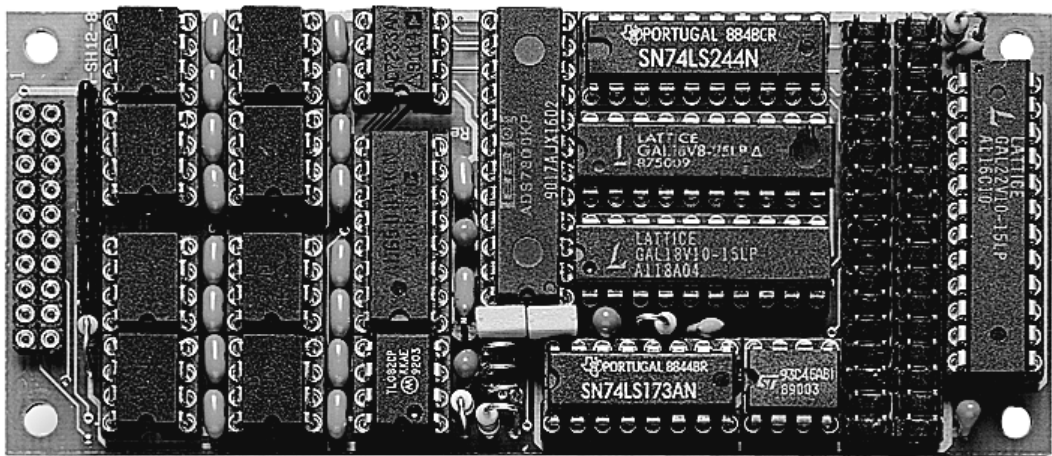


# 6. M-SH12-8

**8 analoge Eingänge, exakt zeitgleiche Abtastung,  
12 Bit Auflösung, 240 kHz max. Summenabtastrate**



M-SH12-8

<b>Funktionsbeschreibung</b>	<b>6-3</b>
Abgleich des Moduls .....	6-5
Blockschaltbild .....	6-7
Technische Daten .....	6-8
Lieferumfang .....	6-9
<b>Konfiguration und Einbau</b>	<b>6-10</b>
Lageplan .....	6-10
Widerstandsnetzwerke an den Eingängen .....	6-11
EEPROM-Inhalte .....	6-12
<b>Steckerbelegung</b>	<b>6-18</b>

<b>Modul-Device-Treiber M-SH12-8</b>	<b>6-19</b>
Installationsparameter .....	6-19
Kanaleigenschaftsstruktur CPS_MSH128.....	6-19
Initialisierung, Betriebsart und Interrupts .....	6-20
Initialisieren des Moduls.....	6-20
Definition der Betriebsart des Moduls.....	6-21
Kanal zur Steuerung der Interrupt-Eingänge.....	6-21
Abtast-Trigger per Software .....	6-22
Analoge Eingänge .....	6-23
Zugriff .....	6-23
Programmierung des analogen Ausgangs .....	6-24
Zugriff .....	6-24
<b>Hochsprachenbibliothek</b>	<b>6-25</b>
<b>Programmierung mit I/O-Zugriffen</b>	<b>6-31</b>
Lokale I/O-Adressen .....	6-31

# Funktionsbeschreibung

Das Modul enthält folgende Funktionseinheiten:

- 8 Single-Ended (SE) Analogeingänge
- Exakt zeitgleiche Abtastung durch 8 Sample/Hold Verstärker
- 12-Bit Auflösung, maximal 10  $\mu$ s Wandlungszeit
- Eingangsspannungsbereich:  $\pm 5$  Volt
- Triggern des Abtastzeitpunktes durch:
  - Software
  - Timer A der Basiskarte
  - Externer Impuls (positive oder negative Flanke), TTL-kompatibel
  - Über- oder Unterschreiten einer programmierbaren Schwelle an separatem Analogeingang
- Interrupt-Auslösung durch Abtast-Trigger
- Analogausgang, 12 Bit Auflösung, zur Einstellung der Trigger-Schwelle oder frei verwendbar,  $\pm 5$  Volt Ausgangsbereich
- Korrekturwerte für Abgleich im EEPROM auf dem Modul

Das Modul M-SH12-8 bietet 8 analoge Eingänge mit 12-Bit Auflösung und einem Eingangsspannungsbereich von  $\pm 5$  Volt. Die Analogeingänge haben einen sehr hohen Eingangswiderstand ( $> 20 \text{ M}\Omega$ ). Alle 8 Analogeingänge können exakt gleichzeitig abgetastet werden, da sich auf dem Modul 8 Sample/Hold-Verstärker befinden. Die Abtastung kann über verschiedene Quellen getriggert werden:

Außer dem Triggern per Software kann auch Timer A der Basiskarte verwendet werden, der dann auch sofort einen Interrupt auslösen kann. Timer A kann auch auf mehreren M-SH12-8 Modulen verwendet werden, alle Module tasten dann exakt zur gleichen Zeit ab. Zusätzlich stehen ein analoger und ein digitaler Trigger-Eingang zur Verfügung.

Bei Verwendung des analogen Trigger-Eingangs TRIG-0 besteht die Möglichkeit, eine Trigger-Schwelle zwischen -5 Volt und +5 Volt vorzugeben. Dabei kann programmiert werden, ob der Zeitpunkt des Überschreitens oder des Unterschreitens der Trigger-Schwelle den Trigger auslöst. Zur Einstellung der Trigger-Schwelle ist auf dem Modul ein 12-Bit D/A-Wandler vorhanden, dessen Ausgangsspannung auch an

einem Pin des Anschlußsteckers zur Verfügung steht. Wenn der analoge Trigger-Eingang also nicht benutzt wird, kann dieser Analogausgang für andere Zwecke verwendet werden (Ausgangsspannungsbereich  $\pm 5$  Volt).

Bei Verwendung des digitalen Trigger-Eingangs TRIG-1 kann die aktive Flanke (positiv oder negativ) programmiert werden.

Es ist immer nur einer der Abtast-Trigger aktiv. Unabhängig davon, welcher Abtast-Trigger angewählt ist, kann die Abtastung jederzeit auch per Software ausgelöst werden.

Gleichzeitig mit dem Auftreten des gewählten Abtast-Triggers (Timer A, analoger Trigger oder digitaler Trigger) kann auch ein Interrupt auf der Basiskarte ausgelöst werden, wobei per Software zwischen 3 Interrupt-Leitungen (IRQ-B, IRQ-C oder IRQ-D) gewählt werden kann.

Die Anwahl des Abtast-Triggers und der Interrupt-Auslösung geschieht über die Einstellung der Betriebsart (Mode 0 bis 15) des Moduls:

<b>Abtast-Trigger</b>	<b>Interrupt-Auslösung</b>	<b>Mode</b>
Per Software	Timer A Ausgang	1
	Externer digitaler Trigger	0
	Externer analoger Trigger	8
Timer A (pos. Flanke)	Timer A Ausgang	3
	Externer digitaler Trigger	2
	Externer analoger Trigger	10
Timer A (neg. Flanke)	Timer A Ausgang	5
	Externer digitaler Trigger	4
	Externer analoger Trigger	12
Externer digitaler Trigger (positive Flanke)	Externer digitaler Trigger	6
Externer digitaler Trigger (negative Flanke)	Externer digitaler Trigger	7
Externer analoger Trigger (Schwellenwertüberschreitung)	Externer analoger Trigger	14
Externer analoger Trigger (Schwellenwertunterschreitung)	Externer analoger Trigger	15

Im EEPROM auf dem Modul sind alle Einstellungen zur Initialisierung, Konfiguration und zum automatischen Abgleich (für Verstärkung und Offset für jeden Analogeingang) gespeichert. Sie bleiben auch nach dem Ausschalten des Systems erhalten. Beim Einschalten oder nach einem Hardware-Reset der MODULAR-4 Karte wird das Modul entsprechend den im EEPROM gespeicherten Daten konfiguriert.

