# Development of a multi sensor logger and a gateway using Bluetooth Low Energy (BLE)

**Team members:**

Dillon M.

Yowhannes. D.

Ngan D.

Tatyana P.

**Advisor:**

Dr. Ismail Uysal

A

# Meet the Team

- In order from left to right: Yowhannes, Dillon, Ngan (Kylie), Tatyana

A

# Problem statement

- **Data Low-Power Operation**: The  sensor loggers and the gateway have to operate efficiently on battery power, optimizing power consumption to extend battery life
- **Multi-Sensor Data Acquisition**: The sensor loggers should be capable of measuring temperature, humidity, and accelerometer data at configurable sampling intervals and logging the data in their memory
- **Gateway Functionality**: The gateway shall synchronize with multiple sensor loggers, configure their settings, retrieve current measurements, download logs, and publish data via WIFI using MQTT
- **User Interface Development:** A web user interface application, namely Grafana connect will subscribe to the gateway to show the data published from the sensor loggers.

K

# Design approach


1

a. Sensor Input: The Bluetooth mesh boards take input from embedded sensors (Temp, humidity, noise, light).

b. Sensor data is stored on board memory until advertisement signal is received.

2

a. Upon receiving the correct serial communication the Bluetooth sensor board will send data packets over secure network to the gateway.

3

a. The ESP32 gateway will advertise data packets over the network to establish the mesh Bluetooth sensor network.

b. During the receiving interval the gateway will receive the data packets from the Bluetooth sensor boards and categorize the data before sending it to the MQTT broker.

4

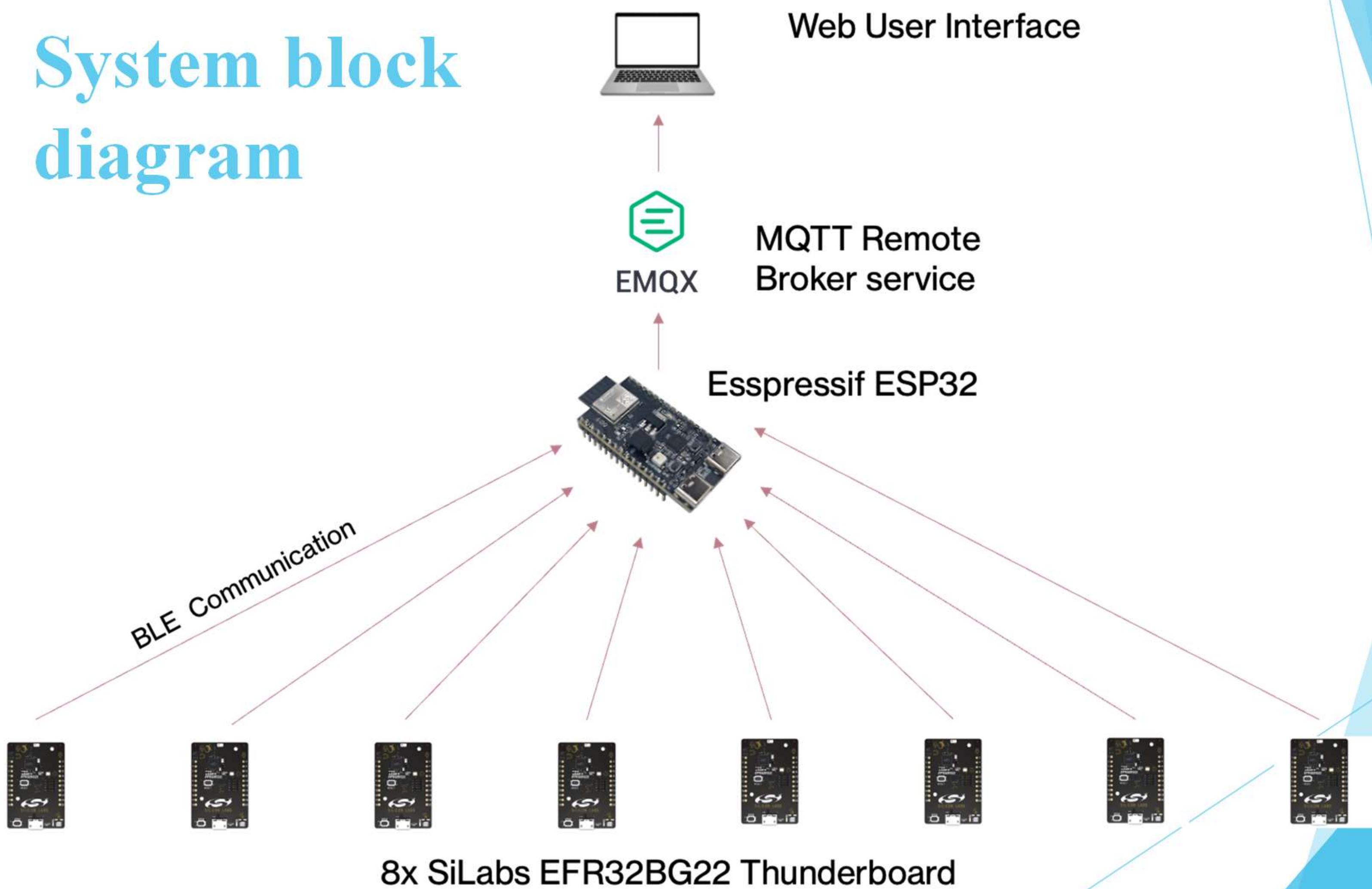
a. The remote MQTT broker service EMQX will facilitate the transfer of the collected and sorted data from the gateway to a web server for storage.

