

# **System Design Document for Roadie**

Sponsor

**Electrical, Computer, Software & Systems Engineering at Embry-Riddle Aeronautical University**

Released 04 December 2014

**Are We There Yet?**

## Revision History

Date	Reason for Change	Version
4 Dec 2014	Initial Release	1.0.0
3 Dec 2014	Function Diagrams	0.5.0
2 Dec 2014	Hardware Diagrams	0.4.0
1 Dec 2014	High Level Architecture	0.3.0
29 Nov 2014	System Diagrams	0.2.0
27 Nov 2014	Initial Draft	0.1.0

## Table of Contents

<b>Revision History .....</b>	ii
<b>1. Introduction.....</b>	1
<b>1.1 Purpose.....</b>	1
<b>1.2 Problem Statement.....</b>	1
<b>1.3 Scope.....</b>	1
<b>1.4 Team Information.....</b>	1
<b>1.5 Overview .....</b>	1
<b>2. System Overview .....</b>	2
<b>2.1 Coordination Subsystem.....</b>	3
<b>2.2 Challenge Subsystem .....</b>	3
<b>2.3 Movement Subsystem .....</b>	3
<b>3. Design Considerations .....</b>	5
<b>3.1 Assumptions.....</b>	5
<b>3.1.1 Operation.....</b>	5
<b>3.1.2 Rule Changes .....</b>	5
<b>3.2 Dependencies .....</b>	5
<b>3.2.1 Intersystem Communication .....</b>	5
<b>3.2.2 Power .....</b>	5
<b>3.3 Constraints.....</b>	6
<b>3.3.1 Size.....</b>	6
<b>3.3.2 System Power Draw .....</b>	6
<b>3.3.3 Monetary.....</b>	6
<b>3.3.4 Technical Expertise.....</b>	6
<b>3.4 Industrial Standards Followed .....</b>	6
<b>3.4.1 Wiring Colors .....</b>	6
<b>3.4.2 Software Standards.....</b>	7
<b>3.5 Safety Constraints and Considerations.....</b>	7
<b>3.5.1 Electrical Hazards.....</b>	7
<b>3.5.2 System Heating.....</b>	7
<b>3.5.3 Battery Nominal Voltage .....</b>	8
<b>3.6 Environmental Considerations .....</b>	8
<b>3.6.1 Component Waste .....</b>	8
<b>4. System Design.....</b>	9
<b>4.1 High Level Architecture .....</b>	9

<b>4.1.1</b>	<b>Coordination Subsystem</b>	9
<b>4.1.2</b>	<b>Movement Subsystem</b>	10
<b>4.1.3</b>	<b>Challenge Subsystem</b>	10
<b>4.2</b>	<b>System Hardware Diagram</b>	11
<b>4.3</b>	<b>System State Definitions</b>	13
<b>4.4</b>	<b>System State Diagram</b>	14
<b>4.5</b>	<b>System Function Diagrams</b>	15
<b>4.5.1</b>	<b>Coordination Subsystem</b>	16
<b>4.5.2</b>	<b>Movement Subsystem</b>	16
<b>4.5.3</b>	<b>Challenge Subsystem</b>	16
<b>5.</b>	<b>Decomposition of Coordination Subsystem</b>	17
<b>5.1</b>	<b>Subsystem Architecture</b>	17
<b>5.1.1</b>	<b>Assumptions</b>	17
<b>5.1.2</b>	<b>Dependencies</b>	18
<b>5.1.3</b>	<b>Constraints</b>	18
<b>5.2</b>	<b>Functional Diagrams</b>	19
<b>5.3</b>	<b>Hardware Diagrams</b>	21
<b>5.4</b>	<b>Interfaces</b>	22
<b>5.5</b>	<b>Parts Budget</b>	24
<b>5.5.1</b>	<b>Components</b>	24
<b>5.5.2</b>	<b>Cost Matrix</b>	25
<b>6.</b>	<b>Decomposition of Challenge Subsystem</b>	26
<b>6.1</b>	<b>Subsystem Architecture</b>	26
<b>6.1.1</b>	<b>Assumptions</b>	26
<b>6.1.2</b>	<b>Dependencies</b>	27
<b>6.1.3</b>	<b>Constraints</b>	27
<b>6.2</b>	<b>Functional Diagrams</b>	28
<b>6.2.1</b>	<b>Simon Detected</b>	29
<b>6.2.2</b>	<b>Etch-A-Sketch Detected</b>	29
<b>6.2.3</b>	<b>Rubik's Cube Detected</b>	29
<b>6.2.4</b>	<b>Playing card Detected</b>	29
<b>6.3</b>	<b>Hardware Diagrams</b>	30
<b>6.4</b>	<b>Interfaces</b>	33
<b>6.5</b>	<b>Parts Budget</b>	35
<b>6.5.1</b>	<b>Components</b>	35
<b>6.5.2</b>	<b>Cost Matrix</b>	36

<b>7. Decomposition of Movement System.....</b>	<b>37</b>
<b>7.1 Subsystem Architecture.....</b>	<b>37</b>
<b>7.1.1 Assumptions.....</b>	<b>37</b>
<b>7.1.2 Dependencies .....</b>	<b>37</b>
<b>7.1.3 Constraints.....</b>	<b>38</b>
<b>7.2 Functional Diagrams .....</b>	<b>38</b>
<b>7.3 Hardware Diagrams .....</b>	<b>39</b>
<b>7.4 Interfaces .....</b>	<b>40</b>
<b>7.5 Parts Budget .....</b>	<b>41</b>
<b>7.5.1 Components .....</b>	<b>41</b>
<b>7.5.2 Cost Matrix.....</b>	<b>42</b>
<b>8. System Test Plan .....</b>	<b>43</b>
<b>8.1 Coordination Test Cases.....</b>	<b>43</b>
<b>8.2 Challenge Test Cases .....</b>	<b>51</b>
<b>8.3 Movement and Misc. Test Cases.....</b>	<b>58</b>
<b>9. Appendix B .....</b>	<b>68</b>
<b>9.1 Competition Course.....</b>	<b>68</b>
<b>10. References.....</b>	<b>69</b>

## Table of Figures

<b>Fig. 1:</b> Division of Roadie into three subsystems.....	2
<b>Fig. 2:</b> High level description of the systems in Roadie. ....	3
<b>Fig. 3:</b> High level design of the system. The system has been divided into Coordination (green), Movement (blue) and Challenge (red) subsystems. Power flowing through the system is represented with a red line, input to the system is represented by a blue line and any output from the system is represented by a black line.....	9
<b>Fig. 4:</b> The above diagram shows how all the components of the system connect to one another.....	11
<b>Fig 5:</b> The picture above is a front view of the whole system. This includes the chassis, all electrical components and all mechanical components. ....	12
<b>Fig 6:</b> The picture above is a front view of the whole system. This includes the chassis, all electrical components and all mechanical components. ....	12
<b>Fig. 7:</b> State diagram for Roadie.....	14
<b>Fig. 8:</b> Function diagram showing the major function calls in the system. Blocks green in color represent functions in the Coordination subsystem, blocks in blue represent the Movement subsystem and blocks in red represent the Challenge subsystem. The cyan rectangle is software that resides on the Arduino and the pink rectangle is software that resides on the Raspberry Pi.....	15
<b>Fig. 9:</b> Decomposition of Coordination subsystem for Roadie. ....	17
<b>Fig. 10:</b> Procedure flow through the Coordination subsystem.....	19
<b>Fig. 11:</b> Diagram which shows the electrical connections between the components of the Coordination subsystem. ....	21
<b>Fig. 12:</b> The picture above shows the sensor array used in the coordination system. The yellow circle marks the pin headers which are used to connect the wires from the Raspberry Pi.....	22
<b>Fig 13:</b> The picture above shows the USB connection between the Arduino Mega and the Raspberry Pi. The yellow circle indicates the USB connector to the Raspberry Pi and the red circle indicates the USB connector to the Arduino Mega .....	22
<b>Fig. 14:</b> The picture above shows the camera system and its connection to the Raspberry Pi. The yellow circle indicates the USB connection from the camera to the Raspberry Pi. The red circle indicates the Camera.....	23
<b>Fig 15:</b> The above picture shows the connection between the microcontroller battery and the Raspberry Pi. The yellow circle indicates the microcontroller battery and the red circle indicates the connection between the battery and the Raspberry Pi.....	23
<b>Fig. 16:</b> Decomposition of Challenge subsystem for Roadie. ....	26
<b>Fig. 17:</b> Function diagram for the Challenge subsystem.....	28
<b>Fig. 18:</b> Hardware diagram showing the electrical connections between the components of the challenge completion system.....	30
<b>Fig. 19:</b> The picture above shows the Etch-a-Sketch interactor.....	31
<b>Fig. 20:</b> The picture above is a CAD rendering of the Etch-a-Sketch interactor. Two gear motors will be mounted on top of the conical knobs which will rotate freely.....	31
<b>Fig. 21:</b> The picture above is a CAD rendering of the Simon interactor. This is the bottom of the part, the small extension will be what pressed the buttons. ....	32
<b>Fig 22:</b> The picture above is a CAD rendering of the Simon interactor. This is a top view of the part.....	32
<b>Fig. 23:</b> The picture above is the motor shield connected to all devices.....	33
<b>Fig. 24:</b> The picture above shows the battery used to power the challenge system components. The yellow circle indicates the connector used to connect the battery to the voltage regulator. ....	33

<b>Fig. 25:</b> The picture above shows the Voltage Regulator used to control the voltage to the challenge system. The yellow circle indicates the input to the Voltage Regulator, which comes from the battery and the red circle indicates the output which goes to the motor shield. ....	34
<b>Fig. 26:</b> The picture above shows the Motor Shield. The yellow circle indicates the power input to the motor shield, which comes from the output terminal of the voltage regulator. The red circles denote the output pins which provided signals to the DC motor and the green circle indicates the pins which provide signals to and from the servo. ....	34
<b>Fig. 27:</b> The picture above shows the Servo. The yellow circle indicates the connector which connects to the motor shield.....	35
<b>Fig 28:</b> The picture above is of the small DC motors in the challenge system. The two yellow circles indicate the two input pins which connect to the output pins on the motor shield.....	35
<b>Fig. 29:</b> Decomposition of Movement subsystem Roadie. ....	37
<b>Fig. 30:</b> Function calls and procedure flows for Movement subsystem. ....	38
<b>Fig. 31:</b> Hardware diagram showing the electrical connections between the components of the Movement subsystem .....	39
<b>Figure 32:</b> The above picture shows the motor shaft and the wheels connected by a 3D printed coupler .....	40
<b>Figure 33:</b> The picture above shows the stepper motor. The yellow circle indicates the wire connections which connect to the motor shield shown in <b>Fig. 20</b> .....	41
<b>Fig. 34</b> Competition course for SoutheastCon[1]. .....	68

# 1. Introduction

## 1.1 Purpose

## 1.2 Problem Statement

To create an autonomous robot to compete in the 2015 IEEE SoutheastCon student hardware competition.

## 1.3 Scope

Roadie is intended to compete in the 2015 IEEE Southeast Con student hardware competition. The system is envisioned to complete four unique challenges:

- Correctly play Simon for 15 seconds
- Draw “IEEE” on an Etch-A-Sketch
- Twist one row of a Rubik’s cube 180 degrees
- Pick up and carry one playing card across the finish line

Roadie system is intended to successfully complete the challenges outlined above within a time limit of five minutes.

Roadie is not intended to serve any other functions or fulfill any other purposes other than competing in the 2015 IEEE SoutheastCon competition.

## 1.4 Team Information

Name	Role
Brian Powell	Team Leader
Michael Philotoff	Software Manager
Alex Senopoulos	Developer
Brian Sterling	Hardware Manager

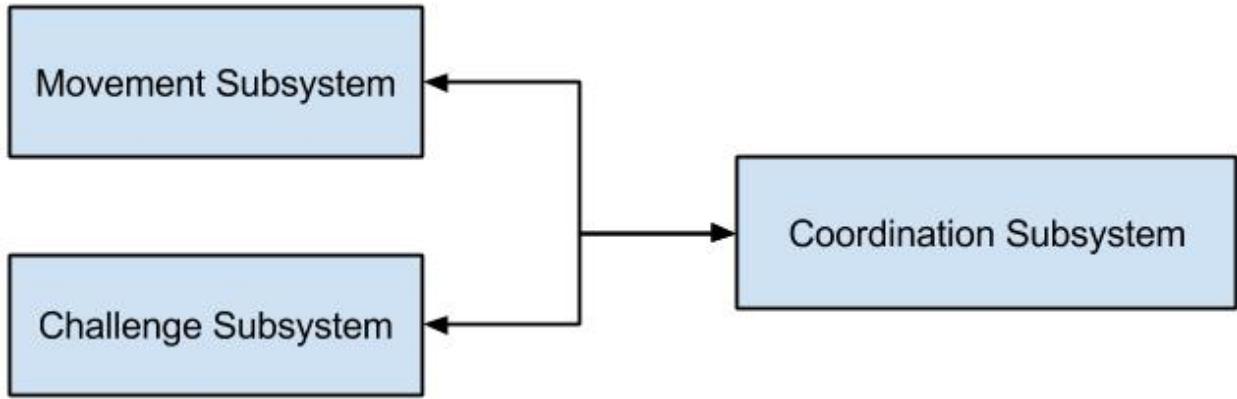
## 1.5 Overview

Section 1 of this document serves as an introduction to the system designed by AWTY. Section 2 serves as an overview as the system as a whole, including high level architecture of the system as well as descriptions of the subsystems. Section 3 includes all design considerations for the system, including assumptions, constraints and industry standards followed. Section 4 serves as a high level system design, including high level architecture, a state diagram and the hardware diagram of the system. Section 5 provides a decomposition of the Coordination subsystem including functional diagrams, hardware diagrams and a parts budget for the subsystem. Section 6 provides a decomposition of the Challenge subsystem including functional diagrams, hardware diagrams and a parts budget for the subsystem. Section 7 provides a decomposition of the Movement subsystem including functional diagrams, hardware diagrams and a parts budget for the subsystem. Section 8 provides the system test plan that will be used to test all components of the system.

Appendix B contains a diagram of the course that will be used during competition.

## 2. System Overview

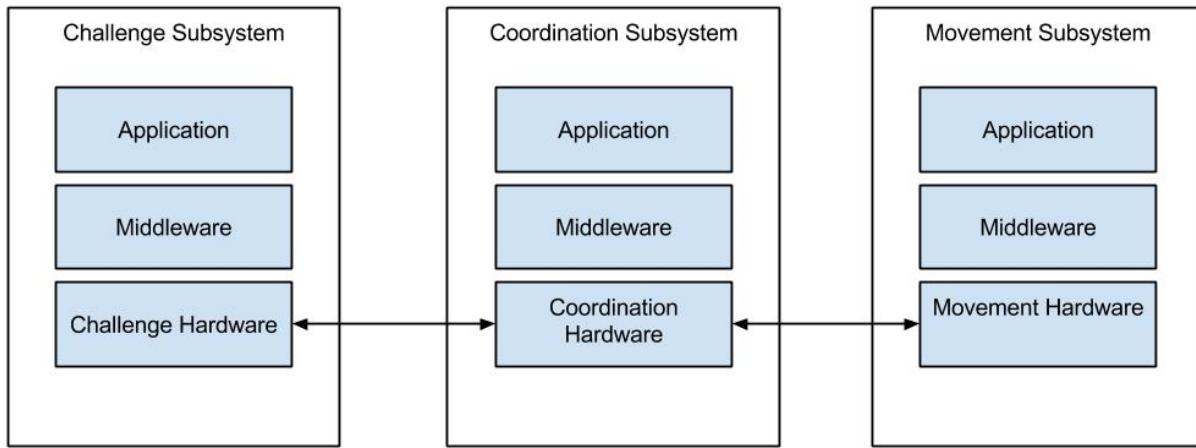
Roadie is broken down three main subsystems: (1) the Coordination subsystem, (2) the Challenge subsystem and (3) the Movement subsystem. The division of these subsystems is illustrated in **Fig. 1**.



**Fig. 1:** Division of Roadie into three subsystems.

The Coordination subsystem relays information to both the Challenge subsystem and the Movement subsystem. The components of Roadie were broken into subsystems based upon what other components they interacted with and what task they set out to perform. For example, all of the interactors for challenges (Simon, Etch-A-Sketch, Rubik's cube and playing card), were divided into the Challenge System since the interactors represent the means through which Roadie will physically manipulate the challenges. The Coordination System encompasses the microcontrollers as well as the object detectors and line sensors. The reasoning behind such decisions is that the object detectors and line sensors will send correction information that will be interpreted by the microcontroller to navigate Roadie to the right challenge as well as identify the correct challenge upon arrival. The Movement system is comprised of the motors and their associated wheels. This is due to the fact that the motors and wheels are responsible for moving the system to the intended destination.

The system architecture of Roadie is designed in a layered approach, depicted in **Fig. 2** below, in order to better divide the work being done and to aid in the conceptualization of the system design.



**Fig. 2:** High level description of the systems in Roadie.

The responsibilities of each system is shown in the sections below.

## 2.1 Coordination Subsystem

The Coordination subsystem is responsible for making sure that all communications get to the correct subsystem and that all systems are operating normally. The Coordination subsystem's applications are in the form of feedback from the sensors. This means that any information from the line sensors and the object detection sensors will have to be interpreted and handled by the subsystem. The interpretation will be handled by the Coordination subsystem's middleware or software. The software is able to interpret the data coming in from the corresponding sensors, be it line information or object information, and send it to the correct interface on its coordination hardware. These interfaces include serial interfaces on both microcontrollers as well as pulse width modulation (PWM) outputs on both microcontrollers.

## 2.2 Challenge Subsystem

The Challenge subsystem is responsible for interacting with all the challenges. This subsystem receives completion instructions from the Coordination subsystem and is able to interpret them correctly. The Coordination subsystem sends completion information to the Challenge subsystem over a physical interface, either serial or PWM. These interfaces then send information to the middleware of the Challenge subsystem, its software. The Challenge subsystem's software are functions to be able to complete the Etch-A-Sketch challenge, the Rubik's Cube challenge, the Simon challenge and the card challenge. The Challenge subsystem's functions are able to manipulate the correct challenge interactor to complete the challenge in the appropriate manner.

## 2.3 Movement Subsystem

The Movement subsystem is responsible for moving the chassis along the course, based on navigation instructions from the Coordination subsystem. The Movement subsystem includes the chassis, which is a vital part to making the entire system move. The Movement subsystem receives information from the Coordination subsystem over PWM signals. The Movement subsystem takes the received signals and translates them with its middleware in the form of software. The software determines the

correction function call, be it moving the system left or right for example, and then executes the call on its application layer. The application layer includes the motor controllers that drive the subsystem's motors.

## 3. Design Considerations

This section outlines the assumptions, dependencies and constraints imposed upon the system as a whole. Additionally, industry standards followed, safety constraints and considerations as well as environmental considerations are enumerated in this section.

### 3.1 Assumptions

During construction of the system, assumptions were made that affect the system as a whole. These assumptions can be found enumerated below.

#### 3.1.1 Operation

It is assumed that the system will only be operated for the purposes and in a manner for which it was designed for. That is to say that the system would not perform successfully were it to be used to fend off an attack from a silver back gorilla. The system has been designed with the sole purpose of competing in the 2015 IEEE SouthEastCon competition. As such, it is assumed that the system will only operate in such a venue. Furthermore, it is assumed that any operation of the vehicle will coincide with the rules and regulations outlined in [1]. Any deviations from the regulations will result in overall system failure.

#### 3.1.2 Rule Changes

It is assumed that rules governing the system will not change in such a manner that the system will be rendered inoperable. Should the rules change to such a degree, the system would ultimately fail as the design of the system would no longer fit the rules.

## 3.2 Dependencies

During system design and construction, design choices were made that have underlying dependencies. Such dependencies are shown in the section below.

#### 3.2.1 Intersystem Communication

The system is designed in a way that all subsystems rely upon each other to effectively and correctly compete. As such, it is important that all subsystems are working properly. If the system is unable to properly communicate amongst itself, the system will not be able to successfully complete the course.

#### 3.2.2 Power

Since all components of the system will draw power from the same power source, all of the components depend upon the power source starting with a full charge. Furthermore, all components are dependent upon the power source operating dependently and normally. Any abnormalities in the power source could prove disastrous for the entire system.

### 3.3 Constraints

During construction of the system, constraints were imposed upon the system which affected the overall system designed. These constraints can be found enumerated below.

#### 3.3.1 Size

According to the rules and regulations for the 2015 IEEE SouthEastCon competition put forth in [1], the system must fit within a one foot by one foot by one foot cube. As such, the system design had to reflect these size constraints. This affected the design choices and implementations since the system had to ultimately fit inside the aforementioned cube.

#### 3.3.2 System Power Draw

Since course rounds last five minutes according to [1], design constraints were imposed upon how much power the system could draw. If the system had very large electrical components, it is probable that the system would not last the entire five minute competition duration.

#### 3.3.3 Monetary

Since the system did not have an unlimited budget, monetary costs were a constraint placed on the system. While it may have been desirable to build the chassis out of titanium or carbon fiber to save on weight, the budget ultimately prohibited such a design consideration. Furthermore, electrical components and accessories were of a lesser accuracy compared to some of their more expensive counterparts due to monetary constraints.

