Isfahan University of Technology

Faculty of Mechanical Engineering

Quadrotor control using robot operating system () ROS

Master's thesis in mechanical engineering (mechatronics orientation)

Nabil Haider

Supervisor

Dr. Maziar Palhang

Dr. Mohammad Mashayikhi

1401

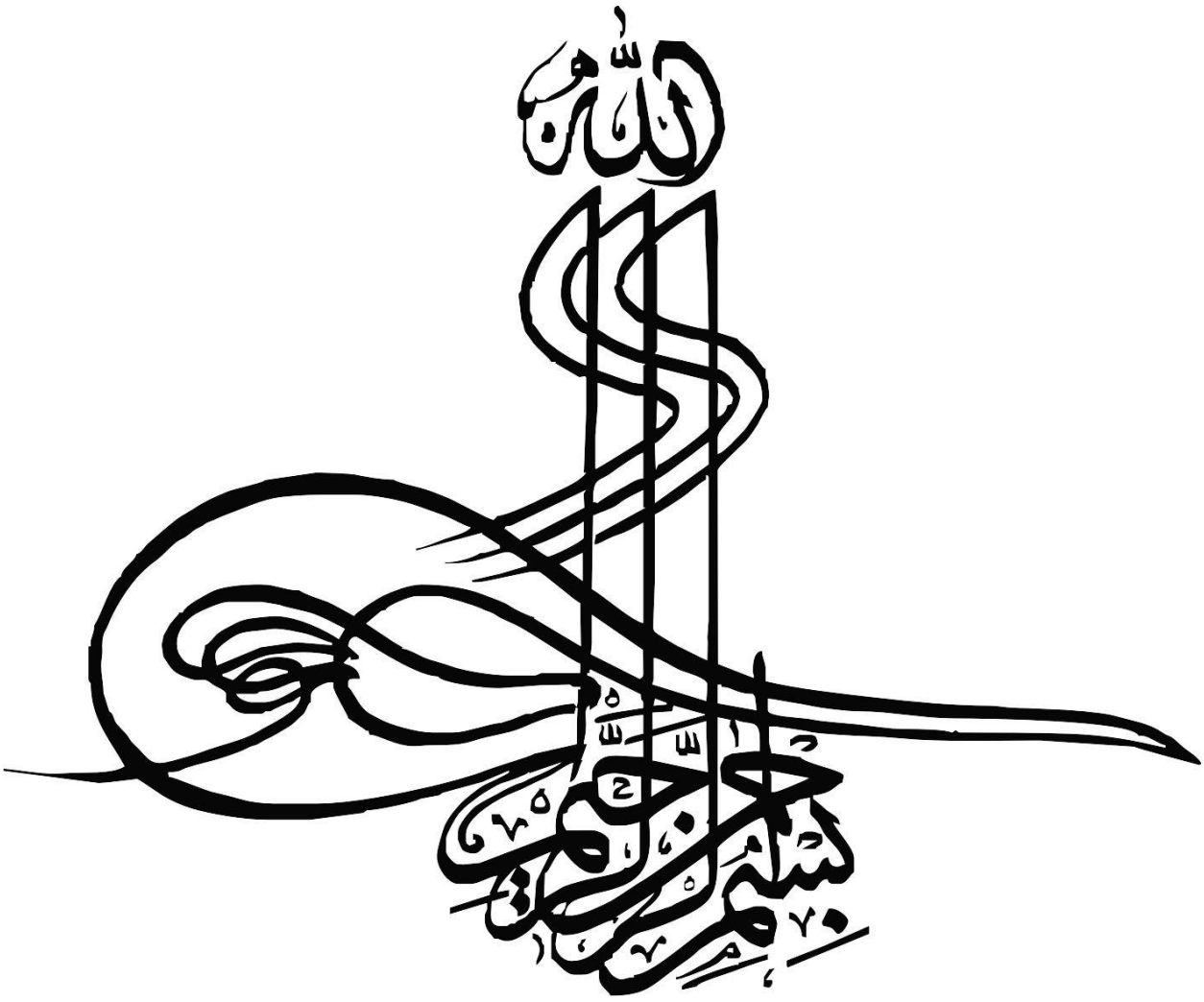

بسم الله الرحمن الرحيم

Table of Contents

1- The first chapter: Introduction and definitions 1

1-1 The goal of the project 1

1-2 An overview of the work done 1

1-2-1 Quadcopter assembly..... 2

2-2-1 Reading articles and searching 2

3-2-1 Quadrotor control with a remote radio controller (**RC**) And**Mission Planner**.....2

4-2-1 Simulation with robot operating system**ROS**) And**GAZEBO**..... 2

5-2-1 Connection of the quadrotor with the robot's operating system**ROS**)..... 3

3-1 Results obtained 3

4-1 Thesis Sections..... 4

2- The second chapter: hardware and software 5

2-1 Hardware..... 5

1-1-2 frame 5

2-1-2 Locusts 6

3-1-2 Engines 6

4-1-2 Electronic speed controllers**ESC**) 7

5-1-2 Flight Controller (Flight Controller) 8

6-1-2 Radio control (transmitter and receiver) 9

7-1-2 radio telemetry module**Sik**Sender and receiver 10

8-1-2 sensor**GPS**..... 11

9-1-2 Battery 11

10-1-2 Propeller protector 12

2-2 Software..... 14

1-2-2 Flight controller..... 14

2-2-2 ground station**Ground Station**) 17

3-2-2 Operating system 19

4-2-2 Robot operating system**ROS**) 19

5-2-2 Simulation 19

3-2 summary and summation..... 20

3- The third chapter: connection and simulation..... 21

1-3 description21..... ....MAVLink

1-1-3MAVLinkwhat?..... 21

2-1-3 Understanding the messages**MAVLink**..... 21

3-1-3 message format**MAVLink**..... 23

2-3 Communication with the ground station 24

3-2-1Installing firmware or operating system 24

2-2-3 connection with cable**USB**..... 25

3-2-3 Communication with telemetry..... 26

4-2-3 Calibration 26

3-3 Starting the engines using27..... .Mission Planner and RC

3-3-1flight modes (Flight Modes) 27

2-3-3 Turn on the engines using**RC**..... 28

3-3-3Turn on the engines using**Mission Planner**..... 29

4-3 simulation with30..... ..... GAZEBO and MAVProxy

1-4-3MAVProxywhat? 30

2-4-3 Simulator**GAZEBO**what? 31

3-4-3 Setting the environment**SITL**and installation**GAZEBO**..... 31

5-3 summary and conclusion..... 31

4- Chapter four: Communication with the robot operating system (..... .....) ROS32

1-4 robot operating system (32..... .....) ROS

1-1-4**ROS**what? 32

2-1-4 Why do we use the robot operating system? 32

3-1-4 Architecture**ROS**..... 33

4-1-4 working space and closed**ROS**what are they 36

2-4What is? ....37 Raspberry Pi

3-4 packs38..... ....MAVROS

1-3-4 package definition**MAVROS**..... 38

2-3-4 closed structure**MAVROS**..... 39

3-3-4 package installation**MAVROS**..... 40

4-4 connecting toon the laptop ....40 SITL in ROS

5-4 actual communication with41..... ..... ROS

1-5-4 communication withSerial Port..... 41

2-5-4 connection with cable**usb**Or Telhamtri 42

6-4 program execution43..... ....ROS

4-7 Additional items: connecting laptop to Raspberry Pi by44..... ..... SSH

5- The fifth chapter: Results..... 46

Not

the attachment 48

the attachment1: Setting up the environment for SITL and its implementation..... 48

the attachment2: GAZEBO installation and its implementation 49

the attachment3: Installing the MAVROS package on the laptop: 51

the attachment4: Installing MAVROS package on 51 .....raspberry pi

the attachment5: Connecting to ROS in SITL on the laptop..... 53

the attachment6: Writing and running the program 55 ..... ROS

Reference 60

List of shapes

Shape1-2 framesF450quadrotor[2]..... 6

Shape2-2 - Propeller 1045 quadrotor[3]..... 6

Shape3-2 - Brushless motorV[5] EMAX MT2216 810K..... 7

Shape4-2 - Speed control20AeriesSimon[7] EMAX S..... 7

Shape5-2 - History of Ardopilot's wins[11]..... 9

Shape6-2 - Win[13] APM2.8..... 9

Shape7-2 - Remote radio controller (transmitter)Z[14] JR DMSS XG6 2.4GH..... 10

Shape8-2 - Radio receiver[15]..... 10

Shape9-2 - Telemetry transmitter and receiver module[18]..... 11

Shape10-2 sensors[19] GPS..... 11

Shape11-2 Lithium polymer battery[21] TATTOO..... 12

Shape12-2 lithium polymer batteryceAens[22] G..... 12

Shape2-13 Protective design in software..... 13

Shape2-14 The shape of the propeller guard on the UAV 13

Shape2-15-The map of the connections of the parts together]25[..... 14

Shape16-2 Ardopilot firmware types[28]..... 17

Shape17-2-Section map of Isfahan University of TechnologylannerPissionM..... 18

Shape18-2-architecture[35] SITL..... 20

Shape3-1-Split message bytesink[38] MAVL..... 22

Shape33-2-How to exchange information between the drone andlannerPissionMby messagesink[39] MAVL..... 23

Shape3-3-Detailed message table[40] COMMAND_LONG..... 23

Shape3-4 firmware installation pagelannerPissionM..... 24

Shape3-5 buttonCONNECTAvailable at the top of the pagelannerPissionM..... 25

Shape3-6-The connection stage with the quadrotor (getting the parameters) 25

Shape7-3 buttonconnectin the softwarelannerPissionM..... 26

Shape8-3lannerPissionMConnecting with quadrotor by telemetry 26

Shape9-3 pages that include the calibration of various partslannerPissionM..... 27

Shape10-3 the three main axes of the quadrotor:roll,pitchAndyaw[47]..... 28

Shape3-11-The mode of arming the engines inRC..... 29

Shape12.3 DivisionUDDERR,THROTTLET,ILERONAAndLEVATOREIn a[50] RC..... 29
ShapePart 13-3ISARM/DRMAATLANNERPISSIONMTo turn on and off the engines..... 30
Shape3-14-ground stationROXYMAVPIn the Linux operating system[31]..... 30
Shape3-15-Quadrotor simulation in the simulatorGAZEBO..... 31
Shape4-1-The connection of two nodes with the main node[55]..... 34
Shape2-4 publisher and subscriber communication in[55] ROS..... 35
Shape4-3-Communication between publisher and subscriber by topic and determining topic name and message type[56]..... 35
Shape4-4 server and client communication (with services)[55]..... 36
Shape4-5 Communication between server and client (with actions)[55]..... 36
Shape6-4 Work space structure[58] ROS..... 37
Shape7-4 win3BRASPBERRY PI][59]..... 38
Shape4-8-Brief map of flight controller communication process withROSAnd also with an external sensor[62]..... 39
Shape4-9-The process of connecting the flight controller withRASPBERRY PIby usingSERIAL PORT[65]..... 41
Shape4-10-ConnectionB3RASPBERRY PIwith portsTELEMETRYFlight controller[66]..... 42
Shape4-11 Active and inactive optionsERIAL[65] S..... 42
Shape4-12-Options given by running the programPY.CONTROL_APM..... 44
Shape4-13-program behaviorPY.CONTROL_APMBy dialing the number 7 44
Shape4-14-program behaviorPY.CONTROL_APMby writingC..... 44
Shape0-1-Top left:CONSOLEand lower left side: terminal and right side: map 49
Shape2-0 SimulatorGAZEBOWithOPTERCRDUAVirtual in it..... 50
Shape0-3-Full list of flight mode numbers[73]..... 55

Chapter 1: Introduction and definitions

1-1 the project goal

The goal of this project is to enter the world of robots through the wide doors of the robotics field, just like any other field of science. In order to start using intelligence in robotics, we must first learn the most suitable methods to do it. One of these methods is the robot operating system. Thank you for your help,One of the main ROS development tools For robotics software engineers, it has been developed to allow the programmers to work and code in a flat language and build complex robots in a very simple and easy way. The use of the robotic system will be discussed more later, and the second case where the project's goal is felt is drones, robotic vehicles. Today's world challenges are technological challenges that leading countries are trying to improve, and we can solve them in all different fields. Animation, animation and filmmaking related to work in the field of drone programming, especially usingAnd sometimes everyone needs knowledge of C++ and Python and ROS. that some of the programs are bilingual developers and sometimes developers need to understand the codes in order to add their own edits to it. Programming was a big part of the project's goal.

The main goal is to develop unmanned aircraft technology and use robotic technology that has inherent intelligence, this thesis is a part of this goal.

2-1 An overview of the work done

1-2-1 Quadcopter assembly

In this project, a quadrut is used in the laboratory of intelligence, which has a kind of computer code. Like all UAVs, this UAV also required several types of hardware and software to perform its work, which is detailed in**second chapter**We will say hardware and software

2-2-1 Reading articles and searching

In order to complete this project, it was necessary to study various documents and information about the hardware used in the robot, how to run and control the robot, etc. the bottom of the earthand clones, including ROS They were

3-2-1 Quadrotor control with a remote radio controller (RC) and Mission Planner

The goal was not to use the engineOne station (MissionPlanner) using RC software Earth) Before turning on the engines, the calibration of the different sensors should be done, and for the best result, the calibration should be done in space. The main inputs that need to be calibrated are: speed, direction, speed tens control (And you can't see it, you can't see it, you can't see it, and you can't see it Use of RAMFZARRC can only be calibrated using ESC, while Mission Planner Calibrating the gear is the same as controlling the speed correctly. Lash Bari Rafdain Mesh Kol reached the result that Mesh Kol fromStep by step "Throttle" button and take it from the RC channel. Tensiles in the fast press, control tens of speed on the broadband ion modulus (PWM and now the engines do not turn on without any problem after calibrating the speed controller, the engines turn on byOr the USB connection between the quadruple and the software through the Mission Planner The telemetry module is performed

4-2-1 Simulation with Robot Operating System (ROS) and GAZEBO

It is always better to run a simulation before doing any work that might cause some damage, if this option is required.GAZEBO robot simulation software It simulates a real world robot and can be controlledConnect to the virtual reality simulation of ROS Do not use software in loopsSITL[1]This is a tool provided by Azandeh Autopilot, which allows the user to know everything using different software. Do not use any simulation hardware in this case, programsIt can be developed and if it is decided to ROS Work well, it will be tested when it is closedIt was created to control Quadrotor using ROS

---

[1] Software in the loop

Easy-to-use instructions for drinking this packagePython ROS is used It is possible to take off and land the quadcopter, change the flight mode, turn off the engines and provide its current coordinates before usingand SITL in simulation, a simple simulation using ROS Benham ground stationIt is concluded that it also allows the server to use any MAVProxy. Control and test the UAV hardware

5-2-1 Quadrotor communication with the robot operating system (ROS)

Coming to the real world, there is a greater possibility that the simulation cannot be simple and very effective using real world soil. There is a difference in the simulation, the software is completely up-to-date and cannot be used by the general public because the control board is old and by the old versions. Seriously, it is not supported by the software, many softwares must be updated and older versions must be installed for this to work.As a computer (Jexin laptop) and Raspberry Pi software was used. The old one is downloaded on it and is ready to be used to connect to Quadrut.ROS ToROS, which can understand Quadrotor in ROS language for Quadrotor, ROS package and ROS message And they translate vice versaIt is ready, the connection was successful and the ROS version that controls the flight and the MAVROS version is not downloaded after many attempts. From [1] MAVROS Written and tested

3-1**The results obtained**

My world is developing rapidly and to have the best performance, the coolest work, and a better net, it is always better to use the hardware that supports it. Especially when the goal to be reached must be a certain type of intelligence inside the laboratory, using aThe case of analysis was decided and a special node was used to write the Python code, MAVROS, a different node index of the rqt_graph package. Also by using the services It is possible to land on the condition of flying the drone and read the MAVROS song by song Its messages and parameters are available using the packageAfter drinking, the motor will turn off and ROS The coordinates of the drone should be tested outside the test at each point of takeoff and landing (the plan to write the report has not been tested outside yet)

4-1**Dissertation sections**

The second chapter shows the different software and hardware used in this project. In this chapter, we will talk in detail about the entire Quadrotor from the outside and inside.

