

1. Record Nr.	UNINA9910555268003321
Autore	Liu Shaoshan
Titolo	Engineering autonomous vehicles and robots : the DragonFly modular-based approach // Shaoshan Liu
Pubbl/distr/stampa	Hoboken : , : Wiley-IEEE Press, , 2020 [Piscataway, New Jersey] : , : IEEE Xplore, , [2020]
ISBN	1-119-57054-9 1-119-57051-4 1-119-57055-7
Edizione	[First edition.]
Descrizione fisica	1 online resource (216 pages)
Disciplina	629.046
Soggetti	Automated vehicles Mobile robots
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	1 Affordable and Reliable Autonomous Driving Through Modular Design 1 -- 1.1 Introduction 1 -- 1.2 High Cost of Autonomous Driving Technologies 2 -- 1.2.1 Sensing 2 -- 1.2.2 HD Map Creation and Maintenance 3 -- 1.2.3 Computing Systems 3 -- 1.3 Achieving Affordability and Reliability 4 -- 1.3.1 Sensor Fusion 4 -- 1.3.2 Modular Design 5 -- 1.3.3 Extending Existing Digital Maps 5 -- 1.4 Modular Design 6 -- 1.4.1 Communication System 7 -- 1.4.2 Chassis 7 -- 1.4.3 mmWave Radar and Sonar for Passive Perception 8 -- 1.4.4 GNSS for Localization 8 -- 1.4.5 Computer Vision for Active Perception and Localization 8 -- 1.4.6 Planning and Control 8 -- 1.4.7 Mapping 9 -- 1.5 The Rest of the Book 9 -- 1.6 Open Source Projects Used in this Book 10 -- References 11 -- 2 In-Vehicle Communication Systems 13 -- 2.1 Introduction 13 -- 2.2 CAN 13 -- 2.3 FlexRay 16 -- 2.3.1 FlexRay Topology 16 -- 2.3.2 The FlexRay Communication Protocol 17 -- 2.4 CANopen 18 -- 2.4.1 Object Dictionary 19 -- 2.4.2 Profile Family 19 -- 2.4.3 Data Transmission and Network Management 20 -- 2.4.4 Communication Models 21 -- 2.4.5 CANopenNode 21 -- References 22 -- 3 Chassis Technologies for Autonomous Robots and Vehicles 23 -- 3.1 Introduction 23 -- 3.2 Throttle-by-Wire 23 -- 3.3