#### 3.3.4 Technical Expertise

The technical expertise was another contribution to the complexity and overall design of the system. While it may have been desirable to have a system that would hover to help eliminate rolling resistance, or a rocket powered system to help with the speed at which the system ran, these designs were not feasible since they fell outside the technical abilities of the design team.

### 3.4 Industrial Standards Followed

In any industry, there are standards set forth so that work completed by one party can be easily understood, analyzed and modified by any other party with the same understanding of said standards. For our system in particular, special consideration was given to the standards of electrical, computer and software engineering.

#### 3.4.1 Wiring Colors

For the robot system we created there are many different items that are wired to batteries, microcontrollers, motors, and other system components. Many of these components are also polarity sensitive, which is to say that when current is applied in the wrong direction, the device breaks and is no longer useable. After several failed attempts at the “guess and check method”, it was decided that it was necessary to use the industry standard for wiring which is, black wires for negative and red wires for positive. This is crucially important since virtually every purchased component uses this standard.

### 3.4.2 Software Standards

The software industry standards being used by team AWTY are using proper naming schemes coupled with integrated agile testing. The testing is done in conjunction with development to produce a better working finish product. Furthermore, the use of unit test generation and execution shall be used in order to help verify functioning and reliable parts of Roadie. Additionally, the usage of runtime error detection is being used with the software to help find defects in the code whether it be logical or syntax errors. Finally, code analysis is being performed to help improve reliability, performance and adaptability.

## 3.5 Safety Constraints and Considerations

As the system will be operating in a public venue, safety is a very large concern. It is imperative that any operation or fault in the system will not cause harm or pose a threat to any member of the audience, nor any member of team AWTY. As such, safety considerations imposed upon the system can be found in the sections that follow.

### 3.5.1 Electrical Hazards

Since the system will be powered by electrical components, especially those which output a relatively high voltage, electrical shocks pose a problem if proper steps are not taken to help mitigate such events. To aid in mitigation efforts, all connections originating from the power source have been properly insulated. Furthermore, all ground connections and power connections have been implemented on opposing sides of the chassis to help mitigate accidental contact. All power sources have a single point of connection, which are also insulated. The single connection point helps to ensure that not only will it prove physically impossible to plug the power source in incorrectly, it also ensures that the power source will not be active if not plugged into the system. The final precaution taken in system design is constructing the system out of a non-conductive material. As the chassis has been constructed from wood, if any connection does come loose and makes contact with the chassis, the chassis will not have an electrical charge since wood is not a conductor of electricity.

### 3.5.2 System Heating

Since the system components such as the motor controllers can draw large amounts of current, there is a potential for the components to generate a large amount of heat. As such, the chassis has been designed with as much airflow as possible. The chassis is an open air design to aid in the dissipation of heat from the electrical components. Additionally, since the system will be driven by stepper motors, attention has been made to the amount of power supplied to the stepper motors, as they have the potential to heat very quickly. For this reason, voltage regulators have been installed to ensure that the voltage supplied to the system is sufficiently below the maximum allowed voltage, also assisting in preventing overheating.

### **3.5.3 Battery Nominal Voltage**

The power source for the system is a lithium polymer (LiPo) battery. LiPo batteries have a very volatile nature, and as such, much attention has to be made to their care and handling. It is for this reason that a battery voltage indicator has been used on the system. Should the battery fall below its nominal voltage of 14.4V, not only will total system power be diminished, but the cells inside the LiPo will be harmed. Continual harming of the battery's cells can cause the battery to become more unstable, creating a safety hazard. It is for this reason that the nominal voltage of the battery will be carefully monitored.

## **3.6 Environmental Considerations**

As students aspiring to enter the field of professional engineering, it is important that the environment is taken into consideration both before and during a project's life cycle. The environmental considerations enumerated below have been addressed during the construction of the system.

### **3.6.1 Component Waste**

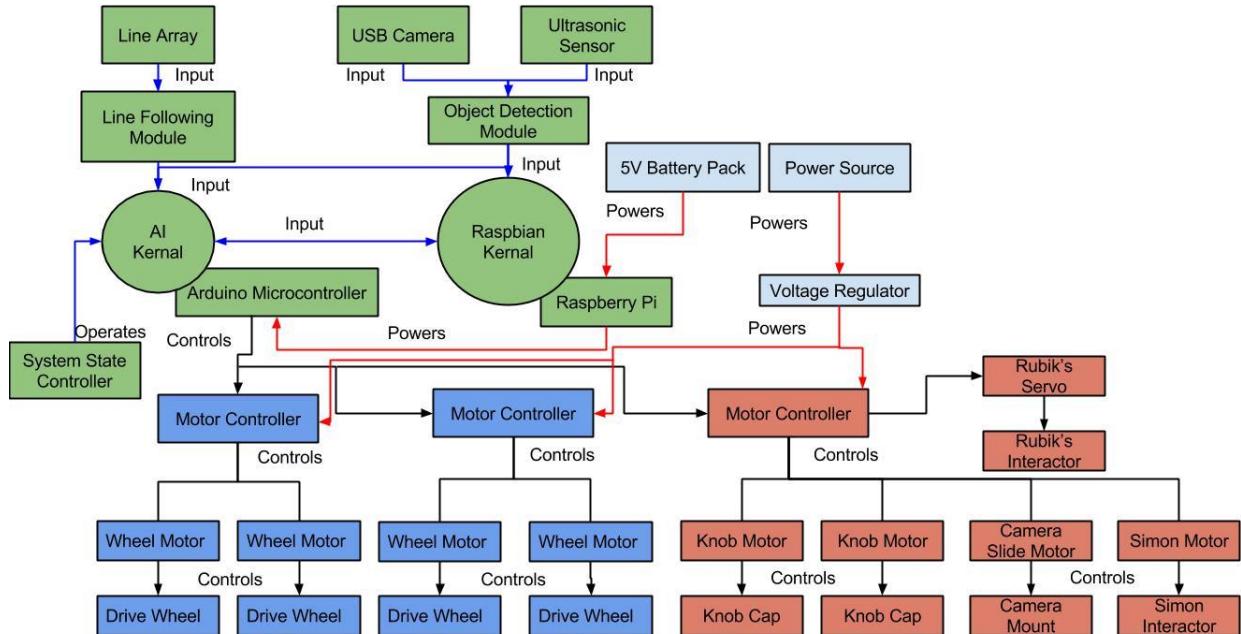
During construction, it is inevitable that components will be broken and or destroyed. As such, it is of the utmost importance that any destroyed component is disposed of properly. This helps to ensure that no hazardous material will be released into the environment.

## 4. System Design

This section outlines the flow of data through the system, as well as how different subsystems interface and interact with the entire system.

### 4.1 High Level Architecture

Roadie has been broken down into three subsystems: Coordination, Movement and Challenge. The flow of inputs, outputs and power between these three subsystems is illustrated in **Fig. 3** below.



**Fig. 3:** High level design of the system. The system has been divided into Coordination (green), Movement (blue) and Challenge (red) subsystems. Power flowing through the system is represented with a red line, input to the system is represented by a blue line and any output from the system is represented by a black line.

The system has been broken down into three subsystems, each with its own core components and responsibilities. Further discussion of the subsystems as well as their location the subsystem can be found in the sections below.

#### 4.1.1 Coordination Subsystem

The Coordination subsystem is responsible for reading in information from the sensors on the system (ultrasonic, camera and line array), interpreting the information and distributing it to the other subsystems. This subsystem is where all of the information will flow into, and out of. That is to say that any information related to course completion will go into this subsystem, at which point the subsystem will issue initiation commands to the other subsystems. This subsystem serves as the master controller for the whole system. All logic will reside within this subsystem.

The major components of this subsystem are a Raspberry Pi, and Arduino Mega, a line following array, a USB camera and an ultrasonic sensor. The sensors working in tandem with the microcontrollers allow the Coordination system to know what is occurring in the environment in which the system has been placed.

#### **4.1.2 Movement Subsystem**

The Movement subsystem is responsible for taking instructions from the Coordination subsystem, parsing them, and performing the appropriate action. For example, if the Coordination subsystem tells the Movement system that it needs to shift the chassis left to be lined up with the Etch-A-Sketch interactor, the Movement system must be able to determine how far to shift left. Additionally, if the Coordination system tells the Movement system that it has overshot the line by some percent, the Movement system must be able to turn this percentage into a quantifiable distance the chassis needs to move. This subsystem is responsible for any movement regarding the chassis. The chassis shall not move in any direction unless directed by the Coordination subsystem.

The major components of this subsystem are the chassis, the four drive motors and their associated wheels, and the motor shields for the drive motors.

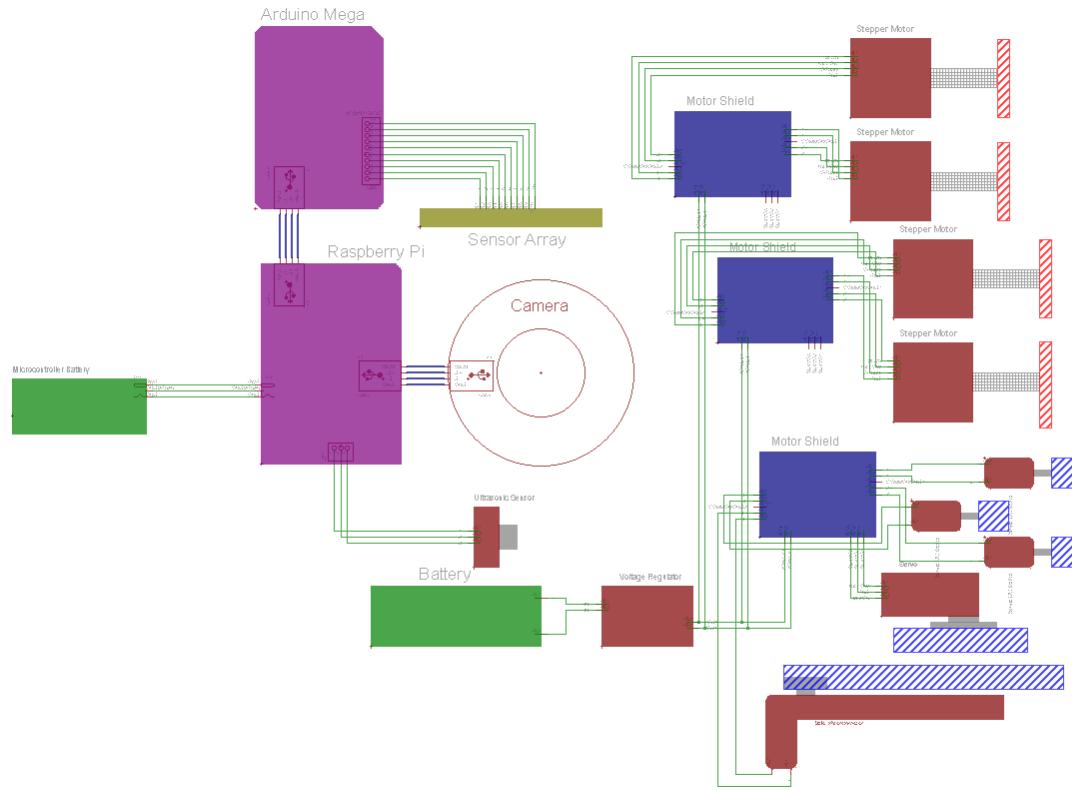
#### **4.1.3 Challenge Subsystem**

The Challenge subsystem is responsible for taking instructions from the Coordination subsystem, parsing them, and performing the appropriate action. For example, if the Coordination subsystem tells the Challenge subsystem it has arrived at the Etch-A-Sketch challenge, the Challenge subsystem must be able to determine what needs to happen to correctly interact with the Etch-A-Sketch. This philosophy is true for all challenges. Upon notification by the Coordination system, the Challenge system must correctly determine what challenge to initiate and when it has completed said challenge. All interactions with challenges will be controlled solely by the Challenge subsystem. No other subsystems will attempt to interact with the challenges.

The major components of this subsystem are the interactors for the challenges (Etch-A-Sketch, Rubik's cube, Simon and playing card), as well as the associated motor shield.

## 4.2 System Hardware Diagram

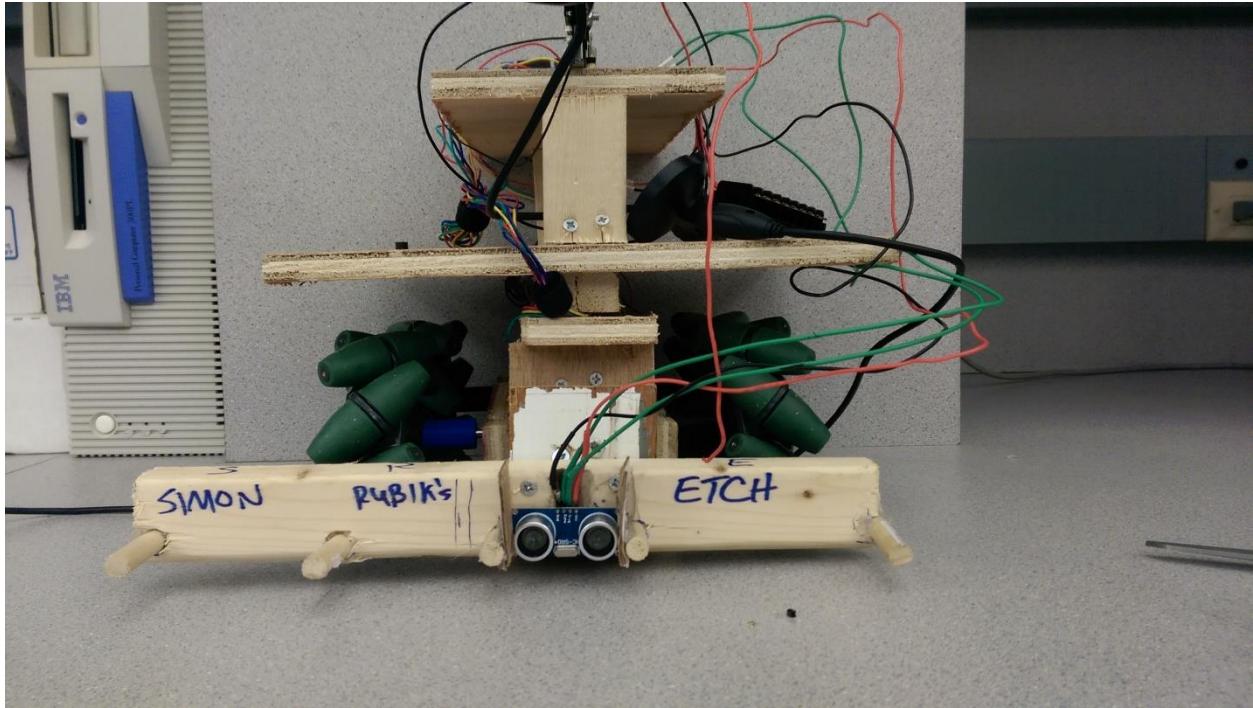
A diagram of all the hardware and their associated interfaces and connections for the system at large is shown in **Fig. 4** below.



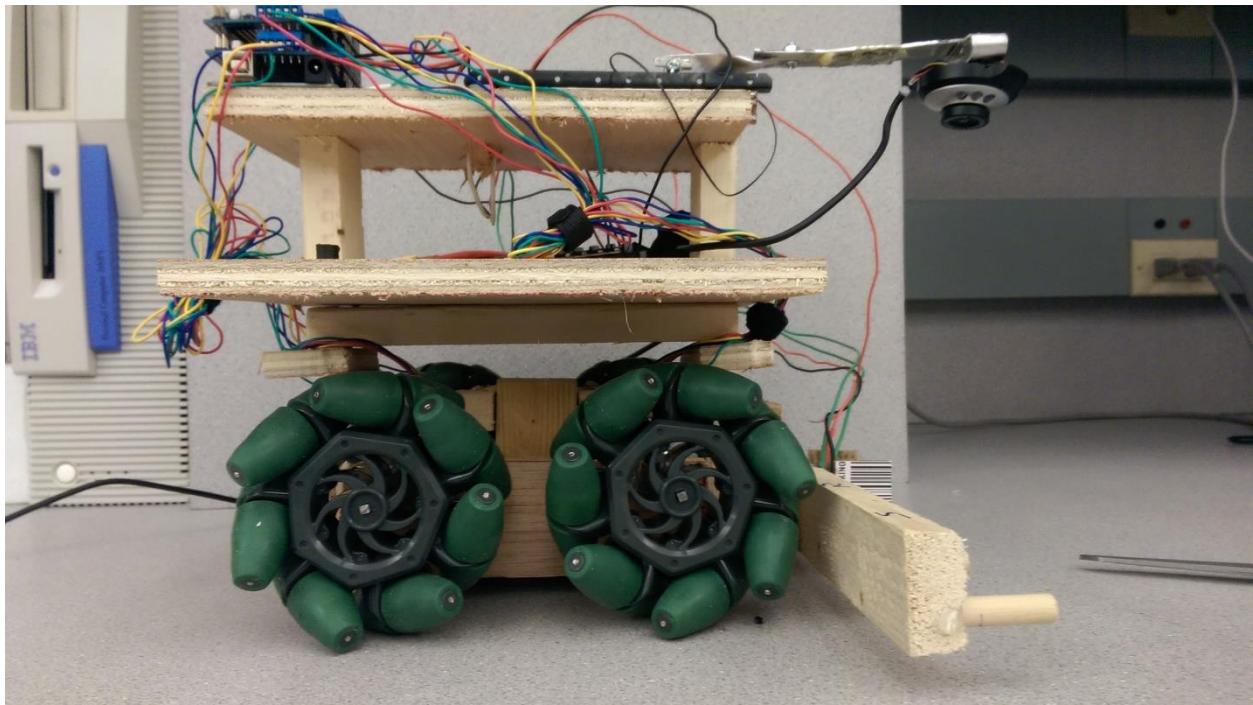
**Fig. 4:** The above diagram shows how all the components of the system connect to one another.

In **Fig. 4** the completed hardware structure is diagramed so that all components can easily be identified with their corresponding sub system and their dependency on one another. It is important to note that on the physical system, each one of the motor shields stacks on top of one another and that whole grouping stacks onto header pins located on the Arduino Mega. In the above diagram these connections were omitted so that the diagram is easier to read and identify connections. It is also important to note that in the diagram, blue connections represent a USB connection while the green connections are simple wire connections. The red hatched areas define the wheels connected to the stepper motor shafts and the blue hatched areas represent non electrical components that will be used in completing the challenges.

The entire system and its associated hardware components is depicted in **Fig 5** and **Fig 6** below.



**Fig 5:** The picture above is a front view of the whole system. This includes the chassis, all electrical components and all mechanical components.



**Fig 6:** The picture above is a front view of the whole system. This includes the chassis, all electrical components and all mechanical components.

### 4.3 System State Definitions

In order to better define transitions and states that the system will occupy, **Table 1** has been constructed, providing both the state name, and the description of the associated state.

State Name	State Description
Approach	The state in which the system will be once it has entered a challenge area. The approach stage will consist of the system placing itself $4.2\text{ cm} \pm 0.1\text{ cm}$ from the challenge.
Challenge Abortion	The state in which the system will enter upon incorrectly interacting with a challenge. When the system enters this state, it will immediately halt execution of the current challenge, exiting the challenge area and proceeding to the line following state.
Challenge Completion	The state in which the system will enter upon successfully completing a challenge. This system will remain in this state until exiting the challenge area.
Challenge Identification	The state in which the system attempts to identify the challenge it has arrived at. The system remains in this state until a positive identification.
Challenge Interaction	The state in which the system will attempt to complete a challenge. The system will remain in this state for as long as it is interacting with a challenge.
Challenge Misidentification	The state in which the system will enter upon falsely identifying the challenge it has arrived at. Should the system enter this state, this will represent a catastrophic failure. The system will proceed to the challenge abortion state.
Etch-A-Sketch	The state in which the system will attempt to complete the Etch-A-Sketch challenge.
Failed Approach	The state in which the system will enter upon stopping closer than or further than $4.2\text{ cm} \pm 0.1\text{ cm}$ . The system will proceed back to the approach state.
Finish	The state in which the system will enter upon crossing the finish line. The system will cease all movement.
Line Abandonment	The state in which the system will enter upon failing to proceed along the guidance tape. Should the system enter this state, the system will reverse direction to the last known location of the guidance tape.
Line Following	The state in which Roadie is following the Scotch Blue Painter's tape located on the competition area.
Playing Card	The state in which the system will attempt to complete the playing card challenge.
Rubik's Cube	The state in which the system will attempt to complete the Rubik's Cube challenge.
Simon	The state in which the system will attempt to complete the Simon challenge.
Staging	The state in which the system commences operation. This state will last from the time the system is placed inside the starting area, until the LED in the starting area is turned off.
Zone Identification	The state in which the system will enter upon recognizing a challenge zone or finishing line.

