Development of IoT based Real Time system for Irrigation and Plantation

Mrs.Selina Khoirom[1], Mr.Vimal Mayengbam[2]

[1]Assistant Professor, Dept. of Computer Science & Engineering, Manipur Institute of Technology, Manipur-795004, India
[2]M.Tech Student, Dept. of Computer Science & Engineering, Manipur Institute of Technology, Manipur-795004, India

---***---

Abstract- *Better plant growth requires a pre-requisite level. The plant growth requires water level in soil in a timely manner. There is scarcity of water for plant irrigation. Undue usage of water must be avoided since water is the most essential part for sustainable of life. Irrigation of plants at regular interval of time is required for efficient use of water without wastage and better plant growth. Plant fields must not be over irrigated and at the same time there should not be shortage of water. For this task to overcome we require manual system or automated system. Since manual system requires much labour, automatic system is mostly preferred. This will reflect in less labour with high plant growth by automation system. Most of the existing automation system is based on the micro processor, micro controller system. These systems may have mush advance system but are more unaffordable and difficult for maintenance. These systems introduced an advance system of irrigation with simple and easy to implement. The designed system can be used by any unskilled labour. The system is developed in a way that when water is required, the automated system will irrigate the field base on the water content of the soil. It is controlled in a way that there will be maximum plant growth with appropriate water supply through automated system*

***Key Words*: Micro Controller system, Micro Processor.**

1. INTRODUCTION

In the current scenario farmers are using different types of irrigation techniques throughout India. Since the time immemorial manual system of irrigation system is used to irrigate the land from time to time. By analyzing this process, it consumes large amount of water and the yield is reduce to some extent. For the efficient use of water automation system of irrigation is introduced to use water efficiently than the manual system of irrigation by measuring the amount of water present on the soil. Plant irrigation is very time consuming process and large resource is required for this type of irrigation activities. Large human resource means large cost factor. All these type of activity is the traditional process.

For the above situation an effective solution has been introduced which is termed as "Development of IoT Based Real Time Irrigation for Agriculture". In this system an automatic real time control is engaged for water pump motor to turn off and turn on. By using this system the farmers can easily implement the real time irrigation system. This helps to use less water in as per the measurement of the moisture content in the soil which in turn helps the farmer to produce more crops than before using the traditional system.

In this paper the efficient use of water is based on testing the moisture content of the soil of crop fields so as to maintain the correct amount of water required in the field. For the designing of this system Raspberry Pi is used to control the whole system. At the initial of developing the project, there may be some hectic. But after completing the project and at the implementation phase it is much easier to implement by any farmers. This project is real time in nature and it facilitates the farmers to irrigate the fields with less amount of human resource.

This project is embedded system where there is a control system of water pump motor and other device automatically. The water content is monitored throughout the day by using the built in system and other output peripherals. There is no need of monitoring the moisture content of the soil since it is monitored automatically. The data of the moisture content can be send to the centralized system for analyzing the moisture content. This can be done local WIFI connection or by using internet. For the local connection a server is required but for the internet, we can store the data to any web server.

In this project, Raspberry Pi is used for controlling the entire system. The moisture content of the soil is displayed on the screen. The moisture is measured by the soil sensor device. This device may be single or multiple as per the requirement of the user. The moisture content are displayed to the monitor according to the time interval set in the Pi system. We are going to set a certain threshold value for the moisture content. Raspberry Pi will received data in the form of code form the controller. After whole process all the data are displayed on the screen of LCD. Once the threshold value is reached to a certain level, the information will be send to controller and the relay switch will make on which in turn make the water pump automatically on. If the soil is dry, the condition is very harmful to plants. So immediately the soil is make wet from the pump.

1.1 Objective

The main objective of this project is developing a real time irrigation system which solves many of the problems for irrigation to the agricultural crops which in turn control the water for efficient use and saving water and electricity. It also emphasized on increasing agricultural productivity with less amount of water, minimizing the human intervention in watering the plants which consume much of the time. It increases the speed watering the plants. All these mentioned features will make an automated system which is sustainable option and can be considered for improving agricultural irrigation in efficient way. The main aim of this project is discovering the efficient automation system for irrigation system that can be done automatically which is controlled by using software in such a way that it allows the farmer to monitor all threshold value of moisture information and to manage the controller automatically. Many researchers have discussed the smart irrigation system and many proposed solutions are also presented [7] [8]. Some solution are also considered.

The aims and objectives to consider are:

- Simplifying the irrigation system by designing and installing irrigation system as a whole.
- Saving energy, that allows smart irrigation system using the application.
- Water consumption is optimized.
- Fully automated system.
- Cost of operation is decreased.
- The system is made easy for farmer.

2. LITERATURE REVIEW:

In paper [1] Anushak, Dr. U B Mahadevaswamy," (2018, 6, 55-67) ,presented a report of using IoT technology , is discussed by the author to collect the information. There is large improvement in agricultural technology with the help of digital technology in the recent decades.

In paper [2] S.N. Ishak, N.N.N.Abd Malik, N.M. Abdul Latiff, N. EffiyanaGhazali, M.A. Baharudin, (2017, IEEE) presented a report, that by using smart irrigation system the human intervention is reduced to a large extent and is discussed by the author. It saves much of the time and consumption of water.

In paper[3]Tanmay Baranwal, Nitika, Pushpendrakumar Pateriya, (2016 IEEE) a prototype is developed by the author for securing crop and grains from various insects. In this paper web camera is installed in the field to collect data on daily basis for information. In this type of method Raspberry pi processor is used and python language is used for the implementation.

In paper[4] Shruti A Jaishetty, Rekhapatil(2016) presented a report , the problems faced by the farmer are discussed. By analysing the problems the author proposed a new system based on Iot. In this technology, sensors and MQTT network are used.

