AN024

Application Note 024 zu MODULAR-4 und Multi-LAB

Einsatz einer MODULAR-4 bzw. Multi-LAB Karte als stand-alone System mit Ankopplung an einen beliebigen Host-Rechner über RS-232

Autor: HB AN024.DOC (17 Seiten)

Kapitel	Inhalt
1	Definition
2	Aufgabenstellung
3	Hardware
3.1	Bus-Stecker für PC-Bus (ML1, ML4 und ML5)
3.2	Versorgungsspannung für Micro-Channel Karten (ML3)
3.3	Anschluß der seriellen Schnittstelle
3.3.3	Verbindungskabel zwischen PC und Karte
3.3.4	Hardware Reset über DTR (Reset-Logik)
4	Software
4.1	Beschreibung des PC-Software-Interrupts INT 63H
4.2	Einbindung des PC-Software-Interrupts INT 63H in bestehende PC-Bibliotheken für die MODULAR-4
4.3	Einbindung des PC-Software-Interrupts INT 63H in bestehende PC-Bibliotheken für die Multi-LAB

1. Definition

Im folgenden wie auch in allen anderen Sorcus Schriften werden folgende Kurzbezeichnungen verwendet:

Kurzbezeichnung	Karte	Bussystem	PC
ML1	Multi-LAB	PC	IBM-PC
ML3	MODULAR-4/Z280	Micro-Channel	PS/2
ML4	MODULAR-4/Z80	PC	IBM-PC
ML5	MODULAR-4/Z280	PC	IBM-PC
ML6	MODULAR-4/386	PC/PC-AT	IBM-PC
ML8	MODULAR-4/486	?	?

2. Aufgabenstellung

Üblicherweise steckt eine MODULAR-4 Karte in einem PC. Programme und Daten werden über den PC-Bus geladen. Da jede MODULAR-4 Karte über einen eigenen Prozessor verfügt, kann sie aber auch außerhalb des PC wie jeder andere Single-board Computer für Meß-, Steuerungs- und Kommunikationsaufgaben eingesetzt werden.

Sie kann dazu z.B. in einem separaten Gehäuse mit eigener Spannungsversorgung (+ 5 Volt und gegebenenfalls zusätzlich +/- 12 Volt) untergebracht werden und dann mit einem Host-Rechner über eine serielle RS-232 Schnittstelle der Basiskarte kommunizieren.

Eine besondere Einsatzmöglichkeit besteht für die ML3-Karte (mit Micro-Channel Bus):

Die Karte kann, auch während sie in einem PC steckt, mit externer Versorgungsspannung betrieben werden. Sie kann deshalb auch weiterarbeiten, falls der PC ausfällt. Dies ist möglich, weil auf der Karte zusätzliche Logik vorhanden ist, die in diesem Fall den Bus von der Karte abtrennt.

Als Host-Rechner kann z.B. ein IBM-kompatibler PC dienen, als Schnittstelle eine der COM1- bis COM4-Schnittstellen, die über ein entsprechendes Kabel mit einer der seriellen Schnittstellen der MODULAR-4 Basiskarte verbunden wird. Bei ML3 und ML5 muß die serielle Schnittstelle A verwendet werden. Im EEPROM der Basiskarte von MODULAR-4 Karte muß ein bestimmtes Bit gesetzt werden (z.B. bei ML4 das Bit 0 von Wort-7). Danach werden alle Zeichen, die über die serielle Schnittstelle empfangen werden, als Makro-Befehle interpretiert.

Bei ML1 und früheren Versionen der MODULAR-4 Karten ist ein spezielles Betriebssystem im EPROM erforderlich. Das ist jetzt bei MODULAR-4 Karten nicht mehr nötig.

Die Grundeinstellung der seriellen Schnittstelle der Basiskarte kann bei MODULAR-4 Karten ebenfalls im EEPROM der Basiskarte vorgenommen werden. Als Standardeinstellung ist dabei 9600 Baud, 8 Bit, No Parity und 1 Stop-Bit vorgesehen. Dies gilt auch für die ML1 Karte, ist dort aber fest eingestellt und nicht Anwender-seitig änderbar.

