

Embedded Systems Architecture

Explore architectural concepts, pragmatic design patterns,
and best practices to produce robust systems



By Daniele Lacamera

Packt

www.packt.com

Embedded Systems Architecture

Copyright © 2018 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author(s), nor Packt Publishing or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

Commissioning Editor: Richa Tripathi

Acquisition Editor: Shriram Shekhar

Content Development Editor: Zeeyan Pinheiro

Technical Editor: Romy Dias

Copy Editor: Safis Editing

Project Coordinator: Vaidehi Sawant

Proofreader: Safis Editing

Indexer: Mariammal Chettiyar

Graphics: Jason Monteiro

Production Coordinator: Aparna Bhagat

First published: May 2018

Production reference: 1280518

Published by Packt Publishing Ltd.

Livery Place

35 Livery Street

Birmingham

B3 2PB, UK.

ISBN 978-1-78883-250-2

www.packtpub.com

Table of Contents

Preface	1
Chapter 1: Embedded Systems – A Pragmatic Approach	7
Domain definition	7
Embedded Linux systems	8
Low-end 8-bit microcontrollers	9
Hardware architecture	10
Understanding the challenge	13
Multithreading	14
RAM	14
Flash memory	16
Interfaces and peripherals	17
Asynchronous UART-based serial communication	17
SPI	18
I2C	19
USB	19
Connected systems	20
The reference platform	21
ARM reference design	22
The Cortex-M microprocessor	22
Summary	24
Chapter 2: Work Environment and Workflow Optimization	25
Workflow overview	25
C compiler	26
Linker	27
Build automation	29
Debugger	32
Embedded workflow	33
The GCC toolchain	35
The cross-compiler	36
Compiling the compiler	38
Linking the executable	39
Binary format conversion	44
Interacting with the target	45
The GDB session	47
Validation	50
Functional tests	50
Hardware tools	51

Testing off-target	52
Emulators	55
Summary	56
Chapter 3: Architectural Patterns	57
Configuration management	57
Revision control	58
Tracking activities	59
Code reviews	60
Continuous integration	60
Source code organization	61
Hardware abstraction	62
Middleware	63
Application code	64
The life cycle of an embedded project	65
Defining project steps	66
Prototyping	67
Refactoring	68
API and documentation	68
Summary	70
Chapter 4: The Boot-Up Procedure	71
The interrupt vector table	71
Startup code	72
Reset handler	74
Allocating the stack	75
Fault handlers	75
Memory layout	76
Building and running the boot code	80
The makefile	80
Running the application	83
Multiple boot stages	83
Bootloader	84
Building the image	87
Debugging a multi-stage system	88
Shared libraries	89
Summary	91
Chapter 5: Memory Management	93
Memory mapping	93
Memory model and address space	94
The code region	95
The RAM regions	95
Peripheral-access regions	96
The system region	96
Order of memory transactions	97

The execution stack	98
Stack placement	99
Stack overflows	100
Stack painting	102
Heap management	103
Custom implementation	104
Using newlib	107
Limiting the heap	108
Multiple memory pools	109
Common heap usage errors	111
The memory protection unit	113
MPU configuration registers	113
Programming the MPU	114
Summary	118
Chapter 6: General-Purpose Peripherals	119
The interrupt controller	120
Peripherals interrupt configuration	121
System time	122
Adjusting the flash wait states	122
Clock configuration	123
Clock distribution	127
Enabling the SysTick	128
Generic timers	131
General-purpose I/O	134
Pin configuration	134
Digital output	135
PWM	137
Digital input	142
Interrupt-based input	143
Analog input	146
The watchdog	149
Summary	152
Chapter 7: Local Bus Interfaces	153
Introducing serial communication	153
Clock and symbol synchronization	154
Bus wiring	155
Programming the peripherals	156
UART-based asynchronous serial bus	157
Protocol description	157
Programming the controller	158
Hello world!	161
Newlib printf	163
Receiving data	164

Interrupt-based input/output	165
SPI bus	166
Protocol description	167
Programming the transceiver	168
SPI transactions	171
Interrupt-based SPI transfers	174
I2C bus	175
Protocol description	177
Clock stretching	179
Multi-master	180
Programming the controller	181
Interrupt handling	185
Summary	185
Chapter 8: Low-Power Optimizations	187
System configuration	188
Hardware design	188
Clock management	189
Voltage control	191
Low-power operating modes	192
Deep-sleep configuration	194
Stop mode	196
Standby mode	199
Wake-up intervals	203
Measuring power	204
Development boards	204
Designing low-power embedded applications	205
Replacing busy loops with sleep mode	205
Deep sleep during longer inactivity periods	207
Choosing the clock speed	207
Power state transitions	208
Summary	210
Chapter 9: Distributed Systems and IoT Architecture	211
Network interfaces	212
Media Access Control	212
Ethernet	213
Wi-Fi	214
Low-Rate Wireless Personal Area Networks (LR-WPAN)	214
LR-WPAN industrial link-layer extensions	215
6LoWPAN	216
Bluetooth	216
Mobile networks	217
Low-power Wide Area Networks (LPWANs)	218
Selecting the appropriate network interfaces	219
	220

The Internet Protocols	221
TCP/IP implementations	222
Network device drivers	223
Running the TCP/IP stack	225
Socket communication	228
Mesh networks and dynamic routing	231
Transport Layer Security	234
Securing socket communication	235
Application protocols	238
Message protocols	239
REST architectural pattern	240
Distributed systems – single points of failure	240
Summary	241
Chapter 10: Parallel Tasks and Scheduling	243
Task management	244
The task block	245
Context switch	248
Creating tasks	250
Scheduler implementation	253
Supervisor calls	253
Cooperative scheduler	255
Concurrency and timeslices	256
Blocking tasks	257
Waiting for resources	262
Real-time scheduling	265
Synchronization	269
Semaphore	270
Mutex	273
Priority inversion	273
System resource separation	274
Privilege levels	275
Memory segmentation	277
System calls	279
Summary	282
Chapter 11: Embedded Operating Systems	283
Real-time application platforms	286
FreeRTOS	286
ChibiOS	288
Low-power IoT systems	289
Contiki OS	290
Riot OS	291
POSIX-compliant systems	292
NuttX	295

Frosted	296
The future of safe embedded systems	297
Process isolation – Tock	298
Summary	299
Other Books You May Enjoy	301
Index	305
