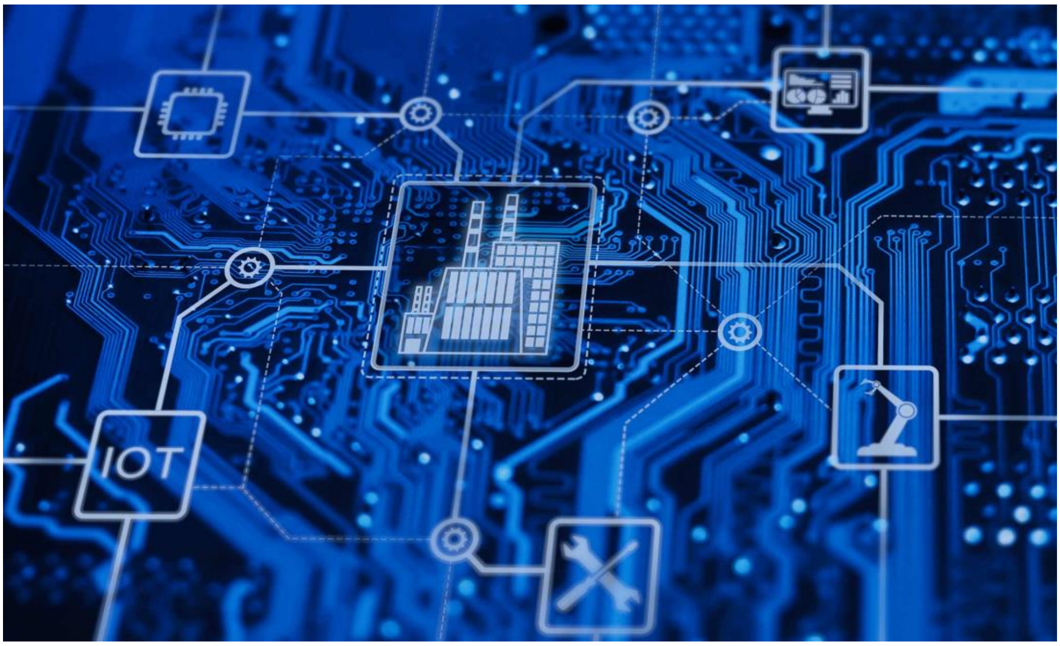
K.RAMAKRISHNAN
COLLEGE OF ENGINEERING
An Autonomous Institution
Permanently Affiliated to Anna University Chennai, Approved by AICTE New Delhi,
ISO 9001:2015, 14001:2015 certified institution, Accredited by NBA and with A grade by NAAC
Samayapuram, Tiruchirappalli – 621 112, Tamilnadu, India.

REPORT
ON
INTERNET OF THINGS
(27.08.24 – 28.08.24)



Submitted By;

Fiza Begum M

3rd year CSE A

KRCE

INDEX

DAY: 1

DATE: 27.08.2024

INTRODUCTION OF IOT:

The Internet of Things or simply IOT refers to a broad realm where physical devices, ranging from simple household equipment to complex manufacturing plants, are connected to the Internet. It's this interrelated quality toward these devices which facilitates the function of information capture and distribution along with its evaluation enhancing the levels of automation, efficiency, and analytical capabilities of the devices.

A good practical example of IoT application at work is in FastTag technology, which is extensively applied in the electronic toll collection system over the highways. In this system, vehicles are fitted with RFID (Radio Frequency Identification) tags that are read by scanners at toll booths. The effect is that the toll fee can easily be paid without the physical methods of stopping in payment umbilicals as the vehicle continues its fast motion. Hence, this reduces bottlenecks and enhances the flow of the traffic.

The operation of the IoT system consists of a multi-layered stack. The basis of the structure is formed by the first layer or the perception layer which contains devices and sensors capturing the environment, for example temperature motion or in the case of FastTag vehicle. That information, in turn, moves to the next tier through the connectivity layer via Wi-Fi, Bluetooth, Zigbee or mobile networking. Proceeding with one more layer, this one is edge computing which aims at bringing the processing of the data at a point as close to its source as possible allowing for lesser latency and bandwidth for the further processing of data.

Following this step is the data accumulation layer where data is progressed into storage and later analytics in the cloud where prospects for big data analytics exist. The upmost layer is the application layer where the users have access to such interface as mobile applications or dashboards which they can use to track the operation of the IoT system, manage it or analyse the performance of the system.

IoT platforms play a vital role in managing the complex network of connected devices. Leading platforms like AWS IoT, Microsoft Azure IoT, Google Cloud IoT, and IBM Watson IoT provide complete solutions to connect, manage, and analyse data from IoT devices. These platforms also offer secure communication, control over devices, real-time data processing, and additional tools like analytics and AI. IoT technology is expanding into areas such as smart homes, healthcare, industrial automation, and smart cities, driving the ongoing digital transformation of the world.

