Wireless, Web-Controlled, Ball-Collecting Robot

Sara C. Wieland
wielsc@loras.edu
Adam J. Dunn
dunnaj@loras.edu
Danial J. Neebel, Associate Professor
dneebel@loras.edu
Department of Mathematics and Computer Science
Loras College

Blake M. Dirksen
dirkbm@loras.edu
Thomas J. Gotto
gotttj@loras.edu
Simon G. Halbur
halbsg@loras.edu
Angela M. Torres
torram@loras.edu
Todd N. Swift, Assistant Professor
toswift@loras.edu
Department of Physics and Engineering
Loras College

Abstract

A wireless, web-controlled, ball-collecting robot was developed using engineering design principles in order to navigate a specified terrain containing walls and obstacles. The robot utilizes an on-board micro controller to control the robot's drive motors and to read its object avoidance sensors. The user accesses the system through a web browser and uses a graphical interface to issue commands. The micro controller on the robot receives commands via a wireless serial communication link from a web server. Designing the robot required students to call upon their experience and knowledge of several disciplines. The team members learned to work together to complete a task more complex than was previously encountered in their undergraduate experience.

Introduction

This paper is the result of a semester-long interdisciplinary project between Computer Science and Electromechanical Engineering students at Loras College in Dubuque, Iowa. The project overview, hardware and software design, and the educational value of this capstone experience will be presented.

Project Overview

The seniors of the Computer Science and Electromechanical Engineering programs were assigned the task of completing an interdisciplinary project. This project was intended to integrate the backgrounds of these two disciplines. The engineering students were to draw upon their experience in mechanical and electronic design and the computer science students utilized their experience in the software development process. It was required for the team to be self-managed such that the project would be completed in one semester. The project was to create a wireless, web-controlled robot that could navigate an arena and retrieve ping-pong balls. The robot needed to be intelligent enough such that it would ignore commands that could cause the robot to damage itself. The robot would be controlled by a hand-held, battery-powered micro controller- the Handy Board. By the end of the semester the team had successfully designed, constructed, tested, and finalized the robot; writing a thesis paper and giving a presentation completed the project.

Hardware Design

The design process began with full group brainstorming in which several basic ideas were discussed. The ideas were evaluated based upon the possibility for success, ease of building, and cost. Once a basic design was selected, a prototype was built and tested. Based upon the results, several modifications were made. Discussed below are the final design specifications.

Size

The robot itself fits into a 10" cube. The maximum height minus the flexible antenna is 8". The maximum length is 10" and the maximum width is 7.25". The weight of the robot with no ping-pong balls contained in the storage chamber is 4.2 lbs.

Chassis

A triangular design was decided upon because the three-wheel assembly naturally creates a triangle. The drive motor is at the tip of the triangle and the other two wheels make up the two bottom corners. The chassis is composed of two levels in order to save space on the robot. The upper level allows the drive wheel to turn a full ninety degrees, and the lower level provides room for the other robot components (see Figure 1). The chassis of the robot is made of two sections of 1/10” Plexiglas joined together with a block of Georgia Pacific Flakeboard’s “Superior MDF” medium density fiberboard.

