K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

Smart Agriculture

Submitted at the end of semester IV in partial fulfilment of requirements

Of Bachelors in Technology in Computer Engineering

by

Deev Patel- 16010121808

Siddhant Parekh- 16010121822

Niyati Rolia- 16010121823

Diya Patel- 16010121824

Guide

Prof. Mansi Kambli
Prof. Pradnya Bhangle

Department of Computer Engineering
K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

Batch 2021 -2024
Certificate

This is to certify that the MINIPROJECT report entitled **Smart Agriculture** submitted by Deev Patel, Siddhant Parekh, Niyati Rolia, Diya Patel, at the end of semester IV of SY B. Tech are bona fide record for partial fulfilment of requirements for the degree of Bachelors in Computer Engineering of University of Mumbai

____

____

Guide Head of the Department

Date:
Place: Mumbai-77

Certificate of Approval of Examiners

We certify that this MiniProject report entitled **Smart Agriculture** is bona fide record of Mini project work done by Deev Patel, Siddhant Parekh, Niyati Rolia, Diya Patel, during semester IV. This Mini project work is submitted at the end of semester IV in partial fulfilment of requirements for the degree of Bachelors in Technology in Computer Engineering of University of Mumbai.

Internal Examiner 1

Internal Examiner 2

Date:

Place: Mumbai-77

DECLARATION

We declare that this written report submission represents the work done based on our and / or others' ideas with adequately cited and referenced the original source. We also declare that we have adhered to all principles of intellectual property, academic honesty and integrity as we have not misinterpreted or fabricated or falsified any idea/data/fact/source/original work/ matter in my submission.

We understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

Signature of the Student **Roll No. 16010121808**	**Signature of the Student** **Roll No. 16010121822**
Signature of the Student **Roll No. 16010121823**	**Signature of the Student** **Roll No. 16010121824**

Date:

Place: Mumbai-77

Table of Contents:

Chapter 1:

Chapter 2:

Chapter 3:

Chapter 4:

List Of Figures

Chapter 1

Introduction

Agriculture is considered as the basic of life for the human species as it is the main source of food grains and other raw materials. It plays vital role in the growth of country's economy. Growth in agricultural sector is necessary for the development of economic condition of the country. Farmers are the backbone of our country[5]. The farmers should be given first priority in all aspects. Unfortunately, many farmers still use the traditional methods of farming which results in low yielding of crops and fruits. Also, there are crops which require water early in morning, while some crops require water at low temperature i.e., during the mid-day. As a result, the farmer has to walk miles to his farm to turn on the motors every time.

Application of IOT in agriculture could be a life changer for humanity and the whole planet[6]. IOT refers to any system of physical devices that receive and transfer data over wireless networks without human intervention[1]. The Internet of Things (IOT) has the potential to transform the ways we live in the world;[8] we have more-efficient industries, more connected cars, and smarter cities, all these as components of an integrated IOT system.[7] The demand for more food has to meet overcoming challenges such as, rising climate change, extreme weather conditions and environmental impact that results from intensive farming practices.

The purpose of this project is to monitor and control the farm appliances using Mobile phone. This can be achieved by use of temperature, humidity, soil moisture, and sound sensor, which senses the temperature, humidity, and soil moisture content as well as presence of pest and its output will be available to the farmer anytime-anywhere in his mobile phone. The main objective of project is to send an alert to farmer regarding the status of motors in ON and OFF condition which will be done. Also, the owner of the farm or the farmer himself can anytime view the temperature, humidity, and soil moisture readings along with which motor is ON or OFF in a graph format with the help of database.

Problem Statement

Agriculture is an important activity which directly and indirectly engages 70% of the population in India.[1] Farmers have to face lot of problems if there is any damage to cultivation in farm. Devices in farm are located at distant locations, operating these devices becomes very difficult since farmer has to go from one place to another. The farm fields of farmers are situated miles away from his home, so days and nights travelling to switch ON and OFF the motor and other appliances becomes a burdensome task to the farmer. Crops are affected if appliances are operated for irrigation when not necessary. Even crops are damaged if they are not irrigated according to the crop requirement and surrounding environmental conditions. Also, water is wasted if irrigation is done when not required. With the help of this cloud based smart agriculture the surrounding environmental conditions such as temperature, humidity, soil moisture content and presence of pest will be provided by the sensor and then user can irrigate the crops and control farm appliances from anywhere easily. The user can also have a glance on the data stored in database to know which devices were operated when in the farm and then decide the further actions accordingly. Controlling the devices in farm that too wirelessly is the main motive of our project. Our system is designed especially for the farmers by keeping in mind their budget, therefore this system is very budget friendly as well as user friendly to use.

Proposed System

The proposed project presents the idea of controlling farm appliances wirelessly. The system uses wireless module which is based on cloud to operate devices located at distant locations in farm remotely without visiting the site. Whenever the user sends the request for getting the information regarding surrounding environmental conditions, the temperature and humidity sensor DHT11 and soil moisture sensor captures the surrounding temperature, humidity, and soil moisture content. Then those readings are forwarded to ThingSpeak for analysis and storing for user reference. This captured temperature, humidity, and soil moisture content are then displayed in form of graph to user. Depending on received data the user decides whether the appliances should be operated or not. If the particular appliance needs to be turned ON or OFF the user sends the corresponding request which is programmed for turning that device ON or OFF. That alert is then forwarded to Raspberry Pi and required action is performed. The status of the device whenever it is turned ON or OFF is also received by the user. If required, the user can also monitor current status of all devices in the farm by sending corresponding request that is programmed for it. A user can view the database for historic as well as current temperature, humidity, and soil moisture readings, devices operated by user along with timestamp, if necessary, to carry out further operations. By this database user can also get to know the activities carried out in the farm.

Chapter 2

Literature Review

In the existing manual irrigation system, the efforts required by a farmer are more and the throughput is less. If timely irrigation is not done, crops can get damaged. This can happen when the farmer is away from the farm. The new scenario of decreasing water, drying up of rivers and tanks, unpredictable environment, present an urgent need of proper utilization of water. To cope up with this use of temperature and moisture, sensors are placed at suitable locations for monitoring the crops. After research in the agricultural field, researchers found that the yield of agriculture is decreasing day by day.[8] However, use of technology in the field of agriculture plays an important role in increasing the production as well as in reducing the man power. Some of the research attempts are done for betterment of farmers that provide systems which use technologies helpful for increasing the agricultural yield.[5] Therefore, we have developed a system that will aid all the problems that a farmer faces.

This project proposes a low cost and efficient temperature, humidity, soil moisture, and acoustic sound sensor connected to Raspberry Pi for acquiring humidity, temperature, and soil moisture content from various locations of field. The farmer can send requests to turn ON or OFF the water motor from anywhere. Additionally, the farmer can also see the past and current temperature, humidity, and soil moisture reading simply by a graph ThingSpeak. It proposes an idea about how automated system was developed to optimize the agriculture tradition of farmers.

1) **"Microcontroller Based Controlled Irrigation System for Plantation" SR Kumbhar et.al, IMECS VOl II, 2013[5]**

- Proposed a microcontroller based controlled remote irrigation system developed for agricultural plantation, which is placed at the remote location and required water is provided for plantation when humidity of the soil goes below the set point value.
- If the set point value is high then the motor is turned on, otherwise the motor is turned off.
- This provides right amount of water at right time. If plants get water at the proper time, then it helps to increase the production from 25% to 30%
- This system can be used to irrigate very large areas as it only needs to divide the whole land into number of sectors and the single microcontroller can control the whole process.
- It is possible to create the various parameters through the controller programming developed in assembly level language and necessary change in action can be implemented in the software

2) **"IoT based Smart Agriculture" Nikesh Gondchawar, Prof. Dr. R. S. Kawitkar [9]**

- The newer scenario of decreasing water tables, drying up of rivers and tanks, unpredictable environment presents an urgent need of proper utilization of water. To cope up with this use of temperature and moisture sensor at suitable locations for monitoring of crops is implemented in.

- An algorithm developed with threshold values of temperature and soil moisture can be programmed into a microcontroller-based gateway to control water quantity.
- The technological development in Wireless Sensor Networks made it possible to use in monitoring and control of greenhouse parameter in precision agriculture.
- A remote sensing and control irrigation system using distributed wireless sensor network aiming for variable rate irrigation, real time in field sensing, controlling of a site-specific precision linear move irrigation system to maximize the productivity with minimal use of water

Chapter 3:

