March 8, 2010

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

***Re: ENSC440 Design Specification for the Robotic Item Retrieval System***

Dear Dr. Rawicz,

Attached is a document from Freedom Innovation Research describing the design specification for the Robotic Item Retrieval System. This document provides detailed technical design guideline for the entire system. The Robotic Item Retrieval System is an automated system that can help disabled people to retrieve item at home.

The design specification will list the high level system design for the robotic item retrieval system. It specifies the design details in the following area: mechanical design, eletrical design and software design. Some detailed graphical description will be displayed to illustrate the concept of design.

Freedom Innovation Research (FIR) consists of five motivated and innovative people: Steven Choi, John Ogawa, Jason Tsai, Kenta Yuan, and Richard Zhang. We are all fifth-year engineering students with at least one year of industrial work experience.

If you have any question or comment about our project, please feel free to contact us at ensc-440-2010sp-fir@sfu.ca.

Sincerely,

John Ogawa
Chief Executive Officer
Freedom Innovation Research
Enclosure: *Design Specification for the Robotic Item Retrieval System*

# Design Specification for the Robotic Item Retrieval System

**Project Team:**
- Steven Choi
- John Ogawa
- Jason Thai
- Kenta Yuan
- Richard Zhang

**Contact Person**
- Steven Choi
- Ensc-440-2010sp-fir@sfu.ca

**Submitted to:**
- Dr. Andrew Rawicz – ENSC440
- Mr. Steve Whitmore – ENSC 305
- School of Engineering Science
  Simon Fraser University

**Issued Date:**
- March 8, 2010

# Executive summary

The design specification for Robotic Item Retrieval System describes the design and development of the entire system in detail. This document presents the design requirements, the implementations and test plan of the robot. The overall system is outlined and test cases are considered.

The robot accomplishes the retrieval task by 4 steps. When the user specifies an item, the robot identifies the current and destination locations. The robot calculates the path to the destination and returns to the user and makes appropriate turns. When the robot arrives to the destination, the platform elevates to the necessary table height and grabs the item tray onto the platform. The robot returns to the user with the item.

The project is divided into mechanical, system, and software components. The mechanical work consists of the platform construction and attaching the wheels, motors, and sensors. The platform is made of aluminum parts and things are held together to the platform by screws. As a system, the sensors, motors, and the controller work with each other. The controller determines the motors' actions accordingly to the sensors' feedback. The software processes the data from the sensors and instructs the robot. Sensor data is displayed to the user on GUI and the user can send instructions from a computer to the robot through wireless network.

To meet the constraints and requirements listed in the functional specification, the detailed selection criteria for the components such as the DC motors, the ultra-sonic sensors, and the servo motors are described. The justification for the chosen microcontroller is also provided. The top-level flow charts for the robot operation and the software program are included for clarification. Finally, the test plan for the subcomponents and the entire system as a whole is provided at the end of the design specification.

At the current design stage, we have realized that there might be room for improvement for some of the system design. We have also become aware that it might be a challenge for us to target every feature listed in the functional specification at the time constraint of the four-month development cycle. Therefore, we will focus on the major features of the Robotic Item Retrieval System and leave the lower priority features as future implementation.

# Table of Content

# List of Figures

# List of Tables

# Acronyms

| | |
|---|---|
| AC | Alternating Current |
| ADC | Analog to Digital Convertor |
| AH | Ampere per Hour |
| CPU | Center Processing Unit |
| FL | Front Left |
| FR | Front Right |
| GPIO | General Purpose Input/Output |
| GUI | Graphical User Interface |
| IC | Integrated Circuit |
| I2C | Inter-IC Bus |
| LDR | Light Dependent Resistor |
| LED | Light-Emitting Diode |
| MB | Megabyte |
| MUX | Multiplexer |
| PC | Personal Computer |
| PWM | Pulse-Width Modulation |
| RL | Rear Left |
| RPM | Round Per Minute |
| RR | Rear Right |
| TCP/IP | Transmission Control Protocol / Internet Protocol |
| USB | Universal Serial Bus |

# Glossary

| | |
|---|---|
| Accelerometer | Device that measures acceleration |
| Best-effort service | Service that does not provide any guarantees that data is delivered |
| Current sensor | Sensor for measuring current through a wire |
| Divx | A video compression format |
| Drive train | The group of components in a motor vehicle that generate power and deliver it to the road surface |
| H-Bridge | Electronic circuit which enables a voltage to be applied across a load in either direction |
| IEEE 802.11 | A wireless communication standard |
| Kernel | The central module of an operating system. The part of the operating system that loads first, and remains in main memory |
| Linux | An Open Source Operating System |
| Photoresistor | A light dependent resistor whose resistance decreases with increasing incident light intensity |
| Ubuntu | A Linux distribution |
| Ultrasonic distance sensor | A device that uses high frequency sound wave for distance measurement |
| VLC | A free media player software for video streaming |
| Wi-Fi | A trademark of the Wi-Fi Alliance often used as a synonym for IEEE 802.11 technology |
| Wlan | Wireless local area network |
| Servo motor | A motor that has error sensing feedback to control steering |
| Solidworks | A CAD program for 3D drawing and simulation |
| Thread | A portion of a program that can run independently of and concurrently with other portions of the program. |

# 1 Introduction

The Robotic Item Retrieval System is a household robot that could self-navigate in a room and retrieve items for the user. The robot is running on a wheeled base and it is equipped with an elevator platform and robotic arms for item retrieval. The robot is equipped with a surveillance camera and ultrasonic distance sensors for navigation and item identification. A computer graphical interface is used to control the robot. The robot will be operating wirelessly through 802.11g/n wireless network to perform bi-directional communication with the host computer GUI. The robot will support two navigation methods: manual remote control and line following. The line following navigation uses line drawn on the floor to guide the robot move through the room [1]. Special markers can be placed on the floor to notify the robot of potential points of interest such as intersection or location of item.

## 1.1 Scope

This document provides the overview of the Robotic Item Retrieval System, describes the complete design requirements of our project and explains how we meet the functional specifications. The test plan is included in the end of the design specification. All the test cases listed in the test plan should be executed carefully after the development phase.

## 1.2 Intended Audience

The design specification is intended for use by all members of Freedom Innovation Research. During the development phase the team will implement the product based on the functional specifications. During testing cycles before the product goes to the market, testers should follow these specifications to help assessing the product and evaluating its features. Lastly, manufacture industry should refer to the specifications for each component in order to meet the overall system requirement.

# 2 System Specifications

The Robotic Item Retrieval System consists of several functional modules: the robot control circuit, the drive train, the robotic arm platform, the sensor module, the wireless module and the computer graphic interface. The figure below detailed the system architecture of our robot.

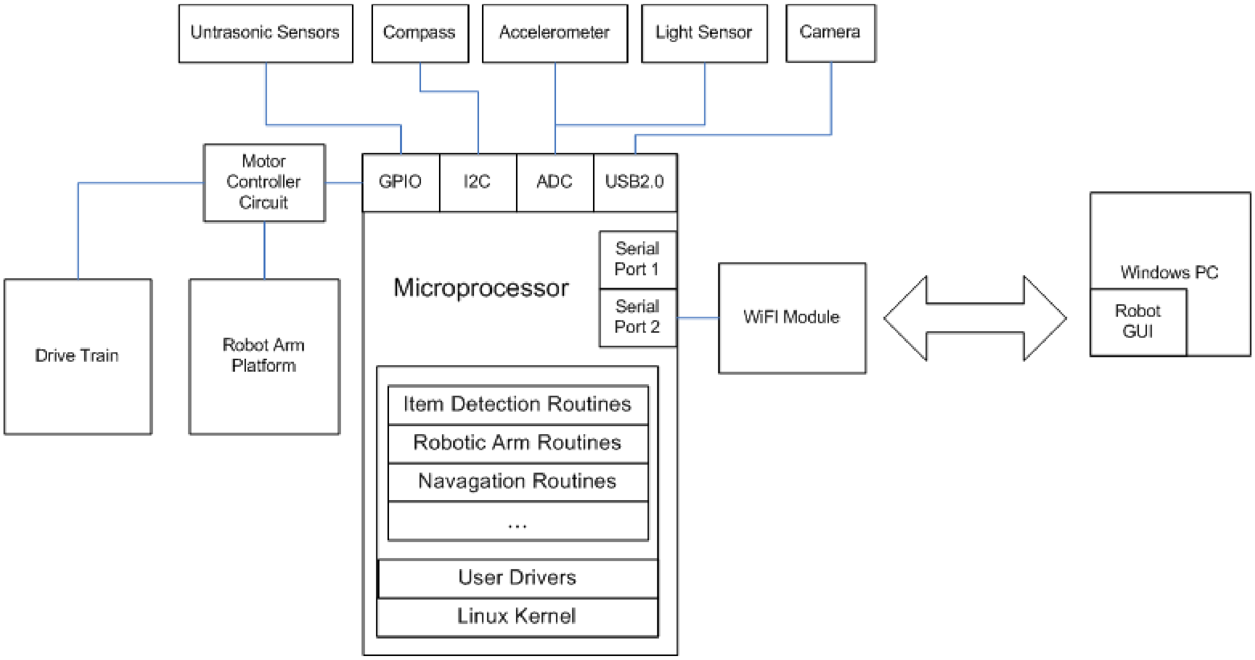


**Figure 1. Architecture of the Robotic Item Retrieval System**

As illustrated in Figure 1, the heart of our project is the robot control circuit built around a DMP Vortex x86DX microprocessor. The Vortex microprocessor provides all necessary interfaces for communicating with the rest of the system. It will be running Linux operating system to host the programs that control the operation of the entire system. The drive train and the robotic arm platform are controlled by the microprocessor through the motor control circuit. The drive train consists of 4 driving wheels, and both the front wheels and rear wheels can be used for steering. The robot arm platform can be raised or lowered per user command. For the functional prototype, we will use a simplified design for the robotic arm which only has the capacity of retrieving items placed in a tray. A more sophisticated robotic arm design will be used for the final product. A combination of a video camera and a variety of sensors are used for robot navigation and item identification. The sensors will be installed all around the robot to provide 360 degrees of surveillance coverage. The robot is controlled wirelessly from a computer GUI that support real time video stream from the robot's camera [2].

