

Table of Contents

Chapter 1.	See a Program Running.....	1
1.1	Translate a C Program into a Machine Program	1
1.2	Load a Machine Program into Memory	4
1.2.1	Harvard Architecture and Von Neumann Architecture	4
1.2.2	Creating Runtime Memory Image.....	7
1.3	Registers.....	11
1.3.1	Reusing Registers to Improve Performance.....	12
1.3.2	Processor Registers	13
1.4	Executing a Machine Program.....	17
1.4.1	Loading a Program	18
1.4.2	Starting the Execution.....	19
1.4.3	Program Completion.....	23
1.5	Exercises.....	24
Chapter 2.	Data Representation.....	27
2.1	Bit, Byte, Halfword, Word, and Double-word	27
2.2	Binary, Octal, Decimal, and Hexadecimal Numbers	29
2.3	Unsigned Integers	30
2.4	Signed Integers	31
2.4.1	Sign-and-Magnitude.....	33
2.4.2	One's Complement	34
2.4.3	Two's Complement	35
2.4.4	Carry Flag for Unsigned Addition or Subtraction	36
2.4.5	Overflow Flag for Signed Addition or Subtraction	39
2.4.5.1	Interpreting the Carry and Overflow Flags	42
2.4.5.2	Two's Complement Simplifies Hardware Implementation.....	45
2.5	Character String.....	49
2.6	Exercises.....	53
Chapter 3.	ARM Instruction Set Architecture	55
3.1	ARM Assembly Instruction Sets	55
3.2	ARM Cortex-M Organization.....	58
3.3	Going from C to Assembly.....	60
3.4	Assembly Instruction Format	63
3.5	Anatomy of an Assembly Program	65
3.6	Assembly Directives.....	69
3.7	Exercises.....	73
Chapter 4.	Arithmetic and Logic.....	75
4.1	Program Status Register	75
4.2	Updating Program Status Flags	77
4.3	Shift and Rotate.....	78

4.4	Arithmetic Instructions	80
4.4.1	Addition and Subtraction Instructions	81
4.4.2	Short Multiplication and Division Instructions	82
4.4.3	Long Multiplication Instructions.....	82
4.4.4	Saturation Instructions	83
4.5	Barrel Shifter	83
4.6	Bitwise Logic Operations	84
4.7	Reversing the Order of Bits and Bytes	89
4.8	Sign and Zero Extension	90
4.9	Data Comparison	91
4.10	Data Movement between Registers	92
4.11	Bit Field Extract	93
4.12	Exercises	94
Chapter 5.	Load and Store	97
5.1	Load Constant into Registers	97
5.1.1	Data Movement Instruction MOV and MVN	97
5.1.2	Pseudo Instruction LDR and ADR.....	98
5.1.3	Comparison of LDR, ADR, and MOV	99
5.2	Big and Little Endian	100
5.3	Accessing Data in Memory	101
5.4	Memory Addressing	101
5.4.1	Pre-index, Post-index, and Pre-index with Update	101
5.4.2	Load and Store Instructions.....	103
5.4.3	PC-relative Addressing.....	104
5.4.4	Example of Accessing an Array	105
5.5	Loading and Storing Multiple Registers	106
5.6	Exercises	108
Chapter 6.	Branch and Conditional Execution	111
6.1	Condition Testing	111
6.2	Branch Instructions	114
6.3	Conditional Execution	117
6.4	If-then Statement	118
6.5	If-then-else Statement	121
6.6	For Loop	122
6.7	While Loop	123
6.8	Do While Loop	124
6.9	Continue Statement	125
6.10	Break Statement	126
6.11	Switch Statement	126
6.12	Exercises	129
Chapter 7.	Structured Programming	133
7.1	Basic Control Structures	133

