VME - AIO16

16 analoge Eingänge und
4 analoge Ausgänge
mit 16 (12) Bit Auflösung

Software-Handbuch

Dokument-Datei:	M:\texte\Doku\MANUALS\VME\AIO16\Deutsch\AIO1637S.ma9
Datum der Druckvorlagen-herstellung:	04.09.2002

Beschriebene Firmware-Version:	aiosrv 07

Änderungen in den Kapiteln

Die hier aufgeführten Änderungen im Anwenderhandbuch betreffen sowohl Änderungen in der **Hardware** als auch reine Änderungen in der **Beschreibung** der Sachverhalte.

Kapitel	Änderungen gegenüber Handbuch-Revision 3.5
3.3, 4.9.1, 4.9.13	VMEbus-Adressen korrigiert.
5.2.4	Formel zur Berechnung von `auxcr` korrigiert.
6.3.3	Kapitel ‘ Selective-D/A-Buffer-Struktur’ eingefügt.

Weitere technische Änderungen vorbehalten.

Der Inhalt dieses Handbuches wurde mit größter Sorgfalt erarbeitet und geprüft. **esd** übernimmt jedoch keine Verantwortung für Schäden, die aus Fehlern in der Dokumentation resultieren könnten. Insbesondere Beschreibungen und technische Daten sind keine zugesicherten Eigenschaften im rechtlichen Sinne.

esd hat das Recht, Änderungen am beschriebenen Produkt oder an der Dokumentation ohne vorherige Ankündigung vorzunehmen, wenn sie aus Gründen der Zuverlässigkeit oder Qualitätssicherung vorgenommen werden oder dem technischen Fortschritt dienen.

Sämtliche Rechte an der Dokumentation liegen bei **esd**. Die Weitergabe an Dritte und Vervielfältigung jeder Art, auch auszugsweise, sind nur mit schriftlicher Genehmigung durch **esd** gestattet.

esd electronic system design gmbh
Vahrenwalder Str. 207
30165 Hannover

Tel.: 0511/372 98-0
FAX : 0511/372 98-68
E-Mail: info@esd.electronics.com
Internet: www.esd-electronics.com

Inhalt

Seite

1. Einführung 3

2. Default-Parameter 5

3. Shared-Adreßbereich 7
3.1 Hinweise zu VMEbus-Zugriffen 7
3.2 Hinweise zu Zugriffen der lokalen CPU 7

4. Kommandos und Parameterübergabe 11
4.1 Belegung des Parameter-Buffers 11
4.2 Karten-Identifikation 13
4.3 Karten-Status 14
4.4 Hardware-Revision 15
4.5 Vorgehensweise beim Kommandoaufruf 16
4.6 Kommando-Sektion 17
4.6.1 Die Zellen sema und cstat 18
4.6.2 Allgemeines zur Kommandozelle cmmd 19
4.6.3 Übersicht der Kommandos und setzbaren Parameter 19
4.7 Beschreibung der System-Kommandos 22
4.7.1 Aktuelle Parameter im I²C-EEPROM speichern 22
4.7.2 System-Neustart incl. Selbsttest 22
4.7.3 System-Neustart mit Default-Parametern 23
4.7.4 Statistik über aktuellen Buffer 24
4.8 Preload 27
4.8.1 Übersicht der lesbaren Zellen des Parameter-Buffers 27
4.8.1.1 Lesen der Timer-Input-Frequenz (tifreq) 28
4.8.1.2 Lesen der aktuellen Conversion Time (cnvtime) 28
4.8.1.3 Lesen der ‘D/A-Wandler-Werte nach RESET’ (dac1p, dac2p...) . . 28
4.8.2 Ausgangswerte der D/A-Wandler nach RESET (dacxp) 29
4.8.3 Trigger-Periodendauer über Timer vorgeben (cnvtime) 29
4.8.4 CPU-Frequenz vorgeben 30
4.9 Steuerparameter 31
4.9.1 Übersicht der lesbaren Zellen des Parameter-Buffers (Steuerparameter) 31
4.9.2 VMEbus-Interrupt-Level setzen (vmelev) 33
4.9.3 VMEbus-Interrupt-Vektor setzen (vmevec) 33
4.9.4 Multiplexer der A/D-Eingänge setzen (muxmode) 34
4.9.5 Multiplexer der D/A-Wandler-Ausgänge setzen (dacmode) 34
4.9.6 Trigger-Quelle für Start der A/D-Wandlung wählen (trigmod) 35
4.9.7 Quelle für LDAC der D/A-Wandler wählen (ldcmod) 36
4.9.8 Auswahl der A/D-Datenverarbeitung (vadsrv) 37
4.9.9 A/D-Kanäle selektieren (vstart, vend) 39
4.9.10 Quelle für VMEbus-Interrupt wählen (vvtrg) 40
4.9.11 A/D-Wandler-Auflösung setzen (vadres) 41
4.9.12 Auswahl der D/A-Datenverarbeitung (vdasrv) 42
4.9.13 Anzahl zu addierender A/D-Werte (vsmcnt) 43
4.9.14 A/D-Buffer einrichten (adbuf) 44
4.9.15 D/A-Buffer einrichten (dabuf) 44

4.9.16 D/A-Kanäle selektieren (dastart, daend) 45
4.9.17 Selective-D/A-Buffer 46
4.10 A/D-Wandler-Status (adstat0...adstat3) 47

5. Lesen der A/D-Meßwerte und Setzen der D/A-Wandler 49
5.1 Lesen der A/D-Meßwerte 49
5.1.1 Rohdaten der A/D-Wandler - adwert01...adwert16 49
5.1.2 Korrigierte A/D-Wandler-Daten - advac01...advac16 51
5.1.3 Summierte A/D-Wandler-Daten - adc01.summed...adc16.summed 53
5.1.4 Offset der A/D-Wandler - offs01...offs16 54
5.1.5 A/D-Werte bei anliegender Referenzspannung - ref501...ref516 55
5.1.6 A/D-Werte bei anliegender D/A-Spannung von 0 V: D/A-Offset (adc001...adc016) 57
5.1.7 A/D-Werte bei anliegendem D/A-Wert $7F00: D/A-Gain (adc101...adc116) 58
5.1.8 A/D-Wandler: Gain-Korrektur-Faktoren (scale01...scale16) 60
5.1.9 Offset-Werte der addierten 32-Bit-A/D-Werte (offs01.summed...offs16.summed) 61
5.2 Auxilary-Inputs 62
5.2.1 Rohdaten der Auxilary-Inputs auxin1...auxin8 62
5.2.2 Korrigierte Daten der Auxilary-Inputs - auxcr1...auxcr8 63
5.2.3 Offset der Auxilary-Inputs - ofsax1...ofsax8 64
5.3 Setzen der D/A-Wandler 66
5.3.1 Hinweise zum Setzen der D/A-Wandler 66
5.3.2 Datenformat der D/A-Wandler-Werte 66

6. Buffer-Mode 67
6.1 Übersicht 67
6.2 Buffer und Frames 67
6.3 A/D- und D/A-Buffer-Status-Strukturen 69
6.3.1 A/D-Buffer-Status-Struktur 69
6.3.2 D/A-Buffer-Status-Struktur 69
6.3.3 Selective D/A-Buffer-Struktur 70
6.4 Zeitlicher Ablauf 71
6.4.1 Beispiel zum *synchronen* Starten 71
6.4.2 Timing-Diagram eines A/D-D/A-Transfers 72
6.5 Buffer-Interrupts 73

7. Buffer-Statistik 75
7.1 Aufbau der Statistik-Struktur 75
7.2 Adressen der Strukturzellen 75

1. Einführung

Die Firmware der VME-AIO16 unterstützt das Setzen der D/A-Wandler und das Auslesen der A/D-Wandler-Werte. Nach jedem Systemstart führt die Firmware einen Selbsttest durch, der auch die Berechnung des Gain- und des Offset-Fehlers der Eingangsschaltung beinhaltet. Daher ist es möglich, dem Anwender bereits korrigierte Werte zur Verfügung zu stellen.

Daneben können weitere Parameter wie die Trigger-Bedingungen der A/D-Wandlung, die CPU-Taktfrequenz, die Timer-Periode oder die Ausgangswerte der D/A-Wandler nach dem Systemstart übergeben werden.

Die Programmierung erfolgt ausschließlich über den VMEbus.

Die Kommandos und Parameter werden durch Setzen von Speicherzellen im Shared-RAM-Bereich (Parameter-Buffer) und Triggern eines Commander-Interrupts übergeben. Die Übergabestruktur mit Semaphore gewährleistet die konfliktfreie Zugriffsmöglichkeit durch die lokale CPU und den VMEbus-Master.

Diese Seite ist bewußt unbedruckt.

2. Default-Parameter

Die AIO16 ist mit einem I²C-EEPROM ausgestattet, in dem Parameter abgelegt werden können.

Bei Auslieferung der AIO16 ist das lokale I²C-EEPROM mit den in den folgenden Kapiteln aufgeführten Default-Parametern besetzt. Der Anwender hat die Möglichkeit, diese Parameter zu ändern und die geänderten Parameter wieder im I²C-EEPROM abzuspeichern. Die neuen Parameter bleiben so auch nach einem RESET erhalten.

Die Firmware überprüft bei jedem Hochlauf das I²C-EEPROM. Sollten hierbei Fehler festgestellt werden, so wird das EEPROM wieder mit den ursprünglichen Default-Parametern besetzt, die ab Werk eingestellt waren.

Der Anwender kann das I²C-EEPROM auch durch das Kommando ‘System-Neustart mit Default-Parametern’ löschen und so die ursprünglichen Parameter wiederherstellen. Dabei gehen alle Parameteränderungen verloren.

Das Kommando ‘System-Neustart mit Default-Parametern’ wird gemeinsam mit den anderen Kommandos in einem späteren Kapitel ausführlich erläutert.

Diese Seite ist bewußt unbedruckt.

3. Shared-Adreßbereich

Der Shared-Adreßbereich bildet die Schnittstelle zwischen der lokalen CPU und dem VMEbus. Auf diese Zellen können der VMEbus-Master und die lokale CPU zugreifen.

3.1 Hinweise zu VMEbus-Zugriffen

Die VMEbus-Basisadresse ist bei Auslieferung der Karte auf

$xx680000

eingestellt.

Zugriffe über den **VMEbus** auf den Shared-Adreßbereich können nur auf jeder zweiten Word-Adresse erfolgen. Daher sind die relativen VMEbus-Adressen in den folgenden Tabellen in einer eigenen Spalte aufgeführt. Die absolute Adresse über den VMEbus errechnet sich wie folgt:

absolute_VME_Address = VME_Base_Address + VMEbus_relative_Address

3.2 Hinweise zu Zugriffen der lokalen CPU

Die Basisadresse für den Shared-Adreßbereich läßt sich in den Registern der CPU 68340 programmieren. Sie ist zur Zeit auf

$0080.0000

eingestellt.

Die absolute lokale Adresse errechnet wie folgt:

absolute_local_Address = local_Base_Address + local_relative_Address

3.3 VMEbus-Adressen und Adressen der lokalen CPU

In der folgenden Tabelle sind in der ersten Spalte die relativen Adressen der lokalen CPU und in der zweiten Spalte die Adressen relativ zur VMEbus-Basis-Adresse der AIO16 aufgeführt.

Zugriff über VMEbus: VMEbus-Base + Adreß-Offset [HEX]	Zugriff über lokale CPU `$0080.0000` + Adreß-Offset [HEX]	Bauteil/Signal	
		Schreibzugriff	Lesezugriff

`7FF7C`
`:`
`00000`
	`3FFBE`
	`:`
	`00000`
		256 kByte Shared SRAM abzüglich AD/DA-Wandler-Bereich, 16 Bit Datenbreite, auf VMEbus über jede zweite WORD-Adresse zugänglich	

`7FF80`	`3FFC0`	D/A-Wandler 1 setzen ohne SWLDAC	D/A-Wandler 1: Wert aus RAM-Zelle 1.1 auslesen
`7FF84`	`3FFC2`	D/A-Wandler 2 setzen ohne SWLDAC	D/A-Wandler 2: Wert aus RAM-Zelle 2.1 auslesen
`7FF88`	`3FFC4`	D/A-Wandler 3 setzen ohne SWLDAC	D/A-Wandler 3: Wert aus RAM-Zelle 3.1 auslesen
`7FF8C`	`3FFC6`	D/A-Wandler 4 setzen ohne SWLDAC	D/A-Wandler 4: Wert aus RAM-Zelle 4.1 auslesen
`7FFA0`	`3FFD0`	Aux_In_8	
`7FFA4`	`3FFD2`	Aux_In_7	
`7FFA8`	`3FFD4`	Aux_In_6	
`7FFAC`	`3FFD6`	Aux_In_5	
`7FFB0`	`3FFD8`	Aux_In_4	
`7FFB4`	`3FFDA`	Aux_In_3	
`7FFB8`	`3FFDC`	Aux_In_2	
`7FFBC`	`3FFDE`	Aux_In_1	

Zugriff über VMEbus: VMEbus-Base + Adreß-Offset [HEX]	Zugriff über lokale CPU `$0080.0000+` Adreß-Offset [HEX]	Bauteil/Signal	
		Schreibzugriff	Lesezugriff
`7FFC0`	`3FFE0`	SWLDAC aktivieren	reserviert
`7FFC4`	`3FFE2`	SWLDAC aktivieren	reserviert
`7FFC8`	`3FFE4`	SWLDAC aktivieren	reserviert
`7FFCC`	`3FFE6`	SWLDAC aktivieren	reserviert
`7FFD0`	`3FFE8`	SWLDAC aktivieren	reserviert
`7FFD4`	`3FFEA`	SWLDAC aktivieren	reserviert
`7FFD8`	`3FFEC`	SWLDAC aktivieren	reserviert
`7FFDC`	`3FFEE`	SWLDAC aktivieren	reserviert
`7FFE0`	`3FFF0`	SWCONV (A/D-Wandlung starten)	reserviert
`7FFE4`	`3FFF2`	DACDMAREQ aktivieren (noch nicht implementiert)	reserviert
`7FFE8`	`3FFF4`	SWCOM (lokaler IRQ von VMEbus ausgelöst)	reserviert
`7FFEC`	`3FFF6`	WRIVEC (Write VMEbus-IRQ-Vector)	reserviert
`7FFF0`	`3FFF8`	nicht belegt	reserviert
`7FFF4`	`3FFFA`	nicht belegt	reserviert
`7FFF8`	`3FFFC`	nicht belegt	reserviert
`7FFFC`	`3FFFE`	nicht belegt	reserviert

Tabelle 3.2.1: Belegung des Shared-Areas mit Adressen des VMEbus und der lokalen CPU

Diese Seite ist bewußt unbedruckt.

