# Introduction

## Architectural Design Specification Purpose and Use

## Project Concept

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.

## Project Scope

The Sight By Touch system shall consist of an external belt that holds the main battery, which connects to the sensors and vibration motors. The batteries will be in a battery pack. When the user is not using the product and wishes to charge the batteries, the batteries will be removed from the system and put into a docking station to charge.

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 scope of the system will be restricted to the front of the user (180 degrees, at least a 3 feet radius horizontally, and from the neck to the ankles vertically). When a sensor detects an object in its range, the sensor will send a signal to the microcontroller, which will then forward the message to the vibration motor. How close the object is will determine the intensity of the vibrations. The system will not be able to detect the absence of flooring such as a hole or a staircase.

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. There can be multiple vibration motors vibrating at the same time and at different intensities depending on where the object is located. 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 system will also have an external belt (placed around the user’s waist) that 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 knob 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.

## Key Requirements

# Architectural Layer Definitions

## Notification Layer

The Notification Layer will receive commands from the Processing Layer and relay them to the appropriate sensor modules so that the system may correctly notify the user of objects, whether the system is on/off, and whenever the battery is low.

# Subsystem Definitions

## Notification Layer: On/Off Notification Subsystem

### General Subsystem Description

The On/Off Notification Subsystem will receive commands from the On/Off Subsystem and interpret them into appropriate sensor module instructions

### Assumptions

* The On/Off Notification Subsystem will support control of eight sensor modules

### Responsibilities

* The On/Off Notification Subsystem will be responsible for interpreting the On/Off Subsystem’s instructions
* The On/Off Notification Subsystem will be responsible for delivering voltage to the sensor modules

### Subsystem Inter-Layer Interfaces

|  |  |  |  |
| --- | --- | --- | --- |
| Method Name | Description | Information Required | Information Returned |
| **notificationListener** | The On/Off Subsystem will listen for instructions from On/Off Subsystem | * None | * Request |

### Subsystem Public Interfaces

|  |  |  |  |
| --- | --- | --- | --- |
| Method Name | Method Description | Data Required | Data Returned |
| **systemOnOff** | Notify the user that the system is On/Off with vibrations | * None | * Request |

### Consumer Producer Relationships

Consumes:

* + Messages from the On/Off Subsystem

Produces:

* + None

## Notification Layer: Object Notification Subsystem

### General Subsystem Description

The Object Notification Subsystem will receive commands from the Logical Subsystem and interpret them into appropriate sensor module instructions

### Assumptions

* The Object Notification Subsystem will support control of eight sensor modules

### Responsibilities

* The Object Notification Subsystem will be responsible for interpreting the Logical Subsystem’s instructions
* The Object Notification Subsystem will be responsible for delivering voltage and instructions to the sensor modules

### Subsystem Inter-Layer Interfaces

|  |  |  |  |
| --- | --- | --- | --- |
| Method Name | Description | Information Required | Information Returned |
| **notificationListener** | The Object Subsystem will listen for instructions from the Logical Subsystem | * None | * Request |

### Subsystem Public Interfaces

|  |  |  |  |
| --- | --- | --- | --- |
| Method Name | Method Description | Data Required | Data Returned |
| **objectNotification** | Notify the user that there is an object nearby with vibrations | * None | * Request |

### Consumer Producer Relationships

Consumes:

* + Messages from the Logical Subsystem

Produces:

* + None

## Notification Layer: Battery Notification Subsystem

### General Subsystem Description

The Battery Notification Subsystem will receive commands from the Logical Subsystem and interpret them into appropriate sensor module instructions

### Assumptions

* The Battery Notification Subsystem will support control of eight sensor modules

### Responsibilities

* The Battery Notification Subsystem will be responsible for interpreting the Logical Subsystem’s instructions
* The Battery Notification Subsystem will be responsible for delivering voltage and instructions to the sensor modules

### Subsystem Inter-Layer Interfaces

|  |  |  |  |
| --- | --- | --- | --- |
| Method Name | Description | Information Required | Information Returned |
| **notificationListener** | The Notification Subsystem will listen for instructions from the Logical Subsystem | * None | * Request |

### Subsystem Public Interfaces

|  |  |  |  |
| --- | --- | --- | --- |
| Method Name | Method Description | Data Required | Data Returned |
| **batteryNotification** | Notify the user that there the system’s battery is low | * None | * Request |

### Consumer Producer Relationships

Consumes:

* + Messages from the Logical Subsystem

Produces:

* + None

# Architectural Testing Considerations

In this section, we will define testing considerations relevant to each layer of the architecture that are addressed in this section. This is not a full testing plan, but ideas and goals that address how we will verify and validate the architecture that we have specified.

This section is divided into two subsections. We define the considerations on how to approach the validation of our system overall and at each layer in the architecture.

## Overall Considerations

### Durability

* **Wearing**: The user shall be able to easily wear the System. Once worn, the System should stay on the user comfortably.
* **Toughness**: The System shall be able to withstand any normal day-to-day contact the user would normally experience during operation.

### User

* **Ease of Use**: All layers should be designed and tested as to have minimal work for the user. The user should only be responsible for wearing the System, attaching connections to the System Control Unit, and communicating information from the user input in the Data Acquisition Layer. Any other functions required for the Sight By Touch System shall be handled by the System itself.
* **Clarity**: The Data Acquisition Layer and Notification subsystems shall clearly communicate to the user.

## Layer-level Considerations

### Data Acquisition Layer

* **Minimal Integration**: Any User input, Sensor input and Battery charge input should communicate individually to its event subsystem inside the Event Handler Layer.
* **Independence**: Any sensors within the Data Acquisition Layer shall be independent of each other, and should work and supply information in the case that any of the other sensors fail. The Data Acquisition Layer’s performance shall be independent of the other layers.
* **Data Integrity**: Data retrieved from the User, Sensors and Battery charger shall be retrieved in a timely manner so that the System Control Unit can produce real time feedback.

### Event Handler Layer

* **Communication**: The Event Handler Layer must be able to gather and transmit appropriate information to the Processing Layer. This information must be preprocessed and prepared before being sent to the Powering/Logical subsystems.
* **Independence**: The Event Handler Layer must not be dependent on any other layers. The Event Handler Layer shall be designed so that if any other event is created, no other components of the architecture shall need to be altered. The Event Handler Layer shall only communicate with the Processing Layer.

### Processing Layer

* **Centralized Control**: All controls and decisions will be made in the Processing Layer. All communication with the Event Handler Layer is for the purposes of retrieving and/or sending information.
* **Independence**: The Processing Layer shall be designed in a way that if a new Processing Layer was composed, no other layers shall need to be altered. Black Box Testing shall be utilized for all subsystems within the Processing Layer.

### Notification Layer

* **Correctness**: The Notification Layer shall be designed with proper code to clearly differentiate between object, on and off notifications.
* **Fault Notification**: The Processing Layer shall be able to detect if there is a simple malfunction in the Notification Layer.