

Technische Referenz



für
TNC3, MS-DOS, Windows NT/Win95,
Atari und Linux

<http://www.swiss-artg.ch>

Software: Jimy Scherer, DL1GJI

E-Mail: dl1gji@qsl.net

Dokumentation: Manfred Widmer, DL2GWA

E-Mail: dl2gwa@qsl.net

©-Hinweis: Diese Dokumentation darf für nichtkommerzielle Zwecke
(auch auszugsweise) mit Quellenangabe weiterverbreitet werden

Inhaltsverzeichnis

| | |
|-----------------------------------------------------------------------------|-----------|
| Mehrere TNC3S über High-Speed-Bus mit 1,6MB-Arbiter..... | 3 |
| Einleitung..... | 3 |
| Aufgabe und Funktionsweise des Arbiters | 3 |
| Aufbau und Bestückung Arbiter | 3 |
| Arbiter für 4 TNCs (6 HF-Ports)..... | 4 |
| Slave-TNCs (Arbiter-Version) | 4 |
| Master-TNC (Arbiter-Version) | 4 |
| Fertigstellung der HS-Bus-Umrüstung | 4 |
| Karte Master-Serial-Link mit RS232-Treiber (optional)..... | 5 |
| High-Speed-Bus-Arbiter der 2. Generation..... | 8 |
| HS-Bus-Treiber-Platine | 13 |
| Der High-Speed-Bus und NetNode | 14 |
| HighSpeedBus-Hardware | 15 |
| Das KISS-Busprotokoll..... | 15 |
| HighSpeedBus-Kiss-Software | 18 |
| HSKISS.APL | 18 |
| KTST.APL | 18 |
| STRESS.APL..... | 19 |
| Literatur | 19 |
| DIP-Schalterstellung am (X)NET-Master-TNC..... | 20 |
| DIP-Schalterstellung Terminal-Baudrate TNC3 (Token-Ring-Master)..... | 20 |

Mehrere TNC3S über High-Speed-Bus mit 1,6MB-Arbiter

Einleitung

Der High-Speed-Bus-Arbiter dient zur Kopplung mehrerer TNC3S als Packet-Radio-Digipeater über den seriellen, schnellen HS-Bus. Es können somit mehrere HF-Ports realisiert werden. Eine Multiport-Arbiter-Version (ohne GAL-ICs) wurde von Walter Zimmer, DL1HAZ, entwickelt und veröffentlicht.

Aufgabe und Funktionsweise des Arbiters

Damit der Datenaustausch auf dem Bus kollisionsfrei ablaufen kann, darf immer nur ein (T)NC zur Zeit auf den Bus zugreifen. Der Bus-Arbiter regelt diesen Zugriff auf den High-Speed-Bus. Die Bus-Geschwindigkeit liegt bei ca. 1,6 Mbit/sec.

Der Bus-Arbiter arbeitet nach dem „Round-Robin“-Verfahren . Das heisst, dass alle Eingänge der Reihe nach abgefragt werden und es wird dabei kein (T)NC bevorzugt oder benachteiligt.

Ein (T)NC der Daten senden möchte, teilt dies dem Arbiter durch das Setzen des RTS-Signals mit. Der Bus-Arbiter gibt dann diesem (T)NC durch das CTS-Signal die Freigabe zum Senden.

Die Eingänge RTS1 bis RTS 4 werden zyklisch abgefragt. Bei einem Takt von 1 MHz wird jeder Eingang alle 4 µsec ($1 \mu\text{sec} * 4$) abgefragt. Ist einer der Eingänge aktiv (RTS = low) so wird der Takt für den Zähler gestoppt. Mit dem nächsten Bus-Takt wird dann auch das entsprechende CTS-Signal aktiv (low). Erst jetzt wird der (T)NC seine Daten auf den Bus senden.

Ist der (T)NC damit fertig, nimmt er das RTS-Signal wieder zurück. Damit werden die Takte für den Zähler wieder freigegeben und die weiteren RTS-Signale getestet. Erst mit dem nächsten Bus-Takt wird das CTS-Signal für den (T)NC zurückgenommen und evt. das CTS-Signal für einen anderen (T)NC erzeugt.

Die Verzögerung des CTS-Signals ist notwendig, weil der MC68302 das Signal schon mit dem Beginn des letzten Bits zurücknimmt, das CTS-Signal aber bis zum Ende des letzten Bits aktiv bleiben muß.

Aufbau und Bestückung Arbiter

Für diesen Arbiter ist eine Platine verfügbar. Diese ist auf einer einseitigen Epoxyd-Platine ausgeführt. Die Durchkontaktierungen werden über Drahtbrücken realisiert. Der gesamte Aufbau gestaltet sich sehr einfach.

Arbitrer für 7 TNCs (14 HF-Ports)

Slave-TNCs (Arbitrer-Version)

An den Bus-Arbitrer werden über die Schnittstellen-Kabel (siehe Stecker-Verbindung) bis zu 3 Slave-TNCs angeschlossen. Zuvor sind die Slave-TNCs noch umzurüsten. Hierzu werden die Gehäuse der TNCs geöffnet und die Modem-Platinen entfernt. Der Schnittstellen-IC MAX232 ist zu entfernen. Hierfür werden optional die ICs U4 und U5 (siehe TNC3-Handbuch von Symek) bestückt = 74HC125 und 74HC126 (unbedingt HC-Typen verwenden!). Die Lage dieser ICs sind auf der Platine nahe der RS232-Schnittstelle zu finden. Es empfiehlt sich, diese Steckplätze zu sockeln. Danach kann das Gehäuse nach Bestückung der Modem-Platinen wieder geschlossen werden. Die Slave-TNCs benötigen die normale TNC3-Software (also keine spezielle High-Speed-Bus-Soft, da dies bereits in der zum TNC3 mitgelieferten Software integriert ist). Die Dip-Schalterstellung ist im TNC3-Handbuch erläutert (DIP 7 = KISS-Mode für High-Speed-Bus).

Wichtig: An die seriellen Schnittstellen der Slave-TNCs darf keine V.24 mehr angeschlossen werden !! Die Bus-Adressen der Slave-TNCs werden über die DIP-Schalter 1-3 an der Rückseite eingestellt. Die Wertigkeit ist binär, wobei zu beachten ist, daß DIP 1 - 3 getauscht sind.

Master-TNC (Arbitrer-Version)

Nun noch den Master-TNC umrüsten. Man entfernt hierzu die 1k2-Modem-Platine an der SCC1 und ersetzt sie durch die High-Speed-Serial-Link-Karte. Sie hat gleiche Ausmaße, wie eine 9k6Bd-Modem-Karte. Das Schnittstellenkabel ist an die SCC1 anzuschließen.

Eine Änderung an der seriellen V.24-Schnittstelle ist nicht erforderlich; die RS232-Schnittstelle des Master-TNC bleibt voll erhalten für den Anschluß eines Terminals oder einer Box etc.

Der Master wird über die 5pol. DIN-Buchse an den Arbitrer angeschlossen. Die DIP-Schalter-Stellung ist aus der Beschreibung (X)Net im Anhang zu entnehmen. Die DIP-Schalter-Stellung 6 konfiguriert HS-Bus. Es wird vorausgesetzt, daß Knotensoftware (X)Net bereits im Master-TNC vorhanden ist.

