Remote weather station

Michał Nowak • 28.02.2022

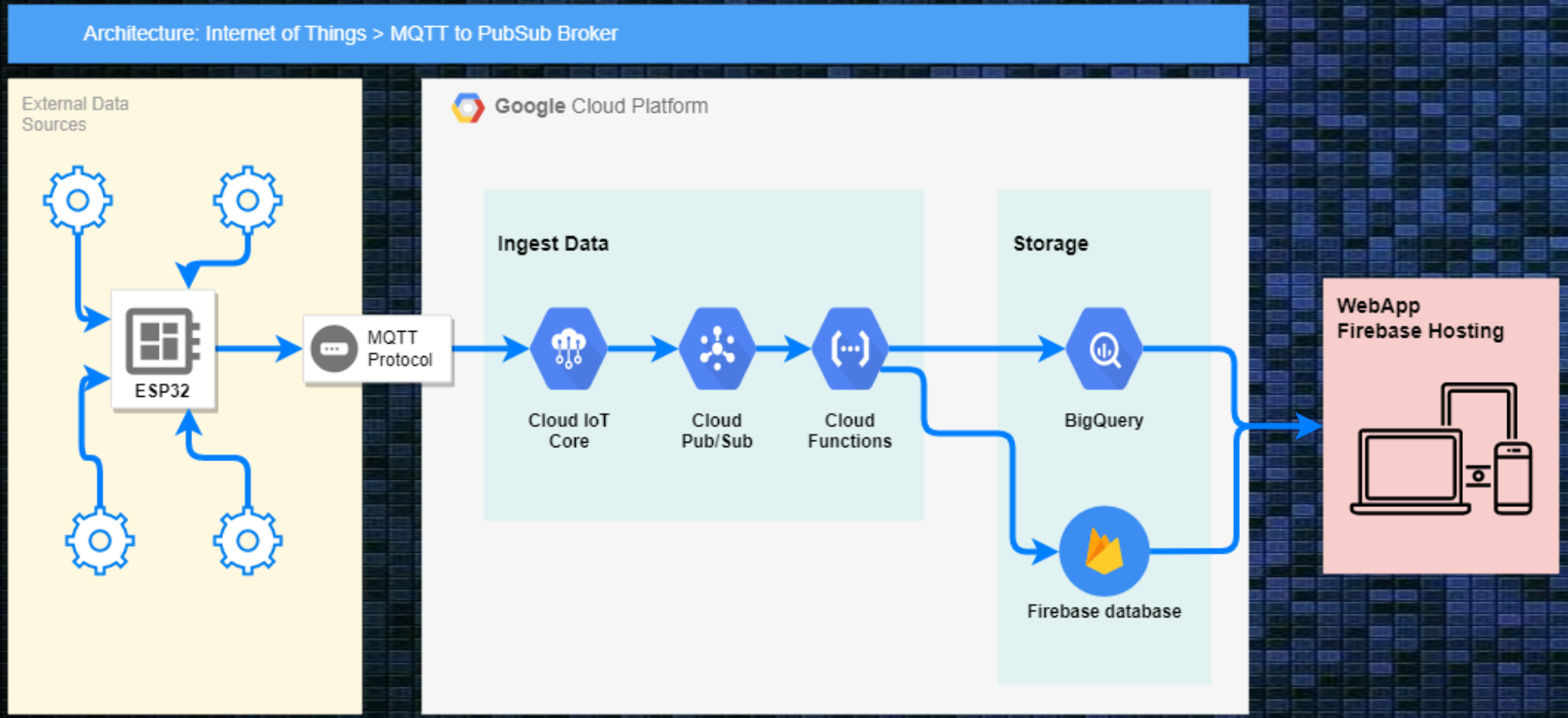
Devices and sensors

- DFRobot: FireBeetle ESP32 IOT Microcontroller(V3.0)
- DFRobot: FireBeetle Covers-Gravity IO Expansion Shield
- Gravity BMP388 Barometric Pressure Sensor
- Gravity: Analog SHT30 Temperature & Humidity Sensor
- Gravity: Analog Ambient Light Sensor
- Gravity: Analog Grayscale Sensor

