



CARMINE NOVIELLO

# MASTERING STM32

A step-by-step guide to the most complete  
ARM Cortex-M platform, using the official  
STM32Cube development environment

**SECOND EDITION**

# Mastering STM32 - Second Edition

A step-by-step guide to the most complete ARM Cortex-M platform, using the official STM32Cube development environment

Carmine Noviello

This book is for sale at <http://leanpub.com/mastering-stm32-2nd>

This version was published on 2022-02-13



Leanpub

This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2015-2022 Carmine Noviello

# **Tweet This Book!**

Please help Carmine Noviello by spreading the word about this book on [Twitter!](#)

The suggested hashtag for this book is [#MasteringSTM32](#).

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

[#MasteringSTM32](#)

*To my wife Anna, who has always blindly supported me in all my projects*

*To my daughter Giulia, who completely upset my projects*

# Contents

<b>Preface</b> . . . . .	<b>i</b>
Who Is This Book For? . . . . .	ii
How to Integrate This Book? . . . . .	iii
How Is the Book Organized? . . . . .	iv
Differences With the First Edition . . . . .	vii
About the Author . . . . .	viii
Errata and Suggestions . . . . .	ix
Book Support . . . . .	ix
How to Help the Author . . . . .	ix
Copyright Disclaimer . . . . .	ix
Credits . . . . .	x
 <b>Acknowledgments to the First Edition</b> . . . . .	 <b>xi</b>
 <b>I Introduction</b> . . . . .	 <b>1</b>
<b>1. Introduction to STM32 MCU Portfolio</b> . . . . .	<b>2</b>
1.1 Introduction to ARM Based Processors . . . . .	2
1.1.1 Cortex and Cortex-M Based Processors . . . . .	4
1.1.1.1 Core Registers . . . . .	4
1.1.1.2 Memory Map . . . . .	7
1.1.1.3 Bit-Banding . . . . .	8
1.1.1.4 Thumb-2 and Memory Alignment . . . . .	11
1.1.1.5 Pipeline . . . . .	13
1.1.1.6 Interrupts and Exceptions Handling . . . . .	14
1.1.1.7 SysTimer . . . . .	16
1.1.1.8 Power Modes . . . . .	17
1.1.1.9 TrustZone™ . . . . .	18
1.1.1.10 CMSIS . . . . .	19
1.1.1.11 Effective Implementation of Cortex-M Features in the STM32 Portfolio . . . . .	20
1.2 Introduction to STM32 Microcontrollers . . . . .	21
1.2.1 Advantages of the STM32 Portfolio... . . . .	22

## CONTENTS

1.2.2	....And Its Drawbacks . . . . .	23
1.3	A Quick Look at the STM32 Subfamilies . . . . .	24
1.3.1	F0 . . . . .	26
1.3.2	F1 . . . . .	27
1.3.3	F2 . . . . .	28
1.3.4	F3 . . . . .	29
1.3.5	F4 . . . . .	31
1.3.6	F7 . . . . .	33
1.3.7	H7 . . . . .	35
1.3.8	L0 . . . . .	37
1.3.9	L1 . . . . .	38
1.3.10	L4 . . . . .	39
1.3.11	L4+ . . . . .	41
1.3.12	L5 . . . . .	42
1.3.13	U5 . . . . .	43
1.3.14	G0 . . . . .	45
1.3.15	G4 . . . . .	46
1.3.16	STM32WB . . . . .	48
1.3.17	STM32WL . . . . .	50
1.3.18	How to Select the Right MCU for You? . . . . .	51
1.4	The Nucleo Development Board . . . . .	54
<b>2.</b>	<b>Get In Touch With SM32CubeIDE . . . . .</b>	<b>60</b>
2.1	Why Choose STM32CubeIDE as Tool-Chain for STM32 . . . . .	60
2.1.1	Two Words About Eclipse... . . . .	62
2.1.2	... and GCC . . . . .	62
2.2	Downloading and Installing the STM32CubeIDE . . . . .	63
2.2.1	Windows - Installing the Tool-Chain . . . . .	64
2.2.2	Linux - Installing the Tool-Chain . . . . .	67
2.2.3	Mac - Installing the Tool-Chain . . . . .	68
2.3	STM32CubeIDE overview . . . . .	70
<b>3.</b>	<b>Hello, Nucleo! . . . . .</b>	<b>76</b>
3.1	Create a Project . . . . .	76
3.2	Adding Something Useful to the Generated Code . . . . .	79
3.3	Connecting the Nucleo to the PC . . . . .	84
3.3.1	ST-LINK Firmware Upgrade . . . . .	85
3.4	Flashing the Nucleo using STM32CubeProgrammer . . . . .	86
<b>4.</b>	<b>STM32CubeMX Tool . . . . .</b>	<b>89</b>
4.1	Introduction to CubeMX Tool . . . . .	89
4.1.1	Target Selection Wizard . . . . .	90
4.1.1.1	MCU/MPU Selector . . . . .	91
4.1.1.2	Board Selector . . . . .	92

