SFWRENG 4G06 - System Design

Group: NextStep (Group 10)

Justin Rosner, rosnej1
Daniel Noorduyn, noorduyd
Mengxi Lei, leim5
Alexander Samaha, samahaa
Tishko Araz, arazt

Department of Computing and Software
McMaster University
February 28, 2022

Contents

List of Tables

List of Figures

1 Revisions

Revision Number	**Date**	**Reason for Change**
Revision 0	December 20, 2021	N/A
Revision 1	February 28, 2022	Updating system design to account for using a Lidar instead of a camera, change from Arduino Uno to Mega, Added more ultrasonic sensors.

Table 1: Revision History

2 Overview

2.1 Purpose

The purpose of this project is to create an assistive device for people with vision impairments that will help them navigate situations where they would often have to use a white cane or other seeing aids. Current devices that aim to assist the visually impaired often rely on the reaction time of the individual after having close contacts with obstacles, rather than preemptively helping the user navigate around them. Additionally, NextStep would allow for the user to walk about in public indoor settings without drawing any unwanted attention to themselves.

NextStep aims to be a wearable device in the form of a hat, that will detect moving and stationary objects in an indoor setting through the use of a Lidar and ultrasonic sensors. This data will be fused together to form an image of the surrounding landscape. It will be able to relay to the user where the obstacles are in their path, thus providing them a way of navigating through any potential hazards.

The following System Design document will cover the operation of NextStep, any undesired error handling, system components, and system behaviour.

2.2 Scope

The system described in the remainder of this document is one that is meant to guide visually impaired people around indoor settings. The user will be able to put on the hat that contains NextStep, turn it on, and from there it will be able to detect any stationary and slow moving obstacles that would potentially be hazards otherwise. Additionally, NextStep will be able to locate where any staircases may be, and prevent the user from sustaining any injuries related to falling down the stairs.

