School of Engineering Science
Simon Fraser University
Burnaby BC V5A 1S6

March 8, 2006

Mr. Steve Whitmore and Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby BC V5A 1S6

Re: ENSC 440 Design Specification for an intelligent walking aid

Dear Mr. Whitmore and Dr. Rawicz,

Please find attached a copy of Incedonex's Design Specification for the Sure-Step® intelligent walking aid. Sure-Step® is a technologically advanced walker created to assist elderly users. It is comprised of three subsystems: the M-Brake™ the Sense-Steer™ and the Nav-Pro™. These modules are integrated and aim to address problems such as abrupt breaking, walker rollback, hazard detection and path planning in order to better assist and protect the user.

The purpose of our design specification is to outline the various components and design steps taken to fabricate a proof-of-concept product. The design and implementation of each of the three modules is discussed as well as test procedures to verify their functionality. We are currently in the process of implementing our design and are targeting project completion by the end of April 2006.

Incedonex is comprised of six highly skilled and ambitious engineering students at Simon Fraser University with backgrounds in Engineering and Computer Science. You can find more information on Mr. Daniel Agyar, Mr. Duncan Chan, Mr. Steven Dai, Mr. Yijun Jing, Mr. Victor Tai and Mr. Li Xu in the attached proposal.

If you have any questions or comments please to contact me by e-mail at lxu@sfu.ca. Alternatively, you may contact me directly by telephone at 604-505-9063.

Sincerely,

Li Xu

Li Xu,
Chief Executive Officer
Incedonex
Enclosure: Design Specification for the Sure-Step® intelligent walking aid**.**

incedonex

SIMPLY MOBILITY, SURELY SAFETY

Design Specification for an Intelligent Walking Aid

Project Team	Daniel Agyar Duncan Chan Steven Dai Yijun Jing Victor Tai Li Xu
Contact Person	Li Xu lxu@sfu.ca
Submitted To	Dr. Andrew Rawicz Mr. Steve Whitmore *Engineering Science* *Simon Fraser University*
Issued Date	March 8, 2006

Executive Summary

The world's elderly population is greatly increasing. According to the World Health Organization, 390 million people are over the age of 65, and this number is expected to double by 2025 [1]. As people age, their mobility deteriorate and are prone to falling, which has become the leading cause for injury-related deaths in recent years [2]. As a result, the use of wheeled walkers has become a popular choice for elderly.

Though walkers offer support for the elderly to walk, they are often difficult to use in certain situations. For example, when walking uphill or downhill, the walker tends to roll in the respective directions, forcing the elderly to provide extra force in order to control the walker. Although a mechanical brake can be used to slow down and stop the walker, the brake usually results in an abrupt stop and may consequently cause injury to the user. Another problem inherent to walkers is the danger of tipping over a sidewalk or on uneven ground. This can result in life-threatening injuries among elderly users.

Moving past convention, Incedonex will address these problems and modify the current wheeled walkers by adding an advanced automatic braking system to prevent the walkers from unwanted rolling backwards or forwards on sloped surfaces. This system also incorporates hazard detection and avoidance via brake-controlled steering to provide additional safety in the vicinity of uneven ground. In addition, when users are in an unfamiliar surrounding, such as a shopping center, where they are unaware of where washrooms or elevators are located, Incedonex provides a solution by incorporating a state-of-the-art path planning navigating system embedded on the walkers to guide them to their destination. Together with these features and by providing a user-friendly interface, our product is the most advanced walker on the planet.

The Incedonex Intelligent Walking Aid Sure-Step® consists of three modules: M-Brake™, Sense-Steer™, and Nav-Pro®. Each module has its own set of functionalities. The functions of each module are outlined below.

- **M-brake Module:**
  Improved braking control for Sure-Step®

- **Sense-Steer Module:**
  Hazard detection and avoidance for Sure-Step®

- **Nav-Pro Module:**
  Navigate and plan a path to the destination for the user

Table of Contents

List of Figures

List of Tables

1 Introduction

The Sure-Step® walking aid is a revolutionary assistive device which aims to improve upon current designs, and provide elderly users with additional safety and control ability. The system is composed of three design modules: the M-Brake™, the Sense-Steer™, and the Nav-Pro™.

System development will consist of 3 phases. In the first stage, we will focus on conceptualization, using various development tools to simulate and test our design and algorithms with offline data. Through design iterations, attempts will be made to try to devise better solutions if current designs or algorithms do not completely meet our functional requirements. The second stage will involve using actual components to assemble individual design modules. This stage will also include testing and fixing any circuit or algorithm deficiencies of each the modules, especially the problematic area where the M-Brake™ and Sense-Steer™ modules directly interact with each other. Finally, we will integrate the whole and also run it in a real-time fashion. We will prepare it for commercial deployment as soon as we are sufficiently confidence about the integrity of our prototype design.

1.1 Scope

This document outlines the design of the Sure-Step® walking aid. We cover the required design detail for our proof-of-concept design as well as additional specifications for the prototype and commercial final product, so that Incedonex will be able to use it as a reference document during implementation.

For a more general overview of the system, which includes financing and marketing, please refer to *Incedonex project proposal for the Sure-Step® intelligent walking aid* [3].

For function specifications, please refer to *Incedonex Functional Specification for the Sure-Step® intelligent walking aid* [4].

1.2 Intended Audience

This documentation is not indented for external release and only represent a finalized design for a proof-of-concept product. This document is intended as reference documentation for the designers and engineers of Incedonex, as well as potential investors. Design Engineers shall use this document as a framework when designing and implementing the various modules of our products. Testing shall be carried out as defined in this document and potential investors shall use this document as a guide of what end consumers can expect of the production version of our product.

1.3 Acronyms

RFID	Radio Frequency Identification. It is an automatic identification method, relying on storing and remotely retrieving data using devices called RFID tags or transponders.
IR	Infrared

2 System Overview

This section outlines the basic functions of the Sure-Step® intelligent walking aid. This is a brief overview to give the reader a better understanding of the overall system operation. The main function of the Sure-step® is to provide safety and mobility to senior's life.

The Sure-Step® walking aid is comprised of three design modules: the M-Brake™, the Sense-Steer ™ and the Nav-Pro™. Figure 1shows a general block diagram of the various subsystems of the Incedonex Sure-Step® intelligent walking aid**.**

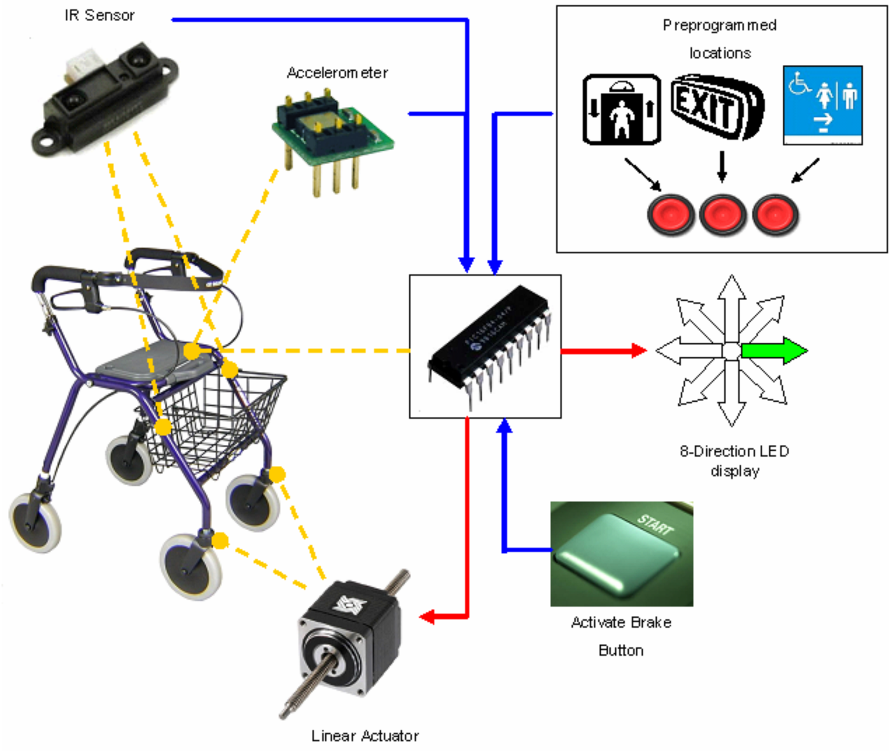


Figure 1: System Overview

For the M-Brake™, a touch switch is placed in the handles of walker as user interface, which is connected to a pair of electronically controlled brakes. An accelerometer is fitted to the walker in order to determine the change in acceleration of the walker. This sensor sends a signal to a microcontroller which will be used to compare acceleration and inclination thresholds in order to automatically engage the brake. Alternatively the user can activate the brake by using the mounted touch switch. When this switch is pressed the microprocessor will execute the control algorithm and generate the proper control signal to drive the brakes so that the walker will come to a complete stop.

For Sense-Steer™, two infrared sensors will be installed on each of the front side legs of the walker. The sensors will detect curb edges or hazards within a range of 8±3 cm on the side and 20±5 cm at the front. These sensors will generate an interrupt signal to the microcontroller which will be used to execute the obstacle avoidance algorithm and determine which side of wheels needs to slow down more to achieve steering function. It then activates the M-Brake™ to slow down on the desired wheel. Furthermore, the user will have more time to react and avoid the ditch.

The Nav-Pro™ module has three switches placed in the front top of walker, which are predefined locations: washroom, elevator, and exit. Once the user selects one of the switches, the LED displays the right direction for the particular location that the user has chosen. A wireless module will read location information from beacons located at specific waypoints. The processor of the system will automatically load the map and utilize the embedded path searching algorithm in order to determine the correct path with the shortest distance to the selected destination. An eight-way arrow LED display will provide visual guidance to the user. There will also be audio warning message to reminder the user to check the direction shown on LED whenever there is a change of direction. The switch will automatically be turned off once the user arrivals at the desired location. In addition, only one of these three switches can be turned on once a time. However, the user can change switches directly by pushing desired switch. Nav-Pro™ will stop the current task and start the new task according to user's latest input.

