
Table of Contents

Preface	15
Chapter 1 • Getting started	19
1.1 • A Micro what?	19
1.2 • The 32-bit Processor	24
1.3 • Microcontroller Programs	25
1.4 • The Clock	30
1.5 • The Fundamentals	32
1.5.1 • Number Systems	32
1.5.2 • Ohm’s Law	36
1.6 • Switches	37
1.7 • Light Emitting Diodes (LEDs)	39
1.8 • Transistors	42
1.9 • Power Supplies	46
1.9.1 • Linear Regulators	47
1.9.2 • Switching Power Supplies	49
1.10 • Development Board Hardware	50
1.11 • Block Diagram	51
1.12 • Power Input	52
1.13 • Main processor, JTAG, clocks, and connections	53
1.14 • ANT 2.4GHz Transceiver	55
1.15 • User IO: LCD, LEDs, Buttons, Beeper	56
1.16 • J-Link On-Board	56
1.17 • Summary	59
Chapter 2 • Development Environment & Version Control	61
2.1 • IAR Integrated Development Environment	61
2.2 • IAR Installation	61
2.3 • Setting Up a New Project	63
2.4 • Files in a New Project	71
2.5 • IAR Simulator and Debugger	75
2.6 • Other Development Tools	81

2.6.1 • Tera Term	81
2.6.2 • GitHub Desktop.	82
2.6.3 • ANTWare II	82
2.6.4 • nRFgo Studio	82
2.6.5 • Windows Settings	83
2.7 • Version control with Git	83
Chapter 3 • ARM Cortex-M3 Assembly Language.	91
3.1 • Core Registers	91
3.2 • Instructions	93
3.3 • Assembly Language Syntax	94
3.4 • Load and Store Instructions	101
3.5 • Hello World in Assembly.	103
Chapter 4 • Embedded C.	115
4.1 • Documentation	115
4.2 • Doxygen	117
4.2.1 • Documenting the “right information”	117
4.2.2 • Create a Configuration File	118
4.2.3 • Special Comment Blocks	120
4.2.4 • Doxygen tags in code	122
4.2.5 • Doxygen Example	123
4.3 • Coding Conventions.	125
4.3.1 • Type Definitions	125
4.3.2 • Hungarian Notation	126
4.3.3 • Preprocessor Symbol Definitions	127
4.3.4 • Braces { }	127
4.3.5 • Switch statements.	127
4.3.6 • White Space	128
4.3.7 • Global Variables	128
4.3.8 • Doxygen Tags	129
4.3.9 • Function Declarations.	129
4.3.10 • State Declarations	130
4.3.11 • Tabs and Indenting	130

4.3.12 • Operator Precedence	130
4.4 • C project file overview	131
4.4.1 • Accessing Registers	134
4.5 • How A Processor Starts Up	137
4.5.1 • Watchdog Timer	137
4.5.2 • Clock and Power Initialization	139
4.5.3 • Implementing the clock setup	141
4.6 • GPIO Initialization	143
4.7 • Program Structures	144
4.7.1 • The Infinite Loop	144
4.7.2 • Operating Systems	145
4.7.3 • State Machine Super Loop	147
4.8 • Implementing the SM Super Loop	150
4.8.1 • Initialization	151
4.8.2 • State Machine Super loop	151
4.9 • helloworld.c	154
4.10 • Next Steps	158
Chapter 5 • GPIO & LED Driver	161
5.1 • SAM3U2 General Purpose Input Output	161
5.2 • External Hardware	162
5.2.1 • Pin Allocation	163
5.3 • The PIO Peripheral	168
5.4 • PIO Internal Hardware	169
5.4.1 • Logic Block Diagram	170
5.5 • PIO Registers	174
5.6 • Adding a New Task	178
5.7 • The LED Driver	181
5.8 • Driver Implementation	184
5.9 • Blinking	188
5.9.1 • Map File	190
5.10 • Chapter Exercise	192

