



CARMINE NOVIELLO

MASTERING STM32

A step-by-step guide to the most complete
ARM Cortex-M platform, using a free
and powerful development environment
based on Eclipse and GCC

Mastering STM32

A step-by-step guide to the most complete ARM Cortex-M platform, using a free and powerful development environment based on Eclipse and GCC

Carmine Noviello

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

This version was published on 2017-05-02



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 - 2017 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:

<https://twitter.com/search?q=#MasteringSTM32>

Contents

Preface	i
Why Did I Write the Book?	i
Who Is This Book For?	ii
How to Integrate This Book?	iii
How Is the Book Organized?	iv
About the Author	vii
Errata and Suggestions	viii
Book Support	viii
How to Help the Author	viii
Copyright Disclaimer	ix
Credits	ix
Acknowledgments	x
I Introduction	1
1. Introduction to STM32 MCU Portfolio	2
1.1 Introduction to ARM Based Processors	2
1.1.1 Cortex and Cortex-M Based Processors	4
1.1.1.1 Core Registers	5
1.1.1.2 Memory Map	7
1.1.1.3 Bit-Banding	9
1.1.1.4 Thumb-2 and Memory Alignment	12
1.1.1.5 Pipeline	13
1.1.1.6 Interrupts and Exceptions Handling	15
1.1.1.7 SysTimer	17
1.1.1.8 Power Modes	17
1.1.1.9 CMSIS	19
1.1.1.10 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
1.2.2and Its drawbacks	23
1.3 A Quick Look at the STM32 Subfamilies	24

CONTENTS

1.3.1	F0	26
1.3.2	F1	27
1.3.3	F2	28
1.3.4	F3	30
1.3.5	F4	32
1.3.6	F7	33
1.3.7	H7	34
1.3.8	L0	35
1.3.9	L1	36
1.3.10	L4	38
1.3.11	W and J STM32 MCUs	39
1.3.12	How to Select the Right MCU for You?	39
1.4	The Nucleo Development Board	42
2.	Setting-Up the Tool-Chain	48
2.1	Why Choose Eclipse/GCC as Tool-Chain for STM32	49
2.1.1	Two Words About Eclipse...	50
2.1.2	... and GCC	50
2.2	Windows - Installing the Tool-Chain	51
2.2.1	Windows - Eclipse Installation	52
2.2.2	Windows - Eclipse Plug-Ins Installation	53
2.2.3	Windows - GCC ARM Embedded Installation	59
2.2.4	Windows - Build Tools Installation	61
2.2.5	Windows - OpenOCD Installation	61
2.2.6	Windows - ST Tools and Drivers Installation	61
2.2.6.1	Windows - ST-LINK Firmware Upgrade	62
2.3	Linux - Installing the Tool-Chain	63
2.3.1	Linux - Install i386 Run-Time Libraries on a 64-bit Ubuntu	64
2.3.2	Linux - Java Installation	64
2.3.3	Linux - Eclipse Installation	64
2.3.4	Linux - Eclipse Plug-Ins Installation	66
2.3.5	Linux - GCC ARM Embedded Installation	72
2.3.6	Linux - Nucleo Drivers Installation	72
2.3.6.1	Linux - ST-LINK Firmware Upgrade	73
2.3.7	Linux - OpenOCD Installation	73
2.3.8	Linux - STM32CubeMX Tool Installation	76
2.3.9	Linux - QSTLink2 Installation	78
2.4	Mac - Installing the Tool-Chain	79
2.4.1	Mac - Eclipse Installation	80
2.4.2	Mac - Eclipse Plug-Ins Installation	82
2.4.3	Mac - GCC ARM Embedded Installation	87
2.4.4	Mac - Nucleo Drivers Installation	87
2.4.4.1	Mac - ST-LINK Firmware Upgrade	88