## CONTENTS

4.1.1.3	Example Selector	92
4.1.1.4	Cross Selector	94
4.1.2	MCU and Middleware Configuration	94
4.1.2.1	Pinout View & Configuration	95
4.1.2.2	Clock Configuration View	100
4.1.3	Project Manager	102
4.1.4	Tools View	104
4.2	Understanding Project Structure	105
4.3	Downloading Book Source Code Examples	113
4.4	Management of STM32Cube Packages	115
<b>5.</b>	<b>Introduction to Debugging</b>	<b>117</b>
5.1	What is Behind a Debug Session	117
5.2	Debugging With STM32CubeIDE	119
5.2.1	Debug Configurations	122
5.3	I/O Retargeting	125
<b>II</b>	<b>Diving into the HAL</b>	<b>129</b>
<b>6.</b>	<b>GPIO Management</b>	<b>130</b>
6.1	STM32 Peripherals Mapping and HAL <i>Handlers</i>	130
6.2	GPIOs Configuration	135
6.2.1	GPIO Mode	136
6.2.2	GPIO Alternate Function	139
6.3	Driving a GPIO	140
6.4	De-initialize a GPIO	140
<b>7.</b>	<b>Interrupts Management</b>	<b>143</b>
7.1	NVIC Controller	143
7.1.1	Vector Table in STM32	144
7.2	Enabling Interrupts	148
7.2.1	External Lines and NVIC	149
7.2.2	Enabling Interrupts with CubeMX	153
7.3	Interrupt Lifecycle	155
7.4	Interrupt Priority Levels	159
7.4.1	Cortex-M0/0+	159
7.4.2	Cortex-M3/4/7/33	164
7.4.3	Setting Interrupt Priority in CubeMX	171
7.5	Interrupt Re-Entrancy	171
7.6	Mask All Interrupts at Once or on a Priority Basis	173
<b>8.</b>	<b>Universal Asynchronous Serial Communications</b>	<b>176</b>
8.1	Introduction to UARTs and USARTs	176

## CONTENTS

8.2	UART Initialization . . . . .	180
8.2.1	UART Configuration Using CubeMX . . . . .	187
8.3	UART Communication in <i>Polling Mode</i> . . . . .	188
8.3.1	Installing a Terminal Emulator in Eclipse . . . . .	192
8.4	UART Communication in <i>Interrupt Mode</i> . . . . .	193
8.4.1	UART Related Interrupts . . . . .	194
8.5	Error Management . . . . .	201
8.6	List of Available Callbacks in the HAL_UART Module . . . . .	202
<b>9.</b>	<b>DMA Management . . . . .</b>	<b>205</b>
9.1	Introduction to DMA . . . . .	205
9.1.1	The Need of a DMA and the Role of the Internal Buses . . . . .	206
9.1.2	The DMA Controller . . . . .	209
9.1.2.1	The DMA Implementation in F0/F1/F3/L0/L1/L4 MCUs . . . . .	210
9.1.2.2	The DMA Implementation in F2/F4/F7 MCUs . . . . .	214
9.1.2.3	The DMA Implementation in G0/G4/L4+/L5/H7 MCUs . . . . .	217
9.2	HAL_DMA Module . . . . .	220
9.2.1	DMA_HandleTypeDef in F0/F1/F3/L0/L1/L4 HALs . . . . .	220
9.2.2	DMA Configuration in G0/G4/L4+/L5/H7 HALs . . . . .	223
9.2.3	DMA_HandleTypeDef in F2/F4/F7 HALs . . . . .	226
9.2.4	How to Perform DMA Transfers in Polling Mode . . . . .	229
9.2.5	How to Perform DMA Transfers in Interrupt Mode . . . . .	232
9.2.6	Using the HAL_UART Module with DMA Mode Transfers . . . . .	233
9.2.7	Programming the DMAMUX With the CubeHAL . . . . .	236
9.2.8	Miscellaneous Functions From HAL_DMA and HAL_DMA_Ex Modules . . . . .	237
9.3	Using CubeMX to Configure DMA Requests . . . . .	238
9.4	Correct Memory Allocation of DMA Buffers . . . . .	239
9.5	A Case Study: The DMA <i>Memory-To-Memory</i> Transfer Performance Analysis . . . . .	240
<b>10.</b>	<b>Clock Tree . . . . .</b>	<b>245</b>
10.1	Clock Distribution . . . . .	245
10.1.1	Overview of the STM32 Clock Tree . . . . .	246
10.1.1.1	The Multispeed Internal RC Oscillator in STM32L/U Families . . . . .	250
10.1.2	Configuring Clock Tree Using CubeMX . . . . .	251
10.1.3	Clock Source Options in Nucleo Boards . . . . .	253
10.1.3.1	Clock Source in Nucleo-64 rev. MB1136 (older ones with ST-LINK V2.1) . . . . .	254
10.1.3.1.1	OSC Clock Supply . . . . .	254
10.1.3.1.2	OSC 32kHz Clock Supply . . . . .	255
10.1.3.2	Clock Source in Nucleo-64 rev. MB1367 (newer ones with ST-LINK v3) . . . . .	256
10.1.3.2.1	OSC Clock Supply . . . . .	256
10.1.3.2.2	OSC 32kHz Clock Supply . . . . .	257