The third chapter is about the connection of the UAVand ion calibration processes and in the case of simulation with Mission Planner usingGAZEBO and MAVProxy, SITL will be discussed

What do we mean?And finally, it shows the MAVROS connection as a summary of the ROS package. drone to Using simulation and in a real way, it explains in this chapter the whole package of ROS. It will be explained for the quadrotor control and all the results will be shown at the end

The last chapter talks about the summary of the problems we have faced and the results

Chapter Two: Hardware and Software

In this project, there is a quadrut working in the intelligence lab, which has a kind of code and computer knowledge, and it was made by Dr. Palhang, and it was used as a drone. The drone also has several types of hardware and software to do the work. We will concentrate

1-2**hardware**

1-1-2 frames[2]

From the quadrut ver frameIt has been used (Figure 2-1) on 450 mm and 55 mm height F450 It has its weightIt is 295 grams. Some of the specifications of this frame [2]:

- Made of high quality glass fiber and durable polyamide nylon

- The arms are reinforced to prevent and reduce breakage

- connectionsPCB[3]Integrated for direct soldering of speed controllers

- Colored arms for orientation to keep flying in the right direction

- It is easy to assemble

---

[2] Frame
[3] Printed circuit board

Shape1-2 frames F450 quadrotor*]2 [*

2-1-2 Locusts

from the grasshopper1045 has been used (Figure 2-2), which means that the length of 10 inches is 254 mm, and the pitch is 4.5 Its weight is inches14 grams [3]

Shape2-2 - Propeller 1045 quadrotor*]3 [*

3-1-2 Engines

Used motors, brushless motors[4](Figure 2-3) EMAX MT2216 810Kv

Power:228 watts

Projectile force:1070 grams

current:15.4 amps

Voltage:14.8 volts

Shaft diameter:4 mm

Weight:62 grams

Brushless[4]

More information about the specifications of this engine in this article]4[is available

Shape3-2 - Brushless motor EMAX MT2216 810Kv*]5[*

4-1-2 Electronic speed controllers (ESC[5])

Electronic speed controllers, which are also called speed controllers, are one of the most important parts of the quadrotor.It is an electronic circuit that controls and tunes the speed of the electric motor, it is also possible (ESC). It provides motor delay and dynamic braking. This unit is responsible for changing the motor speed according to the user's commands through remote control or joystick. Most of the time, these speed controllers are not programmed directly, but they can be programmed in advance by the factory. master programmed or by uploading the operating system or firmware in the master controller, it programs them, a signal is sent from the flight controller thatIncrease or decrease ESC motor voltage if needed and thus change the speed of the propeller

In the project of[6] A speed controller is needed for each motor (Figure 2-4) EMAX Simon Series 20A has it

Shape4-2 - Speed control EMAX Simon Series 20A*][7]*

Electronic Speed Controller[5]

5-1-2 flight controller

If we want to talk about the flight controller from the physics point of view, there is nothing more beautiful than an electronic device that can be compared to a smart phone. In other words, the Quadrotor brain feels like a small box full of electrical equipment and intelligent software that does everything the drone does. And they control, like the brain of different waves, they also control dozens of flights of different sizes and complexity, for the same reason, multi-rotors (in this project, you are quadruple) d) The parts called flight controllers are needed to perform all the necessary control tasks.

The flight controller is connected to a set of sensors, these sensors transmit information such as the altitude, direction and speed of the quadrotor to the flight controller. It includes inertial sensors to determine speed and angles, pressure sensors to determine height, and distance sensors to detect objects. Just as humans are aware of their surroundings, these quadrotors check and control the obtained information and use it for better performance. Turkish people say that the quadrotor speaks much cooler and makes the necessary diagnoses more accurately.]8[

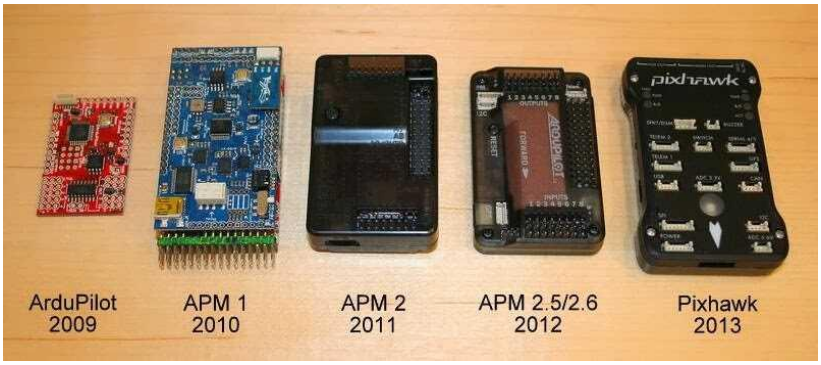
Only by sensing what is happening around the Quadrotor, the flight controller controls the movement of the drone. The Quadrotor is very easy to see. The motor used in it creates a change in speed and this is the reason why it starts to move, so that the quadruple speed reaches the speed to be controlled. It collects data by using sensors to speed up the movement of each person using the sensor, filtering and filtering the sensory information. Estimating the safety and durability of all flights is done by an algorithm and by the flight controller.8[

For this project and since several previous UAV construction tests have been completed, from the flight control panel It has been used (section 6-2), it is necessary to use a support slave [9] APM2.8 It is not recommended to use it when there is a problem, because the user will have to remove it with a new board, so it is better than newer boards, for example,Use [10] Pixhawk

companyIt is based on Arduino and APMs (Figure 5-2) 2.8 Ardupilot is the last board that APMs do not control the hardware of the flight and focus only on the software. K red [9] 2.8 Ardupilot mega[6]It is 2560 and since it is open, it can be reprogrammed

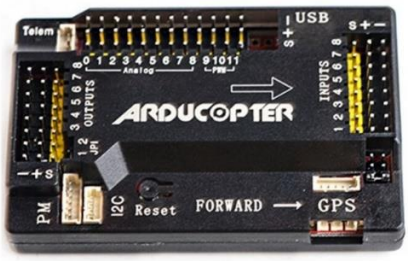
---

[6] Arduino Mega



Shape5-2 - History of Ardopilot's wins*][11]*

After constructionThe world of flight controllers has been conquered by the best flight controller, Pixhawk. economy in this land]12 [and the Ardupilot software is also supported Could be atIt's better because Pixhawk works with APMApple and like Pixhawk 2.8 with a firmWe are developing new AUTOPILOT and Ardupilot firmware every day. In addition to the possibility of hardware, the device has more features in the case of firmware.In the ArduPilot section**!Reference source not found. Error**will be spoken



Shape6-2 - APM2.8 board*][13]*

6-1-2 Radio control (transmitter and receiver)

As a complete radio and control tool for controlling robots and quadrotors, these control devices work only by radio and wireless methods. It is possible to use it with the help of a radio frequency to sync the world with gold to create harmony between quadruple andRC trunk Exclusive to the individual that is specific to the receiving device, while not turning on the two devices, the push button and the receiving device will be activated and in this way, the voltage will be increased. R. B. AIt is connected and there is no internal for RC The devices will not be connected to each other in this projectand JR DMSS XG6 2.4GHz (total number 2-7). Receiver RG831B 2.4GHz is used (Figure 8-2).

Shape7-2 - Remote radio controller (transmitter) JR DMSS XG6 2.4GHz*][14]*

Shape8-2 - Radio receiver*][15]*

7-1-2 Sik radio telemetry module transmitter and receiver

Radi and Tel HemtriAnd the Autopilot station is one of the simplest ways to establish a better connection between SiK Hamtari Tel Radio is terrestrialIt's a small, light and cheap wireless RAM platform that uses SiK copper. more thanIt makes it possible to reach 300 meters (by using one of those modules on the ground, that is, a module that is a station The land is connected, it can be extended up to several kilometers. That's enoughOnly (in the third chapter of MAVLink) should be designed Plane and Rover, Copter, Mission Planner and MAVLink. Know thatThe protocol for communicating with the drone [16] is this module with the MAVLink license They allow to be connected to the ground station and be fully controlled. For example, a special mission that includes different stations can be used from this module. Do not send any kind of wireless communication to the drone]17 (Figure 9-2)

Shape9-2 - Telemetry transmitter and receiver module*][18]*

8-1-2 GPS sensor

SM feelerNEO-M8N Flight Controller GPS Module used in this project is GPS is (Fig2-10) This module also has an internal compass

Shape10-2 GPS sensor*][19]*

For readers of Degani KBAand how to choose GPS and how it works, a complete guide about GPS There is the best option for projects]20 [

9-1-2 Battery

The best choice for such a project is lithium polymer batteries, they are efficient and have high efficiency, and two types of batteries are used, oneYou have 3 cells and 4000 Gens Ace cell and 4200 mAh speed [21] and 3 TATTU (total 2-12) Gens Ace and one TATTU (total 2-11) is milliampere hour]22[

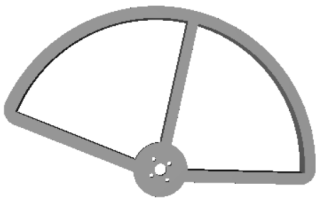
Shape11-2 TATTU lithium polymer battery*]21[*

Shape12-2 Gens Ace lithium polymer battery*]22[*

10-1-2 propeller guard

With the help of RAMFZAR[7]It has been designed for a motor vehicle with a tread guard, [23] Solidworks, CAD (Shape2-13) and installed on the drone (Figure 14-2), this protector protects the drone from damage It helps during testing in the laboratory and it can be shown to the engineers how close their hands can be to the motor, although the design is very simple. , but Aymin says the first word with the help of KKK, all the measurements were done accurately and the first sample was printed as a poster. It was tested on the drone, and the result showed that the length should be longer. After that, it was castrated and then it was installed on the drone. God's help, I got 10 that were made by drone enthusiasts and builders, and they were printed in 3D format.PLA[8]]24 Paying attention to the length of the propeller and being in the frame with the installed engines were designed

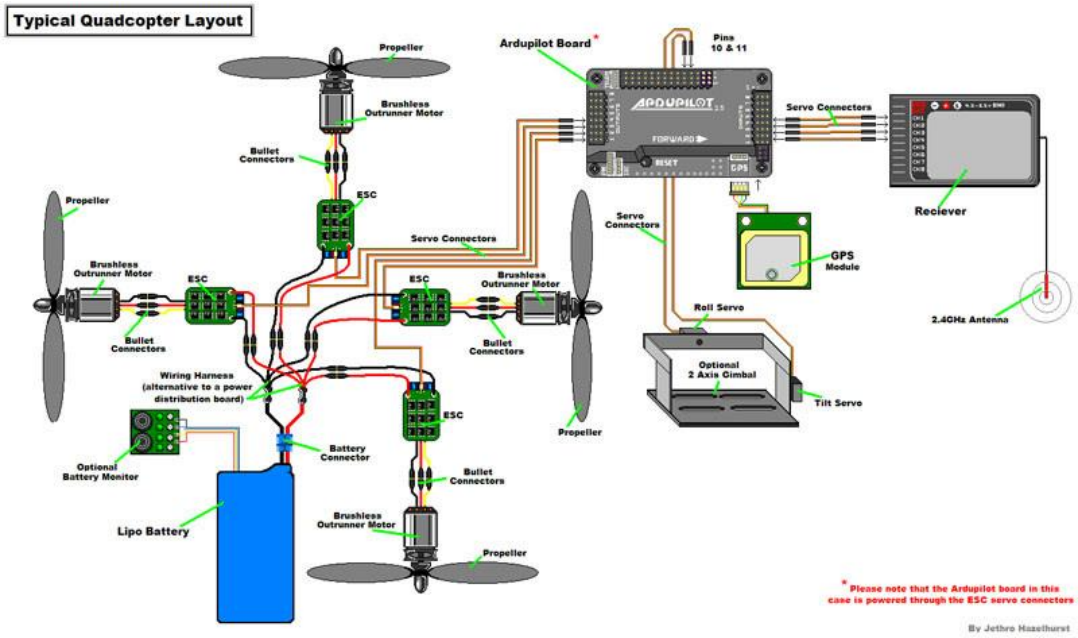
Computer Aided Design[7]
Polylactic Acid[8]



Shape13-2 Protective design in software

Shape14-2 Figure of the propeller guard on the drone

Now the hardware can be mounted on the quadruple (see152-) and brought it to pass, of course Other tools for connecting songsused, later on in ROS**What is the chapter?**: Bat communication My system is related**ROS**For example, many cables, a double batteryand even A OneROS is used to help Raspberry Pi. Raspberry Pi is used. Since the laptop you are using has a new version of Linux, you can download old versions of Linux from Raspberry Pi.Be compatible, there is no basic error, APM is used, the temperature is 2.8 ROS It is important to use laptop, desktop or mobile computerHave ROS



Shape2-15- The map of the connections of parts together*]25[*

2-2**software**

In this section, we will discuss the software used in the project

1-2-2 flight controller

To control the flight, from the firmwareVersion 3.2.1 is used, this is the latest version and in this case, ArduCopter Now it's time to go backThere are many newer versions of Pixhawk that can support APM2.8. Any other newer board can be used

A reliable, all-in-one autopilot system based on a variety of Ardupilot vehicles. Supports: Multiple copters[9] , helicopters, airplanes, boats, submarines, Mars[10]etc. The source code has been created by a large community of professionals and enthusiasts][26]

ArduPilot offers the creation and use of safe, autonomous and unmanned vehicle systems. It makes peace possible for everyoneIt provides solid sets of tools that approximate ArduPilot For any suitable vehicle and program, you can use it as a project based on the continuous development of a large community of users. The development team works with the community and business partners

---

[9] Multicopter
[10] Rover

Redi function bThere is no difficulty in the production of ArduPilot, although ArduPilot It does not work, the general systemon various hardware platforms for ArduPilot control Unmanned vehicles work together with ground control software (ground station), unmanned vehicles thatcan perform advanced functions, including ArduPilot communication electronics Be with the operators immediatelyIt has a great online forum for ArduPilot users. Questions, problems and solutions that help in the first brain of KDBCreated in 2009 by ArduPilot At that time, it has been developed by a team of engineers, scientists, computer scientists and other members of the global community. This autopilot can be approximately Controlling imaginable vehicle systems such as airplanes, door locks [11], multirotors, helicopters, sailboats, electric boats, submarines, ground vehicles and even self-balancing robots.[12]which has been installed in more than 1,000,000 vehicles worldwide, and with an advanced tool for data acquisition, analysis and ArduPilot Analysis and simulation, a deeply tested and reliable autopilot system is the foundation of Mandb's hacked system, meaning it is rapidly evolving. It is always at the edge of technological progressin vehicles from many manufacturers and more widely throughout the ArduPilot industry Self-sufficient global system[13]It has also been set aside for testing and development by institutions and large companies such as NASA.[14], Int. L[15]And And also many schools and universities throughout Boeing the world is used][26]

The summary of the above topic is ArdopiltiaIt is not a hardware, but the ArduPilot software The software is like controlling a flightAPM or Pixhawk[16]It is discussed in the case of controlling dozens of flights in the flight controller section.