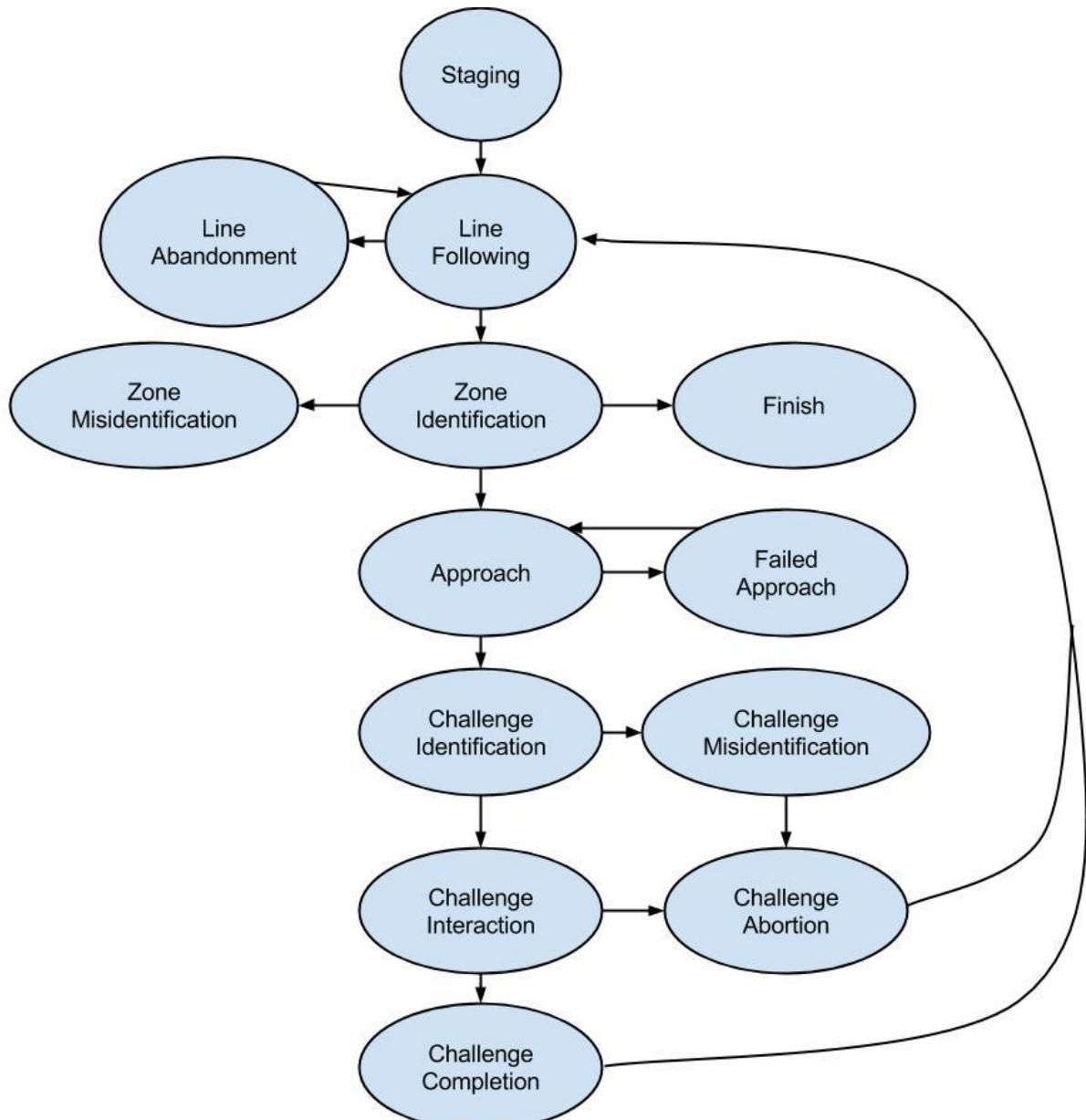
Zone Misidentification	The state in which the system will enter upon failing to recognize a challenge zone. If the system were to enter this state, that would represent a catastrophic failure, resulting in termination of the round.
------------------------	--

**Table 1:** States that the system will occupy with their accompanying description.

By using **Table 1**, it was possible to construct the state diagrams shown in the sections below.

#### 4.4 System State Diagram

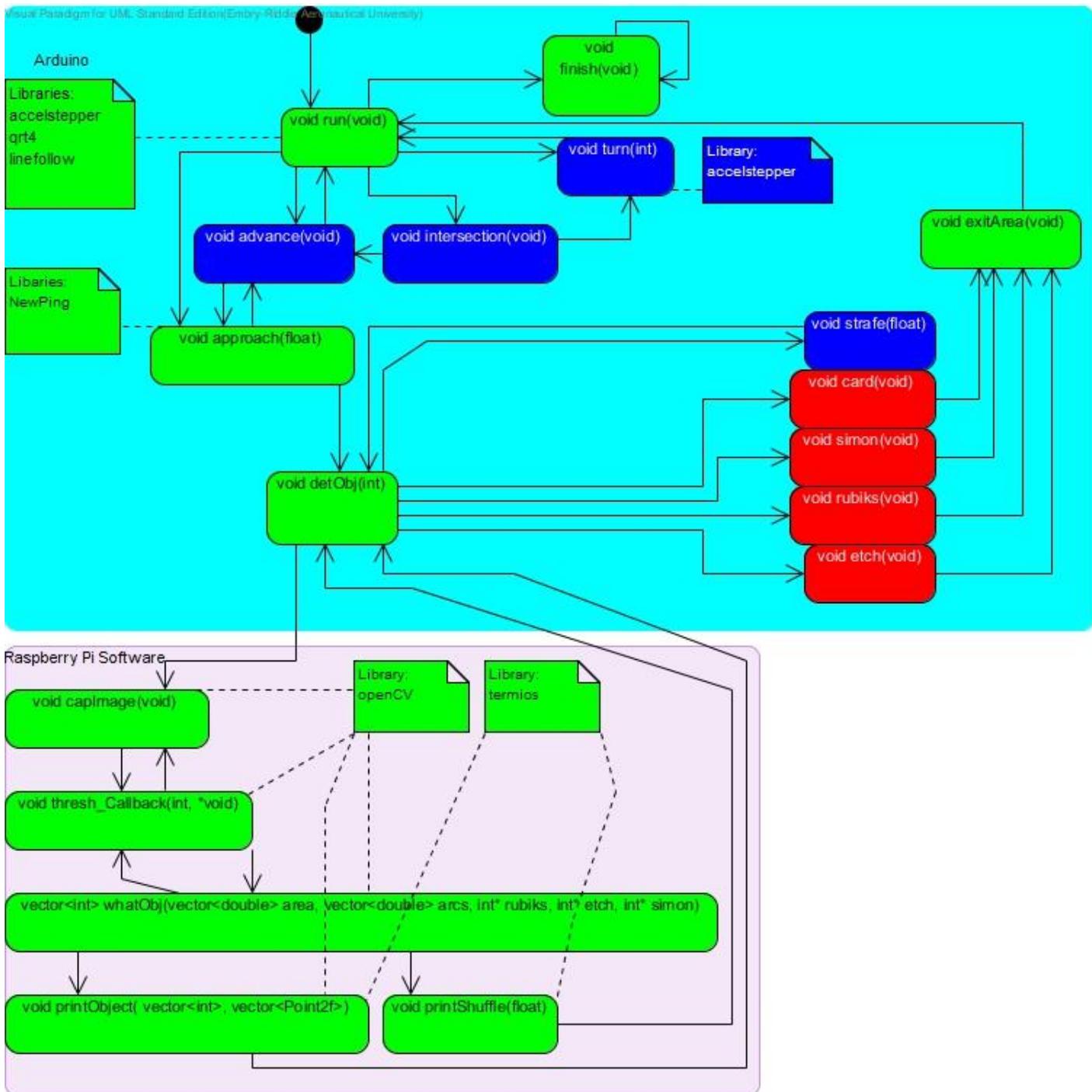
**Fig. 7** below shows the states that the system will be in, and how the system will transition from state to state.



**Fig. 7:** State diagram for Roadie

## 4.5 System Function Diagrams

In **Fig. 8** below, the function diagram for the entire system is shown.



**Fig. 8:** Function diagram showing the major function calls in the system. Blocks green in color represent functions in the Coordination subsystem, blocks in blue represent the Movement subsystem and blocks in red represent the Challenge subsystem. The cyan rectangle is software that resides on the Arduino and the pink rectangle is software that resides on the Raspberry Pi.

The function diagram is further explained by subsystems in the sections below.

#### 4.5.1 Coordination Subsystem

As **Fig. 8** shows, the main function that will constantly be running is the **run** function. The **run** function will be able to determine what must happen (**approach**, **finish**, **turn**, **intersection** or **advance**) based upon the feedback from its sensors. Should the sensors indicate a challenge zone, the **approach** function will be called. The **approach** function will call **advance** until Roadie is close enough to the challenge to make a reliable detection. Once the Roadie is close enough to the challenge to make a reliable detection, the Arduino, via the **detObj** function tells the Raspberry Pi to begin image capture. The Pi captures an image with **capImage**, taking the captured and performing a thresh operation on the image. The thresh operation filters out the unwanted items (think of it as background noise) from the image. The **whatObj** function within the Pi source code is called until a positive image detection has been made. Once control is returned to the Arduino, the Arduino activates the appropriate subsystem. After the subsystem has completed its task, control is once again returned to the Coordination subsystem to either exit the challenge area (challenge is complete) via the **exitArea** function or to re-detect the image (the system needed alignment) via the **detObj** function. Once the finish line is detected, the **finish** function is called to halt all movement.

#### 4.5.2 Movement Subsystem

Should the **run** function detect a turn, it will call the **turn** function, a part of the Movement subsystem. The Movement subsystem will then be able to determine how much of a turn is needed and in which direction, executing the turn in the process. Furthermore, the Movement subsystem will be able to determine if it has arrived at an intersection and handle this case properly via the **intersection** function. The **intersection** function will be able to determine if Roadie needs to turn at the intersection, turning via **turn** or advance through the intersection via the **advance** function.

If the Coordination subsystem determines that alignment relative to the challenge is needed, it will call upon the **strafe** function, part of the Movement subsystem, to move the chassis the required amount to be in perfect alignment with the challenge.

#### 4.5.3 Challenge Subsystem

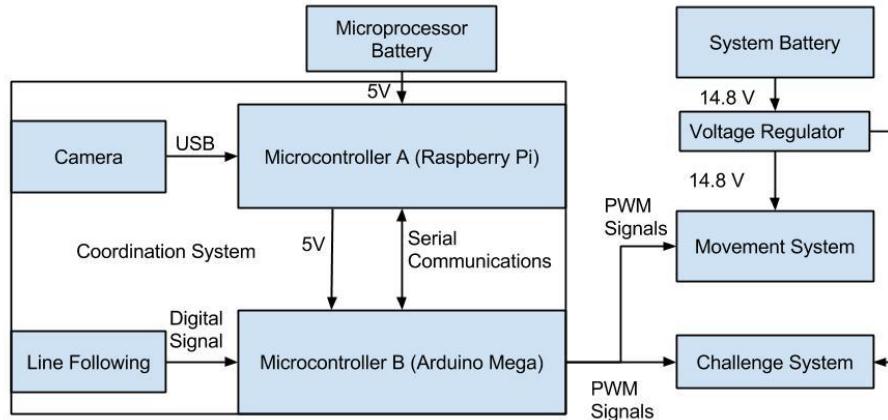
When the **detObj** function has received the object it has arrived at, it calls upon the appropriate function within the Challenge subsystem. For example, if the challenge has been identified to be the Rubik's cube, the **rubiks** function will be called. Similarly, the **card**, **simon**, and **etch**, functions will be called when the system arrives at their respective challenges. Once the challenge has been completed (successfully or not), control will be yielded to the Coordination subsystem via the **exitArea** function to proceed back along the course.

## 5. Decomposition of Coordination Subsystem

The architecture, requirements, use cases, sequence diagrams and requirements traceability matrix for the Coordination subsystem are included in this section.

### 5.1 Subsystem Architecture

**Fig. 9** below, better illustrates the communications that occur amongst the subsystems in Roadie.



**Fig. 9:** Decomposition of Coordination subsystem for Roadie.

As depicted in **Fig. 9**, the Coordination subsystem is composed of two microcontrollers, both communicating with each other over a serial interface. The Arduino Mega is the primary micro controller, interpreting inputs from the reflectance array (line following) and issuing commands based on values received.

The coordination subsystem is responsible for the navigation and challenge identification process of Roadie. The subsystem consists of object detection, line following, and subsystem coordination.

Object detection allows Roadie to identify what object it has arrived at. By using object detection, Roadie will be able to determine what challenge needs to be completed as well as how to align with the challenge.

Line following allows Roadie to traverse the competition area. Roadie will use its line following capabilities to navigate from the starting area to the various game stations.

Subsystem coordination allows Roadie to know what each subsystem is doing at any given moment. This will aide in ensuring that Roadie will successfully complete all tasks.

#### 5.1.1 Assumptions

During operation, it is assumed that Roadie will begin with sufficient battery to complete the course. If the system is started with a battery that is not charged enough, the system will not be able to successfully complete the course.

It is also assumed that Roadie will be operating in the competition area shown in **Fig. 34**, and only the competition area shown in aforementioned figure. Since Roadie has been designed to compete in IEEE Southeast Con 2015, any course modifications will render the system inoperable. Furthermore, it is assumed that the course will be free of obstructions and obstacles. Any obstacles will prevent Roadie from completing the course. Assumptions have also been made regarding the course construction. It is assumed that the course will be constructed according to the methods and materials outlined in [1].

The object detection in the Coordination subsystem will only operate on the challenges outlined in [1]. Any modifications to the challenges or any different items introduced into the challenge area will create a problem for Roadie.

Assumptions have been made regarding the subsystem coordination. If one of the subsystems gives throws an unexpected error, or an error that the master controller is not prepared for, the entire system will fail.

### 5.1.2 Dependencies

The Coordination subsystem depends on the Challenge subsystem and Movement system to relay information back to it so that it may guide Roadie in course completion. If this connection is broken or fails, Roadie will ultimately fail.

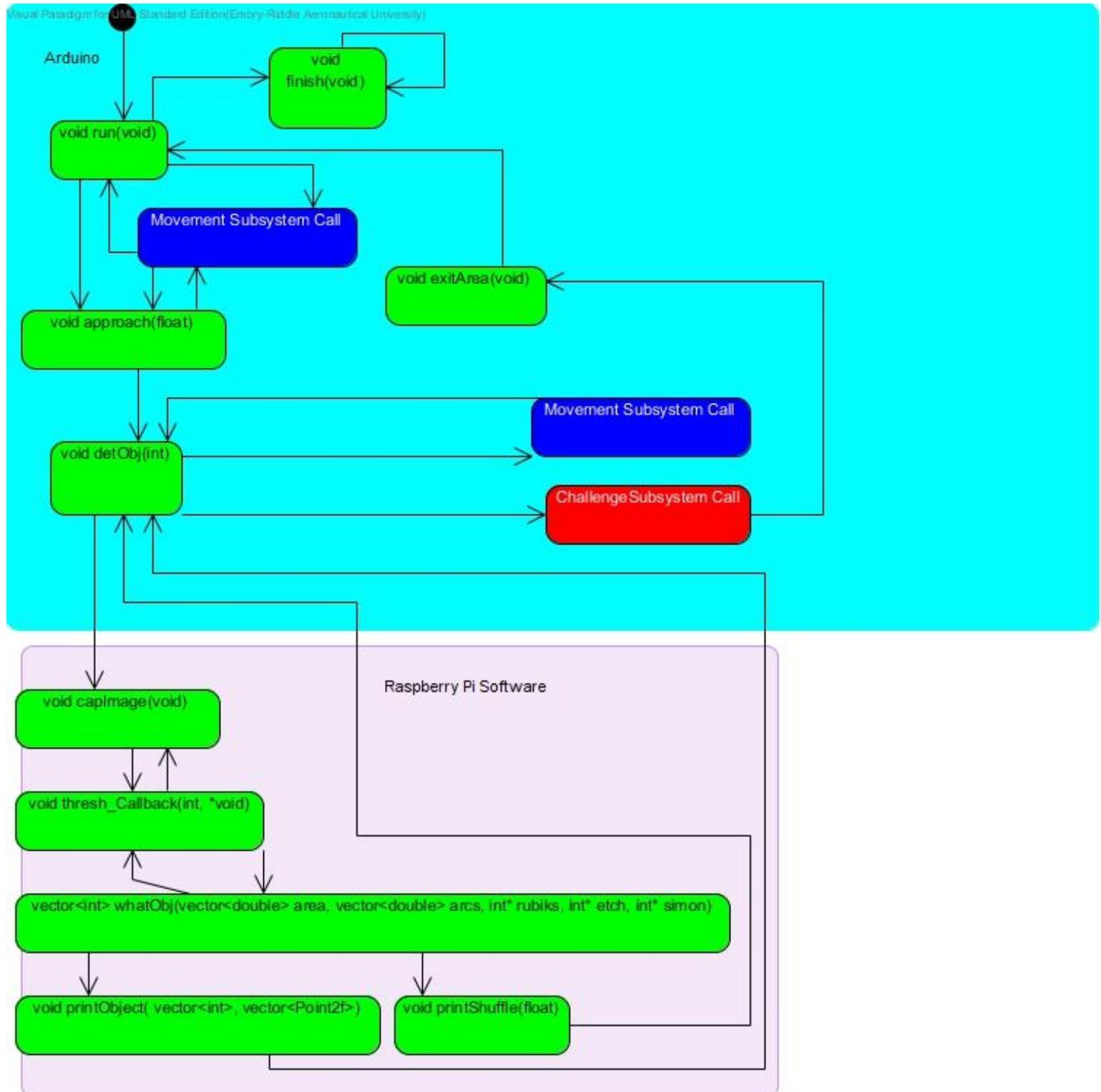
Problems arising from dependencies include any mechanical failure which would render the system inoperable. Furthermore, electrical issues or corrupted data will cause the system to fail the competition round.

### 5.1.3 Constraints

The Coordination subsystem is constrained by the clock speed of the selected microcontroller as well as the interfaces through which the other subsystems will communicate. The system is also constrained by the rules and regulations laid forth in [1]. All code and logic in the Coordination subsystem must fit within the system memory of the microcontrollers.

## 5.2 Functional Diagrams

**Fig. 10** shows the function calls and procedure flows through the Coordination subsystem.



**Fig. 10:** Procedure flow through the Coordination subsystem.

When Roadie is powered on, the Coordination subsystem becomes active. From here, the **run** function coordinates all the happenings of Roadie. The **run** function constants reads the sensors and sends commands to the Movement subsystem to advance Roadie appropriately. For instance, should **run** receive feedback indicating that a turn is needed, it will call upon the Movement subsystem to execute the turn. Once the **run** function has determined that a challenge zone has been reached, **run** will call upon the **approach** function. The **approach** function will use the ultrasonic sensor on Roadie to determine how far Roadie is from the object. The **approach** function will continuously call upon the Movement subsystem to progress Roadie forward until the challenge is in range. Once the challenge is in range, the **detObj** function will be called. The **detObj** function will send a serial signal to the Raspberry Pi to activate the camera and begin capturing images of what is in front of it. The **detObj** function will perform a blocking serial read until it receives data from the Raspberry Pi.

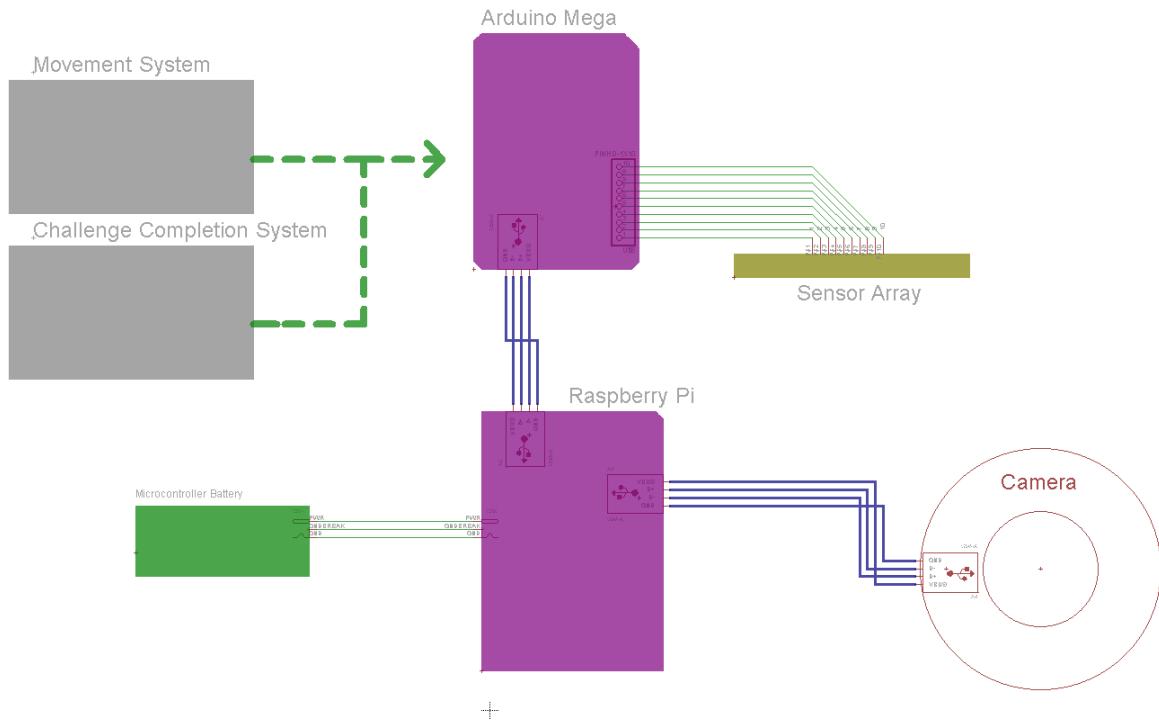
Once the Raspberry Pi has received a signal from the Arduino to begin image capture, it will execute the **capImage** function. This function will capture a single frame off of the camera and begin image processing. Image processing is initiated via the **thresh\_Callback** function. The **thresh\_Callback** function takes the captured image and turns it into a matrix of points. From here, the **thresh\_Callback** calls **whatObj** to determine what object is in front of the camera. If **whatObj** does not find any object, control is passed back to **thresh\_Callback**, which returns control to **capImage** to capture another frame. Conversely, if an image is found, **whatObj** determines the alignment of the object relative to Roadie. If the challenge is not aligned with the center of Roadie, it sends a signal to **detObj**, signifying that Roadie must move. If Roadie must move, the entire image capture process will be restarted. If Roadie is aligned with the object, **printObject** is called to tell **detObj** what object has been found.

