**GESTURE CONTROLLED DISPLAY**

Pradyumna Gudluru

Dhruv Mehta

Course Project Proposal

Fall 2021 ECEN 5823

November 9, 2021

Contents

[1. Section 1 - Project Proposal 3](#_Toc89451822)

[1.1 Student Names 3](#_Toc89451823)

[1.2 Project Overview 3](#_Toc89451824)

[1.3 High Level Requirements 4](#_Toc89451825)

[1.3.1 **Stretch Goals** 5](#_Toc89451826)

[1.4 High Level Design 6](#_Toc89451827)

[1.4.1 Overview 6](#_Toc89451828)

[1.4.2 Wireless Communication - GATT Services and Characteristics 7](#_Toc89451829)

[1.4.3 Client-Server Interfaces 8](#_Toc89451830)

[1.4.4 Functional Hardware Block Diagram 8](#_Toc89451831)

[1.4.5 Functional Software Flow Chart 9](#_Toc89451832)

[1.4.6 Data Flow Diagram 11](#_Toc89451833)

[1.4.7 Division of Labor 12](#_Toc89451834)

[1.4.8 Command Direction Functionality Table 12](#_Toc89451835)

[1.4.9 Design of LCD Displays 13](#_Toc89451836)

[1.5 Subsystem Summary 14](#_Toc89451837)

[1.6 Test Plan 15](#_Toc89451838)

[1.7 Proposed Schedule 16](#_Toc89451839)

[Section 2 - Update 1 17](#_Toc89451840)

[**2.1** **Status** 17](#_Toc89451841)

[**2.2** **Design Changes & Updates** 17](#_Toc89451842)

[**2.3** **Challenges and Issues Faced** 20](#_Toc89451843)

[**2.4** **Schedule Table** 21](#_Toc89451844)

[**2.5** **Test Plan Update** 22](#_Toc89451845)

[Section 3 - Update 2 23](#_Toc89451846)

[**3.1 Status** 23](#_Toc89451847)

[**3.2 Design Changes & Updates** 23](#_Toc89451848)

[**3.3** **Challenges and Issues Faced** 27](#_Toc89451849)

[**3.4** **Schedule Table** 28](#_Toc89451850)

[**3.5** **Test Plan Update** 29](#_Toc89451851)

[Section 4 - Final Report 30](#_Toc89451852)

[Status 30](#_Toc89451853)

[Distribution of Work 30](#_Toc89451854)

[What Was Learned 30](#_Toc89451855)

## Section 1 - Project Proposal

## Student Names

Pradyumna Gudluru – [pradyumna.gudluru@colorado.edu](mailto:pradyumna.gudluru@colorado.edu)

Dhruv Mehta – [dhruv.mehta@colorado.edu](mailto:dhruv.mehta@colorado.edu)

## Project Overview

Section Author: Dhruv Mehta

For the Gesture controlled display, we have chosen to implement Option - 2. This will consist of a Gecko Client consisting of the on-board LCD module to be used as the display and a Gecko Server which will consist of a gesture sensor and an ambient light sensor. The aim of this project is to develop a prototype of a display which is controlled remotely using gestures only. The application of this project would be in areas such as malls, where a user can pan and scroll through the display to discover shops or services, or even in an automotive display where a driver or passenger can scroll through the display without touching it. It can also be used in applications such as an e-reader where the user can flip the pages of a book without touch the display.

For the purpose of this project, the Client will simulate an e-book which contains a block of data which can be read either by scrolling up and down on the display or by gesturing left and right to flip through the pages.

The Server will receive the gesture inputs from the user through an I2C gesture sensor and indicate this data to the Client. The client will then scroll the display in the direction in which the user requested. An analog ambient light sensor will be used to detect the lighting conditions and put the Server into a low power state when it is dark, simulating a scenario where the devices are not used.

Two new GATT Services will be created for each of the sensors. The I2C gesture sensor data will be sent to the client as soon as it is received such that the device appears responsive. Similarly, the ambient light sensor will be interfaced to the ACMP module and set to trigger the server when the light falls or rises above a set threshold. The Client will act on the inputs received and update the display accordingly.

## High Level Requirements

Section Author: Dhruv Mehta & Pradyumna Gudluru

* The system shall consist of a BLE GATT Server and BLE GATT Client implementation, both using custom functionality for the project.
* The system shall implement two new GATT Services – one for gesture sensor updates and one for ambient light updates.
* The Gesture sensor service shall have a single characteristic to update the direction of motion to the client.
* The Ambient light sensor service shall have a single characteristic to update the light intensity to the client.
* The system shall use the LCD for the devices in the following ways:  
  Server: Display connection status, Gesture direction, luminosity.  
  Client: Display a block of text for the user to read.
* The server shall interface with the gesture sensor using I2C protocol.
* The server shall use interrupt driven I2C communication to receive data from the gesture sensor when a motion is made by the user.
* The server shall interface with the ambient light sensor using the ACMP module.
* The server shall maintain a low energy state when the luminosity falls below a pre-set threshold.
* The client shall establish an encrypted connection with the server on bootup and access both services.
* The client shall display a page of data to the user which can be scrolled up or down by gesturing up or down at the server respectively.
* The client shall display a page of data to the user which can be moved to the next or previous page by gesturing right or left at the server respectively.
* The client shall turn off the display when the server indicates that the ambient light is low.

### **Stretch Goals**

* The client shall display a user menu on bootup which consists of different items to view/read, which can be scrolled through by using the server pushbuttons.
* The server shall implement pushbuttons which are used to scroll through the client menu and select an item to view/read.
* Both the client and server shall power down to a low energy state when the server indicates that the ambient light is low.
  + 1. **Non-Implemented Requirements**
* The server shall operate as a battery powered device.
* A bigger LCD display shall be used for readability and graphics support.

## High Level Design

## Overview

Section Author: Pradyumna Gudluru

The design consists of two gecko boards, one used as a GATT Client (named as PAGE block) and the other board used as GATT server (named as USER block). In this project, the USER senses both ambient light from the Ambient Light Sensor and gesture features from Gesture sensor. At the USER gecko, the direction is computed based on the inputs from the gesture sensor. The directions are then transferred over the radio as indications to the PAGE block. The ambient light value is also transferred periodically as a second characteristic to the PAGE board.

As the PAGE board receives the characteristics, it computes or the minimum light for the client and server to run on low light mode. In the low light mode, the display of LCD is turned OFF on both the boards and gesture sensor goes to sleep. When there is sufficient light available in the surroundings, the ambient value sensor reports the data and the system switches ON to the regular mode of operation. When there is a change in the directionality, as per the user, the data on page moves up or down or goes to previous or to the next page.

**Description of Data Types**

Section Author: Dhruv Mehta

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Measurement** | **Units** | **Data Type** | **Valid/Allowed Values**  **(Range)** | **Update Rate** |
| Gesture Direction | Direction | uint8\_t | 0 – 5 (Left, Right, Up, Down, Near, Far) | 5 Hz (4 bytes/sec) |
| Ambient Light | Illuminance | uint16\_t | 0 – 4095 | 1 Hz (2 bytes/sec) |

**Table 1.1**

The table above shows the two sensor measurements and their intended data types. The gesture direction is received by the server and can be processed and passed as part of an enum (indication each direction) to the client. Since only 4 directions and two movements are detectable, a uint8\_t variable is sufficient for this purpose. As of now, the data rate is decided to be 5Hz (resulting in a 4 bytes/sec transfer rate) but this can be changed based on the responsiveness of the client.

The ambient light sensor is interfaced to an ADC or an ACMP module. In the case of an ADC, the Gecko Board has an on-board 12-bit ADC for which a uint16\_t variable is sufficient (resulting in a 2 bytes/sec transfer rate).

