Dr. AMBEDKAR INSTITUTE OF TECHNOLOGY

(An Autonomous Institute, Affiliated to VTU, Belagavi, Accredited by NAAC with 'A' Grade)

Near Jnana Bharathi Campus, Bengaluru – 560056

A PROJECT REPORT
ON

"SMART IRRIGATION MODEL USING IoT"

Submitted in partial fulfilment of the requirement for the award of the Degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by

KUNAL JAYSWAL	**1DA17CS067**
NABIN KHADKA	**1DA17CS087**
POOJIT CHOWDRY	**1DA17CS110**
PRAGIK TIMSINA	**1DA17CS111**

Under the Guidance of

Mrs. Lavanya Santhosh
Assistant Professor,
Dept. of CSE, Dr. AIT

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
2020-2021

Dr. AMBEDKAR INSTITUTE OF TECHNOLOGY

(An Autonomous Institute, Affiliated to VTU, Belagavi, Accredited by NAAC with 'A' Grade)

Near Jnana Bharathi Campus, Bengaluru – 560056

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project work, entitled "**SMART IRRIGATION SYSTEM"** is a bonafide work carried out by **KUNAL JAYSWAL [1DA17CS067], NABIN KHADKA[1DA17CS087],POOJIT CHOWDRY [1DA17CS110], PRAGIK TIMSINA [1DA17CS110]** in the partial fulfilment for the award of Degree in Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belgaum during the year 2020-21. It is certified that all corrections/suggestions indicated for the internal assessment have been incorporated in the report deposited in the departmental library. This project report has been approved as it satisfies the academic requirements in the respect of Project prescribed for Bachelor of Engineering Degree.

INTERNAL GUIDE	HOD, Dept. of CSE	PRINCIPAL
Mrs. Lavanya Santhosh	**Dr. Siddaraju**	**Dr. MEENAKSHI M**
Assistant Professor, Dept of CSE.	HOD, Dept of CSE.	Principal
Dr. AIT	Dr. AIT	Dr. AIT

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of this project would be incomplete without the mention of the people who made it possible, without whose constant guidance and encouragement our efforts would have gone in vain.

We consider ourselves privileged to express our gratitude and respect towards all those who guided us through the completion of this Project work, "**SMART IRRIGATION MODEL USING IOT**".

We consider ourselves proud to be part of **Dr. Ambedkar Institute of Technology**, the institute which stood by us throughout our academic endeavour.

We would like to express our gratitude to **Dr. MEENAKSHI M , Principal, Dr. A.I.T** for providing us with a congenial environment to work in.

We would like to express our gratitude to **Dr. Siddaraju, Professor and Head, Dept. of C.S.E, Dr. A.I.T,** for giving us the support, encouragement and laboratory facilities.

As a token of gratitude, we would like to acknowledge our **Project Guide, Mrs. Lavanya Santhosh, Assistant Professor, Dept. of C.S.E, Dr. A.I.T** for her unlimited support and encouragement provided throughout the process.

We also express our gratitude and sincere thanks to all the teaching and nonteaching **Staff of C.S.E. Department.** Finally, yet importantly, we would like to express our heartfelt thanks to our beloved **Parents** for their blessing and our **Friends** for their help and wishes.

Kunal Jayswal

Nabin Khadka

Poojit Chowdry

Pragik Timsina

ABSTRACT

In the present age Internet of things (IoT) has entered a golden era of rapid growth. The Internet of things is a concept that aims to extend the benefits of the regular Internet—constant connectivity, remote control ability, data sharing, and soon—to goods in the physical world. In this project we summarize the importance of healthy irrigation practices with the help of IOT controlled systems. With advanced farming techniques backed by today's advanced scientific methodologies, the frequency of crop production has increased and with that we have over used the resources required for production. The Smart Model would assist in increasing the production of crops and promote sustainable resource utilization which in turn optimizes the timing and volume of water distribution in the fields. The proposed systems are flexible in nature that would enable us to either increase or decrease the scale for which it would be applied hence targeting both the casual backyard farmers and professional farmers. Our proposed Model is simple enough to be used by a farmer with minimal technical knowledge.

TABLE OF CONTENTS

TABLE OF CONTENTS *i*

LIST OF FIGURES *iv*

1. INTRODUCTION *1*

1.1 OVERVIEW 2

1.2 EXISTING SYSTEM 3

1.2.1 DISADVANTAGES 3

1.3 PROPOSED SYSTEM 3

1.3.1 ADVANTAGES 4

1.3.2 BLOCK DIAGRAM OF PROPOSED SYSTEM 4

1.4 MOTIVATION 4

1.5 PROJECT OBJECTIVE 5

1.6 PROBLEM STATEMENT 5

1.7 SCOPE OF WORK 6

2. LITERATURE SURVEY *7*

3. SYSTEM REQUIREMENTS SPECIFICATION *15*

3.1 PURPOSE 15

3.2 SCOPE 15

3.3 SOFTWARE REQUIREMENTS 16

3.3.1 ARDUINO IDE 16

3.3.2 EMBEDDED C 17

3.4 HARDWARE REQUIREMENTS: 18

3.4.1 Temperature Sensor (LM35): 18

3.4.2 Humidity Sensor 18

3.4.3 Soil Moisture Sensor 19

3.4.4 DC MOTOR/MOTOR DRIVER 20

3.4.5 Arduino 22

3.4.6 LCD Panel 26

3.4.7 Water Pump 27

4. DESIGN ***29***

4.1 SYSTEM ARCHITECTURE 29

4.1.1 Power Source 29

4.1.2 Storage 29

4.1.3 GPIO (General Purpose Input Output) 29

4.1.4 Arduino Communication 30

4.2 RF Module (TRANSMITTER AND RECEIVER) 31

4.3 Sequence diagram 31

4.4 Activity diagram 32

5. TESTING ***33***

5.1 UNIT TESTING 33

5.2 INTEGRATION TESTING 34

5.3 REGRESSION TESTING 35

5.4 BLACK BOX TESTING 35

5.4.1 Testcases 35

5.5 WHITE BOX TESTING 38

5.6 SNAPSHOTS 38

6. IMPLEMENTATION ***40***

6.1 SYSTEM IMPLEMENTATION 40

6.2 HARDWARE IMPLEMENTATION 40

6.2.1 Arduino Nano 40

6.2.2 Jump Wire 41

6.2.3 Arduino Controller IDE 42

6.2.4 ARM Microcontroller 42

6.2.5 Universal Serial Bus(USB) 43

6.2.6 Breadboard 43

6.2.7 Wi-Fi Module 44

6.3 SOFTWARE IMPLEMENTATION **45**

6.3.1 Arduino Programming 45

6.4 CODE and UI **46**

6.4.1 Arduino Code 46

6.4.2 Application UI 50

CONCLUSION ***51***

FUTURE ENHANCEMENT ***52***

BIBLIOGRAPHY ***53***

LIST OF FIGURES

FIGURE 1.1 AVERAGE WATER REQUIRED ON 1 ACRE OF LAND IN % 2
FIGURE 1.2 BLOCK DIAGRAM OF THE PROPOSED MODEL 4
FIGURE 2.1 SYSTEM IMPLEMENTATION FOR REFERENCE PAPER [2] 8
FIGURE 2.2 SYSTEM IMPLEMENTATION FOR REFERENCE PAPER [3] 9
FIGURE 2.3 SYSTEM IMPLEMENTATION FOR REFERENCE PAPER [4] 10
FIGURE 2.4 BLOCK DIAGRAM FOR REFERENCE PAPER [5] 12
FIGURE 2.5 SYSTEM IMPLEMENTATION FOR REFERENCE PAPER [6] 12
FIGURE 2.6 SYSTEM IMPLEMENTATION FOR REFERENCE PAPER [7] 14
FIGURE 2.8 SYSTEM IMPLEMENTATION FOR REFERENCE PAPER[8] 14
FIGURE 3.1 LM35 DZ TEMPERATURE SENSOR 18
FIGURE 3.2 TYPES OF HUMIDITY SENSORS 19
FIGURE 3.3 DHT11 DIGITAL HUMIDITY TEMPERATURE SENSOR MODULE 19
FIGURE 3.4 SOIL MOISTURE SENSOR 20
FIGURE 3.5 DC MOTOR DRIVE 21
FIGURE 3.6 ARDUINO MICROCONTROLLER 23
FIGURE 3.7 2X16 LCD DISPLAY 27
FIGURE 3.8 WATER PUMP 28
FIGURE 4.1 SYSTEM ARCHITECTURE OF PROPOSED SYSTEM 29
FIGURE 4.2 ARDUINO NANO PINOUT 30
FIGURE 4.3 RF-TRANSMITTER AND RECEIVER 31
FIGURE 4.4 SEQUENCE DIAGRAM FOR USER VIA WEB APPLICATION 32
FIGURE 4.5 ACTIVITY DIAGRAM OF PROPOSED MODEL 32
FIGURE 5.1 DHT UNIT TESTING 33
FIGURE 5.2 SOIL MOISTURE UNIT TESTING 33
FIGURE 5.3 WATER PUMP UNIT TESTING 34
FIGURE 5.4 MICROCONTROLLER AND 34
FIGURE 5.5 INTEGRATION TESTING 34
FIGURE 5.6 RESULT OF TESTCASE 1 36
FIGURE 5.7 RESULT OF TESTCASE 2 36
FIGURE 5.8RESULT OF TESTCASE 3 37

FIGURE 5.9 RESULT OF TESTCASE 4	37
FIGURE 5.10 SYSTEM IMPLEMENTATION SNAPSHOT I	38
FIGURE 5.11 SYSTEM IMPLEMENTATION SNAPSHOT II	39
FIGURE 6.1 ARDUINO NANO PINOUT DIAGRAM	41
FIGURE 6.2: JUMP WIRE TYPES	42
FIGURE 6.3: ARDUINO CONTROLLER IDE	42
FIGURE 6.4 ARM MICROCONTROLLER	43
FIGURE 6.5:USB CABLE	43
FIGURE 6.6: BREAD BOARD	44
FIGURE 6.7:ESP-12E:ESP8266	44
FIGURE 6.8:ESP-12E:ESP8266 PIN DESCRIPTION	44
FIGURE 6.9: ARDUINO PROGRAMMING EXAMPLE	45
FIGURE 6.10 APPLICATION UI	50

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

The need to design a smart irrigation system is to replace the monotonous job of checking the field frequently for irrigation status which includes over irrigation, water flow in canals and under irrigation. In a small field this can be manageable. But for large fields this would be very tedious as they have to keep on travelling around their field for frequent inspection. The farmers need to irrigate the field based on their gut feeling which isn't always sunshine and rainbows. Sometimes they under irrigate and have to repeat the irrigation process all over again and sometimes they over irrigate and can do nothing but hope for the best.

About 16% of Indian Economy depends on agriculture. The vast majority of industries are agriculture based and get their raw materials from the agricultural industry which signifies that agricultural production plays a vital role in Indian economy. Considering all these factors, a smart sensor-based system for water irrigation is a must and can be optimized depending upon the planted crops as Indian farmers plant different types of crops throughout the year.

This irrigation model hopes to eradicate this exact problem for the farmer by providing accurate real time data of the fields in their mobile phone. If any problem is detected in the field, the farmer will be notified through an alert on the phone. If this smart irrigation system fails in any point of time and encounters a problem with valves (opening and closing), the farmer can override the system through his phone and gain instant control of the situation.

The water usage at the early stages of a plant's life is high and relatively slows down throughout the plant's life cycle. If the sprout is under-irrigated, it not only affects the growth and development of the crop but also decreases the soil quality. If the sprout is over-irrigated then it will loosen the soil around the roots and cause the sprout to be malnourished. Similar to the sprouts, in later stages of plant's life, under irrigation would not provide enough water for plant's development in the reproduction stage which would result in irregular to no yields and over irrigation tends to sweep away essential nutrients from the soil and causing the plant to unroot as the loose soil would mean that the plant cannot hold it's weight by itself. Hence, adequate amount of water needs to be supplied to the crop for maximum yields and increased overall production.

The following pie chart shows the average percentage of water needed for cultivation of a crop on one acre of land. The proposed "Smart Irrigation System" is compared against the traditional "Flood Irrigation System" and "Drip Irrigation System". From the chart presented below, we can conclude that the proposed smart irrigation system outperforms the already existing system by using less amount of required.

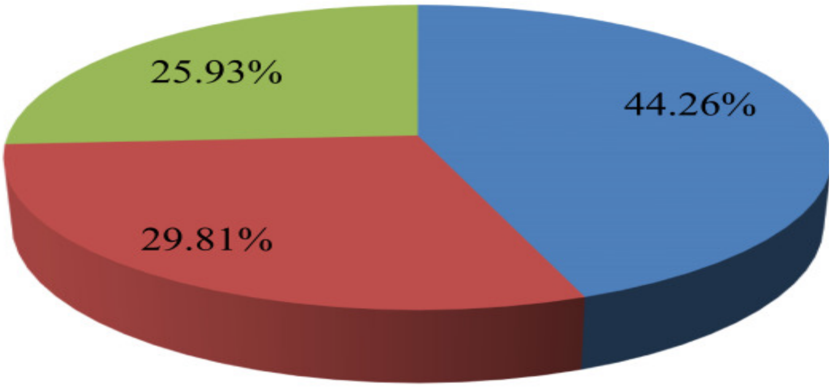

Figure 1.1 Average Water Required on 1 acre of land in %

1.1 OVERVIEW

In this project with an "Arduino Uno" different input and output is interfaced. In input section there is soil moisture sensor, humidity sensor and temperature sensor. In processing section an "Arduino Uno" is used and is equipped with WI-FI module. And on the output, there is a web application and a LCD screen in order to see the data in real time. The soil moisture sensor is placed inside the ground at the user required depth as different plants have different moisture requirement. The humidity sensor and the temperature sensor are placed out in the environment. The data collected from these sensors is fed into the "Arduino Uno" microcontroller. The data is processed and the output is projected on the designed web application in the user's device of choice or on the LCD screen which is directly connected to the microcontroller.

1.2 EXISTING SYSTEM

Some of the existing system intended for lawn maintenance or casual backyard farming are made by the user themselves. These systems are personalized for each user need and often created by users themselves which is not very cost efficient. For professional farmer though, there are no existing systems in place and farmers water their field based on their farming experience and gut feeling. The field is irrigated by seeing the color change in the leaf of a plant or judging the dryness of the soil with naked eyes or checking the weather. Conventional farmers plant crops depending upon the climate cycle. The canal doors are opened and water is let into the field. There is no definitive time allocation for how long the water is to be supplied and level of moisture in the soil to be maintained while irrigating. The existing method is unscientific and entirely depends on the farmer’s experience and level of expertise on the subject matter.

1.2.1 DISADVANTAGES

Some of the disadvantages of the existing irrigation system are:

- Less Resource Utilization.
- More tedious as the field and canals need constant verification.
- Harvest is inconsistent.
- Unpredictable climate conditions either over irrigate or under irrigate the fields.

1.3 PROPOSED SYSTEM

The proposed smart system provides accurate information of the irrigation status with the help of sensors. The input is collected from different sensors placed at different location. Humidity Sensor is used to collect information about possibility of rain at the local level. Soil Moisture Sensor collects information about the moisture level in the soil in real time. Temperature sensor collects information about the temperature of the area it is placed in. Hence, it can be placed outside the soil to measure current environment temperature or inside the soil to measure current soil temperature. These collected inputs are then sent to the microcontroller which processes this information and generates outputs. The outputs are projected to the user preferred device using web application

1.3.1 ADVANTAGES

The advantages of smart irrigation system is discussed below:

- Cost effective and cheap to manufacture.
- More accurate data reading.
- Resource efficient.
- Real time data monitoring.
- User Friendliness.

