

Contents

PREFACE	xlii
1 MECHATRONICS	1
1.1 A History of Increasing Complexity	2
1.2 Mechatronic System Organization	3
1.3 Amplifiers and Isolation	3
1.4 Scope: The Unit Machine	4
1.5 Control	5
1.6 Real-Time Software	5
1.7 Nasty Software Properties	8
1.8 Engineering Design and Computational Performance	9
1.9 Control System Organization	10
1.10 Software Portability	10
1.11 Operator Interface	11
1.12 Multicomputer Systems: Communication	12
1.13 The Design and Implementation Process	12
1.13.1 Performance Specification	12
1.13.2 Design Documentation	13
1.13.3 Simulation	14
1.13.4 Laboratory Prototype	15
1.13.5 Production Prototype	16
1.13.6 Production System	17
1.13.7 Maintenance	17
2 TASKS	19
2.1 Example: Task Selection in a Process System	20
2.2 Tasks and the Control Hierarchy	21
2.2.1 <i>Intertask Communication</i>	22
2.3 Task Structure Examples	22
2.3.1 Velocity Control of a DC Motor	23
2.3.2 Heater Control	24
2.3.3 Toaster Oven	26
2.3.4 Flexible Position Control of a DC Motor	27

2.4	Simulation	29
2.5	More Task Structure Examples	31
2.5.1	Coordinated, Two-Axis Motion	31
2.5.2	A Washing Machine	32
3	STATE TRANSITION LOGIC	35
3.1	States and Transitions	36
3.2	Transition Logic Diagrams	36
3.3	Tabular Form for Transition Logic	37
3.4	Example: Pulse-Width Modulation (PWM)	38
3.5	Transition Logic for the Process Control Example	39
3.6	Nonblocking State Code	41
3.7	State-Related Code	41
3.8	State Scanning: The Execution Cycle	42
3.9	Task Concurrency: Universal Real-Time Solution	43
4	DIRECT REALIZATION OF SYSTEM CONTROL SOFTWARE	45
4.1	Language	45
4.2	Time	47
4.3	Program Format	47
4.4	Simulation	48
4.5	Simulation in Matlab	48
4.5.1	Templates for Simulation Using Matlab	48
4.5.2	Simulation of PWM Generator	53
4.5.3	Simulation of Three-Tank Process System	57
4.6	Intertask Communication	61
4.7	Real-Time Realization	62
4.8	Real-Time Realization with Matlab	62
4.8.1	Heater Control Implementation in Matlab	63
5	SOFTWARE REALIZATION IN C++	67
5.1	Simulation in C++	67
5.2	Templates for Simulation in C++ (group-priority)	68
5.3	PWM Simulation Using C++ (group-priority)	80
5.4	Simulation in C++ (with TranRun4)	82
5.4.1	Components	82
5.4.2	The Master Scheduler	84
5.4.3	Process Objects and Task Lists	85
5.4.4	Task Objects	86
5.4.5	Tasks with No State Object	89
5.4.6	Creating Task Classes	89
5.4.7	State Objects	91
5.4.8	Creating State Classes	93
5.4.9	The Main File and UserMain() Function	94
5.5	Real-Time Realization with C++	97

6	INTERTASK COMMUNICATION	99
6.1	Communication Within a Process	100
6.1.1	Data Integrity	100
6.1.2	Design Rules	102
6.2	Communication Across Processes	105
6.2.1	Message Passing	105
6.2.2	Message Passing in the Group Priority Scheduler	106
6.2.3	Message Passing in the TranRun4 Scheduler	112
6.2.4	Distributed Database	115
6.2.5	Distributed Database in the Group Priority Scheduler	116
6.2.6	Distributed Database in the TranRun4 Scheduler	118
7	TIMING TECHNIQUES ON PC COMPATIBLES	121
7.1	Calibrated Time	121
7.2	Free-Running Timer	122
7.2.1	Hardware Timers on the PC	123
7.2.2	Performance Timers in Unix and Windows	124
7.3	Interrupt-Based Timing	125
8	MULTITASKING: PERFORMANCE IN THE REAL WORLD	127
8.1	Priority-Based Scheduling—Resource Shifting	127
8.1.1	Continuous vs. Intermittent Tasks	128
8.1.2	Cooperative Multitasking Modes	129
8.2	Matlab Template for Minimum-Latency Dispatcher	131
8.2.1	Example: Simulation of PWM-Actuated Heater	131
8.3	Cooperative Multitasking Using C++	133
8.3.1	Inheriting Task Behavior Two PWMs	137
8.4	Preemptive Multitasking Modes	138
8.5	Realization of Interrupt-Based Dispatching	140
8.5.1	How Many Priority Levels Are Necessary?	141
8.5.2	Which Interrupt Sources Will Be Used?	141
8.5.3	Interrupt-Based Dispatching Functions	142
8.5.4	Attaching Dispatching Functions to Interrupts	143
9	A CHARACTER-BASED OPERATOR INTERFACE	145
9.1	Operator Interface Requirements	145
9.2	Context Sensitive Interfaces	146
9.3	User Interface Programming Paradigms	147
9.4	Mechatronics System Operator Interface	147
9.5	Operator Interface Programming	148
9.5.1	The Operator Screen	148
9.5.2	Programming Conventions in C++	149
9.5.3	Heater Control Operator Interface	151

