Vehicle Control Through an IEEE 802.15.4 Mesh Network

Progress Report

Peter Fyon 100652096
John (Bong Geon) Koh 100684909
Andrew Kusz 100685317

Supervisor: Dr. Aitken

Dec 8, 2009

Summary

By John Koh
Collaboratively edited by: Andrew Kusz, Peter Fyon

This report is in regards to the progress of the development of the IEEE 802.15.4 mesh network project. The purpose of this project is to control a commercially available remote controlled (RC) car that has been modified to receive commands over the wireless mesh network.

As of this report, several project goals and deadlines have been met. Initially, all hardware and software components necessary for the project were researched and analyzed. An initial design of the system was then created with all the hardware and software limitations in mind. After establishing a baseline design for the project, many of the hardware milestones outlined in the project proposal were met, as well as the accompanying software requirements. Overall progress of the project has been good, despite a few delays due to unforeseen hardware incompatibilities.

There are no significant changes that need to be made to our deadline schedule.

Introduction

By John Koh
Collaboratively edited by: Peter Fyon, Andrew Kusz

This document is a progress report on the wireless mesh network project performed by Peter Fyon, John Koh, and Andrew Kusz. The project is under the supervision of Dr. Aitken. This report documents the work accomplished over the three month period from September 16th, 2009 to December 7th, 2009. This marks the approximately halfway checkpoint for the project until the project end date of April 7th, 2010.

The project objective is ultimately to drive a remote controlled car by sending commands through an IEEE 802.15.4 mesh network from a computer or other command interface. This network increases the vehicle's operational range without exponentially increasing power consumption as required by a point-to-point RF transmission.

Within this three month time period, we were able to accomplish our milestones outlined in our project proposal, with the exception of the goals reliant on hardware we currently do not have. Additionally, while the reports on our hardware have been started, they cannot be completed until full implementation and testing of each component. Knowing this, we have been working on milestones intended to be completed later on in the project.

Progress

By Andrew Kusz, Peter Fyon, John Koh
Collaboratively edited by: Andrew Kusz, Peter Fyon, John Koh

Project progress is categorized as follows.

Group Progress

Research and analysis of the hardware and software required for the project began immediately from the start of the project. To understand the operation of RC cars better, two RC cars were reverse engineered. Upon analysing two different types of RC cars, an RC car design suitable for project was chosen. For the mesh network, the XBee 802.15.4 RF module was selected for the node hardware since it supported all the functions set out to do in our proposal. For the controller and RC car, the Handy Board microcontroller (MC) system was used as it was readily available and would be capable of driving the vehicles as well as sending/receiving commands over UART.

Peter Fyon

After the group deconstructed our skid-steer car, I documented the major components of the control circuitry for the skid-steer car. I also made the initial block diagram design of our mesh networking system.

Throughout the project, I have written example code (drive motors, get digital inputs, etc) in Interactive C so creating the software components should be simply a matter of modifying the example code. After determining that the XBee modules communicate over UART, I began researching ways to integrate assembly code into Interactive C. Through a combination of multiple sources, I determined it was possible, and found a guide written by a previous 4th year project at Carleton that detailed the steps to compile assembly code for use on the handyboards. I then wrote asm routines to handle data received over the UART and allow access to the ring buffer from Interactive C.

I also maintain the project website and set up the subversion repository to store all our code.

I am in the process of writing a report on the car teardown, and testing and debugging the UART asm routines.

John Koh

After the researching about RC car operation, the car was modified to fit a servo instead of a motor for the front steering of the car. Research on UART controllers (the communication method between the XBee nodes and the Handy Boards) was done. After deciding on the controller hardware, the design of the remote control hardware and software was started.

Andrew Kusz

Analyzing our first RC car, we realized that it relied on “skid steering” which is very inconsistent as steering depended on the friction between the wheels and the surface. It was decided to purchase cars that had a separate motor for moving forward and back and one for turning. Once the new cars were purchased, one car was stripped down and analyzed.

The original XBee nodes we were provided were set up and we were able to have the two XBee nodes connect to each other with one being a Co-ordinator and the other an End-Device using a

loop back connector on the End-Device to echo back messages sent from the Co-ordinator. Unfortunately we soon realized that these XBee nodes were much older revisions and did not actually support the 802.15.4 standard and were instead just 2.4GHz point-to-point RF devices and unable to form mesh networks. Research was done online and new chips were found that would do the job we required and were subsequently ordered.

Conclusion

While the project progress has been delayed due to unforeseen hardware limitations, overall progress has been good. Having outdated XBee nodes did not allow the project to progress further for the node hardware and software design. Adjustments have been made to the schedule but the goals of the project have not changed.

Schedule

By Peter Fyon
Collaboratively edited by: John Koh, Andrew Kusz

Name	Begin date	End date
Reverse Engineer Car (group)	9/21/09	10/5/09
Report on Car Hardware	10/1/09	10/7/09
Research Required Hardware	10/5/09	10/26/09
Design Node Hardware	10/26/09	11/2/09
Design Node Software	11/2/09	11/16/09
Test/Revise Node Hardware/Software	11/16/09	11/23/09
Report on Node Hardware	11/23/09	11/25/09
Design Car Control Circuit	10/26/09	11/9/09
Design Car Control Circuit Software	11/9/09	11/23/09
Test/Revise Car Control Circuit Hardware/Software	11/23/09	11/30/09
Report on Car Control Circuit Hardware	11/30/09	12/7/09
Report on Car Control Circuit Software	11/30/09	12/7/09
Design Remote Control Hardware	12/7/09	1/4/10
Design Remote Control Software	1/4/10	1/18/10
Test/Revise Remote Control	1/18/10	1/25/10
Report on Remote Control	1/25/10	2/1/10
Perform Experiment	2/1/10	3/5/10
Progress Report	11/27/09	12/8/09
Prepare for Oral Presentation	1/10/10	1/17/10
Prepare for Poster Fair	3/5/10	3/19/10
Prepare Final Report	3/10/10	3/24/10
Edit Final Report	3/24/10	4/7/10