

X-DIO-40/i

38 digitale Ein-/Ausgänge, 3 Timer, LED,
12-Kanal Interrupt Controller



1.1. X-DIO-40/i, X-DIO-40 und X-DIO-32

Inhaltsverzeichnis

1.1.	X-DIO-40/i, X-DIO-40 und X-DIO-32	1
1.1.1.	Beschreibung	2
1.1.2.	Modul-Device-Treiber (MDD)	3
1.1.1.1.	Installation	3
1.1.1.2.	Kanaleigenschaftsstruktur CPS_XDIO40	3
1.1.1.3.	Digitale Ein- und Ausgänge (umschaltbar, bidirektional).....	4
1.1.1.4.	Digitale Ausgänge	6
1.1.1.5.	Digitale Eingänge	8
1.1.1.6.	Counter/Timer (nur für X-DIO-40/i)	9
1.1.1.7.	Watchdog (nur X-DIO-40/i)	14
1.1.1.8.	LED.....	15
1.1.1.9.	Device-Index und Datentypen für DIN, DOUT und DIO-Kanäle ...	16
1.1.1.9.1.	Modul-Fertigungsstände A, B und C.....	16
1.1.1.9.2.	Ab Fertigungsstand D	17
1.1.2.	Anschlusspins des Moduls (bezogen auf den Modul-Stecker A)....	18
1.1.3.	Besondere Eigenschaften.....	19

1.1.1. Beschreibung

Das Modul X-DIO-40/i stellt 38 externe digitale TTL-Ein-/Ausgänge, 3 Zähler/Timer, eine on-board LED und einen 12-Kanal Interrupt-Controller mit Interrupt-Overrun Erkennung zur Verfügung.

Sechs der 38 Ein-/Ausgänge können einzeln als Ein- oder Ausgänge konfiguriert werden. Jeder davon ist Interrupt-fähig und kann zusätzliche Funktionen bei Timer 0 bzw. 1 übernehmen. Die übrigen 32 Ein-/Ausgänge sind bei den Fertigungsständen A, B und C in 4-er Gruppen organisiert, wobei jeweils nur eine ganze 4er-Gruppe als Ein- oder Ausgang konfigurierbar ist. Dadurch können gegebenenfalls auch benachbarte Kanäle vom Öffnen eines Kanals betroffen sein. Wenn z.B. ein Kanal mit DIO-2 als Eingang geöffnet wird, sind damit auch DIO-0, DIO-1 und DIO-3 als Eingänge festgelegt.

Ab Fertigungsstand D gelten folgende Abweichungen:

Alle 38 Ein-/Ausgänge können einzeln angesprochen werden, das Zusammenfassen in 4-er Gruppen ist entfallen. Jeweils 16 Ein- bzw. Ausgänge werden zeitgleich abgetastet bzw. gesetzt.

Beim Interrupt-Controller kann jeder der 12 Eingänge einen Interrupt bei einer positiven, negativen oder bei jeder Flanke auslösen. Wenn ein Interrupt noch nicht rechtzeitig bedient wurde und eine weiterer Interrupt am selben Eingang auftritt, wird das als Interrupt-Overrun gewertet und in einem Register gespeichert. Neun der 12 Interrupt-Eingänge sind mit den Ausgängen der 3 Timer, mit den Ein-/Ausgängen 16, 17, 18, 36, 37 und 38 fest verbunden, 3 der Interrupt-Eingänge können mit frei wählbaren Eingängen verbunden werden. Per Software kann dabei je einer der Ein-/Ausgänge 0...15 oder 20...35 angewählt werden. So kann z.B. auch ein Ausgangskanal Interrupt-fähig sein. Das Auftreten von Interrupts wird vom MDD durch Aufrufe von Anwender-Callback-Funktionen signalisiert (s.u.). Die Priorität der einzelnen Interrupt-Quellen bestimmt der Anwender durch die Reihenfolge, in der er Kanäle mit Callback-Funktionen öffnet: Der erste Kanal, der mit einer Callback-Funktion geöffnet wird, hat die höchste Priorität.

Die drei 16-Bit Timer sind 8254-kompatibel (das ist derselbe Timer-Chip wie in allen PCs). Für jeden kann eine von 5 Betriebsarten gewählt werden. Von Timer 0 und 1 kann jeweils der Clock-Eingang CLK, der Gate-Eingang GATE und der Ausgang OUT einzeln an einen externen Ein- bzw. Ausgang gelegt werden. Alle 3 Timer sind Interrupt-fähig. Timer 2 kann, anders als bei PCs, zusätzlich auch als Watchdog konfiguriert werden. In diesem Fall muß er rechtzeitig nachgetriggert werden, andernfalls werden alle als Ausgänge geschalteten digitalen Kanäle hochohmig geschaltet. Außerdem verfügt das Modul über eine per Software schaltbare on-board LED.

Die Variante X-DIO-40 des Moduls enthält keine Timer und keinen Interrupt-Controller.

Die Variante X-DIO-32 des Moduls ist eine Low-Cost-Variante ohne Timer und ohne Interrupt-Controller und verfügt über nur 32 digitale, konfigurierbare Ein-/Ausgänge.

