ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y SISTEMAS DE TELECOMUNICACIÓN

PROYECTO FIN DE GRADO

TÍTULO: A Do-It-Yourself Unmanned Aerial Vehicle for area monitoring and data collecting

AUTOR: SILI LIU

TITULACIÓN: Grado Ingeniería Electrónica de Comunicaciones

TUTOR: Jesús Rodríguez Molina

DEPARTAMENTO: Ingeniería Telemática y Electrónica

VºBº

Miembros del Tribunal Calificador:

PRESIDENTE: Francisco Jóse Arqués Orobón

TUTOR: Jesús Rodríguez Molina

SECRETARIO: Ana Belén García Hernando

Fecha de lectura:

Calificación:

El Secretario,

TELECOMUNICACIÓN

Campus Sur

POLITÉCNICA

Table of Contents

1. Introduction 7
1.1 Description of IoT 7
What is the IoT? 7
Why we need the IoT? 8
Structure of the Internet of Things 9
Applications and development of the Internet of Things [7] 10
1.2 Drones in the Internet of Things 10
What is a drone? 10
UAVs Classification 11
UAVs Components 12
Applications of drones and their advantages and disadvantages 13
1.3 Contribution from the Project 14
2. Technological framework 14
2.1 Computer or Main-board and Autopilot in the UAVs 14
Arduino platform 15
Raspberry Pi platform 17
BeagleBoard platform 18
Other commonly used platforms 19
Arduino platform vs. Raspberry Pi platform vs. BeagleBoard platform 23
Autopilot extension board (Autopilot Shield) 25
2.2 Open Source Drone Projects 26
Ardupilot/APM 26
Pixhawk/PX4 27
Paparazzi [30] 27
LibrePilot/OpenPilot[39] 28
2.3 Use cases of UAVs in IoT 28
3. Hardware assembly 31
3.1 Powered platform 31
3.2 Preparation for installation 31
3.3 Installation process 35
3.4 Remaining parts and tools 42
4. Software assembly 46
4.1 Introduction of BetaFlight 46
4.2 Getting Started 47
4.3 Configurations 48

5. Testing 65
5.1 Test flight 65
5.2 Testing Mission 67
6. Budget list 75
7. Conclusion 76
8. References and Bibliography 79

Table of Figures

Figure 1: Structure of the Internet of Things[6] 9
Figure 2: Arduino Uno Rec3[22] 16
Figure 3: Raspberry Pi Zero [23] 18
Figure 4: BeagleBone Blue[25] 19
Figure 5: OcPoC™ with Altera Cyclone® mini SoC Flight Controller[27] 20
Figure 6: 3DR Pixhawk Flight Controller [29] 21
Figure 7: Paparazzi Chimera V1.0 autopilot board[30] 21
Figure 8: OpenPilot CopterControl3D/CC3D autopilot board[32] 22
Figure 9: Qualcomm® Snapdragon ™ Flight Kit[33] 23
Figure 10: PXFmini autopilot shield [46] 25
Figure 11: : Navio2 autopilot shield + Raspberry PI 2 [37] 26
Figure 12: MD4-1000`s Test Flight [43] 29
Figure 13: Flyability Elios [45] 30
Figure 14: Frame kit 31
Figure 15: Motors 32
Figure 16: The top surface of ESC 32
Figure 17: : The bottom surface of ESC 33
Figure 18: The top surface of Flight Controller(FC) 33
Figure 19: The bottom surface of Flight Controller(FC) 34
Figure 20: The camera of drone 34
Figure 21: Solder to the bottom of the pads of ESC 35
Figure 22: Welding of 4 motors 35
Figure 23: Solder power section (capacitors and wires) 36
Figure 24: Mounting screws and rubbers 37
Figure 25: Install ESC 37
Figure 26: Mounting motors 38
Figure 27: Installing a second level of rubber 38
Figure 28: Installing FC 39

Figure 29: Connection of ESC and FC 39
Figure 30: Installation of antennas and cameras 40
Figure 31: Installing a third level of rubber and camera parts 40
Figure 32: Mounting nut to secure and connect data cable 41
Figure 33: Mounting propellers 41
Figure 34: Select Receive Protocol [47] 42
Figure 35: Receiver Monitor [52] 42
Figure 36: Signal receiver [53] 43
Figure 37: Remote Controller [53] 43
Figure 38: RC's main screen 44
Figure 39: RC`s Setup Interface 44
Figure 40: RC`s Functions setup Interface 45
Figure 41: RC`s Auxiliary Channels setup interface 45
Figure 42: RC`s Channels display interface 46
Figure 43: BetaFlight main interface 47
Figure 44: BetaFlight setup interface 1 48
Figure 45: BetaFlight setup interface 2 49
Figure 46: BetaFlight setup interface and horizontal calibration 49
Figure 47: BetaFlight setup interface and z-axis calibration 50
Figure 48: BetaFlight Ports interface 50
Figure 49: BetaFlight Configuration interface 51
Figure 50: Mixer Configuration 52
Figure 51: ESC/Motor Configuration 52
Figure 52: System Configuration 53
Figure 53: Receiver channel configuration 54
Figure 54: Others Setting Menu 1 54
Figure 55: Others Setting Menu 2 54
Figure 56: BetaFlight Receiver interface 55
Figure 57: Adjustment Controller Rockers and Switches 56
Figure 58: Receiver value bar is changed 56
Figure 59: Channel Map setting 57
Figure 60: Critical Value Setting 57
Figure 61: BetaFlight Modes interface 57
Figure 62: Modes` value setting range 59
Figure 63: Modes` test interface 1 60
Figure 64: Modes` test interface 2 60

Figure 65: BetaFlight Motores interface 61
Figure 66: Test motor`s operation and direction 61
Figure 67: BLHeli main interface 62
Figure 68: BLHeli connecting interface 62
Figure 69: BLHeli`s motor setting interface 63
Figure 70: BetaFlight OSD interface 64
Figure 71: VTX display screenshots 1 64
Figure 72: VTX display screenshots 2 65
Figure 73: UAV in test flight (Preparing for takeoff.) 66
Figure 74: UAV in test flight (Taking off) 66
Figure 75: UAV in test flight (In the air) 67
Figure 76: Path of flight from Google Maps 68
Figure 77: The drone's camera video signal is at point A 68
Figure 78: The drone's camera video signal is at point B 69
Figure 79: The drone's camera video signal is at point C 69
Figure 80: The drone's camera video signal is at point D 70
Figure 81: Paths of flight from Google Maps 71
Figure 82: The drone's camera video signal is in path A I 71
Figure 83: The drone's camera video signal is in path A II 72
Figure 84: The drone's camera video signal is in path A III 72
Figure 85: The drone's camera video signal is in path A IV 73
Figure 86: The drone's camera video signal is in path B I 73
Figure 87: The drone's camera video signal is in path B II 74
Figure 88: The drone's camera video signal is in path B III 74

Tables

Table 1: Table of abbreviations 5
Table 2: Example for application of the IoT 10
Table 3: Cost statistics table 76

Table of acronyms

Abbreviation	Full name
IoT	Internet of Things
PC	Personal Computer
UAV	Unmanned Aerial Vehicle
PWM	Pulse Width Modulation
PoP	Package on Package
PPM	Pulse-position Modulation
GPIO	General-purpose Input/Output
FPV	First-person view drones
ESC	Electronic Speed Controller
FC	Flight Controller
RC	Remote Controller
VTX	Video Transmission
DIY	Do-It-Yourself

Table 1: Table of acronyms

1. Introduction

1.1 Description of IoT

The term "Internet of Things" has become a buzzword that makes it likely to be heard from colleagues, friends articles or advertisements, but its actual meaning might be blurry or even undefined. The concept of Internet of Things is always in a dynamic and expanding process.

The original concept of the Internet of Things was generally recognized at the MIT Auto-ID Center. Professor Ashton first proposed it in 1999 when he was studying RFID [1]. it overlapped with the idea of sensor networks. *"He was working in supply chain optimization, wanted to attract senior management's attention to a new exciting technology called RFID. Because the internet was the hottest new trend in 1999 and because it somehow made sense, he called his presentation "Internet of Things""* [2]

The report of the same name issued by the International Telecommunication Union (ITU) in 2005 [3] changed the definition and scope of the Internet of Things and expanded its coverage. It no longer referred to the Internet of Things just based on RFID technology. The interconnection of any place and any object, the ubiquitous network and the development vision of ubiquitous computing, in addition to RFID technology, sensor technology, nanotechnology, smart terminals and other technologies became more widely used from that time on.

What is the IoT?

The IoT is an acronym for Internet of Things. As the name implies, IoT consists of two parts, the "Things" and the Internet. It can be assumed that the readers of this document know what the Internet is. We are now in the post-development era of the Internet, as it provides support to widespread apps used at the application layer: every day we follow Twitter, watch the news, watch movies, chat by WhatsApp, shop on-line, etc. We use computer, mobile phones, smart TV, iPads and even Smart Watches and other devices to obtain and transmit information and data. In simple terms, the Internet provides a strong support for the communication between people and equipment or machinery, that is, "human-machine communication with Internet"". Human-machine communication involves people as the main party to transmit different data on the network. Similar to M2M (machine-to-machine communication), but uses internet connection. It's not just limited to the connections between machines, it can also link different devices and systems together. It is when the Internet "belongs to" the "Things", or the "Things" use the Internet as same as we humans, that the Internet of Things becomes an

extension of the Internet, replacing the subject position of "people" with "things", that is, “machine-machine communication with Internet". [4]

But what does "machine-machine communication with Internet " mean? Or what's the difference with human-computer communication? For example, a factory worker may have started a new day's assembly line as usual, but due to a malfunction of a production making machine, while the subsequent machines are still operating, the entire assembly line loses control and the production is delayed today. This is a typical human-machine communication. Once a machine fails, other machines are unaware of it, so human participation in the assembly line is required. But in the Internet of Things era, machines can communicate with each other via the Internet. Then, under the same situation, when a production machine fails, the malfunctioning machine will issue instructions to other machines and inform the monitoring system for maintenance personnel to overhaul.

The terminal for data transmission on the Internet will no longer be just a computer or a mobile phone, but will become any possible things. This "thing" idea includes both animals and plants, and all inanimate objects. It can be speculated that many objects will be equipped with sensors equivalent to the human sensory system, labeling technology with neural transmission systems, and embedded systems with their own brains. For example, the clothes ask the washing machine what water temperature it need, or the backpack reminds human what items are missing.

Why we need the IoT?

The history of communication is as long as human history. In other words, the history of human development is the history of communication technology. From beacon towers, post stations, radio telegraphs, fixed telephones and to mobile phones, communication technologies are present everywhere. The reason of human connection is to transfer information, such as emotional communication, information sharing, and social connection. But why should things be connected? Or why we need the IoT? In order for us humans to survive or live better, we must produce more efficiently, and production cannot be separated from tools. It started with primitive wooden stone tools, then it followed with bronze steel, steam engines, and nowadays we have computers. We use more and more tools, more and more high-tech tools. Communication is also a tool. With the Internet, we can expand human communications. The Internet of Things is such a tool to help connect virtually everything in the world. With IoT human can remotely manage public infrastructure, such as street lights, which can reduce energy consumption, or can design and develop a smart bracelet to monitor the elderly and children, or open an unmanned shipping company to provide ocean logistics services to users anytime, anywhere… There are only more than 7 billion people in the world. But there are more objects, such as street lights and cars, and more animals and plants. In the future, these objects

can communicate with each other through the Internet of Things. The number of connected devices in the Internet of Things will reach several times or more the number of human [5]. Then with the development of the Internet of Things, there will be demand. With demand, there is a market. With the market, there will be "Money".

Structure of the Internet of Things

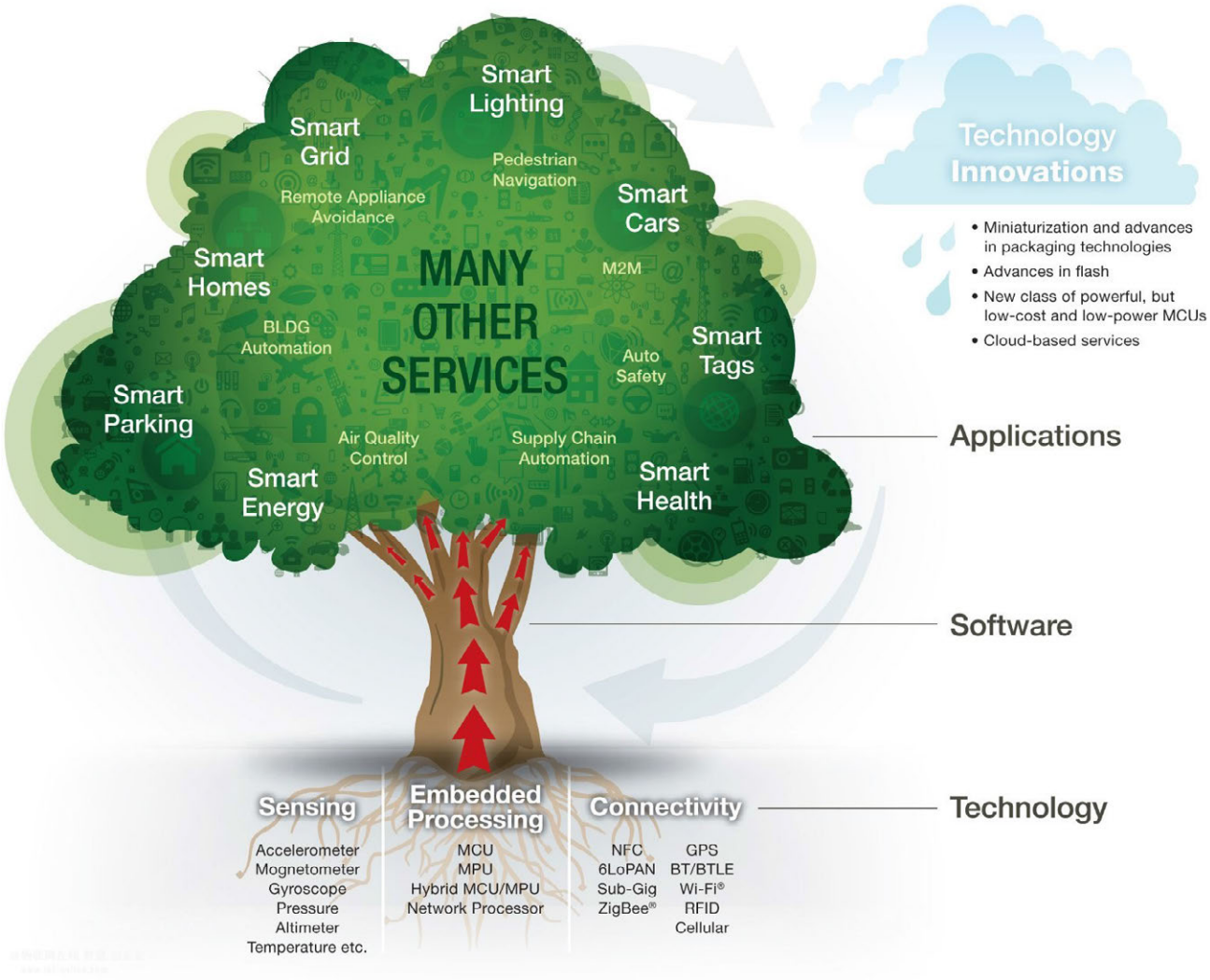

Figure 1: Structure of the Internet of Things[6]

An IoT system can be visually compared to a tree, which consists of three parts: First, the lowest level is the root of the tree, which is the technical part. It consists of sensors, embedded processors and connection parts, and it is the foundation on which the entire tree depends for survival and development. Secondly, above the root of the tree is the trunk, the part of the software. This is the trunk and central nerve of the tree. It includes device driver software, server software and application client software. Finally, at the top of the tree is the canopy, which is the part related to the applications. This is the result of the entire IoT system, which can be divided into two parts: industrial applications and civilian applications.

Applications and development of the Internet of Things [7]

Over time and technology advancement the applications of the Internet of Things have also changed. For example, in the past, what we call the IoT used to be a network based on wireless LAN technology or. Thus, an IoT terminal accessed a wireless router with cameras, smart lights and other hardware devices connected to Wi-Fi and controlled via Wi-Fi, just like a PC. Although Wi-Fi IoT is convenient, it consumes too much power and has a short transmission distance. Fortunately, there are other available technologies now. For example, with the development of LPWAN (Low-Power Wide-Area Network) Internet of Things technology and the development of 4G and 5G communication networks, the Internet of Things is slowly being accepted by the market.

Industry	Example for Application
Transportation	Auto-Parking, Logistics management, Cargo Tracking, Auto-navigation…
Environment	Environmental monitoring, Animal monitoring…
Public facilities	Smart street light, Surveillance cameras…
Medical	Tele-monitoring, Remote medical diagnosis…
Manufacturing	Automated industry, Goods management…
Business	ATM, POS machine, Vending machine…
Family	Smart home system, security monitor…

Table 2: Example for application of the IoT

1.2 Drones in the Internet of Things

In recent years, drones have begun to appear in our daily lives. It can be said that they are well-known and have become one of the current hot topics.

What is a drone?

Firstly, the DRONE is an acronym for “Dynamic Remotely Operated Navigation Equipment”. This basically means an aircraft without a pilot, controlled from the ground, used for taking photographs [8], dropping bombs[9], delivering goods[10], etc. [11] At the same time, it also has a professional name UAV, which is an ancronym for Unmanned Aerial Vehicle or Uncrewed Aerial Vehicle. It is an unmanned aircraft operated by radio remote control equipment and its own program control device, or it is operated completely or intermittently by an on-board computer. [12] [] Drones are also sometimes called aerial robots, unlike manned aircraft, drones are often better suited for those "dull, dirty or dangerous"

[14] tasks. According to the application field, drones can be divided into military and civilian. In military applications, drones are often used as reconnaissance and target aircraft. In some special cases, drones can even be used as attack aircraft (example: MQ-1 Predator USA [15]). In terms of civilian use, a demand model of "drones and industrial applications tied together" has been formed. At present, the applications in aerial photography, agriculture, plant protection, live broadcast, express transportation, disaster rescue and other fields have greatly expanded the use of drones.

Despite usage of proto-drones bating back to 1849, the world's first modern drone(pilotless aircraft) was born in 1916[16], but strictly speaking, in 1935, the British "Queen Bee" drone was considered a real drone. It can return to the take-off point, realize remote control operation, and fly autonomously and independently. [13] And during World War II, much more emerged-for training antiaircraft gunners and performing flight missions. "*Nazi Germany produced and used various drones during the war*"[13]. After World War II, jet engines were used in vehicles such as Australian GAF Jindivik and Teledyne Ryan Firebee I in 1951, while companies such as Beechcraft provided the US Navy with the Model 1001 in 1955 [13]. Until the Vietnam War, however, they were nothing more than remotely controlled aircraft. The actual launch of drones into combat began in the Vietnam War, and was mainly used for battlefield reconnaissance. Afterwards, in local wars such as the Middle East War, Gulf War, Kosovo War, Afghanistan War, and Iraq War (Second Gulf War), drones have appeared frequently and fought repeatedly [13]. Especially on the battlefield in Afghanistan, drones have become well-deserved protagonists and have successfully carried out "decapitation" operations on many occasions.

UAVs Classification

Depending on the application, drones typically fall into one of six functional categories [13]:

1. Target and decoy
2. Reconnaissance
3. Combat - Unmanned combat aerial vehicle (UCAV)
4. Logistics – delivering cargo
5. Research and development – improve UAV technologies
6. Civil and commercial UAVs

Meanwhile, vehicles can be categorized in terms of range / altitude. The following has been advanced as relevant at industry events such as ParcAberporth Unmanned Systems forum [13]:

"

a) *Hand-held 2,000 ft (600 m) altitude, about 2 km range*
b) *Close 5,000 ft (1,500 m) altitude, up to 10 km range*
c) *NATO type 10,000 ft (3,000 m) altitude, up to 50 km range*

d) Tactical 18,000 ft (5,500 m) altitude, about 160 km range

e) MALE (medium altitude, long endurance) up to 30,000 ft (9,000 m) and range over 200 km

f) HALE (high altitude, long endurance) over 30,000 ft (9,100 m) and indefinite range

g) Hypersonic high-speed, supersonic (Mach 1–5) or hypersonic (Mach 5+) 50,000 ft (15,200 m) or suborbital altitude, range over 200 km

h) Orbital low earth orbit (Mach 25+)

" [13]

UAVs Components

In general, we can divide a drone into 5 subsystems: power subsystem, flight control subsystem, sensor subsystem, body or structure subsystem, and remote control subsystem.

- The main function of the power system is to provide power for the drone's flight. It mainly includes the engine, electronic governor, and power supply. At present, civil engines are mainly propeller engines (motor-driven propellers). In the military field, jet or turbofan engines are used in addition to propellers. The electronic governor is actually a component that controls the output power of the engine. In the power supply section, small UAVs mostly use lithium-polymer batteries, while large UAVs use traditional aircraft engines and usually use fuel oil as their energy source.
- The flight control system is a system that can stabilize the drone's flight status and can control the autonomous or semi-autonomous flight of the drone. It is the "brain" of the drone. The performance of drones mainly depends on the flight control system, which is also the core technology of drones. Mainly composed of micro CPU and other computing equipment.
- If the flight control system is the "brain" of the drone, then the sensor system is the "eyes" and "ears" of the drone. Position and movement sensors give information about the aircraft state. Exteroceptive sensors deal with external information like distance measurements, while expropriocepitive ones correlate internal and external states [17] The sensor system mainly includes GPS, gyroscope, infrared ranging sensor, acceleration sensor, etc.
- The body or structure part is equivalent to the human skeleton and constitutes the overall structure of the drone. Unlike manned aircraft, the main difference is that there is no cockpit area and its windows. Usually we can see drones with quadrotor structure, which is also the common appearance of current small drones. Single-rotor and double-rotor (it seems like helicopters) or general aircraft structures (RQ-4 Global Hawk, USA) are generally common structures of large drones.
- The remote control part mainly refers to the part that accepts remote commands, which is used by humans to control the drone in real time, so as to accurately complete various flight actions.

Ordinary drones generally accept radio wave signals or infrared signal. Military drones can also receive satellite signals, etc.

In addition, drones can also have other equipment for different missions. For example, the high-definition cameras and/or cameras used for observation, the hooks used for express cargo, military weapons, etc.

Applications of drones and their advantages and disadvantages

As mentioned earlier, drones can do some "dull, dirty or dangerous" tasks [14]. Therefore, in modern society, drones are widely used in both civilian and military applications. For example:

In the civilian area:

a) Electricity patrol inspection work: Through Unmanned Aerial Vehicles equipped with high-definition cameras and GPS systems, drones can perform autonomous positioning cruise along the power grid, real-time transmission of video recording and monitoring of power grid safety
b) Aerial photography work: A drone can be equipped with a high-definition camera, and in the case of wireless remote control, according to the shooting needs, the remote control is used to shoot from the air.
c) Delivery work: Drones can deliver goods of appropriate weight and size. You only need to enter the GPS address of the recipient into the navigation system, and the drone can take off for delivery tasks.
d) Disaster prevention and post-disaster rescue: Utilizing drones equipped with high-definition cameras to prevent natural disasters such as forest fires and floods, as well as search and rescue after disaster.

In military areas, a drone can be used for battlefield reconnaissance and surveillance, location shooting, damage assessment, electronic warfare, border patrol, nuclear radiation detection[18], etc. It can also be used for civilian purposes, such as traffic patrols, security monitoring, criminal hunting, etc.

UAVs are widely used, with low cost and high efficiency. They can work in high-risk or hard-to-reach areas without risk of personal injury or death [19]. They have diverse equipment deployment and good adaptability. At the same time, the drone can take off and land at a fixed point, and there are not many requirements for the landing point. It can be remotely controlled by radio or by an on-board computer.

But drones also have disadvantages. Drones have low survivability, poor stability, and are easily affected by weather. They are slow and susceptible to signal interference. There are also many

technical and economic issues. Also, due to the popularity of drones, ethical issues such as personal privacy need to be improved and resolved. [19]

1.3 Contribution from the Project

This Bachelor Thesis is focused on the application of IoT drones in civilian areas. The main research focus is on the application of UAVs into the IoT domain. It aims to use the "UAV + IoT" model for current applications and development activities. In the thesis experiments, miniature open source drones made with low volume devices are used. “Open source UAV” is open source on the flight control system and an open source autopilot project based on the concept of open source, which also includes open source software and open source hardware. In recent years, open source UAV flight control systems have greatly contributed to the rapid development of UAVs. Currently there are four representative projects: the Paparazzi UAV, Dronecode / PX4, OpenDroneMap, and the UAV News Lab Operator's Manual [20].