System Architecture

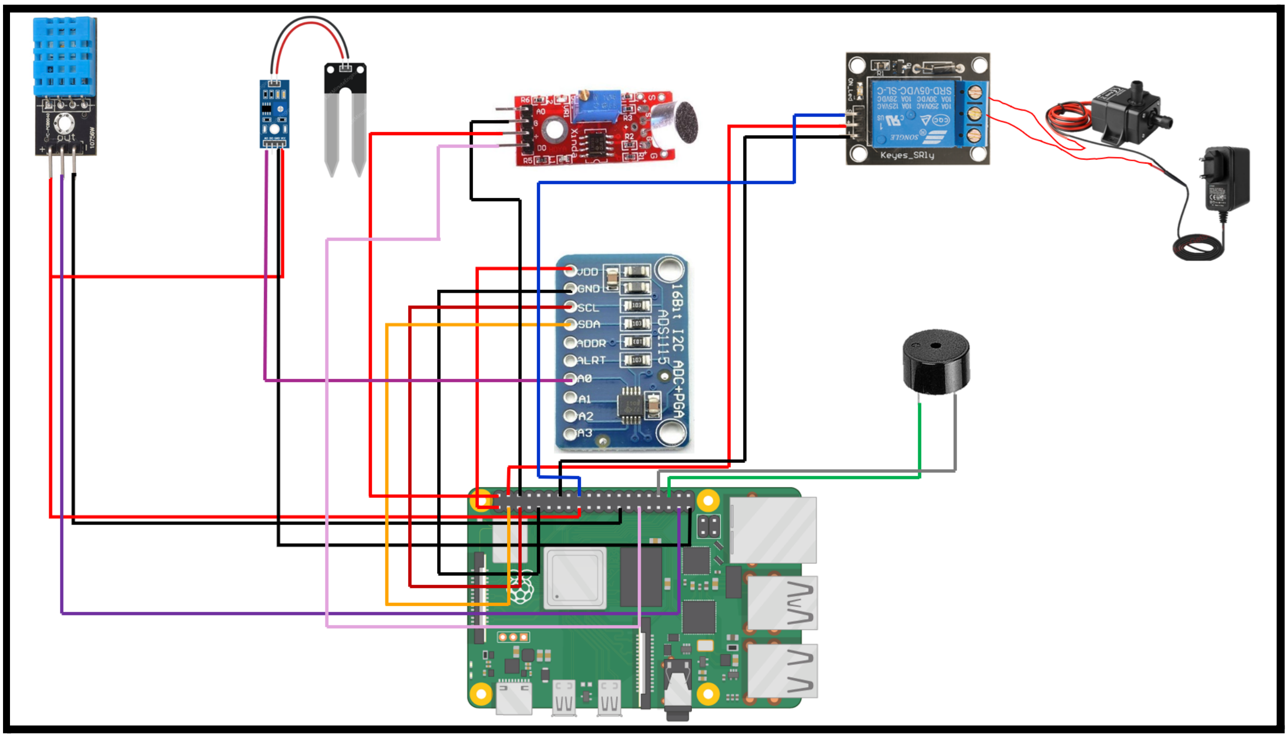

Fig 3.1

- This entire interfacing makes a fully functional system to operate farm appliances based on temperature, humidity, and soil moisture content received by the user and monitor their status.
- Sound sensor collects data about presence of pest and relay circuit controls motor.
- The data is also stored on ThingSpeak Application for further analysis.

Working

When the power supply is switched on, the Raspberry Pi gets initialized. Wait for some time till the circuit gets network and blinks red after 3 seconds. The Raspberry Pi communicates with the ThingSpeak using Wi-Fi. The temperature, soil moisture and sound sensor is connected to Raspberry Pi via jumper wires which captures the temperature, humidity, and soil moisture of the environment and sound sensor collects data about pests availability and sends it to Raspberry Pi. It then gives temperature, humidity, soil moisture and pest availability information captured by the sensor to the user in form of graph. According to received information the user decides which appliance needs to be operate and sends the corresponding request programmed for that appliance to the ThingSpeak. ThingSpeak then sends that request to Raspberry Pi which processes it and automatically switches the appliance ON or OFF. The user can monitor the status of appliances in the farm by sending the corresponding request to the ThingSpeak which then collects sensors' data through Raspberry Pi and the message containing current status of all the appliances in the farm is sent to the user. The current and historic information regarding temperature, humidity, soil moisture and pest availability readings, operated devices is available on ThingSpeak for users to analyse and view it.

Hardware Requirements

- Raspberry Pi
- Relay.
- DHT11 Temperature and Humidity sensor.
- Soil moisture sensor
- Acoustic sound sensor
- Buzzer
- Jumper wires.
- Motor.
- USB power cable
- SD Card
- HDMI wire

Raspberry Pi

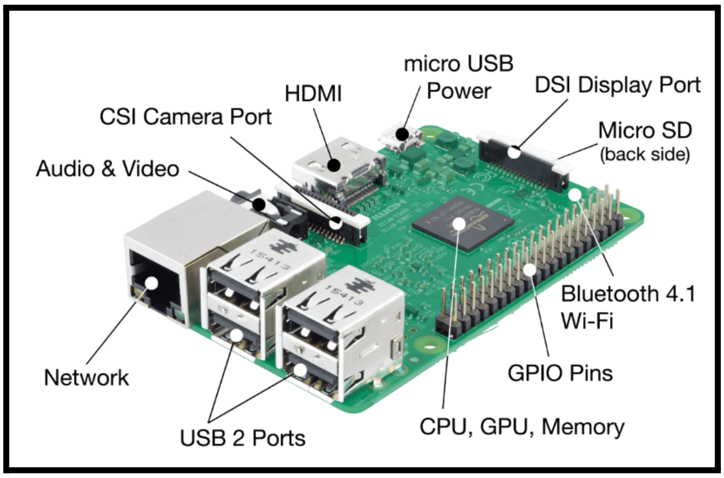

Fig 3.2

Raspberry Pi is a low cost, **credit-card sized computer** that plugs into a computer monitor or TV and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It is capable of doing everything you would expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

The Raspberry Pi 3 Model B is the earliest model of the third-generation Raspberry Pi. It replaced the Raspberry Pi 2 Model B in February 2016. Raspberry Pi 3 Model B was released in February 2016 with a 1.2 GHz 64-bit quad core ARM Cortex-A53 processor, on-board 802.11n Wi-Fi, Bluetooth, and USB boot capabilities.

Technical Specifications: -

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 100 Base Ethernet
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

Relay

Fig 3.3

Power relay module is an electrical switch that is operated by an electromagnet. The electromagnet is activated by a separate low-power signal from a micro controller. When activated, the electromagnet pulls to either open or close an electrical circuit.

A simple relay consists of wire coil wrapped around a soft iron core, or solenoid, an iron yoke that delivers a low reluctance path for magnetic flux, a movable iron armature and one or more sets of contacts. The movable armature is hinged to the yoke and linked to one or more set of the moving contacts. Held in place by a spring, the armature leaves a gap in the magnetic circuit when the relay is de-energized. While in this position, one of the two sets of contacts is closed while the other set remains open.

When electrical current is passed through a coil, it generates a magnetic field that in turn activates the armature. This movement of the movable contacts makes or breaks a connection with the fixed contact. When the relay is de-energized, the sets of contacts that were closed, open and breaks the connection and vice versa if the contacts were open. When switching off the current to the coil, the armature is returned, by force, to its relaxed position.

This is a simple single-channel relay module. Connect power and then switch the relay on by applying a low signal to the "IN" pin. The module has three screw terminal connections connected to the relay that include

- Common
- normally open (N.O.)
- normally closed (N.C.)

Technical Specifications: -

- Supply Voltage 3.75 to 6 V
- Supply Current with Relay De-Energized 2 mA
- Supply Current with Relay Energized 70 to 72 mA
- Input Control Signal Active Low
- Input Control Signal Current 1.5 to 1.9 mA
- Relay Max Contact Voltage 250 VAC or 30 VDC
- Relay Max Contact Current 10

Interfacing Relay with Raspberry Pi

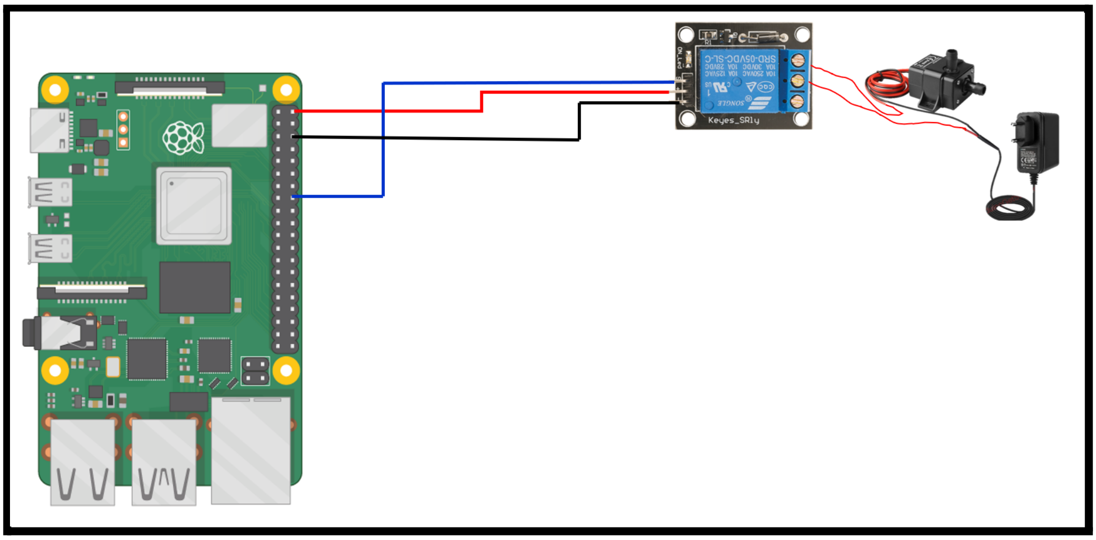

Fig 3.4

- The 1-channel relay is connected to the Raspberry Pi to operate the farm appliances.
- The VCC pin of Relay is connected to 5V VCC pin of Raspberry Pi
- The digital input (IN) pin of relay is connected to pin 7 of Raspberry Pi
- The GND pin of relay is connected to GND pin of Raspberry Pi.

Temperature Sensor

A temperature sensor is a device that collects the data about temperature from a particular source and converts the data into understandable form for a device or an observer. It measures the temperature of its environment and converts the input data into electronic data to record, monitor, or signal temperature changes.