10 GRAPHICAL OPERATOR INTERFACES	155
10.1 Graphical Environments	156
10.1.1 Windowing Software: Events and Messages	156
10.1.2 Operator Interface vs. Standard Windowing Application	157
10.1.3 Simplified Programming for Windowing Systems	157
10.1.4 The Ease-of-Use Challenge	158
10.1.5 Methods of Simplifying Window-Style Programming	158
10.2 The Times-2 Problem	158
10.2.1 Times-2: Character-Based Interface	158
10.2.2 Times-2: Visual Basic	160
10.2.3 Times-2: Bridgeview	162
10.3 Screen Change	166
10.3.1 Screen Change in Visual Basic	166
10.3.2 Screen Change: Bridgeview	168
10.4 Heat Exchanger Control in Bridgeview	171
10.5 Interprocess Communication: DDE	173
10.5.1 DDE: The C++ Side	173
10.5.2 Communicating with Excel	176
10.5.3 A DDE Server in C++	177
10.5.4 DDE Communication Between C++ and Visual Basic	178
10.5.5 DDE Communication Between C++ and Bridgeview	180
10.6 Putting It All Together	181
11 DISTRIBUTED CONTROL I: NET BASICS	185
11.1 Multiprocessor Architectures	186
11.1.1 Symmetric Multiprocessing (SMP)	186
11.1.2 Buses	186
11.1.3 Networks	187
11.1.4 Point-to-Point Connections	189
11.2 TCP/IP Networking	190
11.2.1 The Physical Context	190
11.2.2 Interconnection Protocols	191
11.2.3 TCP and UDP	191
11.2.4 Client/Server Architecture	192
11.3 Implementation of UDP	192
11.3.1 Sockets	192
11.3.2 Setting Up for Network Data Exchange	193
11.3.3 Nonblocking Network Calls	195
11.3.4 Receiving Information	195
11.3.5 Client-Side Setup	196
11.4 The Application Layer	197
11.4.1 Data Coding	197
11.4.2 Building the Packet	198
11.4.3 Parsing a Packet	200

12 DISTRIBUTED CONTROL II: A MECHATRONICS CONTROL APPLICATION LAYER	203
12.1 Control System Application Protocol	203
12.2 Startup of Distributed Control Systems	207
12.3 Testing the Application Protocol	208
12.4 Using the Control Application Protocol	209
12.5 Compiling	212
13 JAVA FOR CONTROL SYSTEM SOFTWARE	213
13.1 The Java Language and API	214
13.1.1 Networking	214
13.1.2 AWT/Swing	214
13.1.3 Multithreading	214
13.2 Preconditions for Real-Time Programming in Java	215
13.2.1 Deterministic Garbage Collection	215
13.2.2 Memory and Hardware Access	215
13.2.3 Timing	216
13.3 Advantages of Java for Control Software Design	216
13.3.1 Modularity	216
13.3.2 Distributed Control	217
13.3.3 Platform Independence and Prototyping	217
13.3.4 Operator Interface Design	217
13.4 Java and the Task/State Design Method	218
13.4.1 Inner Classes	218
13.4.2 Networking	218
13.4.3 Documentation	219
13.5 The Current State of Real-Time Java	219
14 PROGRAMMABLE LOGIC CONTROLLERS (PLCs)	221
14.1 Introduction	221
14.2 Goals	222
14.3 PLC Programming	223
14.3.1 When to Use a PLC	223
14.3.2 Ladder Logic	224
14.3.3 Grafset/Sequential Flow Charts	226
14.4 The Task/State Model	226
14.5 State Transition Logic for a PLC	227
14.5.1 State Variables	227
14.5.2 Ladder Organization	227
14.5.3 Transitions	228
14.5.4 Outputs	229
14.5.5 Entry Activity	229
14.5.6 Action Outputs	229
14.5.7 Exit (Transition-Based) Outputs	229
14.5.8 Common Exit Activities	230

