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
December 20, 2021

Contents

1 Revisions **3**

2 Overview **4**
2.1 Purpose . . . 4
2.2 Scope . . . 4
2.3 Context Diagram . . . 4
2.4 Component Diagram . . . 5
2.5 Assumptions . . . 5

3 System Variables **6**
3.1 Monitored and Controlled Variables . . . 6
3.2 Constants . . . 6

4 Behaviour Overview **6**

5 Component Traceability **7**

6 Component Overview **8**
6.1 Data Collection Module . . . 8
6.2 Object Avoidance System . . . 12
6.3 User Guidance Mechanism . . . 14
6.4 User Inputs . . . 15
6.5 Sensor Calibration . . . 18
6.6 System Diagnostics Module . . . 19

7 Likelihood of Change **20**

8 Normal Operation **20**

9 Undesired Event Handling **20**

10 References **21**

List of Tables

1 Revision History . . . 3
2 Monitored Variables Table . . . 6
3 Controlled Variables Table . . . 6
4 Constant Variables Table . . . 6
5 Component Traceability - Data Collection Module . . . 7
6 Component Traceability - Object Avoidance System . . . 7
7 Component Traceability - User Guidance Mechanism . . . 8
8 Component Traceability - User Inputs Module . . . 8
9 Component Traceability - Sensor Calibration Module . . . 8
10 Component Traceability - System Diagnostics Module . . . 8
11 Data Collection Module - Inputs . . . 9
12 Data Collection Module - Outputs . . . 9
13 Data Collection Module - Exception Handling . . . 9
14 Object Avoidance System - Inputs . . . 13
15 Object Avoidance System - Outputs . . . 13
16 Object Avoidance System - Exception Handling . . . 13
17 User Guidance Mechanism - Inputs . . . 14
18 User Guidance Mechanism - Outputs . . . 14

19 User Guidance Mechanism - Exception Handling . . . 14
20 User Inputs - Inputs . . . 16
21 User Inputs - Outputs . . . 16
22 User Inputs - Exception Handling . . . 16
23 Sensor Calibration - Inputs . . . 18
24 Sensor Calibration - Outputs . . . 18
25 Sensor Calibration - Exception Handling . . . 18
26 System Diagnostics Module - Inputs . . . 19
27 System Diagnostics Module - Outputs . . . 19
28 System Diagnostics Module - Exception Handling . . . 19
29 Likelihood of Change for each component module . . . 20

List of Figures

1 Context Diagram of NextStep . . . 4
2 Component Diagram of the NextStep System . . . 5
3 SPI Connection Between Arduino (ATmega4809) rev2 WIFI and the Pixy 2.0 . . . 10
4 Connection of 3 ultra sonic sensors to track z,y,z coordinates of object to the micro-controllers digital I/O ports . . . 11
5 Wire diagram connection from digital I/O pins of Arduino rev2 WIFI and the haptic motors 15
6 Wire diagram for the connection of the speaker used to the micro-controllers digital I/O port 17
7 Wire diagram for the connection of the microphone used to the micro-controllers analog I/O port . . . 17

1 Revisions

Revision Number	Date	Reason for Change
Revision 0	December 20, 2021	N/A

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 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 camera and ultra-sonic 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