2.3 Context Diagram

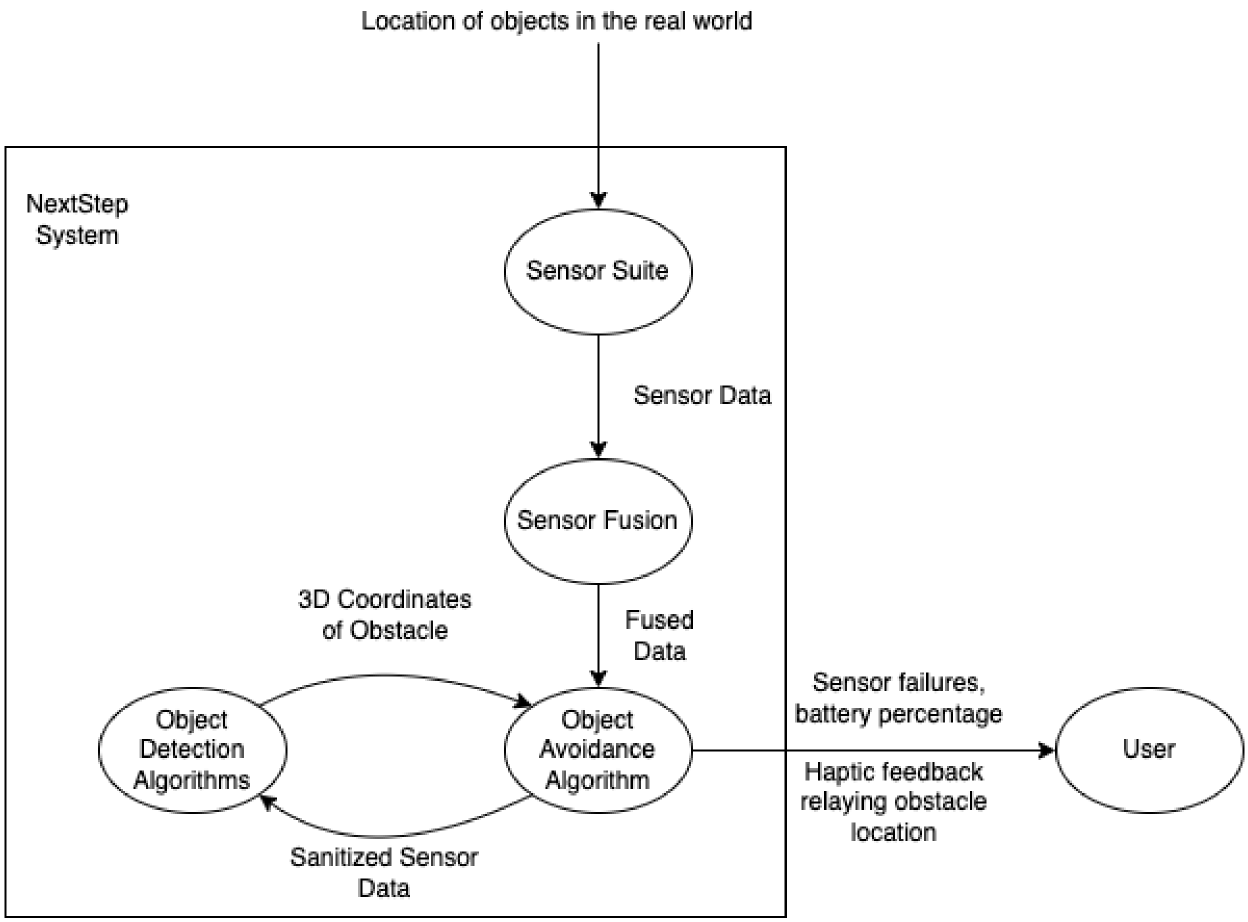


Figure 1: Context Diagram of NextStep

2.4 Component Diagram

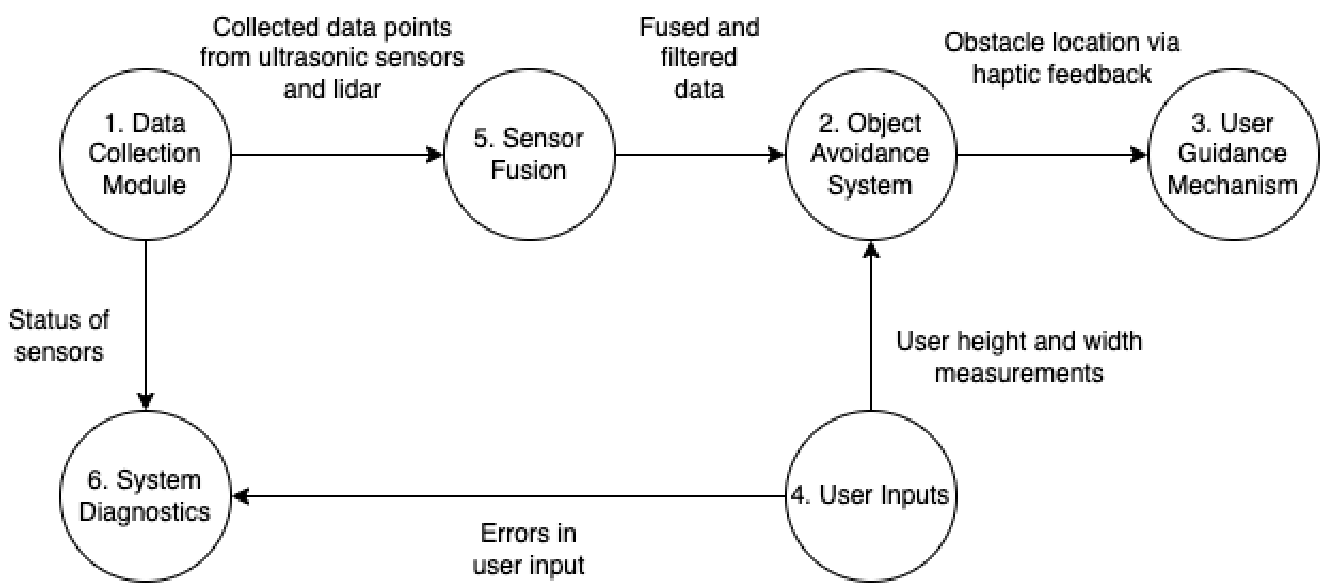


Figure 2: Component Diagram of the NextStep System

2.5 Assumptions

Any design assumptions related to NextStep are listed below.

Assumption 1	The operation environment will consist only of indoor settings.
Rationale	Outdoor environments provide too many different factors to deal with and thus are out of the scope of this project.

Assumption 2	There will be no high speed objects in the indoor environment (e.g. think go kart track).
Rationale	The sensors will not be able to detect and relay the information back to the user quickly enough if there are high speed obstacles.

Assumption 3	Users of NextStep will not make any erratic movements of the head.
Rationale	Since NextStep will be worn on the user's head, any jumping movements or shaking of the head could skew the sensor readings and lead to improper guidance information being relayed to the user.

Assumption 4	The user must be able to properly wear a hat.
Rationale	The components of NextStep are to be located in the top of a hat. If the user can not properly wear the hat then NextStep will be rendered useless.

Assumption 5	The floor of the indoor setting will be suitable for walking (i.e. No slippery areas or wet zones).
Rationale	Detecting slippery spots on the floor is out of the scope of this project.

Assumption 6	The sensors and Lidar will not have dust or debris interfering with their readings.
Rationale	Depending on how the user chooses to store NextStep, there could be debris that gets in the sensors or blocks the Lidar resulting in dust being accumulated on the device. This would cause NextStep to potentially miss obstacles.

Assumption 7	The device will not be worn in situations where it could get water damage.
Rationale	The electronic components of NextStep are not water proof and water damage will lead to catastrophic failures..

3 System Variables

3.1 Monitored and Controlled Variables

Monitored Name	Type	Unit	Description
v_{self}	Speed	m/s	Movement speed of the user who is wearing the device.
$a_{turning}$	Angle	degree/s	Turning angle of the user who is wearing the device.
$height$	Height	m	The height of the user.
$d_{obstacle}$	Distance[ ]	m	Array of distances between the obstacles and the user.
$a_{obstacle}$	Angles[ ]	degree	Array of degrees of where the obstacles are located with respect to the user.
$p_{battery}$	Percentage	%	Percentage of battery life remaining.
$bubble\_boundary$	Float[ ]	cm	Maximum distance considered for obstacles.

Table 2: Monitored Variables Table

Controlled Name	**Type**	**Unit**	**Description**
hap_strength	Float	%	The strength of the buzzing created by the haptic motors.
vol	Volume	decibels	The volume that NextStep uses to communicate.
word_speed	WordSpeed	wpm	The rate at which NextStep communicates at.

Table 3: Controlled Variables Table

3.2 Constants

Constant Name	**Value**	**Unit**	**Description**
$d_{warning_{min}}$	0.5	m	Minimum allowed distance between the obstacle and user before communicating with the user about it.
$t_{warning_{min}}$	2	seconds	Minimum allowed expected collision time between the obstacle and user before communicating with the user about it.
$S_{min\_detection}$	0.01	m	Minimum object detection size.
$S_{min\_distance}$	0.1	m	Minimum object separation.
v_{max}	2	m/s	Maximum velocity of obstacle.
k_i	1.5	$\mathbb{R}$	Scaling constant for bubble boundary.
a_0	$\frac{\pi}{5}$	rad	Individual detection zone angles.
N	5	$\mathbb{N}$	Number of detection zones.

Table 4: Constant Variables Table

4 Behaviour Overview

1. **Data Collection Module:** This component collects and filters data from an indoor setting using ultrasonic sensors and a Lidar. This component will communicate this information to the Sensor Fusion module.

2. **Object Avoidance System** This component receives fused data from the Sensor Fusion component and then uses the Bubble Rebound algorithm to calculate a path through the obstacles for the user to follow.

3. **User Guidance Mechanism** This component receives the necessary information from the Object Avoidance System and relays to the user via haptic feedback guidance on how to avoid running into obstacles.

4. **User Input Module** Responsible for collecting user height to pass on to the Object Avoidance System.

5. **Sensor Fusion Module** This component receives unfiltered data from the Data Collection Module and then proceeds to run a Kalman filter on it. The Kalman filter serves two purposes, to filter and then fuse the two data streams coming from the ultrasonic sensors and the Lidar. When this data is fused we can then send it to the Object Avoidance System to be used in the navigational algorithm.

6. **System Diagnostics Module** This module will relay non-guidance information back to the user. This will include things such as battery level and any sensors that are malfunctioning.

5 Component Traceability

Component Module	Functional and Non-Functional Requirement
Data Collection Module	FR1 FR10 FR15 FR17 FR18 FR19 PR5 PR7 PR8

Table 5: Component Traceability - Data Collection Module

Component Module	Functional and Non-Functional Requirement
Object Avoidance System	FR1 FR2 FR3 FR5 FR6 FR10 FR11 FR12 FR16 FR19 PR1 PR6 PR8 PR9

Table 6: Component Traceability - Object Avoidance System

Component Module	Functional and Non-Functional Requirement
User Guidance Mechanism	FR2 FR2 FR3 FR5 FR6 FR10 FR11 FR12 FR16 PR1

Table 7: Component Traceability - User Guidance Mechanism

Component Module	Functional and Non-Functional Requirement
User Input Module	FR7 FR8

Table 8: Component Traceability - User Inputs Module

Component Module	Functional and Non-Functional Requirement
Sensor Fusion Module	FR1 FR5 FR11 FR16 FR18 FR19 PR6 OE2

Table 9: Component Traceability - Sensor Fusion Module

Component Module	Functional and Non-Functional Requirement
System Diagnostics Module	FR4 FR9 FR13 FR14 UH5 UH6

Table 10: Component Traceability - System Diagnostics Module

6 Component Overview

6.1 Data Collection Module

Description

This module will be used to accept sensor inputs to the system. The sensors gathering data are ultrasonic sensors, a Lidar sensor and an accelerometer. It will clean the gathered data and output coordinates of detected objects to the Object Avoidance System.