This project was tested using the TYRO69 open source lightweight UAV. Compared to other drones, TYRO69 has a lighter weight and smaller size as well as a better price/performance ratio (47.9g. Drone Wheelbase: 105mm, 70 Euros [49]). Although the drone is small, it is very well equipped. As a result, we can effectively reduce the design research and assembly costs of the UAV. And due to the small size of the components, we can also explore and study the miniaturization of the UAV.

At the same time, we analyze the technical status of UAVs in IoT monitoring and observation for IoT direction. And we will identify some open problems found so far and try to solve and improve them. In the following sections, a more detailed and complete step-by-step installation of the UAV software and hardware will also be shown.

The paper is structured as follows: the second section discusses state-of-the-art computers or CPUs in UAVs, system autopilots and some UAVs of the Internet of Things. Sections III and IV describe the drone hardware and software assembly process, followed by Section V for flight testing of some drones. Section VI is composed by the conclusions of this Bachelor Thesis.

2. Technological framework

2.1 Computer or Main-board and Autopilot in the UAVs

As the "brain" of a drone, it must be an essential part of the drone. There are many chip companies in the world that produce UAV master chips:

For example, in the United States we have Qualcomm, Inter, Texas Instruments, NVIDIA and Atmel, in Europe STMicroelectronics and ARM, in South Korea there is Samsung, and in China we have Huawei and DJI as representatives of the company.

In the field of civilian drones and Internet of Things (and considering the economic cost) currently Arduino, Raspberry Pi, STM, BeagleBone, Qualcomm, ARM and other microcomputers or motherboards are used as drone computers. The next subsections will introduce the following common drone computers or motherboards:

Arduino platform

Arduino is an open-source hardware and software company based on a project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. The Arduino project started in 2003 as a student project of the Ivrea Interaction Design Institute in the Ivrea region of Italy [21]. "*The aim is to provide a novice and professional with a low-cost and easy way to create interactions between sensors and the environment Functioning device actuator. Common examples of such devices for beginners include simple robots, thermostats, and motion detectors. Generally, use Atmel AVR single-chip microcomputer, adopt open source software and hardware platform, build on open source simple I / O interface board, and have Processing / Wiring development environment using similar Java and C language"* [21]. Arduino board design uses a variety of microprocessors and controllers. These boards are equipped with a set of digital and analog I / O pins and can be connected to various expansion boards or breadboards-like circuits with additional functionalities (shields) and other circuits. These boards have serial communication interfaces, including Universal Serial Bus (USB) on some models, and are also used to load programs from personal computers. Microcontrollers typically use sketches resembling C / C ++ programming language. In addition to using the traditional compilation toolchain, the Arduino project also provides an integrated development environment based on the Processing language project. [21]

With the rise of drones, as mentioned earlier, more and more open source projects provide us with more references and options for DIY drones. Arduino (Paparazzi) is the earliest open source flight control. It is a flexible open source hardware platform and open environment. Users can obtain hardware design documents from the official website and adjust circuit boards and components to meet their actual design needs. "*Users can also communicate with Arduino View and upload the code you wrote in the accompanying Arduino IDE software. The Arduino IDE is based on the processing IDE. For beginners, it is easy to master and has enough flexibility. The Arduino language is developed based on the wiring language. It is a secondary encapsulation of the avr-gcc library"* [21]. It does not require too much of a microcontroller and programming foundation. After simple learning, it can be

developed quickly. And Arduino IDE can run on mainstream operating systems such as windows, Macintosh and Linux. [21]

In order to get better flight control design source code, Arduino company decided to open its flight control source code, they opened the development path of open source flight control. Many well-known open source flight controllers such as APM (belongs to ArduPilot) are direct derivatives of Arduino flight control, and still use Arduino development environment for development. [37]

This paper mainly analyzes the economic personal drones for the Internet of Things. Considering the economy and the size of the drone, it usually coincides with the small size and low price. For example, Arduino Nano, Arduino Uno, Arduino Mega 2560, etc. are used as the main-board of the UAV, or the computer. Of course, it can also make a simple Arduino PCB version according to the actual situation. Generally speaking, the board of Arduino usually uses Atmel's products as the CPU. According to different tasks, different types of CPU chips are used to meet the task requirements. At the same time, according to the open version of the model, there are different numbers of interfaces and functions. For example, the typical board Arduino Uno is a microcontroller board based on the ATmega328P [22]. It has 14 digital input/output pins, where 6 can be used as Pulse Width Modulation (PWM) outputs, 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started [22]. Through these I / O interfaces, we can effectively connect various parts of the drone, thereby achieving control of the drone.

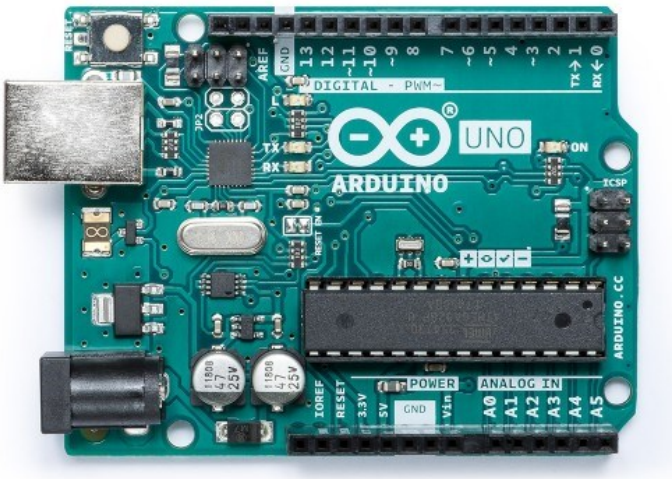

Figure 2: Arduino Uno Rec3[22]

Raspberry Pi platform

Similarly, as the world's most famous computer learning device, the Raspberry Pi can also be used as the central processor of a drone like Arduino. "*The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated, selling outside its target market for uses such as robotics"* [23]. In 2006, the early Raspberry Pi concept was derived from Atmel's ATmega644 microcontroller. Its schematics and PCB layout are available for public download [23]. In March 2012, Eben Epton of the University of Cambridge, UK officially launched Raspberry Pi as the world's smallest desktop computer, also known as a card computer, which is only the size of a credit card but has all the basic functions of a computer. This is the Raspberry Pi motherboard. [23]

It is an ARM-based microcomputer motherboard, with SD / Micro-SD card as the memory hard disk. The card motherboard has USB ports and an Ethernet interface (A type has no network port), which can be connected. Keyboard, mouse, and network cable, as well as a TV output interface for video analog signals and an HDMI high-definition video output interface can be plugged. [23] All the above components are integrated on a motherboard that is only slightly larger than a "*credit card"*. With all the basic functions of a PC, it just has to be connected to a screen and a keyboard to perform functions such as spreadsheets, word processing, videogames, and playing high-definition video. As technology advances, the Raspberry Pi has introduced many models, faster running speed and more functions. By the end of June 2019, the Raspberry Pi Foundation released the Raspberry Pi 4, and the processor was upgraded to 1.5GHz Broadcom BCM2711 (quad-core Cortex-A72), which increased the onboard memory capacity to 1/2/4GB. Bluetooth has been upgraded to 5.0, with 2 USB2.0 interfaces, 2 USB3.0 interfaces, and the power supply using the newer USB-C interface. [23]

As a complete computer, the Raspberry Pi has its own operating system. The Raspberry Pi Foundation provides Raspbian, a Debian-based (32-bit) Linux distribution for download [23]. At the same time, Raspberry Pi also recommends many third-party OSs, such as Linux-based systems such as Ubuntu, as well as Windows 10 IoT Core and Windows 10 ARM systems. [23]. Just like any other desktop or laptop computer running Linux or Windows, you can do many things with Raspberry Pi, although on little bit different way. Common computer motherboards rely on hard disks to store data, but Raspberry Pi uses an SD card as a "hard disk", even though a USB hard disk can be connected as well.

For example, as the smallest in the Raspberry Pi series, the Raspberry Pi Zero, is half the size of a Model A + (65 × 30 mm [23]), but with twice the utility. It has 1GHz single-core CPU, 512MB RAM, with mini HDMI port and micro USB OTG port, use micro USB power, compatible with up to 40-pin connectors and costs only $ 5. [23]

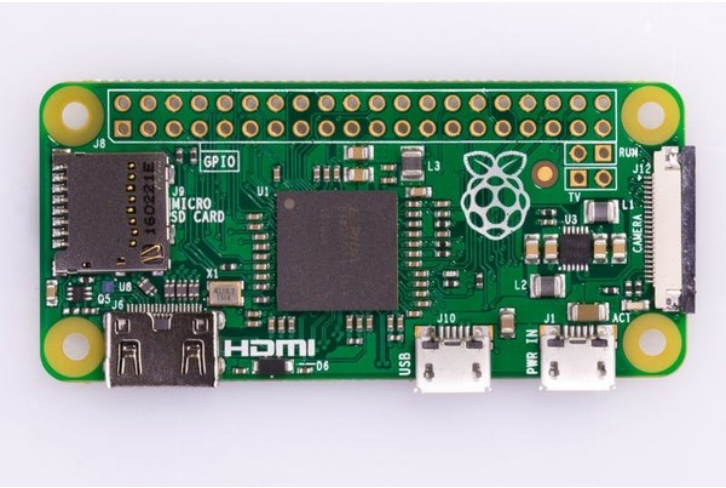

Figure 3: Raspberry Pi Zero [23]

BeagleBoard platform

BeagleBoard, like Arduino and Raspberry Pi, is a well-known and commonly used open source hardware platform. "*The BeagleBoard is a low-power open-source single-board computer produced by Texas Instruments in association with Digi-Key and Newark element14. The BeagleBoard was also designed with open source software development in mind, and as a way of demonstrating the Texas Instrument's OMAP3530 system-on-a-chip (first version)*" [24]. Similar to Raspberry Pi, the BeagleBoard is also a card computer. The BeagleBoard is not only open source software, but also open source hardware. The Raspberry Pi's SoC (System on a Chip) is a BCM series [23]. Broadcom's series of chips are exclusively for the Raspberry Pi, which means that this chip cannot be used on other boards and is hard to be bought online. Other BeagleBoard-based developments might be different. Take BeagleBone Black as an example. Its chip AM335X is from Texas Instruments and it can be bought online. It means that it can be used for development and expanded on the basis of BeagleBone Black, and installed in a board with a different shape and bought to have a custom system.

"BeagleBoard has all the functions of a basic computer, including an ARM Cortex-A8 series CPU (can run Linux, Minix, FreeBSD, OpenBSD, RISC OS, or Symbian; there are many unofficial Android ports)," [24] and some models have functions for accelerating video and Audio decoded, TMS320Cxx DSP and GPU for Imagination Technologies and PowerVR SGX530 or SGX544 for providing accelerated 2D and 3D rendering with OpenGL ES 2.0 support. It is available with separate S-Video and HDMI connections or via expansion card. Provide SD / MMC card slot supporting SDIO, USB On-The-Go port, RS-232 serial connection, JTAG connection and other devices [24]. The built-in storage and memory are provided through a PoP chip, which "*includes 256 MB of NAND flash memory and 256 MB of RAM (128 MB for earlier models). The evaluation board can use up to 2 W of power*

and can be powered via a USB connector or a separate 5 V power supply. Due to the low power consumption, no additional cooling or heat sink is required". [24]

As the host computer of the drone, unlike the Raspberry Pi or Arduino platform, which needs to connect various sensors and other components, the BeagleBoard platform has more types of interfaces and integrates various sensors. For example, BeagleBone Blue.

BeagleBone Blue [25] is a "*Linux-based all-in-one computer, integrating onto a single small (3.5" x 2.15") board the Octavo OSD3358 microprocessor (The Octavo Systems OSD335x System-in-Package (SiP) device is the first device in the OSD335x Family. It integrates the Texas Instruments Sitara ARM® Cortex®-A8 AM335x Processor, DDR3 memory, TPS65217C PMIC, TL5209 LDO, and all needed passive components into a single 27mm X 27mm BGA package [26]) together with Wi-Fi/Bluetooth, BMP280 barometer, MPU9250 for accelerometers, gyroscope and internal compass, power regulation and state-of-charge LEDs for a 2-cell LiPo, H-Bridges, and discrete connectors for 4 DC motors+encoders, 8 servos, and all of the commonly-needed buses for additional peripherals in embedded applications".* Although it is optimized for robotics, this compact and inexpensive board has all the necessary sensors and peripherals required for a flight controller.

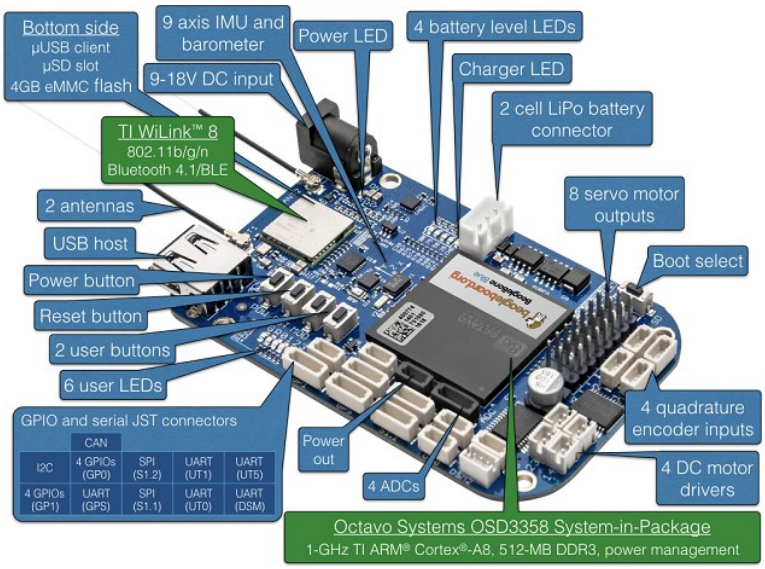

Figure 4: BeagleBone Blue[25]

Other commonly used platforms

In addition to the three platforms mentioned above, there are other platforms for open source drone projects. Here are a few more typical platforms:

1. **FPGA-based platforms (Field-Programmable Gate Array)**

   OcPoC (Octagonal Pilot on Chip)[27] was developed by Aerotenna and was established in 2015. It is located in the Center for Biological Science and Technology Business of the

University of Kansas. OcPoC has expanded its input and output capabilities to include fully programmable PWM, PPM and GPIO pins to integrate with a large number of different sensor add-ons. It also includes many other standard connectors for peripherals, such as GPS, CSI camera links and SD cards. It runs the ArduPilot software platform.

As the first FPGA-based platform, it has two modules: Xilinx Zynq Processor and Altera Cyclone V (Intel), equipped with an ARM Cortex-A9 processor [27]. "*It allows real-time signal processing and can run complex algorithms, bringing new things to artificial intelligence, deep learning, and truly automated, intelligent drones.*" It has more than 30 programmable I / O, supports most standard interfaces, and is very flexible. M8N GPS navigation system, MPU9250 IMU, MS5611barometer, etc., it can freely develop the creativity of developers. Also integrates seamlessly with Aerotenna microwave radar sensors, including uLanding altimeters and uSharp obstacle avoidance sensors. Compatible with AMP or PX4.

Figure 5: OcPoC™ with Altera Cyclone® mini SoC Flight Controller[27]

2. ARM-based platforms

a) Pixhawk/PX4

It is a computer vision research-based flight control designed by the Computer Vision and Geometry Laboratory and the Autonomous System Laboratory of the Federal Institute of Technology in Zurich [28]. It is a PX4 flight controller system and modified connections. It consists of a PX4-Flight Management Unit (FMU) controller and a PX4-

IO integrated on a motherboard, with Additional IO, memory and other functions. In addition, it works closely with the Linux Foundation DroneCode project.

Figure 6: 3DR Pixhawk Flight Controller [29]

b) Paparazzi

This is the first and oldest open source drone hardware and software project. Since 2003, it has been developed at the French National Aviation University (ENAC) drone laboratory [30]. It includes (integrated) autopilot systems and ground station software for multi-helicopter / multi-rotor, fixed-wing, helicopter and hybrid aircraft [31]. In March 2017, ENAC Labs released a new autopilot called Chimera, which is based on the latest STM32F7 microcontroller [21].

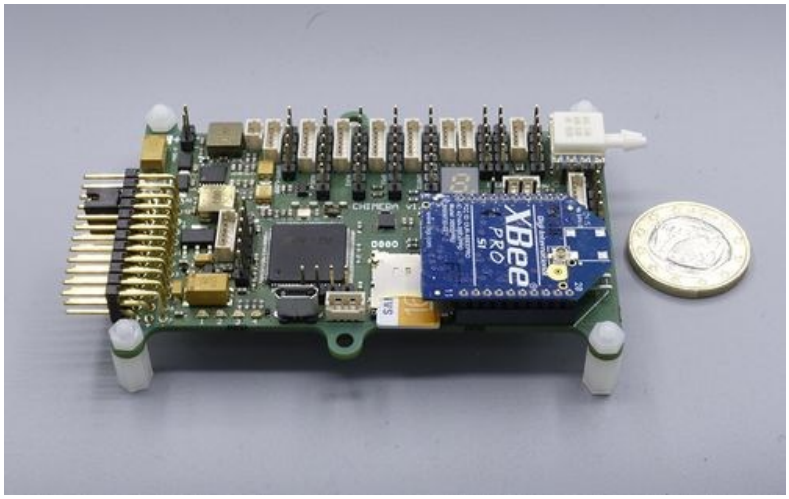

Figure 7: Paparazzi Chimera V1.0 autopilot board[30]

"One of the great advantages of Paparazzi is support for many hardware designs. There is autopilot hardware that is specifically designed to work with Paparazzi from the start, but Paparazzi has also been ported to other popular autopilots."

Most current autopilots are designed around two primary processors:

STM32 series microcontrollers (ARM-Cortex Series MCU based) [30]:

Tawaki, Chimera, Lisa Series, OpenPilot, etc.

LPC21xx series microcontrollers (NXP LPC21xx MCU based) [30]:

NavGo, Umarim Series, Tiny Series, YAPA Series, HB Series, etc.

c) **CC3D and Atom**

They are two flight controllers with the same function but different sizes. They were developed by OpenPilot, which recently became LibraPilot. CC3D and Atom flight controllers have all types of stable hardware running OpenPilot / LibraPilot firmware. They can be configured (installed) using OpenPilot / LibraPilot to fly any airframe from a fixed-wing to an octo-rotor. The project's hardware and software are available under the GPLv3 license [32].

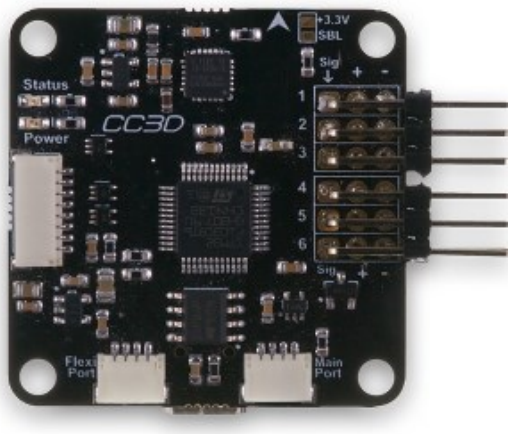

Figure 8: OpenPilot CopterControl3D/CC3D autopilot board[32]

3. **Qualcomm Snapdragon Flight platform**

"The Qualcomm Snapdragon Flight [33] platform is a high-tech autopilot / airborne computer. The DSPAL API can be used to implement the PX4 Flight Stack on the DSP on the QuRT real-time operating system for POSIX compatibility. Compared to Pixhawk, it adds a camera, WiFi and high-end processing power as well as different IOs."

It is also an integrated platform for Computer + Autopilot, it is also an integrated platform of computer + autopilot. It adopts Qualcomm's Snapdragon series CPU and necessary flight sensors, and is also equipped with complex equipment such as cameras, wireless network connections, GPU, etc. It is a very advanced platform and is generally used in commercial drones. (DJI, Skydio,etc.)

Figure 9: Qualcomm® Snapdragon ™ Flight Kit[33]

Arduino platform vs. Raspberry Pi platform vs. BeagleBoard platform

As explained before, Arduino is a microcontroller-based electronic development board that can run some relatively simple applications. Arduino cannot run an operating system, which means that many functions based on the operating system cannot be obtained with Arduino. Because there is no operating system, Arduino's programs run more directly and are more suitable for communicating with sensors and peripheral circuits. The program size is small and the program running efficiency is relatively high. Arduino's program design is relatively simple, and its program needs to be connected to a computer via a USB interface for uploading. Arduino does not have audio and video output interfaces, nor does it have a network interface and a USB interface that can be connected to peripheral devices. There is no extra onboard memory on the development board. Of course, these functions can be realized through the expansion of the Arduino expansion board. The Arduino expansion board can deal with the expansion of various functions. There are many types of expansion boards. In addition to the expansion interface function, the expansion board can also expand GPS, motor control, network connection and other functions. [21]

Unlike Arduino, Raspberry Pi is an ultra-small computer. It needs a TF card to load the operating system to run. Raspberry Pi uses ARM as the processor, and also has on-board memory, HDMI interface, USB interface, audio interface, network interface and so on. Although Raspberry Pi also exports a general-purpose input / output interface (GPIO) that can be used to connect peripheral devices and sensors, it requires the support of the operating system, so the operation of the Raspberry Party GPIO is not as convenient and direct as Arduino. Some Raspberry Pi products have an on-board network connection function. After completing the settings, we can remotely access the Raspberry Pi through a computer to achieve control and program design. [23]

The BeagleBoard is an open source project for educational purposes supported by Texas Instruments, such as the BeagleBone mentioned earlier. Similar to the Raspberry Pi, the BeagleBone is also based

on the Linux system and also uses an ARM architecture processor [24]. It has more interfaces and is equipped with an on-board memory chip. It is twice as fast as the Raspberry Pi (at least its earlier version) [15] [16], and the price is much higher than the previous two (the specific price varies depending on the model). BeagleBone has the good scalability of Arduino and both powerful performance and flexible development environment of Linux, so it is suitable for projects that require extended peripherals and more complex functions.

In consideration of drones being as portable (lighter and smaller) as possible and cost considerations, the first step is to eliminate Beagle, because this project does not require the powerful performance of Beagle, and at the same time, Beagle is much more expensive than the other two platforms [34]. Finally, we chose the Arduino Nano Every model [35] (abbreviated as Nano) and the Raspberry Pi Zero [36] (abbreviated as Zero) for comparison. First, according to the official price, the official price of Nano in Europe is 8 euros [35], and the price of Zero is about 5,4 euros [36]. Although there is still a price difference close to 3 euros, the small price difference can be ignored considering the overall price of the DIY drone (about 70 euros). Secondly, Nano measures 45 × 18mm and weighs only 5 grams [21], while Zero measures 65 × 30 mm and weighs only 9 grams [23]. Both platforms are very portable. Although in terms of portability and price it is already known that the Zero and Nano are small and cheap, there are big differences in performance. As mentioned earlier, Raspberry Pi and Arduino are two platforms with completely different computing levels. Zero can run a complete operating system (Linux) and can run multiple processes at the same time. As a typical microcontroller, Nano can only run one program that has been programmed at a time. In addition, Zero has a more comprehensive interface, including USB, SD card and other interfaces, but Nano is relatively simple and requires the purchase of some additional interface boards. After all, Zero is a complete computer (of course, it is not enough to compare with PC and MAC). It uses an ARM1176JZF-S CPU with a frequency of 1GHz for operation [23]; and Nano is a micro-control The device is only a part of a computer, using the ATMega4809 chip [35], the highest operating frequency is only 20MHz. From this we can see that the processing speed of Zero is 50 times faster than Arduino Nano. At the same time, Zero also has more than 10,000 times of Nano memory (512MB [23] vs. 48 KB [35]). But it can't be said that Nano does not have any advantages. Although Zero has a complete operating system, which also means that it requires longer startup time, Nano's Arduino platform has a single purpose, "real-time" capabilities, and a fast start-up speed. At the same time, Nano is also simpler. Arduino's integrated development environment is much easier than Zero's Linux. It does not require software and code libraries and can be used directly.