3. Hardware

Wenn eine Karte als Stand-alone System eingesetzt wird, muß unbedingt darauf geachtet werden, daß beim Anschließen und bei der Inbetriebnahme zu keinem Zeitpunkt ein Potentialausgleich, z.B. zwischen dem Netzteil und dem PC, über Signalleitungen passieren kann. Dies kann zur Zerstörung von IC's führen. Das gilt insbesondere auch beim Einsatz der ML3 in einem PC mit externer Versorgung.

Empfohlen wird dringend, zuerst die Netzstecker am PC und an allen mit dem PC verbundenen Geräten und am externen Netzteil herauszuziehen. Auch die Karte oder die aufgesteckten Module dürfen keinerlei Kabelverbindungen mit externen Geräten haben. Dann werden zunächst die Masse-verbindungen (Ground) hergestellt. Erst danach dürfen die Netzstecker wieder eingesteckt und die übrigen Verbindungen hergestellt werden.

Wenn eine Karte aus dem PC ausgebaut wird, müssen zunächst alle Verbindungen vom PC bzw. der am PC angeschlossenen Geräte abgetrennt werden. Danach müssen unbedingt alle (!) Verbindungen von der Karte und den Modulen abgetrennt werden. Dann erst darf die Karte aus dem PC herausgenommen werden.

3.1. Der PC-Bus-Stecker für Karten mit PC- oder PC/AT-Bus (ML1, ML4 und ML5)

Die Versorgungsspannungen werden über den PC-Busstecker zugeführt. Erforderlich sind + 5 Volt und +/- 12 Volt. Die +/- 12 Volt werden dabei, außer bei der ML1, lediglich zur Versorgung der seriellen RS-232 Schnittstelle(n) benötigt.

Die Stromaufnahme ist bei MODULAR-4 Karten natürlich abhängig von den aufgesteckten Modulen. Eine ML4 oder ML5 Basiskarte ohne Module benötigt bei 5 Volt ca. 550 mA, bei +/- 12 V jeweils weniger als 10 mA, nähere Angaben hierzu finden Sie in den technischen Datenblättern.

Außerdem sind noch zusätzliche Verbindungen erforderlich, die in folgender Tabelle angegeben sind (alle nicht aufgeführten Anschlüsse bleiben frei):

Pin am PC-Stecker <1>	anzuschließen an	für
B1	GND	Versorgung
B3	+ 5 Volt	Versorgung
B7	- 12 Volt	Versorgung
B9	+ 12 Volt	Versorgung
B10	GND	Versorgung
B29	+ 5 Volt	Versorgung
B31	GND	Versorgung
A11	log. 1 (+ 5 Volt)	PC-Schnittstelle
B2 <2>	log. 0 (GND)	PC-Schnittstelle

