School of Engineering Science ▪ Burnaby, BC ▪ V5A 1S6
http://www.drivomatic.com
Drivomatic Technology Corporation

March 3, 2008

Dr. Patrick Leung
School of Engineering Science
Simon Fraser University
Burnaby, BC V5A 1S6

Re: ENSC 440 Project Design Specifications for a Self-Driving Wheelchair System

Dear Dr. Leung:

Enclosed is the Design Specifications document for a Self-Driving Wheelchair System from Drivomatic Technology Corporation. We are focused on developing a self-driving wheelchair system that allows the mobility impaired to navigate within buildings such as airports or nursing homes with maximum independence.

The design specification content in this document is only applicable to the development of the prototype stage. However, possible designs for the production development stage will also be discussed. In addition, future improvements to the Self-Driving Wheelchair System project will be proposed.

Drivomatic Technology Corporation consists of five team motivated engineering students: Jonathan Hung, Andy Chen, Jian Guo, Benjamin Chang, and Ammar Zaidi. If you have any questions or concerns about the design specifications, please do not hesitate to contact me by phone at (604) 721-0585 or via e-mail at ensc440-spring08-dtc-ensc@sfu.ca.

Sincerely,

Jonathan Hung
President and CEO
Drivomatic Technology Corporation

Enclosure: *Design Specifications for a Self-Driving Wheelchair System*

Drivomatic Technology Corp.

Design Specifications for a
Self-Driving Wheelchair

Project Team: Jonathan Hung
Andy Chen
Jian Guo
Benjamin Chang
Ammar Zaidi

Contact Person: Jonathan Hung
jwhung@sfu.ca

Submitted to: Patrick Leung – ENSC 440
Steve Whitmore – ENSC 305
School of Engineering Science
Simon Fraser University

Issued date: March 3, 2008

Revision: 1.1

EXECUTIVE SUMMARY

Drivomatic Technology Corporation's Mission:

"To provide innovative solutions for the mobility impaired to increase the individuals' independence and security in everyday life."

The design specifications for the Self-Driving Wheelchair (SDW) System provide a set of detailed descriptions of the designs involved in developing the prototype model. The functional requirements, denoted by type A and type B, outlined in the document *Functional Specifications for a Self-Driving Wheelchair* [1], will be addressed in this document in detail. Designs for the production model will also be discussed; however, specifications will not be included.

This document outlines the design for the SDW System and provides justifications for our design choices. To begin with, the high level overall system design is be discussed, which includes the hardware designs and electrical systems designs that will be used for the project Design specifications for the individual subsystems, which includes, Sensor System Design, Navigation System Design, and Wheelchair Control Design, will be discussed. The design for the user interface will include the prototype and production models.

Additional topics such as the System Test Plan will be explained in detail. Engineers in the Quality Assurance team will follow this test plan to certify that the system is functioning as intended. A discussion on future development will also be included.

To conclude the design specifications for the SDW System, possible future R&D projects will be discussed in the Future Development System section. Future R&D projects may include an Anti-Collision System and a Trackless Auto Navigation System.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

The Self-Driving Wheelchair (SDW) System is the control module that integrates with an electric wheelchair to enable auto-navigation on pre-defined maps in buildings. By sensing the current position within a floor, the SDW system intelligently transports the user to a desired location through a calculated path.

1.1 Scope

This document specifies the design of the SDW System and explains how the design meets the functional requirements as described in the *Functional Specification for a Self-Driving Wheelchair* [1] document. Since the design specifications are focused on the prototype function requirements, only the requirements denoted by A and B in the functional specification document will be discussed. The design specification includes all requirements and the implementation methods for a prototype system. The appendices include electric circuit schematics, track designs and process flowcharts to help facilitate the implementation of the SDW System.

1.2 Intended Audience

The design specification is intended for use by all members of Drivomatic Technology Corporation. It shall serve as an overall design guideline to ensure all the functions are implemented and all the requirements are met in the final product. Members in the Quality Assurance team will follow the test plans outlined in this document to verify functionalities of the product.

1.3 Glossary

Pulse Width Modulation	control power sent through the modulation of the duty cycle[7]
PWM	Pulse Width Modulation
RFID	Radio Frequency Identification
SDW	Self-Driving Wheelchair
MIMO	Multiple Input Multiple Output
UART	Universal Asynchronous Receiver/Transmitter
GPIO	General Purpose Input/Output

2 SYSTEM SPECIFICATIONS

The main function of the SDW System is to transport the user safely to a user defined destination. To begin operation of the system, the user must switch the system ON by means of an ON/OFF switch. The system then finds the closest track and moves on to the track. Now in idling mode, the SDW System will wait for an input from the user. As soon as the user selects a destination, the system will control the wheelchair and transport the user to the desired destination through a calculated path. In case of an invalid input from the user, the SDW System will remain in idling mode keeping the wheelchair stationary. In emergency situations, the user has a system cut-off switch on the user interface to disengage the system and thus avoiding accidents.

3 Overall System Design

This section provides a high level overview of the entire system design. Individual subsystems will be discussed, giving details on how they are integrated with one another. Subsequently, the physical design of our system and the component mounting position limitations will also be discussed. Specific details on the subsystem designs are discussed in ***Sections 4 to 7***.

The SDW System will utilize a line following method to navigate within a building. The system function diagram is shown in ***Figure 3-1***.

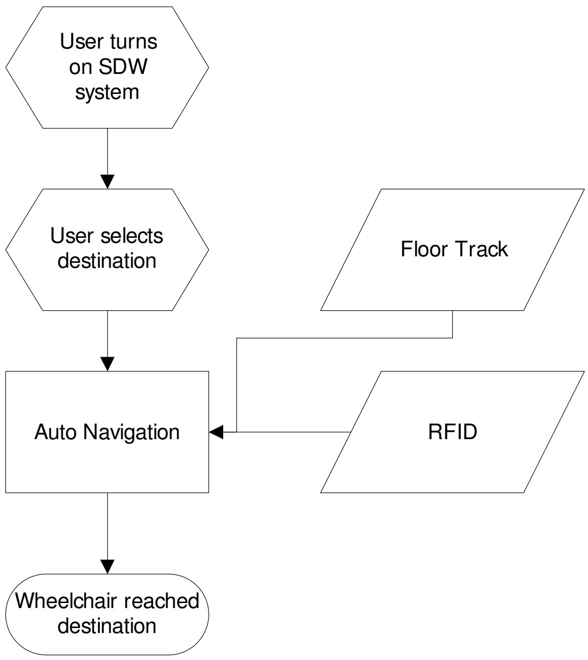


Figure 3-1: SDW System Function Diagram

3.1 High-level System Design

The SDW system is a MIMO system. Input parameters include outputs from sensors and user inputs.

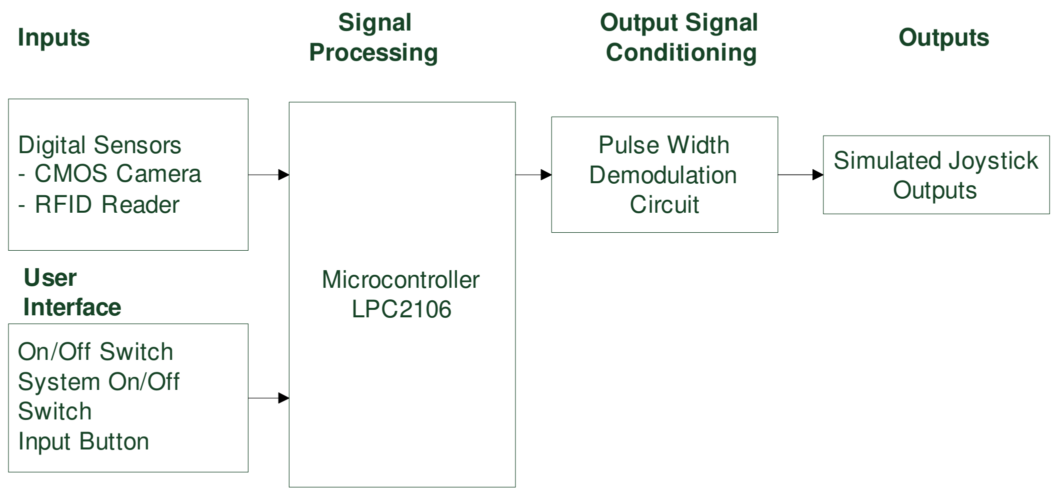


Figure 3-2: High Level System Design Block Diagram

3.2 Hardware Design

3.2.1 Sensor

The main function requirement for the SDW System is to be able to follow a line on the floor and be able to identify different locations. A number of solutions for the line following system have been researched, such as using photo-sensors and visual sensors. Ultimately, video sensors ruled out the photo-sensors because video processing gives the system the capability of making smart decisions with greater flexibility. ***Section 4.1***, discusses the design for the visual sensor in more detail.

