****System Requirements Specifications for the Volt & Pepper System****

Sponsor

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

Released September 18, 2014

**Volt & Pepper Development Team**

**Abstract**: The System Requirements Specifications (SyRS) for the Volt & Pepper System are detailed within this document. The stakeholders are defined and the requirements of the product are specified. This document complies with the Institute of Electrical and Electronics Engineers (IEEE) Std. 1233-1998 [4] and the IEEE Recommended Practice for Software Requirements Specifications [5].

# Revision History

|  |  |  |
| --- | --- | --- |
| **Date** | **Reason for Change** | **Version** |
| Sep. 5, 2014 | Initial draft of document | 0.0.1 |
| Sep. 7, 2014 | Added requirements and user stories | 0.0.2 |
| Sep. 8, 2014 | Revised requirements, definitions added | 0.0.3 |
| Sep. 9, 2014 | Updated Definitions | 0.0.4 |
| Sep. 11, 2014 | Defined document sections | 0.0.5 |
| Sep. 12, 2014 | Compiled components of SyRS together | 0.0.6 |
| Sep. 14, 2014 | Refined document sections and styles | 0.0.7 |
| Sep. 15, 2014 | Abstract, references added | 0.0.8 |
| Sep. 16, 2014 | Use cases added, numbering adjusted | 0.0.9 |
| Sep. 17, 2014 | All sections editted | 0.0.10 |
| Sep. 18, 2014 | Final revisions | 0.0.11 |

**Table of Contents**

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Mission Statement 1

1.3 Scope 2

1.4 Course Components 2

1.5 Team Roles 3

1.6 Overview 3

2. General Description 3

2.1 Product Perspective 4

2.2 Product Functions 4

2.3 User Classes and Characteristics 4

2.4 General Constraints 4

2.5 Assumptions 5

2.6 Dependencies 5

2.7 Use Cases 5

Use Case 1: Startup 6

Use Case 2: Monitor Red LED 7

Use Case 3: Navigation 8

Use Case 4: Simon Carabiner 10

Use Case 5: Etch A Sketch 12

Use Case 6: Rubik’s Cube 14

Use Case 7: Playing Card 16

2.8 Sequence Diagrams 18

3. Requirements 24

3.1 Functional Requirement 24

3.2 Nonfunctional Requirements 25

A. Appendicies 26

A.1. Course component figures 26

A.2. Glossary 28

A.3 Acronyms & Abbreviations 29

References 30

# Introduction

This section introduces the information concerning the purpose of this document, mission statement, scope, team roles and an overview of the document.

## Purpose

The purpose of this document is to provide a detailed account of the scope, high-level description, and system requirements of the Volt & Pepper System, henceforth known as VPS. The requirements include functional and nonfunctional requirements and system constraints. The aim of this document is to clarify the requirements of the VPS.

## Mission Statement

To create a fully autonomous robot that can traverse over a path marked by a white line, and complete four tasks. These tasks include playing Simon (see Figure 1) for 15 seconds, drawing IEEE on an Etch A Sketch (see Figure 2), rotating one row of a Rubik’s Cube (see Figure 3) 180 degrees, and picking up and carrying a playing card from a standard 52-card deck (see Figure 4) to the end of the course.



Figure 1—Simon carabiner [11]

Figure 3—Rubik's Cube [1]

Figure 2—Etch A Sketch [9]



Figure 4—Standard 52- card deck [10]

## Scope

The VPS is intended to compete in the IEEE SoutheastCon 2015 Hardware [7]. SoutheastCon is the annual IEEE Region 3 Technical, Professional, and Student Conference. It brings together Computer Scientists, Electrical, and Computer Engineering professionals, faculty and students to share the latest information through technical sessions, tutorials, and exhibits [6].

## Course Components

The competition course is depicted below in Figure 5. The main components of the course are the starting square, the path, the challenge squares, the finish line. The starting square, labeled in Figure 5 as the white square with a red dot inside, is the 1 ft. x 1 ft. area where the VPS must start. In the starting box, the red dot represents the red LED that signals when the VPS may begin the course. The challenge boxes are in predefined locations on the course. Each challenge box holds the respective item as seen in Figure 5. The finish line is labeled “FINISH” and is the location the VPS must pass to complete the course. The path is marked by the white line connecting all previously mentioned compontents, and is subject to change in length.

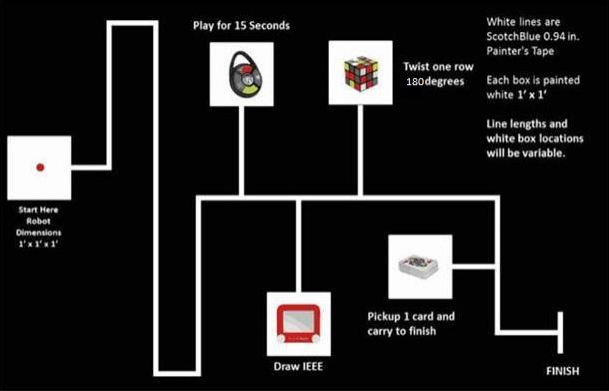


Figure 5—IEEE SoutheastCon 2015 Hardware Competition course [7]

## Team Roles

Table 1 presents all members of the Volt & Pepper System Development Team (VPSDT) and respective role assignments. Each member is accountable for the overesight and advancement of the positions held.

