DESIGN AND IMPLEMENTATION OF SMART IRRIGATION SYSTEM

by

GAZI MOHAMMED ARMAN RAFEE

BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

Department of Electrical and Electronic Engineering

INTERNATIONAL ISLAMIC UNIVERSITY CHITTAGONG

APRIL 2022

DESIGN AND IMPLEMENTATION OF SMART IRRIGATION SYSTEM

by

GAZI MOHAMMED ARMAN RAFEE

A project
submitted as partial fulfillment of the requirement for the degree of

BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

Department of Electrical and Electronic Engineering
INTERNATIONAL ISLAMIC UNIVERSITY CHITTAGONG

APRIL 2022

CERTIFICATE OF APPROVAL

The project entitled as "**Design and Implementation of Smart Irrigation System**" submitted by **Gazi Mohammed Arman Rafee**, bearing Matric ID. **ET171029** of session **Spring 2021**, to the Department of Electrical and Electronic Engineering, International Islamic University Chittagong, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Bachelor of Science in Engineering and approved for the examination held on **April, 2022**.

Supervisor
Mohammed Abdul Kader
Assistant Professor,
Department of Electrical and Electronic Engineering
International Islamic University Chittagong.

DECLARATION

This work has been completed by myself, and no part of it has been submitted to another institution for the purpose of receiving a degree or diploma.

____

Gazi Mohammed Arman Rafee

ACKNOWLEDGMENT

All praises and thanks to Allah, the Lord of the world, the most Beneficent, the most Merciful for helping us to accomplish this work.

I owe a debt of gratitude to a number of people who helped me to complete this work. Additionally, I would like to express my gratitude to our project supervisor Mohammed Abdul Kader (Assistant Professor, Department of Electrical and Electronic Engineering, IIUC), who helped me get started by providing the fundamental concepts and ideas I needed for the project as well as for his ongoing support and guidance. Additionally, I want to thank our older brothers and classmates for assisting me to solve a number of issues. In closing, I want to express my gratitude to my family and friends for putting up with my continuous whining about the project and for helping me get through the challenging moments.

Authors

ABSTRACT

Due to climate change and unpredictable rainfall, ground water levels are currently declining as well. For agricultural plants and crops, water is a need. Therefore, I have created a novel proposal for a smart irrigation system for the benefit of farmers as well as for farms. The water vapor level and temperature around the plant are both sensed by the humidity and temperature sensor. When a plant's soil moisture is low enough, water will be supplied from a reservoir using a level sensor, which monitors the reservoir's water level and then communicates the information to the controlling unit (Arduino Nano). Data is collected from sensors by the controlling unit (Arduino Nano), processed, and sent through IoT to the destination. This article provides an overview of system installation, system design, and system operation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

IoT	Internet of Things
Wi-Fi	Wireless Fidelity
PIC	Peripheral Interface Controller
GSM	Global System for Mobile communication
ARM	Advanced RISC Machine
LED	Light Emitting Diode
LCD	Liquid Crystal Display
PIR	Passive Infrared Sensor
PCB	Printed Circuit Board

CHAPTER 1
INTRODUCTION

1.1 Introduction

When compared to other industries in Bangladesh, agriculture is the most significant one. In our country, there is a larger demand to increase agricultural productivity due to expanding population. The agricultural practice of putting regulated water to land to help with crop production is known as irrigation. Crop output will be increased by building a suitable distribution system and providing an appropriate water supply. Unplanned water consumption unintentionally leads to water waste. Therefore, there is a pressing need to create solutions that reduce water waste without putting farmers under strain. The most significant cultural practice and labor-intensive activity in the everyday agriculture sector is irrigation. The world's most abundant source of energy is solar energy. Solar energy is not only an environmentally benign kind of energy but also a way to solve the current energy issue. In the current energy crisis, solar high-powered irrigation systems may be a viable alternative for farmers. The system pumps water from a bore well to a tank using alternative energy to power the water pumps. There are sensors and techniques to automatically predict when plants may require water. The primary goal of this study is to create an autonomous plant irrigation system that is PIC controlled. The choice about water management, which establishes the process's governing period, is supported by this system as well. Average vegetable yields may be increased by the use of effective irrigation techniques may be maintained or increased.

1.2 MOTIVATION

In general, farmers have switched to crops that use less water to maintain their businesses. There will be a crisis on the farm market if there is not an immediate answer to enhance irrigation systems. So, in order to preserve the water sustainability of the local area, to maintain crop fields in an environmentally friendly manner by preventing soil and earth from becoming flooded or dried out, and, most importantly, to reduce the financial cost of water usage for the farmers and the entire market, this smart water irrigation system was developed. Although a variety of environmental elements are important for agricultural output, the project's exclusive focus is on how much water agriculture uses. The project is a scaled-down representation of what happens on agricultural estates on a daily basis in terms of both size and complexity.

1.3 OBJECTIVES

These are the system's primary goals:

- To make the field better by keeping the root zone wet and ensuring that the water is distributed evenly.
- To remove the labor cost in this irrigation method.
- Water, electricity consumption are reduced to significant amount.

1.4 CHAPTER OUTLINE

The chapters' contents are listed below:

1. The first chapter serves as an introduction and lists the goals and motivation.
2. The second chapter reviews the literature and discusses earlier studies that are relevant to this topic.
3. The third chapter describes the hardware. Every major element that was used in this project is thoroughly detailed in this chapter.
4. The project's system architecture is covered in detail in chapter four, including block diagram, flow chart, and circuit descriptions.
5. Chapter five is system about implementation and results. This chapter also describes about upgradability of system.
6. Chapter six is conclusion where limitations and future works of this system are described.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The following steps are taken during the first investigation: understanding the current methodologies, comprehending the demands, and creating a system abstract. In this study, sensors for temperature, humidity, and soil moisture were put in the plant's root zone and sent data to an Android application. To control the amount of water, a microcontroller was programmed with the threshold value of a soil moisture sensor. The android app displays information on temperature, humidity, and soil moisture. The goal of this work, "Automatic Irrigation System on Sensing Soil Moisture Content," is to develop an automated irrigation mechanism that activates and deactivates the pumping motor in response to the amount of moisture in the soil. Only the soil moisture value is taken into account in this article, however the suggested project added temperature and humidity data as an expansion to the current project.

"Controlled Irrigation System for Plantation based on Microcontroller" Older microcontrollers with less memory are utilized in this work to manage the system, but the suggested system uses an Arduino nano board, which is user-friendly and makes it easier to dump programs. This paper presents "A Wireless Application of Drip Irrigation Automation Supported by Soil Moisture Sensors" (A Wireless Application of Drip Irrigation Automation Supported by Soil Moisture Sensors). Although the suggested system extends to temperature and humidity displays. By consulting the aforementioned papers, it is discovered that no such systems exist with all integrated features; however, the proposed system includes all of these features, including the ability to display temperature, humidity, and soil moisture values as well as the ability to automatically turn on and off a motor based on soil moisture values.

