



Software Manual

MON85XX

MON85XX.SWM.106

	Datum	Name	Benennung	Software Manual	
Erstellt	04.06.2007	JGE	Projekt	MON85XX	
Geprüft	05.06.2007	DAL	Dokumenten-Nr.	MON85XX.SWM.106	
Firma			Freigabestatus	freigegeben	
			Kunde	TQC	Blatt 1 von 60
Datei	MON85XX.SWM.106.doc			Vorlagenrevision: 001	

© TQ-Systems GmbH. Alle Rechte vorbehalten

Mit der Bereitstellung der in diesem Dokument enthaltenen Informationen ist keinerlei Übertragung von Lizenz-, Patent-, Marken-, Urheber- oder sonstigen gewerblichen Schutzrechten verbunden. Sämtliche in diesem Dokument enthaltenen Informationen sind als Betriebs- bzw. Geschäftsgeheimnisse streng vertraulich zu behandeln. Eine vollständige oder teilweise Weitergabe der in diesem Dokument enthaltenen Informationen oder Darstellungen und Kenntnisse an Dritte bedarf der vorherigen schriftlichen Zustimmung der TQ-Systems GmbH.

Inhaltsverzeichnis

1. ZIELBESTIMMUNGEN	7
1.1 ANWENDUNGSBEREICH	7
1.2 ZIELGRUPPEN	7
1.3 BETRIEBSBEDINGUNGEN	7
1.4 MUSSKRITERIEN	8
1.5 WUNSCHKRITERIEN	8
1.6 ABGRENZUNGSKRITERIEN	9
2. PRODUKTUMGEBUNG	10
2.1 SOFTWARE	10
2.2 HARDWARE	10
2.2.1 CPU	10
2.2.2 NOR-Flash	10
2.2.3 NAND-Flash	10
2.2.4 SDRAM	10
2.2.5 EEPROM (Daten)	10
2.2.6 EEPROM (Konfiguration)	11
2.2.7 RTC	11
2.2.8 Temperatursensor	11
2.2.9 CAN	11
2.2.10 Andere Hardware	11
2.3 ORGWARE	11
2.4 PRODUKTSCHNITTSTELLEN	11
3. PRODUKTFUNKTIONEN	12
3.1 MODULEDEKTIERUNG	12
3.1.1 Hardware Information Block	12
3.1.2 Configuration Information Block	13
3.2 POWER-ON SELF TEST (POST)	14
3.2.1 POST 1A	14
3.2.2 POST 1B	16
3.2.3 POST 2	16
3.2.4 Signalisierung von POST-Ergebnissen	16
3.3 INITIALISIERUNG	17
3.3.1 Clocks	17
3.3.2 MPC-Typ	17
3.3.3 Watchdog	17
3.3.4 Memory Management Unit (MMU)	17
3.3.5 Memory-Mapped Register	17
3.3.6 Monitor Bootpage	18
3.3.7 Local Memory Map	18
3.3.8 NOR-Flash	18
3.3.9 NAND-Flash	18
3.3.10 SDRAM	20
3.3.11 CAN-Controller	21
3.3.12 Monitor Speicher-Mapping:	22
3.3.13 Caches und L2-SRAM	23
3.3.14 MON85XX (Systemsoftware)	23
3.3.15 Serielle Schnittstelle	24
3.3.16 I2C-Controller	25
3.3.17 CAN-Controller	25
3.3.18 Timer	25
3.3.19 Hardware-Debugger	25
3.4 BETRIEBSMODI	26
3.4.1 Initialer Betriebsmodus	26
3.4.2 Interaktiver Betriebsmodus	27
3.4.2.1 Serielle Schnittstelle	27
3.4.2.2 Kommandointerpreter	28

3.4.2.3	NOR-Flash Manipulationsfunktionen.....	28
3.4.2.4	NAND-Flash Manipulationsfunktionen.....	29
3.4.2.5	EEPROM Manipulationsfunktionen (I2C).....	30
3.4.2.6	Temperatursensorfunktionen (I2C).....	30
3.4.2.7	RTC Manipulationsfunktionen (I2C).....	30
4.	PRODUKTDATEN	32
4.1	HARDWARE INFORMATION BLOCK (HWIB).....	32
4.2	CONFIGURATION INFORMATION BLOCK (CIB).....	32
4.3	APPLIKATIONSVORSPANN.....	33
5.	PRODUKTLLEISTUNGEN	35
6.	BENUTZUNGSOBERFLÄCHE	36
6.1	KOMMANDOEINGABE	36
6.2	ANWENDERKOMMANDOS	37
6.2.1	COPY – Kopieren von Speicherbereichen	37
6.2.2	CRC – CRC berechnen.....	38
6.2.3	DEF – Benutzerdefinierte Register.....	38
6.2.4	ECHO – Ein-/Ausschalten des Echo	39
6.2.5	ERASE – Löschen von NOR-Flash-Sektoren	39
6.2.6	EREADE – Daten aus einem EEPROM auslesen.....	40
6.2.7	EWRITE – Daten in ein modulinternes EEPROM schreiben.....	41
6.2.8	GO – Starten von Programmen.....	42
6.2.9	HELP – Hilfe-Funktion zu den Anwenderkommandos	43
6.2.10	I2CDEV – Anzeigen der modulinternen I2C-Geräte	43
6.2.11	LOAD – Herunterladen von S-Record-Daten auf das TQ-Modul.....	44
6.2.12	MEMORY – Anzeigen des aktuellen Speicher-Mappings.....	45
6.2.13	NCLEARBBT – Löschen der Bad-Block-Tabelle im RAM mit Update im NAND-Flash.....	46
6.2.14	NDUMP – Anzeige des Inhalts einer NAND-Flash-Seite	46
6.2.15	NERASE – Löschen von NAND-Flash Blocks	46
6.2.16	NINFO – Informationen über das NAND-Flash anzeigen	47
6.2.17	NREAD – Daten kopieren vom NAND-Flash ins RAM.....	48
6.2.18	NWRITE – Daten kopieren vom RAM oder NOR-Flash ins NAND-Flash	48
6.2.19	POST – Abfragen / Löschen der POSTxEN-Flags.....	49
6.2.20	PROTECT – Ändern des Schreibschutzes von TQ-Informationen	50
6.2.21	READ – Speicher auslesen.....	51
6.2.22	REG – Zugriff auf Configuration, Control und Status Register	52
6.2.23	RTC – Einstellen und Ausgeben der aktuellen Uhrzeit und des Datums	53
6.2.24	SAVE – Ausgeben von S-Record-Files	53
6.2.25	SETCI – Speichern des Configuration Information Blocks (CIB).....	54
6.2.26	SETHWI – Speichern des Hardware Information Blocks (HWIB).....	55
6.2.27	TEMPSENS – Ausgabe der aktuellen Temperatur des Moduls	56
6.2.28	WRITE – Speicher beschreiben	56
6.3	MELDUNGEN DES MONITORS.....	57

Tabellenverzeichnis

Tabelle 1: Local Memory Map	18
Tabelle 2: Speicher-Mapping	22
Tabelle 3: Monitor Speicherbedarf im NOR-Flash	22
Tabelle 4: Monitor Speicherbedarf im SDRAM	22
Tabelle 5: Monitor Einsprungadressen	24
Tabelle 6: Parameterübergabe an die Applikation	27
Tabelle 7: Parameter des CIB	32
Tabelle 8: Inhalt des Applikationsvorspanns	33
Tabelle 9: Lage Software Schalter	34
Tabelle 10: Funktionsübersicht Software Schalter	34
Tabelle 11: Beschreibung der Kommandosyntax	36
Tabelle 12: POST-Meldungen	59
Tabelle 13: Rück-/Fehlermeldungen des Monitors nach Kommandoeingabe	60
Tabelle 14: Fehlermeldungen des Monitors beim Booten	60

Abbildungsverzeichnis

Abbildung 1: Schichtenarchitektur	7
Abbildung 2: BNF des HWIB	13
Abbildung 3: BNF des CIB	14
Abbildung 4: BBT-Initialisierung	20
Abbildung 5: Beispiel eines Speicherauszugs des HWIB für ein TQM8560-AA	32
Abbildung 6: Beispiel eines Speicherauszugs des CIB für ein TQM8560-AA	33

Literaturverzeichnis

- [1] MPC8560 PowerQUICC III™ Integrated Communications Processor Reference Manual
 MPC8540 PowerQUICC III™ Integrated Host Processor Reference Manual
 MPC8555E PowerQUICC III™ Integrated Communications Processor Reference Manual
 MPC8548E PowerQUICC III™ Integrated Host Processor Family Reference Manual
- [2] TQM8560.Varianten.0312A.pdf
 TQM8555.Varianten.0112.pdf
 TQM8548.Varianten.0103.pdf
- [3] Application Note "Bit Field Programming for Flash Products Based on 230-nm and 200-nm MirrorBit™ Technology"
http://www.spansion.com/application_notes/bitfieldprogMirrorbit_an_a3_e.pdf

Abkürzungen

BBT	Bad Block Table
BNF	Backus-Naur-Form
CCB	Core Complex Bus
CFI	Common Flash Memory Interface
CIB	Configuration Information Block
GiB	Gibibyte = 2^{30} Byte = 1.073.741.824 Byte
GOT	Global Offset Table
HWIB	Hardware Information Block
KiB	Kibibyte = 2^{10} Byte = 1024 Byte
MiB	Mebibyte = 2^{20} Byte = 1.048.576 Byte
POST	Power-On Self Test
SLC	Single Level Cell

MON85XX.SWM.106.doc

Änderungsindex:

Rev.	Datum	Pos.	Name	Modifikation
100	27.08.04		JGE	Bereitstellung rev009 als rev100
101	07.03.05		JGE	Anpassung Spezifikation: Blink-Reset und Applikationsreboot. Erweiterung der maximalen CPU-Frequenz auf 1 GHz.
	08.03.05		JGE	Anpassung HWIB-BNF und HWIB-/CIB-Speicherauszug. Ergänzung des SETPORC-Kommandos um Parameter RECOVER. Erläuterungen zum abgeschalteten POST 1A hinzugefügt.
	11.03.05		JGE	Erläuterungen zu Boot-Typen und Aufteilung der Code-Sektionen ergänzt.
	11.03.05		JGE	Interne Prüfung, Korrekturen
	17.03.05		JGE	Ergänzungen zu MMU-Initialisierung und Caches
	22.03.05		JGE	Freigabe durch ANW per Email vom 21.03.05
104	04.04.06		JGE	Wechsel auf neue Dokumentenvorlage. Erweiterung um MPC8541E und MPC8555E bis Kapitel 3. Anpassung HWIB BNF an neuen Variantenschlüssel: Wegfall des HIP, neu: E-Option für SEC-Engine
	05.04.06		JGE	Wegfall des Kommandos SETPORC gemäß Kundenwunsch. Erläuterung zum POST-Blinkreset Fall hinzugefügt. Maximale Flash-Größe auf 1 GiB gesetzt.
	07.04.06		JGE	Erläuterung zum Wechsel in den interaktiven Modus bei POST-Blinkreset korrigiert.

©2007 by TQ-Components GmbH

	10.04.06		JGE	Kapitel markiert, die nur zu internen Verwendung bestimmt sind.
	21.04.06		JGE	Ergänzung um RAM-Zugriff für bis zu 4 Chip-selects. Zusätzliche Erläuterungen zur RAM-Detektierung.
	26.04.06		JGE	Interne Prüfung, Korrekturen
	11.07.06		JGE	DS1137 geändert auf DS1337U
	04.08.06		JGE	Überprüfung der Z-Option auf kleiner oder gleich 39 wegen Erweiterung des Variantenschlüssels um den Wert 32 für SPANSION Flashes A/N-Typen. Referenzen auf Variantenschlüsseldokumente auf aktuellen Stand angepasst.
	08.08.06		JGE	Freigabe durch ROE.
105	12.05.06		JGE	Erweiterungen für TQM8548: DDR2-SDRAM, NAND-Flash
	22.05.06		JGE	Erweiterung der Monitor- und POST-Fehlermeldungen
	31.05.06		JGE	Abfrage der E-Option des MPC zu POST1A hinzugefügt.
	06.07.06		JGE	Unterscheidung zwischen DDR1 und DDR2 geändert auf Abfrage des PORDEVSR Registers. Ein CAN UPM-Timing für alle Frequenzen. Neun NAND-Flash UPM-Timings.
	11.07.06		JGE	Maximalwert für Z-Option im Variantenschlüssel auf 39 gesetzt.
	25.07.06		JGE	Hinweis in Kapitel 6.1 geändert: Beim Zugriff auf nicht verfügbare Speicherbereiche blockiert der Prozessor und damit der Monitor. Verweis auf TQM85xx-Variantenschlüssel ersetzt durch Verweise auf Variantenschlüsseldokumente für jede Modulfamilie. Beim MPC8548 und seinen Derivaten wird das L2-SRAM als 512 KiB großes Array konfiguriert. Beim MPC8548 und seinen Derivaten sind derzeit nur 1- und 2-lagige SDRAM Stacks möglich.
	31.07.06		JGE	Einheiten kB, MB und GB gegen Binärpräfixe gemäß IEC 60027-2 ersetzt.
	21.08.06		JGE	Teilverhältnis CCB:LB = 2:1 nach Änderung der UPM-Tabelle für CAN wieder erlaubt.
	05.09.06		JGE	Write Buffer Modus für NOR-Flashes hinzugefügt.
	11.09.06		JGE	Abkürzungstabelle eingefügt. Schriftart in Abbildungen zur HWIB und CIB BNF geändert. MPC8548-Abfrage in POST eingefügt. Erklärung zur UPMB-Initialisierung bzgl. Frequenz und Timingwahl verbessert. Programmablaufplan zur NAND-Flash BBT-Initialisierung zur besseren Verständlichkeit hinzugefügt. NAND-Flash Seitengröße in verschiedenen Kapiteln hinzugefügt und zwischen 1x8 Bit und 2x8 Bit Modus unterschieden. Erklärung, warum bei einem nicht erfolgreichen Programmiervorgang das Kopieren ins NAND-Flash abgebrochen wird, hinzugefügt. Fehlermeldungen der NAND-Flash Funktionen hinzugefügt.
	28.09.06		JGE	„Geprüft“-Status eingetragen.
	29.09.06		JGE	Freigabe durch ROE.
106	04.06.07		JGE	Abschaltung der Port- und Baudratendetektierung hinzugefügt (Tabelle 9 und Tabelle 10).
	06.06.07		JGE	Rechtschreibfehler korrigiert.

1. Zielbestimmungen

Dieses Software-Produkt „MON85XX“, nachfolgend auch „Monitor“ genannt, stellt die Firmware dar, die TQ-Components mit ihren TQM85XX-Modulen ausliefert.

1.1 Anwendungsbereich

Der MON85XX ist eine Modulinitialisierungs- und Kommunikations-Software für alle Modulvarianten der TQM85XX Serie in Verbindung mit dem Starterkit STK85XX.

Der Monitor ist für die folgenden Anwendungsbereiche entwickelt worden:

- Unterstützung der Inbetriebnahme von Basisboards, in denen das Modul zum Einsatz kommt (interaktive Kommunikation u.a. für Registermanipulation etc.).
- Unterstützung von Software-Entwicklungen für das Modul (interaktive Kommunikation für Up-/ Downloads, Speicherzugriff etc.).
- Initialisierung des Moduls mit wahlweise automatischem Start von Kundenapplikationen nach einem Reset bzw. Power-On.

Die Kommunikation zwischen dem Benutzer und der Hardware realisiert der Monitor über eine serielle Schnittstelle (Abbildung 1). Diese verbindet das Starterkit, auf dem sich das Modul befindet, mit einem beliebigen seriellen Kommunikationspartner (z.B. PC mit Terminalprogramm).

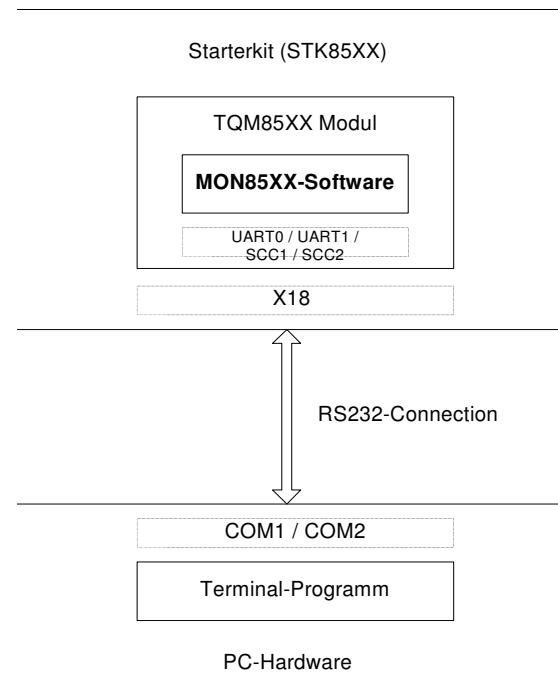


Abbildung 1: Schichtenarchitektur

1.2 Zielgruppen

Der Monitor wurde für zwei Zielgruppen entwickelt. Bei der ersten handelt es sich um Hardware-Entwickler, welche den Monitor zur Unterstützung der Evaluierung und Inbetriebnahme einer Baugruppe benutzen. Die zweite besteht aus Software-Entwicklern, die Applikationen für den MPC85XX erstellen und den Monitor benutzen, um ihre Software in das Modul zu übertragen.

Neben diesen Zielgruppen ist der Monitor auch für weitere Zielgruppen wie Service-Techniker grundsätzlich geeignet, z.B. als Werkzeug für Software-Updates. Hierbei ist zu beachten, dass der Monitor technisch versierte Anwender voraussetzt. So sollten dem Anwender die englischen Fachbegriffe u.a. bzgl. der Prozessorarchitektur geläufig sein und er detaillierte Kenntnisse über die Benutzung von seriellen Schnittstellen (Terminalprogramme, Baudraten, Flusskontrolle etc.) besitzen.

1.3 Betriebsbedingungen

Die physikalische Betriebsumgebung und die täglichen Betriebszeiten sind nicht eindeutig festlegbar, da sie durch die Verwendung des Moduls beim Kunden bestimmt werden. Typische Betriebsumgebungen sind eingebettete Systeme, z. B. Kommunikationssysteme oder Geräte- / Automatensteuerungen.

Es werden zwei mögliche Betriebsarten unterschieden:

- **Initialer Betriebsmodus:**

Initialisierung der Hardware, dann automatischer Start einer Applikation

→ Betriebszeit im Bereich weniger Sekunden jeweils nach dem Einschalten der Hardware

- **Interaktiver Betriebsmodus:**

Initialisierung der Hardware, dann Bearbeiten von Kommandos über die seriellen Schnittstellen

→ Betriebszeit im Bereich von mehreren Stunden pro Tag

1.4 Musskriterien

- Initialisierung des Moduls aus dem Reset-Zustand
- Unterstützte Taktfrequenzen (CPU / CCB / Local Bus):
 - 800 / 267 / 33 MHz
 - 800 / 267 / 67 MHz
 - 833 / 333 / 42 MHz
 - 833 / 333 / 83 MHz
 - 1000 / 333 / 83 MHz
 - 1200 / 400 / 50 MHz
 - 1333 / 533 / 67 MHz
- Kommunikation über eine serielle Schnittstelle (SCC1/2 oder UART0/1)
- Zugriff auf das NOR-Flash (Lesen, Schreiben, Löschen, bis zu 2 Chip-selects)
- Zugriff auf das NAND-Flash (Lesen, Schreiben, Löschen)
- Zugriff auf das SDRAM (bis zu 4 Chip-selects)
- Zugriff auf Memory-Mapped Special Function Registers (CCSRs)
- Ausgabe von Informationen über das System (Taktfrequenzen, Speichergrößen)
- Speicherung von systemspezifischen Informationen (Variante, Seriennummer, MACs)
- Manueller Start von Applikationen
- Automatischer Start von Applikationen
- .CMM-File für Erstinitialisierung und Laden des Monitors über Hardware-Debugger
- POST (Power-On Self Test)
- Vorinitialisierung der CAN-Controller
- Bereitstellung von Schreib-/Lesefunktionen für die Modul-EEPROMs (Konfiguration/Anwender)
- Bereitstellung von Konfigurations-/Lesefunktionen für die Modul-RTC
- Bereitstellung von Konfigurations-/Lesefunktionen für den Modul-Tempersensur

1.5 Wunschkriterien

Es wurden keine Wunschkriterien genannt.

1.6 Abgrenzungskriterien

- Caches werden nicht initialisiert (Caches nicht aktiv)
- Keine Funktionen zur Unterstützung der MPC-Schnittstellen SMC, FCC, MCC, SPI, RapidIO (parallel und seriell), PCI, PCI-X, PCI-Express, TSEC
- Keine Zugriffsfunktionen für CAN-Controller
- Keine Funktionen zur Unterstützung der Security Engine
- Kein automatisches Kopieren und Starten einer Applikation aus dem NAND-Flash
- Kein Wear Levelling bei NOR- und NAND-Flash
- Keine Unterstützung eines Dateisystems bei NOR- und NAND-Flash
- Beim MPC8548 und seinen Derivaten werden 3- und 4-lagige SDRAM Stacks derzeit nicht unterstützt.

2. Produktumgebung

2.1 Software

Der Monitor stellt eine Standalone-Software dar. Er benötigt keine zusätzliche Software (Betriebssystem, Initialisierungsfunktionen etc.). Wird der Monitor im initialen Betriebsmodus benutzt kann durch ihn aber eine beliebige Applikation (z.B. ein Betriebssystem) gestartet werden.