3 M-Brake™ system

3.1 System Architecture

The M-Brake™ system is designed to provide computer assisted electronically controlled braking for the walker. This system offers smooth and gradual braking, automatic anti run-away braking, and automatic emergency braking for fall prevention. The system is realized through the use of linear actuator driven mechanical brakes combined with an accelerometer and an onboard computer. The system architecture diagram below demonstrates the components of the system.

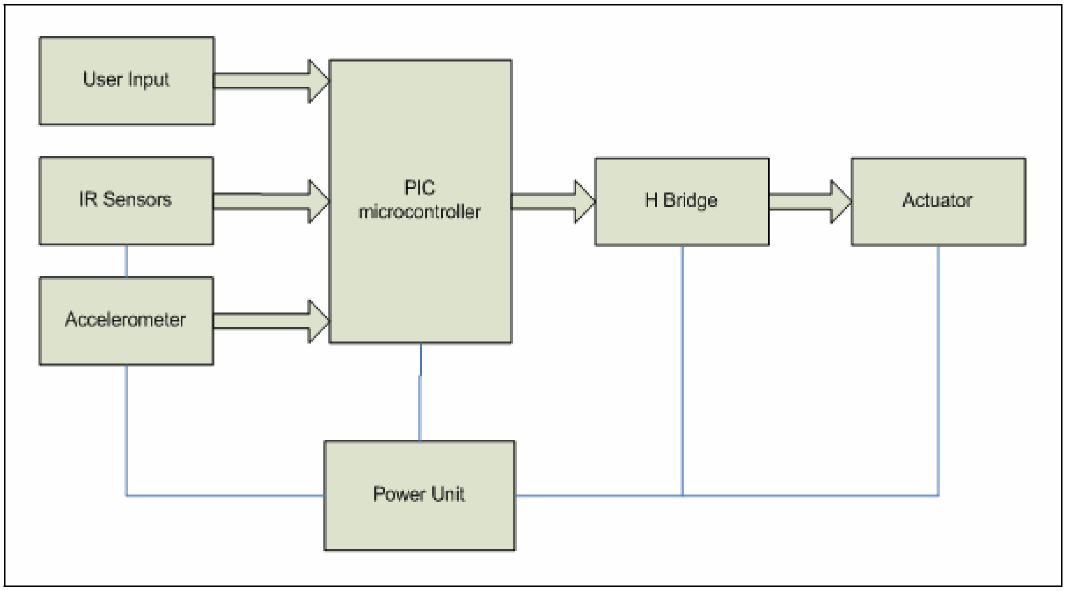


Figure 2: System Architecture Block Diagram

The accelerometer outputs digital read outs of the acceleration of the walker. The IR sensors are used in the Sense-Steer™ module; its output is fed into the same PIC microcontroller as the M-Brake™ module. The M-Brake™ control software which determines when to apply the brake and the required force and duration, is executed inside the PIC microcontroller along with the Sense-Steer™ algorithm. The PIC microcontroller outputs control bits that are used to drive the actuator. An H-Bridge module is used to drive the linear actuators. The linear actuators compress the mechanical brake with duration and force controlled by the driver module when it is activated. The detailed design of each module is described in the subsequent sections.

3.2 Mechanical design

This section outlines the mechanical design of the brake mechanism. Existing walkers use variations of two main designs for the brake. One design is a brake pad made of metal material. The brake pad is pressed against the rubber surface of the wheels when applied. See figure 3.

Figure 3: Walker brake type 1

The mechanism is similar to that of a bike brake. The amount of braking force can be fine tuned by varying the pressure applied on the brake pad. However, current walkers use a manual cable pulling mechanism to apply the pressure onto the brake pad. The braking force varies a large amount with little cable movement. As a result, it is very difficult for the users to fine control the braking force. Especially with elderly users, they can only control the brake in an abrupt manner which results in discomforting jerking motions. Additional disadvantage is that with a cable mechanism, the cable looses its tension after some use, and must be re-tightened once in a while. Another type of metal brake pad type brake uses a vertical screw to fine tune the braking force. See figure 4.

Figure 4: walker brake type 2

The users can adjust the brake strength by turning the screw to increase the pressure on the metal brake pad. Because the pressure is applied through a vertical screw, it can be fine tuned to meet the user's desire. However, the adjustment must be done prior to the application of the desired braking strength. The second type of brake is a slow down braking wheel. See figure [5].

Figure 5: Walker brake type 3

It is a small rubber wheel attached above the walker's wheel. A tuning screw is used to adjust the pressure between the slow down wheel and the walker's wheel. When the user screws down the screw, the slow down wheel is pressed tighter against the wheel's surface, causing more braking force. The advantage of using a slow down wheel instead of a metal pad is that the braking force can be tuned with finer adjustment. However this type of brake still doesn't offer dynamic adjustment of brake force.

M-Brake™ 's mechanical design combines the advantage of the screw driven brake with a high performance linear actuator. The linear actuator essentially replaces the cable pulley mechanism and the tuning screw mechanism. Figure 6 illustrates the mechanical design of the M-Brake™ mechanism.

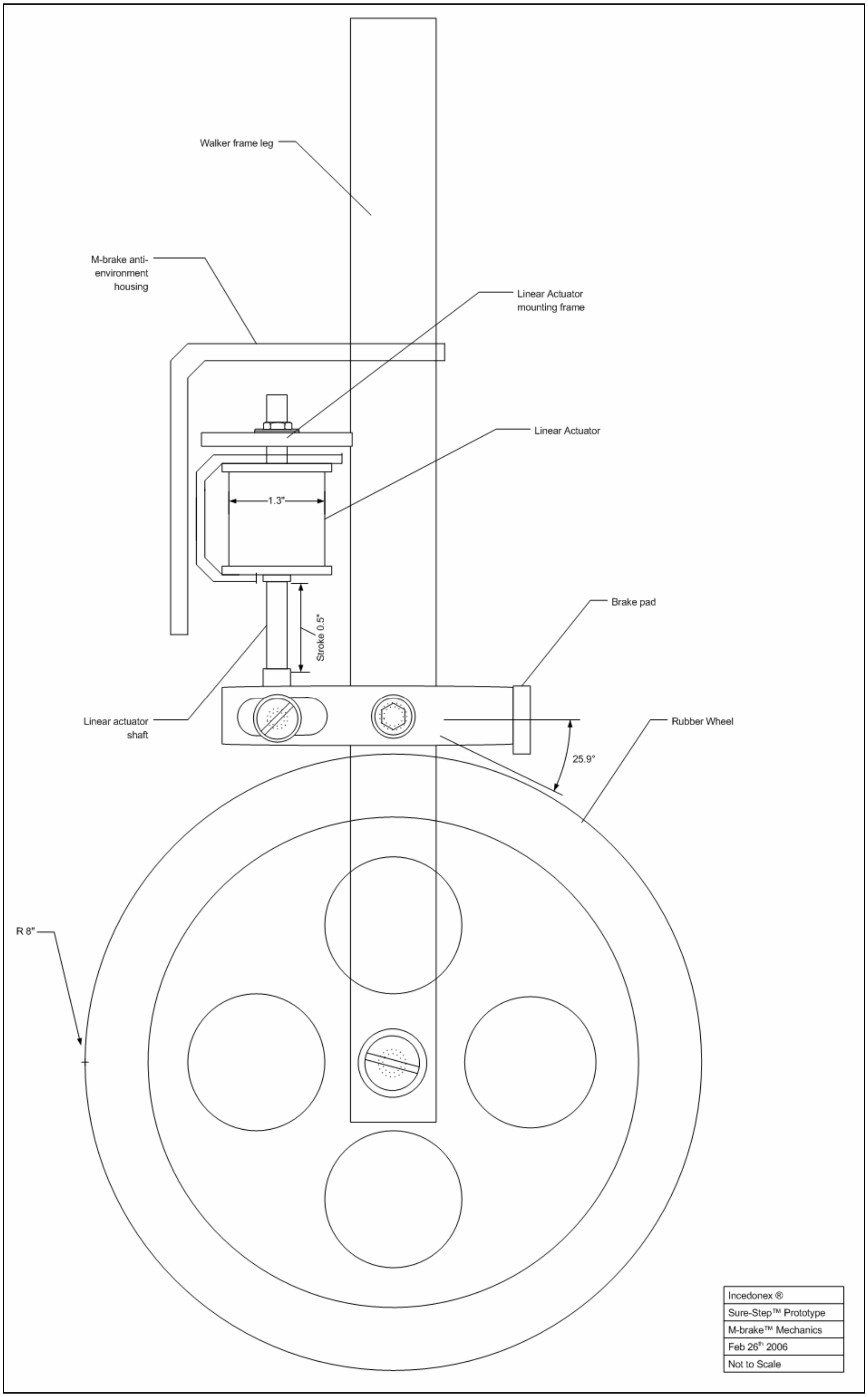


Figure 6: M-Brake™ Mechanics

This design offers minimum modification by utilizing the existing brake mechanism structure. The linear actuator is compact and able to supply up to 25lbs of linear force with step size as fine as 0.001” per step and speed up to 2” per second. This linear actuator provides enough force, speed and finesse required for our application. When the actuator shaft moves upward, the brake pad is pressed down against the wheel surface, thus providing brake force. When the shaft moves down, the pad is lifted off from the wheel, thus releasing the brake. An alternative design is conceived which uses the slow down wheel instead of the metal pad. This might provide even finer adjustment in braking strength. We will experiment with the designs to determine which is better. The detail of the linear actuator technical specifications is listed below.

3.2.1 Linear Actuator

Figure 7 illustrates the physical dimensions of the actuator we are using.