Solutions to location identification that have been explored included bar code scanning, visual processing, and radio frequency identification (RFID). The design criteria was to find a method that requires minimum modifications to its environment and has a balance between reliability and cost. Using visual processing would require minimal modifications, but rigorous computation is needed. In addition, the greatest limitation to visual processing is the degree of artificial intelligence that it can achieve. In contrary, the technology for RFID has been advancing at rapid rate and the cost of using them has greatly fallen. Hence, RFID is the more desirable approach for location identification. ***Section 4.2***, discusses the design for the RFID system in more detail.

3.2.2 Microcontroller

The CMUCam3 is an ARM7TDMI based fully programmable embedded computer vision sensor [2] designed by Carnegie Mellon University. This sensor is ideal for this project because it contains four servo control outputs capable of controlling the wheelchair, and is equipped with sufficient GPIO ports capable of integrating external devices such as a RFID reader. In addition, the CMUCam3 has a built in MMC Flash slot, which could be used to store mapping information and an RFID table.

3.3 Firmware Design

Figure 3-3 illustrates the SDW System's high level firmware design.

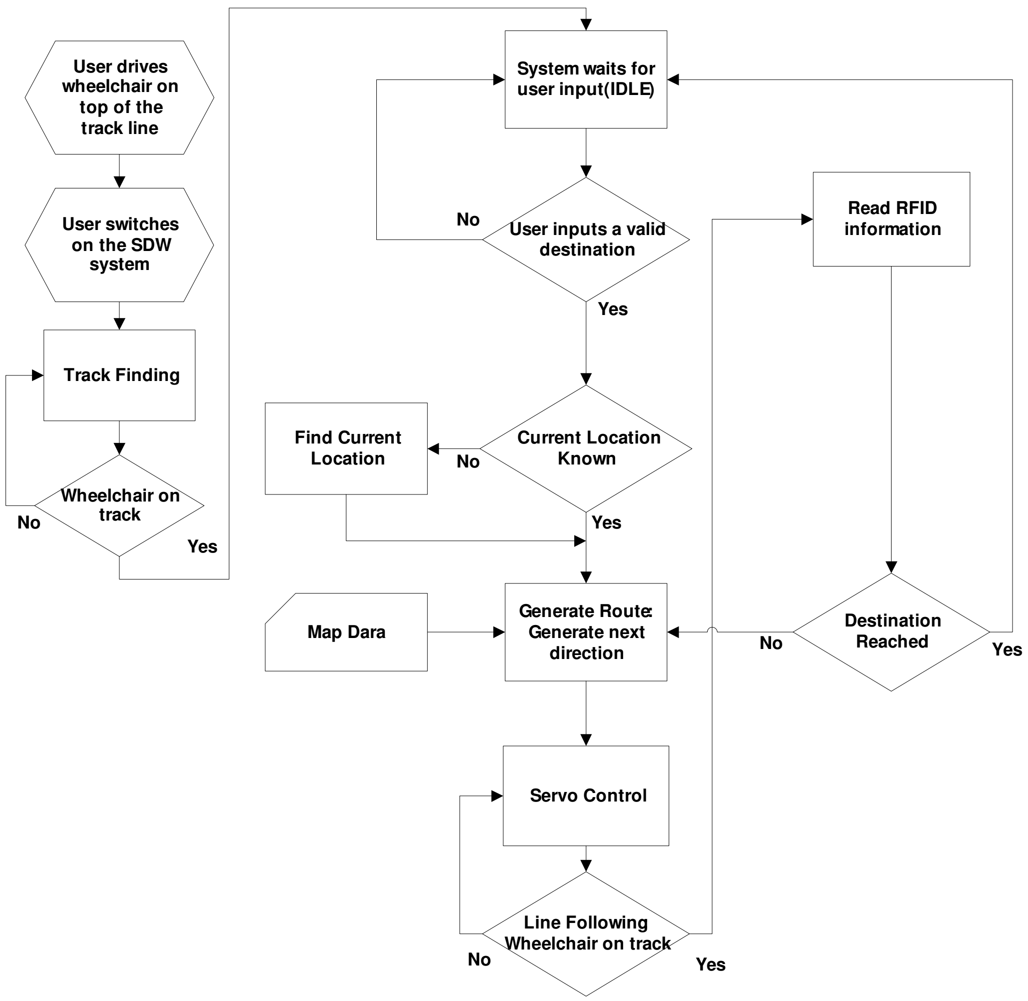


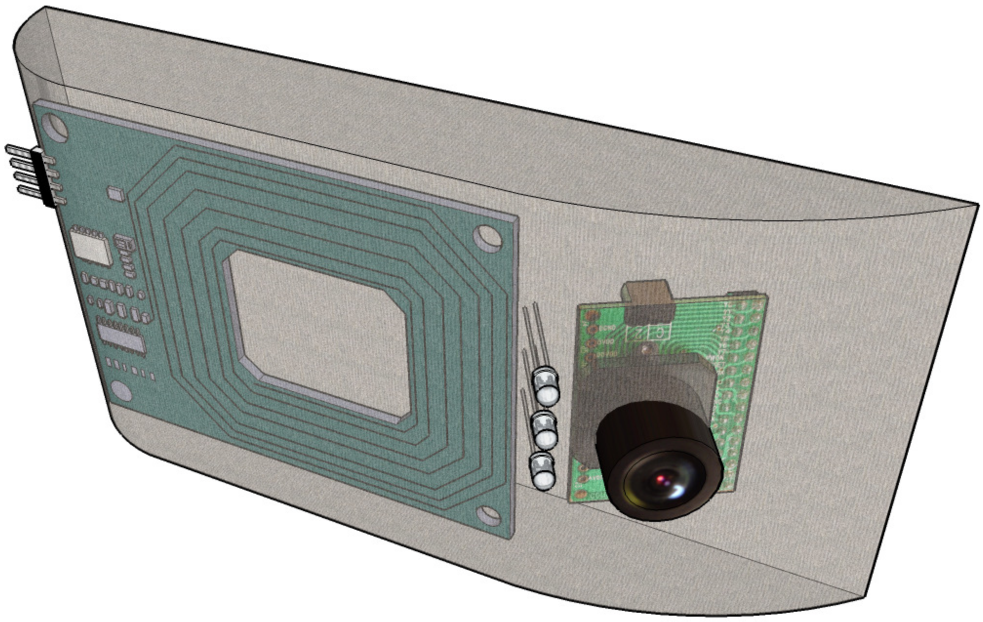
Figure 3-3: Firmware Block Diagram

3.4 Physical Design

The SDW System consists of two devices that are mountable onto the user's existing electric wheelchair. The first device is the user interface, containing the CMUCam3 microprocessor, and will be denoted as the main module throughout this document. The second device contains sensors that read information from the environment, such as the CMUCam3's camera module and the RFID reader; this device will be denoted as the sensor module throughout this document.

The camera and the RFID reader module are encased in plastic enclosures. The enclosures are designed so that they are easily mountable onto most conventional electric wheelchairs. ***Figure 3-4*** and ***Figure 3-5*** illustrates the design of such enclosures.

The user interface enclosure would contain a slot for the MMC card reader, and a data cable line that connects to the sensor enclosure.



]

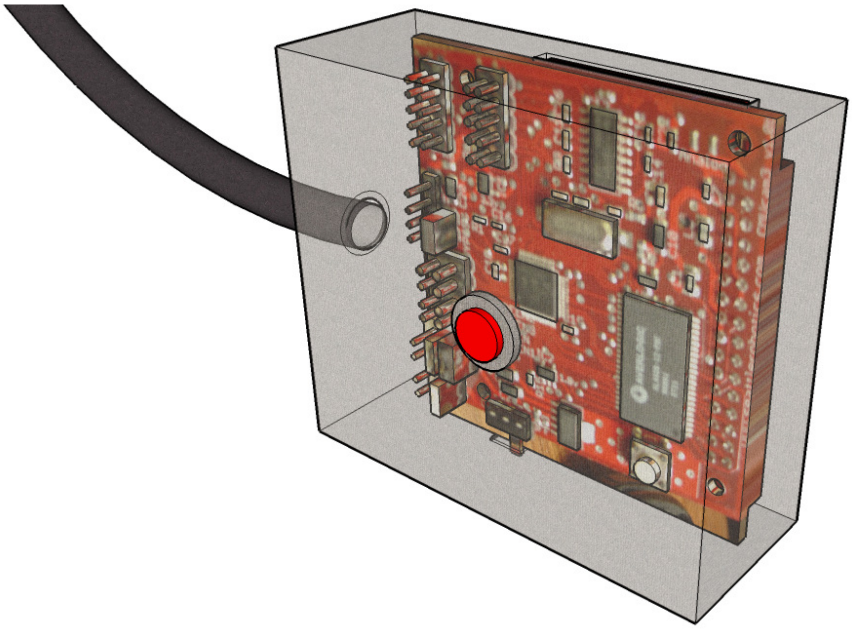


Figure 3-5: User Interface Enclosure

3.5 Component Positions

The main module is universally usable for both left and right handed users. Thus, it may be mounted on either side of the wheelchair's arm rest.

