

5. Hardwarebeschreibung zum C164CI-Experimentalsystem

5.1. Allgemeine Beschreibung

Das C164CI-Experimentalsystem wurde entworfen, um das nanoModul-164, welches von der Firma **Phytec** stammt und mit dem Microcontroller **SAF-C164CI-8EM** von der Firma **Infineon** bestückt ist, an gegebene Aufgaben (Steuerung einer Eisenbahn) anzupassen bzw. mit entsprechender Peripherie für die Eingabe und Ausgabe auszustatten.

Über eine serielle Schnittstelle hat der Bediener die Möglichkeit Programme in den Speicher des Microcontrollers zu laden und somit spezielle Aufgabe zu erfüllen. Dabei kann über das Taster- und Schalterfeld die Abarbeitung des Programmes gesteuert werden, d.h. Anweisungen beantworten, neue Werte eingeben und vieles mehr, sowie über eine Reihe von 7-Segmentanzeigen Zustände bzw. Nachrichten angezeigt werden.

Mittels weiterer Schnittstellen, die sich auf dem C164CI-Experimentalsystem befinden, ist der Benutzer in die Lage versetzt andere Platinen anzuschließen. Eine solche, ist die Pfostenstecker-Schnittstelle die mit den Datenbus und den Adressbus ausgestattet ist.

Außerdem besitzt die Leiterplatte eine CAN-Schnittstelle, die es dem Microcontroller erlaubt, eine Knotennummer innerhalb eines CAN-Bussystems einzunehmen.

Das entwickelte C164CI-Experimentalsystem wird dabei von einem 5V geregelten Netzteil betrieben.

Alle hier genannten Komponenten (Microcontroller, Schnittstellen, CAN ...) werden im Laufe dieses Kapitels noch näher erläutert.

5.2. Komponenten des C164CI-Experimentalsystems

Die verwendeten Komponenten des C164CI-Experimentalsystems wurden zu Funktionsgruppen zusammen gefasst. Dies spiegelt sich in der Anordnung der Komponenten wider, was eine bessere Bedienung zu Folge hat.

Die einzelnen Komponenten des C164CI-Experimentalsystems sind in der folgenden Auflistung sowie in der Abbildung 5.1 dargestellt.

- Zentraleinheit (nanoModul-164),
- Adressierung (74HC154),
- Bustreiber (74HC563, 74HC573),
- Eingabeeinheit (Taster, Dil, Schalter),
- Ausgabeeinheit (7-Segmentanzeige),
- Schnittstellen (Pfeifenstecker, T-Schnittstelle...),
- Spannungsversorgung (externe Spannungsquelle),

