Technical Review of Quadcopter Firmware Design – Gearoid Sheehan (R00151523)

Introduction

From the maiden flight of Louis Breguet's four-rotor helicopter in 1908 to the technologically advanced drones of today, the quadcopter has existed as a design concept for over a decade. Improvements in engineering and the availability of open-source software online have resulted in quadcopter use exploding in recent years. From in general hobbyists to those in professional settings such as photographers, emergency response teams and agriculture, the quadcopter has made its mark as a frequently used tool in the modern world.

The quadcopter is a four-rotor unmanned aerial vehicle or UAV. Two of these rotors spin clockwise, while the other two spin anti-clockwise. This is to satisfy the condition for equilibrium, where for an object to be stable the net forces acting on a body should be zero, or mathematically Fnet = ma = 0. It uses an independent variation of the speed of each rotor to gain control, generating thrust more significant than the weight of the object by rotating at fast speeds, pulling the air downwards and causing the object to become airborne. The quadcopter's flight controller sends information to the motors via their electronic speed control circuits (ESC) information on thrust, RPM, (Revolutions Per Minute) and direction. The flight controller will also combine IMU, Gyro and GPS data before signalling to the quadcopter motors on thrust and rotor speed. More advanced quadcopters make use of mathematical models and algorithms to enhance their flight abilities further. Open-source flight controller projects allow quadcopter owners to pick and choose the software to operate their device. Each usually focuses on some specific hardware, technology or aim. The flight control views offered by the firmware is an important point which will influence a user's choice. First-person view, or FVP, offers the pilot a real-time video stream from a camera mounted on the quadcopter. This video can be viewed by the pilot via a virtual reality headset, on a remote-control screen or even a general mobile device. These cameras use either a CMOS or CCD sensor to capture the footage. Most quadcopters use CMOS cameras as they are cheaper, lighter and take less power to use.

The varied use of quadcopters has resulted in an array of different design and build types. Mini drones such as the Parrot Airborne Mini Drone are cheap and will fit into the palm of your hand. They have a minimalist design and lack unnecessary features such as GPS, and their firmware is simplistic. More advanced hobby and professional quadcopters are larger and are feature-rich, some of which using complex systems depending on their use. Custom hobby quadcopters use open-source firmware's such as ArduPilot, allow the pilot maximum customisation and control of their UAV. Professional quadcopters created by companies such as DJI and Yuneec offer high tech quadcopters for use in professional settings such as live television and military use.

Overview of Quadcopter hardware and control

Control of a stereotypical quadcopter can be achieved through the combination of several essential parts

- Frame
- Battery
- Motor
- Power Distribution Board (PDB)
- Electronic Speed Controllers (ESC)
- Flight Controller
- RC Controller

Frame
Quadcopter frames are built in a cross design, with mounts for the motors at each of the four ends. Their centre acts as the hub where the control parts and battery reside. They are typically built with carbon fiber, and the industry standard for a quadcopter frame is 5 inches. This measurement is a depiction of the maximum size propeller which can be mounted and revolve successfully without impeding its neighbouring propellers, not the length of the frame.

Battery
Nearly all quadcopter batteries are lithium-ion batteries and the voltage emitted from them is directly proportional to the power generated by the motors. It is recommended that batteries which have optimum discharge C rating are used, as batteries with a low discharge rating will cause underperformance and even damage the motors. The physical size and weight of the battery must also be carefully considered. This to prevent causing adverse effects to the drone, such as prompting stability issues due to uneven weight distribution.

Motor
There are two main motor types which quadcopters use, brushed and brushless motors. A brushed DC motor has an armature, a commutator, brushes, an axle and a field. They are cheaper, less durable and heavier than their brushless. Brushless motors use synchronous motors, with four permanent magnets stationed around the motor. They require less maintenance due to the lack of friction in their design. Each propeller in the quadcopter is rotated using a motor powered by the PDB, and its speed is monitored by the ESC. A popular example of a brushless motor used in quadcopter builds is the Emax MT2213-935KV. It offers a thrust of 850g for every volt passed into the motor and can be utilised for various other kinds of multirotor.

Electronic Speed Controllers (ESC)
An electronic speed controller accompanies each of a quadcopters four motors. The ESC components control the amount of power which is left through to each motor. The ESC has three wires input and three wires output. Two of the input cables receive power from the PDB, while the remaining input cable is a signal wire connected to the flight controller and informs the ESC how much power to allow through to the motor. All three of the ESCs output wires connect the motor, sending power to each of the motor’s electromagnets one after another. This pulls the motor around to each electromagnet in a circular fashion and essentially causes the motor to rotate. The faster the power iterates into the motor from the ESCs output wires, the faster the motor will spin.

