

Modul-Device-Treiber M-C16-3

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

Installationsparameter

Parameter	Wert
Dateiname:	ML8D2F00.LIB
Programmnummer:	426h
Tasknummer:	Steckplatz des Moduls
Interruptnummer:	0 ¹
Länge des Datenbereichs:	0
Flags:	809h

¹ = 0, wenn NI-Task, sonst = der Interruptnummer, die bei „Anwahl der Interruptleitung zur Basiskarte (Device-Typ *DEVICE_CTRL*, Device-Index 0) angegeben wird. Dann muß die Task als II-Task installiert werden

Befehl in INS-Datei (z.B. für Steckplatz 1 als II-Task und IRQ-D):

M8INST ML8D2F00 0426 0001 7E 000000 00000809

Kanaleigenschaftsstruktur CPS_MC163

Ofs	Strukturelement	Datentyp	Bedeutung
0	<i>.usDevice</i>	USHORT	Device-Typ
2	<i>.usIndexFirst</i>	USHORT	Index des ersten Devices
4	<i>.usIndexLast</i>	USHORT	Index des letzten Devices
6	<i>.usFlags</i>	USHORT	Flags für zusätzliche Kanaleigenschaften
8	<i>.usReadMode</i>	USHORT	Lesemodus
10	<i>.usWriteMode</i>	USHORT	Schreibmodus
12	<i>.usMode</i>	USHORT	Betriebsart und Unterbetriebsart

Ofs	Strukturelement	Datentyp	Bedeutung
14	<i>.usCounterInput</i>	USHORT	Zähleingang 0 (benutzt bei: Aufwärtszähler, Abwärtszähler, Auf-/Abwärtszähler, Timer, Inkrementalgeber, Frequenzmessung, Pulsbreitenmessung, Periodendauerermessung, Geschwindigkeitsmessung) und auslösende Flanke Zähleingang 1 (benutzt bei: Auf-/Abwärtszähler, Inkrementalgeber, Frequenzmessung, Pulsbreitenmessung, Periodendauerermessung, Geschwindigkeitsmessung) und auslösende Flanke
16	<i>.usCounterOutput</i>	USHORT	Zählerausgang
18	<i>.usCommandInput</i>	USHORT	Steuerungseingang (Start, Stop, Load, Freeze und Reset)
20	<i>.usInterrupt</i>	USHORT	Definition der Interruptbedingung
22	<i>.usTime</i>	USHORT	Referenzfrequenz/Referenzzeit (benutzt bei Timer, Frequenzmessung, Pulsbreitenmessung, Periodendauerermessung, Geschwindigkeitsmessung) bzw. Verzögerungszeit (benutzt bei Inkrementalgeber)
24	<i>.usMultiple</i>	USHORT	Anzahl der Periodendauerermessungen
26	<i>.usValueSize</i> ¹	USHORT	Auflösung des Ergebnisses in Anzahl Byte: 2 Byte = 0..65535 (= Minimal-Auflösung = 16-Bit HW-Counter), 3 Byte = 0..16777215, ... allg.: 256 hoch (<i>usValueSize</i>)

¹ neu seit Version 1.I. vom 17.7.2000

Anwahl der Interruptleitung zur Basiskarte

Ofs	Strukturelement	Datentyp	Wert
0	<i>.usDevice</i>	USHORT	Device-Typ: <i>DEVICE_CTRL</i> = 0401h
2	<i>.usIndexFirst</i>	USHORT	Device-Index (= 0)
4	<i>.usIndexLast</i>	USHORT	= <i>.usIndexFirst</i>
6	<i>.usFlags</i>	USHORT	Flags: Bit 0: <i>_CP_EXCLUSIVE</i> = 1 (exklusiv) Alle anderen Bits = 0.
8	<i>.usReadMode</i>	USHORT	Lesemodus: <i>IO_MODE_DIRECT</i> = 1 (direkt)
10	<i>.usWriteMode</i>	USHORT	Schreibmodus: <i>IO_MODE_DIRECT</i> = 1 (direkt)

Alle anderen Strukturelemente werden nicht ausgewertet.

Die Anwahl der Interruptleitung erfolgt durch einen Schreibzugriff mit **mddx_write_channel_byte** auf den Kanal. Der Übergabeparameter kann dabei folgende Werte annehmen:

<code>_MC163_NO_IRQ</code>	(= 00h)	Keine Interruptleitung anwählen
<code>_MC163_IRQ_A</code>	(= 01h)	Interruptleitung IRQ-A anwählen
<code>_MC163_IRQ_B</code>	(= 02h)	Interruptleitung IRQ-B anwählen
<code>_MC163_IRQ_C</code>	(= 03h)	Interruptleitung IRQ-C anwählen
<code>_MC163_IRQ_D</code>	(= 04h)	Interruptleitung IRQ-D anwählen
<code>_MC163_IRQ_E</code>	(= 05h)	Interruptleitung IRQ-E anwählen
<code>_MC163_IRQ_F</code>	(= 06h)	Interruptleitung IRQ-F anwählen
<code>_MC163_NMI</code>	(= 07h)	Interruptleitung NMI anwählen

Die aktuell angewählte Interruptleitung kann durch einen Lesezugriff mit **mddx_read_channel_byte** auf den Kanal ermittelt werden.

Maskieren/Demaskieren der Interrupteingänge

Dieser Kanal ermöglicht es, die Interrupts INT-0 bis INT-15 des Moduls zu maskieren oder zu demaskieren und jeweils die aktive Flanke zu definieren.

Ofs	Strukturelement	Datentyp	Wert
0	<code>.usDevice</code>	USHORT	Device-Typ: <code>DEVICE_CTRL</code> = 0401h
2	<code>.usIndexFirst</code>	USHORT	Device-Index (= 1)
4	<code>.usIndexLast</code>	USHORT	= <code>.usIndexFirst</code>
6	<code>.usFlags</code>	USHORT	Flags: Bit 0: <code>_CP_EXCLUSIVE</code> = 1 (exklusiv) Alle anderen Bits = 0.
8	<code>.usReadMode</code>	USHORT	Lesemodus: <code>IO_MODE_DIRECT</code> = 1 (direkt)
10	<code>.usWriteMode</code>	USHORT	Schreibmodus: <code>IO_MODE_DIRECT</code> = 1 (direkt)

Alle anderen Strukturelemente werden nicht ausgewertet.

