Design Patterns for Embedded Systems in C

An Embedded Software Engineering Toolkit

Bruce Powel Douglass, PhD

ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO
Newnes is an imprint of Elsevier

Newnes

Contents

Preface ..... *xv*
Acknowledgements ..... *xvii*
About the Author ..... *xviii*
Chapter 1 What Is Embedded Programming? *1*
1.1 What's Special About Embedded Systems? 1
1.1.1 Embedded Design Constraints 3
1.1.2 The Embedded Tool Chain 4
1.1.3 OS, RTOS, or Bareback? 5
1.1.4 Embedded Middleware 6
1.1.5 Codevelopment with Hardware 7
1.1.6 Debugging and Testing 8
1.2 OO or Structured – It's Your Choice 9
1.2.1 Classes 10
1.2.2 Objects 13
1.2.3 Polymorphism and Virtual Functions 14
1.2.4 Subclassing 15
1.2.5 Finite State Machines 25
1.3 What Did We Learn? 33
Chapter 2 Embedded Programming with The Harmony^{TM} for Embedded RealTime Process *35*
2.1 Basic Elements of the Harmony Process 36
2.1.1 A Quick Overview of the Development Workflow 36
2.1.2 What Is a Design Pattern? 44
2.1.3 Basic Structure of Design Patterns 45
2.1.4 How to Read Design Patterns in This Book 46
2.1.5 Using Design Patterns in Development 47
2.1.6 Observer Pattern 66
2.2 The Approach 77
2.3 What's Coming Up 78
Chapter 3 Design Patterns for Accessing Hardware *79*
3.1 Basic Hardware Access Concepts 81

3.2 Hardware Proxy Pattern 85
3.2.1 Abstract 85
3.2.2 Problem 86
3.2.3 Pattern Structure 86
3.2.4 Collaboration Roles 87
3.2.5 Consequences 88
3.2.6 Implementation Strategies 89
3.2.7 Related Patterns 89
3.2.8 Example 89
3.3 Hardware Adapter Pattern 96
3.3.1 Abstract 96
3.3.2 Problem 96
3.3.3 Pattern Structure 96
3.3.4 Collaboration Roles 96
3.3.5 Consequences 97
3.3.6 Implementation Strategies 98
3.3.7 Related Patterns 98
3.3.8 Example 98
3.4 Mediator Pattern 100
3.4.1 Abstract 100
3.4.2 Problem 100
3.4.3 Pattern Structure 101
3.4.4 Collaboration Roles 101
3.4.5 Consequences 102
3.4.6 Implementation Strategies 102
3.4.7 Related Patterns 102
3.4.8 Example 103
3.5 Observer Pattern 111
3.5.1 Abstract 111
3.5.2 Problem 112
3.5.3 Pattern Structure 112
3.5.4 Collaboration Roles 112
3.5.5 Consequences 114
3.5.6 Implementation Strategies 114
3.5.7 Related Patterns 115
3.5.8 Example 115
3.6 Debouncing Pattern 122
3.6.1 Abstract 123
3.6.2 Problem 123
3.6.3 Pattern Structure 123
3.6.4 Collaboration Roles 123
3.6.5 Consequences 125

3.6.6 Implementation Strategies 125
3.6.7 Related Patterns 126
3.6.8 Example 126
3.7 Interrupt Pattern 130
3.7.1 Abstract 130
3.7.2 Problem 130
3.7.3 Pattern Structure 131
3.7.4 Collaboration Roles 131
3.7.5 Consequences 132
3.7.6 Implementation Strategies 134
3.7.7 Related Patterns 135
3.7.8 Example 135
3.8 Polling Pattern 138
3.8.1 Abstract 138
3.8.2 Problem 138
3.8.3 Pattern Structure 138
3.8.4 Collaboration Roles 138
3.8.5 Consequences 141
3.8.6 Implementation Strategies 141
3.8.7 Related Patterns 142
3.8.8 Example 142
3.9 So, What Did We Learn? 147

Chapter 4 Design Patterns for Embedding Concurrency and Resource Management 149

4.1 Basic Concurrency Concepts 152
4.1.1 Identifying Tasks 160
4.1.2 Concurrency in the UML 161
4.1.3 Real-Time Operating Systems 163
Scheduling Patterns 164
4.2 Cyclic Executive Pattern 164
4.2.1 Abstract 165
4.2.2 Problem 165
4.2.3 Pattern Structure 165
4.2.4 Collaboration Roles 165
4.2.5 ConcreteCEThread 166
4.2.6 Consequences 167
4.2.7 Implementation Strategies 167
4.2.8 Related Patterns 167
4.2.9 Example 167
4.3 Static Priority Pattern 170
4.3.1 Abstract 170
4.3.2 Problem 171

4.3.3 Pattern Structure 171
4.3.4 Collaboration Roles 172
4.3.5 Consequences 173
4.3.6 Implementation Strategies 173
4.3.7 Related Patterns 174
4.3.8 Example 175
Task Coordination Patterns 182
4.4 Critical Region Pattern 182
4.4.1 Abstract 182
4.4.2 Problem 182
4.4.3 Pattern Structure 182
4.4.4 Collaboration Roles 183
4.4.5 Consequences 184
4.4.6 Implementation Strategies 184
4.4.7 Related Patterns 184
4.4.8 Example 184
4.5 Guarded Call Pattern 190
4.5.1 Abstract 190
4.5.2 Problem 190
4.5.3 Pattern Structure 190
4.5.4 Collaboration Roles 191
4.5.5 Consequences 192
4.5.6 Implementation Strategies 192
4.5.7 Related Patterns 193
4.5.8 Example 194
4.6 Queuing Pattern 207
4.6.1 Abstract 207
4.6.2 Problem 207
4.6.3 Pattern Structure 208
4.6.4 Collaboration Roles 208
4.6.5 Consequences 209
4.6.6 Implementation Strategies 210
4.6.7 Related Patterns 210
4.6.8 Example 210
4.7 Rendezvous Pattern 224
4.7.1 Abstract 225
4.7.2 Problem 226
4.7.3 Pattern Structure 226
4.7.4 Collaboration Roles 226
4.7.5 Consequences 227
4.7.6 Implementation Strategies 228

