April 26, 2013

Team 19 Overview

**Laboratory # 10: Delivery**

**Morgan, Laura**

**Miaw, Jireh**

**Hauser, Steven**

**Dworak, Catherine**

**Bertoglio, David**

***Work Product***

**Overview of what our team built**

***Document Revision Information***

**April 26, 2013 – Document created**

**Approval Sheet**

**All group members whose names are listed below approve of the document and contributed fairly.**

**Member Names**

**Morgan, Laura**

**Miaw, Jireh**

**Hauser, Steven**

**Dworak, Catherine**

**Bertoglio, David**

**Pledge**

**On my honor, as a student, I have neither given nor received unauthorized aid on this assignment.**

**Names**

**Morgan, Laura**

**Miaw, Jireh**

**Hauser, Steven**

**Dworak, Catherine**

**Bertoglio, David**

Contents

**Word did not find any entries for your table of contents.**  
In your document, select the words to include in the table of contents, and then in the Formatting Palette under Styles, click a heading style. Repeat for each heading that you want to include, and then insert the table of contents in your document. You can also create a table of contents by clicking the Create with Manual Formatting option and then type the entries manually.

Our team was given the task of the developing of the control system for a mobile robot that performs simple motion and has sensors that collect data about the environment. The robot is controlled from a base-station computer that communicates with the robot via Bluetooth to send and receive commands and telemetry data. The hardware is a Lego Mindstorms NXT robot utilizing a programmable microcomputer brick. The goal of this project was to design and create the software for this robot utilizing the software engineering process.

This robot is built out of legos with 3 wheels, 2 motors, and 4 sensors, including light, sound, ultrasonic, and microphone, as well as a swinging arm. It can perform simple movements, such as moving straight forward and backward, moving in an arc forward and backward, left and right, and turn in place left and right. These movements are controlled using the w-a-s-d keys on the keyboard. In addition to these basic movements, the robot has a few special features. 1) The robot has a swinging arm that swings clockwise when the key is pressed. 2) The robot turns exactly 180 degrees when the t key is pressed. 3) As a self-defnese mechanism, the robot moves backward when it detects loud sounds. All of these actions are performed by the interaction of the base station control system and the on-board system.

The base-station system runs on a computer and is operated by a human. This system takes commands from the user and sends messages to the on-board system to perform those commands on the robot. It contains a graphic user interface, consisting of w-a-s-d buttons, text fields to display sensor readings, buttons to refresh the sensor readings, buttons to control the swinging arm and then 180 degree turn, as well as buttons to change the speed of the robot. All of the functionality of the robot can be performed using this GUI. The base-station contains only 2 classes – a baseStation class that performs all the action (creating and sending messages when required), and a GUI class that interacts with the baseStation class when buttons are clicked.

The on-board system software runs on the microcomputer brick physically attached to the robot, and controls the movement of the robot based on commands it receives from the base-station. The on-board software consists of 3 classes – the activator class which establishes Bluetooth connection and sends and receives messages, the messageHandler class which decodes and encodes messages, and the driver class which accesses the NXT motors and sensors to control the movement of the robot and receive telemetry data. This class implements the Lejos API, a package class developed from Java that provides access to NXT motors, sensors, etc.

These two systems send messages back and forth via Bluetooth using a communication protocol developed by the team. The protocol is an agreed upon document that describes the composition of messages. Each message consists of 11 characters. The first two characters determine the type of command, the next 8 characters are used for various parameters for the command, and the last character holds the checksum used for error detection.

Finally, the robot system contains a debugging system. The base-station contains a GUI for the debugger, which can set breakpoints in the code, access state variables, and contains a log of previous messages sent and received. This is used to access the on-board software remotely in case an error occurs in the field.

The development of this robot control system took place over the course of a semester, utilizing 9 team members. These team members were separated into 2 groups, one that developed the on-board system and one that developed the base-station. The interaction and cooperation of every member of these two groups was essential for the completion and success of this robot.