# 3 High-level system design

In this section, we will focus on the overall system of the robot which describes how components are connected and positioned.

The figure below shows a block diagram of major input and output which will be processed by the microcontroller (Arduino Mega) and a mobile x86 CPU (Vortex86DX) with different subcomponents.

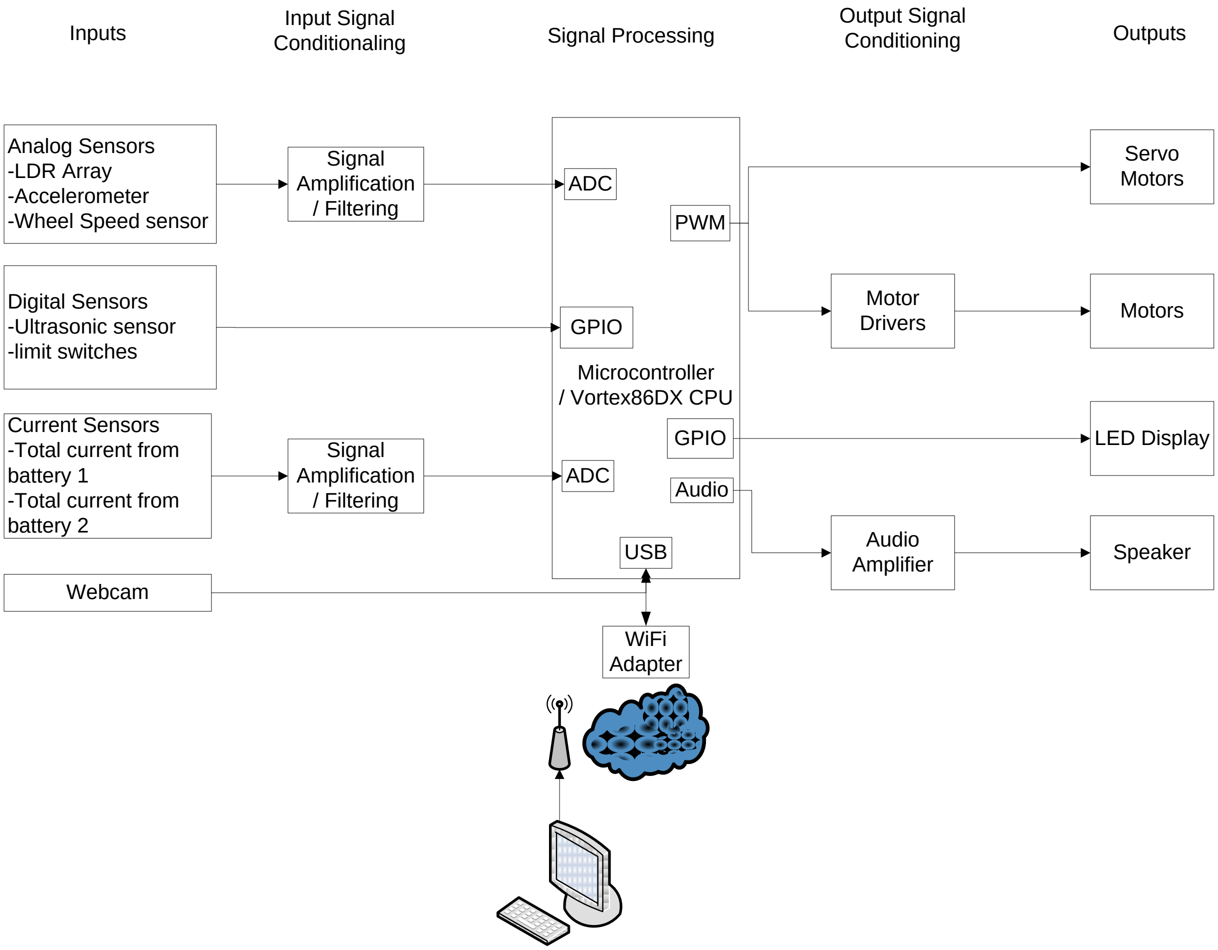


Figure 2. Robot Electrical System Design

## 3.1 Physical and Mechanical Design

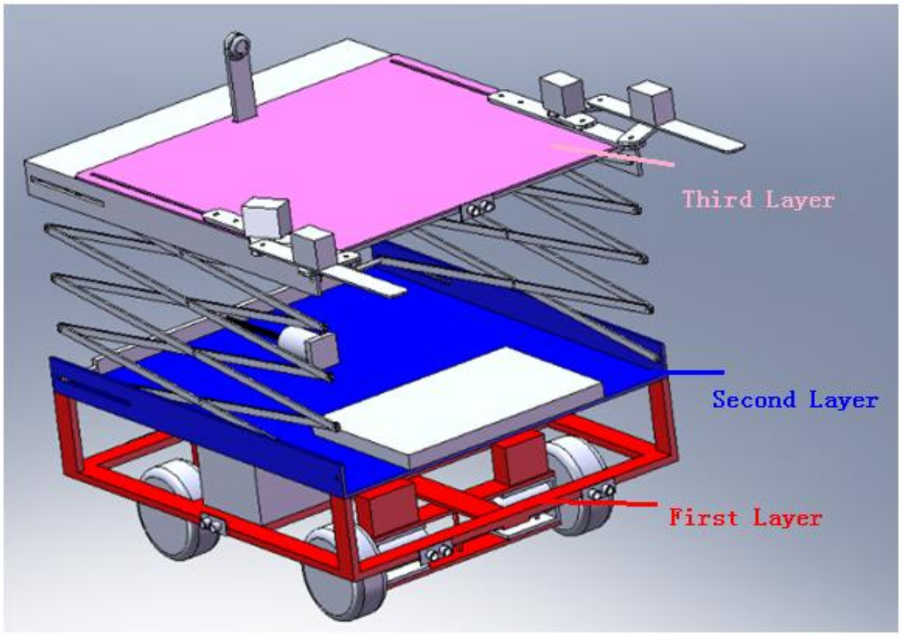


**Figure 3. Robot Physical Structure**

Figure 3 demonstrated the overall physical structure of the robot system. As shown in the figure, the robot's physical structure has three layers. Two batteries are placed in the middle of the first layer. The other two layers are built on top of the first layer. The middle layer is made of plastic board which is supported by four metal bars mounted on each corner of the first layer. Most of the electronic components are mounted on this layer. The third layer consists of elevator platform and robotic arms.

## 3.2 Electrical Design

The following diagram shows a simplified version of how different electrical components connect to each other and their signal direction. Moreover, power lines are shown in bold indicating voltage requirement of different components.

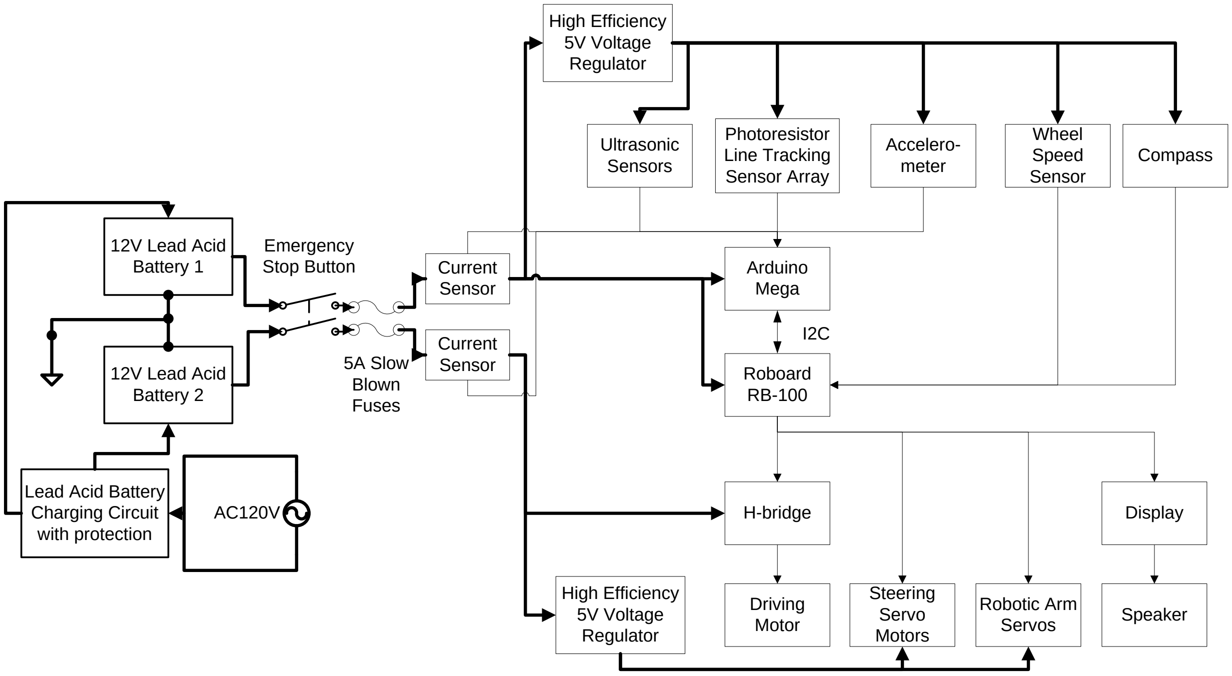


**Figure 4. Electrical Components Layout**

The robot system is powered by two 12V Lead acid batteries. One of them is used to power the onboard processor and sensors while the other one is used to power the motors. We used two separated power sources to protect the sensitive circuits from sudden voltage variations caused by motor load. The home location/base station will provide the charging circuit to both batteries with protection of overcharging.

