Development of a multi sensor logger and a gateway using Bluetooth Low Energy (BLE) with PAwR capability

System Requirements Document (SRD)

Version 1.3

April 22, 2024

Team Member 1 Name: Dillon McKay

Signature:________________________________, date: 04/22/2024

Team Member 2 Name: Yowhannes Dirar

Signature:______ ______________________________ , date: 04/22/2024

Team Member 3 Name: Bao Ngan Dang

Signature:________________________________ , date: 04/22/2024

Team Member 2 Name: Tatyana Pompey

Signature:________________________________ , date: 04/22/2024

Faculty Advisor Name: Dr. Ismail Uysal

Signature:_______ __________________ , date: 04/24/2024, 3:48PM

1. Executive Summary

1.1 Project Overview

The objective of this project is to develop a system that comprises a battery-operated, low power multi-sensor logger and a BLE/Wi-Fi gateway that communicates with at least three sensor loggers and transfer the data using Bluetooth Low Energy (BLE) with PAwR capability in the most power efficient way.

1.2 Purpose and Scope of this System Requirements Document

Purpose:

The document serves as a preliminary guideline for completing a project involving the development of a system with Bluetooth Low Energy (BLE) and Wi-Fi communication capabilities. It outlines the major steps, requirements, and deliverables for the project to ensure clarity and alignment among project members.

In Scope:

The scope of the document encompasses the following key aspects:

- Familiarization with the development environment, including the installation of necessary software tools (Simplicity Studio) and review of user guides.
- Understanding of Bluetooth Low Energy (BLE) concepts, such as advertisements, connections, profiles, and services.
- Development of a network gateway (Bridge) with BLE and Wi-Fi interfaces for data collection and communication with a remote MQTT broker.
- Implementation of configurable devices for temperature and humidity sensing, including logging mechanisms and status services.
- Integration of Periodic Advertising with Response (PAwR) functionality for device communication without establishing a Bluetooth connection.
- Review, testing, and improvement of the integrated system, including code cleanup and bug fixes.

Out of scope:

- Detailed physical design considerations beyond PCB size and weight estimates, including enclosure design and aesthetics.
- Advanced security measures beyond basic encryption and access controls, such as biometric authentication or blockchain integration.
- Compatibility with all possible platforms or hardware configurations beyond specified development environments and hardware kits.
- Extensive regulatory testing, certification processes, or compliance with international standards beyond basic requirements, unless explicitly required.

Audience:

Project Team Members: Dillon McKay, Yowhannes Dirar, Ngan Dang, Tatyana Pompey
Stakeholders: University of South Florida (USF), ZTHRU
Technical Leads: ZTHRU VP of Engineering - Muhammet Yavuz
External Partners: ZTHRU, CTO of ZTRHU Dilek Uysal

1.3 Definitions, Acronyms, and Abbreviations

BLE *– Bluetooth Low Energy*
PAwR *– Periodic Advertisement with Response*
Zthru *– The industry sponsor partnered with USF EE for the capstone project.*
ESL *– Electronic Shelf Label*
IDE *– Integrated Development Environment*
Silabs *– Silicon Labs*
PCB *– Printed Circuit Board*
Thunderboard *– SiLabs BLE Multi-Sensor Development Board*
GATT *– Generic ATTribute Profile*
ATT *– ATTribute Profile*
MQTT*- Messaging Queuing Telemetry Transport*
SDK *– Software Development Kit*
RTC *– Real Time Clock*
EEPROM *– Electrically Erasable Programmable Read-Only Memory*
FCC *– Federal Communications Commission*
UAT *– User Acceptance Training*
SSID *– Service Set Identifier*

1.4 References

See References In appendix.

