Building a Quadcopter with Arduino

Quadcopters, also known as quadrotors, are gaining more and more popularity in today’s world. With the help of these devices, anyone can have an “eye in the sky” and can monitor any place at any time. You can capture photographs and perform automated tasks. In this book, you will be informed about all the basic modules and electronics needed to fly a simple quadcopter. You’ll acquire the knowledge to create a fully functional drone. Through this book, you’ll develop the skills needed to build a DIY drone that can capture pictures and record videos. You’ll understand the prerequisites of building a quadcopter, how to fly it, and also the basic principles to stabilize its flight.

This is an easy-to-follow guide with pictorial depictions and technical descriptions to help you assemble your vehicle.

Who this book is written for

This book is intended for hobbyists who are looking forward to fulfilling their dream of flying a quadcopter entirely made by them. Some basic experience of electronic circuits and aerial vehicles will make it easier for you, but it isn’t mandatory.

What you will learn from this book

- Review and choose the most suitable material for the build
- Set up and calibrate all of your electronic parts
- Control and fly your quadcopter, taking into account the weather conditions
- Engineer the structure of your vehicle
- Develop a flight controller with basic principles
- Reduce the possibility of future malfunctions
- Repair your drone if it crashes or faces any damage

This is an easy-to-follow guide with pictorial depictions and technical descriptions to help you assemble your vehicle.

In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 6 'Flight Controller Setup'
- A synopsis of the book’s content
- More information on Building a Quadcopter with Arduino
Vasilis Tzivaras is a software developer and hardware engineer who lives in Ioannina, Greece. He is currently an undergraduate student of the Department of computer science and Engineering at Ioannina. Alongside his studies, he is working on many projects relevant to robotics, such as drones, home automation, and smart home systems using Arduino and the Raspberry Pi. Furthermore, he is enthusiastic about clean energy solutions and cultural innovation ideas.

He has worked for the University Hospital of Ioannina as an assistant for various computer issues, and he has been part of the support team of his CSE department for over a year. He has participated in IEEE UOI Student Branch and other big organizations, such as FOSSCOMM, with personal presentations for website designing, programming, Linux systems, and drones.

As a member of IEEE University of Ioannina Student Branch, he has proposed many projects and solutions to automate homes and many other life problems by reducing the time of everyday routines. In addition to this, he has come up with ideas to entertain kids with funny and magical projects using Arduino-like hardware and open source software. Many of the projects can be found at his GitHub account under the name of BillyTziv.

Apart from this book, he has also worked on another book Programming in C, which is not yet published. In addition to this, he has written for blogs, forums, guides, and small chapters, explaining and sharing his knowledge of computers, networks, and programming.
Quadcopters have made an unexpected comeback from 1907, where all the research about multicopters paused due to stability issues. Nowadays, this kind of aerial vehicle is the perfect solution for many applications. If you want to capture a photo or record a video, transfer low-weight payload quickly and securely, or join races with other hobbyists, then you will have to buy one or even create one on your own.

As an author, I have tried to keep the simplicity of the quadcopter but also give plenty of instructions on how you can easily create one for your own application. As kids, we always wanted something that could fly and that we could play around with. Through this book, this dream can come true.

What this book covers

Chapter 1, Introduction to Quadcopters, provides information on various categories of drones, and you will be able to understand why a quadcopter is stable and can hover. Furthermore, some possible applications will be mentioned as well.

Chapter 2, Hardware Overview, analyzes every part of the quadcopter and gives some basic instructions about the usage of each component.

Chapter 3, Creating a Frame, analyzes some possible materials and the shapes and purposes of different frames, since each component will be mounted on a frame. Some frames are designed for stability purposes and photography, whereas others are used for quadcopter races.

Chapter 4, Soldering the Electronics, covers how to solder wires and create connections between different parts, as some of the electronics require further modifications as per individual requirements.
Chapter 5, *Electronics Installation*, contains some tips for better stability even though installing the electronics is a piece of cake. Notice that it's quite easy to damage something and have a problem later.

