19. M-AX-16/IRQ

12 optoentkoppelte Eingänge mit prioritätsgesteuertem Interrupt-Controller, 4 Ausgänge

Funktionsbeschreibung **19-2**

Blockschaltbild 19-5
Technische Daten 19-6
Lieferumfang 19-6

Konfiguration und Einbau **19-7**

EEPROM-Inhalte 19-7

Steckerbelegung **19-8**

Programmierung mit I/O-Zugriffen **19-9**

Lokale I/O-Adressen 19-9
Hinweise zur Programmierung 19-12

Funktionsbeschreibung

Das Modul M-AX-16/IRQ ist ein M-AX-16 Modul mit einem Gate-Array XC3042 und dem Design 'IRQ.DLG'. Es bietet 12 interruptfähige Eingänge und 4 Ausgänge. Die Ein- und Ausgänge sind über Optokoppler galvanisch getrennt.

Die zwölf Eingänge sind an einen prioritätsgesteuerten Interrupt-Controller angeschlossen, der mit einer (per Software wählbaren) Interrupt-Leitung der Basiskarte verbunden werden kann. Die interruptauslösende Flanke ist für jeden Eingang getrennt einstellbar. Alle Eingänge können unabhängig voneinander maskiert und demaskiert werden.

Alle Einstellungen werden per Software vorgenommen.

! *Beachten Sie, daß die Definition der Ausgangspegel von der in der Einführung im Abschnitt 'Digitalausgänge' angegebenen Vereinbarung abweicht. Bei diesem Modul gilt ein leitender Ausgangstransistor als log. 1, ein sperrender als log. 0.*

Funktion des Interrupt-Controllers

Der Interrupt-Controller besteht aus folgenden Komponenten:

- Flankeneinstellung für alle Eingänge.
- Interrupt-Pending-Register (IPR), in dem für jeden Eingang ein aufgetretener Interrupt gespeichert wird.
- Interrupt-Mask-Register (IMR) zur Maskierung der einzelnen Eingänge.
- Prioritätsdecoder
- Interrupt-Service-Register, in das die Nummer des Eingangs eingetragen wird, der den aktuellen Interrupt ausgelöst hat.

Sobald an einem Eingang die per Software eingestellte Flanke auftritt, wird im IPR das zugehörige Bit gesetzt. Sofern der Eingang im IMR demaskiert ist, wird durch das Setzen des Bits im IPR ein Interrupt zur Basiskarte ausgelöst und die Nummer des Eingangs in das ISR eingetragen. Die Interrupt-Service-Routine (Hauptprozedur einer DI- oder II-Task) der MODULAR-4 Karte kann durch Auslesen des ISR ermitteln, welcher Eingang den Interrupt ausgelöst hat. Mit dem Lesen wird das ISR auf 0fh (= kein Interrupt liegt an) gesetzt und das entsprechende Bit im IPR gelöscht, so daß der Interrupt-Controller für den nächsten Interrupt bereit ist.

Wenn in der Zeit zwischen Auslösen des Interrupts und Auslesen des ISR ein weiterer Interrupt am gleichen Eingang auftritt, kann er nicht erfaßt werden. Ein Interrupt an einem anderen Eingang wird aber erfaßt. In diesem Fall wird zunächst das entsprechende Bit im IPR gesetzt. Das ISR wird erst dann neu gesetzt, wenn der vorige Interrupt ausgelesen wurde. Falls bis dahin mehrere verschiedene Interrupts aufgelaufen sind, wird der mit der höchsten Priorität in das ISR eingetragen, sobald es ausgelesen wird. Die niedrigste Priorität hat IP-0, die höchste hat IP-11.

Solange beim Auslesen des ISR noch weitere Interrupts anliegen (also Bits im IPR gesetzt sind), ändert sich der Zustand der Interrupt-Leitung zur Basiskarte nicht und es wird kein neuer Interrupt ausgelöst. Die Interrupt-Service-Routine auf der Basiskarte muß also vor Beendigung prüfen, ob noch ein weiterer Interrupt anliegt. Zu diesem Zweck kann der Zustand der Interrupt-Leitung zur Basiskarte gelesen werden. Wenn sie aktiv ist, liegt noch ein weiterer Interrupt an, der genau wie der erste bearbeitet wird. Durch ihn wurde aber kein Interrupt auf der Basiskarte ausgelöst.

