Ziele

* Rechtzeitig erkennen, wenn eine Pflanze zu gießen ist
* Schonung der 2032-Batterien in den Sensoren
* Einbindung der Sensoren in OH2

Technologien

* MiFlora-BLE-Library
* Nutzung beider Kerne (Andreas Spiess: <https://www.youtube.com/watch?v=k_D_Qu0cgu8> )
* Arduino-JSON-Library
* ESP32 RTC-RAM und Deepsleep (<https://www.youtube.com/watch?v=r75MrWIVIw4&t=1s> )

Grundablauf

* ESP32 startet alle 5 Minuten und wird in beiden Kernen genutzt (Multithreading, damit die blockierende BLE-Abfrage die Konfiguration per http-Request nicht stört)  
  Geht nach 1 Minute in den DeepSleep-Modus (Zeit für Konfiguration und parallel dazu der Abfrage eines Sensors im RoundRobin-Verfahren)
  + Thread 0 🡺 Configthread
    - Rundenzähler aus dem RTC-RAM auslesen
      * Mangels der Verfügbarkeit der Echtzeit wird der Rundenzähler als Timestamp bei den Messungen verwendet
        + NTP soll nicht Voraussetzung sein
        + Zentrale Zeitverteilung per MQTT kommt später
        + Idee: Eigener NTP-Server im LAN
    - Einlesen der Konfiguration aus dem SPIFFS
      * SSID, Passwort, Thing-Name
      * Array der MiFloraSensors werden in der Config als JSON-String gespeichert
        + Sensornummer
        + Mac-Adresse
        + Name der Pflanze
        + Timestamp der letzten Messung
        + Timestamp der letzten Übertragung
        + [Letzte Messwerte]
    - http-Server wartet auf Konfiguration
    - Wenn im BLE-Thread eine Abfrage erfolgreich absolviert wurde, werden die Config-Daten neu geschrieben
    - Rundenzähler erhöhen
    - Nach Ablauf der Wakeup-Zeit schickt der Thread den ESP wieder in den DeepSleep
  + Thread 1 🡺 BLE-Thread
    - LastReadSensorIndex aus RTC-RAM auslesen und zyklisch erhöhen
      * Wird bei Reset auf 0 gestellt
    - Nächsten MiFlora-Sensor, dessen letzte Messzeit mindestens eine Stunde alt ist per BLE auslesen
      * Daten per MQTT an Broker übermitteln
      * Timestamp auf aktuellen Rundenzähler setzen
      * Flag für Mainthread setzen, dass ein Wert geändert wurde