Fall 2021 ECEN 5823
Course Project Report
Team 16

Objective: The purpose of the course project is to design an IoT or wireless product based on BLE or Bluetooth Mesh and implement a Proof of Concept (POC) demo of your product idea.

There are a number of deliverables that will be due over time. These deliverables are:

1. Project Proposal (PDF documents)
2. Update 1 (PDF documents)
3. Update 2 (PDF documents)
4. Final Report (PDF documents and URL to GitHub repository included in the PDF)
5. Project Demonstration (in class presentation to other students)

Due dates for these deliverables are in Canvas. It is your responsibility to track and be aware of these due dates. When you get out into the corporate world, your supervisor will assign tasks to you with due dates. Your supervisor will expect you to track these dates and deliverables.

There are 4 sections in this document corresponding to items 1 through 4 above. You will use this file to submit items 1 through 4. As these items become due, you will submit a copy of this document with the section filled out that is due. The sections are:

6. Project Proposal
7. Update 1
8. Update 2
9. Final Report

Is each student required to submit a copy of the deliverable(s)? No. Only 1 set of files is required to be submitted per team/individual. For teams it can be the same student for all 4 deliverable due dates, or it can be different students. The choice of which team member that submits the deliverable(s) is up to each team. **Please note**: What I'd like each team/individual to do when filling out this template is:

At due date for the Proposal:

- Fill out Section 1
- Leave Sections 2-4 as is.
- Submit 1 PDF of this Project Report

At due date for the Update 1:

- Leave Section 1 as previously submitted
- Fill out Section 2
- Leave Sections 3-4 as is.
- Submit 1 PDF of this Project Report

At due date for the Update 2:

- Leave Section 1 as previously submitted
- Leave Section 2 as previously submitted

- Fill out Section 3
- Leave Section 4 as is.
- Submit 1 PDF of this Project Report

At due date for the Final Report:

- Leave Section 1 as previously submitted
- Leave Section 2 as previously submitted
- Leave Section 3 as previously submitted
- Fill out Section 4
- Submit 1 PDF of this Project Report

The goal of this is to build a kind of “running history over time”. We’ll be able to see where you started at the Proposal stage and the changes that occurred over time by the time we get to the Final Report. Key learnings can come out of reviewing the project progress over time.

As you saw in the assignment document for options 2 and 3, I refer to “*You need to convince me that the work was evenly distributed*”. To this end, that is the intention of Section Authors below (among other things I’ll be looking at to make a determination towards equal distribution of work.). The amount of work for each section is not the same. The Proposal and the Final Report are more work, Updates 1 and 2 are less work. So having 1 Section Author per section is my minimum requirement. Some teams in the past have had multiple Section Authors embedded within each major Section (Section 1, 2, 3, 4). This is an improvement of fidelity and I would appreciate seeing that for team projects. But 1 Section Author per major Section is the minimum.

Once you have completed a section, delete all lines with <this text>.

Section 1 - Project Proposal

The goal of this project is to emulate a health tracker to monitor the **pulse rate (heart beat)** and **free fall detection**. Free fall event is generated when a person falls unconscious. The captured heart rate and free fall is then sent to smartphone application via Bluetooth low energy protocol. The pulse rates are captured using pulse rate sensor from Sparkfun and the free fall is detected using accelerometer.
The values of heart beat and free fall are sent to smartphone every 10 seconds.

Student Names

Sayali Mule sayali.mule@colorado.edu

Project Overview

Team 16 has decided to move forward with option 1.

