Integration of Tektork IoT kit with ThingSpeak

1. Introduction

1.1. Overview of ESP32 in IoT Systems

The ESP32 is a versatile and powerful microcontroller featuring integrated Wi-Fi and Bluetooth capabilities, making it a popular choice for Internet of Things (IoT) applications. It supports a broad range of sensors and peripherals and provides robust wireless communication, which is essential for connecting embedded devices to cloud platforms and local servers. The ESP32's low power consumption, rich peripheral sets, and extensive software ecosystem enable developers to implement diverse IoT solutions, including data acquisition, device control, and remote monitoring. This microcontroller bridges the gap between physical sensors and IoT software frameworks such as MQTT, Node-RED, and cloud services like ThingSpeak and Firebase, facilitating seamless device-to-cloud integration and real-time data visualization.

1.2. Objective

This manual aims to guide users through the comprehensive setup, programming, and deployment of various IoT projects by interfacing Tektork IoT Kit with multiple popular platforms and technologies including ThingSpeak, MQTT brokers, Firebase cloud databases, Node-RED for flow-based programming, and the TIG Stack (Telegraf, InfluxDB, Grafana) for advanced data storage, monitoring and visualization. The objective is to provide a step-by-step reference combining hardware setup, firmware development, platform configuration, and dashboard creation, enabling readers to design scalable, networked IoT systems from scratch with detailed technical insights and best practices.

1.3. Required Hardware and Software

Hardware Requirements

For this manual and related projects, the hardware setup is kept minimal and focused on essential components:

- Tektork IoT Kit: The core microcontroller module with built-in Wi-Fi and Bluetooth connectivity, serving as the primary IoT device.
- USB Type-C Cable: Used to both power the ESP32 board and program it from a PC or laptop.
- PC or Laptop: Required for software development, programming, and interfacing with cloud platforms and local systems.

This simplified hardware setup optimizes project accessibility, making it feasible for users to start IoT development with the Tektork IoT Kit, connecting cable, and a host computer without additional peripheral sensors or modules.

Software Requirements:

The software environment consists of tools and platforms necessary for programming the ESP32, managing MQTT communications, cloud data aggregation, and IoT visualization:

- Arduino IDE or ESP-IDF: Primary development environments for writing and uploading firmware to the ESP32 microcontroller.
- MQTT Broker Service: Public brokers like HiveMQ or Mosquitto provide the messaging backbone for IoT device communication.
- ThingSpeak Account: A cloud-based IoT data platform for acquiring, storing, and visualizing data from connected devices.
- Firebase Console: Google's cloud service offering a real-time database, authentication, and hosting for IoT backend needs.
- Node-RED: A visual flow-based programming tool for creating IoT workflows and dashboards without coding complexity.
- TIG Stack Components:
  Telegraf: An agent for collecting and processing time-series data.
  InfluxDB: A time-series optimized database for storing sensor data.
  Grafana: A dashboard platform to visualize and analyze IoT data effectively.
- Supporting Libraries and Tools: MQTT client libraries for ESP32, HTTP client libraries, and SDKs compatible with ESP32 and relevant platforms.

This software setup targets versatility and scalability, enabling development from basic sensor data visualization up to advanced real-time monitoring and control systems.

2. Setting Up the Tektork IoT Kit Development Environment

Refer the Document tilted

a) Tektork-Kit-ArduinoIDE-Driver-Installation-Notes.pdf

b) Tektork-IoT-Board-Hardware-Manual-V1.0.pdf

3. ThingSpeak with ESP32

3.1. Introduction to ThingSpeak

ThingSpeak is an IoT cloud platform that enables wireless data monitoring, collection, and visualization from sensors connected to microcontrollers such as the ESP32. It allows users to send sensor data via HTTP requests and provides real-time dashboards to monitor values like temperature, light intensity, and other environmental metrics. ThingSpeak's API simplifies integrating IoT devices with cloud analytics and visualization tools.

3.2. Creating a ThingSpeak Account and Channel

To start using ThingSpeak:

- Create a free account on the ThingSpeak website.
- After login, create a new channel by navigating to Channels > New Channel.
- Enter a channel name and define fields representing your sensor data, such as Field 1: Potentiometer and Field 2: LDR Sensor.

- Save the channel and copy its Write API Key. This key is essential for ESP32 to send data securely to the channel.

