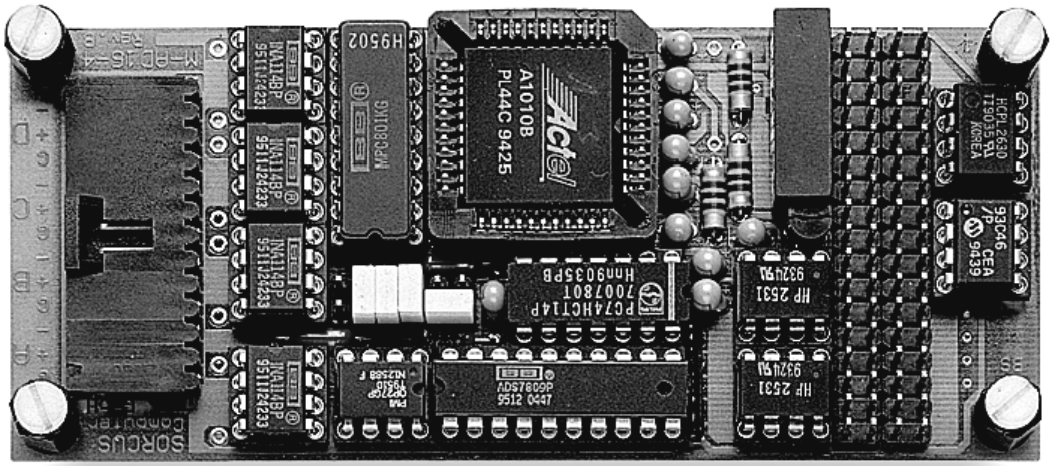
5. M-AD16-4

4 analoge Differenzeingänge, 16 Bit Auflösung galvanisch getrennt



Funktionsbeschreibung **5-3**

Abgleich des Moduls 5-4
Blockschaltbild 5-7
Technische Daten 5-8
Lieferumfang 5-9

Konfiguration und Einbau **5-10**

Lageplan 5-10
Jumper-Konfiguration 5-11
Verstärkungsfaktoren der Instrumentenverstärker 5-12
EEPROM-Inhalte 5-13

Steckerbelegung **5-20**

Modul-Device-Treiber M-AD16-4 **5-21**

Installationsparameter 5-21
Kanaleigenschaftsstruktur CPS_MAD164 5-21
Eingangsbereiche 5-22
Zugriff 5-22

Hochsprachenbibliothek **5-23**

Programmierung mit I/O-Zugriffen **5-27**

Lokale I/O-Adressen 5-27
Hinweise zur Programmierung 5-32

Funktionsbeschreibung

Das Modul M-AD16-4 verfügt über vier differentielle Analogeingänge (AIN-0 bis AIN-3) mit 16 Bit Auflösung und galvanischer Trennung der Eingänge von der Basiskarte und damit vom PC (nicht untereinander). Die Eingänge sind überspannungsfest bis ±40 Volt.

Vier zusätzliche interne Kanäle (AIN-4 bis AIN-7) dienen zur Diagnose des Moduls. Darin enthalten ist auch die Temperatur des A/D-Wandlers, um eine Temperaturkompensation der Meßergebnisse durchführen zu können.

Eingang	Funktion
AIN-0	Differenzeingang 1
AIN-1	Differenzeingang 2
AIN-2	Differenzeingang 3
AIN-3	Differenzeingang 4
AIN-4	Temperaturmessung
AIN-5	+5 Volt
AIN-6	-5 Volt
AIN-7	Masse (GND)

Das Modul besteht aus einem 16-Bit-A/D-Wandler und einem vorgeschalteten Analogmultiplexer.

Die Wandlungsgeschwindigkeit des A/D-Wandlers beträgt ca. 10 µs. Bei vier Eingangskanälen ergibt sich eine maximale Abtastrate von ca. 48 kHz (effektive Summenabtastrate einschließlich Einschwingzeit). Wird nur ein Eingangskanal gemessen, kann eine Abtastrate von bis zu 100 kHz erreicht werden[1].

Das Modul enthält einen Timer, mit dem die Settle-Time (Einschwingzeit) des Analogmultiplexers bei jeder Messung automatisch berücksichtigt werden kann. Die Werte für diesen Timer können im EEPROM abgelegt werden.

[1] Es ist auch ein AD-Wandler mit 5µs Wandlungsdauer lieferbar. Bei Verwendung dieses Wandlers erhöhen sich die erzielbaren Abtastraten.

Vier Eingangsspannungsbereiche können mit Steckbrücken (Jumper) eingestellt werden. Der ausgewählte Bereich gilt jeweils für alle Eingänge:

unipolar:	0 bis 5 Volt oder	0 bis 10 Volt
bipolar:	±5 Volt oder	±10 Volt

Sollen sehr kleine Signale gemessen werden, ist es möglich, die Verstärkungsfaktoren der Eingangs-Instrumentenverstärker anzuheben.

Im EEPROM auf dem Modul sind Korrekturwerte für Verstärkung und Offset abgelegt. Per Software können die Meßergebnisse anhand dieser Werte korrigiert werden.

Die Korrekturwerte können mit dem Programm SNW neu ermittelt und in das EEPROM des Moduls geschrieben werden.

Das Modul ist für alle Basiskarten des MODULAR-4 Systems geeignet. Es ist softwarekompatibel zum Modul M-AD16-8, belegt aber nur einen SPB-Steckplatz.

Abgleich des Moduls

Das M-AD16-4 Modul hat keine Abgleichpotentiometer, der Abgleich wird per Software durchgeführt. Die Korrekturwerte (Gain und Offset) des Moduls wurden im Werk ermittelt und ins EEPROM des Moduls eingetragen.

Sie können die Korrekturwerte mit SNW jederzeit neu bestimmen und so einen Abgleich des Moduls durchführen. Sie benötigen dazu lediglich zwei Referenzspannungen. Öffnen Sie mit 'Test/Modul' das Testfenster zum M-AD16-4 und schalten Sie dann mit 'Test/Test-Optionen/Abgleich' zum Abgleichdialog um. Folgen Sie den dort angegebenen Anweisungen. Für die Bestimmung der Korrekturwerte und deren Verwendung stehen neben SNW in dem mitgelieferten Modul-Device-Treiber Prozeduren zur Verfügung, so daß Sie sich darum in der Regel nicht zu kümmern brauchen. Falls es doch nötig sein wird, gibt das untenstehende Kapitel nähere Auskunft über die Berechnung der Korrekturwerte.

Bestimmung der Korrekturwerte:

Die gewandelten Meßwerte (X_M) des M-AD16-4 Moduls werden vom Modul-Device-Treiber einer linearen Korrektur gemäß den folgenden Formeln unterzogen, worin *GAIN* und *OFFSET* für die sogenannten Korrekturwerte stehen. Dazu müssen nacheinander zwei unterschiedliche Spannungen angelegt werden ($SOLL_1$ und $SOLL_2$). Mit den gemessenen Werten IST_1 und IST_2 können dann die Korrekturfaktoren berechnet werden.

