ECEN5823

IOT EMBEDDED FIRMWARE

VIGNESH VADIVEL

SHRINITHI VENKATESAN

APRIL 21, 2023
UNIVERSITY OF COLORADO BOULDER

CONTENTS

Section 1 - Project Proposal

Section 2 - Update 1

Section 3 - Update 2

Section 4 - Final Report

Section 1 - Project Proposal

1.1 Student Names

1. Vignesh Vadivel
2. Shrinithi Venkatesan

1.2 Project overview

Section Author: Shrinithi Venkatesan

For the ECEN 5823 Internet of Things and Embedded Firmware (IoTEF) course project, we have chosen to implement Option-2, where we will use Bluetooth Smart with a server and a client model.

One of the breakthrough advances in medicine is remote patient monitoring. Healthcare practitioners can employ remote patient monitoring systems to gather and send health data from patients in real-time using linked devices and sensors. This enables for continuous monitoring of vital signs, symptoms, and medical problems, allowing healthcare practitioners to make educated decisions regarding patient treatment without needing to see the patient in person.

Delving specifically on our application today, some studies from the National Library of Medicine have shown that body temperature is closely related to the heart rate in Children. For every one degree increase in centigrade, the heart rate can increase up to 10 beats per minute, rooting to cause fatal issues if unattended almost immediately. As time and data plays a major role in the heath applications, our motto is to provide a solution that could solve the above issue effectively. Hence, arriving at the solution, we want to implement a system that would monitor temperature continuously and in case of any unusual fluctuations from the normal ranges, it would call in effect for the heart rate monitoring device, which would keep immediately update the necessary required reports to the user(doctor) remotely. This would allow the medical facilities to cut some time requiring to generate the routine timely reports on temperature and heart rate measurements and thereby remote access to the reports would effect in immediate attention to the patient.

Discussing on the details of our module, the server device will implement a temperature sensor to sense the temperature value periodically and only when the unusual fluctuations are recorded, the client device holding a heart rate sensor will come in into action. An alert in made to notify the user(doctor) in case the system records deviated fluctuation ranges from normal.

IoT devices in medicine collect sensitive patient data, making security and privacy a top concern. Medical data is highly sensitive and private, and the potential for unauthorized access or breaches of this data is a major concern. Thereby, our system will also address this issue as we plan to encrypt our server and client devices.

Also, by implementing low energy modes, our system intends to acquire data more effectively. Overall, our system intends to provide efficiency in data acquisition, saves time and power and securely holds and transceives the data.

1.3 High Level Requirements.

Section Author: Vignesh Vadivel

1. The system will consist of a BLE server and a BLE client, both running on a Blue Gecko Board.
2. The server will establish an encrypted link with the client via bonding.
3. The Blue Gecko board acting as the server will periodically read values from the external temperature sensor every second using LETIMER0.
4. The server shall implement and advertise a custom GATT service and can have one custom characteristic to send the temperature value.
5. The server shall send the temperature measurement to the client only if the temperature value goes beyond the threshold.
6. The server code shall maintain the lowest possible energy mode for this application.
7. The client shall receive periodic temperature values from the server over the BLE radio.
8. Upon receiving the temperature value, the client shall read the heart rate value from the sensor and check if it is within the proper range.
9. After reading the heart rate value, the client shall write it to the GATT database, and the server can read the value from the GATT database.
10. The server shall display both the heart rate and temperature values on the LCD display.
11. Upon measuring the heart rate data, the client will process physical alerts through the LEDs.
    11.1 If the heart rate is normal and only the temperature is not within the threshold, LED0 on the client side shall be illuminated.
    11.2 If both the heart rate and temperature values are not within the threshold, LED0 and LED1 shall be illuminated on the client side.
12. Both the server and client shall incorporate an LCD display.
    12.1 The server and client shall display the corresponding naming and addresses.
    12.2 The server shall display the current connection status - Advertising/Connected/Bonded.
    12.3 The client shall display the current connection status - Discovering/ Connected/ Bonded/ Handling Indications.
    12.4 The server and client shall display the corresponding temperature and heart rate values read from the GATT database.

1.4 High Level Design

1.4.1 Design Overview

Section Author: Shrinithi

The design consists of a pair of Blue Gecko Boards, one that hosts the server code and the other that hosts the client code. Once the bonding is successful between the server and client, the server will measure the temperature data updating through the **BME280** external sensor on a periodic basis every second. The timer used to perform periodic analysis is the LETIMER0. There will be corresponding GATT characteristics that will record the abnormal temperature in the GATT database and will send the data to the client**.**

The client will receive indications over the BLE radio. On reception, the client will enable the heart rate sensor, **MAX30102** to measure the heart beats per minute and the abnormal ranges are updates through the GATT database, which is fetched by the server to display. An LCD display is used as a display module for both the server and client devices. Based on the range of the readings on both server and client, the client takes action on physical indications. If only the temperature measurements received were abnormal, an alert through lighting LED0 is performed by the client. In case, both the heartrate and temperature both don’t fall under the normal range, the client illuminates both LED0 and LED1.

1.4.2 Description of data types

Section Author: Shrinithi

Measurement	Units	Data Type	Valid/Allowed Values (Range)
Temperature	Degree celsius	uint8_t	0 to 255
Heart rate	Beats/min	uint8_t	0 to 255

Table1.1 Description of data types

1.4.3 Wireless communication details

Section Author: Shrinithi

The GATT server and GATT client implements and advertises one custom GATT service. Each GATT service has one GATT characteristic. The GATT server will perform health thermometer service and it will periodically measure temperature values and stores abnormal temperature readings on the GATT server database. The health thermometer service will then perform data transfer of recorded abnormal temperature reading to the GATT client. The wearable pulse rate service performs heart rate measurement and stores it in the GATT database.

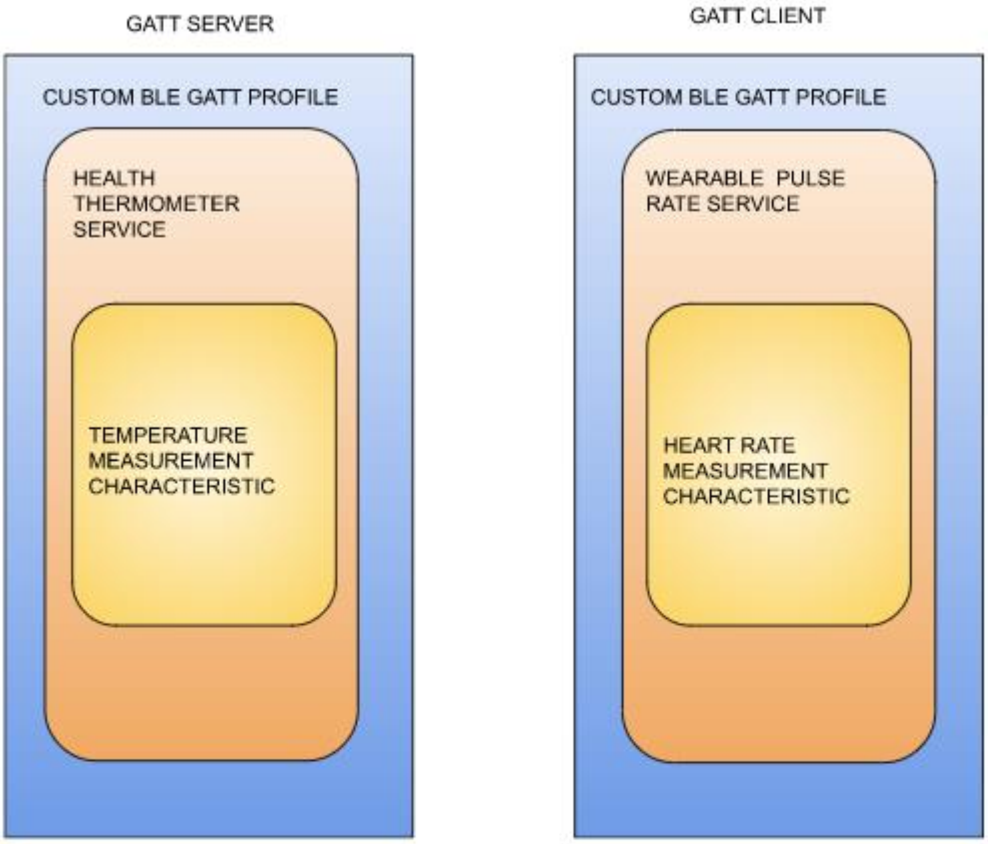

GATT SERVER
CUSTOM BLE GATT PROFILE
HEALTH THERMOMETER SERVICE
TEMPERATURE MEASUREMENT CHARACTERISTIC
GATT CLIENT
CUSTOM BLE GATT PROFILE
WEARABLE PULSE RATE SERVICE
HEART RATE MEASUREMENT CHARACTERISTIC

Fig1.1 Wireless communication block diagram

1.4.4 Hardware Block diagram

Section Author: Vignesh

The hardware block diagram of the two subsystems present namely: GATT server and GATT client. The temperature sensor, BME280 is integrated to the GATT server externally using I2C interface. The I2C bus on the Blue Gecko Board will help write and read to the BME280 sensor. The LCDs found on both sides are part of the Blue Gecko Board. The heart rate sensor, MAX30101 is interfaced to the GATT client subsystem also through I2C interface. For alert and indications, apart from the LCD display, the onboard LEDs are enabled through GPIO interface to the client. The server and clients communicate through the BLE radio.