Table 1—Team roles

|  |  |
| --- | --- |
| **Name** | **Role** |
| Nezar Bahksh | Scrum Master  Development Team |
| Greg Carkin | Development Team |
| Gary Roach | Development Team |
| Brittany Rompa | Prodct Owner  Development Team |

## Overview

This document contains three primary sections. Section 1 contains an introduction to the VPS, section 2 contains the general description of the VPS and section 3 contains the functional and nonfunctional requirements of the VPS. Additional sections include an appendix and references.

# General Description

The information in this section is provided to improve understanding of the VPS. The areas covered include stakeholders, product perspective, product functions, user characteristics, general constraints, assumptions, dependencies, risks, use cases, and sequence diagrams.

**Stakeholders**

The following list contain the individual parties that have a stake in the development, production, and operation of the VPS.

1. **Volt & Pepper System Development Team**

The VPSDT will be graded on the adequate completion of the system by the sponsors (see 2.1.2). Grading of the project includes the VPS itself, along with all artifacts created throughout the 2014-2015 Senior Design course.

1. **Dr. Barott, Dr. Seker, Jorge Torres**

As sponsors of the VPS, Dr. Barott, Dr. Seker, and Jorge Torres are interested in the completion of the VPS and all artifacts created throughout the 2014-2015 Senior Design course. Additionally, Dr. Barott and Dr. Seker are interested in the VPSDT meeting the standards set forth by the Accredation Board of Engineering and Technology (ABET), for a Capstone Senior Design project.

1. **Embry-Riddle Aeronautical University**

The VPS will be one of three contending teams from Embry-Riddle Aeronautical University (ERAU) competing to be sent to the IEEE SoutheastCon 2015 Hardware Competition [7]. If chosen, the final product produced by the VPSDT will uphold the prestige of ERAU. In doing so, the VPS must conform to the standards of ERAU as defined in the 2014-2015 Student Handbook [2]**.**

1. **Department of Electrical Computer Software & Systems Engineering**

The Department of Electrical Computer Software & Systems Engineering (ECSSE) at ERAU is interested in the product staying within budget as well as being delivered on time (TBD as of 9/18/14).

1. **Nova Southeastern University & Broward College**

Nova Southeastern University & Broward College is the institution hosting the IEEE SoutheastCon Hardware Competition. They are interested in the product complying with all regulations for the competition. It is essential to maintain a safe environment by assurring no product will cause harm to the institiution, or any persons which may come into contact with the product.

## Product Perspective

The VPS will be a new self-contained product that will accomplish the series of challenges set forth by the 2015 IEEE SoutheastCon Hardware Competition [7].

## Product Functions

The VPS must pertform the following functions:

* + 1. Monitor the red LED.
    2. Navigate the course.
    3. Play the Simon game.
    4. Draw “IEEE” on the Etch A Sketch.
    5. Rotate one row of the Rubik’s Cube 180 degrees.
    6. Pick up one playing card from a standard 52-card deck.
    7. Cross the finish within 5 minutes while holding the card.

## User Classes and Characteristics

The VPS is intended to be used by the undergraduate senior design team, the VPSDT, and the VPS sponsors. The characteristics required to use the VPS include: technical modifications, intermediate-level programming skills, familiarity with the hardware, and familiarity with the software.

## General Constraints

The VPS is constrained by the IEEE regulations for the SoutheastCon competetion. These constraints include:

* + 1. The physical size of the VPS may not exceed 1ft. x 1 ft. x 1 ft.
    2. The VPS must remain as one unit while operating.
    3. The VPS cannot have human interaction while operating.
    4. While operating, the VPS cannot move any object outside the object station marked by the 1 ft. x 1 ft. white box.
    5. The VPS cannot contain any flammable substances.
    6. The VPS cannot project an object.
    7. While operating, the VPS must cover the white line.
    8. Given the course dimensions the size constrain denies us by default from executing multiple challenges simultaneously.
    9. The VPS is constrained to finish all tasks in a time interval less than 5 minutes.

## Assumptions

The following are assumptions that the VPSDT has made about the VPS.

* + 1. The VPSDT assumes the VPS can complete the course without running out of power.
    2. The VPSDT assumes the VPS can be designed, developed, tested and operated without harming any user.
    3. The VPSDT assumes the VPS can complete the course without falling apart.

## Dependencies

The following are dependencies of the

* + 1. The VPSDT depends on IEEE regulations; as the IEEE regulations change, the VPS requirements will also have to update correspondingly.
    2. The VPSDT depends on funding from the Department of ECSSE at ERAU.
    3. The VPSDT grade depends on the VPS fulfilling all requirements (see section 3).
    4. The VPSDT depends on the reliability of shipping services in order to meet deadlines.

