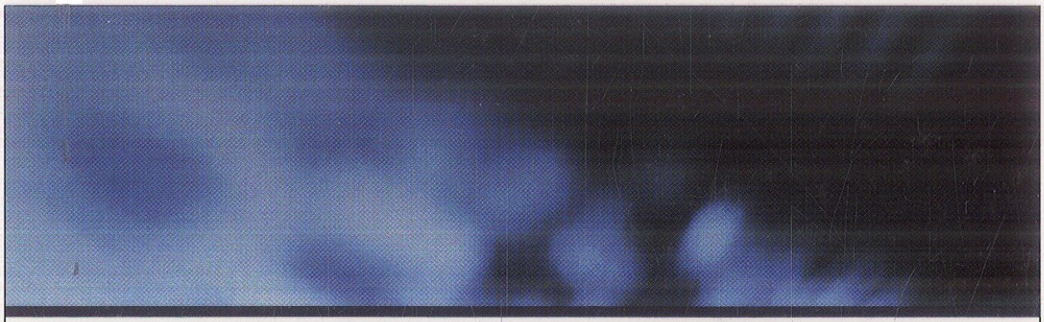
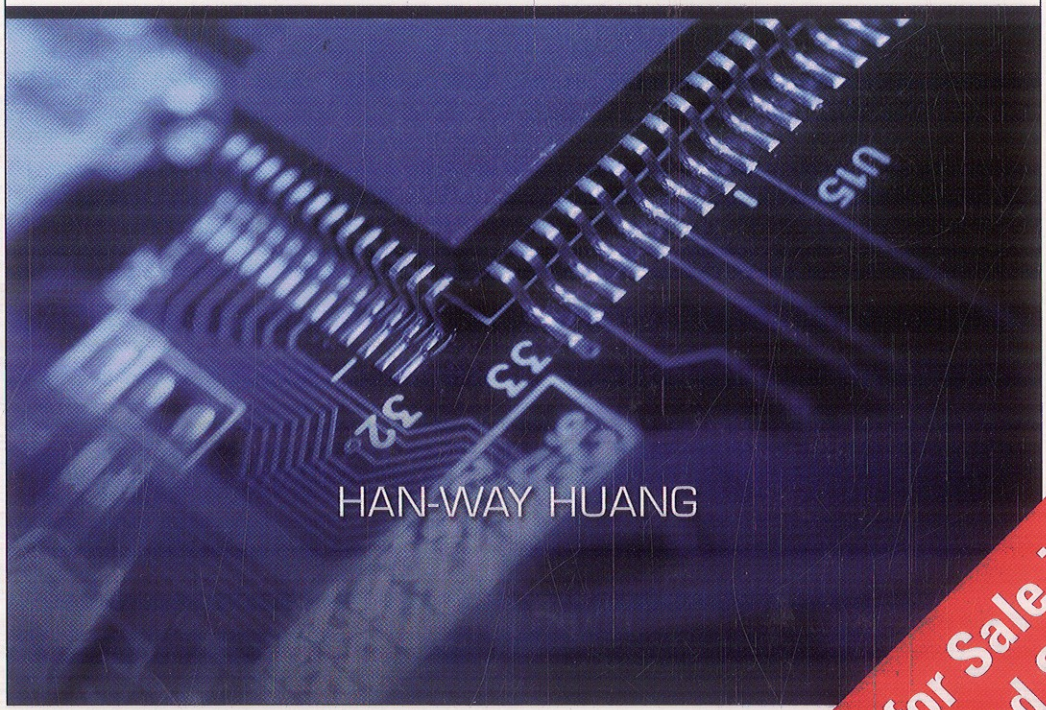
INTERNATIONAL STUDENT EDITION



