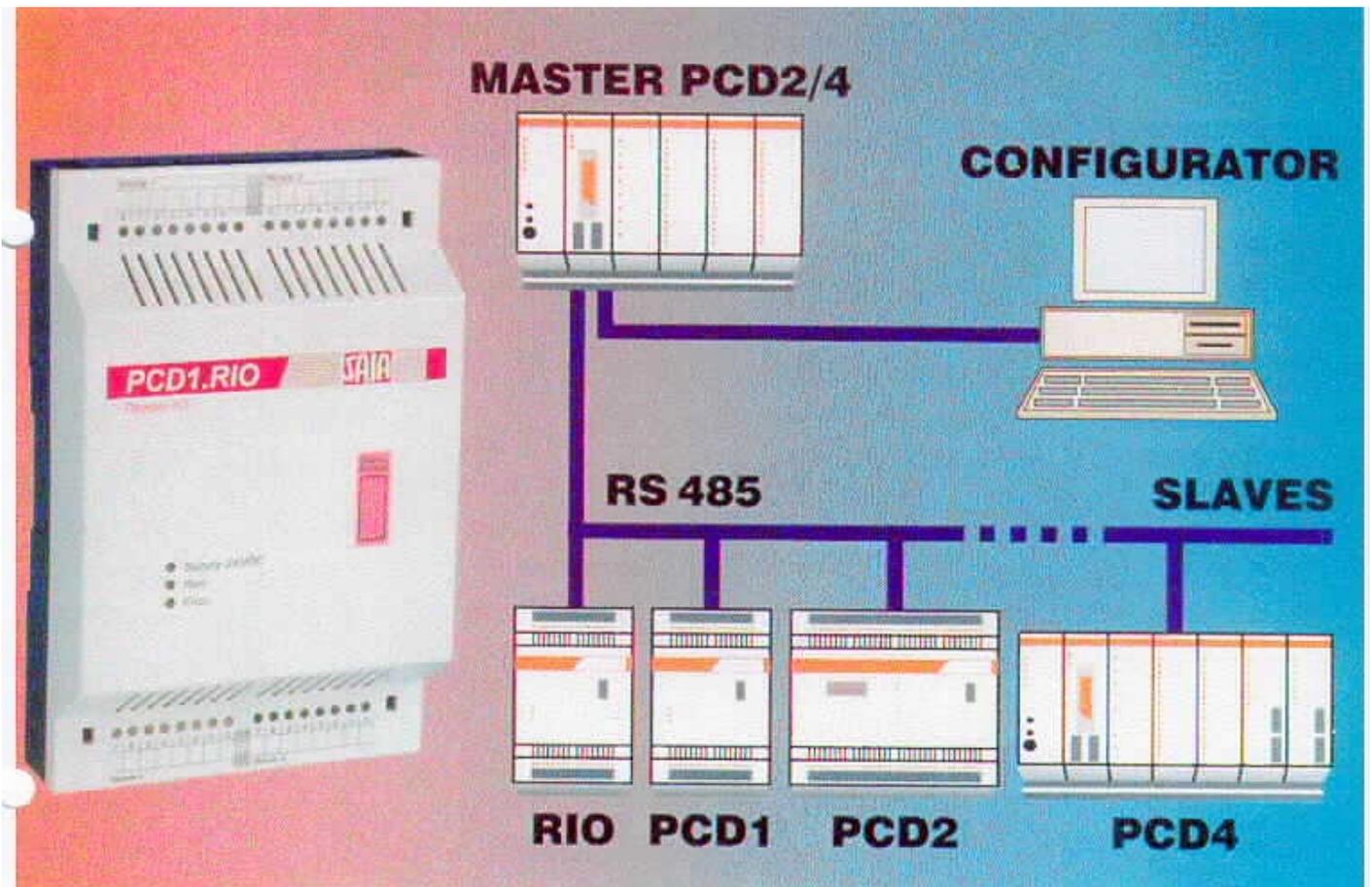


SAIA® PCD Process Control Devices

PCD1.T240 SAIA® Remote I/O



SAIA-Burgess Gesellschaften

Schweiz	SAIA-Burgess Electronics AG Freiburgstrasse 33 CH-3280 Murten ☎ 026 672 77 77, Fax 026 670 19 83	Frankreich	SAIA-Burgess Electronics Sàrl. 10, Bld. Louise Michel F-92230 Gennevilliers ☎ 01 46 88 07 70, Fax 01 46 88 07 99
Deutschland	SAIA-Burgess Electronics GmbH Daimlerstrasse 1k D-63303 Dreieich ☎ 06103 89 060, Fax 06103 89 06 66	Niederlande	SAIA-Burgess Electronics B.V. Hanzeweg 12c NL-2803 MC Gouda ☎ 0182 54 31 54, Fax 0182 54 31 51
Österreich	SAIA-Burgess Electronics Ges.m.b.H. Schallmooser Hauptstrasse 38 A-5020 Salzburg ☎ 0662 88 49 10, Fax 0662 88 49 10 11	Belgien	SAIA-Burgess Electronics Belgium Avenue Roi Albert 1er, 50 B-1780 Wemmel ☎ 02 456 06 20, Fax 02 460 50 44
Italien	SAIA-Burgess Electronics S.r.l. Via Cadamosto 3 I-20094 Corsico MI ☎ 02 48 69 21, Fax 02 48 60 06 92	Ungarn	SAIA-Burgess Electronics Automation Kft. Liget utca 1. H-2040 Budaörs ☎ 23 501 170, Fax 23 501 180

Vertretungen

Gross-britannien	Canham Controls Ltd. 25 Fenlake Business Centre, Fengate Peterborough PE1 5BQ UK ☎ 01733 89 44 89, Fax 01733 89 44 88	Portugal	INFOCONTROL Electronica e Automatismo LDA. Praceta Cesário Verde, No 10 s/cv, Massamá P-2745 Queluz ☎ 21 430 08 24, Fax 21 430 08 04
Dänemark	Malthe Winje Automation AS Håndværkerbyen 57 B DK-2670 Greve ☎ 70 20 52 01, Fax 70 20 52 02	Spanien	Tecnosistemas Medioambientales, S.L. Poligono Industrial El Cabril, 9 E-28864 Ajalvir, Madrid ☎ 91 884 47 93, Fax 91 884 40 72
Norwegen	Malthe Winje Automasjon AS Haukelivn 48 N-1415 Oppegård ☎ 66 99 61 00, Fax 66 99 61 01	Tschechische Republik	ICS Industrie Control Service, s.r.o. Modranská 43 CZ-14700 Praha 4 ☎ 2 44 06 22 79, Fax 2 44 46 08 57
Schweden	Malthe Winje Automation AB Truckvägen 14A S-194 52 Upplands Väsby ☎ 08 795 59 10, Fax 08 795 59 20	Polen	SABUR Ltd. ul. Druzynowa 3A PL-02-590 Warszawa ☎ 22 844 63 70, Fax 22 844 75 20
Suomi/ Finnland	ENERGEL OY Atomitie 1 FIN-00370 Helsinki ☎ 09 586 2066, Fax 09 586 2046		
Australien	Siemens Building Technologies Pty. Ltd. Landis & Staefa Division 411 Ferntree Gully Road AUS-Mount Waverley, 3149 Victoria ☎ 3 9544 2322, Fax 3 9543 8106	Argentinien	MURTEN S.r.l. Av. del Libertador 184, 4° "A" RA-1001 Buenos Aires ☎ 054 11 4312 0172, Fax 054 11 4312 0172

Kundendienst

USA	SAIA-Burgess Electronics Inc. 1335 Barclay Boulevard Buffalo Grove, IL 60089, USA ☎ 847 215 96 00, Fax 847 215 96 06
------------	---

SAIA® Process Control Devices

**SAIA® Remote I/O
PCD1.T240**

für die PCD-Familie

SAIA-Burgess Electronics AG 1996. Alle Rechte vorbehalten
Ausgabe 26/751 D2 - 08.1996

Technische Änderungen vorbehalten

Anpassungen

Handbuch: SAIA[®] Remote I/O PCD1.T240 - Ausgabe D2

Datum	Abschnitt	Seite	Beschreibung

Inhalt

	Seite
1. Beschreibung und Eigenschaften	1-1
1.1 Was ist PCD-RIO?	1-1
1.2 Eigenschaften	1-2
2. SAIA® PCD-Geräte für ein PCD-RIO-Netzwerk	2-1
2.1 Welche Geräte können als Master eingesetzt werden?	2-1
2.2 Welche Geräte können als Slaves eingesetzt werden?	2-2
3. Hardware der PCD1.RIO	3-1
3.1 Allgemeines	3-1
3.2 Technische Daten der PCD1.RIO	3-2
3.2.1 Spezifische Daten	3-2
3.2.2 Allgemeine technische Daten	3-3
3.3 Präsentation und Abmessungen	3-4
3.3.1 Präsentation der PCD1.RIO	3-4
3.3.2 Montage, Massbild und Kabelführung	3-5
3.4 Architektur und Präsentation des Hauptprints	3-6
3.4.1 Architektur	3-6
3.4.2 Präsentation des Hauptprints	3-7
3.5 Die RS 485-Schnittstelle	3-10
3.5.1 Anschluss der RS 485-Schnittstelle	3-10
3.5.2 Verlegung der RS 485 Bus-Leitung	3-11
3.6 Adressierung der Ein- und Ausgangsmodule	3-14

	Seite
4. Konzept	4-1
4.1 Allgemeine Beschreibung	4-1
4.2 Dateien für die RIO-Belange	4-3
4.3 Aktivierung der RIO-Aufgabe	4-6
4.4 Diagnose-Möglichkeiten	4-7
4.5 Verwendung des RIO-Kanals für asynchrone Telegrammübertragungen	4-8
4.6 Auffrisch- und Reaktionszeiten	4-9
4.7 Erforderliche Firmware	4-10
5. Der RIO-Konfigurator	5-1
5.1 Einführung	5-1
5.2 Installation des RIO-Konfigurators	5-2
5.3 Aufruf und Beschreibung des RIO-Konfigurators	5-3
5.3.1 Das Untermenü 'Network' (Netzwerk)	5-4
5.3.2 Das Untermenü 'Edit'	5-9
5.3.3 'Station Parameter...' für die Master-Station	5-11
5.3.4 'Station Parameter...' für eine nicht-intelligente Slave-Station	5-14
5.3.5 'Station Parameter...' für eine intelligente Slave-Station	5-22
5.3.6 Das Untermenü 'View'	5-25
5.3.7 Das Untermenü 'Library'	5-26
5.3.8 Das Untermenü 'Window'	5-27
5.3.9 Das Untermenü 'Help'	5-28
5.3.10 Die Funktionsleiste (Toolbar)	5-29

		Seite
6.	Programmierung	6-1
6.1	Einführendes Beispiel	6-1
6.1.1	Aufgabe	6-1
6.1.2	Konfigurierung der Stationen	6-3
6.1.3	Konfig. der I/O-Module der Slave-Stationen	6-4
6.1.4	Generieren der Hilfsdateien	6-8
6.1.5	Die Definitionsdatei	6-10
6.1.6	Die Konfigurationsdatei	6-11
6.1.7	Die "RIO"-Datei	6-12
6.1.8	Das Anwenderprogramm	6-15
6.1.9	Inbetriebnahme	6-17
6.2	Weiteres Beispiel	6-19
6.2.1	Aufgabe	6-19
6.2.2	Konfigurierung der Stationen	6-21
6.2.3	Konfigurierung der Slave-Stationen	6-22
6.2.4	Generieren der Hilfsdateien	6-28
6.2.5	Die Definitionsdatei	6-29
6.2.6	Die Def.-Datei der intelligenten Slave-Station	6-31
6.2.7	Die Konfigurationsdatei	6-32
6.2.8	Das Anwenderprogramm für die Master-Station	6-33
6.2.9	Das Anwenderprogr. für die intelligente Slave-St.	6-37
6.2.10	Inbetriebnahme	6-39
6.3	Häufige Fehlerursachen	6-41
7.	Diagnose	7-1
7.1	Überwachung der Anwesenheit der Master-Station	7-1
7.2	Überwachung der Anwesenheit der Slave-Stationen (XOB 30)	7-2
7.4	Diagnose-Flags	7-3
7.4	Diagnose-Register	7-6
7.5	Diagnose-DB	7-8
7.6	Konfigurierung	7-10
7.6.1	Konfigurierung der PCD1.RIO-Stationen	7-10
7.6.2	Konfigurierung der zyklischen "Messages"	7-13
7.7	Diagnose in der praktischen Anwendung	7-15



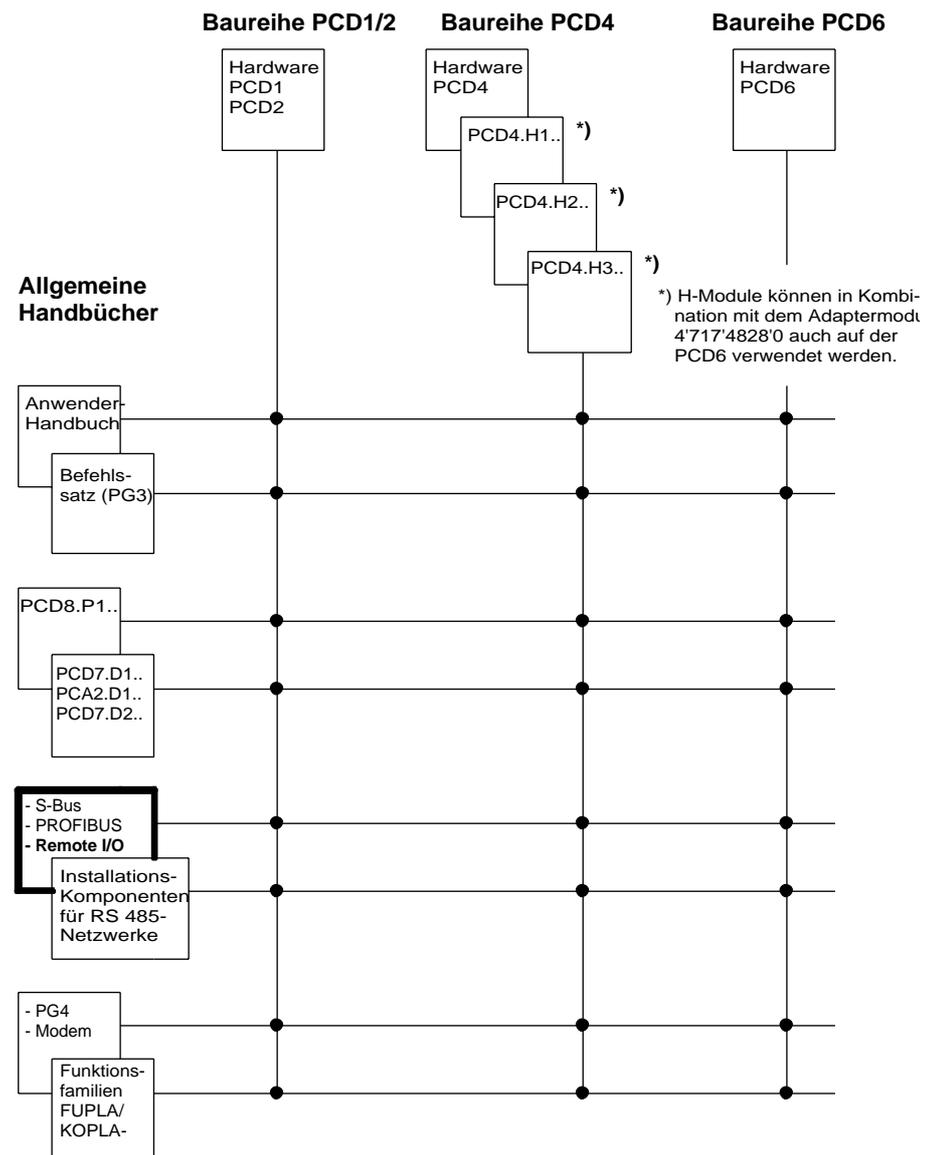
Wichtiger Hinweis:

Um den einwandfreien Betrieb von SAIA® PCD sicherstellen zu können, wurde eine Vielzahl detaillierter Handbücher geschaffen. Diese wenden sich an technisch qualifiziertes Personal, das nach Möglichkeit auch unsere Workshops erfolgreich absolviert hat.

Die vielfältigen Leistungen der SAIA® PCD treten nur dann optimal in Erscheinung, wenn alle in diesen Handbüchern aufgeführten Angaben und Richtlinien bezüglich Montage, Verkabelung, Programmierung und Inbetriebnahme genau befolgt werden.

Damit allerdings werden Sie zum grossen Kreis der begeisterten SAIA® PCD Anwendern gehören.

Übersicht



Zuverlässigkeit und Sicherheit elektronischer Steuerungen

Die Firma SAIA-Burgess Electronics AG konzipiert, entwickelt und stellt ihre Produkte mit aller Sorgfalt her:

- Neuster Stand der Technik
- Einhaltung der Normen
- Zertifiziert nach ISO 9001
- Internationale Approbationen: z.B. Germanischer Lloyd, UL, Det Norske Veritas, CE-Zeichen ...
- Auswahl qualitativ hochwertiger Bauelemente
- Kontrollen in verschiedenen Stufen der Fertigung
- In-Circuit-Tests
- Run-in (Wärmelauf bei 85°C während 48h)

Die daraus resultierende hochstehende Qualität zeigt trotz aller Sorgfalt Grenzen. So ist z.B. mit natürlichen Ausfällen von Bauelementen zu rechnen. Für diese gibt die Firma SAIA-Burgess Electronics AG Garantie gemäss den "Allgemeinen Lieferbedingungen".

Der Anlagebauer seinerseits muss auch seinen Teil für das zuverlässige Arbeiten einer Anlage beitragen. So ist er dafür verantwortlich, dass die Steuerung datenkonform eingesetzt wird und keine Überbeanspruchungen, z.B. auf Temperaturbereiche, Überspannungen und Störfelder oder mechanischen Beanspruchungen auftreten.

Darüber hinaus ist der Anlagebauer auch dafür verantwortlich, dass ein fehlerhaftes Produkt in keinem Fall zu Verletzungen oder gar zum Tod von Personen bzw. zur Beschädigung oder Zerstörung von Sachen führen kann. Die einschlägigen Sicherheitsvorschriften sind in jedem Fall einzuhalten. Gefährliche Fehler müssen durch zusätzliche Massnahmen erkannt und hinsichtlich ihrer Auswirkung blockiert werden. So sind z.B. für die Sicherheit wichtige Ausgänge auf Eingänge zurückzuführen und softwaremässig zu überwachen. Es sind die Diagnoseelemente der PCD wie Watch-Dog, Ausnahme-Organisations-Blocks (XOB) sowie Test- und Diagnose-Befehle konsequent anzuwenden.

Werden alle diese Punkte berücksichtigt, verfügen Sie mit der SAIA[®] PCD über eine moderne und sichere programmierbare Steuerung, die Ihre Anlage über viele Jahre zuverlässig steuern, regeln und überwachen wird.

1. Beschreibung und Eigenschaften

1.1 Was ist PCD-RIO?

"RIO" ist die Abkürzung für "Remote Input/Output", was zu deutsch etwa mit "Fernabfrage und Fernbeeinflussung von Ein- und Ausgängen" übersetzt werden kann. Ein RIO im Sinne der PCD-Baureihe beschränkt sich jedoch nicht nur auf digitale Ein- und Ausgänge sondern erstreckt sich auch auf analoge Ein- und Ausgänge sowie in einer weiteren Ausbaustufe auf den Austausch von Register-, Counter- und Timerwerten.

