<packt>

2ND EDITION

Embedded Systems Architecture

Design and write software for embedded devices to build safe and connected systems

DANIELE LACAMERA

Table of Contents

Preface xi

Part 1 – Introduction to Embedded Systems Development

1

Embedded Systems – A Pragmatic Approach 3

Domain definition 3
Embedded Linux systems 4
Low-end 8-bit microcontrollers 5
Hardware architecture 5
Understanding the challenges 8
Multithreading 9
RAM 9
Flash memory 10
General-purpose input/output (GPIO) 11
ADC and DAC 11
Timers and PWM 12
Interfaces and peripherals 12
Asynchronous UART-based serial communication 12
SPI 13
I²C 13
USB 13
Connected systems 14
Challenges of distributed systems 15
Introduction to isolation mechanisms 15
The reference platform 16
ARM reference design 16
The Cortex-M microprocessor 17
Summary 18

2

Work Environment and Workflow Optimization 19

Workflow overview 20
The C compiler 20
Linker 21
Make: a build automation tool 22

Debugger 25
Embedded workflow 26

Text editors versus integrated environments 28

The GCC toolchain 29

The cross compiler 29
Compiling the compiler 31
Linking the executable 32
Binary format conversion 37

Interacting with the target 37

The GDB session 40

Validation 43

Functional tests 43
Hardware tools 44
Testing off-target 45
Emulators 48

Summary 50

Part 2 – Core System Architecture

3

Architectural Patterns 53

Configuration management 53

Revision control 54
Tracking activities 55
Code reviews 56
Continuous integration 56

Source code organization 57

Hardware abstraction 57
Middleware 58
Application code 59

Security considerations 60

Vulnerability management 60
Software cryptography 61
Hardware cryptography 62
Running untrusted code 62

The life cycle of an embedded project 62

Defining project steps 63
Prototyping 64
Refactoring 65
API and documentation 65

Summary 67

4

The Boot-Up Procedure 69

Technical requirements 69

The interrupt vector table 69

Startup code 71
Reset handler 72
Allocating the stack 73
Fault handlers 73

Memory layout 74

Building and running the boot code 77

The makefile 77

Debugger 25
Embedded workflow 26

Text editors versus integrated environments 28

The GCC toolchain 29

The cross compiler 29
Compiling the compiler 31
Linking the executable 32
Binary format conversion 37

Interacting with the target 37

The GDB session 40

Validation 43

Functional tests 43
Hardware tools 44
Testing off-target 45
Emulators 48

Summary 50

Part 2 – Core System Architecture

3

Architectural Patterns 53

Configuration management 53

Revision control 54
Tracking activities 55
Code reviews 56
Continuous integration 56

Source code organization 57

Hardware abstraction 57
Middleware 58
Application code 59

Security considerations 60

Vulnerability management 60
Software cryptography 61
Hardware cryptography 62
Running untrusted code 62

The life cycle of an embedded project 62

Defining project steps 63
Prototyping 64
Refactoring 65
API and documentation 65

Summary 67

4

The Boot-Up Procedure 69

Technical requirements 69

The interrupt vector table 69

Startup code 71
Reset handler 72
Allocating the stack 73
Fault handlers 73

Memory layout 74

Building and running the boot code 77

The makefile 77

Running the application 80

Multiple boot stages 80

Bootloader 81

Building the image 83

Debugging a multi-stage system 84

Shared libraries 85

Remote firmware updates 87

Secure boot 87

Summary 87

5

Memory Management 89

Technical requirements 89

Memory mapping 90

Memory model and address space 90

The code region 91

The RAM regions 92

Peripheral-access regions 93

The system region 93

Order of memory transactions 93

The execution stack 94

Stack placement 95

Stack overflows 96

Stack painting 98

Heap management 99

Custom implementation 100

Using newlib 103

Limiting the heap 104

Multiple memory pools 105

Common heap usage errors 107

The memory protection unit 109

MPU configuration registers 109

Programming the MPU 110

Summary 114

Part 3 – Device Drivers and Communication Interfaces

6

General-Purpose Peripherals 117

Technical requirements 118

Bitwise operations 118

The interrupt controller 118

Peripherals' interrupt configuration 119

System time 120

Adjusting the flash wait states 121

Clock configuration 121

Clock distribution 126

Enabling the SysTick 127

Generic timers 129

GPIO 133

Pin configuration 133

Digital output 134

PWM 137
Digital input 142
Interrupt-based input 143
Analog input 145
The watchdog 149
Summary 152

7

Local Bus Interfaces 153

Technical requirements 153
Introducing serial communication 154
Clock and symbol synchronization 155
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 167
Protocol description 167
Programming the transceiver 168
SPI transactions 172
Interrupt-based SPI transfers 175
I^2C bus 175
Protocol description 177
Clock stretching 179
Multiple masters 180
Programming the controller 180
Interrupt handling 184
Summary 185

8

Power Management and Energy Saving 187

Technical requirements 188
System configuration 188
Hardware design 188
Clock management 189
Voltage control 192
Low-power operating modes 192
Deep-sleep configuration 193
Stop mode 195
Standby mode 198
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 206
Choosing the clock speed 207
Power state transitions 207
Summary 209

9

Distributed Systems and IoT Architecture 211

Technical requirements 212
Network interfaces 212
MAC 213
Selecting the appropriate network interfaces 218
The IPs 220
Standard protocols, custom implementations 220
The TCP/IP stack 220
Network device drivers 221
Running the TCP/IP stack 223
Socket communication 226
Connectionless protocols 229
Mesh networks and dynamic routing 229
TLS 232
Securing socket communication 234
Application protocols 237
Message protocols 238
The REST architectural pattern 238
Distributed systems – single points of failure 239
Summary 240

Part 4 – Multithreading

10

Parallel Tasks and Scheduling 243

Technical requirements 244
Task management 244
The task block 244
Context switch 247
Creating tasks 249
Scheduler implementation 252
Supervisor calls 252
Cooperative scheduler 254
Concurrency and timeslices 255
Blocking tasks 256
Waiting for resources 262
Real-time scheduling 264
Synchronization 269
Semaphores 269
Mutexes 273
Priority inversion 273
System resource separation 274
Privilege levels 274
Memory segmentation 276
System calls 279
Embedded operating systems 282
OS selection 283
FreeRTOS 284
RIOT OS 286
Summary 289

11

Trusted Execution Environment 291

Technical requirements 292
Sandboxing 292
TrustZone-M 293
Reference platform 294
Secure and non-secure execution domains 295
System resources separation 296
Security attributes and memory regions 296
Flash memory and secure watermarks 300
GTZC configuration and block-based SRAM protection 301
Configuring secure access to peripherals 302
Building and running the example 303
Enabling TrustZone-M 304
Secure application entry point 304
Compiling and linking secure-world applications 305
Compiling and linking non-secure applications 306
Inter-domain transitions 307
Summary 310

Index 311

Other Books You May Enjoy 322