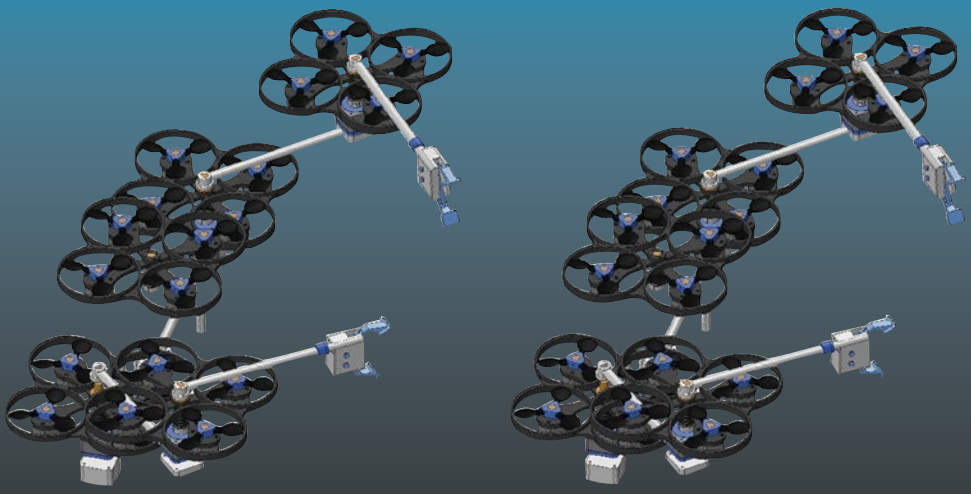
TECHNOLOGY IN ACTION™



Advanced Robotic Vehicles Programming

An Ardupilot and Pixhawk Approach

—

Julio Alberto Mendoza-Mendoza
Victor Gonzalez-Villela
Gabriel Sepulveda-Cervantes
Mauricio Mendez-Martinez
Humberto Sossa-Azuela

Apress®

Advanced Robotic Vehicles Programming

An Ardupilot and Pixhawk Approach

Julio Alberto Mendoza-Mendoza
Victor Gonzalez-Villela
Gabriel Sepulveda-Cervantes
Mauricio Mendez-Martinez
Humberto Sossa-Azuela

Apress®

Advanced Robotic Vehicles Programming: An Ardupilot and Pixhawk Approach

Julio Alberto Mendoza-Mendoza
FI UNAM, Ciudad de Mexico, Mexico

Victor Gonzalez-Villela
FI UNAM, Ciudad de Mexico, Mexico

Gabriel Sepulveda-Cervantes
CIDETEC IPN, Ciudad de Mexico, Mexico

Mauricio Mendez-Martinez
UPIITA IPN, Ciudad de México, Mexico

Humberto Sossa-Azuela
CIC IPN, Ciudad de México, Mexico

ISBN-13 (pbk): 978-1-4842-5530-8
ISBN-13 (electronic): 978-1-4842-5531-5
https://doi.org/10.1007/978-1-4842-5531-5

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Natalie Pao
Development Editor: James Markham
Coordinating Editor: Jessica Vakili

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com/rights-permissions.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-5530-8. For more detailed information, please visit www.apress.com/source-code.

Printed on acid-free paper

To my grandma, girlfriend, teachers, students, friends, co-workers, and family.
Julio Alberto Mendoza-Mendoza, 2019
Xijtemiki, ximonekilli, xichiua.
Dream it, wish it, do it.

Table of Contents

About the Authors

Julio Alberto Mendoza-Mendoza earned his computing doctoral degree at CIC IPN in 2016, where he specialized in underactuated robotics, UAS, and intelligent and nonlinear control. He also earned his advanced-technologies master's degree and mechatronics-engineering bachelor at UPIITA IPN in 2011 and 2008, respectively. Currently, Julio is working on five patents related to his research field and developing his flying serial-robot manipulator theory at FI UNAM through his postdoctoral 2017 DGAPA grant.

Victor Gonzalez-Villela acknowledges the financial support of the DGAPA postdoctoral fellowship, UNAM, given to the first author and above mentioned, the partial financial support of the "Support Program for Research and Technological Innovation Projects" (PAPIIT), UNAM, and the CONACYT for its support given via its "National Researchers System" (SNI file 57520). He wants to thank all of those who contributed in making this book a reality and the members of the Mechatronics Research Group at UNAM for their unconditional support and friendship (https://mechatronicsrg.wixsite.com/home).

