13. Programmierung der Ein- und Ausgänge

Die Ein- und Ausgänge der Multi-LAB/2 Karte werden in der Regel unter Verwendung von Bibliotheken programmiert, vom PC aus zum Beispiel mit den Funktionen von ML2BIB. Wenn allerdings für das von Ihnen verwendete Betriebssystem oder die Programmiersprache keine Bibliotheken vorliegen, oder wenn Sie sehr schnelle Echtzeitprogramme für die Karte schreiben wollen, müssen Sie die Einheiten der Multi-LAB/2 direkt programmieren.

Die Programmierung geschieht über Zugriffe auf I/O-Adressen des Prozessors auf der Multi-LAB/2 Karte. Es gibt 8 Bit breite und 16 Bit breite Ports, die gelesen oder geschrieben werden können. Daraus ergeben sich vier Funktionen, die entsprechend den folgenden Tabellen in verschiedenen Plattformen verwendet werden können.

Neben diesen vier Befehlen wird bei Multi-LAB/2i und /2h noch ein Befehl verwendet, um Abgleichdaten aus dem EEPROM zu lesen.

8 Bit breiten Port lesen

Im folgenden abgekürzt durch: **data = in8_port(adr)**
Verwendet für Ports mit dem Zugriff: **R8**

PC-Programm	Pascal	data := ml2_in8_port(adr)	data: BYTE adr: WORD
	C	data = ml2_in8_port(adr)	data: char adr: ushort
	Makrobefehl	Senden: 27h 22h (adr)L (adr)H Empfangen: 27h (data)	
Lokales Echtzeit-programm	Pascal	data := port[adr]	data: BYTE adr: WORD
	C	data = inportb(adr)	data: char adr: ushort
	Assembler	IN AL, DX oder IN AL, port	

16 Bit breiten Port lesen

Im folgenden abgekürzt durch: **data = in16_port(adr)**
Verwendet für Ports mit dem Zugriff: **R16**

PC-Programm	Pascal	data := ml2_in16_port(adr)	data: WORD adr: WORD
	C	data = ml2_in16_port(adr)	data: ushort adr: ushort
	Makrobefehl	Senden: 27h 32h (adr)L (adr)H Empfangen: 27h (data)L (data)H	
Lokales Echtzeit-programm	Pascal	data := portw[adr]	data: WORD adr: WORD
	C	data = inport(adr)	data: ushort adr: ushort
	Assembler	IN AX, DX oder IN AX, port	

8 Bit breiten Port schreiben

Im folgenden abgekürzt durch: **out8_port(adr,data)**
Verwendet für Ports mit dem Zugriff: **W8**

PC-Programm	Pascal	ml2_out8_port(adr,data)	data: BYTE adr: WORD
	C	ml2_out8_port(adr,data)	data: char adr: ushort
	Makrobefehl	23h 03h (adr)L (adr)H (data)	
Lokales Echtzeit-programm	Pascal	port[adr] := data	data: BYTE adr: WORD
	C	outportb(adr,data)	data: char adr: ushort
	Assembler	OUT DX, AL oder OUT port, AL	

16 Bit breiten Port schreiben

Im folgenden abgekürzt mit: **out16_port(adr,data)**
Verwendet für Ports mit dem Zugriff: **W16**

PC-Programm	Pascal	ml2_out16_port(adr,data)	data: WORD adr: WORD
	C	ml2_out16_port(adr,data)	data: ushort adr: ushort
	Makrobefehl	23h 04h (adr)L (adr)H (data)L (data)H	
Lokales Echtzeit-programm	Pascal	portw[adr] := data	data: WORD adr: WORD
	C	outport(adr,data)	data: ushort adr: ushort
	Assembler	OUT DX, AX oder OUT port, AX	

EEPROM-Wort lesen

Im folgenden abgekürzt durch: **data = read_eeprom(nr)**

PC-Programm	Pascal	data := ml2_read_eeprom_copy(nr)	data: WORD nr: BYTE
	C	data = ml2_read_eeprom_copy(nr)	data: ushort nr: byte
	Makrobefehl	Mit Makrobefehl 4ch (Parameter lesen) zuerst Parameterwort 66h lesen (=*start*), dann Parameterwort *start*+nr lesen (=data)	
Lokales Echtzeit-programm	Pascal	data := ml2rt_read_eeprom_copy(nr)	data: WORD nr: BYTE
	C	data = ml2rt_read_eeprom_copy(nr)	data: ushort nr: byte
	Assembler	Mit Funktionsaufruf 13 (READ_PAR_WORD) zuerst Parameterwort 66h lesen (=*start*), dann Parameterwort *start*+nr lesen (=data)	

13.1. Programmieren der digitalen Ein- und Ausgänge

Die Programmierung der digitalen Ein- und Ausgänge erfolgt über drei I/O-Ports, die in der folgenden Tabelle zusammengestellt sind (ausführlichere Informationen finden Sie in Anhang C).