Inputs and Outputs

Inputs: Sensor input defining:

Input Name	**Input Type**	**Range**	**Units**	**Comment(s)**
lidar_point	Float Struct Lidar Input	(150-12000, 0-360)	(cm, deg)	Distance of a detected Lidar point and its angle
sensor_left	Float Sensor Input	0-500cm	cm	Distance of Object in User's Path
sensor_right	Float Sensor Input	0-500cm	cm	Distance of Object in User's Path
sensor_center_left	Float Sensor Input	0-500cm	cm	Distance of Object in User's Path
sensor_center_right	Float Sensor Input	0-500cm	cm	Distance of Object in User's Path
sensor_center	Float Sensor Input	0-500cm	cm	Distance of Object in User's Path
sensor_top_center	Float Sensor Input	0-500cm	cm	Distance of Object in User's Path, angled upwards
sensor_top_left	Float Sensor Input	0-500cm	cm	Distance of Object in User's Path, angled upwards
sensor_top_right	Float Sensor Input	0-500cm	cm	Distance of Object in User's Path, angled upwards
$user\_acceleration_x$	Float Sensor Input	N/A	m/s^2	User's Acceleration in x Plane
$user\_acceleration_y$	Float Sensor Input	N/A	m/s^2	User's Acceleration in y Plane
$user\_acceleration_z$	Float Sensor Input	N/A	m/s^2	User's Acceleration in z Plane

Table 11: Data Collection Module - Inputs

Outputs: Cleaned data of detected objects and user's acceleration:

Output Name	Output Type	Range	Units	Comment(s)
$scanned\_points[\,]$	Float struct [ ]	(150-12000, 0-360)	(cm, deg)	Cleaned Distance and Angle of Lidar points
$sensor\_left$	Float	0-500cm	cm	Cleaned Distance of a Detected Obstacle
$sensor\_right$	Float	0-500cm	cm	Cleaned Distance of a Detected Obstacle
$sensor\_center\_left$	Float	0-500cm	cm	Cleaned Distance of a Detected Obstacle
$sensor\_center\_right$	Float	0-500cm	cm	Cleaned Distance of a Detected Obstacle
$sensor\_center$	Float	0-500cm	cm	Cleaned Distance of a Detected Obstacle
$user\_velocity$	Float	N/A	$\frac{cm}{second}$	Calculated from moving point average of user's acceleration in x Plane

Table 12: Data Collection Module - Outputs

Exception Handling

Input Name	Input Type	Exception	Exception Handling
$sensor\_left$	Float Sensor Input	$sensor\_left$ = null	Sensor Failure
$sensor\_right$	Float Sensor Input	$sensor\_right$ = null	Sensor Failure
$sensor\_center\_left$	Float Sensor Input	$sensor\_center\_left$ = null	Sensor Failure
$sensor\_center\_right$	Float Sensor Input	$sensor\_center\_right$ = null	Sensor Failure
$sensor\_center$	Float Sensor Input	$sensor\_center$ = null	Sensor Failure
$sensor\_top\_left$	Float Sensor Input	$sensor\_top\_left$ = null	Sensor Failure
$sensor\_top\_right$	Float Sensor Input	$sensor\_top\_right$ = null	Sensor Failure
$sensor\_top\_center$	Float Sensor Input	$sensor\_top\_center$ = null	Sensor Failure
$user\_acceleration_x$	Float Sensor Input	$user\_acceleration_x$ = null	Accelerometer Failure
$user\_acceleration_y$	Float Sensor Input	$user\_acceleration_y$ = null	Accelerometer Failure
$user\_acceleration_z$	Float Sensor Input	$user\_acceleration_z$ = null	Accelerometer Failure
$lidar\_point$	Float Struct Lidar Input	$scanned\_input$ = (> 12000, >360)	Lidar motor and laser out of sync
$lidar\_point$	Float Struct Lidar Input	$scanned\_input$ = null	Lidar failure

Table 13: Data Collection Module - Exception Handling

Timing Constraints

Within $t_{timeout}$, a new set of clean data from the sensors needs to be passed onto the Object Avoidance System.

Initialization

When the device powers on, all array variables initialize to empty arrays and all other variables initialize to null.

