

Rolf Swik / Denis Hatebur
Thomas Rottke

Messen, Steuern und Regeln mit **Windows CE**

Ein konstruktiver Leitfaden zum Aufbau eines Mess-, Steuerungs-
und Regelungssystems mit Windows CE und CAN

Mit 54 Abbildungen

Franzis'

Inhalt

1	Einleitung	9
2	Architektur von Automationssystemen für verteiltes MSR	16
2.1	Einleitung.....	16
2.2	MSR (Messen, Steuern, Regeln).....	17
2.2.1	Anwendungsbereiche.....	17
2.2.2	Begriffe der Regelungstechnik (in Anlehnung an DIN 19226).....	18
2.2.3	Begriffe der Steuerungstechnik (DIN 19237).....	20
2.2.4	Zweipunktregelung.....	21
2.2.5	Typische Hardware für digitales MSR.....	22
2.3	Embedded Systeme.....	23
2.3.1	Begriffsdefinitionen und Anwendungsbereiche.....	23
2.3.2	Hardware für Embedded Systeme.....	24
2.3.3	Echtzeitsysteme.....	25
2.3.4	Betriebssysteme für Embedded Systeme.....	26
2.3.5	Bussysteme und Protokolle.....	29
2.4	3-Ebenen Architektur.....	37
2.4.1	Definition 3-Ebenen Architektur.....	37
2.4.2	Klassische Systemstruktur: SPS-Lösung.....	37
2.4.3	Moderne Systemstruktur: Internet-Ethernet-Lösung.....	38
2.4.4	Hybrid-Lösung mit 3 Ebenen.....	39
3	Aufbau von Windows CE, Programmierwerkzeuge	43
3.1	Einleitung.....	43
3.2	Anwendungsbereiche von Windows CE.....	45
3.3	Aufbau von Windows CE, Betriebssysteminterna.....	46
3.3.1	Struktur/Aufbau von Windows CE.....	46
3.3.2	Prozesse, Threads und Speichermanagement bei Windows CE.....	48
3.3.3	Multitasking und Scheduling bei Windows CE.....	50
3.3.4	Interrupthandling.....	51
3.3.5	Treiber unter Windows CE.....	52

3.3.6	Synchronisation von Prozessen und Threads bei Windows CE	54
3.3.7	Kommunikation zwischen Prozessen und Threads bei Windows CE.....	56
3.3.8	Daten- und Dateiverwaltung bei Windows CE	57
3.4	Programmierschnittstelle (API)	58
3.4.1	Grundsätzliches	58
3.4.2	Threads erzeugen und konfigurieren	59
3.4.3	Synchronisation	60
3.4.4	Message-Handling.....	61
3.4.5	Serielle Schnittstelle	61
3.4.6	TCP/IP	62
3.5	Programmierwerkzeuge	64
3.5.1	Embedded Toolkit	64
3.5.2	Platform Builder	67
3.6	Zusammenfassung	67
4	CAN-Bus	70
4.1	Einleitung.....	70
4.2	Eigenschaften von CAN	71
4.3	CAN OSI-Schichtenmodell	72
4.4	Physikalische Schicht	73
4.5	Datenübertragungsschicht.....	74
4.6	CAN Application Layer (CAL)	82
4.7	CANopen	87
4.8	Zusammenfassung	87
5	Anforderungsanalyse	89
5.1	Einleitung.....	89
5.2	Anforderungen an verteilte MSR-Systeme	91
5.2.1	Liste der allgemeinen Anforderungen	92
5.2.2	Liste der speziellen Anforderungen an das MSR-System	92
5.3	Problemanalyse MSR-System	93
5.4	Interpretation der Anforderungen	99
5.5	Missionstatement	105
5.6	Schnittstellenformate	106
5.6.1	In der Leitebene zum Konfigurieren	109
5.6.2	In der Leitebene zum Visualisieren	110
5.6.3	In der Leitebene zum Steuern und Abfragen.....	112
5.6.4	In der Feldebene	113
5.7	Sequenz-Diagramme (MSC) für das Gesamtsystem	114

5.7.1	Konfigurieren.....	114
5.7.2	Visualisieren	115
5.7.3	Abfragen	117
5.7.4	Regelung	117
5.7.5	Steuern	117
5.7.6	Ausfall einer Komponente	118
5.8	Festlegungen für das MSR-System	119
5.9	Zusammenfassung.....	119
6	Architektur des MSR-Systems in RealTime-UML	122
6.1	Einleitung	122
6.2	Architektur des Beispielsystems	124
6.3	Interne Abläufe des MSR-Systems	125
6.3.1	Konfigurieren des Systems	127
6.3.2	Windows CE-Unterstation einbinden und Konfiguration übernehmen.....	127
6.3.3	Feldkomponente einbinden und Konfiguration setzen	130
6.3.4	Abfragen eines Zustands an einer Feldebene.....	131
6.3.5	Visualisieren von Daten der Feldebene	131
6.3.6	Regeln der Temperatur	134
6.3.7	Senden eines Kommandos an die Feldebene	134
6.3.8	Ausfall einer Komponente der Feldebene.....	138
6.4	Aufbau des MSR-Systems	138
6.4.1	Leitebene.....	140
6.4.2	Unterstationsebene.....	140
6.4.3	Feldebene.....	149
6.5	Zusammenfassung.....	151
7	Implementierung des MSR-Systems.....	154
7.1	Einleitung	154
7.2	Verwendete Hardware.....	157
7.2.1	Verwendete Hardware für die Leitebene	157
7.2.2	Verwendete Hardware für die Unterstation	157
7.2.3	Verwendete Hardware für die Feldkomponenten	158
7.3	Entwicklungsumgebungen	158
7.3.1	Entwicklungsumgebung für Leitreechner.....	158
7.3.2	Entwicklungsumgebung für Unterstation	158
7.3.3	Entwicklungsumgebung für Feldkomponenten	158
7.4	Umsetzung RT-UML / Windows CE (C++).....	159
7.4.1	UML-Prozesse auf Threads abbilden.....	162

7.4.2	Events auf Synchronisations- und Message-Konzept von CE abbilden	164
7.4.3	Software-Umsetzung der Objektklasse des Regelungs-Prozesses	167
7.4.4	Software-Umsetzung der Zustandsmaschine des Regelungs-Prozesses	168
7.4.5	Socket-Programmierung	169
7.4.6	Umsetzung des Konfigurations-Prozesses	172
7.4.7	Umsetzung der Ansteuerung des CAN-Busses	175
7.4.8	Umsetzung des Timers auf dem Mikrocontroller der Feldkomponente	179
7.5	Zusammenfassung	181
8	Zusammenfassung und Ausblick	184
8.1	Zusammenfassung	184
8.2	Ausblick	187
9	Anhang	189
9.1	Glossar	189
9.2	Literatur	194
9.3	Inhalt der beiliegenden CD-ROM	196
	Sachverzeichnis	197

Danksagung

Unser Buch verdankt seine Entstehung vielen Helfern, die uns im Hintergrund unterstützt haben. Martin Lehnhoff hat viele Teile des MSR-Systems implementiert. Olaf Bigalke hat wesentliche Ideen zum CAN-Kapitel beigesteuert. Herr Stöhr und Herr Sommer haben uns bestärkt und ermutigt, das Buch in Angriff zu nehmen. Unsere Frauen haben viel Geduld bewiesen. Ihnen allen sei an dieser Stelle gedankt.