Department of Computer Science and Engineering  
The University of Texas at Arlington

Team: Survivors

Project: Sight By Touch

Team Members:   
*Gerardo Guevara*

*Kevin Tran*

*Victor Rodriguez*

*Margaret Floeter*

*Henry Loh*

Last Updated: 4/1/2014

# Table of Contents

[Table of Contents 2](#_Toc384151714)

[Document Revision History 7](#_Toc384151715)

[List of Figures 8](#_Toc384151716)

[List of Tables 9](#_Toc384151717)

[1. Introduction 11](#_Toc384151718)

[1.1 Document Overview 11](#_Toc384151719)

[1.2 Product Overview 11](#_Toc384151720)

[1.3 Product Scope 14](#_Toc384151721)

[1.4 Acronym Definitions 15](#_Toc384151722)

[2. References 17](#_Toc384151723)

[2.1 Overview 17](#_Toc384151724)

[2.2 System Requirements Specification 17](#_Toc384151725)

[2.2.1 Customer Requirements 18](#_Toc384151726)

[2.2.2 Packaging Requirements 19](#_Toc384151727)

[2.2.3 Performance Requirements 20](#_Toc384151728)

[2.2.4 Safety Requirements 21](#_Toc384151729)

[2.2.5 Maintenance and Support Requirements 22](#_Toc384151730)

[2.2.6 Other Requirements 23](#_Toc384151731)

[2.3 Architecture Design Specification 24](#_Toc384151732)

[2.3.1 Layer Overview 24](#_Toc384151733)

[2.3.2 Subsystem Overview 25](#_Toc384151734)

[2.3.3 Requirements Mapping By Subsystem 27](#_Toc384151735)

[2.4 Detailed Design Specification 28](#_Toc384151736)

[2.4.1 Module Overview 29](#_Toc384151737)

[2.4.2 Requirements Traceability Matrix By Module 31](#_Toc384151738)

[3. Test Items 33](#_Toc384151739)

[3.1 Overview 33](#_Toc384151740)

[3.2 Relational Diagram 34](#_Toc384151741)

[3.3 Hardware Testing Phase 35](#_Toc384151742)

[3.3.1 Description 35](#_Toc384151743)

[3.3.2 Hardware Tests 35](#_Toc384151744)

[3.4 Unit Testing Phase 37](#_Toc384151745)

[3.4.1 Description 37](#_Toc384151746)

[3.4.2 System Control Module Unit Tests 37](#_Toc384151747)

[3.4.3 Sensor Module(s) Unit Tests 38](#_Toc384151748)

[3.4.4 Detection Filter Unit Tests 39](#_Toc384151749)

[3.4.5 Battery Level Monitor Unit Tests 40](#_Toc384151750)

[3.4.6 On Notification Unit Tests 40](#_Toc384151751)

[3.4.7 Object Notification Unit Tests 41](#_Toc384151752)

[3.4.8 Battery Notification Unit Tests 41](#_Toc384151753)

[3.5 Component Testing Phase 42](#_Toc384151754)

[3.5.1 Description 42](#_Toc384151755)

[3.5.2 Hardware Interface Layer Component Tests 42](#_Toc384151756)

[3.5.3 Warning Layer Component Tests 43](#_Toc384151757)

[3.5.4 Notification Layer Component Tests 44](#_Toc384151758)

[3.6 System Verification Phase 45](#_Toc384151759)

[3.6.1 Description 45](#_Toc384151760)

[3.6.2 System Verification Tests 45](#_Toc384151761)

[3.7 System Validation Phase 46](#_Toc384151762)

[3.7.1 Description 46](#_Toc384151763)

[3.7.2 System Validation Tests 46](#_Toc384151764)

[4. Risks 48](#_Toc384151765)

[4.1 Overview 48](#_Toc384151766)

[4.2 Risks Table 48](#_Toc384151767)

[5. Features To Be Tested 50](#_Toc384151768)

[5.1 Overview 50](#_Toc384151769)

[5.2 Customer Requirements 50](#_Toc384151770)

[5.2.1 On and Off 50](#_Toc384151771)

[5.2.2 Detect Obstructions 50](#_Toc384151772)

[5.2.3 Obstruction Notification 51](#_Toc384151773)

[5.2.4 Battery Notification 51](#_Toc384151774)

[5.2.6 Low Battery Notification 51](#_Toc384151775)

[5.2.7 User Friendly 51](#_Toc384151776)

[5.3 Packaging Requirements 51](#_Toc384151777)

[5.3.1 System Control Device 51](#_Toc384151778)

[5.3.2 Sight By Touch System 52](#_Toc384151779)

[5.3.3 Battery 52](#_Toc384151780)

[5.3.4 Charger 52](#_Toc384151781)

[5.3.5 Software 52](#_Toc384151782)

[5.4 Performance Requirements 52](#_Toc384151783)

[5.4.1 Switching Range Latency 52](#_Toc384151784)

[5.4.2 Real-Time Performance 53](#_Toc384151785)

[5.4.3 Detection Quality 53](#_Toc384151786)

[5.4.4 Vibration Intensity 53](#_Toc384151787)

[5.4.5 Boot Up Time 53](#_Toc384151788)

[5.5 Maintenance and Support Requirements 53](#_Toc384151789)

[5.5.1 Troubleshooting 53](#_Toc384151790)

[5.6 Other Requirements 54](#_Toc384151791)

[5.6.1 Adjustable Range 54](#_Toc384151792)

[6. Features Not To Be Tested 55](#_Toc384151793)

[6.1 Overview 55](#_Toc384151794)

[6.2 Customer Requirements 55](#_Toc384151795)

[6.2.1 Maintainable 55](#_Toc384151796)

[6.2.2 Floor Depth Detection 55](#_Toc384151797)

[6.2.3 Inconspicuous Appearance 55](#_Toc384151798)

[6.3 Packaging Requirements 55](#_Toc384151799)

[6.3.1 User Manual 55](#_Toc384151800)

[6.3.2 Cardboard Container 55](#_Toc384151801)

[6.4 Performance Requirements 56](#_Toc384151802)

[6.4.1 Battery Life 56](#_Toc384151803)

[6.5 Safety Requirements 56](#_Toc384151804)

[6.6 Maintenance and Support Requirements 56](#_Toc384151805)

[6.6.1 Future Upgrades and Maintenance 56](#_Toc384151806)

[6.6.2 Hardware Maintenance 56](#_Toc384151807)

[6.6.3 Software Maintenance 56](#_Toc384151808)

[6.7 Other Requirements 56](#_Toc384151809)

[6.7.1 Weight 56](#_Toc384151810)

[6.7.2 Readability 56](#_Toc384151811)

[7. Overall Test Strategy 57](#_Toc384151812)

[7.1 Overview 57](#_Toc384151813)

[7.2 Strategy 57](#_Toc384151814)

[7.3 Metrics 58](#_Toc384151815)

[7.4 Regression 59](#_Toc384151816)

[8. Acceptance Criteria 60](#_Toc384151817)

[8.1 Overview 60](#_Toc384151818)

[8.2 Hardware Testing 60](#_Toc384151819)

[8.3 Unit Testing 60](#_Toc384151820)

[8.4 Component Testing 60](#_Toc384151821)

[8.5 System Verification Testing 61](#_Toc384151822)

[8.6 System Validation Testing 61](#_Toc384151823)

[9. Test Deliverables 62](#_Toc384151824)

[9.1 Overview 62](#_Toc384151825)

[9.2 Deliverables 62](#_Toc384151826)

[9.2.1 System Test Plan 62](#_Toc384151827)

[9.2.2 Test Cases 62](#_Toc384151828)

[9.2.3 Test Result Reports 62](#_Toc384151829)

[9.2.4 Error Log 63](#_Toc384151830)

[9.2.5 Bug Reports 63](#_Toc384151831)

[10. Test Schedule 64](#_Toc384151832)

[10.1 Overview 64](#_Toc384151833)

[10.2 Schedule 64](#_Toc384151834)

[11. Approvals 65](#_Toc384151835)

[11.1 Purpose 65](#_Toc384151836)

[11.2 Approvals 65](#_Toc384151837)

# Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision Number | Revision Date | Description | Rationale |
| 1.0 | 3/21/2014 | Initial System Test Plan Draft ready for review | Initial and Review Draft |
| 2.0 | 4/1/2014 | Baseline Draft | Updated the first draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# List of Figures

Figure # Title Page #

1-1 Sensors on the Sight By Touch System shown around the body 12

1-2 Closer look of a Sensor Module 13

1-3 Sketch of the System Control Module 13

2-1 Architecture Design Diagram 24

2-2 Architecture Subsystem Level Diagram 25

2-3 Module Decomposition Chart 29

3-1 Relational Diagram 34

# List of Tables

Table # Title Page #

1-1 Acronym Definitions 15-16

2-1 Customer Requirements 18

2-2 Packaging Requirements 19

2-3 Performance Requirements 20

2-4 Safety Requirements 21

2-5 Maintenance and Support Requirements 22

2-6 Other Requirements 23

2-7 Subsystem Data Flows 26

2-8 Requirements Mapping Subsystem Level 27

2-9 Module Data Flows 30

2-10 Requirements Mapping Module Level 31

3-1 Hardware Tests 35-36

3-2 System Control Module Unit Tests 37-38

3-3 Sensor Module(s) Unit Tests 38

3-4 Detection Filter Unit Tests 39

3-5 Battery Level Monitor Unit Tests 40

3-6 On Notification Unit Tests 40

3-7 Object Notification Unit Tests 41

3-8 Battery Notification Unit Tests 41

3-9 Hardware Interface Layer Component Tests 42

3-10 Warning Layer Component Tests 43-44

3-11 System Verification Tests 45-46

3-12 System Validation Tests 46-47

4-1 Risks Table 48-49

7-1 Test Metrics 58

10-1 Test Schedule 64

11-1 Approval Form 65

# 1. Introduction

## Document Overview

The System Test Plan (STP) document is intended to break down in detail the testing procedures to ensure that the Sight By Touch System will meet all necessary requirements and is a working end product when delivered to our customers.

In the Architecture Design Specification (ADS) document, the Sight By Touch System was broken up into three layers. Each layer consisted of various amount of subsystems. In the Detailed Design Specification (DDS) document, those subsystems were broken down into smaller and separate modules with specific purposes. Furthermore, we began outlining the blueprint of our system in the DDS document for this test plan.

This document will go into detail of every testing procedure that will be used for every layer, subsystem, and module. It will also be used to ensure that the Sight By Touch System will meet all the critical requirements defined in the System Requirements Specification (SRS) document. These testing procedures will include hardware, module (unit), subsystem (component), layer (integration), and system verification and validation.