## Use Cases

The following use cases demonstrate the proper functionality of the autonomous VPS. These use cases represent the required operation in order to receive maximum points for the competition. Deviation from the normal operation of the VPS may result in a loss of points or disqualification.

Use Case 1: Startup

Scope: VPS

Level: User goal

Primary Actors: Volt and Pepper team member

Preconditions

1. The VPS is not on (power to the VPS is disabled)
2. The VPS is not within the 1 ft. x 1 ft. starting square

Postconditions

1. The VPS is on (power to the VPS is enabled)
2. The VPS is within the 1 ft. x 1 ft. starting square

Main Success Scenario

* + - 1. The team member places the VPS within the 1 ft. x 1 ft. starting square, ensuring that the VPS is facing towards the desired path of travel
      2. The team member activates the VPS (enables power to the VPS)

Extensions (Alternate Flows)

1. The VPS does not turn on
   1. The user restarts the system
      1. If the system turns on
         1. The VPS resumes startup operations
      2. If the system does not turn on
         1. The user must replace the battery
         2. The user restarts the system

Frequency of Occurrence

This use case will occur at the start of each round of the competition. Amount of rounds will vary based on amount of participants, successful completion of the course, as well as the time it takes to achieve successful completion of the course.

Use Case 2: Monitor Red LED

Scope: VPS

Level: User goal

Primary Actors: Red LED

Preconditions

1. The VPS is on (power to the VPS is enabled) as described in Use Case 1: Startup
2. The VPS is within the 1 ft. x 1 ft. starting square as described in Use Case 1: Startup
3. The Red LED light is on (the light is illuminated)

Postconditions

1. The VPS will begin the task of navigation, as described in Use Case 3: Navigation
2. The VPS is within the 1 ft. x 1 ft. starting square, as described in Use Case 1: Startup
3. The red LED light is off (the light is not illuminated)

Main Success Scenario

1. The VPS will wait for the red LED light to turn off
2. The red LED light will turn off
3. The VPS will begin the task of navigation, as described in Use Case 3: Navigation

Frequency of Occurrence

This use case will occur once during each round of the competition. Amount of rounds will vary based on amount of participants, successful completion of the course, as well as the time it takes to achieve successful completion of the course.

Use Case 3: Navigation

Scope: VPS

Level: User goal

Primary Actors: White line

Stakeholders and Interests

<insert stakeholders that apply to this use case>

Preconditions

1. The VPS is on (power to the VPS is enabled), as described in Use Case 1: Startup
2. The VPS is within the 1 ft. x 1 ft. starting square or in front of one of the 4 challenges, as described in Use Case 1: Startup, Use Case 4: Simon Carabiner, Use Case 5: Etch A Sketch, Use Case 6: Rubik’s Cube, and Use Case 7: Playing Card
3. The Red LED light is off (the light is not illuminated) or the challenge the VPS is currently in front of has been successfully completed, as described in Use Case 2: Monitor Red LED, Use Case 4: Simon Carabiner, Use Case 5: Etch A Sketch, Use Case 6: Rubik’s Cube, and Use Case 7: Playing Card

Postconditions

1. The VPS will be halted at the finish line or in front of one of the 4 challenges, as described in Use Case 4: Simon Carabiner, Use Case 5: Etch A Sketch, Use Case 6: Rubik’s Cube, and Use Case 7: Playing Card
2. The VPS will be touching the white line of the finish line or the painted white line which makes up the box that the challenge resides in

Main Success Scenario

1. The VPS will follow the white line, as it heads to the finish line or the next challenge
2. The VPS will cover the white line at all times on route to its destination
3. The VPS will visit and complete each challenge, as described in Use Case 4: Simon Carabiner, Use Case 5: Etch A Sketch, Use Case 6: Rubik’s Cube, and Use Case 7: Playing Card
4. The VPS will, after completion of each challenge(as described in Use Case 4: Simon Carabiner, Use Case 5: Etch A Sketch, Use Case 6: Rubik’s Cube, and Use Case 7: Playing Card), will navigate to and halt at the finish line
5. The VPS will remain halted at the finish line

Extensions (Alternate Flows)

1. The VPS navigates off the line.
   1. The VPS interrupts the current navigation.
   2. The VPS attempts to find nearby line.
      1. If a line is found.
         1. The VPS travels to the line.
         2. The VPS resumes navigation.
      2. If a line is not found.
         1. The VPS reverses.
         2. If the VPS finds a line.
            1. The VPS travels to the line.
            2. The VPS resumes navigation.
         3. If the VPS finds the edge of the course.
            1. The VPS scans immediate area for line

If the VPS finds a line

The VPS travels to the line

The VPS resumes navigation

The VPS does not find a line

The VPS is lost.

Frequency of Occurrence

This use case will occur 5 times during each round of the competition. The use case will be used each time the VPS must navigate to the next challenge (four challenges in total, described in Use Case 4: Simon Carabiner, Use Case 5: Etch A Sketch, Use Case 6: Rubik’s Cube, and Use Case 7: Playing Card), then once more to navigate to the finish line. Amount of rounds will vary based on amount of participants, successful completion of the course, as well as the time it takes to achieve successful completion of the course.