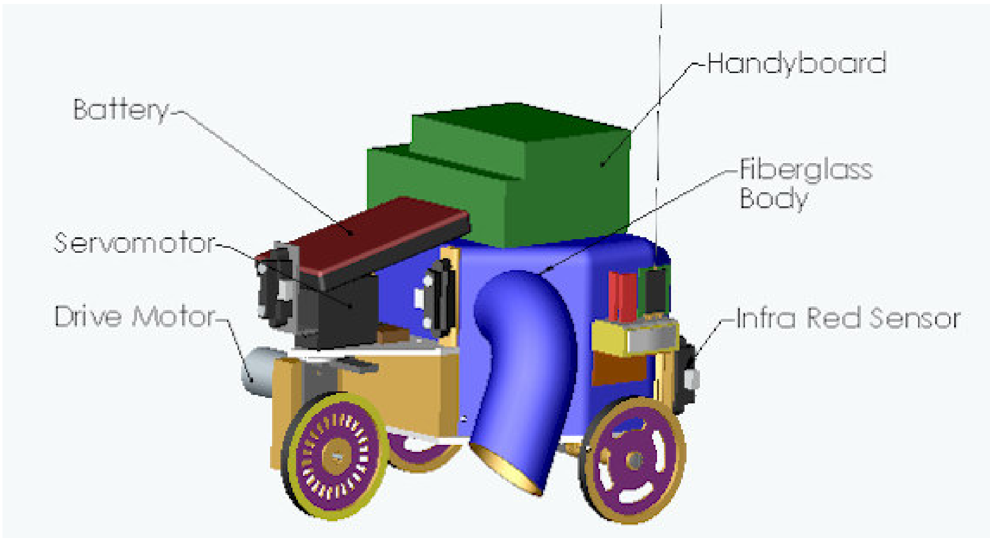

Figure 1: Robot components.

Drive Train

The drive train consists of five main parts: the drive-wheel mounting bracket, drive-wheel, servomotor, drive-motor, and shaft encoder. The robot receives individual commands from the user to move forward, move backward, turn left or turn right. A singular drive wheel and two free spinning wheels were decided upon because of simplicity. A two drive-wheel system is more difficult to control due to inconsistencies between motors. In order to turn with this system, the front wheel assembly is rotated ninety degrees with a servomotor. The drive motor then turns the robot when powered. Another benefit of the design is a small turning radius, which helps prevent the robot from reaching deadlock situations such as being trapped in a corner. The final component of the drive train is the shaft encoder. This component allows the robot to measure its distance traveled while traveling straight and turning.

The drive-wheel mounting bracket serves four main purposes. The first is to attach the drive wheel to the chassis. Second, by attaching to the servomotor, the bracket allows the drive wheel to be rotated. The third purpose is to allow for a place to secure the geared drive motor. The final purpose is to allow for a place to mount the sensor used as a shaft encoder.

The drive-wheel mounting bracket is made out of an aluminum sheet with a thickness of 1/8”. The front wheel consists of a plastic wheel with a cardboard insert that contains 24 slots spaced evenly around the disc. The wheel is located at the horizontal center of the bracket, and a shaft that connects to the drive-motor's shaft runs through its center. A Futaba® FP-S148 servomotor attached to the drive-wheel mounting bracket allows the front wheel to rotate 90 degrees allowing the robot to turn. The servomotor turns the mounting bracket through a simple gear bearing system. A gear on the servo turns another gear attached to the bearing with a one-to-one ratio. A geared 12V DC motor connected to the drive wheel and mounted to the drive-wheel mounting bracket drives the robot. A Quality Technologies QRB1114 Infrared Reflective Optosensor is mounted on the side of the gearbox aligning the sensor with the holes in the drive wheel. The sensor is used as a shaft encoder to determine the distance the drive-wheel has traveled by counting the number of holes it senses. This allows the Handy Board to know what angle the robot has turned and how far it has traveled forward or backward.

The robot has three plastic wheels, each being 2 ¾" in diameter and having a width of 5/16". An outside rubber band wraps each of the wheels, serving the purpose of increasing friction for slip avoidance. The wheels are arranged in a triangle with a wheel located on each vertex. The two rear wheels are able to spin freely and independently of each other.

Ping-Pong Ball Pickup and Storage

A vacuum system was decided upon because of feasibility and simplicity. The vacuum design allows balls to be picked up on a rough terrain or near other objects. This design does not require any major control algorithms. The only activity for picking up the balls is to turn on the fan that generates the vacuum.

The ping-pong balls are stored in a fiberglass body that has a tube located on the robot's left side. The vacuum fan will turn on when the user gives the pick-up command and will stop after five seconds. A wooden board divides the vacuum chamber from the outside air allowing for a vacuum. A fan-motor assembly obtained from a hand vacuum is attached to it. An N-channel power MOSFET transistor acts as a switch to turn the hand vacuum on.

