**Software Requirements Specification**

**for**

**OSPREY**

**Version 2.0 approved**

**Prepared by Alvarez, Olin**

**Ho, Henry**

**Padilla, Danny**

**Song, Jonathan**

**Tran, Kevin**

**RoboSub**

**November 22, 2017**

# Table of Contents

Table of Contents................................................................................................................... <pg 1>

Revision History..................................................................................................................... <pg 2>

1. Introduction................................................................................................................ <pg 2>
	1. Purpose........................................................................................................... <pg 2>
	2. Intended Audience and Reading Suggestions................................................ <pg 3>
	3. Product Scope................................................................................................ <pg 3>
	4. Definitions, Acronyms, and Abbreviations .................................................. <pg 3>
	5. References......................................................................................................<pg 3>
2. Overall Description.................................................................................................... <pg 4>
	1. Product Perspective........................................................................................ <pg 4>
	2. Product Functions...........................................................................................<pg 4>
	3. User Classes and Characteristics....................................................................<pg 4>
	4. Operating Environment.................................................................................. <pg 4>
	5. Design and Implementation Constraints........................................................ <pg 4>
	6. User Documentation...................................................................................... <pg 5>
	7. Assumptions and Dependencies.................................................................... <pg 6>
	8. Apportioning of Requirements...................................................................... <pg 7>
3. External Interface Requirements............................................................................... <pg 7>
	1. User Interfaces............................................................................................... <pg 7>
	2. Hardware Interfaces....................................................................................... <pg 7>
	3. Software Interfaces........................................................................................ <pg 7>
	4. Communications Interfaces........................................................................... <pg 8>
4. Requirements Specification....................................................................................... <pg 9>
	1. Functional Requirements............................................................................... <pg 9>
	2. External Interface Requirements..................................................................<pg 12>
	3. Logical Database Requirements...................................................................<pg 12>
	4. Design Constraints.......................................................................................<pg 12>
5. Other Nonfunctional Requirements.........................................................................<pg 13>
	1. Performance Requirements..........................................................................<pg 13>
	2. Safety Requirements....................................................................................<pg 13>
	3. Security Requirements.................................................................................<pg 13>
	4. Software Quality Attributes.........................................................................<pg 13>
	5. Business Rules.............................................................................................<pg 13>
6. Other Requirements.................................................................................................<pg 13>

Appendix A: Glossary.........................................................................................................<pg 14>

Appendix B: Analysis Models............................................................................................<pg 15>

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Programming Language |  9/10/17 |  Changed from primarily MatLab/C++ to primarily Python for consistency |  2.0 |
| Refactoring/Modularizing code | 11/08/17 | Making code modular for better readability |  2.0 |

#

# 1. Introduction

The Osprey software is responsible for controlling a fully autonomous underwater vehicle (AUV). The AUV will compete in an international competition comprised of a series of visual and acoustic based tasks. These tasks simulate challenges commonly faced by other competing autonomous submarines.

##

## 1.1 Purpose

The purpose of this document is to define a full set of requirements for the software of the Autonomous Underwater Vehicle 2018 version 2.0. The Purpose of the document is **five-fold**:

1. Completely define the overall description for **Osprey - Section 2.**
2. Completely define the external interface requirements for **Osprey - Section 3.**
3. Completely define the requirements specifications for **Osprey - Section 4.**
4. Completely define the non-functional requirements for **Osprey - Section 5.**
5. Completely define the other requirements for **Osprey - Section 6.**

## 1.2 Intended Audience and Reading Suggestions

This document is intended for the software developers. The Software Requirements Specifications (SRS) contains documentation on how the Osprey software works overall for the AUV and list of requirements that the Osprey must meet.

## 1.3 Product Scope

* The software product produced is AUV-Osprey.
* The software will not allow the user to control the AUV.
* The software will provide autonomy to the AUV.
* The software will provide a user interface to modify task parameters.
* Once implemented, the AUV will complete tasks specified in the Competition Requirements.