5

a. The web user interface will receive the data transmitted via the MQTT broker service and store it in a formatted and categorized spreadsheet for user access.

T

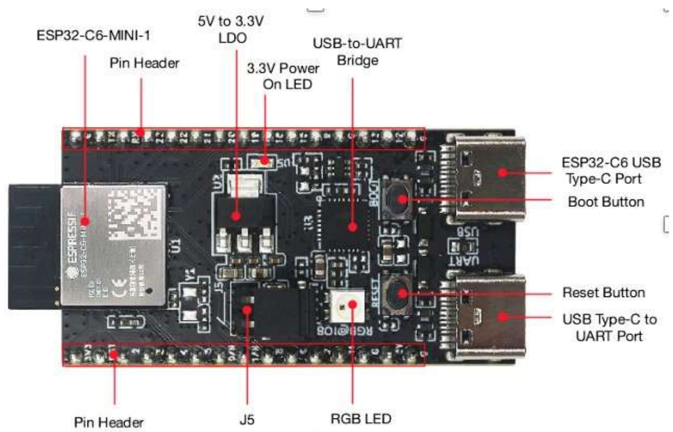
# System block diagram




K

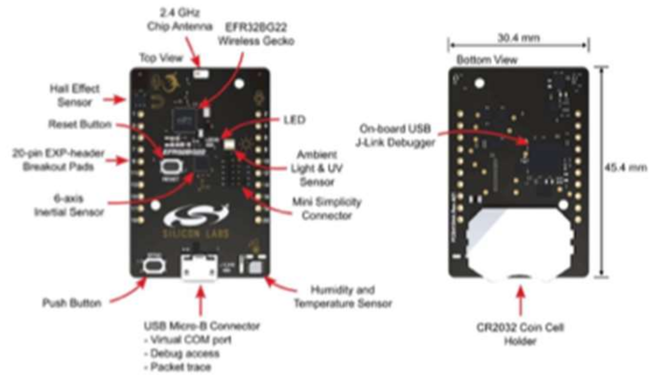
# Design of component(s) and sub-system(s)