4. Kommandos und Parameterübergabe

4.1 Belegung des Parameter-Buffers

Der Parameter-Buffer im Shared-RAM läßt sich in verschiedene Funktionsgruppen aufteilen. Die folgende Tabelle gibt eine Übersicht über diese Gruppen mit Hinweisen zu den einzelnen Kapiteln, in denen sie im folgenden beschrieben werden.

relativer Adreßbereich VMEbus [HEX]	relativer Adreßbereich lokal [HEX]	Funktionsgruppe	Beschreibung ab Seite:
0000 ... 001E	0000 ... 000F	Karten-Identifikation	13
0020 ... 003E	0010 ... 001F	Karten-Status	14
0040 ... 004E	0020 ... 0027	Kommando-Sektion für Parameterübergabe	17
00C0 ... 00DE	0060 ... 006F	Preloads	27
0100 ... 011E	0080 ... 008F	A/D-Buffer-Status-Struktur	69
0120 ... 013E	0090 ... 009F	D/A-Buffer-Status-Struktur	69
0140 ... 015E	00A0 ... 00AF	Steuerparameter	31
0160 ... 017E	00B0 ... 00BF	Selective-D/A-Buffer-Status	32
01F0 ... 01FE	00F8 ... 00FF	A/D-Wandler-Status	47
0200 ... 023C	0100 ... 011F	A/D-Wandler: Rohdaten	49
0240 ... 027C	0120 ... 013F	A/D-Wandler: Korrigierte Daten	51
0280 ... 02F8	0140 ... 017F	A/D-Wandler: Summierte 32-Bit-Werte	53
0300 ... 031C	0180 ... 018F	Aux_In_1-8: Rohdaten	62
0320 ... 033C	0190 ... 019F	Aux_In_1-8: Korrigierte Daten	63
0400 ... 043C	0200 ... 021F	A/D-Wandler: Offset-Werte	54
0440 ... 047C	0220 ... 023F	A/D-Wandler: +5V-Referenz-Daten	55
0480 ... 04BC	0240 ... 025F	D/A-Wandler: Offset-Werte (info only)	57
04C0 ... 04FC	0260 ... 027F	D/A-Wandler: Gain-Werte (info only)	58
0500 ... 053C	0280 ... 029F	A/D-Wandler: Gain-Korrektur-Faktoren	60
0540 ... 05B8	02A0 ... 02DF	A/D-Wandler: 32-Bit Offset-Werte der A/D-Wandler-Summation	61
05C0 ... 05DC	02E0 ... 02EF	Aux_In_1-8: Offset-Werte	64
05E0 ... 05FC	02F0 ... 02FF	Aux_In_1-8: Gain-Werte	65

relativer Adreßbereich VMEbus [HEX]	relativer Adreßbereich lokal [HEX]	Funktionsgruppe	Beschreibung ab Seite:
0800 ... xxxx	0400 ... yyyy	A/D-Buffer (Größe eingeschränkt definierbar)	67
uuuu ... 7FDFE	vvvv ... 3FEFF	D/A-Buffer (Größe eingeschränkt definierbar)	67

Die Tabellen in den anschließenden Kapiteln zeigen die für den Anwender relevanten Zellen des Parameter-Buffers mit ihren relativen Adressen im Shared-RAM. Alle Zellen bis auf `cmmd`, `adstat`, `para` und `sema` dürfen nur gelesen werden. Sie können indirekt, über die `cmmd`-Kommandos und deren Parameter, gesetzt werden.

4.2 Karten-Identifikation

Auf diesen Speicherzellen ist als ASCII-Code die Bezeichnung der AIO16 abgelegt.

relative Adresse VMEbus [HEX]	relative Adresse lokal [HEX]	Länge	Zugriff	Zelleninhalt	Kurzbeschreibung
0...1E	0...F	16 Bytes	read only	`ASCII-Id`	Identifikation der AIO16 über ASCII-Text

Als Identifikationstext wird zur Zeit als ASCII-Code

`'esd_AIO16_LevX.Y'`

zurückgegeben. Dabei wird für '`X`' und '`Y`' die aktuelle Revisionsnummer der Firmware angegeben.

4.3 Karten-Status

Diese Zelle gibt den aktuellen Status des Commanders wieder. Sie sollte nach einem Systemstart vor dem ersten Kommandoaufruf gelesen werden, um zu überprüfen, ob der Selbsttest abgeschlossen, und erfolgreich verlaufen ist.

relative Adresse VMEbus [HEX]	relative Adresse lokal [HEX]	Länge	Zugriff	Parameter-Name	Parameter-Werte [HEX]
20	10	WORD	read only	`card_stat`	Wertebereich: $0000...$8001
					$0001 ... $000n Fehler beim Selbsttest aufgetreten (esd-interner Fehlerstatus)
					$7FFF AIO16 befindet sich im Selbsttest
					$8001 Selbsttest fehlerfrei abgeschlossen

4.4 Hardware-Revision

Diese Zelle enthält Aussagen über die Hardware-Revision des VME-AIO16-Boards.

relative Adresse VMEbus [HEX]	relative Adresse lokal [HEX]	Länge	Zugriff	Parameter-Name	Parameter-Werte [HEX]
24	12	WORD	read only	`HWrev`	Wertebereich: $0000, $0001 $0000 alte Hardware $0001 neue Hardware (mit Fast SRAM)

4.5 Vorgehensweise beim Kommandoaufruf

Das Auslesen der Speicherzellen (z.B. A/D-Werte) kann zu jeder Zeit transparent erfolgen. Beim Setzen von Parametern und Kommandos ist jedoch der folgende Ablauf einzuhalten:

(1.) Nur bei Multi-Master- oder Multi-Tasking-Anwendungen notwendig:
Auswertung der Zelle sema
Über die Zelle `sema` wird abgefragt, ob der Parameter-Buffer frei ist. Ist dies der Fall muß sema besetzt werden ('TAS'-Befehl)

2. Lesen der Zelle `cmmd` (Command)
Ist der Commander frei, d.h. die Ausführung des letzten Kommandos ist abgeschlossen, so wird bei einem Lesezugriff (Word) auf die relative Adresse $22 der Wert '0000' zurückgegeben. Nur wenn dieser Wert gelesen wurde, darf das gewünschte Kommando mit seinen Parametern in die Zellen des Shared-RAM geschrieben werden. Andernfalls ist zu warten, bis der Commander frei ist.

3. Setzen des Kommandos und der Parameter
Sofern die Bedingungen unter Punkt 2 erfüllt sind, kann jetzt ein Kommando mit den zugehörigen Parametern im Shared-RAM gesetzt werden.

4. Triggern des Commander-Interrupts
Durch einen Schreibzugriff vom VMEbus aus mit beliebigen Daten auf die relative Wort-Adresse $7FFE8 wird ein Interrupt der lokalen CPU ausgelöst. Die CPU übernimmt daraufhin das Kommando und führt es aus.

5. Warten auf Ende der Kommandoabarbeitung
Hat die lokale CPU die Zelle `cmmd` = 0000 gesetzt, so ist das Kommando abgeschlossen.

6. Auswerten der Zelle `cstat`
Hier kann kontrolliert werden, ob das Kommando erfolgreich ausgeführt wurde (nach `cmmd` = 0).
(`cstat` = 0 -> OK, `cstat` ≠ 0 -> Fehler)

(7.) Nur bei Multi-Master- oder Multi-Tasking-Anwendungen notwendig:
Freigeben der Zelle `sema`
Sollen keine weiteren Kommandos ausgeführt werden, muß die Zelle sema wieder freigegeben werden.

Hinweis: Die zu erwartenden Abarbeitungszeiten der aufgerufenen Kommandos hängen wesentlich von der Art des Kommandos ab:

- für Kommandos im Bereich von $0xxx : ≈100 µs
- für Kommandos im Bereich von $8xxx : mehrere Millisekunden, (kommandoabhängig)

4.6 Kommando-Sektion

Diese Zellen dienen der Semaphorensteuerung und der Kommandoübergabe. Das Setzen von Zellen im Parameter-Buffer darf nur über den Kommandoaufruf mit `cmmd` und `para` erfolgen! Die einzige Ausnahme bildet die Zelle `sema` (siehe unten).

relative Adresse VMEbus [HEX]	relative Adresse lokal [HEX]	Länge		Zugriff	Zelleninhalt	Kurzbeschreibung
40	20	BYTE		read only	`cstat`	Commander-Status
41	21	BYTE		read/write	`sema`	Semaphore
44	22	WORD		read/write	`cmmd`	Kommando
48	24	WORD (para1.w)	LWORD	read/write	`para`	Parameter zu Kommando
4C	26	WORD (para2.w)				
50	28	WORD (para3.w)				

Die Übergabe der Parameter erfolgt in Abhängigkeit von der Länge wie folgt:

relative Adressen -> [HEX]	VMEbus	48	4C	50
	lokal	24	26	28
`para` ->	WORD-Parameter	WORD **Parameter 1**	WORD **Parameter 2**	WORD **Parameter 3**
	LWORD-Parameter	Upper WORD of LONG Parameter	Lower WORD of LONG Parameter	-

4.6.1 Die Zellen `sema` und `cstat`

Die Zelle `sema` wird von der lokalen Software nicht gesetzt, sie ist der Semaphorensteuerung über den VMEbus vorbehalten. Lediglich nach einem Selbsttest wird die Zelle mit $00 vorbesetzt.

Die Zelle `sema` muß vom VMEbus-Master abgefragt und gesetzt werden, bevor ein Kommandoaufruf erfolgt. Durch diese Lesen/Setzen-Sequenz kann der Anwender sicherstellen, daß nur ein Master, oder eine Task auf den Commander der AIO16 zur gleichen Zeit zugreift.

Nach dem Aufruf kann die Zelle `cstat` ausgewertet werden, um den Erfolg des Zugriffs zu überprüfen.

Adresse [HEX]	Länge	Parameter-Name	Parameter-Werte [HEX]
41	Byte	`sema`	Wertebereich: $0x, $8x
			Nur Bit 7 setzen und auswerten: 0xxx xxxx Commander ist frei 1xxx xxxx Commander belegt

Von der Zelle `sema` sollte nur Bit 7 gesetzt und ausgewertet werden. Die Bits 6 bis 0 sind für zukünftige Anwendungen reserviert.

In die Zelle `cstat` trägt der lokale Server ein, ob das übergebene Kommando erfolgreich ausgeführt wurde.

Adresse [HEX]	Länge	Parameter-Name	Parameter-Werte [HEX]
40	Byte	`cstat`	Wertebereich: $00, $FF
			= 0 kein Fehler ≠ 0 Fehler

4.6.2 Allgemeines zur Kommandozelle `cmmd`

In der Zelle `cmmd` werden die verschiedenen Kommandos gesetzt. Ist ein Kommando abgearbeitet, so besetzt die Firmware die Zelle mit '0000'. Dadurch wird das Ende der Kommandoabarbeitung angezeigt.

4.6.3 Übersicht der Kommandos und setzbaren Parameter

Das Setzen der Parameter erfolgt durch Eintrag eines Kommandos in der Zelle `cmmd` und Übergabe des Parameters in der Zelle `para` im Parameter-Buffer. Die Eingabesequenz ist in dem vorangegangenen Kapitel 'Vorgehensweise beim Kommandoaufruf' bereits beschrieben worden.

Dieses Kapitel erläutert die einzelnen Kommandos mit ihren Parametern und deren Wertebereichen.

Die folgenden Tabellen zeigen eine Übersicht aller bisher implementierten Kommandos mit ihren Parametern.