CONTENTS

2.4.5	Mac – OpenOCD Installation	88
2.4.6	Mac STM32CubeMX Tool Installation	90
2.4.7	Mac - stlink by texane Installation	92
3.	Hello, Nucleo!	94
3.1	Get in Touch With the Eclipse IDE	94
3.2	Create a Project	98
3.3	Connecting the Nucleo to the PC	105
3.4	Flashing the Nucleo	106
3.4.1	Windows	106
3.4.2	Linux	107
3.4.3	Mac OSX	109
3.5	Understanding the Generated Code	110
4.	STM32CubeMX Tool	113
4.1	Introduction to CubeMX Tool	113
4.1.1	Pinout View	117
4.1.1.1	Chip View	117
4.1.1.2	IP Tree Pane	119
4.1.2	Clock View	120
4.1.3	Configuration View	121
4.1.4	Power Consumption Calculator View	122
4.2	Project Generation	123
4.2.1	Generate C Project with CubeMX	124
4.2.1.1	Understanding Generated Code	125
4.2.2	Create Eclipse Project	127
4.2.3	Importing Generated Files Into the Eclipse Project Manually	130
4.2.4	Importing Files Generated With CubeMX Into the Eclipse Project Automatically	135
4.3	Understanding Generated Application Code	136
4.3.1	Add Something Useful to the Firmware	141
4.4	Downloading Book Source Code Examples	142
5.	Introduction to Debugging	146
5.1	Getting Started With OpenOCD	146
5.1.1	Launching OpenOCD	147
5.1.1.1	Launching OpenOCD on Windows	148
5.1.1.2	Launching OpenOCD on Linux and MacOS X.	149
5.1.2	Connecting to the OpenOCD Telnet Console	151
5.1.3	Configuring Eclipse	152
5.1.4	Debugging in Eclipse	159
5.2	ARM Semihosting	164
5.2.1	Enable Semihosting on a New Project	164

CONTENTS

5.2.1.1	Using Semihosting With C Standard Library	167
5.2.2	Enable Semihosting on an Existing Project	170
5.2.3	Semihosting Drawbacks	171
5.2.4	Understanding How Semihosting Works	171
II	Diving into the HAL	176
6.	GPIO Management	177
6.1	STM32 Peripherals Mapping and HAL <i>Handlers</i>	177
6.2	GPIOs Configuration	182
6.2.1	GPIO Mode	184
6.2.2	GPIO Alternate Function	186
6.2.3	Understanding GPIO Speed	188
6.3	Driving a GPIO	191
6.4	De-initialize a GPIO	192
7.	Interrupts Management	194
7.1	NVIC Controller	194
7.1.1	Vector Table in STM32	195
7.2	Enabling Interrupts	199
7.2.1	External Lines and NVIC	199
7.2.2	Enabling Interrupts With CubeMX	203
7.3	Interrupt Lifecycle	205
7.4	Interrupt Priority Levels	209
7.4.1	Cortex-M0/0+	209
7.4.2	Cortex-M3/4/7	214
7.4.3	Setting Interrupt Priority in CubeMX	220
7.5	Interrupt Re-Entrancy	221
7.6	Mask All Interrupts at Once or an a Priority Basis	222
8.	Universal Asynchronous Serial Communications	226
8.1	Introduction to UARTs and USARTs	226
8.2	UART Initialization	230
8.2.1	UART Configuration Using CubeMX	237
8.3	UART Communication in <i>Polling Mode</i>	238
8.3.1	Installing a Serial Console in Windows	242
8.3.2	Installing a Serial Console in Linux and MacOS X	245
8.4	UART Communication in <i>Interrupt Mode</i>	246
8.4.1	UART Related Interrupts	247
8.5	Error Management	254
8.6	I/O Retargeting	256
9.	DMA Management	260