Our system may be mounted onto electric wheelchairs that typically use one set of wheels to propel it, as illustrated in ***Figure 3-6***Figure 3-6: Wheelchair Steering Methodology.. The Sensor module maybe mounted anywhere in front of the main set of wheels. It will not work correctly if it is mounted behind the set of wheels. In ***Figure 3-6***, the green arrows indicate the turning direction of the wheelchair, whereas the arrows enclosed in blue blocks specify the rotation of the actuating wheels.

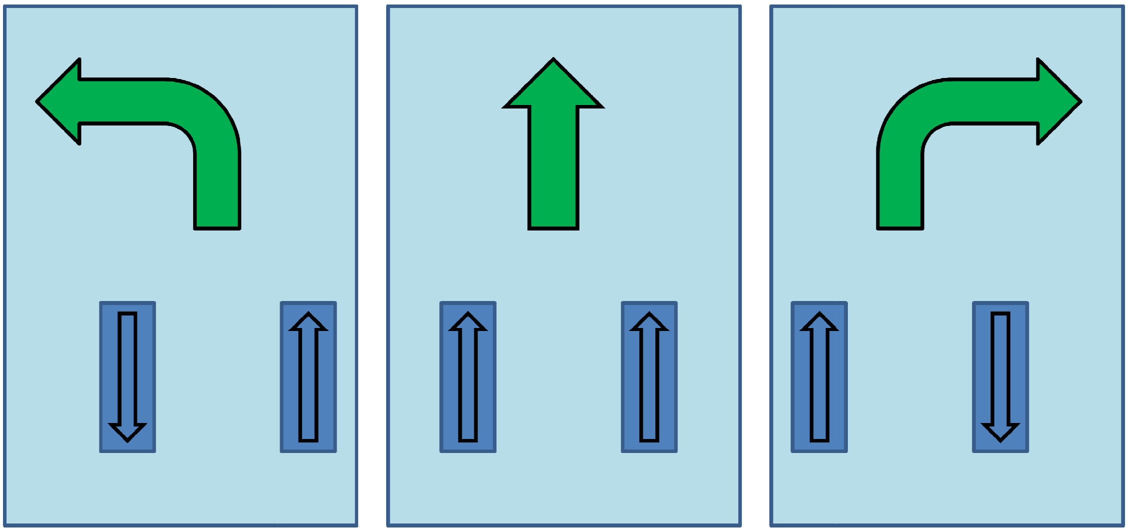


Figure 3-6: Wheelchair Steering Methodology.

3.6 Floor Plan Design

The floor plan will consist of single bidirectional tracks as illustrated in ***Figure 3-7*** and ***Figure 3-8***. For prototype purposes, the system will be incapable of avoiding special situations such as on-coming traffic. ***Section 5.7*** will explore and discuss about special situations and propose possible solutions to the problem.

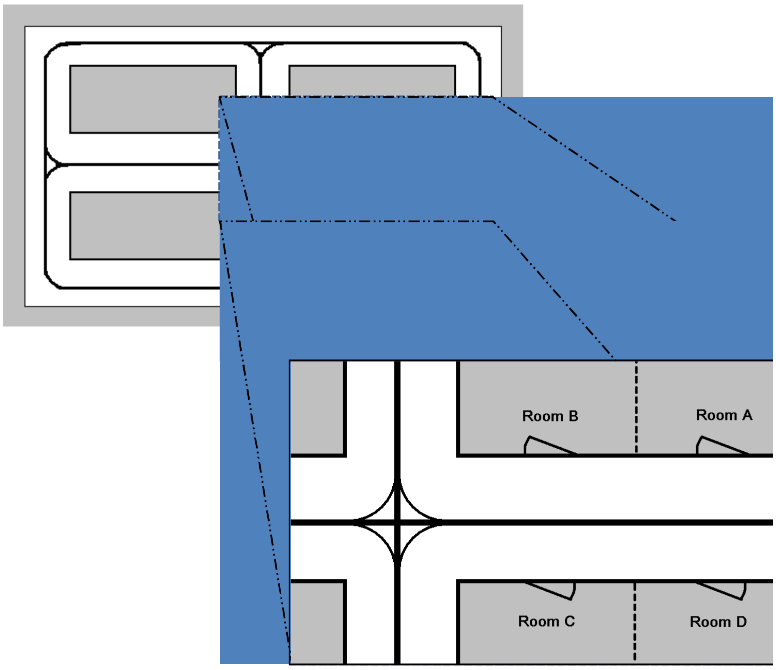


Figure 3-7: Floor Plan Design 1

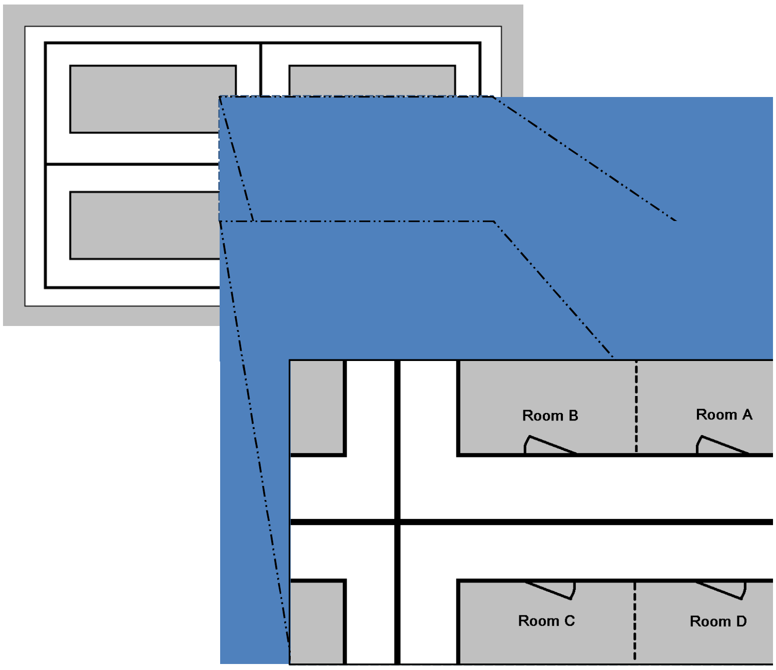


Figure 3-8: Floor Plan Design 2

3.7 Electrical System

The electrical system of the SDW prototype model includes the CMUCam3 vision sensor module, RFID Reader module, pulse width demodulation circuit, and a light source for the vision sensor. These electrical systems will be powered by either a 6V AA battery pack or a 9V battery. The CMUCam3 module has a 5V voltage regulator that allows 6-15V of power input.

4 Sensor System Design

This section discusses the sensor technology associated with the SDW system. It describes in detail what type of sensor system was chosen for video processing and location identification. All the advantages and limitations are discussed to justify the selection of the sensor system.

4.1 Visual Sensor

The SDW system will rely on video feedback to follow the track. A digital CMOS camera will be mounted under the wheelchair's lowest panel (depending on the wheelchair design) and will follow an edge of a track on the floor. The video processing will be performed by the CMUCam3 microcontroller. The visual sensor consists of an OV6620 single-chip CMOS CIF colour digital camera.

Due to its various advantages such as low power dissipation and small size, the OV6620 CMOS camera module is ideal for this project. More detail on the OV6620 can be found in its datasheet [6].

The OV6620 sensor also comes with its limitations: The maximum frame rate is only 60FPS. Combined with the circuit on the CMUCam3 board, the frame rate is reduced even further to 27FPS.

Despite the limitations, the OV6620 is an efficient and easy-to-use vision sensor that can be integrated to perform the track following methods. An image of the CMUCam3 vision sensor module is shown in ***Figure 4-1*** [2].