Adresse	Zugriff	Funktion
01h	W8	Moderegister (für digitale Ein- und Ausgänge) setzen
00h	W8	Ausgangszustände OP-0 bis OP-7 setzen
02h	W8x	Eingangszustände IP-0 bis IP-15 latchen
00h	R16	Eingangszustände IP-0 bis IP-15 aus Latch lesen

Das Moderegister enthält Konfigurationsbits für die Ein- und Ausgänge. Beim Ändern der Konfiguration für die Eingänge muß also darauf geachtet werden, daß die Konfiguration der Ausgänge nicht verändert wird und umgekehrt. Am besten wird in einer globalen Variablen (im folgenden mit ***mode*** bezeichnet) der aktuelle Inhalt von Register 01h gespeichert, so daß gezielt einzelne Bit verändert werden können, ohne andere zu beeinflussen. ***mode*** sollte zum Programmstart mit Null vorbelegt werden.

13.1.1. Lesen der digitalen Eingänge

Die Eingänge werden immer aus einem Latch gelesen. Sie können per Software oder durch eine Flanke an IP-0 gelatcht werden.

1. Mode einstellen

In Register 01h wird eingestellt, ob die Eingänge durch IP-0 gelatcht werden können und mit welcher Flanke das geschieht.

Mögliche Einstellungen:

mode = *mode* AND fdh	Latchen nur per Software (Bit-1 = 0)
mode = *mode* OR 02h	Latchen auch mit IP-0 (Bit-1 = 1)

Einstellen der Flanke, wenn mit IP-0 gelatcht wird:

mode = *mode* AND fbh	Latchen mit positiver Flanke (Bit-2 = 0)
mode = *mode* OR 04h	Latchen mit negativer Flanke (Bit-2 = 1)
out8_port(01h,*mode*)	*mode* in Register eintragen

2. Eingänge latchen

Dieser Schritt kann entfallen, wenn mit IP-0 gelatcht wird.

out8_port(02h,00)	Eingänge werden ins Latch übernommen

3. Latch auslesen

data = in16_port(00h)	*data* enthält die gespeicherten Eingangszustände aus dem Latch

data ist ein 16 Bit-Wert. Bit-0 steht für Eingang IP-0, Bit-1 für Eingang IP-1, usw.

Um neue Werte einzulesen, müssen Schritt 2 und Schritt 3 wiederholt werden. Schritt 2 kann entfallen, wenn mit IP-0 gelatcht wird.

13.1.2. Setzen der digitalen Ausgänge

Die digitalen Ausgänge können über I/O-Port 01h alle gemeinsam aktiviert oder hochohmig geschaltet werden. Nach einem Reset sind sie hochohmig. Sie sollten erst aktiv geschaltet werden, wenn das Ausgangsregister I/O-Port 00h beschrieben wurde.

1. Ausgangswerte setzen

Alle acht Ausgänge sind in einem Byte codiert. Bit-0 steht für OP-0, Bit-1 für OP-1, usw. Wenn nur ein Ausgang geändert werden soll, muß also dafür gesorgt werden, daß alle anderen Bits unverändert bleiben.

out8_port(00h,*data)*	*data* in Ausgangsregister übernehmen

Wenn Bit-0 in I/O-Port 01h auf eins gesetzt ist, liegen die neuen Zustände sofort an den Ausgängen an.

2. Ausgänge aktivieren

Beim Setzen von Bit-0 in I/O-Port 01h dürfen die anderen Bits nicht verändert werden (s.o.).

mode = *mode* OR 01h	Bit-0 = 1 setzen
out8_port(01h,*mode*)	Ausgänge aktivieren

3. Ausgänge verändern

Die Ausgänge können jederzeit durch Schreiben eines neuen Datenbyte geändert werden.

out8_port(00h,*data)*	*data* in Ausgangsregister übernehmen, (wird gleich an die Ausgänge angelegt)

4. Ausgänge deaktivieren (hochohmig schalten)

Die Ausgänge können jederzeit deaktiviert werden. Das Ausgangsregister I/O-Port 00h wird davon nicht beeinflußt. Wenn die Ausgänge wieder aktiviert werden, liegen also wieder dieselben Ausgangszustände an.