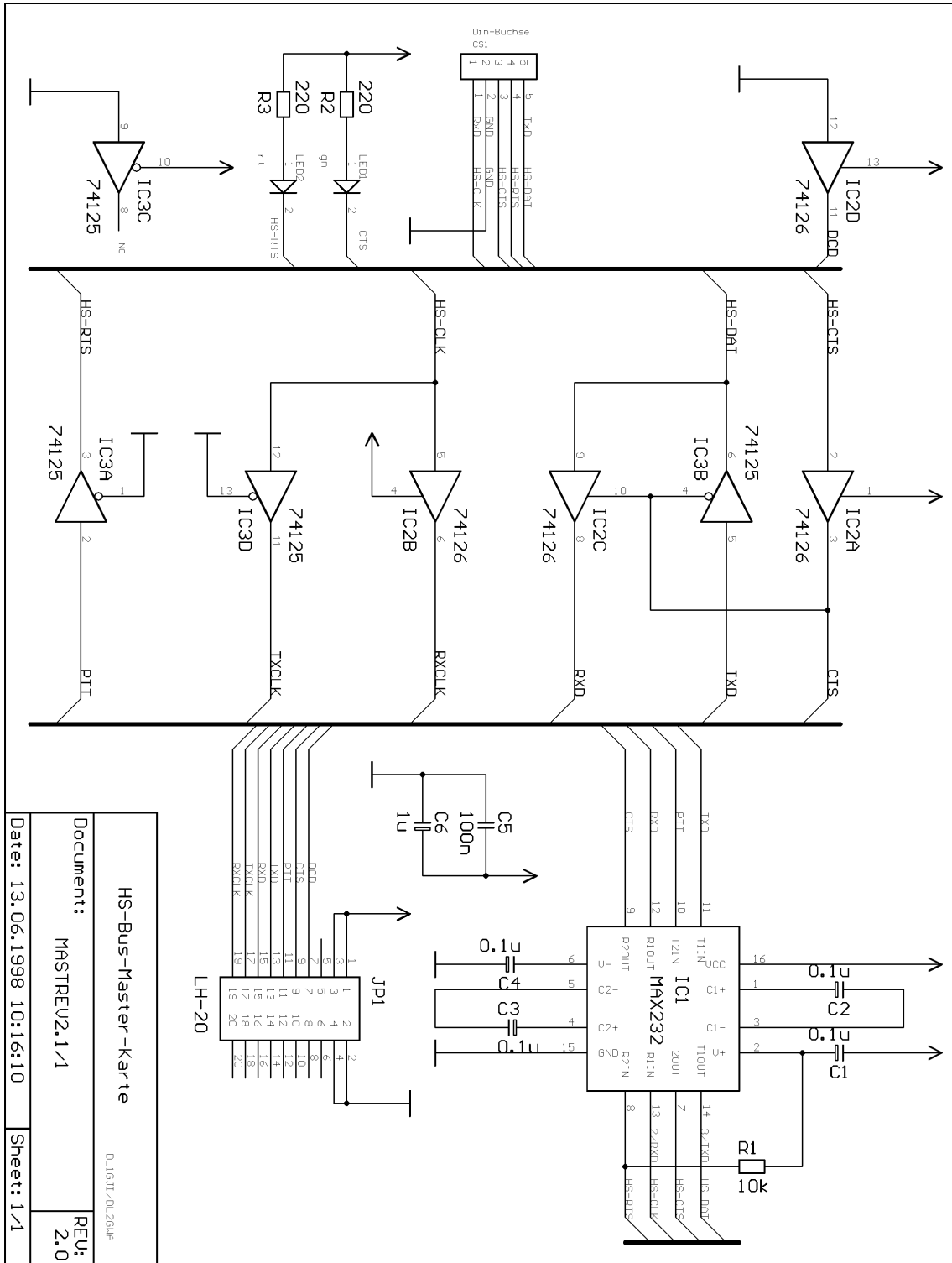
Fertigstellung der HS-Bus-Umrüstung

Damit wäre die Umrüstung abgeschlossen. Der High-Speed-Digi kann in Betrieb genommen werden. Die weiteren Einstellungen des Knotens sind der (X)Net-Dokumentation zu entnehmen.

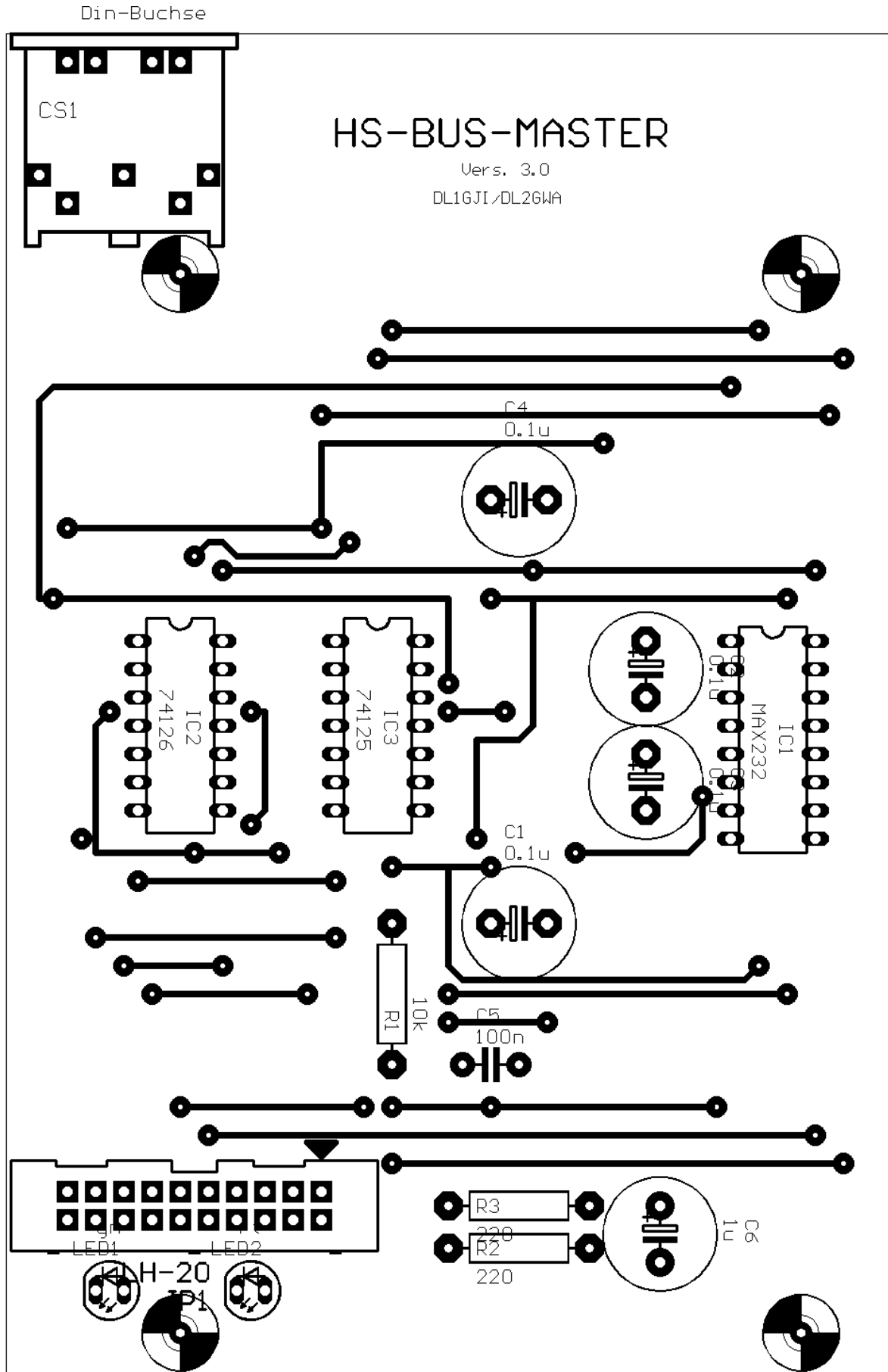
Karte Master-Serial-Link mit RS232-Treiber (optional)

Als Option kann die nachfolgende Master-Serial-Link-Karte als RS232-Schnittstelle im Master-TNC dienen (zum Beispiel über die freie SCC2). Damit steht eine weitere serielle Schnittstelle zur Verfügung, die über das Attach-Kommando der (X)NET-Knotensoftware eingestellt werden kann.

