***Objective:***

Develop your first prototype and devise a test plan for your second. Get customer feedback to improve your prototype.

***Instructions:***

1. Clearly outline the feedback received from your client on the group concept or detailed design. Specify how the feedback will be used to inform future design choices and improve the solution.
2. Develop a prototype which will be used to achieve the objectives your team has set out in the plan created in the last deliverable (i.e. you need to answer the “why”, “what” and “when” of prototyping).
   1. Remember: a prototype is not normal work on your project, it is something that has a smaller, targeted objective with specific tests and measurable results.
3. A simple analysis of critical components or systems should also be included, based on your current knowledge of engineering science or other knowledge.
4. Carefully document your prototyping test plan, analysis and your results (including detailed images of your prototype).
5. You must gather feedback and comments on your ideas and prototype from potential clients/users that you have sought out and identified on your own.
6. If applicable, update your target specifications, detailed design and BOM after tests are completed and analyzed.
7. Finally, teams will outline a prototyping test plan based on the template provided in “Lecture 11 – Prototyping Test Plan” to prepare to build the second prototype in the next deliverable.
   1. Typical objectives include: communicating and getting feedback for ideas, verifying feasibility, analysing critical subsystems or system integration or reducing risk and uncertainty.
   2. You must also define a stopping criteria which will allow you to end the test once you are satisfied that you have achieved your testing objectives.
   3. Be very clear about what you are trying to measure and define an acceptable fidelity based on the objectives of your prototype. See <https://en.wiki.makerepo.com/wiki/Professional_development/Design_thinking/Design_for_manufacturing>.

Since this will be your team’s first prototype, you should focus on creating a basic proof of concept which should be made using materials and components that cost very little (e.g. things found around the house, scraps, etc.). Get creative in order to improve your results.

It is strongly recommended that you start early, as prototyping takes a significant amount of time.

***Task Plan Update:***

1. Update your Wrike task boards to include any changes in estimated task duration, missing tasks, task responsibilities, milestones, or dependencies, based on your better understanding of the project or based on feedback that you have received from your PM/TA.
2. Include more detailed sub-tasks for the tasks that will need to be completed over the next few weeks.
   * *Important note*: It should be possible for ONE person to complete each identified task or sub-task in the allotted time. The allotted time should also be *reasonable*, based on the task owner’s availability. Everyone should be doing their fair share of the work.
3. Verify and update the task start dates and end dates for each task, based on your project progress.
4. Ensure that you have taken into account each team member’s *actual* availability over the next two weeks, as well as significant events, such as particularly high course loads, exams or travel, which might be going to limit actual project work progress.
5. For *each* person in your group, it should be possible to determine:
   * What was completed last week (i.e. “**Completed**” tasks),
   * What will be done next (i.e. “**In Progress**” tasks)
   * If tasks are going to be put “**On Hold**” or “**Cancelled**” altogether
6. Any and all group “Issues” should be discussed and dealt with, ideally with the assistance of your Project Manager (PM). This should happen during **each** of your lab sessions or can happen earlier, using your defined communication methods. As already explained, it is essential to keep your PM/TA “*in the loop*” throughout the term. It is usually *not* a good idea to ignore conflicts between team members. Instead, you should deal with them in a constructive way.

***Submission:***

Each team (***only one person from each team***) must submit a PDF copy of this deliverable by uploading the file as an attachment into Brightspace.

Shane’s feedback

* Shane mentioned that pressure sensors used in the past would suddenly stop working, so to address this issue we have all made sure to get a good quality water pressure sensor to avoid future problems from occurring
* Shane wanted to keep costs down as much as possible and is more than willing to have his team input the volume going into the tank and the depth/size of each batch. Therefore, we will use less sensors and more employee input for calculations to be done.

Current prototype

* Our current prototype involves a liquid/air pressure sensor installed inside a water bottle cap to give us the exact values of how large the water pressure is inside a water bottle
* The objective of prototype 1 was to prove that we can accurately use a sensor to calculate the pressure of a liquid in a tank. Our prototype succeeded.