In paper[5] Keerthi.V. V. V, Dr.G.N.Kodandaramaiah,(2015) presented a report that different types of sensors are used to monitor the conditions of environment in the agricultural fields by the author. In this paper IoT concept is applied. The device include GSM module and micro controller are used in this project. All the activities are displayed and observed on the LCD screen and are present on graph.

3. OVERVIEW

Since agriculture is very much important, this paper will mainly focused on building a system based on "IoT" which allows automation of irrigation process which is fully controlled by the controller. This paper is to aim to control the system with a software and hardware application to find the efficient way of irrigation system with less water where water is scare much now in the field of agriculture. This may be considered as a big challenge for irrigation technique by implementing hardware with software application. The concept of efficient use of irrigation system can be applied to various system. This paper will give the automation system of irrigation by solving the traditional way of irrigation. It will target to use less water at appropriate timing for best yield of crops.

3.1 Overview of the IoT based system for Irrigation and Plantation

3.1.1 System Requirements:

Functional requirements

This project works through different stages the stages are as follows:

1. If the soil moister sensor is wet or is in the certain threshold value set by the user the system will not work since there is enough water in the agricultural field otherwise the system will check the value of the moisture content in the field.

2. If the moisture content of the field is less than the given threshold value then the signal will be sent to the

controller and displayed on the LCD screen. The system will automatically run the pump to irrigate.

3. If the water level is raised and reached the given threshold value the data will be send to the controller and displayed on the LCD screen.

4. When the water is reached to the given value the motor will stop automatically.

3.1.2 Non-functional requirements

Easy implementation

The project materials required is much easy to install for implementing a successful project. Moreover materials that are used in this project is easy to connect each other for building the project and hence more effective. In case if there is any default in the materials that are used, then it can be replace.

Open source

The components and controller used in this project is easily available and open source. The hardware used are priced reasonably and the development software are free.

Strength

The strength which is termed as durability of the components used are high. It can be used for a very long period of time which means that cost factor of the components is less.

Quality

The tools that are required for building this project should be of excellent in quality for operation having a longer period of time. Selecting a quality material is very much important for avoiding wastage of money and for avoidance of the technical problems of the device used and process disturbance. For achieving a successful project an excellent quality of product is required.

Modifiability

The material are chosen based on the ease of the modifiability, since it is common for connections and designs. Anything can be made easy to modify or replace in the near future.

Communication

The system can be connected via a data cable or WiFi as per the convenient of the user.

Accuracy

The data reading of all records are stored in the sd card or to a server for analysis and research.

Performance

The system is working on the real-time basis.

Operational

The system cost of this project must be not much expensive since we are aiming to minimize the number of worker which means reducing the amount of the money in the process of irrigation and solving the water consumption.

3.1.3 Advantages

- Moisture within the root zone can be maintained at field capacity.
- Uniform water distribution.
- Labour cost is made less comparing to other irrigation methods.
- The waste of fertilizers can be made minimal.
- Water and electricity consumption is less to a significant amount.

3.1.4 Existing and Proposed system

Comparison:

Particulars	Existing	Proposed	Remark
Sensor	Moisture, humidity, temperature, Wireless sensor Network , light sensor, ultrasonic sensor	Soil Sensor	Use of less sensor
Water Pump	Water Pump motor	Water Pump Motor	
Controller	Microcontroller	Raspberry Pi	More advance technology with storage capacity
Display	Monitor required	LCD display	Desktop not required for parameter reading
Cost	More in equipment	Less	Less cost
User friendly	Moderate	Easy	Easy to use

3.1.5 Proposed Method:

The main objective of this project is developing a real time irrigation system which solves many of the problems for irrigation to the agricultural crops which in turn control the water for efficient use and saving water and electricity. It also emphasized on increasing agricultural productivity with less amount of water, minimizing the human intervention in watering the plants which consume much of the time. It increases the speed watering the plants. All these mentioned features will make an automated system which is sustainable option and can be considered for improving agricultural irrigation in efficient way. The main aim of this project is discovering the efficient automation system for irrigation system that can be done automatically which is controlled by using software in such a way that it allows the farmer to monitor all threshold value of moisture information and to manage the controller automatically. Many researchers have discussed the smart irrigation system and many proposed solutions are also presented. Some solution are also considered.

The aims and objectives to consider are:

- Simplifying the irrigation system by designing and installing irrigation system as a whole.
- Saving energy, that allows smart irrigation system using the application.
- Water consumption is optimized.
- Fully automated system.
- Cost of operation is decreased.
- The system is made easy for farmer.
- For efficient use of water this project has been initiated. When water is required by the plant it is irrigate lonely at that time. There is already existed irrigation system which depends on pH value of soil, temperature, light and soil humidity. The productivity of the crops depends on these parameters in the agricultural fields.
- Efficient irrigation system will be maintained in office premises, house gardens, buildings and agricultural lands whenever water is required at regular intervals. We can also implement water drip system by using micro controllers which is again controlled by Raspberry Pi.
- Water for agricultural lands are dependent on rain water and bore wells for irrigation.
- Even if water-pump is used in the farm land, farmer's manual intervention is always required for pumping water on and off when water is needed.

3.2 System stakeholders, users, and client:

This project is developed mainly for solving different types of problems which is related to irrigation process. This system mainly targets to agriculture for irrigation in the land. Different field has different problems. So in order to solve the problems we need to study the different conditions and their requirements. The adaptation has to be made with this project to solve the said problem.

This project can be implemented to public and private sector for water supply system. In water supply system instead of soil sensor we can directly use the water level sensor on the large scale industry. If we know the timing for fertilization process, we can implement this system in an automatic manner.