## CONTENTS

10.2	Overview of the HAL_RCC Module . . . . .	257
10.2.1	Compute the Clock Frequency at Run-Time . . . . .	259
10.2.2	Enabling the <i>Master Clock Output</i> . . . . .	260
10.2.3	Enabling the <i>Clock Security System</i> . . . . .	260
10.3	HSI Calibration . . . . .	261
<b>11.</b>	<b>Timers . . . . .</b>	<b>263</b>
11.1	Introduction to Timers . . . . .	263
11.1.1	Timer Categories in an STM32 MCU . . . . .	264
11.1.2	Effective Availability of Timers in the STM32 Portfolio . . . . .	266
11.2	Basic Timers . . . . .	269
11.2.1	Using Timers in <i>Interrupt Mode</i> . . . . .	272
11.2.1.1	Time Base Generation in <i>Advanced</i> Timers . . . . .	274
11.2.2	Using Timers in <i>Polling Mode</i> . . . . .	274
11.2.3	Using Timers in <i>DMA Mode</i> . . . . .	275
11.2.4	Stopping a Timer . . . . .	278
11.2.5	Using CubeMX to Configure a <i>Basic Timer</i> . . . . .	278
11.3	General Purpose Timers . . . . .	278
11.3.1	Time Base Generator with External Clock Sources . . . . .	279
11.3.1.1	External Clock Mode 2 . . . . .	281
11.3.1.2	External Clock Mode 1 . . . . .	284
11.3.1.3	Using CubeMX to Configure the Source Clock of a <i>General Purpose</i> Timer . . . . .	289
11.3.2	Master/Slave Synchronization Modes . . . . .	290
11.3.2.1	Enable Trigger-Related Interrupts . . . . .	296
11.3.2.2	Using CubeMX to Configure the Master/Slave Synchronization . . . . .	296
11.3.3	Generate Timer-Related Events by Software . . . . .	297
11.3.4	Counting Modes . . . . .	299
11.3.5	Input Capture Mode . . . . .	300
11.3.5.1	Using CubeMX to Configure the Input Capture Mode . . . . .	307
11.3.6	Output Compare Mode . . . . .	308
11.3.6.1	Using CubeMX to Configure the Output Compare Mode . . . . .	313
11.3.7	Pulse-Width Generation . . . . .	313
11.3.7.1	Generating a Sinusoidal Wave Using PWM . . . . .	317
11.3.7.2	Using CubeMX to Configure the PWM Mode . . . . .	322
11.3.8	One Pulse Mode . . . . .	322
11.3.8.1	Using CubeMX to Configure the OPM Mode . . . . .	325
11.3.9	Encoder Mode . . . . .	325
11.3.9.1	Using CubeMX to Configure the <i>Encoder Mode</i> . . . . .	331
11.3.10	Other Features Available in <i>General Purpose</i> and <i>Advanced</i> Timers . . . . .	331
11.3.10.1	<i>Hall Sensor</i> Mode . . . . .	332

## CONTENTS

11.3.10.2	Combined Three-Phase PWM Mode and Other Motor-Control Related Features . . . . .	332
11.3.10.3	Break Input and Locking of Timer Registers . . . . .	333
11.3.10.4	Preloading of Auto-Reload Register . . . . .	333
11.3.11	Debugging and Timers . . . . .	334
11.4	SysTick Timer . . . . .	335
11.4.1	Use Another Timer as System Timebase Source . . . . .	336
11.5	A Case Study: How to Precisely Measure Microseconds with STM32 MCUs . . . .	337
<b>12.</b>	<b>Analog-To-Digital Conversion . . . . .</b>	<b>343</b>
12.1	Introduction to SAR ADC . . . . .	343
12.2	HAL_ADC Module . . . . .	348
12.2.1	Conversion Modes . . . . .	350
12.2.1.1	Single-Channel, Single Conversion Mode . . . . .	350
12.2.1.2	Scan Single Conversion Mode . . . . .	351
12.2.1.3	Single-Channel, Continuous Conversion Mode . . . . .	351
12.2.1.4	Scan Continuous Conversion Mode . . . . .	352
12.2.1.5	Injected Conversion Mode . . . . .	352
12.2.1.6	Dual Modes . . . . .	353
12.2.2	Channel Selection . . . . .	353
12.2.3	ADC Resolution and Conversion Speed . . . . .	355
12.2.4	A/D Conversions in Polling Mode . . . . .	356
12.2.5	A/D Conversions in Interrupt Mode . . . . .	360
12.2.6	A/D Conversions in DMA Mode . . . . .	361
12.2.6.1	Convert Multiple Times the Same Channel in DMA Mode . . . . .	365
12.2.6.2	Multiple and not Continuous Conversions in DMA Mode . . . . .	365
12.2.6.3	Continuous Conversions in DMA Mode . . . . .	365
12.2.7	Errors Management . . . . .	365
12.2.8	Timer-Driven Conversions . . . . .	366
12.2.9	Conversions Driven by External Events . . . . .	370
12.2.10	ADC Calibration . . . . .	370
12.3	Using CubeMX to Configure ADC Peripheral . . . . .	371
<b>13.</b>	<b>Digital-To-Analog Conversion . . . . .</b>	<b>373</b>
13.1	Introduction to the DAC Peripheral . . . . .	373
13.2	HAL_DAC Module . . . . .	375
13.2.1	Driving the DAC Manually . . . . .	377
13.2.2	Driving the DAC in DMA Mode Using a Timer . . . . .	379
13.2.3	Triangular Wave Generation . . . . .	383
13.2.4	Noise Wave Generation . . . . .	384
<b>14.</b>	<b>I<sup>2</sup>C . . . . .</b>	<b>385</b>
14.1	Introduction to the I <sup>2</sup> C specification . . . . .	385
14.1.1	The I <sup>2</sup> C Protocol . . . . .	387