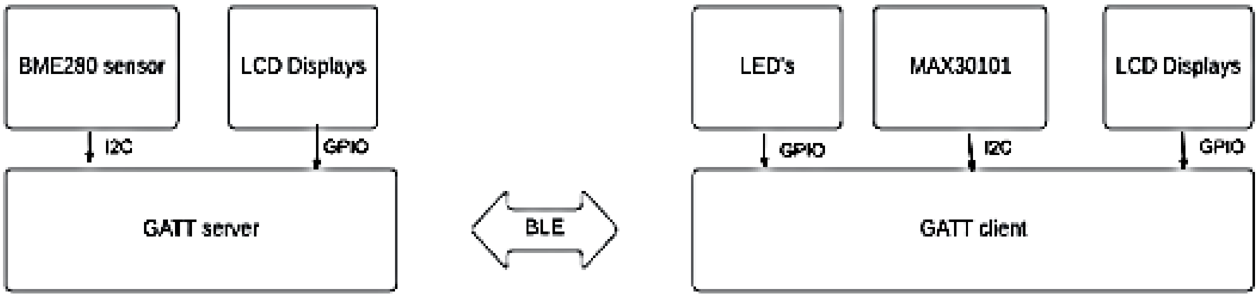


Fig1.2 Hardware block diagram

The hardware needed for major requirements are listed below:

S. NO	PART	Quantity
1	Photodetector Breakout - MAX30101	1
2	Atmospheric Sensor Breakout - BME280	1
3	Blue Gecko board	2

Table1.2 Hardware components

1.4.5 Software Block Diagram

Section Author: Vignesh

1. LETIMER0 would keep generating interrupts for every second, and temperature measurement would be taken in the server.
2. Once the temperature value is ready, we will check if it is in normal range. If not, we will send the temperature measurement value to the client and display the abnormal temperature value in the LCD (server).
3. Once the client receives temperature measurement, it will get the heart rate sensor value and write it to the GATT database. LED0 in the client would be turned ON (to indicate abnormal temperature measurement) and if the heart rate value is also not within the range, we will turn ON LED1.
4. Once the data has been written to the GATT database on the client side, the server would read it and display it in the server LCD.

GATT Server: Software Block Diagram:

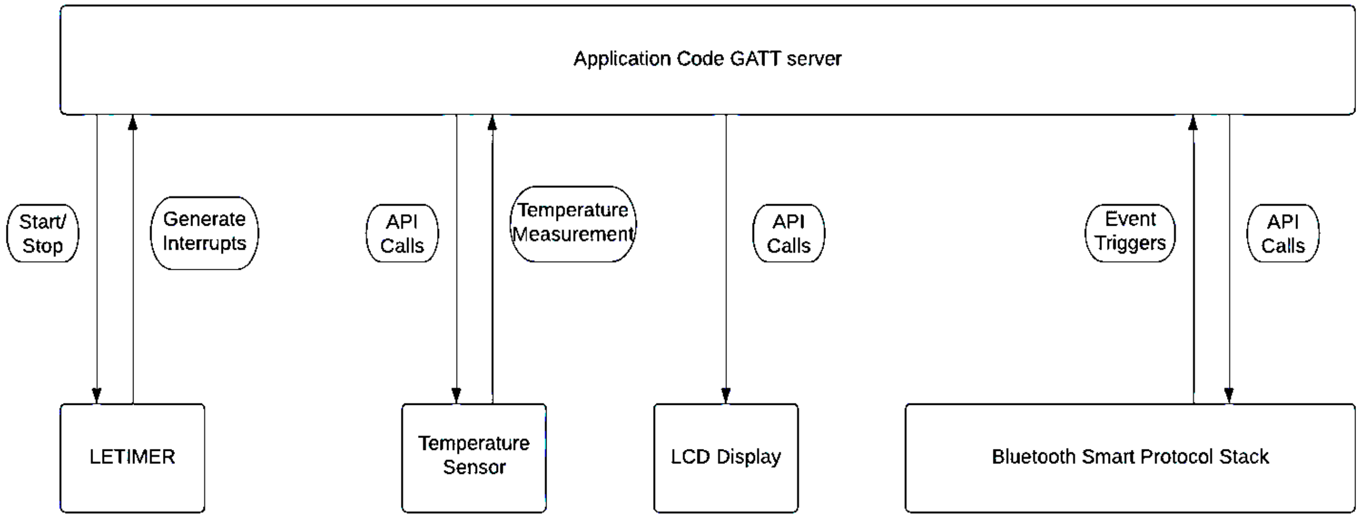


Fig1.3 GATT Server Software block diagram

GATT Client: Software Block Diagram:

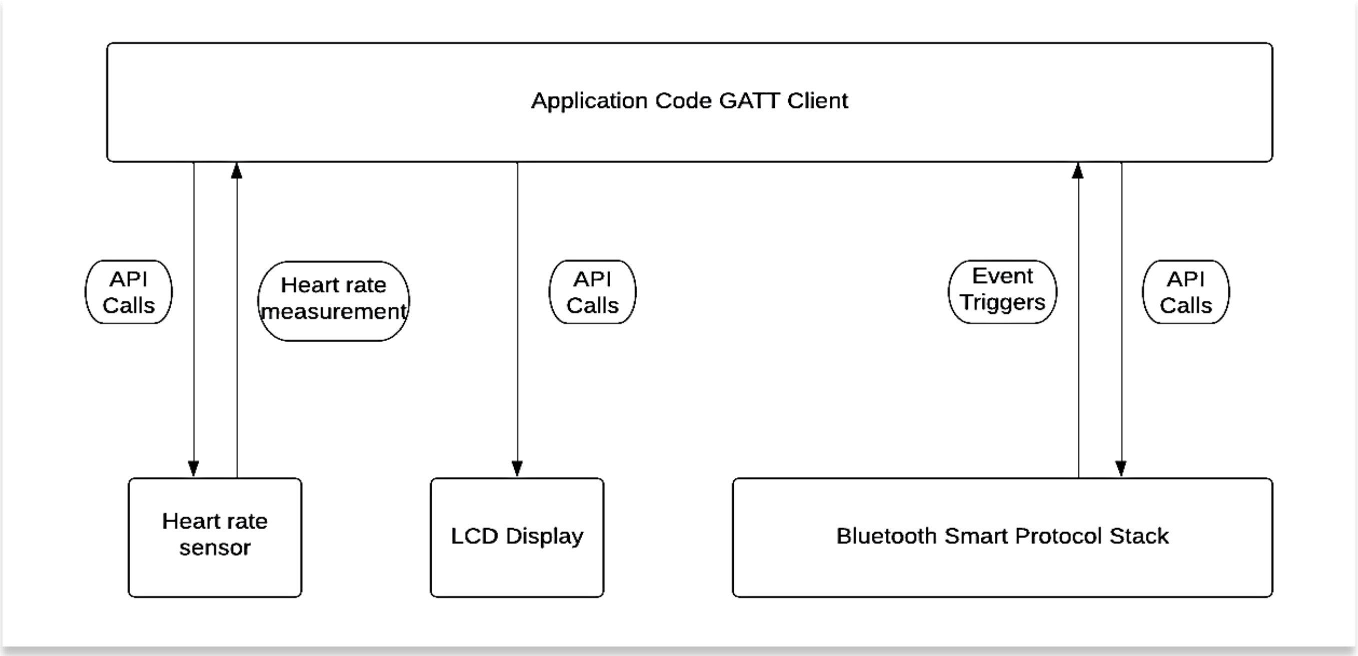


Fig1.4 GATT Client Software block diagram

Software Flowchart

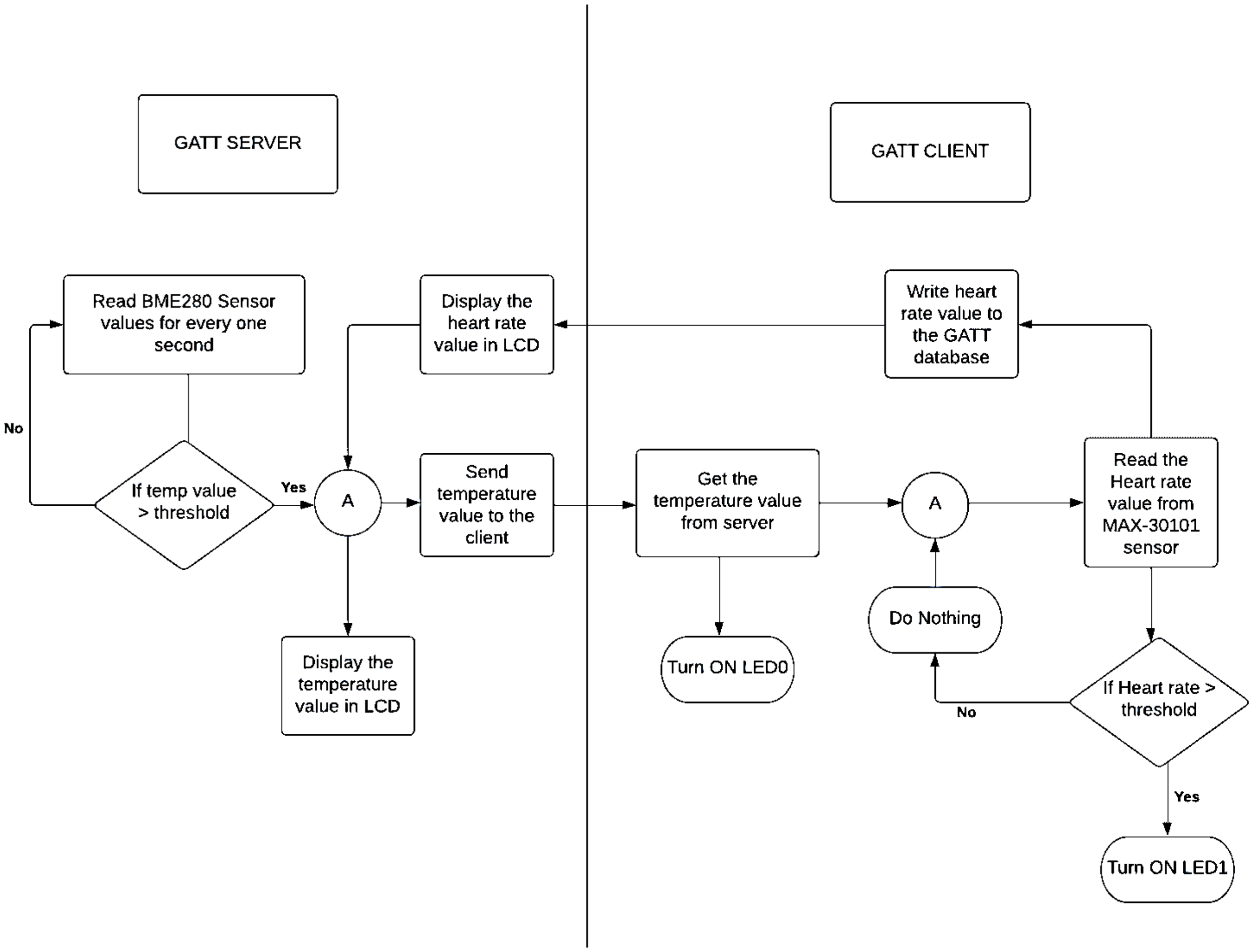

GATT SERVER
GATT CLIENT
Read BME280 Sensor values for every one second
Display the heart rate value in LCD
Write heart rate value to the GATT database
No
If temp value > threshold
Yes
A
Send temperature value to the client
Get the temperature value from server
A
Read the Heart rate value from MAX-30101 sensor
Do Nothing
Display the temperature value in LCD
Turn ON LED0
If Heart rate > threshold
No
Yes
Turn ON LED1

Fig1.5 Software flow block diagram

1.4.7 Division of Labour

Section Author: Shrinithi

Hardware components viz. Blue Gecko boards, Temperature Sensor, Heart Rate sensor are bought off the shelf, hence design of these hardware components is limited just to interconnections. Development related to functionality of this roughly contributes to 10 % of the project.

