Vehicle Control Through an IEEE 802.15.4 Mesh Network

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

Supervisor: Dr. Aitken

Sep 22, 2009

Introduction

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

This document proposes a project to design and implement a vehicle controlled remotely through a mesh network by a human operator. This project will commence September 16th, 2009, and complete with the submission of a final report on April 7th, 2010. This project will be completed by Peter Fyon, John (Bong Geon) Koh, and Andrew Kusz with supervision from Dr. Victor Aitken, and additional support from Mr. Eatherley.

Objective

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

Our 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. We will achieve the following goals:

1. To design and manufacture a node capable of mesh networking using the IEEE 802.15.4 standard.
2. Design custom hardware and software to drive a commercially available remote controlled vehicle using commands received over the mesh network.
3. Design a custom transmitter to send commands across the mesh network.
4. Demonstrate the greater range of the remote controlled car by using the mesh network.
5. Demonstrate the resilience and self healing ability of a mesh network.

The proposed project will provide a solution to the problems inherent with boosting the power of a transceiver to increase a remote controlled vehicle's range.

Motivation

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

Wireless technology allows us to communicate over various distances without the need to physically connect both ends of the connection. This allows us to communicate across oceans, over canyons, or just throughout a building, but it does have its shortcomings. The range of a transmitter is limited by its power source. Along with increased power requirements, it is difficult to control where the signal reaches. Unlike a wired network, the wireless signal is broadcast over an entire area between the source and the max range meaning that anyone in between can potentially intercept the signal. There is also the higher potential of signal noise affecting the signal when sent over larger distances, as well as our signal interfering with other electronics.

With this project, we aspire to show how an IEEE 802.15.4 mesh network can solve many of these problems. The mesh network is fairly low power, with a node lasting up to 2 years

on a single battery. Because each node only broadcasts over a small area, we have much more control over the coverage of the network which lowers the chances of our signal being intercepted. The low power nature of the network also ensures that the chance of interfering with other electronics is minimal and because the signal is being relayed between nodes, it ensures the signal stays intact.

Technical Overview

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

In order to demonstrate the IEEE 802.15.4 mesh network, the project will require a modified RC car (and its remote control), and at least four IEEE 802.15.4 router nodes to show non-linear healing over the network.

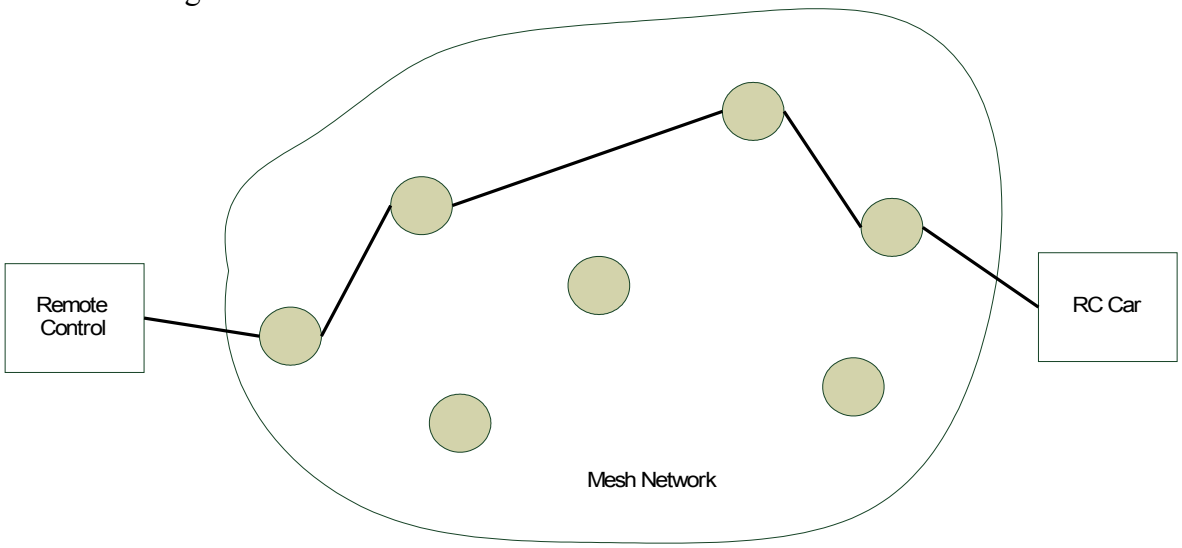


Figure 1: IEEE 802.15.4 Mesh Network With All Nodes Working

In the figure 1, the IEEE 802.15.4 nodes are all functional and routing between each other, forming the operational network for the RC car. If the RC car is a node's range and the remote control is near any operational node, the mesh network can establish a connection between nodes and send the signal to the RC car. The signal may travel from any combination of nodes networked together.

