



# Project Proposal

Motorized Walker Project

Phase 1

By

Robert Pinkerton

Jay Zhuang

Rachel Hua

December 2011 – February 2012

## EXECUTIVE SUMMARY

The motorized walker, comparing to the traditional walkers, is designed to provide additional sensory information, cue the walking speed appropriate for the environment, engage haptic reflexes to control gait, increase mobility, prolong activity, reduce fear of falling, improve quality of life and reduce falls and associated cost and burdens in elderly individuals and individuals with Parkinson's disease. Our purpose is to complete the current model of the automated walker to perform successfully using sensory data and joystick controls.

## OBJECTIVE

To complete the current model of the automated walker to perform successfully using sensory data and joystick controls. Our objective is to complete development of the embedded Arduino code that enables the Automated Walker to perform with the utmost efficiency. As it stands the current embedded Arduino code does not allow the walker to function properly via joystick controls and sensory input. The current version of the software does not interpret the sensory data correctly and the walker is unable to move autonomously. Also the current code does not translate the position of the joystick effectively causing the walker to move poorly. We will produce, an embedded application that properly makes use of the sensory data, making the walker fully autonomous. The application would allow the walker to move along a straight wall and a curved wall whilst still enabling the user to have manual controls via a joystick. We will also adjust the joystick controls to be more effective and combine their functionality with that of the sensory data.

Whether this project is successful would be based on the results of the following test results:

- A. Have walker maneuver through a maze, time trial.
- B. Have a person with Parkinson's disease use device and deliver feedback of use.

## PHASE 1: DESIGN & IMPLEMENTATION

### 1. AUTOMATION AND SENSORY INPUT (MODE 1)

Rewrite the current embedded Arduino code to have the walker be fully autonomous. The walker will use data from the current Sharp IR sensors to determine its position and follow a wall. Our application will allow the walker to autonomously move along a straight and curved wall.

#### STRAIGHT WALL TEST

Enable the walker to move along a straight wall using just sensory data.

#### CURVED WALL TEST

Enable the walker to move parallel to a funky curved wall using sensory data.

#### CORNER TEST

Enable the walker to turn around a corner using sensory data.

### 2. FIX JOYSTICK CONTROLS (MODE 2)

Adjust the Arduino software to interpret the joystick controls position into motor control more effectively. The current joystick software model interprets the location of the joystick irrationally and needs to be adjusted.

### 3. SPEED CONTROL

Reduction of average speed to approximately 1m/sec

The current average speed of the walker is too fast, needs to be reduced to something more comfortable.

### 4. COMBINE MODE 1 AND MODE 2 (MODE 3)

We will integrate the MODE 1 and MODE 2 together allowing the device to function in both modes simultaneously. The controller would override joystick commands based on specific conditions registered by the sensors.

NOTE: Testing of Ultra Sonic sensors may be needed depending on if our software is limited by the current IR Sensors specs.

## PHASE 2: DESIGN IMPROVEMENTS AND ADDITIONAL FEATURES

UPON COMPLETING OF PHASE 1 WE PROPOSE PHASE 2 ASSIGNMENTS THAT OFFERS YOU SOME POTENTIAL DESIGN IMPROVEMENTS AND ADDITIONAL FEATURES.

### 1. DEVELOP FEEDBACK SYSTEM

#### 1.1 Feedback Regulation

Implement a system that allows the walker to adjust its own speed automatically based on ambient sensory input. Device should be able to recognize Doors, Slopes, Stairs and other such obstacles.

#### 1.2 Manual speed control feature

Implementation of a control/system that allows manual speed adjustment of the walker.

### 2. IMPROVE POWER CONSUMPTION

#### 2.1 A Wall battery charger

Implement a way for the walker battery to be charged via a wall outlet.

### 3. IMPROVE RESERVE SPEED

Current motor reverse speed is too slow.

## COMPONENTS

### 1. Software:

- a. Complete Embedded Arduino Code that provides effective MODE 1 and MODE 2 implementation
- b. Extensive Arduino Code that provides MODE 1 and MODE 2 combination allowing user to operate both simultaneously uses the joystick as a manual override.

