Project Deliverable E

GNG 1103

Team D6

“Diamond Hands”

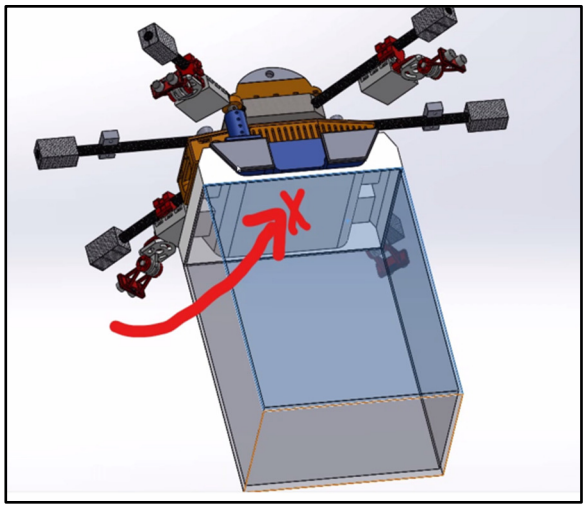
Group Members	Student Number
Connor Harper	300166870
Jason Clapiz	300172134
Leo Tan	300018447
Karsten Lowe	300141177

February 28, 2021

Faculty of Engineering - University of Ottawa

Table of Contents

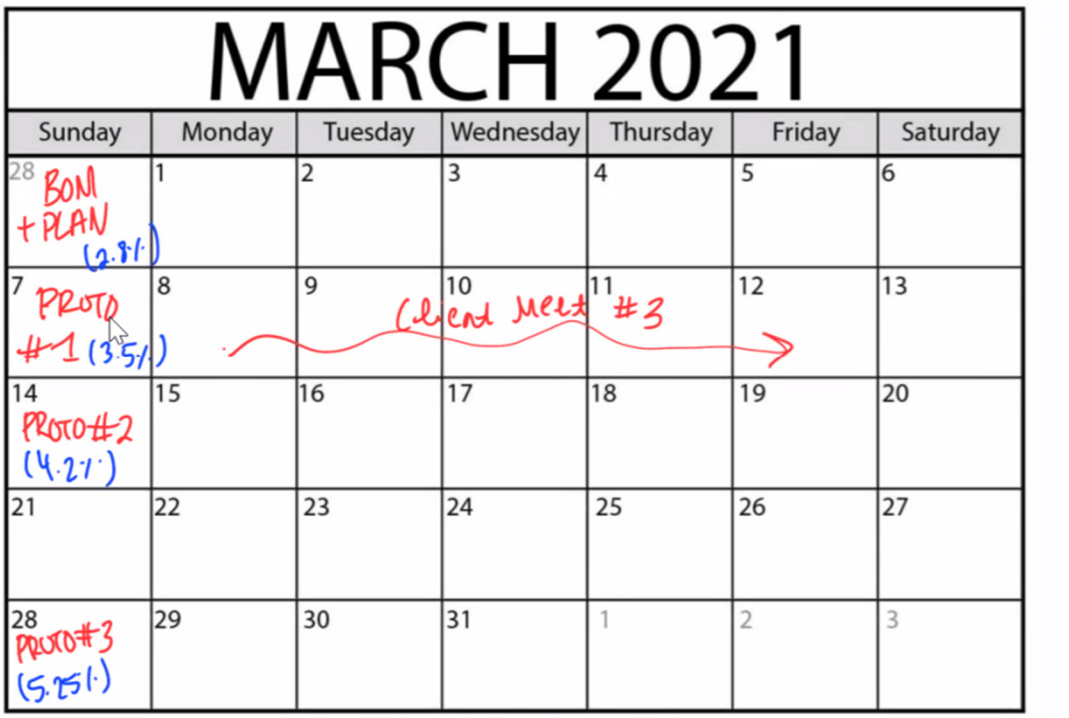
1.0 Introduction



After having our second client meeting with JAMZ on February 22nd, 2021, we answered some very key questions relevant to their needs. Primarily, we converged on a wired solution due to the increased reliability, simplicity, and quality of data communication. This allowed us to decide on the design concept which we are going to pursue, being a wired module that can detect violent shaking. Our module will be housed underneath the drone and above the package as seen in the Figure.