High Level Requirements

1. System shall comprise the pair of server and client : server-> Blue Gecko, Client -> Mobile phone
2. The Blue Gecko Board should implement and advertise GATT services.
3. The service shall have two characteristics- one for the Free fall detection (Accelerometer) and other for the heart beat sensing(pulse rate sensor).
4. Every 10 seconds should send the values to the cellphone(client).
5. The server shall establish the encrypted link with the cell phone via bonding.
6. The server shall send the values to the client after each 10 seconds and should go to the sleep thereafter.
7. The server should be in the lowest possible energy mode.
8. The values would be communicated to the client using BLE.
9. The client(cell phone) shall display values to the user.
10. The server LCD should display the values of free-fall and heart rate.
11. The LCD should display the following :
    a. Connection status : Advertising/Connected/Bonded
    b. Current indications enabled: Indications enabled
12. Client should display the free fall and pulse rate values.

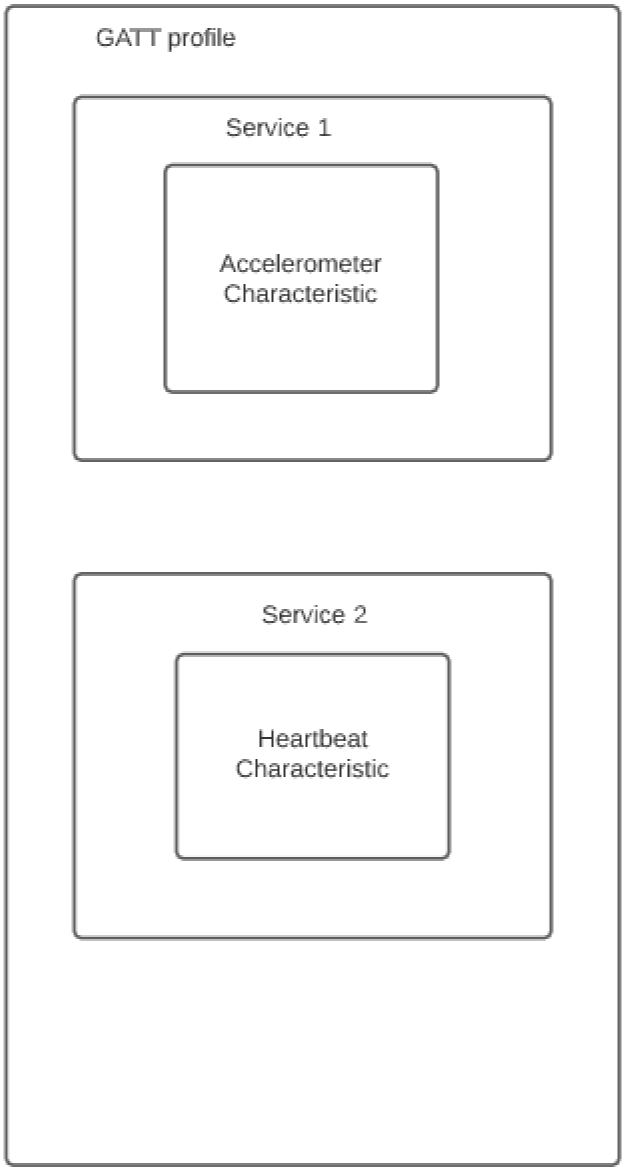
High Level Design

The entire system consists of server part (blue gecko board) and client which is a Bluetooth based application running on mobile phone. The server will measure the free fall (accelerometer value) and heart beat values (pulse rate sensor) which are then communicated to client via BLE protocol.
The client has provision to enable server indications and receive the indications every 10 second. On reception, the client will display the values on the mobile application. Simultaneously, the server LCD will also display the values for both the services.