Autopilot extension board (Autopilot Shield)

1. **PXFmini**

   "*The PXFmini (Pixhawk Fire Cape Mini) autopilot shield manufactured by Erle Robotics is a low cost, open-source autopilot shield for Raspberry Pi, enabling you to create an impromptu flying autopilot. The shield is specifically designed for the Raspberry Pi Zero, but it is also pin-to-pin compatible with other models in the Raspberry Pi family*." [46]

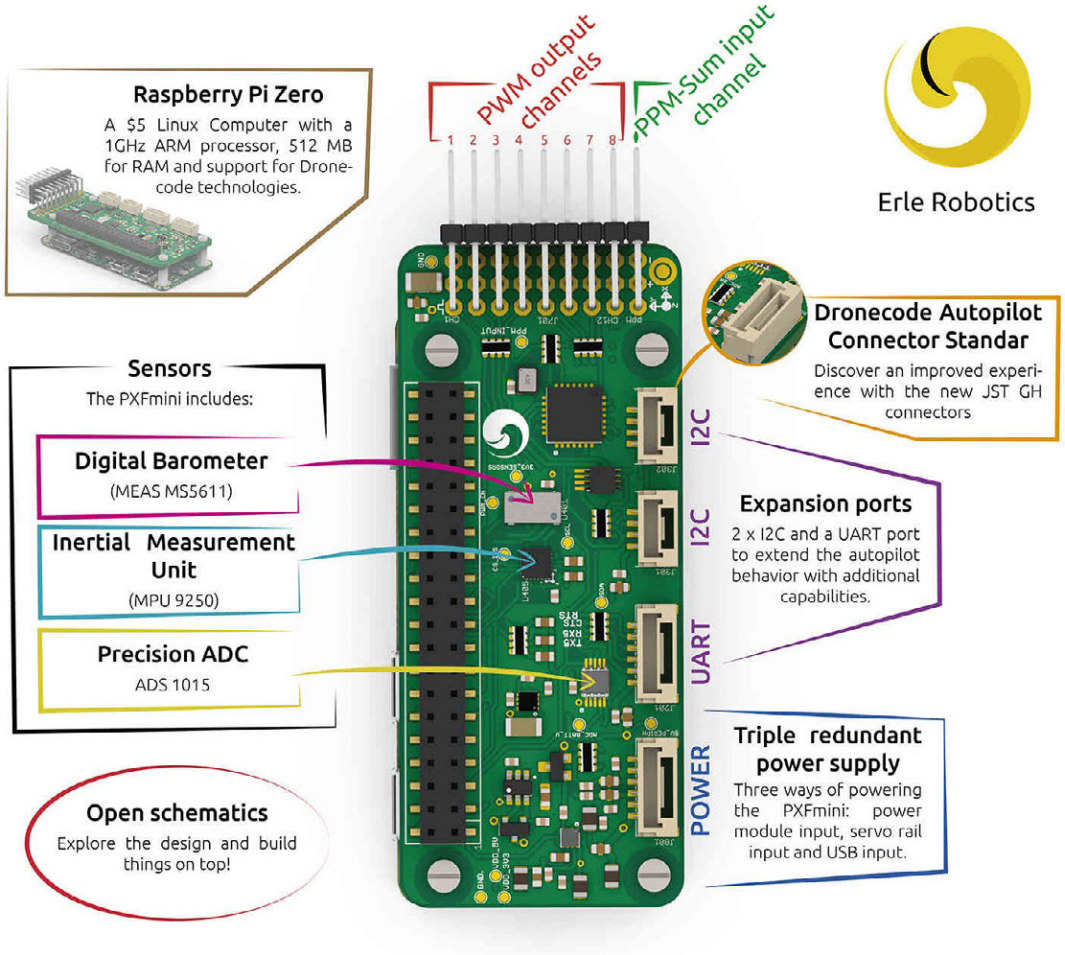

Figure 10: PXFmini autopilot shield [46]

The PXFmini shield only weighs 15 grams and embeds all the power electronics necessary through its I2C and UART ports to comply with most existing components of the drone. "*It has 8 PWM outputs, PPMSUM input, MPU9250 as the IMU, barometer MS5611, UART and I2C for extra extension. PXFmini includes 3 axes gravity sensor, 3 axes gyroscope, 3 axes digital compass, pressure sensor, temperature sensor and an ADC. It includes new JST GH connectors provide an amazing new experience*." [46]

2. **Navio**[37]

   Navio was developed by Emlid and used with Raspberry Pi. The latest model is Navio2, which is not only designed as a Linux version of the APM platform, but also as a platform

for any custom robot project. Navio2 does not require any onboard multiple controllers, as everything is packaged together (for use with the Raspberry Pi), which increases the robustness of the project and facilitates development. With Navio2, you a variety of mobile robots such as cars, ships, multi-rotors and aircraft can be controlled.

Compared to the previous generation of Navio, Navio2 has two IMU chips to improve flight experience and for redundancy. Improved performance of MS5611. It also had replaced the PCA 9685 PWM generator with a microcontroller to avoid problems with motors and servos. Navio + uses DMA to sample PPM signals, which is very system intensive. On Navio2, the microcontroller handles PPM / SBUS sampling while leaving the processor core of the Raspberry Pi 2 for you to perform tasks. And have better Linux integration [36].

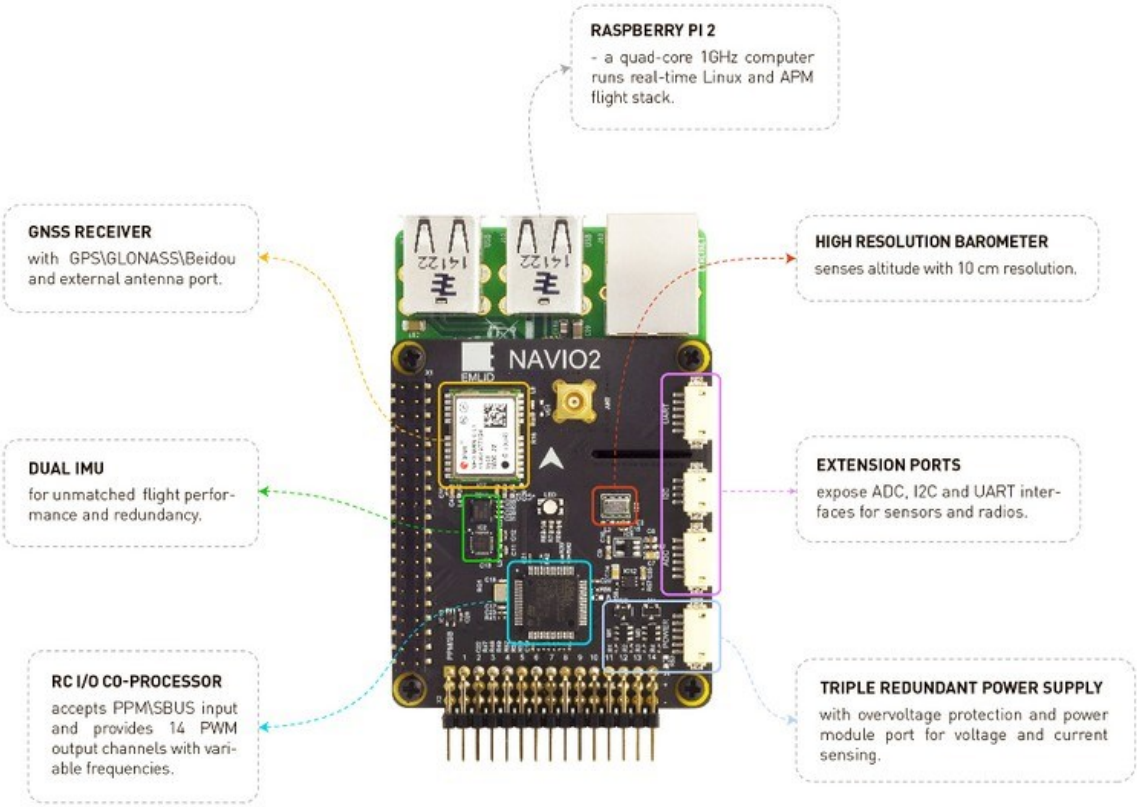

Figure 11: : Navio2 autopilot shield + Raspberry PI 2 [37]

The Navio2 weighs 23 grams and measures 55 × 65mm, it has an MPU9250 IMU and an LSM9DS1 IMU, barometer MS5611, M8N Glonass/GPS/Beidou navigation system, RC I/O co-processor. [37]

2.2 Open Source Drone Projects

Ardupilot/APM

"Ardupilot was one of the early pioneers of open source drone software, the project started with Arduino hardware (hence the name "Ardu") and has evolved into a standout in the drone industry."

[38] ArduPilot provides a complete set of tools for almost all vehicles and applications. As an open source project, it continues to evolve based on rapid feedback from a large number of users. The development team works with the community and business partners to add features to ArduPilot for the benefit of everyone. Ground control software, drones running ArduPilot can have advanced features, including real-time communication with operators. At the same time, ArduPilot has a huge online community dedicated to helping users solve problems and solutions.

Advantages: Strong support; large number of supported flight controllers and Ardupilot-supported hardware boards; "*very advanced control algorithm*"; suitable for many types of tasks; "*GPL license makes more community innovation for the benefit of all users*"; etc. [38]

Disadvantages: "*Limitations of GPL licenses (beneficial for businesses only)*" [38]; decentralized sensors and low integration.

Pixhawk/PX4

"PX4 is part of the Dronecode project, a non-profit organization managed by the Linux Foundation. Dronecode aims to provide an open source software platform for the emerging drone industry." [38] Dronecode provides the user with an "operating system" framework that allows users to make room for innovation on the platform. "*PX4 comes from the Pixhawk project, which is an open source hardware project because they need some open source drone software to run their circuit boards. One of the main differences between PX4 and ArduPilot is that it runs under a BSD license. This feature makes PX4 more attractive for businesses that want to protect their intellectual property. For this reason and including it in the Dronecode project, the PX4 received more funding for development work than ArduPilot*" [38]. A large number of flight control boards also support PX4.

Advantages: Strong support; industry backed project; compatible with many flight controller boards; available for many different types of vehicles; BSD license allows businesses to protect their IP; etc.

Disadvantages: Not for FPV.

Paparazzi [30]

Paparazzi (PPZ) is a fully open source software and hardware project. It started in 2003 and aims to build a very powerful and flexible autonomous driving system through input from the community and encouragement. A major feature of PPZ is that in addition to common flight control hardware, flight control software and ground station software, the open source flight control solution also can use ground station hardware, including various modems, antennas and other equipment. Functionally, PPZ is approaching a complete small UAV system. The flexibility of the PPZ is reflected in the fact that it can choose different aircraft and configure different flight control boards, sensors, communication

modules and peripherals according to the characteristics and focus of the application (flying speed, flight distance, dead time, load capacity, price). PPZ is different from ordinary pure code-centric open source software projects. It not only uses Github's managed code and hardware schematic CAD design drafts and other design materials. Successfully running the PPZ requires the motherboard to adapt to various peripheral sensors, servo motors, corrections, and commissioning, and to set the appropriate parameters for the aircraft carrier. PPZ has built its main website into a WIKI, so that anyone can register for an account and go up to edit and improve entries, supplement and share their experiences. The PPZ project is a large open knowledge base for drones, and individuals, companies, and research institutions can benefit a lot from it.

Advantages: Flexible; software and hardware combination; powerful and feature-rich; etc.

Disadvantages: Ubuntu Linux is difficult for beginners.

LibrePilot/OpenPilot[39]

The LibrePilot open source project was established in July 2015, it originated from the OpenPilot project, and its founding members are long-term contributors to the project. Openpilot is now disbanded. The project focused on the research and development of software and hardware that can be used in a variety of applications, including vehicle control and stabilization, driverless autonomous vehicles, and robotics. One of the main goals of the project is to provide an open and collaborative environment, making it an ideal place to develop innovative ideas. LibrePilot welcomes and encourages communication and cooperation with other projects, such as collaboratively adding support for existing hardware or software in the spirit of open source. LibrePilot source code is released under a GPLv3 license approved by OSI.

Advantages: Flexible; highly integrated; simple hardware architecture; etc.

Disadvantages: Wide variety, difficult to choose accurately

2.3 Use cases of UAVs in IoT

AGRAS MG-1S (Agriculture)

"*The Agras MG-1S integrates a number of cutting-edge DJI technologies, including the new A3 Flight Controller, and a Radar Sensing System that provides additional reliability during flight. The spraying system and flow sensor ensure accurate operations. When used with the MG Intelligent Operation Planning System and the DJI Agriculture Management Platform, a user can plan operations, manage flights in real-time, and closely monitor aircraft operating status*." [40]

EBEE SQ (Agriculture)

EBee SQ is the most effective high coverage agricultural drone on the market. This automated solution can cover hundreds of hectares of land in one flight. The eBee SQ includes an advanced Parrot redwood sensor. In addition to visible RGB images, it can also capture accurate multispectral data in 4 bands for effective crop monitoring and analysis. [41]

Camclone T21 (Infrastructure)

In 2009, CSIRO researchers developed a technology that enabled drones to perform tasks safely and reliably with a small amount of guidance from ground operators. "*The CSIRO information and communication technology (ICT) component of the SmartSkies project will enable drones to safely fly near structures such as power lines, axial, buildings, and cooling towers, and transit in coaxial airspace*." [42]

MD4-1000 (Delivery)

In December 2013, the DHL parcel service subsidiary of Deutsche Post AG tested a "microdrones md4-1000" for delivery of medicine. During a successful test flight at DHL headquarters, the Helicopter was delivered from a pharmacy across the Rhine. The helicopter covered the 2.7-km flight path in about five minutes and landed safely in the green space of the Posttower in Bonn. "The md4-1000 weighs approximately 2.6 kilograms, can stay in the air for 90 minutes, and has a top speed of 50 kilometers per hour. A helicopter can fly over the Rhine in just a few minutes [43]

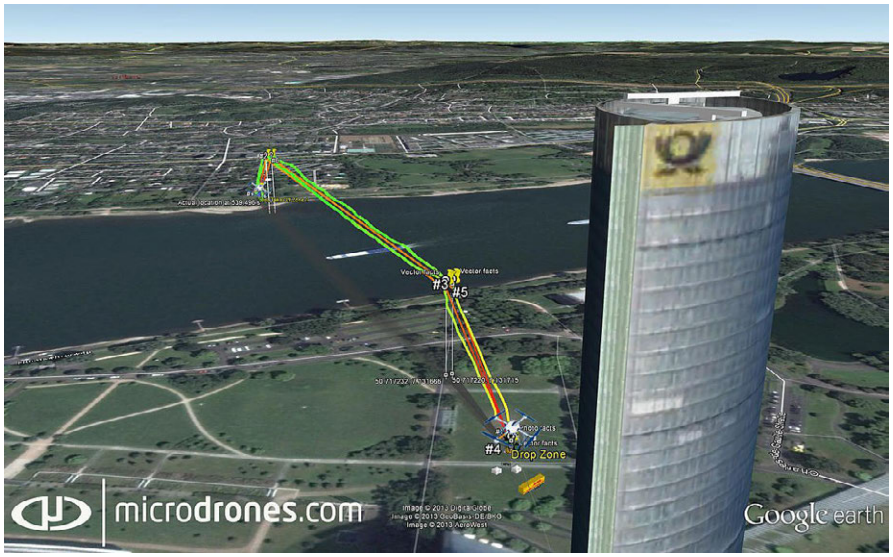

Figure 12: MD4-1000`s Test Flight [43]

Flying COW (Infrastructure)

In early 2017, AT & T "*successfully tested their ridiculous name Flying COW (Cell on Wings). This drone serves as a small alternative to a cellular tower and can be deployed at a disaster site or in a major event to distribute the load on a static tower.*" [44]

DJI Matrice210 (Maintenance and Infrastructure)

"*Telecommunications service provider Advanced Wireless and Technology Group is using DJI Matrice210 to carry its own proprietary AI (Artificial Intelligence) for a series of commercial operations.*" [45] These include: rapid emergency response, preventive maintenance, facility investigations, perimeter security and land investigations. [45]

"*Innovair is the UK 's leading developer of advanced robotic technologies, providing a range of land survey services for companies in the energy, construction and industrial sectors.*" [45] M210 is currently replacing the traditional ground survey technology used to investigate inventory in quarries to complete ground survey tasks faster and more accurately. [45]

Flyability Elios (Maintenance and Infrastructure)

"Aigües de Barcelona, Empresa Metropolitana de Gestió del Cicle Integral de l'Aigua, S.A., is the company responsible for a vast sewer infrastructure below the city of Barcelona and environs". In a storm in 2018, a sewer pipe was damaged. A pipe rupture caused 500 cubic meters of raw sewage per second to leak directly into the Mediterranean Sea. Due to the large amount of sewage in the pipeline, the traditional method of manual inspection is replaced by the Flyability Elios rapid inspection system. Flyability Elios can fly in tight spaces thanks to its protective frame. Drones can actually fly near walls, pipes, or even people without the risk of injury or crash." With Elios, the exact location and extent of the leak can be determined and an emergency bypass created by the engineer. This avoids the discharge of wastewater into the ocean and avoids major environmental disasters." [45]

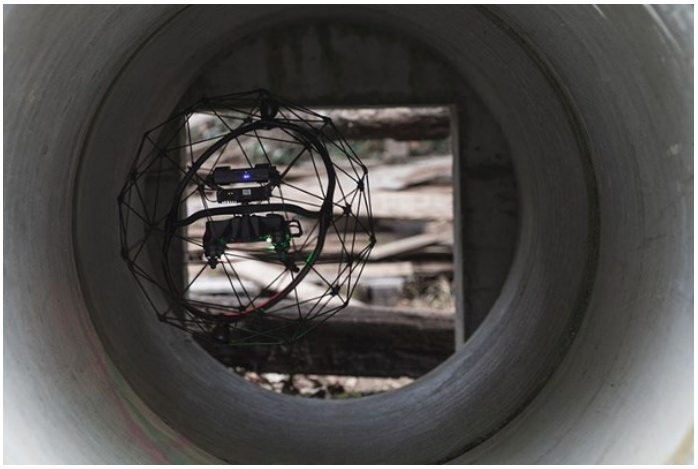

Figure 13: Flyability Elios [45]

3. Hardware assembly

3.1 Powered platform

Four-rotor UAV is a typical multi-rotor UAV with a simple structure, small size and stable flight. This Bachelor Thesis takes four-rotor UAV as an example to study the application of multi-rotor UAV in the Internet of Things. Considering the cost, the project research is carried out using EACHINE's TYRO69 platform.

EACHINE TYRO69 is a low-cost, small-capacity quadcopter that costs about 70 Euros and weighs only 47.9 grams [46]. It is lightweight and easy to carry. At the same time, it is provided as a kit to meet the needs of the project.

3.2 Preparation for installation

Drone main components

1. **Drone body or Frame kit:**

   "Axle Base: 105mm…Material: 3K Carbon Fiber and TPU 3D Printing" [47]

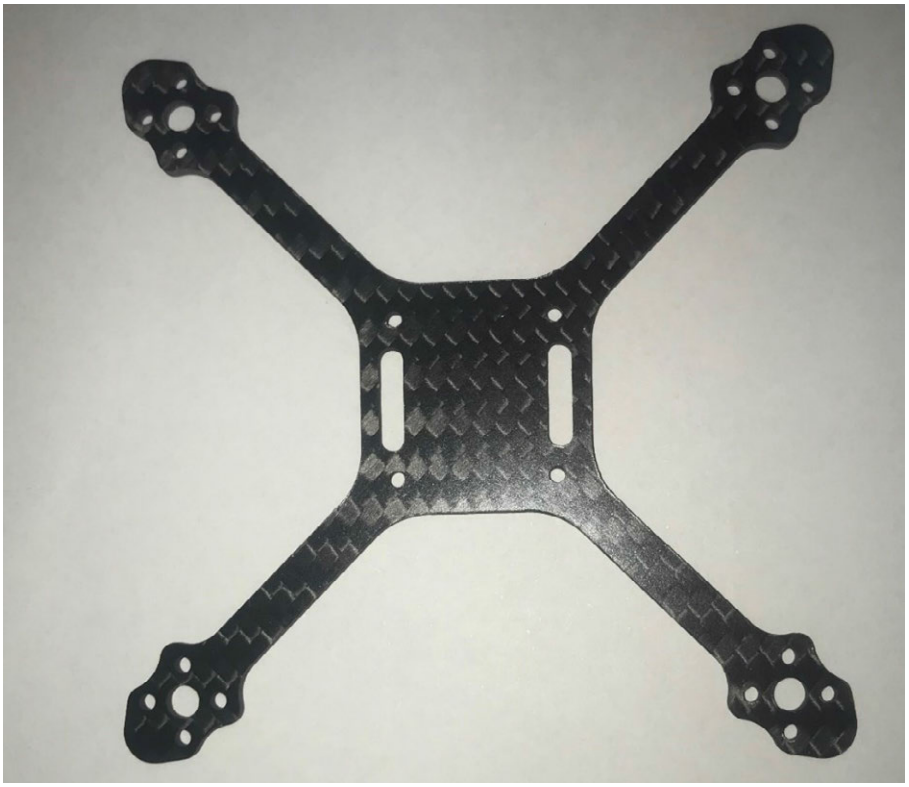

Figure 14: Frame kit

2. **Motors**

   "Model: 1104/8600 kV…Weight: 4.3G …Maximum power: 144W" [47]

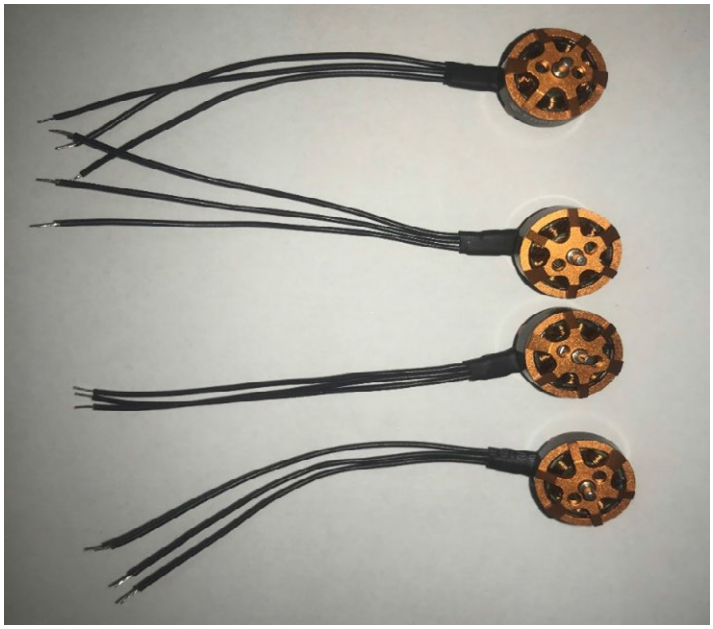

Figure 15: Motors

3. Electronic Speed Controller (ESC)

"Main Control Chip: EFM8BB21F16G…" [47]

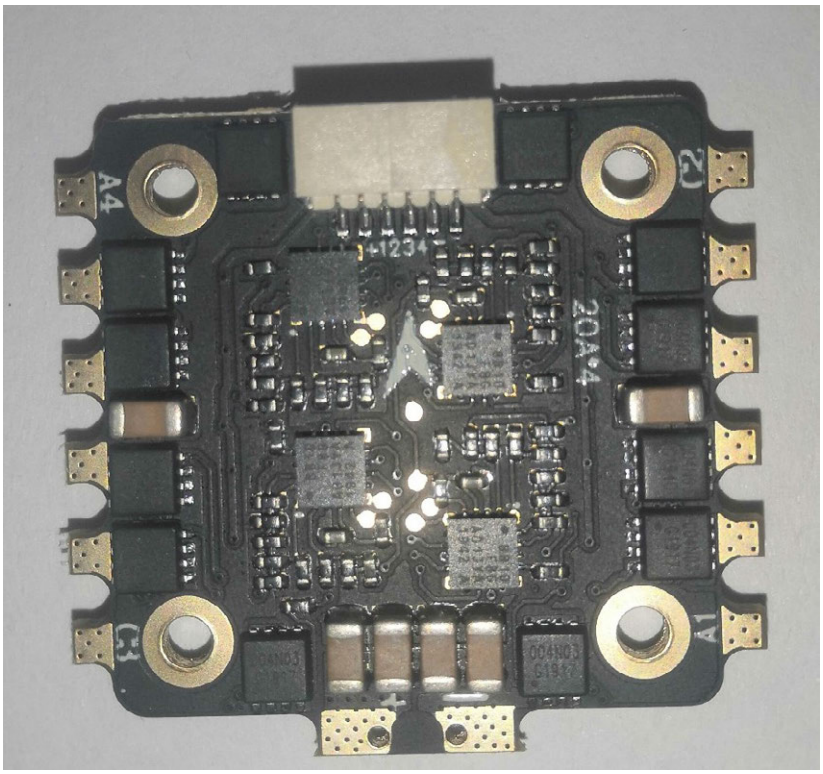

Figure 16: The top surface of ESC

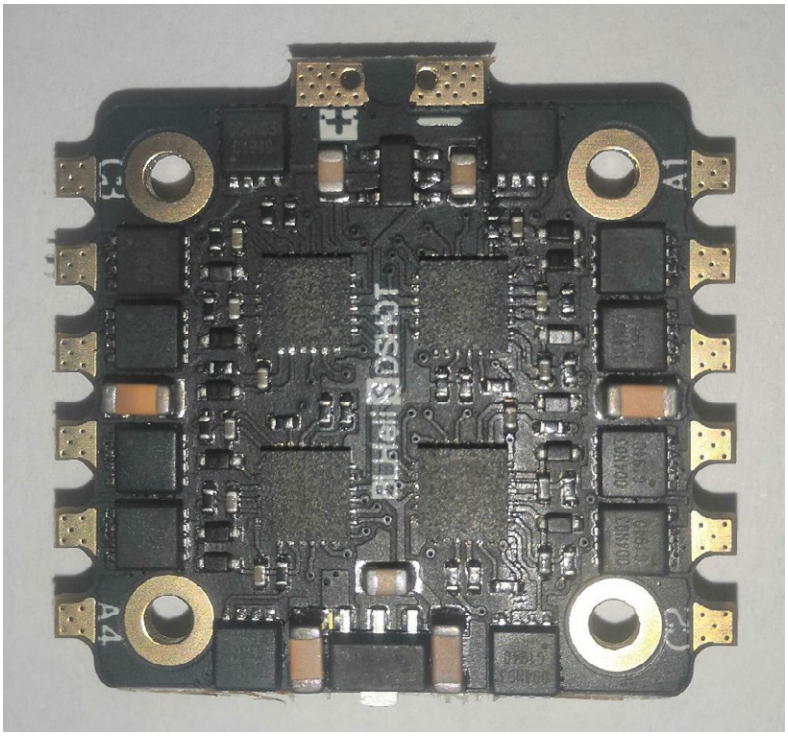

Figure 17: : The bottom surface of ESC

4. Flight Controller(FC)

"Main control chip: stm32f411… IMU: MPU 6000 6 axis sensor..." [47]

