| Figure 3.3.2 | The Device is detected | 35 |
| Figure 3.3.3 | Port selection | 36 |
| Figure 3.3.4 | Code uploading | 36 |
| Figure 3.3.5 | Hardware model | 45 |

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Chapter No.** | **Page No.** |
| 1. | CERTIFICATE | 02 |
| 2. | ACKNOWLEDGEMENT | 03 |
| 3. | ABSTRACT | 04 |
| 4. | LIST OF FIGURES | 05 |
| 5. | CHAPTER1: INTRODUCTION  1.1 Overview | 08-09 |
| 6. | CHAPTER 2: LITERATURE REVIEW  2.1 Overview | 10-11 |
| 7. | CHAPTER3:PROPOSEDWORK   * 1. Overview   2. Hardware Descriptions   3. Software Descriptions   4. Circuit Diagram   5. Hardware Model | 12-45 |
| 8. | CHAPTER 4: RESULTS AND DISCUSSION  4.1 Results  4.2 Discussion | 46-48 |
| 9. | CHAPTER 5: CONCLUSIONS AND FUTURE SCOPE  5.1 Conclusion  5.2 Future Scope | 49-51 |
| 10. | REFERENCE | 52 |

**CHAPTER: 1**

**INTRODUCTION**

#### OVERVIEW

An Internet of Things (IOT) based solar irrigation system is an innovative and sustainable approach to irrigation that utilizes solar power and IOT technology to improve the efficiency of the irrigation process.

This type of system typically consists of a network of sensors, actuators, and controllers’ that are connected to the internet and used to monitor and control the irrigation process. The sensors measure various parameters, such as soil moisture, temperature, humidity, and light intensity, while the controllers use this data to determine the optimal irrigation schedule and volume. In a solar irrigation system, solar panels are used to generate electricity, which is then used to power the various components of the system. This makes the system eco-friendly and sustainable, as it relies on renewable energy sources.

One of the key benefits of an IOT-based solar irrigation system is its ability to optimize water usage, which can help to conserve water resources and reduce costs. The system can also be controlled remotely via a mobile app or web interface, making it easy to monitor and adjust irrigation schedules from anywhere. Overall, an IOT-based solar irrigation system can provide a more efficient and sustainable approaches to irrigation that can help to increase crop yields and improve source management.

* 1. **Why IOT Based Solar Irrigation System**

An IOT-based solar irrigation system offers several advantages over traditional irrigation systems. IOT sensors and controllers can gather data on soil moisture, weather conditions and other factors that affect plant growth and use this information to optimize irrigation schedules and water usage. This can help farmers to use water resources more efficiently, reduce water waste and ultimately save money on their water bills. With an IOT-based system, farmers can remotely monitor and control the irrigation process from a smart-phone, tablet, or computer. This enables them to make adjustments to the irrigation schedule in real-time, respond quickly to changes in weather conditions, and reduce the need for on-site visits. Solar panels are used to generate electricity for the system, reducing the need for traditional energy sources and lowering the system's carbon footprint. Additionally, by optimizing water usage, the system can help to conserve water resources and promote sustainable agriculture practices. By providing plants with the optimal amount of water and nutrients, an IOT- based solar irrigation system can help to improve crop yields and quality. This can help farmers to maximize their profits and meet the increasing demand for food production.

**CHAPTER: 2**

**LITERATURE WORK**

**2.1 OVERVIEW**

IoT-based Solar Powered Smart Irrigation System for Precision Agriculture by S. V. Mane and S. S. Gaikwad. In this paper, the authors proposed an IoT-based solar-powered smart irrigation system that can automatically control the water supply to crops based on the soil moisture level and weather conditions.

Smart Irrigation System using IoT and Solar Power by R. S. S. Bharathi and S. Suganya. This paper presents a smart irrigation system that uses IoT and solar power to optimize water usage in agriculture. The system can monitor soil moisture, temperature, and humidity, and control the water flow to the crops based on these parameters.

An IoT-Based Solar-Powered Automated Irrigation System for Sustainable Agriculture by N. Kumar ravel and P. Balasubramanian. This paper describes an IoT-based solar-powered automated irrigation system that can control the water supply to crops based on the real-time weather conditions and soil moisture level. The system also include mobile app that allows farmers to monitor and control the irrigation process remotely.

Design and Development of IoT-based Solar Powered Irrigation System for Precision Farming by S. K. Das and S. K. Nanda. This paper presents a design and development of an IoT-based solar-powered irrigation system that can optimize water usage in agriculture by monitoring soil moisture and weather conditions. The system also includes a mobile app that allows farmers to receive real-time updates on the irrigation process.

**CHAPTER: 3**

**WORKING METHODOLOGY**

\

**3.1) OVERVIEW**

An IoT-based solar irrigation system combines Internet of Things (IOT) technology with solar power to efficiently manage and control irrigation in agricultural fields. It utilizes various sensors, connectivity, and automation to monitor and optimize water usage, resulting in improved crop yield, reduced water waste, and energy sustainability. Here's an overview of how it works:

* **Solar Power Generation**

The system incorporates solar panels to harness sunlight and convert it into electrical energy. This renewable energy source provides power for the entire irrigation system, reducing reliance on traditional electricity grid sand minimizing operational costs.

###### **Sensor Network:**

###### Soil moisture sensors are deployed throughout the field to measure the moisture content in the soil .The sensors can be placed at different depths to obtain accurate data about the water requirements of crops. Other sensors, such as temperature and humidity sensors, may also be included to collect environmental data.

###### **Data Collection and Transmission:**

The sensors gather data on soil moisture, temperature, and humidity, among other parameters. This data is then transmitted to a central control unit or gateway using wireless communication protocols like Wi-Fi, Bluetooth, or Zig bee. The gateway acts as a bridge between the sensors and the cloud-based application.

###### **Cloud-based Application:**

The collected data is sent to a cloud-based application or platform that processes and analyzes the information. The application uses algorithms and predefined thresholds to determine the optimal irrigation schedule and water requirements based on crop type, weather conditions, and soil moisture levels.

###### **Automated Control:**

Once the optimal irrigation parameters are determined, the system automatically controls the irrigation process. It activate sand deactivates the water pumps, valves, and sprinklers based on the instructions received from the cloud-based application. This automation ensures precise water delivery to crops, avoiding under rover-irrigation.

###### 

###### **Remote Monitoring and Control:**

Farmers and agricultural experts can remotely monitor and control the irrigation system through a mobile application web interface. They can access real-time data, receive alerts and notifications, and make adjustments to the irrigation settings as needed, providing convenience and flexibility in managing the system.

###### **Data Analysis and Insights:**

The collected data can be analyzed over time to generate valuable insights. Patterns and trends in soil moisture levels, weather conditions, and crop growth can be identified, helping farmers make informed decisions about irrigation strategies, crop selection, and resource allocation.

Overall, an IoT-based solar irrigation system optimizes water usage, reduces manual labor, enhances crop productivity, and promotes sustainable farming practices by leveraging solar energy and IoT technologies for efficient irrigation management.

###### **3.2) REQUIRED COMPONENTS**

1. Arduino Uno
2. Servo Motor



3. IR sensor



4. Capacitive Moisture Sensor

5. Motor



6. Pump



7. Solar panel



8. Battery

9. Water pipe

10. Solenoid valve

11. Relay

12. Battery charge protection circuit

13. Wi-Fi Module (esp. 882266)

**3.3) SOFTWARE COMPONENTS**

1. Arduino compiler
2. Arduino Code

**3.2.1) Arduino UNO**

The Arduino UNO is a standard board of Arduino. Here UNO means 'one' in Italian .It was named as UNO to label the ﬁrst release of Arduino Software. It was also the ﬁrst USB board released by Arduino. It is considered as the powerful board used in various projects. Arduino .cc developed the Arduino UNO board.

Arduino UNO is based on an ATmega328P micro controller .It is easy to use compared to other boards, such as the Arduino Mega board, etc. The board consists of digital and analog Input/Output pins (I/O), shields, and other circuits.