Ein Interrupteingang wird durch einen Schreibzugriff mit **mddx_write_channel_dword** auf den Kanal demaskiert:

<code>_MC163_DEMASK_INT_0</code>	(= 0001h)	Demaskiert Interrupteingang INT-0
<code>_MC163_DEMASK_INT_1</code>	(= 0002h)	Demaskiert Interrupteingang INT-1
<code>_MC163_DEMASK_INT_2</code>	(= 0004h)	Demaskiert Interrupteingang INT-2
<code>_MC163_DEMASK_INT_3</code>	(= 0008h)	Demaskiert Interrupteingang INT-3
<code>_MC163_DEMASK_INT_4</code>	(= 0010h)	Demaskiert Interrupteingang INT-4
<code>_MC163_DEMASK_INT_5</code>	(= 0020h)	Demaskiert Interrupteingang INT-5
<code>_MC163_DEMASK_INT_6</code>	(= 0040h)	Demaskiert Interrupteingang INT-6
<code>_MC163_DEMASK_INT_7</code>	(= 0080h)	Demaskiert Interrupteingang INT-7
<code>_MC163_DEMASK_INT_8</code>	(= 0100h)	Demaskiert Interrupteingang INT-8
<code>_MC163_DEMASK_INT_9</code>	(= 0200h)	Demaskiert Interrupteingang INT-9
<code>_MC163_DEMASK_INT_10</code>	(= 0400h)	Demaskiert Interrupteingang INT-10
<code>_MC163_DEMASK_INT_11</code>	(= 0800h)	Demaskiert Interrupteingang INT-11
<code>_MC163_DEMASK_INT_12</code>	(= 1000h)	Demaskiert Interrupteingang INT-12
<code>_MC163_DEMASK_INT_13</code>	(= 2000h)	Demaskiert Interrupteingang INT-13
<code>_MC163_DEMASK_INT_14</code>	(= 4000h)	Demaskiert Interrupteingang INT-14
<code>_MC163_DEMASK_INT_15</code>	(= 8000h)	Demaskiert Interrupteingang INT-15

Zusätzlich muß noch eine ODER-Verknüpfung angegeben werden, wenn die negative Flanke an einem externen digitalen Eingang einen Interrupt auslösen soll (Standardeinstellung: positive Flanke):

<code>_MC163_INT_0_EDGE_NEG</code>	(= 0010000h)	Negative Flanke an INT-0 löst Interrupt aus
<code>_MC163_INT_1_EDGE_NEG</code>	(= 0020000h)	Negative Flanke an INT-1 löst Interrupt aus
<code>_MC163_INT_2_EDGE_NEG</code>	(= 0040000h)	Negative Flanke an INT-2 löst Interrupt aus
<code>_MC163_INT_3_EDGE_NEG</code>	(= 0080000h)	Negative Flanke an INT-3 löst Interrupt aus
<code>_MC163_INT_4_EDGE_NEG</code>	(= 0100000h)	Negative Flanke an INT-4 löst Interrupt aus
<code>_MC163_INT_5_EDGE_NEG</code>	(= 0200000h)	Negative Flanke an INT-5 löst Interrupt aus
<code>_MC163_INT_6_EDGE_NEG</code>	(= 0400000h)	Negative Flanke an INT-6 löst Interrupt aus
<code>_MC163_INT_7_EDGE_NEG</code>	(= 0800000h)	Negative Flanke an INT-7 löst Interrupt aus
<code>_MC163_INT_8_EDGE_NEG</code>	(= 1000000h)	Negative Flanke an INT-8 löst Interrupt aus
<code>_MC163_INT_9_EDGE_NEG</code>	(= 2000000h)	Negative Flanke an INT-9 löst Interrupt aus
<code>_MC163_INT_10_EDGE_NEG</code>	(= 4000000h)	Negative Flanke an INT-10 löst Interrupt aus
<code>_MC163_INT_11_EDGE_NEG</code>	(= 8000000h)	Negative Flanke an INT-11 löst Interrupt aus

Welcher Interrupteingang maskiert bzw. demaskiert ist und welche aktive Flanke eingestellt ist, kann durch einen Lesezugriff mit **mddx_read_channel_dword** auf den Kanal ermittelt werden.

Definition des Latch-Eingangs

Dieser Kanal ermöglicht es, festzulegen, welcher digitale Eingang zum Latchen aller digitalen Eingänge verwendet werden soll, wenn diese als solche verwendet werden sollen und nicht als Zähleringänge für die Zähler.

Ofs	Strukturelement	Datentyp	Wert
0	<i>.usDevice</i>	USHORT	Device-Typ: <i>DEVICE_CTRL</i> = 0401h
2	<i>.usIndexFirst</i>	USHORT	Device-Index (= 2)
4	<i>.usIndexLast</i>	USHORT	= <i>.usIndexFirst</i>
6	<i>.usFlags</i>	USHORT	Flags: Bit 0: <i>_CP_EXCLUSIVE</i> = 1 (exklusiv) Alle anderen Bits = 0.
8	<i>.usReadMode</i>	USHORT	Lesemodus: <i>IO_MODE_DIRECT</i> = 1 (direkt)
10	<i>.usWriteMode</i>	USHORT	Schreibmodus: <i>IO_MODE_DIRECT</i> = 1 (direkt)

Alle anderen Strukturelemente werden nicht ausgewertet.

Der Latch-Eingang wird durch einen Schreibzugriff mit **mddx_write_channel_byte** auf den Kanal definiert:

<i>_MC163_DIN_0_LATCH</i>	(= 00h)	DIN-0 ist Latch-Eingang
<i>_MC163_DIN_1_LATCH</i>	(= 01h)	DIN-1 ist Latch-Eingang
<i>_MC163_DIN_2_LATCH</i>	(= 02h)	DIN-2 ist Latch-Eingang
<i>_MC163_DIN_3_LATCH</i>	(= 03h)	DIN-3 ist Latch-Eingang
<i>_MC163_DIN_4_LATCH</i>	(= 04h)	DIN-4 ist Latch-Eingang
<i>_MC163_DIN_5_LATCH</i>	(= 05h)	DIN-5 ist Latch-Eingang
<i>_MC163_DIN_6_LATCH</i>	(= 06h)	DIN-6 ist Latch-Eingang
<i>_MC163_DIN_7_LATCH</i>	(= 07h)	DIN-7 ist Latch-Eingang
<i>_MC163_DIN_8_LATCH</i>	(= 08h)	DIN-8 ist Latch-Eingang
<i>_MC163_DIN_9_LATCH</i>	(= 09h)	DIN-9 ist Latch-Eingang
<i>_MC163_DIN_10_LATCH</i>	(= 0ah)	DIN-10 ist Latch-Eingang
<i>_MC163_DIN_11_LATCH</i>	(= 0bh)	DIN-11 ist Latch-Eingang

Zusätzlich muß noch eine ODER-Verknüpfung angegeben werden, wenn die negative Flanke den Latchvorgang auslösen soll und/oder der externe Latch-Eingang freigegeben werden soll:

<code>_MC163_LATCH_NEG_EDGE</code>	(= 20h)	Flanke zum Latchen: negativ
<code>_MC163_LATCH_POS_EDGE</code>	(= 00h)	Flanke zum Latchen: positiv
<code>_MC163_LATCH_ENABLE</code>	(= 40h)	Externer Latch-Eingang: enable
<code>_MC163_LATCH_DISABLE</code>	(= 00h)	Externer Latch-Eingang: disable

Welcher Eingang als Latch-Eingang definiert und freigegeben ist und welche Flanke für das Latchen verwendet wird, kann durch einen Lesezugriff mit `mddx_read_channel_byte` auf den Kanal ermittelt werden.