Ein PCD-RIO- Netzwerk basiert auf dem universellen SAIA[®] S-Bus, welcher in allen PCD-Stationen in der Grundausrüstung enthalten ist. Ein RIO-Netzwerk besteht aus einer Master-Station und 1 bis 31 Slave-Stationen. Der Master ist immer eine intelligente Station, d.h. eine PCD mit einer CPU und einem Anwenderprogramm. Die Slave-Stationen können beliebig mit nicht-intelligenten, d.h. mit RIO-Stationen ohne CPU und ohne Anwenderprogramm und intelligenten PCD-Stationen gemischt werden. Unabhängig vom RIO-Mechanismus kann jede intelligente Station die eigenen Anwenderprogramme abarbeiten.

Durch die Zwischenschaltung von bis zu 3 Repeatern, kann das RIO-Netzwerk auf bis zu 125 Stationen ausgebaut werden. Die maximale Übertragungsgeschwindigkeit für den Datenaustausch beträgt 76.8 kBaud.

Das Prinzip des RIO besteht darin, dass in der Master-Station ein Abbild aller Ein- und Ausgänge der unintelligenten Slave-Stationen und bestimmter Ein-/Ausgangsgruppen der intelligenten Stationen aufgebaut ist, welches dauernd aufgefrischt wird. Die Kommunikationsbeziehungen werden im Anwenderprogramm der Master-Station durchgeführt. Will z.B. die Station Nr. 15 einige Eingänge auf der Station Nr. 7 abbilden, so sorgt das Anwenderprogramm im Master dafür, dass die Eingänge der Station Nr. 15 abgefragt und danach an die Ausgänge der Station Nr. 7 geschickt werden.

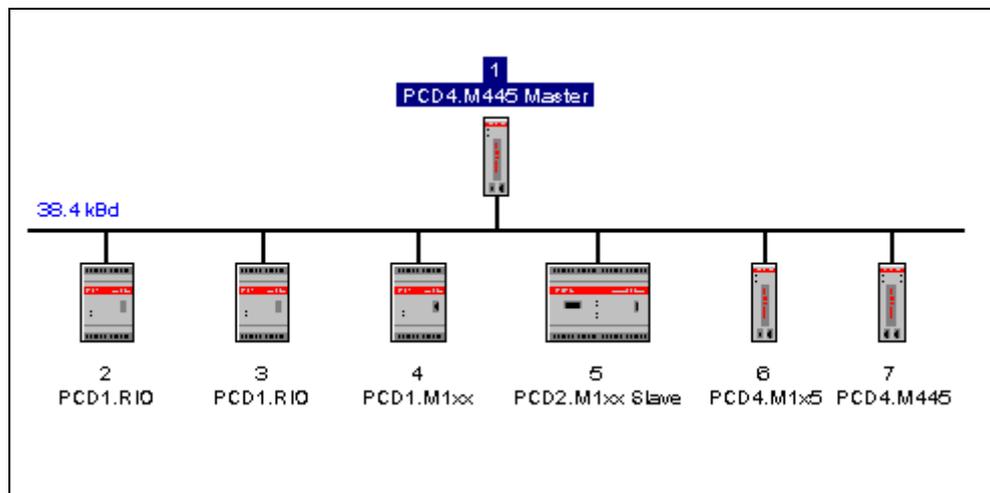
Das dauernde Auffrischen dieses Abbildes braucht natürlich seine Zeit, so dass die preisgünstigen, unintelligenten RIO-Stationen nur dort eingesetzt werden können, wo längere Reaktionszeiten (≥ 100 ms) zulässig sind.

Schnellere Reaktionszeiten werden mit intelligenten Stationen erreicht, wenn die Ereignisse punktuell, d.h. nur dann, wenn das Resultat gebraucht wird, ausgelöst wird. Diese Kommunikation geschieht über das gleiche Netzwerk. Die Ausführungsbefehle werden jedoch im Anwenderprogramm jeder einzelnen Station generiert und gehen nicht via das Abbild in der Master-Station.

1.2 Eigenschaften

- RS 485-Bus mit bewährtem S-Bus Protokoll bis 76.8 kBaud
- Master-Station mit Prozessormodul PCD4.M445, PCD4.M1x5 oder PCD2.M1xx
- Unintelligente Slave-Stationen PCD1.RIO, modular bestückbar mit je 4 digitalen oder analogen Ein-/Ausgangsmodulen
- Intelligente Slave-Stationen als beliebig bestückte PCD1, PCD2 und/oder PCD4
- Bis 31 Slave-Stationen ohne Repeater oder bis 125 Slave-Stationen mit 3 Repeatern PCD7.T100
- Komfortabler Konfigurator unter WINDOWS 3.1 bzw. WINDOWS 95 für die grafisch-menügeführte Konfigurierung des gesamten Netzwerkes.

Beispiel eines kleinen PCD-RIO-Netzwerks:



(Dieses Bildchen wurde mit dem RIO-Konfigurator erstellt).

2. SAIA® PCD-Geräte für ein PCD-RIO-Netzwerk

2.1 Welche Geräte können als Master eingesetzt werden?

Verwendbare Schnittstellen (Ports) und max. Übertragungsraten in kBaud der einsetzbaren (intelligenten) **Master-Stationen**.

Master-PCD	Firmware-Version *)	Port 0	Port 1	Port 2	Port 3
PCD2.M110	V005	38.4	38.4	-	-
PCD2.M120	V005	38.4	38.4	38.4	38.4
PCD4.M125	V00D	38.4	38.4	-	-
PCD4.M145	V00D	38.4	38.4	38.4	38.4
PCD4.M445	2 x V00D	38.4	38.4	76.8	76.8

Bei den PCD4-Geräten hat das Port 0 (PGU) immer die RS 232-Norm. Es ist hier ein Converter RS 232 ↔ RS 485 (PCD7.T120) einzusetzen. Für die Ports 1 bis 3 wird davon ausgegangen, dass das Busmodul PCD4.C340, bestückt mit RS 485-Schnittstellenmodulen (PCD7.F110 oder PCD7.F150) eingesetzt wird. Beim Busmodul PCD4.C130 kann die Schnittstelle Nr. 1 direkt verwendet werden (RS 485), bei den andern Schnittstellen und den andern Busmodulen sind auch die Converter PCD7.T120 einzusetzen.

Die Beschreibungen der PCD4, PCD2 und der PCD1 sind den entsprechenden Hardware-Handbüchern zu entnehmen.

*) siehe Abschnitt 4.7: "Erforderliche Firmware"

2.2 Welche Geräte können als Slaves eingesetzt werden?

Für die **Slave-Stationen** werden in erster Linie die preisgünstigen (unintelligenten) PCD1.RIO-Stationen eingesetzt. Diese können wahlweise mit einer Übertragungsrate von 38.4 oder 76.8 kBaud betrieben werden. Es können auch die in der vorgenannten Tabelle aufgeführten intelligenten (Master-) Geräte als Slave-Stationen eingesetzt werden, wobei für die Schnittstellen-Normen die gleichen Bemerkungen gelten. Auch die intelligenten Geräte der Serie PCD1 können als Slaves eingesetzt werden.

Slave-PCD	Firmware-Version	Port 0	Port 1
PCD1.M110	V001	38.4	-
PCD1.M120	V001	38.4	38.4
PCD1.RIO	V001	-	76.8

In ein- und demselben Netzwerk muss an allen Stationen die gleiche Übertragungsgeschwindigkeit gewählt sein. Die maximale Baudrate von 76.8 kBaud kann also nur in einem Netzwerk mit einer PCD4.M445 auf CPU 1 (mit den Ports 2 oder 3) als Master und den PCD1.RIO-Geräten als Slaves gefahren werden.

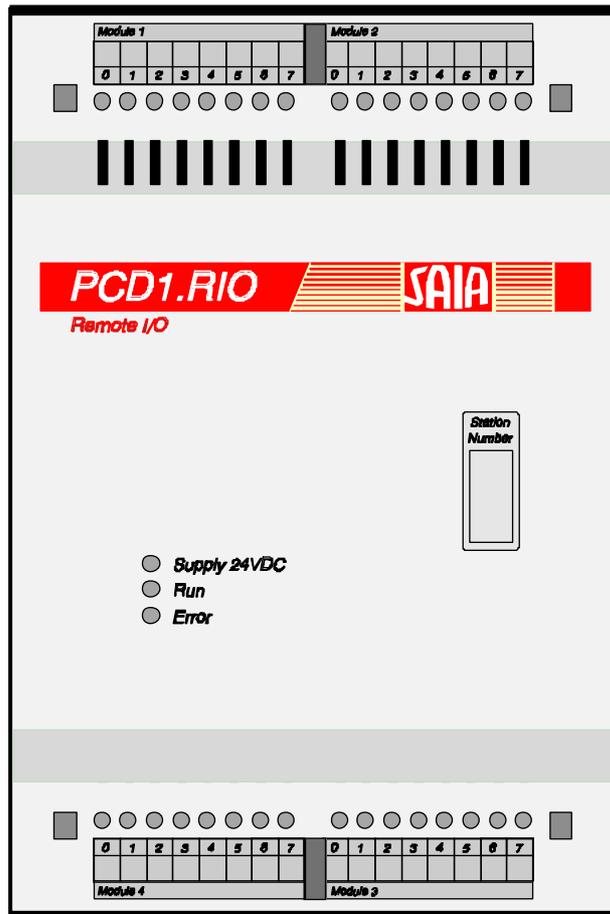
Die Beschreibungen der PCD4, PCD2 und der PCD1 sind den entsprechenden Hardware-Handbüchern zu entnehmen.

Die Beschreibung des PCD1.RIO folgt im nächsten Abschnitt.

3. Hardware der PCD1.RIO (PCD1.T240)

3.1 Allgemeines

Die PCD1.RIO verwendet das gleiche Gehäuse wie die PCD1, hat jedoch keinen PGU-Stecker und ist anders beschriftet. Die PCD1.RIO kann mit 4 PCD2 E/A-Modulen bestückt werden.



Die PCD1.RIO enthält eine RS 485-Schnittstelle zum Anschluss an das RIO-Netzwerk. An einem 8-stelligen DIL-Schalter kann die Stationsnummer und die Übertragungsgeschwindigkeit eingestellt werden. Die PCD1.RIO kann nicht programmiert werden. Ist die PCD1.RIO einmal konfiguriert, gespeist, ans RIO-Netz angeschlossen, die Stationsadresse und die Übertragungsgeschwindigkeit mittels der DIL-Schalter eingestellt, ist das Gerät betriebsbereit und braucht keine weitere Behandlung mehr.

3.2 Technische Daten der PCD1.RIO (PCD1.T240)

3.2.1 Spezifische Daten

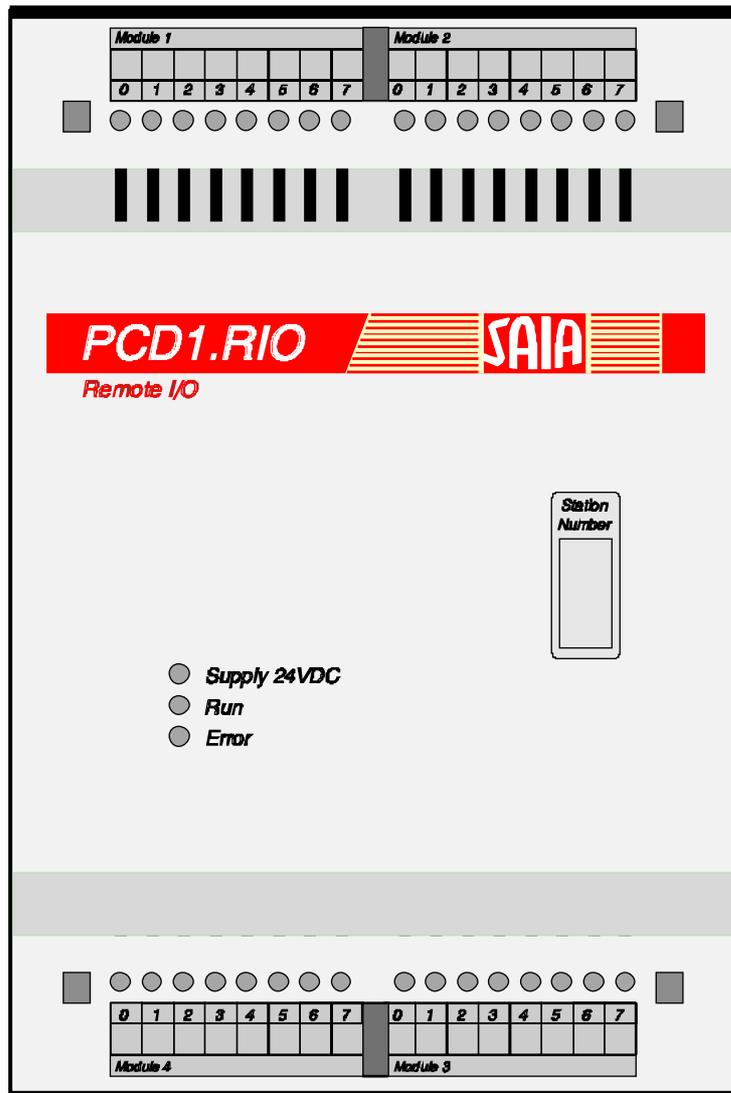
Anzahl Ein-/Ausgänge:	4 digitale PCD2-Ein-/Ausgangsmodule in beliebiger Anordnung oder 4 analoge Ein- oder Ausgangsmodule bzw. digitale und analoge E/A-Module gemischt
Prozessormodul:	1 Zentraleinheit mit einem Mikroprozessor 68340
Anwenderspeicher:	keiner
Hardwareuhr:	keine
Kommunikations-schnittstellen:	keine PGU-Schnittstelle 1 serielles Interface S-Bus/RS 485
Programmiermittel:	keine Anschlussmöglichkeit für ein Programmiergerät

3.2.2 Allgemeine technische Daten

Speisespannung:	24 VDC \pm 20% geglättet, oder 19 VAC \pm 15% zweiweggleichgerichtet (18 VDC)
Leistungsaufnahme:	10W für 16 E/A
Störsicherheit:	Burst nach IEC 801-4, ESD nach IEC 801-2
Sicherheit:	gemäss den Bedingungen nach CE
Umgebungstemperatur:	im Betrieb: 0...+55°C bzw. 0...40°C (siehe Montagelagen) Lagertemp. -20...+85°C
Luftfeuchtigkeit:	95% rel. Feuchte, ohne Betauung (nach DIN 40'040), Klasse F)
Mechanische Festigkeit:	nach IEC 68-2
Montagelagen:	Standard: auf vertikale Fläche (Wandmontage, Klemmen oben und unten) Spezial: auf horizontale Fläche oder Wandmontage mit Klemmen seitlich, bei reduzierter Umgebungstemperatur (max. +40°C)
Anschlüsse:	E/A-Module, Zusatzmodule und Speisung über steckbare Schraubklemmen für Drähte von 1.5 mm ² oder 2 x 0.5 mm ² .
Für Anwendungen nach UL 508:	thermoplastisolierte Cu-Leiter 50/75°C
Anzugsdrehmoment:	0.5 Nm

3.3 Präsentation und Abmessungen

3.3.1 Präsentation der PCD1.RIO



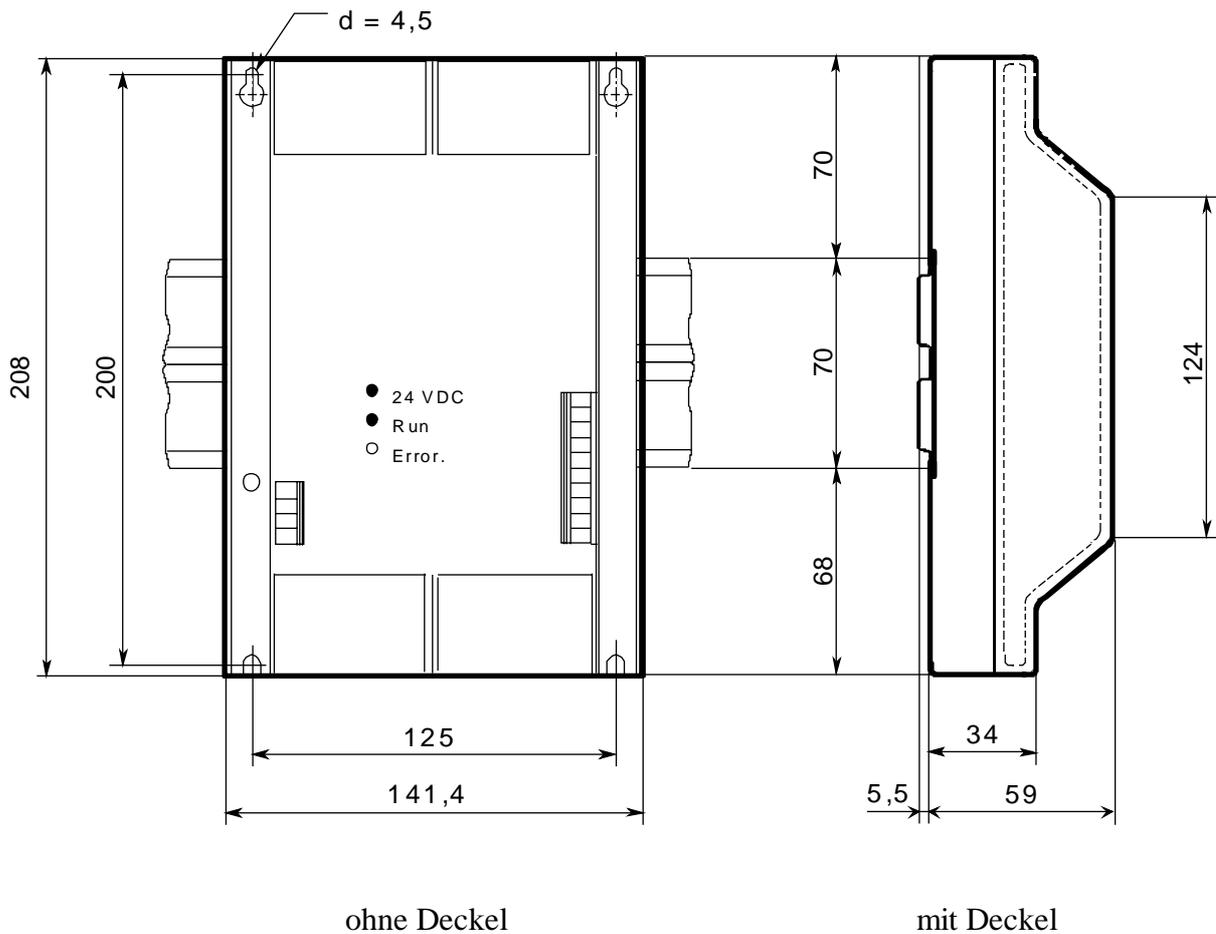
Es besteht nur eine einzige Version des PCD1.RIO:

- Front: Kunststoffabdeckung mit Signallampen (LEDs) zur Anzeige des Betriebsstatus und der Stellung der digitalen Ein- und Ausgänge sowie eine Beschriftungsfläche zum Einschreiben der Stationsnummer.
- Unter der Frontabdeckung liegt der Hauptprint. Dieser enthält neben der Elektronik und der Speisung den 8-stelligen DIL-Schalter für die Wahl der Stationsnummer und der Übertragungsgeschwindigkeit.
- Unter dem Hauptprint befindet sich das Busmodul, welches die Stecker für die 4 individuell bestückbaren Ein- und Ausgangsmodule enthält.

3.3.2 Montage, Massbild und Kabelführung

Die PCD1.RIO kann auf die gleiche doppelte DIN-Schiene (2x35 mm) aufgeschraubt werden wie die Busmodule der PCD4 und die PCD1 bzw. die PCD2. Die Anleitung für die Montage, das Massbild und ein Vorschlag für die Kabelführung sind dem Handbuch PCD2, Abschnitt 1.3.2 zu entnehmen.

