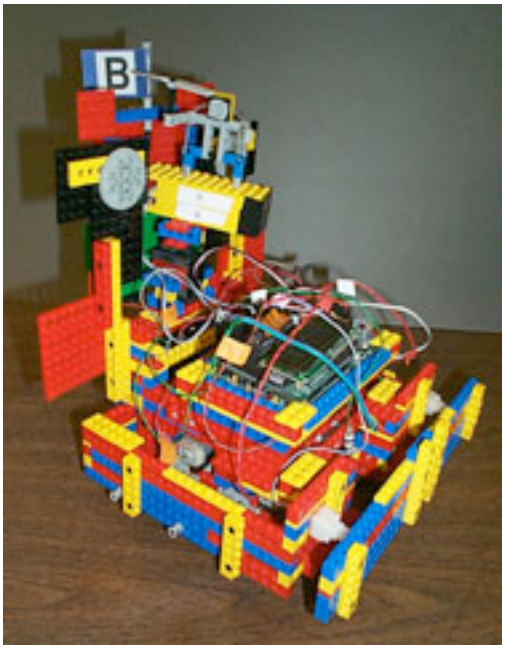
Integrating Bluetooth onto LEGO robots

Interim Project Review
March 7, 2003

Prepared for
Prof. Sree N. Sreenath and Prof. Behnam Malakooti



Prepared by

Team Members

Andrew Jones

Mike Krofcheck

Faculty Advisor(s)

Frank Merat
Associate Professor, EECS

Richard Drushel
Senior Research Associate, Biology

Executive Summary

The student team of Andrew Jones and Mike Krofcheck will add Bluetooth communication capabilities to the microcontroller utilized in EECS 375. The wireless communication capabilities provided by Bluetooth will allow robots designed by students to communicate with each other and with a computer to download program updates. A faster microcontroller and more modern wired serial architecture that provides greater bandwidth, USB, will also be provided on the redesigned microcontroller board for faster serial access if so desired.

The project was broken down into two main sections: hardware and software. The hardware phase was broken into four sub-phases: (1) order and obtain parts; (2) add Bluetooth to the microcontroller board, (3) add USB to revised microcontroller board; (4) layout and build microcontroller if time. To date, the project team has obtained the Bluetooth controller —Windigo BTM02C2XX-R—and the microcontroller—Motorola M68EVB912B32 68HC12 development board—necessary to continue further work on the project. Currently, the project team is working to integrate the Motorola 68HC12 microcontroller unit with the Bluetooth controller. This involves the fabrication of a Printed Circuit Board (PCB) for the Bluetooth module and its supporting components.

The software phase was broken down into five sub-phases: (1) obtain necessary hardware to programming microcontroller; (2) modify microcontroller code for Bluetooth; (3) create user libraries for robots; (4) code validation; (5) create GUI for computer. Once the Bluetooth controller has been integrated with the microcontroller (hardware task), a Hardware Control Interface (HCI) will be established, allowing the microcontroller to control the Bluetooth module.

Introduction

EECS 375, commonly referred to as the "LEGO lab class", has become a popular course offering in the EECS department. In fact, the course has become so popular, that the course instructors, Dr. Richard F. Drushel, Dr. Hillel J. Chiel, and Dr. Randall D. Beer, have to keep a waiting list for the course.

Modeled after the original LEGO lab course, MIT 6.270, designed by Fred Martin, Randy Sargent, Anne Wright, P.K. Oberoi, *et al*. at Massachusetts Institute of Technology, EECS 375 employees a C-programmable 68HC11 microcontroller board which provides significantly more control and functionality than the LEGO Mindstorms set utilized in ENGR 131. However, the microcontroller boards are much more fragile than the robust and durable Mindstorms RCX "bricks." Through the use of 411 students over the last 15 semesters of the class's existence, the microcontroller boards have received a fair amount of use. Due to this heavy use, the boards are now beginning to show their age; components are beginning to break off the boards, cables are coming loose, and the general functional limitations of the boards have resulted in several "copy cat" designs from semester to semester. The instructors of the course are worried that many of the microcontroller boards used in the course won't be able to make it more than another one to two semesters. In addition, the original board suppliers have informed the instructors that production has ceased on the current board design used for the course.