3.3. ESP32 Code to Send Sensor Data to ThingSpeak

Below is the ESP32 Arduino code that reads sensor data connected to GPIO pins (potentiometer on GPIO36, LDR on GPIO39) and uploads it to ThingSpeak every 20 seconds:

```
#include <WiFi.h>
#include <HTTPClient.h>

// Wi-Fi credentials
const char* ssid = "YourWiFiSSID";
const char* password = "YourWiFiPassword";

// ThingSpeak settings
const String writeAPIKey = "YOUR_THINGSPEAK_WRITE_API_KEY";
const char* thingSpeakServer = "http://api.thingspeak.com/update";

// Sensor pins (use GPIOs with ADC capability)
const int potPin = 36;
const int ldrPin = 39;

// Timer for ThingSpeak updates
unsigned long lastThingSpeakUpdate = 0;
const unsigned long updateInterval = 20000; // 20 seconds

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  Serial.print("Connecting to WiFi...");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println(" connected");
}

void loop() {
  if (millis() - lastThingSpeakUpdate > updateInterval) {
    if (WiFi.status() == WL_CONNECTED) {

      // Read sensor values from analog pins
      int potValue = analogRead(potPin);
      int ldrValue = analogRead(ldrPin);

      // Prepare POST data payload for ThingSpeak
      String postData = "api_key=" + writeAPIKey +
                        "&field1=" + String(potValue) +
                        "&field2=" + String(ldrValue);

      HTTPClient http;
```

```
      http.begin(thingSpeakServer);
      http.addHeader("Content-Type", "application/x-www-form-urlencoded");

      int httpResponseCode = http.POST(postData);

      if (httpResponseCode > 0) {
        Serial.println("Data sent to ThingSpeak");
      } else {
        Serial.print("Error sending data: ");
        Serial.println(httpResponseCode);
      }
      http.end();
    } else {
      Serial.println("WiFi disconnected");
    }
    lastThingSpeakUpdate = millis();
  }
}
```

```
Output  Serial Monitor ×
Message (Enter to send message to 'DOIT ESP32 DEVKIT V1' on 'COM5')
11:02:10.253 -> Data sent to ThingSpeak
11:02:39.044 -> ....... connected
11:02:59.626 -> Data sent to ThingSpeak
11:03:21.029 -> Data sent to ThingSpeak
11:03:42.458 -> Data sent to ThingSpeak
11:04:03.769 -> Data sent to ThingSpeak
11:04:24.915 -> Data sent to ThingSpeak
11:04:46.612 -> Data sent to ThingSpeak
11:05:10.864 -> Data sent to ThingSpeak
11:05:32.498 -> Data sent to ThingSpeak
11:05:54.028 -> Data sent to ThingSpeak
11:06:15.302 -> Data sent to ThingSpeak
11:06:36.597 -> Data sent to ThingSpeak
11:06:57.886 -> Data sent to ThingSpeak
11:07:19.220 -> Data sent to ThingSpeak
```

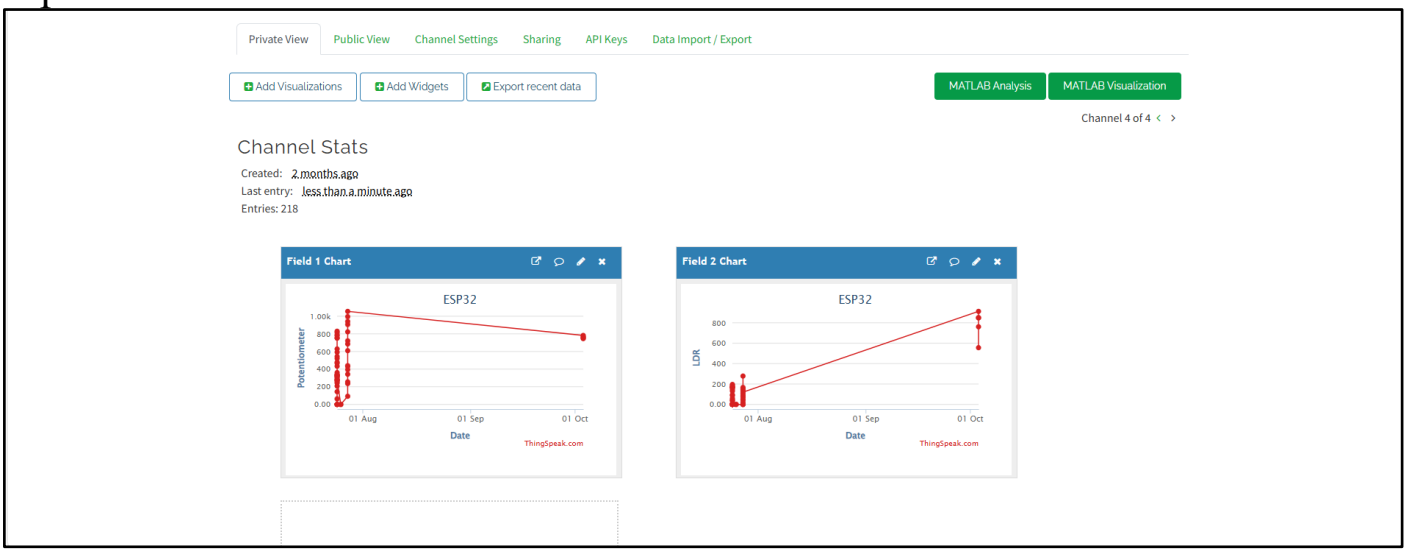
This code continuous connects the ESP32 to Wi-Fi, reads analog sensor values, and sends them to ThingSpeak for cloud storage and visualization.

3.4. Data Visualization on ThingSpeak Dashboard

Once sensor data is sent to ThingSpeak:

- Open your ThingSpeak channel dashboard online.
- Visual graphs will automatically display the incoming values for each defined field.
- Customize the dashboard to include widgets such as line charts, gauges, and numeric displays.
- Real-time updates allow monitoring environmental changes immediately.

ThingSpeak supports MATLAB integration for advanced data analysis capabilities if needed.



3.5. Working Model Demonstration(Sensor to ESP32 to ThingSpeak)

This working model involves:

- Connecting a potentiometer and an LDR to ESP32 analog pins GPIO36 and GPIO39 respectively.
- Programming the ESP32 to read sensor data and push it to ThingSpeak via HTTP POST requests every 20 seconds.
- Viewing sensor readings on the ThingSpeak web dashboard in real-time graphs.
- Ensuring the entire data pipeline from sensor acquisition through wireless transmission to cloud visualization is operational using just the ESP32 development board, USB Type-C cable, and a computer.

This demonstration consolidates the core IoT concept of integrating edge sensing devices with cloud analytics for remote data access using minimal hardware.