ECEN 5823

Internet of Things Embedded Firmware

Course Project Proposal

"Plant Observatory System"

Submitted By –

Team 17

Jithendra H.S.

(jihs6098@colorado.edu)

Suhas Srinivasa Reddy

(susr2271@colorado.edu)

Date: 10th April 2025

OPTION CHOSEN

2

PROJECT OVERVIEW

Author Suhas

Design and develop a prototype system to monitor environmental conditions in a plant nursery setting, such as a greenhouse. The system will:

- Continuously gather key environmental data, including:
  - Humidity
  - Temperature
  - Pressure
  - Ambient Light Intensity
- Timestamp and log the collected data onto an SD card for offline analysis and historical tracking.
- Implement Bluetooth communication between nodes:
  - A Low Power Node will collect sensor data while optimizing energy consumption.
  - A Central Node will receive the data and handle SD card logging.
  - A mobile device can access logged data from the SD card via Central Node.

HIGH LEVEL REQUIREMENTS

Author Suhas

- Environmental Data Collection
  - The LPN shall measure humidity, temperature, pressure, and ambient light intensity every 5 seconds.
  - The Central Node shall timestamp and store the received data on an SD card for offline access and analysis.
- Low Power Operation
  - The LPN shall wake up every 5 seconds, measure sensor values, transmit over BLE, and return to EM3 stop mode.
  - The LPN shall utilize:
    - EM0 during sensor measurement and BLE data transmission.
    - EM1 to allow I2C and ADC operation while CPU is idle.
    - EM2 while waiting for BLE stack events.
    - EM3 between measurement cycles.
- Bluetooth Communication
  - The LPN shall transmit data to the Central Node using BLE.
  - The Central Node shall receive, acknowledge, and store the data.
  - Bluetooth bonding and pairing shall be implemented to ensure secure communication.
- Mobile Device Access
  - A mobile device shall connect via BLE to the Central Node.
  - Upon request, the Central Node shall transmit saved data from the SD card to the mobile device.
- Two-Node Architecture
  - The system consist of only two functional nodes:
    - A Low Power Node for sensing and data transmission
    - A Central Node for data logging and mobile communication

HIGH LEVEL DESIGN DESCRIPTION

Author Suhas

The Plant Observatory System is a compact, energy-efficient monitoring solution built around a two-node Bluetooth architecture. It consists of a two-node Bluetooth architecture: a Low Power Node for sensing and a Central Node for logging and mobile data access.

System Components

Low Power Node (LPN)

Functionality:

- Periodically wakes every 5 seconds to:
    - Measure environmental data (Humidity, Temperature, Pressure, Light).
    - Send data to the Central Node via BLE.
    - Return to low-power state.

Energy Mode Behavior:

- EM3 (Stop Mode): Most of the time between measurements.
- Wake-Up via a timer interrupt (e.g., RTC).
- EM1: For I2C and ADC reads (sensors).
- EM0: For active BLE communication during transmission.
- EM2: Used when waiting on BLE events.

Hardware:

- BLE-capable microcontroller (e.g., EFR32 Blue Gecko).
- Environmental sensors.
- RTC timer or low-frequency clock for periodic wake-up.

Central Node

Functionality:

- Receives BLE data from the LPN.
- Timestamps and logs the data to an SD card.
- Serves logged data to a mobile device over BLE upon request.

Energy Mode Behavior:

- EM2: Idle mode, waiting on BLE events.
- EM0: During data writing to SD card and while handling BLE requests from mobile devices.

Hardware:

- BLE-capable microcontroller with SD card interface.
- Internal clock for timestamping.
- Optional display.

Mobile Device

Functionality:

- Connects to the Central Node via BLE.
- Requests historical data from the SD card.
- Displays received environmental data.

Security:

- BLE bonding and pairing are required for access.

DATA TYPES SYSTEM

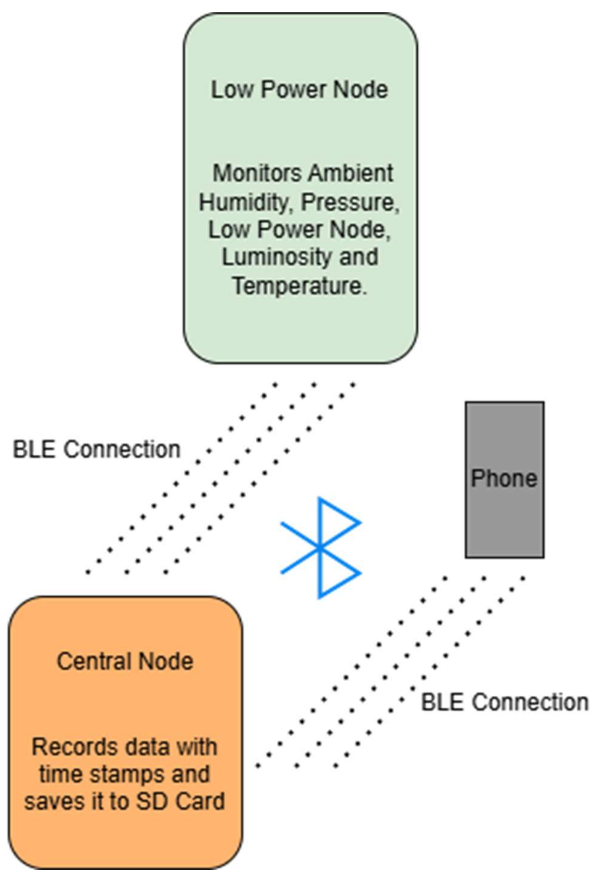
Author Jithendra

The table below shows all types of data exchanged between the client and the server, along with their value ranges, number of bytes per reading, and units.

Measurement	Value range	Length(bytes)	Units	Description
Pressure	300-1100	**2**	**hPa**	Barometric pressure measured using BME280
Humidity	0-100	**2**	**%RH**	Relative humidity measured using Si7021
Luminosity	0-1000	**2**	**Lux**	Ambient light intensity using TEMT6000.
Temperature	-40-125	**2**	**C**	Ambient temperature using Si7021

FUNCTIONAL DIAGRAM

Author Suhas



LCD DISPLAY DESIGN

Author Jithendra

Low Power Node:

- Displays:
  - Server Bluetooth Address (Central Node).
  - Client Bluetooth Address (LPN itself).
  - Connection Status: Advertising → Connected → Bonded.
  - Status Message: "Measuring..." during data collection.

Central Node:

- Displays:
  - Server Bluetooth Address (Central Node).
  - Client Bluetooth Address (LPN or Mobile).
  - LPN Status: "Sleeping" (EM3) or "Active".
  - Connection Status: Discovering → Connected → Bonded.
  - Display recent environmental data received from LPN.
  - Message: "Reading from SD..." during mobile request.

ENERGY MODES AND NODE FUNCTIONALITIES

Author Suhas

Low Power Node (LPN)

Role: Periodically collects sensor data and transmits it over BLE.

Operation Cycle (every 5 seconds):

- EM3 (Stop Mode)
  - Default state; lowest power mode.
  - Device sleeps until RTC triggers the next measurement cycle.
- Wake-up → EM1
  - Used to initialize and read data via I2C and ADC from sensors.
- Switch to EM0
  - BLE stack becomes active to transmit collected sensor data to the Central Node.
- BLE event complete → EM2
  - Waits for BLE stack to finish processing or connection event.
- Return to EM3
  - Cycle ends. Node goes back to sleep.

Summary:

- EM0: BLE communication (TX).
- EM1: I2C/ADC sensor reading.

- EM2: Awaiting BLE stack event.
- EM3: Deep sleep between transmissions.

Central Node

Role: Logs received sensor data to SD card and serves mobile device requests.

Behavior:

- Default: EM2
    - Waits in low-power mode for BLE event (e.g., connection from LPN or mobile device).
- On BLE connection → EM0
    - Active to:
        - Receive data from LPN and log to SD card.
        - Send stored data over BLE to a mobile device on request.
- Return to EM2
    - After completion of BLE and SD tasks.

Summary:

- EM0: BLE RX/TX, SD card operations.
- EM2: Awaiting BLE event.

