

AN041**Application Note zu Modul M-AD12-16**
M-AD12-16 als Ersatz für M-AD16-3

Autor: MH

Datei: AN041.DOC (25 Seiten)

Inhaltsverzeichnis:

1. Hardware	3
1.1. EEPROM-Konfiguration	3
1.2. Zusätzliche I/O-Adressen des M-AD12-16	10
1.3. Abgleich des Moduls	11
1.4. Beispiel	12
2. Software	15
2.1. Hochsprachenbibliothek	15
2.1.1. Einsatz der M-AD16-3 Bibliothek	17
2.1.2. Funktionen der Bibliothek	21

Allgemeines

Diese Application Note richtet sich an diejenigen, die bisher das Modul M-AD16-3 eingesetzt haben und nun das M-AD12-16 Modul einsetzen.

Das Modul M-AD12-16 kann alle Versionen des Moduls M-AD16-3 ohne Änderung der Software ersetzen. Grundsätzlich verhält sich das M-AD12-16 Modul nach dem Einschalten wie ein M-AD16-3 Modul. Lediglich die Eingangsbereichs-Einstellung und der Abgleich des Moduls werden anders vorgenommen.

Waren es bei dem M-AD16-3 diverse Jumper, die umgesteckt werden mußten, so erfolgt die Bereichseinstellung beim M-AD12-16 per Software. Sie können das Modul auf Ihre Basiskarte aufstecken, unabhängig davon, ob es sich dabei um eine MODULAR-4/Z80, /Z280 oder /486 Karte handelt, ohne vorher irgendwelche Einstellungen vornehmen zu müssen.

Bevor das Modul als Ersatz für M-AD16-3 benutzt werden kann, sind folgende Arbeitsschritte notwendig:

1. EEPROM-Daten des M-AD12-16 Moduls notieren.
2. EEPROM des M-AD12-16 Moduls mit den EEPROM-Daten des M-AD16-3 Moduls beschreiben.
3. Meßbereich einstellen.
4. Korrekturwerte einstellen.

Die Punkte 3 und 4 müssen jedesmal nach dem Einschalten des PCs durchgeführt werden. Wenn Sie über ein entsprechendes Betriebssystem verfügen, dann wird das Modul nach jedem Reset so konfiguriert, wie es im EEPROM angegeben ist. Die untenstehende Tabelle zeigt, ab welchem Betriebssystem die automatische Konfiguration enthalten ist. Sie können die Konfiguration aber auch mit SNW (siehe Seite 12) oder mit Ihrem eigenen PC-Programm durchführen (siehe Seite 15).

Karte	Version	Datum
MODULAR-4/Z80	Auf Anfrage	-
MODULAR-4/Z280	Auf Anfrage	-
MODULAR-4/486	3A.03R	26.04.96

1. Hardware

1.1. EEPROM-Konfiguration

Damit sich das M-AD12-16 Modul wie ein M-AD16-3 Modul verhält, ist es notwendig, den EEPROM-Inhalt des M-AD12-16 Moduls durch den EEPROM-Inhalt des M-AD16-3 Moduls zu ersetzen. Das EEPROM enthält wichtige Konfigurationsdaten wie Modultyp, Abgleichdaten, etc., die per Software ausgelesen werden können, um die Hardwarekonfiguration festzustellen.

Bevor der EEPROM-Inhalt ersetzt wird, sollten Sie sich den aktuellen Inhalt notieren, so daß Ihnen die EEPROM-Daten des M-AD12-16 Moduls nicht verlorengehen. Die Daten im EEPROM können Sie mit dem Hilfsprogramm SNW lesen und ändern.

Die werkseitige Einstellung eines M-AD16-3/8x Moduls können Sie als Beispiel der untenstehenden Tabelle entnehmen. Falls Sie schon ein M-AD16-3 Modul besitzen, können Sie auch den EEPROM-Inhalt dieses M-AD16-3 Moduls direkt ins EEPROM des M-AD12-16 Moduls übertragen.

Tragen Sie die Einstellungen der Steckbrücken in WORT-3 so ein, wie sie auf einem M-AD16-3 gesteckt sein müßten. Details dazu können Sie aus der Beschreibung zum M-AD16-3 entnehmen.

Wort	Binär		Hex.	Bedeutung (Kurzinfor)
0	0010 0011	0000 1100	230ch	Modultyp: M-AD16-3
1	0000 0000	0000 0001	0001h	Modulkonfiguration
2	0000 0011	1011 1111	03bfh	Hardware-Ausstattung
3	0011 0110	1100 0000	36c0h	Jumper-Einstellungen
5	0000 0000	0000 0000	0000h	Verstärkungsfaktoren
...
7	0000 0000	0000 0000	0000h	Verstärkungsfaktoren
10	0000 0000	0000 0000	0000h	Korrekturwerte
...
25	0000 0000	0000 0000	0000h	Korrekturwerte

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	0	1	1	0	0

WORT-0: Kennung

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

Modultyp: 0ch = M-AD16-3

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

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

WORT-1: Initialisierung (werks. Einst.)

geändert am:

von:

Init nach Hardreset: 0 = nein, 1 = ja

Reserviert

WORT-2: Ausstattung

Figure 1 illustrates the initial state of the 16-bit registers. The top row shows the initial values for the 16-bit registers, and the bottom row shows the initial values for the 8-bit registers. The registers are divided into two 8-bit halves by a vertical line.