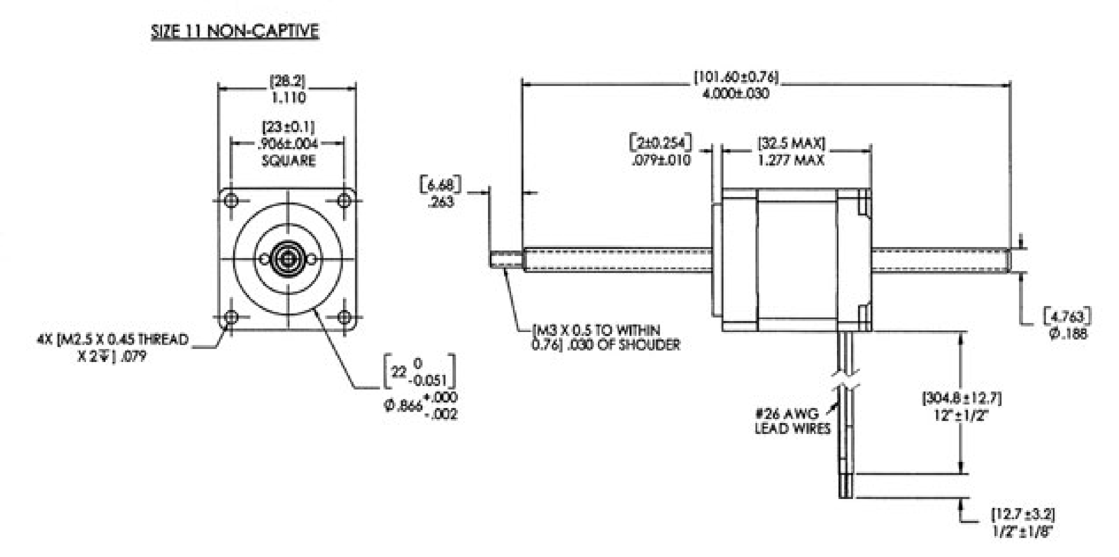


Figure 7: Dimension Drawing of the linear actuator [5]

Figure 8 shows the performance curve of the linear actuator.

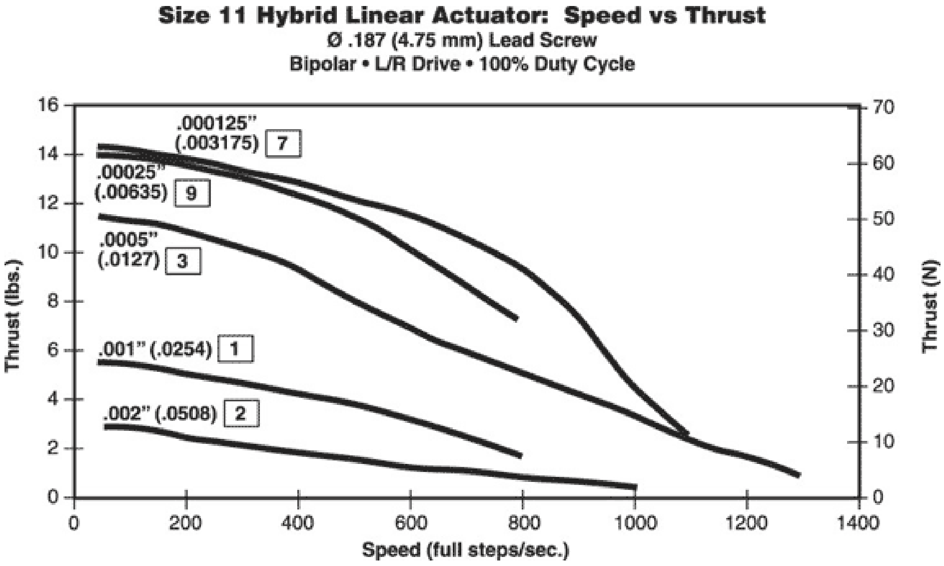


Figure 8: Performance curve of the linear actuator [5]

The following table lists the technical specifications of the linear actuator

Table 1: Technical specification of the linear actuator

Size 11: 28 mm (1.1") Hybrid Linear Actuator (1.8 Step Angle)	
Wiring	Bipolar
Operating voltage	12V
Current/phase	0.18amp
Resistance/phase	68.6ohm
Inductance/phase	39.0 mH
Power consumption	4.2W
Rotor inertia	9.0gcm²
Temperature rise	75 celsius
Weight	119g
Insulation resistance	20 MΩ

3.3 Hardware design

3.3.1 Actuator driver design

In order to control the two linear actuators, we require an H-Bridge circuit in between the microcontroller and the motors. For the particular actuators we have selected, the design of the required H-Bridge is shown in Figure 9

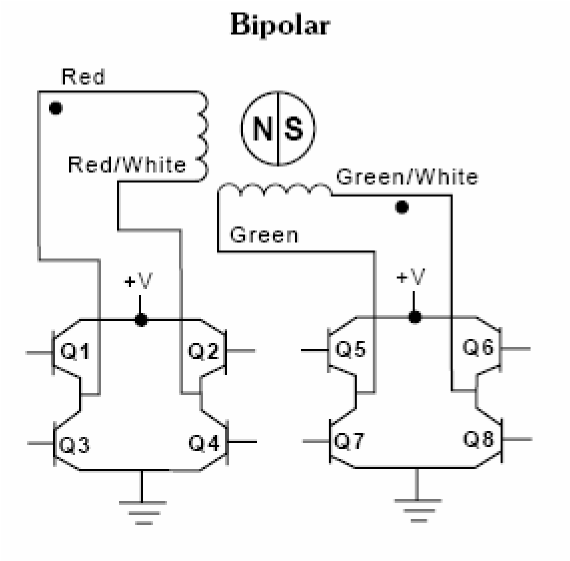
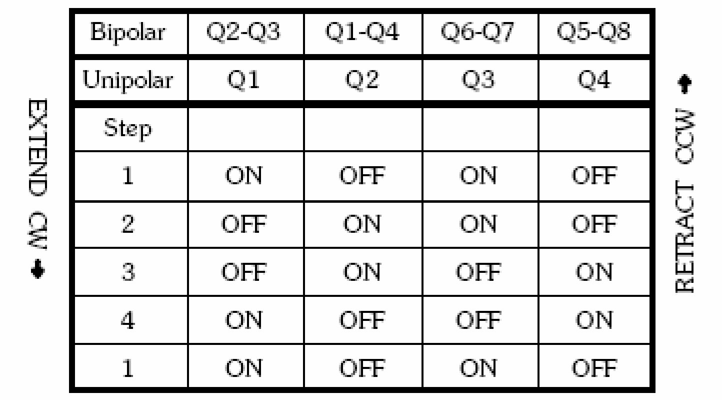


Figure 9: Actuator Wiring [5]

The stepping sequence is specified by Table 2, as shown in the product description [5].

Table 2: Linear Actuator Stepping Sequence [5]



EXTEND CW ↓

Bipolar	Q2-Q3	Q1-Q4	Q6-Q7	Q5-Q8
Unipolar	Q1	Q2	Q3	Q4
Step				
1	ON	OFF	ON	OFF
2	OFF	ON	ON	OFF
3	OFF	ON	OFF	ON
4	ON	OFF	OFF	ON
1	ON	OFF	ON	OFF

RETRACT CCW ↑

In order to implement this we considered two options: designing and building the H-Bridge ourselves using discreet components or using an off-the-shelf driver. The advantage of using first option is that the end circuit will be significantly cheaper. However the disadvantage arises in the time involved for building, testing and debugging the circuit. Considering theses issues we opted to purchase a commercially available driver, the L/R Driver 39105 designed and built by HSI Motors. An additional benefit of using this driver is that it is manufactured by the same company as our selected actuator, and designed to use with their actuators. The key features of the L/R Driver 39105 are outlined below [12].

Table 3: Driver Specification

Input Voltage	7.5 to 32 Vdc
Motor Direction	Selected via on board switch or external control via input/output (I/O) connector
Motor Enable	Enabled/disabled via on board switch or external control via I/O connector.
Step Rate	*Single step operation*: Via on board push button switch *Continuous*: On board or external potentiometer controlled oscillator (10 Hz to 2Khz.) *External Control*: Via I/O connector

This driver provides the option for onboard control via switches or external control via control inputs from a microcontroller. The 39105 is available as a PCB and uses an edge connector to interface with the motors and electronics. The pin-outs are shown in Figure 10.

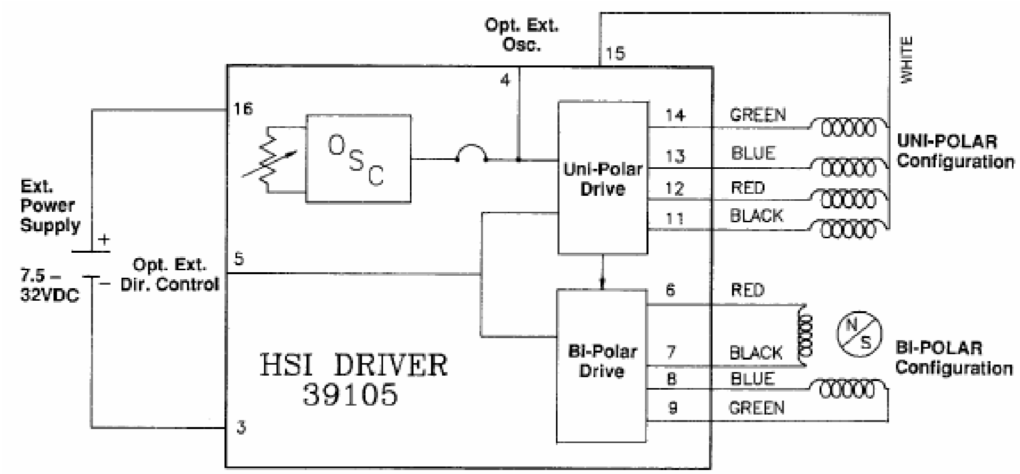


Figure 10: Edge Connector Numbers

It also has an added benefit of a regulated +5V output at pin 1 (not shown in figure above) which can be used to power the other electronic components of the Sure-Step® walker.

