Roboboat Electronics and Pixhawk Communications 2023

Christopher Nassif (chrisjnassif@gmail.com)

Hardware We Used

We used a pixhawk based system with a gps module, a receiver to be able to control the boat with a remote controller, and a pwm signal converter to interpret the pwm signal that came out of the pixhawk into discrete channels, so we could better control each of the motors.

Software We Used

Mission Planner:

- This is the main “ground control” software we used to change flight modes (I will get into this later), change the remote controller settings, and manually set waypoints and missions. We won’t be able to use this effectively during the competition, but it is useful for debugging and calibration. https://ardupilot.org/planner/docs/mission-planner-installation.html

QGroundControl:

- This is the secondary ground control. I recommend not trying to change any settings through here because it will mess up Mission Planner a tiny bit, but it is good for very certain things like calibration. But overall, try to just stick to Mission Planner for most tasks.

pyMavlink:

- This is a python wrapper to a C based library that uses the Mavlink communication scheme to communicate with ardupilot externally. That was a mouthful, but to review, this is a library for python that was originally written in C. It uses Mavlink which is a general scheme to send messages to drones and other vehicles, and ardupilot is the software that pixhawk is built on. Ardupilot is the actual autopilot that is computing everything for us behind the scenes and is what makes all of the cool things about the pixhawk tick. (Sometimes Ardupilot seems to be called ArdupilotMega. I don’t know if there is an actual difference, but just keep that in mind)
- This software has a few dependencies to install before you actually install it but just follow this guide and you’ll be fine: https://pypi.org/project/pymavlink/

Mavlink:

- Mavlink is a communication scheme that works by sending commands and messages to ardupilot
- Basic Info: https://ardupilot.org/dev/docs/mavlink-basics.html
- Commands are fundamentally different from messages so before you send either you have to make sure you are using the right code

- There is a huge and very useful compilation of these commands and messages right here: https://mavlink.io/en/messages/common.html
- You will need to reference this a lot

Pixhawk Introduction and Pathfinding

The pixhawk works as an autopilot flightcontroller. It takes a bunch of information from the gps, remote controller, and such and then is able to communicate with the motors through pwm.

There's a lot of neat stuff it could do like if you send it a GPS point then you can travel to it without ever having to manually control the boat yourself. This is called setting a waypoint and this is crucial for our purposes of communicating with the pixhawk. You can do this in mission planner pretty easily following this tutorial right here:
https://docs.px4.io/main/en/flying/missions.html

But we can't really do this during the competition because we won't know exactly where everything is ahead of time. For that we need a lidar and camera to process that information into directions for the boat and then livestream those directions into the pixhawk.

To add to our problems, we couldn't even set the waypoints with Mission Planner and see if it moves to the correct location because we needed the boat to be fully set up, which we weren't able to do this year, so we really have no idea if even that works (sorry).

This is indicative of a larger problem that the pixhawk caused us throughout this project and that is its complete opacity. We simply couldn't understand what was going on in the pixhawk's mind. In pixhawk land, there are few to no error messages and you usually can't tell if something is working even when it is unless you do a full test.

Regardless I have attempted to compile a list of code resources that could potentially work:

- Pymavlink example code for sending gps coordinate to autopilot
  https://www.ardusub.com/developers/pymavlink.html#send-rangefindercomputer-vision-distance-measurement-to-the-autopilot

- External library that might be able to load waypoints with mavlink
  https://www.colorado.edu/recuv/2015/05/25/mavlink-protocol-waypoints

- Youtube video outlining yet another method of pixhawk movement
  https://www.youtube.com/watch?v=yyt4VjBRG_Y&ab_channel=IntelligentQuads
  I believe this type of set position target is meant to be used with guided mode only

This matches with Mavlink Command #84: SET_POSITION_TARGET_LOCAL_NED
I'm pretty sure they are just equivalent ways of doing the same thing

- Mavlink Comand #16: MAV_CMD_NAV_WAYPOINT
  https://mavlink.io/en/messages/common.html
  I think this only works for auto mode

How to send commands and messages:

- General Overview: https://mavlink.io/en/services/command.html

> MAVLink commands (MAV_CMD) and messages are different! These commands define the values of up to 7 parameters that are packaged INSIDE specific messages used in the Mission Protocol and Command Protocol. Use commands for actions in missions or if you need acknowledgment and/or retry logic from a request. Otherwise use messages.

