
Contents

Part I Concepts

1	Real-time Characteristics and Safety of Embedded Systems	3
1.1	Introduction	3
1.2	Real-time Systems and their Properties	5
1.2.1	Definitions, Classification and Properties	6
1.2.2	Problems in Adequate Implementation of Embedded Applications and General Guidelines	10
1.3	Safety of Embedded Computer Control Systems	13
1.3.1	Brief History of Safety Standards Relating to Computers in Control	16
1.3.2	Safety Integrity Levels	19
1.3.3	Dealing with Faults in Embedded Control Systems	21
1.3.4	Fault-tolerance Measures	23
1.4	Summary of Chapter 1 and Synopsis of What Follows	28
2	Multitasking	29
2.1	Task Management Systems	29
2.1.1	Cyclic Executive	30
2.1.2	Asynchronous Multitasking	32
2.2	Scheduling and Schedulability	34
2.2.1	Scheduling Methods and Techniques	35
2.2.2	Deadline-driven Scheduling	39
2.2.3	Sufficient Condition for Feasible Schedulability Under Earliest Deadline First	41

2.2.4	Implications of Employing Earliest Deadline First Scheduling.....	45
2.2.5	Rate Monotonic vs Earliest Deadline First Scheduling..	46
2.3	Synchronisation Between Tasks.....	50
2.3.1	Busy Waiting	51
2.3.2	Semaphores.....	53
2.3.3	Bolts	54
2.3.4	Monitors	55
2.3.5	Rendezvous	56
2.3.6	Bounding Waiting Times in Synchronisation	57
3	Hardware and System Architectures	61
3.1	Undesirable Properties of Conventional Hardware Architectures and Implementations	62
3.1.1	Processor Architectures	63
3.1.2	System Architectures	67
3.2	Top-layer Architecture: An Asymmetrical Multiprocessor System.....	69
3.2.1	Concept	70
3.2.2	Operating System Kernel Processor	73
3.2.3	Task Processor	78
3.3	Implementation of Architectural Models	82
3.3.1	Centralised Asymmetrical Multiprocessor Model.....	83
3.3.2	Distributed Multiprocessor Model	86
3.4	Intelligent Peripheral Interfaces for Increased Dependability and Functionality.....	86
3.4.1	Higher-level Functions of the Intelligent Peripheral Interfaces	88
3.4.2	Enhancing Fault Tolerance	89
3.4.3	Support for Programmed Temporal Functions.....	90
3.4.4	Programming Peripheral Interfaces	93
3.5	Adequate Data Transfer	93
3.5.1	Real-time Communication	94
3.5.2	Time-triggered Communication	95
3.5.3	Fault Tolerance in Communication	98
3.5.4	Distributed Data Access: Distributed Replicated Shared Memory	100

4	Programming of Embedded Systems	107
4.1	Properties Desired of Control Systems Development	111
4.1.1	Support for Time and Timing Operations	111
4.1.2	Explicit Representation of Control System Entities	116
4.1.3	Explicit Representation of Other Control System Entities	119
4.1.4	Support for Temporal Predictability	120
4.1.5	Support for Low-level Interaction with Special-purpose Hardware Devices	121
4.1.6	Support for Overload Prevention	124
4.1.7	Support for Handling Faults and Exceptions	124
4.1.8	Support for Hardware/Software Co-implementation	130
4.1.9	Other Capabilities	132
4.2	Time Modeling and Analysis	132
4.2.1	Execution Time Analysis of Specifications	135
4.2.2	Execution Time Analysis of Source Code	136
4.2.3	Execution Time Analysis of Executable Code	140
4.2.4	Execution Time Analysis of Hardware Components	141
4.2.5	Direct Measurement of Execution Times	142
4.2.6	Programming Language Support for Temporal Predictability	144
4.2.7	Schedulability Analysis	147
4.3	Object-orientation and Embedded Systems	149
4.3.1	Difficulties of Introducing Object-orientation to Embedded Real-time Systems	150
4.3.2	Integration of Objects into Distributed Embedded Systems	150
4.4	Survey of Programming Languages for Embedded Systems	156
4.4.1	Assembly Language	157
4.4.2	General-purpose Programming Languages	158
4.4.3	Special-purpose Real-time Programming Languages	160
4.4.4	Languages for Programmable Logic Controllers	163

Part II Implementation

5	Hardware Platform	169
5.1	Architecture	169

5.2	Communication Module Used in Processing and Peripheral Units	171
5.3	Fault Tolerance of the Hardware Platform	175
5.4	System Software of the Experimental Platform	176
6	Implementation of a Fault-tolerant Distributed Embedded System	181
6.1	Generalised Model of Fault-tolerant Real-time Control Systems	182
6.2	Implementation of Logical Structures on the Hardware Platform	185
6.3	Partial Implementation in Firmware	187
6.3.1	Communication Support Module	188
6.3.2	Supporting Middleware for Distributed Shared Memory	189
6.3.3	Kernel Processor	190
6.3.4	Implementation of Monitoring, Reconfiguration and Mode Control Unit	195
6.4	Programming of the FTCs	196
6.4.1	Extensions to MATLAB [®] /Simulink [®] Function Block Library	196
6.4.2	Generation of Time Schedules for the TTCAN Communication Protocol	197
6.4.3	Development Process	199
7	Asynchronous Real-time Execution with Runtime State Restoration by <i>Martin Skambraks</i>	201
7.1	Design Objectives	201
7.2	Task-oriented Real-time Execution Without Asynchronous Interrupts	202
7.2.1	Operating Principle	203
7.2.2	Priority Inheritance Protocol	206
7.2.3	Aspects of Safety Licensing	211
7.2.4	Fragmentation of Program Code	213
7.3	State Restoration at Runtime	220
7.3.1	State Restoration at Runtime and Associated Problems	222
7.3.2	Classification of State Changes	226
7.3.3	State Restoration with Modification Bits	227
7.3.4	Concept of State Restoration	229
7.3.5	Influence on Program Code Fragmentation and Performance Aspects	233

8	Epilogue	237
	References	241
	Index	247