

Embedded Robotics

Thomas Bräunl

EMBEDDED ROBOTICS

From Mobile Robots
to Autonomous Vehicles
with Raspberry Pi
and Arduino

Fourth Edition



Springer

Thomas Bräunl
School of Engineering
The University of Western Australia
Perth, WA, Australia

ISBN 978-981-16-0803-2 ISBN 978-981-16-0804-9 (eBook)
<https://doi.org/10.1007/978-981-16-0804-9>

1st–3rd editions: © Springer-Verlag Berlin Heidelberg 2003, 2006, 2008

4th edition: © The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.

The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

PREFACE

This book gives a practical, in-depth introduction to embedded systems and autonomous robots, using the popular Raspberry Pi and Arduino microcontrollers. We demonstrate how to build a variety of mobile robots using a combination of these controllers, together with powerful sensors and actuators. Numerous application examples are given for mobile robots, which can be tested either on the real robots or on our freely available simulation system *EyeSim*.

This book combines teaching and research materials and can be used for courses in embedded systems as well as in robotics and automation. We see laboratories as an essential teaching and learning method in this area and encourage everybody to reprogram and rediscover the algorithms and systems presented in this book.

Although we like simulations for many applications and treat them in quite some depth in several places in this book, we do believe that students should also be exposed to real hardware in both areas, embedded systems and robotics. This will deepen their understanding of the subject area and, of course, create a lot more fun, especially when experimenting with small mobile robots.

We started this robotics endeavor over twenty years ago, when we first interfaced a digital image sensor to our own processor board. The *EyeBot-1* was based on a Motorola 68332 and our own operating system *RoBIOS*. The controller was soon followed by a variety of driving, walking, flying, swimming and diving robots that we called the EyeBot family. More powerful (and more expensive) controller architectures followed, until the availability of cheap, powerful boards like the Raspberry Pi and the Arduino Uno let us reconsider our approach. We redesigned our robot family around these popular controllers, but maintained the look and feel of our easy-to-use operating software.

Even on the simulation side, we have gone new ways. Our original stand-alone robot simulation has now been replaced by our new *EyeSim VR* package, which uses the Unity game environment with a much more realistic physics engine. The companion book *Robot Adventures in Python and C* (Springer 2020) concentrates on the software aspect of mobile robots and extends the hardware/software approach of this book.

For any embedded application, the processor power (and cost) needs to match to the given problem. For low-level control of a mobile robot with two DC motors and a few simple sensors, an 8-bit controller such as the Arduino might be sufficient. However, if we want to do image processing or learning, we need a more powerful controller like the 32-bit Raspberry Pi.

The EyeBot family consists of mobile robots with all sorts of propulsion systems. We and numerous other universities use these robots and simulation systems for laboratory experiments in embedded systems as part of the computer engineering, electrical engineering and mechatronics curriculum.

Acknowledgements

A number of colleagues and students contributed to the chapters, software systems and robotics projects presented in this book.

Working on the RoBIOS robot operating system were Remi Keat (base system), Marcus Pham (high-level control), Franco Hidalgo (low-level control) and Klaus Schmitt (driving routines).

The EyeBot controller board layout was done by Ivan Neubronner, and the EyeSim VR simulator was implemented by Travis Povey (Unity), Joel Frewin (robot models and applications), Michael Finn (terrain, underwater, swarms) and Alexander Arnold (VR).

The following colleagues and former students contributed to this book: Adrian Boeing on the evolution of walking gaits, Mohamed Bourgou on car detection and tracking, Christoph Braunschädel on PID control graphs, Louis Gonzalez and Michael Drtil on AUVs, James Ng on Bug and Brushfire algorithms, David Venkitachalam on genetic algorithms, Joshua Petitt on DC motors, Bernhard Zeisl on lane detection, Alistair Sutherland on balancing robots, Jordan King on traffic sign recognition and Nicholas Burleigh on deep learning for autonomous driving.

Additional Materials

All system software discussed in this book, the RoBIOS operating system, C/C++ compilers for Windows, MacOS and Linux, system tools, image processing tools, simulation system, and a large collection of example programs are available free from:

<http://robotics.ee.uwa.edu.au/eyebot/>

<http://robotics.ee.uwa.edu.au/eyesim/>

Software specific to the Raspberry Pi and Arduino (Nano) controllers can be downloaded from:

Preface

<http://robotics.ee.uwa.edu.au/rasp/>

<http://robotics.ee.uwa.edu.au/nano/>

As further reading, please refer to our companion book on programming of mobile robots: *Robot Adventures in Python and C*. This book makes heavy use of the EyeSim simulator, so it can be used for practical experiments without the need for a physical robot.

Perth, Australia

Thomas Bräunl

January 2022

CONTENTS

PART I: EMBEDDED SYSTEMS

1	Robots and Controllers	3
1.1	Mobile Robots	4
1.2	Embedded Controllers	8
1.3	Robot Design	10
1.4	Operating System	12
1.5	Simulation	14
1.6	Tasks	15
2	Central Processing Unit	17
2.1	Logic Gates	18
2.2	Function Units	23
2.3	Registers and Memory	29
2.4	Retro	32
2.5	Arithmetic Logic Unit	34
2.6	Control Unit	37
2.7	Central Processing Unit	38
2.8	Structured Design	49
2.9	Tasks	52
3	Arduino	53
3.1	Arduino Hardware	54
3.2	Arduino Programming	57
3.3	Arduino Interfacing	58
3.4	Arduino Communication	62
3.5	Beyond Arduino	63
3.6	Tasks	65
4	Raspberry Pi	67
4.1	Raspberry Pi Operating System and Setup	67
4.2	Raspberry Pi Tools and Programming	71
4.3	Raspberry Pi Input/Output Lines	77