2.2 Hardware

Die Hardware setzt sich aus dem Starterkit STK85XX in Verbindung mit dem Minimodul TQM85XX von TQ-Components mit folgenden Ausbaustufen zusammen:

2.2.1 CPU

- MPC8540
- MPC8560
- MPC8541E
- MPC8555E
- MPC8543 und MPC8543E
- MPC8545 und MPC8545E
- MPC8547E
- MPC8548 und MPC8548E

2.2.2 NOR-Flash

- 4 MiB bis 1 GiB Speicherausbau an bis zu 2 Chip-selects (max. 512 MiB je Chip-select, max. 2048 Sektoren)
- SPANSION Flash-Bausteine mit Common Flash Memory Interface (CFI).
Zugriffszeit innerhalb der Grenzen des Memory Controllers frei konfigurierbar.

2.2.3 NAND-Flash

- 128 MiB bis 4 GiB Speicherausbau, realisiert als 1x8 oder 2x8 Bit Device mit 1 bis 4 Chip-selects
- SLC Large Block Flashes (Seitengröße 2048 Datenbytes)

2.2.4 SDRAM

- 64 MiB bis 2 GiB Speicherausbau (max. 512 MiB je Chip-select)
- Gemäß DDR1- und DDR2-Standard

2.2.5 EEPROM (Daten)

- Speicherausbau 4 KiB bis zu 32 KiB
- Ansteuerung über den I2C-Bus des I2C-Controllers im MPC85XX
- Maximal mögliche I2C-Taktfrequenz 400 kHz
- STM: M24C32, M24C64, M24128, M24256
Atmel: AT24C32, AT24C64, AT24C128, AT24C256
Microchip: 24LC32A, 24LC64, 24LC128, 24LC256

2.2.6 EEPROM (Konfiguration)

- Speicherausbau 128 Bytes
- Ansteuerung über den I2C-Bus des I2C-Controllers im MPC85XX
- Maximal mögliche I2C Taktfrequenz 400 kHz
- STM M24C01 und Atmel AT24C01

2.2.7 RTC

- Ansteuerung der prozessorexternen RTC über den I2C-Bus des I2C-Controllers im MPC85XX
- Maximal mögliche I2C-Taktfrequenz 400 kHz
- Dallas / Maxim Bausteine vom Typ DS1337U

2.2.8 Temperatursensor

- Ansteuerung über den I2C-Bus des I2C-Controllers im MPC85XX
- Maximal mögliche I2C-Taktfrequenz 400 kHz
- National Semiconductor oder Philips LM75

2.2.9 CAN

- Intel AS82527 CAN-Controller (CAN 2.0B)

2.2.10 Andere Hardware

Bei entsprechender Kompatibilität der Hardware ist der Einsatz des Monitors prinzipiell auch auf anderen Basisboards möglich. Diese Hardware sollte folgende zusätzliche Eigenschaften besitzen:

- Beschaltung des ENMON#-Signals mit Jumper oder Pegelwandlung für RS232 (Umschaltung zwischen interaktivem und initialem Modus)
- Reset-Taster oder Pegelwandlung an RS232 (Reset über RTS)
- Anschlussmöglichkeit für RS232 (Kommunikation)

Die Funktionsfähigkeit des Monitors kann jedoch mangels Testmöglichkeiten für andere Basisboards nicht garantiert werden.

2.3 Orgware

Für den interaktiven Betriebsmodus wird eine serielle Verbindung zu einem Terminalprogramm benötigt. Als Kommunikationspartner kann jeder Standard-Arbeitsplatzrechner eingesetzt werden, der eine freie serielle Schnittstelle (RS232) besitzt.

2.4 Produktschnittstellen

Der MON85XX reiht sich in die Produktfamilie der Monitorprogramme für die TQ-Module der TQM8XXL/M-, TQM82XX- und TQM83XX-Familie ein. Das Look & Feel sowie der Funktionsumfang des MON85XX wurde an die schon bestehenden Monitorprogramme soweit wie möglich angeglichen.

3. Produktfunktionen

Die Produktfunktionen des Monitors spalten sich in die Bereiche Moduldetektierung, Power-On Self Test (POST), Initialisierung, Betriebsmodi und Dokumentation auf.

3.1 Moduldetektierung

Zur Detektierung der verwendeten Hardware (Modultyp, Ausbaustufe etc.) wurden Funktionen bereitgestellt, die alle Informationen aus dem Hardware Information Block (HWIB) und dem Configuration Information Block (CIB) extrahieren und die syntaktische Korrektheit der Blöcke überprüfen. Ein Beispiel für einen HWIB- und einen CIB-Speicherauszug ist in Kapitel 4 Produktdaten ersichtlich.

3.1.1 Hardware Information Block

Der im NOR-Flash des Moduls abgelegte HWIB wird geparkt und die extrahierten Informationen werden bei syntaktischer Korrektheit in eine globale Datenstruktur kopiert. In dieser Struktur wird auch vermerkt, ob die jeweilige Information erfolgreich geparkt werden konnte. Die Funktion benötigt keine Aufrufparameter und gibt einen Ausführungsstatus zurück. Dieser besitzt den Wert OK falls die Syntax des HWIB der BNF in Abbildung 2 entspricht. Wird während des Parsens ein syntaktischer Fehler detektiert, so bricht die Funktion das Parsen ab und gibt ERROR zurück.

Hinweis:

Die in Abbildung 2 gezeigten Teile <tqld>, <mOption> und <cOption> bilden den Variantenschlüssel des Moduls. Der Punkt am Ende des Feldes <custNumber> gehört jedoch nicht mehr zum Variantenschlüssel.

BNF Syntax of the Hardware Information Block (HWIB)

```

<HWIB>          ::= <tqId><mOption><cOption><rev><serNumber><macs><termination>

<tqId>          ::= 'TQM' {<decimalNum>} * (4, 4)

<mOption>       ::= <opFlash><opSdram><opCan><opConnector><opNand><opEeprom><opZ>
  <opFlash>      ::= <opShift>
  <opSdram>      ::= <alphabetic> { '2' | '3' | '4' } * (0, 1)
  <opCan>        ::= '0' | 'A' | 'B'
  <opConnector> ::= 'A'
  <opEeprom>     ::= '0' | <opShift>
  <opNand>       ::= { <opShift> <decimalNum> 'A' | 'B' } * (0, 1)
  <opZ>         ::= { <decimalNum> } * (1, 2) '-'

<cOption>       ::= <opEncrypt><clock><cpuMask><tRange><custNumber>
  <opEncrypt>    ::= { 'E' } * (0, 1)
  <clock>        ::= { <decimalNum> } * (3, 3) | <cpuClock><ddrSpeed>
  <cpuClock>     ::= { <alphabetic> } * (2, 2)
  <ddrSpeed>     ::= { <alphabetic> } * (1, 1)
  <cpuMask>      ::= <alphabetic> | '?'
  <tRange>       ::= 'I' | 'C'
  <custNumber>   ::= { '-' { <decimalNum> } * (5, 5) } * (0, 1) '.'

<rev>          ::= { <decimalNum> } * (4, 4)

<serNumber>    ::= <space> { <decimalNum> } * (8, 10)

<macs>        ::= <space> { <hexNum> } * (12, 12) <space> <decimalNum>

<termination> ::= 0

<opShift>     ::= 'A' | 'B' | ... | 'K'
<alphabetic>  ::= 'A' | ... | 'Z'
<decimalNum>  ::= '0' | '1' | ... | '9'
<hexNum>      ::= '0' | '1' | ... | '9' | 'A' | ... | 'F'
<space>       ::= ' '

```

Hints about the notation:

- Nonterminal signs X : <X>
- Terminal signs c : 'c'
- Or-Operator : |
- Brackets : []
- Loop 1-2 : { } * (1, 2)

Abbildung 2: BNF des HWIB

3.1.2 Configuration Information Block

Der im NOR-Flash des Moduls abgelegte CIB wird geparkt und die extrahierten Informationen werden bei syntaktischer Korrektheit in eine Datenstruktur kopiert. In dieser Struktur wird auch vermerkt, ob die jeweilige Information erfolgreich geparkt werden konnte. Die Funktion benötigt keine Aufrufparameter und gibt einen Ausführungsstatus zurück. Dieser besitzt den Wert OK falls die Syntax des CIB der BNF in Abbildung 3 entspricht. Wird während des Parsens ein syntaktischer Fehler detektiert, so bricht die Funktion das Parsen ab und gibt ERROR zurück.

BNF Syntax of the Configuration Information Block (CIB)

```

<CIB>          ::= <option>{<space><option>}*<termination>
<option>       ::= <operator><space><parameter>
<operator>     ::= '-'['cu'|'cc'|'cb'|'tf'|'td']
<parameter>   ::= <decimalNum>|<hexNum>
<decimalNum>  ::= {'0'|'1'|...|'9'}*(1,10)
<hexNum>      ::= '0'['x'|'X']{'0'|'1'|...|'9'|'A'|...|'F'|'a'|...|'f'}*(1,8)
<termination> ::= 0
<space>       ::= ' '
    
```

Hints about the notation:

- Nonterminal signs X : <X>
- Terminal signs c : 'c'
- Or-Operator : |
- Brackets : []
- Loop 1-2 : {}*(1,2)

Abbildung 3: BNF des CIB

3.2 Power-On Self Test (POST)

Der POST ist in drei Teile untergliedert, wobei der POST 1A und POST 1B zum POST 1 zusammengefasst sind. Im POST 1 werden alle Tests durchgeführt, die für den Betrieb des Monitors notwendig sind, bzw. sehr wenig Zeit beanspruchen. Im POST 2 werden alle Tests durchgeführt, die für den Betrieb des Monitors nicht unbedingt notwendig sind und merklich (Millisekundenbereich) Zeit in Anspruch nehmen.

Die Phasen POST 1 und POST 2 können vom Benutzer durch Flags jeweils separat abgeschaltet werden (siehe Kapitel 4 Produktdaten). Es wird nicht nur die Ausgabe der Fehlermeldung unterdrückt, sondern die komplette Ausführung der Tests. Im Initialmodus kann dadurch eine Applikation trotz eines POST-Fehlers gestartet werden.

Hinweis:

Diese Flags sind im Wesentlichen für die Entwicklungsphase gedacht. Die Verantwortung für die Unterdrückung der Fehlermeldungen und eventuelle Folgen liegt hier beim Anwender.

Wird der Monitor über einen HW-Debugger ins RAM geladen und dort gestartet werden die POST-Tests nicht ausgeführt.

3.2.1 POST 1A

Die Funktionalität des POST 1A überprüft alle für eine anschließende Initialisierung notwendigen Parameter (HWIB und CIB) auf ihre grundsätzliche Plausibilität. Das Ergebnis des POST 1A (POST_OK oder Fehlercode der gescheiterten Plausibilitätsprüfung) wird zurückgegeben.

Bei syntaktischer Korrektheit des HWIB wird die Plausibilität der folgenden HWIB-Parameter überprüft:

- Entspricht die Modulidentifikation einem bekannten Prozessortyp (MPC8540, MPC8560, MPC8541E, MPC8555E, MPC8543(E), MPC8545(E), MPC8547E oder MPC8548(E))?
- Liegt die gearste NOR-Flash-Größe in einem plausiblen Bereich (4 MiB <= Flash <= 1 GiB)?
- Liegt die gearste NAND-Flash-Größe in einem plausiblen Bereich (128 MiB <= Flash <= 8 GiB)?
- Liegt die gearste RAM-Größe in einem plausiblen Bereich (64 MiB <= RAM <= 2 GiB)?

- Wird die maximale Anzahl (2) von optionalen CAN-Controllern nicht überschritten?
- Ist die geparte Steckerbauhöhe (5mm) korrekt?
- Liegt die geparte EEPROM-Größe in einem plausiblen Bereich (EEPROM \leq 32 KiB)?
- Befindet sich die Z-Option des HWIB in einem plausiblen Bereich ($Z \leq 39$)?
- Befindet sich die E-Option des HWIB (MPC mit oder ohne Security Engine) in einem plausiblen Bereich (Parameter fehlt oder ist gleich `E`)?
- Entspricht die geparte CPU-Taktfrequenz einer plausiblen Frequenz (400 MHz – 1500 MHz)?
- Entspricht die geparte DDR-Geschwindigkeit (= CCB-Taktfrequenz = 2 x DDR-Bustaktfrequenz) einer plausiblen Frequenz (200 MHz – 666,7 MHz) sofern dieser Parameter angegeben ist?
- Entspricht die geparte CPU-Maske / -Revision einer plausiblen Maske / Revision (A – Z oder ?)?

Die Funktion benötigt keine Aufrufparameter und gibt einen Ausführungsstatus zurück. Dieser besitzt den Wert OK falls alle durchgeführten Plausibilitätsprüfungen erfolgreich durchlaufen wurden. Erscheint während der Prüfung ein Parameter nicht plausibel, so vermerkt die Funktion dies in einer globalen Struktur. Anschließende, noch ausstehende Plausibilitätsprüfungen werden nicht mehr durchgeführt. Falls eine Plausibilitätsprüfung gescheitert ist, wird als Ausführungsstatus ERROR zurückgegeben.

Bei syntaktischer Korrektheit des CIB wird die semantische Korrektheit der folgenden CIB-Informationen überprüft:

- Liegt die geparte CPU-Taktfrequenz (-cu) im plausiblen Bereich (400 MHz – 1500 MHz)?
- Ist die geparte CPU-Taktfrequenz (-cu) gleich der vom Monitor ermittelten Frequenz?
- Liegt die geparte CCB-Taktfrequenz (-cc) im plausiblen Bereich (200 MHz – 666,7 MHz)?
- Ist die geparte CCB-Taktfrequenz (-cc) gleich der vom Monitor ermittelten Frequenz?
- Liegt die geparte Local-Bus-Taktfrequenz (-cb) im plausiblen Bereich (16,7 MHz – 166,7 MHz)?
- Ist das Verhältnis von CCB-Taktfrequenz zu Local-Bus-Taktfrequenz plausibel (2:1, 4:1, 8:1 oder 16:1)?
- Liegt das geparte NOR-Flashtiming (-tf) im plausiblen Bereich (Nur Timingbits gesetzt)?
- Liegt die geparte DDR-SDRAM CAS-Latency (-td) im plausiblen Bereich (2 oder 2,5)? Dies gilt nur für DDR1. Für DDR2-SDRAM initialisiert der Monitor den Controller mit festgelegten Werten, abhängig von der DDR-Geschwindigkeit (siehe Kapitel 3.3.10).

Die Funktion benötigt keine Aufrufparameter und gibt einen Ausführungsstatus zurück. Dieser besitzt den Wert OK falls alle durchgeführten Plausibilitätsprüfungen erfolgreich durchlaufen wurden. Erscheint während der Prüfung ein Parameter nicht plausibel, so vermerkt dies die Funktion in einer globalen Struktur. Anschließende, noch ausstehende Plausibilitätsprüfungen werden nicht mehr durchgeführt. Falls eine Plausibilitätsprüfung gescheitert ist, wird als Ausführungsstatus ERROR zurückgegeben.

Ist der POST 1A abgeschaltet, belässt der Monitor den Frequenzteiler für den Local Bus (Register LCRR[CLKDIV]) auf seinem Default-Wert. Die angezeigten Betriebsfrequenzen in der Boot-Meldung (Current-Werte) ergeben sich aus dem gemessenen CCB-Takt und den Default-Einstellungen für den CPU-Taktfrequenzmultiplikator und dem Teiler für den Local Bus.

Hinweis:

Das verwendete NOR-Flash-Timing im CIB (Parameter -tf) passt dann möglicherweise nicht zur Local-Bus-Taktfrequenz. Es erfolgt keine automatische Anpassung des Flash-Timings!

3.2.2 POST 1B

Der POST 1B überprüft die im folgenden aufgeführten Parameter. Das Ergebnis des POST 1B (POST_OK oder Fehlercode der gescheiterten Plausibilitätsprüfung) wird zurückgegeben.

- Konnte der I2C-Controller erfolgreich initialisiert werden?
- Entspricht die geparte NOR-Flash-Größe im HWIB der detektierten Größe und wird der Flash-Typ vom Monitor unterstützt?
- Konnte das NAND-Flash gefunden werden und die BBT erfolgreich eingerichtet werden?
- Entspricht die geparte NAND-Flash-Größe im HWIB der detektierten Größe und wird der Flash-Typ vom Monitor unterstützt?
- Entspricht die Anzahl der NAND-Flash Chip-selects im HWIB der detektierten Anzahl?
- Entspricht die Anzahl der NAND-Flash Devices im HWIB der detektierten Anzahl?
- Entspricht die geparte SDRAM-Größe im HWIB der ermittelten physikalischen SDRAM-Größe?
- Die geparte Anzahl der CAN-Controller im HWIB wird mit der Anzahl der physikalisch vorhandenen CAN-Controller verglichen. Hierzu werden die CAN-Controller abgefragt. Die bedingte Prüfung in Abhängigkeit vom HWIB dient dazu, Zugriffe auf eventuell an das Chip-select „CS_CAN“ angeschlossene kundenspezifische Hardware zu vermeiden. Hierdurch kann eine Überbestückung gegenüber dem HWIB aber nicht entdeckt werden. Das im Normalbetrieb genutzte Acknowledge-Signal des / der CAN-Controller kann für den Test nicht verwendet werden, da damit bei einer Fehlfunktion (Acknowledge bleibt aus) der PowerPC-Bus hängen bleiben könnte. Deshalb wird ausschließlich für den Test über abgeänderte UPM-Sequenzen ein Zugriff realisiert, der so langsam ist, dass ein Acknowledge nicht notwendig ist.

3.2.3 POST 2

Der POST 2 überprüft die im folgenden aufgeführten Parameter. Das Ergebnis des POST 2 (POST_OK oder Fehlercode der gescheiterten Plausibilitätsprüfung) wird zurückgegeben.

- Entspricht die geparte EEPROM-Größe im HWIB der physikalisch ermittelten EEPROM-Größe? Hierzu wird auf das EEPROM mehrfach lesend und schreibend zugegriffen. Die bereits im EEPROM vorhandenen Daten werden nach dem Test wiederhergestellt. Stimmen die Größen nicht überein, so ist kein EEPROM laut HWIB vorhanden, oder es ist kein fehlerfreier Zugriff möglich. Die bedingte Prüfung in Abhängigkeit vom HWIB dient dazu, um Zugriffe auf mögliche am I2C-Bus angeschlossene kundenspezifische Hardware mit der gleichen Adresse, wie das vom Hersteller bestückte EEPROM, zu vermeiden. Eine Überbestückung gegenüber dem HWIB kann aber nicht entdeckt werden.
- Ist die moduleigene RTC über den I2C-Bus ansprechbar?
- Ist der moduleigene Temperatursensor über den I2C-Bus ansprechbar?

3.2.4 Signalisierung von POST-Ergebnissen

Die Signalisierung der POST-Ergebnisse ist abhängig vom gewählten Betriebsmodus des Monitors:

Im initialen Modus wird das Auftreten eines Fehlers beim POST durch Blinken der Reset-LED angezeigt. Dazu versetzt der Monitor das Modul wiederholt in den Reset-Zustand. Eine Kundenapplikation wird nicht gestartet.

Im interaktiven Modus wird nach der Boot-Meldung des Monitors eine POST-Fehlerbeschreibung im Klartext ausgegeben. Die einzelnen Fehlermeldungen sind unter Tabelle 12: POST-Meldungen beschrieben. Der Monitor läuft danach mit ggf. eingeschränkter Funktionalität weiter (z. B. Flash nicht beschreibbar).

Im interaktiven Modus wird in der Startmeldung auf den / die abgeschalteten POST(s) hingewiesen.

3.3 Initialisierung

3.3.1 Clocks

Die PLL-Konfiguration für Core, CCB, DDR und PCI beim MPC85XX ist über die Power-On Reset-Konfiguration ohne SW-Eingriff eingestellt.

Der Core Complex Bus (CCB) Clock wird vom Monitor vor Durchführung des POST gemessen und der CPU (Core) Clock durch Auslesen des entsprechenden Teilerfaktors im PLL-Statusregister davon abgeleitet. Beide Taktfrequenzen werden mit den Einträgen im CIB verglichen. Im Falle einer Abweichung (mehr als +/- 2 % des gemessenen CCB Clocks bzw. Core Clocks) wird eine POST-Fehlermeldung ausgegeben. Die Messung ist aber ausreichend genau, um danach den System Timer und die Baudraten zu initialisieren.

Der CCB Clock und der Local Bus Clock ist durch Parameter im Configuration Information Block definiert (siehe Kapitel 4 Produktdaten). Im POST wird das Teilerverhältnis zwischen CCB Clock und Local Bus Clock berechnet. Wenn sich ein korrektes Teilerverhältnis CCB:LB von 2:1, 4:1, 8:1 oder 16:1 ergibt, wird der Local Bus Clock durch Einstellung des Registers LCRR[CLKDIV] entsprechend konfiguriert. Weicht das errechnete Verhältnis um mehr als +/- 3 % vom nominellen Verhältnis ab, erfolgt eine POST-Fehlermeldung (siehe Tabelle 12: POST-Meldungen). In diesem Fall verändert der Monitor die Einstellung des Local Bus Clock nicht. Der Local Bus Clock resultiert daher aus dem gemessenen CCB Clock geteilt durch den Standardteilerfaktor im Register LCRR.

3.3.2 MPC-Typ

Der MPC-Typ ist im HWIB hinterlegt. Zudem wird eine Ermittlung anhand des System Version Registers (SVR) durchgeführt. Falls im HWIB kein gültiger Wert vorhanden ist, wird der detektierte Wert verwendet. Enthält der HWIB eine Angabe zum MPC-Typ, wird diese verwendet, aber mit dem detektierten Wert verglichen. Im Falle einer Diskrepanz wird eine POST-Fehlermeldung ausgegeben. Entspricht der detektierte Wert keinem unterstützten MPC-Typ, wird ebenfalls eine POST-Fehlermeldung ausgegeben.