Majority, over 90% of the efforts are in the implementing software functionality. Right from functionality of connections, bonding to sensor data analysis and alerts based on those are to be implemented in software. LCD updates and LED indications also require software functionality.

1.4.8 User Interface
Section Author: Shrinithi

The GATT server will have limited UI functionality, just the LCD display. The GATT client will feature an advanced UI that will use the LCD display and LEDs to notify the users.

Their usage is individually discussed below.

GATT server:
LCD display: To display the board name as a server, server address, connection status, boding status, temperature values and heartrate values, and finally project name.

GATT client:
LCD Display: To display the board name as a server, server address, connection status, boding status, temperature values and heartrate values, and finally project name.
LEDs: LED0 is blinked only if the temperature readings are abnormal and both LED0 and LED1 is blinked if measured both abnormal temperature and heart rate.

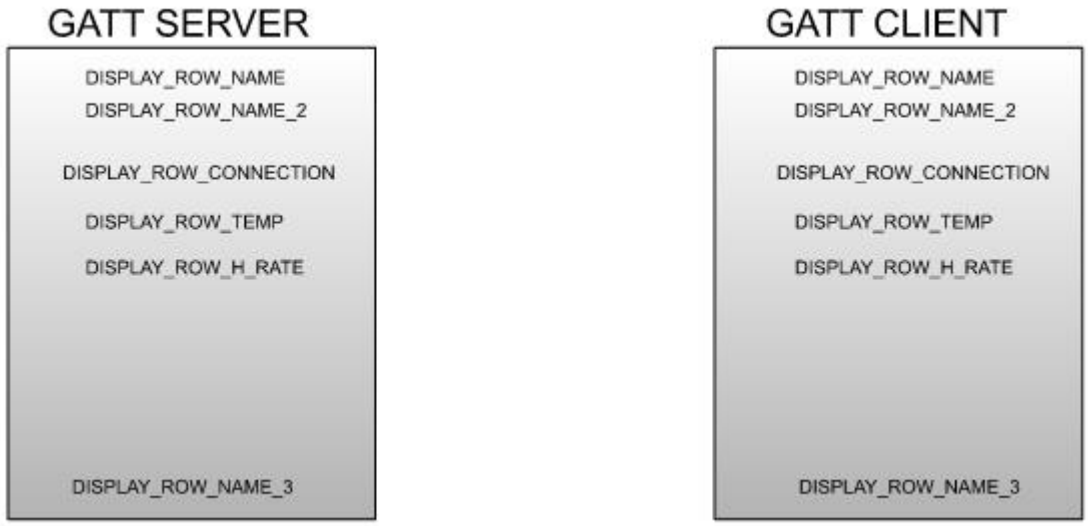


Fig1.6 UI block diagram

DISPLAY_ROW_NAME: GATT server/GATT client
DISPLAY_ROW_NAME_2: GATT server/GATT client address
DISPLAY_ROW_NAME_3: Project name
DISPLAY_ROW_CONNECTION: (GATT server): Advertising/Connected/Bonded
DISPLAY_ROW_CONNECTION (GATT client): Discovering/Connected/Bonded
DISPLAY_ROW_TEMP: Temperature value measured
DISPLAY_ROW_H_RATE: Heart rate value measured

1.5 Subsystem Summary

Section Author: Shrinithi

The project will consist of two subsystems in total and their names go by the role each subsystem is meant for i.e. GATT server and GATT client. Both these subsystems are a combination of hardware and software. Once connection is established between the boards, an encryption link is activated and the boards are bonded to improve security enhancements. The GATT server will then collect temperature samples, updating the GATT server database and reporting it to the GATT client only in case of abnormal readings. The GATT server will update the abnormal temperature value to the user using the LCD interface. Upon receiving the temperature readings, the GATT client will then come into effect for measuring heart rate. If abnormal heart rates are measured, the LCD displays the readings on both server and client. The GATT client has an enhanced user interface to alert the user through blinking LED0 in case of temperature abnormalities and both LED0 and LED1 in case of temperature and heart beat abnormalities.

1.5 Test Plan:

Section Author: Vignesh and Shrinithi

S. NO	Test Description	Planned Date	Actual Date	Test Result	Notes
1	A test to check the functioning of LETIMER, which is needed to take periodic temperature measurement	04/15/2023	TBD	To do	
2	A test to check the BME280 sensor reads the correct value	04/15/2023	TBD	To do	
3	A test to check the bonding happens properly between two Gecko boards	04/16/2023	TBD	To do	
4	A simple test to check if both the LEDs are working fine in the client board	04/16/2023	TBD	To do	
5	A simple test to check if all the rows in LCD works as expected	04/16/2023	TBD	To do	
6	Max30101 Heart rate sensor test	04/17/2023	TBD	To do	
7	A test to send characteristic temperature value to the client from server	04/17/2023	TBD	To do	

Table1.3 Test Plan

1.5 Proposed Schedule:

Section Author: Vignesh and Shrinithi

Task	Student(s)	Target Date	Expected date
Server design	Vignesh Vadivel	Apr 15, 2023	
Server code Implementation	Vignesh Vadivel	Apr 20,2023	
Client design	Srinithi Venkatesan	Apr 15, 2023	
Client code Implementation	Srinithi Venkatesan	Apr 20,2023	
Overall Integration	Vignesh and Srinithi	Apr 23, 2023	
Complete testing	Vignesh and Srinithi	Apr 25, 2023	

Table1.4 Proposed Schedule

Section 2 - Update 1

2.1 Status:

Section Author: Shrinithi

We have procured all parts required for the project as on dates mentioned below. Interfacing these sensors electrically with Blue Gecko board was easy as all the sensor modules have breakout boards and have circuitry compatible with Blue Gecko board.

S.NO	PART	Quantity	Link	Delivery Date
1	Photodetector Breakout - MAX30101	1	Link	04.18.2023 Shrinithi
2	Atmospheric Sensor Breakout - BME280	1	Link	04.18.2023 Vignesh
3	Blue Gecko board	2	Link	

Table2.1 Hardware components

Update:

From last week, we have made changes to our implementation of GATT server. Since we had to make changes to our server design, we have updates our project schedule and we're working on it now. So, overall, tasks for this week were designing the updated GATT server and GATT client prior to jumping on code implementation. We have started sensor integration for our GATT servers. We are pretty much on schedule. By this weekend, we should have our servers integrated with our sensor modules and moving forward we will implement the client.

2.2 Project Schedule:

Section Author: Vignesh

Task	Student(s)	Target Date	Expected date
Server design	Vignesh Vadivel	Apr 15, 2023	Apr 15, 2023
Server code Implementation Temperature sensor	Vignesh Vadivel	Apr 23,2023	Apr 23,2023
Server code Implementation Heart rate sensor	Srinithi Venkatesan	Apr 23, 2023	Apr 23, 2023
Client Implementation	Srinithi and Vignesh	Apr 24,2023	Apr 24,2023
Overall Integration	Vignesh and Srinithi	Apr 25, 2023	Apr 25, 2023
Complete testing	Vignesh and Srinithi	Apr 28, 2023	Apr 28, 2023

Table2.2 Proposed Schedule

2.3 Design changes:

2.3.1 Project Overview:

Section Author: Shrinithi

Discussing on the details of our module, the server device will implement a temperature sensor and a heart rate sensor to sense the temperature value and heart rate periodically. Only when the unusual temperature fluctuations are recorded, the server device will activate the heart rate sensor. The client acts as an alert device through which an alert in made to notify the user(doctor) in case the system records deviated fluctuation ranges from normal.

2.3.2 High Level Requirements:

Section Author: Vignesh

1. The system will consist of a BLE server and a BLE client, both running on a Blue Gecko Board.
2. The server will establish an encrypted link with the client via bonding.
3. The Blue Gecko board acting as the server will hold two sensors
   3.1 It periodically reads values from the external temperature sensor every second using LETIMER0.
   3.2 It periodically reads values from the heart rate sensor.
4. The server shall activate the temperature reading first and will read for this measurement periodically. Only when the temperature value exceeds threshold, the server shall activate the heart rate sensor for measurements.
5. The server shall implement and advertise two custom GATT services to record and send the temperature value and heart rate value.
6. The server shall display the temperature and heart rate value as it is being evaluated onto the server display.
7. The server shall send the temperature and heart rate measurement to the client only if the respective values cross the desired threshold.
8. The server code shall maintain the lowest possible energy mode for this application.
9. The client shall receive periodic display values from the server over the BLE radio.
10. Upon receiving the values, the client will process physical alerts through the LEDs.
    10.1 If the heart rate is normal and only the temperature is not within the threshold, LED0 on the client side shall be illuminated.
    10.2 If both the heart rate and temperature values are not within the threshold, LED0 and LED1 shall be illuminated on the client side.
11. Both the server and client shall incorporate an LCD display.
    11.1 The server and client shall display the corresponding naming and addresses.
    11.2 The server shall display the current connection status - Advertising/Connected/Bonded.
    11.3 The client shall display the current connection status - Discovering/ Connected/ Bonded/ Handling Indications.
    11.4 The server and client shall display the corresponding temperature and heart rate values read from the GATT database.