3.2.1 Hardware Components :

Hardware components used for Development of IoT based Real Time Irrigation for Agriculture are:

Sl No	Hardware Components	Cost (INR)
1	Soil Sensor	90
2	MCP3008 (Analog to digital convertor)	325
3	Water Pump motor	100
4	LCD display (16 x 2)	176
5	Relay Switch	110
		Total =801

Table 3.2.1: Hardware Components

3.2.2 Soil Sensor :

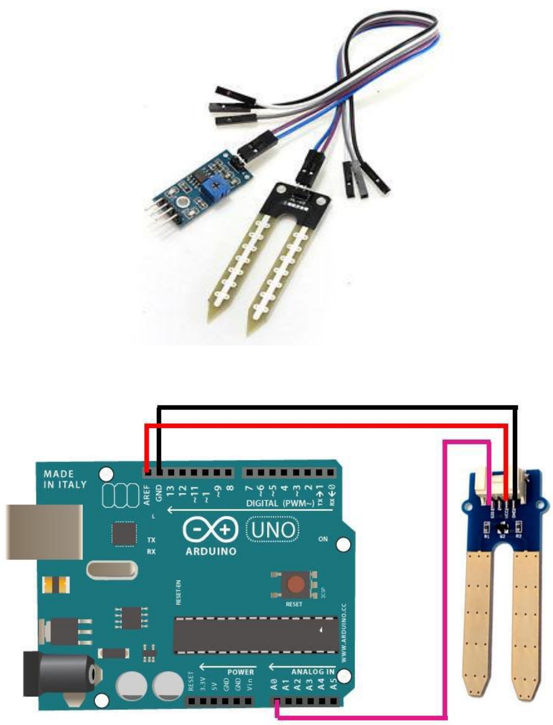

Fig 3.2.2: Soil Sensor

In above figure, Soil moisture sensor is shown, it measures the content of water humidity in the soil. Since direct hydrometric is to measure wetness of free soil which needed drying, removing and sample coefficient, soil sensors wetness depends on the content of water indirectly by victimization another soil property, like constant of nonconductor, phenomenon of electric, or neutrons interactions, as a wetness content proxy. The sensor of soil moisture is comprised of tests which are based to the volumetric degree of water substance. The test permit the water current pass through soil where the resistance plays the major role in conductivity. When more water is present in the soil, less resistance will be there and more power is developed in the soil. But if there is dry soil it will conduct power in week manner and less power is needed and resistance is required more.

3.2.3 MCP3008

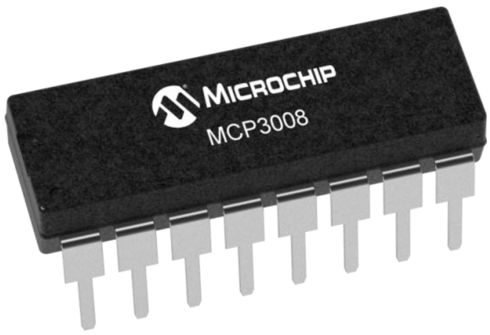

Fig 3.2.3: MCP3008 analog to digital convertor

The MCP3008 is 10 bit analog to digital convertor and its cost is low. It is easily available. This device can read analog signals from a controller and converted into digital signal. This chip is capable of reading analog signals from temperature, light sensor or soil sensor.

The datasheet of MCP3008 plays an important role of resource to have handy and skim.

3.2.4 Wiring:

The MCP3008 is connected to Raspberry Pi by using a SPI connection which is serial connection. We can connect by hardware SPI connection bus, or by using any of the four GPIO pins where software SPI is used to communicate to MCP3008. SPI Software is flexible where it is feasible to connect with any available pins on the Raspberry Pi, and SPI hardware is little bit faster but less flexible since it works on specific pins.

3.2.5 Software SPI :

To connect with Raspberry Pi by MCP3008 we need some pins to be connected with SPI software. It is listed as below

i. VDD to Pi 3.3V
ii. VREF to Pi 3.3V
iii. AGND to Pi GND
iv. DGND to Pi GND
v. CLK to Pi pin 18
vi. DOUT to Pi pin 23
vii. DIN to Pi pin 24
viii. CS/SHDN to Pi pin 25

It is to be noted that swapping of DIN, DOUT, CS/SHDN and MCP3008 CLK can be done to any of the GPIO free pins on the Raspberry Pi system. So that there is room for changing the pin connections.

3.2.6 Hardware SPI:

In order to use hardware SPI we need to set raspi-config tool i.e. enabling the SPI. This option is found in the raspi-config file in Raspberry Pi system. We need to enable both loading SPI kernel and SPI interface. After setting the configuration we need to reboot the Pi.

We can wire MCP3008 and the Raspberry Pi is as follows:

i. VDD to Pi 3.3V
ii. VREF to Pi 3.3V
iii. AGND to Pi GND
iv. DGND to Pi GND
v. CLK to Pi SCLK
vi. DOUT to Pi MISO
vii. DIN to Pi MOSI
viii. CS/SHDN to Pi CE0

3.2.7 Library Install:

The connection has been made between the MCP3008 and the Raspberry Pi. The connection can be made either by the software SPI or hardware SPI, we should install the library of Adafruit MCP3008 Python. This library can be installed from the Python package by using commands with internet connection or we can install the library GitHub source.

3.2.8 Raspberry Pi :

Fig 3.2.8: Raspberry Pi

This project is implemented by using Raspberry Pi 4 Model B. In June 2019 this Model B was released having an ARM Cortex-A72 processor, 1.5 GHz, 64-bit quad core, full Ethernet gigabit (throughput not limited), on-board 802.11ac Wi-Fi, Bluetooth 5, dual-monitor support, two USB 3.0 ports, two USB 2.0 ports, and connection by HDMI Type D ports capable up to resolution of 4K. The Raspberry Pi 4 has a USB-C port for power connection, this enables power to be additional for providing downstream peripherals power, when it is used by PSU appropriately. The board of previous version of Raspberry Pi 4 has some limitation on design flaw, where e-marked USB cables used on Apple Mac Books are identified incorrectly and refused for providing power.