Senden eines Steuerkommandos an einen/mehrere Zähler

Dieser Kanal ermöglicht es, Kommandos an einen oder mehrere Zähler zu senden.

Ofs	Strukturelement	Datentyp	Wert
0	<code>.usDevice</code>	USHORT	Device-Typ: <code>DEVICE_CTRL</code> = 0401h
2	<code>.usIndexFirst</code>	USHORT	Device-Index (= 4)
4	<code>.usIndexLast</code>	USHORT	= <code>.usIndexFirst</code>
6	<code>.usFlags</code>	USHORT	Flags: Bit 0: <code>_CP_EXCLUSIVE</code> = 1 (exklusiv) Alle anderen Bits = 0.
8	<code>.usReadMode</code>	USHORT	Lesemodus: <code>IO_MODE_DIRECT</code> = 1 (direkt)
10	<code>.usWriteMode</code>	USHORT	Schreibmodus: <code>IO_MODE_DIRECT</code> = 1 (direkt)

Alle anderen Strukturelemente werden nicht ausgewertet.

Folgende Steuerkommandos können durch einen Schreibzugriff mit `mddx_write_channel_block` ausgeführt werden. Der Block kann max. 3 Byte groß sein, so daß bis zu 3 Kommandos gesendet werden können, d.h. an jeden Zähler kann auch getrennt ein Kommando geschickt werden. Wird eine 0 eingetragen, wird kein Kommando geschickt. Jedes dieses 3 Bytes kann folgende Werte annehmen:

<code>_MC163_CMD_START_COUNTER</code>	(= 1)	Start eines/mehrerer Zähler
<code>_MC163_CMD_STOP_COUNTER</code>	(= 2)	Stop eines/mehrerer Zähler
<code>_MC163_CMD_LOAD_COUNTER</code>	(= 4)	Laden eines/mehrerer Zähler
<code>_MC163_CMD_FREEZE_COUNTER</code>	(= 8)	Latchen eines/mehrerer Zähler
<code>_MC163_CMD_RESET_COUNTER</code>	(= 16)	Reset eines/mehrerer Zähler

An welchen Zähler das Kommando geschickt werden soll, kann durch eine ODER-Verknüpfung angegeben werden (die Zähler sind kombinierbar):

<code>_MC163_COUNTER_0</code>	(= 32)	Kommando an Zähler 0 senden
<code>_MC163_COUNTER_1</code>	(= 64)	Kommando an Zähler 1 senden
<code>_MC163_COUNTER_2</code>	(= 128)	Kommando an Zähler 2 senden

Lesen des Status eines/mehrerer Zähler

Dieser Kanal ermöglicht das Lesen des Status eines/mehrerer Zähler. Die Struktur ist identisch mit der auf Seite 15-97.

Ofs	Strukturelement	Datentyp	Wert
0	<code>.usDevice</code>	USHORT	Device-Typ: <code>DEVICE_CTRL</code> = 0401h
2	<code>.usIndexFirst</code>	USHORT	Device-Index (= 4)
4	<code>.usIndexLast</code>	USHORT	= <code>.usIndexFirst</code>
6	<code>.usFlags</code>	USHORT	Flags: Bit 0: <code>_CP_EXCLUSIVE</code> = 1 (exklusiv) Alle anderen Bits = 0.
8	<code>.usReadMode</code>	USHORT	Lesemodus: <code>IO_MODE_DIRECT</code> = 1 (direkt)
10	<code>.usWriteMode</code>	USHORT	Schreibmodus: <code>IO_MODE_DIRECT</code> = 1 (direkt)

Alle anderen Strukturelemente werden nicht ausgewertet.

Durch einen Lesezugriff mit **mddx_read_channel_block** (max. 3 Byte) wird der Status aller drei Zähler gelesen (Byte 1 = Status von Zähler 0, Byte 2 = Status von Zähler 1 und Byte 3 = Status von Zähler 2). Der jeweils zurückgelieferte Wert hat folgende Bedeutung:

Bit	Bedeutung
0	RUN: 1 = Zähler läuft 0 = Zähler stop
1	DIRECTION (falls RUN = 1): 0 = vorwärts 1 = rückwärts
2	PRESET: 1 = neuer Wert im Counter-Preset-Register wurde noch nicht übernommen 0 = neuer Wert des Counter-Preset-Register wurde in Zähler übernommen
3	FREEZE: 1 = neuer Wert wurde im Counter-Data-Register gespeichert, aber noch nicht gelesen 0 = Wert im Counter-Data-Register wurde bereits gelesen.
4	OVERRUN: 1 = Overrun aufgetreten 0 = kein Overrun aufgetreten
5-15	reserviert

Soll nur der Status von Zähler 0 gelesen werden, kann die Blocklänge mit 1 angegeben werden, soll der Status von Zähler 0 und 1 gelesen werden, kann die Blocklänge mit 2 angegeben werden.

Programmierung der Zähler

Ofs	Strukturelement	Datentyp	Bedeutung
0	<i>.usDevice</i>	USHORT	Device-Typ: <i>DEVICE_COUNTER</i> = 0601h
2	<i>.usIndexFirst</i>	USHORT	Index des ersten Device(= 0 bis 2)
4	<i>.usIndexLast</i>	USHORT	Index des letzten Device (= 0 bis 2)
6	<i>.usFlags</i>	USHORT	Flags: Bit 0: <i>_CP_EXCLUSIVE</i> = 1 Alle anderen Bits = 0.
8	<i>.usReadMode</i>	USHORT	Lesemodus: <i>IO_MODE_DIRECT</i> = 1 (direkt)
10	<i>.usWriteMode</i>	USHORT	Schreibmodus: <i>IO_MODE_DIRECT</i> = 1 (direkt)
12	<i>.usMode</i>	USHORT	Betriebsart und Unterbetriebsart (s.u.)
14	<i>.usCounterInput</i>	USHORT	Zähleingang 0, Zähleingang 1, Flanken (s.u.)
16	<i>.usCounterOutput</i>	USHORT	Zählerausgang
18	<i>.usCommandInput</i>	USHORT	Steuerung des Zählers: Start, Stop, Load, Freeze und Reset
20	<i>.usInterrupt</i>	USHORT	Definition der Interruptbedingungen (s.u.)
22	<i>.usTime</i>	USHORT	Referenzfrequenz/-zeit (nur Betriebsart 5, 9 bis 12) bzw. Verzögerungszeit (nur Betriebsart 6 bis 8)
24	<i>.usMultiple</i>	USHORT	Anzahl der Periodendauermessungen
26	<i>.usValueSize</i>	USHORT	Auflösung des Ergebnisses in Anzahl Byte: 2 Byte = 0..65535 (= Minimal-Auflösung = 16-Bit HW-Counter), 3 Byte = 0..16777215, ... allg.: 256 hoch (usValueSize)

Im Strukturelement *.usMode* muß die gewünschte Betriebsart angegeben werden. Die vom M-C16-3 unterstützten Betriebsarten sind:

Betriebsart	Bezeichnung	Funktion
0	<i>_MC163_INACTIVE</i>	Zähler nicht aktiv
1	<i>_MC163_COUNTER_UP</i>	Vorwärtszähler
2	<i>_MC163_COUNTER_DN</i>	Rückwärtszähler
3	<i>_MC163_COUNTER_UP_DN_A</i>	Vor-/Rückwärtszähler Typ A
4	<i>_MC163_COUNTER_UP_DN_B</i>	Vor-/Rückwärtszähler Typ B
5	<i>_MC163_TIMER</i>	Timer
6	<i>_MC163_INCREMENT_A</i>	Inkrementalgeber Typ A

Betriebsart	Bezeichnung	Funktion
7, 8	<i>_MC163_INCREMENT_B</i>	Inkrementalgeber Typ B (Zähler 0 für Impulse in Vorwärtsrichtung, Zähler 1 für Impulse in Rückwärtsrichtung)
9	<i>_MC163_FREQUENCY</i>	Frequenzmessung
10	<i>_MC163_PULSEWIDTH</i>	Pulsbreitenmessung
11	<i>_MC163_PERIOD</i>	Periodendauermessung
12	<i>_MC163_VELOCITY</i>	Geschwindigkeitsmessung
4096	<i>_MC163_MULTIPLE_PERIOD</i>	Fortlaufende Periodendauermessung

Die Unterbetriebsarten für Betriebsart 1, 2, 3 und 4:

32	<i>_MC163_AUTOSTART</i>	Autostart
----	-------------------------	-----------

Die Unterbetriebsarten für Betriebsart 5:

32	<i>_MC163_AUTOSTART</i>	Autostart
64	<i>_MC163_CONTINUE</i>	Continue (automatisches Reload nach Null-durchgang)
128	<i>_MC163_EXTERN</i>	Zähltakt kommt von externem Eingang

Die Unterbetriebsarten für Betriebsart 6:

32	<i>_MC163_AUTOSTART</i>	Autostart
0	<i>_MC163_SINGLE_A</i>	Flankenbewertung: Mode Single Phase A
64	<i>_MC163_SINGLE_B</i>	Flankenbewertung: Mode Single Phase B
128	<i>_MC163_DOUBLE_A</i>	Flankenbewertung: Mode Double Phase A
192	<i>_MC163_DOUBLE_B</i>	Flankenbewertung: Mode Double Phase B
256	<i>_MC163_QUADRUPLE</i>	Flankenbewertung: Mode Quadruple

Die Unterbetriebsarten für Betriebsart 7 und 8:

0	<i>_MC163_SINGLE_A</i>	Flankenbewertung: Mode Single Phase A
64	<i>_MC163_SINGLE_B</i>	Flankenbewertung: Mode Single Phase B
128	<i>_MC163_DOUBLE_A</i>	Flankenbewertung: Mode Double Phase A
192	<i>_MC163_DOUBLE_B</i>	Flankenbewertung: Mode Double Phase B
256	<i>_MC163_QUADRUPLE</i>	Flankenbewertung: Mode Quadruple

Die Unterbetriebsarten für Betriebsart 9:

32	<i>_MC163_AUTOSTART</i>	Autostart
64	<i>_MC163_CONTINUE</i>	Continue (automatisches Reload nach Null-durchgang)
128	<i>_MC163_EXTERN</i>	Gate kommt von extern
256	<i>_MC163_AUTOFREEZE</i>	Automatisches Freeze am Ende einer Messung
512	<i>_MC163_AUTORESET</i>	Automatisches Rücksetzen des Zählers am Ende einer Messung
1024	<i>_MC163_AUTOLOAD</i>	Automatisches Load am Ende einer Messung
2048	<i>_MC163_OVERRIDE</i>	Der alte Zählerwert wird bei Freeze überschrieben

Die Unterbetriebsarten für Betriebsart 10, 11 und 12:

32	<i>_MC163_AUTOSTART</i>	Autostart
64	<i>_MC163_CONTINUE</i>	Continue (automatisches Reload nach Null-durchgang)
128	<i>_MC163_EXTERN</i>	Referenzfrequenz kommt von extern (nur bei Betriebsart 10 und 11)
256	<i>_MC163_AUTOFREEZE</i>	Automatisches Freeze am Ende einer Messung
512	<i>_MC163_AUTORESET</i>	Automatisches Rücksetzen des Zählers am Ende einer Messung
1024	<i>_MC163_AUTOLOAD</i>	Automatisches Load am Ende einer Messung
2048	<i>_MC163_OVERRIDE</i>	Der alte Zählerwert wird bei Freeze überschrieben

Zusätzlich kann angegeben werden, ob Zähler kaskadiert werden sollen:

8192	<i>_MC163_CASCADE_0_1</i>	Zähler 0 und Zähler 1 kaskadieren
16384	<i>_MC163_CASCADE_1_2</i>	Zähler 1 und Zähler 2 kaskadieren
32768	<i>_MC163_CASCADE_0_1_2</i>	Zähler 0 bis 2 kaskadieren

Im Strukturelement *.usCounterInput* muß angegeben werden, welche digitalen Eingänge mit Zählleitungen eines Zählers verbunden werden sollen. Die vom M-C16-3 unterstützten Verbindungen sind:

<i>_MC163_DIN_0_1_CNT_0</i>	= 0	Zähleingang 0 von Zähler 0 mit externem Eingang DIN-0 verbinden, Zähleingang 1 von Zähler 0 mit externem Eingang DIN-1 verbinden
<i>_MC163_DIN_2_3_CNT_0</i>	= 4096	Zähleingang 0 von Zähler 0 mit externem Eingang DIN-2 verbinden, Zähleingang 1 von Zähler 0 mit externem Eingang DIN-3 verbinden
<i>_MC163_DIN_4_5_CNT_0</i>	= 8192	Zähleingang 0 von Zähler 0 mit externem Eingang DIN-4 verbinden, Zähleingang 1 von Zähler 0 mit externem Eingang DIN-5 verbinden
<i>_MC163_DIN_6_7_CNT_0</i>	= 12288	Zähleingang 0 von Zähler 0 mit externem Eingang DIN-6 verbinden, Zähleingang 1 von Zähler 0 mit externem Eingang DIN-7 verbinden
<i>_MC163_DIN_8_9_CNT_0</i>	= 16384	Zähleingang 0 von Zähler 0 mit externem Eingang DIN-8 verbinden, Zähleingang 1 von Zähler 0 mit externem Eingang DIN-9 verbinden
<i>_MC163_DIN_10_11_CNT_0</i>	= 20480	Zähleingang 0 von Zähler 0 mit externem Eingang DIN-10 verbinden, Zähleingang 1 von Zähler 0 mit externem Eingang DIN-11 verbinden
<i>_MC163_DIN_6_CNT_1</i>	= 24576	Zähleingang 0 von Zähler 1 mit externem Eingang DIN-6 verbinden
<i>_MC163_CNT_0_CNT_1</i>	= 28672	Zähleingang 0 von Zähler 1 mit Zähleingang 1 von Zähler 0 verbinden
<i>_MC163_DIN_4_CNT_2</i>	= 32768	Zähleingang 0 von Zähler 2 mit externem Eingang DIN-4 verbinden
<i>_MC163_CNT_0_CNT_2</i>	= 36864	Zähleingang 0 von Zähler 2 mit Zähleingang 0 von Zähler 0 verbinden