## Wireless Communication - GATT Services and Characteristics

Section Author: Dhruv Mehta

Graphical user interface

Description automatically generated

**Figure 1.1**

As shown above and explained before, two new custom Services will be implemented in the GATT Profile.  
These are – Gesture Sensor Service and Ambient Light Service. Both services contain one characteristic each. The Gesture Sensor Service contains the Sensor Direction Characteristic and the Ambient Light Service contains the Ambient Light Characteristic. On bootup, the Client connects to the Server and enables indications for both services on the Server. The server initializes the sensors and waits for an event-based input such as a user waving his/her hand or the light sensor falling/rising above a threshold. These indications are sent to the Client for processing.

## Client-Server Interfaces

Section Author: Dhruv Mehta

Graphical user interface, application

Description automatically generated

**Figure 1.2**

The diagram above shows the interfaces between the Client-Server and the sensor subsystems. Both the sensors are interfaced to the Server. The interfaces are shown as I2C and ADC/ACMP respectively. These sensor inputs are processed after being received and converted into meaningful data before an indication is sent to the client. Both sensors have their own sensor service and a characteristic for the value. The Client enables indications for these services during boot-up. Therefore, the interface between the Client and the Server is the BLE Radio. On reception of an indication, the Client sends a confirmation and then processes the data. It first checks the ambient light sensor data against a pre-set threshold and confirms that there is sufficient light to proceed. Else, the display is turned off and the Client is set to a lower power mode. If the light is sufficient, the gesture sensor input is processed and the appropriate action is taken on the display.

For example, if the light is sufficient and a left swipe is received, the display will flip to the next page in the book. Each gesture and its corresponding actions are explained in the sections below.

## Functional Hardware Block Diagram

Section Author: Pradyumna Gudluru

Graphical user interface

Description automatically generated

## Functional Software Flow Chart

Section Author: Pradyumna Gudluru

**GATT Server**

Graphical user interface, application

Description automatically generated

**Figure 1.4**

On the PAGE(client) as soon as the board switches ON, it starts discovering for Bluetooth connections. Once connected, the PAGE receives characteristics from the USER(server) board. The PAGE starts computing for the low light mode operation. If the value is greater than the threshold, it considers the second characteristic of direction update based on the input from the gesture sensor. This value updates the data to print on the LCD display as per the command direction functionality.

**GATT Client**

Chart, radar chart

Description automatically generated

**Figure 1.5**

On the USER(server) as soon as the board is switched ON, it starts advertising and once it gets connected, the sensors will be enabled for the data transfer. The characteristic indications are sent to the PAGE board periodically, in regular mode of operation. If the value from the ambient light sensor is less than a particular threshold, the server goes to low lighting mode and the display is set to OFF as per the LCD display functionality. The direction based on the gesture input is computed in the server and then transferred through a radio to the connected PAGE board, periodically.

## Data Flow Diagram

Section Author: Dhruv Mehta

Diagram

Description automatically generated

**Figure 1.6**

The sequence diagram shown above shows the flow of data between the Sensors subsystem, the Server subsystem and the Client subsystem. After the client has initiated and established a connection, the server periodically checks on the sensors and indicates the received value to the client. Since the ambient light sensor is a simple analog sensor, it can be easily turned on and off to conserve energy. As for the gesture sensor, it has a sleep mode in-built and so is always in that state unless some registers are set and communication takes place. The periodicity of both sensors are shown in the data flow diagram and may be adjusted based on testing, responsiveness or power consumption.

## Division of Labor

Section Author: Dhruv Mehta

**Hardware**: The hardware consists of two Gecko Boards serving as a Client and a Server. Additionally, an I2C gesture sensor and an analog ambient light sensor is also used. There is no complication in hardware other than interfacing it to the board.

**Software**: The software consists of the Server, Client and Sensor subsystems. All of these will require a custom implementation as both sensors are new and are linked to custom services for communication. The client functionality is also new, depending on the sensor values to take an action on the display.

## Command Direction Functionality Table

Section Author: Pradyumna Gudluru

The data from gesture sensor is obtained and computed based on the following functionality table. When the user swipes or waves LEFT, the PAGE TURN OVER command is enabled with printing the next page on the PAGE LCD display. Similarly, when the user swipes or waves RIGHT/ UP/ DOWN, the commands PAGE TURN PREVIOUS/ LINES UP/ LINES DOWN are enabled with printing the previous page/ scroll up/ scroll down on the PAGE LCD display, respectively.

Table

Description automatically generated

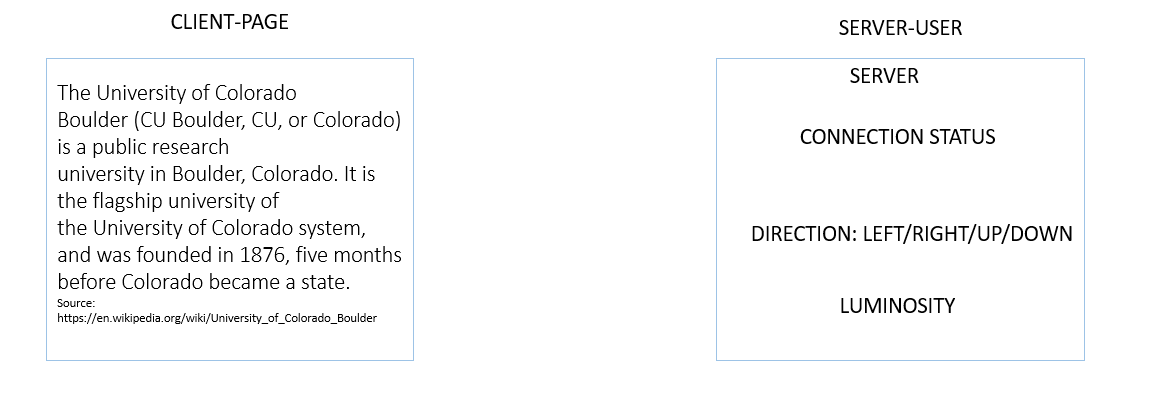
**Table 1.2**

## Design of LCD Displays

Section Author: Pradyumna Gudluru

According to the project requirements defined, on the PAGE(client) board, the data is being displayed as of some information. When the reader has to read the data on next page, a gesture direction to swipe right will make the defined directionality changes on the PAGE LCD display. Similarly, for different functionalities, the data gets updated and is printed on the LCD display according to the user’s perspective.

The LCD display on the USER(server) board prints the basic info required for the user like, the connection status, the luminosity and the direction being updated. This display also shows the info or error messages. Before going into low light mode it displays the transfer information for a particular delay and then the LCD switches OFF. Similarly, in low light mode on the PAGE board, before the display switches OFF, it prints transfer information for a particular delay.



**Figure 1.7**

Table

Description automatically generated

**Figure 1.8**

## Subsystem Summary

Section Author: Dhruv Mehta

The following are the subsystems in this project:  
  
**GATT Client** – Device which connects to the server and receives sensor data from two services to update a display.   
  
**GATT Server** – Device which interfaces with a gesture sensor and an ambient light sensor to send indications to the client based on the processed sensor data.

**Sensor Monitoring Subsystem** – Consists of the gesture sensor and ambient light sensor. Combination of hardware and software interfaces to the sensor to acquire and process raw data for the server.