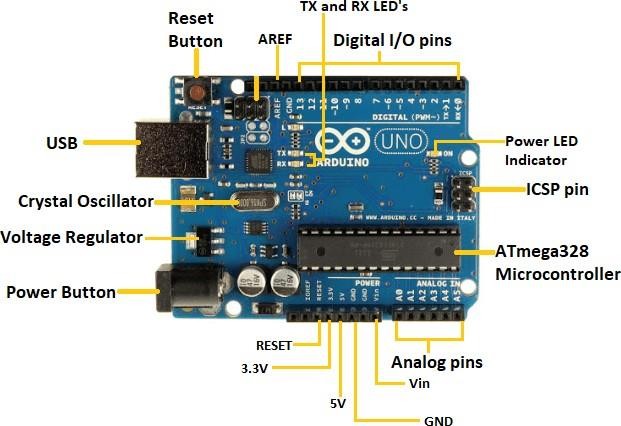
The Arduino UNO includes 6 analog pin inputs, 14 digital pins, a USB connector, a power jack, and an ICSP (In-Circuit Serial Programming) header. It is programmed based on IDE, which stands for Integrated Development Environment. It can run on both online and oﬄine platforms.

The IDE is common to all available boards of Arduino. The Arduino board is shown below:



**Figure3.1–Arduino Uno**

The components of Arduino UNO board are shown below:



**Figure 3.2 Components of Arduino Uno**

**ATmega328 Microcontroller**-

It is a single chip Micro controller of the Atmel family .The processor code inside it is of 8-bit. It combines Memory (SRAM, EEPROM, and Flash), Analog to Digital Converter, SPI serial ports, I/O lines, registers, timer, external and internal interrupts, and oscillator.

**ICSP pin** –

The In-Circuit Serial Programming pin allows the user to program using the ﬁrm ware of the Arduino board.

**Power LED Indicator**-

The ON status of LED shows the power is activated. When the power is OFF, the LED will not light up.

**Digital I/O pins**-

The digital pins have the value HIGH or LOW. The pins numbered from D0 to D13 are digital pin.

**TX and RX LED's** –

The successful ﬂow of data is represented by the lighting of these LED's.

**AREF-**

The Analog Reference (AREF) pin is used to feed a reference voltage to the Arduino UNO board from the external power supply.

**Reset button**-

It is used to add a Reset button to the connection.

**USB**-

It allows the board to connect to the computer. It is essential for the programming of the Arduino UNO board.

**Crystal Oscillator**-

The Crystal oscillator has a frequency of 16 MHz, which makes the Arduino UNO a powerful board.

**Voltage Regulator**-

The voltage regulator converts the input voltage to 5V.

**GND**-

Ground pins. The ground pin acts as a pin with zero voltage.

**Vin**-

It is the input voltage.

**Analog Pins**-

The pins numbered from A0 to A5 are analog pins. The function of Analog pins is to read the analog sensor used in the connection. It can also act as GPIO (General Purpose Input Output) PIN

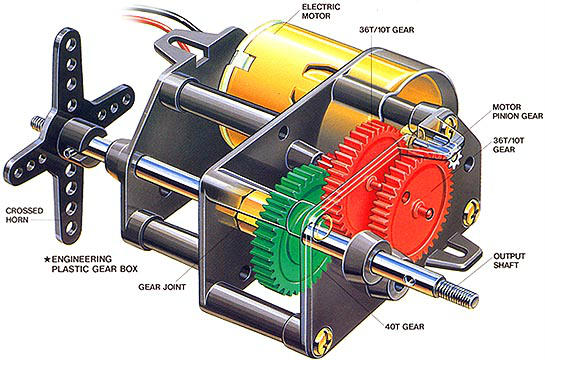
# 3.2.2) Servo Motor

A **servo motor** is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision. If you want to rotate an object at some specific angle or distance, then you use a servo motor. It is just made up of a simple motor which runs through a servo mechanism. If motor is powered by a DC power supply then it is called DC servo motor, and if it is AC-powered motor then it is called AC servomotor. For this tutorial, we will be discussing only about the DC servo motorworking. Apart from these major classifications, there are many other types of servo motors based on the type of gear arrangement and operating characteristics. A servo motor usually comes with a gear arrangement that allows us to get a very high torque servo motor in small and light weight packages. Due to these features, they are being used in many applications like toy car, R Helicopters and planes, Robotics etc.

Servo motors are rated in kg/cm (kilogram per centimeter) most hobby servo motors are rated at3kg/cm or 6kg/cm or12kg/cm. This kg/cm tells you how much weight your servo motor can lift at a particular distance. For example: A 6kg/cm Servo motor should be able to lift 6kg if the load is suspended 1cm away from the motors shaft, the greater the distance the lesser the weight carrying capacity. The position of a servo motor is decided by electrical pulse and its circuitry is placed beside the motor.

## Servo Motor Working Principle:

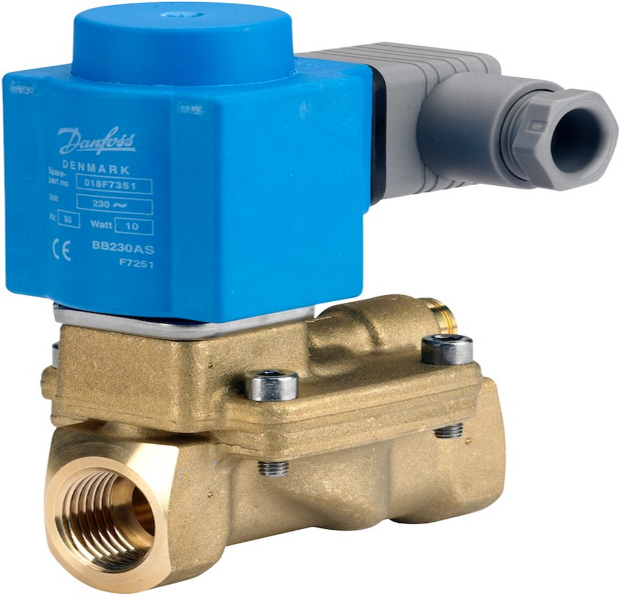
A servo consists of a Motor (DC or AC), a potentiometer, gear assembly, and a controlling circuit. First of all, we use gear assembly to reduce RPM and to increase torque of the motor. Say at initial position of servo motor shaft, the position of the potentiometer knob is such that there is no electrical signal generated at the output port of the potentiometer. Now an electrical signal is given to another input terminal of the error detect or amplifier. Now the difference between these two signals, one comes from the potentiometer and another comes from other sources, will be processed in a feedback mechanism and output will be provided in terms of error signal. This error signal acts as the input for motor and motor starts rotating. Now motor shaft is connected with the potentiometer and as the motor rotates so the potentiometer and it will generate a signal. So as the potentiometer’s angular position changes, its output feedback signal changes. After sometime the position of potentiometer reaches at a position that the output of potentiometer is same as external signal provided. At this condition, there will be no output signal from the amplifier to the motor input as there is no difference between external applied signal and the signal generated at potentiometer and in this situation motor stops rotating.



**Figure3.2–Servo moto**

**3.2.4) SOLENOID VALVE**

Electromagnetic solenoid valves (otherwise also known as solenoid valves) are valves controlled by electric current. They consist of two main parts - the valve body and the solenoid (coil). The solenoid is composed of a wound copper wire that surrounds a core with a movable closing plunger. The task of the coil is to create a magnetic field by means of a passing electric current, which then moves the piston and either opens or closes the valve. Solenoid valves thus use electric current to convert into linear motion.