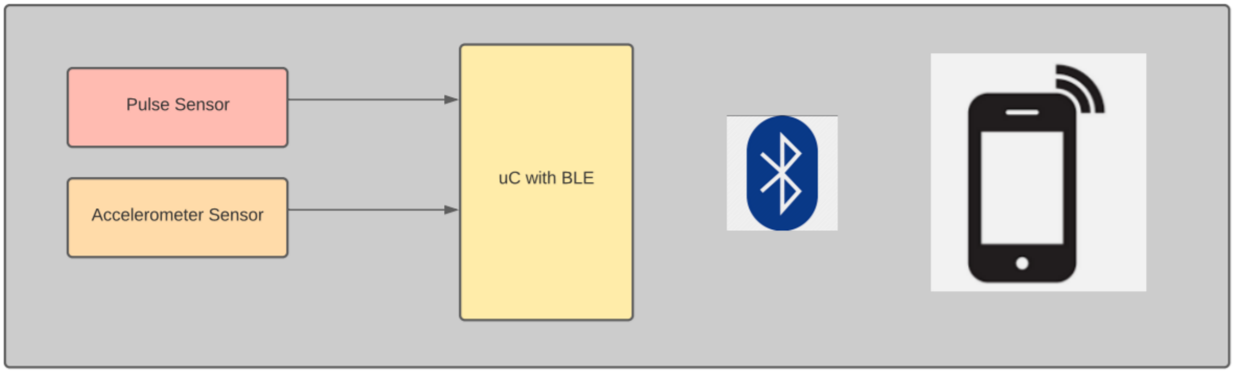
Table 1 : Data Types

Measurement	Unit	Datatype
Heart rate	bpm	uint16_t
Axis orientation(free fall)	milliG	uint16_t

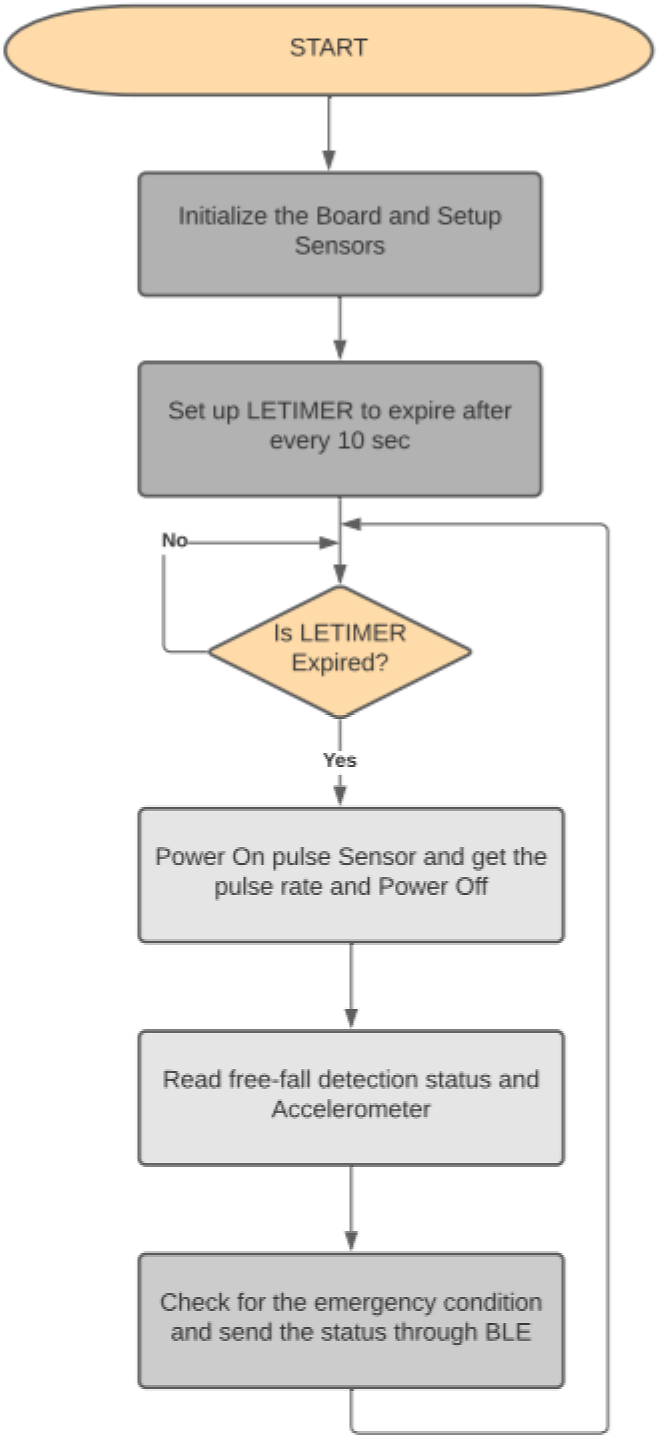
Wireless communication details as appropriate for your product. Examples would include the new GATT services, or Client functionality, or Mesh Node element, model and message types you plan to use.