## Test Plan

Section Author: Dhruv Mehta & Pradyumna Gudluru

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Plan Table | | | | |
| **Test Number** | **Test Description** | **Date last run on** | **Test Result (Pass/Fail)** | **Notes** |
| 1 | A basic test the checks that the ambient sensor is returning measurements | 11/12/2021 | To Do |  |
| 2 | A basic test the checks that the gesture sensor is returning measurements | 11/15/2021 | To Do |  |
| 3 | A test to check whether the characteristic indications are sent to the client periodically | 11/20/2021 | To Do |  |
| 4 | A test to check whether the COMMAND is updated on the client DISPLAY | 11/23/2021 | To Do |  |
| 5 | A test to check LUMINOSITY value updated periodically on server DISPLAY | 11/23/2021 | To Do |  |
| 6 | A test to check whether DIRECTION received from gesture sensor is updated on the server DISPLAY | 11/23/2021 | To Do |  |
| 7 | A test to check if ERROR/ INFO is printed on LCD on client and server DISPLAY | 11/23/2021 | To Do |  |
| 8 | A test to check the IDLE state of the device | 11/28/2021 | To Do |  |
| 9 | Low Power Management: To check the energy profiler in low power mode | 11/29/2021 | To Do |  |
| 10 | A test to indicate the LED 0 on server(USER) when in Low Power Mode | 11/29/2021 | To Do |  |
| 11 | A test to indicate the LED 0 on client(PAGE) when in Low Power Mode | 11/29/2021 | To Do |  |
| 12 | Full system test – The proposed functionality is working, tested with all possible gestures and also low-light testing. | 12/6/2021 | To Do |  |

**Table 1.2**

## Proposed Schedule

Section Author: Dhruv Mehta & Pradyumna Gudluru

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Student(s) Responsible** | **Target Completion Date** | **Expected Completion Date** |
| Gesture Sensor Subsystem Design | Pradyumna Gudluru | Nov 16 |  |
| Gesture Sensor Code Implementation | Pradyumna Gudluru | Nov 18 |  |
| Ambient Sensor Subsystem Design | Dhruv Mehta | Nov 12 |  |
| Ambient Sensor Code Implementation | Dhruv Mehta | Nov 18 |  |
| GATT Server Subsystem Design | Pradyumna Gudluru | Nov 20 |  |
| GATT Server Code Implementation | Pradyumna Gudluru | Nov 23 |  |
| GATT Client Subsystem Design | Dhruv Mehta | Nov 22 |  |
| GATT Client Code Implementation | Dhruv Mehta | Nov 24 |  |
| System Integration and Testing | Dhruv & Pradyumna | Nov 27 |  |
| Stretch Requirements Implementation | Dhruv & Pradyumna | Dec 3 |  |
| Main system testing | Dhruv & Pradyumna | Dec 6 |  |
| Low power requirements testing | Dhruv & Pradyumna | Dec 6 |  |

**Table 1.3**

**GitHub repository URL(s) =** [**https://github.com/CU-ECEN-5823/ecen5823-courseproject-DhruvHMehta**](https://github.com/CU-ECEN-5823/ecen5823-courseproject-DhruvHMehta)

## Section 2 - Update 1

### **Status**

Section Author: Dhruv Mehta & Pradyumna Gudluru

All the parts for the project (1 TEMT6000 Ambient Light sensor and 1 APDS 9960 Gesture Sensor) were obtained before the proposal was due. The tasks for the last week were the design and implementation of the sensor subsystems. For the ambient light sensor, it was decided to use the ADC module as this peripheral can function in low-power modes such as EM2 and EM3. The long-term goal of low-power consumption can thus be achieved if the sensor data can be sampled periodically and then set into a low power state. A simple test to check whether the ambient sensor was returning valid values was written and executed. The received analog values were tested against low light and ambient light for validation. Based on the values obtained, it was concluded that the ambient sensor was functional.

Anticipated date of completion for the integration of gesture sensor is the course update 1, but due to a little higher complexity in the gesture control and the module, the integration of the sensor with server is delayed a bit. Starting my analysis of the ADPS9960 gesture sensor from the datasheet, I could gather some information regarding communication protocol of I2C. Understanding the I/O pin configuration, addresses of the I2C registers for enabling the sensor, reading the data of buffers, etc. helped me design the source code. The integration time and the conversion time are considered for the sensor power up time and data acquisition through the I2C, respectively.

Currently, the project is slightly off schedule as the gesture sensor code has not yet been implemented. However, the revised schedule shown below shows how we plan to complete our tasks for the upcoming update.

### **Design Changes & Updates**

Section Author: Dhruv Mehta

**Ambient Sensor**

For the ambient sensor, it was unclear whether to use the ADC or the ACMP module. For sending data to the Client periodically, it was decided to use the ADC module. This would also allow us to monitor the exact luminosity instead of just checking the analog value against a threshold (ACMP). The ADC can also be configured to interrupt based on set analog thresholds as described in the section highlighted below. (Snippet from the [EFR32xg13-rm](https://www.silabs.com/documents/public/reference-manuals/efr32xg13-rm.pdf) Reference Manual, Page 889).

Text

Description automatically generated

Also, the ADC can function in low-power modes down to EM3 and so we aim to sample the analog data, send an indication to the Client with this data and then power down the module to either EM2/EM3 mode depending on the power consumption. The current status is that the ADC is running without any limitation and the low power modes will be integrated at the end (Refer to project schedule).

**Software Flow Chart for Ambient Light Sensor – Normal Operation**

Diagram

Description automatically generated

**Figure 2.1**

The diagram above shows the normal operation of the TEMT6000 Ambient Light Sensor. In this case, the sensor values are sampled every second and then the device is put to a low power state until the next timer interrupt arrives. Also, the sampled values are indicated to the Client using the Ambient Light Sensor Service.

**Gesture Sensor**Section Author: Pradyumna Gudluru

The gesture sensor works on interrupt. Whenever there is a filled buffer, with the registers, GFIFO\_U, the interrupt is enabled and this interrupt is considered as a GPIO interrupt to the Gecko board. Now, the data from all the other buffers is also read from the sensor. For the calibration of the gesture, we use left, right, up and down states. The difference between the data read from the buffers through I2C, is calculated and the ratio is analyzed. From the obtained results the direction of the gesture can be estimated. For the initiation used the I2CSPM\_Init(), is used. Taking support from professor, got to understand the I2C\_FLAG\_WRITE\_READ flag set for transmission and receiving of data.  
  
Diagram

Description automatically generated

**Figure 2.2**

There are predominantly four states considered, based on the gesture, Left, Right, Up and Down. There is a fail or default state, which deals with all of the else cases. The CmdAuth() function gives the direction of the gesture and based on that, the state is updated.

**Updated Flow Chart for Gesture Sensor**

Diagram

Description automatically generated

**Figure 2.3**

### **Challenges and Issues Faced**

Section Author: Dhruv Mehta

I faced the following challenges in the design and integration of the ambient light sensor:

* The EMLIB library header and source files for the ADC peripheral were not included in the project by default. I modified the Project Configuration file in the Simplicity Studio UI to have it generate and include this file.
* Finding the pin mapping to route a GPIO pin to the ADC peripheral took a lot of time. I had to read the reference manual, datasheet and user manual to figure out how the external pins map to the peripheral. I then had to find a pin that is already soldered onto my Gecko Board and use that to test.
* The initial code did not work. My first suspicion was that I was not configuring any clock gating to the peripheral. I verified this by single-stepping through the ADC\_Init function in the debugger. I read the reference manual to find the appropriate clock gating function and it started working after I enabled it in the CMU\_ClockEnable function.

Going forward, I expect the following challenge I might face with the ADC and ambient light sensor:

* Will the ADC peripheral hold its initialization values and configurations if I put the Gecko into a low power mode such as EM2/EM3?

Section Author: Pradyumna Gudluru  
  
I faced the following challenges in the design and integration of the gesture sensor:

* Working with the I2C communication of gesture and understanding the working of I2C\_FLAG\_WRITE\_READ functionality.
* Understanding the complex gesture calibration logic from the obtained data buffers of GFIFO\_U, GFIFO\_D, GFIFO\_L and GFIFO\_R.

