Design and Development: Raspberry Pi Drone

Project Summary by: Roberto Hernandez (CS Department)
May 15, 2023

Our overarching objective is to develop and implement an Unmanned Aerial Vehicle (UAV) swarm. Our first step is to design a single UAV (Pi Drone) which we've designed to work in conjunction with ROS2 and Pixhawk 4 flight controller. Both the Pixhawk hardware and ROS software are open-source and have been extensively tested, making them ideal for anyone to implement. For the base operating system, we've opted for Ubuntu, with ROS2 serving as the middleware linking our API to our federated learning software. In an ideal scenario, you can purchase smaller or more affordable kits, or even repurpose a hobby drone. The only essentials are the Pixhawk and Raspberry Pi.

We opted against purchasing prebuilt drones with an API/Develop Kit due to restrictions, high costs, and the software stack involved. Additionally, we refrained from acquiring premade drones with a compatible controller due to their expense and the understanding that we would inevitably encounter breakages in the early stages of our project. Furthermore, we didn't invest in drone kits as it was crucial for us to understand the hardware and selectively choose what suited our particular use case as we built out our project.

We selected the following components due to their relative affordability compared to other options, and their potential for substitution with similar parts, thereby facilitating usage by others.

Flight Controller: We went with the Pixhawk 4 because it supports an open standard and can be produced by multiple manufacturers. We specifically chose the fourth revision as it's an older model, while the latest is Pixhawk 6. This allows us to take advantage of better support tools and environments than what the latest hardware might offer.

Micro Controller: We chose the Raspberry Pi 4 for its low power consumption and excellent software tool support. We are considering alternatives which could assist with larger federated learning computations. Depending on our software implementation.

Drone Frame: We picked a standard 500mm frame. Anything smaller wouldn't accommodate the necessary hardware, while anything larger would require a significantly higher investment. This is a common size ideal for software development and small-scale testing.

Drone Motors: We decided on the 935kv 2212. Any motor within the 900-1200kv range should support a drone with all the required hardware.

Propellers: Any 1045 (10 inch, 45 pitch) propeller is suitable for this size.

Electronic Speed Controller: We selected a 2-4S 30A, which enables us to use various types of batteries and upgrade the motors if we decide to enhance the frame for more power.

Power Distribution Board: We chose the FCHUB-12S. While it's overkill for our current use, it gives us plenty of overhead in case of upgrades to the frame in the future.

Battery: We went with 3S/6S. Smaller capacities are ideal for testing, firmware updates, and other non-flying tasks. Larger ones are for flying and provide enough power to sustain flight testing of simple software, when we finalize software implementation we can have a real time flight time, which we can then adjust our batteries and ESC if needed.

Battery Charger: We selected a dual battery charger with BMS that supports 1S-6S batteries for versatility.

Converter 24v to 19v: This connects directly to the power board, allowing the use of higher-end components such as the Intel NUC for computations.

Battery Splitter Cable: This facilitates powering a 19v Intel NUC directly from the battery via the voltage shifter.

Remote RC Receiver: We picked a standard mid-range Remote RC for emergency manual takeover and auto landing in case of software Eternal Guard Failure and for testing purposes.