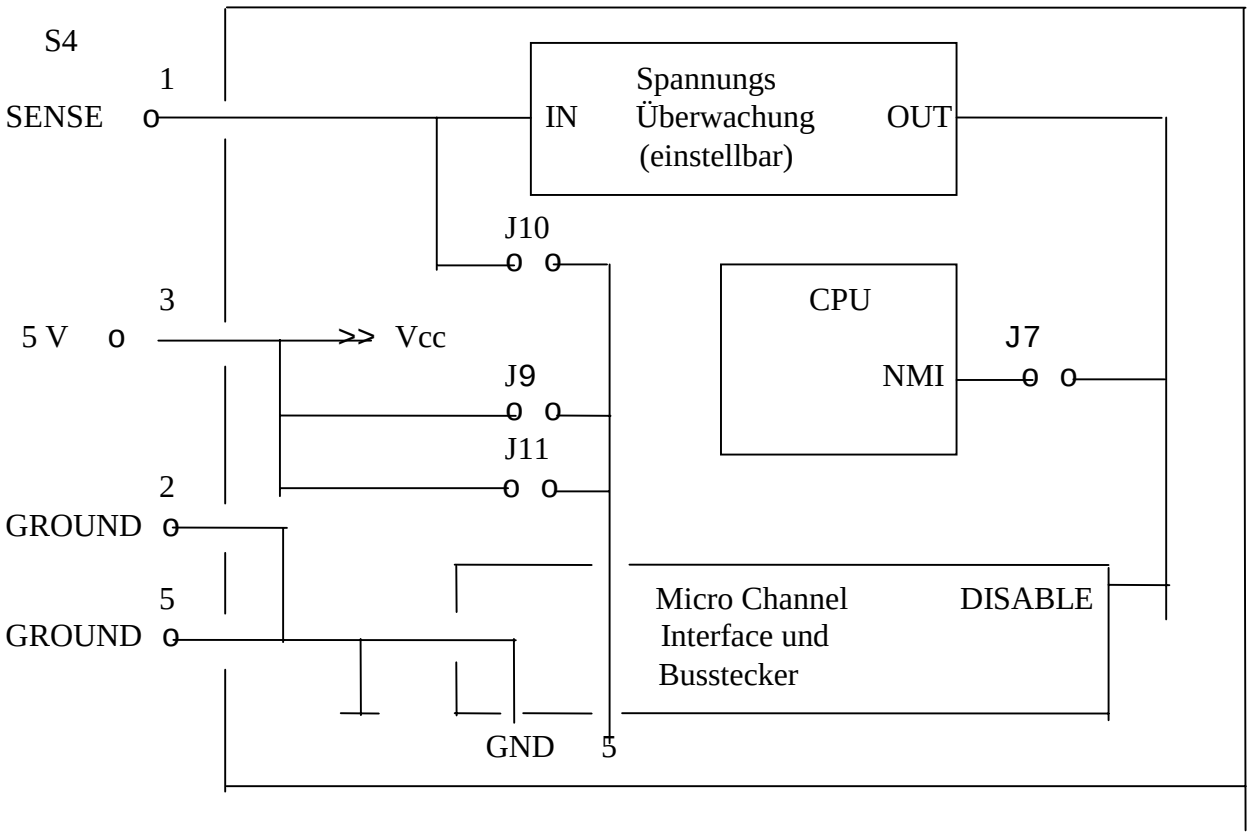
<1>Die Kontaktreihe A befindet sich auf der Bauteileseite, beginnend mit A1 in der Nähe des Befestigungsbügels.

<2> Diese Verbindung ist nur bei ML1 erforderlich.

3.2. Versorgungsspannung für Karten mit Micro-Channel Bus (ML3)

Die ML3-Karte benötigt lediglich eine + 5 Volt Versorgungsspannung, falls keine Module aufgesteckt sind, die +/- 12 Volt benötigen. Alle Spannungen für die RS-232 Schnittstellen der Basiskarte werden direkt auf der Karte erzeugt. Die + 5 Volt Spannung kann über den Pfostenstecker S4 zugeführt werden. In diesem Fall bleiben alle Anschlüsse des Micro-Channel frei.

Prinzipschaltung der Spannungsversorgung und Überwachung der ML3



Bei den verschiedenen Einsatzmöglichkeiten sind die Jumper J9, J10 und J11 und Anschlußpin 1 des Pfostensteckers S4 zu beachten:

Konfiguration	J9	J10	J11	Pin 1
ML3 im PS/2 mit Versorgung über Bus	+	+	+	frei
ML3 im PS/2 mit externer Versorgung über S4	-	+	-	frei
ML3 als stand-alone System mit Versorgung über S4	+	+	+	frei
ML3 als stand-alone System mit Versorgung über Busstecker	+	+	+	frei

3.3. Anschluß der seriellen Schnittstelle

3.3.1. Signale der seriellen Schnittstelle des Host-Rechners (Belegung wie IBM PC/XT/AT)

Die Signale der seriellen Schnittstelle des PC liegen beim 9-pol. bzw. 25-pol. Stecker auf folgenden Pins:

Signal Steckertyp	9-pol. D-Sub	25-pol. D-Sub	Verwendet als
Carrier Detect	1	8	hier nicht verwendet
Receive Data	2	3	Receive Data
Transmit Data	3	2	Transmit Data
DTR	4	20	optional für RESET (s.u.)
Ground	5	7	Ground
DSR	6	6	hier nicht verwendet
RTS	7	4	RTS
CTS	8	5	hier nicht verwendet
Ring Indicator	9	22	hier nicht verwendet