## CONTENTS

14.1.1.1	START and STOP Condition . . . . .	388
14.1.1.2	Byte Format . . . . .	388
14.1.1.3	Address Frame . . . . .	388
14.1.1.4	Acknowledge (ACK) and Not Acknowledge (NACK) . . . . .	389
14.1.1.5	Data Frames . . . . .	389
14.1.1.6	Combined Transactions . . . . .	390
14.1.1.7	Clock Stretching . . . . .	391
14.1.2	Availability of I <sup>2</sup> C Peripherals in STM32 MCUs . . . . .	391
14.2	HAL_I2C Module . . . . .	392
14.2.1	Using the I <sup>2</sup> C Peripheral in <i>Master Mode</i> . . . . .	396
14.2.1.1	I/O MEM Operations . . . . .	403
14.2.1.2	Combined Transactions . . . . .	405
14.2.1.3	A Note About the Clock Configuration in STM32F0/L0/L4 families . . . . .	406
14.2.2	Using the I <sup>2</sup> C Peripheral in <i>Slave Mode</i> . . . . .	406
14.3	Using CubeMX to Configure the I <sup>2</sup> C Peripheral . . . . .	412
<b>15.</b>	<b>SPI . . . . .</b>	<b>414</b>
15.1	Introduction to the SPI Specification . . . . .	414
15.1.1	Clock Polarity and Phase . . . . .	417
15.1.2	Slave Select Signal Management . . . . .	418
15.1.3	SPI <i>TI Mode</i> . . . . .	418
15.1.4	Availability of SPI Peripherals in STM32 MCUs . . . . .	419
15.2	HAL_SPI Module . . . . .	420
15.2.1	Exchanging Messages Using SPI Peripheral . . . . .	422
15.2.2	Maximum Transmission Frequency Reachable using the CubeHAL . . . . .	424
15.3	Using CubeMX to Configure SPI Peripheral . . . . .	424
<b>16.</b>	<b>Cyclic Redundancy Check . . . . .</b>	<b>425</b>
16.1	Introduction to CRC Computing . . . . .	425
16.1.1	CRC Calculation in STM32F1/F2/F4/L1 MCUs . . . . .	428
16.1.2	CRC Peripheral in STM32F0/F3/F7/L0/L4/L5/G0/G4 MCUs . . . . .	430
16.2	HAL_CRC Module . . . . .	431
<b>17.</b>	<b>IWDG and WWDG Timers . . . . .</b>	<b>434</b>
17.1	The <i>Independent Watchdog</i> Timer . . . . .	434
17.1.1	Using the CubeHAL to Program IWDG Timer . . . . .	435
17.2	The <i>System Window Watchdog</i> Timer . . . . .	436
17.2.1	Using the CubeHAL to Program WWDG Timer . . . . .	438
17.3	Detecting a System Reset Caused by a Watchdog Timer . . . . .	439
17.4	Freezing Watchdog Timers During a Debug Session . . . . .	440
17.5	Selecting the Right Watchdog Timer for Your Application . . . . .	440
<b>18.</b>	<b>Real-Time Clock . . . . .</b>	<b>441</b>

## CONTENTS

18.1	Introduction to the RTC Peripheral . . . . .	441
18.2	HAL_RTC Module . . . . .	443
18.2.1	Setting and Retrieving the Current Date/Time . . . . .	444
18.2.1.1	Correct Way to Read Date/Time Values . . . . .	446
18.2.2	Configuring Alarms . . . . .	447
18.2.3	Periodic Wakeup Unit . . . . .	449
18.2.4	Timestamp Generation and Tamper Detection . . . . .	451
18.2.5	RTC Calibration . . . . .	451
18.2.5.1	RTC Coarse Calibration . . . . .	452
18.2.5.2	RTC Smooth Calibration . . . . .	452
18.2.5.3	Reference Clock Detection . . . . .	454
18.3	Using the Backup SRAM . . . . .	454

