Chapter 1

About the Project

1.1 Introduction

India is the land of villages. This being said the major occupation of majority of villages in India is agriculture. Near about 70% people are dependent upon agriculture. Agriculture has been the backbone of the Indian economy and it will continue to remain so for a long time. It has to support almost 17 per cent of world population from 2.3 per cent of world geographical area and 4.2 per cent of world's water resources. The economic reforms, initiated in the country during the early 1990s, have put the economy on a higher growth trajectory. Annual growth rate in GDP has accelerated from below 6 percent during the initial years of reforms to more than 8 percent in recent years. This happened mainly due to rapid growth in non-agriculture sector. The workforce engaged in agriculture between 1980-81 and 2006-07 witnessed a very small decline; from 60.5 percent to 52 percent.

With recent advancements in engineering and technology, there also have been changes in agricultural technology and practices. Though there are advancements people still follow old practices due to the lack of money and high cost of technically advanced agricultural equipment's. According to a study made by ISAE, it is found that hoes, pangas, axes and shovels are the main farm tools used by the farmers in India for agricultural operation. These tools are conventional, time immemorial and no improvement in agricultural practice is adopted. Hence, it is necessary to develop a system which results in drudgery reduction and id user friendly to agricultural community in India.

Farmers in India perform agriculture mostly with manual operation. The pain involved in doing each and every operation has to be reduced by the way of introducing simple technology which is not only user friendly to farmers but also is economical for farmers to adopt. This project deals with the concept of IOT assisted agriculture. Internet of Things (IoT) is the network of physical things embedded with electronic circuits, sensors, software and network connection which enables these things to exchange data from one another. IoT is the fusion of the digital and physical world. In a world of IoT, millions of things or devices will be interconnected and uniquely identified on the Internet. The Internet of Things allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration between the physical world and computer-based systems, and resulting in improved efficiency, accuracy and economic benefit. In near future, IoT is expected to provide many more services like advanced connectivity of physical objects over a wide network and also many applications. It is obvious to think that in using these services provided by this technology, it is possible to control and monitor systems from a distance using the GSM network. Mobile internet are integrated-applications as useful as home automation, industrial applications for handling and remote monitoring of complex systems but also in security systems, and protect property and people. Most physical variables relevant in a greenhouse can be measured by automatic sensors.

IOT is a vision of a world in which most objects are connected; transmitting updates about their performance so the people who use them to do things more intelligently. The basic concept behind the IOT is that virtually every physical thing in this world can also become a computer that is connected to the internet. In this project the sensor will sense the vital parameters of the environment. The sensors are connected to the microcontroller. This project involves connected ***monitoring, control*** of agricultural parameters using ***Internet of things.*** Various sensors are interfaced with the controller and the output is sent over the internet to the develop web using IOT framework.

In India about 73% of population is directly or indirectly depends upon the farming. Hence it is said that India is an agriculturally based country. Near about 70% people are dependent upon agriculture. Agriculture has been the backbone of the Indian economy and it will continue to remain so for a long time. It has to support almost 17 per cent of world population from 2.3 per cent of world geographical area and 4.2 per cent of world's water resources. The economic reforms, initiated in the country during the early 1990s, have put the economy on a higher growth trajectory. Annual growth rate in GDP has accelerated from below 6 percent during the initial years of reforms to more than 8 percent in recent years. This happened mainly due to rapid growth in non-agriculture sector. The workforce engaged in agriculture between 1980-81 and 2006-07 witnessed a very small decline; from 60.5 percent to 52 percent.

With recent advancements in engineering and technology, there also have been changes in agricultural technology and practices. Though there are advancements people still follow old practices due to the lack of money and high cost of technically advanced agricultural equipment's. The literature review is carried out by majority of the research scholars to make agriculture easier for farmers. However now a days find workers and labor is also another problem. Thus, this project handles a wide scope as 70% of the Indian population directly or indirectly depends on agriculture. The project proposes the concept of Automated agriculture system. The proposed project consists of android application as well as web application which will monitor different parameters in the field and inform the farmer over internet using IOT protocols. The proposes system can also permit the farmer to control the different parameters using the android application itself over internet. The system consists of an automatic farm task execution where the farmer can set threshold for tasks such as irrigation. Once the value of sensor drops below the threshold, the system will automatically activate the irrigation system. Thus, the proposed project helps in carrying out agriculture practices without the burden on farmers.

Chapter 2

LITERATURE SURVEY

2.1 Existing system

Before starting with the project, a brief study was made on the currently existing systems in this domain. The work carried out by notable research scholars is studied in depth and its excerpt is documented here in literature review.

The demand for the food crops is more in the present scenario. Now a day the cultivation of the crops in the greenhouse under specified conditions which is suitable for the crops is increased. G. Sandhi et al., in [1] have proposed visually guided operations in green-houses. A vision system to operate in a greenhouse environment designed for tomato cultivation is explained. Here in this work, they are used two PAL color cameras. The signal sent by the cameras is processed by a graphic workstation using a bit-slice microprocessor card for fast image processing.

K. Rangan et al., in [2] have discussed An Embedded Systems Approach to Monitor Green House. They are used an embedded system approach to monitor and control the greenhouse parameters. They are measuring humidity, temperature, pH of the water, soil wetness and light intensity by sensors. The message will be sent to the owner through GSM. The disadvantage of this work is few parameters are measured and the message will not be in local language. So, the uneducated people cannot be able to use this system.

Wei Ai et al., in [3] have proposed Green House Environment Monitor Technology Implementation Based on Android Mobile Platform. China is a large agricultural nation. And agriculture is the foundation of national economy. They have implemented a technology which uses mobile phone as monitoring terminal, monitoring greenhouse environment explained. In this system they used two sensors-temperature and humidity sensors. Sensors are cable type. GPRS is used to send messages. Wireless sensor network is an emerging field that can be used to monitor and control the agricultural parameters in order to make intelligent automated system in greenhouse.

Akshay et al., in [4] have proposed Wireless sensing and control for precision Greenhouse management they used a CPU for monitoring and a ZigBee with PIC microcontroller to establish a wireless communication between two distant locations. The range of the ZigBee is limited. Their main purpose is to monitor and control only the temperature and humidity. AjiHanggoro et al., in [5] have discussed Green House Monitoring and Controlling Using Android Mobile Application. The new system developed to test the indoor humidity. Complete system is designed to monitor and control the humidity inside the greenhouse. The software used is an android phone, connected using WIFI to a central server which connected via serial communication to microcontroller and humidity sensor. S. Thenmozhiet al., in [6] have discussed Greenhouse Management Using Embedded System and ZigBee Technology. The controlling process takes place by both manual and automatic manner. ZigBee wireless network will send status to the control room. And there we can control the activities through PC.M.K. Gayatri et al., in [7] have discussed Providing Smart Agricultural Solutions to Farmers for better yielding using IOT. They explained about the IOT concept. The issues related to the farmers are hampering the cause of our evolution. One of the solutions for these problems is to help farmers using modernization techniques. This paper explains combining the advantages of the major characteristics of emerging technologies such as IOT and web service. P. S. Asolkaret al., in [8] have discussed An Effective Method of Controlling the Greenhouse and Crop Monitoring Using GSM. The greenhouse approach has been presented supporting GSM wireless technology. This effectively monitors and controls the temperature, humidity, soil moisture, light intensity and CO2 gases. Here the GSM will send message. And the owner must reply back to take an appropriate action. Viswanath naik et al., in [9] have presented IOT based greenhouse monitoring system The monitoring of the vital parameters of greenhouse namely temperature and soil moisture through IOT is explained. irrespective of our place where we are, we can control the parameters.

Apart from research papers the existing technology was studied which is documented below.

Greenhouses in India are being deployed in the high-altitude regions where the sub-zero temperature up to - 40° C makes any kind of plantation almost impossible and in arid regions where conditions for plant growth are hostile. The existing set-ups primarily are:

MANUAL SET-UP: This set-up involves visual inspection of the plant growth, manual irrigation of plants, turning ON and OFF the temperature controllers, manual spraying of the fertilizers and pesticides. It is time consuming, vulnerable to human error and hence less accurate and unreliable.

PARTIALLY AUTOMATED SET-UP: This set-up is a combination of manual supervision and partial automation and is similar to manual set-up in most respects but it reduces the labor involved in terms of irrigating the set-up.

FULLY- AUTOMATED: This is a sophisticated set-up which is well equipped to react to most of the climatic changes occurring inside the greenhouse. It works on a feedback system which helps it to respond to the external stimulation efficiently. Although this set-up overcomes the problems caused due to human errors it is not completely automated and expensive.

Chapter 3

OBJECTIVES

1. To develop an agricultural automation system which can use raspberry pi to automate and control the agricultural operation based on sensor data
2. To develop an IOT cloud panel to monitor the sensor data
3. To Implement feed input system which will automatically feed the supplement to the farm at set time thereby maintaining the fertility of the soil

The objective of this project is to design and develop an automated plant irrigation system using Raspberry Pi and IoT technologies to ensure efficient water management and improve plant care. The primary aim is to automate the irrigation process by continuously monitoring soil moisture levels and delivering water precisely when needed, reducing wastage while maintaining optimal soil hydration. The system also seeks to empower users with remote monitoring and control capabilities through an IoT platform, enabling real-time tracking of soil moisture, temperature, and humidity, as well as the ability to adjust irrigation schedules from anywhere. Furthermore, it incorporates data logging to analyse trends in environmental conditions and irrigation patterns, offering valuable insights to optimize water usage over time.

The project emphasizes scalability, allowing for the support of multiple plants or irrigation zones, as well as cost-effectiveness and energy efficiency by using affordable, low-power components and integrating renewable energy sources like solar panels. Additional features such as weather-based irrigation adjustments, mobile notifications for system alerts, and machine learning integration for predictive irrigation based on plant and weather data can enhance its functionality. Ultimately, this project aims to provide an intelligent, sustainable, and user-friendly solution for modern gardening and agriculture, benefiting both small-scale home gardens and larger agricultural operations.