3.3.2. Die verwendete serielle Schnittstelle der Karten

Wenn mehrere serielle Schnittstellen auf der Basiskarte vorhanden sind, wird die serielle Schnittstelle A als Host-Schnittstelle verwendet. Nicht bei allen Karten sind alle Modem-Leitungen vorhanden, sie werden aber hier auch nicht benötigt:

Karte Stecker Steckertyp Signal	ML1 J1 D-Sub	ML3 S2 D-Sub	ML4 J2 D-Sub	ML4 J4 Pfosten	ML5 S2 D-Sub	ML5 S5 Pfosten
DCD	-	1	1	1	1	1
Receive Data	2	2	2	3	2	3
Transmit Data	3	3	3	5	3	5
DTR	-	4	4	7	4	7
Ground	5	5	5	9	5	9
DSR	-	6	-	-	-	-
RTS	7	7	7	4	7	4
CTS	8	8	8	6	8	6
RI	-	9	-	-	-	-

3.3.3. Verbindungskabel zwischen PC und Karte

Für die Kommunikation mit einem Host-Rechner, z.B. einem PC, ist ein Kabel zu verwenden, das folgende Verbindungen herstellt:

PC				Karte	
9-pol. D-Sub	25.pol. D-Sub	Signal		Signal	9-pol. D-Sub
2	3	Receive Data	<<----	Transmit Data	3
3	2	Transmit Data	---->>	Receive Data	2
5	7	Ground	<<--->>	Ground	5
7	4	RTS	---->>	CTS	8
4	20	DTR <1>	---->>	DCD <1>	1

<1> Die Leitung von DTR an DCD wird nicht für die Kommunikation verwendet und kann auch entfallen. Sie kann aber benutzt werden, um über die serielle Schnittstelle des Host-Rechners ein Hardware-Reset der Karte auszulösen (siehe nächstes Kapitel für Hardware und Kapitel 4 für Software).

3.3.4. Hardware-Reset über DTR

Die DTR-Leitung des Host-Rechners kann verwendet werden, um ein Hardware-Reset der Karte auszulösen. Bei den Karten ML3, ML4 (ab Serien-Nr. 80000) und ML5 muß hierfür lediglich ein Jumper aufgesteckt werden:

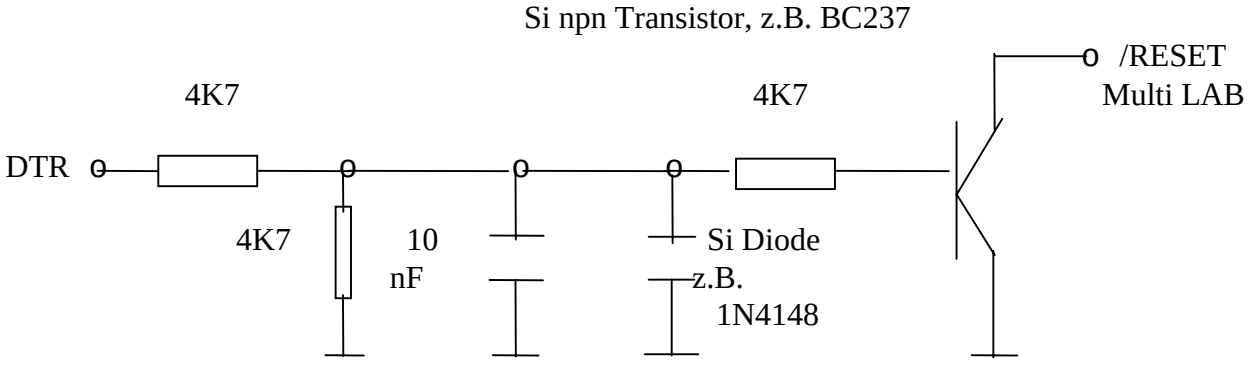
ML3: Stecker S4, Jumper von 4-6

ML4 (ab Serien-Nr. 80000): Jumperfeld J5, Jumper von 1-2

ML4 (Serien-Nr. < 80000): Hier muß eine Drahtverbindung auf der Karte gelötet werden: Von Pin 13 von IC 16 nach Pin 1 von J5 (Reset-Eingang).

