

# **CSCI 197 Final Project Proposal**

*By Michael Gaiman & Christopher Toombs*

---

## **Introduction**

We propose to build a remote control car with remote. The car will be built from a Lego Mindstorm kit. Communication will operate over the IEEE 802.11g wireless standard.

## **Vision**

The car will be constructed from a Lego Mindstorm kit. It will include a wireless access point ("WAP") which will receive commands from the remote control. These commands will be translated into specific signals which shall control the movement of the vehicle.

The remote will be constructed from some system which contains components to allow at least 3 distinct operations (left turn, right turn, forward). The signals from these controls will be translated into specific commands which will be communicated wirelessly to the car.

A protocol will be defined which will provide explicit commands through which the remote will communicate with the car. The protocol will contain at least 3 commands (left turn, right turn, drive forward). The underlying protocols have yet to be determined. One option is to implement the protocol at layer 5, using UDP at layer 4. Another option is to implement this protocol at layer 3, directly on top of 802.11g at layer 2.

## **Functional Requirements**

- Depression of the "left-turn" actuator on the remote control will cause the front wheels to both turn at least 45 degrees to the left.
- Depression of the "right-turn" actuator on the remote control will cause the front wheels to both turn at least 45 degrees to the right.
- Depression of the "drive forward" actuator on the remote control will cause the back wheels to forward-rotate at least 1 full rotation per second.

## **Non-functional Requirements**

- The car shall respond to actuation within 1 second of actuator depression.
- No more than 5 second power-up for the car.
- No more than 5 second power-up for the remote control
- The car shall run for at least 4 hours on four AA batteries
- No additional security will be included in the wireless communication

## Hardware Architecture

The hardware for this project will consist of two units, a remote control car, and a remote control, which communicate over an 802.11g communication link, as shown in Figure 1.

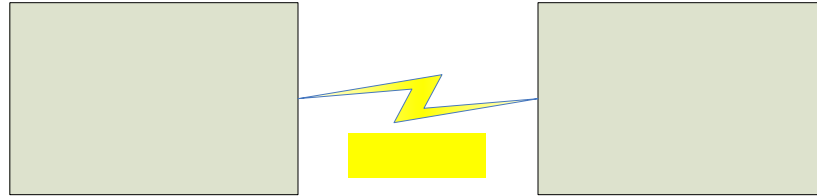


Figure 1 - Overall hardware architecture

The remote control unit has its own architecture, shown in Figure 3. It consists of a number of buttons through which a user interacts with the system.

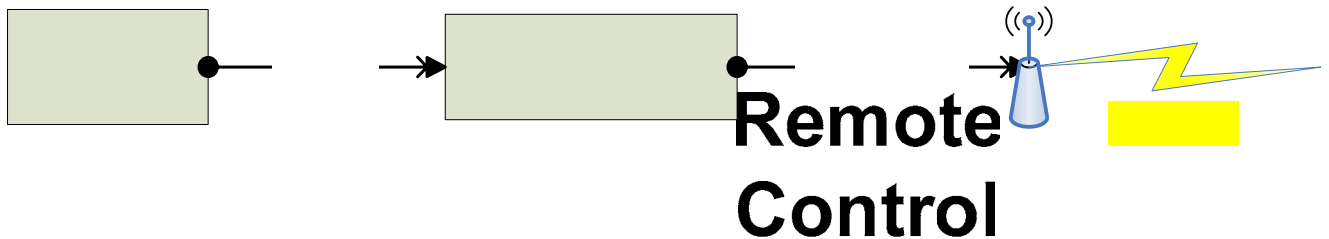


Figure 2 - Remote control hardware architecture

802.

The car unit has its own architecture, shown in Figure 3. The flow of data through these components begins when the WAP receives commands through an 802.11g signal. This command is communicated directly to the microcontroller. The microcontroller converts the command to a signal which appropriate activates the wheels of the car.