1.3.2 BLOCK DIAGRAM OF PROPOSED SYSTEM

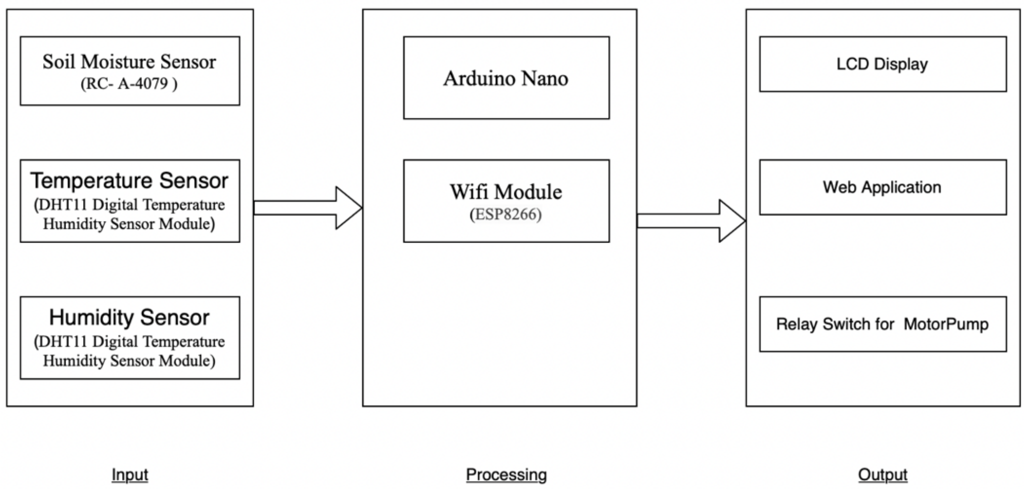

Figure 1.2 Block Diagram of the proposed model

1.4 MOTIVATION

In today's IT world, smart systems are being popular due to easy, flexible means of monitoring and controlling the system according to user comfort and needs. The challenging part lies in simplicity of construction and cost of installing them in user specified areas and varies with increasing number of services to be monitored and controlled. This project named “Smart Irrigation System” is an idea of irrigating fields using IOT Technology with the integration of an web application for easier access to real time data.

The popularity of smart system has been increasing greatly in recent years due to considerable affordability and simplicity through Smartphone and tablet connectivity. Devices may be connected through a home network to allow control by a personal computer, and may allow remote access from the internet. The integration of input sensors and output devices enabled by a microcontroller can communicate with one another which results in convenience, energy efficiency, and safety benefits.

This project forwards the design of Smart Irrigation System using Arduino Nano, a miniaturized computer. Arduino Nano provides the features of a mini computer, additional with its GPIO pins where other components and devices can be connected. GPIO registers of Arduino Nano are used for both the input and output purposes. We need to design a power strip that can be easily connected to GPIO Pins of the Arduino Nano. The sensors are connected as the input on the GPIO ports of Arduino Nano along with the power strip and their status is passed to the Arduino Nano. The real time data of the sensors can be viewed when connected to a network via a web application.

1.5 PROJECT OBJECTIVE

The main objectives of our project are as follows:

- To remotely obtain real time data of the field.
- To monitor the soil moisture level at any given time.
- To save time and utilize the available resources efficiently.

1.6 PROBLEM STATEMENT

Whether you are a terrace farmer or a professional farmer, providing constant attention while irrigating in order to avoid over-irrigation or under-irrigation to the plants you've grown is always a tedious job. With the terrace farmers focusing more on their main job for their livelihood, they don't give much attention to the plants they have planted which results in poor yield or often becomes the waste of their effort. Also due to the change in global temperatures, the global climate has become unexpected and more severe. The weather pattern upon which the professional farmers have been depended for centuries in order to irrigate their fields has been another matter of concern to the farmers.

The contemporary perception of water is that of a free renewable resource that can be used in abundance. But due to increasing summer temperature it has become scarce at the time of need. In addition to the excess cost of irrigating the fields, labor is becoming more and more expensive. As a result, if no effort is invested in optimizing these resources, there will be more money involved in the same process. Technology is probably a solution to reduce costs and prevent loss of resource, this project can be a strong way to tackle such a situation.

1.7 SCOPE OF WORK

Day by day, the field of electronics is booming and have caused great impact on human beings. The project which is to be implemented is an automated irrigation method and has a huge scope for future development. The project can be extended to greenhouses where manual supervision is needed constantly. This project work also presents solution to this problem, having to provide constant attention during irrigation, through real time data projection via the web application that is compatible to the user's preferred device of choice. Approach here is to get real time data collected by the sensors and display it in the web application and take automatic measures to ensure proper water supply to the plants. The principle can be extended to create fully automated gardens and farmlands. Combined with the principle of rain water harvesting, it could lead to huge water savings if applied in the right manner. In agricultural lands with severe shortage of rainfall, this model can be successfully applied to achieve great results with most types of soil.

CHAPTER 2

LITERATURE SURVEY

2. LITERATURE SURVEY

[1] P. K. and D. S. Rathi, "An IoT Based Smart Irrigation System," ***International Journal of Scientific & Engineering Research,*** **vol. 8, no. 5, pp. 44-51, May 2017.**

In this paper, the proposed IoT based irrigation system aims to utilize the features of embedded system to make agriculture simple. Having sensors connected with controller, the system reads the soil moisture, temperature and electrical conductivity of the soil and then the sensed data are processed in the controller. The microcontroller is the decision maker of this system. It checks for moisture value and the temperature. initially the threshold moisture and temperature value must be defined. When the sensed moisture value goes above the threshold value, the controller checks for the temperature. Only if the sensed temperature value is higher than the threshold value, irrigation is done and the user is acknowledged. This is because all crops can withstand in the dry soil moisture condition if the temperature is moderate. This would conserve the water used for irrigation. Sending SMS to the user about the field enables the user to remotely monitor the agriculture area. The SMS include the warning and suggestion to the affected system.

[2] M. D. Chaware, M. M. Panse, M. A. Raut and M. A. Koparkar, "Sensor Based Automated Irrigation System," ***International Journal of Engineering Research & Technology,*** **vol. 4, no. 05, pp. 33-37, May 2015.**

In this paper, the sensors are mounted at a depth of 1 foot near the root zone of the bell pepper plants in the farm. The supply is connected to a 5V battery. The moisture sensor after sensing the output provides the analog output in the range of 3.3 to 5V. These measured values are then fed to the controller in the form of voltages. An average threshold value will be calculated by the help of conduction charts and thus the water requirement will be defined. This sensed value is averaged and compared with the predefined average value whose results will determine the further course of action.

Further a message will be sent to the user via the GSM modem wherein the time of sensing, water status and necessity of water would be mentioned with the starting of the motor. Placing of water level detectors in the water body like tanks, wells etc. will be done with indications of high, medium and low values. If the value is below medium level, the pump will not start and indicate a necessity of water to the user through a text message on his cell phone. This will thus warn the user to arrange a separate water source for watering before the plant wilts within 5 hours which is its stress sustainable limit beyond which it will start wilting.

The water supply will be provided at night as the rate of evapotranspiration is the lowest at that time i.e. between 10p.m-5a.m. Thus this sensing will be monitored after every 5 hours. This monitoring will be done through RTC which will store the given dates and time of the last sensed output and previous data also. It thus gives us a chance to monitor these values over a period of certain days depending on plant growth and thus provide the requisite amount of water to the plants to prevent stress. The watering will stop after the plant is adequately watered and thus ensure a maximum crop yield with meticulous use of water resource and preserving soil nutrients.

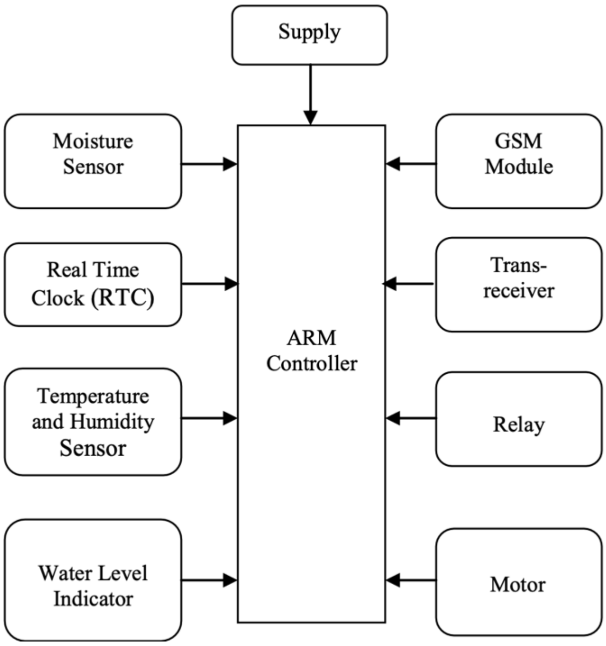

Figure 2.1 System Implementation for Reference Paper [2]

[3] V. L. Akubattin, A. P. Bansode, T. Ambre, A. Kachroo and P. P. SaiPrasad, "Smart Irrigation System," *International Journal of Scientific Research in Science and Technology,* vol. 2, no. 5, pp. 343-345, October 2016.

The paper as the name indicates, “Smart Irrigation System” provides an attractive user interface. This paper gives the idea to maintain the soil moisture content and temperature in a farming area and the user can control Sprinkler using Android phone/tab. This paper is based on Android and raspberry pi platform both of which are Free Open Source Software. So the overall implementation cost is cheap and it is an orderable by a common person. Looking at the current scenario we have chosen Android platform so that most of the people can get benefits. The design consists of Android App by which user can interact with the android phone and send control signal to the raspberry pi which will control sensors also raspberry pi monitor the environment. Thus user can control their farm from remote location by using Android mobile.

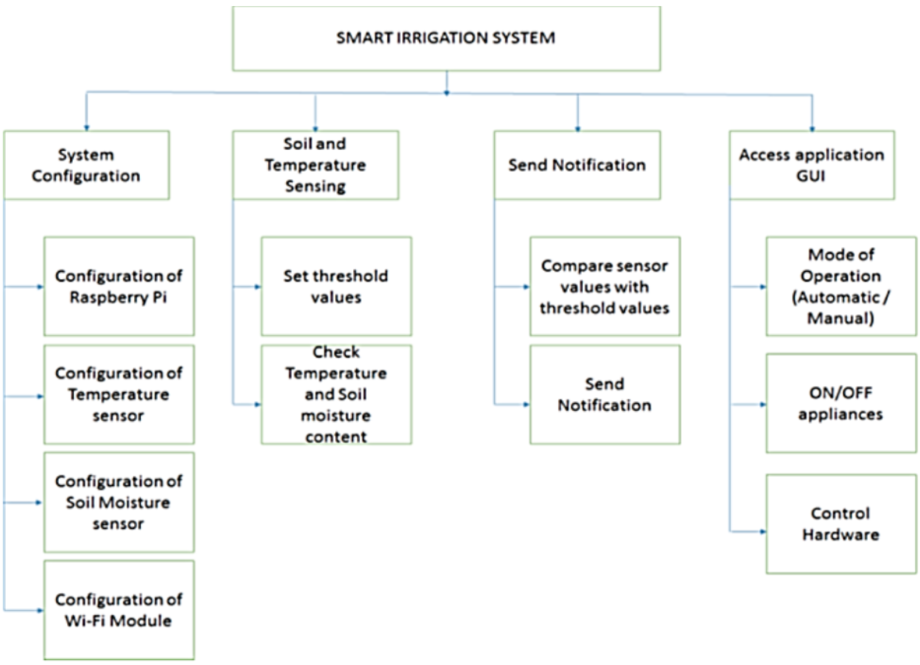

Figure 2.2 System Implementation for Reference Paper [3]

[4] S. Darshna, T. Sangavi, S. Mohan, A. Soundharya and S. Desikan, "Smart Irrigation System," *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE),* vol. 10, no. 3, pp. 32-26, 2015.

This prototype monitors the amount of soil moisture and temperature. A predefined range of soil moisture and temperature is set, and can be varied with soil type or crop type. In case the moisture or temperature of the soil deviates from the specified range, the watering system is turned on/off. In case of dry soil and high soil temperature, it will activate the irrigation system, pumping water for watering the plants. The block diagram of smart irrigation system is represented in Fig 2.3. It consists of a microcontroller (ATmega328) which is the brain of the system. Both the moisture and temperature sensors are connected to the input pins of the controller. The water pump and the servo motor are coupled with the output pins. If the sensors depart from the predefined range, the controller turns on the pump. The servo motor is used to control the angular position of the pipe, which ensures equal distribution of water to the soil. An LED indicator indicates the status of the pump. This system can be implemented on a large scale for farming purposes, which can further prove to be more advantageous. Owing to prevailing conditions and water shortages, the optimum irrigation schedules should be determined especially in farms to conserve water.

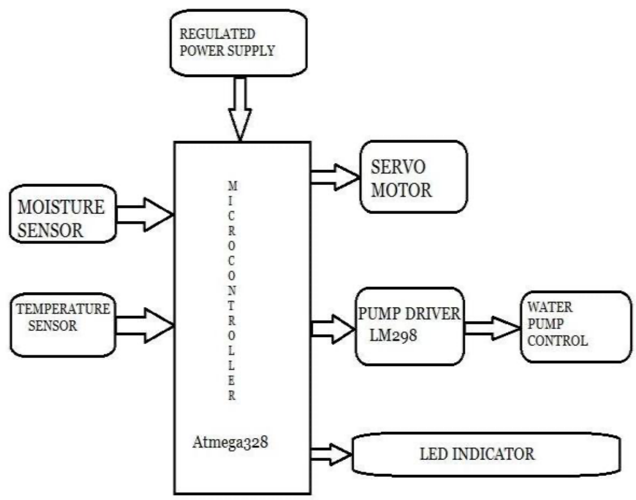

Figure 2.3 System Implementation for Reference Paper [4]

[5] G. Parameshwaran and K. Sivaprasath, "Arduino Based Smart Drip Irrigation System Using Internet of Things," ***International Journal of Engineering Science and Computing,*** **vol. 6, no. 5, pp. 5518-5521, May 2016.**

From the paper, Fig 4. shows the block diagram of smart irrigation system with IoT. Farmers start to utilise various monitoring and controlled system in order to increase the yield with help of automation of agricultural parameters like temperature, humidity, soil moisture, carbon dioxide, light detection, soil pH, etc. are monitored and controlled which can help the farmers to improve the yield. There are different sensors are used to controlling the process of irrigation system like Temperature, PH, Humidity are all measured and checked with the previous data stored in a system. According to the comparator of the system automation process of pump and solenoid valve is opened for needed. Temperature range is exceeded the certain range motor started and that range is up to 30°C. CLM53R Humidity sensor is used to measure humidity. A pH Sensor is used to measure the nutrient content in the soil required for irrigation. The other thing is that correct amount of nutrient is supplied to the soil. Power Supply of 230V AC is supplied which is stepped down to 5V-12V DC using a step-down transformer. All the sensor outputs are connected to the input part of the Arduino Uno Microcontroller which is based on the ATmega328 and consists of 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analogue inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button and Flash Memory of 32 KB of which 0.5 KB is used by boot loader, SRAM uses 2 KB, EEPROM uses 1 KB and has a Clock Speed 16 MHz The Output ports of the microcontroller are connected to one LCD Screen, two driver circuits and a buzzer. One of the driver circuit drives the water pump which requires 5 V and 200 mA of current. The other driver circuit drives solenoid valve which requires 12V and 500 mA of current. The driver circuit is used to control and regulate other circuits so as to regulate the current flow through the circuit. A buzzer is also connected to the microcontroller and requires 12V. The real time data is uploaded to a local server and can be accessed via a web application and/or an android application.