Zusätzlich kann noch eine ODER-Verknüpfung angegeben werden, wenn das Eingangssignal am externen digitalen Eingang invertiert werden soll:

<code>_MC163_DIN_0_INV</code>	= 1	Eingang DIN-0 invertieren/aktive Flanke ist negativ
<code>_MC163_DIN_1_INV</code>	= 2	Eingang DIN-1 invertieren/aktive Flanke ist negativ
<code>_MC163_DIN_2_INV</code>	= 4	Eingang DIN-2 invertieren/aktive Flanke ist negativ
<code>_MC163_DIN_3_INV</code>	= 8	Eingang DIN-3 invertieren/aktive Flanke ist negativ
<code>_MC163_DIN_4_INV</code>	= 16	Eingang DIN-4 invertieren/aktive Flanke ist negativ
<code>_MC163_DIN_5_INV</code>	= 32	Eingang DIN-5 invertieren/aktive Flanke ist negativ
<code>_MC163_DIN_6_INV</code>	= 64	Eingang DIN-6 invertieren/aktive Flanke ist negativ
<code>_MC163_DIN_7_INV</code>	= 128	Eingang DIN-7 invertieren/aktive Flanke ist negativ
<code>_MC163_DIN_8_INV</code>	= 256	Eingang DIN-8 invertieren/aktive Flanke ist negativ
<code>_MC163_DIN_9_INV</code>	= 512	Eingang DIN-9 invertieren/aktive Flanke ist negativ
<code>_MC163_DIN_10_INV</code>	= 1024	Eingang DIN-10 invertieren/aktive Flanke ist negativ
<code>_MC163_DIN_11_INV</code>	= 2048	Eingang DIN-11 invertieren/aktive Flanke ist negativ

Im Strukturelement `.usCounterOutput` muß angegeben werden, ob ein digitaler Ausgang mit einem Zähler verbunden ist (**dieser Ausgang steht dann nicht mehr als einfacher Ausgang zur Verfügung!**):

<code>_MC163_DOUT_0_CNT_0</code>	= 1	DOUT-0 wird als Ausgang für Zähler 0 benutzt
<code>_MC163_DOUT_1_CNT_1</code>	= 2	DOUT-1 wird als Ausgang für Zähler 1 benutzt
<code>_MC163_DOUT_2_CNT_2</code>	= 4	DOUT-2 wird als Ausgang für Zähler 2 benutzt

Im Strukturelement `.usCommandInput` muß die Konfiguration der Steuereingänge Start, Stop, Load, Freeze und Reset definiert werden.

Unterstützte Steuerfunktionen des M-C16-3 für Steuereingang **Start** sind:

<code>_MC163_NO_START</code>	= 0	Start wird nicht über externen Eingang gesteuert
<code>_MC163_START_DIN_0</code>	= 16	Start wird über externen Eingang DIN-0 gesteuert
<code>_MC163_START_DIN_1</code>	= 32	Start wird über externen Eingang DIN-1 gesteuert
<code>_MC163_START_DIN_2</code>	= 48	Start wird über externen Eingang DIN-2 gesteuert

Unterstützte Steuerfunktionen des M-C16-3 für Steuereingang **Stop** sind:

<i>_MC163_NO_STOP</i>	= 0	Stop wird nicht über externen Eingang gesteuert
<i>_MC163_STOP_DIN_1</i>	= 64	Stop wird über externen Eingang DIN-1 gesteuert
<i>_MC163_STOP_DIN_2</i>	= 128	Stop wird über externen Eingang DIN-2 gesteuert
<i>_MC163_STOP_DIN_3</i>	= 192	Stop wird über externen Eingang DIN-3 gesteuert
<i>_MC163_NO_STOP_UFL</i>	= 0	Stop wird nicht bei Unterlauf des Zählers 0 ausgelöst
<i>_MC163_STOP_UFL_CNT_0</i>	= 256	Stop wird bei Unterlauf des Zählers 0 ausgelöst
<i>_MC163_STOP_UFL_CNT_1</i>	= 512	Stop wird bei Unterlauf des Zählers 0 ausgelöst
<i>_MC163_STOP_UFL_CNT_2</i>	= 768	Stop wird bei Unterlauf des Zählers 0 ausgelöst

Unterstützte Steuerfunktionen des M-C16-3 für Steuereingang **Load** sind:

<i>_MC163_NO_LOAD</i>	= 0	Load wird nicht über externen Eingang gesteuert
<i>_MC163_LOAD_DIN_0</i>	= 1024	Load wird über externen Eingang DIN-0 gesteuert
<i>_MC163_LOAD_DIN_1</i>	= 2048	Load wird über externen Eingang DIN-1 gesteuert
<i>_MC163_LOAD_DIN_2</i>	= 3072	Load wird über externen Eingang DIN-2 gesteuert

Unterstützte Steuerfunktionen des M-C16-3 für Steuereingang **Freeze** sind:

<i>_MC163_NO_FREEZE</i>	= 0	Freeze wird nicht über externen Eingang gesteuert
<i>_MC163_FREEZE_DIN_1</i>	= 4096	Freeze wird über externen Eingang DIN-1 gesteuert
<i>_MC163_FREEZE_DIN_2</i>	= 8192	Freeze wird über externen Eingang DIN-2 gesteuert
<i>_MC163_FREEZE_DIN_3</i>	= 12288	Freeze wird über externen Eingang DIN-3 gesteuert

Unterstützte Steuerfunktionen des M-C16-3 für Steuereingang **Reset** sind:

<i>_MC163_NO_RESET</i>	= 0	Reset wird nicht über externen Eingang gesteuert
<i>_MC163_RESET_DIN_0</i>	= 16384	Reset wird über externen Eingang DIN-0 gesteuert
<i>_MC163_RESET_DIN_1</i>	= 32768	Reset wird über externen Eingang DIN-1 gesteuert
<i>_MC163_RESET_DIN_2</i>	= 49152	Reset wird über externen Eingang DIN-2 gesteuert

Zusätzlich kann noch eine ODER-Verknüpfung angegeben werden, wenn das Eingangssignal am externen Eingang invertiert werden soll:

<code>_MC163_DIN_0_INV</code>	= 1	Eingangssignal an DIN-0 invertieren.
<code>_MC163_DIN_1_INV</code>	= 2	Eingangssignal an DIN-1 invertieren.
<code>_MC163_DIN_2_INV</code>	= 4	Eingangssignal an DIN-2 invertieren.
<code>_MC163_DIN_3_INV</code>	= 8	Eingangssignal an DIN-3 invertieren.