For more explanation, here are some of the supported car typesFor any type, like ArduPilot There is a special firmware. (Shape2-16)

Al fun It's a captcha A[17]: Quadrut Var,K, OctaCopter, HexaCopter WadX8[18],TriCopter, Y6 [Helicopter,]27, CoaxCopter, SingleCopter, BiCopter

---

Gliders[11]
Balance-Bots[12]
Global Autonomous Systems Industry[13]
NASA[14]
Intel[15]
ArduPilot Mega[16]
Copters[17]
QuadX8[18]

- Various types of aircraft[27] Equad plane, Plank Flying[19],, Tri-Tilt-Wing QuadPlane Ornithopter, GyroCopter Wing :)Planes

R- Types of Rovers[All types of wheeled robots and boats] 27:Rovers

Depending on the type of robot used, a special firmware for the robot controller must be uploaded, for example, if it is used from an airplane, the firmware must be uploaded.And the Rover submarine should be uploaded, and the same should be uploaded for the Copter, and if the multicopter is used, the Plane firmware should be[20]

and other names of any of the above typesYou are in ArduRover, ArduPlane, ArduCopter Uploading the firmware is not enough, but a lot more must be done, which will be shown later

For example, if the general systemCopter should be uploaded, the controller will realize that the robot is a Copter type. But which type?It ends up that in this program a ground station is detected and the ground stations of Planner Mission are not noticed, it should be assigned to this topic. Use the Copter program in part of**ground station) Ground Station**Is

QuadPlane[19]
Submarine[20]

Shape16-2 Ardopilot firmware types*][28]*

2-2-2 Ground Station

From N.R.M.F.ZarMission Planner was used for the simulation processes, MAVPROXY [31] for some test problems and APM Planner [30] for the main ground station. The latest version is used, which is 1.3.77, although it is from [29] Mission Planner.

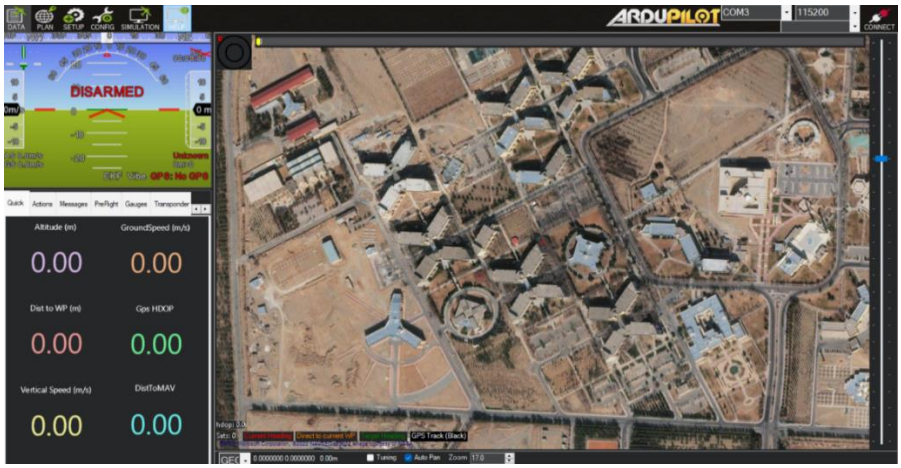
What is a ground station? A ground station is a software program that runs on a computer and communicates with the drone through wireless communication. This real-time data shows the location of the drone and can be used as a "virtual cockpit" and many similar tools. If you were flying in a real plane, it would showGCS[21]It can also be used to control the drone in flight, upload new mission instructions and parameterize the drone. It is also often used to shoot live video streams from drone cameras.

Ground Control Station[21]

H There are about ten different ground control stations on my desk.For tablet/ APM, Mission 2 ear UgCS and QGroundControl, MAVProxy, Planner Planner intelligence MendTower A) 3 There is SidePilot and AndroPilot, MAVPilot, (DroidPlanner He is dead One of those A. B Vote Communicate with[Used] 32 (Copter and Plane, Rover ie) ArduPilot

that by Mike Lob Verne Mission Planner[22]Castrated, it does much more than its name suggests. Here are some of the features of this program:[29]

- Every role can be playedsuch as Mission Planner google maps[23], the roles of G[24]and many others used (Fig2-17)
- Only usingThe drone can be fully controlled by Mission Planner
- It can be used for simulating without using any hardware.In the SITL section **simulation** will be explained)
- Missions and stations on the wayUse the Mission Planner drone to find waypoints sent



Shape2-17- Section map of Isfahan University of Technology in Mission Planner

Michael Oborne[22]
Google[23]
Bing[24]

3-2-2 operating system

Use the general system of LinuxIt only works on Linux (in this project from ROS Wei Reich1.0 of ROS has been used, and 2.0 of ROS has been used on the general system of Levindoze and Mackerel. Can d) Copy Linx Ubuntu and[25]20.04 This is an installation for my laptop and it is a simulation thing. After that, I saw that there was an old Linux distribution andInstall Raspberry Pi on ROS Let's connect toWell done ROS

If my laptop uses Windows, it can be done from a virtual machine[26]To install Ubuntu, he used the process of installing a virtual machine on the Windows operating system in this article.33[is available

4-2-2 Robot Operating System (ROS)

In LPTMB Rai Nass BIt is used, of course, it is installed on Linux, ROS Noetic from ROS Nehroi Windos Shrond Nass b[On the general Linux system (Ubuntu distribution) in this link] 34 ROS There is information about the operating system of the robot in this section**architectureROS**is spoken

5-2-2 Simulation

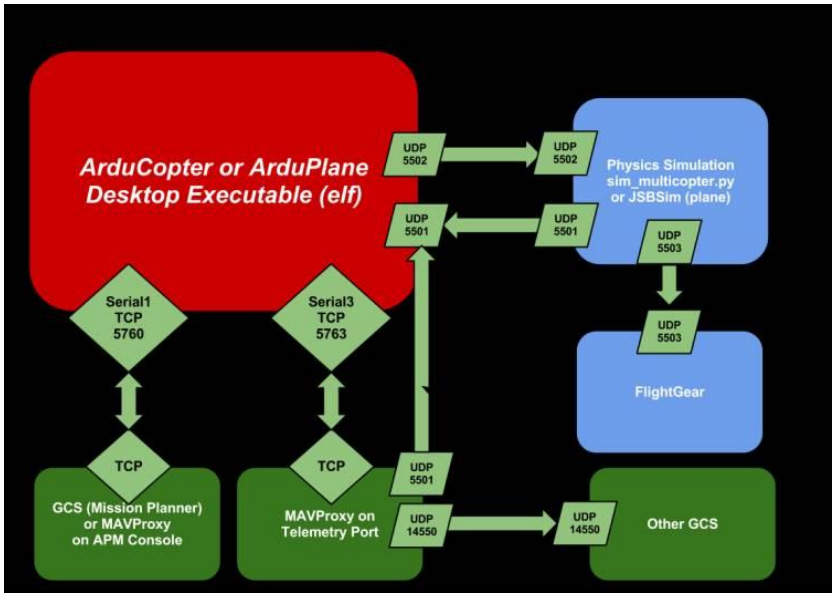
for simulation of a tool called software in the ring (SITL[27]) is used (total*18-2*) SimulatorSITL (Software in the ring) allows the user to run without any hardware.Rover, Copter, Plane slow] [35]

It is possible to run SITL directly on your computer. ArduPilot enables SITL Linux (architecture only[Mac and Windows, or run in a virtual machine] (36, x86).

FromIf the test data is used, start by using the ROS test simulation for the SITL test codes. And then it would be uploaded on the actual hardware, and you can enter every CDWrote and developed it to SITL Easy to test

---

[25] Ubuntu
[26] Virtual Machine
[27] Software in the loop



Shape18-2- SITL architecture*][35]*

3-2**Summary and freezing**

In this chapter, the types of software and hardware used in this project are described. After reading this chapter, the reader should be familiar with the tools. In the next chapter, the stages of preparing the drone for flight in two ways, simulation and real, will be shown. We will put a word

In this chapter, we will talk about the connections made between the UAV on one side and the ground station and radio control on the other side, but before starting, it is necessary to know the form of the messages that are sent from the ground station to the UAV and vice versa. aboutis MAVLink

1-3**Evil**MAVLink

1-1-3MAVLink**What is**

A protocol for communicating between a small unmanned vehicle and a Micro Air Vehicle Link or MAVLink Isgah is terrestrialFirst published in early 2009 by Lawrence Meyer of MAVLink[28]Under licenseLGPL[29]Published mainly for communication between the ground control station (Vehicle orientation and speed, vehicle GPS can be used to transfer the location of the subsystem and unmanned vehicles and in communication between subsystems or GCS. used][16]

Therefore, the UAV sends messages to the ground station, which allows the ground station to report the status of the UAV. The ground station can also send messages to the UAV to perform some actions, such as controlling the movement of the UAV, so the protocolBasically, sets of messages, and their structures and formats, and how they are exchanged between the UAV and MAVLink Specifies the ground station

2-1-3 Understanding MAVLink messages

MessageIt is based on the byte sent by the terminal (called "msg" of MAVLink). land such as d USB encoding and through Mission Planner[30]OrAPM to Telemetry It will be sent when PM is selectedIf you like it, you can configure MAVLink with the package options They give after the word closed or[Used] 37 Packets

Each packageIt is 17 bytes long and has the following structure: (Figure 13-) MAVLink

---

Lorenz Meier[28]
Lesser General Public License[29]
Universal Serial Bus[30]

Message length =17 (6 bytes Header + 9 bytes Payload + 2 bytes Checksum)

Shape1-3- Division of MAVLink message bytes*][38]*

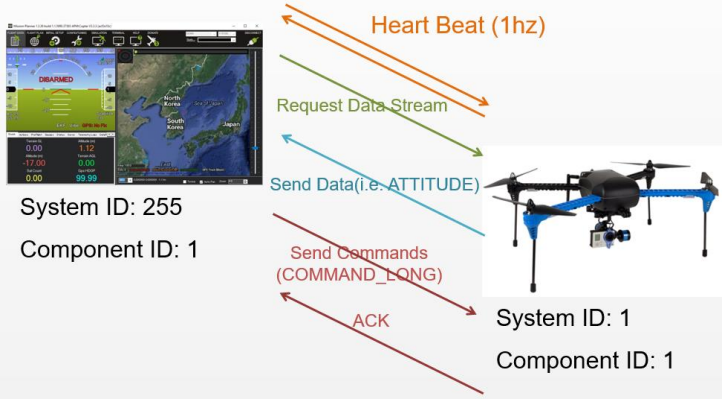
Byte Index	Content	Value	Explanation
0	Packet Start Sign (STX)	0xFE	Indicates start of a new packet
1	Payload Length (LEN)	0-255	Indicates length of the following payload
2	Packet sequence (SEQ)	0-255	Packet transfer sequence information for detecting packet loss
3	System ID (SYS)	1-255	ID of the sending system; Allows to identify multiple platforms on the same network
4	Component ID (COMP)	0-255	ID of the sending component; Allows to identify multiple components on the same platform
5	Message ID (MSG)	0-255	ID of the message; Define what payload means, and how to decode it
6 to (n+6)	Data (Payload)	0-255 (bytes)	Data of message; depends on the message ID
(n+7) to (n+8)	Checksum (CKA and CKB)	ITU X.25/SAE AS-4 hash of bytes 1 to (n+6); It includes MAVLINK_CRC_EXTRA parameter computed from message fields	

What we are interested in is:

- System ID (known as PM source): this source (ie Mission Planner) P Amy to BAn APM is sent from the USB flash drive through the wireless port of the APM. He checks me to know that this message is for him
- Component ID (known as subsystem within the system): every subsystem in the system Mainly, there is currently no subsystem and it is not really used
- Message ID: This specifies what the Payload database means, and how It can be decoded

- Payload: The message data depends on the message ID.

The current byte (from the APM packet that contains the number of bytes (i.e. 17) "msg" air), they send it to the hardware (for example, throughUART telemetry) and decodes the message in the software][37]



Shape2-33- How to exchange information between UAV and Mission Planner by MAVLink messages*][39]*

3-1-3 MAVLink message format

At***Shape3****32*-A red arrow fromThere is a Mission Planner type for the drone that sends commands to the quadrotor this message[38] We will deal with this type of message, COMMAND_LONG, MAVLink

Shape3-3- Detailed table of COMMAND_LONG message*][40 [*

Field Name	Type	Values	Description
target_system	uint8_t		System which should execute the command
target_component	uint8_t		Component which should execute the command, 0 for all components
command	uint16_t	MAV_CMD	Command ID (of command to send).
confirmation	uint8_t		0: First transmission of this command. 1-255: Confirmation transmissions (e.g. for kill command)
param1	float		Parameter 1 (for the specific command).
param2	float		Parameter 2 (for the specific command).
param3	float		Parameter 3 (for the specific command).
param4	float		Parameter 4 (for the specific command).
param5	float		Parameter 5 (for the specific command).
param6	float		Parameter 6 (for the specific command).
param7	float		Parameter 7 (for the specific command).

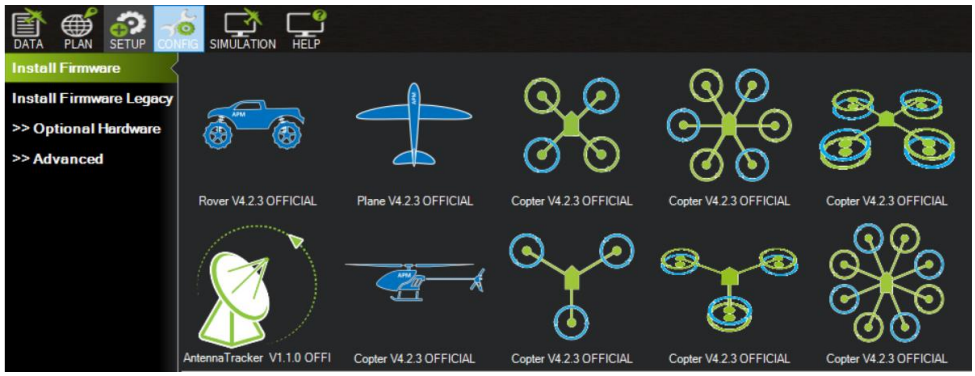
- The target_system part means that the IP address of the system domain is sent (if the system domain is execute the message) usually these types of messages are sent from the ground station to the drone
- The target_component part means which component of this system executes the message
- The command part is the most important part in this type of PM. It is the part responsible for setting the command level. For example, if it is sent by droneThe dial tab will be TAKE_OFF. The tab will be dialed. If the number is 22, it means that the drone is in LAND mode. The number is 21, it is the command mode. numbers in the section)which is available in [40] specifying MAVLink Commands (MAV_CMD They are almost There are 60 types of commands
- If the confirmation part is zero, it means that the PM will be sent for the first time, and if it is not zero, Yes, it means that the message has been sent correctly
- The last seven parts depending on the type of command used are in the third part, for example, if the command used is LAND, that means the whole seven is slower than WiFi. If it is TAKE_OFF, only the most recent level of altitude determination is useful. Is

3-2**Communication with the ground station**

As said before,It has been selected as the primary ground station by the Mission Planner

1-2-3 Installing firmware or operating system

After connecting the boardIn this section, the desired robot frame must be selected and the firmware installed (Figure 4-3), SETUP and go to the Mission Planner option by opening USB to laptop using APM serial