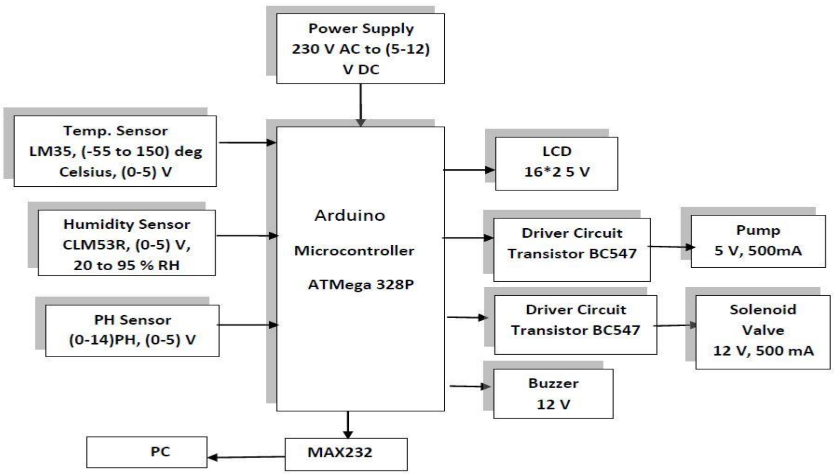

Figure 2.4 Block Diagram for Reference Paper [5]

[6] **G. Prasanna, S. Parvatham and S. Krishna, "Web Based Automatic Irrigation System Using Raspberry Pi Processor on Embedded Linux,"** ***International Journal & Magazine of Engineering, Technology, Management and Research,*** **vol. 3, no. 10, pp. 1036-1039, October 2016.**

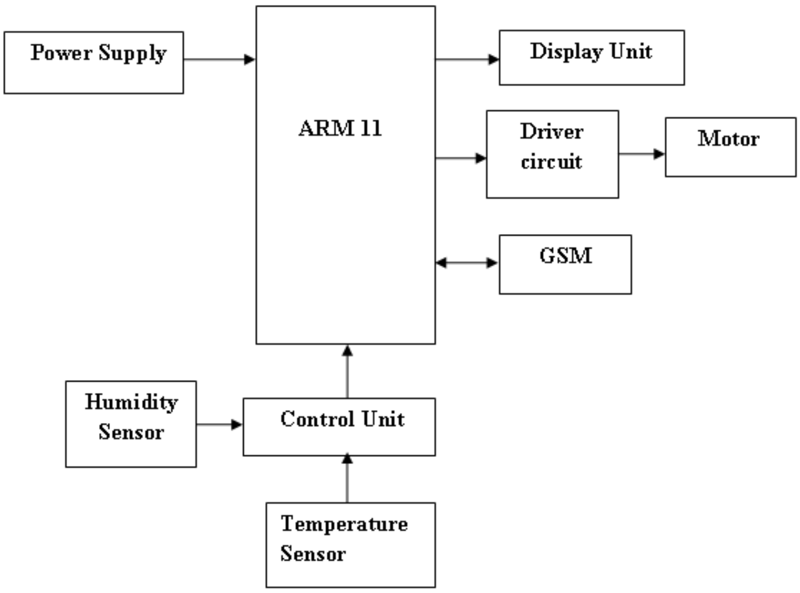

Figure 2.5 System Implementation for Reference Paper [6]

This project provides an excellent solution to the existing method. The farmer need not go to the farm to water his field. The system uses humidity sensor and temperature sensor to know the wetness percentage of the farm. Based on the output of the humidity sensor, the system sends the data to the processor, where the processor receives the relevant data and starts the motor to water the farm and as the output of the humidity sensor comes to the normal value, the system stops the motor. This project is implemented with high-speed raspberry pi processor.

[7] G. Ravikumar, T. VenuGopal, V. Sridhar and G. Nagendra, "Smart Irrigation System," ***International Journal of Pure and Applied Mathematics,*** **vol. 119, no. 15, pp. 1155-1168, 2018.**

This paper emphasizes on the fact that the technique which is incorporated here to monitor the soil's moisture content, enables agriculturalists feasibility of humidity measurement and a conventional automatic irrigation with the potential for eliminating excessive irrigation cycles thereby saving water to a significant extent. Besides the normal modes of measurement and analysis, Lab VIEW stands unique as grooming software in the field of Instrumentation and control engineering, which facilitates engineers to work in one platform with infinite possibilities along with a sophisticated control system. Effective crop treatment and water management is the major requisite in most of the cultivating estates in semi-arid regions. Monitoring the soil's nature, estimating it's moisture content and controlling it concurring to the necessity, proposes a potential solution to endorse landsite irrigation management and thus, to treat desiccated fields and provide prominent yield to producers. In the field of agriculture, the most important part is to get the information about the moisture content of soil.

The paper is designed to develop an automatic irrigation system which switches the pump motor ON/OFF on sensing the moisture content of the soil. In the field of agriculture, use of proper method of irrigation is important. This project is implemented in the Lab VIEW environment by interfacing MyRio. Corresponding to the surface's atmospheric conditions, transmission of the sensed voltage signal from the hardware circuitry on the Rio interfaced to a PC with Lab VIEW, which uses a development environment that is powerful and intuitive which could rapidly develop a user interface for data visualization and automatic irrigation of the soil. The advantage of using this method is to reduce human intervention and still ensure proper irrigation.

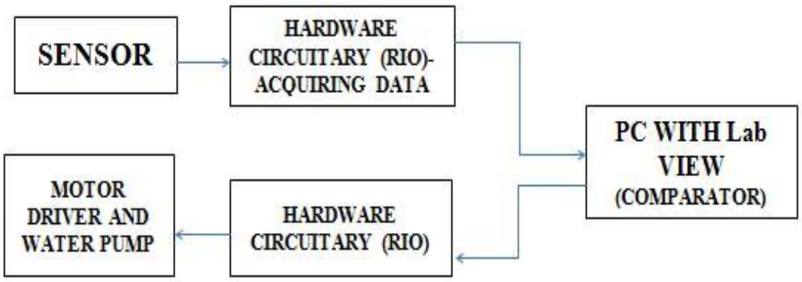

Figure 2.6 System Implementation for Reference Paper [7]

[8] A. Tyagi, N. Gupta, D. J. P. Navani, R. Tiwari and A. Gupta, "Smart Irrigation System," *International Journal for Innovative Research in Science & Technology,* vol. 3, no. 10, pp. 09-12, March 2017.

In this model, Smart Irrigation System is based on Arduino microcontroller. This prototype monitors the amount of soil moisture content in soil. A predefined value of soil moisture is set and can be varied with crops. In case the soil moisture of the soil deviates from the specified range, the watering system is turned ON/OFF. In case of dry soil, it will activate the irrigation system, pumping water for watering the plants. This project is mainly based on PROTEUS based software C programming language. In simulation, pin2 and pin3 are used as a input pin for DC motor and switch respectively.

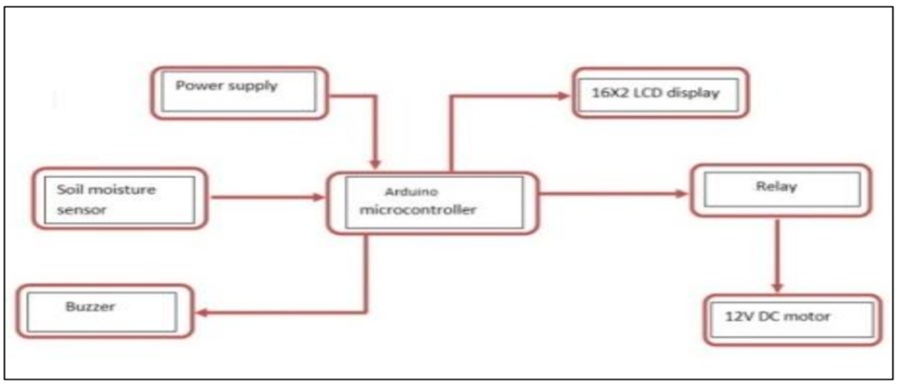

Figure 2.7 System Implementation for Reference Paper[8]

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATION

3. SYSTEM REQUIREMENTS SPECIFICATION

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of tests that describe user interactions that the software must provide. The software requirements specification document enlists enough and necessary requirements that are required for the project development. To derive the requirements we need to have clear and thorough understanding of the products to be developed or being developed. An SRS minimize the time and effort required by developers to achieve desired goals and also minimizes the development cost.

3.1 PURPOSE

The purpose of this SRS document is to provide a detailed overview of our software and hardware product, its parameters and goals. The document describes the projects target audience and its user interface, hardware and software requirements. Software Requirement Specification assures the project management, stakeholders and client that the development team has really understood the business requirement documentation properly. The SRS is documented in such a way that it breaks the deliverables into smaller components. The information is organized in such a way that the developers will not only understand the boundaries within which they need to work, but also what functionality needs to be developed and in what order**.**

3.2 SCOPE

This project work is complete on its own in providing real time data of the fields where the sensor is planted in the ground and irrigate those fields depending upon the value acquired via the sensors. The water pumps will be turned on automatically depending upon the soil moisture level and temperature.

Advantages

- It’s easy to implement
- Fast, efficient and user-friendly
- Increases the productivity of crops

3.3 SOFTWARE REQUIREMENTS

In this chapter the software used and all the program code dumping tools are explained. In this project, we have implemented coding using Embedded c and html in web application which is used for data projection along with a LCD.

3.3.1 ARDUINO IDE

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students with or without a background in electronics and programming. It is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read input- light on a sensor, a finger on a button, or a message - and turn it into an output - activating a motor, turning on an LED, publishing something online and many more. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so we need to use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

The main features of Arduino IDE are:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms.
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OS X, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well.

- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it.
- Open source and extensible software - The Arduino software is published as open source tool and the language can be expanded through C++ libraries.

3.3.2 EMBEDDED C

Embedded C is a set of language extensions for the C Programming language. C is often used for system programming, including implementing applications. Embedded C uses most of the syntax of standard C, e.g., main() function, variable definition, data type declaration, conditional statements (if, switch, case), loops (while, for), functions, arrays and strings, structures etc. It is small and simpler to learn, understand, program and debug. It is efficient & supports access to I/O and provides ease of management of large embedded projects.

The compiler derives its name from the way it works, looking at the entire piece of source code and collecting and reorganizing the instruction. See there is a bit little difference between compiler and an interpreter. Interpreter just interprets whole program at a time while compiler analyses and execute each line of source code in succession, without looking at the entire program.

Advantage of Embedded C

1) It is simpler to learn, understand, program and debug and fairly efficient.
2) Compared to assembly language, C code written is more reliable and scalable, more portable between different platforms.
3) Unlike assembly, C has advantage of processor-independence and is not specific to any particular microprocessor/microcontroller or any system.
4) As C combines functionality of assembly language and features of high level languages, it is treated as a "middle-level computer language” or "high level assembly language”.
5) It supports access to I/O and provides ease of management of large embedded projects.

6) Java is also used in many embedded systems but Java programs require the Java Virtual Machine (JVM), which consumes a lot of resources. Hence it is not used for smaller embedded devices.

3.4 HARDWARE REQUIREMENTS:

3.4.1 TEMPERATURE SENSOR (LM35):

The temperature sensor used to measure the temperature at the field is LM35 DZ. The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade).The LM35 does not require any external calibration or trimming to provide typical accuracies of °C at room temperature and °C over a full -55 to +150C temperature range. Low cost is assured by trimming and calibration at the water level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. The device is used with single power supplies, or with plus and minus supplies. As the LM35 device draws only 60 µA from the supply, it has very low self-heating of less than 0.1°C in still air.

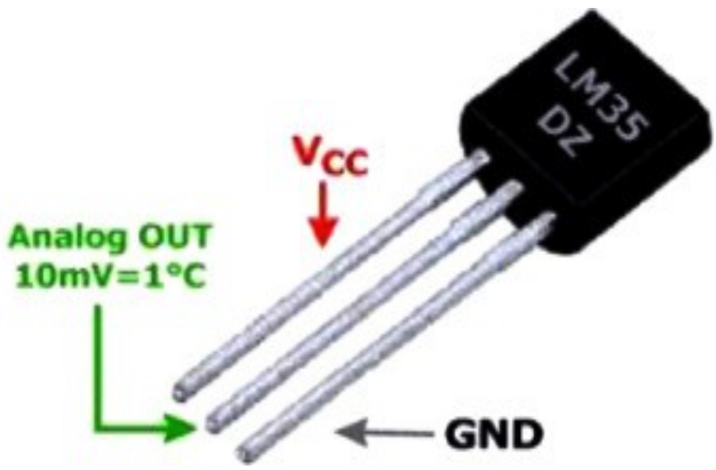

Figure 3.1 LM35 DZ Temperature Sensor

3.4.2 HUMIDITY SENSOR

DHT11 digital temperature and humidity sensor is a composite Sensor contains a calibrated digital signal output of the temperature and humidity. They consist of a humidity sensing component, a NTC temperature sensor (or thermistor) and an IC on the back side of the sensor. For measuring humidity they use the humidity sensing component which has two electrodes with moisture holding substrate between them. So as the humidity changes, the conductivity of the substrate changes or the resistance between these electrodes changes. This change in resistance is measured and processed by the IC which makes it ready to be read by a microcontroller. On the other hand, for measuring temperature of these sensors use a NTC temperature sensor or a thermistor. A thermistor is actually a variable resistor that changes its resistance with change of the temperature. These sensors are made by sintering of semi conductive materials such as ceramics or polymers in order to provide larger changes in the resistance with just small changes in temperature. The term "NTC" means "Negative Temperature Coefficient", which means that the resistance decreases with increase of the temperature.

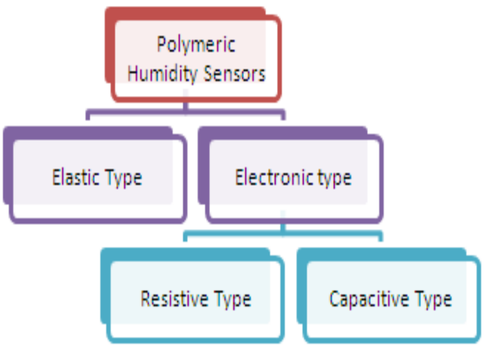

Figure 3.2 Types of Humidity Sensors

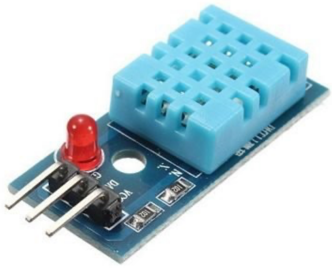

Figure 3.3 DHT11 Digital Humidity Temperature Sensor Module

3.4.3 SOIL MOISTURE SENSOR

Although soil water status can be determined by direct (soil sampling) and indirect (soil moisture sensing) methods, direct methods of monitoring soil moisture are not commonly used for irrigation scheduling because they are intrusive and labour intensive and cannot provide immediate feedback. Soil moisture probes can be permanently installed at representative points in an agricultural field to provide repeated moisture readings over time that can be used for irrigation management. Special care is needed when using soil moisture devices in coarse soils since most devices require close contact with the soil matrix that is sometimes difficult to achieve in these soils.

The basic technique for measuring soil water content is the gravimetric method. Because this method is based on direct measurements, it is the standard with which all other methods are compared. Unfortunately, gravimetric sampling is destructive, rendering repeat measurements on the same soil sample impossible. Because of the difficulties of accurately measuring dry soil and water volumes, volumetric water contents are not usually determined directly.

In the image below there are two components combined together to measure the soil moisture level. The First component is a two-legged lead that goes inside the soil. The soil moisture sensor used here is “RC- A-4079” from “Robocraze”. This first component is then connected to the second component which is an amplifier. The second component is then connected to the microcontroller.