2.2 REVIEW OF PREVIOUS WORK

Previous works of this project are described as follows:

2.2.1 System based on ARM and GSM technology

The automated system based on ARM and GSM technologies was suggested by the author. In this system, a wireless sensor network is used to irrigate utilizing an automatic irrigation approach. The efficient supply of power is provided by photovoltaic cells. The motor will automatically shut off and send a message to the farmer through GSM if the water level exceeds a predetermined high level. [13]

2.2.2 Irrigation system using moisture, humidity & temperature sensor

Rupali S. Sawant: A microcontroller-based automated watering system was suggested by the author. The microcontroller series 8051 is used in this system. The system includes a solar panel, temperature, humidity, and soil moisture sensors. The microcontroller will be told whether or not to pump the water when the soil moisture level changes the threshold value. A humidity sensor is used to determine the ambient temperature. [11]

2.2.3 Smart Irrigation system using moisture & temperature

This prototype measures the moisture and temperature of the soil. A preset range of soil moisture and temperature is established, which can be changed depending on soil type or crop variety. When the moisture or temperature of the soil deviates from the prescribed range, the watering system is activated or deactivated. In the event of dry soil and excessive soil temperature, the irrigation system will be activated, pumping water to the plants. It has an ATmega328 microprocessor, which serves as the system's central processing unit. The input pins of the controller are linked to the moisture and temperature sensors, respectively. The output pins are connected to the water pump and the servo motor. The controller activates the pump if the sensors stray outside of the predetermined range. To guarantee that water is distributed equally to the soil, the servo motor is utilized to adjust the angular position of the pipe. The condition of the pump is shown via an LED indicator.[12]

2.2.4 Arduino Based Smart Irrigation System

Numerous sensors are linked to the input ports of the Arduino in this system, including pressure, pH, soil moisture, DHT11, and PIR sensors. The sensors' detected values are shown on the LCD. The relay circuit, which is coupled to the driver circuit, which aids in switching the voltage, will automatically turn the pump ON/OFF if the measured value exceeds the threshold values established in the program. Through the GSM module and website updates, the farmer will be informed of the present state of the field.[14]

2.3 Comparisons with my project:

I have developed irrigation system. Through this we can get to our field irrigated through pump having temperature, moisture, humidity and also water level sensor. With the help of this people can visually see the conditions of field. Moreover this can also be monitored and supply of water can be controlled automatically. For that it will be easier for field workers.

CHAPTER 3

COMPONENTS

3.1 Introduction

Along with discussing pin descriptions, pin diagrams, and component features, this chapter will also cover the project's primary components.

3.2 List of Components

- Arduino Nano
- Microcontroller PIC16F877A
- LCD
- Relays
- Relay Switching Circuit
- Wi-Fi Module
- Temperature & Humidity Sensor
- Soil Moisture Sensor
- Water Pump
- Battery
- Panel Indicator

3.3 Arduino Nano

Based on the ATmega328 (Arduino Nano 3.x) or ATmega168, the Arduino Nano is a compact, full-featured board that is easy to use with a breadboard (Arduino Nano 2.x). It comes in a different packaging but has about the same capabilities as the Arduino Duemilanove. It just lacks a DC power jack and functions with a Mini-B USB cable rather than a conventional one. The Mini-B USB port, a 6-20V unregulated external power source (pin 30), or a 5V regulated external power supply are all options for powering the Arduino Nano (pin 27). The greatest voltage source is automatically chosen as the power supply. The Arduino IDE program may also be used to program the Arduino Nano. The Nano has a pin 13 led as well. Obviously, ArdunioNano use the same software. However, the user may test out a variety of programs that are included in the Arduino software's built-in examples. Make sure we have chosen the appropriate board and port under the tools menu before uploading the software to Nano.

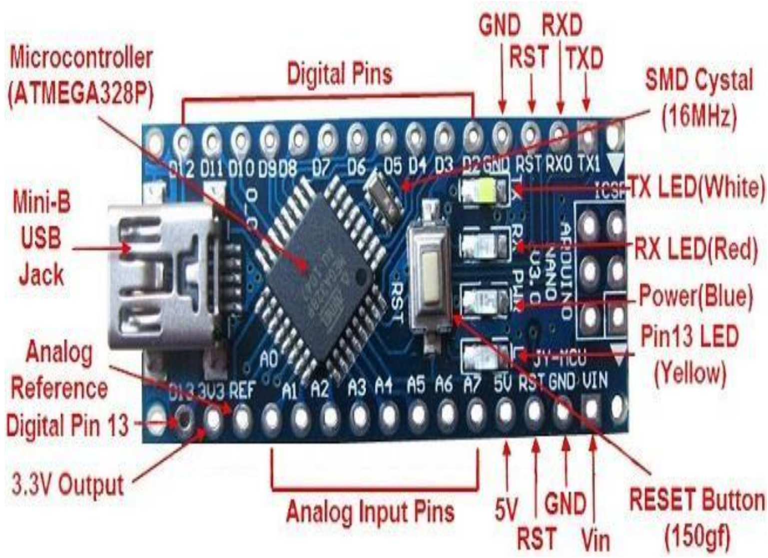

Fig.3.3: ArduinoNano

Pin descriptions of Arduino Nano:

Digital Pins

Pins - 1, 2, 5 to 16

The 14 digital I/O pins of the Arduino Nano, as was already established, may be utilized for either digital input or output. The pins operate at a maximum voltage of 5V, with digital high being 5V and digital low being 0V. Each pin has a pull-up resistance of around 20–50 kohms and a current capacity of 20mA. Using the pinMode (), digitalWrite (), and digitalRead () methods, each one of the 14 digital pins on the Nano pinout may be utilized as an input or an output. The digital pins have 11 additional functions in addition to those for digital input and output.

Serial Communication Pins

Pins - 1, 2

1 - RX and 2 – TX

For TTL serial data connection, these two pins, RX- receive and TX- transmit, are utilized. The USB-to-TTL Serial chip's RX and TX pins are linked to their corresponding pins.

PWM Pins

Pins - 6, 8, 9, 12, 13, and 14

Each of these digital pins provides a pulse width modulation signal with an 8-bit resolution. The PWM signal may be produced by using the analogWrite () function.

SPI Pins

Pins - 13, 14, 15, and 16

We may make advantage of these Serial Peripheral Interface pins when we wish the data to be transferred asynchronously. With SCK serving as the synchronizing clock, these pins provide synchronous communication. Although the hardware has this capability, the Arduino software does not by default have it. In order to use this functionality, we must add a library named SPI Library.

LED

Pin – 16

If we recall the blinking LED code from our first Arduino project, we will undoubtedly see this Pin16. The board's blinking LED is attached to pin 16 of the device.

Arduino Nano Analog Pins

Pins – 18 to 26

As was already established, the Arduino UNO has six analog input pins, while the Arduino Nano has eight, numbered A0 through A7 (19 to 26). This implies that you can integrate *8 channel analog sensor inputs. These analog pins each include an internal ADC with a resolution of 1024 bits (so it will give 1024 values). The pins are usually measured from ground to 5 volts. If you want the reference voltage to range from 0V to 3.3V, you may use the 12 analogReference () function to supply 3.3V to the AREF pin (18th Pin).

3.3.2 The following are the major characteristics of an Arduino NANO:

- The 8-bit AVR family of microcontrollers includes the ATmega328P.
- The working voltage is 5V.
- The input voltage (Vin) is between 7 and 12 volts.
- There are 22 input/output pins.
- The analog I/P pins range from A0 to A5.
- There are 14 digital pins.
- The amount of electricity used is 19 mA.
- The I/O pins' DC current is 40 mA.
- The flash memory size is 32 KB.
- CLK frequency is 16 MHZ.
- Weight-7g.

3.4 PIC16F877A:

The PIC® architecture from Microchip is packed into this powerful (200 millisecond instruction execution) yet simple-to-program (just 35 single word instructions) CMOS

Upwardly compatible with the PIC16C5X, PIC12CXXX and PIC16C7X devices, FLASH-based 8-bit microcontroller. The PIC16F877A has 256 bytes of EEPROM data memory, self-programming, an ICD, two comparators, eight channels of 10-bit Analog-to-Digital (A/D) converter, two capture/compare/PWM functions, and a Universal Asynchronous Receiver Transmitter. It also has a synchronous serial port that can be configured as either a 3-wire Serial Peripheral Interface (SPITM) or a 2-wire Inter-Integrated Circuit (I (USART).