3.3.3 Watchdog

Der Watchdog des MPC85XX wird durch den Monitor nur zum Zweck des Modul-Resets im Falle eines POST-Fehlers im initialen Modus oder im Falle eines Applikationsreboots benutzt. Da der Watchdog nach einem Reset / Power-On beim MPC85XX generell deaktiviert ist, erfolgt kein Zugriff auf die entsprechenden Register im Normalbetrieb. Eine Benutzung und Konfiguration des Watchdogs kann bei Bedarf durch die Kundenapplikation erfolgen.

3.3.4 Memory Management Unit (MMU)

Die Memory Management Unit wird wie folgt konfiguriert:

- 1:1-Übersetzung von effektiven zu physikalischen Adressen
- Deaktivierung der Caches (Data / Instruction)
- Verwendung des TLB1 für Einträge. Die TLB-Einträge erfolgen so, dass sie die in Tabelle 1: Local Memory Map gezeigten Speicherbereiche abdecken.
- Verwendung des Supervisor Modes

3.3.5 Memory-Mapped Register

Die Memory-Mapped Register des MPC85XX (Configuration, Control and Status Register) werden als 1-MiB-Bereich in den Speicher eingeblendet. Die Basisadresse dieses Bereiches wird im Register CCSRBAR festgelegt. Die Default-Einstellung der Basisadresse wird durch den Monitor nicht verändert.

Hinweis:

Eine Überlappung dieses Adressbereichs mit anderen Adressbereichen (z.B. Local Access Windows) ist nicht zulässig.

3.3.6 Monitor Bootpage

Nach einem Reset oder einem Power-On greift der MPC85XX im Default-Fall über die MMU auf den Adressbereich 0xFF80_0000 bis 0xFFFF_FFFF zu, um einen dort platzierten Bootcode auszuführen. Dieses Verhalten wird durch den Monitor nicht verändert.

Hinweis:

Wie bei den Memory-Mapped Registern darf dieser Bereich nicht mit anderen Adressbereichen überlappen. Auch der Bereich der Bootpage wird nicht in den Local Access Windows konfiguriert.

3.3.7 Local Memory Map

Die Local Memory Map bezeichnet den für den MPC85XX sichtbaren 32-Bit-Adressraum, wenn dieser auf Speicherraum oder I/O-Raum zugreift.

Die Local Memory Map des MPC85XX wird mittels der Local Access Window Register wie folgt konfiguriert:

Window	Base Address	End Address	Size	Target Interface Code (Target)
0	0x0000_0000	0x7FFF_FFFF	2 GiB	0b1111 (DDR _x -SDRAM)
1	0x8000_0000	0xBFFF_FFFF	1 GiB	0b0100 (Local Bus Controller NOR-Flash)
2	0xF000_0000	0xF7FF_FFFF	128 MiB	0b0100 (Local Bus Controller, I/O e.g. CAN, NAND-Flash)
3				Free (e.g. SDRAM Reserve)
4				Free (e.g. Rapid IO)
5				Free (e.g. PCI)
6				Free
7				Free

Tabelle 1: Local Memory Map

Hinweis:

Für die internen Memory-Mapped Register des MPC85XX ist die Benutzung eines Local Access Window Registers nicht notwendig.

3.3.8 NOR-Flash

Die Timing-Einstellung im Memory Controller für das Flash wird zunächst mit einem vorläufigen Wert initialisiert, der für alle unterstützten Local Bus Taktfrequenzen gültig ist. Wenn der CIB eine syntaktisch korrekte "-tf"-Option enthält, wird das Timing nachträglich angepasst, andernfalls bleibt die vorläufige Einstellung erhalten.

Falls der Monitor eine zweite Flash-Bank findet, konfiguriert er neben dem Chip-select LCS0 (BR0/OR0) für die erste Bank auch LCS1 (BR1/OR1). Andernfalls wird das Bit BR1[V] auf Null gesetzt.

3.3.9 NAND-Flash

Wenn im HWIB die Ausstattung des Moduls mit NAND-Flash angegeben ist, konfiguriert der Monitor LCS3 (BR3/OR3). Eine weitere Aufteilung auf die einzelnen Device Chip-selects wird durch Adressdekodierung von einer HW-Schaltung auf dem Modul sichergestellt.

Hinweis:

Im Memory Controller wird das kleinstmögliche Speicherfenster mit 32 KiB Größe eingestellt. Tatsächlich ist der Zugriff auf das NAND-Flash aber nur auf die Adressen 0xF0008000, 0xF0008200, 0xF0008400 und 0xF0008600 (Adressen der einzelnen NAND-Flash Chip-selects) zulässig!

Der NAND-Flash-Treiber führt zum Initialisierungszeitpunkt folgende Aufgaben aus:

- **Initialisierung der UPMB:**
Für das NAND-Flash wird die UPMB je nach verwendeter Local Bus Frequenz (25, 33, 42, 50, 66, 83, 100, 133 oder 166 MHz) optimal eingestellt. Sollte die eingestellte Local Bus Frequenz größer als 1 % der unterstützten Frequenzen sein, wählt der Treiber automatisch das Timing für die nächsthöhere unterstützte Frequenz aus. Damit wird ein langsames Timing gewählt, so dass ein Betrieb des NAND-Flashs noch möglich ist. Die gewählte Frequenz wird im interaktiven Modus des Monitors mit dem Kommando MEMORY ersichtlich. Übersteigt die tatsächliche Local Bus Frequenz 166 MHz erfolgt ein POST-Fehler.
- Auflösung des Herstellerschlüssels und des Typ-Identifikatorschlüssels in den Herstellernamen und Bausteintyp (Versorgungsspannung, Bitbreite).
- Ermittlung der Geometrie und Anzahl der Devices (1x8 Bit oder 2x8 Bit) anhand der ID-Bytes.
- Detektierung der Anzahl der Chip-selects.
- **Erstellung einer Bad-Block-Tabelle (BBT) inklusive Ermittlung der vom Hersteller markierten Blöcke:**
Alle vom Monitor vorgenommenen Markierungen der Blöcke finden zunächst in einer RAM-Tabelle statt. Anschließend wird diese Tabelle in das NAND-Flash geschrieben.

Die Bad-Block-Tabelle wird standardmäßig im letzten Block des NAND-Flashs abgelegt. Ist dieser Block selbst als schlecht markiert, wird die Tabelle im zweitletzten Block abgelegt. Ist auch dieser Block als schlecht markiert, erfolgt eine Fehlermeldung im POST. Die beiden letzten Blöcke sind somit zum Ablegen der BBT vom Monitor reserviert (siehe auch Abbildung 4).

Nach jedem Bootvorgang wird zunächst in den Spare Bytes der ersten Seite des BBT-Blocks nach dem ID-String „TQ-MONITOR“ gesucht. Schlug der Versuch fehl, wird im zweitletzten Block gesucht. Wurde auch dort keine ID gefunden, legt der Monitor die Tabelle wie oben beschrieben neu an.

Ist dagegen eine Tabelle vorhanden, wird eine Kopie davon im RAM angelegt. Wird während des Kopiervorgangs aus dem letzten Block ein nicht korrigierbarer Fehler festgestellt (ECC), werden die Einträge der RAM-Tabelle gelöscht und eine neue Abfrage nach herstellereitig markierten Blöcken durchgeführt. Anschließend wird der Block als schlecht markiert und die neue BBT im zweitletzten Block gespeichert.

Eine weitergehendes Fehlermanagement für den zweitletzten Block findet nicht statt. Er dient nur als Rückfalllösung falls der Standardblock nicht verwendbar ist.

Um die BBT platzsparend zu speichern, erfolgt eine Blockmarkierung mit zwei Bits, so dass bis zu vier Markierungen in einem Byte untergebracht werden.

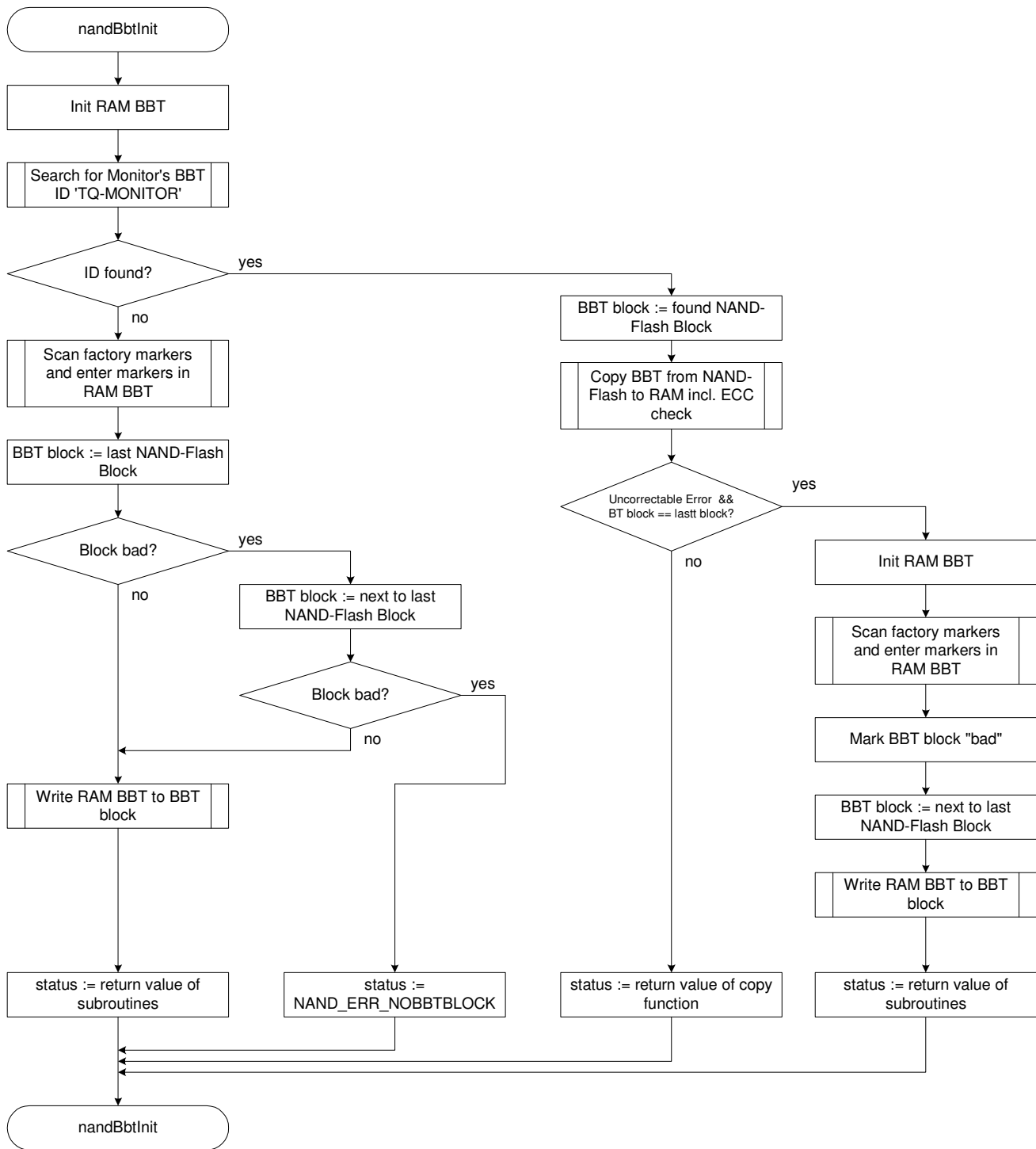


Abbildung 4: BBT-Initialisierung

3.3.10 SDRAM

- Detektierung der SDRAM-Größe:**

Die vorhandene SDRAM-Größe wird automatisch detektiert, sofern am jeweiligen Chip-select ein RAM angeschlossen ist. Dazu greift die Initialisierungsroutine auf die Information im HWIB zurück, die die Anzahl der Chip-selects angibt. Fehlt diese Angabe oder ist der HWIB ungültig, wird standardmäßig die Anzahl 1 angenommen. Die Chip-select Bounds und Config Register der restlichen Chip-selects werden dann deaktiviert.

Hinweis:

Werden im HWIB mehr Chip-selects angegeben als tatsächlich an RAM angeschlossen, startet der Monitor nicht mehr!

Werden im HWIB weniger Chip-selects angegeben als tatsächlich an RAM angeschlossen, initialisiert der Monitor nur den angegebenen Teil.

Beim MPC8548 und seinen Derivaten ist das Bit DDR_SDRAM_CFG[HSE] (Half-strength drive enable) gesetzt, so dass derzeit nur 1- und 2-lagige SDRAM Stacks unterstützt werden.

- **Unterscheidung zwischen DDR1 und DDR2:**

Der Monitor interpretiert das Feld RTYPE im Register PORDEVSR, dessen Inhalt die Power-On Reset-Konfiguration anzeigt. Beim MPC8540, MPC8560, MPC8541E und MPC8555E sind die Bits dieses Felds gleich Null.

- **Timing:**

Der Controller wird für DDR1-SDRAM mit einem Timing für CCB = DDR-Geschwindigkeit = 333 MHz eingestellt. Für DDR2-SDRAM wird der Eintrag im CIB für die CCB-Frequenz herangezogen. Der Monitor unterstützt Timings für die Frequenzen 400 MHz und 533 MHz. Das Timing für 533 MHz ist zugleich die Default-Einstellung. Weicht die im CIB angegebene Frequenz von den unterstützten ab, wählt der Monitor automatisch das Timing für die nächsthöhere unterstützte Frequenz aus. Die gewählte Frequenz wird im interaktiven Modus des Monitors mit dem Kommando MEMORY ersichtlich.

Die CAS-Latency (Defaultwert 2,5) wird für DDR1 abhängig von CIB konfiguriert. Im Falle von DDR2-SDRAM stellt der Monitor für eine DDR-Geschwindigkeit von 533 MHz die CAS-Latency fix auf 4.0 ein und für eine DDR-Geschwindigkeit von 400 MHz fix auf 3.0. Der Eintrag im CIB für die CAS-Latency wird deshalb bei DDR2 ignoriert.

- **Refresh-Interval:**

Bei DDR1 wird das Refresh-Interval in Abhängigkeit von der detektierten Speichergröße eingestellt.

Bei DDR2 ist das Intervall abhängig vom Eintrag für den CCB-Takt im CIB. Der Monitor unterstützt Refresh-Intervalle für die Frequenzen 267 MHz, 400 MHz und 533 MHz. Weicht die im CIB angegebene CCB-Frequenz von den unterstützten ab, wählt der Monitor automatisch das Refresh-Interval für die nächstniedrigere unterstützte Frequenz aus. Die gewählte Frequenz wird im interaktiven Modus des Monitors mit dem Kommando MEMORY ersichtlich.

Bei DDR2-SDRAM wird außerdem noch der Temperaturbereich im Variantenschlüssel berücksichtigt. Ist dieser Wert ungültig, wird standardmäßig das Refresh-Interval für den industriellen Temperaturbereich der jeweiligen CCB-Frequenz eingestellt.

- **Refresh-Recovery-Time:**

Die Refresh-Recovery-Time wird im Startup für DDR2-SDRAM automatisch in Abhängigkeit von der detektierten Speichergröße eingestellt.

3.3.11 **CAN-Controller**

Wenn im HWIB eine Soll-Ausstattung von mindestens einem CAN-Controller angegeben ist, wird der Memory Controller (Base / Option Register und UPMC) soweit vorinitialisiert, dass zur Aktivierung nur noch das Valid Bit im Base Register BR2[V] (LCS2) durch eine Applikation gesetzt werden muss.

Hinweis:

Im Memory Controller wird das kleinstmögliche Speicherfenster mit 32 KiB Größe eingestellt. Tatsächlich ist der Zugriff auf die CAN-Controller aber nur im Adressbereich 0xF0000000 – 0xF00001FF zulässig!

3.3.12 Monitor Speicher-Mapping:

Die einzelnen Speichertypen werden jeweils an eine feste Startadresse gemappt, die Endadresse ergibt sich dann aus der jeweils bestückten Größe. Für die SDRAM- und Flash-Bestückung wurden die in Tabelle 2: Speicher-Mapping beschriebenen Werte festgelegt. Theoretisch kann aufgrund des Speicher-Mappings das SDRAM nur bis maximal 2 GiB und das NOR-Flash nur bis 1 GiB bestückt werden.

Start address	End address max.	Memory type
0x00000000	0x7FFFFFFF	SDRAM, max. 2 GiB
0x80000000	0xBFFFFFFF	NOR-Flash, max. 1 GiB
0xF0000000	0xF0007FFF	CAN: Access only allowed to addresses 0xF0000000..0xF00001FF!
0xF0008000	0xF000FFFF	NAND-Flash: Access only allowed to 0xF0008000, 0xF0008200, 0xF0008400 and 0xF000600
0xFF700000	0xFF7FFFFFFF	MPC85XX Internal Memory

Tabelle 2: Speicher-Mapping

Monitor Speicherbedarf:

Der Monitorcode wird im NOR-Flash in 3 Bereiche aufgeteilt: Resetvektor, Bootcode und restlicher Code. Der Bootcode wird an den Anfang einer speziellen benutzerdefinierten Sektion „bootpage“ gelinkt, der restliche Code in die Sektion „text“. Der Bootcode enthält die Kernfunktionalität des Startups, die Cache-Deaktivierung sowie die MMU Initialisierung. Die Sektion „reset“ enthält den Sprungbefehl in die „bootpage“. Der Monitor belegt folgende Flash- und SDRAM-Speicherbereiche:

Start address	Size	Contents
0x80000000	256 Bytes	Applicationheader
0x80000100	module-dependent	Free up to x - 0x40001
x - 0x40000	512 Bytes	HWIB
x - 0x3FE00	512 Bytes	CIB
x - 0x3FC00	251 KiB	MON85XX Code
x - 0x1000	4092 Bytes	MON85XX Bootpage
x - 4	4 Bytes	Resetvector

Tabelle 3: Monitor Speicherbedarf im NOR-Flash

x = Letzte gültige Adresse im Flash + 1

Start address	Size	Contents
0x00000000	module-dependent	Free
y - 0x3FC00 - z	64 KiB	Stack
y - 0x3FC00	251 KiB	MON85XX Code
y - 0x1000	4092 Bytes	MON85XX Bootpage
y - 4	4 Bytes	Resetvector

Tabelle 4: Monitor Speicherbedarf im SDRAM

y = Letzte gültige Adresse im SDRAM + 1

z = Wert experimentell ermittelt + 100 % Reserve

3.3.13 Caches und L2-SRAM

Der Instruction und Data Cache des MPC85XX wird durch den Monitor nicht benutzt und deshalb deaktiviert. Leitet eine Applikation einen Reboot über den Monitor ein (siehe Kapitel 3.3.14 MON85XX (Systemsoftware)) oder tritt ein POST-Fehler im Initialmodus auf, konfiguriert der Monitor beim MPC8540, MPC8541E, MPC8555E und MPC8560 128 KiB des L2-Caches als SRAM um Daten zwischenspeichern, die nach anschließendem HRESET einen Hinweis auf den Reset-Grund geben. Beim MPC8548 und seinen Derivaten werden 512 KiB als SRAM konfiguriert. Aufgrund des HRESETs erfolgt ein Einsprung nach BOOT_FLASH und die MMU wird wieder wie unter Kapitel 3.3.4 beschrieben initialisiert. Das L2-SRAM wird deaktiviert und steht der Applikation wieder zur Verfügung.

3.3.14 MON85XX (Systemsoftware)

Die Initialisierung der Systemsoftware MON85XX setzt sich aus folgenden Teilfunktionalitäten zusammen:

- SDRAM-Test mit Detektierung der Speichergröße. Diese Funktion arbeitet zerstörungsfrei, d.h. von der Testroutine überschriebene Speicherstellen werden wieder restauriert.
- Kopieren des Monitorcodes ins SDRAM
- Relokation
- Interruptvektoren auf Stubs für Interrupt-Handler setzen
- Stack initialisieren
- NOR-Flash-Typ feststellen und CFI-Informationen auswerten
- Monitorcode und Stack werden ans obere Ende des verfügbaren SDRAMs kopiert.
- Um neben dem normalen Start aus dem NOR-Flash auch den Start aus dem SDRAM zu ermöglichen, werden Teile der Initialisierung bedingt ausgeführt (Siehe hierzu Tabelle 5: Monitor Einsprungadressen).

Damit der Monitorcode unabhängig von seiner Platzierung im SDRAM ausführbar ist, wurden folgende Schritte durchgeführt.

- Kompilieren mit Option -fpic (Position Independent Code)
- Relokation aller globalen Variablen (außer Inhalte globaler Zeiger) und Funktionsadressen. Dies geschieht im Startup-Code durch Anpassung der Global Offset Table (GOT), indem die Einträge der GOT mit einem Offset beaufschlagt werden, entsprechend der Lage des Monitors im SDRAM.
- Relokation der Inhalte aller globalen Zeiger im betreffenden Programm-Modul entsprechend dem im Startup-Code ermittelten Offset.

Es werden 4 Einsprungadressen zum Start des Monitors zur Verfügung gestellt:

Adresse	Makro	Bedeutung
x - 0x1000	BOOT_FLASH	Resetvektor, normaler Start Application Boot Type = -1 = 0xFFFFFFFF Nach der Resetphase des Prozessors liegt die Einsprungadresse bei 0xFFFFFFFFFC. Die Bootpage beginnt bei 0xFFFFF000. Der Monitor ändert die Basisadresse des NOR-Flashs später auf 0x80000000.
x - 0x0FF0	BOOT_SOFT	Wie BOOT_FLASH, jedoch immer in den interaktiven Modus (für Start aus NOR-Flash im Debugger)
0x3F020	BOOT_SDRAM	Nur teilweise Initialisierung für Start per Debugger. Im NOR-Flash liegt diese Einsprungadresse bei x - 0x0FE0.
x - 0x0FD0	BOOT_APP_RE	Vollständige Initialisierung wie BOOT_FLASH, jedoch immer Start der Applikation (für Reboot aus Applikation). Der Application Boot Type wird aus Register R3 übernommen und wieder an die Applikation als Parameter in R3 zurückgegeben.