## 1.4 Definitions, Acronyms, and Abbreviations

All Definitions, Acronyms and abbreviations will be listed in Appendix A: Glossary

## 1.5 References

Website address referenced

1. Robosub Competition
	* <http://www.robonation.org/competition/robosub>
2. Operating System
	* <http://wiki.ros.org/lunar/Installation/Ubuntu>
3. OpenCV Library Documentation
	* <https://docs.opencv.org/trunk/index.html>
4. Robot Operating System Documentation
	* <http://wiki.ros.org/Documentation>
5. Arduino
	* <https://www.arduino.cc/en/Guide/Introduction>
	* <https://www.arduino.cc/en/Reference/Wire>

Documents Referenced

1. Robosub Draft Tasks 2018
	1. <http://www.robonation.org/sites/default/files/2018%20RoboSub%20Tasks_DRAFT%2011%2001%2017.pdf>

1.5.1 Controlling Documents

* There is no other document controlling this document.

1.5.2 Applicable Documents

* No additional applicable document has been used in the production of this document.

1.5.3 Standards

* ROS Python Standards
	+ <http://wiki.ros.org/PyStyleGuide>
* ROS C++ Standards
	+ <http://wiki.ros.org/CppStyleGuide>

# 2. Overall Description

The Osprey software will operate the AUV in an underwater environment. Osprey shall achieve this by manipulating several pieces of hardware through a central hub. Osprey shall detect specific tasks, and obstacles throughout the competition. Once detected, the software shall provide the AUV’s hardware with instructions to reach each task and handle the scope of each task’s goal. Osprey shall guide, and maneuver the AUV around the competition course, as well as provide stabilization. Osprey will also provide a graphical user interface accessible in between runs, such that task parameters may be modified.

## 2.1 Product Perspective

The Osprey relates to products created by other competitors within RoboSub competition. The Osprey is not a component of a larger system. Similarities are in the scope of each task and the operations required to accomplish them. Differences are in the specific techniques and methods applied for each task. Differences also includes libraries and hardware.

## 2.2 Product Functions

Product functions include, but are not limited to:

* Traversing through an underwater environment autonomously
* Completing several different tasks autonomously with the use of different hardware
* Detecting and differentiating between distinct objects and obstacles

## 2.3 User Classes and Characteristics

* User Classes and Characteristics are not applicable for this product

## 2.4 Operating Environment

Hardware Platform

* Mini-ITX SBC MANO882
* Intel i5 CPU
* 8GB RAM
* Samsung 750 Evo 256GB SSD

Operating System

* Ubuntu 17.04
* ROS Lunar-Loggerhead

## 2.5 Design and Implementation Constraints

Regulatory Policies

* Types of tasks that will be in the competition
* Rules and Limitations to how we can accomplish each task

Hardware Limitation

* Available Sensors

Environmental Conditions

* Underwater Submersion
* Lighting Conditions

Memory Constraints

* Microcontroller ATmega2560 has a limited amount of memory, which will prevent it from having too many global variables

##

##

## 2.6 User Documentation

* <http://wiki.ros.org>
* <https://docs.opencv.org/trunk/index.html>
* <http://scikit-learn.org/stable/documentation.html>
* <https://www.arduino.cc/en/Guide/Introduction>

## 2.7 Assumptions and Dependencies

Other Factors

* Competition tasks and rules may be subject to change

Commercial/Third-Party Components:

* Robot Operating System (ROS)
* Linux distribution (Ubuntu 17.04)
* Arduino IDE (ATmega2560/ATmega328P)
* Vector Nav VN100 IMU Linux Driver
* OpenCV
* NumPy
* Scikit Learn

## 2.8 Apportioning of Requirements

There are no other requirements that might be delayed until future versions of the system.

#

# 3. External Interface Requirements

This section is a high level description. Everything in this section shall detail how Osprey interacts with any external interfaces, including, but not limited to, other software and hardware interfaces.

The software is accessed through a Command Line Interface (CLI) for the user to modify the behavior or some of the hardware. It will also allow the user to modify the task parameters. The software should be fully autonomous otherwise.

## 3.1 User Interfaces

The command line interface will be utilized as the user interface for the software. The CLI can be prompted with predefined commands that provide feedback on separate modules.