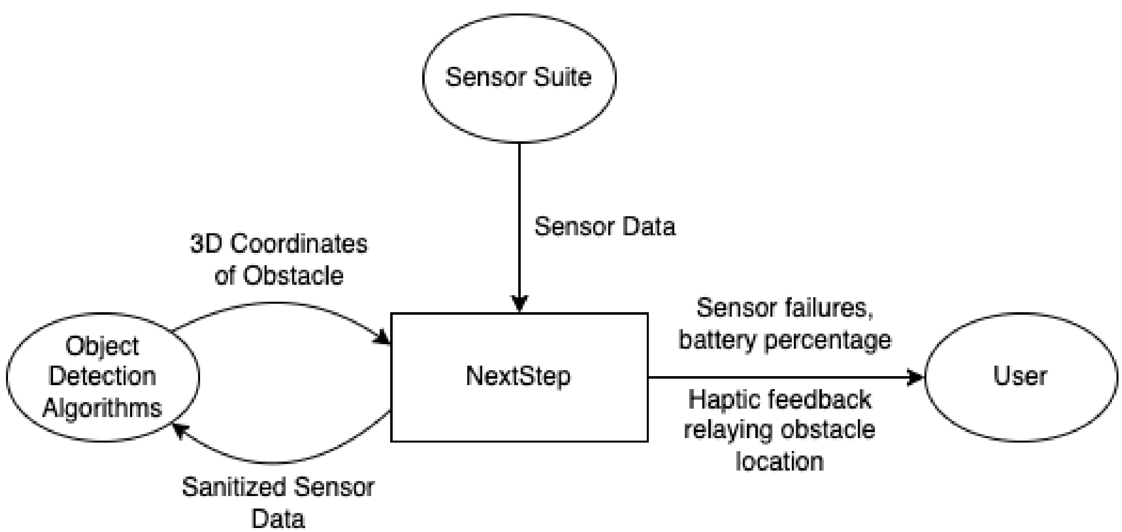

Figure 1: Context Diagram of NextStep

2.4 Component Diagram

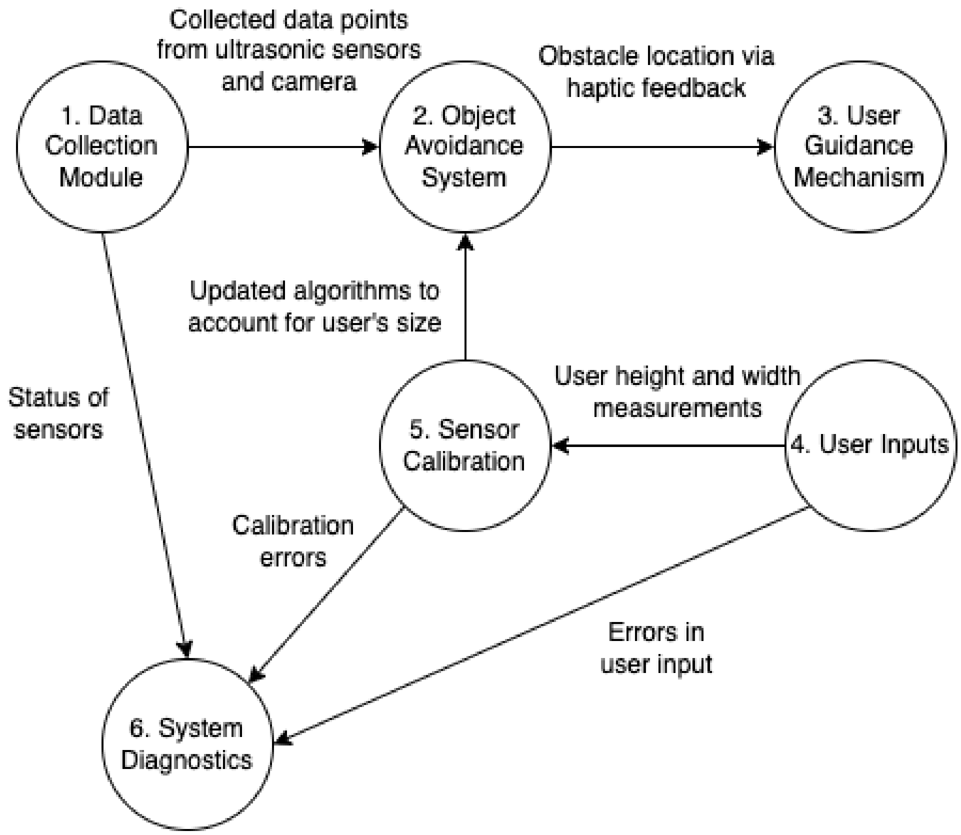

Figure 2: Component Diagram of the NextStep System

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	The indoor setting will be one that is not dark.
Rationale	The camera used to collect information about obstacles will not be able to function properly in the dark.

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 camera 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 camera resulting data being accumulated. 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
$d_{obstacle}$	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.
$v_{obstacle}$	Velocity[ ]	m/s	Array of relative velocities of the obstacles.
$p_{battery}$	Percentage	%	Percentage of battery life remaining.

