**Final Project**

**Voice Controllable Mobile Arm Platform**

**Design Spec**

**Lab Team: Tianshu Bao**

**Shawn Stern**

**Dan Tran**

**Lab Section: EE 478**

**Date: May 17, 2013**

TABLE OF CONTENTS

[1 TABLE OF FIGURES 2](#_Toc356513471)

[2 REQUIREMENT SPECIFICATION 3](#_Toc356513472)

[2.1 System Description 3](#_Toc356513473)

[2.2 Specification of External Environment 3](#_Toc356513474)

[2.3 System Input and Output Specification 3](#_Toc356513475)

[2.3.1 System Inputs 3](#_Toc356513476)

[2.3.2 System Outputs 4](#_Toc356513477)

[2.4 User Interface 4](#_Toc356513478)

[2.5 Use Cases 6](#_Toc356513479)

[2.6 System Functional Specification 7](#_Toc356513480)

[2.7 Operating Specifications 9](#_Toc356513481)

[2.8 Reliability and Safety Specification 9](#_Toc356513482)

[3 SCHEDULE 10](#_Toc356513483)

# TABLE OF FIGURES

[Figure 1 - User app splash screen 5](#_Toc356513484)

[Figure 2 - User app interface 5](#_Toc356513485)

[Figure 3 - Use Case Diagram 6](#_Toc356513486)

[Figure 4 - System Block Diagram 8](#_Toc356513487)

[Figure 5 - Activity Diagram for Distance Measurement 9](#_Toc356513488)

# REQUIREMENT SPECIFICATION

## System Description

This specification describes a user-controlled, multi-wheeled robot with a rotating arm for grabbing nearby objects. The robot is to be able to move at reasonable speeds 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 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. 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.

## 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 0µs to 38ms ± 2µs

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:

* 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 4000 ± 1cm
* 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.
* 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 1 - User app splash screen | E:\Dropbox\EE\EE 478\478 Spring 2013\Final Project\Deliverables\Interface_edited.png  Figure 2 - User app interface |

The field where “Voice command” is listed in **Figure 2** is the location where textual status information will be displayed.

## Use Cases

The cases are given in the below diagram:



Figure 3 - 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.

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 or Arduino loses power, the interface will return to the flash screen until a connection is reestablished.

***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 or Arduino loses power, the interface will return to the flash screen until a connection is reestablished.

***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 the misinterpreted command is not a command usable by the system, nothing will occur and the user will have to repeat themselves.

If Bluetooth connection is lost or Arduino loses power, the interface will return to the flash screen until a connection is reestablished.

***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. If there is no object in front of the robot to be detected that is large enough, the system will return “all clear” indicating that there are no obstacles within sensor range.

Exceptions:

If an object is smaller than .5sq ft. there may be an error in detecting the object correctly and an incorrect distance measurement will be given based on the signal from the sensor.

If Bluetooth connection is lost or Arduino loses power, the interface will return to the flash screen until a connection is reestablished.

## 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 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.

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

**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, clean 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

# SCHEDULE