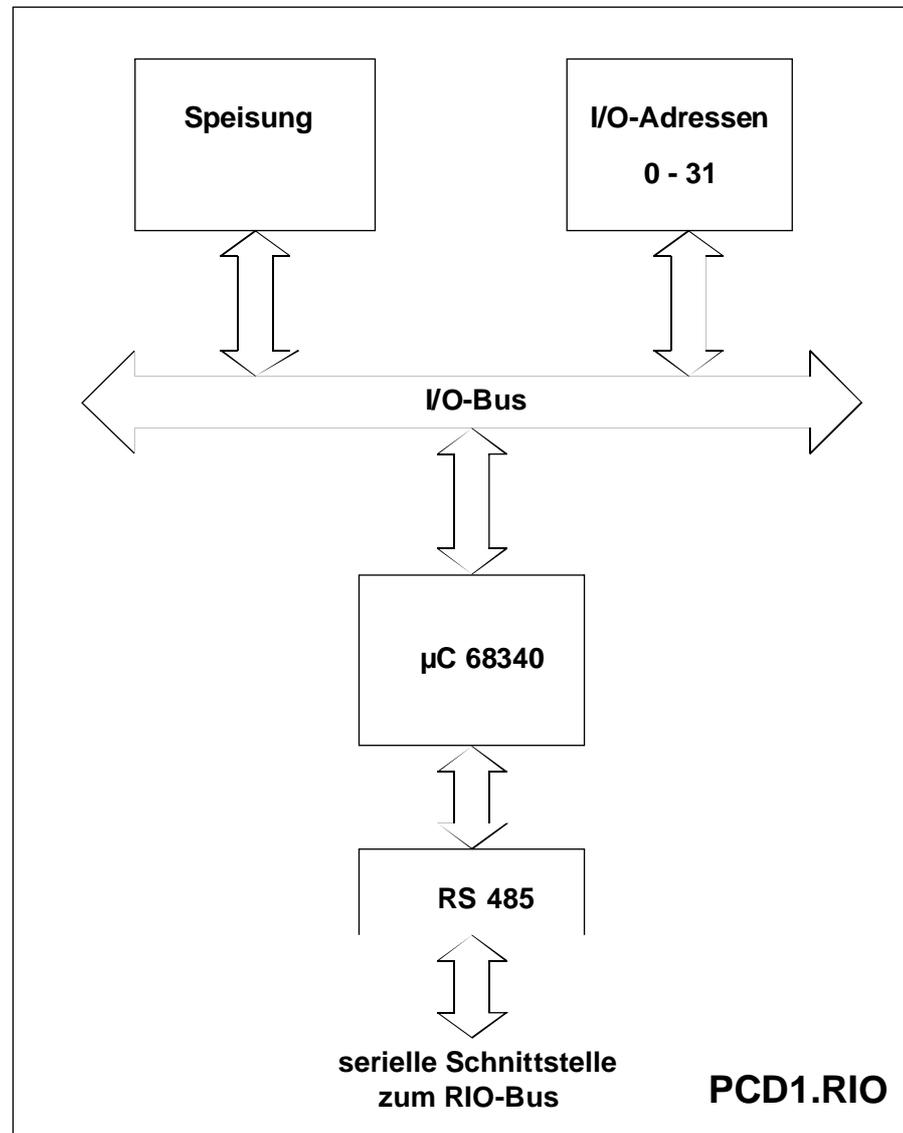
Massbild:



3.4 Architektur und Präsentation des Hauptprints

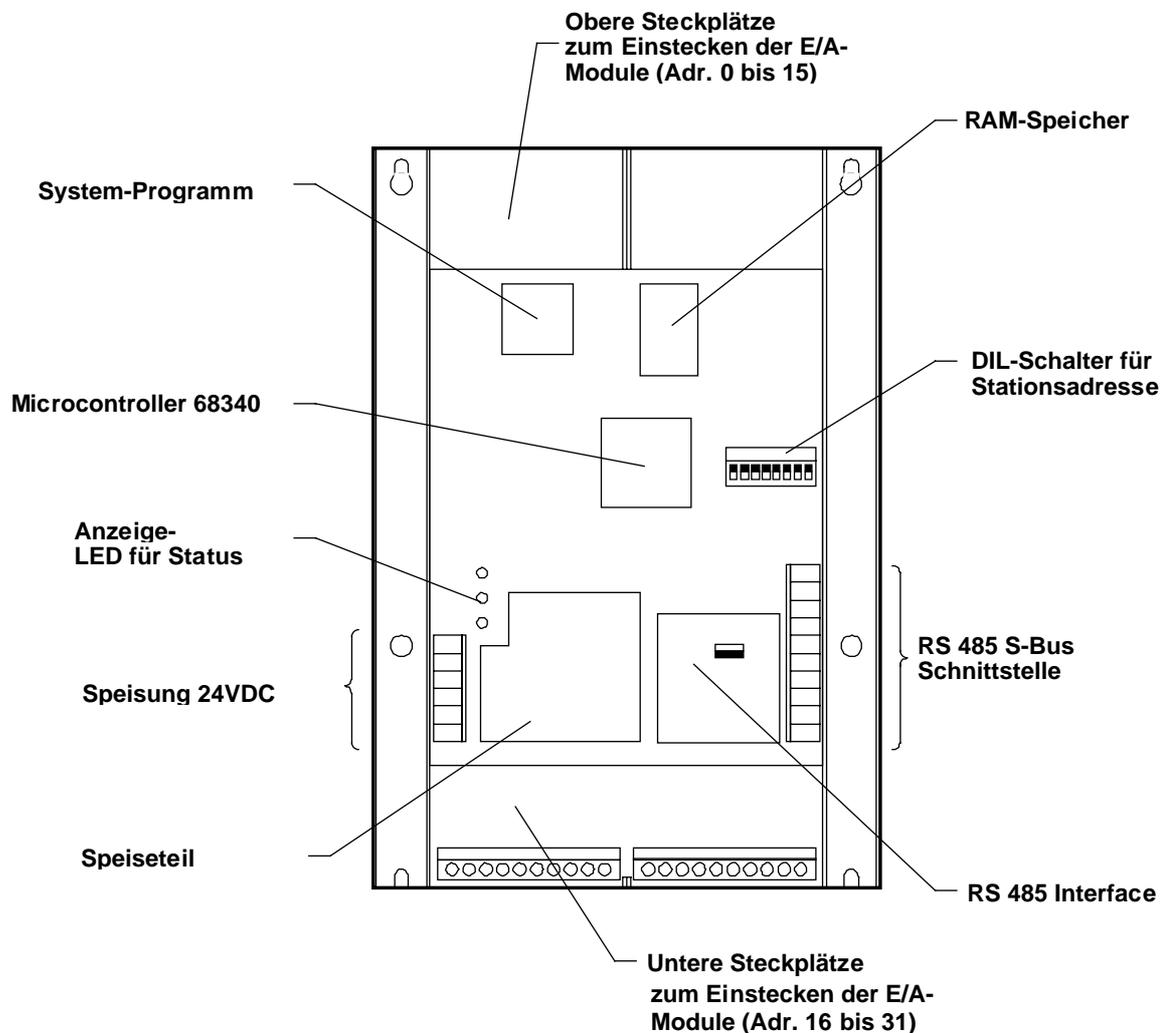
3.4.1 Architektur

Das nachfolgende Schema zeigt die Architektur der PCD1.RIO:



3.4.2 Präsentation des Hauptprints

Die nachfolgende Zeichnung zeigt die PCD1.RIO mit abgenommenem Deckel.

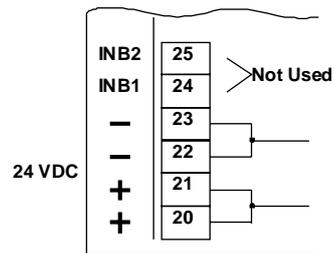


Achtung: Nach dem Abheben des Deckels sind Bauteile berührbar, welche bezüglich elektrostatischen Entladungen empfindlich sind.



Bei eingeschaltetem Gerät dürfen auch keine Manipulationen wie Umstecken von Jumpfern oder Ein-/Ausstecken von E/A-Modulen usw. vorgenommen werden.

Anschluss der Speisung, Sicherung und Störschutz



Der Anschluss erfolgt an den steckbaren Klemmen 20 bis 23 mit Drähten von max. 1.5 mm² (mit Kabelendhülse)

Es kann geglättete oder pulsierende Gleichspannung verwendet werden (siehe technische Daten).

Die LED "24 VDC" zeigt an, wenn die Speisespannung vorhanden ist.

Ein Verpolungsschutz verhindert eine Zerstörung der Schaltung bei falschem Anschliessen.

Der 24V-Schaltkreis ist mit einem PTC-Widerstand versehen, welcher bei Überstrom abschaltet und im Fehlerfall die Schaltung ebenfalls vor Zerstörung schützt.

Sehr wirkungsvolle Störschutzfilter schützen die elektronischen Schaltkreise gegen parasitäre Überspannungen (4 kV gemäss CEI 801-4).

Betriebszustände des Prozessors

Zur Anzeige dienen 3 LED:

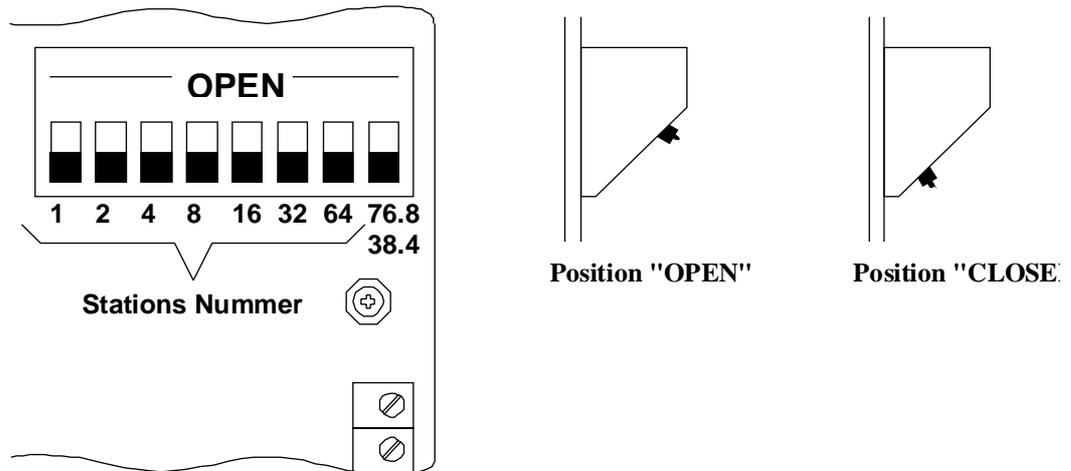
Supply 24 VDC:	LED gelb
Run:	LED gelb
Error:	LED rot

Die LED "24 VDC" zeigt das Vorhandensein der externen Stromversorgung an.

Die eingeschaltete LED "Run" (LED "Error" ausgeschaltet) zeigt, dass die PCD1.RIO korrekt läuft.

Die eingeschaltete LED "Error" zeigt, dass die PCD1.RIO einen Fehler aufweist. Dieser Fall darf in einer Anlage nie auftreten.

DIL-Schalter



Mit dem 8-stelligen DIL-Schalter der PCD1.RIO wird die Stationsnummer und die Übertragungsgeschwindigkeit konfiguriert.

Die Übertragungsgeschwindigkeit kann entweder auf 38.4 oder 76.8 kBaud eingestellt werden. Zur Wahl von 38.4 kBaud wird der Schalter Nr. 8 (ganz rechts aussen) in die Position "CLOSED" (geschlossen) gebracht. Zur Wahl von 76.8 kBaud wird der gleiche Schalter in die Position "OPEN" (geöffnet) geschaltet.

Die Stationsnummer wird mit den 7 ersten Schaltern gewählt. Die Nummerierung erfolgt im binären Format. Es können 128 verschiedene Positionen eingestellt werden (0 bis 127). Die bei jedem Schalter angebrachte Nummer entspricht der Stationsnummer, wenn nur dieser Schalter allein in der Position "OPEN" ist. Wenn sich z.B. von den 7 ersten Schaltern nur der 6. in der Position "OPEN" befindet, ist diese PCD1.RIO als Station Nr. 32 konfiguriert.

Um z.B. die Stationsnummer 93 zu konfigurieren ist die folgende Schalterkombination einzustellen: $93 = 64 + 16 + 8 + 4 + 1$. Es sind also die Schalter 1, 3, 4, 5 und 7 in die Position "OPEN" zu bringen.

Die Konfigurierung ist im spannungslosen Zustand vorzunehmen.

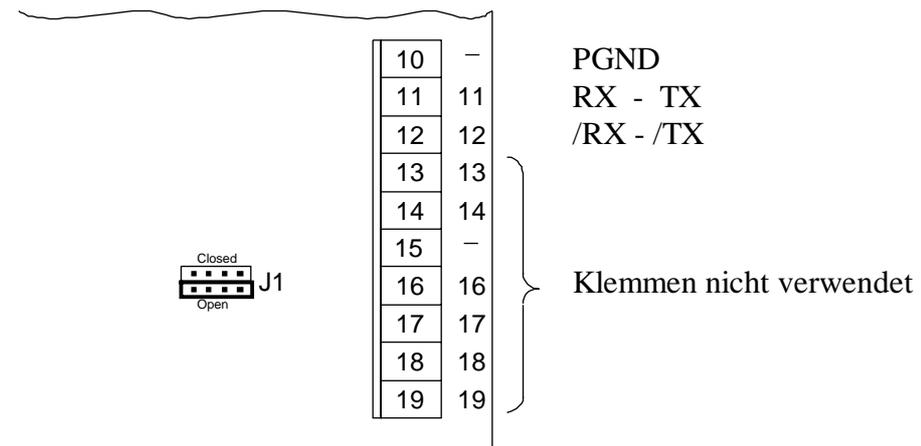


Achtung: Die Stationsnummern 0 und 127 können nicht verwendet werden. Die Adressierung beschränkt sich auf die Stationsnummern 1 bis 126.

3.5 Die RS 485-Schnittstelle

3.5.1 Anschluss der RS 485-Schnittstelle

Die PCD1.RIO ist mit einer Schnittstelle zum Anschluss an das RIO-Netzwerk ausgerüstet. Der S-Bus Slave-Modus ist gemäss den Einstellungen der DIL-Schalter automatisch gewählt.

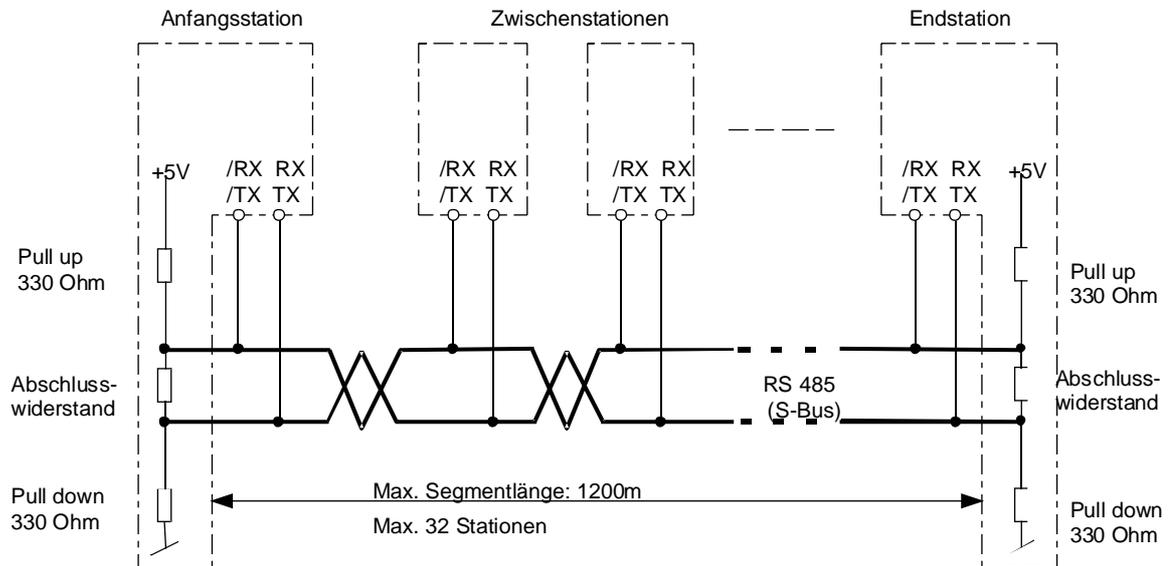


Für die korrekte Einstellung der Abschlusswiderstände (Position des Jumpers J1: "open" → ohne Abschlusswiderstand) und die Auslegung des Buskabels ist das "Handbuch der Baureihe PCD2 - Hardware", das "Handbuch SAIA® S-Bus" und das Handbuch "Installations-Komponenten für RS 485-Netzwerke" zu konsultieren.

Die wichtigsten Punkte sind im folgenden Abschnitt gezeigt. (Auszug aus dem S-Bus Handbuch).

3.5.2 Verlegung der RS 485 Bus-Leitung

Um Störungen zu unterdrücken und Reflexionen zu vermeiden, muss die Bus-Leitung jeweils am **Anfang und Ende** mit den Pullup/Pulldown- und Abschlusswiderständen versehen werden. Diese Widerstände sind in allen Prozessormodulen bzw. Busmodulen eingebaut und können wahlweise eingeschaltet oder angeschlossen werden.



Werden die in den Prozessoren oder Busmodulen eingebauten Abschlusswiderstände verwendet, müssen die Stationen dauernd eingeschaltet sein, ansonst eine Kommunikation über das Netzwerk nicht möglich ist.

Soll das Netzwerk trotz abgeschalteter Anfangs- und/oder Endstation funktionsfähig sein, müssen die Termination Boxen PCD7.T160 eingesetzt werden.

Folgende Punkte sind speziell zu beachten:

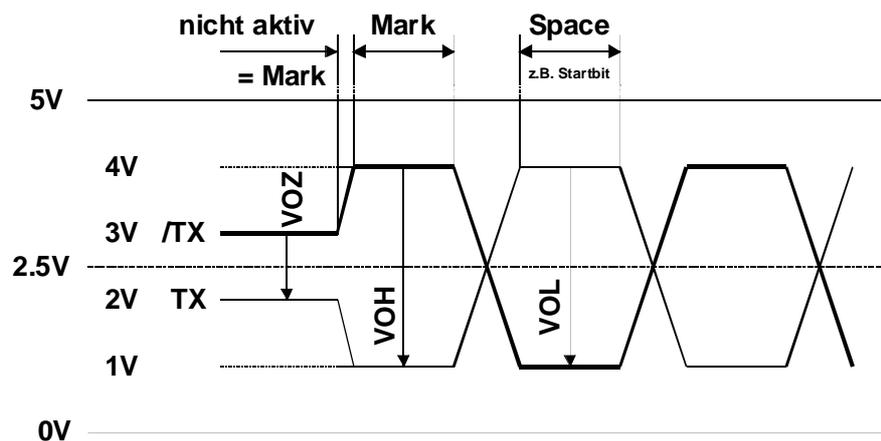
- Beim Anfertigen des Bus-Kabels ist sehr darauf zu achten, dass die Datenleitungen nicht vertauscht werden - also "RX-TX" immer auf "RX-TX" und "/RX-/TX" immer auf "/RX-/TX". Die Bezeichnungen "RX-TX" und "/RX-/TX" sind bei Fremdgeräten oft anders bezeichnet:

RX	D	-RX
/RX	/D	+RX
TX	D	-TX
/TX	/D	+TX

- Es ist weiter darauf zu achten, dass die Bus-Leitung auch dann durchgehend verbunden bleibt, wenn ein oder mehrere Stecker ausgezogen werden.
- Die Stichleitungen dürfen eine Länge von 0.5m nicht übersteigen.
- Als Kabel ist Litze von min. 0,5mm², 2-adrig verdreht und abgeschirmt einzusetzen.