Due to this apparent need, the student group consisting of Andrew Jones and Mike Krofcheck, have decided to redesign the microcontroller board for EECS 375. However, the project is to go beyond just a simple redesign of the board; three new element will be added to the microcontroller board—Bluetooth communication, a USB port, and an updated microcontroller--that will foster additional student creativity in EECS 375.

The wireless communication capabilities provided by Bluetooth will allow robots designed by students to communicate with each other and with a computer to download program updates. It will also eliminate the use of serial cables currently used to program the robots. However, a more modern wired serial architecture that provides greater bandwidth, USB, will also be provided on the redesigned microcontroller board for faster serial access if so desired.

Due to the amount of work involved in such an undertaking, the project was originally broken into two semesters. The first part, conducted during the Fall '02 semester, involved investigating the wireless communication options available, deciding on a standard wireless protocol (or an original protocol), creating a design to demonstrate the wireless capabilities, and prototyping the wireless design in an effort to prove its feasibility for inclusion on a redesigned microcontroller board suitable for use in EECS 375. Due to circumstances outside of the student group's control, the student group was unable to complete all the objectives outlined. After conducting research on the available wireless protocol options, Bluetooth was chosen for inclusion on the microcontroller board due to its low cost, low power requirements, and high data bandwidth. However, acquisition of the Bluetooth modules was more difficult than originally anticipated. As a result, the second part of the project will involve integrating the Bluetooth modules onto the current microcontroller board; skipping the design of a board to demonstrate Bluetooth's capabilities. At the conclusion of the Spring '03 semester, the proposed plan will provide the following deliverables:

- A modified microcontroller board currently used in EECS 375 containing Bluetooth

- Schematics for a redesigned microcontroller board to be used in EECS 375 that includes Bluetooth connectivity, a USB port, and an updated microcontroller; time permitting, the redesigned board will be constructed
- Software for both the computer and microcontroller board supporting the new technologies implemented on the redesigned microcontroller board

No prior work on redesigning the microcontroller board has taken place. However, a previous senior project group consisting of Bruce E. Brown and Dan W. Baker made an attempt at implementing a proprietary wireless protocol for the robots used in EECS 375, Restrictions of the design included the complicated circuit needed to implement the protocol, power requirements, and the inability to quickly start communication between multiple robots. In addition, an MS student designed his own wireless RF communication protocol for use in a wireless wearable health monitoring system. Another senior project group, lead by Phillip Fultz and Ryan Hollinger, worked on a similar project—"Medical Astronaut Monitor System (MAMS)"—but used Bluetooth technology instead of a proprietary protocol.

Since the project involves both a hardware aspect and a software aspect, two separate requirements sets have been created.

The final software will meet the following objectives:

- The microcontroller software as well as the software on the computer will have the ability to identify other Bluetooth devices and differentiate between devices. Differentiation includes the ability to decipher robots from the same team and opposing teams.
- Direct connections will be allowed between Bluetooth devices (i.e. computer-to-micro and micro-to-micro).

