SMART IRRIGATION SYSTEM USING IoT

A PROJECT REPORT SUBMITTED
IN PARTIAL - FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE
OF

MASTER OF COMPUTER APPLICATION

BY

1.Ekansh kumar (MCA/40030/2017)

DEPARTMENT OF COMP. SC. & ENGG.
BIRLA INSTITUTE OF TECHNOLOGY
LALPUR, RANCHI-834001
(2020)

DECLARATION CERTIFICATE

This is to certify that the work presented in the project entitled **“Smart Irrigation System Using IoT”** in partial- fulfillment of the requirement for the award of degree of **MASTER OF COMPUTER APPLICATION,** of Birla Institute Of Technology, Lalpur, Ranchi, is an authentic work carried out under my supervision. To the best of my knowledge, the content of this project does not form a basis for the award of any previous degree to anyone else.

Date: 15/06/2020 **( )**

Prof. Kuntal Mukharjee
Guide Name & Signature
Dept. to Comp.Sc & Engg BIT
Extn. Centre, Lalpur

Academic Coordinator
Dept. Of Comp. Sc. & Engg
Birla Institute Of Technology, Lalpur
Ranchi-834001

In-Charge
Birla Institute Of Technology,Lalpur
Rnachi-834001

CERTIFICATE OF APPROVAL

The foregoing project entitled, **"SMART IRRIGATION SYSTEM USING IOT"** is hereby approved as a creditable study of research topic and has been presented in a satisfactory manner to warrant its acceptance as prerequisite to the degree for which it was submitted.

It is understood that by this approval, the undersigned do not necessarily endorse any conclusion drawn or opinion expressed therein

but approve the thesis for the purpose for which it is submitted.

(Internal Examiner) **(External Examiner)**

(in-Charge)
BITEC,Lalpur,Ranchi

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to our college faculties for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to show my gratitude towards my mentor

Prof. Kuntal Mukherjee Sir.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

ROLL NO:	NAME:	SIGNATURE
MCA/40030/17	EKANSH KUMAR	

PROJECT DESCRIPTION

Agriculture is the backbone of all developed countries. It uses 85% of available percentage continues to be dominant in water consumption because of population growth and increased food demand. Due to this, efficient water management is the major cropping system is arid and semi-arid areas. An automated irrigation system is to overcome irrigation and under irrigation. Over irrigation occurs because of poor distribution or management of waste water, chemical which leads to water pollution. Under irrigation leads to increase soil salinity with consequent buildup of toxic salts on the soil surface in areas with evaporation. To overcome these problems and to reduce the man power smart irrigation system has been used. Most of the farmers use large portions of farming land and it becomes very difficult to reach and track each corner of large lands. Sometime there is a possibility of uneven water sprinkles. This result in the bad quality crops which further leads to financial losses. In this scenario the Smart Irrigation System using Latest IoT **(Internet of Things)** Technology is helpful and leads to ease of farming.

TABLE OF CONTENTS

INTRODUCTION

Agriculture is the major source of income for the largest population in India and is major contributor to Indian economy. However, technological involvement and its usability have to be grown still and cultivated for agro sector in India. Although few initiatives have also been taken by the Indian Government for providing online and mobile messaging services to farmers related to agricultural queries and agro vendor's information to farmers. Based on the survey it is observed that agriculture contributes 27% to GDP, and Provides employment to 70% of Indian population.

IOT is changing the agriculture domain and empowering farmers to fight with the huge difficulties they face. The agriculture must overcome expanding water deficiencies, restricted availability of lands, while meeting the expanding consumption needs of a world population. New innovative IOT applications are addressing these issues and increasing the quality, quantity, sustainability and cost effectiveness of agricultural production.

Agriculture is the backbone of Indian Economy. In today's world, as we see rapid growth in global population, agriculture becomes more important to meet the needs of the human race. However, agriculture requires irrigation and with every year.

we have more water consumption than rainfall, it becomes critical for growers to find ways to conserve water while still achieving the highest yield. But in the present era, the farmers have been using irrigation technique through the manual control in which they irrigate the land at the regular interval.

According to statistics, agriculture uses 85% of available freshwater resources worldwide, and this percentage will continue to be dominant in water consumption because of population growth and increased food demand. There is an urgent need to create strategies based on science and technology for sustainable use of water, including technical, agronomic, managerial and institutional improvements. Agricultural irrigation based on Internet technology is based on crop water requirement rules.

By using Internet technology and sensor network technology we can control water wastage and to maximize the scientific technologies in irrigation methods. Hence it can greatly improve the utilization of water and can increase water productivity.

The Internet of Things (IOT) is a technology where in a mobile device can be used to monitor the function of a device. The Internet of Things (IOT) is concerned with interconnecting communicating objects that are installed at different locations that are possibly distant from each other. Internet of Things (IOT) is a type of network technology, which senses the information from different sensors and makes anything to join the Internet to exchange information.

It can also be used to modify the status of the device. The central processing unit will also include communication device to receive data from the sensors and to be relayed to the user's device. This will be done using a higher communication device such as a Wi-Fi module.

The data processed by the central module is converted to meaningful data and relayed to the user. The user can view the data with the help of a handheld device such as a mobile phone or a tablet. Nowadays water scarcity is a big concern for farming. This project helps the farmers to irrigate the farmland in an efficient manner with automated irrigation system based on soil moisture.

The proposed system has been designed to overcome the unnecessary water flow into the agricultural lands. Temperature, moisture and humidity readings are continuously monitored by using temperature, moisture and humidity sensor and send these values to the assigned IP address. Android application continuously collects the data from that assigned IP address. Once the soil moisture values are exceeded the particular limit then the relay, which is connected to the Adriano microcontroller controls the motor.

The android application is a simple menu driven application, with 4 options. This includes motor status, moisture, temperature and humidity values. The motor status indicates the current status of the pump.

As we know that, plants are the most important thing on the earth. So we have to survive it and should spread greenery as much as we can. People potted plants for gardening and decorated their house by many plants. But in their absence plants are not getting sufficient of water. Generally in the summer and winter season, the humidity of atmosphere get reduced, as a result water level of the soil get reduced and also lack of rain is one of the vital cause of ill plantation . From different survey as we came to know, drying a soil to constant weight at 105°C (sometimes 50-80°C for plant tissue) is the traditional method of arriving at a "dry" sample weight. This temperature is somewhat arbitrary, and clay minerals in particular may contain 10-15% water (dry basis) at 400°C (Gardner 1986).

As temperature grows up more, first water in soil pores evaporates, then water accumulate to mineral surfaces, followed by water between lattice layers and that which forms part of the mineral lattice itself. Irrigation helps to harvest crops, protecting plants against frost, suppressed the weed growth in grain fields; prevent the dust suppression, disposal of sewage and preventing soil consolidation. In earlier, we used watering cans, water channels which are opened and closed manually or backpack sprinklers for irrigation. So, large amount of the water wasted due to that process. So now, we should require improving on the existing or earlier forms of plant irrigation. An automated power plant system needs to be developed to improve efficiency of water use for agricultural crops. So we introduced an intelligent system that has autonomously monitor and control the level of water existing to the plants without any failure or human

effort. Here the system will check continuously the amount of water present in the soil and find out if watering is required for the plants and how much amount is required based on the information acquired from monitoring the soil water content. The system will automatically shut down the motor when sufficient amount of water supplied to the soil. System response tests will determine how much time will take for the system to irrigate potted samples of different soil types, that comes down the dryness level and soil moisture level reaches to the threshold level 1000 to certain value. So there are the advantages of automatic power plant system to the plants include saving money, water, maintenance of human effort and overall convenience.

INTERNET OF THINGS (IOT)

The Internet of things (IoT) is the system of gadgets and home apparatuses, and so on that contain hardware, programming, and availability which enables these things to associate, collaborate and trade data.IoT includes giving Internet network past standard gadgets, for example, work areas, workstations, telephones and tablets, to any scope of customarily non-web empowered physical gadgets and ordinary items. These gadgets are implanted with innovation and can impart and associate over the Internet, and furthermore they can be remotely checked and controlled. IoT contraptions are a bit of the greater thought of home computerization, which can join lighting, warming and cooling, media and security frameworks. Whole deal favorable circumstances could join essentialness reserves by means of therefore ensuring lights and equipment are slaughtered. An insightful home or motorized home could be established on a phase or focuses that control sharp devices and machines. For instance, using Apple's Home Kit, producers can have their home things and additional items obliged by an application in iOS contraptions, for instance, the iPhone and the Apple Watch. This could be a dedicated application or iOS neighborhood applications, for instance, Siri. This can be appeared because of Lenovo's Shrewd Home Fundamentals, which is a line of splendid home contraptions that are controlled through Apple's Home application or Siri without the necessity for a Wi-Fi connect. There are furthermore committed splendid home focus focuses that are offered as free stages to relate particular sharp home things and these fuse the Amazon Reverberation, Google Home, Apple's Home Pod, and Samsung's Smart Things Hub. IoT contraptions can be used to screen and control the mechanical, electrical and electronic systems used in various sorts of structures (e.g., open and private, current, associations, or private) in home motorization and building computerization structures. In this special circumstance, three crucial zones are being solicited in writing: The blend of the Web with structure essentialness the officials systems to make imperativeness successful and IOT-driven "sharp structures".

The potential strategies for steady watching for diminishing imperativeness utilization and checking occupant practices. The consolidation of canny contraptions in the amassed condition and how they may to understand that how for the most part will be used in future applications. The IoT can comprehend the steady compromise of various gathering contraptions outfitted with identifying, ID, taking care of, correspondence, incitation, and frameworks organization limits. In light of such an exceedingly joined sharp cyber physical space, it opens the best approach to make very surprising business and market open entryways for assembling. Framework control and the administrators of amassing equipment, asset and condition the board, or gathering procedure control bring the IoT inside the area of mechanical applications and splendid collecting too. The IoT adroit structures enable quick gathering of new things, dynamic response to thing solicitations, and consistent upgrade of collecting creation and generation arrange frameworks, by frameworks organization mechanical assembly, sensors and control systems together. Modernized control systems to robotize procedure controls, manager instruments and organization information structures to improve plant prosperity and security are inside the area of the IoT.

Anyway it moreover stretches out itself to asset the board through judicious help, verifiable evaluation, and estimations to intensify dependability. Sharp present day organization structures can in like manner be composed with the Savvy Network, thusly engaging consistent imperativeness improvement. Estimations, automated controls, plant upgrade, prosperity and security the board, and various limits are given by a far reaching number of sorted out sensors. The term mechanical Web of things (IoT) is routinely experienced in the amassing organizations, insinuating the cutting edge subset of the IoT.

IoT in amassing could deliver so much business regard that it will at last lead to the Fourth Modern Upheaval, so the supposed Business 4.0. It is surveyed that later on, productive associations will presumably assemble their salary through Web of things by making new arrangements of activity and improve gainfulness, abuse examination for advancement, and change workforce. The ability of advancement by executing IoT may create $12 trillion of overall Gross household item by 2030.

USE OF INTERNET OF THINGS IN THE FIELD OF AGRICULTURE

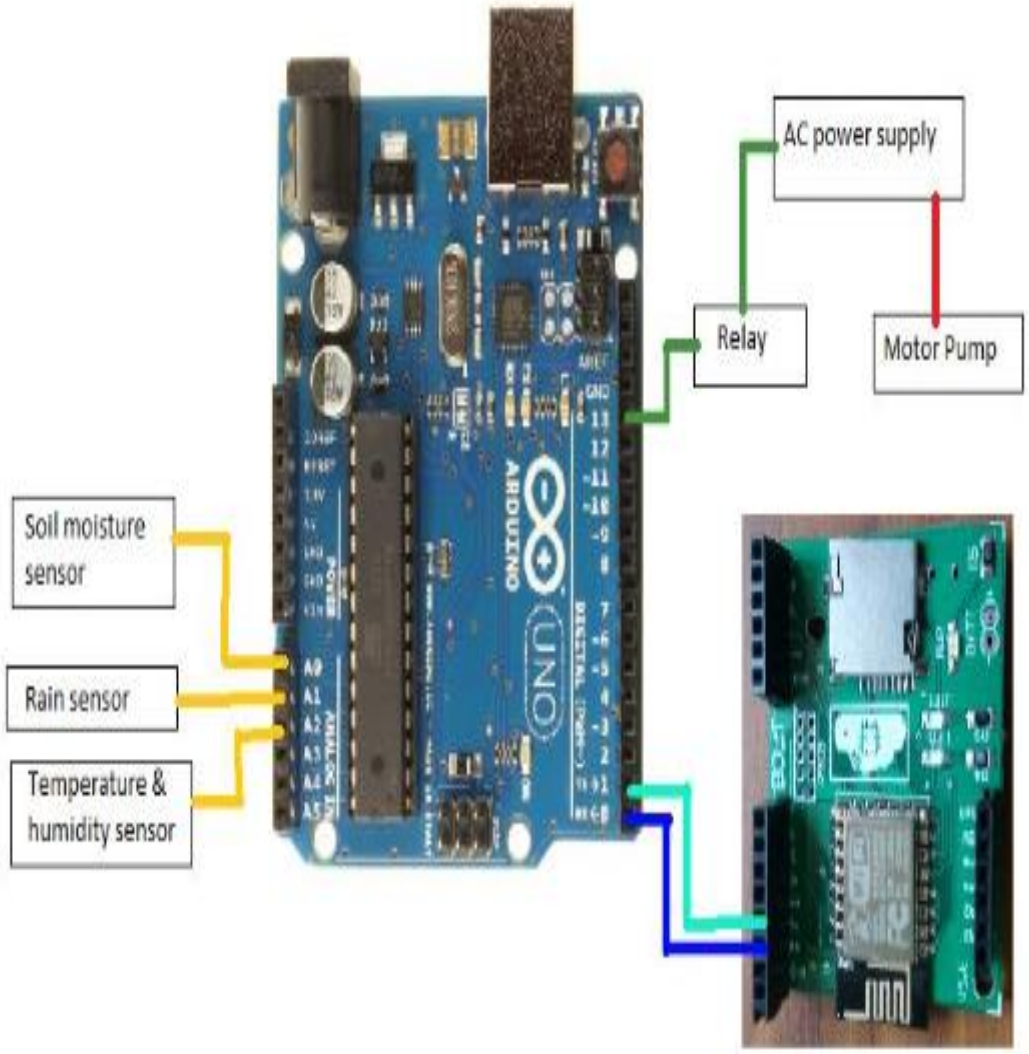
There are different IoT applications in cultivating, for example, gathering information on temperature, precipitation, moistness, wind speed, and soil content. This information can be utilized to mechanize cultivating systems and take educated choices to improve quality, amount and limit hazard and waste, decrease exertion required to oversee crops. For instance, screen soil temperature and dampness from a far distance, and apply IoT-procured information to accuracy preparation programs. There are various examples of use of IoT in practical application and one of them is explained below: In August 2018, Toyota Tsusho began an organization with Microsoft to make fish cultivating devices utilizing the Microsoft Azure application suite for IoT advancements identified with water the board, which was created to some degree by analysts from Kindai University, the water siphon systems utilized man-made brainpower to tally the quantity of fish on a transport line and break down the quantity of fishes and in the long run reason the viability of water stream from the information the fish give. The particular PC projects utilized in the process are under the Azure Machine Learning and the Azure IoT Hub stages.