Chapter 6, *Flight Controller Setup*, covers how to set up ArduPilot, which is an open source and easy flight controller. A flight controller is the core component of your quadcopter. It contains all the software that is crucial for stability. Setting up a controller may be easy or very hard.

Chapter 7, *Flight Instructions*, helps you ask some basic questions and tries to make you think like a professional pilot. Your quadcopter is ready but flying is not so easy. There are many factors that you should think of before going out there and powering up your quadcopter; where you will fly, what will happen if something goes wrong, how much time you have, and so on. All these queries should be answered before flying.

Chapter 8, *Cameras and Extra Functions*, covers vibration issues and mechanisms that you can use or build on your own. Apart from all the basic components, there are more advanced ones that you can easily add to your quadcopter, depending on the purpose of the flight. FPV is a great way to have eyes in the sky and LEDs are necessary for night flights.

Chapter 9, *Repair and Maintenance*, covers how to maintain your vehicle, how to protect it from wear, and why this is crucial for safe flights. Even professionals face a crash. This is something that you will unfortunately experience, but the goal here is to estimate the damage and repair your quadcopter as fast as you can.
In the previous chapter, we installed all the electronic parts and, after connecting the battery, we verified that everything was working. The quadcopter needs to sense the environment. We need to calibrate our transmitter, accelerometer, and any other sensor that we might have. That way, the quadcopter will be able to understand if it is looking to the front, left, back, or right; it can also determine the necessary angles, which will later determine the exact maneuvers required for stabilization. In addition, proper calibration of the flight controller will result in a safe and stable flight. ArduPilot (APM FC) can be connected to a computer with a USB or a MicroUSB cable. This is the right way for all the calibrations that we need to do.

In this chapter, we will cover the following topics:

- Information about the structure of the Mission Planner software
- Uploading software to your ArduPilot flight controller
- Setting up your ArduPilot
- Tuning your flight controller

**About Mission Planner**

The program that we will be using to set up and calibrate our flight controller is called Mission Planner and you can find and download it from the official website of APM.

Download the Mission Planner software from the official website, as mentioned at http://ardupilot.com/downloads/.
Flight Controller Setup

After installing the software, you should be able to start the program and you will see the following interface, or something similar to this in case you download a later version of the software.

It may look like complicated software but after going through some installation procedures that you will read further in this chapter, it will become quite familiar to you. At first, let's analyze its functionalities and let's have a look at how the software is organized. The main menu is as shown in the following screenshot:

Let's look at an overview of the tabs present in the menu of Mission Planner:

- **FLIGHT DATA**: This tab gives you information about the speed, altitude, horizon, and a map of the location of your quadcopter according to GPS.
- **FLIGHT PLAN**: This tab gives you information about various things when preparing your quadcopter for flight plans (for example, navigation by points on a map).
- **INITIAL SETUP**: In this tab, you can set up your vehicle exactly as you want, go through the calibrations of the transmitter and accelerometer, and so on.
• **CONFIG/TUNING**: This tab consists of parameters and variables that can be changed to affect your stabilization system and other extra functionalities.

• **SIMULATION**: Mission Planner is equipped with simulation software for your real vehicle, where you can simulate anything you want before flying the real one.

• **TERMINAL**: This tab is another way of setting up and calibrating anything you want. It provides unlimited access to almost everything in the Mission Planner software. It is intended for Linux users who are familiar on the Linux terminal.

### Uploading firmware

Before connecting the ArduPilot by clicking on the top-right corner, we need to upload the firmware to ArduPilot. The firmware is actually all the blocks and files according to which your quadcopter can fly. The code behind the magic firmware is open source and can easily be found at GitHub ([https://github.com/diydrones/ardupilot](https://github.com/diydrones/ardupilot)). If you are an advanced programmer, you can join the community and help the ArduPilot flight controller to become better.

