Hardware Report

1 System Overview

1.1 Hardware Parts

1. Frame assembly: Assembling the frame of a drone is a critical step in building the drone. The frame serves as the foundation upon which all other components of the drone are mounted. A well-built frame provides a stable and sturdy base for the drone to fly, making it easier to control and maneuver. Typically, drone frames are made from lightweight and durable materials such as carbon fiber, aluminum, or plastic. The frame consists of several key components, including arms, legs, or other structural elements that are attached together to create the main body of the drone. The arms are the longest components of the drone frame, and they serve as the platform for attaching the motors and propellers that provide lift and thrust to the drone. The arms are made of carbon fiber or aluminum, as these materials are lightweight and strong. The legs of the drone frame provide stability and support during takeoff and landing. They are typically shorter than the arms and are attached to the bottom of the drone's body. The legs can be made from the same materials as the arms, but they may also be made from plastic or other lightweight materials. Once the arms and legs are assembled, the next step is to attach them to the main body of the drone. The body may consist of two plates stacked on top of one another. The plates are held together with screws, and the arms and legs are attached to the plates using bolts or screws. It is essential to ensure that all components of the frame are securely attached to each other, as any loose or weak connections can result in the drone crashing or malfunctioning in flight. Additionally, it is important to balance the drone's weight distribution to ensure it is stable during flight.

2. Mounting the motors onto the drone is a crucial step in the drone assembly process, as it directly affects the drone's ability to fly safely and efficiently. The motors are the primary source of propulsion for the drone, and they work together with the propellers to generate lift and control the drone's movement in the air.When mounting the motors onto the drone, it is important to consider several factors to ensure that they are installed correctly. We mount them onto the arms or legs of the drone. This process typically involves using screws or bolts to secure the motor onto the frame, ensuring that it is mounted straight and securely. It is crucial to be extremely careful when tightening the screws that secure the motor onto the frame. If the screws are too long or are tightened too tightly, they can come into contact with the coils of the motor, which can cause damage or even lead to the motor malfunctioning during flight. Therefore, it is essential to use the appropriate screws and to tighten them carefully and evenly. In addition to mounting the motors correctly, it is also important to balance them properly. Balancing involves adjusting the position of the motors and propellers to ensure that the drone is stable during flight. An unbalanced drone can be difficult to control and can result in unstable flight or even crashes.

3. Electronic speed controller (ESC) installation: The ESCs are connected to the motors and regulate the speed and direction of the motors. One of the ESC was burnt, the replacement can function properly except it automatically starts spinning without command at arm stage. The ESC was padded by a layer of foam tape to avoid any short circuit problem with the bottom plate of the drone. Additionally, the orders of wire that are being plugged into the flight management unit have sequence and should be plugged accordingly.
4. Flight management unit installation: The flight management unit is the brain of the drone and controls the movement and stability of the drone. It needs to be mounted on the drone and connected to the ESCs and other components. When trying to plug in cable, we need to be extremely careful with all the wires since they are too thin and vulnerable to strong force. By following the instructions given on official website, each module can be wired correctly.

5. Battery installation: The battery provides power to the drone and needs to be installed in a secure location on the drone. The battery is 14.8 V and 3000mAh and can be rechargeable using the charger that is included in the kit. The battery itself is large and heavy, therefore, we attached it to the bottom of bottom plate to center the mass, and make it more stable during the flight,
6. Propeller installation: The propellers need to be attached to the motors. It's important to make sure that the propellers are mounted in the correct direction and tightened securely. In this project, due to its safety, when demonstrating, we will be removing the propellers.
7. Camera and Raspberry Pi installation: If your drone has a camera and Raspberry Pi, these components need to be mounted onto the drone to achieve real-time data transmission, and flight path control
8. Remote control installation: Depending on the drone design, may need to install a remote control receiver onto the drone and connect it to the flight controller. The drone is paired with an RC receiver and the remote control needs to be set up by QGroundControl.

1.2 Instruction

To power the entire system, connect the battery attached under the drone to the power cable from the main body of the drone. Beep tunes should first come from the rotors, then from the FMU (the white box located at the top of the drone). If one or both of the beep sounds are not heard, troubleshoot. Then turn on the Remote Controller by pressing down the two power buttons on the front panel and holding for 2 seconds. If the Remote Controller does not turn on, check its batteries. If the Remote Controller can show the battery percentage of the drone properly, it means the Remote Controller is set up correctly. If not, troubleshoot.

