**Final Project**

**Voice Controllable Mobile Arm Platform**

**Design Review Deliverables**

**Lab Team: Tianshu Bao**

**Shawn Stern**

**Dan Tran**

**Lab Section: EE 478**

**Date: May 31, 2013**

TABLE OF CONTENTS

[1 TABLE OF FIGURES 2](#_Toc357774253)

[2 REQUIREMENT SPECIFICATION 4](#_Toc357774254)

[2.1 System Description 4](#_Toc357774255)

[2.2 Specification of External Environment 4](#_Toc357774256)

[2.3 System Input and Output Specification 4](#_Toc357774257)

[2.3.1 System Inputs 4](#_Toc357774258)

[2.3.2 System Outputs 5](#_Toc357774259)

[2.4 User Interface 5](#_Toc357774260)

[2.5 Use Cases 6](#_Toc357774261)

[2.6 System Functional Specification 8](#_Toc357774262)

[2.7 Operating Specifications 8](#_Toc357774263)

[2.8 Reliability and Safety Specification 8](#_Toc357774264)

[3 DESIGN SPECIFICATION 10](#_Toc357774265)

[3.1 System Description 10](#_Toc357774266)

[3.2 Specification of External Environment 10](#_Toc357774267)

[3.3 System Input and Output Specification 10](#_Toc357774268)

[3.3.1 System Inputs 10](#_Toc357774269)

[3.3.2 System Outputs 11](#_Toc357774270)

[3.4 User Interface 11](#_Toc357774271)

[3.5 Use Cases 13](#_Toc357774272)

[3.6 System Functional Specification 14](#_Toc357774273)

[3.7 Operating Specifications 16](#_Toc357774274)

[3.8 Reliability and Safety Specification 17](#_Toc357774275)

[4 TESTING 18](#_Toc357774276)

[4.1 Test Plan 18](#_Toc357774277)

[5 FAILURE MODES ANALYSIS 19](#_Toc357774278)

[6 BILL OF MATERIALS 20](#_Toc357774279)

[7 APPENDIX 21](#_Toc357774280)

[8 SCHEDULE 26](#_Toc357774281)

# TABLE OF FIGURES

[Figure 1 - User app splash screen 6](#_Toc357774282)

[Figure 2 - User app interface 6](#_Toc357774283)

[Figure 3 - Use Case Diagram 7](#_Toc357774284)

[Figure 4 - User app splash screen 12](#_Toc357774285)

[Figure 5 - User app interface 12](#_Toc357774286)

[Figure 6 - Use Case Diagram 13](#_Toc357774287)

[Figure 7 – System/Hardware Block Diagram 15](#_Toc357774288)

[Figure 8 - Activity Diagram for Distance Measurement 16](#_Toc357774289)

[Figure 9 - Software Block Diagram 21](#_Toc357774290)

[Figure 10 - Functional Decomposition 22](#_Toc357774291)

[Figure 11 - Control and Data Flow Diagram 23](#_Toc357774292)

[Figure 12 - System Activity Diagram 24](#_Toc357774293)

[Figure 13 - System sequence diagram 25](#_Toc357774294)

# REQUIREMENT SPECIFICATION

## System Description

This specification describes a user-controlled, multi-wheeled robot with a rotating arm for grabbing nearby objects. The arm will have 135 degrees of freedom horizontally, and up to 90 degrees of freedom vertically. The robot is to be able to move at reasonable speeds of up to 0.5m/s through its environment, with the ability to quickly turn in any direction, and to move forward and backwards in addition to braking. The robot is to be able to detect immediate hazards and obstacles in its forward path up to 3 meters away and halt, and provide feedback to the user informing them of the event. The robot is to be able to grasp small, light objects on the ground or in the robot’s immediate vicinity for relocation. All robot functions are to be controllable using an Android phone application with controls for the motion of the robot, in addition to controls for the arm mechanism itself. The robot is to also be able to accept voice commands as a substitute for direct, physical interaction with the android interface. In addition, the robot is to support autonomous navigation from its starting location to a point determined by the user. The robot is to be low cost, flexible, easily reproducible, and intuitive to use.

