Progress Report – Week 5

02/06/2018

Belvis Devise INC – Division B:

John Heyniger, Nate Porochonski, Zachary Smith. Deepak Warrier

Instructor – Dr. Ratcliff, GTA – Chris Chang

Contents

Backwards Looking Summary

Situation:

Over the past four weeks, Division B has been getting familiar with the major systems and components that are available for use in the construction of the Advanced Energy Vehicle (AEV). The AEV consists of the following base components: The LiPO Battery, motors, the Automatic Control Unit (ACU), and the monorail mount (Images of components in Appendix _). The "brain" of the AEV is the ACU. The ACU consists of an Arduino Nano and a motor controller board custom designed for the AEV. The Arduino is programmed using a mixture of C/C++, which is rewritten to C++ during compilation (Images of the ACU in Appendix _).

The first week was spent on learning the basics of the programming library of the ACU. To learn the library's commands, the programming team read through the documentation of its commands and designed code to operate motors attached to a rig, designed to only hold motors in place (Code and Rig Design in Appendix _). The library's commands are the most efficient and direct way to communicate with the AEV's ACU, and it is important to understand the nuances of the commands' syntax and semantics in practice because doing so prevents the event of unexpected actions of the AEV, such as speeding up too rapidly, or not moving at all. The second week spent on understanding the reflectance sensors and how to properly operate them. To understand the reflectance sensors, the user manuals were referenced prior to running a utility to debug the sensors on the AEV (Sensor placement and Associated Code in Appendix _). A complication, however, came up during the lab. The sensors weren't reading the supposed values. Instead of counting up or down, based on the AEV's direction, the sensors only counted up. The reason for this error was that one of the AEV's sensors wasn't flush with the wheel of the AEV, and once that was fixed, the sensors worked fine. The reflectance sensors are important because they allow the AEV to keep track of where it is relative to its starting position. The third week was spent on getting familiar with the Data Analysis tool – a utility that tracks Power vs. Time and Distance. The utility is useful for understanding the efficiency of a design, and if there's any performance bugs, the graphs can show when, and at what power, the bugs occur. To test the Data Analysis tool, a simple program was drafted, and its data pulled to be displayed in the tool's graph (Both graphs and associated code in Appendix _). Week four was spent on reviewing the Design Screening and Scoring Matrices. These systems of Design Analysis are useful in pointing out design flaws and pros and deciding on how to combine and use the features of the designs in question to produce a final design for end product.

Results and Analysis

Programming Basics:

When learning and testing the commands of the AEV's Command Library, a simple program was used to test the motor's functionality and how the commands work (Associated Code in Appendix A). With the first couple of steps of the program, the motors didn't seem to spin, although there was a noise coming from the motors and the code said for the motor to run. The reason for this issue is that the power during these steps (15% - 25%) is not enough power to start turning the motors. As a result, the motors were struggling to turn at the low current. The commands perform very specific actions. For example, the brake command says to stop the motors from moving, and that's all it will do. During the test, it became evident that the command did not imply that it would stop the AEV's motion. This is something to keep in mind for future programs and tests because the program will have to account for AEV's motion after braking the motors. Since the AEV's ACU runs the commands almost instantaneously, it is possible that the testers couldn't see if the program didn't perform as it expected. To alleviate this issue, the programmers added extra comments to keep track of each step as the AEV performed it. It was fairly easy to learn and understand the code, as there was someone on the team who was already familiar with the AEV's programming library (i.e. Arduino C++). It would be easier to teach new programmers and team members some basic programming keywords that are often used in Arduino so that they can make their own functions that can improve the functionality of the AEV in future tests.

Reflectance Sensors:

There are two commands that the AEV can use to change slow down. It can use the brake(); command and the reverse(); command. With the brake command, the power to the AEV's motors is turned off, so the only way for the AEV to slow down using this command is to rely on the force of friction between the AEV's wheels and the track. This is an unreliable way to slow the AEV down. There is no way to determine how far the AEV will go other than through testing, and it takes a while to slow down. The reverse command on the other hand is more reliable and faster than the brake command, but is still difficult to use, and is less energy efficient. Overall, there is no efficient and reliable way to slow down the AEV, making it difficult to achieve the MCR.

The biggest error with the reflectance sensor lab was correctly configuring the sensors. At first when the sensors were attached, the nuts and bolts that came with the AEV kit were used. When the sensors were first tested, the marks only counted upward, which means the AEV registered all movement as only forward, even when the wheels went the other way. After a lot of time trouble shooting and testing the sensors, we realized that one of the nuts came loose, and was between the sensor and the reflectance tape on the wheel. Once this was removed, the sensors were attached with zip ties and they worked properly. During the actual test of code, the AEV moved very little because it was only run at 25 percent power, but it did run properly. In the end, the team was able to solve all of the errors and complete the scenario. Other than the testing one of the reflectance sensors to see if it was broken, the team did not need any additional assistance during the reflectance sensor lab.

Design Thinking:

The goal of Zach's design was speed and aerodynamics. It has a triangle shaped front for cutting through the air efficiently and it was designed with three fans for speedy completion of tasks. Deepak's design was focused on aerodynamics, energy efficiency, and safety. Its main focus is a bubble-like wraparound that will drastically improve aerodynamics and energy efficiency, while also increasing the safety of the AEV. The last two designs, Johns and Nathans, are fairly similar to each other and focus on being well rounded and aerodynamic. They don't focus to much on aerodynamics but are both lightweight and cost effective. The team design will focus on the strengths of primarily Zach's and Deepak's designs, which will be aerodynamics and speed from Zach's design, and energy efficiency and safety from Deepak's design, and weight/reduced cost from Nathans and Johns designs. All of these design focuses help achieve the MCR by making quick and safe travel available to the linden area at a low price. During the brainstorming period, we used the attribute listing and construction techniques to help create ideas for the design process.

The main materials that will be used are plastic and metal, which comprise a majority of the parts in the AEV kit parts. The metal will add strength to the designs, while the plastic will help keep the designs lightweight and cost effective. The only parts at the moment that the team would need to fabricate are the front pieces on each design that have the purpose of making the designs more aerodynamic. These pieces would have to be light weight and carefully fabricated for the best results.