3.4.1 How to Select PIC Microcontroller

Microchip offers a wide range of PIC family microcontrollers. Each MCU has unique benefits and drawbacks. Before choosing an MCU for a project, one must take into account a number of factors. The following are only recommendations that may aid with MCU selection.

o A newcomer studying PIC should choose an MCU with a large number of applications and adequate online community help. These MCUs include the PIC16F877A and the PIC18F4520.

o Take the system's operating voltage into account. Choose a 5V MCU if they are 5V. A 3.3V MCU can be chosen in these cases as some sensors or devices operate and communicate on this voltage.

o If cost and size are constraints, we can use tiny 8-pin MCUs like the PIC12F508. These are also considerably less expensive.

o Determine the MCU modules that we would require based on the sensors and actuators that were employed in the project. For instance, if we need to read a lot of analog signals, we need to make sure the PIC has enough ADC channels and a good resolution. The table above contains information about each module.

o Check that the PIC can handle communication protocols like UART, SPI, I2C, CAN, etc. if your project calls for them. More than one module of the same protocol can be supported by some MCUs.

3.4.2 Programming PIC Microcontroller

A PIC microcontroller may be programmed using a variety of commercial software. Some individuals still program PIC MCUs in assembly language. The information provided here pertains to the most popular and cutting-edge software and compiler created by Microchip. An IDE (Integrated Development Environment), where the programming is done, is required to program the PIC microcontroller. A compiler, where our software is transformed into HEX files, which are readable by MCUs. Our hex file is dumped into our PIC MCUs using an IPE (Integrated Programming Environment). All three of these pieces of software are free thanks to Microchip. Direct downloads of them are available from their official page. For our convenience, I have also included the link. Install these on your PC after downloading. If there are any issues, we may submit them in the comment section below. We will require a gadget known as PIC kit 3 in order to dump or upload our code into PIC. The PIC kit 3 programmer/debugger is a straightforward, inexpensive in-circuit debugger that is managed by a PC running Windows and MPLAB IDE (v8.20 or later) software. The toolkit of a development engineer must include a PIC kit 3 programmer/debugger. We will also require additional gear, such as a breadboard or perf board, a soldering station, PIC ICs, crystal oscillators, capacitors, etc.

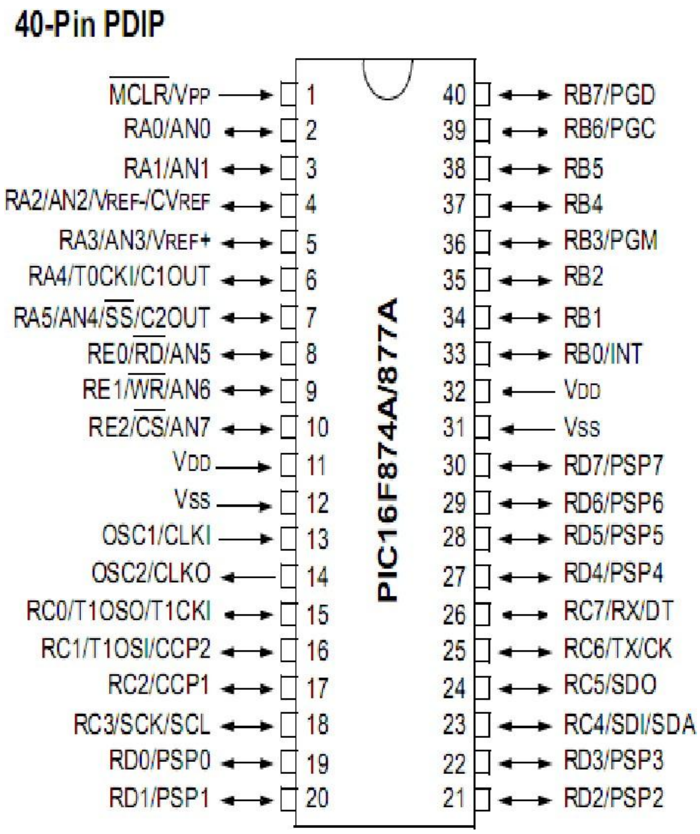

Fig.3.4: PIC16F877A Microcontroller

PIC16F877A Features:

Table 3.4: PIC16F877A Detailed Features

PIC16F877A –Simplified Features	
CPU	8-bit PIC
Number of Pins	40
Operating Voltage (V)	2 to 5.5 V
Number of I/O pins	33
ADC Module	8ch, 10-bit
Timer Module	8-bit(2), 16-bit(1)
Comparators	2
DAC Module	Nil
Communication Peripherals	UART(1), SPI(1), I2C(1), MSSP(SPI/I2C)
External Oscillator	Up to 20Mhz
Internal Oscillator	Nil
Program Memory Type	Flash
Program Memory (KB)	14KB

3.5 LCD Display

A flat-panel display or other electronically controlled optical device that makes advantage of the polarizing and light-modulating capabilities of liquid crystals is known as a liquid-crystal display (LCD). Liquid crystals don't emit light directly; instead, they create pictures in either color or monochrome utilizing a backlight or reflector. There are LCDs that can show random graphics (like those on a general-purpose computer display) or fixed images with little information that can be seen or concealed, such text, numbers, and seven-segment displays like those used in digital clocks. They both make use of the same fundamental technology, although some displays have bigger components, whilst others employ a grid of tiny pixels to create random pictures. Depending on the polarizer configuration, LCDs may be switched between being typically on (positive) and off (negative). In contrast, a character negative LCD will have a black backdrop with letters that are the same color as the backlight and a character positive LCD will have black writing on a background that is the color of the backlight. Blue LCDs have optical filters applied to the white to give them their distinctive look.

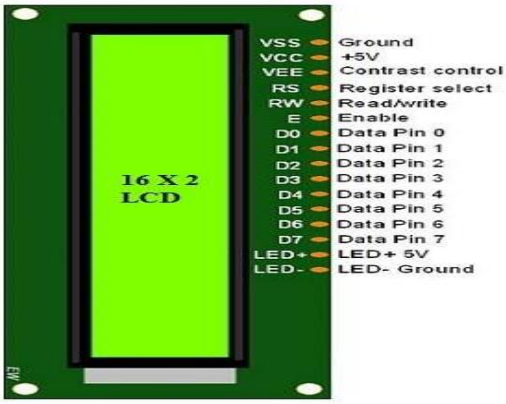

Fig 3.5: LCD Display

Table 3.5: Pin Description of LCD Display

Pin No	**Function**	**Name**
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; through a variable resistor	VEE
4	When the command register is low, and when the data register is high.	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	When a high to low pulse is provided, it sends data to the data pins.	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7

3.6 Relay:

An electrically operated switch. Although many relays use an electromagnet to manually activate a switch, solid-state relays are one form of relay that operate on this concept. When more than one circuit has to be controlled by a single signal or when absolute electrical isolation between the control and controlled circuits is needed, relays are used. The signal from one circuit was repeated and retransmitted on another by the original relays, acting as amplifiers in long-distance telegraph circuits. Relays were commonly used in early computers and telephone exchanges to carry out logical operations. A specific type of relay called a contactor can control the high power required to directly operate an electric motor or other loads. Solid-state relays control power circuits by switching is carried out by a semiconductor chip and there are no moving parts. Relays with calibrated operating parameters and occasionally several working coils are used to protect electrical circuits from overload or flaws; in modern electric power systems, digital instruments still go by the term of "protective relays."