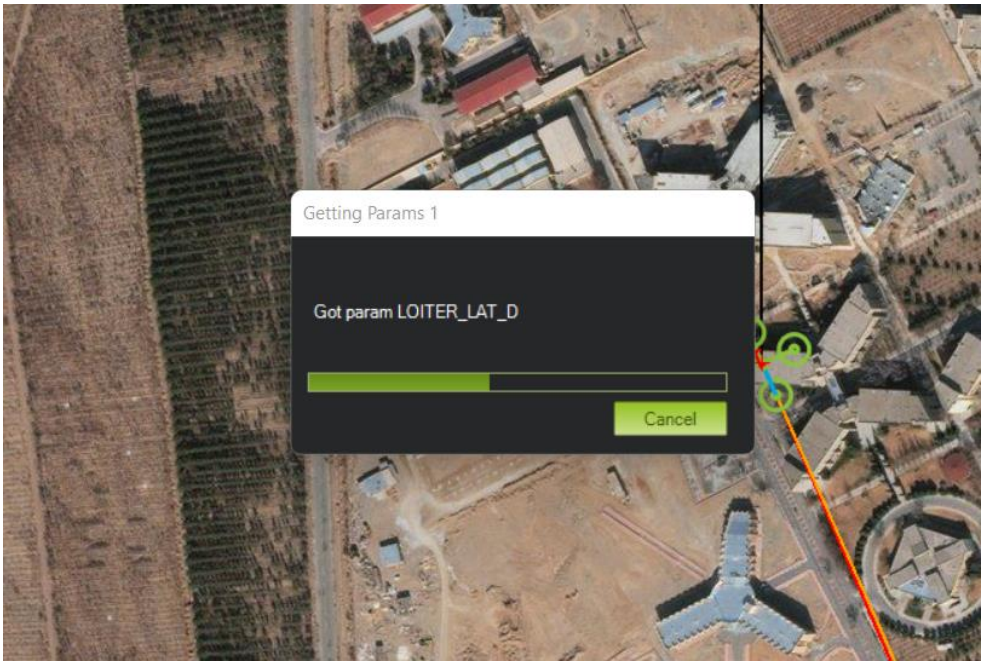
Shape4-3 Mission Planner firmware installation page

Pay attention to thatArdupilot offers the latest versions of the Mission Planner operating system If the hardware used is not compatible with the latest version of the firmware,Mission Planner will do this The latest version of the software that is compatible with the flight controller can be downloaded and installed. Ardopil's website is part of itfirmware download refer to][28]

2-2-3 connection with USB cable

After installing the firmware, using the cableOn the Mission Planner, connect the board to my laptop and plug in the micro USB Connect button (or the sameThe data transfer rate is 115200, that is, 115200 bits are transmitted, the baud rate is 115200 while connecting with the cable, the baud rate is clicked (Figure 3-5) CONNECT It can be done after pressing the button(Figure 63-) Mission Planner flight controller parameters are transferred to CONNECT

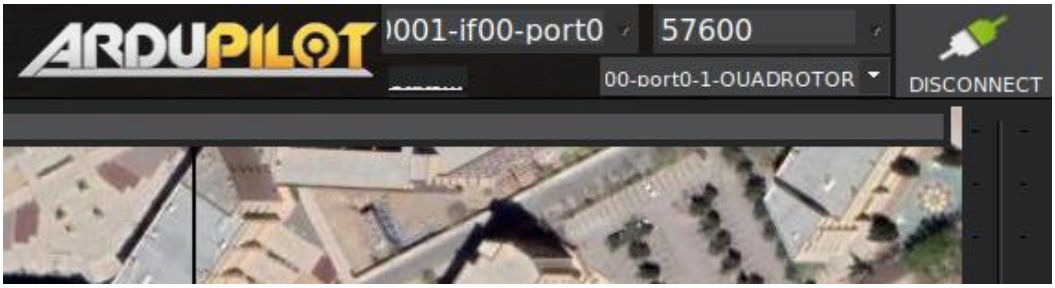
Shape5-3 CONNECT button available at the top of the Mission Planner page



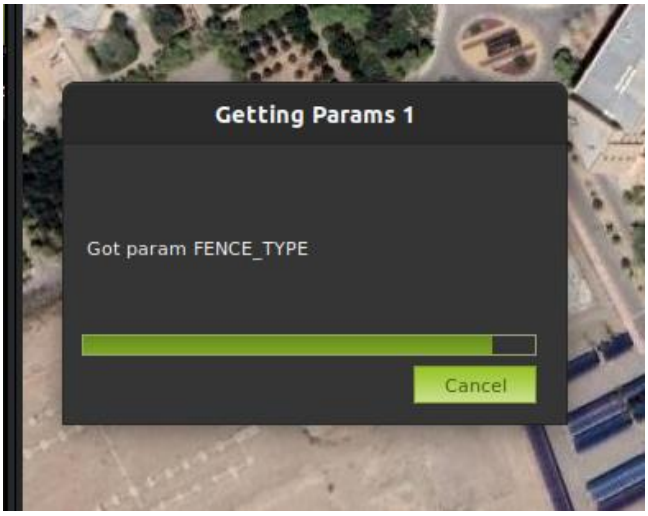
Shape6-3- The connection stage with the quadrotor (receiving parameters)

3-2-3 Communication with telemetry

The ground telemetry module is connected to the laptop and after a short period of time, the lights of the telemetry transmitter and receiver will turn off and do not flash (this indicates the establishment of communication between them), then the connection button inIt is called Mission Planner But here it should be noted that the wind rate (It should be 57,600, not 115,200. This is in the baud rate section. The tone of the parameters changed][41] [42] (Figure 7-3 and Figure 8-3)



Shape7-3 Connect button in Mission Planner software



ShapeConnecting with the quadrotor via Mission Planner 8-3 telemetry

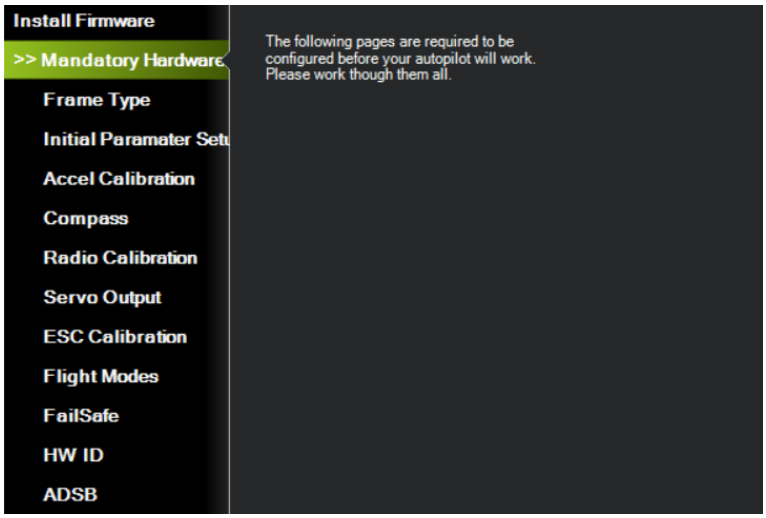
4-2-3 Calibration

Using serialIt is explained, select Mission Planner calibration and as you go in Mandatory Hardware, then connect to setup and then to Mission Planner option or telehmtry, to usb starts (Fig3-9)

First, the type of frame is selected, then the age and compass, and thenare calibrated, these are the main ESC and Radio There are parts that need to be calibrated. Note that all calibrations must be done outside the building

and during calibrationDisconnected to calibrate the radio to a USBs, must, for safety, propellers or ESC connectors senderESC datasheet is needed to calibrate RC[31]It should be read carefully and done exactly as the manufacturer says

Calibrating parts is easy and straightforward, but more and more detailed information for compass calibration]43 and Sheta Sen [44] and radio [45] and speed controller [46] are available



Shape9-3 page that includes calibration of various parts in Mission Planner

3-3**Starting the engines using Mission Planner and RC**

to arm the engines (It has been used for testing purposes, RC motors first from the transmitter (Motor Arming). are lit inside the laboratory. To do this, the safety check parameter (in Safety Check Parameter Parameters section inDisable since the drone is programmed not to fly Mission Planner Unless all sensorsThey work 100% and the drone must be outside to avoid damage

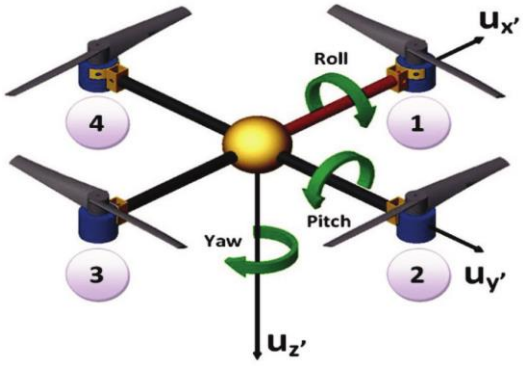
1-3-3 flight modes (Flight Modes**)**

Since the flight modes are related to starting the engines, first we explain some main flight modes firmware It offers options called flight modes to help users choose the best option for their ArduCopter mission Self-Select To test and get started, there is a recommended flight mode pen to use, and we'll explain this mode pen

---

Datasheet[31]

stabilization mode (It self-aligns (Figure 10-3) pitch and roll axis :) Stabilize



Shape10-3 The three main axes of a quadrotor: pitch, roll and yaw*][47]*

altitude hold mode (It self-levels the pitch and roll and keeps the height :) Alt Hold

wandering state (It uses GPS for movements, maintains the height and position, from Loiter :)

return to origin mode (Return above the take-off location, may also include landing :) Return to Launch or RTL be

automatic mode (It executes a pre-defined mission :) Auto

Different flight modes in this Marjad]48[available in detail

2-3-3 start the engines using RC

Very briefly: after calibrating the radio andmove to the right for half a row, turn the rudder down and turn on the throttle and RCs, and after connecting the drone to the battery, you only need to ESC

In general:

Turn on your transmitter, batteryConnect the LiPo to the board. The red and blue lights should blink for a few seconds because the gyroscope is calibrated (do not move the quadrotor) pre-arming checks) Pre-arming Checks It is executed automatically and if any problem is observed,Flashing yellow and ground station failure LED is displayed. Check that the flight mode switch is onPosHold or Loiter, AltHold, ACRO, Stabilize is dimmed (flight modes are discussed in the previous section) If you are using an autopilot with a safety switch, press it until the light goes off If you intend to use an auto mode (such asAuto, RTL, Loiter etc.), you have to quadruple towill turn green to change the LED indicator and keep the monitor until PosHold or Loiter lockArm the engines to the right of the rudder and hold the throttle for 5 seconds while keeping the GPS down.

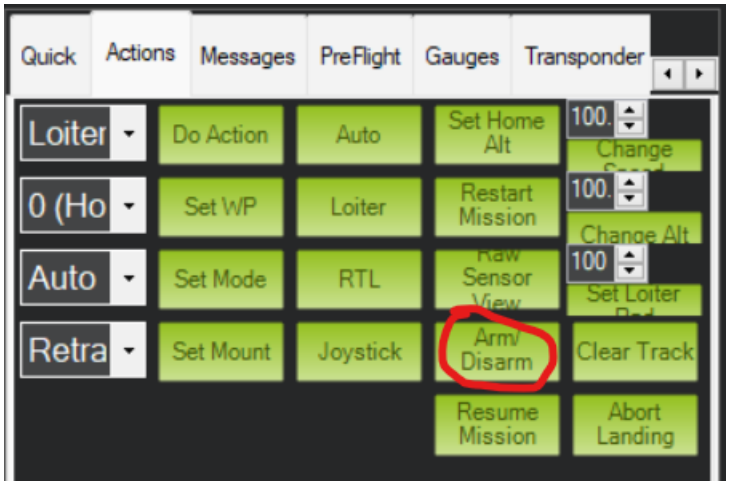
Move (Fig*311*-) MeaningTake a long Rudder in Figure 3-12 is clear Throttle and Rudder (more than15 seconds) Don't hold it right or you will start the AutoTrim feature Once armed, the LEDs will solidify and the propellers will start spinning Raise the throttle to fly [49]

Shape11-3- Motor arming mode in RC

Shape12-3 Division of Aileron, Throttle, Rudder and Elevator in a RC*]50 [*

3-3-3 Turn on the engines using the Mission Planner

As it was said in the previous parts, the dronePress the attached Arm and then just the Mission Planner button and the engines are turned on (Fig3-13) The battery must be connected



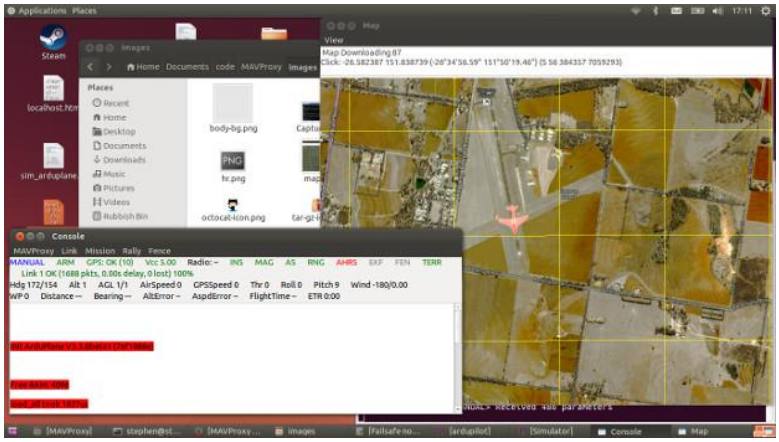
Shape13-3 Arm/Disarm section in Mission Planner to turn engines on and off

4-3**simulation withGAZEBO and MAVProxy**

1-4-3MAVProxy**What is**

UAV ground station software package for systems based onWith the function of GCS is a MAVProxy MAVLink Perfect for UAVs as a portable and expandable ground station for any independent system that uses the protocol It is designed to be used by ArduPilot as a system that supports MAVLink expanded through add-on modules or with another ground station such asetc. completed to provide a graphical user interface [31] (Figure 143-) QGroundControl, APM Planner 2, Mission Planner

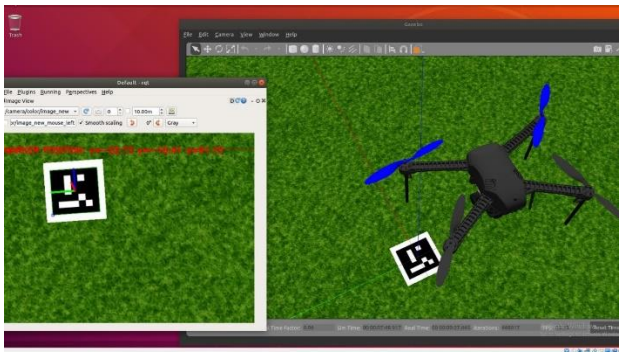
New (Builds) is used to test builds (especially with SITL) usually by MAVProxy developers. So to freeze, this is a ground station with simulationIt is used so that users can SITL programs test yourself in this event][31] The features of this ground station are available



Shape14-3- MAVProxy ground station in Linux operating system*]31[*

3-4-2 What is GAZEBO Simulator?