## BLE/WI-FI GATEWAY SUB-SYSTEM




ESSPRESSIF ESP32

## MULTI-LOGGER SUB-SYSTEM

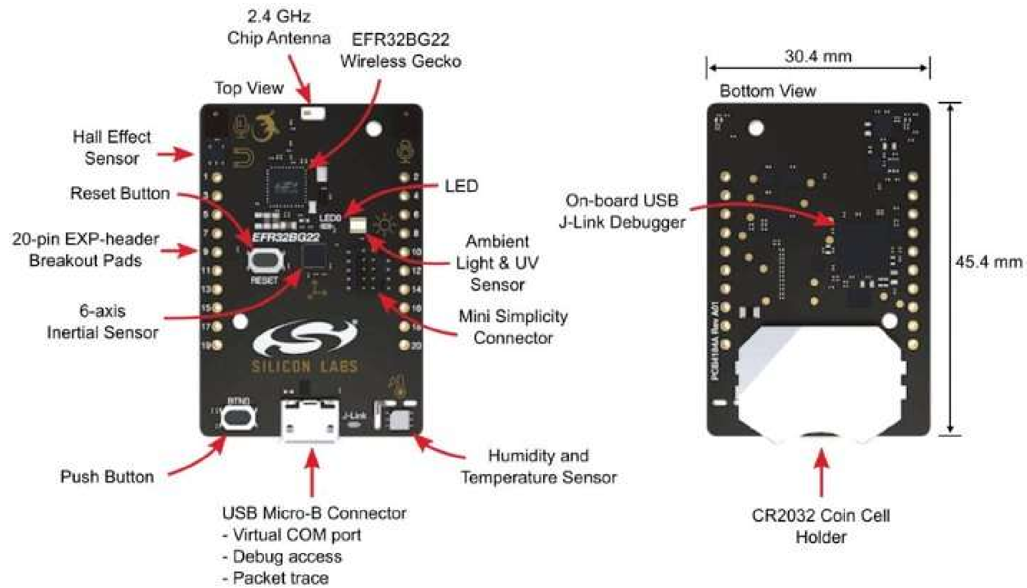



SILABS EFR32BG22 THUNDERBOARD

N

# Multi-sensor logger sub-system

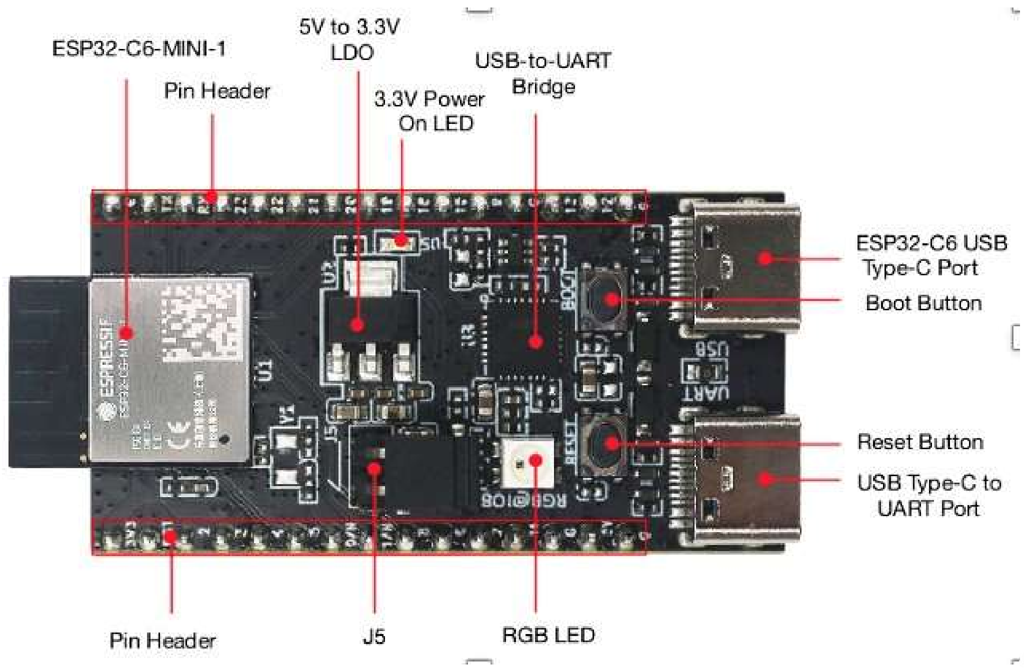
- -Including: temperature sensor, humidity and pressure sensor.
  -BLE 5.2 support for low-power communication
  -Ultra-low power consumption: 1.4 µA in deep sleep, ~3.6 mA when active.
  -Sensor data is logged and stored temporarily in the MCU.
  -Power efficiency: deep sleep modes to extend battery life, optimized BLE connection intervals for minimal power consumption




