Software Requirements Specification

for

Autonomous Underwater Vehicle

Version 1.0 approved

Prepared by Ricardo Medina, Heriberto Gonzalez

California State University Los Angeles, Engineering Department

12/09/2020

**Table of Contents**

Table of Contents................................................................................................................. pg 2

Revision History................................................................................................................... pg 3

 1. Introduction................................................................................................................ pg 4

 1.1. Purpose........................................................................................................... pg 4

 1.2. Intended Audience and Reading Suggestions................................................ pg 4

 1.3. Product Scope................................................................................................ pg 4

 1.4. Definitions, Acronyms, and Abbreviations .................................................. pg 4

 1.5. References...................................................................................................... pg 4

 2. Overall Description.................................................................................................... pg 6

 2.1. System Analysis…......................................................................................... pg 6

 2.2. Product Perspective........................................................................................... pg 6

 2.3. Product Functions........................................................................................... pg 6

 2.4. User Classes and Characteristics.................................................................... pg 6

 2.5. Operating Environment.................................................................................. pg 7

 2.6. Design and Implementation Constraints........................................................ pg 7

 2.7. User Documentation...................................................................................... pg 7

 2.8. Assumptions and Dependencies.................................................................... pg 7

 2.9. Apportioning of Requirements...................................................................... pg 7

 3. External Interface Requirements............................................................................... pg 9

 3.1. User Interfaces............................................................................................... pg 9

 3.2. Hardware Interfaces....................................................................................... pg 9

 3.3. Software Interfaces........................................................................................ pg 9

 3.4. Communications Interfaces........................................................................... pg 9

 4. Requirements Specification....................................................................................... pg 10

 4.1. Functional Requirements............................................................................... pg 10

 4.2. External Interface Requirements................................................................... pg 10

 4.3. Logical Database Requirements.................................................................... pg 10

 4.4. Design Constraints......................................................................................... pg 10

 5. Other Nonfunctional Requirements........................................................................... pg 11

 5.1. Performance Requirements............................................................................ pg 11

 5.2. Safety Requirements...................................................................................... pg 11

 5.3. Security Requirements................................................................................... pg 11

 5.4. Software Quality Attributes........................................................................... pg 11

 5.5. Business Rules............................................................................................... pg 11

 6. Legal and Ethical Considerations.….......................................................................... pg 12

Appendix A: Glossary........................................................................................................ pg 12

Appendix B: Analysis Models........................................................................................... pg 12

Appendix C: To Be Determined List................................................................................. pg 12

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason For Changes | Version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**1. Introduction**.

This software is intended to be implemented in the Autonomous Underwater Vehicle (AUV) under construction by the California State University Senior Design team under the Engineering Department. The goal of the AUV is to compete in an international competition hosted by RoboNation to perform tasks in an underwater environment. These include movement, navigation, object detection and recognition, manipulation of objects, target elimination.

**1.1 Purpose**

The purpose of this document is to give a general overview of all requirements necessary for the successful operation of an Autonomous Underwater Vehicle for the express purpose of competing in the ROBONATION RoboSub Competition. This project is composed of various submodules, each with their own dedicated software requirements. This Document will explain how these submodules are required to interact with each other.

**1.2 Intended Audience and Reading Suggestions**

Due to circumstances outside our control, the physical AUV will not be built. As the AUV project is ongoing and evolving, this document is intended for fellow developers taking over the project in future years. Suggested readings include

* + ROBOSUB Controls Software Requirements Document
	+ ROBOSUB Mission Planning Software Requirements Document
	+ ROBOSUB Navigation Software Requirements Document
	+ ROBOSUB Computer Vision Software Requirements Document

**1.3 Product Scope**

The product will consist of four distinct ROS packages. Controls, Navigation, Computer Vision, and Mission Planning. These packages are intended to be installed on a single platform and execute concurrently to accomplish tasks set forth by the Robosub Competition. It will be responsible in accomplishing tasks yet to be released such as movement, object detection, navigation, object manipulation.

**1.4 Definitions, Acronyms, and Abbreviations**

* AUV – Autonomous Underwater Vehicle
* ROS – Robotic Operating System

**1.5 References**

* ROBOSUB Controls Software Requirements Document
* ROBOSUB Mission Planning Software Requirements Document
* ROBOSUB Navigation Software Requirements Document
* ROBOSUB Computer Vision Software Requirements Document
* 2018 RoboSub\_2018 Mission and Scoring\_v01.50
* https://robonation.org/programs/robosub/

**2. Overall Description**

The annual RoboSub competition requires the development of an Autonomous Underwater Vehicle. Each year the competition releases a set of tasks, like previous years, to demonstrate the autonomous capabilities of the vehicle. Some examples of tasks include, Target recognition and projectile deployment, Gate recognition and navigation through gate, Pinger detection, manipulation of object from location to location. For further description of tasks please refer to the 2018 RoboSub Mission and Scoring Document.

**2.1 System Analysis**

The overall goal of the RoboSub Software is to develop a software that can Autonomously control and navigate the AUV throughout the competition while performing various task using its various sensors and actuated systems.