Once **detObj** has control returned, it determines whether or not Roadie needs alignment. If Roadie required alignment, **detObj** calls upon the Movement subsystem to move Roadie accordingly. After Roadie has been moved, control is handed back to the Coordination subsystem to re-detect the object. If Roadie does not require alignment, control is passed to the Challenge subsystem to complete the appropriate challenge. After the challenge has been completed, control is handed back to the Coordination system via the **exitArea** function.

If the **run** function detects a finish area, run will call upon the **finish** function. Roadie will remain in this function until it has been powered off.

### 5.3 Hardware Diagrams

The hardware components in their associated connections for the Coordination subsystem are shown in **Fig. 11** below.

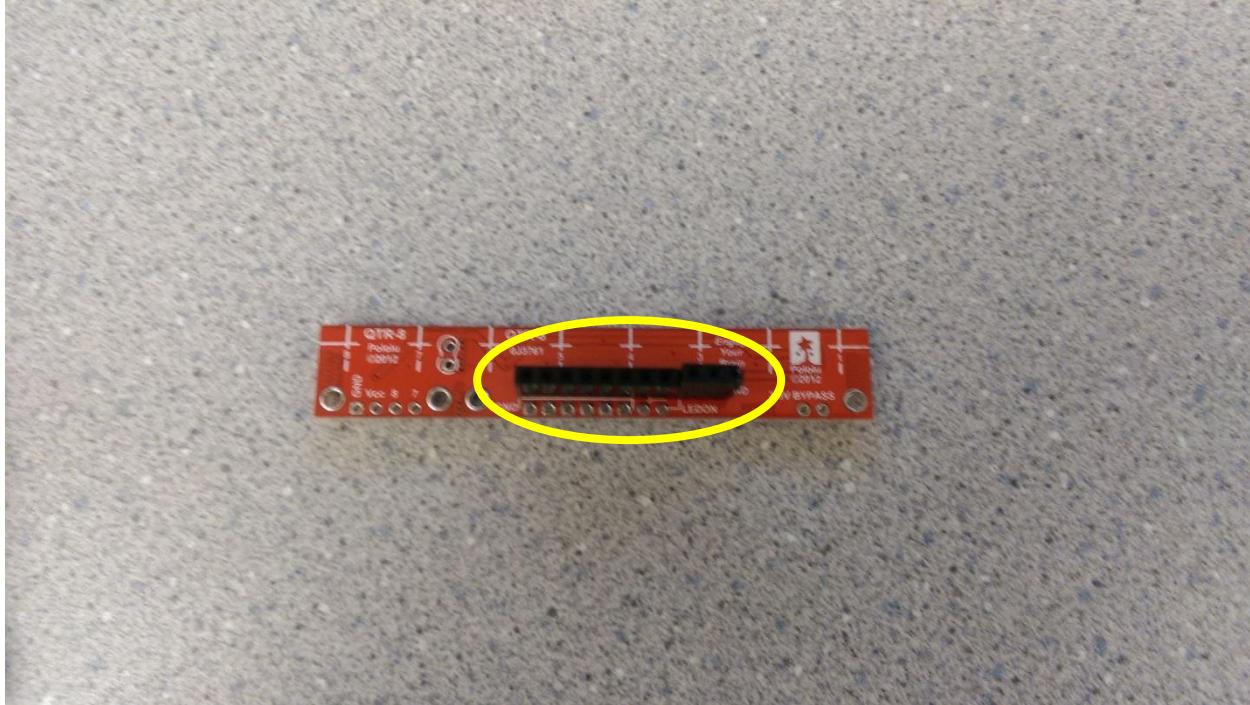


**Fig. 11:** Diagram which shows the electrical connections between the components of the Coordination subsystem.

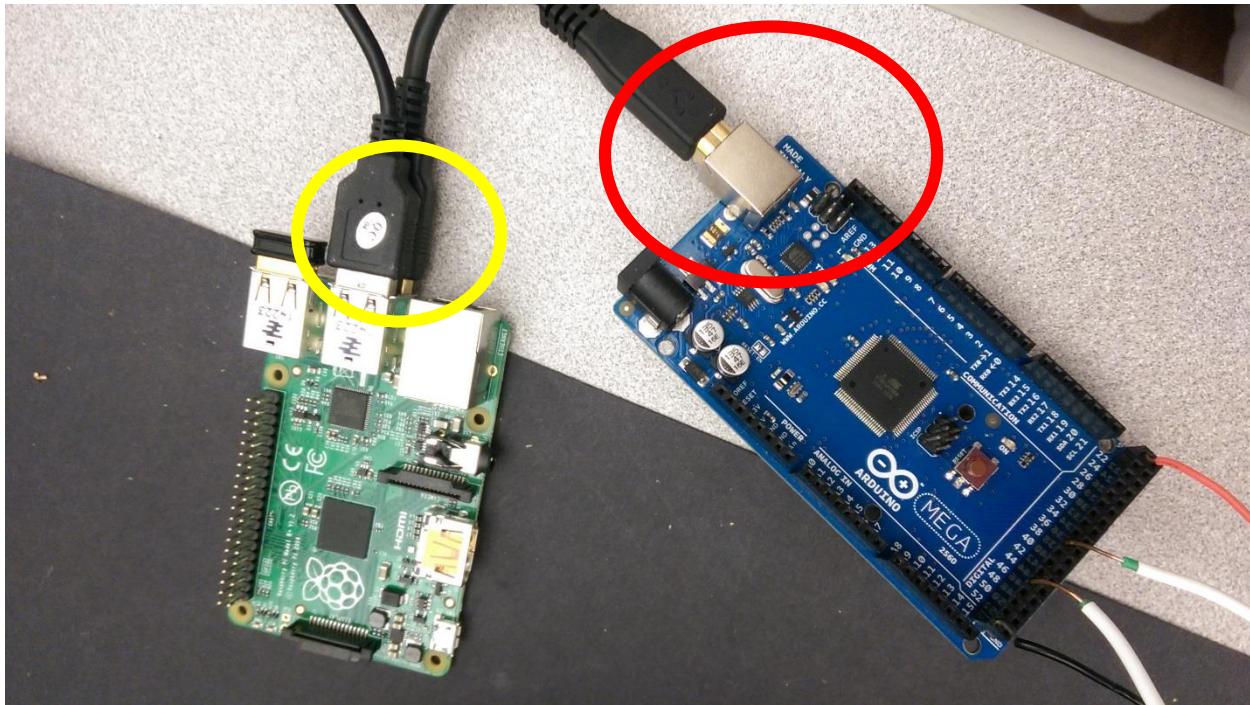
The diagram above shows the connections between the microcontrollers and the components of the system which coordinate movement. All connections in blue are USB connections, and all connections in greens are simple wire connections. It is also important to note that the device and wires connecting the battery and the Arduino is a barrel plug adapter.

## 5.4 Interfaces

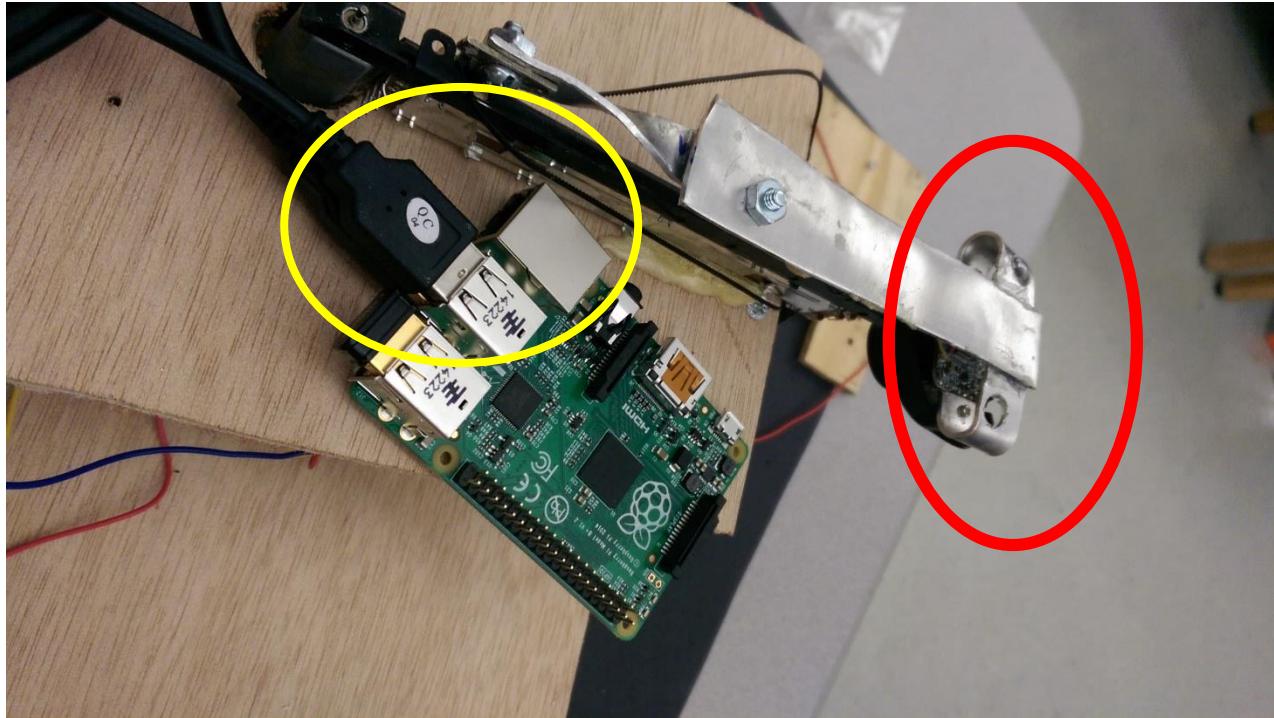
The hardware components used in the Coordination subsystem as well as their associated interfaces are shown in **Fig. 12**, **Fig 13**, **Fig. 14** and **Fig. 15** below.



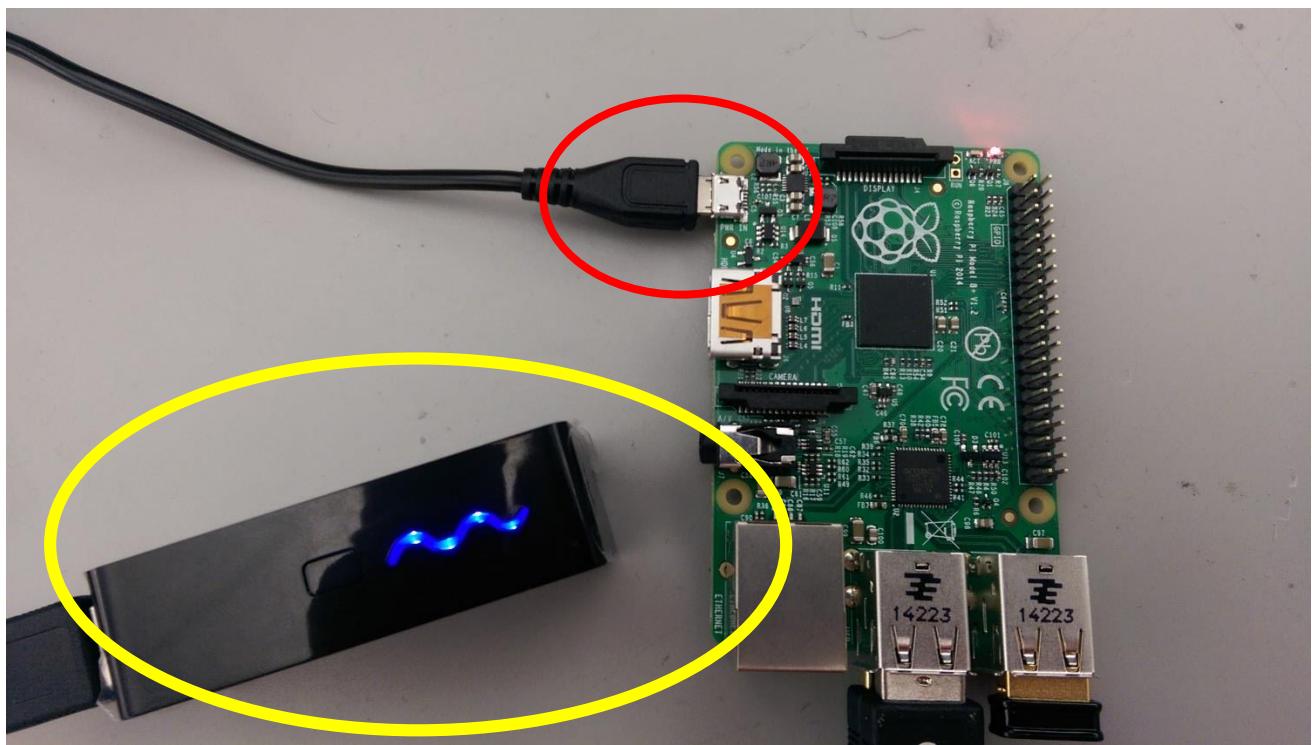
**Fig. 12:** The picture above shows the sensor array used in the coordination system. The yellow circle marks the pin headers which are used to connect the wires from the Raspberry Pi.



**Fig 13:** The picture above shows the USB connection between the Arduino Mega and the Raspberry Pi. The yellow circle indicates the USB connector to the Raspberry Pi and the red circle indicates the USB connector to the Arduino Mega



**Fig. 14:** The picture above shows the camera system and its connection to the Raspberry Pi. The yellow circle indicates the USB connection from the camera to the Raspberry Pi. The red circle indicates the Camera.



**Fig. 15:** The above picture shows the connection between the microcontroller battery and the Raspberry Pi. The yellow circle indicates the microcontroller battery and the red circle indicates the connection between the battery and the Raspberry Pi.

## 5.5 Parts Budget

This section will cover the parts budget for the coordination subsystem. These components are vital in ensuring that coordination between all of the subsystems is accurate and efficient. The different components used in the subsystem will be outlined, and then a cost matrix will show a cost analysis involving subsystem components and the sources of funding.

### 5.5.1 Components

This section will give a short reasoning for the existence of each component in the subsystem. All of these aid in satisfying Roadie's Coordination requirements.

#### Raspberry Pi

The Raspberry Pi was used for image processing purposes. The combination of this component and open CV allows the system to identify challenges for the system. The component aids in satisfying requirements COR 6 through COR 10. [2]

#### Arduino Mega

The Arduino Mega is used to interface with the hardware components in the system. This component aids in satisfying requirements COR 11 and COR 12. [2]

#### Fosman USB Webcam

This camera is the hardware component used to gain raw image data for the Raspberry Pi. This component works in concert with the Raspberry Pi in order to aid in satisfying requirements COR 6 through COR 7.1. [2]

#### SainSmart Ultrasonic Distance Sensor

This component is used to determine the distance from the piece of hardware to an object ahead of it. It is used to allow the system to correctly distance itself from each challenge. This component aids in satisfying requirement COR 4. [2]

#### QTR-8A Sensor Array

This component is used to allow the system to follow the course guidance tape. It uses eight analog QTR sensors which allow greater flexibility for movement software. This component aids in satisfying requirements COR 2 and COR 3. [2]

#### 5 Volt Battery Pack

This component is used to provide portable power for the Raspberry Pi and Arduino Mega. This component satisfies requirement COR 13. [2]

### 5.5.2 Cost Matrix

This section provides a table which includes each component and its cost per unit. It shows which components were department funded and which components were funded by the team. The costs of each unit can be seen in **Table 2** below.

Description	Unit Cost	Department Funded Units	Team Funded Units	Total Department Cost	Total Team Cost
Raspberry Pi	\$ 35.89	0	1	\$ -	\$ 35.89
Arduino Mega	\$ 38.48	0	1	\$ -	\$ 38.48
Fosman USB Webcam	\$ 7.99	1	0	\$ 7.99	\$ -
SainSmart Ultrasonic Distance Sensor	\$ 7.03	0	1	\$ -	\$ 7.03
Pololu QTR-8A Sensor Array	\$ 9.95	0	1	\$ -	\$ 9.95
5 Volt Battery Pack	\$ 19.99	0	1	\$ -	\$ 19.99
Totals:				\$ 7.99	\$ 111.34

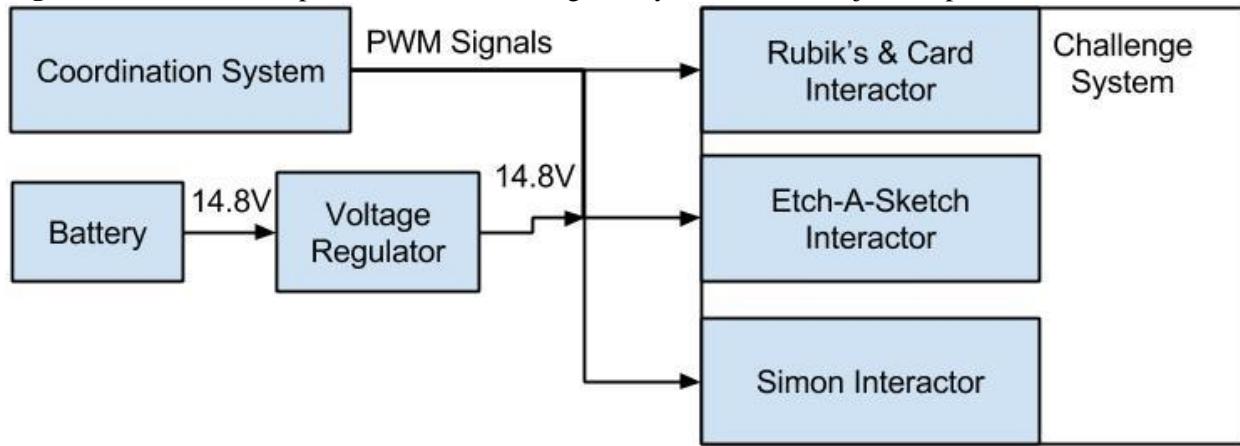
**Table 2:** Cost of hardware for Roadie's Coordination subsystem.

## 6. Decomposition of Challenge Subsystem

The architecture, requirements, use cases, sequence diagrams and requirements traceability matrix for the challenge system are included in this section.

### 6.1 Subsystem Architecture

**Fig. 16** shows the decomposition of the Challenge subsystem into its major components.



**Fig. 16:** Decomposition of Challenge subsystem for Roadie.

The challenge subsystem is responsible for completing all the challenges listed in [1]. This subsystem includes a Rubik's & Card interactor, an Etch-A-Sketch interactor and a Simon interactor.

Interacting with the Rubik's cube means that Roadie will attempt to turn one row of the Rubik's cube 180 degrees. Roadie will be able to positively identify the Rubik's cube and position itself over the Rubik's cube. Interacting with Simon means that Roadie will play Simon for 15 seconds, correctly identifying and pressing the illuminated segments. Interacting with the Etch-A-Sketch means that Roadie will successfully draw "IEEE" on the Etch-A-Sketch. Interacting with the playing card means that Roadie will successfully pick up a playing card and carry it across the finish line.

#### 6.1.1 Assumptions

It is assumed that all challenges will be exactly as described in [1]. Furthermore, it is assumed that all of the challenges will perform as expected. That is to say that it is assumed that the Etch-A-Sketch knobs will perform normally. They will not have encountered unexpected wear during repeated competition rounds. Furthermore, the Rubik's cube will be able to be rotated without an extraordinary amount of effort. If the Rubik's cube requires more effort to be twisted than expected, Roadie will be unable to twist the cube. If the segments on Simon do not illuminate properly after multiple course rounds, Roadie will be unable to properly identify Simon.

It is assumed that Roadie will operate in the competition area shown in **Fig. 34**.

### 6.1.2 Dependencies

The Challenge subsystem is wholly dependent upon the Coordination subsystem. The Coordination subsystem notifies the Challenge subsystem as to what challenge it has arrived at. From here, the Challenge subsystem will determine which interactor to activate. Without notification from the Coordination subsystem, the Challenge subsystem will be unable to perform its task. Furthermore, the Challenge subsystem relies on the challenges being exactly as described as laid forth in [1].