## Specification of External Environment

The system is to operate in an environment such as a home, ideally with minimal clutter on the floor surfaces on which it will be operating. Room temperature and typical in-house lighting are the expected environment.

The unit will run off of a 9V battery pack attached to the robot chassis for easy swapping in the event of a dead battery. Uptime will be up to 5 hours.

## System Input and Output Specification

### System Inputs

The system will accept the following external inputs:

* Analog pulse from the distance sensor, 5V with a pulse width of 0uS to 38ms

The system will accept the following commands by means of button presses on the Android application screen:

* Move forward (move continuously while button is pressed)
* Move backward (move continuously while button is pressed)
* Turn right (move continuously while button is pressed)
* Turn left (move continuously while button is pressed)
* Brake (Halts ALL robot movement)
* Rotate arm left (move continuously while button is pressed)
* Rotate arm right (move continuously while button is pressed)
* Rotate arm up (move continuously while button is pressed)
* Rotate arm down (move continuously while button is pressed)
* Grasp/Release (toggle button)
* Receive Voice Command

The following voice commands will be usable after pressing the “Receive Voice Command” button on the application:

* Go forward (robot moves until told otherwise)
* Go back (robot moves until told otherwise)
* Turn left (robot moves until told otherwise)
* Turn right (robot moves until told otherwise)
* Brake (robot halts ALL motion)
* Arm left ‘x’ degrees
* Arm right ‘x’ degrees
* Arm up ‘x’ degrees
* Arm down ‘x’ degrees
* Grasp
* Release

### System Outputs

The system will report the following information on the user interface via serial interface over Bluetooth connection:

* Current direction of travel (forward, back, turning, braked)
* Arm status (rising, lowering, grasped, open)
* Distance to nearest large object in front of the robot in cm
  + Range from 0 to 4000cm
* PWM signals to the following motors
  + Drive motors: 0-5V DC

## User Interface

The user will be able to continuously control the robot and query its sensors for data using a simple android touch interface or vocal commands. The following buttons are available:

Robot Direction:

* Directional arrows. Robot moves (relative to its current orientation) in accordance with the held arrow until it is released.
* Brake. Robot halts all motion.

Arm Control:

* Directional arrows to control the orientation and elevation of the grasper at the end of the arm. Arm moves as long as the appropriate button is depressed or until the arm reaches the limits of its rotation. Robot will only be able to move one discrete direction at a time.
* Grasp/Release toggle. Pressing the button causes the grasper to switch to the opposite state (from grasping to releasing the object, and vice versa)

Activate Voice Control:

* Press button to launch voice prompt to interpret voice commands.

Measurements and robot status information will be displayed in the top left corner of the android interface as text through a serial interface over Bluetooth.

The android interface will appear as follows: upon loading the application, the user will be presented with a splash screen while the Bluetooth pairing and initialization of the app occur. The application interface itself will have an upper set of buttons for directional control with a central button for sending voice commands. The lower set of buttons will operate the arm assembly, with the center button serving as the grasp/release toggle. The red button emblazoned with a brake pedal, positioned between the sets of controls and off to the right for ergonomic reasons, serves as the ‘Brake’ command that will halt all robot movement until a new input is given to the system.

|  |  |
| --- | --- |
| E:\Dropbox\EE\EE 478\478 Spring 2013\Final Project\Deliverables\Splash Screen.png  Figure - User app splash screen | E:\Dropbox\EE\EE 478\478 Spring 2013\Final Project\Deliverables\Interface_edited.png  Figure - User app interface |

The field where “Voice command” is listed in the location where textual status information will be displayed.

## Use Cases

The cases are given in the below diagram:



Figure - Use Case Diagram

***Control Robot***

The robot chassis will move in accordance with user input. If the Forward button is held down, the robot will continue to move forward in a straight line until the button is released or if an object is detected as too close (robot will automatically brake and the only available movements will be rotation or moving backwards). This system is controlled by the upper set of arrows in the top half of the android interface. In addition, speed control is an option to adjust how quickly the robot chassis may move.