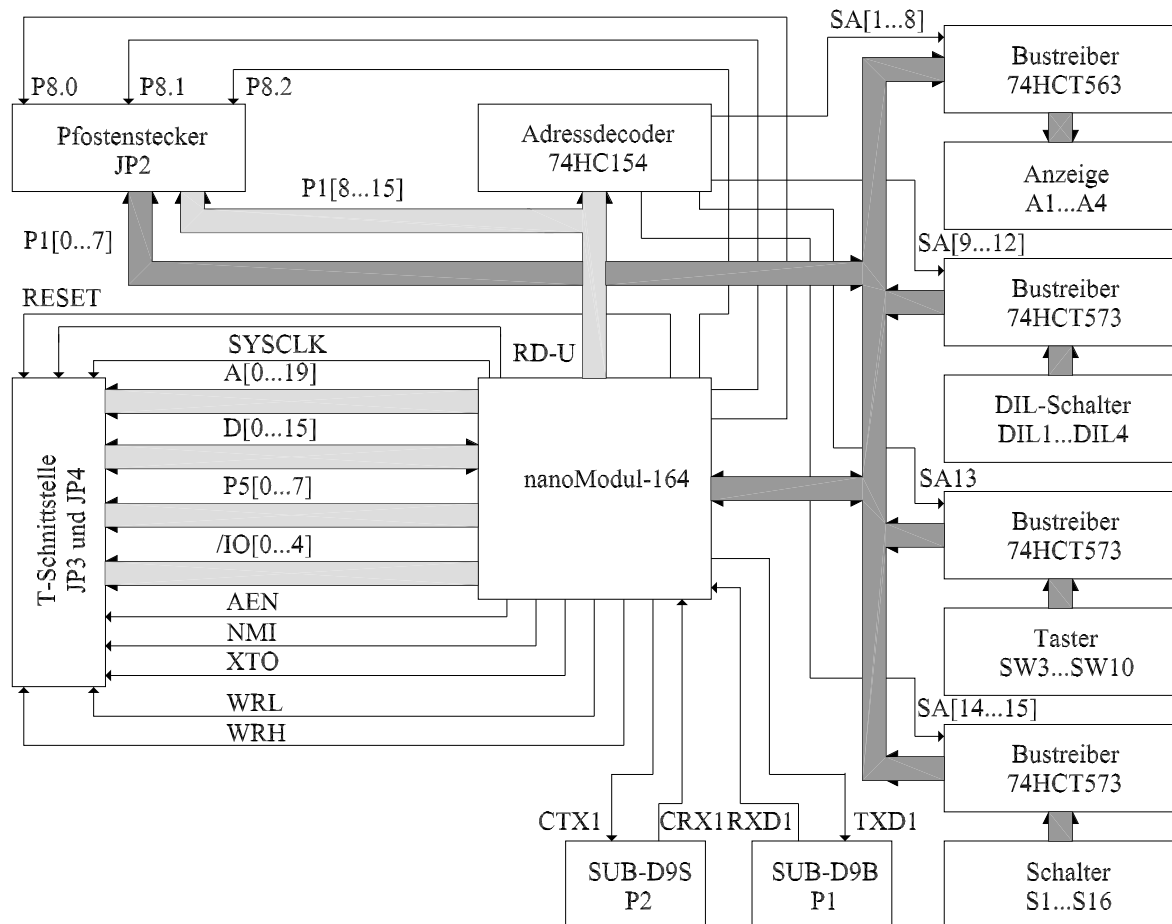


Abbildung 5.1.: Komponenten des C164CI-Experimentalsystems

5.3. nanoModul-164

Das nanoModul-164 [11] wurde von der Firma **Phytec** entwickelt und ist ein universelles Microcontrollerboard im Streichholzschaftelformat, welches mit dem Microcontroller C164CI-8EM (Abschnitt 5.3.1) ausgestattet ist. Es werden zwei zweireihige Anschlussleisten mit je 30 Pins verwendet, um alle erforderlichen Signale des Microcontrollers nach außen zu führen. Das nanoModul-164 besitzt folgende Features:

- Verbesserte Störsicherheit durch Multilayertechnik
- 256 KByte Flash

- 256 KByte SRAM
- On-Chip Bootstrap Loader
- RS-232 Schnittstelle
- CAN-Schnittstelle
- Echtzeituhr
- 4 freie Chip Select Signale, für Anschluss externer Peripherie

5.3.1. Microcontroller C164CI-8EM

Der verwendete Microcontroller stammt von der Firma **Infineon** und gehört zur Familie der 16 Bit Microcontroller (weiter sind C161, C163, C166, C167). Im folgenden sollen einige Funktionsbaugruppen des C164CI-8EM (Abbildung 5.2) vorgestellt und kurz erläutert werden [10].

Die **CPU** ist ein 16 Bit Kern. Die Befehle durchlaufen eine 4-fach Instruction Pipeline, dabei besteht jeder Befehl aus 4 Phasen, der

- **FETCH** – Befehlsholphase
- **DECODE** – Decodierung des Befehls und laden von Operanden
- **EXECUTE** – Ausführen des Befehls
- **WRITE BACK** – Ergebnis des Befehls wird in den Speicher geschrieben

Die **Interrupts** spielen im Bereich der Embedded Systeme eine große Rolle. Aus diesem Grund besitzt der Microcontroller 32 solche Interruptquellen, von denen jede mit einer Priorität versehen werden kann. 16 Interrupts werden von externen Quellen ausgelöst und die anderen 16 werden von den internen Peripherieeinheiten verwendet.