Data Analysis Tool:

The Data Analysis tool is a MATLAB-based GUI that downloads data from the AEV's ACU and graphs the data according to Power vs Time and Power vs. Distance. Division B designed sample code to test the functionality of the tool (Sample Code for Test in Appendix _). The graphs generated by the tool are also in Appendix _ for reference. Each section of the graph represents a step in the code. In the Power V. Time graph, the first five seconds of the test were spent accelerating, as shown by the steady increase in power between zero and five seconds. The straight line from five to six seconds is represented by the straight line. In this graph, the straight lines represent any instance of constant speed, which is what the goFor(); command indicates. The spike in power is from the reverse(); command in the sample code, since the motor has to generate significantly more power to stop the motor's initial inertia and start it in the opposite direction. The drop in power represents the brake(); command, which cuts all power to the motors. The Power v. Distance graph also shows the same trends; however, they are much less noticeable due to the fact the commands did not cause the AEV to move a significant distance. We later realized that the AEV's motor terminals were wired opposite to each other, causing the propellers to spin in opposite directions.

The Data Analysis tool allows us to see how the AEV is using its power, and how it relates to the code. The team can analyze the graphs between different designs in future tests and see the power consumption between each of them.

Screening and Scoring:

For the most part, the AEV sample code test was anticlimactic. The AEV didn't move much at first when it was accelerating forward. It started to move more when the motors were reversed, at which point, the AEV slowly began to move backward. The balance of the AEV during these tests was not ideal. Due to size and shape constraints of the current AEV model used during the test, the battery had to be mounted sideways from the AEV. This causes the battery to extrude out of one side of the AEV, making the design unbalanced on the track. It is not pressing issue during the testing phases of the project, but it will defiantly be resolved in upcoming design changes to the AEV.

In order to better determine which of the teams four AEV designs would be the focus of future research and development, a concept screening matrix and a concept scoring matrix were developed for all four designs. The concept screening matrix is a general chart meant to narrow the pluses and minuses of each design as compared to a reference AEV design. Once this chart is created, a concept scoring matrix can be created to analyze certain aspects of each design, taking into account how much weight each of those aspects should have on the final design.

Now the specific strengths and weaknesses of each AEV design will be mentioned as compared to the reference AEV design (Check the Appendix for the Screening/Scoring charts for more info). The goal of Zach's design was speed and aerodynamics. Overall, it was better than the reference design in aerodynamics, energy efficiency, and speed, similar in safety, and worse in the weight category. It has an overall score of 3.2 and is going to be carried forward in the design cycle. Deepak's design was focused on aerodynamics, energy efficiency, and safety. Overall, it was better than the reference design in aerodynamics, energy efficiency, and safety, similar in speed, and worse in the weight category. It has an overall score of 3.7 and is going to be carried forward in the design cycle as well. The last two designs, Johns and Nathans, are fairly similar to each other and focus on being well rounded and aerodynamic. Both designs are better than the reference design in aerodynamics, but are similar in every other category. Both designs are not being continued in the design process.

Takeaways

1) AEV - Since the current commands available don't offer consistent stopping methods, different propulsion methods are need to be researched.

2) AEV – The polarity/direction of the motors must be checked along with the wiring of the motors. When the wiring of the motors was redone, their polarity also switched which was confusing to the team at first.
3) General - The DAT will be used more in future test. It is the most accurate and scientific way to analyze how changes to the AEV affect performance.
4) General – The project is really open ended with a lot of variables that can affect the success of the AEV. The team needs to stay consistent with what it tests in order to have the most success.

Forwards Looking Situation

Situation:

The following weeks of Division B within Belvis Devise INC will focus on continuing to further develop an efficient and safe AEV design via advanced R&D testing and effective project organization. Group B's research and development will focus on two areas: material weights and aerodynamic design. The results of these tests will coincide with other Belvis Devise INC group data to produce a safe, efficient, and cost effective AEV. The research will compose of testing various design(s) and analyzing the data via software analysis and scoring/screening complexes. This will be done to effectively eliminate poor ideas and develop the most promising aspects of the AEV.

A second aspect of the coming week(s) will be effective project management and presentation. Effective management is important as it keeps the project on track and keeps the group's timeline on pace. This will be executed through committee meetings and grant proposals, as well as oral presentations and reports. Team meeting minutes will also be continually implemented to maintain integrity. This combination of R&D as well as project management will yield effective progress on the MCR in the following weeks.

Weekly Goals:

1) Decide which materials would be optimal to use in terms of weight and aerodynamic properties. (By Feb. 23)
2) Use the concept screening and scoring matrices to determine a final design for our team. (By Feb. 23)
3) Eventually have a safe and functioning prototype that can run on the track. (By March 2)

Weekly Schedule:

All Members Present

Week of:

- 2/11/18 - Grant proposal and Vote, Committee Meeting, Continue R&D
- 2/18/18 - Research and Development midpoint, nearing conclusion on designs
- 2/25/18- Final R&D presentation, Website Update 3

Appendices

Appendix A – Programming Basics

Sample Code for Familiarizing with Programming Library of AEV:

Reference Appendix G – Programming Basics section for sample code used during this lab.