Tabelle 5: Monitor Einsprungadressen

x = Letzte gültige Adresse im Flash +1

Abhängig von der Startart (Einsprungadresse) und dem vorhandenen lokalen SDRAM werden Initialisierungen und Speichertests wie folgt bedingt ausgeführt:

- Beim Einsprung an die Adresse BOOT_FLASH wird der Speichertest durchgeführt, anschließend wird der Monitorcode und sein Stack am oberen Ende des SDRAMs platziert (siehe Tabelle 4: Monitor Speicherbedarf im SDRAM).
- Beim Einsprung an die BOOT_SOFT-Adresse wird die gleiche Initialisierung wie bei BOOT_FLASH verwendet, jedoch immer in den interaktiven Modus verzweigt.
- Beim Einsprung an die Adresse BOOT_SDRAM erfolgt kein SDRAM-Speichertest, kein HWIB- und CIB-Scan und kein POST. Der Monitorcode und sein Stack werden am oberen Ende der minimal zulässigen SDRAM-Ausbaustufe der Module (64 MiB) platziert.
- Beim Einsprung an die Adresse BOOT_APP_RE wird die gleiche Initialisierung wie bei BOOT_FLASH verwendet. Der Monitor führt dazu ein HRESET durch, damit die Applikation von einem definierten Zustand des Prozessors ausgehen kann. Es wird immer in den Initialmodus verzweigt. Vor dem Einsprung in den Monitor kann die Applikation im Register R3 dem Monitor einen Wert übergeben, den der Monitor über den Reset hinweg sichert und anschließend wieder an die Applikation zurückgibt.

Für die Exception "External Interrupt" wird ein Stub zur Verfügung gestellt, der den Systemzustand soweit sichert, dass ein in C geschriebener Interrupt-Handler aufgerufen werden kann. Für alle übrigen Exceptions wird ein Primitiv-Handler bereitgestellt, der das System anhält (Endlosschleife). Im zweiten Fall ist mit Hilfe eines Hardware-Debuggers dann noch eine Fehleranalyse möglich.

3.3.15 Serielle Schnittstelle

Es wird eine RS232-Schnittstelle unterstützt. Diese wird abhängig vom Prozessortyp entweder über die SCC1 / SCC2 (MPC8560) oder den UART0 / UART1 (MPC8540, MPC8541E und MPC8555E) realisiert.

Die serielle Schnittstelle wird abhängig von einer automatischen Baudratenerkennung wie folgt initialisiert:

- Automatische Selektierung des UART0 oder UART1 bzw. SCC1 oder SCC2.

- Es werden Baudraten zwischen 19200 und 115200 beim UART und zwischen 2400 und 115200 bei der SCC unterstützt.
- Das Datenformat besteht aus 8 Datenbits, ein Stopbit und kein Paritätsbit (8N1).
- Flusssteuerung wird nur software-technisch durch XON/XOFF realisiert (kein CTS, RTS).
- Steuerzeichen Control-C wird als Funktionsabbruch für die in Kap. 6.1 Kommandoeingabe genannten Kommandos erkannt.

Die automatische Port- und Baudratendetektierung kann mit einem Flag im Applikationsvorspann deaktiviert werden (siehe Tabelle 9 und Tabelle 10).

3.3.16 I2C-Controller

Es wird der I2C-Controller benutzt, der in allen MPC85XX-Typen vorhanden ist. Der im CPM integrierte I2C-Controller wird nicht unterstützt. Der I2C-Controller wird mit einem I2C-Bustakt von 100 kHz konfiguriert, um sicherzustellen, dass auch I2C-Teilnehmer ohne Konflikte angeschlossen werden können, die nur über den Standard-Mode verfügen. Der im I2C-Controller integrierte Spike-Filter wird in seiner Default-Einstellung belassen.

3.3.17 CAN-Controller

Die auf dem Modul optional vorhandenen CAN-Controller werden so initialisiert, dass sie ein Acknowledge-Signal generieren und der Takt des zweiten CAN-Controllers vom Taktausgang (CLKOUT) des ersten getrieben wird. Diese Konfiguration findet beim Test der CAN-Controller statt, und bleibt anschließend erhalten.

Da der Datendurchsatz beim CAN relativ gering ist, wird die UPMC mit einem Timing (166 MHz) initialisiert, das bei allen unterstützten Local Bus Frequenzen funktioniert.

3.3.18 Timer

Es wird für den internen Gebrauch ein System Timer eingerichtet, der unabhängig vom Systemtakt immer in 1-ms-Schritten zählt.

3.3.19 Hardware-Debugger

Es wurde ein .CMM-File für die Inbetriebnahme und das manuelle Aufspielen des Monitors (Initialisierungsskript) für den Lauterbach ICD in Verbindung mit der SW Trace32 erstellt. Dieses ermöglicht auf einem Modul, den Monitor ins SDRAM zu laden und dort zu starten. Das Skript ist ein eigenständiges Dokument mit folgender Funktion:

Initialisierung des DDR-Controllers für die SDRAMs. Die Initialisierung nutzt jeweils nur die kleinste Speicherausbaustufe. Skript und Monitor funktionieren auch mit anderen Ausbaustufen, nutzen dann aber nur einen Teil des Speichers. Es wird das langsamste Timing für NOR-Flash und SDRAM, aber der schnellste Refresh eingestellt.

Hinweis:

Da der Monitor auch ohne gültigen Hardware Information Block und Configuration Information Block lauffähig ist, werden diese durch das CMM-File nicht erzeugt.

3.4 Betriebsmodi

Hier sind alle Funktionen aufgeführt, welche in den beiden Betriebsmodi (initial oder interaktiv) durch den Monitor verwendet werden. Abhängig von der Einsprungadresse in den Monitor (siehe Tabelle 5: Monitor Einsprungadressen) und vom Zustand des ENMON#-Signals wird in den initialen oder interaktiven Modus verzweigt.

Beim Einsprung an die Adresse BOOT_FLASH wird bei einem Low-Pegel des ENMON#-Signals in den interaktiven Modus verzweigt (nur durch den Monitor benutzbar).

Beim Einsprung an die Adresse BOOT_SOFT oder BOOT_SDRAM wird immer in den interaktiven Modus verzweigt (nur für Debugger).

Beim Einsprung an die Adresse BOOT_APP_RE wird immer in den initialen Modus verzweigt.

3.4.1 Initialer Betriebsmodus

Nach der Initialisierung des Systems und bei erfolgreichem POST, beziehungsweise deaktiviertem POST (POSTxEN-Flags), werden folgende Funktionen verwendet:

Folgende durch den Monitor benutzte Schnittstellen werden deaktiviert:

- Serielle Schnittstellen
- I2C-Controller
- Timer
- Externe Exceptions (MSR[EE])

Es wird der Autostartteil des Applikationsvorspanns im NOR-Flash ausgewertet (siehe Kapitel 4.3). Abhängig vom ID-String im Autostartteil werden nun folgende Aktionen ausgeführt:

- Entspricht der ID-String dem String „**MON85xx Autostart ROM**“ wird der im Autostartteil definierte Parameter „Start“ als Adresse für den Einsprung in die Applikation (liegt im NOR-Flash) benutzt. Die Parameter „Size“ und „Crc32“ des Applikationsvorspanns sind ohne Bedeutung.
- Entspricht der ID-String dem String „**MON85xx Autostart ROMCRC**“ wird der im Autostartteil definierte Parameter „Size“ benutzt, um über die Applikation eine CRC-Summe zu berechnen (Startadresse der CRC-Berechnung ist immer 0x80000100). Diese wird mit dem Parameter „Crc32“ verglichen. Bei einem positiven Ergebnis wird der Parameter „Start“ als Einsprungadresse in die Applikation (liegt im NOR-Flash) benutzt. Schlug der CRC-Check fehl, verzweigt der Monitor in eine Endlos-Warteschleife.
- Entspricht der ID-String dem String „**MON85xx Autostart RAM**“ wird der im Autostartteil definierte Parameter „Size“ benutzt, um den definierten NOR-Flash-Speicherbereich inklusive Applikationsvorspann ins RAM zu kopieren (Startadresse des NOR-Flash-Speicherbereichs ist immer 0x80000000, Zieladresse im RAM ist immer 0x00000000). Anschließend wird die Applikation im RAM an der durch den Parameter „Start“ definierten Adresse gestartet.
- Entspricht der ID-String dem String „**MON85xx Autostart RAMCRC**“ wird der im Autostartteil definierte Parameter „Size“ benutzt, um den definierten NOR-Flash-Speicherbereich inklusive Applikationsvorspann ins RAM zu kopieren (Startadresse des NOR-Flash-Speicherbereichs ist immer 0x80000000, Zieladresse im RAM ist immer 0x00000000). Anschließend wird anhand des Parameters „Size“ eine CRC-Summe über die Applikation berechnet (Startadresse der CRC-Berechnung ist immer 0x00000100). Diese wird mit dem Parameter „Crc32“ verglichen. Schlug der CRC-Check fehl, verzweigt der Monitor in eine Endlos-Warteschleife. Bei einem positiven Ergebnis wird die Applikation im RAM an der durch den Parameter „Start“ definierten Adresse gestartet.

- Sollte der ID-String keine der oben genannten Strings enthalten, so ist der Autostartteil ungültig und der Monitor springt ohne weitere Überprüfungen einfach die feste Adresse 0x80000100 im NOR-Flash an.

Hinweise zum Format des ID-Strings im NOR-Flash:

- Die drei Worte des ID-Strings müssen durch genau ein Leerzeichen getrennt werden.
- Die Auswertung durch den Monitor erfolgt case-sensitive.
- Ein ID-String muss durch mindestens eine binäre Null terminiert werden.
- Die doppelten Anführungszeichen dürfen nicht Teil des ID-Strings sein.

Der gestarteten Applikation werden folgende unabdingbare Informationen über den Konfigurationszustand des Systems mittels der MPC Register R3 bis R6 übergeben (siehe Tabelle 6: Parameterübergabe an die Applikation). Anhand dieser Parameter kann die Applikation z.B. erkennen, ob das System nach einem Reset hochgefahren oder lediglich durch einen Rücksprung in den Monitor neu gebootet wurde (Register R3) oder wie groß das vorhandene modulinterne NOR-Flash ist, um die Lage des Monitorcodes im NOR-Flash berechnen zu können. Schließlich werden auch die CPU-Frequenz und die Lage des Registers CCSRBAR übergeben.

Register	Kürzel	Bedeutung	Wert
R3	startType	Kaltstart (Einsprung in den Monitor bei BOOT_FLASH)	0xFFFFFFFF
		Warmstart (Einsprung in Monitor bei BOOT_APP_RE) Wert entspricht Inhalt von R3 beim Einsprung in den Monitor.	VARIABLEL
R4	CCSRBAR	Adresse, an der sich das Register CCSRBAR befindet	VARIABLEL
R5	coreClock	CPU (Core) Clock in kHz	VARIABLEL
R6	flashSize	Größe des modulinternen NOR-Flash-Speichers in Byte	VARIABLEL

Tabelle 6: Parameterübergabe an die Applikation

Es wird eine Funktion zur Berechnung einer CRC-32 zur Verfügung gestellt. Diese berechnet mittels der Parameter „Start“ und „Size“ die CRC-32 über den angegebenen Speicherbereich. Die CRC-32 Berechnung basiert auf dem Polynom 0x04C11DB7 bzw. (1) 0000 0100 1100 0001 0001 1101 1011 0111 (dieses weitverbreitete Polynom wird z. B. für Ethernet, PKZip, AUTODIN II und FDDI genutzt). Die berechnete CRC wird zurückgegeben.

Es wird eine Speicher-Kopierfunktion bereitgestellt. Diese kann vom NOR-Flash in das RAM oder vom RAM in das RAM beliebige Daten kopieren. Als Parameter werden der Funktion Zeiger auf das Ziel (destination) und die zu kopierende Quelle (source) übergeben, sowie die Menge (size) der zu übertragenden Daten.

3.4.2 Interaktiver Betriebsmodus

Nach der Initialisierung des Systems und nach Durchführung des POST werden folgende Funktionen verwendet.

3.4.2.1 Serielle Schnittstelle

Es wird eine Baudratenerkennung realisiert. Bei der Erkennung wird auf serielle Signale vom UART0 oder UART1 bzw. SCC1 oder SCC2 gewartet. Diese werden durch den Benutzer über ein Terminalprogramm erzeugt, wenn er mehrfach die Returntaste drückt. Bei der kleinsten Baudrate kann ein bis zu 6-maliges Drücken erforderlich sein. Zwischen jedem Drücken der Returntaste muss beim UART eine Pause von mindestens 3 ms liegen (dies ist in erster Linie für automatische Tests notwendig). Die Blockade der Schnittstelle durch einen dauerhaften "Mark"-Pegel (TTL: Low) wird durch einen Timeout verhindert.

Nach einer erfolgreichen Schnittstellenerkennung gibt der Monitor über die aktive serielle Schnittstelle eine Boot-Meldung ab, die folgende Informationen enthält:

- Name des Monitors (MON85XX)
- Revision des Monitorprogramms
- Typ des TQ-Moduls (TQM85XX)
- Copyright-Vermerk
- Kompilierdatum und -uhrzeit
- Status-Informationen zur Plattform (Taktfrequenzen der CPU, des CCB und des Local Bus)
- Status-Informationen zum Power-On Self Test (OK, Error, Skipped)

Programme und Daten können mittels des S-Record-Formats von Motorola über die serielle Schnittstelle auf das TQ-Modul übertragen, bzw. aus dem TQ-Modul ausgelesen werden. Beim Auslesen ist zu beachten, dass die Daten nur als S3-Record-Daten ausgegeben werden.

3.4.2.2 Kommandointerpreter

Nach Ausgabe der Boot-Meldung tritt der Monitor in die Kommandoschleife ein. In dieser werden über den Kommandointerpreter Kommandos und Parameter des Anwenders über die aktive Schnittstelle empfangen, ausgewertet und ausgeführt. Bei syntaktischen, semantischen oder Ausführungsfehlern wird eine entsprechende Fehlermeldung geliefert (siehe Tabelle 13: Rück-/Fehlermeldungen des Monitors nach Kommandoeingabe). Eine Beschreibung der Kommandos befindet sich in Kapitel 6 Benutzungsoberfläche.

Eine Auswahl von Memory-Mapped Special Function Register (CCSRs) des MPC85XX sind intern vordefiniert, und somit für den Anwender sofort durch ihre Symbolnamen (Freescale-ähnliche Registerbezeichnungen) verfügbar.

3.4.2.3 NOR-Flash Manipulationsfunktionen

Der Monitor bietet Funktionen zum Beschreiben und Löschen des Flashs. Um ein versehentliches Löschen oder Überschreiben des Monitorcodes im interaktiven Betriebsmodus durch den Anwender zu verhindern, ist ein Löschen der Flash-Sektoren, in denen sich der Monitorcode befindet, im Standardfall durch den Monitor nicht möglich. Diese Schutzfunktion kann durch ein Anwenderkommando mittels eines Passwortes aufgehoben werden.

Die Flash-Routinen des Monitors können mit allen Common Flash Memory Interface (CFI) konformen Flash-Typen des Herstellers SPANSION verwendet werden.

Hinweis:

SPANSION betont, dass statt einer Einzelwort-Programmierung die Write-Buffer Programmiermethode für die Typen AM29LV-M, S29GL-A und S29GL-M angewandt werden muss (siehe auch [3]). Der Monitor benutzt diesen Schreibmodus sofern das FLASH über einen Write Buffer verfügt.

Folgende Funktionen werden zur Verfügung gestellt:

- Auflösung des Herstellerschlüssels und des Typ-Identifikatorschlüssels in den Herstellernamen und die Bausteinbezeichnung des Herstellers.

Hinweis:

Der Monitor meldet noch vor Ausführung des POST einen Fehler falls die Typ-Indikatorschlüssel der Flash-Bausteine innerhalb einer Bank verschieden sind. Begründung: Ist eines der beiden Flashes defekt, kann der Monitor nicht aus dem Flash starten und somit den POST nicht ausführen. Wird der Monitor hingegen mit einem Hardware-Debugger aus dem RAM gestartet, kann die Fehlermeldung trotzdem im interaktiven Modus gelesen werden, obwohl der POST nicht ausgeführt wird.

- Ermittlung der Sektorenaufteilung und Sektorengrößen basierend auf den CFI-Informationen einer Flash-Bank.
- Löschung mehrerer Sektoren einer Flash-Bank. Als Parameter werden die Startadresse des ersten zu löschenden Sektors und die Endadresse des letzten zu löschenden Sektors übergeben.
- Löschung einer Flash-Bank. Als Parameter wird die Nummer der zu löschenden Flash-Bank übergeben.
- Byte-genaues Schreiben im Flash-Speicher. Als Parameter wird die Startadresse, wohin die Daten ins Flash geschrieben werden sollen, übergeben. Des weiteren benötigt die Funktion eine Adresse der zu kopierenden Daten und die Größe des zu kopierenden Datenblocks.

3.4.2.4 NAND-Flash Manipulationsfunktionen

Folgende Funktionen werden zum Beschreiben und Löschen des NAND-Flashs zur Verfügung gestellt. Weitere Details können den Beschreibungen der jeweiligen Kommandos entnommen werden (Kapitel 6.2).

- Anzeige der NAND-Flash-Eigenschaften.
Es werden Informationen zum Hersteller, Größe, Geometrie sowie eine Auflistung aller Blöcke angezeigt.
- Anzeige des Inhalts einer bestimmten NAND-Flash-Seite inklusive Spare Bytes. Es erfolgt keine Überprüfung der ECC. Es werden auch Seiten aus „bad blocks“ angezeigt.
- Kopieren von Daten vom RAM oder NOR-Flash ins NAND-Flash.
Es können nur ganze NAND-Flash-Seiten kopiert werden (2048 Bytes im 1x8 Bit Modus, 4096 Bytes im 2x8 Bit Modus).
Die Funktion berechnet die ECC für jede Seite und schreibt die ECC in das Spare Byte Array der betreffenden Seite. Vor dem Schreiben wird überprüft, ob der betreffende Block mit einem „bad marker“ gekennzeichnet ist. In diesem Fall wird in den nächsten Block geschrieben. Aus diesem Grund muss sich der Benutzer vorher mit dem Kommando NINFO über bekannte unbenutzbare Blöcke informieren um eine Reserve an freien Blocks berücksichtigen zu können.
Schlägt der Schreibversuch in einen als gut befundenen Block fehl (Überprüfung der Statusbits „Ready“ und „Programming“) wird der Block als schlecht markiert und der Programmiervorgang abgebrochen. In diesem Fall ist nicht bekannt, wie viele Blöcke voraussichtlich freigehalten werden müssen. Es wird eine Fehlermeldung ausgegeben.
- Kopieren von Daten vom NAND-Flash ins RAM.
Es können nur ganze NAND-Flash-Seiten kopiert werden. Die Funktion prüft, ob der betreffende Block als schlecht markiert wurde und überspringt ihn ggf. Vor dem Kopieren jeder Seite wird ihre ECC berechnet und, falls erforderlich, korrigiert. Handelt es sich um einen nicht korrigierbaren Fehler wird der Kopiervorgang abgebrochen. Es wird eine Fehlermeldung ausgegeben.
- Löschung mehrerer Blöcke.
Als Parameter wird die Nummer des ersten und letzten zu löschenden Blocks übergeben.
Es werden nur als „gut“ gekennzeichnete Blocks gelöscht, um die vom Hersteller gesetzten Bad-Blockmarkierungen zu erhalten. Dieser Schutz kann mit einem Parameter (siehe Kommando NERASE) umgangen werden.
- Löschen der Bad-Block-Tabelle.
Die Funktion löscht die Markierungen der im RAM gehaltenen Bad-Block-Tabelle, liest die durch den Hersteller erfolgten Markierungen aus dem NAND-Flash erneut ein und schreibt die neu erhaltene Tabelle in den dafür vom Monitor vorgesehenen Block des NAND-Flash zurück.

Hinweis:

Wear Levelling wird von NAND-Flash Treiber nicht unterstützt.

3.4.2.5 EEPROM Manipulationsfunktionen (I2C)

Der Monitor bietet Funktionen zum Beschreiben der modulinternen I2C-EEPROMs. Hierbei handelt es sich einerseits um das EEPROM, in dem die Konfigurationsdaten enthalten sind und andererseits um das optionale EEPROM, in dem Anwenderdaten hinterlegt werden können. Ein Schreiben auf das EEPROM mit den Konfigurationsdaten ist im interaktiven Betriebsmodus des Monitors standardmäßig nicht möglich. Diese Schutzfunktion kann durch ein Anwenderkommando mittels eines Passwortes aufgehoben werden.

Die Selektion der EEPROMs erfolgt über ihre I2C-Adresse:

- Das EEPROM für die Konfigurationsdaten besitzt die Adresse 0xAE
- Das EEPROM für die Anwenderdaten besitzt die Adresse 0xA0

Es werden folgende Funktionen zur Verfügung gestellt:

- Byte-genaues Schreiben im EEPROM. Als Parameter wird die I2C-Device-Adresse des Bausteins benötigt. Des weiteren benötigt die Funktion eine Adresse der zu kopierenden Daten, die Adresse, wohin die Daten im EEPROM geschrieben werden sollen, und die Größe des zu schreibenden Bereiches.
- Byte-genaues Lesen im EEPROM. Als Parameter wird die I2C-Device-Adresse des Bausteins benötigt. Des weiteren benötigt die Funktion eine Adresse, wohin die gelesenen Daten kopiert werden sollen, die Adresse, ab der die Daten aus dem EEPROM gelesen werden sollen und die Größe des zu lesenden Bereiches.