DHT11 Sensor

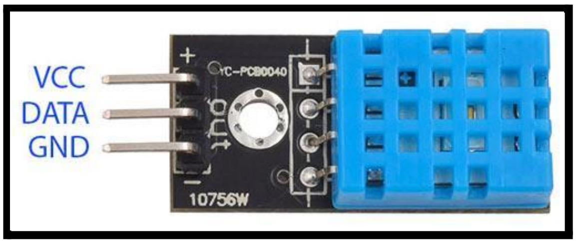

Fig 3.5

The DHT11 is a basic, ultra-low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin (no analog input pins needed). It is fairly simple to use but requires careful timing to grab data.
The DHT11 is a commonly used Temperature and humidity sensor. The sensor comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data.
The sensor can measure temperature from 0°C to 50°C and humidity from 20% to 90% with an accuracy of ±1°C and ±1%.
Technical Specifications: -

- Operating Voltage: 3.5V to 5.5V
- Operating current: 0.3mA (measuring) 60uA (standby)
- Output: Serial data
- Temperature Range: 0°C to 50°C
- Humidity Range: 20% to 90%
- Resolution: Temperature and Humidity both are 16-bit
- Accuracy: ±1°C and ±1%

Interfacing DHT11 with Raspberry Pi

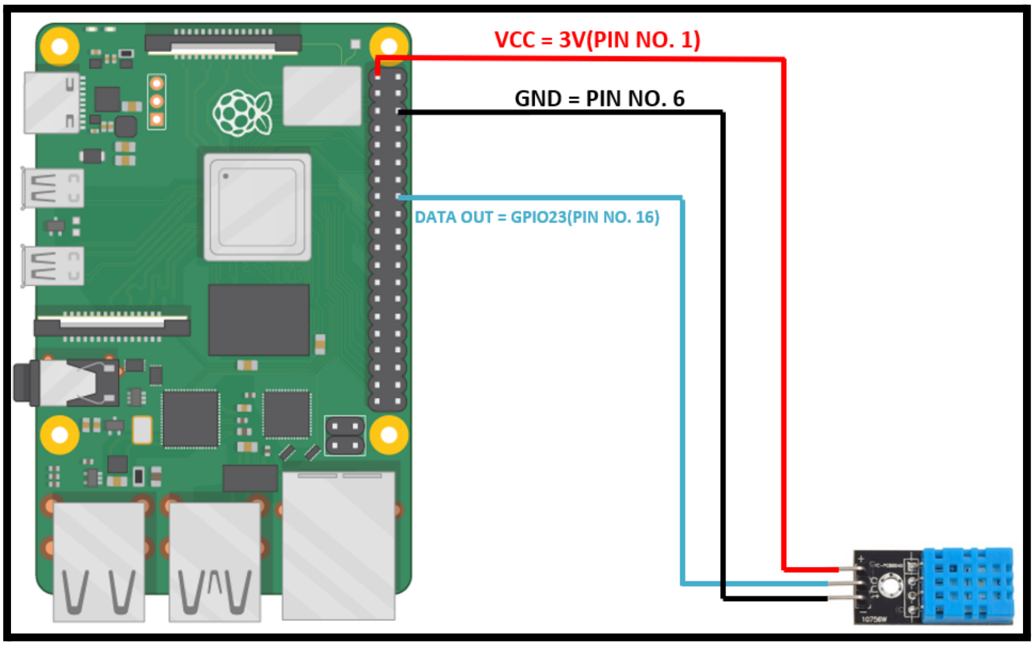

Fig 3.6

- The DHT11 temperature and humidity sensor has three pins VCC, Data, and GND.
- The first pin VCC pin is connected to 3V pin of Raspberry Pi.
- The data pin is connected to pin 11 of Raspberry Pi
- GND pin is connected to GND pin of Raspberry Pi.
- The temperature and humidity readings captured by sensor are then sent through data pin to Raspberry Pi board.

ADS1115

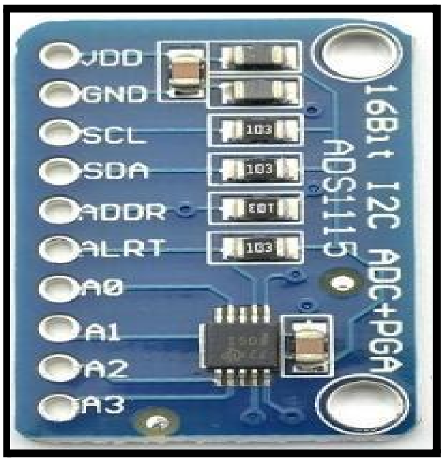

Fig 3.7

The main purpose of the ADS1115 16-Bit 4-Channel ADC is to serve as a low-consumption, powerful and accurate analogue-to-digital converter that can be adapted to perform different conversions.

The supply ranges between 2 and 5.5 V and allows you to adapt your chosen microcontroller, for example an Arduino or Raspberry Pi. Regarding the transmission channels, these can be used as single-ended input channels (4) or differential channels (2). Finally, the I2C interface allows you to select up to 4 slave addresses, and therefore to connect 4 16-Bit ADCs to your microcontroller simultaneously!

This device has a stated typical accuracy of 0.01% (but it has a maximum accuracy of 0.15%). This accuracy includes all sources of error (voltage reference, Gain error, offset and noise).

Technical specifications: -

- Precision: 16 bits
- Operating voltage: 2–5.5 V
- Consumption: 150 µA in continuous mode (single-shot mode: automatic shut-down)
- Programmable data rate: 8SPS to 860SPS (samples per second)
- Low-drift reference voltage
- Internal oscillator
- Internal programmable gain amplifier
- I2C interface: up to 4 pin-selectable addresses
- 4 single-ended channels or 2 differential channels
- Programmable comparator

Soil Moisture Sensor

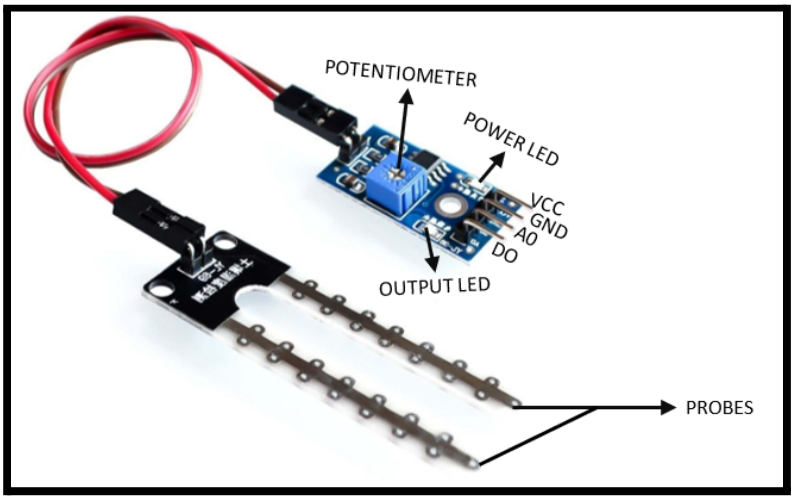

Fig 3.8

When you hear the term ‘smart agriculture’, one of the things that comes to your mind is a system that measures soil moisture and irrigates crops automatically. With this type of system, you can water your crops only when needed and avoid over-watering or under-watering.

The working of the soil moisture sensor is pretty straightforward. The fork-shaped probe with two exposed conductors, acts as a variable resistor (just like a potentiometer) whose resistance varies according to the water content in the soil.

This resistance is inversely proportional to the soil moisture:

The more water in the soil means better conductivity and will result in a lower resistance.

The less water in the soil means poor conductivity and will result in a higher resistance.

The sensor produces an output voltage according to the resistance, which by measuring we can determine the moisture level.

Technical Specifications: -

- Input Voltage 3.3 – 5V
- Output Voltage 0 – 4.2V
- Input Current 35mA

Output Signal Both Analog and Digital

Interfacing soil moisture sensor with Raspberry Pi

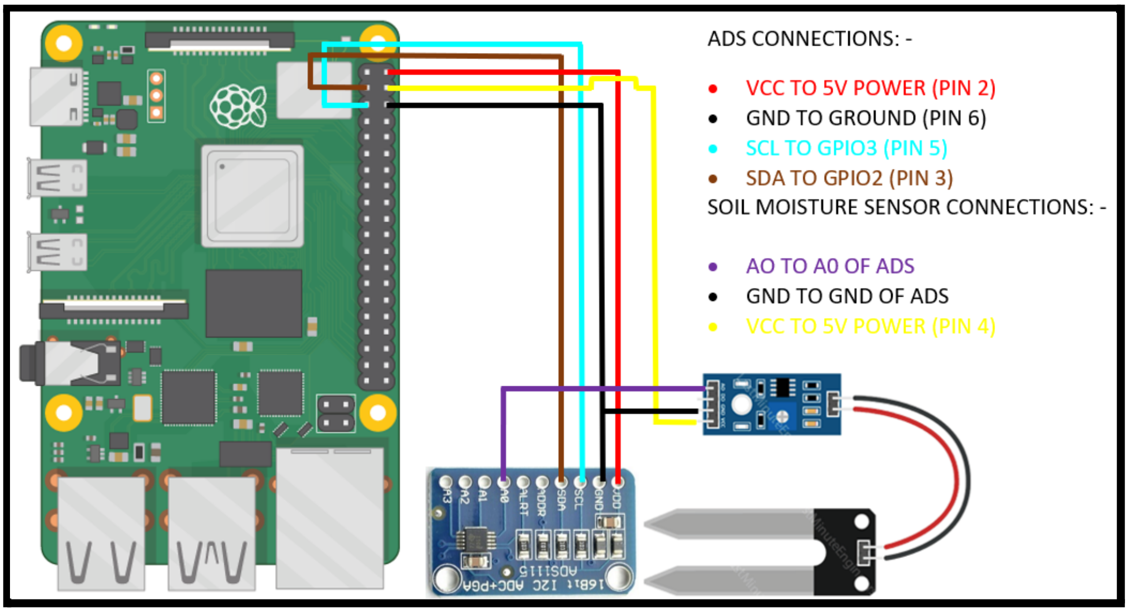

Fig 3.9

- The soil moisture sensor has four pins VCC, DO, AO, and GND.
- The first pin VCC pin is connected to 5V pin of Raspberry Pi.
- The AO pin is connected to A0 pin of ADS1115
- GND pin is connected to GND pin of ADS1115
- VCC pin of ADS1115 is connected to 5V pin of Raspberry Pi
- GND pin of ADS1115 is connected to GND pin of Raspberry Pi
- SCL pin of ADS1115 is connected to GPIO3 pin of Raspberry Pi
- SDA pin of ADS1115 is connected to GPIO2 pin of Raspberry Pi

ACOUSTIC SOUND SENSOR

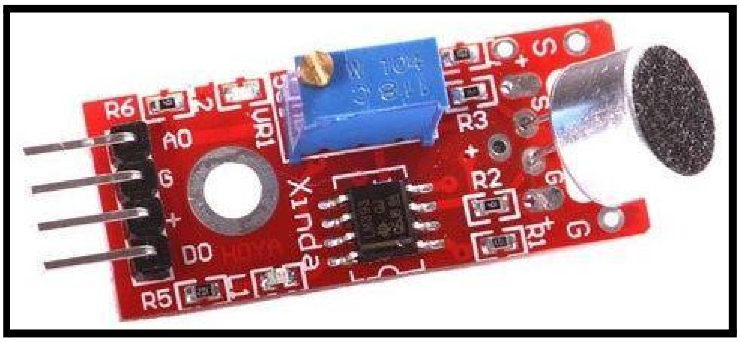

Fig 3.10

An acoustic sound sensor is an electronic device that can measure sound levels. They are called acoustic sound sensors because their detection mechanism is a mechanical (or acoustic) wave. When an acoustic wave (input) travels through a certain material or along the surface of a material, it is influenced by the different material properties and obstacles it travels through. Any changes to the characteristics of this travelling path affect the velocity and/or amplitude of the wave. These characteristics are translated into a digital signal (output) using transducers. These changes can be monitored by measuring the frequency or phase characteristics of the sensor. Then these changes can be translated to the corresponding physical differences being measured.