- Computer software will provide an interface to program and control one or more microcontroller boards using either Bluetooth or a USB connection
- *The hardware will meet the following objectives:*
- Communication interfaces: Bluetooth for wireless and USB for wired connections
- Two wireless devices (i.e. two robots) should be able to communicate with one another without interaction with computer.
- Wireless device (i.e. robot) should be able to communicate with a computer in both directions. Communication functions should include, but are not limited to, downloading microcontroller code (student-written programs) to the wireless device, sending signals to the computer from the wireless device, and receiving control commands from the computer.
- Wireless communication should occur at 500 Kbps or enough bandwidth to allow for 5 messages to be sent between the computer and wireless device every second (5 messages/s).
- 25 ft. circumference zone of communication; enough range for LEGO egg competition, which takes place in EECS 375, and for common updating of microcontroller code. Refer to Figure 1 for an illustration of the egg hunt arena.
- Low interference rate (i.e. packet collision rate) with other common devices (i.e. cell phones, high frequency cordless phones, Wi Fi networks, etc.)
- Abide by FCC regulations where applicable
- Wireless connectivity should have a low rate of power consumption (battery considerations on microcontroller board). Bluetooth chip should have low power mode. Operation in 50 mA range with low power mode in 100 to 200 uA range.

Voltage input between 1.5 and 3 volts. The robots should be able to continue to operate non-stop for a length of time similar to current performance after wireless technology has been added.

- Low cost solution (< $150 per robot)
- Ensure data integrity and security during wireless transmission between two or more devices. Teams should not be able to "eavesdrop" on the commands sent by other teams.
- Leverage (as much as possible) current microcontroller board solution (hardware and software)

The senior project team consists of two members: Andrew Jones and Mike Krofcheck. Both Andrew and Mike will partake in the hardware and software roles of the project, although Andrew will have a larger percentage contribution to the software required for the project. Andrew Jones has been selected to act as the team leader.

Methodology

During the first part of the project, conducted during the Fall '02 semester, the senior project group conducted research into previous work with Bluetooth technology and past attempts at integrating wireless technology onto the microcontroller boards used in EECS 375.

Based on the research conducted by the project group and the experience gained over the last one and a half semesters, it has been determined that the best approach to solving the problem is as follows:

1. Research microcontroller and Bluetooth module architecture.

1.1. Read application notes provided by chip suppliers

1.2. Read trade articles related to Bluetooth and high frequency chip design

1.3. Install and setup microcontroller development environment

2. Integrate Bluetooth onto microcontroller development board to test optimal integration methods.

2.1. Determine hardware design requirements for Bluetooth controller (i.e high frequency concerns and considerations)

2.2. Research Printed Circuit Board (PCB) fabrication companies for high frequency board fabrication

2.3. Research existing high frequency PCB layout designs

2.4. Read trade articles related to Hardware Control Interface (HCI) microcontroller programming

2.5. Write HCI code for microcontroller

2.6. Design PCB for Bluetooth controller

2.7. Build PCB for Bluetooth controller

3. Integrate Bluetooth module with current microcontroller board; writing the necessary software for the microcontroller.

3.1. Remove existing serial connection on microcontroller board utilized in EECS 375

3.2. Connect Bluetooth PCB with serial connection on microcontroller board

3.3. Write software libraries needed for student programs to interact with Bluetooth chip

3.4. Test software libraries needed for Bluetooth connectivity

4. Write supporting software for the computer allowing for student written code to be downloaded to the microcontroller board.

4.1. Learn Cocoa programming language

4.2. Become familiar with Interface Builder and Project Builder development software

4.3. Create Graphical User Interface (GUI) for microcontroller development software

4.4. Create GUI interface to interact with microcontroller board via Bluetooth (send and receive)

5. Redesign the current microcontroller board to include the Bluetooth module, a USB port, the updated microcontroller, and replace out of date parts.
6. Verify the redesigned microcontroller board using simulation software (PSpice).
7. Build redesigned microcontroller board(if time permits)

Current Status

The senior project team has completed the following project tasks since work began on the project in January of 2003:

- Obtained Bluetooth module (Windigo BTM02C2XX-R)
- Obtained Motorola 68HC12 microcontroller development board (M68EVB912B32)
- Installed and setup GNU Development Chain for 68HC11&68HC12
- Installed and setup ExpressPCB board layout software
- Created initial board layout for Bluetooth module (See below)

The GNU Development Chain for 68HC11&68HC12 has been successfully installed and setup under MacOS X version 10.2.4. The GNU Development chain provides a cross compiler for the 68HC11 and 68HC12 microcontrollers that works on any host supported by GNU tools (Solaris, GNU/Linux, FreeBSD, HP/UX, MacOS X, Windows,...). Using standard UNIX tools

like gcc, gdb, and objcopy, C and C++ code can be compiled into the S-code needed by the 68HC11 and 68HC12.