A 3D dynamic simulator capable of accurate and efficient simulation of robot populations in Gazebo's complex indoor environments and is external (Figis the leader in robot simulation [51] This is a tool that hundreds of thousands of users and Gazebo (153) Trusted by developers worldwide.[52]



Shape15-3- Quadrotor simulation in GAZEBO simulator

3-4-3 tons of SITL environment and GAZEBO installation

To start the simulation, you must firstInstall GAZEBO and MAVProxy on MAVProxy Win dose and also on linux edgeOnly on the edge of the line (Gazebo road). It can be run on Windows, but it is better to install it on Linux. Refer to Appendix 1: setting up the gazebo, Gazebo and SITL installation

SITLAnd its implementation and attachment2: Installation**GAZEBO**and its implementation

5-3**Summary and conclusion**

For the general clause, in this chapter, regarding the protocolQuadrut Ver connection to ground station, Calibras Ion, MAVLink Its sensors, its connection toAnd quadruple simulation is done in different software like RC action. Everything in the drone is ready for connectionCompleted above, ROS-ready drone Connecting to the robot's operating system will be discussed in the next chapter

1-4**robot operating system () ROS**

What is ROS 1-1-4

Robots generally have a perceptual unit (are in this mode so that the robot correctly (Actuator) and the operator unit (Sensor). To be able to perform its mission, it is necessary to write codes for each of the foot units and to establish a connection between these codes.A powerful and smooth platform for the integration of related codes (ROS). Provides assistance to each of the robot unitsUnder Linux, the code corresponding to each of the ROS foot units can be used in a node ([To be created] (53) wrote (Topic) and the communication between these nodes through messages called (Node)

Around 2007, the first version of the robot operating system at Stanford University[32]It was created since thenSpread of ROS It has found a large statistical community of programmers in the field of developmentIn fact, they have a ROS activity Not an operating system, but a framework (It provides for robot programming by having Framework standards Explained, the members of a team can work at the same time to program a robot with any of the Python languages[33]Or

+ [Coded] 53 ROS can be used for C

2-1-4 Why do we use the robot operating system?

There are many reasons why engineers and developers use itROS for your robotics projects Prefer that some of these reasons are mentioned below:

Stanford University[32]
Python[33]

A. The same code can be applied to many different types of robots: robotic arms, drones, mobile bases. Once learned about how to communicate between all the nodes of the program, it is easy to set up new parts if it works. different robots, you can easily go between these robots][54]

- usingWrote and communicated between them C++ can be programs using Python and ROS

R-This means that you can have many independent bots that ROS Master can work with multiple ROS Each system[They have their own and all robots can communicate with each other] 54 ROS

D- The main baseIt doesn't take up much space and you can quickly install the main packages and ROS in a few minutes In addition, it can be started fromRaspberry Pi as embedded boards in ROS embedded computers also used (this feature is used in this project) so you can easily start a new project without much trouble][54]

e-It allows developers to ROS their bot before implementing any code or application in the real world Easily simulate in any environment, tools likeThey even make it possible with the robots provided by Gazebo Do not simulate

3-1-4 ROS architecture

It is not possible to enter into the details of the robot's operating system because it is a world itself, but for the following statements, this terminology may be needed, but if the reader has previous information aboutROS does not have, better, for future parts, ROS architecture to know

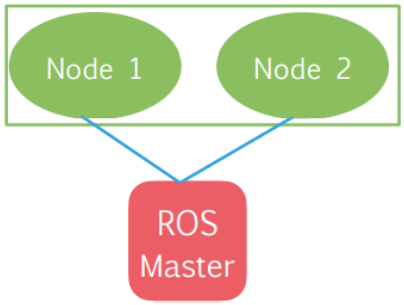
First of all, we should say thatIt has three basic roles in control: it feels, thinks and acts. The ROS sense part It can be a camera, laser sensor, or ultrasonic sensorGPS[34]The thinking part may be artificial intelligence, machine learning or signal processing. The action part can be various stimuli for freezing, a well-known example is a self-driving car, it senses its surroundings, it thinks what to do next. gives and performs actions using stimuli. Self-driving cars usingROS are moving

Scripts or codesNodes ROS[35]are called the connection between these nodes through nodes calledPerform ROS master becomes (Fig14-) It is also called ROS core if the main node[36]If it is broken or stopped, the whole system will fail

---

Global Positioning System[34]
Node[35]
Ros master[36]

This is a weakness inROS is that the whole system is dependent on the on or off of a node. This is a problem in ROS 2.0 went away

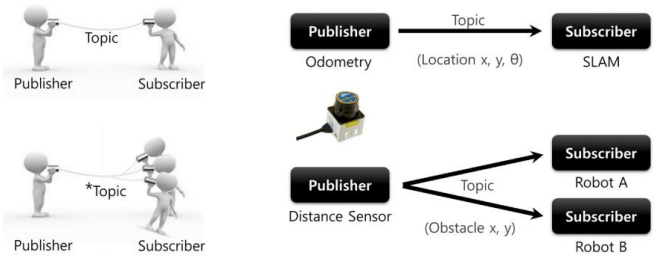


Shape1-4- The connection of two nodes with the main node*][55]*

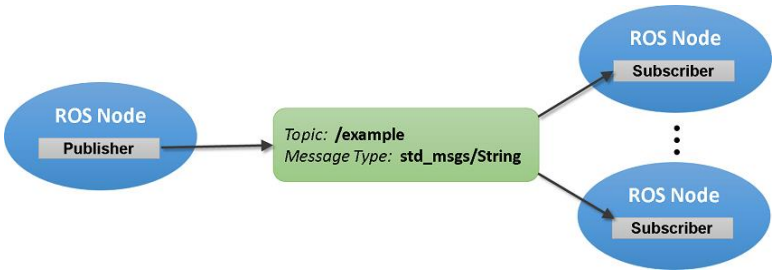
Each node can be a publisher[37]Don't leave Turkey[38]The data publisher node sends the information to the publisher, and the network node receives the data through something called topics.[39]The connection between the publisher and the Turkish network is completed, which means that only the publisher can send the information and the publisher's node can receive the data. Each of the Turkmen mesh group that participates in the topic should send the type of data that they want to send, specifically rejecting the information or data sent by the publisher. It's going to be BSPM[40]It is known that the type of PM should be specified before sending the data, for example, the type of PM sent by the laser sensor.It is because it is calculating the distance of objects in the environment (Figure 4-2) float32

There are virtual SIM topics that send data with a special PM. In order to allow this communication between nodes, the names of the topics must be the same in both nodes (Fig.*43-*)

Publisher[37]
Subscriber[38]
Topics[39]
Message[40]



Shape2-4 publisher and subscriber communication in ROS*][55]*



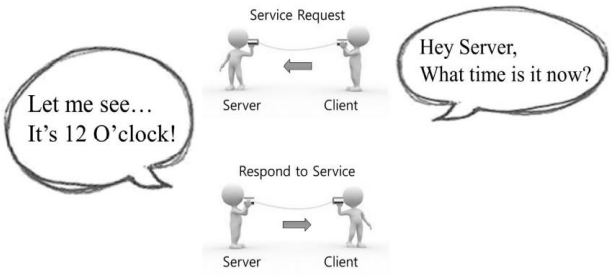
Shape3-4- Publisher and subscriber communication by topic and determining topic name and message type*][56]*

Nodes can also server[41]or the client[42]As a publisher sends data to a subscriber, a server also sends something to a client.

virtual sim this time service[43]It is called and abbreviatedCalled Like a thread, a service message must specify srv The client node (or the client) requests a service from the server node and the monitor waits for the response. The client will stay monitored until the server sends its response (Fig.4-4)

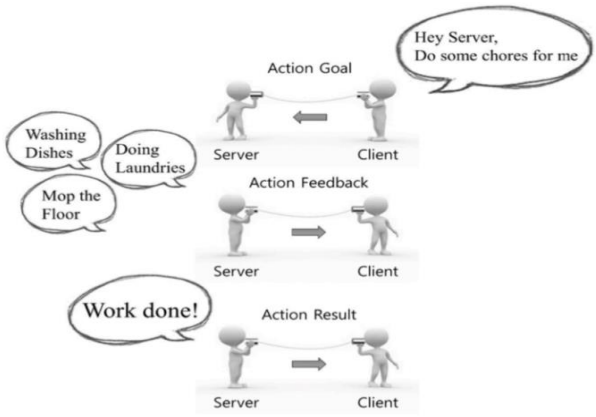
For example, a service(a service that sends the robot to a certain point) is created (goToPoint If the robot has reached the desired point, the service will respond to us at the end (unlike actions, which we will talk about in the next section).

Server[41]
Client[42]
Service[43]



Shape4-4 server and client communication (with services)*][55]*

Servers and clients (clients) are also possible through something called actions[44](Actions) talk to each other is like services (Services), but with one difference, that the actions (Actions) are asynchronous, also the type of messages related to the actions (Actions) must be specified, for example, if an action "Let's do X, this action in every "goToPoint Mileage will send you how many meters are left to reach the desired point. The services do not have such a feature (Fig.4-5)



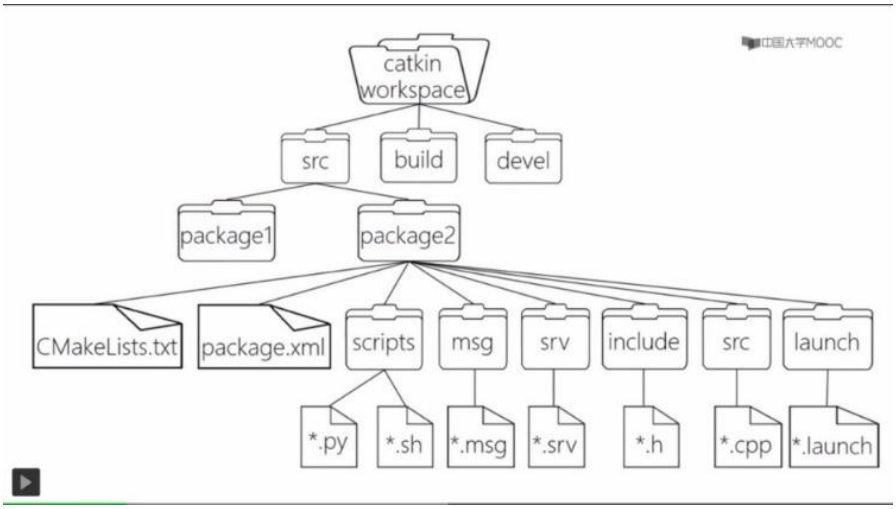
Shape4-5 Communication between server and client (with actions)*][55]*

4-1-4 What are the workspace and ROS package

SpacecraftThe ROS package stack is simply the stack wrappers of the workspace packages. There are folders that contain programs and scripts.src be created so that all the packages can be placed in it, then in the package folder, scripts ++There is also more detailed information available in this article [57] (Figure 6-4) msg and srv or launch are placed in another file such as scripts and Python scripts in the src folder. Roger in a C folder

---

Actions[44]



Shape6-4 ROS workspace structure*][58]*

2-4**What is Raspberry Pi?**

In this project and as it was mentioned before, he won(Raspberry Pi 3BShape7-4) for downloadMAVROS and ROS

It has been used on this Raspberry Pi computer board project. Raspberry Pi is a small computer that has been used since2006 in

It is being developed and its parts are mounted on a motherboard the size of a bank cardIt runs a Raspbian

It is a special version of the Linux operating system that is specially designed for this computer. FromUbuntu mate

16.04 is used as the operating system of this computer because installing ROS on Raspbian takes almost a whole day.

Kills and forCreating (or compiling) a workspace should take between two and eight hours every time, and maybe build

At this time, there will be a problem that the whole work will be done with Raspberry Pi for basic office computing applications, games.

The low level has access to the Internet and e-mail, video playback and many other functions that are normally available from a computer

In the 21st century, it is possible to install a Raspberry Pi with all these features with a very small number of parts, including a processor.

And ARM offers a very low price

Shape7-4 raspberry pi 3B board*][59]*

To install the operating system on the Raspberry Pi, follow the steps belowAll the steps of installation and preparation are needed in detail on the sd card this part]There are 60[of Marajd available and the installation of the robot operating system, on my laptop, follows this Marajd [61]. It is necessary to edit the operating system of the robot that was installedIt is EOL and this editing is kinetic[45]Or it is the end of life, which means it is no longer supported, but only with the old versionTo be installed and all this kinetic work, you need to edit MAVROS It is for the compatibility of the software with the flight controller

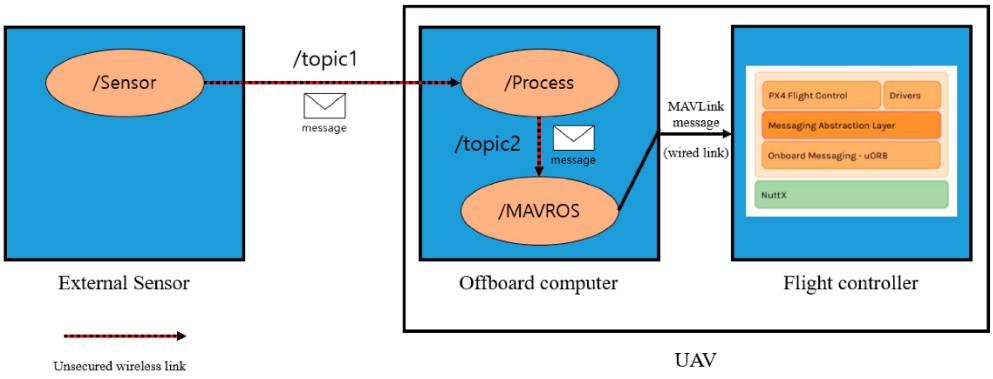
3-4**Closed****MAVROS**

1-3-4 MAVROS package definition

This communication driver package for various autopilots with communication protocolThis package is recognized by the Russian Mohandb (host MAVLink+ROS from its name) MAVLink and the ROS protocol provide a bridge between MAVROS, in addition MAVLink**Vladimir Ermakov**After completing his studies from the university Industrial Russia (MIREA[46]), has been working in the field of robotics and software development[and twelve percent is written in Python] 1 wrote C++ in 2014 Eighty-five percent of this package is in MAVROS The remaining three percent are miscellaneous files such asIt is shown by the threads that MAVROS is using the ROS package in Figure 84 - a brief explanation of the drone communication with xml or txt

End Of Life[45]

Moscow Institute of Radio, Electronics and Automation[46]



Shape8-4- Brief map of flight controller communication process with ROS and also with external sensor*][62]*

2-3-4 MAVROS package structure

PakiLike other packages, it consists of different nodes and topics that allow communication between these MAVROS nodes provides the main node that does almost all the workWe will pay attention to the mavros_node structure first

This thread mavlink/to seems to be the main communication node, which itself belongs to the thread called Mavros_node. dataArduPilot comes from MAVLink

Mavlink/from transmission is responsible for diagnostics and mavlink/from is the publisher of two threads called Mavros_node dataIt is responsible for diagnosing the connection status, etc., diagnostics to the flight controller and MAVLink

As previously mentionedThere are similar parameters: mavros_node, here, in the node, component id and system id have parameters such as MAVLink, the MAVLink package was discussed.

is MAVLink, an integer number is written in it, which is the system ID of the node: system_id -

MAVLink also takes an integer that is the ID of the node: component_id -

FCU (Flight Control System ID: target_system_id -[47])

ID of the flight controller: component_system_id -

It takes the string, a symbol of the flight controller's address, for example, if the flight controller is controlled by a remote control: fcu_url

- If it is connected to the computer, it will look like this:, in network science, and in Linux operating system ./dev/ttyUSB0:57600 in a folderttyUSB, you can see all the ports, in this case, the port to which Telhamtri is connected is named 0 dev It was determined and the wind rateIt is 57600

---

[47] Flight Control Unit

but it specifies v in this (either 1.0 or 2.0), the default is 2.0 MAVLink Edit: fcu_protocol - Project of editingUsed v1.0 due to flight controller incompatibility issue with version 2.0

determined the udp that the GCS connection should be connected to if it is placed on the ground station, the address should be this: gcs_url - For example, the manners can be like this: @udp connects, this means the ground station on port 14855 with protocol gcs_url:=udp://:14855

It is suggested to refer to the main article of this package for a deeper and better understanding of the structure of this package.][63]

With examples in the next sections, the idea will become clearer

3-3-4 MAVROS package installation

Compatible with Noetic and Melodic, Kinetic including ROS with all recent versions of MAVROS

packagesSometimes they need external libraries and tools that must be provided by the operating system, these libraries are ROS and the required tools usually as system dependencies (This dependency is related to the geographic coordinate units. The description of this dependency is not related to the main issue (install rospy, the dependency name is the same as if Python is used in the robot operating system, GeographicLib, MAVROS, some of these dependencies are needed for the system dependency, the most important MAVROS are known to install (System Dependencies). There is more information in this Marjad]64 [There is, however, it is possible to install the package before installing the packages and then Dependencies are installed.