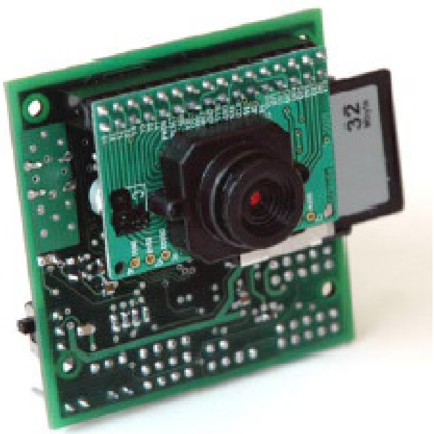


Figure 4-1: CMUCam3 Vision Sensor Module

4.2 RFID

In order to determine the location of the wheelchair, a Radio Frequency Identification (RFID) system will be employed. Similar to the video camera, the RFID reader will be located at the lowest panel of the wheelchair and will be continuously scanning the floor to read the RFID tags. As will be explained in ***Section 5.2***, each location will contain a RFID tag in order to identify the current location of the user.

When the SDW system is driven over a RFID within range, the unique ID of the tag will be transmitted as a 12 byte ASCII string via the serial output pin of the RFID reader. This output will be received by the CMUCam3 microcontroller in order to determine the current location of the wheelchair. Each tag will have its unique identity stored in the microcontroller's flash memory. Therefore, the microcontroller will simply have to compare the received ID with the stored IDs and take actions accordingly. Please refer to ***Section 5.2*** for more details.

Advantages

- Fully integrated low cost method of reading passive RFID transponder tags
- 1 wire serial TTL interface to PC or microcontroller
- 0.1" pin spacing for easy prototyping and integration
- Requires single +5V DC supply
- Low cost – RFID Reader $37.95, RFID tags $2.61 each

Another advantage of the RFID reader from Parallax was its compact size shown in ***Figure 4-2*** [3]. The RFID transponder tag dimensions are shown in ***Figure 4-3*** [3].

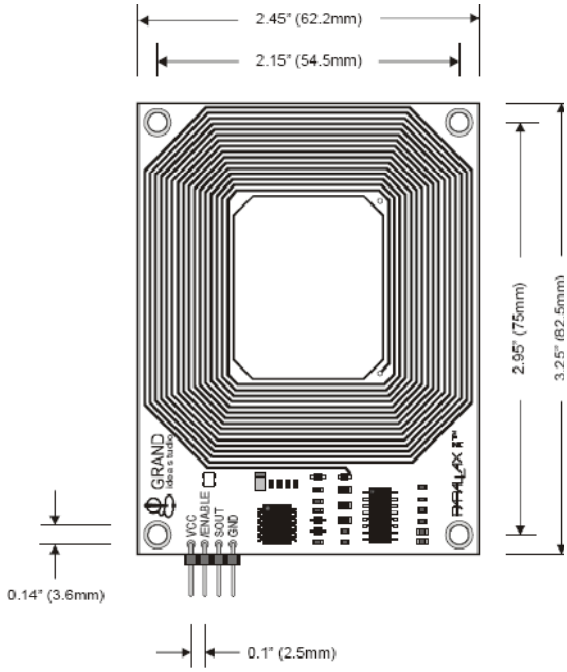


Figure 4-2: Parallax RFID Reader Dimensions

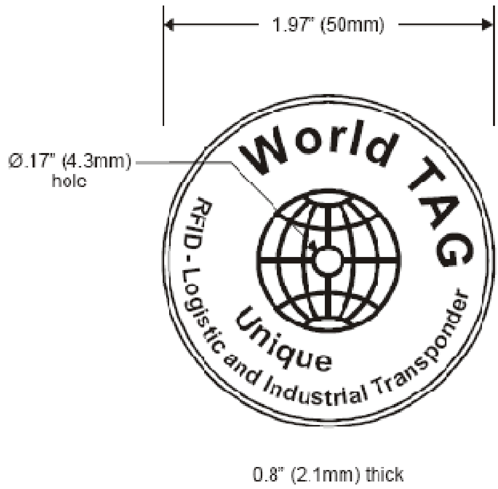


Figure 4-3: Parallax RFID Transponder Tag Dimensions

5 Navigation System Design

In this section, navigation system will be discussed in detail.

5.1 Overall System Design

Navigation system uses the OV6620 CMOS Camera module as its primary feedback from the surrounding environment. RFID tags are used as location identifiers. Colored tracks are laid on the floor for the digital camera to follow.

5.2 Floor Plan Design

Floor track is used to guide the self-driving wheelchair. It consists of coloured lines that are in high contrast to the floor. This allows the visual sensor to process accurately. For the prototype design, the track structure is a combination of turns, intersections, and straight lines. A few designs are mentioned in Section 3.6. The most suitable design will be picked during the testing stage.

Depending on the position of the camera, the width of the track can be varied. In the prototype design, the camera is mounted near to the floor; hence the width of the track should be in the range of 6 ± 1 cm. This allows the track to fit into the camera's viewable window.

As for RFID tags, they are embedded underneath the track at points of interests such as intersections, rooms, and cafeterias, etc. The tags act as location identifiers, which allow the system to determine its current location.

5.3 Image Processing

5.3.1 Edge Following

Digital images can be processed in two main ways to determine the edges; Gradient method and Laplacian method [4]. For the prototype, a type of gradient method called Sobel is utilized to process the images. ***Figure 5-1 [5]***, shows an example of how an image is processed under the Sobel method.

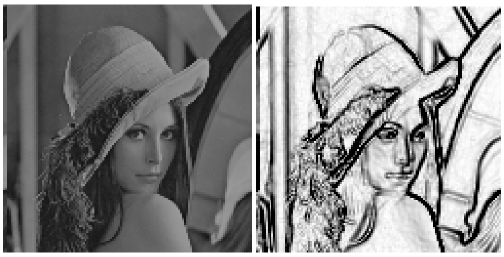


Figure 5-1: Sobel Edge Detection Method Example

The Sobel image processing method allows the microprocessor to convert the images into lines suitable for computation, as shown in ***Figure 5-2.***

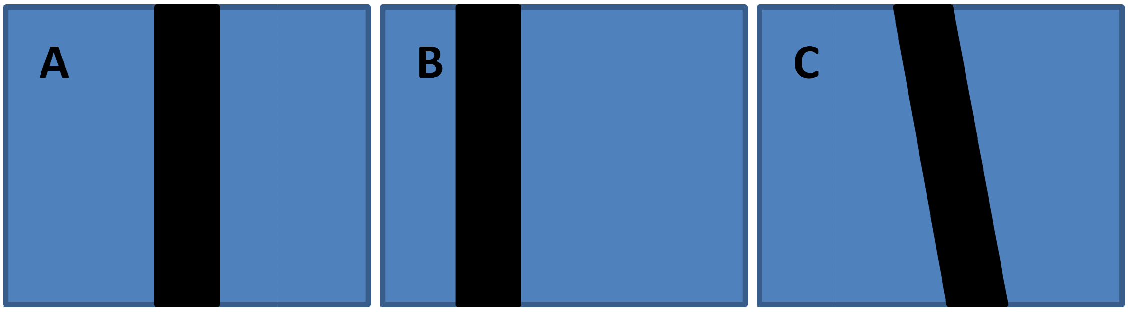


Figure 5-2: Example Images seen from Vision Sensor

Theoretically, the images after Sobel processing method will look like *Error! Reference ource not found.*. The images will then go through linear regression in order to obtain line equations in a form of $y = mx + b$, where *m* represents the slope and *b* represent the *y-intersection*.

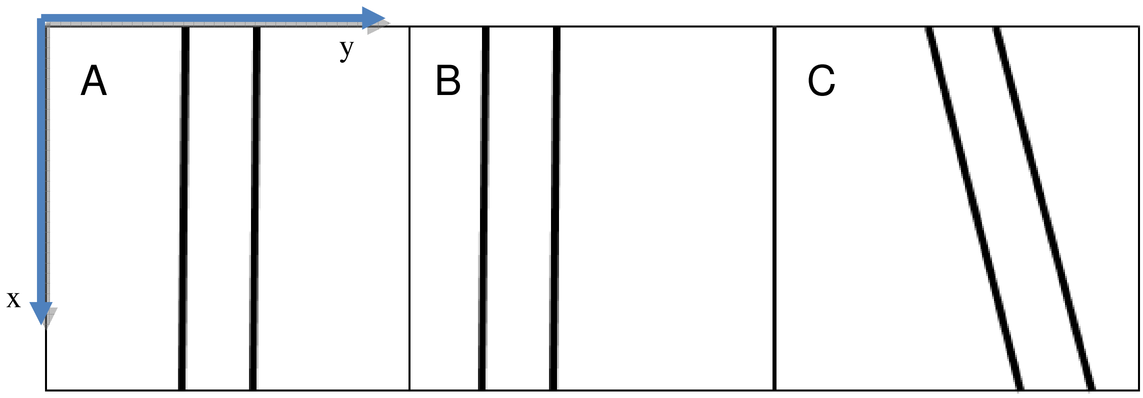


Figure 5-3: Theoretical Sobel Image Processing

The line equations will allow the microprocessor to determine the position of the track and feed the information back for the servo controller. In *Error! Reference source not found.* ***B***, the slope of the lines are around zero. However, the *y-intersections* are slightly lower, in other words, shifted to the left. In a case like this, the microprocessor will call the servo function to turn left and then until the *y-intersections* are centered and then called turn right to align back to the track. In case like in *Error! Reference source not found.* ***C***, a simple turn right command will be called to orient the wheelchair back to position.