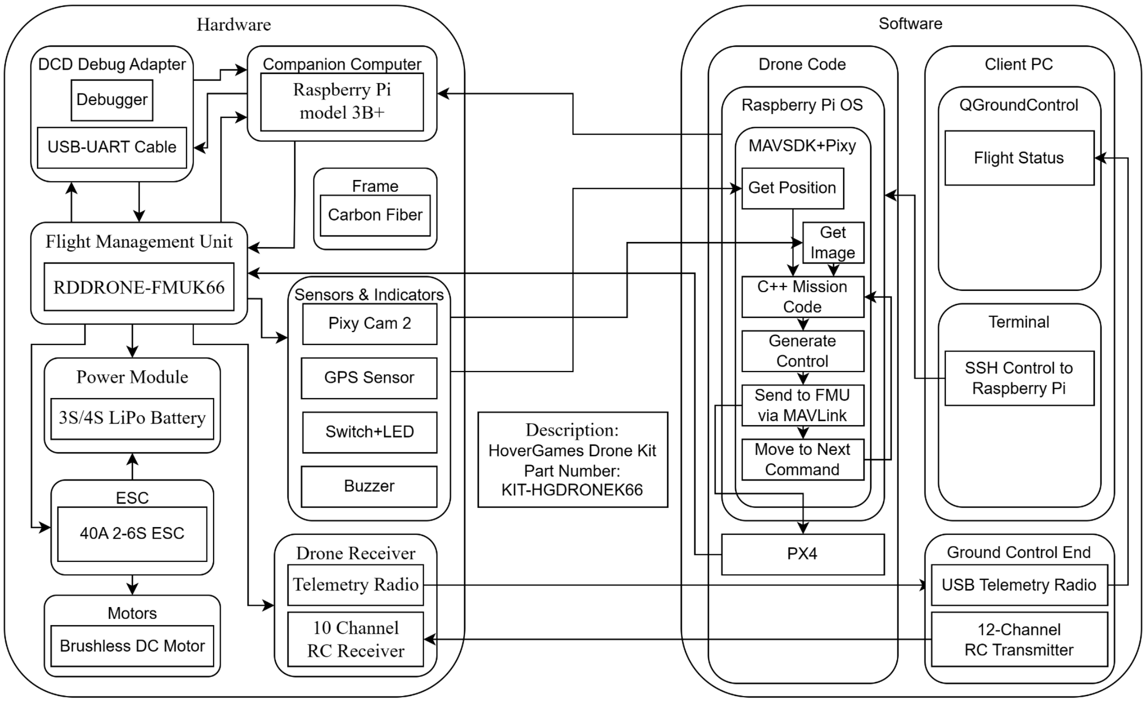
The drone does not require the companion computer (Raspberry Pi 3B+) to fly. However, to run the Pixy Camera module or to run any C++ mission code, the companion computer setup is necessary. Power the companion computer by connecting its micro USB port to any USB-A port of the battery bank attached to the bottom of the drone. Wait for 2 minutes. Then it should allow SSH remote access from any computer connected to the WiFi "BU Guest". Have another cable of the same type. This time, connect the USB-A end of the cable to the companion computer and the micro USB end of the cable to the FMU.

Debug and Telemetry Radio both require QGroundControl on a laptop. To have QGroundControl, Google search QGroundControl and download it from its official website. Allow all drivers during installation.

To debug the drone with a wired connection, connect the micro USB port on the side panel of the FMU to a USB-A port on the laptop which installed QGroundControl. Rebooting solves most connection problems. To use the Telemetry Radio, attach one of the two transceivers (doesn't matter which one) to the drone and connect its GND/RTS/CTS/TX/RX/5V port to FMU as shown in Fig 6. Connect the other Telemetry Radio to the laptop with a micro USB to USB-A cable. The Telemetry Radio does not need separate powering.

To charge the Li-Po battery of the drone, connect the 4-pin white connector from the battery to the charging dock. Wait for 10 seconds. The charging dock should be able to display the voltage of the battery. If not, DO NOT plug the charging dock into wall power! (This means the battery/charging dock/both have problems.) If the voltage is properly displayed, connect the charging dock to the wall power.

1.3 Overview block diagram



The block diagram shows the structure of the drone, separated into hardware and software parts. The most essential part of the drone is the FMU (Flight Management Unit), with the most arrows coming in and out. It has a micro SD card which contains the settings we tested to work. However, the FMU has its own memory so formatting this micro SD card does not guarantee to clear all settings.

The companion computer (Raspberry Pi model 3B+) is the 2nd most important part of the drone. It also has a micro SD card which contains the OS for the companion computer and the C++ mission code we configured to work. ESC is short for Electronic Speed Controller. The Switch+LED is the pressable lighted button on the GPS. The C++ Mission code is not a single file but multiple C++ source code files. There are also some ready-to-run precompiled binaries from these source code files.

