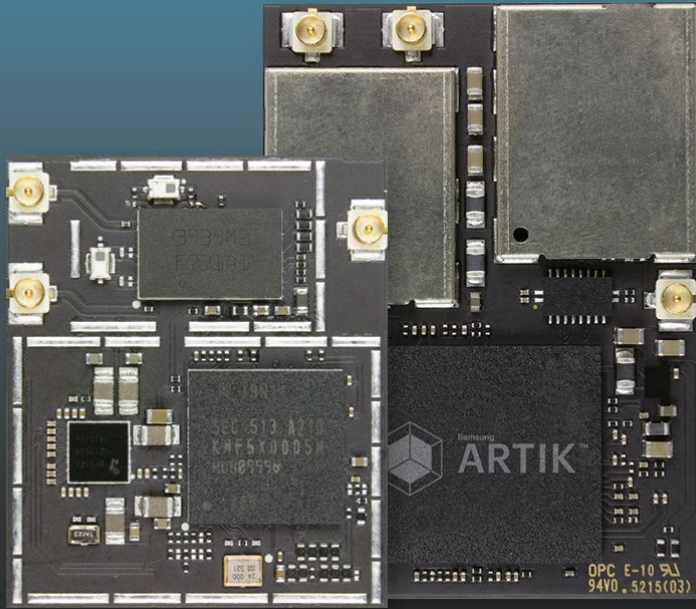




Samsung ARTIK Reference

The Definitive Developers Guide



Cliff Wootton

Samsung ARTIK Reference

The Definitive Developers Guide



Cliff Wootton

Apress®

Samsung ARTIK Reference

Cliff Wootton

Crowborough, East Sussex, United Kingdom

ISBN-13 (pbk): 978-1-4842-2321-5

ISBN-13 (electronic): 978-1-4842-2322-2

DOI 10.1007/978-1-4842-2322-2

Library of Congress Control Number: 2016956942

Copyright © 2016 by Cliff Wootton

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

ARM[®] is a registered trademark owned by ARM Ltd

Mali[™] is a trademark owned by ARM Ltd

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

All other trademarks are the property of their respective owners.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director: Welmoed Spahr

Lead Editor: Jonathan Gennick

Technical Reviewer: Fred Patton

Editorial Board: Steve Anglin, Pramila Balan, Laura Berendson, Aaron Black, Louise Corrigan,

Jonathan Gennick, Todd Green, Robert Hutchinson, Celestin Suresh John, Nikhil Karkal,

James Markham, Susan McDermott, Matthew Moodie, Natalie Pao, Gwenan Searring

Coordinating Editor: Jill Balzano

Copy Editor: Mary Behr

Compositor: SPi Global

Indexer: SPi Global

Artist: SPi Global

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springer.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary materials referenced by the author in this text are available to readers at www.apress.com. For detailed information about how to locate your book's source code, go to www.apress.com/source-code/. Readers can also access source code at SpringerLink in the Supplementary Material section for each chapter.

Printed on acid-free paper

To Annie Parker
Mum, you have always inspired me to do my best.