- All commands and messages can be sent using the `command_long_encode` pymavlink function. This is a C style function, so there is a fixed number of arguments and a lot of them are going to end up being 0.
- Here is an overview of the arguments that you would use with the function

Field Name	Type	Values	Description
target_system	uint8_t		System which should execute the command
target_component	uint8_t		Component which should execute the command, 0 for all components
command	uint16_t	MAV_CMD	Command ID (of command to send).
confirmation	uint8_t		0: First transmission of this command. 1-255: Confirmation transmissions (e.g. for kill command)
param1	float		Parameter 1 (for the specific command).
param2	float		Parameter 2 (for the specific command).
param3	float		Parameter 3 (for the specific command).
param4	float		Parameter 4 (for the specific command).
param5	float		Parameter 5 (for the specific command).
param6	float		Parameter 6 (for the specific command).
param7	float		Parameter 7 (for the specific command).

(from message #76)

- That will create an encoded command and then all you need to do is use mav.send() to send it
- Eg:

```
message = pixhawkConnection.mav.command_long_encode(
    pixhawkConnection.target_system, pixhawkConnection.target_component, # random stuff to include
    mavutil.mavlink.MAV_CMD_NAV_WAYPOINT, 0, # name of the mavlink command
    0, 0.2, 0, 0, latitude, longitude, 0)   # arguments to the command
pixhawkConnection.mav.send(message)
```

- On the other hand, in order to send a message, you need to use the mavutil.mavlink.MAV_CMD_REQUEST_MESSAGE command in place of where you would normally put the name of the command you want to execute and then specify the message in one of the later arguments

Param (:Label)	Description	Values
1: Message ID	The MAVLink message ID of the requested message.	*min:*0 *max:*16777215 *increment:*1
2: Req Param 1	Use for index ID, if required. Otherwise, the use of this parameter (if any) must be defined in the requested message. By default assumed not used (0).	
3: Req Param 2	The use of this parameter (if any), must be defined in the requested message. By default assumed not used (0).	
4: Req Param 3	The use of this parameter (if any), must be defined in the requested message. By default assumed not used (0).	
5: Req Param 4	The use of this parameter (if any), must be defined in the requested message. By default assumed not used (0).	
6: Req Param 5	The use of this parameter (if any), must be defined in the requested message. By default assumed not used (0).	
7: Response Target	Target address for requested message (if message has target address fields). 0: Flight-stack default, 1: address of requestor, 2: broadcast.	*min:*0 *max:*2 *increment:*1

(from command #512)

Flight Modes:

- Main ones are guided, auto, and manual
- Manual is for remote control stuff. If the remote control isn't working, it's because the pixhawk isn't set to manual mode.
- Auto mode is for waypoints
- Guided mode is for the SET_POSITION_TARGET family of mavlink commands

A Word About the Remote Controller Configuration

Right now we have it set that our far left bumper is to arm the pixhawk and one of the bumpers to the right hand side of the controller controls between guided, auto, and manual

mode. I'm sorry I can’t be there in person to point you to which ones exactly but it's not too bad to flip around to find them. The auto, guided, and manual 3-way switch can also be configured inside of the mission planner if you want to change that around.

Future Plans, Notes, and Conclusions

It really seems like the pixhawk was never intended for livestreaming information, and was instead really only meant for missions that were set prior to the flight. It also turns out that the pixhawk was mainly meant for drones and we are just trying to make it work in our boat, which admittedly makes our situation worse. There are so many ways to do anything and there's so much competing documentation, and this document is the best I could come up with. I came into this knowing nothing about communications protocols and standards between vehicles and I have come out of it with the knowledge to never touch the drone community with a 10 foot pole. They can keep their barely functioning software and documentation to themselves. But in all seriousness, my team members and I have been talking about not using the pixhawk at all for future years because communicating with it externally is such a nightmare. Instead opting to use a raspberry pi with ardupilot installed to act as a rudimentary pixhawk. This would add a lot of complexity, but it takes out all of the headaches of dealing with the pixhawk and trying to make a square block fit in a circular hole. Also sorry we weren’t able to do a full test and most of my code for the pixhawkMessager.py script is probably non functional, but hopefully it and all of the stuff I did with setting up the pixhawk and gathering all of these resources will be worthwhile for whoever wants to pick up TJ Roboboat, and if you ever have any questions feel free to email me at any time.