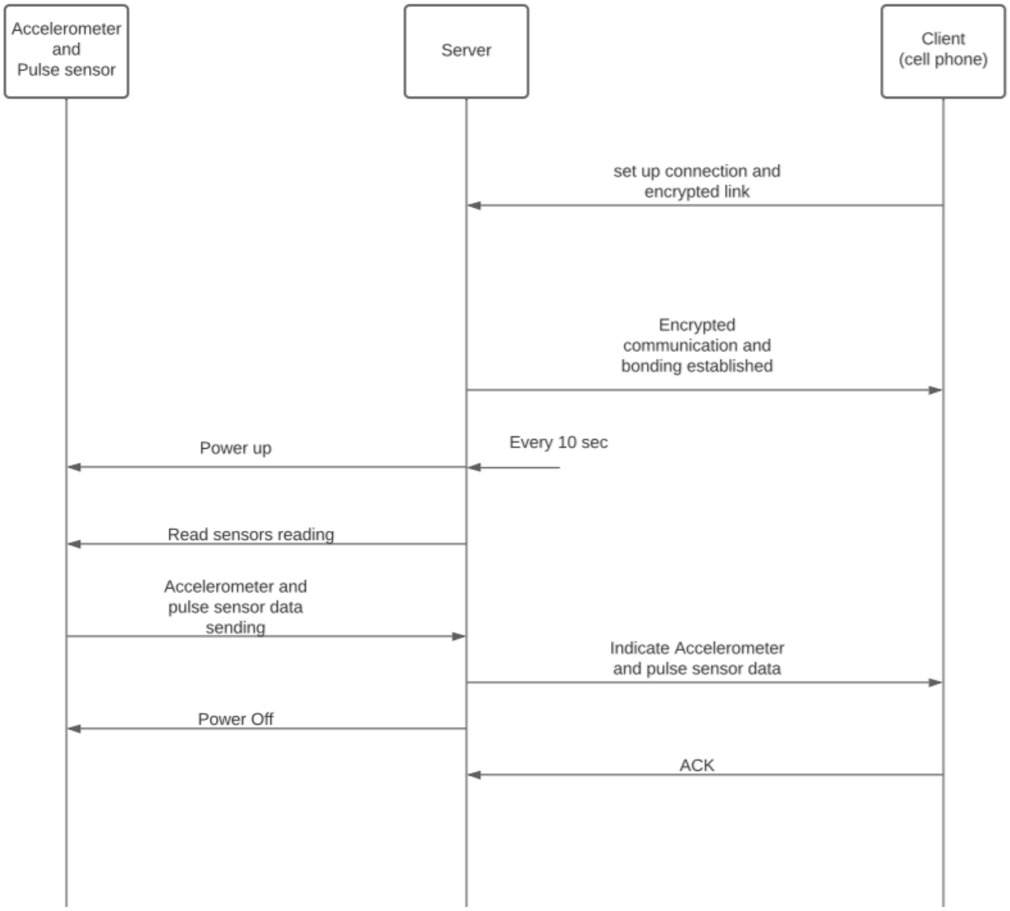
Functional hardware block diagram, including all used existing and added sensors/memories/devices, LCD and user interface description(s).