Table 2: Monitored Variables Table

Controlled 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
$width$	Width	m	The width of the user
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	Type	Value	Unit	Description
$\text{max}_{distance}$	Distance	4	m	The maximum working distance of the ultrasonic sensors.
max_{angle}	Angle	15	degree	The maximum angle of observation of the ultrasonic sensors.
num_{sector}	Integer	TBD	N/A	Number of sectors of vibration warning to user.
$\text{max}_{closeness}$	Integer	TBD	N/A	The maximum level of closeness for vibration warning to user.
max_{height}	Integer	2.13	m	The maximum height of obstacles that NextStep will warn users about.
$\text{t}_{timeout}$	Float	TBD	seconds	The maximum time between sensor readings.

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 camera. This component will communicate this information with the Object Avoidance System.

2. **Object Avoidance System** This component receives sensor data from the Data Collection component and then uses sensor fusion algorithms to combine this data to form one coherent image of the environment.

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 Inputs** Responsible for collecting user height and width to pass on to the Sensor Calibration.

5. **Sensor Calibration** Using the data gathered from the User Inputs component to make sure that the ultrasonic sensors and the camera know their relative location to one another and the ground. This will enable them to provide accurate data points for the sensor fusion algorithms.

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 FR7 FR17 FR18 FR22 FR23 FR24 FR25 PR5 PR7 PR8

Table 5: Component Traceability - Data Collection Module

Component Module	Functional and Non-Functional Requirement
Object Avoidance System	FR1 FR2 FR9 FR14 FR17 FR18 FR25 PR5 PR8

Table 6: Component Traceability - Object Avoidance System

Component Module	Functional and Non-Functional Requirement
User Guidance Mechanism	FR2 FR3 FR7 FR9 FR15 FR17 FR19 FR24 FR25 PR1 PR8

Table 7: Component Traceability - User Guidance Mechanism

Component Module	Functional and Non-Functional Requirement
User Input Module	FR12 FR13 FR20 PR2

Table 8: Component Traceability - User Inputs Module

Component Module	Functional and Non-Functional Requirement
Sensor Calibration Module	FR12 FR14 PR9 OE2

Table 9: Component Traceability - Sensor Calibration Module

Component Module	Functional and Non-Functional Requirement
System Diagnostics Module	FR4 FR14 FR20

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 camera 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)
$camera_x$	Unsigned Integer Camera Input	N/A	cm	x Coordinate of Object in User's Path
$camera_y$	Unsigned Integer Camera Input	N/A	cm	y Coordinate of Object in User's Path
$sensor\_bottom\_left$	Float Sensor Input	N/A	cm	Distance of Object in User's Path
$sensor\_bottom\_right$	Float Sensor Input	N/A	cm	Distance of Object in User's Path
$sensor\_top$	Float Sensor Input	N/A	cm	Distance of Object in User's Path
$user\_acceleration_x$	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)
$sensor\_bottom\_left$	Float	N/A	cm	Cleaned Distance of a Detected Obstacle
$sensor\_bottom\_right$	Float	N/A	cm	Cleaned Distance of a Detected Obstacle
$sensor\_top$	Float	N/A	cm	Cleaned Distance of a Detected Obstacle
$user\_acceleration\_$ coordinates[]	Array of Floats	N/A	$\frac{cm}{second^2}$	Coordinates of User's Acceleration in [x-plane, y-plane, z-plane]
$camera\_coordinates$[]	Array of Unsigned Integer Camera Inputs	N/A	cm	x [x,y] Coordinates of Object in User's Path

Table 12: Data Collection Module - Outputs

Exception Handling

Input Name	Input Type	Exception	Exception Handling
$sensor\_bottom\_left$	Float Sensor Input	$sensor\_bottom\_left$ = null	Sensor Failure
$sensor\_bottom\_right$	Float Sensor Input	$sensor\_bottom\_right$ = null	Sensor Failure
$sensor\_top$	Float Sensor Input	$sensor\_top$ = 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
$camera_x$	Unsigned Integer Camera Input	$camera_x$ = null	Camera Failure
$camera_y$	Unsigned Integer Camera Input	$camera_y$ = null	Camera 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