Die einzelnen Steuerfunktionen für Start, Stop, Load, Freeze und Reset können durch eine ODER-Verknüpfung verbunden werden.

Im Strukturelement `.usInterrupt` muß angegeben werden, welche Bedingung, z.B. Unter-/Überlauf eines Zählers, einen Interrupt auslösen soll. Die vom M-C16-3 unterstützten Interruptbedingungen für die Zähler sind:

<code>_MC163_NO_INT</code>	(= 0)	Keinen Interrupt auslösen
<code>_MC163_INT_CNT_0_UFL</code>	(= 1)	Interrupt auslösen, wenn Unterlauf des Zählers 0
<code>_MC163_INT_CNT_0_OVL</code>	(= 2)	Interrupt auslösen, wenn Überlauf des Zählers 0
<code>_MC163_INT_CNT_0_END</code>	(= 4)	Interrupt auslösen, wenn Messung zu Ende
<code>_MC163_INT_CNT_0_INVALID</code>	(= 8)	Interrupt auslösen, wenn Messung ungültig
<code>_MC163_INT_CNT_1_UFL</code>	(= 16)	Interrupt auslösen, wenn Unterlauf des Zählers 1
<code>_MC163_INT_CNT_1_OVL</code>	(= 32)	Interrupt auslösen, wenn Überlauf des Zählers 1
<code>_MC163_INT_CNT_1_END</code>	(= 64)	Interrupt auslösen, wenn Messung zu Ende
<code>_MC163_INT_CNT_2_UFL</code>	(= 128)	Interrupt auslösen, wenn Unterlauf des Zählers 2
<code>_MC163_INT_CNT_2_OVL</code>	(= 256)	Interrupt auslösen, wenn Überlauf des Zählers 2
<code>_MC163_INT_CNT_2_END</code>	(= 512)	Interrupt auslösen, wenn Messung zu Ende

Die einzelnen Interruptbedingungen können durch eine ODER-Verknüpfung verbunden werden.

Im Strukturelement `.usTime` muß eine Referenzfrequenz/Referenzzeit angegeben werden, wenn eine der Betriebsarten Timer, Frequenzmessung, Pulsbreitenmessung, Periodendauermessung oder Geschwindigkeitsmessung in Strukturelement `.usMode` angegeben wird bzw. eine Verzögerungszeit, wenn eine Betriebsart mit dem Inkrementalgeber angegeben wird.

Wert	Bezeichnung	Referenzfrequenz	Bezeichnung	Referenzzeit/ Verzögerungszeit
0	<i>MC163_NO_FREQ</i>	Keine	<i>MC163_NO_PERIOD</i>	Keine
2	<i>MC163_10_MHZ</i>	10 MHz	<i>MC163_100_NS</i>	100 ns
3	<i>MC163_5_MHZ</i>	5 MHz	<i>MC163_200_NS</i>	200 ns
4	<i>MC163_2_MHZ</i>	2 MHz	<i>MC163_500_NS</i>	500 ns
5	<i>MC163_1_MHZ</i>	1 MHz	<i>MC163_1_US</i>	1 µs
6	<i>MC163_500_KHZ</i>	500 KHz	<i>MC163_2_US</i>	2 µs
7	<i>MC163_200_KHZ</i>	200 KHz	<i>MC163_5_US</i>	5 µs
8	<i>MC163_100_KHZ</i>	100 KHz	<i>MC163_10_US</i>	10 µs
9	<i>MC163_50_KHZ</i>	50 KHz	<i>MC163_20_US</i>	20 µs
10	<i>MC163_20_KHZ</i>	20 KHz	<i>MC163_50_US</i>	50 µs
11	<i>MC163_10_KHZ</i>	10 KHz	<i>MC163_100_US</i>	100 µs
12	<i>MC163_5_KHZ</i>	5 KHz	<i>MC163_200_US</i>	200 µs
13	<i>MC163_2_KHZ</i>	2 KHz	<i>MC163_500_US</i>	500 µs
14	<i>MC163_1_KHZ</i>	1 KHz	<i>MC163_1_MS</i>	1 ms
15	<i>MC163_500_HZ</i>	500 Hz	<i>MC163_2_MS</i>	2 ms
15	<i>MC163_200_HZ</i>	200 Hz	<i>MC163_5_MS</i>	5 ms
17	<i>MC163_100_HZ</i>	100 Hz	<i>MC163_10_MS</i>	10 ms
18	<i>MC163_50_HZ</i>	50 Hz	<i>MC163_20_MS</i>	20 ms
19	<i>MC163_20_HZ</i>	20 Hz	<i>MC163_50_MS</i>	50 ms
20	<i>MC163_10_HZ</i>	10 Hz	<i>MC163_100_MS</i>	100 ms

Lesen:

Das Lesen der Zähler erfolgt durch einen Lesezugriff auf den Kanal mit **mddx_read_channel_block**. Die maximale Länge resultiert aus der Bitbreite des ausgewählten Zählers (16 Bit: BLOCK aus 2 Byte, 32 Bit: BLOCK aus 4 Byte, 48 Bit: BLOCK aus 6 BYTE) bzw. der gewünschten Auflösung (CPS-Element *usValue-Size*) und der verwendeten Betriebsart (Multiple Periodendauermessung, Zähler kaskadiert: BLOCK aus 6 Byte, Multiple Periodendauermessung, nicht kaskadiert: BLOCK aus 4 Byte, Counter 0 und 1 bzw. 1 und 2 kaskadiert: BLOCK aus 4 Byte, alle Counter kaskadiert: BLOCK aus 6 Byte, Incremental Mode B: BLOCK aus 4 Byte).

Wenn als *usReadMode = IO_MODE_LATCH* gewählt wurde, muß der Kanal vorher per Software oder von extern gelatched werden.

Schreiben:

Durch einen entsprechenden Schreibzugriff auf den Kanal kann das Device auf einen Anfangswert vorge setzt werden. Wenn *usWriteMode = IO_LATCH*, dann erfolgt die Übernahme erst nach einem Zugriff per Software über DEVICE_CTRL und Device-Index = 4 oder von extern.

Status:

Über **mddx_get_channel_info (...INFO_DEVICE)** kann der Status einer Messung (z.B. Pulsbreitenmessung) gelesen werden, Bit 0 bis Bit 4: Info für Zähler 0, Bit 8 bis Bit 12: Info für Zähler 1 und Bit 16 bis Bit 20: Info für Zähler 2.

Dabei bedeuteten

- Bit 0, 8, 16: Messung beendet
- Bit 1, 9, 17: Overflow aufgetreten
- Bit 2, 10, 18: Underflow aufgetreten
- Bit 3: Ungültige Messung, Bit 11, 19: Reserviert
- Bit 4, 12, 20: Ergebnis der letzten Messung nicht abgeholt.