**Figure3.3–Solenoid valve**

Solenoid valves are used for many purposes. There are suitable for liquid and gaseous media - for closing, opening, dosing, distribution or mixing in distribution systems. Classic applications include heating systems, irrigation, dishwashers and washing machines, refrigeration and air

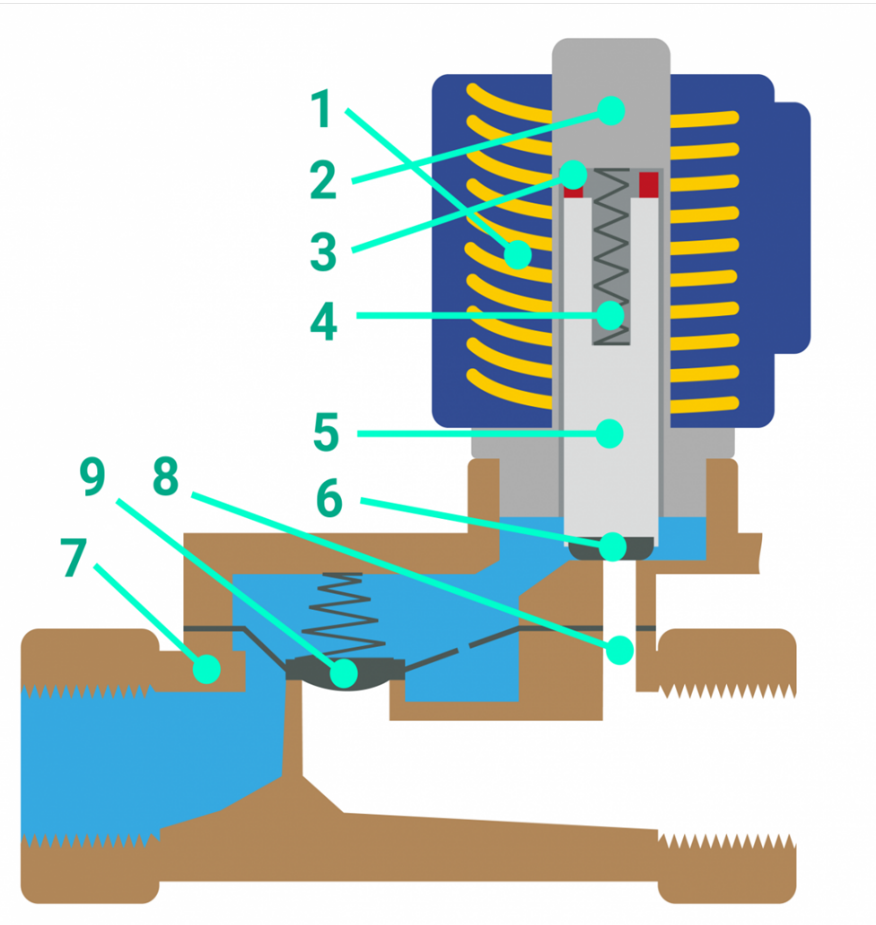
Conditioning systems, medicine, dentistry, industrial cleaning and water tanks.

Solenoid valves come in the common two-way, or even more complex three-way and multi-way constructions used for flow switching and mixing. Most often, valve bodies are made of brass,

stainless steel, aluminum or even plastic. Before purchasing always check that the material is suitable for the intended use and is compatible with the medium.

**VALVE DESIGN**

The valve consists of a valve body and a coil mounted on it. The inlet and outlet are fitted with connections so that it can be connected to the pipeline. Inside the valve body we can find a spring, plunger and seal or diaphragm.



**Figure3.4-Labeling of solenoid valve**

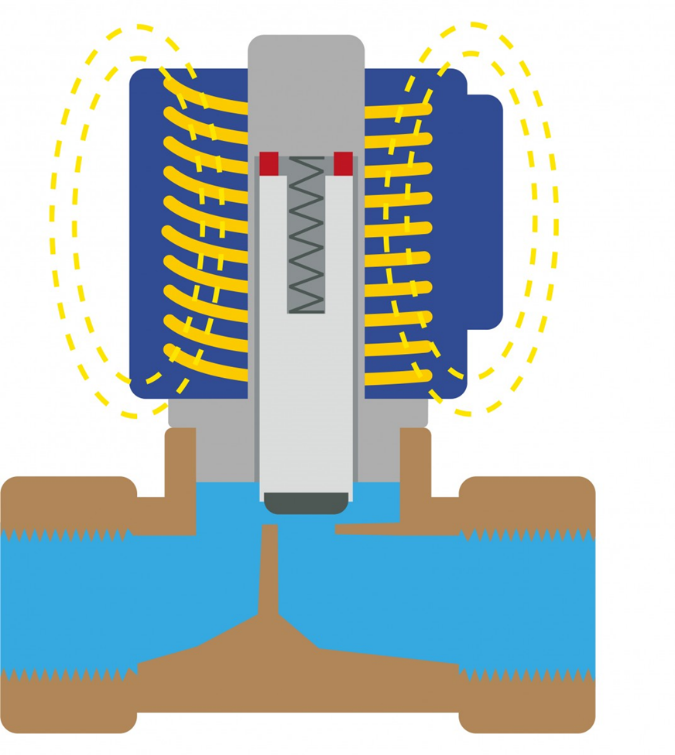
1. coil
2. armature
3. shielding ring
4. spring
5. plunger
6. seal
7. valve body
8. body orifice
9. diaphragm.

The two main categories of solenoids are NO and NC –

normally open and normally closed.

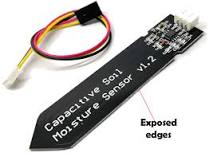
When an electric current is applied to the coil, a magnetic field is generated, the strength of which depends on the current, the number of windings of the wire and the material of the moving core, also referred to as a piston. The magnetic field moves this plunger and thus closes or opens the valve. Without power, the valve can be either closed or open.

With a normally closed valve, when powered the plunger is pulled up due to the magnetic field. The valve thus opens to allow the medium to flow. The force of the magnetic field raises the plunger against the spring, which in turn pushes it back down. When power gets interrupted, magnetic field disappears and the plunger spring compresses it back to its original position. The valve is thus closed. This is a more widely used variant for safety reasons in the events of a power failure.



A normally open valve allows the medium to flow without power. The plunger is pulled always out and the medium can flow through the valve. However, when powered, magnetic field pushes the plunger down and closes the valve. This variant is mainly used in applications where it is more energy efficient to have the valve open for a longer time.

**3.2.5) CAPACITIVE SOIL MOISTURE SENSOR**



**Figure3.5–Capacitive soil moisture sensor**

A soil moisture sensor measures soil moisture levels by capacitive sensing rather than resistive sensing like other sensors on the market. It is made of corrosion resistant material which gives it an excellent service life. Insertitintothesoilaroundyourplantsandimpressyourfriendswithreal-times oil moisture data! This module includes an on-board voltage regulator which gives it an operating voltage range of 3.3~5.5 V. It is perfect for low-voltage MCUs , both 3.3V and 5V. For compatibility with a Raspberry Pi it will need an ADC converter. This soil moisture sensor is compatible with our 3-pin "Gravity" interface, which can be directly connected to the Gravity I/O expansion shield.

**Specification**

* + OperatingVoltage:3.3~5.5VDC OutputVoltage:0~3.0VDC
  + OperatingCurrent:5mA
  + Interface:PH2.0-3P
  + Dimensions: 3.86 x0.905inches(L x W)
  + Weight:15g

**3.2.6) PUMP**

A fluid pump is a mechanical device designed to move fluids, such as liquids or gases, from one location to another. It is commonly used in various industries and applications where fluid transfer or circulation is required. Fluid pumps work by increasing the pressure of the fluid, allowing it to flow through pipes, hoses, or other conduits.

There are many different types of pumps, including centrifugal pumps, positive displacement pumps, and jet pumps, each with their own unique features and applications.



**Figure3.6-Pump**

**3.2.7) RELAY MODULE**



**Figure3.7–Relay module**

Relay modules are simply circuit boards that house one or more relays. They come in a variety of shapes and sizes, but are most commonly rectangular with 2, 4, or 8 relays mounted on them, sometimes even up to a 16 relays.