`cmmd` [HEX]	Kommando-Bezeichnung	`para`	Status lesbar auf Adresse [HEX]	Kurzbeschreibung
8000	Aktuelle Parameter in I²C-EEPROM speichern	bei diesen 'System-Kommandos' werden keine Parameter übergeben	-	Parameter bleiben so auch nach RESET erhalten
8001	System-Neustart incl. Selbsttest (= RESET)			Parameter werden nach Neustart aus I²C-EEPROM entnommen
8002	System-Neustart mit Default-Parametern			Grundinitialisierung: I²C-EEPROM wird wieder mit Default-Parametern besetzt: -> alle geänderten Parameter gehen verloren!
8003	Statistik über aktuellen Buffer	2x WORD	`cstat`	Erstellen der Statistik; Parameter = Maske für die Auswahl der A/D-Wandler-Kanäle und der Auxilary-Inputs
8004	Buffer-Korrektur	3x WORD	`cstat`	Ausführen der Buffer-Korrektur; Parameter = Maske für die Auwahl der Offset- und/oder Gain-Korrektur der A/D-Wandler-Kanäle und der Auxilary-Inputs
8005	Mittelwertbildung	3x WORD	`cstat`	Mittelwertbildung Parameter = Maske für die Auswahl der A/D-Wandler-Kanäle und der Auxilary-Inputs zur Mittelwertbildung

cmmd [HEX]	para	Kommando-Bezeichnung	Status lesbar auf VME, bzw. (lokaler) Adresse [HEX]	Name der Statuszelle	Kurzbeschreibung
1	WORD	VMEbus-IRQ-Level setzen	140 (A0)	vmelev	Interrupt der AIO16 auf dem VMEbus
2	WORD	VMEbus-IRQ-Vektor setzen	141 (A1)	vmevec	Vektor für VMEbus-IACK-Zyklus
3	WORD	Multiplexer der A/D-Eingänge	144 (A2)	muxmode	wird von Firmware bei Abgleich automatisch gesetzt
4	WORD	Multiplexer der D/A-Ausgänge	145 (A3)	dacmode	wird von Firmware bei Abgleich automatisch gesetzt
5	WORD	Quelle für Trigger der A/D-Wandlung wählen	148 (A4)	trigmod	Wandlung kann per Software, Timer o. ext. Trigger gestartet werden
6	WORD	Quelle für LDAC der D/A-Wandler wählen	149 (A5)	ldcmod	LDAC dient zur Übernahme der neuen D/A-Werte
7	WORD	Auswahl der A/D-Datenverarbeitung	14C (A6)	vadsrv	DMA-Mode, Rückgabe korrigierter A/D-Werte, Summation von AD-Werten
8	WORD	Ersten A/D-Kanal zum Auslesen selektieren	14D (A7)	vstart	Auswahl der A/D-Kanäle, deren Daten von lokaler CPU in Shared-RAM aktualisiert werden sollen.
9	WORD	Letzten A/D-Kanal zum Auslesen selektieren	150 (A8)	vend	
A	WORD	Quelle für VMEbus-Interrupt	151 (A9)	vvtrg	Auswahl der Interrupt-Quelle für IRQ auf VMEbus
B	WORD	A/D-Wandler-Auflösung	154 (AA)	vadres	nicht für Anwenderprogrammierung
C	WORD	Auswahl der D/A-Datenverarbeitung	155 (AB)	vdasrv	Buffer-Modes der D/A-Wandler
D	WORD	Anzahl zu addierender A/D-Werte	156 (AC)	vsmcnt	Summation von A/D-Werten
E	2x WORD	A/D-Buffer einrichten	100 (80)	Struktur adbuf	Setzt die Anzahl der Frames/Buffer und die Anzahl der Buffer für die A/D-Kanäle
F	2x WORD	D/A-Buffer einrichten	120 (90)	Struktur dabuf	Setzt die Anzahl der Frames/Buffer und die Anzahl der Buffer für die D/A-Kanäle
10	WORD	Ersten D/A-Kanal für Buffer-Mode selektieren	15C (AE)	dastart	Auswahl der D/A-Kanäle, die im Buffer-Mode gesetzt werden sollen (für andere D/A-Betriebsarten ohne Bedeutung).
11	WORD	Letzten D/A-Kanal für Buffer-Mode selektieren	15E (AF)	daend	
12	2x WORD	Selective D/A-Buffer	160, 168 (B0, B4)	vdasb vdamsk	Auswahl des D/A-Buffers und der D/A-Kanäle
13...1F	-	reserviert	-	-	-

cmmd [HEX]	para	Kommando-Bezeichnung	Status lesbar auf VME, bzw. (lokaler) Adresse [HEX]	Name der Statuszelle	Kurzbeschreibung
20	WORD	D/A-Wert nach RESET für Kanal 1	D0 (68)	dac1p	Ausgangsspannung, die nach einem RESET ausgegeben werden soll
21	WORD	D/A-Wert nach RESET für Kanal 2	D4 (6A)	dac2p	
22	WORD	D/A-Wert nach RESET für Kanal 3	D8 (6C)	dac3p	
23	WORD	D/A-Wert nach RESET für Kanal 4	DC (6E)	dac4p	
24...2F	-	reserviert	-	-	-
30	LONG	Periodendauer für Trigger der Wandlung über Timer vorgeben	C8 (64)	cnvtime	beim Lesen steht in cnvtime der reell erreichbare Wert
31	LONG	CPU-Frequenz vorgeben	C0 (60)	tifreq	in tifreq kann die aktuelle CPU-Frequenz/2 gelesen werden

4.7 Beschreibung der System-Kommandos

Hier werden zuerst die 'System-Kommandos' erläutert. Diese Kommandos werden mit negativen Werten ($8xxx) übergeben. Sie führen übergeordnete Funktionen aus und benötigen daher für die Ausführung mehr Zeit als die anderen Kommandos. In der Regel werden sie nur einmalig bei der Grundinitialisierung benötigt.

Im Anschluß daran folgt die Beschreibung der Kommandos, bei denen Parameter übergeben werden. Die Default-Werte und die neu übergebenen Parameter können im Parameter-Buffer gelesen werden.

4.7.1 Aktuelle Parameter im I²C-EEPROM speichern

cmmd [HEX]	para	Status lesbar auf Adresse [HEX]	Name der Statuszelle	Wertebereich des Parameters
8000	-	-	-	keine Parameterübergabe

Sollen die geänderten Parameter auch nach einem RESET oder einem Software-Neustart erhalten bleiben, so sind sie im I²C-EEPROM abzuspeichern. Sollte das I²C-EEPROM danach wieder gelöscht werden, um die ursprünglichen Default-Parameter, die bei Auslieferung der Platine eingestellt waren, wieder zu aktivieren, so ist das Kommando 'System-Neustart mit Default-Parametern' aufzurufen (siehe folgende Kapitel).

Achtung: Es können nicht alle Parameter im I²C-EEPROM gespeichert werden. Die Parameter der Kommandos '7', 'C', 'E' und 'F' werden z.B. nicht gespeichert. Bei Parametern, die nicht gespeichert werden können, ist dies in der Kommandobeschreibung mit aufgeführt.

4.7.2 System-Neustart incl. Selbsttest

cmmd [HEX]	para	Status lesbar auf Adresse [HEX]	Name der Statuszelle	Wertebereich des Parameters
8001	-	-	-	keine Parameterübergabe

Mit diesem Kommando kann über den VMEbus per Software ein 'RESET' auf der AIO16 ausgelöst werden. Alle Parameter, die nicht vorher im I²C-EEPROM gespeichert worden sind, gehen verloren. Die aktuellen AD/DA-Daten im Shared-RAM gehen ebenfalls verloren. Die Firmware startet neu und auf der Platine wird ein Selbsttest durchgeführt. Der Selbsttest beinhaltet die automatische Bestimmung des Offsets und des Gains der Eingangskanäle.

4.7.3 System-Neustart mit Default-Parametern

cmmd [HEX]	para	Status lesbar auf Adresse [HEX]	Name der Statuszelle	Wertebereich des Parameters
8002	-	-	-	keine Parameterübergabe

Der System-Neustart mit Default-Parametern läuft wie der in vorherigen Abschnitt besprochene Neustart ab. Zusätzlich werden die bei Auslieferung der Platine eingestellten Default-Parameter wieder im I²C-EEPROM abgespeichert. Alle vorherigen Änderungen der Parameter gehen verloren!

4.7.4 Statistik über aktuellen Buffer

cmmd [HEX]	para	Status lesbar auf VMEbus Adresse [HEX]	Status lesbar auf lokaler Adresse [HEX]	Name der Statuszelle	Name des Parameters	Parameter-Werte [HEX]
8003	WORD (para1.w)	40	20	cstat	auxen_stat	Wertebereich: 0000...00FF
	WORD (para2.w)	40	20	cstat	adcen_stat	Wertebereich: 0000...FFFF

Mit diesem Kommando werden die Eingangskanäle für die statistische Auswertung ausgewählt (siehe Kapitel ‘Buffer-Statistik’ ab Seite 75).

Der Erfolg der Kommandoübergabe kann durch Auswertung der Zelle `cstat` geprüft werden.

Gesetzt werden bei diesem Kommando die Parameter `auxen_stat` und `adcen_stat`:

Die Parameter sind wie folgt belegt:

Bit von auxen_stat	Auxilary-Input
0	Enable_Aux_In_1
1	Enable_Aux_In_2
2	Enable_Aux_In_3
:	:
7	Enable_Aux_In_8
8	-
:	:
15	-

Bit von adcen_stat	A/D-Wandler-Kanal
0	Enable_A/D_1
1	Enable_A/D_2
2	Enable_A/D_3
:	:
:	:
:	:
:	:
15	Enable_A/D_16

Wird ein Bit auf ‘1’ gesetzt, so ist der entsprechende Eingangskanal freigeschaltet.

4.7.5 Buffer-Korrektur

cmmd [HEX]	para	Status lesbar auf VMEbus Adresse [HEX]	Status lesbar auf lokaler Adresse [HEX]	Name der Statuszelle	Name des Parameters	Parameter-Werte [HEX]
8004	WORD (para1.w)	40	20	cstat	auxen_kor	Wertebereich: 0000...00FF
	WORD (para2.w)	40	20	cstat	adcen_kor	Wertebereich: 0000...FFFF
	WORD (para3.w)	40	20	cstat	off_gain	Wertebereich: 0000...FFFF

Mit diesem Kommando werden die Eingangskanäle ausgewählt, für deren Werte eine Offset oder/und eine Gain-Korrektur vorgenommen werden soll.

Der Erfolg der Kommandoübergabe kann durch Auswertung der Zelle `cstat` geprüft werden.

Gesetzt werden bei diesem Kommando die Parameter `auxen_kor, adcen_kor` und `off_gain`:

Die Parameter `auxen_kor` und `adcen_kor` sind wie folgt belegt:

Bit von auxen_kor	Auxilary-Input
0	Enable_Kor_Aux_In_1
1	Enable_Kor_Aux_In_2
2	Enable_Kor_Aux_In_3
:	:
7	Enable_Kor_Aux_In_8
8	-
:	:
15	-

Bit von adcen_kor	A/D-Wandler-Kanal
0	Enable_Kor_A/D_1
1	Enable_Kor_A/D_2
2	Enable_Kor_A/D_3
:	:
:	:
:	:
:	:
15	Enable_Kor_A/D_16

Wird ein Bit auf ‘1’ gesetzt, so ist der entsprechende Eingangskanal freigeschaltet.

Parameter `off_gain`:

Mit diesem Parameter wird ausgewählt, ob für die über `auxen_kor` und `adcen_kor` ausgewählten Kanäle nur eine Offset-Korrektur oder eine Offset- und eine Gain-Korrektur durchgeführt werden soll:

`off_gain` = 0 : nur Offset-Korrektur durchführen
`off_gain` ≠ 0 : Offset- und Gain-Korrektur durchführen

4.7.6 Mittelwertbildung

cmmd [HEX]	para	Status lesbar auf VMEbus Adresse [HEX]	Status lesbar auf lokaler Adresse [HEX]	Name der Statuszelle	Name des Parameters	Parameter-Werte [HEX]
8005	WORD (para1.w)	40	20	cstat	auxen_avr	Wertebereich: 0000...00FF
	WORD (para2.w)	40	20	cstat	adcen_avr	Wertebereich: 0000...FFFF
	WORD (para3.w)	40	20	cstat	avr_cnt	Wertebereich: 0000...FFFF

Mit diesem Kommando werden die Eingangskanäle ausgewählt, für die eine Mittelwertbildung erfolgen soll.

Der Erfolg der Kommandoübergabe kann durch Auswertung der Zelle `cstat` geprüft werden.

Gesetzt werden bei diesem Kommando die Parameter `auxen_avr, adcen_avr` und `avr_cnt`:

Die Parameter `auxen_avr` und `adcen_avr` sind wie folgt belegt:

Bit von auxen_avr	Auxilary-Input
0	Enable_Average_Aux_In_1
1	Enable_Average_Aux_In_2
2	Enable_Average_Aux_In_3
:	:
7	Enable_Average_Aux_In_8
8	-
:	:
15	-

Bit von adcen_avr	A/D-Wandler-Kanal
0	Enable_Average_A/D_1
1	Enable_Average_A/D_2
2	Enable_Average_A/D_3
:	:
:	:
:	:
:	:
15	Enable_Average_A/D_16

Wird ein Bit auf ‘1’ gesetzt, so ist der entsprechende Eingangskanal freigeschaltet.

Parameter `avr_cnt`:

Mit diesem Parameter wird ausgewählt, wie viele Werte für die Mittelwertbildung herangezogen werden:

`avr_cnt` = 1, 3, 5, ...

4.8 Preload

In diesen Zellen können die Periodendauer für den Trigger der Wandlungen über Timer, die CPU-Frequenz und die D/A-Wandler-Werte, die nach einem Systemstart an den Ausgängen anliegen sollen, vorgegeben werden.

4.8.1 Übersicht der lesbaren Zellen des Parameter-Buffers

Die folgende Tabelle zeigt die Speicherzellen des Parameter-Buffers, in denen die aktuellen Preload-Werte *gelesen* werden können.

Das Setzen der Werte muß per Kommandoaufruf über die Zellen `cmmd` und `para` erfolgen!

relative Adresse VMEbus [HEX]	relative Adresse lokal [HEX]	Länge der Status-zelle	Zugriff	Name der Statuszelle	Kurzbeschreibung
C0	60	Upper WORD	read only	`tifreq`	Timer-Input-Frequenz [Hz] (= CPU-Frequenz : 2)
C4	62	Lower WORD			
C8	64	Upper WORD	read only	`cnvtime`	aktuelle Conversion Time in [ns]
CC	66	Lower WORD			
D0	68	WORD	read only	`dac1p`	D/A-Wandler 1: Wert nach RESET
D4	6A	WORD	read only	`dac2p`	D/A-Wandler 2: Wert nach RESET
D8	6C	WORD	read only	`dac3p`	D/A-Wandler 3: Wert nach RESET
DC	6E	WORD	read only	`dac4p`	D/A-Wandler 4: Wert nach RESET

4.8.1.1 Lesen der Timer-Input-Frequenz (`tifreq`)

In der Zelle `tifreq` wird die sogenannte Timer-Input-Frequenz zurückgegeben. Diese Frequenz entspricht der halben CPU-Frequenz. Die Angabe erfolgt als 32-Bit-Wert in Hz (nicht wie die Eingabe der CPU-Frequenz in der Zelle `para` als 16-Bit-Wert in kHz!).

```
                  para [kHz]
tifreq [Hz] = ------------- · 1000
                     2
```

Beispiel: Bei der Default-Einstellung der CPU-Frequenz mit 25.165,00 kHz (= $624D kHz) würde der Wert $C00000 zurückgegeben werden.