Serial Peripheral Interface (SPI) Camera Module Connection to Arduino SPI Port

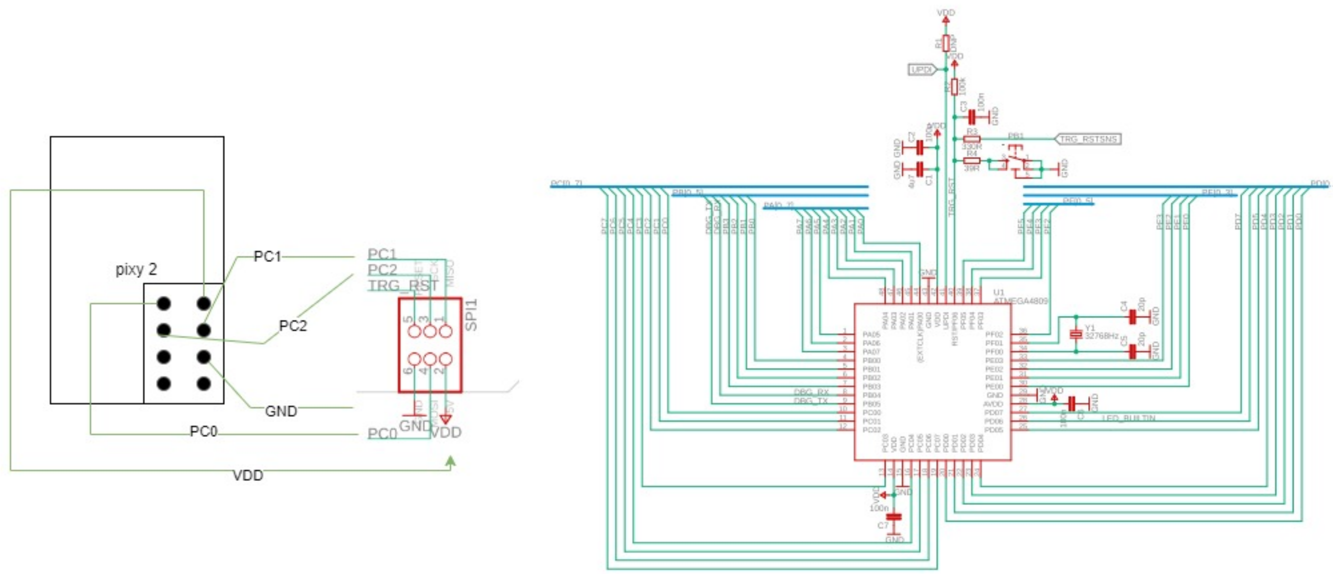

Figure 3: SPI Connection Between Arduino (ATmega4809) rev2 WIFI and the Pixy 2.0

Ultra Sonic Sensors Wiring Diagram

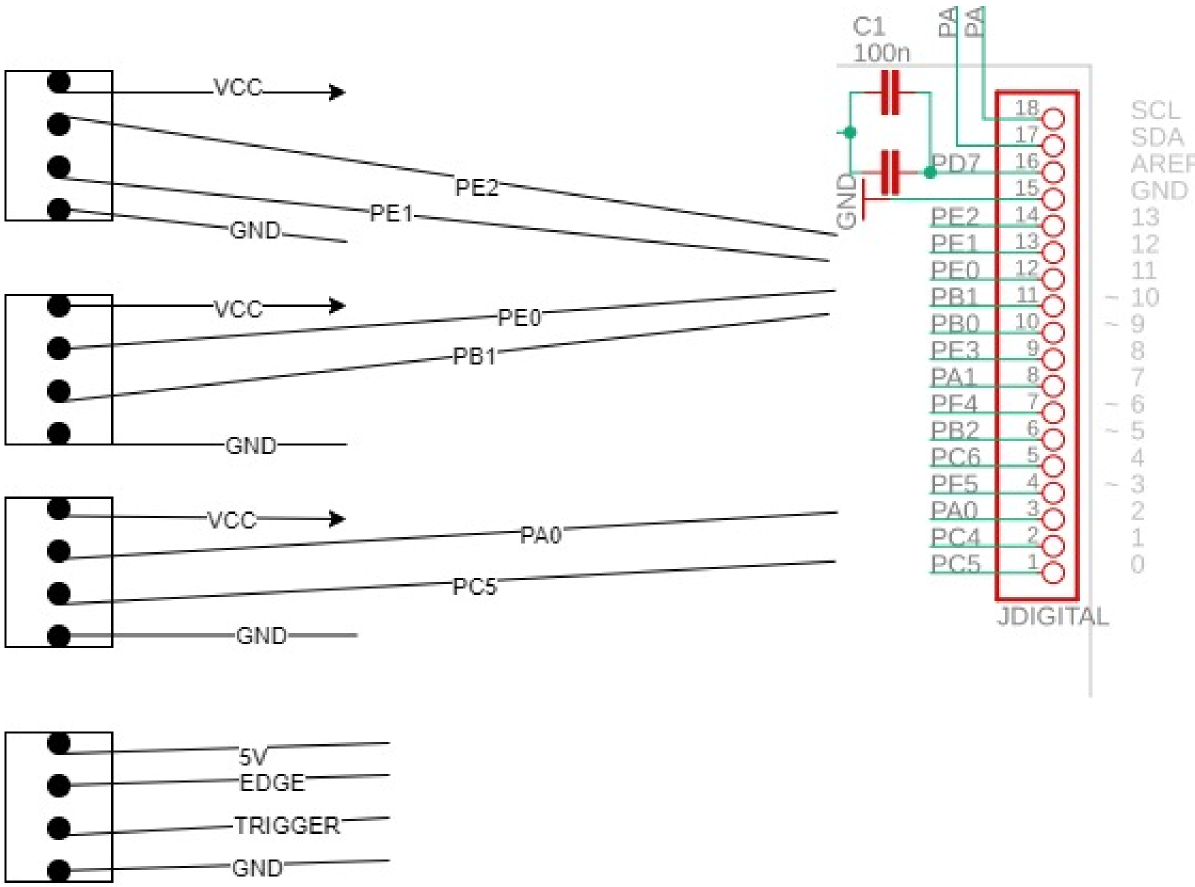

Figure 4: Connection of 3 ultra sonic sensors to track z,y,z coordinates of object to the micro-controllers digital I/O ports

Pixy2.0 specification

Processor: NXP LPC4330, 204 MHz, dual core
Image sensor: Aptina MT9M114, 1296×976 resolution with integrated image flow processor
Lens field-of-view: 60 degrees horizontal, 40 degrees vertical Power consumption: 140 mA typical
Power input: USB input (5V) or unregulated input (6V to 10V)
RAM: 264K bytes
Flash: 2M bytes
Available data outputs: UART serial, SPI, I2C, USB, digital, analog
Integrated light source, approximately 20 lumens
Dimensions: 1.5” x 1.65” x 0.6”
Weight: 10 grams

Arduino Specification

Microcontroller: ATmega4809
Operating Voltage: 5V
Input Voltage (Recommended): 7 - 12V
Digital I/O Pins: 14 — 5 Provide PWM Output
PWM Digital I/O Pins: 5
Analog Input Pins: 6
DC Current Per I/O Pin: 20 mA
DC Current For 3.3V Pin: 50 mA
Flash Memory: 48 KB (ATmega4809)
SRAM: 6,144 Bytes (ATmega4809)
EEPROM: 256 Bytes (ATmega4809)
Clock Speed: 16 MHz
Radio Module: u-blox NINA-W102
Secure Element ATECC608A