Working with Smart Irrigation System using IoT

In the agriculture field, sensors are used like soil moisture. The information received from the sensors is sent to the Database folder through the Android device. In the control section, the system is activated using the application, this is finished using the ON/OFF buttons in the application. Also, this system is automatically activated when the soil moisture is low, the pump is switched ON based on the moisture content.

The application has a feature like taking some time from the user and water the agriculture field when the time comes. In this system, there is a switch used to turn off the water supply if the system fails. Other parameters such as the moisture sensor demonstrate the threshold price and the level of water in the soil.

Further, this project can be enhanced by designing this system for large acres of soil. Also, this project can be incorporated to make sure the value of the soil and the expansion of harvest in each soil. The microcontroller and sensors are successfully interfaced and wireless communication is attained between a variety of nodes.



Smart Irrigation System using IoT

Also, further this proposed system can be enhanced by adding up machine learning algorithms, which are capable to study and recognize the necessities of the crop, this would aid the agriculture field to be an automatic system. The inspections and outcomes tell us that this result can be executed for a lessening of water loss and decrease the manpower necessary for a field.

From the above information, finally, we can conclude that the hardware components of this system interfaces with all the sensors. The system is powered by a power source, and the system has been checked for watering an agriculture field. If you have any queries regarding in implementing irrigation projects, please give your feedback by commenting in the comment section below. Here is a question for you, what is the function of IOT(Internet of Things)

VARIOUS NETWORKS USED FOR WORKING WITH IOT:

The Internet of things handles a huge amount of data in everyday life and to handle such heavy data scalability is the major factor to manage data received from billions of devices, IPv6 network architecture will prove to be a boon for handling the network layer scalability requires huge scalability. To enable light weight data transport IETF's Constrained Application Protocol will provide a great platform. Fog computing is a feasible method to handle the outburst of huge amount of data.

• Various methods of connectivity

❖Wireless connectivity for short range devices:-

- Bluetooth mesh networking – Specification providing a mesh networking variant to Bluetooth low energy with increased number of nodes and standardized application layer
- Light-Fidelity – Wireless communication technology similar to the Wi-Fi standard, but using visible light communication for increased bandwidth.

- Near-field communication – Communication protocols enabling two electronic devices to communicate within a 4 cm range.

- QR codes and barcodes – Machine readable optical tags that contain information about the product to which they are attached.

- Radio-frequency identification (RFID) – It is a technology which uses electromagnetic fields to read data stored in tags embedded in other items.

- Transport Layer Security – Network security protocol.

- Wi-Fi – technology for local area networking based on the IEEE 802.11 standard, where devices may communicate through a shared access point or directly between individual devices.

- ZigBee – It is a communication protocol for personal area networking based on the IEEE 802.15.4 standard. It provides low power consumption, low data rate, low cost, and high throughput.

❖ Medium-range wireless

- Medium-range wireless o LTE-Advanced – It is high-speed communication specification for mobile networks. It provides enhancements to the LTE standard with extended coverage, higher throughput, and lower latency.

❖Long-range wireless

- Low-control wide-area networking – Remote frameworks are proposed to allow long-go correspondence with uninformed rate, definitely decreasing power and cost for transmission. Available technologies and protocols are: LoRaWan, Sigfox, NB-IoT, Weightless.

❖Wired¬

- Ethernet – It is broadly useful systems administration standard utilizing twisted pair and fiber optic connections alongside hubs or switches.
- Power-line communication (PLC) – It is a communication technology which uses electrical wiring to carry power and data. For eg: HomePlug use PLC for networking IoT devices.

PROPOSED SYSTEM

This system is a combination of hardware and software components. The hardware part consists of different sensors like soil moisture sensor, photocell sensor, etc whereas the software part consists of an android based application connected to the arduino board and other hardware components using Internet of Things (IoT). The android based application consists of signals and a database in which readings are displayed from sensors and are inserted using the hardware. The improvement in irrigation system using wireless network is a solution to achieve water conservation as well as improvement in irrigation process. This research tries to automate the process of irrigation on the farmland by monitoring the soil water level of the soil relative to the plant being cultivated and the adaptively sprinkling water to simulate the effect of rainfall.

Now a day agriculture field is facing lots of problems due to lack of water resources. In order to help the farmers to overcome the difficulties, smart irrigation system has been used. In this system, various sensors such as pH,soil moisture, DHT11, PIR (intruder detecting system) and pressure sensors are connected to the input pins of arduino microcontroller. The sensed values from the sensors are displayed inLCD. If the sensed value goes beyond the threshold values set in the program, the pump will be automatically switched ON/OFF by there lay circuit and it is connected to the driver circuit which helps to switch the voltage. The farmer will be intimated about the current field condition through GSM module and also updated in the web page. By using this system, the farmer can access the details about the condition of the field anywhere at any time.