### **Schedule Table**

Section Author: Dhruv Mehta & Pradyumna Gudluru

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Student(s) Responsible** | **Target Completion Date** | **Expected Completion Date** |
| Gesture Sensor Subsystem Design | Pradyumna Gudluru | Nov 16 | Nov 16 |
| Gesture Sensor Code Implementation | Pradyumna Gudluru | Nov 18 | Nov 22 |
| Ambient Sensor Subsystem Design | Dhruv Mehta | Nov 12 | Nov 13 |
| Ambient Sensor Code Implementation | Dhruv Mehta | Nov 18 | Nov 18 |
| GATT Server Subsystem Design | Pradyumna Gudluru | Nov 25 |  |
| GATT Server Code Implementation | Pradyumna Gudluru | Nov 27 |  |
| GATT Client Subsystem Design | Dhruv Mehta | Nov 22 |  |
| GATT Client Code Implementation | Dhruv Mehta | Nov 24 |  |
| System Integration and Testing | Dhruv & Pradyumna | Nov 30 |  |
| Stretch Requirements Implementation | Dhruv & Pradyumna | Dec 3 |  |
| Main system testing | Dhruv & Pradyumna | Dec 6 |  |
| Low power requirements testing | Dhruv & Pradyumna | Dec 6 |  |

**Table 1.1**

The updated schedule table is shown above. A few dates are shifted ahead to compensate for the delay in the gesture sensor code implementation.

### **Test Plan Update**

Section Author: Dhruv Mehta & Pradyumna Gudluru

**Test Plan Completion % = 8.33 %**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Plan Table | | | | |
| **Test Number** | **Test Description** | **Date last run on** | **Test Result (Pass/Fail)** | **Notes** |
| 1 | A basic test the checks that the ambient sensor is returning measurements | 11/14/2021 | Pass |  |
| 2 | A basic test the checks that the gesture sensor is returning measurements | 11/15/2021 | To Do |  |
| 3 | A test to check whether the characteristic indications are sent to the client periodically | 11/20/2021 | To Do |  |
| 4 | A test to check whether the COMMAND is updated on the client DISPLAY | 11/23/2021 | To Do |  |
| 5 | A test to check LUMINOSITY value updated periodically on server DISPLAY | 11/23/2021 | To Do |  |
| 6 | A test to check whether DIRECTION received from gesture sensor is updated on the server DISPLAY | 11/23/2021 | To Do |  |
| 7 | A test to check if ERROR/ INFO is printed on LCD on client and server DISPLAY | 11/23/2021 | To Do |  |
| 8 | A test to check the IDLE state of the device | 11/28/2021 | To Do |  |
| 9 | Low Power Management: To check the energy profiler in low power mode | 11/29/2021 | To Do |  |
| 10 | A test to indicate the LED 0 on server(USER) when in Low Power Mode | 11/29/2021 | To Do |  |
| 11 | A test to indicate the LED 0 on client(PAGE) when in Low Power Mode | 11/29/2021 | To Do |  |
| 12 | Full system test – The proposed functionality is working, tested with all possible gestures and also low-light testing. | 12/6/2021 | To Do |  |

**Table 1.2**

**GitHub repository URL(s) =** [**https://github.com/CU-ECEN-5823/ecen5823-courseproject-DhruvHMehta**](https://github.com/CU-ECEN-5823/ecen5823-courseproject-DhruvHMehta)

## Section 3 - Update 2

### **3.1 Status**

Section Author: Dhruv Mehta

Since the last update, the ambient light sensor has been fully integrated into the system. The server now samples the sensor every second (as per the design proposed in Update 1) and then sends the sensor value as an indication to the client. A new state machine was written for this purpose. The sensor value is visible on both the server and the client display after connection is established. Also, the server runs in EM2 mode. The energy profiler was used to determine the current consumption of the ambient light sensor and the code was optimized to reduce the sensor on time. These profiler screenshots are shown in the design updates section.

For the Client subsystem design and code implementation, the server was first completed with a new service for the ambient light sensor. The client was then updated to recognize and enable indications for this service. The validation of data was completed by displaying the sensor values both on the server and the client. After this, the next task was to handle the gestures and display some data on the display. For this purpose, a new module was created which only handles display updates based on the gesture sensor inputs. Testing was done using the pushbuttons as stub functionality till the gesture inputs could be obtained from the server. As of now, the scroll up, next page and previous page features are tested successfully using the pushbuttons. Once the gesture sensor is integrated fully, the sensor values can be used.

**Gesture Sensor**Section Author: Pradyumna Gudluru

Since the last update, the gesture is fully functioning and partially integrated with the system. The gesture sensor is particularly tested with the waving of hand and the output is verified on the Tera Term window. The functionality of the gesture sensor is as per the designed state chart and by using the authentication function, the directionality is defined. For integrating the whole subsystem to the server module, new design flow is being updated in the next section of this update.

For the Server subsystem design and code are developed. The server is updated with a new service for the ambient sensor. This is a generic service, which doesn’t require bonding for the data to transfer from server to client. The value is successfully sent to the client and the display is updated accordingly. For the gesture sensor, designed the code to be an encrypted service which transfers data only when bonded. Testing was done for the ambient sensor data transfer.

### **3.2 Design Changes & Updates**

Section Author: Dhruv Mehta

There are no design changes in the system. As for the update, the ambient sensor current consumption in EM2 mode is shown below (Figure 3.1). The spikes at every 1s is the ambient light sensor being sampled. It is a bit unclear as to why the average is 1.71mA over a period of 1s as the current consumption is expected to be in the order of uA in EM2 mode. This will be investigated before the final project submission. Also, these screenshots are taken on the server with Bluetooth functionality turned off (As the connection intervals are short and it is hard to identify current consumption of the ambient sensor).

The second image (Figure 3.2) shows the view of a single sample of the ambient light sensor. The sensor is powered on and 1mS is given for stabilization of power. After this, the ADC Start conversion is initiated and the system goes back to sleep and only turns on when the conversion complete interrupt is received. Here, the ADC data is now read from the ADC read register and displayed on the LCD screen as well as logged on the serial monitor. The complete process takes 7.3mS as shown, but it can be further reduced by turning off the LOG statement to the terminal which will be eventually done as we work on the low-power aspects of the system. For now, the result does seem acceptable.

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 3.1**

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 3.2**

As for the client implementation, the images below shows the client display after it is connected to the server. The first 12 rows are the data which the user can scroll through using the gestures. The last row indicates the ambient light sensor value. On pressing PB1, which simulates a left swipe, the next page is shown. Similarly, on pressing PB0, which simulates a right swipe, the previous page is shown. Therefore, the client functionality is verified using the pushbuttons. Some more testing may be required to iron out any unforeseen bugs.