Intertial Measurement Unit: LSM6DS3TR
LED_BUILTIN 25
Length: 68.6 mm
Width: 53.4 mm
Weight: 25 g

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
3 x HC-SR04 Distance Sensor Module

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, this module will determine if the user needs to move to avoid an object.

Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comment(s)
sensor_bottom_left	Float	N/A	cm	Cleaned Distance of a Detected Obstacle
sensor_bottom_right	Float	N/A	cm	Cleaned Distance of a Detected Obstacle
sensor_top	Float	N/A	cm	Cleaned Distance of a Detected Obstacle
*user_acceleration_* coordinates[]	Float[]	N/A	$\frac{cm}{second^2}$	Coordinates of User's Acceleration in [x-plane, y-plane, z-plane]
camera_coordinates	Array of Unsigned Integer Camera Inputs	N/A	cm	x [x,y] Coordinates of Object in User's Path

Table 14: Object Avoidance System - Inputs

Outputs:

Output Name	Output Type	Range	Units	Comment(s)
$obstacle\_coord_x$	Float[]	N/A	cm	Computed x Coordinate of Obstacle in Array
$obstacle\_coord_y$	Float[]	[0, $max\_distance$]	cm	Computed y Coordinate of Obstacle in Array
$obstacle\_coord_z$	Float[]	[0, max_height]	cm	Computed z Coordinate of Obstacle in Array
object_detected	Boolean	[0,1]	N/A	True if an Object is in the User's Path, else False
v_{self}	Float[]	N/A	m/s	The User's Speed in [x-plane, y-plane]