Chapter 4

WORKING PRINCIPLE AND METHODOLOGY

4.1 System Overview

The system consists of automated agriculture framework using IOT and android. The proposed system consists of development of Automated system for the agriculture. proposed system consists of different sensors, which will read different parameters in the field including soil moisture, humidity, temperature, rainfall and notify the farmer regarding the same over the time using IOT application. The parameters can be automated using threshold adjustment system, which will check the required parameters automatically and trigger the irrigation or other parameter adjustment system to provide the necessary parameters for crops.

The data collected form all the sensor nodes can be viewed in real-time in the web application developed as a part of the project. The IOT system developed can show or display the real-time data. The system includes the feed supply system which permits the farmer to provide feed supplement to the farm every 1.5 months. Provision is also made to control the feed supply manually over the cloud using the IOT application developed. The System permits the farmer to control the parameters in the field over internet using two-way control. The proposed system has threshold setting system which permits the farmers to automate the parameters using sensor thresholds which can be activated and deactivated based on the sensor data. The system prototype is developed in such a way that excess water from irrigation is collected from the drip pipes in the irrigation tank to be reutilized again.

4.2 Working Principle:

The automated irrigation system functions by integrating sensors, actuators, and IoT capabilities to optimize water usage for plants. The system's core idea revolves around monitoring soil moisture levels and automating the irrigation process based on predefined thresholds.

1 Data Collection

The process begins with sensors placed in the soil to continuously measure its moisture level. These sensors send electrical signals to the Raspberry Pi, translating them into readable data. Additional environmental sensors, such as temperature and humidity sensors, can be included to collect more comprehensive information about the surroundings. This data provides a clear understanding of the soil and environmental conditions, which forms the basis for decision-making.

2 Decision-Making

The Raspberry Pi processes the sensor data in real-time using Python scripts. It compares the current soil moisture levels against preset thresholds, determining whether the soil is too dry or sufficiently moist. If the soil moisture falls below the threshold, the system recognizes the need for irrigation. This decision-making process ensures water is supplied only when necessary, promoting efficient water usage.

3 Irrigation Control

Once the system decides irrigation is needed, the Raspberry Pi sends a signal to a relay module connected to a water pump. The relay acts as a switch, turning the pump on to deliver water to the plants. As the soil moisture increases, the system continuously monitors the sensors and automatically shuts off the pump once the desired moisture level is achieved. This closed-loop control mechanism ensures that plants receive adequate water without over-irrigation.

4 IoT Integration

The IoT component enhances the system’s capabilities by enabling remote monitoring and control. Sensor data and system status are transmitted to a cloud platform via Wi-Fi, where users can view real-time updates through a web or mobile application. The IoT platform can also send alerts to users about critical events, such as low water levels, sensor malfunctions, or irregularities in the irrigation process. Users can manually override the system through the application if required, providing additional flexibility.

5 Feedback and Optimization

The system includes feedback mechanisms to ensure reliability. For example, a water flow sensor can confirm that water is being delivered correctly, and any discrepancies can trigger alerts. The collected data is stored in a database for future analysis, helping to refine irrigation schedules and optimize water usage patterns over time.

This automated process not only reduces water wastage but also minimizes human intervention, making it a practical and efficient solution for sustainable irrigation management.

4.3 Methodology

The methodology to develop and implement an automated irrigation system using Raspberry Pi and IoT involves several systematic steps, each contributing to the overall functionality and efficiency of the system. These steps ensure that the system is reliable, scalable, and capable of optimizing water usage.

1. Requirement Analysis and Planning

The first step in the methodology is to analyse the requirements of the irrigation system. This involves identifying the specific needs of the plants, such as the type of crops, their water requirements, and the size of the area to be irrigated. Additionally, environmental factors like soil type, climate conditions, and the availability of resources such as water and power are taken into account. Based on this analysis, the hardware and software components required for the system are selected.

2. Hardware Setup

The hardware setup is a crucial step where all physical components are assembled. The soil moisture sensor is placed in the soil to measure its moisture content. A relay module is connected to the Raspberry Pi to control the water pump, which delivers water to the plants. Additional sensors, such as temperature and humidity sensors, can also be included for enhanced monitoring. The Raspberry Pi serves as the central processing unit, interfacing with all the sensors and actuators. Power supplies for the Raspberry Pi and other components are set up to ensure uninterrupted operation.

3. Software Development

The software is developed to control the system and process sensor data. Python programming is typically used to write scripts for reading sensor values, making decisions based on the collected data, and controlling the water pump. The software also integrates IoT functionalities, allowing the system to send data to a cloud platform. Libraries and APIs for MQTT or HTTP communication are incorporated to enable seamless data transmission and remote control.

4. Integration with IoT Platform

The IoT platform integration enhances the system’s capabilities by enabling remote access and monitoring. Data from sensors is transmitted to a cloud platform like Thing Speak, Blynk, or AWS IoT. This platform provides a dashboard where users can view real-time data such as soil moisture levels, temperature, and system status. The IoT platform also facilitates remote control of the system, allowing users to manually activate or deactivate irrigation if necessary. Alerts and notifications are configured to inform users of critical events like low water levels or system failures.

5. Testing and Calibration

Once the hardware and software are integrated, the system is tested to ensure its functionality. The sensors are calibrated to provide accurate readings, and thresholds for soil moisture levels are fine-tuned. Testing involves simulating different environmental conditions to evaluate the system's responsiveness and reliability. Any issues identified during testing are resolved to ensure the system operates smoothly in real-world scenarios.

6. Deployment

After successful testing, the system is deployed in the field. Sensors are installed at appropriate locations to monitor soil moisture effectively, and the water pump is connected to the irrigation network. The IoT platform is set up for regular monitoring, and users are provided with access to the dashboard or mobile application for interaction with the system.

7. Maintenance and Optimization

Post-deployment, the system requires regular maintenance to ensure its longevity and performance. Sensor accuracy and hardware components are periodically checked. The data collected over time is analysed to optimize irrigation schedules and improve water efficiency. Updates to the software and IoT platform can be made to incorporate new features or enhance functionality.

This methodology ensures a structured approach to developing an automated irrigation system that is reliable, scalable, and efficient in conserving water and supporting sustainable agriculture.

Chapter 5

HARDWARE AND SOFTWARE DESCRIPTION

5.1 System architecture

The figure below shows the architecture diagram of the system:

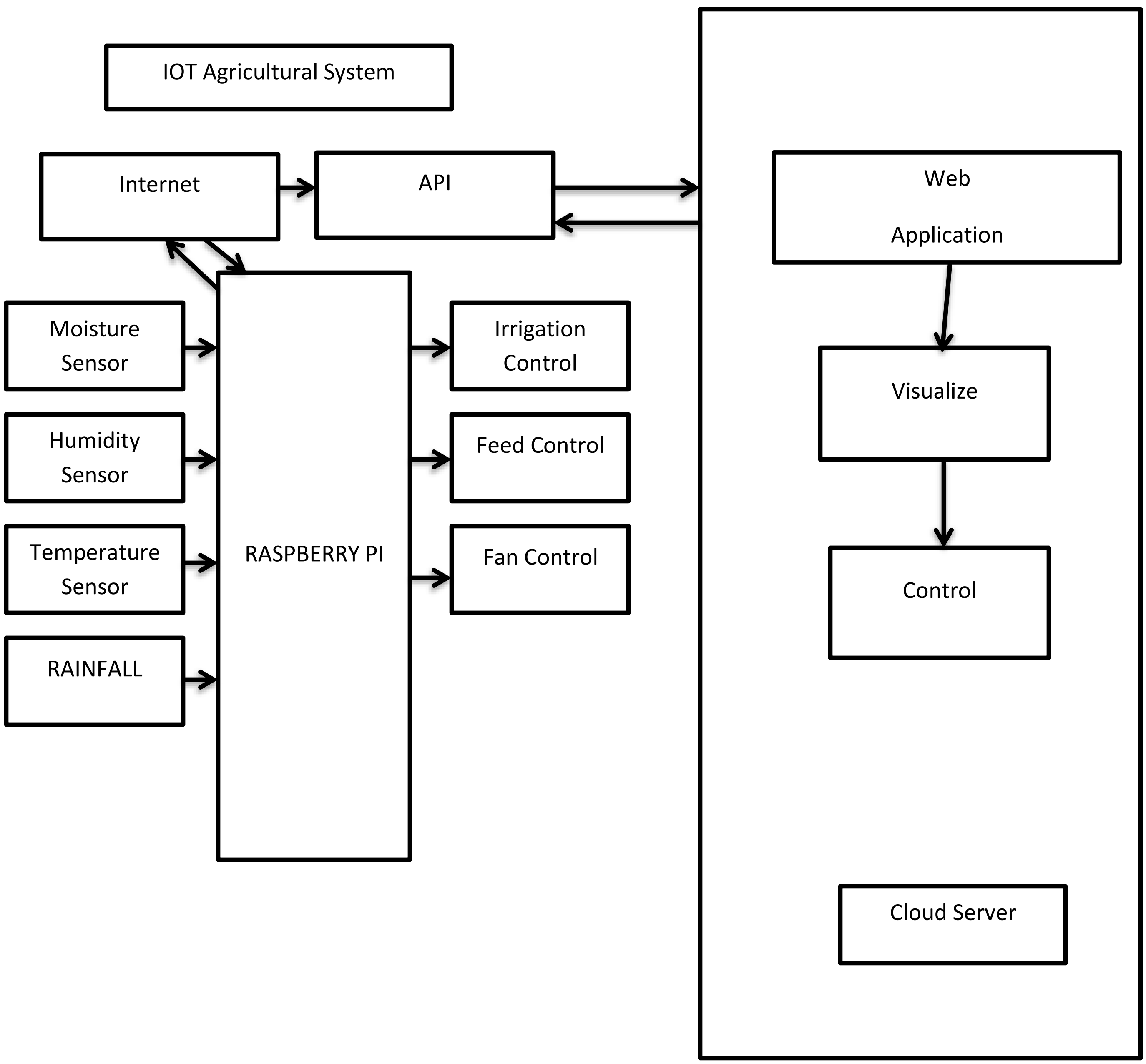

Fig 5.1.1 Block diagram

5.2 Language Description

About Python

Python is a versatile and high-level programming language that has gained immense popularity across various domains such as web development, data science, machine learning, and IoT. Known for its simplicity and readability, Python enables developers to write clear and concise code, making it an ideal choice for both beginners and experienced programmers.