Signalpegel der RS 485 Schnittstellen

Signaltyp	Logischer Zustand	Polarität
Datensignal	0 (space) 1 (mark)	RX-TX positiv zu /RX-/TX /RX-/TX positiv zu RX-TX

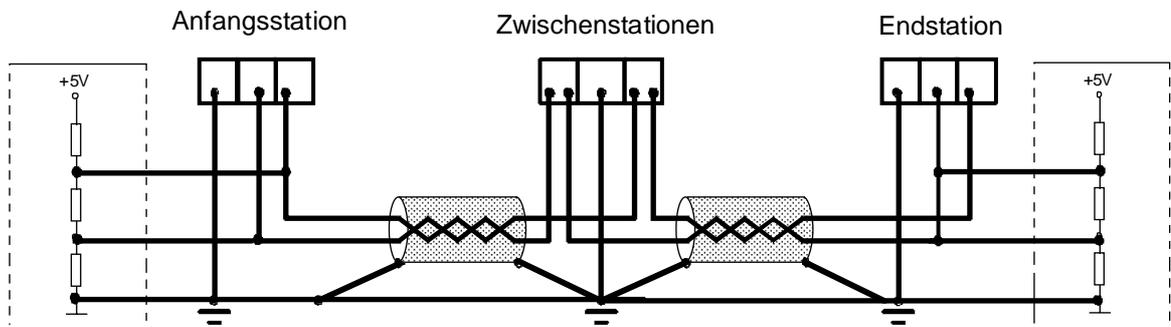


VOZ = 0,9V min... 1,7V max (nicht aktiv) *)
 VOH = 2V min (mit Last) ... 5V max (ohne Last)
 VOL = -2V ... -5V

*) abhängig von den "Pull up"-, "Termination"- und "Pull-down"-Widerständen.

Erdung einer RS 485 Bus-Leitung

Beide Enden der Kabelabschirmung sind immer an eine zuverlässige Erdleitung anzuschliessen. So können Potentialdifferenzen minimal gehalten werden.

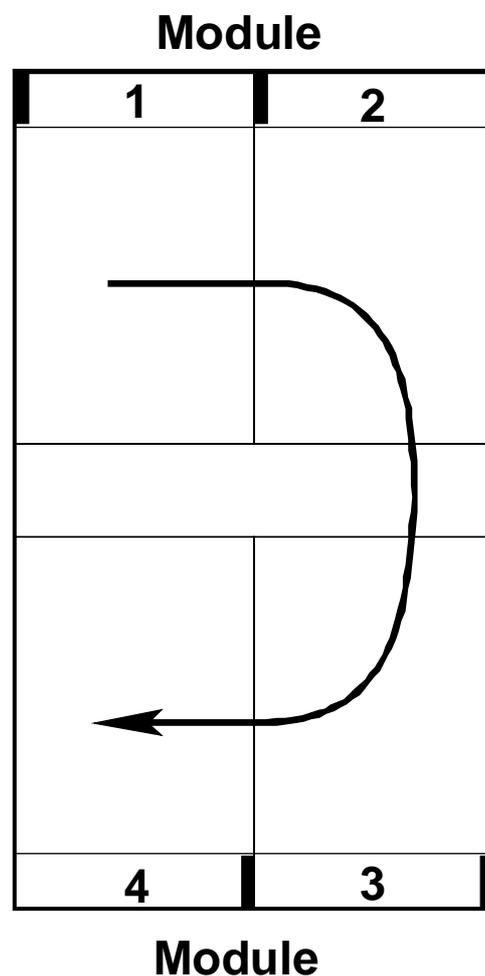


Es wird empfohlen, das RS 485 Kabel nicht in unmittelbarer Nähe von Starkstromkabeln (Motoren) zu verlegen, ansonsten störende Interferenzen auftreten können, es sei denn, die Motorenkabel seien abgeschirmt.

3.6 Adressierung der Ein- und Ausgangsmodule

Bei der PCD1.RIO werden die **Positionen** der Module und nicht, wie bei der PCD1 oder PCD2, die Adressen der einzelnen Ein- und Ausgänge bezeichnet. Die 4 Steckplätze sind mit "Module 1", "Module 2", "Module 3" und "Module 4" betitelt und werden bei der Programmierung bzw. Konfigurierung auch so adressiert.

Jedem belegten Steckplatz muss ein Modultyp der PCD2-Serie zugewiesen werden., z.B. A400. Diese Information wird via den Konfigurator der Master-Station des RIO-Netzwerks übermittelt (siehe Kapitel 4 und 5).



4. Konzept des RIO-Netzwerks

4.1 Allgemeine Beschreibung

Das Prinzip des PCD-RIO beruht darauf, dass alle Medien der Slave-Stationen (Status der digitalen Ein- und Ausgänge und Werte der analogen Ein- und Ausgänge) in der Master-Station abgebildet und zu den Slave-Stationen wieder übertragen werden. Das Anwenderprogramm in der Master-Station wird dann diese Daten wie lokale Medien behandeln. Die Aufgabe des RIO-Prozessors besteht nun darin, die Slaves im Hintergrund zyklisch abzufragen um das Abbild der Medien aufzufrischen und nach der Behandlung durch das Anwenderprogramm wieder zu den Slaves zu übertragen. Die Definitionen des Datenaustausches zwischen dem Master und den Slaves sind in Datenblocks (DB) der Master-Station abgelegt.

Als Master-Station kann eine PCD4.M445, eine PCD4.Mxx5 oder eine PCD2.M1xx eingesetzt werden. Die Liste mit den zu den Slaves zyklisch auszutauschenden Meldungen liegt in der Konfigurationsdatei "RIOCFG.SRC". Die Auffrischung des Medien-Abbildes erfolgt für den Anwender unsichtbar im Hintergrund (in der Firmware).

Eine PCD4.M445 als Master-Station bietet in zeitkritischen Anlagen den Vorteil gegenüber einer PCD4.Mxx5 oder einer PCD2, dass das eigentliche Anwenderprogramm für den Prozess und das Anwenderprogramm für den Datenaustausch im RIO-Netzwerk in je einer eigenen CPU untergebracht werden kann und so der zu steuernde Prozess von der RIO-Prozedur nicht verlangsamt wird. Mit CPU 0 werden in diesem Fall der Prozess und mit CPU 1 alle RIO-Belange bearbeitet.

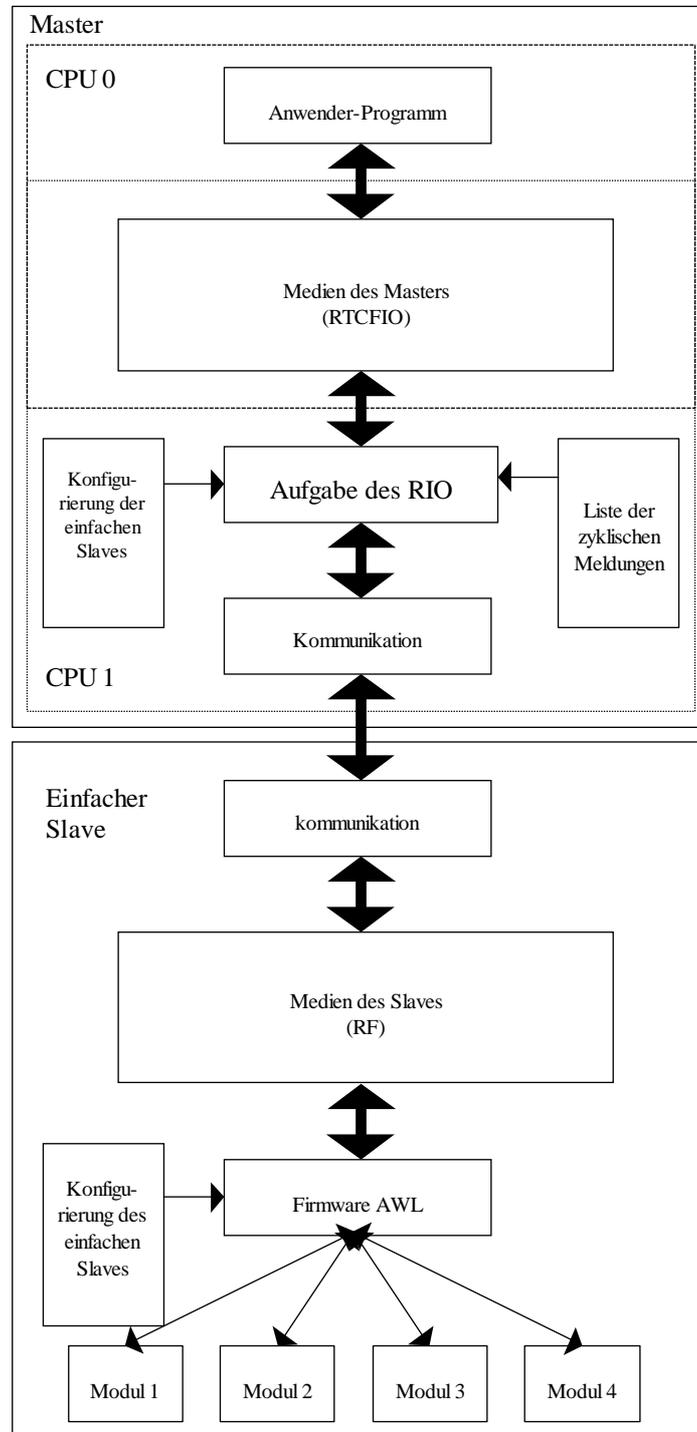
Mit Hilfe des Konfigurators (ausführliche Beschreibung in den Kapiteln 4 und 5) werden einige Dateien erzeugt. Die erste Datei, "RIO.DEF" (oder ein frei gewählter Name xxx.DEF) enthält alle Definitionen der Elemente der Slave-Stationen (EQU), die 2. Datei, "RIOCFG.SRC" (oder ein frei gewählter Name xxx.SRC) enthält die Konfiguration des gesamten Netzwerkes.

Die Datei "RIOCFG.SRC" wird schliesslich zum Anwenderprogramm, welches die Kommunikationsbeziehungen des RIO-Netzwerks beinhaltet, gelinkt. Im Falle einer PCD4.M445 zum Anwenderprogramm in der CPU1, bei den anderen Fällen zum gesamten Anwenderprogramm.

Alle Anwenderprogramme, welche etwas mit dem RIO-Netzwerk zu tun haben, müssen das Programm "RIO.DEF" mit \$INCLUDE einbinden.

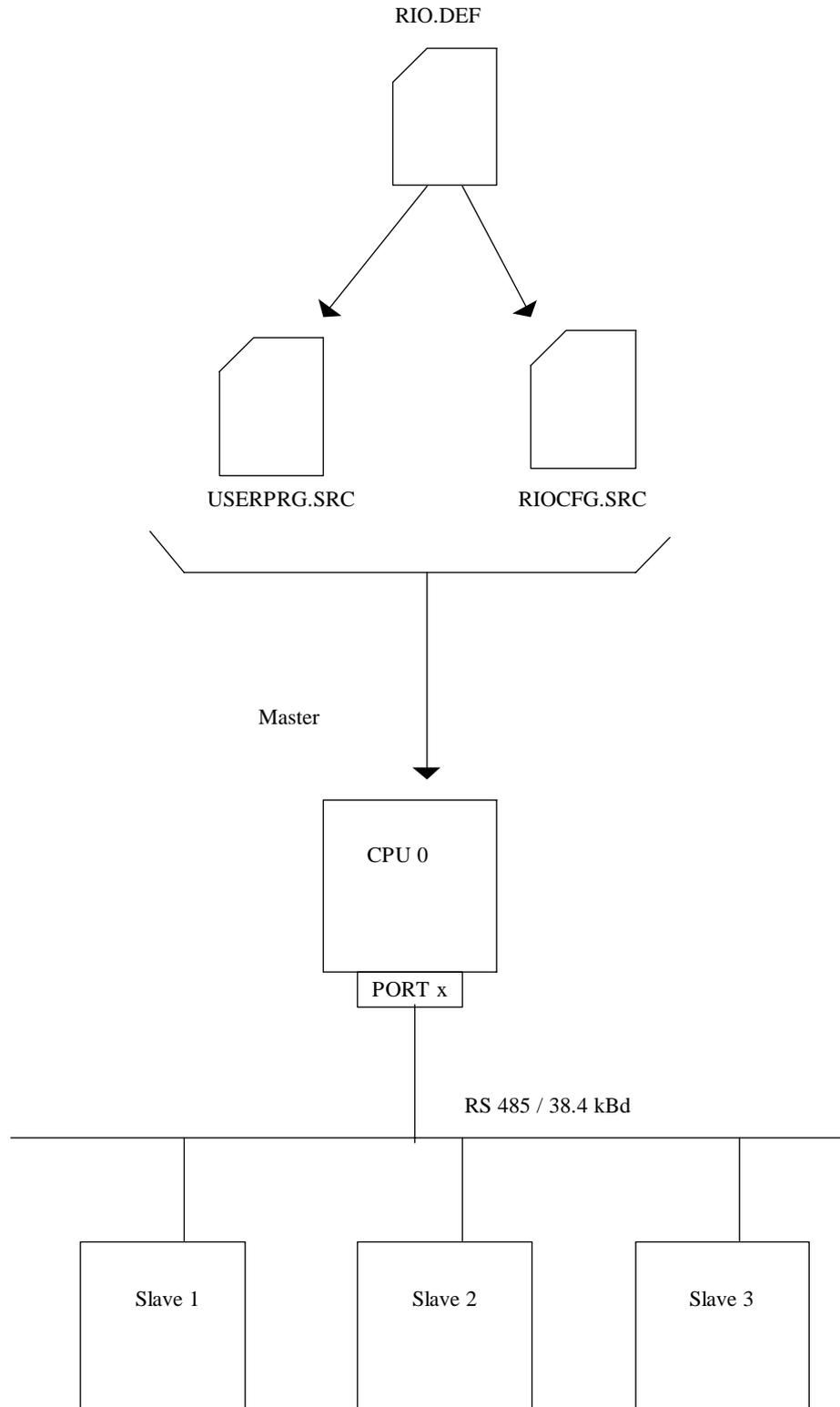
Die nicht-intelligenten PCD1.RIO-Stationen sind für den Anwender nicht programmierbar.

Die nachstehend gezeigte Figur zeigt die Funktion des PCD-RIO mit einer Multiprozessor-CPU (PCD4.445) als Master-Station und einem einzigen PCD1.RIO als Slave.

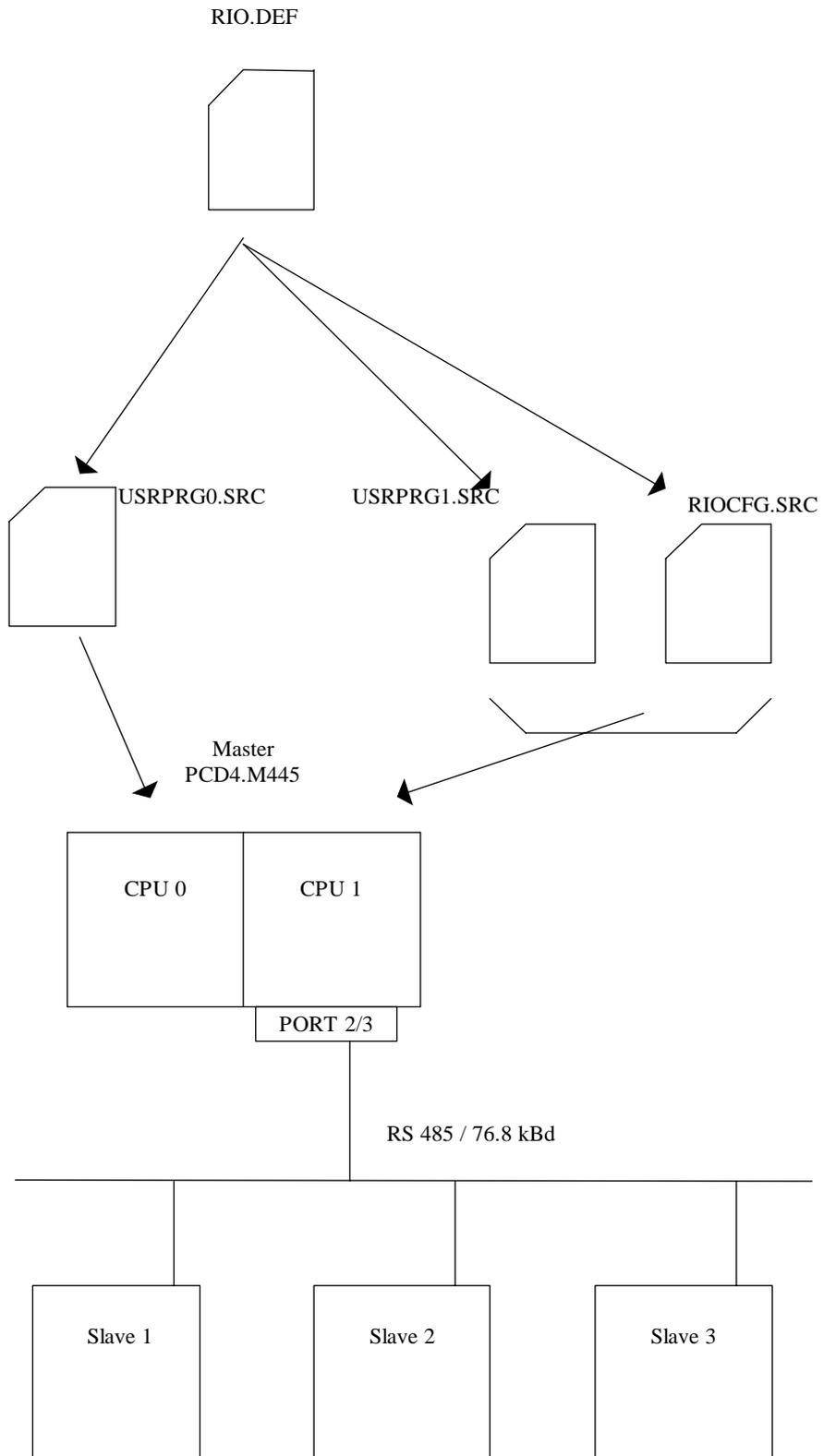


4.2 Dateien für die RIO-Aufgabe

Die nachfolgende Skizze zeigt die verschiedenen Dateien, welche im Fall der Verwendung einer PCD2 oder einer PCD4.Mxx5 als Master-PCD (Singel-Prozessor-CPU) zu linken und zu laden sind.



Die nachfolgende Skizze zeigt die verschiedenen Dateien, welche im Fall der Verwendung einer PCD4.M445 als Master-PCD (Doppel-Prozessor-CPU) zu linken und zu laden sind.



Die Dateien für die RIO-Belange

- **RIO.DEF** (xxx.DEF)

Diese Datei enthält alle Deklarationen (EQU) aller Elemente der Slave-Stationen, welche der Master für die Erfüllung der RIO-Aufgabe braucht. Diese Datei wird schliesslich mit der Direktive \$INCLUDE ins Anwenderprogramm, welches die Kommunikationsbeziehungen behandelt, eingebunden.

Der Anwender muss alle Elemente aller Slave-Stationen (digitale und analoge Ein- und Ausgänge) mit einem symbolischen Namen definieren. Eine absolute Adressierung ist in einem PCD-RIO-Netzwerk nicht möglich. Diese Symbole werden im Anwenderprogramm und in der Konfigurationsdatei "RIOCFG.SRC" verwendet.

Diese Datei wird im RIO-Konfigurator automatisch erzeugt.

