A Project report submitted as partial fulfillment for the award of

BACHELOR OF SCIENCE

IN

COMPUTER HARDWARE

By

V . SHIVA SHANKAR (21-106-022) ,

A . MOUNIKA (21-106-003) ,

K . SWATHIKA (21-106-017),

K . JYOSHITHA (21-106-018) ,

S . MAMATHA (21-106-004) ,

S . RUCHITHA (21-106-011) ,

Submitted to

Department of Computer Hardware

Nizam College, Osmania University

Hyderabad. March 2024

SUPERVISE

By

A. Venu Gopal Rao

House No 2-2-1093/A/6, Golnaka, Amberpet, Hyderabad – 500013
Mob: 9849078120
Email: contact@replicaminiatures.com, replicaind@gmail.com

Satellite controlled model Bombing Drone

CERTIFICATE

Date:30-03-2024

This is to certify that the project work entitled satellite controlled model Bombing Drone Submitted by **V. SHIVA SHANKAR (21-106-022), A. MOUNIKA (21-106-003), K. SWATHIKA (21-106-017), K. JYOSHITHA (21-106-018), S. MAMATHA (21-106-004), S. RUCHITHA (21-106-011) ,** the students of Nizam College for partial fulfillment of the course for the award of the **BACHELOR OF SCIENCE** in **COMPUTER HARDWARE** is a record of bonafide work carried out by them, with our technical support during January1st to March27.

Their attendance and performance during the period was good.

(Project Manager External)

ABSTRACT

A Drone can achieve vertical flight in a stable manner and be used to monitor or collect data in specific region. Such as medical needs, photography, emergencies. Advances in the performances of the low power micro controllers atmel 2560,32-bit ARM Processor,168 Mhz/256 KB RAM/2 MB Flash,32-bit failsafe co-processor, Sensors like Gyroscope, Temperature, magnetic and Accelerometer, that allowed students to develop their drones.

In this project we accommodated the possible features into the hex copter to bring out any specific application in the future.

ACKNOWLEDGEMENT

We would like to express our sincere thanks and gratitude To our principal Prof.B.Bheema and Head of the Department Dr.Kaleem Ahmed Jaleeli, Department of physics, NIZAM college and Replica miniatures for having guided us in developing the requisite capabilities for taking up this project.

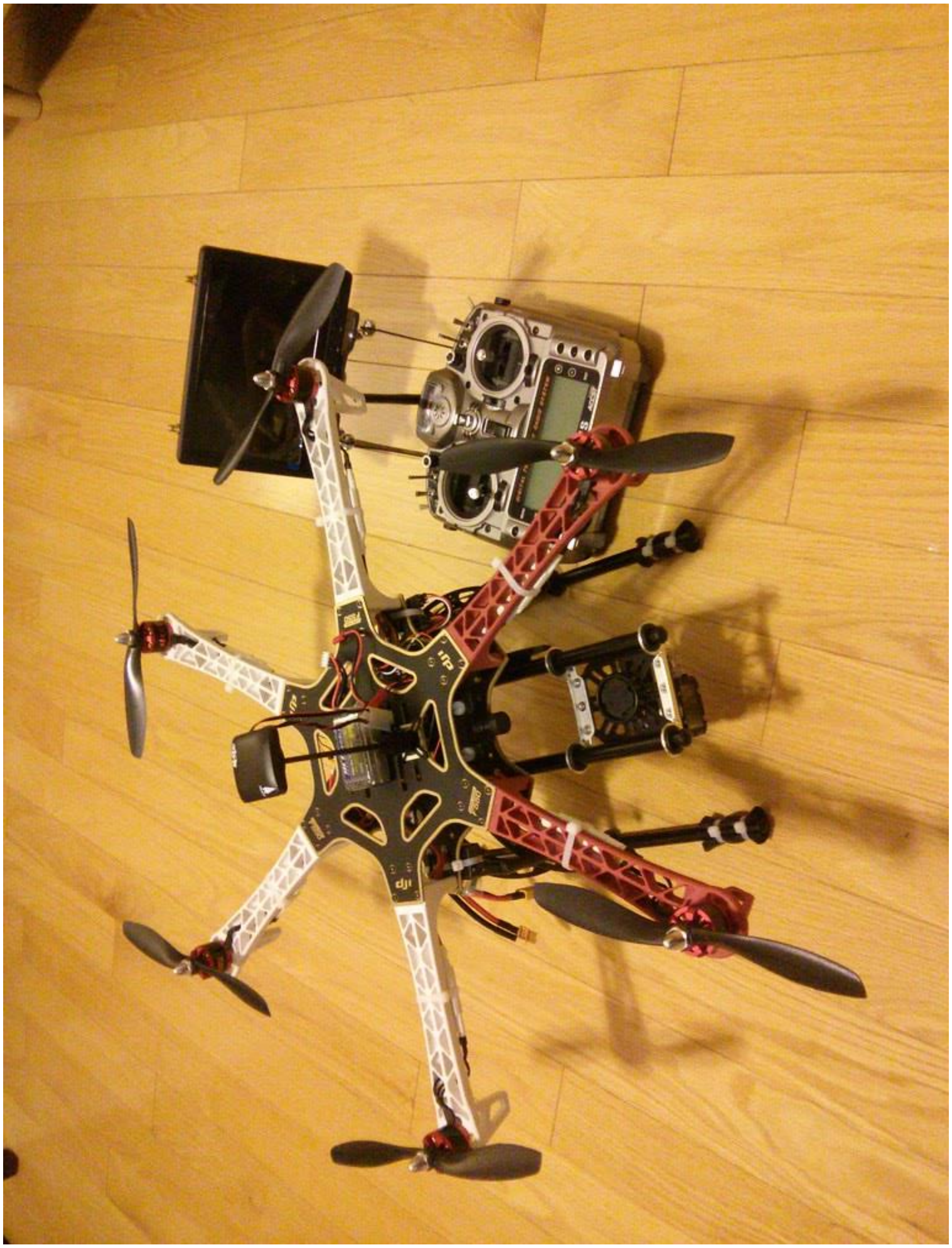
We thank Asst. Prof. A. Venu Gopal Rao, project Co-Ordinator, Computer Hardware for providing seamless support and Right suggestions are given in the development of the project.

We specially thank our internal guide Mrs.A. Mahija Asst.Prof For her suggestions and constant guide in every stage of the project.

On a more personal note we thank our beloved parents for Their moral support during the course of the project.

Table of Contents: (Pgno)

SATELLITE CONTROLLED MODEL BOMBING DRONE



MICROCONTROLLERS

There are different types of microcontrollers.

1.8052 microcontrollers

2.PIC microcontrollers

3.ARM microcontrollers

4.AVR microcontrollers

1.8052 Microcontrollers:

A. 8051 Microcontroller:

Features of 8051:

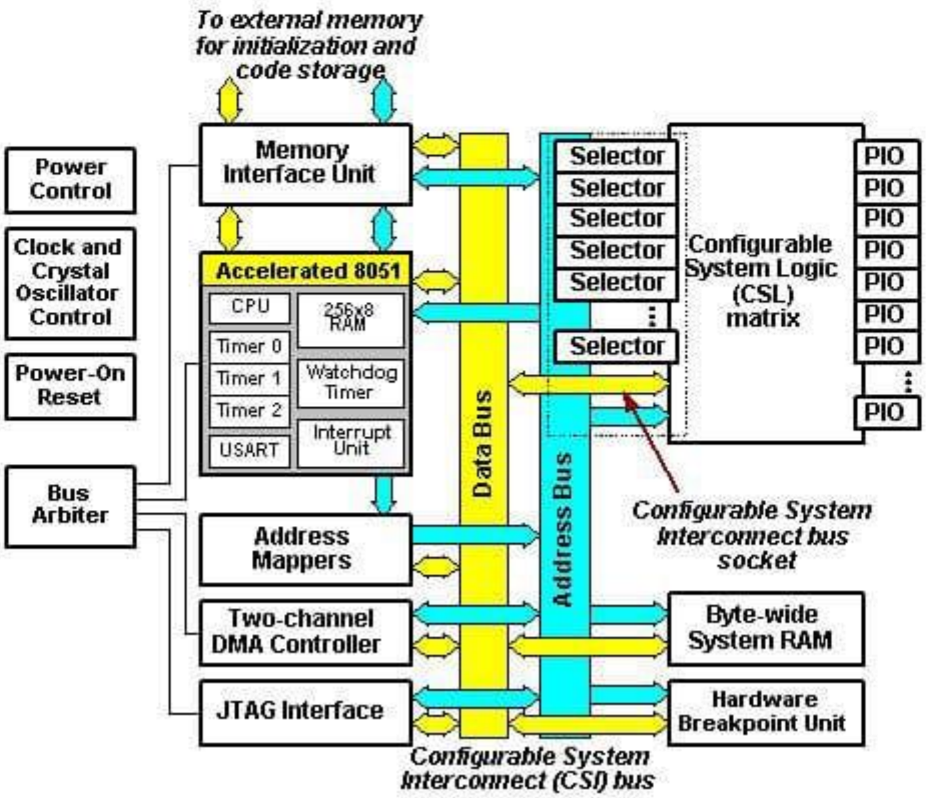
4KB ROM (program memory),128 bytes RAM (data memory),8bit data bus (bidirectional), 16-bit address bus (unidirectional), Two 16-bit timers, Four 8-bit input/output ports, It has 16-bit program counter and data pointer, It has three internal interrupts and two external interrupts.

B. 8052 Microcontroller:

Features of 8052:

8-bit CPU,4KB ROM,128 bytes of RAM,
32 I/O lines, Timer/Counters, Serial
Communication, Interrupts, ADC Support,
Low power modes, External memory interface.

Block diagram of 8052 microcontrollers:



1. PIC Microcontrollers:

A. PIC 16F628:

It’s commonly used in a wide range of applications including Embedded systems, consumer electronics, automotive, industrial Control, and more.

Features and Specifications:

Operating Voltage:2.0V to 5.5V, Flash Program Memory:3.5 KB, RAM Data Memory:224 bytes, EEPROM Data Memory:128 bytes, CPU Speed: Up to 20MHz, I/O Pins:16, Timers:2, Analog-to-digital Convertor (ADC): 8, Channels with 10-bit resolution, USART/UART Module for serial communication, Twocapture/compare/PWMM (CCP) modules, Low power consumption, Wide operating temperature range.

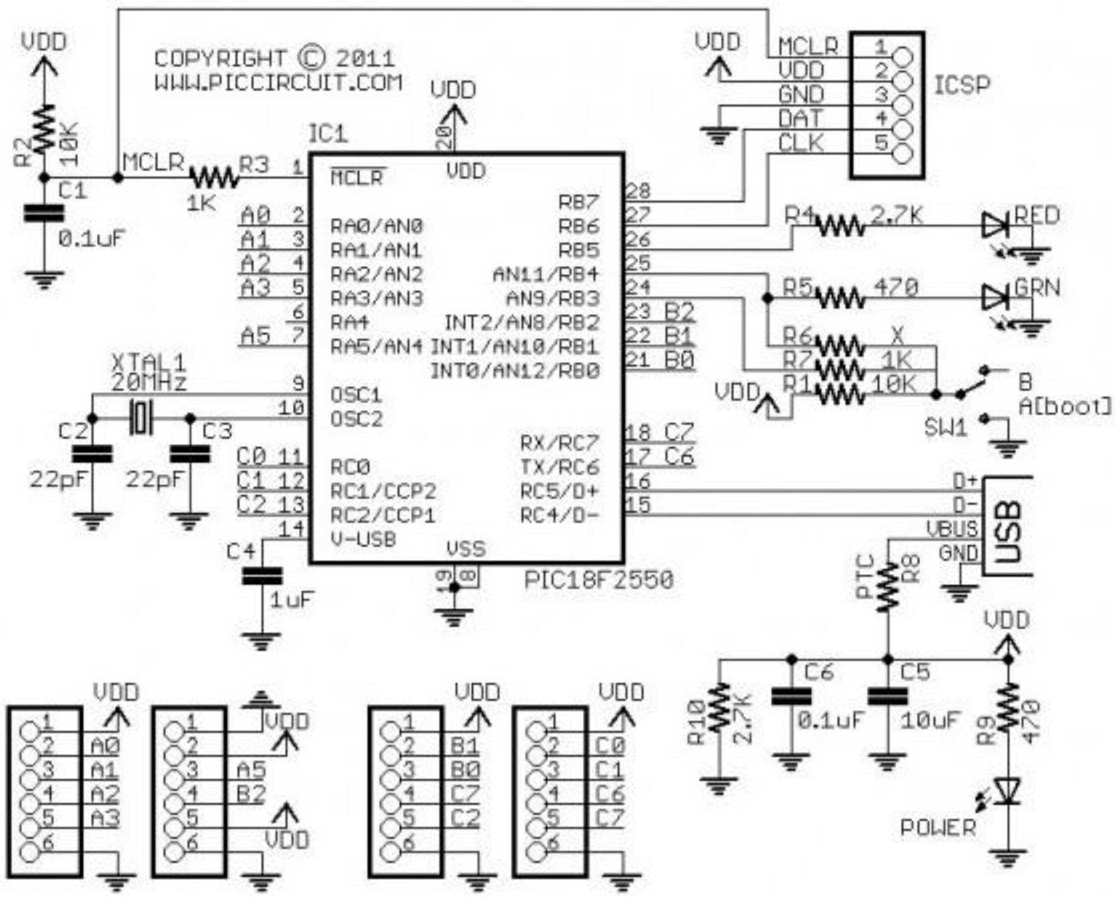
B. PIC 18F2550:

It’s widely used in various applications such as USB peripherals, industrial Control, automation, and consumer electronics.

Features and Specifications:

Processor: 8-bit RISC architecture,Clock speed: Typically operates at up to 48MHz,
Memory: 32KB Flash program memory, 2KB RAM and 256 bytes of ROM, Peripheral Features: Analog-to-digital convertor (ADC) with up to 10 Bit, Multiple timers including timer0, timer1, timer2 and timer3, I/O Ports: Multiple GPIO pins for interfacing with external components, Power management: Low power sleep and idle modes for power efficiency, Operating Voltage: Available in various package options, including PDIP.SOIC, and QFN.

PIC 18F2550 DIAGRAM:



3.ARM Microcontrollers:

ARM microcontrollers are divided into 3 types they are:
*ARM-Cortex Ax-series.
*ARM-Cortex Rx-series.
*ARM-Cortex Mx-series.

ARM-Cortex Ax-series:

The ARM Cortex-Ax-series is a family of processors designed for various applications, including smartphones, tablets, automotive, and IoT devices.

Features and specifications:

Performance, Instruction Set Architecture, Multi-core Support, Cache Hierarchy, Clock speed, Power efficiency,

ARM-Cortex Rx-series:

The ARM Cortex-Rx series features high performance, real-time processing capabilities,
With features optimized for safety-critical applications.

Features and specifications:

Real-time processing, High performance, Scalability, Peripheral support, Low-latency interrupts, Development tools.

1.**ARM-Cortex Mx-series:**

The ARM-Cortex M-series of processors are designed for embedded systems, with a focus on low power consumption and high performance.

Features and specifications:

Architecture, Performance, Clock speed, Memory, Peripherals, Power consumption.