CONTENTS

9.1	Introduction to DMA	260
9.1.1	The Need of a DMA and the Role of the Internal Buses	261
9.1.2	The DMA Controller	264
9.1.2.1	The DMA Implementation in F0/F1/F3/L1 MCUs	265
9.1.2.2	The DMA Implementation in F2/F4/F7 MCUs	269
9.1.2.3	The DMA Implementation in L0/L4 MCUs	272
9.2	HAL_DMA Module	273
9.2.1	DMA_HandleTypeDef in F0/F1/F3/L0/L1/L4 HALs	273
9.2.2	DMA_HandleTypeDef in F2/F4/F7 HALs	276
9.2.3	DMA_HandleTypeDef in L0/L4 HALs	279
9.2.4	How to Perform Transfers in Polling Mode	279
9.2.5	How to Perform Transfers in Interrupt Mode	282
9.2.6	How to Perform <i>Peripheral-To-Peripheral</i> Transfers	284
9.2.7	Using the HAL_UART Module With DMA Mode Transfers	285
9.2.8	Miscellaneous Functions From HAL_DMA and HAL_DMA_Ex Modules	287
9.3	Using CubeMX to Configure DMA Requests	289
9.4	Correct Memory Allocation of DMA Buffers	289
9.5	A Case Study: The DMA <i>Memory-To-Memory</i> Transfer Performance Analysis	290
10.	Clock Tree	296
10.1	Clock Distribution	296
10.1.1	Overview of the STM32 Clock Tree	298
10.1.1.1	The Multispeed Internal RC Oscillator in STM32L Families	302
10.1.2	Configuring Clock Tree Using CubeMX	303
10.1.3	Clock Source Options in Nucleo Boards	305
10.1.3.1	OSC Clock Supply	305
10.1.3.2	OSC 32kHz Clock Supply	306
10.2	Overview of the HAL_RCC Module	307
10.2.1	Compute the Clock Frequency at Run-Time	309
10.2.2	Enabling the <i>Master Clock Output</i>	310
10.2.3	Enabling the <i>Clock Security System</i>	310
10.3	HSI Calibration	311
11.	Timers	313
11.1	Introduction to Timers	313
11.1.1	Timer Categories in an STM32 MCU	314
11.1.2	Effective Availability of Timers in the STM32 Portfolio	316
11.2	Basic Timers	318
11.2.1	Using Timers in <i>Interrupt Mode</i>	321
11.2.1.1	Time Base Generation in <i>Advanced</i> Timers	324
11.2.2	Using Timers in <i>Polling Mode</i>	324
11.2.3	Using Timers in <i>DMA Mode</i>	325
11.2.4	Stopping a Timer	327

CONTENTS

11.2.5	Using CubeMX to Configure a <i>Basic Timer</i>	327
11.3	General Purpose Timers	328
11.3.1	Time Base Generator With External Clock Sources	328
11.3.1.1	External Clock Mode 2	330
11.3.1.2	External Clock Mode 1	334
11.3.1.3	Using CubeMX to Configure the Source Clock of a <i>General Purpose</i> Timer	339
11.3.2	Master/Slave Synchronization Modes	340
11.3.2.1	Enable Trigger-Related Interrupts	345
11.3.2.2	Using CubeMX to Configure the Master/Slave Synchronization	345
11.3.3	Generate Timer-Related Events by Software	346
11.3.4	Counting Modes	348
11.3.5	Input Capture Mode	349
11.3.5.1	Using CubeMX to Configure the Input Capture Mode	356
11.3.6	Output Compare Mode	357
11.3.6.1	Using CubeMX to Configure the Output Compare Mode	362
11.3.7	Pulse-Width Generation	362
11.3.7.1	Generating a Sinusoidal Wave Using PWM	366
11.3.7.2	Using CubeMX to Configure the PWM Mode	371
11.3.8	One Pulse Mode	372
11.3.8.1	Using CubeMX to Configure the OPM Mode	374
11.3.9	Encoder Mode	375
11.3.9.1	Using CubeMX to Configure the <i>Encoder Mode</i>	380
11.3.10	Other Features Available in <i>General Purpose</i> and <i>Advanced</i> Timers	381
11.3.10.1	<i>Hall Sensor</i> Mode	381
11.3.10.2	Combined Three-Phase PWM Mode and Other Motor-Control Related Features	382
11.3.10.3	Break Input and Locking of Timer Registers	382
11.3.10.4	Preloading of Auto-Reload Register	382
11.3.11	Debugging and Timers	383
11.4	SysTick Timer	384
11.4.1	Use Another Timer as System Timebase Source	385
11.5	A Case Study: How to Precisely Measure Microseconds With STM32 MCUs	386
12.	Analog-To-Digital Conversion	392
12.1	Introduction to SAR ADC	392
12.2	HAL_ADC Module	397
12.2.1	Conversion Modes	400
12.2.1.1	Single-Channel, Single Conversion Mode	400
12.2.1.2	Scan Single Conversion Mode	400
12.2.1.3	Single-Channel, Continuous Conversion Mode	401
12.2.1.4	Scan Continuous Conversion Mode	401
12.2.1.5	Injected Conversion Mode	402