Obstacle Avoidance

The final design for obstacle avoidance uses sensors that are read by software to detect objects and walls in the path of the robot. The robot senses an object when it is approximately five inches away and comes to a stop two inches from the detected object. Five Sharp GP2D12 Infrared Object Detectors were positioned around the robot to optimize the range of sight. This model of sensor was decided upon because of its ability to see objects at fairly large distances and to see narrow objects. (Figure 1 shows three of the sensors).

Four of the sensors are mounted on hinges that allow the sensors to be mounted at any desired angle relative to the fiberglass shell. The hinges are fixed at a desired angle by means of a pair of setscrews along the joint of the hinge. The other sensor is mounted to an aluminum L-bracket and faces straight forward.

Handy Board Positioning and Wiring

The Handy Board is located on top of the fiberglass shell. It is attached with Velcro to the shell. This prevents it from falling off the robot and to allow easy access to each of the wires that are running to the Handy Board from the distance sensors, drive motor, shaft encoder, servo motor and power MOSFET.

Wireless System

The wireless system used for this project consists of a transmitter (TXLC-434) and a receiver (RXLC-434) from Reynolds Electronics (www.rentron.com). The wireless system has a baud rate of 4800Kbps, operates at a frequency of 433 MHz and is powered by 5 Vdc. In order to communicate with the robot, the server sends information to the transmitter via a serial interface. The Handy Board will then receive this information through its serial interface with the receiver. This receiver is attached to the side of the robot (see Figure 2).

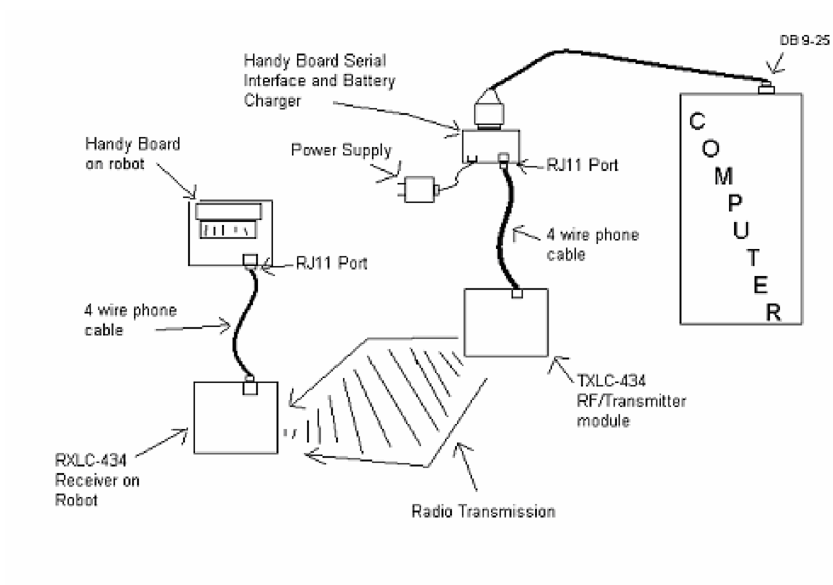

Figure 2: Communication between server and robot.

Hardware Verification

Once the robot was constructed, the reliability of the various systems was tested. To test the shaft encoder system, the robot was given movement commands and the distance traveled was measured. Using this information, the number of holes counted by the sensor per inch could be determined, and the calibration of the robot's motion was achieved. The obstacle avoidance was tested by simply commanding the robot to move towards an object and observing whether it came into contact with the obstacle. If it did not, the distance from the wall was measured. The test of the vacuum system consisted of activating the vacuum, placing the ping-pong ball at various positions relative to the vacuum, and verifying that the ball could be retrieved. The wireless system was tested by sending commands from one computer station to another and verifying that the sent messages were received accurately. The final test consisted of integrating all the systems and testing the full range of available commands.