AVR Microcontrollers:

AVR 328P:

The ATmega328P is an 8-bit microcontroller from the family, commonly used in Arduino boards.

Features and specifications:

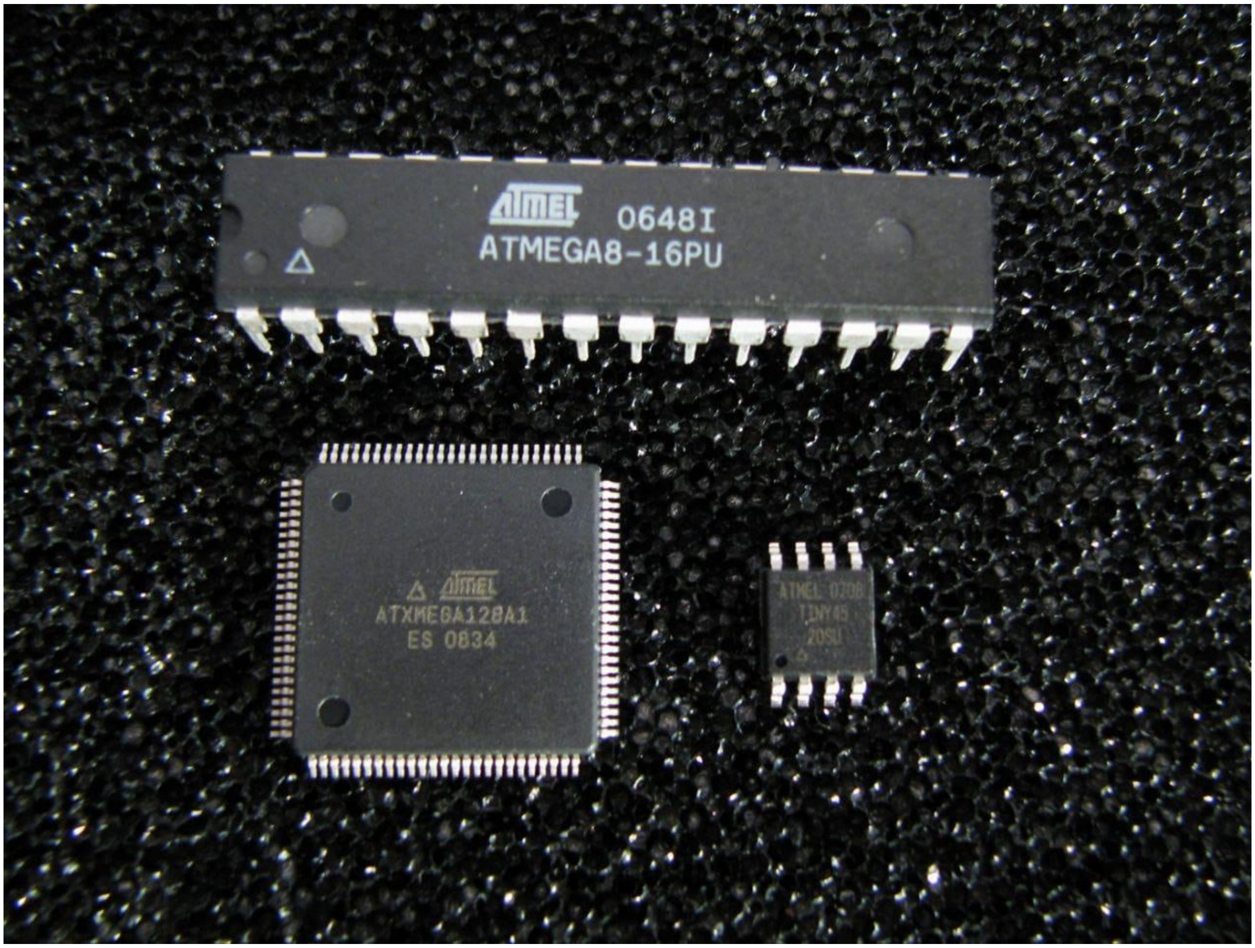
Architecture: 8-bit RISC, **Flash Memory:** 32KB (of which 0.5 KB is used by the bootloader),**SRAM**: 2 KB , **EEPROM**: 1 KB, **Clock speed:** Up to 20MHz, **Digital I/O Pins:** 23 (6 of which provide PWM output), **Analog Input pins**: 6, **Serial Communication:** UART, SPI, I2C, **Timers:**3 (with 16-bit resolution), **Comparators:** 1, **Operating Voltage:** 1.8V to 5.5V.

AVR2560:

Features and specifications:

Architecture: 8-bit RISC, **Flash Memory:** 256KB, **SRAM**: 8 KB, **EEPROM**: 4 KB, **Clock speed:** 16MHz, **Digital I/O Pins:** 54 (of which 15 can be used as PWM output), **Analog Input pins**: 16, **Operating Voltage**: 1.8V to 5.5V, **Serial communication:** USART, SPI, I2C.

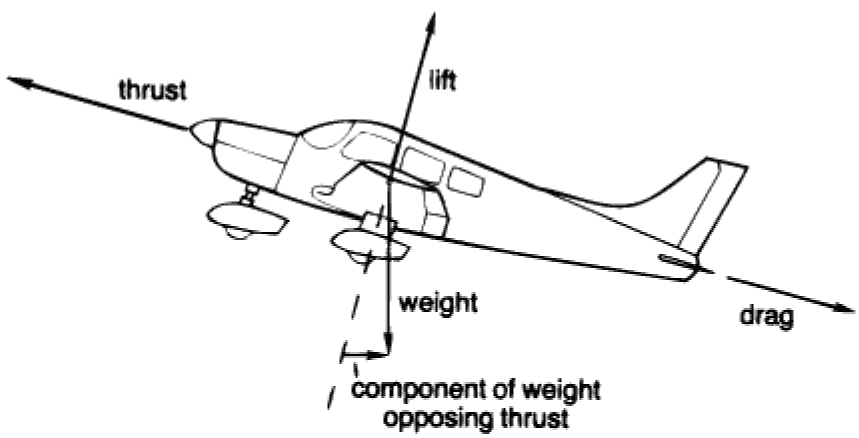
Diagram of AVR microcontrollers:



PRINCIPLE OF AURONAUTICS

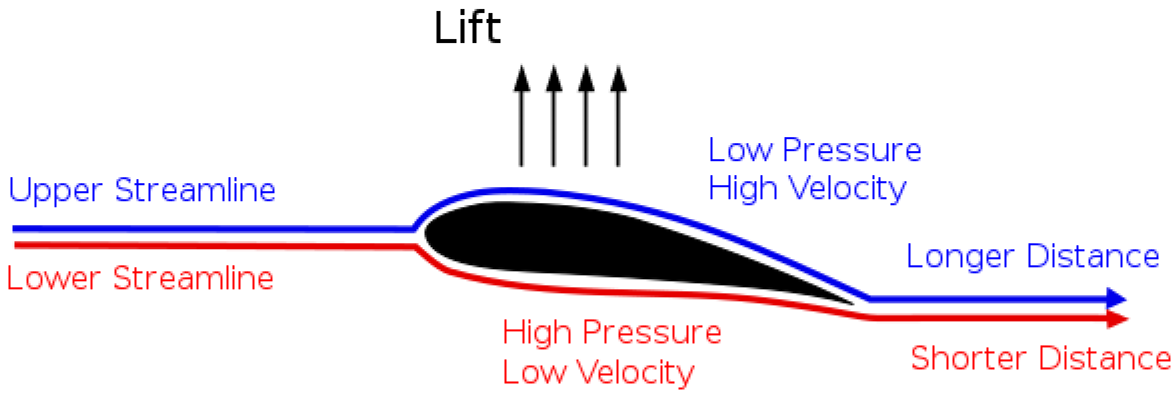
There are certain laws of nature or physics that apply to any object that is lifted from the Earth and moved through the air. Among the aerodynamics forces acting on a airplane during all maneuvers. These basic forces are:

- **Lift**
- **Weight(gravity)**
- **Thrust**
- **Drag**



LIFT

Lift is the upward forces created by an airfoil. When it is moved through the air. Although may be exerted to some extended to some extent by many external parts of the airplane , there are three principal airfoils on airplane the wing , propeller, and horizontal tail surface.

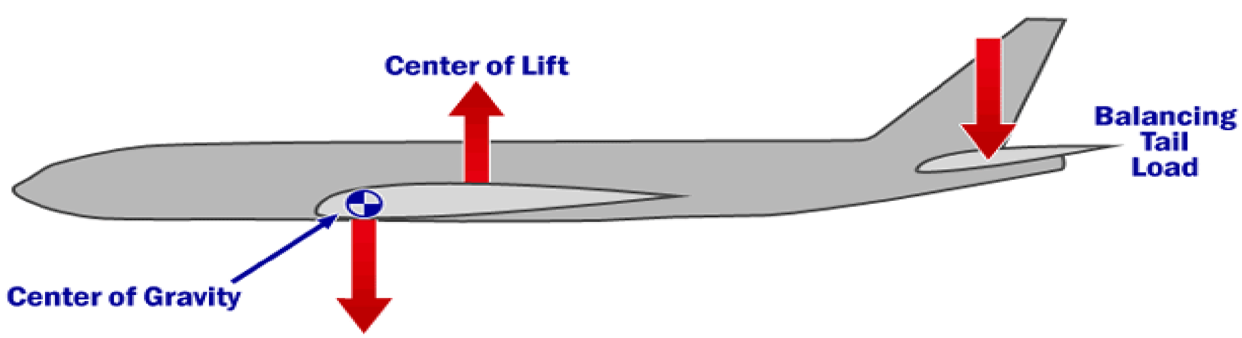


The amount of lift generated by the wing depends upon several factors:

- Speed of the wing through the air
- Angle of attack
- Planform of the wing
- Wing area, and
- the density of the air.

GRAVITY(Weight)

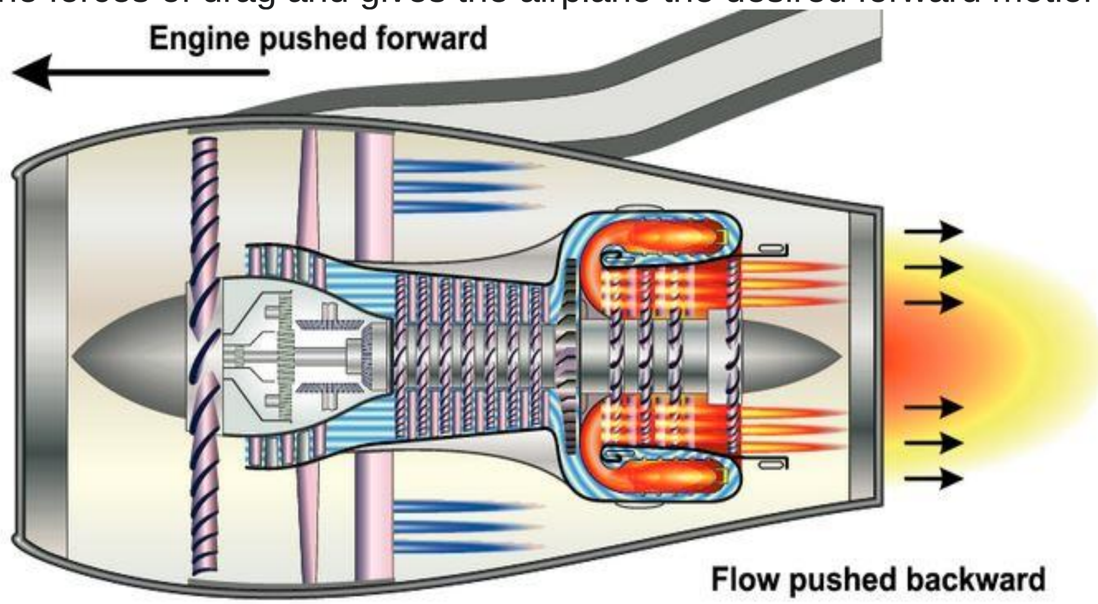
Gravity is the downward forces which tends to draw all bodies vertically towards the center of the earth. The airplane's center of gravity is the point on the airplane at which all weight is considered to concentrated.



The center of gravity is located along the longitudinal centerline of the airplane and somewhere near the center of lift of the wing . The location of the center of the gravity depends upon the location and weight of the load placed in the airplane. this is controlled through weight and balance calculation made by the pilot prior to flight . the exact location of the center of gravity is important during flight, because of its effect on airplane stability and performance.

THRUST

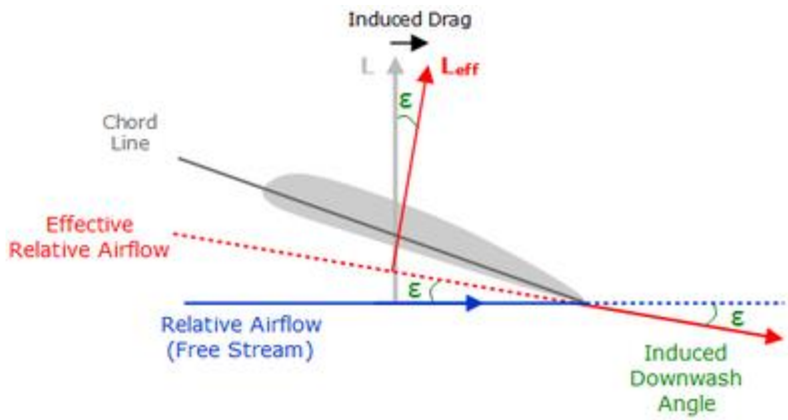
The propeller, acting as an airfoil, produces the thrust ,or forward force that pulls the airplane through the air. it receives its power directly from the engine , and is designed to displace a large mass of air to the rear. It is the rearward displacement that develops the forward thrust that carries the airplane through the air. This thrust must be strong enough to counteract the forces of drag and gives the airplane the desired forward motion.



DRAG

Drag is rearward acting forces which resists the forward movement of the airplane through every part of the airplane which is exposed to the airplane in motion some resistance and contributes to the drag. Total drag may be classified into main types:

- **Induced Drag**
- **Parasite Drag**



Mono-copter or **gyrocopter** is a rotorcraft that uses a single rotating blade. The concept is similar to the whirling helicopter seeds that fall from some trees. The name **gyrocopter** is sometimes applied to monocopters in which the entire aircraft rotates about its center of mass as it flies.

"**Duo-copter**", "Bi-copter" and "Dual-copter" have their origins in the Latin and mean "two", "twice" and "two containing". **One engine left, one right. Both arms can be swiveled. Unbalanced effect of thrust**.

Tri-copter as its name suggests, is made up of three rotors to generate buoyancy and is used for movement and control. And the arm distance is usually 120 degrees, at the meantime, it's typically in a “Y” shape and sometimes in a "T" shape.

Quadcopter needs a frame to house all the other components. Things to consider here are weight, size, and materials. They are strong, light and have a sensible configuration including a built-in power distribution board (PDB) that allows students and enthusiasts to develop their own multi copters for a clean and easy build.

Hexacopter is a member of multi-rotor flying devices which consist of multiple fixed rotors attached to a simple mechanical construction. **Hexacopter** itself consists of 6 six rotors with fixed pitch blades. The main property of such devices is their instability.