Brake-by-Wire 25 -- 3.4 Steer-by-Wire 25 -- 3.5 Open Source Car Control 26 -- 3.5.1 OSCC APIs 26 -- 3.5.2 Hardware 27 -- 3.5.3 Firmware 28 -- 3.6 OpenCaret 29 -- 3.6.1 OSCC Throttle 29 -- 3.6.2 OSCC Brake 29 -- 3.6.3 OSCC Steering 29 -- 3.7 PerceptIn Chassis Software Adaptation Layer 30 -- References 34 -- 4 Passive Perception with Sonar and Millimeter Wave Radar 35 -- 4.1 Introduction 35 -- 4.2 The Fundamentals of mmWave Radar 35 -- 4.2.1 Range Measurement 36 -- 4.2.2 Velocity Measurement 37 -- 4.2.3 Angle Detection 38 -- 4.3 mmWave Radar Deployment 38 -- 4.4 Sonar Deployment 41 -- References 45 -- 5 Localization with Real-Time Kinematic Global Navigation Satellite System 47 -- 5.1 Introduction 47 -- 5.2 GNSS Technology Overview 47 -- 5.3 RTK GNSS 49. 5.4 RTK-GNSS NtripCaster Setup Steps 52 -- 5.4.1 Set up NtripCaster 52 -- 5.4.2 Start NtripCaster 54 -- 5.5 Setting Up NtripServer and NtripClient on Raspberry Pi 55 -- 5.5.1 Install the Raspberry Pi System 55 -- 5.5.2 Run RTKLIB-str2str on the Raspberry Pi 57 -- 5.5.2.1 Running NtripServer on the Base Station Side 57 -- 5.5.2.2 Running NtripClient on the GNSS Rover 58 -- 5.6 Setting Up a Base Station and a GNSS Rover 59 -- 5.6.1 Base Station Hardware Setup 59 -- 5.6.2 Base Station Software Setup 60 -- 5.6.3 GNSS Rover Setup 67 -- 5.6.3.1 Rover Hardware Setup 67 -- 5.6.3.2 Rover Software Setup 68 -- 5.7 FreeWave Radio Basic Configuration 71 -- References 75 -- 6 Computer Vision for Perception and Localization 77 -- 6.1 Introduction 77 -- 6.2 Building Computer Vision Hardware 77 -- 6.2.1 Seven Layers of Technologies 78 -- 6.2.2 Hardware Synchronization 80 -- 6.2.3 Computing 80 -- 6.3 Calibration 81 -- 6.3.1 Intrinsic Parameters 81 -- 6.3.2 Extrinsic Parameters 82 -- 6.3.3 Kalibr 82 -- 6.3.3.1 Calibration Target 83 -- 6.3.3.2 Multiple Camera Calibration 83 -- 6.3.3.3 Camera IMU Calibration 84 -- 6.3.3.4 Multi-IMU and IMU Intrinsic Calibration 84 -- 6.4 Localization with Computer Vision 85 -- 6.4.1 VSLAM Overview 85 -- 6.4.2 ORB-SLAM2 86 -- 6.4.2.1 Prerequisites 86 -- 6.4.2.2 Building the ORB-SLAM2 Library 87 -- 6.4.2.3 Running Stereo Datasets 87 -- 6.5 Perception with Computer Vision 87 -- 6.5.1 ELAS for Stereo Depth Perception 88 -- 6.5.2 Mask R-CNN for Object Instance Segmentation 89 -- 6.6 The DragonFly Computer Vision Module 90 -- 6.6.1 DragonFly Localization Interface 90 -- 6.6.2 DragonFly Perception Interface 92 -- 6.6.3 DragonFly+ 93 -- References 94 -- 7 Planning and Control 97 -- 7.1 Introduction 97 -- 7.2 Route Planning 97 -- 7.2.1 Weighted Directed Graph 98 -- 7.2.2 DijkstraOs Algorithm 99 -- 7.2.3 A* Algorithm 100 -- 7.3 Behavioral Planning 100 -- 7.3.1 Markov Decision Process 101 -- 7.3.2 Value Iteration Algorithm 102 -- 7.3.3 Partially Observable Markov Decision Process (POMDP) 103. 7.3.4 Solving POMDP 104 -- 7.4 Motion Planning 105 -- 7.4.1 Rapidly Exploring Random Tree 105 -- 7.4.2 RRT* 106 -- 7.5 Feedback Control 107 -- 7.5.1 ProportionalIntegralDerivative Controller 108 -- 7.5.2 Model Predictive Control 108 -- 7.6 Iterative EM Planning System in Apollo 110 -- 7.6.1 Terminologies 110 -- 7.6.1.1 Path and Trajectory 110 -- 7.6.1.2 SL Coordinate System and Reference Line 110 -- 7.6.1.3 ST Graph 111 -- 7.6.2 Iterative EM Planning Algorithm 112 -- 7.6.2.1 Traffic Decider 113 -- 7.6.2.2 QP Path and QP Speed 114 -- 7.7 PerceptInOs Planning and Control Framework 116 -- References 118 -- 8 Mapping 119 -- 8.1 Introduction 119 -- 8.2 Digital Maps 119 -- 8.2.1 Open Street Map 120 -- 8.2.1.1 OSM Data Structures 120 -- 8.2.1.2 OSM Software Stack 121 -- 8.2.2 Java OpenStreetMap Editor 121 -- 8.2.2.1 Adding a Node or a Way 123 -- 8.2.2.2 Adding Tags 123 -- 8.2.2.3 Uploading to OSM 124 -- 8.2.3 Nominatim 124 -- 8.2.3.1 Nominatim Architecture 124 -- 8.2.3.2 Place Ranking in