Wenn nur ein Interrupt anliegt und zwischen Auslesen des ISR (ISR wird = 0fh gesetzt, Interrupt-Leitung wird zurückgesetzt) und Prüfen der Interrupt-Leitung ein weiterer Interrupt auftritt (Interrupt-Leitung wird gesetzt, Interrupt zur Basiskarte wird ausgelöst), ergibt die Prüfung der Interrupt-Leitung, daß noch ein Interrupt anliegt. Dieser Interrupt ist aber - im Gegensatz zu den oben beschriebenen - im Interrupt-Controller der Basiskarte gespeichert. In diesem Fall ist es sinnvoll, die Interrupt-Service-Routine zu verlassen und diesen Interrupt erst beim folgenden Aufruf zu bearbeiten. Um zu erkennen, ob der im ISR gespeicherte Interrupt einen Interrupt zur Basiskarte ausgelöst hat oder nicht, dient ein weiteres Register. Das 'Trigger-Register' ist gesetzt, wenn der Eingang, der im ISR eingetragen ist, einen Interrupt zur Basiskarte ausgelöst hat.

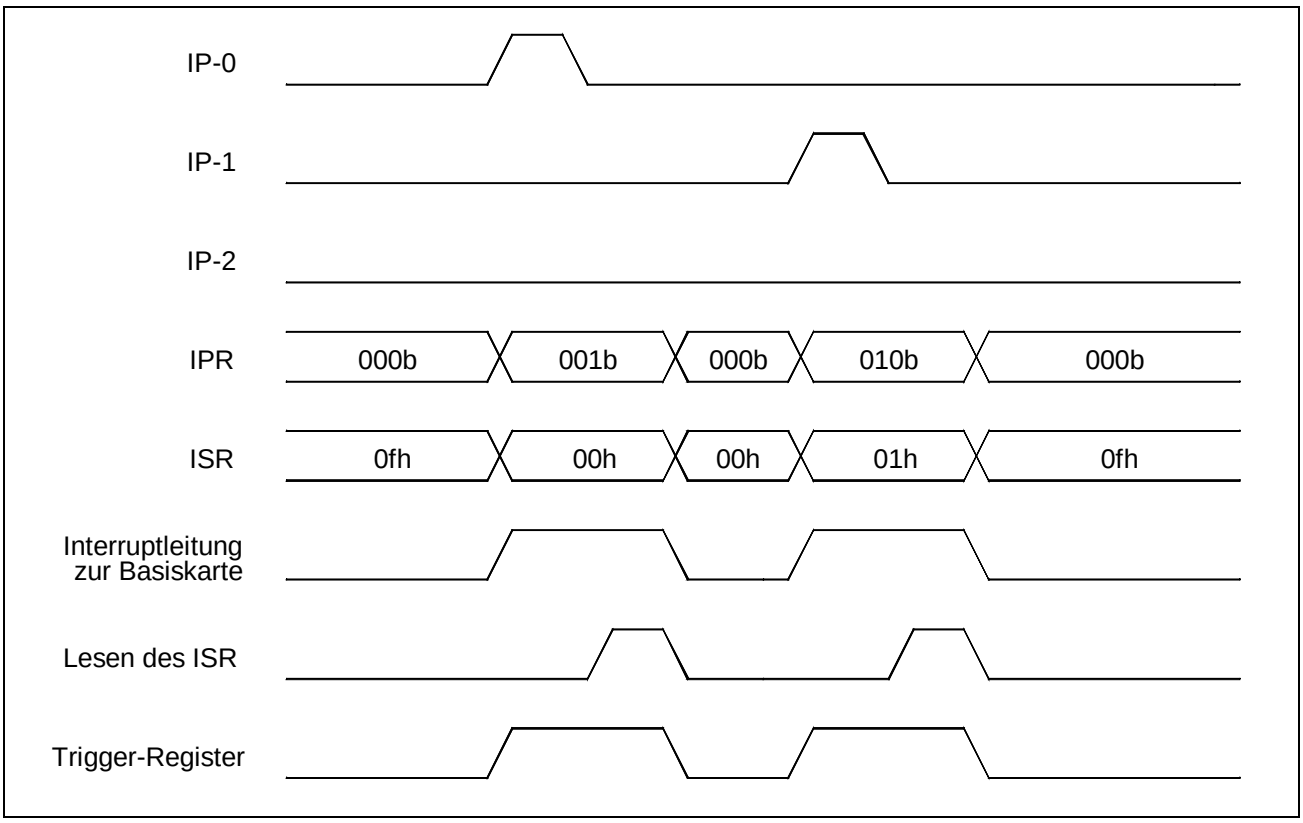


Abb. 19-1: Auftreten von zwei Interrupts nacheinander

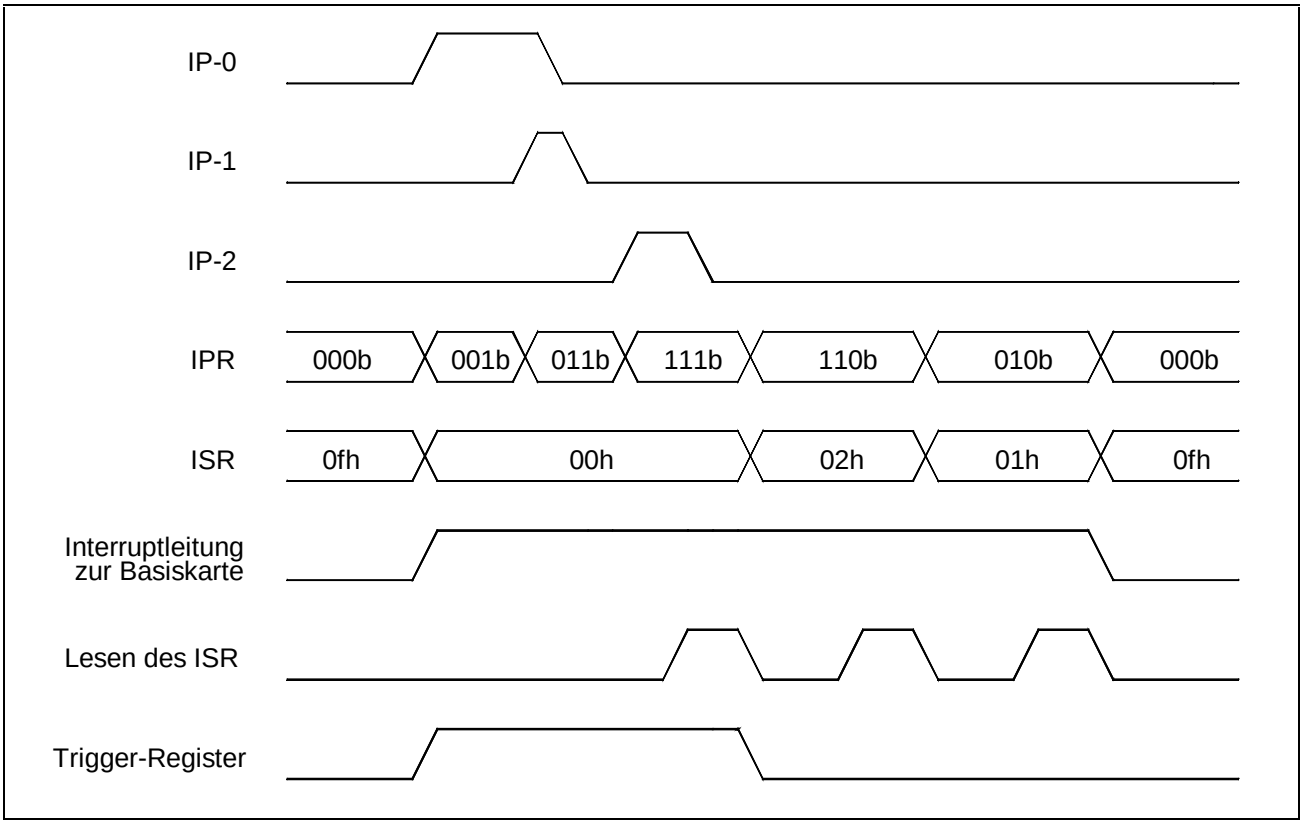
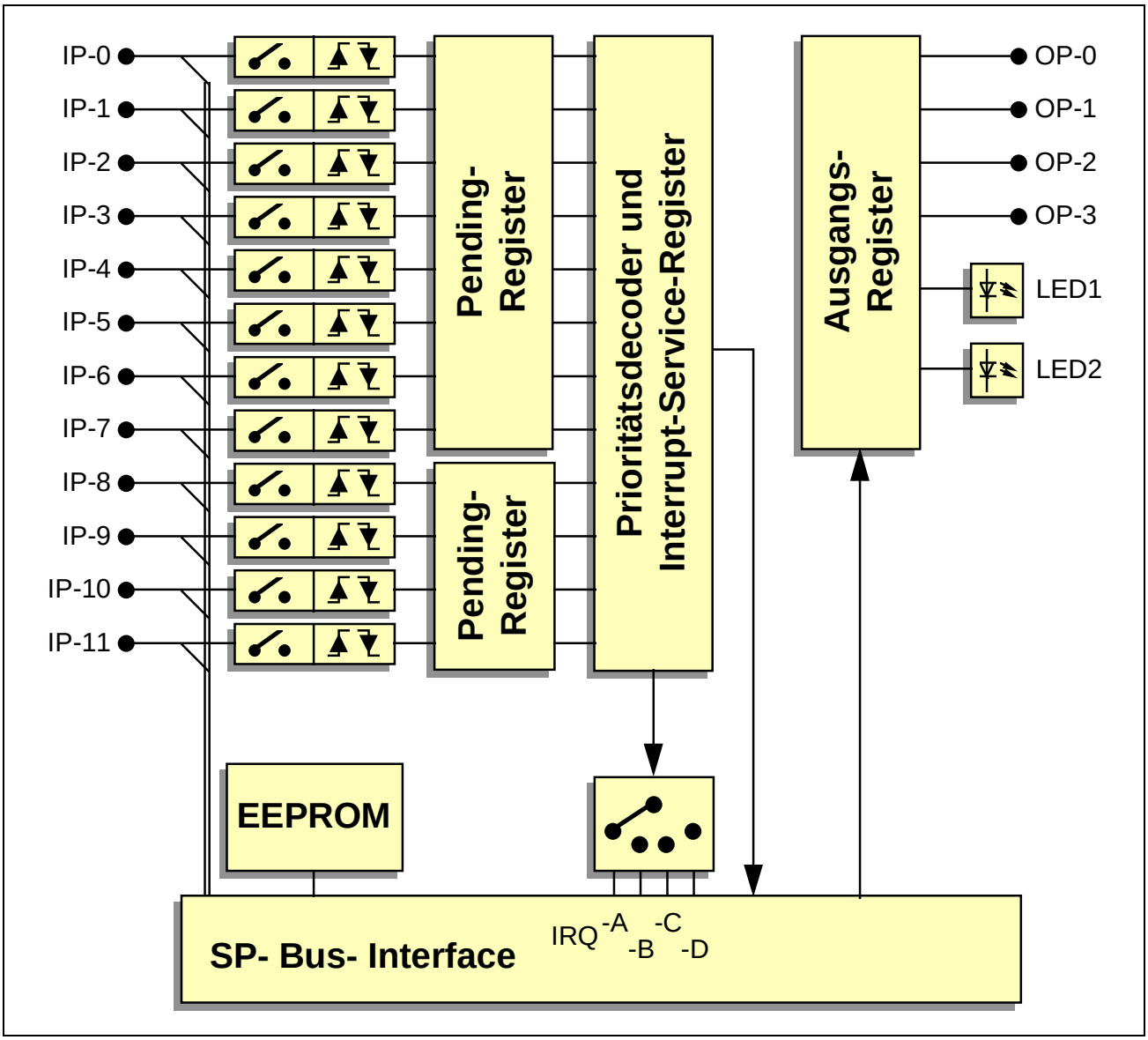