## Hinweise für Optionen

1. Werkseitig kann der digitale Trigger-Eingang auch so konfiguriert werden, daß er als digitaler Ausgang (TTL-Pegel) dient und z.B. das Auftreten des Abtast-Triggers anzeigen kann. Damit ist es dann möglich, mehrere Module auch beim Abtast-Triggern per Software exakt zeitgleich zu triggern. Der Ausgang eines Moduls wird dann mit dem digitalen Trigger-Eingang eines anderen Moduls verbunden.
2. Jeder Kanal hat einen eigenen Eingangsverstärker. Standardmäßig ist das eine Sample/Hold Stufe. Es ist aber möglich, auch ohne diesen Eingangsverstärker zu arbeiten und direkt auf den Eingang des Multiplexers zu schalten. In diesem Fall kann auch der  $\pm 10$  Volt Eingangsspannungsbereich genutzt werden.

## Abgleich des Moduls

Das Modul M-SH12-8 wird digital per Software abgeglichen. Die Korrekturwerte (GAIN und OFFSET) des Moduls wurden im Werk ermittelt und ins EEPROM des Moduls eingetragen. Der Modul-Device-Treiber liest diese Werte und verwendet sie für die Korrektur. Der folgende Abschnitt gibt Auskunft über die Verwendung und Berechnung der Korrekturfaktoren.

Die gewandelten Werte ( $X_M$ ) des M-SH12-8 Moduls werden einer linearen Korrektur gemäß den folgenden Formeln unterzogen, worin GAIN und OFFSET für die sogenannten Korrekturwerte stehen.

$$X_{KOR} = (X_M + OFFSET) \cdot \left( 1 + \frac{GAIN + 128}{4096} \right)$$

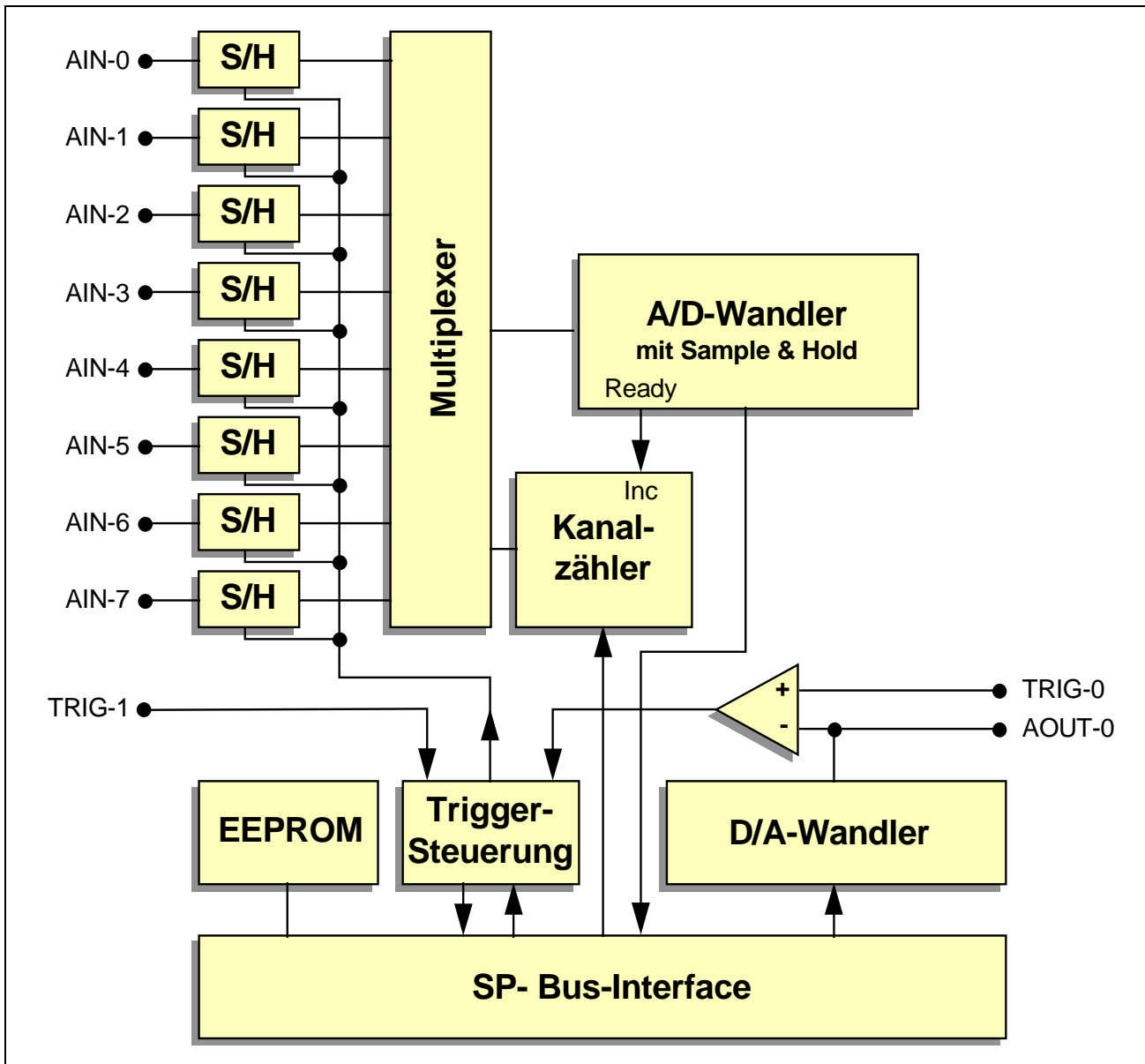
Die Korrekturfaktoren GAIN und OFFSET werden mit Hilfe von zwei Messungen bestimmt. Dabei werden nacheinander zwei Referenzspannungen angelegt ( $SOLL_1$

und  $SOLL_2$ ) und mit dem M-SH12-8 gemessen ( $IST_1$  und  $IST_2$ ). Nach den untenstehenden Formeln lassen sich daraus die Korrekturfaktoren für GAIN und OFFSET berechnen.

$$GAIN = 4096 \cdot \frac{SOLL_2 - SOLL_1}{IST_2 - IST_1} - 4224$$

$$OFFSET = SOLL_1 \cdot \frac{IST_2 - IST_1}{SOLL_2 - SOLL_1} - IST_1$$

## Blockschaltbild



## Technische Daten

Parameter	Wert	Einheit
Analogeingänge AIN-n (single-ended), Anzahl	8	-
Auflösung	12	Bit
Linearität <sup>1</sup>	$\pm 1/2$	LSB
Genauigkeit	$\pm 1$	LSB
Offset, max. <sup>2</sup>	$\pm 5$	mV
Verstärkungsfehler <sup>2</sup>	$\pm 0,1$	%
Eingangsbereich	$\pm 5$	V
Eingangsspannung, max. (absolut)	$\pm 12$	V
Eingangsimpedanz <sup>3</sup>	$> 20$	M $\Omega$
Wandlungszeit, typ.	2,5	$\mu$ s
Wandlungszeit + Akquisitionszeit, typ.	3	$\mu$ s
Summenabtastrate, max.	300	kHz
Analoger Trigger-Eingang TRIG-0		
Schwelle (programmierbar)	-5 bis +5	V
Eingangsspannung, max.	$\pm 12$	V
Eingangsimpedanz	220	k $\Omega$
Digitaler Trigger-Eingang TRIG-1		
Schwelle (Flanke programmierbar)	TTL	-
Eingangsimpedanz	10	k $\Omega$

<sup>1</sup> 1 LSB entspricht 2,44 mV bei  $\pm 5$ -V-Bereich.

