HTTP & TCP in Practice:

IIT Ropar Minor in AI

27th May, 2025

Contents

1 Introduction to HTTP and TCP

1.1 HTTP (Hypertext Transfer Protocol)

HTTP Definition

HTTP defines **what to send** - it's an application layer protocol that specifies the format and rules for communication between clients and servers.

1.1.1 HTTP Methods

HTTP provides several methods to interact with servers:

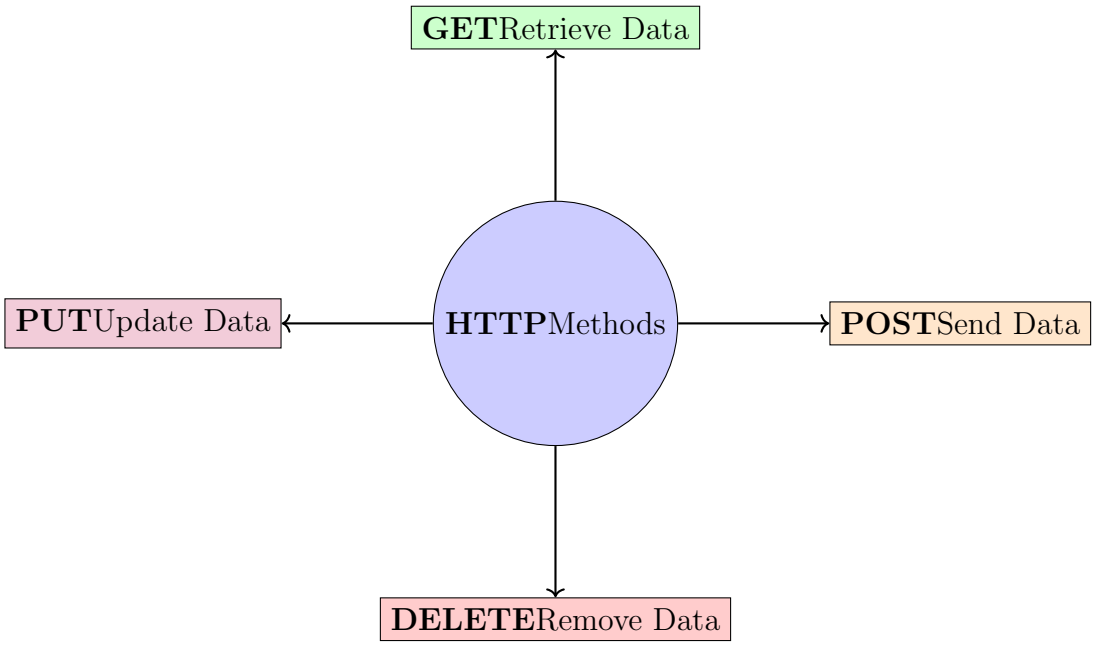

Figure 1: HTTP Methods Overview

Method	Purpose	IoT Usage Example
GET	Retrieve data from server	Reading sensor configurations, downloading firmware
POST	Send data to server	**Most common for IoT** - sending sensor readings
PUT	Update existing data	Updating device settings, calibration values
DELETE	Remove data from server	Clearing old logs, removing device registrations

Table 1: HTTP Methods in IoT Context

1.2 TCP (Transmission Control Protocol)

TCP Definition

TCP ensures **how data is reliably sent** - it's a transport layer protocol that provides reliable, ordered delivery of data between applications.

1.2.1 TCP Features

1. **Connection-oriented**: Establishes a connection before data transfer
2. **Reliable delivery**: Guarantees data reaches the destination
3. **Error detection and correction**: Detects and retransmits lost packets
4. **Flow control**: Manages data flow to prevent overwhelming the receiver
5. **Congestion control**: Adjusts transmission rate based on network conditions