Abb. 19-2: Auftreten von drei Interrupts, bevor das ISR einmal gelesen wurde

Blockschaltbild



Die Ein- und Ausgangs-Optokoppler sind im Blockschaltbild der besseren Übersichtlichkeit wegen nicht eingezeichnet.

Technische Daten

Die aufgeführten technischen Daten ergänzen die, die in der Beschreibung zum M-AX-16 angegeben sind. Die Daten für Stromaufnahme, Eingangsströme usw. finden Sie dort, ebenfalls im Abschnitt 'Technische Daten'.

Parameter	Wert	Einheit
Anzahl der externen Interrupt-Eingänge	12	-
Interruptauslösende Flanke einstellbar	ja	-
Anzahl der digitalen Ausgänge	4	-
Gate-Array-Typ	XC 3042-50	-
Quarzfrequenz (auf dem Modul)	5	MHz

Lieferumfang

- Modul M-AX-16
- 34-poliger Pfostenstecker für Flachbandkabel
- Sortiment Widerstandsnetzwerke WN-1532
- Datenträger mit IRQ-Design

Konfiguration und Einbau

Alle Einstellungen auf dem Modul werden per Software vorgenommen. Nach dem Einschalten der Spannungsversorgung ist das Modul bereits mit dem IRQ-Design aus dem EPROM konfiguriert. Wenn eine aktuellere Version verwendet werden soll, muß das Modul zunächst - wie in der Beschreibung zum Modul 'M-AX-16' im Abschnitt 'Laden eines Logik-Designs' erklärt - mit dem Logik-Design 'IRQ.DLG' geladen werden. Erst dann stehen die hier beschriebenen Funktionen zur Verfügung.

Das Modul muß weder nach einem Reset der Basiskarte noch nach einem Reset des Moduls neu konfiguriert werden.

EEPROM-Inhalte

Die Einstellung von WORT-0 bis WORT-15 ist für alle M-AX-16 Module gleich. Sie finden die Darstellung in der Beschreibung zum Modul 'M-AX-16' im Abschnitt 'EEPROM-Inhalte'. Die designspezifischen Wörter 16 bis 31 sind = 0 gesetzt und dürfen nicht verändert werden.

Steckerbelegung

Das Modul wird über einen 34-poligen Anschlußstecker und ein entsprechendes Flachbandkabel mit der Außenwelt verbunden. Die Angabe - bezeichnet den Emitter des Ausgangs-Optokopplers bzw. die Kathode der LED des Eingangs-Optokopplers, die Angabe + entsprechend den Collector bzw. die Anode.

Pin	Signal
1	- OP-0
2	+ OP-0
3	- OP-1
4	+ OP-1
5	- OP-2
6	+ OP-2
7	- OP-3
8	+ OP-3
9	- IP-0
10	+ IP-0
11	- IP-1
12	+ IP-1
13	- IP-2
14	+ IP-2
15	- IP-3
16	+ IP-3
17	- IP-4
18	+ IP-4
19	- IP-5
20	+ IP-5
21	- IP-6
22	+ IP-6
23	- IP-7
24	+ IP-7
25	- IP-8
26	+ IP-8
27	- IP-9
28	+ IP-9
29	- IP-10
30	+ IP-10
31	- IP-11
32	+ IP-11
33	nicht angeschlossen
34	nicht angeschlossen