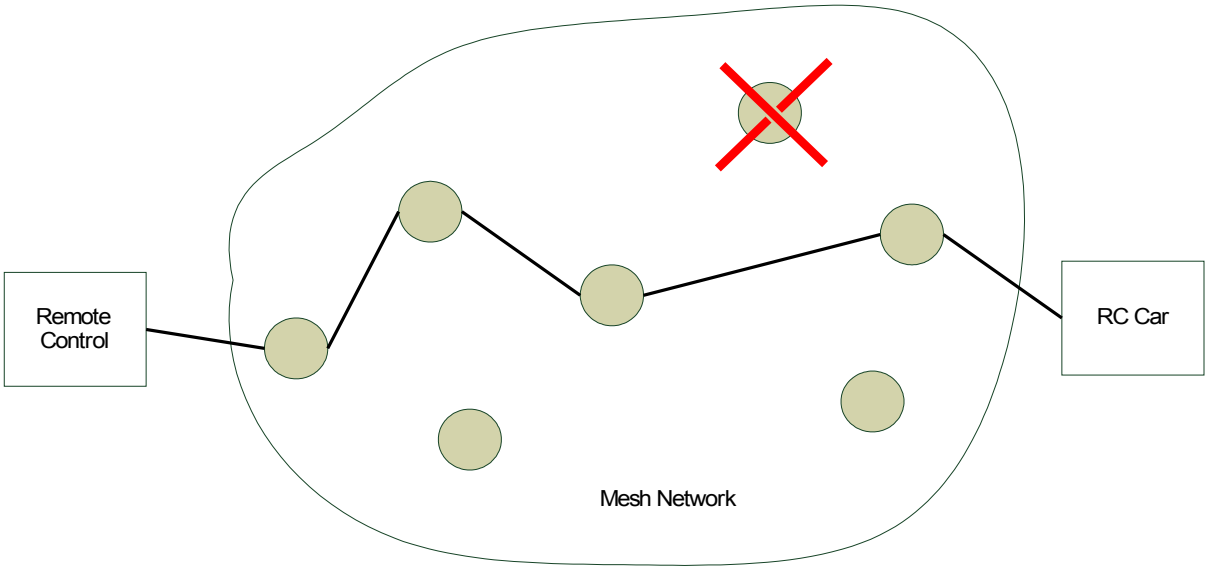


Figure 2: IEEE 802.15.4 Mesh Network With One Less Node

When a node is removed from the network, the network's range will decrease but the RC car will still be functional within the existing range. Using the previous example, an RC car near a node can still be controlled if the remote control is near any operational node. When the path is disrupted, the network will try to establish a new path to the RC car.