Diagrams

UART and PWM wire diagram for RPLidar to Arduino Mega

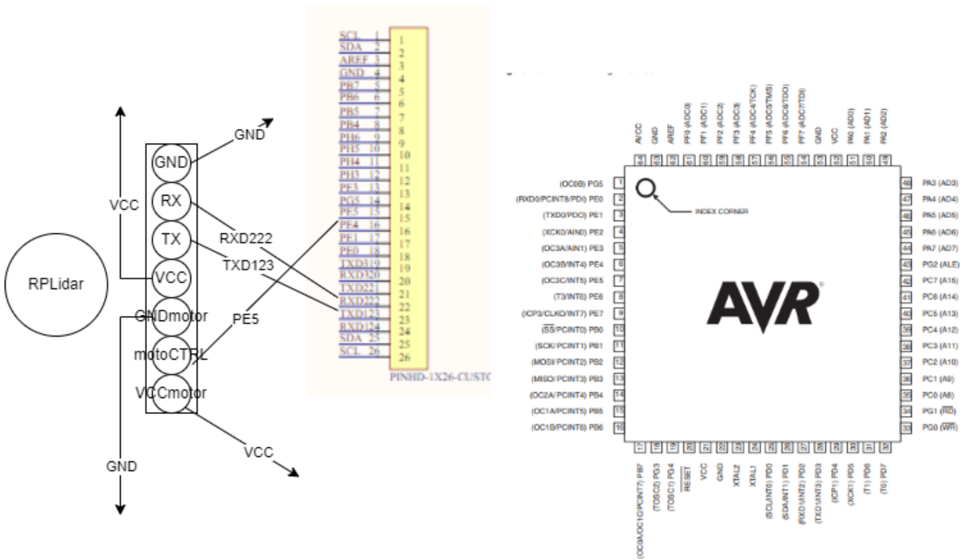
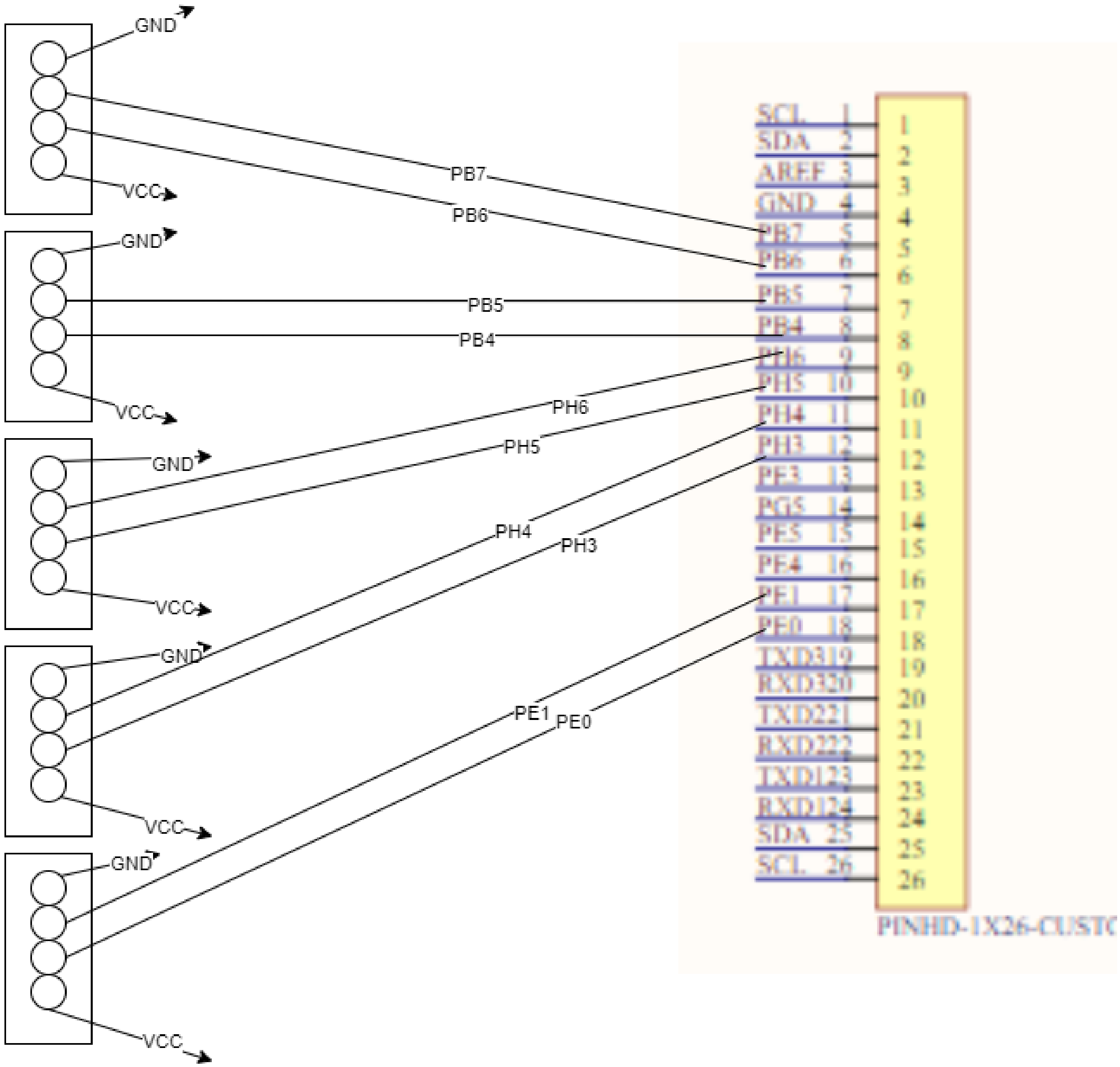


Figure 3: UART and PWM (pin 3) Connection Between Arduino Mega (ATmega2560) rev 3 and RPLidar

Ultra Sonic Sensors Wiring Diagram



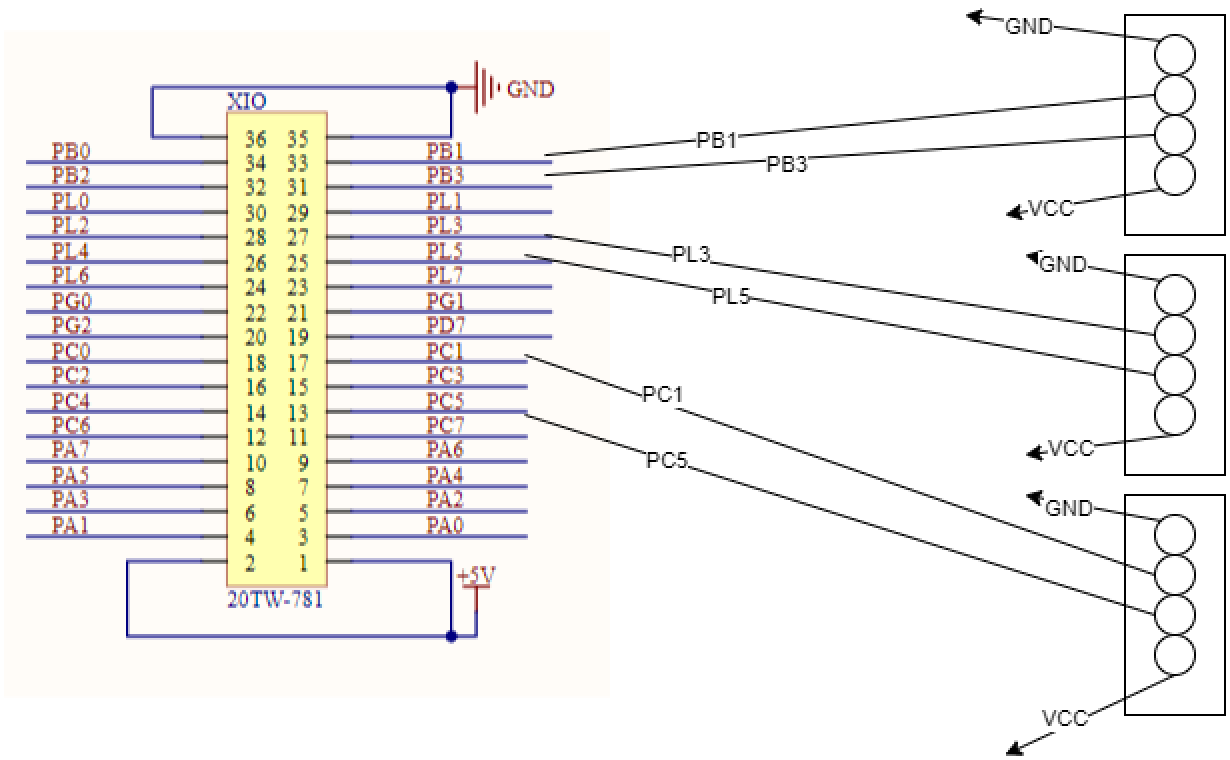


Figure 4: Connection of 8 ultra sonic sensors to detect objects within 5 disjoint ranges to the micro-controllers digital I/O ports