One of Python's standout features is its extensive collection of libraries and frameworks. These tools provide pre-written code to simplify complex tasks, such as data analysis (using libraries like NumPy and Pandas), machine learning (with TensorFlow and PyTorch), and web development (using Django and Flask). Additionally, Python’s libraries for hardware control, such as RPi.GPIO and GPIO Zero, make it a preferred language for IoT applications.

Python is an interpreted language, meaning that it executes code line by line, which simplifies testing and debugging. It is also platform-independent, allowing programs written in Python to run seamlessly on various operating systems, including Windows, macOS, Linux, and Raspberry Pi. This cross-platform capability makes it especially useful in IoT projects where multiple devices may need to communicate.

Python is highly dynamic, as it determines variable types at runtime rather than during compilation. While this flexibility speeds up development, it requires developers to carefully manage variable types to prevent runtime errors. Its syntax is intuitive and closely resembles natural language, making it easy to learn and widely accessible to developers from diverse backgrounds.

The Python community plays a significant role in its success. A vast network of developers contributes to its growing ecosystem by creating libraries, tools, and tutorials. This strong community support ensures that developers can find solutions to problems and continuously improve their Python skills.

In the context of IoT and automation, Python excels due to its ability to control hardware efficiently and integrate with cloud platforms. Libraries like paho-mqtt enable Python to handle

MQTT communication, while others like requests allow seamless interaction with REST APIs. These features, combined with Python's ease of use, make it a powerful language for developing automated systems, such as an irrigation system that leverages Raspberry Pi and IoT technologies.

Python is a high-level, interpreted, and general-purpose programming language known for its simplicity, readability, and versatility. It was created by Guido van Rossum and first released in 1991. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming, making it suitable for a wide range of applications.

Key Features:

1. **Readability**: Python uses an indentation-based syntax, which promotes clean and easily understandable code.

2. **Interpreted Language**: Code is executed line by line, without the need for compilation, making it quick to develop and test.

3. **Dynamic Typing**: Variable types are inferred at runtime, allowing for flexibility in coding.

4. **Extensive Libraries**: Python has a vast standard library and supports additional libraries for tasks ranging from web development to data analysis.

5. **Platform Independence**: Python code can run on various operating systems with minimal modification.

6. **Community Support**: It has a large and active developer community, providing abundant resources and third-party modules.

Usage:

- **Web Development**: Frameworks like Django and Flask make Python a popular choice for building web applications.
- **Data Science and Machine Learning**: Libraries such as NumPy, pandas, TensorFlow, and PyTorch are widely used.
- **Automation and Scripting**: Python excels in writing scripts for automating repetitive tasks.
- **Game Development**: Libraries like Pygame allow for creating simple games.
- **System Administration**: Python scripts are used for managing servers and system resources.

Strengths:

- **Ease of Learning**: Python is beginner-friendly due to its straightforward syntax.
- **Versatility**: It can be used in various fields, including AI, web development, and scientific computing.
- **Rapid Development**: The language enables fast prototyping and iteration.

Limitations:

- **Speed**: Python is generally slower than compiled languages like C++ or Java.
- **Mobile Development**: Python is not natively suited for mobile application development.

Python has established itself as one of the most popular programming languages, powering a wide array of modern applications and tools across industries. Its simplicity and robust capabilities make it a preferred choice for both beginners and experienced developers.

PHP (Hypertext Preprocessor):

PHP is a general-purpose scripting language originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the html source document and interpreted by a web server with PHP processor module, which generates the web page document. It also has evolved to include a command line interface capability and can be used in standalone graphical application.

PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of a HTTP response. Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response. Additionally, PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications[9] and robotic drone control. Arbitrary PHP code can also be interpreted and executed via command-line interface (CLI).

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, with the original implementation acting as the de facto standard which other implementations aimed to follow. Since 2014, work has gone on to create a formal PHP specification.

5.3 Software Description

Thonny IDE:

Thonny is a beginner-friendly integrated development environment (IDE) specifically designed for Python programming. It's simple and intuitive interface makes it an excellent choice for individuals new to coding, as well as for educators teaching Python. Thonny focuses on providing a distraction-free environment with only the most essential tools displayed, enabling users to concentrate on their code.

Thonny is a lightweight and user-friendly Integrated Development Environment (IDE) specifically designed for Python programming, making it an excellent choice for beginners and educational purposes. Developed by **Aivar Annamaa**, Thonny provides a simple interface with features like a built-in Python interpreter, automatic syntax highlighting, and an intuitive debugger that makes understanding program execution easy. It allows users to step through their code line by line, visualize variable states, and track changes in a beginner-friendly way. Thonny also supports managing virtual environments, enabling hassle-free experimentation with different Python packages. Its clean and minimalistic design ensures a distraction-free coding experience, while advanced users can benefit from features like code completion and syntax checking. Thonny is widely appreciated in the educational community for lowering the entry barrier to Python programming while providing tools that foster learning and exploration.

One of Thonny's key features is its built-in Python interpreter, which eliminates the need for separate installations and configurations. This integration allows users to start writing and executing Python programs immediately after installing the IDE. Additionally, its code editor supports syntax highlighting, auto-completion, and error highlighting, which enhance the overall coding experience and help in writing clean, efficient code.

Thonny is well-known for its step-by-step debugging feature, which allows users to visualize their code execution line by line. This is particularly helpful for beginners, as it makes it easier to understand the logic and flow of a program while identifying and fixing errors. Furthermore, the IDE includes a variable explorer that displays real-time changes to variables during program execution, providing a deeper insight into how data is manipulated in the code.

For IoT and hardware-related projects, Thonny is an excellent choice due to its seamless compatibility with Raspberry Pi. Pre-installed on Raspberry Pi OS, it allows developers to write Python code for controlling GPIO pins, sensors, and actuators directly. This makes it ideal for projects like automated irrigation systems and other IoT applications. Additionally, Thonny supports microcontroller programming, enabling users to work with Micro Python or Circuit Python-based devices without requiring extensive setup.

Thonny's simplicity, combined with its powerful debugging and hardware integration features, makes it a versatile IDE for both learning Python and developing real-world applications. Whether used by students for educational purposes or professionals for IoT projects, Thonny provides a user-friendly and efficient platform for Python programming.

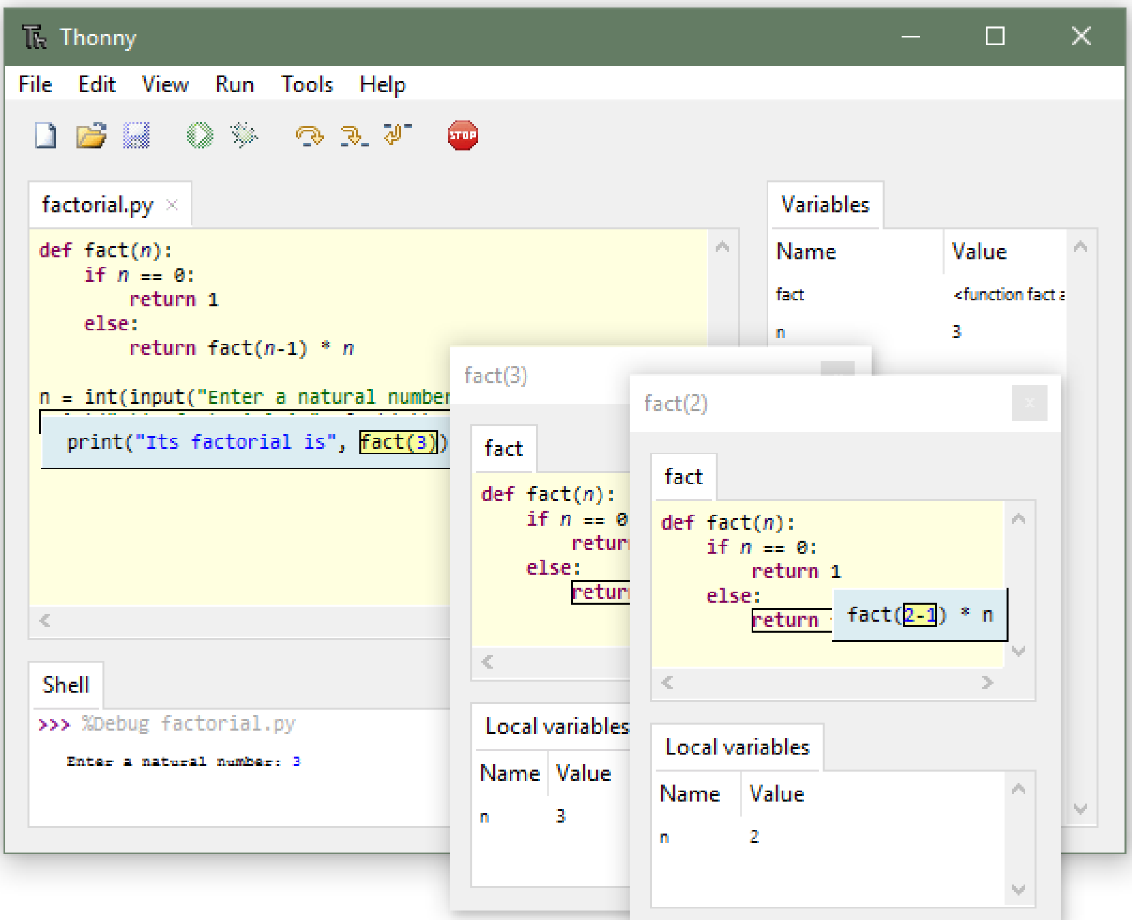

Fig 5.3.1 Thonny IDE

Easy EDA:

EasyEDA, short for Easy Electronic Design Automation, is a powerful, web-based platform designed for creating, simulating, and sharing electronic circuit designs and printed circuit boards (PCBs). It is widely used by electronics hobbyists, students, and professionals due to its accessibility, ease of use, and comprehensive feature set. By integrating various tools into one platform, EasyEDA simplifies the process of turning electronic ideas into reality.

One of EasyEDA's core features is its schematic capture tool, which enables users to create detailed circuit diagrams effortlessly. The platform offers a vast library of pre-built components such as resistors, capacitors, ICs, and connectors, along with the ability to create custom symbols. This makes it easy to design circuits ranging from simple prototypes to complex systems. Additionally, the tool allows users to validate their designs using built-in SPICE simulation for analysing circuit behaviour, ensuring functionality before proceeding to hardware implementation.

