Project Deliverable G

GNG 1103

Team D6

“Diamond Hands”

Group Members	Student Number
Connor Harper	300166870
Jason Clapiz	300172134
Leo Tan	300018447
Karsten Lowe	300141177

March 14, 2021

Faculty of Engineering - University of Ottawa

Table of Contents

1.0 Introduction

In our 3rd client meeting which took place on March 12, 2021, we showcased our first and second prototypes, and also our future plans. To summarize our progress so far, our first prototype validated the data communication pathway and our second prototype implemented our desired method of recording data with an MPU6050. With both of our core elements testing successfully, we are well-positioned to accurately test our final prototype in order to determine a quantitative interpretation of violent shaking before design day. The client's general message for us was to execute our remaining plans as we proposed. JAMZ was also beneficial in clarifying some remaining details regarding the desired response of the final module. For example, JAMZ identified that the ideal output from our violent shake alarm will include an interpreted status alongside a visual element such as a graph.

Included in this document is an overview of our estimated schedule for our second prototype and a summarized analysis of our test results. Our proposed prototyping plan will be updated following additional feedback from JAMZ on our subsequent prototype and test results.

2.0 Prototyping Test Plan

Table 1: Prototyping Test Plan

Test ID	Test Objective ~Why~	Description of Prototype used and Basic Test Method ~What~	Description of Results to be Recorded and how these Results will be used ~How~	Estimated Test Duration and Planned Start Date ~When~
2.1	To read values off the MPU 6050 in the arduino	Physical prototype, MPU 6050 and Arduino open on a computer	Information is being received and printed in serial monitor when the MPU 6050 is moved	1 Day 2021-03-09
2.2	To interpret the values of the MPU6050 into degrees per second	Physical prototype, MPU 6050 and Arduino open on a computer	Data in serial monitor reflects somewhat closely to expected values	1 Day 2021-03-09
2.3	To transfer MPU6050 data to the Raspberry Pi	Physical prototype, MPU 6050, Arduino and Raspberry Pi, computer open on	Values found in test 1.2 are shown in serial monitor on the Raspberry Pi side	1 Day 2021-03-09

		Raspberry Pi		

3.0 Prototype Test Analysis

3.1 Test Fidelity

Table 2: Test Fidelity Analysis

Test ID	Test Fidelity (1-5)	Clarity of the Signal (1-5)	Justification
2.1	2	5	While the raw data is clear and is the exact input we need, it is not usable in its raw form. This test is a focused test and as a result the clarity of the signal is very high. The fidelity of the test is low because the data we are receiving is not very applicable to our needs, it needs to be modified first in order to be meaningful.
2.2	4	3	Test 1.2 has a higher fidelity than test 1.1 as it converts the raw data into fairly accurate real world data that can then be used for what we want. This test is also focused however as a result of the data being modified, the clarity of the signal is reduced as the data itself will never be as accurate as its raw form.
2.3	5	3	Test 1.3 is a comprehensive amalgamation of test 1.2 into the full design. The data itself is the same as test 1.2 and hence the clarity of the signal is exactly the same as test 1.2. Because of the comprehensive nature of the test and its closeness to our intended final design, it has a very high fidelity rating.

3.2 Test Observations and Results

Table 3: Test Observations and Results

Test ID	Observations	Results
2.1	Upon tilting the MPU 6050 in different directions, values appeared in the serial monitor and changed with the amount of tilt in the MPU 6050.	The test of reading values from the MPU 6050 is successful. Hardware connections are correct and data is being successfully recorded in the Arduino.
2.2	Degrees of tilt observed in the serial monitor reflect expected values. Return to zero, 90°, 180° tests all perform well.	Data being recorded in the arduino is now calibrated appropriately and data values are consistent with real world expectations.
2.3	Data is being shown on the serial monitor of the Raspberry Pi through the UART interface.	Data is successfully being recorded, interpreted and passed along to the Raspberry Pi and our core system proof of concept is successful.

4.0 Conclusion

This prototyping cycle focused on the MPU 6050 and getting relevant data from the sensor to the Raspberry Pi. This cycle was broken down into 3 testing phases each with its own significance to our final design. Test 2.1 was our proof of concept of design, the idea was to simply read some values from the MPU 6050 in the serial shell and to see values changing when the sensor was moved. We found online libraries of others who completed similar projects in the past and adapted one to fit with our hardware, once we had values reading in the serial monitor we knew we had completed test 2.1.