T

# BLE/Wi-Fi Gateway Sub-system

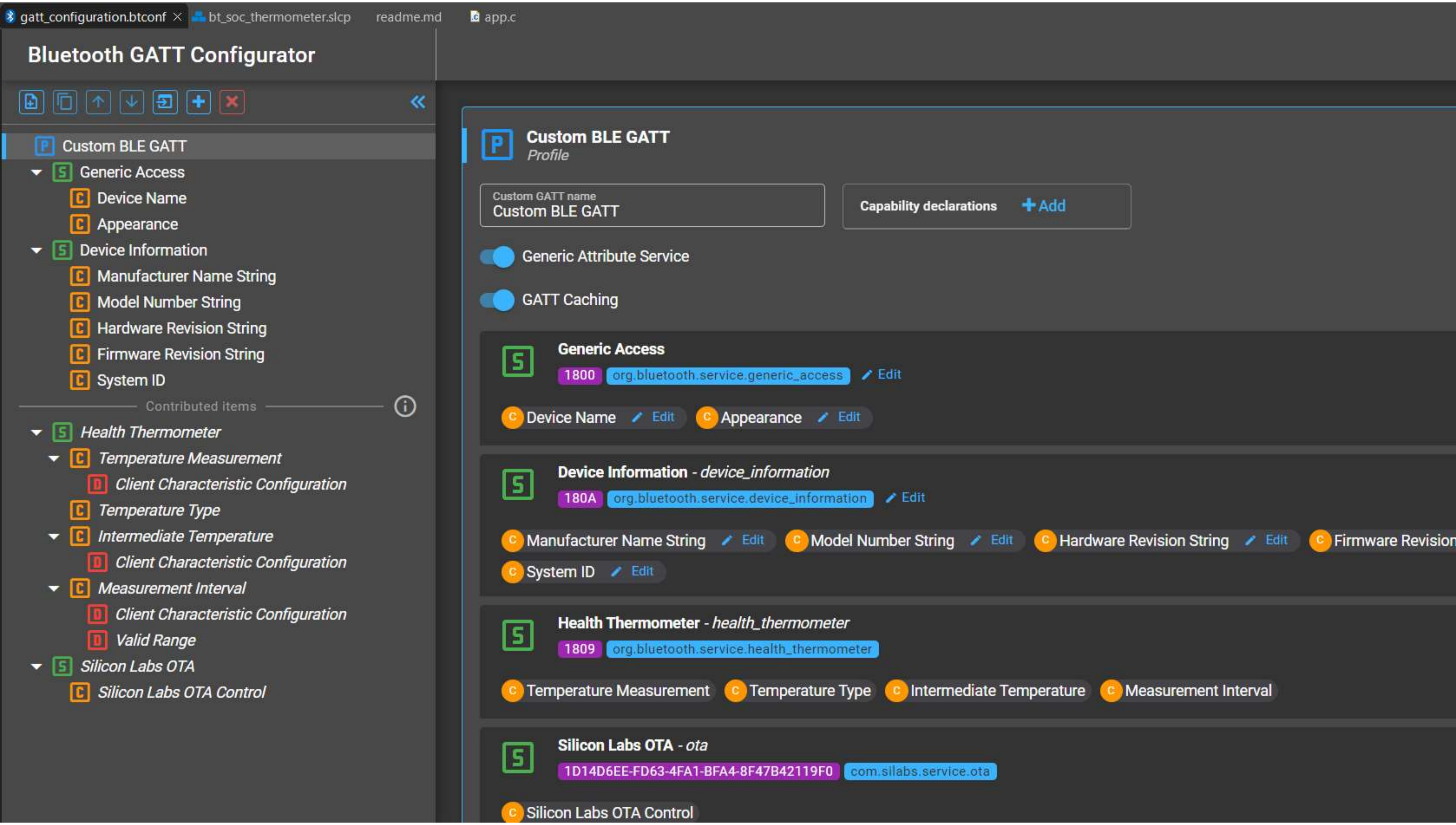
- -Dual-core Xtensa LX6 processor: One core will handle BLE reception, and the other will manage Wi-Fi communication
  -Receive data from at least three sensor loggers using BLE 4.2 or 5.0 protocols.
  -Support simultaneous connections with at least three BLE loggers.
  -Power-efficient BLE stack to minimize active time during data transfers.
  -Low-power modes: ~10 µA in deep sleep, ~160 mA during Wi-Fi transmission.

