SCHOOL OF INFORMATION TECHNOLOGY ENGINEERING

Project Report

Submitted in partial fulfillment for the award of the degree of Bachelor of Technology (Information Technology)

Fall Semester – 2023-24

ITE 4003 – Internet of Things

Slot – **D1+TD1**

Under the guidance of **- Prof. Pounambal M**

IOT based Irrigation Support System

Group Members:

Abhishek KB - 20BIT0403
Harsh Gautam - 20BIT0373
Rohil Saxena - 20BIT0404
Mudit Dhoundiyal - 20BIT0411
Neeraj Kumar Choudhary - 20BIT0397

Index

Sr. no	Content	Page no.
1.	Motivation	3
2.	Scope of Project	4
3.	System Architecture	5
4.	Experimental Setup	6
4.1	MQ3 Sensor	6
4.2	DHT11 Sensor	7
4.3	Arduino Uno	8
4.4	ThingsSpeak	9
5.	Pseudo Code	10
6.	Project Kit	13
7.	Results	14
8.	Outcomes	15
9.	Acknowledgment	15

❖ Motivation

The motivation behind this project reflects a growing awareness of the pressing need to modernize and enhance traditional agricultural practices to meet the demands of a rapidly changing world. Here's a more in-depth exploration of the key points within the provided statement:

1. Challenges in Traditional Agriculture: Traditional agricultural practises have sustained humanity for centuries, but they face new challenges in the twenty-first century. Climate change, resource scarcity, population growth, and the need to produce more food with fewer resources are among the challenges. The project recognises that we cannot effectively address modern challenges using traditional methods.

2. Integration of IoT: The adoption of modern technology, particularly the Internet of Things (IoT), represents a paradigm shift in agriculture. The Internet of Things (IoT) refers to the connection of everyday objects to the internet, allowing them to collect and exchange data. This means that sensors, devices, and machinery in agriculture can be linked to the internet, allowing for real-time data collection, analysis, and control.

3. Data-Driven Decisions: The project emphasises the significance of data-driven agricultural decisions. Farmers and agricultural professionals can gather detailed information about various parameters such as soil moisture, weather conditions, and crop health by deploying sensors and IoT devices in the field. This information can then be analysed to help farmers make informed decisions about irrigation, fertilisation, pest control, and other important aspects of farming.

4. Automation: IoT not only collects data but also allows for automation. Irrigation systems, for example, can be automated to respond to real-time data using IoT. Water is thus delivered precisely when and where it is required, reducing waste and optimising resource utilisation. Other aspects of agriculture, such as monitoring and controlling environmental conditions within greenhouses, can also benefit from automation.
5. Impact of Environmental Conditions: The statement emphasises the importance of environmental monitoring and control. Ammonia levels, humidity, and temperature can all have a significant impact on crop growth and quality. Plants can be harmed by high ammonia levels, and

incorrect humidity and temperature levels can result in diseases or poor yields. IoT sensors can continuously monitor these conditions and trigger responses to keep the parameters within ideal ranges.

6. Resource Utilization: The project intends to optimise resource utilisation by utilising IoT technology. This entails using less water, fertiliser, and energy. Optimising these resources not only increases productivity, but it also contributes to agricultural sustainability, which is critical for long-term food security and environmental preservation.

In summary, the motivation for this project recognises the critical importance of modernising agriculture to meet the challenges of the twenty-first century. IoT technology, data-driven decisions, and automation have the potential to transform farming practises, resulting in increased productivity, resource efficiency, and sustainability, all of which are critical for agriculture's future.

❖ Scope of the project

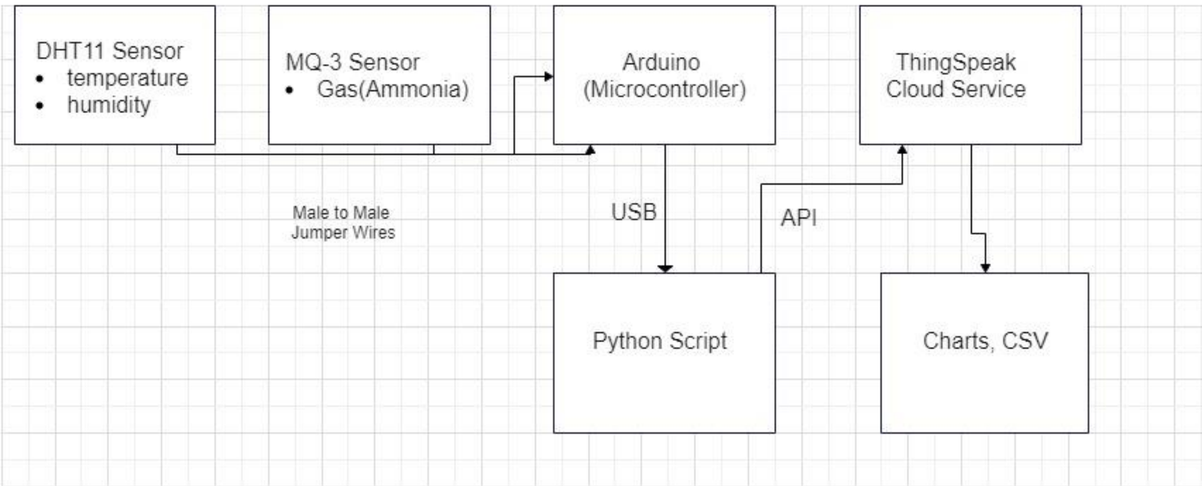
The scope of this project is to design and implement a cost-effective IoT-based agriculture system that can remotely monitor and control environmental parameters. The project focuses on three main aspects:

1. Data Collection: The project involves the use of sensors, including an MQ3 sensor for ammonia percentage and a DHT11 sensor for humidity and temperature. These sensors are placed in an open-air environment to collect real-time data.