Die Adressen der Zelle `tifreq` sind in der Übersichtstabelle aufgeführt.

4.8.1.2 Lesen der aktuellen Conversion Time (`cnvtime`)

In dieser Zelle wird die aktuelle Wandlungszeit in [ns] zurückgegeben. Nach dem Setzen der Wandlungszeit über das Kommando `cmmd` = $30 errechnet die lokale Software die tatsächlich erreichbare Wandlungszeit und trägt sie in diese Zelle ein.

Der Wertebereich dieses Parameters ist abhängig von der gewählten CPU-Frequenz (siehe Kapitel ‘Trigger-Periodendauer über Timer vorgeben (`cnvtime`)’)

4.8.1.3 Lesen der ‘D/A-Wandler-Werte nach RESET’ (`dac1p, dac2p...`)

Die Default-Werte der D/A-Wandler, die nach einem RESET angenommen werden sollen, werden über die Kommandos $20...$23 programmiert. Sie können auf den relativen VMEbus-Adressen $D0...$DC gelesen werden (siehe vorangegangene Tabelle).

4.8.2 Ausgangswerte der D/A-Wandler nach RESET (`dacxp`)

Über dieses Kommando werden die Ausgangsspannungen der D/A-Wandler-Kanäle vorgegeben, die nach einem Systemstart an den Ausgängen an P2 anliegen sollen.

`cmmd` [HEX]	para	Status lesbar auf VMEbus Adresse [HEX]	Status lesbar auf lokaler Adresse [HEX]	Name der Statuszelle	Parameter-Werte [HEX]	
					Wertebereich: 8000...7FFF	Default-Wert: 0000
20	WORD	D0	68	`dac1p`	8000	U_{OUT} = -10 V
21	WORD	D4	6A	`dac2p`	:	:
22	WORD	D8	6C	`dac3p`	0000	U_{OUT} = 0 V
23	WORD	DC	6E	`dac4p`	:	:
					7FFF	U_{OUT} = 10 V - 1 LSB

4.8.3 Trigger-Periodendauer über Timer vorgeben (`cnvtime`)

Hier kann die gewünschte Periodendauer für das Starten der Wandlungen über den CPU-Timer in Nanosekunden vorgegeben werden. Die Firmware errechnet daraus die reell mögliche Wandlungszeit und trägt diesen Wert in die Zelle `cnvtime` ein.

`cmmd` [HEX]	para	Status lesbar auf VMEbus Adresse [HEX]	Status lesbar auf lokaler Adresse [HEX]	Name der Statuszelle	Parameter-Werte [HEX]	
					Wertebereich: $0000.4E20 ... T_{MAX}	Default-Wert: $0002.49E2
30	LONG	C8, CC	64, 66	`cnvtime`	$0000.4E20	-> 20 µs [1*)]
					:	:
					$0002.49E2	-> 149,9 µs
					:	:
					$004F.7967	-> 5,209 ms [2*)]

Der Maximalwert T_{MAX}, der für die Wandlungszeit eingetragen werden kann, ist von der gewählten CPU-Frequenz abhängig. Er errechnet sich wie folgt:

$$T_{MAX}\ [ns] = \frac{10^{9} \cdot 2^{17}}{CPU_Frequenz\ [Hz]}$$

1*) Im Buffer -Mode kann die Periodendauer bis auf 10 µs verringert werden, wenn nur die Daten eines A/D-Wandler- und eines D/A-Wandler-Kanals bearbeitet werden.

2*) Hier ist die maximale Periodendauer für die Default-Einstellung der CPU-Frequenz (25.165.000 Hz) angegeben.

4.8.4 CPU-Frequenz vorgeben

Über dieses Kommando kann der Anwender die gewünschte CPU-Taktfrequenz in kHz setzen. Die Firmware errechnet daraus einen reell möglichen Wert, der dem gewünschten so weit als möglich angenähert ist. **Dieser Wert kann nicht zurückgelesen werden. Es kann jedoch in der Zelle `tifreq` der Wert 'CPU-Frequenz / 2' (in Hz) gelesen werden.**

Die neue Taktfrequenz wird erst dann übernommen, wenn die Parameter im I²C-EEPROM abgespeichert werden und danach eine RESET ausgelöst wird!

cmmd [HEX]	para	Status lesbar auf VMEbus Adresse [HEX]	Status lesbar auf lokaler Adresse [HEX]	Name der Statuszelle	Parameter-Werte [HEX]	
					Wertebereich: $0000...61A8	Default-Wert: $5A1C
31	WORD	C0, C4	60, 62	`tifreq·2`	$0000 : $3E80 : $5A1C : $61A8	-> 0 **k**Hz : -> 16.000 **k**Hz : -> 23.068 **k**Hz : -> 25.000 **k**Hz

4.9 Steuerparameter

In diesem Bereich des Parameter-Buffers sind Steuerparameter zu den A/D- und D/A-Kanälen und zum Interrupt-Handling abgelegt.

4.9.1 Übersicht der lesbaren Zellen des Parameter-Buffers (Steuerparameter)

Die folgende Tabelle zeigt die Speicherzellen des Parameter-Buffers, in denen die Werte der Steuerparameter gelesen werden können.

Das Setzen der Werte muß per Kommandoaufruf über die Zellen `cmmd` und `para` erfolgen!

relative Adresse VMEbus [HEX]	relative Adresse lokal [HEX]	Länge der Status-zelle	Zugriff	Namen der Statuszellen	Kurzbeschreibung
140	A0	BYTE	read only	`vmelev`	aktueller VME-Interrupt-Level
142	A1	BYTE	read only	`vmevec`	VMEbus-Interrupt-Vektor
144	A2	BYTE	read only	`muxmode`	Multiplexer der A/D-Eingänge
146	A3	BYTE	read only	`dacmode`	Multiplexer der D/A-Ausgänge
148	A4	BYTE	read only	`trigmod`	Trigger-Mode für A/D-Eingänge
14A	A5	BYTE	read only	`ldcmod`	Multiplexer für LDAC-Signal
14C	A6	BYTE	read only	`vadsrv`	Auswahl der A/D-Datenverarbei-tung
14E	A7	BYTE	read only	`vstart`	Auswahl der A/D-Kanäle, deren Werte in RAM transferiert werden sollen
150	A8	BYTE	read only	`vend`	
152	A9	BYTE	read only	`vvtrg`	Bedingung für VME-Interrupt
154	AA	BYTE	read only	`vadres`	A/D-Wandler-Auflösung setzen
156	AB	BYTE	read only	`vdasrv`	Auswahl der D/A-Datenverarbei-tung
158	AC	WORD	read only	`vsmcnt`	Anzahl zu addierender A/D-Werte
15C	AE	BYTE	read only	`dastart`	ersten D/A-Kanal für Buffer -Mode selektieren

relative Adresse VMEbus [HEX]	relative Adresse lokal [HEX]	Länge der Status-zelle	Zugriff	Namen der Statuszellen	Kurzbeschreibung
15E	AF	BYTE	read only	`daend`	letzten D/A-Kanal für Buffer-Mode selektieren
100	80	LONG	read only	Struktur `adbuf`	A/D-Buffer einrichten (lesbar ist die A/D-Buffer-Status-Struktur)
120	90	LONG	read only	Struktur `dabuf`	D/A-Buffer einrichten (lesbar ist die D/A-Buffer-Status-Struktur)
160	B0	LONG	read only	`vdasb`	D/A-Buffer-Base
168	B4	WORD	read only	`vdamsk`	D/A-Enable-Word

1FC
1F8
1F4
1F0
	FE
	FC
	FA
	F8
		WORD	read/ write !	`adstat0`
				`adstat1`
				`adstat2`
				`adstat3`
					Status der A/D-Daten im RAM (Achtung: Diese Zellen werden im Gegensatz zu den anderen direkt gesetzt!)

In den folgenden Kapiteln wird das Setzen der Zellen über Kommandoaufrufe und der Wertebereich der Parameter beschrieben.

4.9.2 VMEbus-Interrupt-Level setzen (`vmelev`)

Dieses Kommando setzt den Interrupt-Level des VMEbus-Interrupts, der von der VME-AIO16 ausgelöst werden kann.

`cmmd` [HEX]	para	Status lesbar auf VMEbus Adresse [HEX]	Status lesbar auf lokaler Adresse [HEX]	Name der Statuszelle	Parameter-Werte [HEX]	
1	WORD	140	A0	`vmelev`	Wertebereich: $0000...$0007	Default-Wert (Statuszelle): $05
					$0000 $0001 : $0007	kein VMEbus-Interrupt IRQ-Level 1 : IRQ-Level7

4.9.3 VMEbus-Interrupt-Vektor setzen (`vmevec`)

Dieses Kommando setzt den Interrupt-Vektor, der im IACK-Zyklus auf den VMEbus gegeben wird, wenn die VME-AIO16 einen IRQ ausgelöst hat.

`cmmd` [HEX]	para	Status lesbar auf VMEbus Adresse [HEX]	Status lesbar auf lokaler Adresse [HEX]	Name der Statuszelle	Parameter-Werte [HEX]	
2	WORD	141	A1	`vmevec`	Wertebereich: $0000...00FF	Default-Wert (Statuszelle): $000F
					$0000 : $00FF	Vector $00 : Vector $FF

4.9.4 Multiplexer der A/D-Eingänge setzen (`muxmode`)

Dieses Kommando wird von der Firmware während des automatischen Abgleichs gesetzt. Es sollte vom Anwender nicht verwendet werden!

`cmmd` [HEX]	para	Status lesbar auf VMEbus Adresse [HEX]	Status lesbar auf lokaler Adresse [HEX]	Name der Statuszelle	Parameter-Werte [HEX]	
3	WORD	144	A2	`muxmode`	Wertebereich: $0000...$0003	Default-Wert (Statuszelle): $00
					$0000 $0001 $0002 $0003	externe Signale an A/Ds DACs an A/D-Eingänge +5V-Ref. an A/D-Eing. GND an alle A/D-Eingänge

4.9.5 Multiplexer der D/A-Wandler-Ausgänge setzen (`dacmode`)

Dieses Kommando wird von der Firmware während des automatischen Abgleichs gesetzt. Nach dem Abgleich wird es auf ‘00’ gesetzt. Das Kommando sollte vom Anwender nicht verwendet werden!

`cmmd` [HEX]	`para`	Status lesbar auf VMEbus Adresse [HEX]	Status lesbar auf lokaler Adresse [HEX]	Name der Statuszelle	Parameter-Werte [HEX]	
4	WORD	145	A3	`dacmode`	Wertebereich: $0000...$0001	Default-Wert (Statuszelle): $00
					$0000 $0001	D/A-Ausgänge an P2 D/A-Kanäle zum internen Abgleich an A/D-Eingänge geschaltet

4.9.6 Trigger-Quelle für Start der A/D-Wandlung wählen (`trigmod`)

Hier kann ausgewählt werden, wie die Wandlung der A/D-Kanäle gestartet werden soll. Die Triggerung per Software erfolgt durch einen Schreibzugriff auf die relative lokale Wort-Adresse $7FFE0 (siehe Kap. 'Shared Adreßbereich'). Bei Triggerung über den CPU-Timer ist dieser entsprechend einzustellen (siehe Kommando 'Wandlungszeit für Timer vorgeben')

`cmmd` [HEX]	para	Status lesbar auf VMEbus Adresse [HEX]	Status lesbar auf lokaler Adresse [HEX]	Name der Statuszelle	Parameter-Werte [HEX]	
5	WORD	148	A4	`trigmod`	Wertebereich: $0000...$0002	Default-Wert (Statuszelle): $00
					$0000 Software-Trigger $0001 externer Trigger $0002 Trigger durch Timer	

4.9.7 Quelle für LDAC der D/A-Wandler wählen (`ldcmod`)

Durch Aktivierung des LDAC-Signals übernehmen die D/A-Wandler den neuen Wert an ihren Ausgängen.

`cmmd` [HEX]	`para`	Status lesbar auf VMEbus Adresse [HEX]	Status lesbar auf lokaler Adresse [HEX]	Name des Status-Wortes	Parameter-Werte [HEX]	
6	WORD	149	A5	`ldcmod`	Wertebereich: $0000...$0001	Default-Wert (Statuszelle): $01
					$0000	LDAC wird mit Laden der neuen D/A-Werte oder Schreibzugriff auf LDAC-Adresse gesetzt
					$0001	LDAC wird mit Convert-Start der A/D-Wandler gesetzt

Mode $0000:
Das LDAC-Signal (SWLDAC) kann sofort mit dem Laden der D/A-Wandler, oder separat durch einen Schreibzugriff auf eine Trigger-Adresse erfolgen. Letzteres hat den Vorteil, daß alle D/A-Wandler gleichzeitig ihren neuen Wert am Ausgang übernehmen. Die entsprechenden Adressen sind in dem Kap. ‘VMEbus-Adressen und Adressen der lokalen CPU’ bereits aufgeführt worden.

Mode $0001:
Das LDAC-Signal wird mit jedem neuen Convert-Start-Signal der A/D-Wandler gesetzt. Dadurch ergibt sich bei Regelkreisen eine definierte Verzögerungszeit zwischen Eingangsgröße (A/D-Input) und Stellgröße (D/A-Output). Das Setzen der neuen D/A-Wandler-Werte kann asynchron hierzu erfolgen.