Included in this document is the bill of materials, the circuit diagram, and our estimated prototyping schedule and plan. JAMZ's feedback based on our general concepts was beneficial towards narrowing our focus regarding our proof of concept for prototype 1. Also identified in this document are the potential risks which could arise during the development of our final module.

2.0 Schedule



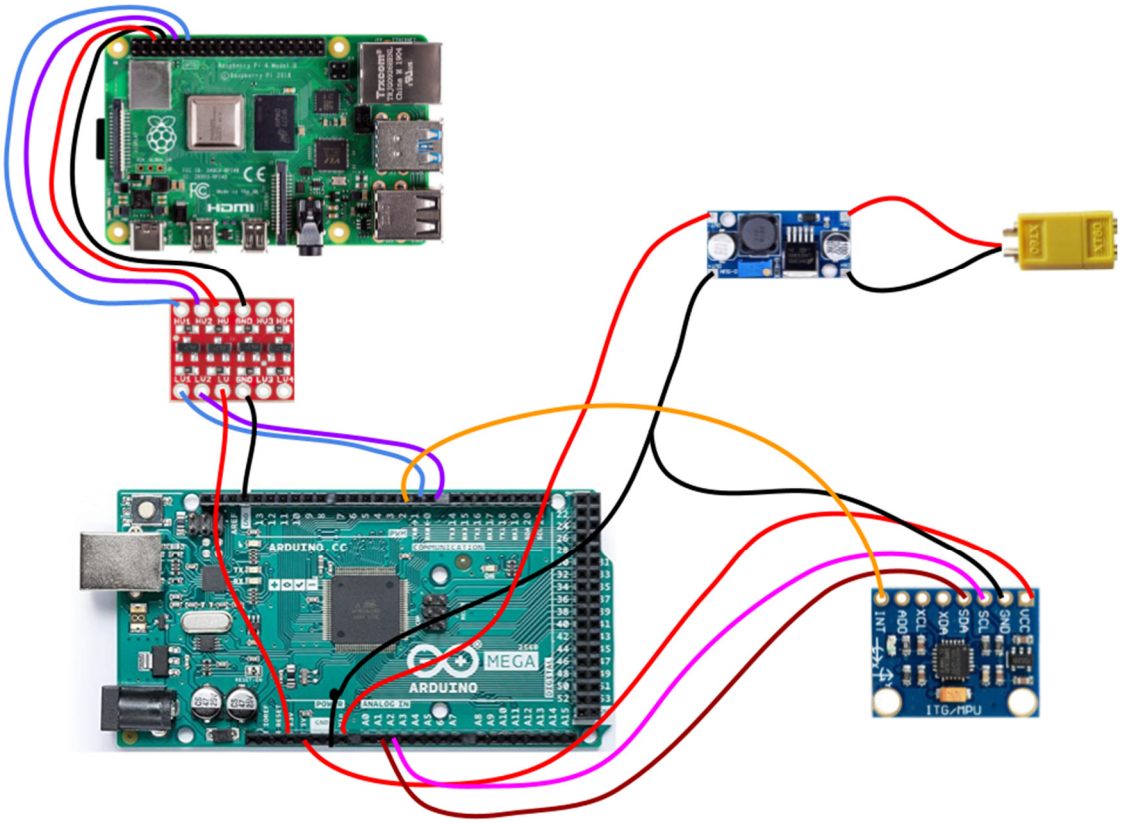
MARCH 2021

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
28 BOM + PLAN (2.8)	1	2	3	4	5	6
7 PROTO #1 (3.5)	8	9	10 Client Meet #3	11	12	13
14 PROTO #2 (4.2)	15	16	17	18	19	20
21	22	23	24	25	26	27
28 PROTO #3 (5.25)	29	30	31	1	2	3