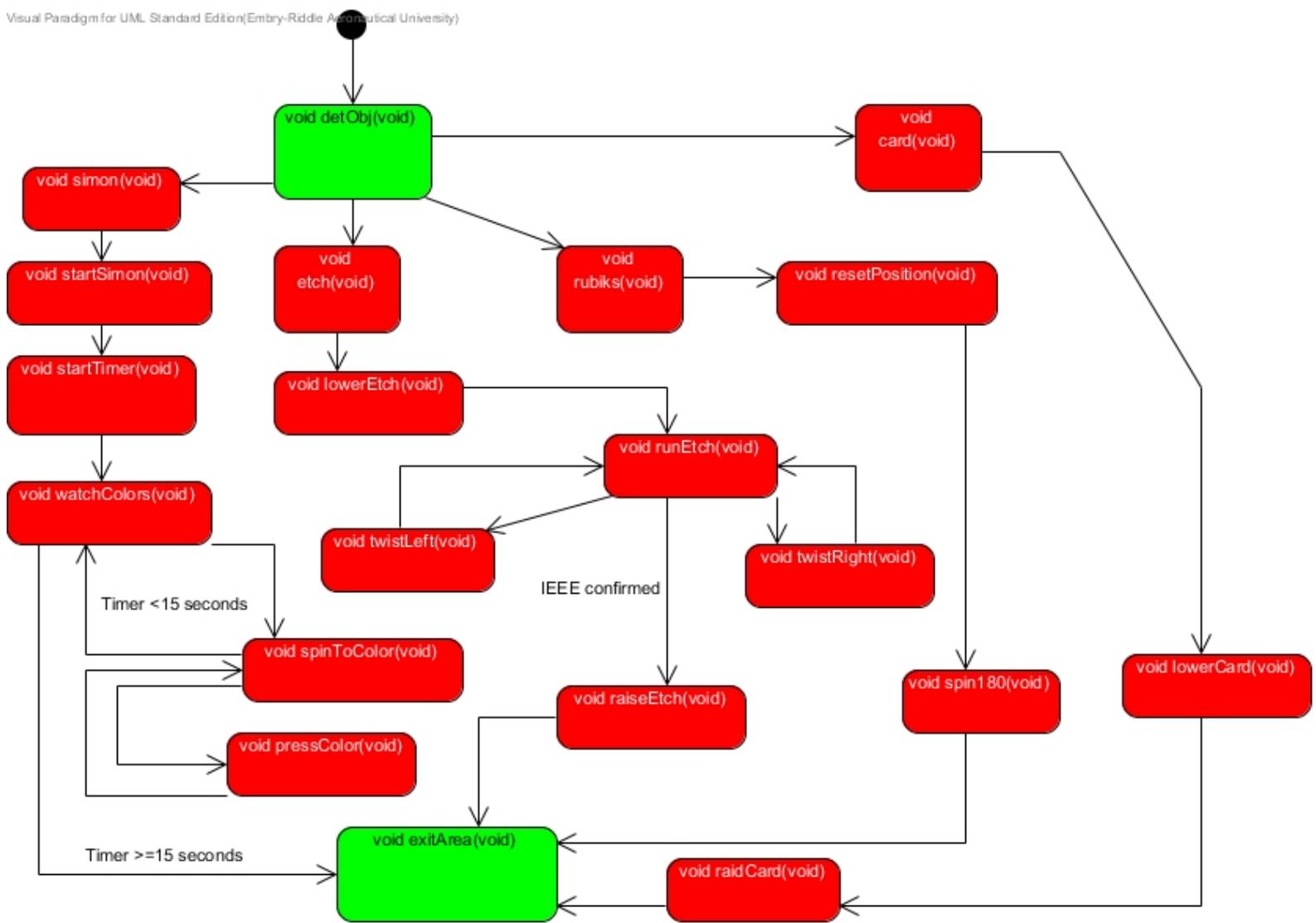
### 6.1.3 Constraints

The Challenge subsystem is constrained by the clock speed and memory in the microcontrollers. The Challenge subsystem must be able to coincide with the code and logic from the other subsystems. Additionally, all hardware being used by the Challenge subsystem must fit on the chassis in a manner in which it will not interfere with the other systems. The Challenge subsystem must abide by all rules in regulations laid forth in [1].

## 6.2 Functional Diagrams

**Fig. 17** shows the function calls and procedure flows through the Challenge subsystem.

Visual Paradigm for UML Standard Edition(Embry-Riddle Aeronautical University)



**Fig. 17:** Function diagram for the Challenge subsystem.

When the Challenge subsystem is activated by the Coordination subsystem, it calls the appropriate function based upon what object the Coordination subsystem has found. Further explanation of the procedure call can be found in the sections below.

### 6.2.1 Simon Detected

If the Coordination subsystem has detected the Simon challenge, the Challenge subsystem will initiate the Simon procedure flow. This flow consists of calling the **simon** function which is responsible for coordinating and calling the other sub-functions to complete the challenge. Once the **simon** function has been called, it begins the timer with the **startTimer** function. This timer runs in the background and ensures that the system is only at the Simon challenge for 15 seconds. After the timer has been started, the **watchColors** function is called to watch for the sequence of lights in the Simon pattern. After the last light has been detected, the **spinToColor** function is called, which will spin to the appropriate color seen. After the color has been spun to, the **pressColor** function is called to press the correct color. At this point, **pressColor** returns control to **spinToColor** to spin to the next color in the sequence. If no more colors are in the sequence, control is returned to the **watchColor** function. Once the timer limit has been reached, control is returned to the Coordination subsystem via a call to the **exitArea** function.

### 6.2.2 Etch-A-Sketch Detected

If the Coordination subsystem has detected the Etch-A-Sketch challenge, the Challenge subsystem will initiate the Etch-A-Sketch procedure flow. This flow consists of calling the **etch** function which is responsible for coordinating and calling the other sub-functions to complete the challenge. Once the **etch** function has been called, it lowers the Etch-A-Sketch interactor via the **lowerEtch** function. After the interactor has been lowered, the **runEtch** function is called. The **runEtch** function coordinates the completion of the Etch-A-Sketch challenge by either calling the **twistLeft** function to twist the left knob, or the **twistRight** function to twist the right knob. Once the **runEtch** function has deemed that the challenge is complete the Etch-A-Sketch interactor is raised with **raiseEtch**. After the interactor has been raised, control is returned to the Coordination subsystem via a call to the **exitArea** function.

### 6.2.3 Rubik's Cube Detected

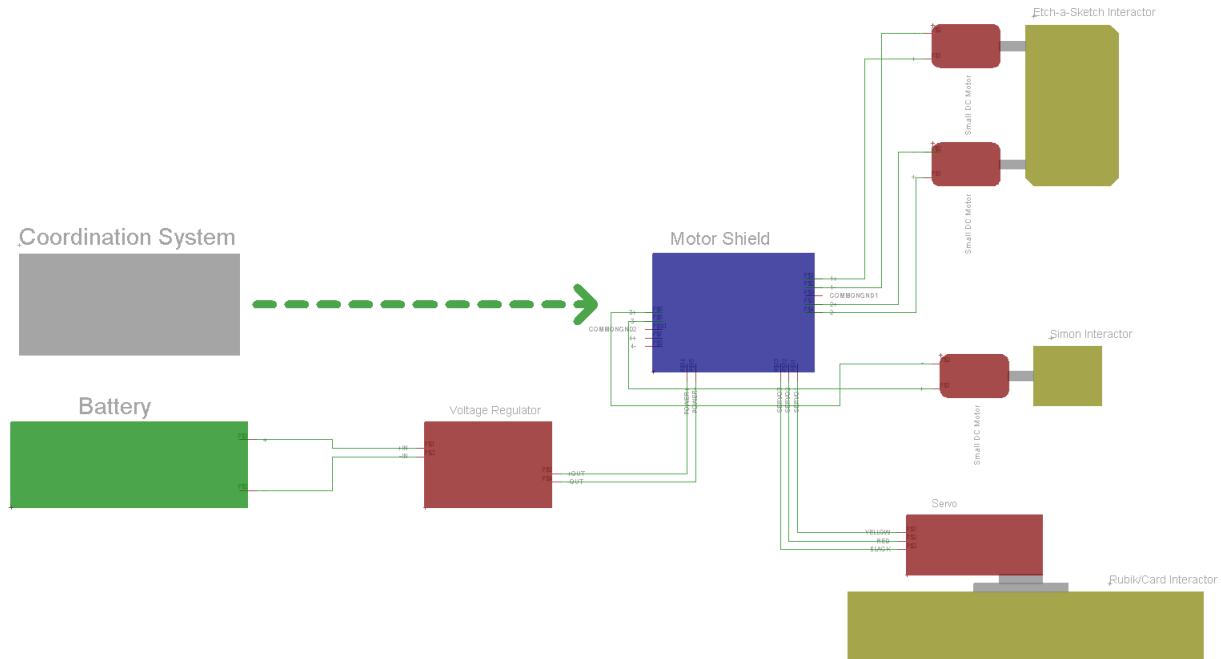
If the Coordination subsystem has detected the Rubik's cube challenge, the Challenge subsystem will initiate the Rubik's cube procedure flow. This flow consists of calling the **rubiks** function which is responsible for coordinating and calling the other sub-functions to complete the challenge. Once the **rubiks** function has been called, it calls the **resetPosition** function to make sure the servo has been reset to its origin. After the position has been reset the **spin180** function is called to spin the Rubik's cube 180 degrees to complete the challenge. Once the **spin180** function has completed, control is returned to the Coordination subsystem via a call to the **exitArea** function.

### 6.2.4 Playing card Detected

If the Coordination subsystem has detected the Rubik's cube challenge, the Challenge subsystem will initiate the Rubik's cube procedure flow. This flow consists of calling the **card** function which is responsible for coordinating and calling the other sub-functions to complete the challenge. Once the **card** function has been called the **lowerCard** function is called to lower the card interactor to pick up a card. Once the card has been picked up, the **raiseCard** function is called to raise the card interactor. After the interactor has been raised, control is returned to the Coordination subsystem via a call to the **exitArea** function.

### 6.3 Hardware Diagrams

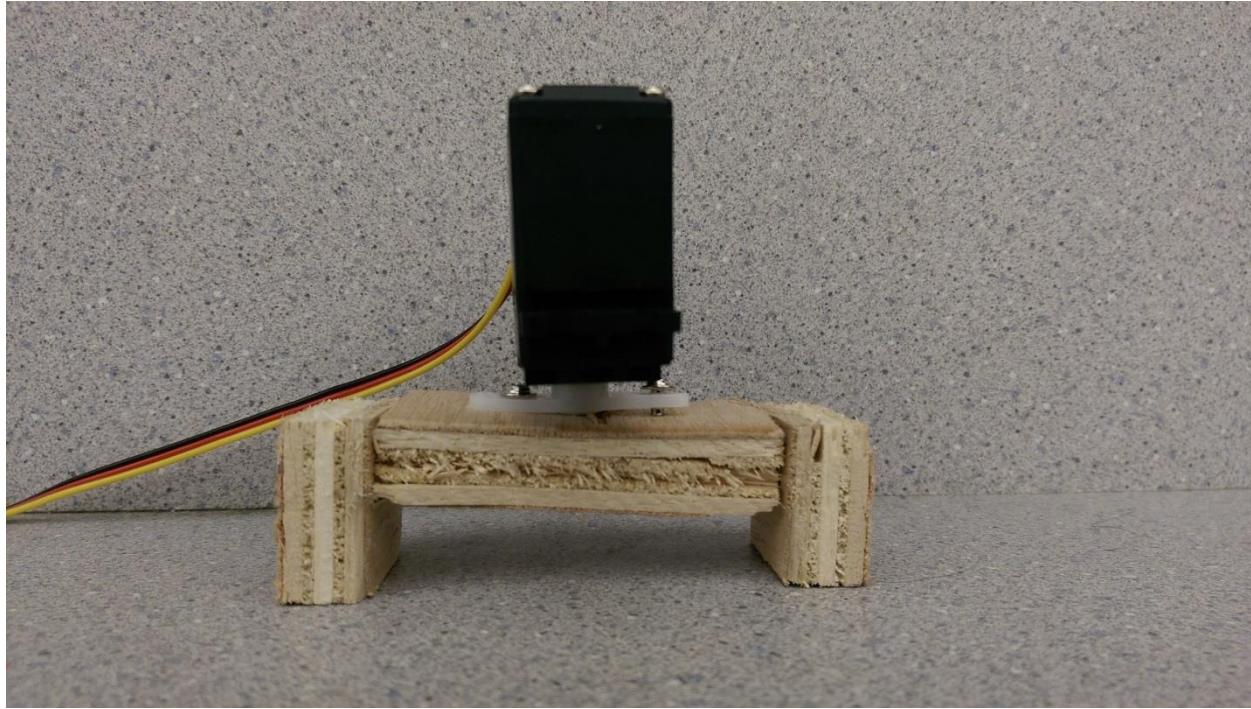
All hardware components and their associated connections for the Challenge subsystem can be seen in **Fig. 18** below.



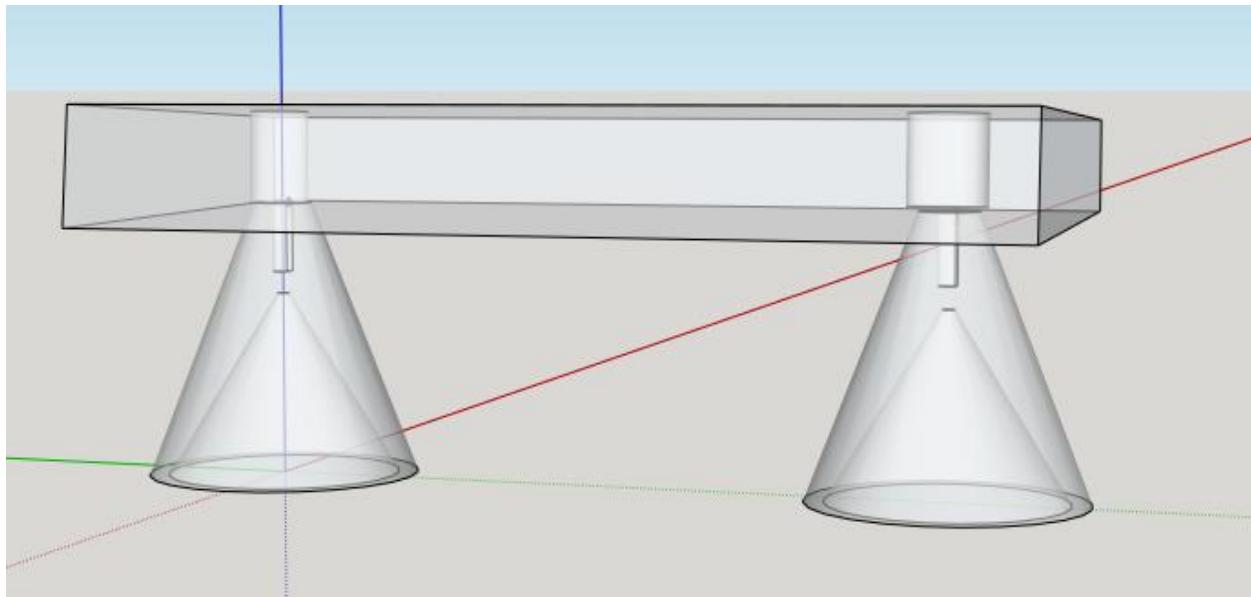
**Fig. 18:** Hardware diagram showing the electrical connections between the components of the challenge completion system.

**Fig. 18** is a diagram of the connections between the parts of the challenge completion system. For this diagram the microcontrollers were omitted to make the connections of the parts specific to the challenge completion system easier to see. In actuality the motor shield is stacked on top of either the Arduino or another motor shield which in turn is stacked on top of the microcontroller using standoff headers. This setup allows the motors connected to the motor shield to receive power from the coordination system while the motor shields allow for multidirectional rotation and precise positioning of the motors.

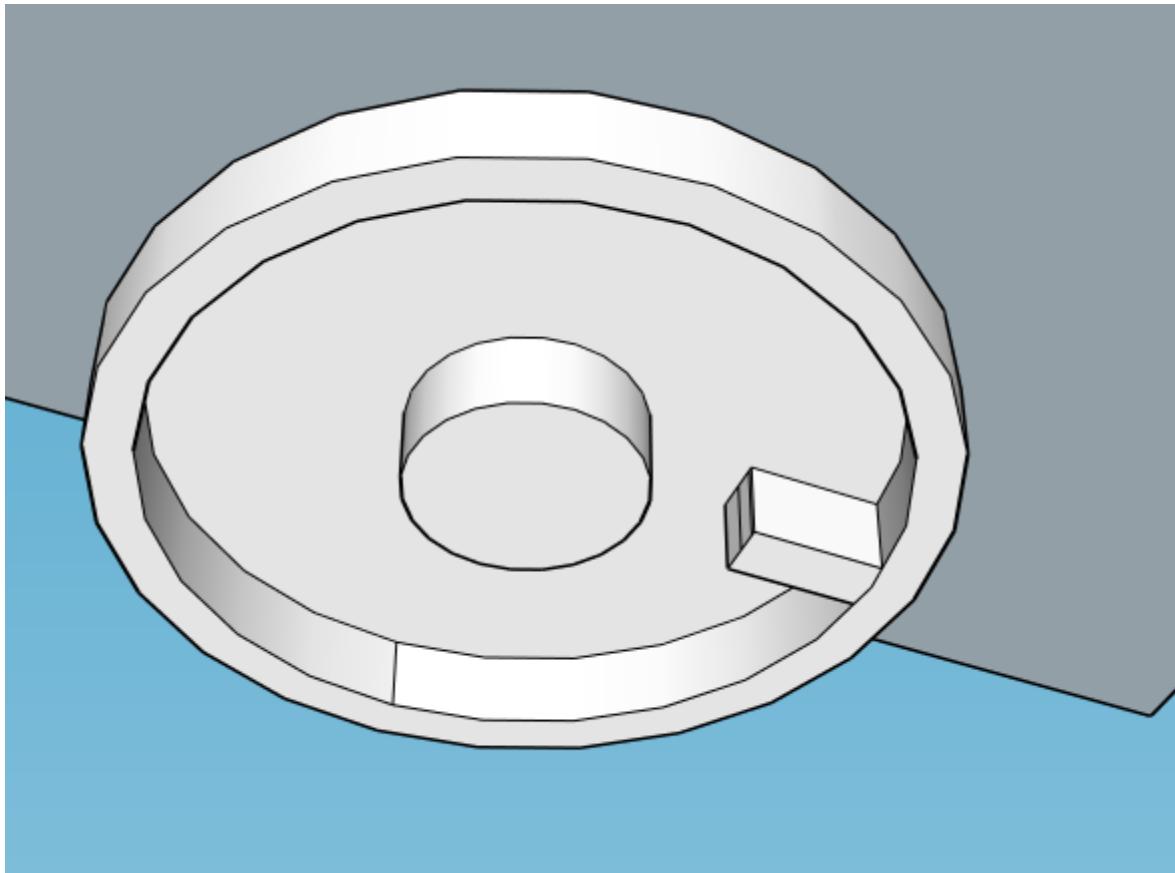
**Fig. 19**, **Fig. 20**, **Fig. 21** and **Fig 22** show the interactors for each of the challenges that Roadie will complete.



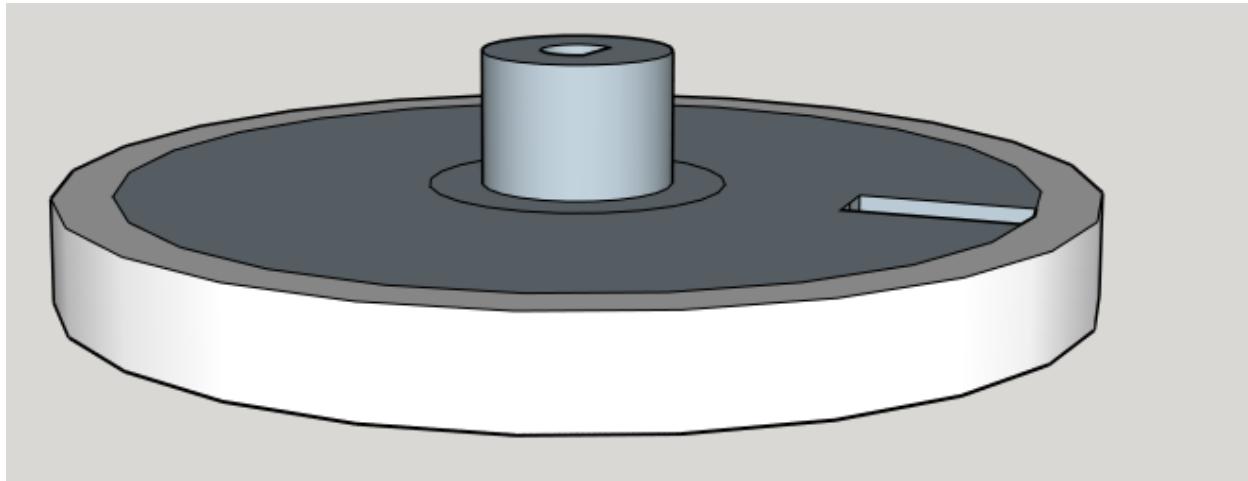
**Fig. 19:** The picture above shows the Etch-a-Sketch interactor.