GATT SERVICES

Author Suhas

- Ambient Humidity and Pressure
- Ambient Light
- Save To SD Card
- Ambient Temperature (Reusing Health Thermometer)

KEY COMPONENTS

Author Jithendra

- BME 280: Humidity and Pressure Sensor
- Si7021: Temperature Sensor
- TEMPT6000: Ambient Light Sensor
- SD Card Interface: To read and write data from SD card
- Blue Gecko EFR32BG13: Main MCU for the system

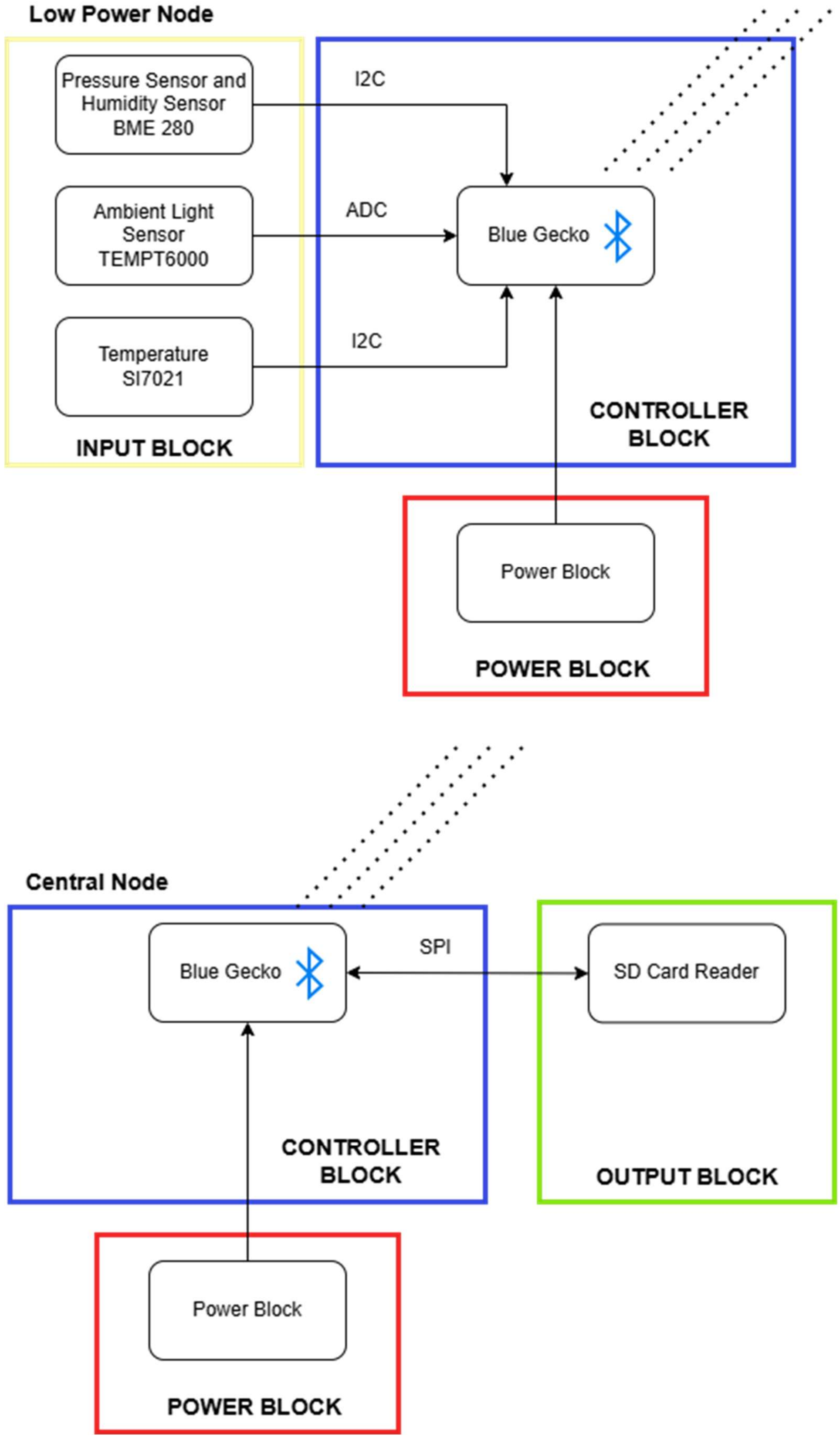
COMMUNICATION PROTOCOLS USED

Author Suhas

- I2C: BME 280 and Si7021 Interface
- ADC: TEMPT6000 Interface
- SPI: SD Card Interface
- BLE: Wireless Communication