Chapter 6 • Interrupts & Button Drivers	193
6.1 • Interrupts	193
6.2 • Interrupts on the SAM3U2	194
6.2.1 • Interrupts depend on hardware.	195
6.2.2 • Interrupts need to be configured by firmware	195
6.2.3 • Interrupts can be enabled and disabled globally	196
6.2.4 • An interrupt forces the processor to run an Interrupt Service Routine	196
6.2.5 • Interrupts have priorities	197
6.2.6 • Interrupts can (and will) occur anytime, anywhere	198
6.2.7 • Interrupts require context preservation	198
6.2.8 • Interrupts set flags that need to be cleared	199
6.2.9 • ISRs should be short and fast.	199
6.3 • Interrupt User Guide Resources	199
6.4 • Interrupts and C	203
6.4.1 • Vector Table	203
6.4.2 • Priorities	203
6.4.3 • Enabling and Disabling Peripheral Interrupt Sources	204
6.5 • Peripheral Interrupts	207
6.5.1 • GPIO Interrupts	207
6.5.2 • Timer / Counter Interrupts	207
6.5.3 • Communication Peripheral Interrupts.	207
6.5.4 • Other Peripheral Interrupts.	208
6.6 • Button Driver Overview and Setup	208
6.6.1 • Debouncing	208
6.6.2 • Button history or edge detection	209
6.6.3 • Button held	209
6.7 • Button Operation	209
6.7.1 • Button Typedefs	211
6.8 • PIO Interrupts	212
6.9 • Button API	222
6.10 • Chapter Exercise	224

Chapter 7 • Sleep, System Tick and Timer Peripheral	225
7.1 • Sleep	225
7.2 • System Tick Configuration	226
7.2.1 • Tick Time and CTRL INIT value	227
7.3 • Timer Peripheral	233
7.4 • Timer Counter Highlights	235
7.5 • Timer Counter Registers	237
7.6 • Timer Driver	239
7.7 • Timer API	240
7.8 • Chapter Exercise	245
Chapter 8 • Pulse Width Modulation	247
8.1 • PWM Concepts	247
8.2 • PWM the Easy Way: SAM3U2 PWM Peripheral	249
8.3 • Peripheral Highlights	250
8.4 • PWM and Audio	251
8.5 • EiE Audio Hardware	252
8.6 • PWM Registers	253
8.7 • Development Board Audio Driver	256
8.7.1 • Audio function initialization	256
8.7.2 • Audio API Functions	258
8.7.3 • PWMAudioOn() and PWMAudioOff()	261
8.8 • PWM the Hard Way: Bit Bashing	262
8.9 • LED PWM Design	263
8.10 • Audio Bits	268
8.11 • Multiple User Tasks	270
8.12 • Chapter Exercise	272
Chapter 9 • DMA and Messaging	273
9.1 • Data Transmission	273
9.2 • Resource Conflicts	275
9.3 • Direct Memory Access – DMA	276
9.3.1 • PDC Registers	279
9.3.2 • PDC Interrupts	281

9.3.3 • Transmitting with DMA	282
9.3.4 • Receiving with DMA	283
9.4 • Linked Lists	284
9.5 • Hard Faults	286
9.6 • EiE Messaging Task	289
9.6.1 • Message Task Data Structures	290
9.6.2 • Message Task Protected Functions	295
9.7 • Messaging Public Functions	305
9.7.1 • QueryMessageStatus().	305
9.8 • Messaging State Machine	306
9.9 • Chapter Exercise.	307
Chapter 10 • Serial and Bugs for Breakfast.	309
10.1 • RS-232 Overview	309
10.1.1 • Clocking	312
10.1.2 • Signaling	313
10.2 • Data Errors.	315
10.3 • ASCII.	316
10.4 • Storing and Displaying Characters.	317
10.5 • SAM3U2 UART Peripheral	320
10.5.1 • Peripheral Highlights	321
10.5.2 • Baud Rate Generator	322
10.6 • UART Registers	324
10.6.1 • EiE UART Driver.	326
10.6.2 • UART Task Data Structures	327
10.6.3 • UART Driver Functions	329
10.7 • UART Interrupts	335
10.8 • UART Driver Design	336
10.8.1 • Data Transmit	336
10.8.2 • Data Receive.	340
10.9 • Dynamic Memory Allocation	342
10.10 • Debug Task.	345
10.11 • Debug API	346

10.11.1 • DebugPrintf()	346
10.11.2 • DebugPrintNumber()	347
10.12 • Reading Character Input.	349
10.12.1 • DebugInitialize()	350
10.12.2 • DebugRxCallback()	351
10.13 • Debug Programmer Access.	352
10.14 • Terminal Control Codes	361
10.15 • Chapter Exercise	362
Chapter 11 • I SPI with my I2C	363
11.1 • SPI Signaling	363
11.2 • SPI Hardware	368
11.3 • SPI Registers	369
11.4 • EiE SPI Driver	373
11.5 • Master Transmit	375
11.6 • Master Receive	375
11.7 • Slave Transmit	376
11.8 • Slave Receive	376
11.9 • Slave Transmit with Flow Control.	376
11.10 • Slave Receive with Flow Control	377
11.11 • Chip Select	377
11.12 • SPI Data Structures	378
11.13 • SPI Driver Functions in Common with UART	380
11.14 • New SPI Driver Functions	381
11.14.1 • SspRequest()	381
11.14.2 • SspRelease()	382
11.14.3 • SspAssertCS() / SspDeassertCS()	383
11.14.4 • SspReadByte() / SspReadData()	384
11.14.5 • SspQueryReceiveStatus()	385
11.14.6 • SspGenericHandler()	386
11.15 • Ssp State Machine	392
11.16 • Blade Daughter Board Interface	396
11.17 • Blade Example Project	398