EMBEDDED SYSTEM DESIGN

WITH THE C8051



HAN-WAY HUANG

Chapter 1 Introduction to Microcontroller and the Intel 8051 1

1.1 Objectives 1
1.2 Number System Issue 2

1.2.1 Converting from Binary to Decimal 2
1.2.2 Converting from Decimal to Binary 2
1.2.3 Why Octal and Hexadecimal Numbers? 5
1.2.4 Convert from Binary to Octal 5
1.2.5 Convert from Octal to Binary 5
1.2.6 Convert from Binary to Hexadecimal 6
1.2.7 Convert from Hexadecimal to Binary 6
1.2.8 Specifying the Number Base 7

1.3 Binary Addition and Subtraction 7
1.4 Two's Complement Numbers 8
1.5 Two's Complement Subtraction 9
1.6 Overflow 10
1.7 Representing Nonnumeric Data 11
1.8 Computer Hardware Organization 12

1.8.1 The Processor 14
1.8.2 Microprocessor 17
1.8.3 Microcontroller 17
1.8.4 Embedded Systems 18

1.9 Memory 19

1.9.1 Magnetic Memory 19
1.9.2 Optical Memory 20
1.9.3 Semiconductor Memory 20
1.9.4 Non-Volatile and Volatile Memory 20

1.9.5 Random Access Memory 20
1.9.6 Read-Only Memory 21

1.10 Memory System Operation 23

1.10.1 Read Operation 23
1.10.2 Write Operation 24

1.11 Program Execution 24

1.11.1 The Circuit of the Program Counter 25
1.11.2 Where does the Processor Start to Execute Program? 26
1.11.3 Instruction Execution Process 27
1.11.4 Instruction Sequence Example 28
1.11.5 Instruction Execution Process 28

1.12 Overview of the 8051 Microcontroller 35
1.13 The 8051 Memory Space 36
1.14 The 8051 Registers 37
1.15 Methods for Addressing Instruction Operands 38

1.15.1 Implied Register Mode 39
1.15.2 Direct Register Mode 40
1.15.3 Immediate Value Mode 40
1.15.4 Indirect Addressing Mode 41
1.15.5 Indexed Addressing Mode 41
1.15.6 Relative Addressing Mode 42
1.15.7 Absolute Addressing Mode 42
1.15.8 Long Address Mode 43
1.15.9 Implied Bit Mode 43
1.15.10 Direct Bit Mode 43

1.16 Examples of 8051 Instructions 46

1.16.1 Data Movement Instructions 46
1.16.2 The ADD Instructions 49
1.16.3 The SUB Instructions 51

1.17 Machine Instruction Timing 53
1.18 Chapter Summary 54
1.19 Exercise Problems 55

Chapter 2 Introductory C8051 Assembly Programming 58

2.1 Objectives 58
2.2 What is an Assembly Language Program? 58

2.2.1 The Label Field 59
2.2.2 The Operation or Directive Field 59
2.2.3 The Operand or Argument Field 60
2.2.4 The Comment Field 60

2.3 The 8051 Memory Classes 61
2.4 The 8051 Assembler Directives 61

2.4.1 Segments 62
2.4.2 Segment Location Counter 62

2.4.3 Directives for Controlling Segments 62
2.4.4 Directives for Defining Symbols 64
2.4.5 Directives for Initializing Memory Locations 67
2.4.6 Directives for Reserving Memory Locations 68
2.4.7 Directives for Controlling Addresses 69
2.4.8 Other Directives 69
2.4.9 Directives for Macro Definitions 70

2.5 Software Development Issue 73
2.6 Assembly Program Template 76
2.7 Writing Programs to Perform Computation 76

2.7.1 The Carry/Borrow Flag 78
2.7.2 Multi-Byte Addition 79
2.7.3 The CY Flag and Subtraction 79
2.7.4 Multi-Byte Subtraction 80
2.7.5 Multiplication and Division 80

2.8 Writing Program Loops 82

2.8.1 The Infinite Loop 82
2.8.2 The For Loop 83
2.8.3 The While Loop 85
2.8.4 The Repeat Until Loop 88

2.9 Jump Table 89
2.10 Looping-Support Instructions 90
2.11 Logical Operations 93
2.12 Rotate Instructions 95
2.13 Boolean Variable Manipulation Instructions 98
2.14 Hardware and Software Development Tools 99