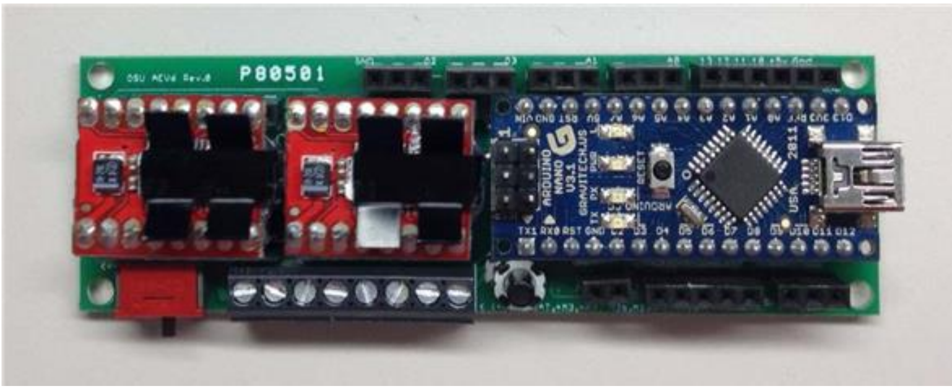


Figure 1: Image of Automatic Control Unit

Source: Lab Manual Preliminary Research and Design

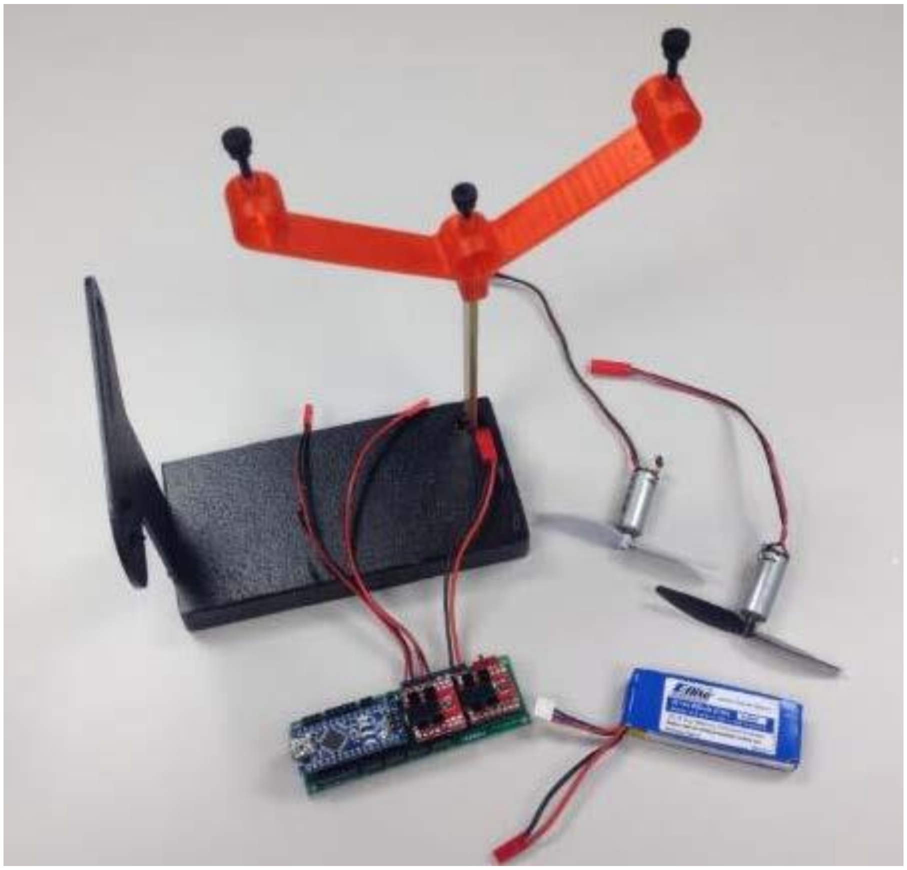


Figure 2: Image of Motor Rig

Source: Lab Manual Preliminary Research and Design

Appendix B - Reflectance Sensors

Code for Testing Reflectance Sensor Functionality:

Reference Appendix G – Reflectance Sensor Testing section for sample code used during this lab.