PLATFORMS:

- Arduino - C program
- Raspberry pi – Python program
- Intel Galileo
- Adafruit
- SparkFun
- ARM mbed
- Particle

ARDUINO UNO:

The Arduino Uno is a widely-used microcontroller board, ideal for beginners. It allows you to control electronic components like LEDs and sensors through simple programming. The board is powered via USB and has pins for easy connections to various components. With the Arduino Uno, you can create projects ranging from basic circuits to more complex systems. It's perfect for learning electronics and coding.

- MicroController – AT mega 328
- Operating Voltage – 5V
- Input Voltage (Recommended) – 7 to 12 V
- Input Voltage (Limit) – 6 to 20 V
- Digital Input, Output Pin – 14 pins (Among 14 Pins, 6 pins are provided to PWM Output)
- Analog Input Pin – 6 Pin
- Flash Memory – 32KB (0.5 KB is used by Boot Loader)
- SRAM – 2KB (AT mega 328)
- Clock Speed – 16 MHZ

IDE – Integrated Developed Environment

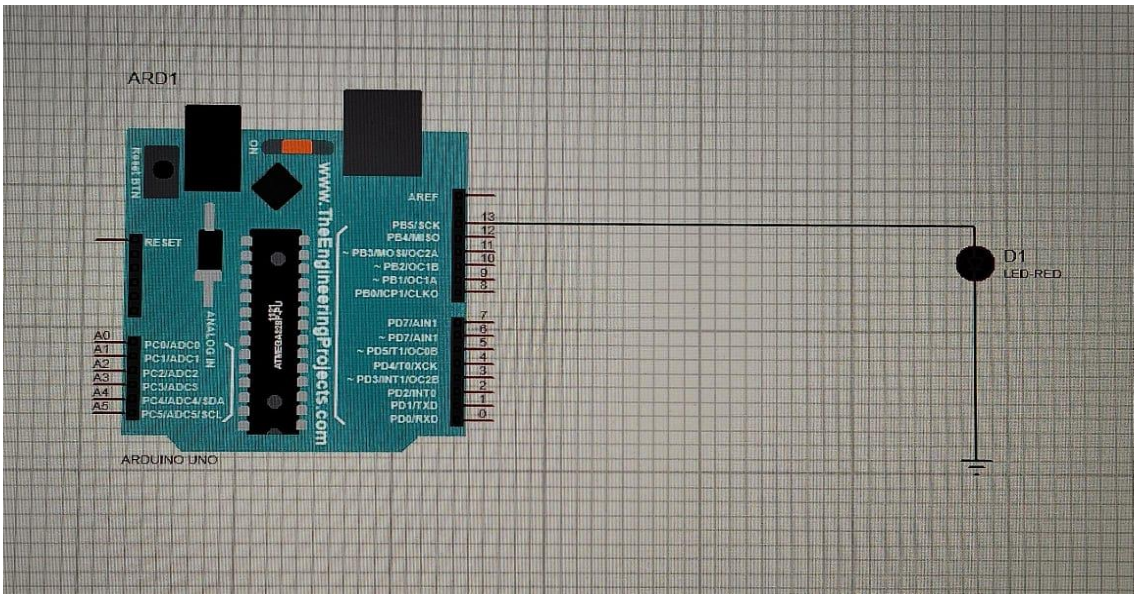
3 Phases are;

1. Controller
2. Verify Status
3. Upload

HANDS-ON PROJECTS (DAY 1)

1. LED BLINK:

An Arduino Uno to control an LED, making it blink at regular intervals. This involves writing a simple program in the Arduino IDE, connecting the LED to the Arduino board, and uploading the program to execute the blink operation.



Materials Required;

- Arduino uno
- Led
- USB cable (Upload)
- Adapter (Power Supply)

Procedure:

1. Set Up the Circuit:

    - Place the LED on the breadboard.
    - Connect the longer leg (anode) of the LED to digital pin 13 on the Arduino Uno.

- Connect the shorter leg (cathode) of the LED to one end of a 220-ohm resistor.
- Connect the other end of the resistor to the ground (GND) pin on the Arduino.
- Use jumper wires to ensure all connections are secure.

2. Write the Code:

```
Int led = 13;
void setup() {
    pinMode(13, OUTPUT); // Set pin 13 as an output pin
   }

   void loop() {
    digitalWrite(13, HIGH); // Turn the LED on
    delay(1000);            // Wait for one second
    digitalWrite(13, LOW);  // Turn the LED off
    delay(1000);            // Wait for one second
   }
```