1.3 HTTP-TCP Duo in IoT

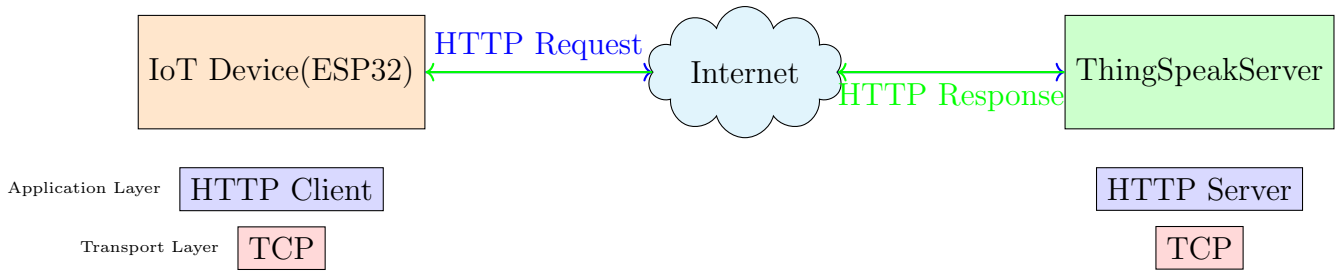

Figure 2: HTTP-TCP Communication in IoT

2 ThingSpeak Platform Setup

2.1 Step 1: ThingSpeak Account Creation

ThingSpeak Access

Website: `thingspeak.mathworks.com`
ThingSpeak is a cloud-based IoT analytics platform service that allows you to aggregate, visualize, and analyze live data streams.

2.1.1 Account Setup Process

1. Navigate to `thingspeak.mathworks.com`
2. Click on "Get Started For Free"
3. Create a MathWorks account or sign in
4. Verify your email address

5. Access the ThingSpeak dashboard

2.2 Step 2: Create New Channel

2.2.1 Channel Configuration

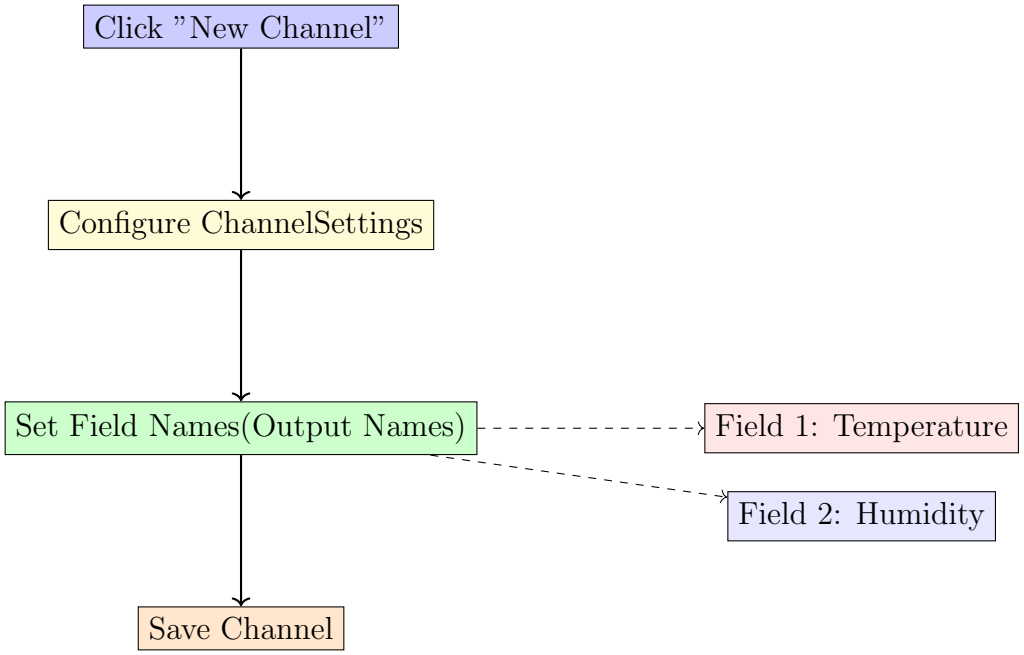

Figure 3: ThingSpeak Channel Creation Process

Example Channel Configuration

- **Channel Name**: "IoT Environmental Monitor"
- **Description**: "Temperature and Humidity monitoring using ESP32"
- **Field 1**: "Temperature" (Units: °C)
- **Field 2**: "Humidity" (Units: %)
- **Tags**: IoT, ESP32, DHT22, Environmental

2.3 Step 3: Dashboard and API Configuration

2.3.1 Obtaining API Credentials

After creating your channel, you'll find:

Credential	Example Value
Channel ID	2972141
Write API Key	R1QV50YBTII5BBZ9
Read API Key	(Generated automatically)

Table 2: ThingSpeak API Credentials

Security Note

Keep your Write API Key secure! This key allows writing data to your channel. Never share it publicly or commit it to public repositories.