The system performs the assigned tasks, but not without the occasional problem. The shaft encoder system is not ideal and therefore distances are occasionally off. They are never off by more then 10 percent of the designated distance or angle. Aesthetically, the robot could have been better designed to be more appealing. Due to the limited number of sensors, there are blind spots in the robot's vision, meaning it will occasionally run into an obstacle or a sharp corner, but it will not run into walls. In the few months the robot has been operating, the body has shown little signs of mechanical fatigue, and overall, the system performs very well.

Software Design

The software components of the project were completed in two parts. The server software, on the computer in Figure 2, accepts or rejects client connection. When a client indicates they want to send a command to the robot, the server processes the command. It then wirelessly sends the commands to the robot via a radio frequency (RF) transmitter. The RF receiver is mounted on the robot and is connected to the Handy Board. The Handy Board utilizes the software loaded on it to interpret and then to execute the command from the receiver.

Java was the language that was used in the final coding process for the client/server software. Java was chosen because of its ability to be run on multiple platforms. On the user's side of the client/server software, it was decided to use Java applets instead of applications to make the process easier for the end user. This method enables the client to be embedded into the user's browser.

Interactive C was the language chosen for the robot's on-board software. Nearly all prior programming experience by the Computer Science team was in C++ or variations thereof. Therefore, Interactive C was a natural selection for this portion of the software design because of the similar syntax and semantics of the languages. This software runs on the Handy Board.

Since the end user applets were event-driven (button clicks, etc.) it was decided to design all the software in this fashion. To visually depict the design of the software, finite state diagrams were used for their capacity to represent event-driven programming.

User Web Interface

Figure 3: Client login page.

The client interface (see Figure 3) is executable on any Java-enabled web browser running at least Java v.1.4.1. The user views a logon page and is prompted to enter a username (any combination of alphanumeric characters, as long as it does not duplicate an existing username) along with a master password. The password is stored in a text file, on the server computer, for security and ease of changeability.

Any number of users may be logged in at one time, but only one user has the ability to control the robot at any particular instant. Newly logged-in users are appended to the bottom of a queue (user list). The user in control has a maximum control time of 10 minutes and is automatically forced to give up control after 2 minutes of inactivity. The user also has the option to end their control session prematurely by clicking a button. After a user's control is terminated, the next user in the queue is granted control and the former controller is appended to the bottom of the queue. The queue is visible to all users such that they know whom the other users are and when control will be granted.

All users are able to access an instruction link on how to use the client. Each user is able to view the robot's arena via an overhead web camera and also has the option to converse with other users by sending text messages (see Figure 4 below).

Chat

Once logged in, the user has the option to communicate with other users via a chat room (an instant messaging service where users type text messages which are displayed to all users). A text box is available for the users to enter messages, and the messages are sent by pressing either the client send button or the keyboard enter key. When a text message is sent, the message appears in a separate window, preceded by the corresponding username. This window displays the running conversation between all users.

