

Autonomous Drones II

Robotics, Computer Vision, AI

Daniel Martínez González

DroneBooks

First Edition
May 2026

© 2026 Daniel Martínez González — All rights reserved

Volume 2 of 2

Autonomous Drones II: Robotics, Computer Vision, AI

Volume 2 of 2

First Edition, May 2026

Author: Daniel Martínez González

© 2026 Daniel Martínez González

ISBN: 9788409871711

Legal Deposit: M-14881-2026

SafeCreative Registration: SC-2026-2605145656280

All rights reserved.

No part of this work may be reproduced, transmitted, or stored in any form or by any means—electronic, mechanical, photocopying, recording, or otherwise—without prior written authorization from the author or rights holder.

GitHub: <https://github.com/DroneBooks/AutonomousDrones>

Contact: info@dronebooks.es

Disclaimer:

This book is an advanced guide for engineers, developers, and professionals who want to implement robotics and artificial intelligence systems in autonomous drones. The examples, techniques, and procedures described are provided for educational purposes. The author and publisher are not liable for damages, losses, or injuries resulting from the misuse of this information.

Special notice: Embedded AI systems in drones require exhaustive testing and constant human supervision. Do not deploy AI systems in production without complete validation. The author is not responsible for AI system malfunctions that result in damages.



Contents

Preface	11
1 Advanced Robotics Applied to Drones	19
1.1 Module Summary	21
1.2 What is ROS2?	22
1.2.1 History and Evolution	22
1.2.2 Why ROS2 for Drones?	22
1.2.3 ROS2 Versions	23
1.3 Installing ROS2 Humble on Ubuntu 22.04	24
1.3.1 Prerequisites	24
1.3.2 Step 1: Configure Ubuntu Repository	24
1.3.3 Step 2: Install ROS2 Humble	25
1.3.4 Step 3: Configure the Environment	25
1.3.5 Step 4: Verify Installation	26
1.4 ROS2 Fundamental Concepts	28
1.4.1 ROS2 Architecture	28
1.4.2 Key Concepts: Nodes, Topics, Services	29
1.5 Creating Your First ROS2 Node in Python	33
1.5.1 ROS2 Package Structure	33
1.5.2 Example 1: Simple Publisher Node	34
1.5.3 Example 2: Subscriber Node	36
1.6 Ardupilot – ROS2 Integration	37
1.6.1 How Does Ardupilot Communicate with ROS2?	37
1.6.2 Installing MAVROS	38
1.6.3 Example: Drone Control via ROS2	39
1.6.4 Complete Package: FlightStatus	42
1.7 Autonomous Navigation with Nav2	46

1.7.1	What is Nav2?	46
1.7.2	Installing Nav2	47
1.7.3	Example: Navigate to Point of Interest (POI)	47
1.8	Building and Running Packages	50
1.8.1	Build Package	50
1.8.2	Run Node	50
1.9	Essential ROS2 Tools	51
1.9.1	ros2 topic: Explore Topics	51
1.9.2	ros2 service: Services	52
1.9.3	ros2 node: Nodes	52
1.10	Simulation: Ardupilot + Gazebo + ROS2	53
1.10.1	Why Simulate?	53
1.10.2	Installing SITL + Gazebo	53
1.10.3	Example: SITL + ROS2 Simulation	53
1.11	Visualisation with RViz	54
1.11.1	Launch RViz	54
1.11.2	Visualise Drone	54
1.12	Sensor Integration in ROS2	55
1.12.1	GPS Sensor	55
1.12.2	IMU Sensor (Inertial Measurement Unit)	57
1.13	Real-World Use Cases	59
1.13.1	Autonomous Aerial Mapping	59
1.13.2	Object Tracking	59
1.13.3	Drone Swarm	60
1.14	Debugging and Troubleshooting	61
1.14.1	Common Issues	61
1.15	Best Practices and Design Patterns	64
1.15.1	Naming Nodes and Topics	64
1.15.2	Professional Logging	65
1.15.3	Robust Error Handling	66
1.15.4	Using Configuration Parameters	67
1.16	Intensive Practical Exercises	69
1.16.1	Exercise 1: ROS2 Chat	69
1.16.2	Exercise 2: Telemetry Monitor	69
1.16.3	Exercise 3: Interactive Navigator	69
1.17	Code Repository	70
1.18	Resources and References	71
1.18.1	Official Documentation	71