3. Upload the Code:

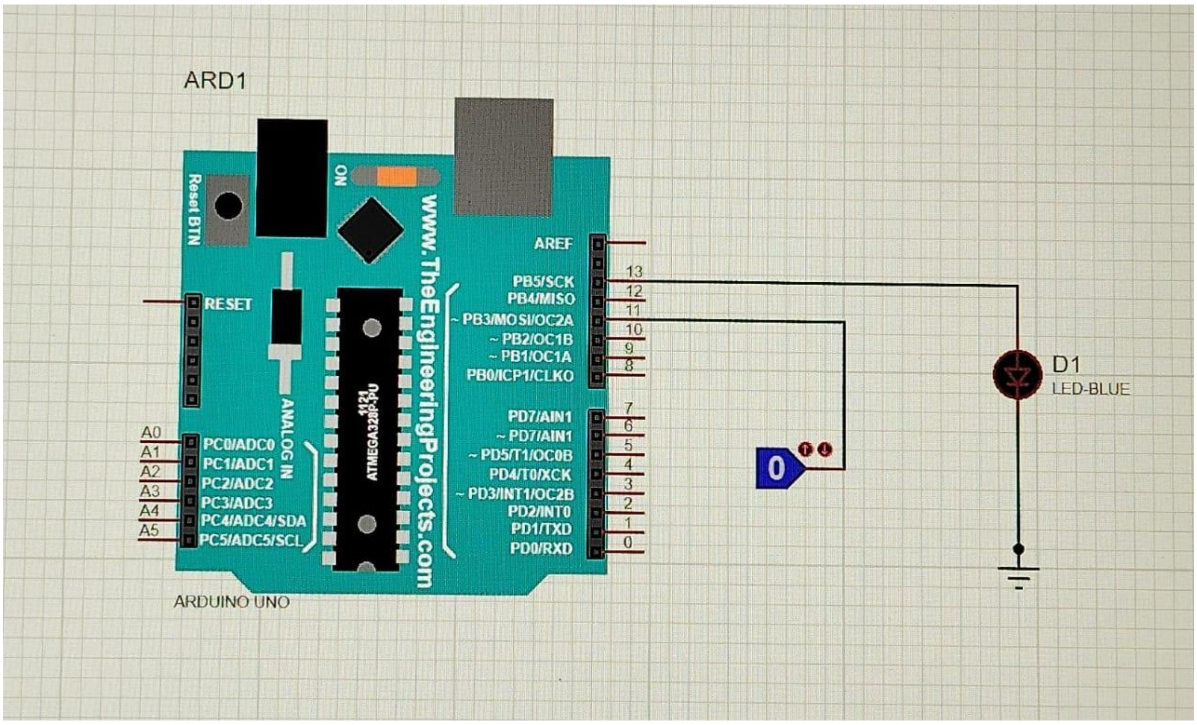
   - Connect the Arduino Uno to your computer using a USB cable.
   - In the Arduino IDE, select the Arduino Uno board and the correct port.
   - Click the upload button in the Arduino IDE to upload the code to the Arduino.

4. Observe the LED:

- After the code is uploaded, the LED should start blinking on and off at one-second intervals.
- Ensure the LED blinks as expected, confirming the project's success.

2. IR SENSORS:

Using an IR sensor to detect the presence of a person and automatically open a door is a practical application of infrared technology. This system enhances convenience and accessibility by allowing doors to open automatically when a person approaches, without the need for physical contact. This report outlines the basic setup and working principle of such a system.



Materials Required;

- Arduino uno
- Led
- IR Sensor
- USB cable (Upload)
- Adapter (Power Supply)

Procedure:

1.Set Up the Circuit:

- Place the LED on the breadboard.
- Connect the longer leg (anode) of the LED to digital pin 13 on the Arduino Uno.
- Connect the shorter leg (cathode) of the LED to one end of a 220-ohm resistor.
- Connect the other end of the resistor to the ground (GND) pin on the Arduino.
- Use jumper wires to ensure all connections are secure.
- Connect the output pin of the IR sensor to a digital input pin on the Arduino (e.g., pin 2).
- Connect the VCC pin of the IR sensor to the 5V pin on the Arduino.
- Connect the GND pin of the IR sensor to a ground pin on the Arduino.

2.Write the Code:

```
int led=13;

int ir=4;


void setup(){
  pinMode(led, OUTPUT);
  pinMode(ir, INPUT);
}
void loop(){
  int A = digitalRead(ir);
  if(A==0){
    digitalWrite(led, HIGH);
  }
  else{
    digitalWrite(led, LOW);
  }
}
```