2. Data Transmission: The collected data is sent to an Arduino microcontroller. Since the Arduino is low-cost and not connected to Wi-Fi, a laptop acts as an intermediary to gather data from the Arduino. This data is then transmitted to the cloud for visualization and analysis.

3. Visualization and Control: Once in the cloud, the data can be visualized and analyzed, providing insights for smart irrigation decisions. Additionally, control mechanisms can be implemented to mitigate the need of resources based on the collected data.

❖ System Architecture



The architecture consists of DHT11 and MQ-3 sensors, measuring temperature, humidity, and Ammonia gas concentration, connected to an Arduino. The Arduino acts as a data aggregator, collecting information from the sensors. Notably, the Arduino lacks built-in Wi-Fi capabilities.

A Python script runs on a separate device with Wi-Fi functionality and connects to the Arduino via USB. This script is responsible for retrieving data from the Arduino, formatting it, and transmitting it to the ThingSpeak cloud service using the ThingSpeak API.

ThingSpeak serves as the central hub for data storage, providing a platform for visualization and analysis. Users can access real-time and historical data for temperature, humidity, and gas concentration through charts and graphs on ThingSpeak. Additionally, the platform allows users to download the collected data in CSV format for further offline analysis.

In summary, this architecture establishes a seamless flow of sensor data from the physical environment to the cloud, enabling real-time monitoring and analysis of environmental parameters through ThingSpeak.

❖ Experimental Setup

The experimental setup for this project consists of the following key components:
1. MQ3 Sensor: This sensor is used to measure the percentage of ammonia in the environment.
2. DHT11 Sensor: The DHT11 sensor is employed to collect data on humidity and temperature.
3. Arduino : The Arduino serves as the central processing unit that collects data from the sensors and acts as an interface between the sensors and the laptop.
4. Laptop: The laptop is used to receive data from the Arduino and transfer it to the cloud for visualization and analysis.
5. Cloud Platform: The cloud platform is where the collected data is stored, visualized, and analyzed. It allows for remote access to the data and control over the irrigation system.

Following components are explained in detail-

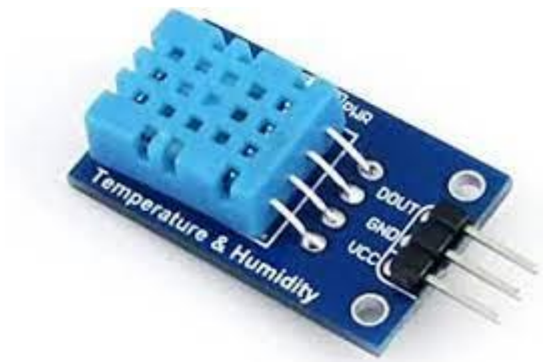
A.) MQ3 sensor

The MQ-3 sensor detects alcohol vapour or ethanol in the air and is widely used in Internet of Things (IoT) applications. It's widely used for a variety of applications, including breathalysers, industrial safety, and home automation systems. Here are some important details about the MQ-3 sensor in the context of IoT:

1. Working Principle: When exposed to alcohol vapour, the MQ-3 sensor relies on the conductivity change of its internal sensing element. It has a tin dioxide (SnO_2) sensing layer with varying resistance depending on the amount of alcohol in the air. When there are alcohol vapours present, the sensor's resistance decreases, which can be measured and used for detection.

2. Calibration: Calibration is required for MQ-3 sensors to provide accurate measurements. The sensor's response is not linear, and it must be set against known alcohol concentrations to ensure accurate readings.
3. Connectivity: MQ-3 sensors are frequently paired with microcontrollers (e.g., Arduino, Raspberry Pi) in IoT applications to collect and transmit data. These sensors can be integrated into IoT devices and systems and send data to a central server or cloud platform for analysis using various communication protocols such as Wi-Fi, Bluetooth, or LoRa.
4. Data Analysis: The MQ-3 sensor's data can be analysed to trigger alerts or actions based on predefined thresholds. For example, if the concentration of alcohol in a specific area exceeds a certain level, an alert can be sent to the appropriate parties or authorities.
5. Limitations: While MQ-3 sensors are effective for detecting alcohol vapour, they may not be suitable for precise alcohol concentration measurements, particularly in legal or medical settings. Calibration is required for precise results.

B.) DHT11 sensor



The DHT11 sensor is a popular choice in IoT (Internet of Things) applications for monitoring temperature and humidity. It is a low-cost digital sensor that provides accurate and consistent readings, making it suitable for a variety of IoT projects. Here are some important details about the DHT11 sensor in the context of IoT:

1. Sensor description: The DHT11 sensor is a combination digital temperature and humidity sensor. It includes a calibrated digital signal output from temperature and humidity measurements. It is made up of a capacitive humidity sensor and a thermistor for measuring temperature. The sensor comes in a small package and can be easily integrated into IoT devices.

