

3.4.1 Wandlung einer Centronics Druckerschnittstelle in eine RS232 Schnittstelle mit XON - XOFF Protokoll und DTR Protokoll.

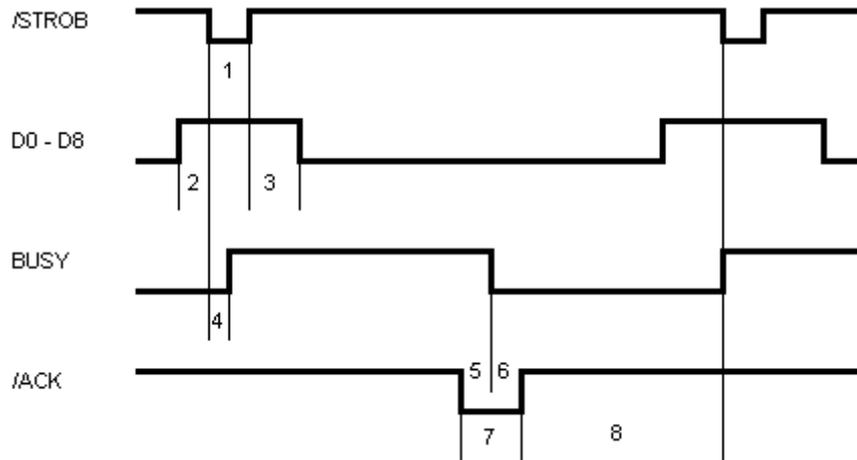
3.4.1.1. Centronics - Signale am ZWERG11A

25pol. Subm.-D männlich		7400	ZWERG11A	Bemerkung	
E/A					
01 E	~STROB	5,10 ---->	8 STRA 8 PA1	1.24 3.11	D1 bis D8 wird übernommen -> BUSY ein.
02 E	D1		PC0	1.11	\
03 E	D2		PC1	1.12	
04 E	D3		PC2	1.13	
05 E	D4		PC3	1.14	_ Druck und Steuerzeichen
06 E	D5		PC4	1.15	
07 E	D6		PC5	1.16	
08 E	D7		PC6	1.17	
09 E	D8		PC7	1.18	/
10 A	~ACK	1	PB0	1.08	Impuls, wenn Drucker wieder bereit
11 A	BUSY	11 <-13	PB1	1.09	Drucker besetzt - Anzeige
12 A	PE		PB2	1.20	Pappierende - Meldung vom Drucker
13 A	SELECT		PB3	1.21	ON-LINE - Meldung vom Drucker
14 E	~AUTO F		PE0	3.02	Drucker auf AUTO-FEED schalten
15 A	~ERR		PB4	1.05	Fehlermeldung des Druckers
16 E	~RESET		PE1	3.04	Drucker rücksetzen
17 E	~SLK IN		PE2	3.06	Drucker ON-LINE schalten
18	GND			1.01	
				1.02	
:	:			1.26	
.	.			3.03	
				3.07	
25	GND			3.25	
			PA4	3.17	1, wenn Drucker im Stop
			PA5	3.19	1, wenn kein DC1 oder DC3 empfangen
			PA6	3.21	1, wenn letztes Zeichen kein DC1 oder DC3
			PA7	3.23	1, wenn OR-Fehler bis Puffer leer

Tabelle 1

3.4.1.2. Impulsdiagramme für Centronics-Interface

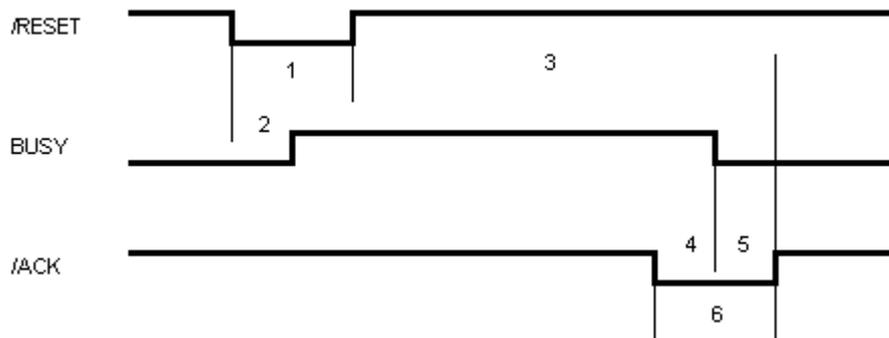
1. Datenübernahme vom Drucker



$t_1 = 1 - 50 \mu\text{s}$ $t_3 \geq 1 \mu\text{s}$ $t_5 \leq 7 \mu\text{s}$ $t_7 = 5 - 10 \mu\text{s}$
 $t_2 \geq 1 \mu\text{s}$ $t_4 \leq 0,5 \mu\text{s}$ $t_6 \leq 7 \mu\text{s}$ $t_8 \geq 0 \mu\text{s}$

