

ARM Cortex-M3 Mikrocontroller

Einstieg und Praxis

Inhaltsverzeichnis

	Einleitung	15
Teil I	Grundlagen	25
I	Orientierung	27
I.1	Kommerzielle IDEs	27
I.1.1	Keil μ Vision	28
I.1.2	IAR Workbench	28
I.1.3	Sourcery Codebench.	28
I.1.4	Atollic TrueSTUDIO	29
I.1.5	CrossWorks for ARM	29
I.2	Herstellergebundene IDEs	30
I.2.1	ATMEL Studio 6.	30
I.2.2	Texas Instruments StellarisWare	30
I.2.3	STMicroelectronics STVD	31
I.3	Freie IDEs	31
I.3.1	CooCox CoIDE	31
I.3.2	NetBeans for C Developers	31
I.3.3	Code::Blocks	32
I.3.4	emIDE	32
I.3.5	Eclipse für C/C++-Entwickler	32
I.4	Vorbereitende Arbeiten.	33
I.4.1	Hardware	34
I.4.2	Software.	37
2	ARM und CMSIS	53
2.1	Einige Hintergrundinformationen.	53
2.1.1	Die Firma ARM Holdings PLC	53
2.1.2	Das Geschäftsmodell	53
2.2	CMSIS	54
2.2.1	Implementierungen	54

2.3	Erzeugung der Bibliotheken	56
2.3.1	libboard: Die Bibliothek für das Entwicklungsboard	57
2.3.2	libchip: Die Bibliothek für den Mikrocontroller-Chip	62
2.4	Weitere Software installieren	64
2.4.1	SAM-BA	64
2.4.2	SEGGER J-Link GDB Server via JTAG	67
3	Das erste Eclipse-Projekt	71
3.1	Erstellen einer Projektschablone	71
3.1.1	Workspace einrichten.	72
3.1.2	Anlegen eines neuen Projekts	73
3.1.3	Projektspezifische Einstellungen.	75
3.1.4	C/C++ Build: Settings	78
3.2	Weitere erforderliche Dateien	94
3.2.1	board_cstartup_gnu.c und syscalls.c	94
3.2.2	Linkerscriptdateien	96
3.3	Konfiguration des Debuggers	104
3.4	Fertigstellen des Templates.	108
3.4.1	Anwendung der Erweiterung.	109
4	Hello World!	111
4.1	Grundlegende Hinweise	111
4.2	Erstellen des Projekts o4_oI_Blinky	112
4.2.1	Importieren der Schablone	114
4.2.2	Der Sourcecode von Blinky	116
4.3	Blinky im Debugger ausführen	126
4.4	Debugging light	132
4.4.1	Was Sie benötigen	132
4.4.2	Konfiguration der Schnittstelle	133
4.4.3	Programm laden und ausführen.	133
4.4.4	Vorteile und Nachteile dieser Methode.	134

Teil II Einfache Grundlagen der Elektronik 137

5	Der ATMEL SAM3S4B	139
5.1	Die ATMEL-SAM3S-Familie	140
5.1.1	Übersicht.	140

5.2	Das Datenblatt DOC 6500	141
5.2.1	Der Aufbau von DOC 6500	142
5.2.2	Mikrocontroller anderer Hersteller	144
5.3	Elektrische Daten des SAM3S4	145
5.3.1	Minimum- und Maximumwerte	146
5.3.2	Elektrische Versorgungsspannungen	147
5.3.3	Gleichstromwerte	149
5.4	System Controller	152
5.5	CHIP_ID	153
5.5.1	Das Projekt 05_01_CHIPID	154
5.5.2	Erläuterungen	158
5.6	Weiterführende Literatur	162
6	Elektronik	165
6.1	Digitale Ausgänge	165
6.1.1	Ports A, B und C im Reset-Zustand	166
6.2	Schalten kleiner Ströme	174
6.2.1	Current Sourcing	174
6.2.2	Current Sinking	176
6.2.3	Dimensionierung bei Current Sourcing und Current Sinking	177
6.2.4	Vor- und Nachteile beider Betriebsarten	180
6.3	Schalten größerer Ströme	181
6.3.1	Bipolare Transistoren	181
6.3.2	Feldeffekt-Transistoren (FETs)	190
6.3.3	Schalten mit Optokopplern	191
6.3.4	Schalten von Leistungstransistoren	192
6.3.5	Schalten induktiver Lasten (Relais, Elektromagnete, Motoren)	192
6.4	Digitale Eingänge	194
6.4.1	Grundlegende Betrachtungen	194
6.4.2	Einfachste Form der Beschaltung	194
6.4.3	Bessere Form der Beschaltung	195
6.4.4	Erfassen größerer Spannungen I	196
6.4.5	Erfassen größerer Spannungen II	198
6.5	Allgemeine Anmerkungen	198