3. Upload the Code:

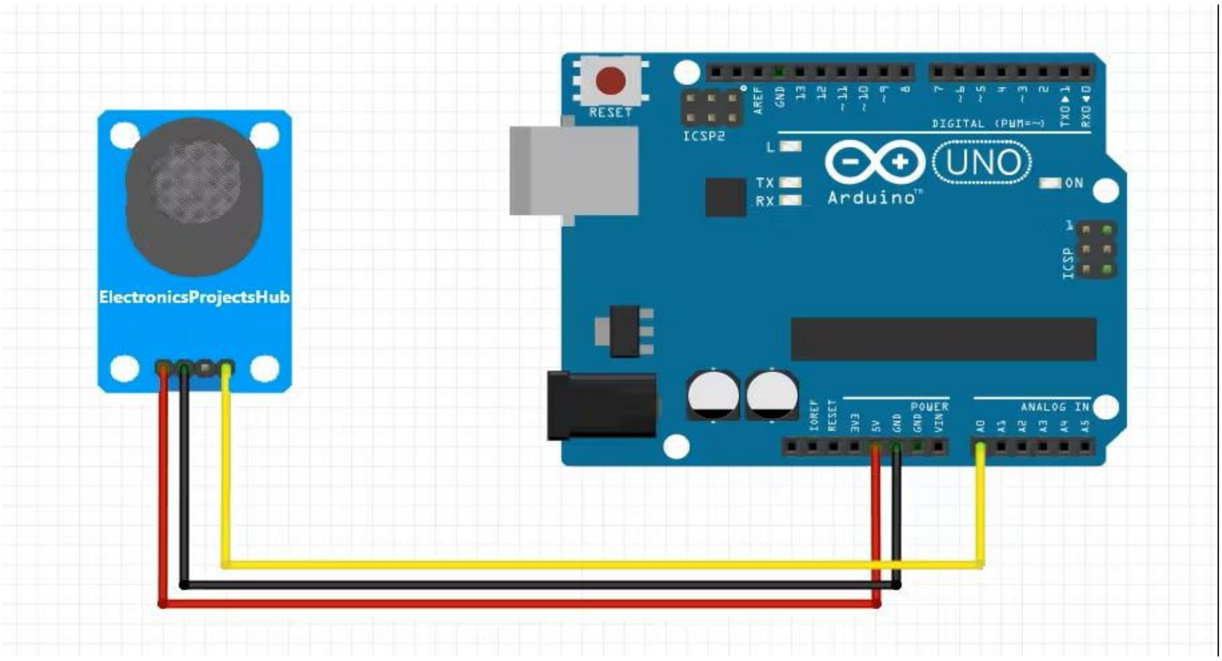
   - Connect the Arduino Uno to your computer using a USB cable.
   - In the Arduino IDE, select the Arduino Uno board and the correct port.
   - Click the upload button in the Arduino IDE to upload the code to the Arduino.

4. Observe the LED:

   - After the code is uploaded, the LED should start blink on sensing the humans using IR Sensor
   - Ensure the LED blinks as expected, confirming the project's success.

3.GAS DETECTOR:

The MQ2 sensor is a versatile gas sensor that can detect various gases, including alcohol. By using the MQ2 sensor, you can create a device that detects the presence of alcohol vapours in the air, making it useful for applications such as breathalyzers. This report outlines the procedure for building a simple alcohol detector using the MQ2 sensor and an Arduino.



Materials Required;

- Arduino uno
- Led
- MQ2 (Alcohol Sensor)
- USB cable (Upload)
- Adapter (Power Supply)

Procedure:

1. Connect the MQ2 Sensor:
   - Connect the VCC pin of the MQ2 sensor to the 5V pin on the Arduino.
   - Connect the GND pin of the MQ2 sensor to a ground pin on the Arduino.
   - Connect the Analog Output (A0) pin of the MQ2 sensor to an analog input pin on the Arduino (e.g., A0).
2. Write the Code:

```
int gas = A0;
void setup(){
 pinMode(gas, INPUT);
 Serial.begin(9600);
}
void loop(){
 int A=analogRead(gas);
 Serial.println(A);
 delay(1000);
}
```

3. Upload the Code:
   - Connect the Arduino Uno to your computer using a USB cable.Select the Arduino Uno board and the correct port.
   - Click the upload button in the Arduino IDE to upload the code to the Arduino.

DAY: 2

DATE: 28.08.2024

COMMUNICATION DEVICES:

In essence, communication devices allow for information and data exchange between different systems, people, or networks altogether. These devices enable communication through media such as radio waves, cables, or wireless networks, and are essential for such fields as telecommunications, networking and personal computing. Among the primary communication devices are smartphones, radios, modems and Bluetooth modules. Bluetooth technology in particular has become extremely popular thanks to its ability to wirelessly link devices at short range--a function that now forms an indispensable part of modern electronics.

Bluetooth is a short-range wireless communication standard that is meant to exchange data between devices over a few feet, typically around 10 meters. It works in the 2.4GHz ISM band and employs frequency-hopping spread-spectrum (FHSS) to reduce interference and increase security. Bluetooth technology is commonly used in multiple applications, such as linking mice, headphones or keyboards to computers and other devices as in the Internet of Things (IoT) for connecting smart home appliances. It is convenient pairing of devices, secure connection protocol and packet-based data transport structure result in a highly efficient and versatile short-range communication method that is easy to use.