7.2	Register Reuse	138
7.3	Example of Factorial Numbers	141
7.4	Example of Counting Ones in a Word	142
7.5	Example of Finding the Maximum of an Array	144
7.6	Example of Counting Digits	146
7.7	Example of Parity Bit	147
7.8	Example of Perfect Numbers	149
7.9	Example of Armstrong Numbers	151
7.10	Example of Palindrome String.....	152
7.11	Example of Converting String to Integer (atoi)	154
7.12	Example of Binary Search	155
7.13	Example of Bubble Sort.....	157
7.14	Exercises.....	159
Chapter 8.	Subroutines	161
8.1	Calling a Subroutine.....	162
8.2	Stack.....	164
8.3	Implementation of Stack via STM and LDM.....	165
8.4	Preserving Runtime Environment via Stack.....	166
8.5	Passing Arguments to Subroutine via Registers.....	169
8.5.1	Pass a Variable by Value and by Reference	170
8.5.2	Example of Passing by Value	173
8.5.3	Write a Subroutine in Different Files	175
8.5.4	Example of Passing by Reference.....	176
8.5.5	Example of Greatest Common Divisor	177
8.5.6	Example of Concatenating Two Strings.....	179
8.5.7	Example of Comparing Two Strings	180
8.5.8	Example of Inserting an Integer into a Sorted Array.....	181
8.5.9	Example of Converting Integer to String (itoa)	182
8.5.10	Example of Matrix Transpose.....	184
8.5.11	Example of Removing a Character from a String	186
8.5.12	Example of Finding Unique Numbers in an Array	187
8.6	Passing Arguments through Stack	190
8.7	Recursive Functions	192
8.7.1	Example of Factorial Numbers.....	194
8.7.2	Example of Reversing a String.....	196
8.7.3	Example of String Permutation	197
8.8	Exercises.....	199
Chapter 9.	64-bit Data Processing	203
9.1	64-bit Addition	203
9.2	64-bit Subtraction.....	204
9.3	64-bit Counting Leading Zeros.....	205
9.4	64-bit Sign Extension	205

9.5	64-bit Logic Shift Left.....	206
9.6	64-bit Logic Shift Right.....	207
9.7	64-bit Multiplication	208
9.8	64-bit Unsigned Division.....	209
9.9	64-bit Signed Division.....	211
9.10	Exercises	213
Chapter 10. Mixing C and Assembly		215
10.1	Data Types and Access	216
10.1.1	Signed or Unsigned Integers.....	216
10.1.2	Data Alignment.....	217
10.1.3	Data Structure Padding.....	219
10.2	Special Variables	222
10.2.1	Static Variables	222
10.2.2	Volatile Variables	226
10.3	Inline Assembly.....	228
10.3.1	Assembly Functions in a C Program.....	228
10.3.2	Inline Assembly Instructions in a C Program	229
10.4	Calling Assembly Subroutines from a C Program.....	230
10.4.1	Example of Calling an Assembly Subroutine	230
10.4.2	Example of Accessing C Variables in Assembly	231
10.5	Calling C Functions from Assembly Programs	232
10.5.1	Example of Calling a C Function.....	232
10.5.2	Example of Accessing Assembly Data in a C Program	233
10.6	Exercises	234
Chapter 11. Interrupts.....		237
11.1	Introduction to Interrupts	237
11.2	Interrupt Numbers	238
11.3	Interrupt Service Routines.....	240
11.4	Interrupt Vector Table.....	241
11.5	Interrupt Stacking and Unstacking.....	243
11.6	Nested Vectored Interrupt Controller (NVIC).....	245
11.6.1	Enable and Disable Peripheral Interrupts.....	247
11.6.2	Interrupt Priority.....	249
11.6.3	Global Interrupt Enable and Disable	253
11.7	System Timer	254
11.8	External Interrupt.....	262
11.9	Software Interrupt	266
11.10	Exercises	267
Chapter 12. Fixed-point and Floating-point Arithmetic.....		269
12.1	Fixed-point Arithmetic	270
12.1.1	Unsigned Fixed-point Representation	271
12.1.2	Signed Fixed-point Representation	272

