RSA Lab #7: Finite State Machines with Magnetometer & Accelerometer

# Part A: EEPROM

This section of the lab comprised of demonstrating the functionality of the EEPROM library and how it interacts with variables.

1. Open the Serial Monitor window; what does this program do? What happens when you hit the reset button on the Arduino? What happens when you unplug the Arduino and plug it back in? (Unplug the Arduino, close the Serial Monitor, plug it back in, and reopen the Serial Monitor— does x reset?) (Normally, variables are lost when the power is lost, while the program is retained.)

This program simply prints the value of x at each iteration, where x is then incremented until it exceeds 100, where it resets back to zero. When the Arduino’s reset button is hit, the program simply stops (which can be indefinite for as long as the button is held), then continues after a short while. If the microcontroller is disconnected and then plugged back into the computer, the variable in question does not reset and just continues normally.

# Part B: IMU: Magnetometer

For this part of the lab, I first tested the magnetometer by displaying raw values of x,y,z, then creating a finite state machine for a button-controlled operation, then implementing the states using switch statements.

1. Make a program that displays the compass direction, in degrees, using the magnetometer on the MinIMU-9 inertial measurement unit (IMU). Submit a screen shot of the Serial Plotter showing the variations in the x- and y-components as you rotate the magnetometer about the z-axis.

Graphical user interface, application

Description automatically generated

1. Make a State Diagram (bubbles for states, arrows for transitions between states) for an Arduino program that does the following (on lab). Submit your state diagram.
2. Write an Arduino program and implement the finite state machine model. Submit your commented Arduino code.

Text

Description automatically generated

Text

Description automatically generated with low confidence

Table

Description automatically generated with low confidence

# Part C: IMU: Accelerometer, and Hobby Servomotor or, “What’s Up, Doc?”

Finally, in this section I was able to measure the acceleration using the accelerometer and then converting the acceleration (in G’s) to degrees, which was then synced with a servo motor to control its rotations.

1. Write an Arduino program to measure the z acceleration components and display it on the Serial Monitor four times per second. Calculate the orientation of the board’s z-axis in degrees away from vertical (i.e. 0° is facing upward, 90° sideways, and 180° is facing downward). Submit your commented Arduino code.

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text

Description automatically generated

1. Finally, use your orientation reading to control the angle of a hobby servomotor (Tower Pro SG90 Micro Servomotor). Print a sample of the Serial Plotter output, showing a range from about 0° to about 180°.

Text

Description automatically generated