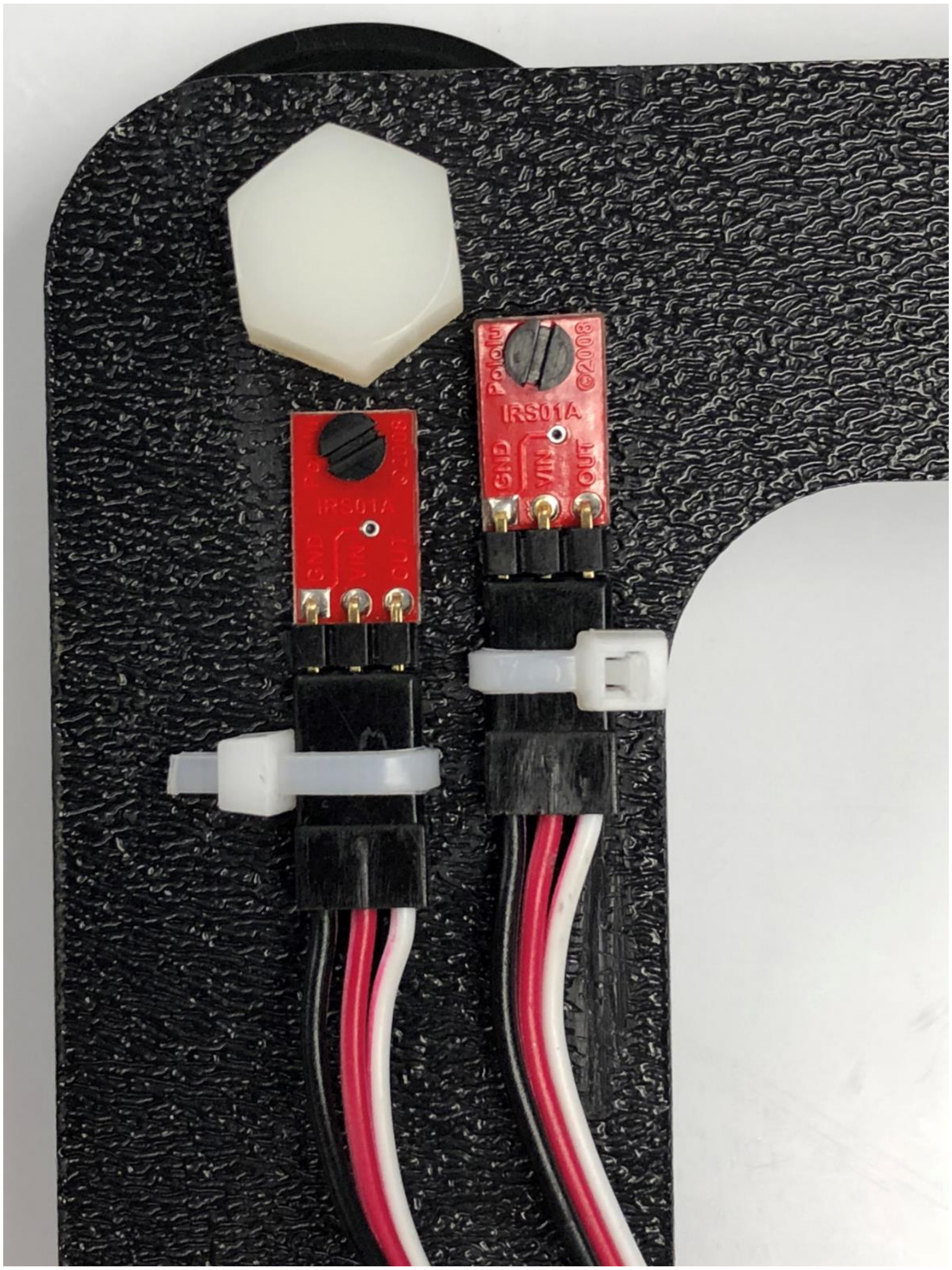


Figure 3: Image of Reflectance Sensors Attached to AEV

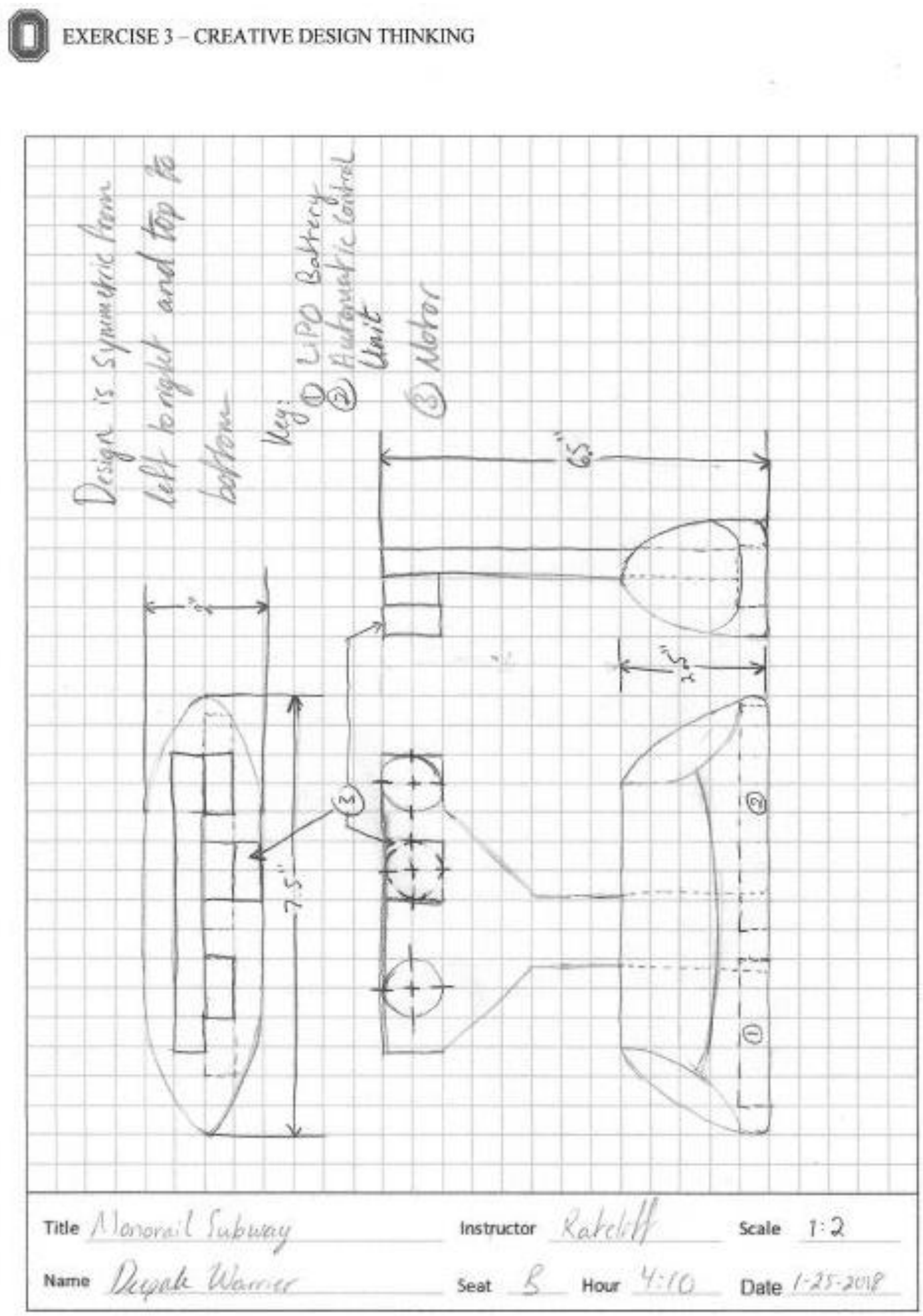
Source: Zach Smith

Appendix C - Design Thinking

Deepak's Design

Estimated Weight – 450 grams

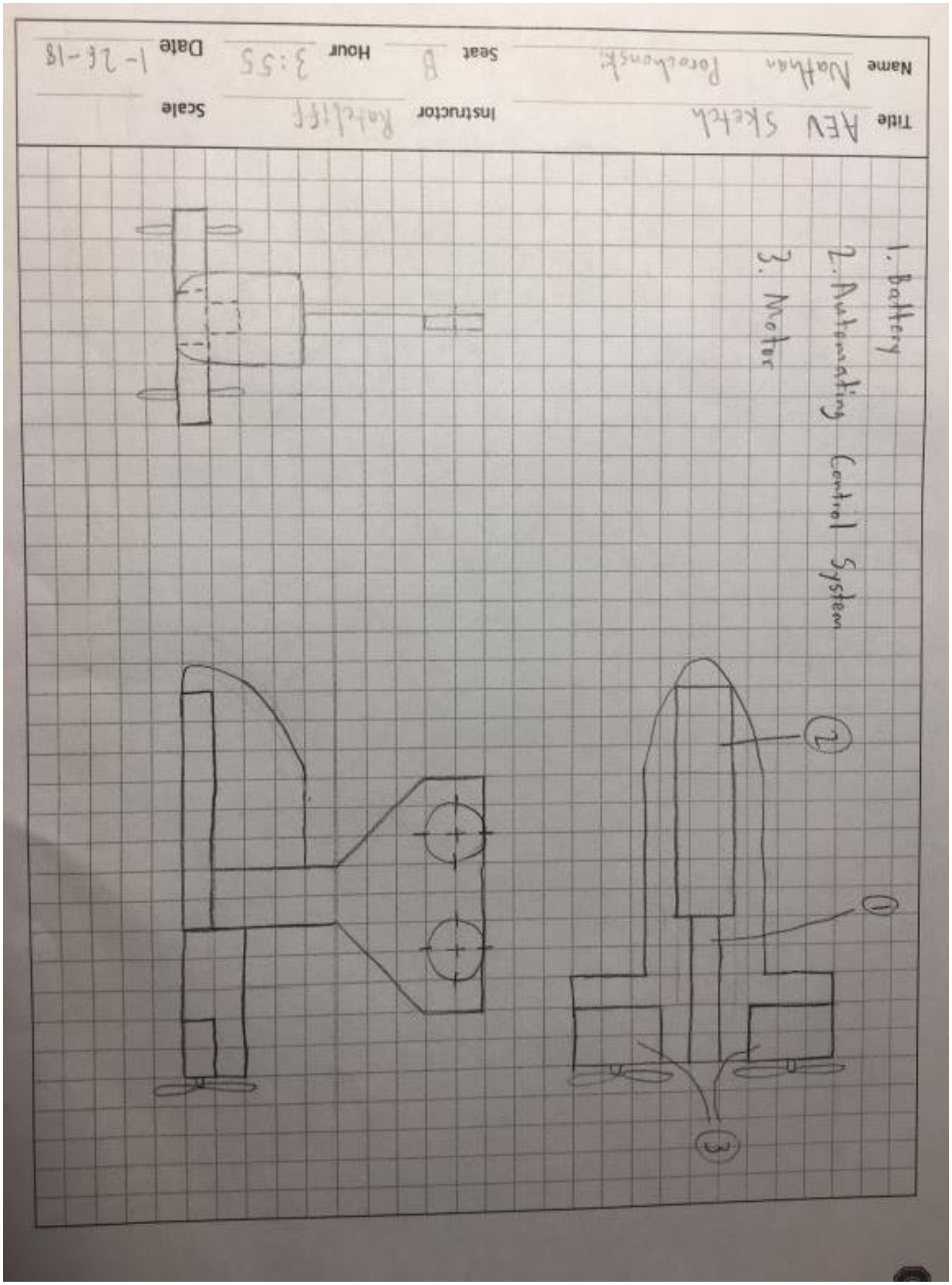