In den bipolaren Bereichen ±5 V bzw. ±10 V (X_M von -32768 bis 32767) gelten folgende Formeln:

Korrekturformel für bipolare Meßbereiche (z. B. ±10 V):

$$X_{Kor} = GAIN \cdot \frac{X_M + 32768}{32768} + OFFSET + X_M$$

$$(-32768 \leq X_M \leq 32767)$$

Abgleichwerte für bipolare Meßbereiche (z. B. ±10 V):

$$GAIN = 32768 \cdot \frac{(SOLL_2 - SOLL_1) - (IST_2 - IST_1)}{IST_2 - IST_1}$$

$$OFFSET = SOLL_1 - IST_1 - GAIN \cdot \frac{IST_1 + 32768}{32768}$$

$$(-32768 \leq SOLL_1 \leq 32767)$$

$$(-32768 \leq SOLL_2 \leq 32767)$$

$$(-32768 \leq IST_1 \leq 32767)$$

$$(-32768 \leq IST_2 \leq 32767)$$

In den unipolaren Bereichen 0 bis 5 V bzw. 0 bis 10 V (X_M von 0 bis 65535) gelten folgende Formeln:

Korrekturformel für unipolare Meßbereiche (z. B. 0 bis 10 V):

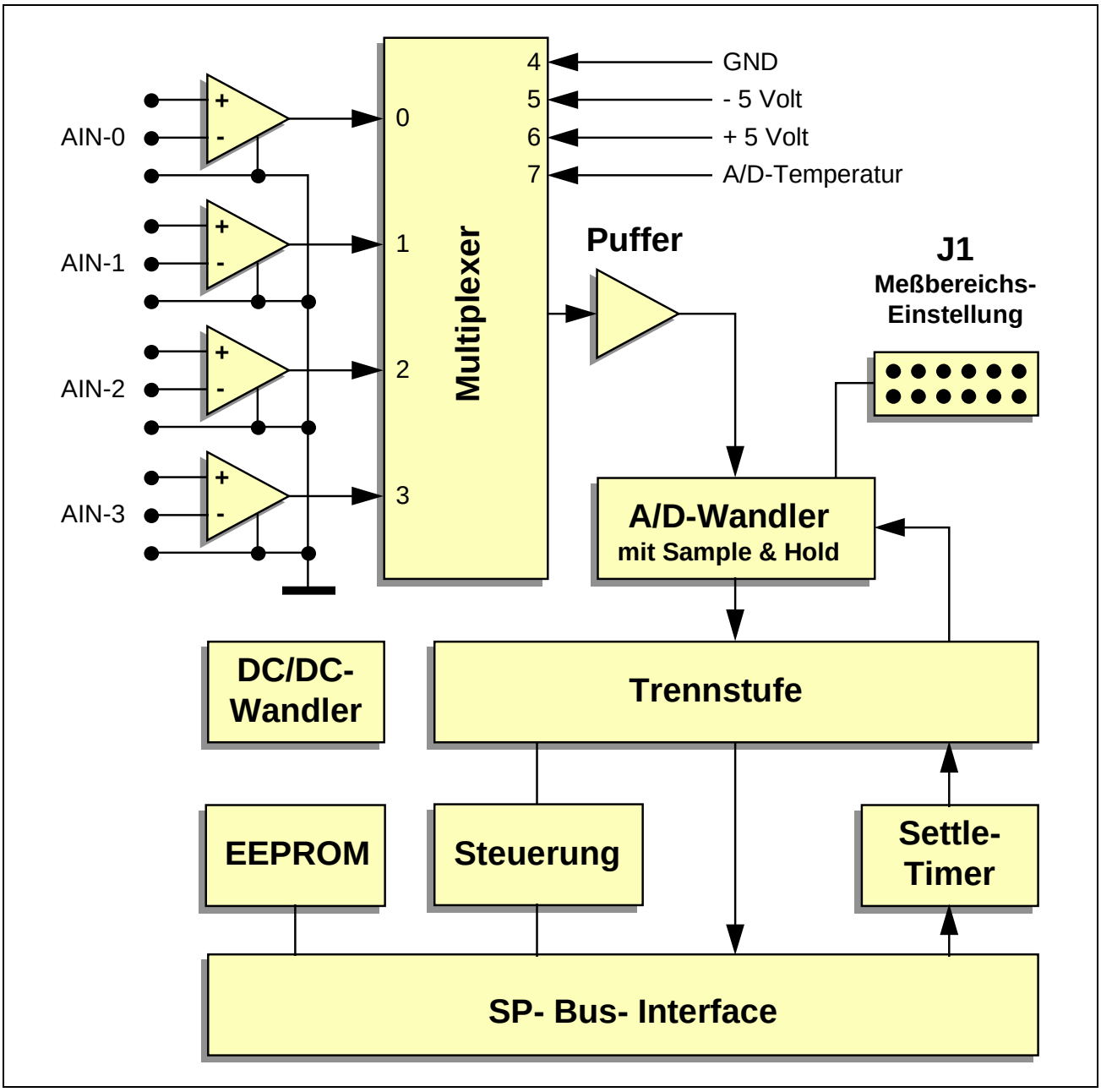
$$X_{Kor} = GAIN \cdot \frac{X_M}{32768} + OFFSET + X_M$$
$$(0 \leq X_M \leq 65535)$$

Abgleichwerte für unipolare Meßbereiche (z. B. 0 bis 10 V)

$$GAIN = 32768 \cdot \frac{(SOLL_2 - SOLL_1) - (IST_2 - IST_1)}{IST_2 - IST_1}$$
$$OFFSET = SOLL_1 - IST_1 - GAIN \cdot \frac{IST_1}{32768}$$
$$(0 \leq SOLL_1 \leq 65535)$$
$$(0 \leq SOLL_2 \leq 65535)$$
$$(0 \leq IST_1 \leq 65535)$$
$$(0 \leq IST_2 \leq 65535)$$

! *Die Abgleichwerte dürfen nur Werte von -32768 bis 32767 annehmen (16-Bit-Zweierkomplement).*

Blockschaltbild



Technische Daten

Parameter	Wert	Einheit
Anzahl der Eingänge:		
differentiell	4	-
intern (für Diagnosezwecke)	4	-
Eingangsspannungsbereiche:		
bipolar	±5 / ±10	V
unipolar	0 bis 5 / 0 bis 10	V
Auflösung	16	Bit
Linearität, integral	±4	LSB
Full-Scale-Error (inkl. Offset), max. [1]	± 50	mV
Offset-Error: [1]		
bipolar Zero, max.	±10	mV
unipolar Zero, max.	±3	mV
Überspannungsfest	±40	V
Eingangsimpedanz	10000	MΩ
Wandlungsdauer (A/D-Wandler ADS7809), max.	10	µs
Wandlungsdauer (A/D-Wandler AD977), max.	5	µs
Effektive Abtastrate pro Kanal (mit MODULAR-4/486, 1 MB, 66 MHz, ADS7809):		
1 Kanal (mit Settle-Time = 0), max.	100	kHz
4 Kanäle (mit 18 µs Settle-Time), max.	48	kHz
Betriebstemperaturbereich	0 bis 60	°C
Stromaufnahme 5 V (±5%)	300	mA
Abmessungen (L x B x H)	106 x 45 x 15	mm

[1] Kann per Software auf ±4 LSB korrigiert werden.