## III Advanced topics . . . . . 456

19.	Power Management . . . . .	457
19.1	Power Management in Cortex-M Based MCUs . . . . .	457
19.2	How Cortex-M MCUs Handle <i>Run</i> and <i>Sleep</i> Modes . . . . .	458
19.2.1	Entering/exiting sleep modes . . . . .	461
19.2.1.1	Sleep-On-Exit . . . . .	463
19.2.2	<i>Sleep</i> Modes in Cortex-M Based MCUs . . . . .	463
19.3	Power Management in STM32F Microcontrollers . . . . .	464
19.3.1	Power Sources . . . . .	464
19.3.2	Power Modes . . . . .	465
19.3.2.1	Run Mode . . . . .	466
19.3.2.1.1	Dynamic Voltage Scaling in STM32F4/F7 MCUs . . . . .	467
19.3.2.1.2	Over/Under-Drive Mode in STM32F4/F7 MCUs . . . . .	468
19.3.2.2	Sleep Mode . . . . .	468
19.3.2.3	Stop Mode . . . . .	469
19.3.2.4	Standby Mode . . . . .	470
19.3.2.5	Low-Power Modes Example . . . . .	470
19.3.3	An Important Warning for STM32F1 Microcontrollers . . . . .	474
19.4	Power Management in STM32L/G Microcontrollers . . . . .	476
19.4.1	Power Sources . . . . .	476
19.4.2	Power Modes . . . . .	478
19.4.2.1	Run Modes . . . . .	478
19.4.2.2	Sleep Modes . . . . .	480
19.4.2.2.1	Batch Acquisition Mode . . . . .	481
19.4.2.3	Stop Modes . . . . .	481
19.4.2.4	Standby Modes . . . . .	482
19.4.2.5	Shutdown Mode . . . . .	483
19.4.3	Power Modes Transitions . . . . .	483

## CONTENTS

19.4.4	Low-Power Peripherals . . . . .	484
19.4.4.1	LPUART . . . . .	484
19.4.4.2	LPTIM . . . . .	485
19.4.4.3	LPGPIO . . . . .	485
19.4.4.4	LPDMA . . . . .	486
19.5	Power Supply Supervisors . . . . .	486
19.6	Debugging in Low-Power Modes . . . . .	487
19.7	Using the CubeMX Power Consumption Calculator . . . . .	487
19.8	A Case Study: Using Watchdog Timers With Low-Power Modes . . . . .	489
<b>20.</b>	<b>Memory layout . . . . .</b>	<b>490</b>
20.1	The STM32 Memory Layout Model . . . . .	490
20.1.1	Flash Memory Typical Organization . . . . .	490
20.1.2	SRAM Memory Typical Organization . . . . .	492
20.1.3	Understanding Compilation and Linking Processes . . . . .	493
20.2	The Really Minimal STM32 Application . . . . .	495
20.2.1	ELF Binary File Inspection . . . . .	499
20.2.2	.data and .bss Sections Initialization . . . . .	501
20.2.2.1	A Word About the COMMON Section . . . . .	508
20.2.3	.rodata Section . . . . .	509
20.2.4	Stack and Heap Regions . . . . .	511
20.2.5	Checking the Size of Heap and Stack at Compile-Time . . . . .	514
20.2.6	Differences With the Tool-Chain Script Files . . . . .	515
20.3	How to Use the CCM Memory . . . . .	520
20.3.1	Relocating the <i>vector table</i> in CCM Memory . . . . .	524
20.4	How to Use the MPU in Cortex-M0+/3/4/7 Based STM32 MCUs . . . . .	527
20.4.1	Programming the MPU With the CubeHAL . . . . .	531
<b>21.</b>	<b>Flash Memory Management . . . . .</b>	<b>534</b>
21.1	Introduction to STM32 Flash Memory . . . . .	534
21.2	The HAL_FLASH Module . . . . .	537
21.2.1	Flash Memory Unlocking . . . . .	538
21.2.2	Flash Memory Erasing . . . . .	538
21.2.3	Flash Memory Programming . . . . .	540
21.2.4	Flash Read Access During Programming and Erasing . . . . .	541
21.3	Option Bytes . . . . .	541
21.3.1	Flash Memory Read Protection . . . . .	543
21.4	Optional OTP and True-EEPROM Memories . . . . .	545
21.5	Flash Read Latency and the ART™ Accelerator . . . . .	547
21.5.1	The Role of the TCM Memories in STM32F7/H7 MCUs . . . . .	549
21.5.1.1	How to Access Flash Memory Through the TCM Interface . . . . .	555
21.5.1.2	Using CubeMX to Configure Flash Memory Interface . . . . .	556
<b>22.</b>	<b>Bootling Process . . . . .</b>	<b>558</b>