4.4	Raspberry Pi Communication	80
4.5	Integration Development Environments	81
4.6	Tasks	83
5	Sensors and Interfaces	85
5.1	Sensor Categories	85
5.2	Synchronous Serial and I ² C Interfaces	87
5.3	Binary Sensors	89
5.4	Shaft Encoders	89
5.5	A/D Converters	91
5.6	Position Sensitive Devices—Sonar, Infrared, Laser	92
5.7	Lidar Sensors	96
5.8	Orientation Sensors	98
5.9	Inertial Measurement Units	99
5.10	Global Navigation Satellite Systems	101
5.11	Digital Image Sensors	104
5.12	Tasks	109
6	Actuators	111
6.1	DC Motors	111
6.2	H-Bridge	115
6.3	Pulse Width Modulation	116
6.4	Stepper Motors	118
6.5	Servos	119
6.6	Tasks	121
7	Control	123
7.1	On-Off Control	123
7.2	PID Control	129
7.3	Derivative Controller	133
7.4	Velocity Control and Position Control	136
7.5	Multiple Motors—Driving Straight	139
7.6	V-Omega Interface	142
7.7	Tasks	146
8	Multitasking	147
8.1	Preemptive Multithreading	148
8.2	Synchronization	149
8.3	Scheduling	154
8.4	Interrupts and Timer-Activated Tasks	157
8.5	Tasks	160
9	Communication	161
9.1	Communication Channels	161
9.2	File Transfer and Remote Access	163
9.3	Radio Library	166

Contents

9.4	Robot to Robot Communication	166
9.5	Robot to PC Communication	168
9.6	Tasks	171

PART II: ROBOT HARDWARE

10	Driving Robots	175
10.1	Single Wheel Drive	175
10.2	Differential Drive	176
10.3	Tracked Robots	189
10.4	Synchro-Drive	190
10.5	Ackermann Steering	192
10.6	Omni-Directional Robots	196
10.7	Drive Kinematics	202
10.8	Tasks	208
11	Walking Robots	209
11.1	Balancing Robots	210
11.2	Six-Legged Robots	215
11.3	Biped Robots	217
11.4	Static Balance	223
11.5	Dynamic Balance	226
11.6	Tasks	232
12	Autonomous Boats and Planes	233
12.1	Autonomous Boats	233
12.2	Autonomous Underwater Vehicles	236
12.3	Unmanned Aerial Vehicles (UAVs)	248
12.4	Tasks	251
13	Robot Manipulators	253
13.1	Homogeneous Coordinates	254
13.2	Manipulator Kinematics	256
13.3	Manipulator Simulation	261
13.4	Teaching and Programming	264
13.5	Industrial Manipulators	265
13.6	Tasks	269

PART III: ROBOT SOFTWARE

14	Localization and Navigation	273
14.1	Localization	273
14.2	Environment Representation	277
14.3	Quadtree	280
14.4	Visibility Graph	284
14.5	Voronoi Diagram and Brushfire Algorithm	286

14.6	Potential Field Method	289
14.7	Wandering Standpoint Algorithm	291
14.8	Bug Algorithm Family	292
14.9	Dijkstra's Algorithm	295
14.10	A* Algorithm	299
14.11	Probabilistic Localization	301
14.12	SLAM	306
14.13	Tasks	309
15	Maze Navigation	311
15.1	Micro Mouse Contest	311
15.2	Maze Exploration Algorithms	314
15.3	Simulated Versus Real Maze Program	321
15.4	Tasks	324
16	Image Processing	325
16.1	Camera Interface	325
16.2	Image File Formats	327
16.3	Edge Detection	330
16.4	Motion Detection	335
16.5	Color Spaces	338
16.6	RBG-to-HSI Conversion	342
16.7	Color Object Detection	345
16.8	Image Segmentation	349
16.9	Image Coordinates Versus World Coordinates	352
16.10	Tasks	355
17	Automotive Systems	357
17.1	Autonomous Automobiles	358
17.2	Drive-By-Wire and Safety Systems	361
17.3	Computer Vision for Autonomous Driving	364
17.4	OpenCV and KITTI	366
17.5	ROS	367
17.6	Carla Simulator	370
17.7	Lane Detection	372
17.8	Vehicle Recognition and Tracking	380
17.9	Automatic Parking	384
17.10	Autonomous Formula-SAE	388
17.11	Autonomous Shuttle Bus	392
17.12	Tasks	399

PART IV: ARTIFICIAL INTELLIGENCE

18	AI Concepts	403
18.1	Software Architecture	403
18.2	Behavior-Based Systems	405

Contents

18.3	Behavior Framework	406
18.4	Behavior-Based Applications	411
18.5	Tasks	419
19	Neural Networks	421
19.1	Neural Network Principles	421
19.2	Feed-Forward Networks	423
19.3	Backpropagation	428
19.4	Neural Network Examples	433
19.5	Neural Robot Control	435
19.6	Tasks	437
20	Genetic Algorithms	439
20.1	Genetic Algorithm Principles	440
20.2	Genetic Operators	442
20.3	Evolution Example	445
20.4	Implementing Genetic Algorithms	448
20.5	Genetic Robot Control	453
20.6	Starman	455
20.7	Evolving Walking Gaits	457
20.8	Tasks	467
21	Deep Learning	469
21.1	TensorFlow and Caffe	471
21.2	Carolo-Cup Competition	472
21.3	Traffic Sign Recognition	474
21.4	End-To-End Learning for Autonomous Driving	478
21.5	Tasks	484
22	Outlook	485
 APPENDICES		
Appendix A: RoBIOS Library		489
Appendix B: EyeBot-IO7 Interface		497
Appendix C: Hardware Description Table		503
Appendix D: Robot Programming Projects		507