For installation stepsAppendix 3: Installing the package, raspberry pi on laptop and on MAVROS**MAVROS**On the laptop: and the attachment4: Install the package**MAVROS**Face**raspberry pi**refer to

4-4**Connect toon SITL leptam in ROS**

As mentioned earlier, it is always better to test in a simulation before going into the real worldSITL It's the easiest way to test because there's no hardware, so programsIf you have any ROS

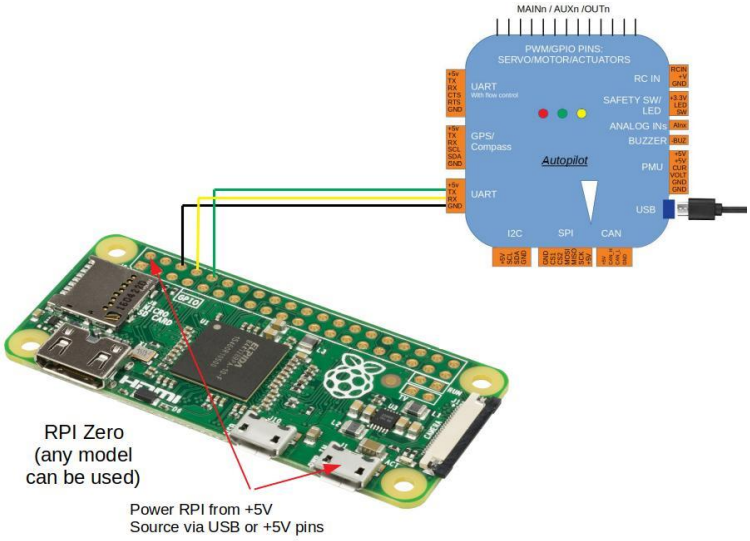
The problem can be easily checked by the connection stepsIn Appendix 5: Link to ROS**ROS**At**SITL**Available on laptop

Is

5-4**Real connection withROS**

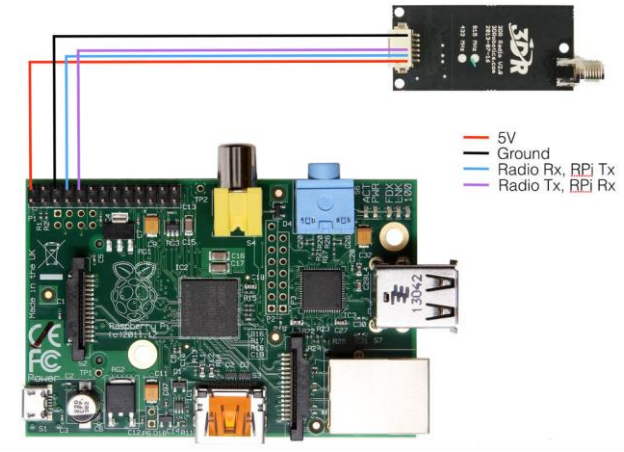
till here,which runs ubuntu 16.04, are installed. Now several raspberry pi's should be on MAVROS and ROS There are ways that we can connect the flight controller to the Raspberry Pi, three of which we call the use of cables. In the MissionPlanner or telemetry section (as before to connect the flight controller with usb**Communication with Kabuland USB Communication with telemetry**said) and through the serial port connection (Connection to Raspberry Pi is possible only for Serial Port usingDid not do ROS, many things can be done using Raspberry Pi other than ROS

1-5-4 communication with Serial Port



Shape9-4- The process of connecting the flight controller with raspberry pi using serial port*][65]*

As in the whole4-9- In particular, it should be in the UART section or anywhere in the TELEMETRY connection. Control the flightRaspberry Pi Ground connection to the UART Male Ground connection to the raspberry pi ten,Changed (Figure 104-) USB router can be connected by connecting +5V to +5V pin through RX to TX and TX to RX

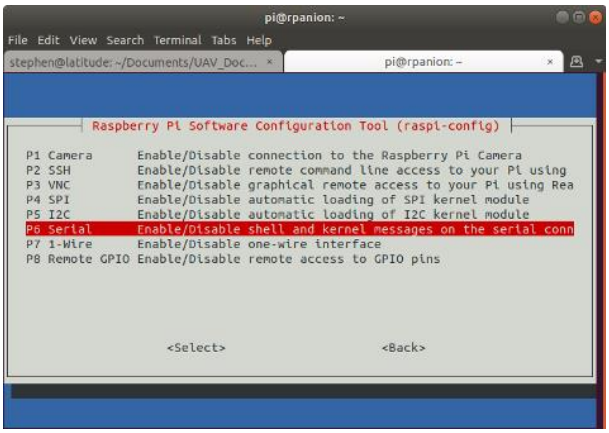


Shape10-4- connecting raspberry pi 3b with TELEMETRY ports of the flight controller*][66]*

Then only in the terminalThis command is executed on raspberry pi

config-sudo raspi

A new page will appear on the optionIt will be activated (Figure 4-11), Serial is clicked and then Interfacing Options



Shape11-4 option to enable and disable Serial*][65]*

With this connection, there is now a connection between the flight controller and the Raspberry Pi and you canConnected ROS

2-5-4 connection with usb cable or telemetry

To connect with KabulThe USB cable connected to the Raspberry Pi is enough for the flight controller and the micro side is the USB side You will be thrown awayOpen ttyACM0

Now using this command, to: Connects to ROS

```
roslaunch mavros apm.launch fcu_url:=/dev/ttyACM0: 115200 fcu_protocol: -
                                                                    v1.0
```

Available is mavros which is in the apm.launch package to start the roslaunch mavros apm.launch code

Fcu_url to determine the address of the flight controller connected by the cable

115,200 to determine the wind rate

Used version 1.0 not 2.0 used because 2.0 MAVLink to determine the revision Fcu_protocol:-v1.0 It does not work on our hardware

Now you can execute the various codes that we wrote or the commands that came and it should search Quadrotor

To communicate with telemetry, the telemetry ground module must be connected to the Raspberry Pi. With this connection, communication must be established between the two modules, now the portRaspberry Pi is activated ttyUSB0

Now using this command, to: Connects to ROS

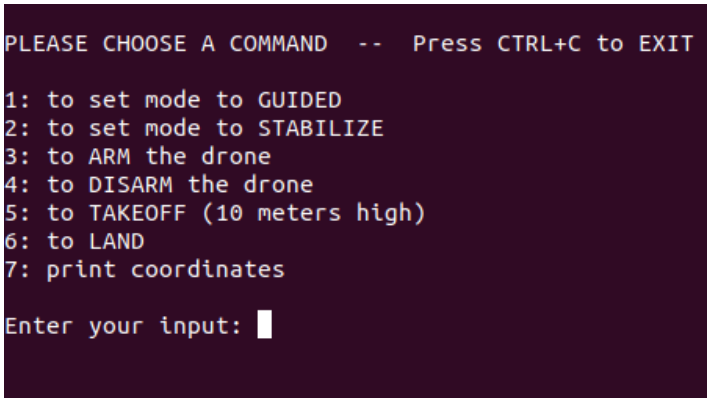
```
mavros apm.launch fcu_url:=/dev/ttyUSB0: 57600 fcu_protocol: -v1.0    roslaunch
```

Only the port and wind rate are changed.

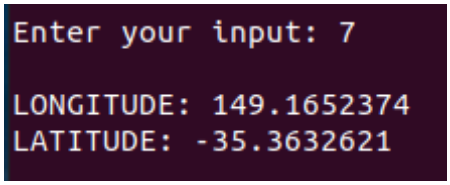
6-4**Run the program**ROS

Now, what is the relationship?It happened, any programs can be written using Python or ROS. Writing and executing the programFor quadruple control in connection 6: download and rent ROS **ROS**Written

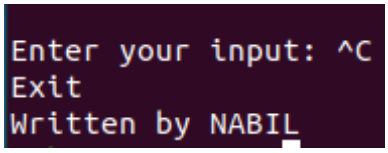
By executing the command`run apm_control apm_control.py` ros(Of course, after communicating with**ROS**) : (Shape124-And Shape134-AndShape144-)



Shape12-4- The options given by running the apm_control.py program



Shape13-4- The behavior of the apm_control.py program by entering the number 7



Shape14-4- The behavior of the apm_control.py program by writing C^

7-4**Extras: Linking Leptom to Raspberry Pi by**SSH

from the protocol)It is used to create a secure connection between the user and the server in this protocol using SSH (Secure Shell From the public key and symmetric encryption, all the content sent between the user and the server is encrypted, and only two parties who have a common agreed key can access the original content.

If laptop and Raspberry Pi are on the same internet network, this connection can be established firstssh on both machines Install:

```
sudo apt-get install ssh
```

Next is moodssh restartbe executed

In the Leptam terminalRaspberry Pi is also written as ipaddress and raspberrypi_username is executed, of course, the real name is instead of ssh raspberrypi_username@ipaddress

Now all the commands that the user wants to apply are applied from my laptop

Chapter Five: Results

Drones are amazing tools that make engineers always eager to see the future of this industry. Of course, the rapid and amazing development of technology surprises us every day with new technological devices that can think and learn. Of course, I am talking about the world of artificial intelligence. For this project, the first step to this world has been done in the field of drones, but there are many other steps. Here, we were able to control a drone using the robot operating system, which today seems to be the best and easiest platform for robot programming in this project. With We have worked with which many special upgrades can be added to the drone, the code is written that gives us Raspberry Pi The ability to control the drone fromGives is simple, many other options can be added and the drone may be ROS perform much more complex actions, but since the goal of the project was achieved and the concept was understood that allows us to do more, it means that it is time to think about other ways to upgrade the drone in the field of artificial intelligence.

We may think, what will happen next? There are many ideas that can be studied and implemented, for example, implementing algorithms.SLAM[48]On flying vehicles it may be a challenge, but this is only one world. Another idea could be controlling and flying drones inside buildings. Some ideas may be in the field of agriculture, some of them may be in the field of firefighting and rescue. He does not have time to rest in this field

---

[48] Simultaneous Localization And Mapping

the attachment

the attachment1: Setting up the environment for SITL and its implementation

It works only on GAZEBO Linux, both on Windows and Edge on Linux, but MAVProxy It runs (there is a way to run it on Windows, but it is better to install it on Linux) Linux is used for it.

To set up the Linux environment, the following steps are followed:[67]

Download Gate (:) git

```
sudo apt-get update -
sudo apt-get install git -
sudo apt-get install gitk git-gui -
```

Then the tank: ArduPilot is cloned

```
ardupilot.git/ArduPilot/com.github//:git clone https
cd ardupilot
recursive--init--git submodule update
```

And now the system is ready to run the simulationIt has been SITL in MAVProxy

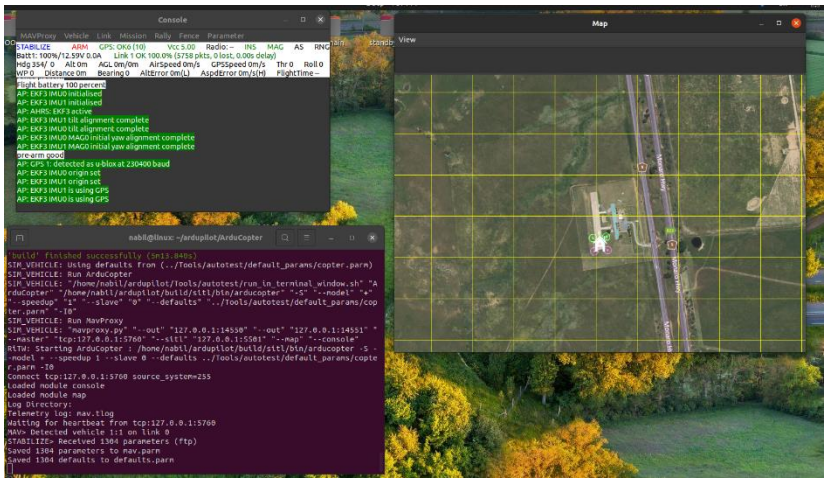
to launchArduCopter: We go to the SITL folder

```
ArduCopter/cd ardupilot
```

Then this command to run the simulation file: We run SITL

```
w-py.sim_vehicle
```

something like the figure16-3 comes



Shape1-0- Upper left side: console and lower side: terminal and right side: map

For testing, some commands are given to the simulation, for example:

stabilize to determine the mode: stabilize mode

To turn on the engines: arm throttle

To fly 10 meters: takeoff 10

To turn off the engines: disarm throttle

To change the mode to circle mode (the quad flies in a circle): mode circle

To convert to landing mode: land mode

The complete list of commands is here]68[is available

the attachment2: Installing GAZEBO and running it

to install[On Linux, these steps should be followed:]69 gazebo

These commands inIt is written "stable.list- file gazebo-" are written in "echo". The first command means that the string after Terminal

'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu-stable lsb_release -cs`c-sudo sh
main" > /etc/apt/sources.list.d/gazebo-stable.list'

orderTo install and unlock the key, w-get is required

- key add-sudo apt| -O-key.gazebo/org.osrfoundation.packages//:wget http

sudo apt update

InstallationNinth edition of Gazebo

dev-sudo apt install gazebo9 libgazebo9

Copy the foldergazebo responsible for simulation in ardupilot

ardupilot_gazebo/khancyr/com.github//git clone https:

Enter the folderardupilot_gazebo

cd ardupilot_gazebo

Create a folder calledMaking the build package and entering it for build

mkdir build
cd build

orderbuild in the cmake folder to build cmake files

.. cmake

orderBuild for make

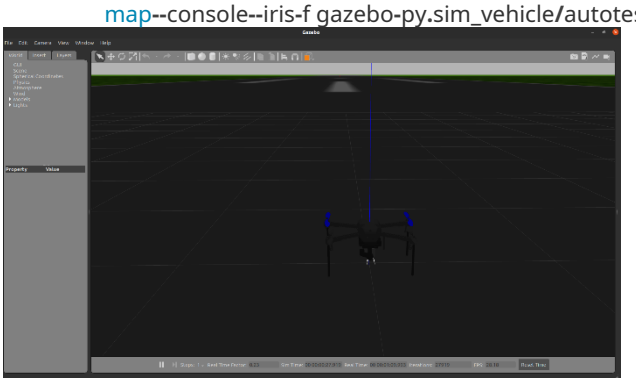
j4-make

"make install" copies only the compiled files (in the previous step) to the appropriate locations.

sudo make install

To beginSITL with gazebo simulator

world.iris_arducopter_runway/verbose worlds--gazebo
ArduCopter/ardupilot/~cd
map--console--iris-f gazebo-py.sim_vehicle/autotest/Tools/..