2. Working principle: The DHT11 sensor measures changes in electrical capacitance caused by humidity as well as resistance changes caused by temperature. It converts these changes into digital values that a microcontroller or IoT device can read.
3. Connectivity: The DHT11 sensor typically communicates via a single-wire digital protocol. It is simple to connect to popular microcontrollers such as Arduino, Raspberry Pi, and ESP8266. The data from these microcontrollers can then be transmitted to a central server or cloud platform for analysis.
4. Data Analysis: The DHT11 sensor's data can be analysed to provide valuable insights or to trigger actions based on predefined thresholds. In a smart greenhouse, for example, sensor data can be used to automate the irrigation system.
5. Limitations: While the DHT11 is cost-effective and reliable for many applications, it does have limitations, such as lower accuracy when compared to more advanced sensors. It might not be appropriate for precise measurements or critical scientific research.

C.) Arduino Uno

In the field of IoT (Internet of Things), Arduino is a widely used and versatile platform. IoT refers to the process of connecting physical objects, devices, and sensors to the internet in order to collect, exchange, and analyse data for a variety of applications. Arduino boards and their associated development environment provide a number of benefits for IoT projects, including:

1. Simpleness and accessibility: Arduino is well-known for its user-friendly platform. It offers an open-source hardware and software ecosystem, making it accessible to users of varying technical expertise. This ease of use is especially beneficial for those new to IoT development.

2. Versatility: Arduino boards are available in a variety of models and configurations, allowing you to select the one that best meets the needs of your IoT project. There's an Arduino board for everyone, whether you need a small, low-power board for battery-powered devices or a more powerful one for data-intensive applications.
3. Rich Community Support: Arduino has a large and active developer, maker, and enthusiast community. This community support gives you access to a plethora of online tutorials, forums, and projects, which makes it easier to find solutions and guidance for your IoT projects.
4. Compatibility: Arduino is compatible with a wide range of sensors, modules, and shields that can be easily integrated into your IoT projects. This compatibility makes connecting and utilising various components easier.
5. Programming: Arduino is programmed using a simplified version of the C/C++ programming language. This broadens its appeal and allows you to quickly programme your IoT devices to collect and transmit data.

D.) ThingsSpeak

ThingSpeak is an Internet of Things (IoT) platform and data analytics service that allows you to collect, analyse, and visualise data from IoT devices. It offers a simple and user-friendly method for connecting sensors and other devices to the internet, storing data, and performing real-time data analysis. Here are some important points to remember about ThingSpeak in the context of IoT:

1. Data Collection: ThingSpeak is intended to collect data from Internet of Things devices. It can receive information from a variety of sensors, microcontrollers, and other connected devices. Temperature, humidity, pressure, GPS coordinates, and other data types are supported by the platform.

2. Data Storage: ThingSpeak offers cloud-based data storage, which allows you to securely store and manage your IoT data in the cloud. This is especially helpful for data logging and archiving, as it ensures that historical data is easily accessible.
3. Real-Time Data: The platform supports the collection and visualisation of real-time data. You can use dashboards and plots to monitor data from your IoT devices. This is useful for quick insights and responses to changing conditions.
4. Data Analysis: ThingSpeak includes data analysis tools that enable you to perform basic data processing and visualisation. Custom MATLAB scripts can be written for more advanced data analysis and automation.
5. Integration with IoT Devices: ThingSpeak can be integrated with a wide range of IoT devices and development platforms, including Arduino, Raspberry Pi, ESP8266, and more. It supports multiple communication protocols, including HTTP, MQTT, and RESTful API, for device-to-cloud connectivity.

❖ Pseudo Code:

The python script used in the project can be explained with the following pseudo code

1. **Importing Libraries:**
   - The code begins by importing several Python libraries, including **serial**, **re** (for regular expressions), **urllib.request**, **json**, and **requests**. These libraries are used for serial communication, data parsing, web requests, and working with JSON data.
2. **Serial Connection Setup:**
   - The **serial.Serial** function is used to establish a serial connection with an Arduino device connected to 'COM3' at a baud rate of 9600. This is the typical setup for communicating with an Arduino over a serial connection.
3. **features Function:**
   - The **features** function is defined. This function appears to be an endless loop that continuously reads data from the Arduino and performs actions based on that data.
4. **Data Retrieval from Arduino:**
   - Inside the loop, it checks if there is data available to read from the Arduino using **ArduinoSerial.inWaiting()**. If data is available, it reads a line of data from the Arduino using

ArduinoSerial.readline(). This data is assumed to be in a specific format.

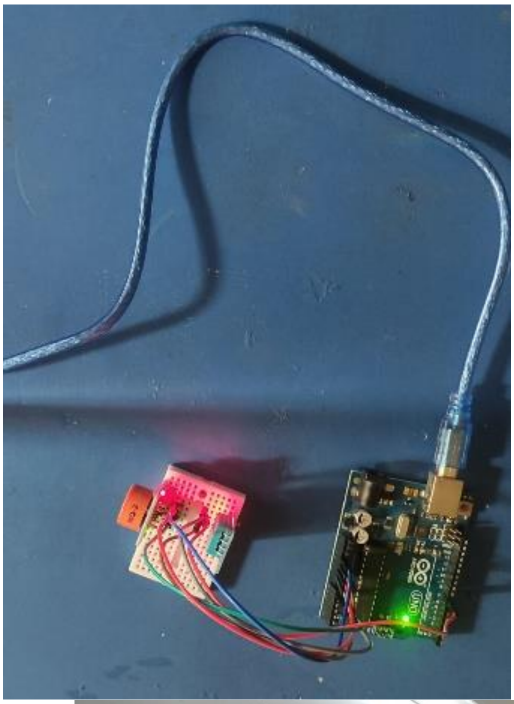
5. **Data Parsing:**
   - The code then parses the data to extract numeric values and information related to ammonia levels, humidity, and temperature. This is done using regular expressions and string manipulation. Extracted values are stored in the **res** list.
6. **Data Transmission to ThingSpeak:**
   - The code sends the extracted data to ThingSpeak, an IoT platform for data storage and analysis. It constructs a URL with the data and sends an HTTP request to update the data fields in a ThingSpeak channel.
7. **Retrieving Data from ThingSpeak:**
   - It retrieves data from ThingSpeak using an HTTP GET request. The data includes fields related to humidity, temperature, and ammonia levels.
8. **Data Analysis:**
   - The code extracts the values for humidity, temperature, and ammonia from the retrieved data.
9. **Standard Thresholds:**
   - The code defines standard threshold values for temperature, humidity, and ammonia levels that are considered favorable for irrigation.
10. **Condition Evaluation:**
    - It checks if the collected data falls within the standard threshold ranges. Based on the conditions, it prints messages to indicate whether the conditions are favorable for irrigation or if there are issues with temperature, humidity, or ammonia levels.
11. **Loop Continuation:**
    - The loop continues, collecting and analyzing data from the Arduino in a continuous manner.
12. **Function Invocation:**
    - Finally, the **features** function is invoked, starting the data collection and analysis process.