* + Relay modules contain other components than the relay unit. These include indicator LEDs, protection diodes, transistors, resistors, and other parts. But what is the module relay, which makes the bulk of the device? You may ask. Here are facts to note about it:
  + A relay is an electrical switch that can be used to control devices and systems that use higher voltages. In the case of module relay, the mechanism is typically an electromagnet.
  + The relay module input voltage is usually DC. However, the electrical load that are lay will control can be either AC or DC, but essentially within the limit levels that the relay is designed for.
  + A relay module is available in an array of input voltage ratings: It can be a 3.2V or 5V relay module for low power switching, or it can be a 12 or 24V relay module for heavy-duty systems.
  + The relay module information is normally printed on the surface of the device for ready reference.
  + This includes the input voltage rating, switch voltage, and current limit.

**Relay Module Working**

The typical relay module connection points include an input side that consists of 3 or 4 jumper pins, and an output side that has 3 screw terminals.

1. When the control signal is applied to the input side of the relay, it [**activates the**](https://study.com/academy/lesson/what-is-an-electromagnet-definition-uses-parts.html)[**electromagnet**](https://study.com/academy/lesson/what-is-an-electromagnet-definition-uses-parts.html)**,** which attracts an armature.
2. This in turn closes the switch contacts on the output (high voltage) side, allowing electricity to flow and power the device or system that is connected to it.
3. To prevent fly back voltage from damaging the relay module circuit and the input device, a diode is often placed in parallel with the electromagnet coil. This diode is known as a [**fly back diode**](https://www.electronicsforu.com/technology-trends/learn-electronics/flyback-transformer-basics)**.** It allows current to flow in only one direction.
4. When a higher level of isolation is required, an opto coupler is used. An opto-isolated relay module has the photoelectric device on the input side, which is used to control the electromagnet’s switching action.
5. Relay modules are available with either normally open (NO) or normally closed (NC) switch configurations.
6. A NO switch is open when the electromagnet is not activated, and closed when it is activated.
7. An NC relay switch, on the other hand, remains closed by default, and only opens when the relay is activated.

**3.2.8) SOLAR PANEL**

Solar panels, also known as photovoltaic (PV) panels, are devices that convert sunlight into electricity. They are made up of multiple interconnected solar cells that absorb sunlight and generate direct current (DC) electrical energy. Solar panels are a key component of solar power systems and are used to harness renewable energy from the sun.



**Figure3.8 -Solar Panel**

Solar panels collect clean renewable energy in the form of sunlight and convert that light in to electricity which can then be used to provide power for electrical loads. Solar panels are comprised of several individual solar cells which are themselves composed of layers of silicon, phosphorous (which provides the negative charge), and boron (which provides the positive charge). Solar panels absorb the photons and in doing so initiate an electric current. The resulting energy generated from photons striking the surface of the solar panel allows electrons to be knocked out of their atomic orbits and released into the electric ﬁeld generated by the solar cells which then pull these free electrons into a directional current. This entire process is known as the Photovoltaic Effect. An average home has more than enough roof area for the necessary number of solar panels to produce enough solar electricity to supply all of its power needs excess electricity generated goes onto the main power grid, paying off in electricity use at night. In a well-balanced grid-connected conﬁguration, a solar array generates power during the day that is then used in the home at night. The AC current can be used to power loads in homes or commercial buildings, recreational vehicles and boats, remote cabins, cottages, or homes, remote traﬃc controls, telecommunications equipment, oil and gas ﬂow monitoring, RTU, SCADA, and much more.

**LDR**

LDR (Light Dependent Resistor) as the name states is a special type of resistor that works on the photoconductivity principle means that resistance changes according to the intensity of light. Its resistance decreases with an increase in the intensity of light. It is often used as a light sensor, light meter, Automatic streetlight, and in areas where we need to have light sensitivity. It is also called a Light Sensor.

LDR are usually available in 5mm, 8mm, 12mm, and 25mm dimensions.

It works on the principle of photoconductivity whenever the light falls on its photoconductive material, it absorbs its energy and the electrons of that photoconductive material in the valence band get excited and go to the conduction band and thus increasing the conductivity as per the increase in light intensity.

Also, the energy in incident light should be greater than the bandgap gap energy so that the electrons from the valence band got excited and go to the conduction band.

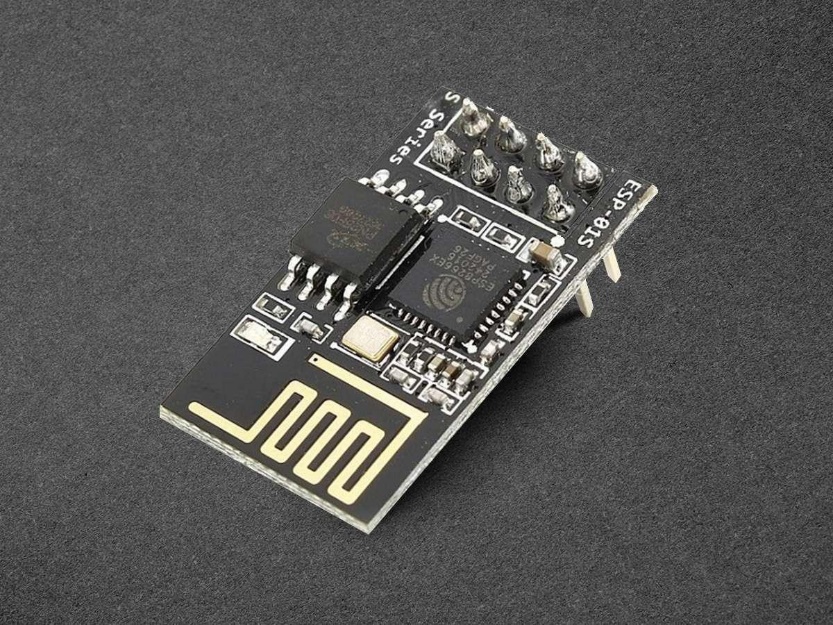
The LDR has the highest resistance in dark around 1012 Ohm and this resistance decreases with the increase in Light.

A picture containing circuit component, electronic component, passive circuit component, electronics

Description automatically generated

**Figure 3.9 LDR (Light dependent resistor)**

**3.2.10) Wi-Fi Module (ESP 8266)**



**Figure3.9–Wi-Fi module**

The ESP8266 is a highly popular Wi-Fi module widely used in Internet of Things (IoT) projects. It integrates a microcontroller and Wi-Fi capabilities into a single chip, providing a cost-effective and compact solution for wireless connectivity.

The ESP8266 WiFi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

ESP8266comeswithcapabilitiesof

* + 2.4GHzWi-Fi(802.11b/g/n, supporting WPA/WPA2),
  + general-purpose input/output (16GPIO),
  + Inter-Integrated Circuit (I²C) serial communication protocol,
  + analog-to-digital conversion (10-bitADC)
  + Serial Peripheral Interface (SPI) serial communication protocol,
  + I²S(Inter-IC Sound) interfaces with DMA (Direct Memory Access) (sharing pins with GPIO),
  + UART (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2)
  + pulse- width modulation (PWM).

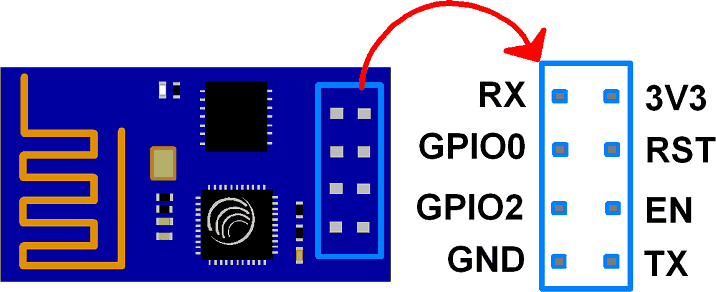
It employs a 32-bit RISC CPU based on the Ten silica X tensa L106 running at 80 MHz (or over clocked to 160 MHz). It has a 64 KB boot ROM, 64 KB instruction RAM and 96 KB data RAM. External ﬂash memory can be accessed through SPI.