Once the S-code has been created using the GNU cross compiler, a serial terminal application is used to download the S-code to the microcontroller's EEPROM and/or FLASH memory. During development, the EEPROM is used instead of the Flash memory so the microcontroller monitor, D-Bug12 residing in FLASH, can be used to observe the execution of the software under development.

Zterm, a popular serial terminal application under MacOS X, is being utilized by the senior project team to download the S-code to the microcontroller's EEPROM. Figure 1 below shows the Zterm interface while communicating with the M68EVB912B32 development board.

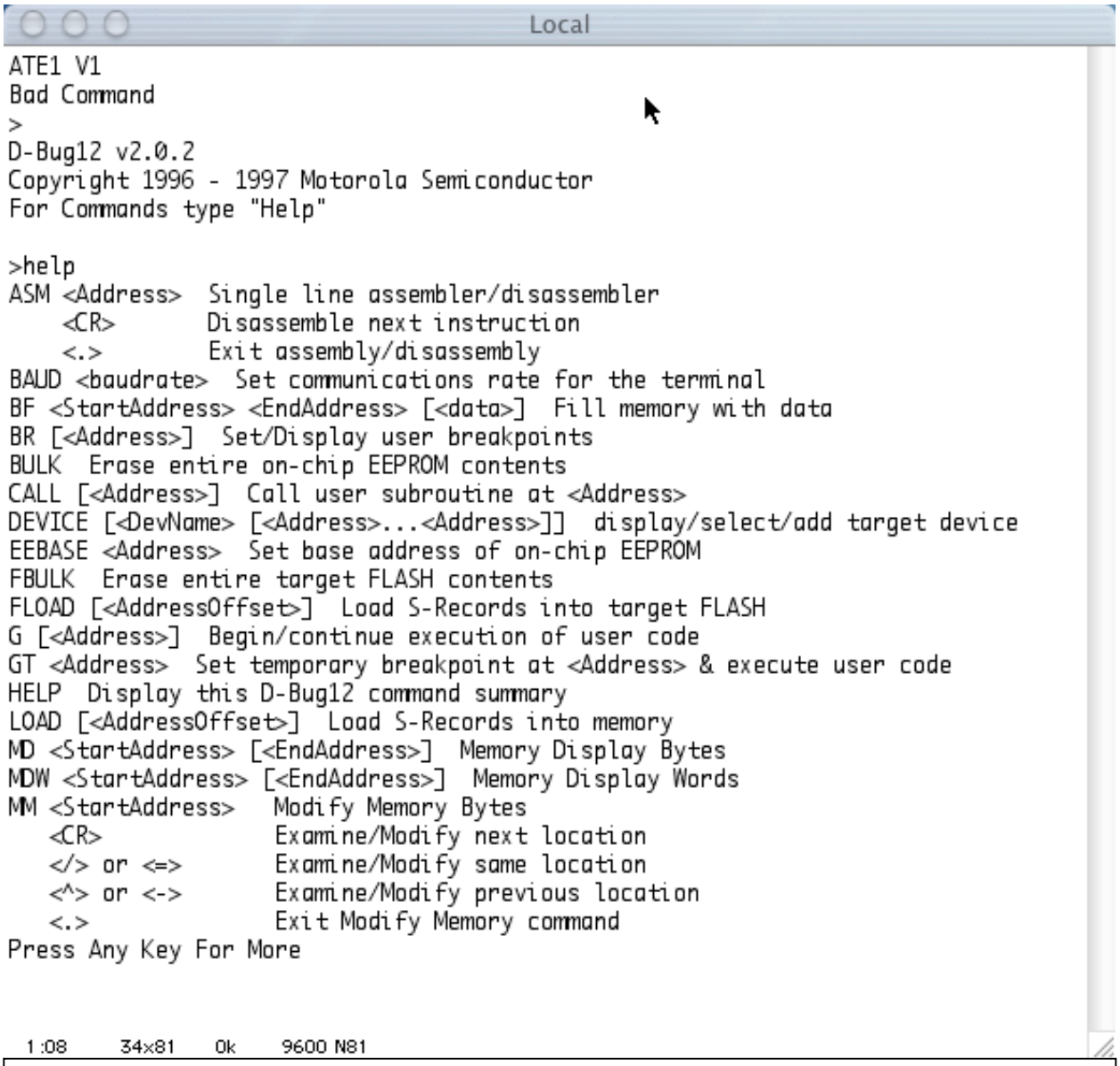


Figure 1: Zterm session window

The next step is the software section of the project involves development of the Hardware Control Interface (HCI) between the microcontroller and Bluetooth module. The HCI will allow the 68HC12 microcontroller to “talk” to the Wintigo Bluetooth module via a serial connection.

Using the Printed Circuit Board software provided by ExpressPCB, the following board layouts have been created for mounting the Bluetooth module (Figure 2 and Figure 3). The module must be mounted on a separate board due to its operation at the 2.4 GHz. frequency. In addition, mounting the Bluetooth module on a separate board will provide the senior project

group with additional flexibility in transferring the Bluetooth module from the 68HC12 development board to the microcontroller board used for EECS 375.