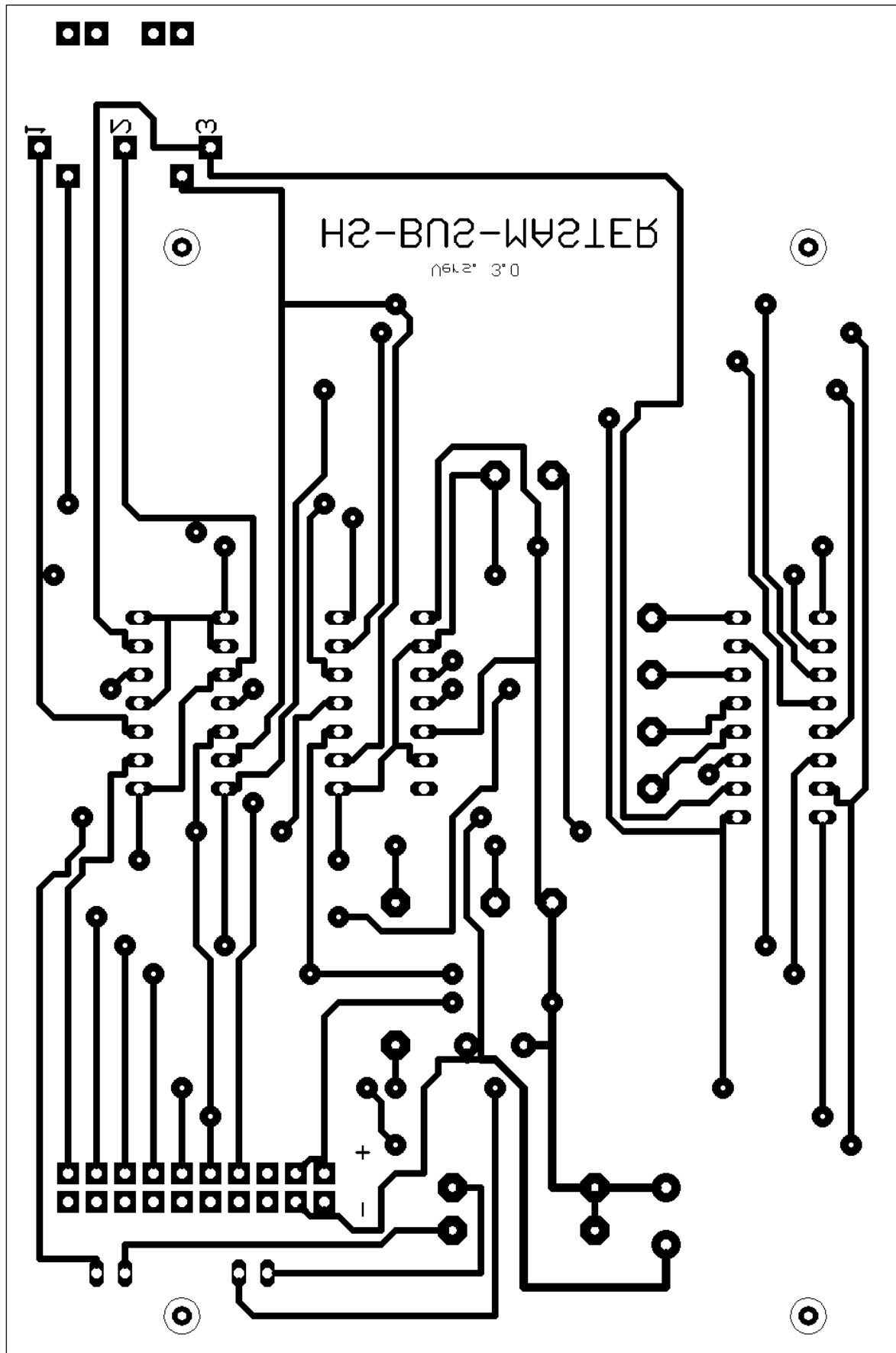
Somit besteht eine Möglichkeit an den Master-TNC neben einem PC-Terminal z.B. eine Box über Draht-Link anzubinden.



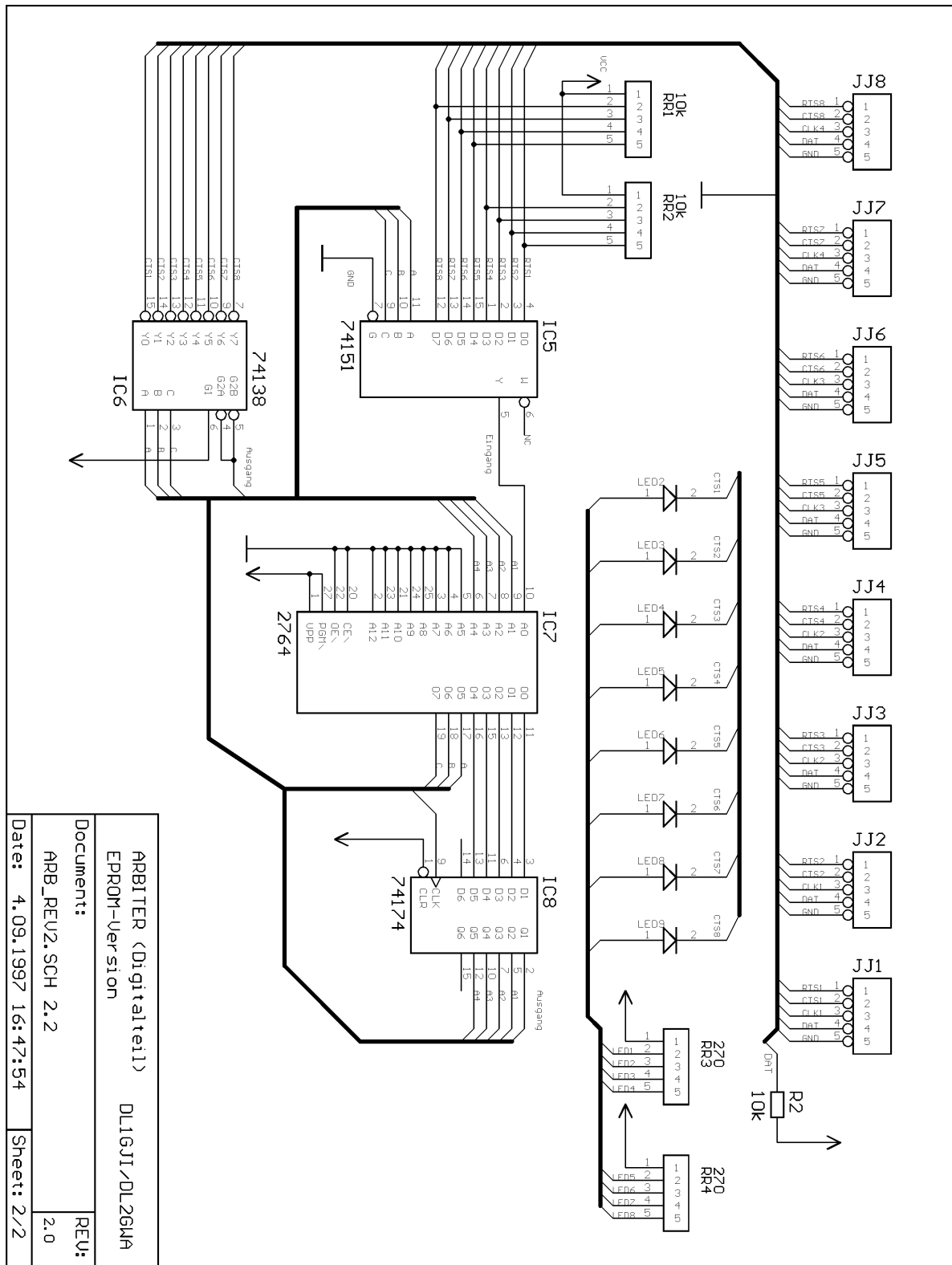
| | | |
|---------------------|---------------------|---------------|
| HS-Bus-Master-Karte | | DL16J1/DL26HA |
| Document: | MASTREV2.1/1 | REV: |
| | | 2.0 |
| Date: | 13.06.1998 10:16:10 | Sheet: |
| | | 1/1 |