ESP8266 module is low cost standalone wireless transceiver that can be used for end-point IoT developments.

To communicate with the ESP8266 module, microcontroller needs to use set of AT commands. Microcontroller communicates with ESP8266-01 module using UART having speciﬁed Baud rate.

There are many third-party manufacturers that produce different modules based on this chip. So, the module comes with different pin availability options like

* + ESP-01 comes with 8 pins (2 GPIO pins) – PCB trace antenna. (shown in above ﬁgure)
  + ESP-02 comes with 8 pins, (3 GPIO pins) – U-FL antenna connector.
  + ESP-03 comes with 14 pins, (7 GPIO pins) – Ceramic antenna.
  + ESP-04 comes with 14 pins, (7 GPIO pins) – No



### **Fig3.10-ESP8266-01ModulePinoutandDescription**

**Esp8266pinout**

**3V3**:-3.3VPowerPin.

**GND**:-Ground Pin.

**RST**:-Active Low Reset Pin.

**EN**:-Active High Enable Pin.

**TX**:-Serial Transmit Pin of UART.

**RX**:-Serial Receive Pin of UART.

**GPIO0 & GPIO2**: - General Purpose I/O Pins. These pins decide what mode (bootor normal) the module starts up in. It also decides whether the TX/RX pins are used for Programming the module or for serial I/O purpose.

To program the module using UART, Connect GPIO0 to ground and GPIO2 to VCC or leave it open. To use UART for normal Serial I/O leave both the pins open (neither VCC nor Ground).

For more information refer <http://esp8266.net/>

Now, before we start with ESP8266 interfacing, we need to update its ﬁrmware.

# Battery Charge Control Module

A Battery Charge Control Module (BCCM), also known as a Battery Management System (BMS), is a device or circuitry that manages and controls the charging process of rechargeable batteries. It ensures safe and efficient charging while protecting the battery from overcharging, over-discharging, and other potentially harmful conditions. Here is a detailed explanation of the components and functions typically found in a Battery Charge Control Module:

1. Voltage Monitoring: The BCCM continuously monitors the voltage of the battery during the charging process. It ensures that the charging voltage remains within safe limits and prevents overcharging, which can damage the battery.
2. Current Monitoring: The module also monitors the charging current flowing into the battery. It prevents excessive charging currents that can cause overheating and damage to the battery.
3. Temperature Monitoring: Battery temperature is an important parameter to monitor during charging. The BCCM includes temperature sensors to measure the battery's temperature and ensures that it stays within safe operating limits. If the temperature rises above a certain threshold, the BCCM may reduce the charging current or temporarily halt the charging process.
4. Charge Control Algorithm: The BCCM uses a charge control algorithm to regulate the charging process. This algorithm determines the optimal charging current and voltage based on the battery's state of charge, temperature, and other factors. It dynamically adjusts the charging parameters to maximize charging efficiency while ensuring battery safety.
5. Charge Stage Management: The BCCM manages the different stages of the charging process, including bulk charging, absorption charging, and float charging. Each stage has specific voltage and current requirements, and the BCCM controls the transitions between these stages based on the battery's needs.
6. Overcharge Protection: To prevent overcharging, the BCCM includes overcharge protection mechanisms. If the battery voltage exceeds a predefined threshold, the BCCM will take action to limit the charging current or terminate the charging process altogether.
7. Over-Discharge Protection: The BCCM also protects the battery from over-discharge, which can be detrimental to its health. It monitors the battery voltage and ensures that it does not drop below a certain threshold. If the voltage reaches a critical level, the BCCM may disconnect the load or initiate a recharge to prevent further discharge.



**Figure3.10 Battery charge control module**

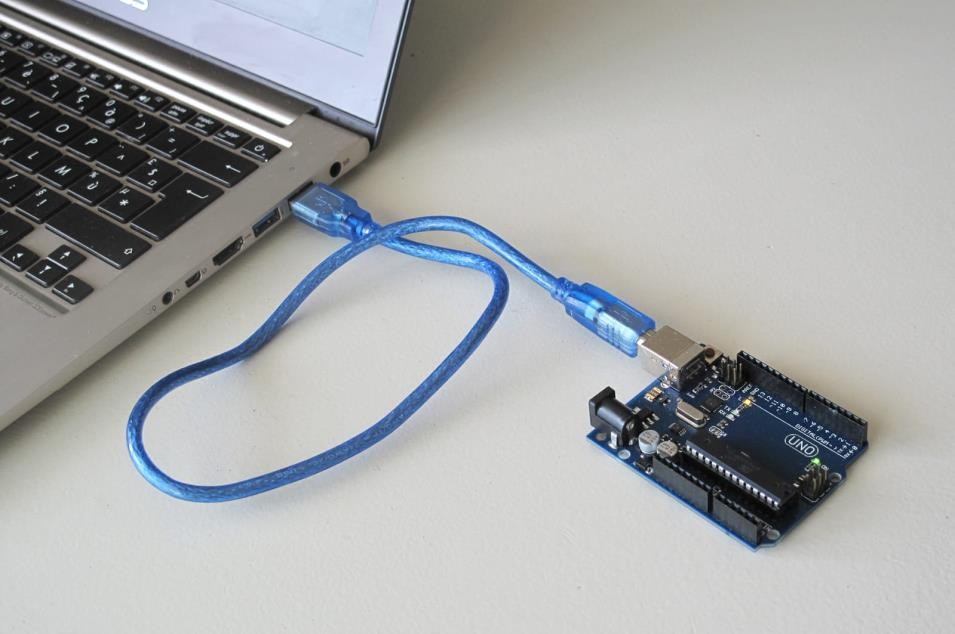
**3.3.1) SOFTWARE DESCRIPTIONS**

HOW TO START WITH ARDUINO UNO

Step 1: Download the Arduino IDE The latest version of the IDE can be downloaded from the official website.



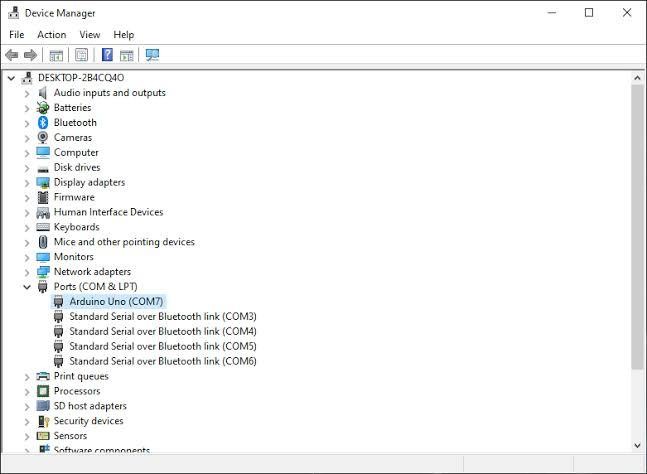
#### Step2: Connect the USB cable to the Arduino



**Figure3.3.2- USB Connection to Arduino and laptop**

#### Step3: Verify if the device is detected or not

Open device manager in your PC and expand the ports, you should see that the Arduino Uno is detected, if not you need to install the required drivers.