mode = *mode* AND feh	Bit-0 = 0 setzen
out8_port(01h,*mode*)	Ausgänge deaktivieren

13.2. Programmieren der Zähler

Die Multi-LAB/2 Karte besitzt zwei Zähler. Zähler-A kann nur als Zeitgeber (Timer) verwendet werden, Zähler-B kann zusätzlich als Zähler mit Inkrementalgeber-interface, zur Ereigniszählung und zur Frequenz-, Pulsbreiten- und Periodendauer-messung verwendet werden. Die Zähler werden über die folgenden sechs I/O-Ports programmiert (ausführlichere Informationen in Anhang C):

Adresse	Zugriff	Funktion
82h	W8	Basistakt (beide Zähler)
07h	W8	Betriebsart (beide Zähler)
0ah	W8	Meßeingang wählen (Zähler-B)
0ch	W16	Zähler-A Data laden, 12-Bit
0ah	W16	Zähler-B Data laden (nur im Timermode), 16-Bit
08h	W8x	Zähler-B latchen (nur im Zähler-/Inkrementalgebermode)
04h	R16	Zähler-B Latch auslesen

Der Basistakt (I/O-Port 82h) für beide Timer wird nach einem Reset vom Betriebssystem so gesetzt, daß der Basistakt immer 500 kHz (≡ 2 µs) beträgt. I/O-Port 82h muß in der Regel also nicht verändert werden.

I/O-Port 07h wird von beiden Zählern benutzt. Beim Programmieren der Betriebsart eines Zählers muß dafür gesorgt werden, daß der andere Zähler davon nicht beeinflußt wird. Im folgenden wird der Einfachheit halber davon ausgegangen, daß Zähler-A läuft, also Bit-5 von I/O-Port 07h immer gesetzt ist.

13.2.1. Ereigniszähler (Zähler B)

1. Zählkanal wählen

Der Zähleingang *channel* kann aus den digitalen Kanälen IP-8 bis IP-15 ausgewählt werden. Dabei entspricht *channel*=0 dem Eingang IP-8, *channel*=1 dem Eingang IP-9, usw.

out8_port(0ah,*channel*)	Zähleingang *channel* (s.o.) einstellen

2. Zählermode und -flanke einstellen, Zähler zurücksetzen

a) Positive Flanken:

out8_port(07h,20h)	Betriebsart einstellen und Zähler auf Null

b) Negative Flanken:

out8_port(07h,28h)	Betriebsart einstellen und Zähler auf Null

3. Zähler starten:

a) Positive Flanken zählen:

out8_port(07h,60h)	Zähler starten

b) Negative Flanken zählen:

out8_port(07h,68h)	Zähler starten

4. Zählerstand auslesen:

Der Zählerstand kann nur aus dem Latch gelesen werden. Weder Lesen noch Latchen haben einen Einfluß auf den Zählvorgang. Es gehen also in keinem Fall Impulse verloren.

out8_port(08h,00h)	Zählerstand latchen
data = in16_port(04h)	Latch auslesen

Wenn der Zählerstand 65535 überschreitet, wird wieder von Null an gezählt und ein Interrupt (TIMER-B-UP = IRQ-4) ausgelöst, sofern dieser Interrupt demaskiert ist.

13.2.2. Zählen von Inkrementalgebersignalen

Ein Inkrementalgeber wird an IP-3 und IP-4 angeschlossen. Mit IP-0 kann der aktuelle Zählerstand ins Latch gespeichert und mit IP-1 der Zähler zurückgesetzt werden. Wenn IP-0 oder IP-1 nicht verwendet werden, sollten sie mit Masse verbunden werden. Das Inkrementalgeberinterface enthält einen 8 Bit-Vorwärts- und einen 8 Bit-Rückwärtszähler. Die Position des Gebers ergibt sich aus der Differenz der beiden Zählerstände. In aller Regel werden Zähler benötigt, die breiter als 8 Bit sind. Dazu wird ein Echtzeitprogramm verwendet, das die Überläufe der beiden Zähler zählt. Ein entsprechendes Echtzeitprogramm (M2P0308.EXE) befindet sich im Lieferumfang, so daß das Programmieren mit den Ports im Normalfall nicht nötig ist.

Im folgenden wird lediglich die Programmierung der Hardware ohne Überlaufüberwachung beschrieben.

1. Zählermode einstellen und Zähler zurücksetzen

Der Inkrementalgeber kann mit Einfach-, Zweifach- oder Vierfach-Auswertung (s. Kapitel 3) betrieben werden.