WORT-2: Ausstattung (werks. Einst.)

geändert am:

von:

IC1 = INA110 (1 = ja)

Prog. Vorverstärkung (1 = ja)

SE-Eingänge (1 = möglich)

Differenzeingänge (1 = möglich)

Typ Sample & Hold (11 = High-Speed)

Typ A/D-Wandler ($10 = 8 \mu\text{s}$)R-Netzwerke ($11 = 100 \text{ k}\Omega$)

Reserviert

WORT-3: Jumper

The diagram illustrates the construction of a 16-bit carry-lookahead adder. It consists of two columns of logic blocks, each with 16 rows. The top row of each column shows the initial carry-in (0) and the first two propagate signals (1, 1). Subsequent rows show the propagation of carry signals through the chain, with carry values (0 or 1) appearing in specific blocks. The final row shows the complete 16-bit carry chain with carry values (0, 0) in the first two blocks.

WORT-3: Jumper (werks. Einst.)

geändert am:

von:

B1: Pin 1 und 2 (1 = verb.)

B1: Pin 3 und 5 (1 = verb.)

B1: Pin 4 und 6 (1 = verb.)

B1: Pin 5 und 7 (1 = verb.)

B1: Pin 6 und 8 (1 = verb.)

B2: Pin 1 und 2 (1 = verb.)

B2: Pin 2 und 3 (1 = verb.)

B2: Pin 4 und 5 (1 = verb.)

B2: Pin 5 und 6 (1 = verb.)

B2: Pin 6 und 7 (1 = verb.)

B3: Pin 2 (1 = 2-1, 0 = 2-3)

B6: Pin 4 und 5 (0 = verb.) bzw.

B3: Pin 5 (1 = 5-4, 0 = 5-6)

B5: Pin 2 (1 = 2-1, 0 = 2-3)

B6: Pin 2 (1 = 2-1, 0 = 2-3)

Reserviert

WORT-4 bis WORT-7: Verstärkungsfaktoren

Jeweils 4 Bit bestimmen den Verstärkungsfaktor für einen bestimmten Kanal:

0000 = x1, 0001 = x2, 0010 = x4, 0011 = x8, 0100 = x16

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 -7: Verstärkungsfaktoren
(werks. Einst.)

W-4 geändert am: von:

W-5 geändert am: von:

W-6 geändert am: von:

W-7 geändert am: von:

0	0	0	0												

Verstärkungsfaktoren				
WORT	4	5	6	7
Kanal	AIN-0	AIN-4	AIN-8	AIN-12
Kanal	AIN-1	AIN-5	AIN-9	AIN-13
Kanal	AIN-2	AIN-6	AIN-10	AIN-14
Kanal	AIN-3	AIN-7	AIN-11	AIN-15

WORT-8 und WORT-9: Reserviert (= 0 setzen)

WORT-10 bis WORT-25: Korrekturwerte für Kanal AIN-0 bis AIN-15

Jedem Kanal ist ein Wort zugeordnet: WORT-10 für Kanal AIN-0, WORT-11 für Kanal AIN-1, etc. Die unteren 8 Bit (Low-Byte) enthalten den Korrekturwert für Offset, die oberen 8 Bit (High-Byte) den für Verstärkung. Das Format (8-Bit-Zweierkomplement) ist für alle Werte gleich:

Binär	Hex.	Dezimal	Offset	Verstärkung	Faktor
1000 0000b	80h	-128	-128	$1 - 128/4096$	0,96875
1000 0001b	81h	-127	-127	$1 - 127/4096$	0,96899
1111 1111b	ffh	-1	-1	$1 - 1/4096$	0,99976
0000 0000b	00h	0	0	1	1
0000 0001b	01h	1	1	$1 + 1/4096$	1,00024
0111 1111b	7fh	127	127	$1 + 127/4096$	1,03101

WORT	Korrekturwerte für Kanal
10	AIN-0
11	AIN-1
12	AIN-2
13	AIN-3
14	AIN-4
15	AIN-5
16	AIN-6
17	AIN-7

WORT	Korrekturwerte für Kanal
18	AIN-8
19	AIN-9
20	AIN-10
21	AIN-11
22	AIN-12
23	AIN-13
24	AIN-14
25	AIN-15

[illegible]

WORT-10 bis -25: Korrekturwerte
(werks. Einst.)

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:

W-15 geändert am: von:

W-16 geändert am: von:

W-17 geändert am: von:

W-18 geändert am: von:

W-19 geändert am: von:

W-20 geändert am: von:

W-21 geändert am: von:

W-22 geändert am: von:

W-23 geändert am: von:

W-24 geändert am: von:

W-25 geändert am: von:

für Offset (+127 bis -128)

für Verstärkung (+127 bis -128)

1.2. Zusätzliche I/O-Adressen des M-AD12-16

Der Meßbereich der analogen Kanäle wird durch das Beschreiben einer I/O-Adresse eingestellt. Die Einstellung kann entweder mit einer kleinen Installationsdatei mittels SNW erfolgen oder über Ihre eigene Software, falls Sie die Möglichkeit haben, diese dahingehend zu ändern.