2.3.3 Design Overview:

Section Author: Shrinithi

The design consists of a pair of Blue Gecko Boards, one that hosts the server code and the other that hosts the client code. Once the bonding is successful between the server and client, the server will measure the temperature data updating through the **BME280** external sensor on a periodic basis every second. The timer used to perform periodic analysis is the LETIMER0. There will be corresponding GATT characteristics that will record the abnormal temperature in the GATT database and will send the data to the client**.** When abnormal temperature changes are observed, the server will enable the heart rate sensor, **MAX30102** to measure the heart beats per minute and the abnormal ranges are updates to the GATT database, which is fetched by the server to display. This data is passed to the client as well.

The client will receive indications over the BLE radio. An LCD display is used as a display module for both the server and client devices. Based on the range of the readings received from the client, it takes action on physical indications. If only the temperature measurements received were abnormal, an alert through lighting LED0 is performed by the client. In case, both the heartrate and temperature both don't fall under the normal range, the client illuminates both LED0 and LED1.

2.3.4 Wireless communication details:

Section Author: Vignesh

The GATT server implements and advertises two custom GATT services. The GATT server will perform health thermometer service and wearable pulse rate service and as it will periodically measure those values and stores abnormal readings on the GATT server database.

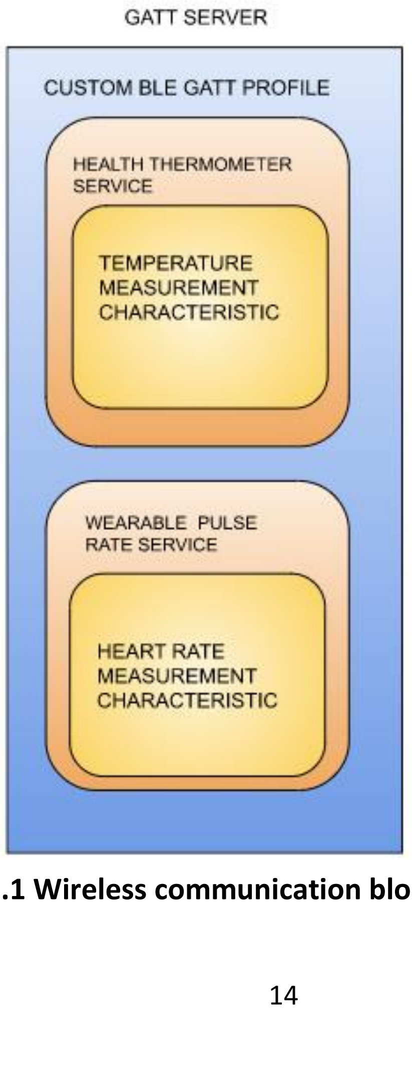

GATT SERVER
CUSTOM BLE GATT PROFILE
HEALTH THERMOMETER SERVICE
TEMPERATURE MEASUREMENT CHARACTERISTIC
WEARABLE PULSE RATE SERVICE
HEART RATE MEASUREMENT CHARACTERISTIC

Fig2.1 Wireless communication block diagram

2.3.5 Hardware Block diagram

Section Author: Shrinithi

The hardware block diagram of the two subsystems present namely: GATT server and GATT client. The temperature sensor, BME280 is integrated to the GATT server externally using I2C interface. The heart rate sensor, MAX30101 is interfaced to the GATT server subsystem also through I2C interface as well. The I2C bus on the Blue Gecko Board will help write and read to sensors. The LCDs found on both sides are part of the Blue Gecko Board. For alert and indications, apart from the LCD display, the onboard LEDs are enabled through GPIO interface to the client. The server and clients communicate through the BLE radio.

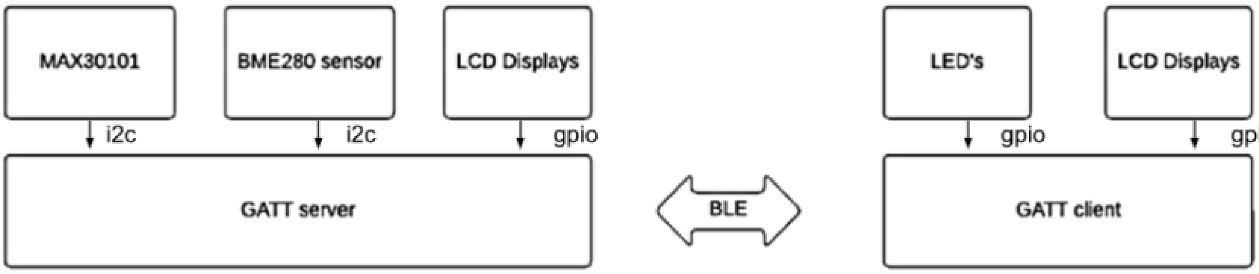


Fig2.2 Hardware block diagram

2.3.6 Software Block Diagram

Section Author: Vignesh

1. LETIMER0 would keep generating interrupts for every second, and temperature measurement would be taken in the server.
2. Once the temperature value is ready, we will check if it is in normal range. If not, we will load the value in the GATT database and we will send the temperature measurement value to display the abnormal temperature value in the LCD.
3. This will also activate the heart rate sensor value and write it to the GATT database. This will be displayed in the LCDs as well.
4. The client will take response to the values received. LED0 in the client would be turned ON (to indicate abnormal temperature measurement) and if the heart rate value is also not within the range, we will turn ON LED1.

GATT Server: Software Block Diagram:

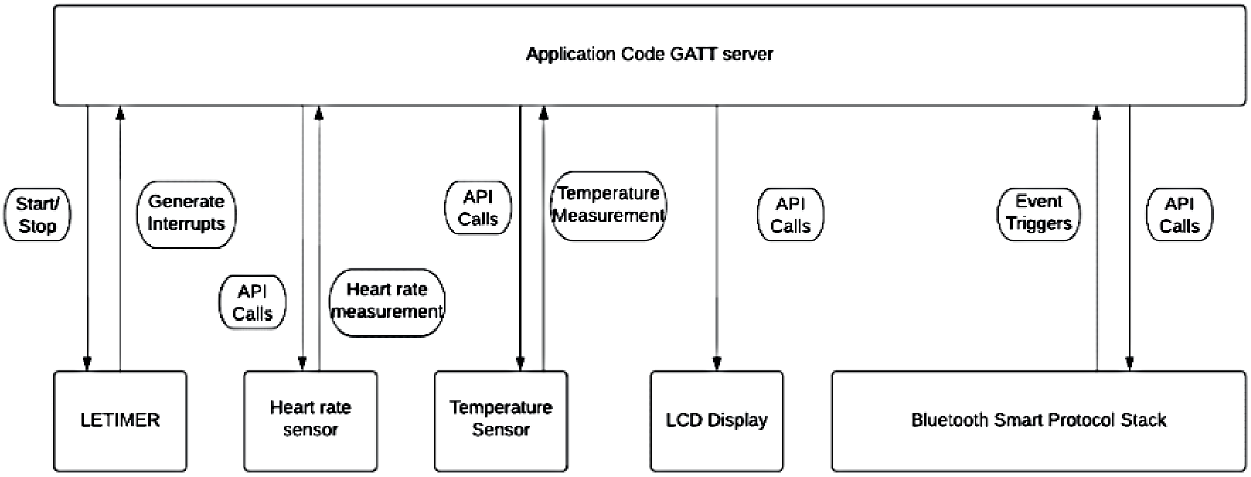


Fig2.3 GATT Server Software block diagram

GATT Client: Software Block Diagram:

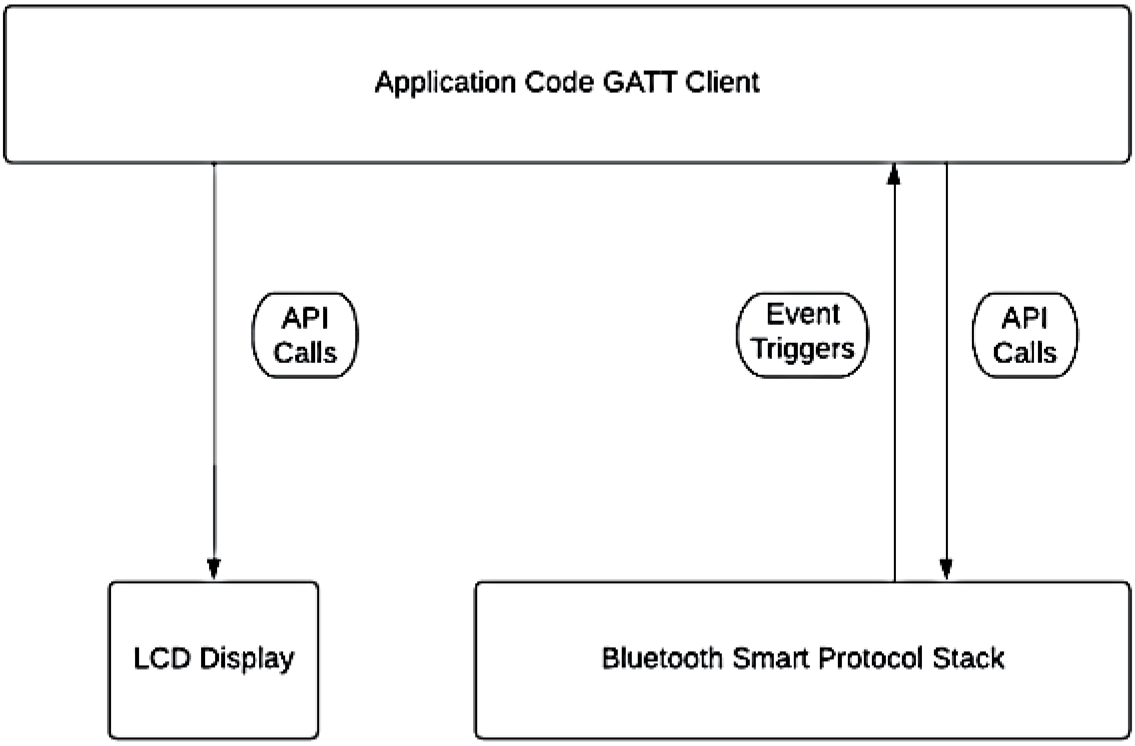


Fig2.4 GATT Client Software block diagram

Software Flowchart

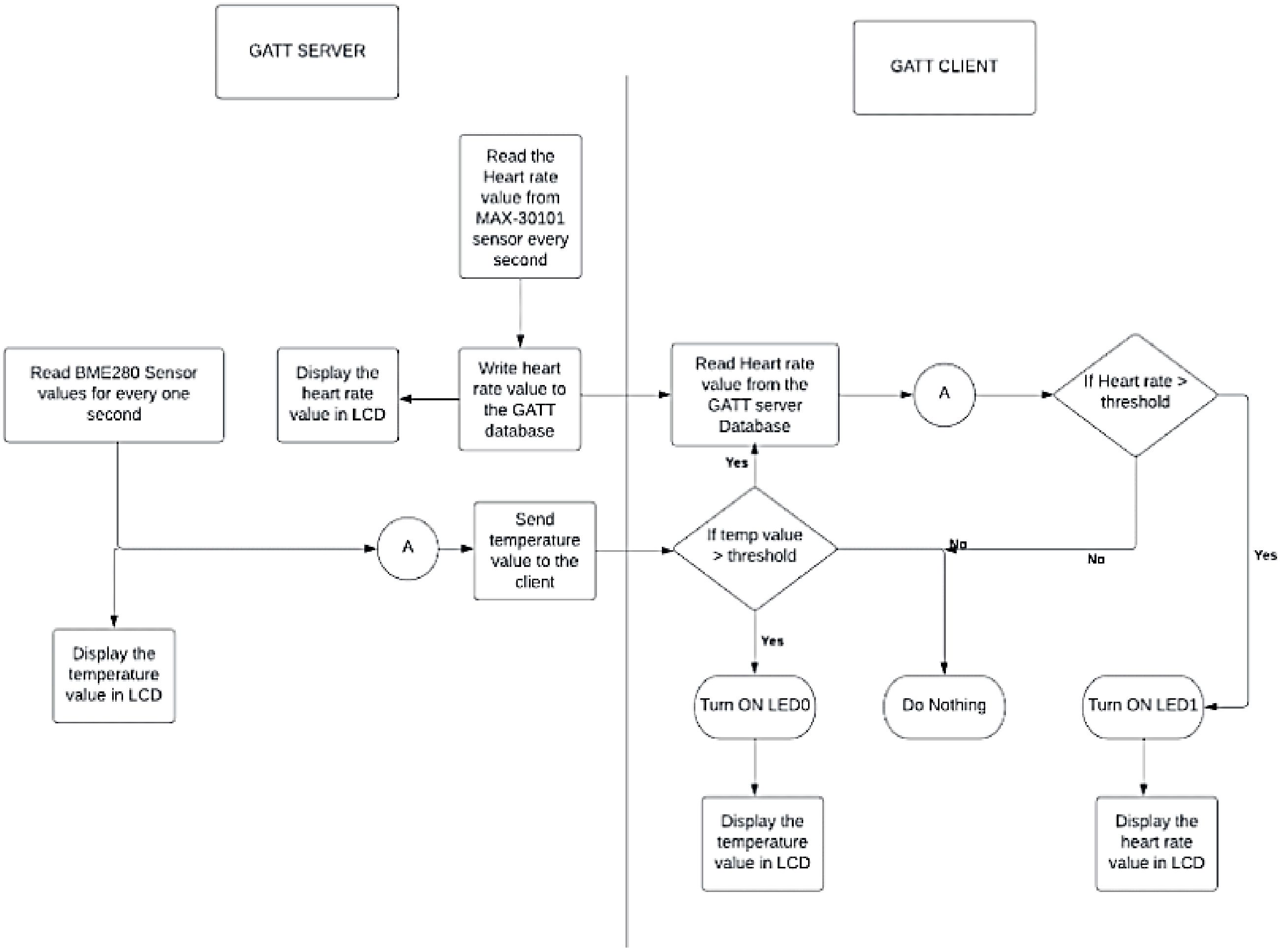

GATT SERVER
GATT CLIENT
Read the Heart rate value from MAX-30101 sensor every second
Read BME280 Sensor values for every one second
Display the heart rate value in LCD
Write heart rate value to the GATT database
Read Heart rate value from the GATT server Database
A
If Heart rate > threshold
Yes
A
Send temperature value to the client
If temp value > threshold
No
No
Yes
Display the temperature value in LCD
Yes
Turn ON LED0
Do Nothing
Turn ON LED1
Display the temperature value in LCD
Display the heart rate value in LCD

Fig2.5 Software flow block diagram

2.4 Challenges:

Section Author: Shrinithi

As we try to integrate the sensors to our clients, it's a new learning for us as the sensors we are dealing with are completely new. It's taking some time for us than expected to learn about the sensors in general and to get data from the datasheet. However, until now we have not been struck with anything. The challenge in the upcoming days might be integrating these two sensors into one server board and communicating with client to make alert display. Only as we move through the implementation we will know the further challenges.

2.5 Test plan:

Section Author: Shrinithi

As we're yet integrating our sensors, we have an overview of our test plan designed in our test plan excel sheet. The basic testing for our hardware functioning has been performed. We're yet to come up with an intensive and well-planned testing for our system, which we plan to execute after our basic sensor integration is done. An overview of the testing shall include manual test procedures. Testing a sensor module requires manual intervention as sensor's readings are dependent upon external conditions. Software testing shall include to check the proper functioning of code to update the sensor values to the GATT database and display accordingly.

The LEDs will be tested to glowing under subjective conditions according to the cases mentioned above.
An updated test plan will be provided with our next update.

2.6 Incorporate feedback from Project Proposal

Section Author: Shrinithi

In Section 1 Proposal, the GATT client implemented our heart rate sensor. Changes were implemented to it as our application needs and we updated the GATT server to hold both the temperature and heart rate sensor.

All the design requirements of the Section 1 that needed to update for this change was made and has been included in the Section 2 update.

The changes include in the fields:
Project Overview
High level requirements
Design overview
Hardware and Software block diagrams
Wireless communication details
Software Flow chart

Gitlink:
https://github.com/CU-ECEN-5823/ecen5823-courseproject-Vignesh-Vadivel.git

Section 3 - Update 2

3.1 Status:

Section Author: Shrinithi

Since last week, we have been working on individual sensor integrations and getting data from the sensors. Progress:

1. We have integrated and are reading data from the BME280 with gecko using custom BME280 API calls for silicon labs.
2. GATT server implemented for BME280.
3. We have integrated MAX30101 with the gecko but yet to get the sensor values.
4. We have configured the client part and integrated the LCD ready to receive sensor data from server.

3.2 Project Schedule:

Section Author: Vignesh

Task	Student(s)	Target Date	Expected date
Server design	Vignesh Vadivel	Apr 15, 2023	Apr 15, 2023
Server code Implementation Temperature sensor	Vignesh Vadivel	Apr 23,2023	Apr 24,2023
Server code Implementation Heart rate sensor	Srinithi Venkatesan	Apr 23, 2023	Apr 29, 2023
Client Implementation	Srinithi and Vignesh	Apr 24,2023	Apr 25,2023
Overall Integration	Vignesh and Srinithi	Apr 25, 2023	Apr 30, 2023
Complete testing	Vignesh and Srinithi	Apr 28, 2023	May 1, 2023

Table3.1 Proposed Schedule

3.3 Update:

Work for this week:

Section Author: Shrinithi

Plan for this week:

1. Get data from the MAX30101 sensor.
2. Create custom GATT services for heart rate sensor and store values in GATT database to be transmitted to the client LCD display.
3. Integrate the server and client devices to update readings on client LCD display.

3.4 New design implementation:

Section Author: Vignesh

From the connection diagram below, we have integrated the 3v3(pin 20) power supply of gecko to supply of BME280, SDA and SCL pins of gecko (pins 16, 15 respectively) to the SDA and SCL of BME280, Ground (pin1) to the gnd of BME280.

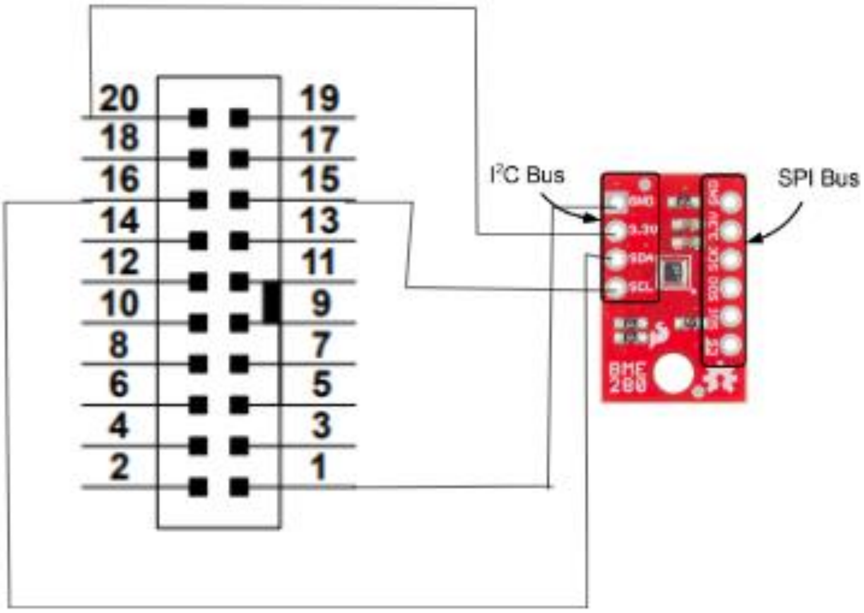


Fig3.1 Connection for BME280

From the connection diagram below, we have integrated the 3v3(pin 20) power supply of gecko to supply of MAX30101, SDA and SCL pins of gecko (pins 16, 15 respectively) to the SDA and SCL of MAX30101, Ground (pin1) to the ground of MAX30101.