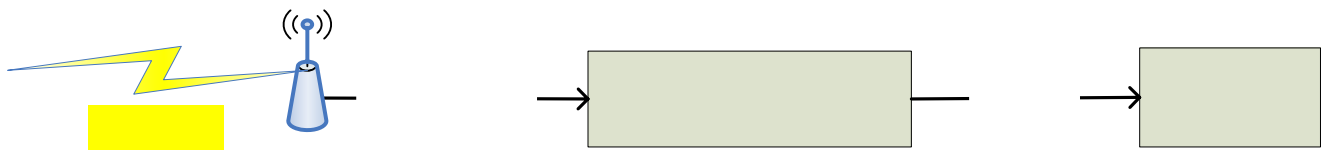


Figure 3 - Car hardware architecture.

## Software Architecture

Similar to the hardware architecture, each component of the system has its own software architecture. First, as shown in Figure 4, the remote control unit consists of three software modules. The Signal Capture module receives the signal from the actuation of buttons on the unit. This signal is passed to the Signal-To-Command Translator module, which will convert the signal to the appropriate command as defined in the

Actuators

Signal

Microc

communication protocol. Finally, this command is sent via the Command Transmission module, which is included in the WAP hardware and will not have to be independently implemented.



Figure 4 - Remote control software architecture.

The car unit has its own software architecture, which is shown in Figure 5. The command is received through the Command Receiver module. This module is included with the WAP hardware and will not have to be implemented. The command is next received by the Command-to-Signal Translator, which produces the appropriate signal. This signal is then sent to the Signal Output module where it will be transmitted to the wheels of the vehicle.

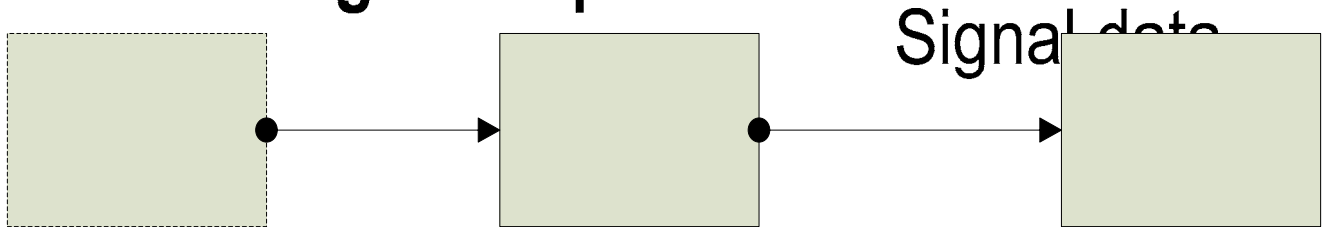


Figure 5 - Car software architecture.

### Actuators

The controls for this system are located entirely on the remote control unit. This unit will include three buttons, SW1, SW2, and SW3. The manner in which the buttons are depressed will control the motion of the car unit. The following list describes the operation executed upon the depression of the noted button:

- SW1: Wheels turn left
- SW3: Wheels turn right
- SW2: Car drives forward

The depression of the particular buttons produces a signal which can be interpreted by the microcontroller. The ports on which these signal are received will be polled by the microcontroller in order to determine the desired command.

### Communications

Command  
Receiver

Com

Communication between the two hardware units is achieved through an IEEE 802.11g communication link. Commands will be transmitted according to a specific but as yet undefined protocol. Each command will correspond to a combination of operations. The protocol will include the following specific commands:

- Drive forward
- Drive forward: Wheels turned left
- Drive forward: Wheels turned right

### **Hardware Sources**

The remote control shall be constructed using the Zilog Z8 Encore! board using the 6403 CPU. This board has been provided by The George Washington University through its CSCI 197 class during the Spring 2005 semester. We have not determined a satisfactory WAP yet which interfaces well with the Z8 Encore! Board. Similarly, we have not found a WAP which interfaces well with the Lego Mindstorms kit. The Lego Mindstorms kit will be donated for the project by Michael Gaiman under the condition that the entire set is returned to him after an instructor evaluation of the final product.