2.4 Step 4: Make Channel Public

2.4.1 Sharing Configuration

1. Go to your channel settings
2. Click on "Sharing" tab
3. Check "Share channel view with everyone"
4. Save settings

This allows others to view your data visualizations without needing API keys.

3 Wokwi Simulation Environment

3.1 Step 1: Wokwi Account Setup

Wokwi Simulator

Website: `www.wokwi.com`
Wokwi is an online electronics simulator that supports Arduino, ESP32, Raspberry Pi Pico, and other microcontrollers.

3.1.1 Getting Started with Wokwi

1. Navigate to `www.wokwi.com`
2. Sign up for a free account
3. Choose from:
   - Featured Projects (pre-built examples)
   - Start from Scratch (blank project)
   - Latest Projects (community projects)

3.2 Step 2: ESP32 Project Creation

3.2.1 Project Setup

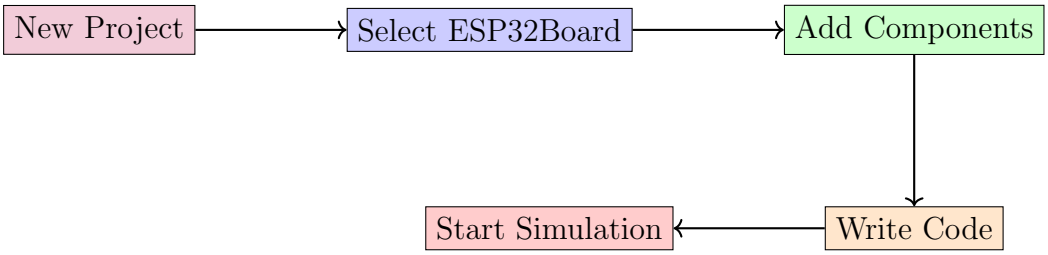

Figure 4: Wokwi Project Creation Workflow

3.3 Step 3: Component Selection and Circuit Design

3.3.1 Required Components

Component	Purpose	Wokwi Part ID
ESP32 DevKit	Main microcontroller	wokwi-esp32-devkit-v1
DHT22	Temperature & Humidity sensor	wokwi-dht22
LED	Status indicator	wokwi-led
Resistor (220)	LED current limiting	wokwi-resistor

Table 3: Required Components for IoT Project

3.3.2 Circuit Connection Diagram

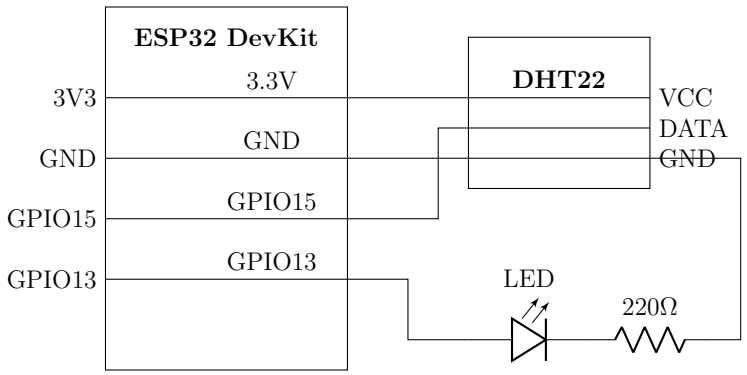

Figure 5: ESP32 Circuit Connections for IoT Environmental Monitor

4 Complete Arduino Code Implementation

4.1 Library Inclusions and Declarations

```
#include <WiFi.h>
#include "DHTesp.h"
```

```
#include "ThingSpeak.h"

// Pin definitions
const int DHT_PIN = 15;
const int LED_PIN = 13;

// WiFi credentials
const char* WIFI_NAME = "Wokwi-GUEST";
const char* WIFI_PASSWORD = "";

// ThingSpeak configuration
const int myChannelNumber = 2972141;
const char* myApiKey = "R1QV50YBTII5BBZ9";
const char* server = "api.thingspeak.com";

// Object instantiations
DHTesp dhtSensor;
WiFiClient client;
```