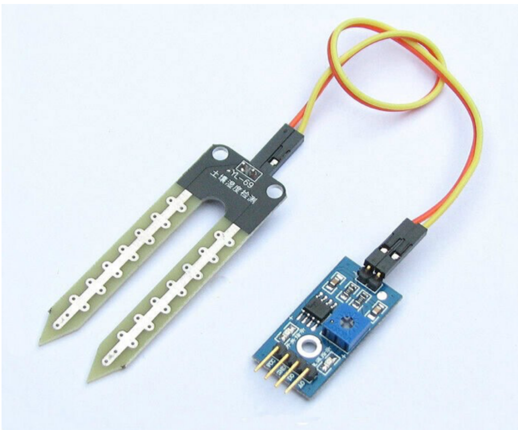

Figure 3.4 Soil Moisture Sensor

3.4.4 DC MOTOR/MOTOR DRIVER

A DC motor relies on the fact that like magnet poles repels and unlike magnetic poles attracts each other. A coil of wire with a current running through it generates an electromagnetic field aligned with the centre of the coil. By switching the current on or off in a coil its magnetic field can be switched on or off or by switching the direction of the current in the coil the direction of the generated magnetic field can be switched 180°. A simple DC motor typically has a stationary set of magnets in the stator and an armature with a series of two or more windings of wire wrapped in insulated stack slots around iron pole pieces (called stack teeth) with the ends of the wires terminating on a commutator. The armature includes the mounting bearings that keep it in the centre of the motor and the power shaft of the motor and the commutator connections. The total amount of current sent to the coil, the coil's size and what it's wrapped around dictate the strength of the electromagnetic field created. By turning on and off coils in sequence a rotating magnetic field can be created. These rotating magnetic fields interact with the magnetic fields of the magnets (permanent or electromagnets) in the stationary part of the motor (stator) to create a force on the armature which causes it to rotate. In some DC motor designs the stator fields use electromagnets to create their magnetic fields which allow greater control over the motor. At high power levels, DC motors are almost always cooled using forced air.

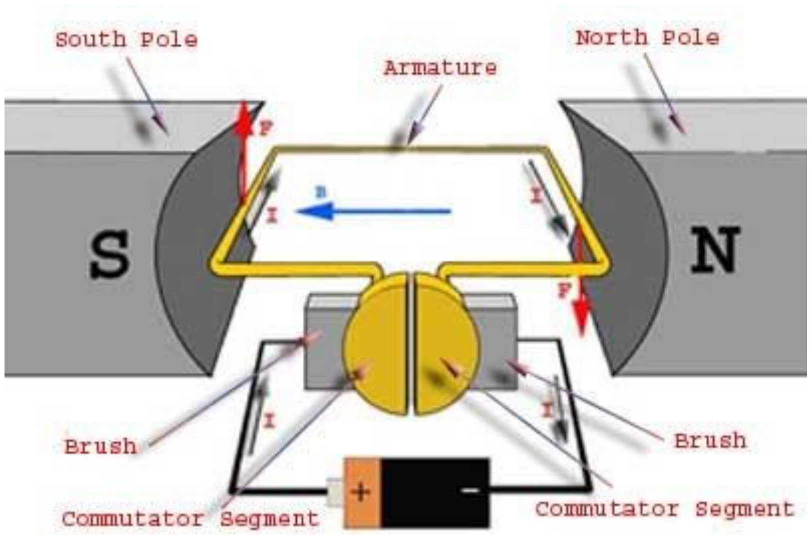

Figure 3.5 DC Motor Drive

The commutator allows each armature coil to be activated in turn. The current in the coil is typically supplied via two brushes that make moving contact with the commutator. Now, some brushless DC motors have electronics that switch the DC current to each coil on and off and have no brushes to wear out or create sparks.

Different number of stator and armature fields as well as how they are connected provides different inherent speed/torque regulation characteristics. The speed of a DC motor can be controlled by changing the voltage applied to the armature. The introduction of variable resistance in the armature circuit or field circuit allowed speed control. Modern DC motors are often controlled by power electronics systems which adjust the voltage by "chopping" the DC current into on and off cycles which have an effective lower voltage.

Since the series-wound DC motor develops its highest torque at low speed, it is often used in traction applications such as electric locomotives, and trams. The DC motor was the mainstay of electric traction drives on both electric and diesel-electric locomotives, streetcars/trams and diesel electric drilling rigs for many years. The introduction of DC motors and an electrical grid system to run machinery starting in the 1870s started a new second Industrial Revolution. DC motors can operate directly from rechargeable batteries, providing the motive power for the first electric vehicles and today's hybrid cars and electric cars as well as driving a host of cordless tools. Today DC motors are still found in applications as small as toys and disk drives, or in large sizes to operate steel rolling mills and paper machines.

If external power is applied to a DC motor it acts as a DC generator, a dynamo. This feature is used to slow down and recharge batteries on hybrid car and electric cars or to return electricity back to the electric grid used on a street car or electric powered train line when they slow down. This process is called regenerative braking on hybrid and electric cars. In diesel electric locomotives they also use their DC motors as generators to slow down but dissipate the energy in resistor stacks. Newer designs are adding large battery packs to recapture some of this energy.

3.4.5 ARDUINO

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. We can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so we use the Arduino programming language (based on wiring), and the Arduino Software (IDE), based on Processing.

The Arduino can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm centre-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

Feature	Specification
Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage(Recommended)	7V-12V
Input Voltage(Limits)	6V-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40mA
DC Current for 3.3V Pin	50mA
Flash Memory	32KB(of which 0.5 KB used by boot loader)
SRAM	2KB
EEPROM	1KB
Clock Speed	16MHz

Table 1 System Specification of Arduino

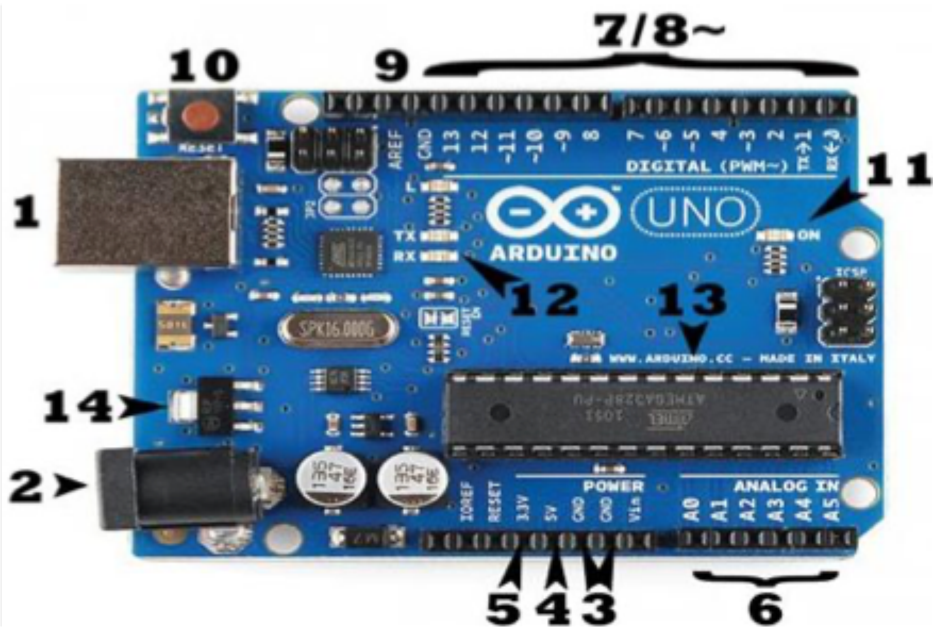

Figure 3.6 Arduino Microcontroller

3.4.5.1 Power

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply. In the picture above the USB connection is labelled **(1)**. The USB connection is also how we will load code onto your Arduino board.

3.4.5.2 Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)

The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjunction with a breadboard and some wire. They usually have black plastic „headers" that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labelled on the board and used for different functions.

GND **(3)**: Short for "Ground". There are several GND pins on the Arduino, any of which can be used to ground your circuit.

5V **(4)** & 3.3V **(5)**: As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.

Analog **(6)**: The area of pins under the „Analog In" label (A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analogue sensor (like a temperature sensor) and convert it into a digital value that we can read.

Digital **(7)**: Across from the analogue pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).

PWM **(8)**: You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM). We have a tutorial on PWM, but for now, think of these pins as being able to simulate analogue output (like fading an LED in and out).

AREF **(9)**: Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analogue input pins.

3.4.5.3 Reset Button

Just like the original Nintendo, the Arduino has a reset button **(10)**. Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn't repeat, but you want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn't usually fix any problems.

3.4.5.4 Power LED Indicator

Just beneath and to the right of the word "UNO" on your circuit board, there's a tiny LED next to the word 'ON' **(11)**. This LED should light up whenever you plug your Arduino into a power source. If this light doesn't turn on, there's a good chance something is wrong

3.4.5.5 TX RX LEDs

TX is short for transmit, RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for serial communication. In our case, there are two places on the Arduino UNO where TX and RX appear – once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs **(12)**. These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (like when we're loading a new program onto the board).

3.4.5.6 Main IC

The black thing with all the metal legs is an IC, or Integrated Circuit **(13).** Think of it as the brains of our Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the ATmega line of IC's from the ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC. If you want to know more about the difference between various IC‟s, reading the datasheets is often a good idea.

3.4.5.7 Voltage Regulator

The voltage regulator **(14)** is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it's for. The voltage regulator does exactly what it says – it controls the amount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course, it has its limits, so we shouldn't hook up the Arduino to anything greater than 20 volts.

3.4.6 LCD PANEL

A liquid-crystal display (LCD) is a flat-panel display or other electronic visual display that uses the light-modulating properties of liquid crystals. Liquid crystals do not emit light directly.

Before an electric field is applied, the orientation of the liquid-crystal molecules is determined by the alignment at the surfaces of electrodes. In a twisted nematic (TN) device, the surface alignment directions at the two electrodes are perpendicular to each other, and so the molecules arrange themselves in a helical structure, or twist. This induces the rotation of the polarization of the incident light, and the device appears gray. If the applied voltage is large enough, the liquid crystal molecules in the center of the layer are almost completely untwisted and the polarization of the incident light is not rotated as it passes through the liquid crystal layer. This light will then be mainly polarized perpendicular to the second filter, and thus be blocked and the pixel will appear black. By controlling the voltage applied across the liquid crystal layer in each pixel, light can be allowed to pass through in varying amounts thus constituting different levels of gray.

Below is a basic 16 character by 2-line display.16×2 LCD module is a very common type of LCD module that is used in 8051 based embedded projects. It consists of 16 rows and 2 columns of 5×7 or 5×8 LCD dot matrices. The module was are talking about here is type number JHD162A which is a very popular one. It is available in a 16-pin package with back light, contrast adjustment function and each dot matrix have 5×8 dot resolution.

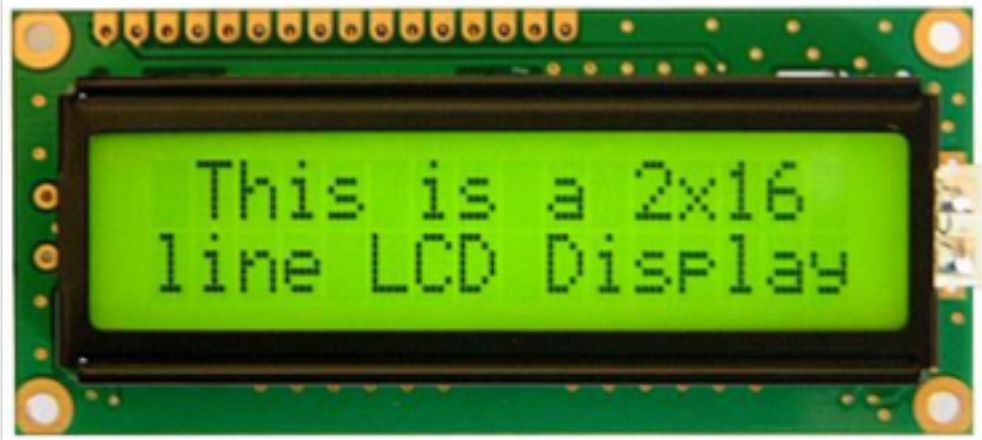

Figure 3.7 2X16 LCD Display

3.4.7 WATER PUMP

The water pump is used to artificially supply water for a particular task. It can be electronically controlled by interfacing it to a microcontroller. It can be triggered ON/OFF by sending signals as required. The process of artificially supplying water is known as pumping. There are many varieties of water pumps used. This project employs the use of a small water pump which is connected to a H-Bridge.

The pumping of water is a basic and practical technique, far more practical than scooping it up with one's hands or lifting it in a hand-held bucket. This is true whether the water is drawn from a fresh source, moved to a needed location, purified, or used for irrigation, washing, or sewage treatment, or for evacuating water from an undesirable location. Regardless of the outcome, the energy required to pump water is an extremely demanding component of water consumption. All other processes depend or benefit either from water descending from a higher elevation or some pressurized plumbing system.

Figure 3.8 Water Pump

CHAPTER 4
SYSTEM DESIGN

4. DESIGN

The design consists of the description of the overall system which includes the architecture, server side database, class diagram, activity diagrams and sequence diagrams formulated in order to implement the proposed system.

4.1 SYSTEM ARCHITECTURE

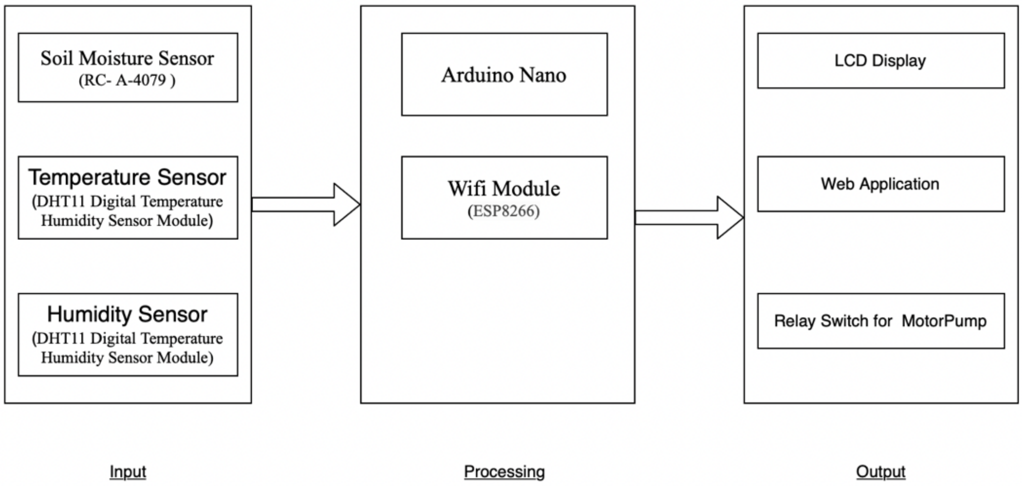

Figure 4.1 System Architecture of Proposed System

4.1.1 POWER SOURCE

The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source. Any good smartphone charger will do the work of powering the Nano.

4.1.2 STORAGE

The Arduino Nano has onboard storage available. The executable set of instruction are compiled using the Arduino IDE and then loaded into the Nano's 32KB Flash Storage.

4.1.3 GPIO (GENERAL PURPOSE INPUT OUTPUT)

General-purpose input/output (GPIO) is a generic pin on an integrated circuit whose behaviour, including whether it is an input or output pin, can be controlled by to user

At user runtime. GPIO pins have no special purpose defined, and go unused by default. The idea is that sometimes the system designer building a full system that uses the chip might find it useful to have a handful of additional digital control lines, and having these available from the chip can save the hassle of having to arrange additional circuitry to provide them. Each of the 14 digital pins on the Nano can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. The Nano has 8 analogue inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the analogReference() function. Analog pins 6 and 7 cannot be used as digital pins.