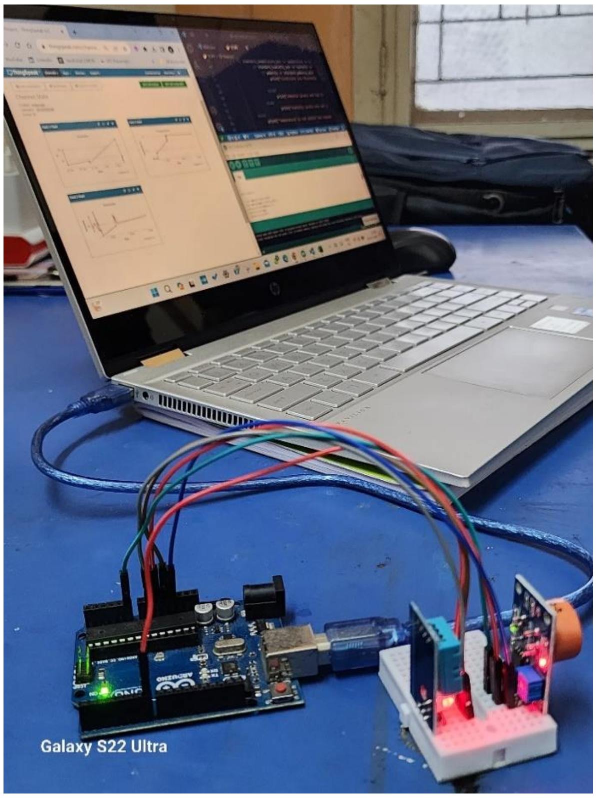
Overall, the code reads data from an Arduino device, sends it to ThingSpeak for storage, retrieves data from ThingSpeak for analysis, and checks if environmental conditions are suitable for irrigation based on predefined standards. The code is designed to run continuously, making it suitable for ongoing monitoring and control of environmental conditions in an agricultural or similar setting.

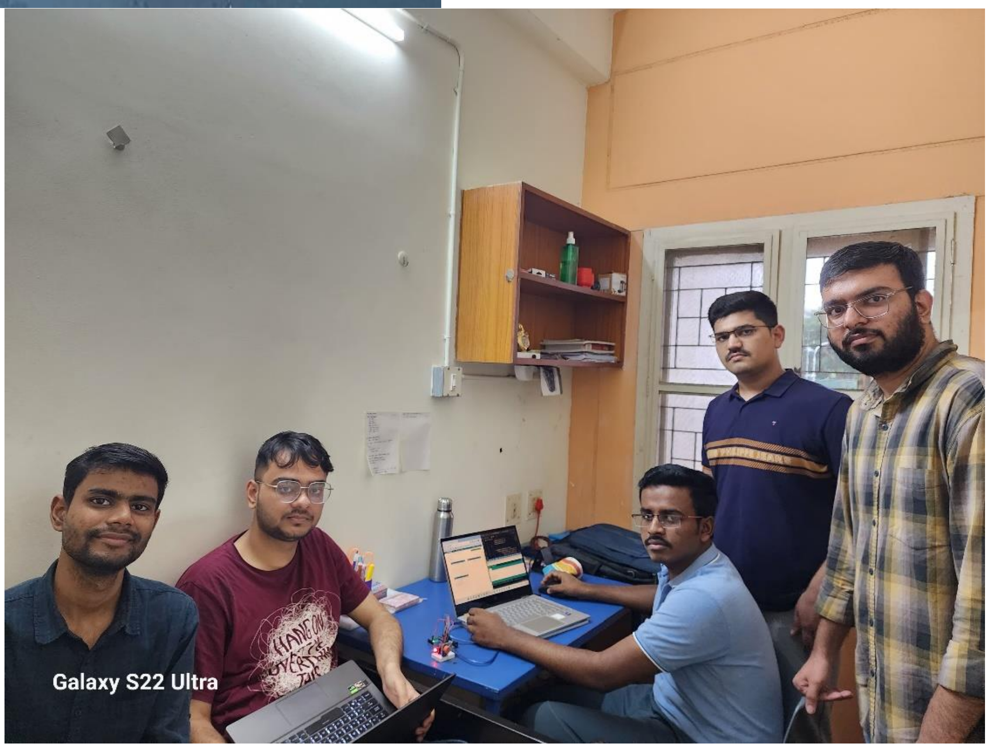
1. **Library Inclusion:**
    - The code starts by including the necessary libraries, "SimpleDHT" for the DHT11 sensor and "Servo" for servo motor control. However, no servo-related code is present in this snippet.
2. **Variable Declarations:**
    - It declares two integer variables: **pinDHT11** and **pinA0** for specifying the pins to which the DHT11 sensor and the ammonia sensor are connected, respectively. The **SimpleDHT11** object is created using the specified pin for the DHT11 sensor.
3. **Setup Function:**
    - In the **setup** function, it initializes the serial communication with a baud rate of 9600 using **Serial.begin(9600)**.
4. **Loop Function:**
    - The **loop** function is the main execution loop that continuously runs.
5. **Ammonia Sensor Reading:**
    - It reads the analog value from pin A0 using **analogRead(A0)** and maps the value to a percentage scale (0-100) to represent ammonia levels. The calculated ammonia value is printed to the serial monitor.
6. **DHT11 Sensor Reading:**
    - It reads data from the DHT11 temperature and humidity sensor using the **dht11.read** function. The temperature and humidity values are extracted, and if the reading is successful, these values are printed to the serial monitor. The DHT11 sensor has a sampling rate of 1Hz.
7. **Delay:**
    - The **delay(500)** line is commented out, but it's typically used to control the sampling rate of the DHT11 sensor. The DHT11 sensor has a default sampling rate of 1Hz, which is roughly equivalent to a 2-second delay.

