BLE Communication on EPS32

ESP32 - CENTRAL

- Start ESP32 **Central**
- BLE Inicialization
- Is BLE on? — No
- Deep Sleep (3min)
- Yes
- Waiting for BLE message
- Message received? — No
- Yes
- Print JSON message
- WIFI Inicialization
- WiFi establsihed? — Yes / No
- Send JSON via MQTT

ESP32 - PERIPHERAL

- Start ESP32 **Peripheral**
- BLE Inicialization
- Is BLE on? — No
- Deep Sleep (3min)
- Yes
- Make measurment - temperature or random value
- Send BLE message
- Message send? — Yes / No

Message

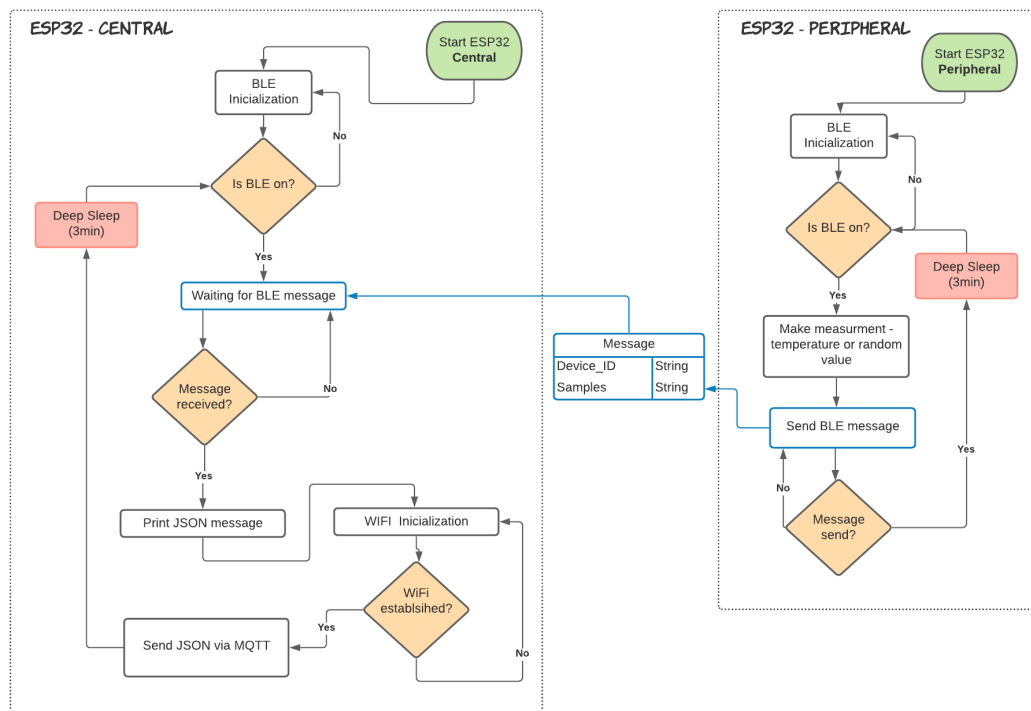| Device_ID	String
Samples	String

Figure 1 Program flowchart diagram

- Operating devices principle: **Peripheral devices based on ESP32 send measured data to cloud via BLE and Central ESP32**

- Communication between devices via Bluetooth Low Energy **(BLE)**

- **Code made in C**, based on ESP-IDF v.4.3. - BLE and WiFi handling code made in separate files .c and .h (preferred: functions run by task in main.c need knowledge in **FreeRTOS**)

- **No magic number** – Define values (like time interval, wifi and mqtt config) in library e.g.: define_list.h

- Communication to cloud: Send received string via **WiFi** based on **MQTT** protocol

- Example of string (message): **{"samples": "-23", "deviceId": "1270FEF2-08D5-49E8-8B2B-42E1B4F22188"}**

  Samples is sensors data read, from temperature sensor etc. DeviceId is for authorisation in cloud

- For connection with MQTT Broker need IP, PORT, LOGIN, PASSWORD and TOPIC, which will be sand when milestone will be created.