- **RIOCFG.SRC** (xxx.SRC)

Diese Datei enthält die Konfiguration der Slaves und die Definitionen, welche für die zyklische Auffrischung des Abbildes der Medien in der Master-Station notwendig sind. Die Datei enthält auch den Assignierungstext für den SASI-Befehl. Die Datei ist zum Anwenderprogramm derjenigen CPU, welche das Programm für die RIO-Belange enthält, zu linken.

Diese Datei wird ebenfalls im RIO-Konfigurator automatisch erzeugt.

- **USRPRG.SRC** (yyy.SRC)

Dies ist das Anwenderprogramm, welches, wie der Name sagt, vom Anwender erstellt wird. Das Anwenderprogramm enthält sowohl die Kommunikationsbeziehungen der RIO-Aufgabe als auch die Steuerung des Prozesses. Der Programmteil mit den Kommunikationsbeziehungen muss in IL (Instruction List) editiert werden. Der Programmteil für die Prozesssteuerung kann im PG3 oder PG4 geschrieben werden. In diesem Fall muss die RIO-Aufgabe im Projekt-Manager des PG4 gelinkt werden.

4.3 Aktivierung der RIO-Aufgabe

Die Aktivierung der RIO-Aufgabe erfolgt, wie alle Kommunikationsaufgaben bei der SAIA® PCD, mit dem SASI-Befehl. Der dazu gehörige SASI-Text ist der folgende:

```
"UART:baudspd[,timeout];"  
"MODE:RM1,Rxxxx,DBxxxx;"  
"CONF:DByyyy;"  
"DIAG:Fzzzz,Rzzzz[,DBzzzz]"
```

Die "UART"-Assignierung ist gleich wie für den Standard-S-Bus. Das Timeout und die Ts- und Tn-Verzögerungen können optional eingegeben werden.

Die bei "MODE" angegebene DB-Nummer gibt die Adresse des 1. einer Reihe von DBs an, welche die Liste der zyklischen Telegramme enthalten. (Rxxx dient der Wahl der S-Bus Station, wenn Telegramme asynchron ausgetauscht werden sollen, wie bei SASI S-Bus).

Die bei "CONF" angegebene DB-Nummer gibt die Adresse des 1. einer Reihe von DBs an, welche die Konfiguration der einfachen (nicht-intelligenten) Slaves enthalten. Ist diese Rubrik nicht definiert heisst dies, dass im Netzwerk keine einfachen Slaves vorhanden sind. In diesem Fall beginnt der Prozessor sofort mit dem Senden der zyklischen Telegramme.

Bei "DIAG" wird die Basisadresse von 24 Diagnose-Flags sowie die Basisadresse von 4 Diagnose-Register definiert. Es kann auch die Basisadresse von 2 Diagnose-DB angegeben werden. Diese beiden DB enthalten statistische Angaben zu jeder Station im Netzwerk. Wird keine Diagnose-DB Adresse angegeben, wird keine Statistik erstellt. Die Adressen der beiden Diagnose-DB muss gleich oder grösser 4000 sein. Näheres dazu in Abschnitt 4.4.

Dieser SASI-Text wird im RIO-Konfigurator automatisch erzeugt.

Nach der Abarbeitung des SASI-Befehls ist die RIO-Schnittstelle assigniert und die einfachen Slave-Stationen sind konfiguriert. Nach der Konfigurierungsphase werden die zyklischen Telegramme dauernd gesandt, dies auch, wenn die CPU in STOP, COND.-RUN oder TRACE ist.

Die Assignierung im Modus "RM1" erlaubt auch das Senden von azyklischen Telegrammen mit den Befehlen STXM und SRXM (siehe Abschnitt 4.5).

Bevor die RIO-Kommunikation aktiviert wird, kann der Anwender diese (RIO-) Schnittstelle im konventionellen S-Bus-Modus assignieren um mit den Befehlen STXM und SRXM gewisse Voreinstellungen durchzuführen.

4.4 Diagnose-Möglichkeiten

Die vielfältigen Diagnose-Möglichkeiten sind im Kapitel 7. zusammengefasst und an einem praktischen Beispiel erläutert.

Die Diagnose kann in die folgenden Gruppen unterteilt werden:

- Überwachung der Anwesenheit der Masterstation
- Überwachung der Anwesenheit der Slavestationen (XOB 30)
- Diagnose-Flags für die Grobdiagnose der Kommunikation
- Diagnose-Register für die Feindiagnose der Kommunikation
- Diagnose Data-Blocks für Überwachung und Statistik der RIO-Aufgabe
- Data-Blocks für die Konfiguration der Stationen
- Data-Blocks für die Konfiguration der zyklischen Telegramme

4.5 Verwendung des RIO-Kanals für asynchrone Telegrammübertragungen

Der RIO-Kommunikationskanal kann mittels des Anwenderprogramms auch zum Übertragen von S-Bus Telegrammen mit den Befehlen STXM und SRXM verwendet werden. Diese Telegramme für die asynchrone Übertragung von Informationen an intelligente Slaves erfolgen zwischen zwei zyklischen Meldungen.

Sobald die Schnittstelle für die Kommunikation im Modus "RM1" assigniert ist, können asynchrone Telegramme im normalen S-Bus-Modus übertragen werden. Die Überlagerung der beiden Kommunikationsmodi ist für den Anwender nicht sichtbar. Im Beispiel "TEST2", Aufgabe Punkt 9), im Abschnitt 6.2, wird diese Technik andeutungsweise gezeigt.

Die Übertragung von azyklischen Telegrammen durch das Anwenderprogramm verlangsamt die Auffrischung des Medien-Abbildes.

4.6 Auffrisch- und Reaktionszeiten

Die Reaktionszeit (T_r) hängt von 4 verschiedenen Faktoren ab:

- Verzögerung durch das Eingangfilter (T_{dio})
- Auffrischzeit (T_{rfr})
- Programmzykluszeit (T_{plc})
- Von der Umschaltzeit der Ausgänge abhängige Verzögerung (T_{dio})

Die Verzögerung, welche vom Eingangfilter herrührt und die Umschaltzeit der Ausgänge seinen in einer ersten Betrachtung vernachlässigt.

Die Auffrischzeit " T_{rfr} " ist die Zeit zum Übertragen aller Meldungen an die Slavestationen oder, anders ausgedrückt, die Zeit die zwischen 2 gleichen Telegrammen verstreicht.

Die Zykluszeit des PCD-Anwenderprogramms " T_{plc} " ist die notwendige Zeit um alle Befehle des Anwenderprogramms einmal abzuarbeiten.

Die Reaktionszeit einer Installation wird die folgende sein:

$$T_{plc} < \text{Reaktionszeit der Installation} < (2 * T_{rfr} + 2 * T_{plc})$$

Die nachfolgende Tabelle zeigt die Ausffrischzeit (T_{rfr}) für verschiedene PCD-Systeme bei unterschiedlicher Anzahl von zu übertragenden Medien.

System	PCD2	PCD4.Mxx5	PCD4.M445	
			CPU0	CPU1
kBaud	38,4	38,4	38,4	76,8
8 Flags	7,3	8,3	7,5	3,8
16 Flags	7,7	8,7	8,0	4,0
32 Flags	8,3	9,2	8,5	4,2
2 Reg.	9,2	10,4	9,6	4,6
16 Reg.	26,6	27,6	27,0	12,8
32 Reg.	46,8	47,8	47,0	22,6

4.7 Erforderliche Firmware

Um die mit der RIO-Kommunikation gebotenen Möglichkeiten ausnützen zu können, muss eine Firmware eingesetzt werden, welche die RIO-Ausgaben bewältigen kann.

Die ersten Firmware-Versionen sind:

- Version \$41 für PCD2.M1xx
- Version \$C3 für PCD4.Mxx5 (CPU 0)
- Version \$C3 für PCD4.M445 (CPU 1)

5. Der RIO-Konfigurator PCD8.C3..

5.1 Einführung

Der Konfigurator ist ein unentbehrliches Werkzeug für den erfolgreichen Aufbau einer PCD-RIO Aufgabe. Wie der Name sagt, wird mit diesem Werkzeug die ganze RIO-Aufgabe konfiguriert:

- Konfigurierung des Netzwerks mit einem Master und einer Anzahl nicht-intelligenten und ev. intelligenten Slaves
- Definition der PCD-Typen jeder RIO-Station
- Definition der Stationsnummern
- Definition der Übertragungsgeschwindigkeit
- Definition der Bestückung jeder einzelnen unintelligenten Slavestation
- Definition der "Messages" für den Datenaustausch mit intelligenten Slaves
- Definition der Basisadressen aller Diagnose- und Assignierungselemente
- Erstellung der RIO.DEF-Datei mit allen EQU-Deklarationen für das Medienabbild in der Master-Station
- Erstellung der SLAVExx.DEF-Datei mit den EQU-Deklarationen für jede einzelne intelligente Slave-Station
- Erstellung RIOCFG.SRC-Datei mit der Konfiguration der Slaves
- Bereitstellung des SASI-Textes für die Assignierung der Aufgabe

Der Konfigurator läuft unter WINDOWS 3.1 bzw. WINDOWS 95.

Die vorliegende 1. Version dieses RIO-Konfigurators ist eine "stand alone" Ausführung, d.h. der Konfigurator ist weder im PG3 noch im PG4 eingebunden. Die Verwaltung der umfangreichen Ressourcen betrifft also nur den Konfigurator allein. Es liegt in der Verantwortung des Programmierers, Konflikte innerhalb der RIO-Aufgabe und mit andern Programmteilen zu verhindern. Es ist das Ziel, eine spätere Version des RIO-Konfigurators ins PG4 einzubauen und alle Ressourcen gemeinsam zu verwalten.

Die vorliegende Version enthält ein umfangreiches "Help" in der WINDOWS-Philosophie in englischer Sprache. Alle Punkte sind im vorliegenden Handbuch in deutsch erläutert.

5.2 Installation des RIO-Konfigurators

Für die vorliegende Version ist noch kein Installationsmenü vorhanden, d.h. die Installation muss "von Hand" erfolgen. Der Konfigurator wird auf 2 Disketten geliefert. Der Konfigurator belegt ca. 4 MBytes auf der Harddisk.

Für die Installation ist folgendermassen vorzugehen:

Installation auf C:\SNET

1. Diskette 1 in Laufwerk A: einlegen und aus dem DOS die Datei SNET.INI in einen ASCII-Editor laden. Pfad und Verzeichnis angeben bzw. anpassen:

```
DeviceResDir=C:\SNET\DEVRES  
RIOLibraryFile=C:\SNET\RIODEV.DAT
```

(oder entsprechend der gewünschten Installation)

2. Die beiden Dateien CTL3DV2.DLL und SNET.INI ins Verzeichnis \WINDOWS kopieren.
3. Dekomprimieren der Installationsdateien mit:

```
PKUNZIP -d SNET.ZIP C:\SNET
```

Gemäss den Weisungen am Bildschirm die entsprechenden Disketten einsetzen.

4. Die WINDOWS Programmgruppe, wo der RIO-Konfigurator zu liegen kommen soll, anwählen.

Mit 'Datei' - 'Neu...' - 'Programm' - 'OK'

Bei 'Beschreibung' Titel des RIO-Konfigurators,

z.B. "PCD-RIO"

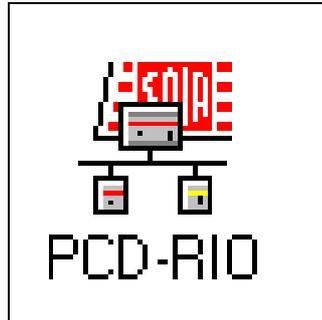
eingeben. Bei Befehlszeile

```
C:\SNET\SNET.EXE
```

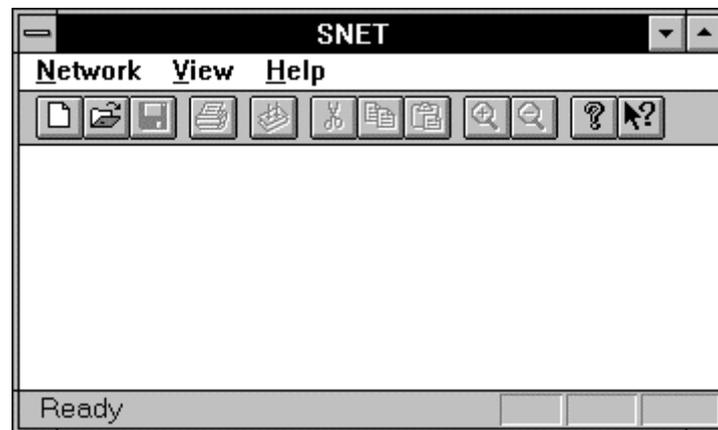
eingeben (oder 'Durchsuchen...') und 'OK'.

Die Ikone für den RIO-Konfigurator erscheint. Die Installation ist abgeschlossen.

5.3 Aufruf und Beschreibung des RIO-Konfigurators



Nach dem Doppelklicken auf die "PCD-RIO"-Ikone erscheint das Hauptmenü des RIO-Konfigurators:

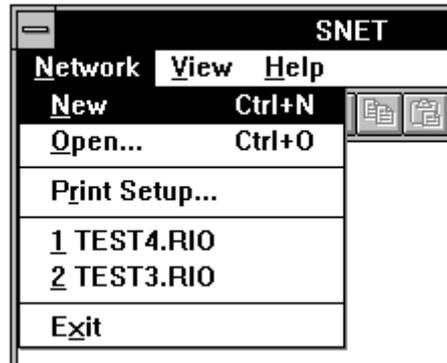


Es werden in der Folge alle Menüs und Untermenüs der Reihe nach, wie diese sich diese auf dem Bildschirm präsentieren, kurz beschrieben.

Eine Beschreibung der Menüs, wie diese in der Praxis zur Anwendung kommen, wird im nächsten Kapitel 6: "Programmierung" gezeigt.

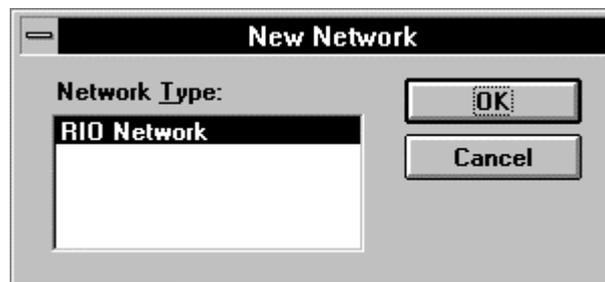
Die Funktionsleiste (Toolbar) ist am Schluss dieses Kapitels, im Unterabschnitt 3.3.10 beschrieben. Die einzelnen Funktionstasten werden bei den entsprechenden Menüpunkten gezeigt.

5.3.1 Das Untermenü 'Network' (Netzwerk)



'New':

Nach dem Anklicken erscheint das folgende Fenster:

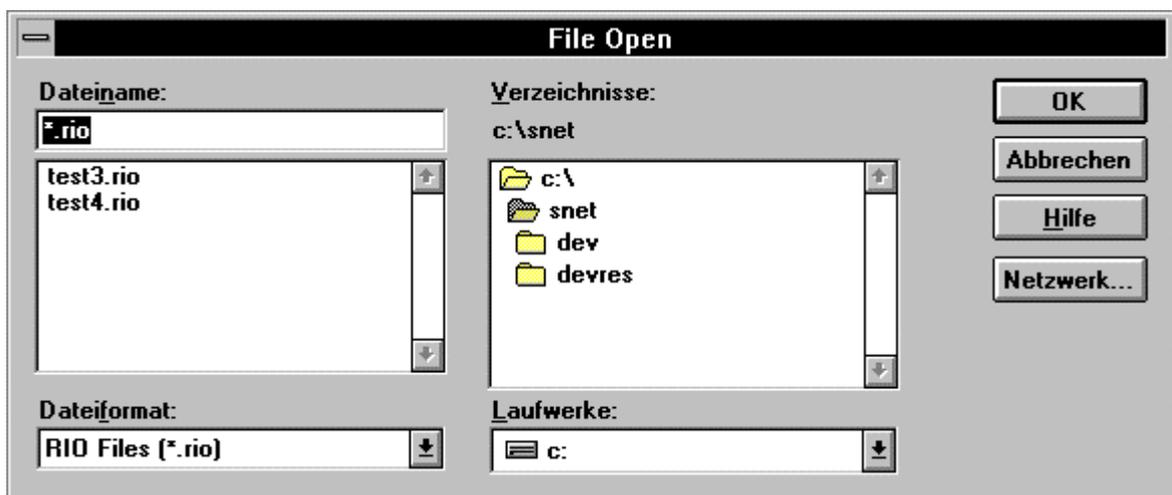


Es kann heute "nur" ein RIO-Netzwerk konfiguriert werden. In einer späteren Version wird möglicherweise auch ein PROFIBUS DP Netzwerk zur Auswahl bereit stehen.

Nach 'OK' beginnt die eigentliche Konfigurierung. Diese erfolgt im folgenden Abschnitt.

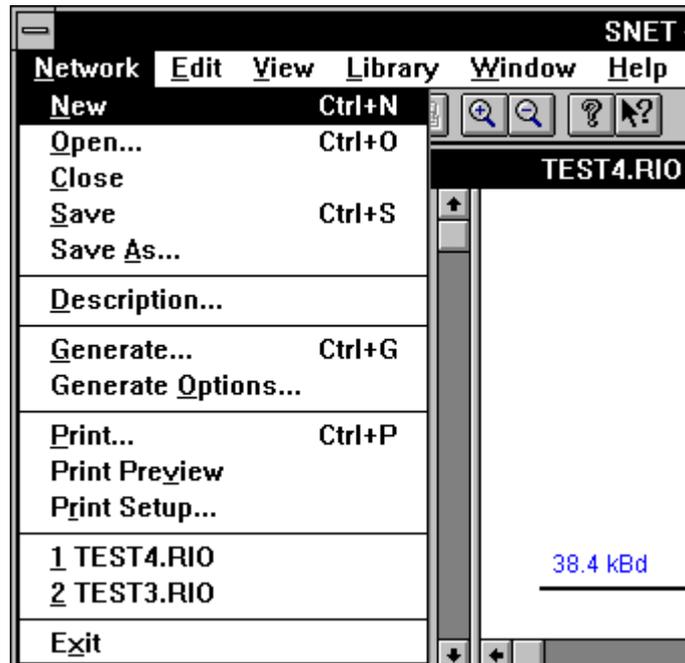
'Open...':

Wurden bereits Konfigurationen durchgeführt und abgespeichert, erscheint das folgende Fenster:



Die bereits erstellten Konfigurationen können wieder hervorgeholt werden.

Nachdem eine Konfigurierung in Arbeit ist, präsentiert sich das Untermenü 'Network' wie folgt:



'Close': Zum Schliessen des aktiven Netzwerk-Dokuments.

'Save': Zum Speichern des Netzwerk-Dokuments, wenn dieses bereits einen Namen hat



'Save as...': Zum erstmaligen Speichern oder zum Speichern unter einem andern Namen.

'Description': Feld für eine Beschreibung des Netzwerks.