<sup>2</sup> Per Software korrigierbar auf  $< 1$  LSB.

<sup>3</sup> Durch steckbare R-Netzwerke konfigurierbar, standardmäßig sind 100-k $\Omega$ -Widerstände eingesetzt.



Parameter	Wert	Einheit
Analogausgang AOUT-0		
Auflösung	12	Bit
Linearität <sup>1</sup>	±1/2	LSB
Genauigkeit	±1	LSB
Offset, max. <sup>2</sup>	±6	LSB
Verstärkungsfehler <sup>2</sup>	±8	LSB
Ausgangsbereich	±5	V
Ausgangsimpedanz (kurzschlußfest)	< 0,1	Ω
Wandlungszeit (Settle-Time), max./typ.	10/5	µs
Stromaufnahme, 5 V (±5%)	236	mA
(typ.) +12 V (±5%)	51	mA
-12 V (±5%)	44	mA

## Lieferumfang

- Modul M-SH12-8
- 20-poliger Pfostenstecker für Flachbandkabel
- Datenträger mit Modul-Device-Treiber und Bibliothek (Pascal und C)

<sup>1</sup> 1 LSB entspricht 2,44 mV bei ±5-V-Bereich.

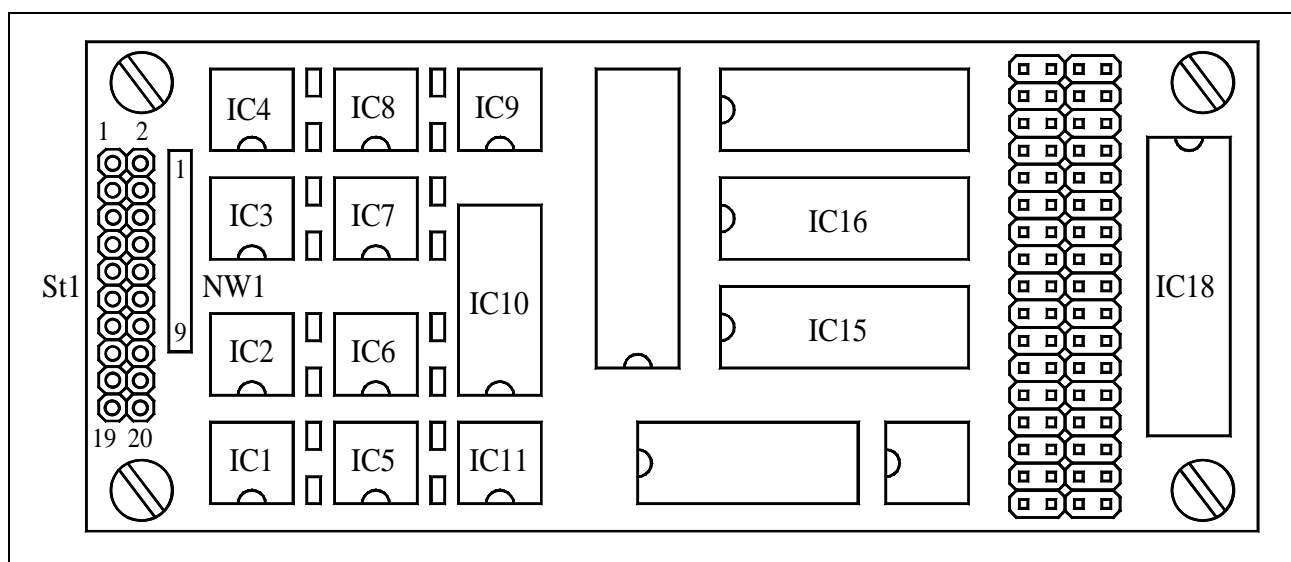
<sup>2</sup> Per Software korrigierbar auf < 1 LSB.

## Konfiguration und Einbau

Vor dem Einbau des Moduls beachten Sie bitte folgende Punkte:

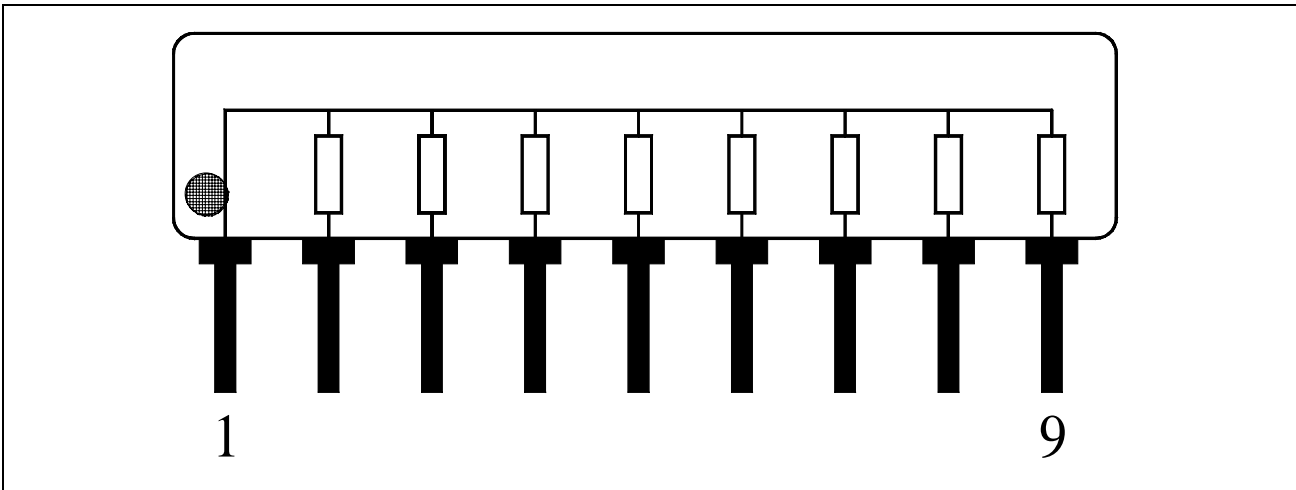
1. Die Anpassung der Eingänge (Widerstandsnetzwerke an den Eingängen) ist zu überprüfen und gegebenenfalls zu ändern (siehe Seite 6-11).
2. Alle Einstellungen erfolgen nach dem Einbau des Moduls per Software.

### Lageplan



## Widerstandsnetzwerke an den Eingängen

Die 8 Eingänge können über ein steckbares Widerstandsnetzwerk (ein 9-poliges Single-In-Line- (SIL) Netzwerk) abgeschlossen werden. Werkseitig ist ein Netzwerk mit 8 Widerständen à 100 k $\Omega$  eingesetzt. Im Lageplan (Seite 6-10) ist das Netzwerk mit NW1 bezeichnet:



## EEPROM-Inhalte

Werkseitig ist bereits eine Konfiguration im EEPROM voreingestellt:

WORT	Binär		Hex.	Bedeutung (Kurzinfor)
0	0010 0011	0001 1000	2318h	Modultyp: M-SH12-8
1	0000 0000	0000 0001	0001h	Init nach Reset
2	0001 0001	0001 0001	1111h	Bestückung
3	0001 0001	0001 0001	1111h	Bestückung
4	0000 0001	0001 0001	0111h	Bestückung, GALs
5	0000 0000	0000 0000	0000h	Konfiguration
6	0000 0000	0000 0000	0000h	Initialisierung von AOUT-0
7	0000 0000	0000 0000	0000h	Korrekturwerte für AIN-0
8	0000 0000	0000 0000	0000h	Korrekturwerte für AIN-1
9	0000 0000	0000 0000	0000h	Korrekturwerte für AIN-2
10	0000 0000	0000 0000	0000h	Korrekturwerte für AIN-3
11	0000 0000	0000 0000	0000h	Korrekturwerte für AIN-4
12	0000 0000	0000 0000	0000h	Korrekturwerte für AIN-5
13	0000 0000	0000 0000	0000h	Korrekturwerte für AIN-6
14	0000 0000	0000 0000	0000h	Korrekturwerte für AIN-7
15	0000 0000	0000 0000	0000h	Reserviert
...		...	...	...
22	0000 0000	0000 0000	0000h	Reserviert
23	0000 0000	0000 0000	0000h	Korrekturwerte für AOUT-0

**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	1	1	0	0	0	1	1	0	0	0

WORT-0: Kennung

								0	0	0	1	1	0	0	0
								0	0	1	1				
			0												
0	0	1													

Modultyp: 24 = M-SH12-8

Revision: 1 = A, 2 = B, 3 = C, etc.