3.2.9 Water pump :

Fig 3.2.9: Water pump

The submmersible water pump is used in the project where thw water is needed for irrigation from the main water tank through pipes. For different typess of applications, this water pump is used in many household including space heating cleaning, flowering and bathing. This type of pump is choosen for this project since it has many major advantages. Some asdvantages is that it is light weight, it is small in size, easy to install and can easily replaced. Moreover it has higher efficiency for pumping for irrigation. It consume only 12Volt. Since it consume low voltage, it requires low power supply which means less cost in the project implementation. It is readily available in the market. Furthermore, it produce low noise.

3.2.10 LCD display :

Fig 3.2.10: LCD display

This project uses 6 x 2 characters display i.e. six columns by 2 rows. Two different ways are there to connect Raspberry Pi to the LCD. One is 4 bit mode and other is 8 bit mode. 6 GPIO pins use 4 bit mode, 10 GPIO pins use 8 bit mode. 4 bit mode is commonly used method in developing the project since less pin are used.

The command and character are sent as byte (8 bits) to LCD screen as data format. All the bytes are sent at once through 8 data wire in the 8 bit mode i.e. one bit is sent per wire. But in 4 bit mode, two sets of 4 bits are split from the bytes. The data are sent in 4 data wires and it has two types of bits, the upper bits and the lower bits.

In theory 8 bit mode is the fastest means of data transfer since it send all the data at once. The data transfer rate is twice the 4 bit mode. In practical, the data took a relatively long time to process by the LCD driver no matter which mode of transfer is used. We cannot notify the difference between 8 bit and 4 bit mode.

3.3 Software Platforms:

In Raspberry Pi, Raspbian operating system is the official OS and used commonly. Most of the user are also want to use it because of its flexibility and user friendly. Raspbian operating system is the Linux version but especially for Raspberry Pi. It has all necessary software with it for every task that is basic for a computer.

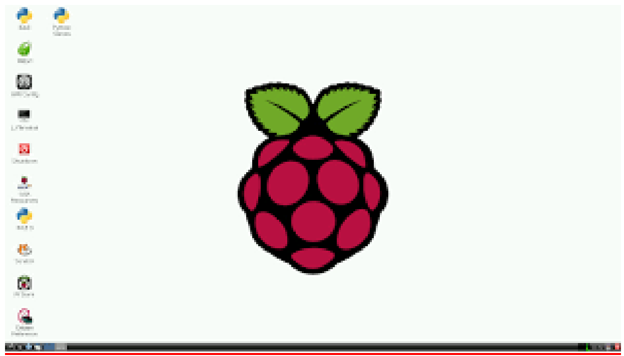

Fig 3.3: Raspberry Pi OS

The Raspberry Pi is a sized of credit-card, low cost computer which can be connected to monitor or smart TV. We also need to connect with keyboard and mouse. In short it is a small portable computer that can also be used with small LCD monitor. This enables many user to use computer and exploring the world of computer and any user can learn programming language like python from scratch.

3.3.1 Installing standard software:

Different ways are there for installing the software. It can be installed from online server to our Raspberry Pi which is dependent on the source where it is available.

APT:

APT (Advance Packing Tool) command line is the common tool for installing the software. There are also some other tool for installation which have different package manners. The APT is used for installing the software from the OS repository of Raspberry Pi. A large number of software packages are available in this repository. We can also search the required software by using the search term from the database for different software packages.

Python:

In Raspberry Pi pip command is used to install python IDE in the OS. Any other software can also be installed form the source using command.

Ruby:

Ruby gems package manager is used to install Ruby software in the Raspberry Pi OS.

3.4 Implementation of the proposed system:

- For efficient use of water this project has been initiated. When water is required by the plant it is irrigate lonely at that time. There is already existed irrigation system which depends on pH value of soil, temperature, and light and soil humidity. The productivity of the crops depends on these parameters in the agricultural fields.
- Efficient irrigation system will be maintained in office premises, house gardens, buildings and agricultural lands whenever water is required at regular intervals. We can also implement water drip system by using micro controllers which is again controlled by Raspberry Pi.
- Water for agricultural lands are dependent on rain water and bore wells for irrigation.
- Even if water-pump is used in the farm land, farmer's manual intervention is always required for pumping water on and off when water is needed.

3.4.1 Complete set up

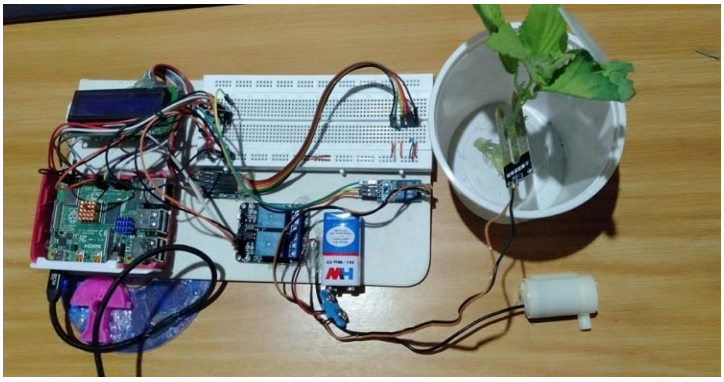

Fig 3.4.1: Development of IoT Based Real Time Irrigation for Agriculture.