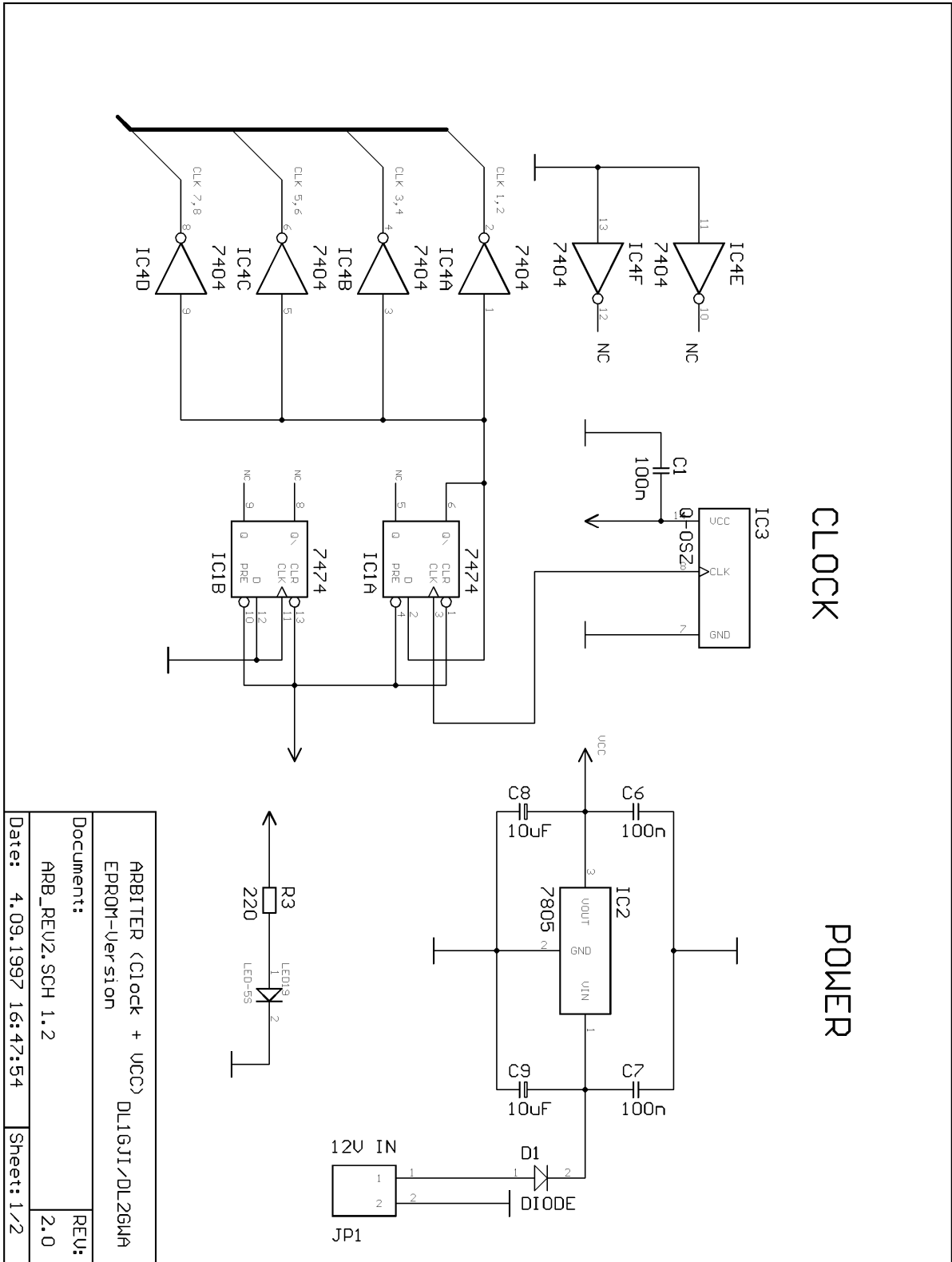
Seriell-Master-Platine Rev. 2.0 (Bestückungsplan)



High-Speed-Bus-Arbitrer der 2. Generation

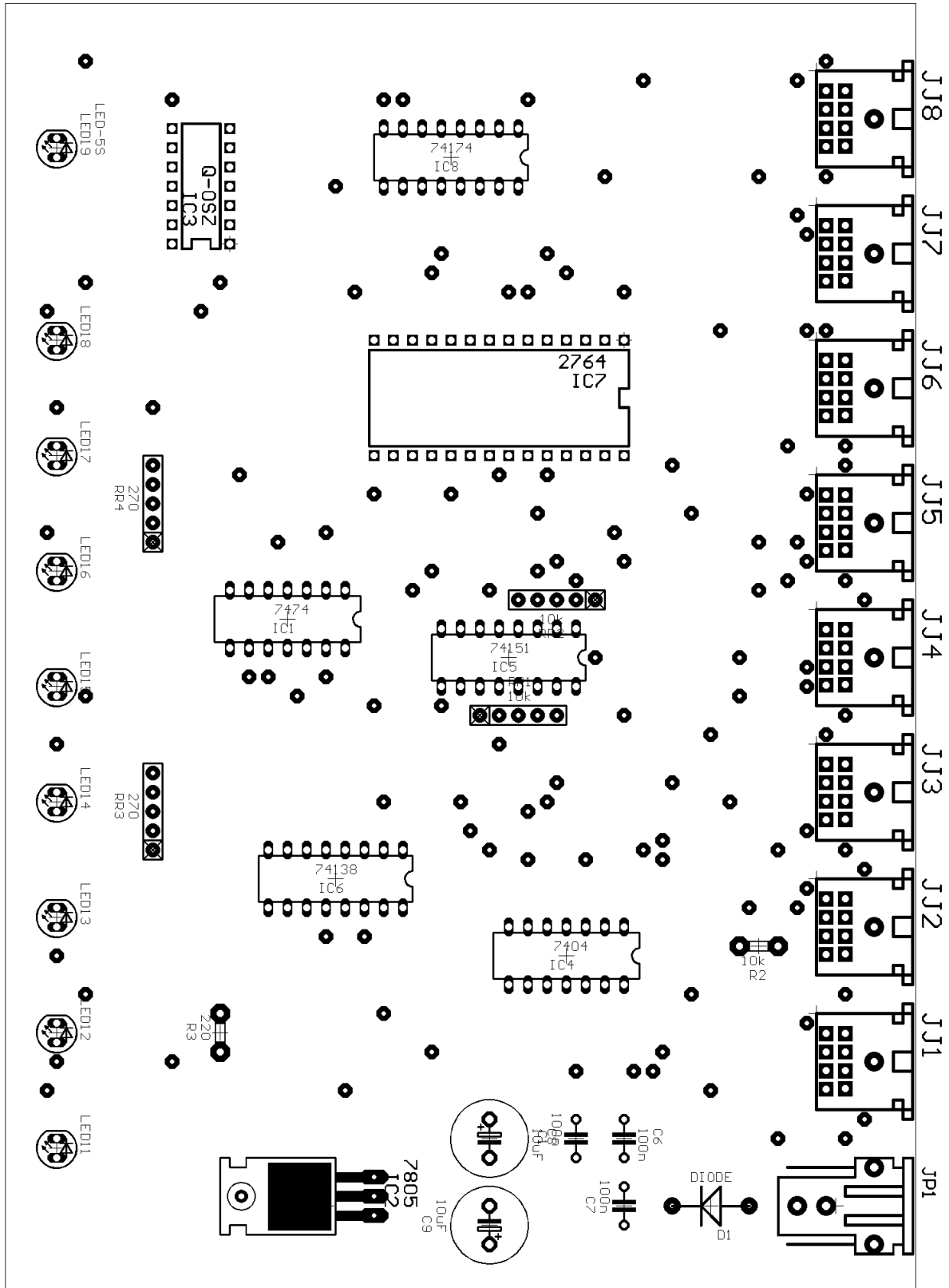


Schaltbild Arbitrer 2. Generation mit EPROM - Rev. 2 (Digitalteil)