Sound detection sensor works similarly to our Ears, having diaphragm which converts vibration into signals. However, what's different is that a sound sensor consists of an in-built capacitive microphone, peak detector, and an amplifier (LM386, LM393, etc.) that's highly sensitive to sound.

With these components, it allows for the sensor to work:

1. Sound waves propagate through air molecules
2. Such sound waves cause the diaphragm in the microphone to vibrate, resulting in capacitance change
3. Capacitance change is then amplified and digitalized for processing of sound intensity

Technical specifications: -

- The range of operating voltage is 3.⅗ V
- The operating current is 4~5 mA
- The voltage gains 26 dB ((V=6V, f=1kHz)
- The sensitivity of the microphone (1kHz) is 52 to 48 dB
- The impedance of the microphone is 2.2k Ohm
- The frequency of microphone is16 to 20 kHz
- The signal to noise ratio is 54 dB

Buzzer

Fig 3.11

Buzzer is a sounding device that can convert audio signals into sound signals. It is usually powered by DC voltage. It is widely used in alarms, computers, printers and other electronic products as sound devices. It is mainly divided into piezoelectric buzzer and electromagnetic buzzer, represented by the letter "H" or "HA" in the circuit. According to different designs and uses, the buzzer can emit various sounds such as music, siren, buzzer, alarm, and electric bell.

Electromagnetic buzzer is composed of oscillator, solenoid coil, magnet, vibration diaphragm, housing, etc. When the power supply is switched on, the audio signal current generated by the oscillator passes through the solenoid coil, which generates a magnetic field. The vibration diaphragm periodically vibrates and sounds under the interaction of the solenoid coil and the magnet. The frequency of the general electromagnetic buzzer is 2-4 kHz.

Technical Specifications:

- Rated Voltage is 5 V
- Operating Voltage is 4~8 V
- Max Rated Current is ≤32 mA
- Min. Sound Output at 10cm is 85 dB
- Resonant Frequency is 2300 ±300 Hz
- Operating Temperature is -20°C to 45°C

Interfacing of sound sensor with Raspberry Pi

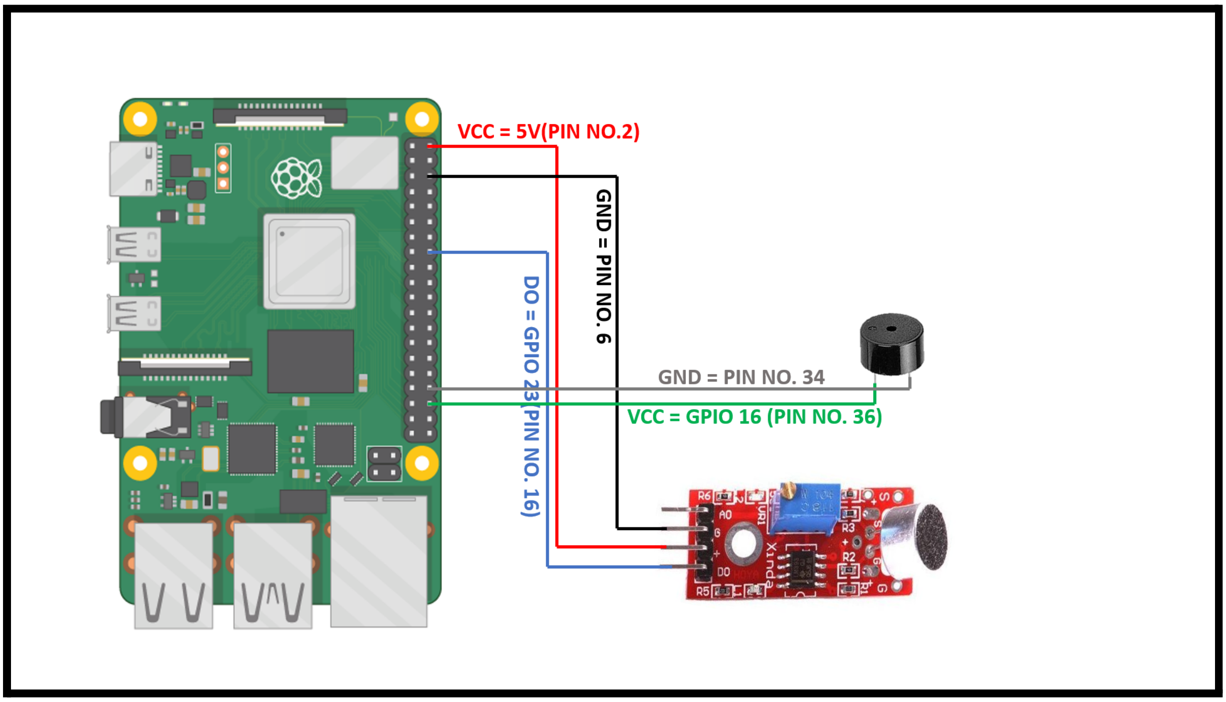

Fig 3.12

- The sound sensor has four pins VCC, DO, AO, and GND.
- The first pin VCC pin is connected to 5V pin of Raspberry Pi.
- GND pin is connected to pin no. 6
- DO pin is connected to GPIO 23

For Buzzer:

- GND pin is connected to pin no. 34
- VCC pin is connected to 3v3 pin of Raspberry Pi

Jumper wires

Fig 3.13

A jump wire (also known as jumper wire, or jumper) 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. There are different types of jumper wires. Some have the same type of electrical connector at both ends, while others have different connectors. Some common connectors are:

- Solid tips – are used to connect on/with a breadboard or female header connector. The arrangement of the elements and ease of insertion on a breadboard allows increasing the mounting density of both components and jumper wires without fear of short-circuits. The jump wires vary in size and color to distinguish the different working signals.
- Crocodile clips– are used, among other applications, to temporarily bridge sensors, buttons and other elements of prototypes with components or equipment that have arbitrary connectors, wires, screw terminals, etc.
- Registered jack (RJ) – are commonly used in telephone (RJ11) and computer networking (RJ45).
- RCA connectors – are often used for audio, low-resolution composite video signals, or other low-frequency applications requiring a shielded cable.

USB power cable

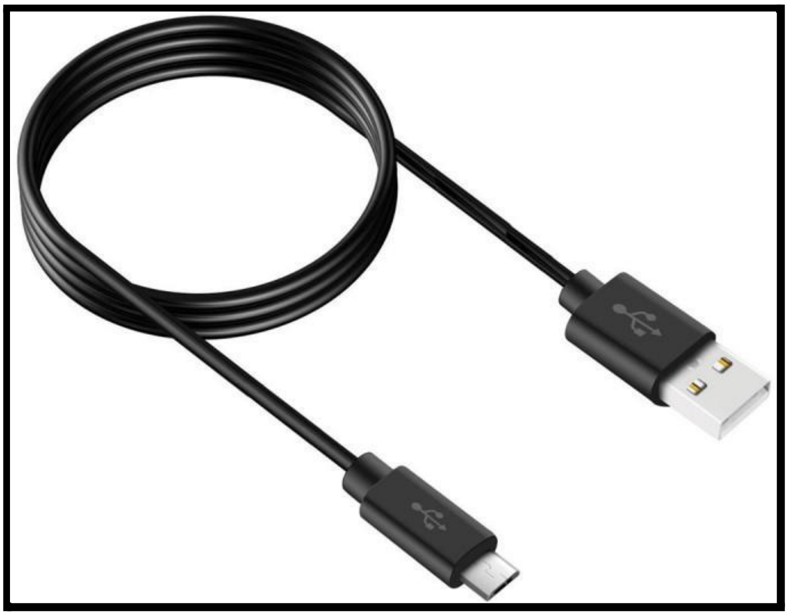

Fig 3.14

A power adapter that generates the 5-volt DC standard required by USB. The amperage varies, typically from .07A to 2.4A. The charger plugs into an AC outlet, and a USB cable plugs into the charger. USB ports on computers have an upper limit of 500 milliamps; however, USB chargers that come with cell phones and other devices handle one or more amps. The devices sense when they are connected to a computer versus the charger and regulate their current draw.

In the USB 1.0 and 2.0 specs, a standard downstream port is capable of delivering up to 500mA (0.5A); with USB 3.0, it moves up to 900mA (0.9A). The charging downstream and dedicated charging ports provide up to 1,500mA (1.5A). USB 3.1 bumps throughput to 10Gbps in what's called SuperSpeed+ mode, bringing it roughly equivalent with first-generation Thunderbolt. It also supports a power draw of 1.5A and 3A over the 5V bus. USB 3.2 does not change these aspects of the standard.

You can plug any USB device into any USB cable and into any USB port, and nothing will explode — and in fact, using a more powerful charger should speed up battery charging.