2. Product/Service Description

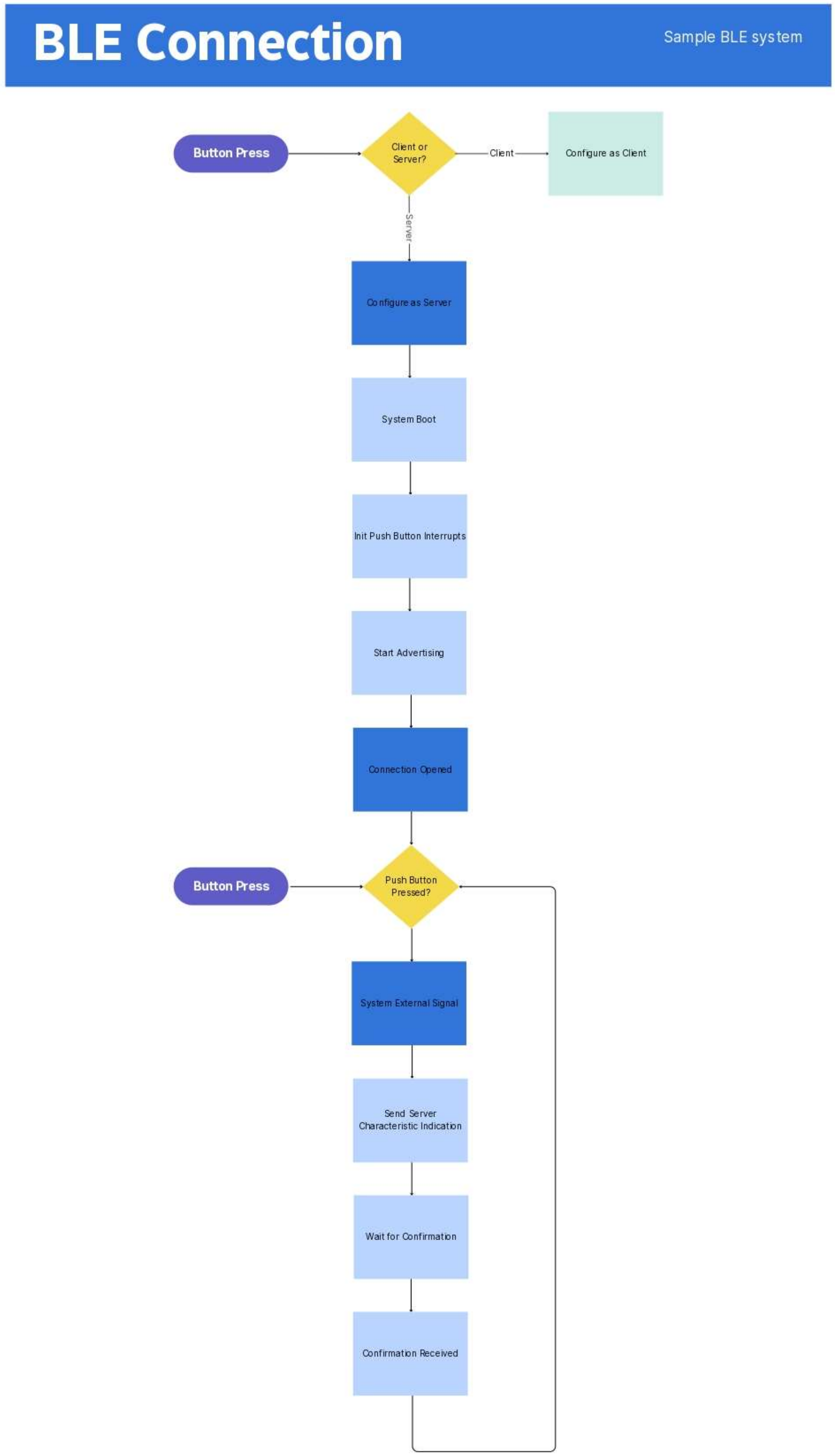
Zthru is a company dedicated to the development of sensor-embedded systems and applications, leveraging cutting-edge technology to redefine connectivity and functionality. Specializing in wireless communication, Zthru predominantly utilizes Bluetooth technology to enable seamless data transmission across its array of sensors.

Zthru is particularly intrigued by PAwR, the latest Bluetooth protocol poised to revolutionize Bluetooth Low Energy (BLE) communication. With its advanced features and capabilities, Zthru believes PAwR holds immense potential, especially for demanding low power sensor solutions.

2.1 Product Context

This product relates to many Bluetooth systems such as ESL's and smart home devices. It is independent and self-contained. The information collected via the sensors will be reported to the gateway and stored for human interpretation. It should interface with any Bluetooth enabled system as long as they are configured with the same profiles.

2.1.1 Block Diagram



2.2 Assumptions

This system makes the following assumptions:

- The system can be developed using C/C++ programming language.
- All components are equipped with a Bluetooth chip capable of BLE operation.
- The developer uses Simplicity Studio IDE.
- The developer has access to the required hardware development: Silab BG22 Dev Kit for sensors and ESP32 DEV Kit for the gateway.

2.3 Constraints

Items that will constrain the design options are as follows:

- Physical size of the PCB.
- Scale of the overall package, the PCB and containing parts.
- Coin slot battery availability for the future.
- Energy cost of normal operation.
- Area in which the devices can broadcast over, distance as well as bandwidth.
- Use of C/C++.
- Use of Simplicity Studio IDE.
    - Prior knowledge of the following:
    - BLE advertisements.
    - BLE connection.
    - Profiles and GATT.
    - Services, discovery, and custom services.
    - Reading/writing characteristic values.
    - Notification and indication.

2.4 Dependencies

Dependencies that affect the requirements:

- This product requires all new sensors to be configured with the correct profile.
- Use of a coin slot battery for power supply.
- 2.4GHz band communication via BLE.
- PAwR Communications.
- Simplicity Studio IDE for development of the profiles.

3. Requirements

Development Environment Setup:

- Simplicity Studio should be installed correctly on all relevant systems.
- Ensure compatibility with various operating systems (e.g., Windows, macOS, Linux).
- Inspect installed Simplicity Studio instances to verify proper installation.