3.0 Bill of Materials

Item	Purpose	Link	Cost	Status	Comments from students	Comments from TA/Pm
Wires	Power and data communication	Amazon - Multicolored Wires - 120 pcs	$3.13	Need	Shipping length: One week	
Buck Converter	Step down power from main battery on drone	Amazon - Buck Converter	$15.20	Need	Not needed in prototype, exclude price from total amount	
Logic Converter	Facilitate communication between arduino and raspberry pi	Amazon - Logic Converter	$9.95	Need	Shipping length: One-day	
MPU 6050	Gyroscope/ Accelerometer	Amazon - MPU 6050	$3.32	Need	Shipping length: One-week	
Arduino Uno	Computer	Amazon - Arduino Uno	$32.99	Completed	Product is already available, do not add to cost needed to complete project	
Raspberry Pi	Computer	Amazon - Raspberry Pi	$53.96	Completed	Product is already available, do not add to cost needed to complete project	
XT60 Connector	Connect to main onboard power	Amazon - XT60	$9.99	Need	Shipping length: One-day	
PLA Filament	For 3D printed housing		$5.00	Completed	Do not include price in final amount	
Shipping					Availability is largely considered	
Total anticipated cost			$16.39		Total anticipated costs includes Tax	

4.0 Diagram



Device/ Connections	Arduino Mega	Raspberry Pi 4	MPU 6050	Buck Converter	XT60
Connections (By colour)	Power Ground Data Transmission Data Transmission Data Transmission Data Transmission Data Transmission Data Transmission	Power Ground Data Transmission Data Transmission	Power Ground Data Transmission Data Transmission Data Transmission	Power Ground	Power Ground

Data communication between the Arduino Mega and Raspberry Pi 4 is facilitated by the logic converter. This is a necessary component as it takes the Arduino's operating power of 5 volts to the Raspberry Pi's operating power of 3.3 volts. The MPU 6050 is wired directly to the Arduino Mega

VIA 2 analog data cables and a single digital data cable. The data from the MPU 6050 is taken to the Arduino and then passed on to the Raspberry Pi via the logic converter. The whole system is powered by the drone's onboard battery through the XT60 power connector and stepped down to a usable power level by the Buck Converter from 44.4 volts to 5 volts.

5.0 Prototyping Plan

5.1 Prototype 1

For the first prototype, we want to have all of the necessary electronics available to be connected to make sure that each electronic component is properly communicating with each other. We can just use a 9V battery to simulate the power delivery from the drone's main battery and step it down through a buck converter. We can also modify some existing code from the readily available MPU 6050 libraries to make sure that the MPU 6050 data is being transmitted to the Arduino and then to the Raspberry Pi.

Table 1: Task List For Prototype 1

Task	Members Assigned	Estimated Time (Days)
Order and receive all components	Jason	7-14
Test if components are working	Jason	2
Plan out data transmission code	All team members	1
Compile readily available MPU 6050 libraries	Connor	1
Compile readily available Arduino to Raspberry Pi communication method	Connor	1
Plan for physical test	All team members	1
Physical test for proof of concept	Jason	1-2
Analyze results from physical test	All team members	1-2

5.2 Prototype 2

Given that the first prototype is fully completed without any delays, for the second prototype, we want to be able to modify the code again to make sure that the Arduino is transmitting required (useful) data to the Raspberry Pi, instead of every single data. Because there is not a definitive metric to measure "violent shaking", we would have to do various tests with the MPU 6050 to define the conditions of the ranges that constitute a high amount, a moderate amount, and little to almost no violent shaking. Required data that can be selected for to be sent to the Raspberry Pi can include the data when the package has experienced a high amount of violent shaking or a moderate amount of violent shaking. Useless data that could be ignored would include the data when the package has experienced little to almost no violent shaking.