Die folgende Tabelle enthält die gegenüber dem M-AD16-3 Modul zusätzlichen I/O-Adressen, die zur Einstellung des Meßbereichs, der Korrekturwerte und zur Ermittlung des Modultyps benötigt werden.

"MBA" ist die sogenannte Modul-Basis-Adresse. Sie ist abhängig vom Steckplatz des Moduls:

Kartentyp	Steckplatz:	1	2	3	4	5	6	7	8	9	10
MODULAR-4/Z80 /Z280		80h	a0h	c0h	e0h	-	-	-	-	-	-
MODULAR-4/486		400h	500h	600h	700h	-	-	-	-	-	-
MODULAR-4/486 mit Modulextender		-	500h	600h	700h	420h	440h	460h	480h	4a0h	4c0h

Adresse	Zugr.	Funktion															
MBA+0dh	R8	Modulerkennung: Dieses Register liefert in Bit-0 eine "1". Das Modul M-AD16-3 liefert auf derselben Adresse immer eine "0" in Bit-0. Die Bit-1 bis Bit-7 enthalten die Gate-Array Revision. Zum Beispiel: Inhalt von MBA+0dh: 0001 1001b = 19h = 25 1 Bit nach rechts: 0000 1100b = 0ch = 12 = FPGA-Vers.															
MBA+0fh	W8	Eingangsbereich einstellen: ¹ <table> <tr> <th></th><th>massebezogen</th><th>Differenz</th></tr> <tr> <td>±5 V</td><td>02h</td><td>03h</td></tr> <tr> <td>±10 V</td><td>00h</td><td>01h</td></tr> <tr> <td>0 bis 10 V</td><td>06h</td><td>07h</td></tr> <tr> <td>0 bis -10 V</td><td>0eh</td><td>0fh</td></tr> </table>		massebezogen	Differenz	±5 V	02h	03h	±10 V	00h	01h	0 bis 10 V	06h	07h	0 bis -10 V	0eh	0fh
	massebezogen	Differenz															
±5 V	02h	03h															
±10 V	00h	01h															
0 bis 10 V	06h	07h															
0 bis -10 V	0eh	0fh															
MBA+05h	W8	Korrekturfaktor für Gain: ¹ Die Formeln zur Berechnung des Gain-Faktors finden Sie auf Seite 11. Die Werte dürfen zwischen 0 und 31 liegen.															
MBA+06h	W8	Korrekturfaktor für Offset: ¹ Die Formeln zur Berechnung des Offset-Faktors finden Sie auf Seite 11. Die Werte dürfen zwischen -32 und +31 liegen.															

¹ Der Inhalt dieses Registers ist nach dem Einschalten undefiniert!

Bei Modulen mit einem Gate-Array "Rev. 1" wird das Register zur Einstellung des Eingangsreichs und die Register zur Einstellung der Korrekturfaktoren nach einem Reset der Karte auf Null gesetzt! Das bedeutet, daß nach jedem Reset der Karte die Einstellungen neu vorgenommen werden müssen!



1.3. Abgleich des Moduls

Der Abgleich des Moduls erfolgt nicht mehr über Potentiometer wie beim M-AD16-3, sondern durch eine Logik im Gate-Array des Moduls. Es werden lediglich Korrekturfaktoren für Gain und Offset in die Register des Moduls eingetragen.

Die gewandelten Analogwerte (X_M) des M-AD12-16 Moduls werden einer linearen Korrektur gemäß der folgenden Formel unterzogen, worin *GAIN* und *OFFSET* für die Korrekturwerte stehen. X_{KOR} ist das korrigierte Ergebnis.

$$X_{KOR} = GAIN \cdot \frac{X_M}{2048} + OFFSET + X_M$$

$$(0 \leq X_M \leq +4095)$$

Die Korrekturfaktoren *GAIN* und *OFFSET* werden folgendermaßen bestimmt. Es werden nacheinander zwei Referenzspannungen angelegt ($SOLL_1$ und $SOLL_2$) und mit dem M-AD12-16 gemessen (IST_1 und IST_2). Aus diesen beiden Messungen berechnen sich die Korrekturfaktoren nach den untenstehenden Formeln. Beachten Sie bitte, daß die ermittelten Werte von *GAIN* innerhalb eines Bereichs von 0 bis +31 und die Werte von *OFFSET* innerhalb von -32 bis +31 liegen müssen.

$$GAIN = 2048 \cdot \frac{(SOLL_2 - SOLL_1) - (IST_2 - IST_1)}{IST_2 - IST_1}$$

$$OFFSET = SOLL_1 - IST_1 - GAIN \cdot \frac{IST_1}{2048}$$

$$(0 \leq SOLL_1 \leq +4095)$$

$$(0 \leq SOLL_2 \leq +4095)$$

$$(0 \leq IST_1 \leq +4095)$$

$$(0 \leq IST_2 \leq +4095)$$

1.4. Beispiel

Es soll ein M-AD16-3 Modul mit folgenden Eigenschaften ersetzt werden:

±5 Volt Eingangsspannungsbereich, 8 Differenzkanäle

Vorgehensweise:

1. EEPROM-Inhalte des M-AD12-16 Moduls notieren.
2. EEPROM-Werte des M-AD16-3 Moduls in das EEPROM des M-AD12-16 Moduls eintragen.
3. Werte für Meßbereich und Korrekturfaktoren ermitteln.