2.14.1 Debug Adapter 100
2.14.2 Demo Boards 100
2.14.3 Software Tools 101

2.15 Using the SiLabs IDE 102

2.15.1 C8051 Program Memory Map 103
2.15.2 Getting Started with SiLabs IDE 106
2.15.3 Tool Chain Integration 106
2.15.4 Creating a New Project 108
2.15.5 Entering Source Programs 108
2.15.6 Adding Files to the Project 109
2.15.7 Building the Project 110
2.15.8 Downloading the Program for Debugging 110
2.15.9 Target Build Configuration 111
2.15.10 Program Execution and Debugging 112

2.16 Tips for Assembly Program Debugging 117

2.16.1 Syntax Errors 117
2.16.2 Logical Errors 118

2.17 Using Keil's μVision 122
2.18 Using the Raisonance's RIDE 122

2.19 Chapter Summary 123
2.20 Exercise Problems 124
2.21 Laboratory Exercise Problems and Assignments 126

Chapter 3 Assembly Programming and Simple I/O Operations 129

3.1 Objectives 129
3.2 Introduction to I/O Ports 129
3.3 Original 8051 I/O Port Structures and Operations 131
3.4 C8051F040 I/O Ports 133
3.4.1 Pin Assignment and Allocation Using the Crossbar Decoder 135
3.4.2 Configuring the Output Modes of the Port Pins 139
3.4.3 Port Data Registers 140
3.4.4 Configuring Port Pins as Digital Inputs 140
3.4.5 Weak Pull-ups 140
3.4.6 Configuring Port 1, 2, and 3 Pins as Analog Inputs 141
3.4.7 Port 4 through Port 7 141
3.5 C8051F040 Special-Function Registers Paging 142
3.5.1 Interrupts and SFR Paging 142
3.6 Simple Output Devices 143
3.6.1 Interfacing with LEDs 143
3.6.2 Interfacing with Seven-Segment Displays 144
3.7 Using Timer 0 to Create Time Delays 153
3.7.1 Mode 1 of Timer 0 153
3.7.2 Create Time Delay Using Mode 1 156
3.8 Simple Input Device 157
3.9 Using the D/A Converter 158
3.9.1 The AD7302 DAC 159
3.9.2 Interfacing the AD7302 with the 8051 Variants 160
3.10 Clock Generation and Control 163
3.10.1 Internal Oscillator 163
3.10.2 Using an External Oscillator as the System Clock 164
3.11 Chapter Summary 166
3.12 Exercise Problems 167
3.13 Laboratory Exercise Problems and Assignments 169

Chapter 4 Advanced Assembly Programming and Subroutines 171

4.1 Objectives 171
4.2 Introduction 171

4.3 The Stack Data Structure 172

4.3.1 Initializing the Stack Pointer 173
4.3.2 Instructions for Stack Operation 173

4.4 The XRAM of the C8051F040 174

4.4.1 Two Miscellaneous Operators 175

4.5 An Example of Subroutine 176
4.6 An Example of Top-Down Design with Hierarchical Refinement 177
4.7 Issues Related to Subroutine Calls 181

4.7.1 Parameter Passing 181
4.7.2 Result Returning 181
4.7.3 Local Variable Allocation and Deallocation 181
4.7.4 Accessing Incoming Parameters and Local Variables in the Stack 182

4.8 Writing Subroutines to Perform Multiprecision Arithmetic 183

4.8.1 Writing Subroutines to Perform 16-bit Unsigned Multiplication 183
4.8.2 Writing Subroutines to Perform 16-bit Signed Multiplication 186
4.8.3 Writing Subroutines to Perform Unsigned Multiprecision Division 188
4.8.4 Shift-and-Subtract Phase 189
4.8.5 Converting an Internal Binary Number into a BCD String 190
4.8.6 Signed Division Operation 193