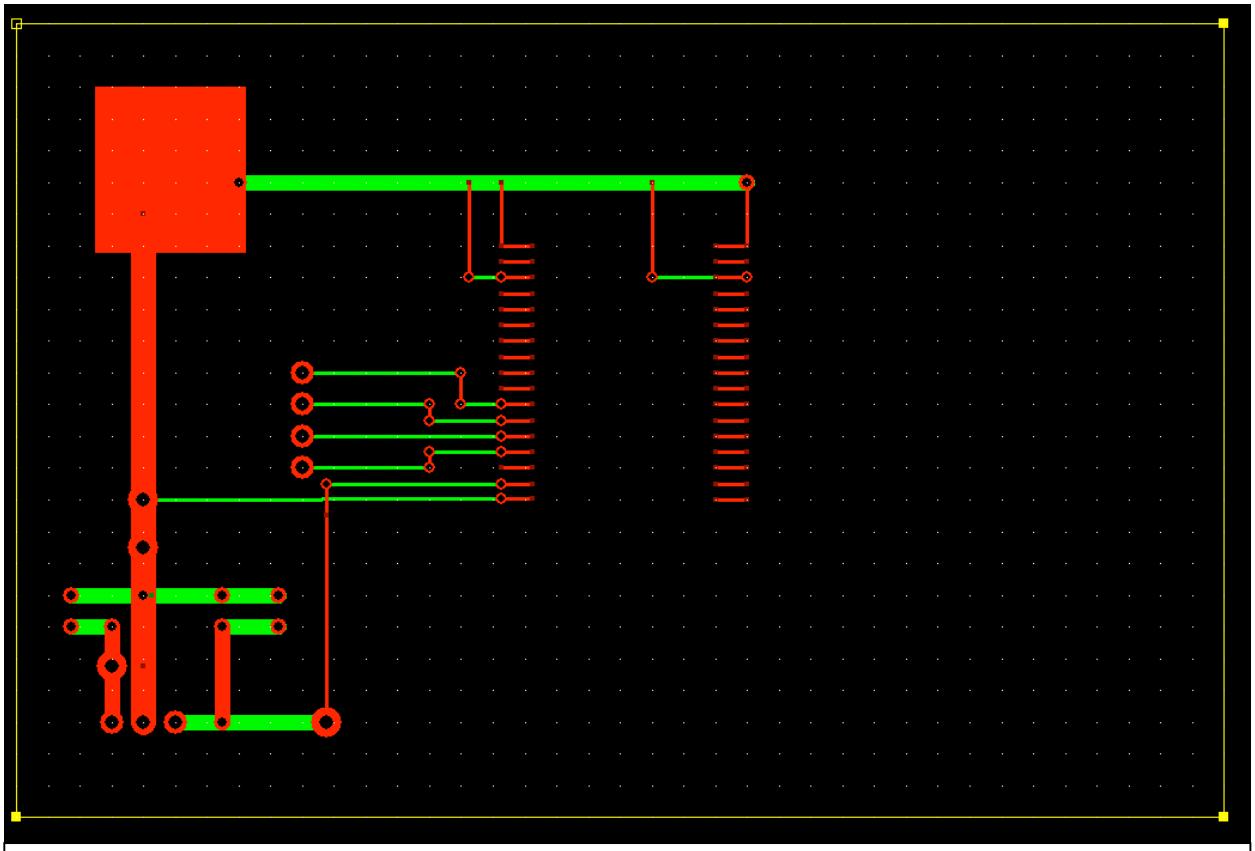


Figure 2: Printed Circuit Board for Bluetooth module (no silkscreen)