3.4.2.6 Temperatursensorfunktionen (I2C)

Der Monitor bietet eine Funktion zum Lesen der Temperatur vom modulinternen I2C-Temperatursensor LM75. Die Selektion des Temperatursensors erfolgt implizit über seine I2C-Device-Adresse (0x90). Der Temperatursensor befindet sich nach dem Power-Up im Komparatormodus. Die folgende Funktion geht von diesem Zustand aus.

Eine Konfiguration des Temperatursensors (Wechseln des Betriebsmodus [Shutdown, Comparator, Interrupt], Manipulation des Pointer-, Konfigurations-, T_{HYST} - und T_{OS} - Register) wurde nur soweit realisiert, um sicherzustellen, dass der Temperaturwert aus dem Sensor ausgelesen werden kann.

3.4.2.7 RTC Manipulationsfunktionen (I2C)

Der Monitor bietet Funktionen zum Lesen und Schreiben der modulinternen Real Time Clock (DS1337). Die Selektion der RTC erfolgt implizit über ihre I2C-Device-Adresse (0xD0). Die folgenden Funktionen wurden hierfür realisiert:

- Lesen des Datums und der Uhrzeit:

Als Parameter wird der Funktion ein String übergeben, in dem der Tag (DD), der Monat (MM), das Jahr (YY), der Stundenmodus (HMODE), die Stunden (HH), die Minuten (MM) und Sekunden (SS) im folgenden Format gespeichert werden:
YY/MM/DD HMODE HH:MM:SS

Beispiel für das Datum 14.10.2005 und die Uhrzeit 21:09:20 im 24 Stundenmodus:
05/10/14 21:09:20

Beispiel für das selbige Datum im 12 Stundenmodus:
05/04/14 PM 09:09:20

Ein Auslesen des Day-Wertes, des Century-Bits, der Alarmregister und des Statusregisters vom Sensor erfolgt nicht.

- Schreiben des Datums und der Uhrzeit:

Als Parameter werden der Funktion Tag, Monat, Jahr, Stundenmodus (12 für 12-Stundenmodus, 24 für 24-Stundenmodus), Stunden, Minuten und Sekunden übergeben. Hierbei wird nur überprüft ob sich die einzelnen Parameter innerhalb der erlaubten Grenzen befinden und ob sie erfolgreich an die RTC übertragen werden konnten. Es erfolgt keine Überprüfung ob die gelieferten Parameter einem korrekten Datum entsprechen. Somit können inkorrekte Datumswerte zu einem undefinierten Verhalten der RTC führen.

Ein Beschreiben der Alarmregister für INTA, INTB und des Kontrollregisters erfolgt nicht.

4. Produktdaten

4.1 Hardware Information Block (HWIB)

Im modulinternen NOR-Flash ist der Hardware Information Block (HWIB) gespeichert. Dieser enthält folgende Parameter:

- Artikel-Kurzbezeichnung (Dekodierung siehe 3.1.1 und [2])
- Seriennummer (dezimal)
- erste vergebene MAC-Adresse
- Anzahl der weiteren MAC-Adressen

Die Daten des HWIB sind in dem für sie reservierten NOR-Flash-Bereich gespeichert (siehe Tabelle 3). Die einzelnen Parameter werden durch jeweils ein Leerzeichen getrennt, und der restliche Speicherbereich wird mit binär 0 aufgefüllt (siehe Abbildung 5: Beispiel eines Speicherauszugs des HWIB).

Hardware Information Block		
Adresse	Speicherbereich dargestellt in Hexadezimal	in ASCII
81FC0000:	54 51 4D 38 35 36 30 44 43 42 41 44 37 2D 41 50	TQM8560DCBAD7-AP
81FC0010:	46 42 43 2E 30 32 30 31 20 31 30 39 39 31 32 31	FBC.0201 1099121
81FC0020:	31 20 30 30 44 30 39 33 30 33 44 32 37 30 20 34	1 00D09303D270 4

Abbildung 5: Beispiel eines Speicherauszugs des HWIB für ein TQM8560-AA

4.2 Configuration Information Block (CIB)

Im modulinternen NOR-Flash ist der Configuration Information Block gespeichert. Dieser enthält folgende Parameter:

Parameter	Abkürzung	Bedeutung	Einheit	Typ
-cb	Clock BUS	Bustakt (Local Bus)	Hz	2
-cc	Clock CCB/CPM	CCB-Takt	Hz	1
-cu	Clock CPU	CPU-(Core-)-Takt	kHz	2
-td	Timing DDR1-SDRAM	CAS Latency Timing DDR1-SDRAM	10-facher Wert	3
-tf	Timing Flash	NOR-Flash-Timing (Registerwert)	32 Bit	3

Tabelle 7: Parameter des CIB

Tipps:

*1 Falls der Parameter nicht im CIB vorhanden ist, wird der gemessene Wert (Referenztimer) benutzt

*2 Falls der Parameter nicht im CIB vorhanden ist, wird er mit Hilfe des gemessenen CCB Clocks berechnet.

*3 Falls der Parameter nicht im CIB vorhanden ist, wird ein Default-Timing an seiner Stelle benutzt. Bei DDR2 wird der Parameter -td derzeit ignoriert.

Alle o.g. Parameter müssen im CIB enthalten sein, damit der POST erfolgreich abgeschlossen wird. Für den interaktiven Betriebsmodus sind diese Parameter nicht notwendig, da sie durch Messungen zur Laufzeit ermittelt werden können. Trotzdem erfolgt eine POST-Fehlermeldung.

Die Dekodierungsvorschrift (BNF) des CIB-Strings wird in Kapitel 3.1.2 beschrieben.

Die Daten des CIB sind in dem für sie reservierten NOR-Flash-Bereich gespeichert (siehe Tabelle 3). Die einzelnen Parameter werden durch jeweils ein einzelnes Leerzeichen getrennt, und der restliche Speicherbereich wird mit binär 0 aufgefüllt (siehe Abbildung 6: Beispiel eines Speicherauszugs des CIB).

MON85XX.SWM.106.doc

Configuration Information Block		
Adresse	Speicherbereich dargestellt in Hexadezimal	in ASCII
81FC0200:	2D 74 64 20 32 35 20 2D 63 75 20 38 33 33 33 33	-td 25 -cu 83333
81FC0210:	33 20 2D 63 62 20 38 33 33 33 33 33 33 20 2D	3 -cb 83333333 -
81FC0220:	74 66 20 30 78 30 30 30 30 43 35 34 20 2D 63	tf 0x0000C54 -c
81FC0230:	63 20 33 33 33 33 33 33 33 33 33 00 00 00 00	c 333333333.....

Abbildung 6: Beispiel eines Speicherauszugs des CIB für ein TQM8560-AA

4.3 Applikationsvorspann

Der Applikationsvorspann ist 256 Bytes groß und enthält folgende Parameter:

Startadresse	Größe	Bezeichnung	Inhalt
0x80000000	28 Bytes	ID-String	String, der die Art des Autostarts festlegt
0x8000001C	32 Bit	Start	Startadresse der Applikation
0x80000020	32 Bit	Size	Größe des Bereichs in Bytes, der ins RAM kopiert werden soll
0x80000024	32 Bit	Crc32	CRC-Summe über den Applikationsbereich
0x80000028	32 Bit	Flags	Software-Schalter
0x8000002C	212 Bytes		Reservierter Bereich für zukünftige Erweiterungen

Tabelle 8: Inhalt des Applikationsvorspanns

Die ersten vier Einträge (40 Bytes) stellen die Parameter für den Autostart dar. Hierbei legt der ID-String fest wie der Monitor die Applikation starten soll. Die Parameter „Start“, „Size“ und „Crc32“ werden abhängig vom ID-String ausgewertet. Die unterschiedlichen Autostartmöglichkeiten sind aus Kapitel 3.4.1 Initialer Betriebsmodus ersichtlich.

Der fünfte Eintrag enthält Software-Schalter (Bitfeld mit 32 Flags). Mit diesen Schaltern können vom Benutzer verschiedene Funktionen des Monitors aktiviert, bzw. deaktiviert werden. Die Lage der einzelnen Schalter ist aus Tabelle 9: Lage Software Schalter ersichtlich, ihre Funktion aus Tabelle 10: Funktionsübersicht Software Schalter.

©2007 by TQ-Components GmbH

Bit	0	1	2	3	4	5	6	7
Flag	<resvd>	POST1EN	POST2EN	BAUD- DETECTEN	<resvd>	<resvd>	<resvd>	<resvd>
Reset	1	1	1	1	1	1	1	1
R / W	R / W							
Addr.	0x80000028							
Bit	8	9	10	11	12	13	14	15
Flag	<resvd>	<resvd>	<resvd>	<resvd>	<resvd>	<resvd>	<resvd>	<resvd>
Reset	0xFF							
R / W	R / W							
Addr.	0x80000029							
Bit	16	17	18	19	20	21	22	23
Flag	<resvd>	<resvd>	<resvd>	<resvd>	<resvd>	<resvd>	<resvd>	<resvd>
Reset	0xFF							
R / W	R / W							
Addr.	0x8000002A							
Bit	24	25	26	27	28	29	30	31
Flag	<resvd>	<resvd>	<resvd>	<resvd>	<resvd>	<resvd>	<resvd>	<resvd>
Reset	0xFF							
R / W	R / W							
Addr.	0x8000002B							

Tabelle 9: Lage Software Schalter

Bit	Flag	Description
0	-	Reserved, should be set.
1	POST1EN	POST 1 Enable. Determines whether POST 1 will be executed 0: No POST 1 1: POST 1 executed (default)
2	POST2EN	POST 2 Enable. Determines whether POST 2 will be executed 0: No POST 2 1: POST 2 executed (default)
3	BAUDDTECTEN	Port- and Baudrate-Detection Enable. Determines whether Monitor automatically detects serial port and baudrate. This flag is evaluated only in interactive mode. 0: UART0/SCC1 is initialized at 115200 Baud. The Monitor's boot message is visible immediately after initialization. UART1/SCC2 cannot be used with the Monitor. 1: Serial port as well as baudrate is automatically detected when the user presses the return key several times (default)
4 - 31	-	Reserved, should be set.

Tabelle 10: Funktionsübersicht Software Schalter

Hinweis zum Setzen / Rücksetzen der Software-Schalter:

Da sich die Software-Schalter in einem Flash befinden, können sie zwar durch einen einfachen Flash-Schreibvorgang von 1 auf 0 gesetzt werden, ein Rücksetzen von 0 auf 1 kann aber nur dadurch erreicht werden, dass der gesamte Flash-Sektor, in dem sich die Software-Schalter befinden, vom Benutzer gelöscht wird.

5. Produkteleistungen

Die folgenden Daten gelten für ein Modul mit dem Ordercode TQM8560-AA:

Antwortzeiten auf ein Kommando betragen nicht mehr als 1 Sekunde. Als Antwortzeit gilt die Zeit vom Absenden eines Kommandos bis zur ersten sichtbaren Reaktion.

Die serielle Übertragungsrate beim Download ins SDRAM beträgt durchschnittlich 10 KiB/s bei 115200 Baud (PC mit 2,8 GHz CPU, 256 MiB RAM, MS Windows 2000). Bei einem Download ins Flash darf die Übertragungsrate aufgrund der Programmierzeiten der Flashes geringer sein.

Die Zeit vom Ende des Reset (HRESET# steigende Flanke) bis zum Einsprung in die Applikation kann zwischen den unterschiedlichen Modulausbaustufen variieren. Für das angegebene Modul beträgt die Zeit ca. 320 ms.

6. Benutzungsoberfläche

6.1 Kommandoeingabe

Die Kommandoeingabe im interaktiven Betriebsmodus erfolgt über die serielle Schnittstelle als zeilenorientierte Kommunikation durch Befehle (inklusive Parameter) als Klartext im ASCII-Format:

- Es wird bei der Eingabe nicht zwischen Groß- und Kleinschreibung unterschieden, außer bei den Parametern des SETCI-Kommandos.
- Jede Kommandozeile beginnt mit einem Kommando, bei Bedarf gefolgt von Parametern (abhängig vom Kommando). Das Kommando REG bildet eine Ausnahme. Hier kann das eigentliche Kommando weggelassen werden.
- Parameter können Zahlen, Schlüsselwörter, Registernamen oder Strings sein. Zahlen werden grundsätzlich hexadezimal interpretiert. Eine weitere Kennzeichnung wie führendes „0x“ oder angehängtes „h“ ist zulässig. Eine Ausnahme bilden die Kommandos SETCI, SETHWI und PROTECT: Zahlenparameter werden hier als (Teil-)Strings interpretiert.
- Jede Eingabe an der Kommandozeile muss mit Carriage-Return abgeschlossen werden, damit die Ausführung gestartet wird. Vor dem Abschluss durch Carriage-Return kann das Kommando mit Control-C abgebrochen werden. Es erscheint ein neuer Prompt.
- Die maximal eingebare Anzahl an Zeichen nach dem Prompt beträgt 206. Damit können z.B. beim WRITE-Kommando mit einer achtstelligen Hexadresse bis zu 64 einzelne Bytes (zweistellig, Hex), getrennt durch Leerzeichen, eingegeben werden.
- Während der Abarbeitung wird das Steuerzeichen Control-C als Funktionsabbruch für die Kommandos SAVE, READ, EREAD, REG (Nur Ausgabe der Registerliste), DEF (Nur Ausgabe der Registerliste) und MEMORY FLASH erkannt.

Hinweis:

Wird durch ein Kommando versucht, auf einen nicht verfügbaren Speicherbereich zuzugreifen, so ist eine einwandfreie Ausführung des Kommandos nicht gewährleistet. Die CPU und damit der Monitor blockiert. Dieser Zustand kann nur durch einen HRESET aufgelöst werden.

Die Tabelle 11: Beschreibung der Kommandosyntax enthält eine Beschreibung der zur Darstellung der Kommandosyntax verwendeten Operatoren und Ausdrücke.

Ausdruck	Bedeutung
GROSS	Großgeschriebene Wörter entsprechen Kommando-Schlüsselwörtern. Sie müssen wie dargestellt vom Benutzer eingegeben werden. Die Groß- und Kleinschreibung der Kommando-Schlüsselwörter ist hierbei ohne Bedeutung.
parameter	Kleingeschriebene Wörter sind als Platzhalter für Parameter zu interpretieren. An ihrer Stelle sind die Parameter vom Benutzer einzugeben.
()	Klammerung von Ausdrücken um die Gültigkeit von [], und ... festzulegen
[]	Optionale Benutzereingaben werden durch eckige Klammern umschlossen. Die Klammern sind nicht durch den Benutzer einzugeben.
	Alternative Benutzereingaben werden durch den senkrechten Trennstrich gekennzeichnet. Auch der Trennstrich ist nicht durch den Benutzer einzugeben.
...	Drei aufeinanderfolgende Punkte deuten eine Wiederholungsmöglichkeit an. Der Benutzer kann an Stelle dieser Punkte eine beliebige Anzahl des vorher abgebildeten Ausdrucks eingeben.

Tabelle 11: Beschreibung der Kommandosyntax

6.2 Anwenderkommandos

In diesem Abschnitt werden alle Produktfunktionen aufgeführt, die durch den Anwender aufrufbare Benutzerkommandos des Monitors darstellen. Zu jedem Kommando wird neben seiner Funktionalität auch die Syntax des Kommandos erklärt und ein Benutzungsbeispiel gegeben.

6.2.1 COPY – Kopieren von Speicherbereichen

Durch dieses Kommando kann der Benutzer zusammenhängende Speicherbereiche eines Mediums (z.B. EEPROM) mit einem Kommandoaufruf in ein anderes Medium (z.B. RAM) kopieren. Von folgenden Medien kann kopiert werden:

- EEPROM
- NOR-Flash
- SDRAM

Die Selektion des EEPROMs erfolgt entweder über seine I2C-Bausteinadresse (7Bit-Identifizier) oder über implizit festgelegte Symbole für die modulinternen EEPROMs. Die Selektion von Flash und SDRAM erfolgt über das Symbol MEM. Das Kommando unterscheidet implizit aufgrund der angegebenen Adressen ob es sich beim Ziel und der Quelle um ein SDRAM oder Flash handelt.

Jedes EEPROM besitzt einen eigenen Adressraum, somit beginnt der Datenraum des EEPROMs jeweils mit der Adresse 0.

Zusätzlich gibt es noch die Teilfunktion Monitor → Flash, welche es ermöglicht, den aktuell laufenden Monitor ins Flash zu schreiben. Dabei erfolgt eine spezielle Behandlung der Global Offset Table (GOT), um die beim Start des Monitors erfolgte Relokation rückgängig zu machen.

Hinweise:

- Falls Daten ins Flash kopiert werden sollen, muss der entsprechende Flash-Bereich durch den Anwender vorher gelöscht worden sein.
- Beim Versuch, mehr Daten in die EEPROMs zu schreiben als Platz ist, werden nur die Daten hineingeschrieben, welche noch hineinpassen. Anschließend wird die Funktion mit einer Fehlermeldung beendet.

Syntax des Kommandos

COPY MONITOR | ((DATA | CONF | [MEM]) start end (device | DATA | CONF | [MEM]) target)

Parameter	Bedeutung des Parameters
DATA	Symbolischer Ausdruck für das optionale EEPROM, das Anwenderdaten enthält
CONF	Symbolischer Ausdruck für das EEPROM, das die TQ-Modul-Konfigurationsdaten enthält
MONITOR	Symbolischer Ausdruck, um den Monitor vom SDRAM ins Flash zu kopieren
MEM	Symbolischer Ausdruck für den Speicher (Flash oder SDRAM, je nach Adresse)
start	Startadresse der Datenquelle, die kopiert werden soll (hexadezimaler 32-Bit-Wert)
end	Letzte Adresse der Datenquelle, die kopiert werden soll (hexadezimaler 32-Bit-Wert)
target	Zieladresse, hexadezimaler 32-Bit-Wert

Eingabebeispiel	Bedeutung des Beispiels
copy mem 10000 100ff data 10	Bereich des SDRAMs von Adresse 0x10000 bis 0x100ff nach 0x10 in das EEPROM für die Anwenderdaten kopieren
copy conf 0 6 mem 10000	Bereich des Konfigurations-EEPROMs von Adresse 0 bis 0x6 ins SDRAM ab Adresse 0x10000 kopieren
copy data 10 20 20000	Bereich des Anwenderdaten-EEPROMs von Adresse 0x10 bis 0x20 ins SDRAM ab Adresse 0x20000 kopieren
copy 10000 100ff 20000 copy monitor	Bereich von Adresse 0x10000 bis 0x100ff nach 0x20000 kopieren Kopieren des sich im SDRAM befindenden Monitors ins Flash

6.2.2 CRC – CRC berechnen

Mit diesem Kommando ist es dem Benutzer möglich, sich über einen frei wählbaren Speicherbereich eine CRC-32 berechnen zu lassen. Als Ergebnis wird die berechnete CRC über den Speicherbereich ausgegeben. Falls der optionale Parameter crc32 eingegeben wurde, erfolgt nach der Berechnung noch ein Vergleich dieses Parameters mit der berechneten CRC. Die Funktion gibt einen Ausführungsstatus aus. Stimmt die berechnete CRC mit der eingegebenen CRC nicht überein, wird die berechnete zusätzlich ausgegeben.

Syntax des Kommandos
CRC start length [crc32]

Parameter	Bedeutung des Parameters
start	Startadresse des zu überprüfenden Bereiches (hexadezimaler 32-Bit-Wert)
length	Länge des zu überprüfenden Bereiches (hexadezimaler 32-Bit-Wert)
crc32	Sollwert, mit dem die CRC-Berechnung verglichen wird

Eingabebeispiel	Bedeutung des Beispiels
crc 80000100 1000	Berechnung der CRC über 0x1000 Bytes ab Adresse 0x80000100 mit anschließender Ausgabe der CRC
crc 80000100 1000 A7383D44	wie oben, jedoch mit anschließendem Sollwertvergleich und Ergebnisausgabe

6.2.3 DEF – Benutzerdefinierte Register

Mit diesem Kommando ist es dem Benutzer möglich, sich für Speicherbereiche (NOR-Flash, RAM, DPRAM und speicher-gemappte Prozessorregister) eigene Registernamen zu definieren und über diese Definitionen auf die Speicherbereiche / Prozessorregister mittels des REG-Kommandos zuzugreifen. Es werden folgende Teilfunktionen unterstützt:

- Neues Register definieren
- Definition löschen
- Definition ausgeben
- Alle Definitionen ausgeben

Hinweise:

- Es können durch den Anwender auch benutzerdefinierte Register angelegt werden, die den gleichen Namen wie die CCSRs besitzen. Eine Benutzung dieser Register ist aber nicht möglich. Es wird anstelle des gleichlautenden benutzerdefinierten Registers immer auf das CCSR zugegriffen.
- Es können maximal 100 eigene Register definiert werden.
- Der Name des Registers kann bis zu 10 Zeichen lang sein.

