Brayton Larson
URCA Drone Research
19 May 2019

Controlling Parrot Drones with Python and Pyparrot Library

Introduction

The Parrot Mambo, Bebop, and Anafi can all be controlled using pyparrot, an open source, wireless python interface. There is extensive documentation online, link included below, that has installation instructions and example code for getting started. Before jumping in though, we should note the applications, abilities, and limitations of the package.

Application

First, pyparrot was developed as an educational platform and as a result is not great for testing much beyond basic stock sensors and flight control. However, for learning the basics of roll, pitch, yaw, and altitude control it has applications. Although I never experimented, basic vision processing may be an option through OpenGL and the onboard camera.

Technical Details

The Parrot Mambo is the drone I experimented with using the library, which has the following sensors and readouts available in Python:
Gyroscope: Readouts in the form of roll, pitch, yaw
Accelerometer: Readouts in form of velocities in the x, y, z direction
Altimeter: An averaged readout between the barometer and ultrasonic sensor
Camera (detachable): Live cast available over Wi-Fi connection to connected computer, could potentially be used for vision processing

For all sensor updates the Mambo has a refresh rate of only 2 Hz, meaning that any flight control algorithms such as PID would be extremely difficult. Unfortunately, even with the more expensive Bebop, the refresh rate is only 10 Hz. This does not apply to the camera though, which updates at very low latency.

Programming

There is a basic beginning for each program, outlined in the attached example code. Most functions are documented online, as before links are below, but some of the Parrot's abilities do not have pre-built functions. If desired you could dig into the "minidrone" class and add your own function, but for most purposes it should not be necessary.

Base Station

Any computer that can run Windows or Linux and has a Wi-Fi or Bluetooth connection (depending on your needs) should be able to act as a base station for the drone. I have personally only experimented with it in Linux, Ubuntu 18.04 (14.04 if integrating with ROS Kinetic), but if you are unfamiliar with the environment, Windows may be the better option. However, I have never tested it with Windows and cannot recommend it. This would be a great application for a Raspberry Pi 3 that has Linux, Bluetooth, and Wi-Fi.

Connecting Wirelessly

The Mambo has two available connections: Wi-Fi (when camera is attached) and Bluetooth (camera detached). For Bluetooth connection the MAC address must be included in the object in the Python script, for Wi-Fi it is not necessary.

Using the Attached Examples

There are 3 python scripts I have included because I believe they are a good starting point for flying the drone. The first gives you control over the drone using the keyboard of your base station to fly it in the XY directions. The second is a demo of the direct flight commands. Finally, the third simply connects to the mambo and commands it to land, this is in case the drone is stuck hovering in the air. There are comments in each that explain further.

Further Thoughts

The pyparrot library is constantly being updated with new features so take anything I have said with a grain of salt. Amy McGovern is the author of the project and is extremely responsive to issues on the GitHub page, always remember it is a resource for finding or posting questions. Also, remember that this is not the only open source development platform for Parrot drones. There is also the MATLAB/Simulink interface that is significantly more capable, however much more complex, and the official SDK release by Parrot that without, the prior two could not exist. The pyparrot interface and Parrot drone could be a great educational tool, but it is my belief that it’s sensors and response rate limit real world results greatly. But I would love to be proven wrong. I encourage further research in this area, even if it is just a steppingstone to bigger ideas.

Links

Git Repository: https://github.com/amymcgovern/pyparrot
Documentation: https://pyparrot.readthedocs.io/en/latest/
Official Parrot SDK: https://developer.parrot.com/docs/SDK3/
MATLAB/Simulink Interface: https://www.mathworks.com/hardware-support/parrot-minidrones.html