**2.2 Product Perspective**

This document describes the overall application of the AUV software and their interactions. It is similar to other implementations of UAV systems in other platforms. This version was specifically designed to run on the California State University AUV. Each subsystem component interacts with the others through the Publisher Subscribers model implemented by ROS. Each subsystem composes of ROS nodes, independently running scripts, that communicate with each other over the ROS publisher/subscriber system. ROS is also responsible for the startup of all nodes of all subsystems. For descriptions of the subsystems and their responsibilities please refer to the respective Requirements Document.

**2.3 Product Functions**

* Object detection and localization
* Movement and navigation
* Manipulation of task objects

Further details of software functions can be found in the subsystem's documentation.

**2.4 User Classes and Characteristics**

The principal user class for this software is the lead programmer responsible for implementation of all software packages for the California State University AUV. Said user must consider the hardware and software requirements detailed in this document before implementation.

The secondary users are software testers. They are responsible for testing and maintaining code, with regards to sensor accuracy, and autonomous navigational corrections. These corrections must be only implemented in this package, without changing the interfaces to other packages unless so specified by both package leads.

**2.5 Operating Environment**

This software package is broken down into submodules. The ROS software package allows for multiple nodes to run concurrently and communicate. It also allows for software communication through connected hardware platforms.

Jetson Tx2

Python2

Linux 18.0.14

Ros Melodic

Arduino Uno

Arduino Mega

**2.6 Design and Implementation Constraints**

Due to the ROS version on the platform, ROS nodes are restricted to be written in one of the following languages:

* Python 2.7
* C++

Any Code to be run on an Arduino must be written in the Arduino language.

Main safety requirements are to not damage any components of the AUV as well as any item involved in the execution of competition tasks. Additionally, we must not cause any harm to the divers observing the competition in the water.

**2.7 User Documentation**

* ROBOSUB Controls Software Requirements Document
* ROBOSUB Mission Planning Software Requirements Document
* ROBOSUB Navigation Software Requirements Document
* ROBOSUB Computer Vision Software Requirements Document

**2.8 Assumptions and Dependencies**

Nonapplicable at this moment.

**2.9 Apportioning of Requirements**

Nonapplicable at this moment.

**3. External Interface Requirements**

**3.1 User Interfaces**

As requirement for implementation, no GIU is required. However, a GUI with capabilities to show status of subsystems, as well as logs would be beneficial to troubleshooting and testing. Additionally, an interface with the ability to select options for run conditions would be an asset. The competition takes part in an arena with four distinct quadrants and the capability to switch between the quadrants in code would be beneficial to the operation of the AUV.

**3.2 Hardware Interfaces**

As a Robotic platform there are various interfaces that need to be controlled. All sensors connected to the motherboard require serial connections. The thrusters are controlled by PWM signals. Additionally, there are expected to be 2 further actuated systems still under development. These will be added as soon as designs are finalized and will be addressed in future updates. For more detailed information on the interfaces please refer to the ROBOSUB\_Controls file or the ROBOSUB\_Navigation file.

**3.3 Software Interfaces**

In general this software will be interfacing with ROS Melodic running on a Linux18.04 system. The API website is located at

http://wiki.ros.org/APIs

**3.4 Communications Interfaces**

For testing and calibration purposes the platform will be accessed over SSH on a Tether with the following specifications: <insert Specifications>

**4. Requirements Specification**

**4.1 Functional Requirements**

The software is to be implemented on an AUV designed by California State University Los Angeles. This software has been divided into 4 subsystem packages. Each package contains their own Functional Requirements. Below are requirements for the competition as generalities. Please refer to the SRD of each subsystem for detailed information.

1. The software shall provide an Autonomous system to accomplish tasks set forth by the ROBOSUB competition
2. The software shall provide a controls system for AUV maneuvering
3. The software shall implement object recognition and tracking
4. The software shall interface with onboard sensors
5. The software shall navigate between tasks in a large underwater environment
6. The software shall monitor subsystems and check for possible errors

**4.2 External Interface Requirements**

This project makes use of the ROS publisher subscriber model. Nodes can create a publisher (data provider) which can publish data to a topic to be subscribed to by any other ROS node. As such data does not have to have a direct recipient. Additionally, custom message types can be created to send through these publishers. This system must be designed and agreed upon by all systems. Topics must include a name, and a data type. These topics need to be designed, so no hard interface requirements exist at this time.

**4.3 Logical Database Requirements**

Nonapplicable at this moment.

**4.4 Design Constraints**

The design of the AUV software is specialized by the components that are being used this includes any software but also the hardware that help the AUV perform the required tasks.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

Performance of this software is based on the results of the performance of the AUV within the competition.

**5.2 Safety Requirements**

Safety requirements present in the AUV software are critical to its performance and safety. The AUV must be always supervised when operating and in use. Before operating the AUV safety checks must be done to assure that all connections and water sealing have been properly connected. Failure to do so can lead to serious damage of AUV and all components with the haul.

**5.3 Security Requirements**

Security requirements for the AUV software are present on the AUVs GitHub repository which holds all software and code that runs the AUV. This repository should be private and only those in the software team must be added and given access to editing and uploading updated code or software.

**5.4 Software Quality Attributes**

Nonapplicable at this moment.

**5.5 Business Rules**

Nonapplicable at this moment.

**6. Legal and Ethical Considerations**

Nonapplicable at this moment.