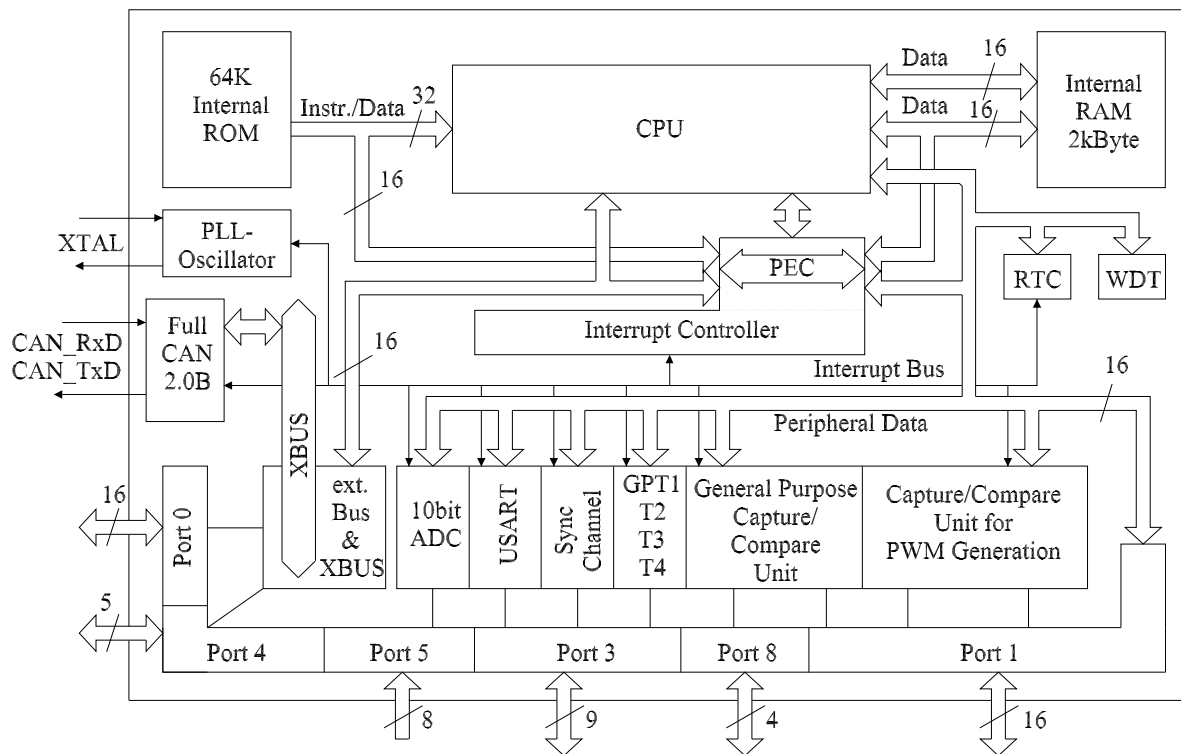


Abbildung 5.2.: Funktionsblöcke des SAF-C164CI-8EM

Häufig muss aber keine Aktion durchgeführt werden, wenn ein Interrupt ansteht. Stattdessen sollen einfach nur Daten von der Peripherie in den RAM-Speicher transferiert werden. Für diesen Zweck steht der **PEC (Peripheral Event Controller)** zur Verfügung. Er reagiert auf Interrupts und überträgt dabei Daten in den Speicher, ohne die CPU in ihrer Arbeit zu unterbrechen.

Die **Capture/Compare Units** sind geeignet für die Erzeugung und Erfassung von komplexen Signalverläufen, wie z.B. Impulsgenerierung, Pulsweitenmodulation, Impulsbreitenmessung. Die CAPCOM2-Unit besitzt zwei 16 Bit Timer (T7, T8) mit Reloadregister. Die CAPCOM6-Unit einen 16 Bit Timer (T12) und einen 10 Bit Timer (T13). Im Capture-Betrieb wird der Inhalt eines laufenden Timers, bei Auftreten eines externen Ereignisses, in das Capture-Register geladen. Im Compare-Betrieb wird der Stand eines laufenden Timers mit dem Inhalt eines Compare-Register verglichen, bei Gleichheit wird ein Interrupt ausgelöst.

Die **General Purpose Timer (GPT) Unit** besitzt eine flexible Timer/Counter-Struktur, die für alle Zeitmessungen verwendet werden kann. Die GPT-Unit verfügt über drei 16 Bit Timer (T2, T3, T4).

Der **Watchdog Timer** ist ein Schutzmechanismus. Bei Programmfehlern und äußeren Störsignalen kommt er zum Einsatz und ermöglicht so ein schnelles Anlaufen des gesamten Systems. Er besteht aus einem 16 Bit Zähler, der bei Überlaufen ein Reset auslöst.

Die **Real Time Clock (RTC)** ist verantwortlich für die korrekte Zeit und das Datum. Dabei kann über einen 48 Bit Timer eine maximale Laufzeit von über 100 Jahren erreicht werden. Die Real Time Clock besteht aus einer Kette von 3 Teilblöcken, einem 8:1 Teiler, einem 16 Bit Timer T14 und einem 32 Bit RTC Timer.

Der **A/D-Converter** verfügt über eine 10 Bit Auflösung, mit vorgeschalteten Sample und Hold Glied. Ein Multiplexer wählt zwischen 8 analogen Inputkanäle aus. Es ist sowohl Einzelwandlung, als auch Mehrfachwandlung mit Interruptbetrieb möglich.

Es gibt zwei **Serielle Channele**. Einen asynchronen/synchronen seriellen Kanal (ASC0) und einen High-Speed synchronen seriellen Kanal (SSC). Jede Schnittstelle besitzt ihren eigenen Baudratengenerator, der eine Übertragungsgeschwindigkeit bis 625 KBAud (ASC0 – Full Duplex Asynchronkommunikation), bis 2.5 MBAud (ASC0 – Halb Duplex Synchronkommunikation, bis 5 MBAud (SSC – Full Duplex Synchronkommunikation) zu lässt.

Der **On-Chip-CAN-Controller** wurde schon im Abschnitt 4.3 erläutert.

Die **Parallelen Ports** sind die I/O-Ports des Microcontrollers und dienen dem Anschluss externer Geräte. Er besitzt 6 Ports mit unterschiedlicher Breite (16, 9, 8, 5 und 4 Bit), wobei einige der I/O-Ports mit Sonderfunktionen ausgestattet sind. Jeder Pin kann einzeln genutzt und eingestellt werden.

5.3.2. Konfiguration des nanoModuls-164

Die Jumperleiste JP1 (Lageplan auf Seite 58) dient der Konfiguration des nanoModuls-164. In der Tabelle 5.1 sind die verschiedenen Einstellungen aufgelistet. Die dazu gehörige Position der Jumper kann in der Abbildung 5.3 ermittelt werden.

PIN	nicht verbunden	verbunden	Bedeutung
1-2	—		Speichermodell FLASH/RAM
1-2		—	Speichermodell RAM/FLASH
3-4	—		/CS deaktiviert
3-4		—	/CS aktiviert
5-7	—		Boot-Taster mit NMI-Pin nicht verbunden
5-7		—	Boot-Taster mit NMI-Pin verbunden
7-9	—		Boot-Taster mit Boot-Pin nicht verbunden
7-9		—	Boot-Taster mit Boot-Pin verbunden
11-12	—		Boot-Taster aktiv
11-12		—	Boot-Taster deaktiviert

Tabelle 5.1.: Jumperbelegung

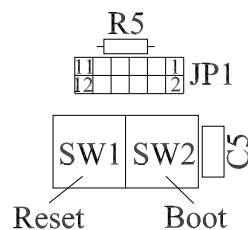


Abbildung 5.3.: Aufbau Jumperleiste JP1

5.4. Adressraum des C164CI-Experimentalsystems

Die Adressierung der einzelnen Peripheriebauelemente (Schiebeschalter, Taster, 7-Segmentanzeigen) ist mit einem 4 zu 16 Adressdecoder (**74HC154**) realisiert. Als Ausgangspunkt des Adressbusses dient das 8 Bit Port P1H (Seite A.4 P1[8...15]) des Microcontrollers. Für die Adressierung der Peripheriebauelemente des C164CI-Experimentalsystems werden die 4 höchsten Bits des Ports P1H verwendet.

In der Tabelle 5.2 sind die einzelnen Adressen aufgelistet. Im Anhang auf Seite 58 kann die Lage der entsprechenden Peripheriebauelemente ermittelt werden.