Pulse Sensor
Accelerometer Sensor
uC with BLE

Functional software block diagram:


START
Initialize the Board and Setup Sensors
Set up LETIMER to expire after every 10 sec
No
Is LETIMER Expired?
Yes
Power On pulse Sensor and get the pulse rate and Power Off
Read free-fall detection status and Accelerometer
Check for the emergency condition and send the status through BLE

Dataflow diagram:


Accelerometer and Pulse sensor
Server
Client (cell phone)
set up connection and encrypted link
Encrypted communication and bonding established
Power up
Every 10 sec
Read sensors reading
Accelerometer and pulse sensor data sending
Indicate Accelerometer and pulse sensor data
Power Off
ACK

Division of labor:

The hardware components (blue gecko board, sensors) are bought off the shelf. Development related to this is limited to 5%. Majority of the efforts are concentrated towards software design and implementation. Firmware implementation for sensor reading and BLE send/receive is the most critical part of this project.

Subsystem Summary

The project will consist of blue gecko server and a client (mobile phone application). The system is combination of hardware and software. The server will be in lowest possible energy mode for most of the time. The values are communicated to the client every 10 seconds. The values of both the GATT services will be displayed on server on LCD and on mobile application on the client end.

Test Plan

Submitted in separate .xlsx file

Proposed Schedule

Task	Target Completion Date	Expected Date
LETIMER Implementation and testing	11/12/2021	11/12/2021
Accelerometer I2C read write	11/15/2021	11/15/2021
Verified functionality of ADC	11/17/2021	11/17/2021
Heart pulse sensor implementation and testing	11/18/2021	11/18/2021
LCD Display	11/19/2021	11/19/2021
Radio Transmission	11/25/2021	11/25/2021
Final Integration and testing	11/30/2021	11/30/2021

GitHub repository URL(s) : **https://github.com/samu7988/ECEN5823_Project_Health_tracker**

Section 2 - Update 1

Section Author: Sayali Mule

Status

Work done:

Hardware:

1) Procurement of hardware (sensors, breadboard, resistors, blue gecko board), and assembling all of them.
2) Needed to solder the accelerometer as well as some pins of the blue gecko board so soldering of the accelerometer is done.

Firmware:

Interfacing of the heart rate sensor (pulse sensor):

1) Basically, pulse sensor is used to determine the person's heart beat in BPM. The ideal range of the BPM for human heart is (40-120).
2) Pulse sensor generates output in analog form. To interface with our microcontroller we require digital values, to achieve that we have programmed the ADC peripheral to sense the analog voltage that is generated by our pulse sensor. The ADC is programmed in 12-bit resolution with internal reference voltage of 5V.

Interfacing of Accelerometer:

1) Implemented setup accelerometer in which following API's are called and tested:
   a. **read_accelerometer_register** : Read DEVID register by this API.
   b. **write_accelerometer_register** : Enable measurement mode by writing value 0x08 into power control register
   c. **clear_setting** : To clear setting of certain accelerometer registers.
   d. **set_free_fall_threshold** : Programmed the user value threshold
   e. **set_free_fall_duration** : function to set the free fall duration.
   f. **Set_interrupt** : Function to setup interrupt for free fall detection.

Work in progress:

1) Free fall detection of accelerometer which has been interfaced with I2C protocol.
2) Development of accelerometer and heartbeat state machine

Work Needs to be done:

1) Sending services and characteristics such as Accelerometer free fall and heartbeat values to the client from blue gecko.
2) Connection and bonding of server and client
3) Assign UUID to all the characteristics to be sent to the clients
4) Testing whether the values sent successfully or not to the client.
5) Integrating the complete system.

