

---

# Robot Adventures in Python and C

---

Thomas Bräunl

# Robot Adventures in Python and C

 Springer

Thomas Bräunl  
Engineering and Mathematical Sciences  
The University of Western Australia  
Perth, WA, Australia

ISBN 978-3-030-38896-6      ISBN 978-3-030-38897-3 (eBook)  
<https://doi.org/10.1007/978-3-030-38897-3>

© Springer Nature Switzerland AG 2020

This work is subject to copyright. All rights are reserved 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 Switzerland AG  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# PREFACE

.....

Contrary to common belief, the three most important topics in robotics are *not* Mechanics, Electronics and Software. Instead, they are Software, Software and Software! Many years ago, I wrote the book *Embedded Robotics* that kept a balance between electronics and software, while also presenting some basic mechanics content. This new book, however, concentrates mainly on software for mobile robots. This is where the real challenge lies and where the real innovation is happening.

In this book we demonstrate how inexpensive mobile robots such as our *EyeBot* robots can be constructed by mounting a Raspberry Pi controller and camera onto a model car or some other simple mechanical drive system. And we introduce our *EyeSim* simulation system, which is freely available and can quite realistically simulate a variety of driving, swimming/diving and even walking robots. Our emphasis is on algorithm development, and we ensure that all software projects can run on the real robot hardware as well as on the simulation system. This means, we do not use any unrealistic simulation assumptions that would never work in the real world.

At The University of Western Australia, we found that students using EyeSim as a supplementary teaching tool in robotics greatly improved their learning rate and understanding of robotics concepts.

All software used in this book, including all example programs, can be downloaded from the links below. There are native applications for MacOS, Windows, Linux and Raspberry Pi.

EyeBot real robots: <http://robotics.ee.uwa.edu.au/eyebot/>  
EyeSim simulation: <http://robotics.ee.uwa.edu.au/eyesim/>

In the following chapters, we will start with simple applications and move on to progressively more complex applications, from a small, simple driving robot to a full-size autonomous car.



## Preface

This book contains source code for most of the problems presented. In order to keep some order, we use a color-coding scheme to distinguish:

- Python programs
- C/C++ programs
- SIM scripts
- Robot definition files
- Environment data files

Tasks and challenges at the end of each chapter will help to deepen the learned concepts and let readers use their creativity in writing robot programs.

I hope you will enjoy this book and have fun recreating and extending the applications presented – and then go on to create your own robotics world!

My special thanks go to the UWA students who implemented EyeSim and also wrote some of the example programs: Travis Povey, Joel Frewin, Michael Finn and Alexander Arnold. You have done a great job!

Thanks for proofreading of the manuscript go to Linda Barbour and the team at Springer-Verlag.

Perth, Australia, March 2020

Thomas Bräunl

# CONTENTS



1	Robot Hardware	1
1.1	Actuators	2
1.2	Sensors	2
1.3	User Interface	5
1.4	Processor	6
1.5	Complete Robot	6
1.6	Communication	9
1.7	Simulation	9
1.8	Tasks	10
2	Robot Software	11
2.1	Software Installation	12
2.2	First Steps in Python	14
2.3	First Steps in C	16
2.4	Driving a Square in Python	17
2.5	Driving a Square in C or C++	18
2.6	SIM Scripts and Environment Files	19
2.7	Display and Input Buttons	20
2.8	Distance Sensors	21
2.9	Camera	25
2.10	Robot Communication	26
2.11	Multitasking	29
2.12	Using an IDE	30
2.13	Tasks	32
3	Driving Algorithms	33
3.1	Random Drive	33
3.2	Driving to a Target Position	38
3.3	Turn and Drive Straight	39
3.4	Circle	40
3.5	Dog Curve	42
3.6	Splines	43
3.7	Tasks	46
4	Lidar Sensors	47
4.1	Lidar Scans	47
4.2	Corners and Obstacles	50
4.3	Tasks	51

5	Robot Swarms	53
5.1	Setting up a Swarm	53
5.2	Follow Me	56
5.3	Multiple Followers	59
5.4	Tasks	63
6	Wall Following	65
6.1	Wall Following Algorithm	65
6.2	Simplified Wall Following Program	68
6.3	Tasks	70
7	Alternative Drives	71
7.1	Ackermann Steering	71
7.2	Omni-directional Drives	73
7.3	Driving in Terrain	77
7.4	Tasks	81
8	Boats and Subs	83
8.1	Mechanical AUV and Boat Structure	83
8.2	Specifying an Underwater Structure	84
8.3	Submarine Diving	86
8.4	Submarine Movement	87
8.5	Tasks	87
9	Mazes	89
9.1	Micromouse	90
9.2	Wall Following	90
9.3	Robustness and Control	94
9.4	Maze Driving with Lidar	96
9.5	Recursive Maze Exploration	100
9.6	Flood-Fill	104
9.7	Shortest Path	105
9.8	Tasks	108
10	Navigation	109
10.1	Navigation in Unknown Environments	109
10.2	DistBug Algorithm	110
10.3	Navigation in Known Environments	114
10.4	Quadtrees	115
10.5	Quadtree Implementation	116
10.6	Shortest Path Algorithm	121
10.7	Tasks	123
11	Robot Vision	125
11.1	Camera and Screen Functions	125
11.2	Edge Detection	127
11.3	OpenCV	131
11.4	Color Detection	132
11.5	Motion Detection	138
11.6	Tasks	141

## Contents

12	Starman	143
12.1	Moving Limbs	144
12.2	Motion Model	146
12.3	Genetic Algorithms	147
12.4	Evolution Run	151
12.5	Tasks	152
13	Driverless Cars	153
13.1	Autonomous Model Car Competitions	153
13.2	Carolo-Cup	154
13.3	Lane Keeping	156
13.4	Intersections and Zebra Crossings	157
13.5	Traffic Sign Recognition	158
13.6	End-to-End Learning	160
13.7	Tasks	162
14	Formula SAE	163
14.1	Electric Driving	163
14.2	Drive by Wire	164
14.3	Safety System	166
14.4	Autonomous Driving	167
14.5	Cone Track Racing	169
14.6	Tasks	174
15	Outlook	175
	Appendix	177