1 Introduction

In this proposal, I outline a project to create a method of drone-to-drone communication, and using this communication to avoid an autonomous mid-air collision with another (autonomous) drone. The ultimate goal will be to have a proper open-source stack of tools that can be used to communicate with other drones and act on that information.

2 Motivation

Technology has always moved us towards automation, and autonomous flying vehicles are the next step in this process. Amazon and other companies have already captured our imagination with promises of package and pizza deliveries, all processed and executed quickly and efficiently. Although regulatory agents and public acceptance remain some of the more immediate roadblocks stopping a drone from dropping some Domino’s pizza on your doorstep, there is another concern.

In a world where there are thousands of drones buzzing back and forth - a world I hope exists in the near future - there needs to be fail-safes in place so that they will not collide with one another. Furthermore information about local congestion may help a drone make smarter decisions about path finding, missing the possibility of a close call altogether.

One way of doing this would be loading up each drone with sensors and radar to ensure that everything around it is identified. This adds cost, making entry costs prohibitive, and weight, making it more dangerous in the event of an engine failure at altitude. This also does not provide a method of negotiating route or altitude adjustments. Two drones which are only sensing each other will have to guess which way to go to avoid a collision, and hope the other drone does not make a similar adjustment.

In another vein, hobbyists will always be flying their drones in places and spaces they should not be (see: drone crash on the White House lawn, drone crash into one of the geysers at Yellowstone, the list goes on). This includes airspace that may be reserved for autonomous drone operation in the future. This drones will be inexpensive and relatively bare-bones compared to their industry-grade counterparts, and will not have the advanced vision and radar systems that you might expect from a delivery drone of the future. Allowing these drones to communicate with the wider ecosystem of drone activity using a method such as mine, on relatively inexpensive hardware like a short-range radio, would allow industrial drones to not worry too much about colliding with a drone whose cross-section might be too small to detect through normal means.
3 Proposal

To accomplish this, I will be working with the PX4 autopilot and PX4-based hardware (either the PX4FMU, or the Pixhawk). I will be working with Professor Shao and whoever may be interested on his team.

Overall, the project will be about this drone-to-drone communication, and the steps leading up to it. It will encompass changes to the real time operating system, the networking capabilities, and the on-board programs of the drone.

The drone itself should not pose too many issues. We will get a drone that runs on open hardware, and (more importantly) open software configuration. This drone will likely run on the Pixhawk, a powerful flight controller designed by a research team at ETH Zurich, an engineering university in Switzerland. There are many ready-to-fly models of quadcopters available online, and I will work with Professor Shao’s research team to find the model that will be most useful to them for future research.

The current drone in the FLINT Lab has been modified quite a bit in the previous research initiative that used it. As such, I have not yet been able to get it in the air, although I can communicate with its RTOS over a serial link. It seems like even in the case where I cannot get the existing PX4 drone in the air, I can still use it to broadcast signals as if it were in the air, leading to an effective one-drone simulation of a flight collision, without risking any hardware.

The PX4 autopilot is built on NuttX, an open-source real time operating system which exposes POSIX-like interfaces. Using the NuttShell, I can communicate with the RTOS and at least simulate flight on the drone that we have currently in the lab. The version of NuttX that is used in the PX4 autopilot project is fully built and flashed when you update the firmware on a flight controller. Because of this, I can modify any layer of the code and still have it loaded onto the board. This gives me a lot of latitude when it comes to programming the board.

The network between the drones is a little sticky. There does not seem to be a proper networking protocol that allows devices to form a homogeneous mesh, with devices popping in and out of range and existence. This is likely because in Internet of Things applications, you can assume there is a solid power supply somewhere, and build a router around that. Because we can make no such assumptions in the air, the only suitable existing networking protocol seems to be Digi’s proprietary DigiMesh solution, built on top of 802.15.4. I can use this to bootstrap my project.

The current standard for communicating with drones in the air is on a protocol called MAVLink. MAVLink is a well-defined messaging protocol for communicating a drone’s location, direction, altitude, and other types of useful information. Right now its main use is for feeding a drone’s location back to a laptop ground station, with the ground station giving the drone orders on-the-fly. I can modify these messages and commands to be sent and received from other drones, while ensuring that the only changes made to the flight of the drone are necessary ones.

The ecosystem of support for all these above-mentioned aspects of the project is strong. There are many active forms and discussion groups for Pixhawk operation, PX4 development, and the like. If I turn frequently to these resources, and the resources that I have talked to Liang and Professor Shao about, I should be able to make steady progress towards the goal of drone-to-drone communication.
4 Deliverables

This is an ambitious project, as I have gone over with Professor Shao, and many of the goals could be put on hold because of my lack of experience with embedded systems programming, hardware issues, or a misjudgment of the capabilities of such a flight system like the Pixhawk and PX4 autopilot. However, I do think I can at least deliver these things:

- Establish a link between two boards running PX4, which can be reestablished after interference, power loss, or anything else.
- Over this link, send commands from one to the other in the form of MAVLink communications, which is the defacto standard for communicating between a drone and a ground station.
- Have an application running on top of NuttX be able to adjust flight parameters, or expose interfaces in NuttX to do so.
- Make a process to negotiate altitude and route changes if two drones detect that they are on a collision path, to ensure that with the route changes they do not collide.

Other parts of this proposal are a little more complicated, although if all goes well I expect to be able to finish all of them. They include

- Get the PX4 drone in the lab flying for the demonstration.
- Use a non-proprietary networking protocol to get a homogeneous mesh network in the air, where you can take nothing for granted.
- Bring it all together. Demonstrate two drones, each one set on a collision-course, negotiating a path around each other and avoiding the collision.