



SER2

Baugruppe für den NDR-Klein-Computer zur
Bereitstellung von 2 seriellen RS232-Schnittstellen
mit maximal 115200 Baud.

Stand: März 2007

Copyright © by Gerald Ebert

Wichtiger Hinweis:

Die in dieser Anleitung wiedergegebenen Schaltungen und Verfahren werden ohne Rücksicht auf die Patentlage oder Lizenzrechte Dritter mitgeteilt. Sie sind ausschließlich für private Zwecke und Lehrzwecke bestimmt und dürfen nicht gewerblich genutzt werden. *)

Alle Schaltungen und technische Angaben in dieser Anleitung wurden vom Autor sorgfältig erarbeitet bzw. zusammengestellt. Trotzdem sind Fehler nicht auszuschließen. Daher kann der Autor weder eine Garantie noch die juristische Verantwortung oder irgend eine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernehmen. Für die Mitteilung eventueller Fehler ist der Autor jederzeit dankbar.

Die Rechte an Firmennamen, Logos und Warenzeichen, die in dieser Anleitung genannt werden, liegen bei den jeweiligen Inhabern.

*) Bei gewerblicher Nutzung ist vorher die Genehmigung des möglichen Lizenz- oder Rechteinhabers einzuholen.

Inhalt:

Vorwort	4
1 Kurzbeschreibung der Funktion	4
2 Technische Daten	4
3 Prinzipbeschreibung	5
3.1 Blockschaltbild SER2	5
3.2 Beschreibung des Blockschaltbildes	6
3.3 Auszüge aus dem Datenblatt des XR88C681	6
3.4 Das Programmbeispiel zur DUART 88C681	8
4 Aufbauanleitung	8
4.1 Umgang mit IC's	8
4.2 Stückliste	9
4.3 Aufbau Schritt für Schritt	9
4.4 Belegung von Steckleisten, Buchsen und Jumpers	11
5 Literaturhinweise und –nachweise	11
Schaltplan	13
Bestückungsplan	14
Layout Bestückungsseite mit Aufdruck	15
Layout Bestückungsseite	16
Layout Lötseite	17
Programmbeispiel zur Steuerung der Baugruppe	18

Vorwort

Die bisher beim NDR-Klein-Computer (NKC) verfügbare Serielle Bausgruppe SER funktioniert zwar einwandfrei, ist aber mit max. 19200 Baud nicht mehr zeitgemäß. Will (muß) man dann auch noch Dateien zwischen PC und NKC austauschen, muß man viel Geduld und Zeit haben.

Beim Kramen in alten IC-Kisten bin ich dann auf die DUART XR88C681 gestoßen. Dieser Baustein kann 115 kBaud Übertragungsrate, also 8-fache Geschwindigkeit im Vergleich zur SER-Baugruppe, verfügt über zwei Kanäle und ist mit wenig Aufwand in den NKC zu integrieren. Das Ergebnis seht Ihr auf den folgenden Seiten dieses Handbuchs.

Eine Platine werde ich aus Kostengründen erst mal nicht produzieren lassen. Meine gefädelt Variante tut's auch.

Da nichts auf dieser Welt perfekt sein kann, werden sich auch hier Fehler eingeschlichen haben. Wenn Sie einen entdecken, so teilen Sie mir dies mit einer möglichst genauen Beschreibung im NKC-Forum (siehe Literatur) mit.

Danke.

1 Kurzbeschreibung der Funktion

Die SER2-Baugruppe stellt eine Modernisierung der bisherigen SER-Baugruppe dar. Ziel der Entwicklung ist zum einen eine einfache Anbindung an den PC zum Datenaustausch mit erträglicher Übertragungsgeschwindigkeit, als auch der Bau einer RS-232-Schnittstelle mit heute noch erhältlichen Bauteilen. Leider konnte hier keine Kompatibilität zur SER-Baugruppe beibehalten werden. Ein Beispiel zur Ansteuerung der DUART (**D**ual **U**niversal **A**synchronous **R**eceiver and **T**ransmitter) befindet sich im Anhang. Ich empfehle jedoch einen Blick ins Datenblatt des Bausteins XR88C681 zu werfen, um die Funktionen zu verstehen.

