

HARVEY MUDD COLLEGE

ENGINEERING 155

MICROPROCESSOR SYSTEMS: DESIGN AND APPLICATION

Internet-Controlled AGV

Authors:

Aaron ROSEN

Alex RICH

Professors:

David HARRIS

Matthew SPENCER

December 11, 2015

Abstract

1 Introduction

The Internet-Controlled Autonomous Ground Vehicle is a treaded chassis that holds an FPGA with bluetooth listening capabilities. A companion website allows a user to interact with the vehicle by commanding it to travel it to various places. The motivation behind this project was to build a tank that could navigate within a map and shoot NERF darts. By building a tank that is remotely controlled over the internet, the more difficult part of this concept is realized.

The project involved an FPGA

2 New Hardware

2.1 BlueSMiRF

2.2 Vehicle

3 Schematics

Alex

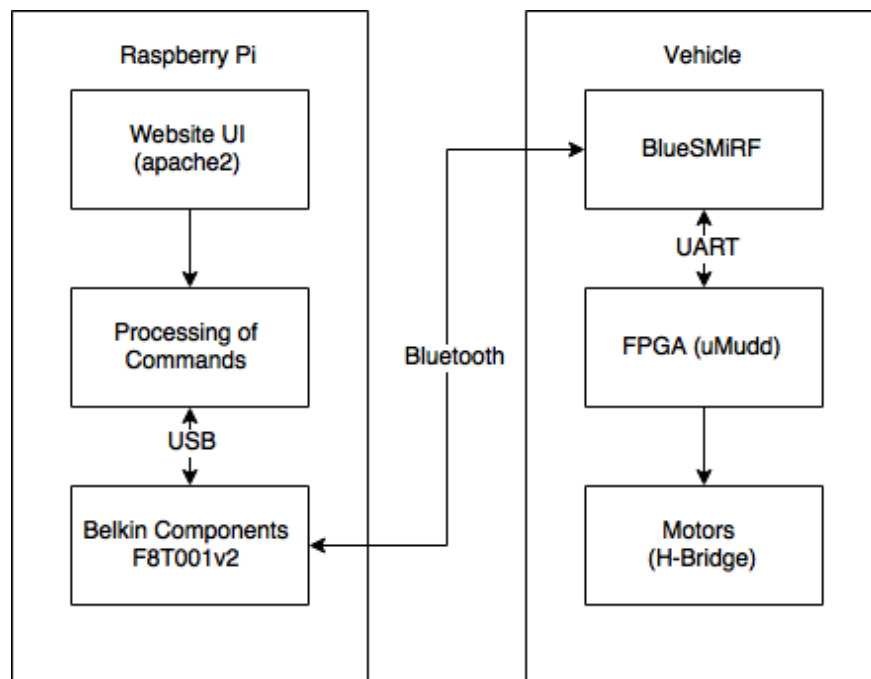


Figure 1: An overview of the system. The system is comprised of two major subsystems: the Raspberry Pi 2 controller and the vehicle, which is controlled by the μ Mudd board.

