Group J - Matthew Caldwell, John Kim, Jacob Phillips, Tarun Pilli Progress Report Week 2

Instructor - Dr. Philip Schlosser, GTA - Melissa Hrivnak 1/24/2017

Week 1

Situation

This week, the group completed some basic tasks in lab to gain familiarity with the Arduino and the AEV parts. First introduced was the AEV controller, which included an Arduino Nano microcontroller. Additionally, a plastic motor stand assembly was introduced. This assembly could hold two motors in place in the air. Propellers were mounted to motors, which were then mounted into the motor stand. These motors were connected to the Arduino for control, and the Arduino was then connected to the battery for power. This gave the group experience in working with physical assemblies involving the Arduino, which will continue to be necessary throughout the project. To program the Arduino, the Arduino IDE was introduced as well as the AEV Sketchbook which provides the necessary commands for interfacing with the controller. The necessary programs and files were installed with the assistance of the GTA, who took the class through the process. Function calls for programming the AEV were introduced, including celerate, motorSpeed, goFor, brake, and reverse. Arduino will be used throughout the entire AEV lab, and thus it will be crucial to be familiarized with Arduino's programing. After everything was set up, the code for the Arduino was written. The group proceeded to follow the guidance of the given scenarios, writing code that performed actions given in the scenario. Function calls were used to perform different actions, such as braking and reversing motors. Once written, the code was uploaded to the Arduino and tested with the motor assembly. A second scenario was also programmed, uploaded, and tested.

Results and Analysis

For the group, both scenarios functioned properly. The propellers experienced slight resistance at the beginning of the code; this was a result of the propellers being positioned too close to the motors. The base of the motors caused friction between the propeller and the motor. However, the propellers began moving soon after the start.

There will be several limitations to Arduino's code that the group will have to consider when coding. Arduino's code will be unable to use one single line of code to command two motors while leaving another motor left untouched (assuming that the AEV has 3 motors). Also, while the brake command will be able to quickly stop any given motor, this command will not be able to stop the entire AEV system as fast.

The only minute error was with confusion regarding the on/off and reset button on the controller. The group was puzzled because the controller would not execute properly, but this was because the reset button kept on getting pushed instead of the on/off button. Once the group realized, everything functioned properly. The team was able to complete both Scenario 1 and Scenario 2 as well. In Scenario 2, the propellers created the rhythm of the Imperial March, an iconic Star Wars Theme.

Due to the lack of familiarity the team members had with the Arduino environment, it took time to get a grasp on the process of programming the device. As a result, it took a long time to complete the first activity of the lab. The second activity had to be rushed through and there was nearly not enough time to test it. As the lab contained many students with varying levels of programming knowledge and experience, it would be prudent to simplify the lab slightly or make it shorter in some other way. This would ensure all students are able to reach an adequate level of understanding with the programming environment and build upon these skills in the future lab sessions.

<u>Takeaways</u>

1) General - To improve efficiency and overall results, the group will try to delegate tasks. The beginning of the first team meeting started out slow as everyone was working on the progress report, but it was never decided who would do each part. Instead the group members timidly worked on the report not trying to be assertive. Eventually tasks became delegated as some members worked on the progress report, and others constructed the AEV which allowed the meeting to flow more smoothly.
2) Controller - General locations of various features on the controller were learned. In order to insert the program from sketchbook to the arduino requires a series of steps. In order to correctly do this, the arduino must be turned on at a certain time, and sometimes reset. At first the group confused which buttons did certain operation on the arduino, but by the end of the lab they were learned and will be used in subsequent labs when the arduino is programmed to maneuver on the track.
3) Programming Basics- Simple programming basics for the AEV were learned using the Arduino IDE. The AEV was placed a ring stand on a desk in order to test if the correct program had been created to meet the requirement in the lab manual. More importantly, fundamental programming basics were learned that will be used in subsequent labs to program the AEV to perform certain tasks on the overhead tracks.

Week 2

Situation

In the coming week, the group will be exposed to various components of the AEV Lab. Namely, external sensors, such as the reflectance sensors. With the assistance of the reflective sensor, the Arduino will be able to record whenever a non-reflective surface passes over the sensor. It will then be possible to instruct the AEV to travel to certain relative distances. In order to do this, new function calls will be utilized. These function calls will be able to make the AEV either travel to a position relative to its current location or travel to an absolute location. Troubleshooting methods will also be touched upon within the next week.

Alongside the observance of external sensors, the group will also examine propulsion efficiency. By testing propellers using wind tunnels, the group will physically observe and measure parameters such as RPM, current, and thrust. Furthermore, the group will calculate other values, including calibrated thrust, advance ratio, and propulsion efficiency. The group will hopefully learn insightful information about propellers and various configurations which will aid in future decisions regarding the build of the AEV.

Weekly Goals

1) Thoroughly prepare for the lab next week by reading the lab manual and taking the pre-lab quiz.
2) Build the sample AEV for Lab 2.
3) Finish the progress report for lab 2 as to meet the required deadline and not fall behind in the AEV project.