## 3.2 Hardware Interfaces

Components for the AUV will be connected to the computer via I2C to the Arduino which is connected to the computer using a serial USB connection, or components will be directly connected to the computer via serial USB connections.

## 3.3 Software Interfaces

There are no software interfaces that will be used with this product.

##

## 3.4 Communications Interfaces

There are no communications interfaces that will be used with this product.

#

# 4. Requirements Specification

This section contains all of the necessary software requirements with enough detail to allow designers to accurately design the software to satisfy those requirements, and to allow testers of the software to verify that all requirements have been satisfied.

## 4.1 Functional Requirements

|  |
| --- |
| Requirements Related to:**1. Operating System** |
| Requirement No. | Requirement Description |
| 4.1.1-1 | The Operating System shall be Ubuntu 17.04. |
| 4.1.1-2 | The Operating System shall have ROS Lunar Loggerhead installed. |
| 4.1.1-3 | The Operating System shall receive data from IMU. |
| 4.1.1-4 | The Operating System shall receive data from Cameras. |
| 4.1.1-5 | The Operating System shall receive data from ATmega2560. |
| 4.1.1-6 | The Operating System shall send data to ATmega2560. |
| 4.1.1-7 | The Operating System shall be run on the Computer. |

|  |
| --- |
| Requirements Related to:**2. User Interface** |
| Requirement No. | Requirement Description |
| 4.1.2-1 | The User Interface shall be a CLI. |
| 4.1.2-2 | The User Interface shall be used with ROS. |
| 4.1.2-3 | The User Interface shall provide methods of changing AUV parameters. |
| 4.1.2-4 | The User Interface shall provide AUV testing capabilities. |

|  |
| --- |
| Requirements Related to:**3. AUV Tasks** |
| Requirement No. | Requirement Description |
| 4.1.3-1 | The AUV Tasks shall be to Find the Casino. |
| 4.1.3-2 | The AUV Tasks shall be to Enter the Casino (Gate). |
| 4.1.3-3 | The AUV Tasks may be to Shoot Craps. |
| 4.1.3-4 | The AUV Tasks may be to Buy a Gold Chip. |
| 4.1.3-5 | The AUV Tasks may be to Try the Slots. |
| 4.1.3-6 | The AUV Tasks may be to Hit the Jackpot. |
| 4.1.3-7 | The AUV Tasks may be to Play Roulette. |
| 4.1.3.8 | The AUV Tasks may be to Cash in Your Chips. |

|  |
| --- |
| Requirements Related to:**4. AUV AI** |
| Requirement No. | Requirement Description |
| 4.1.4-1 | The AUV AI shall be able to Find the Casino. |
| 4.1.4-2 | The AUV AI shall be able to Enter the Casino (Gate). |
| 4.1.4-3 | The AUV AI may be able to Shoot Craps. |
| 4.1.4-4 | The AUV AI may be able to Buy a Gold Chip. |
| 4.1.4-5 | The AUV AI may be able to Try the Slots. |
| 4.1.4-6 | The AUV AI may be able to Hit the Jackpot. |
| 4.1.4-7 | The AUV AI may be able to Play Roulette. |
| 4.1.4-8 | The AUV AI may be able to Cash in Your Chips. |

|  |
| --- |
| Requirements Related to:**5. Arduino** |
| Requirement No. | Requirement Description |
| 4.1.5-1 | The Arduino shall receive data from the Sensors. |
| 4.1.5-2 | The Arduino shall send commands to the Pneumatic Control. |
| 4.1.5-3 | The Arduino shall send commands to the Motor Control. |
| 4.1.5-4 | The Arduino shall receive data from the Computer. |
| 4.1.5-5 | The Arduino shall send data from the Computer. |
| 4.1.5-6 | The Arduino shall be connected to ROS. |

|  |
| --- |
| Requirements Related to:**6. Computer Vision** |
| Requirement No. | Requirement Description |
| 4.1.6-1 | The Computer Vision shall receive data from the Cameras. |
| 4.1.6-2 | The Computer vision shall output processed data. |
| 4.1.6-3 | The Computer Vision shall provide coordinates |
| 4.1.6-4 | The Computer Vision shall detect objects associated with tasks |
| 4.1.6-5 | The Computer Vision shall detect tasks |
| 4.1.6-6 | The Computer Vision shall shall provide depth perception  |
| 4.1.6.7 | The Computer Vision shall use classifiers for object detection |