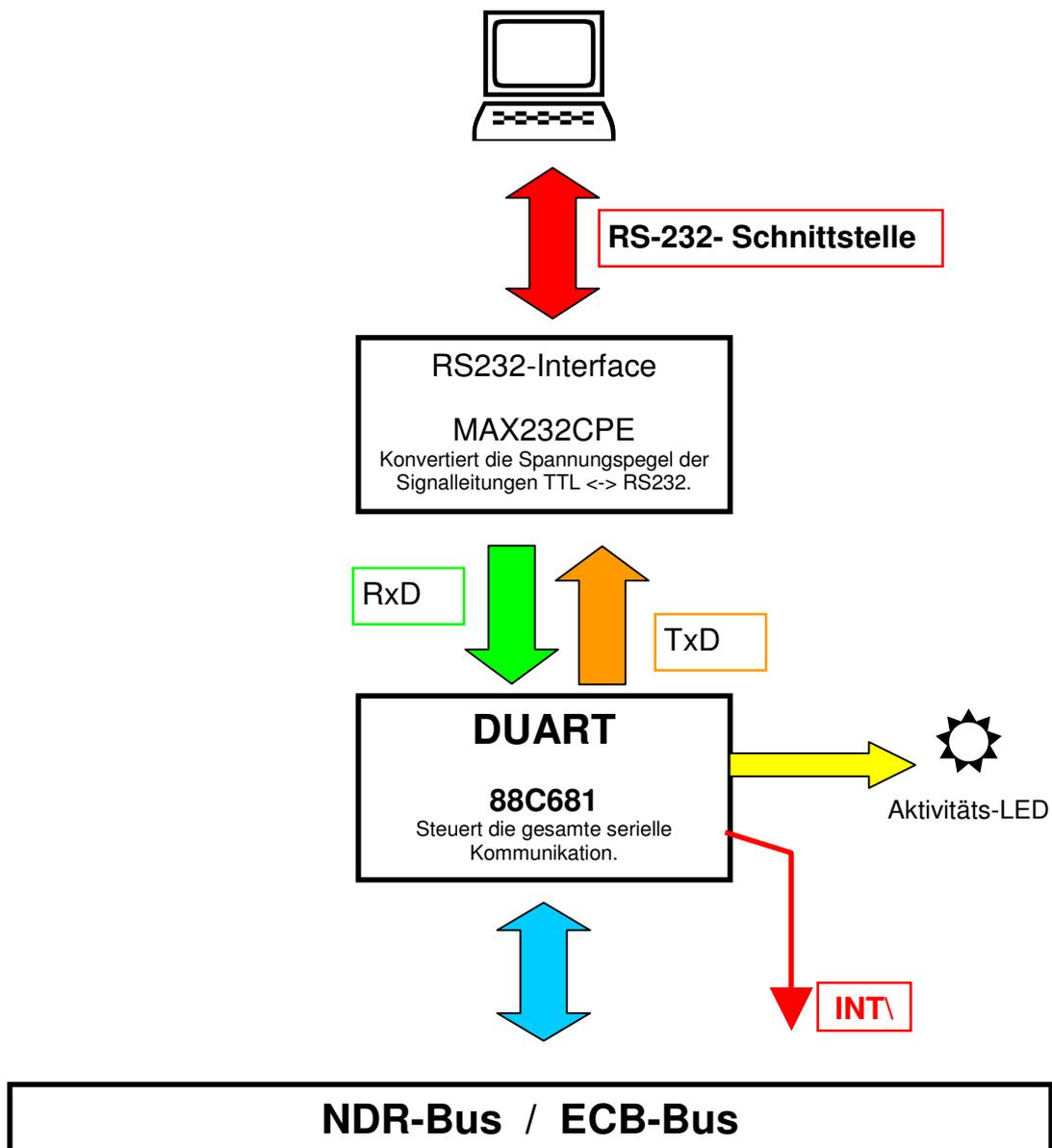
2 Technische Daten

- Europakarte 160 x 100 mm doppelseitig mit Bestückungsaufdruck
- NDR-Bus und ECB-Bus
- Stromaufnahme ca. 160 mA (Bestückung mit LS-Typen)
- DIN-Buchsen zum Anschluß von RS232 Schnittstellen
- Zusätzlich 2x5 polige Stiftheimungen um die D-SUB 9pol-Buchsen wahlfrei in einem Gehäuse plazieren zu können
- Aktivitätsanzeige (TxD) durch LED
- INT\ kann ausgelöst werden, wenn Daten gesendet oder empfangen werden. Mittels Jumper einstellbar.

3 Prinzipbeschreibung

Die eigentliche Schnittstellenkontrolle übernimmt auf dieser Baugruppe der DUART-Baustein XR88C681. Er stellt zusammen mit dem Interfacedriver MAX232CPE die RS-232-Schnittstelle dar. Um auch ältere Bausteine dieser IC-Familie verwenden zu können (ich habe noch ein paar exotische Varianten davon) wurde auf Hardwarehandshake mittels RTS/CTS verzichtet, was auch heutzutage kaum noch benutzt wird. Das Layoutdesign berücksichtigt demzufolge die aktuellen 40-pin Gehäusetypen als auch ältere 28-pin Gehäusetypen. Die Belegung der RS-232-Steckverbindung hat Masterlayout, wie beim PC. Sie benötigen also ein Nullmodemkabel, wenn Sie Ihren PC an die SER-Baugruppen des NKC anschließen wollen.

3.1 Blockschaubild SER2



3.2 Beschreibung des Blockschaltbildes

Der DUART ist der zentrale Baustein der Baugruppe. Er wickelt das gesamte Hardwareprotokoll der RS-232-Schnittstelle ab. Nach einer Initialisierungssequenz, in der die Eigenschaften des Hardwareprotokolls eingestellt werden, können sofort Daten , durch Schreiben in ein Register, gesendet werden. Der DUART wandelt dabei die Bits des Senderegisters in einen serielle Datenstrom mit einem Protokollrahmen um und gibt diesen an den Interfacebaustein MAX232CPE weiter. Der Interfacebaustein wandelt das serielle Signal nur noch von TTL-Pegel in RS-232-Pegel um bzw. zurück. Umgekehrt kann der DUART sofort Daten empfangen, die in einer 3-Byte-FIFO (First In First Out) zwischen gespeichert werden. Die CPU holt die empfangenen Daten vom Empfangsregister ab, bis die FIFO leer ist.

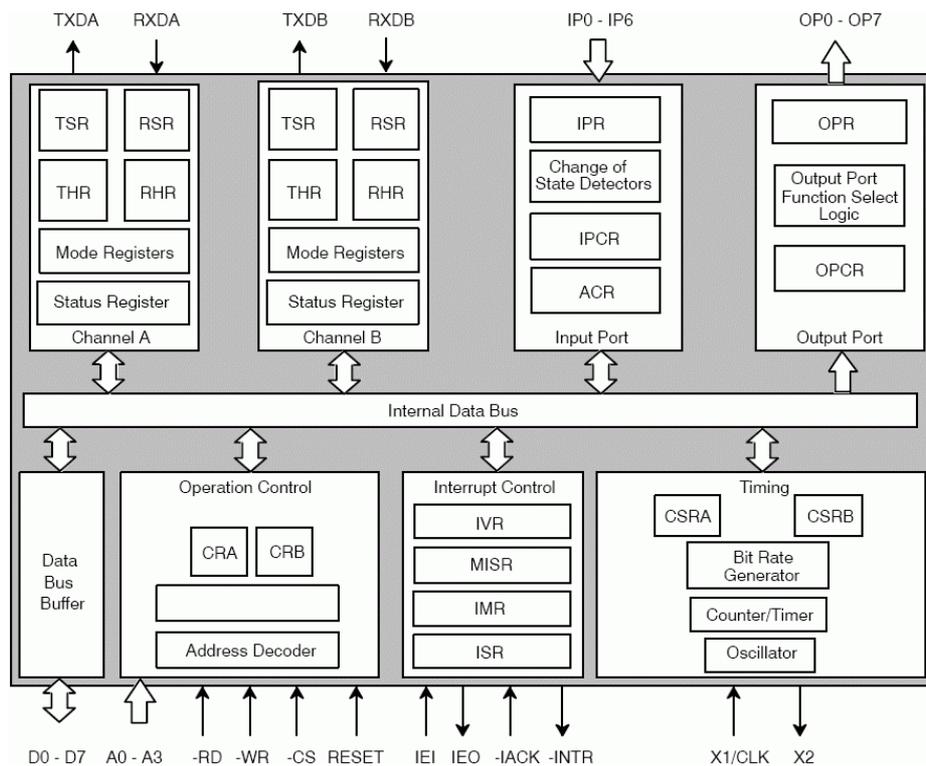
Der DUART kann einen Interrupt auslösen, wenn bei der Initialisierung im Interrupt-Mask-Register einer der 8 möglichen Auslösebedingungen freigeschaltet werden. Dazu muß auch der Jumper auf ST8 gesetzt sein. Die Bitmaske für die Auslösebedingungen finden Sie im Datenblatt.