In conclusion, the code reads data from both a DHT11 sensor and an ammonia sensor. The temperature and humidity data from the DHT11 sensor are read and displayed on the serial monitor. The ammonia sensor data, after analog-to-percentage mapping, is also displayed on the serial monitor.

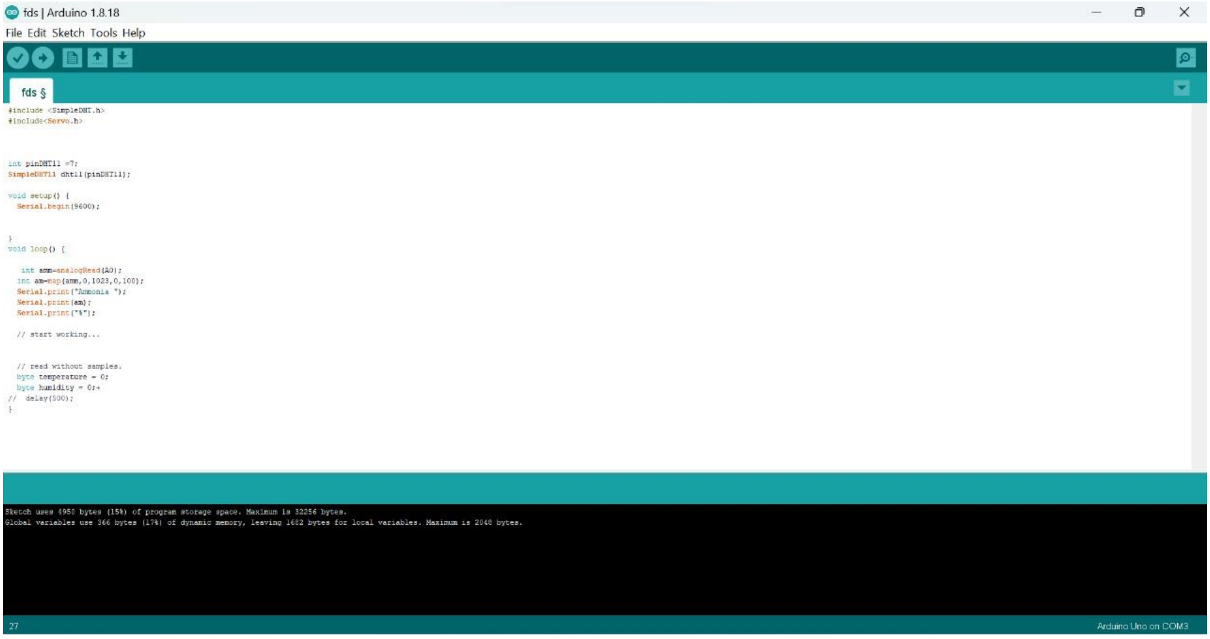
❖ Project Kit



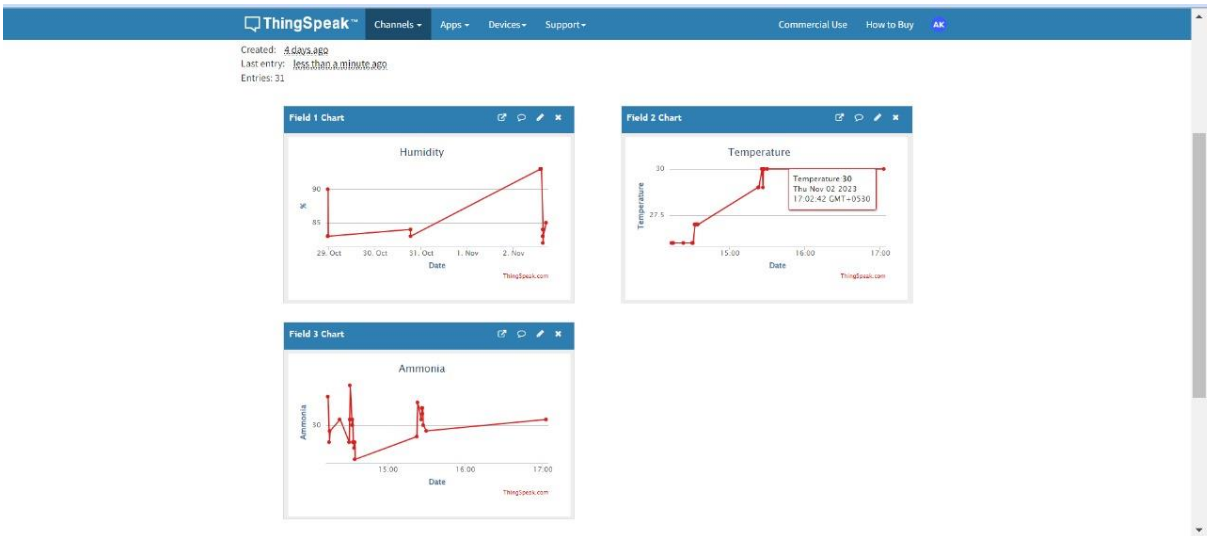

Galaxy S22 Ultra


Galaxy S22 Ultra

❖ Results


fds | Arduino 1.8.18
File Edit Sketch Tools Help
fds §
#include <SimpleDHT.h>
#include<Servo.h>
int pinDHT11 =7;
SimpleDHT11 dht11(pinDHT11);
void setup() {
Serial.begin(9600);
}
void loop() {
int amm=analogRead(A0);
int am=map(amm,0,1023,0,100);
Serial.print("Ammonia ");
Serial.print(am);
Serial.print("%");
// start working...
// read without samples.
byte temperature = 0;
byte humidity = 0;+
// delay(500);
}
Sketch uses 4950 bytes (15%) of program storage space. Maximum is 32256 bytes.
Global variables use 366 bytes (17%) of dynamic memory, leaving 1682 bytes for local variables. Maximum is 2048 bytes.
27
Arduino Uno on COM3


COM3
Send
Ammonia 33% Temperature: 30 C Humidity: 86
Ammonia 33%Read DHT11 failed, err=16,16
Ammonia 33% Temperature: 30 C Humidity: 86
Ammonia 33%Read DHT11 failed, err=16,12
Ammonia 33% Temperature: 30 C Humidity: 86
Ammonia 33%Read DHT11 failed, err=16,16
Ammonia 33% Temperature: 30 C Humidity: 86
Ammonia 33%Read DHT11 failed, err=16,16
Ammonia 33% Temperature: 30 C Humidity: 86
Ammonia 33%Read DHT11 failed, err=16,12
Ammonia 32% Temperature: 30 C Humidity: 86
Ammonia 32%Read DHT11 failed, err=16,16
Ammonia 32% Temperature: 30 C Humidity: 86
Ammonia 32%Read DHT11 failed, err=16,16
Ammonia 33% Temperature: 30 C Humidity: 86
Ammonia 33%Read DHT11 failed, err=16,12
Autoscroll
Show timestamp
Newline
9600 baud
Clear output


ThingSpeak™
Channels
Apps
Devices
Support
Commercial Use