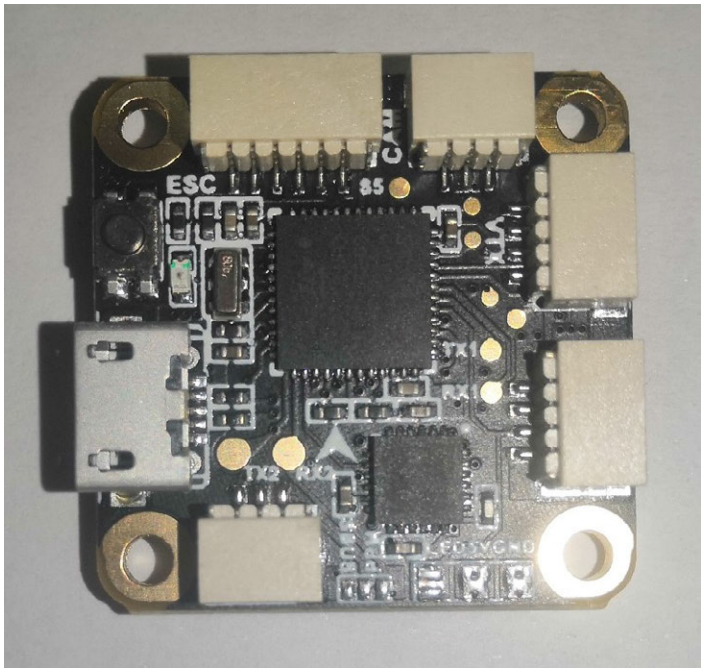

Figure 18: The top surface of Flight Controller(FC)

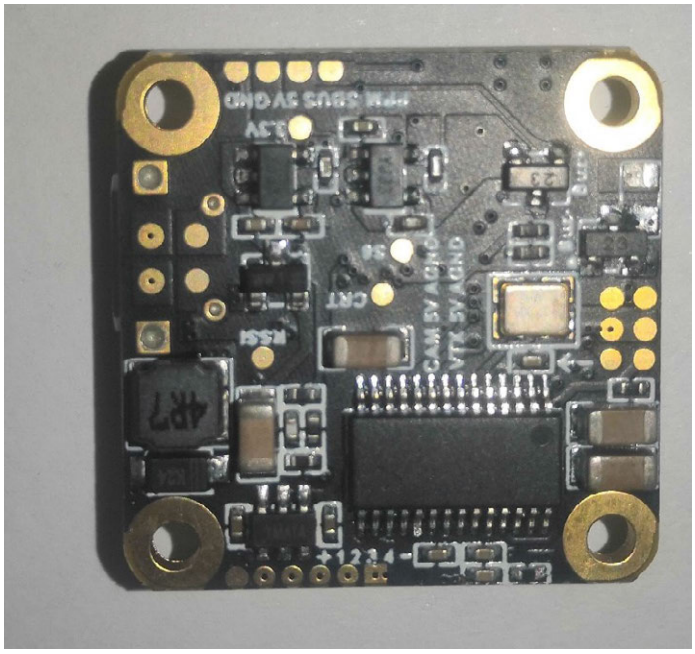

Figure 19: The bottom surface of Flight Controller(FC)

5. Camera

"Output power: 25 MW...Image Sensor: 1/3 "CMOS Sensor" [47]

Figure 20: The camera of drone

Installation tools

A welding station and welding materials, suitable screwdrivers, etc.

3.3 Installation process

A manual-like set of instructions follows to explain all the installation process

1. Welding motors and Electronic Speed Controller

   First of all, solder to the pads of ESC. For easier assembly, solder to the bottom of the pads.

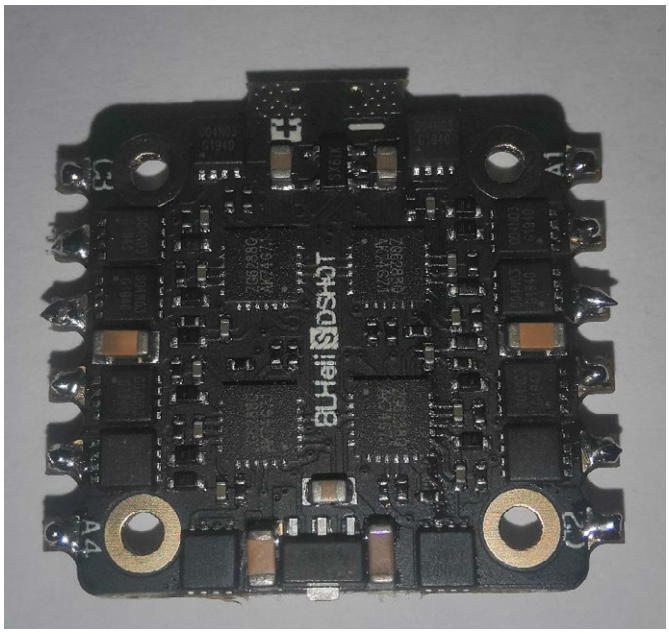

Figure 21: Solder to the bottom of the pads of ESC

   Then solder the wires of the 4 motors to the pads of ESC

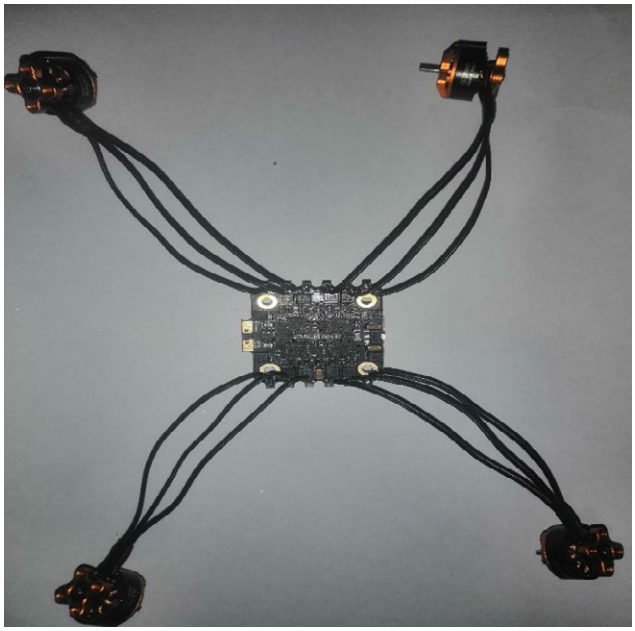

Figure 22: Welding of 4 motors

Finally, solder the power (battery) cable and protective capacitor (150uF 20V) to the ESC.

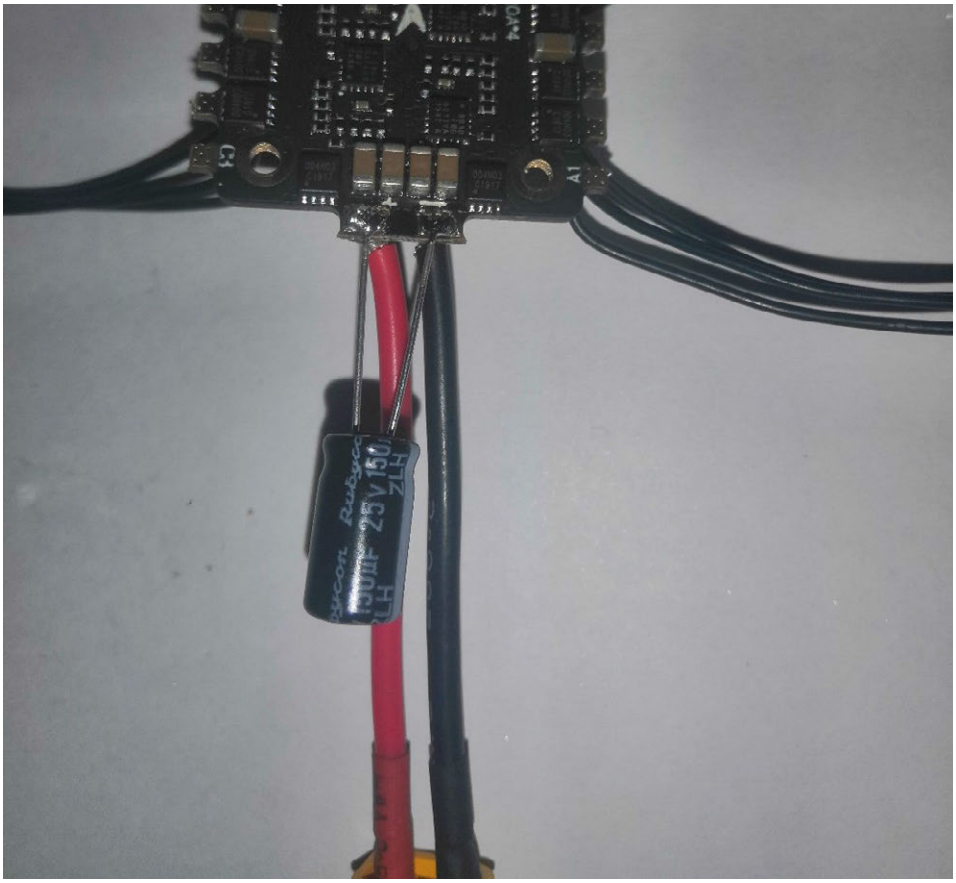

Figure 23: Solder power section (capacitors and wires)

2. Install ESC and motors on frame kit

   First install the 4 longest screws in the parts package on the frame kit, and at the same time install the shock-absorbing rubber on the screws.

Figure 24: Mounting screws and rubbers

Then install the ESC with the top facing up to the frame kit (Note: There is a white arrow on the top of the ESC, this arrow represents the front direction of the drone)

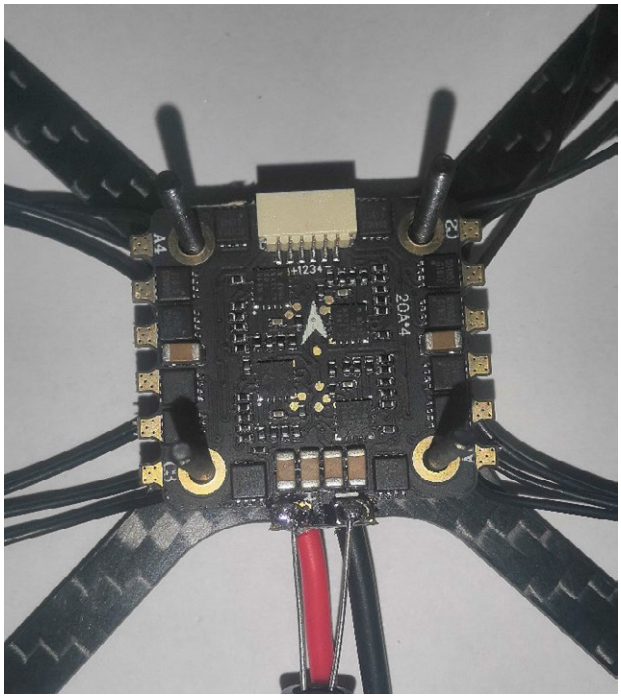

Figure 25: Install ESC

Finally, install the 4 motors on the 4 pins of the frame kit (PS: because the motor cable is too long, temporarily wrap the cable on the frame kit).

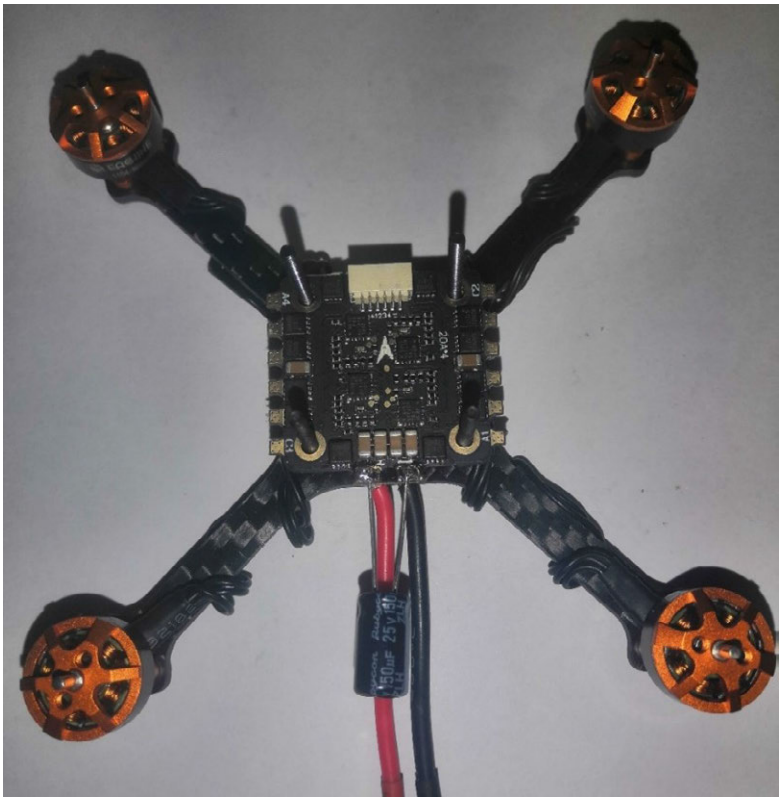

Figure 26: Mounting motors

3. Install Flight Controller and Connect ESC and Flight Controller

   First, it is the same as when installing ESC; the shock-absorbing rubber on the screws must be installed.

Figure 27: Installing a second level of rubber

Install the flight control board to the second floor, and then install the last layer of shock absorbers to the screws. And connect ESC board and FC board with data cable. Note: There is also a white arrow on the top of the FC board pointing directly in front of the drone, which must be in the same direction as the white arrow of the ESC board.

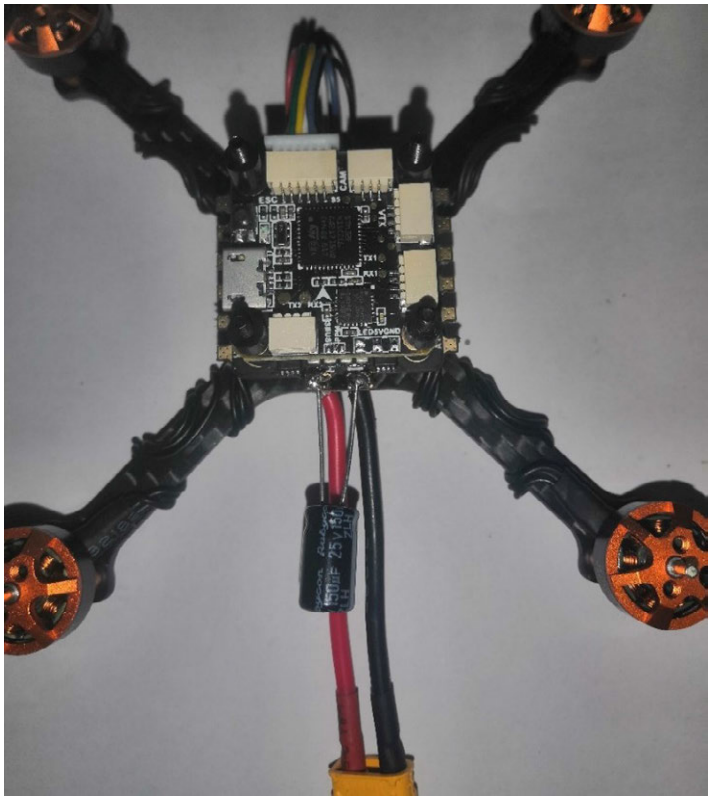

Figure 28: Installing FC

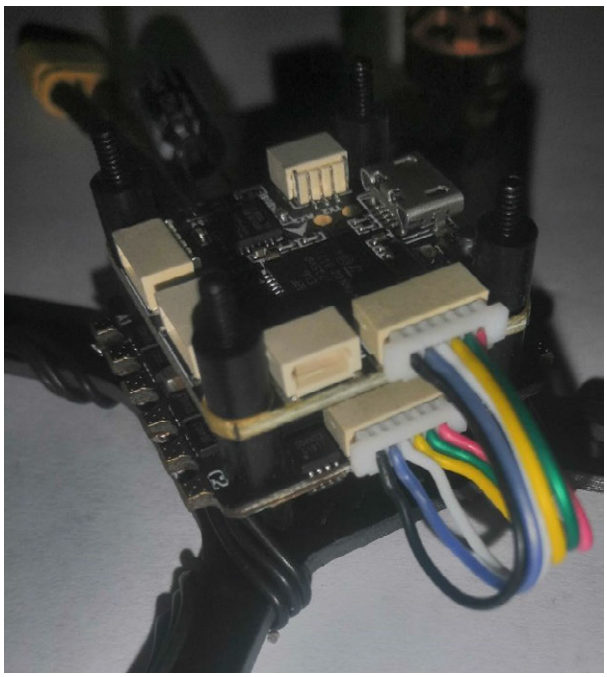

Figure 29: Connection of ESC and FC

4. Install camera and antenna

First install the antenna and camera into the mold.

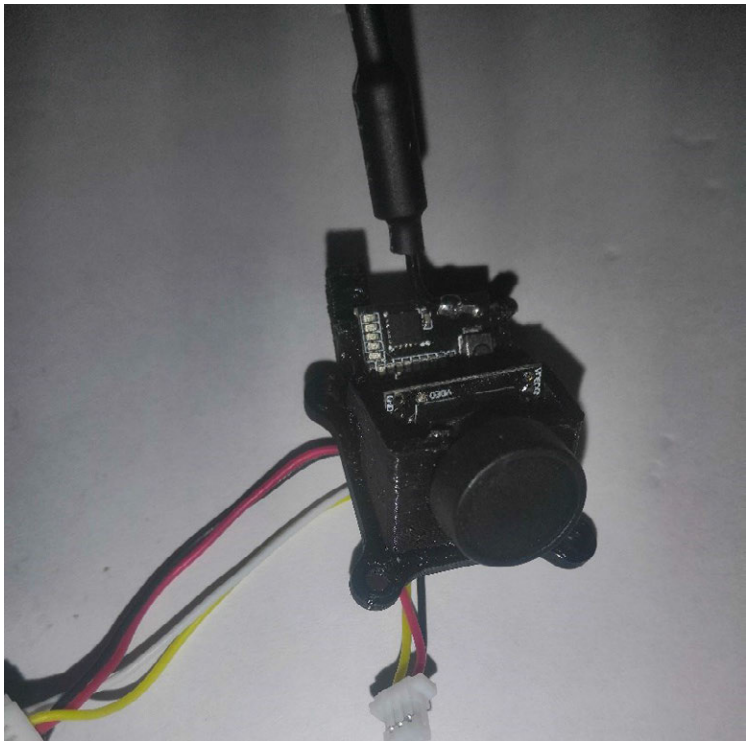

Figure 30: Installation of antennas and cameras

Then install the mold to the uppermost layer and connect the data cable to the FC board (the camera is connected to the CAM interface, and the antenna is connected to the VTX interface).

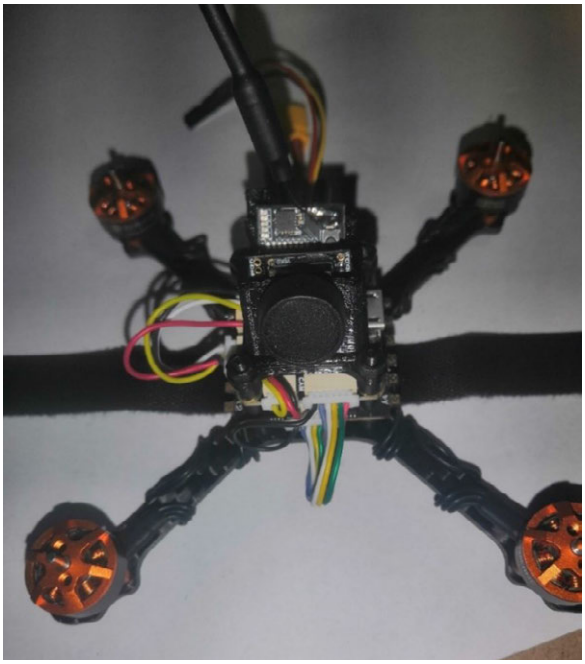

Figure 31: Installing a third level of rubber and camera parts

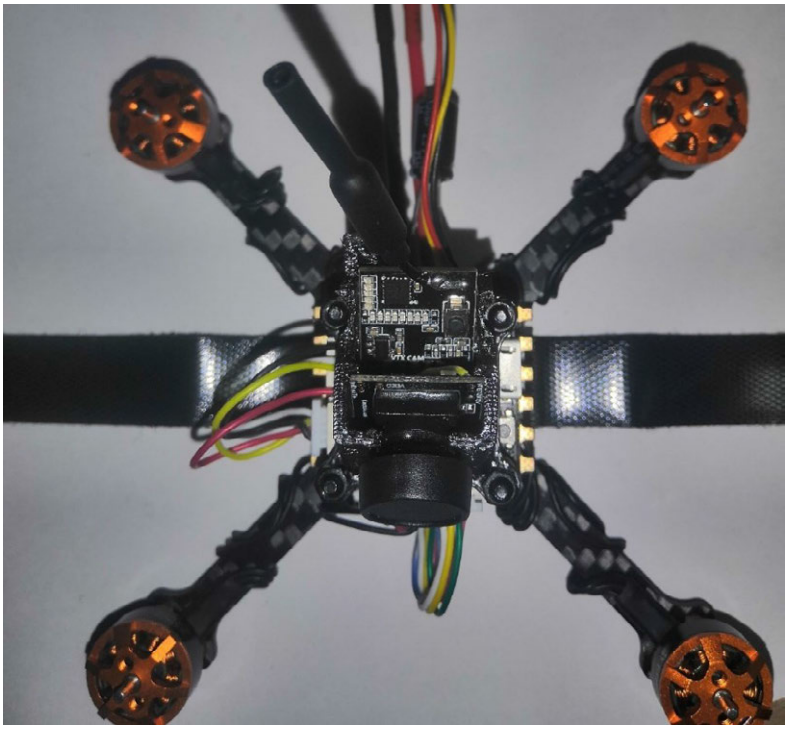

Figure 32: Mounting nut to secure and connect data cable

5. Install the paddle

   Install the propeller on the motor, mainly distinguishing the left and right of the propeller

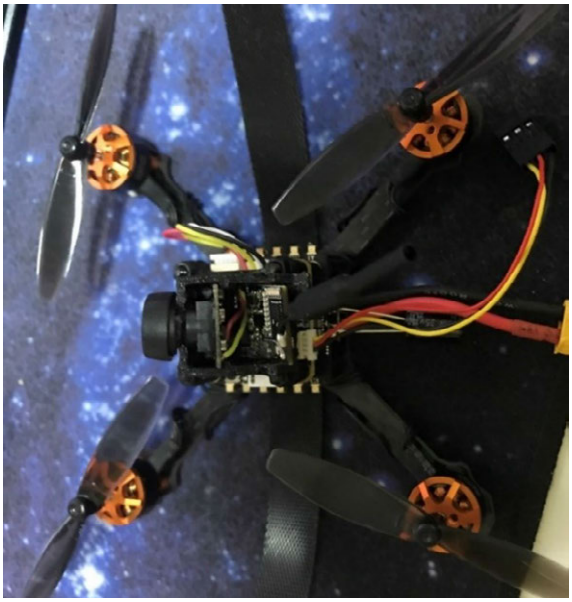

Figure 33: Mounting propellers

Note: Because the receiver model purchased is the FlySky FS-A8S, weighs under 2g and uses the PPM protocol. According to the drone manual, we need to connect the PPM`s pad in the picture below and the pad in the middle.