Use Case 4: Simon Carabiner

Scope: VPS

Level: User goal

Primary Actors: Simon Carabiner

Stakeholders and Interests

<insert stakeholders that apply to this use case>

Preconditions

1. The VPS is on (power to the VPS is enabled), as described in Use Case 1: Startup
2. The VPS is directly in front of the Simon Carabiner game, touching the painted white square in which the game resides, as described in Use Case 3: Navigation

Postconditions

1. The VPS is on (power to the VPS is enabled), as described in Use Case 1: Startup
2. The VPS is directly in front of the Simon Carabiner game, touching the painted white square in which the game resides, as described in Use Case 3: Navigation

Main Success Scenario

1. The VPS will depress (push) the center button on the game to start Simon Carabiner
2. The VPS will wait for the visual and audible pattern emitted from the game.
3. The VPS will then duplicate this pattern by depressing (pushing) the corresponding buttons on the Simon Carabiner game.
4. The VPS will then wait for the next visual and audible pattern to be emitted from the game.
5. The VPS will once again duplicate this pattern by depressing (pushing) the corresponding buttons on the Simon Carabiner game.
6. The VPS, after 15 seconds had passed since depressing (pushing) the center button to start the game, will then depress (push) the center button on the game to stop Simon Carabiner
7. The VPS will then begin navigation, as described in Use Case 3: Navigation

Extensions (Alternate Flows)

1. The VPS presses an incorrect button on Simon.
   1. The VPS interrupts its current tasks for Simon.
   2. The VPS resets Simon.
   3. The VPS resumes playing Simon.

Frequency of Occurrence

This use case will occur once during each round of the competition. Amount of rounds will vary based on amount of participants, successful completion of the course, as well as the time it takes to achieve successful completion of the course.

Use Case 5: Etch A Sketch

Scope: VPS

Level: User goal

Primary Actors: Etch A Sketch

Stakeholders and Interests

<insert stakeholders that apply to this use case>

Preconditions

1. The VPS is on (power to the VPS is enabled), as described in Use Case 1: Startup
2. The VPS is directly in front of the Etch A Sketch game, touching the painted white square in which the game resides, as described in Use Case 3: Navigation
3. The Etch A Sketch will be blank (nothing has been drawn on the Etch A Sketch display)

Postconditions

1. The VPS is on (power to the VPS is enabled), as described in Use Case 1: Startup
2. The VPS is directly in front of the Etch A Sketch game, touching the painted white square in which the game resides, as described in Use Case 3: Navigation
3. The Etch A Sketch will display “IEEE” (the letters “IEEE” have been drawn on the Etch A Sketch display)

Main Success Scenario

1. The VPS will begin to turn the knobs of the Etch A Sketch
2. The VPS will continue to turn the knobs in a manner in which the Etch A Sketch will display the letters “IEEE”
3. The VPS will then begin navigation, as described in Use Case 3: Navigation

Extensions (Alternate Flows)

1. The VPS drops the playing card.
   1. The VPS interrupts its current tasks for the playing card.
   2. The VPS attempts to find the dropped card.
      1. If the card is located.
         1. The VPS picks up the card.
         2. The VPS resumes operation before it was interrupted.
      2. If the card is not located.
         1. The VPS returns to the stack of cards.
         2. The VPS picks up a new card.
         3. The VPS resumes operation before it was interrupt

Frequency of Occurrence

This use case will occur once during each round of the competition. Amount of rounds will vary based on amount of participants, successful completion of the course, as well as the time it takes to achieve successful completion of the course.

Use Case 6: Rubik’s Cube

Scope: VPS

Level: User goal

Primary Actors: Rubik’s Cube

Stakeholders and Interests

<insert stakeholders that apply to this use case>

Preconditions

1. The VPS is on (power to the VPS is enabled), as described in Use Case 1: Startup
2. The VPS is directly in front of the Rubik’s Cube game, touching the painted white square in which the game resides, as described in Use Case 3: Navigation
3. The Rubik’s Cube will be solved (no rows are turned from its initial, in package, condition)

Postconditions

1. The VPS is on (power to the VPS is enabled), as described in Use Case 1: Startup
2. The VPS is directly in front of the Rubik’s Cube game, touching the painted white square in which the game resides, as described in Use Case 3: Navigation
3. The Rubik’s Cube will have one row turned 180 degrees (from its initial, in package, condition)

Main Success Scenario

1. The VPS will begin to turn a row on the Rubik’s Cube
2. The VPS will continue to turn the row on the Rubik’s Cube until it has turned 180 degrees
3. The VPS will then begin navigation, as described in Use Case 3: Navigation

Extensions (Alternate Flows)