In das Register zur Einstellung des Meßbereichs (MBA+0fh) muß 03h eingetragen werden.

Folgende Korrekturwerte wurden für Gain und Offset ermittelt, bzw. wurden dem EEPROM des M-AD12-16 entnommen (siehe Beschreibung zum WORT-6 des EEPROM des M-AD12-16 Moduls):

Z. B.:

Gain = 19 = 13h

Offset = -4 = fch

Beachten Sie bitte, daß die Korrekturwerte auf jeden Fall nach dem Einschalten in das Modul eingetragen werden müssen! Um ein nicht korrigiertes Ergebnis zu erhalten, tragen Sie sowohl für Gain als auch für Offset den Wert 0 ein.

4. Installationsdatei erstellen, um die in Punkt 3. ermittelten Daten ins Modul einzutragen.

Die Installationsdatei einer MODULAR-4/486 Karte für SNW könnte folgendermaßen aussehen (wenn die MODULAR-4/486 die Basisadresse 0380h hat und das M-AD12-16 auf Steckplatz 1 steckt):

```

; Beispiel für MODULAR-4/486, Modul auf Steckplatz 1 (MBA = 400h)
; Beachten Sie bitte, daß alle Zahlen hexadezimal angegeben
; sind.
; Karte auf 0380h anwählen und Reset ausführen.
M8DEVICE 0380 TIMEOUT=10 RESET
;
;
; Meßbereich einstellen (Adresse MBA + 0fh):
; 02h = ±5 Volt, massebezogen
; 00h = ±10 Volt, massebezogen
; 06h = 0 bis 10 Volt, massebezogen
; 0eh = 0 bis -10 Volt, massebezogen
; 03h = ±5 Volt, Differenz-Eingänge
; 01h = ±10 Volt, Differenz-Eingänge
; 07h = 0 bis 10 Volt, Differenz-Eingänge
; 0fh = 0 bis -10 Volt, Differenz-Eingänge
;
; Hier soll ein Bereich von ±5 Volt, Differenz eingestellt
; werden.
M8CMD 23 03 0F 04 03
;
;      ┌───┐ ┌───┐ ┌───┐
;      │   │ │   │ │   │
;      └───┴───┴───┘ Datenbyte (Meßbereich)
;
;      └───┬───┬───┘ I/O-Adresse (Low, High), Steckplatz 1
;
;      └───┬───┬───┬───┘ Makrobefehl (I/O-Port beschreiben)
;
; Korrekturfaktor für Gain einstellen (Adresse MBA + 05h):
; Gain = 13h
M8CMD 23 03 05 04 13
;
; Korrekturfaktor für Offset einstellen (Adresse MBA + 06h):
; Offset = -4 (≡ fch im Zweierkomplement)
M8CMD 23 03 06 04 FC

```

Dasselbe für MODULAR-4/Z80 und /Z280

```

; Beispiel für MODULAR-4/Z80 und /Z280, Modul auf Steckplatz 1
; (MBA = 80h)
; Beachten Sie bitte, daß alle Zahlen hexadezimal angegeben
; sind.
; Karte auf 0380h anwählen und Reset ausführen.
M4DEVICE 0380 TIMEOUT=10 RESET
;
;
; Meßbereich einstellen (Adresse MBA + 0fh):
;   02h = ±5 Volt, massebezogen
;   00h = ±10 Volt, massebezogen
;   06h = 0 bis 10 Volt, massebezogen
;   0eh = 0 bis -10 Volt, massebezogen
;   03h = ±5 Volt, Differenz-Eingänge
;   01h = ±10 Volt, Differenz-Eingänge
;   07h = 0 bis 10 Volt, Differenz-Eingänge
;   0fh = 0 bis -10 Volt, Differenz-Eingänge
;
; Hier soll ein Bereich von ±5 Volt, Differenz eingestellt
; werden.
M4CMD 23 8f 03
;           |   |   |
;           |   |   | Datenbyte (Meßbereich)
;           |   |   | I/O-Adresse, Steckplatz 1
;           |   |   | Makrobefehl (I/O-Port beschreiben)
;
; Korrekturfaktor für Gain einstellen (Adresse MBA + 05h):
;   Gain = 13h
M4CMD 23 85 13
;
; Korrekturfaktor für Offset einstellen (Adresse MBA + 06h):
;   Offset = -4 (≡ fch im Zweierkomplement)
M4CMD 23 86 FC

```

5. Die unter Punkt 4. erstellte Installationsdatei muß nach jedem Einschalten des PCs mit SNW geladen werden. Durch das Hinzufügen der folgenden Zeile in die AUTO-EXEC.BAT-Datei läßt sich dieser Vorgang automatisieren.

SNW /i:"name".INS

6. Aufruf Ihres Anwenderprogrammes.

2. Software

2.1. Hochsprachenbibliothek

Diese Bibliothek für die Module M-AD16-3 und M-AD12-16 ist geeignet für die Basiskarten MODULAR-4/Z80, MODULAR-4/Z280 und MODULAR-4/486. Bei der MODULAR-4/486 Karte kann sie sowohl als Bibliothek für den PC (unter **DOS** und **Windows**) als auch als Bibliothek für Echtzeitprogramme, die lokal auf der Karte laufen, benutzt werden.