Hexacopter has many advantages and is suitable for beginners as well as for demanding photographers and videographers due to its safe and stable flight characteristics. Even commercial drone pilots will find a safe, precise, powerful and flexible tool in it that can be equipped with additional accessories thanks to the increased load capacity.

MultiCopters:

Fig 1.1

Fig 1.2

Tricopter

Quadcopter

Fig 1.3

Fig 1.4

Hexacopter

Octcopter

Hexacopter Parts List:

- **Frame**
- **Motors**
- **Pixhawk**
- **Electronic Speed Control (ESC) x6**
- **Flight Control Board**
- **Radio transmitter and receiver**
- **Propeller x6 (3 clockwise and 3 counter-clockwise)**
- **Battery & Charger**

Multi copter Frames:

Fig 2.0

DJI Flame Wheel F450 Quadcopter

Fig 2.1

DJI Flame Wheel550 Hexacopter

Propeller directions:

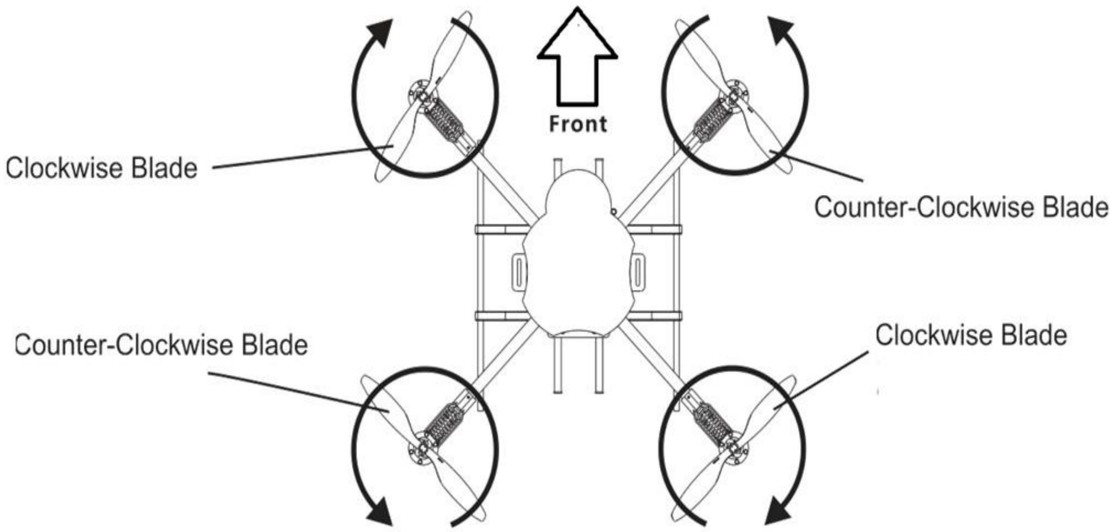


Fig 2.2

Propeller directions for Quadcopter

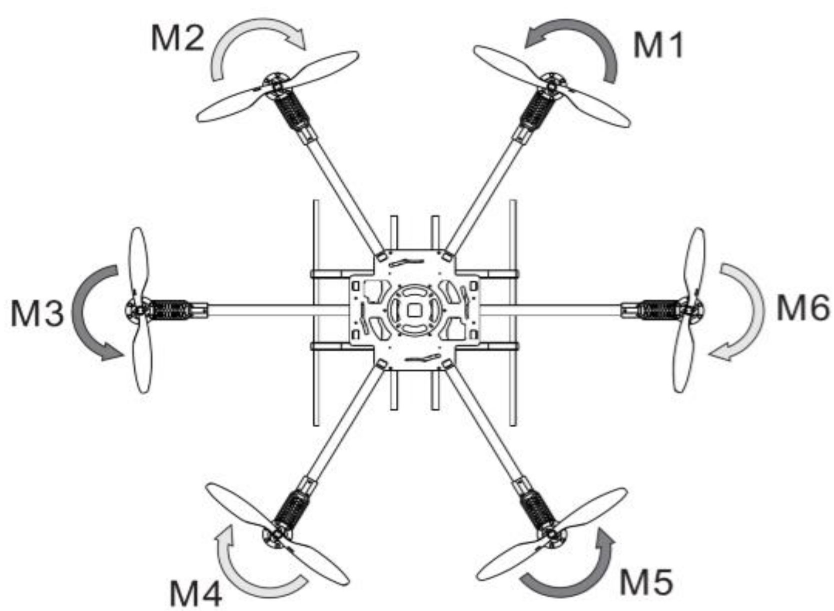


Fig 2.3

Propeller directions for Hexa copter

Motors:

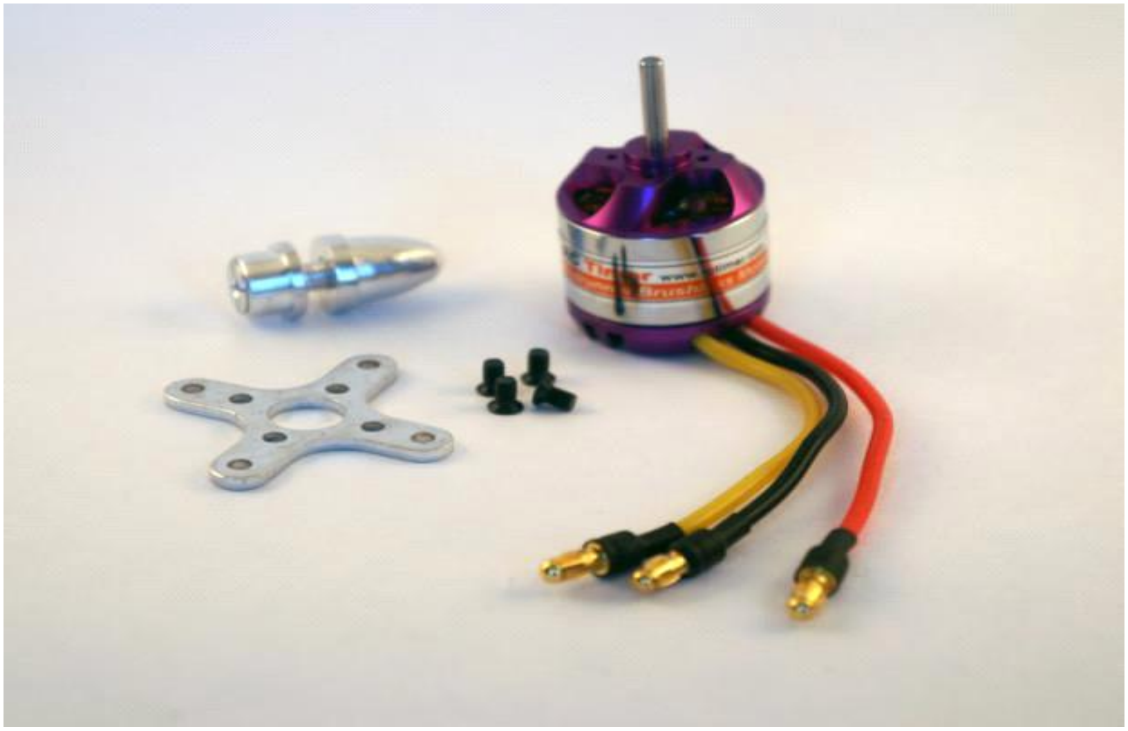


Fig 3.0

Types of Motors:

1. Brushed Motor:

The brushed DC motor is the classic motor that is used in applications like motorized toys, appliances, and computer peripherals. This type of motor is inexpensive, efficient, and especially useful for providing high speed and power in a relatively small package.

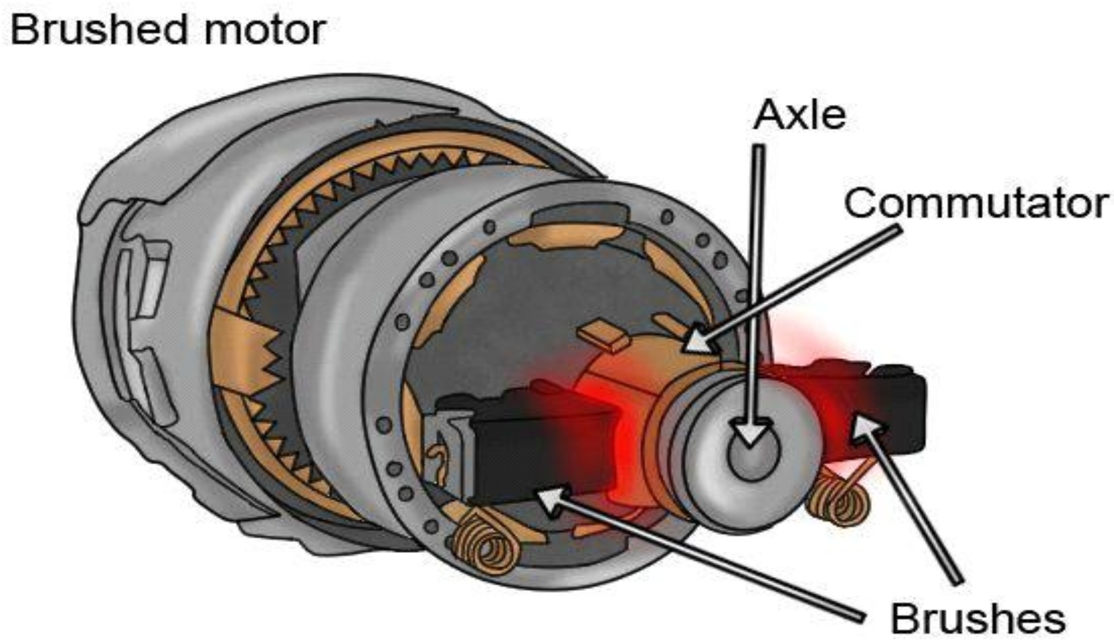


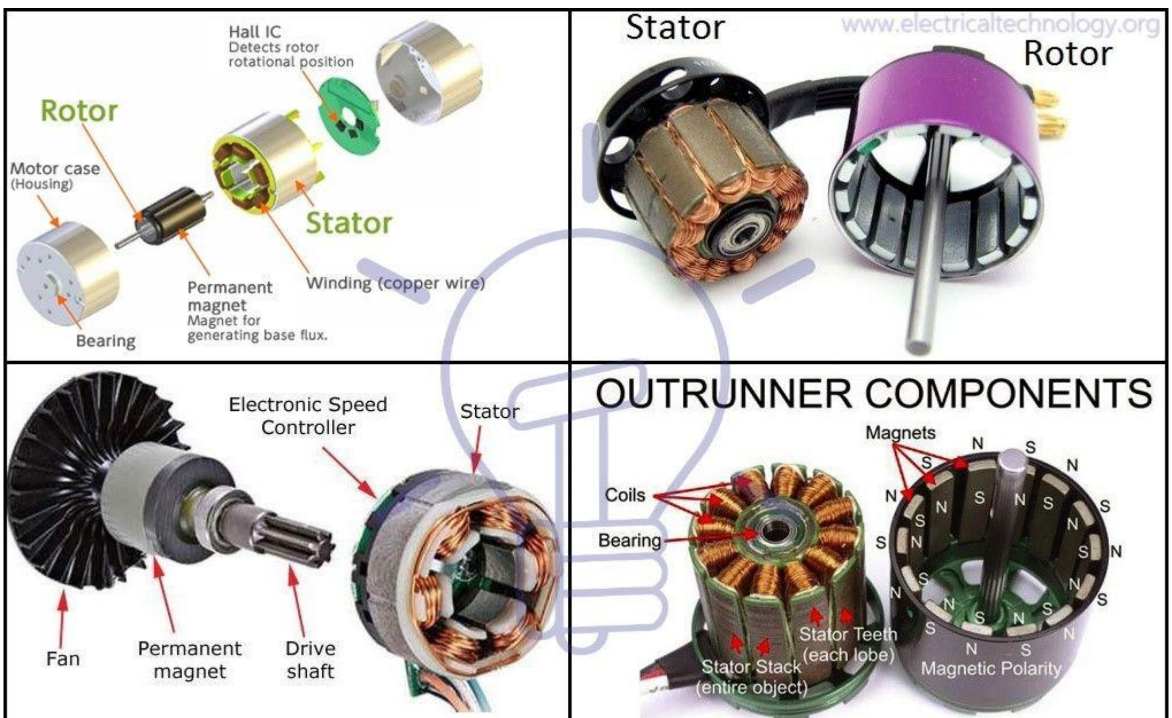
Fig 3.1

2. Brushless Motor:

The Brushless DC (BLDC) motor is the ideal choice for applications that require high reliability, high efficiency, and high power-to-volume ratio. Generally speaking, a BLDC motor is considered to be a high performance motor that is capable of providing large amounts of torque over a vast speed range. BLDC motors are a derivative of the most commonly used DC motor,

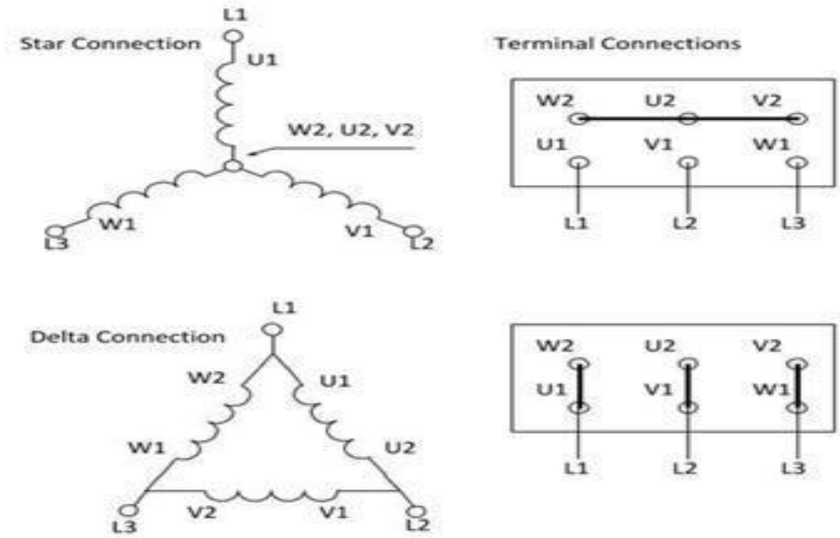
The brushed DC motor, and they share the same torque and speed performance curve characteristics. The major difference between the two is the use of brushes. BLDC motors do not have brushes (hence the name “brushless DC”) and must be electronically commutated.

Open parts of bldc:



Construction of BLDC Motor

Configuration of BLDC Motor:



At a given time two wires are switched ON and the third coil act as a speed detector. The speed detector output is feeded to ESC(Electronic Speed Control) to the analog input such that this signal is used control of the speed of the motor.

Electronic Speed Control:

1.RCTimer 30A SimonK ESC

Fig 4.0

The electronic speed control, or ESC, is what tells the motors how fast to spin at any given time. You need four ESCs for a quad copter, one connected to each motor. The ESCs are then connected directly to the battery through either a wiring harness or power distribution board. Many ESCs come with a built in battery eliminator circuit (BEC), which allows you to power things like your flight control board and radio receiver without connecting them directly to the battery. Because the motors on a quad copter must all spin at precise speeds to achieve accurate flight, the ESC is very important.