1.18.2	Community	71
1.19	Assessment	72
1.19.1	Knowledge Quiz	72
1.19.2	Practical Exercise	74
1.19.3	Integration Project	75
1.19.4	Assessment Solutions	77
1.20	Final Notes and Next Steps	80
1.20.1	Current Limitations	80
1.20.2	Advanced Paths	80
1.20.3	Final Tips	80
1.21	Glossary	81
2	Computer Vision Applied to Drones	83
2.1	Module Summary	85
2.2	Foundations of Computer Vision	87
2.2.1	What is Computer Vision?	87
2.2.2	Typical Vision Pipeline in Drones	87
2.2.3	Vision Computing: CPU vs GPU vs TPU	89
2.3	OpenCV: Fundamentals	91
2.3.1	Installation and Setup	91
2.3.2	Video Capture	91
2.3.3	Basic Image Processing	93
2.4	Object Detection: From Classical to Deep Learning	95
2.4.1	Classical Methods vs Deep Learning	95
2.5	YOLO v8: Real-Time Object Detection	97
2.5.1	Why is YOLO the Standard in Drones?	97
2.5.2	Installing YOLOv8	99
2.5.3	Example 1: Basic Detection	100
2.5.4	Example 2: Real-Time Detection from Camera	101
2.5.5	Example 3: Object Tracking	103
2.5.6	Available YOLO v8 Models	104
2.6	Alternative Technologies (Overview)	106
2.7	YOLO + ROS2 Integration	107
2.8	Optimization for Embedded Hardware	109
2.8.1	Converting to TensorRT	109
2.9	Use Cases in Drones	110
2.9.1	Person Detection + Alert	111
2.9.2	Obstacle Avoidance	111

2.9.3	Real-Time Detection Flow	112
2.10	Intensive Practical Exercises	113
2.10.1	Exercise 5.1: HSV Color Detector	113
2.10.2	Exercise 5.2: YOLO Detection	113
2.10.3	Exercise 5.3: Persistent Tracking	113
2.11	Best Practices	114
2.12	Next Step: Module 6 — Artificial Intelligence Applied to Drones	115
2.13	Additional Resources	116
2.14	Assessment	117
2.14.1	Knowledge Quiz	117
2.14.2	Practical Exercise: Real-Time HSV Color Detector	118
2.14.3	Integrating Project: Complete Vision System in ROS2	119
2.14.4	Assessment Solutions	120
2.15	Glossary of Terms	122
3	Artificial Intelligence Applied to Drones	125
3.1	Module Summary	127
3.2	The Fundamental Problem: Latency in Drones	129
3.2.1	Why is 100ms the Critical Limit?	129
3.2.2	Latency Breakdown in Drones	131
3.3	Why Cloud AI DOES NOT Work in Drones?	133
3.3.1	The Cloud Computing Illusion	133
3.3.2	Additional Cloud AI Problems	134
3.4	Jetson: NVIDIA Platforms for Edge AI	136
3.4.1	What is Jetson?	136
3.4.2	Jetson Product Line	137
3.4.3	Jetson SDK Installation	140
3.5	Online vs Offline Inference	141
3.5.1	Online Inference (During Flight)	142
3.5.2	Offline Inference (Post-Mission)	144
3.6	Power Optimization in Jetson	146
3.6.1	YOLO Power Consumption on Jetson	146
3.6.2	Power Optimization Strategies	147
3.7	Integration: ROS2 + Jetson + Ardupilot	149
3.7.1	Full System Architecture	149
3.7.2	Complete Example: Obstacle-Avoiding Drone	151