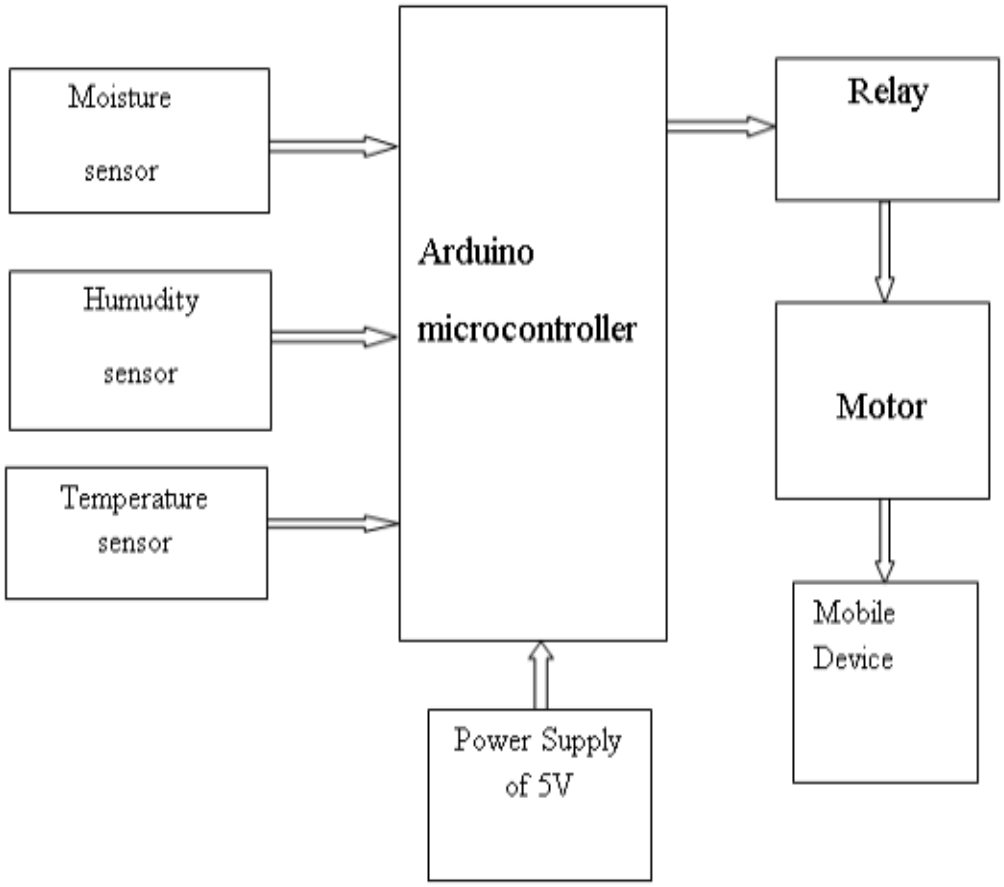


Figure 3.1: Block Diagram of Automatic Irrigation System

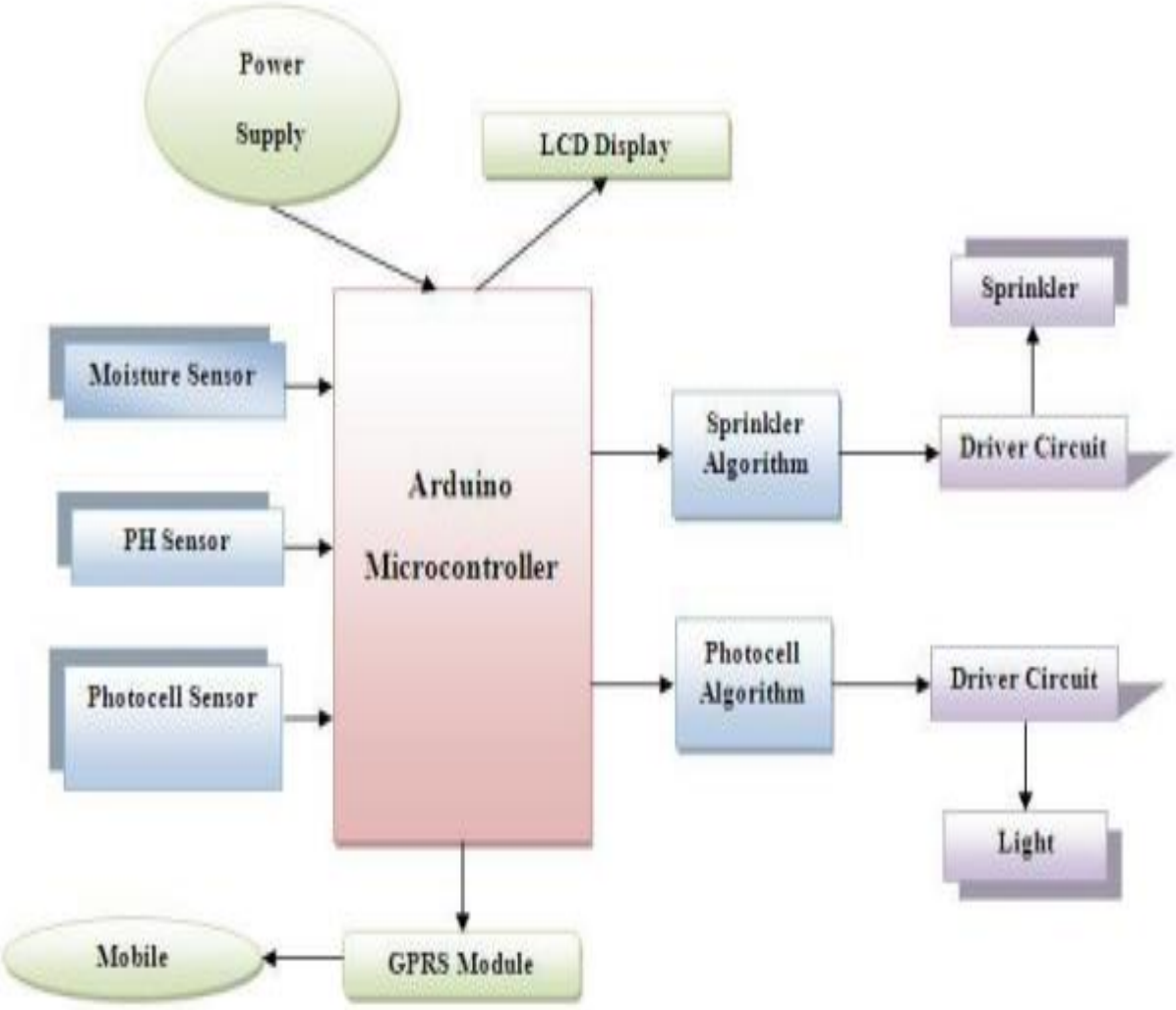


Figure 1. Block Diagram

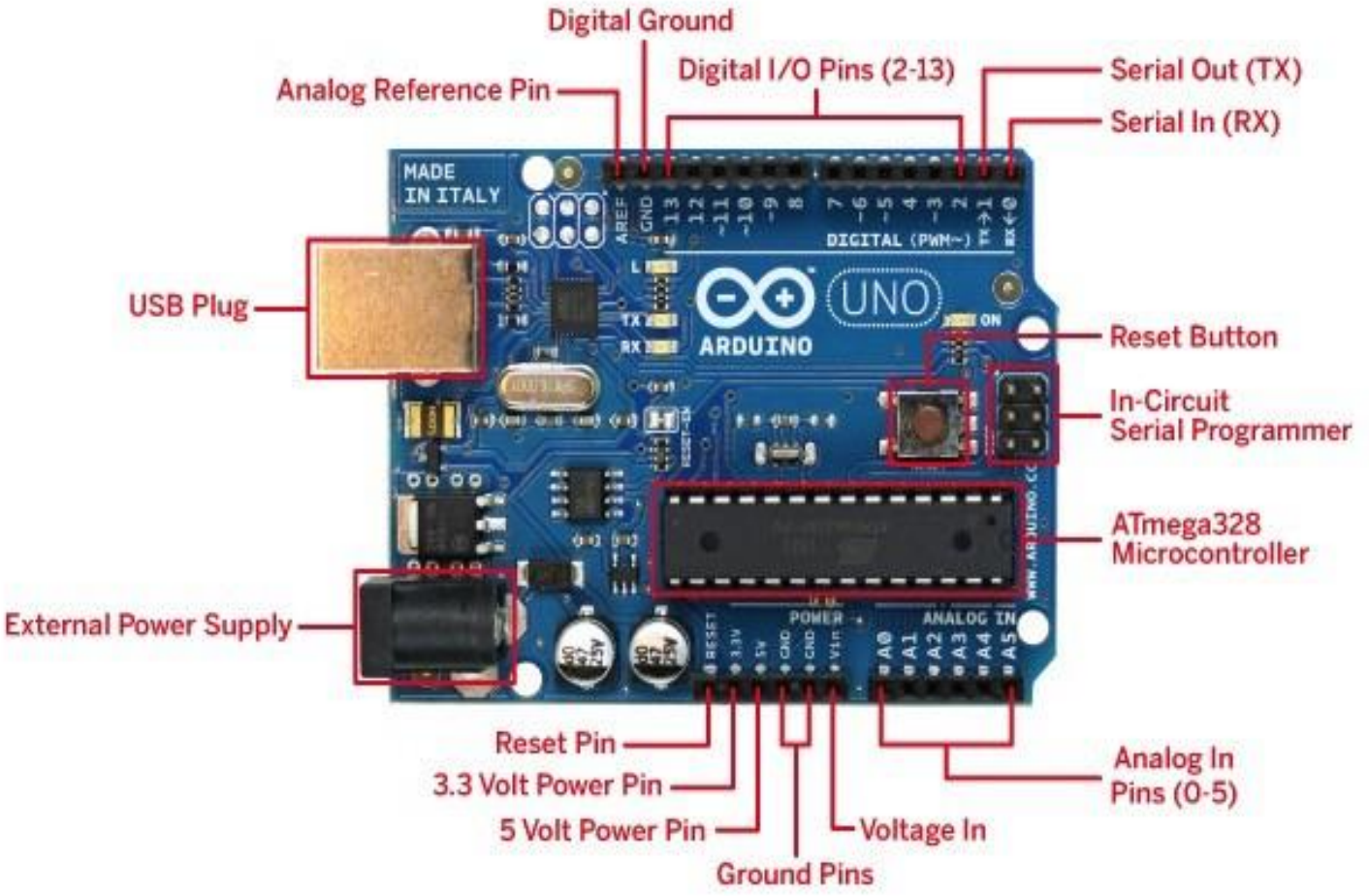
Figure 3.1 shows the block diagram of smart irrigation system with IoT. Farmers start to utilize various monitoring and controlled system in order to increase the yield with help of automation of an agricultural parameters like temperature, humidity and soil moisture are monitored and control the system which can help the farmers to improve the yield. This proposed work includes an embedded system for automatic control of irrigation. This project has wireless sensor network for real-time sensing of an irrigation system. This system provides uniform and required level of water for the agricultural farm and it avoids water wastage. When the moisture level in the soil reaches below threshold value then system automatically switch ON the motor. When the water level reaches normal level the motor automatically switch OFF. The sensed parameters and current status of the motor will be displayed on user's android application.

Arduino UNO Board

The Arduino UNO is one of the most used microcontrollers in the industry. It is very easy to handle, convenient, and use. The coding of this microcontroller is very simple. The program of this microcontroller is considered as unstable due to the flash memory technology. The applications of this microcontroller involve a wide range of applications like security, home appliances, remote sensors, and industrial automation. This microcontroller has the ability to be joined on the internet and perform as a server too.

Arduino UNO

The Microcontroller used here is an Arduino UNO. The UNO is a Microcontroller board based on ATMEGA 328P. The ATMEGA 328P has 32kB of flash memory for storing code. The board has 14 digital input and output pins, 6 analoginputs, 16 MHz quartz crystal, USB, an ICSP circuit and a reset button. The UNO can be programmed with the Arduino software.



SENSORS

SOIL MOISTURE SENSOR

Soil Moisture sensor is used to measure the moisture content present in the soil. When the soil moisture value read by the sensor is above the threshold value, low level (0V) will be the digital output and if it is below the threshold level, high level (5V) will be the digital output. The digital pin is used to directly read current soil moisture value to see if it is above threshold or not. The threshold voltage can be regulated with help of potentiometer.

PH SENSOR

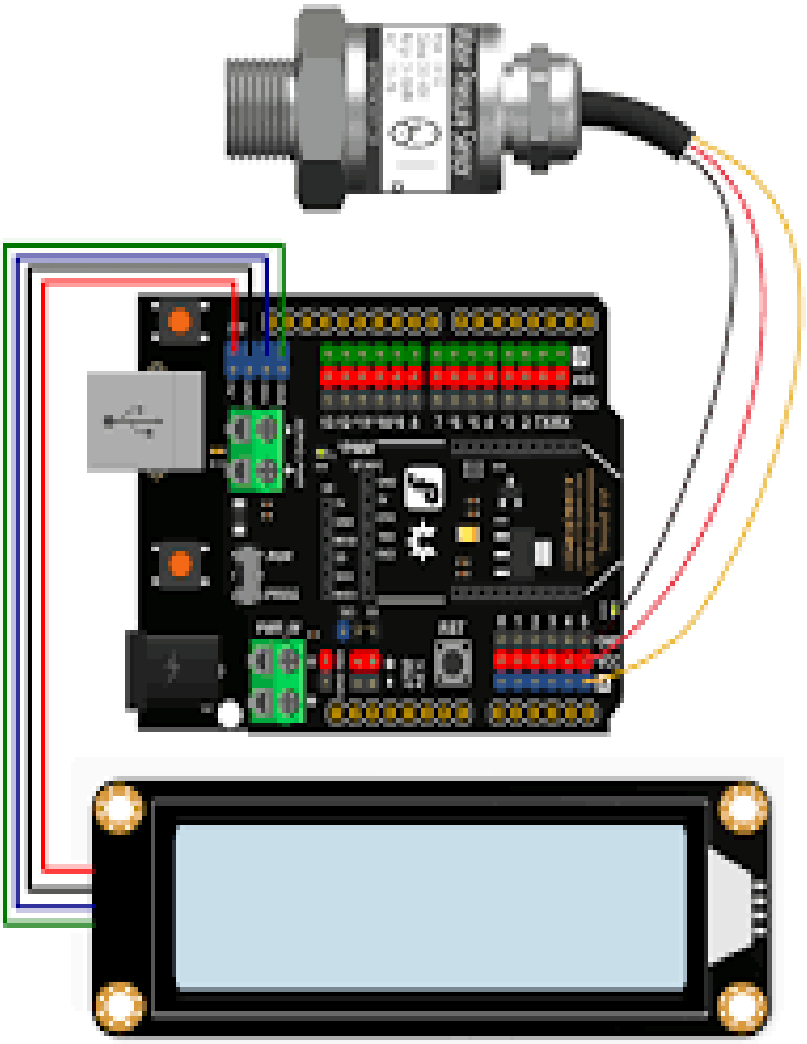
PH is the measure of acidity or alkalinity of water solution which is determined by the relative number of hydrogen (H+) or hydroxyl (OH-) ions present. The pH value (below 7) is said to be acidic and (above 7) is said to be basic. The pH of a solution can change with temperature respectively.

DHT11 SENSOR

DHT11 sensor is used for measuring temperature and humidity. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air. This sensor is cost effective, provides low power consumption and up-to 20 meter signal transmission is possible.

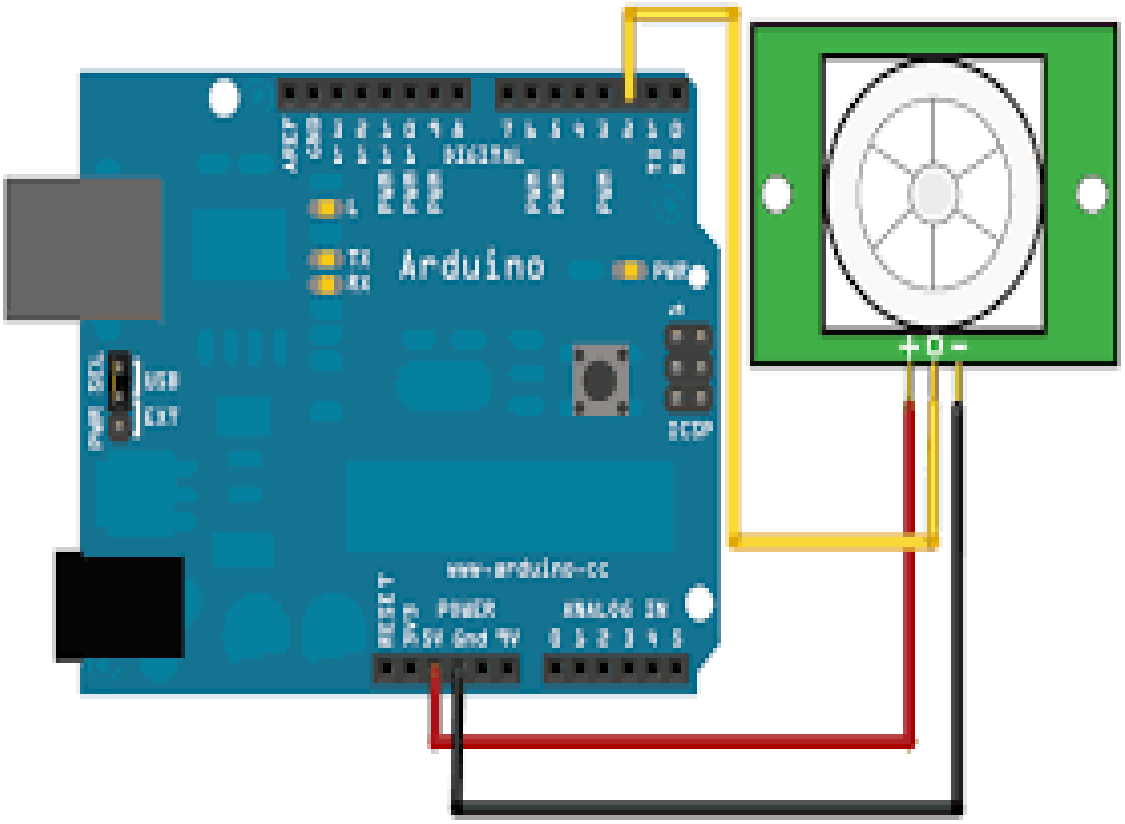
PRESSURE SENSOR

The differential Pressure transmitter is used for measuring trace of differential pressure, PCB will transduce it to differential pressure signal thereby it can be used for weather forecasting.



PIR SENSOR

It is a Passive Infrared sensor which detects the motion with the variation of Infrared radiation. It can cover up to 10 meters at an angle of ±15 degrees. PIR is as same as outdoor light with the motion detector and reacts to movements made by objects that radiate heat.

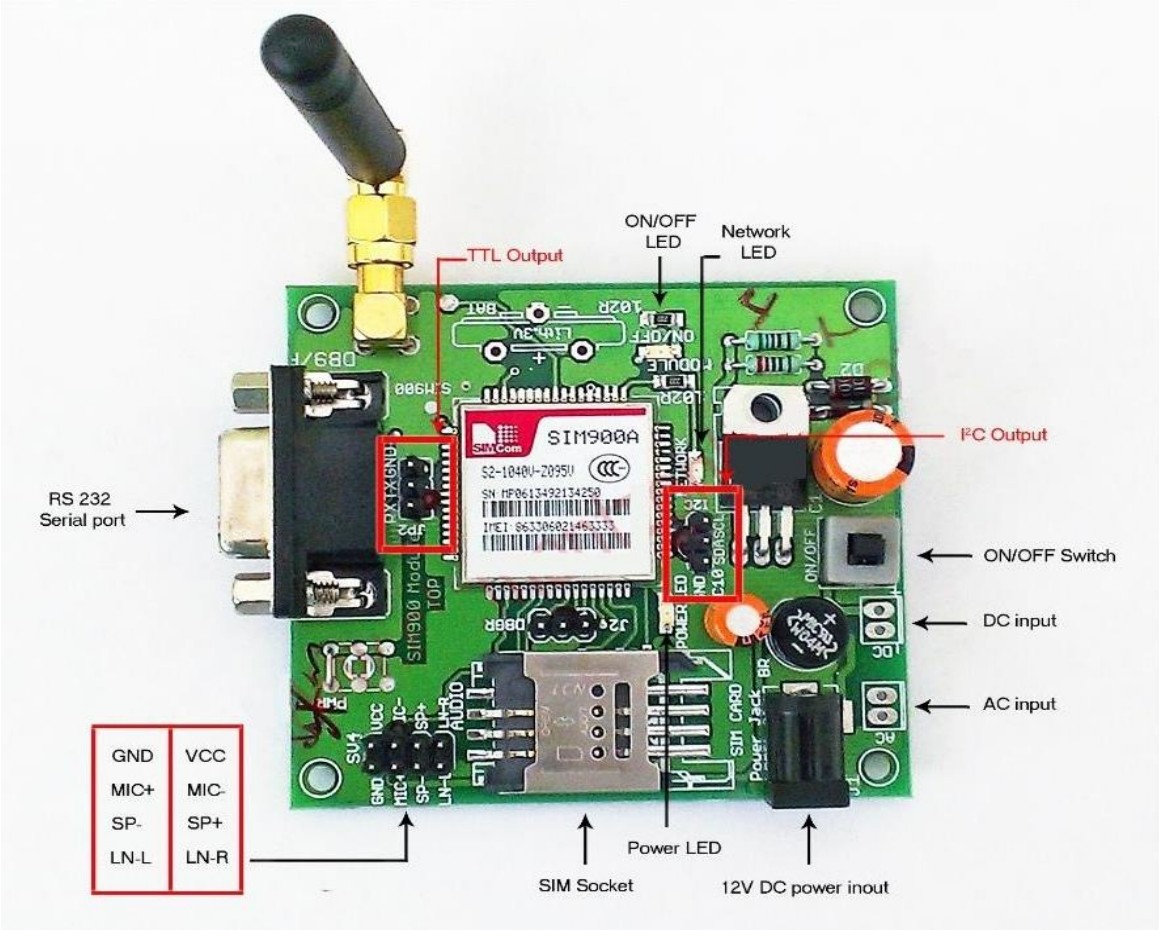


WI-FI MODULE

The ESP8266 Wi-Fi module is a self - contained SOC (System on Chip) with integrated TCP/IP (Transmission Control Protocol/Internet Protocol) protocol stack that can give any microcontroller access to any Wi-Fi network. Each ESP8266 module comes pre-programmed meaning, it can be simply hooked up to Arduino device to get Wi-Fi ability. This
module has a powerful enough on-boarding process and highstorage capacity that allows it to be integrated with the sensors and other application specific devices.

