**Title: Health tracker**

**Student name**: **Sayali Mule**

**Email:** [**samu7988@colorado.edu**](mailto:samu7988@colorado.edu)

**Github link:** [**https://github.com/samu7988/ECEN5823\_Project\_Health\_tracker**](https://github.com/samu7988/ECEN5823_Project_Health_tracker)

**Project overview**

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.

**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 :
    1. Connection status : Advertising/Connected/Bonded
    2. Current indications enabled: Indications enabled
12. Client should display the free fall and pulse rate values.

**High level design requirements**

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.

**Data types table:**

|  |  |  |
| --- | --- | --- |
| **Measurement** | **Unit** | **Datatype** |
| Heart rate | bpm | uint16\_t |
| Axis orientation(free fall) | milliG | uint16\_t |

**Wireless communication details:** Diagram

Description automatically generated

**Hardware block diagram:**

A picture containing diagram

Description automatically generated

**Software block diagram:** Diagram

Description automatically generated

**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.

**Data flow diagram**

Diagram

Description automatically generated

**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 spreadsheet**

I have included the test verification excel file in the github url

**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** |

**Project Update 1:**

**Student name**: **Sayali Mule**

**Email:** [**samu7988@colorado.edu**](mailto:samu7988@colorado.edu)

**Github link:** [**https://github.com/samu7988/ECEN5823\_Project\_Health\_tracker**](https://github.com/samu7988/ECEN5823_Project_Health_tracker)

1. **Status description, with updates/changes present:**

**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 sensors 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:
   1. **read\_accelerometer\_register** : Read DEVID registor by this API.
   2. **write\_accelerometer\_register** : Enable measurement mode by writing value 0x08 into power control register
   3. **clear\_setting** : To clear setting of certain accelerometer registers.
   4. **set\_free\_fall\_threshold** : Programmed the user value threshold
   5. **set\_free\_fall\_duration** : function to set the free fall duration.
   6. **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.
6. **Updated schedule table present**

**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 functionality enable (services and characteristics)** | **11/21/2021** | **11/21/2021** |
| **Final Integration and testing** | **11/30/2021** | **11/30/2021** |

1. **Progress on test plan, % completed listed:**

According to the test plan proposed the 40% of the project work has been done which includes following:

1)LETIMER fires at 1 sec

2)Heart pulse sensor interfacing and testing

3) Accelerometer interfacing with the blue gecko board using I2C protocol.

1. **Test plan spreadsheet (.xlsx file) present :**

I have included the test verification excel file in the github url

1. **Github URL:**

[**https://github.com/samu7988/ECEN5823\_Project\_Health\_tracker**](https://github.com/samu7988/ECEN5823_Project_Health_tracker)

1. **Incorporated feedback from Project Proposal :**

As per feedback from professors and discussion with TA’s during standup, following questions were asked to be included in the next update:

a) *What if the person being monitored should fall when the MCU is asleep (i.e. the letter hasn’t gone off), should the accelerometer generate an interrupt to the MCU on a GPIO pin and wake up the MCU?*

Ans: Yes, the accelerometer (ADXL 345) which is interfaced to the blue gecko board has a dedicated interrupt pin. The interrupt pin is connected to GPIO pin of MCU. So, when free fall is detected when the MCU is asleep, the interrupt would cause the MCU to wakeup due to GPIO\_IRQ\_handler.

*b) Rate and valid ranges for data types?*

Ans:  
 Valid range for Pulse sensor reading is between 40 – 120, so a uint8\_t data type would be sufficient. Pulse sensor values would be transmitted at fixed rate determined by timer period.

Valid range for accelerometer free fall characteristics is either 0 or 1, so a Boolean or uint8\_t is sufficient. Free fall values would be 0 when no fall is detected and would be 1 when fall is detected.

**Attendance at stand ups :**

For the week (15th November to 19th November), I have attended two standups, one with Rajat and another one with Jake.

Standup attendance:  
Rajat: On 16th November (Tuesday)

Jake: On 18th November (Thursday)

Total standup attended for this week: 2