Die Bibliothek ist nahezu identisch mit der M-AD16-3 Bibliothek, jedoch mit dem Unterschied, daß sie um wenige Routinen erweitert wurde. Falls Sie die Möglichkeit haben, Ihre bisherigen Programme für das M-AD16-3 Modul neu zu compilieren, so sollten Sie Ihre alte M012_LIB Bibliothek durch die mitgelieferte M012_LIB Bibliothek ersetzen. Mit dieser neuen Bibliothek können sowohl die bisherigen M-AD16-3 Module als auch M-AD12-16 Module angesprochen werden.

Erkennt die Bibliothek bei der Initialisierung, daß es sich bei dem angesprochenen Modul um ein M-AD12-16 Modul handelt, so wird automatisch der in den EEPROMs eingetragene Eingangsspannungsbereich und Kanaltyp auf dem Modul eingestellt.

Beachten Sie jedoch, daß auf jeden Fall der EEPROM-Inhalt des M-AD12-16 Moduls durch den Inhalt eines M-AD16-3 Moduls ersetzt werden muß (siehe oben), damit die Bibliothek die entsprechenden Einstellungen korrekt vornehmen kann.

Die Bibliothek wurde um zwei Routinen erweitert:

m012_set_type_and_range	Diese Prozedur stellt einen Eingangsspannungsbereich und einen Kanaltyp auf dem Modul ein.
m012_set_hard_correct	Diese Prozedur stellt die hardwaremäßige Korrektur des M-AD12-16 Moduls ein.

Die M012_LIB Bibliothek ist für den Einsatz der Hochsprachen **Borland Pascal (Version 7.0)**, **Turbo-Pascal (Version 6.0 und 7.0)** und **Borland C++ (Version 3.1)** geeignet. Getestet wurde die Bibliothek mit Borland Pascal (Version 7.0) und Borland C++ (Version 3.1.) Sie ist vorwiegend dazu gedacht, das Ansprechen des Moduls zu demonstrieren. Sie ist nicht auf maximale Geschwindigkeit ausgelegt.

Es wird keine Überprüfung der Übergabeparameter vorgenommen.

Allgemeine Hinweise:

1. Der Parameter **micro_slot** wird in allen Routinen verwendet und gibt den Modulsteckplatz an. Bei der MODULAR-4/Z80, der MODULAR-4/Z280 und der MODULAR-4/486 Karte kann er die Werte 1 bis 4 annehmen, bei der MODULAR-4/486 Karte mit Modulextender auch die Werte 5 bis 10.
2. Vor Verwendung von Bibliotheksroutinen (sowohl aus der Basiskartenbibliothek als auch aus der M-AD12-16 Bibliothek) müssen folgende Routinen einmal aufgerufen werden:

MODULAR-4/Z80 und /Z280: **ml4_init**
MODULAR-4/486: **ml8_bib_startup** und
 ml8_reset oder **ml8_start**

Für on-board Programme, die auf der MODULAR-4/486 Karte laufen sollen, ist kein zusätzlicher Aufruf erforderlich.

3. Die Prozedur **m012_bib_startup** muß einmal vor Verwendung der anderen Routinen aus M012_LIB aufgerufen werden.
4. In der Beschreibung der Routinen gilt:

P: folgende Syntax gilt für Turbo-Pascal

C: folgende Syntax gilt für Borland C++

5. Die Angabe **EXPORT** bei der Beschreibung der Routinen für C bedeutet:

bei DOS-Programmen: far pascal
bei Windows-Programmen (DLL): _export far pascal

2.1.1. Einsatz der M-AD16-3 Bibliothek

MODULAR-4/486: Pascal-Programme unter DOS

Hierfür werden folgende Dateien mitgeliefert:

Unterverzeichnis	Dateiname	Kommentar
MAD163\PASCAL	M012_LIB.PAS	Source-Datei in PASCAL
MAD163\ML8\DOS\PASCAL	M012_LIB.OBJ	.OBJ-Datei
MAD163\ML8\DOS\PASCAL	M012_LIB.TP	Konfigurationsdatei

Verwenden Sie die Datei M012_LIB.PAS und geben Sie in der USES-Anweisung ML8BIB und M012_LIB an.

MODULAR-4/486: C-Programme unter DOS

Hierfür werden folgende Dateien mitgeliefert:

Unterverzeichnis	Dateiname	Kommentar
MAD163\C	M012_LIB.C	Source-Datei in C
MAD163\C	M012_LIB.H	Header-Datei für C
MAD163\ML8\DOS\C	M012_LIB.PRJ	Projektdatei für C

Binden Sie die Include-Dateien ML8BIB.H und M012_LIB.H in Ihr C-Programm ein (#include "Dateiname"). Linken Sie ML8MAIN.LIB und M012_LIB.OBJ zu Ihrem C-Programm dazu.