Lieferumfang

Zum Lieferumfang des Moduls gehören:

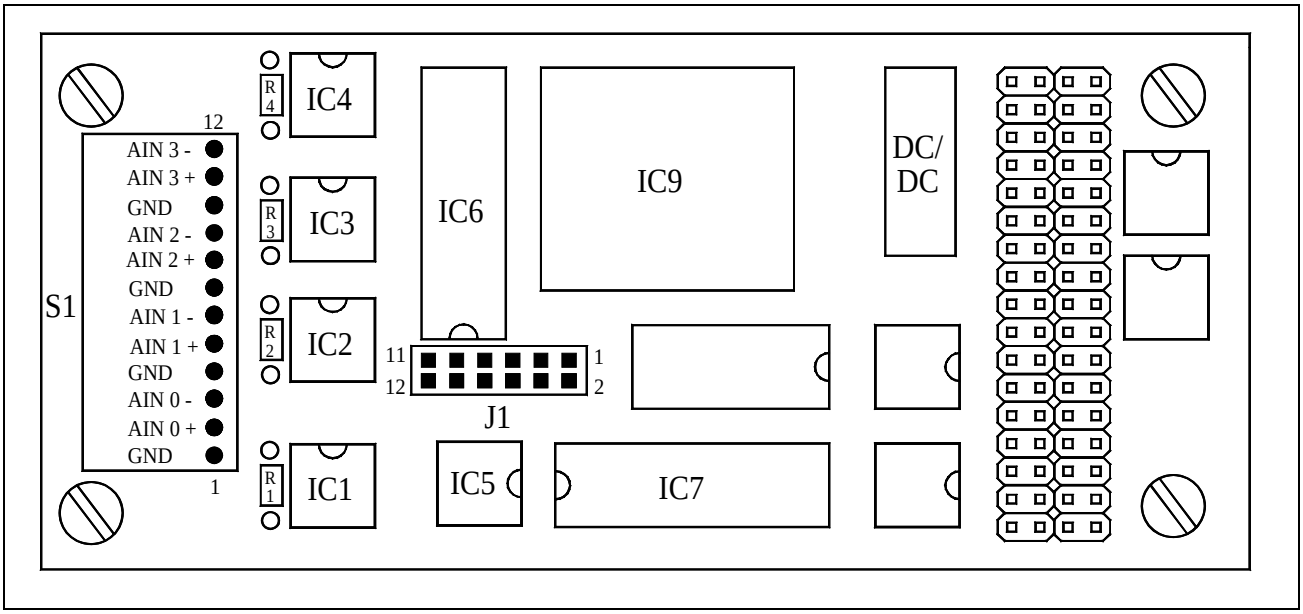
- Modul M-AD16-4.
- Anschlußstecker AMP 12-polig.

SORCUS Bestell-Nummer:	FA-1831	(Buchsenstecker)
	FA-1832	(Buchsenkontakte)
AMP Bestell-Nummer:	1-104257-1	(Buchsenstecker)
	166722-4	(Buchsenkontakte)

- Datenträger mit Modul-Device-Treiber, Testprogramm und Beispielen.

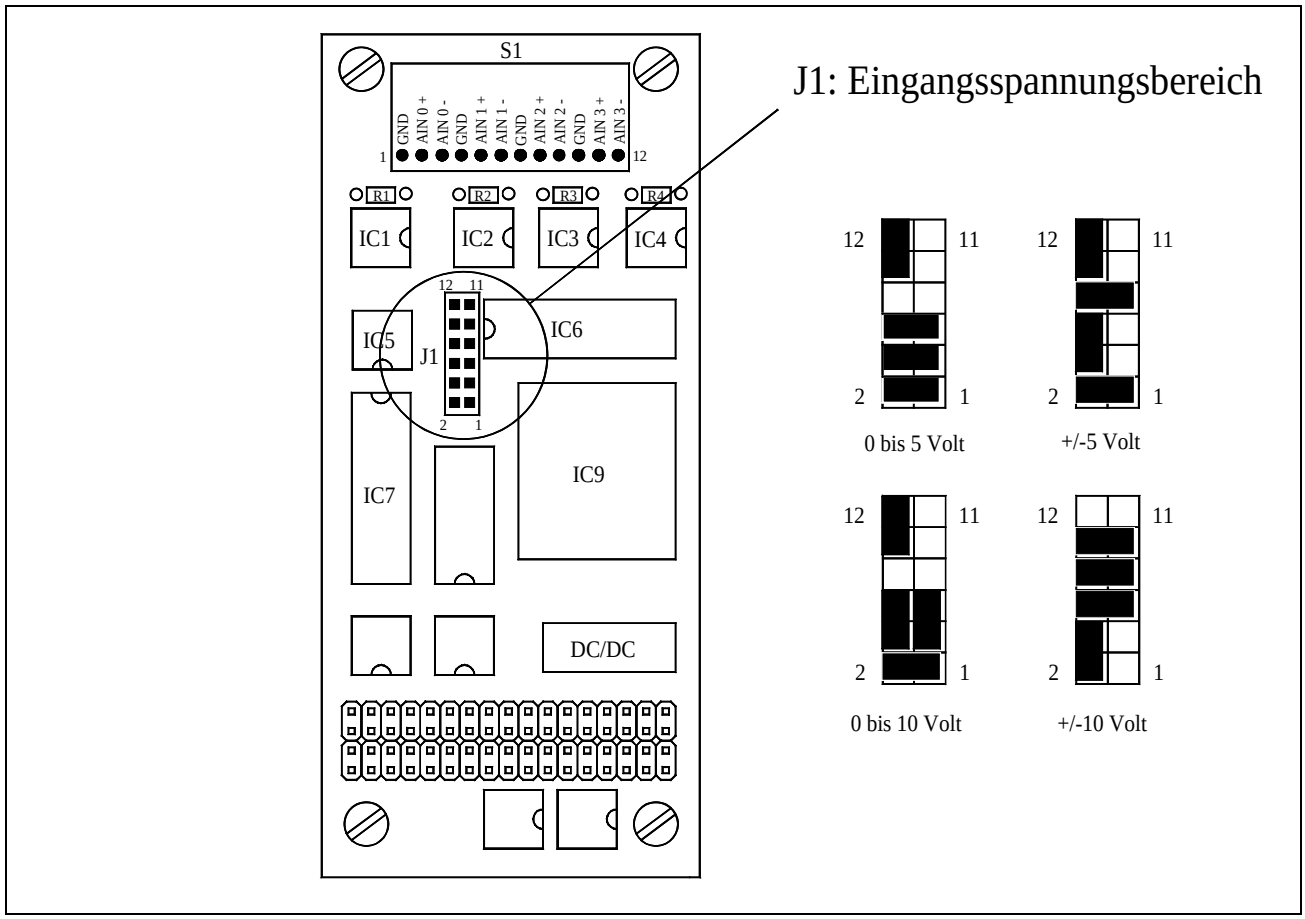
Konfiguration und Einbau

Lageplan



Jumper-Konfiguration

Vor Einbau des Moduls M-AD16-4 muß mittels des Jumperfeldes J1 (12 Pins) der Eingangsspannungsbereich des A/D-Wandlers eingestellt werden.



Die Jumper-Einstellungen müssen in das EEPROM des Moduls eingetragen werden, damit durch die Software festgestellt werden kann, mit welchem Bereich gemessen wird (siehe Seite 5-16).