## Product Overview

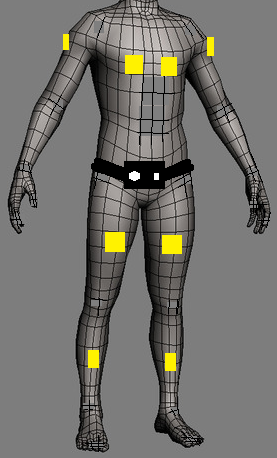
The purpose of the Sight By Touch System is to aid visually impaired individuals since they are unable to rely on their sight. More often than not, visually impaired individuals require some form of aid to help guide them, usually through the use of a cane or a service animal. Our team wishes to remove, or decrease, this dependency by allowing the user to be guided by vibrations from the system. In this way, the user’s hands will be free from having to hold a cane or a leash. With this system, a user shall be able to avoid collision with obstacles that are found in their environment. Our team has decided to make the system an indoor and outdoor system. The system shall lead the user in a safe direction by warning them when an object is within the detection range of the sensors through the use of vibrations from its vibration motors. When the sensors sense an object, the vibration motor closest to that object will vibrate. This system is used solely for the purpose of helping visually impaired users navigate in unknown environments.

The Sight By Touch System shall consist of an external belt that holds the main battery, which connects to the sensors and vibration motors. When the user is not using the product and wishes to charge the battery, it will be removed from the system so it can be charged.

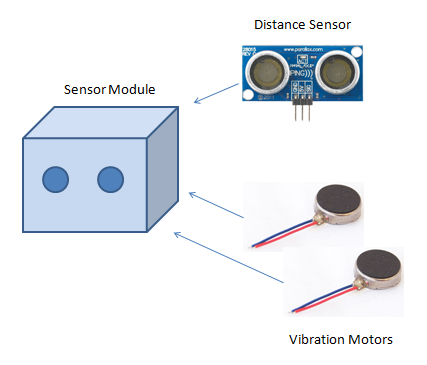
The system will be a full-blown suit with the sensors underneath the clothes and with their wires connected to the microcontroller. The sensors will be detachable to allow for flexibility and cleanliness. The system will focus on the front of the user (180 degrees, at least a 4 foot radius horizontally, and from the neck to the ankles vertically).

There will be multiple vibration motors distributed along the system, embedded in the suit. When the sensor connected to those vibration motors senses an object within the range, the sensor will send a signal to the microcontroller, which will then forward the message to the vibration motor. In order for the user to have a better idea where a detected object is, multiple sensors will detect it and multiple vibration motors will vibrate at different intensities depending on how far the object is. The closer the object is the more intense the vibration will be. In addition, depending on the location of the object being detected, the vibration motors closest to that object will vibrate the most.

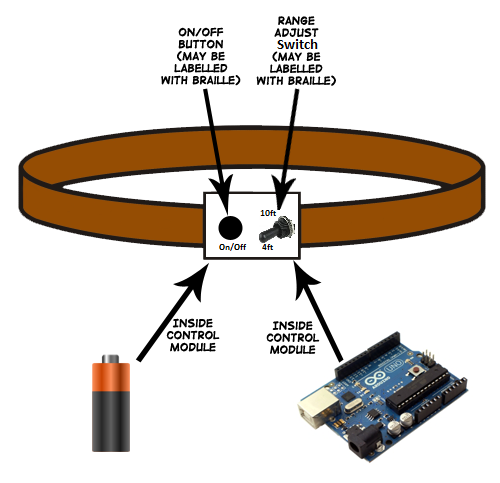
The belt will allow the user to interface with the device. The interface will have an on/off button to turn the sensors on/off. In addition, there will be a switch that allows the user to adjust the range that the sensors will detect. There will be no external elements for all the functions and data manipulation will be done internally.



**Figure 1-1: Sensors on the Sight By Touch System shown around the body**



**Figure 1-2: Closer look of a Sensor Module**



**Figure 1-3: Sketch of the System Control Module**

## Product Scope

The Sight By Touch System is made to aid visually impaired users through the use of vibrations. Users of the Sight By Touch System will be able to move around and gauge where they are. When a user walks near an object, the system will vibrate in the direction where the object is closest. This warns the user that there is an object nearby and thus prevents the user from colliding with the object. With the use of these vibrations created by the system, a visually impaired user will be able to avoid obstacles. The system will not be able to detect the absence of flooring such as a hole or a staircase.

The intended audiences for the Sight By Touch System are individuals who are visually impaired. We consider the visually impaired to be based on the following metrics provided by the American Optometric Association:

* 20/70 to 20/160 is considered moderate visual impairment, or **moderate low vision**
* 20/200 to 20/400 is considered severe visual impairment, or **severe low vision**
* 20/500 to 20/1,000 is considered profound visual impairment, or **profound low vision**
* less than 20/1,000 is considered near-total visual impairment, or **near total blindness**
* no light perception is considered total visual impairment, or **total blindness**

## Acronym Definitions

Below are some frequently used acronyms that are used throughout this document. Some will contain descriptions for more clarification.

|  |  |
| --- | --- |
| ADS | Architecture Design Specification |
| BN | Battery Notification: notifications to the user that the system is at low or critical battery levels |
| CBN | Critical Battery Notification |
| CMDR | Current Maximum Detection Range: the current maximum distance used by the Sight By Touch System to detect an object in the front and sides of the user. This is an integer in the system. |
| DDS | Detailed Design Specification |
| DSR | Distance Sensor Range: the range of distances (2cm to 500cm) in front of a distance sensor in which objects can be detected. |
| DR | Detection Range: the range of distances (between 12 inches and CMDR) the Sight By Touch System will detect objects in the front and sides of the user |
| IDR | Indoor Detection Range: the range of distances (between 12 inches and IMDR) the Sight By Touch System will detect objects in the front and sides of the user for indoors |
| IMDR | Indoor Maximum Detection Range: the maximum distance used by the Sight By Touch System to detect an object in the front and sides of the user for indoors (48 inches). This is an integer in the system. |
| LBN | Low Battery Notification |
| LED | Light Emitting Diode: a hardware component that will light up when voltage is applied |

|  |  |
| --- | --- |
| MINDR | Minimum Detection Range: the minimum distance an object will be detected by the Sight By Touch System in the front and sides of the user (12 inches). This is an integer in the system. |
| ODN | Object Detected Notification |
| ODR | Outdoor Detection Range: the range of distances (between 12 inches and OMDR) the Sight By Touch System will detect objects in the front and sides of the user for outdoors |
| OMDR | Outdoor Maximum Detection Range: the maximum distance used by the Sight By Touch System to detect an object in the front and sides of the user for outdoors (120 inches). This is an integer in the system. |
| ONN | On Notification: notification to the user that the system is powered on and system is ready. |
| SRMO | Sensor Module: a component composed of a distance sensor and vibration motor(s) to detect objects in front and the sides of the user and to notify the user through haptic feedback. |
| SRS | System Requirements Specification |
| STP | System Test Plan |

**Table 1-1: Acronym Definitions**

# ****2. References****

## 2.1 Overview

The STP will include content from the SRS, ADS, and DDS documents. To ensure that the Sight By Touch System is tested and verified accurately, information from these three documents will be taken into consideration when composing this test plan. The subsections below will consist of an overview of the critical contents from other documents that will be utilized in the composition of this test plan.

## 2.2 System Requirements Specification

The SRS consists of the requirements considered essential by Team Survivors, sponsors: Jennifer Svelan and Paul Sassaman, Mr. O’ Dell, and peers that must be implemented in order to provide a complete product. An overview of these requirements will be given below. For more information, reference the SRS document.

### 2.2.1 Customer Requirements

This subsection consists of the requirements provided by our sponsor Jennifer Svelan, which must be implemented in order to provide a complete product. The modification of any of the following requirements will require the approval of the sponsor.

|  |  |  |  |
| --- | --- | --- | --- |
| SRS NO. | Requirement | Description | Priority |
| 3.1 | On /Off | The system shall be able to be turned on/off by the push of a button. The user will be notified that the system was powered on. The preferred method of notification is through vibrations. | 1 – Critical |
| 3.2 | Detect Obstructions | The system shall be able to detect the presence of nearby objects that are in front and/or to the sides of the user that can impede the user from moving forward. The detection area of the system will need to cover at least a 4-foot radius in front of the user including the sides and vertically from the neck to the ankles. | 1 – Critical |
| 3.3 | Obstruction Notification | The system shall be able to warn the user of the presence of nearby objects through the use of vibrations. The intensity of the vibrations felt by the user will correspond to the direction and the distance of the object(s) that the system has detected. | 1 – Critical |
| 3.4 | Battery Powered | A rechargeable battery shall power the system. | 1 – Critical |
| 3.6 | Low Battery Notification | The system shall be able to notify the user when the battery has less than an hour left of charge. | 1 – Critical |
| 3.7 | User Friendly | The system shall be user friendly so that anyone with or without sight can operate it. The functions will be intuitive and easy to learn. | 2 – High |

**Table 2-1: Customer Requirements**

### 2.2.2 Packaging Requirements

This subsection will list and describe each major component of the system. These components include hardware, software and documentation that will be delivered to the user. The system will be packaged with a “Plug-and-Play” type setup.

|  |  |  |  |
| --- | --- | --- | --- |
| SRS NO. | Requirement | Description | Priority |
| 4.1 | System Control Device | The package shall include a device that will allow the user to control and power the system. | 1 – Critical |
| 4.2 | Sight By Touch System | The package shall include the Sight By Touch System with all of the sensor modules (sensors and vibration motors) embedded in the suit. | 1 – Critical |
| 4.3 | Battery | The system shall come with its own rechargeable power source. | 1 – Critical |
| 4.4 | Charger | The package shall include a charger that will be used to recharge the battery. | 1 – Critical |
| 4.5 | Software | The system shall come with the software pre-installed. | 1 – Critical |
| 4.6 | User Manual | The package shall include a user manual (a printed copy and an Audio CD) with instructions on how to wear and use the system. | 3 – Moderate |

**Table 2-2: Packaging Requirements**

### 2.2.3 Performance Requirements

This subsection covers the performance requirements the system must fulfill by specifying how well the system functions overall. Performance requirements will address the quality, coverage and timeliness of our system.

|  |  |  |  |
| --- | --- | --- | --- |
| SRS NO. | Requirement | Description | Priority |
| 5.1 | Switching Range Latency | The system shall be able to switch between any of the predefined ranges in less than 2 seconds. Please see Requirement 8.3 for the predefined ranges. | 2 – High |
| 5.2 | Real-Time Response | The system shall be able to detect objects and notify the user through vibrations between 30 milliseconds to 100 milliseconds. | 1 – Critical |
| 5.3 | Detection Quality | The system shall be able to detect objects within 10 feet max. | 1 – Critical |
| 5.4 | Vibration Intensity | The system shall have a safe vibration intensity of 0.063 m/s2 to 1.15 m/s2. This range ensures that the system can warn the user without harming the user. | 1 – Critical |
| 5.5 | Battery Life | The rechargeable battery shall last between 6-10 hours on full recharge cycles. | 2 – High |
| 5.6 | Boot Up Time | The system shall start up in no more than 5 seconds. | 2 – High |