Bild 1: Impulsdiagramm zur Datenübernahme vom Drucker

2. Initialisierung des Druckers durch ~RESET



$t_1 \geq 50 \mu\text{s}$ $t_3 \leq 5 \text{ ms}$ $t_5 \leq 7 \mu\text{s}$
 $t_2 \leq 30 \mu\text{s}$ $t_4 \leq 7 \mu\text{s}$ $t_6 = 5 - 10 \mu\text{s}$

Bild 2: Impulsdiagramm zur Initialisierung des Druckers durch ~RESET

3.4.1.3. Beschaltung eines 7400 zur Erzeugung von BUSY

Zur Realisierung der $0,5 \mu\text{s}$ von BUSY nach STROB gleich t_4 im Bild 1 ist zusätzlich noch ein 7400 mit folgender Beschaltung erforderlich.

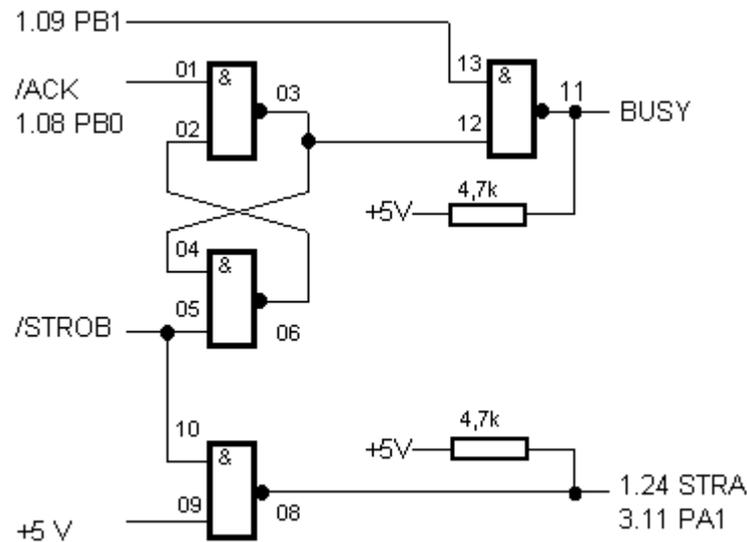


Bild 3: Beschaltung eines 7400 zur Realisierung der $0,5 \mu\text{s}$ von BUSY nach STROB