HC 05 BLUETOOTH MODULE:

- Bluetooth Protocol - Bluetooth V2.0 + EDR (Enhanced Data Rate).
- Operating Voltage - 3.3V to 5V.
- Communication Interface - UART (Universal Asynchronous Receiver/Transmitter) with a default baud rate of 9600 bps.
- Range - Typically up to 10 meters.
- Master/Slave Configuration - Can be configured to operate as either a master or slave using AT commands.
- Connection Pins:

  1.VCC - Power supply (3.3V or 5V).

  2.GND - Ground.

  3.TXD - Transmit data pin (connects to RXD of the microcontroller).

  4.RXD - Receive data pin (connects to TXD of the microcontroller).

  5.EN - Enable pin to switch between data mode and command mode.
- Status LEDs - Indicates the state of the module (blinking means waiting for a connection, solid means connected).
- Configuration - Adjustable using AT commands for settings such as baud rate, device mode, and Bluetooth name.

- Compatibility -Works with various microcontrollers, including Arduino, Raspberry Pi, and others.

ESP 32:

The ESP32 is a powerful microcontroller that's great for IoT (Internet of Things) projects. It has a fast dual-core processor that can handle complex tasks, and it comes with built-in Wi-Fi and Bluetooth, so it can easily connect to the internet or other devices wirelessly. The ESP32 also has many input and output options, allowing it to work with a variety of sensors and components. It's designed to use very little power, making it perfect for projects that run on batteries. With features like touch sensors and temperature sensors built-in, the ESP32 is a flexible and reliable choice for all sorts of electronic projects.

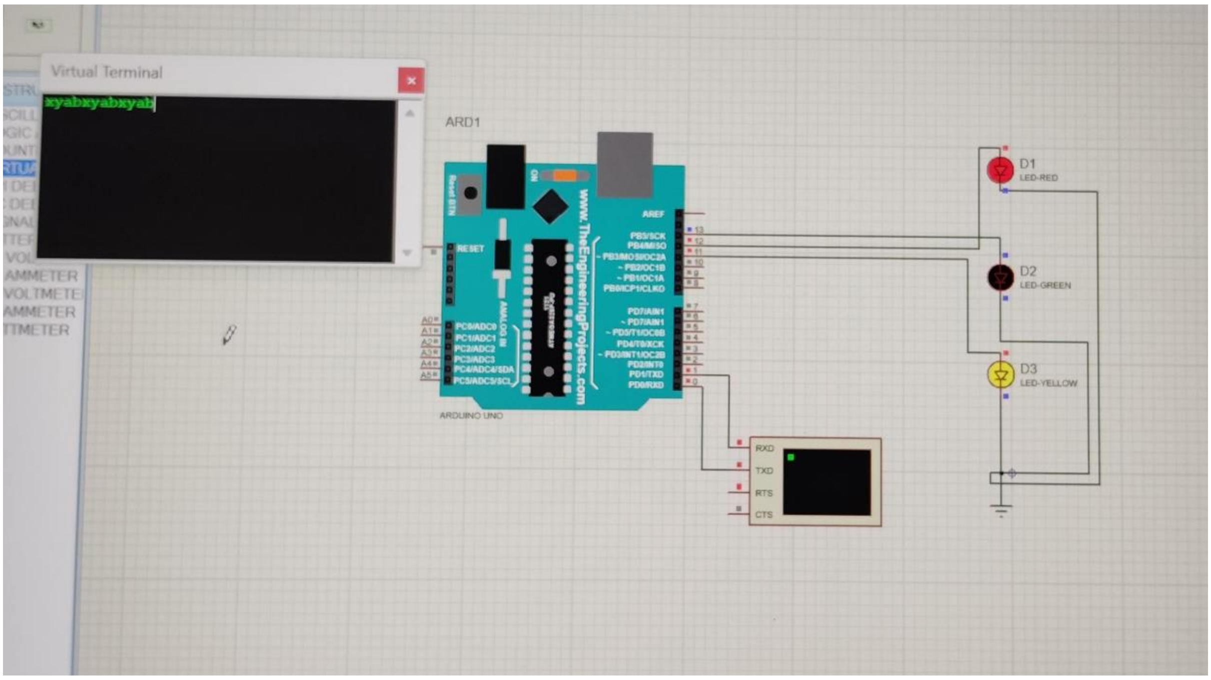
ESP8266:

The ESP8266 is a popular microcontroller that is widely used in IoT (Internet of Things) projects due to its built-in Wi-Fi capabilities. It is known for being cost-effective and easy to use, making it a favourite among hobbyists and developers. The ESP8266 can connect to Wi-Fi networks, allowing devices to communicate over the internet. It has a single-core processor and several GPIO (General Purpose Input/Output) pins for connecting sensors and other components. Despite being less powerful than newer models like the ESP32, the ESP8266 is still a strong choice for projects that require simple internet connectivity, such as home automation or basic data logging. Its small size, low cost, and ease of integration have made it a staple in the world of IoT.