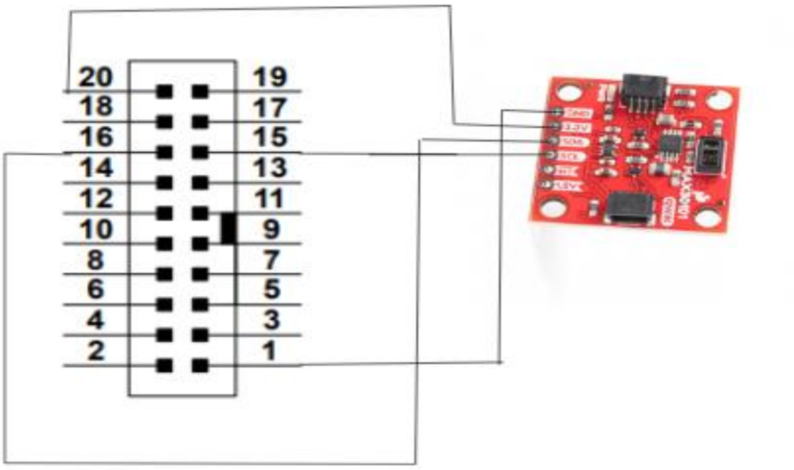


Fig3.2 Connection for MAX30101

3.5 Challenges:

Section Author: Shrinithi

Initially we were trying to write our own I2C interface from the data sheet, but didn't get sensor values as expected. Later we included the silicon labs API calls for BME280 and were able to receive data correctly. So initial process of trying to configure the sensor took us longer time than expected and we wasted a long time on it. Further there are not much resources for MAX31010 and hence we're trying to configure it.

3.6 Incorporate feedback from Update 1

Section Author: Shrinithi

In Section 2 Update, we had missed to update the wiki link properly. So, we have updated the link correctly through this update.
Otherwise, we are a little delayed from our expected timeline as posted in our challenges. We are trying to get data from the Heart rate sensor and integrate our whole system soon by Sunday and start our final leg of testing on Monday.

Gitlink:
https://github.com/CU-ECEN-5823/ecen5823-courseproject-Vignesh-Vadivel.git

Section 4 - Final Report

4.1 Status:

4.1.1 Final Updated schedule:

Section Author: Shrinithi

For the last week, we were working on getting the MAX30101 interfacing, but due to complications and running out of time, we switched our sensor to GPIO based Pulse sensor SEN 11574. We were able to get it working on time.
For this week, tasks were,

1. Integrate SEN 11574 and BME270 in one server board and integrate the LETIMER0 interrupts to update the server GATT database every one second.
2. Update client to fetch data from the server database and update the LCD display if data's cross threshold.
3. Configure alerts on the client board.
4. Though we faced a challenge in the last stage of the project, we adhered to the timeline with an anticipated completion date. We had our project functional and performed testing a well.

Following is the updated and the final schedule which represents project planning and execution timeline throughout the project.

Task	Student(s)	Target Date	Expected date	Status
Server design	Vignesh Vadivel	Apr 15, 2023	Apr 15, 2023	Completed
Server code Implementation Temperature sensor	Vignesh Vadivel	Apr 23,2023	Apr 24,2023	Completed
Server code Implementation Heart rate sensor	Srinithi Venkatesan	Apr 23, 2023	Apr 29, 2023	Completed
Client Implementation	Srinithi and Vignesh	Apr 24,2023	Apr 25,2023	Completed
Overall Integration	Vignesh and Srinithi	Apr 25, 2023	Apr 30, 2023	Completed
Complete testing	Vignesh and Srinithi	Apr 28, 2023	May 1, 2023	Completed

Table4.1 Proposed Schedule

4.2 Design changes from Update 2:

4.2.1 Hardware Block diagram

Section Author: Vignesh

The hardware block diagram of the two subsystems present namely: GATT server and GATT client. The temperature sensor, BME280 is integrated to the GATT server externally using I2C interface. The heart rate sensor, SEN 11574 is interfaced to the GATT server subsystem also through GPIO interface. The I2C bus on the Blue Gecko Board will help write and read to sensors. The LCDs found on both sides are part of the

Blue Gecko Board. For alert and indications, apart from the LCD display, the onboard LEDs are enabled through GPIO interface to the client. The server and clients communicate through the BLE radio.

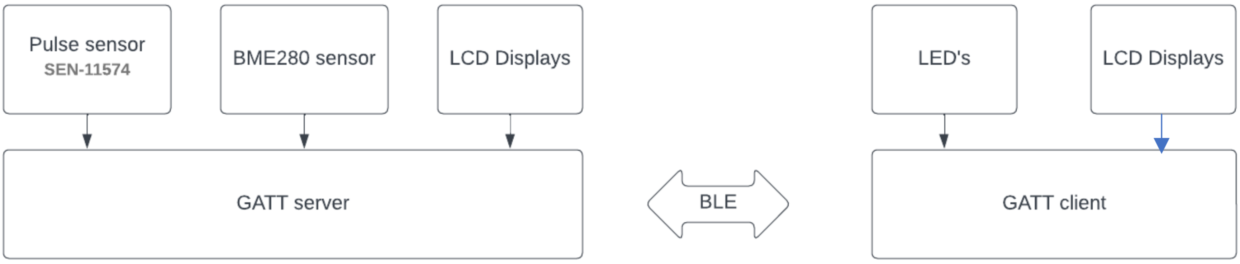


Fig 4.1 Hardware Blockdiagram

4.2.2 Software Block Diagram

Section Author: Vignesh

1. LETIMER0 would keep generating interrupts for every second, and temperature and heart rate measurement would be taken in the server.
2. Once the readings are measured, its written to the GATT server database every second. The readings are updated on the GATT server LCD.
3. The client will read the sensor values from the GATT server database and analyse the values for threshold range. If the values cross the threshold, the client will display the pay on the LCD and LED0 in the client would be turned ON (to indicate abnormal temperature measurement) and if the heart rate value is also not within the range, we will turn ON LED1.

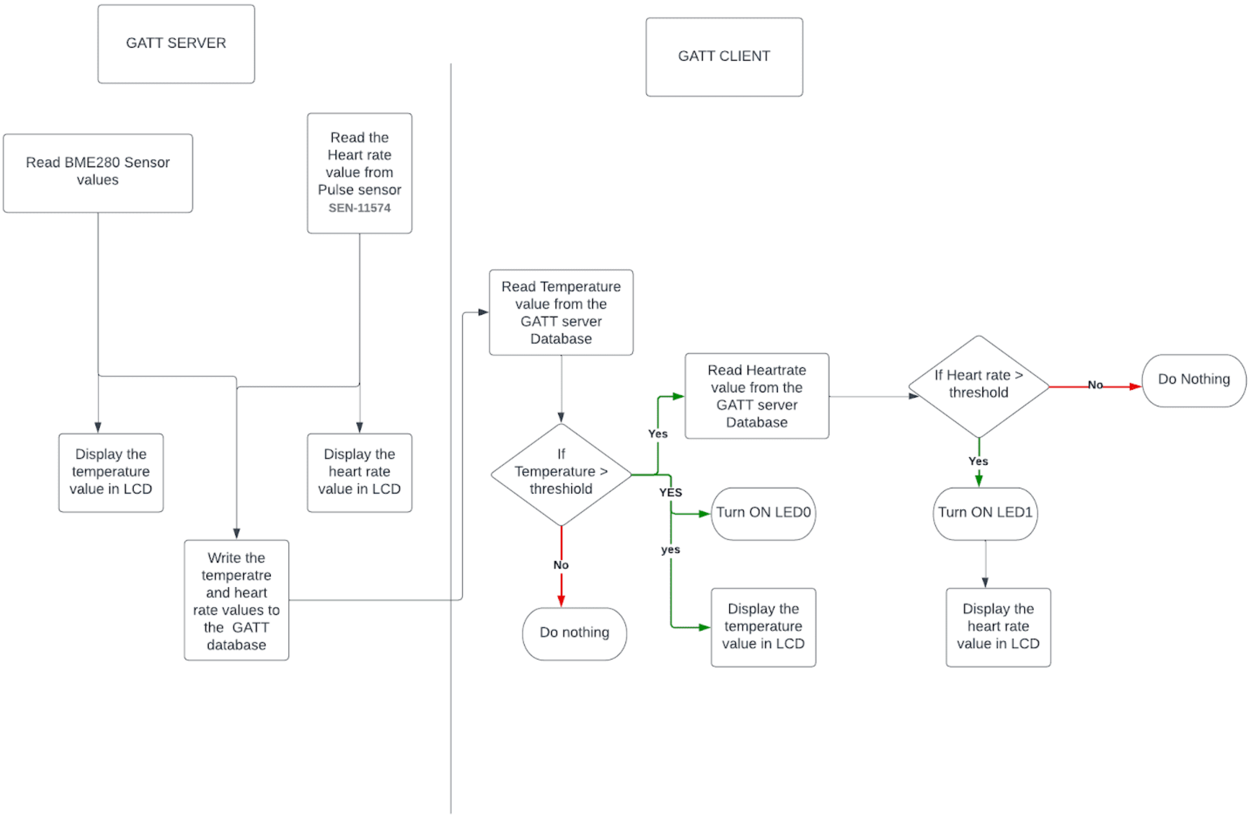

GATT SERVER
Read BME280 Sensor values
Read the Heart rate value from Pulse sensor SEN-11574
Display the temperature value in LCD
Display the heart rate value in LCD
Write the temperatre and heart rate values to the GATT database
GATT CLIENT
Read Temperature value from the GATT server Database
If Temperature > threshold
No
Do nothing
Yes
YES
yes
Read Heartrate value from the GATT server Database
Turn ON LED0
Display the temperature value in LCD
If Heart rate > threshold
No
Do Nothing
Yes
Turn ON LED1
Display the heart rate value in LCD