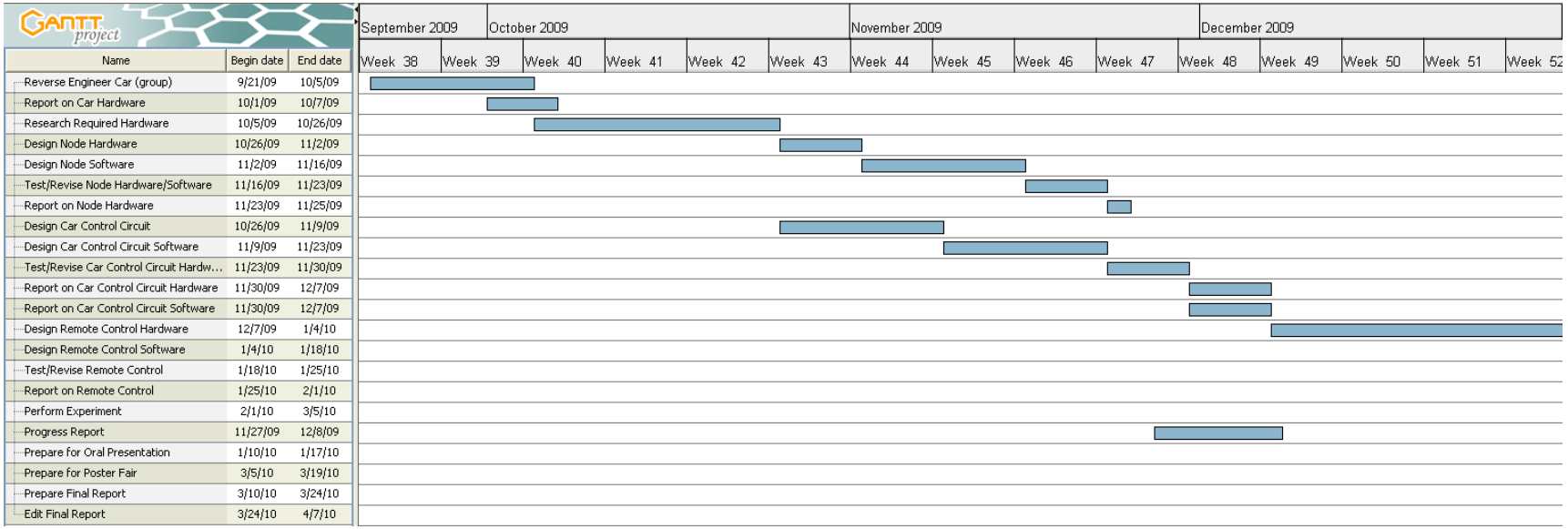
For this project, we plan on removing the internal electrical components of an existing RC car and replacing them with a microcontroller that will work over the IEEE 802.15.4 operating frequency. This process will require reverse engineering of the original hardware to understand how the motors on the RC car are controlled.

Once the RC car is reverse engineered, the RC car and the IEEE 802.15.4 nodes will be designed to accept RC car controls and forward from any node on the network.

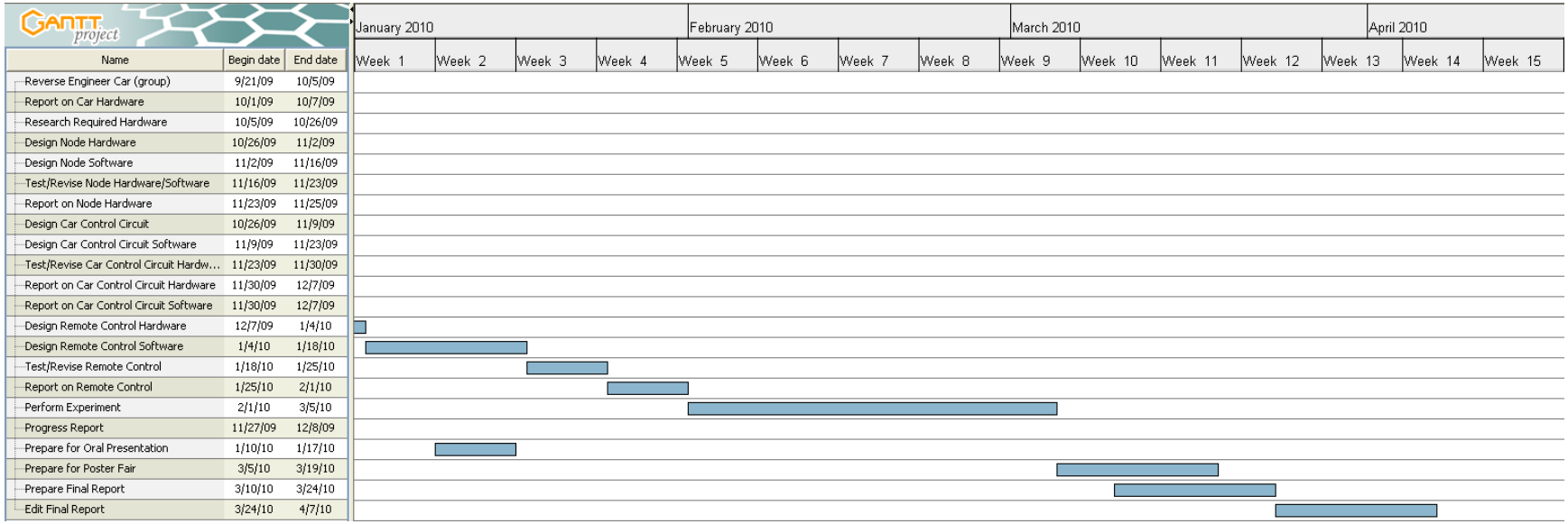
Schedule

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



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 Hardw...	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



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 Hardw...	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