**Figure3.3.3- The Device is detected**

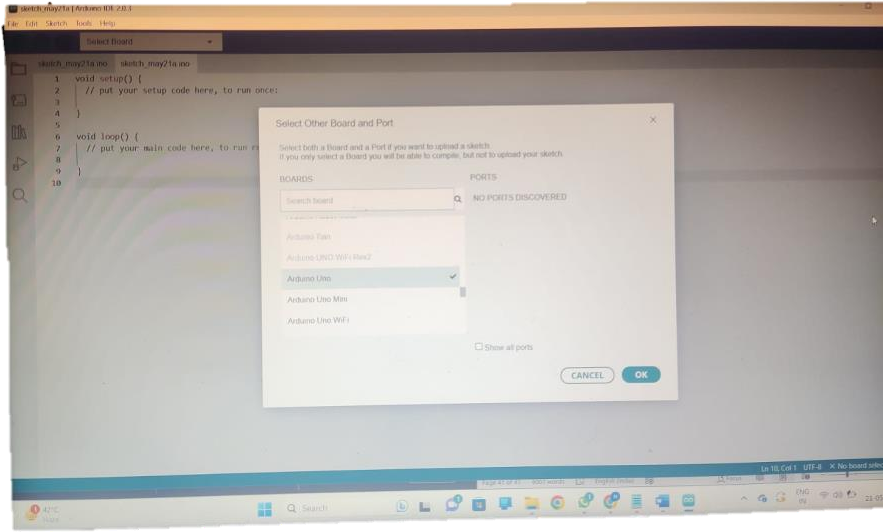
***Note***: Most of the Arduino boards come with a sketch already programmed in them to blink the built-in LED (for testing purposes). Therefore, you should see the amber-colored LED blinking once you connect the board to the PC.

#### Step4: Write your first sketch

Open the Arduino IDE

Right click the Select Board

Select "Arduino Uno" for the board and select the assigned COM port



**Figure3.3.4-Port Selection**

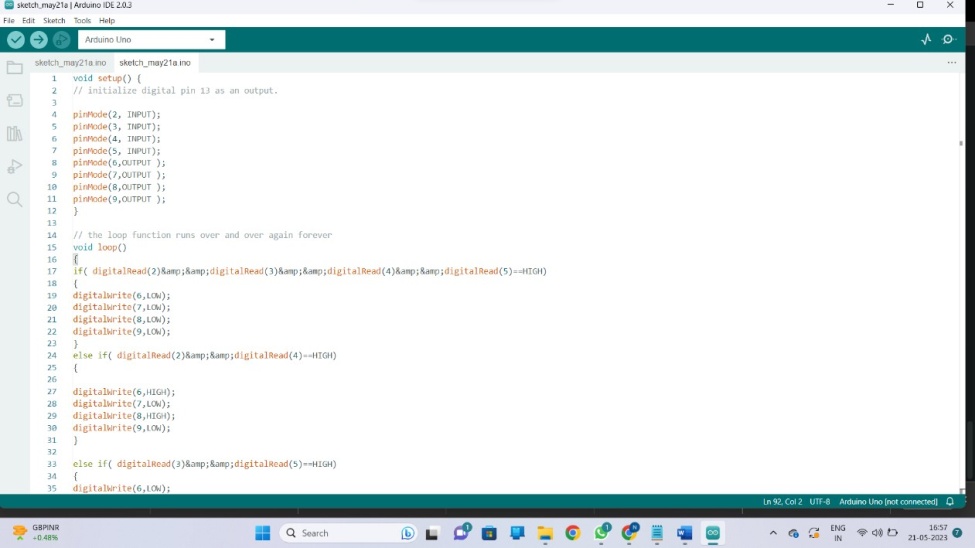
Enter the code in the editor

#### Step5: Compile your sketch

Hit the tick button (compile) in the top left corner of the IDE , you should see that the code is compiled successfully without any errors.

#### Step6: Upload your Code:

Hit the right arrow button (upload) next to the upload button. Now while the code uploads, you should see the LED's next to Tx and Rx blinking indicating data transfer between the board and the computer.



**Figure3.3.5 –Code uploading**

CODING

#define REMOTEXY\_MODE\_\_ESP8266\_HARDSERIAL\_POINT

#include <RemoteXY.h>

#include<Servo.h>

*// RemoteXY connection settings*

#define REMOTEXY\_SERIAL Serial

#define REMOTEXY\_SERIAL\_SPEED 115200

#define REMOTEXY\_WIFI\_SSID "RemoteXY"

#define REMOTEXY\_WIFI\_PASSWORD "12345678"

#define REMOTEXY\_SERVER\_PORT 6377

Servo servo1;

#define ldr\_sensor1 A0

#define ldr\_sensor2 A1

#define soil\_sensor1 A2

#define soil\_sensor2 A3

#define servoPin 6

#define pump 8

#define valve1 9

#define valve2 10

**int** serv =0;

**int** ldrValue1 =0;

**int** ldrValue2 =0;

**int** soilsensor\_val1 =0;

**int** soilsensor\_val2 =0;

**int** valAverage\_LDR1 =0;

**int** valAverage\_LDR2 =0;

**int** valAverage\_SOIL1 =0;

**int** valAverage\_SOIL2 =0;

**int**dryLevel=40;

**int**numAverage=10;

**unsignedlong** startTime1 =0;

**unsignedlong** startTime2 =0;

**unsignedlong** startTime3 =0;

**int** sensor\_duration1 =0;

**int** sensor\_duration2 =0;

**int**forced\_duration=3000;

**int** threshold =20;

**bool** isValve\_ON1 =false;

**bool** isValve\_ON2 =false;

**bool**forcedWatering=false;

**struct**Average{

**int**avg1;

**int**avg2;

};

**struct**Mapped{

**int** map\_val1;

**int** map\_val2;

};

*// RemoteXY configurate*

#pragma pack(push, 1)

**uint8\_t**RemoteXY\_CONF[] =*// 283 bytes*

{ 255,6,0,22,0,20,1,16,25,1,129,0,5,5,17,4,16,83,101,110,

115,111,114,32,49,0,7,20,5,13,13,4,31,26,2,129,0,18,14,2,

3,16,115,0,67,4,34,13,19,4,16,26,11,129,0,5,18,8,2,16,

87,97,116,101,114,105,110,103,0,129,0,5,20,8,2,16,68,117,114,97,

116,105,111,110,0,129,0,34,18,8,2,16,67,117,114,114,101,110,116,32,

77,111,105,115,116,117,114,101,0,129,0,5,26,17,4,16,83,101,110,115,

111,114,32,50,0,7,20,5,34,13,4,31,26,2,129,0,18,35,2,3,

16,115,0,67,4,34,34,19,4,16,26,11,129,0,5,39,8,2,16,87,

97,116,101,114,105,110,103,0,129,0,5,41,8,2,16,68,117,114,97,116,

105,111,110,0,129,0,34,39,15,2,16,67,117,114,114,101,110,116,32,77,

111,105,115,116,117,114,101,0,129,0,4,50,21,5,16,67,111,109,109,111,

110,0,3,131,4,60,15,6,2,26,129,0,5,67,10,2,16,67,114,111,

112,32,83,101,108,101,99,116,0,2,1,4,74,10,6,2,26,31,31,79,

78,0,79,70,70,0,129,0,4,81,10,2,16,87,97,116,101,114,32,78,

111,119,0 };

*// this structure defines all the variables and events of your control interface*

**struct** {

*// input variables*

**int16\_t** Sensor1\_duration; *// 32767.. +32767*

**int16\_t** Sensor2\_duration; *// 32767.. +32767*

**uint8\_t**crop\_select; *// =0 if select position A, =1 if position B, =2 if position C, ...*

**uint8\_t** switch\_1; *// =1 if switch ON and =0 if OFF*

*// output variables*

**char** Sensor1\_Display[11]; *// string UTF8 end zero*

**char** Sensor2\_Display[11]; *// string UTF8 end zero*

*// other variable*

**uint8\_t**connect\_flag; *// =1 if wire connected, else =0*

} RemoteXY;

#pragma pack(pop)

*/////////////////////////////////////////////*

*// END RemoteXY include //*

*/////////////////////////////////////////////*

**void**updateDisplay(**int** soilAvg\_map1, **int** soilAvg\_map2){

sprintf(RemoteXY.Sensor1\_Display, "%d", soilAvg\_map1);

sprintf(RemoteXY.Sensor2\_Display, "%d", soilAvg\_map2);