User Guide Review:

- User guides must provide clear instructions for project members.
- Ensure comprehensive coverage of topics relevant to project tasks.
- Review user guides to assess clarity, completeness, and relevance.

Example Project Execution:

- Example projects supplied by SiLabs should run on the development kit without errors.
- Compatibility of example projects with the specific development kits being used.
- Demonstrate running example projects on development kits to ensure functionality.

BLE Concepts Understanding:

- Project members should understand Bluetooth Low Energy (BLE) concepts.
- Provide comprehensive resources for learning BLE basics (advertisements, connections, profiles, GATT, etc.).
- Demonstrate understanding of BLE concepts through simulations or practical exercises.

Network Gateway Development:

- Develop a network gateway (Bridge) with BLE and Wi-Fi interfaces.
- Implement BLE and Wi-Fi communication protocols for data transfer.
- Verify functionality of the bridge by ensuring successful data transmission over both BLE and Wi-Fi interfaces.

Wi-Fi Access Point Mode:

- The bridge should have a Wi-Fi Access Point mode for initial setup.
- Implement a web application hosted by the bridge for configuration.
- Test Wi-Fi Access Point mode to ensure users can configure Wi-Fi settings successfully.

MQTT Communication:

- The bridge must connect to a remote MQTT broker.
- Implement MQTT communication protocol for data exchange with the broker.
- Verify successful connection and data transmission to the MQTT broker.

Device Configuration:

- Implement a configurable device for temperature & RH sensing.
- Develop a device with configurable parameters such as sampling interval, TX power, etc.
- Test configuration settings to ensure they can be set and saved correctly.

Logging Mechanism:

- Devices must log measurements into EEPROM.
- Implement logging mechanism with options for normal mode or circular buffer mode.
- Test logging functionality to ensure accurate recording and storage of measurements.

PAwR Implementation:

- Implement Periodic Advertising with Response (PAwR) for device communication.
- Develop PAwR functionality for configuration, measurement retrieval, and log downloading.
- Test PAwR operations to verify successful device communication without establishing a Bluetooth connection.

3.1 Following areas should be considered for Functional Requirements

3.1.1 User Interface Requirements

3.1.1.1 Configuration Web Application

- The web application shall allow users to configure WI-FI settings (SSID, password, encryption type) and MQTT settings (broker address, port).
- The web application shall provide an intuitive interface for users to interact with and update settings.

3.1.1.2 Mobile/Desktop Application

- Develop a mobile/desktop application for configuring loggers and the gateway.
- The application shall provide a user-friendly interface for configuring device settings.
- The application shall display collected data in an understandable format.
- The application shall support both Android and IOS platforms

3.1.2 Performance

3.1.2.1 Logger sampling interval

- The logger shall support configurable sampling interval for temperature, acceleration, and humidity measurements.
- The system shall ensure that measurements are taken at the specified intervals accurately.

3.1.2.2 Advertisement Frequency

- The logger shall support configurable advertisement frequency.
- The system shall ensure advertisements are broadcast on the specified frequency channels 37, 38 and 39.

3.1.2.3 Maximum Advertisement Duration:

- The logger shall support configurable advertisement duration.
- The system shall ensure that advertisements are broadcast for the specified duration.

3.1.3 Capacity

- The gateway shall support simultaneous connections with multiple sensor loggers.
- The system shall support communication with at least three sensor loggers concurrently.

3.1.4 Availability

- The system shall operate continuously with minimal downtime.
- Unscheduled maintenance downtime shall be minimized, and restoration procedures shall be promptly initiated.
- The system shall detect and handle errors gracefully, providing appropriate feedback to users.
- The system should maintain a packet delivery rate of at least 99.5%, ensuring dependable communication between sensors and gateway devices.

3.1.5 Latency

- The system shall respond to service requests within a maximum latency of 500 milliseconds.
- The average service request latency shall not exceed 200 milliseconds.

3.1.6 Manageability/Maintainability

3.1.7 Monitoring

- The system should provide alerts or notifications when health parameters exceed predefined thresholds, ensuring proactive maintenance and system stability.
- The system should initiate appropriate actions, such as fault isolation, error recovery, and system restart, to mitigate the impact of failure conditions and maintain operational continuity.
- Error should include relevant information such as error codes, timestamps and potential root causes, enabling efficient error diagnosis and resolution.

3.1.8 Maintenance

- System components should be modularized, allowing for individual maintenance, testing, and replacement without impacting the overall system functionality.
- System interfaces should be user-friendly, with clear navigation, descriptive labels, and contextual help options to assist maintenance personnel in performing tasks effectively.
- Adhere to established coding standards, modular design principles, and best practices to simplify maintenance efforts and enhance system reliability.

3.1.9 Systems Interfaces

- Defining message format and content for BLE communication between sensor loggers and the gateway.
- Specifying protocols for BLE communication, including PAwR implementation details.
- Defining the message format and content for the Wi-Fi communication between the gateway and external systems.
- Specifying protocols for Wi-Fi communication, including MQTT messaging structures.
- Data packets are transmitted wirelessly using BLE advertising connection-oriented communication.

