Intro to Embedded Systems: Final Project Report

Team 6: WiFi-Controlled Jumping Sumo Drone

Students	**Caroline Navega Desouza and Ramiro Alvarez**	Spring 2017
Advisors	**Dr. Bassem Alhalabi**	

I. Objective

To control the Parrot Jumping Sumo drone (a.k.a. "Sumo") via Wi-Fi using an accelerometer, Node.js and Raspberry Pi.

II. Methods of Operation

The accelerometer (MPU-6050) is connected to the GPIO pins of the Raspberry Pi 3. The two devices communicate using the I^2C protocol. The Pi and the Jumping Sumo drone connect using Wi-Fi. This connection happens automatically because we configured the Pi to recognize and connect to the drone's Wi-Fi once it is available. The scripts that will enable the drone's movements are sent from the Pi to the drone via the Wi-Fi communication.

III. Software Operation

Node.js is an event-driven programming language with JavaScript extensions, which allow programmers to write JavaScript code that is interpreted by Google's V8 engine (an engine written in C++). Typically, JavaScript is tied to a web browser, however with the power of the V8 engine, JavaScript can be written on files and executed on a variety of environments. One of the major use cases of Node is non-blocking I/O for server side code.

In our program, we use node to create a socket onto which the Sumo drone and the Pi exchange data via Wi-Fi. More specifically, the Pi collects data from an accelerometer every 500ms (half a second), and decides, based on the input it receives, what data to send to the drone. In response, the drone responds with very specific commands.

Given the acceleration component, we use the dominating component (from the X, Y or Z orientation) to determine the direction in which the drone should move. For example, if the accelerometer senses acceleration values of X = 1200, Y = 150 and Z = 101, we know that the dominating motion will be along the X axis. For the drone, that means move either LEFT or RIGHT, depending on whether the value of X is positive or negative. A positive X value means move RIGHT and a negative X means move LEFT. The same applies for the Y direction, which implies a forward motion with a positive Y-value and a reverse motion with a negative Y-value. The accelerometer's ability to detect changes in acceleration in the x, y and z directions allow for 6 different states which could be used to represent {forward, backward, left, right, jump, stop}. We use this knowledge to collect data from the sensor and control the drone's behavior.

We implemented an existing node.js library found on GitHub, but manipulated it to accept the motion commands based on readings from the accelerometer. The existing library uses commands that are typed into a command line to execute the drone's movements and those movements are not continuous as they are with our implementation.

IV. Future Enhancements

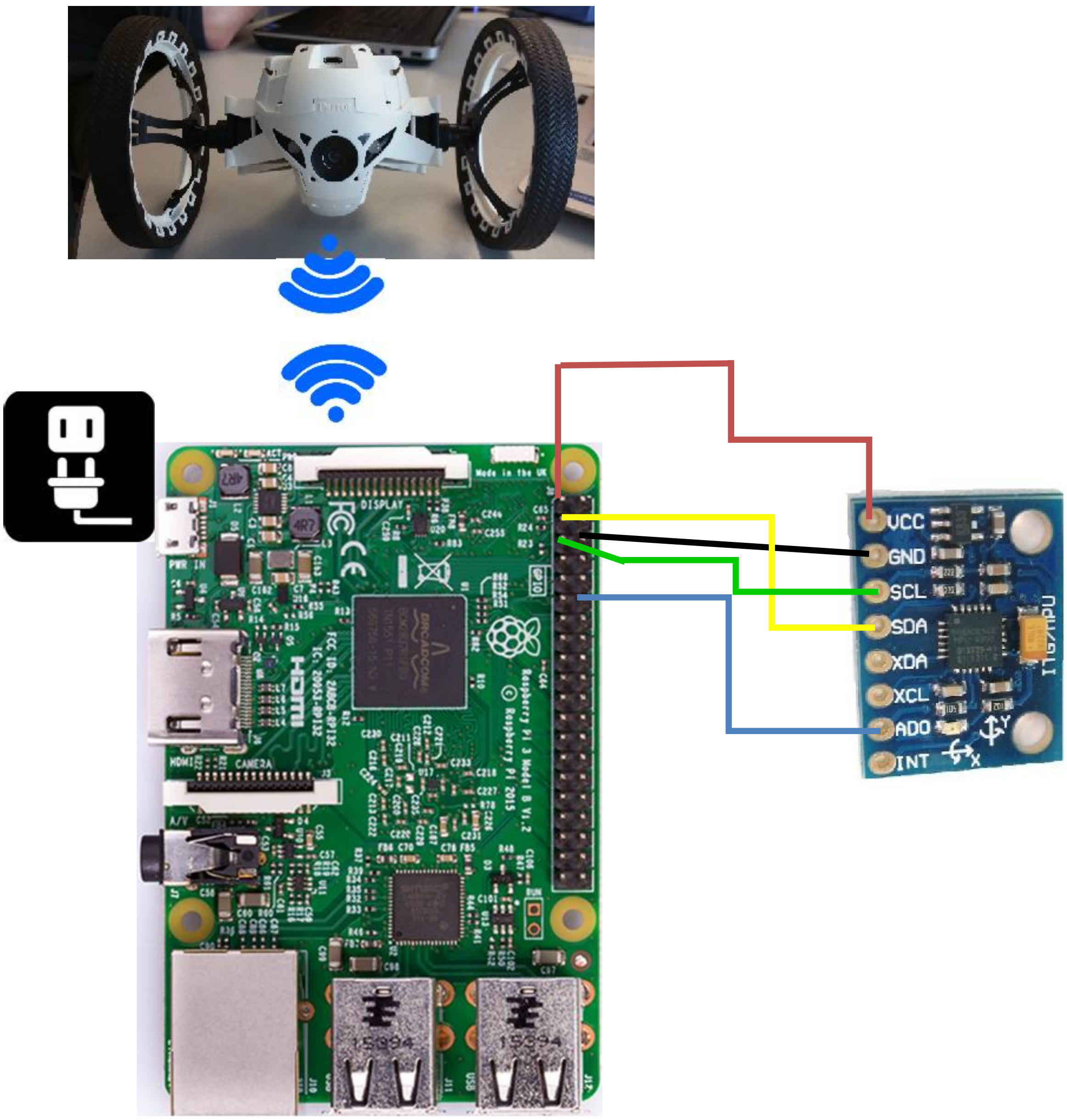
In the future, we'd like to have a better understanding of how sockets operate and be able to create our own code entirely from scratch to manipulate the drone's movements. We'd also like to use the camera's capabilities for object detection.

