

1	Einleitung	8
2	Warum Java?	10
2.1	Portabilität	11
2.2	Objektorientierung	11
2.3	Multithreading	12
2.4	Verteilung	12
2.5	Robustheit	13
2.6	Produktivität	13
3	Was braucht Java für die Echtzeitverarbeitung?	15
3.1	Kurze, garantierte Wartezeiten auf die Unterbrechbarkeit	15
3.2	Schnelle und deterministische Befehlsabarbeitung	19
3.3	Zugriff auf EA-Signale	22
4	Das SICOMP Echtzeit-Java-System	27
4.1	Das Gesamtsystem	27
4.2	Das JFPC-System	28
4.2.1	Zugriff zum JFPC-System	31
4.2.2	Die EA-Proxy Variable	32
4.2.3	Die Gruppe	39
4.2.4	Grenzen der Zugriffsabstraktion	51
4.3	Die SICOMP-RTVM	54
4.4	Die Abbildung der Java-Systemfunktionen auf RMOS	54
4.4.1	Die Speicherplatzverwaltung	55
4.4.2	Die Threadverwaltung	56
4.4.3	Sonstige Systemverwaltung	56
4.5	Der Anschluss von native Funktionen	57
4.5.1	Anbindung von native Funktionen in DLLs	58
5	Standardisierung der Echtzeiterweiterung	61
5.1	Historie	61
5.2	Aktueller Stand: drei Spezifikationen	62
5.2.1	Real-Time Specification for Java	62
5.2.2	Real-Time Core Extensions for the Java Platform	63
5.2.3	Real-Time Data Access	63
6	Laufzeiten von Java auf SICOMP	65
6.1	Einführung und Zielsetzung	65

6.2	Die Messanordnung	67
6.2.1	Die Systemplattform	67
6.2.2	Softwarekomponenten	68
6.3	Messverfahren	68
6.3.1	Differenzzeitmessung	69
6.3.2	Messung mittels Zeitgeber (Timer)	70
6.4	Messungen	72
6.4.1	Differenzzeitmessungen	73
6.4.2	Messung mittels Zeitgeber (Timer)	82
6.5	Zusammenfassung der Messergebnisse	84
6.5.1	Ergebnisse nach der Differenzzeitmessung	84
6.5.2	Ergebnisse nach der Messung mittels Zeitgeber	85
7	Referenzen	87
7.1	ActivateListener-Interface	87
7.2	JpcGroupItf-Interface	88
7.3	JpcItemItf-Interface	95
7.4	JpcNamingEnumeration-Interface	100
7.5	JpcNamingItf-Interface	101
7.6	JpcServerItf-Interface	104
8	Literaturverzeichnis	113
9	Glossar/Abkürzungen	114
	Stichwortverzeichnis	118