There is an emergency stop button that could effectively cut off all power to the system in case of emergency. Current sensors will then provide analog signals to the microcontroller so it can detects abnormal current level and cut off the power to different components (not shown in the diagram). By getting both batteries' voltages and currents, we can calculate the instantaneous power consumption and project how long the system can run before next recharge. As our system is running on batteries, a high efficiency 5V voltage regulator is used to ensure a long life operation time.

Roboard RB-100 [3] represents the x86 CPU which is running Linux and the main control programs. All of its GPIO, PWM and ADC ports are used so we decided to add an Arduino Mega board to expand the I/O port numbers and functionality. In addition, there are many open source programs for Arduino Mega that could capture and process sensor signals. Digital and analog signals will be converted into numerical values inside the Arduino so that the CPU can read these values easily and spend more processing power on other tasks.

Detailed circuit designs for sensors, drive train, and elevator platform will be discussed in the corresponding sections.

### 3.2.1 Power supply design

There are many power regulations circuit to output a constant 5V voltage. The most common and easiest to use one is LM7805. However, we are not satisfied with its low efficiency and massive heat generation. After comparing several alternatives from different manufacturers, we choose the switching voltage regulator [4] made by National Semiconductor because it meets all our requirements: 5A max continuous current draw, high efficiency (above 80%), and stable voltage under load. The circuit diagram of the power supply is shown in the following figure.

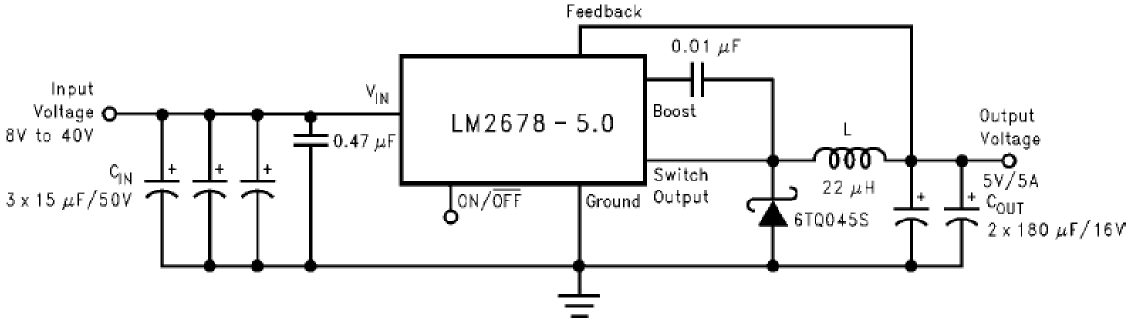


**Figure 5. Circuit Schematic of the Power Supply used for the Robotic Item Retrieval System**

By follow the schematic, we have successfully built the power supply with a measured efficiency of 86%. The following picture demonstrated the actual power supply circuit that will be used on the functional prototype.

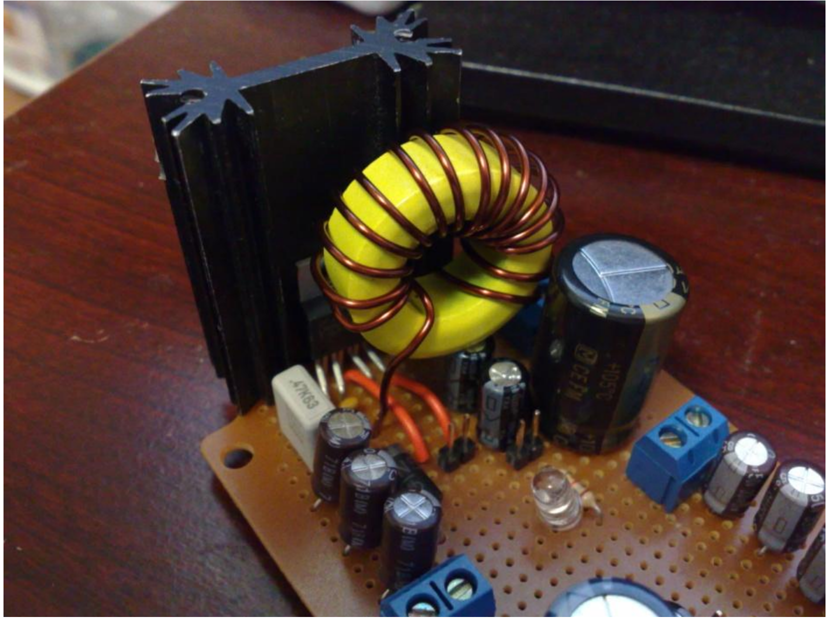


**Figure 6. Picture of Robot Power Supply Circuit**

The robot system is powered by two lead acid batteries. Each single lead acid battery is charged by a complete charger with overcharge protection. By studying the battery statistics, the battery can be divided into three charge stages. It should first being charged by a maximum current of

0.7A (10% of battery capacity, 7AH) until the voltage reaches 14.4V and turns into constant voltage mode. Once the current drops to 0.21A (3% of battery capacity) then it will provides a trickle current and maintain the battery at 100% charge. However, since battery charging is temperature dependent, we plan to spend around $5CAD more to add a temperature compensate and protection circuit to ensure a long battery life and to be green to the earth.

# 4 Sensor Design

## 4.1 Physical and Mechanical Design

Ultrasonic distance sensors will be installed on all sides of the robot to ensure there are no black spot for obstacle detection [5]. Figure 7 illustrates the placement of sensors on the robot base.

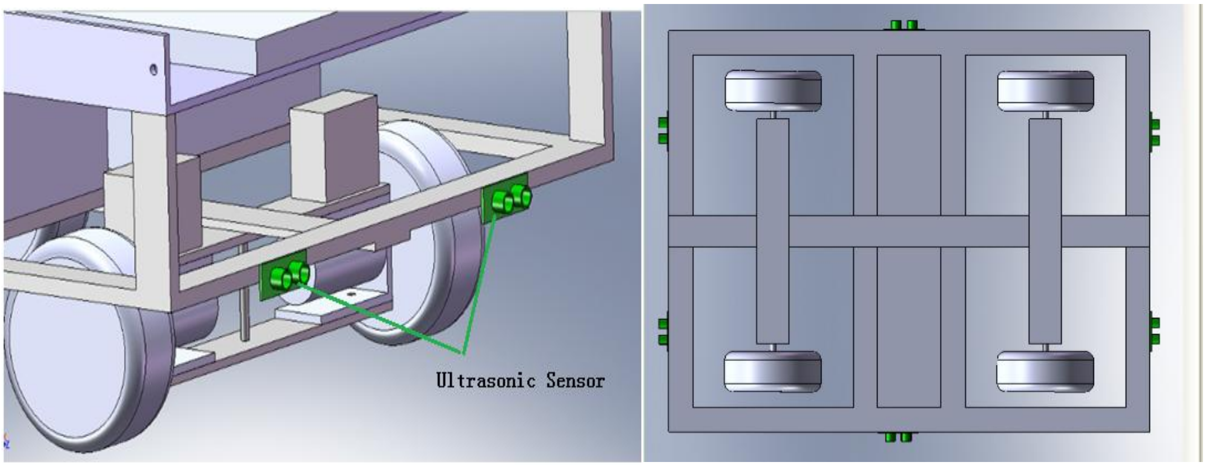


Figure 7. Ultrasonic Sensor Placement

## 4.2 Electrical Design

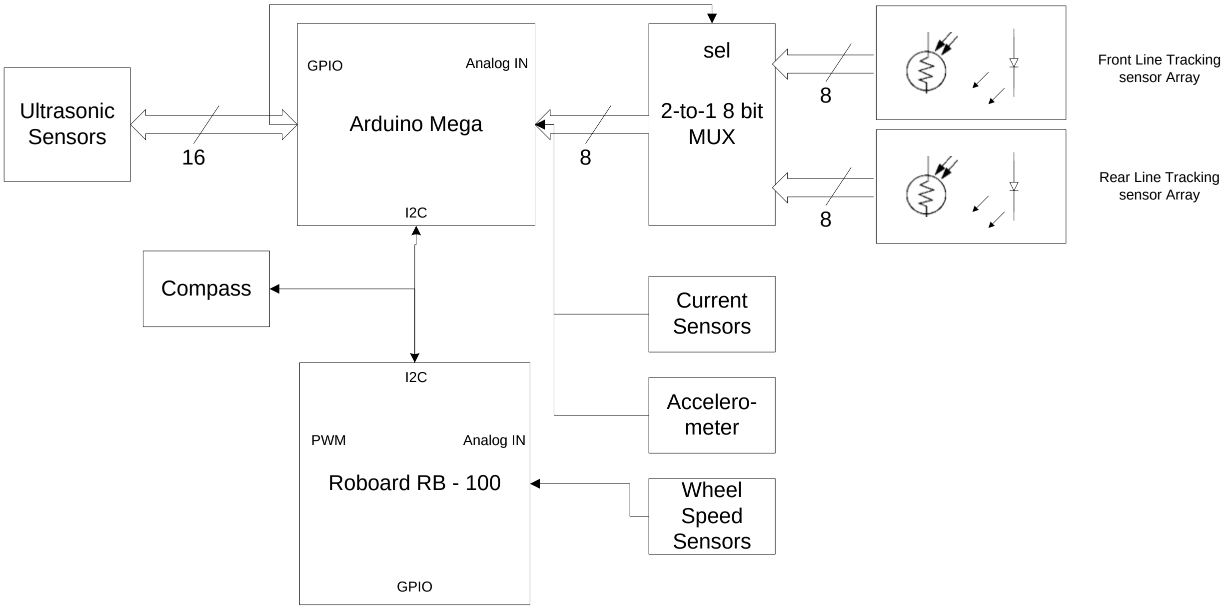


**Figure 8. Electrical Connections of Major Sensor Modules**

As indicated from the diagram above, there are 5 main environment sensors in our robot to provide critical information about location, speed and error detections. Their importance to the robot is as the eyes and ears to humans. All the sensors will be powered by highly regulated 5V voltage so they are free from power supply fluctuation.