Gabriel Sepulveda-Cervantes wants to thank to God and his family for the life and opportunity to grow and for the support received in all his projects. He also wants to thank his hobbies and jobs, the research in animation, virtual reality, haptics, and videogame programming. If you want to play videogame tournaments or work with him, maybe you can challenge him on UMVC3 where he can be found as TA Wero or through his workgroups at https://edissa.com.mx/ and CIDETEC IPN. He also thanks the IPN project under grant number SIP 20190245.

Mauricio Mendez-Martinez acknowledges his family and his students, teachers, and colleagues at UPIITA IPN.

Humberto Sossa-Azuela would like to acknowledge first of all his wife, Rocio, for her kindness, patience, and unconditional support. Secondly, he would like to thank his colleagues for their professional work to complete this book. To end up, he would like to express his deepest acknowledgment to the National Polytechnic Institute (IPN) and CONACYT for the always timely economical support to finish this project under grant numbers SIP 20190007 and 65 (Frontiers of Science), respectively.

Acknowledgments

Julio Alberto Mendoza-Mendoza and Victor Gonzalez-Villela are with the Advanced Engineering Center (CIA), Faculty of Engineering, UNAM. Gabriel Sepulveda is with the CIDETEC of the IPN. Mauricio Mendez is with the Interdisciplinary Professional Unit of Engineering and Advanced Technologies (UPIITA), IPN. Humberto Sossa-Azuela is with the Computer Research Center (CIC), IPN.

This book is a result of the Quetzalcoatl, Ehecatl, Papalotl, and Kukulkan projects for the design and operation of robotic flying arms, with five patent procedures in Mexico and with the invaluable collaboration of

- The National Autonomous University of Mexico (UNAM), through its DGAPA "Program of Postdoctoral Scholarships at UNAM" given to the first author, who is a postdoctoral researcher, since 2018, at the Advanced Engineering Centre, Division of Mechanical and Industrial Engineering, Faculty of Engineering, UNAM
- The National Polytechnic Institute (IPN), which also partially sponsored these works with SIP project 20164905

The first author wants to express appreciation for the access and use of laboratories and facilities and the participation of researchers from ESCOM, CMPL, UPP, Cinvestav, and CIDETEC, Dr. Carlos Aguilar Ibañez, Miguel Santiago Suarez Castañon, Ramon Silva Ortigoza, Ricardo Barron, Erik Zamora, Jesus Chimal, Rogelio Lozano-Leal, Eduardo Steed, Filiberto Muñoz, Leonardo Fonseca Ruiz, Jose Antonio Aquino, and Ignacio Garcia. Special mentions go to Orlando Garcia Perez and Manuel Jesus Rodriguez

from UAEH and UPIITA because without them this book would not exist, the first as our ArduPilot sensei and the second as a great friend and coach.

Gratitude to Aidronix, with broadcast and support of Pedro Matabuena, Tornillos Irator, TDRG and Marco Hugo Reyes Larios with the book-logo design support, Juan Jesus Gonzalez and Hazur Socconini with diffusion and patenting processes, Proyectil MX and projects INADEM 2.4 of the Mexican Secretary of Economy, and Francisco Arteaga and Jesus Castillo for our Quetzalcoatl aerial manipulator logo design (www.behance.net/jcmd).

Julio also thanks to the people of Mexico for their capacity to generate valuable results and the endurance and tenacity to always overcome the worst situations.

Clause of Responsibilities

Neither the publisher, nor the authors, nor the development community of ArduPilot or Pixhawk projects are responsible in any way for the use that the reader could make of any vehicle or robot designed, programmed, or operated by them.

It is the responsibility of each reader to do the following:

1. Read and properly understand the entire text of this book.
2. Have the proper permissions and security measures for their personal projects.
3. Use appropriate materials and equipment for their project.

