91.301 Organization of Programming Languages

OPLspr14 /

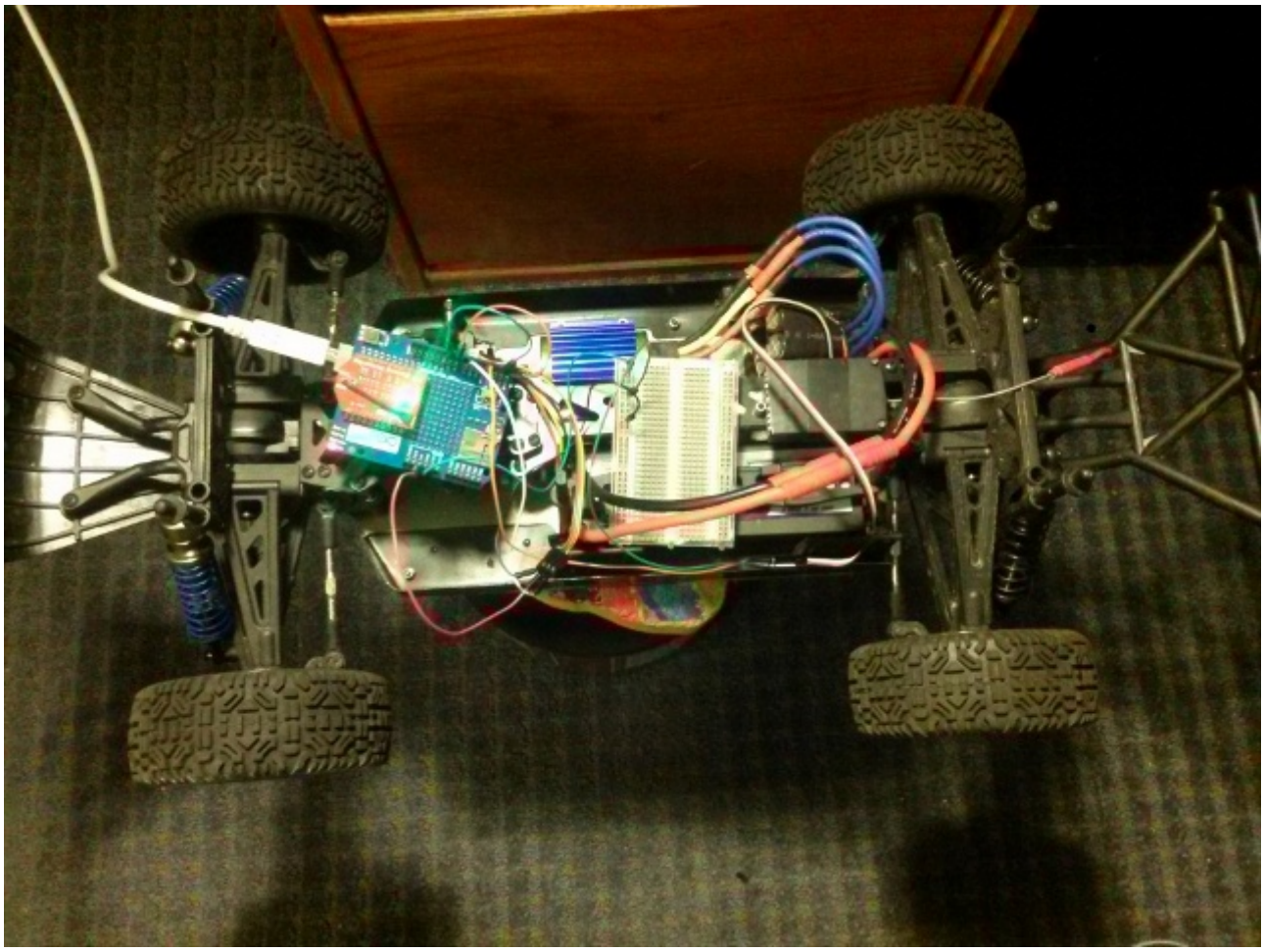
WirelessRCCar

Jheddy Melendez
May 2, 2014

Overview

The goal of my project is to use an arduino UNO microcontroller to control a hobby RC car over Wireless issuing voice commands. A racket program will be running on my computer that takes as input the output from a speech recognition program and converts it to valid commands that get sent to the Arduino over wireless through TCP and then the arduino uses those commands to control the car by sending the corresponding PWM signals.

Screenshot



Concepts Demonstrated

- *Map and filter* used to filter out unrecognized commands and map a function that turns voice commands into integer values.

