

Holger Bönicke

**Flexible und plattformunabhängige
Entwicklung mikrocontrollerbasierter
mechatronischer Systeme
für Nutzer ohne Vorwissen**



Universitätsverlag Ilmenau
2013

Inhaltsverzeichnis

Abbildungsverzeichnis	IX
Tabellenverzeichnis	XV
Abkürzungen	XVII
Kurzfassung	XIX
Abstract	XXI
Vorwort	XXIII
1 Einleitung	1
1.1 Motivation.....	1
1.2 Zielstellung	3
1.3 Aufbau der Arbeit und Neuartigkeit.....	5
2 Grundlagen – Stand der Technik	11
2.1 Mikrocontrollerbasierte mechatronische Systeme	11
2.1.1 Mechatronische Systeme - Begriffsbestimmung	12
2.1.2 Mikrocontrollerbasierte mechatronische Systeme.....	15
2.1.3 Domänenspezifisches Expertenwissen im Bereich der mikrocontrollerbasierten mechatronischen Systeme	19
2.2 Programmieren von Mikrocontrollern.....	21
2.2.1 Plattformunabhängigkeit mikrocontrollerbasierter mechatronischer Software.....	23
2.2.2 Modellbasierte Softwareentwicklung	25
2.2.3 Graphische Softwaremodellierung und Programmierung mikrocontrollerbasierter mechatronischer Systeme.....	27

2.3	Grundlagen zur Mensch-Maschine-Schnittstelle	32
2.3.1	Mensch-Maschine-Schnittstelle.....	32
2.3.2	Ergonomische Anforderungen.....	34
2.3.3	Der Nutzer im Zentrum der Entwicklung.....	35
2.3.4	Ziele der Usability-Evaluierung	36
2.3.5	Methoden der Usability-Evaluierung	38
2.4	Existierende Entwicklungssysteme für mechatronische Systeme	44
2.4.1	Herstellerspezifische textuelle Entwicklungsumgebungen	45
2.4.2	Graphische plattformunabhängige Entwicklung	47
2.4.3	Modulare mechatronische Hardware mit graphischer Ablaufprogrammierung	52
2.4.4	SPS-Programmierung nach IEC 61131-3.....	57
2.4.5	Modulare Elektronikbausteine mit graphischer Ablaufprogrammierung	60
2.4.6	Zusammenfassung	61
2.5	Defizite bestehender Systeme	62
2.5.1	Flexibilität der Hardware vs. Expertenwissen.....	63
2.5.2	Flexibilität der Software vs. Expertenwissen	65
2.5.3	Identifizierte Schwachstellen.....	67
2.6	Zusammenfassung	69
3	EasyKit und EasyKit Starter	71
3.1	Entwicklung mit EasyKit	73
3.1.1	EasyKit Elektronikmodule.....	74
3.1.2	EasyLab Entwicklungsumgebung	77
3.1.3	Fazit.....	82
3.2	EasyKit Starter.....	83
3.2.1	EasyKit Starterboard.....	84
3.2.2	EasyKit Applikationsboard.....	85
3.2.3	Vereinfachung von EasyLab für den EasyKit Starter	86
3.2.4	Vor der Nutzung des EasyKit Starters.....	87

3.2.5 Das EasyKit Gesamtkonzept.....	89
3.3 Nutzertests	90
3.3.1 Expertentests	90
3.3.2 Tests mit Lehrern	91
3.3.3 Nutzertests mit Schülern	94
3.4 Zusammenfassung	100
4 Erweiterte Entwicklungskonzepte.....	103
4.1 Drei-Ebenen-Konzept der Softwaremodellierung.....	103
4.2 Modellierung ereignisgesteuerter Abläufe	114
4.3 Integrierte Entwicklung von Hard- und Software	120
4.3.1 Modellierung der Entwicklungsplattform.....	120
4.3.2 Inbetriebnahme und Programmierung der informationstechnischen Plattform	122
4.3.3 Modellierung der Sensoren und Aktoren.....	123
4.3.3.1 Testaufbauten aus dem Hobby- und Freizeitbereich	123
4.3.3.2 Akademische und industrielle Testaufbauten.....	127
4.3.3.3 Aktoren und Sensoren der Beispielanwendungen	132
4.3.4 Zusammenhänge zwischen Hardware und Software.....	135
4.3.5 Darstellungsmöglichkeiten der Hardware.....	136
4.4 Signale und Datentypen.....	138
4.4.1 Datentypen	139
4.4.2 Signalarten und ihre Datentypen.....	140
4.5 Zusammenfassung	142
5 Implementierung und Nutzertests.....	145
5.1 Implementierung in einer Testumgebung.....	145
5.1.1 Auswahl der Hardwareplattform.....	146
5.1.2 Hauptarbeitsoberfläche	148
5.1.3 Modellierung der Sensoren und Aktoren.....	153
5.1.4 Funktion der Schnittstellenebene.....	154
5.1.5 Behandlung von Datentypen.....	158

5.1.6	Parametrierung der Funktionsbausteine des Datenflussplans ...	160
5.1.7	Implementierung des Drei-Ebenen-Konzepts	161
5.1.8	Vorgehen bei der Entwicklung eines mikrocontrollerbasierten mechatronischen Systems mit Hilfe der Testumgebung	163
5.2	Nutzertests	166
5.2.1	Auswahl der Testnutzer	167
5.2.2	Ablauf der Nutzertests	168
5.2.3	Ergebnisse der Nutzertests.....	174
5.3	Zusammenfassung	181
6	Zusammenfassung und Ausblick.....	183
6.1	Zusammenfassung	183
6.2	Ausblick.....	186
	Literaturverzeichnis	191