

2 Inhalt

1	Vorwort	5
3	Einleitung	13
3.1	SIMATIC IOT2000	13
3.2	SIMATIC IOT2000 und Arduino	15
3.3	SIMATIC IOT2000 mit Eclipse	16
3.4	SIMATIC IOT2000 und dieses Buch	17
4	Inbetriebnahme der SIMATIC IOT2000	19
4.1	Die ySD-Karte und das Image	20
4.2	Der Test mit der ySD-Karte	23
4.2.1	IP-Adresse am PC einstellen (Windows 7 Prof.)	24
4.2.2	Das Tool PuTTY anwenden	25
5	Das erste Programm mit der Arduino IDE	28
5.1	Der Serial-Treiber für Windows 7	28
5.1.1	Treiber laden und Einstellungen am PC ändern	29
5.2	Das Basis-Programm Arduino-Start-Blink	32
6	Das erste Programm mit Eclipse	35
6.1	Das Blink-Beispiel in C++	37
6.1.1	Programmier-Schnittstelle zum IOT2000 einstellen	39
6.1.2	Neues Projekt-Verzeichnis anlegen	43
6.1.3	Prozess Monitor	45
6.1.4	Das Programm in die IOT2000 laden	46
6.1.4.1	Zur Kontrolle	48
6.2	Autostart nach einem Stromausfall	49
7	Ardu_IOsim mit JavaFX	53
7.1	Installation Java JDK (NetBeans)	53
7.2	Installation und Inbetriebnahme NetBeans	54
7.3	Installation <i>JavaFX Scene Builder</i>	55
7.4	Installation Package <i>gnu.io</i> für <i>RXTX</i>	57
7.5	Der Entwurf von <i>Ardu_IOsim</i>	59
7.5.1.1	Datenaustausch zwischen <i>Arduino/IOT2020</i> und <i>Ardu_IOsim</i>	60
7.5.2	Das Programm mit <i>JavaFX</i> und <i>Scene Builder</i>	63
7.5.2.1	Eine kleine Einführung	63
7.5.3	Das Main-Programm	66
7.5.3.1	Die Software-Struktur <i>User2UI</i> und <i>UI2User</i>	67
7.5.3.1.1	ConnectUI2User (<i>UI2User</i>)	70

7.5.3.1.2	<i>PollingSerialInterface (UI2User)</i>	71
7.5.3.1.3	<i>ReadSerialInterface (UI2User)</i>	71
7.5.3.1.4	<i>SetAktuelleSchnittstelle (UI2User)</i>	72
7.5.3.1.5	<i>CHK2UI (User2UI)</i>	72
7.5.3.1.6	<i>SetUI_Instance (User2UI)</i>	72
7.5.3.1.7	<i>AnalogInputFromUI2Array (User2UI)</i>	73
7.5.3.1.8	<i>DigitalInputFromUI2Array (User2UI)</i>	73
7.5.3.1.9	<i>Portregister2UI (User2UI)</i>	74
7.5.3.1.10	<i>PutText2COMinterface (User2UI)</i>	75
7.5.3.1.11	<i>PutText2InfoLabel (User2UI)</i>	76
7.5.3.1.12	<i>RegisterInfo2UI (User2UI)</i>	76
7.5.4	Die serielle Schnittstelle <i>COMport</i>	76
7.5.4.1	<i>OpenSerialPort</i>	77
7.5.4.2	<i>GetAnzSerialPort</i>	78
7.5.4.3	<i>GetLenInputStream</i>	78
7.5.4.4	<i>ReadDaten</i>	79
7.5.4.5	<i>CloseSerialPort</i>	79
7.5.4.6	<i>GetActualPorts</i>	80
7.5.4.7	<i>SetSerialParameter</i>	81
7.5.5	Die Interrupt-Service-Routine	81
7.5.5.1	Die Service-Routine <i>ComServiceRead</i>	82
7.5.5.2	Die Service-Routine <i>ComService</i>	84
8	Der Arduino-Sketch SIMATIC IOT2000 und UNO für <i>Ardu_IOSim</i>	88
8.1	Der Sketsch <i>Ardu_IOSim_IOT2000</i>	92
8.1.1	Das main-Programm.....	92
8.2	Die Klasse <i>Ardu_IOSim</i> im IOT2000-Board	94
8.2.1	Die Methode <i>SerialEvent</i>	95
8.2.2	Die Methode <i>SetInputFromPC</i>	97
8.2.3	Die Methode <i>DigitalRead</i>	99
8.2.4	Die Methode <i>AnalogRead</i>	100
8.2.5	Die Methode <i>PinMode</i>	100

