Software Requirements Specification

for

RoboSub Navigation Subsystem

Version 1.0 approved

Prepared by

Ricardo Medina

Kevin Williams

California State University Department of Engineering

December 8, 2020

**Table of Contents**

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

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

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

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

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

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

1.4. Definitions, Acronyms, and Abbreviations ..........................................................5

1.5. References.............................................................................................................5

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

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

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

2.3. Product Functions.................................................................................................7

2.4. User Classes and Characteristics..........................................................................8

2.5. Operating Environment........................................................................................8

2.6. Design and Implementation Constraints..............................................................8

2.7. User Documentation............................................................................................9

2.8. Assumptions and Dependencies..........................................................................9

2.9. Apportioning of Requirements............................................................................9

3. External Interface Requirements..............................................................................10

3.1. User Interfaces...................................................................................................10

3.2. Hardware Interfaces...........................................................................................10

3.3. Software Interfaces............................................................................................10

3.4. Communications Interfaces...............................................................................10

4. Requirements Specification......................................................................................11

4.1. Functional Requirements...................................................................................11

4.2. External Interface Requirements.......................................................................11

4.3. Logical Database Requirements........................................................................12

4.4. Design Constraints............................................................................................12

5. Other Nonfunctional Requirements.........................................................................13

5.1. Performance Requirements...............................................................................13

5.2. Safety Requirements.........................................................................................13

5.3. Security Requirements......................................................................................13

5.4. Software Quality Attributes..............................................................................13

5.5. Business Rules..................................................................................................13

6. Legal and Ethical Considerations.….......................................................................14

Appendix A: Glossary.....................................................................................................15

Appendix B: Analysis Models........................................................................................15

Appendix C: To Be Determined List..............................................................................15

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason For Changes | Version |
| Ricardo Medina, Kevin Williams | 12/10/2020 | Delivery of the Software Requirements Document | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**1. Introduction**

The purpose of this software is to connect to all onboard sensors on the California State University AUV, gather data from each sensor and send it to the various subsystems, as well as providing a simple navigation system for the purpose of getting the AUV from task to task in the ROBONATION Robosub Competition.

**1.1 Purpose**

The purpose of this document is to determine all requirements and capabilities of onboard sensors on the ROBOSUB AUV Project. It details the sensors and how they must be implemented on hardware as well as the software required to interface with said sensors. It also details the overall requirements on a simple navigation system responsible for maneuvering the AUV from task to task.

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

**1.3 Product Scope**

* IMU ROS package
  + Shall interface with the VECTORNAV IMU and record/send pitch, yaw, roll data to the AUV
* DLV ROS package
  + Shall interface with the DVL and record/send north-south, east-west translational data to the AUV
* Barometer ROS package
  + Shall interface with the Barometer and record/send depth data to the AUV
* Sonar ROS package
  + Shall interface with the Sonar and record/send object detection data to the AUV
* Hydrophone ROS package
  + Shall interface with the Hydrophones and record/send Pinger location data to the AUV
* Navigation Ros package
  + Shall use sensor data, and provide output to controls to navigate the AUV to its next intended task

**1.4 Definitions, Acronyms, and Abbreviations**

DVL – Doppler Velocity Log

AUV – Autonomous Underwater Vehicle

ROS – Robotic Operating System

IMU – Inertial Measuring Unit

**1.5 References**

* Sonar

Bluerobotics/ping-arduino, kklemens, Dec 5, 2019, <https://github.com/bluerobotics/ping-arduino#usage>

* Pathfinder DVL

Pathfinder Doppler Velocity Log (DVL) 600kHz manual, 2017 Teledyne RD Instruments, <https://www.manualslib.com/manual/1447288/Teledyne-Pathfinder.html?page=4#manual>

* Hydrophone

AS-1 Aquarian Scientific Broadband Measurement Hydrophone manual, 2013 Aquarian Scientific, <https://www.aquarianaudio.com/as-1-hydrophone.html> , <https://www.aquarianaudio.com/AqAudDocs/AS-1_manual.pdf>

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

**2. Overall Description**

The Software is intended for use in the AUV under construction by the California State University Engineering department. As such our hardware requirements are handed down by this build. For our part of the software, we are required to run on a Jetson TX2 motherboard and an Arduino Uno. The sensors we are responsible for are the DVL, IMU, SONAR, Barometer and Hydrophones. We are also task in maneuvering the sub from task to task, using data from the sensors as well as object recognition from the Computer vision module. Other restrictions include:

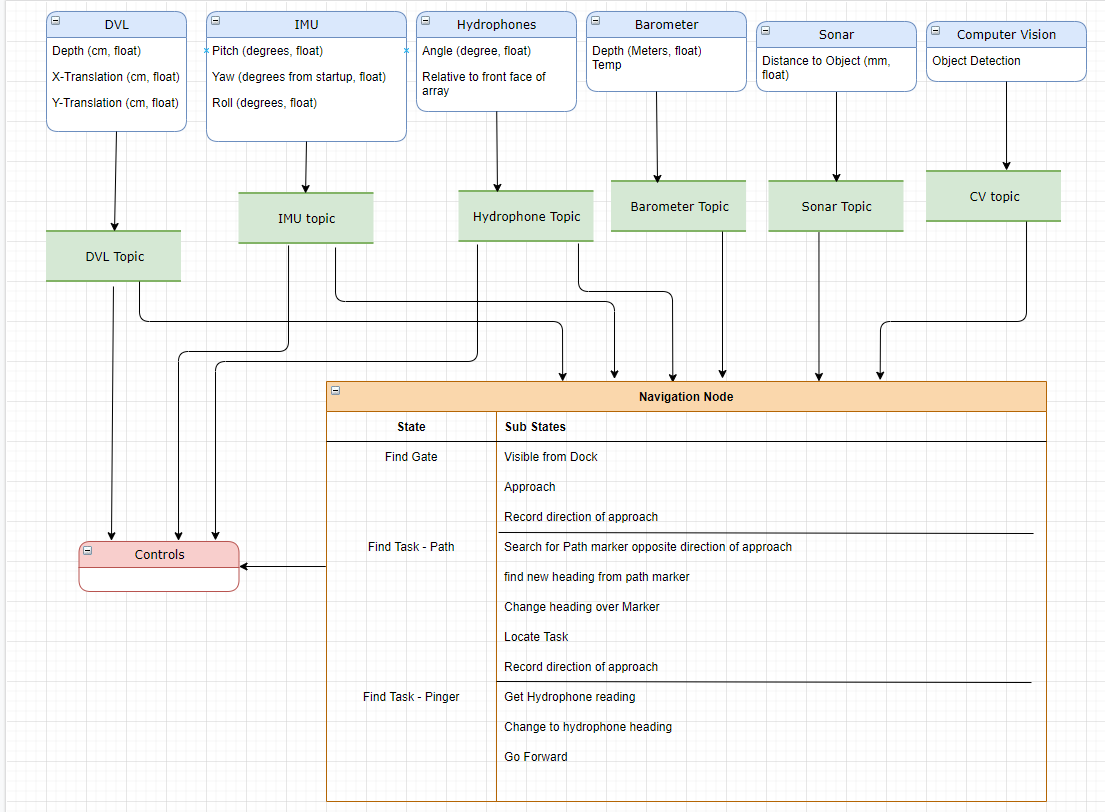
* + Barometer must be connected to an Arduino using I2C connection
  + SONAR, DVL, IMU, Hydrophones must use a serial connection on the TX2 motherboard

**2.1 System Analysis**

1. The RoboSub navigation team is developing a “simple” Navigation function to guide the AUV to and from tasks as well as correct its path if it goes astray.
2. Major hurdles include tracing libraries and source software for sensors, as well as integrating libraries with the required ROS system for data transfer. In addition, this work will have to be done with limited testing time with the actual instruments.
3. After tracing libraries, they will be included in the software package, reducing the need for further searches unless for the purposes of updating libraries. ROS integration should be accomplished by extending the library to add the additional functions needed such as ROS publishers and Subscribers.

**2.2 Product Perspective**

This software is to be used in conjunction with the following software packages also in development e.g. controls, mission planning and computer vision. This will provide data to the other packages, as well as interface with Mission Planning package to automate movement. The package will include standalone programs intended for testing purposes. Preexisting software does not come in a prepackaged form, as such we create this package to simplify implementation of sensors and navigation for the AUV platform in development. The software implements prebuilt libraries for each sensor, as well as adding the required ROS interfaces. Fig 1. shows the proposed interfacing with ROS.



**2.3 Product Functions**

The Navigation subsystem will be responsible for the following

* Obtaining gyro coordinates
* Obtaining angle of Pinger relative to front of AUV
* Obtaining object distance via Sonar
* Obtaining depth measurements
* Obtaining pressure readings
* Directing AUV to correct path
* Passing useful sensor information to Controls subsystem

**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 has 2 hardware requirements

DVL, SONAR, Hydrophones, and IMU shall run as ROS nodes on an Nvidia TX2 motherboard running Ubuntu 18.04, ROS Melodic, Python 2.7

Barometer and SONAR shall run as a ROS node on an Arduino mega, with a serial connection to the TX2 motherboard.