Fig 4.2 Software Blockdiagram

4.2.3 State Diagram:

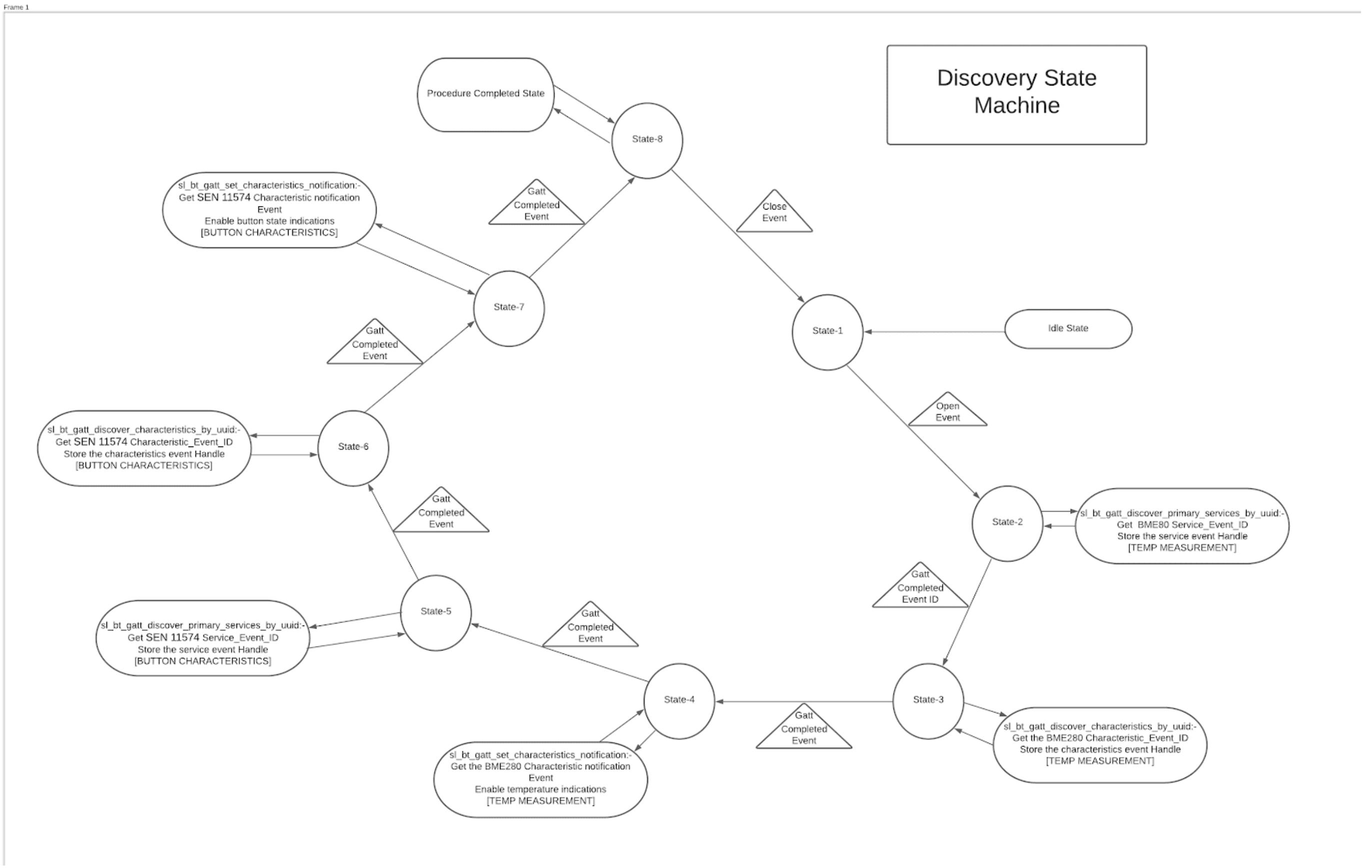

Frame 1
Discovery State Machine
Idle State
State-1
Open Event
State-2
sl_bt_gatt_discover_primary_services_by_uuid:- Get BME80 Service_Event_ID Store the service event Handle [TEMP MEASUREMENT]
Gatt Completed Event ID
State-3
sl_bt_gatt_discover_characteristics_by_uuid:- Get the BME280 Characteristic_Event_ID Store the characteristics event Handle [TEMP MEASUREMENT]
Gatt Completed Event
State-4
sl_bt_gatt_set_characteristics_notification:- Get the BME280 Characteristic notification Event Enable temperature indications [TEMP MEASUREMENT]
Gatt Completed Event
State-5
sl_bt_gatt_discover_primary_services_by_uuid:- Get SEN 11574 Service_Event_ID Store the service event Handle [BUTTON CHARACTERISTICS]
Gatt Completed Event
State-6
sl_bt_gatt_discover_characteristics_by_uuid:- Get SEN 11574 Characteristic_Event_ID Store the characteristics event Handle [BUTTON CHARACTERISTICS]
Gatt Completed Event
State-7
sl_bt_gatt_set_characteristics_notification:- Get SEN 11574 Characteristic notification Event Enable button state indications [BUTTON CHARACTERISTICS]
Gatt Completed Event
State-8
Procedure Completed State
Close Event

Fig 4.3 State diagram

4.3 Requirements completion:

Section Author: Shrinithi

We had proposed the following requirements at the time of the project proposal. All the requirements are met.

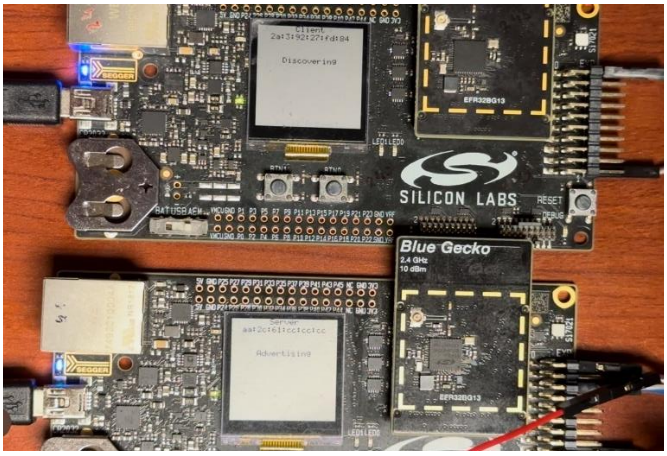
Number	Requirements	Status
1.	The system will consist of a BLE server and a BLE client, both running on a Blue Gecko Board.	Completed
2.	The server will establish an encrypted link with the client via bonding.	Completed
3.	The Blue Gecko board acting as the server will hold two sensors - It periodically reads values from the external temperature sensor every second using LETIMER0. - It periodically reads values from the heart rate sensor.	Completed
4.	The server shall activate the temperature and heart rate reading measurement periodically.	Completed

5.	The server shall implement and advertise two custom GATT services to record and send the temperature value and heart rate value.	Completed
6.	The server shall display the temperature and heart rate value as it is being evaluated onto the server display.	Completed
7.	The client shall update the temperature and heart rate measurement from the server only if the respective values cross the desired threshold.	Completed
8.	The server code shall maintain the lowest possible energy mode for this application.	Completed
9.	The client shall receive periodic values from the server over the BLE radio.	Completed
10.	Upon receiving the values, the client will process physical alerts through the LEDs. - If the heart rate is normal and only the temperature is not within the threshold, LED0 on the client side shall be illuminated. - If both the heart rate and temperature values are not within the threshold, LED0 and LED1 shall be illuminated on the client side.	Completed
11.	Both the server and client shall incorporate an LCD display. - The server and client shall display the corresponding naming and addresses. - The server shall display the current connection status - Advertising/Connected/Bonded. - The client shall display the current connection status - Discovering/ Connected/ Bonded/ Handling Indications. - The server and client shall display the corresponding temperature and heart rate values read from the GATT database.	Completed