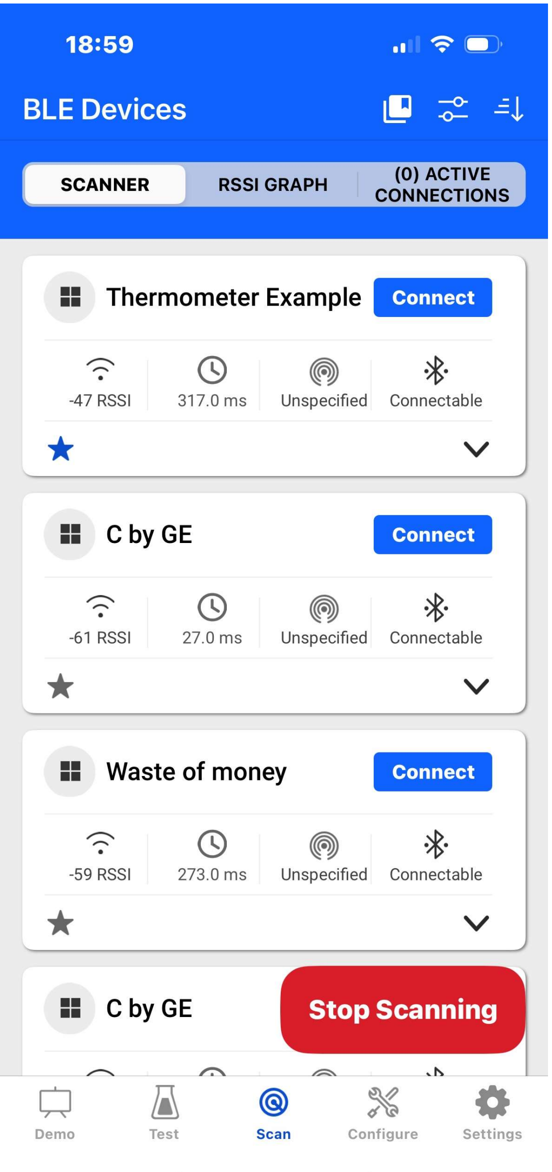


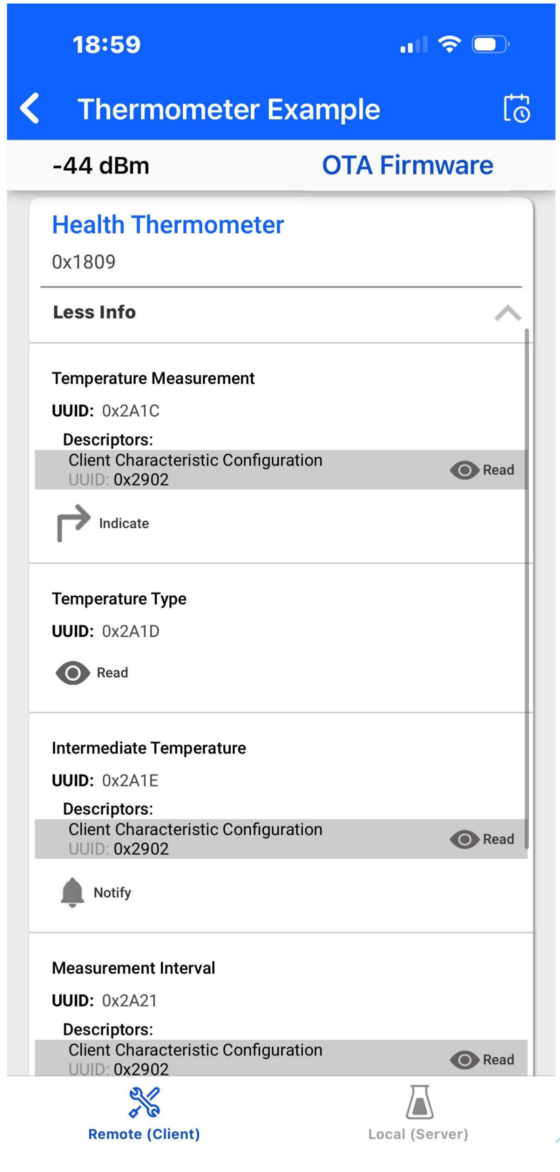

Y

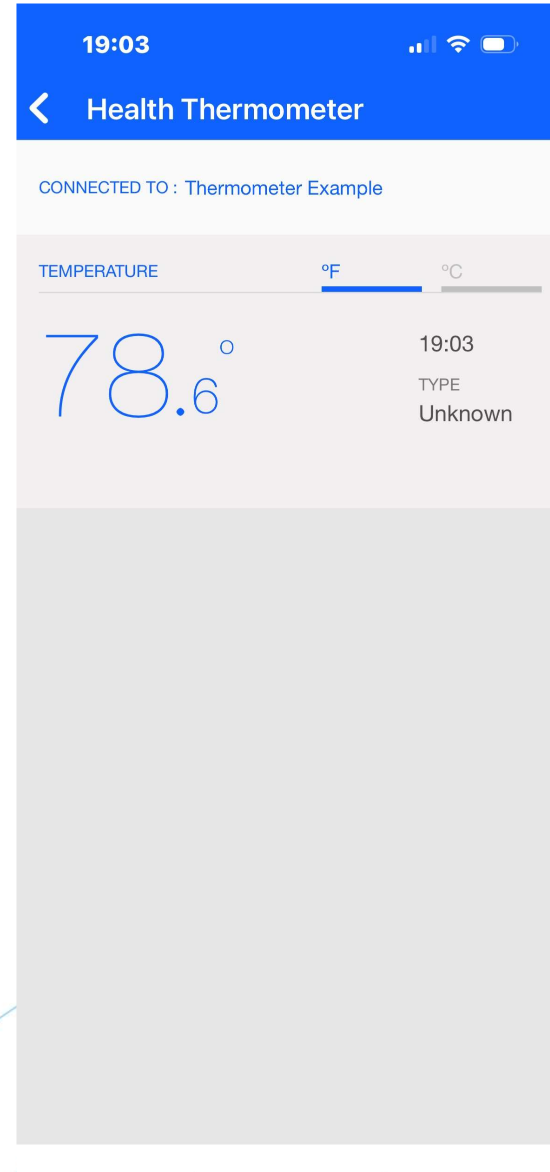
# Hardware/software development to date

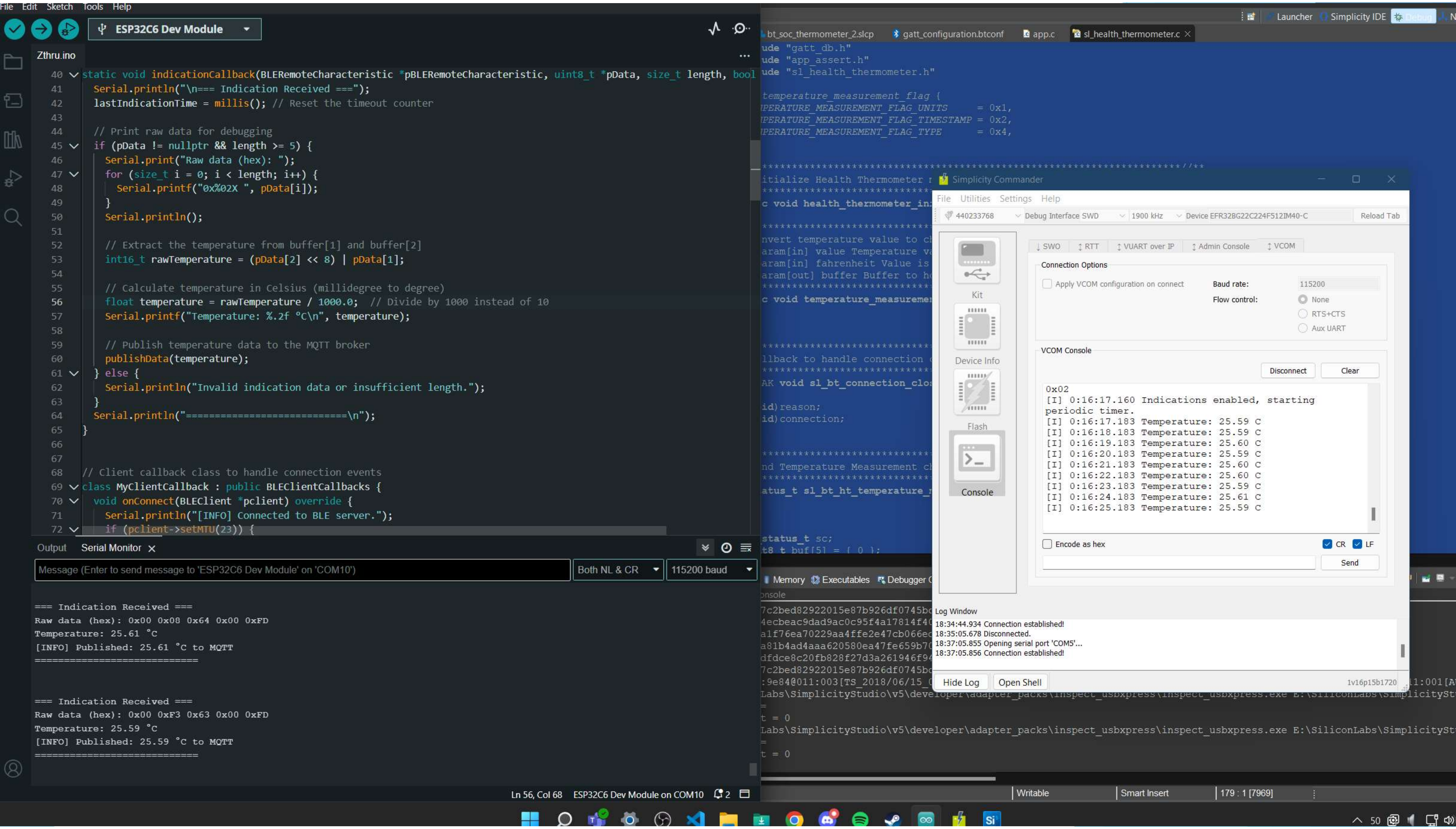
1. Gateway and Bluetooth node set up individually
2. Functions tested individually
3. Integration
4. Expansion of measurements + optimization  (current phase)


gatt_configuration.btconf
bt_soc_thermometer.slcp
readme.md
app.c
Bluetooth GATT Configurator
Custom BLE GATT
Generic Access
Device Name
Appearance
Device Information
Manufacturer Name String
Model Number String
Hardware Revision String
Firmware Revision String
System ID
Contributed items
Health Thermometer
Temperature Measurement
Client Characteristic Configuration
Temperature Type
Intermediate Temperature
Client Characteristic Configuration
Measurement Interval
Client Characteristic Configuration
Valid Range
Silicon Labs OTA
Silicon Labs OTA Control
Custom BLE GATT
Profile
Custom GATT name
Custom BLE GATT
Capability declarations
Add
Generic Attribute Service
GATT Caching
Generic Access
1800
org.bluetooth.service.generic_access
Edit
Device Name
Edit
Appearance
Edit
Device Information - device_information
180A
org.bluetooth.service.device_information
Edit
Manufacturer Name String
Edit
Model Number String
Edit
Hardware Revision String
Edit
Firmware Revision
System ID
Edit
Health Thermometer - health_thermometer
1809
org.bluetooth.service.health_thermometer
Temperature Measurement
Temperature Type
Intermediate Temperature
Measurement Interval
Silicon Labs OTA - ota
1D14D6EE-FD63-4FA1-BFA4-8F47B42119F0
com.silabs.service.ota
Silicon Labs OTA Control


18:59
BLE Devices
SCANNER
RSSI GRAPH
(0) ACTIVE CONNECTIONS
Thermometer Example
Connect
-47 RSSI
317.0 ms
Unspecified
Connectable
C by GE
Connect
-61 RSSI
27.0 ms
Unspecified
Connectable
Waste of money
Connect
-59 RSSI
273.0 ms
Unspecified
Connectable
C by GE
Stop Scanning
Demo
Test
Scan
Configure
Settings