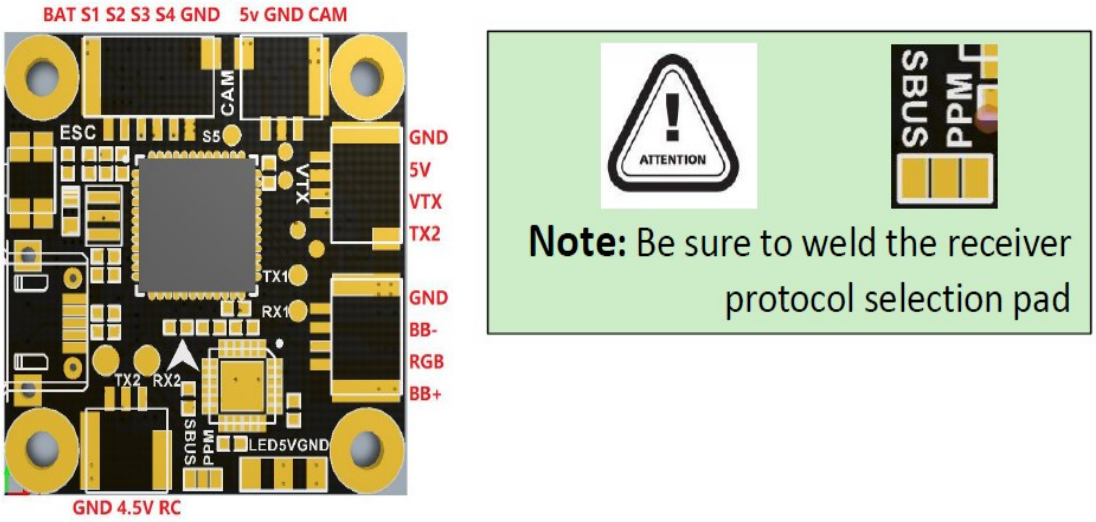

Figure 34: Select Receive Protocol [47]

3.4 Remaining parts and tools

1. FPV 5.8G Receiver Monitor

   A good cheap FPV monitor. It controls the drone flight by receiving the video wireless signal (VTX) sent by the drone camera and provides the critical monitoring function

Figure 35: Receiver Monitor [52]

2. 2.4GHz Controller/Transmitter and Remote Controller (RC) Receiver

   UAV control via 2.4GHz PPM/iBus protocol (FS-A8S)

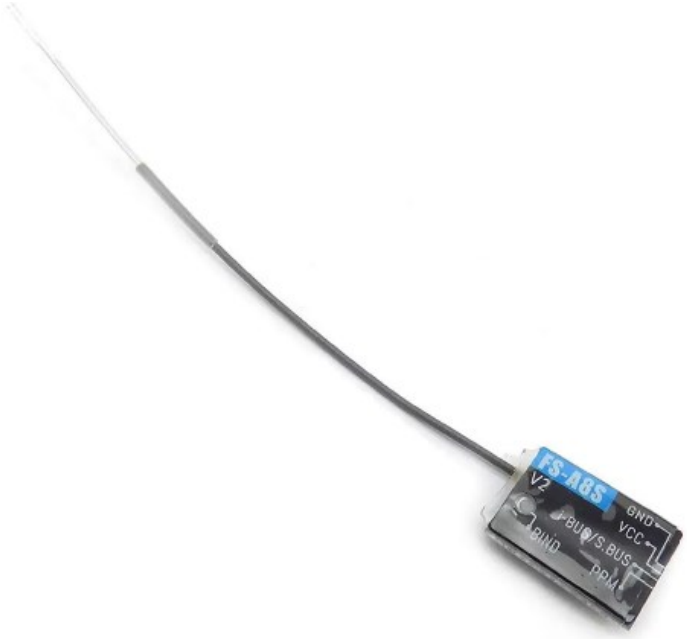

Figure 36: Signal receiver [53]

Figure 37: Remote Controller [53]

The purchased controller and the receiver are bound. If there is no binding, it can follow the steps in MANUAL [48] to bind. In the meantime, according to the experimental required, we need 3 channels in addition to the controller's default of 4, so we need to turn on the controller's additional 3 new channels.

Open additional channel steps:

First, in the controller interface, press and hold the OK button to enter the setup menu interface.

Figure 38: RC's main screen

Then, use the UP/DOWN button to select Functions Setup and press the OK button to enter. Select the Aux.channels option to enter.

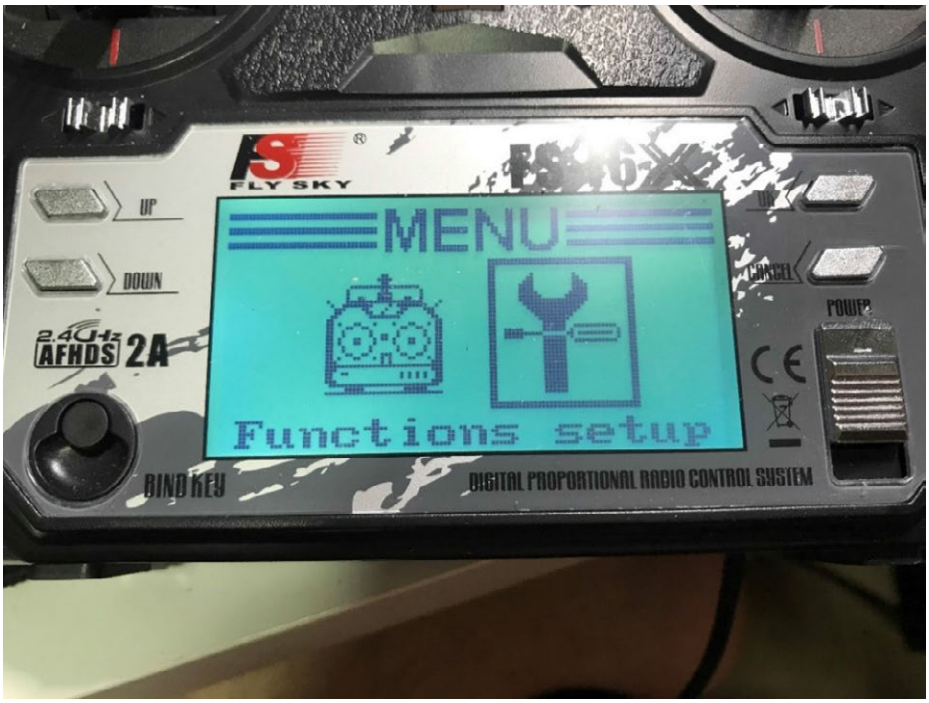

Figure 39: RC`s Setup Interface

Figure 40: RC`s Functions setup Interface

Next, associate channel 567 as switch ABC respectively, press and hold the CANCEL button to save and exit.

Figure 41: RC`s Auxiliary Channels setup interface

Finally, check that all channels are set up successfully in the Display option.

Figure 42: RC`s Channels display interface

3. Batteries & Chargers, etc.

   Uses a 3s (3-cell) 300 mAh battery as recommended by drone manual

4. Software assembly

4.1 Introduction of BetaFlight

With the hardware installation and remote control equipment configuration complete, the software portion of the drone, such as flight control, needs to be set up.

As mentioned earlier when introducing the various drone flight controls, there are 3 major conditions required for the proper operation of the drone. The first is that the drone itself is installed properly, the second is that the remote control system is operating properly, and the last is that the ground station is being properly setup for flight. Although a open source drone is used, most drones on the market today can be used with BetaFlight software Flight control setup as a ground station.

"Betaflight is software that is open source and is available free of charge without warranty to all users.

Betaflight is a Cleanflight fork started by BorisB. It used to work as a beta test platform for Cleanflight and kept pushing the envelope in terms of performance, but it eventually evolved into a fully grown up,

stable and well maintained firmware by various developers. In fact, it looks like it is the number 1 open source multi-rotor firmware according to Google Trends analysis around January 2017.

The name "Beta" comes from the fact it started with a goal to keep trying new things out and have quick and easy test distributions available. The name "Beta" has stuck as the brand name was strong at the moment where it transitioned from beta to stable and there was already a lot of documentation around." [49]

4.2 Getting Started

We can download the latest version of BetaFlight on Github, and connect the drone via USB cable. Usually in the upper right corner of the software, the connection port between the computer and the drone (COM3 as shown) will automatically appear, but if there is no recognition, the user can manually set the correct port. Click on the right Connect button to enter the BetaFlight setup interface. Some flight controls may need not only USB cable but also battery to set up BetaFlight. Also, regardless of whether or not it is the first time the setup is prepared, it is explicitly advised that whenever using the BetaFlight ground station any propeller must be removed due to safety reasons.

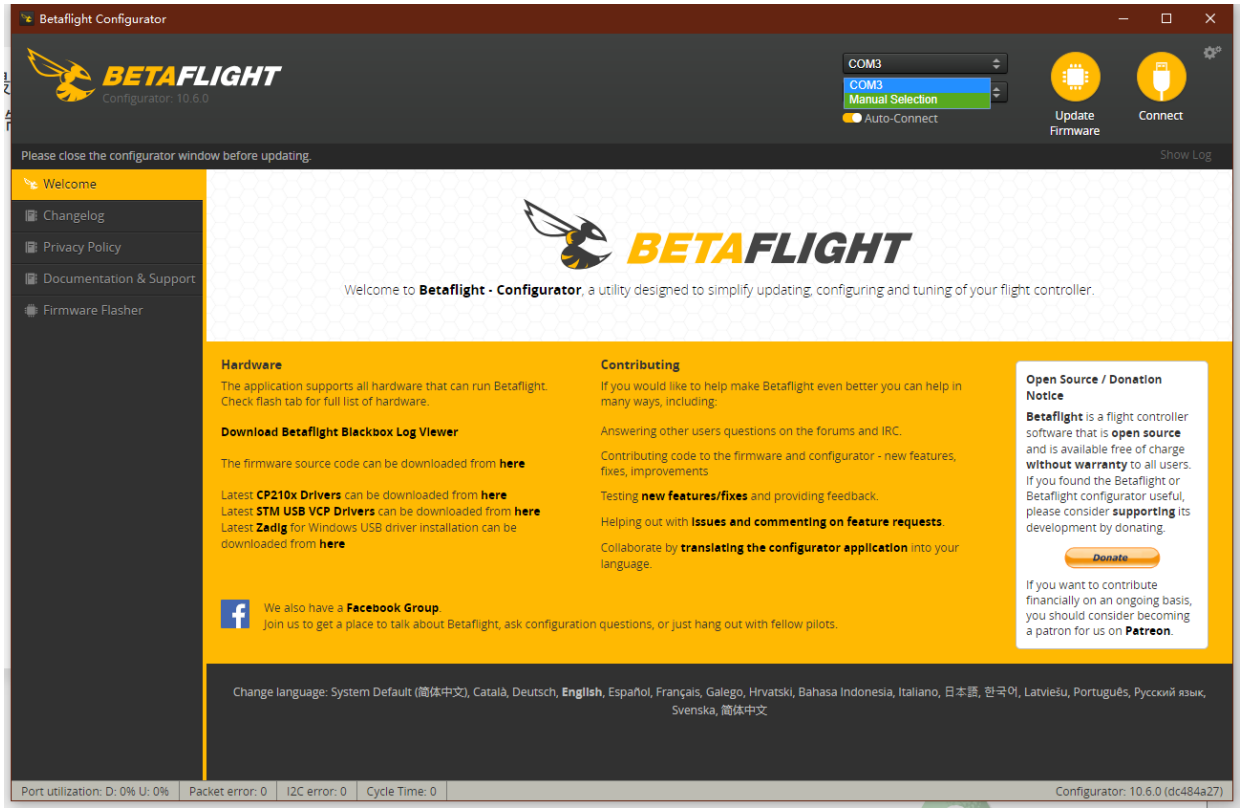

Figure 43: BetaFlight main interface

4.3 Configurations

After entering the BetaFlight settings interface, there will be different menu options as shown below. Again, this part of the Bachelor Thesis has been built as a manual so the configuration process is crystal clear.

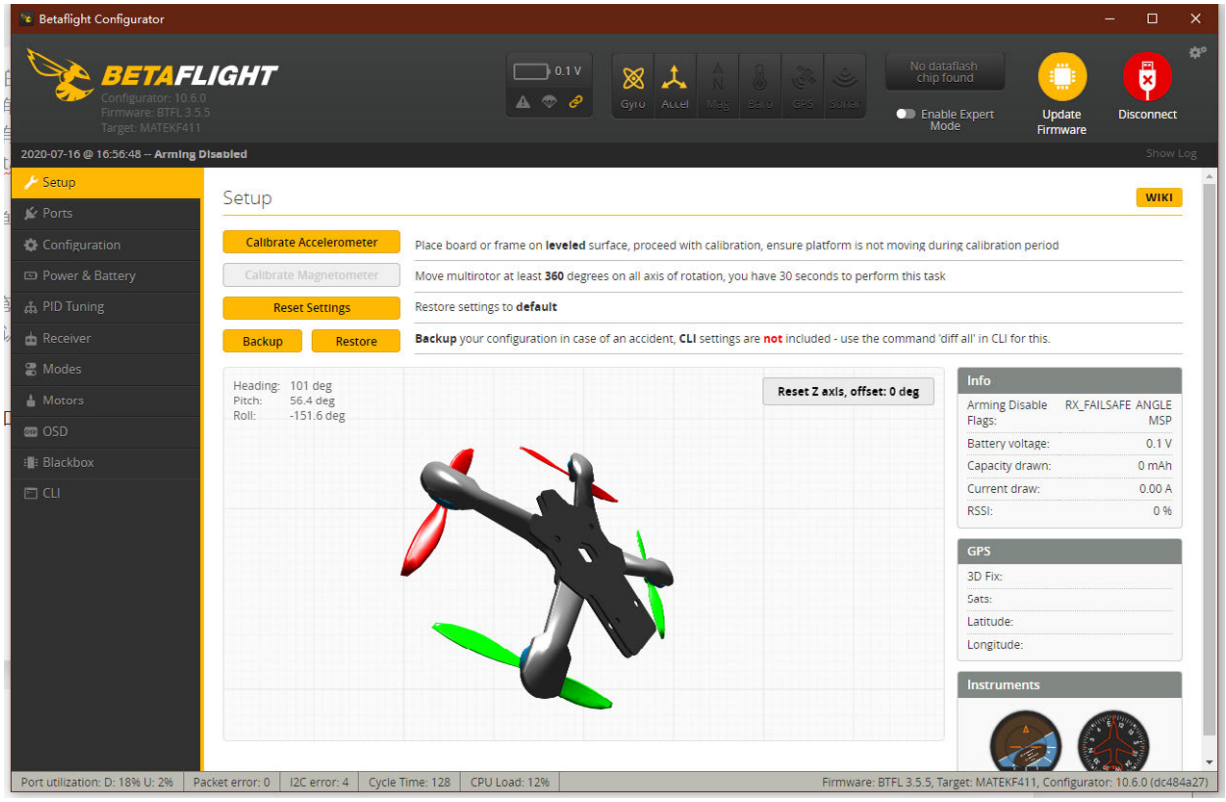

Figure 44: BetaFlight setup interface 1

1. Setup

   In this menu, the software allows you to observe the drone (angle, orientation, etc.). At the same time, we can adjust and calibrate the flight attitude of the drone.

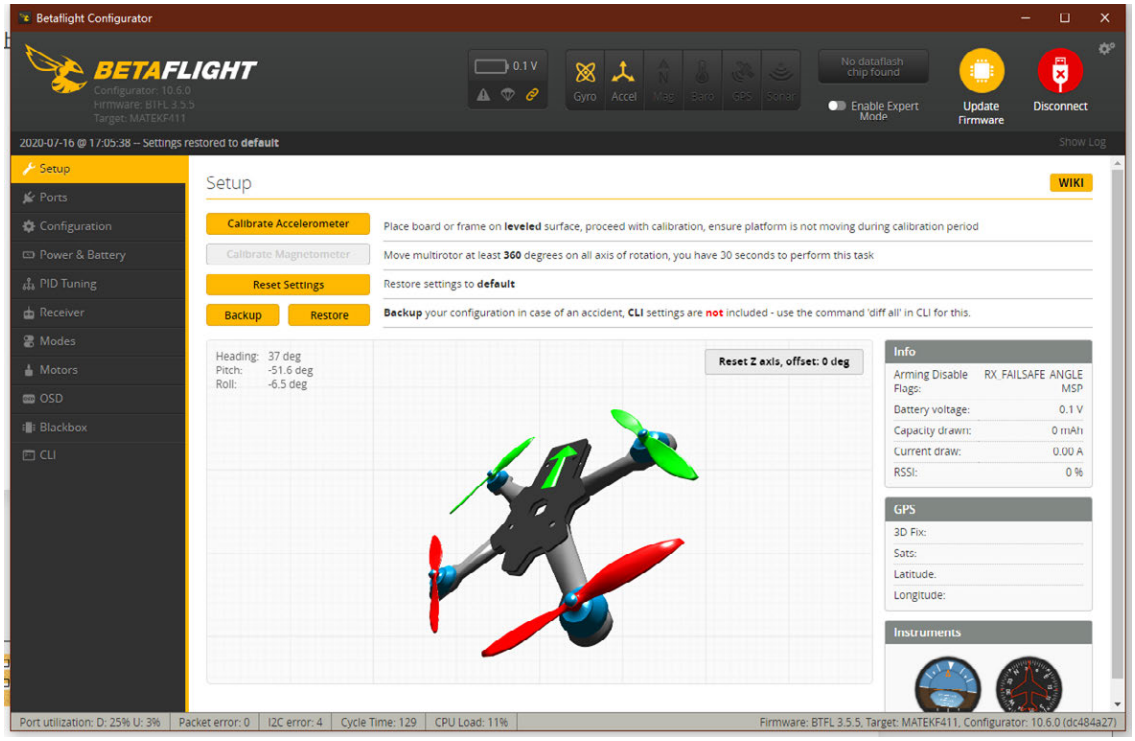

Figure 45: BetaFlight setup interface 2

First of all, place the drone on a flat surface, and observe that the drone model is not in a horizontal attitude in the software (Pitch and Roll values are not 0 deg), click the Calibrate Accelerometer button to calibrate the drone's horizontal attitude.

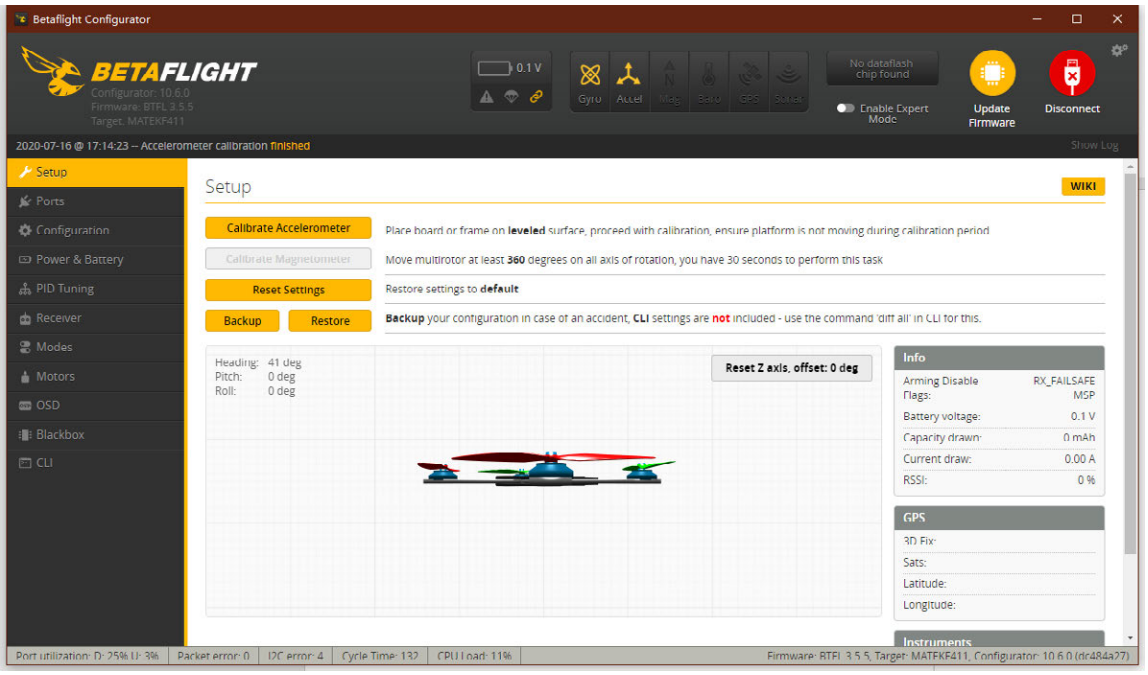

Figure 46: BetaFlight setup interface and horizontal calibration

Then, observe the model in the software, which has a green arrow on it. The green arrow represents the front of the drone (normally if the drone is facing the computer screen, the

green arrow should be facing inward). If the direction of the arrow is wrong, click the Reset Z axis button to correct the direction.

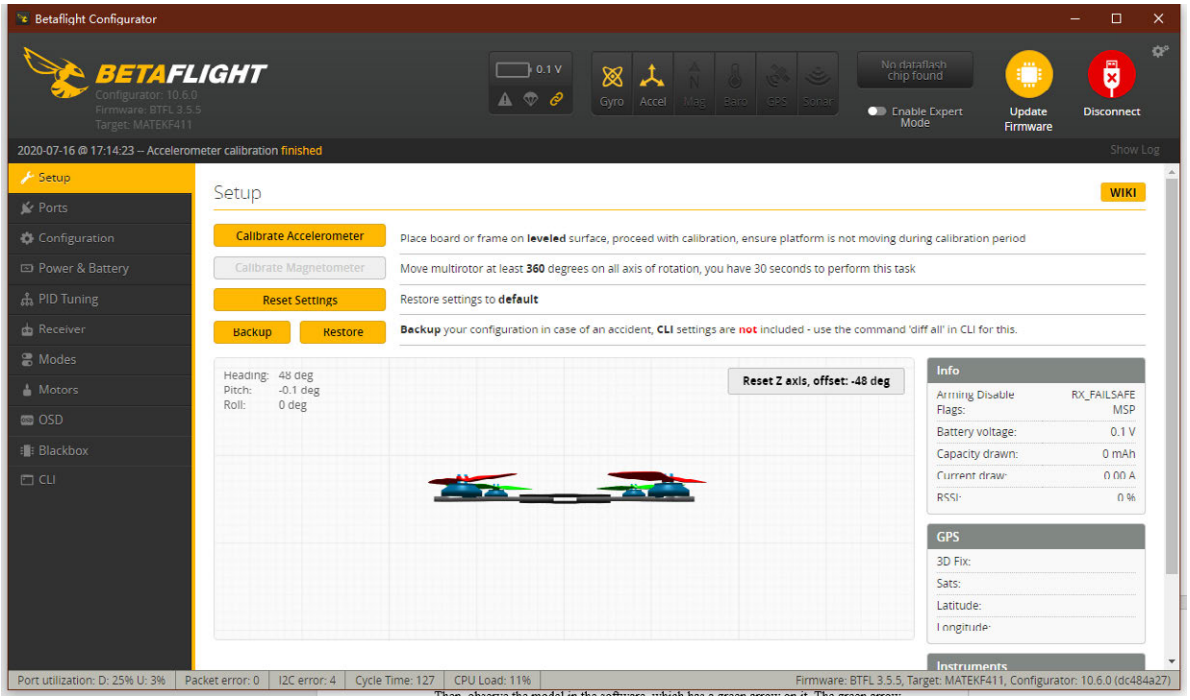

Figure 47: BetaFlight setup interface and z-axis calibration

2. Ports

In the interface Ports, the first thing that happens is that there are different port names depending on the different flight controls.