Verstärkungsfaktoren der Instrumentenverstärker

Durch Hinzufügen eines Widerstandes (R1 bis R4, siehe Lageplan auf Seite 5-10) kann der Verstärkungsfaktor eines Instrumentenverstärkers (INA 114) für jeden der vier Differenzeingänge individuell angehoben werden. Der Verstärkungsfaktor für einen Eingang (V_{Kanal}) berechnet sich wie folgt:

$$V_{Kanal} = 1 + \frac{50\ \text{k}\Omega}{R_{Kanal}}$$

Es ist auch möglich, die Instrumentenverstärker durch einen pinkompatiblen Typ (INA 131, Verstärkungsfaktor 100) zu ersetzen.

EEPROM-Inhalte

Werkseitig ist bereits eine Konfiguration im EEPROM voreingestellt:

WORT	Binär		Hex.	Bedeutung (Kurzinfo)
0	0010 0001	0010 1011	212bh	Kennung
1	0000 0000	0000 0001	0001h	Reserviert
2	0000 0000	0000 0000	0000h	Hardware
3	0000 0000	1110 0100	00e4h	Jumper
4	0000 0000	0000 0000	0000h	Offset-Korrektur IN-0
5	0000 0000	0000 0000	0000h	Gain-Korrektur IN-0
6	0000 0000	0000 0000	0000h	Offset-Korrektur IN-1
7	0000 0000	0000 0000	0000h	Gain-Korrektur IN-1
8	0000 0000	0000 0000	0000h	Offset-Korrektur IN-2
9	0000 0000	0000 0000	0000h	Gain-Korrektur IN-2
10	0000 0000	0000 0000	0000h	Offset-Korrektur IN-3
11	0000 0000	0000 0000	0000h	Gain-Korrektur IN-3
12	0000 0000	0000 0000	0000h	Offset-Korrektur IN-4
13	0000 0000	0000 0000	0000h	Gain-Korrektur IN-4
14	0000 0000	0000 0000	0000h	Offset-Korrektur IN-5
15	0000 0000	0000 0000	0000h	Gain-Korrektur IN-5
16	0000 0000	0000 0000	0000h	Offset-Korrektur IN-6
17	0000 0000	0000 0000	0000h	Gain-Korrektur IN-6
18	0000 0000	0000 0000	0000h	Offset-Korrektur IN-7
19	0000 0000	0000 0000	0000h	Gain-Korrektur IN-7
20	0000 0001	0000 0000	0100h	Settle-Time 5 V
21	0000 0001	0000 0000	0100h	Settle-Time 10 V
22	0000 0001	0000 0000	0100h	Settle-Time ±5 V
23	0000 0001	0000 0000	0100h	Settle-Time ±10 V
24	0000 0000	0000 0001	0001h	Moderegister (Betriebsart)
25	0000 0000	0000 0000	0000h	Reserviert
…	…	…	…	…
31	0000 0000	0000 0000	0000h	Reserviert

WORT-0: Typ und Version des Moduls (darf nicht geändert werden)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	1	0	0	0	0	1	0	0	1	0	1	0	1	1	WORT-0: Kennung
								0	0	1	0	1	0	1	1	Modultyp: 43 = M-AD16-4
				0	0	0	1									Revision: 1 = A, 2 = B
			0													Reserviert
0	0	1														Kennung

WORT-1: Initialisierung

In diesem Wort kann eingestellt werden, ob das Modul nach dem Einschalten und bei einem Hardware-Reset entsprechend den Eintragungen im EEPROM initialisiert wird (Bit-0 = 1) oder nicht (Bit-0 = 0).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	WORT-1: Initialisierung
																geändert am: von:
															1	Init nach Hardreset: 0 = nein, 1 = ja
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		Reserviert

WORT-2: Hardware-Ausstattung (nicht vom Anwender zu ändern)

Dieses Wort enthält die Typen der wichtigsten integrierten Schaltungen auf dem Modul.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	WORT-2: Hardware (werks. Einst.)
												0	0	0	0	A/D-Wandler (IC7) 16 Bit: ADS7809P = 0000 ADS7809PB = 0001 AD977 = 0010 12 Bit: ADS7808 = 0100 ADS7808PB = 0101
								0	0	0	0					Analogmultiplexer (IC6) MPC801KG = 0000 MPC801SG = 0001
				0	0	0	0									Instrumentenverstärker (IC1 bis 4) INA114BP = 0000 INA114BG = 0001 INA114AP = 0010 INA114AG = 0011
0	0	0	0													Reserviert

WORT-3: Jumperfeld J1 (Meßbereich)

Werkseitig ist der Meßbereich auf ±10 Volt eingestellt.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0	WORT-3: Jumper J1 (werks. Einst.)
																geändert am: von:
															0	Pin 1 und 2 (1 = verbunden)
														0		Pin 3 und 4 (1 = verbunden)
													1			Pin 5 und 6 (1 = verbunden)
												0				Pin 3 und 5 (1 = verbunden)
											0					Pin 4 und 6 (1 = verbunden)
										1						Pin 7 und 8 (1 = verbunden)
									1							Pin 2 und 4 (1 = verbunden)
								1								Pin 9 und 10 (1 = verbunden)
							0									Pin 10 und 12 (1 = verbunden)
0	0	0	0	0	0	0										Reserviert

Gültige Einstellungen für EEPROM-WORT-3

Meßbereich	**Binär**		**Hex.**
0 bis 5 V	0000 0001	0000 0111	0107h
0 bis 10 V	0000 0001	0001 1001	0119h
±5 V	0000 0001	0011 0001	0131h
±10 V	0000 0000	1110 0100	00e4h

WORT-4 bis WORT-19: Korrekturwerte

Jedem der acht Eingangskanäle sind im EEPROM zwei Korrekturwörter zugeordnet (WORT-4 bis WORT-19).

Das erste Wort enthält den Korrekturwert für Offset, das zweite Wort den für Verstärkung (Gain) des Kanals. Die Korrekturwerte sind in Form von Zweierkomplementwerten (Zahlenbereich von -32768 bis +32767, Berechnung siehe Seite 5-4) abgelegt.

Binär		Hex.	Dezimal
1000 0000	0000 0000	8000h	-32768
1111 1111	1111 1111	ffffh	-1
0000 0000	0000 0000	0000h	0
0000 0000	0000 0001	0001h	1
0111 1111	1111 1111	7fffh	32767

Kanal	Offset-Korrektur	Gain-Korrektur
0	WORT-4	WORT-5
1	WORT-6	WORT-7
...	...	...
7	WORT-18	WORT-19

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	WORT-4 bis WORT-18: Offset-Korrektur (werks. Einst.)
																geändert am: von:
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Offset (-32768 bis +32767)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	WORT-5 bis WORT-19: Gain-Korrektur (werks. Einst.)
																geändert am: von:
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Gain (-32768 bis +32767)

WORT-20 bis WORT-23: Settle-Timer-Werte

Für jeden der vier möglichen Meßbereiche wird ein Settle-Timer-Wert (in Vielfachen von 100 ns) gespeichert (0 bis 65535 * 100 ns bzw. 0000h bis ffffh * 100 ns).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	WORT-20: Settle-Timer für Bereich 5 V (werks. Einst.)
																geändert am: von:
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Settle-Timer-Wert

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	WORT-21: Settle-Timer für Bereich 10 V (werks. Einst.)
																geändert am: von:
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Settle-Timer-Wert

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	WORT-22: Settle-Timer für Bereich ±5 V (werks. Einst.)
																geändert am: von:
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Settle-Timer-Wert

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	WORT-23: Settle-Timer für Bereich ±10 V (werks. Einst.)
																geändert am: von:
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Settle-Timer-Wert

WORT-24: Moderegister (Betriebsart)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	WORT-24: Moderegister (werks. Einst.)
																geändert am: von:
								0	0	0	0	0	0	0	0	Mode
0	0	0	0	0	0	0	1									Reserviert

Steckerbelegung

Die vier Differenzeingangssignale werden über einen 12-poligen Steckverbinder (AMP) und ein entsprechendes Anschlußkabel mit dem Modul verbunden.