Contents at a Glance

About the Authorxxiii

Acknowledgments.....xxv

Introductionxxvii

■ **Chapter 1: Learning More About ARTIK 1**

■ **Chapter 2: Hardware 9**

■ **Chapter 3: Developer Reference Boards 25**

■ **Chapter 4: About the Operating Systems 41**

■ **Chapter 5: Operating System Internals 45**

■ **Chapter 6: Startup, Sleep, and Shutdown..... 57**

■ **Chapter 7: File Systems..... 93**

■ **Chapter 8: The/sys Virtual File System 103**

■ **Chapter 9: The/dev Virtual File System 115**

■ **Chapter 10: The/proc Virtual File System..... 127**

■ **Chapter 11: The/run Virtual File System 145**

■ **Chapter 12: System Administration 153**

■ **Chapter 13: AXT Module Connectors 165**

■ **Chapter 14: Hardware I/O Connections 197**

■ **Chapter 15: Antennas 221**

■ **Chapter 16: The API Kits..... 229**

■ **Chapter 17: General Purpose Input/Output (GPIO)..... 235**

■ **Chapter 18: Analog Input and IIO 289**

■ **Chapter 19: Pulse Width Modulated Output..... 303**

■ **Chapter 20: Inter-Integrated Circuit (I2C)..... 321**

■ **Chapter 21: Serial Peripheral Interface (SPI) 335**

■ **Chapter 22: Audio and Inter-IC Sound (I2S) 351**

■ **Chapter 23: Graphics and Video 369**

■ **Chapter 24: Conclusions and Next Steps..... 391**

■ **Bibliography 399**

■ **Index 401**

Contents

About the Authorxxiii

Acknowledgmentsxxv

Introductionxxvii

■ Chapter 1: Learning More About ARTIK 1

 Change and Evolution..... 1

 Your Journey Through This Book..... 3

 Samsung Developer Resources..... 4

 Samsung Developer Downloads..... 4

 ARTIK OS Source Code 4

 Samsung Developer Blog 6

 Samsung Developer Forums 6

 Get Your Samsung Developer Account Now 6

 An Even Quicker Start 7

■ Chapter 2: Hardware 9

 The ARTIK Family..... 9

 Provenance..... 10

 Choose the Right Model 12

 Introducing the ARTIK 520..... 12

 General Arrangement..... 14

 Functional Organization..... 15

 Known Firmware Versions..... 16

Introducing the ARTIK 1020.....	16
General Arrangement.....	18
Functional Organization.....	19
Known Firmware Versions.....	20
ARTIK Module Connections	21
Physical Dimensions	22
Buy an ARTIK Development System Now	23
Component Suppliers.....	24
Summary.....	24
■ Chapter 3: Developer Reference Boards.....	25
A Little History.....	25
Connecting External Devices.....	26
Schematic Diagrams	27
Reading the Schematic Diagrams	28
Type 5 - Version 0.5.0.....	29
Type 10 - Version 0.5.0.....	33
Test Points.....	37
Interesting Chip Data Sheets.....	39
Summary.....	40
■ Chapter 4: About the Operating Systems	41
Comparing the ARTIK Operating Systems	41
About Fedora OS	41
But What Is Yocto?.....	42
Other Operating System Choices	43
Tizen Secrets.....	43
Summary.....	44

■ Chapter 5: Operating System Internals	45
Under the Hood	45
The Component Parts of the ARTIK OS	46
U-Boot Loader	46
Device Tree	47
Kernel	47
Fedora Distribution	47
Yocto Build Manager	48
Additional Software	48
Looking Inside the Kernel	48
Kernel Versions	49
Interacting with the Kernel	49
Using the Kernel debugfs Filesystem	50
Standard Input/Output Devices	50
Peripheral Interfaces	53
Programmable I/O Pins	54
Accessing the Peripherals	55
Summary	55
■ Chapter 6: Startup, Sleep, and Shutdown	57
The System Administrator Console	57
Power and Reset Buttons	57
Setting the Boot Mode Switches	59
Cold vs. Warm Boots	61
Starting Up the ARTIK	62
The Boot Loader	62
U-Boot Commands	62
Boot Loader Console Messages	65
Passing Arguments to the Kernel	66

Kernel Boot Options.....	68
Device Tree.....	69
Starting Up the Kernel.....	69
Kernel Startup Messages	70
Fedora Startup Messages	71
Reconfiguring the Startup	74
The /boot Directory.....	75
Login Credentials	75
Shutdown Commands	76
Shutdown Console Messages	77
About Power Management.....	78
Power Management Integrated Circuit (PMIC)	79
Monitoring Power Consumption	79
Arduino Power-Saving Mode	80
Power Management with systemctl	80
/sys/power.....	81
Wake Locks	82
Relevant Power Management AXT Connections.....	83
/sys/class/power_supply.....	84
Power Management Devices	85
BQ2429 - Battery Charger Chip	86
CW2015 - Battery Fuel Gauge Chip	86
The sec_pmic (S2MPS14-PMIC) Chip.....	87
Saving Power by Slowing Things Down.....	91
Summary	91
■ Chapter 7: File Systems.....	93
About the File System	93
Filesystem Hierarchy Standard (FHS).....	93
File System Inodes	93

File System Types.....	96
File Types.....	96
File Access Control	98
File System Trees	99
File System Mapped Properties	101
Summary	102
■ Chapter 8: The /sys Virtual File System	103
About sysfs.....	103
Inside sysfs	104
Kernel Developers Roadmap	104
/sys/devices	105
Memory-Mapped Base Addresses	106
Finding Base Addresses in bash	107
Finding Base Addresses in the C Language	108
Peripheral Interconnect Buses	109
/sys/bus.....	110
/sys/class	112
Summary	113
■ Chapter 9: The /dev Virtual File System	115
About /dev	115
Communicating With Devices.....	115
Listing the Devices	117
About Device Numbers.....	119
Device Types.....	122
Block Devices	122
Character Devices	123
Special Devices	126
Summary	126