Bill of Materials - $157,690



Nathan's Design

Estimated Weight – 370 grams

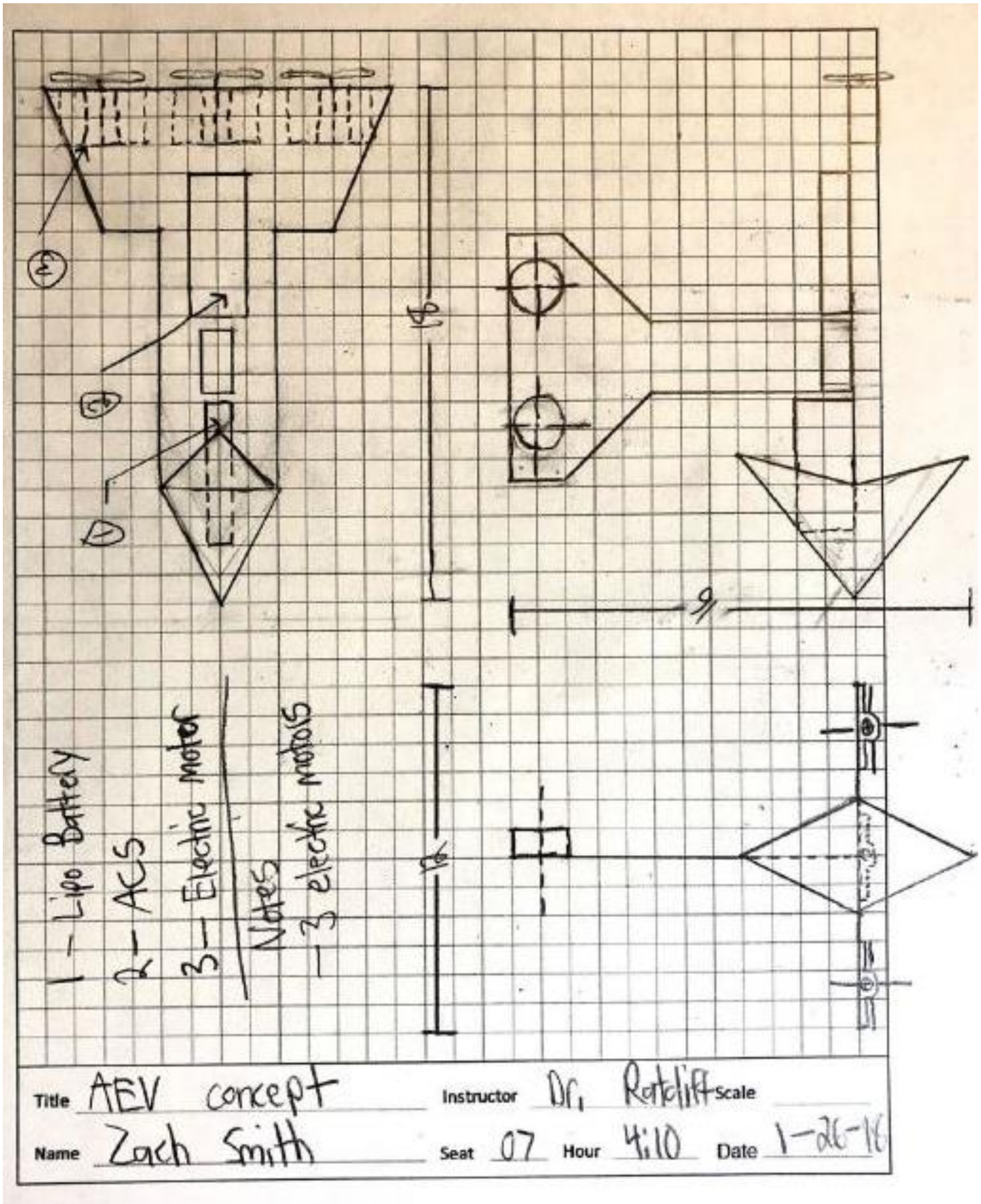
Bill of Materials - $142,750



Zach's Design

Estimated Weight – 500 grams

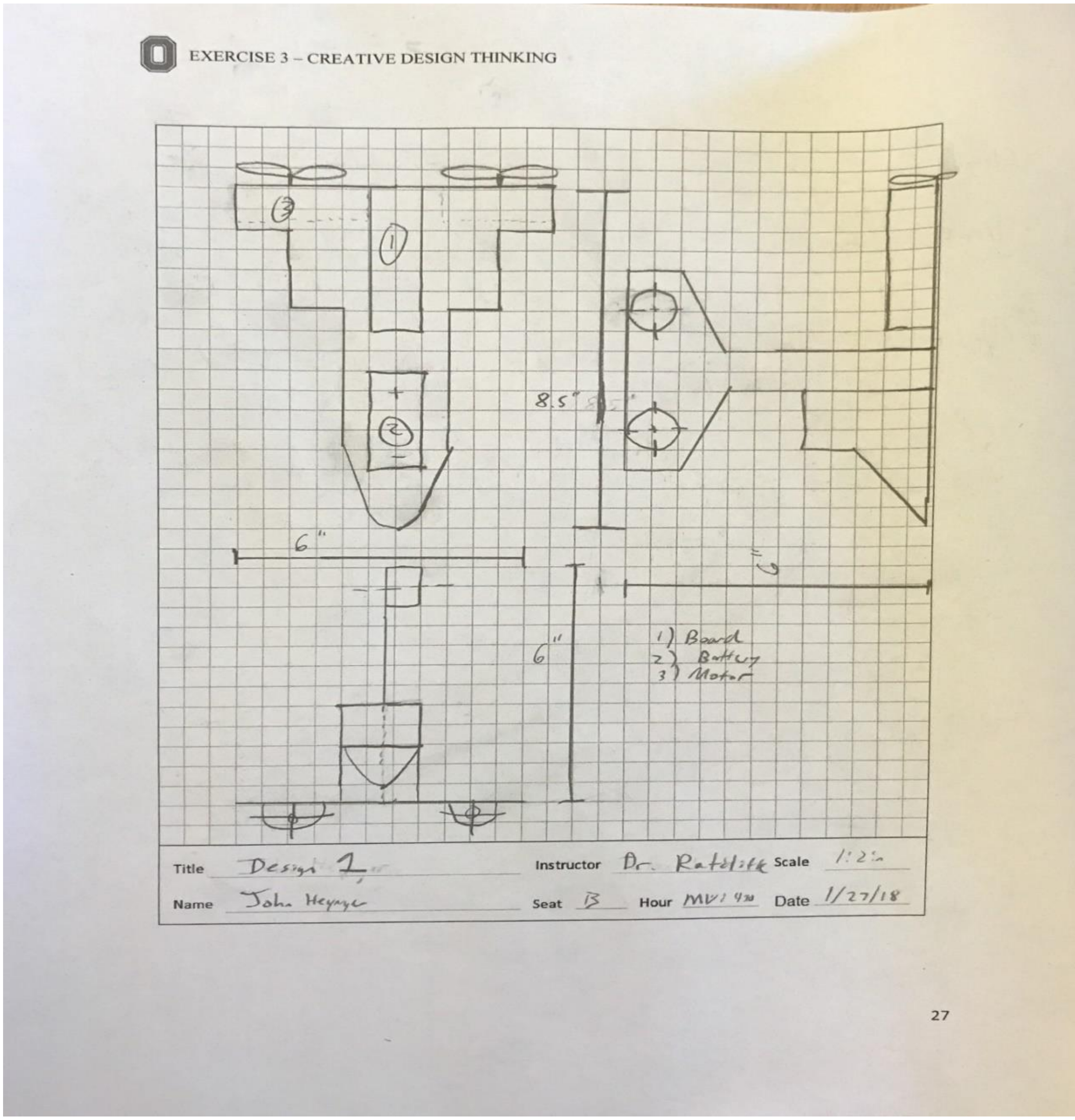
Bill of Materials – $162,810



John’s Design

Estimated Weight – 360 grams

Bill of Materials - $141,430



Appendix D - Data Analysis Tool

Code for Testing Data Analysis Tool:

Reference Appendix G – Data Analysis Tool Testing section for sample code used during this lab.