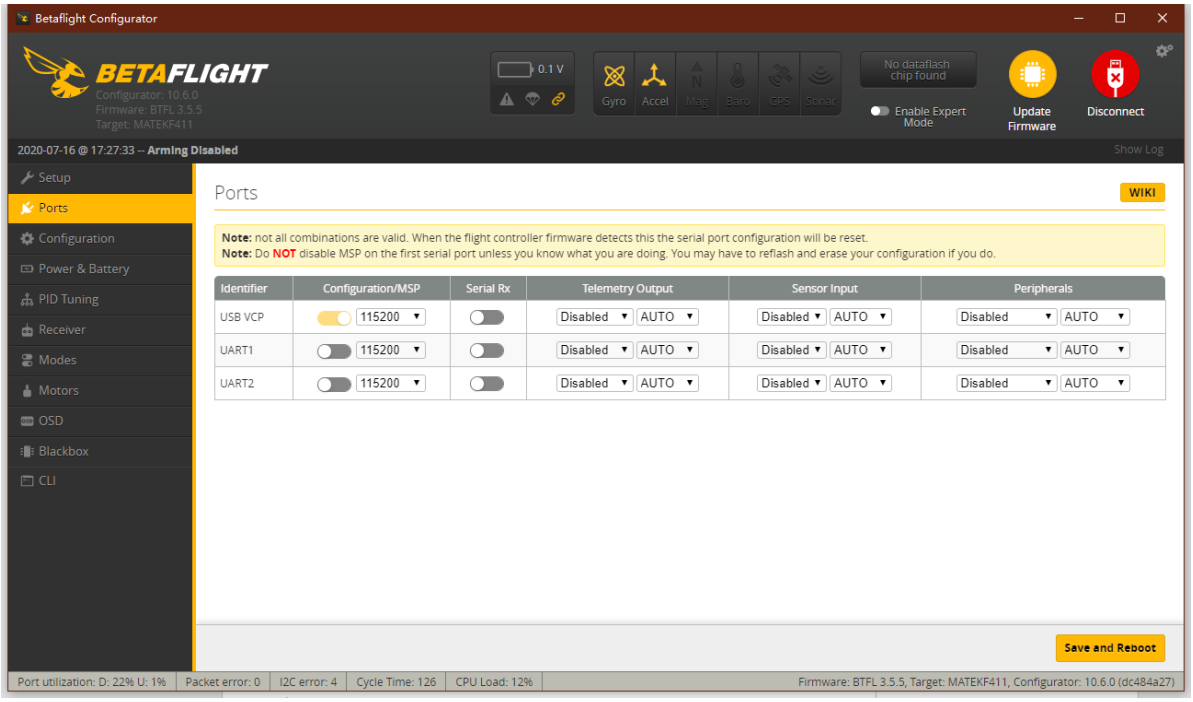

Figure 48: BetaFlight Ports interface

In the first row, there is a USB VCP port, which is the basic default setting for the flight control to connect to the BetaFlight ground station via USB. It is important not to make any settings on this port, otherwise the flight control will not be able to connect to the ground station.

The following UARTX port is the transmitting port of the UAV devices, such as video, GPS, etc. signals, SBUS protocol control signals, etc. These ports are set according to the actual needs, in this experiment only a video signal is needed and does not need to be transmitted to the controller. Also, since PPW is used to transmit control signals, it can be left as default.

Finally, click on the Save and Reboot button in the lower right corner to save.

3. Configuration

Here is the main flight control settings menu, which has many important settings for the drone's flight.

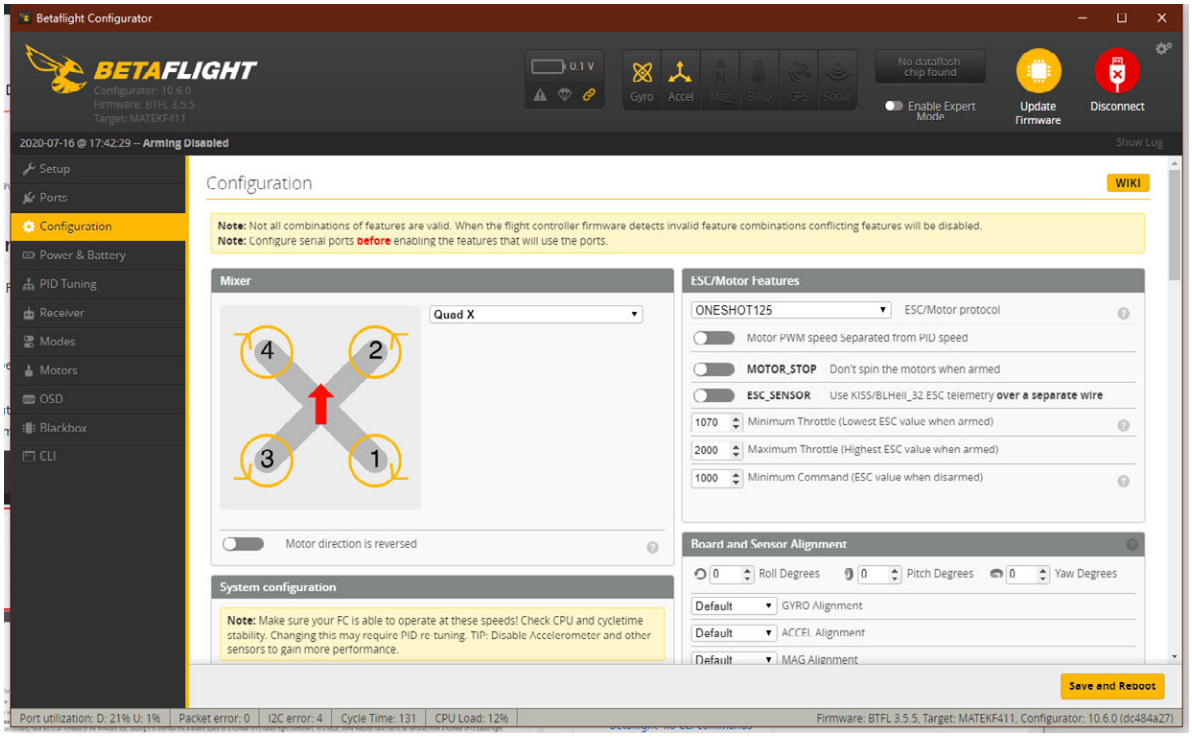

Figure 49: BetaFlight Configuration interface

a) Mixer

In this context, Mixer actually means the type of airframe the drone has, like a 4-axis rotorcraft, 6-axis rotorcraft, or It's a flying wing aircraft, etc. Usually the vast majority of drones in general are of the Quad X type.

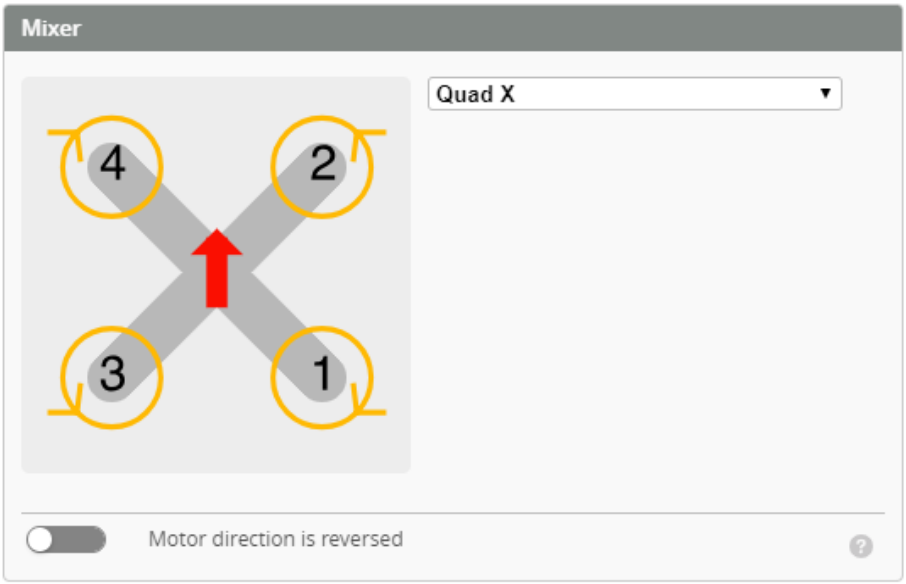

Figure 50: Mixer Configuration

In the mixer interface, you can see the rotation direction of the 4 motors of the drone, the normal flight of the drone is closely related to the normal rotation direction of the 4 motors. If the drone motor rotation direction is wrong, you can re-weld the motor or modify it through the software, this part will be explained in detail in Motors section.

b) ESC/Motor Features

According to the UAV MANUAL, the ESC used by the UAV is based on the ONESHOT125 protocol, so the correct ESC protocol needs to be selected here. Other settings do not need to be changed; default is sufficient.

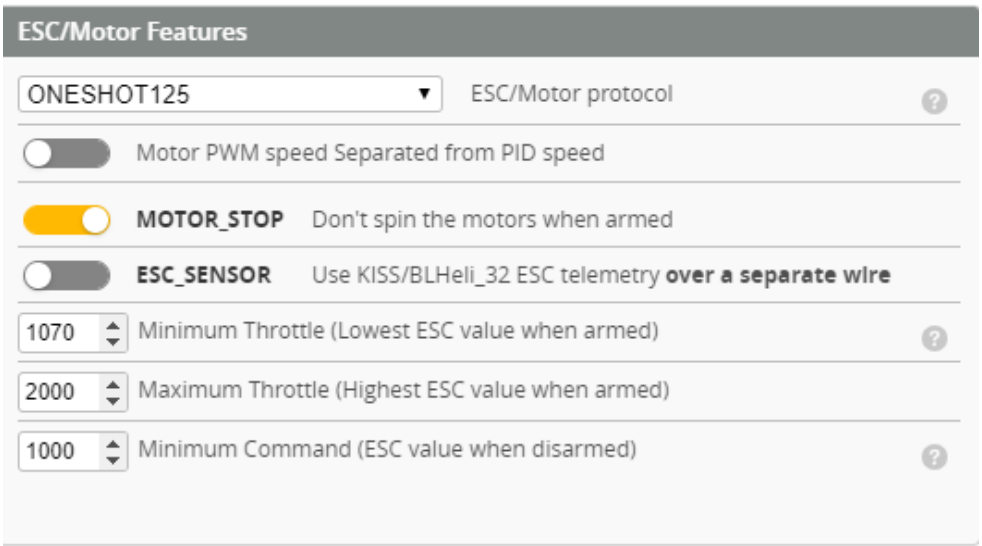

Figure 51: ESC/Motor Configuration

In addition, it is recommended to disable MOTOR_STOP for safety reasons and to turn off the motor rotation when unlocking. If it is off, the motor will rotate at a low speed after the drone is unlocked. This will alert the surrounding area that the drone will be taking off, please stay away for safety.

c) System configuration and Personalization

This is a more complex part, but because the experiment requires a simpler drone, it is only needed to set the Gyro frequency to 8kHz and PID frequency to 2kHz. Because the experiment requires the drone to monitor and observe, turning on the Accelerometer option makes it easier for the drone to self-stabilize. A name can be defined for the drone in Personalization.

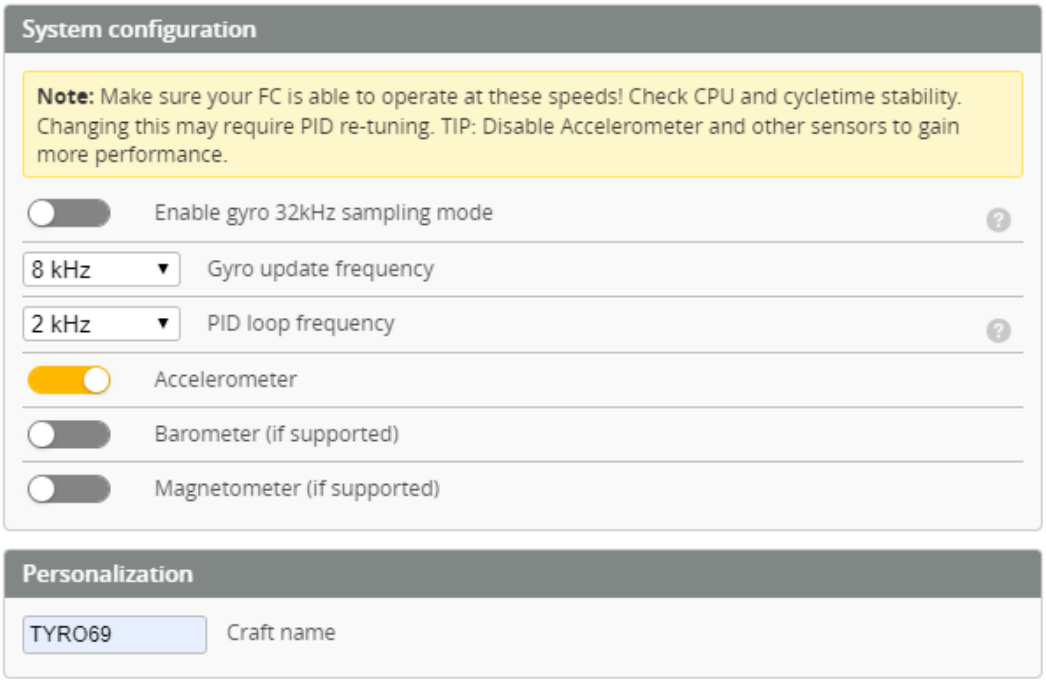

Figure 52: System Configuration

d) Receiver

In the receiver settings, the receiver mode is selected according to the receiver protocol used, e.g. PWM, iBus, sBus. etc. This experiment uses a PWM protocol, so the PWM RX input mode is selected.

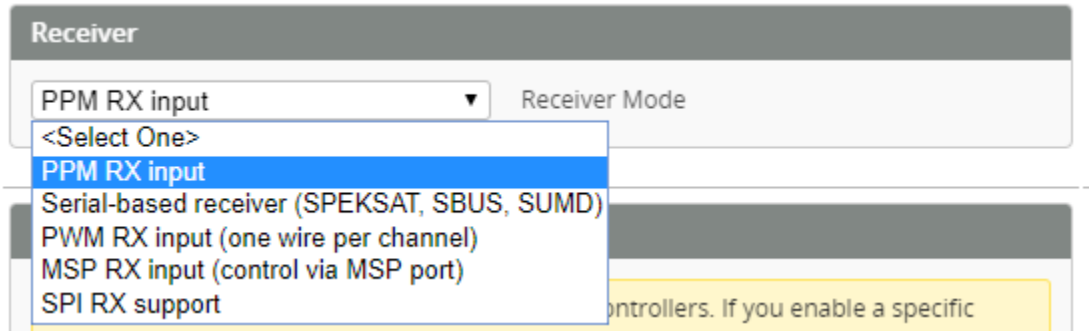

Figure 53: Receiver channel configuration

e) Other settings

In the remaining settings, it can be turned on and off depending on the specific experiment. Default settings are used in this experiment.

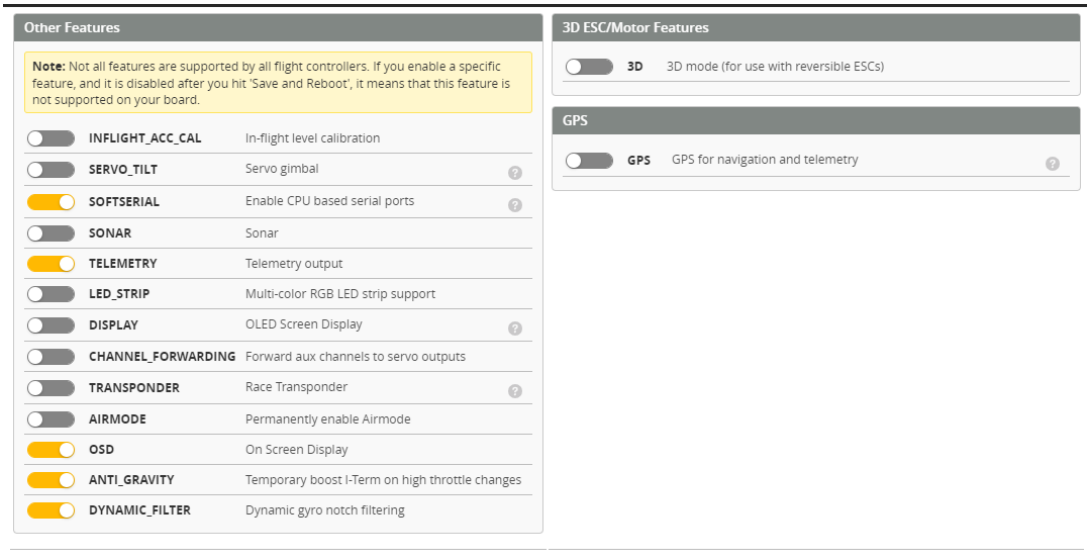

Figure 54: Others Setting Menu 1

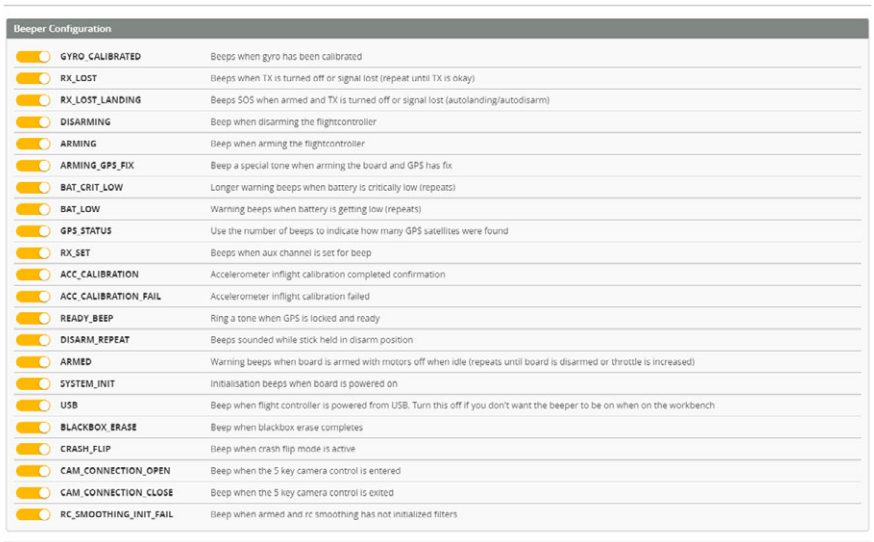

Figure 55: Others Setting Menu 2

Once all the settings are complete, click the SAVE and Reboot button to proceed to the next setting.

4. Receiver interface

The receiver interface is also an important setup interface. In the previous hardware assembly, the receiver and controller were already bonded, and three additional channels were added to the controller. Thus, eight channels can be seen in the receiver interface.

Default 4: Roll, Pitch, Yaw and Throttle.

Additional 3: AUX1, AUX2, AUX3.

Extra 1: AUX4.

Extra 1: AUX4.

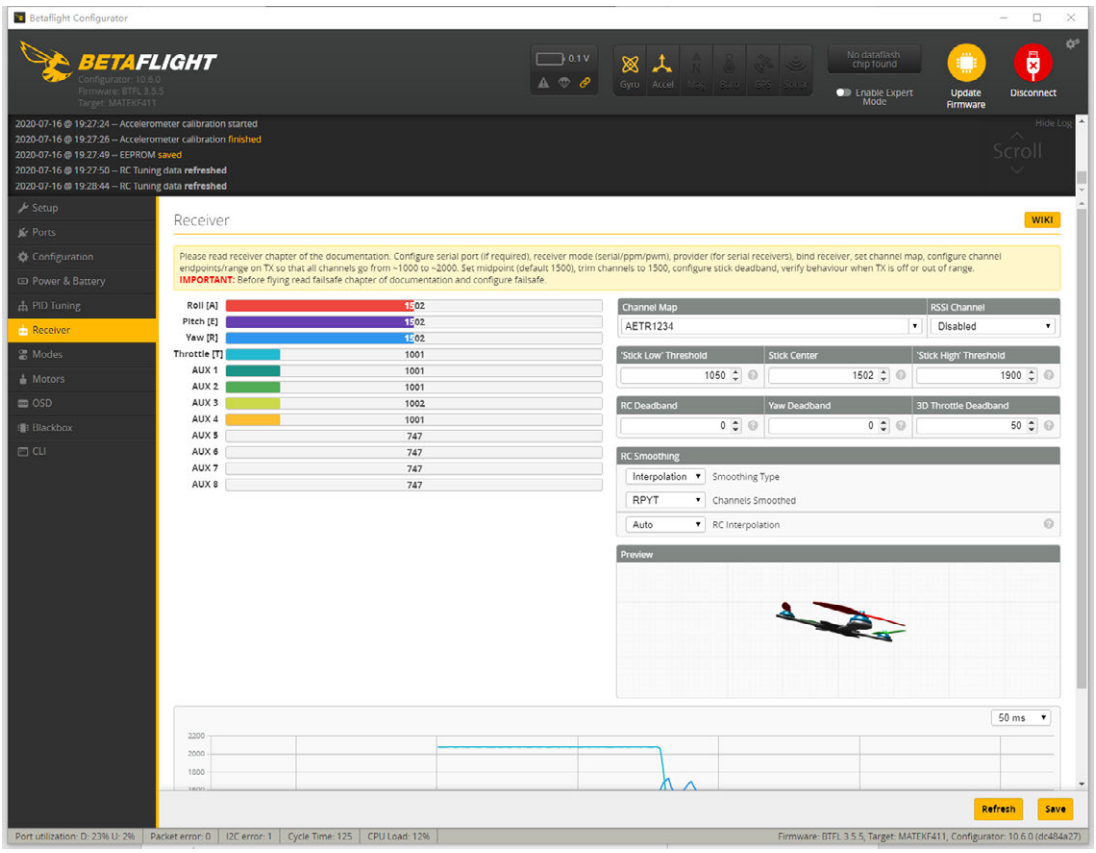

Figure 56: BetaFlight Receiver interface

First, observe whether the channel mapping is consistent with the controller. For example, move the controller's throttle rocker to maximum and observe if the value bar for throttle in the software interface is consistent. As shown in the two images below, the throttle in the controller, the additional SWA (channel 5) and SWC (channel 7) have been It was moved to the maximum, and the value bars for throttle, AUX1 and AUX3 on the software also changed to the maximum.

Figure 57: Adjustment Controller Rockers and Switches

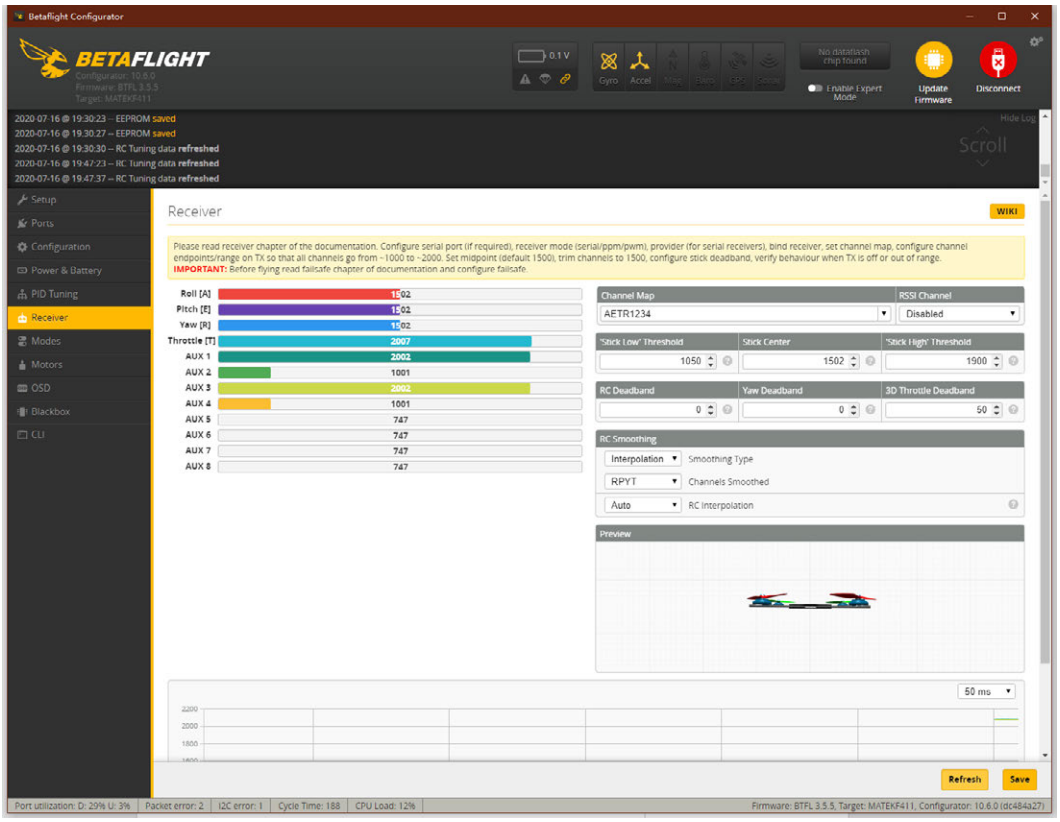

Figure 58: Receiver value bar is changed

If the value changed by the controller does not match the corresponding value bar in the software, it may be that the channel mapping option is incorrect and can be changed in the channel mapping option.

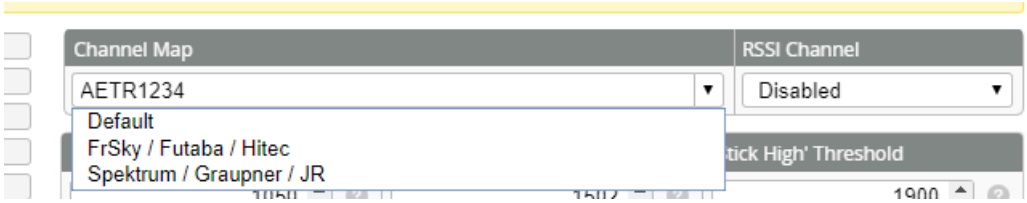

Figure 59: Channel Map setting

In addition, to facilitate drone operation, it is often necessary to set the Stick to the lowest and highest position.