■ **Chapter 10: The /proc Virtual File System..... 127**

 About /proc..... 127

 Inspecting /proc 128

 Special Locations Within /proc 128

 Kernel Subsystems as Objects..... 129

 System-Wide Properties..... 129

 Parent and Child Processes..... 131

 Creating New Processes 132

 Process Identifier Numbers..... 132

 Listing the Running Processes 132

 Sending Signals to Processes 134

 Zombies..... 135

 Special Locations Within /proc/{pid} 135

 Process Property Collections..... 137

 Process Properties 138

 Process-Related Objects 140

 Inspecting the Process Status..... 140

 Resource Usage Monitoring 142

 Altering the Process Priority..... 142

 Processes vs. Threads..... 143

 Summary..... 143

■ **Chapter 11: The /run Virtual File System 145**

 About /run 145

 Why /run Was Created..... 145

 Kernel Subsystems as Objects..... 146

 Runtime System Properties..... 147

 Socket Connections..... 148

 Process Identifier Files..... 148

 Updating Legacy Applications 149

 Summary..... 151

■ Chapter 12: System Administration	153
How to Be a Sys Admin	153
Identifying an ARTIK Module	154
Detecting the OS Version	154
Determining the Kernel Version	155
Dynamically Changing the Host Name	155
Setting the Correct Date	156
Uploading Files	157
Downloading Files	157
Examining the System Configuration	157
Checking the Memory Usage	158
Viewing Process Memory Maps	158
Discovering the Process Limits	159
Monitoring Service Status	159
Quitting and Aborting Processes	159
Determining the Available CPUs	160
Detecting Current Processor Speed	161
Managing Processor Affinity	161
Monitoring Internal Temperature	162
Summary	163
■ Chapter 13: AXT Module Connectors	165
Physical Connections	165
Panasonic AXT Connectors	165
Looking More Closely	166
AXT 40-Pin Connector	166
AXT 60-Pin Connector	167
AXT 80-Pin Connector	167
Ordering AXT Connectors	168
AXT Connections, Pins, and I/O	169

ARTIK 5 - Connectors	170
Connector Locations	171
Connector J3	172
Connector J4	175
Connector J7	178
ARTIK 10 - Connectors	181
Connector Locations	182
Connector J1	183
Connector J2	186
Connector J3	190
Connector J4	192
Summary	195
■ Chapter 14: Hardware I/O Connections	197
Pins and Programmable I/O.....	197
Hardware I/O Pins and Headers	198
Arduino Names vs. Pin Numbers.....	199
Mapping Pins to Connection Headers	200
Pinout Diagrams.....	201
J24 - Analog Input	201
J25 - Power Supply, Ground, and Reset	202
J26 and J27 Arduino Interface	204
J510 - Auxiliary Analog ADC input.....	206
J511 - SPI, UART, I2C, and I2S Interfaces	208
J512 - GPM and Interrupts	210
J513 - PWM and Clock Output	212
JTAG Support.....	212
JTAG Connector	213
Adding a New J-Link Device.....	214
Testing J-Link Connectivity.....	216

Multi-Core Debugging.....	216
Setting Up a Scan Chain.....	217
How the Scan Chain Works.....	217
Configuring Multiple Debuggers.....	219
If It Still Does Not Work!.....	219
Summary.....	220
■ Chapter 15: Antennas.....	221
Antenna Specifications.....	221
About the Antenna Connections.....	222
SMA Connectors.....	222
Locating the SMA Connectors.....	223
Miniature Coaxial Connectors.....	224
Summary.....	227
■ Chapter 16: The API Kits.....	229
Samsung-Provided API Support.....	229
Documents to Gather.....	229
Where Are the APIs?.....	230
Compiled Object Code Libraries.....	231
I/O Structure.....	232
Finding Out About Devices.....	233
Temboo.....	234
Where Else to Look.....	234
Summary.....	234
■ Chapter 17: General Purpose Input/Output (GPIO).....	235
About GPIO in the ARTIK.....	235
Pin Modes.....	235
Digital Input.....	236
Digital Output.....	236

Exploring GPIO	237
Developer Board GPIO Pinouts	237
Reserved Pins	241
Active Levels	241
Slewing Rates	242
Resonant Circuits	243
Interacting With the Hardware	243
Hardware Header Pin Numbers.....	244
Using Arduino Emulation	246
GPIO Via sysfs.....	247
Using the sysfs Interface.....	249
GPIO Drivers	249
Device Base Addresses	250
GPIO Chip Numbers	251
Interacting With sysfs.....	253
Using bash With sysfs	254
/sys/kernel/debug/gpio	255
Accessing GPIO with the C Language.....	256
Using Boolean Data Types in C	256
Finding a GPIO Base Address.....	256
Pin Export to the User Domain.....	258
Previously Exported GPIO Pins.....	259
Creating a Dynamic Path to an Exported Pin	259
Pin Active LOW Setting	259
Pin Direction Setting	260
Digital Value Reading.....	261
Digital Value Setting	262
Edge Detecting	264
Releasing Exported Pins.....	264