Flight Controller
The flight controller is a small circuit board which acts as the brains in the quadcopter. It receives the transmission from the pilot controller and manoeuvres the quadcopter in the desired motion by manipulating the motors appropriately. Typically, flight controllers contain a variation of the STM electronics 32-bit microprocessor chip. Another chip called the gyro senses the angular velocity or the speed at which a quadcopter rotates in the roll, pitch and yaw axis. Using calculus mathematics and gyro inputs, the FPV Drone Flight Controller can estimate the distance a quadcopter has rotated and whether its rotation is accelerating or decelerating. The acceleration force is calculated by the accelerometer, a sensor chip on the flight controller. This can be in turn used to calculate useful info such as the direction, velocity and change of altitude. Other chips may also be present on the flight controller giving functionality such as GPS and Barometers for long-range quadcopters, and current

sensors for in-flight battery management. The quadcopter is programmed by flashing the firmware of choice onto the flight controller via a micro USB port. Popular firmware's include ArduPilot, Px4 Flight Stack, CleanFlight and LibrePilot. Each firmware has its own specific traits, pros and cons, and their compatibility will depend on the flight controller installed in the quadcopter.

Power Distribution Board (PDB)

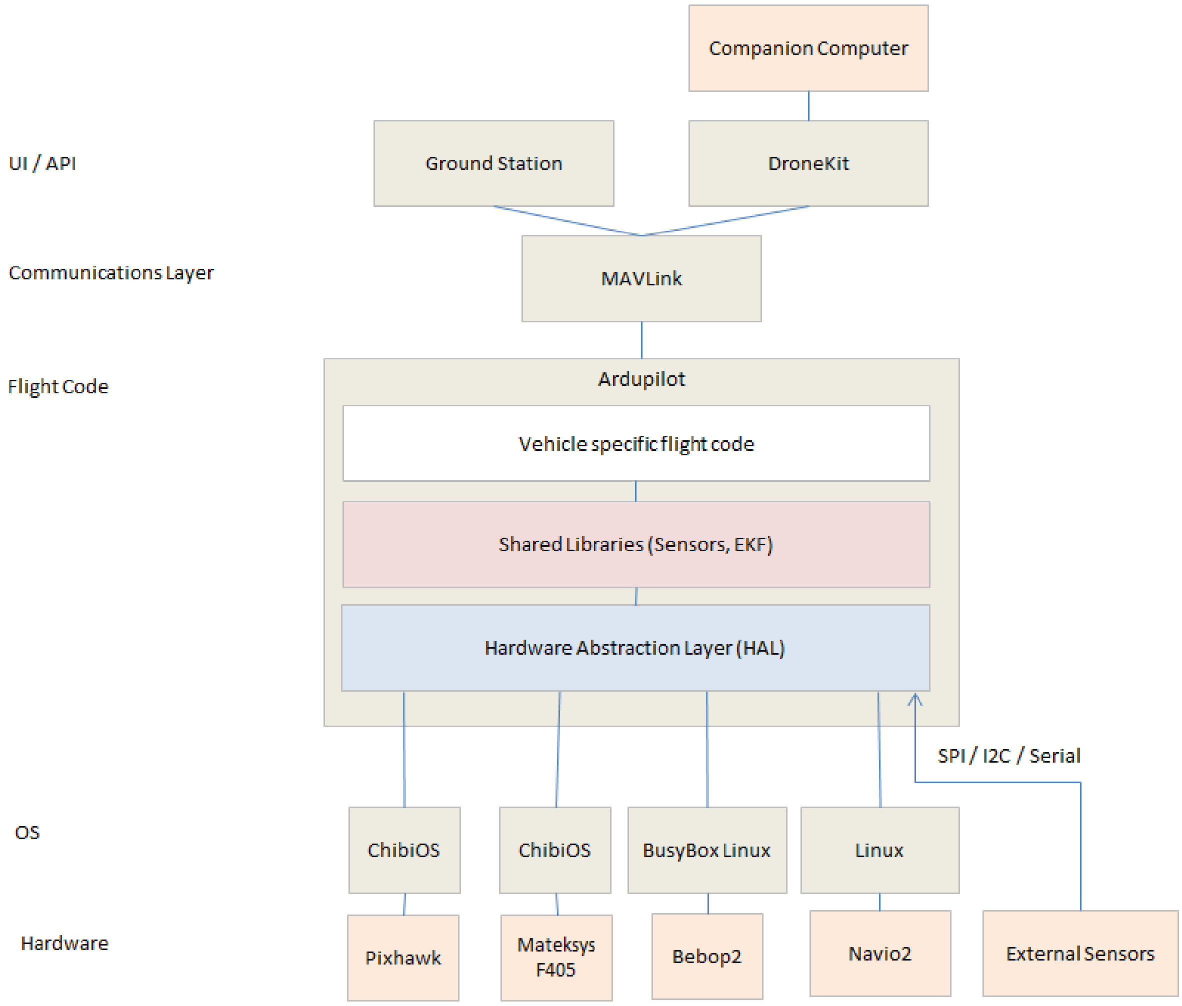
The PDB is a printed circuit board which distributes power to each element of the quadcopter from the battery. When a PDB is chosen for a quadcopter build, the rating of the ESC and the number of them must be considered so that the PDB used can handle the current passed through it. Modern day PDBs regulate can regulate the flow of voltage using voltage regulators, allowing components that are voltage sensitive to receive stable and clean power. This is due to parts such as cameras and video transmitters being present, which require limited voltage. The size and layout of the PDB must also be considered, as the location of the connections in comparison to the components will affect the aesthetics and weight distribution due to the need for cable management. The power wires and the ESC wires are the largest and heaviest and should be taken into consideration first. More recent PDB designs incorporate a hybrid of both the PDB and flight controller as one component.