Tech stack

- Arduino
- C++14
- Python 3.10.2
- Google Cloud IoT Core
- MQTT
- JSON

github.com/nowek7/remote_weather_station


Architecture: Internet of Things > MQTT to PubSub Broker
External Data Sources
ESP32
MQTT Protocol
Google Cloud Platform
Ingest Data
Cloud IoT Core
Cloud Pub/Sub
Cloud Functions
Storage
BigQuery
Firebase database
WebApp
Firebase Hosting

Main loop

github.com/nowek7/remote_weather_station

remote_weather_station.ino

```
void loop()
{
// Start loop codes...

// Connect to google cloud platform.
network::gcp::Config gcpConfig =
{
    .projectId = config["gcp"]["project_id"].as<const char*>(),
    .cloudRegion = config["gcp"]["cloud_region"].as<const char*>(),
    .registryId = config["gcp"]["registry_id"].as<const char*>(),
    .deviceId = config["gcp"]["device_id"].as<const char*>(),
    .mqttBridgeHostname = config["gcp"]["mqtt_bridge_hostname"].as<const char*>(),
    .mqttBridgePort = config["gcp"]["mqtt_bridge_port"].as<int>()
};
network::gcp::IoTClient iotClient(gcpConfig);
iotClient.connect();

// Initializes sensor instances
// ...

// Create payload.
DynamicJsonDocument json(PAYLOAD_SIZE);
json["temperature"] = temperatureSHT30.readValue();
json["humunidity"] = humunidtySHT30.readValue();
json["grayscale"] = grayscale.readValue();
json["lighscale"] = lighscale.readValue();
json["pressure"] = pressureBMP388.readValue();

// Generate the minified JSON and send it to the Serial port.
std::string payload;
const auto writtenBytes = serializeJson(json, payload);
iotClient.publish(payload);

// end loop codes...
}
```

Retrospection

Limitations

- ➔ Analog sensors
- ➔ Lack of unit tests
- ➔ Measurement quality
- ➔ Arduino software

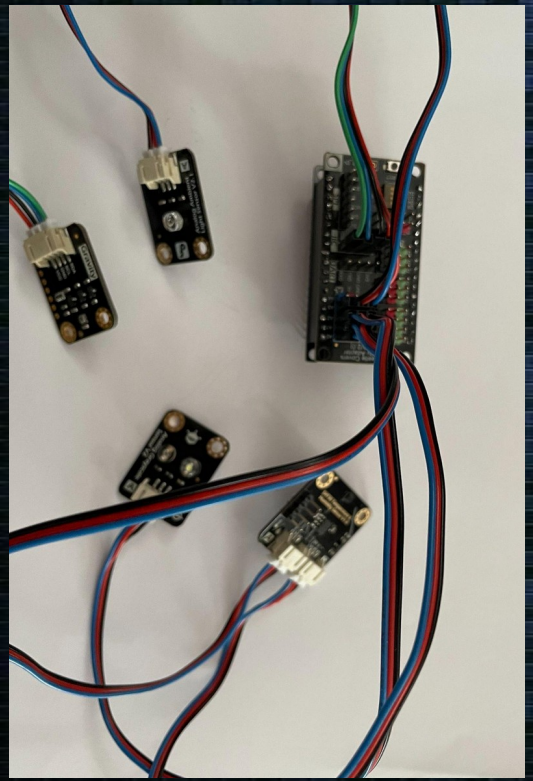
Problems

- ★ The first approach with Micropython
- ★ Flashing binary file into device on Linux
- ★ There was needed modify the Google Cloud IOT Core JWT library on Arduino

Next steps

- ★ RTOS (Zephyr, FreeRTOS)
- ★ Wind sensor
- ★ Air quality sensor
- ★ Visualization tool for iOS / Android

What with the case?



Thank you


Github


LinkedIn