3.4.1.4. Gesamtschaltung für den Interfaceconverter

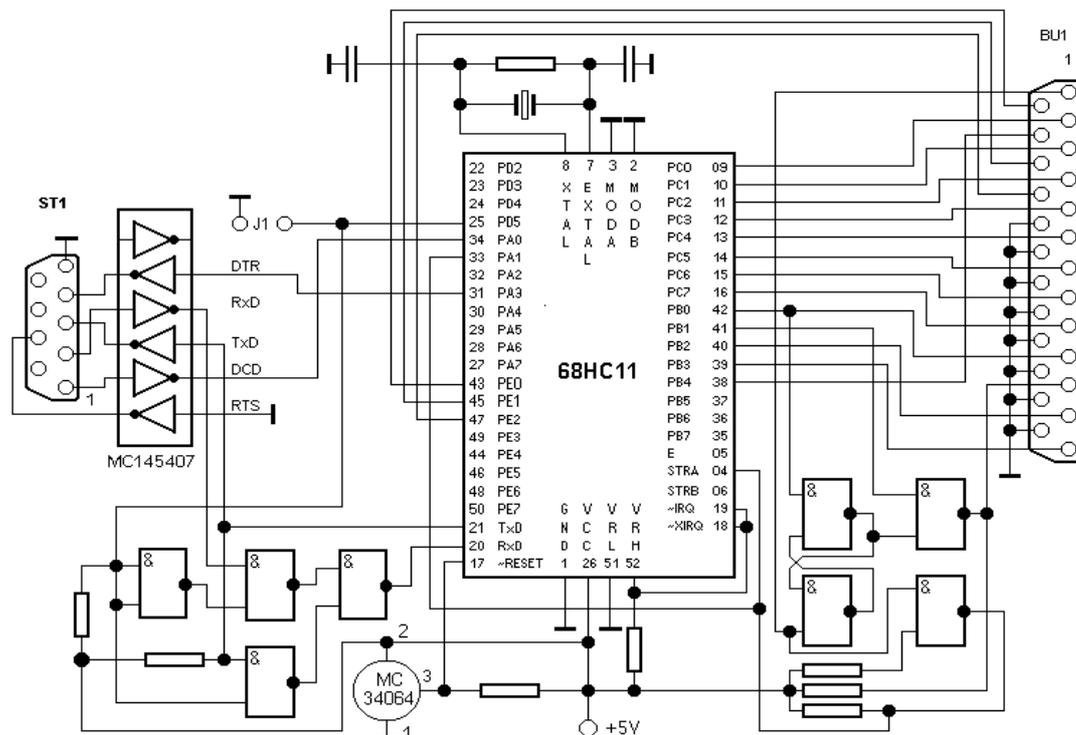


Bild 4: Gesamtschaltung des Interfaceconverters

Der Materialaufwand für diesen Interfaceconverter ist sehr gering, so dass sich die Kosten für die Bauelemente auf ca. 35,00 DM belaufen. Mit einem Gehäuse und einem Steckernetzteil von 5 V ist diese Schaltung für einen Preis der deutlich unter 100,00 DM liegt realisierbar.

3.4.1.5. Programmablauf für Centronics-Interface

Centronics-Wartezyklus

- | | |
|--|---|
| 1. Wurde ein Zeichen mit STROB empfangen | -> Senden an RS232, wenn bereit |
| 2. Testen von Port E | -> DRUCKER-Mode einstellen |
| 3. Ist Puffer wieder frei | -> BUSY wieder r cksetzen und ~ACK Impuls |

Interruptbehandlung von STROB

- | | |
|---------------------------------------|------------------------|
| 1. BUSY setzen | |
| 2. Daten von Port C lesen und puffern | |
| 3. Bei Puffer berlauf | -> Fehlerstatus setzen |

Senden an RS232 zum Drucker

- | | |
|--|--|
| 1. Ist Drucker im XON-Zustand (bereit),
sonst sofort weiter | -> Zeichen senden und Puffer frei machen |
| 2. Ist ein Fehler aufgetreten | -> Fehlerstatus setzen |

Interruptbehandlung von SCI-Rx(Empfang von DC1 oder DC3)

Da im Abfragemode des SCI-Empf ngers h ufig ein Overrun Error auftritt, mu ste der Empf nger interruptgesteuert realisiert werden. Im Falle eines OR-Fehlers ist beim SCI-Interface des 68HC11 im Datenregister nicht das zuletzt empfangene Zeichen, sondern das Zeichen, welches vor dem Auftreten des OR-Fehlers als letztes empfangen wurden ist. Folgende Funktionen mu  das Interruptprogramm realisieren:

- | | |
|---|--------------------------------------|
| 1. Empf ngerstatus testen | -> OR-Fehlerbehandlung, wenn n tig |
| 2. Daten testen ob DC1 oder DC3 | -> Datenfehler behandeln, wenn n tig |
| 3. DC1 oder DC3 f r Druckerstatus speichern | |

3.4.1.6. Parameter der Programmvariante im File CEN_232X.A

Alle Angaben sind f r einen ZWERG11A mit 8 MHz Quarzfrequenz g ltig!

3.4.1.6.1. Centronics Interface

Durch das Programm werden folgende Zeiten f r die Signalspiele nach Bild 1 und Bild 2 realisiert:

- | | | |
|------------|-------------|---|
| aus Bild1: | t4 <= 30 ns | Laufzeit von 3 Gattern des 7400 |
| | t5 = 3,5  s | Zeit des Befehls BCLR = 7 Zyklen * 0,5  s |
| | t6 = 3,5  s | Zeit des Befehls BSET = 7 Zyklen * 0,5  s |
| | t7 = 7,0  s | Zeit von t5 + t6 |

Gemessene Interruptreaktionszeit als Zeit von STROB bis PB1. Von dem gemessenen Wert m ssen die 3,5  s des Befehls BCLR abgezogen werden, um auf die reine Interruptreaktionszeit zu kommen.