4.9.8 Auswahl der A/D-Datenverarbeitung (`vadsrv`)

`cmmd` [HEX]	`para`	Status lesbar auf VMEbus Adresse [HEX]	Status lesbar auf lokaler Adresse [HEX]	Name der Statuszelle	Parameter-Werte [HEX]	
					Wertebereich: $0000...$000B	Default-Wert (Statuszelle): $01
7	WORD	14C	A6	`vadsrv`	$0000 Transfer der A/D-Werte in Shared-RAM (nur für Test-und Diagnose) $0001 Transfer der A/D-Werte in Shared-RAM über DMA-Transfer $0002 wie 01, jedoch zusätzlich Berechnung der korrigierten A/D-Werte und Ablage im Shared-RAM $0003 wie 01, jedoch zusätzlich Summation von A/D-Werten und Offset-Korrektur $000A Buffer-Mode: ADC-One-Shot (siehe Seite 67) $000B Buffer-Mode: Continuoues (siehe Seite 67)	

Im Mode ‘Summation von A/D-Werten’ werden die A/D-Daten addiert, der Offset korrigiert und die errechneten Werte in einem neuen Buffer abgelegt. Die Anzahl der zu addierenden Daten kann über das Kommando ‘Anzahl zu addierender A/D-Werte’ vorgegeben werden. Wie in Mode 02 lassen sich auch in Mode 03 zusätzlich zu den korrigierten Werten auch die nicht korrigierten A/D-Werte lesen.

Die Modes ‘ADC-One-Shot’ und ‘ADC-Buffer-Mode’ werden in einem eigenen Kapitel ausführlich beschrieben, da sie sich grundlegend von den anderen Betriebsarten unterscheiden (siehe Seite 67).
Zu beachten ist, daß beim Speichern der Parameter im I²C-EEPROM die Buffer-Modes nicht mit abgespeichert werden. Es werden nur Modes gespeichert, bei denen das hochwertigste Parameter-Bit auf ‘0’ gesetzt ist.

Für die aufgeführten Modes benötigt die AIO16 unterschiedlich lange Verarbeitungszeiten. Die Transfergeschwindigkeit hängt außerdem noch von der CPU-Taktfrequenz und der Anzahl der gewandelten Kanäle ab.
Die folgende Tabelle zeigt Beispielhaft die benötigte Zeit vom Starten der Wandlung bis zum Setzen des lokalen ‘Data Ready’-Flags. Dieses Flag signalisiert das Eintreffen der Daten im Shared-RAM und triggert (sofern Trigger-Bedingung aktiviert) den VMEbus-Interrupt.

Die angegebenen Werte gelten nur, wenn der lokale SRAM-Speicher als ‘Fast-SRAM’ ausgeführt ist (Zugriffszeit FAST-SRAM ca. 15 ns). Bei Hardware-Versionen größer ‘0’ (siehe Seite 15) ist immer FAST-RAM bestückt.

CPU-Taktfrequenz	para	Verarbeitungsdauer in Abhängigkeit von der Anzahl der A/D-Kanäle, deren Daten in das RAM transferiert werden sollen in [µs]		
		16 Kanäle	8 Kanäle	1 Kanal
25 MHz	$0000	<23	15	10
	$0001	<18	12	8
	$0002	<75	43	18
	$0003 [1*)]	<50	26	12

[1*)] Angegebene Zeiten gelten für Statuszelle `vsmcnt` = 4.

Verarbeitungsgeschwindigkeit in Abhängigkeit von der Anzahl der A/D-Kanäle und der D/A-Kanäle im Buffer-Mode:

Anzahl der A/D-Wandler-Kanäle	Anzahl der D/A-Wandler-Kanäle	Anzahl der Frames [2*)] pro Sekunde
1	1	118 000
2	1	110 000
2	2	103 000
6	4	74 300
16	4	50 700

[2*)] Ein Frame beinhaltet die Daten aller vorher definierten A/D-, bzw. D/A-Kanäle.

Für weitere Kombinationen der Anzahl A/D- und D/A-Kanäle kann die Anzahl der Frames pro Sekunde nach folgender Formel errechnet werden:

```
Frames/Zeit = 1 / (Anzahl_ADC · 0,625µs + Anzahl_DAC · 0,675µs + 6,2µs + 1µs)
```

4.9.9 A/D-Kanäle selektieren (`vstart, vend`)

Die Kommandos ‘Ersten A/D-Kanal zum Auslesen selektieren’ und ‘Letzten A/D-Kanal zum Auslesen selektieren’ werden eingesetzt, um die lokale CPU zu entlasten, wenn nicht alle 16 A/D-Kanäle benötigt werden. Dadurch läßt sich die Abtastrate der verbliebenen A/D-Kanäle erhöhen.

`cmmd` [HEX]	para	Status lesbar auf VMEbus Adresse [HEX]	Status lesbar auf lokaler Adresse [HEX]	Name der Statuszelle	Parameter-Werte [HEX]	
8	WORD	14D	A7	`vstart`	Wertebereich der Parameter: siehe unten	Default-Werte (Statuszellen): `vstart` = $01, `vend` = $10
9	WORD	150	A8	`vend`		

Wertebereich der Parameter:

Den 16 A/D-Wandler-Kanälen sind die positiven Zahlen von 1 bis 16 zugeordnet.

Den 8 Auxilary-Inputs sind die negativen Zahlen von ‘-1’ bis ‘-8’ zugeordnet.

Die folgenden Beispiele verdeutlichen die Funktion der Parameter:

Wert der Parameter		Zum Auslesen selektierte Kanäle	
`vstart`	`vend`	Auxilary-Inputs	A/D-Wandler-Kanäle
1	16	-	ADC_1, ADC_2, ... ADC_16
3	3	-	ADC_3
-1	16	Aux_In_1	ADC_1, ADC_2, ... ADC_16
-6	8	Aux_In_6, Aux_In_5, ... Aux_In_1	ADC_1, ADC_2, ... ADC_8
-8	16	Aux_In_8, Aux_In_7, ... Aux_In_1	ADC_1, ADC_2, ... ADC_16

4.9.10 Quelle für VMEbus-Interrupt wählen (`vvtrg`)

Über dieses Kommando kann ausgewählt werden, bei welcher Bedingung ein VMEbus-Interrupt von der AIO16 ausgelöst werden kann. Drei Möglichkeiten stehen zur Auswahl:

1. **kein Interrupt**

2. **IRQ_on_Acknowledge**
   Dieser Interrupt wird ausgelöst, nachdem die A/D-Daten im SRAM-Speicher zur Verfügung stehen und die Status-Worte `adstat0` oder `adstat1` gesetzt sind. Die Status-Worte können also nach dem Interrupt ausgewertet werden.

3. **IRQ_on_DMA_done**
   Dieser Interrupt wird ausgelöst, nachdem die A/D-Daten im SRAM-Speicher zur Verfügung stehen. Die Status-Worte werden jedoch zu einem späteren Zeitpunkt gesetzt (siehe folgende Abbildung 4.8.1). Werden sie sofort nach dem IRQ abgefragt, kann es sein, daß sie noch nicht aktualisiert worden sind.

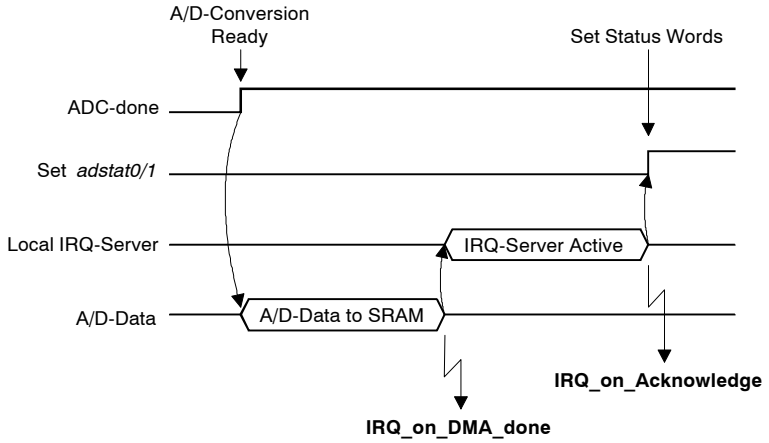

Abb. 4.8.1: Zeitpunkt der Interrupt-Erzeugung

`cmmd` [HEX]	`para`	Status lesbar auf VMEbus Adresse [HEX]	Status lesbar auf lokaler Adresse [HEX]	Name der Statuszelle	Parameter-Werte [HEX]	
A	WORD	151	A9	`vvtrg`	Wertebereich: $0000, $007F, $00FF	Default-Wert (Statuszelle): $00
					$0000	kein VMEbus-Interrupt
					$007F	IRQ_on_Acknowledge
					$00FF	IRQ_on_DMA_done

4.9.11 A/D-Wandler-Auflösung setzen (`vadres`)

In dieser Zelle wird von der lokalen Firmware eingetragen, mit welchem A/D-Wandler-Typ die AIO16 bestückt ist. Die Zelle sollte vom Anwender nicht verändert werden.

`cmmd` [HEX]	para	Status lesbar auf VMEbus Adresse [HEX]	Status lesbar auf lokaler Adresse [HEX]	Name der Statuszelle	Parameter-Werte [HEX]	
B	WORD	154	AA	`vadres`	Wertebereich: $0000...$00FF	Default-Wert (Statuszelle): $00
					$0000	16-Bit A/D-Wandler bestückt
					≠ $0000	12-Bit A/D-Wandler bestückt

4.9.12 Auswahl der D/A-Datenverarbeitung (`vdasrv`)

In dieser Zelle können verschiedene Modes für das Setzen der D/A-Wandler-Kanäle eingetragen werden.

`cmmd` [HEX]	para	Status lesbar auf VMEbus Adresse [HEX]	Status lesbar auf lokaler Adresse [HEX]	Name der Statuszelle	Parameter-Werte [HEX]	
					Wertebereich: $0000...$0002	Default-Wert (Statuszelle): $00
C	WORD	155	AB	`vdasrv`	$0000 normale Betriebsart, d.h Setzen der Kanäle über einzelne VMEbus-Schreibzugriffe (siehe Seite 66) $0001 Buffer-Mode: DAC-One-Shot (siehe Seite 67) $0002 Buffer-Mode: DAC-Wrap-Mode (Continuous) (siehe Seite 67) $0003 Buffer-Mode: Selective DAC-Buffer	

Beim Speichern der Parameter im I²C-EEPROM wird für `vadsrv` immer ‘0’ abgelegt, d.h. die Einstellung der Buffer-Modes kann nicht permanent gespeichert werden!

4.9.13 Anzahl zu addierender A/D-Werte (`vsmcnt`)

Dieses Kommando ist nur von Bedeutung, wenn über das Kommando '`vadsrv`' der Mode 'Summation der A/D-Werte' gewählt ist.

`cmmd` [HEX]	`para`	Status lesbar auf VMEbus Adresse [HEX]	Status lesbar auf lokaler Adresse [HEX]	Name der Statuszelle	Parameter-Werte [HEX]	
D	WORD	158	AC	`vsmcnt`	Wertebereich: $0004...$7FFF	Default-Wert (WORD-Statuszelle): $0000
					Anzahl zu summierender A/D-Werte: (sinnvoll sind Werte ab '4')	

Die lokale Software errechnet sich aus dem übergebenen Parameterwert die nächst mögliche Anzahl der zu addierenden Werte. Beim Lesen des Parameterwertes wird der errechnete, real eingestellte Wert zurückgegeben.

4.9.14 A/D-Buffer einrichten (`adbuf`)

Mit diesem Kommando wird im Shared-RAM ein Speicherbereich für die A/D-Daten, die im Buffer-Mode aufgezeichnet werden sollen, reserviert.

`cmmd` [HEX]	`para`	VMEbus-Startadresse der Struktur [HEX]	lokale Startadresse der Struktur [HEX]	Name der Struktur	Parameter-Werte [HEX]	
E	Upper WORD	100	80	`adbuf`	Anzahl der Frames pro Buffer	Wertebereich: $0000...$7FFF
	Lower WORD				Anzahl der Buffer	Wertebereich: $0000...$7FFF

Die Funktion des Buffer-Modes und die Zusammensetzung der Buffer und Frames wird in einem eigenen Kapitel ab Seite 67 ausführlich beschrieben.
Die übergebenen Parameter dieses Kommandos können nicht im I²C-EEPROM gespeichert werden.

4.9.15 D/A-Buffer einrichten (`dabuf`)

Mit diesem Kommando wird im Shared-RAM ein Speicherbereich für die D/A-Daten, die im Buffer-Mode gesetzt werden sollen, reserviert.

`cmmd` [HEX]	`para`	VMEbus-Startadresse der Struktur [HEX]	lokale Startadresse der Struktur [HEX]	Name der Struktur	Parameter-Werte [HEX]	
F	Upper WORD	120	90	`dabuf`	Anzahl der Frames pro Buffer	Wertebereich: $0000...$FFFF
	Lower WORD				Anzahl der Buffer	Wertebereich: $0000...$FFFF

Die Funktion des Buffer-Modes und die Zusammensetzung der Buffer und Frames wird in einem eigenen Kapitel ab Seite 67 ausführlich beschrieben.
Die übergebenen Parameter dieses Kommandos können nicht im I²C-EEPROM gespeichert werden.

4.9.16 D/A-Kanäle selektieren (`dastart, daend`)

Die Kommandos ‘Ersten D/A-Kanal zum Auslesen selektieren’ und ‘Letzten D/A-Kanal zum Auslesen selektieren’ werden eingesetzt, um die lokale CPU zu entlasten, wenn nicht alle 4 D/A-Kanäle benötigt werden. Dadurch läßt sich die Wandlungsrate der verbliebenen D/A-Kanäle und die Abtastrate der A/D-Kanäle erhöhen.