Listing 1: Complete ESP32 Code for ThingSpeak Integration

4.2 Setup Function - Initialization

```
void setup() {
    // Initialize serial communication
    Serial.begin(115200);

    // Initialize DHT sensor
    dhtSensor.setup(DHT_PIN, DHTesp::DHT22);

    // Initialize LED pin
    pinMode(LED_PIN, OUTPUT);

    // Connect to WiFi
    WiFi.begin(WIFI_NAME, WIFI_PASSWORD);
    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.println("WiFi not connected");
    }

    // WiFi connection successful
    Serial.println("WiFi connected!");
    Serial.println("Local IP: " + String(WiFi.localIP()));

    // Set WiFi mode and initialize ThingSpeak
    WiFi.mode(WIFI_STA);
    ThingSpeak.begin(client);
}
```

Listing 2: Setup Function Implementation

4.3 Main Loop - Data Collection and Transmission

```
void loop() {
    // Read temperature and humidity data
    TempAndHumidity data = dhtSensor.getTempAndHumidity();
```

```
    // Set ThingSpeak fields
    ThingSpeak.setField(1, data.temperature);
    ThingSpeak.setField(2, data.humidity);

    // LED control based on sensor readings
    if (data.temperature > 35 || data.temperature < 12 ||
        data.humidity > 70 || data.humidity < 40) {
        digitalWrite(LED_PIN, HIGH);  // Alert condition
    } else {
        digitalWrite(LED_PIN, LOW);   // Normal condition
    }

    // Send data to ThingSpeak
    int x = ThingSpeak.writeFields(myChannelNumber, myApiKey);

    // Display sensor readings
    Serial.println("Temp: " + String(data.temperature, 2) + " C ");
    Serial.println("Humidity: " + String(data.humidity, 1) + "%");

    // Check if data was sent successfully
    if (x == 200) {
        Serial.println("Data pushed successfully");
    } else {
        Serial.println("Push error: " + String(x));
    }

    Serial.println("---");
    delay(5000);  // Wait 5 seconds before next reading
}
```

Listing 3: Main Loop Implementation

4.4 Code Analysis and Explanation

4.4.1 Library Functions

Library	Function	Purpose
WiFi.h	WiFi.begin()	Connects to wireless network
	WiFi.status()	Checks connection status
	WiFi.localIP()	Gets assigned IP address
DHTesp.h	dhtSensor.setup()	Initializes DHT sensor
	getTempAndHumidity()	Reads sensor data
ThingSpeak.h	ThingSpeak.begin()	Initializes ThingSpeak client
	setField()	Sets data field values
	writeFields()	Sends data to server

Table 4: Key Library Functions Used

4.4.2 Alert Logic Implementation

The LED alert system activates when:

- Temperature $> 35C$ (too hot)
- Temperature $< 12C$ (too cold)
- Humidity $> 70\%$ (too humid)
- Humidity $< 40\%$ (too dry)