Mögliche Einstellungen:

out8_port(07h,25h)	Einfach-Auswertung, Zähler auf Null
out8_port(07h,26h)	Zweifach-Auswertung, Zähler auf Null
out8_port(07h,27h)	Vierfach-Auswertung, Zähler auf Null

2. Flanken für Latch und Reset einstellen

Diese Einstellung ist nur von Bedeutung, wenn mit IP-0 und IP-1 gelatcht oder zurückgesetzt werden soll. Im I/O-Port 05h des Interrupt-Controllers wird eingestellt, ob Latchen und Reset mit positiver oder negativer Flanke ausgeführt werden.

Mögliche Einstellungen:

out8_port(05h,00h)	Latch und Reset mit positiver Flanke
out8_port(05h,01h)	Latch mit pos., Reset mit neg. Flanke
out8_port(05h,02h)	Latch mit neg., Reset mit pos. Flanke
out8_port(05h,03h)	Latch und Reset mit negativer Flanke

3. Zähler starten:

Mögliche Einstellungen:

out8_port(07h,65h)	Einfach-Auswertung, Zähler starten
out8_port(07h,66h)	Zweifach-Auswertung, Zähler starten
out8_port(07h,67h)	Vierfach-Auswertung, Zähler starten

4. Zählerstände ins Latch speichern

Die Zählerstände können nur aus dem Latch gelesen werden. Das Latchen hat keinen Einfluß auf den Zählvorgang. Es gehen also in keinem Fall Impulse verloren. Wenn mit IP-0 gelatcht wird, kann dieser Schritt entfallen.

out8_port(08h,00h)	Zählerstände latchen

5. Latch auslesen und Position bestimmen

Im höherwertigen Byte des Latches steht das Ergebnis des Aufwärtszählers, im niederwertigen das des Abwärtszählers. In der Variablen ***position*** wird die Position des Gebers aus der Differenz beider Zählerstände ermittelt.

data = in16_port(04h) Latch auslesen

position = (*data* AND ff00h) / 256 - (*data* AND 00ffh)

13.2.3. Pulsbreiten-, Periodendauer- und Frequenzmessung

Pulsbreiten-, Periodendauer- und Frequenzmessung beruhen auf unterschiedlichen Prinzipien (siehe Kapitel 3.3.), sind aber von der Programmierung her sehr ähnlich und werden deshalb gemeinsam behandelt.

1. Meßkanal wählen

Der Meßeingang *channel* kann aus den digitalen Kanälen IP-8 bis IP-15 ausgewählt werden. Dabei entspricht *channel*=0 dem Eingang IP-8, *channel*=1 dem Eingang IP-9 usw.

out8_port(0ah,*channel*)	Meßeingang *channel* (s.o.) einstellen

2. Betriebsart einstellen und Zähler-B zurücksetzen

Neben der Betriebsart wird hier bestimmt, ob die jeweilige Messung mit positiven oder mit negativen Flanken gesteuert bzw. gestartet wird.

Frequenzmessung:

Bei der Frequenzmessung werden während einer Referenzzeit positive Flanken am Meßeingang gezählt. Die Referenzzeit kann von Eingang IP-2 oder von Timer-A vorgegeben werden. Üblicherweise wird Timer-A verwendet:

a) Referenzzeit von Timer-A (T_{REF} = *timerdata* * 2µs)

out8_port(07h,21h)	Betriebsart einstellen und Zähler auf Null
out16_port(0ch,*timerdata*)	Referenzzeit einstellen (*timerdata*<4096!)

b) Referenzzeit von IP-2

out8_port(07h,31h)	Betriebsart einstellen und Zähler auf Null

Pulsbreiten- und Periodendauermessung:

Bei Pulsbreiten- und Periodendauermessung werden während der Zeit eines Pulses bzw. während einer Periode Impulse einer Referenzfrequenz (2 MHz ≡ 500 ns) gezählt.

a) Positiven Puls messen bzw. Periode ab steigender Flanke messen

out8_port(07h,22h)	Pulsbreitenmessung wählen, Zähler =0
oder	
out8_port(07h,23h)	Periodendauermessung wählen, Zähler =0

b) Negativen Puls messen bzw. Periode ab fallender Flanke messen

out8_port(07h,2ah)	Pulsbreitenmessung wählen, Zähler =0
oder	
out8_port(07h,2bh)	Periodendauermessung wählen, Zähler =0

3. Zähler für Messung bereitmachen

Vor dem Freischalten muß im Interrupt-Controller das Bit zurückgesetzt werden, das das Ende einer Messung anzeigt:

out8_port(04h,20h)	Bit-5 im Interrupt-Controller löschen

Mit dem nächsten Befehl wird der Zähler freigeschaltet, so daß er, wenn die passende Flanke erscheint, mit dem Zählen beginnen kann. Um den Zähler freizugeben, muß Bit-6 in I/O-Port 07h gesetzt werden. Das heißt, es wird das gleiche Byte wie in Schritt 2 geschrieben, mit dem Unterschied, daß nun Bit-6 gesetzt ist.