Shape2-0 GAZEBO simulator with a virtual ArduCopter in it

The commands mentioned in Marjad][68] It is also used for GAZEBO

the attachment3: Installing the MAVROS package on my laptop:

First, it is installed on my laptop. After installing the robot operating system on your computer, these steps should be followed:

```
ros-kinetic-mavros ros-kinetic-mavros-extras                    sudo apt-get install
```

This command is executed in the terminal if editingKinetic should be installed instead of Kinetic, other than ROS You wrote your own edit, this closed commandInstalls [70] mavros-extras and the mavros package

```
                                                                                                                Wget
https://raw.githubusercontent.com/mavlink/mavros/master/mavros/scripts/install_geogra
                                                                                                    phiclib_datasets.sh

                                                                                                ./install_geographiclib_datasets.sh
```

These two commandsThey install the dependency

Steps to die easilyOr they install Binary Installation, the installation method followed is called MAVROS For more detailed and clear information about this part, please visit Marajed]Follow this installation procedure for your laptop because it is the latest editionWe decided to install it, we have to use a method that allows us to choose a desired edition, it installs the raspberry pi and there is no problem with incompatibility, but when on MAVROS Let's call this methodOr the installation is from source installation

the attachment4: Install MAVROS package on raspberry pi

(It is recommended to edit the Linux operating systemis) kinetic stage and UBUNTU 16.04 robot operating system First installThese two are python-rosinstall-generator and python-catkin-tools respectively MaslavlIt contains rosinstall information and the second files of catkin tools to make the workspace by build Tanks with packagesProduces ROS

```
install python-catkin-tools python-rosinstall-generator –y                    sudo apt-get i
```

A workspace is created to install the packages:

```
                                                                                mkdir -p ~/catkin_ws/src
                                                                                        cd ~/catkin_ws
                                                                                            catkin init
                                                                                    wstool init src
```

In the installation section on my laptop, closedInstalling MAVROS was not installed by command but automatically with MAVLink It should be hereMAVLink is installed from the ROS protocol. This is only the version of MAVLink

```
ll_generator --rosdistro kinetic mavlink | tee /tmp/mavros.rosinstall                    rosinsta
```

Installation: MAVROS

```
ll_generator --upstream mavros --deps | tee -a /tmp/mavros.rosinstall                    rosinsta
```

closed withdrawal: In the MAVROS workspace and adding dependencies

```
                                wstool merge -t src /tmp/mavros.rosinstall
                                                    wstool update -t   src -j4
                            rosdep install --from-paths src --ignore-src –y
```

Install dependencies:geographiclib

```
                                pts/install_geographiclib_datasets.sh              ./src/mavros/mavros/scri
```

Build workspace:

```
                                                                                    catkin build
```

Of course, you must fileMAVROS can be found and rosrun can be closed. This source command is available in the devel workspace in the setup.bash folder.

```
                                                                    source devel/setup.bash
```

Now that we want the flight controller (connect, no communication is established after MAVROS with APM2.8 Several hours and days of troubleshooting, Mr. Ermakov found a way to change versionsMAVROS and MAVLink compatibility With Presented APM2.8

to folderWe run to 0.30.0, we enter mavros which is available in the workspace and use this command to change the editing of mavros.

```
                                                        git checkout v0.30.0 -b release/0.30.0
```

And also forEnter its folder and execute this command mavlink

```
                                                release/2019.7.7    git checkout upstream/2019.7.7 -br
```

the attachment5: binding to ROS in SITL on leptam

First, a habitThis is to always work in a workspace folder, so you need to create one in ROS To be: In a terminal, and the steps of creating a workspace, these commands are executed:

```
mkdir -p ardupilot_ws/src
cd ardupilot_ws
catkin init
cd src
```

nowAs before in the SITL section, SITL is ready to work. Now you need a ROS emulator**simulation with GAZEBO and MAVProxy**It was said, it will be implemented):

```
cd. .
cd ardupilot/ArduCopter
mapsim_vehicle.py -v ArduCopter --console - -
```

Now an exampleis set up, there should be a line like below UDP and TCP will be in the terminal with SITL access have:

```
vproxy.py" "--master" "tcp:127.0.0.1:5760" "--sitl" "127.0.0.1:5501" " - -                    "Ma
        0.0.1:14550" "--out" "127.0.0.1:14551" "--map" "--console"                    out" "127.
```

both "-Using mavros with UDP refers to MAVProxy access created by UDP to the "out" connection We will go to the terminal:) it is ~/ardupilot_ws/src in the working folder, create a new folder for the ROS file

```
mkdir launch
cd launch
```

filePishfar launch : We copy ArduPilot for MAVROS

```
roscp mavros apm.launch apm.launch
```

There should be a new file named "Just have help calling "roscp" in your folder command "apm.launch Closed system command: or open any other editor and in it instead of the first line which is gedit, it is with ROS

```
the  <arg name="fcu_url" default="/dev/ttyACM0:57600" /> <!-- Port et baudrate of
                                                    connection with Pixhawk - - >
```

This line is found:

```
lt="udp://127.0.0.1: 14551@14555 " / >      <arg name="fcu_url" defau
```

Then this command is executed:

```
roslaunch apm.launch
```

If something like this happens, it means that everything is done:

INFO] [1496336768.500953284]: CON: Got HEARTBEAT, connected. FCU: ArduPilotMega / [ArduCopter 61724]: RC_CHANNELS message detected! [INFO] [1496336768.5367 0x0000000000001bcf FO] [1496336769.533950451]: VER: 1.1: Capabilities [IN 03060000 (8a4a2722) NFO] [1496336769.534021653]: VER: 1.1: Flight software: [I) NFO] [1496336769.534146986]: VER: 1.1: Middleware software: 00000000 ([I) (00000000 NFO] [1496336769.534195446]: VER: 1.1: OS software: [I 00000000 36769.534280663]: VER: 1.1: Board hardware: [INFO] [14963 0000:0000 336769.534309086]: VER: 1.1: VID/PID: [INFO] [1496 0000000000000000] [1496336769.534331512]: VER: 1.1: UID: [INFO 769.534370049]: CMD: Unexpected command 520, result 0 [WARN] [1496336 8.533962739]: FCU: APM:Copter V3.6-dev (8a4a2722) [INFO] [149633677 Frame: QUAD [INFO] [1496336778.534247677]: FCU: 4163]: PR: parameters list received [INFO] [1496336779.02113 P: mission received [INFO] [1496336783.535151119]: W

The first line is the most important line, it means that the flight controller has arrived, that is, it is connected, it can be seen that it was able to read the information of the flight controller, and now it is only a command monitor.

Information about implementationIn this part [72] is available SITL with ROS

For testing, you can use Python ++write and cut the result, you can directly use some of C Your instructions are emptyIt is available, use KRD part, write KDB Python for MAVROS part Finally, in this section, I have prepared commands for both methods (write a CD using ready-made commands), one for simulation and one for communication. At the end of the day, the instructions are here [1] are available in the Utility commands section

For example, to change the flight mode:

c 0-rosrun mavros mavsys mode

This order is ready to fly0 changes (that is, STABILIZE) the complete list in flight mode. Error! Reference source not found

```
STABILIZE =     0,  // manual airframe angle with manual throttle
ACRO =          1,  // manual body-frame angular rate with manual throttle
ALT_HOLD =      2,  // manual airframe angle with automatic throttle
AUTO =          3,  // fully automatic waypoint control using mission commands
GUIDED =        4,  // fully automatic fly to coordinate or fly at velocity/direction using GCS immediate commands
LOITER =        5,  // automatic horizontal acceleration with automatic throttle
RTL =           6,  // automatic return to launching point
CIRCLE =        7,  // automatic circular flight with automatic throttle
LAND =          9,  // automatic landing with horizontal position control
DRIFT =        11,  // semi-automous position, yaw and throttle control
SPORT =        13,  // manual earth-frame angular rate control with manual throttle
FLIP =         14,  // automatically flip the vehicle on the roll axis
AUTOTUNE =     15,  // automatically tune the vehicle's roll and pitch gains
POSHOLD =      16,  // automatic position hold with manual override, with automatic throttle
BRAKE =        17,  // full-brake using inertial/GPS system, no pilot input
THROW =        18,  // throw to launch mode using inertial/GPS system, no pilot input
AVOID_ADSB =   19,  // automatic avoidance of obstacles in the macro scale - e.g. full-sized aircraft
GUIDED_NOGPS = 20,  // guided mode but only accepts attitude and altitude
SMART_RTL =    21,  // SMART_RTL returns to home by retracing its steps
FLOWHOLD  =    22,  // FLOWHOLD holds position with optical flow without rangefinder
FOLLOW    =    23,  // follow attempts to follow another vehicle or ground station
ZIGZAG    =    24,  // ZIGZAG mode is able to fly in a zigzag manner with predefined point A and point B
SYSTEMID  =    25,  // System ID mode produces automated system identification signals in the controllers
AUTOROTATE =   26,  // Autonomous autorotation
```

Shape0-3- The complete list of flight mode numbers*][73]*

To arm the engines (start the engines):

```
rosrun mavros mavsafety arm
```

And to turn off the engines insteadIt is written disarm, arm

And the same can be done for different ordersstarted and saw the simulation in 3D GAZEBO (To execute In the GAZEBO section**.Error! Reference source not found**was explained)

the attachment6: Writing and running the ROS program

First, a workspace is created for this project, then in this workspace a package with the desired name (for example: : is created (apm_control

```
mkdir -p ~/proj/src
cd ~/proj/
catkin_make

cd ~/proj/src
_msgs rospy  catkin_create_package apm_control std

cd ~/proj/
catkin build
```

Now open a folder: Built-in scripts depending on the name of src

```
cd ~/proj/src
mkdir scripts
```

Now the file in which we want to write the code is created:

```
touch apm_control.py
```

And the code is written in it, and the work environment again(source setup.bash) the source code becomes

setup.bash, it should be forgotten that you should always build the file

By executing this command, the code will be executed:

```
rosrun apm_control apm_control.py
```

code:apm_control.py

```
#!/usr/bin/env python
# Entering the konabkhanehs and types of messages required by this program.
import rospy
from sensor_msgs.msg import NavSatFix
# Message responsible for transferring coordinates
# It is GPS
from mavros_msgs.srv import *
# Definition of two variables for coordinates.
latitude = 0.0
longitude = 0.0

# Definition of the function to change the flight mode:
def setToGuided():
    rospy.wait_for_service('/mavros/set_mode')
    try:
        flightModeService = rospy.ServiceProxy('/mavros/set_mode', mavros_msgs.srv.SetMode)
        isModeChanged = flightModeService(custom_mode='GUIDED')
        # switch flight mode
        if isModeChanged.mode_sent == True:
            print "GUIDED Mode could not be set. "Check that GPS is enabled"
        # If possible, a message will be sent to the user if the mode is not changed, which may be a problem of recognition
        else:
            print "service set_mode call failed: %s."
    except rospy.ServiceException, e:
        print "service set_mode call failed: %s. Unable to be the position"

# Definition of the function to change the flight mode:
def setToStabilize():
    rospy.wait_for_service('/mavros/set_mode')
    try:
```

```
                                        =flightModeService                  . 21
 SetMode ).srvmavros_msgs.,'/mavros/set_mode'ServiceProxy(rospy.
                                        =isModeChanged                      . 22
#return true or false)'STABILIZE'=custom_mode(flightModeService
                    E:asServiceExceptionrospy.except                        . 23
   "service set_mode call failed: %s. GUIDED(print                          . 24
            ) %eMode could not be set. "Check that GPS is enabled"
                                                                            . 25
          Definition of the function for quadrotor landing:()setToLanddef   . 26
           ) '/mavros/cmd/land'wait_for_service(rospy.                      . 27
                                                :try                        . 28
                                          =landService                      . 29
                                , '/mavros/cmd/land'ServiceProxy(rospy.
                                          CommandTOL).srvmavros_msgs.
, 0=latitude,0=altitude(landService=isLanding                               . 30
                                ) 0=yaw,0=min_pitch,0=longitude
                    E:asServiceExceptionrospy.except                        . 31
  "service land call failed: %s. The vehicle(print                          . 32
                                          ) %ecan't land
                                                                            . 33
       Definition of the function to turn on the engines:()setToArmdef      . 34
        ) '/mavros/cmd/arming'wait_for_service(rospy.                       . 35
                                                :try                        . 36
                                          =armService                       . 37
                                , '/mavros/cmd/arming'ServiceProxy(rospy.
                                          CommandBool ).srvmavros_msgs.
                                        ) True(armService                   . 38
                    E:asServiceExceptionrospy.except                        . 39
        ) %e"Service arm call failed: %s"(print                             . 40
                                                                            . 41
    Definition of the function to turn off the engines:()setToDisarmdef     . 42
           ) '/mavros/cmd/arming'wait_for_service(rospy.                    . 43
                                                :try                        . 44
                                          =armService                       . 45
                                , '/mavros/cmd/arming'ServiceProxy(rospy.
                                          CommandBool ).srvmavros_msgs.
                                        ) False(armService                  . 46
                    E:asServiceExceptionrospy.except                        . 47
           ) %e"Service arm call failed: %s"(print                          . 48
                                                                            . 49
                                                                            . 50
Definition of the function for flying at a height of ten meters:()setToTakeoffdef . 51
           ) '/mavros/cmd/takeoff'wait_for_service(rospy.                   . 52
                                                :try                        . 53
```

```
                                                =takeoffService                                 . 54
                                    , '/mavros/cmd/takeoff'ServiceProxy(rospy.
                                                                CommandTOL).srvmavros_msgs.
                    , 0=latitude,10=altitude(takeoffService                                     . 55
                                                ) 0=yaw,0=min_pitch,0=longitude
                                        E:asServiceExceptionrospy.except                        . 56
                    ) %e"Service takeoff call failed: %s"(print                                 . 57
                                                                                                . 58
                                                                                                . 59
                                                                                                . 60
                    :)globalPositionCallback(globalPositionCallbackdef                          . 61
                        Definition of the function responsible for determining the coordinates of the quadrotor position.  . 62
                                                                latitudeglobal                  . 63
                                                                longitudeglobal                 . 64
                        latitudeglobalPositionCallback.=latitude longitude                      . 65
                    globalPositionCallback.=longitude                                           . 66
                                    #print ("longitude: %.7f" %longitude )                      . 67
                                    #print ("latitude: %.7f" %latitude )                        . 68
                                                                                                . 69
Defining the function to write different options at the beginning of #:()menudef                . 70
                                                                                    Program
                                        #print ("\nWritten by Nabil\n" )                        . 71
                                ) "\nPLEASE CHOOSE A COMMAND\n"(print                           . 72
                                    ) "1: to set mode to GUIDED"(print                          . 73
                                ) "2: to set mode to STABILIZE"(print                           . 74
                                        ) "3: to ARM the drone"(print                           . 75
                                    ) "4: to DISARM the drone"(print                            . 76
                            ) "5: to TAKEOFF (10 meters high)"(print                            . 77
                                                    ) "6: to LAND"(print                        . 78
                                ) "\n"7: print coordinates(print                                . 79
                                                                                                . 80
    Definition of the function based on the user's choice of functions #:()choicesdef          . 81
                                                                    performs different
                                                                        '1'=x                   . 82
                                    :is_shutdown()))rospy.not((while                            . 83
                                                                ) (menu                         . 84
                                ;)"Enter your input: "(input=x :)"guided"==                     . 85
                xor"GUIDED"==xor'1'==x(if                                                       . 86
                                                    ) (setToGuided                              . 87
:)"stabilize"==xor""STABILIZE==xor'2'==x(elif                                                   . 88
                                                ) (setToStabilize                               . 89
                                :)arm==xor""ARM==xor'3'==x(elif                                 . 90
                                                        ) (setToArm                             . 91
                    :)"disarm"==xor""DISARM==xor'4'==x(elif                                     . 92
```

```
                                                    ) (setToDisarm                                  . 93
    :)takeoff==xor""TAKEOFF==xor'5'==x(elif                                                         . 94
                                                ) (setToTakeoff                                     . 95
            :)land==xor"LAND"==xor'6'==x(elif                                                       . 96
                                                    ) (setToLand                                    . 97
        :)print==xor""PRINT==xor'7'==x(elif                                                         . 98
                                                latitudeglobal                                      . 99
                                               longitudeglobal                                     . 100
    ) %longitudeLONGITUDE: %.7f"\n"(print                                                           . one hundred and one
        ) %latitude"\n"LATITUDE: %.7f(print                                                         . 102
                                                                :otherwise                          . 103
                                                ) "Exit"(print                                      . 104
                                    ) "Written by NABIL"(print                                      . 105
                                                                                                    . 106
                                                                                                    . 107
                                                                                                    . 108
                                                                                                    . 109
Define the main function where the function #:'__main__'==__name__if The options will be           . 110
 implemented and the program will not be shut down unless the user wants it.
                                                The name of the node is specified here
            ) True=anonymous,'apm_control'init_node(rospy.                                          . 111
      , "/mavros/global_position/raw/fix"Subscriberrospy.                                           . 112
                                                ) globalPositionCallback,NavSatFix
                                                                                                    . 113
                                                                ) (choices                          . 114
                                                            spin()rospy.                            . 115
                                                                                                    . 116
```

[1] V. Ermakov, "mavros," [Online]. Available: https://github.com/mavlink/mavros.

