

# Contents

<b>Preface</b> -----	<b>v</b>
Who is this book for? -----	v
What do you need?-----	v
What is in this book? -----	v
Trademark Attributions-----	vi
<b>Contents</b> -----	<b>ix</b>
<b>Chapter 4: Enhanced Pan-Tilt Commando with RPi4B</b> -----	<b>1</b>
4.1 Using TASK -----	3
4.1.1 <i>Dual Control from Mobile Device and/or PC Virtual Interface</i> -----	3
4.1.2 <i>Dual Control with Autonomous Obstacle Avoidance</i> -----	12
4.2 Using MicroPython -----	17
4.2.1 <i>Dual Control from Mobile Device and/or PC Virtual Interface</i> -----	17
4.2.2 <i>Dual Control with Autonomous Obstacle Avoidance</i> -----	25
4.3 Using Standard Python on RPi4B and Desktop PC-----	29
4.3.1 <i>RPi4B as Vision Processor/Remote Controller</i> -----	30
4.3.2 <i>Desktop PC as Central Data Hub</i> -----	33
4.4 Using C/C++ on RPi4B and Desktop PC -----	35
4.4.1 <i>RPi4B as Vision Processor/Remote Controller</i> -----	35
4.4.2 <i>Desktop PC as Central Data Hub</i> -----	38
4.5 Using DXL-HAT and Dynamixel SDK -----	39
4.5.1 <i>RPi4B as Vision Processor/Remote Controller</i> -----	42
4.5.2 <i>CM-550 as Smart Avoider</i> -----	51
4.6 Desktop PC as Supervisory Controller -----	54
4.6.1 <i>Revising Robot Communications Scheme</i> -----	54
4.6.2 <i>Supervisory Control Demonstration for E-PTC</i> -----	60
4.7 References -----	63
<b>Chapter 5: Quadruped and its Variants with RPi4B</b> -----	<b>65</b>

5.1 Using TASK -----	68
5.1.1 Dual Control from RPi4B/Desktop PC-----	68
5.1.2 Dual Control with Autonomous Obstacle Avoidance -----	76
5.1.3 Articulated Four-Wheeled Platform (A4WP-123)-----	77
5.1.4 Hybrid A4WP (A4WP-H)-----	82
5.2 Using MicroPython -----	89
5.2.1 Using MTN3 Motion Files-----	90
5.2.2 Using Time Control and Motion Arrays -----	97
5.3 Using Python on RPi4B/Desktop PC-----	114
5.3.1 RPi4B as Vision Processor/Remote Controller-----	114
5.3.2 Desktop PC as Central Data Hub-----	115
5.4 Using C++ on RPi4B and Desktop PC -----	116
5.4.1 RPi4B as Vision Processor/Remote Controller-----	116
5.4.2 Desktop PC as Central Data Hub-----	117
5.5 Applying IDN Concept to A4WP-H with DXL-HAT + DXL-SDK -----	117
5.5.1 Mechanical Cabling Issue -----	117
5.5.2 CM-550 to RPi4B Communications Issue -----	119
5.5.3 CM-550's TASK & MicroPython Solutions -----	120
5.5.4 RPi4B's C++ Solution -----	129
5.6 Supervisory Control of A4WP-H (and E-PTC) -----	154
5.6.1 RPi4B C++ Solution for A4WP-H-----	154
5.6.2 Desktop PC Python Solution for A4WP-H and E-PTC-----	159
5.6.3 Supervisory Control Demonstration for A4WP-H-----	162
5.6.4 Supervisory Control Demonstration for A4WP-H + E-PTC-----	165
5.7 References-----	167
<b>Chapter 6: Enhanced SPI: CM-550 vs. RPi4B vs. Jetson Nano -----</b>	<b>169</b>
6.1 Camera-less Solution in TASK/MicroPython-----	170
6.2 Machine Vision Solution with RPi4B -----	177
6.2.1 Hardware/Communication Configurations for RPi4B + E-SPI-----	178
6.2.2 CM-550 Solution "IDN_SPI_SA_RC_RPi.py"-----	179
6.2.3 RPi4B Solution "IDN_SPI_SA_SSC_VSR_RPi.cpp"-----	183
6.3 Machine Vision Solution with Jetson Nano 2GB -----	191

6.3.1 Hardware/Communication Configurations for J-Nano + E-SPI-----	192
6.3.2 CM-550 Solution “IDN_SPI_SA_RC_JN.py”-----	194
6.3.3 J-Nano Solution “IDN_SPI_SA_SSC_VSR_JN.cpp”-----	194
6.4 References-----	198
<b>Chapter 7: Enhanced MAX-E2 with RPi4B-----</b>	<b>199</b>
7.1 Hardware/Communication Configurations for RPi4B + E-ME2 -----	199
7.2 Motion Offsets Determination -----	201
7.3 CM-550 Solution in TASK/MicroPython-----	202
7.4 RPi4B Solution in C++ -----	206
7.5 References -----	210
<b>Appendix B: Using RPi4B-----</b>	<b>211</b>
B.1 Hardware and Mechanical Design Considerations-----	211
<i>B.1.1 Power Options</i> -----	211
<i>B.1.2 Memory Storage Options</i> -----	213
<i>B.1.3 Compatible Peripherals</i> -----	213
B.2 RPi OS Software Considerations -----	215
B.3 Setting up BlueTooth Services between RPi and Windows PC-----	218
B.4 Setting up Python 3 and Visual Studio Code-----	220
B.5 Setting up Code::Blocks -----	220
B.6 Setting up ZigBee SDK-----	220
B.7 Setting up BOOST libraries -----	221
B.8 Setting up OpenCV-----	222
B.9 Setting up DXL-HAT and Dynamixel SDK -----	222
B.10 Using Book Example Codes-----	225
<b>Appendix C: Using NVIDIA Jetson Nano -----</b>	<b>227</b>
C.1 Hardware and Mechanical Design Considerations-----	227
<i>C.1.1 Power Options</i> -----	227
<i>C.1.2 Memory Storage Options</i> -----	228
<i>C.1.3 Compatible Peripherals</i> -----	229
C.2 Linux Tegra OS Software Considerations-----	230
C.3 BlueTooth Services between J Nano and Windows PC? -----	231
C.4 Setting up Python 3 and Visual Studio Code-----	231

C.5 Setting up Code::Blocks -----	231
C.6 Setting up ZigBee SDK-----	231
C.7 Setting up Dynamixel SDK -----	232
C.8 Using OpenCV -----	232
C.9 Using GStreamer Framework-----	233
C.10 Using Book Example Codes-----	233