CONTENTS

12.2.1.6	Dual Modes	403
12.2.2	Channel Selection	403
12.2.3	ADC Resolution and Conversion Speed	404
12.2.4	A/D Conversions in Polling Mode	405
12.2.5	A/D Conversions in Interrupt Mode	409
12.2.6	A/D Conversions in DMA Mode	410
12.2.6.1	Convert Multiple Times the Same Channel in DMA Mode	413
12.2.6.2	Multiple and not Continuous Conversions in DMA Mode	414
12.2.6.3	Continuous Conversions in DMA Mode	414
12.2.7	Errors Management	414
12.2.8	Timer-Driven Conversions	415
12.2.9	Conversions Driven by External Events	418
12.2.10	ADC Calibration	418
12.3	Using CubeMX to Configure ADC Peripheral	419
13.	Digital-To-Analog Conversion	422
13.1	Introduction to the DAC Peripheral	422
13.2	HAL_DAC Module	425
13.2.1	Driving the DAC Manually	426
13.2.2	Driving the DAC in DMA Mode Using a Timer	428
13.2.3	Triangular Wave Generation	432
13.2.4	Noise Wave Generation	433
14.	I²C	435
14.1	Introduction to the I ² C specification	435
14.1.1	The I ² C Protocol	437
14.1.1.1	START and STOP Condition	438
14.1.1.2	Byte Format	438
14.1.1.3	Address Frame	438
14.1.1.4	Acknowledge (ACK) and Not Acknowledge (NACK)	439
14.1.1.5	Data Frames	440
14.1.1.6	Combined Transactions	440
14.1.1.7	Clock Stretching	441
14.1.2	Availability of I ² C Peripherals in STM32 MCUs	442
14.2	HAL_I2C Module	443
14.2.1	Using the I ² C Peripheral in <i>Master Mode</i>	446
14.2.1.1	I/O MEM Operations	454
14.2.1.2	Combined Transactions	456
14.2.1.3	A Note About the Clock Configuration in STM32F0/L0/L4 families	458
14.2.2	Using the I ² C Peripheral in <i>Slave Mode</i>	458
14.3	Using CubeMX to Configure the I ² C Peripheral	464
15.	SPI	466

CONTENTS

15.1	Introduction to the SPI Specification	466
15.1.1	Clock Polarity and Phase	469
15.1.2	Slave Select Signal Management	470
15.1.3	SPI <i>TI Mode</i>	470
15.1.4	Availability of SPI Peripherals in STM32 MCUs	472
15.2	HAL_SPI Module	472
15.2.1	Exchanging Messages Using SPI Peripheral	474
15.2.2	Maximum Transmission Frequency Reachable using the CubeHAL	476
15.3	Using CubeMX to Configure SPI Peripheral	477
16.	Cyclic Redundancy Check	478
16.1	Introduction to CRC Computing	478
16.1.1	CRC Calculation in STM32F1/F2/F4/L1 MCUs	481
16.1.2	CRC Peripheral in STM32F0/F3/F7/L0/L4 MCUs	483
16.2	HAL_CRC Module	484
17.	IWDG and WWDG Timers	488
17.1	The <i>Independent Watchdog</i> Timer	488
17.1.1	Using the CubeHAL to Program IWDG Timer	489
17.2	The <i>System Window Watchdog</i> Timer	490
17.2.1	Using the CubeHAL to Program WWDG Timer	492
17.3	Detecting a System Reset Caused by a Watchdog Timer	494
17.4	Freezing Watchdog Timers During a Debug Session	494
17.5	Selecting the Right Watchdog Timer for Your Application	494
III	Advanced topics	495
18.	Power Management	496
18.1	Power Management in Cortex-M Based MCUs	496
18.2	How Cortex-M MCUs Handle <i>Run</i> and <i>Sleep</i> Modes	497
18.2.1	Entering/exiting sleep modes	500
18.2.1.1	Sleep-On-Exit	502
18.2.2	<i>Sleep</i> Modes in Cortex-M Based MCUs	503
18.3	Power Management in STM32F Microcontrollers	503
18.3.1	Power Sources	504
18.3.2	Power Modes	505
18.3.2.1	Run Mode	505
18.3.2.1.1	Dynamic Voltage Scaling in STM32F4/F7 MCUs	506
18.3.2.1.2	Over/Under-Drive Mode in STM32F4/F7 MCUs	507
18.3.2.2	Sleep Mode	507
18.3.2.3	Stop Mode	508
18.3.2.4	Standby Mode	509