Motor

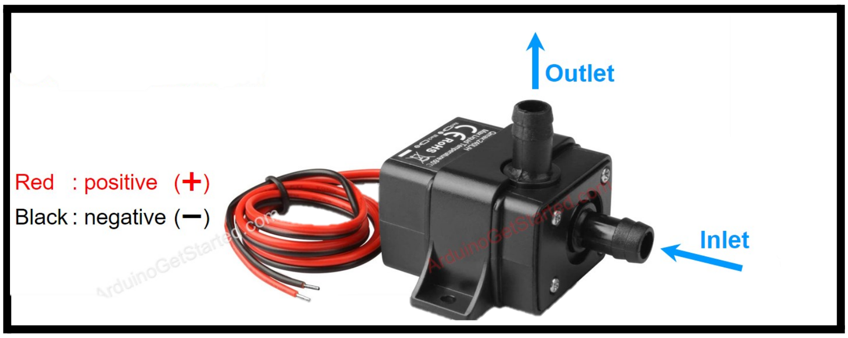

Fig 3.15

A DC motor is any of a class of rotary electrical motors that converts direct current electrical energy into mechanical energy. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current in part of the motor.
DC motors were the first form of motor widely used, as they could be powered from existing direct-current lighting power distribution systems. A DC motor's speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of current in its field windings. Small DC motors are used in tools and appliances. The universal motor can operate on direct current but is a lightweight brushed motor used for portable power tools and appliances. Larger DC motors are currently used in propulsion of electric vehicles, elevator, and hoists, and in drives for steel rolling mills. The advent of power electronics has made replacement of DC motors with AC motors possible in many applications.
A simple DC motor has a stationary set of magnets in the stator and an armature with one or more windings of insulated wire wrapped around a soft iron core that concentrates the magnetic field. The windings usually have multiple turns around the core, and in large motors there can be several parallel current paths. The ends of the wire winding are connected to a commutator. The commutator allows each armature coil to be energized in turn and connects the rotating coils with the external power supply through brushes.

SD CARD

Fig 3.16

Stands for "Secure Digital." SD is a type of memory card used to store data in portable electronic devices. Examples include digital cameras, video recorders, smartphones, and portable music players. SD cards are considered removable storage (instead of internal or external storage), since they can be inserted and removed from a compatible device.

The first SD cards were introduced in 1999. They used almost the same form factor as the existing Multimedia Card (MMC) format but were slightly thicker (2.1 mm vs 1.4 mm). The dimensions of a standard SD card are:

- 24 mm wide
- 32 mm tall
- 2.1 mm thick

Each SD card has a slanted upper-right corner to ensure the card can only be inserted one way. The left side of a card has a physical slider that prevents the card from being written (read-only) when moved to the LOCK position. The "secure" part of "Secure Digital" refers to its built-in DRM protection technology. The SD format supports Content Protection for Recordable Media (CPRM), which prevents protected content from being read from another storage device. Since its introduction in 1999, the SD card format has gone through several iterations.

Another version of SD, called microSD, was introduced in 2005. microSD cards are much smaller than SD cards, with dimensions of 11 mm x 15 mm x 1mm. Small devices like GoPros and smartphones have microSD slots to reduce the overall device size as much as possible. microSD cards can be used in standard SD card slots using a microSD-to-SD adapter

HDMI wire

Fig 3.17

HDMI stands for High-Definition Multimedia Interface, a standard for simultaneously transmitting digital video and audio from a source, such as a computer or TV cable box, to a computer monitor, TV, or projector. Originally developed by a consortium of electronics manufacturers, it has been widely adopted with almost all televisions and computer monitors supporting the interface.
The HDMI interface allows a port to send high-resolution digital video, theatre-quality sound, and device commands through a connector and down a single cable. There are several types of HDMI cable, each designed to support a video resolution and features in the HDMI specification.
HDMI connectors are available in three sizes:

- Standard,
- Mini and
- Micro.

Software Requirements

Raspbian OS

Fig 3.18

Raspberry Pi OS (formerly Raspbian) is a Debian-based operating system for Raspberry Pi. Since 2015, it has been officially provided by the Raspberry Pi Foundation as the primary operating system for the Raspberry Pi family of compact single-board computers. The first version of Raspbian was created by Mike Thompson and Peter Green as an independent project. The initial build was completed in June 2012.

User Interface: Raspberry Pi OS looks similar to many common desktops, such as macOS and Microsoft Windows. The menu bar is positioned at the top and contains an application menu and shortcuts to Terminal, Chromium, and File Manager. On the right is a Bluetooth menu, a Wi-Fi menu, volume control, and a digital clock.

Package Management

Packages can be installed via APT, the Recommended Software app, and by using the Add/Remove Software tool, a GUI wrapper for APT.

Block Diagram

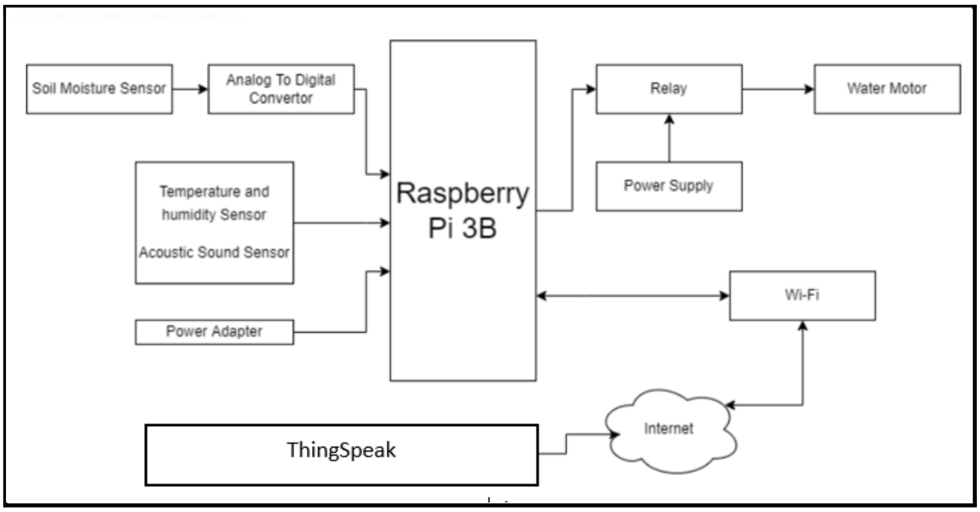

Fig 3.19

User using his computing device i.e., cell phone or PC connects to ThingSpeak cloud through Wi-Fi. On switching ON Raspberry Pi and connecting it to Wi-Fi, it also gets connected with ThingSpeak as well as all the sensors and relay.

Relay controls DC water motor as per the commands given by user through ThingSpeak which is processed by Raspberry Pi. Thus, irrigation could be done from distant place also.

All the sensors collect data and forwards it to ThingSpeak cloud via Raspberry Pi circuit which is provided in form of graph to the user. The sound sensor activates the buzzer if any pest is detected and each record of it is stored in cloud.

Timeline

WEEKS / ACTIVITIES	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LEARNING																
PLANNING																
INTERFACING																
CODE																
TESTING																
DEPLOYMENT																
DOCUMENTATION																

Fig 3.20

A schedule or a timetable above consists of list of events, tasks or actions undertaken to build this project. It shows the planned time period for each event or task to complete, and all the tasks are listed in a chronological order. All the tasks listed in above figure were planned and completed successfully according to the above timeline.