Bitte beachten Sie, daß GND am Bezugspotential der Signalquelle angeschlossen werden muß.

Pin	Anschluß
1	GND
2	AIN-0 (+)
3	AIN-0 (-)
4	GND
5	AIN-1 (+)
6	AIN-1 (-)
7	GND
8	AIN-2 (+)
9	AIN-2 (-)
10	GND
11	AIN-3 (+)
12	AIN-3 (-)

Modul-Device-Treiber M-AD16-4

Allgemeine Hinweise zum Umgang mit Modul-Device-Treibern finden Sie in der Einführung im Abschnitt 'Grundlagen zum Modul-Device-Treiber'.

Installationsparameter

Parameter	Wert
Dateiname:	ML8D2B00.LIB
Programmnummer:	42Dh
Tasknummer:	Steckplatz des Moduls
Interruptnummer:	0
Länge des Datenbereichs:	0
Flags:	800h

Befehl in INS-Datei (z.B. für Steckplatz 1):

M8INST ML8D2B00 042D 0001 00 000000 00000800

Kanaleigenschaftsstruktur CPS_MAD164

Ofs	Strukturelement	Datentyp	Bedeutung
0	*.usDevice*	USHORT	Device-Typ: *DEVICE_AIN_DIFF* = 0202h
2	*.usIndexFirst*	USHORT	Index des ersten Device[1] (0 bis 7)
4	*.usIndexLast*	USHORT	Index des letzten Device (0 bis 7)
6	*.usFlags*	USHORT	Mögliche Flags: Bit 0: *_CP_EXCLUSIVE* = 1 (exklusiv) Bit 2: *_CP_UNCORRECTED* = 4 (unkorrigiert) Alle anderen Bits = 0.
8	*.usReadMode*	USHORT	Lesemodus: *IO_MODE_DIRECT* = 1 (direkt) *IO_MODE_LATCH* = 2 (aus dem Eingangspuffer[2].)
10	*.usOverSampling*	USHORT	Oversampling: 1, 2, 4, 8, 16, 32, 64, 128 (1 = ohne)

[1] Device 4 bis 7 sind on-Board Zusatzkanäle: AD-Temperatur, -5V, +5V, GND.

[2] Schnellstmögliche Betriebsart (nur bei Einzelkanal-Messungen möglich): es wird das Ergebnis der vorherigen Wandlung ausgegeben und die nächste Wandlung gestartet. Der angegebene Oversampling-Faktor wird ignoriert.

Eine Besonderheit des Moduls M-AD16-4 besteht in der Möglichkeit, die Abtastrate zu verdoppeln (.usOversampling = 1, .usReadMode = IO_MODE_LATCH). Dabei wird das Ergebnis aus dem Latch des AD-Wandlers gelesen und die Wandlung erneut gestartet. Bitte beachten Sie, daß die Ergebnisse jeweils zur vorherigen Wandlung gehören.

Eingangsbereiche

Der Eingangsbereich der Analogeingänge wird per Jumper für alle Kanäle gleichzeitig gesetzt.

Zugriff

Die Eingänge können gelesen werden mit **mddx_read_channel_short** (Einzelkanal) bzw. **mddx_read_channel_block** (Mehrkanal).

Hochsprachenbibliothek

Wie die Bibliothek eingebunden und verwendet wird, finden Sie in der Einführung im Abschnitt 'Hochsprachenbibliotheken'. Der Name der Bibliothek (*libname*) lautet **M043_LIB**, Sie finden sie im Verzeichnis (*pathname*) **SPB_MOD\BIB\M-AD16-4**. Vor allen anderen Routinen muß die Prozedur **m043_bib_startup** einmal aufgerufen werden.

m043_bib_startup — Initialisiere Modulbibliothek

Pascal PROCEDURE m043_bib_startup (micro_slot: byte);

C void EXPORT m043_bib_startup (byte micro_slot);

Funktion Diese Prozedur initialisiert die Modulbibliothek M043_LIB.

m043_set_conf_eeprom — Setze EEPROM-Konfiguration

Pascal PROCEDURE m043_set_conf_eeprom (micro_slot: byte);

C void EXPORT m043_set_conf_eeprom (byte micro_slot);

Funktion Diese Prozedur setzt die Konfiguration so, wie sie im EEPROM des Moduls angegeben ist. Korrekturwerte, Settle-Time und Default-Einstellungen nach dem Reset der Karte werden übernommen. Korrekturwerte, die mit der Prozedur **m043_set_correct_values** festgelegt wurden, werden überschrieben.

m043_set_correct_values Setze Korrekturwerte

Pascal PROCEDURE m043_set_correct_values (micro_slot: byte; channel: byte; corgain, coroffset: integer; settle: word);

C void EXPORT m043_set_correct_values (byte micro_slot, byte channel, short corgain, short coroffset, ushort settle);

Funktion Diese Prozedur setzt die Korrekturwerte Offset und Gain und die Settle-Time für einen Kanal. Die Korrekturwerte werden nur vorübergehend in der Bibliothek gespeichert und nicht im EEPROM abgelegt. Bei Neustart eines Programms werden sie gelöscht, d. h. = 0 gesetzt.

Parameter *corgain*: Faktor für Verstärkungskorrektur (siehe Seite 5-4)

coroffset: Faktor für Offsetkorrektur (siehe Seite 5-4)

settle: Wert für den Settle-Timer (16 Bit, 0000h = 0 bis ffffh = 65535). Die Settle-Time berechnet sich aus der folgenden Gleichung:

$$T_{Settle} = settle \cdot \frac{4}{TCLK}$$

TCLK ist der Timereingangstakt der Basiskarte und kann per Software auf 2,5 oder 10 MHz eingestellt werden (siehe Benutzerhandbuch zur MODULAR-4/486 Karte, EEPROM-Wort-3, Bit-4).[1]

[1] Bei MODULAR-4/486 Rev. B. ist der Takt nicht per Software einstellbar; je nach Bestückung (siehe Prüfbericht) beträgt er 2,5 oder 10 MHz.

m043_get_correct_values — Lies Korrekturwerte

Pascal
PROCEDURE m043_get_correct_values (micro_slot: byte;
channel: byte; var corgain, coroffset: integer; var settle: word;
var range, convtype: byte);

C
void EXPORT m043_get_correct_values (byte micro_slot,
byte channel, short *corgain, short *coroffset, ushort *settle,
byte *range, byte *convtype);

Funktion
Diese Prozedur ermittelt neben den Korrekturwerten für Verstärkung und Offset die eingestellte Settle-Time, den im EEPROM eingetragenen Meßbereich und die Datenbreite des A/D-Wandlers.