Beispiele:

out8_port(07h,61h)	Frequenzmessung mit Timer-A bereit
out8_port(07h,63h)	Periodendauermessung (pos.) bereit
out8_port(07h,72h)	Pulsbreitenmessung (neg.) bereit

Um das Latch freizuschalten, muß es einmal ausgelesen werden. Das Ergebnis dieses Zugriffs ist ohne Bedeutung und kann verworfen werden.

dummy = in16_port(04h)	Latch freischalten

4. Prüfen, ob Messung beendet

Am Ende jeder Messung kann ein Interrupt ausgelöst werden. Wenn das Abholen der Daten nicht von diesem Interrupt gesteuert wird, muß vor dem Lesen des Ergebnisses geprüft werden, ob die Messung beendet ist und ein gültiger Meßwert vorliegt. Nach Ende der Messung ist Bit-5 im Interrupt-Pending-Register (I/O-Port 02h) gesetzt.

pending = in8_port(02h)	Pending-Register lesen
(*pending* AND 20h)>0 ?	Bit-5 gesetzt? Wenn ja: Weiter mit Schritt 5 Wenn nein: Schritt 4 wiederholen

5. Prüfen, ob Messung gültig ist

Bit-4 im Pending-Register bzw. Interrupt IRQ-4 (TIMER-B-UP)

(*pending* AND 10h) = 0 ?	Wenn ja: OK Wenn nein: Überlauf Timer-B

6. Lesen des Ergebnisses

Der Meßwert wird am Ende einer Messung automatisch in das Latch übernommen und die nächste Messung gestartet. Wenn die nächste Messung beendet ist, bevor der alte Wert gelesen wurde, wird der neue Wert ins Latch übernommen. Wenn ohne Interrupts gearbeitet wird, sollte das Bit, das das Ende der Messung anzeigt, vor dem Lesen des Ergebnisses zurückgesetzt werden. Beim Arbeiten mit Interrupts wird das entsprechende Bit vom Echtzeitprogramm zurückgesetzt.

out8_port(04h,20h)	Bit-5 im Interrupt-Controller löschen
data = in16_port(04h)	Zählerstand lesen
dummy = in16_port(04h)	Zähler wieder bereit machen

7. Umrechnen des Meßwertes

puls	=	500 ns * *data*	
periode	=	500 ns * *data*	
frequenz	=	500 kHz * *data* / *timerdata*	Referenz von Timer-A
frequenz	=	*data* / T_{REF}	Referenzzeit T_{REF} von IP-0

8. Weitere Messungen

Um den nächsten und alle weiteren Meßwerte abzuholen, verfahren Sie so, wie ab Schritt 4 beschrieben.

! *Der erste ermittelte Wert muß verworfen werden.*

13.3. Programmieren der Analogeingänge

Die analogen Eingänge werden über sechs I/O-Ports angesprochen. Die folgende Tabelle zeigt eine Übersicht. Ausführlichere Informationen finden Sie in Anhang C.

Adresse	Zugriff	Funktion
40h	W8	Konfigurationsregister 1 - Interne oder externe Signalquellen messen - Auto-Inkrement aktivieren / nicht aktivieren
41h	W8	Konfigurationsregister 2 - Eingangsbereich einstellen - massebezogene oder Differenzmessung
42h	W8	Konfigurationsregister 3 Kanalwahl
43h	R8	Status der Wandlung
43h	W8	Settle-Time (7-Bit, Bit-7 = 0 setzen)
44h	W8	Korrekturfaktor für Gain (nur Multi-LAB/2i und /2h)
45h	W8	Korrekturfaktor für Offset (nur Multi-LAB/2i und /2h)
40h	R16	Ergebnis der Wandlung lesen.

Alle beschreibbaren Ports enthalten Konfigurationsdaten. Diese Daten werden nur durch einen Reset der Karte und durch Software verändert. Sie müssen also nicht für jede Messung neu gesetzt werden. Die Korrekturfaktoren können bei Multi-LAB/2i und /2h aus dem EEPROM (siehe Anhang L) entnommen werden. Außerdem sind sie auf dem Prüfbericht zu Ihrer Karte abgedruckt. Die Werte werden in der weiteren Beschreibung mit ***gain*** und ***offset*** bezeichnet.