Continuous Reads and Writes.....	265
Trailing Carriage Returns.....	265
Access Directly Via the Kernel.....	265
GPIO Ports.....	266
Shared Registers	269
GPIO Registers.....	269
Mapping the Bits	271
Port GPX0.....	274
Port GPX1	275
Port GPA0.....	276
Port GPA1	277
Port GPA2.....	278
Port GPD0	279
Port GPB2	280
Programming Via the Kernel Interface	281
Prototype Example Code.....	283
GPIO Pin Multiplexing	286
/sys/kernel/debug/pinctrl	286
Modifying the GPIO Pin Multiplexed State	287
Learn More About pinctrl Multiplexing	288
Summary	288
■ Chapter 18: Analog Input and IIO	289
Reading Analog Inputs.....	289
Analog ADC Pin Connections	289
Analog ADC Input with Arduino IDE	291
Using sysfs	291
Working Out the Base Address	292
Reading a Pin Voltage.....	292
Automatic Base Addresses in bash	293

Automatic Base Addresses in the C Language.....	294
Reading an ADC Value.....	297
Scaling the Raw Value	298
Using the /sys/bus Devices	298
About the New IIO Subsystem	300
Summary	301
■ Chapter 19: Pulse Width Modulated Output.....	303
What Is Pulse Width Modulation?	303
PWM Support in the ARTIK Modules.....	304
PWM Output Connectors	304
Using PWM with Arduino Calls	305
Accessing PWM via sysfs.....	305
PWM Entities in sysfs	306
/sys/class/pwm/pwmchip0	306
PWM Channel Properties	307
PWM Timing Control	308
Inverting the PWM Waveform	309
Configuring the PWM Interface with bash.....	309
Requesting a Channel Count	309
Exporting a Channel to the User Space	309
Setting the Timing Properties for a Channel.....	310
Turning On the Output Waveform.....	310
Detecting Whether the PWM Is Running.....	311
Inverting the Waveform	311
Turning Off the Output Waveform	311
Relinquishing a Channel.....	312
Programming PWM with the C Language.....	312
Requesting a Channel Count	312
Exporting a Channel to the User Space	313
Synthesizing a Path to the PWM Channel Container	313

Setting the Duration Value for a Channel.....	313
Setting the Duty Cycle for a Channel.....	314
Setting the Timing Properties for a Channel.....	315
Turning On the Output Waveform.....	315
Detecting Whether the PWM Is Running.....	316
Inverting the Waveform	317
Turning Off the Output Waveform	318
Relinquish a Channel.....	318
Utility Helper Function	319
PWM-Related AXT Connections.....	319
Summary.....	320
■ Chapter 20: Inter-Integrated Circuit (I2C)	321
What Is I2C?	321
How Does I2C Work?	323
I2C on the ARTIK Modules	326
I2C Tools	327
The Device Detector Tool (i2cdetect)	327
The Value Reading Tool (i2cget).....	328
The Value Setting Tool (i2cset).....	328
The Register Dump Tool (i2cdump).....	329
Accessing I2C via sysfs.....	329
I2C Device Nodes.....	329
I2C Containers and Properties	330
Built-in Drivers.....	330
Instantiating and Removing Devices	331
Access I2C from the C Language	332
Breakout Connections	332
I2C-Related AXT Connectors	333
Summary.....	334

- **Chapter 21: Serial Peripheral Interface (SPI) 335**
 - How Does SPI Work? 335
 - Interfacing Signal Lines..... 337
 - Alternative Naming Conventions 337
 - Transmitting Data 339
 - SPI Internal Architecture 339
 - Control Register (SPCR) 340
 - Status Register (SPSR) 341
 - Clock Rate Setting 342
 - Data Register (SPDR)..... 343
 - Interacting with an SPI Device 343
 - External Includes 344
 - Global Variables 344
 - Opening an SPI Device..... 344
 - Initializing the SPI Port 344
 - Reading and Writing Data 345
 - AX88796 Ethernet Controller (ARTIK 520) 346
 - AX88760 USB and Ethernet Controller (ARTIK 1020)..... 346
 - Arduino and SPI..... 346
 - Programming via the Arduino SPI Library..... 347
 - The Arduino SPI library 348
 - Multiplexing Digital Pins for SS Addressing 349
 - Summary 349
- **Chapter 22: Audio and Inter-IC Sound (I2S) 351**
 - Audio Capabilities..... 351
 - Audio Output..... 352
 - Audio Input 352
 - ALSA Audio Support 352
 - Exploring ALSA 353
 - Configuring ALSA..... 354