GSM MODULE

GSM (Global System for Mobile Communication) is a standard developed by the European Telecommunication Standards Institute (ETSI) to describe protocols for second-generations (2G) digital cellular networks used by mobile phones. GSM describes a digital, circuit-switched network optimized for full duplex voice telephony and also expanded to include data communications, packet data transport via GPRS (General Packet Radio Services). The longest distance the GSM specification supports in practical is 35 kilometers (22mi).



METHODOLOGY

Water sprinkler control was achieved by setting a threshold value at which irrigation should begin. When the sensors setect moisture content before the threshold, the sprinklers are switched on till the soil is completely moist. Figure 5 shows the flow chart of the system.

The hardware device comprises of moisture sensors, Arduino Uno and GSMGPRS SIM900A modem. The GSM modem is primary responsible for transmitting the data obtained from arduino to internet. This is done using AT commands which are depicted in the following table.

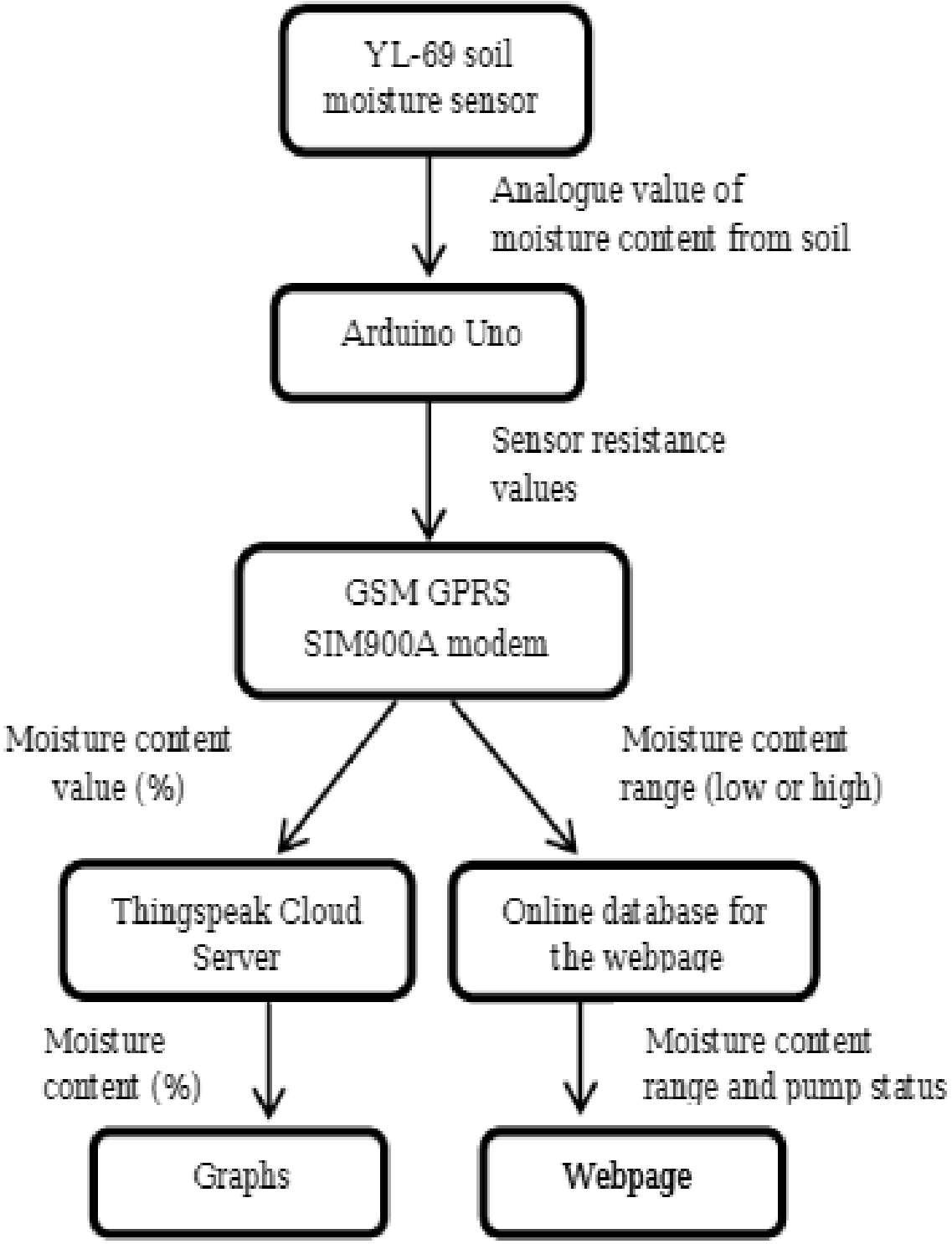


Figure 4 : Flow chart of the system

Table 1 : AT Commands

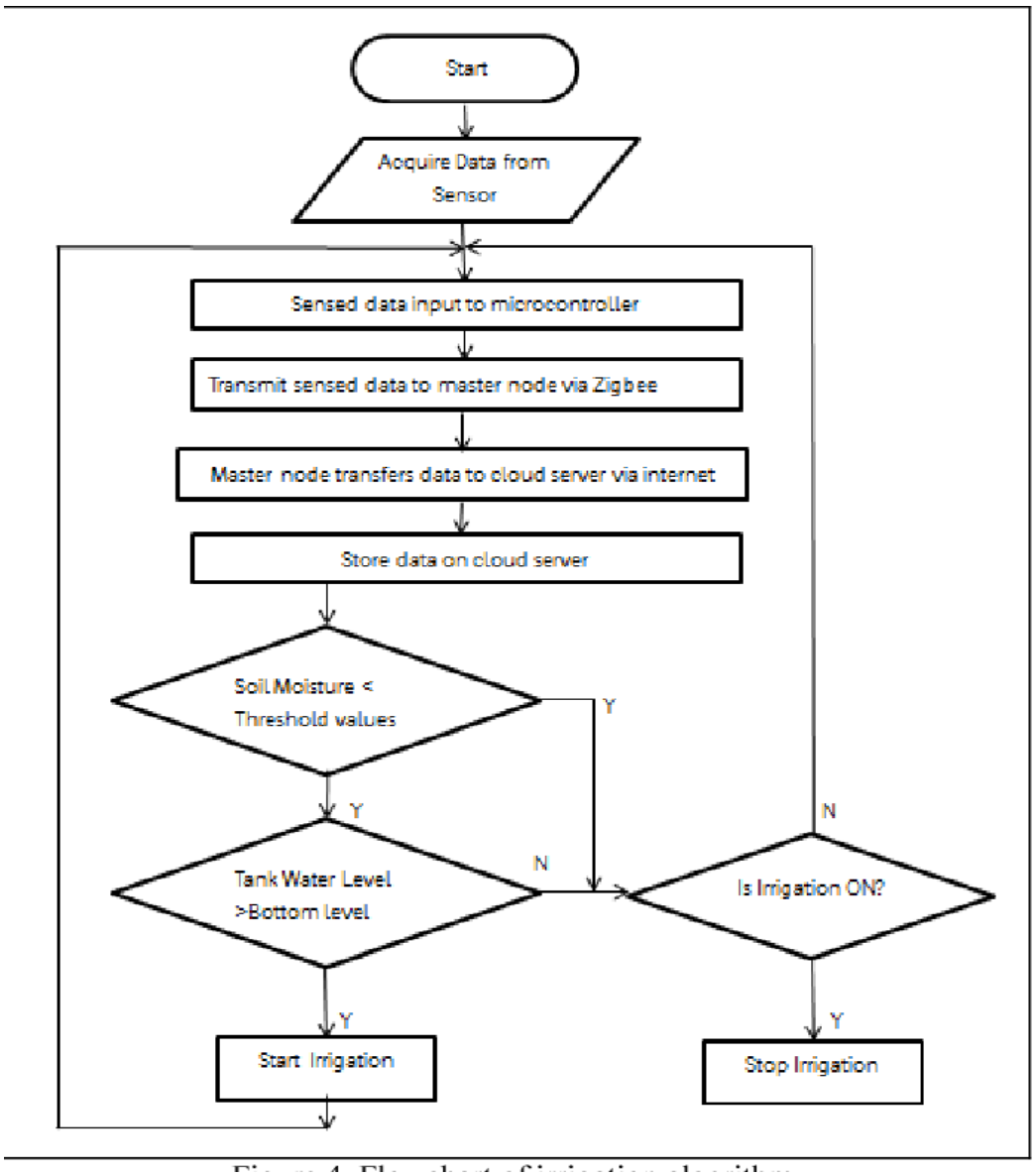
Command	Meaning
AT+CSQ	Signal Quality Check
AT+CGATT	Attach or Detach from GPRS support
AT+SAPBR= 3,1,\"CONTYPE\",\"GPRS\"	Setting the SAPBR connection type to GPRS
AT+SAPBR= 3,1,\"APN\",\"airtel.gprs.com\"	Setting the APN string
AT+SAPBR=1,1	Setting the SAPBR
AT+HTTPPARA=\"URL\", \"http://smartfarmsiot.orgfree.com \"	Setting the httppara, website which needs to be accessed
AT+HTTPACTION=0	Submit the request
AT+CIPSEND	Request initiation of data sending

The information from sensors is transmitted to an online database from where it is used to display on a website. The webpage displays the moisture content in soil which has been divided into two categories : Low and High. Pump is to be switched on when the moisture content is low. The threshold values depend on the type of soil used.

Readings from the two sensors were also transmitted to a THINGSPEAK channel to obtain graphs. ThingSpeak is an open data platform and API for the Internet of Things that enables you to collect, store, analyze, visualize, and act on data from sensors or actuators, such as Arduino.

FLOW CHART

A flowchart is a graphic representation of a logic sequence, work or manufacturing process, organization chart, or similar formalized structure. The flowchart is a means to visually present the flow of data through an information processing systems.



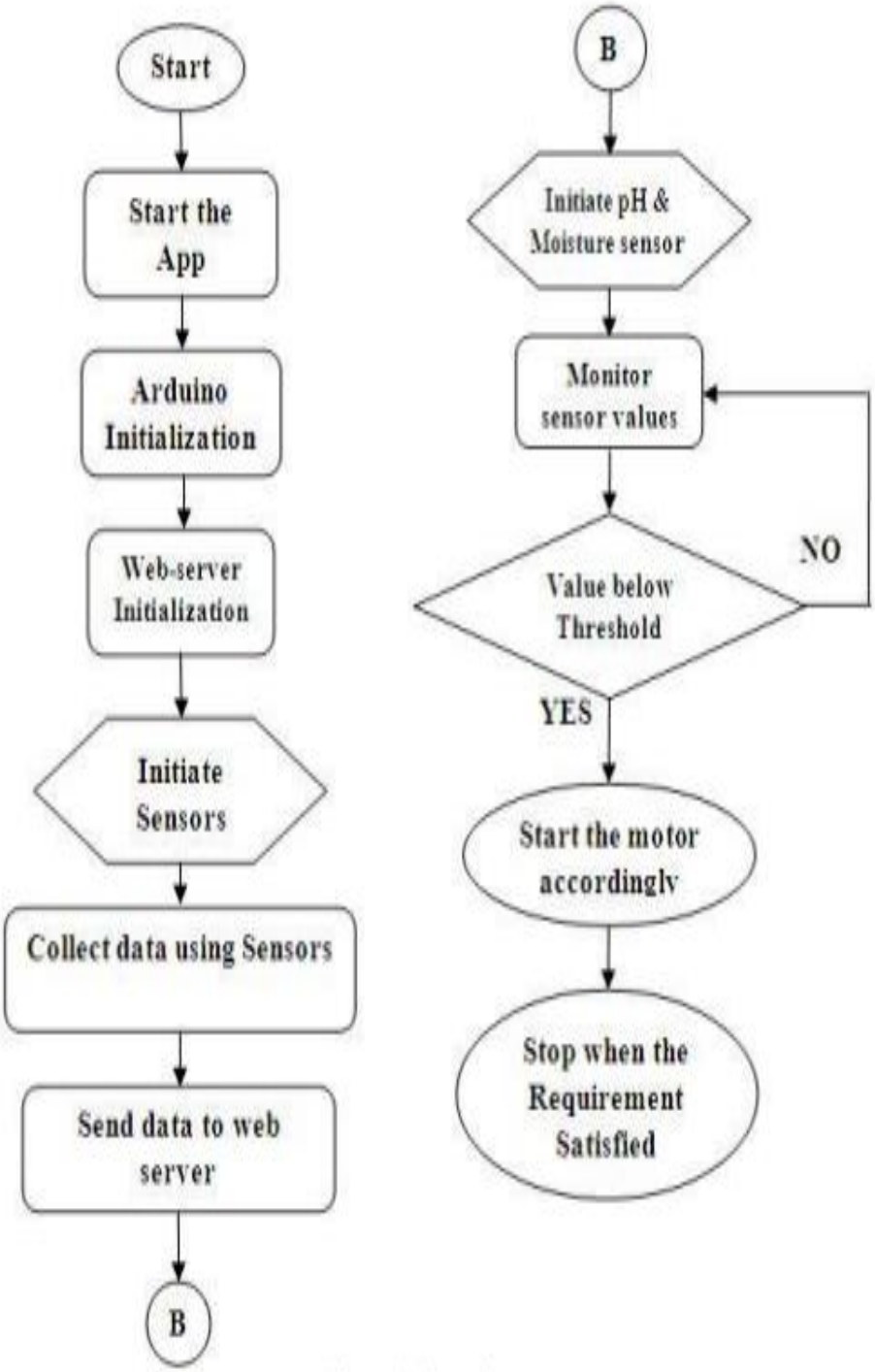


Figure 2. Flow Chart

In this paper, we are using a Mesh topology in which sensor nodes are placed in the farm area. Sensors in our proposed topology are mobile where as the base station is stationary and it collects the data from sensor nodes and process them. This work proposes that how to deploy the sensed data to the base station in Wireless Sensor Networks. For this purpose firstly set the farm area . pH sensor checks whether the soil is alkaline or acidic. Soil should have proper proportion of nutrients which is essential for the plant growth. Also, with help of pH we can determine for what type of plant the soil is feasible. In this project, we have added a pH sensor to check the acidity of the soil and give constant updates to the android application about the same. Sprinkler Algorithm The moisture sensor gives the water content level in the soil and sends it to the arduino. It will process this data by comparing it with the threshold value if it is less than the predefined threshold value then start the irrigation.