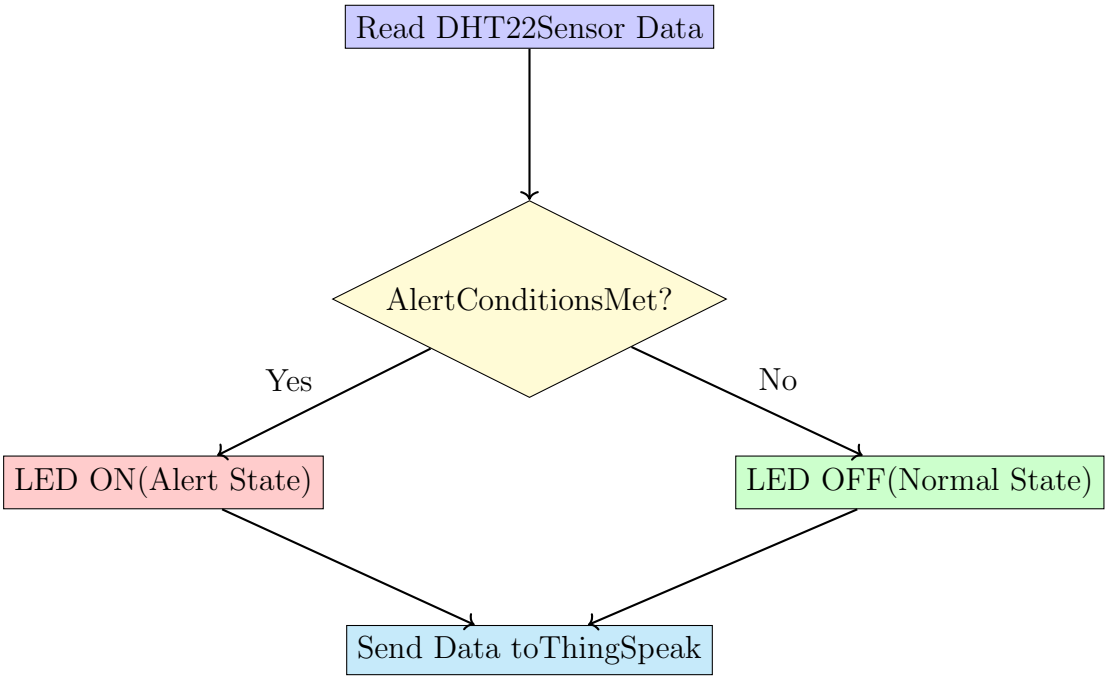

Figure 6: Alert Logic Flow Diagram

5 Library Installation and Dependencies

5.1 Required Libraries in Wokwi

Wokwi Library Installation

Wokwi automatically handles most library installations. However, you may need to add some libraries manually through the Library Manager.

5.1.1 Core Libraries

1. **WiFi Library** - Usually pre-installed with ESP32 board package
2. **DHTesp Library** - For DHT22 sensor communication
3. **ThingSpeak Library** - For cloud data transmission

5.1.2 Library Manager Access

1. Click on the "Library Manager" icon in Wokwi
2. Search for required libraries:
   - "DHTesp" by beegee_tokyo
   - "ThingSpeak" by MathWorks
3. Install the libraries
4. Restart the simulation if needed

6 Data Visualization and Dashboard

6.1 ThingSpeak Visualization Setup

6.1.1 Step 8: Adding Widgets

ThingSpeak provides various visualization widgets:

Widget Type	Best For	Configuration
Gauge	Current value display	Min/Max values, color zones
Line Chart	Time-series trends	Time range, multiple fields
Numeric Display	Simple value showing	Decimal places, units
Bar Chart	Comparative data	Aggregation period

Table 5: ThingSpeak Widget Types

6.1.2 Step 9: Field Declaration and Gauge Setup

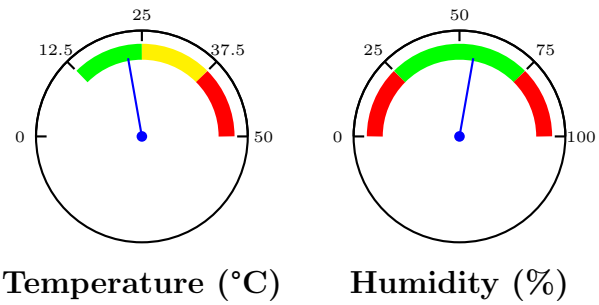

Figure 7: ThingSpeak Gauge Visualization for Temperature and Humidity

6.1.3 Gauge Configuration Parameters

Temperature Gauge Settings

- **Field**: Field 1 (Temperature)
- **Min Value**: 0°C
- **Max Value**: 50°C
- **Color Zones**:
  - Green: 12-35°C (Normal range)
  - Yellow: 35-40°C (Warm warning)
  - Red: 40+°C or ¡12°C (Alert conditions)

Humidity Gauge Settings

- **Field**: Field 2 (Humidity)
- **Min Value**: 0%
- **Max Value**: 100%
- **Color Zones**:
  - Red: 0-40% (Too dry)
  - Green: 40-70% (Comfortable range)
  - Red: 70-100% (Too humid)