You have two options here. Either click on a figure and upload the latest firmware for your quadcopter, octocopter, and so on, or click on **Previous Versions**. After fetching what is necessary, you can upload a previous (maybe more stable) version of the firmware. But first of all, try to upload the ArduCopter V3.2.1 Quad firmware:
Connecting ArduPilot

ArduPilot can be easily connected to a computer via a USB or a MicroUSB cable. In Windows systems, ArduPilot is usually connected to COM3 but this actually depends on the devices that are also connected, so it might be a different COM. As you can see in the following screenshot, since I have more devices connected, ArduPilot is connected to COM5.

For the purpose of troubleshooting, please have a look at the Device Manager section that you can find in the Computer Management program:

On Mission Planner, you can now see that the selected port is COM5 and the bound rate is 115200 (default):
As soon as you click on CONNECT, there are several things that will happen. First, the software will load all the current parameters that your ArduPilot already has (it might take some time). After that, information in the bottom-left section will appear, as seen in the following screenshot. Also, you can now see the horizon of your quadcopter and maybe some error messages:

Note that when the figure was captured, my GPS was not installed and an error message appeared over the horizon section. As you can see, there is another message about my compass. I am using ArduPilot v2.8, in which they have removed the inner compass and I have not connected an external one, so to make my quad work, I had to disable the compass checklist. This is something that we will check later.
Calibrating ArduPilot

Now we are ready to calibrate the flight controller. If you click on the INITIAL SETUP tab there will be a new list, as shown in the next screenshot:

In the Mandatory Hardware section, we can calibrate anything we want.
Step 1 – frame type
Click on the Frame Type option and choose your frame if not previously selected:

Step 2 – accelerometer calibration
To calibrate your accelerometer, you need to have something with a 90 degree angle because you will need to set this vertically and it’s better to set the device as precisely as you can. The acceleration calibration will ask you to place the device as it is; so place the flight controller on a stable table. After that, click on Next and place the controller in various different positions according to the instructions in the Mission Planner software. Just make sure to check which side is where in the flight controller.
Flight Controller Setup

Step 3 – compass calibration

The next step is compass calibration. If you have an earlier version than 2.8, maybe you are lucky enough to have an internal compass, so this will work for you. From version 2.8, the inner compass was removed, so don’t do anything since we will disable this feature later.

![Compass Calibration Screenshot]

Step 4 – radio calibration

Using the Radio Calibration option, your ArduPilot will learn the minimum and maximum values of your transmitter. Just click on Calibrate Radio and move all your transmission sticks from the minimum to the maximum value. You should see some red lines crossed in the calibration software.
Now, we are almost finished. But there are a few more things that we need to set up. As we said earlier, if your ArduPilot version is 2.8 or 2.6, there will be no compass. This is a problem and your flight controller will never arm itself. As a result, your motors will never spin. This happens because as soon as the flight controller realizes that you want to arm it, it goes through a check list to see if everything is OK before letting you arm the craft and spin the motors. We will need to disable the compass check using Mission Planner, so the flight controller will bypass that check and let you arm itself.
Connect your ArduPilot and go to the CONFIG/TUNING tab. Then click on **Full Parameter List** and, on the right side, click on **Find** and search for the word **COMPASS**. You should get the following result:

![Parameter List](image)

Change the value from 1 to 0 in the **COMPASS_USE** field. This disables the compass.

**Summary**

In this chapter, we went through the Mission Planner software, which is free, very powerful, and easy to use for ArduPilot flight controllers. Unfortunately, not every flight controller has its own software and is obviously not compatible with Mission Planner. So, in case you have chosen another flight controller, you have to find out another way of doing everything we discussed. It should be clear that anything you buy must be calibrated and configured according to your setup. After going through the setup procedure, you will be able to arm and have a stable quadcopter.

In the following chapter, there will be instructions about your first flight and where and how you can fly safely without risk of damaging your quadcopter.
Where to buy this book
You can buy Building a Quadcopter with Arduino from the Packt Publishing website. Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers. Click here for ordering and shipping details.