Reserviert

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
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

WORT-1: Initialisierung

geändert am:

von:

Init nach Hardware-Reset: 0 = nein, 1 = ja

Reserviert

**WORT-2 bis WORT-4: Bestückung (nicht vom Anwender zu ändern)**

Jeder Analogeingang hat einen eigenen Eingangsverstärker. Standardmäßig ist das eine Sample/Hold-Stufe. Dieses Wort gibt Auskunft darüber, welches Eingangs-IC verwendet wird.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1

WORT-2: Bestückung

--	--	--	--	--	--	--	--

				0	0	0	1
--	--	--	--	---	---	---	---

Typ Eingangs-IC für AIN-0

0 = kein, 1 = Sample/Hold

--	--	--	--	--	--	--	--

0	0	0	1				
---	---	---	---	--	--	--	--

Typ Eingangs-IC für AIN-1

				0	0	0	1
--	--	--	--	---	---	---	---

--	--	--	--	--	--	--	--

Typ Eingangs-IC für AIN-2

0	0	0	1				
---	---	---	---	--	--	--	--

--	--	--	--	--	--	--	--

Typ Eingangs-IC für AIN-3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1

WORT-3: Bestückung

--	--	--	--	--	--	--	--

				0	0	0	1
--	--	--	--	---	---	---	---

Typ Eingangs-IC für AIN-4

0 = kein, 1 = Sample/Hold

--	--	--	--	--	--	--	--

0	0	0	1				
---	---	---	---	--	--	--	--

Typ Eingangs-IC für AIN-5

				0	0	0	1
--	--	--	--	---	---	---	---

--	--	--	--	--	--	--	--

Typ Eingangs-IC für AIN-6

0	0	0	1				
---	---	---	---	--	--	--	--

--	--	--	--	--	--	--	--

Typ Eingangs-IC für AIN-7

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	1	0	0	0	1

WORT-4: Bestückung, GALs

--	--	--	--	--	--	--	--

				0	0	0	1
--	--	--	--	---	---	---	---

Version GAL IC15

1=A, 2=B, etc.

--	--	--	--	--	--	--	--

0	0	0	1				
---	---	---	---	--	--	--	--

Version GAL IC16

				0	0	0	1
--	--	--	--	---	---	---	---

--	--	--	--	--	--	--	--

Version GAL IC18

0	0	0	0				
---	---	---	---	--	--	--	--

--	--	--	--	--	--	--	--

Reserviert

WORT-5: Konfiguration des Moduls (für Initialisierung)

15141312111098

00000000

76543210

00000000

WORT-5: Konfiguration

geändert am: von:

00000000

Mode (0 bis 15)

Interrupt-Leitung

0 = keine, 5 = IRQ-B, 6 = IRQ-C,  
7 = IRQ-D

WORT-6: Wert für Initialisierung des Analogausgangs

Ausgangsspannung (Volt)	Ausgabewert (Hex.)
4,9976	07ffh
0,0000	0000h
-5,0000	f800h

15141312111098

00000000

76543210

00000000

WORT-6: Analogausgang AOOUT-0

geändert am: von:

00000000

00000000

12-Bit-Zweierkomplement

WORT-7 bis WORT-22:

Korrekturwerte für die analogen Eingänge

Jedem Analogeingang ist ein Wort für Korrekturwerte zugeordnet. Die unteren 8 Bit (Low Byte) enthalten den Korrekturwert für OFFSET, die oberen 8 Bit (High Byte) den für GAIN. WORT-15 bis -22 sind für zukünftige Entwicklungen vorgesehen.

Wort	Korrekturwerte für
7	AIN-0
8	AIN-1
9	AIN-2
10	AIN-3
11	AIN-4
12	AIN-5
13	AIN-6
14	AIN-7

15141312111098

00000000

76543210

00000000

WORT-7 bis WORT-14: Korrekturwerte

W-7 geändert am: von:

W-8 geändert am: von:

W-9 geändert am: von:

W-10 geändert am: von:

W-11 geändert am: von:

W-12 geändert am: von:

W-13 geändert am: von:

W-14 geändert am: von:

00000000

für OFFSET (+127 bis -128)

für GAIN (+127 bis -128)



WORT-23: Korrekturwerte für den analogen Ausgang

Das Format ist dasselbe wie für die analogen Eingänge (siehe WORT-7 bis WORT-22).

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
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WORT-23: Korrekturwerte für AOUT-0

geändert am: von:

für OFFSET (+127 bis -128)

für GAIN (+127 bis -128)

## Steckerbelegung

Das Modul wird über einen 20-poligen (2 x 10) Steckverbinder und ein entsprechendes Flachbandkabel mit der Außenwelt verbunden. Die ungeraden Pins (Pin 3 bis 19) sind mit Masse verbunden, die geraden Pins (Pin 4 bis 18) sind die Signalleitungen.

<b>Pin</b>	<b>Anschluß</b>
1	AOUT-0 (Analogausgang ( $\pm 5$ Volt))
2	TRIG-0 (Analoger Trigger-Eingang)
3	GND
4	AIN-0
5	GND
6	AIN-1
7	GND
8	AIN-2
9	GND
10	AIN-3
11	GND
12	AIN-4
13	GND
14	AIN-5
15	GND
16	AIN-6
17	GND
18	AIN-7
19	GND
20	TRIG-1 (Digitaler Trigger-Eingang)

# Modul-Device-Treiber M-SH12-8

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

## Installationsparameter

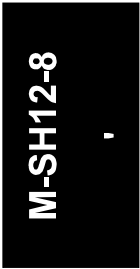
Parameter	Wert
Dateiname:	ML8D1800.LIB
Programmnummer:	42Eh
Tasknummer:	Steckplatz des Moduls
Interruptnummer:	0
Länge des Datenbereichs:	0
Flags:	800h

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

**M8INST ML8D1800 042E 0001 00 000000 00000800**

## Kanaleigenschaftsstruktur CPS\_MSH128

Ofs	Strukturelement	Datentyp	Bedeutung
0	.usDevice	USHORT	Device-Typ
2	.usIndexFirst	USHORT	Index des ersten Device
4	.usIndexLast	USHORT	Index des letzten Device
6	.usFlags	USHORT	Eigenschaftsflags
8	.usReadMode	USHORT	Lesemodus
10	.usWriteMode	USHORT	Schreibmodus



## Initialisierung, Betriebsart und Interrupts

Ofs	Strukturelement	Datentyp	Bedeutung
0	<i>.usDevice</i>	USHORT	Device-Typ: <i>DEVICE_CTRL</i> = 0401h
2	<i>.usIndexFirst</i>	USHORT	Device-Index: 0 = Initialisierung 1 = Betriebsart 2 = Interrupt
4	<i>.usIndexLast</i>	USHORT	= <i>.usIndexFirst</i>
6	<i>.usFlags</i>	USHORT	Mögliche Flags: Bit 0: <i>_CP_EXCLUSIVE</i> = 1 (exklusiv) Alle anderen Bits = 0.