3.3 Auszüge aus dem Datenblatt des XR88C681

Eigenschaften des XR88C681:

- Zwei unabhängige Vollduplex Kanäle
- Asynchroner Empfänger und Sender
- 4-fach gepufferter Empfänger und zwei gepufferter Sender
- Programmierbare Stop-Bits in 1/16 Schritten
- Interner Bitratengenerator
- Unabhängige Bitrateneinstellung für Empfänger und Sender
- Maximale Bitrate 115200 Baud bei 3,6864 MHz
- Multifunktionaler 16-Bit Zähler/Timer
- Interrupt-Ausgabe mit 8 maskierbaren Interrupt Bedingungen
- Standby Modus zur Stromreduzierung

Blockschaltbild des XR88C681:



Die Registeradressen des XR88C681

Address (Hex)	Read Mode Registers		Write Mode Registers	
	Register Name	Symbol	Register Name	Symbol
00	Mode Register, Channel A	MR1A, MR2A	Mode Register, Channel A	MR1A, MR2A
01	Status Register, Channel A	SRA	Clock Select Register, Channel A	CSRA
02	Masked Interrupt Status Register	MISR	Command Register A	CRA
03	Rx Holding Register, Channel A	RHRA	Tx Holding Register, Channel A	THRA
04	Input Port Change Register	IPCR	Auxiliary Control Register	ACR
05	Interrupt Status Register	ISR	Interrupt Mask Register	IMR
06	Counter/Timer Upper Byte Register	CTU	Counter/Timer Upper Byte Register	CTU
07	Counter/Timer Lower Byte Register	CTL	Counter/Timer Lower Byte Register	CTL
08	Mode Register, Channel B	MR1B, MR2B	Mode Register, Channel B	MR1B, MR2B
09	Status Register, Channel B	SRB	Clock Select Register, Channel B	CSRB
0A	RESERVED		Command Register, Channel B	CRB
0B	Rx Holding Register, Channel B	RHRB	Tx Holding Register, Channel B	THRB
0C	Interrupt Vector Register	IVR	Interrupt Vector Register	IVR
0D	Input Port	IP	Output Port Configuration Register (OP0 - OP7)	OPCR
0E	Start Counter/Timer Command	SCC	Set Output Port Bits Command	SOPBC
0F	Stop Counter/Timer Command	STC	Clear Output Port Bits Command	COPBC

3.4 Das Programmbeispiel zur DUART 88C681

Im Programmbeispiel im Anhang habe ich eine Minimalinitialisierung der DUART programmiert. Die Register müssen mit den entsprechenden Werten gesetzt werden. Der Baustein kann aber noch mehr. Beispielsweise kann die Initialisierung so erweitert werden, daß der Empfang von Daten vollständig interruptgesteuert Abgewickelt werden kann. Das hat den Vorteil, daß im Gegensatz zum Pollingbetrieb der Schnittstelle nur minimal Rechenzeit der CPU verbraucht wird und keine Daten verloren gehen, wenn die CPU mal etwas länger mit anderem beschäftigt ist; z.B. Ausgabe der empfangenen Daten auf dem Monitor.

Die Routinen zum Senden und Empfangen von Daten können Sie einfach übernehmen, unabhängig für welche Betriebsart Sie sich entschieden haben.

Ich habe den Quelltext möglichst ausführlich mit Dokumentaren versehen, damit Sie den Algorithmus nachvollziehen können.

Wenn Sie die Funktionsweise des DUART-Bausteins verstehen wollen, empfehle ich Ihnen noch einmal das ausführliche Studium des Datenblattes (im Internet leicht zu finden).

4 Aufbauanleitung

4.1 Umgang mit IC's

CMOS-Bausteine sind hochempfindlich gegen elektrostatische Aufladung! Bewahren oder Transportieren Sie CMOS-Bausteine nur auf leitenden Schaumstoff! Alle Pins müssen kurzgeschlossen sein.

Achten Sie darauf, daß Sie Verbindung mit einer Erdungsmöglichkeit haben, bevor Sie mit diesen Bausteinen arbeiten. Geeignete ESD-Artikel gibt es im Fachhandel.

4.2 Stückliste

Anzahl	Kennung	Bauteil (klassisch)	Bauteil (alternativ)
1			Platine
1	IC 1		XR88C681CP28
1	IC 2	74LS245	74HCT245
1	IC 3		MAX232CPE
1	IC 4	74LS688	74HCT688
1	IC 6	74LS04	74HCT04
1	Q1		Quarz 3,6864 MHz
1	C 1		Kondensator 10µF Tantal
4	C 2 – 5,		Kondensator 100nF MKS-2
4	C 7 - 10		Kondensator 0,1 µF Keramik oder MKS-02
4	C11 - 14		Kondensator 1,2nF MKS-2
2	C15, 16		Kondensator 22pF Keramik
2	R 1 - 4		Widerstand 470 Ω
1	R 5, 6		Widerstand 100 Ω
2	R 7, 8		Widerstand 330 Ω
2	R 9, 10		Widerstand 10 kΩ
1	RN 1		Widerstandsnetzwerk 4 x 4,7 kΩ
2	T 1, 2		Transistor BC337
2	D 1, 2		LED 3mm Standard
1	ST 1		Messerleiste DIN 41612 Bauform C 64 pol für ECB-Bus
1	ST 2		Stiftleiste 1 x 36 pol & 1 x 18 pol gew. für NDR-Bus
2	ST 3, 4		Stiftleiste 2 x 5 pol
2	ST 5, 6		D-SUB 9pol-Einbaubuchse
1	ST 8		Stiftleiste 2 x 4 pol
1	ST 7		Stiftleiste 1 x 2 pol
alternativ:			
1	IC 5		XR88C681CP40 oder Ei88C681