Exceptions: If terrain is too cluttered, the robot may not be able to maintain true, straight motion as its path will be disrupted by debris. If Bluetooth connection is lost, error will occur.

***Control Arm***

The arm assembly will move in accordance with user input. If the user holds the Down button, the grasping end of the arm will lower itself closer to the ground until either the user releases the button, or if the grasping end reaches the limit of its lowering range. The grasping end of the arm can be commanded to either close its grip or release it. This system is controlled by the lower set of arrows and central button in the bottom half of the android interface.

Exceptions: If the object is too heavy or too large, the arm assembly is not guaranteed to be able to properly grasp the object, or to be able to lift it without potentially destabilizing the robot. If Bluetooth connection is lost, error will occur.

***Voice Control***

The user interface will support the conversion from spoken words in English to their respective commands as needed to control the robot and/or arm assembly as appropriate. For example, the user may instruct the arm to “Arm up 20 degrees” at which point the robot arm will rise such that the difference from its previous position and new position is 20 degrees, within tolerances. The mode may be activated by pressing the speaker icon in the center of the user display.

Exceptions: If user does not speak clearly with proper enunciation, their statement may be misinterpreted. If Bluetooth connection is lost, error will occur.

***Measure Distance***

The robot will return the distance from the front section of the robot to the nearest object in front of it that is at least .5sq ft in dimension. The button for this function is still under design consideration.

Exceptions: If an object is smaller than .5sq ft. there may be an error in detecting the object correctly. If Bluetooth connection is lost, error will occur.

***Autonomous Mode***

The robot will accept a given target distance to reach based on its current direction. The robot will then autonomously navigate to that location, keeping track of its position in the environment and navigating around nearby obstacles.

Exceptions: If the path is blocked with debris that blocks the path of the sensor, error may occur in position tracking. System is heavily dependent on the user to specify the correct destination as the robot lacks means of identifying location other than distance and orientation.

## System Functional Specification

The system is intended to operate and move around in the environment based on user input, with the input from the distance sensor as a safeguard from collisions. A measurement system will be implemented to convert the output pulse of the distance sensor into a user-accessible distance value, selectable as either inches or cm. This distance value will also be used directly by the robot for safety reasons to halt robot movement if it is in danger of a head-on collision.

The user will be able to select either continuous motion of the robot (moves until stopped) or motion that only lasts as long as the respective button is being held down.

## Operating Specifications

The system shall operate in a typical household environment:

Temperature Range 40 – 120F

Humidity up to 90%

Power 9V battery

The system shall operate for a minimum of 5 hours on a fully charged 9V battery

## Reliability and Safety Specification

MTBF: Minimum 2,500 hours

The system will follow the safety standards for household electronics outlined by the standard UL 22.

# DESIGN SPECIFICATION

## System Description

This specification describes a user-controlled, multi-wheeled robot with a rotating arm for grabbing nearby objects. The arm will have 135 degrees of freedom horizontally, and up to 90 degrees of freedom vertically. The robot is to be able to move at reasonable speeds of up to 0.5m/s through its environment, with the ability to quickly turn in any direction, and to move forward and backwards in addition to braking. The robot is to be able to detect immediate hazards and obstacles in its forward path up to 3 meters away and halt, and provide feedback to the user informing them of the event. The robot is to be able to grasp small, light objects on the ground or in the robot’s immediate vicinity for relocation. All robot functions are to be controllable using an Android phone application with controls for the motion of the robot, in addition to controls for the arm mechanism itself. The robot is to also be able to accept voice commands as a substitute for direct, physical interaction with the android interface. In addition, the robot is to support autonomous navigation from its starting location to a point determined by the user. The robot is to be low cost, flexible, easily reproducible, and intuitive to use.

## Specification of External Environment

The system is to operate in an environment such as a home, ideally with minimal clutter on the floor surfaces on which it will be operating. Room temperature and typical in-house lighting are the expected environment.

The unit will run off of a 9V battery pack attached to the robot chassis for easy swapping in the event of a dead battery. Uptime will be up to 5 hours.