'Generate': Erstellt die Dateien RIO.DEF, SLAVExx.DEF und RIOCFG.SRC am Ende der Konfigurierung. (später)



'Generate Options...':

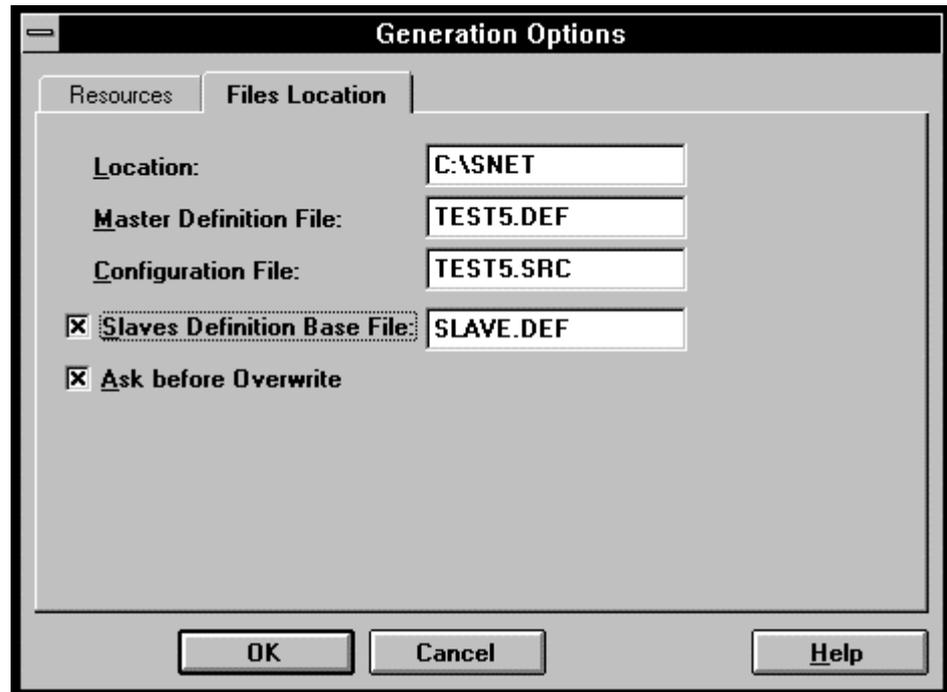
Es werden hier die Basisadressen und der symbolische Name der Basiselemente der verschiedenen Konfigurations- und Diagnosemedien definiert. Eine Standard-Einstellung ist vorgegeben und kann übernommen werden.

	Address:	Symbol:	
First Configuration DB:	10	db_cfg	
First Messages DB:	32	db_msg	
First Diagnostic DB:	4500	db_diag	<input checked="" type="checkbox"/> Use DB diags
First Diagnostic Flag:	500	f_diag	
First Diagnostic Register:	500	r_diag	
SASI Text Number:	100	txt_rio	
Partner Stn Nb Register:	100	r_dst	

'Resource': Die Bedeutung der einzelnen Mediengruppen wird später erläutert.

Es sei auch hier wieder darauf hingewiesen, dass der Programmierer darauf zu achten hat, dass keine Konflikte zwischen den Mediengruppen entstehen.

'Files Location': Es werden hier die Namen und das Verzeichnis der RIO-Dateien spezifiziert.



Location: Name des Verzeichnisses, in welchem die Dateien der RIO-Aufgabe abgelegt werden.

Master Definition File: Name der Definitionsdatei mit den EQU-Deklarationen für das Medienabbild in der Master-Station. Diese Datei wird in alle Dateien, welche die RIO-Aufgabe betreffen, mit "\$include" eingebunden.

Configuration File: Name der Konfigurationsdatei mit den Konfiguration der Slaves. Diese Datei wird mit dem Anwenderprogramm gelinkt.

Slaves Definition Base File: Diese Datei ist optional. Die Datei enthält die EQU-Deklarationen für jede einzelne intelligente Slave-Station. Diese Datei wird in alle Dateien, welche die intelligenten Slave-Stationen der RIO-Aufgabe betreffen, mit "\$include" eingebunden. Der Stationsname erscheint im Dateinamen, z.B. für Station 15: "SLAVE15.DEF".

Ask before Overwrite: Wenn eingeschaltet, wird vor dem Überschreiben einer Datei rückgefragt.

'Print': Ausdrucken des Netzwerks und der Konfiguration.
(siehe ...)



'Print Preview...':

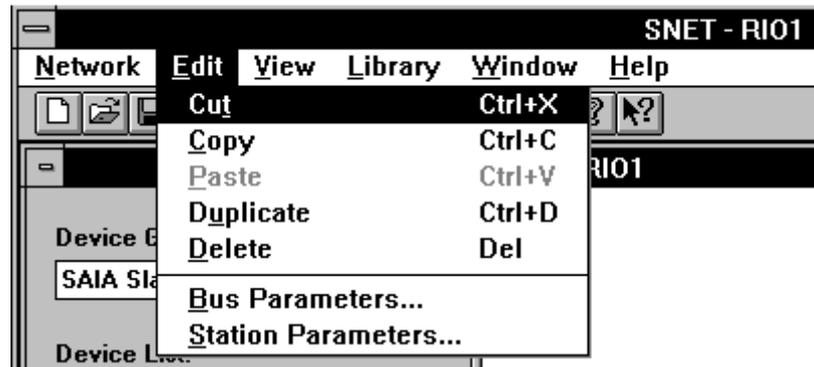
Zeigt eine Vorschau auf den Ausdruck

'Print Setup...': Konfigurierung des Druckers

'Recent Files': Zeigt die letzten 4 Konfigurations-Dateien (falls bereits vorhanden).

'Exit': Verlassen des RIO-Konfigurators.

5.3.2 Das Untermenü 'Edit'



'Cut': 'Ausschneiden'



'Copy': 'Kopieren'



'Paste': 'Einfügen'

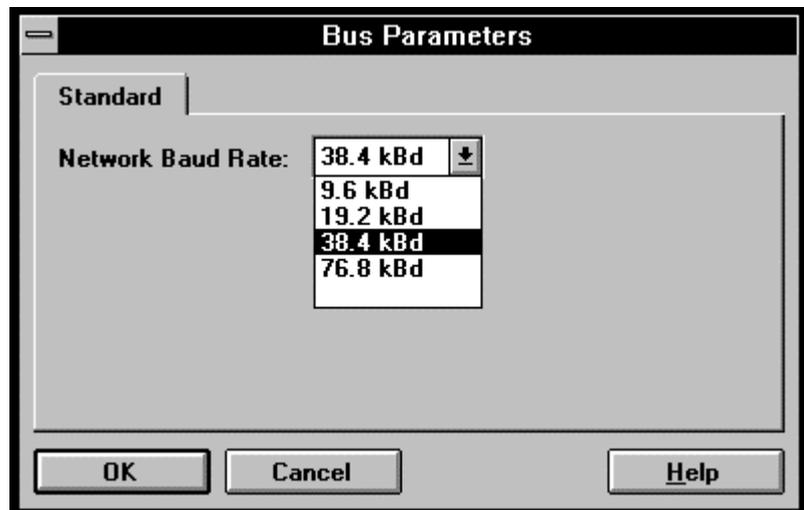


'Duplicate': 'Duplizieren'

'Delete': 'Löschen'

'Bus Parameters': 'Standard':
 Die Standard-Baudrate ist 38.4 kBd. Es kann eine andere Baudrate gewählt werden. Wird eine zu hohe Baudrate gewählt, wird nach einer Meldung automatisch die höchst zulässige Baudrate eingestellt.

Standard
 WINDOWS-
 Funktionen

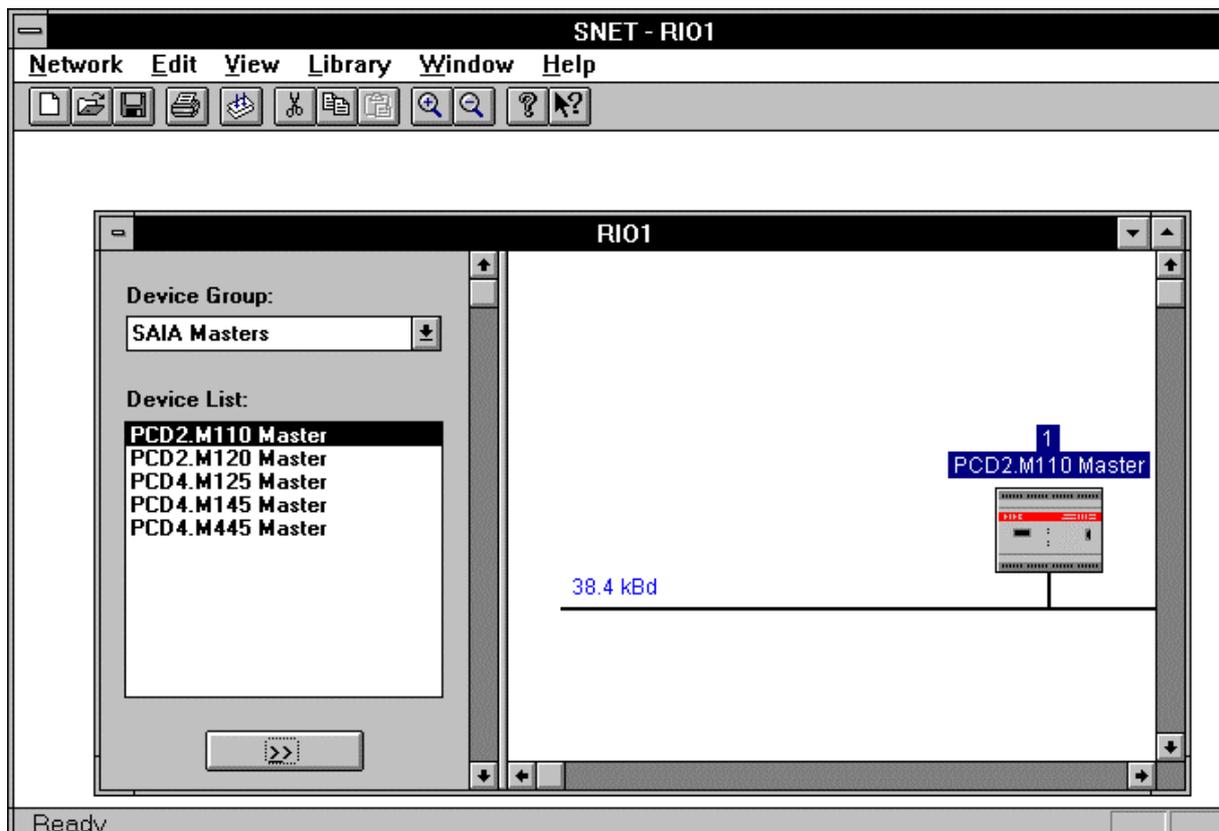


Das vorangehende Fenster kann auch durch einen Doppelklick auf die Bus-Leitung geöffnet werden (siehe nächstes Bild)

Nachdem aus der Liste der möglichen Master-Stationen ein PCD-Typ ausgewählt, mit der Betätigung der Taste



ins Konfigurationsfeld eingefügt wurde, können die Stations-Parameter für die Master-Station definiert werden.



'Station Parameters...':

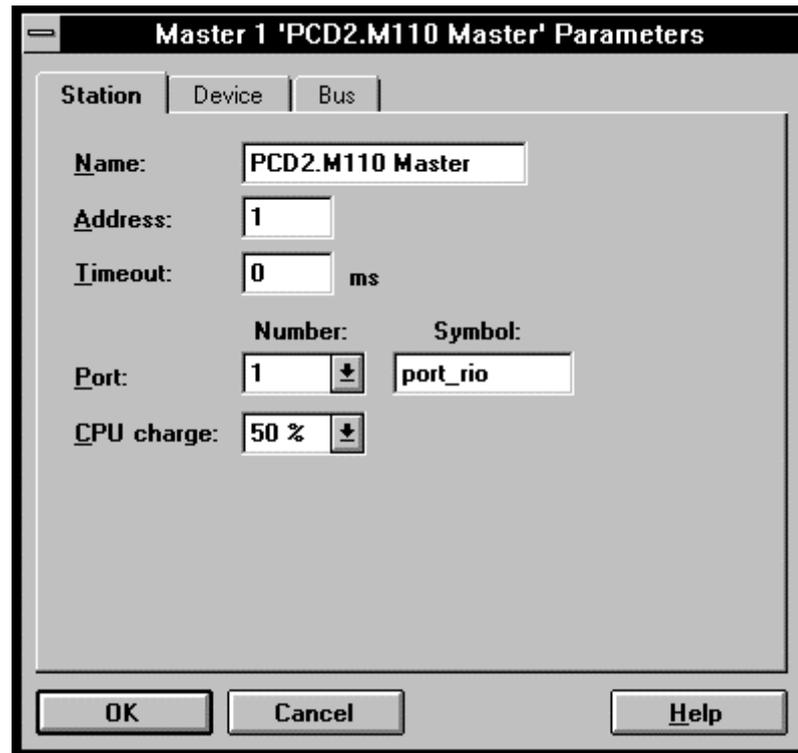
Dieser wichtige und umfangreiche Menüpunkt wird in der Folge in 3 eigenständige Unterabschnitte aufgeteilt:

- 5.3.3 'Station Parameter...': (für die Master-Station)
- 5.3.4 'Station Parameter...': (für eine nicht-intelligente Slave-Station)
- 5.3.5 'Station Parameter...': (für eine intelligente Slave-Station)

5.3.3 'Station Parameters...': (für die Master-Station)

'Station': Es können die Stationsnummer, das S-Bus Timeout, die Schnittstelle und die CPU-Belastung (vorgewählt werden).

Dieses Fenster kann auch durch einen Doppelklick auf die Master-Station im Konfigurationsfeld geöffnet werden.



Name: Name bzw. Typ der Master-Station (max 32 Zeichen)

Address: Adresse (Stations-Nummer) im RIO-Netzwerk.
Adressbereich: 1 - 254

Timeout: wenn = 0, werden die S-Bus Standardwerte eingesetzt

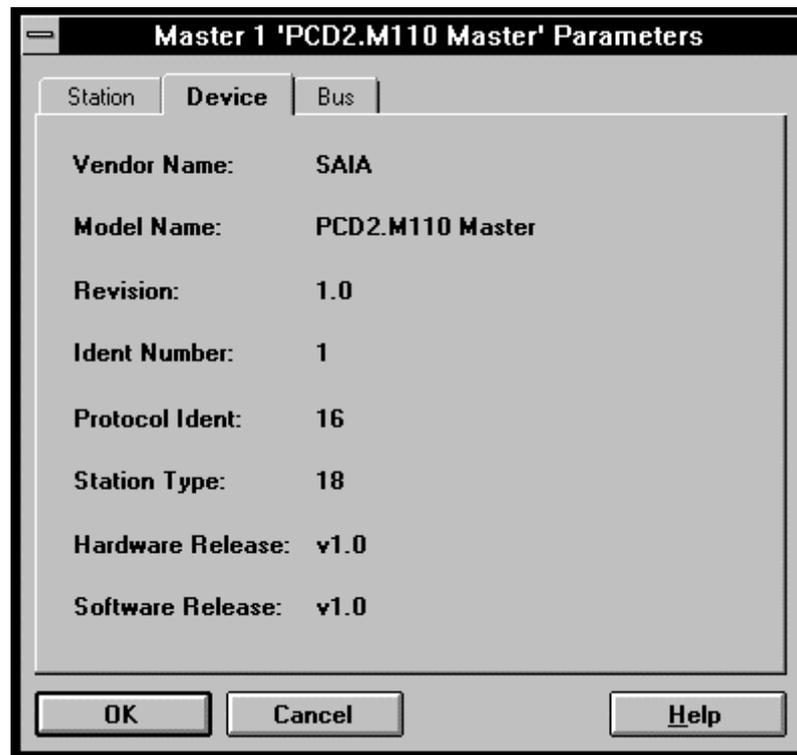
für 9'600 Bd	→	250 ms
für 19'200 Bd	→	200 ms
für 38'400 Bd	→	100 ms
für 76'800 Bd	→	50 ms

Port: Für die RIO-Aufgabe verwendete Schnittstelle.

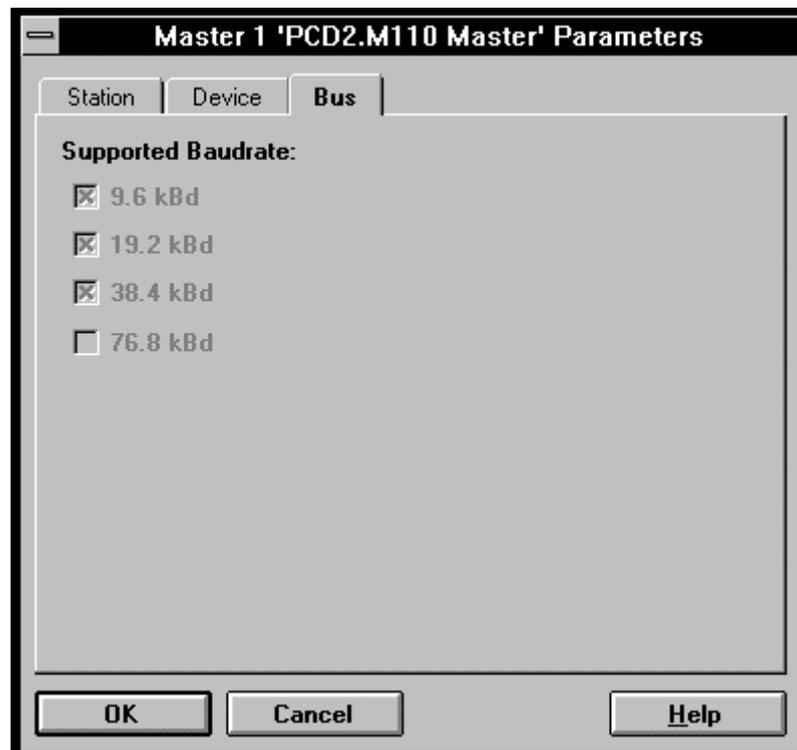
CPU charge: 25%, 50% oder 75%

Definiert die CPU-Belastung für die Kommunikation. Je höher der Wert, desto langsamer wird das Anwenderprogramm. (Noch nicht implementiert).

'Device': Alle Werte beziehen sich auf die gewählte Master-Station. Es kann nichts verändert werden.



'Bus': Alle Angaben sind von der gewählten Station abhängig und können nicht verändert werden.

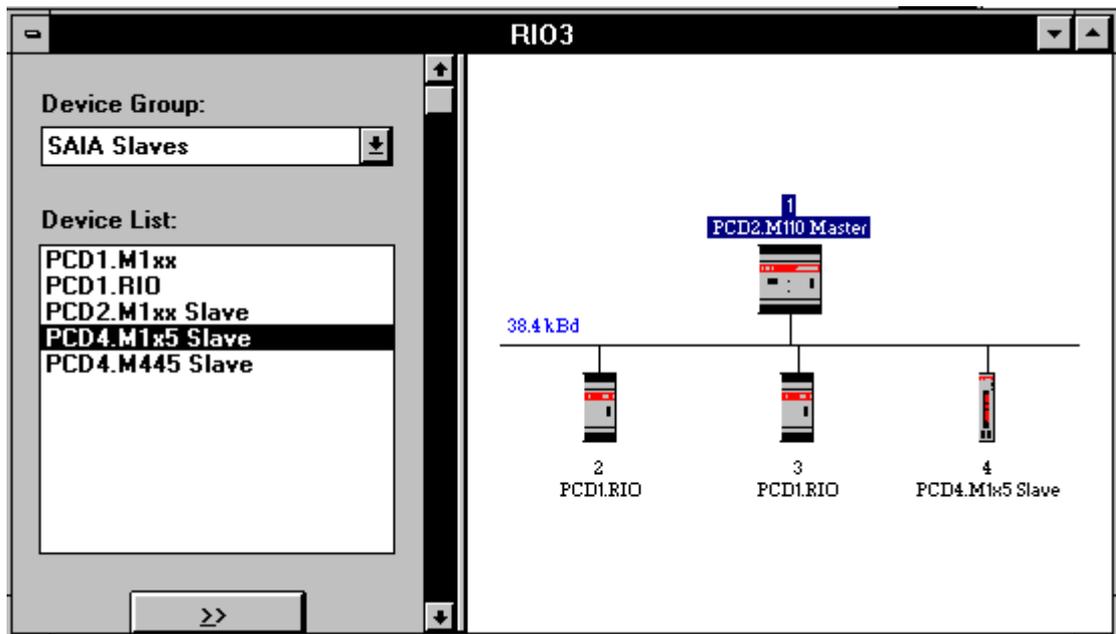


Nachdem aus der Liste der möglichen Slave-Stationen ein oder mehrere PCD-Typen ausgewählt und jeweils mit der Taste



ins Konfigurationsfeld eingefügt wurden, können die Stations-Parameter für die Slave-Stationen definiert werden.