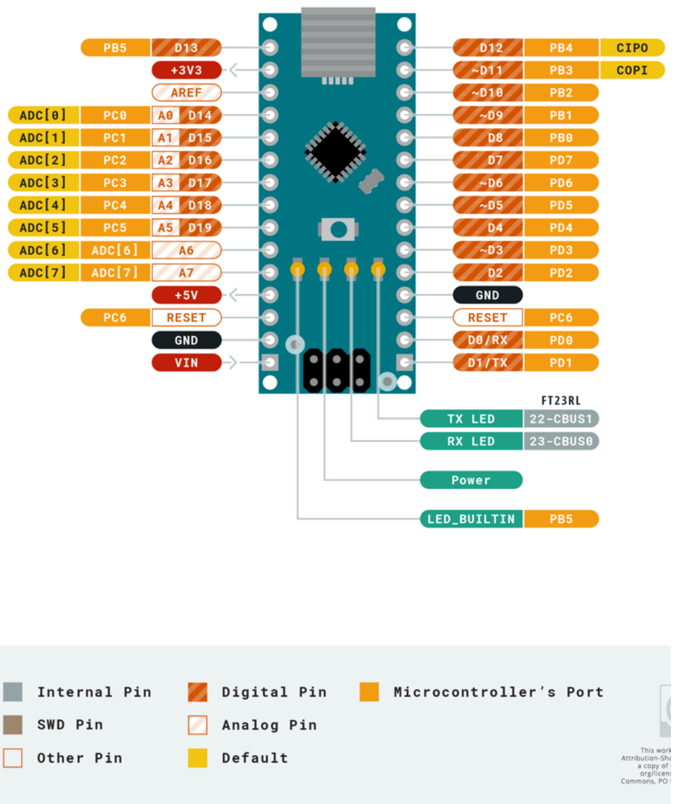

Figure 4.2 Arduino Nano Pinout

4.1.4 ARDUINO COMMUNICATION

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provide UART

TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Nano's digital pins. The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. To use the SPI communication, please see ATmega328 datasheet.

4.2 RF MODULE (TRANSMITTER AND RECEIVER)

This RF module comprises of an RF Transmitter and an RF Receiver. The transmitter/receiver (Tx/Rx) pair operates at a frequency of 434 MHz. An RF transmitter receives serial data and transmits it wirelessly through RF through its antenna connected at pin4. The transmission occurs at the rate of 1Kbps - 10Kbps.The transmitted data is received by an RF receiver operating at the same frequency as that of the transmitter. The RF module is often used along with a pair of encoder/decoder. The encoder is used for encoding parallel data for transmission feed while reception is decoded by a decoder. HT12E-HT12D, HT640-HT648, etc. are some commonly used encoder/decoder pair ICs.

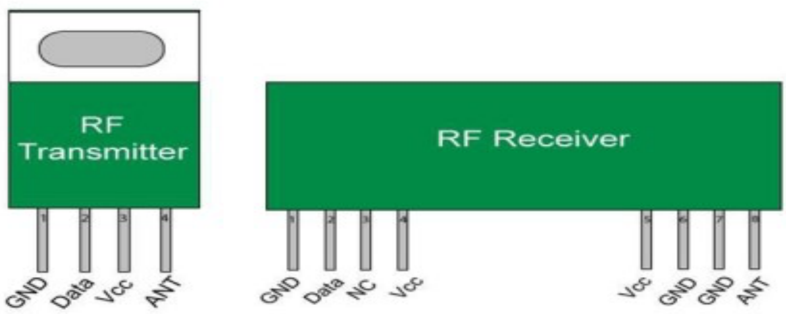

Figure 4.3 RF-Transmitter and Receiver

4.3 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

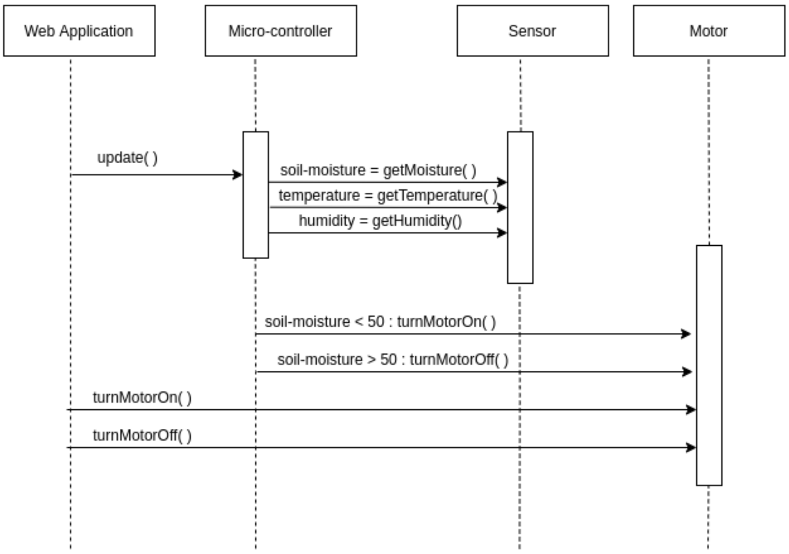

Figure 4.4 Sequence Diagram for User via Web Application

4.4 ACTIVITY DIAGRAM

The Activity diagram shows the flow of activities or the actions that occur when user tries to interact with the system and tries to access the functionality provided by the system.

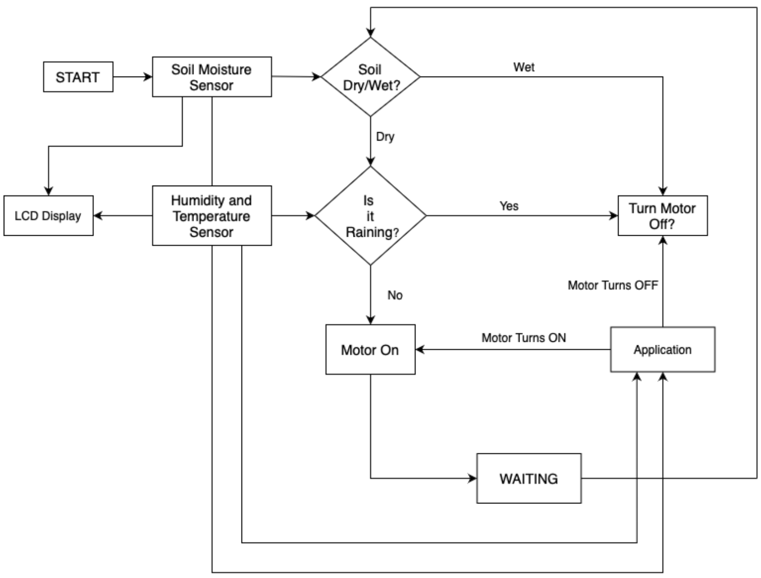

Figure 4.5 Activity Diagram of Proposed Model

CHAPTER 5
TESTING

5. TESTING

Software Testing is the process used to identify the completeness, correctness, security and quality of the product being developed. Testing involves technical investigation which is performed on behalf of end users. This is intended to reveal quality related information about the product in a context in which it is intended to operate. This is just not limited to the process of executing a program or application with the intent of finding errors. Quality is not absolute and will never completely establishes the correctness of computer software.

The software testing is usually performed for the following objectives:-

a) Verification and Validation: It is to test that the product is build the right way .i.e. are correct procedure is used for the development of software so that it can meet the user requirements.

b) Software Reliability Estimation: The objective is to discover the residual designing errors before delivery to the customer. The failure data during process are taken down in order to estimate the software reliability.

5.1 UNIT TESTING

Unit testing is a method in which each unit or set of one or more modules are tested with associated data, operating procedure and usage method for determining fitness of product for use. Each unit can be defined as smallest testable part of the product. A unit can be entire module but commonly it is commonly individual procedure or function. We have tested all the parts of the project individually such as sensors and all the function are also tested and the expected result is obtained.

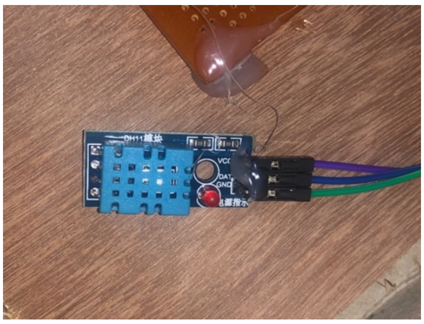

Figure 5.1 DHT Unit Testing

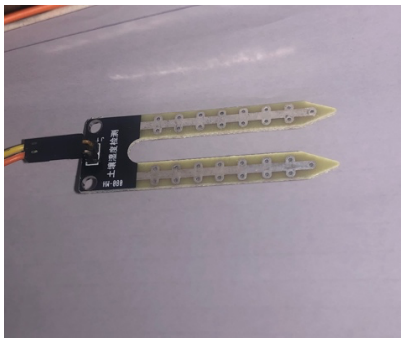

Figure 5.2 Soil Moisture Unit Testing

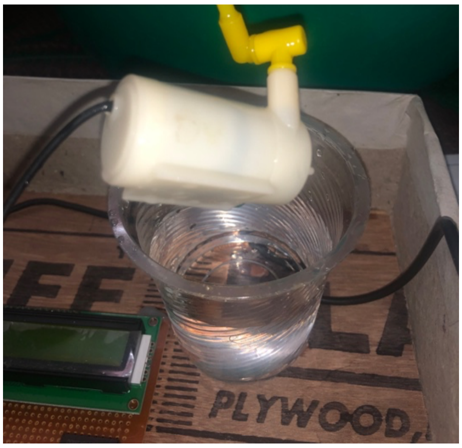

Figure 5.3 Water Pump Unit Testing

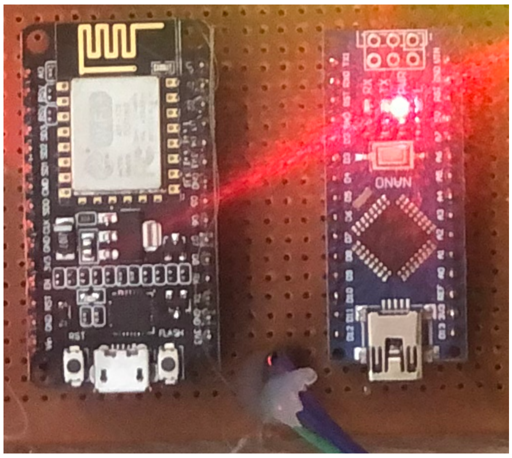

Figure 5.4 Microcontroller and Network Module's Unit Testing

5.2 INTEGRATION TESTING

Integration testing is a part of software testing in which each individual module are combined into a group and tested. It occurs after finishing unit testing. Integration takes modules that have been tested in unit testing as input, groups them and applies test case defined. The need of integration testing is to verify performance, functional and reliable requirements mentioned during design phase. Test cases are defined to test that all components connected interact correctly using procedure calls. If some module fails to meet minimum requirement, changes are made to that module so that it starts functioning properly. The various modules of the automated system are integrated and tested to check whether they perform well together or not.

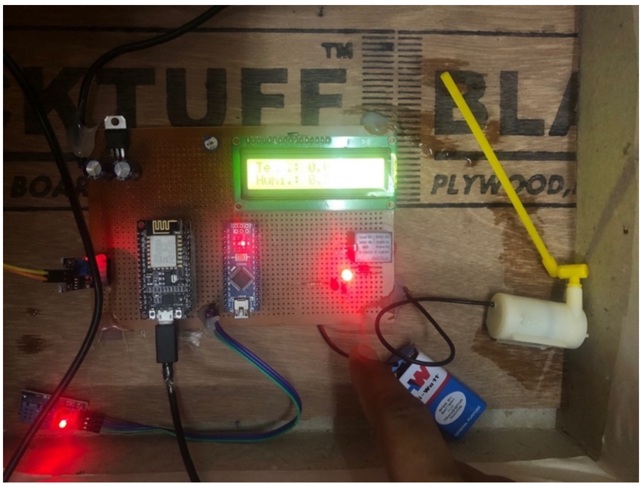

Figure 5.5 Integration Testing

5.3 REGRESSION TESTING

Regression testing is any type of software testing that seeks to uncover new software bugs, or regressions, in existing functional and non-functional areas of a system after changes, such as enhancements, patches or configuration changes, have been made to them. The intent of regression testing is to ensure that a change such as those mentioned above has not introduced new faults. One of the main reasons for regression testing is to determine whether a change in one part of the software affects other parts of the software. Common methods of regression testing include rerunning previously-completed tests and checking whether program behaviour has changed and whether previously-fixed faults have re-emerged.

5.4 BLACK BOX TESTING

Black box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well. Specific knowledge of the application's code/internal structure and programming knowledge in general is not required. The tester is aware of *what* the software is supposed to do but is not aware of *how* it does it.

5.4.1 TESTCASES

Test cases are built around specifications and requirements, i.e., what the application is supposed to do. Test cases are generally derived from external descriptions of the software, including specifications, requirements and design parameters. Although the tests used are primarily functiona*l* in nature, non-functional tests may also be used. The test designer selects both valid and invalid inputs and determines the correct output, often with the help of an oracle or a previous result that is known to be good, without any knowledge of the test object's internal structure.

1. **Testcase**: - Soil Moisture Sensor should read the moisture level and project in percentage in the web application.

   Expected Result: - Moisture level is calculated and projected in percentage in the web application.

Test Result: Successful

Screenshot:

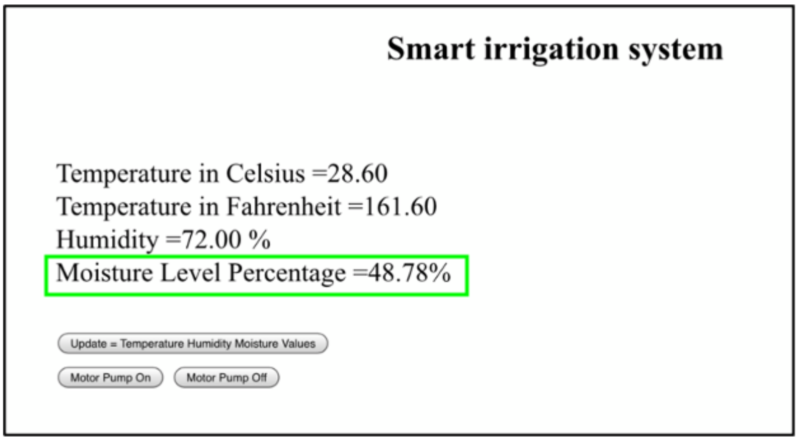

Figure 5.6 Result of Testcase 1

2.**Testcase**:- DHT Sensor should read the temperature and humidity value and display the acquired value in the web application.

Expected Result: - Projection of Temperature and Humidity Value in the web application

Test Result: - Successful

Screenshot-

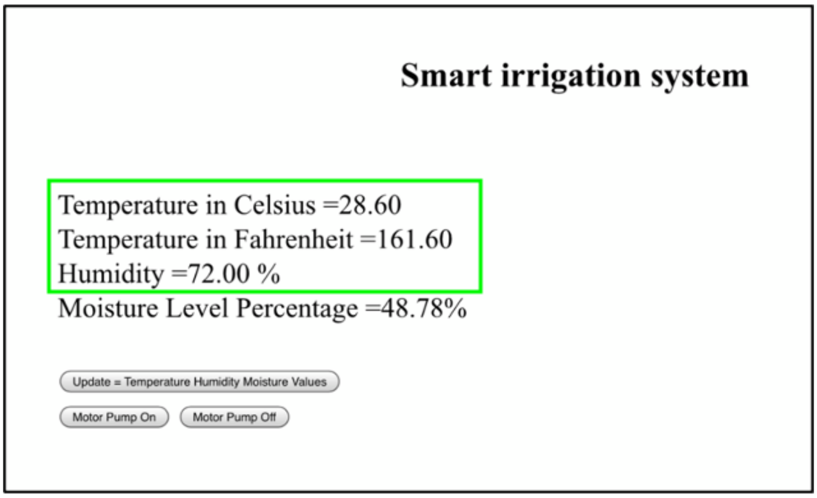

Figure 5.7 Result of Testcase 2

3.**Testcase**: - DHT Sensor should projected the acquired value to the LCD Display.

Expected Result: - Data Should be projected in LCD Display.

Test Result: - Successful

Screenshot-

Figure 5.8Result of Testcase 3

4.**Testcase**: - Motor pump should be Turned ON/OFF from the web application

Expected Result: - Motor is Turned ON/OFF

Test Result: - Successful

Screenshot-

Figure 5.9 Result of Testcase 4

5.5 WHITE BOX TESTING

White Box Testing (also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code Based Testing or Structural Testing) is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming knows how and the implementation knowledge is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a system.