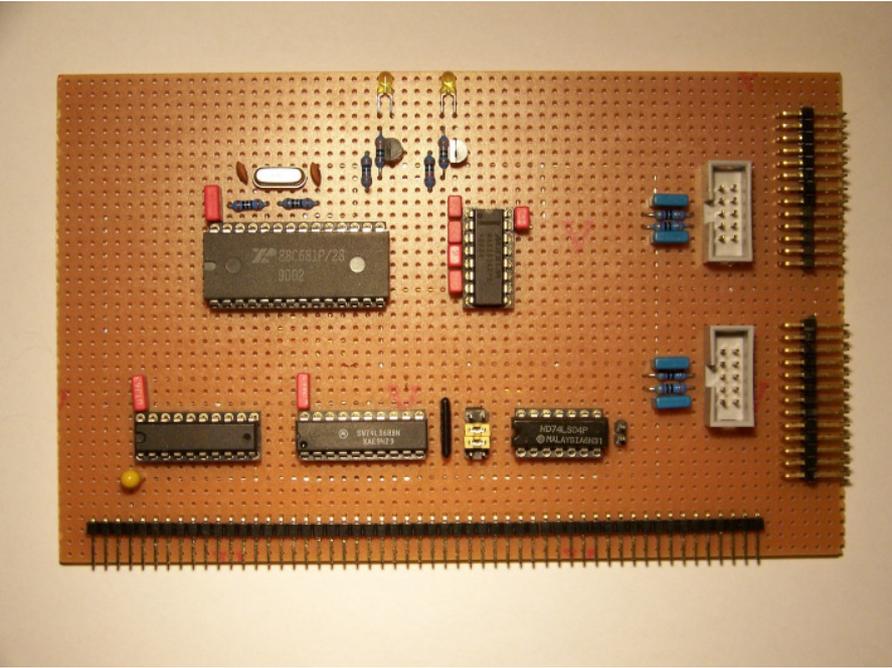
4.3 Aufbau Schritt für Schritt

Entfällt

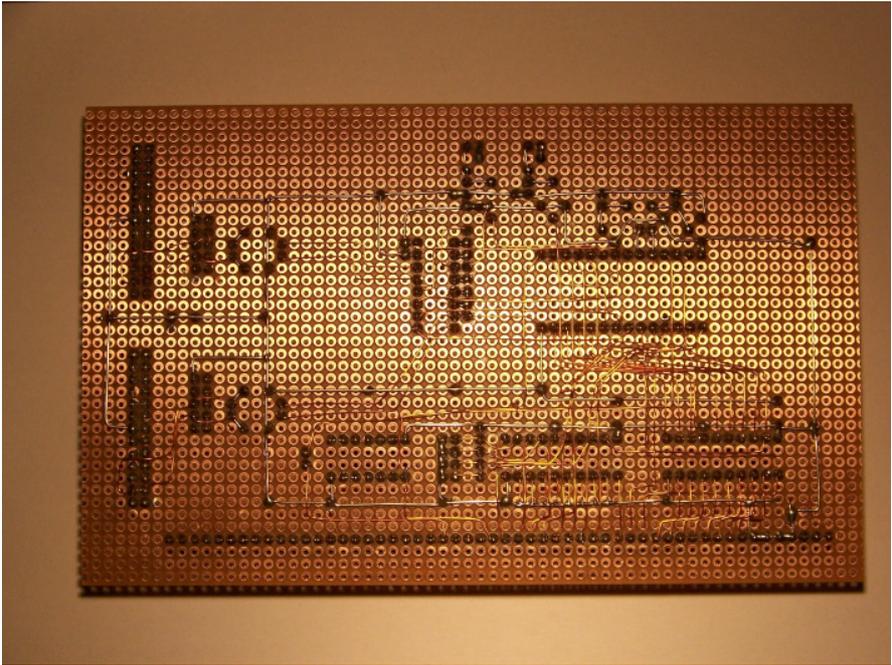
Eine Platine werde ich aus Kostengründen vorerst nicht produzieren lassen. Deshalb spare ich mir eine detaillierte Anleitung.

Die Schaltung ist so einfach, daß mit ein wenig Erfahrung in der Fädeltechnik, die Platine ca. 3-4 Stunden fertig gelötet werden kann.

Ich habe ich mir eine Prototypenplatine möglichst layoutnah gefädelt. Als Anregung zum Nachbau die Bilder meiner Baugruppe:



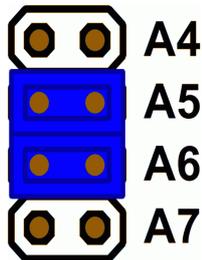
Prototypenplatine gefädelt, Bestückungsseite



Prototypenplatine gefädelt, Lötseite

4.4 Belegung von Steckleisten, Buchsen und Jumpers

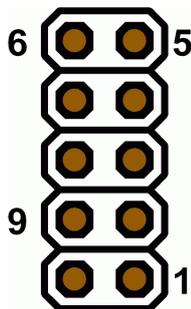
ADDR:



Hier ist die Jumperstellung der Standard-I/O-Adresse für die SER2-Baugruppe (\$90-\$9F) dargestellt. Grundsätzlich kann der I/O-Adressbereich nach den eigenen Wünschen eingestellt werden. Denken Sie aber daran, daß das Grundprogramm dann die Baugruppe nicht mehr unterstützen kann.

Ein gesetzter Jumper bedeutet ein 0-Bit in der Adreßabgabe.

ST3, 4:



PIN	Bezeichnung
1	nc
2	RxD
3	TxD
4	nc
5	GND
6	nc
7	nc
8	nc
9	nc

ST8:



Es wird kein Interrupt ausgelöst.

Der DUART kann je nach Initialisierung einen Interrupt auslösen.

5 Literaturhinweise und –nachweise

Bücher

- ❖ Rolf-Dieter Klein
„Rechner modular“
Der NDR-Klein-Computer – selbstgebaut und programmiert
Franzis-Verlag, München. ISBN 3-7723-8721-7
- ❖ Rolf-Dieter Klein
“Die Prozessoren 68000 und 68008“
Rechnerarchitektur und Sprache im NDR-Klein-Computer
Franzis-Verlag, München. ISBN 3-7723-7651-7

Datenblätter

- * EXAR XR88C681, Rev. 2.11 vom Juni/2006
- * MAXIM Multichannel-RS-232-Drivers, Rev. 9 vom Mai/2000