Initial Screen Pressing PB1 – next Page Pressing PB0 – previous Page

A picture containing text, electronics, circuit

Description automatically generated A picture containing text, electronics

Description automatically generated A hand holding a computer chip

Description automatically generated with low confidence

**Figure 3.3**

**Gesture Sensor**Section Author: Pradyumna Gudluru

There is a design change based on the implementation of the gesture sensor initialization and functioning. The functions used for reading data through the I2C from the sensor and writing to the registers from the gecko board is done with four different functions namely,

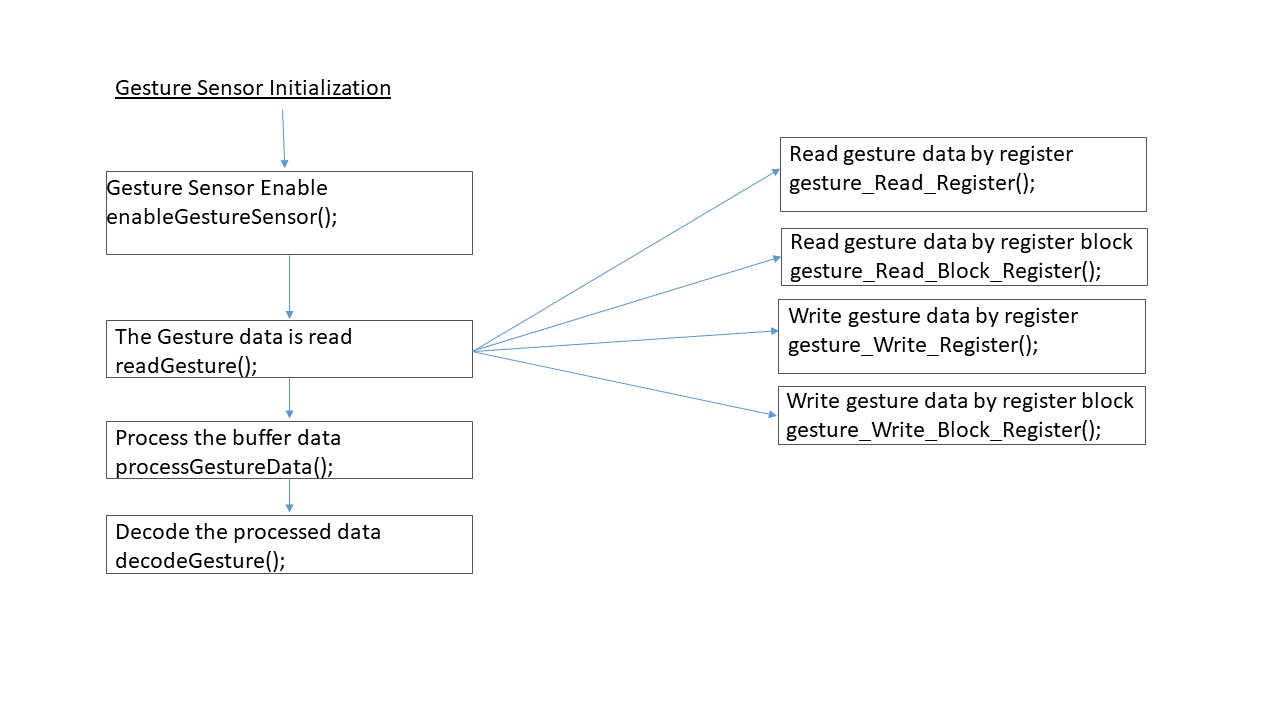
***gesture\_Write\_Register()***

***gesture\_Write\_Block\_Register()***

***gesture\_Read\_Block\_Register()***

***gesture\_Read\_Register()***

According to the flag of *I2C\_FLAG\_WRITE\_READ*, updated the code for read and write operations on the I2C functionality. The working of the gesture sensor is as per the application of the ADPS9960, with the datasheet. The registers of gesture sensor are initialized to zero. Then the sensor is enabled and powered. The sensor starts to read the buffers, i.e, UP, DOWN, LEFT and RIGHT. With this buffer data, the values of difference ratios are calibrated. This calculated differences of ratios are updated in the decode function for defining the direction of the gesture. This is updated to a state-chart as per the design. The following diagram gives the design for processing and decoding of gesture data.



**Figure 3.4**

### **Challenges and Issues Faced**

Section Author: Dhruv Mehta

I faced the following challenges in the design and integration of the client subsystem:

* Initially, the client was only able to display some incorrect ambient light values. I suspected that only 1 byte of information was being transferred or read on the client side. The ambient light value has a range of 0-4095 and so I had allocated 2 bytes in the local GATT database. However, this was fixed as I identified a bug in my code which only read the first (lower) byte. The relevant commit for the bug fix is at [this link](https://github.com/CU-ECEN-5823/ecen5823-courseproject-DhruvHMehta/commit/bacb6f0109eb6ebc91df017ae5bdb049d22775cc#diff-5ff84b161f240c533f7c08ac012102c0e1add491c08ad7d3ce792373b5c37201) (lines 818-819).
* Implementing the display scroll up, next page and previous page features proved to be a bit of a challenge. It probably took as much time as the sensor integration itself as I had to account for a lot of cases and figure out what sort of delimiters to use to scroll the page or move to the next page. Also, I had to make sure to not print more than 20 characters on a single line, otherwise the extra characters would be discarded. This required me to read the lcd.c implementation and figure out what I could and could not do with the displayPrintf API.

**Gesture Sensor**Section Author: Pradyumna Gudluru

I faced the following challenges while working on the gesture sensor development and integration:

* Working with the functionality of the previous design of initialization of gesture sensor, there was no clear algorithm for the reading of gesture. By taking support from ADPS9960 application algorithm, Silicon Labs driver level source code was able to understand the modularity in implementing the code. The signal flow of the gesture functionality is comprehended and updated accordingly in the present project update.
* While developing the main block register read and write functions, I was facing issues in the exact transfer of data for read and write on the I2C data lines. With the understanding from the reference manual and different flags in the transferring of data, I could manage to get the data transferred on to the I2C bus.

### **Schedule Table**

Section Author: Dhruv Mehta & Pradyumna Gudluru

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Student(s) Responsible** | **Target Completion Date** | **Expected Completion Date** |
| Gesture Sensor Subsystem Design | Pradyumna Gudluru | Nov 16 | Nov 16 |
| Gesture Sensor Code Implementation | Pradyumna Gudluru | Nov 18 | Nov 22 |
| Ambient Sensor Subsystem Design | Dhruv Mehta | Nov 12 | Nov 13 |
| Ambient Sensor Code Implementation | Dhruv Mehta | Nov 18 | Nov 18 |
| GATT Server Subsystem Design | Pradyumna Gudluru | Nov 25 | Nov 28 |
| GATT Server Code Implementation | Pradyumna Gudluru | Nov 27 | Nov 29, ambient only |
| GATT Client Subsystem Design | Dhruv Mehta | Nov 22 | Nov 24 |
| GATT Client Code Implementation | Dhruv Mehta | Nov 24 | Nov 29, ambient only |
| System Integration and Testing | Dhruv & Pradyumna | Dec 5 |  |
| Stretch Requirements Implementation | Dhruv & Pradyumna | Dec 6 |  |
| Main system testing | Dhruv & Pradyumna | Dec 6 |  |
| Low power requirements testing | Dhruv & Pradyumna | Dec 6 |  |

**Table 1.1**

The updated schedule table is shown above. A few dates are shifted ahead to compensate for the delay in the gesture sensor code implementation.

### **Test Plan Update**

Section Author: Dhruv Mehta & Pradyumna Gudluru

**Test Plan Completion % = 41.67 %**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Plan Table | | | | |
| **Test Number** | **Test Description** | **Date last run on** | **Test Result (Pass/Fail)** | **Notes** |
| 1 | A basic test the checks that the ambient sensor is returning measurements | 12/03/2021 | Pass | Confirmed on LCD. |
| 2 | A basic test the checks that the gesture sensor is returning measurements | 12/03/2021 | Pass | Confirmed on LCD. |
| 3 | A test to check whether the characteristic indications are sent to the client periodically | 12/03/2021 | Pass | Tested with ambient sensor. |
| 4 | A test to check whether the COMMAND is updated on the client DISPLAY | 11/23/2021 | To Do |  |
| 5 | A test to check LUMINOSITY value updated periodically on server DISPLAY | 12/03/2021 | Pass | Client and Server display the value. |
| 6 | A test to check whether DIRECTION received from gesture sensor is updated on the server DISPLAY | 11/23/2021 | To Do |  |
| 7 | A test to check if ERROR/ INFO is printed on LCD on client and server DISPLAY | 11/23/2021 | To Do | May not be required. |
| 8 | A test to check the IDLE state of the device | 12/03/2021 | Pass | Devices connect and waits for input. |
| 9 | Low Power Management: To check the energy profiler in low power mode | 11/29/2021 | To Do |  |
| 10 | A test to indicate the LED 0 on server(USER) when in Low Power Mode | 11/29/2021 | To Do |  |
| 11 | A test to indicate the LED 0 on client(PAGE) when in Low Power Mode | 11/29/2021 | To Do |  |
| 12 | Full system test – The proposed functionality is working, tested with all possible gestures and also low-light testing. | 12/6/2021 | To Do |  |

**Table 1.2**

**GitHub repository URL(s) =** [**https://github.com/CU-ECEN-5823/ecen5823-courseproject-DhruvHMehta**](https://github.com/CU-ECEN-5823/ecen5823-courseproject-DhruvHMehta)

## Section 4 - Final Report

### **4.1 Status**

Section Author: Dhruv Mehta

The project was implemented completely as per the requirements. It is completely functional; however some low-power techniques could be improved and it is not an optimal implementation. This could be something that could be improved upon in the future.