2. Lumenier BLHeli_32 32bit 23A 4in1 ESC 2-4s BEC DSHOT.

Fig 4.1

The 4-in-1 ESC is low profile and compact in size, saving weight and making builds easy. This ESC includes a powerful on board 5v 3A BEC as well as a 5v 1A BEC. The ESC features new BLHeli_32 firmware pre-installed on them.

3. Parallax xRotor 20A Electronic Speed Controller.

Fig 4.2

These Parallax rotors 20A ESCs (Electronic Speed Controllers) are designed specifically for multi-rotor aircraft, and further customized for the ELEV-8 v3 Quad copter Kit.

- Pre-soldered bullet connectors for quick, solder-free quad copter assembly and repair
- Pre-loaded with industry-leading BLHeli firmware configured for the ELEV-8 v3 quad copter
- Re-programmable and adjustable for custom applications
- Regenerative breaking prolongs battery life for longer flights.

ESC schematic:

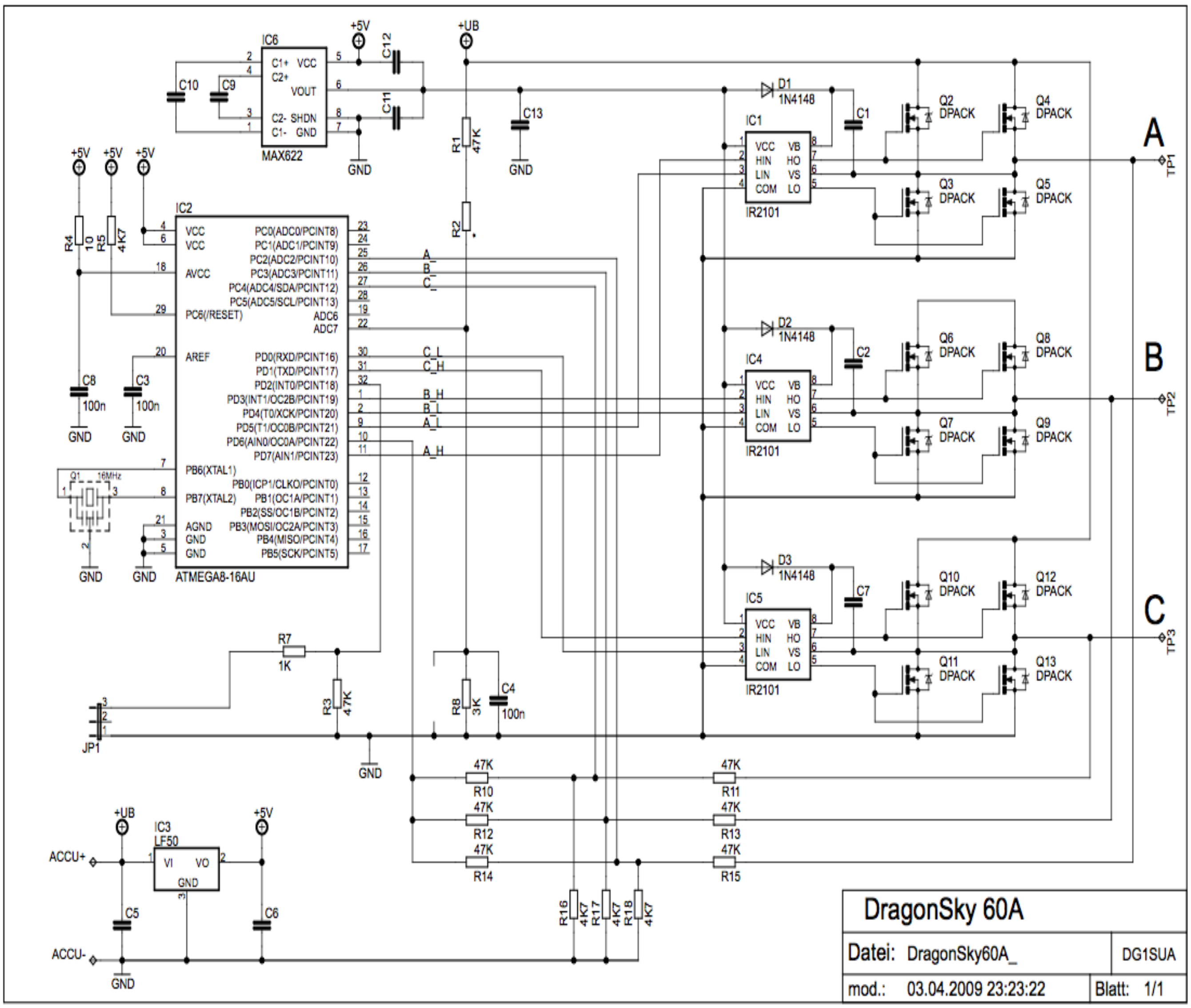


Fig 4.3

ESC layout: Front end.

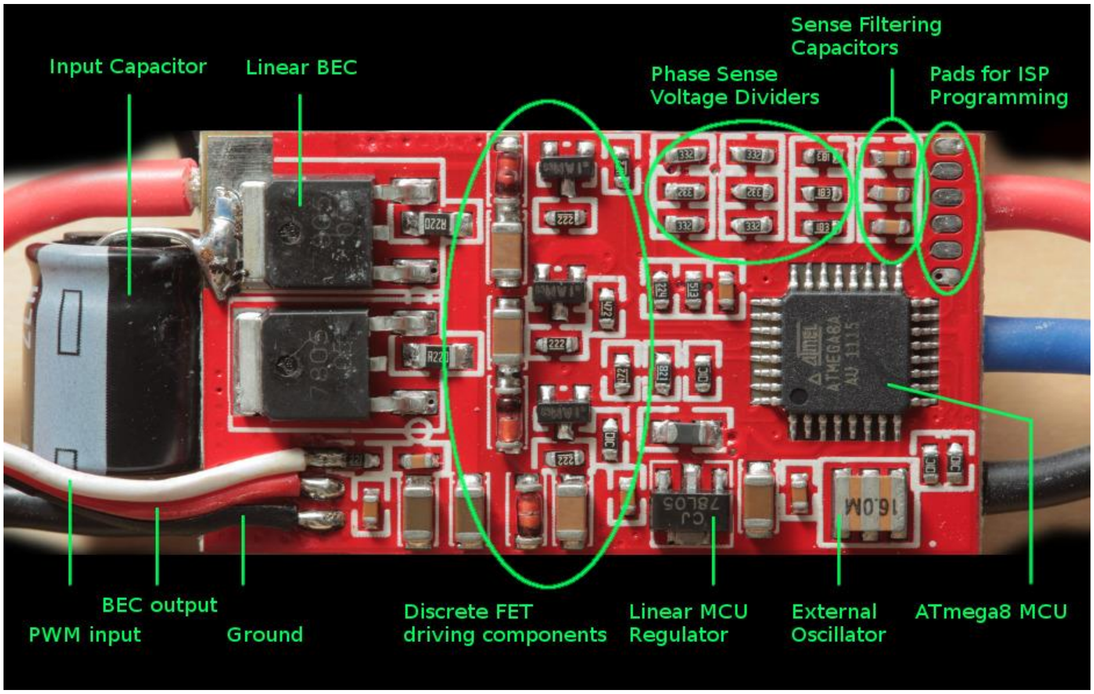


Fig 4.4

ESC layout: Back end.

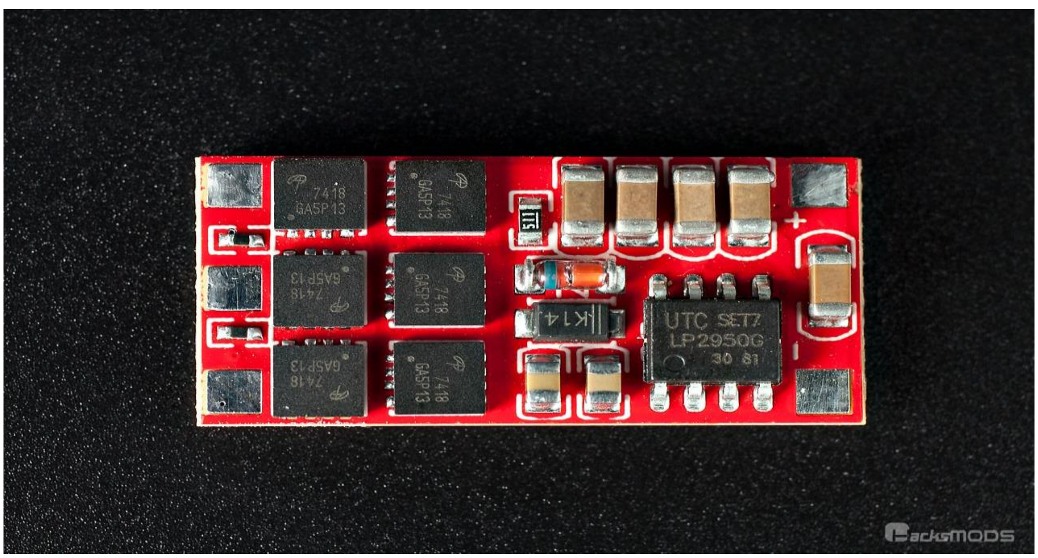


Fig 4.5

Flight Controllers

Pixhawk:

Pixhawk is an independent open-hardware project that aims to provide the standard for readily-available, high-quality and low-cost autopilot hardware designs for the academic, hobby and developer communities. Pixhawk supports multiple flight stacks: PX4 ® and ArduPilot ®.

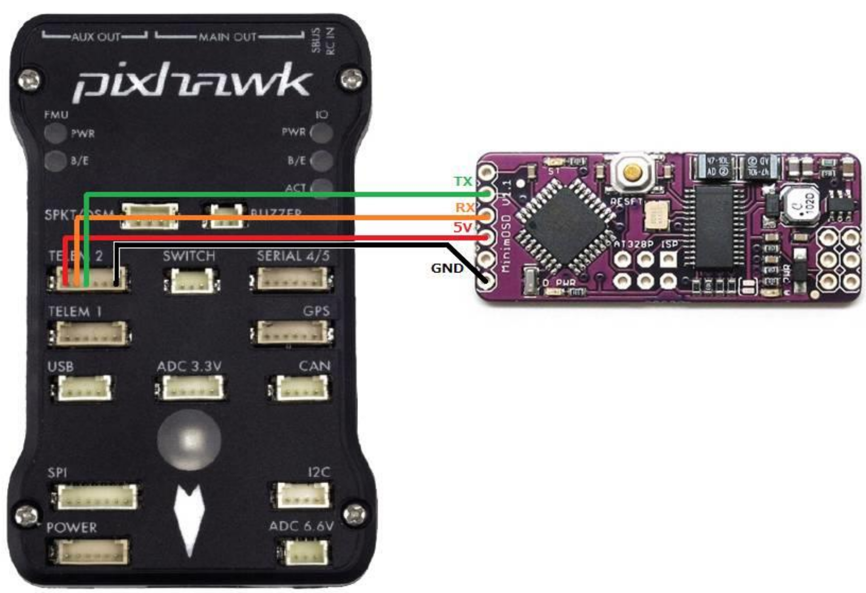
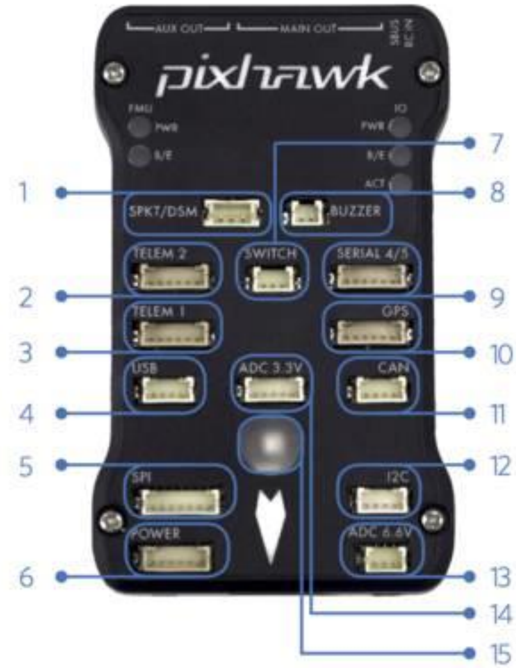
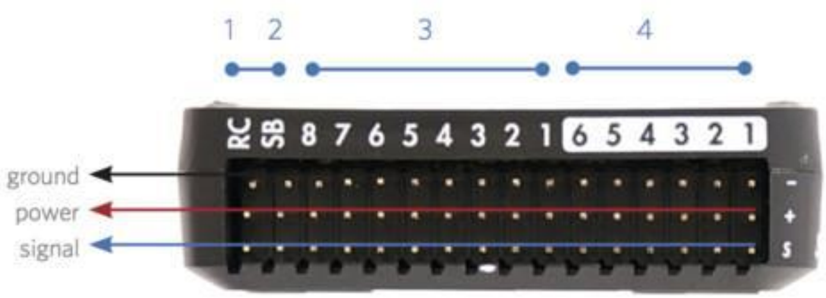


Fig 5.0



1 Spektrum DSM receiver
2 Telemetry (on-screen display)
3 Telemetry (radio telemetry)
4 USB
5 SPI (serial peripheral interface) bus
6 Power module
7 Safety switch button
8 Buzzer
9 Serial
10 GPS module
11 CAN (controller area network) bus
12 I²C splitter or compass module
13 Analog to digital converter 6.6 V
14 Analog to digital converter 3.3 V
15 LED indicator

1 Input/output reset button
2 SD card
3 Flight management reset button
4 Micro-USB port



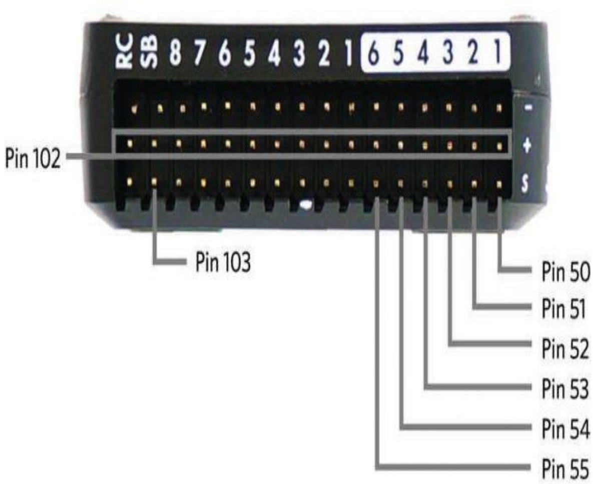
1 Radio control receiver input
2 S.Bus output
3 Main outputs
4 Auxiliary outputs

Fig 5.1

The Pixhawk flight controller:

A decade ago, “Pixhawk” was a student project at ETH Zurich. Little did the team know that it would end up having such a large influence on the drone industry .During this time, the team created MAVLink, Pixhawk, PX4, and Q Ground control — which are today’s most used standards for flight control hardware and autopilot software in the drone industry. This was the beginning of a story of a very successful open source project outperforming individual corporate development.