3.1.9.1 System and hardware Interface/Integration

- Define interfaces between the system components (BG22 Dev Kit, ESP32 Dev Kit) and associated hardware peripherals (sensors).
- Specify electrical interfaces and communication protocols for seamless integration.
- It follows standard USB communication protocols for data transfer.
- Firmware and code updates are transmitted from simplicity studio program to the ESP32 development kit.

3.2 System Requirements Matrix

Table 1 Requirements

Requirement #	Function	Requirement	Comments	Date Reviewed	Faculty Approval
1	Familiarization with Development Environment	Installation of Simplicity Studio	Ensure Simplicity Studio is installed correctly on all relevant systems.		
2	Familiarization with Development Environment	Review User Guides	Review user guides to assess clarity and completeness of instructions.		
3	Familiarization with Development Environment	Running Examples on Hardware	Demonstrate running example projects on development kits to ensure functionality.		
4	Development Environment Setup	Ensuring Development Environment Readiness	Verify that the development environment is properly set up and functional.		
5	BLE Basics	Learning BLE Advertisements	Demonstrate understanding of BLE advertisements and how they function.		
6	BLE Basics	Learning BLE Connection	Demonstrate understanding of BLE connection establishment.		
7	BLE Basics	Understanding Profiles and GATT	Demonstrate understanding of BLE profiles and GATT.		
8	BLE Basics	Understanding Services, Characteristics, and Notifications	Demonstrate understanding of BLE services, characteristics, and notification mechanisms.		
9	Network Internals and ESP SDK	Testing Wi-Fi Access Point Mode	Inspect and demonstrate functionality of Wi-Fi access point mode for initial configuration.		
10	Network Internals and ESP SDK	Testing Wi-Fi Connection	Demonstrate successful connection to specified Wi-Fi network.		
11	Network Internals and ESP SDK	Testing MQTT Communication	Demonstrate successful communication with remote MQTT broker.		
12	Implementation without PAwR	Configuring Device Parameters	Configure device parameters and verify successful application of settings.		
13	Implementation without PAwR	Testing Logging Mechanism	Test logging of measurements into EEPROM to ensure accuracy and functionality.		

14	Implementation without PAwR	Verifying Status Service Functionality	Verify functionality of custom status service to retrieve device attributes.		
15	PAwR Implementation	Testing PAwR Functionality	Test PAwR operations for device configuration, measurement retrieval, and log downloading.		
16	PAwR Implementation	Firmware Upgrade Testing	Test firmware upgrade functionality using PAwR (optional).		
17	System Integration and Testing	Comprehensive System Integration Testing	Test the integrated system to ensure all components work together as expected.		
18	System Integration and Testing	Code Cleanup and Bug Fixes	Test code cleanup, improvements, and bug fixes to ensure they do not introduce new errors.		

4. User Scenarios/Use Cases

BLE devices can be used for various business applications, including asset tracking, environmental condition monitoring such as temperature, humidity, or pressure. It can also be used to locate assets in the event of theft or loss due to it’s remarkable range and reliability.

It’s uses include but are not limited to:

- Guided tours
- Geofencing
- Firefighter tracking
- Grocery store layout optimization
- Hotel room door unlocking
- Asset protection
- Space utilization

The ZTHRU BLE system is divided as follows:

Network Internals and ESP SDK:

- A network gateway (Bridge) with BLE and Wi-Fi interfaces.
- Wi-Fi access point mode for initial setup.
- Establish Wi-Fi connection and connect to a remote MQTT broker.
- MQTT communication protocol and Espressif SDK.

Implementation without PAwR:

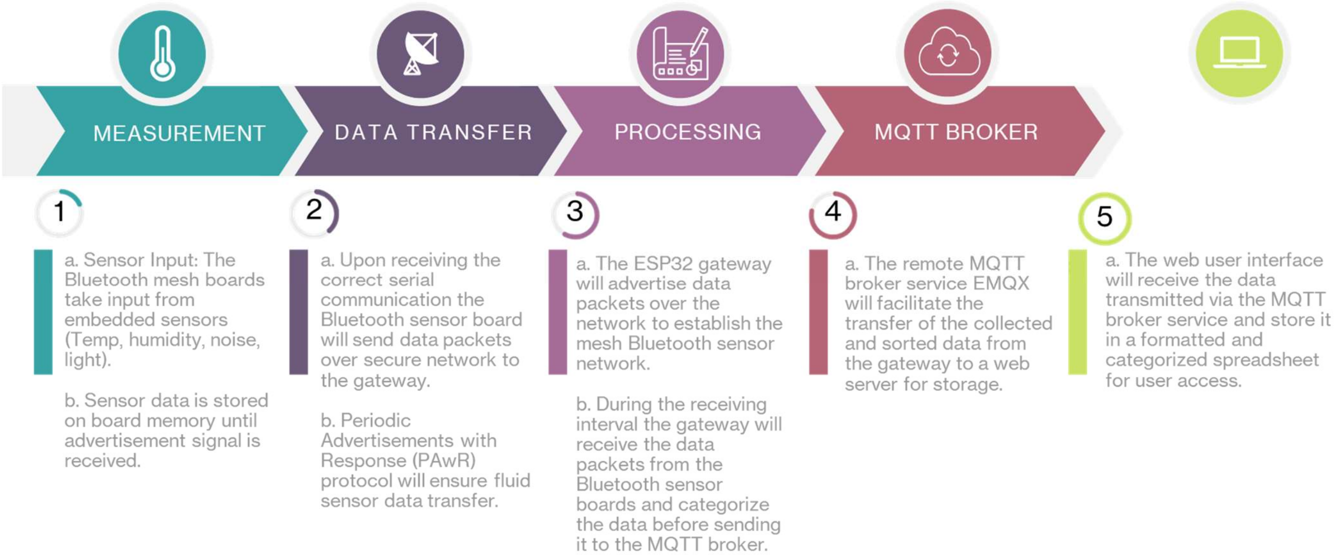
- A configurable device for temperature & RH sensing.
- A custom configuration service with various parameters like time, temperature unit, sampling interval, etc.
- Support for commands such as start logging, stop logging, and download logs.
- Logging of measurements into EEPROM, with options for normal mode or circular buffer mode.
- A custom status service to retrieve device attributes.

PAwR Implementation:

- Implementation of Periodic Advertising with Response on sensor devices and bridges.
- Configure devices, request current measurements, and download logs without establishing a Bluetooth connection.
- Firmware upgrades using PAwR.