The PCB design module in EasyEDA provides a robust environment for designing layouts. It supports multi-layer PCBs and includes tools for routing traces, placing components, and creating ground planes. The software also incorporates design rules and error checks, ensuring that layouts meet manufacturing standards. EasyEDA’s integration with manufacturers like JLCPCB further streamlines the process by allowing users to order custom PCBs directly from their designs.

EasyEDA’s cloud-based nature makes it accessible from any device with a web browser, including Windows, macOS, and Linux. Users can also opt for an offline desktop version if needed. The platform supports collaboration by enabling users to share projects and work on designs with team members. This feature is particularly useful for group projects or professional design teams.

A key advantage of EasyEDA is its seamless integration with LCSC, a leading electronic component supplier, and JLCPCB, a PCB manufacturing service. This integration enables users to order components and PCBs directly from the platform, reducing the time and effort required to move from design to production. The platform also offers cost-effective solutions for prototyping and mass production, making it ideal for IoT projects, educational purposes, and professional applications.

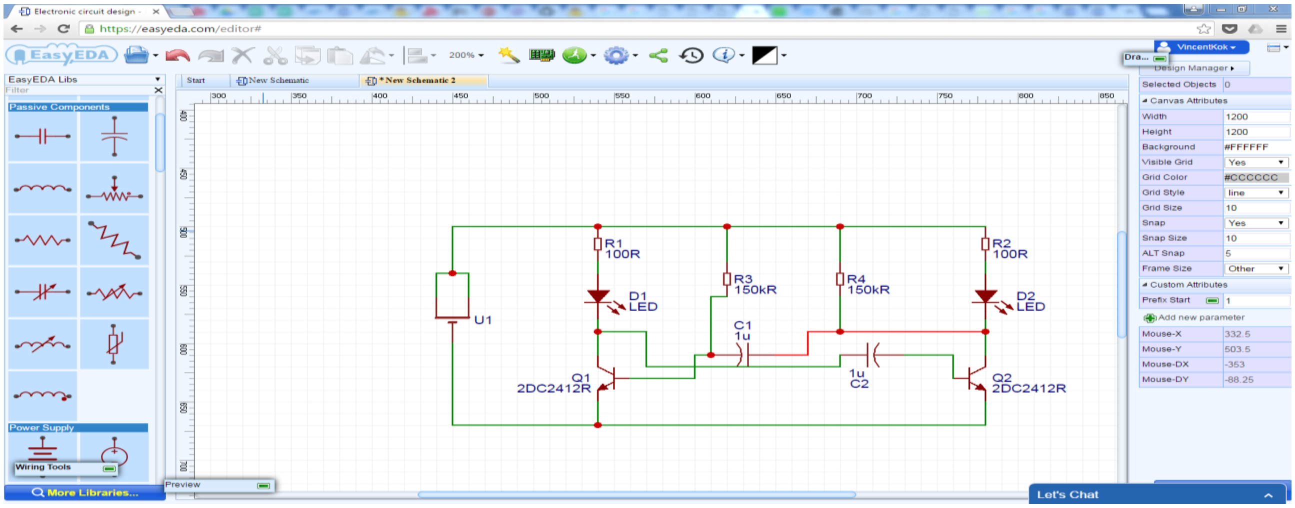

Fig 5.3.2 Easy EDA

WAMP server

Fig 5.3.3 logo of WAMP server

WAMP Server 2.0 is a collection of web development tools &software's. It provides an environment for developing web pages &applications. It contains Apache Web Server, MySQL Database Management System &PHP Programming Language. Sonow you can develop your applications locally on you home PC and once you havedeveloped your applications you can upload it to your webhost. Wamp Server 2.0 alsoprovides some nice little tools for easy management of your databases, php my adl11in &SqlLite Manager are already installed. Interface of Wamp Server 2.0 is neat and clean. Being an open-source software, you can customize it the way you want. Most of thesettings of this software can be accessed using a menu. You can directly access thesesetting right from the taskbar. It is available in around 20+ languages. You can alsoupdate it automatically using the menu from the taskbar. Apache & MySQL are the most popular software used in web development and if you use PHP as your language for developing web application then this software is a must.

PHP MyAdmin

phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. phpMyAdmin supports a wide range of operations on MySQL and MariaDB. Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc) can be performed via the user interface, while you still have the ability to directly execute any SQL statement.

Features

- Intuitive web interface
- Support for most MySQL features:
  - browse and drop databases, tables, views, fields and indexes
  - create, copy, drop, rename and alter databases, tables, fields and indexes
  - maintenance server, databases and tables, with proposals on server configuration
  - execute, edit and bookmark any SQL-statement, even batch-queries
  - manage MySQL user accounts and privileges
  - manage stored procedures and triggers
- Import data from CSV and SQL
- Export data to various formats: CSV, SQL, XML, PDF, ISO/IEC 26300 - OpenDocument Text and Spreadsheet, Word, $\LaTeX$ and others
- Administering multiple servers
- Creating graphics of your database layout in various formats
- Creating complex queries using Query-by-example (QBE)
- Searching globally in a database or a subset of it
- Transforming stored data into any format using a set of predefined functions, like displaying BLOB-data as image or download-link

Brackets IDE

Brackets is a lightweight, yet powerful, modern text editor. We blend visual tools into the editor so you get the right amount of help when you want it without getting in the way of your creative process. You'll enjoy writing code in Brackets.

Fig 5.3.4 Bracket IDE

5.4 Hardware Description

The following hardware components are used in this project:

1. **Buzzer**

Buzzers can be both fun and useful in electric circuits. We'll use them a lot in MakeCrate projects, so let's take a look at what is going on inside a buzzer to produce sound. The buzzer consists of an outside case with two pins to attach it to power and ground. Inside is a piezo element, which consists of a central ceramic disc surrounded by a metal (often bronze) vibration disc.

When current is applied to the buzzer it causes the ceramic disk to contract or expand. Changing the This then causes the surrounding disc to vibrate. That's the sound that you hear. By changing the frequency of the buzzer, the speed of the vibration's changes, which changes the pitch of the resulting sound.

Fig 5.4.1 Buzzer

2. I2C module

To connect LCD display 16×2 or 20×4 to Arduino you know you'll need at least 6 wires to connect, it means sacrificing some IO's that could be used for connecting other components such as sensors or motors. another way is to use 74HC595 Shift register for interfacing.

In the module left side we have 4 pins, and two are for power (Vcc and GND), and the other two are the interface I2C (SDA and SCL). The plate pot is for display contrast adjustment, and the jumper on the opposite side allows the back light is controlled by the program or remain off for power saving. By default, the module is configured with the address 0x27, but you can change this address using the pins A0, A1 and A2 Jumper settings.

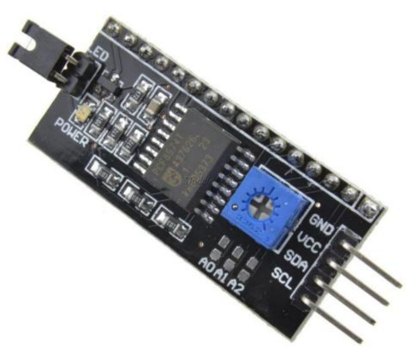

Fig 5.4.2 I2C Module

3. LCD display

The liquid-crystal display has the distinct advantage of having a low power consumption than the LED. It is typically of the order of microwatts for the display in comparison to some order of milliwatts for LEDs. Low power consumption requirement has made it compatible with MOS integrated logic circuit. Its other advantages are its low cost, and good contrast. The main drawbacks of LCDs are additional requirement of light source, a limited temperature range of operation (between 0 and 60° C), low reliability, short operating life, poor visibility in low ambient lighting, slow speed and the need for an ac drive.

A liquid crystal cell consists of a thin layer (about 10 u m) of a liquid crystal sand-wiched between two glass sheets with transparent electrodes deposited on their inside faces. With both glass sheets transparent, the cell is known as *transmittive type cell.* When one glass is transparent and the other has a reflective coating, the cell is called *reflective type.* The LCD does not produce any illumination of its own. It, in fact, depends entirely on illumination falling on it from an external source for its visual effect

Fig 5.4.3 LCD Display

4. Raspberry Pi 3 Model B+

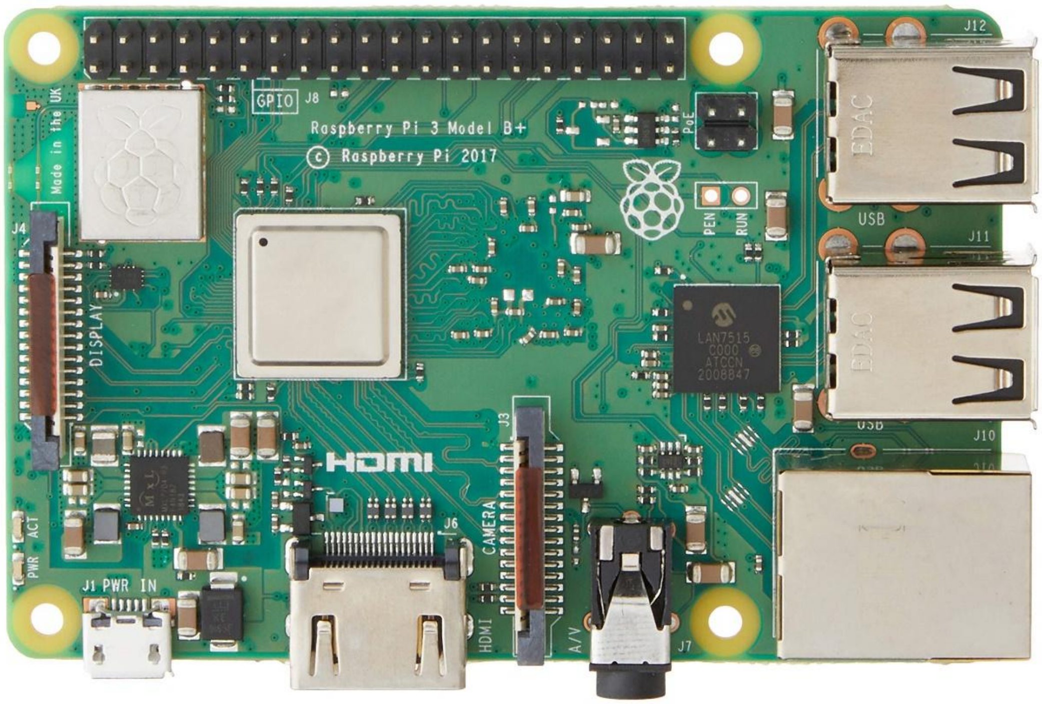

Fig 5.4.4 Raspberry pi 3

The **Raspberry Pi 3 Model B+** is a versatile, credit-card-sized single-board computer designed by the Raspberry Pi Foundation. It is an upgraded version of the Raspberry Pi 3 Model B and offers enhanced performance, connectivity, and efficiency, making it suitable for a wide range of applications, from educational projects to IoT systems and media centers.