Upgrading ALSA	355
The Audio Mixer Tool (amixer)	355
The Audio Recorder Tool (arecord).....	356
The Audio Player Tool (aplay).....	357
The Audio/Visual Player Tool (mPlayer).....	358
Finding the Sound Card	359
Testing the Audio Outputs.....	360
Recording Audio	360
Playing Audio Files.....	361
Developing Audio Applications	362
Pulse Audio Support	363
How Does I2S Work?	363
The ARTIK Implementation.....	364
Sample Rates	365
Clock Timings	365
AK4953A - Stereo Codec Chip	366
Find Out More About I2S.....	367
Audio-Related AXT Connections	367
Audio Experiments	368
Summary	368
■ Chapter 23: Graphics and Video	369
About Graphics and Video	369
Graphics Support.....	370
Video Support.....	370
Display Connectors.....	370
Display Control	372
Video4Linux Support	373
Video Support in sysfs.....	374
Video Nodes	374
Multi-Format Codec	375

Programming Video4Linux	376
Connecting Video Cameras.....	376
ARTIK 5 - S5K6B2 Camera.....	377
ARTIK 10 - S5K3L2 Camera	378
ARTIK 10 - S5K5EA Camera.....	378
Recommended MIPI Camera	379
MIPI Camera Interfaces	379
USB-Attached Cameras	380
Image Capture Tool (fswebcam).....	380
Video Conversion Tool (ffmpeg).....	382
Media Streaming Server (ffserver).....	384
Video Processing Toolkit (gstreamer).....	385
AXT Connections	385
Summary.....	389
■ Chapter 24: Conclusions and Next Steps.....	391
Forensic Inspection	391
Reverse Engineering	391
Validating With Multiple Sources.....	392
Coping with Undocumented Features	392
Defensive Coding Strategies	394
Looking Over the Horizon	394
Porting Projects from Other Architectures.....	395
Hardware and Tools.....	395
Creating a Bare-Bones Breakout Board	395
Servo-Controlled Camera	396
Sample Source Code and Illustrations	396
How to Get More Help and Support.....	397
My Challenge to You	397
■ Bibliography	399
■ Index	401

About the Author



Cliff Wootton is an award-winning former Interactive TV systems architect at the BBC, specializing in Interactive TV, content management systems and digital video playout with commodity hardware. Previously invited as a guest speaker on pre-processing for video compression at the Apple WWDC developer conference, he has also taught about the IoT, real-world computing with Arduino, video compression, metadata, and how to build multimedia art installations in an MA course at the University of the Arts, London. Cliff now concentrates on research and development projects; building digital media tools for creating audiovisual content; multimedia; electronic book publishing; writing; teaching; and playing the bass guitar.

Acknowledgments

Huge thanks are due to the following people who created resources online and contributed ideas and practical help while I was writing this book. Their assistance was truly invaluable and is gratefully acknowledged.

- Jill Balzano and Jonathan Gennick at Apress for two projects in one year! Amazing. Great times, as always.
- Glenn Cameron at Samsung, who provided prototype hardware and technical resources.
- Fred Patton at Samsung for his excellent photographic contributions once again, for wrangling the ARTIK blogs, and for helping out with the technical review process.
- Kevin Sharp for his helpful contributions to the ARTIK blog.
- Robert Nelson for contributing extremely helpful online advice on how to build a new OS for ARTIK modules.
- All the artik.io Forum posters whose questions inspired me to cover interesting topics.
- Gaynor Bromley at Panasonic Electric Works UK Ltd for the AXT connector images.

Introduction

You are no longer a beginner in the Samsung ARTIK community. It is time to flex your coding muscles and extend your skills by trying out more advanced techniques.

The companion book, *Beginning Samsung ARTIK - A Guide for Developers*, focused on getting you started. That book concentrated on helping you get your development systems up and running, and showing you how to use the tools.

This reference book builds on what you learned earlier and takes you deeper into the ARTIK world. The narrative is not designed to be a linear reading experience. Reference books need to help you find things in a random fashion when you need them.

A lot of material in here was gathered by inspection and reverse engineering, coupled with the analysis of the schematic diagrams for the developer reference boards. I have presented this information in tables that are designed to help you as you design and build your own products. The coverage on the Samsung developer web site is an excellent introduction; if you read that information first, this book builds on what you find there and describes the foundations that underpin that material. The online sources of reference that I found are also described here for you to study.

The posts in the discussion forum suggested topics that readers would like to know more about. These posts inspired both books. The fruits of my own forensic research for *Beginning Samsung ARTIK* and further experimentation are distilled into this book to help you find what you need and quickly move on to more advanced things.