Alle anderen Strukturelemente werden nicht ausgewertet.

## Initialisieren des Moduls

Die Initialisierung wird durch einen Schreibzugriff mit **mddx\_write\_channel\_byte** auf den Kanal gesteuert (der Übergabewert ist ungültig).

Die aktuelle Betriebsart (s.u.) kann durch einen Lesezugriff mit **mddx\_read\_channel\_byte** auf den Kanal ermittelt werden.

## Definition der Betriebsart des Moduls

Die Betriebsart wird durch einen Schreibzugriff mit **mddx\_write\_channel\_byte** auf den Kanal gesteuert. Das entsprechend zu setzende Byte kann folgende Werte annehmen:

<i>MSH128_SW_EXT_DIG_TRIG</i>	(= 0)	Betriebsart 0
<i>MSH128_SW_TIMER_A</i>	(= 1)	Betriebsart 1
<i>MSH128_TIMER_A_POS_EXT_DIG_TRIG</i>	(= 2)	Betriebsart 2
<i>MSH128_TIMER_A_POS_TIMER_A</i>	(= 3)	Betriebsart 3
<i>MSH128_TIMER_A_NEG_EXT_DIG_TRIG</i>	(= 4)	Betriebsart 4
<i>MSH128_TIMER_A_NEG_TIMER_A</i>	(= 5)	Betriebsart 5
<i>MSH128_EXT_DIG_POS_EXT_DIG_TRIG</i>	(= 6)	Betriebsart 6
<i>MSH128_EXT_DIG_NEG_EXT_DIG_TRIG</i>	(= 7)	Betriebsart 7
<i>MSH128_SW_EXT_AN_TRIG</i>	(= 8)	Betriebsart 8
<i>MSH128_TIMER_A_POS_EXT_AN_TRIG</i>	(= 10)	Betriebsart 10
<i>MSH128_TIMER_A_NEG_EXT_AN_TRIG</i>	(= 12)	Betriebsart 12
<i>MSH128_EXT_AN_TRIG_A_EXT_AN_TRIG</i>	(= 14)	Betriebsart 14
<i>MSH128_EXT_AN_TRIG_B_EXT_AN_TRIG</i>	(= 15)	Betriebsart 15

Die aktuelle Einstellung der Betriebsart kann durch einen Lesezugriff mit **mddx\_read\_channel\_byte** auf den Kanal ermittelt werden.

## Kanal zur Steuerung der Interrupt-Eingänge

Dieser Kanal ermöglicht es, einen der Interrupt-Eingänge IRQ-B, IRQ-C und IRQ-D zu aktivieren oder zu deaktivieren.

Ein Interrupt-Eingang wird durch einen Schreibzugriff mit **mddx\_write\_channel\_byte** auf den Kanal angewählt.

<i>MSH128_NONE</i>	= 0	Wählt keinen Interrupt-Eingang an
<i>MSH128_IRQ_B</i>	= 5	Wählt Interrupt-Eingang IRQ_B an
<i>MSH128_IRQ_C</i>	= 6	Wählt Interrupt-Eingang IRQ_C an
<i>MSH128_IRQ_D</i>	= 7	Wählt Interrupt-Eingang IRQ_D an

Welcher Interrupt-Eingang angewählt ist, kann durch einen Lesezugriff (mit **mddx\_read\_channel\_byte**) auf den Kanal ermittelt werden.

## Abtast-Trigger per Software

Ofs	Strukturelement	Datentyp	Bedeutung
0	<i>.usDevice</i>	USHORT	Device-Typ: <i>DEVICE_TRIG</i> = 0301h
2	<i>.usIndexFirst</i>	USHORT	Device-Index: 0 = Sample-Trigger 1 = Hold-Trigger
4	<i>.usIndexLast</i>	USHORT	= <i>.usIndexFirst</i>
6	<i>.usFlags</i>	USHORT	Mögliche Flags: Bit 0: <i>_CP_EXCLUSIVE</i> = 1 (exklusiv) Alle anderen Bits = 0.

Alle anderen Strukturelemente werden nicht ausgewertet.

Das Modul M-SH12-8 ermöglicht das gleichzeitige Abtasten aller analogen Eingänge AIN-0 bis AIN-7. Bei einem Schreibzugriff auf den Kanal 'Hold-Trigger' (mit **mddx\_trigger\_channel**) werden die Zustände der Eingänge AIN-0 bis AIN-7 zeitgleich in einem Latch auf dem Modul zwischengespeichert. Die im Latch gespeicherten Werte können mit einem entsprechend geöffneten I/O-Kanal (*DEVICE\_AIN\_SE*) gelesen werden. Mit einem Zugriff auf einen Kanal 'Sample-Trigger' werden die analogen Eingangskanäle in den Sample-Mode geschaltet.

## Analoge Eingänge

Ofs	Strukturelement	Datentyp	Bedeutung
0	<i>.usDevice</i>	USHORT	Device-Typ: <i>DEVICE_AIN_SE</i> = 0201h
2	<i>.usIndexFirst</i>	USHORT	Index des ersten Device (0 bis 7)
4	<i>.usIndexLast</i>	USHORT	Index des letzten Device (= <i>.usIndexFirst</i> oder 7)
6	<i>.usFlags</i>	USHORT	Mögliche Flags: Bit 0: <i>_CP_EXCLUSIVE</i> = 1 (exklusiv) Bit 2: <i>_CP_UNCORRECTED</i> = 4 (unkorrigiert) Alle anderen Bits = 0.
8	<i>.usReadMode</i>	USHORT	Wert lesen (direkt): <i>IO_MODE_DIRECT</i> = 1

Alle anderen Strukturelemente werden nicht ausgewertet.

## Zugriff

Das Modul M-SH12-8 besitzt 8 analoge Eingänge (AIN-0 bis AIN-7). Der Treiber ermöglicht zur Zeit entweder die Auswahl eines einzigen Device (*.usIndexFirst* = *.usIndexLast*) oder eine Gruppierung aus allen 8 Devices.

Das Lesen der Eingänge erfolgt durch einen Lesezugriff mit **mddx\_read\_channel\_short** oder **mddx\_read\_channel\_block** auf den Kanal. Der jeweilige Datentyp resultiert aus der Anzahl der ausgewählten Eingänge (1 Eingang: SHORT, 8 Eingänge: Block aus 16 Bytes).

Hinweis: Vor erstmaligem Lesen eines Analogeingangs muß Sample/Hold auf Sample geschaltet werden (z.B. mit *DEVICE\_CTRL*, Device-Index = 3). Nach dem Lesen wird Sample/Hold immer auf Sample geschaltet.

## Programmierung des analogen Ausgangs

Ofs	Strukturelement	Datentyp	Bedeutung
0	<i>.usDevice</i>	USHORT	Device-Typ: <i>DEVICE_AOUT</i> = 0203h
2	<i>.usIndexFirst</i>	USHORT	Device-Index (= 0)
4	<i>.usIndexLast</i>	USHORT	= <i>.usIndexFirst</i>
6	<i>.usFlags</i>	USHORT	Mögliche Flags: Bit 0: <i>_CP_EXCLUSIVE</i> = 1 (exklusiv) Bit 2: <i>_CP_UNCORRECTED</i> = 4 (unkorrigiert) Alle anderen Bits = 0.
8	<i>.usReadMode</i>	USHORT	Wert zurücklesen (aus RAM): <i>IO_MODE_RAM</i> = 4
10	<i>.usWriteMode</i>	USHORT	Wert ausgeben: <i>IO_MODE_DIRECT</i> = 1 (direkt)

Alle anderen Strukturelemente werden nicht ausgewertet.