$$t_{\text{int}} = t_{\text{me }} - 3,5 \mu\text{s} = 8,5 \dots 11 \mu\text{s}$$

- | | | |
|------------|-------------|---------------------|
| aus Bild2: | t2 < 20  s | gemessen |
| | t4 = 3,5  s | gleich t5 aus Bild1 |
| | t5 = 3,5  s | gleich t6 aus Bild1 |

t6 = 7,0 µs gleich t7 aus Bild1

Das Centronics-Interface realisiert eine Datenübertragungsrate von ca. 8 k Byte/s (gemessen). Die empfangenen Bytes werden in einen 160 Byte großen Puffer im ZWERG11A geschrieben. Wenn der Puffer voll ist, wird BUSY erst wieder freigegeben, wenn nur noch 60 Byte im Puffer sind. Dadurch wird vom Rechner zum ZWERG11A eine höhere Datenübertragungsrate realisiert als auf dem RS232-Interface zum Drucker, was mit maximal 9600 BAUD arbeitet.

3.4.1.6.2. RS232-Interface

Das RS232-Interface wird durch das Programm CEN_232X.A auf folgende Parameter eingestellt:

Datenrate: 9600 Baud
Datenbits: 8
Stopbits: 1
Parität: keine

Damit ergibt sich eine Datenübertragungsrate von 960 Byte/s für das RS232-Interface. Vom Programm werden dabei folgende Signale des RS232-Interface benutzt und sind wie folgt mit dem Drucker zu verbinden:

9pol. Subm.-D Stecker am ZWERG11A		Signal		Signal am Drucker
1		DCD		DTR oder +12V bei XON-XOFF
2		RxD		TxD oder -12V bei DTR
3		TxD		RxD
4	-12V	DTR	ständig negativ für Brücke mit RxD bei DTR	
5		GND		GND
7	+12V	RTS	ständig positiv für Brücke mit DCD bei XON-XOFF	

Am Drucker sind je nach Angaben des Herstellers eventuell noch folgende Signale mit +12V zu belegen. Das kann durch Brücken am Drucker mit dem RTS-Signal des Druckers realisiert werden. Folgende Signale müssen mit RTS verbunden werden:

DCD, DSR und RI

3.4.1.7. Testbedingungen

Für die Austestung der Software und Hardware wurden 2 Drucker mit seriellem Interface verwendet. Zum Test des XON-XOFF-Protokolls wurde ein LA75 von DEC verwendet. Der LA75 wurde im Setup wie folgt eingestellt:

LA75-S Code Version(P2.0)

Feature		Value
1	Baud Rate	8 9600
2	Daten Bits and Parity	7 8-None
3	Protocol	2 Other
4	Form Length	1 12 inches
5	Character Set (DEC)	1 U.S. ASCII
6	Supplemental Character Set (DEC)	1 DEC Supplemental
7	Print Density (DEC)	2 Memo
8	Printer ID (DEC)	1 LA50 ID
9	Text Mode Right Margin (DEC)	2 Wrap
10	Auto LF on CR (DEC)	2 No auto LF

11	Auto CR on LF (DEC)	2	No auto CR
12	CAN Control Code (DEC)	1	Cancel Control Functions
13	80 or 132 Columns (DEC)	2	132 Columns
14	Paper Out Bell (Emulation)	2	On
15	Slashed Zero (Emulation)	1	Zero with slash
16	Auto LF on CR (Emulation)	2	No auto LF
17	Active Character Set (Emulation)	2	Graphics set B
18	Auto CR on LF (Emulation)	2	No auto CR
19	Power-up Density (Emulation)	1	Draft
20	LQ or NLQ (Emulation)	2	Select LQ

Der LA75 wurde mit seinem Original-Interface-Kabel mit 25poligem Stecker über einen Adapter auf einen 9poligen Stecker umgesetzt und an den ZWERG11A angeschlossen. Da der LA75 nicht alle ESC-Sequenzen realisiert, wie sie für EPSON/IBM-Printer definiert sind, wurde noch ein Treiber auf dem IBM-PC installiert, der diese Sequenzen in die Sequenzen des LA75 umwandelt. Dieser Treiber hat den Namen IBM-LA75.EXE und wird durch Aufruf von

IBM-LA75 1 oder lh IBM-LA75 1

für das Printerport LPT1 installiert. Er wandelt folgende Sequenzen um:

NLQ ein	ESC,x,1	in	ESC,G
NLQ aus	ESC,x,0	in	ESC,H
5 CPI ein	SO (0E)	in	ESC,W,01
5 CPI aus	ESC,SO	in	ESC,W,01
12 CPI ein	DC4 (14)	in	ESC,W,00
12 CPI aus	ESC,M	in	ESC,;
RESET	ESC,P	in	DC2
BELL	ESC,@	in	ESC,[,?,5,8,h,ESC,@,ESC,6
	BELL (07)	in	ESC,6

Die Angaben in Klammern entsprechen der hexadezimalen Kodierung der ASCII-Steuerzeichen SO, DC4 und BELL. Das Steuerzeichen 1b Hex wird mit ESC angegeben. Alle anderen Zeichen sind druckbare ASCII-Zeichen, außer wenn sie nicht durch Komma getrennt sind. Die Ziffernfolgen 00 und 01 sind hexadezimal.