3.4.2 Circuit Diagram:

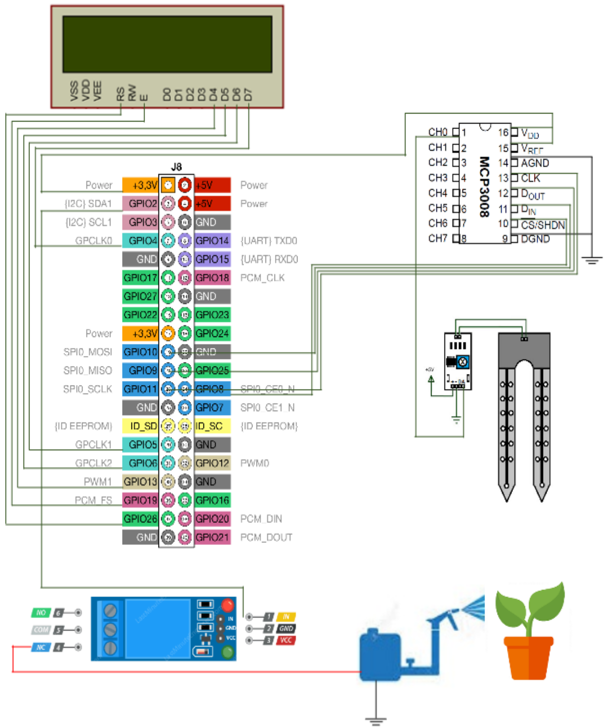

Fig 3.4.2: Circuit diagram for Development of IoT Based Real Time Irrigation for Agriculture.

3.4.3 Connection of Raspberry Pi with LCD Screen

LCD_RS = 26 (GPIO pin)

LCD_E = 19 (GPIO pin)

LCD_D4 = 13 (GPIO pin)

LCD_D5 = 6 (GPIO pin)

LCD_D6 = 5 (GPIO pin)

LCD_D7 = 4 (GPIO pin)

#LCD_D7 = 11 (GPIO pin)

LED_ON = 15 (GPIO pin)

3.4.4 Connection of Raspberry Pi with Relay Switch

Relay Switch = 2 (GPIO pin)

3.4.5 Connection of Raspberry Pi with MCP3008

V_{DD} = 1 (GPIO pin)

$V_{REF = 1}$ (GPIO pin)

AGND = Ground

CLK = 8 (GPIO pin)

D_{OUT} = 9 (GPIO pin)

D_{IN} = 10 (GPIO pin)

CS/SHON = 11 (GPIO pin)

DGND = Ground

3.4.6 Connection of Soil Sensor Pi with MCP3008 :

Analog output (Soil Sensor) = CH0 (Pin 1) of MCP3008

3.4.7 Block Diagram:

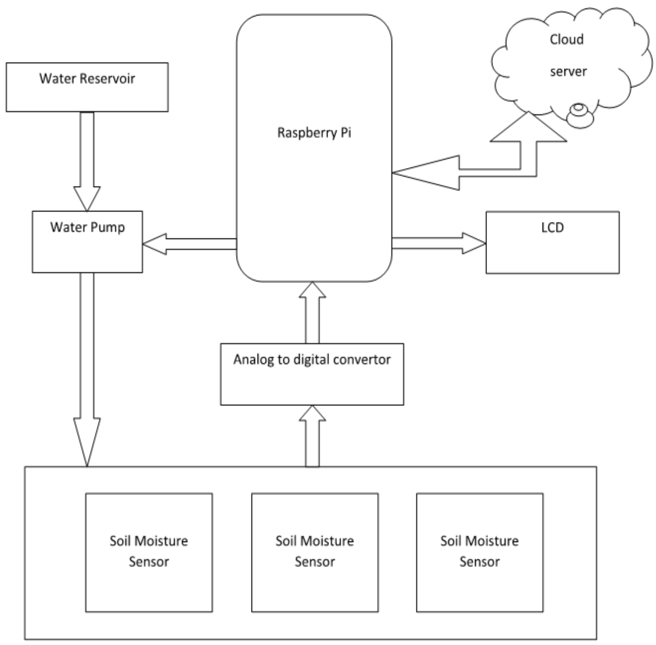

Fig 3.4.7: Block diagram of Development of IoT Based Real Time Irrigation for Agriculture

The block diagram of irrigation system is divided into different blocks like water reservoir water pump, raspberry pi LCD, analog to digital convertor and soil moisture sensor block.

In the above block diagram from the soil sensor block the moisture content will be sent to the analog to digital convertor. The signals from the soil sensor are sent as an analog signal to the convertor. The analog signals are then converted to digital signal. The digital signals are then accepted by Raspberry Pi which is again displayed on the LCD screen. The data can also be sent to the cloud for further reference by connecting to internet.

We are given a certain threshold value of 30% for soil moisture. This value is set according to the moisture content. When the value is below 30% the water will pump from the reservoir through water pump. The water will pump till it reaches the threshold value. When it has reached the value, the pump will automatically stop.

3.4.8 Algorithm:

Steps:

1. Start the program.
2. Print "Development of IoT Based Real Time Irrigation for Agriculture" on LCD.
3. Print Soil moisture content on the LCD and send data to cloud server.
4. Check if Moisture content < 30%.
5. If yes turn on the motor to pump water and go to step 3.
6. If no Print the new value of moisture content, send data to server and repeat step 2

3.4.9 Flowchart :

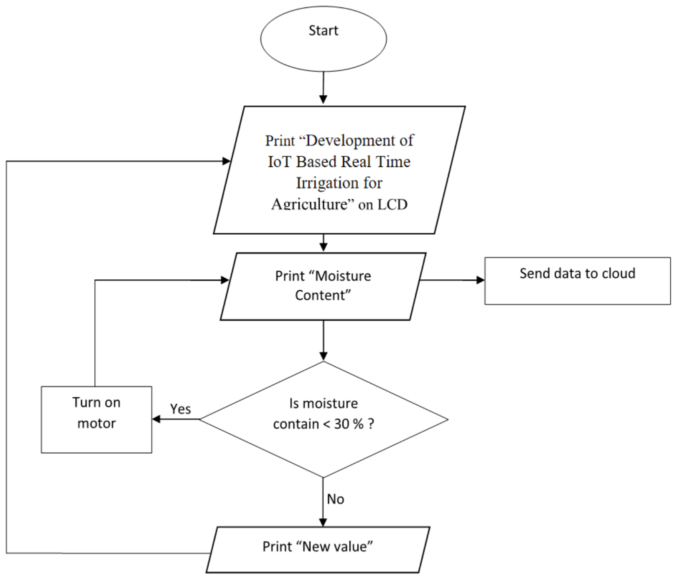

Fig 3.4.9: Flow chart of Development of IoT Based Real Time Irrigation for Agriculture.