## CONTENTS

22.1	The Cortex-M Unified Memory Layout and the Booting Process . . . . .	558
22.1.1	Software <i>Physical Remap</i> . . . . .	559
22.1.2	Vector Table Relocation . . . . .	560
22.1.3	Running the Firmware From SRAM Using the STM32CubeIDE . . . . .	562
22.2	Integrated STM32 Bootloader . . . . .	563
22.2.1	Starting the STM32 Bootloader from the On-Board Firmware . . . . .	566
22.2.2	The Booting Sequence in the STM32CubeIDE Tool-chain . . . . .	567
22.3	Developing a Custom Bootloader . . . . .	568
22.3.1	<i>Vector Table</i> Relocation in STM32F0 Microcontrollers . . . . .	579
22.3.2	How to Use the <code>flasher.py</code> Tool . . . . .	581
<b>23.</b>	<b>Running FreeRTOS . . . . .</b>	<b>584</b>
23.1	Understanding the Concepts Underlying an RTOS . . . . .	585
23.2	Configuring FreeRTOS and the CMSIS-RTOS v2 Wrapper . . . . .	591
23.2.1	The FreeRTOS Source Tree . . . . .	592
23.2.1.1	How to Configure FreeRTOS Using CubeMX . . . . .	593
23.3	Thread Management . . . . .	594
23.3.1	Thread States . . . . .	597
23.3.2	Thread Priorities and Scheduling Policies . . . . .	598
23.3.3	Voluntary Release of the Control . . . . .	602
23.3.4	The <i>idle</i> Thread . . . . .	602
23.4	Memory Allocation and Management . . . . .	604
23.4.1	Dynamic Memory Allocation Model . . . . .	604
23.4.1.1	<code>heap_1.c</code> . . . . .	605
23.4.1.2	<code>heap_2.c</code> . . . . .	606
23.4.1.3	<code>heap_3.c</code> . . . . .	606
23.4.1.4	<code>heap_4.c</code> . . . . .	607
23.4.1.5	<code>heap_5.c</code> . . . . .	607
23.4.1.6	FreeRTOS Heap Definition . . . . .	608
23.4.2	Static Memory Allocation Model . . . . .	608
23.4.2.1	<i>idle</i> Thread Allocation with Static Memory Allocation Model . . . . .	609
23.4.3	FreeRTOS and the C <code>stdlib</code> . . . . .	609
23.4.3.1	How to Configure <code>newlib</code> to Handle Concurrency with FreeRTOS . . . . .	610
23.4.3.2	How to Use <code>malloc()</code> and <code>malloc()</code> -dependant <code>newlib</code> Functions With FreeRTOS . . . . .	613
23.4.3.3	STM32CubeMX Approach to Thread-Safety . . . . .	621
23.4.4	Memory Pools . . . . .	622
23.4.5	Stack Overflow Detection . . . . .	625
23.5	Synchronization Primitives . . . . .	627
23.5.1	Message Queues . . . . .	627
23.5.2	Semaphores . . . . .	630
23.5.3	Event and Thread Flags . . . . .	633

## CONTENTS

23.6	Resources Management and Mutual Exclusion . . . . .	637
23.6.1	Mutexes . . . . .	637
23.6.1.1	The Priority Inversion Problem . . . . .	639
23.6.1.2	Recursive Mutexes . . . . .	640
23.6.2	Critical Sections . . . . .	640
23.6.3	Interrupt Management With an RTOS . . . . .	641
23.6.3.1	FreeRTOS API and Interrupt Priorities . . . . .	642
23.7	Software Timers . . . . .	643
23.7.1	How FreeRTOS Manages Timers . . . . .	645
23.8	A Case Study: Low-Power Management With an RTOS . . . . .	645
23.8.1	The <i>idle</i> Thread Hook . . . . .	646
23.8.2	The Tickless Mode in FreeRTOS . . . . .	647
23.8.2.1	A Schema for the <i>tickless</i> Mode . . . . .	649
23.8.2.2	A Custom <i>tickless</i> Mode Policy . . . . .	652
23.9	Debugging Features . . . . .	660
23.9.1	configASSERT() Macro . . . . .	660
23.9.2	Run-Time Statistics and Thread State Information . . . . .	661
23.9.3	FreeRTOS Debugging in STM32CubeIDE . . . . .	665
23.9.4	FreeRTOS Kernel-Aware Debugging in STM32CubeIDE . . . . .	668
23.10	Alternatives to FreeRTOS . . . . .	670
23.10.1	AzureRTOS . . . . .	670
23.10.2	ChibiOS . . . . .	670
23.10.3	Contiki OS . . . . .	671
23.10.4	OpenRTOS . . . . .	671
<b>24.</b>	<b>Advanced Debugging Techniques . . . . .</b>	<b>673</b>
24.1	Understanding Cortex-M Fault-Related Exceptions . . . . .	673
24.1.1	The Cortex-M Exception Entrance Sequence and the ARM Calling Convention . . . . .	675
24.1.1.1	How to Interpret the Content of the LR Register on Excep- tion Entrance . . . . .	680
24.1.2	Fault Exceptions and Faults Analysis . . . . .	681
24.1.2.1	<i>Memory Management</i> Exception . . . . .	682
24.1.2.2	<i>Bus Fault</i> Exception . . . . .	682
24.1.2.3	<i>Usage Fault</i> Exception . . . . .	683
24.1.2.4	<i>Hard Fault</i> Exception . . . . .	684
24.1.2.5	<i>Secure Fault</i> Exception . . . . .	685
24.1.2.6	Enabling Optional Fault Handlers . . . . .	685
24.1.2.7	Fault Analysis in Cortex-M0/0+ Based Processors . . . . .	686
24.2	STM32CubeIDE Advanced Debugging Features . . . . .	686
24.2.1	Expressions and Live Expressions . . . . .	686
24.2.1.1	Memory Monitors . . . . .	688
24.2.2	Watchpoints . . . . .	689