CONTENTS

18.3.2.5	Low-Power Modes Example	510
18.3.3	An Important Warning for STM32F1 Microcontrollers	514
18.4	Power Management in STM32L Microcontrollers	515
18.4.1	Power Sources	515
18.4.2	Power Modes	517
18.4.2.1	Run Modes	517
18.4.2.2	Sleep Modes	519
18.4.2.2.1	Batch Acquisition Mode	520
18.4.2.3	Stop Modes	520
18.4.2.4	Standby Modes	521
18.4.2.5	Shutdown Mode	522
18.4.3	Power Modes Transitions	523
18.4.4	Low-Power Peripherals	523
18.4.4.1	LPUART	523
18.4.4.2	LPTIM	524
18.5	Power Supply Supervisors	524
18.6	Debugging in Low-Power Modes	525
18.7	Using the CubeMX Power Consumption Calculator	525
18.8	A Case Study: Using Watchdog Timers With Low-Power Modes	527
19.	Memory layout	528
19.1	The STM32 Memory Layout Model	528
19.1.1	Understanding Compilation and Linking Processes	530
19.2	The Really Minimal STM32 Application	533
19.2.1	ELF Binary File Inspection	537
19.2.2	.data and .bss Sections Initialization	539
19.2.2.1	A Word About the COMMON Section	546
19.2.3	.rodata Section	547
19.2.4	Stack and Heap Regions	549
19.2.5	Checking the Size of Heap and Stack at Compile-Time	552
19.2.6	Differences With the Tool-Chain Script Files	553
19.3	How to Use the CCM Memory	555
19.3.1	Relocating the <i>vector table</i> in CCM Memory	558
19.4	How to Use the MPU in Cortex-M0+/3/4/7 Based STM32 MCUs	561
19.4.1	Programming the MPU With the CubeHAL	565
20.	Flash Memory Management	569
20.1	Introduction to STM32 Flash Memory	569
20.2	The HAL_FLASH Module	573
20.2.1	Flash Memory Unlocking	573
20.2.2	Flash Memory Erasing	573
20.2.3	Flash Memory Programming	575
20.2.4	Flash Read Access During Programming and Erasing	576

CONTENTS

20.3	Option Bytes	576
20.3.1	Flash Memory Read Protection	578
20.4	Optional OTP and True-EEPROM Memories	580
20.5	Flash Read Latency and the ART™ Accelerator	581
20.5.1	The Role of the TCM Memories in STM32F7 MCUs	584
20.5.1.1	How to Access Flash Memory Through the TCM Interface	590
20.5.1.2	Using CubeMX to Configure Flash Memory Interface	591
21.	Booting Process	593
21.1	The Cortex-M Unified Memory Layout and the Booting Process	593
21.1.1	Software <i>Physical Remap</i>	594
21.1.2	Vector Table Relocation	595
21.1.3	Running the Firmware From SRAM Using the GNU ARM Eclipse Toolchain	597
21.2	Integrated Bootloader	598
21.2.1	Starting the Bootloader From the On-Board Firmware	600
21.2.2	The Booting Sequence in the GNU ARM Eclipse Tool-chain	601
21.3	Developing a Custom Bootloader	604
21.3.1	<i>Vector Table</i> Relocation in STM32F0/L0 Microcontrollers	615
21.3.2	How to Use the <code>flasher.py</code> Tool	618
22.	Running FreeRTOS	621
22.1	Understanding the Concepts Underlying an RTOS	622
22.2	Introduction to FreeRTOS and CMSIS-RTOS Wrapper	628
22.2.1	The FreeRTOS Source Tree	629
22.2.1.1	How to Import FreeRTOS Manually	630
22.2.1.2	How to Import FreeRTOS Using CubeMX and CubeMXImporter	631
22.2.1.3	How to Enable FPU Support in Cortex-M4F and Cortex-M7 Cores	633
22.3	Thread Management	633
22.3.1	Thread States	636
22.3.2	Thread Priorities and Scheduling Policies	637
22.3.3	Voluntary Release of the Control	641
22.3.4	The <i>idle</i> Thread	641
22.4	Memory Allocation and Management	642
22.4.1	Dynamic Memory Allocation Model	643
22.4.1.1	<code>heap_1.c</code>	644
22.4.1.2	<code>heap_2.c</code>	645
22.4.1.3	<code>heap_3.c</code>	645
22.4.1.4	<code>heap_4.c</code>	645
22.4.1.5	<code>heap_5.c</code>	646
22.4.1.6	How to Use <code>malloc()</code> and Related C Functions With FreeRTOS	646
22.4.1.7	FreeRTOS Heap Definition	647
22.4.2	Static Memory Allocation Model	647
22.4.2.1	<i>idle</i> Thread Allocation With Static Memory Allocation Model	648