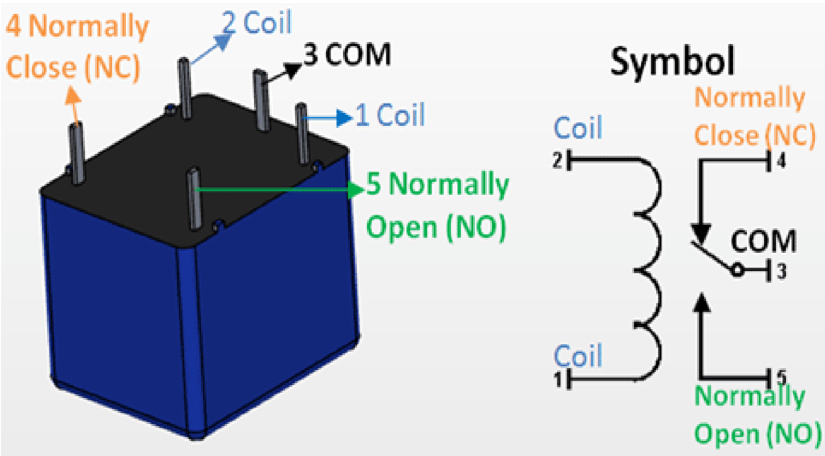

Fig.3.6: Relay

3.7 NPN Relay Switch Circuit:

Depending on the input voltage level, an NPN transistor switch, TR1, as illustrated, drives the coil in a standard relay switch circuit. The transistor is cut off and functions as an open switch when its base voltage is 0 (or negative). Since relay coils are current devices and no current flows through them when no current runs through the Base, they are de-energized in this situation when no Collector current flows.

If a sufficiently strong positive current is now being delivered into the Base to saturate the NPN transistor, the current flowing from Base to Emitter (B to E) governs the larger relay coil current running through the transistor from the Collector to Emitter. For the majority of bipolar switching transistors, the relay coil current flowing into the collector would be 50 to 800 times more than the base current required to drive the transistor into saturation. The current gain, or beta value (), of the all-purpose BC109 is typically 290 at 2 mA. (Datasheet).

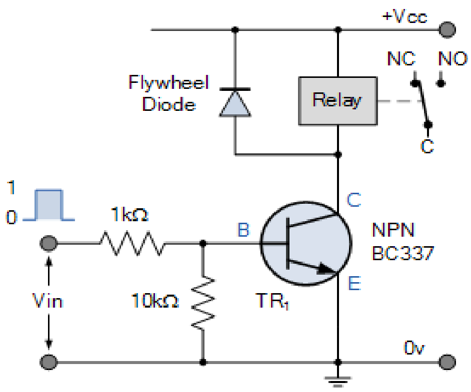

Fig.3.7: NPN Switching Circuit

Keep in mind that the relay coil doubles as an inductor and an electromagnet. The coil's DC resistance, as determined by Ohm's Law (I = V/R), will cause a maximum current to flow when power is provided to the coil as a result of the transistor's switching action. Some of this electrical energy is stored in the relay coil's magnetic field. When the transistor is turned "OFF," the magnetic field decreases and the amount of current flowing through the relay coil decreases. However, while it tries to maintain the current flowing through the relay coil, a reverse voltage arises across the coil because the magnetic field's stored energy must go somewhere.

When this is done, a significant voltage spike is experienced by the relay coil. If this voltage spike is allowed to build up, it may damage the switching NPN transistor. As a result, to protect the semiconductor transistor from harm, a "flywheel diode," also known as a freewheeling diode, is connected across the relay coil. By limiting the reverse voltage across the coil to around 0.7V, the flywheel diode protects the switching transistor and releases the stored energy. Flywheel diodes can only be used with polarized DC voltages. An RC circuit is used in this instance because an AC coil requires a different kind of protection.

3.8 Wi-Fi Module (ESP8266):

In order to host the program or to offload all Wi-Fi networking tasks from another application processor, ESP8266 delivers a comprehensive and self-contained Wi-Fi networking solution. It may boot up directly from an external flash when the ESP8266 is the only application processor in the system and is hosting the software. It has an inbuilt cache to enhance system performance in specific applications and lower memory requirements. The ESP8266 Wi-Fi Module, a self-contained SOC with an integrated TCP/IP protocol stack, enables any microcontroller to connect to a Wi-Fi network. The ESP8266 is capable of hosting an application or delegating all Wi-Fi networking duties to another application processor. All we need to do to have about the same degree of Wi-Fi capability as a Wi-Fi Shield is attach an ESP8266 module to our Arduino device because each of them is already pre-programmed with an AT command set software.

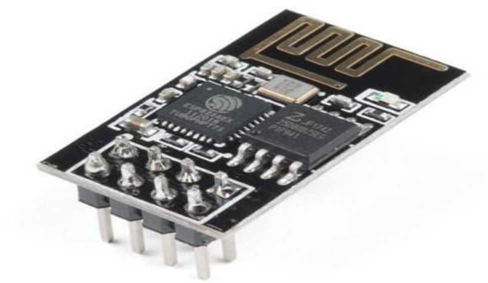

Fig.3.8: Wi-Fi Module (ESP8266)

The ESP8266 module has a huge and growing community and is a relatively inexpensive board. Thanks to its strong on-board processing and storage capabilities, this module may be connected with sensors and other application-specific devices through its GPIOs with a minimum of upfront programming and runtime loading. It only needs a tiny amount of external circuitry because to the high level of on-chip integration, and even the front-end module is designed to take up little room on the PCB.

3.8.1 Wi-Fi Module (ESP8266) Pin Configuration:

Table 3.8: Wi-Fi Module (ESP 8266) Pin Configuration

Pin	Pin Name	Normally used for	Alternate purpose
1	Ground	connected to the circuit's ground.	
2	TX	Connected to RX pin of programmer to upload the program.	When not being utilized as a TX, can function as a general-purpose input/output pin.
3	GPIO-2	General purpose input output pin.	
4	CH_EN	Chip Enable –Active high.	
5	GPIO-0	General purpose input output Pin.	When held low at startup, move the module to serial programming.
6	Reset	Resets the module.	

3.9 DHT-11 Temperature and Humidity Sensor:

The temperature and stickiness sensor complex with a calibrated digital signal input is highlighted by the DHT11 Temperature & Humidity Sensor. It assures excellent dependability and amazing long-term stability by utilizing the selected computerized sign securing strategy and temperature and dampness sensor technology. This sensor combines a resistive-sort moisture estimation segment with an NTC temperature estimation segment to deliver high quality, rapid response, obstruction resistance, and cost effectiveness. A high-performance 8-bit microcontroller is connected to it.

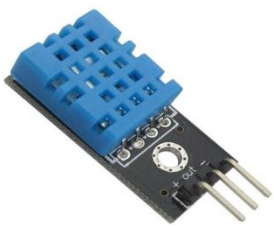

Fig.3.9: DHT11 Temperature and Humidity Sensor

Every DHT11 component will be completely balanced in the lab, making wetness adjustment very precise. The correction coefficients are stored as OTP memory projects and used by the sensor's internal sign-discrimination algorithm. The framework reconciliation is quick and easy because to the single-wire serial interface. It is the greatest option for a variety of applications, including the most popular ones, because to its small size, low power consumption, and up to 20 meter signal transmission range. The component is a single line, 3-pin pin bundle. Associating has several benefits, and special packages might be offered in response to demand.

3.9.1 Technical Specification

Table 3.9: Technical Specification

Measurement Range	20-90% RH 0-50℃
Humidity Accuracy	± 5% RH
Temperature Accuracy	± 2℃
Resolution	1
Package	3 Pin Single Row

3.10 Soil Moisture Sensor:

With the help of this moisture sensor, plants within a greenhouse may communicate with one another for human assistance or determine how wet the soil is surrounding the sensor. They are quite simple to use; just bury it in the ground and then read it. It will be possible to notify the plant that it needs water by using this sensor.