The code of the developed project has been tested successfully with various inputs to achieve the desired output. For example, the code including the port that connects the temperature sensor and smoke sensor provide input to the respective port mentioned in the code. Thus the developed project also satisfies the white box testing.

5.6 SNAPSHOTS

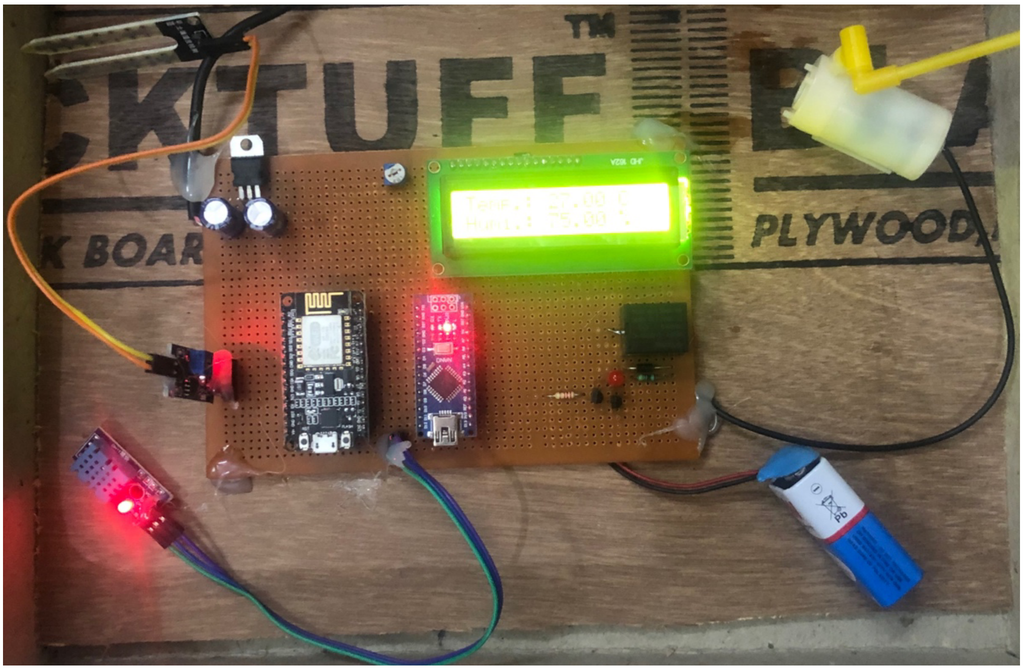

Figure 5.10 System Implementation Snapshot I

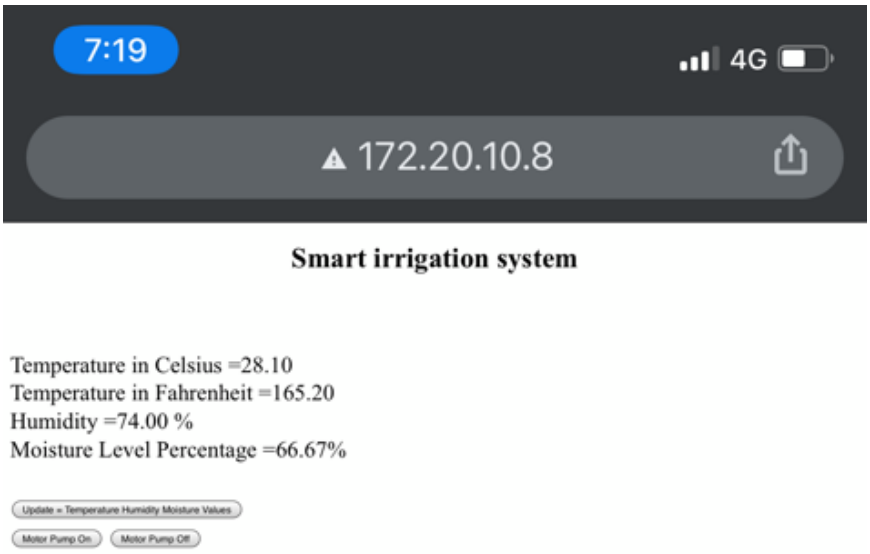

Figure 5.11 System Implementation Snapshot II

CHAPTER 6

IMPLEMENTATION

6. IMPLEMENTATION

6.1 SYSTEM IMPLEMENTATION

An implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through computer programming and deployment. Many implementations may exist for a given specification or standard. For example, web browsers contain implementations of World Wide Web Consortium-recommended specifications, and software development tools contain implementations of programming languages.

To implement a system successfully, a large number of inter-related tasks need to be carried out in an appropriate sequence. Utilizing a well-proven implementation methodology and enlisting professional advice can help but often it is the number of tasks, poor planning and inadequate resourcing that causes problems with an implementation project, rather than any of the tasks being particularly difficult. Similarly, with the cultural issues it is often the lack of adequate consultation and two-way communication that inhibits achievement of the desired results.

Developing “Smart Irrigation System” was a major challenge for us. Various hardware and software are being used to make an effective system. The microcontroller used for this project is Arduino Nano. This microcontroller will help to coordinate all the activities of the robot. The microcontroller receives the measured value about the moisture level of the soil from the soil moisture sensor in the analogue form and digitizes them. The Humidity and Temperature sensor both receive signal in binary for which will be read, converted and put into working by the microcontroller. The microcontroller will have predefined functions in its memory and executes the necessary commands as per the input received.

6.2 HARDWARE IMPLEMENTATION

6.2.1 ARDUINO NANO

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs and turn it into an output. We can tell the microcontroller board what to do by sending a set of instructions to it. To do so we use the Arduino programming language (based on wiring), and the Arduino

Software(IDE), based on Processing. The Arduino Nano can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm centre-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

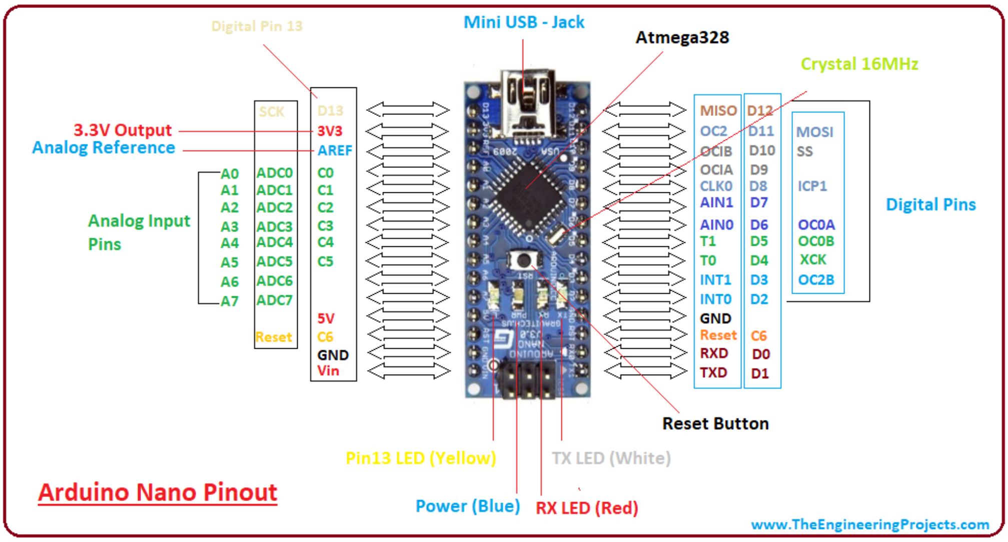

Figure 6.1 Arduino Nano Pinout diagram

6.2.2 JUMP WIRE

A jump wire (also known as jumper, jumper wire, jumper cable, DuPont wire or cable) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering. Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

Figure 6.2: Jump Wire Types

6.2.3 ARDUINO CONTROLLER IDE

Arduino Controller IDE helps to implement the software realization into the hardware and makes to hardware to follow the written instruction. The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Figure 6.3: Arduino Controller IDE

6.2.4 ARM MICROCONTROLLER

ARM, previously Advanced RISC Machine, originally Acorn RISC Machine, is a family of reduced instruction set computing (RISC) architectures for computer processors, configured for various environments. British company ARM Holdings develops the architecture and licenses it to other companies, who design their own products that implement one of those architectures—including systems-on-chips (SoC) and systems-on-modules (SoM) that incorporate memory, interfaces, radios, etc. It also designs cores that implement this instruction set and licenses these designs to a number of companies that incorporate those core designs into their own products.

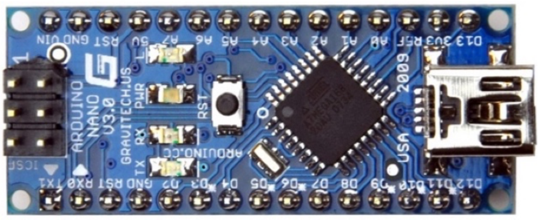

Figure 6.4 ARM Microcontroller

6.2.5 UNIVERSAL SERIAL BUS(USB)

USB was designed to standardize the connection of computer peripherals (including keyboards, pointing devices, digital cameras, printers, portable media players, disk drives and network adapters) to personal computers, both to communicate and to supply electric power. It has largely replaced interfaces such as serial ports and parallel ports, and has become commonplace on a wide range of devices. USB connectors have replaced other types for battery chargers.

Figure 6.5:USB cable

6.2.6 BREADBOARD

A breadboard, or protoboard, is a construction base for prototyping of electronics. Originally the word referred to a literal bread board, a polished piece of wood used when slicing bread. In the 1970s the solderless breadboard (a.k.a. plugboard, a terminal array board) became available and nowadays the term "breadboard" is commonly used to refer to these.

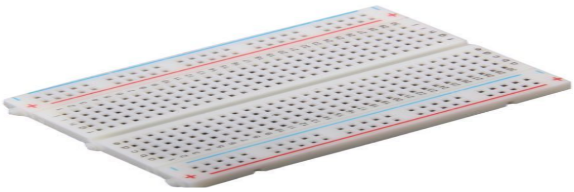

Figure 6.6: Bread Board

6.2.7 WI-FI MODULE

The Wi-Fi module used here to communicate with the web application is *ESP-12E:ESP8266* Serial Port WIFI Wireless Transceiver Module For Arduino. This small WIFI transceiver is the perfect solution for IoT applications and for our Smart Irrigation System. It can be the replacement to NRF24L01. It can talk to the Wi-Fi router directly through the UART MCU's (Rx, Tx).The ESP-12 module is one of the most complete of the ESP family as it allows us to use the biggest amount of pins of all of them. We can program this module to work stand alone with the Arduino IDE or with LUA as Node MCU. The features of this module are: energy saving VoIP, quickly switch between the sleep/wake patterns, low-power operation, adaptive radio bias, front-end signal processing functions. The ESP8266-03 is one of the smallest versions of this module. It includes a built-in ceramic antenna.

Figure 6.7:ESP-12E:ESP8266

Figure 6.8:ESP-12E:ESP8266 Pin Description

6.3 SOFTWARE IMPLEMENTATION

6.3.1 ARDUINO PROGRAMMING

Arduino programs are written in the Arduino Integrated Development Environment (IDE). Arduino IDE is a special software running on your system that allows you to write sketches (synonym for program in Arduino language) for different Arduino boards. The Arduino programming language is based on a very simple hardware programming language called processing, which is similar to the C language. After the sketch is written in the Arduino IDE, it should be uploaded on the Arduino board for execution.

The first step in programming the Arduino board is downloading and installing the Arduino IDE. The open-source Arduino IDE runs on Windows, Mac OS X, and Linux.

Figure 6.9: Arduino Programming Example

6.4 CODE AND UI

6.4.1 ARDUINO CODE

```
#include <ESP8266WiFi.h>
 #include "DHT.h"  // DHT11's Predefined Library
#define DHTTYPE DHT11   // DHT 11
#define dht_dpin 0      //GPIO-0 D3 pin of nodemcu

int Raw       = A0;      //Analog channel A0 used to measure temperature
int threshold = 16;  // Digital pin water sensor read-GPIO16---D0 of NodeMCU
int Solenoid = 13;       // GPIO13---D7 of NodeMCU--Motor connection
const char* ssid = "POCO PHONE2";
const char* password = "akj010101";
DHT dht(dht_dpin, DHTTYPE);
WiFiServer server(80);

void setup(void)
{
  dht.begin();
  Serial.begin(9600);
  delay(10);
  pinMode(threshold,INPUT_PULLUP); //Pin#13 as output-Activate pullup at
pin 13
  pinMode(Solenoid, OUTPUT);       //D7 as output
  digitalWrite(Solenoid, LOW);     //Deactivate Solenoid
  // Connect to WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);     //Begin WiFi

 while (WiFi.status() != WL_CONNECTED) {
   delay(500);
```

```
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  // Start the server
  server.begin();
  Serial.println("Server started");

  // Print the IP address on serial monitor
  Serial.print("Use this URL to connect: ");
  Serial.print("http://");    //URL IP to be typed in mobile/desktop browser
  Serial.print(WiFi.localIP());
  Serial.println("/");
}
void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }
  // Wait until the client sends some data
  Serial.println("new client");
  while(!client.available()){
    delay(1);
  }
  // Read the first line of the request
  String request = client.readStringUntil('\r');
  Serial.println(request);
  client.flush();

 float h =0.0;  //Humidity level
```

```
float t =0.0;  //Temperature in Celsius
float f =0.0;  //Temperature in Fahrenheit
float percentage = 0.0; // Calculating percentage of moisture
float reading    = 0.0; //Analog channel moisture read

 // Match the request
 int value = LOW;
 if (request.indexOf("/Up=ON") != -1)  {
    h = dht.readHumidity();    //Read humidity level
    t = dht.readTemperature(); //Read temperature in Celsius
    f = (h * 1.8) + 32;        //Temperature converted to Fahrenheit
    reading = analogRead(Raw); //Analog pin reading output voltage by water
moisture rain sensor
    percentage = ( 100 - ( (reading/1023.00) * 100 ) );  //Converting the raw value
in percentage

   if (percentage>=50){  // If less moisture in soil start the motor otherwise stop
   digitalWrite(Solenoid, LOW);
 //  value = HIGH;
   }
   else {
   digitalWrite(Solenoid, HIGH);
 //  value = LOW;
   }


 }


 if (request.indexOf("/Solenoid=ON") != -1)  {   //Motor ON
   digitalWrite(Solenoid, HIGH);
   value = HIGH;
 }
 if (request.indexOf("/Solenoid=OFF") != -1)  {  //Motor OFF
```

```
  digitalWrite(Solenoid, LOW);
  value = LOW;
}
// Return the response
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println(""); //  do not forget this one
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<h1 align=centre>Smart irrigation system</h1><br><br>");
client.println("Temperature in Celsius =");
client.println(t);
client.println("<br>");
client.print("Temperature in Fahrenheit =");
client.println(f);
client.println("<br>");
client.print("Humidity =");
client.println(h);
client.print(" %");
client.println("<br>");
client.println();
client.print("Moisture Level Percentage =");
client.print(percentage);
client.print("%");
 if(digitalRead(threshold)==HIGH)
{
//client.println("Threshold Reached = Rain detected / Moisture exceeded /
Water detected");
}
client.println("<br><br>");
client.println("<br><br>");
```

```
client.println("<a href=\"/Up=ON\"\"><button>Update = Temperature Humidity Moisture Values</button></a><br />");
client.println("<a href=\"/Solenoid=ON\"\"><button>Motor Pump On </button></a>");
client.println("<a href=\"/Solenoid=OFF\"\"><button>Motor Pump Off </button></a><br />");
client.println("</html>");
delay(1);
Serial.println("Client disconnected");
Serial.println("");
}
```