3.3.2 Sensor circuit design

The M-Brake uses the Memsic MX2125 dual-axis accelerometer which is capable of measuring dynamic and static acceleration with a range of ± 2g. A key feature in selecting this accelerometer was its packaging. We required a DIP packaging so that it could be easily used with a bread board. Another key aspect was the ability to sense acceleration in a horizontal direction. The MX2125 senses acceleration on two axes and depending on how it's positioned, can cover two of the three, X, Y and Z axes. When positioned one way, it can sense up/down or it can sense forward/backward which is applicable to our project [6]. Figure 11 shows the schematic for the accelerometer

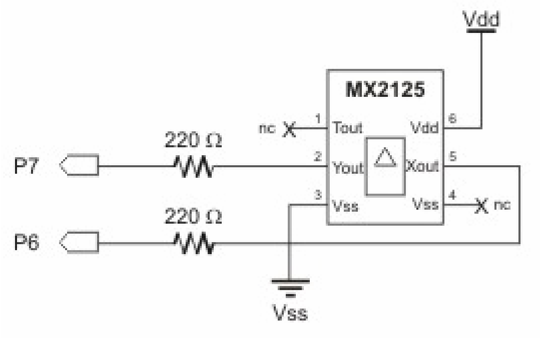


Figure 11: Accelerometer Schematic [6]

Some key features (as listed in the data sheet) are as follows:

- Measure 0 to ±2 g on either axis; less than 1 mg resolution
- Fully temperature compensated over 0° to 70° C range
- Simple, pulse output of g-force for X and Y axis – direct connection to BASIC Stamp
- Analog output of temperature (TOut pin)
- Low current operation: less than 4 mA at 5 vdc

Connecting the sensor is straightforward since it only requires two input pins, one for each axis it is measuring. The analog temperature output will not be used in our project. This particular accelerometer is designed to interface easily with a BASIC Stamp module but we are confident it is suitable with a PIC microcontroller since the readout is a pulse.

In order to calculate the acceleration, we first consider the output pulse of the sensor as shown in Figure 12

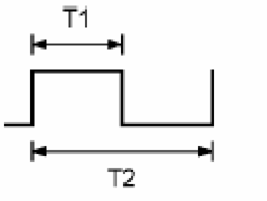


Figure 12: Accelerometer Output [7]

The duty cycle changes in proportion to the acceleration and is set to 50% at 0g. The formula for calculating g force is given by

$$A(g) = ((T1/T2) - 0.5) / 12.5\ \% \qquad (1)$$

The T2 duration is calibrated to 10 milliseconds at 25° C enabling us to convert this output to a corresponding acceleration.

In order to measure the tilt of the walker, we first consider the way we will mount the accelerometer. This is depicted as shown in Figure 13

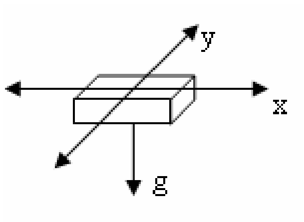


Figure 13: Accelerometer mounting

According to [8], this type of mounting results in good resolution up to ± 60° arc with respect to the x or y axis. To measure the inclination, a calculation is necessary since the accelerometer output simply represents the acceleration of gravity [8]. This relation is given by

$$Ax = g \cdot \sin(\alpha) \tag{2}$$
$$Ay = g \cdot \sin(\beta) \tag{3}$$

where Ax and Ay represent the accelerometer outputs, g is the acceleration due to gravity and α and β are the inclination angles of the x and y axes respectively. Alternatively, the angles can be represented by

$$\alpha = \arcsin(Ax/g) \tag{4}$$
$$\beta = \arcsin(Ay/g) \tag{5}$$

Equations (4) and (5) can be plotted and the resulting graph is shown in Figure 14.

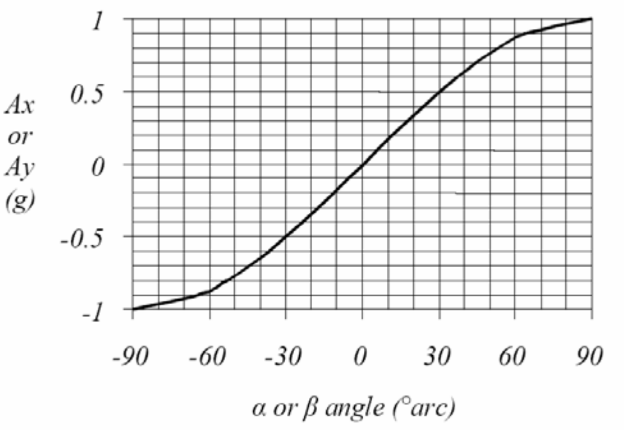


Figure 14: Acceleration vs. Inclination angle [8]

We observe that the range applicable to our project (-20° to 20°), a linear approximation will suffice to calculate the inclination angle according to

$$\alpha = k \cdot Ax$$
$$\beta = k \cdot Ay$$

For this particular accelerometer the manufacturer has calculated and documented the constant k, and angle error when using the linear approximation method. These results are summarized in Table 4

Table 4: Inclination angle error when using linear approximation [8]

Inclination range (° arc)	k (°arc/g)	Maximum error (°arc)
± 10	57.50	±0.02
± 20	58.16	±0.16
± 30	59.04	±0.48
± 40	60.47	±1.13
± 50	62.35	±2.24

As specified, for our maximum inclination of 20° an error of ±0.16° is acceptable for our proof-of-concept design.

3.4 Control Software design

The M-brake™ control software that runs on the onboard computer is the heart and soul of the M-brake™ system. It controls the operation of the brake mechanism by outputting control bits to the actuator driver. The functions of the control software include sampling input from onboard sensors, making decision on the duration and strength of braking, and output corresponding control to the actuator driver. The high level design flow chart of the control software algorithm is presented below.

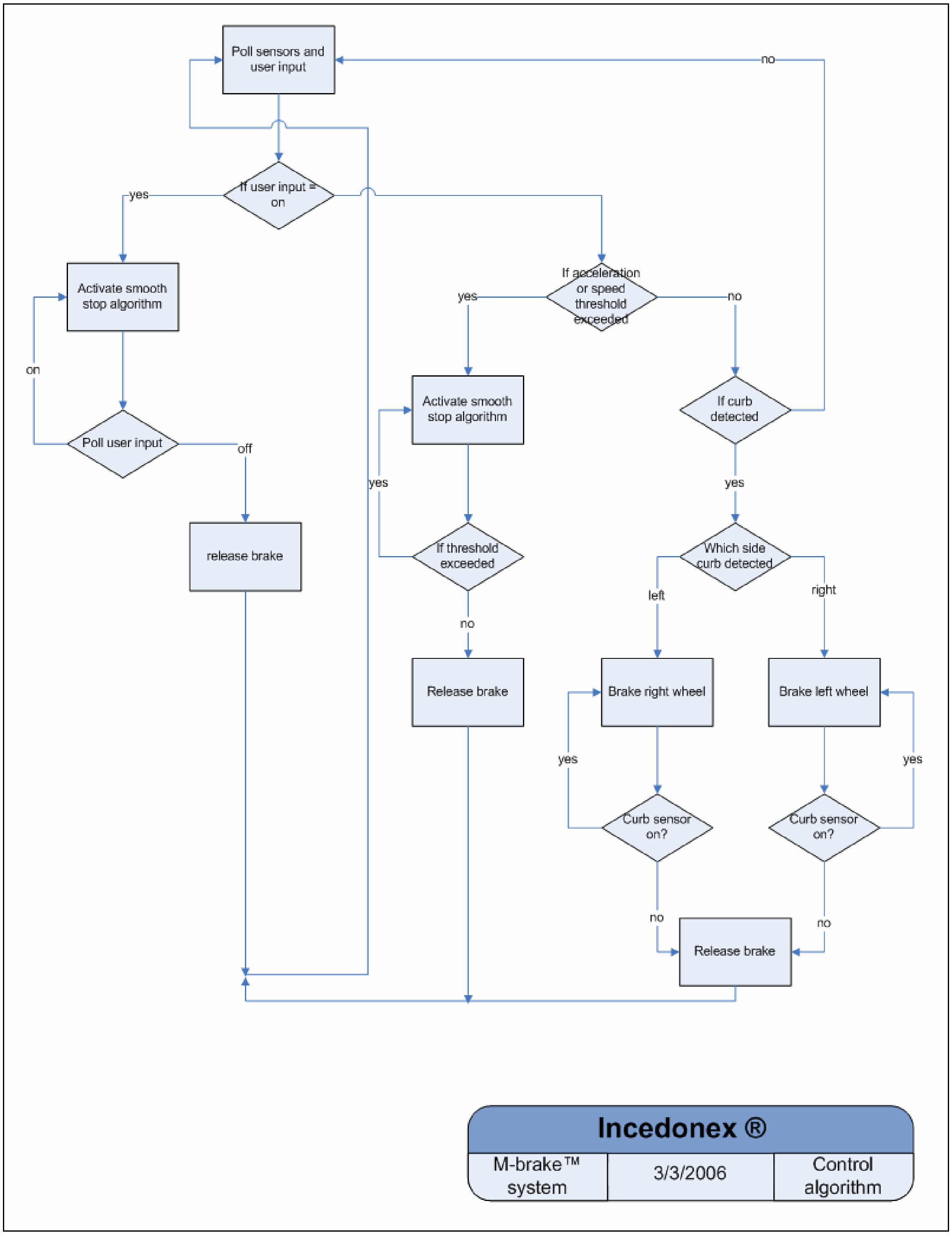


Figure 15: M-Brake™ control algorithm flow chart

3.4.1 Main Control loop

The main program loop polls the state variable array. The state variable array contains three variables: User input button, IR sensor, and Accelerometer. When the user presses the brake button on the handle of the walker, user input is set with highest priority. When accelerometer detects acceleration exceeding preset threshold and/or the tilt angle changes angle bracket (0, 5, 10, 15, 20), accelerometer is set with second highest priority. When the IR sensor detects a curb, IR sensor is set with lowest priority. If the user input is set, the program enters the smooth braking algorithm. If the accelerometer is set, the program enters the acceleration control algorithm. If the IR sensor is set, the program enters the curb correction algorithm.