1. The VPS misplaces the Etch A Sketch.
   1. The VPS interrupts its current tasks for the Etch A Sketch.
   2. The VPS attempts to find the misplaced Etch A Sketch.
      1. If the Etch A Sketch is located.
         1. The VPS realigns to the Etch A Sketch
         2. The VPS reattaches to the Etch A Sketch
         3. The VPS resumes its interrupted task
      2. If the Etch A Sketch is not located.
         1. The VPS scans the immediate area
            1. If the VPS locates Etch A Sketch

The VPS realigns to the Etch A Sketch

The VPS reattaches to the Etch A Sketch

The VPS resumes its interrupted task

* + 1. If the VPS does not locate the Etch A Sketch
       1. The VPS moves on to the next task

Frequency of Occurrence

This use case will occur once during each round of the competition. Amount of rounds will vary based on amount of participants, successful completion of the course, as well as the time it takes to achieve successful completion of the course.

Use Case 7: Playing Card

Scope: VPS

Level: User goal

Primary Actors: Deck of Playing Cards

Stakeholders and Interests

<insert stakeholders that apply to this use case>

Preconditions

1. The VPS is on (power to the VPS is enabled), as described in Use Case 1: Startup
2. The VPS is directly in front of the deck of playing cards, touching the painted white square in which the game resides, as described in Use Case 3: Navigation
3. The deck of cards will contain 52 cards

Postconditions

1. The VPS is on (power to the VPS is enabled), as described in Use Case 1: Startup
2. The VPS is directly in front of the deck of playing cards, touching the painted white square in which the game resides, as described in Use Case 3: Navigation
3. The VPS will have possession of one playing card
4. The deck of cards will contain 51 cards

Main Success Scenario

1. The VPS will pick up the playing card
2. The VPS will maintain possession of the playing card
3. The VPS will then begin navigation, as described in Use Case 3: Navigation

Extensions (Alternate Flows)

1. The VPS misplaces the Rubik’s Cube
   1. The VPS interrupts its current task
   2. The VPS attempts to find the Rubik’s Cube
      1. If the Rubik’s Cube is found
         1. The VPS aligns to the Rubik’s Cube
         2. The VPS holds the Rubik’s Cube
         3. The VPS resumes the interrupted task
      2. If the Rubik’s Cube is not found
         1. Scan the immediate area for the cube
            1. If the Rubik’s Cube is found

The VPS aligns to the Rubik’s Cube

The VPS holds the Rubik’s Cube

The VPS resumes the interrupted task

* + - * 1. If the Rubik’s Cube is not found

Navigate to next task

Frequency of Occurrence

This use case will occur once during each round of the competition. Amount of rounds will vary based on amount of participants, successful completion of the course, as well as the time it takes to achieve successful completion of the course.

## Sequence Diagrams

The following diagrams provide a sequence of actions in order to complete a task. The tasks are based on the requirements needed to complete the course. Initially, the VPS must be turned on and be setup to await the starting signal. Once the VPS recieves the start signal it then transitions into a navigation phase where it remains on the white line as it moves to a challenge. The navigation phase occurs between each of the challenges. The four challenges, Simon, Etch A Sketch, Rubik’s cube, and the playing card each are considered their own tasks. The components of the VPS are broken up into the RobotController, sensors, interactors, and MovementSystem. The RobotController acts as the central hub of the system where commands are given out. The sensors are the sensors for the VPS. The interactors are the physical components used to touch the challenges. Lastly the MovementSystem is the components of the VPS dedicated to traversing the course.

The setup sequence diagram depicts the sequence of actions needed for the VPS to complete its setup of components and start of navigation. The RobotController sends a message to each of the components to setup. Once the sensors, interactors and MovementSystems have setup the Robotontroller sends a message to the sensors to sense the LED for a start signal. Once the start signal is received the RobotController sends messages to the sensors and MovementSystem to start navigation.

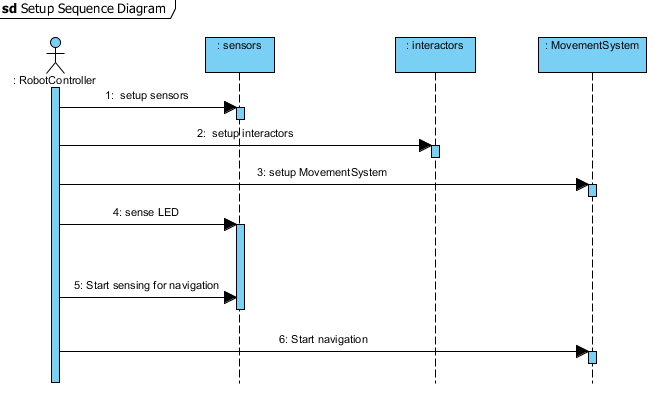


Figure 5—Setup sequence diagram

The navigation sequence diagram depicts the sequence of actions needed for the VPS to successfully traverse the course. Durring this sequence the RobotController is constantly messaging the sensors and MovementSystem to ensure the VPS is staying on the white line.