Nominatim 125 -- 8.3 High-Definition Maps 125 -- 8.3.1
Characteristics of HD Maps 126 -- 8.3.1.1 High Precision 126 --
8.3.1.2 Rich Geometric Information and Semantics 126 -- 8.3.1.3 Fresh
Data 126 -- 8.3.2 Layers of HD Maps 126 -- 8.3.2.1 2D Orthographic
Reflectivity Map 127 -- 8.3.2.2 Digital Elevation Model 127 -- 8.3.2.3
Lane/Road Model 127 -- 8.3.2.4 Stationary Map 127 -- 8.3.3 HD Map
Creation 127 -- 8.3.3.1 Data Collection 127 -- 8.3.3.2 Offline
Generation of HD Maps 128 -- 8.3.3.2.1 Sensor Fusion and Pose
Estimation 128 -- 8.3.3.2.2 Map Data Fusion and Data Processing 129
-- 8.3.3.2.3 3D Object Location Detection 129 -- 8.3.3.2.4
Semantics/Attributes Extraction 129 -- 8.3.3.3 Quality Control and
Validation 129 -- 8.3.3.4 Update and Maintenance 129 -- 8.3.3.5
Problems of HD Maps 130 -- 8.4 Perception-Map 130 --
8.4.1 Topological Map 130 -- 8.4.2 Perception-Map Creation 131 --
References 133 -- 9 Building the DragonFly Pod and Bus 135 -- 9.1
Introduction 135 -- 9.2 Chassis Hardware Specifications 135.
9.3 Sensor Configurations 136 -- 9.4 Software Architecture 138 -- 9.5
Mechanism 142 -- 9.6 Data Structures 144 -- 9.6.1 Common Data
Structures 144 -- 9.6.2 Chassis Data 146 -- 9.6.3 Localization Data
149 -- 9.6.4 Perception Data 150 -- 9.6.5 Planning Data 153 -- 9.7
User Interface 158 -- References 160 -- 10 Enabling Commercial
Autonomous Space Robotic Explorers 161 -- 10.1 Introduction 161 --
10.2 Destination Mars 162 -- 10.3 Mars Explorer Autonomy 163 --
10.3.1 Localization 163 -- 10.3.2 Perception 164 -- 10.3.3 Path
Planning 165 -- 10.3.4 The Curiosity Rover and Mars 2020 Explorer
165 -- 10.4 Challenge: Onboard Computing Capability 168 -- 10.5
Conclusion 169 -- References 170 -- 11 Edge Computing for
Autonomous Vehicles 171 -- 11.1 Introduction 171 -- 11.2
Benchmarks 172 -- 11.3 Computing System Architectures 173 -- 11.4
Runtime 175 -- 11.5 Middleware 177 -- 11.6 Case Studies 178 --
References 179 -- 12 Innovations on the Vehicle-to-Everything
Infrastructure 183 -- 12.1 Introduction 183 -- 12.2 Evolution of V2X
Technology 183 -- 12.3 Cooperative Autonomous Driving 186 -- 12.4
Challenges 188 -- References 189 -- 13 Vehicular Edge Security 191
-- 13.1 Introduction 191 -- 13.2 Sensor Security 191 -- 13.3
Operating System Security 192 -- 13.4 Control System Security 193 --
13.5 V2X Security 193 -- 13.6 Security for Edge Computing 194 --
References 196 -- Index 199.

Sommario/riassunto

"The first book of its kind on the detailed steps for creating an autonomous vehicle or robot, this book provides an overview of the technology and introduction of the key elements involved in developing autonomous vehicles, and offers an excellent introduction to the basics for someone new to the topic of autonomous vehicles and the innovative, modular-based engineering approach called DragonFly. Engineering Autonomous Vehicles and Robots: The DragonFly Modular-based Approach covers everything that technical professionals need to know about: CAN bus; chassis; sonars; radars; GNSS; computer vision; localization; perception; motion planning etc. Particularly, it covers Computer Vision for active perception and localization, as well as mapping and motion planning. The book offers several case studies on the building of an autonomous passenger pod, bus, and vending robot etc. It features a large amount of supplementary material, including the standard protocol and sample codes for chassis, sonar, and radar. GPSD protocol/NMEA protocol and GPS deployment methods are also provided. Most importantly, readers will learn the philosophy behind the DragonFly modular-based design approach, which empowers readers to design and build their own autonomous vehicles and robots with flexibility and affordability"--