- *Cond* used to replace nested if statements.
- *recursive Recursion* used in main loop to keep reading input.
- *set!* used to modify state.
- *string & list* manipulation.
- *lambda functions* used to map over a list.
- *equality check(eq?,equal?,etc...)* used to make sure word recognized is valid.
- *Racket IO ports(PS5)* reading and writing IO ports from the tcp connection and from the subprocess created to run the speech recognition program within racket.

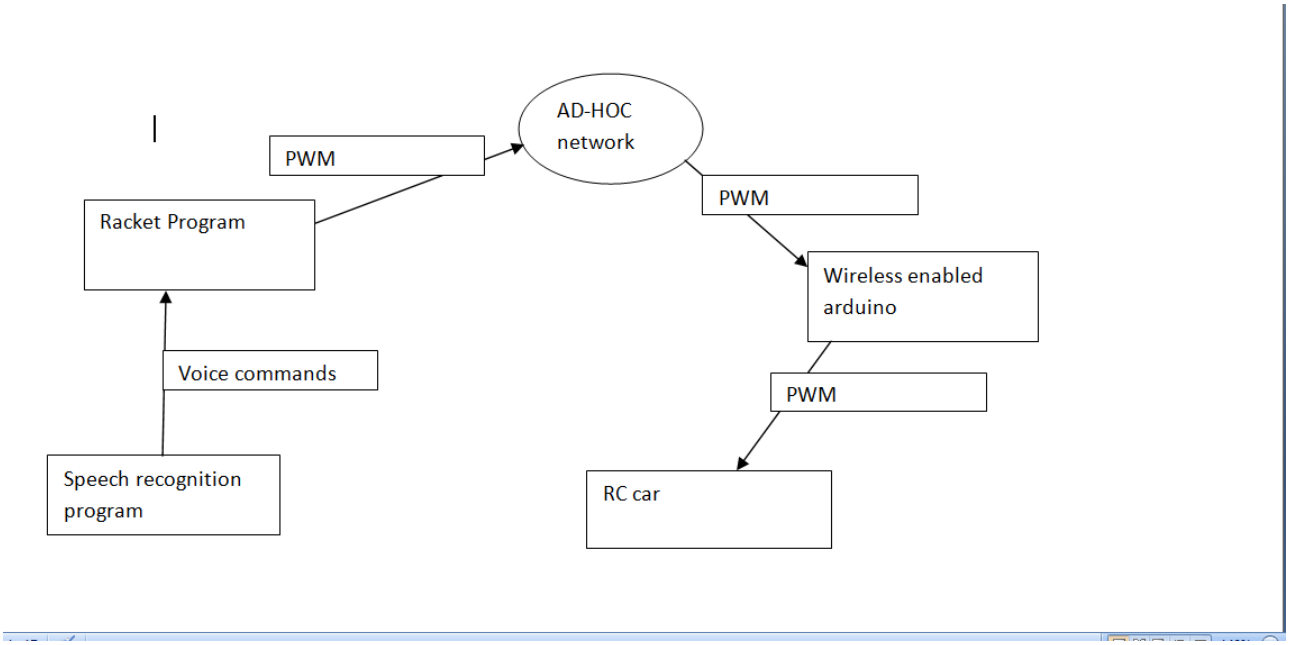
External Technology

One of the external technologies I used is the tcp library in racket, this helps me create a tcp connection to the micro-controller over wireless and send it commands. The other technology I used is a micro-controller(arduino) to implement the other half of the project together with a wireless module connected to it to enable it to communicate with my computer over wireless. Micro-controllers are basically like mini computers with inputs and outputs that can run programs that can interact with the outside world trhough sensors motors etc....

Innovation

The Innovation in my project is being able to control an RC car with just speech, which is an unusual way to drive an RC car and is not usually done since RC cars are usually driven using a remote control.

Technology Used Block Diagram



Additional Remarks

While testing I found that the slowest speed I could make the RC car go is actually really fast(The Speed Controller probably needs to be calibrated for PWM signals because it usually receives PPM signals from the transmitter not PWM), so I don't think I will demonstrate the forward. reverse and stop commands because I don't want the car to go out flying and crash.

Since the slowest speed that I can make the car go is actually really fast when I make it go forward with a voice command the voice recognition program stops recognizing words because of the loud noise, this could be fixed by maybe using noise robust models for the speech recognition program.

Page last modified on May 02, 2014, at 10:23 AM