Fall 2021 ECEN 5823

Course Project Report

Team <fill in your team number here>

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:

* 1. Project Proposal
  2. Update 1
  3. Update 2
  4. 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

<Use this section to describe your project proposal>

<Each section shall be authored (written) by 1 student>

Section Author: <Awesome Student>

### Student Names

<enter names of students working on this project + student’s CU email address, 1 student per line>

<Example: Dave Sluiter [David.Sluiter@colorado.edu](mailto:David.Sluiter@colorado.edu)>

### Project Overview

<Describe which option you’ve chosen to implement: Option 1, 2 or 3. Something like our team (or I) chose to implement option 2.>

<Describe what problem(s) or service(s) this product performs or addresses. Provide 1 or more uses cases.>

### High Level Requirements

<

List of major requirements: The product shall have …, the product shall provide…, the product shall perform…

Requirements should include features and functionality. Perhaps there are low-power requirements or goals.

The list of requirements can include:

* stretch requirements (i.e. stretch goals, if there is time I/we plan to implement X, Y and Z.)
* non-implemented requirements that would/may be required for a real product intended for production

- please note non-implemented requirements here and in the Final Report section below.

This can be a bullet point list.

>

### High Level Design

<

This section must include:

A description of the proposed solution to meet the high level requirements and how this product addresses the problem(s) or service(s) described in Product Overview above.

Description of data types (unsigned integer, signed integer, float etc.), number of bytes per value and valid range (min and max allowed), and units. Also include the data rates; could be expressed in bytes/sec or samples/sec. Tables work well to document your data types. Something like this is useful:

Table 1 : Data Types

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Measurement** | **Units** | **Data Type** | **Valid/Allowed Values**  **(Range)** | **Update Rate** |
| Temperature | Degrees C | uint32\_t | -25 to 125 | 0.5 Hz (2 bytes/sec) |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

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). Look at the diagrams I included in the assignment document to use as a guide for your block diagram(s).

Functional software block diagram. This could be flowcharts, UML diagrams, state machine bubble diagrams

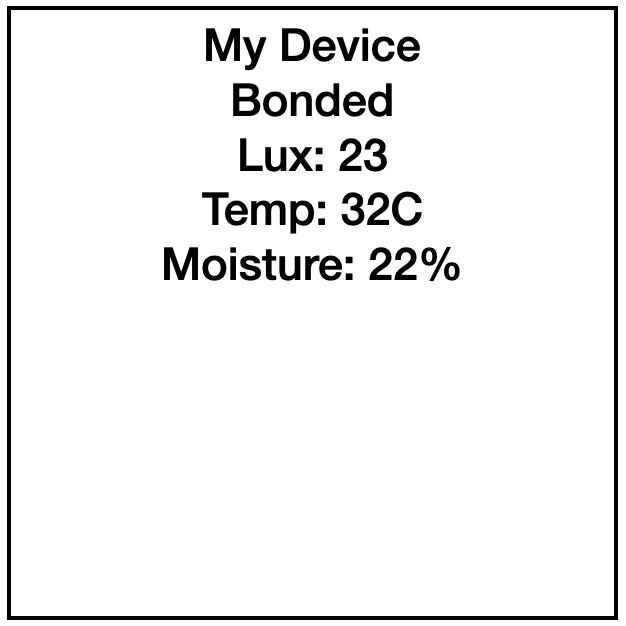
etc. References to named subsystems in the Subsystem Summary (see below) is advised.

(you can draw your block diagrams in Powerpoint or Libre office and paste them in here)

Describe how data (that you described in the data types table above) flows through the product. You must document this as a diagram.

2 Key Items to cover in this sub-section:

Division of labor: What functionality is implemented by hardware, and what functionality is implanted by software: The information conveyed in this section should describe the division of labor between hardware and software. You must document this.

Design of your LCD displays, something like:

Have one of these figures for each LCD display in your project.

>

### Subsystem Summary

<

This is a summary of individual subsystems within your product. Create names for each subsystem. Some subsystems are just hardware. Some subsystems are just software. Some subsystem are a combination of hardware and software. Describe the role/function of each of the subsystems. An example could be: Acceleration Monitoring Subsystem. This subsystem runs on a 1 second BT stack software timer. The timer ISR reads the acceleration in the Z direction using the I2C <manufacturer model#> accelerometer device and reports the measured acceleration to the Central Control software module. Remember: This is a summary. Not every single detail is required to be in a subsystem summary. Just enough to describe the basic notion/scheme/plan. Note: The plans may have to change when you get into development as you discover unknown issues.