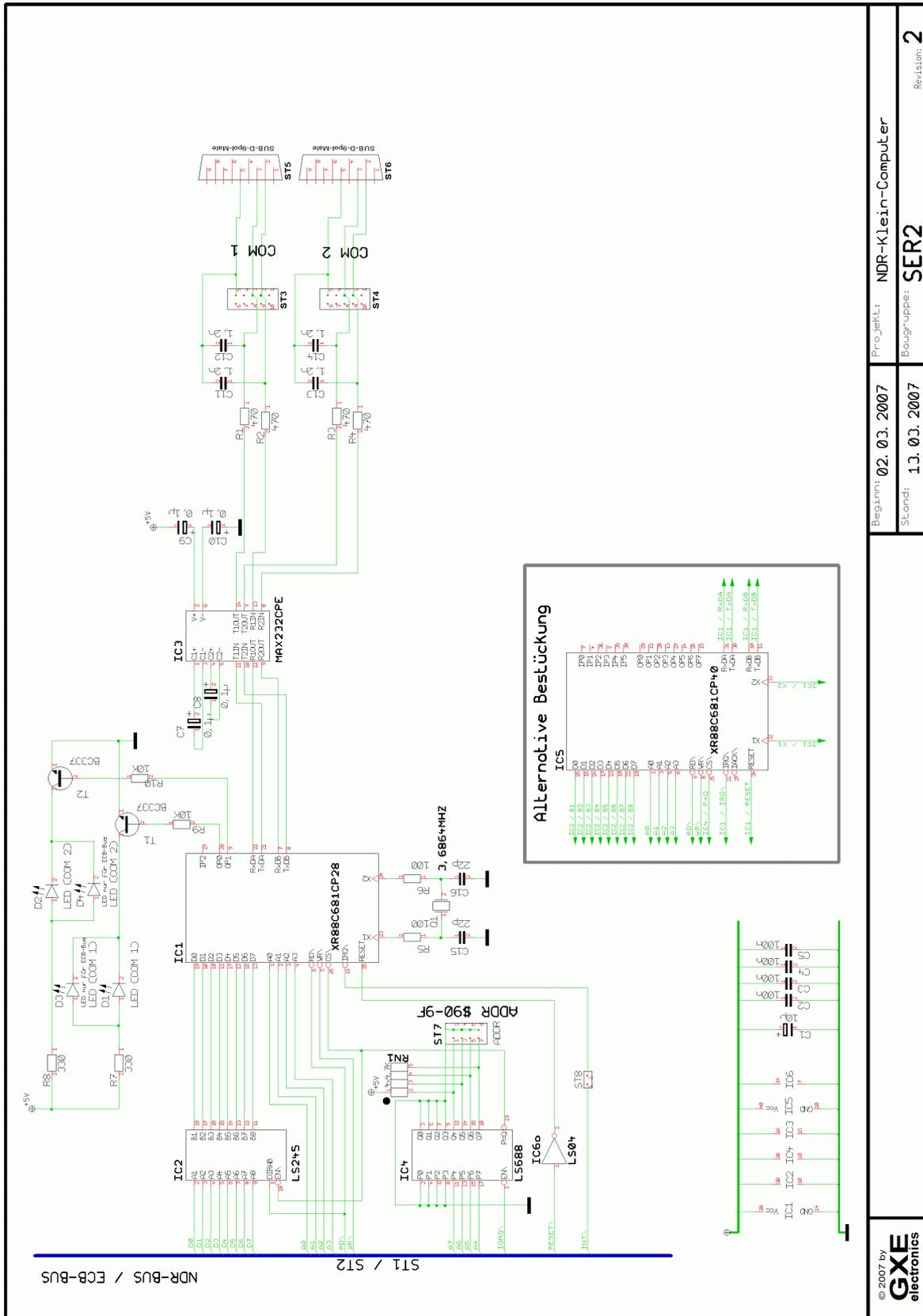
Quellen im Internet

<http://www.exar.com>
<http://www.maxim-ic.com>

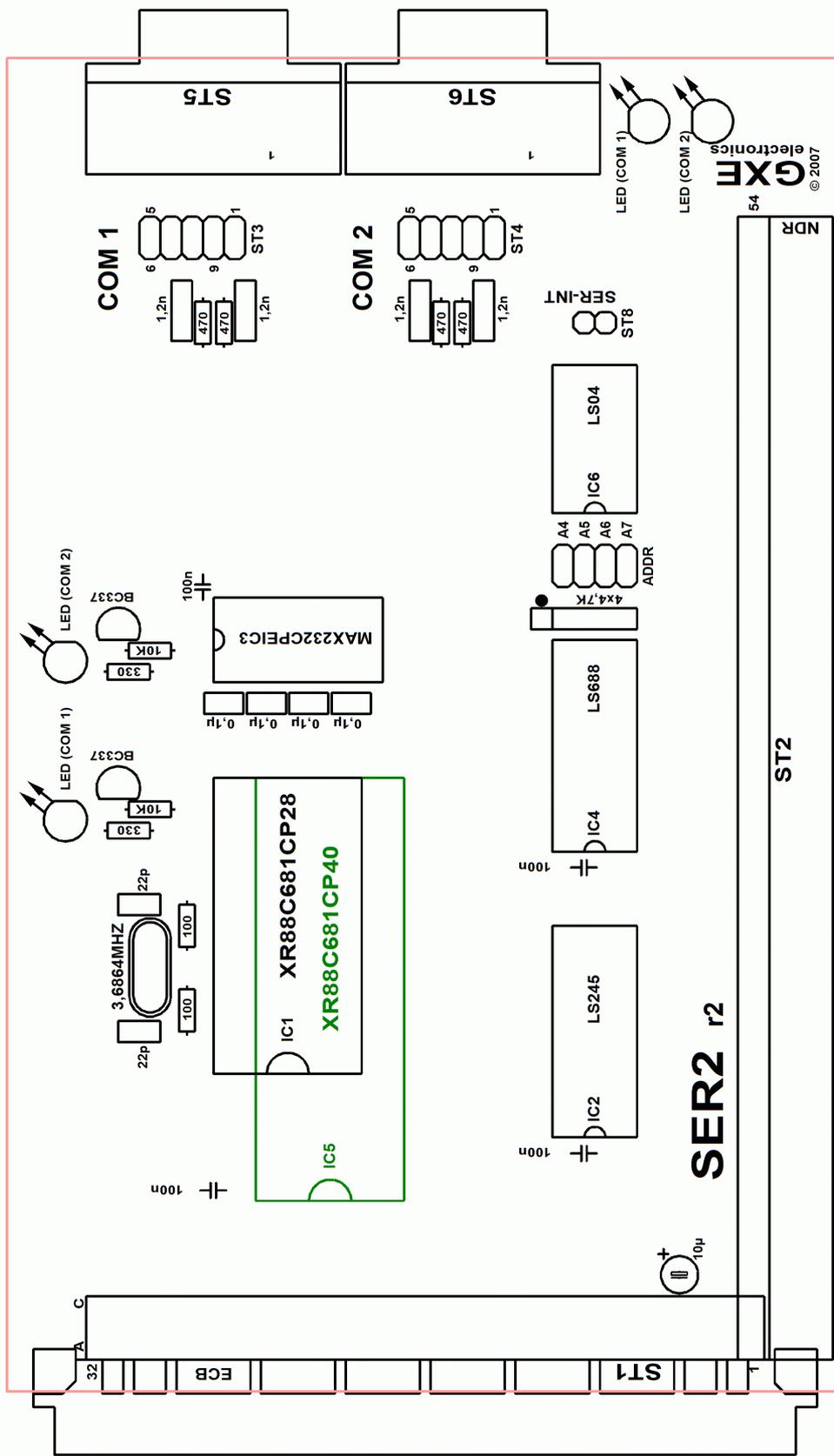
NKC-Forum

<http://www.dr crazy.de/forum/>

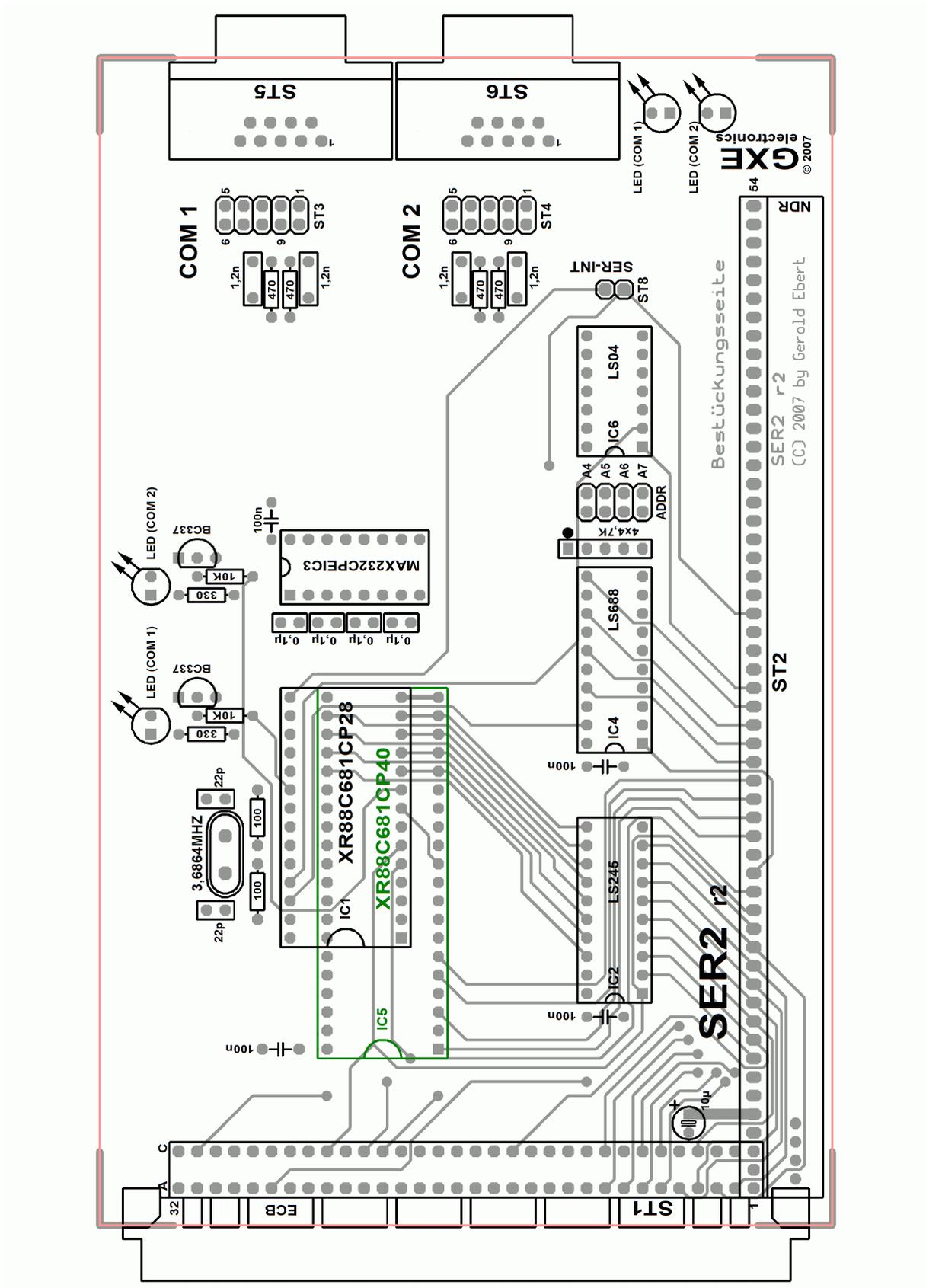
Schaltplan