**2.6 Design and Implementation Constraints**

Arduino Mega and Uno

* Limited to Arduino Language
* Limit of 40mA on each pin
* Total limit of 200mA for all pins
* Recommended input of 7~12V
* Maximum Input of 6~20V
* Total output of 5.5V
* Current outputs USB connection 500mA - 1A

Blue Robotics Sonar

* Must be grounded
* Input of 5.5V
* Uses Arduino library or Python 3 with raspberri pi
* Serial TTL to usb required for computer use
* Output display on Arduino Serial Monitor or Blue Robotics PingViewer application

**2.7 User Documentation**

* Pathfinder DVL user manual – pdf
* AS-1 Hydrophone user manual –pdf
* Blue Robotics Ping Sonar online guide – website: <https://github.com/bluerobotics/ping-arduino#usage>
* ROBOSUB Overview Software Requirements Document
* ROBOSUB Mission Planning Software Requirements Document
* ROBOSUB Controls Software Requirements Document
* ROBOSUB Computer Vision Software Requirements Document

**2.8 Assumptions and Dependencies**

The navigation ROS node was built using ROS Melodic Morenia, ROS version 1.14.X, which only natively supports python 2.7. Future ROS builds, such as the most current ROS operating system named, ROS Noetic, version 1.15.X, fully supports python 3.0. A future user updating the ROS package will need to choose to branch Melodic and the new ROS package together which can add risks for regressions. The user will also have to find updated ROS dependencies to support python 3, which can mean finding replacements for ROS dependency keys (rosdep key) and support for dependent ROS packages as well as changing the source code to support python 3.

**2.9 Apportioning of Requirements**

Not applicable at this time

**3. External Interface Requirements**

**3.1 User Interfaces**

No user interface for AUV.

**3.2 Hardware Interfaces**

All sensors shall be connected to the motherboard via serial connections.

**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**

The Bar30 Pressure sensor and Ping Sonar will be communicating with the Arduino Mega via serial connections while the Vector Nav-IMU as well as the DVL and Hydrophones will be connected to the motherboard via serial connections.

**4. Requirements Specification**

**4.1 Functional Requirements**

1. The software shall create ROS publisher topics for each sensor on the platform
   1. Each sensor shall have a predetermined custom data type to share information between submodules
2. The software shall provide object detection and location from the Blu Robotics Ping Sonar
3. The software shall provide the current yaw, pitch, and roll measurements from the IMU
   1. The software should provide yaw in relation to magnetic north
4. The software shall provide the translational movement of the AUV in both the north-south and east-west direction using DLV data
5. The software shall provide the depth measurement reported by the Barometer.
6. The software shall provide the angle relative to the front face of the AUV of an identified underwater Pinger
7. The software shall use sensor data to navigate from point a to point b
   1. The software should create a map of environmental features to be used for future runs
   2. The software shall interface with the Mission Planning subsystem to coordinate movement between the expected tasks.
   3. The software shall keep track of all AUV movement when undergoing a mission.
   4. The software shall communicate with the Computer Vision subsystem to confirm detection of mission tasks.
8. The software may contain a GUI used for testing and troubleshooting onboard sensors

**4.2 External Interface Requirements**

Implementation of the Navigations system in conjunction with mission planning is still underway. Below is the proposed interface requirements between Navigation and the AUV subsystems. As we are using the ROS publisher/subscriber interfaces, below are the topics created that subsystems will subscribe to.

* DVL
  + Translational movement in the north-south, east-west directions
  + Range, accuracy tolerance TBD
  + Units of measure mm
  + ROS Data Package
    - North-south – float
    - East-west – float
* IMU
  + Yaw, pitch, roll of AUV
  + Range, accuracy tolerance TBD
  + Units of measure Degrees
  + ROS Data Package
    - Yaw float
    - Pitch float
    - Roll float
* Barometer
  + Depth of AUV
  + Range, accuracy tolerance TBD
  + Units of measure mm
  + ROS Data Package
    - Depth – float
* Hydrophones
  + Location of underwater Pinger
  + Range, accuracy tolerance TBD
  + Units of measure degrees
  + ROS Data Package
    - Angle to pinger

**4.3 Logical Database Requirements**

Not applicable for Navigation

**4.4 Design Constraints**

The ROS Navigation nodes shall be written in python 2.7 as ROS Melodic (ROS version 1.14.X) does not support python 3. An update from ROS Melodic to ROS noetic will cause the ROS dependencies and libraries to be updated causing regressions

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

The Blue Robotics Ping Sonar connected with the Arduino Mega (Mega 2560) operates at 115,200 baud with 4 connections to the Arduino, power, ground, RX, and TX.

DVL and IMU connect to motherboard using a serial connection at 115,200 baud rate over USB

Barometer interfaces with an IC2 connection on Arduino

**5.2 Safety Requirements**

Please refer to ROBOSUB\_Overview\_Software\_Requirement\_Document.docx

**5.3 Security Requirements**

Please refer to ROBOSUB\_Overview\_Software\_Requirement\_Document.docx

**ROBOSUB\_Overview\_Software\_Requirement\_Document.docx**

**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.

**Appendix A: Glossary**

DVL – Doppler Velocity Log

AUV – Autonomous Underwater Vehicle

ROS – Robotic Operating System

IMU – Inertial Measuring Unit

Baud - A baud is a unit of transmission speed equal to the number of times a signal changes state per second which means that the baud rate is the number bits per second.

Melodic – ROS version 1.14.X

Noetic – ROS version 1.15.X

**Appendix B: Analysis Models**

**Appendix C: To Be Determined List**