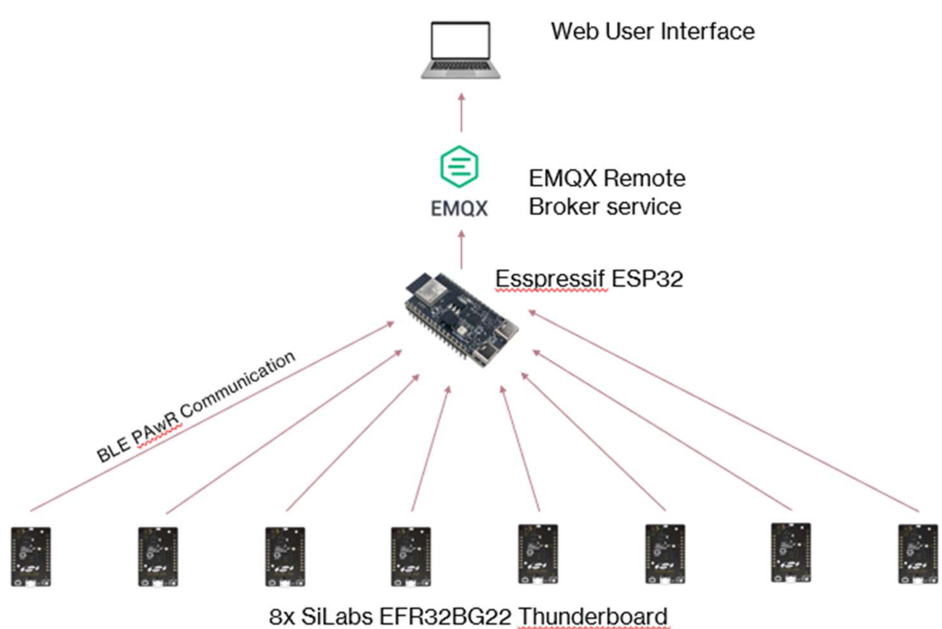
5. Analysis Models

5.1 4.1 Sequence Diagrams



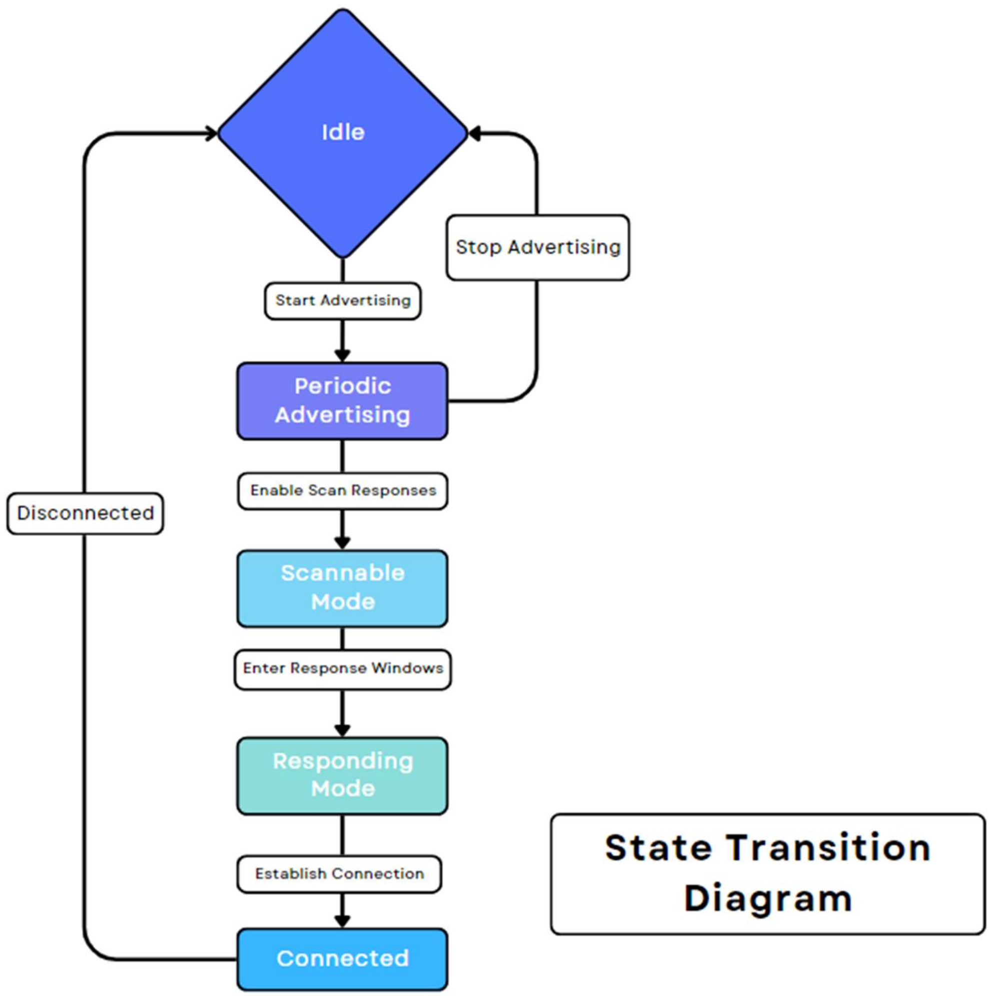
This diagram is useful because it provides a clear and structured view of the data flow and interactions between different components of the PAwR system. The diagram begins with a measurement, typically from a sensor, that captures raw data. This data is then transferred from the sensor to a processing system/gateway, where it is cleaned, analyzed, or prepared for the broker. Following processing, the data is sent to an MQTT broker, a lightweight messaging protocol often used in IoT applications. The MQTT broker acts as a central hub, allowing various subscribers to access the processed data. Finally, the data is displayed in a web user interface, where users can interact with, visualize, or monitor the information.

5.2 4.3 Data Flow Diagrams



The sensor logger (EFR32) is designed to measure temperature, humidity, and accelerometer data with a configurable sampling interval, logging this information in its memory. It serves as a crucial component within the system, interfacing with the gateway (ESP32) to collect sensor data and transmit it to a remote MQTT broker for further processing and display in the web user interface.

5.3 4.2 State-Transition Diagrams



This state diagram is useful for understanding the interactions and transitions in a Bluetooth PAwR system. The diagram begins with the Idle state, indicating that the system is not actively transmitting or receiving. When the broadcaster initiates periodic advertising, it transitions to the Periodic Advertising state, sending out advertising packets at regular intervals. From Periodic Advertising, the system can move into Scannable Mode, where the broadcaster is ready to respond to scan requests from receivers. Following this, the broadcaster may enter Responding Mode, indicating that it can respond to requests from receivers within a response window. When a connection is established, the system transitions to the Connected state, allowing data exchange. The transitions can also return to Idle when advertising stops or a connection is terminated. The Listening Mode represents when a receiver is scanning for periodic advertising, transitioning to Responding Mode upon detecting an advertising packet.

5.4 SWaP

Please provide Size, weight, and power estimates, and show your methodology of how you estimated them.

5.4.1 Size

No size requirements have been given, but the general size of the sensor network will be scaled based on the application. The sensor boards themselves are the size of a debit/credit card.

Weight

There are no weight requirements, however, a battery-powered sensor board's total mass is less than 10 grams.

5.4.2 Power

For the system to qualify as Bluetooth low energy the total current rating per sensor board is < 5mA, and the power range is 0.01-0.50 W.

5.5 System Performance

Please provide performance estimates, i.e., how fast, latency, response time etc. Show your methodology of how you estimated it

Industry standards for BLE technology are:

- ✓ Range up to 100m(<330ft).
- ✓ Data transfer up to 2 Mbit/s.
- ✓ Throughput range of 0.27-1.37 Mbit/s.
- ✓ Minimum 3ms data transfer time.

6. Requirements Test Matrix

Inspection

- Visually inspect all development kits to verify that there are no physical defects.
- Confirm that the development environment is set up properly and all development kits are functioning as expected.
- Verify that user guides provide clear instructions for project members to follow.

Demonstration

- Run example projects on development kits to ensure they execute without errors.
- Test BLE advertisements to ensure they are broadcast correctly and can be detected by other devices.
- Test reading/writing characteristic values to ensure data can be transmitted/received accurately.
- Test Wi-Fi access point mode to ensure the bridge can be configured successfully via a web application.
- Verify that commands such as start logging, stop logging, and download logs are executed properly.