How to Buy
Created: 4 days ago
Last entry: less than a minute ago
Entries: 31
Field 1 Chart
Humidity
Date
Field 2 Chart
Temperature
Temperature 30
Thu Nov 02 2023
17:02:42 GMT+0530
Date
Field 3 Chart
Ammonia
Date

❖ Outcome

The expected outcome of this project includes the following:
1. Real-time monitoring of ammonia levels, humidity, and temperature in an open-air agricultural environment.
2. Data visualization and analysis in the cloud, providing insights into environmental conditions.
3. The ability to make data-driven decisions leading to efficient resource usage and increased crop yield.

❖ Acknowledgement

We would like to express our special thanks of gratitude to Prof. Pounambal Ma'am of VIT Vellore who gave us the golden opportunity to do this wonderful course which also helped us in getting good knowledge of Iot devices and we have come to know about so many new things. Secondly, we would also like to thank our family and friends who helped us a lot in finishing this project within the limited period. It helped us increase our knowledge and skills.

ITE4003 Internet of Things
IOT Based Irrigation Support System

Motivation

- Addressing the need for efficient management of agriculture practices, the project aims to create a IoT-based Smart Agriculture system .
- It focuses on optimizing resource usage and crop health.
- By monitoring and controlling these factors, we can optimize resource utilization, leading to increased productivity and sustainability in agriculture.

Scope of the Project

- The project involves using MQ3 and DHT11 sensors connected to Arduino for real-time data collection.
- Data is transmitted to the cloud for visualization, enabling farmers to make data-driven irrigation decisions.
- Additionally, control mechanisms can be implemented to mitigate the need of resources based on the collected data.