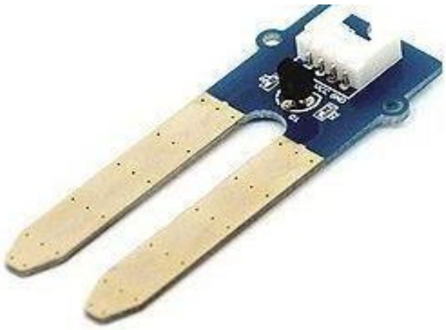

Fig 3.10: Soil Moisture sensor

Table 3.10: Technical Specification

Soil Condition	Transducer Optimum Range
Dry	0V
Optimum Level	1.9-3.5V
Slurry Soil	>3.5V

CHAPTER 4

SYSTEM DESIGN

4.1 Introduction

The most crucial stage of a project is the design process. In this chapter interfacing different devices with microcontroller, designing the block diagram, circuit diagram and flow chart will be described.

4.2 Interfacing different devices with Arduino

This project includes the interfacing of DHT11 sensors, LCD display, interfacing of Moisture sensors, microcontroller with Arduino. All the interfacings are shown below.

4.2.1 Interfacing DHT11 Sensors

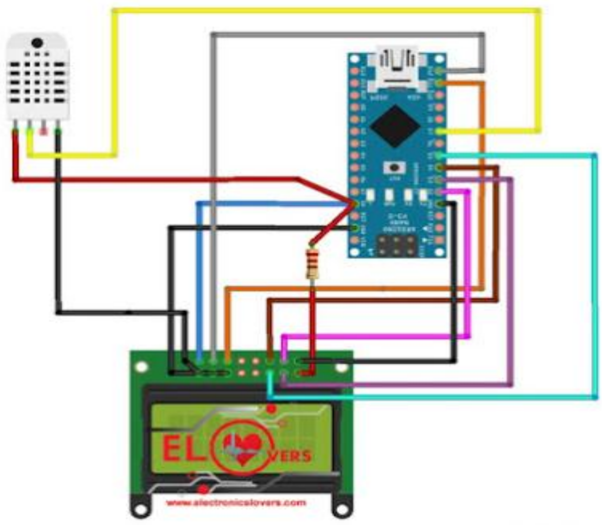

Fig.4.1: Interfacing DHT11 sensor

LCD Pins Configuration:

- Enable pin - digital pin 11
- RS pin - digital pin 12
- D4 pin - digital pin 5
- D5 pin - digital pin 4
- D6 pin - digital pin 3
- D7 pin - digital pin 2
- R/W pin - ground

- VDD - 5V
- VSS - Ground
- Vee or Vo - Ground

When integrating additional sensors or components with the Arduino microcontroller, library upgrades are quite helpful. In this situation, we will require the DHT 11 sensor library, and we must ensure that library is updated. after the copying or extraction of the DHT11 library into the Arduino library folder.

4.2.2 Interfacing Soil Moisture Sensors

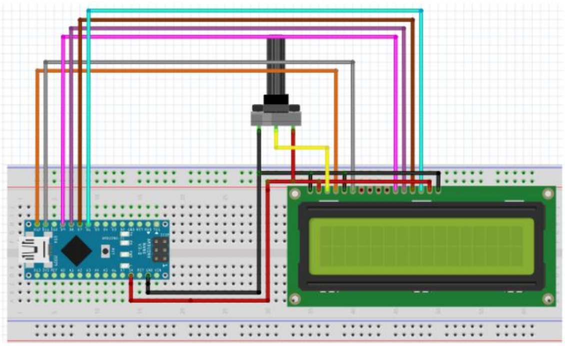

Fig.4.2: Interfacing of LCD Display

LCD Pins Configuration :

- RS pin - 12
- R/W - ground
- ENABLE - 11
- LCD D4 - 9
- LCD D5 - 8
- LCD D6 - 7
- LCD D7 - 6

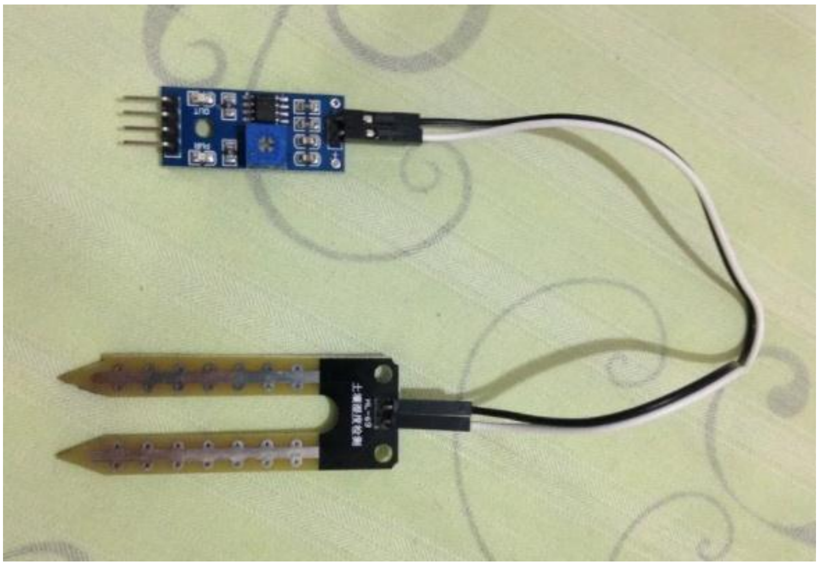

Fig.4.3: Interfacing of Soil Moisture Sensor

Pins Configuration of Soil Moisture Sensor:

- VCC----> +5 volt
- GND---->GND
- A0-------->A0

4.3 Block Diagram:

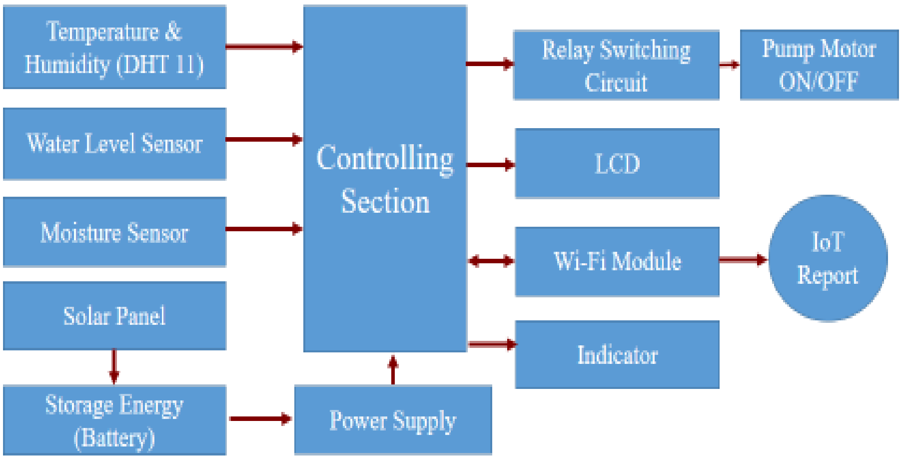

Fig.4.4: Interfacing of Soil Moisture Sensor

Here in the middle we've seen controlling section. Temperature & humidity, soil moisture and water level sensors will give value to controlling section and according to that pump motor will on or off through relay switching. The value given to the controlling section will show on LCD. Beside it will show to the IOT report through Wi-

Fi module. The system will be powered by battery which also can be charged by solar panel.