Reading<Threshold value >Start the irrigation System

Photocell Algorithm Light is very important for the plants. Plants convert light energy into their own building material. The photocell sensor measures the level of availability of the lights to the plants. Arduino compares the level of light with the predefined threshold value if it is less than the threshold value then we can provide artificial lights for the plants so that the plants can get appropriate level of light.

Reading<Threshold value >Start the artificial light

TEMPERATURE AND HUMIDITY SENSOR

This below Figure 4.1 shows the sensed values of temperature and humidity.

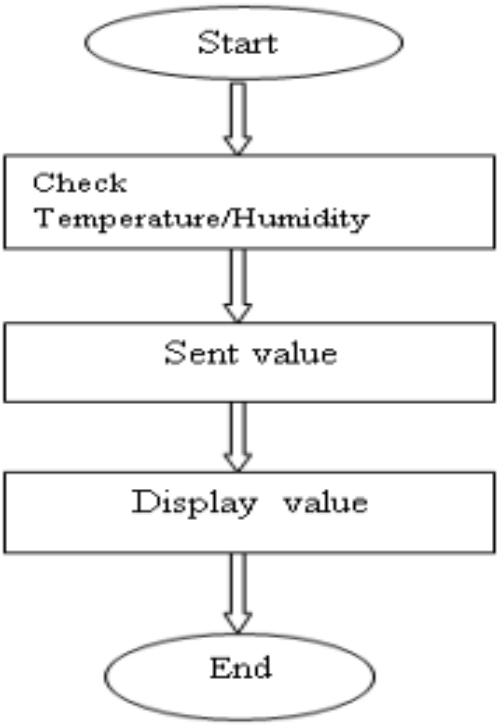


Figure 4.1: Flowchart of Temperature/Humidity Sensor

The DHT11 is a basic, digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermostat to measure the surrounding air, and spits out a digital signal on the data pin(no analog pins needed).It is simple to use, but requires careful timing to grab data. Humidity sensors are used for measuring moisture content in the atmosphere. Then current temperature, humidity values are send to the microcontroller, those values will display in the users screen.

SOIL MOISTURE SENSOR

This below Figure 4.2 shows the procedure of displaying soil moisture value .

Soil moisture sensors measure the water content in soil. Moisture in the soil is an important component in the atmospheric water cycle.Sensor module outputs a high level of resistance when the soil moisture is low. It has both digital and analog outputs. Digital output is simple to use, but it is not as accurate as analog output based on moisture level motor gets turn on/off automatically.

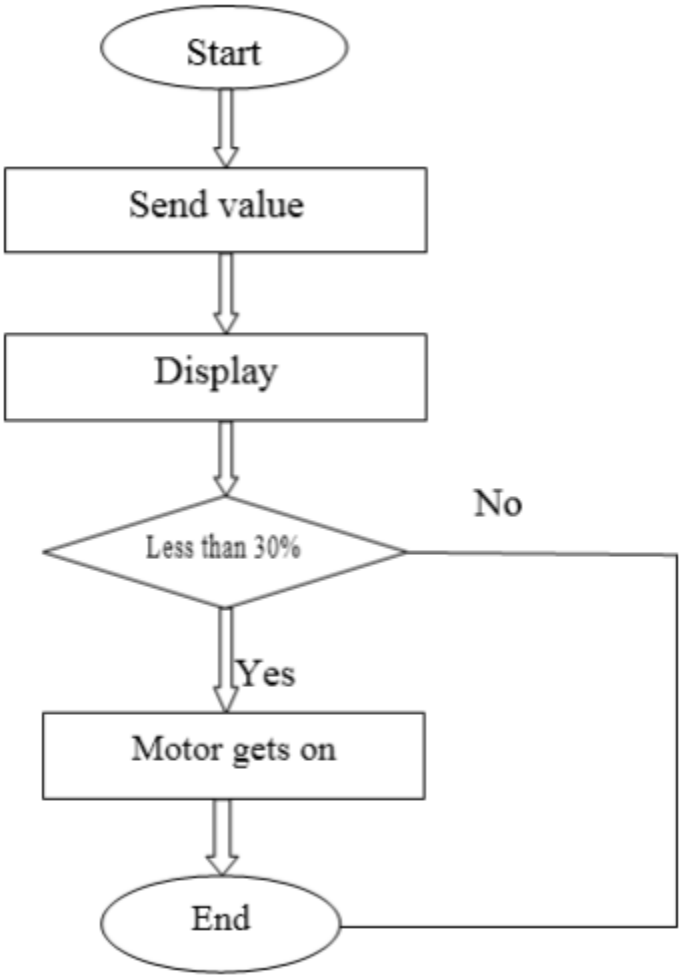


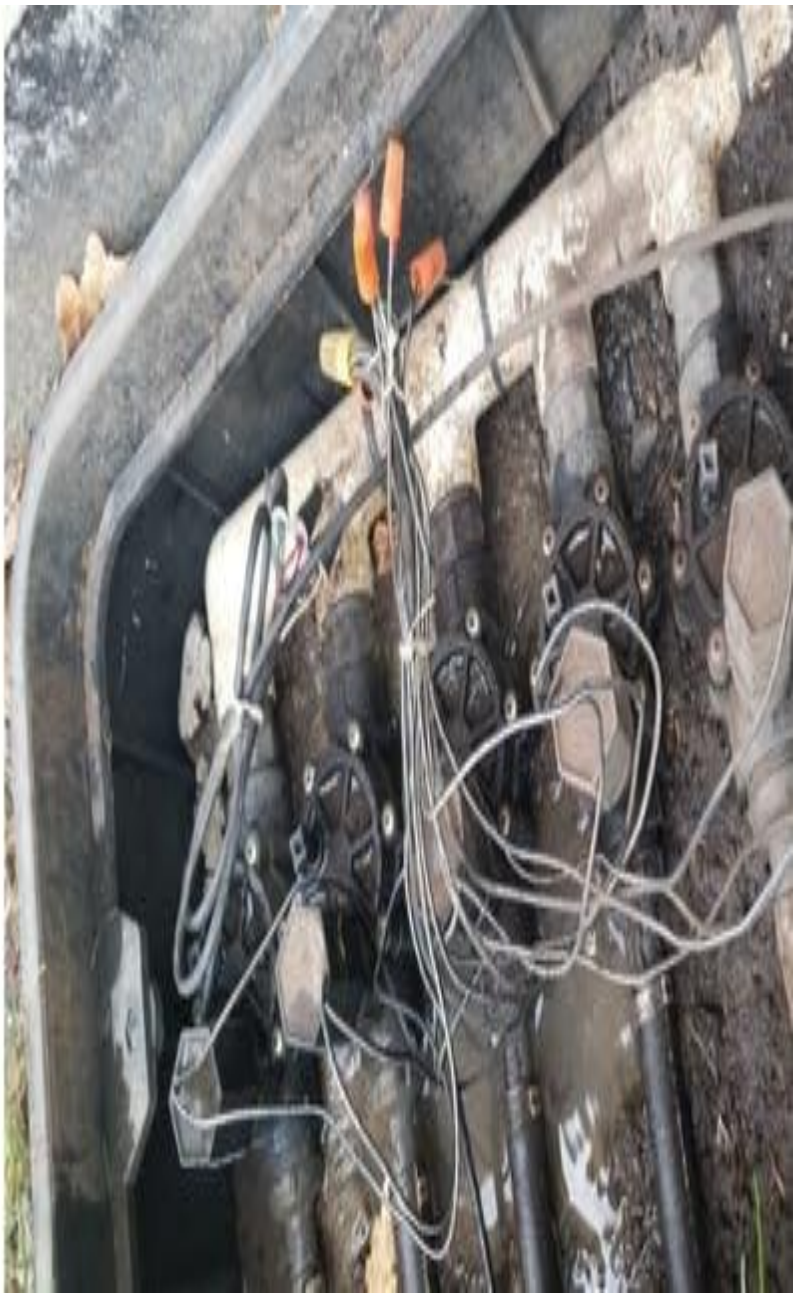
Figure 4.2: Flow chart of Soil moisture sensor

Things used in this project

Hardware components

- NodeMCU ESP8266 Breakout Board
- PCF8575 IO Expander Board Module I2C to 16IO
- 8-Channel Relay Module
- AC/DC buck converter
- Soil Moisture Sensor Module
- Water Pump Module
- Relay Module
- DHT11
- Connecting Wires

Irrigation system basic functionality



Before replacing the controller, lets define what is needed. Most irrigation system is powered by 24v AC (controller & solenoid valves) like the one installed at my house. Each valves define an area (Ex.: Zone 1, Zone 2, Zone 3, etc.) and when powered, water start flowing trough the pipes and activate one or multiple sprinklers for

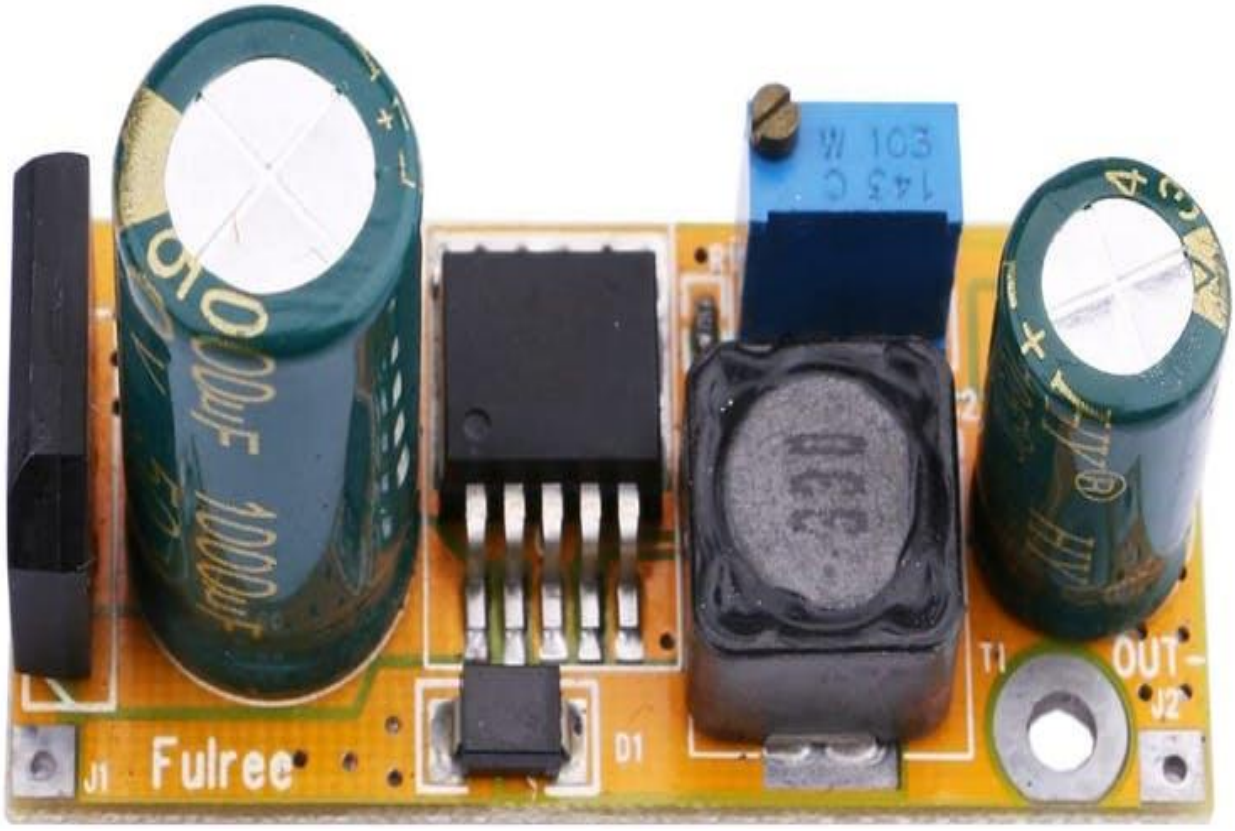
water irrigation.

The wires connectivity is as followed. Tie all common wires together (see diagram) and attach (common wires) to ONLY one of the 24vac wire. The other wire from the 24vac will be added to the relay board (see below for detail).