1. **Key Specifications:**

- **Processor**: Broadcom BCM2837B0, a 64-bit quad-core ARM Cortex-A53 processor clocked at 1.4 GHz, providing improved processing power over its predecessor.
- **Memory**: 1 GB LPDDR2 SDRAM.
- **Networking**: Enhanced networking capabilities with built-in **802.11ac Wi-Fi** (dual-band 2.4 GHz and 5 GHz), **Gigabit Ethernet (via USB 2.0)**, and Bluetooth 4.2, making it ideal for IoT and network-intensive applications.
- **USB Ports**: Four USB 2.0 ports for connecting peripherals such as keyboards, mice, or storage devices.
- **HDMI Output**: Full-size HDMI port for connecting to monitors or TVs.
- **GPIO**: 40-pin General Purpose Input/Output (GPIO) header for interfacing with external hardware, sensors, and custom circuits.
- **Storage**: Bootable microSD card slot for operating system and data storage.
- **Power Supply**: 5V/2.5A micro-USB power supply.
- **Video & Audio**: Composite video and stereo audio output via a 3.5mm jack.

- **Camera & Display Interface**: Dedicated ports for connecting the Raspberry Pi Camera Module and Display.

2. **Features and Advantages:**
3. **Improved Performance**: The 1.4 GHz processor and better thermal management enhance the performance for demanding applications.
4. **Enhanced Connectivity**: Faster Wi-Fi and Ethernet improve data transfer rates, enabling better integration with IoT systems and networks.
5. **Backward Compatibility**: Supports earlier Raspberry Pi add-ons and accessories, making it flexible for upgrades.
6. **Energy Efficiency**: Consumes low power, making it suitable for battery-powered projects.
7. **Applications:**

- **IoT and Home Automation**: Acts as a hub for IoT devices with its robust connectivity options.
- **Education and Prototyping**: Ideal for learning programming, electronics, and building prototypes.
- **Media Centers**: Powers media center solutions like Kodi for streaming and entertainment.
- **Embedded Systems**: Used in robotics, weather stations, and other embedded applications.

The Raspberry Pi 3 Model B+ is a powerful and cost-effective tool for enthusiasts, students, and professionals alike, offering a balance of performance and features tailored to modern computing and automation needs.

5. **DHT11 Humidity and temperature sensor**

This comes with calibrated digital signal output. This comes with NTC temperature measurement component and resistive type humidity measurement component. It offers excellent quality, high performance and fast response also it is cost effective. This sensor is extremely accurate on humidity calibration.it is small size, low power consumption and upto 20-meter signal transmission.

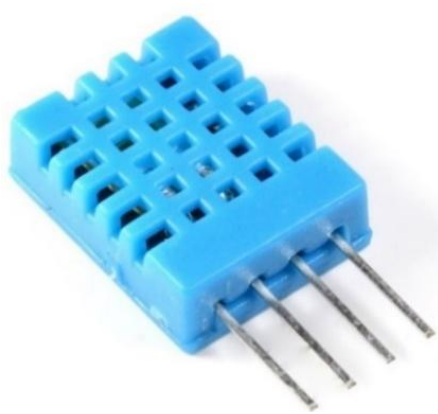

Fig 5.4.5 Humidity & Temperature Sensor

Parameters	Conditions	Minimum	Typical	Maximum
Humidity				
Resolution		1%RH	1%RH	1%RH
			8 Bit	
Repeatability			±1%RH	
Accuracy	25℃		±4%RH	
	0-50℃			±5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0℃	30%RH		90%RH
	25℃	20%RH		90%RH
	50℃	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25℃, 1m/s Air	6 S	10 S	15 S
Hysteresis			±1%RH	
Long-Term Stability	Typical		±1%RH/year	
Temperature				
Resolution		1℃	1℃	1℃
		8 Bit	8 Bit	8 Bit
Repeatability			±1℃	
Accuracy		±1℃		±2℃
Measurement Range		0℃		50℃
Response Time (Seconds)	1/e(63%)	6 S		30 S

Fig 5.4.6 parameters table

6. Soil moisture sensor

This sensor is used to test moisture level in soil. When this component come in contact with soil it will send signal to the micro controller. If soil contains water, then it will show output at high level and when soil contains no water then it shows output at low level. This comes with triple output mode digital output is simple output, analog output is more accurate and serial output is exact same reading.

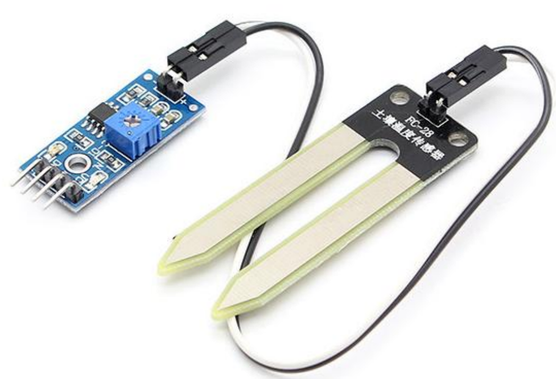

Fig 5.4.7 Soil Moisture Sensor

This module comes with probe which has multiple soil moisture sensor on it. The common type of soil moisture sensor is capacitance sensor. Soil moisture is detected via is effect on dielectric constant by measuring the capacitance between two electrodes implanted in soil.

7. **Relay board**

Relays are called electromechanical switches. They have very high current rating and both AC motors and DC motors can be controlled through them. Because the motors are completely external part of the complete system

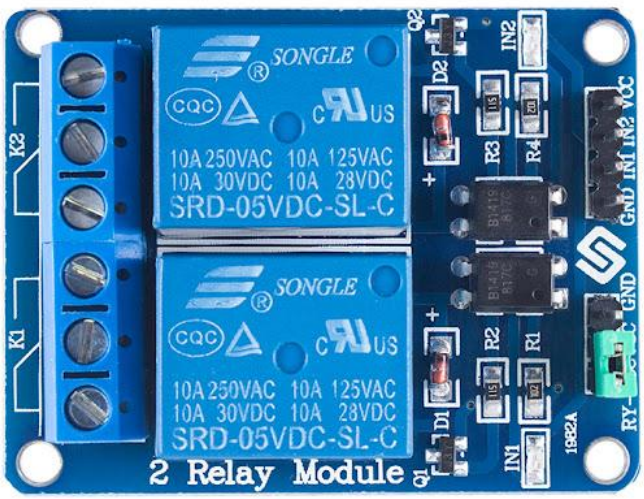

Fig 5.4.8 Channel Relay Board

8. **DC Pump:**

A 12 V dc pump is used to irrigate the field. It is also used to supply the feed from the feed tank to the field.

Fig 5.4.9 DC Pump

The specifications of the pump are as shown below.

- Rated Voltage: DC 9 V to 12V (1 amps)
- Load current: 0.7A (Max)
- Flow: 1.5 to 2 Ltr/min
- Max Lift: 3m
- Max Suction: 2m
- Max Water Temp: 70 °C
- Pump Size: 90mm * 40mm * 35mm approx
- Input/output tube diameter: outer 8.5mm, inner 6mm approx
- Max Current: Up to 2 Amps while starting up.
- Main Color: Silver Tone, White
- Material: Metal, Plastic
- Life: up to 2500 Hours

9. Rain fall Sensor

A rainfall sensor is an electronic device designed to detect the presence and intensity of rain. It is commonly used in weather monitoring, irrigation systems, and automation applications. The sensor typically consists of a water-sensitive plate that changes its electrical properties when exposed to rain, allowing it to measure rainfall presence or volume.

Key Components:

1. Rain Detection Plate: A surface with conductive traces that detect water droplets through changes in resistance or capacitance.
2. Controller Module: Processes signals from the detection plate and provides an output (digital or analog) for integration with microcontrollers or other systems.
3. Output Signal:
    - Digital Output: Indicates the presence of rain (on/off).
    - Analog Output: Provides a variable signal proportional to the intensity of the rain.
4. Power Supply: Typically operates at 3.3V or 5V, compatible with most microcontroller platforms.

Features:

- Sensitivity Adjustment: Some sensors allow calibration of sensitivity using a potentiometer.
- Compact Design: Lightweight and easy to integrate into various systems.
- Weather Resistance: Designed to endure outdoor environments, often coated to resist corrosion.

Applications:

1. Weather Monitoring: Measures rainfall for weather stations and agricultural purposes.
2. Irrigation Systems: Automatically disables irrigation when rain is detected to conserve water.
3. Home Automation: Triggers actions like closing windows or activating alarms in smart home setups.
4. Automotive Systems: Used in rain-sensing wiper systems in vehicles.

Working Principle:

The rainfall sensor detects water when it falls on the detection plate. The water bridges the conductive traces, changing the resistance or triggering a short circuit, which the controller module translates into a usable signal.

Rainfall sensors are widely appreciated for their simplicity, reliability, and effectiveness, making them essential components in many weather-dependent automation and monitoring systems

Fig 5.4.10 Rain Sensor

5.5 Pseudo Code

This section deals with pseudocode used in the hardware project.