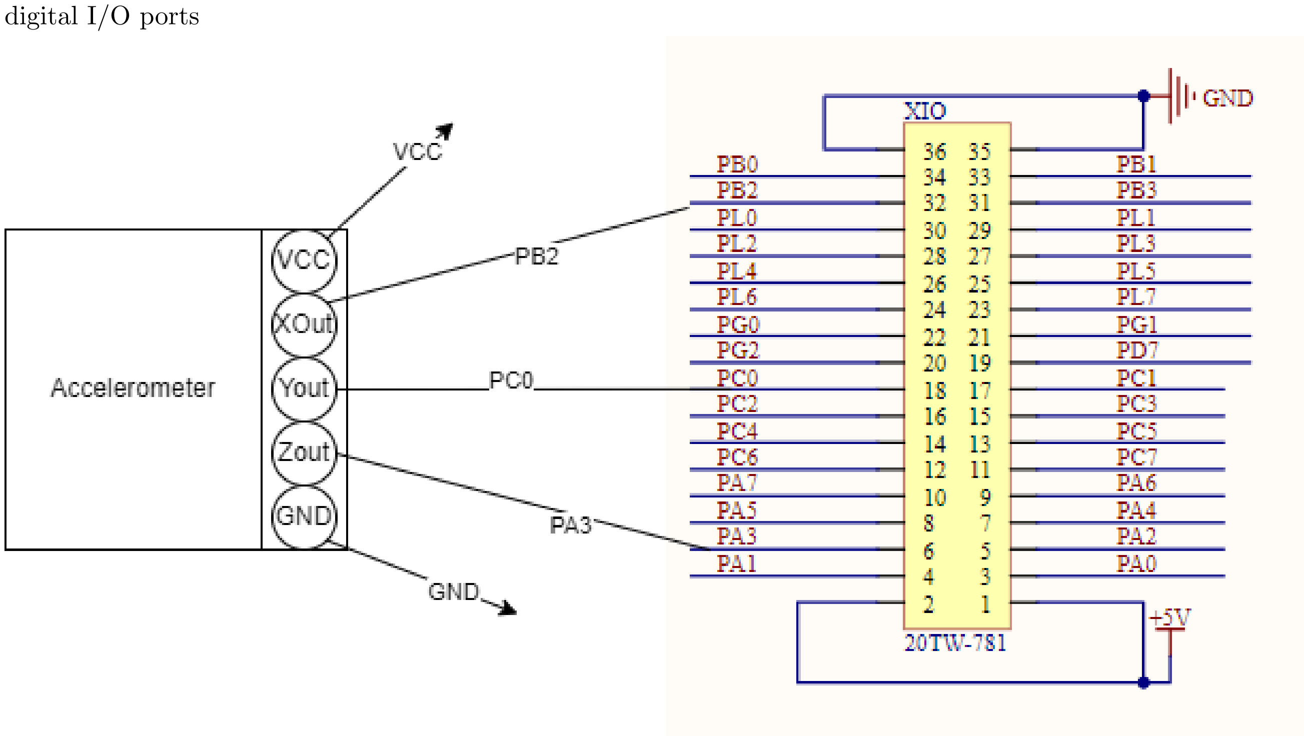


Figure 5: Connection of ADXL335 Accelerometer to the micro-controllers digital I/O ports

RPLidar A1M8 specification

Scanner Voltage: 4.9-5.5V
Motor System Voltage: 5-9V
Scanner Current: 300-350mA
Motor System Current: 100mA
Distance Range: 0.15-12m
Angular Range: 0-360°
Scan Field Flatness: -1.5°- 1.5°
Sample Duration: 0.125ms
Sample Frequency: max 8010Hz
Scan Rate: 1-10Hz, typical 5.5Hz
Laser Wavelength: 775-795nm, typical 785nm
Laser Power: typical 3mW, max 5mW
Pulse Length: typical 110us, max 300us
Band Rate: 115200bps
Weight: 170g
Temperature Range: 0-40°C

Arduino Mega 2560 Rev3 specification

Microcontroller: ATmega2560
Operating Voltage: 5V
Input Voltage: 7-12V
Digital I/O Pins: 54
PWM Input Pins: 15
Analog Input Pins: 16
DC Current per I/O Pin: 20mA
Flash Memory: 256KB
SRAM: 8KB
EEPROM: 4KB
UART: 4
Clock Speed: 16MHz
Weight: 37g
Width: 53.3mm
Length: 101.5mm

Ultra Sonic sensor Specification

Working Voltage: 5V(DC)
Static current: Less than 2mA
Output signal Electric frequency signal, high level 5V, low level 0V
Mode of connection 4 pins VCC, Trig (control side), echo (receiver), out (empty),GND
8 x HC-SR04 Distance Sensor Module

Accelerometer ADXL335 Specification

Measurement Range: 3.6g
Acceleration Rating: max 10,000g
Operating Voltage: 1.8-3.6V
Supply Current: 250μA
Bandwidth X_{OUT}, Y_{OUT} : 1600Hz

Bandwidth Z_{OUT}: 550Hz
Sensor Resonant Frequency: 5500Hz
Operating Temperature: -40-85 °C
Dimensions: 4×1.45mm

6.2 Object Avoidance System

Description

This module is where the cleaned data from the Data Collection module will be fused together (sensor fusion) to create one picture. Using this created environment picture, NextStep will determine if the user should change their path.

Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comment(s)
fused_distances	Float[ ]	[0-9*—.—0-9*5]	cm	Fused distance readings of detected objects.
user_velocity	Float	N/A	$\frac{cm}{second}$	Calculated velocity of the user, used to update the bubble boundary

Table 14: Object Avoidance System - Inputs

Outputs:

Output Name	Output Type	Range	Units	Comment(s)
rebound_angle	Float	$0 - \pi$	radians	Determines which haptic motor should buzz to direct a user safely around an obstacle.

Table 15: Object Avoidance System - Outputs

Exception Handling

Input Variable	Input Type	Exception	Exception Handling
user_velocity	Float	user_velocity = null	Accelerometer failure, use default velocity 100cm/s

Table 16: Object Avoidance System - Exception Handling

Timing Constraints

The time from the input of a new set of data to output of all variables (described above in the module's outputs) must be within time $t_{timeout}$.