12.1.3	Converting to Fixed-point Format.....	273
12.1.4	Fixed-point Range and Resolution Tradeoff	274
12.1.5	Fixed-point Addition and Subtraction	275
12.1.6	Fixed-point Multiplication.....	277
12.1.7	Fixed-point Division.....	278
12.2	Floating-point Arithmetic.....	279
12.2.1	Floating-point Representation	279
12.2.2	Special Values.....	284
12.2.3	Overflow and Underflow.....	285
12.2.4	Subnormal Numbers	286
12.2.5	Tradeoff between Numeric Range and Resolution.....	287
12.2.6	Rounding Rules.....	289
12.3	Software-based Floating-point Operations	292
12.3.1	Floating-point Addition	293
12.3.2	Floating-point Multiplication	297
12.4	Hardware-based Floating-point Operations	299
12.4.1	FPU Registers	299
12.4.1.1	Floating-point General-purpose Registers (S0-S31, or D0-D15)	299
12.4.1.2	Coprocessor Access Control Register (CPACR)	301
12.4.1.3	Floating-point Status and Control Register (FPSCR)	302
12.4.1.4	Floating-point Context Address Register (FPCAR)	304
12.4.1.5	Floating-point Context Control Register (FPCCR)	306
12.4.2	Load and Store Floating-point Numbers.....	308
12.4.3	Copy Floating-point Numbers	309
12.4.4	Copy and Set the Status and Control Register	309
12.4.5	Single-precision Arithmetic Operations	310
12.4.6	Single-precision comparisons.....	311
12.4.7	Precision Conversion.....	312
12.4.8	FPU Exception and Exception handling.....	314
12.4.9	Example Assembly Programs	316
12.4.9.1	Look up a Float Array.....	316
12.4.9.2	Sine Function of Argument in Radians.....	317
12.5	Exercises.....	319
Chapter 13.	Instruction Encoding and Decoding.....	323
13.1	Tradeoff between Code Density and Performance	323
13.2	Dividing Bit Streams into 16- or 32-bit Instructions	324
13.3	Encoding 16-bit Thumb Instructions.....	326
13.4	Encoding 32-bit Instructions	327
13.5	Calculating Target Memory Address	329
13.6	Instruction Decoding Example 1.....	330
13.7	Instruction Decoding Example 2.....	334
13.8	Exercises.....	339
Chapter 14.	General Purpose I/O (GPIO).....	341

14.1	Introduction to General Purpose I/O (GPIO)	341
14.2	GPIO Input Modes: Pull Up and Pull Down	342
14.3	GPIO Input: Schmitt Trigger	343
14.4	GPIO Output Modes: Push-Pull and Open-Drain	346
14.4.1	GPIO Push-Pull Output	346
14.4.2	GPIO Open-Drain Output.....	347
14.5	GPIO Output Speed: Slew Rate	350
14.6	Memory-mapped I/O	351
14.7	Lighting up an LED	355
14.8	Push Button	360
14.9	Keypad Scan	365
14.10	Exercises	371
Chapter 15.	General-purpose Timers	373
15.1	Timer Organization and Counting Modes	373
15.2	Compare Output	376
15.2.1	Setting Output Mode.....	377
15.2.2	Example of Toggling LED.....	379
15.2.3	Timer Update Events	382
15.3	PWM Output	384
15.3.1	PWM Alignment	389
15.3.2	PWM Programming Flowchart	391
15.4	Input Capture	395
15.4.1	Configuring Input Capture	399
15.4.2	Input Capture in Slave Mode with Reset.....	404
15.4.3	Interfacing to Ultrasonic Distance Sensor	407
15.5	Exercises	413
Chapter 16.	Stepper Motor Control	415
16.1	Bipolar and Unipolar Stepper Motor	415
16.2	Step Angle	417
16.3	Wave Stepping	418
16.4	Full Stepping	419
16.5	Half Stepping	421
16.6	Micro-stepping	423
16.7	Driving Stepper Motor	425
16.8	Exercises	426
Chapter 17.	Liquid-crystal Display (LCD)	427
17.1	Static Drive	428
17.2	Multiplexed Drive	429
17.3	STM32L Internal LCD Driver	433
17.3.1	Basic Introduction.....	433
17.3.2	Generic LCD Driver to Display Strings.....	439
17.4	Interfacing with External Character LCD Controllers	444