3.4.2 Smooth braking algorithm

The smooth braking algorithm will run for as long as the user holds down the brake button. When the brake button is pressed down, the algorithm will generate control bits to drive the actuator to travel a certain distance, and hold the position for a certain amount of time. Then it will drive the actuator further press down the brake, and hold for a certain amount of time. The position determines how hard the brake pad is pressed onto the wheel and thus determines the strength of the braking. By varying this position gradually, we achieve variable smoothly increasing brake strength. The hold time determines how long the brake will supply the current resistance force. We hardcode these two variables to change in a preset sequence, in order to follow a smoothly varying braking force curve. The exact value of the variables will be determined experimentally. When the user releases the button, the algorithm exits.

3.4.3 Acceleration control algorithm

The acceleration control loop is triggered by a state variable set from the accelerometer calculation algorithm. The input for the algorithm is two variables: the acceleration and the tilt angle. The loop checks to see if the acceleration has exceeded a hard coded constant threshold value which is determined experimentally. If the threshold is exceeded, the smooth braking algorithm is called. The smooth braking algorithm runs until the threshold is not exceeded. If the tilt angle is changed from one bracket into another bracket, for example, from 5 degrees to 10 degrees, the corresponding preset constant braking force will be applied by sending out control bits to the driver to move the actuator to the corresponding position. When the above two operations are done, the algorithm exits.

3.4.4 Curb correction algorithm

The IR sensor sampling algorithm from Sense-Steer™ detects a curb and generates an interrupt. The curb correction algorithm triggers on this interrupt.

This algorithm has an input from the IR sensor algorithm which specifies whether the left or right curb is detected. If the right curb is detected, the software will send control bit to drive the actuator to brake the left wheel. By doing this, the walker will turn away from the curb. The braking of the left wheel is released when the IR sensor no longer detects the curb. The same operation applies to the left curb. When the IR sensor is no longer triggered, the algorithm exits.

4 Sense-steer™ system

The obstacle detection and avoidance module is used to detect hazards, prevent the walker from tipping over the curb, and avoiding obstacle in front of the walker. Since this document only specifies the proof-of-concept design, only curb detection and avoidance is described. The following figure shows an overview of how of the module work.

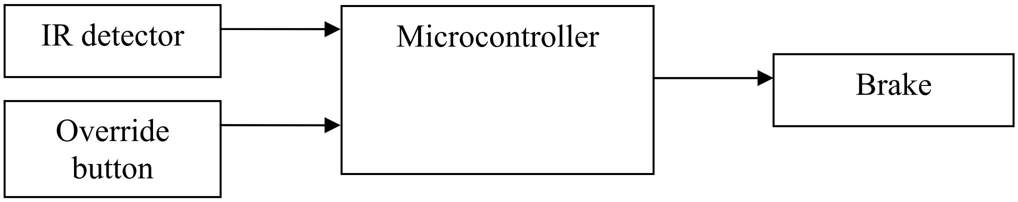


Figure 16: Overview of the curb detection module

IR detector detect when the walker is close to the curb, and sends signal to the microcontroller. The microcontroller then applies the brake to avoid the curb. Once the walker is away from curb, the detector sends another signal to the microcontroller. The controller then stops braking. One note is that the avoidance system ideally would only be activated if the walker is approaching the curb at angle less than 45° with respect to the curb. If angle is greater, then the user is assumed to want to cross the curb, and no action is taken. Because it's difficult for system to act perfectly, a user override button is added, so user can easily bypass the avoidance system. A more detailed description is presented in the following sections.

4.1 Infrared detector

The Infrared detector is used to detect a curb. A curb is essentially a difference in height between sidewalk and road. Thus, the sensor should be able to detect a difference in distance.

When choosing the detector, other detectors considered include ultrasound, capacitive and inductive sensors. However, ultrasound is more expensive than infrared, and provides sensing range that is too large for the application.

Capacitive detectors detect distance based on material's dielectric constant [9]. Thus, if dielectric constant changes, then measured distance also changes. For curb detection, the pavement is sometimes wet, and thus the distance measured will be different. Thus, capacitive sensors aren't reliable enough for the application.

Inductive detectors are only for metallic materials, and thus are not suitable for this application [9].

The particular infrared sensor selected is Sharp GP2D15 sensor. The sensor outputs logical high when an obstacle is within 24cm of sensor, and low otherwise. The sensor's dimension is approximately 40x14x13mm. Its small size is suitable to be mounted on the walker. The sensor's datasheet is protected by Adobe Reader. Thus, only Internet link is provided for the datasheet. Please refer to [10]

A concern with IR detector is the absorption and reflection of material. Indoor testing of the Sharp sensor shows fairly consistent result, but outdoor usage may cause new problems. An alternative may be needed if the detector cannot perform outdoor.

Two Sharp detectors are mounted on both sides of the walker as shown in the figure below.

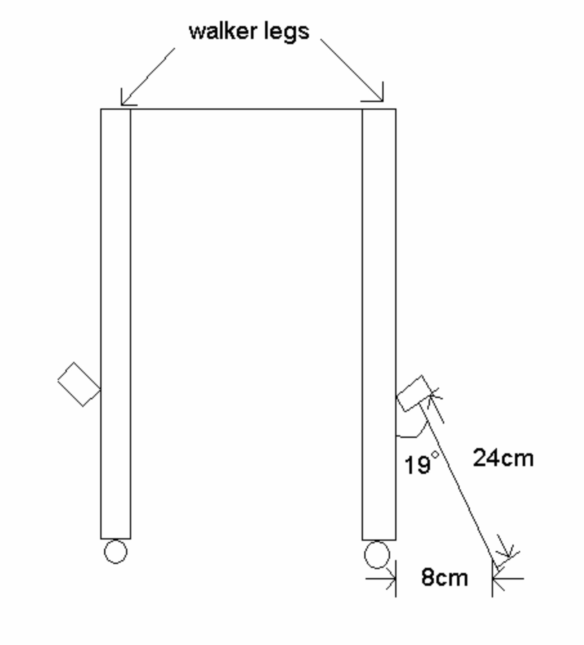


Figure 17: Mounting of IR sensors on the walker

Two detectors allow the walker to detect curbs on both side of the walker. The detectors are mounted on the legs, and make 19° angles with the legs. Because its detection range is 24cm, the detectors are able detect a curb 8cm away from the walker. When no curb exists within 8cm of the walker, the detector outputs logical high or "1" because an obstacle, the pavement, is detected. When the curb exists within 8 cm, the detector outputs low or “0” because no obstacle is detected due to height drop caused by the curb. The exact orientation of the detector on the walker needs to be experimentally found, so that the detector outputs correct readings at 45° to curb.

For a proof-of-concept design, glue, tape, and paper are used to mount the detectors in a quick and crude approach. For a production design, screws and metal holders will be used. Since this document only deals proof-of-concept, a detailed mounting procedure is not discussed.

The detector requires one external resistor, and 5V power supply. The output from detector can be directly fed to microcontroller. The schematic is shown in the following figure.

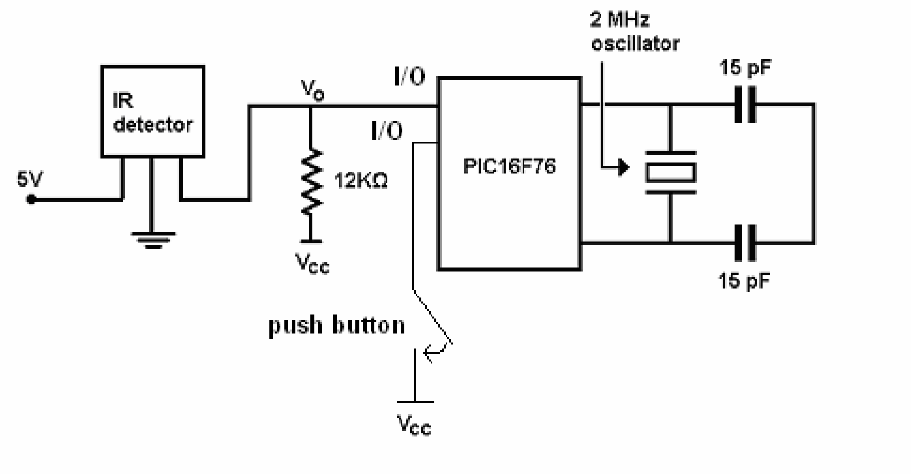


Figure 18: IR Detector Connections

User inputs through a push button that connects to the microcontroller’s I/O pin. The button is mounted on the handle of walker, so the user can easily press it. As the above figure shows, the push button is modeled as a switch that when pressed, connects to logic high voltage, and otherwise is logic low. Thus, when user presses the button, logic low becomes high at microcontroller’s I/O pin, and an interrupt is generated. There are many push buttons available. A suitable button can easily be found depending on factors such as mounting and size, and thus a particular button is not described here.

4.2 Sensing algorithm

Microchip PIC6F76 microcontroller is used to process the output signal from the IR detector and control the brake. The reason for choosing this microcontroller is its wide availability, and available programmer and software in Engineering Lab 1 at Simon Fraser University.

The following figures show the microcontroller’s algorithm for curb detection.