Initialization

When the device powers on, all array variables initialize to empty arrays and all other variables initialize to null.

6.3 User Guidance Mechanism

Description

This module receives the necessary information from the Object Avoidance System to give the user guidance on how to get to their destination and avoid hitting obstacles. There are 5 haptic feedback motors on the user's head: center (forehead), left and right temple, and left and right side of the head. These haptic motors will only pulse if the user needs to change direction. The calculated rebound angle from the Object Avoidance System will be used to determine if a change is necessary. A haptic motor will only pulse if the user needs to change direction; therefore, if there are no objects in front of the user, all haptic motors will be still indicating the user should move forward in a straight line. If an object is detected, a light pulse will come from either the left or right temple motors (or from the left or right side of their head depending on how far they have to alter their course) to direct the user to turn to the left or right to avoid the detected obstacle. After they have turned far enough, a pulse will come from the center motor to indicate to the user to proceed in a straight line again.

Inputs and Outputs

Inputs:

Input Name	input Type	Range	Units	Comment(s)
rebound_angle	Float	$0 - \pi$	radians	Determines which haptic motor should buzz to direct a user safely around an obstacle.

Table 17: User Guidance Mechanism - Inputs

Outputs:

Output Name	Output Type	Range	Units	Comment(s)
vibration_sector	Integer	[0,4]	N/A	Sector for vibration warning.

Table 18: User Guidance Mechanism - Outputs

Exception Handling

Variable	Type	Exception	Exception Handling
rebound_angle	Float	*rebound_angle* = -1	User Rotate

Table 19: User Guidance Mechanism - Exception Handling

Timing Constraints

The time from the input of a new set of data to output of all variables (described above in the module's outputs) must be within time $t_{timeout}$.

Initialization

All variables initialized to null.

Diagrams

Haptic Motor Wiring Diagram

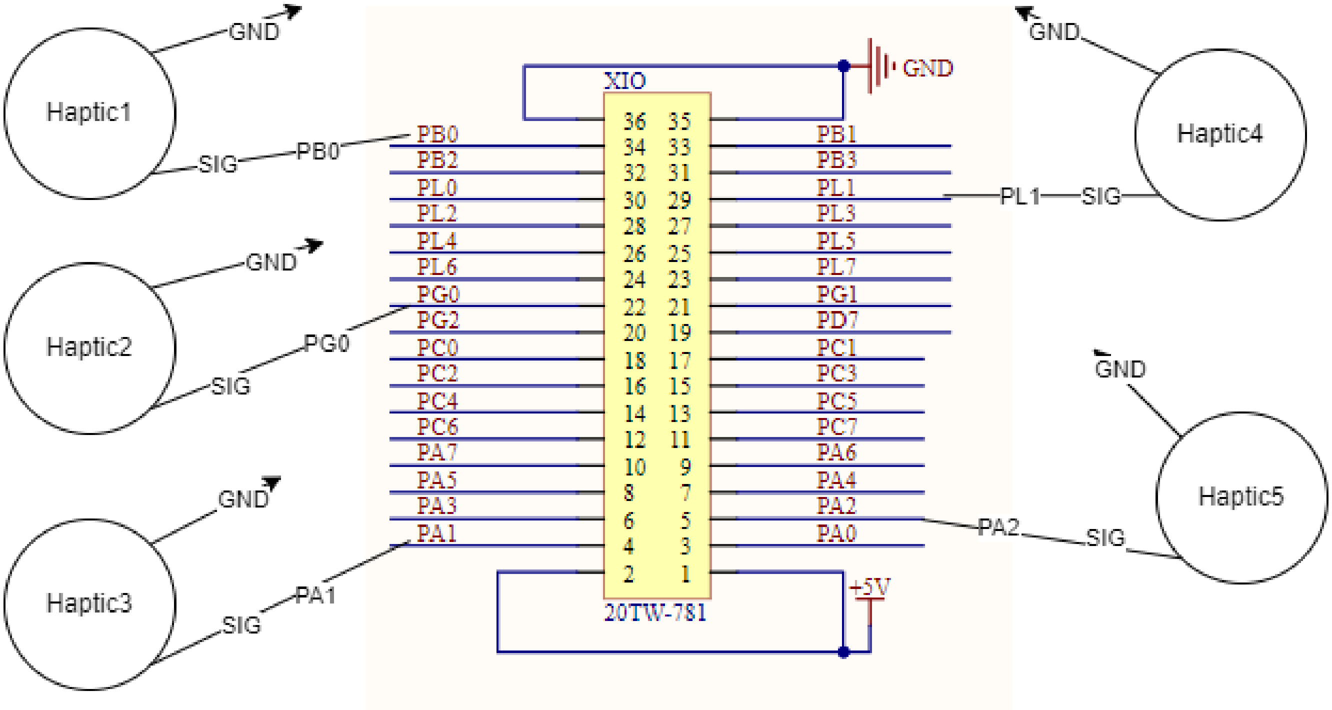


Figure 6: Wire diagram connection from digital I/O pins of Arduino Mega rev3 and the haptic motors

Motor Specification

Digi-Key Part Number 1597-1244-ND
Manufacturer Seeed Technology Co., Ltd
Manufacturer Product Number 316040001
Description VIBRATION ERM MOTOR 3V

6.4 User Input Module

Description

This module is responsible for collecting feedback from the user through the use of a button. The user will push the button according to instructions transmitted through a speaker. These button presses will be translated and passed to the system modules requiring the information.

Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comment(s)
input_field	String[]	[A-Za-z0-9]*	N/A	Inputs required from user.
user_button_input	Int	0,1	N/A	User pushing the button.

Table 20: User Inputs - Inputs

Outputs:

Output Name	Output Type	Range	Units	Comment(s)
user_input	String[]	N/A	N/A	User's input to required field.

Table 21: User Inputs - Outputs

Exception Handling

Variable	Type	Exception	Exception Handling
input_field	String[]	[A-Za-z0-9]*	N/A
user_button_input	Int	user_button_input = null	Button Failure

Table 22: User Inputs - Exception Handling

Timing Constraints

Timing constraints are based on receiving user's audio input from microphone within time $t_{timeout}$.

Initialization

All arrays variables upon system start up are initialized to an empty array and other variables initialized to null.