14.6 PLC Multitasking	230
14.7 Modular Design	231
14.8 Example: Model Railroad Control	231
14.9 Simulation – Portability	232
15 ILLUSTRATIVE EXAMPLE: ASSEMBLY SYSTEM	235
15.1 The Assembly System	235
15.2 System Simulation	237
15.3 Development Sequence	237
15.4 Belt Motion Simulation (Glue00)	238
15.4.1 Modeling Belt Dynamics	238
15.4.2 Definition of Task Classes	239
15.4.3 Instantiating Tasks: the Main File	241
15.4.4 The Simulation Task	242
15.4.5 The Data Logging Task	244
15.4.6 Timing Mode	245
15.4.7 Compiling	246
15.4.8 Results	246
15.5 Oven Temperature Simulation (Glue01)	247
15.6 PID Control of Belt Position and Oven Temperature (Glue02)	247
15.6.1 Keeping Classes Generic	248
15.6.2 The PIDControl Class	248
15.6.3 Results	250
15.7 Better Control of Motion (Glue03)	250
15.7.1 Trapezoidal Motion Profile	251
15.7.2 Motion Profile Class	252
15.7.3 Profiler State Structure	253
15.7.4 Round-Off Error	257
15.7.5 Discretization Errors in Simulation	257
15.8 A Command Structure for Profiled Motion (Glue04)	260
15.8.1 Message-Based Command Structure	260
15.8.2 State Transition Audit Trail	261
15.8.3 Motion Results	263
15.9 Clamps (Glue05)	263
15.10 Robots (Glue06)	265
15.11 Cure/Unload (Glue07)	266
15.12 Making Widgets (Glue08)	271
16 THE GLUING CELL EXERCISE IN TRANRUN4	273
16.1 The Gluing System	273
16.2 Simulation and Prototyping	274
16.3 The Project Components	274
16.4 Glue00: Conveyor Simulation	275
16.4.1 The Dynamic Model	275
16.4.2 Creating the Conveyor Task	277

16.4.3 The Data Logging Task	280
16.4.4 Data Communication Between Tasks	284
16.4.5 The Main File	286
16.4.6 Glue00 Results	288
16.5 Glue01: An Oven Simulation	288
16.5.1 Configuration and Status Printouts	289
16.6 Glue02: PID Control	291
16.7 Glue03: The Operator Interface	292
16.7.1 Results	297
16.8 Glue04: Motion Profiling	299
16.9 Glue05: Belt Sequencing	306
16.10 Glue06: The Glue Application Machine	307
16.11 Glue07: Transport Task Supervision	309
16.12 Glue08: The Completed Assembly System	311
17 THE GLUING CELL EXERCISE IN TRANRUNJ	315
17.1 Getting Started	315
17.1.1 Program Entry Point	315
17.1.2 The userMain Method	316
17.2 Writing Custom Tasks and States	317
17.2.1 Creating a Task Class	317
17.2.2 Creating a State Class	319
17.3 Implementing State Transition Logic	320
17.4 Global Data and Intertask Messaging	321
17.4.1 Global Data Items	321
17.4.2 Task Messages	322
17.5 Continuous vs. Intermittent Tasks	323
17.6 Scheduler Internals	324
17.6.1 Operating System Processes vs. CProcess	324
17.6.2 Foreground vs. Background Execution Lists	325
17.6.3 Scheduling Modes	325
17.7 Execution Profiling	325
17.8 Intertask Messaging Across Different Processes	326
17.9 Tips And Tricks	328
17.9.1 Judicious Use of Execution-Time Profiling	328
17.9.2 Integer Labels for Global Data and Task Message Inboxes	328
17.9.3 The TaskMessageListener Interface	328
17.9.4 Scheduler Sleeping	329
17.9.5 Anonymous State Classes	329
17.10 Additional Information	330
BIBLIOGRAPHY	331
INDEX	333