Programmierung mit I/O-Zugriffen

Lokale I/O-Adressen

Adresse	Zugr.	Funktion
MBA+00h	W8	Interrupt-Leitung zur Basiskarte anwählen: 0: Keine Interrupt-Leitung 1: IRQ-A 2: IRQ-B 3: IRQ-C 4: IRQ-D Alle anderen Werte sind unzulässig
MBA+01h	W8	Interrupt-Mask-Register IMR1 für IP-0 bis IP-7 Jeder Eingang wird durch ein Bit repräsentiert (Bit-0: IP-0, ..., Bit-7: IP-7). Wenn ein Bit = 1 gesetzt ist, ist der zugehörige Interrupt demaskiert, wenn es = 0 gesetzt ist, ist er maskiert.
MBA+02h	W8	Interrupt-Mask-Register IMR2 für IP-8 bis IP-11 Jeder Eingang wird durch ein Bit repräsentiert (Bit-0: IP-8, ..., Bit-3 IP-11). Wenn ein Bit = 1 gesetzt ist, ist der zugehörige Interrupt demaskiert, wenn es = 0 gesetzt ist, ist er maskiert. Bit-4 bis Bit-7 müssen immer = 0 gesetzt werden.
MBA+00h	R8	Interrupt-Service-Register (ISR) Dieses Register enthält die Nummer des Eingangs (IP-0 = 0, IP-1 = 1, ...), der den aktuellen Interrupt ausgelöst hat. Wenn kein Interrupt anliegt, ist das Register = 0fh.
MBA+01h	R8	Zustand der Eingänge IP-0 bis IP-7 Bit-0 bis Bit-7 entsprechen den Eingängen IP-0 bis IP-7 (wenn ein Bit = 0 ist, ist der zugehörige Eingang stromlos).

Adresse	Zugr.	Funktion
MBA+02h	R8	Zustand der Eingänge IP-8 bis IP-11, des Trigger-Bits und der Interrupt-Leitung zur Basiskarte. Bit-0 bis Bit-3: Zustand der Eingänge IP-8 (= Bit-0) bis IP-11 (= Bit-3) (wenn ein Bit = 0 ist, ist der zugehörige Eingang stromlos). Bit-4: Trigger-Bit: Dieses Bit zeigt an, ob der Eingang, der im ISR eingetragen ist, einen Interrupt zur Basiskarte ausgelöst hat (= 1) oder nicht (= 0). Bit-5: Zustand der Interrupt-Leitung zur Basiskarte: Wenn dieses Bit gesetzt ist, ist die Interrupt-Leitung aktiv. Dieses Bit ist auch gültig, wenn keine Interrupt-Leitung zur Basiskarte angewählt ist. Bit-6 und Bit-7 sind undefiniert.
MBA+03h	W8	Interruptauslösende Flanke an IP-0 bis IP-7 Mit diesem Register wird die aktive Flanke für die Eingänge IP-0 (= Bit-0) bis IP-7 (= Bit-7) eingestellt. Wenn ein Bit = 0 gesetzt ist, wird ein Interrupt ausgelöst, wenn am entsprechenden Eingang eine positive Flanke (Wechsel von stromlosem zu stromdurchflossenen Optokoppler) auftritt. Wenn das Bit = 1 ist, wird ein Interrupt durch eine negative Flanke ausgelöst.
MBA+04h	W8	Interruptauslösende Flanke an IP-8 bis IP-11 Mit diesem Register wird die aktive Flanke für die Eingänge IP-8 (= Bit-0) bis IP-11 (= Bit-3) eingestellt. Wenn ein Bit = 0 gesetzt ist, wird ein Interrupt ausgelöst, wenn am entsprechenden Eingang eine positive Flanke (Wechsel von stromlosem zu stromdurchflossenen Optokoppler) auftritt. Wenn das Bit = 1 ist, wird ein Interrupt durch eine negative Flanke ausgelöst. Bit-4 bis Bit-7 müssen =0 gesetzt werden.
MBA+03h	R8	Interrupt-Pending-Register (IPR1) für IP-0 bis IP-7 Dieses Register zeigt an, von welchen Eingängen unbearbeitete Interrupts anstehen (Bit = 1). Bit-0 ist IP-0 zugeordnet, Bit 7 entsprechend IP-7.