Table 2: Task List For Prototype 2

Task	Members Assigned	Estimated Time (Days)
Adjust code to test for ranges of violent shaking	Connor, Karsten, Leo	1-2
Plan out test for the ideal ranges of data for our module	All team members	1
Physical test to figure out the ranges of violent shaking	Jason	3-5
Analyse the data of the ranges of violent shaking	All team members	3-5
Re-adjust code to match the ranges of violent shaking	Connor, Karsten, Leo	3-5
Test the readjusted code to confirm Arduino will send required data to Raspberry Pi	All team members	2-4

5.3 Prototype 3

Given that the second prototype is fully complete without any delays, for the third prototype, we want to have our code fully working without bugs, our electronics fully, correctly wired up, and have our electronics strategically placed in a housing case that is nearly identical to the drone's housing case. Once everything is in place, we can do final testing in an environment to simulate the terrible conditions that a drone might encounter in the real world that would show a package being violently shaking.

Table 3: Task List For Prototype 3

Task	Members Assigned	Estimated Time (Days)
Complete wiring of full project	Jason	2
Ensure code is running as intended	Connor, Karsten, Leo	14
Ensure sensors are reporting data as intended	All team members	14
Finalise any features in the software	Connor, Karsten, Leo	14
Final test to simulate violent shaking of a drone	All team members	1
Print housing for sensor	Jason	1

6.0 Risk Analysis

As we develop our violent shake module, it is inevitable that there will be problems and obstacles which we will encounter and have to be prepared for. The most important risks to prepare for are poor estimation of timelines and/or unaccounted for inefficiencies. This is very important because we are following a strict deadline with JAMZ and we must have a working, presentable module by April 8th, 2021. This places added priority to our time management and organization of tasks to ensure we meet our deadline. This being said, we foresee 3 potential issues which may have a likelihood of occurring, and affecting our timeline:

- The first issue with the potential of slowing our progress is the general inconsistency with shipping times due to COVID-19. If we are unable to receive our necessary parts in time this is the most detrimental risk to our ability to present a working module on time.
    - In the event we encounter issues with product shipping, substitutes for some components can be made, for instance an ultrasonic sensor in place of the MPU 6050. As long as we can show proof of concept of the data transfer, everything else will eventually work as initially planned. Should we not receive items such as the logic converter, we would be forced to use a USB cable for our proof of concept.
- A second issue which could affect our timeline is the event where we receive faulty or poor quality parts. This would affect our ability to build our fully functioning module by the deadline.
    - The response for this kind of potential issue is similar to that of not receiving the part in the first place, we would be forced to find a replacement and/ or downgrade the level of quality temporarily such that we can show some proof of our ideas working for the JAMZ team.
- The final risk which has the potential to significantly impact our timeline is the situation where we are unable to implement all of our desired features into the code in time. If this happens, we will not be able to present our initially proposed design by April 8th, 2021.

- In the event we cannot finalize the desired code as initially proposed for prototype 3, we will have to remove certain features which we perceive to not be as relevant to the core functionality of our module. This will ensure that we have a fully working design to present by the deadline.

7.0 Conclusion

Following our meeting with JAMZ we have settled on some solutions for the violent shake alarm. After asking questions and receiving more critical information on hardware specifications, software and clarifying some ambiguous information, we are moving forward with a wired, sensor on hook design. This style of design will utilise the best of both the consistency of wired data transfer as well as the accuracy of the data coming straight from the hook itself as opposed to from the drone.

We have created a bill of materials of the specific components we wish to use for our solution. These components have been researched and have a track record of consistency and quality we aim to achieve with our solution. Our schematic shows how we plan on physically connecting all the components. All of these components would eventually fit into the drone and hook mechanism but we are unable to simulate those environments for the foreseeable future.

For our upcoming prototyping phases we want to first create a proof of concept for all our general subsystems. Stepping down the power level, transmitting data from the Arduino to Raspberry Pi etc. Our second prototype phase will focus solely on transmitting critical data from the MPU 6050 to the Raspberry Pi from the Arduino, while analysing and filtering out unnecessary data. The last prototype phase will focus on implementing all of the subsystems into a final working product, encapsulating all the key elements that we have identified in previous deliverables.

We identified some potential risks and how to mitigate them. Essentially, we would have to substitute/ replace parts that don't arrive on time, arrive but are damaged or don't arrive at all. In the event we are not able to fully implement all of our desired features into our software, we will have to cut some features of the code or remove less critical non software related parts from our prototyping phase.