}

**void**durationUpdate(){

sensor\_duration1 = RemoteXY.Sensor1\_duration \*1000;

sensor\_duration2 = RemoteXY.Sensor2\_duration \*1000;

}

Average calculateAvg(**int** sensor\_1, **int** sensor\_2, **int**numAvg){

Average result;

result.avg1 =0;

result.avg2 =0;

**for**(**int** count =0; count <numAvg; count++){

**int** sensor\_value1 =analogRead(sensor\_1);

**int** sensor\_value2 =analogRead(sensor\_2);

result.avg1 = result.avg1 + sensor\_value1;

result.avg2 = result.avg2 + sensor\_value2;

}

result.avg1 /=numAvg;

result.avg2 /=numAvg;

**return**result;

}

**void**calculateServoPosition(**int** sensor\_1, **int** sensor\_2, **int**numAvg, **int** threshold){

Average average\_LDR=calculateAvg(sensor\_1, sensor\_2, numAvg);

valAverage\_LDR1 = average\_LDR.avg1;

valAverage\_LDR2 = average\_LDR.avg2;

**if**(valAverage\_LDR1 > valAverage\_LDR2 + threshold)

{

**if**(serv <180){

serv +=1;

servo1.write(serv);

}

}

**elseif**(valAverage\_LDR2 > valAverage\_LDR1 + threshold)

{

**if**(serv >0){

serv -=1;

servo1.write(serv);

}

}

}

Mapped calculateMoistureLevel(**int** sensor\_1, **int** sensor\_2, **int**numAvg){

Mapped mapped;

Average average\_MOISTURE=calculateAvg(sensor\_1, sensor\_2, numAvg);

valAverage\_SOIL1 = average\_MOISTURE.avg1;

valAverage\_SOIL2 = average\_MOISTURE.avg2;

mapped.map\_val1 =map(valAverage\_SOIL1, 385, 840, 100, 0);

mapped.map\_val2 =map(valAverage\_SOIL2, 385, 840, 100, 0);

updateDisplay(mapped.map\_val1, mapped.map\_val2); *// Update the Moisture Values on the App*

**return**mapped;

}

**void**watering(**int**mappedValue, **int** valve, **int**on\_duration, **unsignedlong**\*startTime, **bool**\*isON){

**if**(mappedValue<dryLevel){

**if**(!\*isON){

digitalWrite(valve, LOW); *// Turn on the valve Relay is Active LOW*

digitalWrite(pump, LOW); *// Turn on the pump Relay is Active LOW*

\*startTime=millis() +on\_duration; *// Set the new start time based on the current time + duration*

\*isON=true;

}

**if** (\*isON&& (millis() >=\*startTime)) {

digitalWrite(valve, HIGH);

digitalWrite(pump, HIGH);

\*isON=false;

\*startTime=millis(); *// Reset the starting time for the next interval*

}

}

}

**void**forceWatering(**int** valve, **int**on\_duration, **unsignedlong**\*startTime, **bool**\*isON){

**if**(!\*isON){

digitalWrite(valve, LOW); *// Turn on the valve Relay is Active LOW*

digitalWrite(pump, LOW); *// Turn on the pump Relay is Active LOW*

\*startTime=millis() +on\_duration; *// Set the new start time based on the current time + duration*

\*isON=true;

}

**if** (\*isON&& (millis() >=\*startTime)) {

digitalWrite(valve, HIGH);

digitalWrite(pump, HIGH);

\*isON=false; *// Reset the forcedWatering flag after completing the forced watering*

\*startTime=millis(); *// Reset the starting time for the next interval*

RemoteXY.switch\_1 =0;

}

}

**void**setup() {

pinMode(pump ,OUTPUT);

pinMode(valve1,OUTPUT);

pinMode(valve2,OUTPUT);

servo1.attach(servoPin);

digitalWrite(valve1, HIGH);

digitalWrite(valve2, HIGH);

digitalWrite(pump, HIGH);

sensor\_duration1 = RemoteXY.Sensor1\_duration \*1000;

sensor\_duration2 = RemoteXY.Sensor2\_duration \*1000;

}

**void**loop() {

RemoteXY\_Handler ();

**switch** (RemoteXY.crop\_select) {

**case**1:

dryLevel=65;

**break**;

**case**2:

dryLevel=50;

**break**;

**case**3:

dryLevel=35;

**break**;

default:

dryLevel=50;

**break**;

}

calculateServoPosition(ldr\_sensor1, ldr\_sensor2, numAverage, threshold);

Mapped mapped=calculateMoistureLevel(soil\_sensor1, soil\_sensor2, numAverage);

**if** (RemoteXY.switch\_1 ==1) {

forceWatering(valve1, forced\_duration, &startTime3, &forcedWatering);

}

watering(mapped.map\_val1, valve1, sensor\_duration1, &startTime1, &isValve\_ON1);

watering(mapped.map\_val2, valve2, sensor\_duration2, &startTime2, &isValve\_ON2);

durationUpdate();

}

**3.4) WORKING**

* Sensing Sunlight: The project starts by using an LDR sensor to detect the sunlight intensity. The LDR sensor measures the amount of light falling on it and provides an analog signal to the Arduino.
* Solar Tracking: Based on the LDR sensor's input, the Arduino controls a servo motor to adjust the position of the solar panel. The solar panel is single-axis, meaning it moves along one axis to face the sun and maximize its exposure to sunlight.
* Monitoring Soil Moisture: The Arduino reads the analog signals from the two capacitive soil moisture sensors. These sensors measure the moisture level in the soil and provide corresponding analog signals to the Arduino.
* Determining Moisture Threshold: The Arduino compares the moisture readings from the sensors with a predefined moisture level threshold. This threshold is set to determine when the soil moisture is below the desired level and requires irrigation.
* Activating the Pump: If any of the capacitive soil moisture sensors indicate a moisture level below the threshold, the Arduino triggers the pump. The pump is activated to provide water for irrigation.
* Controlling Solenoid Valves: The Arduino also controls two solenoid valves connected to the water supply lines. These valves allow or restrict the water flow to specific areas or plants. The Arduino activates the appropriate solenoid valves based on the moisture readings from the sensors that indicate the areas needing irrigation.
* ESP8266 Wi-3Fi Communication: The Arduino communicates with the ESP8266 WiFi module using the Remote XY library. The ESP8266 module establishes a connection to the internet and creates a wireless network. This allows users to connect to the system using their smartphones.
* Displaying Moisture Levels: Users can access the system through a mobile app. Arduino sends the current soil moisture levels to the ESP8266 module, which then displays the information on the user's phone. Users can remotely monitor the moisture levels of different areas or plants in real-time.
* Continuous Monitoring and Control: The system continuously monitors the sunlight intensity and soil moisture levels. The Arduino periodically reads the sensor values, adjusts the solar panel's position, and activates the pump and solenoid valves as necessary to maintain the desired soil moisture levels.

**3.4) HARDWARE MODEL**

A solar panel with wires and wires

Description automatically generated with low confidence

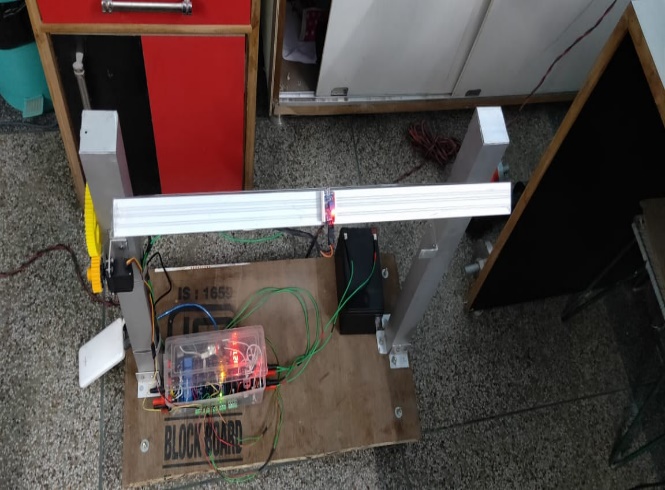
. **Fig3.4.1 Hardware model of project**

**CHAPTER: 4**

**RESULTS AND DISCUSSION**

###### **RESULT**







###### **DISCUSSIONS**

**Economic Benefits**:

The IOT-based solar irrigation system offers significant economic benefits to farmers. The optimized water usage reduces water costs, while the integration of solar energy lowers energy expenses. Additionally, the improved crop yield and quality result in higher profits. However, it's important to note that the initial setup costs and maintenance requirements of the system should be carefully considered for a comprehensive economic analysis.