MODULAR-4/486: Pascal-Programme unter Windows (DLL)

Hierfür werden folgende Dateien mitgeliefert:

Unterverzeichnis	Dateiname	Kommentar
MAD163\PASCAL	M012_LIB.PAS	Source-Datei PASCAL
MAD163\ML8\WINDOWS\PASCAL	M012_LIB.DLL	.DLL-Datei
MAD163\ML8\WINDOWS\PASCAL	M012_LIB.LIB	.LIB-Datei
MAD163\ML8\WINDOWS\PASCAL	M012_LIB.DEF	.DEF-Datei
MAD163\ML8\WINDOWS\PASCAL	M012_LIB.TPW	.TPW-Datei
MAD163\ML8\WINDOWS\PASCAL	M012_LIB.TP	Konfigurationsdatei

Geben Sie in der USES-Anweisung WML8BIB und M012_LIB an. Die Dateien WML8BIB.DLL, WML8BIB.TPW, M012_LIB.DLL und M012_LIB.TPW müssen im Default-Verzeichnis sein. Die Import-Unit M012_LIB.TPW wurde aus der Datei M012_LIB.PAS erzeugt.

MODULAR-4/486: C-Programme unter Windows (DLL)

Hierfür werden folgende Dateien mitgeliefert:

Unterverzeichnis	Dateiname	Kommentar
MAD163\C	M012_LIB.C	Source-Datei in C
MAD163\C	M012_LIB.H	Header-Datei für C
MAD163\ML8\WINDOWS\C	M012_LIB.PRJ	Projektdatei
MAD163\ML8\WINDOWS\C	M012_LIB.DLL	.DLL-Datei
MAD163\ML8\WINDOWS\C	M012_LIB.LIB	.LIB-Datei
MAD163\ML8\WINDOWS\C	M012_LIB.DEF	.DEF-Datei

Binden Sie die Include-Dateien ML8BIB.H und M012_LIB.H in Ihr C-Programm ein (#include "Dateiname"). Linken Sie WML8BIB.LIB und M012_LIB.LIB zu Ihrem C-Programm dazu. Die Dateien WML8BIB.DLL und M012_LIB.DLL müssen zur Laufzeit im Default-Verzeichnis sein.

MODULAR-4/486: Lokale Pascal-Programme auf der Karte

Hierfür werden folgende Dateien mitgeliefert:

Unterverzeichnis	Dateiname	Kommentar
MAD163\PASCAL	M012_LIB.PAS	Source-Datei in PASCAL
MAD163\ML8\RT\PASCAL	M012_LIB.OBJ	.OBJ-Datei für PASCAL
MAD163\ML8\RT\PASCAL	M012_LIB.TP	Konfigurationsdatei

Verwenden Sie die Datei M012_LIB.PAS und geben Sie in der USES-Anweisung ML8RTBIB und M012_LIB an.

MODULAR-4/486: Lokale C-Programme auf der Karte

Hierfür werden folgende Dateien mitgeliefert:

Unterverzeichnis	Dateiname	Kommentar
MAD163\C	M012_LIB.C	Source-Datei in C
MAD163\C	M012_LIB.H	Header-Datei für C
MAD163\ML8\RT\C	M012_LIB.PRJ	Projektdatei für C

Binden Sie die Include-Dateien ML8RTBIB.H und M012_LIB.H in Ihr C-Programm ein (#include "Dateiname"). Linken Sie ML8RTBIB.OBJ und M012_LIB.OBJ zu Ihrem C-Programm dazu.

Weitere Hinweise zu den Hochsprachenbibliotheken finden Sie im Benutzerhandbuch zur MODULAR-4 Karte in den Kapiteln "PC-Hochsprachenbibliotheken" und "Echtzeitbibliotheken".

MODULAR-4/Z80 und /Z280: Pascal-Programme unter DOS

Hierfür werden folgende Dateien mitgeliefert:

Unterverzeichnis	Dateiname	Kommentar
MAD163\PASCAL	M012_LIB.PAS	Source-Datei in PASCAL
MAD163\ML4\DOS\PASCAL	M012_LIB.OBJ	.OBJ-Datei
MAD163\ML4\DOS\PASCAL	M012_LIB.TP	Konfigurationsdatei

Verwenden Sie die Datei M012_LIB.PAS und geben Sie in der USES-Anweisung ML4BIB, ML4DEB und M012_LIB an.

MODULAR-4/Z80 und /Z280: C-Programme unter DOS

Hierfür werden folgende Dateien mitgeliefert:

Unterverzeichnis	Dateiname	Kommentar
MAD163\C	M012_LIB.C	Source-Datei in C
MAD163\C	M012_LIB.H	Header-Datei für C
MAD163\ML4\DOS\C	M012_LIB.PRJ	Projektdatei für C

Binden Sie die Include-Dateien ML4BIB.H und M012_LIB.H in Ihr C-Programm ein (#include "Dateiname"). Linken Sie ML4MAIN.LIB und M012_LIB.OBJ zu Ihrem C-Programm dazu.

MODULAR-4/Z80 und /Z280: Pasacal-Programme unter Windows (DLL)

Hierfür werden folgende Dateien mitgeliefert:

Unterverzeichnis	Dateiname	Kommentar
MAD163\PASCAL	M012_LIB.PAS	Source-Datei PASCAL
MAD163\ML4\WINDOWS\PASCAL	M012_LIB.DLL	.DLL-Datei
MAD163\ML4\WINDOWS\PASCAL	M012_LIB.LIB	.LIB-Datei
MAD163\ML4\WINDOWS\PASCAL	M012_LIB.DEF	.DEF-Datei
MAD163\ML4\WINDOWS\PASCAL	M012_LIB.TPW	.TPW-Datei
MAD163\ML4\WINDOWS\PASCAL	M012_LIB.TP	Konfigurationsdatei

Geben Sie in der USES-Anweisung DML4BIB, DML4DEB und M012_LIB an. Die Dateien DML4BIB.DLL, WML4BIB.TPW, DML4DEB.DLL, WML4DEB.TPW, M012_LIB.DLL und M012_LIB.TPW müssen im Default-Verzeichnis sein. Die Import-Unit M012_LIB.TPW wurde aus der Datei M012_LIB.PAS erzeugt.