Things do not stand still for very long in the Samsung ARTIK world, and just as I was drawing things to a close, more new developer resources were released. I have incorporated as much of that new material as I can. Keep a close watch on the Samsung developer web site for more new material because there are still many new and exciting things to come.

—Cliff Wootton
Crowborough UK
Summer 2016

CHAPTER 1



Learning More About ARTIK

It is time to start stage two of your ARTIK journey. By now, you should have a working ARTIK development system up and running, and you should have explored the hardware and software. You are ready to move on to more complex challenges and develop a totally awesome and amazing project. If you don't have your system up and running yet, take a look at the companion book, *Beginning Samsung ARTIK*, for help in getting started.

This reference book uses the ARTIK command line environment and the GCC compiler to illustrate coding examples. Your terminal emulator should be set up and your serial connection working so you can interact with your ARTIK module. If you have access across your Local Area Network (LAN), you can use telnet. Experienced UNIX developers will feel at home in this environment. The command line environment is a powerful tool and it can seem daunting at first. It has a varied and useful collection of small tools and utilities. Each command does one thing very well. You can learn about them one by one. They are all easy to understand when looked at individually. The power comes from the way you combine the commands together to create processing workflows. The learning process is much easier than you might think.

To fully understand your ARTIK, you must be conversant with the developer reference board circuit schematics, U-Boot, the kernel, and operating system internals. Then you can begin to work with the peripheral interfaces such as GPIO and I2C. To get the whole picture, aggregate the knowledge from all of these resources with what you see by inspection. This is not a trivial task but if you tackle it step by step, it is possible to learn things quickly and it can be enjoyable as you become more adept.

Change and Evolution

The Samsung Developer Conference 2016 (SDC) introduced a lot of useful new technologies to help you with your ARTIK projects. There are more new things to come. Keeping up with the changes is a challenge in itself.

Model numbers that describe each different type of ARTIK have changed to accommodate potential future additions to the range. For now, the Commercial Beta of the ARTIK 5 is a Model 520 and the ARTIK 10 is a Model 1020. Updated versions of the ARTIK modules will use variations of that naming convention. The ARTIK modules are usually described as an ARTIK 5 or ARTIK 10 to distinguish between the two currently available products unless it makes sense to describe version-specific details.

ARTIK Cloud is the new name for the SAMI infrastructure that was documented in *Beginning Samsung ARTIK*. Because this forms such a crucial part of the IoT ecosystem, it has been adopted as part of the ARTIK product family. If you are using the first edition of *Beginning Samsung ARTIK* to help you get started, where you see SAMI mentioned, do a global search-and-replace in your mind to convert that to ARTIK Cloud. Read about developing for ARTIK Cloud at the Samsung developer web site at <https://developer.artik.cloud/>.

Electronic supplementary material The online version of this chapter (doi:[10.1007/978-1-4842-2322-2_1](https://doi.org/10.1007/978-1-4842-2322-2_1)) contains supplementary material, which is available to authorized users.

The **ARTIK IDE** is a specially designed development environment for creating and debugging applications for your ARTIK. It is based on the Eclipse CHE platform and it supports local and remote debugging. Consider the ARTIK IDE as an alternative to installing Eclipse and a cross-compiling toolchain as it might offer some advantages. Samsung will release more documentation for it as it becomes ready for deployment. Read about the CHE tools at the Eclipse web site at www.eclipse.org/che/artik/.

IoTivity is an IoT open source integration platform maintained by the Linux Foundation as a library to download and use in your own projects. It is designed to help you communicate from device to device. Find out more at the IoTivity web site at www.iotivity.org/.

The **Resin•io** system provides tools for updating your IoT systems remotely over the air. This can be a challenging problem to solve if you have to develop your own solution. Find out more at the Resin•io web site at <https://resin.io/>.

The SDC 2016 developer presentations and tutorials will all be made available on the developer web site in due course. There is also a YouTube channel that represents the developer support services. Reach the `samsung_dev` home page to access the video resources via one of these routes:

www.youtube.com/results?search_query=samsung_dev
www.youtube.com/user/SMInnov8

More recently, the ARTIK OS source code has been released under an open source license. There are also new, up-to-date, and complete data sheets for the ARTIK model 520 and 1020 modules that describe all of the hardware characteristics.

The open source components used in the ARTIK operating system will change. These all contribute to a gradual evolution and improvement of the ARTIK. These changes often happen without any fanfare. They are only discovered when they affect the development process or when something breaks as a consequence. Table 1-1 summarizes the potential areas you should expect to change and the possible consequences for technologies aside from the development work that the Samsung engineering team is working on.