Neither the publisher, nor the authors, nor the development community of ArduPilot or the Pixhawk projects will answer questions or observations related to the personal projects of each reader, no matter how important or urgent they may be. Understand that this book contains enough detailed material and in very specific cases, there exist online forums:

- http://discuss.px4.io/
- https://discuss.ardupilot.org/
- http://ardupilot.org/dev/docs/apmcopter-programming-libraries.html at Community section

CLAUSE OF RESPONSIBILITIES

Neither the publisher, nor the authors, nor the development community of ArduPilot or the Pixhawk projects are responsible for the damage that the reader could cause to their computer equipment or embedded systems during or after the installation process. It is the responsibility of each reader to follow the provided instructions.

Neither the publisher nor the authors are responsible for variations of code and syntax made by the ArduPilot development communities or the Pixhawk projects. This is understood since the software and hardware evolve, so this book should be interpreted as a base template to understand future modifications.

Foreword

Although the purpose of this book is to teach the capabilities of the ArduPilot libraries in conjunction with the Pixhawk autopilot exemplified with a quadcopter generic multi-rotor, it also provides guidelines for extending the knowledge acquired in other types of custom-designed aircraft (see the appendix on omnidirectionality) or terrestrial or aquatic vehicles, as well as other autopilots compatible with the aforementioned libraries.

This book is composed of an introductory section where the characteristics of the autopilot and the libraries are presented, a section of sequential programming focused on a didactic understanding of the most important parts of the ArduPilot libraries, where each main component of the code is described, and a final section of advanced character where the acquired knowledge is expanded to real-time applications.

Each section has a detailed description of the code and its components, its application and interaction, and of course, a bibliography suggested by the authors for those who want to deepen the exposed topics.

It should be mentioned that the use of the ArduPilot libraries is not exclusive to the Pixhawk autopilot and can be extended to many other platforms. However, this combination was adopted by the authors' preference, as well as by its performance capacities. This implies that reading this book offers training in these libraries and allows the end user to adapt their work to a wide range of autopilots and test platforms.

Warning

The use of aircraft and in general all types of vehicles in certain areas and countries are restricted by taxes, laws, and, of course, penalties, due the fact that equipment, buildings, living beings, and people can be damaged.

Even if the user has all the legal permissions to operate their vehicles, this book has the purpose of teaching to develop prototypes, so user tests must be carried out in closed spaces under adequate safety conditions or in open non-public spaces designed for those experiments.

The readers must avoid public places or restricted areas. The readers, and nobody else, are responsible for their tests. Avoid infringements and accidents.

Pay particular attention to LIPO mobile batteries. These batteries can be incendiary and explosive, which is an additional risk.

Finally, the cost of an aerial vehicle and its components can become expensive if you do not have the required care and design and this is therefore the responsibility of the reader.

Under no circumstances will the authors reply to private or public projects, regardless of the urgency. Instead, the reader is recommended to do additional research on forums.

Prior Knowledge

This book is aimed to anyone with a level of education equivalent at least to the last year of high school or a technical baccalaureate, since you must be familiar with the following concepts.

Mandatory Knowledge

Programming: Knowing how to use an Arduino is the necessary starting point.

Financial intelligence: The world of unmanned vehicles represents a moderate to high cost in the purchase of various components. You are responsible and solvent for the acquisition or assembly of your own vehicle and corresponding spare parts.

English: Frequently, it will be necessary to access forums, stores, and videos to resolve certain concerns. Given that the topics addressed here do not have such extensive information in local languages, it is necessary to have a moderate level of English.

Desirable but Not Essential

Mathematics: You should know basic derivation and elementary operations of vectors and matrices.

Physics and control: You should understand the concept of damped harmonic oscillator and PD control, the concept of force and torque, the use of coordinate systems, and how to obtain the components of movement and force.

Expected Results

With this book you will learn

- To program in an advanced way the Pixhawk, which is an open autopilot created by the ETH and widely used in the world of research and development of autonomous vehicles
- To use the ArduPilot libraries, one of the software development interfaces for unmanned vehicles research and development with most users around the world
- To model and implement elementary semiautomatic controls in any unmanned vehicle of the aerial, terrestrial, and aquatic type and in a very specific way in a quadcopter
- To link theory with practice in the development of unmanned vehicles
- To select hardware and software components during the design process of an unmanned vehicle
- To use other hardware and software development packages compatible with those described here
- The most employed scientific and technical nomenclature in the field
- To become familiar with the most relevant articles and books in the field