**Table 2-3: Performance Requirements**

### 2.2.4 Safety Requirements

This subsection covers the safety requirements the system must fulfill. These requirements will cover the safety of both the system and the user. The system must be safe in all aspects regarding hardware, software, and the user’s safety.

|  |  |  |  |
| --- | --- | --- | --- |
| SRS NO. | Requirement | Description | Priority |
| 6.1 | Wearable Material | The system shall not contain materials that could jeopardize the user’s health including, but not limited to: conductive material, allergic material, sharp objects, rusted material, etc. | 2 – High |
| 6.2 | Exposed Circuitry Protection | The system shall have no exposed wires or electrical components that will directly come into contact with the user’s skin. | 1 – Critical |
| 6.3 | Power Supply Protection | The system shall keep the power supply covered by a material that provides protection from any possible power leaks (ex. chemical, electrical, battery meltdown, etc.) | 1 – Critical |
| 6.4 | Heat Dissipation | The system shall dissipate heat produced by the components of the system to prevent overheating. | 1 – Critical |
| 6.5 | Water Resistance | The system shall be water resistant to light rain (precipitation rate less than 2.5 millimeters (0.098 in) per hour). This also includes sweat (32-48oz of fluid per hour). | 1 – Critical |
| 6.6 | Skin Irritation | The system shall not irritate the skin of the user. Vibrations should be kept at a safe range to avoid harming the skin of the user. | 1 – Critical |
| 6.7 | Sensor Module Protection | The system shall keep the sensor module components inside an enclosure to prevent any harm to the user as well as any damage to the components. | 1 – Critical |
| 6.8 | Static Electricity | The system shall not produce static electricity that would harm the user, the system, and anybody that comes into contact with the user | 2 – High |

**Table 2-4: Safety Requirements**

### 2.2.5 Maintenance and Support Requirements

This subsection will list and describe the maintenance and support that will be provided for the delivered system. This includes documentation and maintenance.

|  |  |  |  |
| --- | --- | --- | --- |
| SRS NO. | Requirement | Description | Priority |
| 7.1 | Troubleshooting | The system shall include a troubleshooting guide to identify problems and recommend possible solutions. | 2 – High |
| 7.2 | Future Upgrades and Maintenance | The team will not be responsible for future upgrades and maintenance after the due date of the project. | 3 – Moderate |
| 7.3 | Hardware Maintenance | The team shall provide documentation and hardware details to aid future development teams and users to allow them to extend our product. | 3 – Moderate |
| 7.4 | Software Maintenance | The team shall provide documentation and source code to aid future development teams and users to allow them to extend our product. | 3 – Moderate |

**Table 2-5: Maintenance and Support Requirements**

### 2.2.6 Other Requirements

This subsection includes requirements not previously mentioned in other sections of the document. The addition of these requirements will be needed in order for the product to be deemed complete.

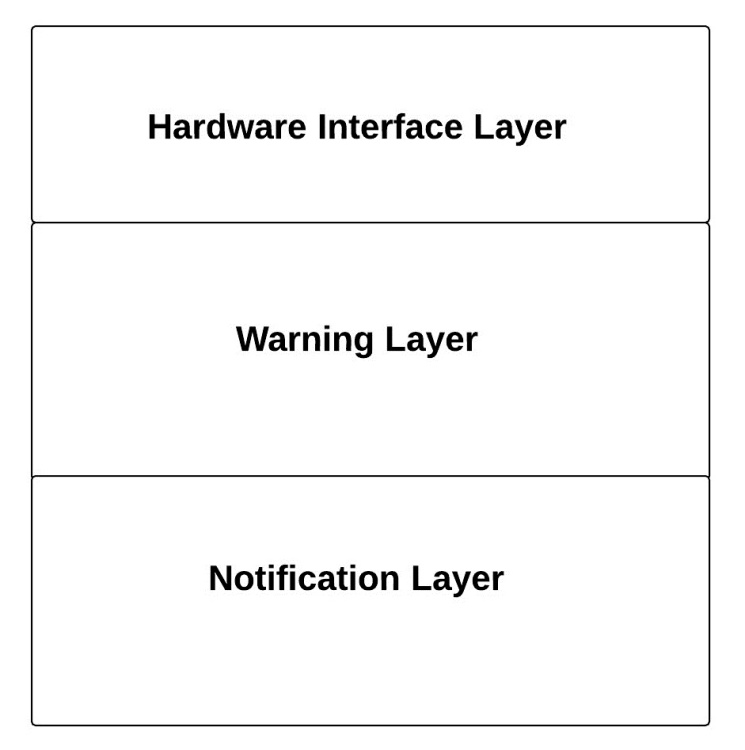
|  |  |  |  |
| --- | --- | --- | --- |
| SRS NO. | Requirement | Description | Priority |
| 8.1 | Weight | The entire system shall be a weight under six pounds. | 3 – Moderate |
| 8.2 | Readability | The system shall have braille next to the on/off button and the sensory range to improve utility for the visually impaired. | 3 – Moderate |
| 8.3 | Adjustable Range | The system shall be able to allow the user to select from the predefined max range radius of 4 or 10 feet. | 2 – High |

**Table 2-6: Other Requirements**

## 2.3 Architecture Design Specification

This section describes the Sight By Touch System architecture, which is the overall structure of how the system will be built. The Sight By Touch System architecture consists of three main layers. The layers include the Hardware Interface Layer, the Warning Layer, and the Notification Layer. An overview of these layers and its subsystems will be given below. For more information, reference the Architecture Design Specification (ADS) document.

### 2.3.1 Layer Overview



**Figure 2-1: Architecture Design Diagram**

**Hardware Interface Layer**

This layer is responsible for handling input to the system and output from the system by providing an interface between the system and any external inputs and outputs. This layer will consist of the System Control Module and the Sensor Modules. This layer will be standardizing the input from the user and distance sensors as well as the output to the vibration motors.

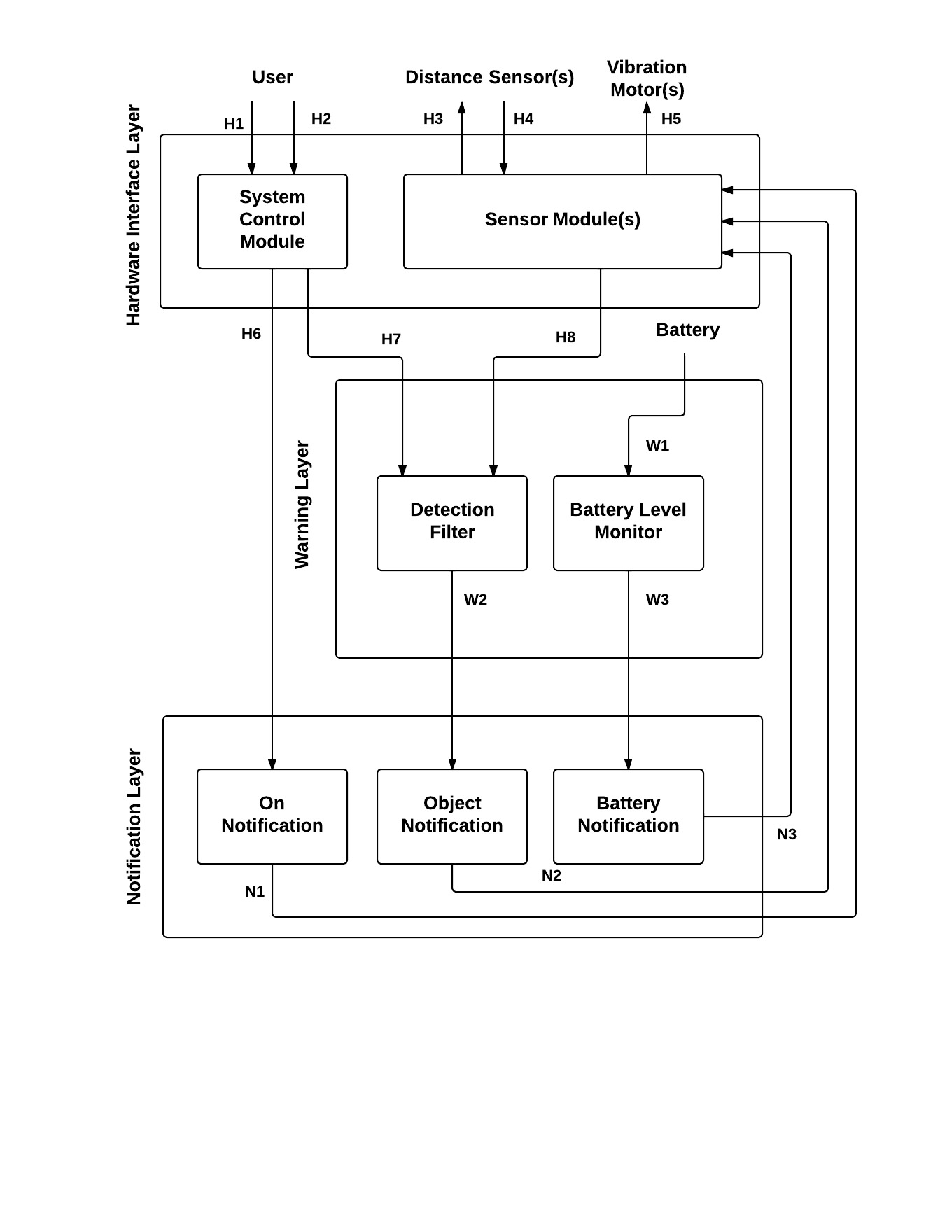
**Warning Layer**

This layer is responsible for processing all the input and output signals. It will be responsible for making decisions based on the input received from the Hardware Interface Layer and triggering notification messages to the Notification Layer.

**Notification Layer**

This layer is responsible for generating notification messages based on data sent from the Warning Layer and sending them to the Hardware Interface Layer to notify the user. The types of notifications are the On Notification, the Objected Detected Notification, the Low Battery Notification, and the Critical Battery Notification.