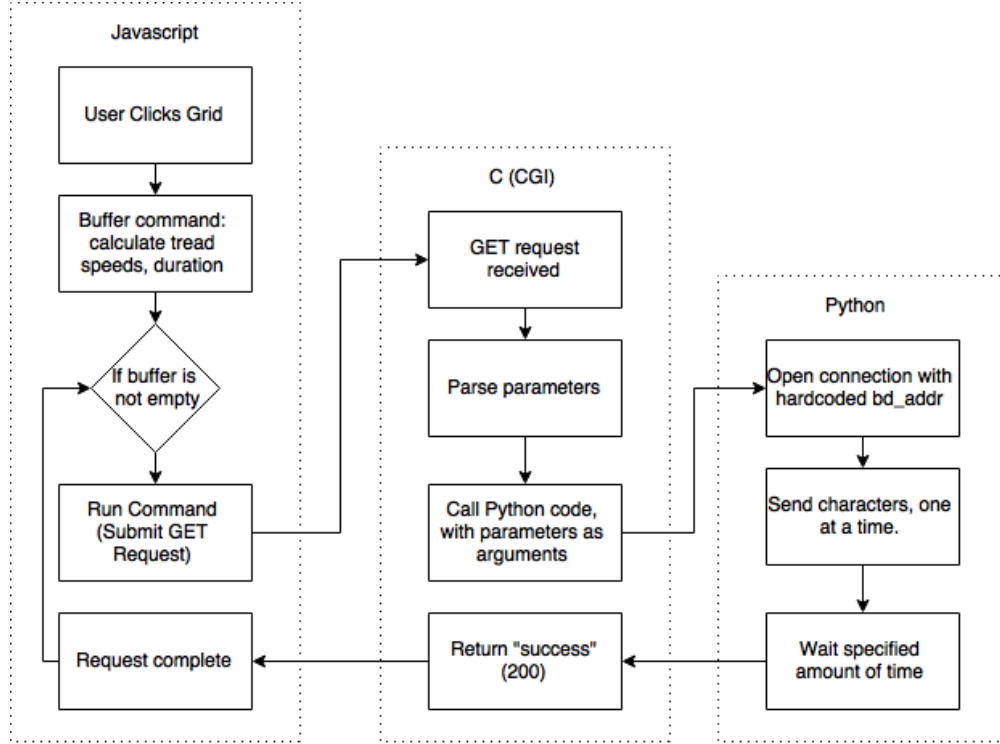


Figure 2: The flow of control and data through the Pi.

4 Raspberry Pi

The Raspberry Pi 2 serves several functions. It hosts an Apache2 website accessible via the internet and contains code to submit commands to a bluetooth dongle. These two functions are integrated such that a user interacts with the website, indirectly sending commands over bluetooth. Figure 2 show the flow of data and control through the Pi.

4.1 Website

The website contains a visual user interface that contains instructions for use, a grid on which to input locations for the robot to maneuver to, a list of the commands currently buffered for sending. The code for the website is shown in Appendix A. The website was built using HTML, JavaScript, and Bootstrap CSS.

The Pi hosts the website. Upon clicking in a grid space, the page's JavaScript calculates the left and right tread speeds and duration of movement required to get the tank to move from its original position to the new position. The current algorithm involves first moving north/south, turning, moving east/west, and finally turning to reorient itself vertically. Once the commands are generated, the JavaScript makes an HTTP GET request to the inputChars resource of the Pi. When the request is completed, the page updates, either submitting the next command in the buffer or waiting for another input from the user.

4.2 Python/C

After receiving a GET request, the common gateway interface (CGI) reads the input parameters (three integers) and converts them into a format that the vehicle understands. This involves converting the numbers to sign/magnitude bit representations. Because of how the UART works, the C also reverses the bits.

calls a Python script. The Python script utilizes the `bluetooth` module to allow sending data using the bluetooth dongle. Since the robot has a constant bluetooth device address, this address is hard coded into the Python script. When called, the python script sends the commands, one at a time, to the robot. Currently, the system sleeps for an amount of time to theoretically give the vehicle enough time to move. However, in the future, the Pi will wait for acknowledgement from the FPGA. The code on the Pi is shown in Appendix B.

// Depending on room, I can include the code used to call Python, since that seemed to be a common problem.

5 FPGA Design

Aaron

6 Results

7 References

[1] REFERENCES?

8 Parts List

This is the bill of materials for the project.

Item	Description	Source	Cost
Tracked Vehicle Chassis Kit	Chassis for the tank, includes treads and frame.	Amazon	15.39
Tamiya 70168 Double Gearbox	Gives tank flexibility to turn by controlling each tread independently.	Amazon	11.99
μ Mudd Board	Controls the vehicle.	E155	0.00
Raspberry Pi 2	Provides website interface and sends commands to vehicle.	E155	0.00
BlueSMiRF	Wirelessly communicate via Bluetooth between Pi and μ Mudd board.	E155	0.00
Belkin Components FT8001	Send bluetooth data from Pi.	E155	0.00
2X TrustFire 14500	Li Ion battery, 3.7 V	Aaron Rosen	0.00
			27.38

9 Appendices