Das Modul ist in folgenden Bestückungsvarianten lieferbar:

Modul-Variante	Modul-Typ	Modul-Subtyp	Digital-I/O	Timer	Interrupt-Controller	on-board LED
X-DIO-40/i	40	1	38	3 bzw. 2 und 1 Watchdog	12	ja
X-DIO-40	40	0	38	-	-	ja
X-DIO-32	40	2	32	-	-	ja

1.1.2. Modul-Device-Treiber (MDD)

1.1.1.1. Installation

Der Modul-Device-Treiber (MDD) für OsX hat die Programmnummer **8028h** und den Dateinamen **mxdio40.exe**. Der Modul-Device-Treiber für Windows hat den Namen **mxdio40.sys**. Der Modul-Device-Treiber für WindowsCE hat den Namen **mxdio40.sys**.

Der MDD ist für alle drei Varianten des Moduls einsetzbar. Die Installation aus einem PC-Programm (z.B. für Steckplatz 1, Layer 0) geschieht wie folgt:

```
Error = max_load_mdd (hModul, 1, 0, 0, 0x8028, NULL, &hMDD);
```

Der Befehl in einer INS-Datei (z.B. für Steckplatz 1, Layer 0) lautet:

```
MAXLOADMDD slot=1 layer=0 progno=8028
```

1.1.1.2. Kanaleigenschaftsstruktur CPS_XDIO40

Die CPS für das Modul hat den Namen CPS_XDIO40 bzw. CPS_XDIO40_A. Der Typ CPS_XDIO40.A wird ab MDD-Version 1.H unterstützt. Die Struktur wurde gegenüber CPS_XDIO40 um die Parameter ulTimeout, ulCallbackEvents und rcCallbackEvents erweitert.

1.1.1.3. Digitale Ein- und Ausgänge (umschaltbar, bidirektional)

Um auf die umschaltbaren, bidirektionalen digitalen Ein- und Ausgänge zugreifen zu können, muss folgende CPS verwendet werden:

Strukturelement	Werte	Bedeutung
<code>.usDevice</code>	<code>DEVICE_DIO</code>	Umschaltbarer DIN/DOOUT Kanal. Zum Umschalten der Richtung stehen die Kanal-Steuerkommandos <code>CMD_DIR_OUTPUT</code> und <code>CMD_DIR_INPUT</code> zur Verfügung.
<code>.usIndexFirst</code>	siehe Tabelle	Nummer des ersten DIOs
<code>.usIndexLast</code>	siehe Tabelle	Nummer des letzten DIOs
<code>.usReadMode</code>	<code>IO_MODE_DIRECT</code>	Direkter Lesezugriff
<code>.usWriteMode</code>	<code>IO_MODE_DIRECT</code>	Direkter Schreibzugriff
<code>.usFlags</code>	<code>_CP_EXCLUSIVE</code>	Wenn ein bidirektionaler Kanal mit dem dazugehörigem Steuerbefehl als Ausgang geschaltet wird, muß der Zugriff exklusiv erfolgen. Das gilt auch, wenn er als Ausgangskanal konfiguriert wird.
	<code>_CP_SYNC_CALLBACK</code>	Nur in Verbindung mit OsX verwendbar! Der Aufruf der Callback-Funktion erfolgt direkt. Ansonsten werden die Aufrufe der Callback-Funktion in eine Warteschlange eingereiht, welche nach Verlassen der Interruptroutine des MDD abgearbeitet wird.
<code>.usMode</code>	0	Reservierter Parameter
<code>.rcCallbackEvents</code>		Dieser Parameter regelt, welche DIOs Interrupts auslösen und damit die Anwender-Callback-Funktion aufrufen sollen. Das Modul kann bei den DIOs 16, 17, 18, 36, 37 und 38 sowie bei 3 frei wählbaren DIOs Interrupts auslösen. Mit den Elementen ausPosEdge und ausNegEdge der Struktur kann festgelegt werden, welche DIOs Interrupts auslösen sollen. Die DIOs sind innerhalb der Arrays rechtsbündig angeordnet, d.h., ist der Kanal auf die DIOs 20...35 geöffnet, so legt Bit 0 in <code>ausPosEdge[0]</code> fest, ob DIO 20 und Bit 15, ob DIO 35 einen Interrupt auslösen soll.
(vom Typ:		
<code>struct</code>		
{		
<code>USHORT ausPosEdge[3];</code>		
<code>USHORT ausNegEdge[3]</code>		
}		
<code>XDIO40_CALLBACK_EVENTS</code>		
)		

Anmerkung

Nach dem Öffnen ist ein bidirektionaler Kanal zunächst als Eingang konfiguriert.

Eingabe- und Ausgabedienst

Der Datentyp des Kanals ist abhängig von den Werten in *.usIndexFirst* und *.usIndexLast* (siehe Tabelle). Der Zugriff auf das Device erfolgt, je nach Datentyp, mit:

Datentyp DATA_UCHAR:

- **max_write_channel_uchar**
- **max_read_channel_uchar**

oder Datentyp DATA_USHORT:

- **max_write_channel_ushort**
- **max_read_channel_ushort**

Sonderdienst

- **max_channel_control**, Steuerbefehl CMD_DIR_INPUT, CMD_DIR_OUTPUT: Umschalten zwischen DIN und DOUT. Dem Dienst werden keine Daten übergeben.