6.4.2 APPLICATION UI

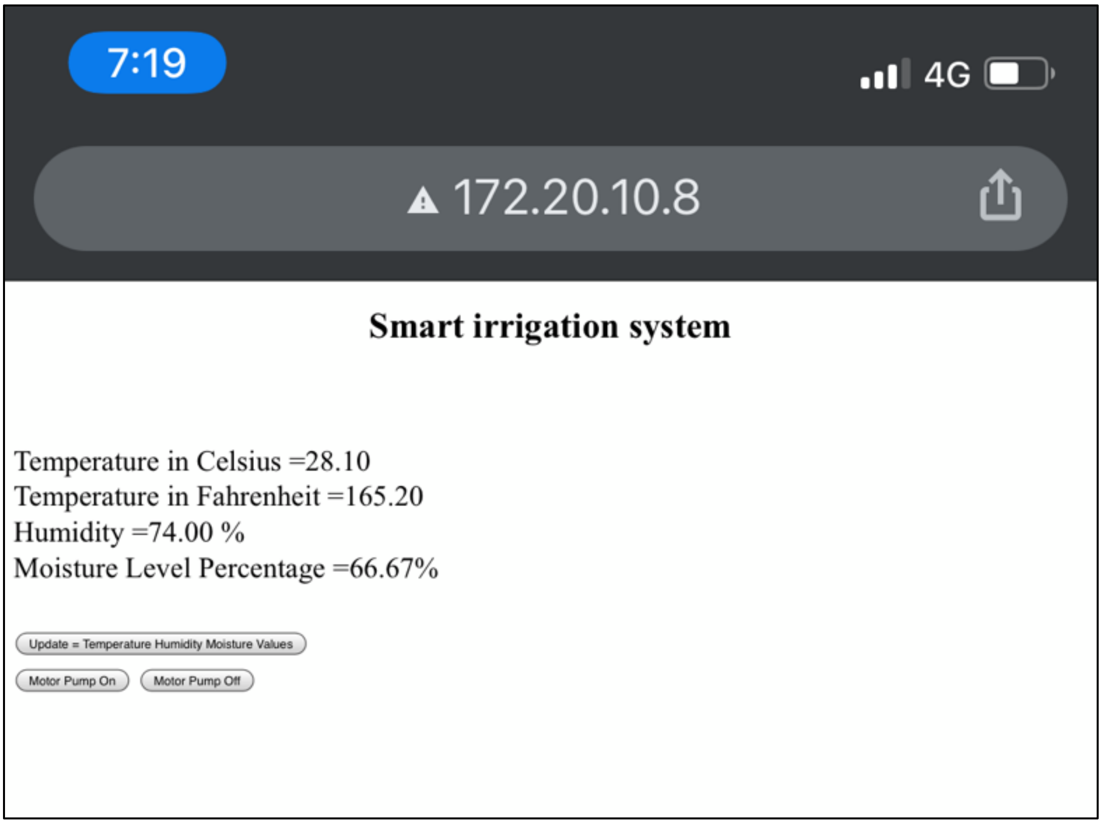

Figure 6.10 Application UI

CONCLUSION

With the decrease in water resources, the effective utilization of water is very important. It is necessary to have a system in place that not only irradicates the tedious job of irrigating huge fields for the professional farmers but also for backyard farmers who have their day job and cannot provide proper attention to the plants they plant. With a hike in global temperature, the summers are getting hotter by the year which means the soil moisture is being evaporated at a faster rate than anticipated and cracks in soil are developing at an alarming rate. These problems have negatively affected natural irrigation procedure like rain and as farmer turn to ground water for irrigation, dependencies on the ground water have increased so much that proper water utilization through effective irrigation is a must. The vast majority of Indian industries are agriculture based and get their raw materials from the agricultural industry. A smart sensor-based system for water irrigation is a must and can be optimized depending upon the planted crops as Indian farmers plant different types of crops throughout the year. The proposed smart irrigation system decides the necessity of irrigating the field according to the information received by the microcontroller from the attached sensors. It is obvious that the "Smart Irrigation System" consumes less water for production and saves more water for future use. The proposed system would be no-contact i.e. the farmer need not be involved with the operation of the system. The proposed system would also be non-invasive in nature to the plants i.e. the operation of the system in no way affects the habitat of the plants, it's growth and overall it's lifespan.

FUTURE ENHANCEMENT

There are a variety of enhancements that could be made to this system to achieve

Greater accuracy in sensing and event detection.

⇒ Additional sensors can be added to the existing system to read accurate data and increase functionality of the system depending upon the user's need and requirement.

⇒ Addition of software modules that generate notification based on the data collected by the sensors so the user is updated automatically even over remote connection.

⇒ Use of cluster system in order to cover large fields which could be managed by a powerful central system would be more reliable than standalone systems.

⇒ Use of cloud server instead of the existing local server in order to provide collected data to a remote server and execute remote commands.

⇒ Use of more complex processing unit would enable easier and efficient use of cloud server system and would also aid in executing remote commands, internet connectivity and multiple request handling.

⇒ Platform specific applications can be created to utilize maximum device resource and functionality to enable more user friendliness of the system.

⇒ Use of APIs instead of physical sensor in case of cluster system would make it more cheap to produce and takes less time to configure while adding new node under a cluster head.

BIBLIOGRAPHY

[1] P. K. and D. S. Rathi, "An IoT Based Smart Irrigation System," *International Journal of Scientific & Engineering Research,* vol. 8, no. 5, pp. 44-51, May 2017.

[2] M. D. Chaware, M. M. Panse, M. A. Raut and M. A. Koparkar, "Sensor Based Automated Irrigation System," *International Journal of Engineering Research & Technology,* vol. 4, no. 05, pp. 33-37, May 2015.

[3] V. L. Akubattin, A. P. Bansode, T. Ambre, A. Kachroo and P. P. SaiPrasad "Smart Irrigation System," *International Journal of Scientific Research in Science and Technology,* vol. 2, no. 5, pp. 343-345, October 2016.

[4] S. Darshna, T. Sangavi, S. Mohan, A. Soundharya and S. Desikan, "Smart Irrigation System," *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE),* vol. 10, no. 3, pp. 32-26, 2015.

[5] G. Parameshwaran and K. Sivaprasath, "Arduino Based Smart Drip Irrigation System Using Internet of Things," *International Journal of Engineering Science and Computing,* vol. 6, no. 5, pp. 5518-5521, May 2016.

[6] G. Prasanna, S. Parvatham and S. Krishna, "Web Based Automatic Irrigation System Using Raspberry Pi Processor on Embedded Linux," *International Journal & Magazine of Engineering, Technology, Management and Research,* vol. 3, no. 10, pp. 1036-1039, October 2016.

[7] G. Ravikumar, T. VenuGopal, V. Sridhar and G. Nagendra, "Smart Irrigation System," *International Journal of Pure and Applied Mathematics,* vol. 119, no. 15, pp. 1155-1168, 2018.

[8] A. Tyagi, N. Gupta, D. J. P. Navani, R. Tiwari and A. Gupta, "Smart Irrigation System," *International Journal for Innovative Research in Science & Technology,* vol. 3, no. 10, pp. 09-12, March 2017.

International Journal of All Research Education and Scientific Methods (IJARESM), ISSN: 2455-6211
Volume 9, Issue 7, July -2021, Impact Factor: 7.429, Available online at:www.ijaresm.com

Smart Irrigation Model Using IOT

Mrs. Lavanya Santhosh[1], Mrs. Veena Potdar[2], Kunal Jayswal[3], Nabin Khadka[4], Pragik Timsina[5], Poojit Chowdry[6]

[1]Assistant Professor, Department of Computer Science and Engineering, Dr. Ambedkar Institute of Technology, Bangalore
[2]Associate Professor, Department of Computer Science and Engineering, Dr. Ambedkar Institute of Technology, Bangalore
[3,4,5,6]Student, Department of Computer Science and Engineering, Dr. Ambedkar Institute of Technology, Bangalore

---***************---

ABSTRACT

In this paper we summarize the importance of healthy irrigation practices with the help of IOT controlled systems. With advanced farming techniques backed by today's advanced scientific methodologies, the frequency of crop production has increased and with that we have over used the resources required for production. The Smart Systems would assist in increasing the production of crops and promote sustainable resource utilization which in turn optimizes the timing and volume of water distribution in the fields. The proposed systems are flexible in nature that would enable us to either increase or decrease the scale for which it would be applied hence targeting both the casual backyard farmers and professional farmers. Our proposed system is simple enough to be used by a farmer with minimal technical knowledge.

Keywords: Irrigation, IOT, Farmer, Crop, Sensor, Soil Moisture, Temperature Sensor, Humidity Sensor

INTRODUCTION

The need to design a smart irrigation system is to replace the monotonous job of checking the field frequently for irrigation status which includes over irrigation, water flow in canals and under irrigation. In a small field this can be manageable. But for large fields this would be very tedious as they have to keep on travelling around their field for frequent inspection.

The farmers need to irrigate the field based on their gut feeling which isn't always sunshine and rainbows. Sometimes they under irrigate and have to repeat the irrigation process all over again and sometimes they over irrigate and can do nothing but hope for the best. About 16% of Indian Economy depends on agriculture.

The vast majority of industries are agriculture based and get their raw materials from the agricultural industry which signifies that agricultural production plays a vital role in Indian economy. Considering all these factors, a smart sensor- based system for water irrigation is a must and can be optimized depending upon the planted crops as Indian farmers plant different types of crops throughout the year.

This irrigation model hopes to eradicate this exact problem for the farmer by providing accurate real time data of the fields in their mobile phone. If any problem is detected in the field, the farmer will be notified through an alert on the phone.

If this model fails at any point in time and encounters a problem with valves (opening and closing), the farmer can override the system through his phone and gain instant control of the situation. The water usage at the early stages of a plant's life ishigh and relatively slows down throughout the plant's life cycle.

If the sprout is under-irrigated, it not only affects the growth and development of the crop but also decreases the soil quality. If the sprout is over-irrigated then it will loosen the soil around the roots and cause the sprout to be malnourished. Similar to the sprouts, in later stages of plant's life, under irrigation would not provide enough water for the plant's development in the reproduction stage which would result in irregular to no yields and over-irrigation tends to sweep away essential nutrients from the soil and causing the plant to unroot as the loose soil would mean that the plant cannot hold its weight by itself. Hence, an adequate amount of water needs to be supplied to the crop for maximum yields and increased overall production.

LITERATURE SURVEY

The objective of this project was to improve the farming and provide a big help to the farmer using it. The main idea of the project by G. Parameswaran and K. Sivaprasath [1], is to water the plants automatically based on the sensor data so, the growth of the plant can be enhanced. Actually, at this time it is very difficult to find caretaker for the garden since everyone thinks doing business is the key to get quick success. But they are forgetting that agriculture is the very basic to what every human being needs. Also, the people who are taking good care of their garden, are not always there for their garden, meaning they might get stuck in a busy schedule and may not be able to see their garden for some time. So, we have proposed this model so that it helps the people do their work and alongside they can take care of their farm by sitting on their table and the result will be better than before. Here we have a web page idea by G. Prasanna, S. Parvatham, and D. Krishna [2], which helps us view the sensor data on our phone and laptop.

Here in this system, the moisture in the air which helps us to know whether it is raining or not and the moisture in the soil helps us to know the exact amount of water in the soil, or if the soil is dry or wet, if the soil is dry we can water the plants and enhance the growth of plants. Sometimes what happens is, the soil looks like it is wet from outside but it is not from inside. Here we can know the moisture of the soil and water the land as per requirements.

People are getting busy as the world is growing too quick to get alongside, we need to do multi-tasking for that we have a local server web application which will help to know the exact information in your laptop or mobile phone and get easy access to the data and the motor control. Here we can send the data which will help you know whether is your farm doing good or not.

EXISTING SYSTEM

The existing system in place is pretty much farmer’s experience and gut feeling. The field is irrigated by seeing the colour change in the leaf of a plant or by judging the dryness of the soil with naked eyes. The canal doors are opened and water is let into the field. There is no definitive time allocation for how long the water is to be supplied and level of moisture in the soil to be maintained while irrigating. The existing method is unscientific and entirely depends on the farmer’s experience and level of expertise on the subject matter.

PROPOSED SYSTEM

The proposed smart system provides accurate information of the irrigation status with the help of sensors. The input is collected from different areas. Rain Projection API is used to collect information about possibility of rain. Although the proposed Rain Projection API can be replaced with a rain sensor and humidity sensor for generating data at the local level, this replacement increases the cost of manufacturing and also would not provide any forecasting data. Soil Moisture Sensor collects information about the moisture level in the soil in real time. Temperature sensor collects information about the temperature of the area it is placed in. Hence, it can be placed outside the soil to measure current environment temperature or inside the soil to measure current soil temperature. The environment temperature sensor can be replaced with a Temperature API for large fields which will provide aggregate temperature of the environment. There are APIs that predict soil temperature via satellite imaging but soil temperature depends on different factors like moisture level and shades froma plant which leads to inaccurate data. Hence, a temperature sensor inside a soil is recommended for measuring soil temperature. These collected inputs are then sent to the microcontroller which processes this information and generates outputs. The outputs are projected on a smart phone. Figure 1 shows the block diagram of the proposed system.

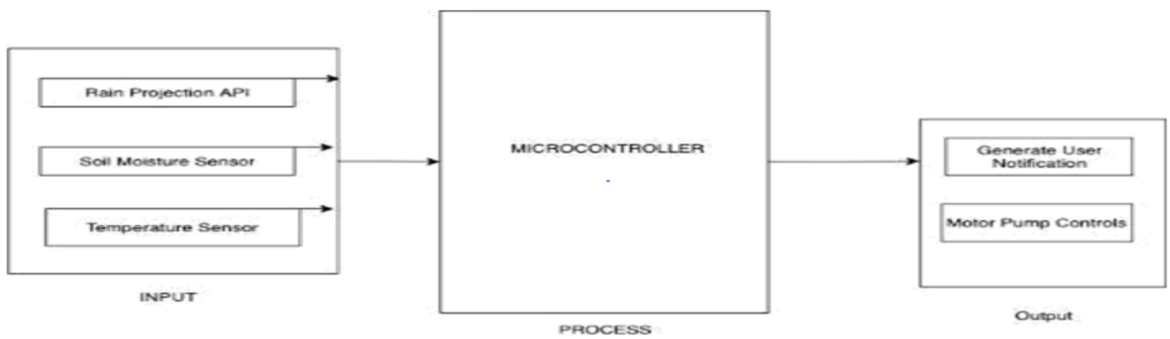

Fig 1: Block Diagram of Proposed System.

WORKING PRINCIPLE

In the Figure 2 which shows the work-flow diagram, when the system starts, it reaches the START state where it first collects data from the soil moisture sensor. The microcontroller analyses the data and determines whether the soil is wet or dry. If the soil is wet, it notifies the user and proceeds to the STOP state by checking if the motor is on or off. If the motor is off, it proceeds to the STOP state and if the motor is on, it proceeds to turn the motor off and loops back to notify the user. If the soil is dry it proceeds to the next state. In the next state, the microcontroller determines the possibility of rain by collecting data from the rain sensor. If the data from the rain sensor is above threshold, then the system proceeds to STOP state by performing necessary operation like notifying user and determining if the motor is ON or OFF. If the data from the rain sensor is below threshold, then the system turns the motor on. When the motor is turned on, the system goes into WAITING state where it waits for a certain period of time and loops to the top of the flowchart until the soil is wet or it is raining.

