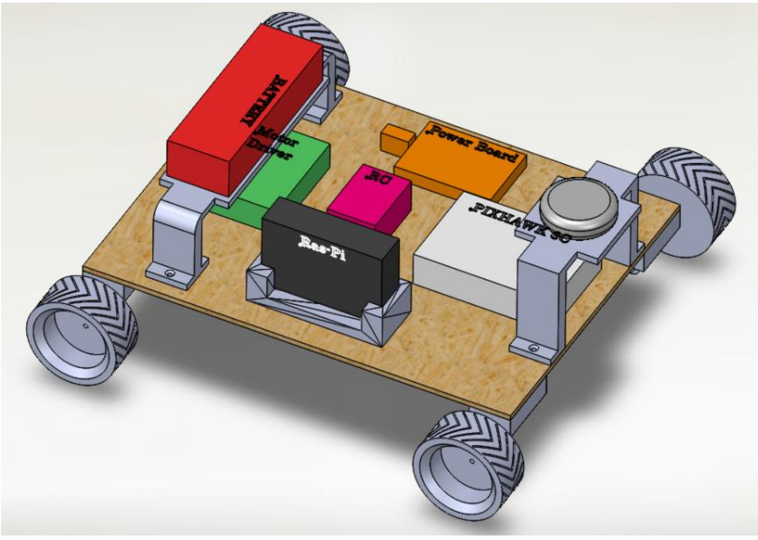
Drone Setup Instructions

Step 1 - Build the drone

To replicate the rover used in this project, you will need the following parts:

- Pixhawk 6C Flight Controller
- Holybro PM07-V2.4 14S Power distribution board
- Sabertooth 12A 6-24V Regenerative Motor Driver
- Two High torque, low rpm RC car motors
- Raspberry Pi 3
- RC Receiver
- Pwm breakout board compatible with Pixhawk 6C
- Lipo Batteries (3S 70C)
- M8N GPS module



Above is an example of how all these parts can be fit onto an A4 sized piece of MDF.

Step 2 – Install Firmware

To begin the firmware setup, the flight controller needs to be configured to control a rover type vehicle. To do this take the Pixhawk 6C and plug it into your computer via USB, then open the Ardupilot Mission Planner software.

(Installation Instructions here: https://ardupilot.org/planner/docs/mission-planner-installation.html)

With the board plugged into your computer, navigate to: SETUP < INSTALL FIRMWARE and click on the rover option then follow the onscreen instructions. The flight controller should now be setup as a rover.

To complete the setup, you will need to wire up all the other components (except for the Raspberry Pi), plug in the Lipo battery to the power board then reconnect the Pixhawk to your computer. Navigate to the top right of the mission planner window and select auto from the left-hand dropdown and click connect. This will connect the autopilot running on the Pixhawk to the mission planner user interface.

Before the rover will properly function, you need to calibrate all the modules. The way to do this is to go to: SETUP > Mandatory Hardware within mission planner. In this window you will need to complete the Accel Calibration and Compass.

Step 3 – RC Control

Your rover should now work, however there is currently no way to control it. You need to pair an RC controller. To do this bridge the B/VCC pins on the RC receiver then plug the Lipo into the rover to power it on, the light on the receiver should be rapidly flashing red indicating it is in binding mode. On the RC controller, turn the power switch on whilst holding down the bind button. The flashing light on the receiver should turn solid and the controller will notify you it is connected.

From here, you need to calibrate the radio. Connect the rover to mission planner again and go to SETUP > Mandatory Hardware > Radio Calibration, follow the onscreen instructions to calibrate the radio then press ok. To now allow the radio to drive the rover, you need to assign the radios instructions to the PWM pins that connect to the motor driver. This can be done in the Servo Output section of the mandatory hardware menu in mission planner, once the correct sticks are assigned to steering/throttle the rover will be fully functional.

Step 4 – Raspberry Pi setup

To enable the rover to move and 'think' by itself, it needs a flight computer. Which is, in this design, a Raspberry Pi 3. Start by flashing a fresh install of PiOS onto the devices you are planning to use.

Connecting to the Pis remotely can sometimes be a bit tricky so I highly recommended that the first thing you install is a RemoteIt server on the RasPis. To do this open a raspberry Pi console window either by remote SSH or via HDMI and hardware peripherals then enter this sequence of commands:

- sudo apt update && sudo apt upgrade-y
- sudo apt install remoteit

This will update everything on the Pi then install a new version of Remote It, it will output a unique device link once the install is completed which can be used to add the raspberry Pi to your device list on the desktop Remote It application. Once this has been done you will be able to remotely access the Pi from any internet connection even mobile data without needing the IP address of the Pi.

To install the remaining required functionality, Restart the Pi then access the Console via Remote It. The input these commands:

- sudo apt update && sudo apt upgrade-y
- sudo apt-get install ccache-y
- sudo apt-get install git-y
- sudo apt-get install gitk git-gui-y
- git clone–recurse-submodules https://github.com/ardupilot/ardupilot.git

- cd ardupilot
- Tools/environment_install/install-prereqs ubuntu.sh-y
- .~/.profile
- sudo apt install lua5.4 liblua5.4-dev-y
- pip3 install mavsdk
- sudo apt update && sudo apt upgrade-y

This installs the ardupilot environment onto the Pis and allows us to access mavproxy for establishing telemetry links between the drones and the ground station.

Step 5 – Establishing Communication

First, ensure Pymavlink is installed on the Pis:

- pip install pymavlink

Now that the groundwork is laid, the files found on this GitHub Repository should function:

https://github.com/S-WLewis/Drone-Swarm-Communication

The code named "MavProxy-DataMoniter" broadcasts the data from the drone it is on via a python socket whilst simultaneously listening for other broadcasts. Once another broadcast has been found, the distance and angle between the drones is calculated then once the script is exited the raw data for the drone the script is running on is exported to a csv file.

The second script uses the RC override feature of Mavproxy to spoof an rc input and move the rover, this script repeatedly makes the rover move forwards and backwards.