HANDS-ON PROJECTS (DAY 2)

1. BLUETOOTH:

This project demonstrates how to control an LED using a mobile application via a Bluetooth connection with the HC-05 module and an Arduino. The LED will blink when you send a command from the mobile app.



```
xyabxyabxyab
```

Materials Required;

- Arduino uno
- Led
- HC 05
- USB cable (Upload)
- Adapter (Power Supply)
- Mobile App

Procedure:

1. Connect the LED:
   - Connect the longer leg (anode) of the LED to a digital pin on the Arduino (e.g., pin 13) through a 220-ohm resistor.
   - Connect the shorter leg (cathode) of the LED to the ground (GND) on the Arduino.

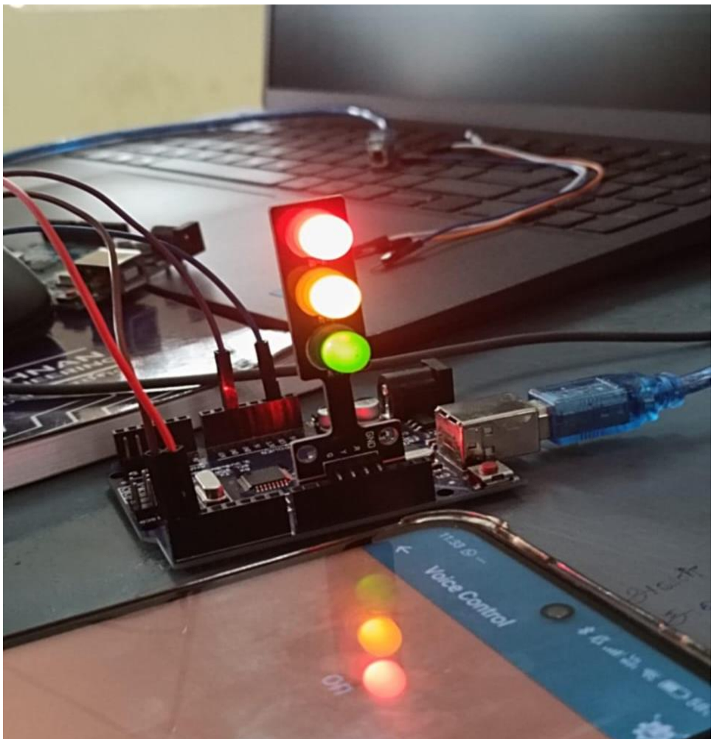
2. Connect the HC-05 Bluetooth Module:
   - Connect the **VCC** pin of the HC-05 to the **5V** pin on the Arduino.
   - Connect the **GND** pin of the HC-05 to the ground (GND) on the Arduino.
   - Connect the **TXD** pin of the HC-05 to the **RX** pin on the Arduino (pin 0).
   - Connect the **RXD** pin of the HC-05 to the **TX** pin on the Arduino (pin 1).
3. Write the Arduino Code:

```
int ledg = 13;
int ledr = 12;
int ledy = 11;
void setup(){
  pinMode(ledg, OUTPUT);
  pinMode(ledr, OUTPUT);
  pinMode(ledy, OUTPUT);
  Serial.begin(9600);
}
void loop(){
  int A= Serial.read();
  if(A=='x'){
    digitalWrite(ledg, HIGH);
    digitalWrite(ledr, HIGH);
    digitalWrite(ledy, HIGH);
  }
  if(A== 'y'){
    digitalWrite(ledg, LOW);
    digitalWrite(ledr, LOW);
    digitalWrite(ledy, LOW);
  }
  else if(A=='a'){
    digitalWrite(ledg, HIGH);
    digitalWrite(ledr, LOW);
    digitalWrite(ledy, LOW);
  }
  else if(A=='b'){
    digitalWrite(ledg, LOW);
    digitalWrite(ledr, HIGH);
    digitalWrite(ledy, HIGH);
  }
}
```