4.9 Using the UART0 Module to Perform I/O 194
4.10 A Few More Tips on Assembly Program Debugging 200

4.10.1 What to Do When the Program Gets Stuck? 200
4.10.2 General Debugging Strategy 201

4.11 Chapter Summary 201
4.12 Exercise Problems 203
4.13 Laboratory Exercise Problems and Assignments 205

Chapter 5 C Language Programming 207

5.1 Objectives 207
5.2 Introduction to C 207
5.3 Types, Operators, and Expressions 209

5.3.1 Data Types 209
5.3.2 Variable Declarations 209
5.3.3 Constants 210
5.3.4 Arithmetic Operators 210
5.3.5 Bitwise Operators 211
5.3.6 Relational and Logical Operators 213
5.3.7 Precedence of Operators 213

5.4 Control Flow 213

5.4.1 If Statement 213
5.4.2 If-Else Statement 215
5.4.3 Multiway Conditional Statement 215
5.4.4 Switch Statement 216
5.4.5 For-Loop Statement 217
5.4.6 While Statement 217
5.4.7 Do-While Statement 217
5.4.8 GOTO Statement 218

5.5 Input and Output 218
5.6 Functions and Program Structure 220

5.6.1 Function Prototype 223
5.6.2 Writing a C Program with Multiple Functions 223

5.7 Pointers, Arrays, Structures, and Unions 225

5.7.1 Pointers and Addresses 225
5.7.2 Arrays 226
5.7.3 Pointers and Arrays 229
5.7.4 Passing Arrays to a Function 229
5.7.5 Initializing Arrays 230
5.7.6 Structures 233
5.7.7 Unions 234

5.8 Miscellaneous Items 235

5.8.1 Automatic/External/Static/Volatile 235
5.8.2 Scope Rules 236
5.8.3 Type Casting 237

5.9 Using the C Compiler 238
5.10 C Language Extensions for the 8051 MCU 238

5.10.1 Memory Model 238
5.10.2 Storage Class Language Extensions 239
5.10.3 Pointers 241
5.10.4 Generic Pointers 242
5.10.5 Memory-Specific Pointers 242
5.10.6 Absolute Variable Location 243
5.10.7 Parameters and Local Variables 243
5.10.8 Passing Parameters in Registers 243
5.10.9 Function Return Values 244
5.10.10 Function Declaration 244
5.10.11 Specifying Register Bank for a Function 245
5.10.12 In-Line Assembly Instructions 245
5.10.13 Header Files 246

5.11 C Library Functions 247
5.12 Using SiLabs IDE to Develop C Programs 248
5.13 Building Multiple File Projects 251
5.14 Using UART C Functions 253

5.15 Tips for C Program Debugging 255
5.16 Chapter Summary 257
5.17 Exercise Problems 259
5.18 Laboratory Exercise Problems and Assignments 260

Chapter 6 Interrupts and Resets 263

6.1 Objectives 263
6.2 Fundamental Concepts of Interrupts 263
6.2.1 What is an Interrupt? 263
6.2.2 Why are Interrupts Used? 265
6.2.3 Enabling and Disabling Interrupts 265
6.2.4 Handling Multiple Interrupts by Prioritizing 266
6.2.5 Servicing the Interrupt 266
6.2.6 What is an Interrupt Vector? 267
6.2.7 How to Write an Interrupt-Driven Program? 267
6.3 Resets 268
6.4 The 8051 Interrupts 269
6.4.1 The 8051 Interrupt Priority Structure 271
6.4.2 INT0 and INT1 Pins Interrupts 273
6.5 The 8051 Reset 282
6.5.1 The RST Pin Reset 284
6.5.2 Power-On Reset 284
6.5.3 Power-FailureReset 285
6.5.4 JTAG Debug Reset 285
6.5.5 Missing Clock Detector (MCD) Reset 285
6.5.6 Comparator 0 Reset 286
6.5.7 External CNVSTR0 Pin Reset 286
6.5.8 Watchdog Timer Reset 286
6.6 Power-Saving Modes of Operation 288
6.6.1 The IDLE Mode 289
6.6.2 The STOP Mode 290
6.7 Chapter Summary 291
6.8 Exercise Problems 292
6.9 Laboratory Exercise Problems and Assignments 293