## System Input and Output Specification

### System Inputs

The system will accept the following external inputs:

* Analog pulse from the distance sensor, 5V with a pulse width of 0uS to 38ms

The system will accept the following commands by means of button presses on the Android application screen. Each command is sent over Bluetooth on the 9600 baud serial connection as a specific character or string specifying an action for the Arduino to take:

* “w”, Move forward (move continuously while button is pressed)
* “s”, Move backward (move continuously while button is pressed)
* “d”, Turn right (move continuously while button is pressed)
* “a”, Turn left (move continuously while button is pressed)
* “b”, Brake (Halts ALL robot movement)
* “l”, Rotate arm left (move continuously while button is pressed)
* “r”, Rotate arm right (move continuously while button is pressed)
* “u”, Rotate arm up (move continuously while button is pressed)
* “n”, Rotate arm down (move continuously while button is pressed)
* “g”, Grasp/Release (toggle button)
* Receive Voice Command (activates API to convert speech to text, which is then converted to one of the following serial commands)

The following voice commands will be usable after pressing the “Receive Voice Command” button on the application:

* “1”, Go forward (robot moves until told otherwise)
* “2”, Go back (robot moves until told otherwise)
* “3”, Turn left (robot moves until told otherwise)
* “4”, Turn right (robot moves until told otherwise)
* “5”, Brake (robot halts ALL motion)
* “l”, Arm left ‘x’ degrees (character sent repeatedly until rotation complete)
* “r”, Arm right ‘x’ degrees (character sent repeatedly until rotation complete)
* “u”, Arm up ‘x’ degrees (character sent repeatedly until rotation complete)
* “n”, Arm down ‘x’ degrees (character sent repeatedly until rotation complete)
* “g”, Grasp
* “r”, Release

### System Outputs

The system will report the following information on the user interface via 9600 baud serial interface over Bluetooth connection:

* Current direction of travel as a string (“forward”, “back”, “turning”, “braked”)
* Arm status as a string (“rising”, “lowering”, “grasped”, “open”)
* Distance to nearest large object in front of the robot in cm
  + Range from 0 to 4000 ± 1cm
* PWM signals to the following motors
  + Drive motors: 0-5V DC with duty cycle from 0 to 100 in 1mV steps

## User Interface

The user will be able to continuously control the robot and query its sensors for data using a simple android touch interface or vocal commands. The following buttons are available:

Robot Direction:

* Directional arrows. Robot moves (relative to its current orientation) in accordance with the held arrow until it is released.
* Brake. Robot halts all motion.

Arm Control:

* Directional arrows to control the orientation and elevation of the grasper at the end of the arm. Arm moves as long as the appropriate button is depressed or until the arm reaches the limits of its rotation. Robot will only be able to move one discrete direction at a time.
* Grasp/Release toggle. Pressing the button causes the grasper to switch to the opposite state (from grasping to releasing the object, and vice versa)

Activate Voice Control:

* Press button to launch voice prompt to interpret voice commands.

Measurements and robot status information will be displayed in the top left corner of the android interface as text through a serial interface over Bluetooth.

The android interface will appear as follows: upon loading the application, the user will be presented with a splash screen while the Bluetooth pairing and initialization of the app occur. The application interface itself will have an upper set of buttons for directional control with a central button for sending voice commands. The lower set of buttons will operate the arm assembly, with the center button serving as the grasp/release toggle. The red button emblazoned with a brake pedal, positioned between the sets of controls and off to the right for ergonomic reasons, serves as the ‘Brake’ command that will halt all robot movement until a new input is given to the system.

|  |  |
| --- | --- |
| E:\Dropbox\EE\EE 478\478 Spring 2013\Final Project\Deliverables\Splash Screen.png  Figure - User app splash screen | E:\Dropbox\EE\EE 478\478 Spring 2013\Final Project\Deliverables\Interface_edited.png  Figure - User app interface |

The field where “Voice command” is listed in the location where textual status information will be displayed.