ML5: Stecker S6, Jumper von 4-6

ML1: Es muß eine spezielle "Reset-Logik" zwischen DTR und Reset-Eingang (Pin 2 von J9) geschaltet werden:



Alternativ kann auch ein Gatter eines invertierenden RS-232 Empfängerbausteins verwendet werden, z.B. 75189 bzw. MC1489:

Pin von 75189	an
7	GND
14	+ 5 Volt
1	DTR-Ausgang von Host-Schnittstelle
3	Reset-Eingang von Multi-LAB (Pin 2 von J9)

4. Software

Bei der ML1 und bei früheren Betriebssystem-Versionen der ML4 Karte muß ein spezielles EPROM eingesetzt werden, um die serielle Schnittstelle als Host-Schnittstelle verwenden zu können. Bei der ML1 ist ein EPROM mit der Bezeichnung ML1-4Cx.SIO (x >= 4) einzusetzen. Die Standardeinstellung der seriellen Schnittstelle der Karte ist dabei 9600 Baud, 8 Bit, No Parity und 1 Stop-Bit.

Bei der ML4 sollte die neueste Betriebssystem-Version eingesetzt werden. Damit kann die Konfiguration der seriellen Schnittstelle als Host-Schnittstelle per Software eingestellt werden. Es ist damit auch möglich, die parallele PC-Schnittstelle und die serielle Schnittstelle gleichzeitig als Host-Schnittstelle einzusetzen, was z.B. auch zum Debuggen sehr hilfreich sein kann.

Da die RTS-Leitung der seriellen Schnittstelle des PC nach dem Einschalten des PC oder nach einem Hardware-Reset des PC auf logisch "1" (entspr. -12 Volt) liegt, muß diese Leitung per Software auf logisch "0" (entspr. +12 Volt) gelegt werden, damit eine Kommunikation mit der MODULAR-4 Karte stattfinden kann.

Der BIOS Software-Interrupt INT 14 kann für die Standard-Schnittstellen COM1 bis COM4 nicht verwendet werden, da damit das Setzen der RTS- oder DTR-Leitung nicht möglich ist. Deshalb ist von Sorcus Computer das Programm MXSIO.COM verfügbar, das einen neuen Software-Interrupt (standardmäßig 63H) enthält, der die Kommunikation über COM1, COM2, COM3 oder COM4 mit der Karte durchführt.

Das Setzen der RTS- oder DTR-Leitung wird durch einen Aufruf dieses Software-Interrupts ermöglicht (AH = 1 bzw. AH = 6). Die Initialisierung der Schnittstelle geschieht ebenfalls über einen Aufruf dieses Software-Interrupts (AH = 2).

Das PC-Programm MXSIO.COM muß einmal (z.B. nach dem Einschalten des Systems) aufgerufen werden. Danach ist es resident im Speicher des PC. Alle für die Kommunikation erforderlichen Routinen werden als Aufrufe über diesen Software-Interrupt (s.u.) zur Verfügung gestellt.

Beim Aufruf des Programms MXSIO.COM wird ein Parameter übergeben:

MXSIO /63

63 Nr.des Software-Interrupt auf dem PCals Hexadezimal-Wert (optional)

Sollte der Software-Interrupt 63H bereits von anderen Programmen verwendet werden, kann das Programm MXSIO.COM auch unter Angabe eines anderen Parameters aufgerufen werden. Ohne Angabe des Parameters wird 63H verwendet.

4.1. Beschreibung des neuen PC-Software-Interrupts 63H

Der Software-Interrupt wird durch Aufruf des Programms MXSIO.COM installiert (s.o.) und Speicher-resident gemacht. Er stellt verschiedene Routinen für die Kommunikation über eine der seriellen Schnittstellen des PC (COM1 bis COM4) mit der Karte zur Verfügung:

4.1.1. Anwahl einer seriellen Schnittstelle des PC

Dieser Systemaufruf überprüft, ob die angegebene serielle Schnittstelle vorhanden ist. **Hiermit wird die angegebene Schnittstelle für alle nachfolgenden Aufrufe angewählt. Dieser Aufruf ist vor Beginn der Kommunikation einmal auszuführen.**