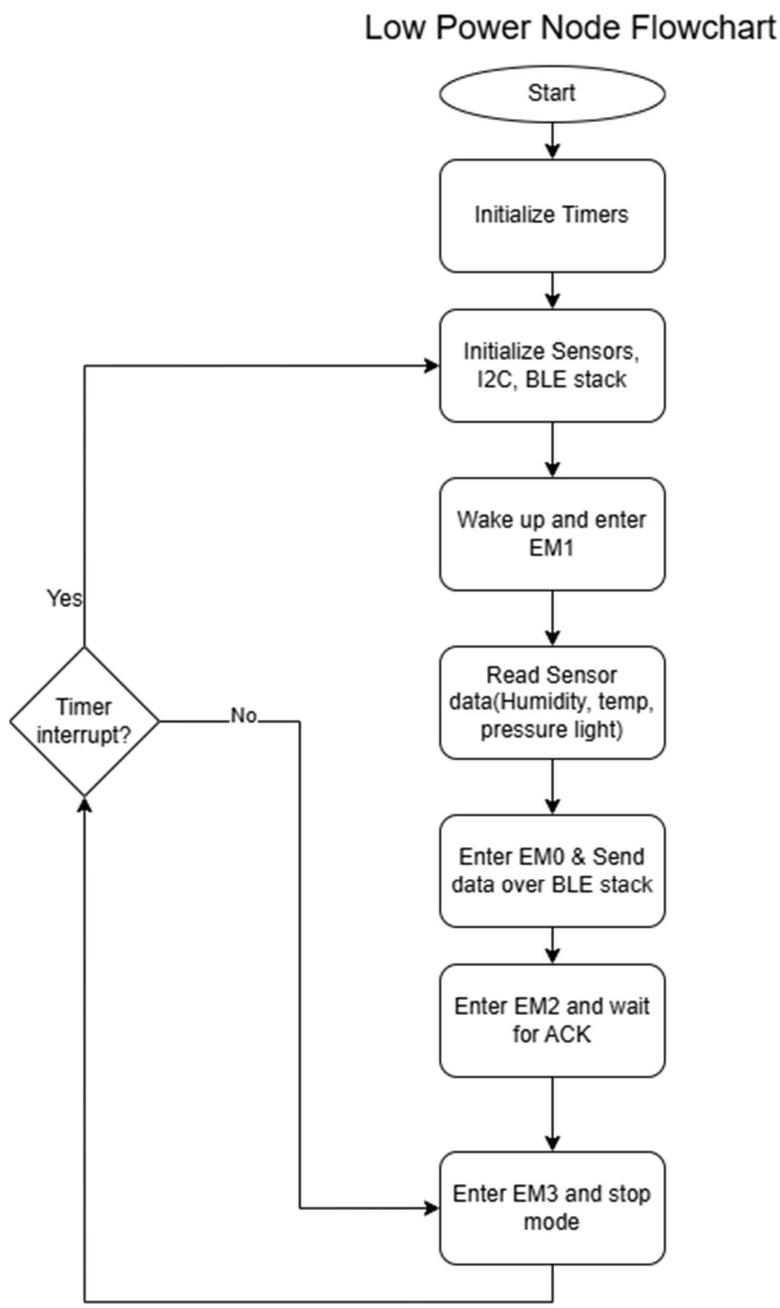
HARDWARE BLOCK DIAGRAM

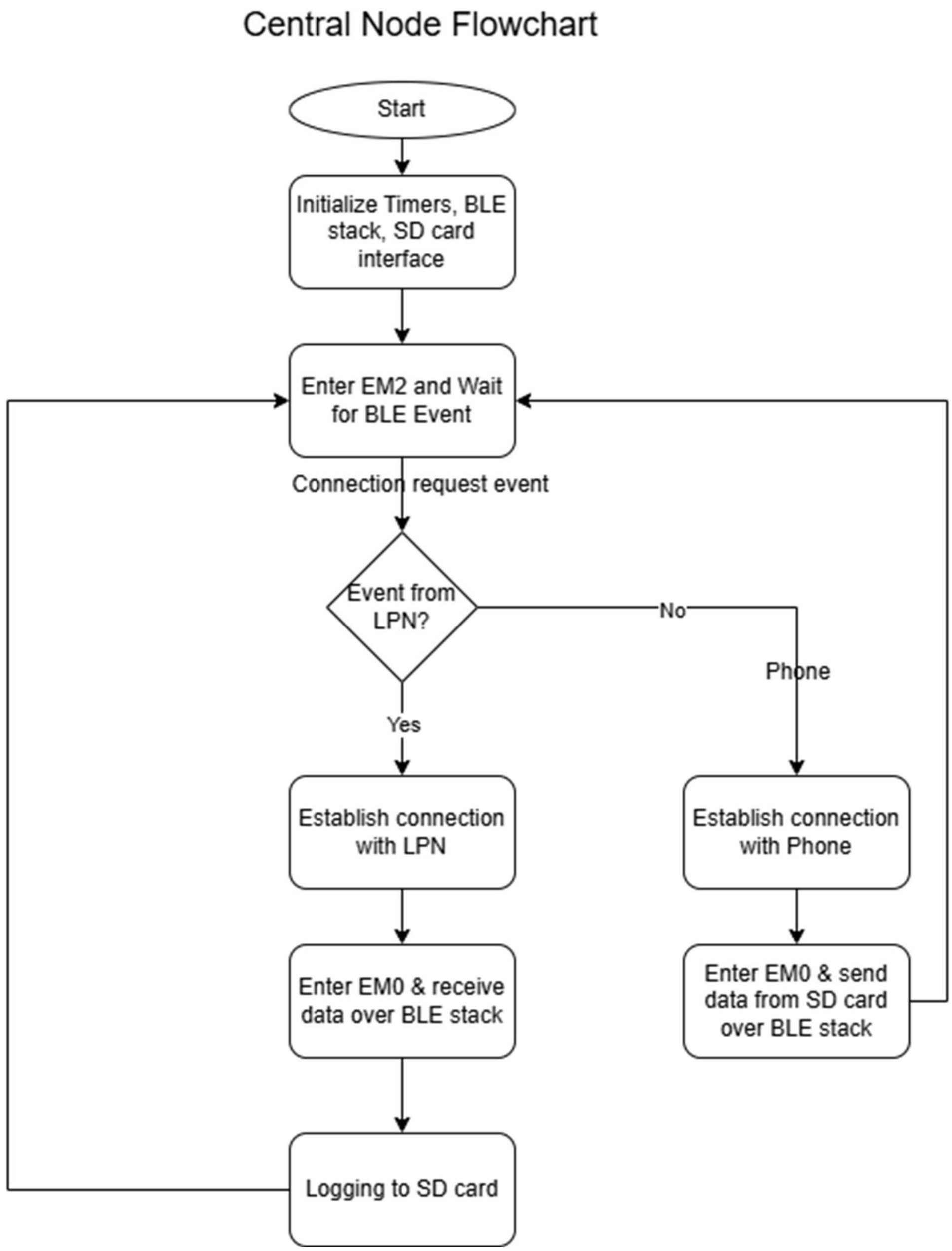
Author Suhas



SOFTWARE BLOCK DIAGRAM - FLOWCHART WITH DATA FLOW

Author Jithendra





DIVISION OF LABOR: HARDWARE VS SOFTWARE

Author Suhas

Hardware components including the Blue Gecko EFR32BG13 microcontrollers, BME280 (Humidity & Pressure Sensor), Si7021 (Temperature Sensor), TEMT6000 (Ambient Light Sensor), and SD card interface are off-the-shelf modules. Hardware design is thus limited to power-efficient interfacing and signal routing between sensors, the microcontroller, and peripherals. This portion contributes to approximately 20% of the overall project effort.

The majority of the development (~80%) is focused on software functionality. This includes:

- Implementing sensor drivers (I2C communication for BME280, Si7021, and TEMT6000).
- Managing power states and energy modes (EM0–EM3) for low-power operation on the Low Power Node.
- Developing the Bluetooth communication stack, including advertising, connections, bonding, and pairing.

- Handling data timestamping, logging to SD card, and BLE service exposure for mobile access on the Central Node.
- Updating LCD displays with connection status and sensor data.
- Ensuring data security through pairing and bonding mechanisms.
- Designing event-driven software architecture to operate efficiently across sleep-wake cycles.

All higher-level tasks such as data collection scheduling, BLE event handling, state transitions, and data parsing for mobile requests are managed in software, making it the core of system functionality.

SUBSYSTEM SUMMARY

Author: Jithendra

In our plant observatory project is composed of several interconnected subsystems, each responsible for a distinct functionality to enable energy-efficient, real-time environmental monitoring. The Sensing Subsystem on the Low Power Node (LPN) includes the BME280, Si7021, and TEMT6000 sensors, which measure humidity, pressure, temperature, and ambient light, respectively. These sensors interface via I2C and ADC protocols and operate within a strict power management framework using energy modes EM0 to EM3 to minimize energy consumption. The Bluetooth Communication Subsystem facilitates wireless data transmission from the LPN to the Central Node, leveraging BLE for secure and efficient communication, including bonding and pairing mechanisms. The Power Management Subsystem ensures the LPN alternates between active sensing and deep sleep modes (EM3), maximizing battery life while maintaining periodic data transmission every 5 seconds. The Data Logging Subsystem resides on the Central Node, which timestamps incoming data and logs it to an SD card via SPI. This node also handles BLE-based requests from mobile devices and serves stored historical data on demand. The Mobile Access Subsystem allows smartphones to securely connect to the Central Node and retrieve logged data, ensuring user-friendly interaction and accessibility. Additionally, the LCD Display Subsystem on both nodes provides real-time visual feedback, such as connection status, sensor readings, and operational states.

TEST PLAN

Author: Jithendra

Our test plan is structured to validate the system at each stage of the project implementation. This incremental testing approach helps identify bugs early and reduces the risk of failure as system complexity increases. We begin by testing fundamental modules such as the timer, sensor data acquisition, and the final state machine. Only after confirming the stable operation of these core components will we integrate the Bluetooth stack.

Bluetooth functionality will be tested for successful connection establishment and proper handling of connection closures. Once verified, we will proceed to the next phase—ensuring that each service operates correctly in response to individual and combined button sequences.

Subsequently, we will validate the logging of sensor data to the SD card and confirm that the logged data can be accurately accessed by client devices. The system's connection status and sensor data will also be verified for correct display on the LCD, following the specified button sequences.

We will then evaluate the board's low-power modes by pushing it into these states and analyzing the resulting energy profile plots. The final step will be a full system test—covering the complete workflow from Bluetooth connection establishment, sensor data acquisition, and data logging, to connection termination and system reset.

Test Plan Table

Test Number	Test Description	Planned to check	Test Result (Pass/Fail)	Notes
1	Test LE timers and interrupts	04-15-2025		
2	Read data and conversion of sensor data	04-18-2025		
3	Verify the FSM transition and deterministic state for each sesnor	04-19-2025		
4	Bluetooth Connection establishment and close functionality	04-23-2025		
5	SD card logging with timestamp	04-24-2025		
6	Sensor services enable and disable of indication and read data from each using the buttons	04-25-2025		
7	LCD test	04-26-2025		
8	Low power mode current measurement	04-27-2025		
9	Final full system testing	04-28-2025		

PROPOSED SCHEDULE

Author: Jithendra

Task	Owner	Expected completion date	Actual completion date
Team formation	-	Apr 1	Apr 1
Project selection	Jithendra, Suhas	Apr 5	Apr 6
Sensor procurement	Jithendra	Apr 8	Apr 10
GATT service Ambient Humidity and pressure	Suhas	Apr 12	
GATT service Ambient light	Jithendra	Apr 12	
Ambient sensor interfacing	Jithendra	Apr 14	
Ambient Humidity and pressure sensor interfacing	Suhas	Apr 14	
Ambient sensor FSM design	Jithendra	Apr 16	
Ambient Humidity and pressure sensor interfacing FSM design	Suhas	Apr 16	
LPN Bluetooth stack event handler implementation	Jithendra	Apr 20	
Client Bluetooth stack event handler implementation	Suhas	Apr 20	
Bluetooth connection establishment and SD integration	Jithendra	Apr 22	
Service handling integration	Suhas	Apr 24	
Unit and regression testing	Jithendra, Suhas	Apr 26	

GITHUB REPOSITORY

Author: Jithendra

https://github.com/CU-ECEN-5823/ecen5823-courseproject-JithendraHS

UPDATE 1 **DATE: 4th April 2025**

STATUS DESCRIPTION

Author: Suhas

- Modified I2C driver to interface BME280 and successfully measured Humidity.
- Created ADC driver to interface TEMPT6000 and measured Ambient light.
- Created GATT Services.

Status: We are on track and up to date with the project.

The updates are detailed below:

Ambient humidity Measurement via BME280:

Author: Suhas

- Referred to the datasheet and interfaced BME280 to measure relative humidity.

```
144000:Info :readHumidityFSM: Humidity Raw Data: MSB 110 LSB 177
144000:Info :readHumidityFSM: Relative Humidity: 27.672852 RH
147000:Info :readHumidityFSM: Humidity Raw Data: MSB 110 LSB 175
147000:Info :readHumidityFSM: Relative Humidity: 27.670898 RH
150000:Info :readHumidityFSM: Humidity Raw Data: MSB 110 LSB 173
150000:Info :readHumidityFSM: Relative Humidity: 27.668945 RH
153000:Info :readHumidityFSM: Humidity Raw Data: MSB 110 LSB 175
153000:Info :readHumidityFSM: Relative Humidity: 27.670898 RH
156000:Info :readHumidityFSM: Humidity Raw Data: MSB 110 LSB 178
156000:Info :readHumidityFSM: Relative Humidity: 27.673828 RH
159000:Info :readHumidityFSM: Humidity Raw Data: MSB 110 LSB 181
159000:Info :readHumidityFSM: Relative Humidity: 27.676758 RH
162000:Info :readHumidityFSM: Humidity Raw Data: MSB 110 LSB 184
162000:Info :readHumidityFSM: Relative Humidity: 27.679688 RH
```

Resource:

https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bme280-ds002.pdf

Ambient Luminosity Measurement via TEMPT6000:

Author: Jithendra

- Interface of Luminosity sensor (TEMT6000):
  Made use of example code given from silicon chip vendors to enable ADC at the EFR32BG13 board and read the light intensity value as given below:

```
0:Info :app_process_action: ADC output : 14
0:Info :app_process_action: ADC output : 14
0:Info :app_process_action: ADC output : 36
0:Info :app_process_action: ADC output : 34
0:Info :app_process_action: ADC output : 1640
0:Info :app_process_action: ADC output : 1896
0:Info :app_process_action: ADC output : 2001
0:Info :app_process_action: ADC output : 20
0:Info :app_process_action: ADC output : 15
0:Info :app_process_action: ADC output : 14
0:Info :app_process_action: ADC output : 15
```