## CONTENTS

24.2.3	Instruction Stepping Mode . . . . .	690
24.2.4	SFRs View . . . . .	690
24.2.5	Fault Analyzer . . . . .	691
24.2.5.1	Tracing Fault-Related Registers Without the IDE Support . . . . .	693
24.2.6	Build Analyzer . . . . .	696
24.2.7	Static Stack Analyzer . . . . .	697
24.3	Serial Wire Viewer Tracing . . . . .	698
24.3.1	Enabling SWV Debugging . . . . .	700
24.3.2	Configuring SWV . . . . .	701
24.3.3	SWV Views . . . . .	703
24.3.3.1	SWV Trace Log . . . . .	704
24.3.3.2	SWV Exception Trace Log . . . . .	704
24.3.3.3	SWV Data Trace . . . . .	705
24.3.3.4	SWV Data Trace Timeline Graph . . . . .	706
24.3.3.5	SWV ITM Data Console . . . . .	707
24.3.3.6	SWV Statistical Profiling . . . . .	708
24.4	Debugging Aids from the CubeHAL . . . . .	709
24.5	External Debuggers . . . . .	709
24.6	Debugging two Nucleo Boards Simultaneously . . . . .	711
24.7	ARM Semihosting . . . . .	712
24.7.1	Enable Semihosting on a Project . . . . .	713
24.7.2	Semihosting Drawbacks . . . . .	715
24.7.3	Understanding How Semihosting Works . . . . .	716
<b>25.</b>	<b>FAT Filesystem . . . . .</b>	<b>720</b>
25.1	Introduction to FatFs Library . . . . .	720
25.1.1	Adding FatFs Library in Your Projects . . . . .	723
25.1.1.1	The <i>Generic Disk Interface</i> API . . . . .	724
25.1.1.2	The Implementation of a Driver to Access SD Cards in SPI Mode . . . . .	725
25.1.2	Relevant FatFs Structures and Functions . . . . .	725
25.1.2.1	Mounting a Filesystem . . . . .	725
25.1.2.2	Opening a File . . . . .	726
25.1.2.3	Reading From/Writing into a File . . . . .	727
25.1.2.4	Creating and Opening a Directory . . . . .	728
25.1.3	How to Configure the FatFs Library . . . . .	731
<b>26.</b>	<b>Develop IoT Applications . . . . .</b>	<b>733</b>
26.1	Solutions Offered by STM to Develop IoT Applications . . . . .	734
26.2	The W5500 Ethernet Controller . . . . .	736
26.2.1	How to Use the W5500 Shield and the <code>ioLibrary_Driver</code> Module . . . . .	740
26.2.1.1	Configuring the SPI Interface . . . . .	742
26.2.1.2	Configuring the Socket Buffers and the Network Interface . . . . .	743



## CONTENTS

26.2.2	Socket APIs . . . . .	745
26.2.2.1	Handling Sockets in TCP Mode . . . . .	746
26.2.2.2	Handling Sockets in UDP Mode . . . . .	747
26.2.3	I/O Retargeting to a TCP/IP Socket . . . . .	748
26.2.4	Building up an HTTP Server . . . . .	749
26.2.4.1	A Web-Based Oscilloscope . . . . .	752
<b>27.</b>	<b>Universal Serial Bus . . . . .</b>	<b>765</b>
27.1	USB 2.0 Specification Overview . . . . .	766
27.1.1	The “Before-To-Die” Guide to USB . . . . .	766
27.1.2	USB Physical Architecture Overview . . . . .	769
27.1.3	USB Logical Architecture Overview . . . . .	772
27.1.3.1	Device States . . . . .	772
27.1.3.2	Communication Endpoints . . . . .	774
27.1.4	USB 2.0 Communication Protocol Overview . . . . .	777
27.1.4.1	Packet Types . . . . .	778
27.1.4.2	Transaction Types . . . . .	780
27.1.4.2.1	Control Transactions . . . . .	780
27.1.4.2.2	IN/OUT Transactions . . . . .	784
27.1.4.3	Device and Interface Descriptors . . . . .	785
27.1.4.3.1	Device Descriptors . . . . .	786
27.1.4.3.2	Configuration Descriptors . . . . .	788
27.1.4.3.3	Interface Descriptors . . . . .	789
27.1.4.3.4	Endpoint Descriptors . . . . .	789
27.1.4.3.5	String Descriptors . . . . .	791
27.1.4.4	USB Classes . . . . .	791
27.2	STM32 USB Device Library . . . . .	793
27.2.1	Understanding Generated Code . . . . .	794
27.2.2	USB Initialization Sequence . . . . .	798
27.2.3	USB Enumeration Sequence . . . . .	801
27.2.4	The USB CDC Class . . . . .	804
27.2.4.1	USB CDC Descriptors . . . . .	805
27.2.4.2	USB CDC Class Initialization . . . . .	809
27.2.4.3	USB CDC Class Operations . . . . .	810
27.3	Building Custom USB Devices . . . . .	815
27.3.1	The USB HID Class . . . . .	817
27.3.1.1	USB HID Descriptors . . . . .	818
27.3.1.2	Overview of the <i>Report Descriptor</i> . . . . .	820
27.3.1.3	USB HID Class-Specific Requests . . . . .	824
27.3.2	Building a Vendor-Specific USB HID Device . . . . .	825
27.4	Debugging USB Devices . . . . .	837
27.4.1	Software Sniffers and Analyzers . . . . .	837
27.4.2	USB Hardware Analyzers . . . . .	837