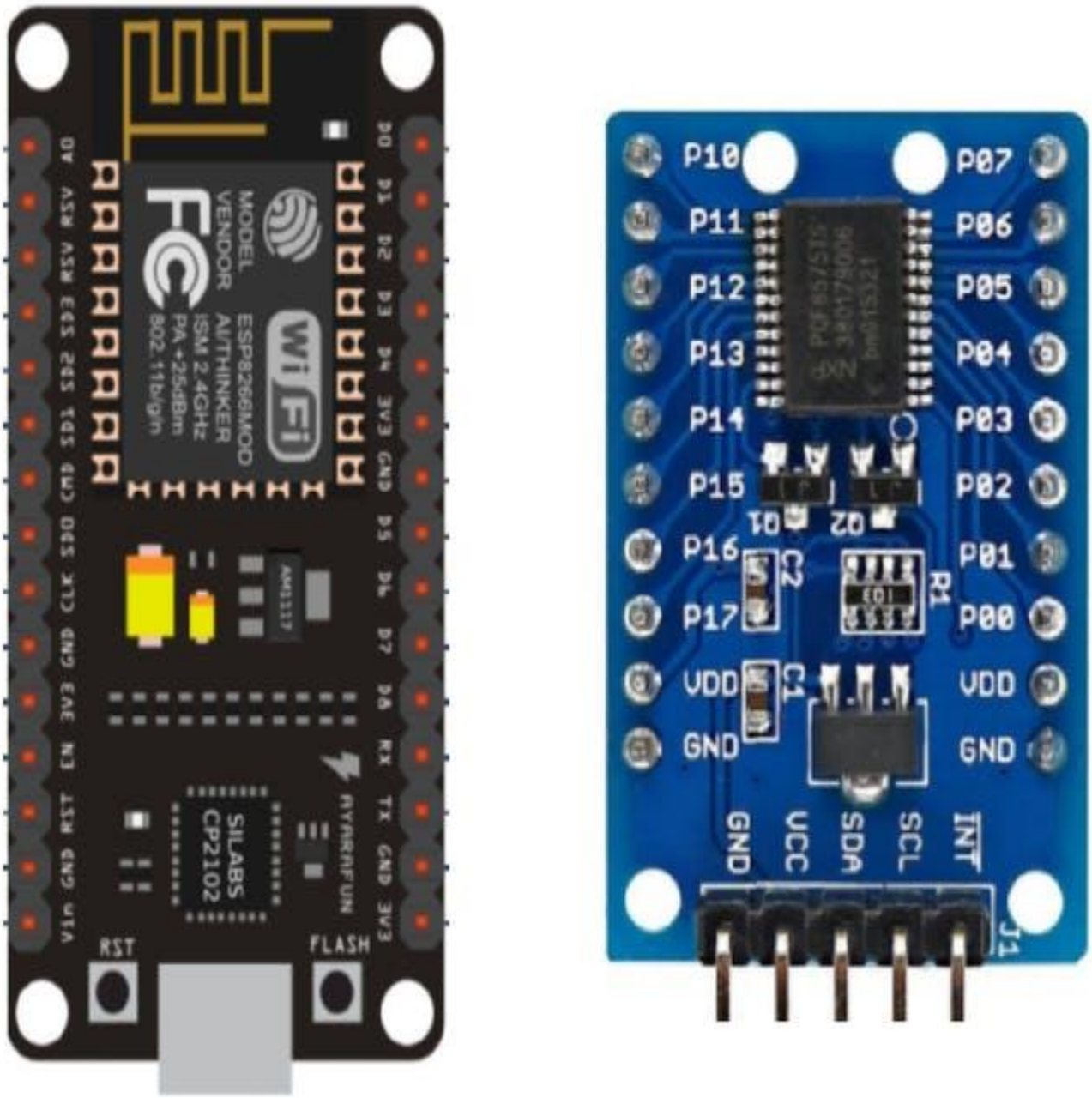
PARTS

I decided to keep the original power supply of 24v AC and reuse it to power my own controller and the solenoid valves.

I decided to keep the original power supply of 24v AC and reuse it to power my own controller and the solenoid valves.



AC/DC to DC Step Down Converter - Because I want to use ONLY one power supply (the original 24vAC), its needed to convert the input AC voltage and output (using trim pot) 5v DC to power the electronic components (relay, nodeMCU, IO expander) of my controller replacement.



Node MCU&PCF8575(Blue)

The NodeMCU (ESP12-E) is the brain of the operations for most of the activities performed (see below), but lacks in IO ports (this is where the PCF8575 shine).

- Fetching the UTC time
- Hosting a web portal
- Activate or stop any zone (valve)
- Switch between manual & automatic mode

The PCF8575 is an IO expander containing 16-bit quasi-bidirectional port. Communication to this chip is done with I2C (only 2 wires needed). What I like about this chip is:

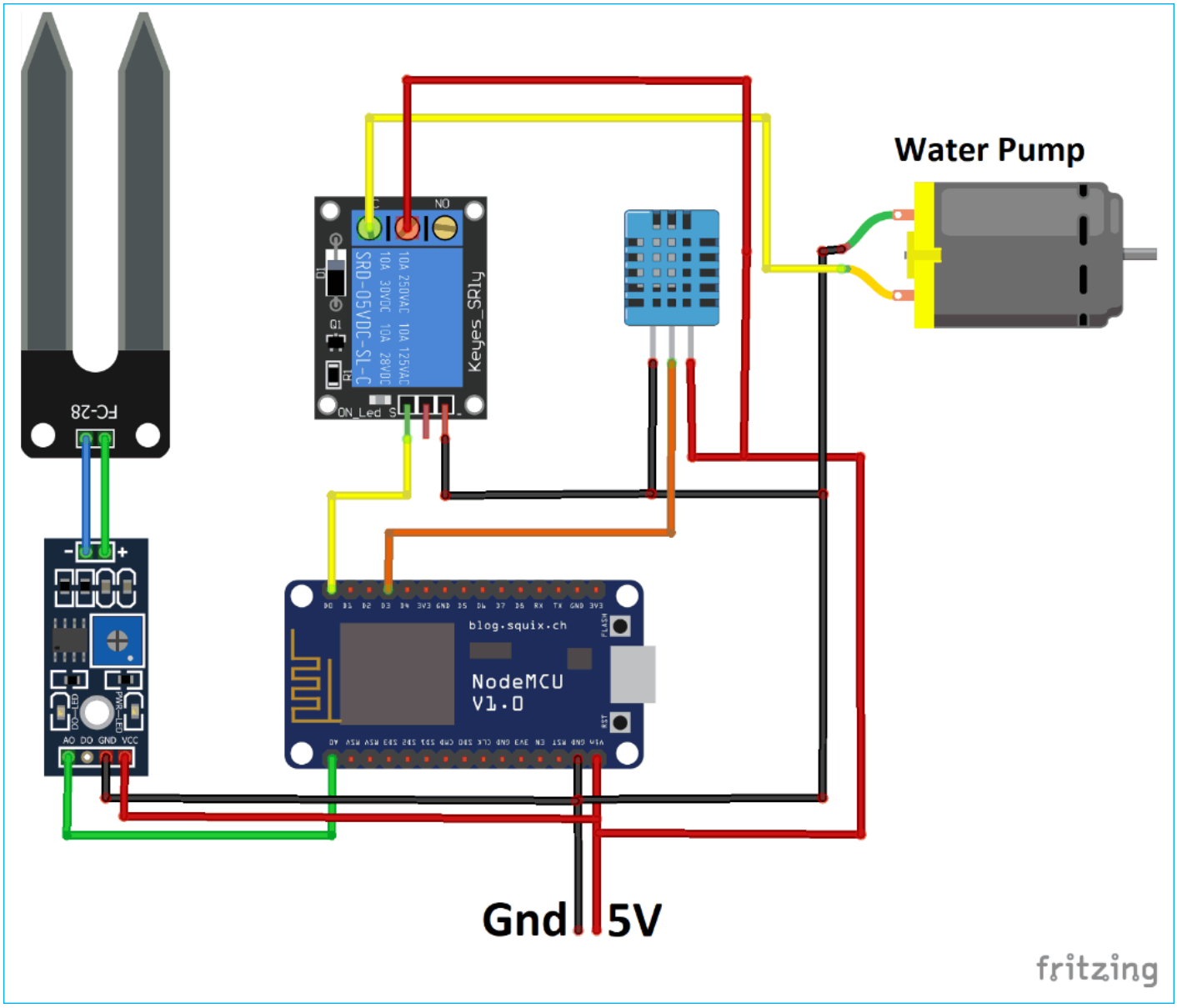
- benefit of using 16 ports (quasi-bidirectional)
- low current consumption with high current drive capability (sink current)
- Operating supply voltage is between 2.5 to 5.5 V

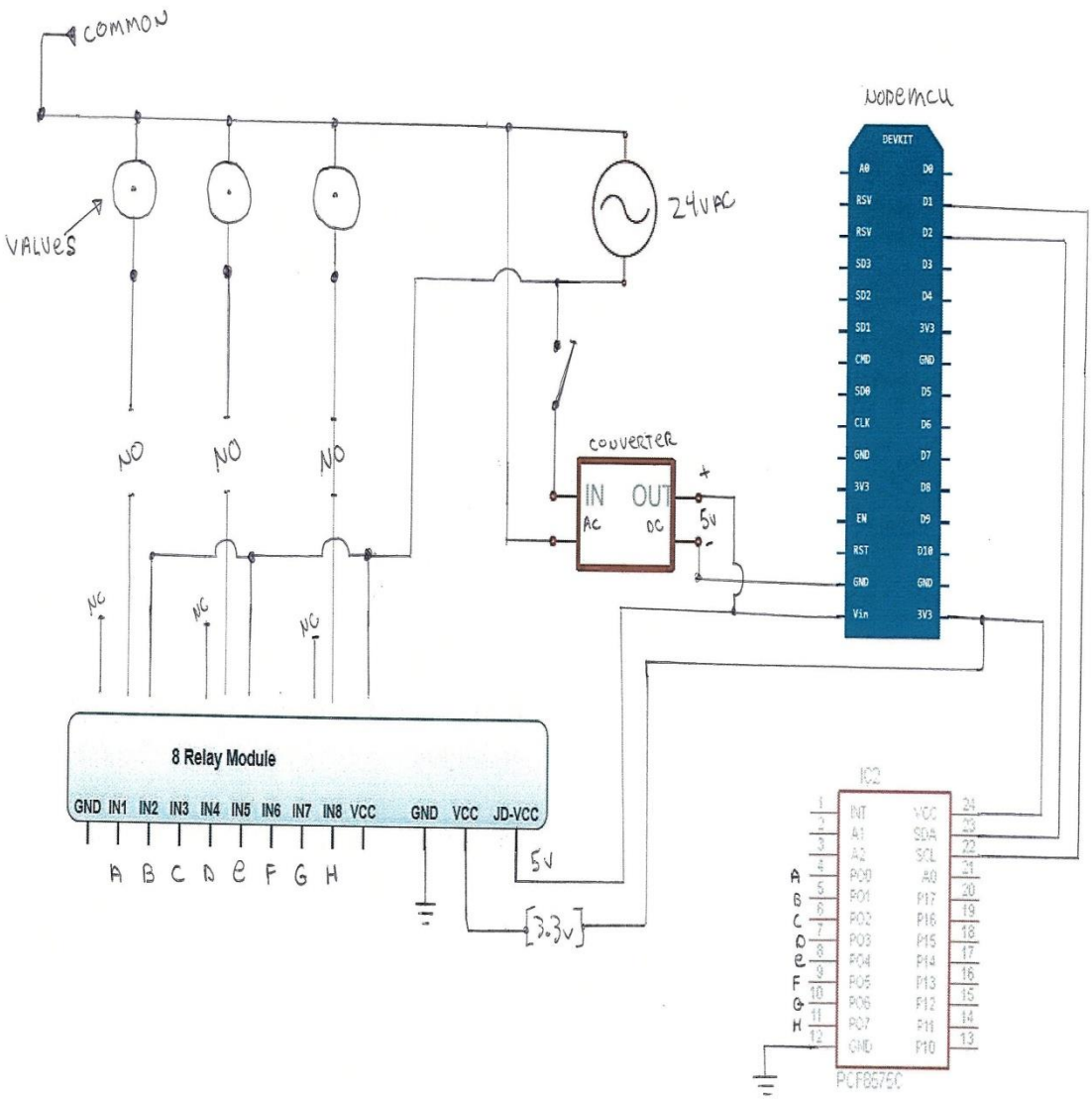
Relay Module

Once the MCU give the signal for a specific zone (valve), it will communicate (through I2C) and activate a port on the PCF8575 to sink current (port = 0V) from the relay pin and activate the relay valve (or zone). The relay will close the circuit and permit power (24vac) from the original power supply into the solenoid valves and since I have 8 valves, 8 relays are needed.