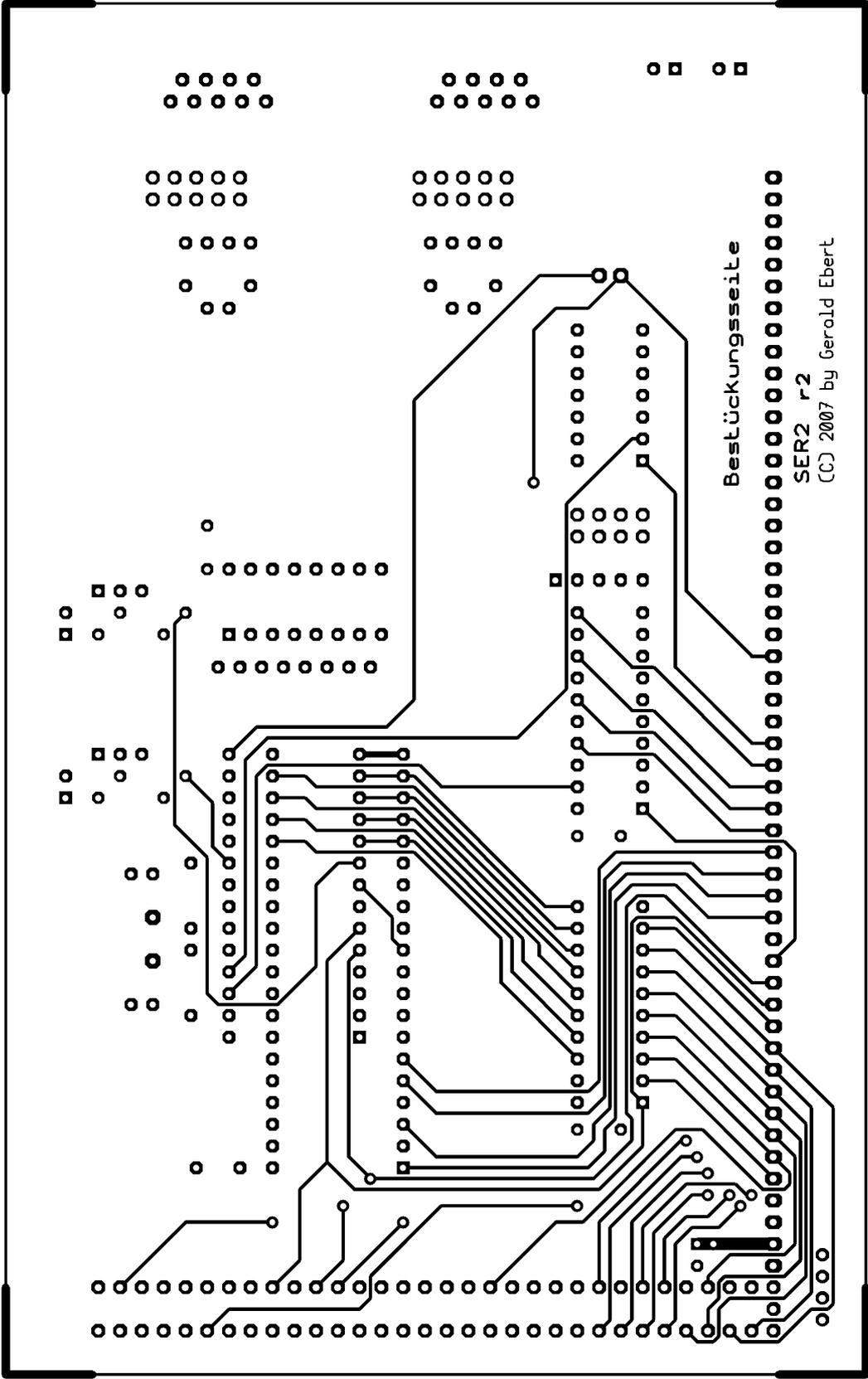
Bestückungsplan



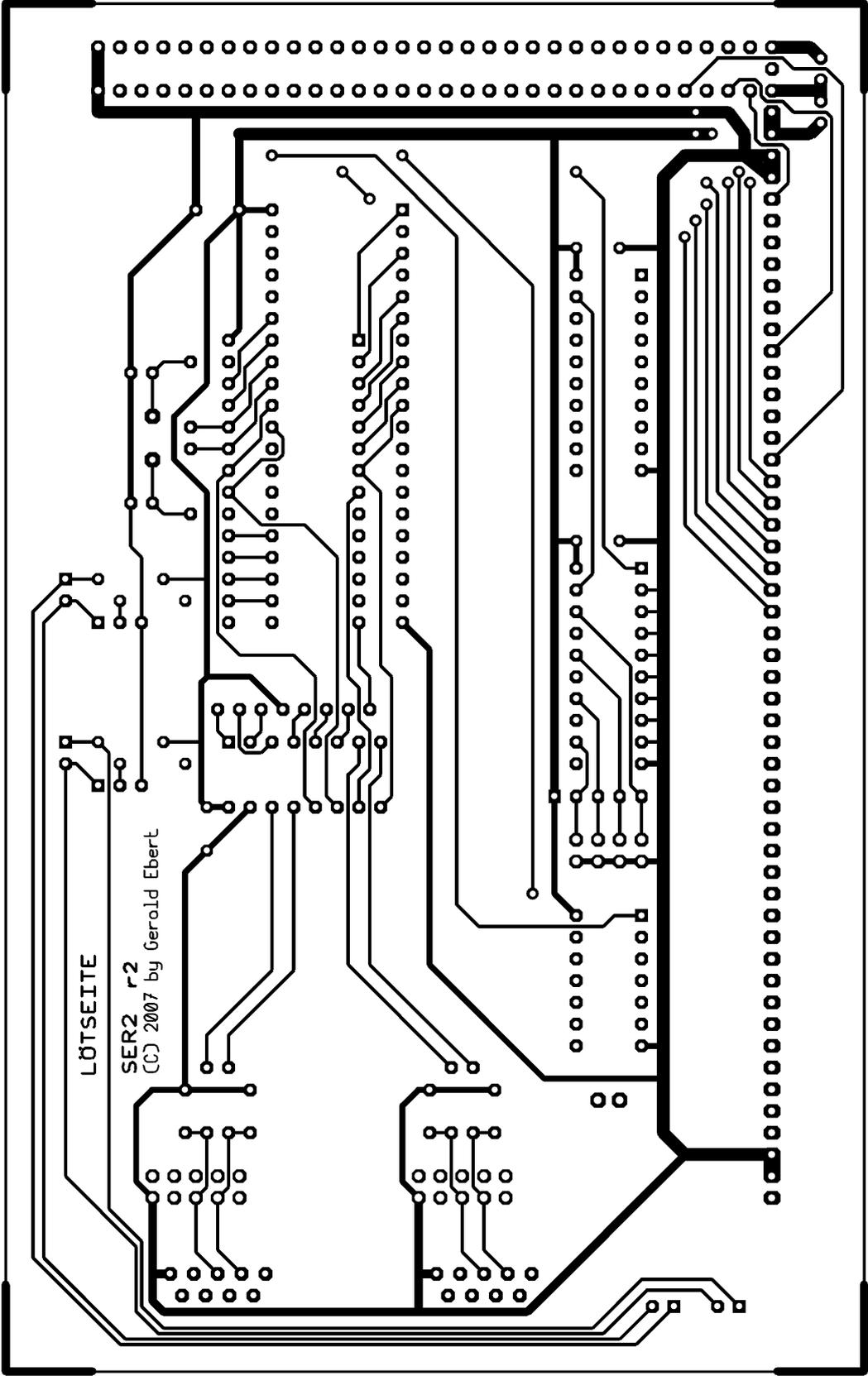
Layout Bestückungsseite mit Aufdruck



Layout Bestückungsseite



Layout Lötseite



Programmbeispiel zur Steuerung der Baugruppe

```

*****
***'                                     '***
***'.....SER2 Testprogramm.....'***
***'   DUART   XR88C681CP28/CP40       '***
***'                                     '***
***'   (P)+(C) 2007   Gerald Ebert     '***
***'                                     '***
*****