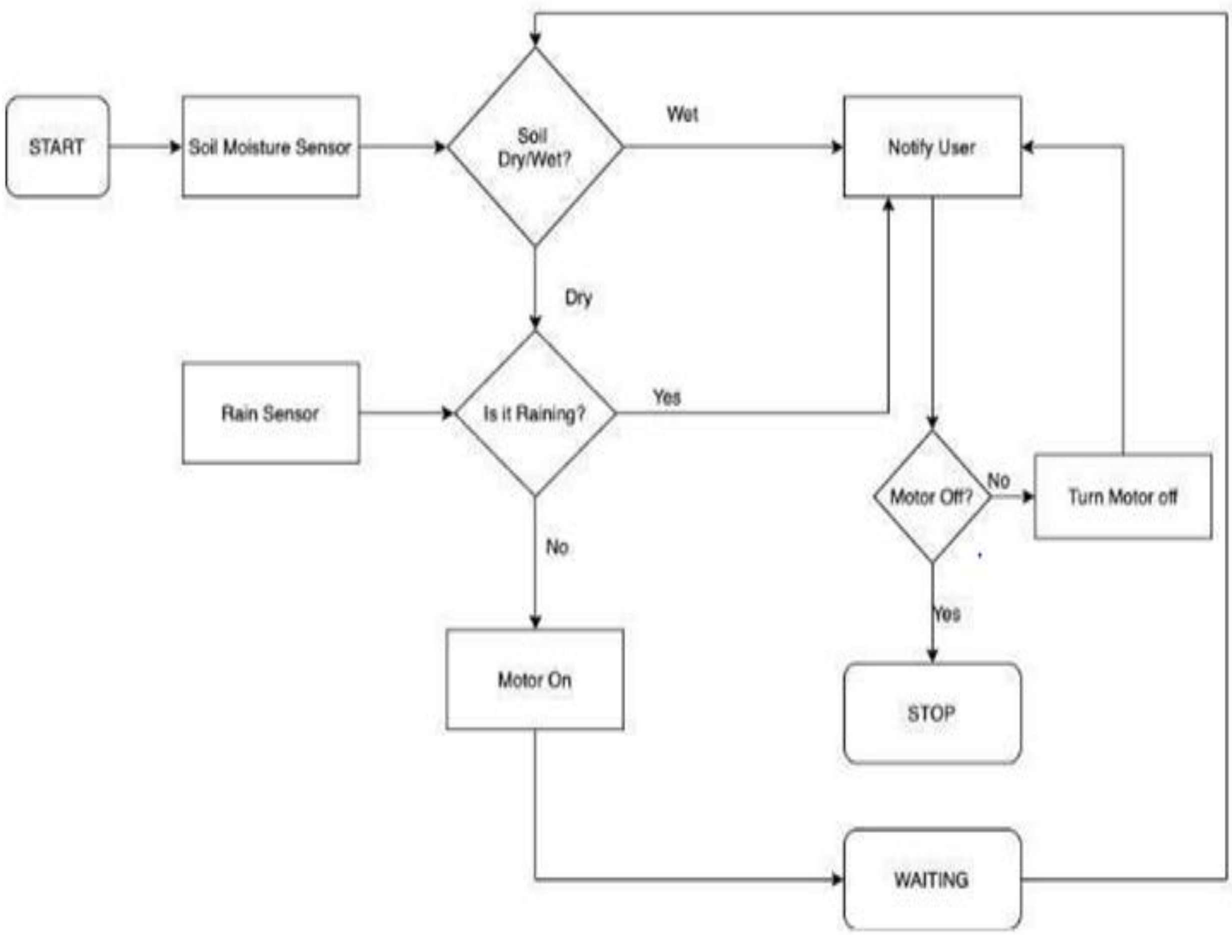

Fig 2: Work-flow diagram

COMPONENTS USED

Soil Moisture Sensor

In the Figure 3 which shows the Soil Moisture Sensor, there are two components combined together to measure the soil moisture level. The first component is a two-legged lead that goes inside the soil. The soil moisture sensor used here is “RC-A-4079” from “Robocraze”. This first component is then connected to the second component which is an amplifier. The second component is then connected to the microcontroller. This type of soil moisture sensor measures the moisture level of the soil by measuring the electric resistance and dielectric constant. The output generated with these components can either be analogue or digital. This all depends upon the use case and project requirement. The analogue output is a number between 0 and 1024 where larger obtained number signifies dry soil and lesser obtained number signifies wet soil. With analogue output exact dryness or wetness of the soil can be known. However, with digital output, we either know if the soil is in dry or wet state. This can be projected on an onboard LED of the microcontroller by Programming the sensor.

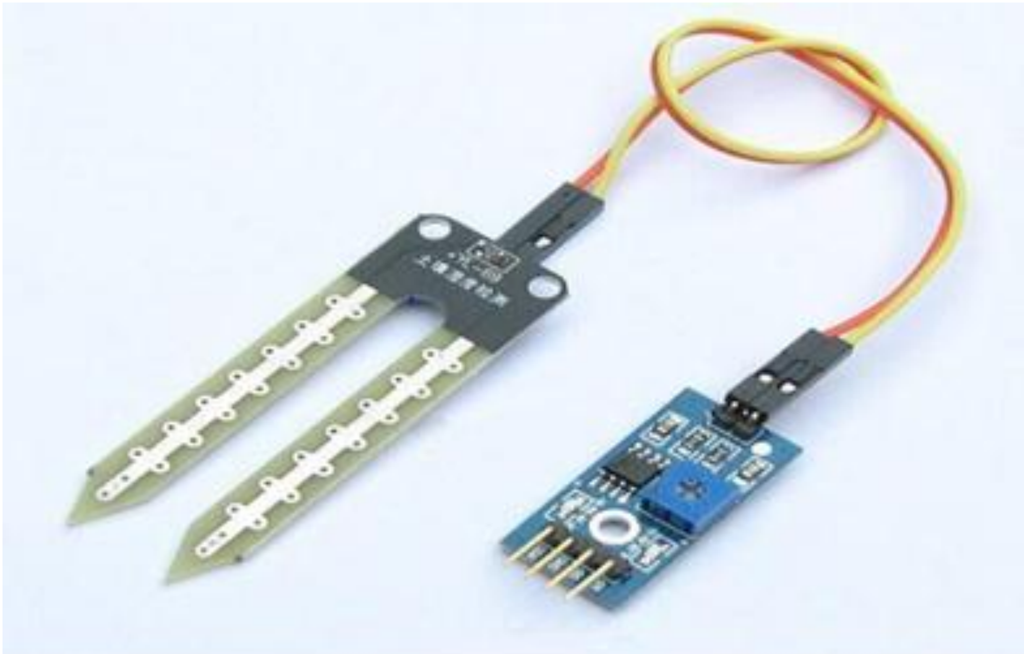

Fig 3: Soil Moisture Sensor

Temperature Sensor

Figure 4 shows the Temperature Sensor Temperature, which are variable resistors that change their resistance with temperature. In the proposed irrigation system, LM 35 DZ temperature sensor is used. Its operating temperature is −55° and + 150° C and operates between 4 V and 30 V. These sensors linearly produce an output voltage of

10 mV per degree centigrade change intemperature. These sensors are made from a semiconducting material that has been heated and compressed to form a temperature sensitive conducting material which contains charge carriers that allow current to flow through it. High Temperature causes the semiconducting material to release more charge carriers and hence temperature is detected.

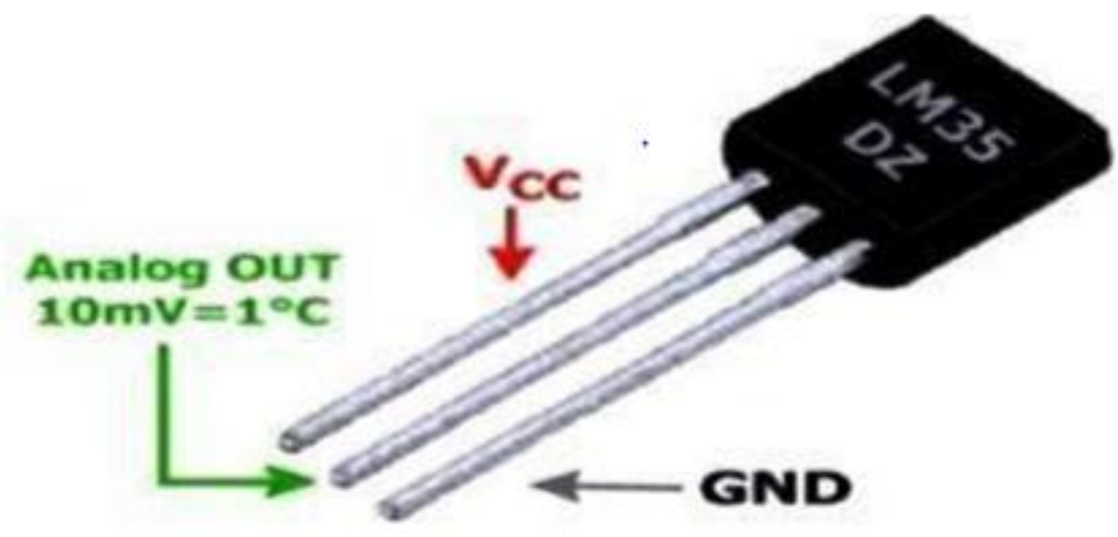

Fig 4: Temperature Sensor

International Journal of All Research Education and Scientific Methods (IJARESM), ISSN: 2455-6211
Volume 9, Issue 7, July -2021, Impact Factor: 7.429, Available online at: www.ijaresm.com

Humidity Sensor

Figure 5 shows the Humidity sensor, which is an electronic device used to measure the humidity in the environment and convert those signalsto electrical signal. The amount of water vapor in the air can be measured

by adjusting the potentiometer in the sensor module and the value is inversely proportional to the amount of resistance at a fixed temperature. The humidity sensor used in this project is “DHT11 Digital Temperature Humidity Sensor Module”. Its operational range is 20% to 95% at a temperature range from 0 to 60 degree Celsius. Its operational voltage requirement is between 3.3 V to 5 V and is able to produce both analogue and digital output.

Fig 5: Humidity sensor

A microcontroller is an integrated circuit (IC) used to process the information or data received from other individual components connected to its input terminal. It processes these collected information or data via a microprocessor unit (MPU), on board volatile memory, input and output peripherals. These devices are optimized for embedded applications and have minimal requirements for memory and program length with no operating system and low software complexity. They usually contain general input/output pins or GPIOs. These pins are software configurable to either an input or output state. To information sent by the sensors is in analogue form. Since Microcontrollers are digital devices, an Analogue to Digital Converter is used in order to make it understandable to the microcontroller. A dedicated pulse-width modulation (PWM) block is used to control power converters, resistive loads and motors by the CPU. A Universal Asynchronous Receiver/Transmitter (UART) block is used to receive and transmit data over a serial line with very little load on the CPU.

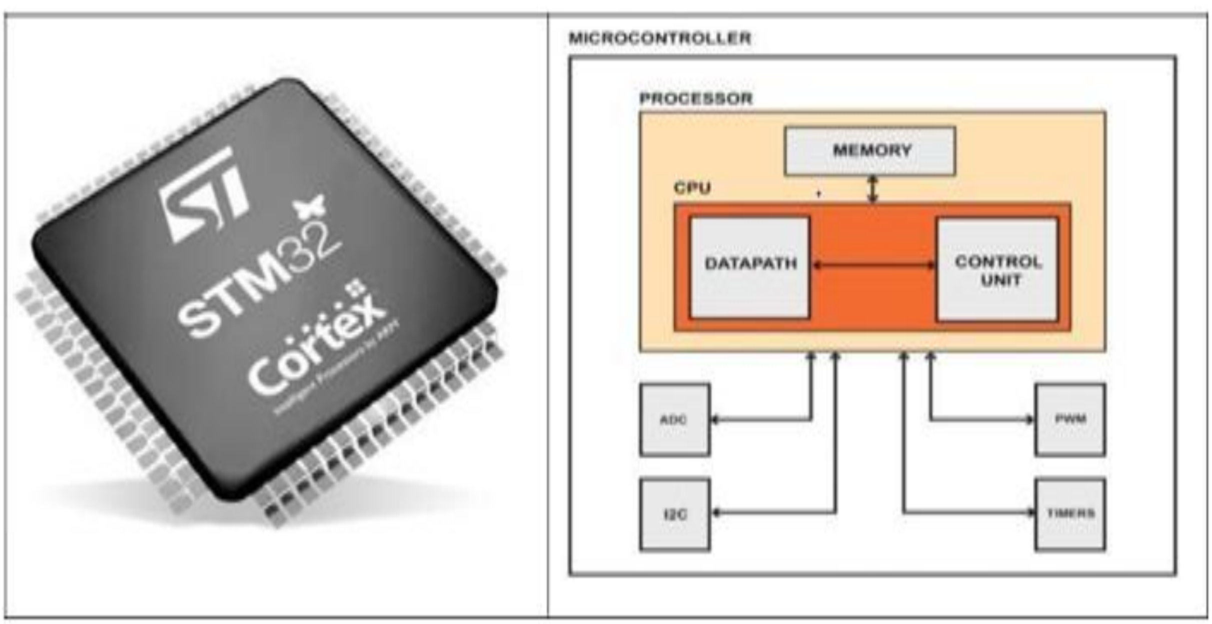

Fig 6: Microcontroller

CONCLUSION

The proposed smart irrigation system decides the necessity of irrigating the field according to the information received by the microcontroller from the attached sensors. It is obvious that the "Smart Irrigation System" consumes less water for production and saves more water for future use. The proposed system is a no-contact i.e., the farmer need not be involved with the operation of the system. The proposed system would also be non-invasive i.e., the operation of the system in no way affects the habitat of the plants, its growth and overall, it's lifespan. So, by using this model we can grow good crops by providing the field very less time so at same time we could take care of the field as well as another work which we want to do.

FUTURE SCOPE

The world is growing quick and to match up with it we need to be advanced and do multi-tasking, we need to earn more money but we need food too. Farming might seem easy and less time consuming but it is not that it takes a lot of care of the crops to grow some healthy crops and a small mistake could lead to very less crop growth or it might not grow at all. The proposed system can further be expanded with equipping other non-irrigation related sensor like "Soil pH Sensor" which measures the acidity in the soil, "Ammonium Sensor" and "Nitrogen Sensor" which measure the soil's ammonia and nitrogen level respectively to increase the effectiveness of the system.

REFERENCES

[1] G. Parameswaran and K. Sivaprasath, "Arduino Based Smart Drip Irrigation System using Internet of Things, "*International Journal of Engineering Science and Computing,* vol. 6, no. 5, pp. 5518-5521, May 2016.

[2] G. Prasanna, S. Parvatham, and D. Krishna, "Web Based Automatic Irrigation System Using Raspberry Pi Processor on Embedded Linux," *International Journal& Magazine of Engineering, Technology, Management and Research, vol. 3,* no. 10, pp. 1036-1039,2016.

[3] A. Tyagi, N. Gupta, D. J. P. Navani, R. Tiwari and A. Gupta, "Smart Irrigation System," *International Journal for Innovative Research in Science & Technology,* vol. 3, no. 10, pp. 9-12, 2017.

[4] S. Darshna, T. Sangavi, S. Mohan, A. Soundharya and S. Desikan, "Smart Irrigation System," *IOSR Journal of Electronics and communication Engineering" (IOSR-JECE),* vol. 10, no. 3, pp. 32-36, 2015.

[5] G. Ravikumar, T. V. Gopal, V. Sridhar and G. Nagendra, "Smart Irrigation System,". *International Journal of Pure and Applied Mathematics,* vol. 119, no. 15, pp. 1155-1167, 2018.

IJARESM

ISSN: 2455-6211, New Delhi, India

International Journal of All Research Education & Scientific Methods

An ISO & UGC Certified Peer-Reviewed Multi-disciplinary Journal

Certificate of Publication

Kunal Jayswal

Student, Department of Computer Science and Engineering, Dr. Ambedkar Institute of Technology, Bangalore

TITLE OF PAPER

Smart Irrigation Model Using IOT

has been published in

IJARESM, Impact Factor: 7.429, Volume 9 Issue 7, July- 2021

Paper Id: IJARESM/July21

Date: 31-07-2021

Website: www.ijaresm.com
Email: editor.ijaresm@gmail.com

Authorized Signatory