Test

- Verify BLE connections by establishing connections between devices and ensuring data exchange is successful.
- Test profiles and GATT to ensure proper implementation and compatibility with other devices.
- Verify services, discovery, and custom services to ensure they are correctly implemented and can be discovered by other devices.
- Confirm notification and indication mechanisms are functioning as expected.
- Verify Wi-Fi connection by ensuring the bridge connects to the specified wireless network.
- Test MQTT communication to ensure data can be transmitted to the remote broker reliably.
- Verify compatibility and functionality with Espressif SDK.
- Test the configuration service to ensure parameters can be set and saved correctly.
- Test logging of measurements into EEPROM to ensure data integrity and retrieval.
- Confirm functionality of the custom status service to retrieve device attributes accurately.
- Test PAwR implementation to ensure devices can be configured, current measurements can be requested, and logs can be downloaded without establishing a Bluetooth connection.
- Verify data transmission reliability and integrity during PAwR operations.
- Optionally, test firmware upgrade functionality using PAwR.

Analysis

- Perform comprehensive system integration testing to verify the interaction between all components.
- Conduct thorough testing of all functionalities to identify and fix any bugs or issues.
- Test code cleanup and improvements to ensure they do not introduce new errors or regressions.
- Verify that improvements made are effective in enhancing system performance and stability.

List all the requirements stated in Table 1 here and identify with supporting description the verification methods, as described above, for example:

Table 2 Test Requirements Matrix

Requirement #	Function	Requirement	Test Method	Brief Test Description	Faculty Approval
1	Familiarization with Development Environment	Installation of Simplicity Studio	Inspection	Verify proper installation of Simplicity Studio on all relevant systems.	JH
2	Familiarization with Development Environment	Review User Guides	Inspection	Review user guides to assess clarity and completeness of instructions.	JH
3	Familiarization with Development Environment	Running Examples on Hardware	Demonstration	Demonstrate running example projects on development kits to ensure functionality.	JH
4	Development Environment Setup	Ensuring Development Environment Readiness	Inspection	Verify that the development environment is properly set up and functional.	JH
5	BLE Basics	Learning BLE Advertisements	Demonstration	Demonstrate understanding of BLE advertisements and how they function.	JH
6	BLE Basics	Learning BLE Connection	Demonstration	Demonstrate understanding of BLE connection establishment.	JH
7	BLE Basics	Understanding Profiles and GATT	Demonstration	Demonstrate understanding of BLE profiles and GATT.	JH
8	BLE Basics	Understanding Services, Characteristics, and Notifications	Demonstration	Demonstrate understanding of BLE services, characteristics, and notification mechanisms.	JH
9	Network Internals and ESP SDK	Testing Wi-Fi Access Point Mode	Inspection, Demonstration	Inspect and demonstrate functionality of Wi-Fi access point mode for initial configuration.	JH
10	Network Internals and ESP SDK	Testing Wi-Fi Connection	Demonstration	Demonstrate successful connection to specified Wi-Fi network.	JH
11	Network Internals and ESP SDK	Testing MQTT Communication	Demonstration	Demonstrate successful communication with remote MQTT broker.	JH
12	Implementation without PAwR	Configuring Device Parameters	Test	Configure device parameters and verify successful application of settings.	JH
13	Implementation without PAwR	Testing Logging Mechanism	Test	Test logging of measurements into EEPROM to ensure accuracy and functionality.	JH

14	Implementation without PAwR	Verifying Status Service Functionality	Test	Verify functionality of custom status service to retrieve device attributes.	
15	PAwR Implementation	Testing PAwR Functionality	Test	Test PAwR operations for device configuration, measurement retrieval, and log downloading.	
16	PAwR Implementation	Firmware Upgrade Testing	Test	Test firmware upgrade functionality using PAwR (optional).	
17	System Integration and Testing	Comprehensive System Integration Testing	Test	Test the integrated system to ensure all components work together as expected.	
18	System Integration and Testing	Code Cleanup and Bug Fixes	Test	Test code cleanup, improvements, and bug fixes to ensure they do not introduce new errors.	

7. Project Risk

Technical Complexity: The project involves working with Bluetooth Low Energy (BLE) technology, network communication protocols (Wi-Fi, MQTT), and embedded systems development. Technical challenges such as compatibility issues, protocol implementation errors, or hardware limitations could arise, leading to project delays or performance issues.

Resource Availability: Dependencies on third-party tools (e.g., Simplicity Studio, example projects) and documentation (user guides, BLE specifications) may pose risks if these resources are not readily available, up-to-date, or compatible with project requirements. Delays in obtaining necessary resources could impact project timelines.

Hardware Limitations: Development kits and hardware components may have limitations in terms of processing power, memory, or connectivity options. Designing a system that meets performance requirements while operating within hardware constraints is critical. Inadequate hardware resources could lead to system instability or functionality limitations.