Speaker Diagram

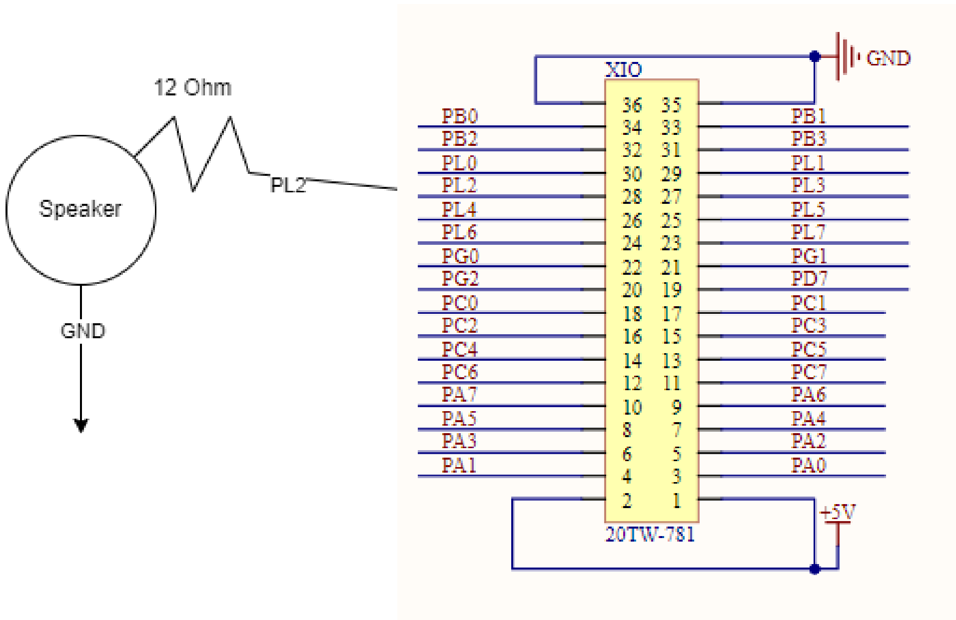


Figure 7: Wire diagram for the connection of the speaker used to the micro-controllers digital I/O port

Push Button

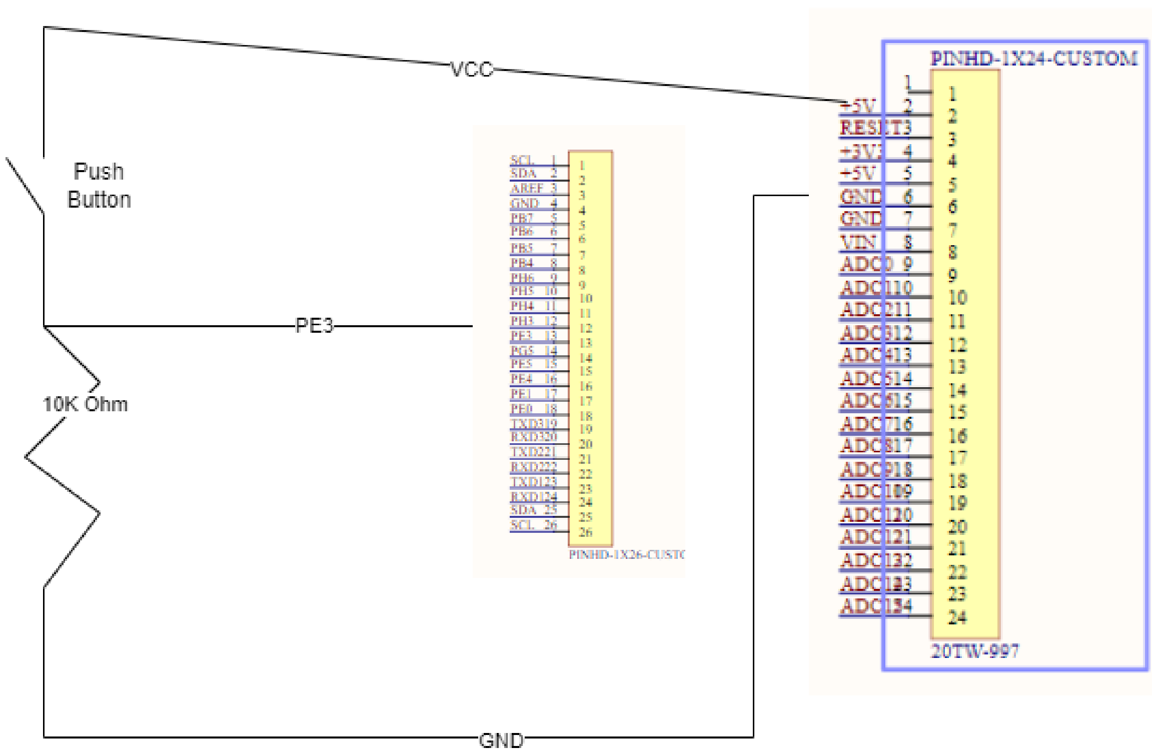


Figure 8: Wire diagram for the connection of the push button used to the micro-controllers digital I/O port

Speaker Specification

Speaker model: Visaton K15 S
Device resistance: 8 Ohms
Device power: 0.5 Watts
Resonance Frequency: 1000 Hz

Push Button Specification

Challenge Electronics Microphone Breakout
Voltage range: 2.7V up to 5.5V

6.5 Sensor Fusion Module

Description

This module uses the data gathered from the User Input Module to make sure that the sensors in the Data Collection Module know their relative location to the ground so that they can provide accurate readings. Secondly, this module provides the fusion of multiple sensors to provide accurate readings of distance to the obstacle avoidance.

Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comment(s)
user_information	String[]	[0-9, A-Z, a-z]*	N/A	User's information required to calibrate the sensors.
scanned_points[ ]	Float struct [ ]	(150-12000, 0-360)	(cm, deg)	Cleaned Distance and Angle of Lidar points
sensor_left	Float	0-500cm	cm	Cleaned Distance of a Detected Obstacle
sensor_right	Float	0-500cm	cm	Cleaned Distance of a Detected Obstacle
sensor_center_left	Float	0-500cm	cm	Cleaned Distance of a Detected Obstacle
sensor_center_right	Float	0-500cm	cm	Cleaned Distance of a Detected Obstacle
sensor_center	Float	0-500cm	cm	Cleaned Distance of a Detected Obstacle

Table 23: Sensor Fusion - Inputs

Outputs:

Output Name	Output Type	Range	Units	Comment(s)
calibration_status	boolean	[0,1]	N/A	Whether sensor calibration is successful or not.
calibration_error	String[]	[A-Za-z0-9]*	N/A	Errors encountered during calibration, empty if no errors.
fused_distance	float[]	[0-9*—.—0-9*5]	cm	the array containing the distances of each sensor at a specific time as well as the fused data between the Lidar and one ultrasonic sensor if ranges of angle intersect.