Die Meßergebnisse der Analogwandlung sind, unabhängig vom eingestellten Bereich, immer Integer-Werte zwischen -2048 und 2047, wobei -2048 für die Untergrenze und 2047 für die Obergrenze des eingestellten Bereichs steht (siehe auch Tabelle in Kapitel 3.4.)

13.3.1. Messen eines Eingangs

Um einen Eingang zu messen, gehen Sie wie folgt vor:

1. Konfigurationsregister 1 einstellen

out8_port(40h,40h)	Konfigurationsregister 1 einstellen

2. Settle-Time und Korrekturfaktoren einstellen:

out8_port(43h,50)	Settle-Time auf (z.B.) 5 µs einstellen bei 10 MHz CPU-Takt

Wenn Sie Multi-LAB/2a oder /2d benutzen, können Sie direkt mit Punkt drei fortfahren. Andernfalls werden zuerst noch die Korrekturfaktoren aus dem EEPROM gelesen und in die entsprechenden Register übertragen. Für jeden Spannungsbereich ist ein Wort im EEPROM reserviert. Das höherwertige Byte enthält den Wert für *offset*, das niederwertige den Wert für *gain*. Die folgende Tabelle zeigt die Zuordnung zwischen Eingangsbereich und Nummer des Wortes im EEPROM.

Wort	Bereich	Wort	Bereich	Wort	Bereich
26	± 10 V	32	0 bis +10 V	38	0 bis -10 V
27	± 5 V	33	0 bis +5 V	39	0 bis -5 V
28	± 2,5 V	34	0 bis +2,5 V	40	0 bis -2,5 V
29	± 1,25 V	35	0 bis +1,25 V	41	0 bis -1,25 V
30	± 625 mV	36	0 bis +625 mV	42	0 bis -625 mV
31	± 312,5 mV	37	reserviert	43	reserviert

Beispiele:

korr = read_eeprom(26)	Korrekturwerte für ±10 V lesen
korr = read_eeprom(33)	Korrekturwerte für 0 bis +5 V lesen
gain = lowbyte(*korr*)	*gain* ist das niederwertige Byte von *korr*
offset = highbyte(*korr*)	*offset* ist das höherwertige Byte von *korr*
out8_port(44h,*gain*)	Gain-Korrekturwert eintragen
out8_port(45h,*offset*)	Offset-Korrekturwert eintragen

3. Eingangsbereich einstellen

Der Eingangsbereich wird mit dem Konfigurationsregister 2 eingestellt. Die in Anhang C angegebenen Bits müssen je nach gewünschtem Bereich gesetzt werden.

Bit-7 bestimmt, ob die erste Wandlung sofort nach Einstellen des Kanals gestartet wird (Auto-Start, Bit-7 = 1) oder nicht (siehe auch Kapitel 3). Üblicherweise wird mit Auto-Start gemessen, da die eingestellte Settle-Time dann immer richtig benutzt wird.

Beispiele:

out8_port(41h,80h)	±10 V, massebez. Eingänge, Auto-Start
out8_port(41h,83h)	±5 V, Differenzeingänge, Auto-Start
out8_port(41h,86h)	0 bis 10 V, massebezogen, Auto-Start
out8_port(41h,a2h)	±1,25 V, massebezogen, Auto-Start

4. Kanal einstellen

Die Kanalnummer (*channel)* gibt an, welcher Analogeingang gemessen wird. Sie liegt für massebezogene Eingänge zwischen 0 und 15, für Differenzeingänge zwischen 0 und 7.

out8_port(42h,*channel*)	Kanalnummer *channel* einstellen

5. Meßwert auslesen

meßwert = in16_port(40h)	Ergebnis der Wandlung in *meßwert*

Wenn mit Auto-Start gearbeitet wird, wird der A/D-Wandler nach Einstellen des Kanals automatisch gestartet (sobald der Settle-Timer abgelaufen ist). Der Lesebefehl wird auf jeden Fall so lange verlängert, bis die Wandlung beendet ist, und gibt dann den Meßwert zurück.

Wenn nicht mit Auto-Start gearbeitet wird, wird der A/D-Wandler durch diesen Zugriff sofort gestartet, der Settle-Timer wird nicht benutzt. Deshalb muß sichergestellt sein, daß zwischen dem Einstellen des Kanals und dem Starten der Wandlung genug Zeit vergeht, damit der Analogteil eingeschwungen ist. Im Zweifelsfall muß zwischen beiden Schritten eine Zeitschleife programmiert werden.