7	Anwendungen	199
7.1	LC-Displays	199
7.1.1	Die Hardware	200
7.1.2	Projekt 07_01_LCD	202
7.2	7-Segment-Anzeigen	223
7.2.1	Kein Datenblatt verfügbar?	225
7.2.2	Eine Möglichkeit der Ansteuerung	226

Teil III Basiskomponenten 237

8	NVIC, PMC, Clock Generator und SUPC	239
8.1	Allgemeines zu Interrupts	239
8.1.1	Asynchrone Ereignisse	240
8.1.2	Der NVIC – Nested Vector Interrupt Controller	240
8.1.3	Zuordnung der Interrupt-Quellen	244
8.1.4	Tail Chaining	246
8.1.5	CMSIS-Funktionen für den NVIC	247
8.1.6	(Kein) Beispiel	249
8.1.7	Software-Interrupts	251
8.1.8	Tipps und Empfehlungen	252
8.2	Der Clock Generator / Taktgenerator	252
8.2.1	Funktionen des Clock Generators	252
8.3	Der PMC – Power Management Controller	256
8.3.1	Aufgaben des PMC	257
8.3.2	Die Taktsignale des PMC	257
8.3.3	Weitere Informationen zum PMC	259
8.3.4	Ausgewählte Register des PMC	260
8.4	Der SUPC – Supply Controller	264
9	Parallel Input/Output Controller	265
9.1	Port-Register und -Betriebsarten	265
9.1.1	PIOA, PIOB und PIOC	266
9.1.2	Die Register von PIOA, PIOB und PIOC	269
9.2	Input-Ports in der Praxis	289
9.2.1	Das Projekt 09_01_INPUT_SAMPLE	290
9.2.2	Das Ergebnis	296

10	Timer und Counter, Teil 1	299
10.1	Real-time Timer RTT	300
10.1.1	Projekt 10_01_RTT	300
10.1.2	Die Register des RTT	313
10.2	RTC – Die Echtzeituhr	314
10.2.1	Das Projekt 10_02_RTC	315
10.2.2	Projekt 10_02_RTC_Advanced	320
10.2.3	Die Register der RTC	339
10.3	Der Watchdog-Timer WDT	344
10.3.1	Projekt 10_03_WDT	345
10.3.2	Register des WDT	352
10.4	Der System-Timer SysTick	353
10.4.1	Grundlegende Funktion	353
10.4.2	Anwendung von SysTick	354
10.4.3	Konfiguration des SysTick	355
10.4.4	Register des System-Timers SysTick	356
10.4.5	SysTick-Interrupt	358
10.5	Abschlussbetrachtung	358
11	Timer und Counter, Teil 2	359
11.1	Timer/Counter, Grundlagen	360
11.1.1	Einsatzgebiete von Timern und Countern	360
11.1.2	Grundlegende Betrachtungen	361
11.1.3	Triggern der Counter	361
11.2	Timer/Counter programmieren	362
11.2.1	PIO-Controller konfigurieren	362
11.2.2	PMC konfigurieren	365
11.2.3	NVIC konfigurieren	365
11.3	Die Register der Timer/Counter	366
11.3.1	TC- und TC-Channel-Register	366
11.4	Projekt 11_01_TIMER_COUNTER	375
11.4.1	global.h	375
11.4.2	tcWave.h und tcWave.c	376
11.4.3	tcCapture.h und tcCapture.c	383
11.4.4	main.c	385

Teil IV Weiterführende Komponenten	389
I2 Peripheral DMA Controller (PDC)	391
I2.1 Prinzipieller Aufbau	391
I2.1.1 Voll-Duplex-fähige Peripherie	392
I2.1.2 Halb-Duplex-fähige Peripherie	392
I2.1.3 Monodirektionale Peripherie	393
I2.1.4 Voll-Duplex- und Halb-Duplex-Kanäle	393
I2.1.5 Monodirektionale Kanäle	393
I2.2 PDC-Register	393
I2.2.1 Receive Pointer Register (PERIPH_RPR)	394
I2.2.2 Receive Counter Register (PERIPH_RCR)	394
I2.2.3 Transmit Pointer Register (PERIPH_TPR)	394
I2.2.4 Transmit Counter Register (P_TCR)	394
I2.2.5 Weitere Receive- und Transmit-Register	394
I2.2.6 Transfer Control Register (PERIPH_PTCR)	395
I2.2.7 Transfer Status Register (PERIPH_PTSR)	395
I2.3 Schlussbetrachtung	395
I3 PWM – Pulsweitenmodulation	397
I3.1 Was ist Pulsweitenmodulation?	397
I3.2 Pulsweitenmodulation – aber wozu?	398
I3.3 Der PWM-C der AT91SAM3S-Familie	398
I3.3.1 Abhängigkeiten des PWM-C	399
I3.3.2 Die CMSIS-Funktion des PWM-C	400
I3.3.3 Zuordnung der PWM-Anschlüsse	409
I3.4 Projekt I3_OI_PWM	410
I3.4.1 board_olimex.h	410
I3.4.2 pulsewidthmod.h	414
I3.4.3 pulsewidthmod.c	414
I3.4.4 terminal.c	416
I3.4.5 main.c	418
I3.5 Drehzahlregelung eines DC-Motors	423
I3.5.1 Dimensionierung der Schaltung	423
I3.5.2 Drehrichtungswechsel	424
I4 Analoge und digitale Größen	425
I4.1 Vereinfachte Grundlagen	425
I4.2 DACC – Digital-to-Analog Converter Controller	426
I4.2.1 DACC-Register	427

