Development of a Remote Control Station for iPhone-Controlled, Mechatronics and Robotics Projects

RETs: Matthew Fischer and Vanessa Geraldi
Graduate Student Advisor: Jared A. Frank

Abstract
The goal of our research was to build and explore mechatronics and robotics projects that can be controlled and monitored remotely by the iPhone using the simplest hardware and software setup possible. In order to communicate using these lightweight, portable, graphically superior devices, a wireless router was used to host a Wi-Fi network that the project and the iPhone could connect to. A User Datagram Protocol (UDP) connection with the project could connect, and a Internet Protocol (IP) address could be transmitted to send Open Sound Control (OSC) messages from the iPhone to the semi-autonomous project in order to control or perform some desired task. Previous efforts to control robots in the lab resulted in projects, such as an iRobot with a lego robotic arm, an RC Truck, and a Smart Home, that involved having to physically mount a wireless router and an ethernet-capable microcontroller directly onto the project. In order to reduce the cost, weight, complexity, and other burdens of robotic projects controlled by a remote user from a mobile device, a software and hardware platform was developed for relaying the UDP messages to individual mobile robots or to multiple mobile robots simultaneously, by way of radio communication. Radio Frequency (RF) Transceivers are used to wirelessly communicate iPhone commands to the robot as well as sensor measurements back to a laptop or desktop computer where the data can be processed and displayed using software such as MATLAB.

A Parallax BOE-Bot was used as a base for testing. The BOE-Bot, too small to support a router and other large hardware, was driven using an iPhone via this control station, and the success led to ideas for using the station to distribute commands to, and gather data from, multiple robots without the need for expensive Wi-Fi adapters on each robot. This research involved all of the principles of robotic systems, including mechanical and electrical subsystem design, software and hardware interfacing, as well as C and BASIC programming.

Results and Conclusions
This has led to the idea for using the station to remotely gather information (via radio frequency) from a group of mobile robots in order to perform some complex control algorithm, one that would be difficult or impossible using the hardware and software of the miniature robots, that will ___ the whole group. Using the platform developed in the lab, more sophisticated forms of robotic control and data acquisition may be possible using single or groups of robots that are unusually compact and programmatically simple. If the station is plugged into the internet, the entire process may be commanded from a TCP/UDP-capable cellphone application located anywhere in the world.

Future Work
• Removed the Router/MAKE from the project
• Discovered communication with MATLAB/Simulink
• For interfacing with lab equipment/experiment
• Designed and built DAC platform
• For interfacing any RF mechatronic/robotic project

Hardware
• The iPod or iPhone communicates using the Open Sound Control (OSC) Protocol. Then, a MAKE controller utilizes a hardware implementation of Open Sound Control (OSC) over User Datagram Protocol (UDP) ports to communicate remotely to the iPod or iPhone.
• 802.11n Wireless Router
• Make Controller an input/output controller
• 802.15.4B Microcontroller equipped with servos for navigation, photoresistor sensor, RF transceivers
• One BASIC Stamp 2 (BS2) was used as an intermediate device to serially receive and transmit commands from the MAKE controller and then to transmit these messages to a second robotic BS2. The robotic BS2 was developed to navigate, take sensor readings, and transmit data back to the original BS2. Finally, this information was relayed to MATLAB.

Software
• PBASIC: Developed by Parallax, a microcontroller based version of the BASIC language. Once code is written it is tokenized and loaded into the microcontroller EEPROM. Commands such as PULSOUT, HIGH, LOW, DEBUG, and FREQOUT are native to PBASIC and used for special purposes.
• MAKE Controller: manufacturer-installed default firmware in C language.
• MATLAB: Powerful, widely used engineering software with data acquisition, processing, computation, and control applications

Simulink Diagram/ MATLAB Program

Acknowledgements
This research is supported by the National Science Foundation under NSF Grant #: EEC-0807286