Line tracking sensors are used to detect predefined path on the ground. Although there are many line tracking sensors available in the market, their reliable reading range is too short and they are not cost effective considering their low performance. As a result, we decided to build our own transmitter and receiver pair. We have experimented with IR transmitter and receiver pair, white LED and IR receiver pair, and white LED and photoresistor pair. We have found that the white LED and photoresistor pair works the best even when the black line is being used on colored background. Two arrays of LED and photoresistor pairs are used so we can calculate the deviation angle more accurately. In order to save ADC channel, we decided to add an analog MUX. Their sample rate is low enough to let the MUX switch over easily while only half of the ADC channels are used. Furthermore, the imperfection between each photoresistor will be fine tuned in the software.

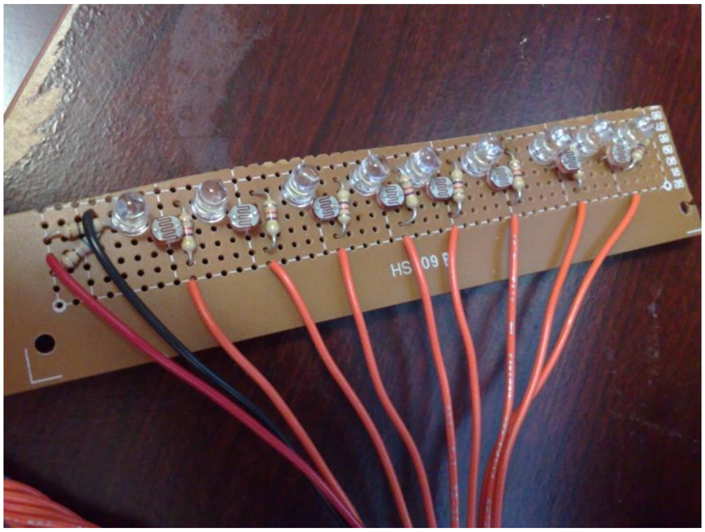


Figure 9. Picture of Line Tracking Sensor using LED and Photoresistor Pair

Ultrasonic distance sensors will help the robot to detect obstacles and their location accurately [6]. We have tested a pair of IR distance sensors produce by Sharp, however, we found that it gives different distance reading between different materials which is undoubtedly unacceptable. For our project, the high accuracy (measured from 5cm – 100cm range, +/- 0.25cm error) of ultrasonic sensors are needed in identifying all surrounding object and determines their distances. There are many selections in the market, pricing from $5CAD to $80CAD. After comparing their ranges, angle coverage, functionalities (e.g. integrated temperature compensation) and connection types (e.g. $I^2C$, TTL, PWM), we decided to use the generic $5CAD ultrasonic sensor as it functionality fully meets our requirement: detection range from 3cm-200cm and 10° degree coverage. The figure below shows the picture of the generic ultrasonic sensor.

Figure 10. Picture of Ultrasonic Distance Sensor

Wheel speed sensors will be mounted on all 4 wheels to monitor the current rotation speed of individual wheels. They are also used to detect wheel slipperage (Wheel has different speed than the others) and stuck wheel (wheel does not move even power is supplied to driving motor). Since we do not have clearance concern in here, small IR transmitters and receivers are used to collect wheel rotational information [7]. Once we know the rotation speeds of the wheels we can calculate the robot speed and distance travelled. The speed sensors mentioned above is show in Figure 11.

**Figure 11. Picture of Wheel Speed Sensor**

A Compass will be installed on the robot because the line tracking sensors and distance sensors do not provide information about the direction of traveling. With the addition of the compass, navigation will become easier and more reliable. We decide to use an I2C interface 2 axis compass with internal calibration to ease our development. "Zhichuan Electronics I2C Magnetic Compass Module" caught our eyes because of its low cost and well-supported functionality [8]. It will be powered by the 5V rail and connected to the CPU I2C channel directly. The picture of the compass module is shown in the following figure.

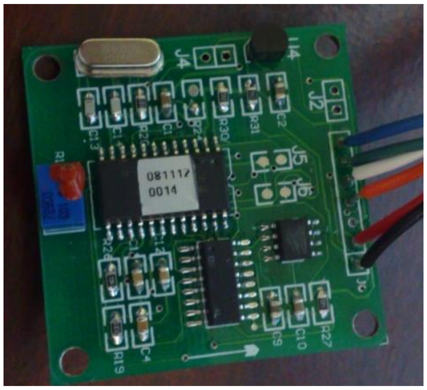


**Figure 12. Picture of Magnetic Compass Module**

An accelerometer is used to detect sudden stop or change of direction of the robot. We decide to use analog accelerometer due to easy connections to the Arduino board and implementation. The next picture shows the accelerometer that we used in the project.

**Figure 13. Picture of Accelerometer Module**

Current sensor is used to monitor the current consumption of the robot. The current sensor is designed base on a simple Hall Effect detection device, which can sense the flux caused by the current flowing and amplify the signal into readable format. The current sensor used is shown in the following picture:

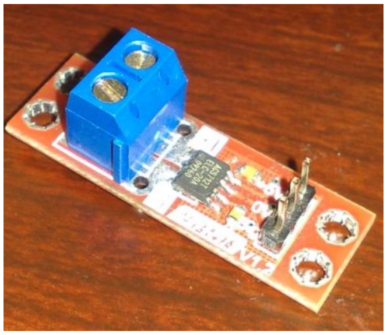


**Figure 14. Picture of Hall Effect Current Sensor**

We decide to use the Logitech C905 Webcam for video streaming because of its reasonable price, good quality, and native Linux support. In addition, the stable grip is convenient to use to attach to the robot [9].

# 5 Drive Train

## 5.1 Physical and Mechanical Design

The drive train, which is the first layer, will be the base of the whole system. It is designed to be as a rectangle at 45cm x 60cm dimension. The drive train will be made of aluminum bar and the structure is shown in the following picture.

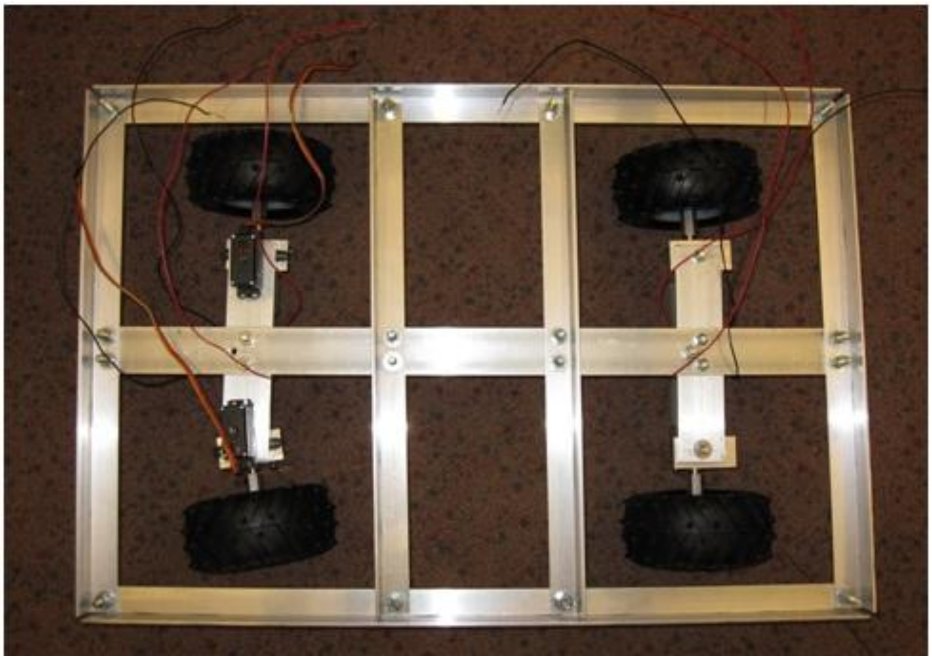


Figure 15. Picture of Drive Train

The four wheels will support all the weight of the system. The front wheels and back wheels will be at least 36cm apart and each pair of the wheels will be separated by at least 25cm. This is to avoid flipping while the robot is moving or carrying items. The gear motors which drive the wheels will be mounted on a small L-shape metal as shown bellow.

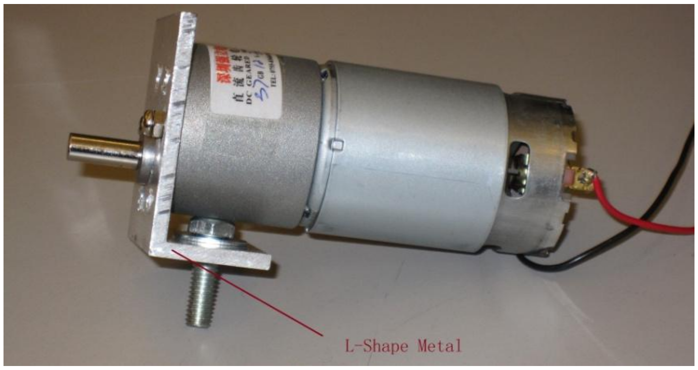


Figure 16. Picture of Gear Motor Mounted on L-shape Metal

There are a couple ways to design the steering, for example, by changing the driving direction of the wheels. However, we found that the best and easiest way to steer is to install servo motor on each driving motor [10]. The servo motor will be installed on the back side of the L-Shape metal. Each of the servos motor will be programmed to have maximum 45 degree of rotation. The following figures demonstrate the above idea.