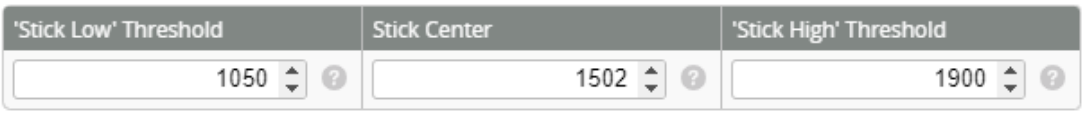

Figure 60: Critical Value Setting

In general, the value of the Stick ranges from 1000 to 2000, but due to some information interference or production process problems, the software is not able to set the value of the Stick. The value bars in it are all about 5% inaccurate, which doesn't affect the flight control of the drone. Note, however, that in Stick Low Threshold it is slightly above the minimum value of the Stick. This is a convenient way for the drone to determine if the controller is zeroing the throttle valve and thus unlocking it.

5. Modes.

In the mode menu, there are a number of already set drone modes that can be selected via the controller channel, as well as the number and type of modes that the user has depending on the drone model.

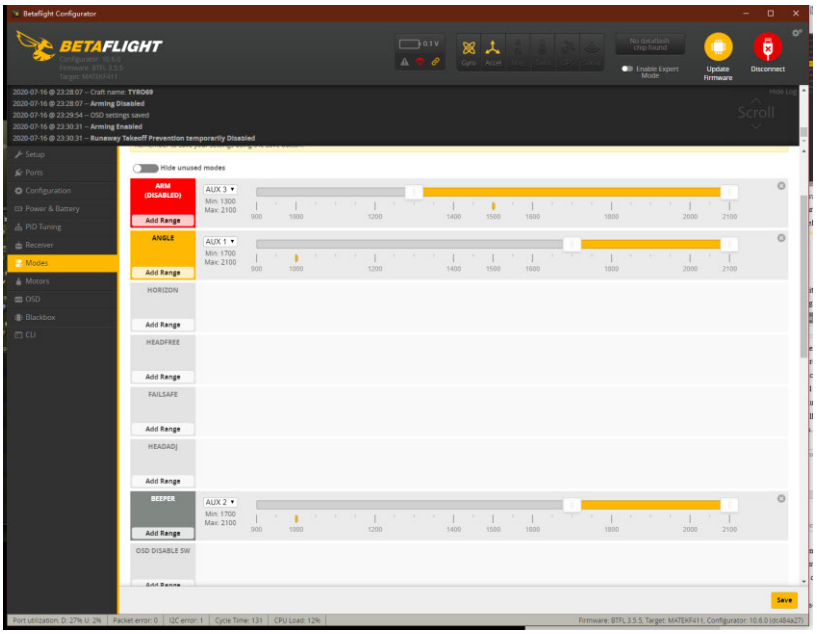

Figure 61: BetaFlight Modes interface

According to GitHub the user can find a detailed description of the pattern and its features. [50]

"There are various modes that can be toggled on or off. Modes can be enabled/disabled by stick positions, auxiliary receiver channels and other events such as failsafe detection.

ARM: Enables motors and flight stabilization.

ANGLE: Legacy auto-level flight mode

HORIZON: Auto-level flight mode

ANTI GRAVITY: Prevents dips and rolls on fast throttle changes

MAG: Heading lock

HEADFREE: Head Free - When enabled yaw has no effect on pitch/roll inputs

HEADADJ: Heading Adjust - Sets a new yaw origin for HEADFREE mode

CAMSTAB: Camera Stabilization

PASSTHRU: Pass roll, yaw, and pitch directly from RX to servos in airplane mix

BEEPERON: Enable beeping - useful for locating a crashed aircraft

LEDLOW: Switch off LED_STRIP output

CALIB: Start in-flight calibration

OSD: Enable/Disable On-Screen-Display (OSD)

TELEMETRY: Enable telemetry via switch

SERVO X: Servo X

BLACKBOX: Enable BlackBox logging

FAILSAFE: Enter failsafe stage 2 manually

AIRMODE: Alternative mixer and additional PID logic for more stable copter

3D: Enable 3D mode

FPV ANGLE MIX: Apply yaw rotation relative to a FPV camera mounted at a preset angle

BLACKBOX ERASE: Erase the contents of the onboard flash log chip (takes > 30 s)

CAMERA CONTROL X: Control function X of the onboard camera (if supported)

FLIP OVER AFTER CRASH: Reverse the motors to flip over an upside down craft after a crash (DShot required)

BOXPREARM: When arming, wait for this switch to be activated before actually arming

BEEP GPS SATELLITE COUNT: Use a number of beeps to indicate the number of GPS satellites found

VTX PIT MODE: Switch the VTX into pit mode (low output power, if supported)

USERX: User defined switch X. Intended to be used to control an arbitrary output with PINIO

PID AUDIO: Enable output of PID controller state as audio

PARALYZE: Permanently disable a crashed craft until it is power cycled

GPS RESCUE: Enable 'GPS Rescue' to return the craft to the location where it was last armed

ACRO TRAINER: Enable 'acro trainer' angle limiting in acro mode

DISABLE VTX CONTROL: Disable the control of VTX settings through the OSD

LAUNCH CONTROL: Race start assistance system" [50]

Depending on the experimental situation, only 4 modes are needed: ARM, ANGLE, BEEPER and AIR MORE. Channel 5 and 6 are two-stage switches, i.e., closed value is 1000 and open is 2000. for 1000, 1500 for the middle and 2000 for the top.

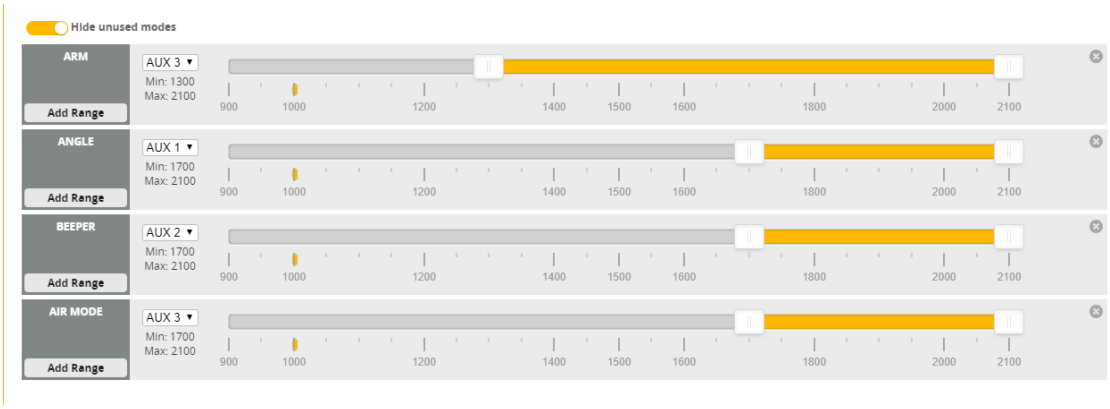

Figure 62: Modes` value setting range

ARM: The actual setting is based on the English ARMED, which is the actual unlock mentioned above, and the drone's motor will only run when it is unlocked.

AIR MODE: "*Air mode will enable full PID correction during zero throttle and give an ability for nice zero throttle gliding but also the cornering / turns will be much tighter now as there is always maximum possible correction performed.*" [50]

The controller's SWC (channel 7) is a 3-stage switch that sets the unlock function and flight mode on a single channel. As shown above, adjust the ARM value range to 1300 to 2100 and AIR MODE to 1700 to 2100. 2100. so that when the switch is in the middle, the drone unlocks, and when the switch is at the top, the drone enters flight mode.

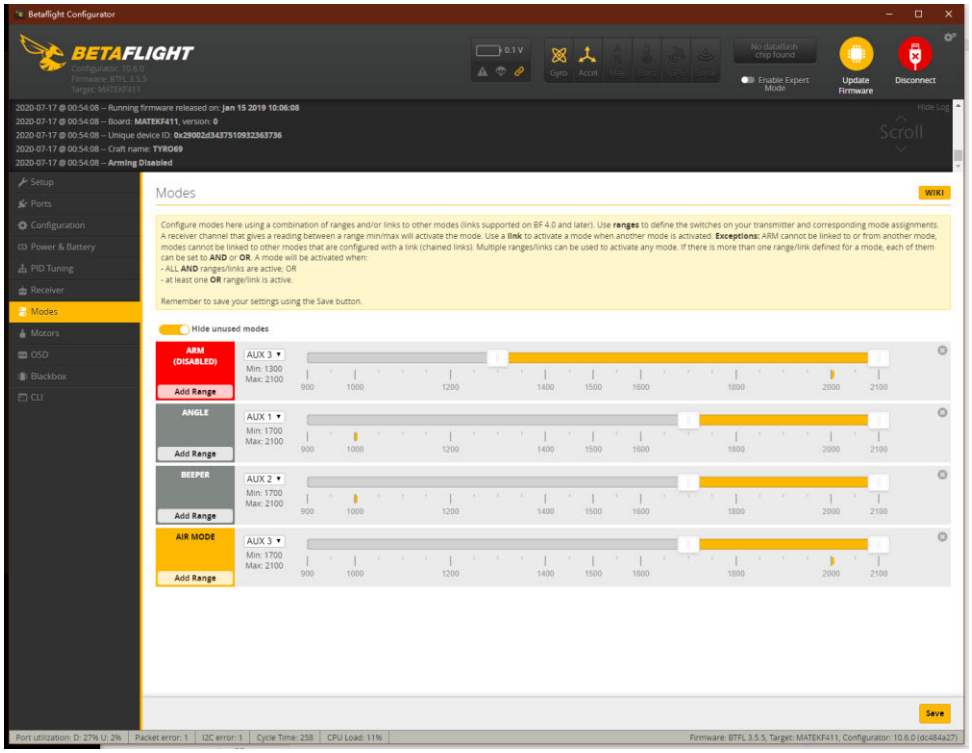

Figure 63: Modes` test interface 1

ANGLE: Turn on the drone self-stabilization mode, more convenient to observe and monitor the associated SWA (channel 5)

BEEPER: Facilitates finding the drone after a crash, associated SWB (Channel 6).

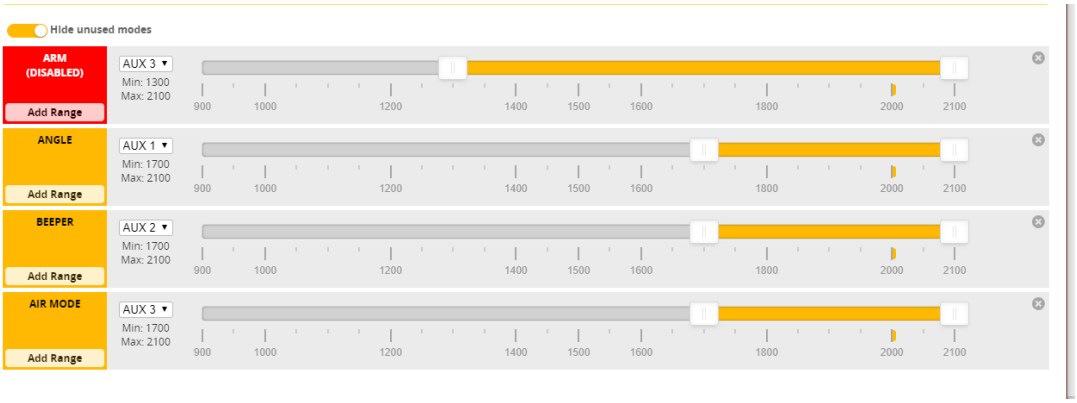

Figure 64: Modes` test interface 2

Finally, verify that the mode is on and save.

6. Motors

In the motors menu, the main feature is that the user can test each motor's operation and direction of rotation separately. Once again, the propeller must be removed for safety reasons.

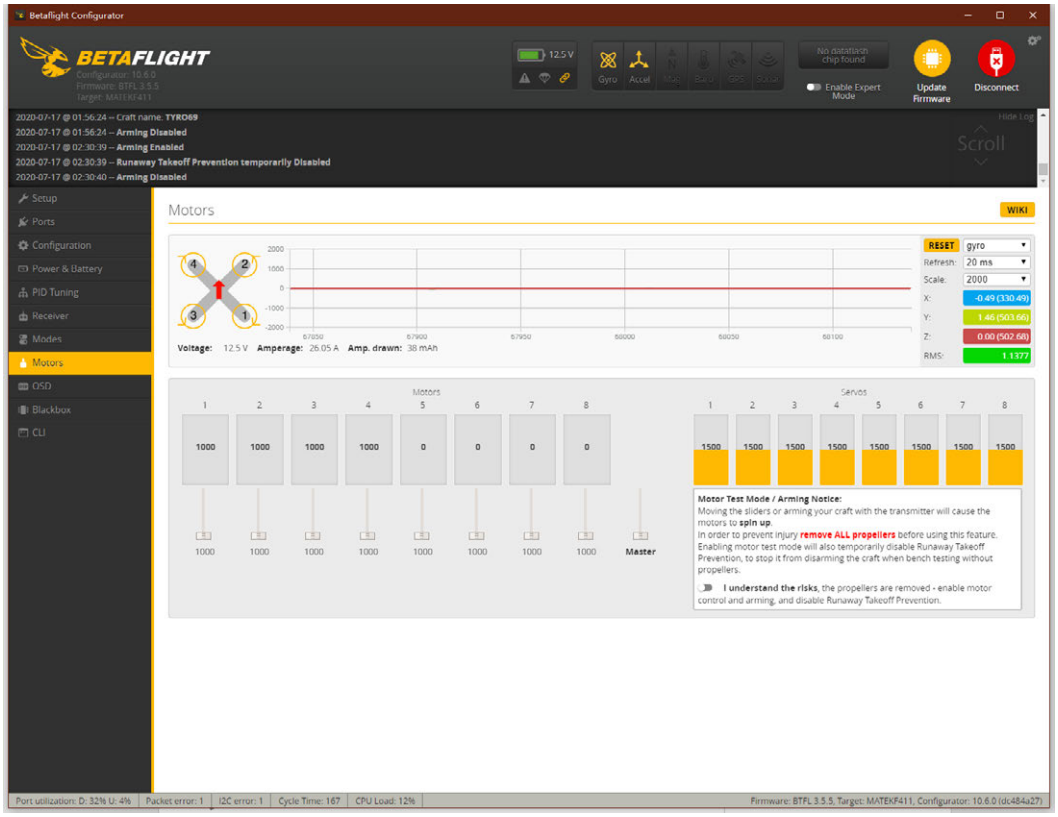

Figure 65: BetaFlight Motores interface

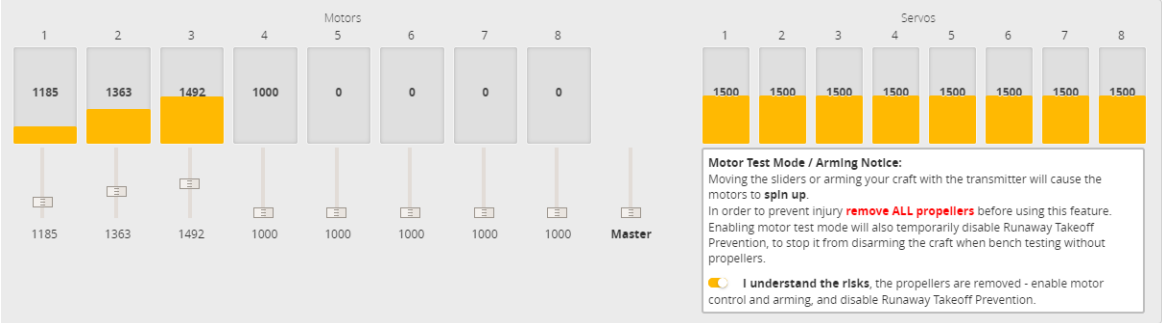

Figure 66: Test motor`s operation and direction

The test motor's operation and direction of rotation can be adjusted manually by turning on the lower right switch and connecting the battery.

If the motor is steering incorrectly, the BLHeli software can be used to correct it.

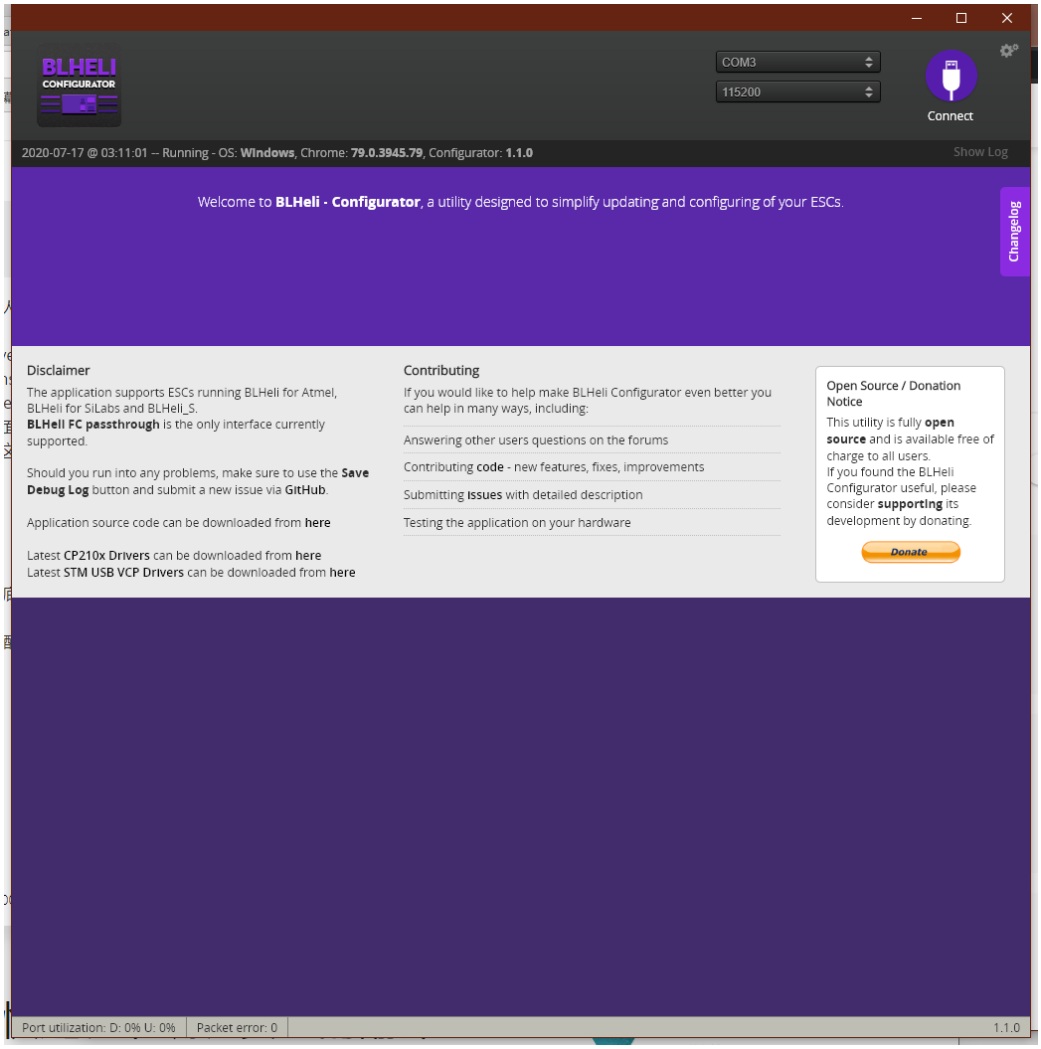

Figure 67: BLHeli main interface

First, disconnect the BetaFlight while powering up the drone and click the Connect button to connect.

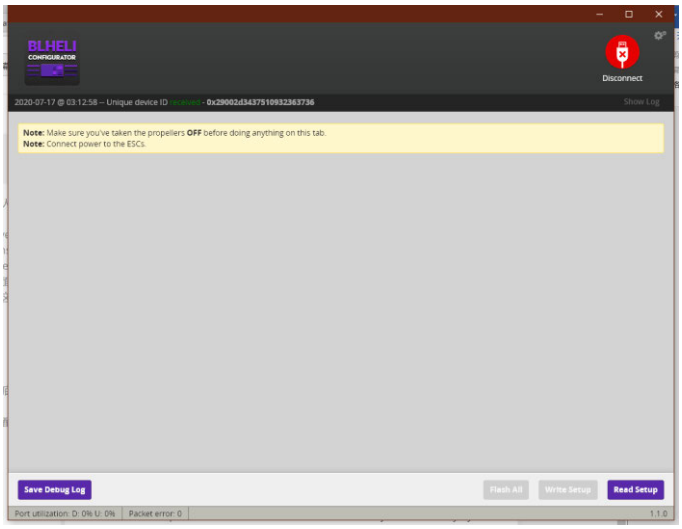

Figure 68: BLHeli connecting interface

Then select the bottom right button to read the current motor data.

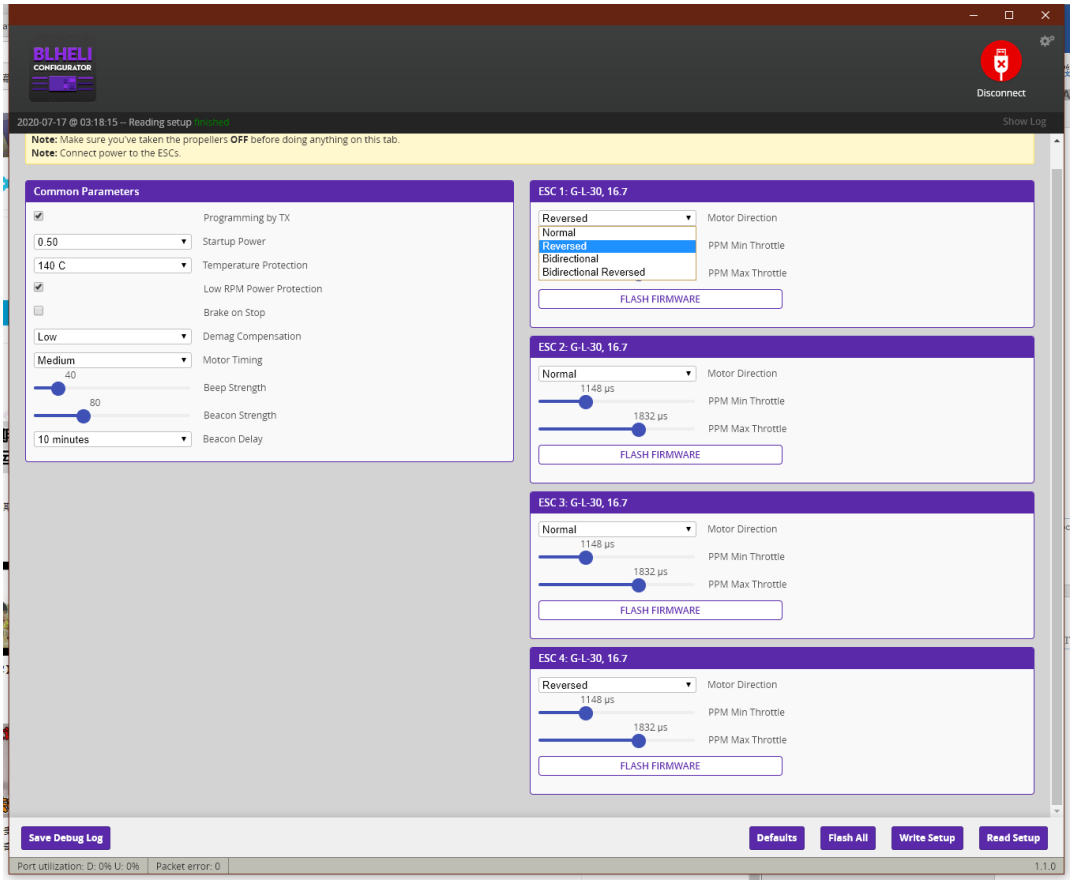

Figure 69: BLHeli`s motor setting interface

In Motor Direction you can adjust the direction of motor rotation, as shown in the figure of the drone used in the experiment. The steering was reversed for motors #1 and #4. Finally click on Write SetUp to write a new motor configuration to the ESC.

7. OSD

The OSD settings allow you to select elements according to the situation, such as battery voltage, flight mode, calculator, warning boost and other elements.