MODULAR-4/Z80 und /Z280: C-Programme unter Windows (DLL)

Hierfür werden folgende Dateien mitgeliefert:

Unterverzeichnis	Dateiname	Kommentar
MAD163\C	M012_LIB.C	Source-Datei in C
MAD163\C	M012_LIB.H	Header-Datei für C
MAD163\ML4\WINDOWS\C	M012_LIB.PRJ	Projektdatei
MAD163\ML4\WINDOWS\C	M012_LIB.DLL	.DLL-Datei
MAD163\ML4\WINDOWS\C	M012_LIB.LIB	.LIB-Datei
MAD163\ML4\WINDOWS\C	M012_LIB.DEF	.DEF-Datei

Binden Sie die Include-Dateien WML4BIB.H, WML4DEB.H und M012_LIB.H in Ihr C-Programm ein (`#include "Dateiname"`). Linken Sie WML4BIB.LIB, WML4DEB.LIB und M012_LIB.LIB zu Ihrem C-Programm dazu. Die Dateien DML4BIB.DLL, DML4DEB.DLL und M012_LIB.DLL müssen zur Laufzeit im Default-Verzeichnis sein.

2.1.2. Funktionen der Bibliothek

Initialisiere Modulbibliothek

P: PROCEDURE **m012_bib_startup (micro_slot: byte);**

C: void EXPORT **m012_bib_startup (byte micro_slot);**

Diese Prozedur initialisiert die Modulbibliothek M012_LIB. Es werden u. a. die Initialisierungsdaten aus den EEPROMs aller Module M-AD16-3 übernommen, die sich auf der Basiskarte befinden. Wird statt eines M-AD16-3 Moduls ein M-AD12-16 Modul verwendet, so stellt diese Prozedur zusätzlich den Kanaltyp (massebezogen oder Differenzkanal) und den Eingangsspannungsbereich ein (± 5 Volt bzw. 0 bis 10 Volt oder ± 10 Volt), die im EEPROM angegeben sind.

Setze Konfiguration wie im EEPROM angegeben

P: PROCEDURE **m012_set_conf_eeprom (micro_slot: byte);**

C: void EXPORT **m012_set_conf_eeprom (byte micro_slot);**

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 "m012_set_correct_values" oder "m012_set_gain" Werte festgelegt wurden, werden sie überschrieben.

Verstärkungsfaktor für einen Kanal setzen

P: PROCEDURE **m012_set_gain** (micro_slot: byte; channel: byte; gain: integer);

C: void EXPORT **m012_set_gain** (byte micro_slot, byte channel, short gain);

Diese Prozedur setzt den Verstärkungsfaktor **gain** (0 = Faktor 1, 1 = Faktor 2, 2 = Faktor 4, 3 = Faktor 8, 4 = Faktor 16) für den angegebenen Kanal **channel** (0 bis 15).

Der Verstärkungsfaktor wird nur vorübergehend in der Bibliothek gespeichert und nicht im EEPROM abgelegt. Bei Neustart eines Programmes ist er gelöscht, d.h. = 0 gesetzt.

Verstärkungsfaktor für einen Kanal lesen

P: PROCEDURE **m012_get_gain** (micro_slot: byte; channel: byte;
var gain: integer);

C: void EXPORT **m012_get_gain** (byte micro_slot, byte channel, short *gain);

Diese Prozedur liest den Verstärkungsfaktor **gain** (0 = Faktor 1, 1 = Faktor 2, 2 = Faktor 4, 3 = Faktor 8, 4 = Faktor 16) für den angegebenen Kanal **channel** (0 bis 15).

Korrekturwerte für einen Kanal setzen

P: PROCEDURE **m012_set_correct_values** (micro_slot: byte; channel: byte; corgain,
coroffset: integer);

C: void EXPORT **m012_set_correct_values** (byte micro_slot, byte channel,
short corgain, short coroffset);

Diese Prozedur setzt die Korrekturwerte Offset (**coroffset**) und Gain (**corgain**) für den angegebenen Kanal **channel** (0 bis 15).

Mit dem Parameter **corgain** kann die Verstärkung des angegebenen Kanals korrigiert werden. Es sind nur ganze Zahlen erlaubt. Der Korrekturbereich liegt zwischen dem Faktor 0,96875 und 1,03101, z. B.:

corgain	Verstärkungsfaktor
-128	$1 - 128/4096 = 0,96875$
-127	$1 - 127/4096 = 0,96899$
...	...
...	...
-1	$1 - 1/4096 = 0,99976$
0	$1 = 1$
1	$1 + 1/4096 = 1,00024$
...	...
...	...
126	$1 + 126/4096 = 1,03076$
127	$1 + 127/4096 = 1,03101$

Mit dem Parameter **coroffset** kann der Offset des angegebenen Kanals korrigiert werden. Der Korrekturbereich liegt zwischen -128 und +127. Es sind nur ganze Zahlen erlaubt.