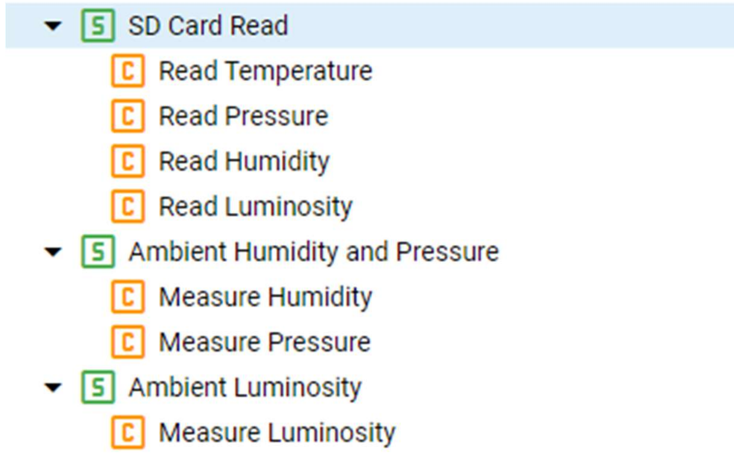
Resource:
https://github.com/SiliconLabs/peripheral_examples/blob/master/series1/adc/adc_single_polled/src/main_s1.c

Created the following GATT Services

Author: Suhas

- Ambient Humidity and Pressure
- Ambient Light
- SD Card Read
- Reusing Health Thermometer

SD Card Read Service (UUID FFF3)			
Characteristics	**Characteristics UUID**	**Characteristics ID**	**Length (Bytes)**
Read Temperature	2B01	read_temperature	17
Read Pressure	2B02	read_pressure	17
Read Humidity	2B03	read_humidity	17
Read Luminosity	2B04	read_luminosity	17
Ambient Luminosity (UUID FFF2)			
Characteristics	**Characteristics UUID**	**Characteristics ID**	**Length (Bytes)**
Measure Luminosity	2D01	measure_luminosity	17
Ambient Humidity and Pressure (UUID FFF1)			
Characteristics	**Characteristics UUID**	**Characteristics ID**	**Length (Bytes)**
Measure Humidity	2C01	measure_humidity	17
Measure Pressure	2C02	measure_pressure	17



PROPOSED SCHEDULE

Author: Jithendra

Task	Owner	Expected completion date	Actual completion date
Team formation	-	Apr 1	Apr 1
Project selection	Jithendra, Suhas	Apr 5	Apr 6
Sensor procurement	Jithendra	Apr 8	Apr 10
GATT service Ambient Humidity and pressure	Suhas	Apr 12	Apr 17
GATT service Ambient light	Jithendra	Apr 12	Apr 17
Ambient light interfacing	Jithendra	Apr 14	Apr 17
Ambient Humidity and pressure sensor interfacing	Suhas	Apr 14	Apr 17
Ambient light FSM design	Jithendra	Apr 16	
Ambient Humidity and pressure sensor interfacing FSM design	Suhas	Apr 16	Apr 17
LPN Bluetooth stack event handler implementation	Jithendra	Apr 20	
Client Bluetooth stack event handler implementation	Suhas	Apr 20	
Bluetooth connection establishment and SD integration	Jithendra	Apr 22	
Service handling integration	Suhas	Apr 24	
Unit and regression testing	Jithendra, Suhas	Apr 26	

TEST PLAN

Author: Jithendra

Updated Test plan:

Test Plan Table

Test Number	Test Description	Planned to check	Test Result (Pass/Fail)	Notes
1	Test LE timers and interrupts	04-15-2025	Pass(100%)	LE timer and I2C interrupt are happening at a regular delay as defined in the FSM
2	Read data and conversion of sensor data	04-18-2025	Pass(100%)	Able to read data from Luminosity, pressure, temperature, and humidity, conversion needs to be taken care of during BLE implementation
3	Verify the FSM transition and deterministic state for each sesnor	04-19-2025	Inprogress(50%)	In development
4	Bluetooth Connection establishment and close functionality	04-23-2025		
5	SD card logging with timestamp	04-24-2025		
6	Sensor services enable and disable of indication and read data from each using the buttons	04-25-2025		
7	LCD test	04-26-2025		
8	Low power mode current measurement	04-27-2025		
9	Final full system testing	04-28-2025		

Display Row information:

Author: Jithendra

Row number	*Information displayed*
1	*Role of the Node*
2	*Node Bluetooth address*
3	*Alias Node Bluetooth address*
4	*Client address*
5	*Connection status*
6	*Passkey*
7	*Action*
8	*Temperature*
9	*Humidity*
10	*Pressure*
11	*Luminosity*
12	*Plant Observatory system*

GITHUB REPOSITORY

Author: Jithendra

https://github.com/CU-ECEN-5823/ecen5823-courseproject-JithendraHS

UPDATE 2 **DATE: 25th April 2025**

STATUS DESCRIPTION

Author: Suhas

- Integrated Ambient Light and Humidity/Pressure sensor drives into the codebase using a Finite State Machine architecture.
- Completed Low Power Node functionality by successfully receiving sensor data over BLE on the Si Connect app.
- Developed an SPI driver and ported the diskio and FatFS libraries to interface with an SD card breakout board.
- Interfaced the SD card breakout board and verified successful data write operations to the card.

Status: We are on track and up to date with the project.

The updates are detailed below:

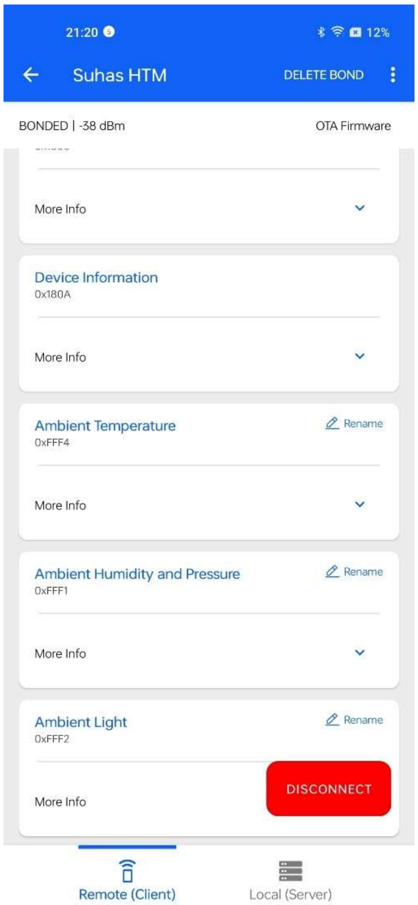
Ambient humidity, pressure, light, temperature integration on the Low Power Node:

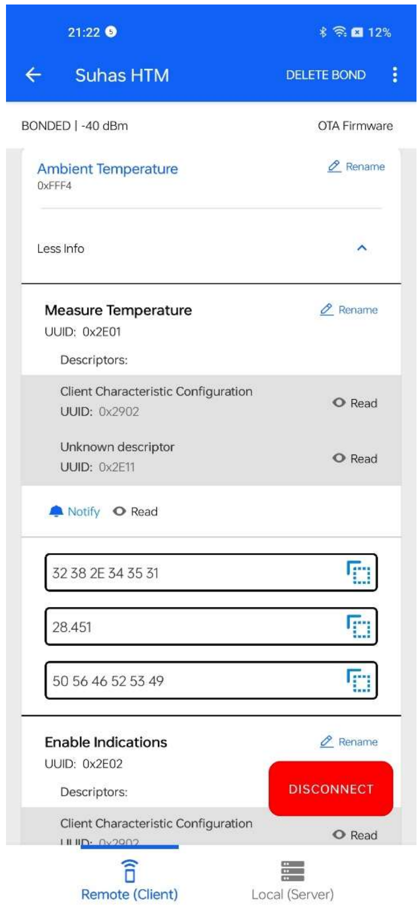
Author: Suhas

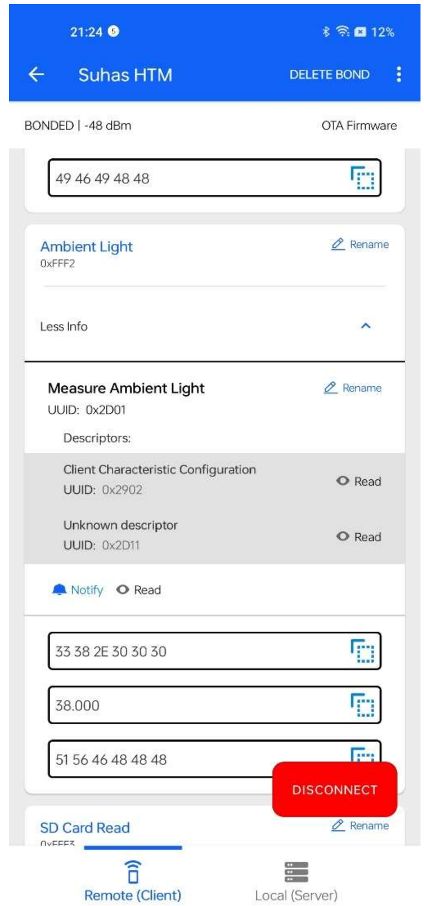
```
60000:Info :readTempFSM: Ambient Temperature: 28.375835 C
60000:Info :readHumidityFSM: Relative Humidity: 27.247070 % RH
60000:Info :readHumidityFSM: Pressure: 1.100000 hpa
60000:Info :readLightFSM: Ambient Light: 53.000000 mV
65000:Info :readTempFSM: Ambient Temperature: 28.386560 C
65000:Info :readHumidityFSM: Relative Humidity: 27.293945 % RH
65000:Info :readHumidityFSM: Pressure: 1.118750 hpa
65000:Info :readLightFSM: Ambient Light: 36.000000 mV
70000:Info :readTempFSM: Ambient Temperature: 28.386560 C
70000:Info :readHumidityFSM: Relative Humidity: 27.266602 % RH
70000:Info :readHumidityFSM: Pressure: 1.118750 hpa
70000:Info :readLightFSM: Ambient Light: 36.500000 mV
```