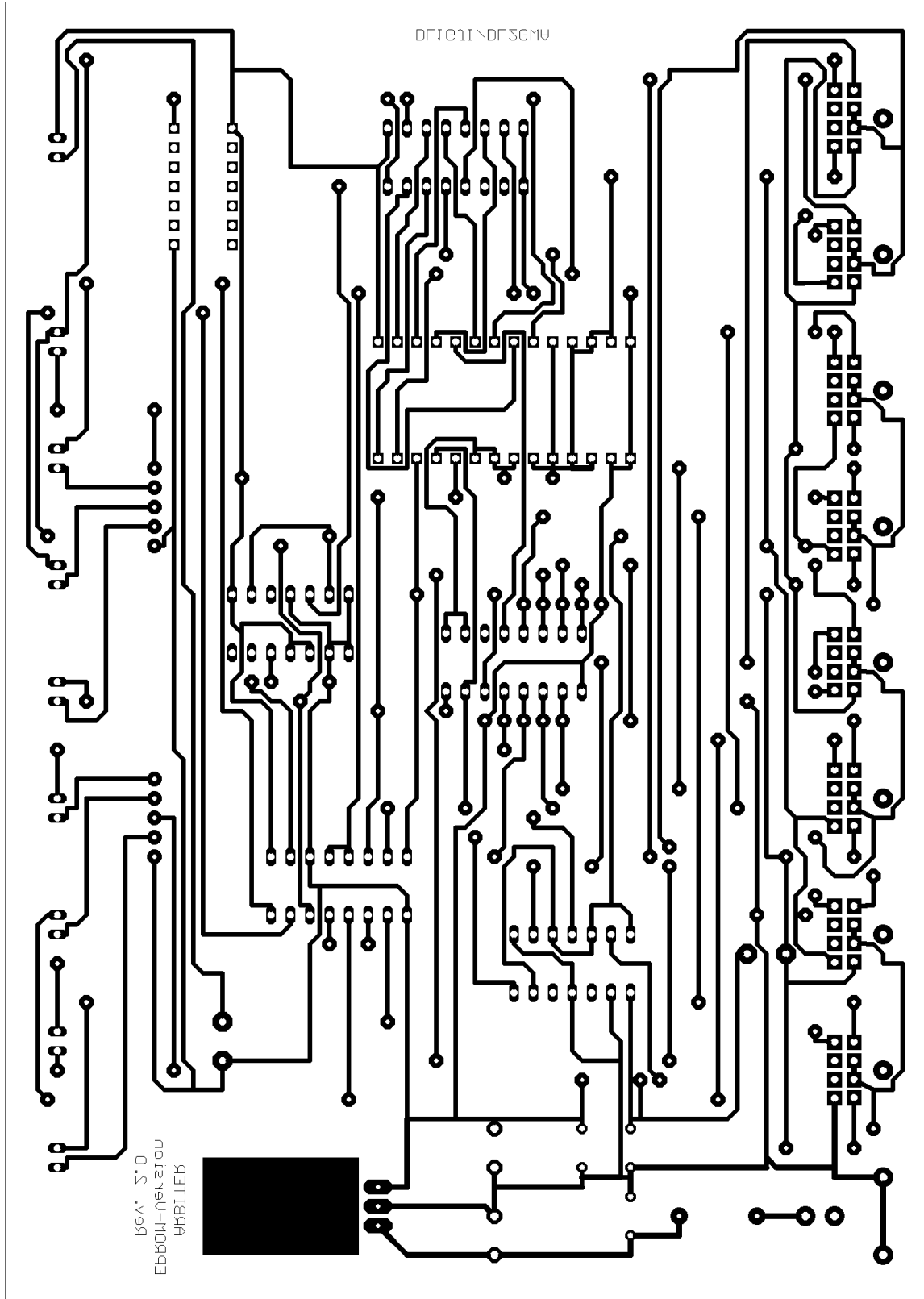


Schaltbild Arbitrer 2.Generation mit EPROM - Rev. 2 (Clock/Power)

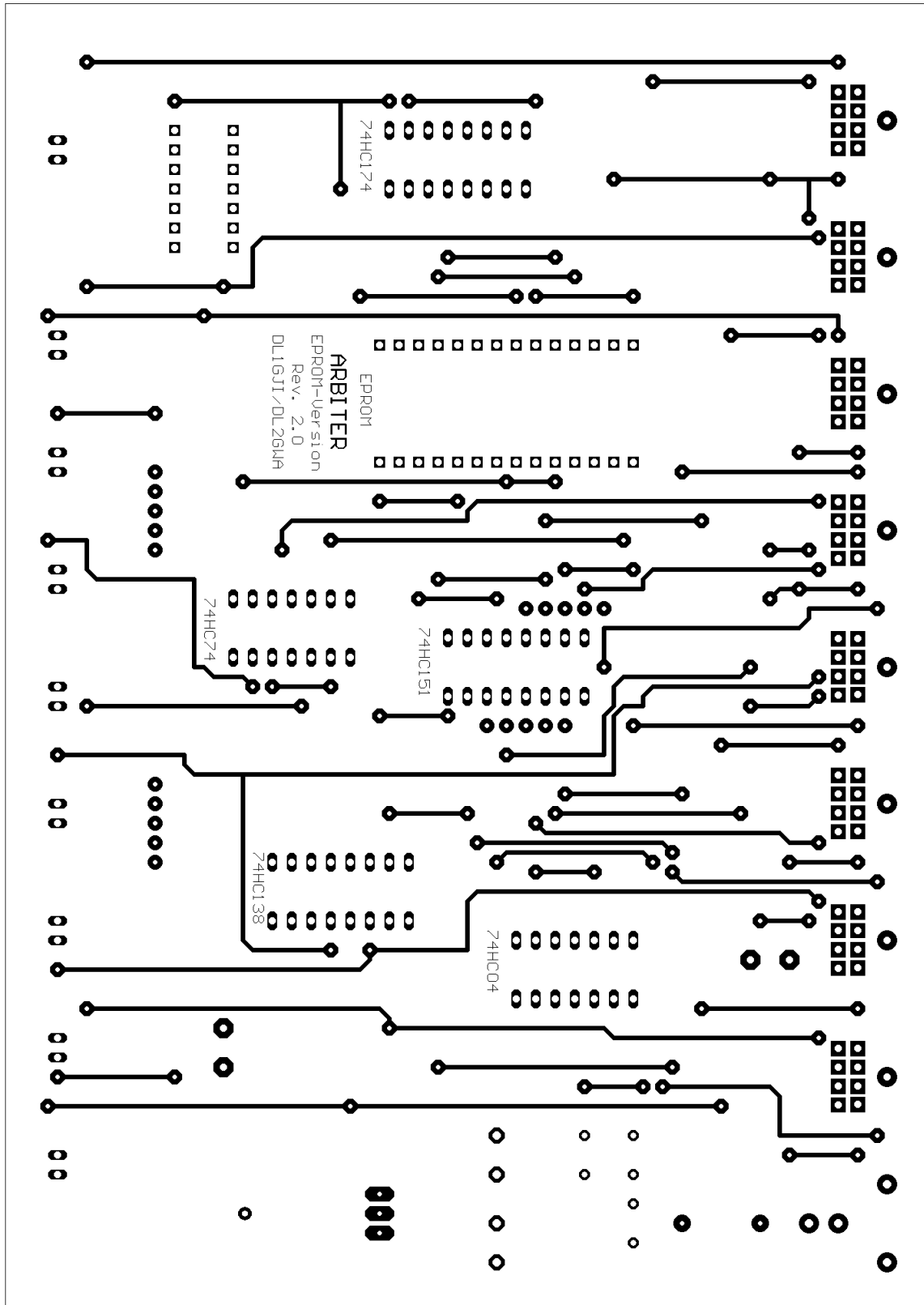
| | |
|-------------------------------------------------------|-------------|
| ARBITTER (Clock + UCC) DL16J1/DL26M4 EPROM-Version | |
| Document: ARB_REV2.SCH 1.2 | REV: 2.0 |
| Date: 4.09.1997 16:47:54 | Sheet: 1/2 |