System Architecture

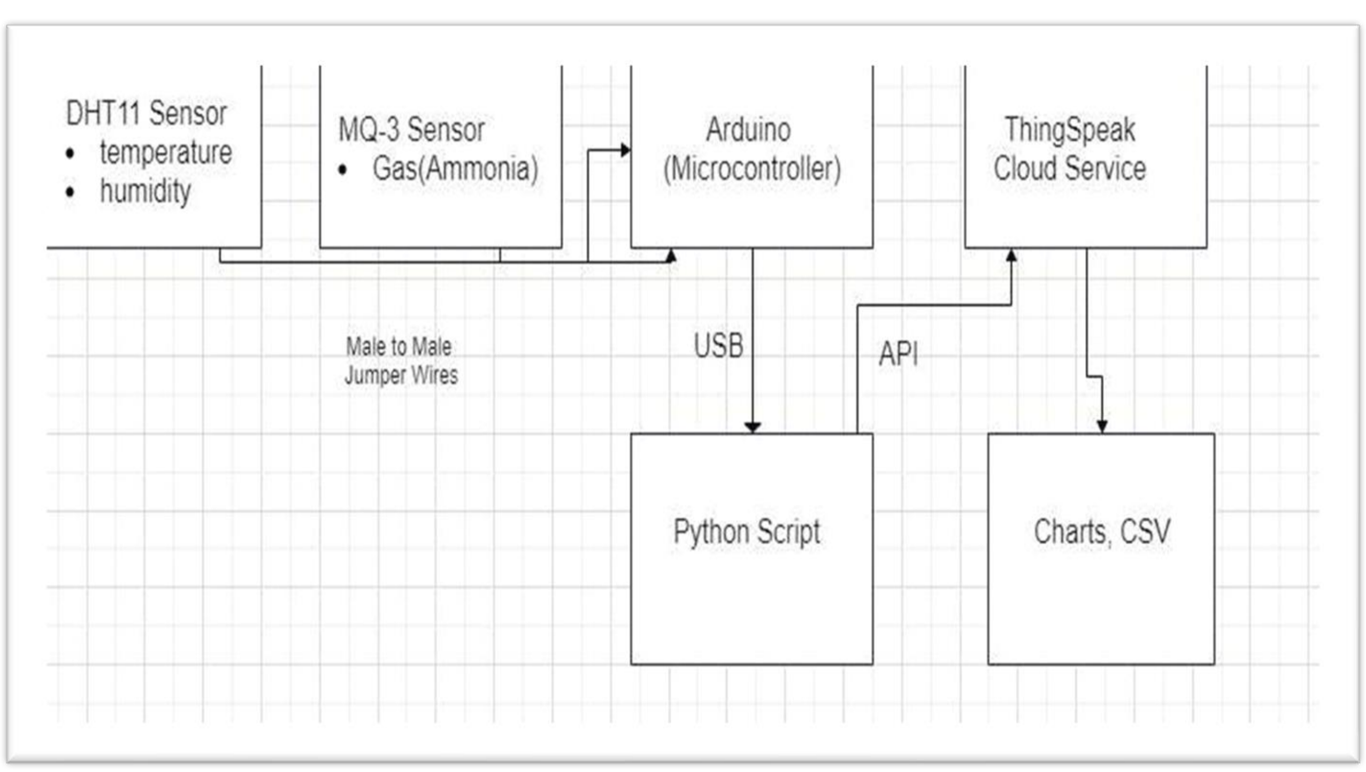
Description

•Sensors:
- DHT11 (Temperature, Humidity)
- MQ-3 (Gas - Ammonia)

•Arduino:
- Collects data from sensors.

•Python Script:
- Connects to Arduino via USB.
- Sends data to ThingSpeak.

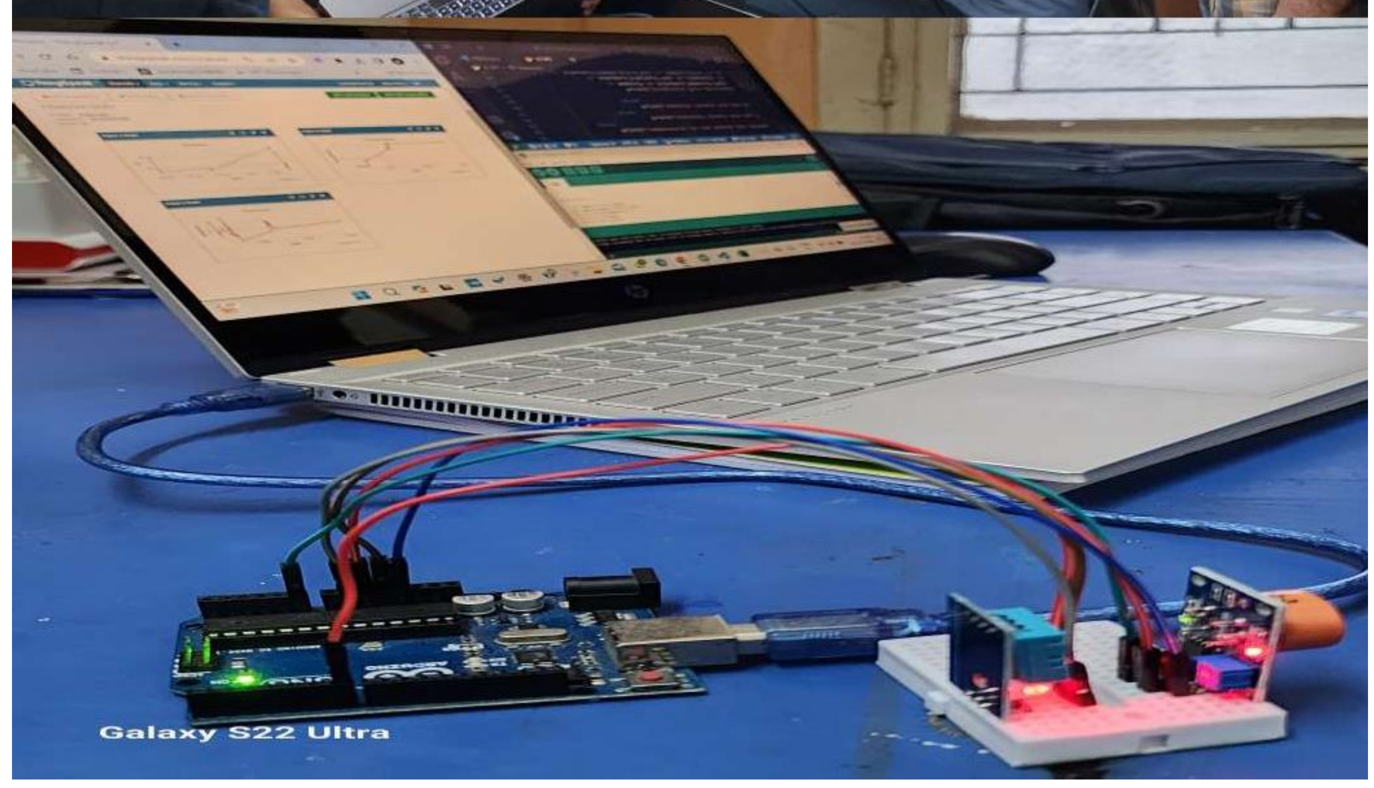
•ThingSpeak:
- Cloud service for data storage.
- API for data upload.
- Provides visualization tools.
- Can make to notify the user by giving mail and other call services.