V. Components Used

- Parrot Mini Drone: Jumping Sumo
- Raspberry Pi 3
- MPU6050 (accelerometer/gyroscope)

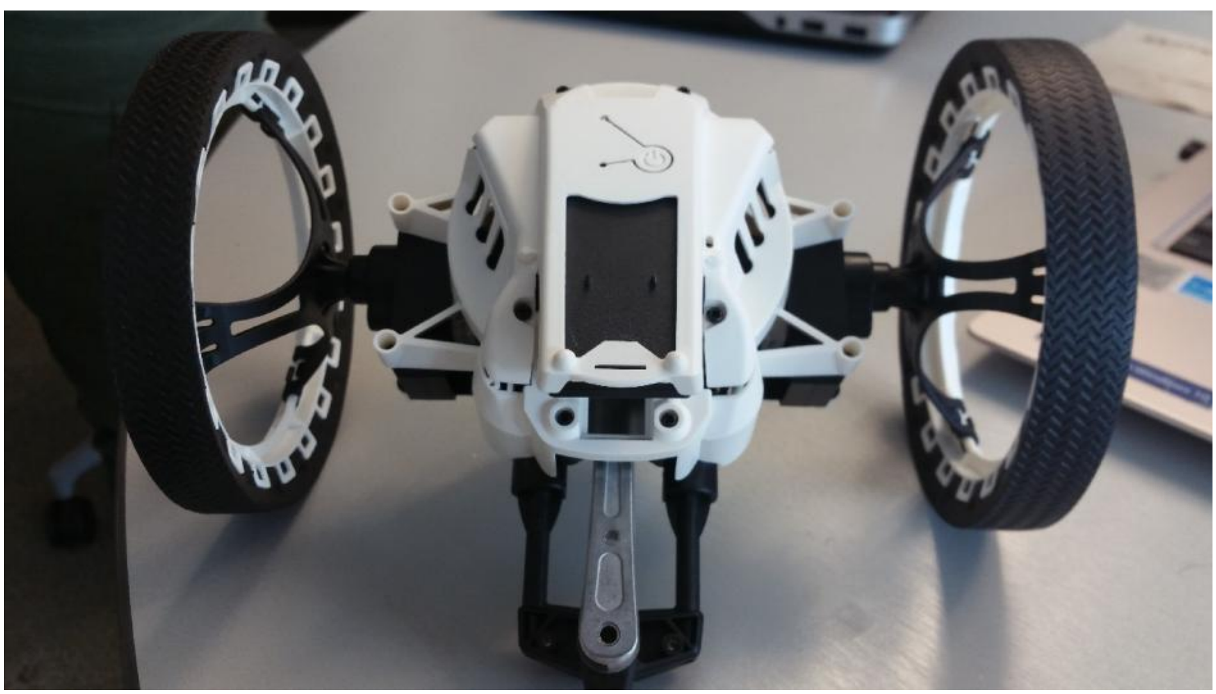
Intro to Embedded Systems: Final Project Report

VI. Block Diagram



Intro to Embedded Systems: Final Project Report

VII. Completed Project Photos



FLORIDA ATLANTIC UNIVERSITY
COLLEGE OF ENGINEERING & COMPUTER SCIENCE

Intro to Embedded Systems: Final Project Report