Task	Target Completion Date	Expected Date
Procurement of hardware and assembly	11/11/2021	11/11/2021
LETIMER Implementation and testing	11/12/2021	11/12/2021
Accelerometer I2C read write	11/15/2021	11/15/2021
Verified functionality of ADC	11/17/2021	11/17/2021
Free Fall detection	11/19/2021	11/19/2021
Heart pulse sensor implementation and testing	11/16/2021	11/16/2021
Heart pulse sensor testing the range	11/17/2021	11/17/2021
LCD Display	11/19/2021	11/19/2021
Radio Transmission	11/25/2021	11/25/2021
Bluetooth functionality enable (services and characteristics)	11/21/2021	11/21/2021
Final Integration and testing	11/30/2021	11/30/2021

GitHub repository URL(s) = **https://github.com/samu7988/ECEN5823_Project_Health_tracker**

Section 3 - Update 2

Section Author: Sayali Mule

Status

The goal for this week was to complete all the code implementation elements on both the GATT server and verify whether the sensor values (GATT characteristics) are sent to client side successfully. My project is on schedule.
I am planning to finish the remaining task as early as possible and leave some amount of buffer time to debug any unexpected issues.

Work done:

Firmware:

Pulse sensor:

1) The pulse sensor generates value within range of 40-120 BPM
2) The BPM characteristics values of 1 byte were successfully transmitted from server to client side
3) Also, implemented functionality to manually enable and disable the pulse sensor by use of push button PB1. (Pressing PB1 will enable pulse sensor to read the values of heartbeat in BPM, pressing it again will disable the pulse sensor- this will put the system in sleep mode when pulse sensor is not enabled)

Accelerometer (ADXL345) to detect free fall:

1) Was able to get the accelerometer to generate an interrupt on its INT pin when free fall is detected.
2) The INT pin of accelerometer is connected to GPIO PD13 of blue gecko board.
3) The system is in sleep mode and when free fall is detected, the GPIO IRQ handler will detect the interrupt generated by accelerometer and will set an event for free fall that is further processed in the health tracker state machine.
4) Also, added functionality for user to clear the free fall event that was previously detected by pressing PB0.
5) Overall design would be as follows:
    a. When system boots up, the system would be in low power mode and in sleep state
    b. When free fall happens, the accelerometer will generate an interrupt which will cause the system to wakeup and send a value of 0x01 to client
    c. Now, its responsibility of the user to clear the value sent to client in order to detect the next free fall event.
    d. To clear the previous free fall event, the user needs to press PB0.

Misc.:

1) Used the circular buffer previously designed in assignment in order to handle sending both the pulse sensor and accelerometer value simultaneously to the client side.
2) Added GATT characteristics for pulse sensor(gatt_BPM) and accelerometer(gatt_free_fall) in gatt database using BT configurator.

Challenges encountered:

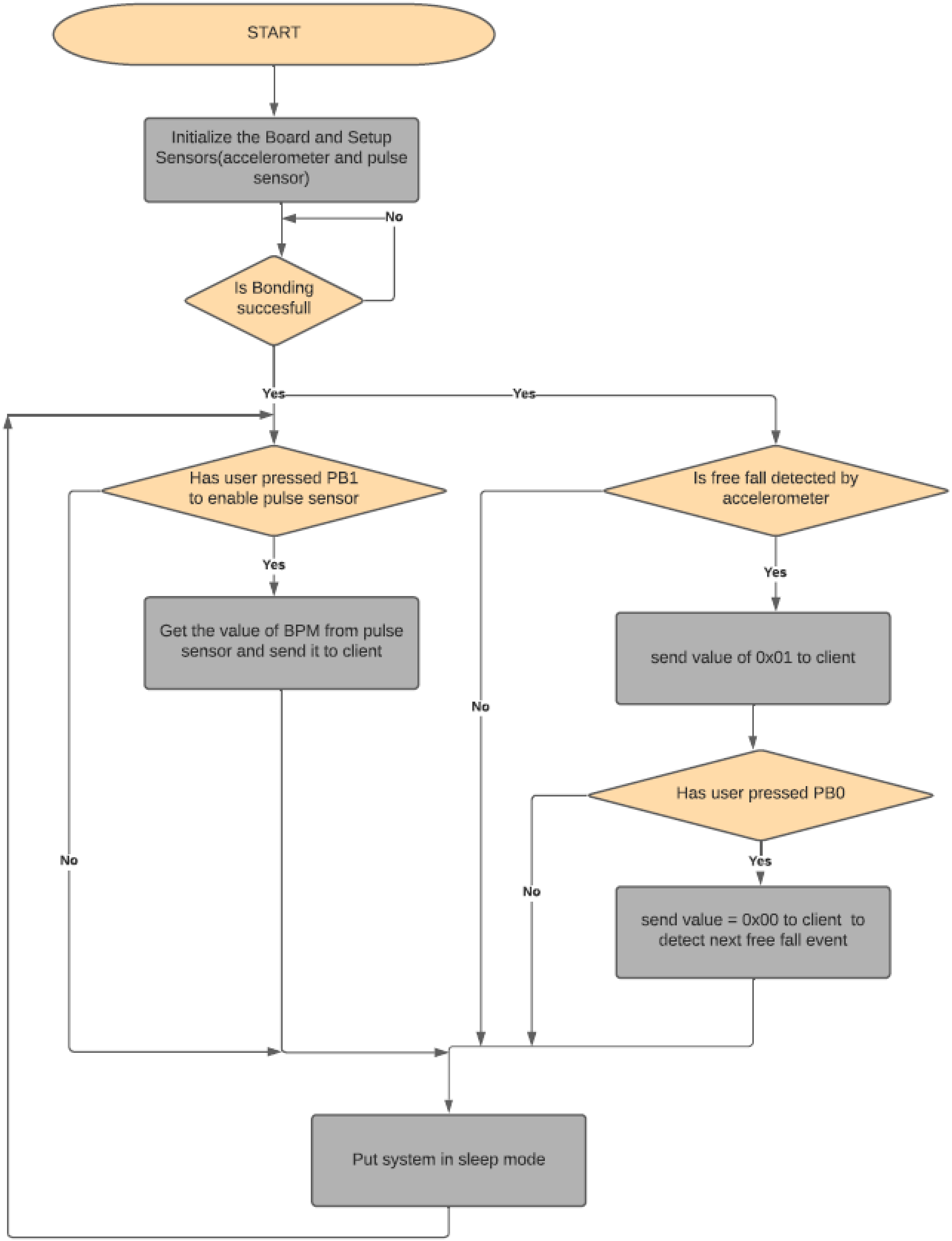
1) Initially GPIO PC10 was connected to INT1 pin of accelerometer to detect the free fall interrupt. However, PC10 is also used by LCD display. This caused the software to detect the false free fall interrupt even though free fall wasn’t generated. After reading the reference manual, I got to know that PC10 is already in use by LCD related functions. I changed the INT1 pin to be connected to PD12
2) Previously, the PB1 button used to enable / disable the pulse sensor was configured to only generate an interrupt on rising edge. This caused the button to generate the interrupt only when the button was pressed. However, the system was supposed to enable / disable the pulse sensor when button was pressed and then released. So, I had to change the configuration of PB1 in gpio_init() function to generate the interrupt on rising/falling edge.
3) The free fall interrupt is generated only when the accelerometer values fall below certain threshold set in certain register. I had to try and test various threshold value in order to get the free fall interrupt to work.

Proposed schedule:

Task	Target Completion Date	Expected Date
Procurement of hardware and assembly	11/11/2021	11/11/2021
LETIMER Implementation and testing	11/12/2021	11/12/2021
Accelerometer I2C read write	11/15/2021	11/15/2021
Verified functionality of ADC	11/17/2021	11/17/2021
Free Fall detection	11/19/2021	11/19/2021
Heart pulse sensor implementation and testing	11/16/2021	11/16/2021
Heart pulse sensor testing the range	11/17/2021	11/17/2021
LCD Display	11/19/2021	11/19/2021
Radio Transmission	11/25/2021	11/25/2021
Bluetooth functionalities enable (services and characteristics)	11/21/2021	11/21/2021
GATT characteristics for pulse sensor	11/20/2021	11/20/2021
GATT characteristics for free fall	11/20/2021	11/20/2021
Sent BPM values from pulse sensor to client	11/21/2021	11/21/2021
Send free fall values from accelerometer to client	11/22/2021	11/22/2021
Clear previously generated free fall event	11/25/2021	11/25/2021
Enable/Disable the pulse sensor	11/23/2021	11/23/2021