Chapter 7 Advanced Parallel I/O 295

7.1 Objectives 295
7.2 Electrical Characteristic Considerations for I/O Interfacing 295
7.2.1 Voltage-Level Compatibility 296
7.2.2 Current Drive Capability 297
7.2.3 Timing Compatibility 299

7.3 Liquid Crystal Displays (LCDs) 300
7.4 The HD44780U LCD Controller 301
7.4.1 Display Data RAM 304
7.4.2 Character Generator ROM (CGROM) 304
7.4.3 Character Generator RAM (CGRAM) 304
7.4.4 Registers 304
7.4.5 Instruction Description 306
7.4.6 Interfacing the HD44780U to the 8051 Microcontroller 308
7.5 Using Keypad as an Input Device 320
7.5.1 How to Perform Keypad Scanning 321
7.5.2 How to Perform Keypad Debouncing 322
7.5.3 ASCII Code Lookup 323
7.6 Driving Stepper Motor 326
7.6.1 Stepper Motor Drivers 327
7.7 Chapter Summary 331
7.8 Exercise Problems 332
7.9 Laboratory Exercise Problems and Assignments 335

Chapter 8 Timers and Programmer Counter Array 338

8.1 Objectives 338
8.2 Introduction 338
8.3 Timer 0 and Timer 1 340
8.3.1 Mode 0: 13-bit Counter/Timer 341
8.3.2 Mode 1: 16-bit Counter/Timer 342
8.3.3 Mode 2: 8-bit Counter/Timer with Auto-Reload 343
8.3.4 Mode 3: Two 8-bit Counter/Timers (Timer 0 only) 343
8.3.5 Using Timer 0 and Timer 1 in Measuring Signal Parameters 344
8.4 Timer 2, Timer 3, and Timer 4 349
8.4.1 Configuring Timers 2, 3, and 4 to Count Down 351
8.4.2 Timers 2, 3, and 4 Capture Mode 351
8.4.3 Timers 2, 3, and 4 Auto-Reload Mode (Up or Down Counter) 353
8.4.4 Toggle (Programmable) Clock-Out Mode 353
8.4.5 Timer 2 Baud Rate Generation Mode 355
8.4.6 Applications of Timer 2 355
8.5 Programmable Counter Array 372
8.5.1 PCA Timer/Counter 372
8.5.2 Capture/Compare Modules 373
8.5.3 Edge-triggered Capture Mode 377
8.5.4 16-bit Software Timer and High-Speed Output (Toggle) 381

8.5.5 C8051F040 PCA Frequency-Output Mode 388
8.5.6 8-bit Pulse-Width Modulation Mode (Fixed Frequency) 390
8.5.7 16-bit Pulse-Width Modulator Mode (Fixed Frequency) 392

8.6 DC Motor Control 394

8.6.1 DC Motor Driver ICs 395
8.6.2 Driving a DC Motor Using the SN754410 395

8.7 Chapter Summary 397
8.8 Exercise Problems 398
8.9 Laboratory Exercise Problems and Assignments 399

Chapter 9 Universal Asynchronous Receiver Transceiver (UART) 406

9.1 Objectives 406
9.2 Concepts of Serial I/O 406
9.3 The RS-232 Standard 407

9.3.1 EIA-232 Electrical Specification 408
9.3.2 EIA-232 Functional Specification 408
9.3.3 EIA-232 Mechanical Specification 412
9.3.4 EIA-232 Procedural Specification 412
9.3.5 Data Format 414
9.3.6 UART Detectable Errors 416
9.3.7 Null Modem Connection 416