```

*>>>>>>

LABELS

```

cpu          equ     2                * M68000/10
s2base      equ     $ffffff90        * Basis I/O-Adresse ($90-$9F)

s2mra       equ     (s2base+0)*cpu   * Mode Register Channel A
s2sra       equ     (s2base+1)*cpu   * Status Register Channel A
s2csra      equ     s2sra            * Clock Select Register Channel A
s2misr      equ     (s2base+2)*cpu   * Masked Interrupt Status Register
s2cra       equ     s2misr          * Command Register Channel A
s2rhra      equ     (s2base+3)*cpu   * Rx Holding Register Channel A
s2thra      equ     s2rhra          * Tx Holding Register Channel A
s2ipcr      equ     (s2base+4)*cpu   * Input Port Change Register
s2acr       equ     s2ipcr          * Auxiliary Control Register
s2isr       equ     (s2base+5)*cpu   * Interrupt Status Register
s2imr       equ     s2isr           * Interrupt Mask Register
s2ctu       equ     (s2base+6)*cpu   * Counter/Timer Upper Byte Register
s2ctl       equ     (s2base+7)*cpu   * Counter/Timer Lower Byte Register
s2mr        equ     (s2base+8)*cpu   * Mode Register Channel B
s2sr        equ     (s2base+9)*cpu   * Status Register Channel B
s2csr       equ     s2sr           * Clock Select Register Channel B
s2res       equ     (s2base+10)*cpu  * Reserved
s2cr        equ     s2res           * Command Register Channel B
s2rhrb      equ     (s2base+11)*cpu  * Rx Holding Register Channel B
s2thrb      equ     s2rhrb         * Tx Holding Register Channel B
s2ivr       equ     (s2base+12)*cpu  * Interrupt Vector Register
s2ip        equ     (s2base+13)*cpu  * Input Port
s2opcr      equ     s2ip           * Output Port Configuration Register
s2sopb      equ     (s2base+14)*cpu  * Set Output Port Bits
s2copb      equ     (s2base+15)*cpu  * Clear Output Port Bits

```

*>>>>>>

Tabelle für Baudrate-Auswahl

```

s2bd9600    equ     $bb
s2bd19k2    equ     $cc
s2bd28k8    equ     $66
s2bd57k6    equ     $77
s2bd115k    equ     $88

```

*>>>>>>

START DES TESTPROGRAMMS

```

        move.b   #$03, s2sopb.w      * LEDs ausschalten

;' serielle Schnittstelle Channel A initialisieren
        move.b   #s2bd115k, d0      * 115 kbaud einstellen
        bsr     s2initA

;' Text auf serielle Schnittstelle Channel A ausgeben
loop1:  lea     text(pc), a0
        move.b   (a0)+, d0
        beq     next1
        bsr     s2sendA
        bra     loop1

;' Zeichen von serieller Schnittstelle holen und ausgeben
next1:  moveq    #!CSTS, d7          * Taste gedrueckt

```

```

        trap    #1
        tst     d0
        bne    next2          * Ja, Einlesen abbrechen
loop2:   bsr     s2readA
        bcs.s  next1          * kein Zeichen da, warten
        moveq  #!CO, d7       * Zeichen in d0 ausgeben
        trap   #1
        bra.s  next1

next2:   moveq  #!CI, d7       * Zeichen von Tastatur noch holen
        trap   #1
        rts                    * ENDE

text:    dc.b   'Hallo PC. Ich bin ein NDR-Klein-Computer.', 13, 10
        dc.b   0

*>>>>>>   GLOBALE RUECKSPRUNGROUTINEN

carryclr: andi  #$fe, ccr
        rts
carryset:  ori   #$01, ccr
ende:     rts

*>>>>>>   WARTESCHLEIFE FUER INIT
;'
wawa:     nop
        rts

*>>>>>>   INITIALISIERE CHANNEL A
;'
s2initA:  move.b  d0, s2csra.w   * Setze Baudrate
        bsr     wawa
        move.b  #$03, s2mra.w   * Even Parity, 8 bit Daten
        bsr     wawa
        move.b  #$07, s2mra.w   * Ein Stop bit
        bsr     wawa
        move.b  #$80, s2cra.w   * Setze Extent Bit fuer Rx Cha A
        bsr     wawa
        move.b  #$a0, s2cra.w   * Setze Extent Bit fuer Tx Cha A
        bsr     wawa
        move.b  #$15, s2cra.w   * Starte Channel A
        rts

*>>>>>>   SENDE EIN BYTE AUF CHANNEL A
;'
s2sendA:  movem.l  d1, -(a7)
        move.b  #$01, s2copb.w   * Channel A LED einschalten
        bsr     wawa
lbl1:     move.b  s2sra.w, d1     * Hole Channel A Status register
        btst.b  #2, d1          * Ausgaberegister leer
        beq.s   lbl1           * Nein -> weiter warten
        move.b  d0, s2thra.w     * Byte ins Ausgaberegister
        bsr     wawa
        move.b  #$01, s2sopb.w   * Channel A LED ausschalten
        movem.l  (a7)+, d1
        rts

*>>>>>>   HOLE EIN BYTE VON CHANNEL A
;'
s2readA:  move.b  s2sra.w, d0     * Hole Channel A Status Register
        btst.b  #0, d0          * Stehen Daten in Empfangsfifo
        beq     carryset        * Nein -> fifo leer -> C = 1
        move.b  s2rhra.w, d0    * Byte aus Empfangsfifo holen
        bra     carryclr        * Zeichen in d0 gültig -> C = 0

*>>>>>>   INITIALISIERE CHANNEL B

```

```

;'      d0 => Baudrate (siehe Tabelle)
s2initB:  move.b  d0, s2csrb.w      * Setze Baudrate
          bsr    wawa
          move.b  #$03, s2mrb.w    * Even Parity, 8 bit Daten
          bsr    wawa
          move.b  #$07, s2mrb.w    * Ein Stop bit
          bsr    wawa
          move.b  #$80, s2crb.w    * Setze Extent Bit fuer Rx Cha B
          bsr    wawa
          move.b  #$a0, s2crb.w    * Setze Extent Bit fuer Tx Cha B
          bsr    wawa
          move.b  #$15, s2crb.w    * Starte Channel B
          rts

*>>>>>>  SENDE EIN BYTE AUF CHANNEL B
;'      d0 => zu sendendes Byte

s2sendB:  movem.l  d1, -(a7)
          move.b  #$02, s2copb.w    * Channel B LED einschalten
          bsr    wawa

1b12:    move.b  s2srb.w, d1        * Hole Channel B status register
          btst.b  #2, d1           * Ausgaberegister leer
          beq.s   1b12            * Nein -> weiter warten
          move.b  d0, s2thrb.w     * Byte ins Ausgaberegister
          bsr    wawa
          move.b  #$02, s2sopb.w    * Channel B LED ausschalten
          movem.l  (a7)+, d1
          rts

*>>>>>>  HOLE EIN BYTE VON CHANNEL B
;'      d0 <= empfangenes Byte, wenn C = 0

s2readB:  move.b  s2srb.w, d0      * Hole Channel B Status Register
          btst.b  #0, d0          * Stehen Daten in Empfangsfifo
          beq     carryset        * Nein -> fifo leer -> C = 1
          move.b  s2rhrb.w, d0    * Byte aus Empfangsfifo holen
          bra     carryclr        * Zeichen in d0 gültig -> C = 0

          end

```