Die Korrektur eines Wertes wird nach folgender Formel vorgenommen:

$$y = (x + coroffset) \cdot \left(1 + \frac{corgain}{4096}\right)$$

wobei x der am A/D-Wandler gemessene Wert und y der korrigierte Wert ist.

Die Korrekturwerte werden nur vorübergehend in der Bibliothek gespeichert und nicht im EEPROM abgelegt. Bei Neustart eines Programmes sind sie gelöscht, d. h. = 0 gesetzt.

Korrekturwerte für einen Kanal lesen

P: PROCEDURE **m012_get_correct_values** (micro_slot: byte; channel: byte;
var corgain, coroffset: integer);

C: void EXPORT **m012_get_correct_values** (byte micro_slot, byte channel,
short *corgain, short *coroffset);

Nach Aufruf dieser Prozedur enthalten die Variablen **coroffset** und **corgain** die für den angegebenen Kanal **channel** (0 bis 15) in der Bibliothek gesetzten Werte.

Spannungswert korrigieren

P: PROCEDURE **m012_correct** (microslot: byte; channel: byte;
var data_var: integer);

C: void EXPORT **m012_correct** (byte microslot, byte channel, short *data_var);

Diese Prozedur korrigiert einen Spannungswert, der in der Variablen **data_var** übergeben wird, entsprechend den für den Kanal **channel** (0 bis 15) gesetzten Korrekturwerten. Nach Aufruf der Prozedur enthält die Variable **data_var** den korrigierten Wert.

Spannung eines Moduleingangs messen

P: PROCEDURE **m012_in** (micro_slot: byte; channel: byte; flag: byte; settle_time: word; var data_var: integer);

C: void EXPORT **m012_in** (byte micro_slot, byte channel, byte flag, ushort settle_time, short *data_var);

Nach Aufruf dieser Prozedur enthält die Variable **data_var** den Wert des Analogeingangskanals, der in **channel** (0 bis 15) angegeben ist. Der Wertebereich ergibt sich aus der Konfiguration des Kanals, z. B. entspricht ein Spannungsbereich von ± 5 Volt (= bipolar) einem Wertebereich von -2048 bis +2047 und ein Spannungsbereich von 0 bis 10 Volt (= unipolar) einem Wertebereich von 0 bis 4095. Im Parameter **settle_time** muß die Zeit angegeben werden, die der Multiplexer bzw. programmierbare Vorverstärker zum Einschwingen benötigt. Der Parameter **flag** gibt an, ob der gewandelte Wert korrigiert werden soll (flag = 1) oder nicht (flag = 0).

Soll der gemessene Wert hinterher korrigiert werden, muß noch die Prozedur **m012_correct** aufgerufen werden, die dann die für den angegebenen Kanal definierten Korrekturwerte berücksichtigt.

Meßbereich einstellen

P: PROCEDURE **m012_set_type_and_range** (micro_slot: byte; typ: byte; range: byte);

C: void EXPORT **m012_set_type_and_range** (byte micro_slot, byte typ, byte range);

Diese Prozedur setzt die Kanaleigenschaften des M-AD12-16 Moduls, die bei dem M-AD16-3 Modul per Jumper eingestellt wurden. **typ** spezifiziert den gewünschten Kanaltyp. Die folgenden Tabellen enthalten nähere Informationen zur Einstellung des Kanaltyps und des Eingangsbereichs. Die dort angegebenen Konstanten finden Sie in der Header-Datei bzw. in der Pascal-Unit.

Wert	Konstante	Bedeutung
0	M012_SINGLE_ENDED	massebezogener Eingang
1	M012_DIFFERENTIAL	Differenzeingang

range spezifiziert den gewünschten Eingangsbereich des angegebenen Kanals:

Wert	Konstante	Bedeutung
4	M012_UPP_10V	Unipolar von 0 bis 10 V
2	M012_BP_5V	Bipolar ± 5 V
0	M012_BP_10V	Bipolar ± 10 V

Eingestellten Meßbereich ermitteln

P: PROCEDURE **m012_get_type_and_range** (micro_slot: byte;
VAR typ: byte; VAR range: byte);

C: void **EXPORT m012_get_type_and_range** (byte micro_slot,
byte *typ, byte *range);

Diese Prozedur ermittelt die eingestellten Kanaleigenschaften des M-AD12-16 Moduls (siehe oben **m012_set_type_and_range**).

Hardware-Abgleichwerte einstellen

P: PROCEDURE **m012_set_hard_correct** (micro_slot: byte; hardgain: byte;
hardoffset: byte);

C: void **EXPORT m012_set_hard_correct** (byte micro_slot, byte hardgain,
byte hardoffset);

Diese Prozedur setzt die Korrekturwerte **hardgain** und **hardoffset** für den angegebenen Eingangsbereich. Die Korrektur wird hardwaremäßig in dem Gate-Array des Moduls durchgeführt und ersetzt die Abgleichpotentiometer des M-AD16-3 Moduls.

Die Korrekturwerte werden nur vorübergehend in der Bibliothek gespeichert und nicht im EEPROM abgelegt. Bei Neustart des Programmes sind sie gelöscht, d. h. = 0 gesetzt.