4.4 Circuit Diagram:

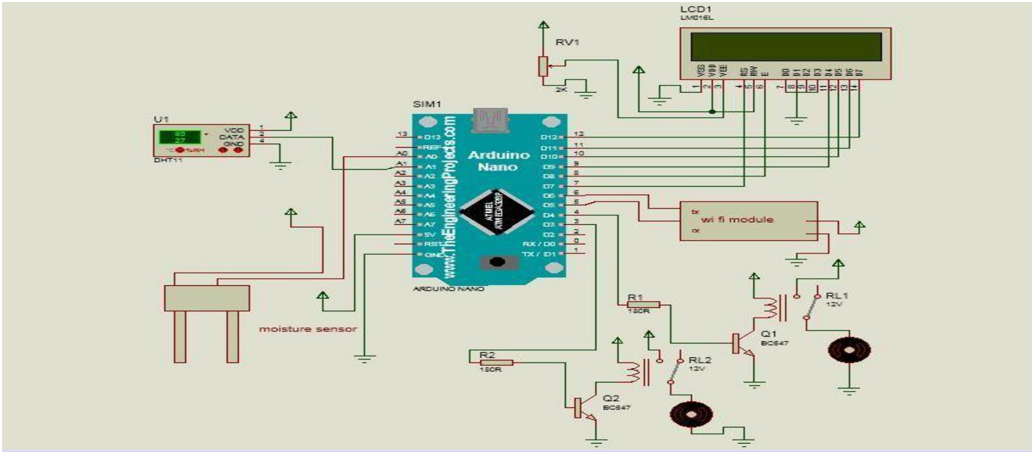

Fig.4.5: Circuit Diagram for IoT Monitoring & Sensor Assembling

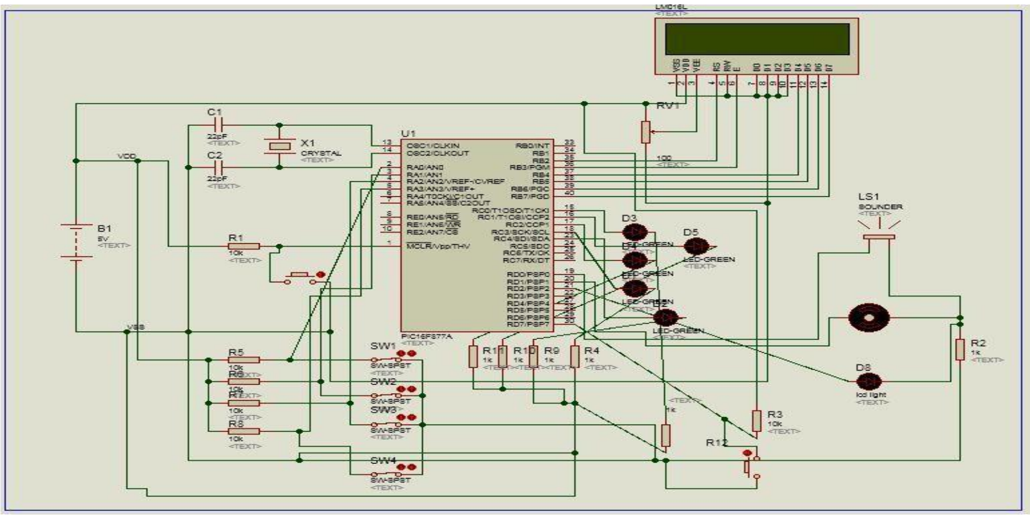

Fig.4.6: Circuit Diagram for Pump Motor Management System

Working Principle of the project

This system is design for automatic load control of a building property which saves our time and electrical energy. The principal goal of the system is to control our required load for required time by IoT system and also manual control. If we want to turn ON any load of our building in our absence we can do it by IoT system and controlled by Online. The functional block diagram of automatic load control is shown in figure. The system always runs with the 5Volt DC power supply. When the system is start to run, first the

MCU wait for Wi-Fi module signal. After getting signal microcontroller unit (MCU) performs operations in three modes. LCD display will always show all the readings. Transistor relay circuit controls the switching of the loads. We can operate the system also manual mode.

4.5 Flow Chart:

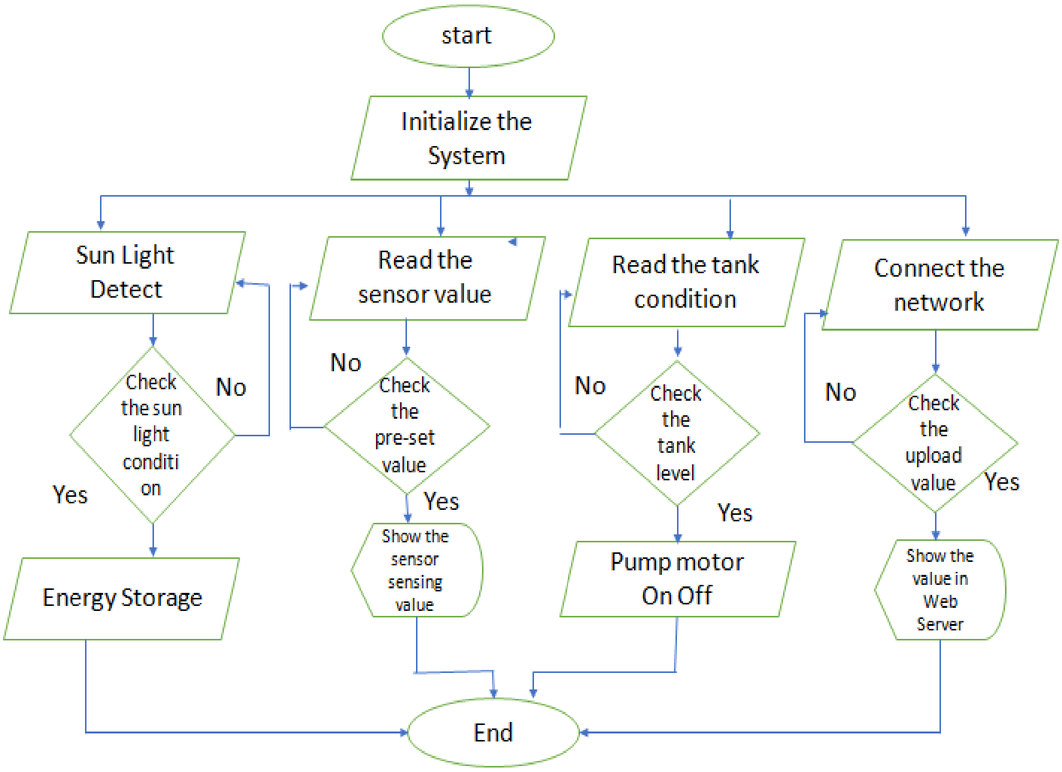

Fig.4.7: Flow Chart

Firstly the system will initialize then it will detect sunlight. If it is YES it will go downward. If it is NO then it will go backward. Similarly it will read the sensors value, tank condition and connection of network. If it is YES it will go downward. If it is NO then it will go backward.

CHAPTER 5

IMPLEMENTATIONS AND RESULT

5.1 Introduction

After completing all implementation as per design we turned power on then we confirmed that all the equipment works properly. This chapter contains the results obtained and discussion about the full project. We have also covered discussions about objective, advantages, limitation and application. PIC16F877A, Arduino Nano, Pump Motor, Humidity and temperature Sensor, Moisture Sensor, LCD, Wi-Fi module, Relay switching circuit are the main equipment's used in this project which give the exact value respectively. These sensor and input value in the correct result according to the plants condition. Those result can be seen on the below present on the project.

Our project operates by two method likes one method is controlling by website automatically and another method is controlling by manually.

Website Result:

Website is two parts:

1. Controlling.
2. Monitoring the whole System.

5.2 Controlling & Monitoring Section Result

After turning on Total System, we will see the below text on the display and the whole system will start working in 10 seconds.