Callback-Funktion (nur verwendbar für Modul-Variante X-DIO-40/i)

Eine Callback-Funktion kann nur bei den Kanälen angegeben werden, bei denen in *rcCallbackEvents* mindestens ein Bit = 1 gesetzt ist. Wenn beim Öffnen des Kanals eine Callback-Funktion angegeben wird, wird diese aufgerufen, wenn eine positive bzw. negative Flanke an einem der dafür konfigurierten DIOs auftritt (Interrupt). Die Callback-Funktion bekommt eine Struktur vom Typ XDIO40_INTERRUPT übergeben.

```
struct
{
    USHORT ausPending[3];
    USHORT ausOverrun[3];
}XDIO40_INTERRUPT;
```

Das Element **ausPending** gibt an, welche DIOs einen Interrupt ausgelöst haben. Das Element **ausOverrun** gibt an, bei welchen DIOs ein Interrupt-Überlauf aufgetreten ist. Die Daten sind dabei rechtsbündig angeordnet: wurde z.B. ein Kanal zu den DIOs 4..8 geöffnet, zeigt das unterste Bit in *ausPending[0]* an, dass DIO 4 einen Interrupt verursacht hat. Nur *ausPending [0]* und *ausOverrun [0]* werden derzeit verwendet. Die restlichen Array Elemente müssen = 0 gesetzt werden.

1.1.1.4. Digitale Ausgänge

Um auf digitale Ausgänge zugreifen zu können, muss ein Kanal mit folgender CPS geöffnet werden:

Strukturelement	Werte	Bedeutung
<i>.usDevice</i>	<i>DEVICE_DOUT</i>	Kanal zu einem digitalen Ausgang
<i>.usIndexFirst</i>	siehe Tabelle	Nummer des ersten Ausgangs
<i>.usIndexLast</i>	siehe Tabelle	Nummer des letzten Ausgangs
<i>.usReadMode</i>	<i>IO_MODE_DIRECT</i>	Direkter Lesezugriff
<i>.usWriteMode</i>	<i>IO_MODE_DIRECT</i>	Direkter Schreibzugriff
<i>.usFlags</i>	<i>_CP_EXCLUSIVE</i> <i>_CP_SYNC_CALLBACK</i>	Der Zugriff erfolgt immer exklusiv Nur in Verbindung mit OsX verwendbar! Der Aufruf der Callback-Funktion erfolgt direkt.
<i>.usMode</i>	0	Reservierter Parameter
<i>.rcCallbackEvents</i>		Dieser Parameter regelt, welche DOUTs Interrupts auslösen und damit die Anwender-Callback-Funktion aufrufen sollen. Das Modul kann bei den DOUTs 16, 17, 18, 36, 37 und 38 sowie bei 3 frei wählbaren DOUTs Interrupts auslösen. Mit den Elementen ausPosEdge und ausNegEdge der Struktur kann festgelegt werden, welche DOUTs Interrupts auslösen sollen. Die DOUTs sind innerhalb der Arrays rechtsbündig angeordnet, d.h., ist der Kanal auf die DOUTs 20...35 geöffnet, so legt Bit 0 in <i>ausPosEdge[0]</i> fest, ob DOUT 20 und Bit 15, ob DOUT 35 einen Interrupt auslösen soll.

(vom Typ:

```

struct
{
USHORT  ausPosEdge[ 3 ];
USHORT  ausNegEdge[ 3 ]
}XDIO40_CALLBACK_
EVENTS
)

```

Eingabe- und Ausgabedienst

Der Datentyp des Kanals ist abhängig von den Werten in *.usIndexFirst* und *.usIndexLast* (siehe Tabelle), der Zugriff erfolgt, je nach Datentyp, mit:

Datentyp DATA_UCHAR:

- **max_write_channel_uchar**
- **max_read_channel_uchar**

oder Datentyp DATA_USHORT:

- **max_write_channel_ushort**
- **max_read_channel_ushort**

Callback-Funktion (nur für X-DIO-40/i)

Eine Callback-Funktion kann nur bei solchen Kanälen angegeben werden, bei denen in

rcCallbackEvents mindestens ein Bit = 1 gesetzt wird. Wenn beim Öffnen des Kanals eine Callback-Funktion angegeben wird, wird diese aufgerufen, wenn eine positive bzw. negative Flanke an einem der dafür konfigurierten DOUTs auftritt, d.h., wenn der Zustand des Ausgangs umgeschaltet wird. Die Callback-Funktion bekommt eine Struktur vom Typ XDIO40_INTERRUPT übergeben.

```
struct
{
    USHORT ausPending[3];
    USHORT ausOverrun[3];
}XDIO40_INTERRUPT;
```

Das Element **ausPending** gibt an, welche DOUTs einen Interrupt ausgelöst haben. Das Element **ausOverrun** gibt an, bei welchen DOUTs ein Interrupt-Überlauf aufgetreten ist. Die Daten sind dabei rechtsbündig angeordnet: wurde z.B. ein Kanal zu den DOUTs 4...8 geöffnet, zeigt das unterste Bit in **ausPending[0]** an, dass DOUT 4 einen Interrupt verursacht hat. Nur **ausPending [0]** und **ausOverrun [0]** werden derzeit verwendet. Die restlichen Array Elemente müssen = 0 gesetzt werden.