Input: AH = 0
DX = DOS-Kanalnummer (0 = COM1, 1 = COM2,
2 = COM3, 3 = COM4)

Output: AH = 0: Anwahl durchgeführt
AH = 5: Keine Serielle Schnittstelle im PC vorhanden

3.1.2. Reset der Karte

Dieser Systemaufruf setzt den DTR-Ausgang der angewählten seriellen Schnittstelle des PC kurzzeitig (ca. 1 ms) auf + 12 Volt (logisch "0") und anschließend wieder auf - 12 Volt (logisch "1"). Außerdem wird die RTS-Leitung der angewählten Schnittstelle auf + 12 Volt (logisch "0") gelegt.

Dieser Aufruf muß nach POWER-ON der Karte einmal ausgeführt werden.

Input: AH = 1

Output: AH = 0: Reset durchgeführt

4.1.2. Initialisieren der seriellen Schnittstelle des PC

Dieser Aufruf initialisiert die angewählte serielle Schnittstelle des PC.

Input : AH = 2
AL = Init Parameter:

Baudrate	Parity	Stop-Bit	Länge
7 6 5	4 3	2	1 0
000- 110 |	00 - None	0 - 1	10- 7 Bits
001 - 150	01 - Odd	1 - 2	11 - 8 Bits
010 - 300	11 - Even		
011 - 600			
100 - 1200			
101 - 2400			
110 - 4800			
111 - 9600			

Output: AH = 0: Kein Fehler aufgetreten
AL = Line Status (siehe Kapitel 3.1.3.)

4.1.3. Status der seriellen Schnittstelle des PC

Dieser Systemaufruf liest das Line Status Register der angewählten seriellen Schnittstelle des PC.

Input: AH = 3

Output: AH = 0: Kein Fehler aufgetreten
AL = Line Status
Bit 7 = 0
Bit 6 = Transmit Shift Register Empty
Bit 5 = Transmit Holding Register Empty (d.h.: es kann ein Zeichen über die serielle Schnittstelle gesendet werden)
Bit 4 = Break Detect
Bit 3 = Framing Error
Bit 2 = Parity Error
Bit 1 = Overrun Error
Bit 0 = Data Ready (d.h.: es ist ein Zeichen in der seriellen Schnittstelle empfangen worden)

4.1.4. Senden eines Zeichens über die serielle Schnittstelle

Dieser Systemaufruf sendet ein Zeichen vom PC zur Karte über die angewählte serielle Schnittstelle. Es wird der Status der seriellen Schnittstelle abgefragt und geprüft, ob Bit 5 (Transmit Holding Register Empty) = 1 ist. Ist Bit 5 = 1, wird das Zeichen in AL gesendet, sonst wird weiter abgefragt, maximal 65536 mal, danach wird Register AH = 254 (Time-Out) gesetzt und der Aufruf beendet.

Input: AH = 4
AL = Zeichen

Output*:* AH = 0: Kein Fehler aufgetreten, AL = Zeichen
AH= 254: Time-Out: Es konnte kein Zeichen über die serielle Schnittstelle gesendet werden, AL = Line Status (s.o.)
AL = Line Status
Bit 7 = 0
Bit 6 = Transmit Shift Register Empty
Bit 5 = Transmit Holding Register Empty
(d.h.: es kann ein Zeichen über die serielle Schnittstelle gesendet werden)
Bit 4 = Break Detect
Bit 3 = Framing Error
Bit 2 = Parity Error
Bit 1 = Overrun Error
Bit 0 = Data Ready (d.h.: es ist ein Zeichen in der seriellen Schnittstelle empfangen worden)

4.1.5. Lesen eines Zeichens von der seriellen Schnittstelle

Dieser Systemaufruf liest ein Zeichen von der angewählten seriellen Schnittstelle des PC. Es wird der Status der seriellen Schnittstelle abgefragt und geprüft, ob Bit 0 (Data Ready) = 1 ist. Ist Bit 0 = 1, wird ein Zeichen gelesen und in AL geschrieben, sonst wird weiter abgefragt, maximal 65536 mal, danach wird Register AH = 255 (Time-Out) gesetzt und der Aufruf beendet.