18:59
Thermometer Example
-44 dBm
OTA Firmware
Health Thermometer
0x1809
Less Info
Temperature Measurement
UUID: 0x2A1C
Descriptors:
Client Characteristic Configuration
UUID: 0x2902
Read
Indicate
Temperature Type
UUID: 0x2A1D
Read
Intermediate Temperature
UUID: 0x2A1E
Descriptors:
Client Characteristic Configuration
UUID: 0x2902
Read
Notify
Measurement Interval
UUID: 0x2A21
Descriptors:
Client Characteristic Configuration
UUID: 0x2902
Read
Remote (Client)
Local (Server)


19:03
Health Thermometer
CONNECTED TO : Thermometer Example
TEMPERATURE
°F
°C
78.6 °
19:03
TYPE
Unknown

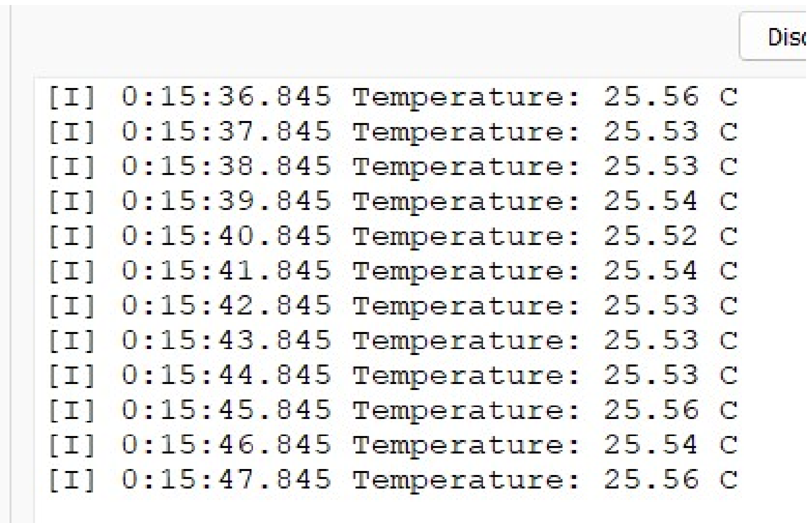


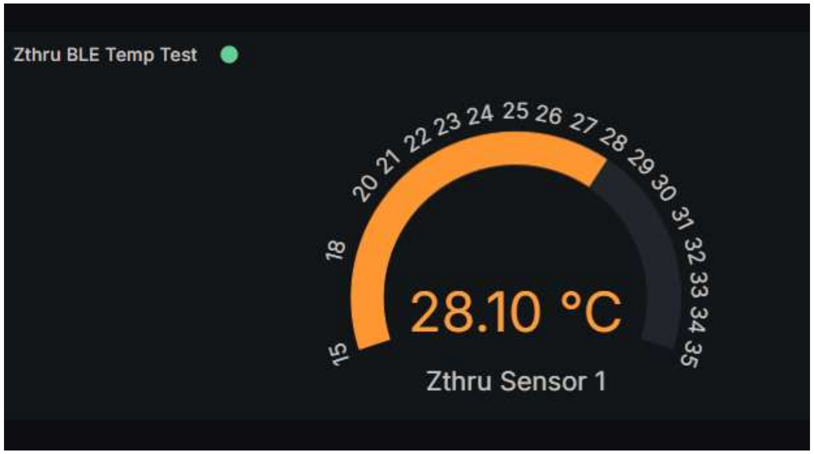
File Edit Sketch Tools Help
ESP32C6 Dev Module
Zthru.ino
static void indicationCallback(BLERemoteCharacteristic *pBLERemoteCharacteristic, uint8_t *pData, size_t length, bool
Serial.println("\n=== Indication Received ===");
lastIndicationTime = millis(); // Reset the timeout counter
// Print raw data for debugging
if (pData != nullptr && length >= 5) {
Serial.print("Raw data (hex): ");
for (size_t i = 0; i < length; i++) {
Serial.printf("0x%02X ", pData[i]);
}
Serial.println();
// Extract the temperature from buffer[1] and buffer[2]
int16_t rawTemperature = (pData[2] << 8) | pData[1];
// Calculate temperature in Celsius (millidegree to degree)
float temperature = rawTemperature / 1000.0; // Divide by 1000 instead of 10
Serial.printf("Temperature: %.2f °C\n", temperature);
// Publish temperature data to the MQTT broker
publishData(temperature);
} else {
Serial.println("Invalid indication data or insufficient length.");
}
Serial.println("===========================\n");
}
// Client callback class to handle connection events
class MyClientCallback : public BLEClientCallbacks {
void onConnect(BLEClient *pclient) override {
Serial.println("[INFO] Connected to BLE server.");
if (pclient->setMTU(23)) {
Output
Serial Monitor
Message (Enter to send message to 'ESP32C6 Dev Module' on 'COM10')
Both NL & CR
115200 baud
=== Indication Received ===
Raw data (hex): 0x00 0x08 0x64 0x00 0xFD
Temperature: 25.61 °C
[INFO] Published: 25.61 °C to MQTT
===========================
=== Indication Received ===
Raw data (hex): 0x00 0xF3 0x63 0x00 0xFD
Temperature: 25.59 °C
[INFO] Published: 25.59 °C to MQTT
===========================
Ln 56, Col 68 ESP32C6 Dev Module on COM10
Launcher Simplicity IDE
bt_soc_thermometer_2.slcp
gatt_configuration.btconf
app.c
sl_health_thermometer.c
Simplicity Commander
File Utilities Settings Help
440233768
Debug Interface SWD
1900 kHz
Device EFR32BG22C224F512IM40-C
Reload Tab
Kit
Device Info
Flash
Console
SWO RTT VUART over IP Admin Console VCOM
Connection Options
Apply VCOM configuration on connect
Baud rate: 115200
Flow control: None RTS+CTS Aux UART
VCOM Console
Disconnect
Clear
0x02
[I] 0:16:17.160 Indications enabled, starting periodic timer.
[I] 0:16:17.183 Temperature: 25.59 C
[I] 0:16:18.183 Temperature: 25.59 C
[I] 0:16:19.183 Temperature: 25.60 C
[I] 0:16:20.183 Temperature: 25.59 C
[I] 0:16:21.183 Temperature: 25.60 C
[I] 0:16:22.183 Temperature: 25.60 C
[I] 0:16:23.183 Temperature: 25.59 C
[I] 0:16:24.183 Temperature: 25.61 C
[I] 0:16:25.183 Temperature: 25.59 C
Encode as hex
CR LF
Send
Log Window
18:34:44.934 Connection established!
18:35:05.678 Disconnected.
18:37:05.855 Opening serial port 'COM5'...
18:37:05.856 Connection established!
Hide Log
Open Shell
Memory Executables Debugger
Writable Smart Insert 179 : 1 [7969]