Zum Test des DTR-Protokolls wurde ein K6314 mit V24-Interface verwendet. Die DIL-Schalter des K6314 mit der Steuerprogrammnummer 4.35-3-85/06 hatten dabei folgende Einstellung:

Schalter	Funktion	ON	OFF	Einstellung
5-1	Haubenkontakt	unwirksam	wirksam	OFF
5-2	keine Funktion			
6-1	Papierendekontakt	unwirksam	wirksam	OFF
6-2	Summer	unwirksam	wirksam	OFF
7-1	Auto LF	CR + LF	nur CR	OFF
7-2	Auto CR	nur LF	LF + CR	ON
8-1	Zeichensatz	1	2	OFF
8-2	Puffer voll Bedingung	nur Druck	Druck + LF	ON
9-1	Abbruch Modus (CAN)	ungültig	gültig	OFF
9-2	Zeichencode	Schneider	IBM	OFF
10-1	\ Schriftart eingestellt auf Pica = 1/10"			OFF
10-2	/			OFF

11-1	Nulldarstellung	mit /	ohne /	ON
11-2	Zeilenabstand	1/8"	1/6"	OFF
12-1	\ Formatlänge eingestellt auf 12"			OFF
12-2	/			OFF
13-1	1" Vorschub am Blattende	unwirksam	wirksam	ON
13-2	Steuercode kompatibel zu		IBM/Schneider	OFF
14-1	NLQ	unwirksam	wirksam	OFF
14-2	\			OFF
15-1	- Baudrate eingestellt auf 9600			OFF
15-2	/			OFF
16-1	Parität ein/aus	keine Parität	Parität ein	ON
16-2	Parität gerade/ungerade	gerade	ungerade	OFF
17-1	Protokoll	XON-XOFF	DTR	OFF
17-2	Betriebsart (XON/XOFF)	Halbduplex	Duplex	OFF
18-1	Datenbits	8	7	ON
18-2	Stopbits	2	1	ON

Schriftarteneinstellung	DIL-Schalter	10-1	10-2
PICA	1/10"	OFF	OFF
ELITE	1/12"	ON	OFF
komprimiert	1/17"	ON	ON
Fettschrift	1/10"	OFF	ON

Formatlänge	DIL-Schalter	12-1	12-2
12,0"		OFF	OFF
11,0"		ON	OFF
6,0"		OFF	ON
5,5"		ON	ON

Datenrate in Baud	DIL-Schalter	14-2	15-1	15-2
9600		OFF	OFF	OFF
4800		OFF	ON	OFF
2400		ON	ON	OFF
1200		OFF	OFF	ON
600		ON	OFF	ON
300		OFF	ON	ON
200		ON	OFF	OFF
150		ON	ON	ON

Die Programmnummer des Druckers kann ermittelt werden durch Betätigen der Taste Line-Feed-Vorwärts vor dem Einschalten des Druckers und Festhalten, bis der Zeichensatz ausgedruckt wird. Nach dem Loslassen geht der Drucker in den normalen Betriebsmode über. Wird die Taste länger festgehalten, dann wird der Zeichensatz mehrfach wiederholt.

Das Interfacekabel hatte folgende Verbindungen:

9pol. Subm.-D Stecker am ZWERG11A	Signal	Signal am Drucker	V24- Signal	26pol. EFS Stecker am K6314
1	DCD	DTR	108	B08
2	RxD	TxD	103	A03
3	TxD	RxD	104	B04
4	-12V	DTR		
5	GND	GND	102	A01
7	+12V	RTS		

Folgende Signale sind am Drucker miteinander zu verbinden:

Signal am Drucker	V24- Signal	26pol. EFS Stecker am K6314
RTS	105	A05
CTS	106	B06
DSR	107	A07
DCD	109	A09

Um dieses Kabel auch zum direkten Anschluß an eine serielle Schnittstelle des Rechners verwenden zu können, sind am 9poligen SubD-Stecker noch folgende Signale miteinander verbunden:

9pol. Subm.-D Stecker	Signal
1	DCD \
6	DSR -- Brücke 1 (ist mit DTR vom Drucker verbunden!)
8	CTS /
7	RTS _ Brücke 2
9	RI /

Da der verwendete K6314 nur die EPSON-Befehle behandelt, ist noch der Treiber IBM-EPS.EXE installiert wurden. Das erfolgt durch Eingabe von

IBM-EPS 1 oder lh IBM-EPS 1

wodurch der Treiber für LPT1 installiert wird. Unter den angegebenen Bedingungen konnte der Test mit DTR-Protokoll erfolgreich durchgeführt werden. Danach wurde der K6314 auf XON-XOFF-Protokoll umgeschaltet und ebenfalls erfolgreich getestet. Auch die Verwendung eines K6313 mit der Programmnummer 3.29-3-A5/04, eingestellt auf IBM-Mode, konnte erfolgreich mit beiden Protokollen getestet werden.

3.4.1.8. Einsatzmöglichkeiten des Interfaceconverters

Die Anwendung des Interfaceconverters ist vor allem bei LAPTOP's von Interesse, da diese häufig nur eine serielle Schnittstelle besitzen, welche dann schon durch die Mouse belegt ist. Außerdem ist keine Uminstallierung der Software nötig, wenn beim Kunden nur Drucker mit seriellem Interface vorhanden sind. Ein weiterer Anwendungsfall entsteht, wenn die benutzte Software direkt zum LPT1-Port zugreift, ohne dabei die DOS- oder BIOS-Dienste zu benutzen. In solchen Fällen ist eine Änderung des Druckerports nur durch eine neue Installation des Programms möglich, wenn dabei diese Option vorhanden ist.

3.4.1.9. Quellliste des Programms CEN232X.A

***** Beispiel für Interface-Wandler *****

```

*
* Date          Version      What                      Who
* 07.08.94      1.00         create                   Scö
* -----
*
*                               - Notes -
* Wandlung einer Druckerschnittstelle mit Centronics Interface in eine
* RS232 Schnittstelle mit XON - XOFF Protokoll.
*
* 1. Festlegung der Signale
*
* 25pol.
* Subm.-D
* männlich
* E/A   7400 Zwerg11A   Bemerkung
*
* 01 E ~STROB 5,10 8 STRA 1.24 D1 bis D8 wird übernommen -> BUSY ein.
*          --> 8 PA1 3.11
*
* 02 E D1          PC0 1.11 \
* 03 E D2          PC1 1.12 |
* 04 E D3          PC2 1.13 |
* 05 E D4          PC3 1.14 |_ Druck und Steuerzeichen
* 06 E D5          PC4 1.15 |
* 07 E D6          PC5 1.16 |
* 08 E D7          PC6 1.17 |
* 09 E D8          PC7 1.18 /
*
* 10 A ~ACK 1      PB0 1.08 Impuls, wenn Drucker wieder bereit
* 11 A BUSY 11<-13 PB1 1.09 Drucker besetzt - Anzeige
*
* 12 A PE          PB2 1.20 Pappierende Meldung vom Drucker
* 13 A SELECT      PB3 1.21 ON-LINE Meldung vom Drucker
* 14 E ~AUTO F      PE0 3.02 Drucker auf AUTO-FEED schalten
* 15 A ~ERR        PB4 1.05 Fehlermeldung des Druckers
* 16 E ~RESET      PE1 3.04 Drucker rücksetzen
* 17 E ~SLK IN     PE2 3.06 Drucker ON-LINE schalten
*

```

```

* 18 GND          1.01
*                1.02
* : :            1.26
* . .            3.03
*                3.07
* 25 GND          3.25
*
*                PA4  3.17  1, wenn Drucker im Stop
*                PA5  3.19  1, wenn kein DC1 oder DC3 empfangen
*                PA6  3.21  1, wenn letzte Zeichen kein DC1 oder DC3
*                PA7  3.23  1, wenn SCI-OR Error im laufenden Druck
*
* -----
*                - Usage -
* -----
* Copyright (C) Dr. Siegmur Schöne
*****

```

```
#include <target.h>
```

```

* Interrupt durch STRA wird wie /IRQ durch Sprung zu Adresse auf BFF2,F3
* behandelt. (BFF2: 00 EE)