Input: AH = 5

Output: AH = 0: Kein Fehler aufgetreten, AL = Zeichen
AH= 255: Time-Out: Es konnte kein Zeichen von der seriellen Schnittstelle empfangen werden, AL = Line Status (s.o.)
AL = Line Status
Bit 7 = 0
Bit 6 = Transmit Shift Register Empty
Bit 5 = Transmit Holding Register Empty
(d.h.: es kann ein Zeichen über die serielle Schnittstelle gesendet werden)
Bit 4 = Break Detect
Bit 3 = Framing Error
Bit 2 = Parity Error
Bit 1 = Overrun Error
Bit 0 = Data Ready (d.h.: es ist ein Zeichen in der seriellen Schnittstelle empfangen worden)

4.1.6. Setzen der RTS-Leitung der seriellen Schnittstelle auf logisch "0"

Dieser Systemaufruf legt die RTS-Leitung der angewählten seriellen Schnittstelle des PC auf +12 Volt (logisch "0").

Dieser Aufruf muß nach POWER-ON und Hardware-Reset des PC einmal ausgeführt werden, weil durch POWER-ON und Hardware-Reset des PC die RTS-Leitung auf -12 Volt (logisch "1") gelegt wird (s. 3.1.2.)

Input: AH = 6

Output: AH = 0: Kein Fehler aufgetreten

4.2. Einbinden des Software-Interrupts in die PC-Bibliotheken für die MODULAR-4 Karten

Für die Kommunikation zwischen PC und Karte können die bisher gelieferten PC-Bibliotheken für Turbo-PASCAL (ML4LIB.PAS), Turbo-BASIC (ML4LIB.BAS) und Microsoft-C (ML4LIB.C) weiter verwendet werden, es ist dabei aber folgendes zu beachten:

Die eventuell noch vorhandenen Delays bei der Abfrage des Uhrenbausteins und des EEPROMS in den Prozeduren, die in den Kapiteln 6.6.6. bis 6.6.10, 6.7.2 und 6.2.9 im MODULAR-4 Benutzer-Handbuch beschrieben sind, müssen entfallen.

Die Anwahl der seriellen Schnittstelle (Aufruf des Software-Interrupts mit AH = 0) und die Initialisierung (Aufruf mit AH = 2) müssen im Anwenderprogramm vor Benutzung der Bibliotheksroutinen vorgenommen werden.

Die Prozeduren bzw. Funktionen ML4_READ, ML4_WRITE und ML4_RESET in der Datei ML4LIB.PAS sind wie unten angegeben zu ändern. Entsprechendes gilt für die Prozeduren m4read, m4write und m4reset in der Datei ML4LIB.C und für die Prozeduren ML4READ, ML4WRITE und ML4RESET in der Datei ML4LIB.BAS.

```
function ml4_read : byte;
var
  regs: record
        case integer of
        0 : (ax,bx,cx,dx,bp,si,di,ds,es,flags : integer);
        1 : (al,ah,bl,bh,cl,ch,dl,dh : byte);
        end;

begin
  ml4_err := 0;
  with regs do
    begin
      ax := $500;              (* Lesen eines Zeichens von der *)
      intr($63,regs);          (* seriellen Schnittstelle *)
      ml4_read := lo(ax);      (* Zeichen an Funktion übergeben *)
      ml4_err := hi(ax);       (* Fehlercode an Fehlervariable übergeben *)
    end;
end;
```

```
procedure ml4_write(datbyte : byte);
var
  regs: record
        case integer of
        0 : (ax,bx,cx,dx,bp,si,di,ds,es,flags : integer);
        1 : (al,ah,bl,bh,cl,ch,dl,dh : byte);
        end;

begin
 ml4_err := 0;
 with regs do
   begin
     ax := $400+datbyte;      (* Zeichen über serielle *)
     intr($63,regs);          (* Schnittstelle senden *)
     ml4_err := hi(ax);       (* Fehlercode an Fehlervariable übergeben *)
   end;
end;
```

```
procedure ml4_reset(var err1 : byte);
var
  regs: record
        case integer of
        0 : (ax,bx,cx,dx,bp,si,di,ds,es,flags : integer);
        1 : (al,ah,bl,bh,cl,ch,dl,dh : byte);
        end;

begin
 ml4_err := 0;
 with regs do
   begin
     ax := $100;      (* Reset der MODULAR-4 über die serielle Schnittstelle des PC
                         durchführen *)
     delay (2000)     (* 2 Sekunden Pause bis fertig *)
     intr($63,regs);
     if ah = 0 then   (* Kein Fehler aufgetreten : *)
       begin
         ax := $300;  (* Status der seriellen Schnittstelle abfragen *)
         intr($63,regs);
         if(al and 1) = 1 then  (* Bit 0 = 1: Zeichen in der seriellen Schnittstelle *)
           begin
             ax := $500;        (* Zeichen von der seriellen Schnittstelle lesen *)
             intr($63,regs);
           end;
       end;
     ml4_err := ah;   (* Fehlercode an Fehlervariable übergeben *)
   end;
end;
```