Function setup ()

{

Start Serial communication

Communicate with WiFi chip

Connect to Internet

Set all sensors as input

Set relay as output

Set LCD as output

Initialize the system

End

}

```
Function loop ()
{
Read data from sensors
Upload it to server
IF (JSON received):
        Control outputs
IF (Threshold active):
                IF (Sensor Data >threshold):
                        Trigger Control
If (rainfall found):
        notify
}
If (feed time> 1.5 Month):
        Provide Supplement
}
END
```

Chapter 6

TESTING AND VALIDATION

Testing is a process, which reveals errors in program. It is the measure quality measure employed during software development. During testing, the program is executed with a set of conditions known as test cases and output is evaluated to determine whether the program is performing as expected. The Primary and Larger objective of testing is to deliver quality software. Quality software is one that is devoid of errors and meets with a customer's stated requirements.

If errors are found then the software must be debugged to locate these errors in the various programs. Corrections are then made. The program/system must be tested once again after corrections have been implemented – this time with an additional objective of finding out whether or not corrections in one part of system have introduced any new errors elsewhere in the system.

Once all errors are found, then another objective must be accomplished that is check whether or not the system is doing what it is supposed to do. So, another aspect of testing is that it must also ensure that the system meets with user requirements.

Techniques of testing are given below:

- Black Box Testing
- White Box Testing
- Equivalence Portioning
- Boundary Value Analysis
- Ad-hoc Testing

Testing Strategies

A testing strategy is a general approach to the testing process rather than a method of devising particular system or components tests. Different strategies may be adopted depending on the

type of system to be tested and the development process used.

The testing strategies which discuss in this are:

- **Top-down testing** where testing starts with the most abstract component and works downwards.
- **Bottom-up testing** where testing starts with the fundamental components and works upwards.
- **Thread testing** which is used for systems with multiple processes where the processing of transaction threads its way through these processes.
- **Stress testing** which relies on stressing the system by going beyond its specified limits and hence testing how well the system can cope with over-load situations.
- **Back-to-back testing** which is used when versions of system are available the system are tested together and their outputs are compared.

Large systems are usually tested using a mixture of these testing strategies rather than any approach. Different strategies may be needed for different parts of the system and at different stages in the testing process. Whatever testing strategy is adopted, it is always sensible to adopt an incremental approach to sub-system and system testing. Number of software testing strategies is proposed. Testing begins at the module/well and works “outward” towards the integration of the entire computer-based system. Different testing techniques are appropriate at different point of time. The developer of the software and independent test group conducts testing. Testing and debugging must be accommodated in any testing strategy.

6.1 Unit Testing

Unit Testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of software. It usually has one or a few inputs and usually a single output. In procedural programming a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit Testing is normally performed by software developers themselves or their peers. In rare cases it may also be performed by independent software testers.

6.2 Acceptance Testing

Acceptance Testing is a level of the software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

Acceptance testing: Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.

Usually, Black Box Testing method is used in Acceptance Testing. Testing does not normally follow a strict procedure and is not scripted but is rather ad-hoc. Acceptance Testing is performed after System Testing and before making the system available for actual use.

Chapter 7

IMPLEMENTATION

The implementation section describes the implementation of the project in details. This includes the hardware developed, Schematic Developed and PCB.

7.1 The hardware Developed:

The hardware development involves schematic and PCB development using Easy EDA software. The hardware is designed using the Easy EDA schematic capture and then the PCBs are fabricated to make the complete working of the project. The printed circuit board (PCB) acts as a linchpin for almost all of today's modern electronics. If device needs to do some sort of computation-such as is the case even with the simple digital clock. Chances are there is the PCB inside of it. PCBs bring electronics to life by routing electrical signals where they need to go to satisfy all of the device's electronic requirements.

There are three main types of circuit boards that get manufactured on a consistent basis, and it's important to understand the differences between each so you can decide the right circuit board for your requirements. The three main types of circuit boards in current manufacture are:

- **Single-Sided Circuit Boards:** These boards when made with a FR4 base have rigid laminate of woven glass epoxy material, which is then covered on one side with a copper coating that is applied in varying thicknesses depending on the application
- **Double-Sided Circuit Boards:** Double-sided boards have the same woven glass epoxy base as single-sided boards — however, in the case of a double-sided board, there is copper coating on both sides of the board, also to varying thicknesses depending on the application.
- **Multi-Layer Boards:** These use the same base material as single and double-sided boards, but are made with copper foil instead of copper coating — the copper foil is used to make "layers," alternating between base material and copper foil until the number of desired layers is reached.

7.2 Parts of PCB

- **Substrate:** The first, and most important, is the substrate, usually made of fiberglass. Fiberglass is used because it provides core strength to the PCB and helps resist breakage. Think of the substrate as the PCB's "skeleton".
- **Copper Layer:** Depending on the board type, this layer can either be copper foil or a full-on copper coating. Regardless of which approach is used, the point of the copper is still the same — to carry electrical signals to and from the PCB, much like your nervous system carries signals between your brain and your muscles.
- **Solder Mask:** The third piece of the PCB is the solder mask, which is a layer of polymer that helps protect the copper so that it doesn't short-circuit from coming into contact with the environment. In this way, the solder mask acts as the PCB's "skin".
- **Silk screen:** The final part of the circuit board is the silkscreen. The silkscreen is usually on the component side of the board used to show part numbers, logos, symbols switch settings, component reference and test points. The silkscreen can also be known as legend or nomenclature.

The implementation of an automated irrigation system involves turning the planned design into a functional system. It includes assembling hardware, coding software, and integrating IoT services. Below are the detailed steps for implementation:

1. Hardware Assembly

The first step in implementation is assembling the hardware components. A soil moisture sensor is embedded in the soil to measure its moisture content. This sensor is connected to the Raspberry Pi's GPIO pins to enable real-time data collection. A relay module is wired to control the water pump, acting as a switch to turn it on or off based on commands from the Raspberry Pi. The water pump is then connected to an irrigation network, such as drip irrigation or sprinkler systems. Additional sensors like temperature and humidity sensors can also be set up if required. Ensuring proper power supply connections is critical at this stage to avoid malfunctions during operation.

2. Raspberry Pi Configuration

The Raspberry Pi acts as the system's control unit. It is first configured with an operating system, typically Raspberry Pi OS (formerly Raspbian). Once the OS is installed, necessary libraries and tools, such as Python, GPIO libraries, and MQTT or HTTP communication modules, are installed. Network configurations are also set up, ensuring the Raspberry Pi can connect to the internet for IoT integration. Static IP configuration may be done to maintain a consistent address for easier remote access.

3. Sensor and Actuator Integration

The soil moisture sensor is calibrated to provide accurate readings. Its output is tested by placing it in dry and wet soil to determine the moisture thresholds. The relay module is tested to ensure it can control the water pump effectively. Wiring is double-checked for any loose connections or potential short circuits. The entire system is powered on, and the Raspberry Pi is programmed to read inputs from the sensors and send control signals to the relay.

4. Software Implementation

Python scripts are written to automate the decision-making process. These scripts: Read real-time data from the soil moisture sensor. Compare the sensor readings with predefined thresholds. Send signals to the relay to activate or deactivate the water pump based on soil conditions. For IoT integration, the software includes modules for sending sensor data to a cloud platform. Libraries like paho-mqtt for MQTT communication or HTTP-based APIs are used to transmit data. Code is tested in small modules to ensure each component (e.g., reading data, activating the pump, transmitting to IoT) works as intended.

5. IoT Platform Integration

The IoT platform chosen (e.g., Thing Speak, Blynk, AWS IoT) is configured to receive data from the Raspberry Pi. A dashboard is created to visualize sensor data like soil moisture levels, temperature, and humidity. Alerts and notifications are set up for events like low soil moisture or system malfunctions. Remote control options are implemented, enabling users to manually activate or deactivate irrigation via a web or mobile app. The IoT platform's reliability is tested by simulating data and monitoring its updates on the dashboard.

6. Testing and Calibration

Comprehensive testing is conducted to ensure the system functions as expected. Sensors are tested under various conditions, such as dry, moist, and oversaturated soil, to validate the accuracy of readings and the corresponding actions of the water pump. The thresholds for moisture levels are adjusted to suit the specific crop or soil type. IoT features are tested by monitoring the dashboard and triggering remote commands to check their responsiveness.

7. Deployment in the Field

After successful testing, the system is deployed in the field. Sensors are strategically placed in the soil to cover areas with varying moisture conditions. The water pump is connected to a reliable water source, and the irrigation system (e.g., pipes or sprinklers) is set up to distribute water evenly. The Raspberry Pi is placed in a weather-protected enclosure to safeguard it from environmental factors. Power backup, such as a battery or solar panel, can be provided for uninterrupted operation.

8. Monitoring and Maintenance

Post-deployment, the system is monitored to ensure smooth functioning. Data from the IoT dashboard is analyzed to track performance and identify patterns in water usage. Regular maintenance is performed on the sensors, relay module, and water pump to ensure they remain functional. Software updates may be deployed to enhance features or fix bugs. Historical data collected from the IoT platform is analyzed to optimize irrigation schedules and improve water efficiency over time.