```

```

SCIVEC      equ    VECTBL
STRAVEC     equ    VECTBL+42      * vect=tbl+offset=c4+3*14

PULAE       equ    160
PULAES      equ    PULAE-6       * Sperren BUSY, wenn >= PULAES Zeichen
PULAEF      equ    60           * Freigabe BUSY, wenn < PULAEF Zeichen

                bss
                org    $0000

poi_in      ds.w   1             * Empfangs-Pointer
poi_out     ds.b   1             * Anzahl Zeichen im Puffer
dr_kom      ds.b   1             * letztes Druckerkommando an Port E
dc1dc3      ds.b   1             * letztes Zeichen vom Drucker
lae_rd      ds.b   1             * busy on off

pu_anf      ds.b   PULAE        * Empfangspuffer
pu_end      equ    *

stk         ds.b   27           * MAX Stacktiefe des PG=9Byte new_mod +
2INT
stack       equ    *           * muß kleiner $00c4 sein!!!

                text
                org    PROG

start       lds    # $00c3      * stack! ($00c3)

INTERRUPT

LOOPOFF

                clr    ddrC     * Port C alles Eingang
                ldaa   #$f9
                staa   portB    * Port B ~ACK und ~ERR 1 setzen
                ldaa   #$80
                staa   pactl    * PA7 ist Output
                ldaa   #$08     * PA3 = DTR setzen, damit -12V

```

```

        staa    porta
*   Arbeitszellen initialisieren

        ldx    #pu_anf
        stx    poi_in
        stx    poi_out
        clr    anz_zei
        ldaa   #3
        staa   dr_kom           * no AF, no RESET, ON-LINE
        ldaa   #$11
        staa   dc1dc3
        ldaa   #PULAES
        staa   lae_rd

*   Interrupt für STRA und SCI initialisieren

        ldaa   #$7e
        staa   STRAVEC         * jmp code = $7e
        ldy    #pint
        sty    STRAVEC+1      * STRA int handler
        staa   SCIVEC         * jmp code = $7e
        ldy    #sciint
        sty    SCIVEC+1       * SCI int handler

        ldaa   #$62           * IntEN, C_out=open-drain + pos_FI
        staa   pioc

*   SCI initialisieren

        ldaa   #$30           * f/(13*1) = 9600 Baud
        staa   baud
        ldaa   #$2c           * Freigabe RxINT, Tx und Rx
        staa   sccr2

        ldx    #REG           * Anfang Registerblock

        cli                   * ints on

*   Centronics-Wartezyklus   *****

cen_zy    ldab   anz_zei
          cmpb  #0
          beq   no_zei

          bsr   ze_i_druk

          bra   old_mod

*   Testen von Port E --> Druckermodus einstellen

no_zei    equ    *
          bclr  $80, oportax    * PA7 RD-Fehler clr
          ldaa  porte
          cmpa  dr_kom
          beq   old_mod         * kein neuer MODE

          bsr   new_mod

old_mod   ldab   anz_zei
          cmpb  lae_rd          * Pufferlänge

```

```

        bcc    stop                * BUSY nicht freigeben !!

        brset 2,oporta,x,cen_zy   * STROB noch aktiv
        brset 2,oporta,x,cen_zy   * STROB noch aktiv
        brset 2,oporta,x,cen_zy   * STROB noch aktiv
        brset 2,oporta,x,cen_zy   * STROB noch aktiv

*      BUSY rücksetzen, wenn noch gesetzt und ~ACK Impuls

        brset 2,oportb,x,cen_zy   * BUSY ist nicht gesetzt

        bclr  1,oportb,x          * negative Flanke ~ACK
        bset  2,oportb,x          * BUSY aus
        bset  1,oportb,x          * positive Flanke ~ACK

Zeichen      ldaa  #PULAES        * Sperren BUSY, wenn >= PULAES

        staa  lae_rd
        bra   cen_zy

stop         ldaa  #PULAEF        * Freigabe BUSY, wenn < PULAEF Zeichen
        staa  lae_rd
        bra   cen_zy

*      Centronics-Wartezyklus Ende *****

*****
*
*      Ein Zeichen an Drucker über RS232 senden *
*
*****

zei_druck   equ   *
            jsr   dr_sta
            inca
            beq   puout           * Zeichen aus Puffer drucken
            rts                  * Drucker nicht bereit

puout       dec   anz_zei
            ldy   poi_out
            iny
            cpy   #pu_end
            bne   no_endr

no_endr     ldy   #pu_anf
            sty   poi_out
            ldaa  0,y
            jmp   sciput

*****
*
*      Neues MODE-Kommando bearbeiten *
*
*****

new_mod     equ   *
            bita  #$2             * RESET ?
            beq   reset          * Ja
            tab
            eora  dr_kom         * Bits, die sich verändert haben
            stab  dr_kom         * neuen Status merken
            bita  #$1           * hat sich ~AUTO F verändert ?