### 2. Hardware

Battery Charger \$40 (<http://www.amazon.com/Schumacher-SE-1-Trickle-Battery-Charger/dp/B000H94F7S>)

Ultra Sonic Sensor \$30 ([http://www.robotshop.com/parallax-ping-ultrasonic-sensor.html?utm\\_source=google&utm\\_medium=base&utm\\_campaign=jos](http://www.robotshop.com/parallax-ping-ultrasonic-sensor.html?utm_source=google&utm_medium=base&utm_campaign=jos))

## STATEMENT OF CONCEPT

- a. First we must trace through and fully understand the current embedded software written by Adam Larkin as a starting point.
- b. Once we fully understand the walker design, we will begin reengineering the software. We will adjust the methods and algorithms to be more effective.
- c. We will rewrite the joystick control, MODE 1, functions to be more effective, resolving the issues described above.
- d. We will the begin writing the software that allows the walker to perform autonomously. MODE 2.
- e. Upon completion of the software model we will test the device along a straight wall and a curved wall
- f. We will integrate the MODE 1 and MODE 2 together allowing the device to function' in both modes simultaneously. The user would be able to control the walker using the joystick as a manual override to the sensory data controls.

## TIME LINE AND BUDGET

Time	Tasks	Budget (Hours)
November 29-December 10	Concepts a	10
December 10-24	Concept b	15
January 4-February 16	Concept c&d	40
January 14-23	Concept e&f	35
<b>TOTAL HOURS:</b>		<b>100</b>

## Specific Timeline:

### Task

#### ▼ 1) phase 1: preparation

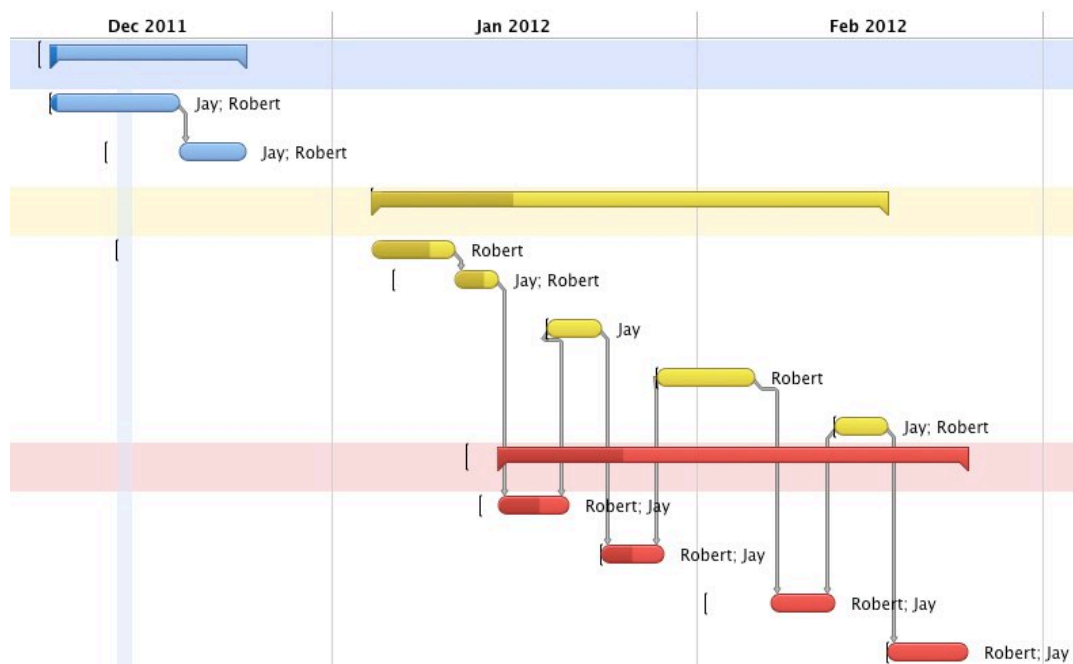
- 1.1) understand the current embedded software
- 1.2) adjust methods and algorithms

#### ▼ 2) phase 2: developing

- 2.1) fix joystick control
- 2.2) write code to have walker move along straight wall
- 2.3) write code to have walker move along slightly curved wall
- 2.4) write code to have walker make 90 degree turn
- 2.5) combine all four movements together if possible

#### ▼ 3) phase 3: test

- 3.1) Test 1 : straight wall
- 3.2) Task 2: curve wall
- 3.3) Task 3: 90 angle wall
- 3.4) Task 4: ending test



## CONTACT INFORMATION

**Robert Pinkerton**

[rpinkert@gmail.com](mailto:rpinkert@gmail.com)

(631) 943-7937

**Jay Zhuang**

[jayzhuang1990@gmail.com](mailto:jayzhuang1990@gmail.com)

(917) 310-8295

**Rachel Hua**

[rachul49cs@hotmail.com](mailto:rachul49cs@hotmail.com)

### **Google Group Discussion Board**

Visit this group's homepage:

<http://groups.google.com/group/Walkergroup?hl=en>

Email this group:

[Walkergroup@googlegroups.com](mailto:Walkergroup@googlegroups.com)