1.1.1.5. Digitale Eingänge

Um auf die digitalen Eingänge zugreifen zu können, muss ein Kanal mit folgender CPS geöffnet werden:

Strukturelement	Werte	Bedeutung
<i>.usDevice</i>	<i>DEVICE_DIN</i>	Kanal zu einem digitalen Eingang
<i>.usIndexFirst</i>	siehe Tabelle	Nummer des ersten Eingangs
<i>.usIndexLast</i>	siehe Tabelle	Nummer des letzten Eingangs
<i>.usReadMode</i>	<i>IO_MODE_DIRECT</i>	Direkter Lesezugriff
<i>.usWriteMode</i>	0	Kein Schreibzugriff
<i>.usFlags</i>	<i>_CP_EXCLUSIVE</i>	Der Zugriff muss ab Fertigungsstand D nicht mehr zwingend exklusiv erfolgen sondern kann auch nicht exklusiv erfolgen.
	<i>_CP_SYNC_CALLBACK</i>	Nur in Verbindung mit OsX verwendbar! Der Aufruf der Callback-Funktion erfolgt direkt. Ansonsten werden die Aufrufe der Callback-Funktion in eine Warteschlange eingereiht, welche nach Verlassen der Interruptroutine des MDD abgearbeitet wird.
<i>.usMode</i>	0	Reservierter Parameter.
<i>.rcCallbackEvents</i>		Dieser Parameter regelt, welche DINs Interrupts auslösen und damit die Anwender-Callback-Funktion aufrufen sollen. Das Modul kann bei den DINs 16, 17, 18, 36, 37 und 38 sowie bei 3 frei wählbaren DINs Interrupts auslösen. Mit den Elementen ausPosEdge und ausNegEdge der Struktur kann festgelegt werden, welche DINs Interrupts auslösen sollen. Die DINs sind innerhalb der Arrays rechtsbündig angeordnet, d.h., ist der Kanal auf die DINs 20...35 geöffnet, so legt Bit 0 in <i>ausPosEdge[0]</i> fest, ob DIN 20 und Bit 15, ob DIN 35 einen Interrupt auslösen soll.
(vom Typ:		
<i>struct</i>		
{		
<i>USHORT ausPosEdge[3];</i>		
<i>USHORT ausNegEdge[3]</i>		
<i>}XDIO40_CALLBACK_EVENTS</i>		
)		

Eingabediens

Der Datentyp des Kanals ist abhängig von den Werten in *.usIndexFirst* und *.usIndexLast* (siehe Tabelle). Der Zugriff auf das Device erfolgt, je nach Datentyp mit:

Datentyp `DATA_UCHAR`:

- `max_read_channel_uchar`

oder Datentyp `DATA_USHORT`:

- `max_read_channel_ushort`

Callback-Funktion (nur für X-DIO-40/i)

Eine Callback-Funktion kann nur bei solchen Kanälen angegeben werden, bei denen in `rcCallbackEvents` zumindest ein Bit = 1 gesetzt ist. Wenn beim Öffnen des Kanals eine Callback-Funktion angegeben wird, wird diese aufgerufen, wenn eine positive bzw. negative Flanke an einem der dafür konfigurierten DINs auftritt (Interrupt). Die Callback-Funktion bekommt eine Struktur vom Typ `XDIO40_INTERRUPT` übergeben.

```
struct
{
    USHORT ausPending[3];
    USHORT ausOverrun[3];
}XDIO40_INTERRUPT;
```

Das Element `ausPending` gibt an, welche DINs einen Interrupt ausgelöst haben. Das Element `ausOverrun` gibt an, bei welchen DINs ein Interrupt-Überlauf aufgetreten ist. Die Daten sind dabei rechtsbündig angeordnet: wurde z.B. ein Kanal zu den DINs 4...8 geöffnet, zeigt das unterste Bit in `ausPending[0]` an, dass DIN 4 einen Interrupt verursacht hat. Nur `ausPending [0]` und `ausOverrun [0]` werden derzeit verwendet. Die restlichen Array Elemente müssen = 0 gesetzt werden.

1.1.1.6. Counter/Timer (nur für X-DIO-40/i)

Die Modul-Variante X-DIO-40/i verfügt über 3 Counter/Timer, die dem Chip 82C54 (Hersteller z.B. Intel, NEC, Harris), der in PCs eingesetzt wird, entsprechen. Jeder der 3 Kanäle verfügt über einen Zählengang CLK, einen Triggereingang GATE und eine Ausgang OUT. Um auf einen der 3 Timer des Moduls zugreifen zu können, muss ein Kanal mit folgender CPS geöffnet werden:

Strukturelement	Werte	Bedeutung
.usDevice	DEVICE_TIMER	Timer
.usIndexFirst	0, 1 oder 2	Nummer des Timers
.usIndexLast	.usIndexFirst	Nummer des Timers
.usReadMode	IO_MODE_DIRECT	Direkter Lesezugriff
.usWriteMode	IO_MODE_DIRECT	Direkter Schreibzugriff
.usFlags		
	----- _CP_EXCLUSIVE -----	Muss gesetzt sein. Es kann nur ein Kanal pro Timer geöffnet werden.
		Eines der beiden folgenden Flags muss gesetzt sein:
	_XDIO40_CLOCK_1MHZ	Ein auf dem Modul erzeugter Takt von 1 MHz wird als Eingangstakt an CLK verwendet.
	_XDIO40_CLOCK_DIO	Ein externer Takt, der über DIO 16 (für Timer 0) bzw. DIO 36 (für Timer 1) geliefert wird, wird verwendet.*
		Ab Fertigungsstand D gelten zusätzlich folgende Erweiterungen:
	_XDIO40_CLOCK_DIO16	Ein externer Takt, der über DIO 16 geliefert wird, wird als CLK verwendet.*
	_XDIO40_CLOCK_DIO10MHZ	Ein auf dem Modul erzeugter Takt von 10 MHz wird als CLK verwendet.
	_XDIO40_CLOCK_DIO36	Ein externer Takt, der über DIO 36 geliefert wird, wird als CLK verwendet.*

	_XDIO40_GATE_SOFTWARE	Eines der 3 folgenden Flags muss gesetzt sein: Der Eingang GATE ist per Software schaltbar.
	_XDIO40_GATE_T2OUT	GATE wird mit OUT von Timer-2 verbunden (nur für Timer 0 und 1 verwendbar).
	_XDIO40_GATE_DIO	GATE wird mit DIO 17 (Timer 0) bzw. DIO 37 (Timer 1) verbunden.
		Ab Fertigungsstand D gelten zusätzlich folgende Erweiterungen:
	_XDIO40_GATE_DIO17	Das Gate liegt an DIO 17.*
	_XDIO40_GATE_DIO37	Das Gate liegt an DIO 37.*

	_XDIO40_OUT_DIO	OUT wird mit DIO 18 (Timer 0) bzw. DIO 38 (Timer 1) verbunden.*

	_CP_SYNC_CALLBACK	Nur in Echtzeitprogrammen verwendbar! Der Aufruf der Callback-Funktion erfolgt direkt. Ansonsten werden die Aufrufe gepuffert und im „Hintergrund“ (NI-Task) abgearbeitet

Strukturelement	Werte	Bedeutung
<i>.usMode</i>		Aus den folgenden zwei Gruppen muss jeweils genau ein Flag gesetzt sein!

	<i>_XDIO40_TIMER_MODE_0</i>	Interrupt bei Zählende
	<i>_XDIO40_TIMER_MODE_1</i>	Programmierbares Monoflop
	<i>_XDIO40_TIMER_MODE_2</i>	Rategenerator
	<i>_XDIO40_TIMER_MODE_3</i>	Rechteckgenerator
	<i>_XDIO40_TIMER_MODE_4</i>	Software-getriggertter Impuls
	<i>_XDIO40_TIMER_MODE_5</i>	Hardware-getriggertter Impuls

	<i>_XDIO40_TIMER_MODE_BCD</i>	Der Timer zählt im BCD-Modus
	<i>_XDIO40_TIMER_MODE_BINARY</i>	Der Timer zählt im Binär-Modus
<i>.ulCallbackEvents</i>	-----	
	<i>XDIO40_EVENT_POS_EDGE</i>	Eine positive Flanke an OUT löst einen Interrupt aus.
	<i>XDIO40_EVENT_NEG_EDGE</i>	Eine negative Flanke an OUT löst einen Interrupt aus.

*** Für Timer-2 steht diese Option nicht zur Verfügung!**

Anmerkung

Nach dem Öffnen eines Kanals muss der Timer mit dem Ausgabedienst gestartet werden. Soll ein DIO als CLOCK, GATE oder OUT verwendet werden, so kann dieses Signal nicht als „normaler“ DIO verwendet werden und umgekehrt.

Eingabe- und Ausgabedienst

Der Datentyp des Kanals ist DATA_USHORT:

- **max_write_channel_ushort**
- **max_read_channel_ushort**

Sonderdienst

- **max_channel_control**, Steuerbefehl CTRL_XDIO40_SET_GATE:
Ist das Gate per Software schaltbar, kann dies mit dem Steuerkommando erfolgen. Dem Dienst wird ein ULONG-Wert übergeben, in den je nach gewünschtem Zustand eine 0 oder 1 eingetragen werden muss.

Callback-Funktion

Wenn beim Öffnen des Kanals eine Callback-Funktion angegeben wird, wird diese aufgerufen, wenn eine positive bzw. negative Flanke am Ausgang OUT des Timers auftritt. Die Callback-Funktion bekommt einen ULONG Wert übergeben, der aus einer ODER-Verknüpfung der folgenden Konstanten bestehen kann:

- `XDIO40_INTERRUPT_PENDING`: Interrupt aufgetreten
- `XDIO40_INTERRUPT_OVERRUN`: Interrupt-Überlauf aufgetreten (d.h., bevor ein Interrupt bedient werden konnte, ist bereits ein weiterer Interrupt aufgetreten).

Timer Mode

Die Timer-Modi 0 ... 5 entsprechen den Modi des programmierbaren Intervall-Timers 82C54. Der Timer arbeitet immer als Abwärtszähler.

`_XDIO40_TIMER_MODE_0` (Interrupt bei Zählende, Event Counting)

Nach dem Öffnen des Kanal wird das Zählregister des Timers mit dem darauf folgenden CLK-Impuls geladen. Der Ausgangspin OUT ist zunächst 0. Erreicht der Zähler den Wert 0, so geht OUT auf 1, bis der Zähler neu initialisiert wird. `GATE=0` stoppt den Zähler, `GATE=1` läßt ihn weiterlaufen. Der Zustand von GATE hat keinen direkten Einfluß auf OUT.