4.3. Einbinden des Software-Interrupts in die PC-Bibliotheken für die Multi-LAB Karte

Für die Kommunikation zwischen PC und Multi-LAB Karte können die bisher gelieferten PC-Bibliotheken für Turbo-PASCAL (MLABLIB.PAS), Turbo-BASIC (MLABLIB.BAS) und Microsoft-C (MLLIB.C) weiter verwendet werden, es ist dabei aber folgendes ist zu beachten:

Die eventuell noch vorhandenen Delays bei der Abfrage des Uhrbausteins in den Prozeduren, die in den Kapitel 6.6.6. bis 6.6.10 und 6.7.2 im Multi-LAB Handbuch beschrieben sind, müssen entfallen.

Die Anwahl der seriellen Schnittstelle (Aufruf des Software-Interrupts mit AH = 0) und die Initialisierung (Aufruf mit AH = 2) müssen im Anwenderprogramm vor Benutzung der Bibliotheksroutinen vorgenommen werden.

Die Prozeduren bzw. Funktionen MLREAD, MLWRITE und MLRESET in der Datei MLABLIB.PAS sind wie unten angegeben zu ändern. Entsprechendes gilt für die Prozeduren mlread, mlwrite und mlreset in der Datei MLLIB.C und für die Prozeduren MLREAD, MLWRITE und MLRESET in der Datei MLABLIB.BAS.

```
procedure MLREAD (var datbyte : byte);
var
  regs: record
        case integer of
        0 : (ax,bx,cx,dx,bp,si,di,ds,es,flags : integer);
        1 : (al,ah,bl,bh,cl,ch,dl,dh : byte);
        end;

begin
 err := 0;
 with regs do
   begin
     ax := $500;              (* Lesen eines Zeichens von der *)
     intr($63,regs);          (* seriellen Schnittstelle *)
     datbyte := lo(ax);       (* Zeichen an Funktion übergeben *)
     err := hi(ax);           (* Fehlercode an Fehlervariable übergeben *)
   end;
end;
```

```
procedure mlwrite(datbyte : byte);
var
  regs: record
        case integer of
        0 : (ax,bx,cx,dx,bp,si,di,ds,es,flags : integer);
        1 : (al,ah,bl,bh,cl,ch,dl,dh : byte);
        end;

begin
 err := 0;
 with regs do
   begin
     ax := $400+datbyte;      (* Zeichen über serielle *)
     intr($63,regs);          (* Schnittstelle senden *)
     err := hi(ax);           (* Fehlercode an Fehlervariable übergeben *)
   end;
end;
```

```
procedure mlreset(var err1 : byte);
var
  regs: record
        case integer of
        0 : (ax,bx,cx,dx,bp,si,di,ds,es,flags : integer);
        1 : (al,ah,bl,bh,cl,ch,dl,dh : byte);
        end;

begin
 err := 0;
 with regs do
   begin
     ax := $100;      (* Reset der Multi-LAB über die serielle Schnittstelle des PC durchführen *)
     intr($63,regs);
     err := ah;       (* Fehlercode an Fehlervariable übergeben *)
end;
```