- To scale the knowledge applied with the quadcopter to three-dimensional vehicles through the basic theory of omnidirectional vehicles (those that can achieve “any” position with “any orientation”)
- To identify the most relevant complexities and processes for the operation of an unmanned vehicle if you plan to build your own autopilot (by programming microcontrollers, DSPs, FPGAs, embedded cards, or any other method that you prefer)

Licenses

ArduPilot libraries and Mission Planner software are GPLv3 license-free software. They can be redistributed and/or modified under GNU GPLv3 terms and restrictions as described by the Free Software Foundation (www.fsf.org/).

The code and programs are distributed with the hope of being useful but without any guarantee, even without the implied warranty of merchantability or capacity for a particular purpose. See the General Public License section of the GNU project for more details.

- ArduPilot libraries can be downloaded from http://ardupilot.org/dev/docs/apmcopter-programming-libraries.html.
- Mission Planner software can be downloaded from http://ardupilot.org/planner/.
- Pixhawk autopilot has a CC-BY-SA 3.0 license (https://creativecommons.org/licenses/by-sa/3.0/deed.es) which belongs to Lorenz Meier.
- Its official documentation is at https://dev.px4.io/en/contribute/licenses.html.
- PX4 libraries have a BSD 3-clause license (https://opensource.org/licenses/BSD-3-Clause).

Almost any terminal is public domain software. We recommend terminal.exe, putty, or any other equivalent.

LICENSES

The Java SE Development Kit 8u111 update with the executable named `jdk-8u111-windows-i586.exe` belongs to Oracle and, only if needed for the correct execution of the Eclipse version included with the libraries, can be downloaded from www.oracle.com/technetwork/java/javase/downloads/java-archive-javase8-2177648.html.

PART I

Introduction

CHAPTER 1

Hardware and Software Description

In this chapter, we will show you what an autopilot is. We'll also introduce the characteristics and history of the hardware and software used throughout this book, which are the ArduPilot libraries and the Pixhawk autopilot. You will learn the difference between a GUI and an SDK, and how many types of SDKs there are for programming autopilots. Additionally, we will discuss other compatible projects and you will learn to distinguish between clone and original versions.

Autopilot

An autopilot is an embedded card designed to perform on-board operations during the unmanned tasks of a vehicle, such as the flight of an aircraft, the journey of an autonomous car, the immersion of a submarine robot, or any other type of mobile robot.

Unlike a development card, the autopilot usually has a greater capacity for processing and data transfer. This is because

1. Orientation and position sensors are read.
2. Signals are read from the remote control.
3. Other sensors coupled to the system are read, either through analog ports or digital or serial transmission protocols.

J. A. Mendoza-Mendoza et al., *Advanced Robotic Vehicles Programming*,
https://doi.org/10.1007/978-1-4842-5531-5_1

4. Flight data is stored for later statistical or graphical use.
5. The unmanned vehicle is intercommunicated with other vehicles or a base on the ground, using wireless networks.
6. The battery is measured.
7. Visual and sound alerts are sent.
8. The control is processed.
9. The data obtained is filtered.
10. The control is written to the motors.
11. Selected processes are executed in real-time modules.
12. Demanding mathematical operations are performed in very short times, such as multiplication of large dimension matrices, calculation of trajectories, and estimation of speeds and accelerations.

With the demand for resources, a development card tends to collapse or simply can't achieve such performance. For example, the Arduino development board, in its mega model, cannot operate more than a brushless motor at 490hz since in principle its clock barely manages 300hz for a single motor, compromising the operation of the rest of the ports and systems.

Now, if we compare it against another type of development cards or even more sophisticated and specialized processors such as a Raspberry Pi or an FPGA, the autopilot only contains the minimum equipment necessary and is only optimized for the teleoperation of a vehicle; that is, writing to an adequate number of motors (from 4 to 12, for example), writing to auxiliary motors (servos, for example), reading of positioning and orientation data, data feedback and control by the remote user, storage of flight data, and additional reading of on-board equipment (distance sensors, GPS redundant modules, etc.). Therefore, space, weight, and power consumption are optimized for the task of driving a vehicle.

