SDR Passionate Pursuit

Anusha Datar
Olin College of Engineering | Fall 2017

Learning Goals

My goals over the course of this passionate pursuit were to:
- Learn about signal processing and radio transmission
- Improve my software development skills in C for the radio portion
- Improve my Python skills to make an easy-to-use API for the RC car
- Obtain a HAM radio general license

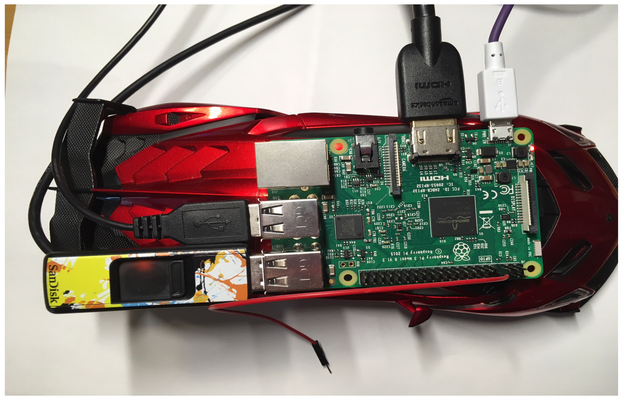

Figure 1: The Raspberry Pi I used for my project as well as the RC car I decided to control.

General Timeline

- **September 14 - October 15:** Became familiar with the C programming language and RC signals through books like *Modern C* and *Software Defined Radio* and online resources
- **October 15 - November 1:** Acquired Raspberry Pi and learned about its internal hardware through online resources and experimentation. Started implementing radio controller.
- **November 1 - November 20:** Studied for and acquired HAM radio technician and general class licenses.
- **November 20 - December 5:** Worked on implementing radio controller for the car in C. Tried several implementations, including sending clock signals to a pin and using DMA, and I decided DMA was better.
- **December 5 - December 12:** Wrote python program for sending different signals out quickly and calibrated for remote-controlled car that I had purchased.
- **December 12 - December 18:** Prepared poster, demonstration, and presentation for expo.

Summary and Final Deliverables

Generally, transmitting radio signals using a Raspberry Pi, a small and low-cost computer, requires the purchase of additonal hardware. For my passionate pursuit, I decided to learn about radios, embedded programming, and higher-level programming to make a software-defined radio controller for the Raspberry Pi that could be used to control a remote controlled car. Technically, to send signals to my car, I needed a general class HAM radio license, so I also studied for that exam and got that certifcation over the course of the semester. As a final deliverable, I have my general class license and a demonstration with a remote-controlled car.

Results: HAM Radio License

After spending the first half of the smester reading and ideating about creating the transmitter for the car, I had also read a great deal of information about radios in general. Therefore, after spending a few hours going through the specific laws about transmitting, I was able to pass the technician and general class HAM license exams, which gave me access to the frequencies needed for the RC car I was hoping to control.

Call Sign	KC1IQU
Status	Active
Dates	
Grant	12/04/2017
Effective	12/04/2017
Licensee Information	
FRN	0027023431
Licensee Name	
Datar, Anusha 11Kingsdale Street Burlington, MA 01803	
Amateur Data	
Operator Class	General
Group	D
Eligibility Code	

Figure 2: My call sign and information as listed in the public FCC Directory.

Results: Raspberry Pi Transmitter

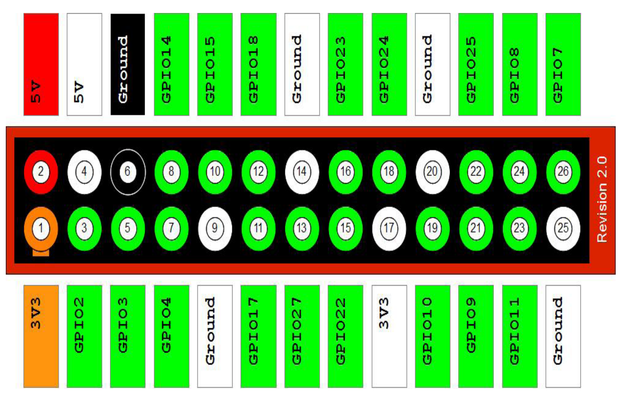

Figure 3: Raspberry Pi Pinout, as taken from official documentation. GPIO pin 4 was used in this project.

The programs here had two main components: the C program that made transmitting signals possible, and the python code that made it easier to pass a variety of signals to the car (so that the car could have a teleoperated mode with relative ease) and develop additional software more easily overall. To transmit signals, I used the DMA controller to output data from a GPIO pin in real time, and I used a wire as an antenna..

RC Vehicle Interface

To reliably test this project, I decided to use a standard remote-controlled car. After building a basic transmitter, I created some Python programs that made it relatively easy to quickly test various frequencies and configurations so I could find the patterns needed for each direction the car traveled in. After doing that and making the transmitter the best I could, I created an interactive control program and autonomous driving program with a commands for each direction so that the car could essentially pilot itself. Building this component helped me improve my Python software development skills and added a fun, interactive, and legal way for me to see the results of my code.

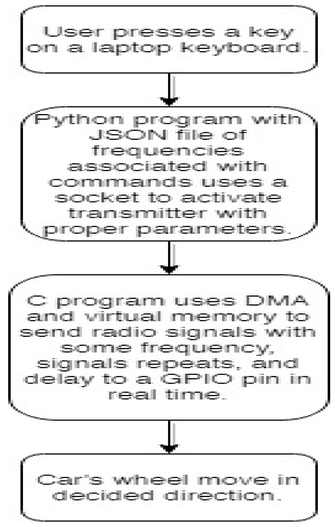

Figure 4: RC car workflow

Conclusions and Extensions

Sending signals through a Raspberry Pi is very doable, but my solution is neither complete nor satisfying. I would love to continue working on this project in two dimensions: with the radio and with the car. With the radio, I could work on improving transmission quality: it would be neat to build a raspberry pi hamshack completely in software. With the car, I could add additional sensors and make a small autonomous vehicle that could serve some direct practical purpose or act as an affordable, easy to create teaching and learning platform.