

Drone Software Engineer Challenge

A new company has approached Freefly, and they want to use our Alta X drone to 3d print giant foam structures! They have a dispenser that mounts underneath the aircraft, and spits out foam that expands and hardens. What they need from Freefly is an application that can command the Alta X to fly to specific XYZ locations they specify in a text file so it can dispense foam along the way, building the structure.

You, as the assigned drone engineer, need to create the following:

A MavSDK module written in C++ or python that can control a drone and fly an arbitrary pattern that is defined in a text file. The aircraft only needs to be able to fly in straight lines- any curves in the structure are turned into short straight segments by the customer's software.

Please provide the following:

1. Source tree and build directions for the module you created
2. Include a sample text file that draws a pattern as an example. Include a definition of the text file format so the customer can develop an export tool from their software.
3. A Screen capture video of your module controlling one of the PX4 simulators (jmavsim or gazebo recommended)
4. Include the log from your sample flight. It can be found in the PX4 build/px4_sitl_default/logs folder. Include any pictures or plots you see fit to demonstrate your module is working as intended.
5. A 1-2 page document describing any considerations you made in the design of the module, such as where the module might run in the overall system, compatibility with new drone types in the future, control handoff between pilot and module, safety considerations, failsafes, flight performance limitations, etc. Also include high level steps of what you'd do next to take the design from simulator to testing on a real aircraft.
6. Any personal touches you'd like to add- details, thoughts for improvement, whatever!

You will need to clone the mavsdk repository or get prebuilt libraries. The PX4 repo will need to be cloned in order to build the simulator. QGroundControl can be downloaded as a binary, but is not strictly necessary for this project.

We anticipate that this should take 4 to 8 hours of development and test time. Feel free to reach out to us if you have questions. Have fun! We're looking forward to seeing what you create!

Links:

PX4 repository:

<https://github.com/PX4/PX4-Autopilot/tree/v1.11.3>

PX4 simulation:

<https://docs.px4.io/master/en/simulation/>

MAVSDK:

<https://mavsdk.mavlink.io/develop/en/index.html>

QGroundControl (the “normal” PX4 ground station):

<http://qgroundcontrol.com/>

Online PX4 log review:

<https://logs.px4.io/>

Other log analysis tools:

https://docs.px4.io/master/en/dev_log/flight_log_analysis.html

Plot juggler log plotting tool

<https://github.com/facontidavide/PlotJuggler/releases/tag/2.8.4>

Example screenshot showing simulator (right) doing a manually created mission using QGroundControl(left), for reference.