9.4 The 8051 UART Module 417

9.4.1 UART Pins 417
9.4.2 UART Operation Mode 418
9.4.3 Registers Associated with the UART 419
9.4.4 Baud Rate Generation for the Original 8051 419
9.4.5 Baud Rate Generation for the C8051F040 UART Modules 423

9.5 The Operation of UART Module 425

9.5.1 UART Mode 0 425
9.5.2 UART Mode 1 425
9.5.3 UART Mode 2 426
9.5.4 UART Mode 3 426

9.6 Applications of the UART Mode 0 426
9.7 Applications of UART Mode 1 432

9.7.1 Direct Connection of UART in Mode 1 433
9.7.2 Data Exchange with UART Mode 1 via EIA232 Interface 438

9.8 Applications of UART Modes 2 and 3 444

9.8.1 Multiprocessor Application of UART of the Original 8051 445
9.8.2 Multiprocessor Application of UART of C8051F040 445
9.9 Chapter Summary 447
9.10 Exercise Problems 448
9.11 Laboratory Exercise Problems and Assignments 449

10.1 Objectives 451
10.2 Introduction to the SPI Function 451
10.3 SPI Signal Pins 452
10.4 Registers Related to SPI 453
10.5 SPI Operation 455

10.5.1 Transmission Formats 455
10.5.2 SPI Master Mode Operation 456
10.5.3 SPI Slave Mode Operation 457
10.5.4 SPI Baud Rate 457

10.6 SPI Interrupt Sources of the C8051F040 457
10.7 SPI Circuit Connection 460
10.8 SPI-Compatible Chips 462
10.9 The 74LV595 Shift Register 462
10.10 The LTC1661 D/A Converter 467

10.10.1 Signal Pins 468
10.10.2 Data Format 468
10.10.3 DAC Control Functions 469
10.10.4 LTC1661 Output Voltage 469
10.10.5 Interfacing the LTC1661 with the C8051F040 469

10.11 EEPROM with SPI Interface 475

10.11.1 The 25AA080A EEPROM 476
10.11.2 Read Operation 477
10.11.3 Write Operation 477
10.11.4 Write Status Register 477
10.11.5 Data Protection 478
10.11.6 Interfacing the 25AA080A with the C8051F040 478

10.12 Matrix LED Displays 487

10.12.1 The Organization of Matrix LED Displays 487
10.12.2 Colors of Matrix LED Displays 488
10.12.3 Connection Method 488
10.12.4 Dimension of Matrix LED Displays 489
10.12.5 Method of Driving Matrix LED Displays 489

10.13 The MAX6952 Matrix LED Display Driver 490

10.13.1 Pin Functions 490
10.13.2 Internal Registers 492
10.13.3 Blinking Operation 498
10.13.4 Choosing Values for R_{SET} and C_{SET} 500

10.14 Chapter Summary 507
10.15 Exercise Problems 508
10.16 Laboratory Exercise Problems and Assignments 512

Chapter 11 I²C Bus and SMBus 513

11.1 Objectives 513
11.2 Overview of Protocols 513

11.2.1 Overview of I²C 513
11.2.2 Overview of SMBus 514
11.2.3 Differences between I²C and SMBus Protocol 515

11.3 I²C and SMB Protocols 516

11.3.1 Communication Procedure of I²C and SMBus 516
11.3.2 Signal Levels 517
11.3.3 I²C and SMBus Data-Transfer Signal Components 518
11.3.4 Bus Arbitration 520
11.3.5 Synchronization 520
11.3.6 Data Transfer Format 522

11.4 The SMBus of the C8051F040 524

11.4.1 The SMB0CN Register 524
11.4.2 The SMB0CR Register 526
11.4.3 SMBus Data Register (SMB0DAT) 527
11.4.4 Address Register 527
11.4.5 Status Register 527