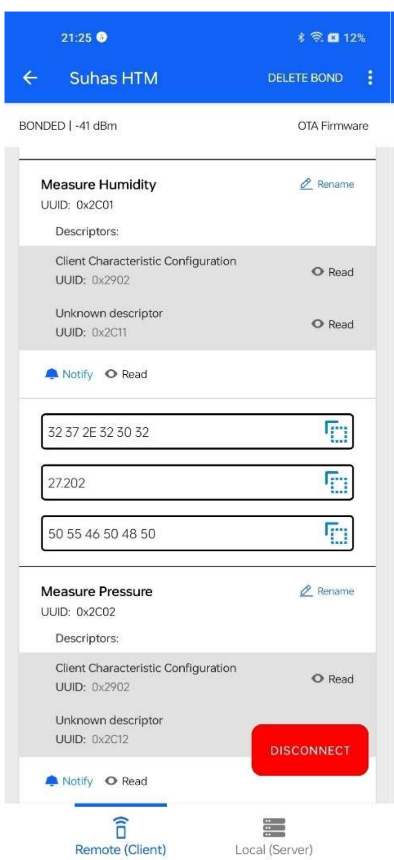
GATT Services Low Power Node:

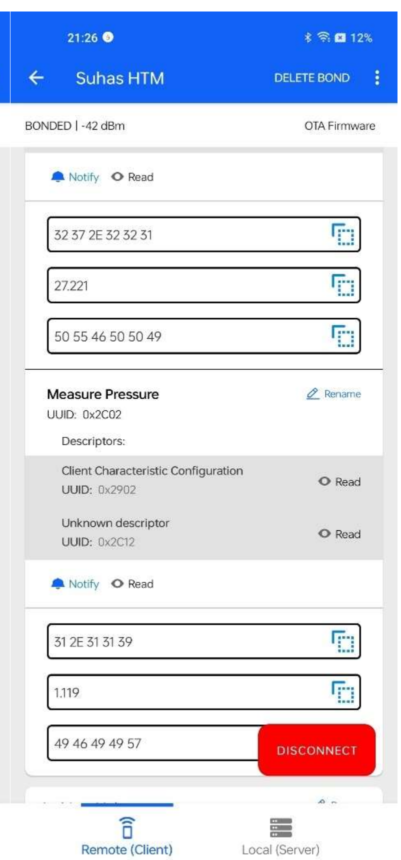
Author: Suhas











Updated GATT Services Table

Author: Suhas

SD Card Read Service (UUID FFF3)			
Characteristics	**Characteristics UUID**	**Characteristics ID**	**Length (Bytes)**
Read Temperature	2B01	read_temperature	17
Read Pressure	2B02	read_pressure	17
Read Humidity	2B03	read_humidity	17
Read Luminosity	2B04	read_luminosity	17
Ambient Luminosity (UUID FFF2)			
Characteristics	**Characteristics UUID**	**Characteristics ID**	**Length (Bytes)**
Measure Luminosity	2D01	measure_luminosity	17
Ambient Humidity and Pressure (UUID FFF1)			
Characteristics	**Characteristics UUID**	**Characteristics ID**	**Length (Bytes)**
Measure Humidity	2C01	measure_humidity	17
Measure Pressure	2C02	measure_pressure	17
Ambient Temperature (UUID FFF4)			
Characteristics	**Characteristics UUID**	**Characteristics ID**	**Length (Bytes)**
Measure Temperature	2E01	measure_temperature	17
Enable Indications	2E02	Enable_indications	1

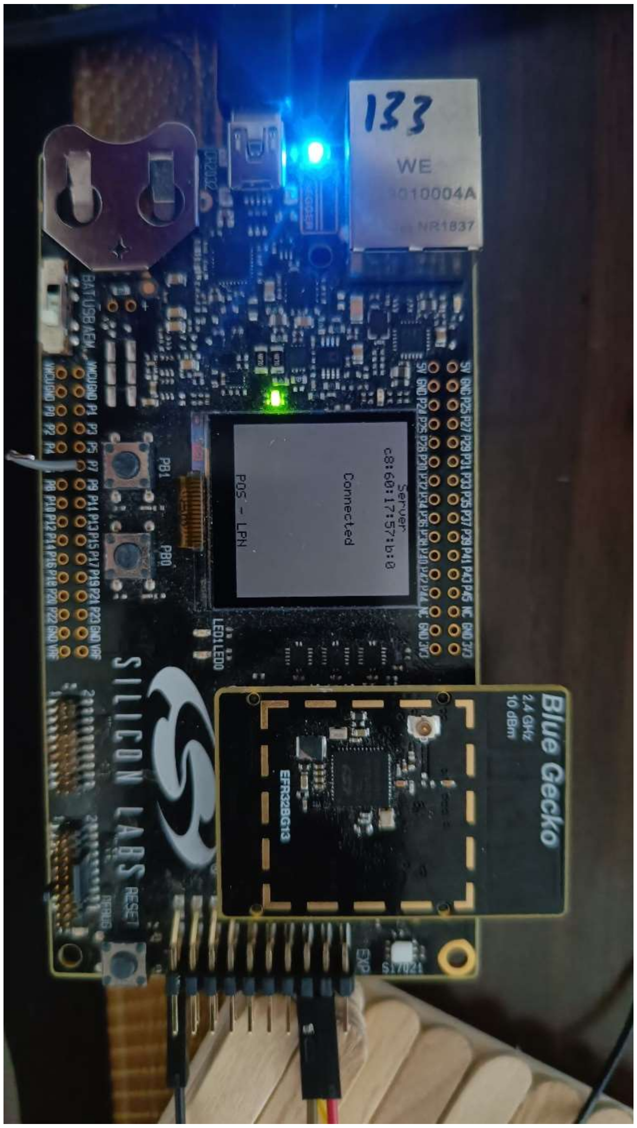
Added Ambient Temperature with Measure **Temperature** and **Enable Indications** characteristics to have more consistent data format across the sensors. The data from the Low Power Node will be passed as strings which makes it easier to write it to the SD card and read it. Enable Indication characteristics provides isolated control to start measurements from the sensors.

Note: All Characteristics have notified enable which makes it possible to get the latest sensor data.

Low Power Node – GATT Server

Author: Suhas

Before Indication is enabled

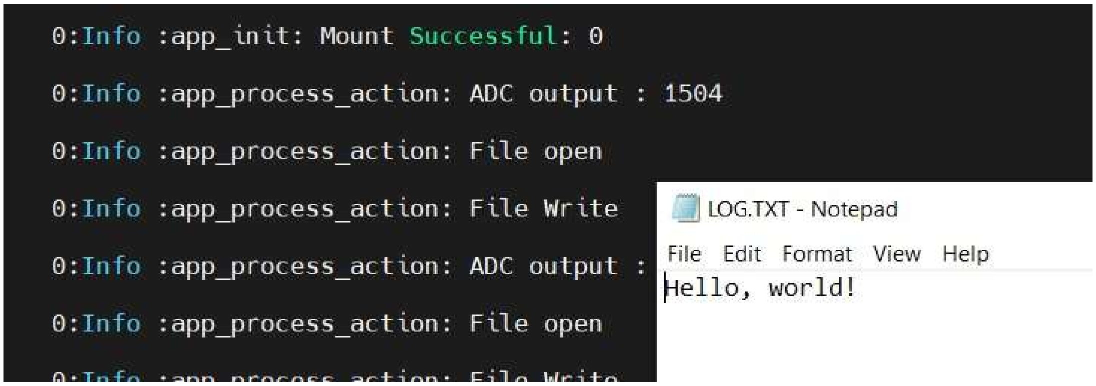


After Indication is enabled

SD card Interfacing:

Author: Jithendra

Ported FatFS open-source code with the help of partial implementation example available for silicon lab kits.



Resource:

1) http://elm-chan.org/fsw/ff/00index_e.html
2) https://github.com/ryankurte/efm32-base/blob/master/hardware/kit/common/drivers/microsd.c

SCHEDULE

Author: Jithendra

Task	Owner	Expected completion date	Actual completion date
Team formation	-	Apr 1	Apr 1
Project selection	Jithendra, Suhas	Apr 5	Apr 6
Sensor procurement	Jithendra	Apr 8	Apr 10
GATT service Ambient Humidity and pressure	Suhas	Apr 12	Apr 17
GATT service Ambient light	Jithendra	Apr 12	Apr 17
Ambient light interfacing	Jithendra	Apr 14	Apr 17
Ambient Humidity and pressure sensor interfacing	Suhas	Apr 14	Apr 17
Ambient light FSM design	Jithendra	Apr 16	
Ambient Humidity and pressure sensor interfacing FSM design	Suhas	Apr 16	Apr 17
LPN Bluetooth stack event handler implementation	Jithendra	Apr 20	Apr 24
Client Bluetooth stack event handler implementation	Suhas	Apr 20	
Bluetooth connection establishment and SD integration	Jithendra	Apr 22	Apr 25
Service handling integration	Suhas	Apr 24	
Unit and regression testing	Jithendra, Suhas	Apr 26	

TEST PLAN

Author: Jithendra

Updated Test plan:

Test Plan Table

Test Number	Test Description	Planned to check	Test Result (Pass/Fail)	Notes
1	Test LE timers and interrupts	04-15-2025	Pass(100%)	LE timer and I2C interrupt are happening at a regular delay as defined in the FSM
2	Read data and conversion of sensor data	04-18-2025	Pass(100%)	Able to read data from Luminosity, pressure, temperature, and humidity, conversion needs to be taken care of during BLE implementation
3	Verify the FSM transition and deterministic state for each sesnor	04-19-2025	Pass(100%)	Able to successfully do a transition
4	Bluetooth Connection establishment and close functionality	04-23-2025	Pass(100%)	Low powe node able to connect and publish data to the GATT services, verified using SI connect app
5	SD card logging with timestamp	04-24-2025	Inprogress(80%)	Able to write to the SD card, timeatamp, and code integration remaining
6	Sensor services enable and disable of indication and read data once connected	04-25-2025	Pass(100%)	Receiving all sensor data successfully
7	LCD test	04-26-2025		
8	Low power mode current measurement	04-27-2025		
9	Final full system testing	04-28-2025		

Overall : 60%

GITHUB REPOSITORY

Author: Jithendra

https://github.com/CU-ECEN-5823/ecen5823-courseproject-JithendraHS

ECEN5823 IoTEF

PLANT OBSERVATORY SYSTEM

FINAL PROJECT REPORT BY –

TEAM 17

JITHENDRA H S

SUHAS S R

DATE: 1ST MAY 2025

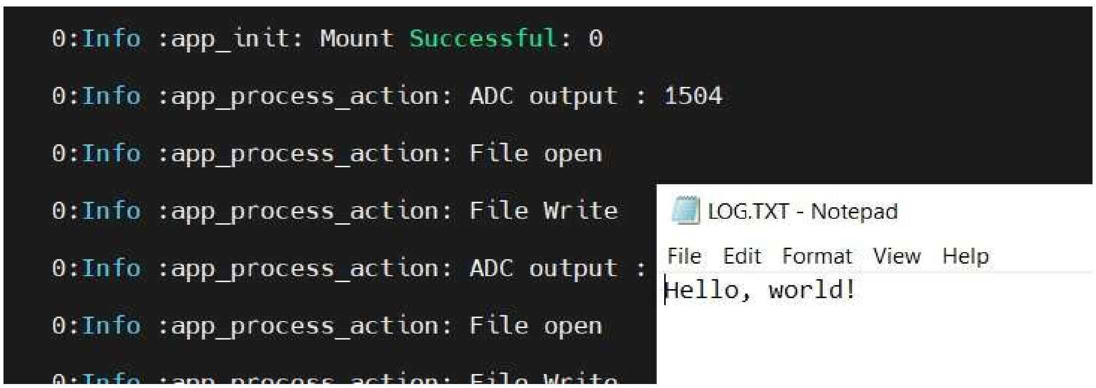
STATUS DESCRIPTION

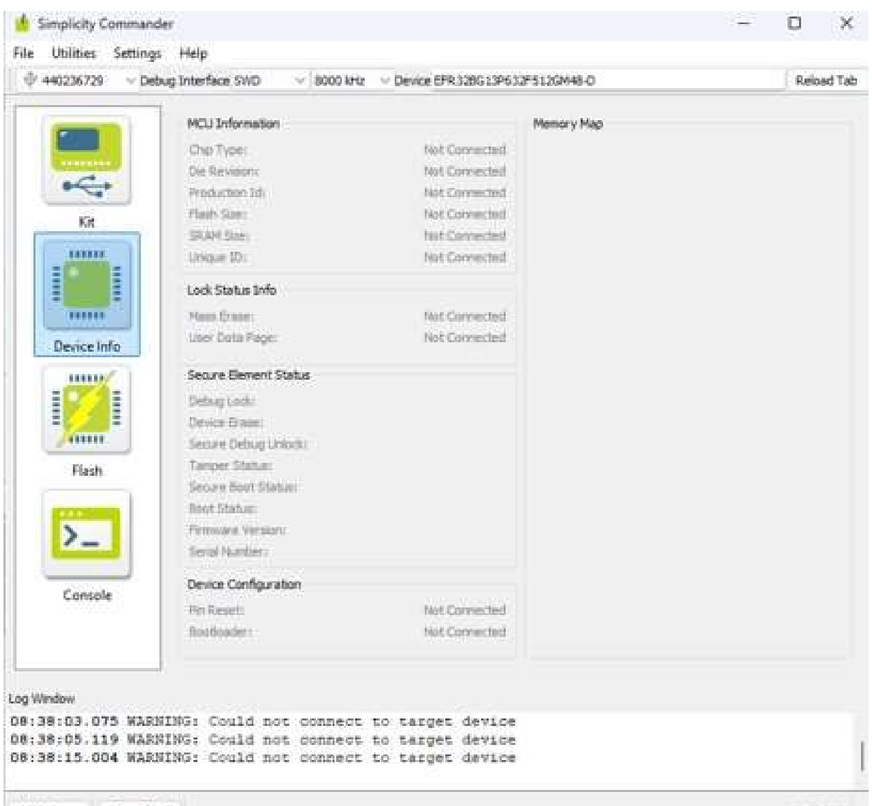
Author: Suhas

Node	Role	Functionality	Requirement Met?
Central Node (CN)	GATT Client to LPN	Receives data from LPN	Yes
	GATT Server to Si Connect app	Logs data to SD card, after packetization	Partial, Integration caused MCU crashes
		Maintains simultaneous BLE connections	Yes
		Must be Bonded to send notifications to GATT Client	Yes
		Active in **EM1** during SD logging Sleeps in **EM2** when idle	Yes
Low Power Node (LPN)	GATT Server	Interfaces with	
		Si7021 (Temperature)	Yes
		BME280 (Humidity/Pressure)	Yes
		TEMPT6000 (Ambient Light)	Yes
		Advertises and notifies sensor data	Yes
		Wakes to **EM1** every 5 seconds for I2C communication Sleeps in **EM2**, maintains BLE connection	Yes

Issues Faced and Reasons for Incompletion

- The **SD card module** was assigned to **USART2**, as **USART0** and **USART1** were already used by the **display**, **SPI flash**, and **VCOM**.
- Porting **FATFS** to the EFR32BG13 took significant effort. After successful porting, we encountered issues where the MCU would **crash or become unresponsive**, requiring **bootloader recovery** and re-flashing.
- Though we were able to **mount the FATFS file system**, **open/write/read files**, and verify functionality in standalone tests, the **integration into the final CN project led to instability and crashes**, which we could not resolve in the given time.
- Debugging this took a substantial portion of our time near the end of the project timeline, preventing a stable integration.

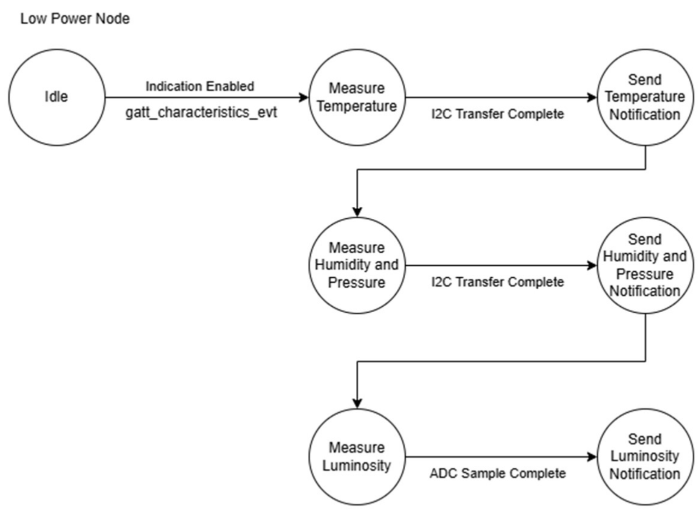


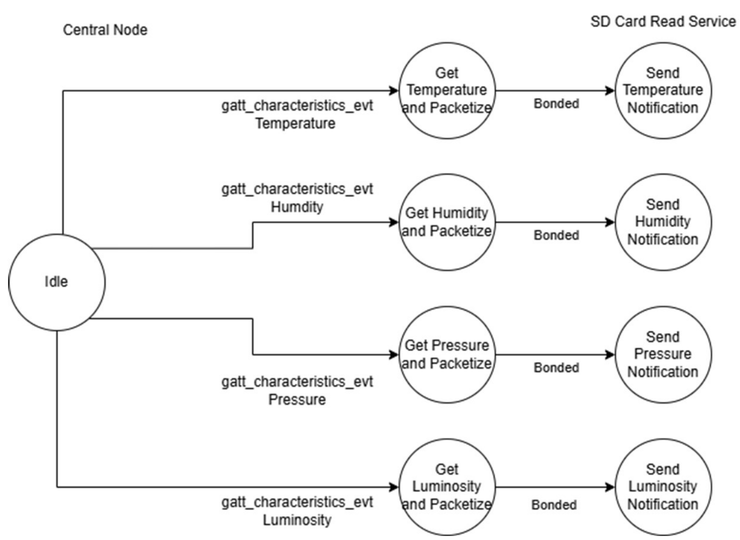


Design Updates

- While the **overall system architecture remained the same**, we made some changes to the **data representation and path**:
  - Sensor data was sent as a **formatted string** (including timestamp and units) to make SD logging easier.
  - This changed the **expected packet size** from the original design. The packet size was updated to **36 bytes** to accommodate the string format.