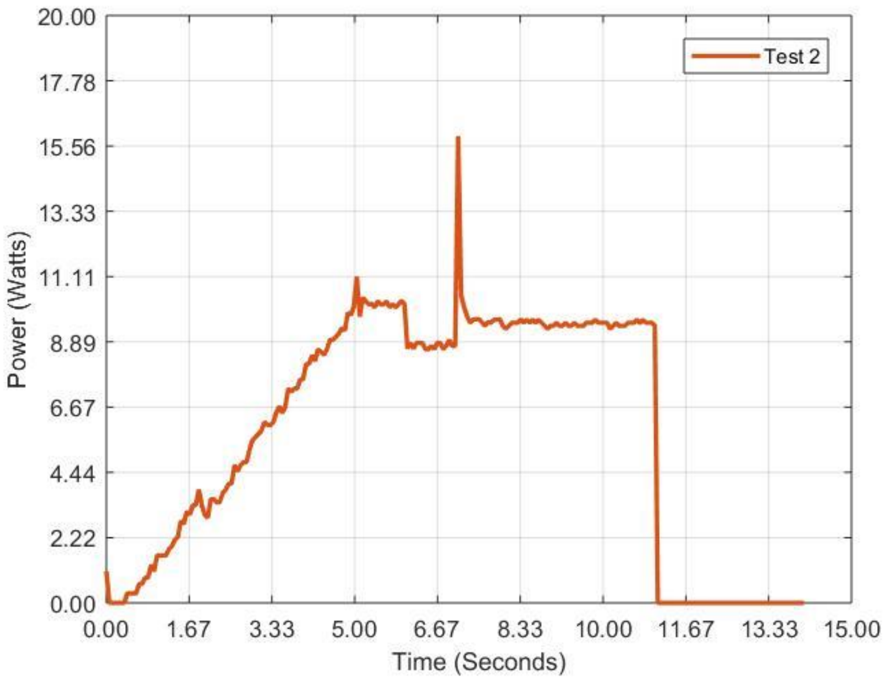


Figure 4 - Power vs. Time for Sample Code

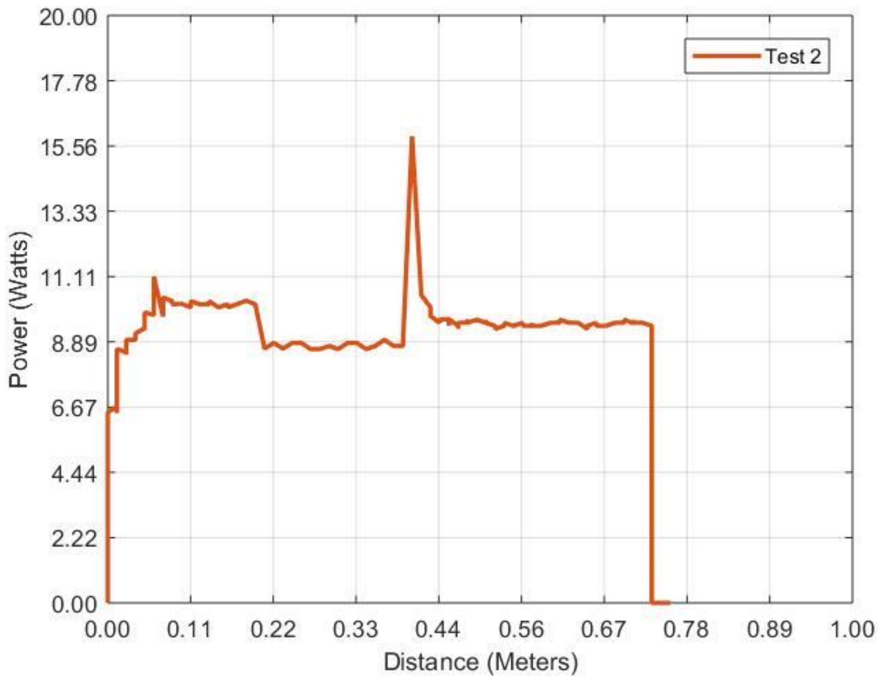


Figure 5 – Power vs. Distance for Sample Code

Appendix E - Screening and Scoring

Table 1: Concept Screening Matrix for Preliminary Designs

Cancept Screening Matrix

Success Criteria	Reference	Zach	Deepak	John	Nate
Aerodynamics	0	+	+	+	+
Weight	0	-	-	0	0
Safety	0	0	+	0	0
Energy Efficient	0	+	+	0	0
Speed	0	+	0	0	0
Sum +'s	0	3	3	1	1
Sum 0's	5	1	1	4	4
Sum -'s	0	1	1	0	0
Net Score	0	2	2	1	1
Continue?	Combine	Yes	Yes	Combine	Combine

Table 2: Concept Scoring Matrix for Preliminary Designs

Cancept Scoring Matrix

		Sample AEV		Zach		Deepak	
Success Criteria	Weight	Rating	Weighted Score	Rating	Weighted Score	Rating	Weighted Score
Aerodynamics	0.2	2	0.4	4	0.8	5	1
Weight	0.2	3	0.6	2	0.4	2	0.4
Safety	0.25	3	0.75	3	0.75	4	1
Energy Efficient	0.25	2	0.5	3	0.75	4	1
Speed	0.1	3	0.3	5	0.5	3	0.3
Total Score			2.55		3.2		3.7
Continue?		No		Yes		Yes	

The five criteria that the team decided were most important to address in the design of the AEV in order to complete the MCR were Aerodynamics, Weight, Safety, Energy Efficiency, and Speed. The team decided that safety and energy efficiency were the two most important, with Aerodynamics and weight coming in as the next two important criteria. Speed is a consideration in our design, but it is the least important criteria of the five. Below are some descriptions of what each criterion will be based around.

Aerodynamics –This section is focused on creating an AEV with the least drag possible. The more effective the AEV is a cutting through the air without wasting energy, the better.

Weight – This section is focused on efficient material use on the building the AEV, such as only materials needed, and using lightweight materials. The lighter the overall AEV design, the better. Cost could also be grouped here since the less materials used, the cheaper the AEV will be.

Safety – This section is focused on the overall design of the AEV and how balanced it is. The more balanced and well designed the AEV is, the better.

Energy Efficient – This section is a mainly a combination of the weight and aerodynamics criteria since the performance in each of these areas are crucial for energy efficiency. Overall, this section will be focused on how much energy the battery uses while the AEV is performing its task; the less energy used, the better.

Speed – This section is pretty straight forward, the faster the AEV can complete its task without being unsafe, the better.

Appendix F - Team Meeting Minutes

Meeting 1:
Date: 1-12-18
Location/Time: 3:55 (Face-to-Face), Hitchcock Hall Room 224
Members Present: Zach Smith, Nathan Porochonski
Topics Discussed: Major AEV components to test. Both members present initiated ideas.

Objective:
Today's main focus was on being introduced to the AEV design project, receiving our AEV kits, signing some preliminary paperwork, and discussing the major AEV components/aspects that we would be testing to improve our design.