## Zugriff

Das Setzen des Ausgangs erfolgt durch einen Schreibzugriff mit **mddx\_write\_channel\_short** auf den Kanal. Durch einen Lesezugriff mit **mddx\_read\_channel\_short** kann der aktuell ausgegebene Wert ermittelt werden.



# Hochsprachenbibliothek

Hinweise zum Einbinden der Bibliothek finden Sie in der Einführung im Abschnitt 'Hochsprachenbibliotheken'. Der Name der Bibliothek ist "m024\_lib" (= *libname*), Sie finden sie im Verzeichnis (*pathname*) **SPB\_MOD\BIB\M-SH12-8**. Bevor Sie Routinen aus der Bibliothek verwenden können, müssen Sie einmal die Prozedur **m024\_bib\_startup** aufrufen.

## **m024\_bib\_startup** **Initialisiere Modulbibliothek**

Pascal	PROCEDURE m024_bib_startup (micro_slot: byte);
C	void EXPORT m024_bib_startup (byte micro_slot);
Funktion	Diese Prozedur initialisiert die Modulbibliothek M024_LIB und das SPB-Modul M-SH12-8 auf dem angegebenen Steckplatz. Es werden u.a. die Initialisierungsdaten aus den EEPROMs aller Module M-SH12-8 übernommen, die sich auf der Basiskarte befinden.

## **m024\_set\_conf\_eeprom** **Setze EEPROM-Konfiguration**

Pascal	PROCEDURE m024_set_conf_eeprom (micro_slot: byte);
C	void EXPORT m024_set_conf_eeprom (byte micro_slot);
Funktion	Diese Prozedur setzt die Konfiguration so, wie sie im EEPROM des Moduls angegeben ist. Korrekturwerte und Default-Einstellungen nach dem Reset der Karte werden übernommen. Falls bereits mit der Prozedur m024_set_correct_values Werte festgelegt wurden, werden sie überschrieben.

**m024\_set\_mode****Setze Betriebsart**

Pascal	PROCEDURE m024_set_mode (micro_slot: byte; mode: byte);
C	void EXPORT m024_set_mode (byte micro_slot, byte mode);
Funktion	Diese Prozedur definiert die Betriebsart (Mode) des Moduls M-SH12-8. Sie legt fest, welcher Abtast-Trigger verwendet werden soll und wodurch ein Interrupt zur Basiskarte ausgelöst werden soll. Für die Verwendung der Bibliothek auf dem PC ist nur der Abtast-Trigger "Per Software" (mode = 0) sinnvoll, die anderen Betriebsarten sind für die Verwendung der Bibliothek innerhalb von Echtzeitprogrammen sinnvoll.
Parameter	<i>mode</i> : Betriebsart des Moduls. Den zu setzenden Wert entnehmen Sie bitte der Tabelle auf Seite 6-4.



**Hinweis** Diese Prozedur verändert die Interrupt-Einstellung. Prozedur m024\_define\_interrupt muß aufgerufen werden, wenn eine Interrupt-Leitung definiert werden soll.

**m024\_define\_interrupt****Wähle Interrupt-Leitung**

Pascal	PROCEDURE m024_define_interrupt (micro_slot: byte; irq: integer);
C	void EXPORT m024_define_interrupt (byte micro_slot, short irq);
Funktion	Diese Prozedur wählt die Interrupt-Leitung zur Basiskarte an. Diese Prozedur kann nur sinnvoll eingesetzt werden, wenn die Bibliothek innerhalb von Echtzeitprogrammen verwendet wird.
Parameter	<i>irq</i> : Gibt die anzuwählende Interrupt-Leitung an. Für den Parameter sind folgende Werte zulässig:

<b>Interrupt-Leitung</b>	<b>irq</b>
kein Interrupt	0
IRQ-B	5
IRQ-C	6
IRQ-D	7

**m024\_set\_trigger\_level****Setze Analogausgang**

Pascal	PROCEDURE m024_set_trigger_level (micro_slot: byte; level: integer);
C	void EXPORT m024_set_trigger_level (byte micro_slot, short level);
Funktion:	Diese Prozedur setzt den analogen Ausgang AOOUT-0 bzw. die analoge Trigger-Schwelle des Moduls.
Parameter	<i>level</i> : Digitalwert für Ausgangsspannung. Für <i>level</i> sind Werte zwischen -2048 (entspricht -5 V) und +2047 (entspricht +4,9975 V) zulässig.

**Hinweis** Diese Prozedur verändert die Interrupt-Einstellung. Prozedur m024\_define\_interrupt muß aufgerufen werden, wenn eine Interrupt-Leitung definiert werden soll.

M-SH12-8

**m024\_activate\_trigger****Aktiviere den Abtast-Trigger**

Pascal	PROCEDURE m024_activate_trigger (micro_slot: byte);
C	void EXPORT m024_activate_trigger (byte micro_slot);
Funktion	Diese Prozedur aktiviert den eingestellten Abtast-Trigger, der durch m024_set_mode definiert wurde. Danach wartet das Modul auf das Eintreffen des entsprechenden Abtast-Triggers (die Sample & Hold Verstärker gehen auf SAMPLE).

**m024\_soft\_trigger****Löse Abtastung per Software aus**

Pascal	PROCEDURE m024_soft_trigger (micro_slot: byte);
C	void EXPORT m024_soft_trigger (byte micro_slot);
Funktion	Diese Prozedur löst die Abtastung per Software aus (die Sample & Hold Verstärker gehen auf HOLD). Der Aufruf dieser Prozedur löst immer einen Abtast-Trigger aus, unabhängig davon, welche Betriebsart eingestellt ist. Das bedeutet, daß Abtast-Trigger per Software immer ausgeführt werden können. Allerdings muß vorher der Abtast-Trigger aktiviert worden sein, sonst hat der Aufruf dieser Prozedur keinen Effekt. Nach Aufruf dieser Prozedur können dann die Meßwerte ausgelesen werden. Dies muß innerhalb von 20 ms geschehen!

**m024\_hold\_status****Lies Status der S/H-Verstärker**

Pascal	PROCEDURE m024_hold_status (micro_slot: byte; var status_var: byte);
C	void EXPORT m024_hold_status (byte micro_slot, byte *status_var);
Funktion	Die Funktion liest den Status der Sample & Hold Verstärker.
Parameter	<i>status_var</i> : In dieser Variable wird der S/H-Status (0 = HOLD, 1 = SAMPLE) zurückgeliefert.

**m024\_set\_correct\_values****Setze Korrekturwerte**

Pascal	PROCEDURE m024_set_correct_values (micro_slot: byte; ain_nr: byte; cgain, coffset: integer);
C	void EXPORT m024_set_correct_values (byte micro_slot, byte ain_nr, short cgain, short coffset);
Funktion	Diese Prozedur setzt die Korrekturwerte Offset und Gain für einen Analogeingang.
Parameter	<i>ain_nr</i> : Nummer des Analogeingangs. Zulässig sind Werte von 0 bis 7.  <i>cgain</i> : Faktor für Verstärkungskorrektur. Erlaubt sind Werte zwi- schen -128 und + 127.  <i>coffset</i> : Wert für Offset-Korrektur. Zulässig sind Werte zwischen -128 und +127.