`cmmd` [HEX]	para	Status lesbar auf VMEbus Adresse [HEX]	Status lesbar auf lokaler Adresse [HEX]	Name der Statuszelle	Parameter-Werte [HEX]	
					Wertebereich der Parameter: $0001...$0004 $0001 ≤ erster Kanal ≤ letzter Kanal ≤ $0004	Default-Werte (Statuszellen): `dastart` = $01, `daend` = $04
10	WORD	15C	AE	`dastart`	Beispiele: `dastart` = $02, `daend` = $04 -> Die D/A-Kanäle 2 bis 4 werden ausgewertet. `dastart` = $03, `daend` = $03 -> Nur D/A-Kanal 3 wird ausgewertet.	
11	WORD	15E	AF	`daend`		

4.9.17 Selective-D/A-Buffer

cmmd [HEX]	para	Status lesbar auf VMEbus Adresse [HEX]	Status lesbar auf lokaler Adresse [HEX]	Name der Statuszelle	Name des Parameters	Parameter-Werte [HEX]
12	Upper WORD	40	20	cstat	buffer_no	Wertebereich: 0000...FFFF
	Lower WORD	40	20	cstat	dac_mask	Wertebereich: 0000...000F

Mit diesem Kommando werden der D/A-Buffer und der (oder die) D/A-Wandler-Kanäle ausgewählt. Vor dem Aufruf dieses Kommandos müssen die D/A-Buffer über das Kommanod $F (D/A-Buffer einrichten) eingerichtet werden.

`buffer_no` = 1, 2, ...N (N = Number_of_Buffers)

Number _of_Buffers = maximale Anzahl der D/A-Buffer nach Aufruf von Kommando $F

Die Buffer -Start-Adresse der D/A-Buffer kann in der Zelle `vdacsb` ($B0) gelesen werden (siehe D/A-Buffer-Status-Struktur Seite 70).

`dac_mask`: Dieser Parameter bestimmt die an dem Buffer beteiligten D/A-Wandler-Kanäle.

`dac_mask` : bit 0 - DAC 1
bit 1 - DAC 2
bit 2 - DAC 3
bit 3 - DAC 4

Der Wert von `dac_mask` kann in der Zelle `vdamsk` ($B4) zurückgelesen werden.

4.10 A/D-Wandler-Status (`adstat0...adstat3`)

In diesen Zellen kann der aktuelle Status der A/D-Wandlung gelesen werden. Die Zelle kann z.B. durch Polling abgefragt werden, um das Ende der Wandlung zu erfahren, wenn kein VMEbus-Interrupt nach abgeschlossener Wandlung erfolgen soll.

Nach dem Lesen der Daten sollte die jeweilige Zelle auf einen Wert ≠ FFFF gesetzt werden, da sie vom Server nicht selbständig zurückgesetzt wird. Im Gegensatz zu den anderen Parametern werden diese Zellen direkt gesetzt, d.h. nicht über den `cmmd` und `para`!

Die Zelle `adstat0` zeigt an, daß die unkorrigierten Daten im SRAM bereitliegen und die Zelle `adstat1` zeigt an, daß die korrigierten Werte im SRAM zur Verfügung stehen.

Die Zelle `adstat2` wird zur Zeit nicht benutzt. Sie ist reserviert für zukünftige Anwendungen.

In der Zelle `adstat3` wird angezeigt, ob die addierten A/D-Werte in der Betriebsart ‘Summation der A/D-Werte’ fertig berechnet sind.

relative Adresse VMEbus [HEX]	relative Adresse lokal [HEX]	Länge	Zugriff	Name der Statuszelle	Parameter-Werte [HEX]	

1FC
1F8
1F4
1F0
	FE
	FC
	FA
	F8
		WORD	read/ write	`adstat0`
				`adstat1`
				`adstat2`
				`adstat3`
					Wertebereich: $0000...$FFFF	

					$FFFF	neue A/D-Daten sind im RAM abgelegt
					≠ $FFFF	neue Daten sind noch nicht eingetroffen

Diese Seite ist bewußt unbedruckt.

5. Lesen der A/D-Meßwerte und Setzen der D/A-Wandler

5.1 Lesen der A/D-Meßwerte

Im folgenden werden die Zellen der analogen Meßwerte beschrieben. Diese Zellen werden von der lokalen Firmware gesetzt und dürfen nur gelesen werden!

Hinweis: Korrektur der A/D-Daten mit Anwenderparametern:
Die beim Selbsttest ermittelten Parameter `offs01...offs16`, `offs01.summed...offs16.summed`, sowie `scale01...scale16` können vom Anwender zwecks *eigener Korrektur* überschrieben werden!

5.1.1 Rohdaten der A/D-Wandler - `adwert01...adwert16`

Hier können die A/D-Werte gelesen werden, die von den A/D-Wandlern ausgegeben werden. Die Werte sind nicht korrigiert.

relative Adresse VMEbus [HEX]	relative Adresse lokal [HEX]	Länge	Zugriff	Zelleninhalt	Kurzbeschreibung
200	100	WORD	read only	`adwert01`	Rohdaten des A/D-Wandlers 1
204	102			`adwert02`	: 2
208	104			`adwert03`	: 3
20C	106			`adwert04`	: 4
210	108			`adwert05`	: 5
214	10A			`adwert06`	: 6
218	10C			`adwert07`	: 7
21C	10E			`adwert08`	: 8
220	110			`adwert09`	: 9
224	112			`adwert10`	: 10
228	114			`adwert11`	: 11
22C	116			`adwert12`	: 12
230	118			`adwert13`	: 13
234	11A			`adwert14`	: 14
238	11C			`adwert15`	: 15
23C	11E			`adwert16`	Rohdaten des A/D-Wandlers 16

Zellen-Inhalt	Parameter-Werte [HEX]
`adwert01...adwert16`	Wertebereich: 8000...7FFF
	8000 -10 V : : 0000 0 V : 7FFF + 10 V -1 LSB

5.1.2 Korrigierte A/D-Wandler-Daten - `advac01...advac16`

Hier können nach der Wandlung die um den Gain- und den Offset-Fehler korrigierten A/D-Werte gelesen werden, wenn der Wandlungsmode ‘2’ (siehe Kommando `vadsrv`) gewählt worden ist.

relative Adresse VMEbus [HEX]	relative Adresse lokal [HEX]	Länge	Zugriff	Zelleninhalt	Kurzbeschreibung
240	120	WORD	read only	`advac01`	Korrigierte A/D-Daten Kanal 1
244	122			`advac02`	: 2
248	124			`advac03`	: 3
24C	126			`advac04`	: 4
250	128			`advac05`	: 5
250	12A			`advac06`	: 6
258	12C			`advac07`	: 7
25C	12E			`advac08`	: 8
260	130			`advac09`	: 9
264	132			`advac10`	: 10
268	134			`advac11`	: 11
26C	136			`advac12`	: 12
270	138			`advac13`	: 13
274	13A			`advac14`	: 14
278	13C			`advac15`	: 15
27C	13E			`advac16`	Korrigierte A/D-Daten Kanal 16

Zellen-Inhalt	Parameter-Werte [HEX]
`advac01...advac16`	Wertebereich: 8000...7FFF
	8000 -10 V : : 0000 0 V : : 7FFF + 10 V -1 LSB

Die korrigierten Werte werden wie folgt berechnet:

$$advac = (adwert - offs)\left(1 + \frac{scale}{2^{16}}\right)$$

mit

`adwert`	= Rohdaten der A/D-Wandler-Kanäle
`offs`	= Offset-Wert der A/D-Wandler-Kanäle
`scale`	= Gain-Korrektur-Faktor * 2^{16}
`advac`	= korrigierte A/D-Werte

Die Korrekturfaktoren sind ab Seite 54 beschrieben.

5.1.3 Summierte A/D-Wandler-Daten - `adc01.summed...adc16.summed`

Hier können nach der Wandlung und Summation die errechneten A/D-Werte gelesen werden, wenn der Wandlungsmode '3' (siehe Kommando `vadsrv`) gewählt worden ist.

relative Adresse VMEbus [HEX]	relative Adresse lokal [HEX]	Länge	Zugriff	Zelleninhalt	Kurzbeschreibung
280	140	LONG	read only	`adc01.summed`	Summierte A/D-Daten Kanal 1
288	144			`adc02.summed`	: 2
290	148			`adc03.summed`	: 3
298	14C			`adc04.summed`	: 4
2A0	150			`adc05.summed`	: 5
2A8	154			`adc06.summed`	: 6
2B0	158			`adc07.summed`	: 7
2B8	15C			`adc08.summed`	: 8
2C0	160			`adc09.summed`	: 9
2C8	164			`adc10.summed`	: 10
2D0	168			`adc11.summed`	: 11
2D8	16C			`adc12.summed`	: 12
2E0	170			`adc13.summed`	: 13
2E8	174			`adc14.summed`	: 14
2F0	178			`adc15.summed`	: 15
2F8	17C			`adc16.summed`	Summierte A/D-Daten Kanal 16

Zellen-Inhalt	Parameter-Werte [HEX]
`adc01.summed ... adc16.summed`	Wertebereich: 80000000...7FFFFFFF
	80000000 -10 V : : 00000000 0 V : : 7FFFFFFF + 10 V -1 LSB(32)

Nach der Summation und Re-Formatierung der A/D-Werte erfolgt zudem noch eine Offset-Korrektur mit den Daten `offs01.smmd`...`offs16.smmd`. Der so errechnete A/D-Wandler-Wert wird immer linksbündig im Langwort dargestellt. Soll nur eine Auswertung von 16 Bits erfolgen, so sind die beiden höherwertigen Bytes zu lesen.

5.1.4 Offset der A/D-Wandler - `offs01...offs16`

Diese Meßwerte sind beim Selbsttest aufgenommen worden. Es sind die Meßwerte, die sich ergeben, wenn die A/D-Wandler-Eingänge über Multiplexer an GND gelegt werden und geben somit den Offset-Fehler der Eingangsschaltung wieder.

relative Adresse VMEbus [HEX]	relative Adresse lokal [HEX]	Länge	Zugriff	Zelleninhalt	Kurzbeschreibung	
400	200	WORD	read/ (write)	`offs01`	Offset des A/D-Wandlers	1
404	202			`offs02`	:	2
408	204			`offs03`	:	3
40C	206			`offs04`	:	4
410	208			`offs05`	:	5
414	20A			`offs06`	:	6
418	20C			`offs07`	:	7
41C	20E			`offs08`	:	8
420	210			`offs09`	:	9
424	212			`offs10`	:	10
428	214			`offs11`	:	11
42C	216			`offs12`	:	12
430	218			`offs13`	:	13
434	21A			`offs14`	:	14
438	21C			`offs15`	:	15
43C	21E			`offs16`	Offset des A/D-Wandlers	16

Zellen-Inhalt	Parameter-Werte [HEX]
`offs01...offs16`	Wertebereich: 8000...7FFF
	8000 -10 V : : 0000 0 V : : 7FFF + 10 V -1 LSB

5.1.5 A/D-Werte bei anliegender Referenzspannung - `ref501...ref516`

Diese Meßwerte sind beim Selbsttest aufgenommen worden. Es sind die Meßwerte, die sich ergeben, wenn die A/D-Wandler-Eingänge über Multiplexer mit der +5 V-Referenzspannung verbunden sind. Der Offset-Fehler ist in diesen Werten bereits berücksichtigt worden.

relative Adresse VMEbus [HEX]	relative Adresse lokal [HEX]	Länge	Zugriff	Zelleninhalt	Kurzbeschreibung	
440	220	WORD	read only	`ref501`	A/D-Wert bei +5V-Ref Kanal	1
444	222			`ref502`	:	2
448	224			`ref503`	:	3
44C	226			`ref504`	:	4
450	228			`ref505`	:	5
454	22A			`ref506`	:	6
458	22C			`ref507`	:	7
45C	22E			`ref508`	:	8
460	230			`ref509`	:	9
464	232			`ref510`	:	10
468	234			`ref511`	:	11
46C	236			`ref512`	:	12
470	238			`ref513`	:	13
474	23A			`ref514`	:	14
478	23C			`ref515`	:	15
47C	23E			`ref516`	A/D-Wert bei +5V-Ref Kanal	16

Zellen-Inhalt	Parameter-Werte [HEX]
`ref501...ref516`	Wertebereich: 8000...7FFF
	8000 -10 V : : 0000 0 V : : 7FFF + 10 V -1 LSB

Die Werte errechnen sich wie folgt:

$$\texttt{ref5} = (\texttt{adwert}_{\texttt{bei 5V}} - \texttt{offs})$$

mit

$\texttt{adwert}_{\texttt{bei 5V}}$ = Rohdaten der A/D-Wandler-Kanäle bei anliegender 5V-Referenzspannung
`offs` = Offset-Wert der A/D-Wandler-Kanäle
`ref5` = um Offset korrigierter A/D-Wert bei +5 V

5.1.6 A/D-Werte bei anliegender D/A-Spannung von 0 V: D/A-Offset (`adc001...adc016`)

Diese Meßwerte sind beim Selbsttest aufgenommen worden. Es sind die Meßwerte, die sich ergeben, wenn die A/D-Wandler-Eingänge über Multiplexer mit den D/A-Wandler-Ausgängen verbunden sind. Die D/A-Wandler sind in diesem Fall auf 0 V eingestellt.

Aus den Meßwerten läßt sich der Offset-Fehler der D/A-Wandler ermitteln.