|  |
| --- |
| Requirements Related to:**7. Sensors** |
| Requirement No. | Requirement Description |
| 4.1.7-1 | The Sensor data shall be published to ROS. |

|  |
| --- |
| Requirements Related to:**8. Pneumatic Control** |
| Requirement No. | Requirement Description |
| 4.1.8-1 | Pneumatic Control shall be capable of receiving commands from ROS. |

|  |
| --- |
| Requirements Related to:**9. Motor Control** |
| Requirement No. | Requirement Description |
| 4.1.9-1 | Motor Control shall receive commands from the Arduino. |
| 4.1.9-2 | Motor Control shall operate at command level. |
| 4.1.9-3 | Motor Control shall allow the AUV full horizontal and vertical movement. |

## 4.2 External Interface Requirements

There are no external interfaces that will be used with this product.

## 4.3 Logical Database Requirements

There are no logical databases that will be used with this product

## 4.4 Design Constraints

Regulatory Policies

* Types of tasks that will be in the competition
	+ See referenced Robosub Draft Tasks 2018
* Rules and Limitations to how we can accomplish each task
	+ See robonation website for rules

Hardware Limitation

* Available Sensors
	+ Camera
		- Only forward and downward facing cameras
	+ Barometer
	+ IMU
	+ Hydrophone

Environmental Conditions

* Underwater Submersion
	+ Camera images will contain extra noise.
* Lighting Conditions
	+ Lighting will vary.

Memory Constraints

* Microcontroller ATmega2560 has a limited amount of memory, which will prevent it from having too many global variables.

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

|  |  |
| --- | --- |
| Requirement No. | Requirement Description |
| 5.1-1 | The AUV components shall operate at 100Hz |
| 5.1-2 |  |

##

## 5.2 Safety Requirements

|  |  |
| --- | --- |
| Requirement No. | Requirement Description |
| 5.2-1 | AUV shall have a software kill switch for motors. |
| 5.2-2 |  |

## 5.3 Security Requirements

There are no security requirements that will be used with this product

## 5.4 Software Quality Attributes

|  |  |
| --- | --- |
| Requirement No. | Requirement Description |
| 5.3-1 | Osprey shall be modular where possible. |
| 5.3-2 | Osprey shall be adaptable for future competition |

## 5.5 Business Rules

There are no software interfaces that will be used with this product.

# 6. Other Requirements

There are no other requirements that need to be listed for this software product.

# Appendix A: Glossary

**Abbreviations and Acronyms**

* AUV - Autonomous Underwater Vehicle
* CLI - Command Line Interface
* CV - Computer Vision
* GUI - Graphical User Interface
* Hz - Hertz
* IMU - Inertial Measurement Unit
* ROS - Robot Operating System
* SRS - Software Requirements Standard
* USB - Universal Serial Bus

**Definitions**

* Arduino - Provides hardware interfaces. See <https://www.arduino.cc/en/Guide/Introduction>
* Barometer - Provides depth of AUV in meters.
* Computer Vision - Image processing to detect objects. See <https://en.wikipedia.org/wiki/Computer_vision>.
* Hertz - frequency per a millisecond.
* Hydrophone - Microphone designed to be used underwater.
* I2C - I2C is a serial protocol for two-wire interface to connect low-speed devices like microcontrollers, EEPROMs, A/D and D/A converters, I/O interfaces and other similar peripherals in embedded systems. See <http://i2c.info/> for more information.
* Inertial Measurement Unit - Provides AUV with leveling and direction information. See <https://www.vectornav.com/products/vn-100> for the AUV’s IMU.
* Module - An independent and interchangeable part of the program that serves a specific function.
* Node - Module with ROS publisher and/or subscribers attached
* Pneumatics - Gas powered systems. See <https://en.wikipedia.org/wiki/Pneumatics>.

# Appendix B: Analysis Models



* Squares - Hardware
* Circles - ROS Node
* Diamond - ROS