CONTENTS

22.4.3	Memory Pools	648
22.4.4	Stack Overflow Detection	650
22.5	Synchronization Primitives	652
22.5.1	Message Queues	652
22.5.2	Semaphores	656
22.5.3	Thread Signals	659
22.6	Resources Management and Mutual Exclusion	660
22.6.1	Mutexes	660
22.6.1.1	The Priority Inversion Problem	661
22.6.1.2	Recursive Mutexes	662
22.6.2	Critical Sections	663
22.6.3	Interrupt Management With an RTOS	664
22.6.3.1	FreeRTOS API and Interrupt Priorities	665
22.7	Software Timers	666
22.7.1	How FreeRTOS Manages Timers	667
22.8	A Case Study: Low-Power Management With an RTOS	668
22.8.1	The <i>idle</i> Thread Hook	668
22.8.2	The Tickless Mode in FreeRTOS	669
22.8.2.1	A Schema for the <i>tickless</i> Mode	671
22.8.2.2	A Custom <i>tickless</i> Mode Policy	675
22.9	Debugging Features	682
22.9.1	configASSERT() Macro	682
22.9.2	Run-Time Statistics and Thread State Information	683
22.10	Alternatives to FreeRTOS	687
22.10.1	ChibiOS	687
22.10.2	Contiki OS	687
22.10.3	OpenRTOS	688
23.	Advanced Debugging Techniques	689
23.1	Understanding Cortex-M Fault-Related Exceptions	689
23.1.1	The Cortex-M Exception Entrance Sequence and the ARM Calling Convention	691
23.1.1.1	How the GNU ARM Eclipse Tool-chain Handles Fault-Related Exceptions	696
23.1.1.2	How to Interpret the Content of the LR Register on Exception Entrance	698
23.1.2	Fault Exceptions and Faults Analysis	699
23.1.2.1	Memory Management Exception	699
23.1.2.2	Bus Fault Exception	700
23.1.2.3	Usage Fault Exception	701
23.1.2.4	Hard Fault Exception	702
23.1.2.5	Enabling Optional Fault Handlers	703
23.1.2.6	Fault Analysis in Cortex-M0/0+ Based Processors	703
23.2	Eclipse Advanced Debugging Features	704
23.2.1	Expressions	704

CONTENTS

23.2.1.1	Memory Monitors	705
23.2.2	Watchpoints	706
23.2.3	Instruction Stepping Mode	707
23.2.4	Keil Packs and Peripheral Registers View	708
23.2.5	Core Registers View	711
23.3	Debugging Aids From the CubeHAL	712
23.4	External Debuggers	712
23.4.1	Using SEGGER J-Link for ST-LINK Debugger	714
23.4.2	Using the ITM Interface and SWV Tracing	718
23.5	STM Studio	719
23.6	Debugging two Nucleo Boards Simultaneously	721
24.	FAT Filesystem	724
24.1	Introduction to FatFs Library	724
24.1.1	Using CubeMX to Include FatFs Library in Your Projects	727
24.1.1.1	The <i>Generic Disk Interface</i> API	728
24.1.1.2	The Implementation of a Driver to Access SD Cards in SPI Mode	729
24.1.2	Relevant FatFs Structures and Functions	730
24.1.2.1	Mounting a Filesystem	730
24.1.2.2	Opening a File	730
24.1.2.3	Reading From/Writing Into a File	731
24.1.2.4	Creating and Opening a Directory	732
24.1.3	How to Configure the FatFs Library	735
25.	Develop IoT Applications	738
25.1	Solutions Offered by STM to Develop IoT Applications	739
25.2	The W5500 Ethernet Controller	741
25.2.1	How to Use the W5500 Shield and the <code>ioLibrary_Driver</code> Module	745
25.2.1.1	Configuring the SPI Interface	747
25.2.1.2	Configuring the Socket Buffers and the Network Interface	748
25.2.2	Socket APIs	750
25.2.2.1	Handling Sockets in TCP Mode	752
25.2.2.2	Handling Sockets in UDP Mode	752
25.2.3	I/O Retargeting to a TCP/IP Socket	753
25.2.4	Setting up an HTTP Server	755
25.2.4.1	A Web-Based Oscilloscope	758
26.	Getting Started With a New Design	771
26.1	Hardware Design	771
26.1.1	PCB Layer Stack-Up	772
26.1.2	MCU Package	773
26.1.3	Decoupling of Power-Supply Pins	774
26.1.4	Clocks	776