Table 15: Object Avoidance System - Outputs

Exception Handling

Input Variable	Input Type	Exception	Exception Handling
sensor_bottom_left	Float	N/A	N/A
sensor_bottom_right	Float	N/A	N/A
sensor_top	Float	N/A	N/A
*user_acceleration_* coordinates	Float[]	N/A	N/A
camera_coordinates	Array of Unsigned Integer Camera Inputs	N/A	N/A

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 them guidance on how to get to their destination and avoid hitting obstacles. It takes the coordinates of the list of potential obstacles and the velocity of the user, calculate the obstacles that user needs to avoid and relay it to user via vibration. There are 5 haptic feedback motors on the user's head: center (forehead), left and right temple, left and right side of head. If there are no objects in the user's path, a light pulse will come from the forehead motor indicating to the user that they should walk in a straight line. If 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 move) to direct the user to move to the left or right to avoid the detected obstacle. The intensity of the feedback will increase based on the user's increasing distance to impact, indicating the urgency of the required movement.

Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comment(s)
$object\_coord_x$	Float[]	N/A	cm	Computed x Coordinate of Obstacle in Array
$object\_coord_y$	Float[]	[0, $max\_distance$]	cm	Computed y Coordinate of Obstacle in Array
$object\_coord_z$	Float[]	[0, max_height]	cm	Computed z Coordinate of Obstacle in Array
$object\_detected$	Boolean	[0,1]	N/A	True if an Object is in the User's Path, else False
v_{self}	Float[]	N/A	m/s	The User's Speed in [x-plane, y-plane]

Table 17: User Guidance Mechanism - Inputs

Outputs:

Output Name	Output Type	Range	Units	Comment(s)
$vibration\_sector$	Integer[]	[0, max_sector]	N/A	Sector for vibration warning.
$vibration\_strength$	Integer[]	[0, $max\_strength$]	N/A	Strength of vibration for each sector.

Table 18: User Guidance Mechanism - Outputs

Exception Handling

Variable	Type	Exception	Exception Handling
$object\_coord_x$	Float	N/A	N/A
$object\_coord_y$	Float	$object\_coord_y$ outside of range	Object Range Regulation
$object\_coord_z$	Float	$object\_coord_z$ outside of range	Object Range Regulation
$object\_detected$	Boolean	[0,1]	N/A
v_{self}	Float[]	N/A	N/A