Table 24: Sensor Fusion - Outputs

Exception Handling

Variable	Type	Exception	Exception Handling
$user\_information$	String[]	[A-Za-z0-9]*	N/A
$fused\_distance$	int	[0-9]*	sensor_timeout and system diagnostic trigger for bad sensors
$scanned\_points$[ ]	Float struct [ ]	Out Of Bounds	Bad data received above 12000
$sensor\_left$	Float	Out Of Bounds	bad data received above 12000
$sensor\_right$	Float	Out Of Bounds	bad data received above 12000
$sensor\_center\_left$	Float	Out Of Bounds	bad data received above 12000
$sensor\_center\_right$	Float	Out Of Bounds	bad data received above 12000
$sensor\_center$	Float	Out Of Bounds	bad data received above 12000

Table 25: Sensor Fusion - Exception Handling

Timing Constraints

The sensors should be calibrated within $t_{timeout}$ time.

Initialization

Upon system startup, it will load previous calibration data if those exists, otherwise initialize to null. For others, all arrays variables upon system start up are initialized to an empty array and other variables initialized to null. The sensor objects shall also be initialized such that we can receive data from these objects.

6.6 System Diagnostics Module

Description

This module is responsible for relaying to the user information about the system. This information includes sensor malfunction notices and battery low notices. This information will be relayed to the user through speaker.

Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comment(s)
error_code	String	N/A	char	String containing the error code to be trigger.
sensor_id	int	[0-8]	N/A	Integer specifying which sensor the error occurred on.
confirmed	boolean	True,False	N/A	Boolean representing whether the error is confirmed or could be a false positive.
sensor_reading	float	N/A	cm	Sensor reading which will be used to check whether an error might have occurred.

Table 26: System Diagnostics Module - Inputs

Outputs:

Output Name	Output Type	Range	Units	Comment(s)
error_occurred	boolean	True,False	N/A	Boolean representing whether the sensor reading inputted contains an error or not.

Table 27: System Diagnostics Module - Outputs

Exception Handling

Variable	Type	Exception	Exception Handling
error_code	String	Non-Valid String	Ignore the call.
error_code	String	Doesn't match list of error code	Ignore the call.
sensor_id	int	Doesn't match list of sensor id	Ignore the call.

Table 28: System Diagnostics Module - Exception Handling

Timing Constraints

The time from the input of a new set of data to output of all variables (described above in the module's outputs) must be within time $t_{timeout}$.

Initialization

When the device powers on all variables initialize to null or 0 (depending on data type).

Diagrams

Refer to section 6.4 for the diagrams relevant to this section

7 Likelihood of Change

Module	**Likelihood of Change**	**Rationale**
Data Collection Module	Unlikely	Dependent on if sensors are added or removed from the final product.
Object Avoidance System	Unlikely	Dependent on if sensors are added or removed from the final product.
User Guidance Mechanism	Very Unlikely	Fundamental Component of the Product.
Sensor Fusion Module	Very Unlikely	Fundamental Component of the Product.
User Input Module	Very Unlikely	Fundamental Component of the Product.
System Diagnostics Module	Very Unlikely	Fundamental Component of the Product.

Table 29: Likelihood of Change for each component module

8 Normal Operation

NextStep is an assistive device for people with a visual impairment. It does not require frequent intervention by a user during its use unless there is a fault with the device. NextStep is meant to help a user navigate indoors around static objects or slower moving objects in their path.

NextStep is fashioned to a hat which will be worn on a user's head while it is on. When a user turns on the device for the first time, they will have to use the built in tactile button and speaker to provide their height so that the sensors can triangulate objects correctly.

Once the device is ready, the user can walk around their indoors environment and the device will detect objects in their path. NextStep will then determine the safest course of action the user must take to clear the path in their way. NextStep will identify an object and also communicate to the user the safest path they can take to avoid the object. The device relays this feedback to the user through small haptic feedback motors that contour the front of the forehead attached to the inner band of the hat. As the user moves to the left or the right away from an object, the vibration feedback from the motors shall stop until another object is detected in the users path.

NextStep runs off of an external power supply which is designed to guarantee at least three hours of use. The device can be safely charged by using any USB-C type cable to a power source.

9 Undesired Event Handling

NextStep can detect faults and undesired behaviours such as a sensor malfunction, low battery power or a vibration feedback motor malfunction. In the event of a low power warning, when $p_{battery}$ falls below 20%, the device will output a text-to-speech audio to warn the user that they will need to charge the device. In the event of a malfunction of one of the sensors or the vibration feedback module, the device will warn the user to not use the device using a text-to-speech audio warning.

10 References

Arduino Mega 2560 Rev3 data sheet
(1) "Mega 2560 Rev3: Arduino documentation," Arduino Documentation — Arduino Documentation. [Online]. Available: https://docs.arduino.cc/hardware/mega-2560. [Accessed: 21-Feb-2022

Accelerometer ADXL335 data sheet
(2) "ADXL335" [Online]. Available: https://www.analog.com/en/products/adxl335.html. [Accessed: 21-Feb-2022].

Microcontroller data sheet
(3) "ATmega640/V-1280/V-1281/V-2560/V-2561/V data sheet." [Online]. Available: https://content.arduino.cc/assets/ATmega640-1280-1281-2560-2561-Datasheet-DS40002211A.pdf. [Accessed: 21-Feb-2022].

RPLidar A1M8 data sheet
(4) "RPLidar A1: Introduction and Datasheet" [Documentation]. [Online]. Available: https://bucket-download.slamtec.com/d1e428e7efbdcd65a8ea111061794fb8d4ccd3a0/LD108_SLAMTEC_rplidar_datasheet_A1M8_v3.0_en.pdf. [Accessed: 21-Feb-2022].

Ultrasonic Sensors data sheet
(5) "HC-SR04 Ultrasonic Sonar Distance Sensor + 2 x 10K resistors data sheet." [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/Avago%20PDFs/ADNS-3050.pdf%3E. [Accessed: 20-Dec-2021].

Motor data sheet
(6) "Vibrating Motor Data Sheet" [Online]. Available: https://b.hatena.ne.jp/entry/s/media.digikey.com/pdf/Data%20Sheets/Laird%20Technologies/Antenna_Catalog_2007.pdf. [Accessed: 20-Dec-2021].

Speaker data sheet
(7) "K 15 S 8 ohm data sheet," Visaton K 15 S, 8 Ohm. [Online]. Available: https://www.visaton.de/en/products/drivers/miniature-speakers/k-15-s-8-ohm. [Accessed: 20-Dec-2021].

Microphone data sheet
(8) "Challenge Electronics: Omni-Directional Foil Electret Condenser Microphone," SparkFun. [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Sound/CEM-C9745JAD462P2.54R.pdf. [Accessed: 20-Dec-2021].