Arbitr 2. Generation mit EPROM (Bestückungsplan)



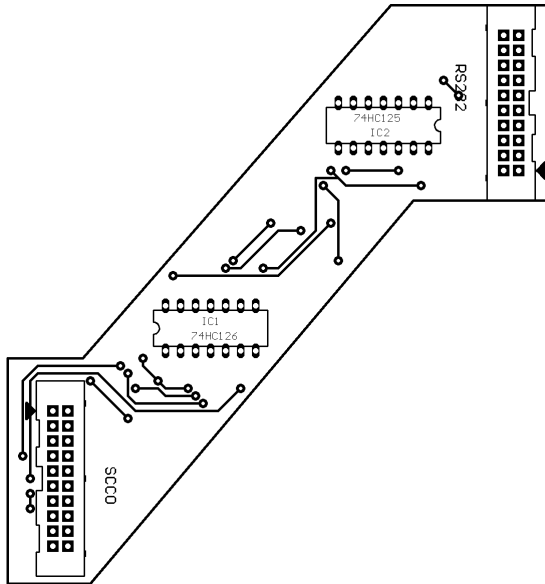
Arbitrer 2.Generation mit EPROM (Lötseite)



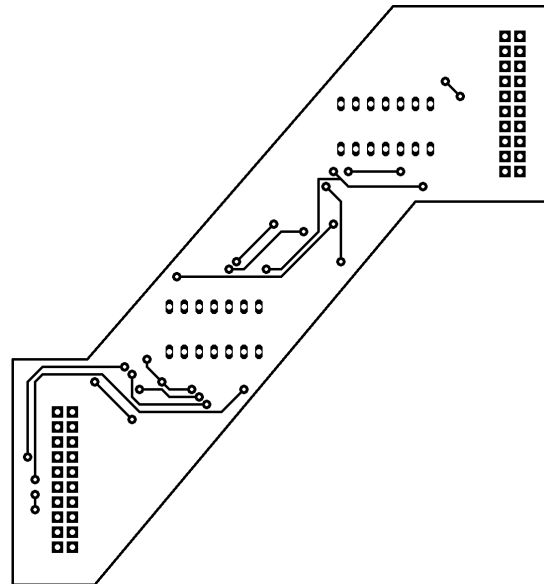
Arbiter 2.Generation mit EPROM (Bestückungsseite mit Durchkontaktierungen)

HS-Bus-Treiber-Platine für Slave-TNC3S - K (SMD) zur Anbindung an HS-Bus-Arbiter

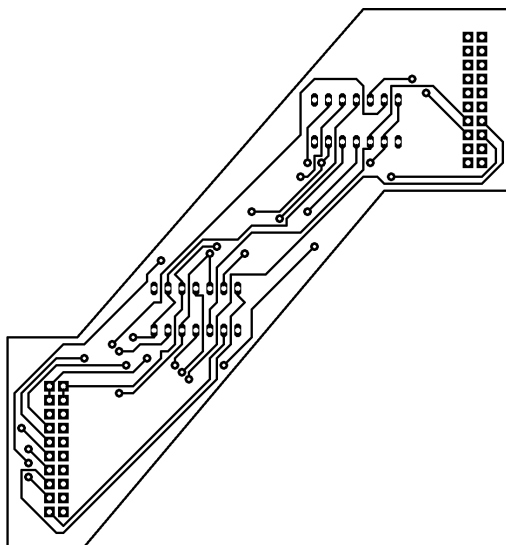
Das TNC3S der Bauserie K in SMD-Ausführung verfügt **nicht** mehr über die optionalen IC-Steckplätze für die HS-Treiber-Ics. Deshalb wurde eine spezielle Platine entwickelt, die in den TNC3S eingesteckt werden kann und SCC3, der seriellen Schnittstelle (RS232), mit der 25poligen SubD-Buchsenleiste verbindet.



Bestückungsseite



Durchkontaktierungen auf der Oberseite



Lötseite

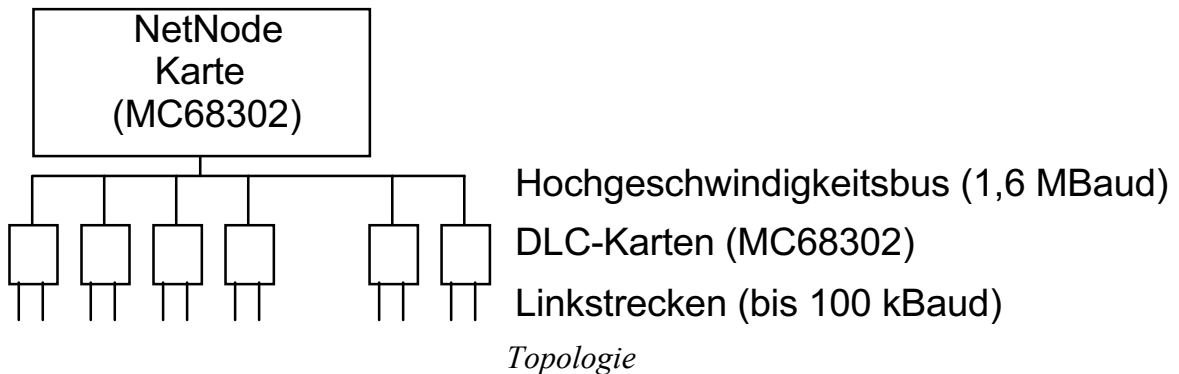
Für die Draht-Verbindung von der RS232-Schnittstelle des Slave-TNCs zum Arbiter, wird ein spezielles Datenkabel angefertigt.

TNC3 - HighSpeedBus

Der HighSpeedBus ist eine einfache und schnelle Vernetzungsoption für den TNC3. Besonders interessant ist diese Vernetzungsmöglichkeit für Digipeater. Hier wird am Beispiel eines NetNode-Digi die Realisierung einer HighSpeedBus-Kopplung erläutert.

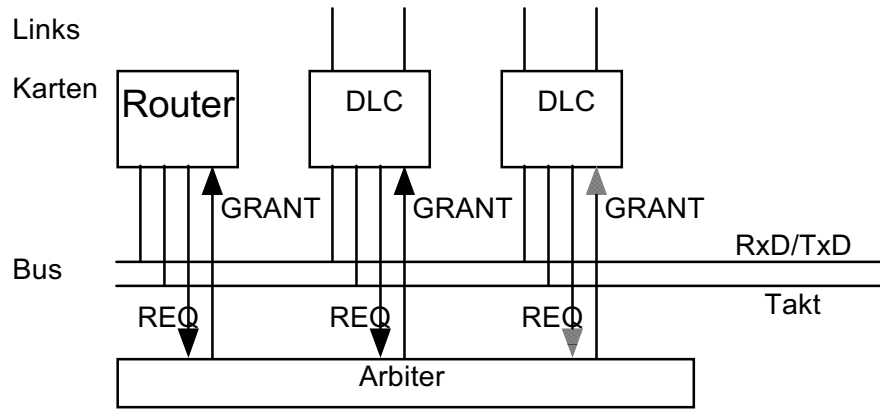
Der High-Speed-Bus und NetNode

Ein NetNode-Knotenrechner besteht aus einem NetNode-Master-Rechner (TNC3 mit 256K RAM), und mehreren Dual-Link-Controllern (DLCs). Der NetNode-Rechner ist mit genügend Speicher und einer Uhr (RTC) ausgestattet. Die DLCs sind minimal ausgestattete TNC3s, sie sind im Wesentlichen für die Aussendung der AX-25-Rahmen (Medium-Access-Control-Ebene MAC) zuständig. Alle Karten werden über den Hochgeschwindigkeitsbus (HS-Bus) vernetzt.



