

Contents

Preface -----	v
Who is this book for? -----	vi
What do you need? -----	vi
What is in this book?-----	vi
Trademark Attributions-----	vii
Contents -----	ix
Chapter 1: Hardware and Software Overview -----	1
1.1 CM-550-----	1
1.2 2XL430-W250-T and XL430-W250-T-----	3
1.3 Software Tools -----	4
1.4 ROBOTIS Dynamixel Network -----	7
1.5 References-----	10
Chapter 2: SimpleBot With Arms -----	13
2.1 Using TASK -----	14
2.2.1 “STEP” Position Control mode (PC Mode 1) -----	16
2.1.2 “RECTANGULAR” Position Control mode (PC Mode 2)-----	22
2.1.3 “TRAPEZOIDAL” Position Control mode (PC Mode 3) -----	25
2.1.4 “TIMED” Position Control mode (PC Mode 4)-----	28
2.1.5 Position Control with SyncWrite -----	34
2.1.6 Motion Play Features: Motion Speed, Stop Page, Joint Offset-----	35
2.1.7 Using Smart Device -----	40
2.2 Using MicroPython -----	46
2.2.1 “STEP” Position Control mode (PC Mode 1) -----	49
2.2.2 “RECTANGULAR” Position Control mode (PC Mode 2)-----	57
2.2.3 “TRAPEZOIDAL” Position Control mode (PC Mode 3) -----	60
2.2.4 “TIMED” Position Control mode (PC Mode 4)-----	62
2.2.5 Position Control with SyncWrite -----	66

2.2.6	<i>Motion Play Features: Motion Speed, Stop Page, Joint Offset</i>	68
2.2.7	<i>Using Smart Device</i>	74
2.3	<i>Using OpenCV-Python & PySerial on Desktop</i>	80
2.3.1	<i>Python Implementation of Remocon Packet Creation on PC Side</i>	81
2.3.2	<i>Handling of Remocon Packet on CM-550 Side</i>	84
2.3.3	<i>Setting Remote Port and App Port on CM-550</i>	86
2.3.4	<i>Tracking Colored Target with OpenCV on Desktop Side</i>	87
2.3.5	<i>Coordinating Actions by CM-550 in TASK</i>	95
2.3.6	<i>Coordinating Actions by CM-550 in MicroPython</i>	98
2.3.7	<i>Creating User Project on ENGINEER App</i>	99
2.3.8	<i>Run-Time Coordination for CM-550/Smart Device/Desktop PC</i>	100
2.4	<i>Using OpenCV and ZigBee SDK in C/C++ on Desktop</i>	100
2.4.1	<i>Using ZigBee SDK with VS 2019</i>	101
2.4.2	<i>Project “SBwA_Color_Tracker_GP.cpp”</i>	104
2.4.3	<i>Project “SBwA_Color_Tracker_XY.cpp”</i>	117
2.5	<i>References</i>	118
Chapter 3:	Commando with Pan-Tilt Platform + Camera	121
3.1	<i>Using MANAGER</i>	123
3.2	<i>Using TASK</i>	125
3.2.1	<i>Hi-Res Display of Random Finger Touches</i>	125
3.2.2	<i>Wheel Synchronization & Maneuver Compensation with IMU</i>	128
3.2.3	<i>Pi Camera: Color Detection & Visual Servoing/Ranging</i>	140
3.2.4	<i>Color Track Following Using Pi Camera</i>	148
3.2.5	<i>Scheduling Maneuvers Using ROBOTIS QR Markers</i>	157
3.2.6	<i>Dual-Bot RC Using ZigBee Broadcast</i>	165
3.3	<i>Using MicroPython</i>	176
3.3.1	<i>Hi-Res Display of Random Finger Touches</i>	176
3.3.2	<i>Wheel Synchronization & Maneuver Compensation with IMU</i>	179
3.3.3	<i>Pi Camera: Color Detection & Visual Servoing/Ranging</i>	189
3.3.4	<i>Color Track Following Using Pi Camera</i>	194
3.3.5	<i>Scheduling Maneuvers Using ROBOTIS QR Markers</i>	199
3.3.6	<i>Dual-Bot RC Using ZigBee Broadcast</i>	204

3.4 Using Standard Python-----	214
3.4.1 <i>USB Camera: Color Detection & Visual Servoing/Ranging</i> -----	214
3.4.2 <i>Dual-Bot RC Using ZigBee Broadcast</i> -----	233
3.4.3 <i>Dual-Bot Relay-RC Using ZigBee Broadcast</i> -----	248
3.4.4 <i>Dual-Bot RC Using Dual BT-210s with PySerial</i> -----	271
3.5 Using C/C++-----	279
3.5.1 <i>USB Camera: Color Detection & Visual Servoing/Ranging</i> -----	280
3.5.2 <i>Dual-Bot RC Using ZigBee Broadcast</i> -----	292
3.5.3 <i>Dual-Bot Relay-RC Using ZigBee Broadcast</i> -----	303
3.5.4 <i>Dual-Bot RC Using Dual BT-210s with BOOST.ASIO</i> -----	323
3.6 Wrapping up Volume 1 -----	336
Appendix A: Software Usage Tips-----	337
A.1 ROBOTIS Software-----	337
A.2 Thonny IDE -----	337
A.3 OpenCV Software for C/C++ Applications -----	338
A.4 Boost Libraries for C/C++ Applications-----	338
A.5 Visual Studio 2019 IDE for C/C++ Applications -----	339
A.6 Using Book Example Codes-----	344