Flowchart

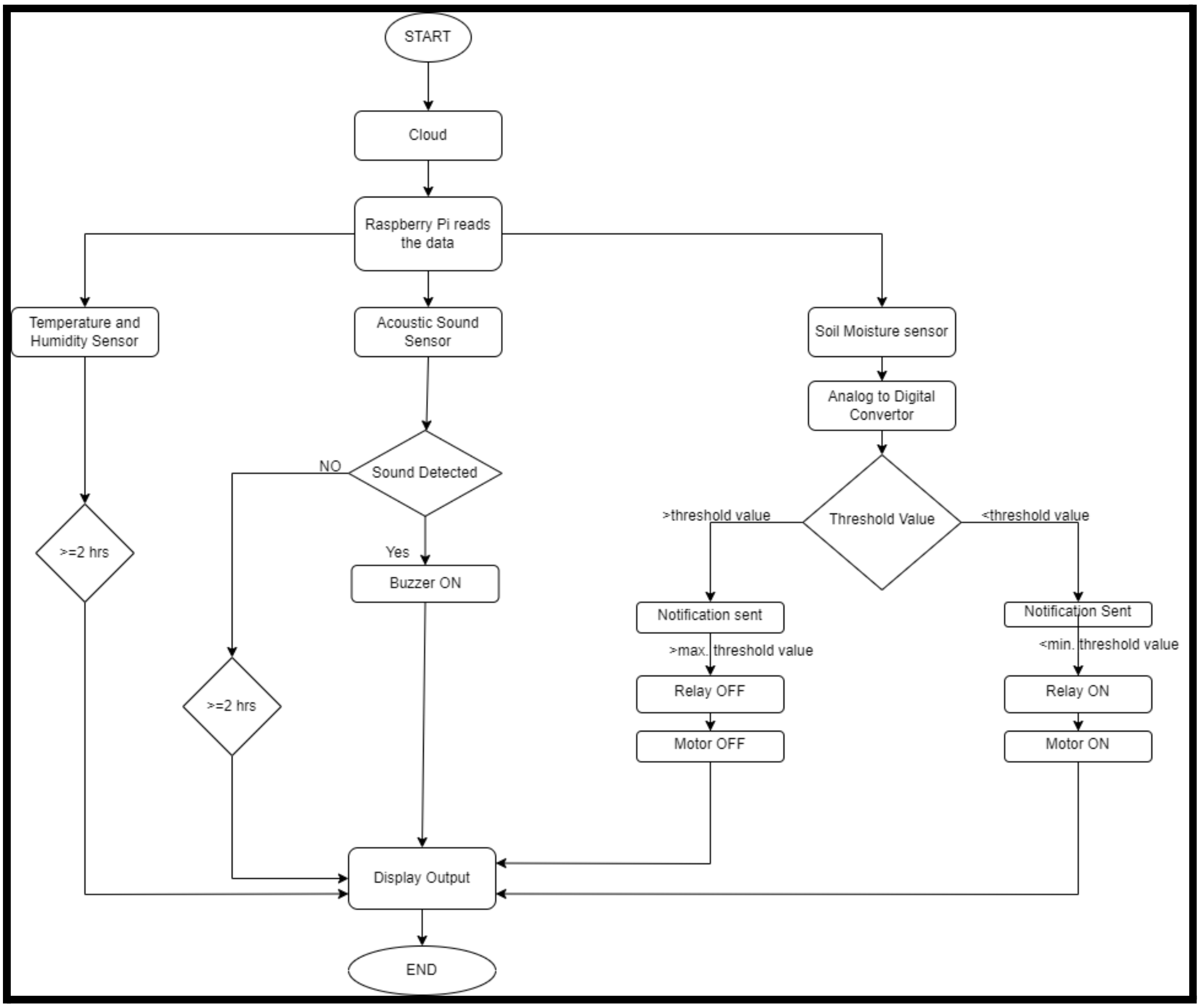

Fig 3.21

Program

```
import RPi.GPIO as GPIO
import time
import http.client as httplib
import urllib
import sys
import Adafruit_DHT
GPIO.setwarnings(False)
channel = 2
GPIO_TRIGGER =23
GPIO_ECHO =24
buzzer=16
GPIO.setmode(GPIO.BCM)
GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
GPIO.setup(GPIO_ECHO, GPIO.IN)
GPIO.setup(channel,GPIO.IN)
GPIO.setup(buzzer,GPIO.OUT)
key = "MNB4I9YCUZB1BM5N"
def alert():
#Disable warnings (optional)
    #Select GPIO mode
    #Set buzzer - pin 23 as output
    #buzzer=23
    #GPIO.setup(buzzer,GPIO.OUT)
#Run forever loop
    for v in range(0,5):
        GPIO.output(buzzer,GPIO.HIGH)
        print ("Beep")
        time.sleep(5) # Delay in seconds
        GPIO.output(buzzer,GPIO.LOW)
        print ("No Beep")
        time.sleep(0.5)
def relay(a):
       if(a==1):
              GPIO.output(16,GPIO.LOW)
       if(a==0):
               GPIO.output(16,GPIO.LOW)
       #while True:
              #data=urllib.urlopen("
https://api.thingspeak.com/channels/1336169/fields/5.json?results=2")
              #select=repr(data.read());
              #select-select[300:];
              #pick=re.search('field5":"(.+?)",',select);
              #relay=pick.group(1);
              #if(b==200):
```

```
                    #if (relay==1):
                            #GPIO.output(16,GPIO.HIGH)
                            #break
                    #if(relay==0):
                            #GPIO.output(16,GPIO.LOW)
                            #break
def distance():
    # set Trigger to HIGH
    GPIO.output(GPIO_TRIGGER, True)

    # set Trigger after 0.01ms to LOW
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGGER, False)

    StartTime = time.time()
    StopTime = time.time()

    # save StartTime
    while GPIO.input(GPIO_ECHO) == 0:
        StartTime = time.time()

    # save time of arrival
    while GPIO.input(GPIO_ECHO) == 1:
        StopTime = time.time()

    # time difference between start and arrival
    TimeElapsed = StopTime - StartTime
    # multiply with the sonic speed (34300 cm/s)
    # and divide by 2, because there and back
    distance = (TimeElapsed * 34300) / 2

    return distance

def callback(ch):
        if GPIO.input(ch):
                print ("no Water Detected!")
        else:
                print (" Water Detected!")

GPIO.add_event_detect(channel, GPIO.BOTH, bouncetime=300)  # let us
know when the pin goes HIGH or LOW
GPIO.add_event_callback(channel, callback)  # assign function to GPIO
PIN, Run function on change

def cloud():
    while True:
        humi,temp = Adafruit_DHT.read_retry(11, 4)
        dist = distance()
        callback(channel)
        params=[urllib.parse.urlencode({'field2': temp,
'key':key}),urllib.parse.urlencode({'field1': humi,
'key':key}),urllib.parse.urlencode({'field3': dist, 'key':key})]
        for i in range(0,3):
        #Calculate CPU temperature of Raspberry Pi in Degrees C
```

```
            headers = {"Content-typZZe":
"application/x-www-form-urlencoded","Accept": "text/plain"}
            conn = httplib.HTTPConnection("api.thingspeak.com:80")
            try:
                conn.request("POST", "/update", params[i], headers)
                response = conn.getresponse()
                print("Temp: {0:0.1f} C  Humidity: {1:0.1f}
%".format(temp,humi))
                print ("Measured Distance = %.1f cm" % dist)
                #print("level of water in soil is %.1f" % soil)
                print(response.status, response.reason)
                data = response.read()
                conn.close()
            except:
                print("connection failed")
        if (dist<=4.7) or (4.7>=dist):
            alert()
        if(temp>33):
            relay(1)
        if(temp<33):
            relay(0)
        break
if __name__ == "__main__":
    try:
        while True:
            cloud()
    except KeyboardInterrupt:
        print("program stopped by User")
        GPIO.cleanup()
```

Result

1) With our project we became successful to demonstrate with regarding the objectives of the project.
2) Turn ON or OFF and check the status of the appliance through remote control.
3) Farmer successfully got the status of temperature, humidity, soil moisture content as well as presence of pest in his field by just accessing ThingSpeak from his master mobile.
4) We successfully stored the past and current values of temperature, humidity, soil moisture content as well as presence of pest data along with the devices operated by the user and with timestamp on cloud.

Step 1: Install all the required libraries through Raspberry Pi terminal and prepare the logic.