Syntax des Kommandos

DEF [name [=] [addr [L | S | B]]]

Parameter	Bedeutung des Parameters
name	Zu definierender Registername (alphanumerisch, max. 10 Zeichen)
addr	Zu definierende Registeradresse (hexadezimaler 32-Bit-Wert)
[L S B]	Registerbreite, L = long (32 Bit), S = short (16 Bit), B = byte (8 Bit); Default ist 32 Bit

Eingabebeispiel	Bedeutung des Beispiels
def foo = c0000000 L	Definieren des Registers „FOO“ an Adresse 0xC0000000 mit Breite 32 Bit
def x1 12345678	Definieren des Registers „X1“ an Adresse 0x12345678 mit Default-Breite 32 Bit
def foo c0000000 b	Umdefinieren des Registers „FOO“ auf Adresse 0xC0000000 und Breite 8 Bit
def foo	Anzeigen der Definition für Register „FOO“
def foo =	Löschen der Registerdefinition „FOO“
def	Anzeigen der Definitionen aller benutzerdefinierten Register (seitenweise)

6.2.4 ECHO – Ein-/Ausschalten des Echo

Mit diesem Kommandos wird das Echo-Verhalten des Monitors gesteuert. Befindet sich der Monitor im Echo-Modus (ECHO ON = Default-Einstellung), so werden alle empfangenen Zeichen durch den Monitor über die serielle Schnittstelle an das Terminalprogramm zurückgesendet. Befindet sich der Monitor im Echo-Modus OFF so unterbleibt dieses Zurücksenden. Die Ausgaben des Monitors (Prompt, Fehlermeldungen, Ausgaben im Zuge von Kommandoausführungen etc.) bleiben hiervon unberührt.

Syntax des Kommandos

ECHO [ON | OFF]

Parameter	Bedeutung des Parameters
ON	Einschalten des Echos
OFF	Ausschalten des Echos

Eingabebeispiel	Bedeutung des Beispiels
echo off	Echo ausschalten
echo	Ausgabe der aktuellen Einstellung des Echo-Modus

6.2.5 ERASE – Löschen von NOR-Flash-Sektoren

Mit Hilfe dieses Kommandos werden einzelne oder mehrere Sektoren, einzelne Flash-Speicherbänke oder der gesamte vorhandene Flash-Bereich gelöscht. Der für den Monitor reservierte Bereich ist dabei geschützt, sofern der Schutz nicht vorher aufgehoben wurde (Kommando PROTECT).

Die maximale Zeit für den Löschvorgang wird durch die CFI-Information der Flash-Bausteine festgelegt. Dieser Timeout wird durch den Monitor aus Sicherheitsgründen noch um 20 % erhöht. Wird diese Zeit (Fehlerfall = maximale Ausführungsdauer) beim Löschen eines Flash-Bereichs überschritten, so wird das Löschen abgebrochen und eine Fehlermeldung ausgegeben.

Hinweis:

Verfügt ein TQ-Modul nur über eine Flash-Bank, so ist das Kommando "Bank löschen" und "Gesamtes Flash löschen" äquivalent.

Syntax des Kommandos
ERASE (start end) (BANK (0 1)) ALL

Parameter	Bedeutung des Parameters
start	Startadresse des ersten Sektors, der gelöscht werden soll (hexadezimaler 32-Bit-Wert)
end	Endadresse des letzten Sektors, der gelöscht werden soll (hexadezimaler 32-Bit-Wert)
BANK (0 1)	Wählt entweder Bank 0 oder Bank 1 für das bankweise Löschen aus
ALL	Wählt den kompletten Flash-Speicher (nur ungeschützte Sektoren) zum Löschen aus

Eingabebeispiel	Bedeutung des Beispiels
erase 80000000 8005FFFF	Inhalte von mehreren Sektoren werden gelöscht
erase bank 0	Inhalte der Flash-Bank 0 werden gelöscht
erase all	Inhalte des gesamten Flash-Speichers werden gelöscht

6.2.6 ERead – Daten aus einem EEPROM auslesen

Mit diesem Kommando können Speicherbereiche eines EEPROMs (unterstützte Typen siehe Kapitel 2.2), das am I2C-Bus angeschlossen ist, als Dump über die serielle Schnittstelle ausgegeben werden. Die Ausgabe der Daten erfolgt sowohl im Hexadezimalformat als auch im ASCII-Format. Zur besseren Lesbarkeit der Ausgabe werden nur 16 Datenbytes pro Zeile ausgegeben. Jeder Datenzeile wird als Identifikation die Adresse (32Bit) vorangestellt, aus der die Datenzeile im Speicher stammt. Die Selektion des EEPROMs erfolgt entweder über seine I2C-Bausteinadresse (7Bit-Identifizier) oder über implizit festgelegte Symbole für die modulinternen EEPROMs.

Jedes EEPROM besitzt einen eigenen Adressraum, somit beginnt der Datenraum des EEPROMs jeweils mit der Adresse 0.

Das Kommando kann durch Eingabe der Tastenkombination Control-C unterbrochen werden.

Syntax des Kommandos
ERead (device DATA CONF) [start [end]]

Parameter	Bedeutung des Parameters
device	I2C-Bausteinadresse des EEPROMs (hexadezimaler 8-Bit-Wert)
DATA	Symbolischer Ausdruck für das optionale EEPROM, das Anwenderdaten enthält. Wurde im HWIB keine oder eine ungültige Größenangabe oder Größe = 0 angegeben, wird eine Fehlermeldung ausgegeben.
CONF	Symbolischer Ausdruck für das EEPROM, das die TQ-Modul Konfigurationsdaten enthält
start	Startadresse, ab der die Daten aus dem EEPROM ausgegeben werden sollen (hexadezimaler 32-Bit-Wert). Ist die Startadresse größer oder gleich der EEPROM-Größe werden die Inhalte der letzten beiden Bytes des EEPROMs ausgegeben.
end	Endadresse - letzte auszugebende Adresse (hexadezimaler 32-Bit-Wert) Falls der Parameter fehlt wird als Default-Endadresse die Startadresse + 0xFF (Länge 256 Bytes) angenommen. Ist die Endadresse größer oder gleich der EEPROM-Größe wird die Endadresse auf das letzte Byte des EEPROMs gesetzt.

Eingabebeispiel	Bedeutung des Beispiels
eread a0 100 120	Speicherinhalt des EEPROMs mit der I2C-Adresse 0xA0 von Adresse 0x100 bis 0x120 ausgeben
eread data 100	Speicherinhalt des EEPROMs für Anwenderdaten von Adresse 0x100 bis 0x1FF ausgeben
eread conf	Speicherinhalt des EEPROMs für Konfigurationsdaten von Adresse 0x0 bis 0x7F ausgeben

6.2.7 EWRITE – Daten in ein modulinternes EEPROM schreiben

Mit diesem Kommando kann der Benutzer direkt byteweise Daten in ein EEPROM (unterstützte Typen siehe 2.2), das am I2C-Bus angeschlossen ist, schreiben. Die Daten können dem Kommando als Sequenz von Strings oder hexadezimalen Werten oder einer Kombination aus beiden übergeben werden. Die Selektion des EEPROMs erfolgt entweder über seine I2C-Bausteinadresse (7Bit-Identifizier) oder über implizit festgelegte Symbole für die modulinternen EEPROMs.

Jedes EEPROM besitzt einen eigenen Adressraum, somit beginnt der Datenraum des EEPROMs jeweils mit der Adresse 0.

Hinweis:

Beim Versuch, mehr Daten zu schreiben als Platz im EEPROM ist, werden nur die Daten hineingeschrieben, welche noch hineinpassen. Anschließend wird die Funktion mit einer Fehlermeldung beendet.

WARNUNG:

EINE ÄNDERUNG DES INHALTS IM KONFIGURATIONS-EEPROM ERFOLGT AUF EIGENE GEFAHR. FÜR SCHÄDEN AM MODUL SOWIE ANGESCHLOSSENER KUNDENSPEZIFISCHER HARDWARE, DIE DURCH FALSCHES POWER-ON RESET KONFIGURATIONS DATEN HERVORGERUFEN WERDEN, ÜBERNIMMT TQ-COMPONENTS KEINE HAFTUNG!

Syntax des Kommandos

EWRITE (device | DATA | CONF) addr (value | string) [...]

Parameter	Bedeutung des Parameters
device	I2C-Bausteinadresse des EEPROMs (hexadezimaler 8-Bit-Wert)
DATA	Symbolischer Ausdruck für das optionale EEPROM, das Anwenderdaten enthält. Wurde im HWIB keine oder eine ungültige Größenangabe oder Größe = 0 angegeben, wird eine Fehlermeldung ausgegeben.
CONF	Symbolischer Ausdruck für das EEPROM, das die TQ-Modul-Konfigurationsdaten enthält
addr	Adresse, ab der die Daten ins EEPROM geschrieben werden (hexadezimaler 32-Bit-Wert)
value	Zu schreibende(r) Wert(e) (hexadezimal, 8 Bit)
string	Zu schreibende(r) Wert(e) im String-Format. Folgende Escape-Sequenzen werden erkannt: \a == 0x07 ASCII BEL (Bell) \b == 0x08 ASCII BS (Backspace) \f == 0x0C ASCII FF (Form Feed) \r == 0x0D ASCII CR (Carriage Return) \t == 0x09 ASCII HT (Horizontal Tab) \v == 0x0B ASCII VT (Vertical Tab) \n == 0x0A ASCII LF (Line Feed) \0 == 0x00 ASCII NUL (Binary Null) Alle anderen auf ein '\' folgenden Zeichen, einschließlich '"' (Anführungszeichen) und '\' selbst werden 1:1 geschrieben Eine terminierende Null wird <u>nicht</u> automatisch an den String angehängt.

Eingabebeispiel	Bedeutung des Beispiels
ewrite a0 100 af fe ca fe 01 23	Sechs Bytes ab Adresse 0x100 in das EEPROM mit der I2C-Bausteinadresse 0xA0 schreiben
ewrite data 200 "Hello"	String "Hello" ohne termin. Null ab Adresse 0x200 schreiben
ewrite conf 50 "\" "\"	Anführungszeichen und Backslash in das EEPROM mit den TQ-Konfigurationsdaten an Adresse 0x50 schreiben

6.2.8 GO – Starten von Programmen

Durch dieses Kommando kann ein beliebiges Programm vom Monitor angesprungen und somit gestartet werden. Um unerwünschte Interrupts im gestarteten Programm zu unterbinden, werden vor dem Einsprung die MPC Core External Exceptions, der System Timer, der I2C-Controller und die serielle Schnittstelle deaktiviert.

Wird anstelle der Startadresse dem Kommando das Symbol AUTOSTART übergeben, wird eine Applikation abhängig von den Informationen im Applikationsvorspann angesprungen. Hierdurch wird das gleiche Verhalten erzeugt wie im Initialmodus (siehe Kap 3.4.1 Initialer Betriebsmodus).

Wird der Befehl ohne Angabe einer Startadresse aufgerufen, wird die Startadresse aus den Informationen des letzten S-Record-Downloads ermittelt (inklusive angegebenen Offset). Falls aber vorher kein Download stattgefunden hat, ist diese Kommandoabkürzung unzulässig und es wird eine Fehlermeldung hierzu ausgegeben.

Dem Programm, welches vom Monitor gestartet wird, werden noch wichtige System-Informationen als Parameter mittels Register übergeben (siehe Kapitel 3.4.1 Initialer Betriebsmodus). Im Register R3 wird dabei als startType immer „Kaltstart“ übergeben.

Syntax des Kommandos

GO [AUTOSTART] | [addr]

Parameter	Bedeutung des Parameters
addr	Startadresse des Programms
AUTOSTART	Startadresse, die durch den Applikationsvorspann definiert ist. Ist der Applikationsvorspann ungültig, wird die Default-Startadresse 0x80000100 angesprungen.

Eingabebeispiel	Bedeutung des Beispiels
go 80000100	Programm mit Einsprungadresse 0x80000100 starten
go autostart	Starten eines Programms abhängig vom Applikationsvorspann
go	Programm mit der Einsprungadresse des letzten S-Record Downloads starten

6.2.9 HELP – Hilfe-Funktion zu den Anwenderkommandos

Mit diesem Kommando kann der Benutzer sich entweder eine Kommandoübersicht oder zu einem bestimmten Kommando eine Hilfe über die serielle Schnittstelle ausgegeben lassen.

Der Hilfstext zu einem Kommando enthält folgende Informationen:

- Beschreibung der Kommandofunktionalität
- Kurzinfo zu den verwendeten Parameter

Alle ausgegebenen Informationen werden in der englischen Sprache ausgegeben.

Syntax des Kommandos

HELP [command]

Parameter	Bedeutung des Parameters
command	Ausdruck für das Kommando, zu dem eine Hilfestellung ausgegeben werden soll

Eingabebeispiel	Bedeutung des Beispiels
help def	Ausgabe der Hilfestellung für das Kommando "def"
help	Kurzübersicht zu allen verfügbaren Kommandos des Monitors

6.2.10 I2CDEV – Anzeigen der modulinternen I2C-Geräte

Mit diesem Kommando werden Informationen zu den angeschlossenen I2C-Geräten über die serielle Schnittstelle ausgegeben.

Es handelt sich hierbei um folgende Informationen:

- Art des I2C-Gerätes (z.B. RTC, Temperatursensor oder EEPROM)
- I2C-Bausteinadresse des Gerätes
- Zusatzinfo abhängig von der Art des Gerätes (z.B. Uhrzeit, Temperatur, EEPROM-Größe). Die Größe des Anwender-EEPROMs wird aus dem HWIB, sofern vorhanden, ermittelt. Andernfalls wird in der Default-Variante des Kommandos „User data EEPROM not found“ und in der Scan-Variante der Wert „unknown“ angezeigt. Das Konfigurations-EEPROM besitzt eine festgelegte Größe.

Das Kommando kann in zwei Varianten benutzt werden:

- In der Variante ohne Parameter werden nur die während des POST erkannten modulinternen I2C-Geräte ausgegeben.
- In der Scan-Variante wird ein Polling (Bausteinadresse) auf dem I2C-Bus eingeleitet, bei dem auch unbekannte (modulexterne Geräte) detektiert werden. Das Ergebnis des Pollings wird dann ausgegeben.

Hinweis:

Für das Polling wird ein Byte vom I2C-Gerät gelesen. Führt das Gerät ein internes Pointer-Register o.ä. (wie z.B. die EEPROMs oder die RTC), wird dieses durch das Lesen verändert! Dies gilt auch für angeschlossene kundenspezifische I2C-Geräte!

Syntax des Kommandos	
I2CDEV [SCAN]	

Parameter	Bedeutung des Parameters
SCAN	Durchführen des Pollings auf dem I2C-Bus mit anschließender Ausgabe der Ergebnisse

Eingabebeispiel	Bedeutung des Beispiels
i2cdev	Ausgabe der Info über alle vorhandenen modulinternen I2C-Geräte
i2cdev scan	Scannen und Ausgeben der Info über alle vorhandenen I2C-Geräte

6.2.11 LOAD – Herunterladen von S-Record-Daten auf das TQ-Modul

Mit diesem Kommando können Daten oder Programme, welche im Motorola S-Record-Format vorliegen müssen, über die serielle Schnittstelle auf das TQ-Modul heruntergeladen werden.

Es werden folgende Record-Typen unterstützt:

- S0 (Modul-Header)
- S1 bis S3 (Daten)
- S7 bis S9 (Startadresse)

Der Typ S5 (Zähler) wird ignoriert. Beim Laden kann zu den im S-Record-File enthaltenen Adressen ein Offset addiert werden. Durch folgende Ereignisse wird das Herunterladen abgebrochen:

- Es wurde versucht, einen fehlerhafter S-Record herunterzuladen
- Es wurde versucht, den S-Record in einen geschützten Speicherbereich zu schreiben
- Es wurde durch den Benutzer während des Herunterladens die Tastenkombination Control-C übertragen

Die S-Record-Daten können sowohl ins NOR-Flash als auch in das SDRAM heruntergeladen werden. Vor dem Laden ins Flash müssen die entsprechenden Sektoren ggf. gelöscht werden.

Hinweis:

Wird durch das LOAD-Kommando versucht, auf einen nicht verfügbaren Speicherbereich zuzugreifen, siehe Kap. 6.1 Kommandoeingabe.

Syntax des Kommandos

LOAD offset (FLASH | F | RAM | R)

Parameter **Bedeutung des Parameters**

offset	Hexadezimaler 32-Bit-Wert, der zu den im S-Record enthaltenen Adressen addiert wird
FLASH F	Die herunterzuladenden Daten werden im moduleigenen Flash abgelegt
RAM R	Die herunterzuladenden Daten werden im moduleigenen SDRAM abgelegt

Eingabebeispiel **Bedeutung des Beispiels**

load 1000000 r	Es werden S-Record-Daten ins RAM heruntergeladen. Zu der im S-Record implizit vorhandenen Adressierung wird jeweils ein Offset von 0x1000000 hinzuaddiert.
----------------	--

6.2.12 MEMORY – Anzeigen des aktuellen Speicher-Mappings

Durch dieses Kommando wird die Größe und das Mapping aller auf dem Modul vorhandenen Speicher (NOR-Flash, NAND-Flash und SDRAM) mit Ausnahme der seriellen EEPROMs ausgegeben. Zusätzlich werden die vom Monitor automatisch gewählten Frequenzen für das SDRAM-Timing, das SDRAM-Refresh-Interval und NAND-UPM-Timing angezeigt, wenn der Monitor nicht vom Debugger gestartet wurde.

Durch den optionalen Parameter „FLASH“ werden folgende Common Flash Memory Interface (CFI) Informationen zum NOR-Flash-Speicher ausgegeben:

- Query Identification String (Vendor Command Set und Control Interface ID Code)
- System Interface Information (wie z.B.: Betriebsspannung, Typische Timeouts)
- Geometrie (Startadresse der Sektoren, Größe der Sektoren)

Durch den optionalen Parameter „FLASH“ werden folgende Informationen zum Flash-Speicher zusätzlich zu den CFI-Informationen ausgegeben:

- Device-ID und Hersteller-ID gemäß JEDEC und die dadurch ableitbare dekodierte Bausteinbezeichnung. Bezeichnungen können nur von den Bausteinen ausgegeben werden, welche für das TQ-Modul freigegeben wurden und zum Zeitpunkt der Monitor-Programmerstellung bekannt waren.
- Des weiteren wird zu jedem Sektor angegeben, ob der Sektor schreibgeschützt ist oder nicht (Punkt-Markierung).

Syntax des Kommandos

MEMORY [FLASH]

Parameter **Bedeutung des Parameters**

FLASH	Gibt im Anschluss an das Speicher-Mapping die CFI-Informationen und die NOR-Flash-Sektorentabelle je Flash-Bank aus.
-------	--

Eingabebeispiel **Bedeutung des Beispiels**

memory flash	Das Speicher-Mapping und die Informationen zum NOR-Flash werden seitenweise angezeigt. Abbruch ist durch Control-C möglich.
--------------	---

6.2.13 NCLEARBBT – Löschen der Bad-Block-Tabelle im RAM mit Update im NAND-Flash

Dieses Kommando löscht alle Markierungen in der vom Monitor im RAM gehaltenen Bad-Block-Tabelle. Anschließend werden die vom Hersteller gesetzten Markierungen vom NAND-Flash neu eingelesen und die neue Bad-Block-Tabelle ins NAND-Flash an den für sie reservierten Block geschrieben.

Syntax des Kommandos
NCLEARBBT

6.2.14 NDUMP – Anzeige des Inhalts einer NAND-Flash-Seite

Mit diesem Kommando kann sich der Anwender den Inhalt einer Seite inklusive des Spare Arrays anzeigen lassen. Es werden auch Seiten aus Bad Blocks oder der BBT angezeigt.

Die Ausgabe der Daten erfolgt sowohl im Hexadezimalformat als auch im ASCII-Format. Zur besseren Lesbarkeit der Ausgabe werden nur 16 Datenbytes pro Zeile ausgegeben. Jeder Datenzeile wird als Identifikation die Zeilennummer vorangestellt. Das Kommando kann durch Eingabe der Tastenkombination Control-C unterbrochen werden.

Beim Versuch, eine nicht vorhandene Seite anzeigen zu lassen, wird eine Fehlermeldung ausgegeben.

Die Seitennummerierung entspricht dem Schema wie vom Kommando NINFO angezeigt. Alle Seiten des NAND-Flashs werden vom Monitor durchgehend, d.h. block- und chip-select-übergreifend, nummeriert.

Syntax des Kommandos
NDUMP page

Parameter	Bedeutung des Parameters
page	Nummer der Seite, deren Inhalt angezeigt werden soll (hexadezimal).

Eingabebeispiel	Bedeutung des Beispiels
ndump 20	Inhalt der Seite 0x20 ausgeben.

6.2.15 NERASE – Löschen von NAND-Flash Blocks

Der Anwender kann Blöcke des NAND-Flashs löschen. Blöcke, die als schlecht markiert sind, werden nur gelöscht, wenn der Anwender den Parameter ‚force‘ angibt. Wird nach Abfrage der Statusbits des NAND-Flashs festgestellt, dass das Löschen eines Blocks nicht erfolgreich war, wird der Block als schlecht markiert. Es wird eine Meldung ausgegeben und anschließend mit dem nächsten Block fortgesetzt.

Der NAND-Flash-Treiber nummeriert die Blöcke von 0 beginnend block- und chip-select-übergreifend bis zum letzten verfügbaren Block.

Beispiel: Besitzt ein NAND-Flash-Device 4 Chip-selects mit 8192 Blöcken je Chip-select, dann werden die Blöcke von 0 bis 0x7FFF (=32767) durchnummeriert.

Während des Löschens zeigt die Routine den aktuellen Fortschritt an.

Hinweis:

Die letzten beiden Blöcke sind für die Bad-Block-Tabelle des Monitors reserviert. Sie können nur gelöscht werden, wenn der Anwender den Parameter ‚force‘ angibt.

Achtung: ‚force‘ löscht ALLE angegebenen Blöcke einschließlich ALLER darin enthaltene Bad-Block-Markierungen! Somit können auch die Markierungen des Herstellers nicht wiederhergestellt werden, außer die entsprechenden Bits können sowieso nicht mehr gelöscht werden.