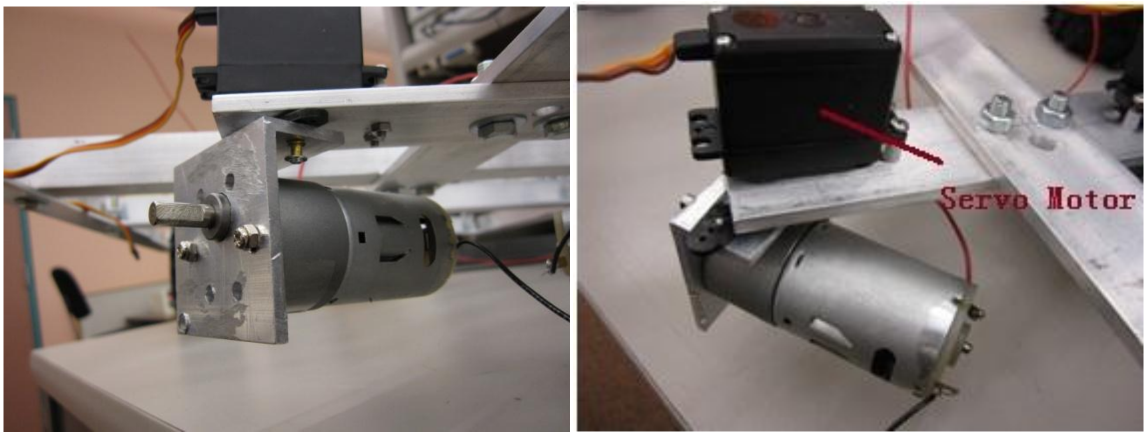


**Figure 17. Pictures of the Steering Mechanism**

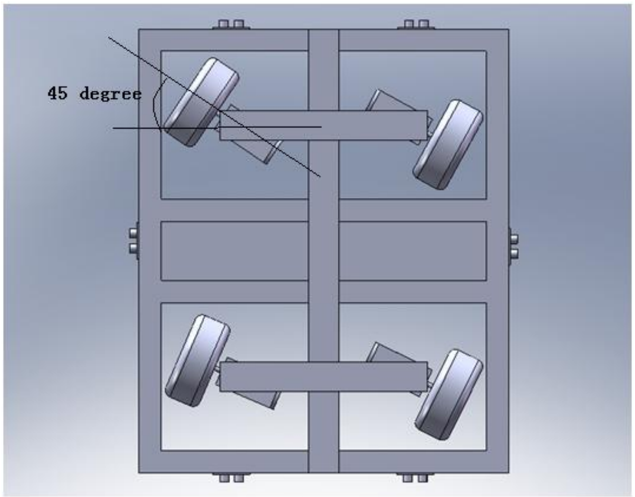


**Figure 18. Steering System Design**

The diameter of each wheel is approximately 12cm. The maximum allowed weight of the robot including the retrieved item will be 20kg, and the friction coefficient between the wheel and carpet is around 1. Hence, the torque that each motor must exceed is 20KG/4*9.8m/s^2*1*(0.12/2 m)=2.94 kgm. We have tested and concluded that the torque generated from the gear motor are able to handle this much of torque.

## 5.2 Electrical Design

All wheels of the robot are driving by gear motors. Due to power restriction, we decide to have the robot running with two low power motors to assist with steering and two high power motor at the rear to do main driving. Then each side is control by an independent H-bridge so a differential speed can be obtained by PWM signal. The wheel speed will be immediately sent back to the CPU by the wheel speed sensors. Typical speed for the driving motors is 15rpm. However, these motors are capable of running at 30rpm when speed is a priority. Thanks to the nature of servo motor, it will map a default set of angle to PWM width – 0° to 180° as 1ms to

2ms PWM signal. No external feedback is needed as the servo motor will steer to desire angle in the fastest time and strongest force set by the command. Since servo motor cannot withstand a voltage above 6V, we decide to supply it with 5V regulated power and the H-bridge/driving motor will be supplied by 12V battery power directly.
High gauge wires are needed for drive motors and H-bridge to handle continuously high current draw.

# 6 Elevator Platform

## 6.1 Physical and Mechanical Design

The elevator platform as illustrated in the following figure will be driven by a screw connected to a gear motor. This mechanism is similar to the car jack mechanism. The purpose of using this design is to guarantee having enough torque from the gear motor to lift up the item. The platform is designed to be able to rise from 40cm to 1m.

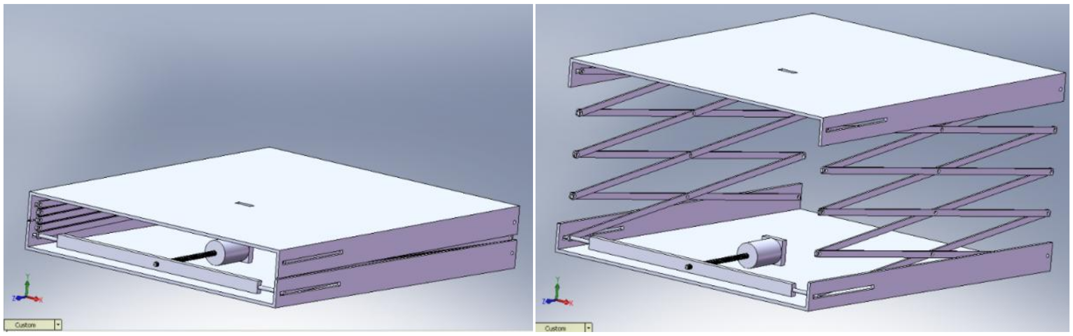


Figure 19. Elevator platform mechanism

Regarding to the material of the robotic arm, it will be made of aluminum as well. Robotic arms are driven by motors to extend forward and backward. The distance from the center of the robot to one of its wheel will be 15cm. The total weight of the robot will be around 20kg. Hence, in order to avoid flipping while the arms are carrying the item (max 1kg), the maximum distance from the edge that the arms can extend should not be larger than 30cm. Once the arms reach the tray, it will be notified by the micro switches which are installed at the end of the arms. The arms will stop and then the platform will rise up 1cm again to raise the tray. Item will be retrieved back afterward. The following picture shows the design idea of the arms.

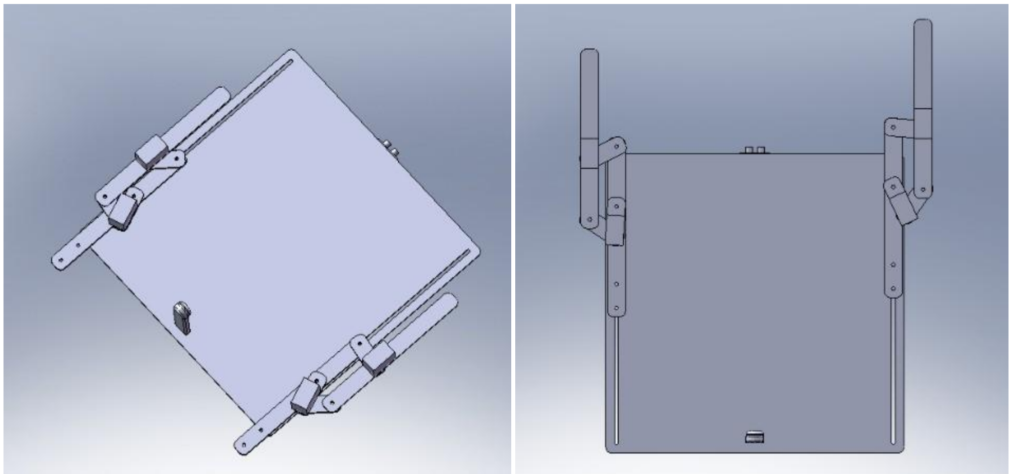


Figure 20. Robot arms mechanism

## 6.2 Electrical Design

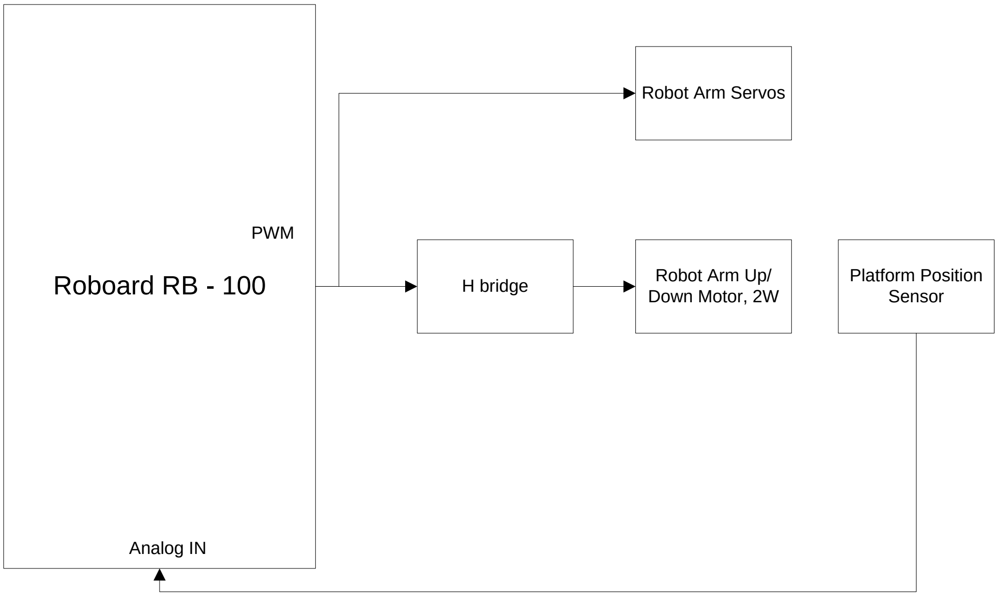


Figure 21. Design of Motor Control Circuit

As shown in the above figure, the electrical design for the elevator platform consists of a feedback control system so the robot can grab the item accurately. Once a command is issued, the system will rise to a predefined height. We will determine the height by two major sensors, the first one is similar to the wheel speed sensor, it senses how many turns the motor has accomplished and return the signal to the CPU so the current height can be calculated. Then low-and-high limit switch will be activated when it has reached the mechanical limit. Once it is reaching the desire height, the speed of rising will be slowed down by adjusting PWM signal to the H-bridge and an ultrasonic sensor will start to look for the edge of the table or flat surface. Then the robot arm servo [11] will operate under direction of CPU to retrieve the item. After that, the platform will lower down to the lowest level.

We will experiment with different servos with different properties in order to find the best combination to work with robotic arm.