3.4.10 Work Outs:

The workouts for Development of IoT Based Real Time Irrigation for Agriculture are shown in the below chart:

3.4.11 Gantt chart:

Task	Start Date	End Date	Duration
Gathering idea	01-Dec-19	28-Dec-19	27
H/W requirement	29-Dec-19	23-Jan-20	25
Raspberry Pi set up	24-Jan-20	30-Jan-20	6
Board assemble	31-Jan-20	11-Feb-20	11
H/w communications	12-Feb-20	21-Feb-20	9
Coding	22-Feb-20	29-Mar-20	36
Bugs Fixes	30-Mar-20	13-Apr-20	14
Improvements	14-Apr-20	27-Apr-20	13
Final testing	28-Apr-20	03-May-20	5
Release version	04-May-20	06-May-20	2

3.4.12 Gantt chart:

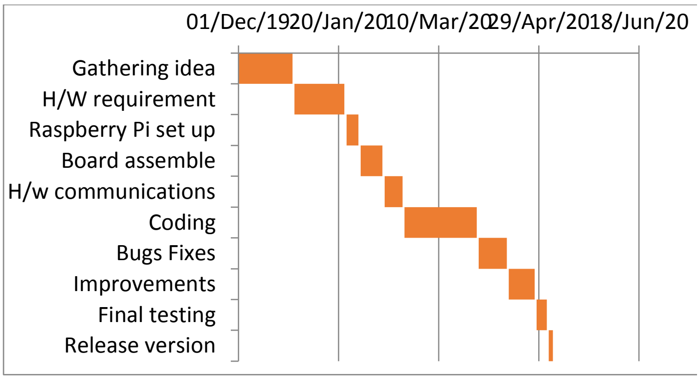

Fig 3.4.12: Gantt chart

4. Results and discussion:

4.1 Operation of Development of Iot Based Real Time Irrigation For Agriculture:

Operation of the project requires all the necessary components get the required power supply. The circuits that are designed required 5volt dc supply to work. MCP3008 require 3 volt to work on the circuit. The relay switch also requires 5volt input to operate. This input voltage can be given from the Raspberry Pi system. All the power supply can be given from the Pi since there is output voltage of both 5 volt and 3 volt from the system. For power supply of water pump, a separate power supply can also be used.

In this project all the device are operated from the Raspberry Pi. The soil sensor is connected to the MCP3008 where it converts analog signal from the sensor to digital signal. The digital signal is send to the Raspberry Pi. Then the signal will be shown on the screen of LCD. We are given a threshold value of 3% moisture content. If the threshold value is less than given value, a signal will be send to the relay switch to on the water pump. The water will pump until it reaches the given value. When it reach the value water pump will automatically stop. All the values of the moisture content are shown on the screen. This data can also be sent to the server if we connect to internet for further analysis

4.2 Data Analysis:

During the analysis of the manual irrigation, we irrigate the plants on every day basis in the morning without knowing the content of moisture and humidity of the day. It has been observed that whether the water content is high or low we irrigate the water which makes the water to use in useless manner where the water is very scare.

After implementing the real time irrigation system, we don't need to irrigate water each and every day. The moisture content is automatically detected by soil sensor. When we set the threshold value to a certain value the water will irrigate based on the given value. Hence it is observed that there is no need for manual irrigation.

4.3 Data Table:

Days	Threshold value
1	90
2	50
3	40
4	30
5	100
6	88
7	65
8	49
9	28
10	27
11	104

12	94
13	82
14	74
15	60
16	52
17	44
18	39
19	32
20	29
21	102

Table 4.3: Threshold Data

The development of IoT based real time irrigation for agriculture is plotted in graph below.

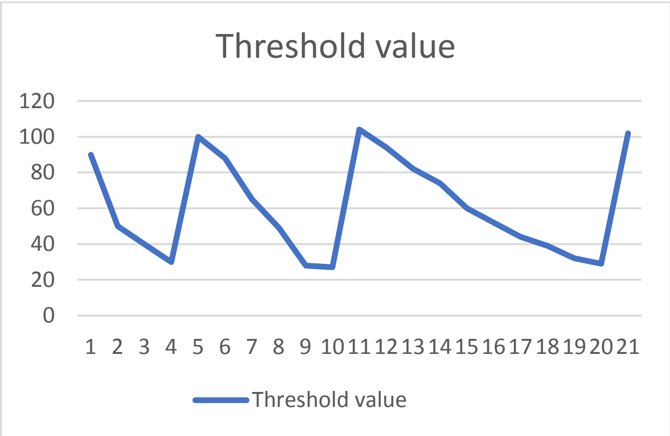

Fig 4.3: Graph of water content in soil

Here in the above graph, water content is shown in graph. It is based on the average temperature of 27^0 C in the month of May. The water content is observed from the flower pot which is kept inside the room. On the first day water is pumped and the threshold vale is 100 +. On the following days it decreases by a certain level. On the day 4^{th}, the value is decreased at the level of 30. When it reach 30 the water will pump again to irrigate the plant. It is observed that inside the room water content is decreased slowly up to the level of 30 and it takes nearly 3 to 4 days depending on the temperature and humidity.