Circuit Diagram

Circuit diagram for this **IoT Smart Irrigation System** is given below:




COMMON
VALVES
24V AC
NODEMCU
DEVKIT
A0
RSV
RSV
SD3
SD2
SD1
CMD
SD0
CLK
GND
3V3
EN
RST
GND
Vin
D0
D1
D2
D3
D4
3V3
GND
D5
D6
D7
D8
D9
D10
GND
3V3
NO
NO
NO
NC
NC
NC
CONVERTER
IN
OUT
AC
DC
+
5V
-
8 Relay Module
GND IN1 IN2 IN3 IN4 IN5 IN6 IN7 IN8 VCC
GND VCC JD-VCC
A B C D E F G H
5V
3.3V
IC2
INT
A1
A2
P00
P01
P02
P03
P04
P05
P06
P07
GND
VCC
SDA
SCL
A0
P17
P16
P15
P14
P13
P12
P11
P10
A
B
C
D
E
F
G
H
PCF8575C

CODE

```
/*  Irrigation system
   Version 1.0
  PCF8575N & PCF8574N is 'reverse' logic in as much it SINKS current.
       so HIGH is OFF and LOW is ON.
       Turn OFF all pins by sending a high byte (1 bit per byte)
// OFF       B11111111,B11111111     // OFF for all 16 IO
// ON        B00000000,B00000000     // ON  for all 16 IO
*/
#include <Arduino.h>
#include <NodeMCU_Pinouts.h>
#include <Wire.h>
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <time.h>
#include <EEPROM.h>
const char* ssid     = "MySSID";
const char* password = "Password";
String host_name = "PoC";
int timezone = (4 * 60 * 60);              // For UTC -5.00 : -5 * 60 * 60 : -18000
const int EPOCH_1_1_2019 = 1546300800;     // For NTP - 1546300800 =
01/01/2019 @ 12:00am (UTC)
const char* ntpServer = "pool.ntp.org";
```

```
time_t now;

static unsigned long lastGetTimeClock;      // Counter for updating UTC time after a delay

static const int ZoneOthers = 0xFF ;

const long zoneTimerInterval = 350000UL;    // timer ~ 5min

unsigned long zoneTimerPrevious = 0;        // timer will store last time timer was updated

int zoneTimerState = 0;                     // timer

int EEaddress = 0;                          // EEprom address

bool flag_isAutoDay;

bool flag_isAutoDay_Executed;

bool flag_isAutoDay_notExecute = true;      // Reset the flag autoIrrigation to 'Execute' when odd day is coming

bool flag_isAutoHourReady = false;

String thisTime = "none";

String autoLastrunTime = "";


bool autoPilot = true;

int  autoPilotSetZone;

int  allZones_counter;

bool allZones_flag = false;

bool allZonesTimer_flag = false;

ESP8266WebServer server(80);

int address = 0x20; //0100000 (7bit)          //address is |0100|A0|A1|A2|
```

```
static const uint8_t ku8TWISuccess    = 0;      //I2C/TWI success (transaction was successful).

static const uint8_t ku8TWIDeviceNACK = 2;      //I2C/TWI device not present (address sent, NACK received).

static const uint8_t ku8TWIDataNACK   = 3;      //I2C/TWI data not received (data sent, NACK received).

static const uint8_t ku8TWIError      = 4;      //I2C/TWI other error.


static const uint8_t Zone_1 = 0b11111110 ;

static const uint8_t Zone_2 = 0b11111101 ;

static const uint8_t Zone_3 = 0b11111011 ;

static const uint8_t Zone_4 = 0b11110111 ;

static const uint8_t Zone_5 = 0b11101111 ;

static const uint8_t Zone_6 = 0b11011111 ;

static const uint8_t Zone_7 = 0b10111111 ;

static const uint8_t Zone_8 = 0b01111111 ;

static const uint8_t Zone_off = 0b11111111 ;


String htmlpage = "";         // Web page

// ================================ Subroutines


void Relay_off()

{

      Wire.beginTransmission(address);
```

```
        Wire.write(lowByte(Zone_off));
        Wire.write(lowByte(Zone_off));
        Wire.endTransmission();


  zoneTimerState = 0;               // disable the 'zoneTimerState - if condition' in the 'loop' function
        delay(2500);
}
void Execute_Zone(uint8_t zone)
{
  Wire.beginTransmission(address);     // Set the Relay for the Zone
        Wire.write(lowByte(zone));
        Wire.write(lowByte(zone));
        Wire.endTransmission();


  zoneTimerState = 1;               // 0 = off, 1 = ON // Enable the 'zoneTimerState' if condition in the 'loop' function
  zoneTimerPrevious = millis();       // start the countdown using millis to compare with 'ZoneTimerNow' in 'loop' function
}
void autoMode_onORoff(bool val)
{
  if(val)
  {
```

```
        autoPilot = true;
  }
  else
  {
    autoPilot = false;
  }
   EEaddress = 0;                          // EEprom address of autoPilotSetting
  EEPROM.write(EEaddress, autoPilot);          // Write to memory
  EEPROM.commit();                          // written to flash
  }
void webpageMain()
{
  htmlpage += "<!DOCTYPE html>";
  htmlpage += "<head><title>IoT Irrigation</title>";
  htmlpage += "<style>";
  htmlpage += "#header  {background-color:blue;      font-
family:Tahoma,Verdana,Serif,sans-serif; width:1024px; padding:5px;
color:white; text-align:center; }";
  htmlpage += "#section {background-color:#C2DEFF;   font-
family:Tahoma,Verdana,Serif,sans-serif; width:1024px; padding:5px;
color:blue;  font-size:12px;}";
  htmlpage += "#footer  {background-color:steelblue; font-
family:Tahoma,Verdana,Serif,sans-serif; width:1024px; padding:5px;
color:white; font-size:9px; clear:both;}";
  htmlpage += "</style></head>";
```

```
  htmlpage += "<script type=\"text/javascript\"> function reloadPage() {location.reload(true)} </script>";

  htmlpage += "<body>";

  htmlpage += "<div id=\"header\"><h1>Irrigation System 1.0</h1></div>";

  htmlpage += "<div id=\"section\"><h3>";

  htmlpage += "<form action=\"/\" method=\"POST\">";

  htmlpage += "<center>";

  htmlpage += "<input type=\"radio\" name=\"zone\" value=\"zone1\"> Zone-1";

  htmlpage += "<br><br>";

  htmlpage += "<input type=\"radio\" name=\"zone\" value=\"zone2\"> Zone-2";

  htmlpage += "<br><br>";

  htmlpage += "<input type=\"radio\" name=\"zone\" value=\"zone3\"> Zone-3";

  htmlpage += "<br><br>";

  htmlpage += "<input type=\"radio\" name=\"zone\" value=\"zone4\"> Zone-4";

  htmlpage += "<br><br>";

  htmlpage += "<input type=\"radio\" name=\"zone\" value=\"zone5\"> Zone-5";

  htmlpage += "<br><br>";

  htmlpage += "<input type=\"radio\" name=\"zone\" value=\"zone6\"> Zone-6";

  htmlpage += "<br><br>";
```

```
  htmlpage += "<input type=\"radio\" name=\"zone\" value=\"zone7\"> Zone-7";

  htmlpage += "<br><br>";

  htmlpage += "<input type=\"radio\" name=\"zone\" value=\"zone8\"> Zone-8";

  htmlpage += "<br><br>";

  htmlpage += "<input type=\"radio\" name=\"zone\" value=\"zoneX\"> Execute all zones";

  htmlpage += "<br><br>";

  htmlpage += "<input type=\"radio\" name=\"zone\" value=\"stop\">   Stop irrigation";

  htmlpage += "<br><br>";

  htmlpage += "<input type=\"radio\" name=\"zone\" value=\"auto-OR-man\">   Automatic or manual mode";

  htmlpage += "<br><br>";

}

void webpage_Execute()

{

         String strAutoPilotMode = "";

        if(autoPilot)

  {

    strAutoPilotMode = "automatic";

  }

  else
```

```
  {
    strAutoPilotMode  = "manual";
  }
   htmlpage = "";  // reset webpage to this form - else will have multiple pages...
  webpageMain();
  htmlpage += "<br>";
  htmlpage += "<input type=\"submit\">";
  htmlpage += "<br><br>";
  htmlpage += "<input type=\"button\" value=\"Refresh\" onclick=\"reloadPage()\" />";
  htmlpage += "</form>";
  htmlpage += "<br><br>";
  htmlpage += "Irrigation mode : " + strAutoPilotMode + "";
  htmlpage += "<br><br>";
  htmlpage += "Last execution time  " + autoLastrunTime + "";
  htmlpage += "<br><br>";
  htmlpage += "Last check UTC time  " + thisTime + "";
  htmlpage += "<br>";
  htmlpage += "</center>";
  htmlpage += "</h5>";
  htmlpage += "</div></body></html>";
server.send(200, "text/html", htmlpage);
```

```
}
void returnFail(String msg)
{
  server.sendHeader("Connection", "close");
  server.sendHeader("Access-Control-Allow-Origin", "*");
  server.send(500, "text/plain", msg + "\r\n");
}
void handleSubmit()
{
  if (!server.hasArg("zone")) return returnFail("BAD ARGS");
  String val_Zone = server.arg("zone");
 if (val_Zone == "zone1") {
          Relay_off();
          Execute_Zone(Zone_1);
    webpage_Execute();
  }
  else if (val_Zone == "zone2") {
          Relay_off();
          Execute_Zone(Zone_2);
    webpage_Execute();
  }
   else if (val_Zone == "zone3") {
          Relay_off();
```

```
        Execute_Zone(Zone_3);
  webpage_Execute();
}
else if (val_Zone == "zone4") {
        Relay_off();
        Execute_Zone(Zone_4);
  webpage_Execute();
}
 else if (val_Zone == "zone5") {
        Relay_off();
        Execute_Zone(Zone_5);
  webpage_Execute();
}
else if (val_Zone == "zone6") {
        Relay_off();
        Execute_Zone(Zone_6);
  webpage_Execute();
}
  else if (val_Zone == "zone7") {
        Relay_off();
        Execute_Zone(Zone_7);
  webpage_Execute();
}
```

```
  else if (val_Zone == "zone8") {
          Relay_off();
          Execute_Zone(Zone_8);
    webpage_Execute();
  }
 else if (val_Zone == "zoneX") {                    // Execute all zones
          Relay_off();
    allZones_flag = true;
    allZonesTimer_flag = true;
    allZones_counter = 1;                          // Execution for all zones will be
set to zone 1
    webpage_Execute();
  }
    else if (val_Zone == "stop") {
          Relay_off();
    allZones_flag = false;
    allZonesTimer_flag = false;
    webpage_Execute();
  }
  else if (val_Zone == "auto-OR-man") {                    // Automatic or manual
mode
    bool SetAutomode;
 if(autoPilot)
```

```
    {SetAutomode = false;}
   else
    {SetAutomode = true;}
       autoMode_onORoff(SetAutomode);
   webpage_Execute();
  }
  else {webpage_Execute(); }
}
void handleRoot()
{
  if (server.hasArg("zone"))
  {
   webpage_Execute();
        handleSubmit();
  }
  else {
   webpage_Execute();
  }
}
void returnOK()
{
  server.sendHeader("Connection", "close");
  server.sendHeader("Access-Control-Allow-Origin", "*");
```

```
  server.send(200, "text/plain", "OK\r\n");
}
void handleNotFound()
{
  String message = "File Not Found\n\n";
  message += "URI: ";
  message += server.uri();
  message += "\nMethod: ";
  message += (server.method() == HTTP_GET)?"GET":"POST";
  message += "\nArguments: ";
  message += server.args();
  message += "\n";
  for (uint8_t i=0; i<server.args(); i++){
    message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
  }
  server.send(404, "text/plain", message);
}
void updateClock()
{
  if ( (millis() - lastGetTimeClock) >= 900000UL)       // every 15min = 900000UL
- updates clock time every 30min = 1800000UL - update every hour =
3600000UL
  {
```

```
  time_t now;
  struct tm * timeNow;
  time(&now);
  timeNow = localtime(&now);

  int setDay  = timeNow->tm_mday;
  int setHour = timeNow->tm_hour;
  int setMin  = timeNow->tm_min;

  char snumDay[5];
  char snumHour[5];
  char snumMin[5];

  String thisDay  = itoa(setDay, snumDay, 10);     // use only to display in html page
  String thisHour = itoa(setHour, snumHour, 10);   // use only to display in html page
  String thisMin  = itoa(setMin, snumMin, 10);     // use only to display in html page

 thisTime = "DayOfWeek:" + thisDay + "   Time:" + thisHour + "-" + thisMin;

if(setDay % 2 == 0)                  // if day is even (ex.: 2, 4, 8, 10, etc.) disable auto pilot
```

```
   {
     flag_isAutoDay = false;               // autoPilot = false;
     flag_isAutoDay_notExecute = true;      // Reset the flag autoIrrigation to 'Execute' when odd day is coming
     flag_isAutoHourReady = false;
   }
   else                                   // day is odd (ex.: 5, 11, 15, 17, etc.) Set auto Pilot to 'true'
   {
     flag_isAutoDay = true;
     if(setHour == 6)
     {
       flag_isAutoHourReady = true;
     }
     else{flag_isAutoHourReady = false;}     // change flag value - before and after the 'authorize set time'
   }
   lastGetTimeClock = millis();              // Reset lastGetTimeClock counter
  }
   }
// ================================= End SubRoutines
void setup()
{
        Serial.begin(115200);              // Serial Window (debugging)
```

```
EEPROM.begin(32);
      Wire.begin();                  // I2C Two Wire initialisation

      // PCF8575 - Turn OFF all pins by sending a high byte (1 bit per byte)
      Wire.beginTransmission(address);
      Wire.write(lowByte(Zone_off));
      Wire.write(highByte(Zone_off));
      Wire.endTransmission();

      // ----- wifi
      WiFi.hostname(host_name);
 Serial.println();
 Serial.print("Connecting to ");
 Serial.println(ssid);
      WiFi.begin(ssid, password);
       while (WiFi.status() != WL_CONNECTED) {
      delay(1200);
      Serial.print(".");
 }
Serial.println("");
 Serial.println("WiFi connected");
 Serial.println("IP address: ");
 Serial.println(WiFi.localIP());
```

```
  Serial.println("hostname : ");
  Serial.println(WiFi.hostname());
        // ----- end of wifi
        // ----- Set Time with UTP
  configTime(timezone, 0, ntpServer);
  while (now < EPOCH_1_1_2019)
  {
    now = time(nullptr);
    delay(500);
    Serial.print("*");
  }
  // Read EEProm memory and set irrigation for automatic or manual mode
  EEaddress = 0;
  autoPilot = EEPROM.read(EEaddress);
 // Enable server mode
  server.on("/", handleRoot);
  server.onNotFound(handleNotFound);
        server.begin();                 //Start server
        Serial.println("HTTP server started");
}
void loop(void)
{
  updateClock();
```