5.3.2 Histogram Matching

5.3.2 Histogram Matching

Histogram matching technique is used to determine whether the wheelchair is on the track. This technique samples a portion of the grey scaled image with each pixel value between 0 and 225. An example of histogram matching technique is shown below in ***Figure 5-4*** to ***Figure 5-6***.

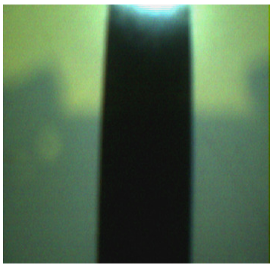


Figure 5-4: Trained Histogram

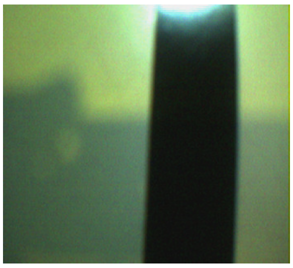


Figure 5-5: Sample 1 Histogram


Figure 5-6: Sample 2 Histogram

Figure 5-4 is the trained image which is used for matching. ***Figure 5-5*** and ***Figure 5-6*** are the run time generated images that will be compared against the trained image. The graph below shows the histograms of the three images shown above.

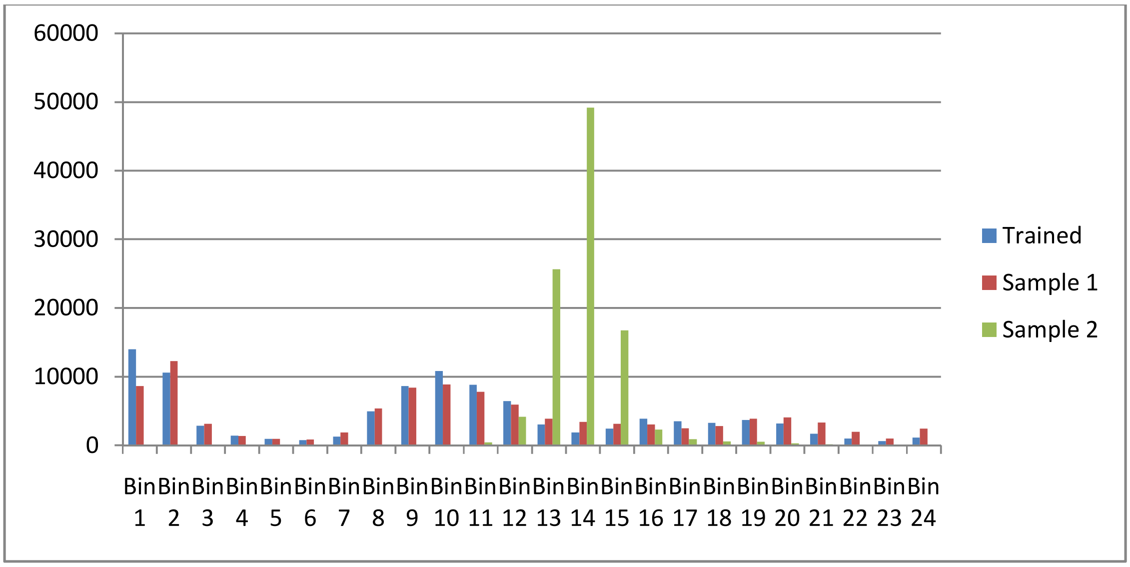


Figure 5-7: Histogram Example Data Results

Observing ***Figure 5-7***, it is obvious the histogram of *Sample 1* matches closely with the *Trained* image. In contrast, *Sample 2* deviates significantly from the *Trained* image.

In order to determine whether the image is a match, the difference of each bin between sample and trained image is taken and then summed up. This value is called total error.

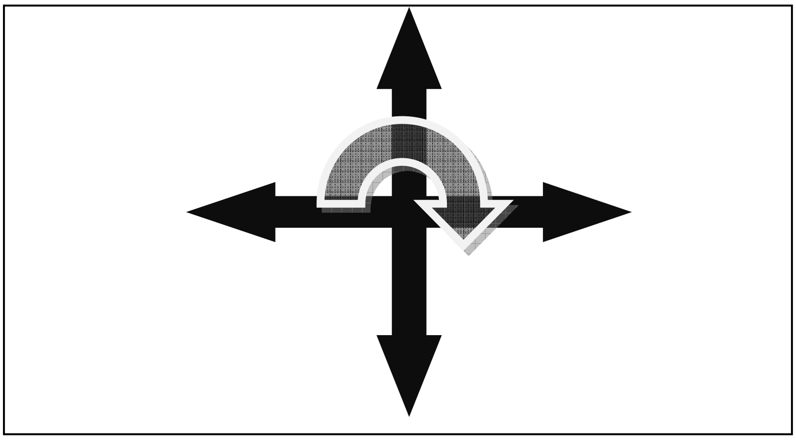
If the total error is lower than a threshold, then it would be considered a match. The threshold value will be determined during the calibration stages.

5.4 Track Searching Algorithm

5.4.1 On-Track Searching

On-track searching routine is used in two situations. One situation is when the SDW system first begins and has no information on its current location. To determine its current location the system must go over two RFID tags. Second situation is when the SDW system is at an intersection and needs to search for the intersecting lines.

The algorithm utilizes the histogram matching technique to find tracks. When on-track searching algorithm is called, the direction of search and number of tracks must be specified. The on track searching will use the specified direction, either clockwise or counterclockwise, and use histogram matching to count how many tracks has been detected. An example is given below to explain the method.



An example function called for right turn would be,

On_Track_Searching(CW, 1);

This means the wheelchair will turn clockwise and find the first track to follow.

5.4.2 Off-Track Searching

The off-track searching routine is used in a number of situations. One such situation is when the user decides to let the SDW system to search for the track upon start up of the system. The off track searching routine is particularly important for cases when the camera loses its sight of the track while in operation and needs to recover by searching for the track again. ***Figure 5-9*** shows the off track searching algorithm.

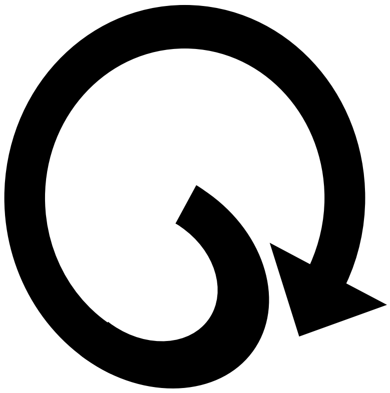


Figure 5-8 Off-Track Searching Pattern

When off-track searching algorithm is called, the SDW system will start spiral out from its current position. The algorithm will terminate when either track is found or timer is up. For prototype design, the search pattern will be hard coded into the algorithm. However, for future improvement, the obstacle detection algorithm will be able to perform off track searching while avoiding obstacles, and dynamically change search pattern.

5.5 Route Planning

A pre-calculated map structure is to be loaded into the system, giving the system instructions to travel from one location to another. Each instruction inside the map structure is a linked-list node containing information of a desired location, as illustrated in ***Table 5-1***.

Table 5-1: Path Map

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	**X**	node(2,1)	node(3,1)	node(4,1)	node(5,1)	node(6,1)	node(7,1)
2	node(1,2)	**X**	node(3,2)	node(4,2)	node(5,2)	node(6,2)	node(7,2)
3	node(1,3)	node(2,3)	**X**	node(4,3)	node(5,3)	node(6,3)	node(7,3)
4	node(1,4)	node(2,4)	node(3,4)	**X**	node(5,4)	node(6,4)	node(7,4)
5	node(1,5)	node(2,5)	node(3,5)	node(4,5)	**X**	node(6,5)	node(7,5)

6	node(1,6)	node(2,6)	node(3,6)	node(4,6)	node(5,6)	**X**	node(7,6)
7	node(1,7)	node(2,7)	node(3,7)	node(4,7)	node(5,7)	node(6,7)	**X**

There are two variables associated with each node, x_0,x_1; x_0 refers to the starting location, and x_1 refers to the destination. In the example mapping shown in ***Figure 5-10***, to travel from location 2 to 1, the desired path is to go through locations 4 and 7. When each via point location is reached, the next node is fetched until the destination matches the location.