Die maximale Übertragungsrate auf dem HS-Bus liegt bei 1,6 MBit/s (bedingt durch MC68302). Dabei dürfen die Linkstrecken jedoch nicht mit mehr als 100 kBaud betrieben werden. Sind höhere Funkbaudraten gewünscht, kann der Bus mit 512 kBit/s betrieben werden, die maximalen Funkbaudraten sind dann jeweils ebenfalls 512 kBaud pro Link.

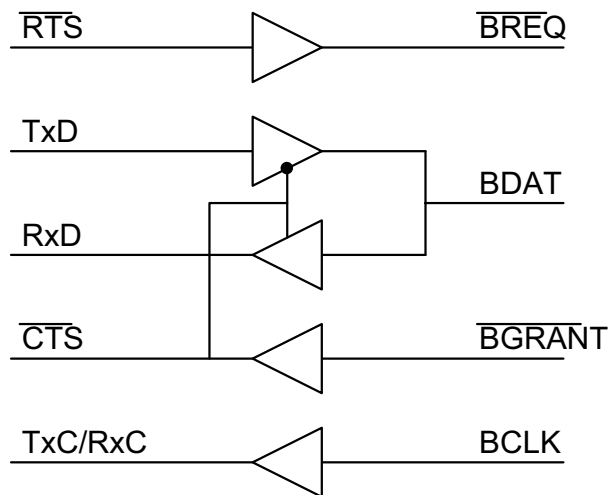
HighSpeedBus-Hardware



DLC = Dual Link Control

Verdrahtung des Busses

Der HighSpeedBus besteht aus einfacher Elektronik. Auf dem TNC3 sieht die Verschaltung des Busses mit dem entsprechenden SCC-Port so aus:



TNC3

HighSpeedBus

HighSpeedBus-Anschaltung

Ein Arbitrer ist zum Betrieb des HighSpeedBusses notwendig. Diese Schaltung regelt die Buszugriffe und verhindert Kollisionen. Die bei NetNode bisher größtenteils softwaremäßig realisierte Steuerung des Buszugriffs (Token-Polling-Verfahren) ist damit hardwaremäßig gelöst.

Das KISS-Busprotokoll

Das primitive Busprotokoll ist dem TNC-KISS-Protokoll nachempfunden. Die KISS-Daten sind jedoch in einen HDLC-Rahmen gepackt. Dadurch können alle Zeichendopplungen (FESC, TFEND,...) aufgrund der Transparenz der HDLC-Rahmen entfallen. Übertragungsfehler sind wegen kurzer Leitungen und der Kollisionsfreiheit selten. Falls Übertragungsfehler dennoch auftreten werden sie mit Hilfe der automatisch generierten Prüfsumme der HDLC-Rahmen (CRC) zu 99,99% erkannt.

Die Adressen sind länger kodiert als bei KISS. Die Adreßerkennung übernimmt der MC68302-RISC-Controller vollständig.

Busprotokoll basierend auf "KISS-MODE"



KISS-Rahmen

Adreßfelder sind notwendig um die einzelnen DLC-Karten anzusprechen. Die DLC-Software (HSKISS.APL) ist in jedem TNC3 bereits standardmäßig enthalten. Gestartet wird diese Software durch die DIP-Schalterstellung 7. Da keine RS232-Baudrate eingestellt werden muß, können die drei Baudraten-Bits am DIP-Schalter zur Einstellung der DLC-Nummer verwendet werden. Achtung die Bits sind etwas "verdreht":

| DIP-Schalter: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | DLC Nummer | NetNode-Ports Modem1 Modem2 | |
|---------------|---|---|---|---|---|---|---|---|---------------|--------------------------------|----------------|
| | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 ¹ |
| | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 2 | 2 | 3 |
| | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 4 | 4 | 5 |
| | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 6 | 6 | 7 |
| | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 8 | 8 | 9 |
| ... | | | | | | | | | | | |
| | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 14 | 14 | 15 |

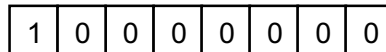
Durch die Verdrahtung der DIP-Schalter ergibt sich eine Invertierung:

- 1 = Open = OFF = DIP-Schalter **oben**
- 0 = Closed = ON = DIP-Schalter **unten**

Da NetNode nur max. 16 Ports unterstützt können hier auch nur Portnummern von 0-15 eingestellt werden.

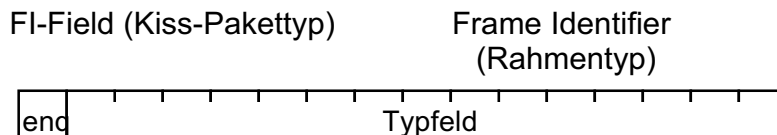
Die NetNode-Karte hat eine fest vorgegebene HighSpeedBus-Adresse:

NetNode-Karte



Adresse der NetNode-Karte

Das FI-Feld ist zwei Bytes groß und hat folgendes Bitmuster



FI-Feld

¹ Bei der NET_XS-Version werden diese Port-Nummern schon für die Ports am Master-NC belegt.

Die Kodierung des Typfeldes entspricht KISS

| Typfeld | Name | Beschreibung | Größe [Bytes] |
|---------|---------------------------|----------------------------------------------------------------|---------------|
| 0x0 | Daten | Daten folgen | 0 bis 328 |
| 0x1 | TxDelay | TxDelay [ms] | 2 |
| 0x2 | Persistenz | 0 bis 255 | 1 |
| 0x3 | Slottime | Slottime [ms] | 2 |
| 0x4 | TxTail | TxTail [ms] | 2 |
| 0x5 | Vollduplex | 0=Aus, 1=Ein, n=PTT Hold Time [s] | 1 |
| 0x6 | DAMA | 0=Aus, 1= Ein | 1 |
| 0xD | Reset | DLC-Reset auslösen | 1 |
| 0xE | Frames gesendet | Meldung im DAMA- Mode, daß alle Daten ausgesendet wurden | 1 |
| 0x8001 | TxDelay? | | 2 |
| 0x8002 | Persistenz? | | 1 |
| 0x8003 | Slottime? | | 2 |
| 0x8004 | TxTail? | | 2 |
| 0x8005 | Vollduplex? | | 1 |
| 0x8006 | DAMA? | | 1 |
| 0x8007 | Baudrate? | Modem-Baudrate [100 Hz] | 2 |
| 0x8100 | DLC Programm- version? | Zeichenkette | n |

Unterschiede zu KISS:

Alle Zeitparameter sind in Millisekunden angegeben, denn bei zukünftigen TRXen ist mit kleineren TxDelays und Slottimes zu rechnen. Die meisten numerischen Werte werden in zwei Bytes kodiert, 68000-Format. "Escape"-Sequenzen bzw. Zeichendopplung ("Character Stuffing") in den Daten entfallen.

Eine weitere KISS-Ergänzung: Mit Hilfe des Bits enq lassen sich auch Werte abfragen (enq = 1). Die abgefragten Werte werden im gleichen Format an die abfragende Station zurückgesendet.

Die Vorteile gegenüber dem TheNet-Token-Ring liegen auf der Hand:

- 20 bis 40 mal höhere Baudraten auf dem lokalen Bus, wesentlich höherer Durchsatz, höhere Dynamik
- wesentlich geringere Antwortzeiten da keine Zugriffskontrolle mittels Tokens, Verzögerungszeiten im Mikrosekundenbereich
- Kollisionsfreiheit garantiert durch Arbitrer; die Bitfehlerrate auf dem Bus ist gering
- Fehlererkennung durch Verpackung aller KISS-Rahmen in HDLC-Rahmen mit CRC-Prüfsumme
- Keine RS232-Pegelwandler notwendig
- Prüfsummenberechnung und Prüfung durch den 68302-RISC-Communications-Controller, keine CPU-Belastung
- Datenaustausch über den Bus belastet die einzelnen DLCs nicht (optimale Unterstützung durch die Adreßerkennung des MC68302 RISC-Communications-Controller).