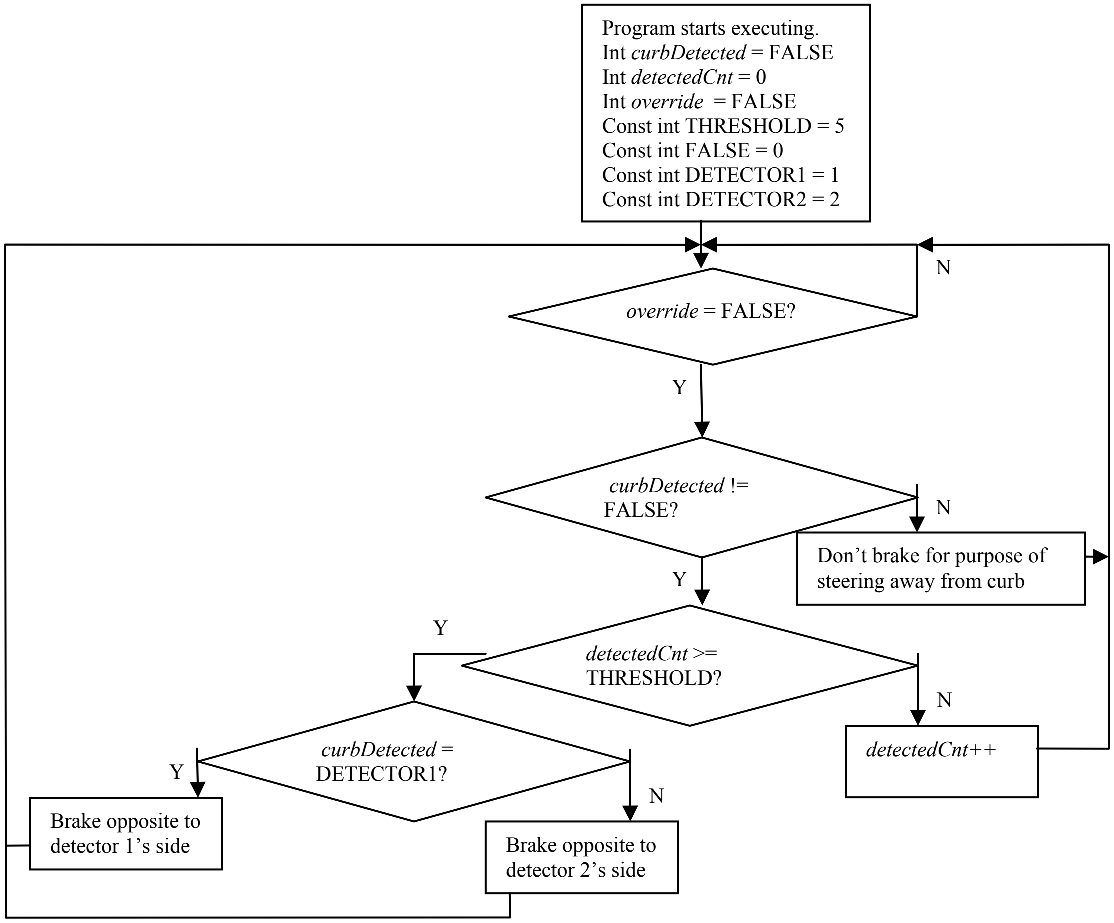


Figure 19: Main program algorithm on microcontroller for curb detection. Note any time during program execution, the program can be interrupted by an interrupt service routine (ISR). The ISR mechanisms are shown in next figure.

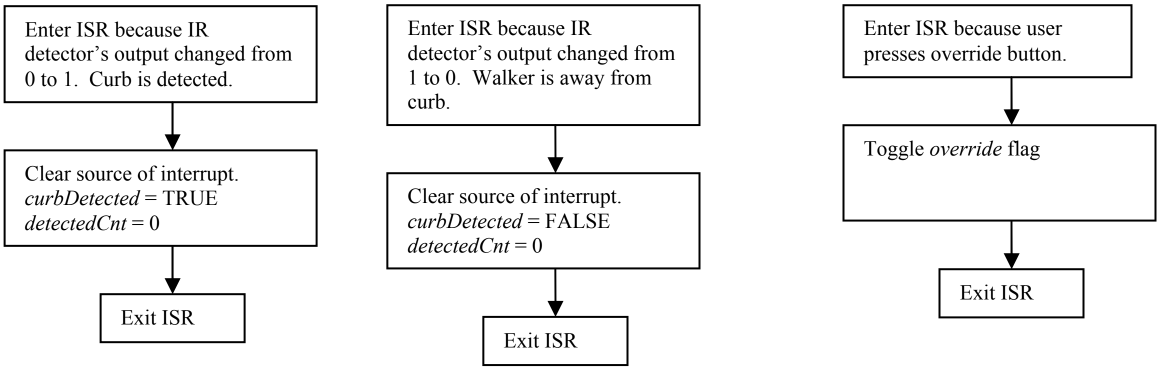


Figure 20: ISR mechanisms for curb detection. There are 3 possible ISR sources. One for moving into curb, one for moving away from curb, and one from user.

As the above figures show, the program for curb detection is essentially an infinite loop. The program starts out assuming the walker is not near a curb, and then continuously checks the *curbDetected* flag to see if the walker is near the curb. In order to filter out random noise that may cause the detector to change from 0 to 1, which would falsely indicate the walker is near the curb, *detectedCnt* is used as a counter that measures the amount of time that the detector is reading 1. If the amount of time has past a threshold, then we can be fairly certain that the walker is near a curb, instead of erroneous readings.

Once the program determines the walker is near a curb, it engages the brake on the opposite side from the detector that's detecting the curb. Because only 1 wheel is braking, as a user pushes the walker, the walker will turn in the direction opposite to the curb, thus, steering the walker away from the curb. Once the walker is sufficiently away from the curb, the microcontroller stops braking.

Interrupts are used to signal when the walker is moving closely to the curb, or away from the curb. Two different interrupt service routines (ISR) are invoked when the detector changes from 0 to 1, and from 1 to 0. For 0 to 1 case, *curbDetected* flag is set to true to indicate walker is close to the curb, and for 1 to 0 case, *curbDetected* flag is set to false to indicate walker is away from the curb. ISR interrupts main program execution to inform the program of curb detection status. Another ISR comes from user, and it toggles *override* flag. The flag is used to determine if the avoidance system should be activated.

4.3 Brake interface

To estimate the braking force needed to avoid tipping over a curb, several assumptions are made. The maximum speed of user is approximately the average human walking speed which is approximately 1.3 meters/second. This assumption is also discussed with reference in function specification. The maximum angle at which the walker approaches the curb is 45º with respect to the curb. The reason is that for angles greater than 45º, the user is assumed to want to past the curb. Thus, no intervention is taken for angles greater than 45º. Given the assumptions, we can design the braking force for the worst case scenario, which is 1.3 meters/second walking speed at 45º. For any other case, the braking force needed is less. Because analytically and accurately calculating the braking force is difficult, we will experimentally determine the braking force given the assumptions.

5 Nav-Pro™ System

5.1 Positioning and Directional Module

To determine the location of the walker, the Nav-Pro™ system will use a series of wireless beacons to provide information to the walker. RFID and a wireless solution have been considered for this application. RFID is very good solution for this application as it features the following advantages:

- Low power usage (using passive tags)
- Low production costs.

However, high development kit costs and lack of pre-made modules have led us to choose the wireless solution. In addition, with the wireless system, other information such as current temperature, advertisement information can be sent to the walker as an add-on to the ID information.

The direction at which the walker is heading is dictated by the reading of the digital compass that is inserted into the Nav-Pro™ system. The digital compass being used is a solid state hall effect device capable of detecting eight directions, and four of which (N,W,S,E) will be employed as part of the path finding algorithm to determine the optimal path for the user [13]. At the output of the algorithm, four LED lights are used to signal which direction to turn for the user. Both the positioning directional controls are programmed on the same PIC, and thus will share the battery and memory.

5.2 System Architecture

The position system wireless beacons will be comprised of several MMcc1000 modules placed in different locations, each beacon having a unique ID.

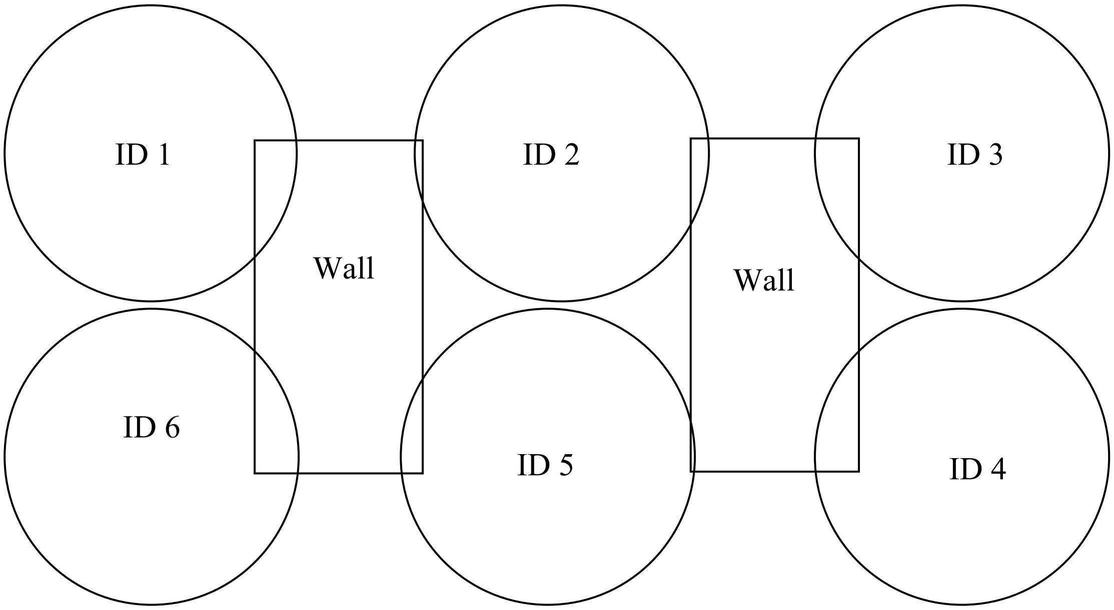


Figure 21 System architecture diagram. The circles denote the range of the each wireless beacon.