Parameter

channel: Nummer des Kanals, dessen Korrekturwerte und dessen Settle-Time ermittelt werden soll.

corgain: In der Bibliothek gespeicherter Wert für Verstärkungs-Korrektur (siehe m043_set_correct_values).

coroffset: In der Bibliothek gespeicherter Wert für Offset-Korrektur (siehe m043_set_correct_values).

settle: In der Bibliothek gespeicherter Wert für Settle-Time

range: In dieser Variablen wird der Meßbereich zurückgeliefert, der sich aus den im EEPROM eingetragenen Jumpereinstellungen ergibt.

Wert	Spannungsbereich
0	0 bis 5 V
1	0 bis 10 V
2	±5 V
3	±10 V

convtype: Gibt an, wieviel Bits der A/D-Wandler liefert. Mögliche Werte sind 12 und 16.

m043_in Lies Moduleingänge

Pascal PROCEDURE m043_in (micro_slot: byte; channel: byte; mode: byte; var data_var: longint);

C void EXPORT m043_in (byte micro_slot, byte channel, byte mode, long *data_var);

Funktion Die Prozedur liest die Eingangsspannung eines Kanals unter Berücksichtigung der Korrekturwerte und der Settle-Time. Dabei kann eingestellt werden ob:

- der angegebene Kanal gewandelt und das Ergebnis zurückgeliefert wird (normale Betriebsart).
- der angegebene Kanal gewandelt, aber das Ergebnis der vorherigen Wandlung zurückgeliefert wird. In dieser Betriebsart ist der Prozeduraufruf ca. 10 µs kürzer. Diesen Modus können Sie benutzen, wenn Sie mehrere Kanäle direkt hintereinander abtasten wollen.
- der angegebene Kanal wird mehrfach gewandelt (8, 16, 32 oder 64 mal) und der Mittelwert dieser Messungen zurückgeliefert. Mit diesem Verfahren kann statistisches Rauschen minimiert werden. Die Settle-Time wird nur bei der ersten Messung abgewartet.

Parameter *channel*: Nummer des zu wandelnden Kanals

mode: Betriebsart:

Wert	Funktion
0	Kanal *channel* wandeln und Ergebnis zurückliefern
1	Kanal *channel* wandeln und Ergebnis der vorigen Wandlung zurückliefern
2	Kanal *channel* 8fach wandeln und Mittelwert zurückliefern
3	Kanal *channel* 16fach wandeln und Mittelwert zurückliefern
4	Kanal *channel* 32fach wandeln und Mittelwert zurückliefern
5	Kanal *channel* 64fach wandeln und Mittelwert zurückliefern

datavar: Ergebnis der Wandlung.

Programmierung mit I/O-Zugriffen

Lokale I/O-Adressen

Alle Register werden bei einem Reset = 0 gesetzt.

Adresse	Zugriff	Funktion
MBA+1dh	W8	Modulreset
MBA+17h	W8x	Power Down (alle Optokoppler stromlos)
MBA+1ah	W16	Settle-Timer einstellen (wird bei einem Reset auf 0100h gesetzt)
MBA+1ch	R8/W8	Moderegister einstellen/lesen (siehe Seite 5-29) Bit-0: **immer = 1 setzen (M-AD16-4)** 0 M-AD16-8 Mode 1 M-AD16-4 Mode Bit-1: **Settle-Timer-Takt** 0 Settle-Timer-Takt = TCLK / 4 1 Settle-Timer-Takt = TCLK (TCLK = Timer-Taktfrequenz, SP-Bus Pin B15) Bit-2: **Timer-Mode** 0 Timer-A inaktiv 1 Timer-A (pos. Flanke) startet Settle-Timer/ Wandlung Bit-3: **Wandler-Typ** 0 16-Bit-A/D-Wandler ADS7809/AD977 1 12-Bit-A/D-Wandler ADS7808 Bit-4: **Ergebnisformat** 0 Offset-Binary (0 bis 65535) 1 Zweierkomplement (-32768 bis 32767) Bit-5: **Autostart**[1] 0 Settle-Timer startet Wandlung automatisch 1 keine automatische Wandlung

[1] Erst ab FPGA-Version 1.7 verfügbar.

Adresse	Zugriff	Funktion
MBA+08h	W8	Kanal einstellen, Settle-Timer starten (Settle-Timer startet Wandlung automatisch, siehe Mode-Register)
MBA+01h	W8x	Start Conversion (ohne Settle-Timer)
MBA+08h	R8	Status und Kanal lesen Bit-0 bis 2: Kanal (0 = 000b bis 7 = 111b) Bit-6[1] : Settle-Timer läuft (= 0) Settle-Timer fertig (= 1) Bit-7: Settle-Timer/Wandlung läuft (= 0) Settle-Timer/Wandlung fertig (= 1)
MBA+02h	R16	Ergebnis lesen
MBA+10h	W8x	Kanal 0 einstellen
MBA+11h	W8x	Kanal 1 einstellen
MBA+12h	W8x	Kanal 2 einstellen
MBA+13h	W8x	Kanal 3 einstellen
MBA+14h	W8x	Kanal 4 einstellen
MBA+15h	W8x	Kanal 5 einstellen
MBA+16h	W8x	Kanal 6 einstellen
MBA+17h	W8x	Kanal 7 einstellen
MBA+1eh	R8	FPGA-Version lesen Die unteren vier Bit enthalten die Revisionsnummer, die oberen vier Bit enthalten die Version: z. B. 0001 0101 = Version 1, Revision 5 0010 0001 = Version 2, Revision 1

[1] Erst ab FPGA-Version 1.7 verfügbar.

Ansteuerung des Moduls

Bedeutung der Modes (Moderegister MBA+1ch) für den Ablauf einer Wandlung

Die Betriebsart, in der das Modul betrieben wird, wird durch die Bits im Moderegister bestimmt. Vor der Verwendung des Moduls muß dieses Register gesetzt werden.

Die sechs Bits des Moderegisters können mit einem Zugriff auf I/O-Adresse MBA+1ch gesetzt oder zurückgelesen werden.

Mode-Bit-0: M-AD16-4/M-AD16-8 Umschaltung

Die Umschaltung des Moduls in den M-AD16-8-kompatiblen Mode (alle Mode-Bits = 0) ermöglicht den Einsatz des Moduls M-AD16-4 mit bereits vorhandener M-AD16-8 Software.

Nach dem Einschalten des Moduls (Reset) verhält sich das Modul M-AD16-4 wie ein M-AD16-8 Modul.

Mode-Bit-1: Takt des Settle-Timers einstellen

Da es Basiskarten mit unterschiedlichen Timer-Taktfrequenzen gibt (SP-Bus Pin B15 = TCLK), kann eine Anpassung der Settle-Timer-Laufzeit notwendig sein, um die optimale Einschwingzeit des Analogmultiplexers einstellen zu können. Mit SNW kann die Settle-Timer-Laufzeit automatisch bestimmt werden (nur für M-AD16-4 Mode).

Wenn Mode-Bit-1 = 0 ist (z. B. nach Reset), wird der Settle-Timer mit TCLK / 4 getaktet, sonst mit TCLK. TCLK ist der Timereingangstakt der Basiskarte und kann per Software auf 2,5 oder 10 MHz eingestellt werden.[1]

Mode-Bit-2: Timer-A-Mode

In dieser Betriebsart kann die Wandlung (inkl. Settle-Timer) durch eine positive Taktflanke an Timer-A der Basiskarte ausgelöst werden (nur im M-AD16-4 Mode). Ein Echtzeitprogramm auf der Basiskarte, das ebenfalls durch einen Interrupt von Timer-A aufgerufen wird, liest das Ergebnis ein, sobald es vorliegt.