- 17.4.1 External Connection Diagram 444
- 17.4.2 Internal Font Encoding 447
- 17.4.3 Sending Commands and Data to LCD 448
- 17.4.4 Programming Fonts 451
- 17.5 Exercises 452**
- Chapter 18. Real-time Clock (RTC) 453**
- 18.1 Epoch Time 453
- 18.2 RTC Frequency Settings 455
- 18.3 Oscillator Frequency Accuracy 456
- 18.4 Binary Coded Decimal (BCD) Encoding 457
- 18.5 RTC Initialization 459
- 18.6 RTC Alarm 462
- 18.7 Using RTC to Wake Processors up from Sleep Mode 465
- 18.8 Exercises 468
- Chapter 19. Direct Memory Access (DMA) 469**
- 19.1 Advanced Microcontroller Bus Architecture (AMBA) 470
- 19.2 Interfacing a Peripheral without and with DMA 472
- 19.3 DMA Channels 474
- 19.4 Programming DMA 476
- 19.5 DMA Circular Mode 478
- 19.6 DMA Interrupts 479
- 19.7 Exercises 480
- Chapter 20. Analog-to-Digital Converter (ADC) 481**
- 20.1 ADC Architecture 481
 - 20.1.1 Digital Quantization 482
 - 20.1.2 Sampling and Hold 484
- 20.2 ADC Sampling Error 485
- 20.3 ADC Diagram 487
- 20.4 ADC Conversion Modes 488
 - 20.4.1 One Input Channel 488
 - 20.4.2 Multiple Input Channels 489
- 20.5 ADC Data Alignment 490
- 20.6 ADC Input Channels 491
- 20.7 ADC Triggers 493
- 20.8 Measuring the Input Voltage 496
- 20.9 ADC Configuration Flowchart 496
- 20.10 ADC with DMA 501
- 20.11 DMA with Ping-Pong Buffering 503
- 20.12 ADC Calibration 505
- 20.13 Exercises 506
- Chapter 21. Digital-to-Analog Converter (DAC) 507**

21.1	DAC Architecture	507
21.2	DAC on STM32L Processors	508
21.3	Conversion Trigger	510
21.4	Buffered Output	511
21.5	Generating a Sinusoidal Wave via Table Lookup	512
21.6	DAC with Software Trigger	515
21.7	Using Timer as a Trigger to DAC	517
21.8	Musical Synthesizing	520
21.8.1	Musical Pitch.....	520
21.8.2	Musical Duration	521
21.8.3	Amplitude Modulation of Tones.....	521
21.9	Exercises	526
Chapter 22.	Serial Communication Protocols	527
22.1	Universal Asynchronous Receiver and Transmitter	527
22.1.1	Communication Frame	528
22.1.2	Baud Rate.....	530
22.1.3	UART Standards	531
22.1.4	UART Communication via Polling	533
22.1.5	UART Communication via Interrupt	537
22.1.6	UART Communication via DMA	540
22.1.7	Serial Communication to Bluetooth Module.....	543
22.1.7.1	Bluetooth Transfer Protocols	543
22.1.7.2	Pairing Bluetooth Modules	544
22.2	Inter-Integrated Circuit (I²C)	546
22.2.1	I ² C Pins	546
22.2.2	I ² C Protocol	547
22.2.3	I ² C Data Frame	549
22.2.4	Interfacing Serial Digital Thermal Sensors via I ² C	551
22.2.5	I ² C Programmable Timings	553
22.2.5.1	Rise time and fall time	553
22.2.5.2	Data hold time	555
22.2.5.3	Data setup time.....	556
22.2.5.4	Master clock's minimum high and low time	556
22.2.5.5	Example of setting the I ² C timing.....	557
22.2.6	Sending Data to I ² C Slave via Polling	559
22.2.7	Receiving Data from I ² C Slave via Polling	560
22.2.8	Interfacing a Temperature Sensor via Polling	561
22.2.9	Transferring Data via DMA on I ² C Master	566
22.3	Serial Peripheral Interface Bus (SPI)	568
22.3.1	Data Exchange	569
22.3.2	Clock Configuration	571
22.3.3	Using SPI to Interface a Gyroscope	572
22.4	Universal Serial Bus (USB)	578