[2] dji, "flame wheel arf," dji, [Online]. Available: https://www.dji.com/cz/flame-wheel - arf/feature.

[3] ROBO.IN, "1045(10×4.5) SF Propellers Black 1CW+1CCW-1pair-Normal Quality," Generic (China), [Online]. Available: https://robu.in/product/10x4-5-sf-props-black-cw-2pc/.

[4] EMAX, "EMAX Cooling New MT2216 II 810KV Brushless Motor CW CCW with 1045 Propeller for RC Multicopter," EMAX, [Online]. Available: https://emaxmodel.com/products/emax - cooling-new-mt2216-ii-810kv-brushless-motor-cw-ccw-with-1045-propeller-for-rc - multicopter.

[5] ROBO.IN, "emax-mt2216-810kv-bldc-motor-cw-prop1045-combo-original," EMAX, [Online] . Available: https://robu.in/product/emax-mt2216-810kv-bldc-motor-cw-prop1045-combo - original/.

[6] EMAX, "EMAX Simon Series 20A For Muti-Copter," EMAX, [Online]. Available: https://emaxmodel.com/products/emax-simon-series-20a-for-muti-copter.

[7] Amazon, "Dilwe RC Drone ESC Emax Simon Series BLHeli 12A 20A 30A ESC 2-3S Battery for 130-210mm Racing Drone RC Part(20A)," Dilwe, [Online]. Available: https://www.amazon.com/Dilwe-BLHeli-Battery-130-210mm-Racing/dp/B07MFQJBGS.

[8] T. Gudde, "Flight Controllers explained for everyone.," Fusion Engineering, [Online] . Available: https://fusion.engineering/flight-controllers-explained-for-everyone/.

[9] ArduPilot, "Archived:APM 2.5 and 2.6 Overview," ArduPilot, 2021. [Online]. Available: https://ardupilot.org/copter/docs/common-apm25-and-26-overview.html.

[10] pixhawk, "Products," [Online]. Available: https://pixhawk.org/products/.

[11] ArduPilot Dev Team, "History of ArduPilot," [Online]. Available: https://ardupilot.org/planner2/docs/common-history-of-ardupilot.html.

[12] H. JIN, "Best Drone Controller 2022: Top Review For You," LUCIDCAM, 05 09 2022. [Online] . Available: https://lucidcam.com/best-drone-controller/.

[13] Banggood, "Ardupilot APM 2.8 Flight Controller Board Bend Pin with Protective Case for RC Multi Rotor Drone - With Compass," Topacc, [Online]. Available:

https://usa.banggood.com/Ardupilot-APM-2_8-Flight-Controller-Board-Bend-Pin-with - Protective-Case-for-RC-Multi-Rotor-Drone-p-1123217.html .

[14] HobbyKing, JR, [Online]. Available: https://hobbyking.com/en_us/jr-xg6-6-channel-2-4ghz - dmss-transmitter-w-telemetry-rg612bx-receiver-mode-2.html.

[15] AKCELL, "JR RG831B 8 CHANNEL DMSS RECEIVER," JR, [Online]. Available: https://akcelltrader.com/product/jr-rg831b-8-channel-dmss-receiver/.

[16] Wikipedia, "MAVLink," [Online]. Available: https://en.wikipedia.org/wiki/MAVLink.

[17] ArduPilot Dev Team, "SiK Telemetry Radio," ArduPilot, [Online]. Available: https://ardupilot.org/copter/docs/common-sik-telemetry-radio.html.

[18] Iranian Robotic, "Flight Controller and Equipment," 3DR, [Online]. Available: http://robosigma.com/single - ttl-3drobotics-3dr-radio-telemetry-kit-433mhz-500mw-for-apm-2-6-px4-flight-controller.

[19] Alitools, "NEO-M8N Flight Controller GPS Module with On-board Compass M8 Engine PX4 Pixhawk TR For OCDAY Drone GPS," OXA, [Online]. Available: https://alitools.io/en/showcase/neo-m8n-flight-controller-gps-module-with-on-board - compass-m8-engine-px4-pixhawk-tr-for-ocday-drone-gps-4001339980107.

[20] Techplayon, "What is GPS, how GPS module is used for base station applications, how to choose GPS module?," 04 10 2017. [Online]. Available: https://www.techplayon.com/gps - gps-module-used-base-station-applications-choose-gps-module/.

[21] "TATTU 4200mAh 3s 35c Lipo Battery," TATTU, [Online]. Available: https://www.getfpv.com/tattu-4200mah-3s-35c-lipo-battery.html.

[22] RC Corner, "GENS ACE 11.1V 50C 3S 4000MAH LIPO BATTERY PACK WITH XT60 PLUG," Gens Ace, [Online]. Available: https://www.rccorner.ae/gens-ace-11-1v-50c-3s-4000mah-lipo - battery-pack-with-xt60-plug.

[23] Solidworks, Dassault Systemes, [Online]. Available: https://www.solidworks.com/.

[24] Wikipedia, "Polylactic acid," [Online]. Available: https://en.wikipedia.org/wiki/Polylactic_acid.

]25[ArduPilot Dev Team, "Archived:APM2.x Wiring QuickStart," ArduPilot, [Online]. Available: https://ardupilot.org/copter/docs/connecting-the-apm2.html.

[26] ArduPilot, "What is ArduPilot?," ArduPilot, [Online]. Available: https://ardupilot.org/.

[27] ArduPilot Dev Team, "Vehicle Types Supported by ArduPilot," ArduPilot, [Online]. Available: https://ardupilot.org/copter/docs/common-all-vehicle-types.html.

[28] ArduPilot autotest system, "ArduPilot Firmware builds," ArduPilot, [Online]. Available: https://firmware.ardupilot.org/.

[29] M. Oborne, "Mission Planner Home," [Online]. Available: https://ardupilot.org/planner/.

[30] ArduPilot Dev Team, "APM Planner 2 Home," ArduPilot, [Online]. Available: https://ardupilot.org/planner2/.

[31] ArduPilot Dev Team, "MAVProxy," 2019. [Online]. Available: https://ardupilot.org/mavproxy/#:~:text=MAVProxy%20is%20a%20fully%2Dfunctioning,%E2%80%9Cdeveloper%E2%80%9D%20ground%20station%20software. .

[32] ArduPilot Dev Team, "Choosing a Ground Station," ArduPilot, [Online]. Available: https://ardupilot.org/copter/docs/common-choosing-a-ground-station.html.

[33] And. Khatkapour, "How to install Linux on Windows, Bat Tashin Jatzi," 05 01 2021. [Online]. Available: https://kaliboys.com/install-linux-on-virtualbox/.

[34] S. Loretz, "Ubuntu installation of ROS Noetic," Robot Operating System, [Online]. Available: http://wiki.ros.org/Installation/Ubuntu.

[35] ArduPilot Dev Team, "SITL Simulator (Software in the Loop)," ArduPilot, [Online]. Available: https://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html.

[36] 3D Robotics, "Setting up a Simulated Vehicle (SITL)," 3D Robotics, 2015-2016. [Online]. Available: https://dronekit-python.readthedocs.io/en/latest/develop/sitl_setup.html.

[37] S. Balasubramanian, "MavLink Tutorial for Absolute Dummies (Part – I)".

[38] MAVLink, "MAVLink Messages," [Online]. Available: https://mavlink.io/en/messages/common.html#COMMAND_LONG.

[39] ArduPilot Dev Team, "MAVLink Basics," 2021. [Online]. Available: https://ardupilot.org/dev/docs/mavlink-basics.html.

[40] MAVLink, "MAVLink Commands (MAV_CMD)," [Online]. Available: https://mavlink.io/en/messages/common.html#MAV_CMD.

[41] ArduPilot Dev Team, "Configuring a Telemetry Radio using Mission Planner," 2021. [Online] . Available: https://ardupilot.org/copter/docs/common-configuring-a-telemetry-radio-using - mission-planner.html.

[42] ArduPilot Dev Team, "Complete Parameter List," [Online]. Available: https://ardupilot.org/copter/docs/parameters.html.

[43] ArduPilot Dev Team, "Compass Calibration," [Online]. Available: https://ardupilot.org/copter/docs/common-compass-calibration-in-mission-planner.html.

[44] ArduPilot Dev Team, "Accelerometer Calibration," [Online]. Available: https://ardupilot.org/copter/docs/common-accelerometer - calibration.html#:~:text=Calibration%20steps, -

0the%20left%2Dside%20menu.&text=Click%20Calibrate%20Accel%20to%20start,each%20ax.
Warning&text=Under%20Setup%20%7C%20Mandatory%20Hardware%2C%20select,from%2

[45] ArduPilot Dev Team, "Radio Control Calibration," [Online]. Available: https://ardupilot.org/copter/docs/common-radio-control-calibration.html.

[46] ArduPilot Dev Team, "Electronic Speed Controller (ESC) Calibration," [Online]. Available: https://ardupilot.org/copter/docs/esc-calibration.html.

[47] V. Prithiviraj, "Yaw, pitch and roll rotations of a Quadcopter.," 04 2015. [Online]. Available: https://www.researchgate.net/figure/Yaw-pitch-and-roll-rotations-of-a-Quadcopter_fig1_280573614.

[48] ArduPilot Dev Team, "Flight Modes," [Online]. Available: https://ardupilot.org/copter/docs/flight-modes.html.

[49] ArduPilot Dev Team, "Arming the motors," [Online]. Available: https://ardupilot.org/copter/docs/arming_the_motors.html.

[50] aLDEID, "Category:Drones/Radio," 18 03 2018. [Online]. Available: https://www.aldeid.com/wiki/Category:Drones/Radio.

[51] GAZEBO, "Beginner: Overview," 2014. [Online]. Available: https://classic.gazebosim.org/tutorials?cat=guided_b&tut=guided_b1.

[52] B. Aderinola, "What is Gazebo simulation?," 18 09 2019. [Online]. Available: https://www.theconstructsim.com/ros-5-mins-028-gazebo-simulation/.

[53] N. Zarrabi, "Terrible dynamics according to the ROS-based adaptive system," [Online]. Available: https://faradars.org/courses/fvmec9809-introduction-to-robot-operating-system.

[54] NIRYO, "8 reasons why you should use ROS for robotics projects," [Online]. Available: https://service.niryo.com/en/blog/8-reasons-use-ros-robotics-projects.

[55] Y. Pyo, H. Cho, L. Jung and D. Lim, ROS Robot Programming, Seoul: ROBOTIS Co., Ltd, 2017.

[56] A. Anwar, "Part 3: Create Your First ROS Publisher and Subscriber Nodes," 11 02 2021. [Online]. Available: https://medium.com/swlh/part-3-create-your-first-ros-publisher-and-subscriber-nodes -2e833dea7598#:~:text=A%20ROS%20Node%20can%20be,is%20published%20to%20the%20TOPIC. .

[57] ROS, "ROS Tutorials," [Online]. Available: http://wiki.ros.org/ROS/Tutorials.

[58] ProgrammerSought, "ROS file structure," [Online]. Available: https://www.programmersought.com/article/86944167082/.

[59] Raspberry Pi, "Raspberry Pi 3 Model B," [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-3-model-b/.

[60] Chinmay, "Installing Ubuntu MATE on a Raspberry Pi," 04 April 2020. [Online]. Available: https://itsfoss.com/ubuntu-mate-raspberry-pi/.

[61] ROS, "Ubuntu installation of ROS Kinetic," [Online]. Available: http://wiki.ros.org/Installation/Ubuntu.

[62] H. Lee, J. Yoon, M.-S. Jang and K.-J. Park, "A Robot Operating System Framework for Secure UAV Communications",*sensors*, p. 19, 2021.

[63] V. Ermakov, "mavros," 2018. [Online]. Available: http://wiki.ros.org/mavros.

[64] C. Karney, "geographiclib," [Online]. Available: https://github.com/sotex/geographiclib.

[65] ArduPilot Dev Team, "Communicating with Raspberry Pi via MAVLink," [Online]. Available: https://ardupilot.org/dev/docs/raspberry-pi-via-mavlink.html.

[66] ArduPilot Dev Team, "Making a MAVLink WiFi bridge using the Raspberry Pi," [Online] . Available: https://ardupilot.org/dev/docs/making-a-mavlink-wifi-bridge-using-the-raspberry - pi.html.

[67] ArduPilot Dev Team, "Setting up SITL on Linux," [Online]. Available: https://ardupilot.org/dev/docs/setting-up-sitl-on-linux.html.

[68] ArduPilot Dev Team, "Using SITL," [Online]. Available: https://ardupilot.org/dev/docs/using - sitl-for-ardupilot-testing.html.

[69] ArduPilot Dev Team, "Using Gazebo Simulator with SITL," [Online]. Available: https://ardupilot.org/dev/docs/using-gazebo-simulator-with-sitl.html.

[70] V. Ermakov, "mavros_extras," [Online]. Available: http://wiki.ros.org/mavros_extras.

[71] V. Ermakov, "mavros-installation," [Online]. Available: https://github.com/mavlink/mavros/tree/master/mavros#installation.

[72] ArduPilot Dev Team, "ROS with SITL," [Online]. Available: https://ardupilot.org/dev/docs/ros - sitl.html.

[73] A. Khanal, "Developing in Ardupilot(1)," [Online]. Available: https://www.prokurainnovations.com/developing-in-ardupilot1/.

[74] ROS, "Home," 2022. [Online]. Available: https://www.ros.org/.

[75] L. Meier, "MAVLink Developer Guide," 2009. [Online]. Available: https://mavlink.io/en/.

[76] Y.-M. Kwon, J. Yu and K.-J. Park, "Empirical Analysis of MAVLink Protocol Vulnerability for Attacking Unmanned Aerial Vehicles",*Semantic Scholar*, vol. 6, p. 3, 2018.