Somit kann eine maximale Abtastrate erzielt werden, wobei außerdem die Wandlung zu einem definierten Zeitpunkt gestartet wird.

---

[1] Bei MODULAR-4/486 Rev. B. ist der Takt nicht per Software einstellbar; je nach Bestückung (siehe Prüfbericht) beträgt er 2,5 oder 10 MHz.

Mode-Bit-3: 16-/12-Bit-A/D-Wandler Umschaltung

Das Modul kann auch als 12-Bit-A/D-Modul eingesetzt werden, wenn statt des 16-Bit-Wandler-ICs ein pinkompatibler 12-Bit-Typ verwendet wird. Die Verwendung des 12-Bit-Wandlers erfordert das Setzen von Bit-3 im Moderegister.

Mode-Bit-4: Ergebnisformat Umschaltung

Das Wandlungsergebnis wird, unabhängig vom verwendeten A/D-Wandler-Baustein, immer als 16-Bit-Wert zurückgegeben. Im Zweierkomplement-Modus liefert das Modul das Ergebnis im 16-Bit-Zweierkomplement-Format (Mode-Bit-4 = 1).

Setzt man Mode-Bit-4 auf 0, so werden unabhängig vom Eingangsspannungsbereich alle Ergebnisse im Offset-Binary-Format ausgegeben:

Ergebnisformat Zweierkomplement (Mode-Bit-4 = 1):

	16-Bit	12-Bit
Obere Bereichsgrenze (+FS)	7fffh	07ffh
Bereichsmitte	0000h	0000h
Untere Bereichsgrenze (-FS)	8000h	f800h

Ergebnisformat Offset-Binary (Mode-Bit-4 = 0):

	16-Bit	12-Bit
Obere Bereichsgrenze (+FS)	ffffh	0fffh
Bereichsmitte	8000h	0800h
Untere Bereichsgrenze (-FS)	0000h	0000h

Die Berechnung des Spannungswertes erfolgt je nach Format:

$U = WERT \cdot LSB$ (Zweierkomplement)

bzw.

$U = WERT \cdot LSB + (-FS)$ (Offset-Binary)

$1\,LSB = \frac{+FS - (-FS)}{65536}$ (16 Bit)

bzw.

$1\,LSB = \frac{+FS - (-FS)}{4096}$ (12 Bit)

Die Umschaltung des Moduls in den M-AD16-8-kompatiblen Mode (alle Mode-Bits = 0) ermöglicht den Einsatz des Moduls M-AD16-4 mit bereits vorhandener

M-AD16-8 Software. Nach dem Einschalten des Moduls (Reset) verhält sich das Modul M-AD16-4 wie das M-AD16-8 Modul. Die Wandlungsergebnisse werden im Offset-Binary-Format ausgegeben.

Mode-Bit-5: Autostart

Wird Mode-Bit-5 = 1 gesetzt, startet die Wandlung nicht automatisch nach Ablauf des Settle-Timers. Der Status des Settle-Timers kann abgefragt werden und nach Ablauf der Settle-Time die Wandlung durch einen Schreibzugriff auf MBA+01h gestartet werden. In dieser Betriebsart ist es möglich, während der Wandlung bereits den nächsten Kanal anzuwählen (überlappende Messung)[1].

[1] Erst ab FPGA-Version 1.7 verfügbar.

Hinweise zur Programmierung

Messen im M-AD16-4 Mode (Mode-Bit-0 = 1, Mode-Bit-2 = 0)

In diesem Mode wird der interne Settle-Timer des Moduls benutzt.

Um das Modul zu initialisieren, müssen vor der eigentlichen Messung einmalig Mode-Bit-0 = 1 und Mode-Bit-2 = 0 gesetzt werden. Vor einer Wandlung wird die eingestellte Settle-Time berücksichtigt.

1. Mode einstellen (MBA+1ch).
2. Settle-Timer laden (zwei 8-Bit-Zugriffe (MBA+18h/19h) oder ein 16-Bit-Zugriff (MBA+1ah)).
3. Kanal einstellen: 0 bis 3 bzw. 4 bis 7 für Diagnose (MBA+08h)
   Settle-Timer startet automatisch
   Wandlung startet automatisch
4. Status Wandlung abfragen (MBA+08h). Solange Statusbit 7 = 0 wiederhole Punkt 4, sonst weiter mit Punkt 5.
5. Ergebnis einlesen (ein 16-Bit-Zugriff (MBA+02h)).

! *Es wird das Ergebnis der vorherigen Wandlung ausgegeben! Soll das aktuelle Ergebnis gelesen werden, muß eine zweite Wandlung gestartet werden. Diese kann bereits den nächsten Kanal messen, liefert aber das Ergebnis der vorherigen Wandlung.* [1]

6. Falls ein weiterer Wert vom selben Kanal gewandelt werden soll, weiter mit Punkt 7.
   Falls ein Wert von einem anderen Kanal gewandelt werden soll, weiter mit Punkt 3.
7. Start Conversion (ohne Settle-Timer, MBA+01h), weiter bei Punkt 4.

[1] Nach einem Reset liefert die 1. Wandlung ein undefiniertes Ergebnis zurück. Dieses Ergebnis muß verworfen werden.

Messen im M-AD16-4 Mode ohne Autostart[1]
(Mode-Bit-0 = 1, Mode-Bit-2 = 0, Mode-Bit-5 = 1)

In diesem Mode wird der interne Settle-Timer des Moduls benutzt.

Um das Modul zu initialisieren, müssen vor der eigentlichen Messung einmalig Mode-Bit-0 = 1, Mode-Bit-2 = 0 und Mode-Bit-5 = 1 gesetzt werden.

1. Mode einstellen (MBA+1ch).
2. Settle-Timer laden (zwei 8-Bit-Zugriffe (MBA+18h/19h) oder ein 16-Bit-Zugriff (MBA+1ah)).
3. Kanal einstellen: 0 bis 3 bzw. 4 bis 7 für Diagnose (MBA+08h)
   Settle-Timer startet automatisch
4. Settle-Timer-Status abfragen (MBA+08h). Solange Statusbit 6 = 0, wiederhole Punkt 4, sonst weiter mit Punkt 5.
5. Wandlung starten (MBA+01h).
6. Settle-Timer für nächsten Kanal starten (MBA+08h, optional).
7. Status Wandlung abfragen (MBA+08h). Solange Statusbit 7 = 0, wiederhole Punkt 7, sonst weiter mit Punkt 8.
8. Ergebnis einlesen (ein 16-Bit-Zugriff (MBA+02h)).
9. Weiter bei Punkt 4 oder Ende

[1] Erst ab FPGA-Version 1.7 verfügbar.

Messen im Timer-A-Mode (Mode-Bit-0 = 1, Mode-Bit-2 = 1)

In diesem Mode wird die Wandlung (mit Berücksichtigung der Settle-Time) durch eine positive Flanke von Timer-A ausgelöst.

Um das Modul zu initialisieren, müssen vor der eigentlichen Messung einmalig Mode-Bit-0 = 1 und Mode-Bit-2 = 1 gesetzt werden.