9. Schematic Development:

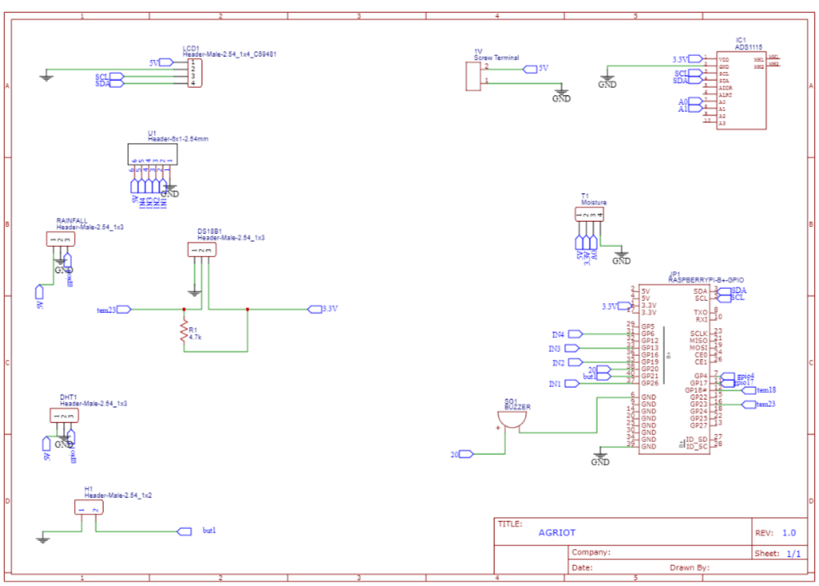

Fig 7.2.1 Schematic development

10. PCB Developed:

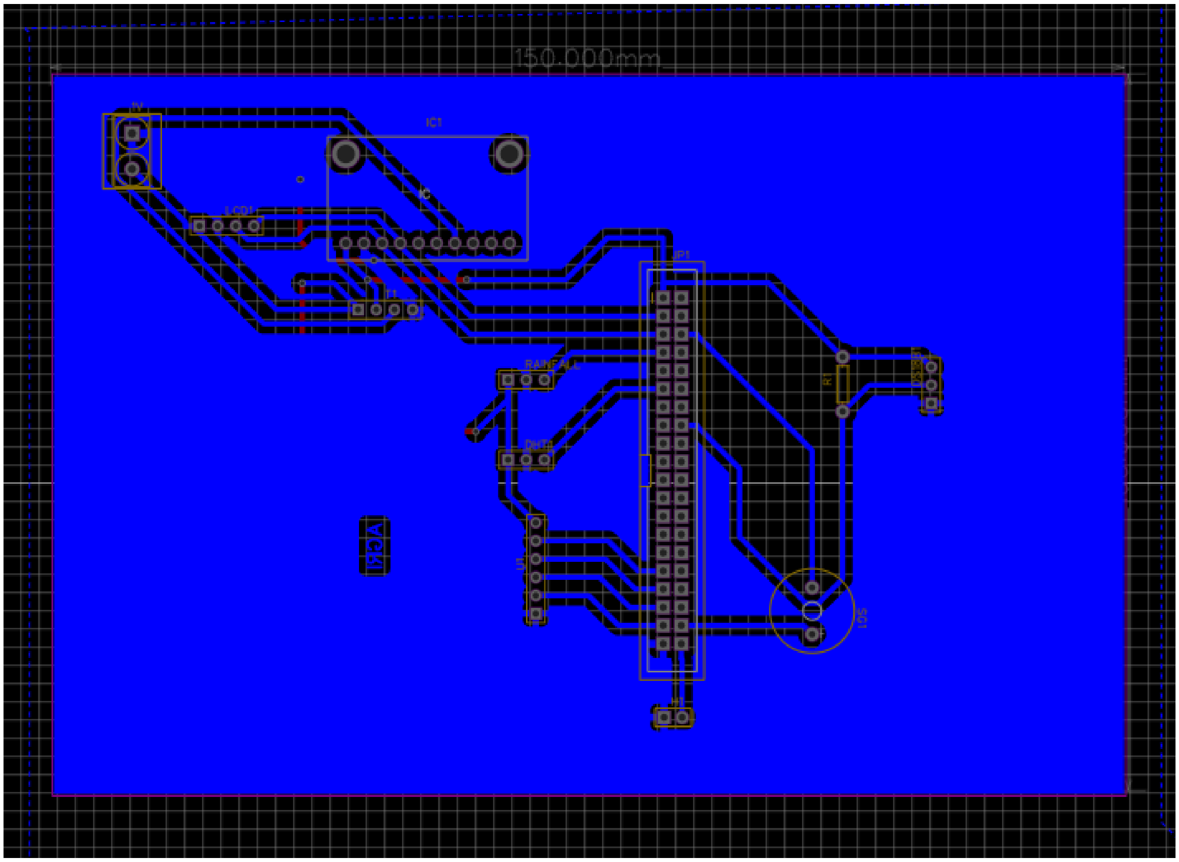

Fig 7.2.2 PCB

Software Developed:

The software for the **Automated Irrigation System using Raspberry Pi and IoT** manages sensor data, controls irrigation, and provides remote access. It uses Python to read soil moisture levels and activate a water pump via a relay when moisture falls below a set threshold. IoT integration with platforms like **Blynk** or **ThingSpeak** allows real-time monitoring and control through mobile or web interfaces. Weather API integration optimizes water usage by adjusting for rainfall forecasts. The software also provides alerts and notifications, enhancing user convenience and contributing to efficient, sustainable irrigation management.

7.3 Web Application developed:

Live data

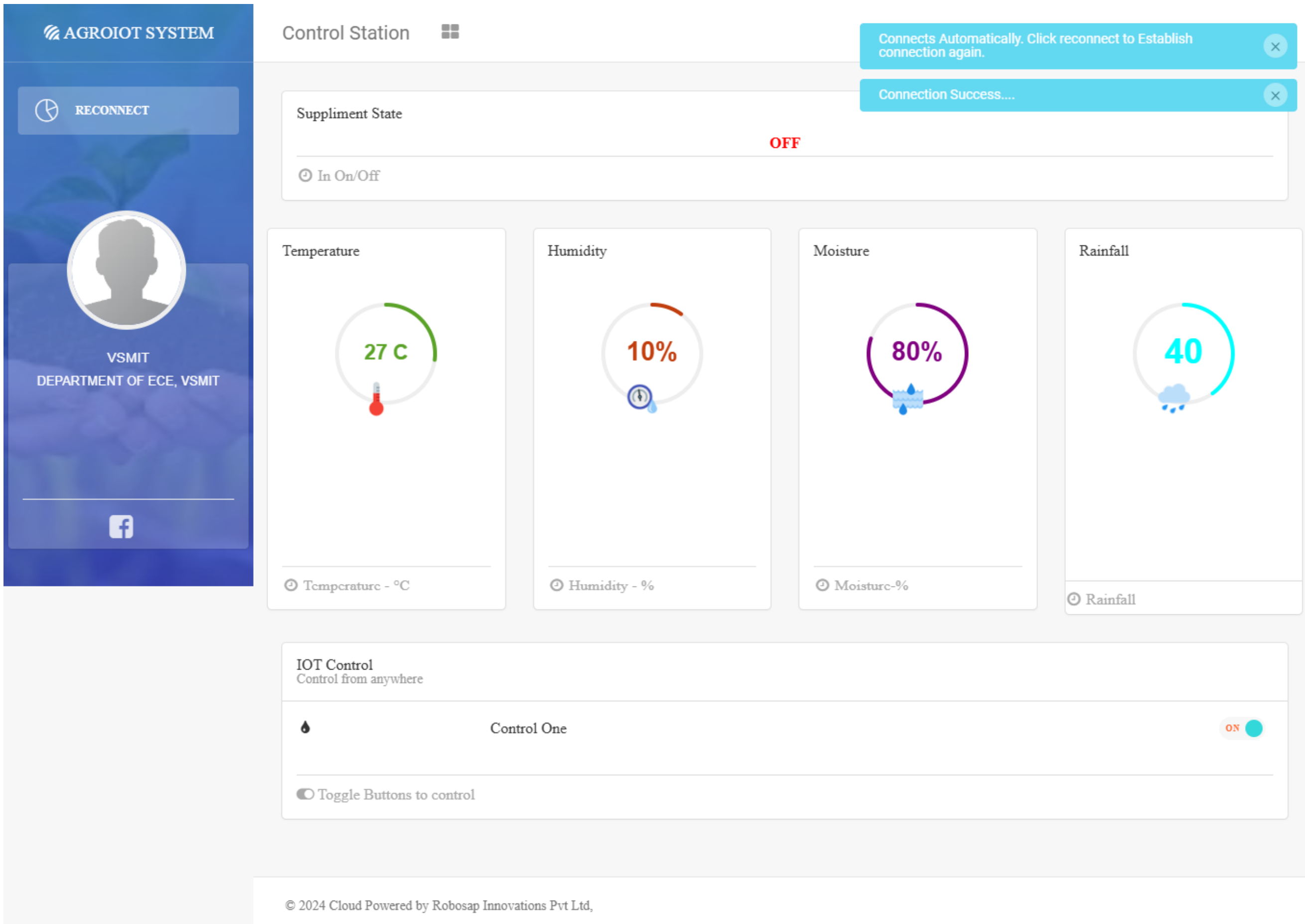

Fig 7.3.1 live Data

Control Option

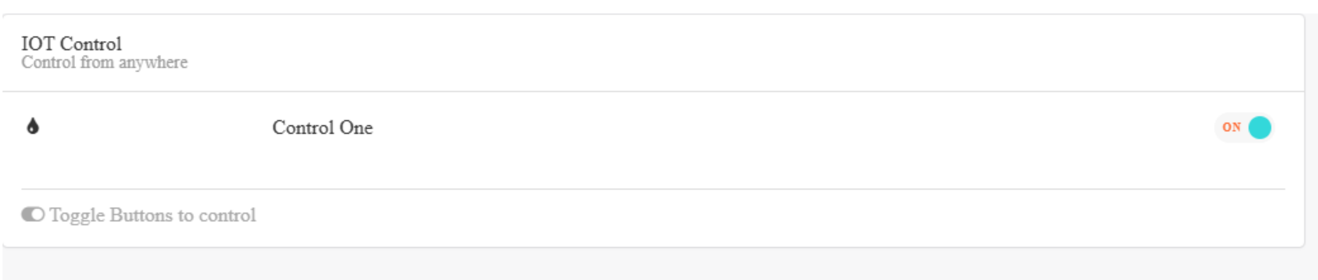

Fig 7.3.2 Control option

Feed Status Display:

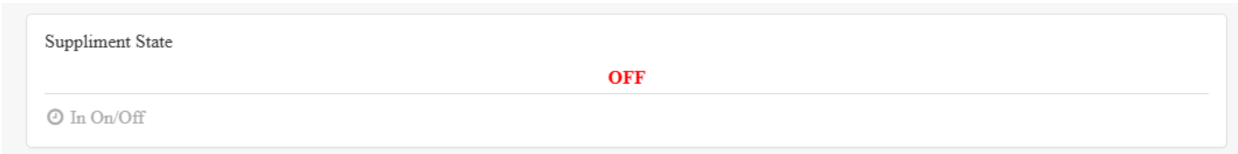

Fig 7.3.3 feed status

Live Sensor data:

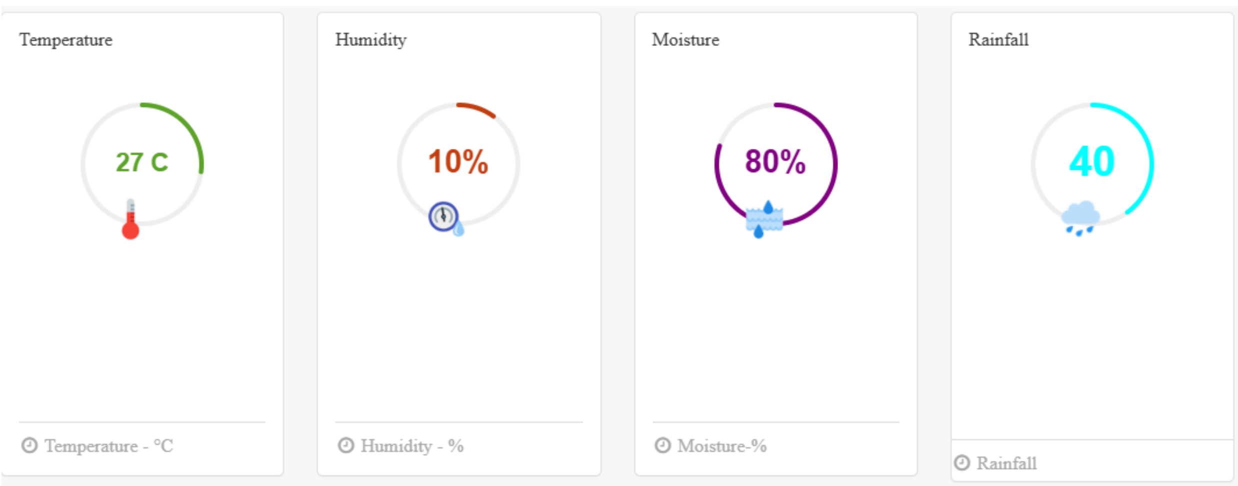

Fig 7.3.4 Live sensor data

7.4 Hardware Snap Shots:

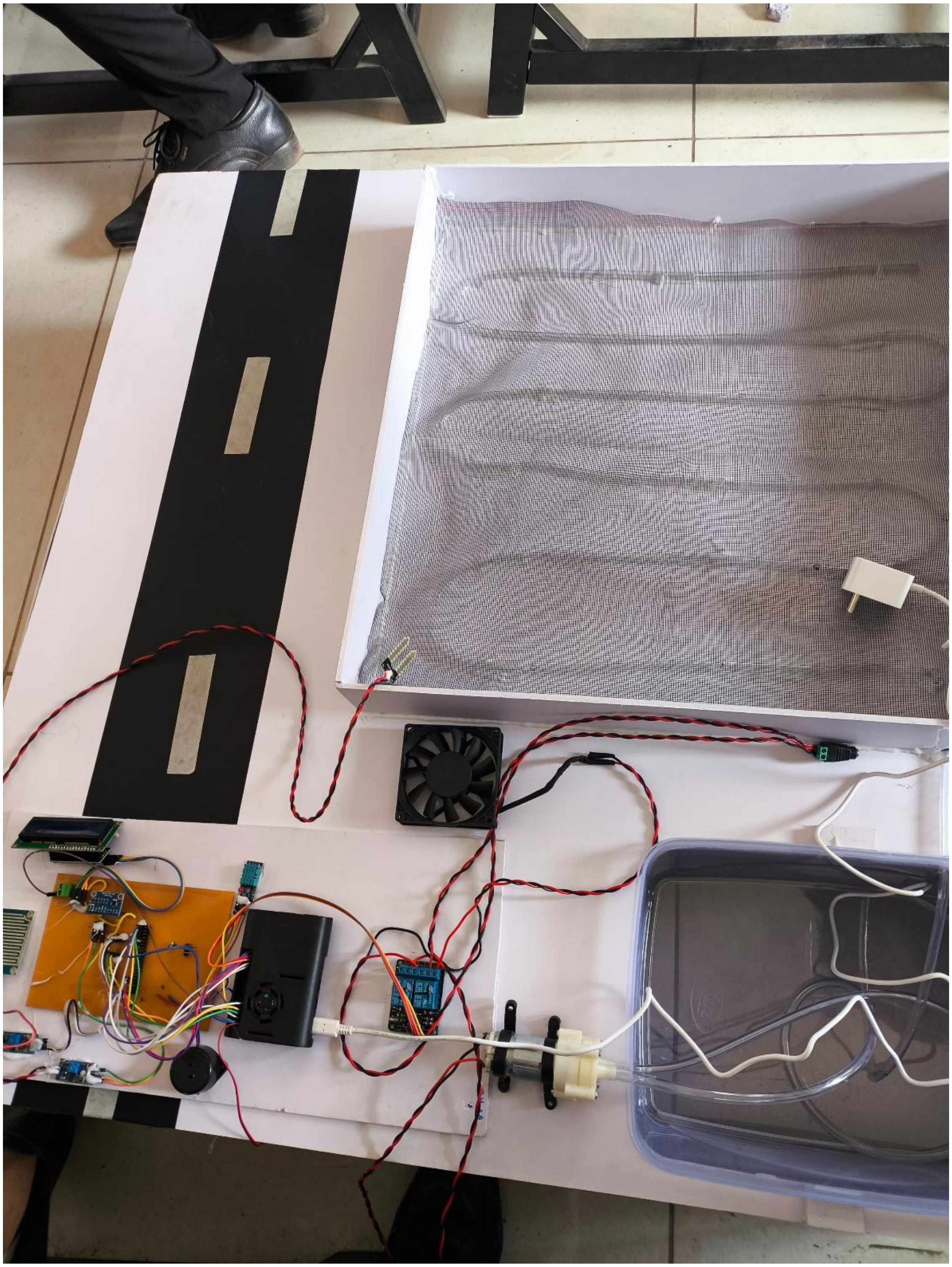

Fig 7.4.1 Hardware snap shot

Chapter 8

ADVANTAGES AND APPLICATIONS

8.1 Advantages

- IoT-based systems enable farmers to monitor and control field parameters remotely, reducing manual labour.
- Automated irrigation and feed distribution optimize resource use and enhance efficiency.
- Reuse of irrigation water reduces water wastage and operational costs.
- Automation minimizes human errors and ensures precise farming operations.
- Real-time data from sensors aids in better decision-making and crop management.
- Increased agricultural productivity due to optimized resource allocation and timely actions.
- Alerts and notifications improve field monitoring and mitigate risks effectively.
- The system is scalable and can integrate additional sensors or automation components.
- It supports sustainable agriculture practices by minimizing resource wastage.
- Improved quality of life for farmers with reduced workload and higher yield potential.

8.2 Applications

- Precision agriculture for better crop management and resource allocation.
- Smart greenhouses for automated environmental control.
- Irrigation systems for efficient water management in farms.
- Livestock monitoring with feed and water automation.
- Integration with weather forecasting for adaptive farming strategies.

- Real-time field monitoring for large-scale agricultural operations.
- Automated pest and disease management using sensor networks.
- Early warning systems for environmental or weather-related risks.
- Data-driven farming insights for yield prediction and market planning.
- Automated irrigation for large-scale farms to optimize water usage and increase crop yield.
- Smart irrigation for home gardens or lawns to maintain greenery with minimal manual intervention.
- Water management in urban green spaces and public parks.
- Automated irrigation in industrial gardens and eco-parks.
- Efficient watering of flowers and ornamental plants in nurseries and landscape gardening.

Chapter 9

CONCLUSION AND FUTURE WORK

The proposed IoT-based automated agricultural system significantly enhances farming practices by introducing remote monitoring, automation, and efficient resource management. With features like automated irrigation, feed distribution, and threshold-based controls, the system reduces labor, minimizes resource wastage, and improves crop yield. The integration of IoT technology ensures real-time data access, enabling farmers to make informed decisions and optimize their operations. This project demonstrates how modern technologies can transform traditional agriculture into a more sustainable and productive practice, improving the overall quality of life for farmers.

The implemented project has wide scope for further modification. The proposed project can be implemented with robotics to helps farmer carry out physical tasks in the field as well using internet of things. The future scope of the IoT-based automated agricultural system is vast, with potential advancements in integrating artificial intelligence for predictive analytics in crop health, yield optimization, and disease detection. Robotics can be incorporated for tasks like planting, weeding, and harvesting, while drones equipped with sensors can provide aerial monitoring of large farms. The system can also be enhanced with renewable energy sources like solar power for sustainability and blockchain technology for traceability and transparency in the supply chain. Weather-adaptive systems and more diverse sensor networks can further improve precision farming. Additionally, advanced mobile application features, such as multilingual support, voice commands, and real-time AI recommendations, can make the system more user-friendly and scalable globally, adapting to diverse geographical and climatic conditions.

Chapter 10

REFERENCES

1. G. Sandhi et al. “Visually Guided Operation in Green-House”, Intelligent Robots and Systems '90. 'Towards a New Frontier of Applications', Proceedings. IROS '90. IEEE International Workshop on, July 1990.

2. K. Rangan et al. “An Embedded Systems Approach to Monitor Green-House”, Recent Advances in Space Technology Services and Climate Change (RSTSCC), 2010.

3. Wei Ai et al., “Green House Environment Monitor Technology Implementation Based on Android Mobile Platform”, Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on, August 2011.

4. C. Akshay et al., “Wireless Sensing and Control for Precision Green House Management”, Sensing Technology (ICST), 2012 Sixth International Conference, December 2010.

5. Aji Hanggoro et al., “Green House Controlling and Monitoring Using Android Mobile Application”, QiR (Quality in Research), 2013 International Conference, June 2013.

6. Thenmozhiet al., “Green House Management Using Embedded System and ZigBee Technology”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, February 2014.

7. Gayatri et al., “Providing Smart Agricultural Solutions to Farmers for better yielding using IOT”, Technological Innovation in ICT for Agriculture and Rural Development (TIAR), 2015 IEEE, July 2015.

8. P. S. Asolkaret al., “An Effective Method of Controlling the Greenhouse and Crop Monitoring Using GSM”, Computing Communication Control and Automation (ICCUBEA), 2015 International Conference, February 2015.

9. Vishwanath Naik et al. “IOT Based Green-House Monitoring System”, International Journal of Electronics and Communication Engineering & Technology (IJECET), June 2015.