Aus dem Menü 'View' ist 'Zoom to Fit' zu wählen, um die unten gezeigte Darstellung zu erhalten (Erläuterung erfolgt in Abschnitt 5.3.6).



Die Konfigurierung der nicht-intelligenten Slave-Stationen PCD1.RIO und die intelligenten Slave-Stationen wie PCD1.M1xx, PCD2.M1xx, PCD4.M1x5 oder PCD4.M445 ist unterschiedlich.

Bei den nicht-intelligenten Slaves muss die genaue Bestückung angegeben werden, damit die Master-Station das Medien-Abbild konfigurieren und den zyklischen Datenaustausch vorbereiten kann.

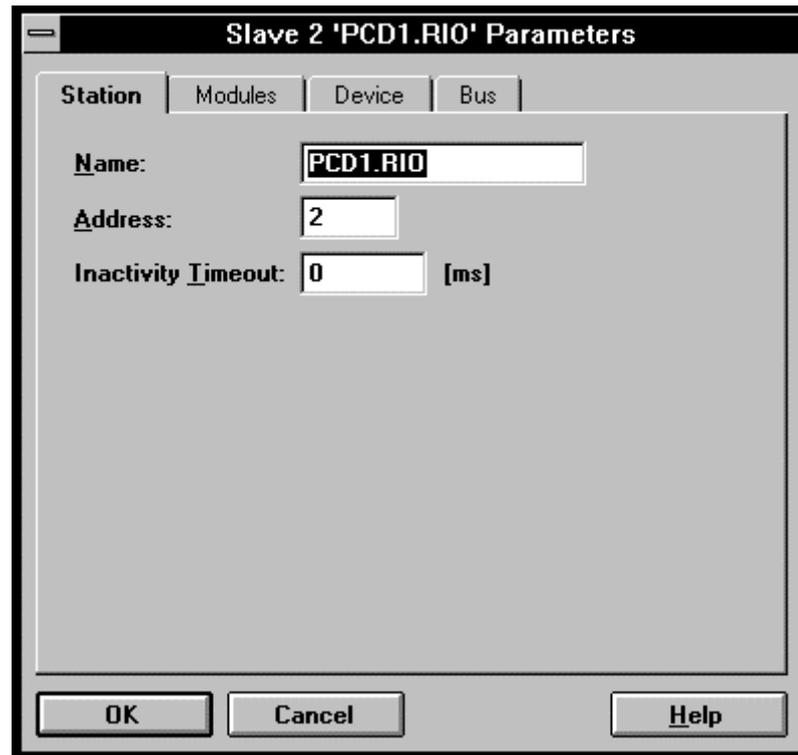
Bei den intelligenten Slaves muss definiert werden, welche Elemente eine RIO-Kommunikationsbeziehung haben sollen.

Wir beginnen mit den nicht-intelligenten Stationen, da dies in der Praxis sicher der Standardfall sein wird.

5.3.4 'Station Parameters...': (für eine nicht-intelligente Slave-Station)

'Station': Es kann für die vorliegende Station die Adresse und das Timeout verändert werden.

Dieses Fenster kann auch durch einen Doppelklick auf die entsprechende Slave-Station im Konfigurationsfeld geöffnet werden.



Name: Name bzw. Typ der Master-Station
(max 32 Zeichen)

Address: Adresse (Stations-Nummer) im RIO-Netzwerk.
Adressbereich: 1 - 126

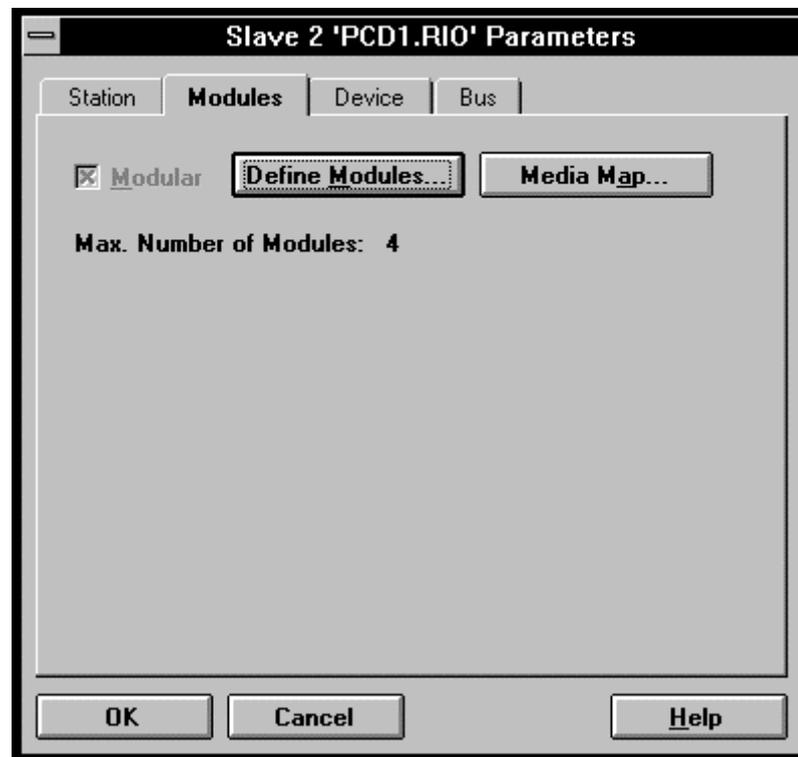
Timeout: Es kann das Rückschalten der Ausgänge und das Nullsetzen der Analogwert-Register nach einem Kommunikationsfehler definiert werden.

Zulässige Werte: 50 - 3'600'000 ms (1 Stunde) mit Inkrements von 10.

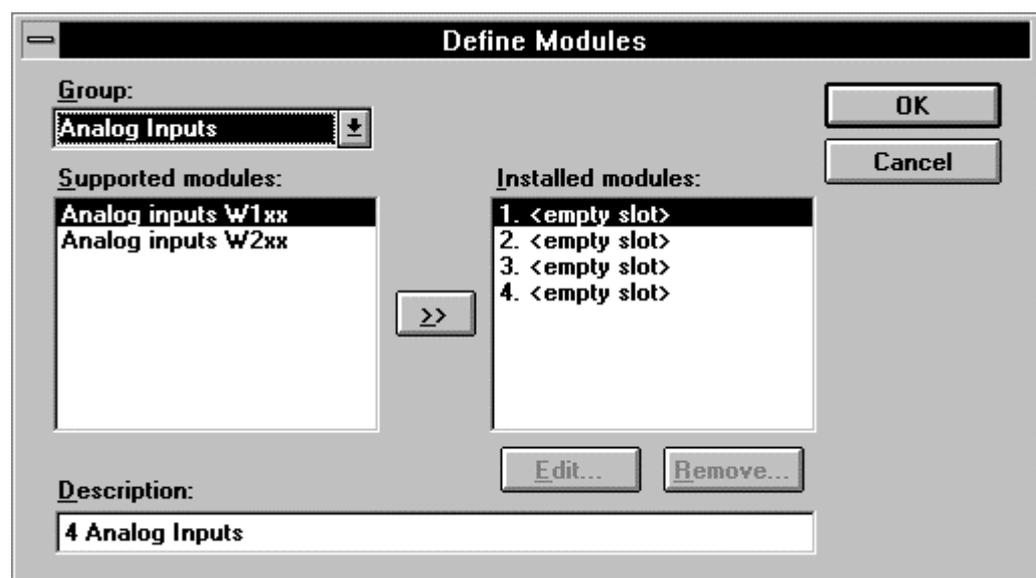
Wird der Wert Null gewählt, ist die Ausschaltfunktion nicht aktiviert.

'Modules': In diesem Fenster wird mit 'Define Modules...' die Bestückung der RIO-Station definiert.

Mit 'Media Map...' können die Elemente des Medien-Abbildes eingesehen und im Bedarfsfall angepasst werden.

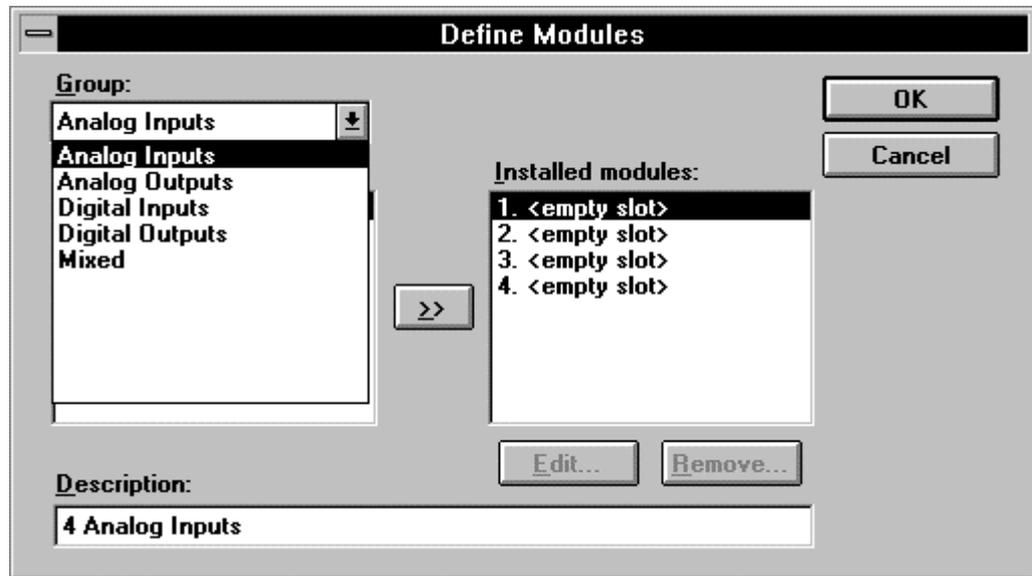


Nach dem Anklicken der Schaltfläche 'Define Modules...' erscheint das folgende Fenster:

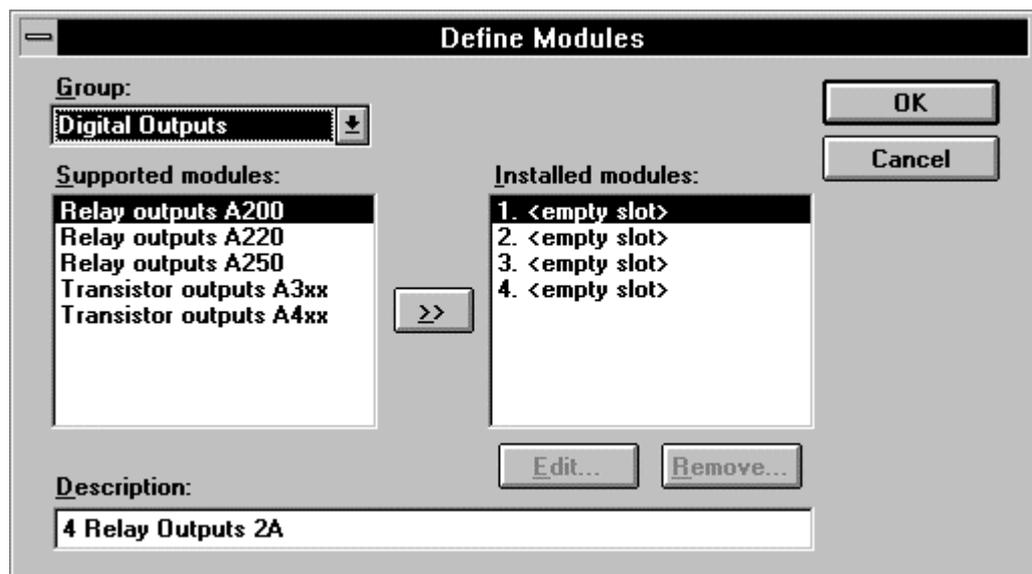


'Installed modules': Das gewählte Modul hat 4 Steckplätze, welche noch alle leer sind <empty slot>

'Group': Es kann die Auswahl der einsetzbaren Module aufgerufen werden (alphabetisch geordnet, beginnend mit 'Analog Inputs')



Wird z.B. 'Digital Outputs' gewählt, erscheinen unter der Rubrik 'Supported modules' alle einsetzbaren digitalen Ausgangsmodule.

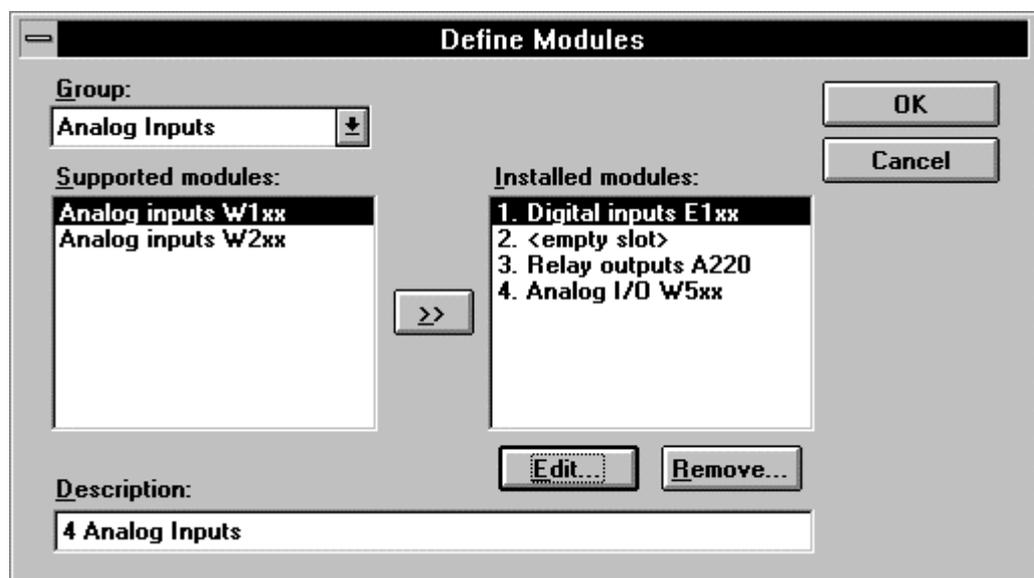


Wir wollen als Beispiel folgende Bestückung definieren:

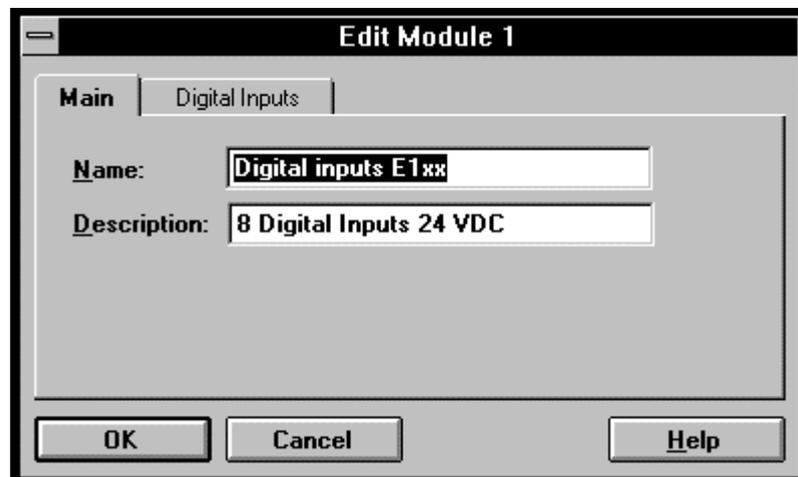
- Steckplatz 1: 8 digitale Eingänge PCD2.E110
- Steckplatz 2: leer
- Steckplatz 3: 6 digitale Relaisausgänge PCD2.A220
- Steckplatz 4: analoges I/O-Modul PCD2.W5xx

Es wird aus 'Group': 'Digital Inputs' gewählt, dann aus 'Supported modules': 'Digital inputs E1xx'. Bei 'Installed modules' wird der 1. Steckplatz gewählt und mit der Doppelpfeiltaste das definierte Modul eingesetzt

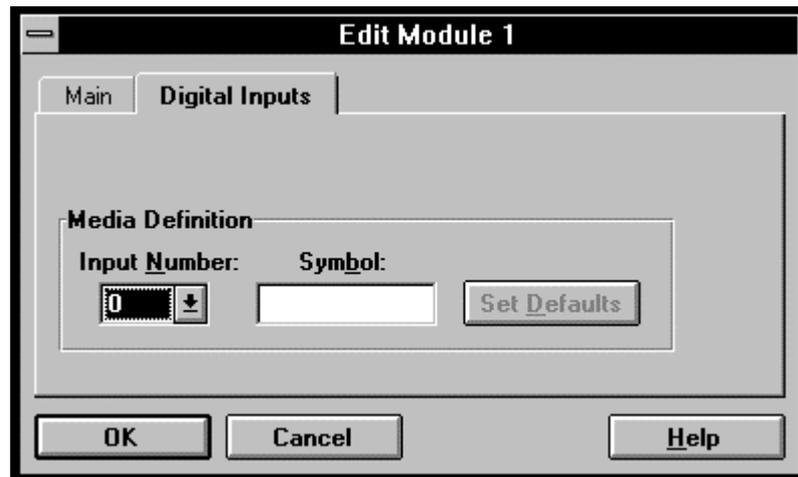
Mit den anderen Modulen wird gleich vorgegangen, so dass sich bei 'Installed modules' nach 'OK' folgendes Bild zeigt.



Die einzelnen Module müssen jetzt editiert werden. Wir beginnen mit den digitalen Eingängen. Nach 'Edit...' erscheint das folgende Bild für die Position 'Main':



Wir klicken 'Digital Inputs'. Folgendes Fenster erscheint:



Es muss jetzt jedem in der Anlage verwendeten Element, d.h. allen Elementen aller Stationen, ein Symbol zugewiesen werden, wobei das gleiche Symbol im ganzen Netzwerk nur ein einziges mal vorkommen darf. Eine direkte Adressierung, wie dies bei kleinen Aufgaben auf einer einzelnen PCD oft gemacht wird, ist hier, auch für Versuchszwecke, nicht möglich.

Schreibweise der Symbole:

- Symbole müssen mit einem Buchstaben (a-z, A-Z) beginnen.
- Ein Symbol muss aus mindestens 2 und höchstens 10 Charakter bestehen
- Das Unterstreichzeichen "_" ist zugelassen, jedoch nicht als 1. Charakter.
- .,:;\ usw. sind nicht zugelassen
- Das "ß" (Doppel-S) darf nicht verwendet werden.
- Es dürfen keine Assembler-Deklarationen (PUBL, AUTO, EQU...) und keine Mnemonics (STH, COB, BITI...) als symbolische Namen verwendet werden.

Damit der Überblick auch bei grösseren Netzwerken erhalten bleibt, ist eine gute Organisation der symbolischen Namen unerlässlich.