Use of circular buffer to integrate both BPM and free value sending.	11/27/2021	11/27/2021
Low power testing	11/30/2021	12/03/2021
Final Integration and testing	12/04/2021	12/04/2021

Changes in software flow:


START
Initialize the Board and Setup Sensors(accelerometer and pulse sensor)
No
Is Bonding succesfull
Yes
Yes
Has user pressed PB1 to enable pulse sensor
Is free fall detected by accelerometer
Yes
Yes
Get the value of BPM from pulse sensor and send it to client
send value of 0x01 to client
No
No
Has user pressed PB0
Yes
No
send value = 0x00 to client to detect next free fall event
Put system in sleep mode

GitHub repository URL(s) = **https://github.com/samu7988/ECEN5823_Project_Health_tracker**

Section 4 - Final Report

<Use this section to provide your final report content>
<Each section shall be authored (written) by 1 student, with the noted exception below for “What Was Learned”>
Section Author: <Awesome Student>

Status

<
Copy & paste the schedule table from the previous section here, leaving the original Target Completion Date column in tact. Update the schedule dates in the “Expected Completion Date” column with the actual dates of completion for each row. If a row was not completed, mark that with the text Not completed.

Do not paste your test plan in here. **Instead include your test plan .xlsx file with your submission**.

Discuss which of your requirements were implemented and which requirements were not implemented and why.

Discuss what is and is not working, and why. Discuss requirements, tasks and functions that were not implemented due to schedule, bugs, or blockages and why. Comment on whether all requirements and features were implemented.

Also include any changes to your design. Possible candidates could include updated: data types, hardware block diagram, software block diagram, data flow diagram, or any other changes/updates.
>

Distribution of Work

<
Here is where to attest to what percentage of the work you personally contributed, as best as you are able to quantify. List each student’s name and percentage of contribution. This isn’t time per se, but a measure of effort. Think of it as a combination of time spent (brainstorming, designing, problem solving, being available for Q&A) + actual work completed (number of lines of code, debugging, bug fixing, filling out the 4 sections of this file etc). For a team of 4 I am expecting that each section of this document would be written by a different student. For a team of 2, each student would fill out 2 sections of this document. Section 4 is not entirely written by 1 student as each student must write their own “what was learned” below.

Also indicate here which files were owned/authored by each student. Format is:

file_name1.c/.h	Student Name1
file_name2.c/.h	Student Name2
file_name3.c/.h	Student Name3

etc.

This should also be reflected in the file headers, the name of the file owner should be the Creator/Author/Editor of the file.
>

What Was Learned

<Whether this was done as an individual or a team, each student that worked on this project must write about 3 lessons that you personally learned while working on this project. These learnings are above and beyond anything taught in lecture, quizzes, exams and this course's previous programming assignments. Write about what changes to the design, as described in the Project Proposal section, had to be made as a result of issues discovered over the course of developing the proof of concept design. Additionally, what did you find challenging? What did you like and/or have fun with on this project?

The format for this is:

Student Name:
Text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text. Text text text text text text text text text text text text text text etc.

Student Name:
Text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text. Text text text text text text text text text text text text text text etc.
>

<Final Report Deliverables: Each team (or individual) shall submit 1 PDF of this file with this section filled out and 1 copy of your test plan .xlsx file.>

Make sure to keep your GitHub schedule wiki page updated every 2 days over the course of the project. This is the live feedback I can observe while you work on your project.

GitHub repository URL(s) = <URL to your GitHub repository, make sure all your final code has been pushed to your repository.>