Controls

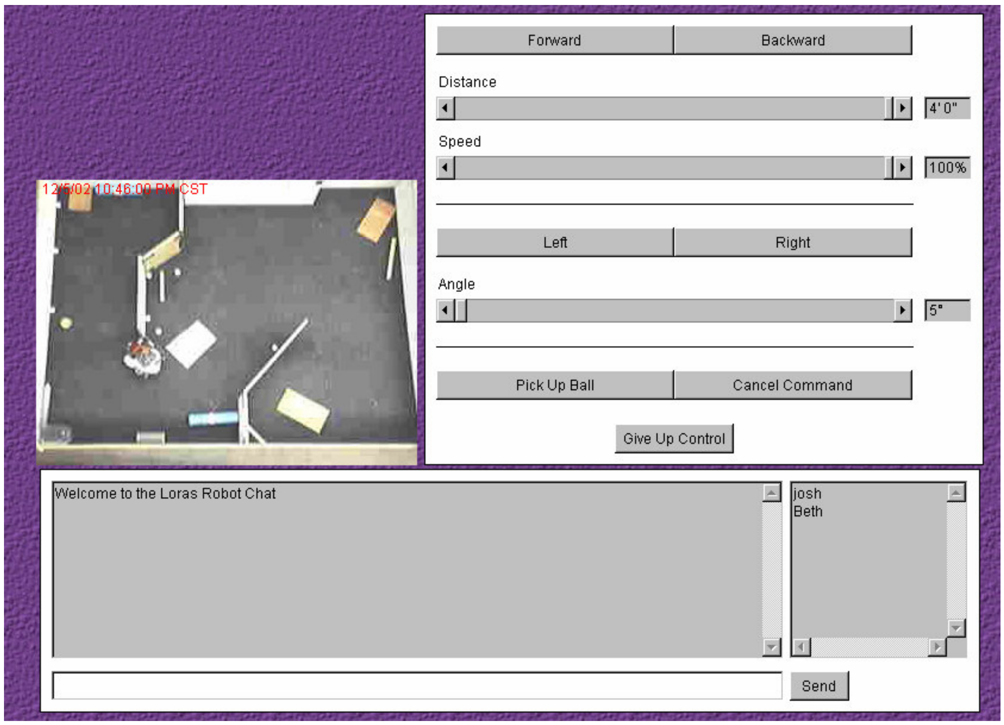

Figure 4: Client control page showing web camera view (top left), controls (top right) and chat window (bottom).

The user commands the robot to move using buttons and sliders on the client interface. Buttons are used for forward and backward motion, left and right turns, giving up control of the robot, the action of picking up a ping-pong ball, and canceling commands. While a command is being executed, all user inputs are ignored; the only exception to this is if the user cancels a sent command by clicking the cancel button on the client interface.

Slider bars are used for selecting the distance and speed of motion as well as the angle of turning. Speed is displayed to the user as a percentage of maximum speed. The fastest the user is allowed to traverse the robot is 12 inches/second (100%) and the slowest is 1.2 inches/second (10%). Distance is displayed to the user as a range between 0 feet, 1 inch and 4 feet, 0 inches. The user specifies angular movement for left and right turns, which is displayed as an angle ranging from 5 to 180 degrees.

Communication Specifications

The Handy Board's serial port, by default, is set for 9600 bps, no parity, and 1 stop bit. The baud rate was set at 300 bps in order to allow for compatibility between the server computer and the wireless transmitter-receiver set. Requests from the user are processed by the server and sent through the communication channel. Depending on the specific command, each transmission is formatted by two start bytes, one command byte, one variable byte, two stop bytes, and two different error detection devices that are one byte apiece.

Table 1: Breakdown of transmission.

Portion of Transmission	**Size**	**Bytes Used (in decimal)**
Beginning of Transmission (BOT)	2 Bytes	16 02
Commands	1 Byte	01 (forward)
		02 (backward)
		03 (right)
		04 (left)
		05 (pick up)
		06 (cancel)
Variables for Commands	1 Byte	1-48 (Distance in inches- commands 01 & 02)
		10-100 (Speed in percent- commands 01 & 02)
		5-180 (Angle in degrees- commands 03 & 04)
End of Transmission (EOT)	2 Bytes	16 03
Checksum	1 Byte	Sum Mod 256
	1 Byte	Sum Div 8

Software Flow of Control

Login Server

The purpose of the login server is to verify the access rights of a user through the use of a username and password. Upon entering this information into the login applet, the server checks if the entered username is already in use and also compares the entered password to the correct password stored on the server. If both the username and password are verified, the user is then "logged in" to the system and entered into the queue for controlling the robot (see Figures 3 and 4).

Chat Server

The purpose of the chat server is to allow the users to communicate with each other by sending text messages. When a user enters a message into the chat applet, the message is sent to the chat server. The server sends it to every user's chat applet, which display it on their message windows.