Figure 21 shows the location of the beacons and the different IDs each beacon will have. The walker MMcc1000 module constantly transmits a request for ID every 5 seconds. If the walker is within range of the beacon, the beacon will receive this request and reply with its own ID. After receiving this ID, the walker wireless module will send this information to the path finding software for further processing. Using this arrangement, the most power consuming part of the processing is focused on the walker module, which will have a battery with more capacity. With the beacons in sleep mode for most of the time, this can increase the battery life and maintenance issues that arise because of a limited battery life. The system will operate at a frequency of 433 MHz. This frequency band is not regulated in many countries and is allowed for unrestricted use, as long as the transmit power is below 10 mW [11].

5.3 MMcc1000 Wireless Module

The MMcc1000 wireless module is a wireless module pre-built by Propex that is used for wireless communications. The module is based around the Chipcon CC1000. The module offers the following features:

- Low supply voltage: 2.1 – 3.6V
- Programmable output power
- Programmable transmit/receive frequency
- Software programmable for all configurations
- 3 wire connection with microcontroller for communications

For this initial implementation, the modules will be powered by a lithium watch battery. Other power sources of power, such as solar panels and power from the input RF waveform, has been considered and is under testing for possible use.

5.4 Microcontroller - 18LF4520/18LF2520

To interface with the wireless controller, we will use Microchip's PICmicro family of microcontrollers. The PICmicro family offers many built-in features such as onboard A/D converter, onboard serial communications support, and a simple instruction set. In addition, the PICmicro is low cost and development tools are readily available in the Engineering Science labs. For the wireless modules, a special version of the controller is used. The 18LF designation allows the microcontroller to function with a V_{dd} from 2.0V to 5.5V. This function will help lower the power consumption of the chips. For the beacons, a smaller version of the 18LF, the 18LF2520, will be used. The 18LF2520 is identical to the 18LF4520 except that the former has less I/O pins and a smaller physical size. To save power during idle periods, the 18LF microcontroller has several power down modes that can be used to reduced power usage in idle periods. Table 5 lists the available modes on the 18LF microcontroller.

Mode	CPU Clocking	Peripheral Clocking	Description
Sleep	Off	Off	All CPU functions are off, except for sections needed for wake support.
PRI_RUN	On	On	Normal full power mode.
SEC_RUN	On	On	Use the Timer1 oscillator and shutdown primary oscillator to save power.
RC_RUN	On	On	Use internal RC clock for clocking. No power savings compared to PRI_RUN.

PRI_IDLE	Off	On	Fastest wakeup time. Using primary oscillator
SEC_IDLE	Off	On	Using Timer1 oscillator as clock for peripherals.
RC_IDLE	Off	On	Use internal RC clock for peripheral clock.

Table 5 Power saving modes on the 18LF microcontroller.

5.5 Software Implementation

The software program to control the wireless module will make use of the PICmicro's built-in SPI communication modules. The SPI interface makes use of three data lines, Serial Data Out (SDO), Serial Data In (SDI), and Serial Clock (SCK). This link will be used to configure the device on startup and also to receive and transmit data.

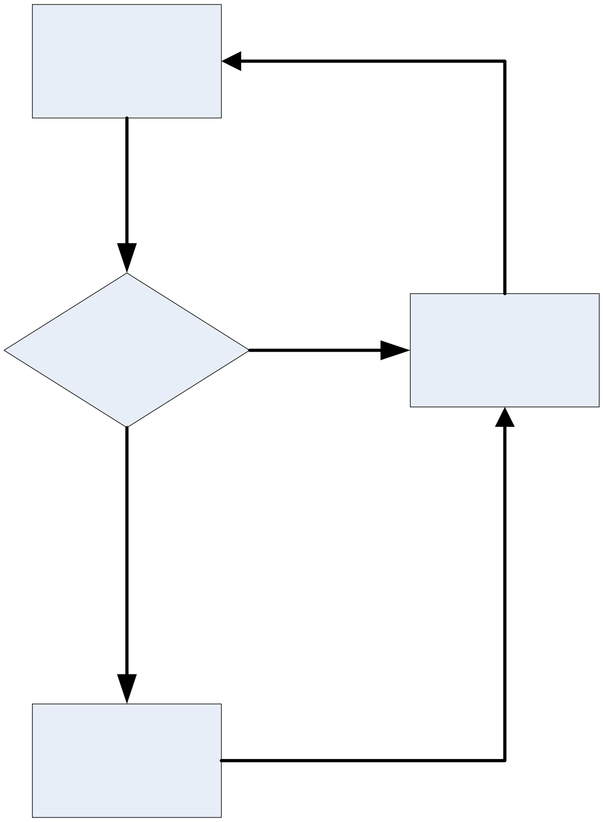


Figure 22 Flow chart for beacon software

Figure 22 shows the general flow chart for the beacon software. The sleep period is currently set to 500 ms, however, this number may change with testing of battery life versus performance.

5.6 Path Finding Algorithm

The shortest path algorithm requires three pieces of information in order to execute. The direction of the walker is pointing, generated by the digital compass's two terminals, the positioning of the walker, and the destination position set by the user. The two terminals are sufficient for four possible directions provided by the output of the digital compass. The digital compass circuitry as well as the positioning of the walker together comprises the Data Acquisition unit in the path finding algorithm.

Once the direction and the position of the walker are determined, the information is passed to the Shortest Path Algorithm unit to compute the shortest path. This unit actually uses only the destination position and the current position to determine the next position the walker needs to be at, which is denoted by the beacon ID.

The direction of the walker is pointing, as well as the current and next position beacon ID are then passed to the final for processing, the Path Processing unit. This unit will compute the direction the user needs to turn to arrive that the next position beacon. At the output of the Path Processing unit, one of four directions is specified, and the final output is displayed as one of the LED lights shown to the user. Figure 23 below illustrates the relationship of each unit.

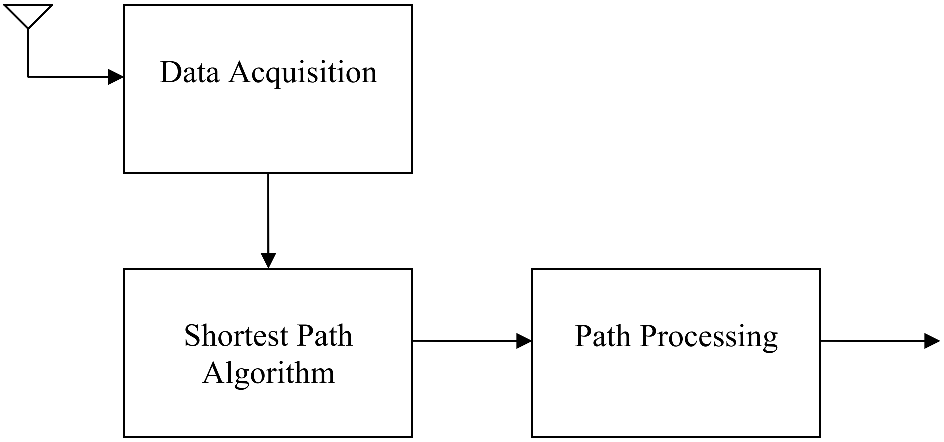


Figure 23 Shortest Path Algorithm Flow chart

To facilitate functioning of the units, the following C-programming functions are required:

int shortestPath(int pos, int dest)	- returns the next beacon ID, as determined by the shortest path algorithm - the inputs are the current position beacon ID and the destination beacon ID
pathDirect(int pos, int dest, int dir)	- returns the direction change the walker for the walker - inputs are current and destination beacon IDs, and current direction pointing

In the *shortestPath* function, information about the floor plan must be pre-programmed. That means the relationships of interaction between all beacons are stored. Denoting the path between two beacons as a branch, and assigning a weight on each branch, where the weight is a function of the length of the branch. The information is pre-programmed, and is used by the shortest path algorithm. The shortest path algorithm first starts at the current beacon ID, then search through all its adjacent neighbors and keep track of its branch weights. If the adjacent neighbor is the destination, the algorithm is completed. If not, the branch weight is stored, and for each adjacent beacon ID, we continue the process, until the destination beacon ID is found. Once the destination beacon ID is found, we trace back the path taken with the least accumulated branch weights, and the output the beacon ID that is directly next to the current beacon ID.

The *PathDirect* function also requires pre-programmed information stored before execution. The information involves keeping the direction of the neighboring beacon IDs for each beacon for the prototype design. When the current direction of the walker is passed into this function, this function will compute whether to turn left, right, straight, or backwards, to reach to the direction that leads to the next position beacon ID. This new direction is then converted to the global direction of N, E, W, or S, and will be outputted.

5.7 Direction Control

When an LED light is on to signal a direction change, the shortest path algorithm is invoked, and a new direction is specified to the user. This is inevitable because when the user changes direction of the walker, the digital compass provides a new reading, and thus the new direction will be fed into the shortest path algorithm to generate a new direction to the user. As shown in Figure 24,

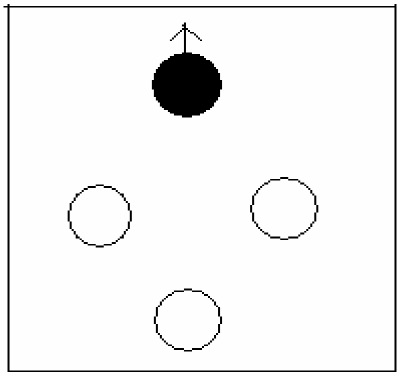


Figure 24 Orientation of LED lights

6 Power Unit Design

The power unit is required to supply power to each of the three modules. The Nav-Pro™ requires 5V for the microcontroller and 3.3V for the wireless module. The Sense-Steer™ and M-Brake™ share a microcontroller which also requires 5V power. Finally, the actuators require +12VDC power. We plan to use a 12VDC 800mAHr battery. This will be connected to the L/R 39105 drivers which will control the actuators. The +5VDC outputs available on pin 1 from these drivers will be used to power the two microcontrollers on the walker. In order to provide 3.3V for the wireless modules, we will build a simple op-amp circuit using two TL084 op-amps configured according to Figure 25.