`_XDIO40_TIMER_MODE_1` (per Hardware retriggerbares Monoflop)

Nach dem Öffnen des Kanals liegt der Ausgang OUT zunächst auf 1. OUT geht auf 0 (Start des Monoflops) nach dem einem pos. Triggerimpuls an GATE folgenden CLK-Impuls und bleibt auf diesem Pegel, bis der Zähler den Wert 0 erreicht hat. OUT geht dann wieder auf 1. Einen CLK-Impuls nach dem nächsten Trigger geht OUT erneut auf 0.

Tritt während des Zählvorgangs ein Trigger-Impuls auf, wird der Zähler erneut mit dem Anfangswert geladen. Ein Schreibvorgang hat solange keine Auswirkungen, bis ein neuer Trigger-Impuls auftritt. Der Zustand von GATE hat keinen direkten Einfluß auf OUT.

`_XDIO40_TIMER_MODE_2` (Rategenerator, periodisch)

Nach dem Öffnen des Kanals ist OUT zunächst 1 und der Zähler beginnt mit dem darauf folgenden CLK-Impuls mit dem Zählen. Erreicht der Zähler den Wert 1, so geht OUT für einen CLK-Impuls auf 0. Anschließend wird der Anfangswert automatisch neu geladen und der Zählvorgang beginnt erneut. `GATE=0` stoppt, `GATE=1` läßt ihn weiter laufen. Geht GATE während eines Ausgangsimpulses auf 0, eht OUT sofort auf 1. Ein pos. Inputs an GATE lädt den Zähler wieder mit dem Anfangswert und der Zählvorgang wird neu gestartet. Ein Schreibbefehl lädt und startet den Zähler ebenfalls erneut.

_XDIO40_TIMER_MODE_3 (Rechteckgenerator, periodisch)

In diesem Modus wird ein periodisches Rechtecksignal erzeugt. Nach dem Öffnen des Kanal liegt der Ausgang OUT auf 1. Ist der Zähler zur Hälfte abgelaufen, geht OUT auf 0. Erreicht der Zähler den Wert 0, geht OUT wieder auf 1 und der Zähler wird automatisch erneut geladen. GATE=0 stoppt den Zähler, GATE=1 läßt ihn weiter laufen. Geht GATE auf 0, während OUT auf 0 liegt, so geht OUT sofort auf 1. Eine pos. Falnke an GATE (von 0 auf 1) lädt und startet den Zähler mit dem nächsten CLK-Impuls. Das Beschreiben des Timers beeinflusst einen laufenden Zählvorgang nicht. Erst nach dem Ende des aktuellen Halbzyklus wird der Wert geladen.

_XDIO40_TIMER_MODE_4 (Software-getriggter Impuls)

Nach Öffnen des Kanals liegt der Ausgang OUT auf 1. Ist der Zählerstand 0 erreicht, so geht OUT für einen CLK-Impuls auf 0 und dann wieder auf 1. GATE=0 stoppt den Zähler, GATE=1 läßt ihn weiter laufen. Nach dem Beschreiben des Timers wird der Zähler mit dem darauf folgendem CLK-Impuls geladen und mit dem nächsten CLK-Impuls gestartet. Erfolgt während des Zählvorgangs ein Schreibzugriff, wird der Wert beim darauf folgenden CLK-Impuls geladen und der Zählvorgang mit diesem Wert fortgesetzt.

_XDIO40_TIMER_MODE_5 (Hardware-getriggter Impuls)

Die Impulsform am Ausgang OUT stimmt mit der von Mode 4 überein. Die Triggung erfolgt jedoch durch einen Übergang von 0 auf 1 an GATE. Danach wird durch den darauf folgenden CLK-Impuls der Zähler geladen und gestartet. Ist der Zählerwert 0 erreicht, so geht OUT für einen CLK-Impuls auf 0. Tritt während des Zählvorgang ein Trigger-Impuls auf, wird der Anfangswert erneut geladen und der Zählvorgang mit diesem Wert fortgesetzt. Ein Schreibvorgang wirkt sich auf den laufenden Zählvorgang nicht aus. Erst beim nächsten Trigger-Impuls wird der Wert übernommen.

1.1.1.7. Watchdog (nur X-DIO-40/i)

Der Watchdog wird mit Timer-2 des Moduls realisiert. Wenn Timer-2 bereits anders verwendet wird, kann ein Watchdog mehr benutzt werden (und umgekehrt).