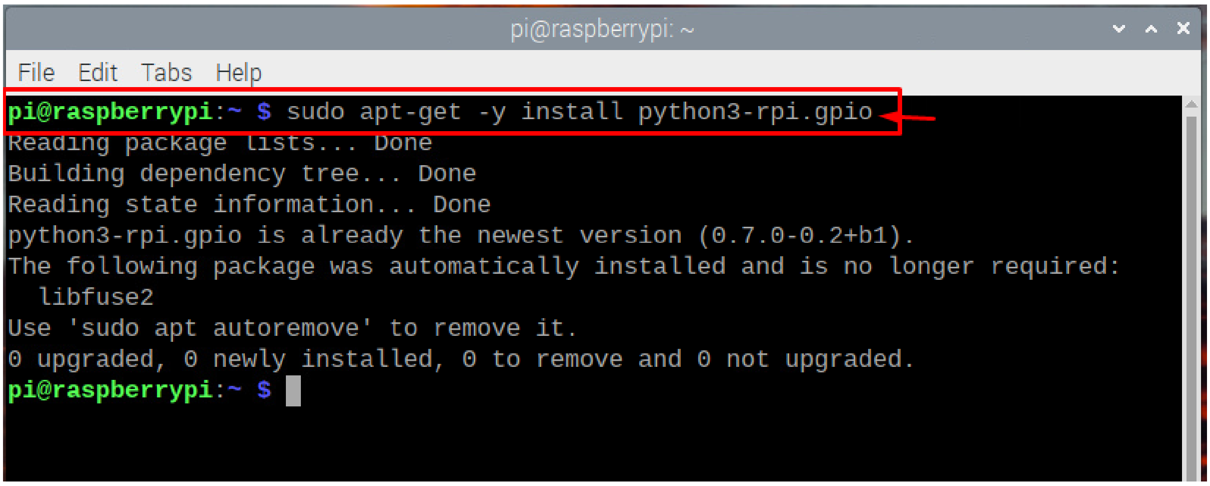

Fig 3.22

Step 2: Write the code for all the sensors and connecting circuit to cloud

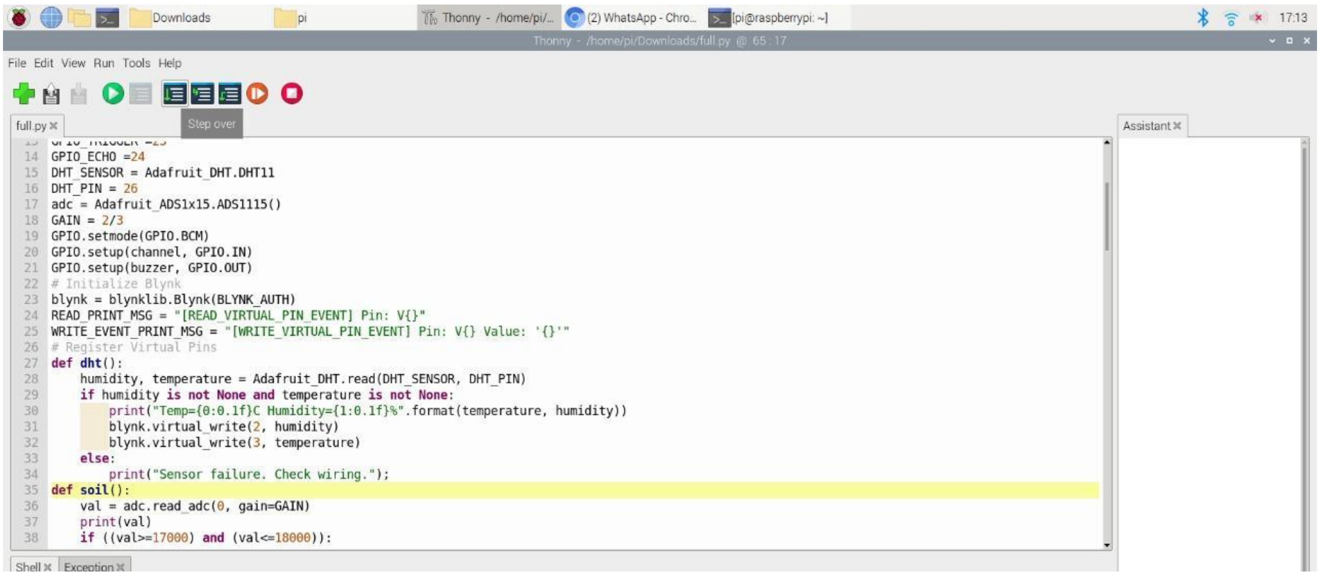

Fig 3.23

Step 3: Save the code

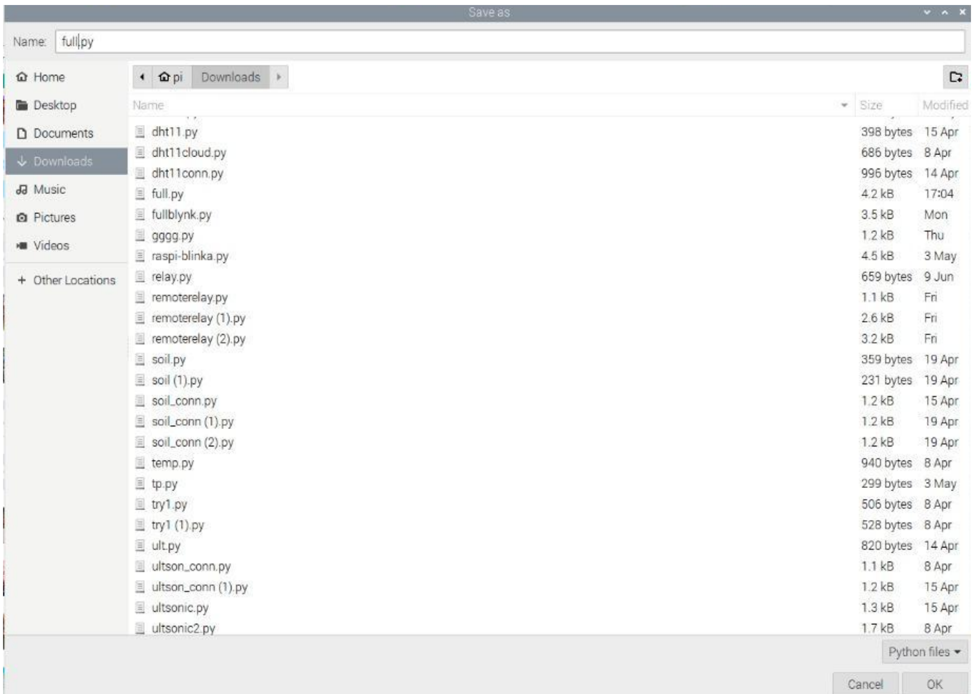

Fig 3.24

Step 4: Verify the code and check for errors.

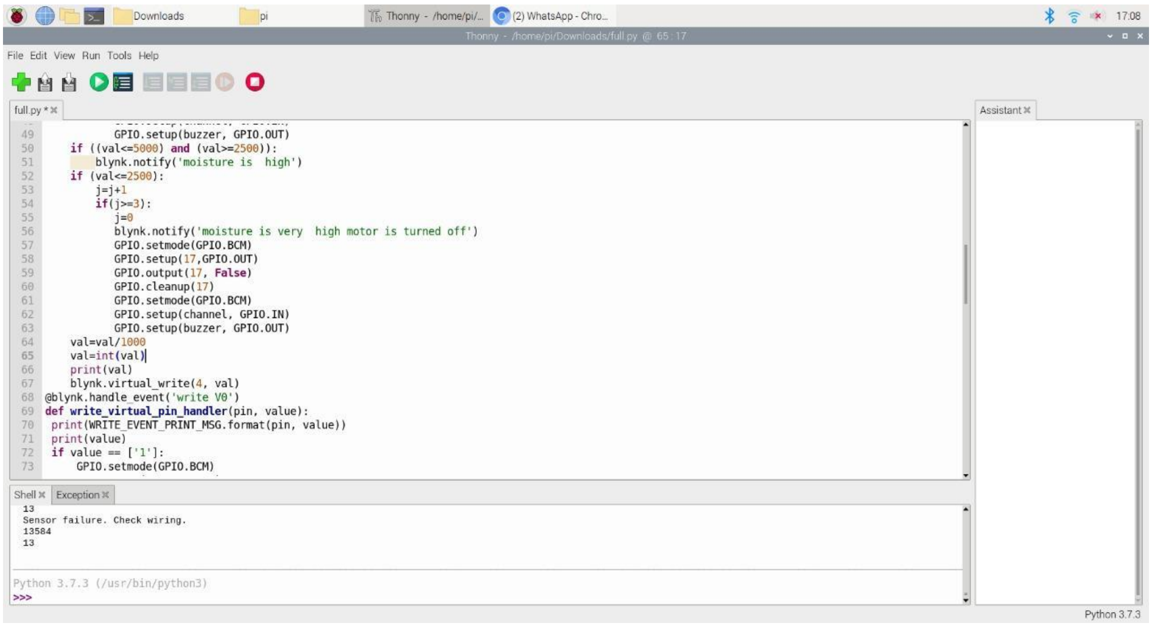

Fig 3.25

Step 5: Switch ON the power supply then power will be distributed to all the components.