### 2.3.2 Subsystem Overview



**Figure 2-2: Architecture Subsystem Level Diagram**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Data Element*** |  | ***Descriptions*** | ***Source*** | ***Sink*** |
| ***H1.*** | **User physically presses the button of the system control module to power the system on or off** | | **User** | **System Control Module** |
| ***H2.*** | **User physically moves the switch of the system control module to adjust the sensing range** | | **User** | **System Control Module** |
| ***H3.*** | **A signal to the ultrasonic sensor to operate** | | **Sensor Module(s)** | **Distance Sensor(s)** |
| ***H4.*** | **A response that contains the distance of the closest object detected** | | **Distance Sensor(s)** | **Sensor Module(s)** |
| ***H5.*** | **A signal to the vibration motors to operate** | | **Sensor Module(s)** | **Vibration Motor(s)** |
| ***H6.*** | **The data from the system control module regarding the new state of the system** | | **System Control Module** | **On Notification** |
| ***H7.*** | **The data from the system control module regarding the new maximum detection range** | | **System Control Module** | **Detection Filter** |
| ***H8.*** | **The sensor information containing the sensor ID and distance** | | **Sensor Module(s)** | **Detection Filter** |
| ***W1.*** | **The battery level** | | **Battery** | **Battery Level Monitor** |
| ***W2.*** | **A package containing the object notification** | | **Detection Filter** | **Object Notification** |
| ***W3.*** | **A package containing the battery notification** | | **Battery Level Monitor** | **Battery Notification** |
| ***N1.*** | **A unique signal that will notify the user that the system has been turned on** | | **On Notification** | **Sensor Module(s)** |
| ***N2.*** | **A unique signal that will notify the user that the system has detected an object at a particular range** | | **Object Notification** | **Sensor Module(s)** |
| ***N3.*** | **A unique signal that will notify the user that the battery is low** | | **Battery Notification** | **Sensor Module(s)** |

**Table 2-7: Subsystem Data Flows**

### 2.3.3 Requirements Mapping By Subsystem

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Subsystems** | **System Control Module** | **Sensor Module(s)** | **Detection Filter** | **Battery Level Monitor** | **On Notification** | **Object Notification** | **Battery Notification** |
| **No.** | **Requirement** |  |  |  |  |  |  |  |  |
| **3.1** | On and Off |  | X |  |  |  | X |  |  |
| **3.2** | Detect Obstructions |  |  | X |  |  |  |  |  |
| **3.3** | Obstruction Notification |  |  |  | X |  |  | X |  |
| **3.4** | Battery Powered |  |  |  |  | X |  |  |  |
| **3.6** | Low Battery Notification |  |  |  |  | X |  |  | X |
| **3.7** | User Friendly |  | X | X |  |  | X | X | X |
| **5.2** | Real-Time Response |  |  | X | X | X | X | X | X |
| **5.3** | Detection Quality |  |  | X |  |  |  |  |  |
| **5.4** | Vibration Intensity |  |  |  |  |  | X | X | X |
| **5.5** | Battery Life |  | X | X | X | X | X | X | X |
| **6.4** | Heat Dissipation |  | X | X | X | X | X | X | X |
| **6.6** | Skin Irritation |  |  | X |  |  | X | X | X |
| **8.3** | Adjustable Range |  | X |  | X |  |  |  |  |

**Table 2-8: Requirements Mapping Subsystem Level**

Based off of the requirements mapping for the subsystems, we see that many of the subsystems that are critical to fulfilling the requirements of the Sight By Touch System will play an important part during the testing phase. The subsystems in the Notification Layer and the Sensor Module(s) subsystems fulfill the greatest of key requirements, so these subsystems will be imperative to the overall system and thus. must be tested thoroughly and completely. In contrast, the Battery Level Monitor subsystem fulfills the least amount of key requirements and therefore, the testing conducted will not be as exhaustive, though necessary tests will still be conducted. The System Control Module and Detection Filter are in the middle in terms of the key requirements they fulfill. Since they are not as critical to the overall system, tests will not be as detailed compared to the subsystems in the Notification Layer and Sensor Module(s) subsystem, but they will still be tested with enough detail because they are a priority for the system to fulfill key requirements. The table also shows that all key requirements are addressed by multiple subsystems, and any modification to these subsystems will affect how those key requirements are addressed. Real-time performance, Battery Life, and Heat Dissipation are key requirements that are dependent on the majority of the subsystems. Modification on any subsystems will have the biggest impact on these requirements. This means that, for every modification, these subsystems need to be tested to ensure they are still working properly.

## 2.4 Detailed Design Specification

The Detailed Design Specification (DDS) document is intended to break down in detail each module from each subsystem that was discussed in the ADS document. An overview of these modules will be given below. For more information, reference the DDS document.

### 2.4.1 Module Overview

Module Decomposition Chart revised.emf

**Figure 2-3: Module Decomposition Chart**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Data Element*** |  | ***Descriptions*** | ***Source*** | ***Sink*** |
| ***H1.*** | **User physically presses the button of the system control module to power the system on or off** | | **User** | **Power** |
| ***H2.*** | **User physically turns the knob of the system control module to adjust the sensing range** | | **User** | **Range Setting** |
| ***H3.*** | **A voltage signal to the ultrasonic sensor to operate** | | **Detection** | **Distance Sensor(s)** |
| ***H4.*** | **A signal from the Distance Sensors(s) signifying an object has been detected** | | **Distance Sensor(s)** | **Detection** |
| ***H5.*** | **A voltage signal to the vibration motors to operate** | | **Routing** | **Vibration Motor(s)** |
| ***H6.*** | **A call to the Range Setting module to update the maximum detection range** | | **Power** | **Range Setting** |
| ***H7.*** | **A call to the On Notification module to generate an On Notification to send to the Sensor Modules** | | **Power** | **On Notification** |
| ***H8.*** | **The position of the toggle switch as a value** | | **Range Setting** | **Range Adjuster** |
| ***H9.*** | **The ID of the sensor that operated and the distance to an object from the sensor** | | **Detection** | **Distance Analyzer** |
| ***W1.*** | **The ID of the sensor that detected an object and the distance to that object from the sensor** | | **Distance Analyzer** | **Object Pattern** |
| ***W2.*** | **The voltage from the battery powering the system** | | **Battery** | **Voltage Reader** |
| ***W3.*** | **The current voltage provided by the battery as a percentage of its full voltage capacity** | | **Voltage Reader** | **Battery Analyzer** |
| ***W4.*** | **A call to the Low Battery Pattern module to generate a Low Battery Notification to the Sensor Modules** | | **Battery Analyzer** | **Low Battery Pattern** |
| ***W5.*** | **A call to the Critical Battery Pattern module to generate a Critical Battery Notification to the Sensor Modules** | | **Battery Analyzer** | **Critical Battery Pattern** |
| ***N1.*** | **A unique vibration pattern and the vibration motor id representing all vibration motors for an On Notification** | | **On Pattern** | **Routing** |
| ***N2.*** | **A unique vibration pattern and the vibration motor id associated with the sensor that detected an object for an Object Detected Notification** | | **Object Pattern** | **Routing** |
| ***N3.*** | **A unique vibration pattern and the vibration motor id representing all vibration motors for a Low Battery Notification** | | **Low Battery Pattern** | **Routing** |
| ***N4.*** | **A unique vibration pattern and the vibration motor id representing all vibration motors for a Critical Battery Notification** | | **Critical Battery Pattern** | **Routing** |

**Figure 2-9: Module Data Flows**

### 2.4.2 Requirements Traceability Matrix By Module

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Modules** | **Power** | **Range Setting** | **Detection** | **Routing** | **Interpreter** | **Range Adjuster** | **Distance Analyzer** | **Voltage Reader** | **Battery Analyzer** | **On Pattern** | **Object Pattern** | **Low Battery Pattern** | **Critical Battery Pattern** |
| **No.** | **Requirement** |  |  |  |  |  |  |  |  |  |  |  | **Object** |  |  |
| **3.1** | On and Off |  | x |  |  |  |  |  |  |  |  | x |  |  |  |
| **3.2** | Detect Obstructions |  |  |  | x |  |  | x | X |  |  |  |  |  |  |
| **3.3** | Obstruction Notification |  |  |  |  | x |  |  |  |  |  |  | x |  |  |
| **3.4** | Battery Powered |  | x |  |  |  |  |  |  | x | x |  |  | x | x |
| **3.6** | Low Battery Notification |  |  |  |  | x |  |  |  | x | x |  |  | x | x |
| **3.7** | User Friendly |  | x | x |  |  |  |  |  |  |  |  |  |  |  |
| **5.2** | Real-Time Response |  |  |  | x | x | x | x | X | x | x | x | x | x | x |
| **5.3** | Detection Quality |  |  |  | x |  |  | x | X |  |  |  |  |  |  |
| **5.4** | Vibration Intensity |  |  |  |  |  |  |  |  |  |  | x | x | x | x |
| **5.5** | Battery Life |  | x | x | x | x | x | x | X | x | x | x | x | x | x |
| **6.4** | Heat Dissipation |  | x | x | x | x | x | x | X | x | x | x | x | x | x |
| **6.6** | Skin Irritation |  |  |  |  | x |  |  |  |  |  |  |  |  |  |
| **8.3** | Adjustable Range |  |  | x |  |  |  | x | X |  |  |  |  |  |  |

**Figure 2-10: Requirements Mapping Module Level**

From the requirements mapping of the modules, three requirements that every module in our system addresses must be tested in detail because they constrain the design of the system. The first two requirements come from the performance requirements which are the Real-Time Response and Battery Life, while the third requirement is concerned with Heat Dissipation, which explains the linear flow and the existence of a few modules. The diagram also shows that each module helps to satisfy more than one requirement, and they are related to the performance and safety requirements than to the customer requirements (the actual functionality of the system). Because the modules are related more towards the performance and safety requirements, tests geared towards these requirements will play more of an importance. Tests checking how efficiently the data arrives to each module will be imperative as well as tests that ensure that the system is safe, not only for the user, but for the components as well. The On Pattern, Intensity Level, Low Battery Pattern, and Critical Battery Pattern are the most independent modules as they do not really communicate with each other and there is little communication with other modules from other subsystems. Because of this, these modules can be tested more independently, but not totally isolated from the rest of the system. The Routing module is now a critical module in terms of notification requirements since it is responsible for sending vibration patterns to the vibration motors to notify the user. Since the Routing module is critical, tests that check that the system is notifying the user with the right vibration pattern and in a timely manner will be checked thoroughly.

# ****3. Test Items****

## 3.1 Overview

The Sight By Touch System will be tested in phases in order to ensure a completely working product. Each phase will test a different level of the system. For clarification, the phases and their corresponding level of the system are Hardware Testing Phase: the hardware level, Unit Testing Phase: the module level, Component Testing Phase: the subsystem level, System Verification Phase: the overall system level, and System Validation Phase: the overall system level with respect to requirements. There is no Integration Phase for the layer level since the functionalities of each layer are implemented by one independent subsystem, which the Unit Testing Phase and Component Testing Phase covered.

The Relational Diagram in the following subsection is a visual representation of the phases and the flow of the System Test Plan (STP). The subsections following that will describe each of the phases of the STP in more detail.

## 3.2 Relational Diagram



**Figure 3-1: Relational Diagram**

The Relational Diagram illustrates how the STP is to be implemented. Each path can be implemented concurrently; however, each path must be traversed in order, from left to right, as represented by the diagram. In order for the product to be completely functional, the lowest level items must be completely tested first to ensure that they function as expected. Once finished, these items must be integrated in different stages and tested in order to confirm that they function together properly. Finally, once everything is integrated into one system, the entire system must be completely tested to verify that all elements work as expected and all requirements of the system are satisfied. As a result, each phase on a path must be successfully completed before moving on to the next phase; hence, the required order of each phase for each path must be followed.

## 3.3 Hardware Testing Phase

### 3.3.1 Description

This phase of the STP will test the Sight By Touch System at the hardware level, which is comprised by the actual hardware components. The purpose of this phase is to ensure that all hardware components are functioning as expected before integrating them with the modules of the system in order to reduce potential risks that may be found in the Unit Testing Phase.