Um auf den Watchdog zugreifen zu können, muss folgende CPS verwendet werden:

Strukturelement	Werte	Bedeutung
<i>.usDevice</i>	<i>DEVICE_WATCHDOG</i>	Kanal auf den Watchdog
<i>.usIndexFirst</i>	<i>0</i>	Nummer des Watchdog
<i>.usIndexLast</i>	<i>0</i>	Nummer des Watchdog
<i>.ulTimeout</i>	<i>1...260</i>	Timeout für Watchdog in ms
<i>.usReadMode</i>	<i>0</i>	Reservierter Parameter
<i>.usWriteMode</i>	<i>IO_MODE_DIRECT</i>	Direkter Schreibzugriff
<i>.usFlags</i>	<i>_CP_EXCLUSIVE</i>	Der Zugriff erfolgt exklusiv
	<i>_CP_SYNC_CALLBACK</i>	Nur in Verbindung mit OSX verwendbar! Der Aufruf der Callback-Funktion erfolgt direkt. Ansonsten werden die Aufrufe der Callback-Funktion in einer Warteschlange eingereiht, welche nach verlassen der Interruptroutine des MDD abgearbeitet wird.
<i>.ulCallbackEvents</i>	<i>XDIO40_EVENT_WATCHDOG</i>	Ist der Watchdog abgelaufen, ruft er die Callback-Funktion auf.

Ausgabedienst

Der Datentyp ist DATA_VOID. Der Zugriff erfolgt mit:

- **max_trigger_channel**

Anmerkung

Damit der Watchdog nicht aktiv wird, muss er zumindest einmal während der in der CPS angegebenen Timeout-Zeit nachgetriggert werden. Bleibt das aus, läuft der Watchdog-Timer ab und alle Ausgänge werden hochohmig geschaltet.

Das Deaktivieren des Watchdog erfolgt durch Schließen des Kanals.

Callback-Funktion

Wenn beim Öffnen des Kanals eine Callback-Funktion angegeben wird, wird diese aufgerufen, wenn der Watchdog nicht rechtzeitig nachgetriggert wurde. Die Callback-Funktion bekommt bei ihrem Aufruf keine Parameter übergeben.

Sonderdienst

- **max_channel_info**, Infotyp INFO_DEVICE: Der Zustand des Watchdog kann jederzeit abgefragt werden. Die Funktion liefert den Status des Watchdog als ULONG-Wert zurück, wobei 0 bedeutet, dass der Watchdog nicht abgelaufen ist, 1 bedeutet, dass der Watchdog abgelaufen ist.
- **max_channel_control**, Steuerbefehl CMD_START: Mit diesem Sonderdienst kann der Watchdog, nachdem er abgelaufen war, wieder gestartet werden. Dem Dienst werden keine Parameter übergeben.

1.1.1.8. LED

Das Modul bietet eine on- board LED, die über folgende CPS angesprochen werden kann:

Strukturelement	Werte	Bedeutung
<i>.usDevice</i>	<i>DEVICE_LED</i>	Kanal zur on-board LED
<i>.usIndexFirst</i>	0	Nummer der LED
<i>.usIndexLast</i>	0	Nummer der LED
<i>.usReadMode</i>	<i>IO_MODE_DIRECT</i>	Direkter Lesezugriff
<i>.usWriteMode</i>	<i>IO_MODE_DIRECT</i>	Direkter Schreibzugriff
<i>.usFlags</i>	0	Keine Bedeutung
	<i>_CP_EXCLUSIVE</i>	Der Zugriff auf die LED erfolgt exklusiv
<i>.usMode</i>	0	Reservierter Parameter

Eingabe- und Ausgabedienst

Um die LED ein- bzw. auszuschalten muss eine 1 bzw. 0 in den Kanal geschrieben werden. Der Datentyp des Kanals ist DATA_UCHAR.

- **max_write_channel_uchar**
- **max_read_channel_uchar**

1.1.1.9. Device-Index und Datentypen für DIN, DOUT und DIO-Kanäle

1.1.1.9.1. Modul-Fertigungsstände A, B und C

In der folgenden Tabelle sind für DIN, DOUT und DIO-Kanäle die erlaubten Kombinationen von `.usIndexFirst` und `.usIndexLast` sowie der Datentyp des Kanals aufgelistet. Da die Ein-/Ausgänge jeweils in 4er-Gruppen organisiert sind, kann ein Kanal nur über eine oder mehrere Gruppen geöffnet werden.

<code>.usIndexFirst</code>	<code>.usIndexLast</code>	Datentyp	Anmerkung
0	3	DATA_UCHAR	
0	7	DATA_UCHAR	
0	11	DATA_USHORT	
0	15	DATA_USHORT	
4	7	DATA_UCHAR	
4	11	DATA_UCHAR	
4	15	DATA_USHORT	
8	11	DATA_UCHAR	
8	15	DATA_UCHAR	
12	15	DATA_UCHAR	
16	16	DATA_UCHAR	nicht bei X-DIO-32
17	17	DATA_UCHAR	nicht bei X-DIO-32
18	18	DATA_UCHAR	nicht bei X-DIO-32
20	23	DATA_UCHAR	
20	27	DATA_UCHAR	
20	31	DATA_USHORT	
20	35	DATA_USHORT	
24	27	DATA_UCHAR	
24	31	DATA_UCHAR	
24	35	DATA_USHORT	
28	31	DATA_UCHAR	
28	35	DATA_UCHAR	
32	35	DATA_UCHAR	
36	36	DATA_UCHAR	nicht bei X-DIO-32
37	37	DATA_UCHAR	nicht bei X-DIO-32
38	38	DATA_UCHAR	nicht bei X-DIO-32

1.1.1.9.2. Ab Fertigungsstand D

Ab Fertigungsstand D gelten folgende Erweiterungen: alle Kanäle sind einzeln ansprechbar. Zu beachten ist, dass sich *.usIndexFirst* und *.usIndexLast* jeweils in derselben Gruppe finden müssen: Gruppe_0_15 = Kanal 0...15, Grupper_16_18 = Kanal 16...18, Gruppe_20_35 = Kanal 20...35 und Grupper_37_39 = Kanal 37...39. Innerhalb dieser Gruppen gibt es keine Einschränkungen.