After the operation has been performed it has been observed that a farmer need not to be on agricultural fields for irrigation, manual intervention on field is not necessary since the project is based on real time irrigation. All the modules of the project are placed with a reason and carefully, thereby contributing best outcome of the project for each unit in the project. The project has been tested automatically. The moisture level of the plant is measured by the soil moisture sensor. If the moisture level goes below the given level of moisture, a signal will be sent to Pi board. The board triggers the relay switch to operate the pump to turn ON to supply the required water to agricultural fields. When the moisture level is reached up to the given value, the relay switch stops the water pump. The entire system of the project has been tested and it is functioned successfully.

5. Conclusion:

5.1 Challenges:

From the initial stage many challenges were faced. All the materials that are used in developing the project are not easily available in the market easily any any instant of time. We need to purchase from online store. It take more time to reach home and hence delayed the practical session of the project. At a very low percent the materials arrived has defect so we need to replace. The next challenged we faced is the wire connection. The wire connection has more than 30 pin connections. If there is a single mistake in connection of the wire, all the devices may fused easily. So care must be taken in connecting the wire. Furthermore, the battery used in the project is drained fast. So external source of power supply is required for smooth running of the project. The controller of the Raspberry Pi were very challenging a single mistake can damage any electrical part. Since more than 5 devices are connected to Pi, it is very difficult to connect.

5.2 Conclusion:

The project has been decoratively designed and is tested successfully. It is developed by using various hardware components. This project is intended to use for household purpose but it can be implemented on the large scale basis. So that it can be used on agricultural fields. The plant irrigation can be used from various source of water either it can be used from pond or a fixed tank or by flipping the water source. This project is very much flexible and user friendly to use. Since it is automated and real time in nature, the user or farmer need not necessary to stay at home and water the plants.

5.3 Future work:

Having limited time bound some features may not be added. In the near future many features can be implemented. Features like humidity sensor for calculating the threshold value and control the whole process from the mobile app may be added. Controller can also be added for video capturing and at the same time sending it to the farmer.

5.4 Financial Cost of the Project:

Sl No	Hardware Components	Cost (INR)
1	Soil Sensor	90
2	MCP3008 (Analog to digital convertor)	325
3	Water Pump motor	100
4	LCD display (16 x 2)	176
5	Relay Switch	110
		Total =801

Table: Financial Cost

6. References:

[1] Anushak , Dr. U B Mahadevaswamy ,"Automatic IoT Based Plant Monitoring and Watering System using Raspberry Pi", *I.J. Engineering and Manufacturing,* 2018, 6, 55-67 .

[2] S.N. Ishak, N.N.N.Abd Malik, N.M. Abdul Latiff, N. EffiyanaGhazali, M. A. Baharudin' *"Smart Home Garden Irrigation System Using Raspberry Pi",* 978-1-5386-3132-4/17/$31.00 ©2017 IEEE.

[3] TanmayBaranwal, Nitika, PushpendrakumarPateriya, Department of Computer Science, Lovely professional university, phagwara, Punjab."Development of IoT based Smart Security and Monitoring Devices for Agriculture".978-1-4673-8203-8/16/$31.00 ©2016 IEEE.

[4] Shruti A Jaishetty, Rekhapatil M.Tech department of computer science and engineering, poojyadoddappaappa college of engg, kalaburgi ,Karnataka "IOT sensor approach network based approach for agricultural field monitoring and control",Journel of Research in Engineering and technology, vol-05 issue 06 June 2016.

[5] Keerthi.V. V. V, Dr.G.N.Kodandaramaiah, Department of embedded systems, kuppam engineering college, JNT University, AP, Department of ECE ,kuppam engineering college, JNT University, AP. "cloud IoT Based Greenhouse monitoring system" Journal of Engineering Research and Applications, vol.5, issue 10,(part-3) October 2015.