Security Vulnerabilities: Implementing network communication features such as Wi-Fi and MQTT introduces potential security risks. Failure to implement proper encryption, authentication, and access controls could result in data breaches or unauthorized access to the system, compromising user privacy and system integrity.

Testing Challenges: Verifying the functionality, reliability, and interoperability of the system components may be challenging, especially when dealing with complex interactions between hardware and software modules. Inadequate testing procedures or insufficient test coverage could lead to undetected defects or performance issues in the deployed system.

Regulatory Compliance: Depending on the intended use and deployment environment of the system, regulatory compliance requirements (e.g., FCC regulations for wireless communication devices, data protection laws) may need to be addressed. Non-compliance with relevant regulations could result in legal liabilities or restrictions on product deployment.

Dependency Risks: The project relies on dependencies such as third-party libraries, SDKs, or communication protocols. Changes or updates to these dependencies, including discontinuation of support or compatibility issues, could impact project continuity or require significant rework.

8. Standards

The BLE PAwR will use WSN and LAN to transfer data and to communicate across systems.

- IEEE WSN Standards:
  https://en.wikipedia.org/wiki/IEEE_802.15

- IEEE Standard for Telecommunications and Information Exchange Between Systems:
  https://standards.ieee.org/standard/802_15_1-2002.html

9. Engineering Ethical Responsibility

Bluetooth Low Energy (BLE) Periodic Advertising with Response (PAwR) is a feature that enables efficient and scalable communication between a broadcaster and multiple receivers in a Bluetooth environment. The team, Zthru, that designed this system considered industry standards, especially those established by the Bluetooth Special Interest Group (Bluetooth SIG), which oversees the development of Bluetooth technology. Adhering to these standards ensures compatibility among Bluetooth devices, reduction with difficulties in integration and allowing future adaptation. This design approach is similar to practices found in other devices, such as Ethernet or USB, where maintaining the IEEE (Institute of Electrical and Electronics Engineers) standards is crucial for successful deployment. By following industry guidelines, the PAwR team is testing a product that aligns with established Bluetooth protocols and maintains an elevated level of security and reliability.

BLE PAwR has a complex socioeconomic impact on a range of stakeholders and sectors. In sectors like healthcare, transportation, and smart home automation, this technology can encourage increased connectivity and integration, which could improve operational efficiency and quality of life. However, given the dangers of data interception and unauthorized access in wireless environments, public safety considerations are crucial. The design includes security features like authentication and encryption to reduce these risks and guarantee the security of user data. When considering societal, cultural, environmental, or global effects, ethical dilemmas may occur. For example, higher connectivity may result in higher energy use and electronic waste, which could be harmful to the environment. A comprehensive strategy is needed to resolve these ethical dilemmas, such as encouraging the use of energy-efficient designs and the proper disposal of electronic components. Stakeholders affected by these conflicts include consumers, manufacturers, regulators, and environmental organizations, all of whom play a role in resolving these issues and promoting sustainable practices.

10. Change Management Process

The process in place for managing the changes for the project and this living SRD is as follows:

- The team will coordinate and communicate via Microsoft teams.
- The team will have biweekly meetings with the sponsor ZTHRU to discuss progress and changes of scope or requirements as needed.
- The use of a living shared document via Microsoft word will facilitate the entire team to simultaneously update the document.
- All team members can submit changes and the integrated change tracking included with office will be used.

APPENDIX

References:

1. https://www.silabs.com/documents/public/data-sheets/efr32bg22-datasheet.pdf
2. https://en.wikipedia.org/wiki/Bluetooth_Low_Energy
3. https://www.youtube.com/watch?v=itybLuMc8gw
4. https://docs.silabs.com/bluetooth/latest/bluetooth-start/
5. https://www.silabs.com/wireless/technology
6. https://docs.silabs.com/bluetooth/latest/bluetooth-fundamentals-overview/
7. https://www.silabs.com/documents/public/user-guides/ug103-14-fundamentals-ble.pdf
8. https://docs.silabs.com/bluetooth/7.0.1/bluetooth-fundamentals-advertising-scanning/
9. https://docs.silabs.com/bluetooth/7.0.1/bluetooth-fundamentals-connections
10. https://docs.silabs.com/bluetooth/7.0.1/bluetooth-gatt/
11. https://docs.silabs.com/bluetooth/7.0.1/bluetooth-fundamentals-system-performance
12. https://docs.espressif.com/projects/espressif-esp-dev-kits/en/latest/esp32c6/esp32-c6-devkitm-1/user_guide.html
13. https://data.energizer.com/pdfs/cr2032.pdf
14. https://eu.mouser.com/datasheet/2/368/efr32bg22_datasheet-1830163.pdf
15. https://eu.mouser.com/new/silicon-labs/silicon-labs-efr32bg22/#:~:text=These%20devices%20deliver%20industry%2Dleading,for%20up%20to%20ten%20years.