14.2.2	CMSIS-Funktionen zum DACC	429
14.2.3	I4_01_DACC_SIGNAL_GENERATOR_WITH_INTERRUPT	430
14.2.4	Hilfsprogramm: I4_02_TABLE_GENERATOR	434
14.3	ACC – Analog Comparator Controller	442
14.3.1	Die Register des ACC	443
14.3.2	CMSIS-Funktionen zum ACC	445
14.3.3	Projekt I4_03_ACC	446
14.4	ADC – Analog-to-Digital Converter	448
14.4.1	Eigenschaften des ADC	448
14.4.2	Die ADC-Register	449
14.4.3	CMSIS-Funktionen des ADC	457
14.4.4	Projekt I4_06_ADC_TS_UND_POTI	458

Teil V Serielle Kommunikation 475

15	Serielle Schnittstellen I	477
15.1	Hardware	478
15.1.1	RS-232 (EIA 232)	479
15.1.2	RS-485	481
15.1.3	TWI (I ² C)	482
15.1.4	Serial Peripheral Interface (SPI)	482
15.1.5	Synchronous Serial Controller (SSC)	482
15.2	Serielle Schnittstellen der AT91SAM3S-Familie	483
15.2.1	Grundlegende Begriffe	483
15.3	Universal Asynchronous Receiver Transceiver (UART)	489
15.3.1	UART-Eigenschaften beim AT91SAM3S	489
15.3.2	UARTs auf dem Olimex SAM3-P256	489
15.3.3	UART-Register	490
15.3.4	RS232_0 und Retargeting	493
15.4	Universal Synchronous Asynchronous Receiver Transceiver (USART)	496
15.4.1	USART-Eigenschaften beim AT91SAM3S	497
15.4.2	USARTs auf dem Olimex SAM3-P256	498
15.4.3	USART-Register	499
15.5	Two-wire Interface (TWI)	507
15.5.1	TWI-Eigenschaften beim AT91SAM3S	509
15.5.2	TWI auf dem Olimex SAM3-P256	509
15.5.3	TWI-Register	509

16	Serielle Schnittstellen II	515
16.1	SD Card (stark vereinfacht)	515
16.1.1	Ausführungsformen und Anschlüsse	516
16.1.2	Versorgung und Stromaufnahme	517
16.1.3	Speicherkapazitäten und Zugriffsraten	517
16.2	SD-Karten im SPI-Modus	518
16.2.1	Grundlagen zum SPI	518
16.2.2	Initialisierung des SPI	518
16.2.3	Lesen und Schreiben von Rohdaten	536
16.3	High Speed MultiMedia Card Interface (HSMCI)	538
16.3.1	Merkmale des HSMCI	538
16.3.2	Informationen zu den Protokollen	539
16.3.3	Anschluss eines SD-Kartenslots	540
16.3.4	Die HSMCI-Register	540
16.3.5	Hinweis zur Nutzung des HSMCI	543
16.4	Synchronous Serial Controller (SSC)	543
16.4.1	Merkmale des SSC	544
16.4.2	Die wichtigsten Register des SSC	545
A	Glossar	547
A.1	Architektur	547
A.2	ARM	547
A.3	ARM-Befehlssatz	547
A.4	Big.LITTLE-Konzept	548
A.5	BSS	548
A.6	CMSIS	548
A.7	Cortex	548
A.8	Debugging	549
A.9	Echtzeit-Betriebssysteme	549
A.10	Embedded Linux	549
A.11	FIFO	550
A.12	Firmware	550
A.13	Heap	550
A.14	JTAG	550
A.15	LIFO	551
A.16	OCD	551
A.17	SAM-BA	551
A.18	Stack	551

A.19	SWD	552
A.20	TDMI	552
A.21	Text-Segment	552
A.22	Thumb-Befehlssatz	553
B	Ressourcen	555
B.1	Hardware	555
B.2	Software	556
C	Literatur	559
C.1	Literatur (Buchversion)	559
C.2	Literatur (Online-Version)	559
C.3	Weitere allgemeine Quellen	561
D	Erfahrungen	563
D.1	Wechsel der Toolchain	563
D.2	GNU Tools for ARM Embedded Processors	563
D.3	Nochmals: Verwendung der Nano-Libs	564
D.4	Updates von Eclipse und dem CDT	564
D.5	Andere Probleme mit Eclipse und dem CDT	564
D.6	Debugger	565
D.7	Versionsverwaltung	565
	Stichwortverzeichnis	566