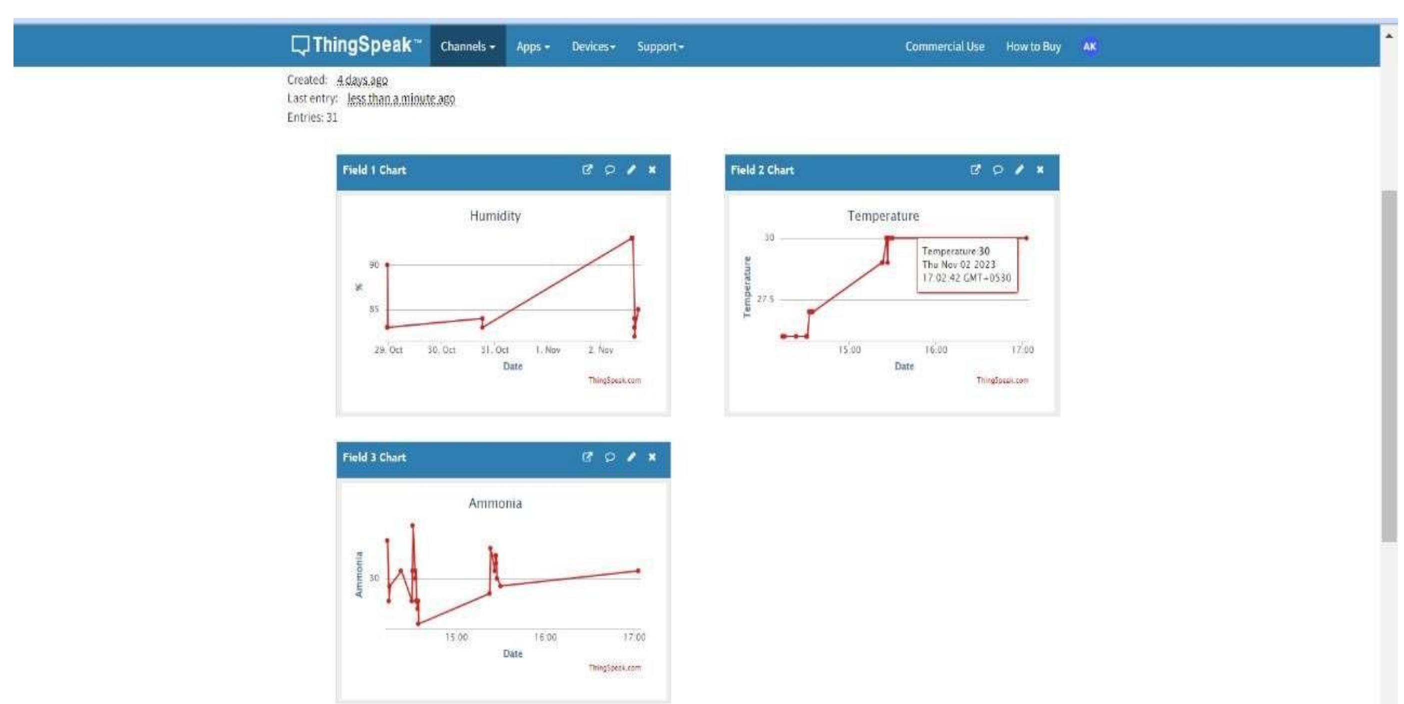
Experimental Setup

- Utilizes MQ3 sensor for ammonia data and DHT11 sensor for humidity and temperature.
- Arduino gathers data from sensors and is cost-effective but not Wi-Fi enabled.
- Python Script extracts data from physical Arduino memory and transfers it to the cloud on which data visualization graphs are generated for the uploaded data.

Project Kit



Results



Outcome

- Effective support for crop irrigation .
- Remote access to data through cloud visualization.
- Cost-effective and data-driven agriculture.

Guided By
Faculty Name: **Dr. POUNAMBAL M**

Team Members:
(Abhishek K B)(20BIT0403)……………………SCORE
(Rohil Saxena)(20BIT0404)……………………SCORE
(Harsh Gautam)(20BIT0373) ……………………SCORE
(Neeraj Kumar Choudhary)(20BIT0397)……………………SCORE
(Mudit Dhoundiyal)(20BIT0411)……………………SCORE

Acknowledgments
Thanks to VIT & SCORE