6. Weitere Meßwerte lesen

Das Lesen weiterer Werte vom gleichen Kanal geschieht genauso wie das Lesen des ersten Wertes. Unabhängig davon, ob mit oder ohne Auto-Start gearbeitet wird, wird der A/D-Wandler durch den Zugriff gestartet und gewartet, bis das Ergebnis vorliegt. Der Settle-Timer wird nicht mehr benötigt, da die Kanaleinstellung nicht geändert wurde.

meßwert = in16_port(40h)	Ergebnis der Wandlung in *meßwert*

13.3.2. Messen mehrerer Eingänge

Beim Messen mehrerer Eingänge gibt es verschiedene Möglichkeiten:

- ***Messen mehrerer Kanäle, die nicht aufeinanderfolgen oder mit unterschiedlichen Eingangsbereichen gemessen werden***

In diesem Fall werden die Kanäle nacheinander gemessen wie in Kapitel 13.3.1. beschrieben. Für die erste Messung sind die Schritte 1 bis 5 durchzuführen. Alle weiteren Kanäle werden durch Ausführen der Schritte 2 bis 5 gemessen. Wenn der Meßbereich zwischen zwei Messungen nicht geändert wird, genügen Schritt 4 und 5.

- ***Messen mehrerer aufeinanderfolgender Kanäle (beginnend ab Kanal 0) mit demselben Eingangsbereich***

Hierfür ist der Mode 'Auto-Inkrement' gedacht, der im folgenden beschrieben wird:

1. Konfigurationsregister 1 einstellen

out8_port(40h,50h)	Konfigurationsregister 1 einstellen

2. Settle-Time und Korrekturfaktoren einstellen:

out8_port(43h,50)	Settle-Time auf (z.B.) 5 µs einstellen
out8_port(44h,*gain*)	Gain-Korrektur (nur Multi-LAB/2i, /2h)
out8_port(45h,*offset*)	Offset-Korrektur (nur Multi-LAB/2i, /2h)

3. Eingangsbereich einstellen

Der Eingangsbereich wird mit dem Konfigurationsregister 2 eingestellt. Die in Anhang C angegebenen Bits müssen je nach gewünschtem Bereich gesetzt werden.

Bit-7 bestimmt, ob die Wandlung des ersten Kanals durch Lesen von I/O-Port 40h (Bit-7 = 0) oder durch Timer-A gestartet wird. Das Messen mit Starten durch Ti-

mer-A wird üblicherweise nur für Echtzeitprogramme verwendet, die mit fester Abtastrate (von Timer-A erzeugt) Meßwerte einlesen. In diesem Fall wird das Echtzeitprogramm als DI- oder II-Task unter dem Interrupt von Timer-A installiert.

Beispiele:

out8_port(41h,00h)	±10 V, massebezogen, kein Auto-Start
out8_port(41h,83h)	±5 V, Differenzeingänge, Auto-Start
out8_port(41h,86h)	0 bis 10 V, massebezogen, Auto-Start
out8_port(41h,22h)	±1,25 V, massebezogen, kein Auto-Start

4. Endkanal einstellen

Der erste gewandelte Kanal ist immer Kanal-0. In Register 42h wird die Nummer (*channel*) des Endkanals eingetragen. Sie liegt für massebezogene Eingänge zwischen 0 und 15, für Differenzeingänge zwischen 0 und 7.

out8_port(42h,*channel*)	Endkanal-Nummer *channel* einstellen

5. Meßwerte auslesen

Der folgende Lesezugriff wird so oft durchgeführt, wie Kanäle gewandelt werden sollen:

meßwert = in16_port(40h)	Ergebnis der Wandlung in *meßwert*

Durch den ersten Lesezugriff auf I/O-Port 40h (bzw. durch Timer-A) wird die Wandlung von Kanal 0 gestartet. Der Zugriff wird automatisch verlängert bis die Wandlung beendet ist. Anschließend wird sofort der Kanal inkrementiert und die nächste Wandlung gestartet. Wenn alle Kanäle gelesen sind (muß vom Programm mitgezählt werden), wird automatisch wieder Kanal 0 eingestellt. Die Wandlung von Kanal 0 wird danach erst gestartet, wenn wieder ein Lesezugriff auf I/O-Port 40h erfolgt (bzw. Timer-A abläuft).

6. Weitere Meßwerte lesen

Schritt 5 kann nun beliebig oft wiederholt werden. Achten Sie darauf, daß jedesmal genau *channel*+1 Lesezugriffe durchgeführt werden müssen (die Kanalzählung beginnt ab 0, deshalb "1").