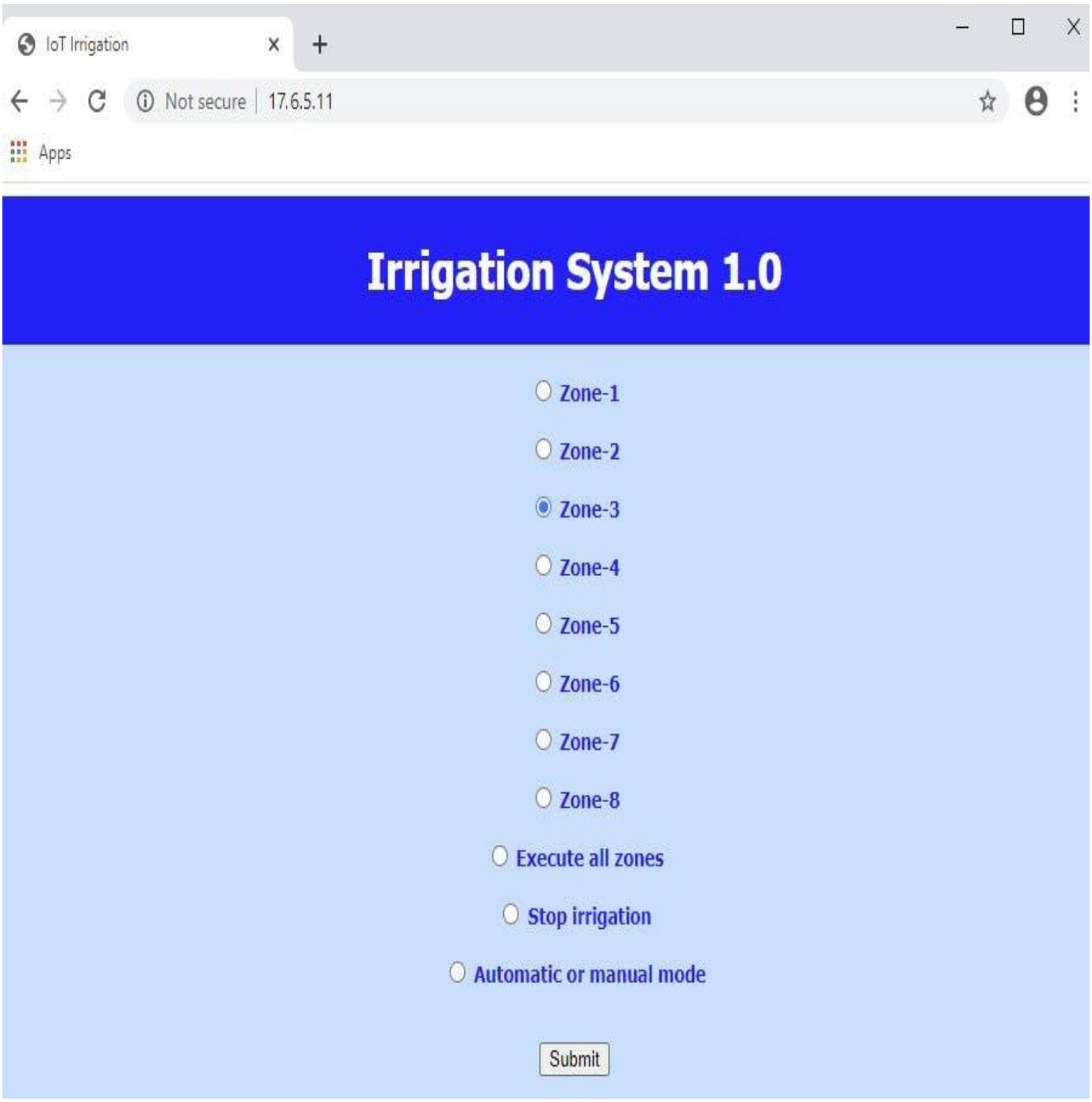
```
  server.handleClient();


  if(autoPilot)                        // Enable or Disable the 'Automatic irrigation'
    {
      if(flag_isAutoDay && flag_isAutoDay_notExecute &&
flag_isAutoHourReady)    // start irrigation system
      {
       flag_isAutoDay_notExecute = false;  // Prevent re-executing this 'if
statement'


       allZones_flag = true;              // Ready to Execute all zones (1 to 8)
       allZonesTimer_flag = true;         // Ready - Permit to re-enter the all
zones 'if statement' when zone X delay is finish
       allZones_counter = 1;              // Reset order (zone 1 to 8) to start at zone
1
          autoLastrunTime = thisTime;  // Record time when AutoPilot is ready
to execute all zones
      }
    } // end of if condition


  if(zoneTimerState == 1)                // if irrigation is running - validate time
left from timer
  {
    unsigned long ZoneTimerNow = millis();
if(ZoneTimerNow - zoneTimerPrevious >= zoneTimerInterval )
```

```
   {
      zoneTimerPrevious = ZoneTimerNow;
      Relay_off();
      allZonesTimer_flag = true;    // USED only by allZones_flag 'if condition' in void loop - Will wait until timer is finish
                                    // before executing another zone...
   }
 }
 if(allZones_flag == true && allZonesTimer_flag == true)
 {
   switch (allZones_counter)
   {
     case 1:
      allZones_counter++ ;             // increment counter for next zone to execute after timer
      allZonesTimer_flag = false;      // disable 'if condition' because timer is not finished and not starting another zone
      Execute_Zone(Zone_1);
      break;
      case 2:
      allZones_counter++ ;             // increment counter for next zone to execute after timer
      allZonesTimer_flag = false;      // disable 'if condition' because timer is not finished and not starting another zone
```

```
      Execute_Zone(Zone_2);
      break;
  case 3:
      allZones_counter++ ;            // increment counter for next zone to execute after timer
      allZonesTimer_flag = false;      // disable 'if condition' because timer is not finished and not starting another zone
      Execute_Zone(Zone_3);
      break;
case 4:
      allZones_counter++ ;            // increment counter for next zone to execute after timer
      allZonesTimer_flag = false;      // disable 'if condition' because timer is not finished and not starting another zone
      Execute_Zone(Zone_4);
      break;
  case 5:
      allZones_counter++ ;            // increment counter for next zone to execute after timer
      allZonesTimer_flag = false;      // disable 'if condition' because timer is not finished and not starting another zone
      Execute_Zone(Zone_5);
      break;
     case 6:
```

```
      allZones_counter++ ;            // increment counter for next zone to
execute after timer

      allZonesTimer_flag = false;      // disable 'if condition' because timer is
not finished and not starting another zone

      Execute_Zone(Zone_6);

      break;

  case 7:

      allZones_counter++ ;            // increment counter for next zone to
execute after timer

      allZonesTimer_flag = false;      // disable 'if condition' because timer is
not finished and not starting another zone

      Execute_Zone(Zone_7);

      break;

  case 8:

      allZones_counter++ ;            // increment counter for next zone to
execute after timer

      allZonesTimer_flag = false;      // disable 'if condition' because timer is
not finished and not starting another zone

      Execute_Zone(Zone_8);

        allZones_flag = false;          // All zones are executed.  Now disable the
'if condition - allZones_flag' from void loop

      break;

 }

  } // end for executing allzones

}
```

WEB PAGE


IoT Irrigation
Not secure | 17.6.5.11
Apps
Irrigation System 1.0
Zone-1
Zone-2
Zone-3
Zone-4
Zone-5
Zone-6
Zone-7
Zone-8
Execute all zones
Stop irrigation
Automatic or manual mode
Submit

Testing

The aim of the system testing process was to determine all defects in our project. The program was subjected to a set of test inputs and various observations were made and based on these observations it will be decided whether the program behaves as expected or not. Our Project went through three levels of testing

1.Unit Testing

2.integration Testing

3.Beta Testing

4.Hardware testing

1. Unit Testing

Unit testing is undertaken when a module has been created and successfully reviewed. In order to test a single module we need to provide a complete environment.
Unit testing was done on each and every module that is described under module description

- Test for Maps Module:
  This module is used for authentication of destinations
- Sharing with friends:
  In this section the user share location with friends ove social media.

2. Integration Testing

In this type of testing we test various integration of the project module by providing the input .The primary objective is to test the module interfaces in order to ensure that no errors are occurring when one module invokes the other module.

3. Beta Testing

In this test we have tested the complete developed application with all possible input and output.

4. Hardware Testing

After finishing a prototype, it is time to do some testing to make sure that the prototype behaves as expected.

The most relevant issues to test are:

- RF parameters
- Current consumption

4)Functional Testing :-

This is a type of Black Box testing that is based on the test cases based on the specifications of the software component under this test. Feeding them input and then examining the output for the given input test functions. In this type of testing internal program structure is not considered. Here the emphasis is laid on the output of a given input without caring the algorithms working inside that program.

Levels of Testing:-

Unit Testing :-

This is a method where the individual source code is tested to determine if it is fit to use. Here a unit is the smallest part of the application that can be tested individually. These are the test that is conducted by us while programming the software each individual component is tested before we can proceed further with the next phase of programming. There were bugs that were found out while unit testing the product.

Integration Testing:-

This type of testing is done while integrating the project components. The testing is basically done to check whether the components that were working well individually are working properly with others. It occurs after the unit testing. There are two types of Integration testing i.e. Top Down and Bottom up. Our application is Bottom up tested. Here what actually we have done is that we have tested the lower level modules like the Draw module, Screenshot module and the settings module. Here the functions for some low level module are tested after integration

System Testing:-

This is a type of test which is done on the whole product After completion of the project to know the merit and demerit of the product under different Android operating System .

How the software will perform when made run on the Froyo 2.2, Honeycomb, Gingerbread etc. and also we can know the security issue ,how much it is scalable in different condition.

As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing. The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together. System testing is a more limited type of testing; it seeks to detect defects within the system as a whole.

Testing the project:-

On performing all the above test such as Functional Testing, Structural Testing ,Unit Testing, Integration Testing . We found that our Application works fine. This Application is basically Developed on Android Froyo 2.2 Operating System. There are all most 18 Activity working under this Application And All Activity show no Error Under Prescribed Operating System . The Layout which are designed under this also work fine.

Implementation:-

Implementation is one of the important phases of project development. In this phase the project is actually implements in real environment after successful testing of the project from every aspect.

Implementation of the project

The project is supported by all the android devices with android O.S. (version 2.2 or above). The application is implements on devices using .apk file. On successfully installation of .apk file the application can be used and all features can be accessed.

Installing/Implementing Application

1. Download and install the Apps Installer application from the Android Market if you do not have one inbuilt.
2. Open the application and click on install.
3. When Installation is completed launch icon appears in applications and application can be opened from the

icon and features can be accessed.

Technical and Managerial lessons learnt

Working on the project was a great experience and helped to learn a lot trough out new technologies were learnt during development and it was an opportunity to implement what has been learnt throughout the degree.
Working as a team in professional environment under the guidance of mentors taught managerial skills, how to deal with things, how to plan things, how to work and deal with different people.

Technical Lessons :-

□ Working on Android Application development was a great experienced and got to learn a new technology which in great demand now a days.
□ JAVA Programming was brushed up during the internship.
□ Development was carried out on Eclipse.
□ Dalvik Debug Monitor Server (DDMS)
□ LogCat

Managerial Lesson:-

The mentors guidance, team work and professionalism helped to develop managerial skills. Time Management:-Ever phase was given a time line and was carried out in the given time. Proper Planning:-Development compromises of great planning before implementing, throughout the phases proper planning was carried out before getting into the phase

directly.

Handling Queries:-Queries were put on during presentations and seminars to judge the individual performance, this helped to learn how to tackle queries

Solving Problems:-Various problems were phase during development and were overcome successfully with group discussion and mentors advice.

Professionalism:-Working in professional environment helped to develop a professional behaviour and deal with different individuals better and learnt to work in industry.

Team Work:-Project was carried out in a team, developed leadership skill, team spirit, motivation skills.

Building Confidence:-Giving presentation in front of panel and other teams helped building up the confidence to address and face people.

Communication:-Presentations and seminars conducted helped to improve communications and overcome hesitation.

Effective Presentation:-After completion of a phase successfully group presentations were

LIMITATIONS AND FUTURE SCOPE

Limitations of the System:

- **Smart watering system is a bit expensive. Depending on the size of your property, you will need more systems. Of course saving on water bills will lead to less cost.**
- **If you want to use this system for lawn watering, it's better to fix it under the ground before planting. Because some parts of the lawn will be harm because of holing**

Future Scope of the System :

● Collect real-time water usage data through remote sensing technologies that allow you to make smarter decisions about water usage.

● Trigger actions based on sensor data to switch water on and off, depending on irrigation needs and level of resources.

● Use insights gathered from data to prevent misuse and underutilization of water resources.

● Quickly detect leaks and faults in pipelines to address issues immediately and r educe water waste.

● Stay on top of changing conditions across agricultural environments by tracking temperature, rainfall, humidity, and wind with smart sensors.

● Remotely measure and monitor water moisture levels in soil to ensure that crops are getting optimal water resources.

● Automatically trigger sprinkler systems to address low moisture levels in soil to prevent crop damage or loss.

● Capture valuable data to indicate trends and make forecasts based on a variety of conditions to meet crop demands.

Conclusion:

The smart irrigation system implemented is cost effective for optimizing water resources for agricultural production. The proposed system can be used to switch on/off the water sprinkler depending on the soil moisture levels thereby making the process simpler to use. Through this project it can be concluded that there can be considerable development in irrigation with those of IOT and automation. Thus this system is a solution to the problems faced in the existing process of irrigation.

BIBLIOGRAPHY

Websites:

https://ijarcce.com/upload/2017/september-17/IJARCCE%2039.pdf
https://www.ripublication.com/ijems_spl/ijemsv8n1_08.pdf
https://www.researchgate.net/publication/313779760_IOT_based_Smart_Irrigation_System
https://www.ijitee.org/wp-content/uploads/papers/v8i12S/L10541081 2S19.pdf

Books:

1.S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, "Cloud computing - The business perspective," Decis. Support Syst., 2011.

2. F. T. -, "Smart Agriculture Based on Cloud Computing and IOT," J. Converg. Inf. Technol., 2013.

3. M. S. Mekala and P. Viswanathan, "A Survey: Smart agriculture IoT with cloud computing," in 2017 International Conference on Microelectronic Devices, Circuits and Systems, ICMDCS 2017, 2017.

4. S. A. Arduino, "What is Arduino?," Arduino Doc., 2015