**m024\_get\_correct\_values****Lies Korrekturwerte**

Pascal	PROCEDURE m024_get_correct_values (micro_slot: byte; ain_nr: byte; var cgain_var, coffset_var: integer);
C	void EXPORT m024_get_correct_values (byte micro_slot, byte ain_nr, short *cgain_var, short *coffset_var);
Funktion	Die Funktion liest die in der Bibliothek gesetzten Korrekturwerte für Offset und Gain des angegebenen Analogeingangs.
Parameter	<p><i>ain_nr</i>: Nummer des Analogeingangs. Zulässig sind Werte von 0 bis 7.</p> <p><i>cgain_var</i>: Gibt den Wert des Faktors für Verstärkungskorrektur zurück (siehe m024_set_correct_values).</p> <p><i>coffset_var</i>: Gibt den Wert für Offset-Korrektur zurück (siehe m024_set_correct_values).</p>

**m024\_correct****Korrigiere Spannungswert**

Pascal	PROCEDURE m024_correct (microslot: byte; ain_nr: byte; var data_var: integer);
C	void EXPORT m024_correct (byte microslot, byte ain_nr, short *data_var);
Funktion	Diese Prozedur korrigiert einen Spannungswert entsprechend den für den angegebenen Analogeingang gesetzten Korrekturwerten.
Parameter	<p><i>ain_nr</i>: Nummer des Analogeingangs. Zulässige Werte sind 0 bis 7.</p> <p><i>data_var</i>: In dieser Variablen wird der zu korrigierende Wert übergeben. Wenn die Prozedur durchlaufen ist, enthält <i>data_var</i> den korrigierten Wert.</p>

**m024\_in****Lies einen Analogeingang**

Pascal	PROCEDURE m024_in (micro_slot: byte; ain_nr: byte; flag: byte; var data_var: integer);	
C	void EXPORT m024_in (byte micro_slot, byte ain_nr, byte flag, short *data_var);	
Funktion	Die Funktion liefert den Wert eines angegebenen Eingangskanals zurück. Dabei kann angegeben werden, ob bei dem zurückgelieferten Wert die Korrekturwerte berücksichtigt werden oder nicht.	
Parameter	<i>ain_nr:</i>	Nummer des Analogeingangs. Zulässige Werte sind 0 bis 7.
	<i>flag:</i>	Wenn flag = 1 ist, wird der gewandelte Wert einer Korrektur entsprechend der gesetzten Korrekturwerte unterzogen. Wenn flag = 0 ist, wird direkt der gewandelte Wert zurückgeliefert (bei Bedarf kann dieser Wert später noch mit m024_correct korrigiert werden).
	<i>data_var:</i>	Diese Variable liefert das Meßergebnis zurück (-2048 bis +2047)



**Hinweis** Vor Aufruf dieser Prozedur müssen die Prozeduren m024\_set\_mode und m024\_activate\_trigger aufgerufen werden.

# Programmierung mit I/O-Zugriffen

## Lokale I/O-Adressen

Funktion	R/W	Adresse	Data
Abtast-Trigger auslösen, Analogeingang AIN-n anwählen (für n sind die Werte 0 bis 7 zulässig), S/H = HOLD setzen	W8	MBA + 0	n
Abtast-Trigger aktivieren, ersten Analogeingang AIN-n (für n sind die Werte 0 bis 7 zulässig) anwählen, S/H = SAMPLE setzen	W8	MBA + 2	n
Mode <sup>1</sup> = m setzen	W8	MBA + 4	m
Interrupt-Leitung zur Basiskarte anwählen <sup>1</sup>	W8	MBA + 6	<sup>1</sup>
Setzen des D/A-Wandlers <sup>1</sup>	W8	MBA + 6	<sup>1</sup>
A/D-Wandlung starten und FF <sup>2</sup> = 0 setzen, danach nächsten Analogeingang anwählen	R8x	MBA + 0	
Wenn FF <sup>2</sup> = 0: Ergebnis A/D-Wandlung mit Autowait lesen (Bit 11 = MSB, Bit 0 = LSB, Bit 12 bis Bit 15 ungültig), FF <sup>2</sup> = 1 setzen	R16	MBA + 4	Bit 0 - Bit 11
Wenn FF <sup>2</sup> = 1: A/D-Wandlung starten und FF <sup>2</sup> = 0 setzen, danach nächsten Analogeingang anwählen	R16x	MBA + 4	
Status A/D-Wandler lesen (1 = busy)	R16	MBA + 2	Bit 15
Status FF lesen	R16	MBA + 8	Bit 15
Status HOLD lesen: 1 = Sample, 0 = Hold	R16	MBA + 0ah	Bit 15
Lies Status Analog-Trigger (1 = überschwellig, 0 = unterschwellig)	R16	MBA + 0ch	Bit 15

<sup>1</sup> Bitte die Kapitel "Setzen des D/A-Wandlers" (Seite 6-19), "Betriebsart (Mode) setzen" (Seite 6-33) und "Interrupt-Leitung zur Basiskarte anwählen" (Seite 6-34) beachten.

<sup>2</sup> FF ist ein modulinternes Flip-Flop.

## Initialisieren des Moduls

Funktion	R/W	Adresse	Data
D/A-Wandler-Logik deaktivieren und keine Interrupt-Leitung anwählen	W8	MBA + 6	0

## Setzen des D/A-Wandlers (für Analogausgang bzw. Schwelle)

Hierzu müssen die D/A-Wandler-Logik aktiviert und die 16 Bit einzeln (ein Schreibzugriff pro Bit) in den D/A-Wandler eingetragen werden. Das Eintragen beginnt bei Bit 15 und endet bei Bit 0. Bit 15 bis Bit 12 sind ohne Bedeutung, Bit 11 ist das höchstwertige, Bit 0 das niedrigstwertige Bit. Nachdem alle Bit gesetzt sind, muß nach einem Dummy-Zugriff die D/A-Wandler Logik wieder deaktiviert werden (siehe unten).

Schritt	Funktion	Zugriff	Adresse	Data
1	D/A-Wandler-Logik deaktivieren	W8	MBA + 6	= 4
2	Dummy-Zugriff	W8	MBA + 6	= 0
3	D/A-Wandler-Logik aktivieren und Bit 15 setzen <sup>1</sup>	W8	MBA + 6	= 8
4	3 Dummy-Zugriffe (Bit 14 bis 12 setzen)	W8	MBA + 6	= 8
5	Bit-11 bis Bit-0 setzen <sup>2</sup> (je Bit ein Zugriff, also 12 Zugriffe)	W8	MBA + 6	= 8 oder = 9
6	Ausgabe des neuen Wertes	W8	MBA + 6	= 8
7	D/A-Wandler-Logik deaktivieren	W8	MBA + 6	= 0

**!** Durch die Zugriffe zum Setzen des D/A-Wandlers wird die Interrupt-Anwahl verändert, danach ist kein Interrupt angewählt. Diese Einstellung muß deshalb anschließend wie gewünscht (noch einmal) vorgenommen werden.

<sup>1</sup> Der Status des A/D-Wandlers muß vor diesem Zugriff = 0 sein (entspricht: not busy).

<sup>2</sup> Jedes Bit wird einzeln gesetzt (für jedes Bit ist ein Schreibzugriff erforderlich). Um ein Bit = 0 zu setzen, muß als Data = 8 geschrieben werden, um es =1 zu setzen, muß als Data = 9 geschrieben werden.



### Betriebsart (Mode) setzen

Der Mode (siehe Seite 6-4) legt fest, welcher Abtast-Trigger verwendet wird und wodurch ein Interrupt zur Basiskarte ausgelöst werden kann. Zum Setzen des Modes sind 6 Schreibzugriffe erforderlich (siehe folgende Tabelle). Der Status des A/D-Wandlers muß vor dem 1. Schritt =0 sein (not busy). Nach Einstellung des Mode ist Analogeingang AIN-0 und kein Interrupt angewählt.

Schritt	Funktion	R/W	Adresse	Data
1	Mode einstellen, Beginn	W8	MBA+6	= 4
2	(Modulintern)	W8	MBA+6	= 0
3	(Modulintern)	W8	MBA+6	s.u.
4	(Modulintern)	W8	MBA+0	= 0
5	(Modulintern)	W8	MBA+6	= 0
6	Mode einstellen, Ende	W8	MBA+4	s.u.