### 3.3.2 Hardware Tests

The following table lists the tests to be conducted on the hardware components.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test ID | Individual Hardware Component | Purpose | Input | Expected Output/Action | Test Method | Priority |
| H1 | Arduino Mega 2560 | To test if the Arduino can be powered by a battery | Voltage from a battery | Arduino Mega 2560 powers on | A battery will be wired to the Arduino Mega 2560. | High |
| H2 | Arduino Mega 2560 | To test if the Arduino can read a voltage through a digital pin | 5V signal | Arduino Mega 2560 successfully reads voltage sent into a digital pin | A 5V signal will be sent to the digital pin on the Arduino. | Critical |
| H3 | Arduino Mega 2560 | To test if the Arduino can read a voltage through an analog pin | 5 V signal | Arduino Mega 2560 successfully reads voltage sent into an analog pin | A 5V signal will be sent to the analog pin on the Arduino. | Low |
| H4 | Arduino Mega 2560 | To test if the Arduino can output a voltage through a digital pin | HIGH (Arduino constant) | Arduino Mega 2560 delivers 5V out of a digital pin | A 5V signal will be delivered from the digital pin though a test program. | Critical |
| H5 | Arduino Mega 2560 | To test if the Arduino can output a voltage through a digital PWM pin | An integer between 0-255 representing the duty cycle of a square wave | Arduino Mega 2560 delivers a steady square wave simulating 0-5V out of a digital PWM pin | A 0-5V simulated signal will be delivered from the digital PWM pin though a test program. | Critical |
| H6 | HC-SR04 Ultrasonic Sensor | To test if the sensor can detect objects | 5V Signal  Object within DSR | A voltage signal to the Arduino. | The sensor will be wired to the Arduino. The Arduino will deliver a 5V signal to the sensor to operate. | Critical |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| H7 | DC Vibration Motor | To test if the vibration motors are operational | Voltage signal | Motor operates | After the vibration motor is wired to the Arduino, the Arduino will apply a 5V signal to the motor to operate. | Critical |
| H8 | SPST High-Current Mini Toggle Switch | To test if a voltage passes through when the switch is flipped up | User input via switch flip (up)  Voltage signal | Physical position of toggle switch is changed  Voltage passes through when switch flipped up | A battery will be wired to the switch. The switch will then be flipped to the up position. | Medium |
| H9 | SPST High-Current Mini Toggle Switch | To test if a voltage does not pass through when the switch is flipped down | User input via switch flip (down)  Voltage signal | Physical position of toggle switch is changed  Voltage does not pass through when switch flipped down | A battery will be wired to the switch. The switch will then be flipped to the down position. | Medium |
| H10 | Flat Ribbon Cable | To test if a voltage goes through the cable | Voltage signal | Voltage flows through the cable | A voltage signal will be applied to the cable on one end. | Critical |
| H11 | SPST 3-Amp "Soft-Feel" Push On-Push Off Switch | To test if a voltage passes through when the push switch is pushed down | User input via button push (down)  Voltage signal | Physical position of push switch is changed  Voltage passes through when push switch pushed down | A battery will be wired to the push switch to give voltage. The push switch will then be pushed to the down state. | High |
| H12 | SPST 3-Amp "Soft-Feel" Push On-Push Off Switch | To test if a voltage does not pass through when the push switch is pushed up | User input via button push (up)  Voltage signal | Physical position of push switch is changed  Voltage does not pass through when push switch pushed up | A battery will be wired to the push switch to give voltage. The push switch will then be pushed to the up state. | High |
| H13 | Resistor | To test if the resistor reduces the voltage | Voltage signal | A reduced voltage | A battery will be wired to the resistor to apply voltage to it. | Low |
| H14 | Rechargeable Alkaline AA Batteries w/ Charger | To test if the rechargeable AA alkaline batteries recharge | Voltage from the Charger | Rechargeable AA alkaline batteries are recharged with more voltage | The rechargeable AA alkaline batteries will be connected to the charger. The charger will then be connected to an electrical power source. | High |
| H15 | Rechargeable Battery Pack (Rechargeable AA Alkaline Batteries inside Portable 12VDC 8-Cell Battery Power Supply) | To test if the rechargeable battery pack powers the Arduino. | none | Rechargeable Battery Pack powers Arduino Mega 2560 | The rechargeable battery pack will be wired to the Vin and Gnd pins of the Arduino to supply voltage. | High |

**Table 3-1: Hardware Tests**

## 3.4 Unit Testing Phase

### 3.4.1 Description

This phase of the STP will test the Sight By Touch System at the module level, which is comprised by the modules described in the DDS. The purpose of this phase is to ensure that all modules are functioning as expected before integrating them into the subsystems of the system in order to reduce potential risks that may be found in the Component Testing Phase. The following subsections are organized by the subsystems of the Sight By Touch System described in the ADS.

### 3.4.2 System Control Module Unit Tests

The following table lists the tests to be conducted on the modules of the System Control Module subsystem.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test ID | Module | Purpose | Input | Expected Output/Action | Test Method | Priority |
| USCM1 | Power | To test if all global variables and interrupts were initialized | User input via button push (down) | All global variables and interrupts were initialized | A battery will be wired to the push switch, and the push switch will be wired to the Arduino through the Vin pin. Then, the push switch will be pushed down to allow voltage to flow to the Arduino. | Medium |
| USCM2 | Power | To test if the a call to generate the ONN is made | User input via button push (down) | A call to generate the ONN | A battery will be wired to the push switch, and the push switch will be wired to the Arduino through the Vin pin. Then, the push switch will be pushed down to allow voltage to flow to the Arduino. | High |
| USCM2 | Power | To test if power is cut off | User input via button push (up) | Power is cut off, causing all modules to shut off | A battery will be wired to the push switch, and the push switch (pushed down) will be wired to the Arduino through the Vin pin with voltage flowing. Then, the push switch will be pushed to its up position to stop voltage flow to the Arduino. | High |
| USCM3 | Range Setting | To test if the module can read the switch up position | User input via switch flip (up) | Output HIGH (Arduino constant) representing the up position of the switch. | The switch will be wired to the Arduino to receive and give voltage. The switch will be flipped up to allow the voltage received to flow back into the Arduino. | Medium |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| USCM4 | Range Setting | To test if the module can read the switch down position | User input via switch flip (down) | Output LOW (Arduino constant) representing the down position of the switch. | The switch will be wired to the Arduino to receive and give voltage. The switch will be flipped down to stop the voltage received from flowing back into the Arduino. | Medium |

**Table 3-2: System Control Module Unit Tests**

### 3.4.3 Sensor Module(s) Unit Tests

The following table lists the tests to be conducted on the modules of the Sensor Module(s) subsystem.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test ID | Module | Purpose | Input | Expected Output/Action | Test Method | Priority |
| USM1 | Detection | To test if the module can determine the distance of detected objects correctly. | None | The distance (integer) to the detected object in inches and ID (integer) of the sensor that fired | Call the detection module and place and object with in DSR. | Critical |
| USM2 | Detection | To test if the module can handle if no object was detected | None | An integer with the value of 0 and ID (integer) of the sensor that fired | Call the detection module and place and object outside DSR. | Critical |
| USM3 | Routing | To test if the module can send the vibration pattern and delay pattern to the correct vibration motor | A unique vibration pattern (integer), a unique delay pattern (integer), and the vibration motor id (integer) representing a specific vibration motor | A voltage signal is sent to the proper vibration motor to operate | A vibration motor will be wired to the Arduino as output. The module will be called with an integer representing a vibration pattern, an integer representing a delay pattern, and an integer representing an existing vibration motor. | Critical |
| USM4 | Routing | To test if the module can send the vibration pattern and delay pattern to all vibration motors | A unique vibration pattern (integer), a unique delay pattern (integer), and the vibration motor id (integer) representing all vibration motors | A voltage signal is sent to all vibration motors to operate | All vibration motors will be wired to the Arduino as outputs. The module will be called with an integer representing a vibration pattern, an integer representing a delay pattern and an integer representing all vibration motors. | Critical |

**Table 3-3: Sensor Module(s) Unit Tests**

### 

### 3.4.4 Detection Filter Unit Tests

The following table lists the tests to be conducted on the modules of the Detection Filter subsystem.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test ID | Module | Purpose | Input | Expected Output/Action | Test Method | Priority |
| UDF1 | Range Adjuster | Setting the maximum detection range used by the system to OMDR | Arduino Constant HIGH | Updates the CMDR to OMDR | The module will be called with the Arduino Constant HIGH representing the up switch position. | Medium |
| UDF2 | Range Adjuster | Setting the maximum detection range used by the system to IMDR | Arduino Constant LOW | Updates the CMDR to IMDR | The module will be called with the Arduino Constant LOW representing the down switch position. | Medium |
| UDF3 | Distance Analyzer | To test if the module can determined if the object is within DR | An integer representing the ID of the sensor that fired and an integer representing the distance to the object detected in inches. | The id (integer) of the sensor that detected the object and the distance (integer) to the object in inches | The module will be called with an integer representing a sensor and an integer representing a distance within DR. | High |
| UDF4 | Distance Analyzer | To test if the module can determined if the object is outside of DR | An integer representing the ID of the sensor that fired and an integer representing the distance to the object detected in inches | The data is ignored because the detected object is outside of DR | The module will be called with an integer representing a sensor and an integer representing a distance outside of DR. | High |

**Table 3-4: Detection Filter Unit Tests**

### 

### 3.4.5 Battery Level Monitor Unit Tests

The following table lists the tests to be conducted on the modules of the Battery Level Monitor subsystem.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test ID | Module | Purpose | Input | Expected Output/Action | Test Method | Priority |
| UBLM1 | Voltage Reader | To test if the module can determine the charge percentage left of the battery | None | An integer (between 0 to 100) | A battery will be wired to a voltage divider circuit to reduce the voltage to safe operating levels (see DDS for details). The output of the voltage divider circuit will be wired to the Arduino as input. Then, the module will be called to read voltage of the battery. | Low |
| UBLM2 | Battery Analyzer | To test if the module can determine if the battery is low | An integer representing a low battery percentage | A call to generate a LBN | The module will be called with a percentage (integer) representing a low battery percentage. | Low |
| UBLM3 | Battery Analyzer | To test if the module can determine if the battery is critical | An integer representing a critical battery percentage | A call to generate a CBN | The module will be called with a percentage (integer) representing a critical battery percentage. | Low |
| UBLM4 | Battery Analyzer | To test if the module can determine if the battery is not low | An integer representing a battery percentage above low and critical | None | The module will be called with a percentage (integer) greater than a low battery percentage. | Low |

**Table 3-5: Battery Level Monitor Unit Tests**

### 3.4.6 On Notification Unit Tests

The following table lists the tests to be conducted on the modules of the On Notification subsystem.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test ID | Module | Purpose | Input | Expected Output/Action | Test Method | Priority |
| UON1 | On Pattern | To test if the module can generate the ONN | None | An integer representing the ONN vibration pattern, an integer representing the ONN delay pattern, and an integer representing all of the vibration motors | A call to the module will be invoked. | High |