13.4. Programmieren der Analogausgänge

Jeder analoge Ausgang wird über je zwei I/O-Ports programmiert. Ein Port nimmt Konfigurationsdaten entgegen, der andere enthält den eigentlichen Ausgabewert.

Adr	Zugriff	Funktion
46h	W8	Konfiguration von DA-0 setzen - Bereichseinstellung - Zeitpunkt der Freigabe
47h	W8	Konfiguration von DA-1 setzen - Bereichseinstellung - Zeitpunkt der Freigabe
48h	W16	DA-0 Analogausgabewert schreiben
4ah	W16	DA-1 Analogausgabewert schreiben

Alle I/O-Ports sind voneinander unabhängig. Sie werden nach dem Reset nur durch Software verändert. Einmal geschriebene Werte werden also gespeichert, bis sie überschrieben werden. Deshalb muß z.B. die Bereichseinstellung nur einmal vorgenommen werden.

Die I/O-Ports 48h und 4ah nehmen zunächst nur den zu setzenden Wert entgegen und speichern ihn. Wann dieser Wert wirklich am analogen Ausgang erscheint, hängt von den I/O-Ports 46h und 47h ab. Dort kann eingestellt werden, ob beide Kanäle die Werte beim Schreiben von I/O-Port 48h bzw. 4ah übernehmen.

13.4.1. Setzen eines Ausgangs

Hier wird am Beispiel von Kanal 0 gezeigt, wie ein Ausgang gesetzt wird. Für Kanal 1 müssen nur die entsprechenden I/O-Adressen geändert werden.

1. Konfigurationsregister setzen (Kanal 0)

Im Konfigurationsregister wird der Ausgangsbereich eingestellt (nur für Multi-LAB/2i und /2h). Bit-2 muß gesetzt sein, damit der Ausgangswert beim Schreiben von I/O-Port 48h sofort übernommen wird.

Mögliche Bereiche:

out8_port(46h,04h)	0 bis 5 V bzw. 0 bis 20 mA
out8_port(46h,05h)	0 bis 10 V
out8_port(46h,06h)	±5 V
out8_port(46h,07h)	±10 V

2. Ausgang setzen (Kanal 0)

out16_port(48h, *wert*)	Ausgabewert *wert* wird gesetzt und am Ausgang übernommen

Je nach eingestelltem Bereich ist *wert* eine Zahl zwischen 0 und 4095 oder zwischen -2048 und 2047 (siehe Tabelle in Kapitel 3.5.).

13.4.2. Beide Ausgänge setzen

Wenn beide Kanäle gesetzt werden sollen, muß unterschieden werden, ob die Ausgangswerte unabhängig voneinander oder gleichzeitig übernommen werden sollen. Wenn sie unabhängig voneinander gesetzt und übernommen werden sollen, können sie entsprechend der Vorgehensweise im vorigen Kapitel programmiert werden.

Im folgenden wird gezeigt, wie vorzugehen ist, wenn beide Kanäle gleichzeitig gesetzt werden sollen:

1. Konfigurationsregister setzen

Im Konfigurationsregister wird der Ausgangsbereich eingestellt (nur für Multi-LAB/2i und /2h). Bit 2 wird nur in einem der beiden Register =1 gesetzt. Dadurch wird festgelegt, ob beide Ausgangswerte beim Schreiben von Kanal 0 oder beim Schreiben von Kanal 1 übernommen werden. In diesem Beispiel wird davon ausgegangen, daß Kanal 0 zuerst beschrieben wird und die Ausgangswerte beim Beschreiben von Kanal 1 übernommen werden.

Mögliche Bereiche (Kanal 0)

out8_port(46h,00h)	0 bis 5 V bzw. 0 bis 20 mA
out8_port(46h,01h)	0 bis 10 V
out8_port(46h,02h)	±5 V
out8_port(46h,03h)	±10 V

Mögliche Bereiche (Kanal 1)

out8_port(47h,04h)	0 bis 5 V bzw. 0 bis 20 mA
out8_port(47h,05h)	0 bis 10 V
out8_port(47h,06h)	±5 V
out8_port(47h,07h)	±10 V

2. Ausgänge setzen

out16_port(48h,*wert0*)	Ausgabewert *wert0* wird für Kanal 0 gesetzt, aber nicht übernommen

out16_port(4ah,*wert1*)	Ausgabewert *wert1* wird für Kanal 1 gesetzt, beide Kanäle werden übernommen

Je nach eingestelltem Bereich sind *wert0* und *wert1* Zahlen zwischen 0 und 4095 oder zwischen -2048 und 2047 (siehe Tabelle in Kapitel 3.5.).