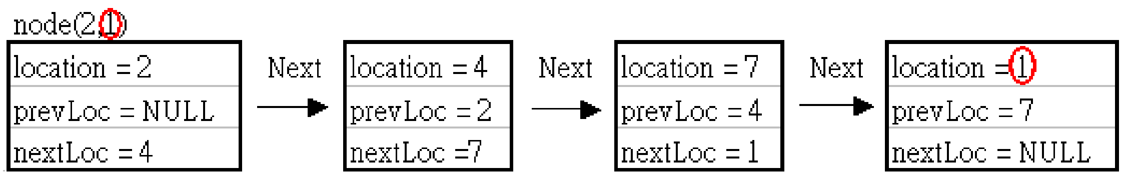


Figure 5-9: Path Generation Example

The pre-calculated map data could be generated manually or calculated by using the closest path method from graphing theory [8] [9]. To calculate such data, a table of adjacent matrices must be manually generated first. ***Table 5-2*** is an illustration of the adjacent matrices generated for a map described by ***Figure 5-10***.

Table 5-2: Example Adjacent Matrix with respect to Figure 5 8

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	**0**	0	0	0	0	0	1
2	0	**0**	0	1	0	0	0
3	0	0	**0**	0	0	1	1
4	0	1	0	**0**	0	0	1
5	0	0	0	0	**0**	1	0
6	0	0	1	0	1	**0**	0
7	1	0	1	1	0	0	**0**

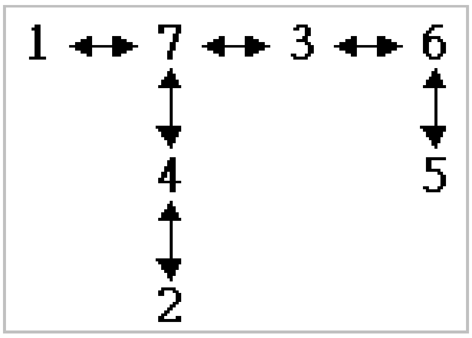


Figure 5-10: Example Map Definition

A ‘1’ in a cell indicates that a valid adjacent path exists between the two locations, where a ‘0’ indicates that no path exists.

5.6 *Direction Determination*

At any current location, given the information of previous location and next location, the determination of turning direction is crucial to the ability to follow a route. Since the track only consists of 90°, 180°, and 270° turns, a simple algorithm is sufficient. Based on the enforced 4-way coordination which is similar to compass coordinates, a table can be generated for direction determination. ***Figure 5-11*** illustrates the enforced 4-way coordination and ***Table 5-3*** demonstrates the table.

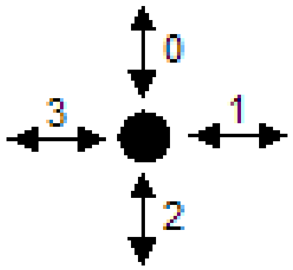


Figure 5-11: Region Definition

Table 5-3: Relative Region with respect to Figure 5-10: Example Map Definition

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	**X**	X	X	X	X	X	1
2	X	**X**	X	0	X	X	X
3	X	X	**X**	X	X	1	3
4	X	2	X	**X**	X	X	0
5	X	X	X	X	**X**	0	X
6	X	X	3	X	2	**X**	X
7	3	X	1	2	X	X	**X**

To explain ***Table 5-3*** in more detail, consider an X in row1 column2. The X indicates that location1 and location2 are not adjacent; a 0 in row2 column4 indicates that location4 is located at direction 0 of location2. Using ***Formula 1***, we are able to calculate which direction to take to reach the next location.

$$
\begin{aligned}
&(4+2-a+b)\%4 = turndirection \\
&a = TableX(currentLocation, prevousLocation) \\
&b = TableX(currentLocation, nextLocation) \\
&turndirection: 0 = Forward; \quad 1 = TurnRight; \quad 2 = TurnBack; \quad 3 = TurnLeft;
\end{aligned}
$$

Formula 1: Turn Direction Formula

An example demonstrating the use of ***Formula 1*** is given in the appendix.

5.7 Special Situation Analysis

Situations that may be encountered during the use of SDW system, but are not included in the prototype design are listed as special situations in this section.

5.7.1 Traffic

Traffic is a very common situation for users of the SDW system because the environment is usually shared. Due to the single track design approach (see Section 3.6), wheelchairs may collide while traveling on the same track. Also, pedestrians may be harmed if standing too close to the track. Because of the time constraint on the project, handling of traffic is not included in the design of the prototype device. However, solutions to these problems are considered and addressed in Section 9.1.

5.7.2 Narrow Path

In case a narrow path in the floor where only one wheelchair can go through, the path must be designed as one-way only. This approach can be easily implemented using our design of path definition and direction determination (see Section 5.5and 5.6).

5.7.3 Undefined track

When the environment has some tracks implemented prior to the installation of the SDW System, the prototype device may malfunction. Hence, a clean and well-maintained track dedicated to the SDW System is required for the prototype to function correctly. Another solution is to eliminate the need of track for the SDW system, which is discussed in Section 9.2.

5.8 Speed Limitation

Due to safety considerations and the limitations of the RFID reader, the speed of the wheelchair needs to be low. To ensure that RFID reads the tag correctly, the speed of travel will be kept at roughly 0.2m/s.

6 Wheelchair Control Design

The wheelchair of the system is controlled by a joystick which communicates to a servo control module. Tapping into the servo control module inputs and simulating the joystick signals are direct ways of simulating controlling the wheelchair.

6.1 Simulation of joystick controller

The joystick output consists of four analog signals which can be viewed as two sets of signals as shown in ***Figure 6-1***. Each set of pins sums up to be exactly 5V..

JS0	JS1	JS2	JS3

Figure 6-1: Joystick Output Pin Definition

The first set of signals (JS0 and JS1) controls the forward or backward motion of the wheelchair, while the second set of signals (JS2 and JS3) control the wheelchair to move left or right. Therefore, the joystick control signals can be simulated with four analog signals.

The CMUCam3 board is equipped with four servo ports with pulse width modulation (PWM) outputs. A low pass filter is used to filter the PWM signal to get a DC voltage proportional to the duty cycle of the servo. Doing so creates a relatively constant DC voltage output, usable for the wheelchair controller. Using a diode to drop the filtered DC signal by 0.7V and a non-inverting op-amp circuit to amplify the voltage, a desired range of voltage centered around 2.5 volts can be generated to imitate one of the joystick signals. The other joystick signal in the same pair can be obtained simply with an adder/subtracter circuit. The circuit to transform PWM signal to a DC voltage is illustrated in ***Figure 6-2***.

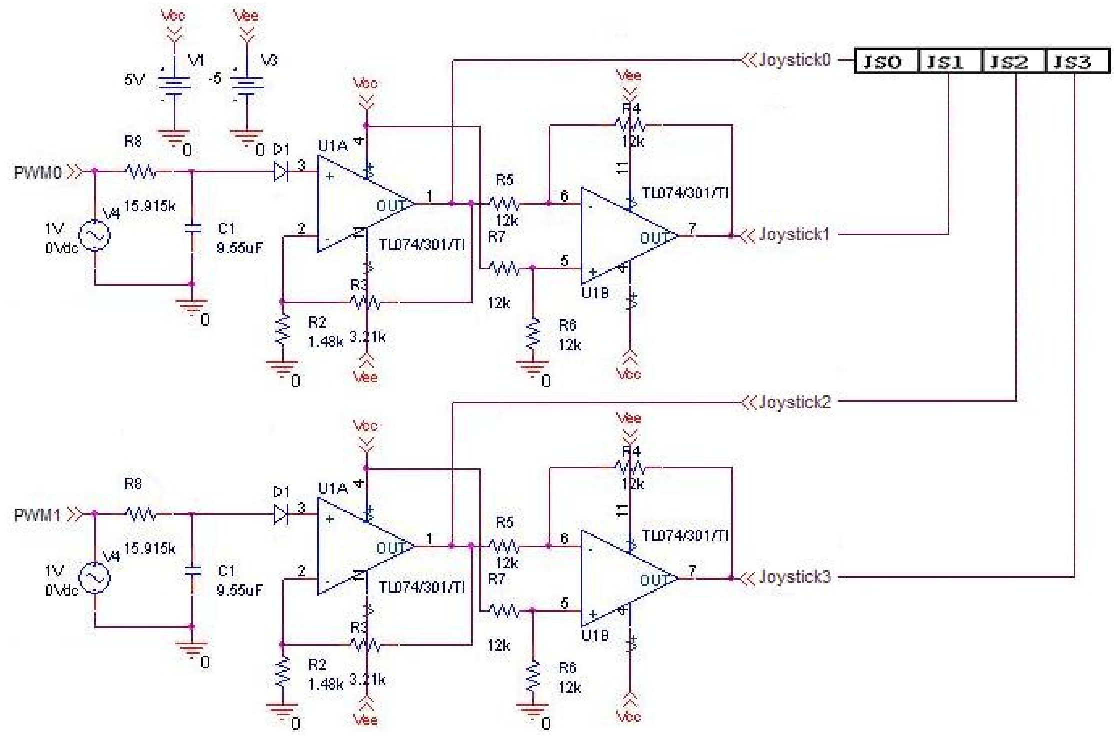


Figure 6-2: Pulse Width Demodulation Circuit

Note that while the CMUCam3 board supplies the +5V power, it does not supply the -5V needed. The -5V supply can be obtained by using a DC-DC converter, such as the ICL7660 [10].