Table 1-1. *Changes and Consequences*

Component	Consequences
Kernel	The kernel is maintained independently of the OS. There is a lot of work being done on the inside plumbing. Expect changes to <code>sysfs</code> , <code>procfs</code> , <code>runfs</code> , and <code>udevfs</code> as various driver items are moved around. Where appropriate, any expected changes are mentioned in this book when they may affect what you do. There are many components of the kernel that are maintained as separate projects. The IIO interface for analog inputs is a typical example.
Fedora OS	Fedora has been upgraded from version 20 in the Alpha ARTIK modules to version 22 in the Commercial Beta modules. The current developer open source code version of Fedora is somewhat more advanced than that. One of the later revisions will become the dominant version when the ARTIK OS is upgraded.
yum vs. dnf	One consequence of the latest upgrade is that the <code>yum</code> installer in the ARTIK module has been completely replaced by a new command called <code>dnf</code> . The <code>yum</code> command is no longer available. The <code>dnf</code> command is upwards compatible with <code>yum</code> but adds many new features. This was covered in <i>Beginning Samsung ARTIK</i> but to clarify things further, <code>dnf</code> is now always used inside the ARTIK module for installing packages directly there. On your development workstation, the installer depends on the version and revision of Linux. Your installer there may be <code>dnf</code> , <code>yum</code> , <code>rpm</code> , or <code>apt-get</code> . Be sure that you distinguish between commands you execute on the ARTIK command line and others that are only appropriate on your developer workstation command line.
Yocto	Changes to Yocto are generally benign and are concerned with making the OS build easier to manage.

(continued)

Table 1-1. (continued)

Component	Consequences
U-Boot	U-Boot has been stable for a long time; fundamental changes to how it works are rare.
ALSA	The audio tools revolve around the ALSA project and the mPlayer application. These are likely to be stable until the industry develops new technologies for audio systems. The I2S support for digital audio is very mature and not prone to changes.
Video4Linux	The video processing uses Video4Linux, which is moving forward as new TV formats are deployed to the broadcast industry. Most new work is with 4K video; on the horizon, 8K video may start to become popular from 2020 onwards. Video coding has used the H.264 standard for some years. That is well-supported by the <code>ffmpeg</code> tool, which is actively maintained as an open source project. The newer HEVC codecs will be more useful as the 4K video format is taken up by the industry. Upgrades to the MIPI standards will affect the V4L source code as they are incorporated.

Topics that were addressed in *Beginning Samsung ARTIK* are covered in more detail in this book. For example, hardware debugging with the JTAG port driven from a Segger J-Link was suggested as a way forward in *Beginning Samsung ARTIK*. In Chapter 14, that topic is revisited in much more depth to cover scan chains and recommended configurations of the debugger for diagnosing applications running in multiple cores. It is also possible to run several debug sessions simultaneously when you have factored your application into several separate processes.

Peripheral devices are dealt with in much greater detail and all the inner workings are discussed for GPIO, I2C, I2S, IIO, and SPI. Each interface has a chapter devoted entirely to the inner workings.

Your Journey Through This Book

The programming examples in this book are more sophisticated than the ones in *Beginning Samsung ARTIK*. They are designed to help you build and maintain reusable code libraries of your own to avoid reinventing the wheel every time you start a new project. Where *Beginning Samsung ARTIK* was designed to be a coherent narrative that took you step by step through the setup process, this reference guide is designed for random access to in-depth coverage of the topics. However, if you do want to read the whole book, the content is organized in a logical way, and each chapter builds on the earlier ones.

The first few chapters talk about the platform internals to help you construct a robust mental map of where everything is. It is essential to get to know the territory before you go on a quest for individual API components. Some of the topics that were addressed more simplistically in *Beginning Samsung ARTIK* are expanded in this book.

Understanding how the kernel builds the virtual file systems and maps the hardware devices to them is valuable knowledge. You can then choose the right approach for developing an application to interface with GPIO pins or I2C-driven sensors.

Topics dealt with in this book are occasionally quite deep and complex. I have tried to break them down into small and easily digested pieces. This makes it easier to assimilate the knowledge. As you read through the chapters, new concepts are revealed one by one.

Please do not be daunted by the complexity and scope of the technology inside the ARTIK. The development process becomes easier as you become more familiar with the technology, and the time devoted to understanding the internals of your ARTIK is well spent.

Samsung Developer Resources

Following SDC 2016, the previously available developer documentation has been updated with new material, and the Downloads page has new items added from time to time. June 2016 saw the release of the source code for the ARTIK OS and the ARTIK 10 modules via a Git repository. New firmware for the OS in the ARTIK 5 and 10 arrived also, and the developer documentation was updated with a lot of new material for users interested in using audio and video on their ARTIK projects.