Der Settle-Timer wird mit dem jeweiligen Wert geladen (falls immer derselbe Kanal gemessen werden soll, kann der Settle-Timer = 0 gesetzt werden).

Timer-A muß auf einen Wert eingestellt werden, der größer als die zu erwartende Wandlungsdauer ist. Timer-A muß gleichzeitig einen Interrupt (positive Flanke) auf der Basiskarte auslösen (Timer-Mode 2).

1. Mode-Bits 0 und 2 des Moduls = 1 setzen (MBA+1ch).
2. Settle-Timer laden (zwei 8-Bit-Zugriffe (MBA+18h/19h) oder ein 16-Bit-Zugriff (MBA+1ah)).
3. Kanal einstellen: 0 bis 3 bzw. 4 bis 7 für Diagnose (MBA+08h).
4. Timer-A setzen (Timer-Mode 2) und Interrupt demaskieren.
5. Timer-A läuft ab und startet die Wandlung. Der dadurch ausgelöste Interrupt startet eine zugehörige Interrupt-Task auf der Basiskarte.
6. Die Interrupt-Task fragt den Status der Wandlung ab. Sobald die Wandlung abgeschlossen ist, wird das Ergebnis gelesen.

! *Es wird immer das Ergebnis der vorherigen Wandlung ausgegeben (siehe oben)! Erst durch das Lesen des Ergebnisses kann eine weitere Wandlung durch Timer-A gestartet werden.*

Anschließend kann (z. B. mit einem neuen Kanal) bei Punkt 5 fortgefahren oder die Messung durch Maskieren von Timer-A angehalten werden.

Beispiele zur Echtzeitprogrammierung (MODULAR-4/486)

Um maximale Abtastraten erzielen zu können, ist es notwendig, eigene Echtzeitprogramme zur Meßwerterfassung zu erstellen. Dabei empfiehlt sich die Verwendung von Timer-A, um zum einen die Wandlung auszulösen (Mode-Bit-0 und Mode-Bit-2 = 1), zum anderen eine II- oder DI-Task aufzurufen (durch einen Interrupt von Timer-A).

In der Hauptprozedur der Task (**main_proc()**) wird dann das Statusbit abgefragt und, sobald die Wandlung fertig ist, das Ergebnis eingelesen und abgespeichert.

Die vollständigen Beispiele zur schnellen Meßwerterfassung sind als M8P0322.C und M8P0323.C auf der mit dem Modul mitgelieferten Diskette enthalten.

Beispiel 1: DI-Task zur schnellen Meßwerterfassung eines Kanals
(mit Borland C++ 3.1)

Mit Beispiel 1 (Dateiname M8P0322.C) lassen sich Abtastraten von bis zu 100 kHz erreichen (MODULAR-4/486DX66 mit 10 MHz Timer-Takt und Settle-Time = 0, AD-Wandler ADS7809).

```
                                        .
                                        .
                                        .
void start(void)                    /* Startprozedur                        */
{
 ml8rt_entry();
                                    /* Modulbasisadresse berechnen          */
 mba = 0x100 * parameter.modul + 0x300;
 outportb(mba + 0x1C, 0x01);        /* Messmode einstellen (TA deaktiv.)    */
                                    /* Settle Timer einstellen              */
 outport(mba + 0x1A, parameter.settle);

                                    /* Zeiger auf den Datenbereichsanfang   */
 ml8rt_real_adr(parameter.tdt.datafirst, &dataptr);

 cntmess = 0;                       /* Anzahl Messungen zurücksetzen        */
                                    /* Timer-A initialisieren               */
 ml8rt_set_timer(TIMER_A, parameter.abtastrate, INT_MODE);

 parameter.status = RUNNING;

 outportb(mba + 0x1c, 0x07);        /* Messmode einstellen (TA aktiv.)      */
                                    /* Durch das Demaskieren des Timers     */
 ml8rt_unmask_int(IRQ_TIMER_A);     /* wird die Messung gestartet.          */

 ml8rt_exit();
}
```

```
void stop(void)                     /* Stop-Prozedur                        */
{
 ml8rt_entry();

 ml8rt_mask_int (IRQ_TIMER_A);      /* Timer-Interrupt sperren              */
 parameter.status = STOPPED;        /* Status auf "abgebrochen"             */

 ml8rt_exit();
}


void main_task(void)                /* HAUPTPROZEDUR DER TASK               */
{                                   /* Wird durch Aufruf des Timer-A Inter- */
 ml8rt_entry();                     /* rupts aufgerufen. Die Wandlung läuft.*/

 do{                                /* Status Wandlung lesen                */
   }while((inportb(mba + 0x08) & 0x80) != 0x80);

 *dataptr++ = inport(mba + 0x02);   /* Ergebnis lesen und abspeichern       */


 if(cntmess++ > parameter.anzmess) /* Wenn alle Messungen gemacht wurden : */
   {
    ml8rt_mask_int(IRQ_TIMER_A);
    outportb(mba + 0x1C, 0x00);     /* Messmode einstellen (TA deaktiv.)    */
    parameter.status = FINISHED;
   }
 outportb(SLAVE_INT_CONTR, 0x20);   /* EOI an den Slave-Controller senden   */
                                    /* (weil Timer-A am Slave-Controller    */
                                    /* angeschlossen ist                    */
 outportb(MASTER_INT_CONTR, 0x20); /* EOI an den Master-Controller senden  */
 ml8rt_exit_interrupt();
}
                                         .
                                         .
                                         .
```

Beispiel 2: DI-Task zur schnellen Meßwerterfassung von vier Kanälen (mit Borland C++ 3.1)

Beispielprogramm 2 (Dateiname M8P0323) erreicht eine Summenabtastrate von max. 36 kHz bei vier Kanälen (MODULAR-4/486DX33 mit 10 MHz Timer-Takt und Settle-Time = 18 µs).

```
                                        .
                                        .
                                        .
void main_task(void)                /* HAUPTPROZEDUR DER TASK               */
{                                   /* Wird durch Aufruf des Timer-A Inter- */
 ml8rt_entry();                     /* rupts aufgerufen. Die Wandlung läuft.*/

                                    /* nächsten Kanal anwählen              */
 outport(mba + 0x10 + parameter.kanal, 0x00);
 parameter.kanal %= 4;              /* nur Kanäle 0..3 zulässig             */

 do{                                /* Status Wandlung lesen                */
   }while((inportb(mba + 0x08) & 0x80) != 0x80);

 *dataptr++ = inport(mba + 0x02);  /* Ergebnis lesen und abspeichern       */

 if(cntmess++ > parameter.anzmess) /* Wenn alle Messungen gemacht wurden : */
   {
    ml8rt_mask_int(IRQ_TIMER_A);
    outportb(mba + 0x1C, 0x00);    /* Messmode einstellen (TA deaktiv.)    */
    parameter.status = FINISHED;
   }

 outportb(SLAVE_INT_CONTR, 0x20);  /* EOI an den Slave-Controller senden   */
                                   /* (weil Timer-A am Slave-Controller    */
                                   /* angeschlossen ist                    */
 outportb(MASTER_INT_CONTR, 0x20); /* EOI an den Master-Controller senden  */

 ml8rt_exit_interrupt();
}
                                        .
                                        .

.
```

Notizen