**Fig. 20:** The picture above is a CAD rendering of the Etch-a-Sketch interactor. Two gear motors will be mounted on top of the conical knobs which will rotate freely.



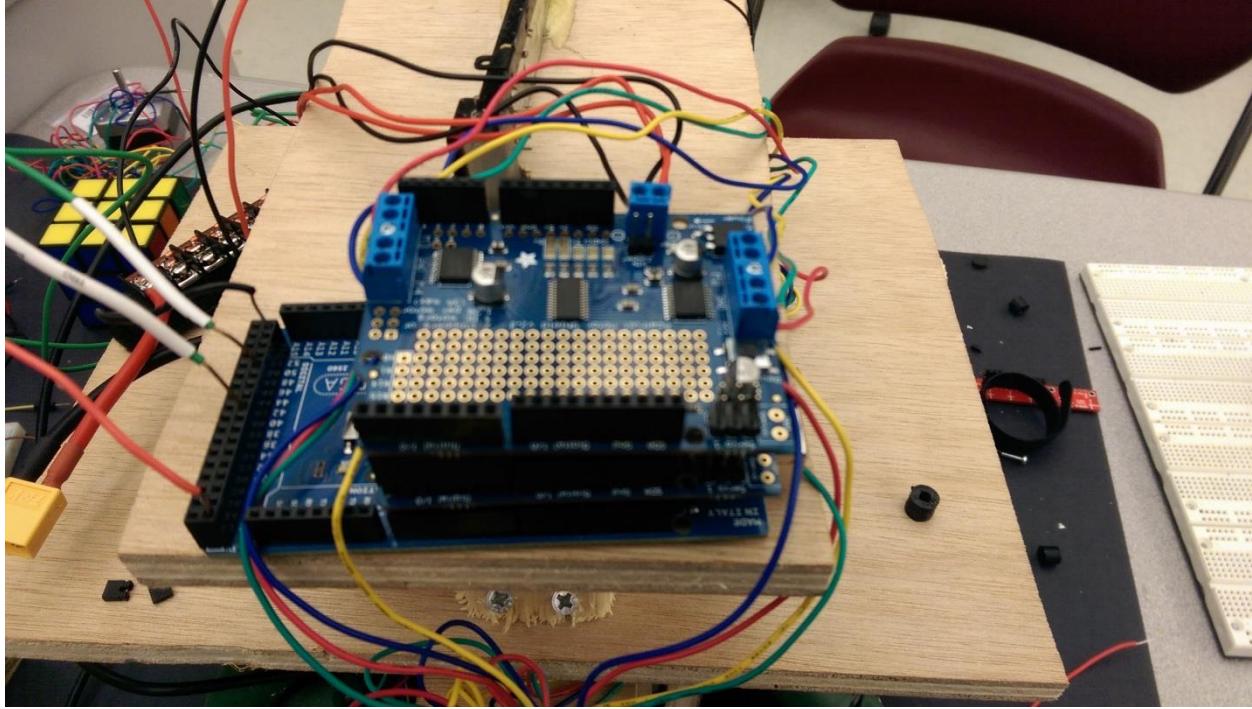
**Fig. 21:** The picture above is a CAD rendering of the Simon interactor. This is the bottom of the part, the small extension will be what pressed the buttons.



**Fig 22:** The picture above is a CAD rendering of the Simon interactor. This is a top view of the part.

## 6.4 Interfaces

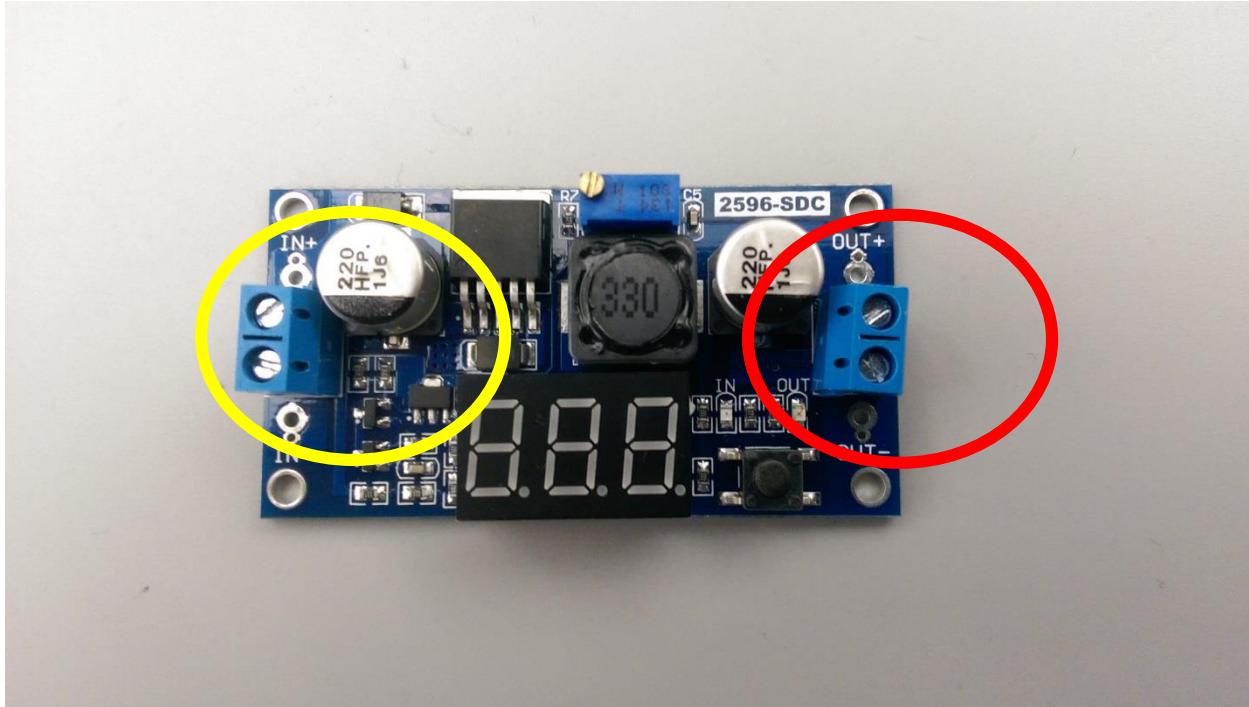
The hardware interfaces for the components used to build the Challenge subsystem are shown in **Fig. 23**, **Fig. 24**, **Fig. 25**, **Fig. 26**, **Error! Reference source not found.** and **Error! Reference source not found.** below.



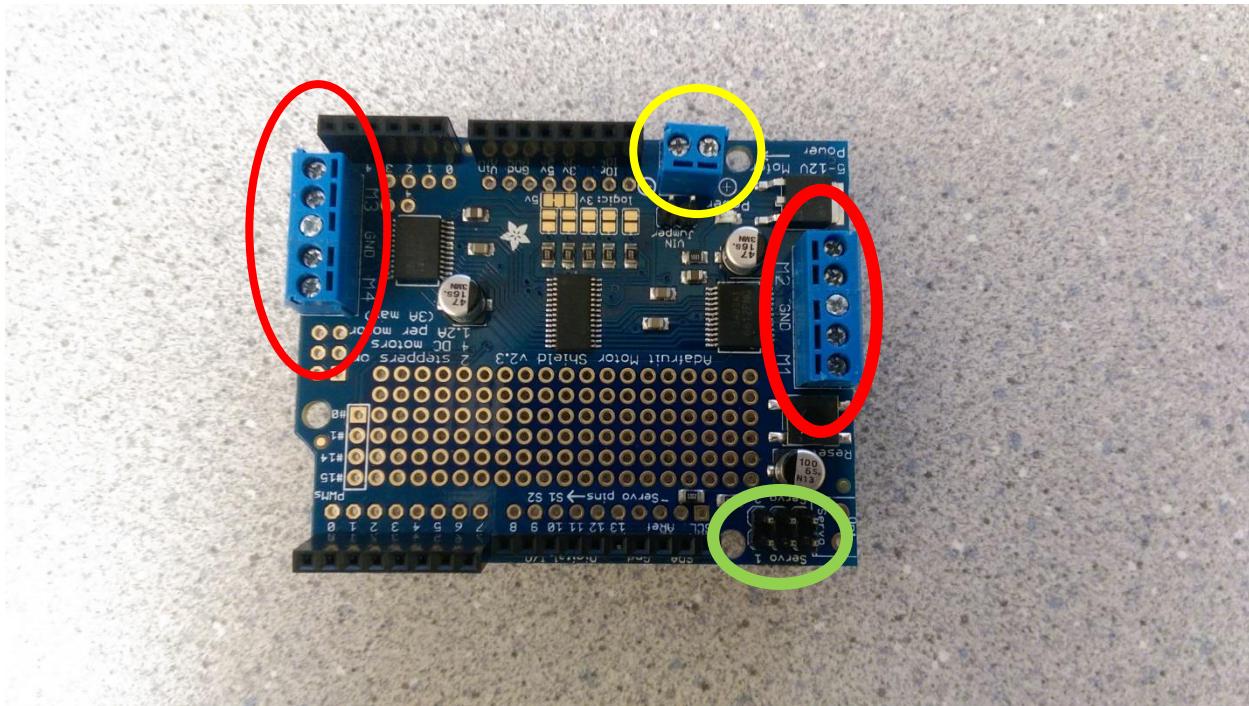
**Fig. 23:** The picture above is the motor shield connected to all devices



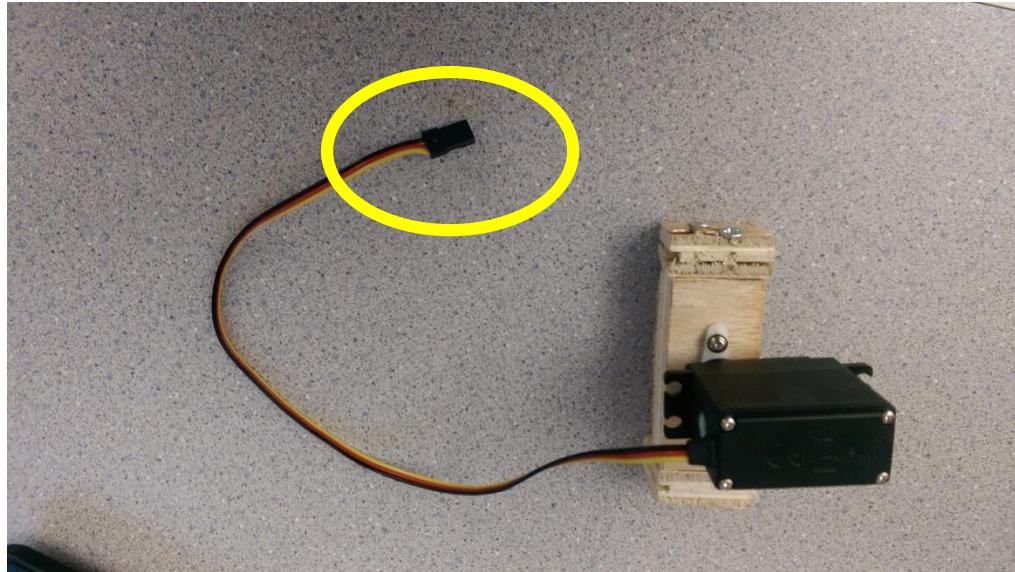
**Fig. 24:** The picture above shows the battery used to power the challenge system components. The yellow circle indicates the connector used to connect the battery to the voltage regulator.



**Fig. 25:** The picture above shows the Voltage Regulator used to control the voltage to the challenge system. The yellow circle indicates the input to the Voltage Regulator, which comes from the battery and the red circle indicates the output which goes to the motor shield.



**Fig. 26:** The picture above shows the Motor Shield. The yellow circle indicates the power input to the motor shield, which comes from the output terminal of the voltage regulator. The red circles denote the output pins which provided signals to the DC motor and the green circle indicates the pins which provide signals to and from the servo.



**Fig. 27:** The picture above shows the Servo. The yellow circle indicates the connector which connects to the motor shield.



**Fig 28:** The picture above is of the small DC motors in the challenge system. The two yellow circles indicate the two input pins which connect to the output pins on the motor shield.

All parts shown in the figures above connect with their individual interfaces to make up the challenge completion system. All motors and servo will have non-electrical attachments which aid in challenge completion. **Fig. 23** is the completed wiring of all components to the motor shield for the challenge system. All other figures are individual components which connect to the others as defined in the figure descriptions.

## 6.5 Parts Budget

This section will cover the parts budget for the challenge subsystem. These components are vital in ensuring the proper and efficient completion of course challenges. The different components used in the subsystem will be outlined, and then a cost matrix will show a cost analysis involving subsystem components and the sources of funding.

### 6.5.1 Components

This section will give a short reasoning for the existence of each component in the subsystem. All of these aid in satisfying Roadie's challenge completion requirements.

### **3D Printed Etch-A-Sketch Knob Handles**

This component is made with 3D printing material. It is a cheap alternative that is easy to prototype and implement. It allowed for early flexibility in design modifications. This component aids in satisfying requirements CHA 14 through CHA 18. [2]

### **Micro Servo**

This component is used to manipulate the top row of the Rubik's Cube. It interacts with a custom built wooden frame to accomplish this task. It is inexpensive and easy to implement. This component aids in satisfying requirements CHA 11 through CHA 13. [2]

### **Micro Gearmotor**

This component is used with the knob handles for the Etch-A-Sketch challenge, and also for the Simon Says challenge. It allows for easy integration with software and continuous 360 degree rotation. This component aids in satisfying requirements CHA 1, CHA 2, CHA 8 through CHA 10, and CHA 14 through CHA 18. [2]

### **Adafruit Motor Shield**

This component is used to control all of the gear motors. With the ability to control up to four DC motors, this shield is a cost-efficient method of controlling all of the motors needed for challenge completion. Since it is used to efficiently control the gear motors, it helps to satisfy the same requirements as the micro gearmotors for the challenge subsystem.

### **Motorized Slide Potentiometer**

This component is used in conjunction with Roadie's camera. It allows the system to extend and retract the camera. This aids in satisfying the size requirement for the system, MOV 1. It also helps the Fosman camera satisfy its requirements in the coordination subsystem.

## **6.5.2 Cost Matrix**

This section provides a table which includes each component and its cost per unit. It shows which components were department funded and which components were funded by the team. The costs of each unit can be seen in **Table 3** below.

Description	Unit Cost	Department Funded Units	Team Funded Units	Total Department Cost	Total Team Cost
3D Printing Material (1 Kilogram)	\$ 30.00	0	1	\$ -	\$ 30.00
Micro Servo (0.18 seconds/60 degrees)	\$ 10.95	0	1	\$ -	\$ 10.95
Micro Gearmotor (90 rpm)	\$ 12.95	3	0	\$ 38.85	\$ -
Adafruit Motor Shield	\$ 27.99	1	0	\$ 27.99	\$ -
Motorized Slide Potentiometer	\$ 19.95	1	1	\$ 19.95	\$ 19.95
Totals:				\$ 86.79	\$ 60.90

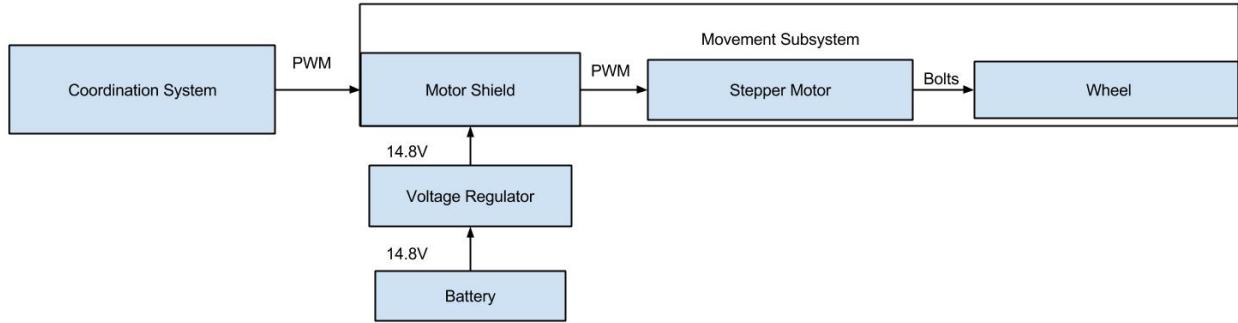
**Table 3:** Cost of hardware for Roadie's Challenge subsystem.

## 7. Decomposition of Movement System

The following section describes the architecture, requirements, use cases, sequence diagrams, and requirements traceability of the Movement subsystem.

### 7.1 Subsystem Architecture

**Fig. 29** shows the decomposition of Roadie's Movement subsystem into major components.



**Fig. 29:** Decomposition of Movement subsystem Roadie.

The Movement subsystem for Roadie consists of the drive motors, wheels and the chassis. This subsystem is responsible for interpreting the movement commands sent to it from the Coordination subsystem. In turn, the Movement subsystem will advance the chassis to the specified location.

#### 7.1.1 Assumptions

It is assumed that the competition area will be as shown in **Fig. 34**. The Movement subsystem has been calibrated to respond to the surfaces laid out in the aforementioned figure. If the surface is different from what is described, there is no guarantee that Roadie will be able to correctly move. It is also assumed that the Movement subsystem will be able to interpret all of the commands originating from the Coordination subsystem. If the Movement subsystem is unable to do so, Roadie will not move as expected.

#### 7.1.2 Dependencies

The Movement subsystem is purely dependent upon the Coordination subsystem for the direction and distance to move Roadie. The Movement subsystem also depends on the environment it is placed in being free of obstacles or other movement inhibitors. Furthermore, the environment which Roadie is placed in must be the same as the competition area as shown in **Fig. 34**. Roadie has been constructed to work with this course format and no other.

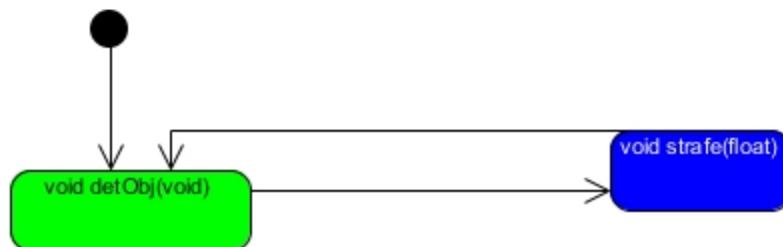
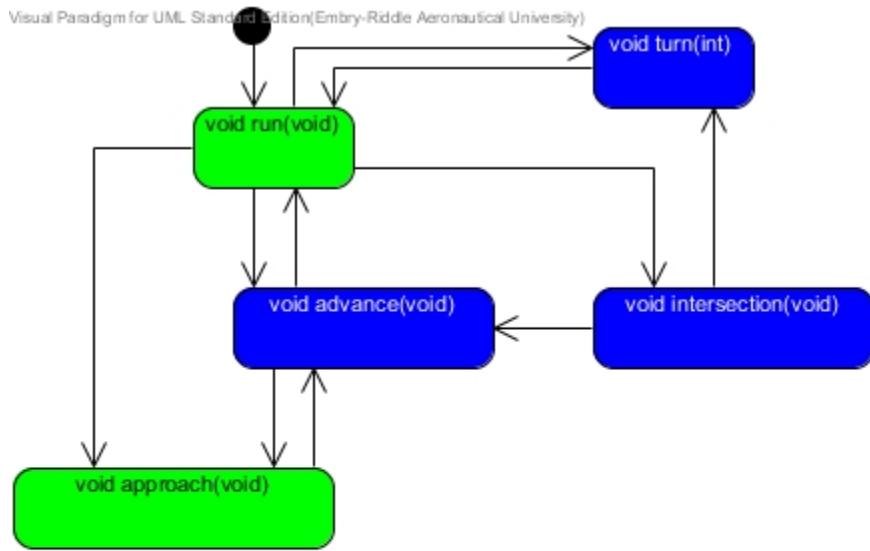
One problem arising from dependency is that the system will fail to move. If the Movement subsystem is unable to get information from the Coordination subsystem, Roadie will be unable to move, representing a total system failure.

### 7.1.3 Constraints

All of the code and logic for the Movement subsystem must fit within the space allotted on the microcontroller. Furthermore, the speed at which the code can run is constrained by the clock speed on the microcontroller. The Movement subsystem may only move as fast as the motors that have been selected to drive Roadie. All components in the Movement subsystem must comply with all rules and regulations set forth in [1].

## 7.2 Functional Diagrams

In **Fig. 30** below, the procedure flows and function calls for the Movement subsystem are laid out.