# Issues

- 1. ESP32 stack size was initially too large
- 2. Data type conflicts between both devices
- 3. Published packet container containing mixed symbols
- 4. EFR requiring special case for indications

```
void publishData(float sensorValue) {
  char msg[50];
  snprintf(msg, sizeof(msg), "Sensor Value: %.2f", sensorValue);
  client.publish(mqtt_topic, msg);
  Serial.print("Published: ");
  Serial.println(msg);
}
```


Disc
[I] 0:15:36.845 Temperature: 25.56 C
[I] 0:15:37.845 Temperature: 25.53 C
[I] 0:15:38.845 Temperature: 25.53 C
[I] 0:15:39.845 Temperature: 25.54 C
[I] 0:15:40.845 Temperature: 25.52 C
[I] 0:15:41.845 Temperature: 25.54 C
[I] 0:15:42.845 Temperature: 25.53 C
[I] 0:15:43.845 Temperature: 25.53 C
[I] 0:15:44.845 Temperature: 25.53 C
[I] 0:15:45.845 Temperature: 25.56 C
[I] 0:15:46.845 Temperature: 25.54 C
[I] 0:15:47.845 Temperature: 25.56 C


Zthru BLE Temp Test
15 18 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
28.10 °C
Zthru Sensor 1

D

# Testing Plan

| | |
|---|---|
| EFR32BG22 to ESP32 Communication | •Ensure sensors collect accurate data (temperature, humidity) and communicate with the gateway via BLE.<br>•Test BLE connectivity between the EFR32BG22 and ESP32.<br>•Verify that sensor data is correctly transmitted via BLE to the gateway. |
| ESP32 to MQTT Communication | •Data published successfully, with delays resolved by optimizing firmware.<br>•ESP32 devices transmitted sensor data to an MQTT broker. |
| Data Visualization on Grafana | •Validated real-time visualization of sensor data on Grafana.<br>•Successfully displayed data, resolved topic misalignment issues. |
| Sensor Accuracy and Performance | •Verify sensor accuracy against reference instruments.<br>•Reading within 1.5% of reference values, planned filtering for unstable data. |
| Power Consumption testing | •Ensure that the sensor logger and ESP32 enters low-power modes when idle and operates efficiently during active periods.<br>•EFR32BG22:~25mA during transmission; ESP32:160mA, accounted for power spikes. |
| System Robustness Under Multiple Conditions | • Assessed performance under varying temperature, humidity, and Wi-Fi conditions.<br>•Stable performance expected, with minor delays in weak Wi-Fi areas. |

TK

# Division of work

| Hardware Development | Software Development | Testing and Validation | Documentation and Reporting |
|---|---|---|---|
| • Ngan D.<br>• Tatyana P. | • Dillon M.<br>• Yowhannes. D. | • Dillon M.<br>• Yowhannes. D.<br>• Ngan D.<br>• Tatyana P | • Dillon M.<br>• Yowhannes. D.<br>• Ngan D.<br>• Tatyana P |

A

# Hardware demo

- https://xdemonfeverx.grafana.net/