Syntax des Kommandos

```
NERASE firstBlock lastBlock [force]
```

Parameter	Bedeutung des Parameters
firstBlock	Nummer des ersten zu löschenden Blocks (hexadezimal)
lastBlock	Nummer des letzten zu löschenden Blocks (hexadezimal)
force	Löschen von schlechten Blöcken und der BBT-Blöcke erlauben

Eingabebeispiel	Bedeutung des Beispiels
nerase 3 7a1	Löscht die Blöcke 0x3 bis 0x7a1
nerase 3 7a1 force	Löscht die Blöcke 0x3 bis 0x7a1 inklusive darin enthaltene Bad-Block-Marker

6.2.16 NINFO – Informationen über das NAND-Flash anzeigen

Das Kommando zeigt folgende Informationen an:

- Hersteller und Hersteller ID
- Versorgungsspannung und Organisation
- Device ID
- Größe in MiB
- Geometrie und Anzahl der Chip-selects
- Anzahl der Bad Blocks
- Eine Auflistung aller Blöcke.

Es wird die Blocknummer sowie die Nummer der ersten und letzten Seite des Blocks angezeigt. Die Blöcke und Seiten sind durchgehend nummeriert, d.h. block- und chip-select-übergreifend. Weiterhin wird eine evtl. vorhandene Blockmarkierung angezeigt: Der Marker „wear“ (Wert 0x1) bedeutet, dass der Marker durch den Monitor aufgrund von Fehlern gesetzt wurde. Der Marker „factory“ (Wert 0x3) bedeutet, dass der Marker durch den Hersteller gesetzt wurde.

Syntax des Kommandos

```
NINFO [firstBlock lastBlock] | [bb]
```

Parameter	Bedeutung des Parameters
firstBlock	Nummer des ersten anzuzeigenden Blocks (hexadezimal)
lastBlock	Nummer des letzten anzuzeigenden Blocks (hexadezimal)
bb	Nur Anzeige der Bad Blocks

Eingabebeispiel	Bedeutung des Beispiels
ninfo	Zeigt Informationen zu allen Blöcken an
ninfo bb	Zeigt neben den allgemeinen Informationen nur noch die Bad Blocks an
ninfo 0 10	Zeigt neben den allgemeinen Informationen eine Liste der Blöcke 0x00 bis 0x10 an

6.2.17 NREAD – Daten kopieren vom NAND-Flash ins RAM

Es können Daten vom NAND-Flash seitenweise ins RAM kopiert werden. Bad Blocks werden dabei übersprungen.

Die Seitengröße beträgt 2048 Bytes im 1x8 Bit Modus und 4096 Bytes im 2x8 Bit Modus.

Vor dem Kopieren jeder Seite wird ihre ECC berechnet und, falls erforderlich, korrigiert. Handelt es sich um einen nicht korrigierbaren Fehler wird der Kopiervorgang abgebrochen und eine Fehlermeldung ausgegeben.

Der erste Parameter ist die Nummer der ersten zu kopierenden Seite. Die Seitennummerierung entspricht dabei dem Schema wie vom Kommando NINFO angezeigt. Alle Seiten des NAND-Flashs werden vom Monitor durchgehend nummeriert.

Während des Kopierens zeigt die Routine den aktuellen Fortschritt an.

Hinweise:

- Stößt die Routine vor dem Kopieren der ersten Seite eines Blocks auf einen Bad Block Marker wird dieser Block übersprungen, d.h. der Lesezeiger wird auf den Beginn des nächsten Blocks gesetzt. Die Anzahl der zu kopierenden Seiten bezieht sich immer auf die Seiten der guten Blöcke. Entsprechend verschiebt sich auch die Nummer der letzten zu lesenden Seite! Die Routine informiert den Anwender während des Kopiervorgangs über die Verschiebung.
- Wird versucht, an eine ungültige RAM-Adresse zu schreiben, siehe Hinweis in Kapitel 6.1.

Syntax des Kommandos

```
NREAD srcPage dstRamAddr nrOfPages
```

Parameter	Bedeutung des Parameters
srcPage	Nummer des ersten zu kopierenden Seite (hexadezimal)
dstRamAddr	Zieladresse im RAM (hexadezimal)
nrOfPages	Anzahl der zu kopierenden Seiten (hexadezimal)

Eingabebeispiel	Bedeutung des Beispiels
nread abc 50000 10	Kopiert 0x10 NAND-Flash-Seiten beginnend bei Seite 0xABC an die RAM-Adresse 0x50000.

6.2.18 NWRITE – Daten kopieren vom RAM oder NOR-Flash ins NAND-Flash

Es können Daten vom RAM oder NOR-Flash ins NAND-Flash kopiert werden. Es werden nur ganze NAND-Flash-Seiten kopiert. Dementsprechend werden die Bytes page-aligned geschrieben. Bad Blocks werden übersprungen.

Die Seitengröße beträgt 2048 Bytes im 1x8 Bit Modus und 4096 Bytes im 2x8 Bit Modus.

Vor dem Kopieren jeder Seite wird ihre ECC berechnet. Die ECC wird innerhalb des Spare Arrays an Offset 0x28 einer Seite abgelegt. Die ECC-Routine erzeugt 3 ECC-Bytes je 256 Datenbytes. Bei einer Seitengröße von 2048 Datenbytes je Flash-Baustein ergibt dies 24 ECC-Bytes.

Der zweite Parameter des Kommandos ist die Nummer der Zielseite. Die Seitennummerierung entspricht dabei dem Schema wie vom Kommando NINFO angezeigt. Alle Seiten des NAND-Flashs werden vom Monitor durchgehend nummeriert.

Während des Kopierens zeigt die Routine den aktuellen Fortschritt an.

Hinweise:

- Stößt die Routine vor dem Kopieren der ersten Seite eines Blocks auf einen Bad Block Marker wird dieser Block übersprungen, d.h. der Schreibzeiger wird auf den Beginn des nächsten Blocks gesetzt. Die Anzahl der zu kopierenden Seiten bezieht sich immer auf die Seiten der guten Blöcke. Entsprechend verschiebt sich auch die Nummer der letzten zu schreibenden Seite! Die Routine informiert den Anwender während des Kopiervorgangs über die Verschiebung.

Der Anwender muss darauf achten, dass aufgrund einer evtl. Verschiebung kein Schreibzugriff auf eine NAND-Flash-Seite erfolgt, die bereits Daten enthält. Ansonsten droht Datenverlust! Informationen über Bad Blocks sollten vorher mit dem Kommando ‚NINFO BB‘ eingeholt werden.

- Stellt der Treiber nach Abfrage der Statusbits fest, dass der Programmiervorgang nicht erfolgreich war, wird der betreffende Block als schlecht markiert. Der gesamte Kopiervorgang wird abgebrochen und eine Fehlermeldung ausgegeben, da die Anzahl der freien Blöcke aufgrund einer Verschiebung, wie oben beschrieben, im Voraus nicht bekannt ist.
- Das Kommando bricht die Bearbeitung ab, wenn durch den Kopiervorgang die reservierten Blöcke für die BBT überschrieben würden.
- Erfolgt beim Kopieren ein Zugriff auf eine Adresse außerhalb des RAMs, NOR-Flash, CCSR oder CAN, siehe Hinweis in Kapitel 6.1.

Syntax des Kommandos

```
NWRITE srcRamAddr dstPage nrOfPages
```

Parameter	Bedeutung des Parameters
srcRamAddr	Startadresse im RAM oder NOR-Flash (hexadezimal)
dstPage	Nummer der ersten Seite im NAND-Flash (Ziel) (hexadezimal)
nrOfPages	Anzahl der zu kopierenden Seiten (hexadezimal)

Eingabebeispiel	Bedeutung des Beispiels
nwrite 50000 abc 10	Kopiert 0x10 NAND-Flash-Seiten beginnend bei RAM-Adresse 0x50000 an die Seite 0xABC.

6.2.19 POST – Abfragen / Löschen der POSTxEN-Flags

Mit diesem Befehl kann durch den Benutzer gesteuert werden, ob die Power-On Self Tests (POST) bei einem Power-On / Reset des TQ-Moduls vom Monitor durchgeführt werden sollen.

Im Auslieferungszustand des Moduls werden immer alle POST-Tests durchgeführt. Gesteuert wird dieses Verhalten durch zwei Konfigurationsflags, die sich im Applikationsvorspann im NOR-Flash-Speicher befinden. Es sind hierfür zwei vom Benutzer änderbare Bits mit den Namen POST1EN („Power-On Self Test 1 Enable“) und POST2EN („Power-On Self Test 2 Enable“) definiert.

Zur Wirkung der POSTxEN-Flags siehe Kapitel 4.3. Wird der Applikationsvorspann gelöscht, stellt sich somit der Default-Fall ein. Die POST-Tests können mit dem POST-Kommando (einfacher Flash-Schreibvorgang auf 0) deaktiviert werden.

Um die POST-Tests wieder zu aktivieren, muss der gesamte Sektor, in dem sich der Applikationsvorspann befindet, gelöscht werden. Dies kann mit dem Kommando ERASE erreicht werden.

Syntax des Kommandos	
POST [1 2] [OFF]	

Parameter	Bedeutung des Parameters
1 2	POST1 oder 2 sind ausgewählt. Fehlt der Parameter, sind beide ausgewählt
OFF	Deaktiviert den / die Power-On Self Test(s)

Eingabebeispiel	Bedeutung des Beispiels
post	Abfrage der Flags POSTEN1 und POSTEN2
post 1	Abfrage der Flags POSTEN1
post 2 off	Der POST2 wird nicht mehr durchgeführt (POST2EN-Flag auf 0)
post off	Sowohl POST1 als auch POST2 werden in Zukunft nicht mehr durchgeführt

6.2.20 PROTECT – Ändern des Schreibschutzes von TQ-Informationen

Mit diesem Kommando ist es möglich den Schreib- und Löscheschutz für den TQ-Monitor und das TQ-Konfigurations-EEPROM einzustellen. Dies ist ein rein software-basierter Schutzmechanismus und ist deshalb nur im interaktiven Betriebsmodus des Monitors wirksam. Ein Schutz der TQ-Informationen vor Manipulation oder Zerstörung durch andere Programme als den TQ Monitor ist nicht vorhanden.

Der Schutz kann durch jeweils ein fest definiertes Passwort aktiviert bzw. deaktiviert werden. Im Standardfall (Power-On / Reset) ist der Schutz aktiviert.

Wurde der Schutz für den TQ-Monitor aufgehoben kann der Flash-Bereich, in dem sich der TQ-Monitor, der CIB und der HWIB befinden, gelöscht (Kommando ERASE) und beschrieben (Kommando COPY, SETHWI, SETCI) werden.

Wurde der Schutz für die TQ-Konfigurationsdaten aufgehoben, so kann in das EEPROM mit den Konfigurationsdaten geschrieben werden (Kommando EWRITE).

Wird das Passwort als einziger Parameter eingegeben, bezieht sich das Aktivieren bzw. Deaktivieren des Schreib- / Löscheschutzes immer auf den Monitor-Bereich (Abwärtskompatibilität). Für das Konfigurations-EEPROM muss der Parameter „CONF“ explizit angegeben werden.

Syntax des Kommandos	
PROTECT [(unprotect_key protect_key) ([MON] CONF)]	

Parameter	Bedeutung des Parameters
unprotect_key	Dieser Parameter enthält das Passwort, welches den Schreib- / Löscheschutz für den gewählten Bereich deaktiviert
protect_key	Dieser Parameter enthält das Passwort, welches den Schreib- / Löscheschutz für den gewählten Bereich aktiviert
MON	Symbol für den Flash-Bereich, in dem der Monitorcode, HWIB und CIB untergebracht ist.
CONF	Symbol für das EEPROM, in dem die TQ-Konfiguration untergebracht ist. Der Status des Monitorbereichs bleibt bei Verwendung dieses Parameters unberührt.

Eingabebeispiel	Bedeutung des Beispiels
protect xxxx	Kommando zum Aufheben des Schreibschutzes für den Bereich in dem der TQ-Monitor, HWIB und CIB platziert ist. Der Ausdruck xxxx dient als Platzhalter für das Passwort (unprotect_key).
protect xxxx conf	Kommando zum Aufheben des Schreibschutzes für den Bereich in dem die TQ-Konfiguration gespeichert ist. Der Ausdruck xxxx dient als Platzhalter für das Passwort (unprotect_key).
protect yyyy mon	Kommando zum Aktivieren des Schreibschutzes für den Bereich in dem der TQ-Monitor, HWIB und CIB platziert ist. Der Ausdruck yyyy dient als Platzhalter für das Passwort (protect_key).
protect	Ausgabe des Status der schützbaeren Bereiche TQ-Monitor und TQ-Konfiguration.

6.2.21 READ – Speicher auslesen

Mit diesem Kommando können Speicherbereiche als Dump über die serielle Schnittstelle ausgegeben werden. Die Ausgabe der Daten erfolgt sowohl im Hexadezimalformat als auch im ASCII-Format. Zur besseren Lesbarkeit der Ausgabe werden nur 16 Datenbytes pro Zeile ausgegeben. Jeder Datenzeile wird als Identifikation die Adresse (32 Bit) vorangestellt, aus der die Datenzeile im Speicher stammt. Das Kommando kann durch Eingabe der Tastenkombination Control-C unterbrochen werden.

Hinweise:

- Wird durch das LOAD-Kommando versucht, auf einen nicht verfügbaren Speicherbereich zuzugreifen, siehe Kap. 6.1 Kommandoeingabe.
- Wird ein READ-Kommando ohne Parameter eingegeben, so wird als Startadresse die Adresse gewählt, deren Daten beim letzten READ-Kommando gerade nicht mehr ausgegeben wurden. Falls kein vorhergehendes READ-Kommando eingegeben wurde, beginnt der Dump an Adresse 0x00000000.
- Anstelle einer Startadresse kann auch ein symbolischer Ausdruck (z.B. READ HWIB) verwendet werden. Das READ-Kommando gibt in diesem Fall den Speicherbereich oder den Beginn des Speicherbereichs aus, der durch das Symbol identifiziert wird.

Syntax des Kommandos

```
READ [MONITOR | HWIB | CIB | BOOTPAGE | RSTVEC] | [start [end] ]
```

Parameter	Bedeutung des Parameters
MONITOR	Symbolischer Ausdruck für die Startadresse des Monitorcodes im NOR-Flash
HWIB	Symbolischer Ausdruck für die Startadresse des Hardware Information Blocks im NOR-Flash
CIB	Symbolischer Ausdruck für die Startadresse des Configuration Information Blocks im NOR-Flash
BOOTPAGE	Symbolischer Ausdruck für die Startadresse der Monitor Bootpage im NOR-Flash
RSTVEC	Symbolischer Ausdruck für die Startadresse des Resetvektors im NOR-Flash (gibt den Resetvektor und die 12 davor liegenden Bytes aus).
start	Startadresse, ab der die Daten ausgegeben werden sollen (hexadezimaler 32-Bit-Wert)
end	Endadresse - letzte auszugebende Adresse (hexadezimaler 32-Bit-Wert) Falls der Parameter fehlt, wird als Default-Endadresse die Startadresse + 0xFF (Länge 256 Bytes) angenommen.

Eingabebeispiel	Bedeutung des Beispiels
read 80000100 80000120	Speicherinhalt ab Adresse 0x80000100 bis 0x80000120 ausgeben.
read monitor	Speicherinhalt ab der Startadresse des Monitorcodes im Flash bis Startadresse + 0xFF ausgeben.
read hwib	Speicherinhalt ab der Startadresse des HWIB im Flash bis Startadresse + 0xFF ausgeben. Die letzten 256 Bytes des HWIB können durch ein weiteres Absetzen eines READ-Kommandos ohne Parameter ausgelesen werden.
read cib	Speicherinhalt ab der Startadresse des CIB im Flash bis Startadresse + 0xFF ausgeben. Die letzten 256 Bytes des CIB können durch ein weiteres Absetzen eines READ-Kommandos ohne Parameter ausgelesen werden.
read	Speicherinhalt ab Endadresse des letzten READ-Kommandos bis Endadresse + 0xFF ausgeben.

6.2.22 REG – Zugriff auf Configuration, Control und Status Register

Über dieses Kommando können nahezu alle im Speicher gemappten Configuration, Control und Status Register (CCSR) des MPC85XX sowie alle benutzerdefinierten Register unter ihrem symbolischen Namen benutzt werden. Es werden folgende Teilfunktionen unterstützt:

- Register schreiben
- Register lesen
- Alle Register ausgeben (Abbruch durch Tastenkombination Control-C möglich)

Hinweise:

- Das REG-Kommando ist fakultativ, d.h. das Befehlssymbol REG kann weggelassen werden.
- Wurde ein benutzerdefiniertes Register im NOR-Flash-Speicher definiert, so kann auf dieses nur lesend zugegriffen werden. Zwar kann durch den Benutzer ein REG Schreibkommando auf das Flash ausgeführt werden, dieses wirkt sich aber nicht aus.
- Das REG Schreibkommando überprüft nicht, ob das Beschreiben des Registers erfolgreich war.
- Die Register der Security Engine werden derzeit nicht unterstützt.
- Bei einem Schreib- oder Leseversuch auf ein nicht vorhandenes Register der eingesetzten MPC85xx-Variante wird der Befehl abgebrochen und eine Fehlermeldung ausgegeben.
- Bei einem Schreib- oder Leseversuch auf eine DPRAM-Adresse oder eine SDRAM-Adresse wird der Befehl abgebrochen und eine Fehlermeldung ausgegeben.

Syntax des Kommandos
[REG] [name [[=] value]]

Parameter	Bedeutung des Parameters
name	Name eines vordefinierten CCSR oder eines benutzerdefinierten Registers
value	Zuzuweisender Wert, hexadezimal, max. 32 Bit (bei kleineren Registerbreiten werden ggf. MSBs abgeschnitten)

Eingabebeispiel	Bedeutung des Beispiels
reg sar0	Register SAR0 ausgeben
sar0 = CAFE	Register SAR0 auf 0xCAFE setzen
reg	Alle Register seitenweise ausgeben (die erste Seite zeigt nur das Register CCSRBAR)

6.2.23 RTC – Einstellen und Ausgeben der aktuellen Uhrzeit und des Datums

Mit diesem Kommando kann die Uhrzeit und das Datum der modulinternen RTC (I2C) ausgelesen und gesetzt werden.

Hinweise:

- Es erfolgt nur eine Bereichsüberprüfung der Eingabeparameter. Dies stellt keine Überprüfung dar, ob die eingegebenen Parameter einem korrekten Datum entsprechen. Somit können inkorrekte Datums- werte zu einem undefinierten Verhalten der RTC führen.
- Werden Datum und Uhrzeit gemeinsam eingegeben, muss das Datum der Uhrzeit vorangestellt sein (siehe Eingabebeispiel).
- Die Eingabe der doppelten Anführungszeichen ist erforderlich (siehe Eingabebeispiel).
- Das Century-Bit der RTC wird fest auf 1 gesetzt.

Syntax des Kommandos

RTC [[TIME | DATE] [=] string]

Parameter	Bedeutung des Parameters
TIME	Eingabe der Uhrzeit
DATE	Eingabe des Datums
string	Parameter der Uhrzeit oder des Datums oder beider. Das Datum muss im Format „YY/MM/DD“ angegeben werden. Die Uhrzeit muss im Format „HH:MM:SS“ oder „MODE HH:MM:SS“ angegeben werden. Falls die Eingabe des Datums im 12 Stundenmodus erfolgt, darf der Ausdruck MODE nur dem String „PM“ oder „AM“ entsprechen.

Eingabebeispiel	Bedeutung des Beispiels
rtc	Ausgabe des aktuellen Datums und der Uhrzeit der RTC
rtc=	Ausgabe des aktuellen Datums und der Uhrzeit der RTC
rtc time = "23:59:01"	Eingabe der Uhrzeit für den 24-Stundenmodus
rtc time "PM 11:59:01"	Eingabe der Uhrzeit für den 12-Stundenmodus
rtc date="04/12/31"	Eingabe des Datums
rtc = "04/12/31 AM 11:59:01"	Eingabe des Datums und der Uhrzeit für den 12 Stundenmodus

6.2.24 SAVE – Ausgeben von S-Record-Files

Mit diesem Kommando können Daten oder Programme im Motorola S-Record-Format aus einem Speicherbereich des TQ-Moduls ausgegeben (Abspeicherung) werden. Es werden ausschließlich S3-Records (Daten mit 32-Bit-Adressfeld) generiert. Die Ausgabe erfolgt über die derzeit aktive serielle Schnittstelle. Das Abspeichern wird abgebrochen wenn der Benutzer während des Vorgangs die Tastenkombination Control-C drückt.

Hinweis:

Wird durch das SAVE-Kommando versucht, auf einen nicht verfügbaren Speicherbereich zuzugreifen, siehe Kap. 6.1 Kommandoeingabe.

Syntax des Kommandos

SAVE start length

Parameter	Bedeutung des Parameters
start	Startadresse des abzuspeichernden Bereiches (hexadezimaler 32-Bit-Wert)
length	Länge des abzuspeichernden Bereichs (hexadezimaler 32-Bit-Wert)

Eingabebeispiel	Bedeutung des Beispiels
save 80000100 100	Ab der Adresse 0x80000100 werden die nächsten 0x100 Bytes als S-Record über die serielle Schnittstelle ausgegeben

6.2.25 SETCI – Speichern des Configuration Information Blocks (CIB)

Mit diesem Kommando können vom Benutzer CIB-Informationen in den CIB-Bereich des NOR-Flashs geschrieben werden. Hierbei prüft das Kommando, ob der Bereich des CIB schon gelöscht ist und ob der Schreibschutz für den Bereich aufgehoben wurde. Falls dies der Fall ist, werden die CIB-Informationen ohne syntaktische und semantische Überprüfung in den CIB-Bereich des Flashs kopiert. Die Daten werden nicht in Großbuchstaben umgewandelt. Die Speicherung erfolgt wie eingegeben, d.h. case-sensitive. Die Syntax und Semantik sowie die Startadresse der Information sind aus Kapitel 4.2 Configuration Information Block (CIB) ersichtlich.