7. Appendix :

Codes :

```
importbusio

importdigitalio

import board

import adafruit_mcp3xxx.mcp3008 as MCP

from adafruit_mcp3xxx.analog_in import AnalogIn from time import sleep

# create the spi bus

spi = busio.SPI(clock=board.SCK, MISO=board.MISO, MOSI=board.MOSI)

# create the cs (chip select)

cs = digitalio.DigitalInOut(board.D5)

sleepTime = 1

# create the mcp object

mcp = MCP.MCP3008(spi, cs)

# create an analog input channel on pin 0

chan = AnalogIn(mcp, MCP.P0)

sleepTime = 1

print('Moisture Percentage = ', (63702 - chan.value)/350)

sleep(sleepTime)

#______________________________________________________

#________________________ Start LCD Display ___________

#______________________________________________________

#import

importRPi.GPIO as GPIO

import time

# Define GPIO to LCD mapping

LCD_RS = 26

LCD_E  = 19

LCD_D4 = 13

LCD_D5 = 6

LCD_D6 = 5

LCD_D7 = 4

#LCD_D7 = 11

LED_ON = 15

# Define some device constants

LCD_WIDTH = 16    # Maximum characters per line
```

```
LCD_CHR = True
LCD_CMD = False
LCD_LINE_1 = 0x80 # LCD RAM address for the 1st line
LCD_LINE_2 = 0xC0 # LCD RAM address for the 2nd line
# Timing constants
E_PULSE = 0.00005
E_DELAY = 0.00005
#_______________________________________________
#____________________ End LCD Display ___________
#_______________________________________________

#____________________ Start Switch _____________
#_______________________________________________
in1 = 2
#GPIO.setmode(GPIO.BOARD)
GPIO.setmode(GPIO.BCM)
GPIO.setup(in1, GPIO.OUT)
GPIO.output(in1, True)
#_______________________________________________
#____________________ End Switch _______________

#_______________________________________________
#____________________ Start LCD Display _________
#_______________________________________________
try:
while (1):
def main():
    # Main program block
    # Initialise display
lcd_init()
    # Toggle backlight on-off-on
GPIO.output(LED_ON, True)
    #time.sleep(.5)
GPIO.output(LED_ON, False)
    #time.sleep(.5)
GPIO.output(LED_ON, True)
    #time.sleep(.5)
    # Send some centred test
lcd_byte(LCD_LINE_1, LCD_CMD)
lcd_string("Real Time Irri",2)
lcd_byte(LCD_LINE_2, LCD_CMD)
lcd_string("gation System",2)
time.sleep(1) # 3 second delay
    #_________________________________________
    #________________ Start MCP3008_____________
#_______________________________________________
    # Send some right justified text
lcd_byte(LCD_LINE_1, LCD_CMD)
lcd_string("Moisture Pc = ",2)
lcd_byte(LCD_LINE_2, LCD_CMD)
lcd_string( ('%.2f'%((63702 - chan.value)/350)) + '%',2)
time.sleep(1)
    # Turn off backlight
    #print('Moisture Percentage = ', ('%.3f'%(1324343032.324725235))
print('Moisture Percentage = ', '%.2f'%((63702 - chan.value)/350))
sleep(sleepTime)
    #_______________________________________________
#____________________ End MCP3008______________
#_______________________________________________

GPIO.output(LED_ON, False)
```

```
#_____________________ Start Switch ______________
    #_______________________________________________
if ((63702 - chan.value)/350 > 30):
GPIO.output(in1, True)
lcd_byte(LCD_LINE_1, LCD_CMD)
lcd_string("Moisture Pc = ",2)
lcd_byte(LCD_LINE_2, LCD_CMD)
lcd_string( ('%.2f'%((63702 - chan.value)/350)) + '%',2)
time.sleep(1)
GPIO.output(LED_ON, False)
else:
       #_______________________________________________
#_____________________ Start MCP3008______________
#_______________________________________________
       # Send some right justified text
lcd_byte(LCD_LINE_1, LCD_CMD)
lcd_string("Motor On",2)
lcd_byte(LCD_LINE_2, LCD_CMD)
lcd_string( "",2)
       # Turn off backlight
       #print('Moisture Percentage = ', ('%.3f'%(1324343032.324725235))
print('Moisture Percentage = ', '%.2f'%((63702 - chan.value)/350))
sleep(sleepTime)
       #_______________________________________________
#_____________________ End MCP3008______________
#_______________________________________________
GPIO.output(in1, False) # Motor on
       #time.sleep(2) # Wait for 2 seconds
       #GPIO.output(in1, True) # Motor Off
GPIO.output(LED_ON, False)
#_______________________________________________
#_____________________ End Switch ________________
    # Below are functions for lcd display.
deflcd_init():
GPIO.setmode(GPIO.BCM)     # Use BCM GPIO numbers
GPIO.setup(LCD_E, GPIO.OUT)  # E
GPIO.setup(LCD_RS, GPIO.OUT) # RS
GPIO.setup(LCD_D4, GPIO.OUT) # DB4
GPIO.setup(LCD_D5, GPIO.OUT) # DB5
GPIO.setup(LCD_D6, GPIO.OUT) # DB6
GPIO.setup(LCD_D7, GPIO.OUT) # DB7
GPIO.setup(LED_ON, GPIO.OUT) # Backlight enable
     # Initialise display
lcd_byte(0x33,LCD_CMD)
lcd_byte(0x32,LCD_CMD)
lcd_byte(0x28,LCD_CMD)
lcd_byte(0x0C,LCD_CMD)
lcd_byte(0x06,LCD_CMD)
lcd_byte(0x01,LCD_CMD)
deflcd_string(message,style):
     # Send string to display
     # style=1 Left justified
     # style=2 Centred
     # style=3 Right justified
if style==1:
message = message.ljust(LCD_WIDTH," ")
elif style==2:
message = message.center(LCD_WIDTH," ")
elif style==3:
message = message.rjust(LCD_WIDTH," ")
fori in range(LCD_WIDTH):
```

```
lcd_byte(ord(message[i]),LCD_CHR)
deflcd_byte(bits, mode):
    # Send byte to data pins
    # bits = data
    # mode = True  for character
    #        False for command
GPIO.output(LCD_RS, mode) # RS
    # High bits
GPIO.output(LCD_D4, False)
GPIO.output(LCD_D5, False)
GPIO.output(LCD_D6, False)
GPIO.output(LCD_D7, False)
if bits&0x10==0x10:
GPIO.output(LCD_D4, True)
if bits&0x20==0x20:
GPIO.output(LCD_D5, True)
if bits&0x40==0x40:
GPIO.output(LCD_D6, True)
if bits&0x80==0x80:
GPIO.output(LCD_D7, True)
    # Toggle 'Enable' pin
time.sleep(E_DELAY)
GPIO.output(LCD_E, True)
time.sleep(E_PULSE)
GPIO.output(LCD_E, False)
time.sleep(E_DELAY)
    # Low bits
GPIO.output(LCD_D4, False)
GPIO.output(LCD_D5, False)
GPIO.output(LCD_D6, False)
GPIO.output(LCD_D7, False)
if bits&0x01==0x01:
GPIO.output(LCD_D4, True)
if bits&0x02==0x02:
GPIO.output(LCD_D5, True)
if bits&0x04==0x04:
GPIO.output(LCD_D6, True)
if bits&0x08==0x08:
GPIO.output(LCD_D7, True)
    # Toggle 'Enable' pin
time.sleep(E_DELAY)
GPIO.output(LCD_E, True)
time.sleep(E_PULSE)
GPIO.output(LCD_E, False)
time.sleep(E_DELAY)
if __name__ == '__main__':
main()
exceptKeyboardInterrupt:
GPIO.cleanup()
#____________________________________________________
#_______________________ End LCD Display ___________
#____________________________________________________
```