Figure 22. Servo Motors used by the Robotic Arm

# 7 Signal Processing and Computation

Table 1 listed the sensors and actuators used in the robot design.

**Table 1. List of Sensors and Actuators**

| **Input device** | **Total number of device** | **Analog/ Digital** | **I/O** | **Total number of bit** | **Sampling rate** | **Priority** | **Description** |
|---|---|---|---|---|---|---|---|
| Ultrasonic sensor | 7 | Digital TTL | I | 14 | 10Hz | Medium | Detect obstacles and distances between wall or obstacles |
| Line Tracking sensor array | 2 | Analog | I | 8 | 10Hz | High | Follow predefined path for robot to follow |
| Accelerometer | 1 | Analog | I | 4 | 1Hz | Low | Detects tilt and unintended acceleration that could cause damage to robot |
| Compass | 1 | Digital I2C | I | 2 | 10Hz | Medium | Read heading current heading direction |
| Current Sensor | 2 | Analog | I | 2 | 10Hz | Medium | Measure power consumption and detect problems |
| Wheel Speed Sensor | 4 | Analog | I | 4 | 100Hz | High | Read current wheel speed for location calculation |
| Elevator platform motor sensor | 1 | Analog | I | 1 | 100Hz, On demand | High | Provide height of the platform |
| Limit switch | 2 | Digital | I | 2 | 100Hz, On demand | High | Stop platform movement when mechanical limit reached |
| Servo Motors | 10 | Digital PWM | O | 10 | NA | High | Control robot movement and steering |
| Driving Motor | 4 | Digital PWM | O | 2 | NA | High | Control speed of robot |
| Elevator platform motor | 1 | Digital PWM | O | 1 | NA | High, On demand | Control height of platform |
| LED display | 5 | Digital | O | 5 | NA | Low | Show current state of operation |

As detailed in the above sections, sensors and actuators are connected to different boards: Arduino Mega and Roboard RB-100. The raw data from each sensor will be processed individually and converted into standard format that are readily to be used by the software. The control of actuators will be done through a set of user drivers which will be discussed in detail in the software design section.

# 8 Wireless Communication Design

In order to control the robot remotely we need to establish wireless communication between the robot and the host computer. We decided to use Wi-Fi technology for wireless communication because it is wildly available and have good range and data throughput. We choose a generic Wi-Fi adapter based on the Ralink 2860 chipset because of its good compatibility and great Linux driver support.

# 9 Navigation System Design

## 9.1 Navigation by Line Tracking

The line following method is used for robot navigation because it is very reliable and easy to setup. The robot is equipped with two arrays of photoresistors that could read the color underneath it. The following diagram shows a conceptual model of the line following system.

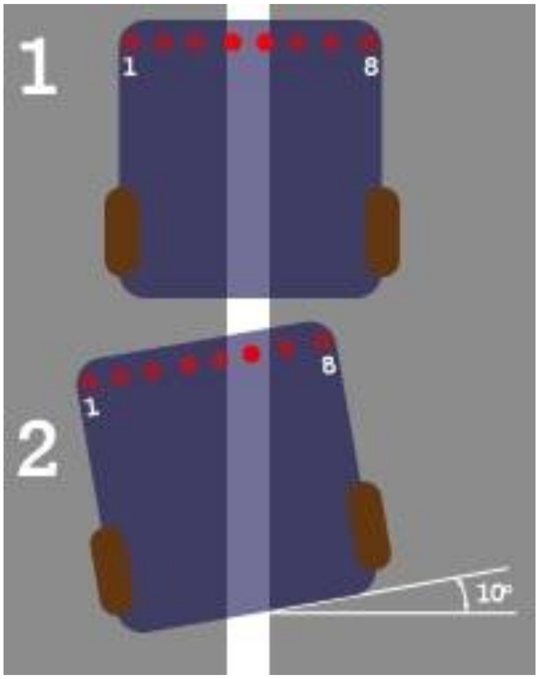


**Figure 23. Conceptual Model of the Line Following System**

When the robot is operating in line following mode, the photoresistor array would detect the distance between the center of the robot and the line [12]. For every 100 milliseconds, these data would be used to adjust the steering angle of the wheels. Proportional control is used to ensure that the robot would track the line both smoothly and effectively. The following table illustrated the relationship between the sensor that detects the line and the resulting steering angle. Note that negative angle means steering to the left and positive angle to the right.

**Table 2. Relationship between Sensor Seeing the Line and Steering Angle**

| Sensor ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Steering Angle | -45° | -30° | -15° | 0° | 0° | +15° | +30° | +45° |

### 9.1.1 Special Markers

Special markers are used to notice the robot of intersections, locations of interest, or end of line. Normally, only one sensor or two adjacent sensors will detect the line. When multiple photoresistors detect the color of the line, an advanced navigation routine will be invoked to handle the special marker.

### 9.1.2 Collision Detection

The robot uses the onboard ultrasonic distance sensors for collision detection. When the robot detects any close obstacles in front of it or along its side, it will send an event to the client and halt at its current action. The operation of the line following algorithm with collision detection is detailed in the following flowchart.

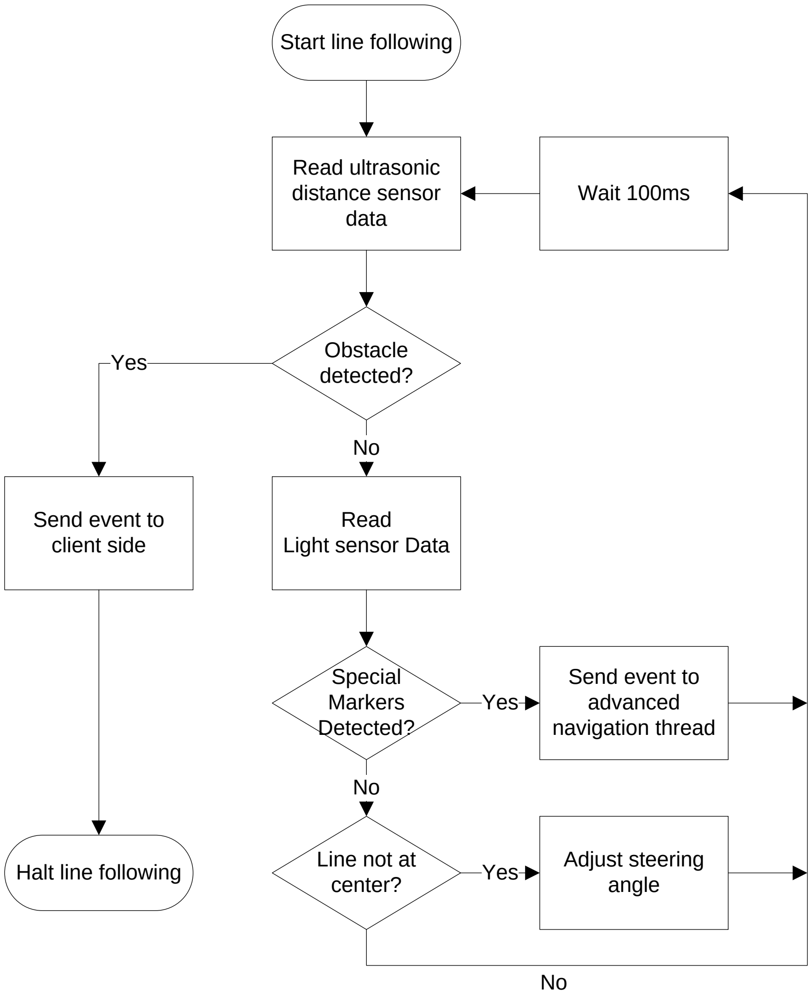


**Figure 24. Flowchart of Line Following Algorithm with Collision Detection**

## 9.2 Navigation by Manual Control

Other than the automatic navigation, the robot can be manually controlled [13] by the user from the client program. The manual control is necessary if the user wants the robot to do specific tasks or if the engineers want to debug problems. The up, down, left, and right buttons on the GUI program will send instructions to drive or steer the robot. Along with the video stream from the robot's camera, the user can control the robot to go off the lines to anywhere and grab anything on a table.

# 10 Robot Software Design

Figure 25 illustrated the software architecture of the robot.