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

Diagrams

Haptic Motor Wiring Diagram

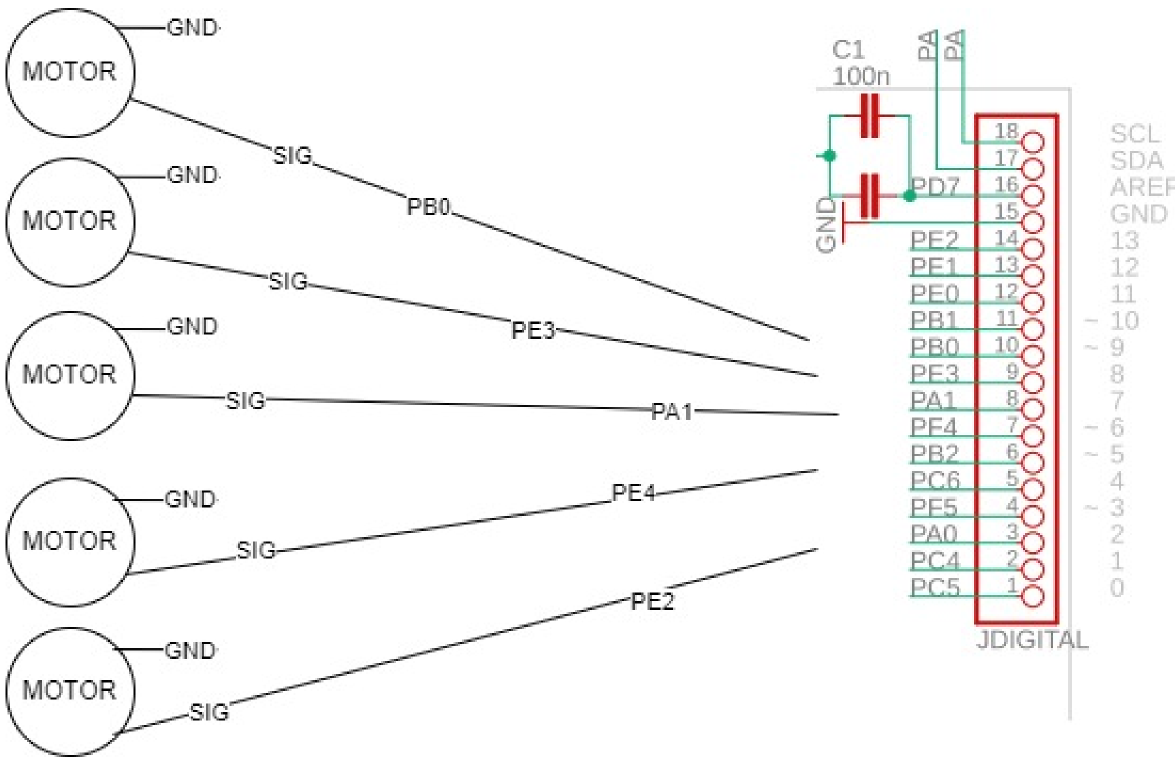

Figure 5: Wire diagram connection from digital I/O pins of Arduino rev2 WIFI and the haptic motors

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 Inputs

Description

This module is responsible for collecting information about the user, transforming the audio input to text and passing the gleaned information on to other modules, such as Sensor Calibration module.

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_audio_input	Audio User Input	N/A	N/A	Audio of user's input for required field.

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_audio_input	Audio User Input	user_audio_input = null	Microphone 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

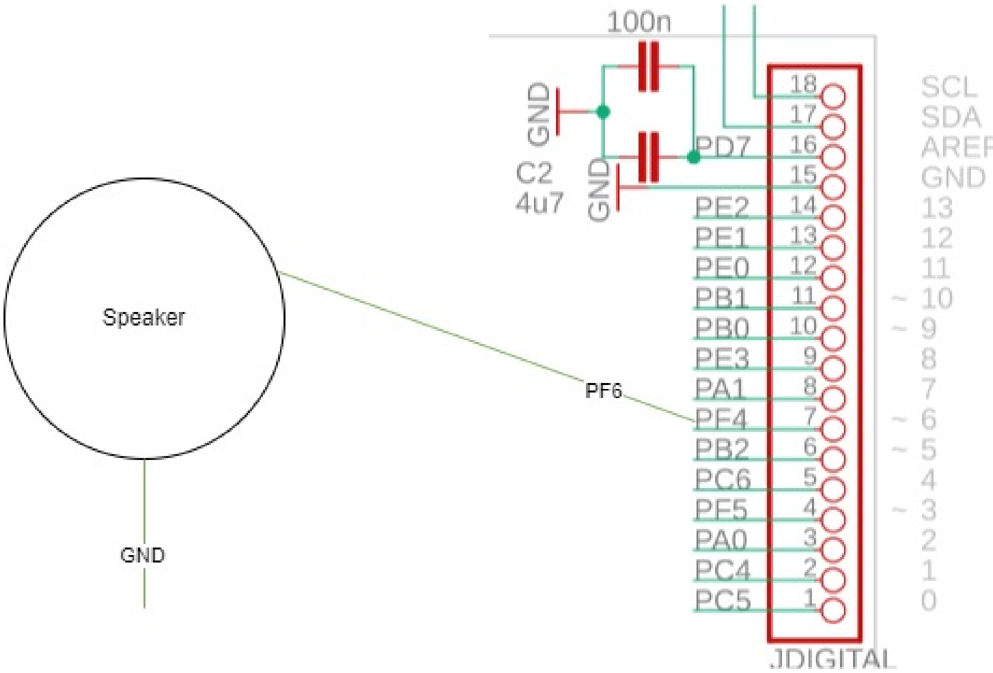

Figure 6: Wire diagram for the connection of the speaker used to the micro-controllers digital I/O port