Test 2.2 consisted of understanding and modifying the code we found so that it would reflect expected real-world values. This consisted of modifying offsets and calibration methods so that the chip would always remain consistent and would self-calibrate each time it was powered on. We were able to create two versions of the code that would calculate values in degrees per second and degrees relative to its original position. We want to ensure that we will be able to deliver the data that JAMZ requires and after the meeting and presentation we will know what approach we will develop further on.

Finally, test 2.3 consisted of transferring the data from the Arduino to the Raspberry Pi. This test was the most straightforward as our previous test 1.2 already solved most of the issues regarding the transmission of data over the GPIO UART connection. After a small amount of testing, we were able to read the values from the serial shell of the Raspberry Pi in real-time as the MPU 6050 is moved about. Overall our core subsystem of the MPU 6050's data is now complete and we are aiming to further polish this system as well as introduce more quality of life features to the violent shake alarm during the next prototyping phase.

5.0 Appendix

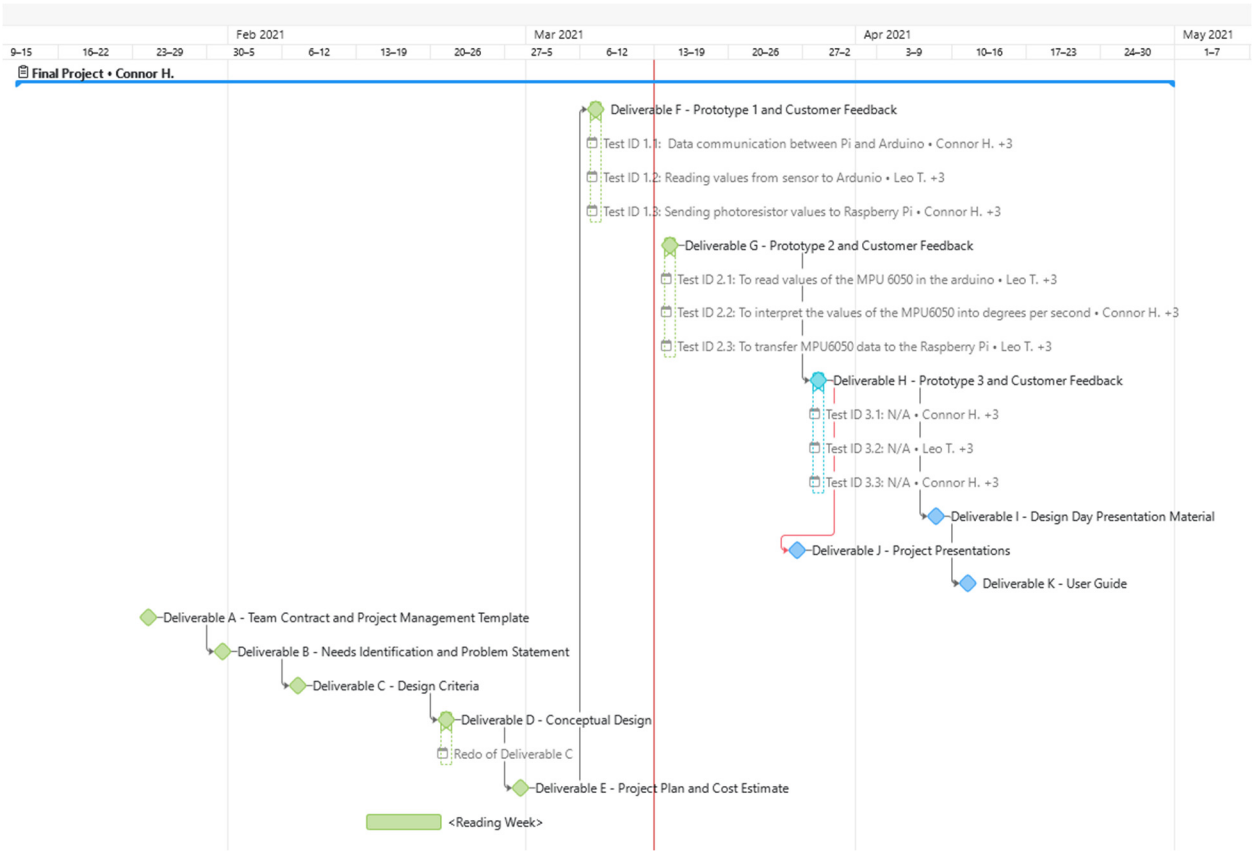

Fig 5.1 Project Development Timeline