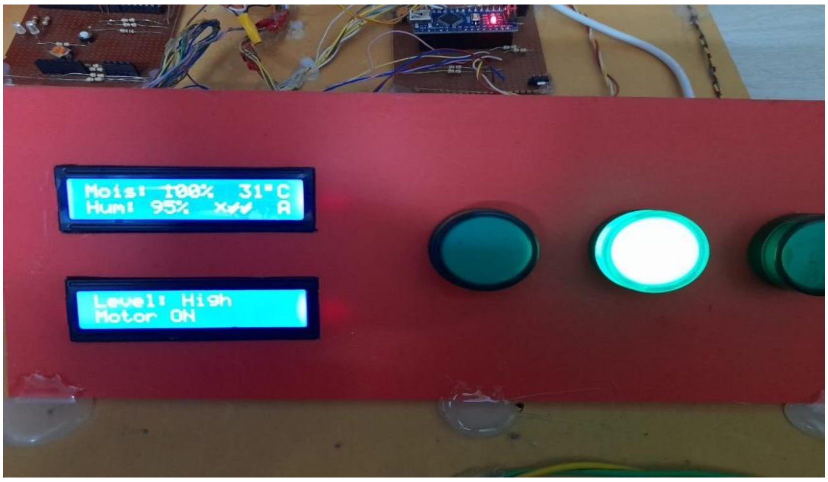

Fig.5.1: System initialize the device to internet

The measured analog value of the humidity & temperature and moisture sensor is being converted to digital value via Arduino, which we can see in the below display.

Fig.5.2: Sensor level value

We have used the password method in the security system. To go to the password method, go to the web site link, then log in the server user name and password, when press the correct username and password log in the web page, and show the result. It is shown in the figure below:

Fig.5.3: Web-Server log in

When we monitor the system with IoT, we can see every parameter, humidity, temperature moisture sensor value, pump motor condition as well as some loads on or off on the server which we can see as in the figure.

Fig.5.4: IoT Monitoring & Controlling

By using Internet of Things technology, this autonomous irrigation system turns on and off a motor pump by detecting the soil's moisture level (IOT). Because of global warming and unpredictable rainfall today, ground water levels have also fallen. For agricultural plants and crops, water is a need. When a plant's soil moisture is low enough, water will be supplied from a reservoir using a level sensor, which monitors the reservoir's water level and then communicates the information to the controlling unit (Arduino Nano). Now we will show the pump motor running condition at below:

Fig.5.5: Pump Motor Condition

Table 5.1: Experimental Value

SL No	Moisture Value	Humidity Value	Temperature Value	Water level	Pump Motor
01	100%	80%	30º C	Full	Off
02	90%	76%	29º C	High	Off
03	80%	71%	32º C	Medium	Off
04	35%	63%	30º C	Low	On

We have noted some data that we have received at different times from the sensors used in our project. It is shown in the table below 5.2:

Table 5.2: Sensor Data

SL No	Temperature Sensor	Humidity Sensor	Moisture Sensor
01	31º C	95 %	100%
02	30º C	80 %	45 %
03	33º C	100 %	60 %
04	29º C	70 %	85 %
05	27º C	60 %	90 %
06	23º C	75 %	100 %
07	29º C	90 %	35 %
08	30º C	85 %	85 %
09	32º C	100 %	90 %
10	34º C	95 %	100%

5.3 Cost Calculation:

The costing of this project is given below:

Table 5.3: Cost Analysis

SL NO	Device Name	Amount	SL NO	Device Name	Amount
01	Arduino Nano	400	11	DC Relay	60
02	LCD Display	140	12	Switch	10
03	DHT11 Sensor	120	13	Transistor	10
04	Moisture Sensor	130	14	Buzzer	15
05	Variable Resistor	05	15	Crystal	10
06	DC pump motor	440	16	Resistor	05
07	Solar Panel	500	17	Vero Board	30
08	Battery	450	18	Power Diode	05
09	Wi-Fi module	220	19	Panel Indicator	120
10	Regulator IC	10	20	Microcontroller	350

5.4 Hardware Result:

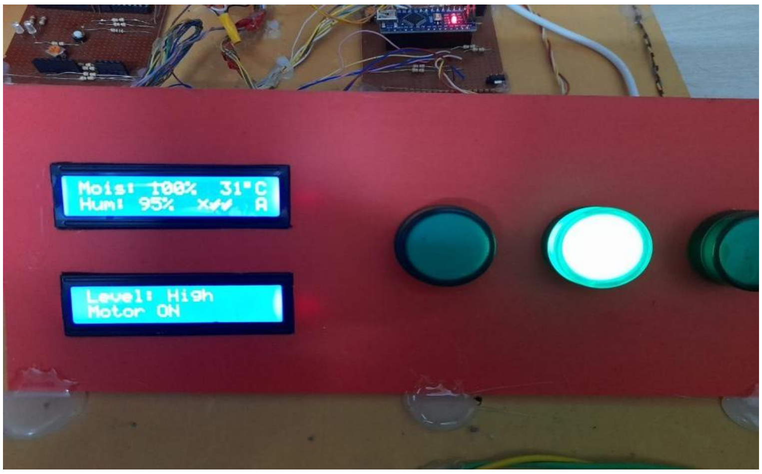

Fig.5.6: Data upload to the server

In the figure above we have seen that data uploaded to the server. Here in the above LCD display will show the value of moisture, humidity and temperature data. The next LCD will show the value of water level. And on the basis of water level it will show the condition of motor whether it is on or off. The indicator will ON or OFF on the basis of power of system.

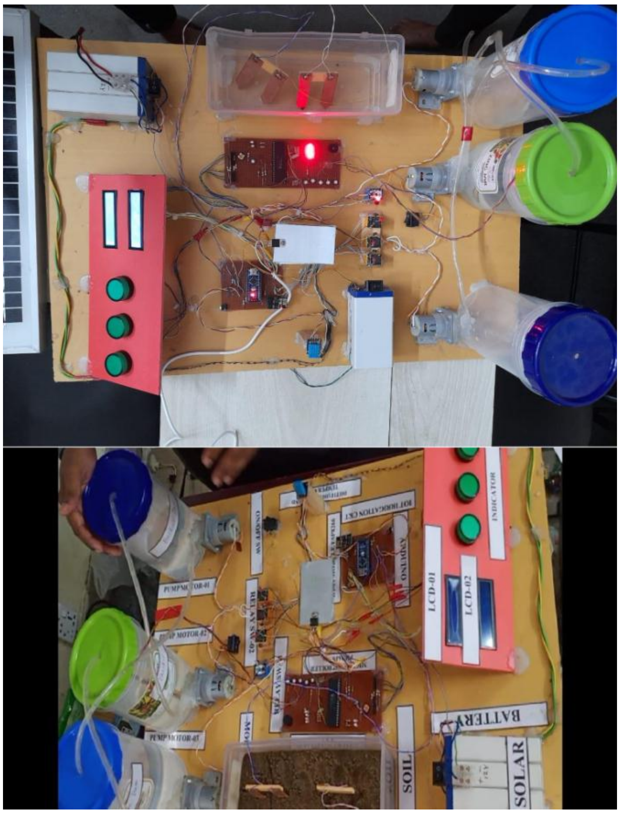

Fig.5.7: Total System

In the figure above we have seen the whole system. In the system there are three pots which are determined as dam, reservoir and supply tank. The whole system will be powered by battery and the battery will charged by solar panel. There is soil where we put soil moisture sensor. And there we’ve seen a panel where indicator and LCD are connected. In the LCD the value will be shown. There is also a Wi-Fi module which is attached for getting IoT report. In the system there are LED lights also seen which will ON or OFF on the basis of water level.

CHAPTER 6

CONCLUSION

6.1 Conclusion

The installed intelligent irrigation system makes efficient use of available water supplies for agricultural produce. The method is made easier to use by the suggested system, which may be used to turn on and off the water sprinkler depending on the soil moisture levels. Through this experiment, it can be inferred that irrigation may significantly advance thanks to IOT and automation. Thus, the issues with the current irrigation procedure are resolved by this approach..