CONTENTS

26.1.5	Filtering of RESET Pin	777
26.1.6	Debug Port	777
26.1.7	Boot Mode	779
26.1.8	Pay attention to “pin-to-pin” Compatibility...	780
26.1.9	...And to Selecting the Right Peripherals	781
26.1.10	The Role of CubeMX During the Board Design Stage	781
26.1.11	Board Layout Strategies	784
26.2	Software Design	785
26.2.1	Generating the binary image for production	785

Appendix 788

A. Miscellaneous HAL functions and STM32 features	789
Force MCU reset from the firmware	789
STM32 96-bit Unique CPU ID	789
B. Troubleshooting guide	791
GNU ARM Eclipse Installation Issues	791
Eclipse related issue	791
Eclipse cannot locate the compiler	792
Eclipse continuously breaks at every instruction during debug session	793
The step-by-step debugging is really slow	793
The firmware works only under a debug session	794
STM32 related issue	794
The microcontroller does not boot correctly	794
It is Not Possible to Flash or to Debug the MCU	796
C. Nucleo pin-out	798
Nucleo-F446RE	799
Arduino compatible headers	799
Morpho headers	799
Nucleo-F411RE	800
Arduino compatible headers	800
Morpho headers	800
Nucleo-F410RB	801
Arduino compatible headers	801
Morpho headers	801
Nucleo-F401RE	802
Arduino compatible headers	802
Morpho headers	802
Nucleo-F334R8	803
Arduino compatible headers	803

CONTENTS

Morpho headers	803
Nucleo-F303RE	804
Arduino compatible headers	804
Morpho headers	804
Nucleo-F302R8	805
Arduino compatible headers	805
Morpho headers	805
Nucleo-F103RB	806
Arduino compatible headers	806
Morpho headers	806
Nucleo-F091RC	807
Arduino compatible headers	807
Morpho headers	807
Nucleo-F072RB	808
Arduino compatible headers	808
Morpho headers	808
Nucleo-F070RB	809
Arduino compatible headers	809
Morpho headers	809
Nucleo-F030R8	810
Arduino compatible headers	810
Morpho headers	810
Nucleo-L476RG	811
Arduino compatible headers	811
Morpho headers	811
Nucleo-L152RE	812
Arduino compatible headers	812
Morpho headers	812
Nucleo-L073R8	813
Arduino compatible headers	813
Morpho headers	813
Nucleo-L053R8	814
Arduino compatible headers	814
Morpho headers	814
D. STM32 packages	815
LFBGA	815
LQFP	815
TFBGA	816
TSSOP	816
UFBGA	816
UFQFPN	817
VFQFP	817

CONTENTS

WLCSP	817
E. History of this book	819
Release 0.1 - October 2015	819
Release 0.2 - October 28th, 2015	819
Release 0.2.1 - October 31th, 2015	819
Release 0.2.2 - November 1st, 2015	820
Release 0.3 - November 12th, 2015	820
Release 0.4 - December 4th, 2015	820
Release 0.5 - December 19th, 2015	820
Release 0.6 - January 18th, 2016	821
Release 0.6.1 - January 20th, 2016	821
Release 0.6.2 - January 30th, 2016	821
Release 0.7 - February 8th, 2016	821
Release 0.8 - February 18th, 2016	822
Release 0.8.1 - February 23th, 2016	822
Release 0.9 - March 27th, 2016	822
Release 0.9.1 - March 28th, 2016	823
Release 0.10 - April 26th, 2016	823
Release 0.11 - May 27th, 2016	823
Release 0.11.1 - June 3rd, 2016	824
Release 0.11.2 - June 24th, 2016	824
Release 0.12 - July 4th, 2016	824
Release 0.13 - July 18th, 2016	825
Release 0.14 - August 12th, 2016	825
Release 0.15 - September 13th, 2016	825
Release 0.16 - October 3th, 2016	826
Release 0.17 - October 24th, 2016	826
Release 0.18 - November 15th, 2016	826
Release 0.19 - November 29th, 2016	827
Release 0.20 - December 28th, 2016	827
Release 0.21 - January 29th, 2017	827
Release 0.22 - May 1st, 2017	827