Pixhawk pin configuration:



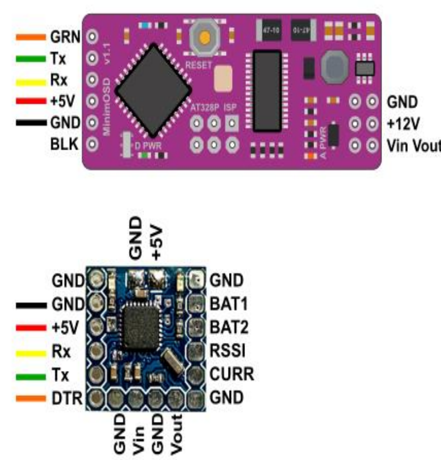


Fig 5.2

Input pin configuration of pixhawk **Telemetry**

Pixhawk configuration in drone:

This chart Illustrates a quadcopter with a variety of options.

- A 4 in 1 ESC is used instead of individual ESCs reducing EMF and allowing a neater installation and no power distribution board.
- Actual Motor number to Quadcopter frame location is shown below and will require appropriate signal and motor wire routing.

- A 3DR PPM-Sum encoder with conventional receiver is illustrated, A PPM-Sum receiver could be used instead.
- Generally a 3S or 4S LiPo battery is appropriate for a Quadcopter and the XT60 connectors shown are a reliable choice.
- An optional battery warning indicator can provide quick battery status and provide an audible warning when nearby.
- Optional 3DR telemetry radios may be connected to your Pixhawk and tablet or laptop to permit real time monitoring and operation.
  - A (GCS (Ground Control Station) program like Mission Planner, AndroPilot or DroidPlanner provides in flight information and control.

  An optional GoPro camera and brushless gimbal are illustrated and can provide very high quality photo and video results.

-
  - It is quite reasonable to use a GoPro camera and brushless gimbal on 400 and larger Quadcopters.
  - Brushless camera gimbals perform more smoothly and adequately than the older servo type gimbals.

- The brushless gimbal illustrated above has an anti-vibration mount which is important for good photo results.

- For cameras larger than GoPro or “Sport” types, a larger QuadCopter, HexaCopter or OctoCopter should be used.

You could also install an optional FPV (First Person View) video system but it is not shown in the illustration above.

- An FPV system consists of a tiny video camera and transmitter on the copter plus a receiver and video goggles or an LCD screen.

- Almost any size Quadcopter can support a First Person View system.

- An optional OSD (On Screen Display) can add a video heads up display of current in flight conditions to your FPV flight display.

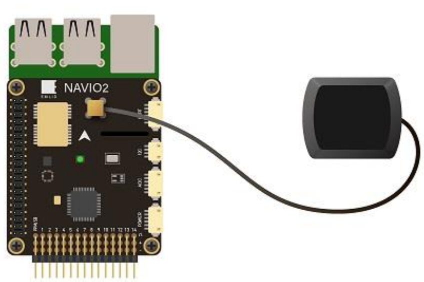
The telemetry, camera, gimbal, FPV and OSD options mentioned above can be combined as you wish for your copter but start simple.

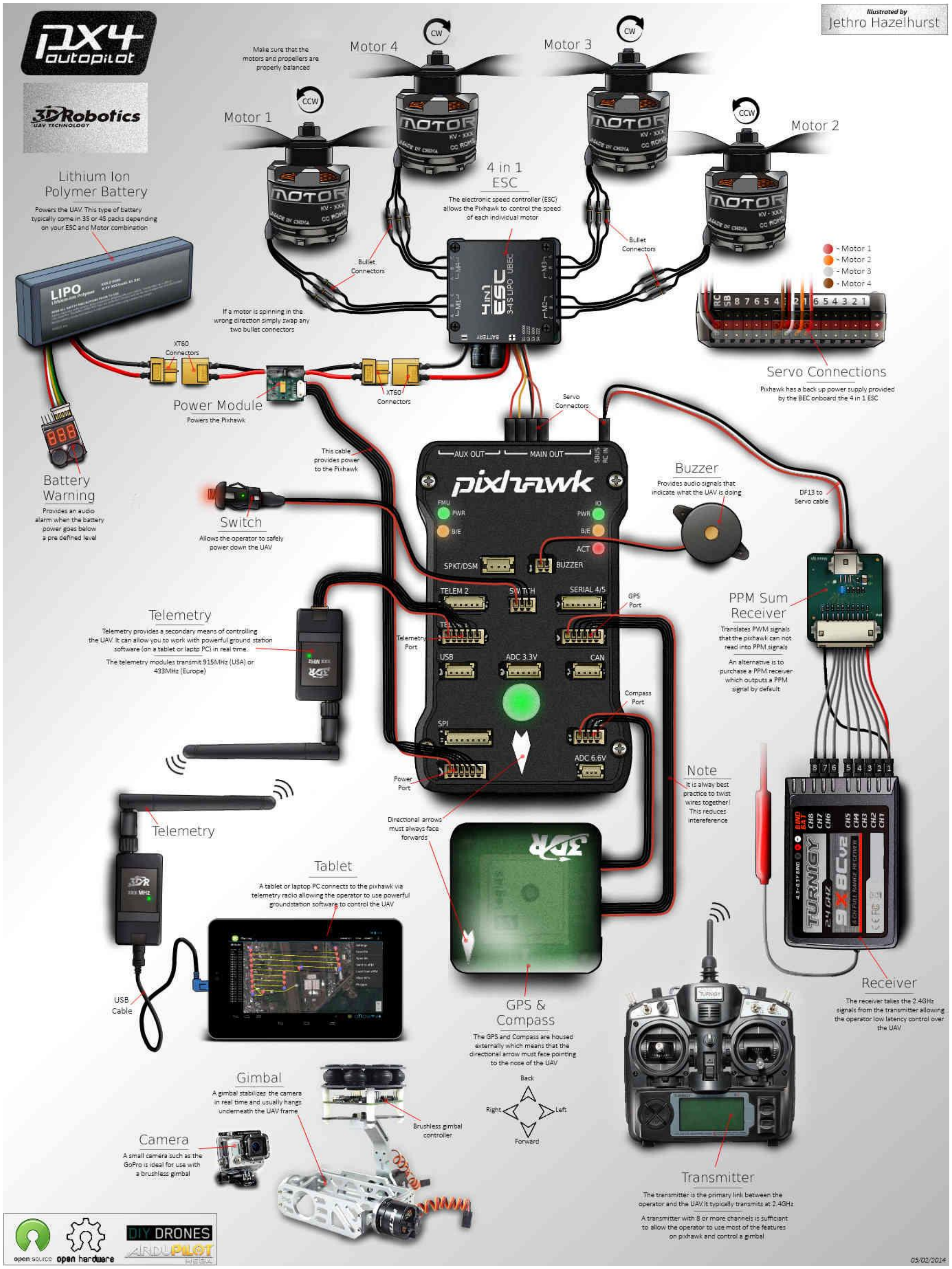
Connect remote control inputs:

NAVIO2 supports PPM and SBUS signals as an RC input. To connect receivers that do not support PPM output you can use PPM encoder. PPM receiver is powered by NAVIO2 and does not require power on the servo rail.

GNSS Antenna:

GNSS antenna is plugged into the MCX port on top of NAVIO2.





px4 autopilot
3DRobotics
Illustrated by Jethro Hazelhurst
Motor 1
Motor 2
Motor 3
Motor 4
CW
CCW
Make sure that the motors and propellers are properly balanced
4 in 1 ESC
The electronic speed controller (ESC) allows the Pixhawk to control the speed of each individual motor
Bullet Connectors
If a motor is spinning in the wrong direction simply swap any two bullet connectors
Lithium Ion Polymer Battery
Powers the UAV. This type of battery typically come in 3S or 4S packs depending on your ESC and Motor combination
XT60 Connectors
Power Module
Powers the Pixhawk
Battery Warning
Provides an audio alarm when the battery power goes below a pre defined level
- Motor 1
- Motor 2
- Motor 3
- Motor 4
Servo Connections
Pixhawk has a back up power supply provided by the BEC onboard the 4 in 1 ESC
Servo Connectors
This cable provides power to the Pixhawk
Switch
Allows the operator to safely power down the UAV
Buzzer
Provides audio signals that indicate what the UAV is doing
DF13 to Servo cable
AUX OUT
MAIN OUT
pixhawk
SPKT/DSM
BUZZER
TELEM 2
SWITCH
SERIAL 4/5
GPS Port
TELEM 1
Telemetry Port
USB
ADC 3.3V
CAN
SPI
I2C
Compass Port
ADC 6.6V
Power Port
PPM Sum Receiver
Translates PWM signals that the pixhawk can not read into PPM signals
An alternative is to purchase a PPM receiver which outputs a PPM signal by default
Telemetry
Telemetry provides a secondary means of controlling the UAV. It can allow you to work with powerful ground station software (on a tablet or lapto PC) in real time.
The telemetry modules transmit 915MHz (USA) or 433MHz (Europe)
Note
It is alway best practice to twist wires together! This reduces intereference
Directional arrows must always face forwards
Tablet
A tablet or laptop PC connects to the pixhawk via telemetry radio allowing the operator to use powerful groundstation software to control the UAV
USB Cable
GPS & Compass
The GPS and Compass are housed externally which means that the directional arrow must face pointing to the nose of the UAV
Back
Right
Left
Forward
Receiver
The receiver takes the 2.4GHz signals from the transmitter allowing the operator low latency control over the UAV
Gimbal
A gimbal stabilizes the camera in real time and usually hangs underneath the UAV frame
Brushless gimbal controller
Camera
A small camera such as the GoPro is ideal for use with a brushless gimbal
Transmitter
The transmitter is the primary link between the operator and the UAV. It typically transmits at 2.4GHz
A transmitter with 8 or more channels is sufficiant to allow the operator to use most of the features on pixhawk and control a gimbal
open source
open hardware
DIY DRONES
ARDUPILOT
05/02/2014

Fig 5.3

KK2 Flight controller

This makes it possible to tune the parameters without having to hook it up to a computer. You can do the tuning right on the controller, tuning the parameters using the menu you can see in the display screen.This controller board does not have any of the advanced features like GPS, altitude hold, go home or autonomous navigation using waypoints.

Advantages:

- The onboard display allows you to define flight parameters

  without needing to use a laptop.

- Can support from 4 to 8 rotors

- Very inexpensive

Fig 5.4

Disadvantages:

- Does not have advanced features like GPS, Go Home and

  autonomous navigation

- self-leveling is slow and not very impressive

Apm Flight controller

This is another feature rich flight controller - and the onboard GPS is excellent. This controller board is also used in rc airplanes and rc cars - mainly because the GPS is so very good.It is also easy to tune and setup. It has a 3 axis gyroscope, a 5 Hz GPS, a 3 axis accelerometer, a barometer and a magnetometer. All of this makes this a great choice for professional videography and telemetry.

Advantages:

- Very feature rich
- Sensors are very accurate

Fig 5.4

- Ideal for professional videographers or those interested in telemetry and autonomous flight

Disadvantages:

- Expensive

Naze32 Flight controller

This is a 32 bit flight controller and has a host of features. It comes in two versions - Acro Naze 32 and Full Naze 32. As the name suggests, the Acro Naze 32 is more suited for users who are interested in sports flying and racing. Full Naze 32 has two additional sensors - a compass and a barometer.

Its main advantage is flight stability. In regard to this particular flight characteristic, its performance exceeds that of many other boards like KK2 controller. And the control response is very good.

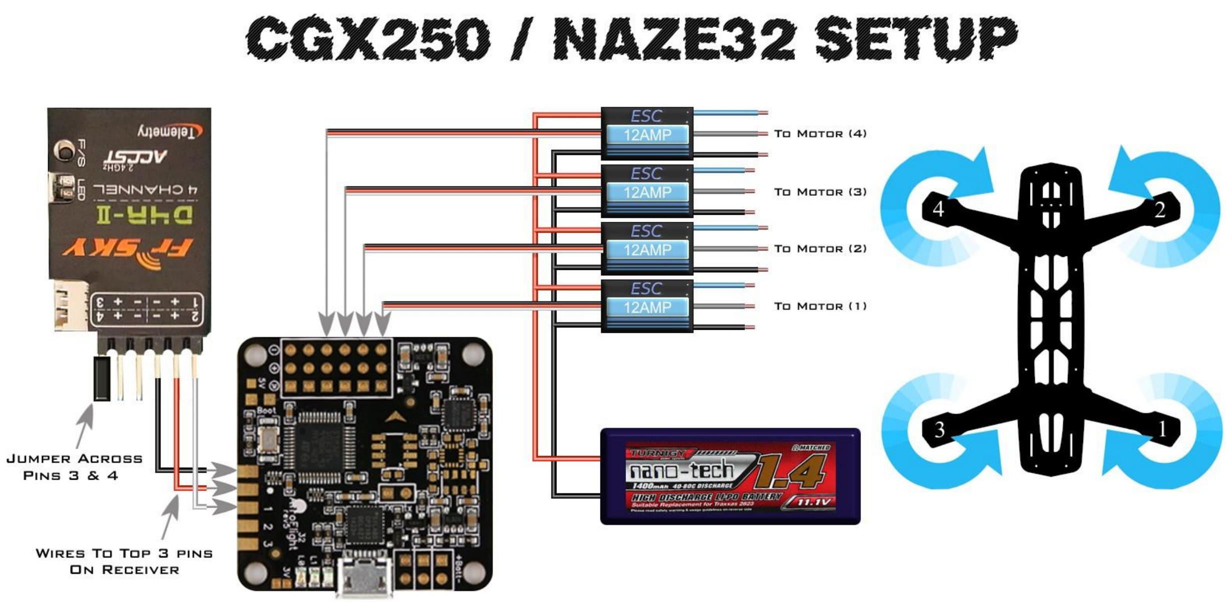
Advantages:

- Very well designed and built.
- Very stable flight and very fast control response.
- Relatively inexpensive considering the features
- Very small and very light weight - ideal for sports quads

Disadvantages:

- The GPS is not very reliable

- Does not have an onboard display - so you have to hook it up to a computer for tuning.



DJI NAZA

It is extremely stable, it can Go Home, hold altitude, fly autonomously and more - and all of these are very accurate and reliable. DJI's strength is their user friendliness - their stuff is plug and play and works well, without you having to do a lot of the fine tuning. That's one reason why they are the market leaders when it comes to high end quadcopters - they build quads that are truly plug and play. It can help you capture the smoothest and highest quality video - with minimal input on your part.

Advantages:

- It has just about every feature a professional photographer would want
- Very stable
- Easy to setup and tune

Disadvantage:

- Expensive

Fig 5.6

Programming languages

- **C LANGUAGE**
- C is a general-purpose, imperative programming language known for its efficiency, flexibility, and low-level
- capabilities. It has been widely used since its creation in the 1970s and remains relevant today in various
- domains. Some common uses of C include:
- **1.System Programming**: C is often used for writing system software such as operating systems (e.g.,
- Linux kernel), device drivers, firmware, and utilities.
- **2.Embedded Systems**: C is popular for programming embedded systems and microcontrollers due to its
- low-level features, close-to-hardware capabilities, and efficiency in resource-constrained environments.
- **3.Application Software**: Despite the rise of higher-level languages, C is still used for developing
- application software, particularly when performance and efficiency are critical, such as in game
- development, audio/video processing, and graphics programming.
- **4.Compilers and Interpreters**: Many compilers and interpreters for other programming languages are
- implemented in C or have significant components written in C due to its efficiency and portability.
- 5.**Database Systems**: Some database management systems (DBMS) and database engines are written
- in C or have critical components implemented in C for performance reasons.
- **6.Networking**: C is used for network programming, including building network servers, clients, and
- protocols. It's commonly used in developing networking applications such as web servers, email servers.