4.7.7 Related Patterns 228
4.7.8 Example 228
Deadlock Avoidance Patterns 231
4.8 Simultaneous Locking Pattern 232
4.8.1 Abstract 232
4.8.2 Problem 233
4.8.3 Pattern Structure 233
4.8.4 Collaboration Roles 234
4.8.5 Consequences 236
4.8.6 Implementation Strategies 236
4.8.7 Related Patterns 236
4.8.8 Example 236
4.9 Ordered Locking 242
4.9.1 Abstract 242
4.9.2 Problem 243
4.9.3 Pattern Structure 243
4.9.4 Collaboration Roles 244
4.9.5 Consequences 247
4.9.6 Implementation Strategies 248
4.9.7 Related Patterns 248
4.9.8 Example 248
4.10 So, What Have We Learned? 255

Chapter 5 Design Patterns for State Machines 257

5.1 Oh Behave! 259
5.2 Basic State Machine Concepts 261
5.2.1 OR-States 263
5.2.2 AND-States 265
5.2.3 Special Stuff: Timeouts, Conditionals, and More 267
5.2.4 Synchronous versus Asynchronous? 270
5.3 Single Event Receptor Pattern 274
5.3.1 Abstract 274
5.3.2 Problem 274
5.3.3 Pattern Structure 274
5.3.4 Collaboration Roles 275
5.3.5 Consequences 278
5.3.6 Implementation Strategies 278
5.3.7 Related Patterns 278
5.3.8 Example 278
5.4 Multiple Event Receptor Pattern 287
5.4.1 Abstract 287
5.4.2 Problem 287

5.4.3 Pattern Structure 288
5.4.4 Collaboration Roles 288
5.4.5 Consequences 290
5.4.6 Implementation Strategies 290
5.4.7 Related Patterns 290
5.4.8 Example 290
5.5 State Table Pattern 296
5.5.1 Abstract 296
5.5.2 Problem 296
5.5.3 Pattern Structure 296
5.5.4 Collaboration Roles 297
5.5.5 Consequences 299
5.5.6 Implementation Strategies 299
5.5.7 Related Patterns 299
5.5.8 Example 300
5.6 State Pattern 308
5.6.1 Abstract 308
5.6.2 Problem 308
5.6.3 Pattern Structure 308
5.6.4 Collaboration Roles 310
5.6.5 Consequences 312
5.6.6 Implementation Strategies 313
5.6.7 Related Patterns 313
5.6.8 Example 314
5.7 AND-States 327
5.8 Decomposed AND-State Pattern 330
5.8.1 Abstract 331
5.8.2 Problem 331
5.8.3 Pattern Structure 331
5.8.4 Collaboration Roles 331
5.8.5 Consequences 333
5.8.6 Implementation Strategies 333
5.8.7 Related Patterns 334
5.8.8 Example 334
5.9 OK, What Have We Learned? 356

Chapter 6 Safety and Reliability Patterns 357

6.1 A Little Bit About Safety and Reliability 359
6.1.1 Safety and Reliability Related Faults 360
6.1.2 Achieving Safety and Reliability 361
6.2 One's Complement Pattern 362
6.2.1 Abstract 362
6.2.2 Problem 362

6.2.3 Pattern Structure 362
6.2.4 Collaboration Roles 362
6.2.5 Consequences 363
6.2.6 Implementation Strategies 363
6.2.7 Related Patterns 363
6.2.8 Example 363
6.3 CRC Pattern 367
6.3.1 Abstract 367
6.3.2 Problem 367
6.3.3 Pattern Structure 368
6.3.4 Collaboration Roles 368
6.3.5 Consequences 369
6.3.6 Implementation Strategies 369
6.3.7 Related Patterns 369
6.3.8 Example 369
6.4 Smart Data Pattern 380
6.4.1 Abstract 380
6.4.2 Problem 381
6.4.3 Pattern Structure 381
6.4.4 Collaboration Roles 382
6.4.5 Consequences 382
6.4.6 Implementation Strategies 383
6.4.7 Related Patterns 383
6.4.8 Example 383
6.5 Channel Pattern 395
6.5.1 Abstract 395
6.5.2 Problem 395
6.5.3 Pattern Structure 395
6.5.4 Collaboration Roles 395
6.5.5 Consequences 396
6.5.6 Implementation Strategies 397
6.5.7 Related Patterns 397
6.5.8 Example 397
6.6 Protected Single Channel Pattern 402
6.6.1 Abstract 402
6.6.2 Problem 402
6.6.3 Pattern Structure 403
6.6.4 Collaboration Roles 404
6.6.5 Implementation Strategies 405
6.6.6 Related Patterns 405
6.6.7 Example 405

6.7 Dual Channel Pattern 413
6.7.1 Abstract 413
6.7.2 Problem 414
6.7.3 Pattern Structure 414
6.7.4 Collaboration Roles 414
6.7.5 Consequences 417
6.7.6 Implementation Strategies 417
6.7.7 Related Patterns 417
6.7.8 Example 421
6.8 Summary 422

Appendix A UML Notation 425
1.1 Class Diagram 425
1.1.1 Semantic Elements 425
1.2 Sequence Diagram 429
1.2.1 Semantic Elements 429
1.3 State Diagram 430
1.3.1 Semantic Elements 430

Index 437