3.8	Practical Use Cases: Embedded AI in Drones	154
3.8.1	1. Autonomous Search and Rescue	154
3.8.2	2. Infrastructure Inspection	154
3.8.3	3. Precision Agriculture	154
3.8.4	4. Aerial Mapping with 3D Analysis	155
3.9	Step-by-Step Deployment: Your First AI Drone	157
3.9.1	Step 1: Prepare Jetson	157
3.9.2	Step 2: Create ROS2 Package with YOLO	158
3.9.3	Step 3: Build and Run	158
3.10	Troubleshooting and Optimization	159
3.10.1	Problem: YOLO is Too Slow	159
3.10.2	Problem: Jetson Overheats and Throttles	159
3.10.3	Problem: Battery Drains Quickly	159
3.11	Practical Exercises	160
3.11.1	Exercise 6.1: Measure Real Latency	160
3.11.2	Exercise 6.2: Convert Model to TensorRT	161
3.11.3	Exercise 6.2b (Optional): Power Monitoring on Jetson	162
3.11.4	Exercise 6.3: Person-Following Drone	163
3.11.5	Exercise 6.4: Post-Mission Analysis	164
3.12	Additional Resources	166
3.13	Assessment	167
3.13.1	Knowledge Quiz	167
3.13.2	Practical Exercise: Real Latency Measurement	168
3.13.3	Integration Project: Autonomous Drone with Embedded AI	169
3.13.4	Assessment Solutions	170
3.14	Conclusion: The Future of AI-Powered Drones	173
3.15	Glossary of Terms	174
A	Git for Drones	179
A.1	Introduction	181
A.2	Git Installation	182
A.2.1	Windows	182
A.2.2	macOS	182
A.2.3	Linux (Ubuntu/Debian)	182
A.3	Initial Configuration	183
A.4	Cloning Your First Repository	184

A.4.1	Step 1: Find the Repository URL	184
A.4.2	Step 2: Open Terminal / Command Prompt	185
A.4.3	Step 3: Create a Folder for the Courses	186
A.4.4	Step 4: Clone the Repository	187
A.5	Structure of the Downloaded Project	188
A.6	Detailed Practical Examples	189
A.6.1	Example 1: Download Volume 1 Code (Hardware, Ardupilot, MAVLink)	189
A.6.2	Example 2: Download Code from Both Volumes	190
A.6.3	Example 3: Update Existing Code	191
A.7	Updating the Code	192
A.8	Essential Commands (Cheat Sheet)	193
A.9	Assessment	194
A.9.1	Knowledge Quiz	194
A.9.2	Practical Exercise	196
A.9.3	Integration Project	197
A.9.4	Assessment Solutions	199
A.10	Glossary	202
A.10.1	Essential Git Terms	202
A.10.2	Platform Terms	203
A.10.3	Quick Git Commands (Summary)	203
A.10.4	Project Educational Terms	204
B	C++ for Drones	205
B.1	Introduction	207
B.2	Installing ROS2 and Compilers	208
B.2.1	ROS2 Humble (Ubuntu 22.04)	210
B.2.2	Compilers	210
B.3	Basic C++ Theory	211
B.3.1	Data Types	211
B.3.2	Conditionals: if, else, switch	212
B.3.3	Loops: for and while	213
B.3.4	Functions	214
B.4	Structure of a ROS2 Package in C++	216
B.5	Downloading Course Examples	218
B.5.1	Option 1: On Linux and macOS (Recommended)	218
B.5.2	Option 2: On Windows with Git Bash	218
B.5.3	Option 3: Download ZIP directly	219

B.6	Your First Node: Publisher	220
B.6.1	Complete Code	220
B.6.2	Code Explanation	221
B.6.3	What Happens When You Run It	221
B.6.4	Adapting for Drone Applications	221
B.7	Compilation with CMake	222
B.8	Build and Run	225
B.9	Publisher vs Subscriber	226
B.10	Advantages of C++ in Drones	227
B.11	Assessment	228
B.11.1	Knowledge Quiz	228
B.11.2	Practical Exercise	230
B.11.3	Integration Project	232
B.11.4	Assessment Solutions	234
B.12	Glossary of Terms	236
B.12.1	C++ Terms	236
B.12.2	ROS2 Terms	236
B.12.3	Build Terms	237
B.12.4	Drone Application Terms	237
	Glossary	239
	References	241