## Use Cases

The cases are given in the below diagram:



Figure - Use Case Diagram

***Control Robot***

The robot chassis will move in accordance with user input. If the Forward button is held down, the robot will continue to move forward in a straight line until the button is released or if an object is detected as too close (robot will automatically brake and the only available movements will be rotation or moving backwards). This system is controlled by the upper set of arrows in the top half of the android interface. In addition, speed control is an option to adjust how quickly the robot chassis may move.

Exceptions: If terrain is too cluttered, the robot may not be able to maintain true, straight motion as its path will be disrupted by debris. If Bluetooth connection is lost, error will occur.

***Control Arm***

The arm assembly will move in accordance with user input. If the user holds the Down button, the grasping end of the arm will lower itself closer to the ground until either the user releases the button, or if the grasping end reaches the limit of its lowering range. The grasping end of the arm can be commanded to either close its grip or release it. This system is controlled by the lower set of arrows and central button in the bottom half of the android interface.

Exceptions: If the object is too heavy or too large, the arm assembly is not guaranteed to be able to properly grasp the object, or to be able to lift it without potentially destabilizing the robot. If Bluetooth connection is lost, error will occur.

***Voice Control***

The user interface will support the conversion from spoken words in English to their respective commands as needed to control the robot and/or arm assembly as appropriate. For example, the user may instruct the arm to “Arm up 20 degrees” at which point the robot arm will rise such that the difference from its previous position and new position is 20 degrees, within tolerances. The mode may be activated by pressing the speaker icon in the center of the user display.

Exceptions: If user does not speak clearly with proper enunciation, their statement may be misinterpreted. If Bluetooth connection is lost, error will occur.

***Measure Distance***

The robot will return the distance from the front section of the robot to the nearest object in front of it that is at least .5sq ft in dimension. The button for this function is still under design consideration.

Exceptions: If an object is smaller than .5sq ft. there may be an error in detecting the object correctly. If Bluetooth connection is lost, error will occur.

***Autonomous Mode***

The robot will accept a given target distance to reach based on its current direction. The robot will then autonomously navigate to that location, keeping track of its position in the environment and navigating around nearby obstacles.

Exceptions: If the path is blocked with debris that blocks the path of the sensor, error may occur in position tracking. System is heavily dependent on the user to specify the correct destination as the robot lacks means of identifying location other than distance and orientation.

## System Functional Specification

The system is intended to operate and move around in the environment based on user input, with the input from the distance sensor as a safeguard from collisions. A measurement system will be implemented to convert the output pulse of the distance sensor into a user-accessible distance value, selectable as either inches or cm. This distance value will also be used directly by the robot for safety reasons to halt robot movement if it is in danger of a head-on collision.

The user will be able to select either continuous motion of the robot (moves until stopped) or motion that only lasts as long as the respective button is being held down.

The system comprises of several major blocks, as given in the following system block diagram.



Figure – System/Hardware Block Diagram

**Bluetooth Subsystem –** the Bluetooth subsystem facilitates communication between the Android user interface and the Arduino. The system controls serial communication between the two systems as necessary to allow system control and user interaction with the system.

**Ultrasonic Sensor** – upon being sent a minimum 10µs pulse, the sensor will send a burst of sound waves out and record their return times as a pulse whose length is proportional to the distance traveled. Because of the measurement being done via pulse length as opposed to a voltage level, an A/D converter is not required.

**Orientation Sensor** – Analog sensor that reports orientation of the robot relative to magnetic north, allowing the robot to know its current orientation. This combined with knowledge of its previous movements allows an approximation of its current location relative to its starting position.

**Time Base** – the time base in this case is the internal, 16MHz oscillator on the Arduino Uno board.

**Speed Control** – using SPI, a digital potentiometer and current control circuit determine how much power can drive the motors at a given time, enabling control of the speed of the robot’s motion.

**Drive and Arm Motors** – these are the simple motors that will be controlled using PWM signals based on user input. They operate on 2.7 to 5V DC.

**Interface Feedback** – this is a system that controls what output is sent from the Arduino to the user interface on the Android device, such as status messages about robot movement, arm movement, and object detection.