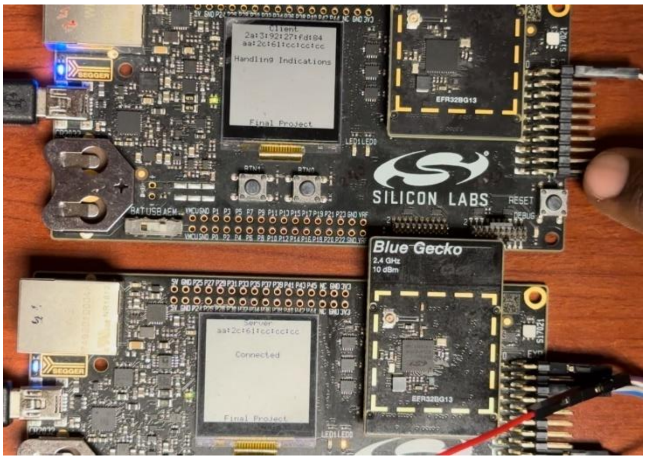
Table 4.2 Requirements status

4.4 Working (Bonding/sending indication/alert notification):

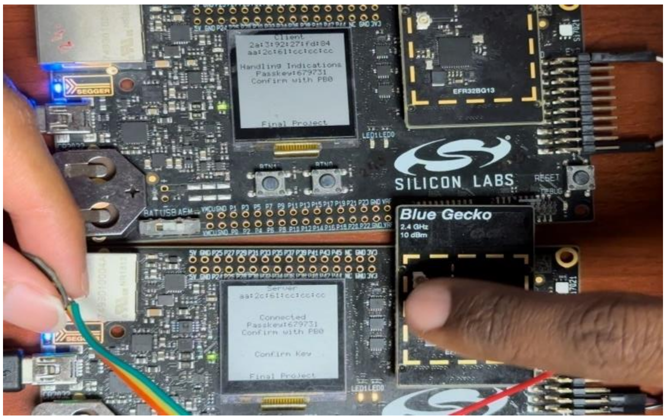
Section Author: Vignesh



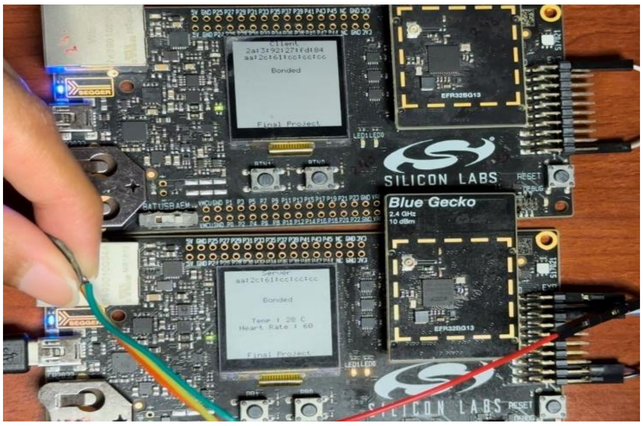
Discovery/Advertising



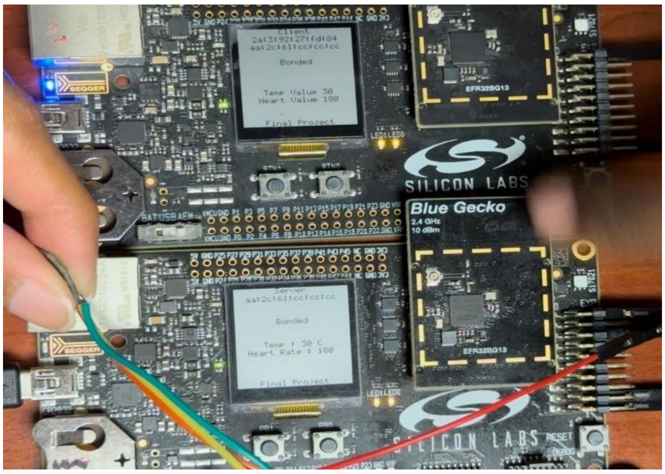
Connection status



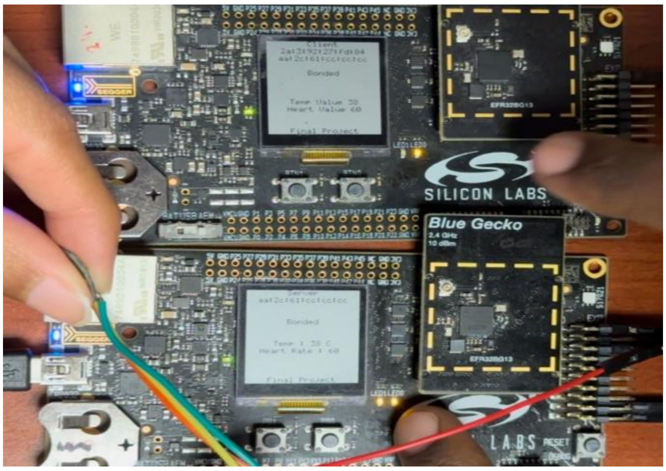
Initiating Bonding/Pairing



Bonded status



Alert when both temperature and heart rate exceeds(LED0 and LED1)



Alert when both temperature exceeds threshold (LED0)

4.5 Test Plan Section:

Section Author: Shrinithi

We completed 100% of the tests by this sprint. Attached test plan Excel Sheet shows the tests, their statues and description.

4.6 Distribution of Work:

Section Author: Shrinithi

The requirements and proper planning of tasks as posted on the test plan and wiki page acted as good guidelines and gave us a measure of work done and work to be done at every sprint. We both were equally involved in debugging the issues and handling challenges we faced during the project tenure. When we tried to integrate everything together, we worked together to sort issues and so our work contributions were equal perfectly throughout the submissions. The test plan clearly explains what work we did and its self-explainable for our contributions.

Name	Percent Contribution
Shrinithi	50%
Vignesh	50%

Table 4.3 Distribution of work

ble.c/.h	Vignesh and Shrinithi
gpio.c/.h	Vignesh
I2c.c/.h	Vignesh and Shrinithi
irq.c/.h	Vignesh and Shrinithi
oscillators.c/.h	Shrinithi
timers.c/.h	Vignesh
bme280.c	Silicon labs repo

Table 4.4 Files of project

4.7 Outcomes:

Section Author: Shrinithi and Vignesh

Shrinithi:

According to me, I felt that we have deviated from our idea from our first proposal and the reason for that would be estimating the project very lightly. We under estimated the time that would be required for configuring our sensors and as we worked practically, we understood that some things are very challenging to be configured. So, we had to change our ideas as we started implementing our project step by step.

The incremental work my giving our updates every week was very efficient way of working in my opinion. If we hadn't followed this approach, in the last minute we wouldn't have known our shortcomings earlier and deadlines was very helpful in the successful completion of project. I learnt the importance of unit testing and designing. Testing before integrating the codes together helped us while debugging. Designing before coding is the major lesson that I have learnt. Interfacing two sensors individually was quite easier as compared to integrating them together.

Vignesh:

The biggest challenge to me was when we tried to integrate both the sensors onto a single board, we realised it was difficult to configure the LETIMER to be utilized by both sensors together. So, we had to spend some time figuring and changing the functioning of our system as mentioned in the above software flow block diagram. Secondly, working incrementally helped us stick to deadlines and that was really helpful in finishing our project on time. Similarly, switching to GPIO based heart rate sensor helped us achieve our system working in the last minute and this made us realize the importance of having back up plans. One major learning for me was to stick to deadlines and to figure out the system more in detail before execution. If we had known about the sensor complications earlier, we would have deployed a different application, but again, thanks to the back-up plan and incremental work flow.

4.7 Energy modes:

We used the Silicon Lab's Energy Profiler to characterize the low power performance of the GATT server subsystem. When the two sensors (BME280 and SEN11574) are activated, the server device does not disconnect unless there is a connection failure but it does sleep in the lowest energy mode possible - EM2. Reason: Since Bluetooth doesn't work on EM3, we have fully implemented EM2 throughout our application.

This figure shows one complete cycle for both the sensors together and the observed current is 1.66mA, which is the lowest possible current in EM2.

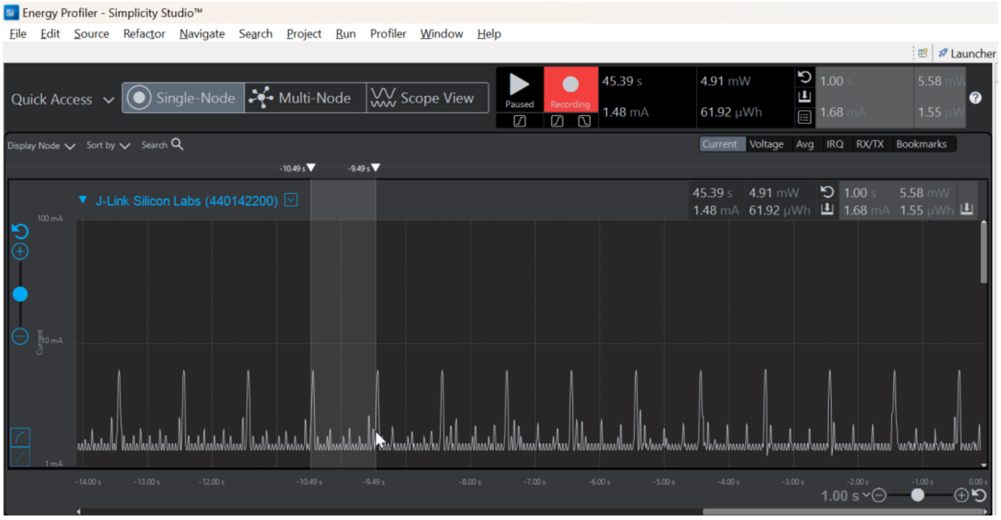


Fig4.4 EM2 mode

For the purpose of comparison, we ran our application in EM0 and we observed the average current for one complete cycle for both the sensors were around 6.18mA as seen in the figure below. So this proves our validation for utilizing EM2 and the above figure shows the efficiency of energy savings.

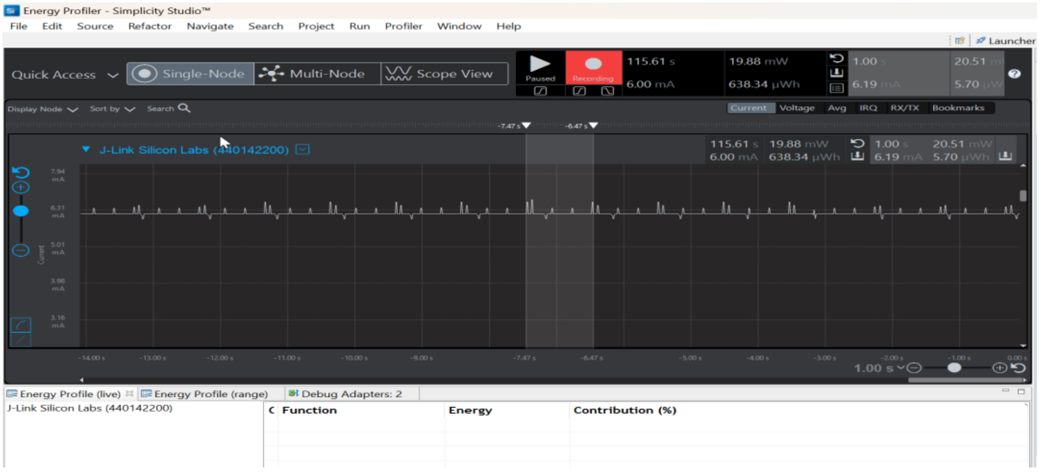


Fig4.5 EM0 mode

Gitlink:
https://github.com/CU-ECEN-5823/ecen5823-courseproject-Vignesh-Vadivel.git