Section Author: Pradyumna Gudluru

After the second update, the gesture service is created with fully encrypted link from server to the client Bluetooth. The data is being transferred only after the bonding is done in between the client and the server. The direction is handled at the client side and the gesture is fully integrated with the server. This is tested using the EFR phone application. For the low power mode on the server side, the whole application is run in EM2 mode. Gesture sensor is using I2C for the transfer of information and for writing and reading data from the sensor, SPM module is used. Since the working of the APDS9960 gesture sensor is interrupt based, the I2C data transfer is only initiated when an interrupt comes, this reduced the power consumption on a whole.

### **4.1 Design Updates**

Section Author: Dhruv Mehta & Pradyumna Gudluru

There are no design changes for the ambient light sensor or the server from update 2. However, there are changes from Update 1. The earlier design was that the ambient light sensor should turn off the gesture sensor and the BLE stack on low light. When the light would rise above a set threshold, the ADC interrupt should have woken up the system and re-connected to the client. This would ensure optimal power consumption. However, I was unable to disconnect and turn off the BLE stack. The disconnection worked, but the server immediately entered advertising and I was receiving Bluetooth events. In hindsight, I could have turned off advertising on connection closure and observed the behavior from there. Due to shortage of time, I did not pursue that route.  
  
**Low-power Server implementation**  
As for the low-power implementation, please refer to the images below:

Graphical user interface

Description automatically generated

**Figure 4.1**

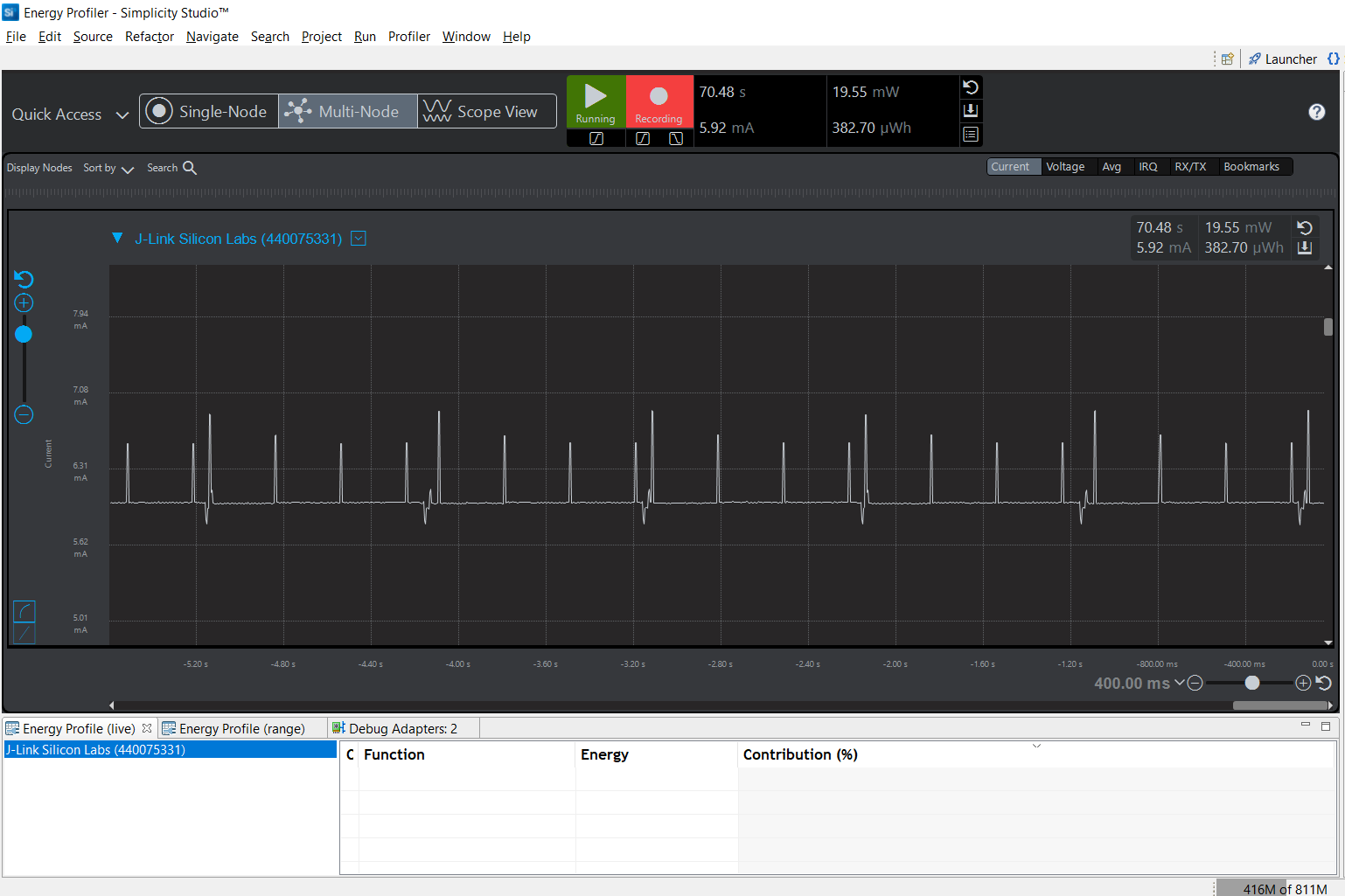
A screenshot of a computer