Jeder D/A-Wandler wird durch die Mehrzahl der A/D-Wandler viermal gemessen. Diese vier Meßwerte sollten in etwa gleich sein.

relative Adresse VMEbus [HEX]	relative Adresse lokal [HEX]	Länge	Zugriff	Zelleninhalt	Kurzbeschreibung			
480	240	WORD	read only	`adc001`	A/D-Wert bei	DAC1=0 V	Kanal	1
484	242			`adc002`	:	DAC2	:	2
488	244			`adc003`	:	DAC3	:	3
48C	246			`adc004`	:	DAC4	:	4
490	248			`adc005`	:	DAC1	:	5
494	24A			`adc006`	:	DAC2	:	6
498	24C			`adc007`	:	DAC3	:	7
49C	24E			`adc008`	:	DAC4	:	8
4A0	250			`adc009`	:	DAC1	:	9
4A4	252			`adc010`	:	DAC2	:	10
4A8	254			`adc011`	:	DAC3	:	11
4AC	256			`adc012`	:	DAC4	:	12
4B0	258			`adc013`	:	DAC1	:	13
4B4	25A			`adc014`	:	DAC2	:	14
4B8	25C			`adc015`	:	DAC3	:	15
4BC	25E			`adc016`	A/D-Wert bei	DAC4=0 V	Kanal	16

Zellen-Inhalt	Parameter-Werte [HEX]
`adc001...adc016`	Wertebereich: 8000...7FFF
	8000 -10 V : : 0000 0 V : : 7FFF + 10 V -1 LSB

5.1.7 A/D-Werte bei anliegendem D/A-Wert $7F00: D/A-Gain (`adc101...adc116`)

Diese Meßwerte sind beim Selbsttest aufgenommen worden. Es sind die Meßwerte, die sich ergeben, wenn die A/D-Wandler-Eingänge über Multiplexer mit den D/A-Wandler-Ausgängen verbunden sind. Die D/A-Wandler sind in diesem Fall auf mit dem Wert $7F00 geladen. Aus den Meßwerten läßt sich der Gain-Fehler der D/A-Wandler ermitteln. Jeder D/A-Wandler wird durch die Mehrzahl der A/D-Wandler viermal gemessen. Diese vier Meßwerte sollten in etwa gleich sein.

Der Offset-Fehler und die Gain-Korrektur der A/D-Kanäle sind in diesen Werten bereits berücksichtigt. Außerdem sind die Werte um den Offset-Fehler der D/A-Kanäle korrigiert.

relative Adresse VMEbus [HEX]	relative Adresse lokal [HEX]	Länge	Zugriff	Zelleninhalt	Kurzbeschreibung
4C0	260	WORD	read only	`adc101`	A/D-Wert bei DAC1=$7F00 Kan. 1
4C4	262			`adc102`	: DAC2 : 2
4C8	264			`adc103`	: DAC3 : 3
4CC	266			`adc104`	: DAC4 : 4
4D0	268			`adc105`	: DAC1 : 5
4D4	26A			`adc106`	: DAC2 : 6
4D8	26C			`adc107`	: DAC3 : 7
4DC	26E			`adc108`	: DAC4 : 8
4E0	270			`adc109`	: DAC1 : 9
4E4	272			`adc110`	: DAC2 : 10
4E8	274			`adc111`	: DAC3 : 11
4EC	256			`adc112`	: DAC4 : 12
4F0	278			`adc113`	: DAC1 : 13
4F4	27A			`adc114`	: DAC2 : 14
4F8	27C			`adc115`	: DAC3 : 15
4FC	27E			`adc116`	A/D-Wert bei DAC4=$7F00 Kan. 16

Zellen-Inhalt	Parameter-Werte [HEX]
`adc101...adc116`	Wertebereich: 8000...7FFF
	8000 -10 V : : 0000 0 V : : 7FFF + 10 V -1 LSB

Die Werte berechnen sich wie folgt:

$$\texttt{adc1} = (\texttt{adwert}_{\text{bei DAC=7F00}} - \texttt{adc0})$$

mit

$\texttt{adwert}_{\text{bei DAC=7F00}}$	=	gemessener A/D-Wert, wenn D/A-Wandler auf $7F00 gesetzt sind
`adc0`	=	Offset-Wert der D/A-Wandler-Kanäle
`adc1`	=	um Offset korrigierter D/A-Wert bei $7F00 zur Ermittlung der GAIN-Korrektur

5.1.8 A/D-Wandler: Gain-Korrektur-Faktoren (`scale01...scale16`)

Der Gain-Korrektur-Faktor wird zu Berechnung der korrigierten A/D-Werte herangezogen. Der Faktor wird nach der Messung der A/D-Werte mit der +5 V-Referenzspannung ermittelt.

relative Adresse VMEbus [HEX]	relative Adresse lokal [HEX]	Länge	Zugriff	Zelleninhalt	Kurzbeschreibung	
500	280	WORD	read/ (write)	`scale01`	Gain-Faktor des A/D-Wandlers	1
504	282			`scale02`	:	2
508	284			`scale03`	:	3
50C	286			`scale04`	:	4
510	288			`scale05`	:	5
514	28A			`scale06`	:	6
518	28C			`scale07`	:	7
51C	28E			`scale08`	:	8
520	290			`scale09`	:	9
524	292			`scale10`	:	10
528	294			`scale11`	:	11
52C	296			`scale12`	:	12
530	298			`scale13`	:	13
534	29A			`scale14`	:	14
538	29C			`scale15`	:	15
53C	29E			`scale16`	Gain-Faktor des A/D-Wandlers	16

Zellen-Inhalt	Parameter-Werte [HEX]
`scale01...scale16`	Wertebereich: 8000...0000...7FFF

$$scale = \frac{adsollwert - adwert}{adsollwert} \cdot 2^{16}$$

$$adwert_z = adwert_{in} - offs$$

$$adwert_{korr} = adwert_z + \frac{adwert_z * scale}{2^{16}}$$

mit

`adwert` = gemessener A/D-Wert bei anliegender +5 V-Referenzspannung
`adsollwert` = idealer Sollwert bei +5V
`scale` = Gain-Korrektur-Faktor

5.1.9 Offset-Werte der addierten 32-Bit-A/D-Werte (`offs01.summed...offs16.summed`)

Die hier lesbaren Werte entsprechen den in der Selbsttestphase bei 0 V gemessenen und summierten Werten. Sie können über VMEbus-Zugriffe verändert werden.

relative Adresse VMEbus [HEX]	relative Adresse lokal [HEX]	Länge	Zugriff	Zelleninhalt	Kurzbeschreibung	
540	2A0	LONG	read/ write	`offs01.summed`	32-Bit-Offset A/D-Kanal	1
548	2A4			`offs02.summed`	:	2
550	2A8			`offs03.summed`	:	3
558	2AC			`offs04.summed`	:	4
560	2B0			`offs05.summed`	:	5
568	2B4			`offs06.summed`	:	6
570	2B8			`offs07.summed`	:	7
578	2BC			`offs08.summed`	:	8
580	2C0			`offs09.summed`	:	9
588	2C4			`offs10.summed`	:	10
590	2C8			`offs11.summed`	:	11
598	2CC			`offs12.summed`	:	12
5A0	2D0			`offs13.summed`	:	13
5A8	2D4			`offs14.summed`	:	14
5B0	2D8			`offs15.summed`	:	15
5B8	2DC			`offs16.summed`	32-Bit-Offset A/D-Kanal	16

Zellen-Inhalt	Parameter-Werte [HEX]
`offs01.summed ... offs16.summed`	Wertebereich: 80000000...7FFFFFFF
	80000000 -10 V : : 00000000 0 V : : 7FFFFFFF + 10 V -1 LSB

5.2 Auxilary-Inputs

Die Rohdaten der Auxilary_Inputs sind lokale Speicherzellen der VME-AIO16, die vom VMEbus aus gesetzt werden können. Die Daten dieser Speicherzellen können wie die A/D-Wandler-Daten von der lokalen Firmware weiterverarbeitet werden.

5.2.1 Rohdaten der Auxilary-Inputs `auxin1...auxin8`

Hier können die A/D-Werte gelesen werden, die über die VMEbus-Schnittstelle vorgegeben worden sind. Die Werte haben keine Korrektur erfahren.

relative Adresse VMEbus [HEX]	relative Adresse lokal [HEX]	Länge	Zugriff	Zelleninhalt	Kurzbeschreibung
300	180	WORD	read only	`auxin8`	Rohdaten des Auxilary-Inputs 8
304	182			`auxin7`	: 7
308	184			`auxin6`	: 6
30C	186			`auxin5`	: 5
310	188			`auxin4`	: 4
314	18A			`auxin3`	: 3
318	18C			`auxin2`	: 2
31C	18E			`auxin1`	Rohdaten des Auxilary-Inputs 1

Zellen-Inhalt	Parameter-Werte [HEX]
`auxin1...auxin8`	Wertebereich: 8000...7FFF
	8000 : 0000 : 7FFF

5.2.2 Korrigierte Daten der Auxilary-Inputs - `auxcr1...auxcr8`

relative Adresse VMEbus [HEX]	relative Adresse lokal [HEX]	Länge	Zugriff	Zelleninhalt	Kurzbeschreibung
320	190	WORD	read only	`auxcr8`	Korrigierte Daten des Aux-Inp. 8
324	192			`auxcr7`	: 7
328	194			`auxcr6`	: 6
32C	196			`auxcr5`	: 5
330	198			`auxcr4`	: 4
334	19A			`auxcr3`	: 3
338	19C			`auxcr2`	: 2
33C	19E			`auxcr1`	Korrigierte Daten des Aux-Inp. 1

Zellen-Inhalt	Parameter-Werte [HEX]
`auxcr1...auxcr8`	Wertebereich: 8000...7FFF
	8000 : 0000 : 7FFF

Die korrigierten Werte werden wie folgt berechnet:

$$\text{auxcr} = (\text{auxwert} - \text{offs})\left(1 + \frac{\text{scale}}{2^{16}}\right)$$

mit

`auxwert` = Rohdaten der Auxilary-Inputs
`offs` = Offset-Wert der Auxilary-Inputs
`scale` = Gain-Korrektur-Faktor * 2^{16}
`auxcr` = korrigierte Auxilary-Input-Werte

Die Korrekturfaktoren sind ab Seite 64 beschrieben.

5.2.3 Offset der Auxilary-Inputs - `ofsax1...ofsax8`

relative Adresse VMEbus [HEX]	relative Adresse lokal [HEX]	Länge	Zugriff	Zelleninhalt	Kurzbeschreibung	
5C0	2E0	WORD	read only	`ofsax8`	Offset des Auxilarx-Inputs	8
5C4	2E2			`ofsax7`	:	7
5C8	2E4			`ofsax6`	:	6
5CC	2E6			`ofsax5`	:	5
5D0	2E8			`ofsax4`	:	4
5D4	2EA			`ofsax3`	:	3
5D8	2EC			`ofsax2`	:	2
5DC	2EE			`ofsax1`	Offset des Auxilarx-Inputs	1

Zellen-Inhalt	Parameter-Werte [HEX]
`ofsax1...ofsax8`	Wertebereich: 8000...7FFF
	8000 : 0000 : 7FFF

5.2.4 Gain-Korrektur-Faktoren der Auxilary-Inputs (`sclax1...sclax8`)

relative Adresse VMEbus [HEX]	relative Adresse lokal [HEX]	Länge	Zugriff	Zelleninhalt	Kurzbeschreibung	
5E0	280	WORD	read only	`sclax8`	Gain-Faktor des Auxilary-Inputs	8
5E4	282			`sclax7`	:	7
5E8	284			`sclax6`	:	6
5EC	286			`sclax5`	:	5
5F0	288			`sclax4`	:	4
5F4	28A			`sclax3`	:	3
5F8	28C			`sclax2`	:	2
5FC	28E			`sclax1`	Gain-Faktor des Auxilary-Inputs	1

Zellen-Inhalt	Parameter-Werte [HEX]
`sclax1...sclax8`	Wertebereich: 8000...FFFF

$$auxcr = (auxwert - ofsax)\left(1 + \frac{sclax}{2^{16}}\right)$$

5.3 Setzen der D/A-Wandler

Die D/A-Wandler werden über VMEbus-Schreibzugriffe auf die D/A-Wandler-Adressen gesetzt. Die D/A-Wandler-Adressen sind auf Seite 8 aufgeführt.

5.3.1 Hinweise zum Setzen der D/A-Wandler

Bei Schreibzugriffen auf die D/A-Wandler werden gleichzeitig Zellen des Shared-RAMs mit den D/A-Werten besetzt. Ein Lesezugriff führt auf das SRAM und der vormals gesetzte Wert kann zurückgelesen werden.

5.3.2 Datenformat der D/A-Wandler-Werte

Das Datenformat für die D/A-Wandler-Werte ist wie folgt:

D/A-Wandler-Daten [HEX]	Ausgangsspannungswert [V]
8000	-10 V
:	:
0000	0 V
:	:
7FFF	+ 10 V -1 LSB

6. Buffer-Mode

6.1 Übersicht

Der Buffer-Mode bietet im Vergleich zu den bisher beschriebenen Betriebsarten die Möglichkeit, die A/D-Wandler-Meßwerte sequentiell abzulegen. Die Meßwerte werden also in der zeitlichen Reihenfolge, in der sie ermittelt wurden in einem Bereich des Shared-RAM, dem sogenannten 'Buffer' abgelegt. Die sequentielle Ablage führt außerdem zu einer höheren Wandlungsrate. Vergleichswerte zu den Wandlungsraten finden Sie auf Seite 38.
Während der Aufnahme der analogen Eingangsdaten im Buffer-Mode werden die statischen Werte, wie 'Rohdaten' und 'Korrigierte Daten' versorgt.

6.2 Buffer und Frames

Die A/D-Daten und die D/A-Daten werden im Buffer-Mode organisiert in 'Buffern' und 'Frames' abgelegt. Ein Frame enthält die A/D- oder D/A-Werte aller zur Bearbeitung freigegebenen Wandler. Ein Frame kann für die analogen Eingänge also (-8)...16 A/D-Werte enthalten. Die Anzahl der zu wandelnden Kanäle wird wie bei den anderen Betriebsarten über die Parameter `vstart` und `vend` definiert. Ein Frame mit D/A-Wandler-Daten kann ein bis vier D/A-Werte enthalten. Die Auswahl wird über die Parameter `dastart` und `daend` getroffen.

