

Projet Homelab IoT avec ESP32-S3

1. Description du projet

Ce projet a pour objectif de créer un homelab IoT capable de détecter la présence physique des appareils à proximité via BLE et de montrer leur état via Wi-Fi. Les données sont publiées sur un broker MQTT local, stockées dans InfluxDB et visualisées avec Grafana.

Le projet est conçu pour être réalisé avec un seul ESP32-S3 DevKitC-1 et peut être enrichi par les smartphones personnels pour simuler plusieurs appareils à surveiller. Il est entièrement local, sans dépendance à des services cloud payants.

Objectifs

- Détecter automatiquement les appareils à proximité (physiquement) avec BLE.
 - Suivre l'état des appareils connectés au réseau via Wi-Fi (heartbeat MQTT).
 - Publier les données sur un broker MQTT sécurisé.
 - Visualiser les métriques sur Grafana.
 - Être valorisable sur un CV pour des compétences DevOps / DevSecOps / Systèmes d'information.
-

2. Matériel utilisé

- **ESP32-S3 DevKitC-1** (Wi-Fi + BLE)
- Câble USB-C pour programmation et debug
- Smartphones personnels (Android/iOS) pour heartbeat MQTT
- PC ou serveur local pour faire tourner Docker et les services suivants :
- Mosquitto MQTT Broker
- InfluxDB 1.8
- Grafana OSS
- Telegraf

Optionnel : - CC1101 + antenne pour projets futurs (RF 433 MHz), non utilisé dans cette version.

3. Logiciels / Stack utilisée

- **ESPHome** : pour flasher l'ESP32 et gérer les capteurs BLE et Wi-Fi
 - **Docker & Docker Compose** : pour exécuter Mosquitto, InfluxDB, Grafana et Telegraf localement
 - **Telegraf** : consommateur MQTT et insertion dans InfluxDB
 - **InfluxDB** : stockage des séries temporelles
 - **Grafana** : dashboards et alertes
-

4. Fichier ESPHome YAML (ESP32-S3)

```
esphome:  
  name: esp32_s3_homelab  
  platform: ESP32  
  board: esp32-s3-devkitc-1  
  
esp32_ble_tracker:  
  
wifi:  
  ssid: "TON_SSID"  
  password: "TON_PASSWORD"  
  
mqtt:  
  broker: "IP_DE_TON_PC"  
  username: "mqttuser"  
  password: "mqttpassword"  
  discovery: false  
  
logger:  
  level: DEBUG  
  
ota:  
  
sensor:  
  - platform: wifi_signal  
    name: "ESP32 WiFi RSSI"  
    update_interval: 60s  
  
binary_sensor:  
  - platform: status  
    name: "ESP32 Status"  
  
  - platform: ble_presence  
    mac_address: "AA:BB:CC:DD:EE:FF"  
    name: "Phone_Sarah_BLE"  
    id: phone_sarah_ble  
    on_press:  
      - mqtt.publish:  
          topic: "presence/phone_sarah/ble"  
          payload: "1"  
    on_release:  
      - mqtt.publish:  
          topic: "presence/phone_sarah/ble"  
          payload: "0"  
  
interval:  
  - interval: 60s  
    then:  
      - lambda: |-
```

```

char buf[256];
int rssi = WiFi.RSSI();
unsigned long up = millis() / 1000;

snprintf(buf, sizeof(buf), "{\"rssi\": %d, \"uptime\": %lu}", rssi, up);
id(mqtt_client).publish("home/esp32/metrics", buf);

```

Notes: - Remplacer les placeholders par vos informations : SSID, mot de passe Wi-Fi, IP du PC, identifiants MQTT, MAC BLE. - BLE scanning détecte les appareils autour ; la fiabilité dépend du type d'appareil (Android vs iOS). - Le heartbeat Wi-Fi garantit un suivi même si le BLE n'est pas détectable.

5. Docker Compose minimal

```

version: '3.8'
services:
  mosquitto:
    image: eclipse-mosquitto:2
    restart: unless-stopped
    volumes:
      - ./mosquitto/config:/mosquitto/config
      - ./mosquitto/data:/mosquitto/data
      - ./mosquitto/log:/mosquitto/log
    ports:
      - "1883:1883"

  influxdb:
    image: influxdb:1.8
    restart: unless-stopped
    environment:
      - INFLUXDB_DB=metrics
    volumes:
      - ./influxdb:/var/lib/influxdb
    ports:
      - "8086:8086"

  grafana:
    image: grafana/grafana-oss
    restart: unless-stopped
    volumes:
      - ./grafana:/var/lib/grafana
    ports:
      - "3000:3000"

  telegraf:
    image: telegraf:latest
    restart: unless-stopped
    volumes:
      - ./telegraf/telegraf.conf:/etc/telegraf/telegraf.conf:ro

```

```
depends_on:  
  - mosquitto  
  - influxdb
```

6. Notes de sécurité

- Activer **authentification MQTT** (disable allow_anonymous) et choisir un mot de passe fort.
 - Ne pas exposer les ports MQTT, Grafana ou InfluxDB sur Internet.
 - Wi-Fi sécurisé (WPA2/WPA3).
 - Sauvegarder les données InfluxDB régulièrement.
-

7. Workflow attendu

1. ESP32-S3 scanne les appareils BLE et publie des messages MQTT.
 2. Les smartphones publient périodiquement leur heartbeat via Wi-Fi + MQTT.
 3. Telegraf consomme les messages MQTT et les stocke dans InfluxDB.
 4. Grafana affiche un dashboard avec la présence et les métriques des appareils.
 5. Alertes possibles si batterie faible ou appareil absent.
-

8. Bonus / Extensions possibles

- Ajouter des capteurs externes (DHT, PIR) pour compléter la supervision.
 - Intégrer Home Assistant pour automatisation locale.
 - Ajouter un module CC1101 pour monitorer d'anciens capteurs RF 433 MHz.
 - Ajouter des alertes Telegram / email via Grafana ou un petit script Python.
-

**Ce document constitue un guide complet pour mettre en place un homelab IoT fonctionnel avec un seul ESP32-S3 et des appareils mobiles pour la détection et le monitoring.