###### **Environmental Impact**:

The project contributes to environmental sustainability by minimizing water wastage, reducing reliance on non-renewable energy sources, and decreasing greenhouse gas emissions. The use of solar power aligns with renewable energy goals and reduces the carbon footprint associated with traditional irrigation methods. The project's positive environmental impact makes it an attractive option for promoting sustainable agriculture practices.

###### **Challenges and Limitations:**

While the IoT-based solar irrigation system brings numerous advantages, there are challenges to consider. Connectivity issues, power interruptions, and sensor accuracy can affect system performance. Additionally, the initial investment for infrastructure and sensor deployment might be a barrier for small-scale farmers. Ensuring proper training and technical support can help overcome these challenges.

###### **Future Enhancements:**

The project opens avenues for future enhancements. Integration with advanced data analytics and machine learning algorithms can further optimize irrigation schedules, taking in to account additional parameters such as crop growth stages, soil quality, and evapotranspiration rates. Collaborations with weather services and local agricultural institutions can improve the accuracy of weather forecasts and support decision-making processes.

**CHAPTER: 5**

**CONCLUSION AND FUTURE SCOPE**

**5.1 CONCLUSION:**

The IOT-based solar irrigation system project showcases the potential for integrating IOT technology and solar power in the field of agriculture. By effectively utilizing real- time data, connectivity, and renewable energy sources, the project has demonstrated significant improvements in water efficiency, energy efficiency, and crop yield.

The system's ability to monitor soil moisture levels, weather conditions, and adjust irrigation schedules accordingly has resulted in reduced water wastage and optimized water usage. This not only conserves water resources but also leads to cost savings for farmers. Furthermore, the integration of solar power has reduced dependence on conventional energy sources, resulting in lower energy costs and a reduced carbon footprint.

The project's scalability and flexibility have made it adaptable to different field sizes and crop types. The wireless connectivity and remote monitoring capabilities have provided convenience and ease of management, empowering farmers to monitor and control irrigation processes from anywhere using mobile devices.

While the project offers economic benefits and contributes to environmental sustainability, there are challenges to consider, such as connectivity issues, sensor accuracy, and initial investment costs. Overcoming these challenges will require proper training, technical support, and continuous improvement of the system.

Advanced data analytics and machine learning algorithms can further optimize irrigation schedules, considering additional parameters for precise decision- making. Collaborations with weather services and agricultural institutions can improve the accuracy of weather forecasts, enhancing the system's performance.

In conclusion, the IoT-based solar irrigation system project demonstrates the potential to revolutionize irrigation practices, making them more efficient, sustainable, and economically viable. By harnessing the power of IoT and solar energy, this project paves the way for a modern and environmentally conscious approach to agriculture, benefiting farmers, the environment, and the agricultural industry.

**5.2 FUTURE SCOPE:**

The future scope of the IoT-based solar irrigation system project is promising, with several potential areas for development and improvement. Here are some key areas of future scope:

•Advanced Data Analytics:

Integrating advanced data analytics techniques and machine learning algorithms can enhance the system's capabilities. By analyzing historical and real-time data, the system can learn and predict optimal irrigation schedules based on crop growth stages, soil conditions, and climate patterns. This would enable even more precise and efficient water management.

•Sensor Technology Advancements:

Continued advancements in sensor technology can improve the accuracy and reliability of data collected by the system. New sensors with enhanced capabilities, such as multi- spectral imaging or nutrient level monitoring; can provide additional insights into plant health and nutrient requirements. Integration of these sensors can further optimize irrigation decisions and improve overall crop productivity.

•Integration with Smart Farming Systems:

The IOT-based solar irrigation system can be integrated with other smart farming systems and technologies. For example, combining it with automated pest control systems or crop monitoring drones can provide a comprehensive and integrated approach to agriculture. This integration would allow farmers to monitor and control multiple aspects of their farming operations through a single platform.

•Cloud-Based Monitoring and Decision Support:

The system can benefit from cloud-based platforms that offer centralized data storage, analysis, and decision support. Cloud-based solutions enable real-time monitoring and remote management, allowing farmers to access and control their irrigation systems from anywhere. Additionally, cloud-based platforms can provide farmers with valuable insights and recommendations based on aggregated data from multiple farms, improving decision-making and resource allocation.

• Collaborative Networks and Data Sharing:

Creating collaborative networks among farmers, agricultural experts, and research institutions can foster knowledge sharing and collaboration. By sharing data, best practices, and research findings, stakeholders can collectively work towards improving irrigation techniques, optimizing water usage, and addressing agricultural challenges. Such networks can also facilitate the development of standardized protocols and guidelines for implementing IOT-based solar irrigation systems.

Overall, the future scope of the IOT-based solar irrigation system project lies in leveraging emerging technologies, improving data analysis capabilities, and fostering collaboration. By continuously refining and expanding the system's functionalities, it has the potential to revolutionize irrigation practices, enhance agricultural productivity, and contribute to sustainable and efficient farming practices.

**REFERENCES**

* + 1. "Solar Powered Smart Irrigation System" Authors: S. Harishankar, R. Sathish Kumar, S. Sudarshan, K. Vignesh, T. Viv Ekanath Conference: 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)(2016)
    2. "Solar energy based automatic irrigation system" Authors: S. K. Nanda, S. Panda, S. K. Mohapatra Journal: International Journal of Engineering, Science and Technology Year: 2011
    3. "Design and Implementation of Solar Powered Automatic Irrigation System Using Microcontroller “Authors: S. K. Nanda, S. Panda, S. K. Mohapatra Conference: 2011 International Conference on Energy, Automation and Signal (ICEAS) Year: 2011
    4. Design and Implementation of Solar Powered Automatic Irrigation System Using Arduino" Authors: S. K. Nanda, S. Panda, S. K. Mohapatra Journal: International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Year: 2014
    5. Neelima Measured the moisture level of soil and the controlled water pump on its basis and sent the data through Wi-Fi, International Journal of Innovative Research in Computer and Communication Engineering.
    6. Himani Mahajan, used solar panel to operate the micro controller and measured moisture and temperature to turn the water pump on and off, International Journal for Research in Applied Science & Engineering Technology (IJRASET).
    7. Srishti Rawal Made an online application for data retrieving and managing actions, International Journal of Computer Applications.
    8. R. Ram Priya, N.Surya, Introduced sprinklers in the field along with Arduino board and moisture sensor, International Journal of Research in Engineering, Science and Management.
    9. Mane, S. V., & Gaikwad, S. S. (2017). IoT-based Solar Powered Smart Irrigation System for Precision Agriculture. International Journal of Advanced Research in Computer Science, 8(9), 586-589.
    10. Bharathi, R. S. S. & Sukanya, S. (2018). Smart Irrigation System using IoT and Solar Power. International Journal of Advanced Research in Computer Science and Software Engineering, 8(5), 1057-1060.
    11. Kuma Ravel, N., & Balasubramanian, P. (2017). An IoT-Based Solar-Powered Automated Irrigation System for Sustainable Agriculture. International Journal of Engineering and Technology, 9(3), 2394-2399.
    12. Das, S. K., & Nanda, S. K. (2019). Design and Development of IoT-based Solar Powered Irrigation System for Precision Farming. In Proceedings of the International Conference on Intelligent Sustainable Systems (pp.791-796). Springer, Singapore.