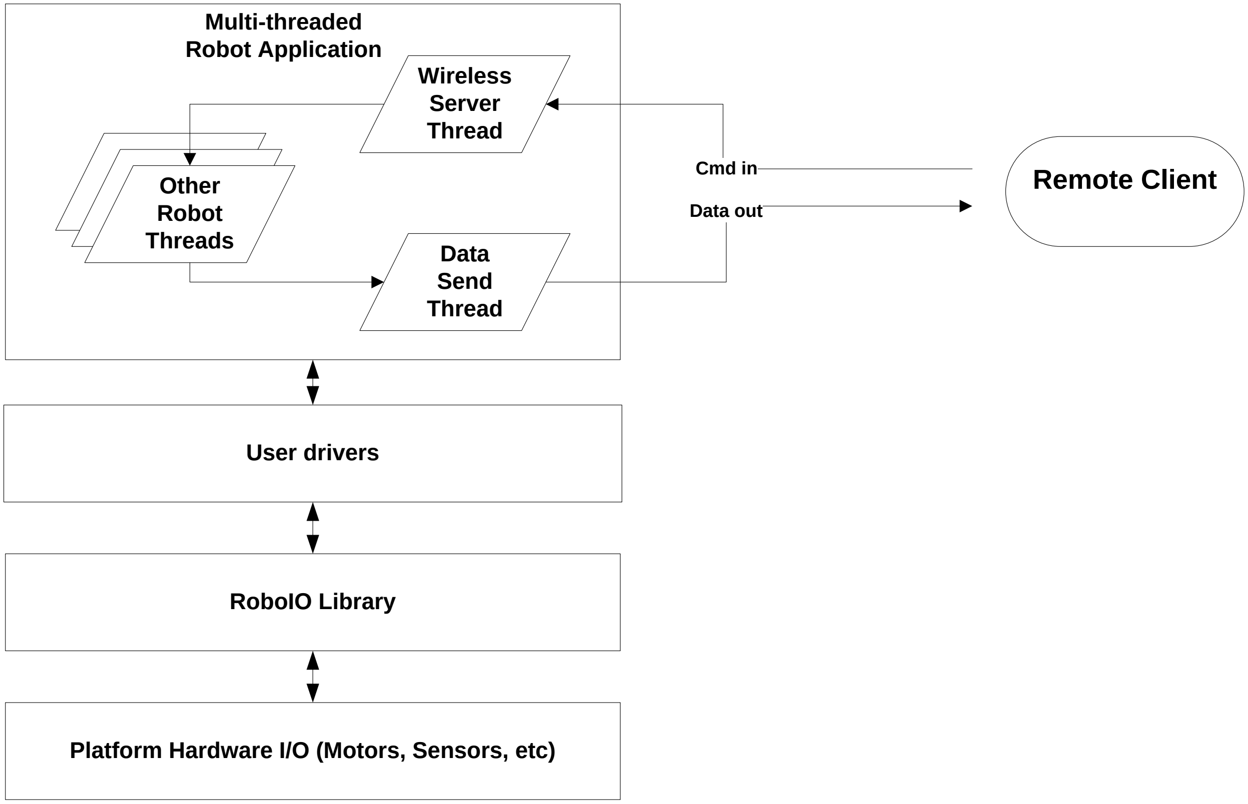


**Figure 25. High Level Software Architecture**

## 10.1 Operation System

A modified Ubuntu Linux is selected as the robot operating system due to its great performance and reliability. To further increase performance, we have recompiled the Linux kernel to include only the minimum required functionally. As the result, the system has fast boot up time and the memory usage of the operating system has been reduced from ~100MB to ~14MB, leaving more memory to our application.

## 10.2 User Drivers

The Roboard development Kit comes with a RoboIO library to control peripherals such as GPIO, PWM and ADC. However, this library is basically functional and not made for easy use. We decide to implement a set of user driver based on the RoboIO library to abstract the low level I/O operations and to provide easy control of the sensors and the actuators. The user drivers should implement the following functionalities:

1) Motor control

  i) Set direction
  ii) Set speed
  iii) Set servo motor angle
  iv) Read servo motor angle

2) Sensor reading.
  i) Read I2C compass
  ii) Read ultrasonic sensor
  iii) Read photoresistor array
  iv) Read motor speed

## 10.3 Multi-Threading Software Design

As we can see from Figure 25, the robot application will have multiple threads running in parallel. The following table contains the description of these threads.

**Table 3. List of Thread for the Robot Software**

| **Thread Name** | **Functionality** |
|---|---|
| Wireless server thread | • Wait for connection from Client (remote computer)<br>• Monitor input from client<br>• Depend on input received, call appropriate function/send message to other module of the software |
| Data collection thread | • Periodically record all sensors reading, motor speed/angle, internal/external control event to a global circular buffer. |
| Data send thread | • Connect to client PC after the wireless server thread established connection.<br>• Periodically send sensor data to client PC. |
| Collision detection thread | • Running in loop. Use data collected by data collection thread to predict potential collisions.<br>• In case of potential collision: generate event to stop robot |
| Line following thread | • Uses photoresistor reading to generate steering servo controls.<br>• Generate event when detect special line marks |
| Advanced navigation thread | • Handling special case in line following<br>• Control direction and speed of movement |
| Manual control thread | • Take manual control command from client, generate appropriate internal control event to adjust moving direction, motor speed, etc. |

| | |
|---|---|
| Robotic arm thread | • Receive events to invoke different retrieving/returning sequence<br>• Use sensor data for object identification, positioning and collision detection. |
| Item retrieval thread | • High level thread handling the process of item retrieving.<br>• Control the robot to move to designated location (line following), grab item (robotic arm control) and then return to the user.<br>• Control the robot to move to designated location (line following), put back the item (robotic arm control) and then return to user. |

## 10.4 Robot Wireless Communication Protocol

For the wireless communication thread, we have two choices of the communication mode: the connection-Oriented mode and the connection-less mode. The advantage of the connectionless communication is that there is no setup overhead and delay . However, the connectionless mode is a best-effort service meaning that the packets arrived at the client are possible out-of-order or loss. For our project, we would like to deliver a reliable product that is very response to the user. Therefore, we have choosen to deploy the connection-oriented communication design [14]to the wireless communication between the Robotic Item Retrivel System and the user.

## 10.5 Video Streaming

The video streaming feature is implemented using the VLC media player. VLC is open source cross-platform media player software that could be used as both streaming server and streaming client. On the robot, the VLC player is configured as a Linux service running in the background. It takes raw video input from the camera, encode it into Divx format and stream it to the client PC. On the client side, A windows version of VLC player is used to decoding and display the video stream. The detail will be discussed in the client software section.

# 11 Client Side Software Design

The client side software is a GUI program that controls the operation of the robot through Wi-Fi wireless network. The client connects to the robot using TCP/IP protocol. On the computer, the user simply presses buttons to send data to the server to control the robot. The client waits for server's data in receiving thread. As soon as we implement the server program to retrieve data from sensors to the robot, the client program will display the sensors' readings.

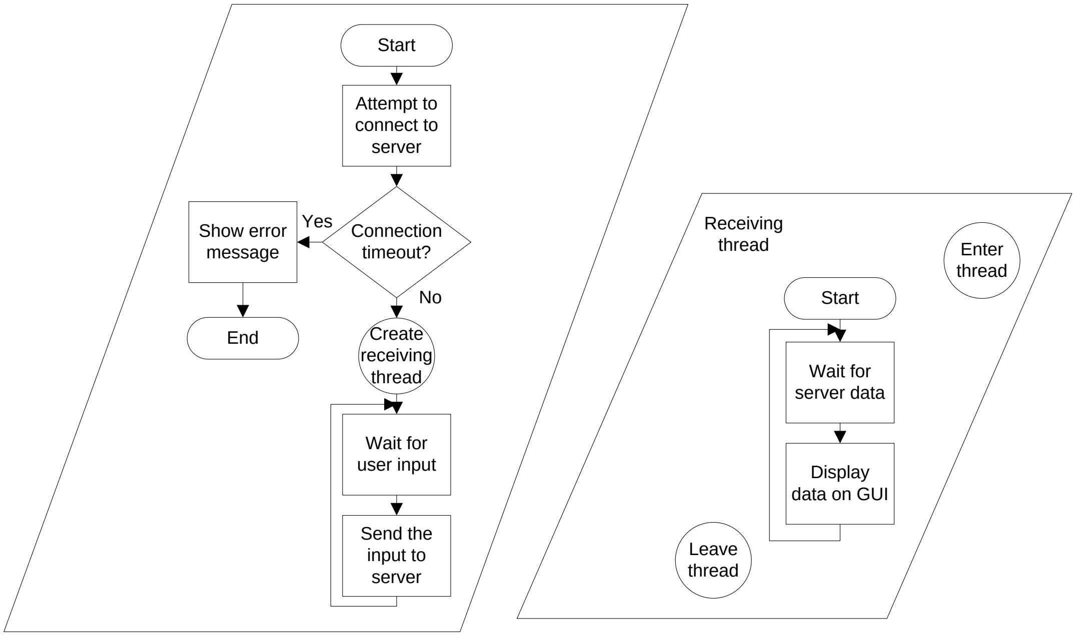


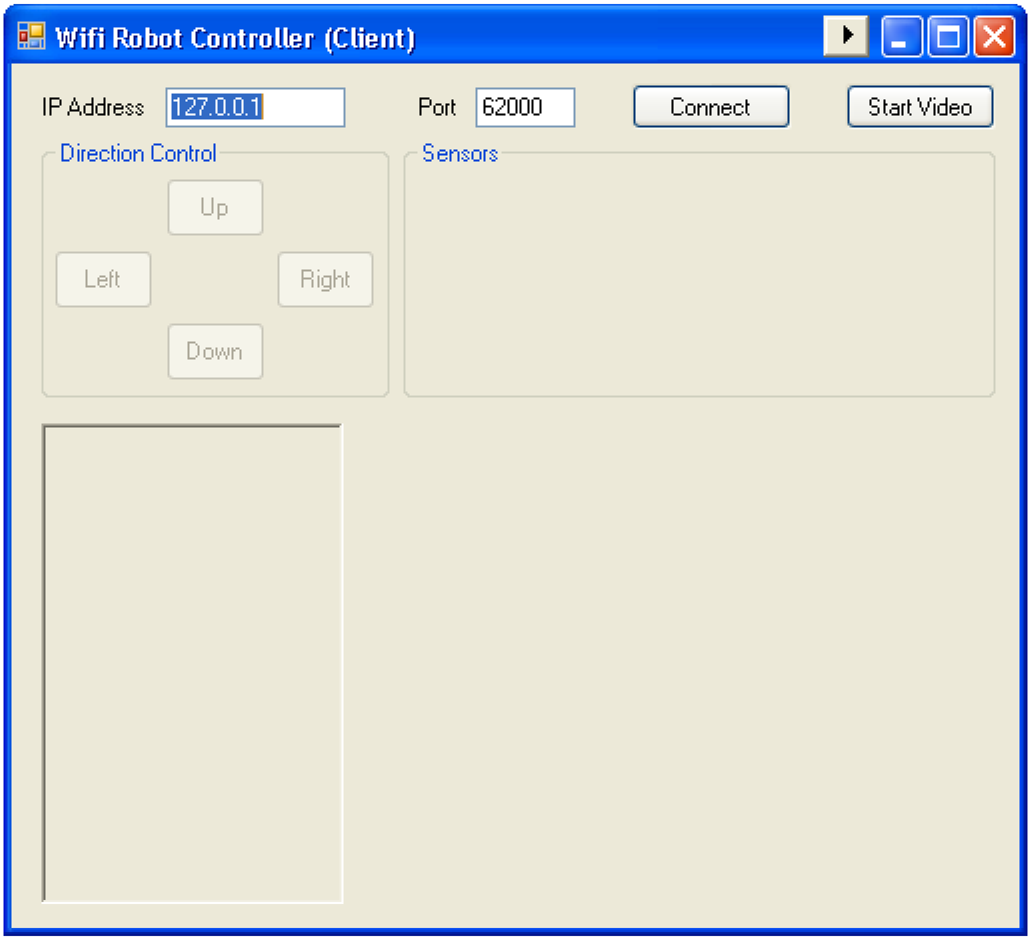
**Figure 26. Simplified Flowchart of the Client Program.**

The GUI program runs in the format of pressing a button to call functions by interrupt [15]. Thus describing all the functionalities with a flowchart requires too much detail and space. Figure 26 shows the logic flow only when the connect button is pressed. The basic idea of the logic flow can be summarized in 3 steps:

1. Make connection to the server
2. Create a new thread that receives and processes data from the server
3. Get user input and send it to the server

The connection can be lost, for example due to wireless network problem, then such corner case is handled as follows. If the connection is lost any where in the logic flow, both threads will terminate and printout the error message. In this way, the user and the robot will know the loss of connection. The robot will automatically try to re-establish the connection [16] and the user can press connect button again.