The Telemetry Radio consists of two transceivers, one for connecting to the drone, and one for a laptop. It is used combined with the software “QGroundControl” to display information about the drone on the laptop and to allow remote control of the drone from the laptop. Notice that the Telemetry Radio is not a hard requirement for flight (which means that the drone can fly without the telemetry radio). It also adds load to the FMU which can cause “CPU load too high” warnings, which can prevent the drone from passing pre-takeoff status checking.

The 10-Channel RC Receiver on the drone and the 12-Channel RC Transmitter, on the other hand, is a hard requirement for flight. During testing, we have encountered many times that the drone cannot take off without the receiver and transmitter installed properly. It is a good idea to leave the RC Receiver on the drone and never take it off.

1.4 User interface.

There are three interfaces for controlling the drone.

1. RC (Remote Controller)
2. Telemetry Radio + QGroundControl Software
3. SSH into the Raspberry Pi 3B+ then Run Customized Control Scripts

The RC is mandatory for proper takeoff of the drone AT ALL TIME. Bypassing it by forcefully changing the rotor speeds in QGroundControl does not make the drone go through its takeoff routine!

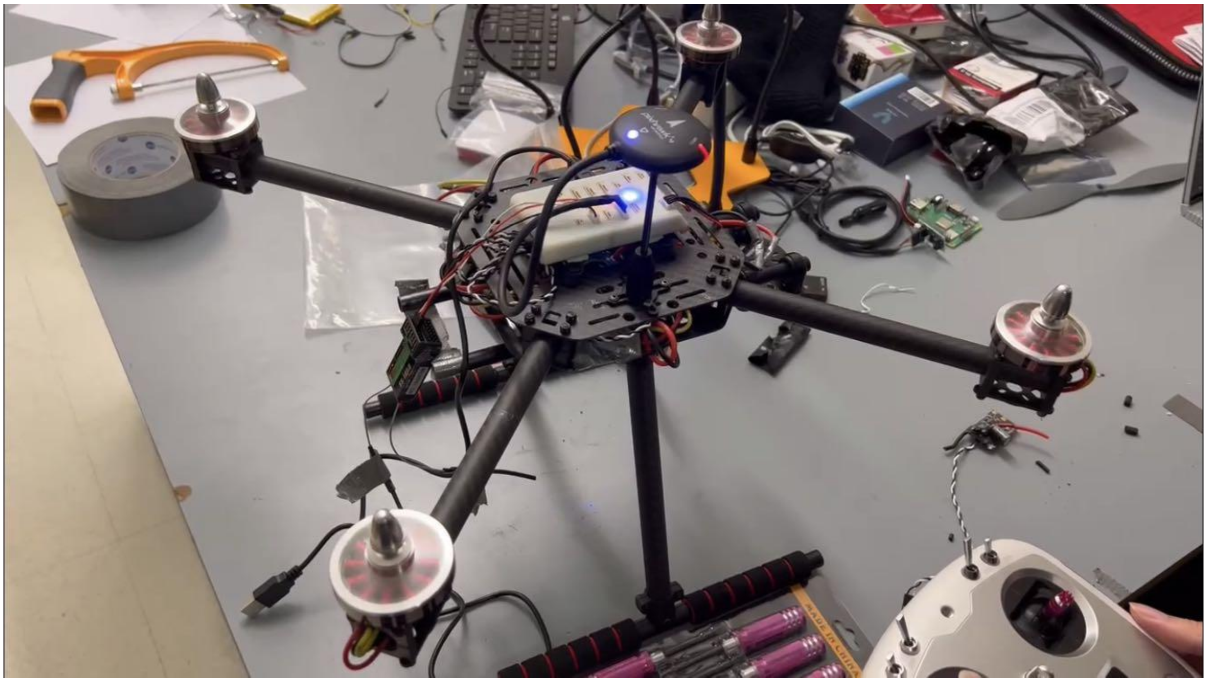
The following list ranks the amount of information obtainable from each user interface from the most to the least:

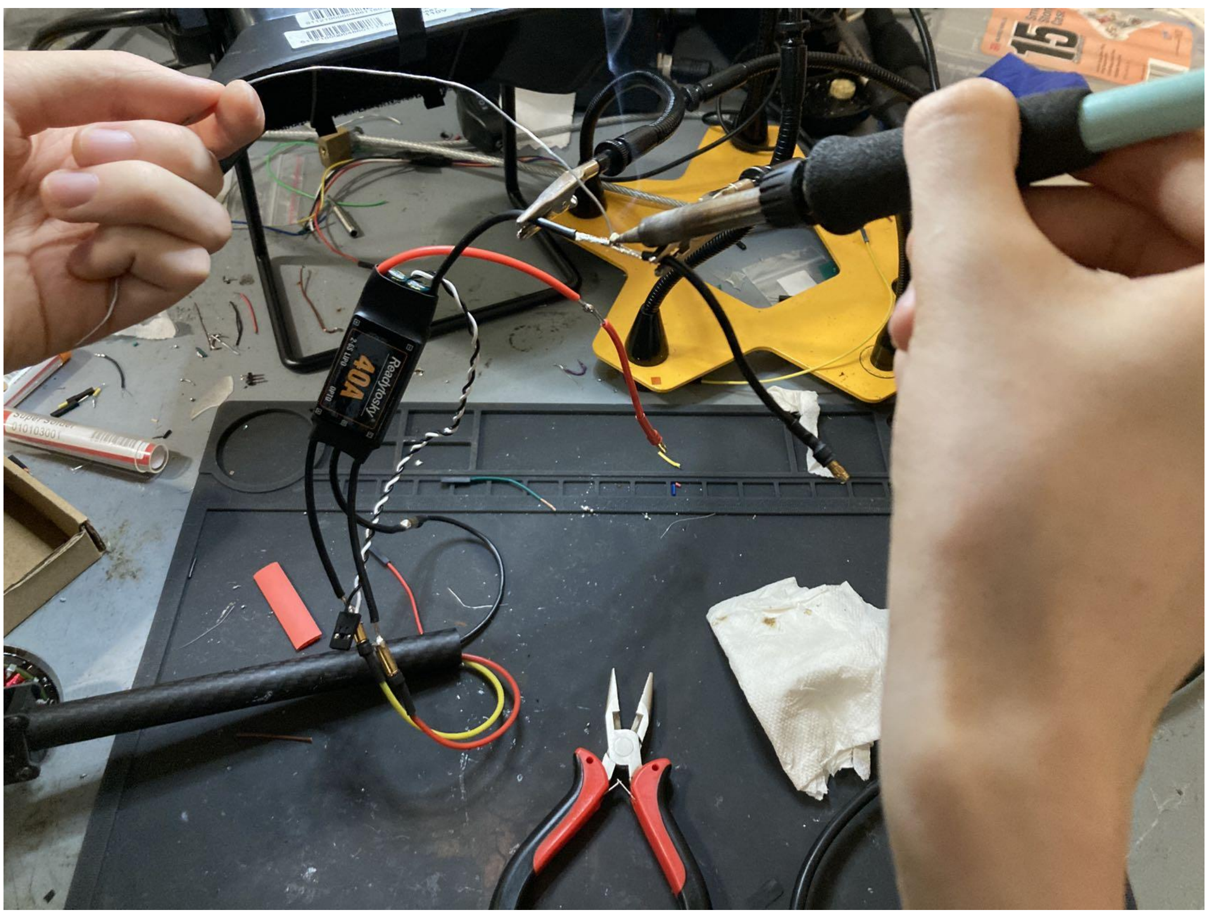
1. Telemetry Radio + QGroundControl Software

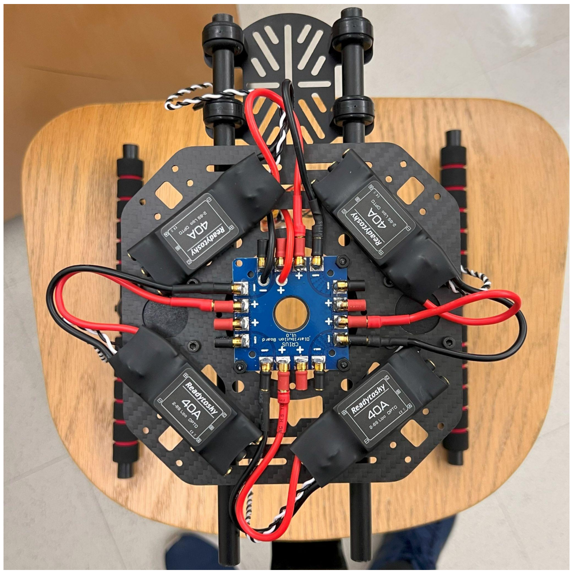
2. Terminal Output from Customized Control Scripts in Raspberry Pi 3B+
3. RC (Remote Controller)