Grenzwerte:

Über **mddx_send_channel_control_block** können Grenzwerte für die Messungen festgelegt werden: mit *usCtrl = 0x11* wird der Minimal-Wert festgelegt und mit *usCtrl = 0x22* der Maximal-Wert. PCtrlData enthält dabei den eigentlichen Wert und *usSize* die Anzahl der Bytes die gesetzt werden sollen (muß = CPS-Element *usValueSize* sein).

Interrupt:

Wenn für z.B. für Unterlauf oder Überlauf Interrupts ausgelöst werden sollen (*usInterrupt* \neq 0), werden interne Routinen aufgerufen, die dieses Ereignis festhalten (siehe Status) und die, wenn das CPS-Element *usValueSize* > HW-Counter ist, in Software die Anzahl der aufgetretenen Ereignisse mitzählen.

Beim Lesen des Ergebnisses wird dann Hardware- und Software-Resultat zurückgeliefert: bei einem Zähler (=16 Bit) und *usValueSize = 3* also Bit 0 bis 15 = Hardware-Resultat und Bit 16 bis 23 = Software-Resultat.

Bei Erreichen des Grenzwertes (Minimum oder Maximum) wird das „Zwischen“-Ergebnis in das Endergebnis übertragen und eine Anwenderprozedur aufgerufen bzw. ein Service-Request zum PC ausgelöst (default).

Message-Modul-Device-Treiber:

Unter Verwendung des Message-Modul-Device-Treibers kann auch die Funktion **mddx_open_channel_call_back** verwendet werden, der der Anwender die Adresse ei-

ner aufzurufenden Funktion übergeben kann. Diese Funktion wird dann, z.B. durch das Auslösen eines Interrupts aufgerufen.

Das Öffnen eines Modul-MDD-Kanals mit Callback-Funktionalität, geschieht über die neue Bibliotheksfunktion

```
HMDD8 mdd_open_channel_callback (USHORT slot,
                                  USHORT size,
                                  void *pModulCPS,
                                  ULONG param,
                                  void *pCallbackFunction,
                                  USHORT mode)
```

Die ersten drei Parameter entsprechen denen von *mdd_open_channel* und werden intern direkt für das Öffnen eines Kanals zum Modul-MDD verwendet. Dessen Handle liefert die Funktion als Rückgabewert.

param ist zur Zeit noch nicht verwendet (=0 setzen).

Im Parameter *pCallbackFunction* gibt der Anwender seine Callback-Prozedur an. Sie muß folgenden Aufbau haben:

```
void UserCallback (HMDD8 handle, ULONG param, ULONG size, void *pData);
```

Auf OsX-Seite ist die Prozedur mit far pascal Aufrufkonvention zu compilieren. Dort ist die Funktion mit dem Makro *mddx_entry_callback* zu beginnen, das das Daten-segment-Register setzt. Am Ende der Funktion muß dieses mit *mddx_exit_callback* restauriert werden. Der Kompatibilität wegen stehen diese Makros auch in der PC-Bibliothek zur Verfügung, sie müssen dort allerdings nicht aufgerufen werden.

Die an die Prozedur übergebenen Parameter haben folgende Bedeutung:

1. handle Handle des Kanals, der den Callback-Vorgang in Gang gesetzt hat.
2. param reserviert (=0)
3. size Länge der übergebenen Nutzdaten (Anzahl Bytes)
4. pData Zeiger auf die übergebenen Nutzdaten. Das 1. Byte enthält den Inhalt des Interrupt-Vektor-Registers, das 2. Byte des Status der Messung, danach folgen die Meßdaten, z.B. die Periodendauer. Die Auflösung (Anzahl Byte) ergibt sich aus $size-2$.

Befindet sich die Callback-Prozedur auf der Echtzeitseite, ist zu beachten, daß die Speicheradresse der Nutzdaten als physikalische Adresse übergeben wird, so daß über das OsX auf die Daten zugegriffen werden muß. Damit muß die Prozedur auf OsX-Seite in „C“ z.B. wie folgt aussehen:

```
void MyProc (HMDD8 handle, ULONG param, ULONG size, void *pData)
{
    USHORT DataBuffer[MAX_BUFFER_SIZE];

    mddx_entry_callback();
    mlxrt_read_ram(size, *(ULONG*)pData, DataBuffer);
    // jetzt kann mit den im DataBuffer-Array stehenden Daten gearbeitet
    // werden ...
    mddx_exit_callback();
}
```

Im Parameter *mode* kann durch das Flag **CBP_SYNC_CALL** angegeben werden, daß die Routine direkt (ohne Zwischenpufferung) beim Auftreten des Ereignisses aufgerufen wird.

Andernfalls puffert der Message-MDD die Daten, die der Callback-Prozedur übergeben werden sollen, zwischen. Die Entnahme erfolgt vom Betriebssystem gesteuert. Diese Einstellung ist nur möglich, wenn sich MDD und die Callback-Prozedur auf derselben CPU befinden.

Beispiele:

Nachfolgend einige Beispiele, wie die Strukturelemente bei unterschiedlichen Konfigurationen initialisiert werden müssen:

Beispiel 1: Vorwärtszähler

Verwendet wird der Zähler 0 und der digitale Eingang DIN-2. Der Zähler soll über eine positive Flanke am digitalen Eingang DIN-0 gestartet werden und ein Überlauf soll einen Interrupt erzeugen.

Initialisierung der Strukturelemente:

```
.usDevice =          DEVICE_COUNTER
.usIndexFirst =      0
.usIndexLast =       0
.usFlags =           0
.usReadMode =        IO_MODE_DIRECT
.usWriteMode =        IO_MODE_DIRECT
.usMode =            _MC163_COUNTER_UP
.usCounterInput =    _MC163_DIN_2_3_CNT_0
.usCommandInput =    _MC163_START_DIN_0+
                    _MC163_NO_STOP+
                    _MC163_NO_STOP_UFL+
                    _MC163_NO_LOAD+
                    _MC163_NO_FREEZE+
                    _MC163_NO_RESET+
                    _MC163_DIN_0_INV
.usInterrupt =       _MC163_INT_CNT_0_OVL
.usTime =            0
.usValueSize =       2
```

Beispiel 2: Rückwärtszähler

Verwendet wird der Zähler 2 und der digitale Eingang DIN-4. Der Zähler soll über eine negative Flanke am digitalen Eingang DIN-2 bzw. bei Unterlauf von Zähler 0 angehalten werden und bei Unterlauf soll kein Interrupt erzeugt werden. Der Zähler soll auf einen Wert vorge setzt werden, der automatisch übernommen werden soll.

