Abrahm Williams, Aziz Bokari, Dejonte Greathouse, Nina Tavernier
Group L-Instr. Schrock, GTA-Patankar

AEV Lab 2
February 4,2016

Executive Summary

The purpose of this lab was to explore the basic functions and usage of Adruino, automatic control, system by providing hands-on application and visualization of function calls: “celerate,” “motorSpeed,” and others. A crucial aspect of this lab was becoming familiar with the automatic control system’s hardware components, in order to properly upload programs onto the AEV and test its outcome. The main components of the lab included two scenarios serving as programing exercises.

The objective of this lab was to prepare the team for the future task of applying these basic Adruino commands to the overall program of the AEV. The Arduino will be responsible for tasks such as controlling the motor functions and recording system data. Therefore, it is crucial that the team understands the ins and outs of the Arduino Nano microcontroller before beginning the building process.

When the basic program was applied to the AEV, it allowed the motor functions of each motor to be observed. One thing that was noticed was the startup of the motors. Sometimes the rotation would sputter or hesitate when the motors were activated. This was most prevalent when the motors functioned at lower speeds, or percentages and can be attributed to the motors warming up to perform the rest of the program. The motors also seemed to accelerate quickly to the target percentage instead of coasting to the speed.

One of the commands utilized during the lab was “brake(m);”. The command, contrary to its name did not break the AEV, but rather cut off the power supply of the motors. This command could be considered inefficient due to its inability to stop the AEV quickly. This aspect of the brake command has the potential to delay the rest of the code. **(more on how it will affect the AEV in the lab practical)**

The brake command can be considered in the category of error. This is however, easy to fix by having another team member proof read the program to check any errors made. Inherent errors occupied the programming sequence as well. Because arduino language is not specific for a certain motor, there is a broad definition of the percentages of power output, possibly causing one motor to be moving faster than the other. This as well as lag between arduino commands and motor functions cannot be fixed unless there is an upgrade in technology.

The lab also aided students in exploring arduino files. A common problem was the inclusion of semicolons in the code. More instructions and explaining from the GTA and TAs would solve this as it is a minimal problem. Another problem was the code having to be written on only one of the computers which made it hard for all students to participate in coding. A recommendation would be to use teamviewer so everyone is able to make suggestions.

Basic functions of the arduino coding and microcontroller were apparent to students by the conclusion of the lab. By becoming familiar with the built in functions in arduino, coding efficiency will be improved leading to better results in the overall performance of the AEV. Such errors like misspellings, semicolons or the transfer of the code to the device will be avoided in the future, giving the resistance the upper hand against the empire.

Appendix:

Scenario 1 code:

```
celerate(1,0,15,2.5);       \\Accelerate motor one from start to 15% power in 2.5 s.

motorSpeed(1,15)            \\Run motor one at a constant speed of 15% power for 1 s.
goFor(1);

brake(1);                   \\Brake motor one.

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

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

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

brake(2);                   \\Brake motor two.

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

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

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

brake(2);                   \\Brake motor 2. Keep motor one running at 35% power for 3 s
motorSpeed(1,35);
goFor(3);

brake(4);                   \\Brake all motors for 1 second.
goFor(1);

reverse(1);                 \\Reverse the direction of motor one.
celerate(1,0,19,2);         \\Accelerate motor one from start to 19% power over 2 seconds.

motorSpeed(2,35);           \\Run motor two at 35% power and motor one at 19% power for 2
motorSpeed(1,19);
goFor(2);
```

```
motorSpeed(4,19);          \\Run both motors at a constant speed (19% power) for 2 seconds.
goFor(2);

celerate(4,19,0,3);         \\Decelerate both motors to 0% power in 3 seconds.
brake(4);                   \\Brake all motors.

\\Save Program
\\end
```

Scenario 2 code:

```
reverse(4);                  \\Reverse all motors.

motorSpeed(4,25);           \\Power all motors at 25% power for 0.5 second.
goFor(0.5)

brake(4);                   \\Brake all motors for 0.1 seconds.
goFor(0.1);

motorSpeed(4,25);          \\Power all motors at 25% power for 0.5 second.
goFor(0.5)

brake(4);                  \\Brake all motors for 0.1 seconds.
goFor(0.1);

motorSpeed(4,25);         \\Power all motors at 25% power for 0.5 second.
goFor(0.5)

brake(4);                 \\Brake all motors for 0.1 seconds.
goFor(0.1);

motorSpeed(4,15);        \\Power all motors at 15% power for 0.3 second.
goFor(0.3);

brake(4);                 \\Brake all motors for 0.05 seconds.
goFor(0.05);

motorSpeed(4,40);        \\Power all motors at 40% power for 0.3 second.
goFor(0.3);

motorSpeed(4,25);        \\Power all motors at 25% power for 0.5 second.
goFor(0.5);
```

```
motorSpeed(4,15);        \\Power all motors at 15% power for 0.3 second.
goFor(0.3);

brake(4);                \\Brake all motors for 0.05 seconds.
goFor(0.05);

motorSpeed(4,40);       \\Power all motors at 40% power for 0.3 second.
goFor(0.3);

motorSpeed(4,25);       \\Power all motors at 25% power for 0.5 second.
goFor(0.5);

brake(4);                \\Brake all motors for 0.5 seconds.
goFor(0.5);

motorSpeed(4,55);       \\Power all motors at 55% power for 0.5 second.
goFor(0.5)

brake(4);               \\Brake all motors for 0.1 seconds
goFor(0.1);
   .
                        \\Repeat steps 14 and 15 a total of 2 times

motorSpeed(4,55);       \\Power all motors at 55% power for 0.5 second.
goFor(0.5)

brake(4);              \\Brake all motors for 0.1 seconds
goFor(0.1);

motorSpeed(4,55);      \\Power all motors at 55% power for 0.5 second.
goFor(0.5)

brake(4);              \\Brake all motors for 0.1 seconds
goFor(0.1);

motorSpeed(4,65);     \\Power all motors at 65% power for a total of 0.3 second.
goFor(0.3);

brake(4);             \\Brake all motors for 0.05 seconds.
goFor(0.05)

motorSpeed(4,40);     \\Power all motors at 40% power for 0.3 second.
goFor(0.3);
```

```
motorSpeed(4,20);   \\Power all motors at 20% power for 0.5 second.
goFor(0.5);

motorSpeed(4,15)    \\Power all motors at 15% power for 0.3 second.

goFor(0.4);
brake(4);
goFor(0.05);                \\Brake all motors for 0.05 seconds.

motorSpeed(4,40);      \\Power all motors at 40% power for 0.3 second.
goFor(0.3);

 motorSpeed(4,25);      \\Power all motors at 25% power for 0.5 second
goFor(0.5);
brake(4);                    \\Brake all motors.
```

Save Program as (Save As:) ProgramBasicsStarWars