Adressebus (P1H)	Peripheriebauelemente
10 _H	A1/1
20 _H	A1/2
30 _H	A2/1
40 _H	A2/2
50 _H	A3/1
60 _H	A3/2
70 _H	A4/1
80 _H	A4/2
90 _H	DIL 1
A0 _H	DIL 2
B0 _H	DIL 3
C0 _H	DIL 4
D0 _H	Taster SW3–SW10
E0 _H	Schiebeschalter S1–S8
F0 _H	Schiebeschalter S9–S16

Tabelle 5.2.: Adressraum der Leiterplatte

5.5. Eingabeeinheiten

Durch die entsprechenden Eingabeeinheiten, die in den folgenden Abschnitten näher erläutert werden, ist es dem Bediener möglich, feste Werte vorzugeben, Abläufe zu steuern und noch vieles anderes.

Für das Abfragen der Eingabegeräte werden 8 Bit nichtinvertierende Bustreiber mit D-Latch (Zwischenspeicher) und Tri-State-Ausgängen verwendet (**74HCT573**). Sie sind so beschaltet, dass der binäre Wert des Eingabegerätes am Ausgang des Bustreibers anliegt. Die Ausgänge des Bustreibers liegen am 8 Bit Datenbus an, der am Port P1L (Seite A.4 P1[0...7]) des Microcontrollers endet.

5.5.1. DIL-Schalter

Auf dem C164CI-Experimentalsystem befinden sich 4 DIL-Schalterblöcke mit jeweils 8 Schaltern. Dabei können die aktuellen Werte der einzelnen Schalter über ihre jeweiligen Adressen (90_H , $A0_H$, $B0_H$, $C0_H$) eingelesen werden. Die Verschaltung der DIL-Schalter wurde so vorgenommen, dass der linke Schalter, die kleinste binäre Einheit bildet, d.h. bei einer Einstellung eines 8 Bit Wertes ist der linke Schalter das 0 Bit, siehe Abbildung 5.4.

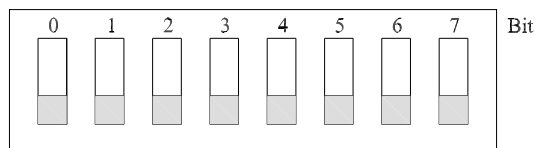


Abbildung 5.4.: Anordnung der 8 Bits beim DIL-Schalter

Da die DIL-Schalter nicht für den häufigen Gebrauch geeignet sind, sollte die Verwendung auf einen festen Wert beschränkt werden, z.B. Einstellen eines Identifiers, 11 Bit oder 29 Bit, beim CAN-Bus.

5.5.2. Schiebeschalter

Die 16 Schiebeschalter, die sich auf dem C164CI-Experimentalsystem befinden, sind zu je 2 Blöcken zusammen geschaltet und können über ihre entsprechende Adresse abgefragt werden ($E0_H$: S1–S8, $F0_H$: S9–S16). Die Verschaltung der Schiebeschalter wurde genau wie den DIL-Schaltern vorgenommen (kleinstes Bit ist links).

Da die Schiebeschalter robuster als die DIL-Schalter sind, sollten sie auch bevorzugt benutzt werden.

5.5.3. Taster

Auf dem C164CI-Experimentalsystem befinden sich 8 Taster ($D0_H$), die es dem Bediener erlauben Programmabfragen mit einem Tastendruck zu bestätigen.

Der linke obere Taster (SW3) ist das 0. Bit und der linken unteren Taster (SW7) das 4. Bit. Die Bitgröße steigt nach rechts an.

5.6. Ausgabeeinheit

Der Anwender hat die Möglichkeit, über eine Reihe von 7-Segmentanzeigen, Programmabschnitte, die hervorzuheben sind, optisch darzustellen. Dies können bestimmte Werte oder Anweisungen sein, die mit einer Bestätigung quittiert werden müssen oder einfach eine Begrüßungsschrift.

Für die Ansteuerung der 7-Segmentanzeigen wurden 8 Bit invertierender Bustreiber mit D-Latch (Zwischenspeicher) und Tri-State-Ausgängen verwendet (**74HCT563**). Sie sind so beschaltet, dass der binäre Wert gehalten wird und invertiert am Bustreiber Ausgang anliegt. Die Eingänge der Bustreiber sind an den 8 Bit Datenbus geschaltet, der am Port P1L des Microcontrollers anliegt.

Die verwendete Anzeige ist eine doppelte 7-Segmentanzeige, die von der Firma **Hewlett Packard** stammt und die Bezeichnung **HDSP5521** trägt. Die Abbildung 5.5 zeigt die einzelnen Segmente der Anzeige, die in der Tabelle 5.3 durch den entsprechenden Wert des Datenbuses angesprochen werden.

