

IOT for beginners

Module 1: Introduction to IoT

► 1.1 What is IoT?

- **Definition:** IoT (Internet of Things) is a system of interrelated devices that are connected to the internet, capable of collecting and exchanging data.
 - Example: A smart thermostat adjusts your home temperature automatically based on weather conditions and your preferences.
- **Key Components:**
 - **Devices:** Sensors, actuators, and microcontrollers.
 - **Connectivity:** Wi-Fi, Bluetooth, Zigbee, LoRa, etc.
 - **Cloud:** For data storage and analytics.
 - **User Interface:** Apps or dashboards for monitoring and controlling.



► 1.2 Why IoT Matters?

- **Economic Impact:** IoT is expected to contribute billions to the global economy by increasing efficiency in industries like healthcare, agriculture, and logistics.
- **Benefits:**
 - Automation: Saves time and effort.
 - Real-time monitoring: Improves decision-making.
 - Cost savings: Reduces operational costs.
 - Innovation: Enables new business models.



▶ 1.3 IoT Ecosystem Overview

- **Perception Layer:** Sensors and actuators to collect and respond to data.
- **Network Layer:** Connects devices using protocols like MQTT or HTTP.
- **Application Layer:** Provides user-friendly interfaces for control and visualization.

▶ **Activity:**

- Show videos of real-life IoT applications like:
 - Smart city traffic lights.
 - Smart refrigerators.
 - Wearable health devices (e.g., Fitbit).

Module 2: IoT Hardware Basics

▶ 2.1 Common IoT Devices and Sensors

- **Microcontrollers:**

- Arduino: Basic, easy-to-use microcontroller for beginners.
- ESP32: IoT-friendly board with built-in Wi-Fi and Bluetooth.
- Raspberry Pi: A mini-computer for advanced projects.

- **Sensors:**

- **Temperature and Humidity:** DHT11, DHT22.
- **Motion:** PIR sensor.
- **Light:** LDR (Light Dependent Resistor).
- **Air Quality:** MQ2 gas sensor.

- **Actuators:**

- LEDs for indicators.
- Servo motors for movement.



► 2.2 IoT Communication Interfaces

- **GPIO Pins:** For connecting sensors and actuators.
- **I2C:** Simplifies multiple device communication with fewer wires.
- **SPI:** Used for high-speed communication between microcontrollers and peripherals.
- **UART:** Serial communication for debugging or basic data transfer.

2.3 Hands-On Activity

•Blink an LED Using Arduino:

- Components: Arduino Uno, LED, resistor, breadboard.
- Code:

```
void setup() {  
  pinMode(13, OUTPUT); // Set pin 13 as output  
}  
void loop() {  
  digitalWrite(13, HIGH); // Turn LED on  
  delay(1000); // Wait for 1 second  
  digitalWrite(13, LOW); // Turn LED off  
  delay(1000); // Wait for 1 second  
}
```

- Output: LED blinks on and off every second.

Module 3: IoT Networking Basics

► 3.1 IoT Communication Protocols

- **Short-range protocols:**

- **Bluetooth:** For wearables and smart devices.
- **Zigbee:** Low-power protocol for home automation.

- **Long-range protocols:**

- **LoRaWAN:** Low-power wide-area network for IoT.
- **Cellular:** IoT over 4G/5G for high data transmission.

- **Application Layer Protocols:**

- **HTTP:** For web-based communication.
- **MQTT:** Lightweight protocol widely used in IoT for sending data.



▶ 3.2 Network Architecture

- **Edge Layer:** Sensors, actuators, and microcontrollers.
- **Gateway Layer:** Converts data into formats suitable for the cloud.
- **Cloud Layer:** Performs storage, analytics, and decision-making.

3.3 Hands-On Activity

Connect ESP32 to Wi-Fi and send a temperature reading to an MQTT broker using this code snippet

```
#include <WiFi.h>

#include <PubSubClient.h>

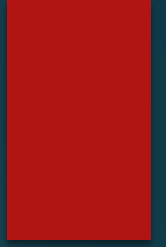
const char* ssid = "YourWiFiSSID";
const char* password = "YourWiFiPassword";
const char* mqtt_server = "broker.hivemq.com";

WiFiClient espClient;
PubSubClient client(espClient);

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  client.setServer(mqtt_server, 1883);
}

void loop() { if (!client.connected()) { reconnect(); } client.loop();
client.publish("iot/test", "Hello from ESP32"); delay(5000); }
```

Module 4: IoT Software Platforms



▶ 4.1 Introduction to IoT Platforms

- Overview of IoT platforms:
 - **Google Cloud IoT**: Advanced, scalable solution.
 - **AWS IoT Core**: Integrates with Amazon services.
 - **Thingspeak**: Simple, beginner-friendly platform.

▶ 4.2 Interfacing Devices with Platforms

- Steps:
 - Collect sensor data.
 - Send data to the cloud using MQTT or REST APIs.
 - Visualize data using dashboards.

▶ 4.3 Hands-On Activity

- Use **Thingspeak** to visualize temperature data.
 - Send data from an ESP32/Arduino to Thingspeak using their API.

Module 5: Data Handling in IoT

► 5.1 Importance of Data

- Why data matters:
 - Monitor performance.
 - Predict failures.
 - Optimize operations.

► 5.2 Big Data and IoT

- IoT systems generate terabytes of data daily.
- Use of machine learning for insights.

► 5.3 Hands-On Activity

- Log temperature and humidity data from a DHT11 sensor and plot it on Thingspeak

Module 6: Security and Privacy in IoT

▶ 6.1 Challenges in IoT Security

- Vulnerabilities:
 - Lack of updates.
 - Unsecured communication.
- Solutions:
 - Encrypt data (e.g., TLS).
 - Regular firmware updates.

▶ 6.2 Privacy Concerns

- Data ownership and protection laws like GDPR.

▶ 6.3 Hands-On Activity

- Secure an MQTT connection using a username and password.

Module 7: IoT Applications

▶ 7.1 IoT in Action

- **Smart Cities:**
 - Sensors for traffic optimization.
- **Smart Agriculture:**
 - Automated irrigation systems.

▶ 7.2 Hands-On Activity

- Build a simple weather monitoring system using an ESP32 and DHT11.

Module 8: Final Project

▶ 8.1 Project Development

- Example: Build a smart home lighting system with an app to control LEDs.