>

### Test Plan

<

Write about how you plan to test subsystems, and test the main (top, whole, entire) system. Describe the scheme, process(es) or procedure(s) you plan to use to perform the testing. All tests that are run should result in a “Pass” or “Fail” message or notion. The test plan shall be described here and use an accompanying spreadsheet (template provided) that lists all of the tests to be run and the test result (Pass/Fail) from the most recent set of tests. **Do not paste your test plan in here**. **Instead include your test plan .xlsx file with your submission**. In the test plan .xlsx file **include what percentage of your test plan is complete at this point in the project.**

>

### Proposed Schedule

<

This is where you make estimates when subsystems and the main system will be completed AND most importantly, what order the subsystems should be completed and then integrated into the main system. For both individual and team projects, the ability to divide development into tasks that can run in parallel and/or do not create “blockages” (a blockage is where task B cannot start until task A has been completed) is important to execute your schedule on time. Each subsystem will take time to design, implement and test. In general, start with the most separable or lowest level subsystems first, as these usually can be designed, implemented and tested in a stand-alone manner. Over time, once subsystems are tested and determined to be complete and bug-free, these subsystems are integrated together to start assembling the main system. If you find that in order to test a subsystem it requires data/stimulus from another subsystem, then in order to test the subsystem, you may have to create dummy data/stimulus.

Example schedule: Say the main system consists of subsystem A and subsystem B. A schedule might look like:

>

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Student(s) Responsible | Target Completion Date | Expected Completion Date |
| Subsystem A design |  | Nov 9 |  |
| Subsystem A code implementation |  | Nov 14 |  |
| Subsystem B design |  | Nov 17 |  |
| Subsystem B code implementation |  | Nov 21 |  |
| Subsystem A and B integrated into main system |  | Nov 24 |  |
| Main system testing |  | Nov 27 |  |
| Low power requirements testing |  | Nov 29 |  |

<

Fill out the column with students name(s) responsible for the work (sometimes called milestones). Make your best estimates for target completion dates. These are usually expressed as end of day (EOD), so midnight on that date for us. Take time to think about all 3 aspects of: designing, coding and testing. Do not modify these target dates once you submit this document for the Project Proposal due date. If you experience schedule slips, you can provide updates when you expect to complete a task/subsystem etc in the “Expected Completion Date” column date for Update 1, Update 2 and the Final Report. For this Project Proposal section leave the “Expected Completion Date” column empty.

Make sure you plan time for testing and debugging. This can take longer than you think if you run into challenging bugs.

>

<Project Proposal 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 create and keep up-to-date a project schedule page that reflects the “live” up-to-date schedule of your project in your GitHub repository following this example: <https://github.com/CU-ECEN-5823/ecen5823-courseproject-Sankalppund> The schedule updates in your deliverable PDFs shall be snapshots (where you were at the time of deliverable submission) of your live schedule on your wiki page.

GitHub repository URL(s) = <URL to your GitHub repository, can be an empty or filled repository at this point in time.>

## Section 2 - Update 1

<Use this section to provide update 1 content>

<Each section shall be authored (w

Section Author: Sayali Mule

### Status

<

This is where you write text to describe whether the project is on or off schedule. Describe challenges and/or issues you have encountered. Most projects encounter unknown issues or blockages that end up delaying completion dates. Discuss what, if any, design changes had to be made and why. Talk about that here.

Then 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. If the target date for a row is still valid, enter that date. If the target date for a row has slipped (has or is taking longer), entered the new expected completion date. If a row becomes invalid due to a design change, leave the row in the table and note that by using strike-through text on the full row. Update the “Expected Completion Date” column with Removed. If rows were added to the table, note that by adding the text Added in the “Target Completion Date” column.

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

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

>

<Update 1 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>

## 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:
   1. When system boots up, the system would be in low power mode and in sleep state
   2. 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
   3. Now, its responsibility of the user to clear the value sent to client in order to detect the next free fall event.
   4. 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 eventhough 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 falls 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.

|  |  |  |
| --- | --- | --- |
| **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:

Diagram

Description automatically generated

<Update 2 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) = [**https://github.com/samu7988/ECEN5823\_Project\_Health\_tracker**](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 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 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.>