To do/Action Items:
-Meeting minutes 1 - Zach

-Website Update 1 - All

Decisions:
We decided that we would rotate responsibility for the AEV kit every week. It's a big thing to carry around and it was most fair to share responsibility for it.

Reflections:
-Hard to do some of the task with only 2 of 4 teammates present.

-There are a lot of materials, requirements, and due dates for this project. Keeping up with all of them is going to be crucial.

-Also, since there are multiple groups working in the company, communication within and between the groups is going to be important.

Meeting 2:
Date: 1-19-18
Location/Time: 3:55 (Face-to-Face), Hitchcock Hall Room 224
Members Present: Zach Smith, Nathan Porochonski, Deepak Warrier, John Heyniger
Topics Discussed: AEV code

Objective: To become familiar with programming and building the AEV. We also needed to test the AEV through programming to make sure it worked right.

To do/Action Items:

Meeting minutes 2 - Nathan

Website update 2 - All

Individual AEV sketches - All

Decisions:

The next meeting, we will need to test the reflectance sensor and get a new reflectance if the test shows the old one is broken.

Reflections:

-One of the reflectance sensors was not working properly so we will have to get a new one next week.

-Following the instructions for building the AEV was simple and straightforward.

-The engines were working properly when we programmed them.

Meeting 3:
Date: 1-26-18
Location/Time: 3:55 (Face-to-Face), Hitchcock Hall Room 224
Members Present: Zach Smith, Nathan Porochonski, Deepak Warrier, John Heyniger
Topics Discussed: Reflectance Sensor fix, Lab 3 and AEV Design,

---

Objective: The first objective was to fix the reflectance sensor problem that tied us up at the end of the last meeting. Depending on the result and the time taken, we will then skip to part 4 of the lab and go back to part 3 if time permits.

To do/Action Items:

Meeting Minutes 3 - Zach

Website Update 2 - All

Finish up last part of lab 3 - All

Decisions: After fixing the reflectance sensor issue, we decided to skip the last part of lab 3 to move on to lab 4. This would allow us to stay on track for lab 4.

Reflections: We need to pay close attention when things go wrong, we spent a lot of time trying to fix what we thought was a broken sensor when in reality, a nut was just stuck between the sensor and the reflectance tape on the wheel.

Meeting 4:
Date: 2-2-18
Location/Time: 3:55 (Face-to-Face), Hitchcock Hall Room 224
Members Present: Zach Smith, Nathan Porochonski, Deepak Warrier, John Heyniger
Topics Discussed: Fixing the motor wiring, The group AEV design

Objective: To finish lab 3, fix the wiring on the motor, test the AEV again, and gather data from the tests.

To do/Action Items:

Create the team concept sketch of the AEV - All

Upload the team concept sketch to the website - All

Meeting Minutes 4 - Nathan

Screening/Scoring Assessment of Designs - All

Decisions:

We discussed our individual concept designs for the AEV and came up with ideas for our team AEV design. We also began a basic outline of our the criteria for our screening/scoring tables.

Reflections: We fell slightly behind in the previous labs because we had to repair a couple of things during them, but we worked hard to catch up and finished up labs 2 and 4. At the end of the class we began to finish lab 3 and start lab 5.

Appendix G – Arduino Code

Programming Basics – Sample Code

During the programming basics lab, the following program was creating

```
void prgmBasics(){

  //Scenario 1

  celerate(1, 0, 15, 2.5);  //1.  Accelerate motor one from start to 15% power in 2.5 seconds.

  motorSpeed(1, 15);        //2.  Run motor one at a constant speed (15% power) for 1 second.

  goFor(1);

  brake(1);                 //3.  Brake motor one.

  celerate(2, 0, 27, 4);    //4.  Accelerate motor two from start to 27% power in 4 seconds.

  motorSpeed(2,27);         //5.  Run motor two at a constant speed (27% power) for 2.7 seconds.

  goFor(2.7);

  celerate(2, 27, 15, 1);   //6.  Decelerate motor two to 15% power in 1 second.

  brake(2);                 //7.  Brake motor two.

  reverse(2);               //8.  Reverse the direction of only motor 2.

  celerate(4, 0, 31, 2);    //9.  Accelerate all motors from start to 31% power in 2 seconds.

  motorSpeed(4, 35);        //10. Run all motors at a constant speed of 35% power for 1 second.

  goFor(1);

  brake(2);                 //11. Brake motor two but keep motor one running at a constant speed

  motorSpeed(1, 35);        //    (35% power) for 3 seconds.

  goFor(3);

  brake(4);                 //12. Brake all motors for 1 second.

  goFor(3);

  reverse(1);               //13. Reverse the direction of motor one.

  celerate(1, 0, 19, 2);    //14. Accelerate motor one from start to 19% power over 2 seconds.

  goFor(2);                 //15. Run motor two at 35% power while simultaneously running motor
```

```
  motorSpeed(2, 35);        //    one at 19% power for 2 seconds.

  goFor(2);

  motorSpeed(1, 19);        //16. Run both motors at a constant speed (19% power) for
2 seconds.

  motorSpeed(2, 19);

  goFor(2);

  celerate(1, 19, 0, 3) ;    //17. Decelerate both motors to 0% power in 3 seconds.

  celerate(2, 19, 0, 3);

  brake(4);                  //18. Brake all motors

}
```

Reflectance Sensor Testing – Sample Code

```
void ExternalSensorsOutside(){

  motorSpeed(4,25);

  goFor(2);

  motorSpeed(4, 20);

  goToAbsolutePosition(convertInchesToMarks(144));

  reverse(4);

  motorSpeed(4, 30);

  goFor(1.5);

  brake(4);

}
```

Data Analysis Tool Testing – Sample Code

```
void CSS1() {
  resverse(4);
  celerate(4, 0, 45, 5);     //Step 1
  goFor(1);                        //Step 2
  motorSpeed(4, 40);      //Step 3
  goFor(1);
  reverse(4);                    //Step 4
  motorSpeed(4, 45);     //Step 5
  goFor(4);
  brake(4);                        //Step 6
}
```