

Contents

Part I Trustworthy Embedded Systems

1	Introduction. OpenComRTOS role in a unified systems engineering methodology	3
1.1	Introduction	3
1.2	A systematic engineering methodology based on Unified Semantics and Interacting Entities	6
1.3	Interacting entities for the software domain	9
1.3.1	Silicon technology advances	9
1.3.2	Silicon technology limitations	10
1.3.3	The world becomes connected	11
1.4	A link with the work plan in a systems engineering project	11
1.5	System engineering methods and engineering standards	12
1.6	Where do formal techniques fit in?	12
2	Requirements and specifications for the OpenComRTOS project	15
2.1	Background of OpenComRTOS	15
2.2	Early requirements derived from the Virtuoso RTOS	17
2.3	Real-time embedded programming	19
2.3.1	Why real-time?	19
2.3.2	Why a simple loop is often not enough	20
2.3.3	Superloops and static scheduling	21
2.3.4	Rate Monotonic Analysis	24
2.3.5	The application of RMA in OpenComRTOS	26
2.3.6	The issue of priority inversion and its inadequate solution	27
2.4	Next generation requirements	30
2.5	Top level requirements for OpenComRTOS	32
2.6	Specifications derived from Requirements	33
2.7	Systems and application grammar of OpenComRTOS	36
2.7.1	Base principles and definitions	36
2.7.2	A note on typing conventions	37

2.7.3	OpenComRTOS L1 System and Application Grammar	37
-------	---	----

Part II Formal Modeling Fundamentals

3	The Choice of TLA⁺/TLC: Comparing Formal Methods	43
3.1	Formal Methods Survey and Pre-Selection	43
3.2	Case Study	44
3.2.1	Introduction	45
3.2.2	The Algorithm	45
3.2.3	Remarks	47
3.2.4	Drawbacks	47
3.2.5	Related Work	48
3.3	TLA ⁺ and TLC	49
3.3.1	Overview	49
3.3.2	Model developed	51
3.4	Promela and SPIN	57
3.4.1	Overview	57
3.4.2	Model developed	60
3.5	Comparison	63
4	Basic Formal Specification in TLA⁺	71
4.1	Introduction	71
4.1.1	Goal: awareness in specifying systems	71
4.1.2	A two-step approach	71
4.2	Structure of TLA ⁺ specifications	72
4.2.1	Basic structure	72
4.2.2	Module structure	73
4.3	Introducing TLA ⁺ by example	74
4.3.1	Basic TLA ⁺ notions	74
4.3.2	Basic examples: TLA ⁺ sequences and OpenComRTOS lists	75
4.3.3	An extended example: the module <i>Port</i>	77
4.4	Conclusion	83

Part III OpenComRTOS Design

5	Formal modelling of the RTOS entities	87
5.1	Introduction	87
5.2	OpenComRTOS environment model	88
5.2.1	Term definitions	89
5.2.2	Constants	89
5.2.3	Variables representing the System State	89
5.2.4	The L1-Packet	90
5.2.5	General Constraint for all Models	91
5.3	Formal Model of the Semaphore-Entity	91
5.3.1	Constants	91
5.3.2	Variables	92

5.3.3	Initialisation	92
5.3.4	Signalling the Semaphore	92
5.3.5	Testing the Semaphore	95
5.3.6	Constraints	99
5.3.7	Defining the Next State	99
5.3.8	Properties to check	99
5.3.9	Proof obligations	100
5.3.10	Checking the models	102
5.4	Model verification	102
5.5	Conclusion	103
6	Final architecture of the RTOS	105
6.1	The Building Blocks of OpenComRTOS	105
6.1.1	The Hub Entity of OpenComRTOS	106
6.1.2	Tasks	111
6.1.3	Packets	113
6.2	The Semaphore Loop	113
6.2.1	The semaphore loop in detail	114
6.2.2	Heterogeneous Multiprocessor Systems and their issues	116
6.3	OpenComRTOS development process for applications	117
6.4	Summary	117
7	Task interaction models in OpenComRTOS	119
7.1	Introduction	119
7.2	Modelling Task Interaction	121
7.3	Timing Properties of Task Interactions	124
7.4	Notes on Asynchronous Interactions	126
7.5	Conclusions	129
8	Results: code size and performance	131
8.1	Summary	131
8.2	Metrics of success	131
8.2.1	Code size	132
8.2.2	Total memory use	135
8.2.3	Influence of processor architecture	136
8.2.4	Semaphore loop	136
8.2.5	Interrupt latency	137
Part IV APPENDIX		
A	OpenComRTOS-Suite 1.3 Usage Tutorial	141
A.1	Developing a Single Node Semaphore-Loop Project	141
A.2	Going Distributed with OpenComRTOS	148
A.3	Tracing in OpenComRTOS	153
A.3.1	How to enable tracing	154
A.3.2	How to retrieve a trace	154

A.3.3	Retrieving and displaying traces from distributed systems . .	157
A.4	Measuring the interrupt latency of OpenComRTOS	157
A.4.1	Designing distributed heterogeneous systems using the OpenComRTOS Suite	157
A.4.2	Presenting the measurement results	160
A.4.3	Specifying the system	160
A.4.4	Implementation	162
A.4.5	Application	162
A.4.6	Collected Measurement Results	163
A.5	Summary	164
B	Foundations for TLA⁺ and Temporal Logic	167
B.1	Introduction	167
B.1.1	Goal: Increased awareness in specifying systems	167
B.1.2	Approach and overview	168
B.2	A unifying formalism	169
B.2.1	Rationale	169
B.2.2	Syntax	169
B.2.3	Style of use	171
B.2.4	Introducing TLA ⁺ via Funmath	175
B.3	Faithful formalization of informal specifications	177
B.3.1	Choice of proper data abstractions	178
B.3.2	Auxiliary functions in formal specifications	180
B.4	Calculational reasoning and patterns in TLA ⁺	182
B.4.1	Capturing temporal logics by temporal calculi	182
B.4.2	A functional temporal calculus (FTC)	183
B.4.3	Defining the <i>Temporal Calculus of Actions</i> (TCA)	186
B.4.4	Calculational reasoning in TCA/TLA ⁺	188
B.4.5	Applications to patterns in TLA ⁺	190
B.5	Conclusions	192
C	Comparison of Formal Methods	193
C.1	TLA ⁺ model of Harris' Algorithm	193
C.2	Promela model of Harris' Algorithm	199
	List of Figures	205
	List of Tables	207
	Glossary	209
	References	211
	Index	215