The activities required to perform a distance measurement are as given in the following activities diagram.



Figure - Activity Diagram for Distance Measurement

## Operating Specifications

The system shall operate in a typical household environment:

Temperature Range 40 – 120F

Humidity up to 90%

Power 9V battery

The system shall operate for a minimum of 5 hours on a fully charged 9V battery

## Reliability and Safety Specification

MTBF: Minimum 2,500 hours

The system will follow the safety standards for household electronics outlined by the standard UL 22.

# TESTING

## Test Plan

To ensure the design meets the specified requirements, a number of tests must be performed on the system. The following sub-systems must be tested for correct performance and reliability: Bluetooth connection, serial connection via Bluetooth, Android interface, distance sensor, motor control, and autonomous motion.

The Bluetooth connection must be checked that it can correctly pair with the user’s Android device quickly and reliably with the ability to send and receive serial data to and from the paired Arduino at a baud rate of 9600. The quality of data must also be checked, to see that the proper data is arriving and being transmitted at both ends of the connection. Secondly, the interface developed for the Android device must be tested for correct operation. The interface buttons are to appear depressed when touched by the user finger, and the proper user feedback shall be displayed in the text field of the interface as necessary. User speech interpretation must also be verified for correctness based on the user’s accent and ability to speak clear English. Lastly, the motor control system must be checked for consistency and reliability. The motor must rotate in the correct direction for the given PWM, at the correct speed for the given voltage, amperage, and gear ratio with which the motor is configured. This also applies to the servos, which are expected to rotate a specific number of degrees when told to do so. Autonomous motion must also be verified for accuracy and correctness of the motion algorithm.

# FAILURE MODES ANALYSIS

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Item/ Function** | **Potential Failure Mode** | **Potential Cause** | **Local Effects** | **Probability** | **Severity** | **Mitigation Requirements** |
| Bluetooth adapter | Loss of pairing with android | Distance from paired device, current overload | Loss of control of robot | (B) Remote | (V) Critical | System will hang and attempt to regain paired connection. Robot will not operate until repaired to prevent unwanted movement. |
| Ultrasonic Sensor | Failure to return pulse | Current damage to circuitry, insufficient pulse from Arduino | Distance data unavailable | (B) Remote | (III) Minor | Robot will operate as normal without the collision detection function available. |
| Servo | Motor failure | Insufficient power, mechanical failure | Inability to move related section of arm | (A) Extremely Unlikely | (IV) Moderate | Arm will move based on still available servos, degrees of freedom from damaged servo will not be available |
| Voice interpreter | Improper recognition | User speaks unclearly, with heavy foreign accent | Command values may be misread, command type misread | (D) Reasonably Possible | (II) Very minor | User must speak clearly or be able to recognize the improper command and use the halt command to cancel movement. |

# BILL OF MATERIALS

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Quantity** | **Price/unit** | **Total** |
| Arduino Uno | 1 | $21.95 | $21.95 |
| HS-422 Servo Motor | 4 | $9.99 | $39.96 |
| Bluetooth Module | 1 | $9.49 | $9.49 |
| Vehicle chassis + wheels | 1 | $20.04 | $20.04 |
| H-bridge | 1 | $6.00 | $6.00 |
| Standoffs (Bulk) | 1 | $10.00 | $10.00 |
| Ultrasonic Sensor | 1 | $5.19 | $5.19 |
| Acrylic sheet | 1 | $20.00 | $20.00 |
| Jumper wire set | 1 | $10.00 | $10.00 |
| Breadboard | 1 | $13.00 | $13.00 |
| Tamiya Double Gearbox | 1 | $10.19 | $10.19 |
| 10k Digital Potentiometer | 1 | $1.50 | $1.50 |
|  |  |  |  |
|  |  | **Grand Total** | **$167.32** |

# APPENDIX



Figure - Software Block Diagram



Figure - Functional Decomposition



Figure - Control and Data Flow Diagram



Figure - System Activity Diagram



Figure - System sequence diagram

# SCHEDULE