6.2 Wheelchair driving velocity

The driving velocity of the wheelchair is dependent on the magnitude of the input to the JS0 to JS3 pins. To keep the wheelchair stationary, 2.5V is supplied to all 4 Joystick pins. As the JS0 output increases and the JS1 output decreases, the wheelchair turns more sharply towards the left; as the JS2 output increases and the JS3 output decreases, the wheelchair moves backward at a higher speed. ***Table 6-1*** shows a few example of how the Joystick pin voltages affect the velocity of the wheelchair,

Table 6-1: Joystick Pin Outputs

	JS0	JS1	JS2	JS3
Idle	2.50	2.50	2.50	2.50
Backward	2.50	2.50	3.7	1.28
Forward	2.50	2.50	1.25	3.73
Left	3.79	1.17	2.50	2.50
Right	1.09	3.91	2.50	2.50
Left & Forward	3.39	1.57	1.88	3.06
Right & Backward	1.85	3.13	3.26	1.72

6.3 Calibration Compatibility Consideration

Due to the limited resistor tolerance, the analog circuit that was built to simulate the joystick signals will not have exact theoretical value desired (i.e. the mean of the obtained voltage does not equal to 2.5V). The PWM signals need to be carefully generated and calibrated to ensure desired velocity can be obtained.

7 User Interface Design

This section discusses the user interface hardware that will be accessed by the wheelchair user. It describes the physical design and layout of the hardware and how the user will select the various features available. The prototype and production versions of the user interface will be discussed separately.

7.1 User Interface Hardware

7.1.1 User Interface for Prototype

Initially the user interface will only contain one button in the prototype model. This button will be used to reach three different locations within an in-door environment. The interface will function as follows:

- Press once to reach location 1
- Press twice to reach location 2
- Press three times to reach location 3

The user interface will be located on the right armrest of the wheelchair for convenient access. ***Figure 7-1*** show a 3D model and the dimensions of the prototype.

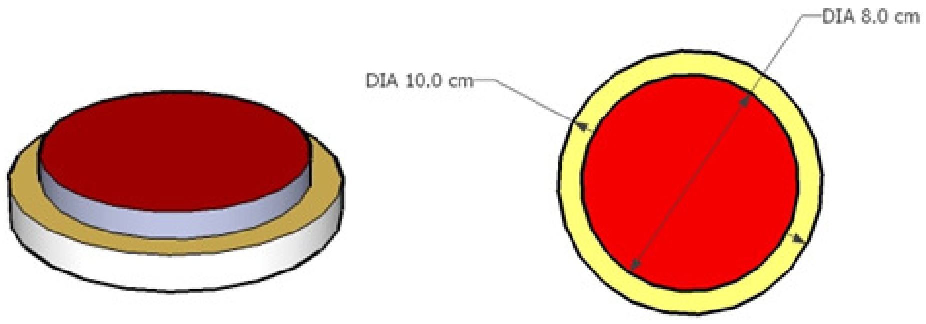


Figure 7-1: Prototype User Button with Dimensions

7.1.2 User Interface for Production

The production model user interface will consist of large sized buttons for convenient access. Each location will have a corresponding button for the user to select.

For safety precautions, the joystick will be implemented to act as manual override of the auto-navigation system. In emergency situations, such as to avoid an obstacle ahead, the user would pull the wheelchair's joystick backwards on instinct stop the wheelchair and switch back to manual mode. A possible design of the production user interface is shown in ***Figure 7-2***.

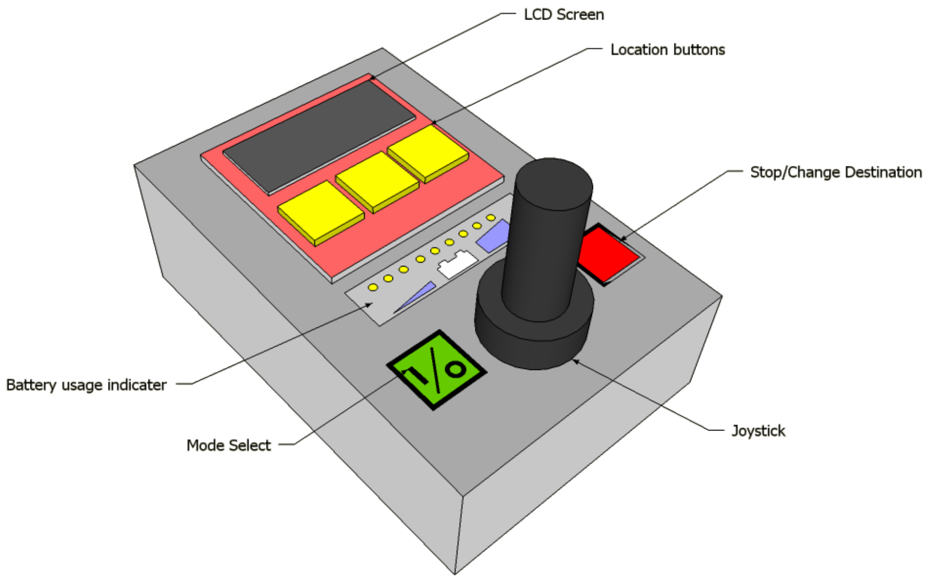


Figure 7-2: Production User Interface Design

8 System Test Plan

Before testing the overall system, individual subsystems or modules are tested for functionality and for optimization. ***Table 8-1*** contains the test plans for each subsystem and lists the test case number, description, input and desired outputs of the functional SDW System.

Table 8-1: Test Case Table

Test Case	Description	Input/Method	Desired Output
DTC-001	Response Time of RFID	Driving over RFID tag at various wheelchair speeds	RFID reader successfully reads the correct tag ID
DTC-002	Range of RFID	Driving over RFID tags with reader at different elevation	RFID reader successfully reads the correct tag ID
DTC-003	RFID communication	Enable microcontroller to read different RFID tags	Microcontroller returns correct ID tags
DTC-004	Straight Line tracking algorihm	straight line track, microcontroller controlled wheelchair	wheelchair follows the track in a straight line
DTC-005	Curve tracking algorithm	left & right curve, automated wheelchair	wheelchair follows the curves smoothly
DTC-006	Path Calculation	pre-defined map, automated wheelchair, tracks	wheelchair follows the correct calculated path
DTC-007	Path Finding Algorithm	automated wheelchair in off track position, one closest track	wheelchair finds the track and follows the track
DTC-008	Pin triggered interrupt	microcontroller, raising of a desired status pin	trigger an interrupt and performs interrrupt service routine
DTC-009	PWM demodulation circuit	microcontroller servo ports	PWM demodulation circuit outputs DC voltage in desired range
DTC-010	Wheelchair Servo Control	microcontroller servo ports	wheelchair wheels moves correspondingly to the servo signals
DTC-011	Intersection Right	Send right turn decision at an intersecction	Wheelchair follow right path

	Turn		
DTC-012	Intersection Left Turn	Send left turn decision at an intersecction	Wheelchair follow left path
DTC-013	Intersection Go Straight	Send go straight decision at an intersecction	wheelchair follow forward track without stopping
DTC-014	Intersection Turn Back	Send turn back decision at an intersecction	Wheelchair executes the correct "turn back" algorithm and follow track
DTC-015	Route Determination	Input location information to the program	Returns a valid linked list with correct route description
DTC-016	Turning Direction Determination	Input location information to the program	Returns a correct turning decision
DTC-017	Edge Generation	Take a picture of the track	Output a corresponding line equation
DTC-018	Histogram Matching	Train the system with template image	Be able to detect track when it's in site of camera window
DTC-019	Edge Following	Input different orientation of track	Output proper servo signal
DTC-020	Track Following	Input different type of floor track	Determine which one gives the best outcome
DTC-021	On Track Searching	Input different turn commans	Be able to trun to desired track
DTC-022	Off Track Searching	Place the system at different location off the track	Be able to find the track with maximum probability

9 Future Development Systems

Due to limited resources, some functions will not be implemented in the prototype device. These functions are listed in this future development section.