11.17.1 • Blade Firmware Configuration Defaults and Interface	398
11.17.2 • UserApp1Initialize()	401
11.17.3 • UserApp1SM	402
11.18 • Chapter Exercise	403
Chapter 12 • I²C & ASCII LCD	405
12.1 • Inter-Integrated Circuit (I ² C) Communication	405
12.2 • I ² C Hardware	406
12.3 • I ² C Signaling	408
12.4 • EiE TWI Hardware	411
12.5 • I ² C (TWI) on SAM3U2	411
12.6 • TWI and PDC	412
12.7 • TWI Registers	414
12.8 • TWI Driver	416
12.8.1 • TWI Data Structures	417
12.8.2 • TWI Driver Functions	418
12.9 • TWI State Machine and ISR	423
12.9.1 • TWI Transmit	423
12.10 • TWI Receive	428
12.10.1 • NACK and Errors	431
12.11 • ASCII LCD	433
12.11.1 • LCD Hardware	433
12.11.2 • LCD Controllers	435
12.11.3 • LCD Interface	436
12.12 • Character and Control Data	437
12.13 • Using the LCD Controller	438
12.14 • Control byte with Co and Rs	440
12.14.1 • Character RAM Addresses	441
12.14.2 • LCD Command Set	442
12.14.3 • LCD Initialization	443
12.15 • LCD Application	446
12.15.1 • LcdCommand()	447
12.15.2 • LcdMessage()	447

12.15.3 • LcdClearChars()	448
12.16 • Chapter Exercise	449
Chapter 13 • Analog to Digital Conversion	451
13.1 • ADC background	451
13.1.1 • Quantization	451
13.1.2 • Sampling	451
13.1.3 • Bandwidth and Aliasing	452
13.1.4 • Nyquist Frequency	453
13.1.5 • Resolution	454
13.1.6 • Clipping	455
13.2 • Characteristics of ADCs	455
13.2.1 • Precision, Error, and ENOB	456
13.2.2 • Missing codes	457
13.2.3 • Reference Voltages	457
13.2.4 • Noise	457
13.2.5 • Single vs. Differential Measurement	458
13.2.6 • Signal Conditioning	458
13.3 • EiE ADC Hardware	460
13.4 • SAM3U2 12-bit ADC Peripheral	460
13.4.1 • ADC Registers	462
13.5 • EiE ADC Driver	464
13.5.1 • ADC Initialization	466
13.5.2 • ADC Interrupt	467
13.5.3 • ADC State Machine	468
13.6 • EiE ADC API	468
13.6.1 • void Adc12AssignCallback()	468
13.6.2 • bool Adc12StartConversion()	469
13.7 • Chapter Exercise	470
Chapter 14 • ANT Radio System	473
14.1 • ANT Wireless Radio	474
14.2 • Building the ANT Stack	475
14.2.1 • ANT Physical Layer	476

14.3 • ANT Message Protocol and Usage	477
14.3.1 • ANT Protocol: Sections 1 thru 4.	477
14.3.2 • ANT Protocol Section 5: Channel Parameters	478
14.3.3 • ANT Channel ID.	479
14.3.4 • Transmit Data Types	479
14.3.5 • ANT Protocol Section 6: Pairing	481
14.3.6 • ANT Protocol Section 7: ANT Interface	482
14.3.7 • ANT Protocol Sections 8 and 9: Examples and Appendix.	483
14.4 • Message Handling	483
14.4.1 • Messaging with Channel Closed.	484
14.4.2 • Messages When Channel is Open.	485
14.5 • Debugging an ANT System	486
14.6 • Programming the ANT Sub-System	488
14.6.1 • Firmware Design ant.c and ant_api.c	488
14.6.2 • Data Structures.	490
14.6.3 • Serial Drivers and ANT hardware interface	491
14.6.4 • Task access to send and receive	517
14.6.5 • ANT_TICK and ANT_DATA.	518
14.7 • ANT State Machine	526
14.7.1 • Initializing the ANT SM.	526
14.8 • Implementing the ANT State Machine	529
14.9 • API Summary	532
14.9.1 • ANT Configuration and Status Message	533
14.9.2 • ANT Data Messages	536
14.10 • Chapter Exercise	539
14.11 • Conclusion	540
Glossary	541
Index	547