In der folgenden Tabelle sind für DIN-, DOUT- und DIO-Kanäle einige erlaubte Kombinationen von *.usIndexFirst* und *.usIndexLast* sowie der Datentyp des Kanals aufgelistet. Dennoch verliert die oben genannte Tabelle nicht an Gültigkeit! Der Zugriff auf alle Kanäle kann mit DATA_USHORT erfolgen. Dies sollte aus Geschwindigkeitsgründen beachtet werden. Da alle Register 16 Bit orientiert sind, erspart man sich die Umrechnung (typcast) mit Bitshift und Maskierung bei einem DATA_UCHAR Zugriff. Die Register bei den (älteren) Fertigungsständen A, B und C waren 8 Bit organisiert. Dort sollte der Zugriff mit DATA_UCHAR erfolgen.

.usIndexFirst	.usIndexLast	Datentyp	Anmerkung
0	15	DATA_USHORT	
2	14	DATA_USHORT	
1	1	DATA_UCHAR	
9	13	DATA_UCHAR	
:	:		
:	:		
16	18	DATA_UCHAR	nicht bei X-DIO-32
16	16	DATA_UCHAR	nicht bei X-DIO-32
20	35	DATA_USHORT	
21	21	DATA_UCHAR	
29	32	DATA_UCHAR	
:	:		
:	:		
36	38	DATA_UCHAR	nicht bei X-DIO-32
37	37	DATA_UCHAR	nicht bei X-DIO-32

1.1.2. Anschlusspins des Moduls (bezogen auf den Modul-Stecker A)

Kanal	Pin	Kanal	Pin
DIO-0	1	DIO-20	21
DIO-1	2	DIO-21	22
DIO-2	3	DIO-22	23
DIO-3	4	DIO-23	24
DIO-4	5	DIO-24	25
DIO-5	6	DIO-25	26
DIO-6	7	DIO-26	27
DIO-7	8	DIO-27	28
DIO-8	9	DIO-28	29
DIO-9	10	DIO-29	30
DIO-10	11	DIO-30	31
DIO-11	12	DIO-31	32
DIO-12	13	DIO-32	33
DIO-13	14	DIO-33	34
DIO-14	15	DIO-34	35
DIO-15	16	DIO-35	36
DIO-16 bzw. Timer 0 Clock bzw. Timer 0/1 Clock *)	17 ¹	DIO-36 bzw. Timer 1 Clock bzw. Timer 0/1 Clock *)	37 ¹
DIO-17 bzw. Timer 0 Gate bzw. Timer 0/1 Gate *)	18 ¹	DIO-37 bzw. Timer 1 Gate bzw. Timer 0/1 Gate *)	38 ¹
DIO-18 bzw. Timer 0 Output bzw. Timer 0/1 Output *)	19 ¹	DIO-38 bzw. Timer 1 Output bzw. Timer 0/1 Output *)	39 ¹
GND	20	GND	40

*) Ab Fertigungsstand D des Moduls kann als Quelle der Signale, die an den Pins 17, 18, 19 und 37, 38, 39 zur Verfügung gestellt werden, zwischen Timer 0 und Timer 1 gewählt werden.

¹ bei X-DIO-32 angeschlossen, aber ohne Funktion

1.1.3. Besondere Eigenschaften

Parameter	Wert	Einheit
Anzahl externer digitaler Ein- bzw. Ausgänge		
Modul-Variante X-DIO-40/i	38	-
Modul-Variante X-DIO-40	38	-
Modul-Variante X-DIO-32	32	-
Eingangsspannung (andere Logik-Standards auf Anfrage) (kompatibel mit 5V-TTL und 5V-, 3,3V-, 2,5V-CMOS)		
log. 0	< 0,8	V
log. 1	> 2,0	V
Input Leakage Current, max.	10	µA
Ausgangsspannung		
log. 0, max. (IOL = 8mA)	0,4	V
log. 1, min. (IOH = -4mA)	2,4	V
Ausgangstrom log. 0, min./typ.	12/36	mA
log. 1, min./typ.	-12/-24	mA
Überspannungsfestigkeit der Eingänge		
für Impulse < 10 ns und < 200mA	-0,5 .. +5,5	V
	-2,0 .. +7,0	V
Watchdog-Timer (nur bei X-DIO-40/i), programmierbar	1...260	ms
Timer (nur bei X-DIO-40/i), Anzahl	3	-
Auflösung	16	Bit
Betriebsarten (z.B. Event Counter, Monoflop, Rategenerator, Rechteckgenerator, Impulsgenerator)	5	-
Temperatur-Bereich, Betrieb	0...+70	°C
optional (auf Anfrage)	-40...+85	°C
Abmessungen	29 x 58 x 8	mm
Gewicht	8,85	g
Stromaufnahme (3,3V) min./max. (alle Anschlüsse n.c.) (die X-Bus Spannungen ±12V werden nicht verwendet)	455/500	mA