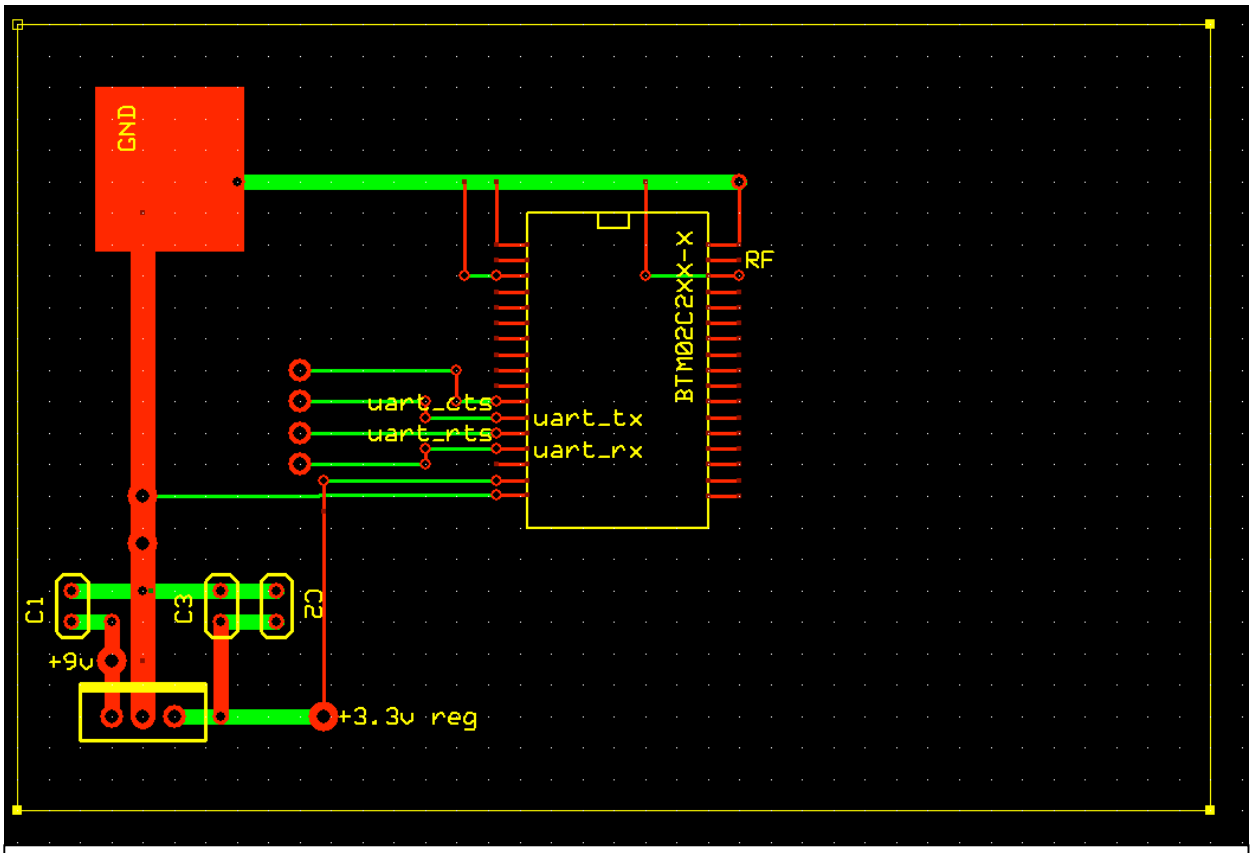


Figure 3: Printed Circuit Board for Bluetooth module (silkscreen)

Following a spot check of the above PCB designs by Dr. Frank Merat and Bob Leskovec, the PCB will be send to ExpressPCB for fabrication. The process, taking up to 1 week, will also involve manually mounting several components on the board.

Once the microcontroller and Bluetooth module are connecting and communicating properly via the HCI, the microcontroller will be able to control the Bluetooth module, sending signals to other Bluetooth modules. At this time, verification testing can begin on the Bluetooth module as it relates to the transmission distance, communication speed, and interference produced by external devices operating at high frequencies.

Implications

The senior project group anticipates that the project will have economic effects, provided the solution meets all design requirements and can be produced as a low cost alternative to similar to products without wireless communications. The microcontroller board will provide extra features not available on other microcontroller board, making it a more attractive product. In addition, the microcontroller board with integrated Bluetooth could appeal to the hobbyist market.

Conclusions

The senior project has met many of the initial milestones setout in the project proposal. We have obtained the necessary parts (Bluetooth module and the Microcontroller development board) and are currently working to integrate both the microcontroller and Bluetooth module. The microcontroller development environment has been setup and is currently being used to create the Hardware Controller Interface (HCI) code necessary to allow for communication between the microcontroller and the Bluetooth module. In addition, a preliminary printed circuit board (PCB) design for the Bluetooth module has been drafted. The circuit board will include pins for serial communication, power, and an external antenna. Upon fabrication of the PCB, the microcontroller and Bluetooth module can be physically integrated.

Modifications

After conducting more work on the project, the senior project team has been able to better define the necessary steps for each major project milestone. The project schedule has been

modified to reflect the new and/or modified project steps (See Appendix B). In addition, time will spent on the project during CWRU's "spring break" to assist in the completion of the project.

The verification methods outlined in the project proposal are still valid and will be implemented at the appropriate project phases. See Appendix A for a reprint of the project verification steps.

Appendix A

Verification

Hardware validation will be performed using several methods. Test LEDs will be placed on prototype board. The status of these LEDs will indicate the status of various signals and chips. Individual chip signals can be checked with an oscilloscope or DMM for integrity and accuracy. The prototype will also have a self-check mode. If a problem exists, the LCD display will indicate the problem. Verification of code, downloaded to the prototype, will be performed by comparing the code in Flash memory with the original code on the host computer.

Software validation will be performed inline with development. This type of verification follows accepted software development practice. This type of testing requires that software is debugged and functioning properly before moving on to the next stage of development. The specific tests that will be run on the software will begin by feeding the application known data inputs, which will give known outputs. This will allow for the software team to verify that the application is processing data correctly. Further testing then will verify the application's robustness to handle skewed input from a user.

Appendix B

Project Plan