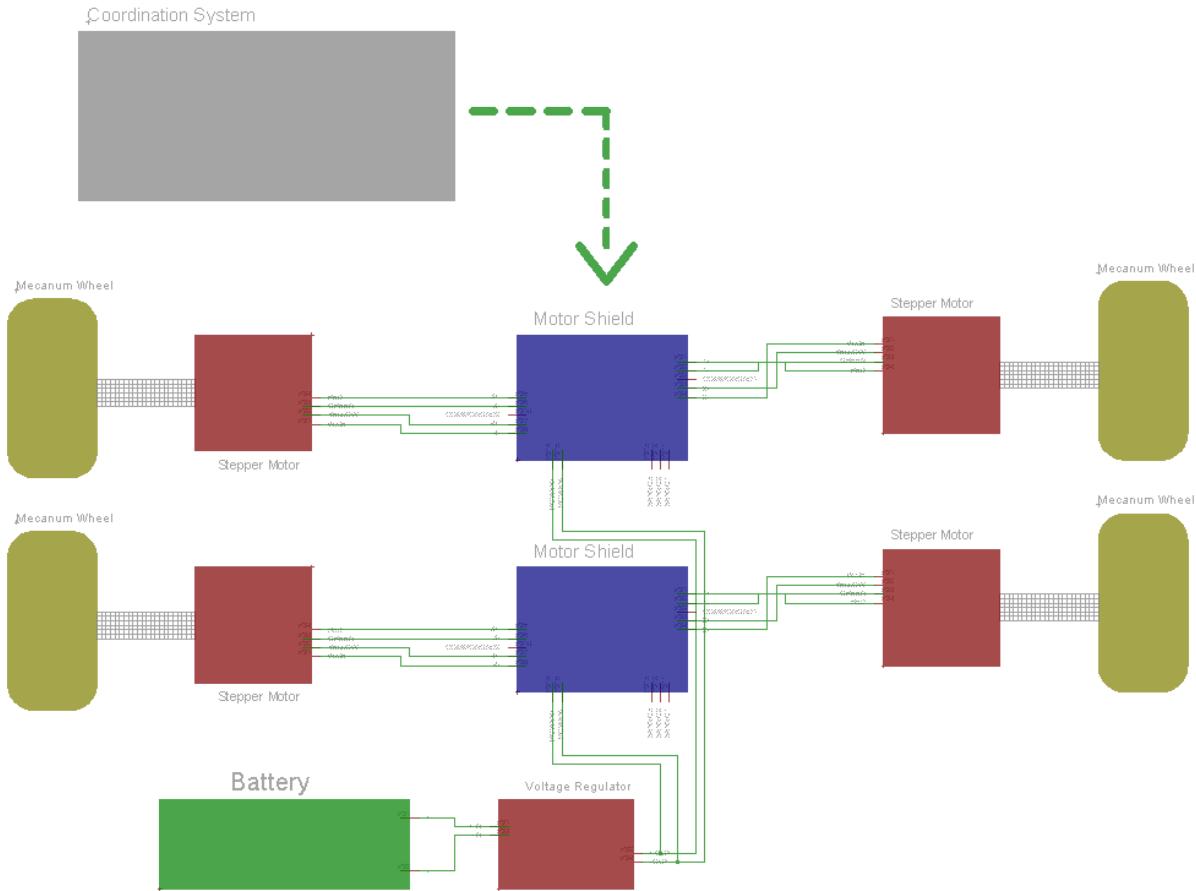
**Fig. 30:** Function calls and procedure flows for Movement subsystem.

The Movement subsystem is instructed to move by the Coordination subsystem. The **run** function, part of the Coordination subsystem, determines whether or not Roadie must progress straight or turn. If the Coordination subsystem determines that it is on the line, it calls the Movement subsystem's **advance** function. The **advance** function simply move Roadie forward. If the Coordination subsystem detects that Roadie has encountered an intersection, it calls upon **intersection** to determine if Roadie should progress forward or perform a turn. The **turn** function executes a left or right turn, depending upon the input passed to it. For example, any negative input would represent a turn to the left, and any positive input would represent a turn to the right.

When alignment with an object is required, the **strafe** function is called. The **strafe** function will move Roadie laterally the distance specified. For example, if a negative value is passed, Roadie will move that increment to the left. A positive value will signal Roadie needs to move to the right.

### 7.3 Hardware Diagrams

All of the hardware components and their associated connections for the Movement subsystem are shown in **Fig. 31** below.



**Fig. 31:** Hardware diagram showing the electrical connections between the components of the Movement subsystem

In **Fig. 31** above, the connections between the components of the movement system are shown. Similar to the diagram referencing the challenge completion system, the motor shields are stacked on top of one another and then stacked on to the microcontroller, thus creating continuity throughout the system. These connections were omitted in **Fig. 31** for easier readability.

In **Fig. 32** below, the coupler that joins the shafts on the stepper motors and the drive shafts for the wheels is depicted.

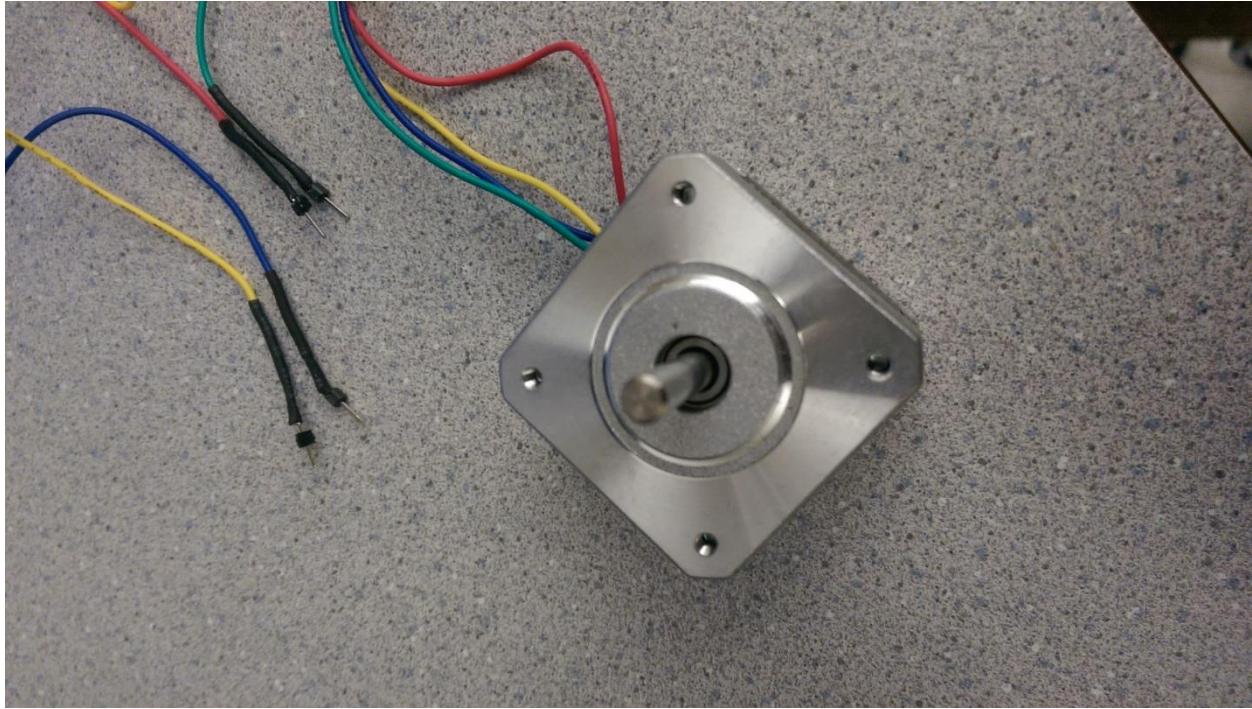


**Fig. 32:** The above picture shows the motor shaft and the wheels connected by a 3D printed coupler

#### 7.4 Interfaces

**Fig. 23** in the previous system section, shows the connections of the components of the movement system to the motor shield as well as the challenge completion system. The motor shield is a vital component of both systems and is consequently present in both system.

**Fig. 33**, below, shows the stepper motors that will be used to drive the system.



**Fig. 33:** The picture above shows the stepper motor. The yellow circle indicates the wire connections which connect to the motor shield shown in **Fig. 26**.

All other interfaces are referenced in **Fig. 24**, **Fig. 25**, and **Fig. 26** in the challenge completion system. The figures reference the battery, voltage regulator and motor shield respectively.

## 7.5 Parts Budget

This section will cover the parts budget for the movement subsystem. These components are vital in facilitating the effective and accurate movement around the course. The different components used in the subsystem will be outlined, and then a cost matrix will show a cost analysis involving subsystem components and the sources of funding.

### 7.5.1 Components

This section will give a short reasoning for the existence of each component in the subsystem. All of these aid in satisfying Roadie's movement requirements.

#### NEMA 16 Stepper Motor

This component is used in combination with the wheels to move the system. The stepper motor was selected in order to provide precise movements when aligning with challenges. This component aids in satisfying requirements MOV 5 through MOV 12. [2]

#### Custom Wooden Chassis

A custom chassis was selected since it is a cheaper alternative than commercial products. It allowed for multiple inexpensive iterations of chassis prototypes and a final unique design. This component is within the constraints of requirement MOV 1. [2]

**Mecanum Wheels**

This brand of wheel was chosen to give the system the freedom of strafing. This is especially beneficial when aligning with challenges. This component satisfies requirement MOV 4, and aids in satisfying requirements MOV 5 through MOV 12. [2]

**Square Aluminum Shaft**

This component is used to connect the wheels to the chassis. It allows for flexibility with a unique chassis design. This requirement is within the constraints of requirement MOV 1, and allows the Mecanum Wheels to satisfy their requirements. [2]

**Venom 5000 mAh 14.8V LiPo**

This component is mainly used to power the Nema 16 Stepper Motors. It was selected for the purpose of a long lasting portable power source. This component satisfies requirement MOV 2. [2]

**DROK DC Adjustable Voltage Regulator**

This component is used in combination with the Venom LiPo power source. It allows for a greater margin of safety and hardware protection for powered components. This component allows the Venom LiPo to satisfy its requirements.

**Adafruit Motor Shield**

This component is used as a method of controlling the Nema 16 Stepper Motors. It allows for the control of multiple motors, along with greater flexibility for any potential additional hardware. Therefore, this component aids in satisfying requirements MOV 5 through MOV 12. [2]

**7.5.2 Cost Matrix**

This section provides a table which includes each component and its cost per unit. It shows which components were department funded and which components were funded by the team. The costs of each unit can be seen in **Table 4** below.

Description	Unit Cost	Department Funded Units	Team Funded Units	Total Department Cost	Total Team Cost
Nema 16 Stepper Motor	\$ 14.95	4	0	\$ 59.80	\$ -
Custom Wooden Chassis	\$ 35.00	0	1	\$ -	\$ 35.00
Mecanum Wheels (Set of 4)	\$ 59.99	1	0	\$ 59.99	\$ -
Square Aluminum Shaft	\$ 2.99	0	1	\$ -	\$ 2.99
Venom 5000mah 14.8V LiPo	\$ 79.43	1	0	\$ 79.43	\$ -
DROK DC Adjustable Voltage Regulator	\$ 8.25	0	0	\$ -	\$ -
Adafruit Motor Shield	\$ 27.99	0	2	\$ -	\$ 55.98
Totals:				\$ 199.22	\$ 93.97

**Table 4:** Cost of hardware for Roadie's Movement subsystem.

## 8. System Test Plan

The test plan outlines the processes in which team AWTY will take in order to test Roadie for the completion of the requirements set by team AWTY to compete in the IEEE 2015 Southeast Competition.

### 8.1 Coordination Test Cases

The following test cases in **Table 5**, **Table 6**, **Table 7**, **Table 8**, **Table 9**, **Table 10**, and **Table 11** test the coordination system of Roadie. **Table 5** tests Roadie's ability to detect and identify an object in front of Roadie when reaching a challenge zone. **Table 6** tests Roadie's ability to identify the Rubik's Cube used in the IEEE competition and Roadie's ability to align to the Rubik's Cube. **Table 7** tests Roadie's ability to identify Simon Carabiner and align to Simon Carabiner. **Table 8** tests Roadie's ability to identify the Etch-A-Sketch and align to the Etch-A-Sketch. **Table 9** tests Roadie's ability to identify the Playing Card Deck and align to the Playing Card Deck. **Table 10** tests the Arduino's ability to handle bad values that might be passed from the Pi. **Table 11** test the Pi's ability to handle bad values that might be passed from the Arduino.

<b>Project Name:</b> Roadie						
<b>Test Case ID:</b> Coordination-1				<b>Test Design By:</b> Michael Philotoff		
<b>Test Priority:</b> High				<b>Test Designed Date:</b> November 30, 2014		
<b>Module Name:</b> Coordination				<b>Test Executed By:</b>		
<b>Test Title:</b> Identify challenge in 10 seconds				<b>Test Execution Date:</b>		
<b>Description:</b> Identify a challenge once arriving to that challenge zone within 10 seconds						
<b>Pre-conditions:</b> Roadie arrives at a challenge zone						
<b>Requirement Satisfied:</b> COR 6 and COR 6.1						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Notes
1	Turn on the ultra-sonic		Ultra-sonic operational			
2	Detect an object within 4 inches of Roadie	Object Detected	Detects an object			
3	Turns the camera on		Camera is operational			
4	Camera identifies the challenge within 5 seconds	Object Identification	Camera identifies the object			
<b>Post-conditions:</b> Identifies the challenge present in front of Roadie						

Table 5: Coordination Test Case 1 Challenge Identification

<b>Project Name:</b> Roadie						
<b>Test Case ID:</b> Coordination-2				<b>Test Design By:</b> Michael Philotoff		
<b>Test Priority:</b> High				<b>Test Designed Date:</b> November 30, 2014		
<b>Module Name:</b> Coordination				<b>Test Executed By:</b>		
<b>Test Title:</b> Rubik's Cube Identification and Alignment				<b>Test Execution Date:</b>		
<b>Description:</b> Roadie correctly identifies the challenge Rubik's Cube within 5 seconds of the camera being turn on. Then Roadie aligns itself up with Rubik's Cube.						
<b>Pre-conditions:</b> Roadie has arrived at a challenge zone and has detected a challenge within 4 inches of Roadie						
<b>Requirement Satisfied:</b> COR 6.3, COR 7, and COR 7.1						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Notes
1	Camera identifies Rubik's Cube		Rubik's Cube is identified			
2	Camera measures the distance between	Pixels converted to inches	Distance measured			
3	Strafe the distance measured by the Camera		Roadie strafes			
4	Roadie moves forward 4 inches		Roadie moves forward			
5	Roadie signals that it has arrived at the challenge		Roadie is at the Rubik's Cube			
<b>Post-conditions:</b> Roadie has aligned with the challenge and signals that it has arrived.						

**Table 6:** Coordination Test Case 2 Rubik's Cube Identification and Alignment

<b>Project Name:</b> Roadie						
<b>Test Case ID:</b> Coordination-3				<b>Test Design By:</b> Michael Philottoff		
<b>Test Priority:</b> High				<b>Test Designed Date:</b> November 30, 2014		
<b>Module Name:</b> Coordination				<b>Test Executed By:</b>		
<b>Test Title:</b> Simon Carabiner Identification and Alignment				<b>Test Execution Date:</b>		
<b>Description:</b> Roadie correctly identifies the challenge Simon Carabiner within 5 seconds of the camera being turn on. Then Roadie aligns itself up with Simon Carabiner.						
<b>Pre-conditions:</b> Roadie has arrived at a challenge zone and has detected a challenge within 4 inches of Roadie						
<b>Requirement Satisfied:</b> COR 6.2, COR 7, and COR 7.1						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Notes
1	Camera identifies Simon Carabiner		Simon Carabiner is identified			
2	Camera measures the distance between Simon Carabiner and Roadie	Pixels converted to inches	Distance measured			
3	Strafe this distance measured from the Camera		Roadie strafes			
4	Roadie moves forwards 4 inches		Roadie moves forward			
5	Roadie signals that it has arrived at the challenge		Roadie is at the Simon Carabiner			
<b>Post-conditions:</b> Roadie has aligned with the challenge and signals that it has arrived at the challenge.						

**Table 7:** Coordination Test Case 3 Simon Carabiner Identification and Alignment

<b>Project Name:</b> Roadie						
<b>Test Case ID:</b> Coordination-4				<b>Test Design By:</b> Michael Philottoff		
<b>Test Priority:</b> High				<b>Test Designed Date:</b> November 30, 2014		
<b>Module Name:</b> Coordination				<b>Test Executed By:</b>		
<b>Test Title:</b> Etch-A-Sketch Identification and Alignment				<b>Test Execution Date:</b>		
<b>Description:</b> Roadie correctly identifies the challenge Etch-A-Sketch within 5 seconds of the camera being turn on. Then Roadie aligns itself up with Etch-A-Sketch.						
<b>Pre-conditions:</b> Roadie has arrived at a challenge zone and has detected a challenge within 4 inches of Roadie						
<b>Requirement Satisfied:</b> COR 6.4, COR 7, and COR 7.1						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Notes
1	Camera identifies Etch-A-Sketch		Etch-A-Sketch is identified			
2	Camera measures the distance	Pixels converted to inches	Distance measured			
3	Strafe the distance measured from the Camera		Roadie strafes			
4	Roadie moves forward 4 inches		Roadie moves forward			
5	Roadie signals that it has arrived at the challenge		Roadie is at the Etch-A-Sketch			
<b>Post-conditions:</b> Roadie has aligned with the challenge and signals that it has arrived at the challenge.						

Table 8: Coordination Test Case 4 Etch-A-Sketch Identification and Alignment

<b>Project Name:</b> Roadie						
<b>Test Case ID:</b> Coordination-5						
<b>Test Priority:</b> High						
<b>Module Name:</b> Coordination						
<b>Test Title:</b> Playing Card Deck Identification and Alignment						
<b>Description:</b> Roadie correctly identifies the challenge Playing Card deck within 5 seconds of the camera being turn on. Then Roadie aligns itself up with Playing Card deck.						
<b>Pre-conditions:</b> Roadie has arrived at a challenge zone and has detected a challenge within 4 inches of Roadie						
<b>Requirement Satisfied:</b> COR 6.5, COR 7, and COR 7.1						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Notes
1	Camera identifies Playing Card deck		Playing Card deck is identified			
2	Camera measures the distance	Pixels converted to inches	Distance measured			
3	Strafe the distance measured from the Camera		Roadie strafes			
4	Roadie moves forward 4 inches		Roadie moves forward			
5	Roadie signals that it has arrived at the challenge		Roadie is at the Playing Card deck			
<b>Post-conditions:</b> Roadie has aligned with the challenge and signals that it has arrived at the challenge.						

Table 9: Coordination Test Case 5 Playing Card Deck Identification and Alignment

<b>Project Name:</b> Roadie						
<b>Test Case ID:</b> Coordination				<b>Test Design By:</b> Michael Philotoff		
<b>Test Priority:</b> Medium				<b>Test Designed Date:</b> December 4, 2014		
<b>Module Name:</b> Coordination				<b>Test Executed By:</b>		
<b>Test Title:</b> Pi passes bad value to Arduino				<b>Test Execution Date:</b>		
<b>Description:</b> The pi passes a bad value to the Arduino when the Arduino ask for a value.						
<b>Pre-conditions:</b> Roadie is in the approach state in a challenge zone						
<b>Requirement Satisfied:</b>						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Notes
1	Roadie turns on the Camera		Camera is powered on			
2	Arduino waits for the Pi to pass a value		Arduino waits for a value			
3	The Pi passes the Arduino a bad value	Bad value	Pi sends the Arduino a bad values			
4	Arduino identifies the bad value and ask the Pi for another value		Arduino does not crash and ask the Pi for another value			
<b>Post-conditions:</b> Arduino does not crash from the bad value and ask the Pi for another value						

**Table 10:** Coordination Test Case 6 Pi passes bad value to Arduino

<b>Project Name:</b> Roadie						
<b>Test Case ID:</b> Coordination				<b>Test Design By:</b> Michael Philotoff		
<b>Test Priority:</b> Medium				<b>Test Designed Date:</b> December 4, 2014		
<b>Module Name:</b> Coordination				<b>Test Executed By:</b>		
<b>Test Title:</b> Arduino passes bad value to the Pi				<b>Test Execution Date:</b>		
<b>Description:</b> The Arduino passes a bad value to the Pi while the Pi is waiting for a value						
<b>Pre-conditions:</b>						
<b>Requirement Satisfied:</b>						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Notes
1	Pi waits for a value from the Arduino		The Pi waits for a value			
2	The Arduino passes the Pi a bad value	Bad value	The Arduino sends the Pi a bad values			
3	The Pi identifies the bad value and ask the Arduino for another value		The Pi does not crash and ask the Arduino for another value			
<b>Post-conditions:</b> The Pi does not crash from the bad value and ask the Arduino for another value						

Table 11: Coordination Test Case 7 Arduino passes bad value to the Pi

## 8.2 Challenge Test Cases

The following test cases in

**Table 12, Table 13, Table 14, Table 15, Table 16, and Table 17** test Roadie's Challenge subsystem.

**Table 12** tests Roadie's ability to play the Simon Carabiner game for 15 seconds without pressing the wrong buttons. **Table 13** tests Roadie's ability to react if Roadie presses the wrong button when playing Simon Carabiner game. **Table 14** test Roadie's ability to complete the Playing Card challenge by picking up a single playing card. **Table 15** test Roadie's ability to carry the Playing Card to the finish line after completing the Playing Card challenge. **Table 16** test Roadie's ability to twist one row of a Rubik's Cube 180 degrees. **Table 17** test Roadie's ability to draw "IEEE" using an Etch-A-Sketch.