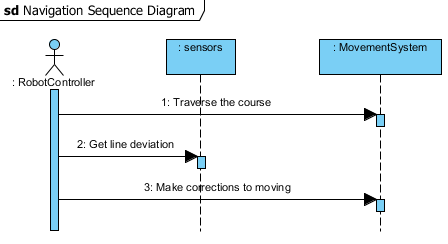


Figure 6—Navigation sequence diagram

The simon sequence diagram depicts the sequence of actions needed for the VPS to successfully complete the simon challenge and enter navigation. Initially the RobotController sends a message to the sensors to find simon. Once the VPS has found simon the RobotController sends messages to the MovementSystem and the interactors to align with simon. Once the VPS is aligned the RobotController sends messages to the sensors and interactors to play simon until the challenge is complete. After the VPS has completed the simon challenge the RobotController sends messages to the sensors, and interactors to stop sensing and interacting with simon. Lastly the RobotController sends messages to the sensors and MovementSystem to start navigation.

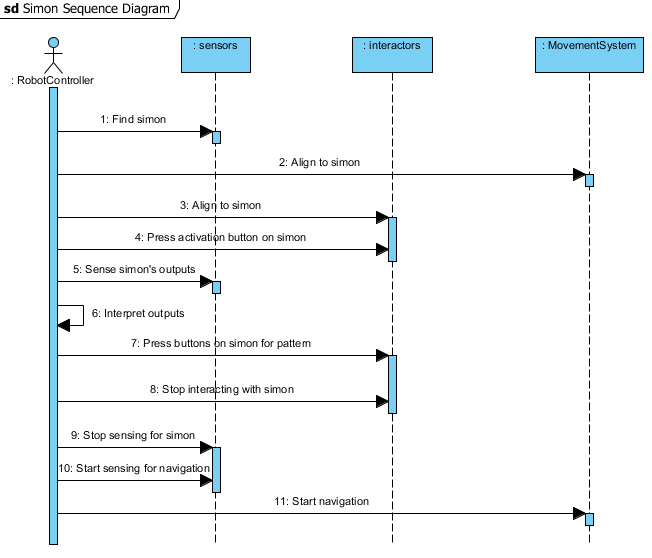
****

Figure 7—Simon sequence diagram

The Etch A Sketch sequence diagram depicts the sequence of actions needed for the VPS to successfully complete the Etch A Sketch challenge and enter navigation. Initially the RobotController sends a message to the sensors to find the Etch A Sketch. Once the VPS has found the Etch A Sketch the RobotController sends messages to the MovementSystem and the sensors to align with the Etch A Sketch. Once the VPS is aligned the RobotController sends messages to the interactors to print IEEE onto the Etch A Sketch. After the VPS has printed IEEE on the Etch A Sketch the RobotController sends messages to the sensors, and interactors to stop sensing and interacting with the Etch A Sketch. Lastly the RobotController sends messages to the sensors and Movement system to start navigation.

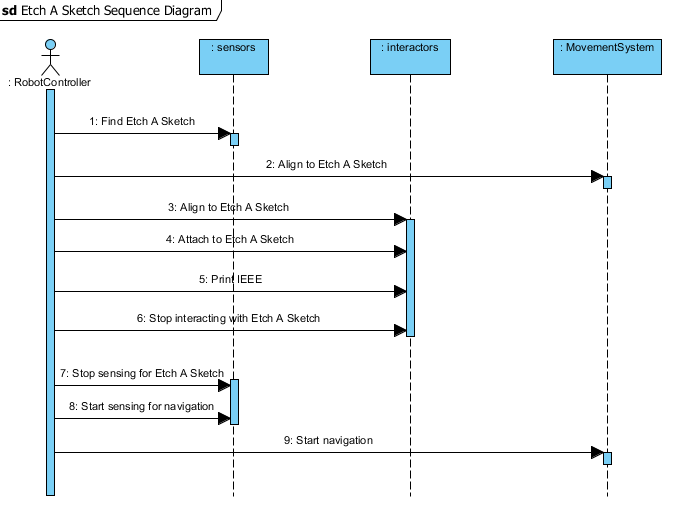


Figure 8—Etch A Sketch sequence diagram

The Rubik’s Cube sequence diagram depicts the sequence of actions needed for the VPS to successfully complete the Rubik’s Cube challenge and enter navigation. Initially the RobotController sends messages to the sensors to find the cube. Once the sensors locate the cube the RobotController sends messages to the MovementSystem and the interactors to align with the cube. Once the VPS is aligned the RobotController sends messages to the interactors to rotate the cube. After the VPS has rotated the cube the RobotController sends messages to the sensors and interactors to stop sensing and interacting with the cube. The RobotController then sends messages to the sensors and MovementSystem to start navigation.