The following screenshot shows the cilent side GUI program under development. The “Start Video” button will start the VLC player on the client PC with the parameters for connecting to the VLC streaming server on the robot. The VLC player will decode and display the video stream on a seperated window beside the GUI.



IP Address 127.0.0.1

Port 62000

Figure 27. Screenshot of the GUI Program (Under Development)

# 12 Test plan

Three types of testing will be done for the robot to ensure proper function of the system. First, the individual components that made up the sub-system will be tested before they are installed. Then the sub-system testing will be conducted after each sub-system is built to verify its functionality. Finally, at the integration stage of we will put the complete system under a series of scenario test. The robot is expected to handle some common corner cases so they will be tested as well.

## 12.1 Unit Testing

Servo motor:

1. Verify the angle turned match the PWM signal send by the user
2. Verify the torque generated by the motor is as specified in the datasheet
3. Verify the current consumption is within the limit of the controller circuit

DC motor:

1. Check the speed of the motor under load
2. Verify the torque generated by the motor is as specified in the datasheet
3. Verify the current consumption is within the limit of the controller circuit

Ultrasonic distance sensor:

1. Check the maximum and minimum working range
2. Check the maximum operating angle
3. Check the sensor response time

Photoresistor:

1. Verify the photoresisor changes resistance with respect to the color it sensed.
2. Verify all photoresistors have similar threshold for color detection

Accelerometer:

1. Verify that the output of accelerometer is accurate to with 0.1g

Voltage regulator:

1. Check the efficiency of the voltage regulator
2. Check the temperature under load

## 12.2 Sub-system Testing

Server and client connection:

1. Verify the client could connect to the server
2. Disconnect from the client side, and verify the client could reconnect to the server
3. After connection established, move the wireless module far away so that the connection is lost. Verify that the client could try to re-establish the connection.

Elevating platform and robotic arm module:

1. Verify the elevating platform could lift a 1kg load
2. Verify the elevating platform could rise to the height of 1 meter while still maintaining stability.
3. Verify the robot arms could extend forward
4. Verify the robot arms has enough grip to drag a 1kg item back to the platform

Navigation module:

1. Verify the photoresistor could differentiate the black and white color and send different signal to the control.
2. Verify the line following module could correctly process the photoresistors' data and instruct the wheels to make proper amount of turning angle.
3. Verify the ultrasonic sensors could prevent from collision by sensing the approaching obstacle and send a signal to the controller to stop the robot.
4. Verify the compass reads the correct orientation of the robot and send the data to the controller.

## 12.3 System Testing

We will integrate the functional prototype after verifying the functionality of each sub-system. We will verify the system to ensure the functions listed in the functional specification are met. The robot should perform the following series of steps when the user specifies an item to retrieve.

1. The robot interpret the current and destination location
2. The robot figures out the correct path to take
3. The robot follows the track to the destination
4. The robot stops at the destination and elevate the platform to the height of the item
5. The robot extend the robot arm,
6. The robot arm grip the tray where the item is at
7. The robot retrieve the tray
8. The robot follows the track to the user's location

## 12.4 Corner Case Testing

Table 4 listed the corner cases that the robot should be able to handle. They will be tested after we build the prototype.

**Table 4. List of Corner Cases and Expected Actions**

| **Corner Case** | **Expected Solution** |
|---|---|
| Robot runs out of the track | Reverse back to the last known position on |

| | |
|---|---|
| | the track |
| Robot is lifted by external force | Stop all current actions and alert the user by beeping and send notification to the GUI |
| Robot is low in power | Robot automatically returns to the charging station |
| Robot is unable to drag a item that exceeds the load limit | Robot gives up the operation after several attempts and notify the user on the GUI |
| Robot is blocked by a obstacle | Robot stop all action and notify the user on the GUI |

# 13 Conclusion

In this document, we have discussed the necessary design choices and methods adapted to meet the requirements listed in the functional specification. We have also mentioned the possible future design solutions for the product. The design specification provides a clear guideline for the development for our Robot Item Retrieval System. The test plan included in this document will be executed fully to ensure all the functional requirements are presented in the final prototype.

# 14 Reference

[1] Dale A. Heatherington, “Line Follower Robot,” Oct. 1999. [Online]. Available:http://www.wa4dsy.net/robot/linefollower. [Accessed: Jan. 19, 2010]

[2] Wei-Meng Lee, “Monitor Your Web Cam from a Remote Computer,” Jan. 31, 2007. [Online]. Available: http://www.devx.com/dotnet/Article/33637. [Accessed: Feb. 2, 2010]

[3] DMP Electronics Inc., “Roboard Starter Kit,” RB-100 datasheet, Apr. 2009.

[4] National Semiconductor, “SIMPLE SWITCHER: High Efficiency 5 A Step-Down Voltage Regulator,” LM2678 datasheet, Mar. 2000.

[5] U. Rembold, T. Lueth, and T. Ogasawara, “From Autonomous Assembly Robots to Service Robots for Factories,” Sept. 16, 1994. [Online]. Available: http://ieeexplore.ieee.org/search/freesrchabstract.jsp?arnumber=407567&isnumber=9158&punumber=3221&k2dockey=407567@ieeecnfs&query=(robot)<in>metadata&pos=1. [Accessed: Jan. 26, 2010]

[6] Libo Yang and Steven M. LaValle, “The Sampling-Based Neighborhood Graph: An Approach to Computing and Executing Feedback Motion Strategies,” Jun. 2004. [Online] Available: http://ieeexplore.ieee.org/search/freesearchresult.jsp?history=yes&queryText=(sampling-based+neighborhood+graph). [Accessed: Jan. 23, 2010]

[7] Pololu Corporation, “Pololu QTR Reflectance Sensor Application Note,” 2009. [Online]. Available: http://www.robotshop.com/content/PDF/pololu-qtr-reflectance-sensor-application-note.pdf . [Accessed: Mar. 2, 2010]

[8] Shanghai ZhiChuan Electronic Tech Co., Ltd., “$I^2C$ Electronic Compass,” ZCC210N datasheet.

[9] Logitech, “Webcam C905,” 2009. [Online]. Available: http://www.logitech.com/index.cfm/webcam_communications/webcams/devices/5868&cl=ca,en . [Accessed: Mar. 4, 2010]

[10] Seattle Robotics Society, “Standard Technologies of the Seattle Robotics Society,” Nov. 10, 2004. [Online]. Available: http://www.seattlerobotics.org/guide/. [Accessed: Jan. 14, 2010]

[11] Hitec RCD USA, “Standard Deluxe Servo,” HS-422 datasheet.

[12] Ibrahim Kamal, “Line Tracking Sensors and Algorithms,” Apr. 15, 2008. [Online]. Available: http://www.ikalogic.com/tut_line_sens_algo.php. [Accessed: Jan. 19, 2010]

[13] K. Hagan, M. Hillman, and S. Hagan, “The Design of a Wheelchair Mounted Robot,” Mar. 7,

1997. [Online]. Available: http://ieeexplore.ieee.org/search/freesrchabstract.jsp?arnumber=641219&isnumber=13917&punumber=5100&k2dockey=641219@ieeecnfs&query=(robot)<in>metadata&pos=2. [Accessed: Jan. 26, 2010]

[14] A. Leon-Garcia and I. Widjaja, Communication Networks, 2/e, McGraw-Hill Higher Education, 2004.

[15] Chen Yimin, Zhang Tao, and Wang Di, "A Robot Simulation, Monitoring and Control System Based on Network and Java3D," Jun. 2002. [Online]. Available: http://ieeexplore.ieee.org/search/freesrchabstract.jsp?arnumber=1022085&isnumber=21984&punumber=7949&k2dockey=1022085@ieeecnfs&query=(robot)<in>metadata&pos=0. [Accessed: Jan. 26, 2010]

[16] Joo-Hyung Kim, Jeong-Eom Lee, and Hyun-Gu Lee, "Identification and Control for an Unkown Robot in Intelligent Space," Jun. 2004. [Online]. Available: http://ieeexplore.ieee.org/search/freesrchabstract.jsp?arnumber=5326287&isnumber=5326035&punumber=5306517&k2dockey=5326287@ieeecnfs&query=(robot)<in>metadata&pos=19. [Accessed: Jan. 24, 2010]

[17] Energy Star, "Product Specifications: Program Requirements," [Online]. Available: http://www.energystar.gov/index.cfm?c=product_specs.pt_product_specs. [Accessed: Feb. 5, 2010]

[18] Federal Communications Commission, "Radio Frequency Safety," Jul. 1, 2009. [Online]. Available: http://www.fcc.gov/oet/rfsafety/. [Accessed: Jan. 30, 2010]

[19] Jonathan Bennett, "Wifi Robot," Aug. 2008. [Online]. Available: http://www.jbprojects.net/projects/wifirobot/#hardware. [Accessed: Jan. 19, 2010]