SOftware requirements specification

for

<Project Name>

Version 1.0

Prepared by <Author>

PMEL/EDD

<Date Created>

Table of Contents

[REVISION HISTORY 3](#_Toc453666794)

[1 Introduction 4](#_Toc453666795)

[1.1 Purpose 4](#_Toc453666796)

[1.2 Document Conventions 4](#_Toc453666797)

[1.3 Intended Audience and Reading Suggestions 4](#_Toc453666798)

[1.4 Produce Scope 4](#_Toc453666799)

[1.5 References 4](#_Toc453666800)

[2 Product description 6](#_Toc453666801)

[2.1 Product Perspective 6](#_Toc453666802)

[2.2 Product Functions 6](#_Toc453666803)

[2.3 User Classes and Characteristics 6](#_Toc453666804)

[2.4 Operating Environment 6](#_Toc453666805)

[2.5 Design and Implementation Constraints 6](#_Toc453666806)

[2.6 User Documentation 7](#_Toc453666807)

[2.7 Assumptions and Dependencies 7](#_Toc453666808)

[3 External interface Requirements 8](#_Toc453666809)

[3.1 User Interfaces 8](#_Toc453666810)

[3.2 Hardware Interfaces 8](#_Toc453666811)

[3.3 Software Interfaces 8](#_Toc453666812)

[3.4 Communications Interfaces 8](#_Toc453666813)

[4 System Features 10](#_Toc453666814)

[4.1 System Feature 1 10](#_Toc453666815)

[4.1.1 Description and Priority 10](#_Toc453666816)

[4.1.2 Stimulus/Response Sequences 10](#_Toc453666817)

[4.1.3 Functional Requirements 10](#_Toc453666818)

[4.2 System Feature 2 (and so on) 10](#_Toc453666819)

[5 Other NonFunctional Requirements 12](#_Toc453666820)

[5.1 Performance Requirements 12](#_Toc453666821)

[5.2 Safety Requirements 12](#_Toc453666822)

[5.3 Security Requirements 12](#_Toc453666823)

[5.4 Software Quality Attributes 12](#_Toc453666824)

[5.5 Business Rules 12](#_Toc453666825)

[6 Other Requirements 14](#_Toc453666826)

[Appendix A: Glossary 15](#_Toc453666827)

[Appendix B: Analysis Models 16](#_Toc453666828)

[Appendix C: To Be Determined List 17](#_Toc453666829)

# REVISION HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| **NAME** | **DATE** | **CHANGES** | **VERSION** |
| M.Casari | 6/14/2016 | First draft | 1.0 |

# Introduction

## Purpose

*<Identify the product whose system/software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>*

## Document Conventions

*<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>*

## Intended Audience and Reading Suggestions

## Produce Scope

## References

*<List any other documents or Web address to which this SRS refers. These may include interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date and source or location.>*

# Product description

## Product Perspective

*<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirement of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>*

## Product Functions

*<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>*

## User Classes and Characteristics

*<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>*

## Operating Environment

The CID system will be an embedded system operating a standalone PCB. The OceanServer OS-4000T will be located on the PCB, which will in turn be installed in the Gill Wind Sensor housing that the current Sparton sensor is located.

*<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>*

## Design and Implementation Constraints

*<Describe any items or issues that will limit the options available to the developers. These might include: NOAA, DOC or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software.>*

## User Documentation

*<List the user documentation component (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>*

## Assumptions and Dependencies

# External interface Requirements

**PRIORITY DEFINITIONS**

* Priority 1 – The requirements is a “must have” as outlined by policy.
* Priority 2 – The requirement is need for improved processing, and the fulfillment of the requirement will create immediate benefits.
* Priority 3 – The requirement is “nice to have” which may include new functionality.

## User Interfaces

*<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>*

## Hardware Interfaces

The hardware will conform to the same

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>*

## Software Interfaces

*< Describe the connection between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shard across software components. If the data sharing mechanism must be implement in a specific way (for example, use of a global data area in multitasking operating system), specify this as an implementation constraint>.*

## Communications Interfaces

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocol, electronics forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>*

# System Features

*<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>*

## System Feature 1

*<Don’t really say “System Feature 1.” State the feature name in just a few words.>*

### Description and Priority

*<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>*

### Stimulus/Response Sequences

*<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>*

### Functional Requirements

*<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to the anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>*

*<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>*

REQ-1:  
REQ-2:

## System Feature 2 (and so on)

# Other NonFunctional Requirements

## Performance Requirements

*< If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify timing relationships for real-time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>*

## Safety Requirements

The power rail for the OceanServer OS-4000T will be fused to prevent unintentional shorting. PTC fuse is acceptable.

## Security Requirements

None.

## Software Quality Attributes

*<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>*

# Other Requirements

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

# Appendix A: Glossary

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>*

# Appendix B: Analysis Models

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>*

# Appendix C: To Be Determined List

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*