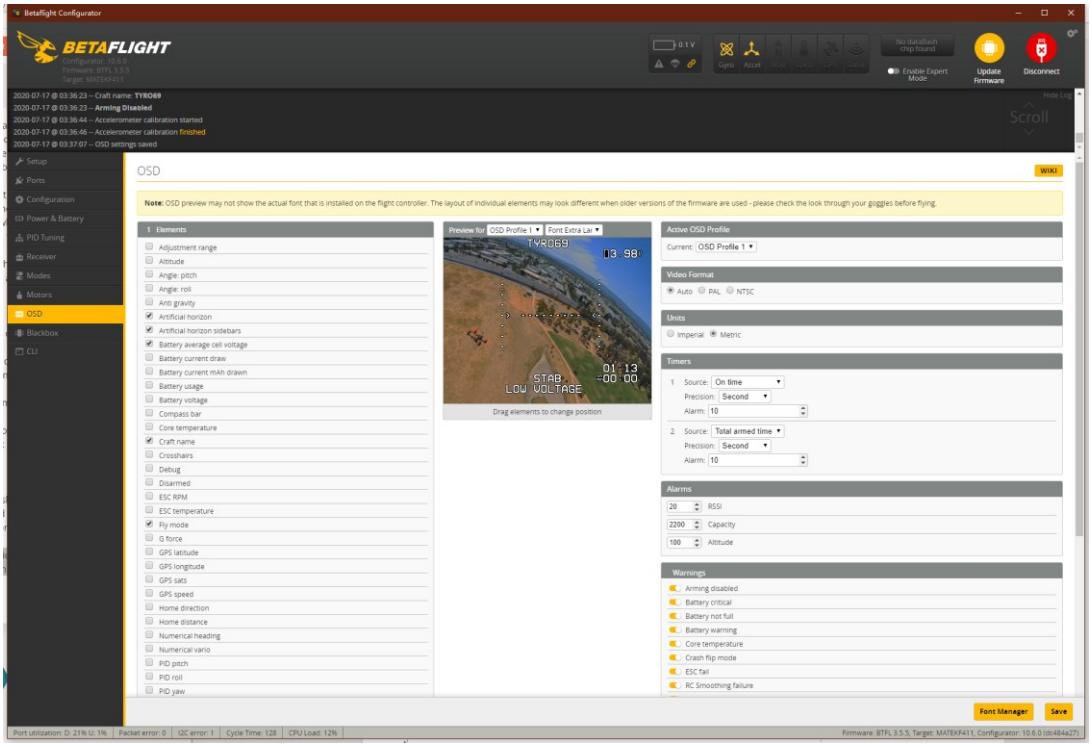

Figure 70: BetaFlight OSD interface

Finally, click save and check the OSD settings at the screen receiver.

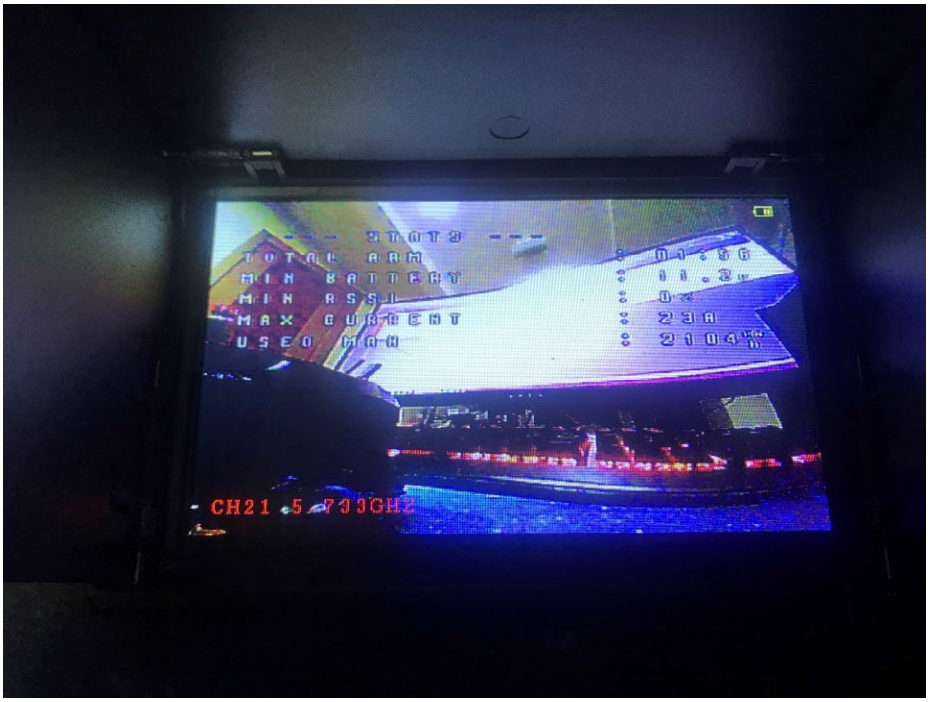

Figure 71: VTX display screenshots 1

Figure 72: VTX display screenshots 2

5. Testing

5.1 Test flight

When the hardware and software configuration of the aircraft is complete, flight tests are conducted in nearby open areas (parks, squares, etc.). Test whether the drone's flight attitude is normal, whether the unlock is successful, whether the automatic stabilization function is activated, etc.

Figure 73: UAV in test flight (Preparing for takeoff.)

Figure 74: UAV in test flight (Taking off)

Figure 75: UAV in test flight (In the air)

After the flight test, the drone's flight attitude is correct, all modes can be started normally, and the camera equipment transmission signal is normal. It can proceed to the next test to actually observe the surveillance mission.

5.2 Testing Mission

With the successful test flight of a drone, a fully designed test mission is needed to simulate the effects of aerial monitoring by a drone, thus verifying the viability of a drone in IoT applications.

1. Test 1

   Test scenario: select a road and use a drone to fly along the road to observe it.

   Test Purpose: To monitor the road with the drone, simulating its task of inspecting power transmission lines or traffic conditions.

   Test process: fly along the road as shown in the figure below, and observe and film at 4 points on the way, ABCD.

Figure 76: Path of flight from Google Maps

Figure 77: The drone's camera video signal is at point A

Figure 78: The drone's camera video signal is at point B

Figure 79: The drone's camera video signal is at point C

Figure 80: The drone's camera video signal is at point D

2. Test 2.

   Test scenario: select a fixed target and use a drone to observe it from flight.

   Test Purpose: To use a drone through two directions to simulate the task of observing buildings or animal protection, etc. with a drone.

   Test process: follow the path shown below, and observe and film on the way.

Figure 81: Paths of flight from Google Maps

Figure 82: The drone's camera video signal is in path A I

Figure 83: The drone's camera video signal is in path A II

Figure 84: The drone's camera video signal is in path A III

Figure 85: The drone's camera video signal is in path A IV

Figure 86: The drone's camera video signal is in path B I

Figure 87: The drone's camera video signal is in path B II

Figure 88: The drone's camera video signal is in path B III

Experimental results: first of all, because the flight operation is difficult and I am not skilled, the flight test can only put the UAV to a simple horizontal straight line flight test. But even a simple straight line flight can also reflect the role of the drone in monitoring work, if the drone is equipped with a simple

autopilot chip using software to control the flight attitude, there will be greater enhancements. Through the test, it was found that drones can improve the efficiency of inspection and reduce safety risks, and due to the flexibility of drones, they can work in harsh environments and places that are difficult for people to reach.

6. Budget list

With the end of the experiment, the entire paper has been completed. Here is a list of the costs of the experiment

Item Name	Description	Item prices
TYRO69	UAV main components	70 Euro
FS-I6X+FS-A8S	Flysky RC Transmitter and Receiver for FPV/UAV	52.38 Euro
5.8G 48CH FPV Monitor	FPV monitor	27.47 Euro
Happymodel 3S Lipo 11.4V 300mAh Li-ion Battery	Drone`s battery	24.99 Euro
welding station and welding materials	Tools and materials used to assemble the drone	22.49 Euro
Labor cost	At €20 per hour, this includes labor costs for drone installation and testing. (Approximately 11 hours for installation and 4 hours for testing.)	300 Euro

Total	497.33 Euro

Table 3: Cost statistics table

7. Conclusion

This Bachelor Thesis introduces in detail the current status of the Internet of Things (IoT) and drones and the development of "IoT drones", their characteristics and some application examples.

At the same time, the experiment of "DIY UAV monitoring in IoT" is conducted to explain the assembly and testing of DIY UAV (or FPV) by introducing the experimental steps in detail.

Finally, the results of the experiment are analyzed to study the feasibility of drones in the Internet of Things; the future development trend of drones in the Internet of Things is also presented.

It has been found through testing that drones have a higher efficiency, wider range and lower risk than humans in some applications of the Internet of Things. In addition, due to the low cost of drones, a large number of drones can be used at the same time for various observation tasks. For example, the TYRO69, with the load capacity provided by four high-powered motors, can carry different battery capacities within the load range, changing the flight time and range. Therefore, using some of the previous examples of UAVs in the IoT as a reference, it is entirely possible to consider a UAV with a modular design that allows it to carry different devices to perform various tasks.

Drones can choose their body structure according to the scope of the work area: helicopters or multi-axis drones can be used for small scale missions, and large fixed-wing drones can be used for large scale operations. Fixed-wing drones used for a wide range of work can use fuel as an energy source, which can provide a large load and long flight time, but fuel combustion will produce crop pollution, and the higher cost of overall maintenance difficulties will come as a drawback too. Compared to fuel drones, electric drones are easier to operate and maintain, have lower costs, and their motor life is longer than fuel motors, but because of the limitations of the battery, the load must be small and the working time is short.

1. In agriculture, drones can replace manual spraying of pesticides and fertilizers or sowing of seeds, in addition to real-time monitoring of farmlands, livestock and so on. This type of drone can be selected from the body type, battery-powered or fuel-powered, depending on the scope of work. They are also equipped with pesticide spraying systems, seed planting systems, high definition

cameras and advanced sensors (3D maps for high definition plot and soil analysis), infrared and spectral sensors (drought detection and crop health assessment) depending on the job requirements.

2. In the delivery industry, drones can replace manual delivery to improve delivery efficiency and reduce labor costs. In this environment it is recommended to adopt a multi-axis UAV performance because the flight of multi-axis UAV is more stable compared to other airframes, which can reduce the probability of loss of deliverables in flight, using battery power to facilitate reuse, equipped with high-performance motors to increase the load, accurate GPS and high-precision sensors to guarantee the accuracy of delivery. At the same time, due to the limitation of battery power, it is necessary to set up express distribution points and distribution bases for drones, and adopt the "honeycomb" transit point mode to provide charging and maintenance points for drones.
3. In infrastructure construction, such as high-voltage power transmission circuit inspections, different drones are selected according to work categories. For example, for inspecting "point" targets such as high-voltage transmission towers, multi-axis or helicopter drones can be used to carry high-definition equipment for close observation. For long-distance "line" targets such as transmission lines, fixed-wing drones can be used, equipped with high-speed cameras to inspect line channels and the surrounding environment.
4. In the firefighting, in addition to the above example of choosing a different body and energy, a professional fire detection system needs to be used as well, such as an infrared thermal imaging system (detection of fire ignition point), equipped with a firefighting system in a large UAV for safe remote firefighting, transport rescue ropes, breathing apparatuses and other pieces of equipment to help firefighting.

Therefore, the airframe structure of the UAV will be related to the needs and conditions of its mission, and the application equipment it carries will vary according to the mission requirements of the UAV.

In future experiments, larger UAVs will be used for equipping more experimental parts. First of all, more high-definition cameras can be added to observe various targets in detail and take higher-definition photos for target identification, as well as some special cameras, such as infrared cameras and thermal imaging night vision lenses, to test night flight capabilities. Then, by adopting a new flight control system to replace the current "radio signal + controller" one, we can adopt simple software programming, so that it can be more convenient and direct control in the computer or mobile phone and other mobile devices. Then, more sensors can be installed, and according to the sensor data feedback and the basic program previously programmed, it can be upgraded and optimized to design a semi-automatic automatic flight program. Although there are some commercial drones on the market today that already have some of the automatic flight capabilities, the current flight capabilities of these drones need to be improved. At the same time, these commercial drones are expensive and function

basic functions such as aerial photography, which cannot meet some industrial specialized operations. So the last step is to keep the drone cost as low as possible, if possible, and try to design modular drone components so that the drones can easily and quickly carry out specialized work with different requirements and objectives.

UAV is one of the future trends in aviation, currently in the field of aviation airframe materials, flight positioning, aviation control breakthroughs. As the technology matures and the cost of spare parts decreases, small civilian drones, mainly multi-rotor drones, will become a hotspot in the IoT market. Due to their range, speed and other characteristics, multi-rotor drones oriented to secure, low speed and stability-oriented requirement missions have a large space for application, especially in specific flight characteristics of aerial photography, aerial inspection and other areas of the Internet of Things. Since the current battery technology is not especially large, multi-rotor UAV stall time is generally about 20-30 minutes, and there are several other constrains taking place (radio or 4G/5G wireless network control are of low reliability, anti-interference ability is weak, stability can be found lacking sometimes, the accuracy of GPS system, barometer altimetry system, accelerometer, gyroscope, geomagnetometer and other equipment commonly used in aviation is far from reaching the accuracy requirements of unmanned control), the real, industrial-scale IoT UAVs that become large-scale, high-precision and stable will take some time.

With the development of science and technology, in order to make possible the automated intelligent, autonomous and complex industrial drones in the future, the next direction of work is to carry out research and development of autonomous flight of drones through onboard artificial intelligence, namely "flight control of artificial intelligence drones". These technologies involve technical challenges in various fields such as UAV perception, 3D modeling, image processing, autonomous decision-making, etc., and will achieve standardization and fine inspection that cannot be achieved by manual control in current UAV inspection applications.

8. References and Bibliography

[1] Ashton, K. (22 June 2009). “That ‘Internet of Things’ Thing”. *RFID Journal.* Retrieved February 8, 2020, https://www.rfidjournal.com/articles/view?4986

[2] Lueth, L.K. (19 December 2014). “Why the Internet of Things is called Internet of Things: Definition, history, disambiguation”. *IoT anaalytics*. Retrieved February 24, 2020, https://iot-analytics.com/internet-of-things-definition/

[3] International Telecommunications Union. (2005). *The Internet of Things* [PDF file]. Retrieve February 8, 2020, https://www.itu.int/dms_pub/itu-s/opb/pol/S-POL-IR.IT-2005-SUM-PDF-E.pdf

[4] “IoT vs M2M — What is the Difference?” (16 January 2019). *Avsystem*. Retrieved February 24, 2020, https://www.avsystem.com/blog/iot-and-m2m-what-is-the-difference/

[5] Fingerman, A. (15 Feb 2019). “The IoT 2019: Many "things" ahead, but here are four predictions”, *ClickZ,* Retrieve February 24, 2020, https://www.clickz.com/whats-ahead-for-iot-many-things-2019/226144-2/226144/

[6] Internet of Things Applications - From Research and Innovation to Market Deployment - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/The-IoT-Different-Services-Technologies-Meanings-for-Everyone-77_fig6_278798179 [accessed 8 Feb, 2020]

[7] Wikipedia contributors. (2020, February 7). Internet of things. In Wikipedia, The Free Encyclopedia. Retrieved 17:35, February 8, 2020, from https://en.wikipedia.org/w/index.php?title=Internet_of_things&oldid=939589508

[8] Jensen, J. (26 April 2019). “Drone Photography: A Guide to Capturing Images Like A Professional”, *UAV Coach*, Retrieve February 24, 2020, https://uavcoach.com/drone-photography/

[9] Reid, D. (11 Jan 2018). “A swarm of armed drones attacked a Russian military base in Syria”, *CNBC,* Retrieve February 24, 2020, https://www.cnbc.com/2018/01/11/swarm-of-armed-diy-drones-attacks-russian-military-base-in-syria.html

[10] Wilke, J. (05 June 2019), “A drone program taking flight”, *Amazon,* Retrieve February 24, 2020, https://blog.aboutamazon.com/transportation/a-drone-program-taking-flight

[11] Oxford Learner`s Dictionaries. Retrieved February 9, 2020, https://www.oxfordlearnersdictionaries.com/definition/english/drone_1?q=drone

[12] International Civil Aviation Organization. (2011). *Unmanned Aircraft Systems (UAS)* [PDF file]. Retrieved February 9, 2020, https://www.icao.int/Meetings/UAS/Documents/Circular%20328_en.pdf

[13] Wikipedia contributors. (2020, February 7). Unmanned aerial vehicle. In *Wikipedia, The Free Encyclopedia*. Retrieved 19:38, February 9, 2020, from https://en.wikipedia.org/w/index.php?title=Unmanned_aerial_vehicle&oldid=939590146

[14] Tice, Brian P. (Spring 1991). "Unmanned Aerial Vehicles – The Force Multiplier of the 1990s". *Airpower Journal*. Archived from the original on 24 July 2009. Retrieved February 9, 2020, from https://web.archive.org/web/20090724015052/http://www.airpower.maxwell.af.mil/airchronicles/apj/apj91/spr91/4spr91.htm *"When used, UAVs should generally perform missions characterized by the three Ds: dull, dirty, and dangerous"*.

[15] Wikipedia contributors. (2020, February 21). General Atomics MQ-1 Predator. In *Wikipedia, The Free Encyclopedia*. Retrieved 00:04, February 25, 2020, from https://en.wikipedia.org/w/index.php?title=General_Atomics_MQ-1_Predator&oldid=941942704

[16] O'Donnell, S. (16 June 19). "A Short History of Unmanned Aerial Vehicles", *Consortiq,* Retrieved February 24, 2020, From https://consortiq.com/en-gb/media-centre/blog/short-history-unmanned-aerial-vehicles-uavs

[17] Floreano, Dario; Wood, Robert J. (27 May 2015). "Science, technology and the future of small autonomous drones". *Nature*. 521 (7553): 460–466. Bibcode: 2015Natur. 521..460F. Doi: 10.1038 / nature14542. PMID 26017445.

[18] "Drones to venture into Fukushima containment vessels", (22 March 2018), *World nuclear news*, Retrieved February 24, 2020, https://www.world-nuclear-news.org/Articles/Drones-to-venture-into-Fukushima-containment-vesse

[19] Colaboradores de Wikipedia. *Vehículo aéreo no tripulado* [en línea]. Wikipedia, La enciclopedia libre, 2020 [fecha de consulta: 9 de febrero del 2020]. Disponible en <https://es.wikipedia.org/w/index.php?title=Veh%C3%ADculo_a%C3%A9reo_no_tripulado&oldid=122931945>.

[20] Jason, B. (6 June 2017). "4 Open Source Drone Projects". *Open Health News.* Retrieved February 10, 2020, from http://www.openhealthnews.com/story/2017-06-06/4-open-source-drone-projects

[21] Wikipedia contributors. (2020, February 14). Arduino. In *Wikipedia, The Free Encyclopedia*. Retrieved 15:22, February 14, 2020, from https://en.wikipedia.org/w/index.php?title=Arduino&oldid=940745861

[22] Arduino Store, Retrieved February 14, 2020, From https://store.arduino.cc/arduino-uno-rev3

[23] Wikipedia contributors. (2020, February 14). Raspberry Pi. In *Wikipedia, The Free Encyclopedia*. Retrieved 17:20, February 14, 2020, from https://en.wikipedia.org/w/index.php?title=Raspberry_Pi&oldid=940774973

[24] Wikipedia contributors. (2020, February 12). BeagleBoard. In *Wikipedia, The Free Encyclopedia*. Retrieved 19:22, February 16, 2020, from https://en.wikipedia.org/w/index.php?title=BeagleBoard&oldid=940473164

[25] Mouser Electronis, Retrieved February 15, 2020, From https://eu.mouser.com/new/beagleboardorg/beaglebone-blue

[26] Octavo Systems, Retrieved February 15, 2020, From http://octavosystems.com/octavo_products/osd335x/

[27] Aerotenna, Retrieved February 15, 2020, From https://aerotenna.com/ocpoc-cyclone/

[28] Wikipedia contributors. (2019, October 15). PX4 autopilot. In *Wikipedia, The Free Encyclopedia*. Retrieved 21:52, February 16, 2020, from https://en.wikipedia.org/w/index.php?title=PX4_autopilot&oldid=921403090

[29] Pixhawk Wiring Quick Start, Retrieved February 15, 2020, From https://docs.px4.io/v1.9.0/en/assembly/quick_start_pixhawk.html

[30] Paparazzi The Free Autopilot, Retrieved February 15, 2020, From http://wiki.paparazziuav.org/wiki

[31] E. Baskaya, G. Manfredi, M. Bronz and D. Delahaye, "Flexible open architecture for UASs integration into the airspace: Paparazzi autopilot system," *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, Sacramento, CA, 2016, pp. 1-7.

[32] LibrePilot/OpenPilot Wiki, Retrieved February 15, 2020, From https://opwiki.readthedocs.io/en/latest/user_manual/cc3d/cc3d.html

[33] Dronecode PX4 Autopilot, Retrieved February 15, 2020, From https://docs.px4.io/v1.9.0/en/flight_controller/snapdragon_flight.html

[34] Okdo, Retrieved February 15, 2020, From https://www.okdo.com/p/beagleboard-beaglebone-blue/

[35] Arduino Store, Retrieved February 15, 2020, From https://store.arduino.cc/arduino-nano-every

[36] Kubii Store, Retrieved February 15, 2020, From https://www.kubii.es/1401-raspberry-pi-zero-v13-kubii-3272496006973.html?src=raspberrypi

[37] Navio2 Overview, Retrieved February 15, 2020, From https://ardupilot.org/rover/docs/common-navio2-overview.html

[38] Caleberg (March 28, 2019), "Open Source Drone Software Projects". *Drone Dojo*. Retrieved February 16, 2020, from https://dojofordrones.com/open-source-drone/

[39] Wikipedia contributors. (2019, September 24). LibrePilot. In *Wikipedia, The Free Encyclopedia*. Retrieved 03:08, February 17, 2020, from https://en.wikipedia.org/w/index.php?title=LibrePilot&oldid=917569377

[40] DJI AGRAS MG-1S, Retrieved February 17, 2020, From https://www.dji.com/es/mg-1s

[41] Parrot eBee SQ, Retrieved February 17, 2020, From https://tycgis.com/ebee-sq-sensefly/

[42] "Camclone T21 unmanned autonomous vehicle (UAV) fitted with CSIRO guidance system". *Science Image*. Retrieved February 17, 2020, From https://www.scienceimage.csiro.au/library/transportation/i/10876/camclone-t21-unmanned-autonomous-vehicle-uav-fitted-with-csiro-guidance-system/

[43] "Microdrones in test flight for DHL". *Microdrones*. Retrieved February 17, 2020, From https://www.microdrones.com/en/content/microdrones-in-test-flight-for-dhl/

[44] Pregler Art (February 21, 2017). "When COWs Fly: AT&T Sending LTE Signals from Drones", *AT&T*. Retrieved February 17, 2020, From http://about.att.com/innovationblog/cows_fly

[45] "Top 5 commercial drone use cases".(05-09-2018).*COPTRZ commercial drone experts*, Retrieved February 17, 2020, From https://www.coptrz.com/top-5-commercial-drone-use-cases/

[46] "Archived Topic: PXFmini". *Ardupoilot*, Retrieved 21:52, February 16, 2020, from https://ardupilot.org/copter/docs/common-pxfmini.html

[47] EACHINE Retrieved March 18, 2020, From https://www.eachine.com/Eachine-Tyro69-105mm-F4-OSD-2.5-Inch-2-3S-DIY-FPV-Racing-Drone-PNP-w--Caddx-Beetle-V2-1200TVL-Camera-p-1583.html

[48] EACHINE. *TYRO69 Quick Start Guide (DIY)* [PDF file]. Retrieve March 14, 2020, http://myosuploads3.banggood.com/products/20190822/20190822004627TYRO69.pdf

[49] FLYSKY. *FS-i6X Instruction Manual* [PDF file]. Retrieve July 10, 2020, https://static1.squarespace.com/static/5bc852d6b9144934c40d499c/t/5d6f66305487dc0001e34939/1567581800555/FS-i6X+User+manual+20160817.pdf

[50] "BetaFlight Wiki".(14-06-2020).*GitHub*, Retrieved July 10, 2020, From https://github.com/betaflight/betaflight/wiki

[51] "BetaFlight Modes".(24-01-2020).*GitHub*, Retrieved July 10, 2020, From https://github.com/betaflight/betaflight/blob/master/docs/Modes.md

[52] FPV monitor receiver, Retrieve July 10, 2020, From https://www.banggood.com/5_8G-48CH-4_3-Inch-LCD-480x22-169-NTSCPAL-FPV-Monitor-Auto-Search-With-OSD-Build-in-Battery-p-1364626.html?rmmds=search&cur_warehouse=CN

[53] Remote Controller, Retrieve July 10, 2020, From https://www.banggood.com/Flysky-FS-i6X-2_4GHz-10CH-AFHDS-2A-RC-Transmitter-With-X6B-i-BUS-Receiver-p-1090406.html?rmmds=myorder&ID=42482531076&cur_warehouse=CN