Description automatically generated with medium confidence

**Figure 4.2**

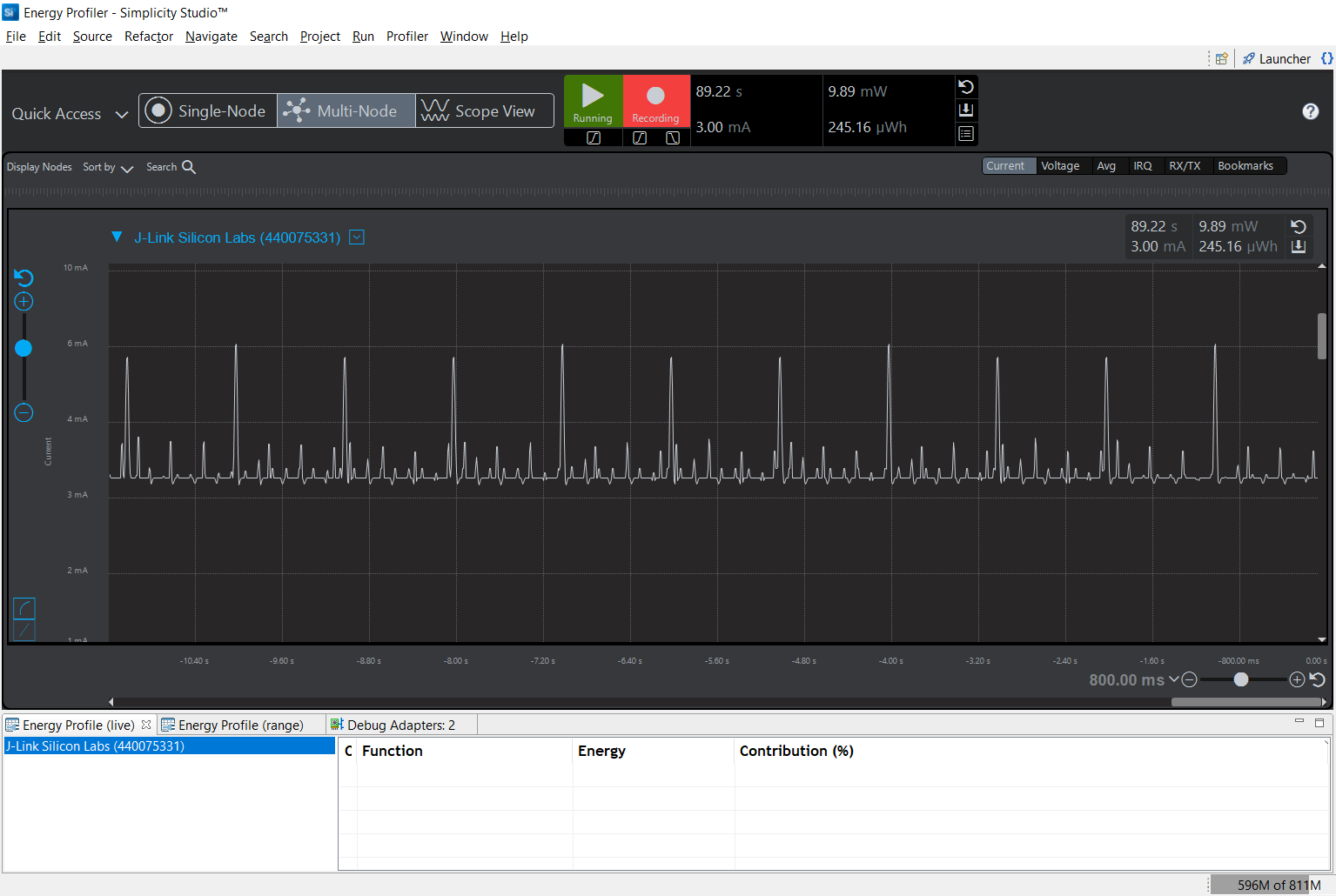
Figure 4.1 shows the power consumption of the ambient light sensor over a single period (1s). Here, I faced a problem where I was continuously receiving an external signal from the stack even though I did not explicitly set that event. I could not trace the source of this bug but it probably contributed to a significant amount of power consumption.

Figure 4.2 shows the power consumption for a single cycle. The ADC is powered on, single conversion interrupt is set up and the system goes to sleep. When the conversion complete interrupt arrives, the data is received and sent as an indication to the client. The total on time is about 7.3mS, but that is including a LOG print removing which, the time is about 2.5mS only.



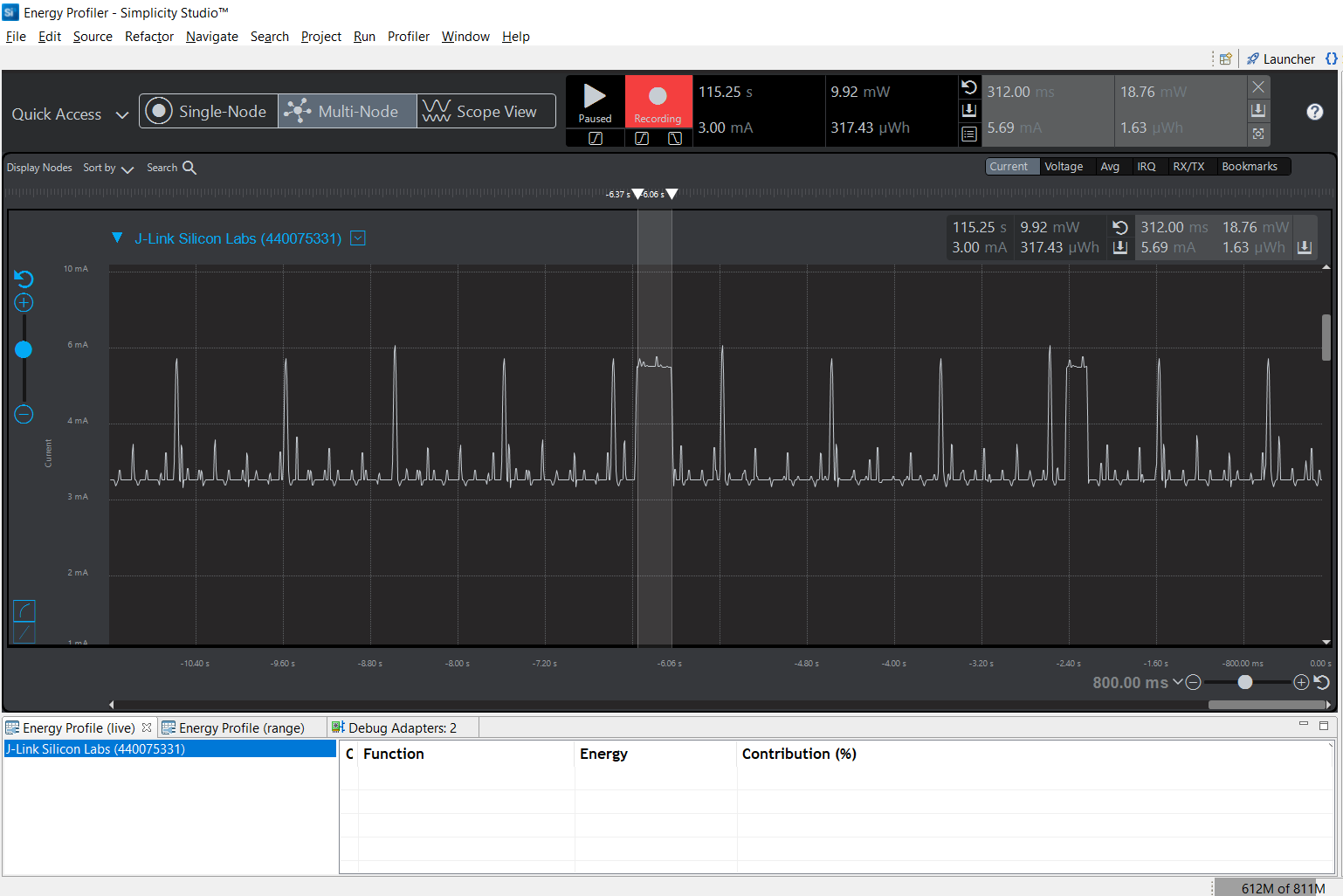
**Figure 4.3**

Figure 4.3 shows the power consumed by the server overall, with both the gesture and the ambient sensors working. Since the gesture sensor works only on the interrupt and the ambient sensor is functional every 1 second, the overall current consumption on an average is 5.92mA.



**Figure 4.4**

Figure 4.4 shows the gesture sensor power consumption when not in use and powered on. The gesture sensor with the BLE data transfer consumes around 3.00mA when in idle state. Since the gesture sensor transfers the data on the I2C only when the interrupt comes, it consumes lesser power when compared to sensing values periodically. This makes the gesture sensor to consume 5.69mA (Figure 4.5) for getting the data on I2C to the Gecko board and transferring the data via the Bluetooth from the server to client.



**Figure 4.5**

**4.1 Requirement Status**

Section Author: Pradyumna Gudluru

At the time of project proposal, we have made a list of the below requirements and this the status of those requirements.