11.5 Using the C8051F040 SMBus 529

11.5.1 Implementation Choices 529
11.5.2 Interfacing the Serial EEPROM 24LC08 with SMBus 530

11.6 Using the Digital Thermostat DS1631A 545

11.6.1 Pin Assignment 545
11.6.2 Functional Description 546
11.6.3 DS1631A Registers 546
11.6.4 The DS1631A Operation 547
11.6.5 DS1631A Command Set 549
11.6.6 Interfacing the DS1631A with the C8051F040 550

11.7 Interfacing with I²C Serial Real-Time Clock DS1337 558

11.7.1 Signal Functions 559
11.7.2 Address Map 560
11.7.3 Clock and Calendar 560
11.7.4 Special Registers 560
11.7.5 Alarms 561
11.7.6 Interfacing the DS1337 with C8051F040 563

11.8 Chapter Summary 572
11.9 Exercise Problems 573
11.10 Laboratory Exercise Problems and Assignments 575

Chapter 12 Analog-to-Digital and Digital-to-Analog Converters 576

12.1 Objectives 576
12.2 Basics of A/D Conversion 576

12.2.1 A Data Acquisition System 577
12.2.2 Analog Voltage and Digital Code Characteristics 577
12.2.3 A/D Conversion Algorithms 579
12.2.4 Interpreting A/D Conversion Result 580
12.2.5 Voltage Amplifying Circuit 581
12.2.6 Voltage Translation Circuit 582

12.3 The C8051F040 A/D Converters 583
12.4 The 12-bit ADC0 A/D Converter 583

12.4.1 Analog Multiplexer and Programmable Gain Amplifier 583
12.4.2 Dedicated External Analog Inputs 584
12.4.3 High-Voltage Difference Amplifier (HVDA) 585
12.4.4 Additional Analog Input from Port 3 586
12.4.5 Analog Multiplexer Channel Selection 587
12.4.6 Programmable Gain Amplifier (PGA) 589
12.4.7 A/D Conversion Clock 589
12.4.8 A/D Conversion Start Methods 589
12.4.9 Analog Input Signal Tracking (Sampling) 590
12.4.10 A/D and D/A Reference Voltages 591
12.4.11 Interpreting the Conversion Result 593
12.4.12 ADC0 Programmable Window Detector 594
12.4.13 On-Chip Temperature Sensor 594

12.5 The C8051F040 ADC2 A/D Converter 601

12.5.1 Analog Multiplexer and PGA 601
12.5.2 ADC2 Operation 602
12.5.3 Tracking Modes 606
12.5.4 ADC2 Programmable Window Detector 606

12.6 Using the Temperature Sensor TC1047A 608
12.7 Measuring Barometric Pressure 611
12.8 Measuring Humidity 617
12.9 Digital-to-Analog Converter of the C8051F040 620

12.9.1 Signal Pins Related to DACs 620
12.9.2 Registers Related to DACs 621
12.9.3 DAC Operation 622

12.10 Chapter Summary 630
12.11 Exercise Problems 632
12.12 Laboratory Exercise Problems and Assignments 633

Chapter 13 Controller Area Network (CAN) 635

13.1 Objectives 635
13.2 Overview of Controller Area Network 635
13.2.1 Layered Approach in CAN 636
13.2.2 General Characteristics of CAN 636
13.3 CAN Messages 638
13.3.1 Data Frame 638
13.3.2 Remote Frame 642
13.3.3 Error Frame 642
13.3.4 Overload Frame 643
13.3.5 Interframe Space 644
13.3.6 Message Filtering 645
13.3.7 Message Validation 645
13.3.8 Bit Stream Encoding 646
13.4 Error Handling 646
13.4.1 Bit Error 646
13.4.2 Stuff Error 646
13.4.3 CRC Error 646
13.4.4 Form Error 646
13.4.5 Acknowledgment Error 647
13.4.6 Error Signaling 647
13.5 Fault Confinement 647
13.5.1 CAN Node Status 647
13.5.2 Error Counts 647
13.6 CAN Message Bit Timing 647
13.6.1 Nominal Bit Time 648
13.6.2 Length of Time Segments 649
13.7 Synchronization Issue 649
13.7.1 Resynchronization Jump Width 649
13.7.2 Phase Error of an Edge 650
13.8 The C8051F040 CAN Module 650
13.8.1 The Structure of the CAN Controller 651
13.8.2 Registers Related to CAN Module 652
13.8.3 CAN0ADR Auto-incrementing Features 654
13.8.4 The CAN Core Registers 655
13.8.5 Message Interface Register Sets IF1 and IF2 659
13.8.6 Message Handling Registers 664
13.8.7 Message Objects in the Message Memory 667
13.9 CAN Operation Modes 668
13.9.1 Normal Mode 668
13.9.2 Basic Mode 668

13.9.3 Test Mode 669
13.9.4 Silent Mode 669
13.9.5 LoopBack Mode 669
13.9.6 LoopBack Combined with Silent Mode 670

13.10 CAN Module Operation 671

13.10.1 Managing Message Objects 671
13.10.2 Data Transfer to/from Message RAM 674
13.10.3 Transmission of Messages 674
13.10.4 Sending Remote Frame 674
13.10.5 Auto Reply to Remote Frames 675
13.10.6 Acceptance Filtering of Received Messages 675
13.10.7 Reception of Data Frame 676
13.10.8 Reception of Remote Frame 676
13.10.9 Receive/Transmit Priority 676

13.11 Configuration of Message Objects 677

13.11.1 Configuration of a Message Object to be Transmitted 678
13.11.2 Updating a Transmit Object 679
13.11.3 Configuration of a Receive Object 679

13.12 Handling of Received Messages 682
13.13 Configuration of a FIFO Buffer 682
13.14 Handling of CAN Interrupts 685
13.15 Setting the CAN Bit Timing Parameters 686
13.16 Physical CAN Bus Connection 690

13.16.1 The SN65HVD230 CAN Bus Transceiver 690
13.16.2 Interfacing the SN65HVD230 to the C8051F040 692

13.17 Chapter Summary 700
13.18 Exercise Problems 702
13.19 Laboratory Exercise Problems and Assignments 703

Chapter 14 Memory Subsystem 706

14.1 Objectives 706
14.2 Introduction 706
14.3 Basic Concepts of Buses 707
14.4 Representing Bus Signals 708
14.5 Bus Transactions 710
14.6 Bus Multiplexing 711
14.7 The C8051F040 Flash Program Memory 712

14.7.1 Programming the Flash Memory 712
14.7.2 Non-Volatile Data Storage 716
14.7.3 Security Options 717

14.8 External Data Memory Interface and On-Chip XRAM 718
14.9 Configuring the External Memory Interface 719

14.9.1 Port Selection and Configuration 719

14.9.2 Multiplexed and Non-Multiplexed Selection 720
14.9.3 Memory Mode Selection 721
14.9.4 Split Mode with Bank Select 722
14.9.5 Timing 723

14.10 Issues Related to Adding External Memory 726

14.10.1 Memory Space Assignment 726
14.10.2 Address Decoder Design 727
14.10.3 Timing Verification 727

14.11 Memory Devices 728

14.11.1 The CY7C1399BN SRAM 728
14.11.2 AT28BV256 EEPROM 730

14.12 Example of External Memory Expansion for the C8051F040 735

14.12.1 Memory Space Assignment 736
14.12.2 Address Latch 736
14.12.3 Address Decoder Design 736
14.12.4 Timing Verification 737

14.13 Chapter Summary 738
14.14 Exercise Problems 739
14.15 Laboratory Exercise Problems and Assignments 740

Appendix A The 8051 Instruction Execution Times (Courtesy of Intel and Silabs) 741

Appendix B Tutorial for Using the Keil's μVision IDE 747

Appendix C Tutorial for Using the Raisonance's Ride IDE 767

Appendix D C8051F040 Special Function Registers 783

Appendix E C8051F040 SFR PAGE Definition (Keil and Raisonance) 788

Appendix F Procedure for Setting up HyperTerminal 790

Appendix G Keil C Library Functions 795