The Samsung developer resources are organized into several categories. The documentation takes you through a straightforward getting-started process. This is useful if you are using a Windows workstation; there is some useful material for Linux users, too. For Linux and the Apple Macintosh development workstations, consult the companion book *Beginning Samsung ARTIK* for additional material.

The online developer guide provides some basic instruction on accessing GPIO pins. This is backed up here with code examples. Chapter 17 illustrates the GPIO internals in some detail. Check out the Samsung developer support channel on YouTube for video tutorials and presentations. Third-party material is available at the hackster.io web site or on the Instructables blog. Access more resources via these links:

<https://developer.artik.io/overview>
<https://developer.artik.io/documentation/>
<https://developer.artik.io/documentation/getting-started-beta/>
<https://developer.artik.io/documentation/developer-guide/>
<https://developer.artik.io/documentation/tutorials/>
<https://developer.artik.io/documentation/tutorials/color-mqtt-client.html>
www.hackster.io/
www.instructables.com/
www.youtube.com/channel/UC4rolvSm8ikmnydbbzNjw

Samsung Developer Downloads

Check the Samsung developer downloads page from time to time. Useful resources to help your development process are added when the Samsung engineers release new features. Look out for new firmware downloads to install. Installing new firmware wipes out anything that you previously installed in your ARTIK module, including any configuration or experimental applications you might have written and compiled there. Make sure that anything important to you is documented and backed up before installing new firmware. The following is the link to the Downloads page. Sign in with your Samsung developer account to reach it.

<https://developer.artik.io/downloads>

Getting copies of everything on the Downloads page should be a routine activity; once a month you should check to see if there is anything new. Collect an archive of older versions to track the evolution of the ARTIK family and to be able to revert back to an earlier configuration if necessary. Download these items and read them for useful coverage of your ARTIK internals:

- Developer board schematics (Type 5 and Type 10)
- Data sheets (ARTIK models 520 and 1020)
- Software developers guide (not yet released as of July 2016)

ARTIK OS Source Code

Now that the Commercial Beta ARTIK 1020 has been released, the new downloadable developer resources also include Git repositories where the source code for the ARTIK OS is maintained. Check this web page regularly to see if new components have been introduced at the head of the Samsung ARTIK code repository: <https://github.com/SamsungARTIK>. Table 1-2 describes the items that have been released.

Table 1-2. ARTIK Git Source Code Repositories

Component	Description
build-artik	This code helps to create an ARTIK SD Fuse image. This is loaded onto a micro SD card to install new software into the eMMC memory of your ARTIK. Due to the long build time of a Fedora operating system image, the root file system is provided by a pre-built binary. Download it from the server during the build process. The binary image of the Fedora OS is built as a separate process.
fedora-spin-kickstarts	Creates an ARTIK Fedora root file system. The ARTIK Fedora OS is based on Fedora 22 ARM version and is customized for ARTIK 5 and 10 modules.
initrd-artik	Initial boot time RAM disk source for ARTIK 5 and 10. It recovers the eMMC partitions from the micro SD card Fuse image.
linux-artik	Linux kernel source for ARTIK 5 and 10. The base kernel version of ARTIK is Linux version 3.10.93 and is conditionally compiled as the Samsung Exynos kernel.
u-boot-artik	This is the U-Boot source for ARTIK 5 and 10. It is based on the Samsung Exynos U-Boot variant.
slate	The Slate API documents generator helps you create beautiful, intelligent, responsive API documentation.
artik-discourse	Discourse is a discussion platform built for the next decade of the Internet. It works as a mailing list, discussion forum, or a long-form chat room.
heroku-buildpack-nodejs-grunt	This is a slightly modified version of the official Heroku buildpack for Node.js applications with added support for the Grunt notification system.
nginx-buildpack	Runs the NGINX web delivery platform on your application server that runs Heroku.
heroku-buildpack-webpack	This is a Heroku buildpack for web applications that use webpack.
heroku-buildpack-php	The official PHP buildpack for Heroku.

The following are the links for the repositories containing ARTIK OS components that have already been released. Put them into your Git client tools to map them into your workspace and replicate the repository into your developer workstation to have a copy of the source code to work on:

<https://github.com/SamsungARTIK/build-artik>
<https://github.com/SamsungARTIK/fedora-spin-kickstarts>
<https://github.com/SamsungARTIK/initrd-artik>
<https://github.com/SamsungARTIK/linux-artik>
<https://github.com/SamsungARTIK/u-boot-artik>

The following repositories contain useful utilities to help you manage your development process; they are not part of the canonical ARTIK OS source kit:

<https://github.com/SamsungARTIK/slate>
<https://github.com/SamsungARTIK/artik-discourse>