Bei Schritt 3 und 6 müssen in Abhängigkeit des gewünschten Mode folgende Werte (Data) gesetzt werden:

Mode	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Data bei Schritt 3	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3
Data bei Schritt 6	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7

Die Einstellung des Mode verändert die Interrupt-Einstellung.

!

## Interrupt-Leitung zur Basiskarte anwählen

Bei den Modes 2 bis 7 und 10 bis 15 kann durch den Abtast-Trigger auch gleichzeitig ein Interrupt auf der Basiskarte ausgelöst werden. Drei Interrupt-Leitungen der Basiskarte stehen zur Auswahl: IRQ-B, IRQ-C und IRQ-D. Die aktive Flanke zur Interrupt-Auslösung muß, unabhängig von der Mode-Einstellung auf dem Modul, auf der Basiskarte (!) per Software eingestellt werden (siehe Funktion **ml8rt\_set\_int\_edge**). Soll z.B. bei Schwellenwertunterschreitung ein Interrupt auf der Basiskarte ausgelöst werden, muß die zugehörige Interrupt-Flanke auf 'fallend' gesetzt werden. Bei positiver Flankeneinstellung wird der Interrupt bei Überschreitung des Schwellenwertes ausgelöst.

Bei Mode 0, 1 und 8 wird die Abtastung per Software getriggert. Dadurch wird kein Interrupt zur Basiskarte ausgelöst, unabhängig davon, ob eine Interrupt-Leitung angewählt wurde. Der Interrupt-Mechanismus und die beiden externen Trigger-Eingänge des Moduls können dann aber für andere Zwecke verwendet werden, z.B. als Start-Trigger für eine Messung.

Wenn Timer A als Abtast-Trigger dient und gleichzeitig damit ein Interrupt ausgelöst werden soll, so kann hierfür direkt der Interrupt von Timer A der Basiskarte verwendet werden. Timer A sollte in Mode 2 des Timer-Bausteins betrieben werden. Auf dem Modul M-SH12-8 sollte Mode 2 oder 10 eingestellt werden. Wenn die damit verbundene Interrupt-Auslösung über den externen digitalen oder analogen Trigger nicht benötigt wird, wird bei der Interrupt-Einstellung "keine Interrupt-Leitung anwählen" eingestellt.

Die Interrupt-Einstellung erfordert zwei Schreibzugriffe:

Schritt	Funktion	R/W	Adresse	Data
1	Interrupt-Einstellungen vorbereiten	W8	MBA + 6	= 4
2	Interrupt-Leitungen anwählen	W8	MBA + 6	i
	i=0: keine			
	i=5: IRQ-B			
	i=6: IRQ-C			
	i=7: IRQ-D			

**!** Die Interrupt-Einstellung wird durch Setzen des Mode und durch Setzen des D/A-Wandlers verändert. Die Anwahl der gewünschten Interrupt-Leitung muß somit in diesem Falle nochmals erfolgen.

## A/D-Wandlung durchführen

Unabhängig vom gewählten Abtast-Trigger muß vor dem Start der Messung der gewünschte erste Analogeingang AIN-n angewählt und der Abtast-Trigger aktiviert werden (Schreibzugriff auf MBA+2). Wenn der Abtast-Trigger auftritt, werden alle 8 Sample/Hold Verstärker gleichzeitig auf HOLD (S/H=0) geschaltet. Damit sind alle 8 Analogsignale auf dem Modul zwischengespeichert und können nun gewandelt werden. Die Zeit zwischen Abtast-Trigger (Umschaltung auf HOLD) und A/D-Wandlung sollte 2 ms nicht überschreiten. Der Zustand HOLD kann nur durch erneutes Aktivieren des Abtast-Triggers wieder aufgehoben werden. Der Abtast-Trigger löst in der Regel auch einen Interrupt aus.

Wenn Timer A oder ein externer Eingang als Abtast-Trigger verwendet wird, dann kann in der Interrupt-Service-Routine die erste Wandlung sofort gestartet werden (Lesezugriff auf MBA+0).

Wenn die Abtastung per Software getriggert werden soll, kommt der Interrupt üblicherweise von einer anderen Quelle, z.B. von einem Timer. In der Interrupt-Service-Routine muß dann als erstes der Abtast-Trigger per Software ausgelöst werden (Schreibzugriff auf MBA+0). Danach kann die erste Wandlung durch einen Lesezugriff auf MBA+0 gestartet werden.

Der weitere Ablauf ist in allen Fällen gleich. Das Ergebnis der Wandlung wird durch zwei 16-Bit-Lesezugriffe auf MBA+4 abgeholt. Der erste Zugriff liefert das Ergebnis der Wandlung, der zweite Zugriff wählt automatisch den nächst höheren Analogeingang AIN-(n+1) an und liefert ungültige Daten. Ein Analogeingang kann aber auch jederzeit neu angewählt werden (Schreibzugriff auf MBA+0).

Wenn die erforderlichen Analogeingänge gewandelt wurden, muß wieder der gewünschte erste Analogeingang angewählt, auf SAMPLE geschaltet und der Abtast-Trigger neu aktiviert werden. Diese drei Aktionen werden mit einem Schreibzugriff auf MBA+2 erledigt. Damit kann die Interrupt-Service Routine beendet werden.

*Wenn Sie einen neuen Analogeingang anwählen (Schreibzugriff auf MBA + 0), müssen Sie immer darauf achten, daß gerade keine Wandlung läuft (prüfen mit Lesezugriff auf MBA + 2). Anderenfalls kann es zur Anwahl eines falschen Analogeingangs kommen.*

In der folgenden Tabelle ist ein kompletter Meßzyklus beschrieben. Bei Schritt 2 sind 4 Alternativen angegeben, entsprechend der möglichen Abtast-Triggerung. Nur ein Abtast-Trigger ist erforderlich. Das Ergebnis der A/D-Wandlung kann mit Auto-wait abgeholt werden. Schritt 5 entfällt dann.

<b>Schritt</b>	<b>Funktion</b>	<b>R/W</b>	<b>Adresse</b>	<b>Data</b>
1	Abtast-Trigger aktivieren (schaltet S/H= SAMPLE) und ersten Analogeingang n anwählen	W8	MBA + 2	n
2a	Externer digitaler Abtast-Trigger (schaltet S/H auf HOLD)	-	-	-
2b	Externer analoger Abtast-Trigger (schaltet S/H auf HOLD)	-	-	-
2c	Timer A als Abtast-Trigger (schaltet S/H auf HOLD)	-	-	-
2d	Abtastung per Software triggern (schaltet S/H = HOLD) und Analogeingang n anwählen	W8	MBA + 0	n
3	1 $\mu$ s Verzögerung (4 Zugriffe auf Adresse 3fh)	W8x	3fh	
4	Erste Wandlung starten. Wenn die Wandlung fertig ist, dann wird automatisch der nächst höhere Analogeingang angewählt.	R8x	MBA + 0	
5	Status lesen, warten bis Bit = 0 (kann bei Autowait entfallen)	R16	MBA + 2	Bit 15
6	Ergebnis lesen (immer mit Autowait)	R16	MBA + 4	Bit 0 - Bit 11
7	A/D-Wandlung starten und danach automatisch nächsten Analogeingang anwählen: Wenn noch nicht alle Eingänge gelesen sind, dann weiter bei Schritt 5	R16x	MBA + 4	
8	Status lesen, warten bis Bit = 0	R16	MBA + 2	Bit 15
9	Abtast-Trigger wieder aktivieren (schaltet S/H = SAMPLE) und ersten Analogeingang n anwählen	W8	MBA + 2	n