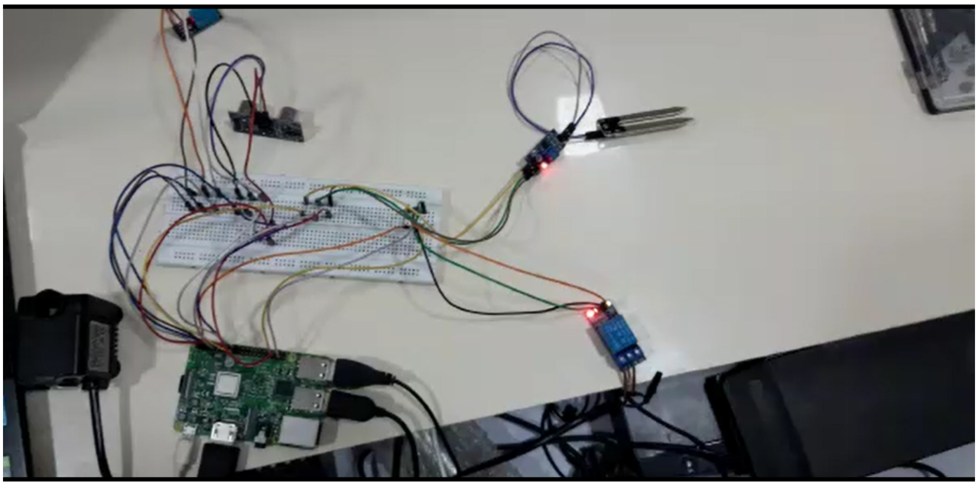

Fig 3.26

Step 6: Connect cloud with user and Raspberry Pi

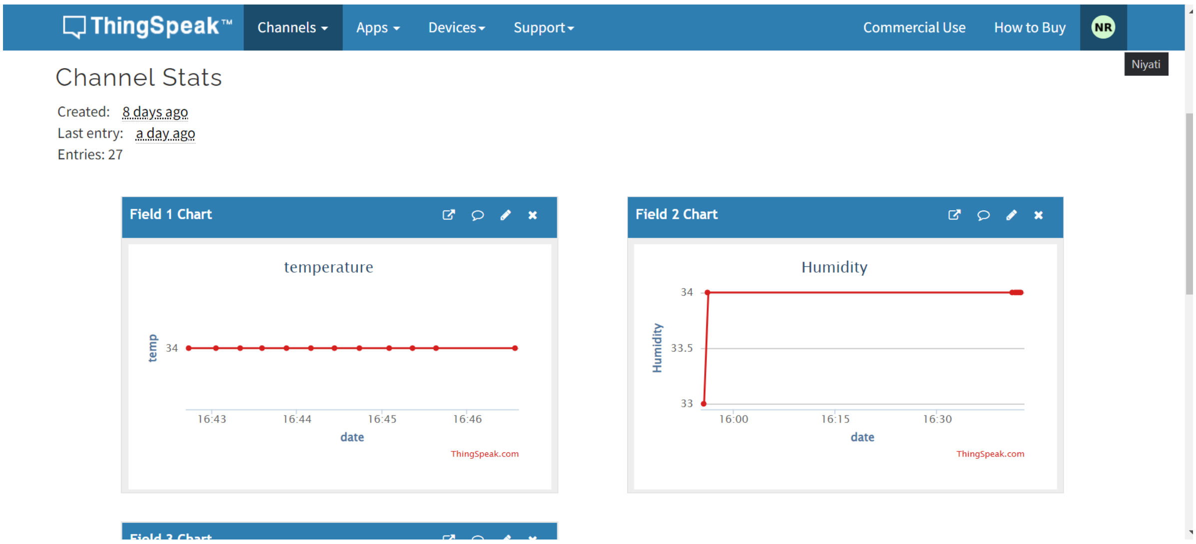

Fig 3.27

Step 7: Run the code

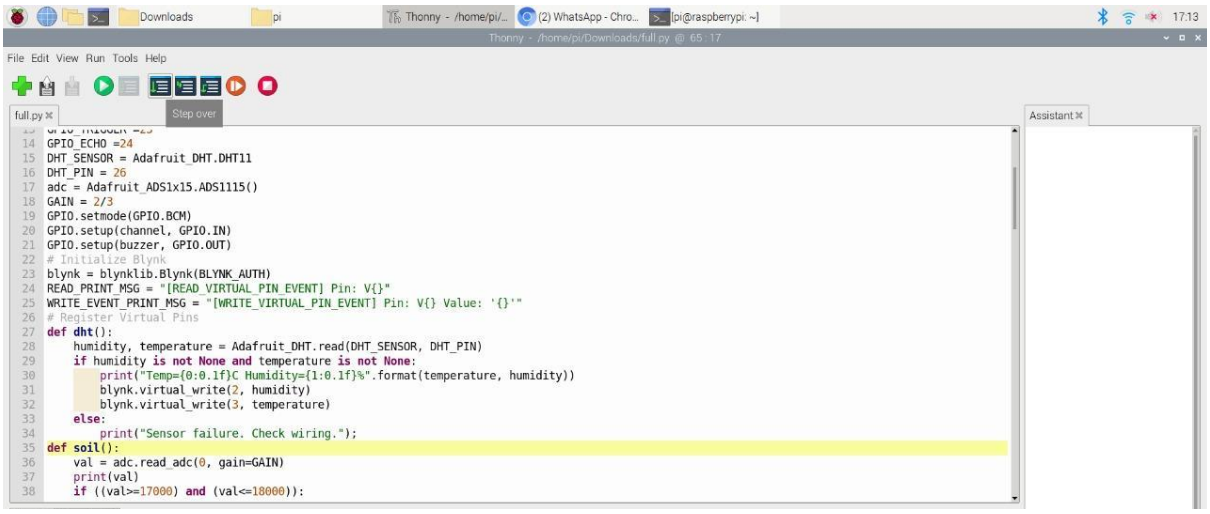

Fig 3.28

Step 8: All the data collected by the sensor will be uploaded in form of graph

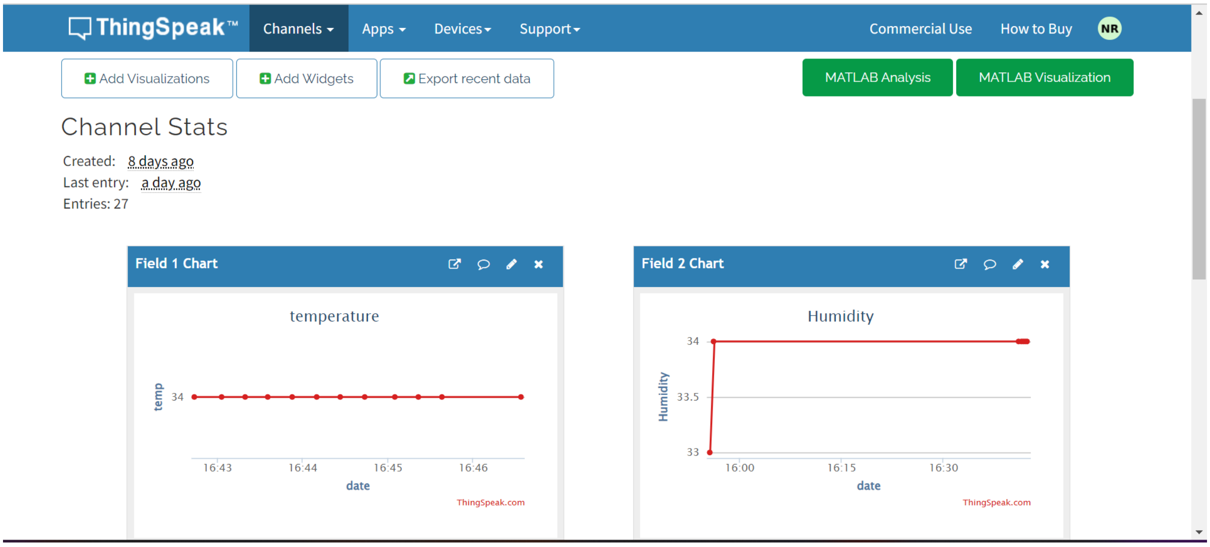

Fig 3.29

Step 9: We can also export data into .csv file

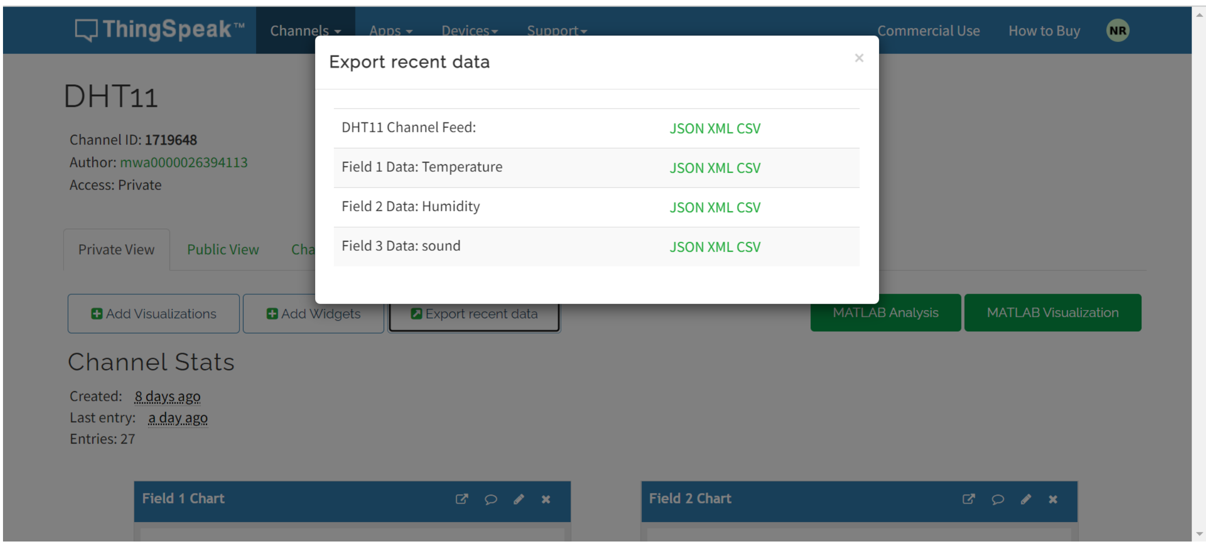

Fig 3.30

Chapter 4

Future Scope

The scope of our project "Smart Agriculture" is immense. The future implications of the project are very great considering the amount of time, money and resources it saves.

The project we have undertaken can be used as a reference or as a base for realizing a scheme to be implemented in other projects of greater level such as:

- Web services for alerting and analytics can be added, in addition to a monitoring user interface
- We can also use web camera to capture live crop images over Wi-Fi
- Crop disease detection using IOT and ML
- Smart farming with remote controlled robot
- Considering weather parameters, best suitable crop to be grown predicted using IOT and ML

Application

1) Nowadays due to the increase in the different facilities population is increasing day by day and since food is one of the basic needs of human beings, to make it available for the billion trillion of people traditional type of agricultural is not sufficient. The system we have designed is based on Agricultural technology.[7]
2) In the agricultural country like India this system can enhance their production in the scientific way.[3]
3) Since farmer can know the status of the fields from anywhere through his/her mobile and it consumes less time and capital.[5]
4) This project also prevents crops from pests which is a great help in farming.[1]

Conclusion

As per this system, the Raspberry Pi continuously monitors the status of the appliances and reports the status or any operations to be performed as requested by the user using ThingSpeak. The system also allows the farmer to remotely control the appliances by using the cloud service. The system can notify the on-field condition to the farmer anywhere in the world as long as there is active internet service. This project also gives information about environmental conditions such as temperature, humidity, and soil moisture content with the help of DHT11 and soil moisture sensor based on which crops are irrigated preventing wastage of water and excessive or less irrigation problems, it also detects pest through Acoustic Sound sensor and activates buzzer thus preventing damage of crop. These readings are also stored on cloud with timestamp. This project results in achieving adequate water management, saves men power and time more efficiently.

References

YouTube references: -

1. https://www.youtube.com/watch?v=ul2L3A57-aU
2. https://www.youtube.com/watch?v=aljmwxrV738
3. https://www.youtube.com/watch?v=kPLafcrng-c
4. https://www.youtube.com/watch?v=9LxrX5Eeukg
5. https://www.youtube.com/watch?v=jYeUS7JxmA8
6. https://www.youtube.com/watch?v=M3RuHX6jEXI
7. https://www.youtube.com/watch?v=hblqZLyWroE

Website references: -

1. https://www.instructables.com/Control-LED-Using-Raspberry-Pi-GPIO/
2. https://iotdesignpro.com/projects/how-to-send-data-to-thingspeak-cloud-using-raspberry-pi
3. https://iotdesignpro.com/projects/home-automation-with-node-red-and-raspberry-pi
4. https://electronicshobbyists.com/raspberry-pi-sending-data-to-thingspeak-simplest-raspberry-pi-iot-project/
5. https://www.electroniclinic.com/soil-moisture-sensor-with-raspberry-pi-circuit-diagram-and-python-code/
6. https://robu.in/raspberry-pi-ultrasonic-sensor-interface-tutorial/
7. https://www.teachmemicro.com/raspberry-pi-ultrasonic-sensor-tutorial/
8. https://www.academia.edu/download/55859640/IRJET-V5I1307.pdf IoT Based Agriculture Monitoring and Smart Irrigation System Using Raspberry Pi Mrs. T. Vineela, J. NagaHarini, Ch. Kiranmai, G. Harshitha, B. AdiLakshmi Assistant Professor, Dept. of ECE, VVIT, ANDHRA PRADESH, INDIA Student, Dept. of ECE, VVIT, ANDHRA PRADESH, INDIA
9. https://www.researchgate.net/profile/C_Balamurugan2/publication/319316898_Development_of_Raspberry_pi_and_IoT_Based_Monitoring_and_Controlling_Devices_for_Agriculture/links/59a435fda6fdcc773a373532/Development-of-Raspberry-pi-and-IoT-Based-Monitoring-and-Controlling-Devices-for-Agriculture.pdf Development of Raspberry pi and IoT Based Monitoring and Controlling Devices for Agriculture C.R.Balamurugan R.Satheesh Karpagam College of Engineering, Coimbatore, India, Arunai Engineering College, Tiruvannamalai
10. https://pdfs.semanticscholar.org/45ba/03cb224aad120e82a7097bd6f5c1559a4d29.pdf Real-time and low-cost IoT based farming using raspberry Pi Wahidur Rahman, Elias Hossain, Rahabul Islam, Harun-Ar-Rashid, Nur-A-Alam, Mahmodul Hasan Department of Computer Science and Engineering, Mawlana Bhashani Science and

Technology University, Bangladesh 2Department of Software Engineering, Daffodil International University, Bangladesh

11. https://www.researchgate.net/profile/Mohammad_Alshammari/publication/336266496_Design_and_Learning_Effectiveness_Evaluation_of_Gamification_in_e-Learning_Systems/links/5daee953a6fdccc99d92b461/Design-and-Learning-Effectiveness-Evaluation-of-Gamification-in-e-Learning-Systems.pdf#page=504 IoT based Temperature and Humidity Controlling using Arduino and Raspberry Pi Lalbihari Barik Department of Information Systems, Faculty of Computing and Information Technology in Rabigh King Abdulaziz University, Kingdom of Saudi Arabia
12. https://ieeexplore.ieee.org/abstract/document/8286464/