Data Flow FSM

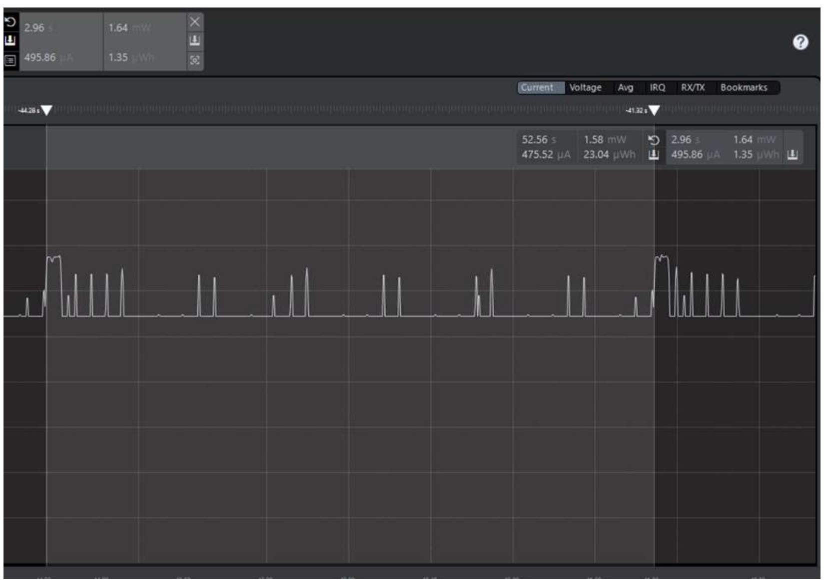




How well did you achieve your energy modes/goals/targets?

Author: Suhas

Advertising (232.14 uA for 3 advertising packets)



One Measurement Period (~500 uA)

Low Power Node (LPN)

- **Target Behavior**:
  - Operate in **EM1** during I2C sensor reads and BLE communication.
  - Transition to **EM2** between measurements to conserve energy while maintaining BLE connection.
- **Measurement Period**: Every 5 seconds
- **Measured Average Current**: ~**500 µA** per cycle
- **Result**:
  - Successfully met the expected low-power profile during operation.
  - The LPN remained in EM2 most of the time, only briefly waking up to EM1 to perform sensor reads and data transmission.

Central Node (CN)

- **Target Behavior**:
  - Wake up to **EM1** when receiving data from LPN and logging to SD card.
  - Sleep in **EM2** when idle.
- **Result**:
  - EM1 and EM2 transitions worked correctly.
  - However, due to instability during SD card integration, the node sometimes remained longer in EM1 than planned.
  - Optimizations were limited due to debugging focus during final stages.

What Was Learned

Suhas

- Gained hands-on experience in designing and implementing accurate state machines for managing sensor measurement sequences and ensuring reliable timing.
- Deepened understanding of the Bluetooth Low Energy (BLE) stack, specifically:
  - Working with indications, notifications, and BLE security.
  - Enabling a single node to act as both a GATT Client and Server simultaneously.
- Explored interfacing with Flash memory and learned that:
  - It involves more than just connecting the SPI interface.
  - Requires proper file system formatting (e.g., FATFS), porting external libraries, and adapting them to fit the application logic and constraints.

Jithendra

- Got an opportunity to create state machines to read sensor values using ADC and manage timing properly.
- Used an SD card for the first time in a project to save sensor data. Learned that:
  1) It's not just about using SPI, need to handle file systems too.
  2) Ported the FatFS library and made it work with my microcontroller setup.
- Built a working data logger and now feel confident to use SD cards in future projects.
- Learned more about Bluetooth Low Energy (BLE):
  1) Designed and set up GATT services to act as a peripheral.
  2) Understood how BLE devices share data and connect to others.
- This project helped me understand full BLE IoT systems, starting from sensor data to sending it wirelessly, advertising, scanning and bonded.

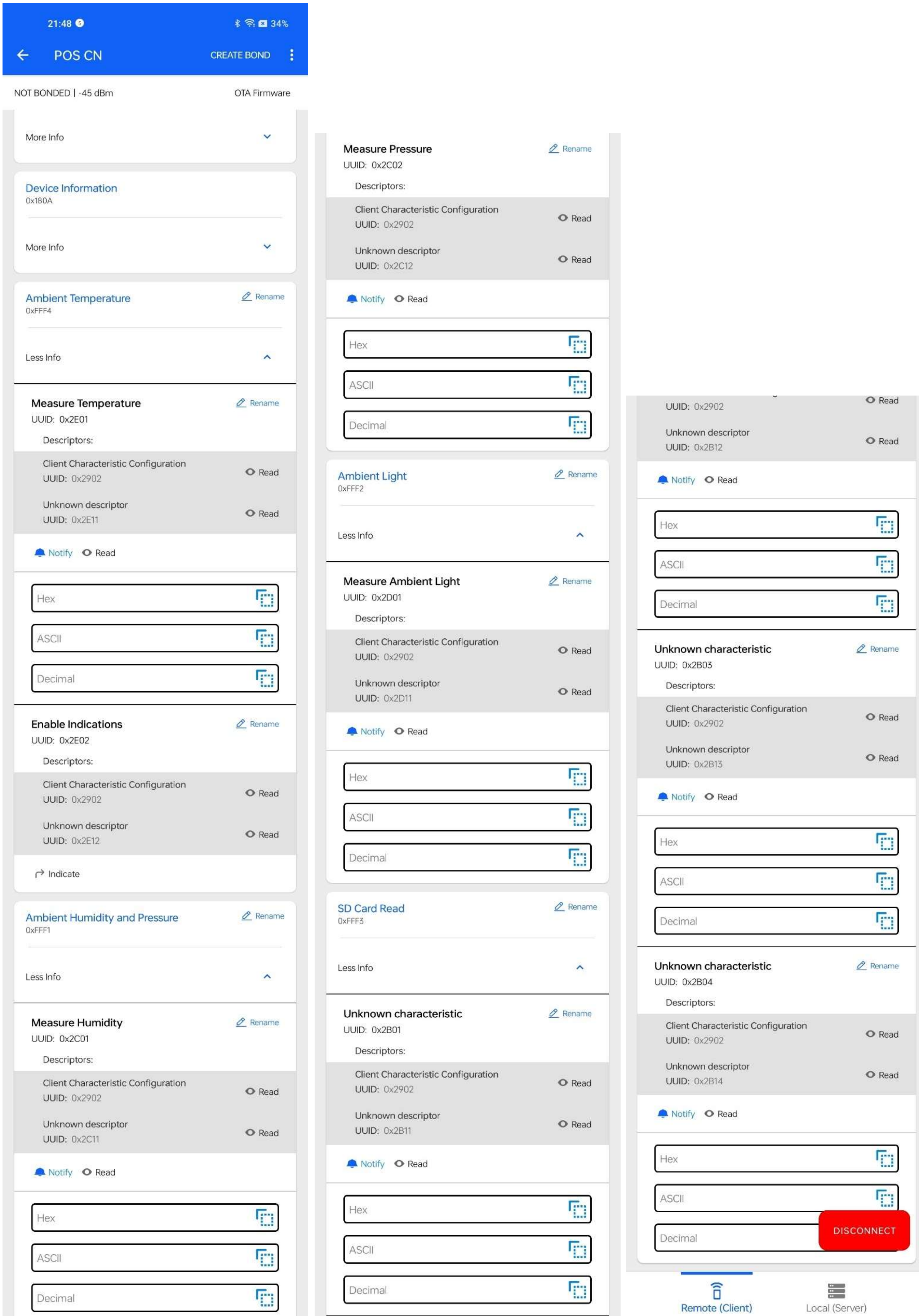
Project implements required features:

Author: Suhas

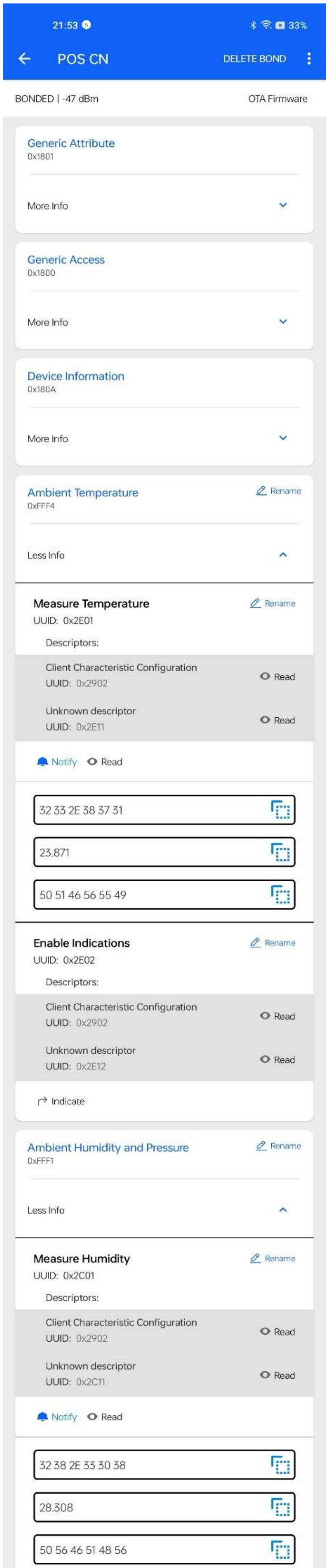
Project implements an encrypted link:

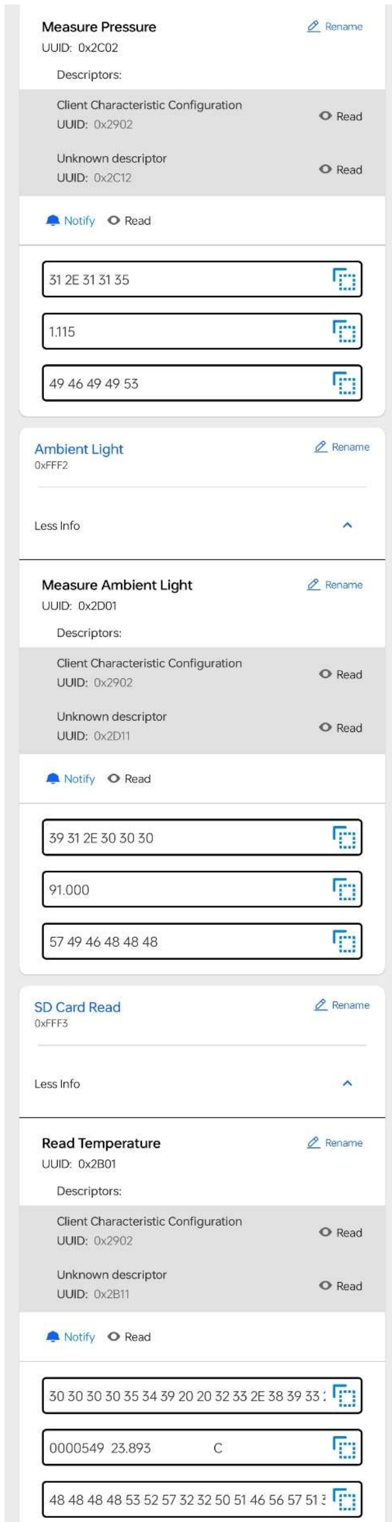
When the Central Node acts as GATT server the GATT client must be bonded to get notifications and read values from the service.

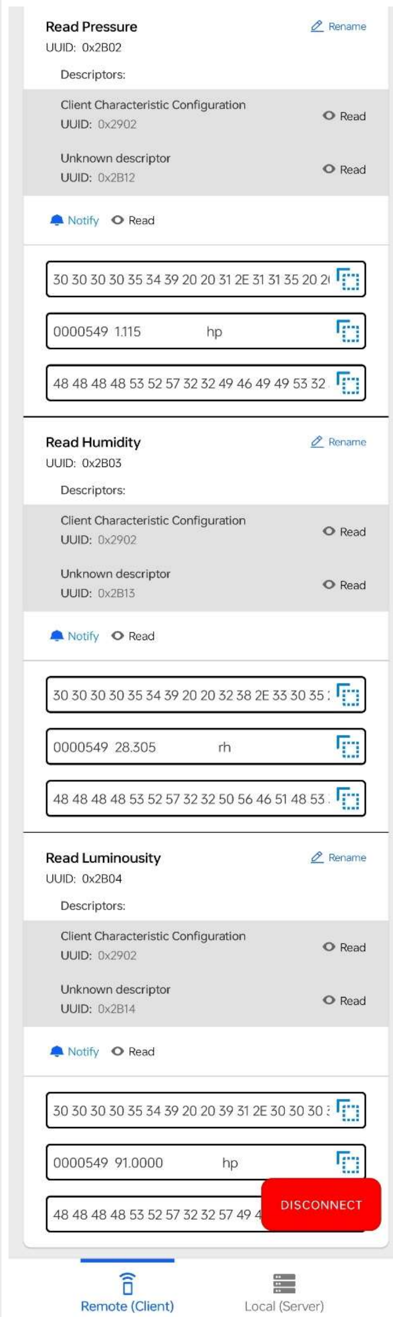
When Si connect tires to get notifications when not bonded:



When Si connect tires to get notifications when bonded:







Project implements the required number of new sensors/devices:

- The Project Low Power Node interfaces with TEMPT6000 Luminosity sensor and BME280 Humidity/Pressure sensors.
- The Central Node Interfaces with the SD Card breakout board.

Project implements the required number of GATT services, Client functionality, and/or Mesh node elements and models:

SD Card Read Service (UUID FFF3)			
Characteristics	**Characteristics UUID**	**Characteristics ID**	**Length (Bytes)**
Read Temperature	2B01	read_temperature	36
Read Pressure	2B02	read_pressure	36
Read Humidity	2B03	read_humidity	36
Read Luminosity	2B04	read_luminosity	36
Ambient Luminosity (UUID FFF2)			
Characteristics	**Characteristics UUID**	**Characteristics ID**	**Length (Bytes)**
Measure Luminosity	2D01	measure_luminosity	17
Ambient Humidity and Pressure (UUID FFF1)			
Characteristics	**Characteristics UUID**	**Characteristics ID**	**Length (Bytes)**
Measure Humidity	2C01	measure_humidity	17
Measure Pressure	2C02	measure_pressure	17
Ambient Temperature (UUID FFF4)			
Characteristics	**Characteristics UUID**	**Characteristics ID**	**Length (Bytes)**
Measure Temperature	2E01	measure_temperature	17
Enable Indications	2E02	Enable_indications	1

Project implements LCD:

LPN LCD Display:

Low Power Node While Advertising

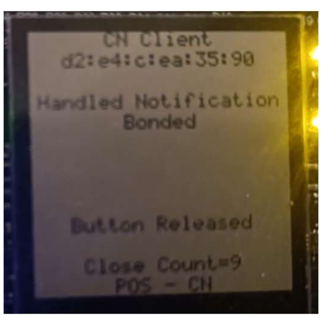
Low Power Node when connected and measuring (Indication enabled (LED0))

CN LCD Display:

Central Node when Discovering and Advertising

Central Node when it has discovered LPN, but it keeps advertising

Central Node it receives Bonding request



Central Node when bonded

SCHEDULE

Author: Jithendra

Task	Owner	Expected completion date	Actual completion date
Team formation	-	Apr 1	Apr 1
Project selection	Jithendra, Suhas	Apr 5	Apr 6
Sensor procurement	Jithendra	Apr 8	Apr 10
GATT service Ambient Humidity and pressure	Suhas	Apr 12	Apr 17
GATT service Ambient light	Jithendra	Apr 12	Apr 17
Ambient light interfacing	Jithendra	Apr 14	Apr 17
Ambient Humidity and pressure sensor interfacing	Suhas	Apr 14	Apr 17
Ambient light FSM design	Jithendra	Apr 16	Apr 26
Ambient Humidity and pressure sensor interfacing FSM design	Suhas	Apr 16	Apr 17
LPN Bluetooth stack event handler implementation	Jithendra	Apr 20	Apr 24
Client Bluetooth stack event handler implementation	Suhas	Apr 20	APR 29
Bluetooth connection establishment and SD integration	Jithendra	Apr 22	Apr 25
Service handling integration	Suhas	Apr 24	APR 30
Unit and regression testing	Jithendra, Suhas	Apr 26	APR 30

TEST PLAN

Author: Jithendra

Test Plan Table

Test Number	Test Description	Planned to check	Test Result (Pass/Fail)	Notes
1	Test LE timers and interrupts	04-15-2025	Pass(100%)	LE timer and I2C interrupt are happening at a regular delay as defined in the FSM
2	Read data and conversion of sensor data	04-18-2025	Pass(100%)	Able to read data from Luminosity, pressure, temperature, and humidity, conversion needs to be taken care of during BLE implementation
3	Verify the FSM transition and deterministic state for each sesnor	04-19-2025	Pass(100%)	Able to successfully do a transition
4	Bluetooth Connection establishment and close functionality	04-23-2025	Pass(100%)	Low powe node able to connect and publish data to the GATT services, verified using SI connect app
5	SD card logging with timestamp	04-24-2025	Failed(100%)	Able to write to the SD card, timestamp, and standalone worked, but code integration failed
6	Sensor services enable and disable of indication and read data once connected	04-25-2025	Pass(100%)	Receiving all sensor data successfully
7	LCD test	04-26-2025	Pass(100%)	Able to test the BLE address, connection status, and sensor status
8	Low power mode current measurement	04-27-2025	Pass(100%)	Measured using the SI energy profiler
9	Final full system testing	04-28-2025	Pass(100%)	Everything worked as expected apart from SD card logging

Overall : 100%

WORK DISTRIBUTION

Author: Jithendra

Jithendra worked on interfacing sensor modules such as the Ambient Light sensor and ported the FatFS library for the EFR32BG13 microcontroller. He also designed the SPI driver to store and retrieve data from the SD card on the central node. Additionally, he developed a finite state machine (FSM) for client-side sensor data acquisition, created the GATT service for ambient light measurement, and brought up the Bluetooth stack on the peripheral node.

Suhas worked on interfacing sensor modules, including ambient humidity and pressure sensors using I2C, as well as utilizing the onboard temperature sensor. He handled the conversion of sensor data into string format to ensure consistency for transmission to the mobile client. He also designed the FSM for sensor operation and developed GATT services and the BLE stack on the server side, enabling the node to function as both central and peripheral based on project requirements.

GITHUB REPOSITORY

Author: Jithendra

https://github.com/CU-ECEN-5823/ecen5823-courseproject-JithendraHS

Folder structure:

1) Central node: Simplicity project files related central node
2) Low power node: Simplicity project files related peripheral node
3) Final sensor interface: Simplicity project files to SD card proof of concept implementation
4) Presentation slides

Demo link:

https://drive.google.com/drive/folders/1qdvMlqueotaw4wapRebJ4QaJb0W57iqI?usp=sharing