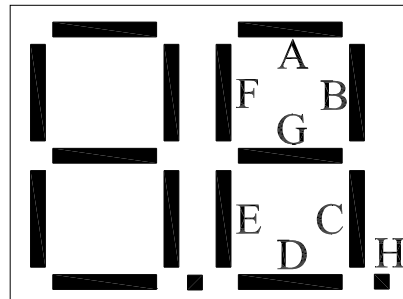


Abbildung 5.5.: Anordnung der Segmente

Datenbus (P1L)	Segment
01H	A
02H	B
04H	C
08H	D
10H	E
20H	F
40H	G
80H	H

Tabelle 5.3.: Ansteuerung der 7-Segmentanzeige

5.7. Schnittstellen

Auf der Leiterplatte befinden sich die verschiedensten Schnittstellen, die den Benutzer in die Lage versetzt, einmal dem Microcontroller zu programmieren zum anderen Erweiterungen vorzunehmen oder Daten auszutauschen. In den nachfolgenden Abschnitten werden diese Schnittstellen und ihre Funktionen erläutert.

5.7.1. Serielle Schnittstelle

Die serielle Schnittstelle (P1, Seite 58) dient zum programmieren des Microcontrollers. Dafür benötigt man einen IBM-kompatiblen PC mit einer seriellen Schnittstelle und ein entsprechendes Schnittstellenkabel, welches dann mit der DB9-Buchse des C164CI-Experimentalsystem verbunden wird.

5.7.2. CAN-Schnittstelle

Durch die CAN-Schnittstelle (P2, Seite 58) kann das C164CI-Experimentalsystem ein Netzknoten innerhalb eines CAN-Netzes sein.

Dabei bildet der Schaltkreis **PCA82C250**, der von der Firma **Philips** stammt, die Verbindung zwischen dem CAN-Protokollcontroller und dem physikalischen Bus. Die Verbindung zum physikalischen Bus wird über den DB9-Stecker erreicht.

5.7.3. Pfostenstecker-Schnittstelle

Die Pfostenstecker-Schnittstelle (JP2, Seite 58) besitzt auf der Leiterplatte einen zweireihigen Pfostensteckverbinder der mit den Datenbus (P1L) und dem Adressbus (P1H) verbunden ist, sowie 3 weiteren Leitungen die für eventuelle Steuerungszwecke genutzt

werden können. Im Anhang auf Seite 68 kann die Pinbelegung der Pfostenstecker-Schnittstelle nachgelesen werden.

5.7.4. T-Schnittstelle

Der Name der Schnittstelle (JP3 und JP4, Seite 58) ist frei gewählt und bezieht sich auf die Anordnung (Abbildung 5.6) der beiden Buchsenleisten. Dabei handelt es sich um zwei zweireihige Anschlussbuchsen mit je 25 Pins und mit je 10 Pins.

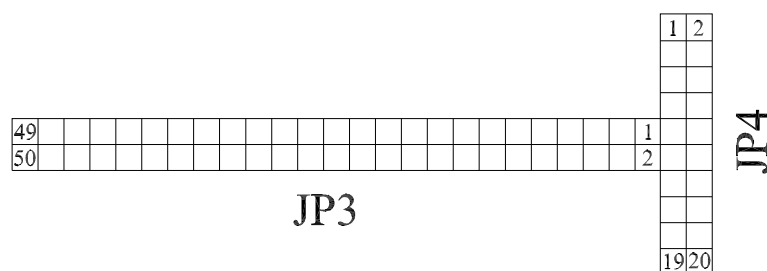


Abbildung 5.6.: Aufbau der T-Schnittstelle

Die T-Schnittstelle ist mit Leitungen (Datenleitung, Adressleitung) und Signalen (/IO, /RD, /WR, ...) ausgestattet, die notwendig sind um eine Erweiterung des nanoModuls-164, hinsichtlich ihres Speichers zu machen bzw. einen direkten Eingriff des Speichers vorzunehmen. Außerdem sind die analogen Inputkanäle des A/D-Converter zur Schnittstelle geführt. Die Referenzspannung des A/D-Converter ist auf den Wert 5V eingestellt.

Im Anhang auf den Seiten 69 und 70 kann die Pinbelegung der T-Schnittstelle nachgelesen werden.