Adresse	**Zugr.**	**Funktion**
MBA+04h	R8	Interrupt-Pending-Register (IPR2) für IP-8 bis IP-11 Dieses Register zeigt an, von welchen Eingängen unbearbeitete Interrupts anstehen (Bit = 1). Bit-0 ist IP-8 zugeordnet, Bit-3 entsprechend IP-11. Bit-4 bis Bit-7 sind undefiniert.
MBA+05h	W8	Ausgänge setzen Bit-0 bis Bit-3 entsprechen OP-0 bis OP-3. Ein gesetztes Bit (= 1) schaltet den Ausgang niedrigohmig, ein zurückgesetztes (= 0) Bit hochohmig.
MBA+06h	W8	LED 1 setzen (Bit-0 = 1: LED ein)
MBA+07h	W8	LED 2 setzen (Bit-0 = 1: LED ein)
MBA+07h	R8x	Modul-Reset Mit dem Auslesen dieser Adresse werden alle Register des Moduls = 0 gesetzt. Die Konfiguration des Moduls mit dem Logik-Design 'IRQ' bleibt aber erhalten.

Hinweise zur Programmierung

Im folgenden wird beschrieben, wie man den Interrupt-Controller des Moduls programmiert. Idealerweise wird die Initialisierung des Moduls von der DI- bzw. II-Task durchgeführt, die auch die Interrupts bedient, z. B. in der Auto-Init-Prozedur oder in einer globalen 'Start'-Prozedur. Weitere Informationen zur Echtzeitprogrammierung finden Sie im Benutzerhandbuch zur Basiskarte.

Initialisierung

1. Modul-Reset durchführen (MBA+07h)
2. Interrupt-Kanal einstellen, der auf der Basiskarte noch nicht benutzt ist (MBA+00h)
3. Positive Interrupt-Flanke für diesen Interrupt-Kanal auf der Basiskarte einstellen (ml8rt_set_int_edge)
4. Gewünschte aktive Interrupt-Flanken für die Moduleingänge einstellen (MBA+03h, 04h, 06h, 07h)
5. Die Moduleingänge, die Interrupts auslösen sollen, demaskieren (MBA+01h, 02h)
6. Interrupt-Eingang der Basiskarte demaskieren (ml8rt_unmask_int)

Sobald nun an einem demaskierten Eingang des Moduls die aktive Flanke auftritt, wird ein Interrupt auf der Basiskarte ausgelöst und die Hauptprozedur der zugehörigen DI- bzw. II-Task aufgerufen.

Bedienen eines Interrupts

Die folgenden Schritte müssen Teil der Hauptprozedur der II- bzw. DI-Task sein.

1. Zuerst wird der Interrupt-Controller der Basiskarte durch Senden eines EOI (End of Interrupt) freigegeben (ml8rt_end_of_int).
2. Interrupt-Service-Register (ISR) auslesen um zu ermitteln, welcher Eingang den Interrupt ausgelöst hat (MBA+00h).
3. Aktionen ausführen, die die Anwendung erfordert.
4. Status der Interrupt-Leitung zur Basiskarte auslesen (MBA+02h, Bit-5). Wenn das Bit = 0 ist, kann die Interrupt-Service-Routine verlassen werden, wenn nicht, steht noch ein Interrupt an. Dann weiter bei 5.

5. Durch Lesen des Zustands des Trigger-Bits (MBA+02h, Bit-4) wird ermittelt, ob der anstehende Interrupt einen Interrupt auf der Basiskarte ausgelöst hat (Bit = 1) oder nicht (Bit = 0). Wenn kein Interrupt zur Basiskarte ausgelöst wurde (Bit = 0), muß der anstehende Interrupt sofort bearbeitet werden (Weiter bei 2.), andernfalls kann die Interrupt-Service-Routine verlassen werden. Der anstehende Interrupt ist im Interrupt-Controller der Basiskarte gespeichert, wodurch sichergestellt ist, daß die Interrupt-Service-Routine erneut durchlaufen wird.

Notizen