Control Server

The purpose of the control server is to allow the user at the top of the queue control of the robot. By manipulating the control applet's slider bars and buttons, the user may command the robot to move forward or backward, turn left or right, pick up a ping-pong ball, or cancel a command already in progress. These commands are relayed wirelessly to the robot.

Threads

The login, chat, and control portions of the software operate through the use of threads. For example, all three aspects use listen threads to monitor for user input. Upon reception, the listen thread spawns another thread to process the input. Specifically, this includes validating login information as well as transmitting text messages and robot commands.

Robot

The software that is loaded on the Handy Board receives forward/backward movement, left/right turning, ball retrieval, and cancel commands wirelessly from the server and then causes the robot to act on those commands. This software also constantly monitors the obstacle avoidance sensors, such that a motion command is terminated if continued execution could cause damage to the robot (e.g., running into a wall).

Software Verification

Each of the software projects (Java Server, Java Applets, and Robot Software) was broken down in to sub-projects, which were individually coded and tested before bringing them all together. Each of the sub-projects, as well as the combined project, was tested on multiple desktop and laptop computers running Windows 2000 and Sun's Java Runtime Environment v 1.4.1. The web browsers that were used in testing were Microsoft's Internet Explorer 5 and 6. All testing was done over the same local network as the server computer.

Java Software Testing

The combined java servers and applets were tested with up to eight users at a single time. The login applet was tested to make sure that duplicate login names were not allowed.

Sending consecutive alphabetical characters to multiple users tested the chat applet; this was done in order to ensure no chat messages were being lost in transit. Testing was also completed to make certain that users could not access the control page without first logging into the server.

The control applet was tested to verify that the correct robot commands were being transmitted to the server. The most important part of testing this portion of the combined project was to make sure that the control of the robot was transferred from user to user. Extensive testing was also completed to guarantee that when a user in control of the robot presses the give up control button, has been inactive for two minutes, has been in control for ten minutes, or logs out of the system, control would then be passed to the next user in the queue.

Robot Software Testing

The robot software was tested as a whole by setting the robot in the arena and sending commands to the robot. It was tested to make sure that the robot was traveling and turning the specified distance and that it would not collide with obstacles. The Cancel command was tested to see if it was successfully terminating the current action of the robot. Finally, a test program that activated and deactivated the vacuum apparatus at set time intervals was loaded onto the Handy Board.

Educational Capstone Experience

Interdisciplinary Project

Designing the robot required students to call upon their experience and knowledge of several disciplines. Senior Computer Science and Electromechanical Engineering students worked jointly to incorporate proven software development processes as well as electrical and mechanical engineering design principles to produce the robot. The interdisciplinary approach to this project was intended to emulate the nature of professional careers.

Demonstration of Undergraduate Background

The team members learned to work together to complete a task more complex than was previously encountered in their undergraduate experience. Knowledge gained from the team’s undergraduate studies was utilized in the following ways: the Computer Science students drew upon their knowledge of software engineering, timed events, concurrent programming, TCP/IP communication via the Internet, logic, and system architecture, while the Engineering students demonstrated their background in circuits, sensors, logic, mechanics, and a wide variety of materials.

Student Management Plan

Under the supervision of two instructors (one Computer Scientist and one Engineer), the team was required to manage itself throughout the project. In order to do so, goals and deadlines were organized into a timeline, a logbook was utilized for each session, and leadership roles were rotated among the members. In addition, individuals were assigned particular tasks, which were expected to be completed according to the timeline. It was necessarily critical that these tasks be successfully completed in order for the project to be accomplished on time.

Conclusion

This capstone project was successfully completed given the specified design requirements. The team members not only learned a great deal within their individual disciplines but also of the extreme significance for quality documentation and communication.

References

Campione, M., Huml, A., & Walrath K. (2001). *The Java Tutorial, Third Edition*, New York: Addison-Wesley.

Martin, Fred G. (2000. November 15). *The Handy Board Technical Reference*. Retrieved from the World Wide Web: http://handyboard.com/techdocs/hbmanual.pdf