**Table 3-6: On Notification Unit Tests**

### 3.4.7 Object Notification Unit Tests

The following table lists the tests to be conducted on the modules of the Object Notification subsystem.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test ID | Module | Purpose | Input | Expected Output/Action | Test Method | Priority |
| UO1 | Object Pattern | To test if the module can generate a unique ODN for the MINDR | An integer representing the distance and another integer representing the sensor ID. | An integer representing the unique ODN vibration pattern, an integer representing the unique ODN delay pattern, and integer representing the vibration motor that will be activated | The module will be called with the MINDR and an integer representing the ID of a sensor. | Critical |
| UO2 | Object Pattern | To test if the module can generate a unique ODN for the IMDR | An integer representing the distance and another integer representing the sensor ID. | An integer representing the unique ODN vibration pattern, an integer representing the unique ODN delay pattern, and integer representing the vibration motor that will be activated | The module will be called with the IMDR and an integer representing the ID of a sensor. | Critical |
| UO3 | Object Pattern | To test if the module can generate a unique ODN for the OMDR | An integer representing the distance and another integer representing the sensor ID. | An integer representing the unique ODN vibration pattern, an integer representing the unique ODN delay pattern, and integer representing the vibration motor that will be activated | The module will be called with the OMDR and an integer representing the ID of a sensor. | Critical |

**Table 3-7: Object Notification Unit Tests**

### 3.4.8 Battery Notification Unit Tests

The following table lists the tests to be conducted on the modules of the Battery Notification subsystem.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test ID | Module | Purpose | Input | Expected Output/Action | Test Method | Priority |
| UBN1 | Low Battery Pattern | To test if the module can generate the LBN | None | An integer representing the LBN vibration pattern, an integer representing the LBN delay pattern, and an integer representing all of the vibration motors | A call to the module will be invoked. | Low |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| UBN2 | Critical Battery Pattern | To test if the module can generate the CBN | None | An integer representing the CBN vibration pattern, an integer representing the CBN delay pattern, and an integer representing all of the vibration motors | A call to the module will be invoked. | Low |

**Table 3-8: Battery Notification Unit Tests**

## 3.5 Component Testing Phase

### 3.5.1 Description

This phase of the STP will test the Sight By Touch System at the subsystem level, which is comprised by the subsystems described in the ADS. The purpose of this phase is to ensure that all subsystems are functioning as expected before integrating them into the system in order to reduce potential risks that may be found in the System Verification Phase. The following subsections are organized by the layers of the Sight By Touch System in the ADS.

### 3.5.2 Hardware Interface Layer Component Tests

The following table lists the tests to be conducted on the subsystems of the Hardware Interface Layer. The functionalities of the Sensor Module(s) subsystem are not tested since each of these are implemented by one independent module, which the Unit Testing Phase covered. Also, certain functionalities of the System Control Module will not be tested in this phase since they were also covered in the Unit Testing Phase.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test ID | Subsystem | Purpose | Input | Expected Output/Action | Test Method | Priority |
| CH1 | System Control Module | To test if the subsystem outputs the switch up position when the push switch is pushed down | User input via button push (down) | Output HIGH (Arduino constant) representing the up position of the switch. | A battery will be wired to the push switch, the push switch will be wired to the Arduino through the Vin pin, and the toggle switch flipped up will be wired to the Arduino as digital input. Then, the push switch will be pushed down to allow voltage to flow to the Arduino. | Medium |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| CH2 | System Control Module | To test if the subsystem outputs the switch down position when the push switch is pushed down | User input via button push (up) | Output LOW (Arduino constant) representing the down position of the switch. | A battery will be wired to the push switch, the push switch will be wired to the Arduino through the Vin pin, and the toggle switch flipped down will be wired to the Arduino as digital input. Then, the push switch will be pushed down to allow voltage to flow to the Arduino. | Medium |

**Table 3-9: Hardware Interface Layer Component Tests**

### 3.5.3 Warning Layer Component Tests

The following table lists the tests to be conducted on the subsystems of the Warning Layer subsystem.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test ID | Subsystem | Purpose | Input | Expected Output/Action | Test Method | Priority |
| CW1 | Detection Filter | To test if the ID of the sensor and the distance to an object are outputted after the CMDR is changed to IMDR and the object is within IDR | Arduino Constant LOW  An integer representing the ID of the sensor that fired and an integer representing the distance to the object detected in inches | The id (integer) of the sensor that detected the object and the distance (integer) to the object in inches | The subsystem will set the CMDR to IMDR by passing the Arduino Constant LOW. Then call the subsystem again with an integer representing a sensor and an integer representing a distance within IDR. | High |
| CW2 | Detection Filter | To test if the ID of the sensor and the distance to an object are outputted after the CMDR is changed to OMDR and the object is within ODR | Arduino Constant HIGH  An integer representing the ID of the sensor that fired and an integer representing the distance to the object detected in inches | The id (integer) of the sensor that detected the object and the distance (integer) to the object in inches | The subsystem will set the CMDR to OMDR by passing the Arduino Constant HIGH. Then call the subsystem again with an integer representing a sensor and an integer representing a distance within ODR. | High |
| CW3 | Detection Filter | To test if the ID of the sensor and the distance to an object are not outputted after the CMDR is changed to IMDR and the object is outside IDR | Arduino Constant LOW  An integer representing the ID of the sensor that fired and an integer representing the distance to the object detected in inches | The data is ignored because the detected object is outside of the IDR | The subsystem will set the CMDR to IMDR by passing the Arduino Constant LOW. Then call the subsystem again with an integer representing a sensor and an integer representing a distance outside IDR. | High |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| CW4 | Detection Filter | To test if the ID of the sensor and the distance to an object are not outputted after the CMDR is changed to OMDR and the object is outside ODR | Arduino Constant HIGH  An integer representing the ID of the sensor that fired and an integer representing the distance to the object detected in inches | The data is ignored because the detected object is outside of the ODR | The subsystem will set the CMDR to OMDR by passing the Arduino Constant HIGH. Then call the subsystem again with an integer representing a sensor and an integer representing a distance outside ODR. | High |
| CW5 | Battery Level Monitor | To test if the subsystem can determine if the battery is low | None | A call to generate a LBN | A battery with a voltage level considered low by the subsystem will be wired to a voltage divider circuit to reduce the voltage to safe operating levels (see DDS for details). The output of the voltage divider circuit will be wired to the Arduino as input. Then, the subsystem will be called to read voltage of the battery. | Low |
| CW6 | Battery Level Monitor | To test if the subsystem can determine if the battery is critical | None | A call to generate a CBN | A battery with a voltage level considered critical by the subsystem will be wired to a voltage divider circuit to reduce the voltage to safe operating levels (see DDS for details). The output of the voltage divider circuit will be wired to the Arduino as input. Then, the subsystem will be called to read voltage of the battery. | Low |
| CW7 | Battery Level Monitor | To test if the subsystem can determine if the battery is above the low and critical charge levels | None | The data is ignored since the battery level is not low enough to need a BN | A battery with a voltage level considered greater than a low battery by the subsystem will be wired to a voltage divider circuit to reduce the voltage to safe operating levels (see DDS for details). The output of the voltage divider circuit will be wired to the Arduino as input. Then, the subsystem will be called to read voltage of the battery. | Low |

**Table 3-10: Warning Layer Component Tests**

### 

### 3.5.4 Notification Layer Component Tests

This layer does not require any testing at the subsystem level. The functionalities of each subsystem are each implemented by one independent module, which the Unit Testing Phase covered.

## 3.6 System Verification Phase

### 3.6.1 Description

This phase of the STP will test the Sight By Touch System at the system level, which is comprised by the layers. The purpose of this phase is to ensure proper functionality across layers and that the system is working correctly according to the design. At this point, the system is assumed to be completely constructed.

### 3.6.2 System Verification Tests

The following table lists the tests to be conducted on the entire system.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | System Test | Input(s) | Expected Output(s)/Action(s) | Test Method | Priority |
| SVE1 | Turn on the Sight By Touch System and generate the ONN | Push the power button (down) | System will give an ONN through the vibration motors to indicate that it is system ready. | Push the power button down. | High |
| SVE2 | Turn off the Sight By Touch System | Push the power button (up) | The power cuts off from the entire system, causing it to shut down. | Push the power button to up position. | High |
| SVE3 | Change the DR to IDR | Flip the toggle switch (down) | Updates the CMDR to IMDR | Flip the toggle switch down, and an object is placed within IDR and then moves out of IDR. | Medium |
| SVE4 | Changes the DR to ODR | Flip the toggle switch (up) | Updates the CMDR to OMDR | Flip the toggle switch up, and an object is placed within ODR and then moves out of ODR. | Medium |
| SVE5 | Detect Object within IDR | None | The SRMO that detected the object will provide a unique ODN | Place object within IDR and move it closer and farther from the system while staying within IDR. | Critical |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| SVE6 | Detect Object within ODR | None | The SRMO that detected the object will provide a unique ODN | Place object within ODR and move it closer and farther from the system while staying within ODR. | Critical |
| SVE7 | Detect Object outside of IDR | None | Nothing is done since the detected object is not within the IDR | Place object out of IDR and move it closer and farther from the system while staying out of IDR. | Critical |
| SVE8 | Detect Object outside of ODR | None | Nothing is done since the detected object is not within the ODR | Place object out of ODR and move it closer and farther from the system while staying out of ODR. | Critical |
| SVE9 | Generate LBN | None | All vibration motors activate with a LBN pattern | Power system using a battery with a voltage level considered low by the system. | Low |
| SVE10 | Generate CBN | None | All vibration motors activate with a CBN pattern | Power system using a battery with a voltage level considered critical by the system. | Low |

**Table 3-11: System Verification Tests**

## 3.7 System Validation Phase

### 3.7.1 Description

This phase of the System Test Plan will test the Sight By Touch System at the system level, in which the entire system will be tested based on its requirements. The purpose of this phase is to ensure that the Sight By Touch System is fully functional and satisfies all requirements specified in the SRS. At this point, the system is assumed to be completely constructed.