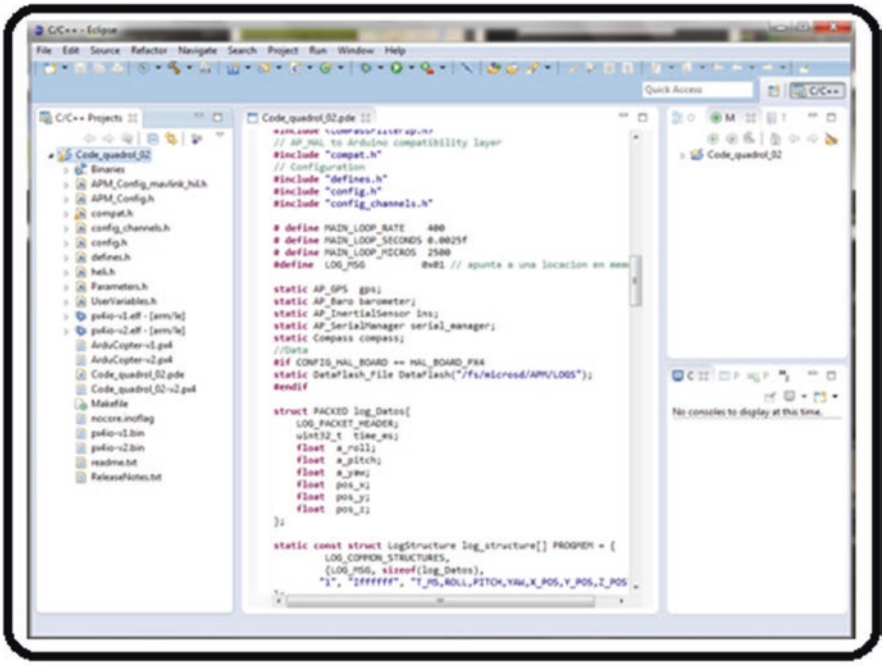
Among the best known autopilots are the Pixhawk, the Naza, the ArduPilot, the Crazyflie, and the CC3D.

Kinds of Autopilot: SDK vs. GUI

There are two types of autopilots, as shown in Figure 1-1.

- **Closed or semi-closed architecture programming**: In this case, a GUI (graphical user interface) is usually available. It is known as semi-closed programming because the user only has the ability to modify the parameters of predefined functions through a highly visual and interactive interface that does not imply knowing a specific programming language, where you can, for example, configure the path to be completed by the drone, but you cannot program each motor independently with a controller designed by the user. A GUI example is Mission Planner; it's used throughout this book but only as a way for graphic visualization, matching of telemetry communicators, loading designed programs to the autopilot, and extracting flight memory data.
- **Open architecture programming**: This case makes use of an SDK (software development kit). Here the programming is open because the users can modify flight parameters and also perform by themselves the whole algorithm of flight execution. This goes from the reading and filtering of the sensors, the incorporation of its own sensors, the choice of the data to be stored, up to the individual writing of each engine. This way requires the knowledge of a specific programming language.

GUI



SDK

Figure 1-1. *GUI vs. SDK interfaces*

Kinds of SDKs

For vehicles and especially drones there are three types of SDKs, as shown in Figure 1-2.

- **Cartesian command mode with yaw rotation**: In this case, the vehicle can only be "controlled" as a mass moving in X, Y, and Z directions and capable of turning on its vertical axis. In this case, the SDK only accepts position and rotation references, and it is not possible to command each component at a low level (motors, sensors, main control, etc.). As an example, see the appendix on dronekit.
- **Altitude command mode and attitude**: Here the vehicle can only be commanded as a mass moving in Z direction and capable of spinning on its three axes of movement. It's a moderately powerful development interface because although it allows the vehicle to have a command closer to what is necessary for experts in control, robotics, and artificial vision (among other areas), it still does not allow the most basic level of control and design: the motor level control. This is the case of the most recent SDK versions of the Ardrone multicopters.
- **Command mode on each motor**: This SDK allows you to control each engine. The flight of the unit is the responsibility of the designer. This is the case of the Pixhawk and the ArduPilot libraries and although it is risky if you do not read an extensive manual or this book, it provides the designer with greater control over all the components and behaviors of the aircraft or vehicle (remember that "with great power comes great responsibility").