<b>Project Name:</b> Roadie						
<b>Test Case ID:</b> Challenge-1				<b>Test Design By:</b> Michael Philotoff		
<b>Test Priority:</b> High				<b>Test Designed Date:</b> December 3, 2014		
<b>Module Name:</b> Challenge				<b>Test Executed By:</b>		
<b>Test Title:</b> Simon Carabiner Challenge				<b>Test Execution Date:</b>		
<b>Description:</b> Roadie plays Simon Carabiner correctly for 15 seconds						
<b>Pre-conditions:</b> Roadie has identified and aligned itself up with the Simon Carabiner						
<b>Requirement Satisfied:</b> CHA 1-10						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Notes
1	Roadie Presses the start button		Roadie powers on Simon			
2	Roadie reads in the light values from Simon to determine which buttons to press	Array of light values	Roadie reads the light values from Simon			
3	Roadie presses the color buttons in the order they lite up		Roadie correctly press the color buttons in order			
4	Repeat steps 2 and 3 for 15 seconds	Time	Roadie plays Simon for 15 seconds			
5	Roadie starts the process to leave the challenge zone					
<b>Post-conditions:</b> Roadie has correctly played Simon Carabiner for 15 seconds						

Table 12: Challenge Test Case 1 Simon Carabiner Challenge

<b>Project Name:</b> Roadie					
<b>Test Case ID:</b> Challenge-2				<b>Test Design By:</b> Michael Philotoff	
<b>Test Priority:</b> Low				<b>Test Designed Date:</b> December 3, 2014	
<b>Module Name:</b> Challenge				<b>Test Executed By:</b>	
<b>Test Title:</b> Fails Simon Carabiner Challenge				<b>Test Execution Date:</b>	
<b>Description:</b> Roadie pushes the wrong color button when					
<b>Pre-conditions:</b> Roadie has identified and aligned itself up with the Simon Carabiner					
<b>Requirement Satisfied:</b>					
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Roadie Presses the start button		Roadie powers on Simon		
2	Roadie reads in the light values from Simon to determine which buttons to press	Array of light values	Roadie reads the light values from Simon		
3	Roadie presses the wrong color button		Roadie correctly press the color buttons in order		
4	Roadie reads the failure signal from Simon Carabiner	Failure signal	Roadie stops playing Simon Carabiner		
5	Roadie starts the process to leave the challenge zone		Roadie starts the process to leave the challenge zone		
<b>Post-conditions:</b> Roadie has fails to press the correct button when choose which color button to press and starts the process to leave the challenge zone					

**Table 13:** Challenge Test Case 2 Fails Simon Carabiner Challenge

<b>Project Name:</b> Roadie						
<b>Test Case ID:</b> Challenge-3				<b>Test Design By:</b> Michael Philotoff		
<b>Test Priority:</b> High				<b>Test Designed Date:</b> December 3, 2014		
<b>Module Name:</b> Challenge				<b>Test Executed By:</b>		
<b>Test Title:</b> Card Challenge				<b>Test Execution Date:</b>		
<b>Description:</b> Roadie picks up a single playing card						
<b>Pre-conditions:</b> Roadie has identified and aligned itself up with the deck of playing cards						
<b>Requirement Satisfied:</b> CHA 19 and CHA 21						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Notes
1	Roadie presses down on the deck of playing cards		Roadie applies pressure to the deck of cards			
2	Roadie picks up a single playing card		Roadie picks up a single playing card			
3	Roadie starts the process to leave the challenge zone		Roadie starts leaving the challenge zone			
<b>Post-conditions:</b> Roadie has a pick up a playing card						

Table 14: Challenge Test Case 3 Card Challenge

<b>Project Name:</b> Roadie						
<b>Test Case ID:</b> Challenge-4				<b>Test Design By:</b> Michael Philotoff		
<b>Test Priority:</b> High				<b>Test Designed Date:</b> December 3, 2014		
<b>Module Name:</b> Challenge				<b>Test Executed By:</b>		
<b>Test Title:</b> Card Carrying				<b>Test Execution Date:</b>		
<b>Description:</b> Roadie carries a single playing card to the finish line						
<b>Pre-conditions:</b> Roadie has completed the card challenge						
<b>Requirement Satisfied:</b> CHA 20 and CHA 21						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Notes
1	Roadie holds on to the playing card that was pick up from the Card Challenge		Roadie is holding onto the playing card			
2	Roadie executes the leaving procedure		Roadie leaves the challenge zone			
3	Roadie enters line following state after executing the leave procedure		Roadie starts line following			
4	Roadie crosses the finish line		Roadie reaches the finish line			
<b>Post-conditions:</b> Roadie carried a single playing card to the finish line						

**Table 15:** Challenge Test Case 4 Card Carrying

<b>Project Name:</b> Roadie						
<b>Test Case ID:</b> Challenge-5				<b>Test Design By:</b> Michael Philotoff		
<b>Test Priority:</b> High				<b>Test Designed Date:</b> December 3, 2014		
<b>Module Name:</b> Challenge				<b>Test Executed By:</b>		
<b>Test Title:</b> Rubik's Cube Challenge				<b>Test Execution Date:</b>		
<b>Description:</b> Roadie twist a row of the Rubik's Cube 180 degrees						
<b>Pre-conditions:</b> Roadie has identified and aligned up with the Rubik's Cube						
<b>Requirement Satisfied:</b> CHA 11-13						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Notes
1	Roadie holds on to the Rubik's Cube		Roadie has ahold of the Rubik's Cube			
2	Roadie places turning device for the Rubik's Cube over the Rubik's Cube		Roadie placed the turning device onto the Rubik's Cube			
3	Roadie turns the row of the Rubik's Cube 180 degrees		Roadie turned a row of the Rubik's Cube 180 degrees			
4	Roadie removes the turning device from the Rubik's Cube		Roadie has removed the turning device			
5	Roadie starts the process to leave the challenge zone		Roadie starts to leave the challenge zone			
<b>Post-conditions:</b> Roadie twisted a row of the Rubik's Cube 180 degrees						

**Table 16:** Challenge Test Case 5 Rubik's Cube Challenge

<b>Project Name:</b> Roadie						
<b>Test Case ID:</b> Challenge-6				<b>Test Design By:</b> Michael Philotoff		
<b>Test Priority:</b> High				<b>Test Designed Date:</b> December 3, 2014		
<b>Module Name:</b> Challenge				<b>Test Executed By:</b>		
<b>Test Title:</b> Etch-A-Sketch Challenge				<b>Test Execution Date:</b>		
<b>Description:</b> Roadie uses the Etch-A-Sketch to draw "IEEE" onto the Etch-A-Sketch						
<b>Pre-conditions:</b> Roadie has identified and aligned up with the Etch-A-Sketch						
<b>Requirement Satisfied:</b> CHA 14-18						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Notes
1	Roadie places the two caps over the left and right nobs of the Etch-A-Sketch		Roadie has placed the caps over the Etch-A-Sketch			
2	Roadie twists the Etch-A-Sketch nobs to draw "IEEE" onto the Etch-A-Sketch		Roadie drew "IEEE" on the Etch-A-Sketch			
3	Roadie removes the two caps over the two Etch-A-Sketch nobs		Roadie removed the two caps over the Etch-A-Sketch			
4	Roadie starts the process to leave the challenge zone		Roadie starts to leave the challenge zone			
<b>Post-conditions:</b> Roadie draws "IEEE" on the Etch-A-Sketch and starts to leave the challenge zone						

Table 17: Challenge Test Case 6 Etch-A-Sketch Challenge

### 8.3 Movement and Misc. Test Cases

The following test cases in **Table 18**, **Table 19**, **Table 20**, **Table 21**, **Table 22**, **Table 23**, **Table 24**, **Table 25**, and **Table 26** test Roadie's movement subsystem. Additionally, some miscellaneous test cases have been added to this section. **Table 18** tests Roadie's size to make sure Roadie does not violate the maximum system dimensions. **Table 19** tests all of Roadie's subsystems to insure Roadie can handle 3 different rounds at the completion. **Table 20** tests Roadie's ability to turn on and off by using a switch. **Table 21** tests Roadie's ability to not leave the playing field while completing the course. **Table 22** tests Roadie's ability to follow the guidance tape. **Table 23** tests Roadie's ability to make a right turn when needed. **Table 24** tests Roadie's ability to make a left turn when needed. **Table 25** tests Roadie's ability to make correction if Roadie's veers off to the right of guidance tape. **Table 26** tests Roadie's ability to make correction if Roadie's veers off to the left of guidance tape.

<b>Project Name:</b> Roadie						
<b>Test Case ID:</b> Movement-1				<b>Test Design By:</b> Alex Senopoulos		
<b>Test Priority:</b> Low				<b>Test Designed Date:</b> November 29, 2014		
<b>Module Name:</b> Movement				<b>Test Executed By:</b>		
<b>Test Title:</b> Sizing				<b>Test Execution Date:</b>		
<b>Description:</b> Confirm size requirements for Roadie						
<b>Pre-conditions:</b> Roadie will be motionless with its camera retracted.						
<b>Requirement Satisfied:</b> MOV 1						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Notes
1	Leave Roadie motionless					
2	Measure dimensions	Width/Length/Height	Within 1ft.x1ft.x1ft.			
<b>Post-conditions:</b> Roadie will remain motionless						

Table 18: Movement Test Case 1 Sizing

<b>Project Name:</b> Roadie						
<b>Test Case ID:</b> Movement-2				<b>Test Design By:</b> Alex Senopoulos		
<b>Test Priority:</b> High				<b>Test Designed Date:</b> November 29, 2014		
<b>Module Name:</b> Movement				<b>Test Executed By:</b>		
<b>Test Title:</b> Roadie stress test				<b>Test Execution Date:</b>		
<b>Description:</b> Test life of Roadie's power to ensure at least 30 minute lifespan						
<b>Pre-conditions:</b> Roadie will start suspended with no wheel contact on ground and a full battery charge. Modified code uploaded to keep motors and sensors continually running.						
<b>Requirement Satisfied:</b> MOV 2						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Notes
1	Suspend Roadie in place					
2	Run all subsystems at normal operating capacity					
3	Measure time until noticeable deterioration of wheel speed	Time (minutes)	Greater than 30 minutes			
<b>Post-conditions:</b> Roadie power source will have a less than nominal charge.						

Table 19: Movement Test Case 2 Roadie Stress Test

<b>Project Name:</b> Roadie						
<b>Test Case ID:</b> Movement-3				<b>Test Design By:</b> Alex Senopoulos		
<b>Test Priority:</b> Low				<b>Test Designed Date:</b> November 29, 2014		
<b>Module Name:</b> Movement				<b>Test Executed By:</b>		
<b>Test Title:</b> Power switch				<b>Test Execution Date:</b>		
<b>Description:</b> Ensure Roadie has easily accessible power switch						
<b>Pre-conditions:</b> Roadie will be powered off at the starting area.						
<b>Requirement Satisfied:</b> MOV 3						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Notes
1	Use power switch to provide power to Roadie		Roadie will receive power			
<b>Post-conditions:</b> Roadie will have power and execute loaded Arduino instructions.						

Table 20: Movement Test Case 3 Power Switch

<b>Project Name:</b> Roadie						
<b>Test Case ID:</b> Movement-4				<b>Test Design By:</b> Alex Senopoulos		
<b>Test Priority:</b> Medium				<b>Test Designed Date:</b> November 29, 2014		
<b>Module Name:</b> Movement				<b>Test Executed By:</b>		
<b>Test Title:</b> Surface Contact				<b>Test Execution Date:</b>		
<b>Description:</b> Ensure Roadie remains in contact with competition surface						
<b>Pre-conditions:</b> Roadie will be un-powered at starting area						
<b>Requirement Satisfied:</b> MOV 4, MOV 5						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Notes
1	Turn on Roadie					
2	Observe Roadie's execution of the competition from start to finish.		Roadie remains in contact with competition surface.			
<b>Post-conditions:</b> Roadie is motionless at the finish line.						

Table 21: Movement Test Case 4 Surface Contact

<b>Project Name:</b> Roadie						
<b>Test Case ID:</b> Movement-5				<b>Test Design By:</b> Alex Senopoulos		
<b>Test Priority:</b> High				<b>Test Designed Date:</b> November 29, 2014		
<b>Module Name:</b> Movement				<b>Test Executed By:</b>		
<b>Test Title:</b> Minimum oscillation line following				<b>Test Execution Date:</b>		
<b>Description:</b> Ensure Roadie can efficiently follow straight guidance tape.						
<b>Pre-conditions:</b> Roadie will be in a line following state with its sensor array over straight guidance tape.						
<b>Requirement Satisfied:</b>						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Notes
1	Allow Roadie to run continuously along guidance tape.		Roadie will keep guidance tape within the bounds of its own wheels.			
<b>Post-conditions:</b> Roadie will arrive at the end of the guidance tape and have the tape between the bounds of the Roadie's wheels.						

Table 22: Movement Test Case 5 Minimum Oscillation Line Following

<b>Project Name:</b> Roadie						
<b>Test Case ID:</b> Movement-6						
<b>Test Priority:</b> High						
<b>Module Name:</b> Movement						
<b>Test Title:</b> Right turn						
<b>Description:</b> Ensure Roadie can navigate a right turn correctly						
<b>Pre-conditions:</b> Roadie will approach a right turn in the guidance tape while in the line following state.						
<b>Requirement Satisfied:</b>						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Notes
1	Roadie approaches right turn with sensors over guidance tape					
2	Roadie rotates leaves line following state and turns right 90 degrees	Angle of Roadie relative to new section of guidance tape	10 degree margin of error			
3	Roadie resumes line following state					
<b>Post-conditions:</b> Roadie will be in a line following state navigating along the guidance tape.						

Table 23: Movement Test Case 6 Right Turn

<b>Project Name:</b> Roadie					
<b>Test Case ID:</b> Movement-7				<b>Test Design By:</b> Alex Senopoulos	
<b>Test Priority:</b> High				<b>Test Designed Date:</b> November 29, 2014	
<b>Module Name:</b> Movement				<b>Test Executed By:</b>	
<b>Test Title:</b> Left turn				<b>Test Execution Date:</b>	
<b>Description:</b> Ensure Roadie can navigate a left turn correctly					
<b>Pre-conditions:</b> Roadie will approach a left turn in the guidance tape while in the line following state.					
<b>Requirement Satisfied:</b>					
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Roadie approaches left turn with sensors over guidance tape				
2	Roadie rotates leaves line following state and turns 90 degrees left	Angle of Roadie relative to new section of guidance tape	10 degree margin of error		
3	Roadie resumes line following state				
<b>Post-conditions:</b> Roadie will be in a line following state navigating along the guidance tape.					

Table 24: Movement Test Case 7 Left Turn

<b>Project Name:</b> Roadie					
<b>Test Case ID:</b> Movement-8				<b>Test Design By:</b> Alex Senopoulos	
<b>Test Priority:</b> Medium				<b>Test Designed Date:</b> November 29, 2014	
<b>Module Name:</b> Movement				<b>Test Executed By:</b>	
<b>Test Title:</b> Roadie veer too far right				<b>Test Execution Date:</b>	
<b>Description:</b> Roadie corrects movement for being off the line too far to the right					
<b>Pre-conditions:</b> Roadie is in the line following state with the two left most sensors over the guidance tape and the sensor array front facing away from the guidance tape.					
<b>Requirement Satisfied:</b>					
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Roadie adjusts right side motor speed to a larger value than the left side	Motor rpm values for each wheel	Right side motors = 5000, Left side motors = 2000		
<b>Post-conditions:</b> Roadie's sensor array front will be facing towards the line					

Table 25: Movement Test Case 8 Roadie Veer too Right

<b>Project Name:</b> Roadie						
<b>Test Case ID:</b> Movement-9				<b>Test Design By:</b> Michael Philotoff		
<b>Test Priority:</b> Medium				<b>Test Designed Date:</b> December 4, 2014		
<b>Module Name:</b> Movement				<b>Test Executed By:</b>		
<b>Test Title:</b> Roadie veer too far left				<b>Test Execution Date:</b>		
<b>Description:</b> Roadie corrects movement for being off the line too far to the left						
<b>Pre-conditions:</b> Roadie is in the line following state with the two right most sensors over the guidance tape and the sensor array front facing away from the guidance tape.						
<b>Requirement Satisfied:</b>						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Notes
1	Roadie adjusts left side motor speed to a larger value than the right side	Motor rpm values for each wheel	Left side motors = 5000, Right side motors = 2000			
<b>Post-conditions:</b> Roadie's sensor array front will be facing towards the line						

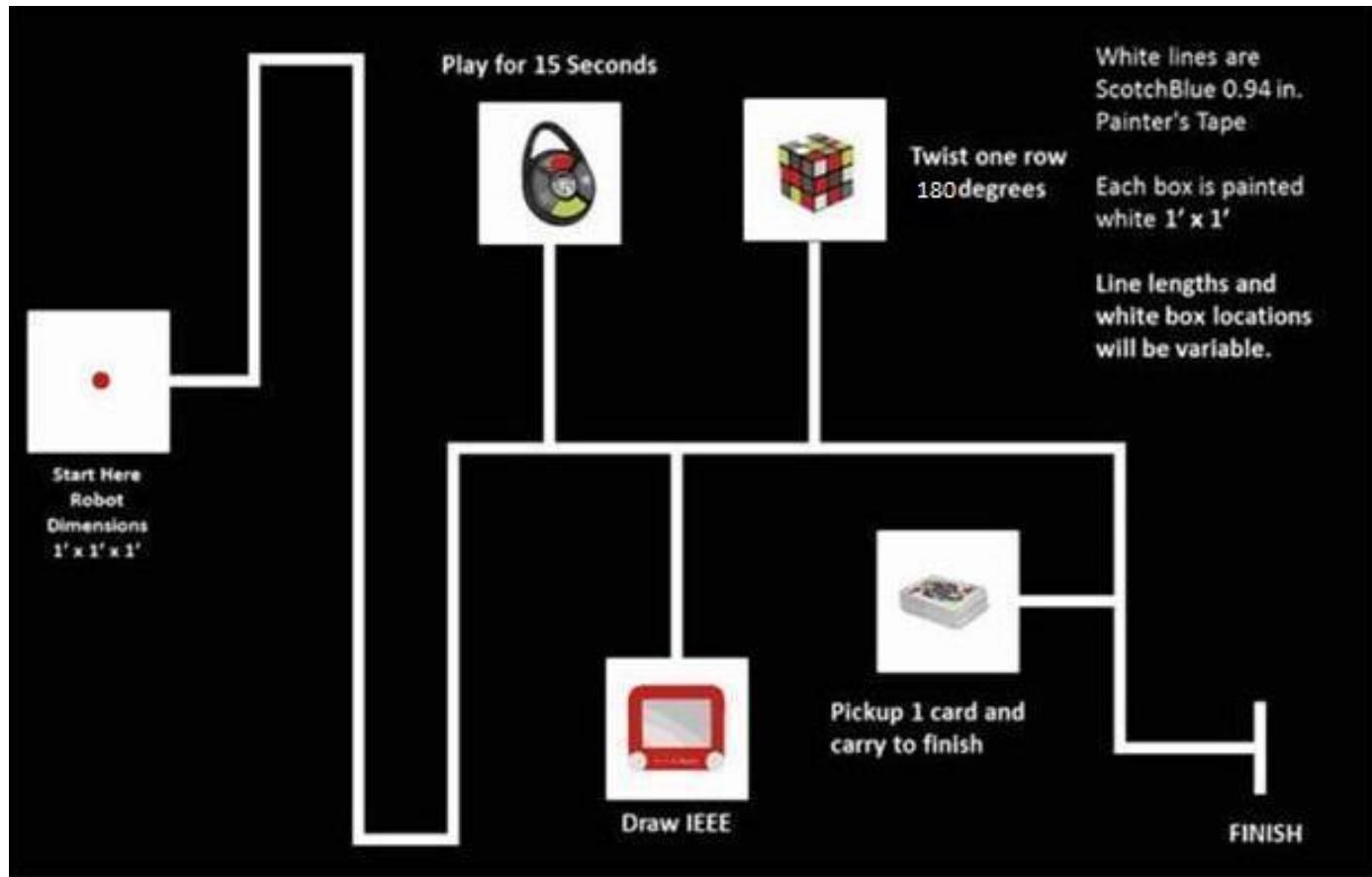
Table 26: Movement Test Case 9 Roadie Veer too Left

## 9. Appendix B

This appendix includes a diagram of the competition course as well as pictures of the individual challenges the system must complete. Also included is a picture of the tape that will designate the line the system must follow.

### 9.1 Competition Course

The course, as shown in **Fig. 34** below, shows the rough outline of the track the system will follow, as well as what a challenge station would look like.



**Fig. 34** Competition course for SoutheastCon[1].

## 10. References

- [1] IEEE Nova Southeastern University. (2014, September 7). IEEE SoutheastCon 2015 Student Program - Hardware Competition. Retrieved September 7, 2014, from IEEE SoutheastCon 2015: <http://www.ewh.ieee.org/reg/3/southeastcon2015/StudentProgram.html>
- [2] AWTY (2014, December 4) AWTY Level 2 Requirements. Retrieved December 4, 2014