### 3.7.2 System Validation Tests

The following table lists the tests to be conducted on the entire system.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test ID | System Test | Requirement ID | Input(s) | Expected Output(s)/Action(s) | Test Method | Priority |
| SVA1 | User turns on the Sight By Touch System | 3.1 | User pushes the power button (down) | System will give an ONN to the user through the vibration motors to indicate that it is system ready. | User pushes the power button down. | High |
| SVA2 | User turns off the Sight By Touch System | 3.1 | User pushes the power button (up) | The power cuts off from the entire system, causing it to shut down. | User pushes the power button to up position. | High |
| SVA3 | User changes the DR to IDR | 8.3 | User flips the toggle switch (down) | Updates the CMDR to IMDR | User flips the toggle switch down, and an object is placed within IDR and then moves out of IDR. | Medium |
| SVA4 | User changes the DR to ODR | 8.3 | User flips the toggle switch (up) | Updates the CMDR to OMDR | User flips the toggle switch up, and an object is placed within ODR and then moves out of ODR. | Medium |
| SVA5 | Detect Object within IDR | 3.2, 3.3, 5.3, 5.4, 8.3 | None | An ODN is generated based on the distance to the detected object and sent to the SRMO that detected the object to notify the user. | Place object within IDR and move it closer and farther from the system while staying within IDR. | Critical |
| SVA6 | Detect Object within ODR | 3.2, 3.3, 5.3, 5.4, 8.3 | None | An ODN is generated based on the distance to the detected object and sent to the SRMO that detected the object to notify the user. | Place object within ODR and move it closer and farther from the system while staying within ODR. | Critical |
| SVA7 | Detect Object outside of IDR | 3.2, 3.3, 5.3, 5.4, 8.3 | None | Nothing is done since the detected object is not within the IDR | Place object out of IDR and move it closer and farther from the system while staying out of IDR. | Critical |
| SVA8 | Detect Object outside of ODR | 3.2, 3.3, 5.3, 5.4, 8.3 | None | Nothing is done since the detected object is not within the ODR | Place object out of ODR and move it closer and farther from the system while staying out of ODR. | Critical |
| SVA9 | Generate LBN | 3.4, 3.6, 5.4 | None | All vibration motors activate with a LBN pattern | Power system using a battery with a voltage level considered low by the system. | Low |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| SVA10 | Generate CBN | 3.4, 3.6, 5.4 | None | All vibration motors activate with a CBN pattern | Power system using a battery with a voltage level considered critical by the system. | Low |

**Table 3-12: System Validation Tests**

# ****4. Risks****

## 4.1 Overview

This section covers the potential risks that may arise from testing the Sight By Touch System. Each risk can have an impact on the system at a level of high (expected to occur), medium (likely to occur), or low (unlikely to occur) depending on its severity. The following subsection will describe each risk in tabular form.

## 4.2 Risks Table

This table lists the potential risks that may arise from testing the Sight By Touch system. Each risk lists the impact it can have on the system, a severity level, the element(s) affected by the risk, and a risk strategy, which serves as a possible approach to either eliminate or mitigate the risk.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk ID | Risk | Impact | Severity | Risk Strategy |
| R1 | Fixing bugs creates new bugs | Delay completion of end product | High | Find origin of bug |
| R2 | Damage of hardware components | System may not work properly or gives wrong results. Delays completion of end product as well as affects cost of product | Medium | Verify setup and use of component is correct |
| R3 | Logical errors may be overlooked | Incorrect functionality of system | Low | Regression testing to ensure logic is 100% accurate |
| R4 | Unpredictable behavior on software/hardware components | Delay completion of end product | Low | Test at lowest level prior to progressing to next test level |
| R5 | Issues from user testing system prototype | Question critical features or usability if not intuitive to the user | Medium | Provide a prototype to customer prior to deadline |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| R6 | Dependencies among subsystems/layers of the system | Modifying one subsystem/layer will significantly affect others dependent on that subsystem/layer. May need to revisit architecture and detailed design. | Low | Verify design during implementation |

**Table 4-1: Risks Table**

# 

# 5. Features To Be Tested

## 5.1 Overview

The following features of the Sight By Touch System to be tested are derived from the requirements stated in the SRS. These features will be tested to ensure that they completely satisfy their corresponding requirement. The descriptions discuss the feature satisfying the given requirement. The risks included are described as follows:

**High:** The feature could not be implemented, tested, or is not able to work appropriately.

**Medium:** The feature that has been tested; however, there may be certain cases where the particular function is not guaranteed to work, and therefore, the system will take a best-effort approach.

**Low:** The feature has been tested; it is fully functional and works as described in the SRS.

## 5.2 Customer Requirements

### 5.2.1 On and Off

**Description:** The system shall be able to be turned on/off by the push of a button. The user will be notified that the system was powered on. The preferred method of notification is through vibrations.

**Risk:** Low

**Test Approach:** We will test to verify that the system is initialized when turned on and deactivated when turned off.

### 5.2.2 Detect Obstructions

**Description**:The system shall be able to detect the presence of nearby objects that are in front and/or to the sides of the user that can impede the user from moving forward. The detection area of the system will need to cover at least a 4-foot radius in front of the user including the sides and vertically from the neck to the ankles.

**Risk:** Low

**Test Approach:** We will test to verify that the system’s distance sensors respond correctly when an object is detected within the specified range.

### 5.2.3 Obstruction Notification

**Description:** The system shall be able to warn the user of the presence of nearby objects through the use of vibrations. The intensity of the vibrations felt by the user will correspond to the direction and the distance of the object(s) that the system has detected.

**Risk:** Low

**Test Approach:** We will test to verify that the system’s vibration motors respond correctly when an object is detected within the specified range.

### 5.2.4 Battery Notification

**Description:** The system shall be powered by a rechargeable battery.

**Risk:** Medium

**Test Approach:** We will test to verify that the system will be able to function correctly from a battery source that is rechargeable and portable.

### 5.2.6 Low Battery Notification

**Description:** The system shall be able to notify the user when the battery has less than an hour left of charge.

**Risk:** Medium

**Test Approach:** A low level battery will be connected to the system and we will verify that the system does in fact notify the user through vibrations that the battery is low.

### 5.2.7 User Friendly

**Description:** The system shall be user friendly so that anyone with or without sight can operate it. The functions will be intuitive and easy to learn.

**Risk:** High

**Test Approach:** A survey will be conducted to get feedback about the system. Due to limited time and resources available until the project deadline, there will be limitations to this survey: a limited number and variety of testers and an uncontrolled environment.

## 5.3 Packaging Requirements

### 5.3.1 System Control Device

**Description:** The package shall include a device that will allow the user to control and power the system.

**Risk:** High

**Test Approach:** The System Control Device will be tested to ensure all functionality is working properly with the Sight By Touch System.

### 5.3.2 Sight By Touch System

**Description:** The package shall include the Sight By Touch System with all of the sensor modules (sensors and vibration motors) embedded in the suit.

**Risk:** High

**Test Approach:** The Sight By Touch System will be tested to ensure all functionality is working properly.

### 5.3.3 Battery

**Description:** The system shall come with its own rechargeable power source.

**Risk:** High

**Test Approach:** The battery will be tested to ensure it can power the Sight By Touch System.

### 5.3.4 Charger

**Description:** The package shall include a charger that will be used to recharge the battery.

**Risk:** High

**Test Approach:** The charger will be tested to ensure that the battery can be charged.

### 5.3.5 Software

**Description:** The system shall come with the software pre-installed.

**Risk:** Low

**Test Approach:** The software will be tested to ensure that it is working properly with the hardware.

## 5.4 Performance Requirements

### 5.4.1 Switching Range Latency

**Description:** The system shall be able to switch between any of the predefined ranges in less than 2 seconds.

**Risk:** Medium

**Test Approach:** We will measure the time it takes to update the maximum detection range by flipping the toggle switch and measuring how long it takes to print the new maximum detection range.

### 5.4.2 Real-Time Performance

**Description:** The system shall be able to detect objects and notify the user through vibrations between 30 milliseconds to 100 milliseconds.

**Risk:** Medium

**Test Approach:** We will verify this by measuring the time it takes for the system to detect an object and notify the user.

### 5.4.3 Detection Quality

**Description:** The system shall be able to detect objects within 10 feet max.

**Risk:** Low

**Test Approach:** We will verify this by placing an object within, at, and more than 10 feet away and ensuring that the system can detect this object.

### 5.4.4 Vibration Intensity

**Description:** The system shall have a safe vibration intensity of 0.063 m/s2 to 1.15 m/s2. This range ensures that the system can warn the user without harming the user.

**Risk:** High

**Test Approach:** We will verify this by measuring the vibration intensity of the system by the output produced by the system.

### 5.4.5 Boot Up Time

**Description:** The system shall start up in no more than 5 seconds.

**Risk:** Medium

**Test Approach:** We will verify this by testing how long it takes the system to provide an ONN.

## 5.5 Maintenance and Support Requirements

### 5.5.1 Troubleshooting

**Description:** The system shall include a troubleshooting guide to identify problems and recommend possible solutions.

**Risk:** Low

**Test Approach:** We will create problems that are addressed in the guide and see if a new user can fix the problem based on the solutions given.

## 5.6 Other Requirements

### 5.6.1 Adjustable Range

**Description:** The system shall be able to allow the user to select from the predefined max range radius of 4 or 10 feet.

**Risk:** Low

**Test Approach:** We will test to verify this by placing objects within and outside of IDR when the switch is flipped down and placing objects within and outside of ODR when the switch is flipped up.

# 6. Features Not To Be Tested

## 6.1 Overview

The following features will not be tested since some are verified by design from the Sight By Touch System. In addition, testing may not be feasible on these features or these features may not be implemented. These features describe aspects that have minimum or no functionality with respect to the product.

## 6.2 Customer Requirements

### 6.2.1 Maintainable

By default, the product will be able to use components used to replace old or faulty components if they are the same component type. The distributor of the Sight By Touch System will be responsible for replacing components of the Sight By Touch System.

### 6.2.2 Floor Depth Detection

This is a future feature. It will not be tested since it will not be implemented by project deadline.

### 6.2.3 Inconspicuous Appearance

This feature is subjective to each user’s point of view, and due to time constraints this feature will not be implemented therefore, it will not be possible to conduct tests.

## 6.3 Packaging Requirements

### 6.3.1 User Manual

This will be verified by inspection of the printed copy and Audio CD packaged.

### 6.3.2 Cardboard Container

By default, the product should fit within the dimensions of the container. It will be visually inspected to determine if the product can be placed within the container.

## 6.4 Performance Requirements

### 6.4.1 Battery Life

With the limited time available before project deadline (May 2, 2014), live testing of life of the battery powering the Sight By Touch System is not feasible. Instead, calculations based on the power consumption of the Sight By Touch System and the capacity of the battery will be made to provide an estimate of the battery life for verification purposes.

## 6.5 Safety Requirements

Safety features are physical properties of the product built to ensure that there is no harm to the user or damage to the product itself.

## 6.6 Maintenance and Support Requirements

### 6.6.1 Future Upgrades and Maintenance

The team will not be responsible for future upgrades and maintenance after the deadline of the project.

### 6.6.2 Hardware Maintenance

The team shall provide documentation and hardware details to aid future development teams and users to allow them to extend the product.

### 6.6.3 Software Maintenance

The team shall provide documentation and source code to aid future development teams and users to allow them to extend the product.

## 6.7 Other Requirements

### 6.7.1 Weight

By default, the product should be under the specified weight based on the combined weight of the components of the system.

### 6.7.2 Readability

The Braille labeling will be verified by visual and haptic inspection.