RC Controller

Quadcopters can be operated by a varied means of user input. The most common method is using a pilot operated RC controller. These controllers send input as radio frequencies which are in turn picked up by the onboard receiver and sent to the flight controller. Each input on the RC controller requires its own individual channel, which maps to a channel on the receiver. The minimum number of channels required for a quadcopter to run is four, one for each of the basic controls needed for a quadcopters flight. These controls are throttle, yaw, pitch and roll. Many RC controllers have built-in displays which relays the FPV back to the pilot when a camera is installed on board. Companies such as DJI have introduced applications which allow quadcopters to be operated using mobile phones for certain models, such as the Mavic Pro 2 and Mavic mini. Enthusiasts have also been successfully in modding their UAV so that it can be controlled using a PC and joystick. Quadcopters are typically operated using the radio frequencies 2.4 GHz and 5.8 GHz, with newer quadcopters being controlled using 5.8GHz. This is because this frequency is less likely to become entangled with other frequencies in the same band when flying in urban areas, as 2.4GHz is the same radio frequency which many wireless computers operate at. As these radio frequency signals are considered loss of sight signals, they are not suitable for long range flight. Instead, LTE Networks are used to control the quadcopter via satellite signals. This presents its own challenges, due to issues such as delay, instability and the constant change of IP address when moving cellular area.

Models and firmware design
One of the most popular firmware's used in quadcopter builds is ArduPilot. The first flight control boards were Arduino based, and to keep up with the demand for increased computational power in quadcopter builds robotics company 3DR began constructing new boards called ArduPilotMegas (APM). This were designed in a co-partnership alongside the ArduPilot software. While ArduPilot no longer associate with 3DR, the term APM has continued to exist when referring to ArduPilot.

Architecture of ArduPilot:

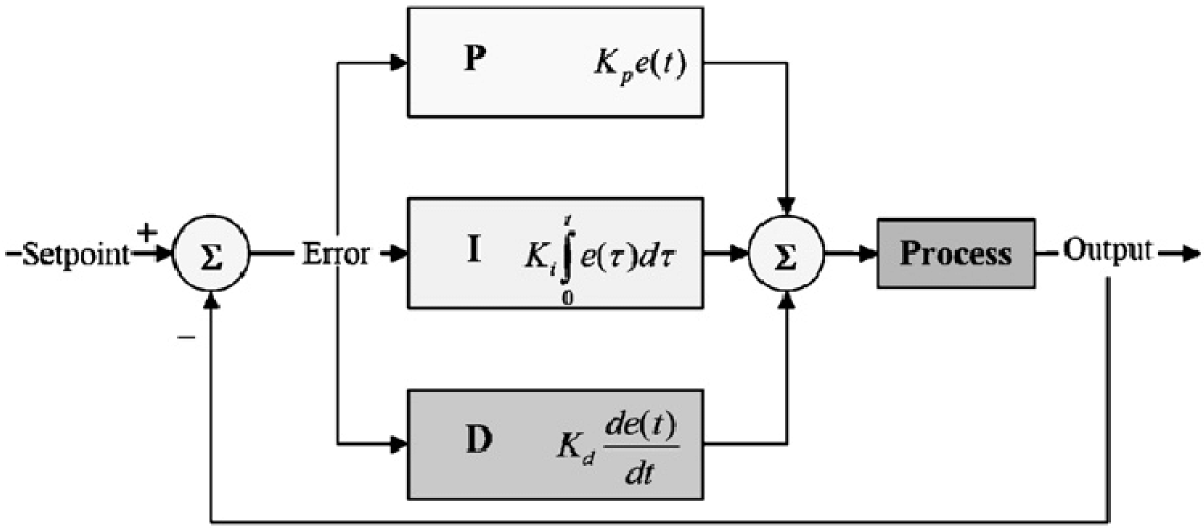


ArduPilot is designed using a multi-layered structure, consisting of three layers. These are the vehicle-specific code, the shared-library code and the hardware abstraction layer, as seen in the above diagram. The hardware abstraction layer, or HAL, is a module which makes the ArduPilot software transferable to many different platforms. The full codebase is comprised of these layers alongside the tool directories and the external support code. The layers communicate through MAVLink, and onto the ground station or drone kit. MAVLink is a protocol for communicating with UAVS and other small vehicles. The ArduPilot's roots in Arduino based architecture is apparent from the setup() and loop() structures used in Arduinos. The ArduPilot's software is multithreaded and has event-driven architecture. ArduPilot is universally applicable to a range of UAVS. The GUI provided by Arduino is a no-party GUI and is not customised to any vehicle. ArduPilot is written in C++, and its development is automated using continuous integration on GitHub. This continuous integration is carried out using an integration service called as Travis CI.

PID Controller

Proportional Integral Derivative control, or the acronym PID, is used to keep a variable or state constant. It uses a control loop feedback to ensure the desired outcome is achieved. It keeps the output constant based on feedback from an input such as a sensor. On a quadcopter, the PID is used for stabilisation and autonomous control. The looptime is the amount of time for the PID to complete a single loop, and modern-day UAVs can have looptimes of up to 32KHz. This equates to a loop every 0.00032 times a second. Each loop, the PID calculates the margin of error from the constant and re-calibrates the device by trying to amend this error. Therefore, some quadcopters can save themselves from becoming grounded when they are thrown off course or even upside down, by making thousands of calculations in and re-adjusting itself before it is too late. The larger the looptime, the sharper the control of the quadcopter. The PID can be calibrated to be sharper or softer depending on the pilots wishes.

PID Diagram:



Each letter of the PID acronym is a value used to correct the error, which is the difference between the value measured by the gyro sensor and the desired rotation speed.

P-Gain
This is the effort which the flight controller puts in to correct error. While sharp control of the quadcopter is usually desirable, if the P-Gain is set too high then it can be too sensitive and very prone to over correcting itself. Too little P-Gain will cause the quadcopter to become sluggish and will be difficult to control.

I-Gain
This sets how much effort the quadcopter puts in to resisting external forces. These forces might include weather related and other physical forces. If the I-Gain is set too high, the quadcopter will become stiff in its control. Too low I-Gain will cause low-frequency oscillation. This will result in the vehicles movement pulsating and sweeping in an undesirable manner.

D-Gain
This reduces over-correction and overshoots by damping the P-Gain. This works in a similar way to car suspension. If the D-Gain is too low, the quadcopter will suffer from whiplash when correcting itself from obstructions, causing the vehicle to become uncontrollable.

ArduPilot loads the quadcopters it is flashed to with default values, however offers advanced tuning options where the PID values can all be optimized as the pilot wishes. It also offers in-flight tuning, where values can be changed in the quadcopter while airborne. Other values which can also be tuned are the roll/pitch, yaw, altitude, loiter and filter.

Simulation
Simulation allows for safe testing and practice before a quadcopter is physically flown. ArduPilot uses a Flight Dynamics Model to simulate a UAVs flight. It uses a software called SITL which simulates the inputs and outputs and runs the ArduPilot firmware just as a physical quadcopter would. The virtual quadcopter is controlled using a device such as a gamepad or joystick, or a ground control station (GCS). A GCS is an application which communicates with the quadcopter using telemetry and a computer, providing real-time data with regards the quadcopters position and performance. It is sometimes called a virtual cockpit, as it includes many of the devices that would be present on a real plane. A popular GCS used in conjunction with ArduPilot is Mission Planner. Mission Planner is built for Mac and Windows operating systems and is open source. It supports autopilot for quadcopters and has features such as point-and-click-waypoint, mission commands from drop-down menus and runs its own SITL simulation for all ArduPilot vehicles.