6.2 Limitation

This system has some loopholes

- Salt efflorescence can result from careless irrigation and overwatering.
- The spilling irrigation water provides insects with a habitat, which is hazardous to the environment and can lead to a number of diseases.

6.3 Future Works

- Bio pesticides and fertilizers can be added automatically into the water.
- PIR sensors can be used to detect the motions of animal in the agriculture field.
- If the system doesn’t work properly due to any problem, then the user should be notified via SMS as well, it can be added.

REFERENCES

[1] Prabhakaran, S. K. Monisuthan, and V.S. Pavithra. "Farmers Can Use An Automated Irrigation System Through A Mobile GSM Android Application." ISSN974:2115 *Journal of Chemical and Pharmaceutical Sciences* .

[2] "Automated irrigation system employing a wireless sensor network and GPRS module," by Joaqun Gutiérrez et al. 2014: 166–176. *IEEE Transactions on Instrumentation and Measurement*, 63.1.

[3] *Going with the flow: sustainable water management as ontological cleaving*, Stephanie Lavau. 31.3 (2013): 416-433 for Environment and Planning D: Society and Space

[4] "*Automatic plant watering device*," W. Crane. 3,758,987 U.S. Patent. 18 Sep.

[5] "Arduino Based Smart Drip Irrigation System Using Internet of Things," by G. Parameswaran and K. Sivaprasath. *J. Eng. Sci.* 5518 (2016).

[6] "Automatic Irrigation System on Sensing Soil Moisture Content," by R. Vagulabranan, M. Karthikeyan, and V. Sasikala. 0072-0056 3.03 *International Research Journal of Engineering and Technology* (2016).

[7] Nikhil, Smita Singhal, and Agrawal. "Smart Arduino and Raspberry Pi drip irrigation system." *International Conference on Computing, Communication, and Automation (ICCCA)*, 2015 IEEE.

[8] Semih Ozden, Mahir, and Dursun. "A wireless application of soil moisture sensors supporting drip irrigation automation." 1573–1582, *Scientific Research and Essays* 6.7 (2011).

[9] "Automated irrigation system employing a wireless sensor network and GPRS module," by Joaqun Gutiérrez et al. 2014: 166–176. *IEEE Transactions on Instrumentation and Measurement,* 63.1.

[10] Agrawal, Nikhil, and Smita Singhal. "Smart drip irrigation system using raspberry pi and arduino." *Computing, Communication & Automation (ICCCA),* 2015 international Conference on. IEEE, 2015.

[11] "Solar Panel Based Automatic Plant Irrigation System," Sawant, Rupali S., et al. 2.3 (2015): 698–701 *International Journal of Innovative Science, Engineering, and Technology*

[12] e-ISSN 2278-2834, p-ISSN 2278-8735, *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE).* May-June 2015, pp. 32–36, Volume 10, Issue 3, Ver. II.

[13] Riyaj Kazi, Lakshata, and Bakade. *"Review on a Wire-less Sensor Network Automated Irrigation System."*

[14] "Arduino based smart irrigation system utilizing IoT." Nandhini, R., et al. The 2017 edition of the *third national conference on intelligent information and computing technologies. 2017.*

APPENDIX

```
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
#dene fire A3
#define buzz 13
#define light 10
#define fan 11
#define pump 12
#define wifi Serial

LiquidCrystallcd (7, 6, 5, 4, 3, 2);
SoftwareSerialgsm(9, 8);

const char num[] = "01768225410";
String wifiSSID = "iot";
String wifiPASS = "12345678";
String HOST = "iotweathermonitor.000webhostapp.com";

nt gas, temp;
boolserverMode = 0;
boolsmsFlag = false;
void setup() {
wifi.begin(115200);
gsm.begin(9600);
lcd.begin(16, 2);

pinMode(fire, INPUT);
pinMode(buzz, OUTPUT);
pinMode(light, OUTPUT);
pinMode(fan, OUTPUT);
pinMode(pump, OUTPUT);

wifi.println("AT+CWMODE=1; checkResponse();
wifi.println((String)"AT+CWJAP=\"" + wifiSSID + "\",\"" + wifiPASS + "\"");
checkResponse();
}
void loop() {
gas = analogRead(A5);
temp = analogRead(A7);
temp = temp * 0.4887;
checkServer();
delay(500);
if(serverMode == 0){
if(temp > 32) digitalWrite(light, 1);
else if (temp < 28) digitalWrite(light, 0);
if(gas > 100) digitalWrite(fan, 1);
else if (gas < 100) digitalWrite(fan, 0);
if(!digitalRead(fire)) digitalWrite(pump, 1);
elsedigitalWrite(pump, 0);
if(!smsFlag&& (temp > 32 || gas > 100 || !digitalRead(fire))) {
sendSMS(num);
```

```
smsFlag = true;
 }
elsesmsFlag = false;
 }
lcdprint(0, 0, (String)temp + (char)223 + "C G:" + gas + " F:" + !digitalRead(fire) + " ");
lcdprint(0, 1, (String)"L=" + digitalRead(light) + " F=" + digitalRead(fan) + " P=" + digitalRead(pump) +
" " + serverMode);
send2Server();
delay(500);
}
voidlcdprint(byte x, byte y, String txt){
lcd.setCursor(x, y);
lcd.print(txt);
}
void send2Server(){
 String post = (String)"GET /update_machine.php?&temp=" + temp + "&gas=" + gas + "&fire=" +
!digitalRead(fire) + " HTTP/1.1";
int index = post.length() + 66;
wifi.println("AT+CIPMUX=0");
checkResponse();
wifi.println("AT+CIPSTART=\"TCP\",\"" + HOST + "\",80");
checkResponse();
wifi.println((String)"AT+CIPSEND=" + index);
checkResponse();
wifi.println(post); // 2
wifi.println("HOST: " + HOST); // 8 + 35
wifi.println("Connection: Close\r\n"); // 21
checkResponse();
}
voidcheckServer(){
 String post = (String)"GET /stat_machine.php HTTP/1.1";
int index = post.length() + 66;
wifi.println("AT+CIPMUX=0");
checkResponse();
wifi.println("AT+CIPSTART=\"TCP\",\"" + HOST + "\",80");
checkResponse();
wifi.println((String)"AT+CIPSEND=" + index);
checkResponse();
wifi.println(post); // 2
wifi.println("HOST: " + HOST); // 8 + 35
wifi.println("Connection: Close\r\n"); // 21
 String check = checkResponse();
if(check.indexOf("server=0") != -1) serverMode = 0;
else if(check.indexOf("server=1") != -1) serverMode = 1
if(serverMode == 1){
if(check.indexOf("light=0") != -1)
digitalWrite(light, 0);
else if(check.indexOf("light=1") != -1)
 digitalWrite(light, 1);
if(check.indexOf("fan=0") != -1)
 digitalWrite(fan, 0);
else if(check.indexOf("fan=1") != -1)
 digitalWrite(fan, 1);
```

```
if(check.indexOf("pump=0") != -1)
 digitalWrite(pump, 0);
else if(check.indexOf("pump=1") != -1)
digitalWrite(pump, 1);
 }
}
String checkResponse(){
 String r;
while(!wifi.available());
while(wifi.available()){
 r = wifi.readString();
 //Serial.println(r);
 }
delay(50);
return r;
}
voidGSMinit(){
lcd.clear();
lcd.print("GSM Init...");
delay(15000);
voidsendSMS(char *number){
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Sending SMS..");
digitalWrite(buzz, 1);
gsm.print("AT+CMGF=1\r\n");
delay(200);
gsm.print("AT+CMGS=\"");
delay(200);
gsm.print(number);
gsm.print("\"\r\n");
delay(200);
delay(3000);
lcd.clear();
digitalWrite(buzz, 0);
}
```