# ****7. Overall Test Strategy****

## 7.1 Overview

The Sight By Touch System will need to be tested to ensure that it meets the requirements that were defined in the SRS and must be verified that the construction and operation of the product is consistent with the architecture defined in the ADS and DDS documents. This section will cover the overall testing strategy that Team Survivors will use to test the Sight By Touch prototype.

## 7.2 Strategy

Testing of the Sight By Touch System will be conducted in phases. Each phase will contain a set(s) of test items that shall be tested against the requirements defined in the SRS and the architecture specified in the ADS and DDS. Tests in each phase will be tested in order from high to low priority. The outcome of each test will be recorded as either pass/fail. The Test Deliverables section will go into further detail of what information will be recorded. A phase must be fully completed before beginning the next phase, but each set of tests in the Unit Testing and Component Testing Phases can be performed independently/concurrently. The order of test phases and their test items are as follows:

1. **Hardware Testing Phase**
2. **Unit Testing Phase**

* System Control Module Unit Tests
* Sensor Module(s) Unit Tests
* Detection Filter Unit Tests
* Battery Level Monitor Unit Tests
* On Notification Unit Tests
* Object Notification Unit Tests
* Battery Notification Unit Tests

1. **Component Testing Phase**

* Hardware Interface Layer Component Tests
* Warning Layer Component Tests

1. **System Verification Testing Phase**
2. **System Validation Testing Phase**

## 7.3 Metrics

Each test has a priority that will be determined by metrics defined by Team Survivors. The priority of each test will be ranked properly according to the requirements defined in the SRS and how it affects the system. The following priorities are defined below:

|  |  |  |  |
| --- | --- | --- | --- |
| Priority | Description | Pass Criteria | Fail Criteria |
| Critical | If a test item of this priority fails, then testing cannot continue until all test items of this priority successfully pass. These tests are related to the system’s main ability to detect objects and notify the user. | 100 % | Less than 100% |
| High | If a test item of this priority fails, then testing should not continue until all test items of this priority successfully pass. These tests are related to the system’s ability to allow the user to interface with the system and the operation on a portable battery will be compromised. | More than or Equal to 95% | Less than 95% |
| Medium | If a test item of this priority fails, then testing could continue without all test items of this priority successfully passing. These tests are related to the system’s ability to allow the user to configure the system. | More than or Equal to 75% | Less than 75% |
| Low | If a test item of this priority fails, then testing can continue without all test items of this priority successfully passing. These tests are related to the system’s ability to monitor the portable battery. | More than or Equal to 50% | Less than 50% |

**Table 7-1: Test Metrics**

## 7.4 Regression

Team Survivors will use regression testing during the implementation phase of the Sight By Touch System. This testing will be conducted each time a bug or issue is found and every time a new component is integrated with the Sight By Touch System such as hardware, modules, subsystems, and layers. It is likely that new bugs or issues will arise when a new component is integrated or when the team fixes existing bugs or issues. This testing will consist of the following steps:

1. Identify/locate bug and attempt to fix
2. Test affected component with original tests and new tests for the bug
3. Repeat all original tests prior to the discovery of the bug to ensure all components tested so far are working correctly

Once all tested components have successfully passed regression testing, normal testing of the Sight By Touch System’s components will resume.

# ****8. Acceptance Criteria****

## 8.1 Overview

Team Survivors will use the acceptance criteria below to grade each test based on a pass or fail. The tests will be divided into five different categories: hardware, unit, component, system verification testing, and system validation testing. Note: Each testing phase does not require tests that check for invalid input since the system has predefined input that the user can only choose from, and any internal input is provided directly by external hardware or is static.

## 8.2 Hardware Testing

**Pass**

The individual hardware functions as expected and provides the expected output/action given the expected input.

**Fail**

The individual hardware does not function as expected and/or does not give the expected output/action given the expected input.

## 8.3 Unit Testing

**Pass**

The module returns the expected output/action given the expected input.

**Fail**

The module does not return the expected output/action given the expected input.

## 8.4 Component Testing

**Pass**

The subsystem returns the expected output/action given the expected input.

**Fail**

The subsystem does not return the expected output/action given the expected input.

## 8.5 System Verification Testing

**Pass**

The Sight By Touch System returns the expected output/action given the expected input.

**Fail**

The Sight By Touch System does not return the expected output/action given the expected input.

## 8.6 System Validation Testing

**Pass**

The Sight By Touch System returns the expected output/action given the expected input. In addition, the Sight By Touch System meets the minimum requirements defined in the Scope Analysis of the SRS document.

**Fail**

The Sight By Touch System does not return the expected output/action given the expected input. In addition, the Sight By Touch System does not meet the minimum requirements defined in the Scope Analysis of the SRS document.

# ****9. Test Deliverables****

## 9.1 Overview

The purpose of this section will be to describe the different artifacts that will be delivered to the stakeholders after each testing phase. The team will also keep a record of each artifact for future reference.

## 9.2 Deliverables

The following is a list of deliverables (artifacts) with a description of the content that each deliverable must contain in order to be considered complete.

### 9.2.1 System Test Plan

This document will provide the necessary information to completely describe the STP for the Sight By Touch System.

### 9.2.2 Test Cases

Each Test Case will need to contain the following information:

* + ID – The test ID.
  + Creator – The name of the team member that created this Test Case.
  + Purpose – The reason for the Test Case.
  + Preconditions – Any condition(s) that must always be true just prior to what the Test Case is testing.
  + Test Code – The source code that will be used to execute the Test Case.
  + Expected Results – The desired results that the test should yield.
  + Comments – This section will be used by the creator for providing any additional information.

### 9.2.3 Test Result Reports

Each Test Result will need to contain the following information:

* + ID – The test ID.
  + Tester – The name of the team member that ran and evaluated the Test Case.
  + Time Stamp – The date and time at which the test was executed.
  + Verdict – The outcome of the Test (passed or failed)
  + Comments – This section will be used by the tester for providing any additional information.
  + Errors – A description about the failure (if any)

### 9.2.4 Error Log

Each Error Log will contain the following information:

* + Error ID
  + Error Description – A description of the error found.
  + Tester – The name of the tester who found the error.
  + Time Stamp – The same as the Time Stamp for the Test Result Report that detected the error.
  + Priority – A number between 1 to 3 where 1 means that the error must be fixed as soon as possible and 3 means that the error can wait.
  + Impact Level – A number between 1 to 3 where 1 means that the error compromises a big portion of the system and 3 means that the error has little to no impact towards the functionality of the system.
  + Development Impact – A description of the error’s impact on development.
  + Testing Impact – A description of the error’s impact on testing.
  + Status – The current state of the error (Waiting, Pending, or Resolved)

### 9.2.5 Bug Reports

Each Bug Report will need to contain the following information:

* + ID – The Bug ID.
  + Time Stamp – The same as the Time Stamp for the Test Result Report that detected the error.
  + Priority – A number between 1 to 3 where 1 means that the error must be fixed as soon as possible and 3 means that the error can wait.
  + Impact Level – A number between 1 to 3 where 1 means that the error compromises a big portion of the system and 3 means that the error has little to no impact towards the functionality of the system.
  + Tester – The name of the team member that ran and evaluated the Test Case.
  + Bug Description – A summary about the failure.
  + Developer – The name of the team member that will fix the bug.
  + Status – The current state of the error (Waiting, Pending, or Resolved).
  + Resolved Date – The date when the bug was officially considered fixed.

# ****10. Test Schedule****

## 10.1 Overview

The purpose of this section will be to provide the planned test schedule that the team will follow during the testing phases.

## 10.2 Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task ID | Task Name | Owner | Planned Start | Planned Finish |
| **3.7.5** | **Hardware Testing** | **H,K** | **Mon 3/17/14** | **Fri 3/28/14** |
| 3.7.5.1 | Push Switch | H | Mon 3/17/14 | Fri 3/21/14 |
| 3.7.5.2 | SPST Toggle Switch | K | Mon 3/17/14 | Fri 3/21/14 |
| 3.7.5.3 | Ultrasonic Sensor | K | Mon 3/24/14 | Fri 3/28/14 |
| 3.7.5.4 | Vibration Motor | K | Mon 3/24/14 | Fri 3/28/14 |
| 3.7.5.5 | Battery Pack | H | Mon 3/24/14 | Fri 3/28/14 |
| 3.7.5.6 | Arduino Mega 2560 | H | Mon 3/17/14 | Fri 3/21/14 |
| **3.7.6** | **Unit/Modules** | **T** | **Fri 3/28/14** | **Wed 4/9/14** |
| 3.7.6.1 | Power | H | Sat 3/29/14 | Wed 4/2/14 |
| 3.7.6.2 | Detection | K | Sat 3/29/14 | Wed 4/2/14 |
| 3.7.6.3 | Routing | V | Sat 3/29/14 | Wed 4/2/14 |
| 3.7.6.4 | Distance Analyzer | M | Sat 3/29/14 | Wed 4/2/14 |
| 3.7.6.5 | Object Pattern | G | Sat 3/29/14 | Wed 4/2/14 |
| 3.7.6.6 | On Pattern | G,M | Wed 4/2/14 | Sat 4/5/14 |
| 3.7.6.7 | Range Setting | K | Wed 4/2/14 | Sat 4/5/14 |
| 3.7.6.8 | Range Adjuster | V | Wed 4/2/14 | Sat 4/5/14 |
| 3.7.6.9 | Voltage Reader | H | Wed 4/2/14 | Sat 4/5/14 |
| 3.7.6.10 | Critical Battery Pattern | M,G | Sun 4/6/14 | Wed 4/9/14 |
| 3.7.6.11 | Low Battery Pattern | K,V | Sun 4/6/14 | Wed 4/9/14 |
| 3.7.6.12 | Battery analyzer | H | Sun 4/6/14 | Wed 4/9/14 |
| **3.7.7** | **Component/Subsystems** | **T** | **Wed 4/9/14** | **Sun 4/13/14** |
| 3.7.7.1 | System Control Module | G,V | Wed 4/9/14 | Sun 4/13/14 |
| 3.7.7.2 | Battery Level Monitor | H,K | Wed 4/9/14 | Sun 4/13/14 |
| 3.7.7.3 | Detection Filter | M | Wed 4/9/14 | Sun 4/13/14 |
| 3.7.8 | System Verification and Validation | T | Mon 4/14/14 | Tue 4/22/14 |

**Table 10-1: Test Schedule**

# ****11. Approvals****

## 11.1 Purpose

This section is reserved for all individuals involved in the project to sign if they approve

## 11.2 Approvals

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Title | Signature | Date |
| Mike O'Dell | Project Supervisor |  |  |
| Jennifer Svelan | Project Sponsor |  |  |
| Paul Sassaman | Project Sponsor |  |  |
| Gerardo Guevara | Project Lead |  |  |
| Margaret Floeter | Team Member |  |  |
| Victor Rodriguez | Team Member |  |  |
| Kevin Tran | Team Member |  |  |
| Henry Loh | Team Member |  |  |

**Table 11-1: Approval Form**