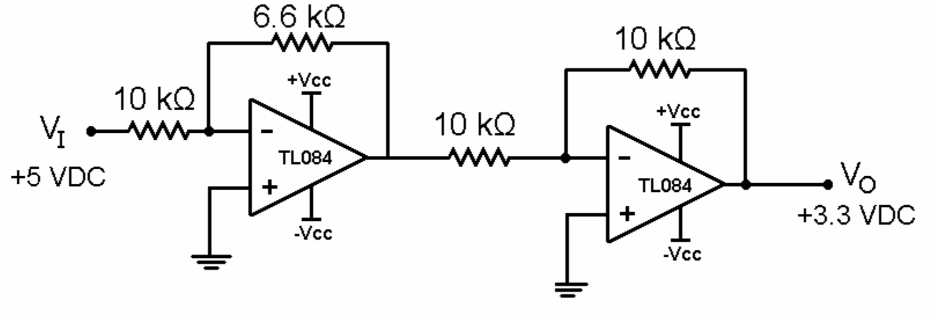


Figure 25: Inverting Op-Amp configuration to provide 3.3 V

Here, V_I is supplied from the 5V output of the driver. The voltage output from the first op-amp is

$$V_O = -\left(\frac{6.6k\Omega}{10k\Omega}\right)V_I = -3.3V \qquad (6)$$

The second op-amp is configured as an inverting buffer with an output, $V_O = -1V_I$. Thus the output of the entire circuit is +3.3V. This will be used to power the wireless modules in the Nav-Pro™ system.

7 Test Plan

7.1 M-brake™ Module Test plan

The following sections outline the test plan for the M-Brake™ module

7.1.1 M-brake™ Sensor Unit Test Plan

The test plan for the M-brake™ sensor unit will evaluate performance in two fields: acceleration detection and incline or decline angle detection.

First, we will verify the operation of the sensor itself by attaching the M-brake™ sensor to the walker, and then monitoring the sensor output with an oscilloscope.

i. The empty, still walker will be manually pushed with three different forces on a smooth plane, and the different pulses of the output will be observe on the oscilloscope. These output signals should each have a different duty cycle as per specified by the data sheet. Equation 1 will then be used to approximately calculate the resulting acceleration of each push

ii. We will place the walker on planes with angles of -20°, -10°, 0°, 10°, and 20° with respect to level ground and let walker move itself without external force. We will observe readout similar to case i and approximately calculate the acceleration and then the angle using equations 4 and 5. The sensor operation will be verified if the calculate angle is within ±10% tolerance of actual their actual value.

Next we will connect the sensor X and Y outputs to I/O pins on the microcontroller and quantify the readouts. We will use these values in various scenarios and use equations 1, 4 and 5 to verify that these outputs are indeed accurate.

iii. Similar to point i) the empty, still walker will be manually pushed with three different forces on a smooth plane. However, this time, the outputs will be read and stored by the microcontroller. Using equation 1 we will calculate the acceleration, and verify the accuracy of the outputs.

iv. In order to verify the incline measurements, we will subject the walker to the same scenarios as specified in point ii). Using equation 4 and 5 we will calculate the five incline angles and will compare the outputs to the actual scenarios of -20°, -10°, 0°, 10°, and 20°.

By carrying out the preceding steps we verify sensor operation and the integrity of the output signals to be used in the software control algorithm.

7.1.2 M-brake™ Control Algorithm Test Plan

To test the control algorithm, the software will first run on simulator. Then the software will be run on the microcontroller with test bed connect to the microcontroller so that test signals can be driven into the pins of the microcontroller manually. The test cases to carry out are listed below.

i. User input is set, actuator extension bit will be observed.
ii. Acceleration threshold bit is set, actuator extension bit will be observed.
iii. Tilt angle bit is set, actuator extension bit will be observed.
iv. Right curb bit is set, actuator extension bit will be observed.
v. Left curb bit is set, actuator extension bit will be observed.
vi. All input bits are cleared, actuator retract bit will be observed.

7.1.3 M-brake™ Actuator Unit Test Plan

The actuator executes the braking action on the wheels. The test plan for this component will verify the operation of the actuator itself as well of the motor driver circuit used to interface with the microcontroller.

i. First, we will verify the actuator operates correctly by connecting it to a 12 VDC power supply and switching the current flow through the actuator to verify its movement. We will then verify whether or not the actuator can provide the required force to fully stop the moving walker.

ii. We will build the required H-Bridge and connect it to the microcontroller and actuator. Using Table <_> we will verify each test case and thus verify the operation the H-Bridge

7.1.4 M-brake™ System Test Plan

This test plan is to evaluate the performance of the complete M-brake™ module, so the subsequence step will be carried out precisely.

i. Maximum acceleration test: The walker is pushed from a state of rest with acceleration greater than the threshold acceleration. The test will be success if the M-Brake™ is automatically activates the braking mechanism to keep the acceleration at the threshold level.

ii. Maximum speed test: The system will always update the current speed. The walker is maintained at speed of 1.3 meter/second no matter how hard we put the additional push force on. The successful test will verify the activation of the M-Brake™ system and its ability to maintain the threshold speed.

iii. Anti-roll test. The walker will be able to stay stationary due to the assistance provide by the M-Brake™ when walker is placed on an incline or decline of 20° with no additional weight imposed on it. The walker must also abide by the conditions specified in points i and ii if external force is applied. The test will then be carried out on 10°, 5°, -5° and -10° inclines.

iv. User interface test. There are two different situations M-brake™ should handle. Firstly, the tester will push the walker around on level ground. The successful test will show that the tester will feel the smooth braking force applied on the walker and the walker will eventually fully stop once the tester activate the braking by using the input device for the brake. Secondly, the tester will push the walker around on slopes. The walker should be brought to smooth stop and remain stationary for as long as the tester manually activate the braking system by pressing down the input device.

v. Walking without brake. The tester should not feel the effect of the braking system, when the tester pushes the walker on flat and inclined road under condition that both speed and acceleration are below the threshold values.

7.2 Sense-steer™ Module Test Plan

The Sense-Steer™ test plan discussed in this section is emphasized on the module's functionality to ensure it operates correctly.

i. To ensure that the walker will be able to avoid the curb in time, a user will push the walker at average human walking speed at 1.3m/s towards the curb at 15°, 30° and 45° with respect to the curb. The test is repeated with walking speed of 0.6m/s and 0.3m/s. The walker must be able to successfully move away from the curb.

ii. To ensure that walker is able to cross curb when a user intends to, the user will push the walker at 1.3m/s towards the curb at 60°, 75°, and 90° with respect to the curb. In 90° case, the walker is moving perpendicular towards the curb. The test is repeated with walking speed of 0.6m/s and 0.3m/s. The walker must allow the user to walk towards the curb without interference.

iii. To ensure override button works, user is to walk towards curb in the same conditions as in curb avoidance test, but when walker is 6cm from curb, the user activates the override button. The avoidance system should then be deactivated. The user then deactivates the button, and the avoidance system should be activated.

Passing the above tests indicates the module functions correctly.

7.3 Nav-Pro™ Module Test Plan

The position subsystem requires the following tests to be performed:

i. Verify that the correct ID is sent by the beacons and received by the walker module.

ii. Verify the operation range is approximately 2 meters.

The direction of the subsystem requires the following tests to be performed:

iii. Verify that when the user has changed direction of the walker, the direction provided by the subsystem will update to the direction relative to the user's current direction.

iv. Verify at all regions covered by the beacons, the reading of the digital compass is accurate. The test is repeated at each beacon to make sure all beacons can be read accurately.

v. Verify that the shortest path algorithm is indeed the shortest path to the specified destination. The test is repeated at random starting locations to test each destination.

vi. Verify that the shortest path algorithm is providing to the user the correct destination. The test is repeated at random starting locations.

vii. Verify that digital compass is performing accurate and outputting the correct direction the walker is pointing at. The test is repeated at each direction to make sure the digital compass is providing accurate readings.

8 Conclusion

The design for The Sure-Step® walking aid is defined in this document, including the M-Brake™ module, the Sense-Steer™ module, and the Nav-Pro™ module. We are currently in the process of implementing each individual module. During the implementation progress, we may change our current design to achieve better functionality for our project. We expect to have a functional proof-of-concept by the end of April, 2006. Optimization of the prototype will be pursued further thereafter.

9 References

[1] World Health Organization. 2006. *Elderly People: Improving oral health amongst the elderly*. Jan. 18, 2006 <http://www.who.int/oral_health/action/groups/en/index1.html>

[2] National Center for Injury Prevention and Control. Sep. 9, 2005. *Falls and Hip Fractures Among Older Adults*. Jan. 19, 2006 <http://www.cdc.gov/ncipc/factsheets/falls.htm>

[3] Incedonex project proposal for the Sure-Step® intelligent walking aid

[4] Incedonex Functional Specification for the Sure-Step® intelligent walking aid

[5] HSI Motors. *2800 Series Size 11 Linear Actuator* Product Brochure <http://www.hsimotors.com/PDF/Hybrids-Size-11-2005.pdf>

[6] Parallax. *Stamps in Class Memsic Tutorial* <http://www.parallax.com/dl/docs/prod/compshop/SICMemsicTut.pdf>

[7] Parallax. *Memsic Demo Kit Documentation* <http://www.parallax.com/dl/docs/prod/acc/memsickit.pdf>

[8] Memsic. Application Note Inclination Sensing with Thermal Accelerometers

[9] Solarbotics.net *Proximity sensors* <http://www.solarbotics.net/library/circuits/sensors_prox.html#ir>

[10] *Infrared Sensor Datasheet* <http://www.active-robots.com/products/sensors/sharp/gp2d12_15.pdf>

[11] White Paper - *Part 1: Active and Passive RFID: Two Distinct, But Complementary, Technologies for Real-Time Supply Chain Visibility*

[12] HSI Motors. *L/R Driver Manual Model 39105* <http://www.hsimotors.com/PDF/39105LRmanual.pdf>

[13] 1490 Digital Compass, 2006 <http://www.imagesco.com/articles/1490/01.html#fig2>