- **PYTHON LANGUAGE**
- Python is a versatile programming language used for a wide range of applications, including web
- development, data analysis, artificial intelligence, scientific computing, automation, and more. Some
- common uses of Python include:
- **1.Web Development**: Python is used to create web applications using frameworks like Django, Flask,
- and Pyramid.
- 2.**Data Science and Machine Learning**: Python's libraries such as NumPy, Pandas, Matplotlib, and
- scikit-learn are extensively used for data analysis, visualization, and machine learning tasks.
- **3.Scripting and Automation**: Python's simplicity and readability make it a popular choice for writing
- scripts to automate repetitive tasks, system administration, and various workflows.
- **4.Scientific Computing**: Python is widely used in scientific computing and research, thanks to libraries
- like SciPy and specialized tools such as Jupyter Notebooks.
- **5.Game Development**: Python is utilized in game development, either for building entire games or for
- scripting within game engines like Unity or Godot.
- 6.**Desktop GUI Applications**: Python's Tkinter library is commonly used for building desktop Graphical
- User Interface (GUI) applications.
- **7.Backend Development**: Python is employed in developing server-side logic for web applications,
- utilizing frameworks like Django, Flask, and FastAPI.
- **8.DevOps and System Administration**: Python is used for automation, configuration management, and
- monitoring tasks in DevOps workflows.
- **9.Education**: Python's simplicity and readability make it an excellent choice for teaching programming
- concepts, and it's widely used in educational settings.
- **10.Networking**: Python is used for network programming, such as building network applications, scripting

- **JAVA PROGRAMMING LANGUAGE**
- Java is a widely-used programming language known for its portability, scalability, and versatility. It's used
- in a variety of domains and applications. Here are some common uses of Java
- **1.Web Development**: Java is used for building dynamic and interactive web applications. Java-based
- web frameworks like Spring, Hibernate, and Struts are widely used for developing enterprise-level web
- applications.
- **2.Mobile App Development**: Java is used for developing Android applications. Android Studio, the
- official IDE for Android development, uses Java as the primary programming language for creating mobile
- apps.
- **3.Enterprise Software Development**: Java is extensively used in enterprise environments for developing
- large-scale, mission-critical applications. It's well-suited for building robust and scalable systems due to its
- strong support for concurrency, multi-threading, and networking.
- **4.Desktop Application Development**: Java provides tools and libraries for building cross-platform
- desktop applications. JavaFX and Swing are popular frameworks for creating graphical user interfaces
- (GUIs) for desktop applications.
- **5.Big Data Technologies**: Java is used in various big data technologies and frameworks such as Apache
- Hadoop, Apache Spark, and Apache Flink. These frameworks leverage Java's scalability and
- performance for processing and analyzing large volumes of data.
- **6.Web Services**: Java is widely used for developing web services, including SOAP and RESTful APIs.
- Java EE (Enterprise Edition) provides a comprehensive platform for building and deploying web services
- **7.Cloud Computing**: Java is used for developing cloud-based applications and services. Platforms like

.

- **C++ LANGUAGE**
- C++ is a powerful programming language with a wide range of applications across various domains. Here
- are some common uses of C++:
- **1.System Software Development**: C++ is often used for developing system software such as operating
- systems, device drivers, compilers, interpreters, and firmware. Its low-level features and direct access to
- hardware make it suitable for writing system-level code.
- **2.Game Development**: C++ is extensively used in the gaming industry for developing high-performance
- games and game engines. Popular game engines like Unreal Engine and Unity are written in C++.
- 3.**Application Software Development**: C++ is used for building desktop applications, including graphical
- user interfaces (GUIs), productivity tools, multimedia software, and office suites. Applications like Adobe
- Photoshop and Microsoft Office have components written in C++.
- **4.Embedded Systems Development**: C++ is commonly used for programming embedded systems, IoT
- devices, and real-time systems. Its ability to produce efficient and compact code makes it suitable for
- resource-constrained environments.
- **5.Graphics Programming**: C++ is used for graphics programming and computer graphics applications,
- including 2D and 3D graphics rendering, animation, and visual effects. Libraries like OpenGL and DirectX
- are often used for graphics programming in C++.
- **6.High-Performance Computing (HPC)**: C++ is used in scientific computing, simulations, and numerical
- analysis where performance is crucial. It allows for fine-grained control over memory management and
- optimization techniques, making it suitable for HPC applications.
- **7.Financial and Trading Systems**: C++ is widely used in the finance industry for building high-frequency
- trading systems, algorithmic trading platforms, risk management software, and financial modeling tools.
- Its performance and low-latency capabilities are essential for handling large volumes of financial data.

Programs to run BLD motors:

Single motor functioning program:

```
// these constants won't change. They're used to give names to the pins used:
int led0=11; // pwm output pin that the led is attached to
int P=A0;// Analog input pin that the potentiometer is attached to
void setup() {
//initializing pinmodes to the output pin an input pin
  pinMode(led0,OUTPUT);
  pinMode(P,INPUT);
  Serial.begin(9600);
//initialize serial communications at 9600bps to print on the PC monitor:
  Serial.println("ADC : VALUE :MAP");//read the analog in value:
  Serial.println("--------------------");//print new line
}

void loop() {
   int R=analogRead(P);//value  read from the pot
  int Q=map(R,0,1023,0,255);//map it to the range of the analog out:
  int S=map(Q,0,255,120,165);//map the range of led value:
   Serial.print(R);//print the result of pot
   Serial.print("---");
   Serial.print(Q);//print the result of range of analog out
  Serial.print("---");
  Serial.print(S);//print the changed analog out range
  Serial.println();
 analogWrite(led0,Q);//write analog value in led1:
 delay(20); //
}
```

Two motors functioning program:

```
// These constants won't change. They're used to give names to the pins used:
 int led1=11; / /pwm output pin that the led is attached to
int led2=10; // pwm output pin that the led is attached to
int PWM1=A0;// Analog input pin that the potentiometer is attached to
int PWM2=A1;// Analog input pin that the potentiometer is attached to
void setup(){
//pinmodes for output and input pins
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(PWM1,INPUT);
  pinMode(PWM2,INPUT);
  Serial.begin(9600);
//initialize serial communications at 9600bps to print on the PC monitor:
 Serial.println("ADC1 : VALUE1   ADC2 : VALUE2");//read the analog in value:
Serial.println("------------------");//print new line
}
void loop() {
   int R1=analogRead(PWM1); //value  read from the pot1
 int Q1=map(R1,0,1023,150,170); //map it to the range of the analog1 out:
 Serial.print(R1);// print the result of pot1
 Serial.print("---");//print the new line
   Serial.print(Q1);// print the result of range of analog1 out
   Serial.print("       ");//print the new line
 analogWrite(led1,Q1);// write the analog value in led1:
 int R2=analogRead(PWM2); //value  read from the pot2
int Q2=map(R2,0,1023,150,170);  //map it to the range of the analog2 out:
Serial.print(R2); // print the result of pot2
 Serial.print("---");//print the new line
 Serial.print(Q2); // print the result of range of analog2 out
Serial.println();//print the new line
 analogWrite(led1,Q2); // write the analog value in led1:
 delay(20);//delay to print result
}
```

POTENTIOMETER:

Fig 7.0

To control the drone through the Arduino, the first thing you need to figure out is what kind of signal the physical controller is generating.

For most of the drone controller, there are two joysticks on it which control the horizontal and vertical directions separately. The signal generated by the joysticks are both analog. You could use a multimeter to figure that out.

Typically, for each joystick, there are two potentiometers on it. By pulling the joystick, the resistance values are changing thus different analog signals are generated. There are 3 wires for each potentiometer, which are GND, VCC and Control Signal (the voltage of the resistor). Usually the GND is Black, the VCC is Red and the Control Signal is other color (say, Yellow, White and

Sometimes Brown). Use a multimeter to identify those wires and the voltages between Control Signal and GND.

Program for potentiometer:

```
// These constants won't change. They're used to give names to the pins used:
int led1=11; / /pwm output pin that the led is attached to
int led2=10; / /pwm output pin that the led is attached to
int led3=13; / /pwm output pin that the led is attached to
int led4=9; / /pwm output pin that the led is attached to
int PWM1=A0; // Analog input pin that the potentiometer is attached to
int PWM2=A1; // Analog input pin that the potentiometer is attached to
int PWM3=A2; // Analog input pin that the potentiometer is attached to
int PWM4=A3; // Analog input pin that the potentiometer is attached to

void setup() {
//pinmodes for output and input pins
 pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(led3,OUTPUT);
  pinMode(led4,OUTPUT);
  pinMode(PWM1,INPUT);
  pinMode(PWM2,INPUT);
  pinMode(PWM3,INPUT);
  pinMode(PWM4,INPUT);
  Serial.begin(9600);
//initialize serial communications at 9600bps to print on the PC monitor:
Serial.println("ADC1 : VALUE1   ADC2 : VALUE2  ADC3  :VALUE3   ADC4
: VALUE4");
//read the analog in value:

 Serial.println("-----------------------------------------------------------");
//print the new line
}

void loop() {
 int R1=analogRead(PWM1); //read the value from pot1:
 int Q1=map(R1,0,1023,150,170); //map it to the range of the analog1 out:
Serial.print(R1); // print the result of pot1
  Serial.print("---");//print the new line
Serial.print(Q1); // print the result of range of analog1 out:
 Serial.println();//print the new line
```

```
analogWrite(led1,Q1); // write analog value in led1:
 int R2=analogRead(PWM2); //read the value from pot2:
 int Q2=map(R2,0,1023,150,170); //map it to the range of the analog2 out:
 Serial.print(R2); // print the result of pot2
Serial.print("---");//print the new line
Serial.print(Q2); // print the result of range of analog2 out:
 Serial.println();//print the new line
analogWrite(led1,Q2); ); // write analog value in led1:
 int R3=analogRead(PWM3); //read the value from pot3:
  int Q3=map(R3,0,1023,150,170); //map it to the range of the analog3 out:
 Serial.print(R3); // print the result of pot3
 Serial.print("---");//print the new line
Serial.print(Q3); // print the result of range of analog3 out:
 Serial.println();//print the new line
 analogWrite(led1,Q3); ); // write the analog value in led1:
int R4=analogRead(PWM4); //read the value from pot4:
 int Q4=map(R4,0,1023,150,170); //map it to the range of the analog4 out:
 Serial.print(R4); // print the result of pot4
Serial.print("---");//print the new line
Serial.print(Q4); // print the result of range of analog4 out:
 Serial.println();//print the new line
------------------------+----------------- analogWrite(led1,Q4); ); // write analog
value in led1:
 delay(20); //delay to print result
}
```

PWM Signal Specification:

Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off. The duration of "on time" is called the pulse width. To get varying analog values, you change, or modulate, that pulse width. If you repeat this on-off pattern fast enough with an LED for example, the result is as if the signal is a steady voltage between 0 and 5v controlling the brightness of the LED.

Program using PWM technique:

```
int led0=11; // pwm output pin that the led is attached to
void setup() {
//pinmodes for output and input pins
 pinMode(led0,OUTPUT);
}
void loop() {
  for(int val=0;val<=255;val++)
//using condition writing the range led glow
  {
  analogWrite(led0,val);//writing  value to led0
  delay(5);//delay for  led glow
   }
  for(int val=255;val>=0;val--)
//using condition writing the range led glow
{
 analogWrite(led0,val); //writing  value to led0
 delay(5); //delay for  led glow
}
```

Radio Transmitter and Receiver:

1.Turnigy 10x2.4GHz 10 Channel Radio

The radio transmitter and receiver allow you to control the quad copter. There are many suitable models available, but you will need at least four channels for a basic quad copter with the KK2.0 control board. We recommend using a radio with 8 channels, so there is more flexibility for later projects that may require more channels. The Turnigy 10x is a great choice for a first radio. It's inexpensive yet still has some advanced functionality.

One transmitter to rule them all
Number of channels:

You also want to consider the number of channels a transmitter has before you buy it. A channel is one particular control. So a 4-channel radio, for example, will be able to control 4 different things, not more. A 6-channel radio will be able to control 6 things, and so on.

To fly a hobby-grade quad copter (or multicopter), you need a minimum of 4 channels. Not because there are 4 motors, but because there are 4 controls you must use to keep the quad copter flying:

- **Throttle**
- **Pitch**
- **Yaw**
- **Roll**

Throttle:

Throttle is used control the vertical up and down motion of the drone.

+throttle makes the drone fly higher.

-throttle makes the drone fly lower.

Pitch:

Rotation around **side-to-side** axis is called **pitch .**

Or

It refers to the rotation of an aircraft around its **lateral axis** .

Simply we can say that **head up or tail up** .

Pitch is the moment of the drone either in forward and backward. Forward pitch is achieved by pushing the electron stick forward which makes the drone tilt and to move forward ,vice versa to backward pitch.

Yaw:

Yaw is the rotation of an aircraft around its **vertical axis**.

Yaw is used for rotating the head of drone either to right or left controlled by rudder.

Roll:

Rotation around **Front-to-back** axis is called **roll.**

Or

Roll refers to the rotation of an aircraft around **longitudal axis.**

Roll is making the drone fly either to left or right.

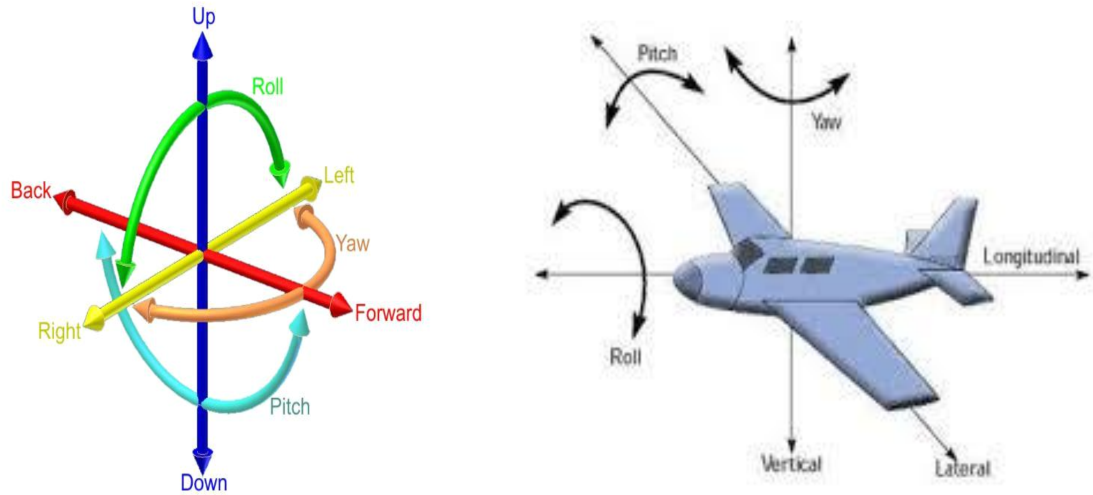
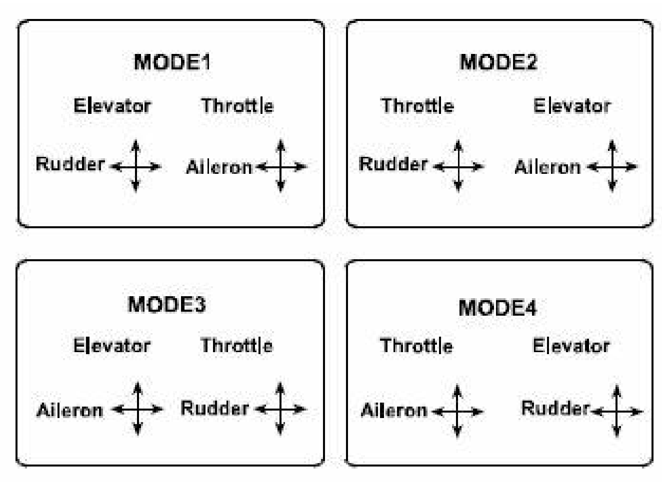


Fig 6.0

DIRECTIONS OF ROLL,PITCH,YAW,THROTTLE

These four channels are assigned to the four axes on your transmitter each of the two sticks has two axes up/down and left/right.The remaining channels would be assigned to the dials and switches on your transmitter.Additionally, you may want to have an extra channel assigned to a switch on your transmitter that will change the flight mode from say GPS mode to manual mode, and you may even want another switch for a failsafe or return-to-home.