- 22.4.1 USB Bus Layer 579
- 22.4.2 USB Device Layer 581
- 22.4.3 USB Function Layer 584
 - 22.4.3.1 USB Descriptors 584
 - 22.4.3.2 Endpoint-Oriented Communication 587
 - 22.4.3.3 USB Enumeration..... 588
- 22.4.4 USB Class Layer 592
- 22.4.5 Human Interface Device (HID) 593
- 22.5 Exercises..... 598**
- Chapter 23. Multitasking599**
- 23.1 Processor Mode and Privilege Level 599**
 - 23.1.1 Control Register 600
 - 23.1.2 Exception Return Value (EXC_RETURN) 601
 - 23.1.3 Selection of MSP and PSP in Thread Mode..... 602
- 23.2 Supervisor Call (SVC)..... 603**
- 23.3 CPU Scheduling..... 607**
- 23.4 Example of Round Robin Scheduling 609**
- 23.5 Exercises..... 616**
- Chapter 24. Digital Signal Processing (DSP)617**
- 24.1 Fixed-point and Floating-point DSP 617**
- 24.2 Fixed-point Data Types in DSP 618**
- 24.3 Saturation..... 619**
- 24.4 Arithmetic Instructions 621**
 - 24.4.1 Parallel 8-bit Add and Subtract 623
 - 24.4.2 Parallel 16-bit Add and Subtract 627
 - 24.4.3 32-bit Add and Subtract 629
 - 24.4.4 Sum of Absolute Difference 630
 - 24.4.5 Extension and Add 631
 - 24.4.6 Add and Subtract Halfwords with Exchange 633
 - 24.4.7 16-bit and 32-bit Multiplication 635
 - 24.4.8 16-bit Multiply and Accumulate with 64-bit Result 638
 - 24.4.9 16-bit Multiply and Accumulate with 32-bit Result 641
 - 24.4.10 16×32 Multiply and Accumulate with 32-bit Result..... 643
 - 24.4.11 32×32 Multiply and Accumulate with 32-bit Result..... 643
 - 24.4.12 Unsigned Long Multiply with Accumulate Accumulate 644
- 24.5 Packing Halfwords into a Word 646**
- 24.6 Signed and Unsigned Extension 647**
- 24.7 GE Flags 648**
- 24.8 Byte Selection Instruction 649**
- 24.9 Basic DSP Functions 650**
 - 24.9.1 Vector Negate 650
 - 24.9.2 Vector Absolution Value 651
 - 24.9.3 Vector Offset with Saturation 653

24.9.4	Vector Shift Left with Saturation	655
24.9.5	Vector Mean	656
24.9.6	Vector Pairwise Multiplication	658
24.9.7	Vector Dot Product	660
24.9.8	Vector Min and Max	661
24.10	Exercises	662
Appendix A: GNU Compiler		665
A-1.	Introduction	665
A-2.	GNU and ARM Assembly Syntax Comparison	667
A-3.	Mixing C and Assembly	671
A-4.	Linker Script	674
A-5.	Programming and Debugging the board	676
Appendix B: Cortex-M3/M4 Instructions		678
Appendix C: Floating-point Instructions (Optional on Cortex-M4 and Cortex-M7) ...		680
Appendix D: DSP Instructions on Cortex-M4 and Cortex-M7		682
Appendix E: Cortex-M0/M0+/M1 Instructions.....		685
Appendix F: Cortex-M3 16-bit Thumb-2 Instruction Encoding.....		687
Appendix G: Cortex-M3 32-bit Thumb-2 Instruction Encoding		689
Appendix H: HID Codes of a Keyboard		696
Appendix I: GPIO Alternate Functions (STM32L4)		698
Bibliography	705	
Index	711	