Weekly Schedule

Task	**Teammate(s)**	**Start Date**	**Due Date**	**Time Needed**
Progress Report for Lab 2	All	1/18/17	1/24/17	3 Hours
AEV Construction	Tarun Pilli & John Kim	1/23/17	1/24/17	1 Hour
Prepare for Lab 2	All	1/23/17	1/24/17	30 Minutes

Appendix A

Date: 1/23/17
Time: 5:30 P.M.
Members Present: Jacob Phillips, Matthew Caldwell, Tarun Pilli, & John Kim
Topics Discussed: Lab 2 Progress Report, and AEV Construction

---

Objective:
Group J met before Lab 2 on Tuesday, January 24th in order to complete the progress report for Lab 2, and build the AEV that would be tested during Lab 2 in class.

Tasks:
-Progress Report for Lab 2
Task was started by Tarun early the past week, but was completed at the team meeting by the rest of the group members. Much of the report had been completed before the team met, as only the appendix with the Arduino code, and the team meeting notes were missing beforehand. The progress report was completed during the team meeting before the group meeting had ended.
-AEV Construction
The AEV was built by Tarun and John during the team meeting. No prior work had been put into the construction beforehand, and the task was completed before the meeting ended.

To do/Action Items:
-Add the progress report to the project portfolio on U.osu.edu before lab 3(All Members)
-Prepare for lab 3 by completing any pre-class reading and the pre-lab quiz(All Members)

Reflections:
Thus far Group J has no problem meeting deadlines with assignments or working well together. Decisions were made as a group with no disagreement on anything, and all of the work was completed promptly and at a standard that every group member was pleased with.

Appendix B

PrgrmBasics code

```
celerate(1, 0, 15, 2.5); //accelerate motor one to 15% power in 2.5s
motorSpeed(1, 15); //set motor one's speed to 15%
goFor(1); //maintain above speed for 1 sec
brake(1); //brake motor one
celerate(2, 0, 27, 4); //accelerate motor two to 27% power in 4s
motorSpeed(2, 27); //set motor two's speed to 27%
goFor(2.7); //maintain above speed for 2.7s
celerate(2, 27, 15, 1) //decelerate motor two to 15% in 1s
brake(2); //brake motor two
reverse(2); //reverse motor two
celerate(4, 0, 31, 2); //accelerate all motors to 31% in 2s
motorSpeed(4, 35); //set all motors' speeds to 35%
goFor(1); //maintain above speed for 1s
brake(2); //brake motor two
motorSpeed(1, 35); //set motor one's speed to 35%
goFor(3); //maintain above speed for 3s
brake(4); //brake all motors
goFor(1); //keep motors braked for 1s
reverse(1); //reverse motor one
celerate(1, 0, 19, 2); //accelerate motor one to 19% power in 2s
motorSpeed(2, 35); //set motor two's speed to 35%
motorSpeed(1, 19); //set motor one's speed to 19%
goFor(2); //maintain above speeds for 2s
motorSpeed(1, 19) //set motor one's speed to 19%
motorSpeed(2, 19) //set motor two's speed to 19%
goFor(2); //maintain above speeds for 2s
celerate(1, 19, 0, 3); //decelerate motor one to 0% power in 3s
celerate(2, 19, 0, 3); //decelerate motor two to 0% power in 3s
brake(4); //brake all motors
```

PrgmBasicsStarWars code

```
reverse(4); //reverse all motors
motorSpeed(4, 25); //set all motor speeds to 25%
brake(4); //brake motor 4
goFor(0.1); //do above for 0.1s
motorSpeed(4, 25); //set all motor speeds to 25%
brake(4); //brake motor 4
goFor(0.1); //do above for 0.1s
motorSpeed(4, 15); //set all motor speeds to 15%
goFor(0.3); //do above for 0.3s
brake(4); //brake all motors
goFor(0.05); //do above for 0.05s
motorSpeed(4, 40); //set all motor speeds to 40%
goFor(0.3); //do above for 0.3s
motorSpeed(4, 25); //set all motor speeds to 25%
goFor(0.5); //do above for 0.5s
motorSpeed(4, 15); //set all motor speeds to 15%
goFor(0.3); //do above for 0.3s
brake(4); //brake all motors
goFor(0.05); //do above for 0.05s
motorSpeed(4, 40); //set all motor speeds to 40%
goFor(0.3); //do above for 0.3s
motorSpeed(4, 25); //set all motor speeds to 25%
goFor(0.5); //do above for 0.5s
brake(4); //brake all motors
goFor(0.5); //do above for 0.5s
motorSpeed(4, 55); //set all motor speeds to 55%
goFor(0.5); //do above for 0.5s
brake(4); //brake all motors
goFor(0.1); //do above for 0.1s
motorSpeed(4, 55); //set all motor speeds to 55%
goFor(0.5); //do above for 0.5s
brake(4); //brake all motors
goFor(0.1); //do above for 0.1s
motorSpeed(4, 65); //set all motor speeds to 65%
goFor(0.3); //do above for 0.3s
brake(4); //brake all motors
goFor(0.05); //do above for 0.05s
```

```
motorSpeed(4, 40); //set all motor speeds to 40%
goFor(0.3); //do above for 0.3s
motorSpeed(4, 20); //set all motor speeds to 20%
goFor(0.5); //do above for 0.5s
motorSpeed(4, 15); //set all motor speeds to 15%
goFor(0.3); //do above for 0.3s
brake(4); //brake all motors
goFor(0.05); //do above for 0.05s
motorSpeed(4, 40); //set all motor speeds to 40%
goFor(0.3); //do above for 0.3s
motorSpeed(4, 25); //set all motor speeds to 25%
goFor(0.5); //do above for 0.5s
brake(4); //brake all motors
```