Hinweise:

- Der Schreibschutz (siehe PROTECT-Kommando) muss vor der Ausführung des Befehls für den TQ-Monitorbereich aufgehoben worden sein. Ansonsten scheitert das Speichern des CIB.
- Im Anschluss an den Schreibvorgang wird keine erneute Analyse (Re-scan) der CIB-Informationen durchgeführt.
- Das Kommando SETCI ohne Parameter gibt keine Informationen über die detektierten Frequenzen oder Timings aus. Es gibt nur die beim letzten Boot-Vorgang extrahierten CIB-Informationen als dekodierte Informationen aus.
- Der CIB-Bereich im Flash kann maximal einmal mit dem Kommando beschrieben werden. Vor einem zweiten Schreibversuch muss der Monitorbereich evtl. komplett gelöscht werden, da sich der CIB im gleichen Flash-Sektor befindet wie der Beginn des Monitorcodes. Dies hat zur Folge, dass auch der Monitor wieder ins Flash geschrieben werden muss (Kommando COPY MONITOR).
- Bei Angabe des optionalen Symbols RECOVER im Kommandoaufruf wird, falls vorhanden, ein sich im SDRAM befindendes Backup der CIB-Informationen ins Flash gespeichert. Diese Option kann genutzt werden, um zum Beispiel bei einem Monitor-Update die ursprünglich (vor der Löschung des Flashs) vorhandene CIB-Information wieder im Flash herzustellen.
- Hinweise für die Benutzung der RECOVER-Funktion:
 - Zum Initialisierungszeitpunkt des Monitors muss ein gültiger CIB-Eintrag existieren. Dieser Eintrag wird ins SDRAM kopiert.
 - Wird nach dem Löschen des Flash-Bereichs ein HRESET ausgeführt, ist die CIB-Information im SDRAM verloren!

Syntax des Kommandos

```
SETCI [ (RECOVER | (operator parameter) ...) ]
```

Parameter	Bedeutung des Parameters
RECOVER	Symbol für die im Monitor vorhandenen CIB-Informationen
operator	Der Operator, welcher in den CIB geschrieben werden soll
parameter	Der zum Operator gehörige Parameter, der in den CIB geschrieben werden soll (32-Bit-Wert, <u>Default dezimal (!)</u> , mit vorangestelltem "0x" wird der Wert hexadezimal interpretiert)

Eingabebeispiel	Bedeutung des Beispiels
setci -cu 8333333 -cc 3333333333 -cb 833333333 -tf 0x00000c54 -td 25	Die im String nach dem Kommando angegebenen Informationen werden in den CIB-Bereich des Flashs geschrieben.
setci recover	Die im Monitor im SDRAM vorhandenen CIB-Informationen werden in den CIB-Bereich des Flashs geschrieben.
setci	Ausgabe der CIB-Daten als dekodierte Information (sofern vorhanden)

6.2.26 SETHWI – Speichern des Hardware Information Blocks (HWIB)

Mit diesem Kommando können vom Benutzer HWIB-Informationen in den HWIB-Bereich des NOR-Flashs geschrieben werden. Hierbei prüft das Kommando, ob der Bereich des HWIB schon gelöscht ist und ob der Schreibschutz für den Bereich aufgehoben wurde. Falls dies der Fall ist, erfolgt die Umwandlung (u.a. Konvertierung in Grossbuchstaben) der HWIB-Informationen und wird ohne eine syntaktische und semantische Überprüfung in den HWIB-Bereich des Flashs kopiert. Die Syntax und Semantik sowie die Startadresse der Information sind aus Kapitel 4.1 Hardware Information Block (HWIB) ersichtlich.

Hinweise:

- Der Schreibschutz (siehe PROTECT-Kommando) muss vor der Ausführung des Befehls für den TQ-Monitorbereich aufgehoben worden sein. Ansonsten scheitert das Speichern des HWIB.
- Im Anschluss an den Schreibvorgang wird keine erneute Analyse (Re-scan) der HWIB-Informationen durchgeführt.
- Das Kommando SETHWI ohne Parameter gibt keine Informationen über die detektierte HW aus. Es gibt nur die beim letzten Boot-Vorgang extrahierten HWIB-Informationen als dekodierte Informationen aus.
- Der HWIB-Bereich im Flash kann maximal einmal mit dem Kommando beschrieben werden. Vor einem zweiten Schreibversuch muss der Monitorbereich evtl. komplett gelöscht werden, da sich der HWIB im gleichen Flash-Sektor befindet wie der Beginn des Monitorcodes. Dies hat zur Folge, dass auch der Monitor wieder ins Flash geschrieben werden muss (Kommando COPY MONITOR).
- Bei Angabe des optionalen Symbols RECOVER im Kommandoaufruf wird, falls vorhanden, ein sich im SDRAM befindendes Backup der HWIB-Informationen ins Flash gespeichert. Diese Option kann genutzt werden, um zum Beispiel bei einem Monitor-Update die ursprünglich (vor der Löschung des Flashs) vorhandene HWIB-Information wieder im Flash herzustellen.
- Hinweise für die Benutzung der RECOVER-Funktion:
 - Zum Initialisierungszeitpunkt des Monitors muss ein gültiger HWIB-Eintrag existieren. Dieser Eintrag wird ins SDRAM kopiert.
 - Wird nach dem Löschen des Flash-Bereichs ein HRESET ausgeführt, ist die HWIB-Information im SDRAM verloren!

Syntax des Kommandos

SETHWI [(RECOVER | (string ...))]

Parameter	Bedeutung des Parameters
RECOVER	Symbol für die im Monitor vorhandenen HWIB-Informationen
String	In den HWIB-Bereich zu schreibender String (ohne Anführungszeichen)

Eingabebeispiel	Bedeutung des Beispiels
sethwi TQM8540ABBAD7-APFBC-20002.0102 19465278 00D093123456 4	Die im String nach dem Kommando angegebenen Informationen werden in den HWIB-Bereich des Flashs geschrieben.
sethwi recover	Die im Monitor im SDRAM vorhandenen HWIB-Informationen werden in den HWIB-Bereich des Flashs geschrieben.
sethwi	Ausgabe der HWIB-Daten als dekodierte Information

6.2.27 TEMPSENS – Ausgabe der aktuellen Temperatur des Moduls

Mit diesem Kommando wird die aktuelle Temperatur des TQ-Moduls (CPU-Nähe) in Grad Celsius und Fahrenheit ausgegeben. Die Ausgabe der Temperatur erfolgt ohne Nachkommastellen, da der eingesetzte Sensor nur eine Genauigkeit von +/- 2 Grad Celsius besitzt.

Syntax des Kommandos

TEMPSENS

Eingabebeispiel	Bedeutung des Beispiels
tempsens	Ausgabe der aktuellen Temperatur in Celsius und Fahrenheit

6.2.28 WRITE – Speicher beschreiben

Mit diesem Kommando kann der Benutzer direkt byteweise Daten in das SDRAM schreiben. Die Daten können dem Kommando als Sequenz von Strings oder hexadezimalen Werten oder einer Kombination aus beiden übergeben werden.

Hinweise:

- Wird durch das WRITE-Kommando versucht auf einen nicht verfügbaren Speicherbereich zuzugreifen, siehe Kap. 6.1 Kommandoeingabe.

Mit dem WRITE-Kommando können Daten nur ins SDRAM geschrieben werden. Die Daten können dann mit folgenden Kommandos in andere Speicherbereiche übertragen werden:

- Kommando COPY (NOR-Flash und EEPROM)
- Kommando NWRITE (NAND-Flash)

POST-Meldung	Bedeutung der Ausgabe
CAN different from HWIB	Beim CAN-Test wurde eine andere Anzahl von CAN Controllern detektiert als im HWIB definiert ist.
CCB clock frequency different from CIB	Die im CIB angegebene Core Complex Bus Taktfrequenz weicht um mehr als die zulässige Toleranz von der gemessenen Taktfrequenz ab.
CIB parsing failed	Der CIB ist syntaktisch nicht korrekt.
CPU clock frequency different from CIB	Die im CIB angegebene CPU-Taktfrequenz weicht um mehr als die zulässige Toleranz von der ermittelten Taktfrequenz ab.
FLASH size different from HWIB	Beim Flash-Test wurde eine andere Größe des NOR-Flashs detektiert als im HWIB definiert ist.
HWIB parsing failed	Der HWIB ist syntaktisch nicht korrekt.
I2C bus collision	Der I2C-Controller entdeckte einen Zugriffskonflikt auf dem I2C-Bus.
I2C device did not acknowledge	Das I2C-Device sendete kein Acknowledge zurück.
I2C device is busy	Das I2C-Device kann im Moment keinen Befehl annehmen. Der I2C-Bus ist blockiert.
I2C initialization failed	Der I2C-Controller konnte nicht erfolgreich initialisiert werden.
Invalid bus clock frequency	Die im CIB angegebene Local Bus Taktfrequenz wird nicht unterstützt.
Invalid CCB/Bus clock divider ratio	Ungültiger Teilerfaktor zwischen den CIB Parametern CCB Clock (-cc) und Local Bus Clock (-cb).
Invalid CCB clock frequency	Die im CIB angegebene Core Complex Bus Taktfrequenz wird nicht unterstützt.
Invalid connector height value	Die im HWIB angegebene Steckerbauhöhe entspricht nicht der Steckerbauhöhe des TQ-Moduls.
Invalid CPU clock frequency	Die im HWIB oder CIB angegebene CPU-Taktfrequenz wird nicht unterstützt.
Invalid CPU mask value	Die im HWIB angegebene CPU-Maske wird nicht unterstützt.
Invalid DDR SDRAM CAS latency value	Der im CIB angegebene CAS-Latency-Wert für das DDR1-SDRAM wird nicht unterstützt.
Invalid DDR speed value	Die im HWIB angegebene DDR-Geschwindigkeit des SDRAMs wird nicht unterstützt.
Invalid encryption option	Der im HWIB angegebene Wert für die Encryption Option liegt außerhalb des gültigen Bereichs.
Invalid FLASH size	Die im HWIB angegebene NOR-Flash-Größe wird vom Monitor nicht unterstützt.
Invalid FLASH timing value	Der im CIB angegebene Wert für das NOR-Flash-Timing entspricht keinem gültigen Registerwert zur Timing-Einstellung.
Invalid NAND FLASH size	Die im HWIB angegebene NAND-Flash-Größe wird vom Monitor nicht unterstützt.
Invalid number of CAN controllers	Die im HWIB angegebene Anzahl der CAN-Controller wird vom Monitor nicht unterstützt.
Invalid SDRAM size	Die im HWIB angegebene SDRAM-Größe wird vom Monitor nicht unterstützt.
Invalid temperature range	Die im HWIB angegebene Temperatur liegt außerhalb des gültigen Bereichs, oder es wurde keine Angabe zur Temperatur gemacht.
Invalid temperature sensor values	Der Temperatursensor liefert ungültige Werte zurück.
Invalid user EEPROM size	Die im HWIB angegebene Größe des EEPROMs für die Anwenderdaten wird vom Monitor nicht unterstützt.
Invalid value for optional equipment	Der im HWIB angegebene Parameter für die optionale Bestückung des TQ-Moduls hat einen ungültigen Wert.
Missing encryption option in HWIB	Der Prozessor besitzt eine Security Engine. Diese Angabe fehlt jedoch im HWIB.
Missing serial number	Im HWIB fehlt die Angabe der Modulseriennummer
Missing MAC address(es)	Im HWIB fehlt die Angabe der MAC-Adresse oder der die Angabe zur Anzahl zusätzlicher MAC-Adressen.
Missing module revision number	Im HWIB fehlt die Angabe der Modulrevisionsnummer
Missing user EEPROM size	Im HWIB fehlt die Größenangabe für das Anwenderdaten-EEPROM.
MPC has no security engine	Im HWIB wurde eine Encryption Option angegeben. Der Prozessor unterstützt diese Funktion aber nicht.
MPC type different from HWIB	Der im HWIB angegebene PowerPC-Typ unterscheidet sich vom detektierten Typ.
NAND FLASH initialization failed	Das NAND-Flash konnte nicht gefunden werden oder wird nicht unterstützt oder es trat ein Fehler bei der BBT-Initialisierung auf.
NAND FLASH size different from HWIB	Beim NAND-Flash-Test wurde eine andere Größe des Flashs detektiert als im HWIB definiert ist.
Number of NAND FLASH chip-selects different from HWIB	Beim NAND-Flash-Test wurde eine andere Anzahl von Chip-selects detektiert als im HWIB definiert ist.
Number of NAND FLASH devices different from HWIB	Beim NAND-Flash-Test wurde eine andere Anzahl von Devices detektiert als im HWIB definiert ist.
POST OK	Die Durchführung von POST 1 und 2 erfolgte fehlerfrei.
POST error:	Präfix, das vor einer POST Fehlermeldung ausgegeben wird.

POST-Meldung	Bedeutung der Ausgabe
POST message buffer full!	Der vorgesehene Speicher für POST-Meldungen kann keine weitere Meldungen aufnehmen.
POST 1 skipped	Der POST 1 Test ist durch den Software-Schalter deaktiviert und wurde nicht ausgeführt, oder der Monitor wurde mit BOOT_SDRAM gestartet.
POST 2 skipped	Der POST 2 Test ist durch den Software-Schalter deaktiviert und wurde nicht ausgeführt, oder der Monitor wurde mit BOOT_SDRAM gestartet.
RTC invalid time	Alle Einträge der RTC wurden mit Wert = 0 gelesen.
SDRAM size different from HWIB	Beim SDRAM-Test wurde eine andere Größe des SDRAMs detektiert als im HWIB definiert ist.
Unknown FLASH type	Es wurde kein NOR-Flash gefunden bzw. der Typ des NOR-Flashs ist unbekannt.
Unknown MPC type	Der Prozessortyp ist unbekannt.
User EEPROM different from HWIB	Beim EEPROM-Test wurde eine andere Größe des EEPROMs für Anwenderdaten detektiert als im HWIB definiert ist.

Tabelle 12: POST-Meldungen

Alle Rück- und Fehlermeldungen, die als Klartext im ASCII-Format nach einer Kommandoingabe ausgegeben werden können, sind in Tabelle 13 und Tabelle 14 abgebildet:

Kommando Rück-/ Fehler-Meldung	Bedeutung der Ausgabe
Argument expected	Die Syntax des eingegebenen Kommandos ist nicht korrekt. Es fehlen ein oder mehrere Parameter.
Break	Kommando wurde durch Control-C abgebrochen.
Configuration information already set	Das SETCI Kommando kann die CIB-Informationen nicht ins NOR-Flash schreiben da schon Informationen vorhanden sind.
Copy pages aborted, because next good block may contain data.	Das Kopieren vom RAM oder NOR-Flash ins NAND-Flash wurde aufgrund eines Fehlers abgebrochen. Um eine Datenkorruption zu vermeiden, wurde nicht in den nachfolgenden Block geschrieben.
CRC check failed	Die CRC-Prüfung des Kommandos ist fehlgeschlagen.
Data transmission error	Bei der durch das Kommando eingeleiteten seriellen Datenübertragung ist ein Übertragungsfehler aufgetreten.
Date expression expected (YY/MM/DD)	Die Eingabe des Datums fehlt oder ist ungültig.
Download error	Bei dem durch das Kommando eingeleiteten Download ist ein allgemeiner Fehler aufgetreten.
ECC check: Uncorrectable data error!	Beim Lesen vom NAND-Flash wurde ein nichtkorrigierbarer Fehler entdeckt. Die Operation wurde danach abgebrochen.
EEPROM is write protected	Das EEPROM ist schreibgeschützt. Der Schreibschutz kann nur mit einem Passwort aufgehoben werden.
EEPROM access failed	Auf das EEPROM kann nicht zugegriffen werden, z.B. aufgrund ungültiger Parameter oder eines I2C-Busfehlers.
End address is outside FLASH	Dem Kommando COPY wurde eine Startadresse für die Quelle übergeben, die im NOR-Flash liegt. Die Endadresse befindet sich jedoch außerhalb.
Erasing not successful!	Das Löschen eines NAND-Flash-Blocks schlug fehl. Der Block wurde markiert und die Operation danach abgebrochen.
Error erasing FLASH memory	Beim Löschen des NOR-Flashs ist ein Fehler aufgetreten (Timeout).
FLASH programming error	Es ist ein Fehler beim Schreiben ins NOR-Flash aufgetreten, z.B. konnten Daten nicht erfolgreich geschrieben werden, da der Flash-Bereich nicht gelöscht wurde, oder einige Sektoren sind geschützt (siehe Kommando MEMORY FLASH).
FLASH sector(s) protected	Es wurde versucht, einen durch den Monitor geschützten NOR-Flash-Sektor zu löschen.
GOT buffer too small (MON_GOT_MAXLEN)	Bei der Relokation ist ein Fehler aufgetreten (Interner Fehler).
Hardware information already set	Das SETHWI Kommando kann die HWIB-Informationen nicht ins NOR-Flash schreiben da schon Informationen vorhanden sind.
Hex number expected	Die Syntax des eingegebenen Kommandos ist nicht korrekt. Es wird ein Parameter in hexadezimaler Schreibweise erwartet.
I2C access failed	Beim I2CDEV Kommando ist ein Fehler beim Scannen des I2C-Busses aufgetreten.
Invalid address	Vor dem Ausführen des Kommandos GO ohne Parameter wurde kein gültiger Download einer S-Record-Datei durchgeführt, oder zwischen Download und Kommando GO wurde ein Reset durchgeführt.
Invalid argument	Das eingegebene Argument ist ungültig (falscher Wertebereich, falsches Symbol oder Schlüsselwort).

Kommando Rück-/ Fehler-Meldung	Bedeutung der Ausgabe
Invalid data from device!	Das NAND-Flash enthält ungültige Daten in den ID-Bytes.
Invalid FLASH bank	Die mit dem Kommando ausgewählte NOR-Flash-Bank ist nicht vorhanden oder ungültig.
Invalid FLASH sector address	Die mit dem Kommando eingegebenen Start- und/oder Endadresse für den Sektor ist ungültig.
Invalid hex number	Die eingegebene Hexzahl enthält unzulässige Zeichen.
Invalid parameter!	Ein NAND-Flash Funktion wurde mit einem ungültigen Parameter aufgerufen.
Invalid s-record	Der durch das Kommando verarbeitete S-Record ist fehlerhaft.
Invalid string expression	Am Ende des Strings werden doppelte Anführungszeichen erwartet.
Invalid temperature sensor values	Der vom Temperatursensor gelesene Temperaturwert ist außerhalb des für den Sensor spezifizierten Bereichs.
Module has no EEPROM (according to HWIB scan)	Im HWIB wurde für das Anwenderdaten-EEPROM keine oder eine ungültige Größenangabe oder Größe = 0 angegeben. Beim Einsprung über BOOT_SDRAM wurde kein HWIB-Scan durchgeführt.
Module has no NAND-FLASH (according to HWIB scan)	Es wurde versucht auf das NAND-Flash zuzugreifen, das gemäß HWIB nicht vorhanden ist. Beim Einsprung über BOOT_SDRAM wurde kein HWIB-Scan durchgeführt.
Module has no RTC (according to HWIB scan)	Es wurde versucht auf die Real-Time Clock zuzugreifen, die gemäß HWIB nicht vorhanden ist. Beim Einsprung über BOOT_SDRAM wurde kein HWIB-Scan durchgeführt.
no FLASH or incompatible FLASH	Das NOR-Flash wurde nicht gefunden oder wird nicht unterstützt.
No recovery data available	Bei der Ausführung des Kommandos SETCI RECOVER oder SETHWI RECOVER konnten keine Informationen gefunden werden, die in den CIB oder HWIB geschrieben werden können.
No user defined registers	Es wurden keine benutzerspezifischen Register definiert.
Programming not successful!	Das Programmieren einer NAND-Flash-Seite schlug fehl. Der Block, in dem sich die Seite befindet wurde markiert und die Operation danach abgebrochen.
Start address is greater than end address	Die angegebene Startadresse ist größer als die Endadresse. Die Startadresse darf höchstens gleich groß sein.
String expression expected	Es muss ein String, eingeschlossen durch doppelte Anführungszeichen, eingegeben werden.
Target end address is outside FLASH	Dem Kommando COPY wurde eine Startadresse für das Target übergeben, die im NOR-Flash liegt, die Endadresse befindet sich jedoch außerhalb.
Time expression expected ([AM PM] HH:MM:SS)	Die Eingabe der Zeit fehlt oder ist ungültig.
Timeout. Device not ready!	Das NAND-Flash ist zur Zeit nicht bereit, Daten oder Kommandos anzunehmen.
Too many user registers	Es wurden zuviele benutzerdefinierte Register definiert (max 100).
Unexpected argument	Die Syntax des eingegebenen Kommandos ist nicht korrekt. Es wurde ein oder mehrere Parameter zuviel eingegeben.
Unknown command or register	Das eingegebene Kommando / Register ist unbekannt.
Unknown device ID!	Das NAND-Flash wird nicht unterstützt
Unknown I2C device	Die Datenquelle als Parameter für das Kommando COPY muss das Konfigurations- oder Anwenderdaten-EEPROM sein.
Unknown register	Das eingegebene Register ist unbekannt.
Value out of range	Die Eingabeparameter eines Kommandos liegen außerhalb des gültigen Bereichs.

Tabelle 13: Rück-/Fehlermeldungen des Monitors nach Kommandoeingabe

Fehlermeldung	Bedeutung der Ausgabe
Different FLASH device IDs in FLASH bank x	Die IDs der NOR-Flash-Bausteine innerhalb einer Bank sind verschieden. Die Banknummer wird anstelle x angegeben.

Tabelle 14: Fehlermeldungen des Monitors beim Booten