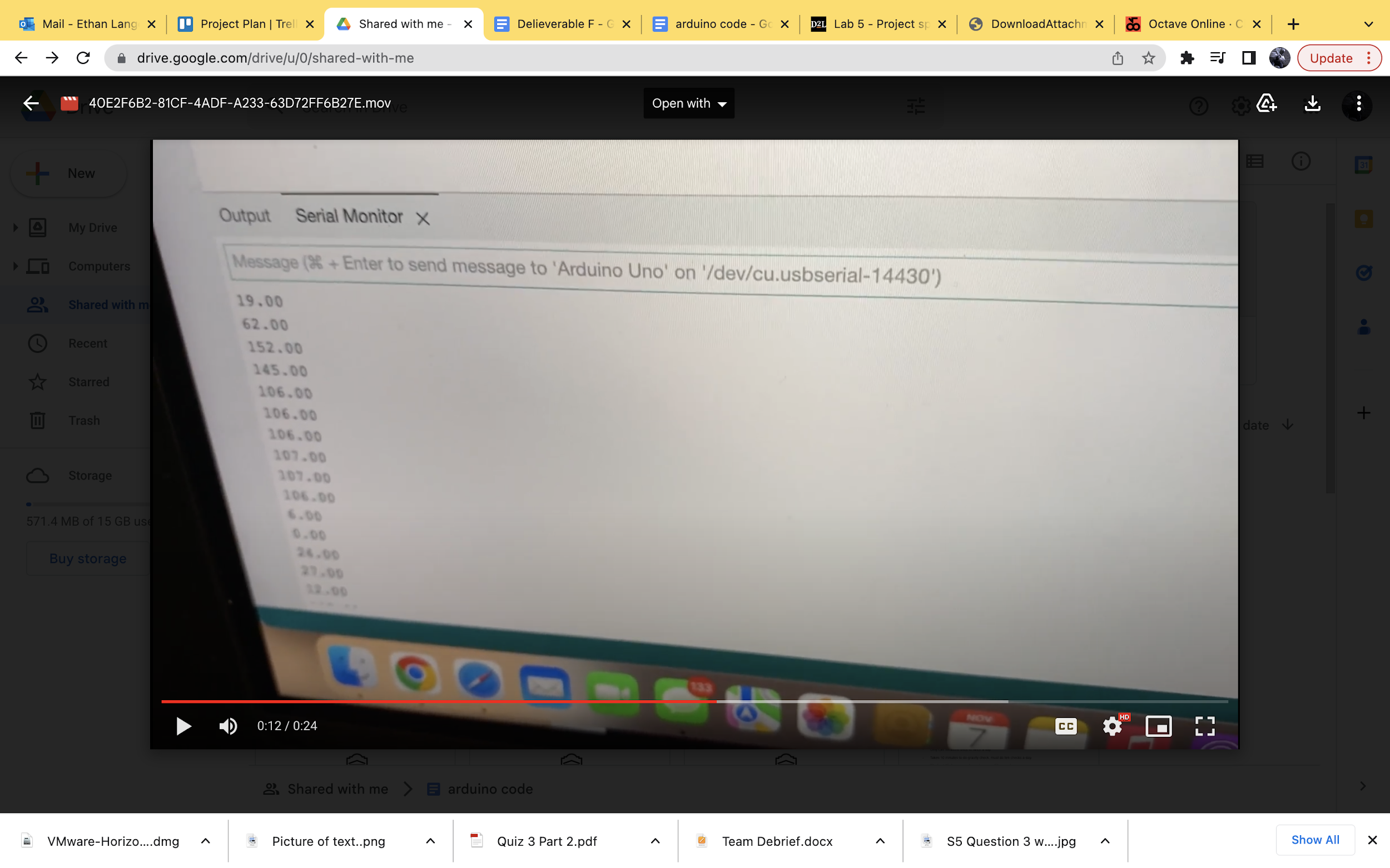
Equipment

* Water pressure sensor
* Water bottle
* Arduino UNO
* Wires
* Computer
* Tape
* Bread Board

Test Plan:

1. Create a simple circuit and code using arduino. It should simply receive and record the readings of the pressure sensor every 2.5 seconds.
2. Put the sensor head through the cap of the water bottle.
3. Tip the bottle upside down so the sensor is experiencing the water pressure.
4. See if the readings of pressure stay consistent.

Picture of Data from test video. You can see that for the 15 second period when the bottle was upside down, the pressure stayed consistent at 106-107.



Next prototype

* Find a way to test water pressure without a big mess of water on the floor
* Will attempt to calculate the specific gravity with an accurate result.
* We will install a screen that will display all current pressure values.

Video of test:

This is a demonstration of our prototype in action. You can see the pressure sensor actively reading values to us.

<https://drive.google.com/file/d/1THAbaTUI5QVgUWSC7dpDbqMoGDmd0KMA/view?usp=share_link>

Code:

This is the code used to print values from the pressure sensor to the serial monitor.

const int pressureInput = A0;

float pressureValue = 0;

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

}

void loop() {

// put your main code here, to run repeatedly:

pressureValue = analogRead(pressureInput);

Serial.println(pressureValue);

delay(2500);

}

Trello:

