

Table of Contents

	Preface	<i>xix</i>
	Acknowledgement	<i>xxi</i>
1	Introduction	1
1.1	Wireless Connectivity Alternatives	1
1.2	Goals	2
1.3	The Fundamental Problem	3
1.4	Audience	4
1.5	Recommended Reading	5
1.6	Can One Size Fit All?	6
1.7	Hardware History	6
1.8	On-the-move Connectivity Problems	7
1.9	Reference Implementations	8
1.10	Reference Microcontroller/OS Platform	9
1.11	Reference Cellular Communication Modules (CCMs) Family	11
1.12	A Few Words on Advice, Practices, and Efficiency	13
1.12.1	<i>Best Practice</i> or <i>Good Practice</i>	14
1.12.2	Efficiency Is a Large Umbrella	14
1.12.2.1	Spatial and Temporal Efficiency	14
1.12.2.2	Data Efficiency	15
1.12.2.3	Developmental Efficiency	17
1.13	3G, 4G, 5G, and 6G	17
2	Platforms, Tools, and Debugging	19
2.1	Importance of Toolchain Selection	19
2.2	An Expanded View of the Tool Chain	21
2.3	Computing/OS Platforms	21
2.4	Programming Language Choices	23

2.5	Running the Same Code on Development Computer and IoT Device	23
2.6	Running IoT Connectivity Code in a Debugger	24
3	Cellular Network Basics	27
3.1	Standards	27
3.2	What Do Cellular Networks Do?	28
3.3	Are Cellular Networks Wireless?	29
3.4	What Is a Cell? What Is a Sector?	30
3.5	Omnidirectional Cellular Coverage	33
3.6	Cell Towers	34
3.7	How Are Cellular Networks Identified?	37
3.8	How Are IoT Devices Identified?	39
3.9	eNodeB IDs and Cell #	40
3.10	Tracking Areas and Paging	40
3.11	Frequency and Modulation	42
3.11.1	Modulation	43
3.11.1.1	Radio Telegraphy	43
3.11.1.2	Amplitude Modulation (AM Radio)	44
3.11.1.3	Frequency Modulation (FM Radio)	45
3.11.1.4	Phase Modulation	47
3.12	Spectral Efficiency	48
3.13	Error Detection	49
3.13.1	Luhn Algorithm	50
3.14	Error Correction	52
3.15	LTE Modulation Techniques	54
3.15.1	Binary Phase Shift Keying (BPSK)	56
3.15.2	Quadrature Phase Shift Keying (QPSK)	57
3.15.3	Quadrature Amplitude Modulation (QAM)	57
3.16	Bandwidth and Latency	57
3.17	Range	58
3.18	Frequency Bands	59
3.18.1	Frequency Affects Range	60
3.19	Radio Access Technologies (RAT) and Categories	61
3.20	SIM Cards	63
3.21	What Happens When a Cellular Modem Switches On?	64
3.21.1	Network Selection, Cell Selection, Camping, and Cell Reselection	64
3.21.2	Network Registration	65
3.22	Handoff (Also Called Handover)	67
3.22.1	Maintaining Connectivity	67

3.22.2	Load Balancing	69
3.23	Multiple Access	70
3.24	Timing Advance	70
3.24.1	Why Is Timing Advance Useful?	73
3.24.2	How Accurate Are Distance Estimates Using Timing Advance?	73
3.24.3	Timing Advance Band Depth and Maximum Range	77
3.25	Expressing Power	77
4	SIM/USIM Card Basics	81
4.1	Mobile Virtual Network Operators (MVNOs)	81
4.2	Size	82
4.3	Native Versus Multi-SIMs or MNO Versus MVNO	84
4.4	Home Versus Roaming Access	85
4.5	SIM Factors Affecting Price and Coverage	86
4.5.1	How Much Do SIM Cards Cost?	88
4.5.2	Is There a Monthly Activation Fee?	88
4.5.3	Are There Fees for Activating and/or Deactivating SIM Cards?	88
4.5.4	How Much Does Data Cost?	88
4.5.5	Is the Monthly Data “Pooled”?	89
4.5.6	Are There Fees for Deactivated (but Not Terminated) SIMs Cards?	90
4.5.7	Is There a Not-yet-activated Fee?	90
4.6	Text Messages (SMS)	91
4.7	Usage Limits	91
4.8	Firewalls	92
4.9	Replacing SIMs and/or Network Providers	94
4.10	Access Point Name (APN)	96
5	Verify Cellular Connectivity	97
5.1	Preparation	98
5.1.1	Adequate Power	98
5.1.2	Activated SIM Card	100
5.1.3	Base Station in Range	100
5.1.4	SIM Card Authorization	100
5.1.5	Band Configuration	100
5.1.6	RAT Configuration	101
5.1.7	Automatic Registration	101
5.2	Try to Auto-register	101
5.3	What Can Go Wrong?	102
5.3.1	Operating System Interference	102
5.3.2	Communicating with Modem	103

- 5.3.3 Malformed AT Commands 103
- 5.3.4 Parsing Responses to AT Commands 103
- 5.3.5 Timing Problems 104
- 5.3.6 Unset or Incorrect Access Point Name (APN) 107
 - 5.3.6.1 Pitfall: Failing to Explicitly Set the APN 107
- 5.4 Modem Configuration for Auto-registration 108

- 6 Let's Move Some Data 111**
 - 6.1 Low-level Sockets or High-level Protocols 112
 - 6.2 Verify *ServerServer* Is Running 116
 - 6.3 Verify *EchoServer* Is Running 117
 - 6.4 USB or UART? 117
 - 6.4.1 Pitfall: *USB Device Names Are Not Fixed and Should Not Be Hard Coded* 118
 - 6.5 AT Commands—A Troubled Past 119
 - 6.6 Unsolicited Response Codes (URCs) 120
 - 6.7 A Handy *Modem* Program 123
 - 6.8 AT Commands Manuals 130
 - 6.9 Communicating with the Cellular Modem 131
 - 6.10 Getting *EchoServer* Information from *ServerServer* 134
 - 6.10.1 Before Step 1: Enable Unsolicited Response Codes (URCs) 135
 - 6.10.2 Step 1: Configure PDP Context for HTTP GET 136
 - 6.10.3 Step 2: Configure a PDP Context 136
 - 6.10.3.1 Test Commands 137
 - 6.10.3.2 Read Commands 137
 - 6.10.3.3 Write Commands 137
 - 6.10.4 Step 3: Activate the PDP Context 139
 - 6.10.5 Step 4: Allow/Disallow Response Header for HTTP GET 140
 - 6.10.6 Step 5: Set HTTP(S) URL 141
 - 6.10.7 Step 6: Send HTTP(S) GET Request 141
 - 6.10.8 Step 7: Read HTTP(S) Response 142
 - 6.10.9 After Step 7: Deactivate the PDP Context 143
 - 6.10.10 Using *Modem* to Interact with *ServerServer* 143
 - 6.11 Bouncing Data Off *EchoServer* 145
 - 6.11.1 Before Step 1: Enable Unsolicited Response Codes (URCs) 145
 - 6.11.2 Step 4: Open a TCP Client Socket Connection 146
 - 6.11.3 Step 5: Send Mixed-case Text to *EchoServer* 146
 - 6.11.4 Step 6: Read (Uppercase) Response from *EchoServer* 148
 - 6.11.5 Step 7: Close TCP Client Socket Connection 151
 - 6.11.6 Step 8: Deactivate PDP Context 151
 - 6.12 No Problems Is Bad Luck 151

7	Cellular Connectivity Regions	153	
7.1	How Geography, Topology, and Population Density Affect Connectivity	154	
7.1.1	Geography and Topology	154	
7.1.2	Population Density	154	
7.2	Region Categories	156	
7.2.1	Rural	156	
7.2.2	Rural Town	156	
7.2.3	Flat Farmland/Flat Arid	156	
7.2.4	Mountainous	157	
7.2.5	Suburban	157	
7.2.6	Dense City	157	
7.2.7	Interstate Highway	157	
7.2.8	Uninhabited	158	
8	Cellular Communication Modules (CCMs)	159	
8.1	CCM Worldwide Market Share	162	
8.2	Frequency Band Usage	162	
8.3	Protocol Implementation	164	
8.4	Similarities and Differences Across CCMs	165	
8.4.1	Single or Dual AT Command Channels	165	
8.4.2	Different AT Command Sets	166	
8.4.3	Different Response Times for Similar or Identical Commands	167	
8.4.4	Differing Response Formats	167	
8.4.5	Differing Responses for Compound Statements	167	
8.4.6	Different Timing Requirements	168	
8.4.7	AT Commands Are Not Thread-safe	168	
8.4.8	Support for Different Protocols	168	
8.5	Consider the Whole CCM Family	169	
8.6	CCM Firmware Bugs	169	
8.7	CCMs Are a Lot Like Sensors: Imprecise and Not Entirely Reliable	170	
9	AT Commands (A New Approach)	171	
9.1	Purpose of AT Commands	171	
9.2	Problems of AT Commands	173	
9.2.1	Maximum Response Time for an AT Command	174	
9.3	Traditional Solution to Executing AT Commands and Extracting Responses	175	
9.4	Command Independent Processing (CIP)	179	

9.4.1	The Central Observation Underlying CIP	180
9.4.2	Fundamental Elements of CIP	181
9.4.2.1	AtParams	181
9.4.2.2	AtCommand	181
9.4.3	AT Commands in CIP	182
9.4.3.1	Step 1: Define a Name for a Command	183
9.4.3.2	Step 2: Create a Set of Parameters for Each Command	183
9.4.3.3	Step 3: Store the Command Name and AtParams Object in a Map	184
9.4.3.4	Step 4: Create a Command Object	185
9.4.3.5	Step 5: Pass Arguments to the Command Object (if Necessary)	185
9.4.3.6	Step 6: Perform the Command	185
9.4.3.7	Step 7: Verify Success or Failure	186
9.4.3.8	Step 8: Extract Response Information	187
9.4.3.9	AT Commands with Parameters	192
9.4.3.10	Timing Out	194
9.4.4	Using CIP Across CCM Families and Across Manufacturers	196
10	CIP Design and Details	197
10.1	Pseudocode Conventions	198
10.1.1	Identifier Names	198
10.1.2	Angle Brackets	198
10.1.3	Constructors	199
10.1.4	Dot Operator	199
10.1.5	Unified Modeling Language (UML)	199
10.2	A Note on Objected-orientation and Threads	199
10.3	AT Command Basics	200
10.3.1	Echoing	200
10.3.2	Enable/Disable Response Codes	201
10.3.3	Short or Long Response Codes	201
10.3.4	Line Terminators	201
10.3.5	Housekeeping	201
10.4	Categories of Responses to AT Commands	202
10.4.1	OK_ONLY	203
10.4.2	TEXT_OK	203
10.4.3	AFTER_COLON	203
10.4.4	OK_PLUS_AFTER_COLON	204
10.4.5	MULTI_RECEIVE, MULTI_SEND, and MULTI_AFTER_COLON	205

10.5	Details of Command Independent Processing (CIP)	205	12
10.5.1	AtStep Purpose	205	12
10.5.2	AtStep Attributes	207	12
10.6	A “Factory Method” for Creating AtCommand Objects	208	12
10.7	Performing AT Commands	210	12
10.7.1	Why AT Commands Fail	212	12
10.7.2	Timing Out	212	12
10.7.3	Details of the Execute Method	217	12
10.7.4	Response Length	219	12
10.7.5	Hardware Timing	220	12
10.7.6	Combining Parameter Settings—Method Chaining	221	12
10.7.7	Assessing Success and Multiple Tries	221	12
10.7.8	Multi-line AT Commands—AtStep	223	12
10.7.9	A Second Example with Regular Expressions	225	12
10.7.10	Integrating AtStep into the Execute Methods	227	12
10.8	AT Commands for Multiple Modems	228	12
10.8.1	The Simplest Case	230	12
10.8.2	Connectors	231	12
10.8.2.1	All Connectors Are Also Threads	234	12
10.8.2.2	Connectors Are Created Using a Factory Method	235	12
10.8.2.3	Custom AT Commands Are Added in Static Blocks of Connectors	236	12
10.8.2.4	Where to Override Methods or Parts of Methods	237	12
10.8.3	An Asymmetrical Case—AtParamsNoOp	239	12
11	Geographical Coverage, Signal Strength, and Quality	243	12
11.1	Radio Access Technologies (RATs)	243	12
11.2	Cellular Network Coverage Maps	245	12
11.3	Signal Strength and Quality: RSSI, RSRP, RSRQ, SINR	246	12
11.3.1	RSSI and RSRP	247	12
11.3.2	RSRQ	249	12
11.3.3	SINR	249	12
11.3.4	Using <i>Modem</i> to Report Signal Strength and Quality	250	12
11.4	Antenna Selection and Performance	251	12
11.4.1	Antenna Size	253	12
11.4.2	Passive Versus Active Antennas	255	12
11.4.3	Antenna Connectors	256	12
11.4.4	Antenna Placement	257	12
11.5	Antenna Testing	258	12
11.6	Geography and Signal Strength Must Be Considered Together	259	12

12	Network Selection and Registration	261
12.1	Network Registration	261
12.2	Radio Access Technology (RAT)	262
12.3	Network Frequency Band Selection	264
12.4	PLMN Selection	266
12.4.1	Manual PLMN Selection	267
12.4.2	Automatic PLMN Selection	267
12.5	How to Create Your Own User Preference List	268
12.5.1	Reading the UPL and OPL	269
12.5.2	Modifying the UPL	271
12.6	Once a PLMN Is Auto-selected, Is It Always Selected?	273
12.7	Forcing the CCM Back to the PLMN Preference List	274
12.8	A Mysterious PLMN Selection Behavior	275
12.9	Troubleshooting Registration Problems	276
12.9.1	New Modem, Never Registered	276
12.9.2	Old Modem, Previously Registered	278
12.10	Anomalous Behavior	278
13	Communication Protocols TCP, UDP, and PPP	281
13.1	Internet Protocol	281
13.2	Transmission Control Protocol (TCP)	283
13.3	Considering Data Consumption	285
13.4	User Datagram Protocol (UDP)	285
13.5	TCP Pros and Cons	287
13.6	Point-to-point Protocol (PPP)	290
13.7	AT Commands for Data Transfer Are Completely Unstandardized	292
13.8	PPP on Linux	292
13.8.1	Debugging PPP	295
13.9	Alternatives to PPP	298
14	Thin Air	301
14.1	A Most Dramatic Case	302
14.1.1	Watching the Server	304
14.1.2	Packets Not Getting to the Server	305
14.2	What Was Going On? Thin Air	306
14.3	Why Did Thin Air Persist Over Hundreds of Miles?	308
14.4	How to Detect Thin Air	312
14.5	What to Do About Thin Air	313
14.6	Minimizing the Size of a Thin Air Region	313

- 14.7 A Hybrid UDP Protocol for Detecting Thin Air 314
- 14.8 Reducing (or Eliminating) Thin Air by PLMN or Band Selection 315
- 14.8.1 The Most Direct Approach 316
- 14.9 Putting the Hybrid Protocol to Second Use 322
- 15 Time and Location (GNSS) 325**
 - 15.1 Clarifying Terminology 325
 - 15.2 Time 326
 - 15.3 Location 328
 - 15.4 Obtaining Time Information 331
 - 15.4.1 Real-time Clock 331
 - 15.4.2 Cellular Modem 332
 - 15.4.2.1 Additional Configuration 333
 - 15.4.2.2 Local Time or UTC 333
 - 15.4.2.3 Daylight Saving Time 335
 - 15.4.2.4 Using *Modem* to Read the Clock 336
 - 15.4.3 Get Time from a GNSS Receiver 336
 - 15.4.4 Get Time from a Server 337
 - 15.5 Sources of Location Information 337
 - 15.6 Pros and Cons of CCM's GNSS Receiver Versus Stand-alone GNSS Receiver 338
 - 15.7 Cold Start, Warm Start, Hot Start 339
 - 15.8 Assisted GPS 340
 - 15.9 GNSS Antenna Selection 340
 - 15.10 GNSS Receiver Placement 341
 - 15.11 GNSS Accuracy and Precision 342
 - 15.11.1 Improving Accuracy 345
 - 15.12 NMEA Sentences 346
 - 15.12.1 Using *Modem* to Read GNSS Sentences 347
 - 15.13 Three Ways to Obtain Location Information 348
 - 15.13.1 Simple AT Command Request for Location 348
 - 15.13.2 Read Streaming Data from *gpsd* 350
 - 15.13.2.1 For a Stand-alone GNSS Receiver 350
 - 15.13.2.2 For a CCM's GNSS Receiver 351
 - 15.13.3 Read Streaming Data Directly from CCM's GNSS receiver 355
 - 15.14 Understanding *gpsd* JSON Output 356
 - 15.15 Writing Software to Capture and Process *gpsd* Output 358
 - 15.16 GNSS Data Streamed from a CCM 359
 - 15.17 NMEA 0183 359
 - 15.17.1 Talker Sentence Format 360

15.17.1.1	GSA Sentence Format	361	
15.17.1.2	RMC Sentence Format	362	
15.17.1.3	GSV Sentence Format	362	
15.17.2	NMEA Checksums	364	
15.17.3	CCM GNSS Receivers Only Stream Some NMEA Sentences		365
15.18	Some Additional <i>gpsd</i> Utilities	366	
16	Establishing and Maintaining a Cellular Connection		369
16.1	Modem Selection	370	
16.2	Foundational Tasks	371	
16.2.1	State 1: Detecting CCM	373	
16.2.1.1	Using <i>Modem</i> to Detect a CCM	373	
16.2.2	State 2a: Initializing CMM	374	
16.2.2.1	Viewing <i>Modem</i> 's Initializations	374	
16.2.3	State 2b: Waiting to Retry	376	
16.2.4	State 3: Set Mobile Network Operator	376	
16.2.5	State 4: Checking Registration Status	376	
16.2.5.1	Using <i>Modem</i> to Check Registration Status	377	
16.2.6	State 5: Connecting	378	
16.2.7	State 6: Manage Connection	378	
17	Sending and Receiving Text Messages (SMS)		379
17.1	Why Send/Receive Text Messages?	380	
17.1.1	Need to "Push" Information to an IoT Device	380	
17.1.2	Serverless IoT Devices That Interact with End-users	383	
17.2	Cost of Text Messaging via Cellular Modem	383	
17.3	Application-to-person (A2P) Messaging Is Often Regulated		384
17.4	Overview of Sending/Receiving Text Messages	385	
17.5	Sending Text Messages	386	
17.5.1	Set the Message Format	386	
17.5.2	Set Parameters for Sending	386	
17.5.3	Specify the Destination Phone Number and the Text to Send		389
17.5.4	What If Sending an SM Fails?	390	
17.5.5	Using <i>Modem</i> to Send a Text Message	390	
17.6	Receiving and Reading a Text Message	391	
17.6.1	Configure the CCM	392	
17.6.1.1	Set the Message Format	392	
17.6.1.2	Configure SMS Storage	392	
17.6.1.3	Check for a Received Text Message	394	
17.6.1.4	Using <i>Modem</i> to List Text Messages	395	
17.6.1.5	Delete a Text Message	396	

17.6.1.6	Using <i>Modem</i> to Delete a Text Message	396
17.7	SMS with Constrained Devices	397
17.7.1	Set the Message Format	397
17.7.2	Set Parameters for Writing to Mem-2	397
17.7.3	Specify the Destination Phone Number and Text to Store	397
17.7.4	Send a Text Message Already Stored in Mem-2	398
17.7.5	Verifying a Text Message Was Sent from Mem-2	399
17.8	Integrating SMS into CIP	400
18	Power Saving Modes and Techniques	403
18.1	What Are Low-power CCMs (LP-CCMs)	404
18.2	Plenty of Power, Most of the Time	405
18.3	Low-power IoT Devices	407
18.3.1	Microcontroller Energy Consumption	407
18.3.2	Temperature Sensor Energy Consumption	408
18.4	Battery Capacity	408
18.5	Transmitter Power	409
18.6	Legacy (GSM) Power Consumption	410
18.7	Cellular Modem Energy Consumption	413
18.7.1	Additional Energy Consumption	414
18.8	Network Registration States—RRC_CONNECTED and RRC_IDLE	414
18.8.1	RRC_CONNECTED (Without DRX)	416
18.8.1.1	Scenario 1—Sending a Location Packet	417
18.8.1.2	Scenario 2—Fetching an Over-the-air Update	418
18.8.2	RRC_IDLE (Without DRX)	420
18.8.3	Discontinuous Reception (DRX)	422
18.8.3.1	Discontinuous Reception in RRC_IDLE (iDRX)	423
18.8.3.2	Discontinuous Reception in RRC_CONNECTED (cDRX)	424
18.8.4	Registration Characteristics Summary	426
18.9	Latency	427
18.10	Using Low-power CCM—Cat M and NB-IoT and Cat 1 bis	429
18.11	Power Saving Mode (PSM)	431
18.11.1	How to Enable PSM	434
18.11.1.1	Using <i>Modem</i> to Enter PSM	436
18.11.2	Verifying PSM and Possible Problems or Surprises	437
18.11.2.1	Using <i>Modem</i> to Check PSM Status	439
18.11.3	Actual PSM Cycle Length	439
18.11.4	Exiting PSM	440
18.11.4.1	Using <i>Modem</i> to Exit PSM	441
18.11.5	Sending Data from PSM Inactive	442

18.11.6	PSM Effectiveness	442
18.11.7	Integrating PSM into CIP	443
18.12	Extended Discontinuous Reception (eDRX)	445
18.12.1	How to Enable eDRX	447
18.12.1.1	Using <i>Modem</i> to Enable eDRX	448
18.12.2	Verifying eDRX Cycle Length	449
18.12.2.1	Using Modem to Check eDRX Status	450
18.12.3	Disabling eDRX	451
18.12.3.1	Using <i>Modem</i> to Disable eDRX	451
18.12.4	Integrating eDRX into CIP	451
18.13	When to Use PSM, eDRX, or Both	454
18.14	Don't Trust the Numbers	454
	A Unified Modeling Language (UML) Primer	455
A.1	Assumptions	455
A.2	UML Syntax	456
A.3	Visibility (Private, Protected, Public)	457
A.4	Attribute/Parameter/Method Names and Types	457
A.5	Class Attributes and Methods	458
A.6	Aggregation	459
A.7	Multiplicities	459
A.8	Inheritance	460
A.9	Interfaces	460
A.10	Hidden Attributes	461
A.11	Layout	462
A.12	State Diagrams	462
	B 3GPP AT Commands Used in This Book	465
	C The <i>Modem</i> Utility	469
C.1	Invoking the <i>Modem</i> Program	470
C.2	Flags	470
C.3	Commands	471
	Glossary	479
	Closing Notes	485
	Index	487