Beispiel für den Aufbau eines A/D-Frames:

A/D-Value(`vstart`)	A/D-Value(`vstart`+1)	A/D-Value(`vstart`+2)	...	A/D-Value(`vend`)

Beispiel für den Aufbau eines D/A-Frames:

D/A-Value(`dastart`)	D/A-Value(`dastart`+1)	D/A-Value(`dastart`+2)	...	D/A-Value(`daend`)

Mehrere Frames werden zu einem Buffer zusammengefaßt. Für die A/D-Wandler-Daten und für die D/A-Wandler-Daten kann die Anzahl der Frames pro Buffer definiert werden. Außerdem kann die Anzahl der gewünschten Buffer vorgegeben werden.

Die maximale Anzahl der Buffer wird durch die verfügbare Speicherkapazität eingeschränkt. Der lokalen Firmware steht ein Speicherbereich von $800 (VMEbus-Adresse) bis zu $7FDFE zur Verfügung. Wird jetzt z.B. zuerst der Speicherbereich für die A/D-Daten angefordert, so wird ein Bereich von $800 an aufsteigend reserviert. Im Falle der D/A-Daten wird ein Speicherbereich von $7FDFE abfallend reserviert. Die Grenzen des tatsächlich zur Verfügung stehenden Speicherbereiches schreibt die Firmware in die A/D-, bzw. D/A-Buffer-Status-Struktur.
Diese Art der Speicherplatzvergabe bietet den Vorteil, daß der verfügbare Platz je nach Bedarf mehr für A/D- oder D/A-Werte genutzt werden kann.

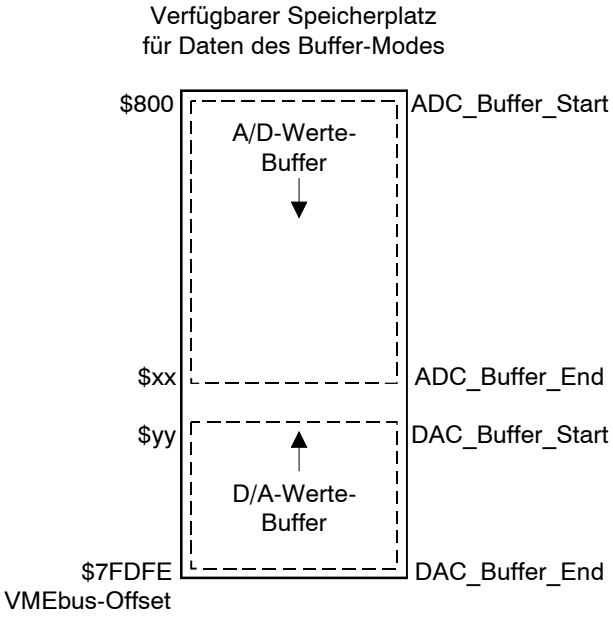

Abb. 6.2.1: Belegung des verfügbaren Speicherplatzes mit den Daten des A/D-Buffers und des D/A-Buffers

6.3 A/D- und D/A-Buffer-Status-Strukturen

Die A/D-Buffer-Status-Struktur und die D/A-Buffer-Status-Struktur sind Speicherbereiche, in denen Informationen zu den jeweiligen Buffern abgelegt werden. Hier kann der Anwender lesen, wo der bereitgestellte Speicherbereich der A/D- und D/A-Buffer tatsächlich beginnt und endet.

6.3.1 A/D-Buffer-Status-Struktur

Adreß-Offset VME (local) [HEX]	Länge	Zelleninhalt	Default-Werte [HEX]	Bemerkung
100 (80)	LONG	ADC_Buffer_Start_Offset	0000 0800	es wird der Offset innerhalb des Speichers für VMEbus-Zugriffe zurückgegeben
108 (84)	LONG	ADC_Buffer_End_Offset	0000 0800	
110 (88)	WORD	Number_of_ADCs_per_Frame	0	Wertebereich: 1...16 [1*)]
114 (8A)	WORD	Frames_per_Buffer	0	z.B. $400 für 1024
118 (8C)	WORD	Number_of_Buffers	0	1...N
11C (8E)	WORD	Buffer_Number_in_Work	0	1...N

[1*)] Number_of_ADCs_per_Frame = `vend` - `vstart` +1

6.3.2 D/A-Buffer-Status-Struktur

Adreß-Offset VME (local) [HEX]	Länge	Zelleninhalt	Default-Werte [HEX]	Bemerkung
120 (90)	LONG	D/A_Buffer_Start_Offset	0007 FE00	es wird der Offset innerhalb des Speichers für VMEbus-Zugriffe zurückgegeben
128 (94)	LONG	D/A_Buffer_End_Offset	0007 FE00	
130 (98)	WORD	Number_of_D/As_per_Frame	0	Wertebereich: 1...4 [2*)]
134 (9A)	WORD	Frames_per_Buffer	0	z.B. $400 für 1024
138 (9C)	WORD	Number_of_Buffers	0	1...N
13C (9E)	WORD	Buffer_Number_in_Work	0	1...N

[2*)] Number_of_D/As_per_Frame = `daend` - `dastart` +1

6.3.3 Selective D/A-Buffer-Struktur

Adreß-Offset VME (local) [HEX]	Länge	Zelleninhalt	Bemerkung
160 (B0)	LONG	Pointer_to_Buffer (`vdasb`)	D/A-Buffer-Base-Address
168 (B4)	WORD	Mask (`vdamsk`)	D/A-Enable-Word

6.4 Zeitlicher Ablauf

Der Buffer-Mode wird sofort nach dem jeweiligen Kommando ($7 oder $C) aktiv ! Voraussetzung ist jedoch,
- daß `vend`, `vstart`, `daend` und `dastart` gesetzt sind und
- daß der Speicherbereich für die Buffer vorher definiert worden ist (`adbuf`, `dabuf`),

6.4.1 Beispiel zum *synchronen* Starten

Zum synchronen Starten des A/D-Buffer-Modes und des D/A-Buffer-Modes sind die folgenden Schritte auszuführen:

Nr.	Kommando	`cmmd` [HEX]	`para` [DEZ]
1	Quelle für Trigger der A/D-Wandlung wählen	5	`trigmod` = 00 (Software-Trigger)
2	Ersten und letzten A/D-Kanal zur Wandlung auswählen	8, 9	`vstart` = ... `vend` = ...
3	Ersten und letzten D/A-Kanal zur Wandlung auswählen	10, 11	`dastart` = ... `daend` = ...
4	A/D-Buffer einrichten [1*)]	E	`adbuf` = ...
5	D/A-Buffer einrichten [1*)]	F	`dabuf` = ...
6	Auswahl der A/D-Datenverarbeitung (Buffer-Mode aktivieren)	7	`vadsrv` = 10 oder 11
7	Auswahl der D/A-Datenverarbeitung (Buffer-Mode aktivieren)	C	`vdasrv` = 1 oder 2
8	Quelle für LDAC der D/A-Wandlung wählen	6	`ldcmod` = 1
9	Periodendauer für Trigger der Wandlung durch Timer vorgeben	30	`cnvtime` = yx
10	Quelle für Trigger der A/D-Wandlung wählen	5	`trigmod` = 02 (Trigger by Timer)

[1*)] Einmalig nach RESET oder nach Änderung von `vstart/vend`, bzw. `dastart/daend`.

6.4.2 Timing-Diagram eines A/D-D/A-Transfers

Das Timing der D/A-Wandlung ist eng mit dem der A/D-Wandlung verknüpft, um einen synchronen I/O-Transfer zu ermöglichen. Damit der D/A-Wandler beim Start einen definierten Ausgangswert aufweist, wird er noch vor dieser Synchronisation mit dem ersten Wert des D/A-Buffers vorbesetzt.

Bei einem synchronen Start von A/D-Wandlung und D/A-Wandlung wird das physikalische D/A-Wandler-Ausgangssignal erst nach zwei Sample-Intervallen gültig! Dies ist die Folge davon, daß

1. `ldcmod` auf '1'gesetzt ist, d.h. das 'conv_start'-Signal zum Setzen des LDAC-Signals führt und
2. nach Beendigung des A/D-Transfers der nächste D/A-Wert in das 'D/A-Hold'-Register geschrieben wird.

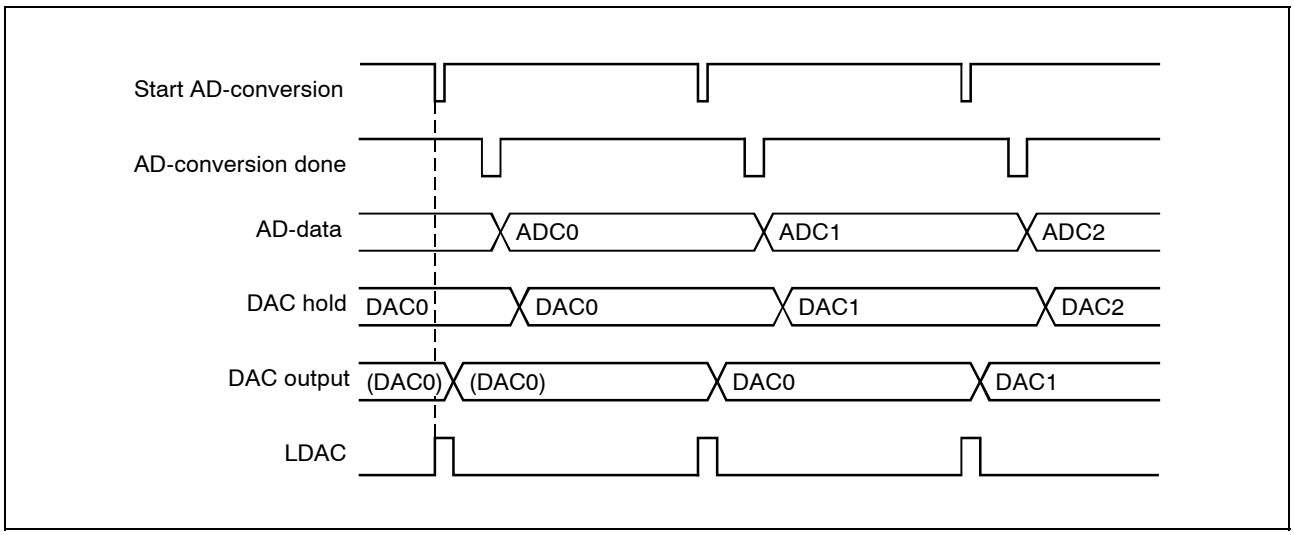

Abb. 6.4.1: Zusammenhang zwischen A/D-Wandler-Timing und D/A-Wandler-Timing

6.5 Buffer-Interrupts

Nach Beendigung eines ADC/DAC-Buffers wird IMMER ein VMEbus-Interrupt getriggert (unabhängig von `vvtrg`)! Zu diesem Zeitpunkt ist der Eintrag in der Buffer-Status-Struktur 'Buffer_ Number_in_Work' gültig.

Beispiel: ADC-Transfer, 16 Frames/Buffer, 4 Buffer, Continuous Transfers

Set `vadsrv` = 11_{DEZ}	-> Buffer_Number_in_Work = 1	
16 A/D-Frames übertragen	-> Buffer_Number_in_Work = 2	-> VMEbus-Interrupt
16 A/D-Frames übertragen	-> Buffer_Number_in_Work = 3	-> VMEbus-Interrupt
16 A/D-Frames übertragen	-> Buffer_Number_in_Work = 4	-> VMEbus-Interrupt
16 A/D-Frames übertragen	-> Buffer_Number_in_Work = 1	-> VMEbus-Interrupt

Diese Seite ist bewußt unbedruckt.

7. Buffer-Statistik

7.1 Aufbau der Statistik-Struktur

In der Statistik-Struktur können für jeden A/D-Wandler-Kanal und jeden Auxilary-Input statistische Werte abgelegt werden. Für jeden Kanal stehen hierfür 8 Bytes zur Verfügung. Sie sind wie folgt belegt:

Adreß-Offset (lokal):	0	2	4	6
Inhalt:	`Average`	`Max_Wert`	`Min_Wert`	reserviert

7.2 Adressen der Strukturzellen

relativer Adreßbereich VMEbus [HEX]	relativer Adreßbereich lokal [HEX]	Kanal
0600 ... 060E	0300 ... 0307	Stat. Aux_In_8
0610 ... 061E	0308 ... 030F	Stat. Aux_In_7
0620 ... 062E	0310 ... 0317	Stat. Aux_In_6
0630 ... 063E	0318 ... 031F	Stat. Aux_In_5
0640 ... 064E	0320 ... 0327	Stat. Aux_In_4
0650 ... 065E	0328 ... 032F	Stat. Aux_In_3
0660 ... 066E	0330 ... 0337	Stat. Aux_In_2
0670 ... 067E	0338 ... 033F	Stat. Aux_In_1
0680 ... 068E	0340 ... 0347	Stat. ADC_1
0690 ... 069E	0348 ... 034F	Stat. ADC_2
06A0 ... 06AE	0350 ... 0357	Stat. ADC_3
06B0 ... 06BE	0358 ... 035F	Stat. ADC_4
06C0 ... 06CE	0360 ... 0367	Stat. ADC_5
06D0 ... 06DE	0368 ... 036F	Stat. ADC_6
06E0 ... 06EE	0370 ... 0377	Stat. ADC_7
06F0 ... 06FE	0378 ... 037F	Stat. ADC_8
0700 ... 070E	0380 ... 0387	Stat. ADC_9
0710 ... 071E	0388 ... 038F	Stat. ADC_10
0720 ... 072E	0390 ... 0397	Stat. ADC_11
0730 ... 073E	0398 ... 039F	Stat. ADC_12
0740 ... 074E	03A0 ... 03A7	Stat. ADC_13
0750 ... 075E	03A8 ... 03AF	Stat. ADC_14
0760 ... 076E	03B0 ... 03B7	Stat. ADC_15
0770 ... 077E	03B8 ... 03BF	Stat. ADC_16