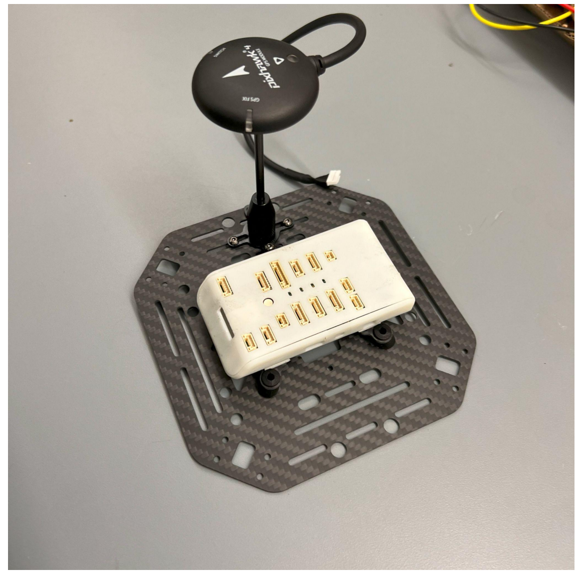
The first method gives a lot more information than the RC. If the FMU is connected directly to the laptop, it is possible to get even more information. In a real flight, it is not practical to get terminal outputs in method 2, as the drone will very likely lose connection to the WiFi. Method 2, however, is the most useful one for debugging the Customized Control Scripts.

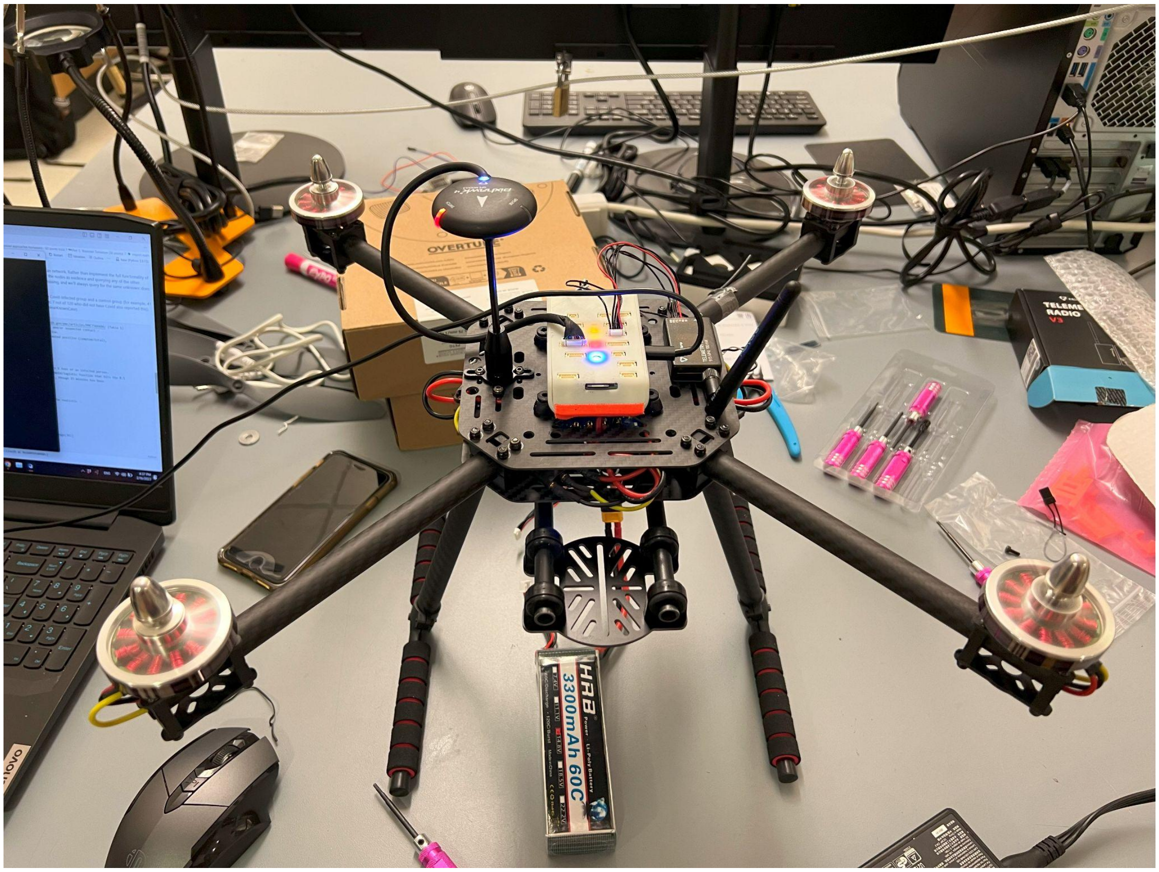
1.4 Picture of system






Readytosky
40A




HRB
3300mAh 60C