Microphone Diagram

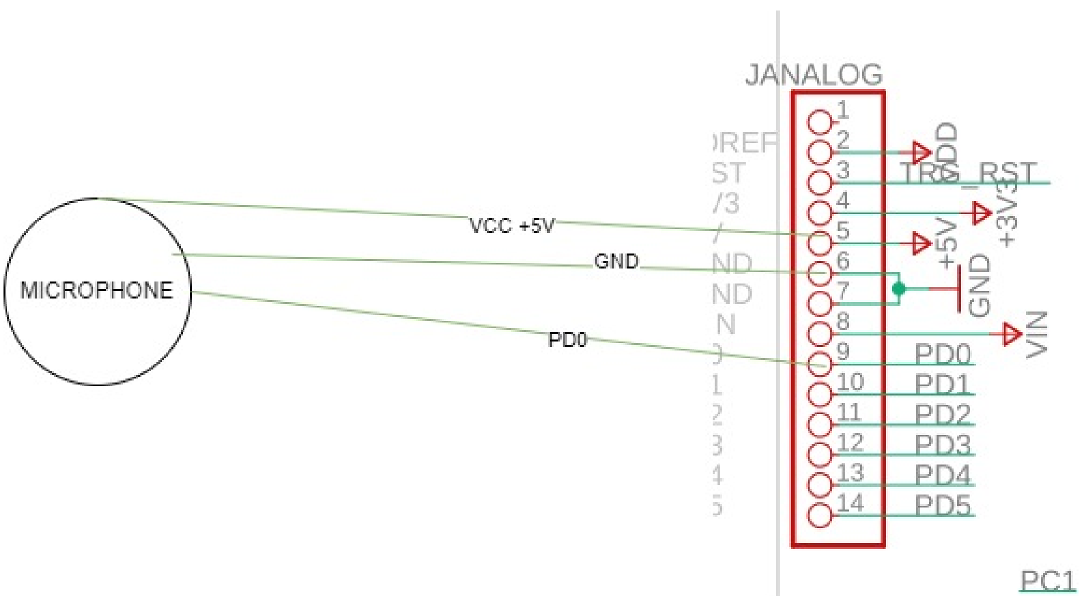

Figure 7: Wire diagram for the connection of the microphone used to the micro-controllers analog I/O port

Speaker Specification

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

Microphone Specification

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

6.5 Sensor Calibration

Description

This module uses the data gathered from the User Inputs module to make sure that the sensors in the Data Collection module know their relative location to each other and the ground so that they can provide accurate readings.

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.

Table 23: Sensor Calibration - 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.

Table 24: Sensor Calibration - Outputs

Exception Handling

Variable	Type	Exception	Exception Handling
$user\_information$	String[]	[A-Za-z0-9]*	N/A

Table 25: Sensor Calibration - 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.

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 audio (i.e. a speaker).

Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comment(s)
trigger	String	N/A	char	String containing which system notice to trigger.

Table 26: System Diagnostics Module - Inputs

Outputs:

Output Name	Output Type	Range	Units	Comment(s)
text_to_speech	String	N/A	char	String of .wav file to be sent to speaker.

Table 27: System Diagnostics Module - Outputs

Exception Handling

Variable	Type	Exception	Exception Handling
trigger	String	Non-Valid String	Trigger Error

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.

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	Very Unlikely	Fundamental Component of the Product.
Object Avoidance System	Unlikely	Dependent on if sensors are added or removed from the final product.
User Guidance Mechanism	Moderately Likely	Dependent on if a user can understand the vibrational feedback and if it is accurate enough on its own.
User Input Module	Very Unlikely	Fundamental Component of the Product.
Sensor Calibration Module	Unlikely	Dependent on if sensors are added or removed from the final 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 microphone and speaker to provide their height and width dimensions for the sensors to be calibrated properly.

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 vibration motors that contour the front of the forehead. As the user gets closer to an object the vibration shall increase in intensity. As the user moves to the left or right, the vibration feedback from the motors shall move towards the center of the forehead as the user follows the route that NextStep was mapped.

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 10%, 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 Uno Wifi Rev2 data sheet

(1) "Uno WIFI Rev 2: Arduino documentation," Arduino Documentation — Arduino Documentation. [Online]. Available: https://docs.arduino.cc/hardware/uno-wifi-rev2. [Accessed: 20-Dec-2021

Microcontroller data sheet

(2) "ATMEGA4808/4809 data sheet." [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega4808-4809-Data-Sheet-DS40002173A.pdf. [Accessed: 20-Dec-2021].

Pixy cam data sheet

(3) "Pixycam Documentation," wiki:v2:porting_guide [Documentation]. [Online]. Available: https://docs.pixycam.com/wiki/doku.php?id=wiki%3Av2%3Aporting_guide. [Accessed: 20-Dec-2021].

Ultrasonic Sensors data sheet

"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

(5) "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

(6) "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

(7) "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].