Initialisierung der Strukturelemente:

```

.usDevice =          DEVICE_COUNTER
.usIndexFirst =      2
.usIndexLast =       2
.usFlags =           0
.usMode =            _MC163_COUNTER_DN+
                    _MC163_AUTOSTART
.usCounterInput =    _MC163_DIN_4_CNT_2
.usCommandInput =    _MC163_STOP_DIN_2+
                    _MC163_STOP_UFL_CNT_0+
                    _MC163_NO_START+
                    _MC163_NO_LOAD+
                    _MC163_NO_FREEZE+
                    _MC163_NO_RESET+
                    _MC163_DIN_2_INV
.usInterrupt =        _MC163_NO_INTERRUPT
.usTime =            0
.usValueSize =       2

```

Beispiel 3: Frequenzmessung

Verwendet werden sollen Zähler 0 und Zähler 1 (kaskadiert), die digitalen Eingänge DIN-0 und DIN-1 sollen Zähleringänge sein. Die Referenzzeit soll 1 ms betragen. Am Ende der Messung soll der Zählerstand automatisch in das Latch übertragen werden und der Zähler wieder zurückgesetzt werden. Das Ende der Messung soll einen Interrupt auslösen.

Initialisierung der Strukturelemente:

```

.usDevice =          DEVICE_COUNTER
.usIndexFirst =      0
.usIndexLast =       1
.usFlags =           0
.usReadMode =        IO_MODE_DIRECT
.usWriteMode =       IO_MODE_DIRECT
.usMode =            _MC163_FREQUENCY+
                    _MC163_CASCADE_0_1+
                    _MC163_AUTOFREEZE+
                    _MC163_AUTORESET
.usCounterInput =    _MC163_DIN_0_1_CNT_0
.usCommandInput =    _MC163_NO_CTRL
.usInterrupt =        _MC163_INT_CNT_0_END

```

```
.usTime =          _MC163_1_MS
.usValueSize =     4
```

Beispiel 4: Inkrementalgeberinterface

Verwendet werden soll der Zähler 0, die digitalen Eingänge DIN-4 und DIN-5 sollen Zähleringänge sein. Die Flanken sollen vierfach ausgewertet werden. Die Referenzzeit soll 100 ns betragen. Gestartet und geladen werden soll der Zähler über den externen Eingang DIN-0, angehalten durch den externen Eingang DIN-1, gelatched durch den externen Eingang DIN-3 und zurückgesetzt durch den externen Eingang DIN-2. Ein Interrupt soll bei Über- und Unterlauf des Zählers ausgelöst werden.

```
.usDevice =        DEVICE_COUNTER
.usIndexFirst =    0
.usIndexLast =     0
.usFlags =         0
.usReadMode =     IO_MODE_DIRECT
.usWriteMode =    IO_MODE_DIRECT
.usMode =         _MC163_INCREMENT_A+
                 _MC163_QUADRUPLE
.usCounterInput = _MC163_DIN_4_5_CNT_0+
                 _MC163_DIN_4_INV
.usCommandInput = _MC163_START_DIN_0+
                 _MC163_STOP_DIN_1+
                 _MC163_LOAD_DIN_0+
                 _MC163_FREEZE_DIN_3+
                 _MC163_RESET_DIN_2
.usInterrupt =    _MC163_INT_CNT_0_OVL+
                 _MC163_INT_CNT_0_UFL
.usTime =         _MC163_100_NS
.usValueSize =    2
```

Programmierung der digitalen Ausgänge

Das Modul M-C16-3 besitzt 4 digitale Ausgänge (DOUT-0 bis DOUT-3), die direkt gesetzt werden können, wenn sie nicht als Zählerausgänge verwendet werden.

Ofs	Strukturelement	Datentyp	Bedeutung
0	<i>.usDevice</i>	USHORT	Device-Typ: <i>DEVICE_DOUT</i> = 0102h
2	<i>.usIndexFirst</i>	USHORT	Index des ersten Device(= 0 bis 3)
4	<i>.usIndexLast</i>	USHORT	Index des letzten Device (= 0 bis 3)
6	<i>.usFlags</i>	USHORT	Flags: Bit 0: <i>_CP_EXCLUSIVE</i> = 1 (exklusiv) Alle anderen Bits = 0.
8	<i>.usReadMode</i>	USHORT	Lesemodus: <i>IO_MODE_RAM</i> = 4 (aus RAM)
10	<i>.usWriteMode</i>	USHORT	Schreibmodus: <i>IO_MODE_DIRECT</i> = 1 (direkt)

Alle anderen Strukturelemente müssen = 0 gesetzt werden.

Das Setzen des Ausgänge erfolgt durch einen Schreibzugriff mit **mdx_write_channel_byte** auf den Kanal. Durch einen entsprechenden Lesezugriff kann der aktuell ausgegebene Wert ermittelt werden.

Programmierung der digitalen Eingänge

Das Modul M-C16-3 besitzt 12 digitale Eingänge (DIN-0 bis DIN-11), die direkt gelesen werden können, wenn sie nicht als Zählereingänge verwendet werden.

Ofs	Strukturelement	Datentyp	Bedeutung
0	<i>.usDevice</i>	USHORT	Device-Typ: <i>DEVICE_DIN</i> = 0101h
2	<i>.usIndexFirst</i>	USHORT	Index des ersten Device(= 0 bis 11)
4	<i>.usIndexLast</i>	USHORT	Index des letzten Device (= 0 bis 11)
6	<i>.usFlags</i>	USHORT	Flags: Bit 0: <i>_CP_EXCLUSIVE</i> = 1 (exklusiv) Alle anderen Bits = 0.
8	<i>.usReadMode</i>	USHORT	Lesemodus: <i>IO_MODE_DIRECT</i> = 1 (direkt) <i>IO_MODE_LATCH</i> = 2 (aus Latch)

Alle anderen Strukturelemente müssen = 0 gesetzt werden.

Das Lesen der Eingänge erfolgt je nach Anzahl Devices durch einen Lesezugriff mit **mddx_read_channel_byte** bzw. **mddx_read_channel_word** (>8 Devices) auf den Kanal.

Programmierung der LEDs

Ofs	Strukturelement	Datentyp	Bedeutung
0	<i>.usDevice</i>	USHORT	Device-Typ: <i>DEVICE_LED</i> = 0402h
2	<i>.usIndexFirst</i>	USHORT	Index des ersten Device(= 0 bis 1)
4	<i>.usIndexLast</i>	USHORT	Index des letzten Device (= 0 bis 1)
6	<i>.usFlags</i>	USHORT	Flags: Bit 0: <i>_CP_EXCLUSIVE</i> = 1 (exklusiv) Alle anderen Bits = 0.
8	<i>.usReadMode</i>	USHORT	Lesemodus: <i>IO_MODE_RAM</i> = 4 (aus RAM)
10	<i>.usWriteMode</i>	USHORT	Schreibmodus: <i>IO_MODE_DIRECT</i> = 1 (direkt)

Alle anderen Strukturelemente müssen = 0 gesetzt werden.

Das Setzen der LEDs erfolgt durch einen Schreibzugriff mit **mddx_write_channel_byte** auf den Kanal, folgende Parameterwerte sind möglich:

<i>_MC163_NO_LED</i> = 0	Keine LED einschalten
<i>_MC163_LED_0</i> = 1	LED-0 einschalten
<i>_MC163_LED_1</i> = 2	LED-1 einschalten

Durch einen Lesezugriff mit **mddx_read_channel_byte** kann die aktuell eingeschaltete LED ermittelt werden.