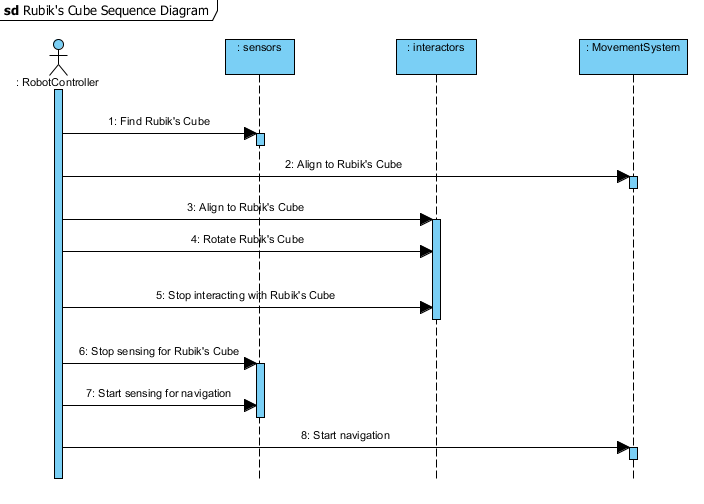


Figure 9—Rubik's Cube sequence diagram

The card sequence diagram depicts the sequence of actions needed for the VPS to successfully complete the card challenge and enter navigation. Initially the RobotController sends a message to the sensors to find the playing card. Once the VPS has found the card the RobotController sends messages to the MovementSystem and the interactors to align with the card. After the VPS is aligned the RobotController sends a message to the interactors to take hold of a card. The RobotController then sends messages to the sensors stop sensing for the card. Lastly the RobotController sends messages to the sensors and the MovementSystem to start navigation.

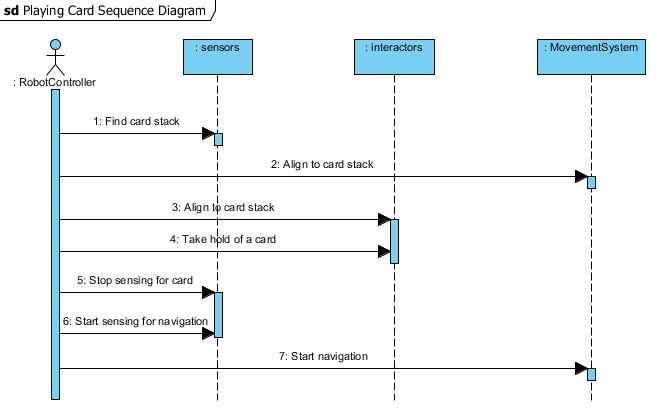


Figure 10—Playing card sequence diagram

# Requirements

## Functional Requirement

* + 1. The VPS shall exit the 1 ft. x 1 ft. starting square when the red LED turns off.
    2. The VPS shall traverse the white line on the course.
       1. The VPS shall remain on the white line for at least TBD % of the course.
       2. The VPS shall move to each challenge.
    3. The VPS shall play the Simon game for 15 seconds.
       1. The VPS shall press the middle button on the Simon game to activate the Simon game.
       2. The VPS shall respond to the Simon game after the Simon game dictates a sequence
          1. The VPS shall press the buttons on the Simon game in the sequence dictated by the Simon game.
          2. The VPS shall respond to the Simon game before the Simon game’s error buzzer sounds.
       3. The VPS shall leave the Simon game in the Simon game area.
    4. The VPS shall draw the letters “IEEE” on the Etch A Sketch in a TBD lettering style.
       1. The VPS shall rotate the Etch A Sketch knobs to draw the letters “IEEE”.
       2. The VPS shall leave the Etch A Sketch in the Etch A Sketch area.
    5. The VPS shall rotate one row of the Rubik’s Cube 180 degrees relative to the other two rows.
    6. The VPS shall leave the Rubik’s Cube in the Rubik’s Cube area.
    7. The VPS shall obtain one playing card from the stack of cards on the course.
       1. The VPS shall leave the remaining cards in the playing card area.
    8. The VPS shall cross the finish line holding the obtained playing card.
    9. The VPS shall cross the finish line after completing all games.

## Nonfunctional Requirements

* + 1. The VPS shall fit within 1 ft. x 1 ft. x 1 ft. if it is in the starting square.
    2. The VPS shall fit within 1 ft. x 1 ft. x 1 ft. to cross the finish line.
    3. The VPS shall operate without human interaction after activation.
    4. The VPS shall operate under its own onboard power supply.
    5. The VPS shall remain within the course.
    6. The VPS shall cross the finish line in no more than 5 minutes after the LED turns off.
    7. The VPS shall remain in the 1 ft. x 1 ft. starting square while the LED is on.

# A. Appendicies

## A.1. Course component figures

The following figures are a supplemental visual aid of the IEEE SoutheastCon 2015 Hardware Competition course and challenge components.