We have completed all the proposed requirements as per the proposal, but due to time crunch and unexpected complexity in developing the gesture sensor algorithm, we couldn’t update the stretch goals.

|  |  |  |
| --- | --- | --- |
| S.No. | Requirement | Status |
| 1 | The system shall consist of a BLE GATT Server and BLE GATT Client implementation | Done |
| 2 | The system shall implement two new GATT Services –  one for gesture sensor updates and one for ambient light updates. | Done |
| 3 | The Gesture sensor service shall have a single characteristic to update the direction of motion to the client. | Done |
| 4 | The Ambient light sensor service shall have a single characteristic to update the light intensity to the client. | Done |
| 5 | The system shall use the LCD for the devices in the following ways: Server: Display connection status, Gesture direction, luminosity. Client: Display a block of text for the user to read. | Done |
| 6 | The server shall interface with the gesture sensor using I2C protocol. | Done |
| 7 | The server shall use interrupt driven I2C communication to receive data from the  gesture sensor when a motion is made by the user. | Done |
| 8 | The server shall interface with the ambient light sensor using the ACMP module. | Done |
| 9 | The server shall maintain a low energy state when the luminosity falls below a pre-set threshold. | Done |
| 10 | The client shall establish an encrypted connection with the server on bootup and access both services. | Done |
| 11 | The client shall display a page of data to the user which can be scrolled up or down by gesturing up or  down at the server respectively. | Done |
| 12 | The client shall display a page of data to the user which can be moved to the next or previous page by  gesturing right or left at the server respectively. | Done |
| 13 | The client shall turn off the display when the server indicates that the ambient light is low. | Done |

### **4.1 Distribution of Work**

Section Author: Dhruv Mehta

The work was distributed as evenly as possible, as highlighted in the table below

|  |  |  |
| --- | --- | --- |
| **File** | **Subsystem / Function** | **Owner** |
| app.c/h | Initializations/State machine calls | Dhruv Mehta |
| Initializations/State machine calls | Pradyumna Gudluru |
| adc.c/h | ADC initialization | Dhruv Mehta |
| gesture\_i2c.c/h | I2C Read/Write functions | Pradyumna Gudluru |
| gesture\_main.c/h | Gesture Sensor Implementation | Pradyumna Gudluru |
| ble.c/h | Server code implementation | Pradyumna Gudluru |
| Client code implementation | Dhruv Mehta |
| gpio.c/h | ADC power pin setup | Dhruv Mehta |
|  | Gesture interrupt pin setup | Pradyumna Gudluru |
| i2c.c/h | I2C initialization | Pradyumna Gudluru |
| irq.c/h | Timer/ADC IRQ Handler | Dhruv Mehta |
| GPIO Handler | Pradyumna Gudluru |
| PageDisplay.c/h | Client LCD Display functionality, scrolling and gesture handling | Dhruv Mehta |
| scheduler.c/h | Ambient light state machine | Dhruv Mehta |
| Gesture sensor state machine | Pradyumna Gudluru |

### **4.2 Schedule Table**

Section Author: Dhruv Mehta

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Student(s) Responsible** | **Target Completion Date** | **Expected Completion Date** |
| Gesture Sensor Subsystem Design | Pradyumna Gudluru | Nov 16 | Nov 16 |
| Gesture Sensor Code Implementation | Pradyumna Gudluru | Nov 18 | Nov 22 |
| Ambient Sensor Subsystem Design | Dhruv Mehta | Nov 12 | Nov 13 |
| Ambient Sensor Code Implementation | Dhruv Mehta | Nov 18 | Nov 18 |
| GATT Server Subsystem Design | Pradyumna Gudluru | Nov 25 | Nov 28 |
| GATT Server Code Implementation | Pradyumna Gudluru | Nov 27 | Nov 29, ambient only Dec 4 complete |
| GATT Client Subsystem Design | Dhruv Mehta | Nov 22 | Nov 24 |
| GATT Client Code Implementation | Dhruv Mehta | Nov 24 | Nov 29, ambient only Dec 5 complete |
| System Integration and Testing | Dhruv & Pradyumna | Nov 30 | Dec 5 |
| Stretch Requirements Implementation | Dhruv & Pradyumna | Dec 3 | N/C |
| Main system testing | Dhruv & Pradyumna | Dec 6 | Dec 6 |
| Low power requirements testing | Dhruv & Pradyumna | Dec 6 | Dec 6 |

### **4.3 Test Plan Update**

Section Author: Dhruv Mehta & Pradyumna Gudluru

**Test Plan Completion % = 100 %**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Plan Table | | | | |
| **Test Number** | **Test Description** | **Date last run on** | **Test Result (Pass/Fail)** | **Notes** |
| 1 | A basic test the checks that the ambient sensor is returning measurements | 12/07/2021 | Pass | Confirmed on LCD. |
| 2 | A basic test the checks that the gesture sensor is returning measurements | 12/07/2021 | Pass | Confirmed on LCD. |
| 3 | A test to check whether the characteristic indications are sent to the client periodically | 12/07/2021 | Pass | Tested with ambient sensor and gesture sensor. |
| 4 | A test to check whether the COMMAND is updated on the client DISPLAY | 12/07/2021 | Pass | Confirmed on LCD. |
| 5 | A test to check LUMINOSITY value updated periodically on server DISPLAY | 12/07/2021 | Pass | Client and Server display the value. |
| 6 | A test to check whether DIRECTION received from gesture sensor is updated on the server DISPLAY | 12/07/2021 | Pass | Confirmed on LCD. |
| 7 | A test to check if ERROR/ INFO is printed on LCD on client and server DISPLAY | 12/07/2021 | N/A | Removed from test plan – Not required. |
| 8 | A test to check the IDLE state of the device | 12/07/2021 | Pass | Devices connect and waits for input. |
| 9 | Low Power Management: To check the energy profiler in low power mode | 12/07/2021 | Pass | Checked Server in EM2 mode. |
| 10 | A test to indicate the LED 1 on server(USER) when in Low Power Mode | 12/07/2021 | Pass | LED 1 turns off, meaning indications are off for gesture sensor. |
| 11 | A test to indicate the LED 0 on client(PAGE) when in Low Power Mode | 12/07/2021 | N/A | Removed from test plan – Not required. |
| 12 | Full system test – The proposed functionality is working, tested with all possible gestures and also low-light testing. | 12/07/2021 | Pass | Works as per requirements. |

### **4.4 What was Learned**

Section Author: Dhruv Mehta

Design of a project before implementation is my major takeaway from this project. Until now, I have been jumping right into the code implementation and have kept the design in my mind. This approach works, but it is hard to collaborate with project members and explain the design when you have nothing on paper. Providing timely updates and designing software flow charts, UML (sequence) diagrams and writing the system requirements helped me figure out what is exactly needed. I also do not need to think about the design once I have it on paper, since I can simply refer to it and implement my code. Also, it becomes easier to refer to the project at a later point in time since everything is documented.

Another thing I learned is spending time to read the datasheet, user guide and reference manual to implement the ADC peripheral code. It is a simple sensor but since the ADC peripheral was completely new, it took my longer than expected to get it working.

Section Author: Pradyumna Gudluru

Understanding the gesture sensor for the data transfer through I2C protocol and to process and decode the obtained buffer values is a big time learning for me from this project. Identifying a real world problem and trying to find a suitable solution with the knowledge we gathered throughout the course is a great learning experience. Spending reasonable time on the process of project completion, from designing based on the requirements, then developing the source code based on design, verifying the developed application as per the requirements and finally updating the software as per the issues, the complete project lifecycle helped me in getting to understand the industrial process scenario. Debugging is a new skill that I have updated in, while working with this project. By walking through the code, while implementing the gesture algorithm, I could analyze the code better and made my debugging process easy.