4. Upload the Code:

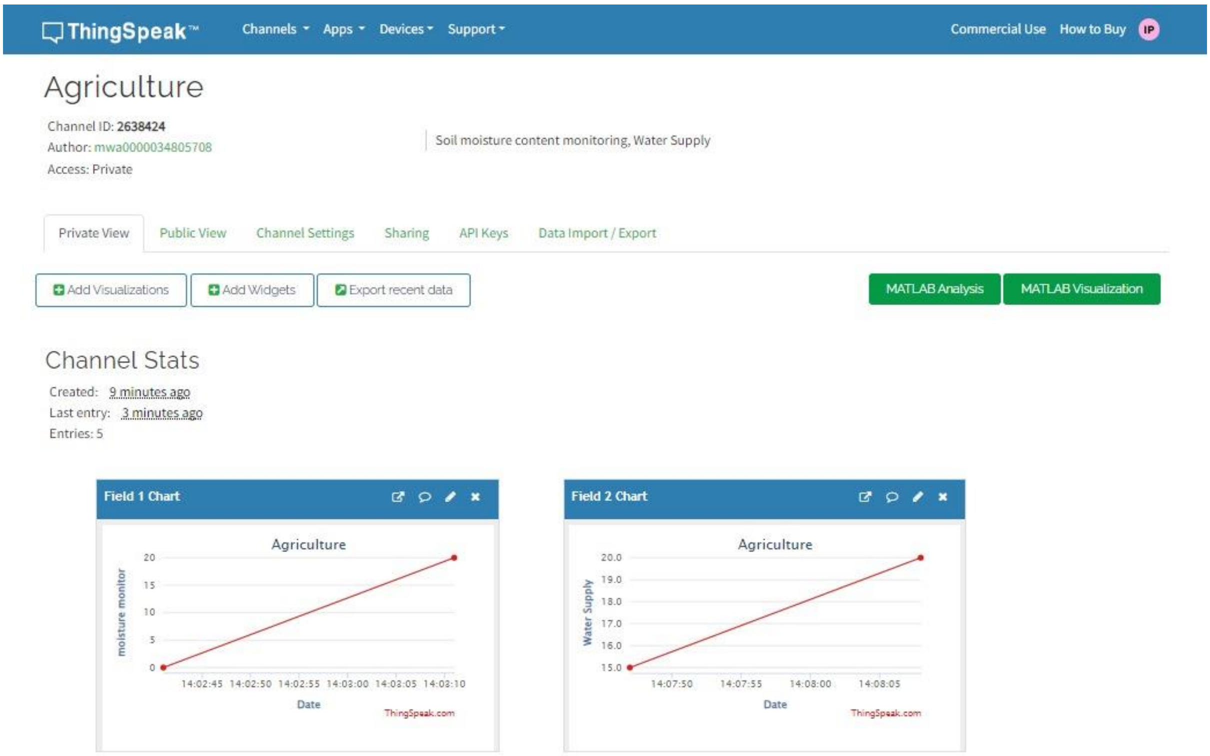
   - Connect the Arduino to your computer using a USB cable.
   - Select the correct board and port in the Arduino IDE.
   - Upload the code to the Arduino.

5. Pair the HC-05 with Your Mobile Phone:
    - Turn on Bluetooth on your phone and search for devices.
    - Pair with the HC-05 (the default pairing code is usually "1234" or "0000").
    - Open the Bluetooth terminal app and connect to the HC05 module.
6. Control the LED:
    - In the Bluetooth terminal app, send the character 'x' to turn on the LED.
    - Send the character 'y' to turn off the LED.



2.ThinkSpeak:

Creating a cloud project in ThingSpeak allows you to collect, store, and visualize data from IoT devices in a centralized platform. To start, you'll need to create a ThingSpeak account and set up a new channel where your data will be stored. This involves naming the channel and defining the data fields you want to monitor, such as temperature or humidity. Next, you'll program your microcontroller, like an Arduino or ESP8266/ESP32, to collect sensor data and send it to your ThingSpeak channel using the provided API. Once data is being sent, you can visualize it on the ThingSpeak dashboard through customizable graphs and widgets. ThingSpeak also offers features like data analysis with MATLAB integration and the ability to set up alerts or trigger actions based on your data. This setup is ideal for projects requiring remote monitoring and real-time data analysis, making it a powerful tool for IoT applications.