Modes of Transmitter:



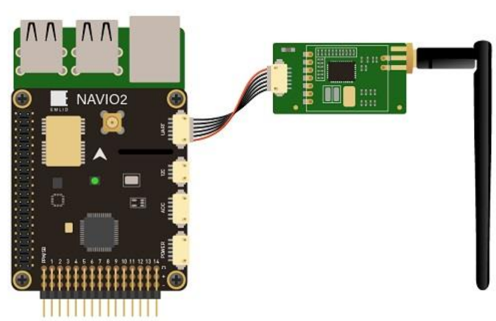
Telemetry:

Radio modems can be connected either over UART or over USB.

- UART radio

For UART port

.



- USB radio

Telemetry transmitter and receiver

Telemetry receivers and **transmitters** are data acquisition from the multicopter flight controller.we get data copter position and data copter temperature**.**
Telemetry receivers and **transmitters** can be produced as separate **receiver** and **transmitter** units ,or combined into a device known as a transceiver.

GPS:

The global positioning system is a satellite navigation system that uses a radio receiver to collect signals from orbiting satellites to determine position, speed, and time. This navigation system is more accurate than over forms of navigation, and provides position knowledge to within a few meters. Advanced GPS systems can provide even better accuracies to within a few centimeters. The miniaturization of integrated circuits has allows GPS receivers to be highly economical, and available to everyone. GPS is a broadcast radio system that reaches almost all areas of the planet, so it is highly accessible.

GPS

Propellers:

Fig 14.0

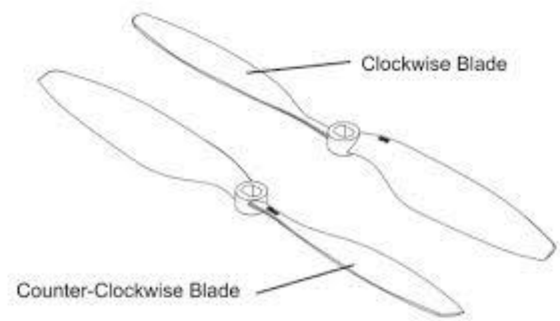


Fig1 4.1

Two wing Propeller

Directions of propellers

Fig1 4.2

Fig 14.3

Three Wing Propeller

Directions of propellers

A quad copter has four propellers, two “normal” propellers that spin counter-clockwise, and two “pusher” propellers that spin clockwise. The pusher propellers will usually be labeled with an R after the size. For the quad copter configuration in this post, were using 9×4.7 props. This is a good size for the motors and ESCs were using.

Battery:

Fig 15.0

Fig 15.1

NanoTech 3000Mah LiPo

Quad copters typically use ()LiPo batteries which come in a variety of sizes and configurations. We typically use a 3S1P battery, which indicates 3 cells in parallel. Each cell is 3.7 volts, so this battery is rated at 11.1 volts. LiPo batteries also have a C rating and a power rating in mAh (which stands for milliamps per hour). The C rating describes the rate at which power can be drawn from the battery, and the power rating describes how much power the battery can supply. Larger batteries weigh more so there is always a tradeoff between flight duration and total weight. A general rule of thumb is that doubling the battery power will get you 50% more flight time, assuming your quad copter can lift the additional weight.

Fig 15.2

Fig 15.3

Nano- Tech 4000mAh LiPo

- It depends on the particular setup. The only way to know for sure is test out both. 4s is more efficient than 3s but on the flip side you're carrying an extra battery cell (extra dead weight). The best setup is a balance of motors, props, copter weight and battery capacity/voltage. There's no easy answer unfortunately. If you are still fairly new I would suggest sticking with 3s. Less weight, les complication, cheaper battery cost all weigh in their favor.

- 4S batteries will result in a fast and really agile multirotor - too agile in fact for a beginner. Start slow and learn how to fly before trying to do more crazy stuff."

Gimbal:

Fig 12.0

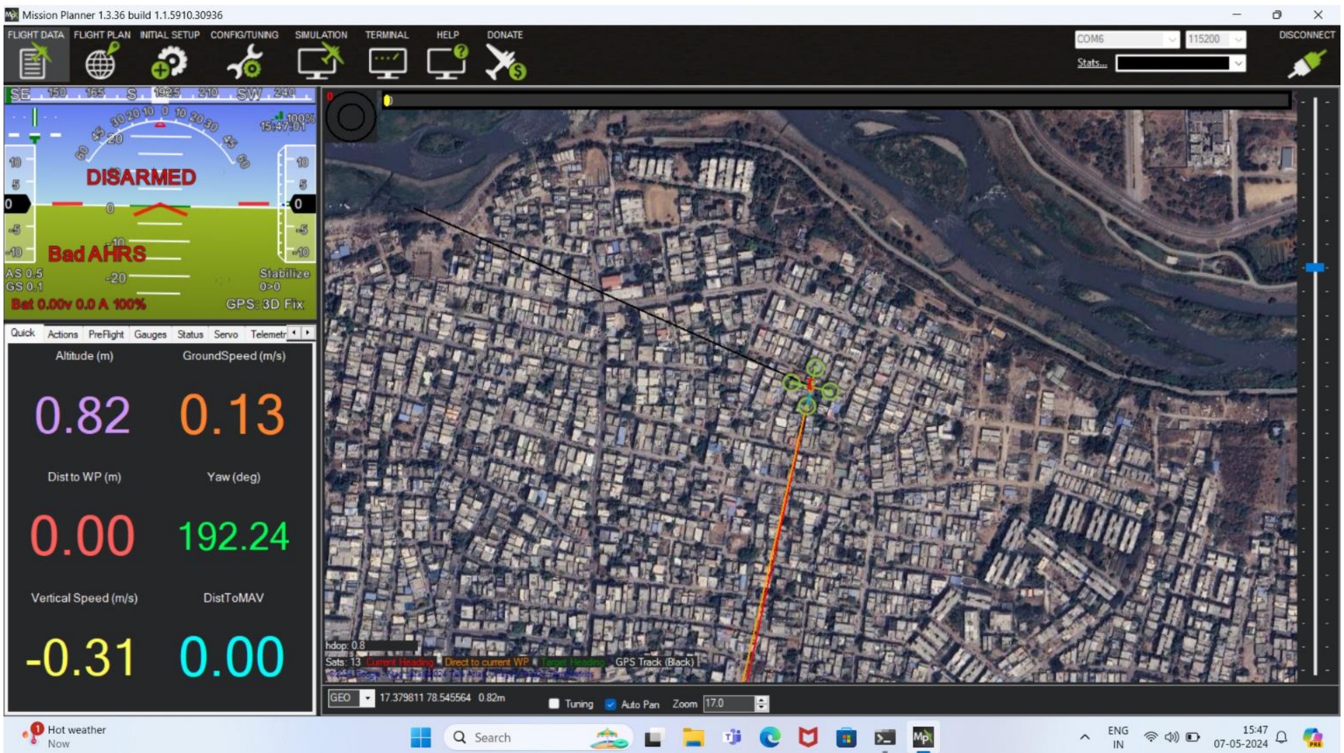
The Drone gimbal is the pivoting mount that rotates about the x, y and z axes to provide stabilization and pointing of cameras or other sensors.

Software we used for drone:

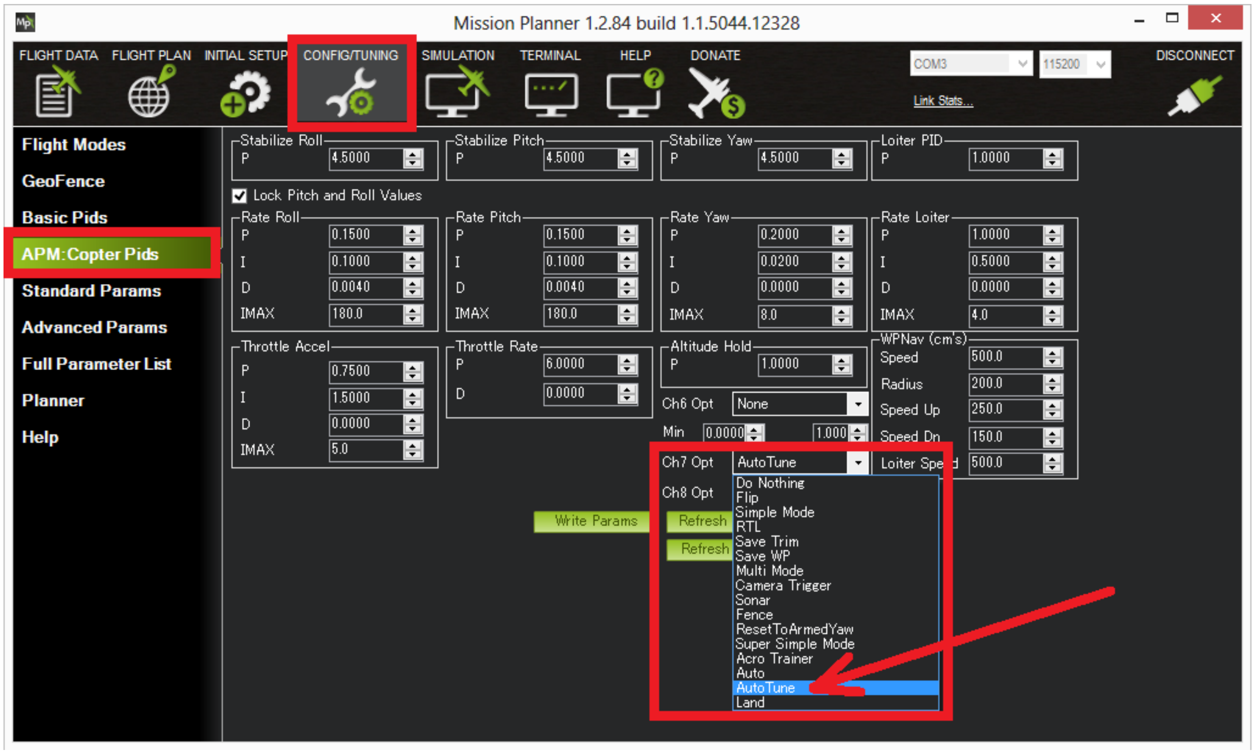
Mission Planner is a full-featured ground station application for the ArduPilot open source autopilot project. This page contains information on the background of Mission Planner and the organization of this site.Mission Planner is a ground control station for Plane, Copter and Rover. It is compatible with Windows only. Mission Planner can be used as a configuration utility or as a dynamic control supplement for your autonomous vehicle. Here are just a few things you can do with Mission Planner:

- Load the firmware (the software) into the autopilot (APM, PX4...) that controls your vehicle.
- Setup, configure, and tune your vehicle for optimum performance.
- Plan, save and load autonomous missions into you autopilot with simple point-and-click way-point entry on Google or other maps.
- Download and analyze mission logs created by your autopilot.
- Interface with a PC flight simulator to create a full hardware-in-the-loop UAV simulator.
- With appropriate telemetry hardware you can:
    - Monitor your vehicle’s status while in operation.
    - Record telemetry logs which contain much more information the on-board autopilot logs.
    - View and analyze the telemetry logs.

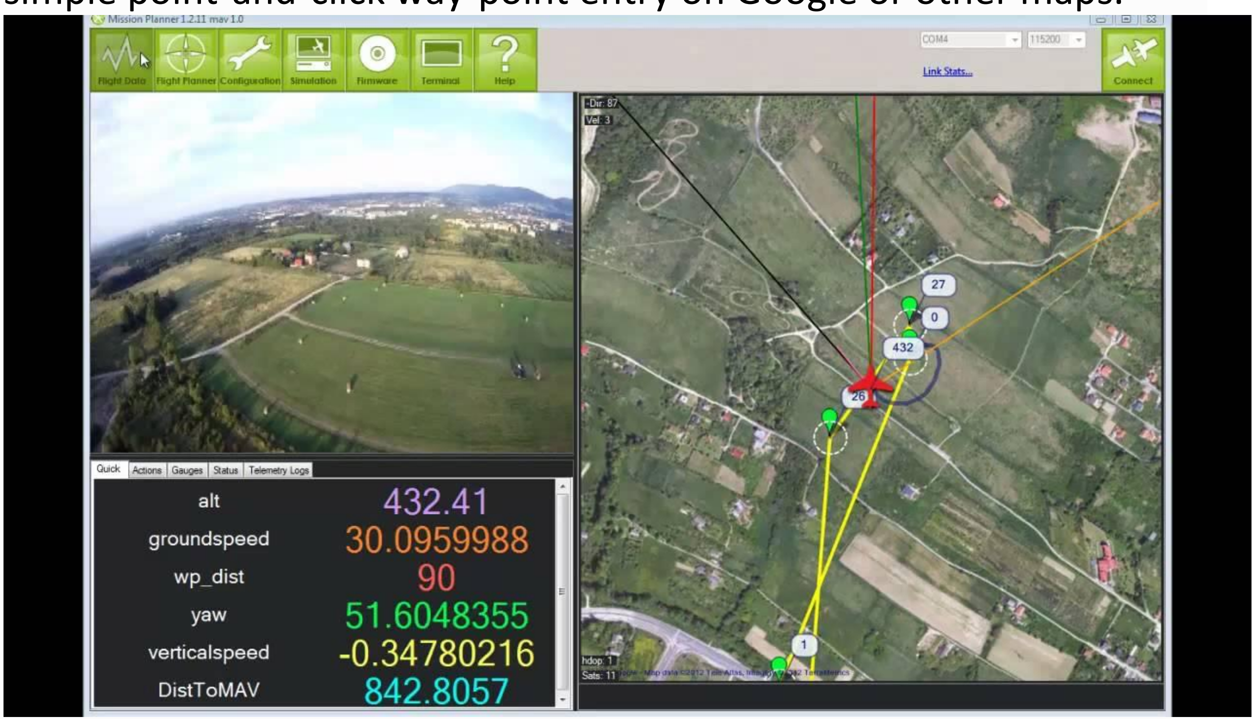
- Load the firmware (the software) into the autopilot (APM, PX4...) that controls your vehicle.