```

```

no_af      beq    no_af      * Nein
           bitb   #$1
           beq    af_on    * AUTO FEED Mode ein
           bra    af_off    * AUTO FEED Mode aus
           bita   #$4      * hat sich ~SLK IN verändert ?
           beq    no_sli   * Nein
           bitb   #$4
           beq    sli_on   * Drucker ON-Line schalten
no_sli     bra    sli_off  * Drucker OFF-Line schalten
           rts

reset      bset   2,oportb,x  * busy on setzen
           bsr   druck0      * Beenden einer Grafiksequenz
           ldaa  #$1b
           bsr   druck
           ldaa  #"@"
           bsr   druck
           sei

res_off    brclr  2,oporte,x,res_off  * Warten auf RESET aus
           jmp   start

af_on      equ    *          * ESC 5 1 senden
           ldaa  #$1b
           bsr   druck
           ldaa  #"5"
           bsr   druck
           ldaa  #"1"
           bra   druck•

af_off     equ    *          * ESC 5 0 senden
           ldaa  #$1b
           bsr   druck
           ldaa  #"5"
           bsr   druck
           ldaa  #"0"
           bra   druck

sli_on     equ    *          * DC1 senden
           ldaa  #$11
           bra   druck

sli_off    equ    *          * DC3 senden
           ldaa  #$13
           bra   druck

*****
*
* Beenden einer Grafiksequenz durch Senden von 0
*
*****

druck0     pshy
druck01    ldy   #1920      * Maximale Punktzahl in einer Sequenz
           ldaa  #0        * keine Nadel oder NOP
           bsr   druck
           dey
           bne  druck01
           puly
           rts

```

*
* Drucker Status *
*

dr_sta	tst	scsr	* SCI Sender bereit ?
	bpl	no_rdy2	* NEIN
	ldaa	porta	* PA0=DCD=DTR vom Drucker
	anda	#1	
	beq	rdy1	* DTR vom Drucker nicht aktiv
no_rdy	equ	*	
	bset	\$10,oporta,x	* PA4 set, wenn im Stop
no_rdy2	clr	rts	* A=00, wenn Drucker besetzt
rdy1	ldaa	dc1dc3	
	cmpa	#\$11	
	bne	no_rdy	* zuletzt wurde DC3 empfangen
	bclr	\$50,oporta,x	* PA4+6 clr, wenn im Start
	ldaa	#\$ff	* A=FF, wenn Drucker bereit
	rts		

*
* Ein Zeichen an Drucker über RS232 senden *
*

druck	equ	*	* Drucken mit Warten
	psha		
dr_nord	bsr	dr_sta	
	inca		
	bne	dr_nord	
	pula		

*
* Ausgabe seriell *
*

sciput	tst	scsr
	bpl	sciput
	staa	scdr
	rts	

* Interruptbehandlung für STRA = STROB vom Centronics Interface

pint	bclr	2,oportb,x	* busy on setzen
	ldaa	pioc	* Res Interruptflag
	ldaa	portcl	* Daten lesen
	ldy	poi_in	
	iny		
	cpy	#pu_end	
	bne	no_end	
	ldy	#pu_anf	
no_end	sty	poi_in	
	staa	0,y	
	cpy	poi_out	* Pufferüberlauf ??

no_err	bne bclr inc rti	no_err \$10,oportb,x anz_zei	* ~ERR setzen * Anzahl Zeichen im Puffer +1
* Interruptbehandlung für Rx vom SCI			
sciint	equ ldab andb beq ldaa bitb bne bitb beq bset staa bra	* scsr #\$2e no_dc scdr #6 no_dc #8 no_rder \$80,oporta,x dc1dc3 inv_dc	* RDRF, OR, NF, FE * kein Zeichen da, kein Fehler * NF und FE * BREAK, oder Leitung gestört * OR * PA7 set, wenn RD-Fehler * vorletztes Zeichen speichern * und invertieren
no_rder	cmpa beq cmpa beq bset bra	#\$13 dc_ok #\$11 dc_ok \$60,oporta,x no_dc	* DC3 = Stop * DC1 = Start * PA5+6 set, wenn nicht DC1 oder DC3
inv_dc	ldaa	#2	
dc_ok	eora	dc1dc3	* DC1 in DC3 oder DC3 in DC1
no_dc	staa rti	dc1dc3	

* Ende IFW Cen -> RS232