Agriculture

Channel ID: 2638424
Author: mwa0000034805708
Access: Private

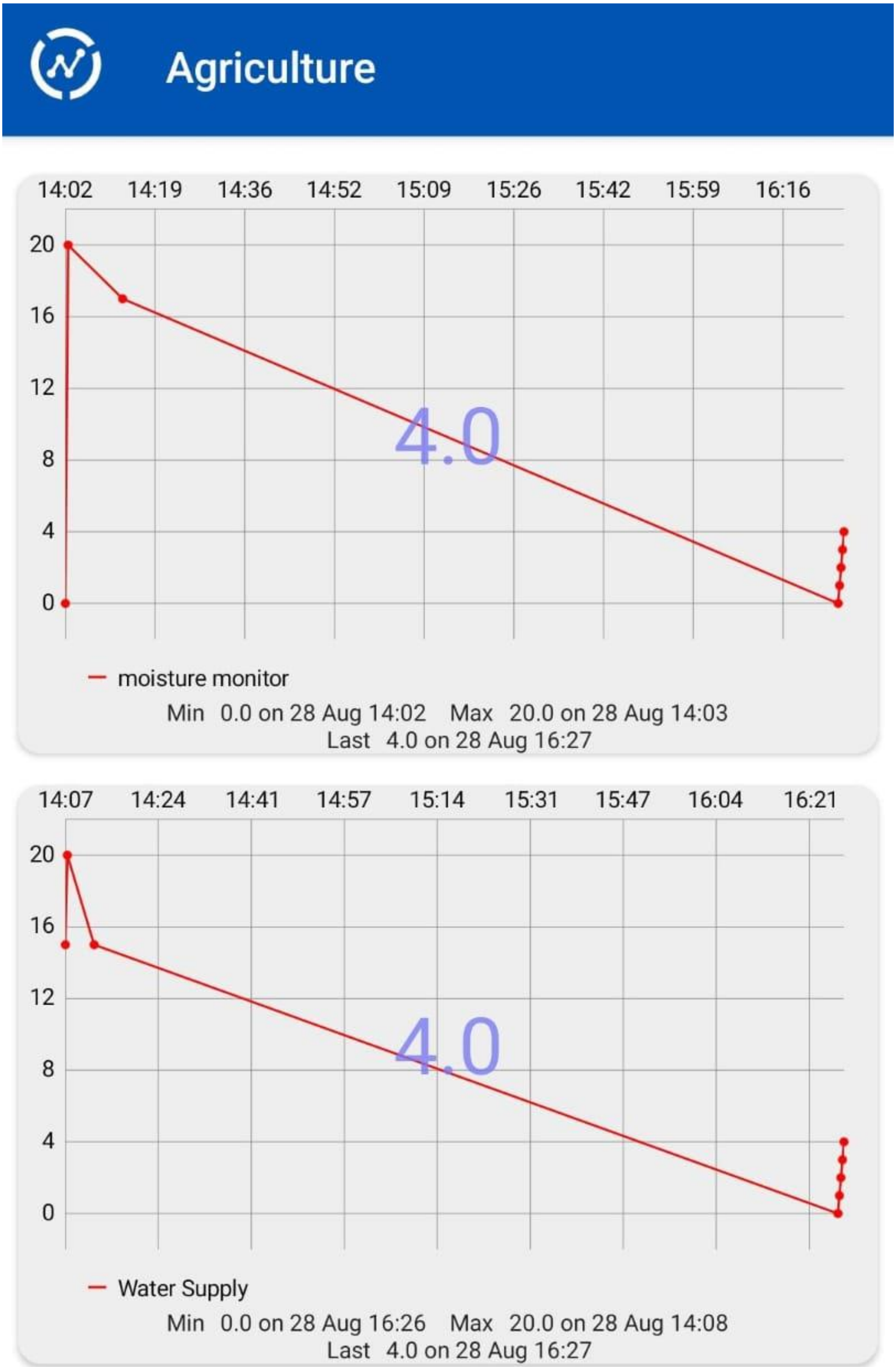
Soil moisture content monitoring, Water Supply

Channel Stats

Created: 9 minutes ago
Last entry: 3 minutes ago
Entries: 5

Write a code:

```
#include <ThingSpeak.h>
#include <ESP8266WiFi.h>
#define LED D0

const char *ssid = "ABCD";     // replace with your wifi ssid
and wpa2 key
const char *pass = "breeze@5";
const char* server = "api.thingspeak.com";

WiFiClient client;
unsigned long myChannelNumber =   2638424; //Your
Channel Number (Without Brackets)
const char * myWriteAPIKey = "DFS07L78D395TY9L";
//Your Write

int A, B;

void setup()
{
  Serial.begin(9600);
  pinMode(LED, OUTPUT);    // LED pin as output.
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED)
  {
    digitalWrite(LED, 1);   // turn the LED on (HIGH is the
voltage level)
    delay(500);                       // wait for a second
    digitalWrite(LED, 0);    // turn the LED off by making the
voltage LOW
    delay(500);
  }

 ThingSpeak.begin(client);
}

void loop()
{
 ThingSpeak.setField( 1, A);
 ThingSpeak.setField( 2, B);
 ThingSpeak.writeFields(myChannelNumber,
myWriteAPIKey);
 delay(15000);
 A++;
 B++;
}
```



Agriculture

— moisture monitor

Min 0.0 on 28 Aug 14:02 Max 20.0 on 28 Aug 14:03
Last 4.0 on 28 Aug 16:27

— Water Supply

Min 0.0 on 28 Aug 16:26 Max 20.0 on 28 Aug 14:08
Last 4.0 on 28 Aug 16:27

CONCLUSION:

In summary, the field of IoT (Internet of Things) leverages a range of technologies to create smarter, interconnected systems. Platforms like ThingSpeak facilitate data collection and visualization from various IoT devices, while microcontrollers such as the Arduino Uno provide a hands-on approach to learning and building simple automation projects. Sensors like IR sensors and gas detectors enhance functionality by detecting environmental changes and ensuring safety. Communication devices, including Bluetooth modules like the HC-05, enable wireless control and data exchange, exemplified by applications such as remotely controlling an LED. Advanced microcontrollers like the ESP32 and ESP8266 further expand possibilities with integrated Wi-Fi and Bluetooth, offering powerful solutions for data transmission and connectivity. Together, these technologies enable innovative and efficient IoT applications, transforming how we interact with and manage connected systems.