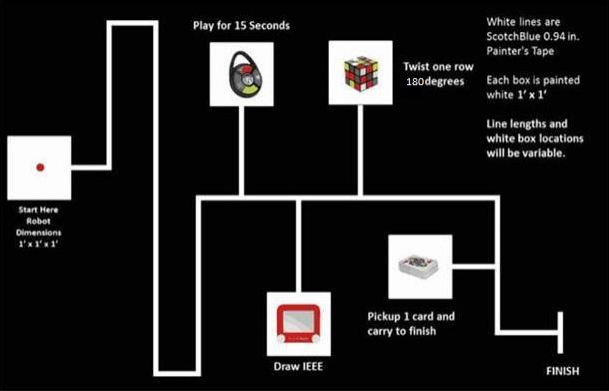
[](#_Table_of_Figures)

Figure —Course for IEEE SoutheastCon 2015 Hardware Competition [7]



Figure 2—Etch A Sketch [9]



Figure 3—Standard 52-card deck of playing cards [10]



Figure 4—Rubik’s 3x3 Cube [1]



Figure 5—Simon Carabiner [11]

## A.2. Glossary

|  |  |  |
| --- | --- | --- |
| **Entry** | **Definition** | **Alias** |
| autonomous | When activated, it is independent with no outside influence |  |
| challenge | One of the four tasks- Simon, Etch A Sketch, Rubik’s Cube, or playing card | Task |
| course | 5/8 in. x 4 ft. x 8 ft. Sanded Pine Plywood |  |
| deck of cards | Standard 52-card deck (see Appendix A.1, Figure 3) | Deck |
| foot | A unit of measurement | ft. |
| draw | To produce or print |  |
| emergency | Unexpected occurrence requiring human intervention |  |
| Etch A Sketch | Pocket Etch A Sketch by: Ohio Art (see Appendix A.1, Figure 2) |  |
| finish line | Refer to “FINISH” (see Appendix A.1, Figure 1) | Finish |
| interact | Physically affecting by executing the functional requirments |  |
| line | Scotch Blue 0.94 in. x 60 yd. Painter’s Tape |  |
| obtain | To have possesion of |  |
| playing card | A card from the standard 52-card deck (see Appendix A.1, Figure 3) | Card |
| robot | The platform being built for the IEEE SoutheastCon 2015 Hardware Competition. | VPS |
| Rubik’s Cube | Rubik’s 3x3 Cube (see Appendix A, Figure 4) |  |
| sequence | Simon, Etch A Sketch, Rubik’s Cube, playing card, finish line |  |
| Simon | Simon Carabiner (see Appendix A, Figure 5) |  |
| traverse | To move across |  |

## A.3 Acronyms & Abbreviations

|  |  |
| --- | --- |
| **Entry** | **Expanded Phrase** |
| ABET | Accredation Board for Engineering and Technology, Inc. |
| ERAU | Embry-Riddle Aeronautical University |
| ECSSE | Electrical, Computer, Software & Systems Engineering |
| IEEE | Institution of Electrical and Electronics Engineers, Inc. |
| SyRS | System Requirements Specifications |
| VPS | Volt & Pepper System |
| VPSDT | Volt & Pepper System Development Team |
| in. | Inch |
| ft. | Foot |
| yd. | Yard |

# References

1. Booyabazooka. *Rubik's Cube.* March 5, 2008. http://commons.wikimedia.org/wiki/File:Rubik%27s\_cube.svg (accessed September 13, 2014).
2. ERAU. "Student Handbook - Embry-Riddle Aeronautical University." *ERAU - Daytona Beach, FL.* 2014. http://daytonabeach.erau.edu/Assets/daytonabeach/forms/daytonabeach-student-handbook.pdf (accessed September 13, 2014).
3. IEEE. *IEEE Citation Reference.* September 9, 2014. http://www.ieee.org/documents/ieeecitationref.pdf (accessed 2014).
4. IEEE. *IEEE Guide for Developing System Requirements Specifications.* 1998. New York, NY: IEEE, Decembeer 22, 1998.
5. IEEE. *IEEE Recommended Practice for Software Requirements Specifications.* New York, NY: Institute of Electrical and Electronics Engineers, Inc., 1998.
6. —. *IEEE Region 3.* 2014. http://www.ewh.ieee.org/reg/3/southeastcon/ (accessed 2014).
7. —. "SoutheastCon 2015 Hardware Competition Rules (DRAFT)." *IEEE.* March 19, 2014. http://sites.ieee.org/sb-unfc/files/2014/07/hardwareComp2015.pdf (accessed September 13, 2014).
8. IEEE Standards Association. *2014 IEEE-SA Standards Style Manual.* New York, NY: IEEE, 2014.
9. Ohio Art. *Pocket Etch A Sketch - Red.* Ohio Art. 2014. http://www.toysrus.com/buy/etch-a-sketch-doodle-pro/pocket-etch-a-sketch-red-5163-2395954 (accessed September 13, 2014).
10. Plank Fitness. *Deck of Cards "Anywhere" Workout.* January 3, 2014. http://plankavl.com/wp-content/uploads/2014/01/deck-cards.png (accessed September 13, 2014).
11. "System Requirements Specifications for Birdinatore Autonomous System." ECSSE, Embry-Riddle Aeronautical University, 2013.
12. The Bridge Direct, Inc. *Simon: Carabiner Edition.* 2012. http://www.thebridgedirect.com/sim\_prd\_carabiner.php (accessed September 13, 2014).