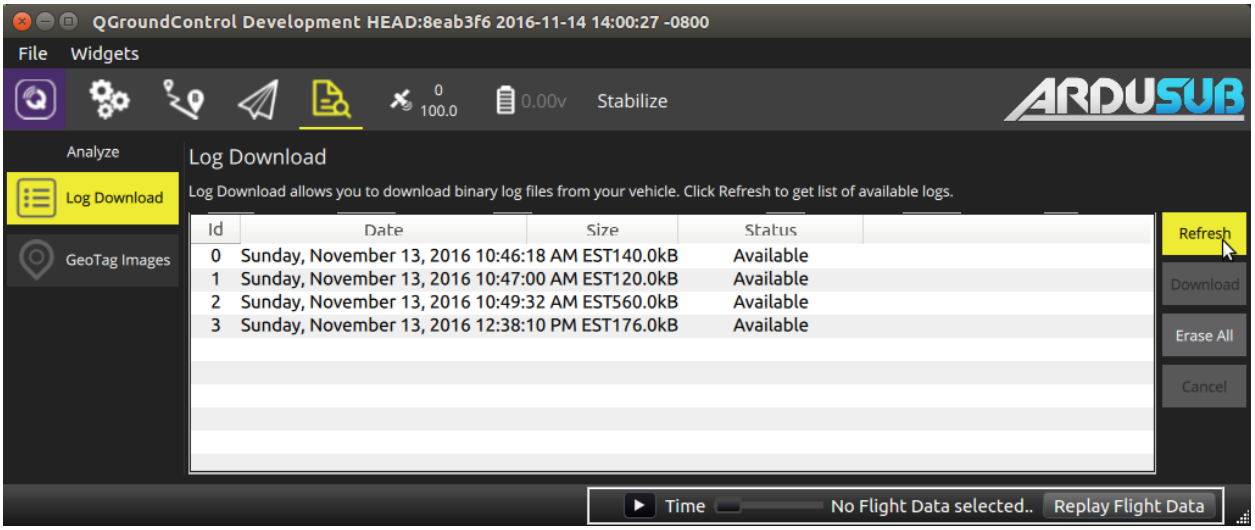
- setup, configure, and tune your vehicle for optimum performance.



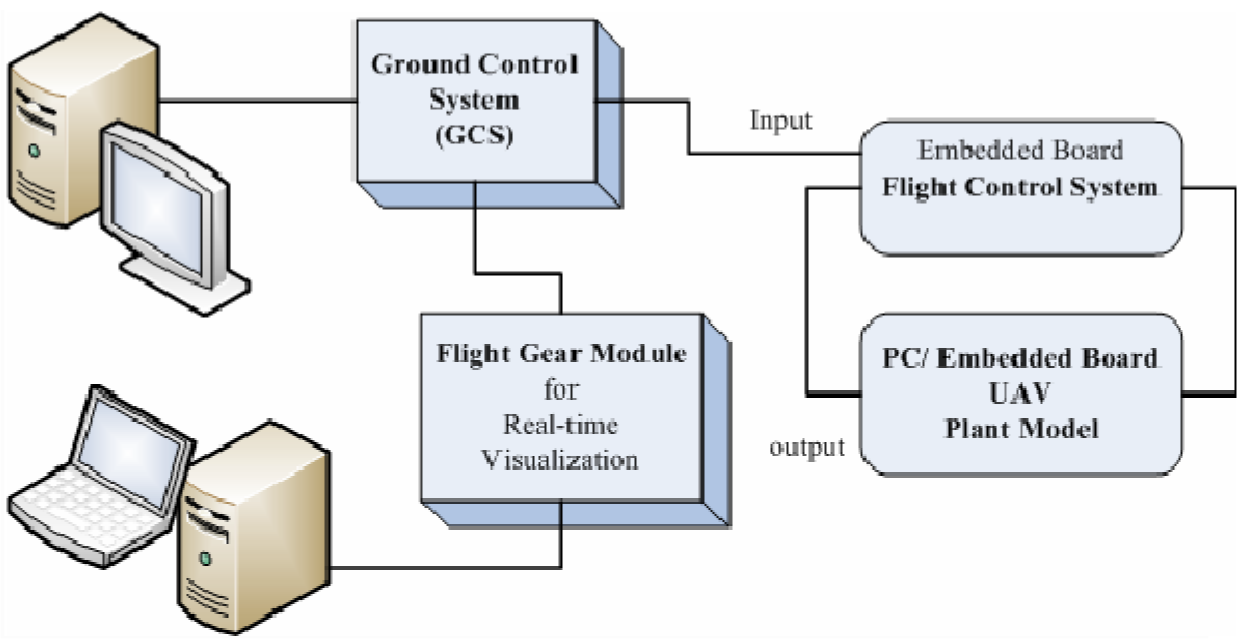
- plan, save and load autonomous missions into you autopilot with simple point-and-click way-point entry on Google or other maps.



Download and analyze mission logs created by your autopilot.

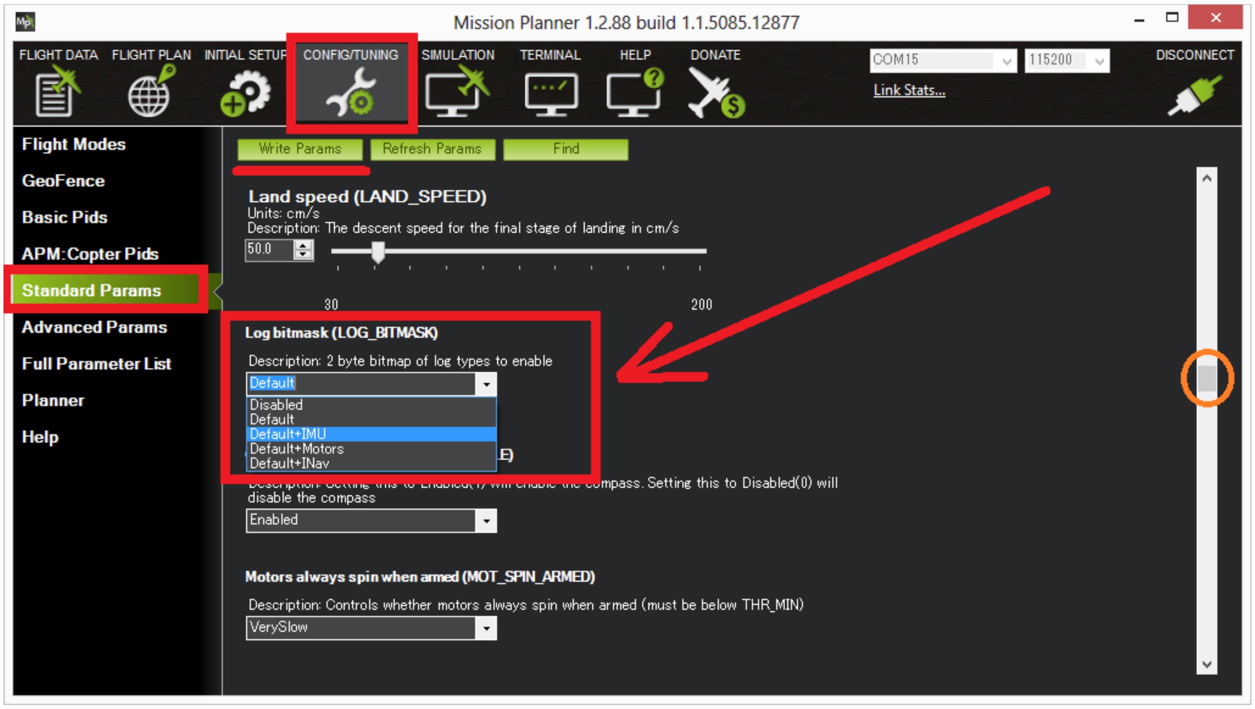


- Interface with a PC flight simulator to create a full hardware-in-the-loop UAV simulator.



- Monitor your vehicle's status while in operation.

Record telemetry logs which contain much more information the the on-board autopilot logs.



○

View and analyze the telemetry logs.

Operate your vehicle in FPV (first person view).

Example program to create menu and selection:

```
int led1=3;
int led2=5;
int selectbutton=A0;
int upbutton=A1;
int downbutton=A2;
int sel=0;
int pin;
int value1=0,value2=0;
int up1=0,up2=0;
//int down1=0,down2=0;
int selectbuttonstate,upbuttonstate,downbuttonstate;

void setup() {
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(selectbutton,INPUT);
  pinMode(upbutton,INPUT);
  pinMode(downbutton,INPUT);
```

```
 digitalWrite(selectbutton,HIGH);
 digitalWrite(upbutton,HIGH);
 digitalWrite(downbutton,HIGH);
 Serial.begin(9600);
 Serial.println("sel : up1 : down1 : up2 : down2");
 Serial.println("----------------------------");
 }
void loop() {
 selectbuttonstate=digitalRead(selectbutton);
if(selectbuttonstate==LOW)
{
sel=sel+1;
 delay(200);

if(sel==3)
{
 sel=0;
}
}
```

```
if(sel==1)
{

upbuttonstate=digitalRead(upbutton);
downbuttonstate=digitalRead(downbutton);

 if(upbuttonstate==LOW)
 {
  up1=up1+1;
  if(up1>=255)
  up1=255;
  value1=up1;
   analogWrite(led1,value1);

 }

 if(downbuttonstate==LOW)
 {
    up1=up1-1;
   if(up1<=0)
```

```
  up1=0;
 {
  value1=up1;
  analogWrite(led1,value1);
 }
  }
 }
 delay(20);

   if(sel==2)
   {
     upbuttonstate=digitalRead(upbutton);
downbuttonstate=digitalRead(downbutton);
    if(upbuttonstate==LOW)
 {
 up2=up2+1;
 if(up2>=255)
 up2=255;
  value2=up2;
  analogWrite(led2,value2);
```

```
 }
  if(downbuttonstate==LOW)


   {
     up2=up2-1;
     if(up2<=0)
     up2=0;
  {
   value2=up2;
   analogWrite(led2,value2);
 }
   }
 }
Serial.print(sel);
Serial.print("---");
Serial.print(up1);
Serial.print("---");
 //Serial.print(down1);
//Serial.print("---");
 Serial.print(up2);
```

```
//Serial.print("---");
 //Serial.print(down2);
Serial.println();
}
```

MISSION PLANNER CONFIGURATION

Loading Firmware

These instructions will show you how to download the latest firmware onto the autopilot using the Mission Planner ground station,which already has ArduPilot firmware installed . See Loading Firmware onto boards without existing ArduPilot firmware .

Connect autopilot to computer

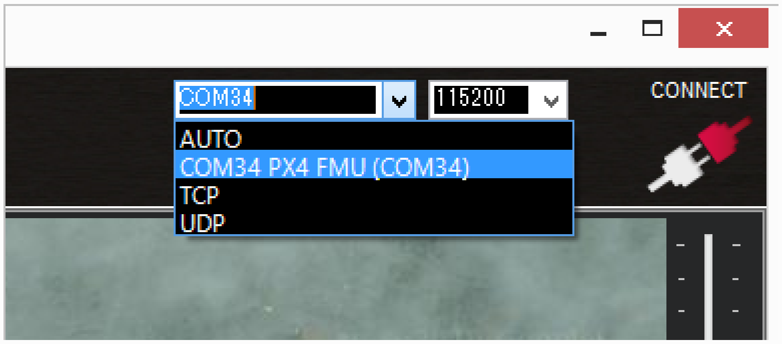
Once you've installed a ground station on your computer, connect the autopilot using the micro USB cable as shown below. Use a direct USB port on your computer (not a USB hub).

Pixhawk USB Connection

Windows should automatically detect and install the correct driver software.

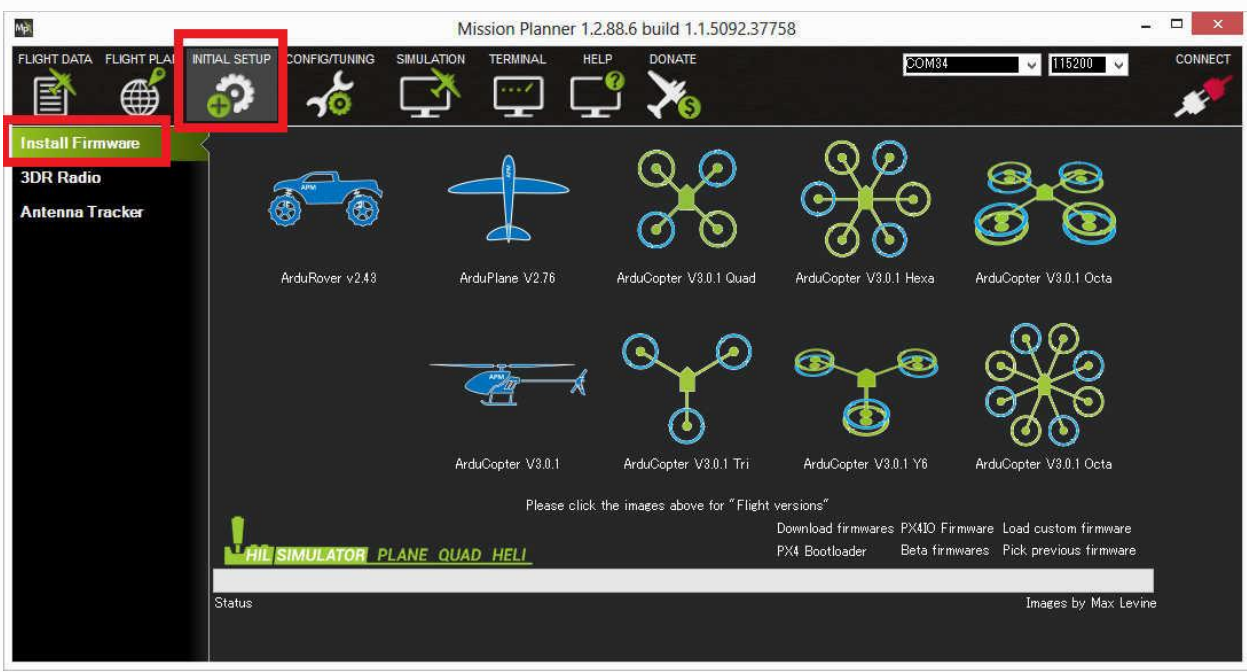
Select the COM port

If using the *Mission Planner* select the COM port drop-down on the upper-right corner of the screen (near the **Connect** button). Select **AUTO** or the specific port for your board. Set the Baud rate to **115200** as shown. Don't hit **Connect** just yet.



Install firmware

On the Mission Planner's **SETUP | Install Firmware** screen select the appropriate icon that matches your frame (i.e. Quad, Hexa). Answer **Yes** when it asks you "Are you sure?".



Mission Planner: Install FirmwareScreen

Next it will try to detect which board you are using and it may ask you to unplug the board, press OK, and plug it back in to detect the board type.

Mission Planner: Install FirmwarePrompt

If all goes well you will see some status appear on the bottom right including the words, "erase...", "program...", "verify.." and "Upload Done". The firmware has been successfully uploaded to the board.

It usually takes a few seconds for the bootloader to exit and enter the main code after programming or a power-up. Wait to press CONNECT until this occurs.

Testing

You can test the firmware is basically working by switching to the *Mission Planner Flight Data* screen and pressing the **Connect** button. The HUD should update as you tilt the board.