9.1 Anti Collision System

Anti-collision is a crucial system for ensuring the safety of the passenger. It will interact with the microcontroller to steer the wheelchair away from obstacles thus avoiding possible accidents. Most importantly, the anti-collision system will also help the wheelchair to avoid pedestrians in hallways. With addition of this feature, the SDW system will truly become a convenience for its users. One possible design for implementing this feature is through the use of ultrasonic parking sensors. By storing the states of the SDW system right before the detection of obstacles, the wheelchair will be able to avoid obstacles as well as getting back on the track.

9.2 Trackless Auto Navigation System

As suggested by its name, the Trackless Auto Navigation System removes the necessity of the tracks, thus enable the SDW system to become a standalone system. Without the track, the SDW system will rely solely on video processing to perform its functions. This will eliminate the need for modification of the surroundings. Hence, a more powerful microcontroller is needed for the rigorous computations required for the video processing.

10 CONCLUSION

The proposed design solutions of the SDW System have been discussed in this document. These design specifications will be cross-referenced with the functional specifications and will guide this project to its completion. With the test plans included in this document, we can ensure all the required functions of the system are present and implemented correctly. This design specification also provides clear goals for future improvement of the system.

11 REFERENCES

[1] **Functional Specifications for a Self-Driving Wheelchair** [Report] : Functional Specifications / auth. Corporation, Drivomatic Technology / Engineering Science ; Simon Fraser University. - Burnaby : [s.n.], 2008.

[2] **CMUCAM3 Datasheet** [Report] : Datasheet / auth. Anthony Rowe Adam Goode and Carnegie Mellon University / Carnegie Mellon University. - 2007.

[3] **RFID Reader Module (#28140)** [Report] / auth. Parallax, Inc.. - 2005.

[4] **Other Methods of Edge Detection** [Online] // Image Processing: Edge Detection. - http://www.owlnet.rice.edu/~elec539/Projects97/morphjrks/moredge.html.

[5] **Edge Detection Tutorial** [Online]. - http://www.pages.drexel.edu/~weg22/edge.html.

[6] **Advanced Information Preliminary OV6620 datasheet** [Report] / auth. Omnivision. - http://www.cs.cmu.edu/~cmucam/Downloads/ov6620DSLF.PDF

[7] **Pulse Width Modulation** [Online] // Wikipedia. - http://en.wikipedia.org/wiki/Pulse-width_modulation.

[8] **Graph Theory** [Online] // Wikipedia. - http://en.wikipedia.org/wiki/Graph_theory.

[9] **Linear Algebra and its Applications”** [Book] / auth. Lay David C..

[10] **ICL7660 Datasheet** [Report] / auth. MAXIM. - http://datasheets.maxim-ic.com/en/ds/ICL7660-MAX1044.pdf

12 Appendix

12.1 Bill of Materials

Table 12-1 lists the materials used for the SDW System prototype model.

Table 12-1: Bill of Materials

Name	Part #	Qty	Designed by	Manufactured by
CMUCam3	RB-Sea-04	1	Carnegie Mellon University	Seattle Robotics
RFID Reader	28140	1	Parallax	Parallax
RFID Round Tag	28142	10	Parallax	Parallax
DC-DC Converter	ICL7660	1	Maxim	Maxim
Quad Amp.	TL074	1	Fairchild	Fairchild

12.1 Enclosure Dimensions

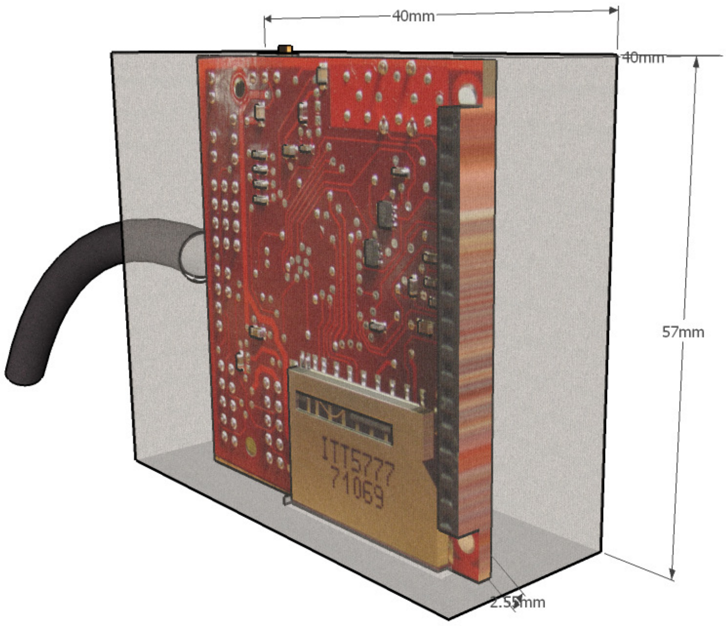


Figure 12-1: User Interface Enclosure Dimensions

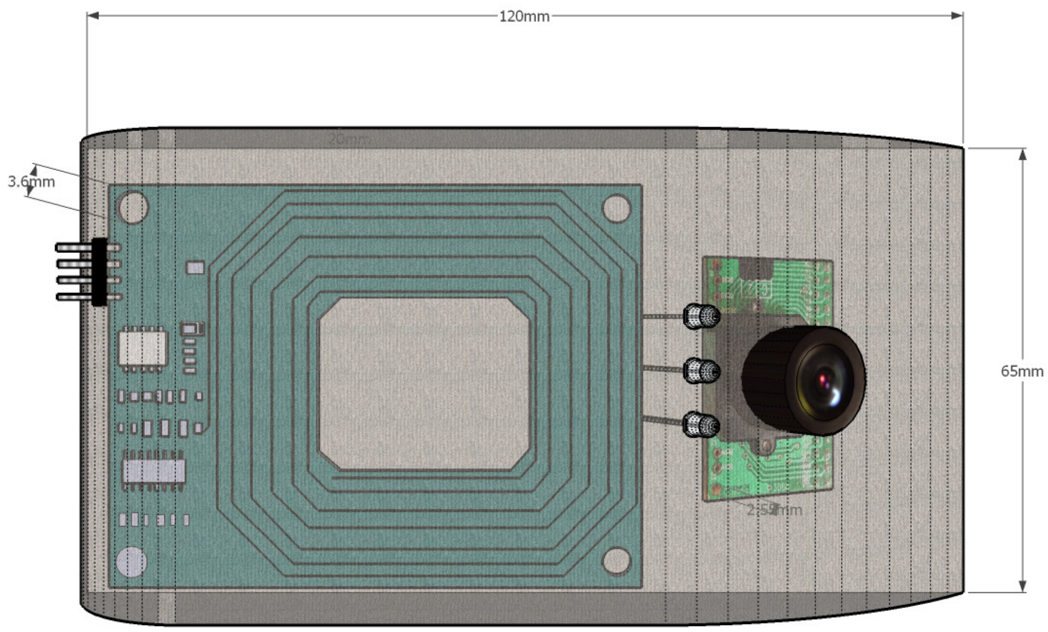


Figure 12-2: Sensor Enclosure Dimensions

12.2 Direction Determination and Route Planning

The turning direction determination method is derived from first enforcing the 4 different area around a location to 0,1,2,3 as shown in figure x.

Assigning forward to be 0, right turn 1, turning back 2 and left turn3, a table of all possible entrance and exit direction is generated to give the result shown in table x

Table 12-2: Possible Entrance and Exit Direction Results

EntrenceRegion	ExitRegion	Turn Direction
0	0	2
1	0	1
2	0	0
3	0	3
0	1	3
1	1	2
2	1	1
3	1	0
0	2	0
1	2	3
2	2	2

3	2	1
0	3	1
1	3	0
2	3	3
3	3	2

Thus Formula 1: Turn Direction Formula (equation in section 5) is devised from observing Table 12-2.

Example for ***Formula 1***

Consider the scenario where you are coming from location 4, with current location at 7, and trying to reach location 3.

Table 12-3: Direction Determination Example

Variable	Calculation	Result
a	Table 5-3 (7,4)	2
b	Table 5-3 (7,3)	1
Turning Direction	(4+2-2+1)&4	2 = Turn Right