## CONTENTS

27.5	Optimizing the STM32 USB Device Library . . . . .	838
27.6	Going to the Market . . . . .	839
<b>28.</b>	<b>Getting Started with a New Design . . . . .</b>	<b>842</b>
28.1	Hardware Design . . . . .	842
28.1.1	PCB Layer Stack-Up . . . . .	843
28.1.2	MCU Package . . . . .	844
28.1.3	Decoupling of Power-Supply Pins . . . . .	845
28.1.4	Clocks . . . . .	847
28.1.5	Filtering of RESET Pin . . . . .	848
28.1.6	Debug Port . . . . .	848
28.1.7	Boot Mode . . . . .	850
28.1.8	Pay attention to “pin-to-pin” Compatibility... . . . .	851
28.1.9	...And to Selecting the Right Peripherals . . . . .	852
28.1.10	The Role of CubeMX During the Board Design Stage . . . . .	852
28.1.11	Board Layout Strategies . . . . .	856
28.2	Software Design . . . . .	857
28.2.1	Generating the binary image for production . . . . .	857

## **Appendix . . . . . 860**

### **A. Miscellaneous HAL functions and STM32 features . . . . . 861**

Force MCU reset from the firmware . . . . .	861
STM32 96-bit Unique CPU ID . . . . .	861

### **B. Troubleshooting Guide . . . . . 863**

STM32CubeIDE Issues . . . . .	863
Debugging Continuously Breaks at Every Instruction During a Debug Session . . . . .	863
The Step-by-Step Debugging is Really Slow . . . . .	863
The Firmware Works Only Under a Debug Session . . . . .	864
STM32 Related Issues . . . . .	864
The Microcontroller Does Not Boot Correctly . . . . .	864
It is Not Possible to Flash or to Debug the MCU . . . . .	866

### **C. Nucleo pin-out . . . . . 868**

Nucleo-G474RE . . . . .	869
Arduino compatible headers . . . . .	869
Morpho headers . . . . .	869
Nucleo-F446RE . . . . .	870
Arduino compatible headers . . . . .	870
Morpho headers . . . . .	870
Nucleo-F401RE . . . . .	871
Arduino compatible headers . . . . .	871

## CONTENTS

Morpho headers . . . . .	871
Nucleo-F303RE . . . . .	872
Arduino compatible headers . . . . .	872
Morpho headers . . . . .	872
Nucleo-F103RB . . . . .	873
Arduino compatible headers . . . . .	873
Morpho headers . . . . .	873
Nucleo-F072RB . . . . .	874
Arduino compatible headers . . . . .	874
Morpho headers . . . . .	874
Nucleo-L476RG . . . . .	875
Arduino compatible headers . . . . .	875
Morpho headers . . . . .	875
Nucleo-L152RE . . . . .	876
Arduino compatible headers . . . . .	876
Morpho headers . . . . .	876
Nucleo-L073R8 . . . . .	877
Arduino compatible headers . . . . .	877
Morpho headers . . . . .	877
<b>D. Differences with the 1st edition . . . . .</b>	<b>878</b>
Chapter 1 . . . . .	878
Chapter 2 . . . . .	878
Chapter 3 and 4 . . . . .	878
Chapter 5 . . . . .	878
Chapter 6 . . . . .	879
Chapter 7 . . . . .	879
Chapter 8 . . . . .	879
Chapter 9 . . . . .	879
Chapter 10 . . . . .	879
Chapter 11 . . . . .	879
Chapter 12-22 . . . . .	880
Chapter 23 . . . . .	880
Chapter 24 . . . . .	880
Chapter 25-26 . . . . .	880
Chapter 27 . . . . .	880
Chapter 28 . . . . .	880