- Empfang und Senden von KISS-Rahmen auf dem Bus läuft vollständig im Hintergrund; es sind keine Interrupts notwendig. (Empfangen und Senden steuert der RISC-Communications-Controller zusammen mit dem Arbiter)

Die obigen Punkte sprechen für sich. Es sei trotzdem erwähnt, daß es nicht möglich ist mit Hilfe eines parallelen Busses die obigen Leistungsmerkmale an Durchsatz und Antwortzeiten bei gleichzeitig minimaler Prozessorbelastung zu erreichen, es sei den mit Hilfe (teurer) intelligenter Spezialhardware. (1,6 MBit/s entspricht 200 KBytes/s => für die programmgesteuerte Übertragung eines Bytes stünden nur 5 µs zur Verfügung (dies ist mit programmgesteuerter paralleler Übertragung nicht zu erreichen, da auch noch das "Handshaking" per Programm zu regeln ist), der Prozessor wäre während der Busdatenübertragung voll ausgelastet, es wäre demnach hier keine Parallelarbeit des Prozessors möglich).

HighSpeedBus-Kiss-Software

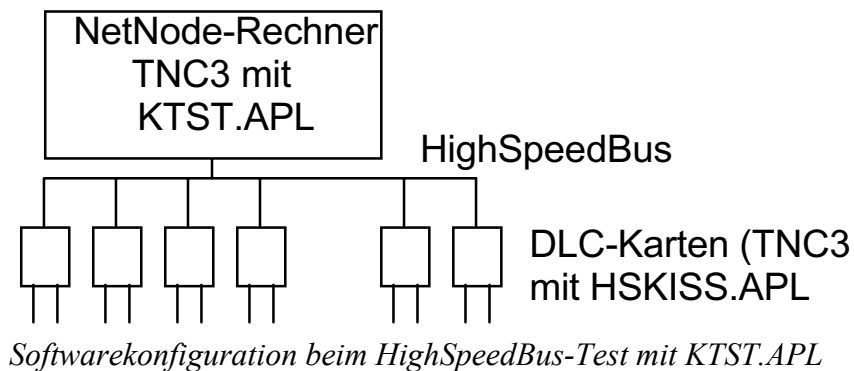
Programme, die den HighSpeedBus benutzen sind:

HSKISS.APL

HighSpeedBus-DLC-Software. Dieses Programm ist in jedem TNC3 schon enthalten und ermöglicht es jeden TNC3 am HighSpeedBus als DLC (Dual Link Controller) zu verwenden. Die HighSpeedBus-Portnummern werden direkt an den DIP-Schaltern für die RS232-Baudrate eingestellt.

KTST.APL

KTST.APL ist ein HighSpeedBus-Testmonitor. Das Programm wird auf der Master-Karte gestartet und zeigt alle angeschlossenen HighSpeedBus-DLCs und deren Konfiguration an. Es eignet sich sehr gut um die Funktionsfähigkeit des Busses zu prüfen und die eingestellten Kartenadressen der DLCs anzuzeigen. Das Programm macht sogar die Modem-Baudraten der einzelnen DLC-Ports sichtbar.



```
r:>ktst
=====
Kiss - HighSpeedBus - Testmonitor

      (b)roadcast: Busteilnehmer anzeigen
      (k)iss Konfiguration anzeigen
      (q)uit

=====
```

KTST.APL ist auf der DIGI-Betriebsdiskette und muß von dort geladen werden.

STRESS.APL

STRESS.APL ist ebenfalls ein Testprogramm für den HighSpeedBus. Beim Aufruf wird die Nummer des DLCs angegeben der getestet werden soll. Auch im Dauerbetrieb dürfen keine Fehler auftreten.

```
r:>stress 1
*** HighSpeedBus - Stress-Test ***

Taste...

Statistik DLC [01]:
  Fehler (gesamt) :          0
  Pakete (gesamt) :          0
  Bytes (gesamt)  :          0
  Bitfehlerrate  :          0 E-8
  Pakete/s       :          0
  Bit/s (Netto)  :          0

  Discarded Frames:         0
  Abort Sequences :         0
  CRC-Errors      :         0
  Retransmissions:         0
  Non matched Adr.:         0
```

Die kompletten Quelltexte inklusive HighSpeedBus-Implementierung und der NetNode-Sourcen sind übrigens im TNC3-Entwicklungssystem mit enthalten.

Literatur

- [ADAC92] Roland Alber, Joachim Scherer, Hochgeschwindigkeits (T)NC, ADACOM 3/1992
- [ADAC93] Roland Alber, Joachim Scherer, Das Hochgeschwindigkeits (T)NC-Konzept in der Praxis, ADACOM 5/1993
- [MOTO90] Motorola, MC68302 Integrated Multi-Protocol Processor Users Manual, Rev 1, Motorola 1990
- [TANE87] Andrew S. Tanenbaum, Operating Systems Design and Implementation, Prentice-Hall International 1987
- [TANE89] Andrew S. Tanenbaum, Computer Networks, Second Edition, Prentice Hall International 1989

DIP-Schalterstellung am (X)NET-Master-TNC

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Autostart-Software |
|---|---|---|---|---|---|---|---|-----------------------------------------------|
| x | x | x | ↓ | ↓ | ↓ | ↓ | ↓ | DIP 0: MC68K Betriebs- und |
| x | x | x | ↓ | ↓ | ↓ | ↓ | ↑ | DIP 1: TNC Konfiguration |
| x | x | x | ↓ | ↓ | ↓ | ↑ | ↓ | DIP 2: 3-Port Single Board |
| x | x | x | ↓ | ↓ | ↓ | ↑ | ↑ | DIP 3: NOS-Frontend |
| x | x | x | ↓ | ↓ | ↑ | ↓ | ↓ | DIP 4: DieBox-Digi-Frontend |
| x | x | x | ↓ | ↓ | ↑ | ↓ | ↑ | DIP 5: Low Cost Digi |
| x | x | x | ↓ | ↓ | ↑ | ↑ | ↓ | DIP 6: HighSpeedBus |
| x | x | x | ↓ | ↓ | ↑ | ↑ | ↑ | DIP 7: Token-Ring --> EINSTELLUNG !! |
| x | x | x | ↓ | ↑ | ↓ | ↓ | ↓ | DIP 8: Mixed |
| x | x | x | ↓ | ↑ | ↓ | ↓ | ↑ | DIP 9: frei |
| x | x | x | ↓ | ↑ | ↓ | ↑ | ↓ | DIP 10: frei |
| x | x | x | ↓ | ↑ | ↓ | ↑ | ↑ | DIP 11: frei |
| x | x | x | ↓ | ↑ | ↓ | ↑ | ↑ | DIP 12: frei |
| x | x | x | ↓ | ↑ | ↓ | ↑ | ↑ | DIP 13: frei |
| x | x | x | ↓ | ↑ | ↓ | ↑ | ↑ | DIP 14: frei |
| x | x | x | ↓ | ↑ | ↓ | ↑ | ↑ | DIP 15: Benutzerdefinierte Konfiguration |
| x | x | x | ↑ | ↑ | ↑ | ↑ | ↑ | DIP 31: Totalreset, alle Daten im RAM löschen |

DIP-Schalterstellung Terminal-Baudrate TNC3 (Token-Ring-Master)

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Terminal-Baudrate |
|---|---|---|---|---|---|---|---|---------------------------|
| ↓ | ↓ | ↓ | x | x | x | x | x | Externer Takt (*16) |
| ↓ | ↓ | ↑ | x | x | x | x | x | Token Baudrate 2400 Baud |
| ↓ | ↑ | ↓ | x | x | x | x | x | Token Baudrate 4800 Baud |
| ↓ | ↑ | ↑ | x | x | x | x | x | Token Baudrate 9600 Baud |
| ↑ | ↓ | ↓ | x | x | x | x | x | Token Baudrate 19k2 Baud |
| ↑ | ↓ | ↑ | x | x | x | x | x | Token Baudrate 38k4 Baud |
| ↑ | ↑ | ↓ | x | x | x | x | x | Token Baudrate 57k6 Baud |
| ↑ | ↑ | ↑ | x | x | x | x | x | Token Baudrate 115k2 Baud |

Quelle: DL2GWA