Part A

Class B Tech CSE 4th Year **Sub: Internet of Things Lab**

Aim: *Arduino IoT cloud platform to control LED and DHT sensor data display*.

Prerequisite: Basics of programming, microcontrollers and basic electronics

Outcome:

1. Understanding of IoT Cloud and various service providers.
2. Creating things and converting them into smart things over IoT Cloud.
3. Connect and control smart things with the Arduino IoT Cloud.
4. Display the data of various sensors over the Arduino IoT Cloud dashboard

Theory:

1. **Study of IoT Cloud Service Providers:** Analyze key IoT cloud platforms such as Arduino IoT Cloud, AWS, Microsoft Azure IoT, and Blynk, focusing on their functionalities and features.
   - **Arduino IoT Cloud:** A cloud platform designed for users to develop IoT projects with Arduino hardware. It provides capabilities for real-time monitoring, device management, and automation, through a simplified, user-friendly interface.
   - **Microsoft Azure IoT:** A robust cloud solution offering services for device connectivity, data analytics, security, and machine learning, supporting comprehensive management of IoT ecosystems.
2. **LED Control via Arduino IoT Cloud:**
   - **Hardware Setup:** An LED is connected to an Arduino pin, forming a basic circuit that can be controlled using a virtual switch on the Arduino IoT Cloud Dashboard.
   - **Cloud Configuration:** A device (referred to as a "thing") is configured in the Arduino IoT Cloud to control the LED. A variable is assigned to represent the LED's status (on/off) and linked to the dashboard.
   - **Programming:** The Arduino Cloud API is utilized to develop code that maps the LED control variable to the physical pin, enabling the LED to respond to cloud commands sent via the dashboard.
3. **Displaying DHT Sensor Data on the Arduino IoT Cloud Dashboard:**
   - **Cloud Setup:** Two variables (temperature and humidity) are defined in the Arduino IoT Cloud, linked to the data received from the DHT sensor.
   - **Dashboard Display:** Real-time sensor data is transmitted to the Arduino IoT Cloud and displayed on the dashboard using widgets, such as gauges or text boxes, for temperature and humidity readings.
4. **Temperature and Humidity Charting on Arduino IoT Cloud Dashboard:**
   - **Analysis and Monitoring:** A live chart is implemented to monitor temperature and humidity trends, offering valuable insights for projects requiring continuous environmental data tracking.

Part B (Write for an individual)

Steps:

Project 1: LED On/Off Using the Arduino IoT Cloud Platform

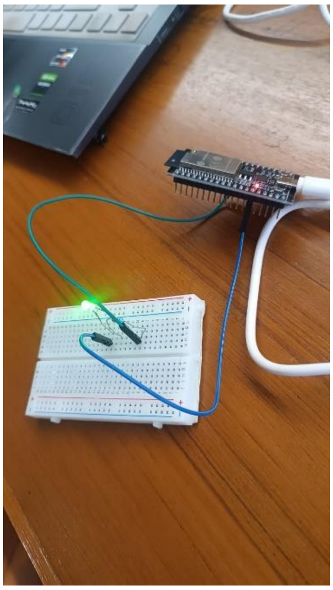
- **Setup the LED circuit**: Connect the LED to a digital pin (e.g., D2) on the Arduino board and the other end to ground (GND).
- **Create an Arduino IoT Cloud account**: Go to the Arduino IoT Cloud platform and create an account if you don't have one.
- **Create a new thing**: In the Arduino IoT Cloud, create a new "thing" and add your Arduino device (e.g., Arduino Uno, Nano, ESP32).
- **Add a variable**: Add a new boolean variable (e.g., ledStatus) that will control the LED's state (true for ON, false for OFF). This will be linked to the LED.
- **Configure a dashboard**: Create a dashboard in Arduino IoT Cloud and add a toggle switch widget that controls the ledStatus variable to turn the LED on/off.
- **Write the Code:**

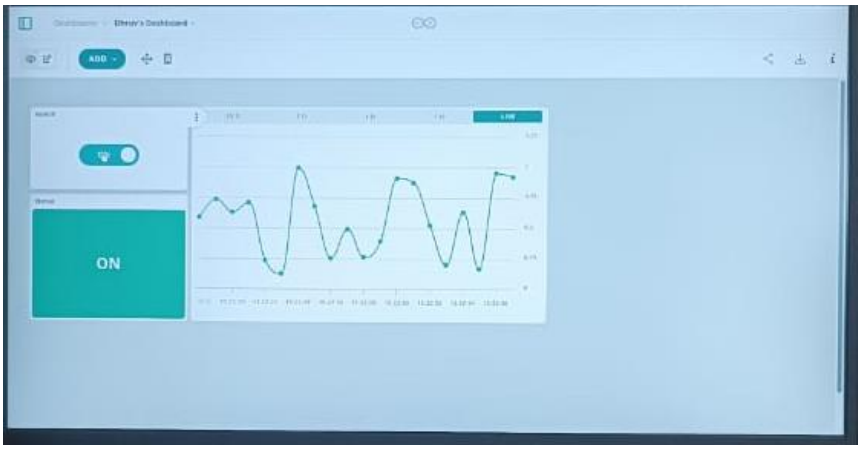
```
 #include "thingProperties.h"  // Include IoT properties

const int ledPin = 2;

void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
  initProperties();
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
}

void loop() {
  ArduinoCloud.update();  // Keep cloud connection alive

  // Control LED based on cloud variable status
 If(ledStatus){
  digitalWrite(ledPin, HIGH );
}else{
digitalWrite(ledPin, LOW);
}
}
```

- **Upload the code and control**: After uploading the code to the Arduino, open the Arduino IoT Cloud Dashboard and use the toggle switch to turn the LED on and off.

.

Output:





Observation & Learning:

1. The I2C interface simplifies the connection of the LCD with the ESP32 and Arduino, using only two wires (SDA and SCL).
2. Mapping the sensor values to readable formats is essential for presenting clear and accurate data on the LCD screen.
3. Real-time data from various sensors (potentiometer and humidity sensor) can be effectively displayed using libraries like LiquidCrystal_I2C.
4. Using cloud platforms like Arduino IoT Cloud makes it easy to control devices and visualize sensor data remotely.
5. Creating interactive dashboards with widgets (switches, gauges, charts) enhances monitoring and control of IoT devices.
6. Arduino IoT Cloud provides real-time data visualization tools, making it useful for environmental monitoring projects.

Conclusion:

The integration of IoT platforms like Arduino IoT Cloud with sensors and devices like LEDs, potentiometers, and DHT sensors offers a powerful way to create real-time, interactive projects. By using simple libraries and cloud-based dashboards, even beginners can set up systems for data monitoring and remote control. Displaying sensor data such as temperature, humidity, and user-controlled devices on the IoT Cloud platform provides insight into the flexibility and scalability of IoT systems.