8.3	Test der Simulation mit der IOT2000 und dem <i>Java</i> -Programm	101
8.3.1	Analog-Eingänge testen.....	103
8.3.2	Test der digitalen Signale.....	103
9	Das Konzept einer Klein-SPS für die SIMATIC IOT2000	105
9.1	Zielsetzung.....	105
9.2	Die PLC und der Vergleich C++ mit FUP.....	107
9.2.1	Zuweisung, UND-Block und ODER-Block.....	108
9.2.1.1	XOR-Block (<i>Exklusiv Oderverknüpfung</i>)	112
9.2.2	Trigger (Flanken-Merker).....	113
9.2.3	Timer	114
9.2.3.1	Timer Folgesteuerung	115
9.2.3.2	Takt-Merker.....	116
9.2.4	Zähler.....	118
9.2.5	Analogwerte lesen und skalieren.....	119
9.2.6	Analogwerte schreiben.....	121
9.2.6.1	Analogausgabe an einen Pin	122
9.3	Das Absturz-Problem mit C++-Klassen bei der IOT2000	123
10	PLC-Programm in C++ mit der Arduino IDE.....	125
10.1	Die Klassendefinition.....	125
10.1.1	Die Klassendefinition <i>PLC_IOT.h</i>	125
10.1.2	Datentypen zur <i>PLC_IOT</i>	127
10.1.3	Die Klasse <i>PLC_IOT-Teil-1</i>	128
10.1.4	Die Klasse <i>PLC_IOT-Teil-2</i>	129
10.2	Die Methoden der Klasse <i>PLC_IOT.cpp</i>	129
10.2.1	Konstruktor und Destruktor.....	130
10.2.2	<i>InitPLC</i>	130
10.2.3	Read- und Write für die digitalen Pins.....	131
10.2.4	<i>Trigger</i>	131
10.2.5	<i>Timer</i>	132
10.2.6	<i>Counter</i>	133
10.2.7	<i>Scale</i>	134
10.2.8	<i>UnScale</i>	134
11	Eclipse, Arduino und Linux ?.....	135
11.1	Arduino und Eclipse haben eine gemeinsame Schnittstelle.....	136
11.1.1	Der IO-Test Eclipse und Arduino.....	136
11.2	Die Schnittstelle zu Linux mit Arduino	140
11.2.1	Von der ySD-Karte lesen.....	140
11.2.2	Auf die ySD-Karte schreiben	142
11.2.3	System-Befehle (Linux-Systemzeit).....	144
11.2.4	Chip-Temperatur lesen.....	145
12	Multi-Kommunikation mit der SIMATIC IOT2000.....	147

12.1	Fingierte Aufgabenstellung.....	147
12.1.1	IOT2000 in das Netzwerk einbinden.....	149
12.2	Die Programmstruktur.....	151
12.3	Projekt zur PLC S7-1500® mit dem TIA Portal erstellen.....	152
12.3.1	Der <i>DB1</i>	152
12.3.2	Die Sinusfunktion im <i>FB1</i>	154
12.3.3	Das <i>main</i> -Programm.....	154
12.4	Projekt WinCC mit dem TIA Portal erstellen.....	155
12.4.1	Die Zeitscheibe als Synchronisation.....	157
12.4.2	Der Skript <i>UserAction()</i>	159
12.4.3	Der Skript <i>WriteEnabled</i>	162
12.4.4	Der Skript <i>WriteDatas()</i>	162
12.5	Der Webserver mit dem CGI.....	164
12.5.1	Das Programm zur Webseite.....	164
12.5.1.1	<i>plc.PutMessage2Client</i>	166
12.5.1.2	<i>PrintHTML2WebSite</i>	167
12.6	Die Verbindung zur S7-1500® in C++.....	169
12.6.1	<i>plc.GetFloatFromDB</i>	170
12.6.1.1	<i>GetDINT</i>	174
12.6.2	<i>ReadFromClient</i>	175
12.7	Die C/C++-Schnittstelle zu WinCC und UDP zur IOT2000.....	176
12.7.1	Inbetriebnahme.....	176
12.7.2	Das C/C++-Programm.....	178
12.7.2.1	Mit UDP an die IOT2000 senden.....	178
12.8	Das Arduino-Programm UDP-Server für WinCC.....	182
12.8.1	<i>PrintHTML2Website</i>	183
12.8.2	Der Test mit der Website.....	184
13	Verschiedenes.....	186
13.1	Kleine Einführung in das Dateiformat Linux.....	186
13.1.1	<i>top</i>	186
13.1.2	<i>ps - el</i>	187
13.1.3	<i>kill</i>	188
13.1.4	<i>ls</i> und <i>cd</i>	188
13.1.5	<i>mkdir</i>	188
13.1.6	<i>rmdir</i> und <i>rm</i>	189
13.1.7	<i>clear</i>	189
13.1.8	<i>pwd</i>	189
13.1.9	Dateien anschauen.....	189
13.1.10	Texteditor <i>nano</i>	191
13.1.11	<i>chmod</i>	192
13.1.12	<i>Reboot</i>	193