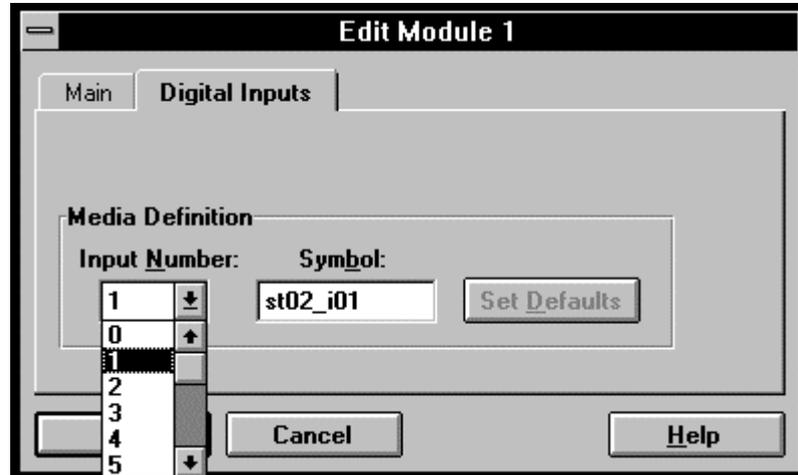
Für unsere Versuche werden die Symbole so gewählt, dass sowohl die Stationsnummer als auch die Elementadressen sofort erkannt werden können. So werden z.B. die digitalen Ausgänge auf dem 3. Steckplatz der Station Nr. 25 mit

st25_o16 ... st25_o23

oder der 1. analoge Eingang (Eingang 0) auf dem 2. Steckplatz der Station 13 mit

st13_ai08 bezeichnet.

Es sind vorteilhafterweise alle Elemente des Moduls mit einem Symbol zu versehen. Es müssten eigentlich nur diejenigen Elemente definiert werden, welche im Prozess tatsächlich verwendet werden. Es ist jedoch auch für die spätere Anwendung wahrscheinlich bequemer, wenn alle Elemente definiert sind.



Falls die Elemente, wie vorgeschlagen, fortlaufend bezeichnet werden, kann mit <Ctrl> + <C> das erste Element in die Zwischenablage gebracht und <Ctrl> + <V> im nächsten Feld wieder eingefügt und angepasst werden.

Da bereits das Editieren der Module einer kleineren RIO-Aufgabe ein recht mühsames Unterfangen ist, wurde eine Vereinfachung implementiert. Die Vereinfachung besteht darin, dass der symbolische Name des ersten Elementes eines Moduls angegeben wird und dann die nachfolgenden Ein- oder Ausgänge den gleichen Namen übernehmen und an der letzten Stelle eine automatische, fortlaufende Numerierung editiert wird.

Beispiel: Element 0: Symbol: st02_i16

nach Klick auf 'Set Defaults' (nur aktiv wenn "Input Number" <0> gewählt ist) werden die nachfolgenden Elemente automatisch mit

Element 1:	Symbol:	st02_i17
Element 2:	Symbol:	st02_i18
Element 3:	Symbol:	st02_i19
Element 4:	Symbol:	st02_i20
Element 5:	Symbol:	st02_i21
Element 6:	Symbol:	st02_i22
Element 7:	Symbol:	st02_i23

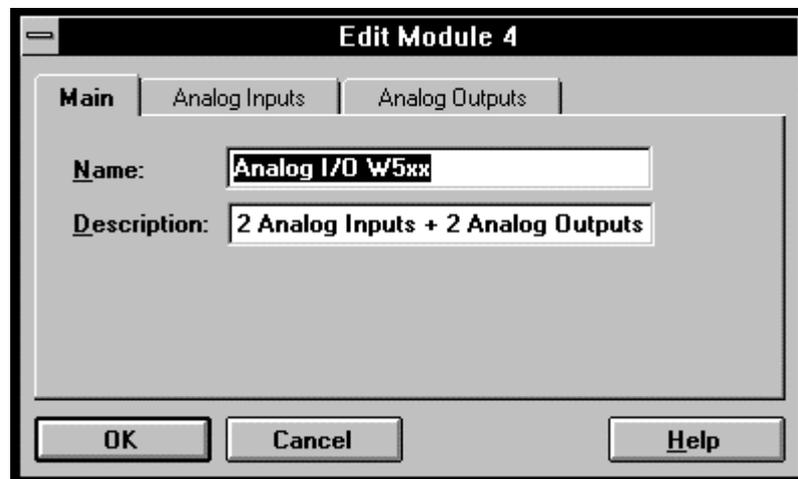
bezeichnet.

Endet der symbolische Name nicht mit einer Ziffer, beginnt die Numerierung mit 0.

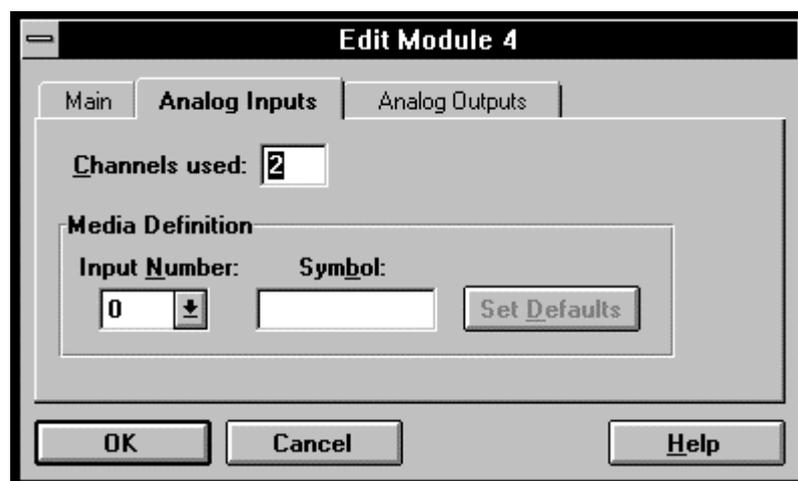
Wird z.B. der symbolische Name "Motor" definiert, wird nach 'Set Default' → "Motor0" bis "Motor7" oder bei "ABC_7" → "ABC_7" bis "ABC_14" usw. (bei Modulen mit 8 Elementen).

Die Editierung der andern digitalen Ein- und Ausgangsmodule erfolgt in gleicher Weise.

Das Editieren der Analogmodule geht in ähnlicher Weise vor sich. Wir editieren das Modul Nr. 4, ein gemischtes Modul mit Ein- und Ausgängen. Nach dem Anklicken des 4. Module in der Liste 'Installed modules' erscheint das folgende Fenster:



Nach dem Anklicken von 'Analog Inputs' erscheint das folgende Fenster:



Es kann angegeben werden ob beide oder nur 1 Kanal verwendet werden soll. Anschliessend werden in bekannter Art die symbolischen Namen eingegeben. Mit jedem Symbol wird bei den Analogmodulen ein Register bezeichnet.

Für die analogen Ausgänge wird gleich verfahren.

Wird aus dem Fenster 'Slave Parameters' die Schaltfläche 'Media Map...' angeklickt, erscheint das folgende Fenster:

Section	Base Media Type	Base Media Address	Used Media
Digital Inputs	Flag	64	8
Digital Outputs	Flag	2064	8
Analog Inputs	Register	64	2
Analog Outputs	Register	2064	2

Es werden die Basis Flag- und Register-Adressen welche für das Medienabbild in der Master-Station verwendet werden. Die Basisadressen können hier im Bedarfsfall angepasst werden.

Zurück zum Hauptfenster von 'Station Parameters...'

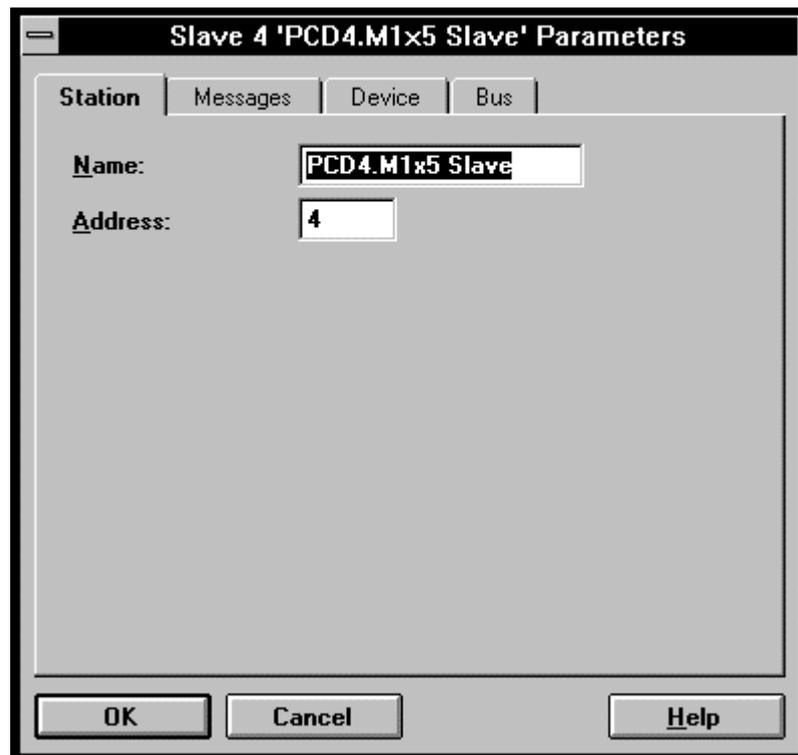
'Device': Das Anklicken von 'Device' lässt eine Aufstellung von stationsspezifischen Daten erscheinen, welche nicht verändert werden können.

'Bus': Das Anklicken von 'Bus' lässt eine Aufstellung der möglichen Baudraten erscheinen. Es kann auch hier nichts verändert werden.

5.3.5 'Station Parameters...': (für eine intelligente Slave-Station)

'Station': Es kann für die vorliegende Station die Adresse angepasst werden.

Dieses Fenster kann auch durch einen Doppelklick auf die entsprechende Slave-Station im Konfigurationsfeld geöffnet werden.



Abweichend vom Fenster für die Konfigurierung der nicht-intelligenten Slave-Stationen werden hier nicht die Module konfiguriert, sondern es werden unter der Rubrik 'Messages' Gruppen von Medien definiert, welche zyklisch behandelt werden, d.h. ins Medienabbild aufgenommen werden.

'Messages': Es erscheint das folgende Fenster:



Um Mediengruppen zu definieren, ist 'Add..' zu klicken. Es erscheint das folgende Fenster:



'Function': Die möglichen Beziehungen zwischen dem Master und dem intelligenten Slave sind vorgegeben und können gewählt werden.

'Description': Die gewählte Beziehung wird angezeigt

'Media definition':

Number of media to Transfer:

Anzahl zusammenhängender Medien, welche in einer Gruppe zu übertragen sind

Position, Symbol:
(Set Defaults)

Definition der einzelnen Medien mit einem symbolischen Namen. Es gelten die gleichen Regeln wie bei den nicht-intelligenten Slaves.

'Media Map':

First Master media Type/Address:

Mediatyp und Basisadresse der Mediengruppe im Master

First Slave media Type/Address:

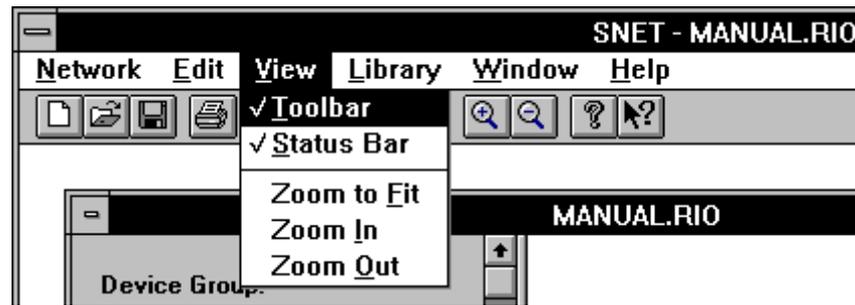
Mediatyp und Basisadresse der Mediengruppe in der intelligenten Slave-Station

Ein Beispiel dazu folgt im Kapitel 6.2: "Weiteres Beispiel".

'Device': Das Anklicken von 'Device' lässt eine Aufstellung von stationsspezifischen Daten erscheinen, welche nicht verändert werden können.

'Bus': Das Anklicken von 'Bus' lässt eine Aufstellung der möglichen Baudraten erscheinen. Es kann auch hier nichts verändert werden.

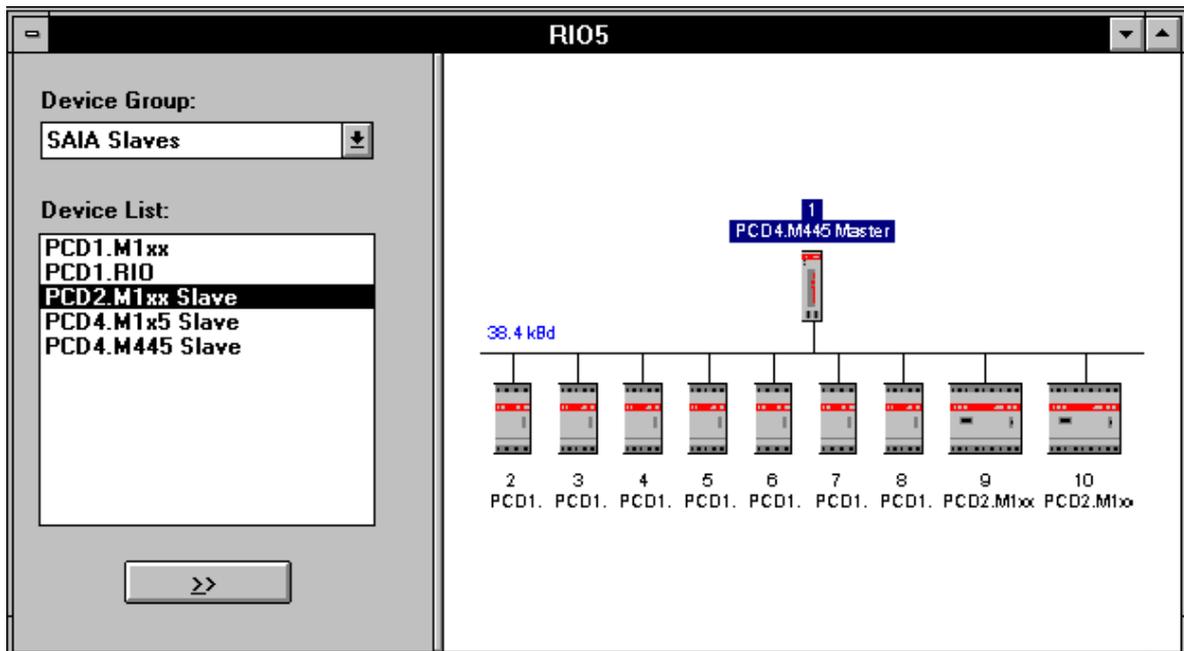
5.3.6 Das Untermenü 'View'



'Toolbar': Es kann die Werkzeugleiste ein- bzw. ausgeschaltet werden. Die Werkzeugleiste selbst wird im Abschnitt 5.3.10 beschrieben.

'Status Bar': Es kann die Statuszeile ein- bzw. ausgeschaltet werden. (Die Statuszeile befindet sich am unteren Bildrand des Hauptmenüs).

'Zoom to Fit': Es kann die RIO-Konfiguration im Konfigurationsfeld auf die Fenstergröße angepasst werden.



'Zoom In':



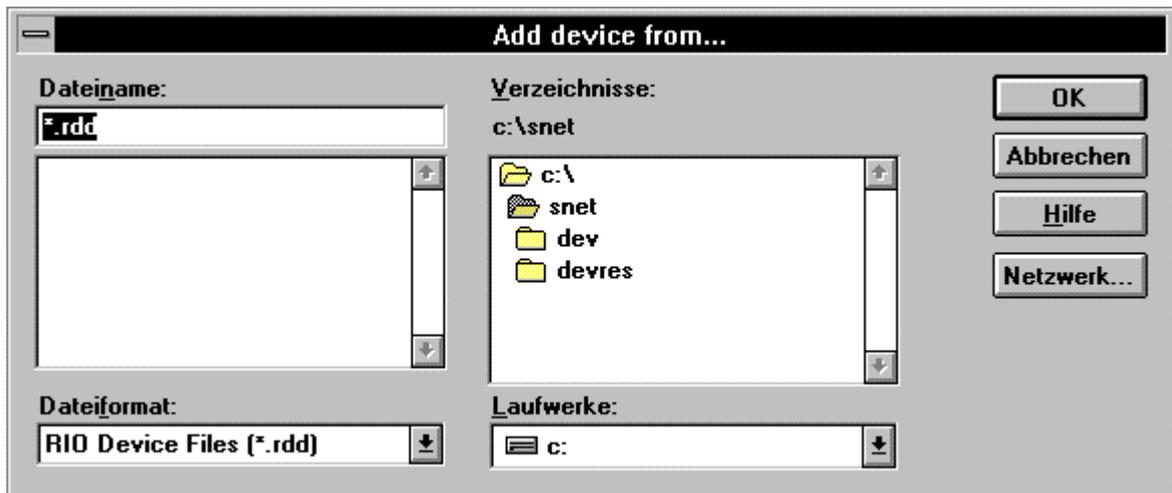
Die Konfiguration kann vergrößert und mit den Bildlaufleisten verschoben (gescrollt) werden. (Kommt bei größeren Konfigurationen zur Anwendung).

'Zoom Out':



Die vergrößerte Konfiguration wird auf die Größe wie beim Editieren zurückgebracht.

5.3.7 Das Untermenü 'Library'



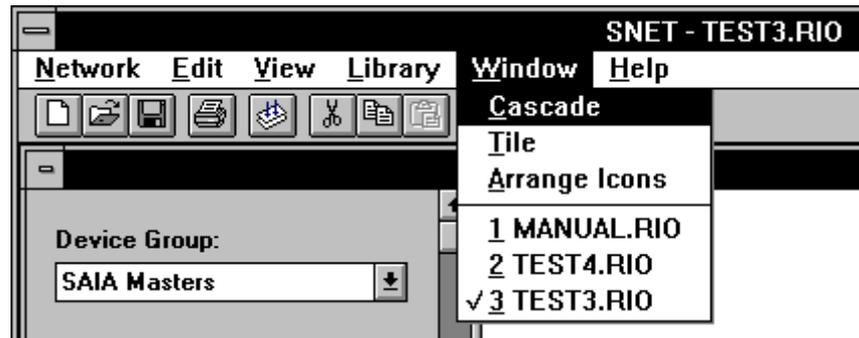
'Add Device...': Es können weitere RIO-Master- oder Slave-Typen zur bereits bestehenden Bibliothek zugefügt werden.

'Remove Device...': Es können RIO-Master- oder Slave-Typen aus der Bibliothek entfernt werden. Die entfernten Typen stehen danach nicht mehr für eine Konfiguration zur Verfügung.

Beim Versuch, einen "Device" zu entfernen, erscheint zuerst eine Warnung, welche quittiert werden muss.



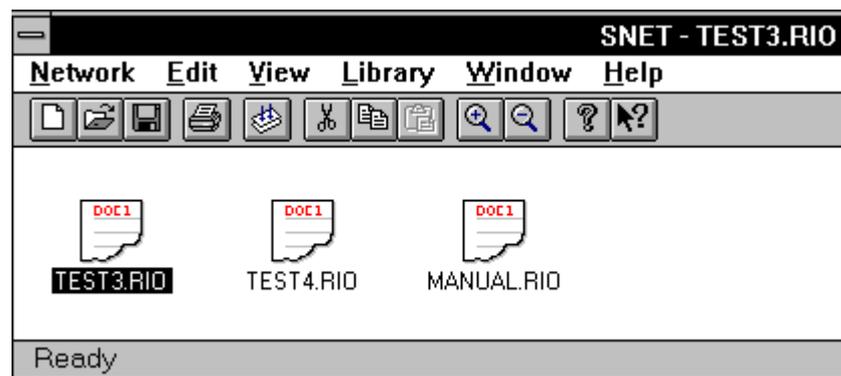
5.3.8 Das Untermenü 'Window'



'Cascade': Sind verschiedene Projekte offen, werden die einzelnen Fenster so angeordnet, dass jede Titelzeile sichtbar ist.

'Tile': Sind verschiedene Projekte offen, werden die einzelnen Fenster so von oben nach unten angeordnet, dass der Anfang jedes Projektes sichtbar bleibt.

'Arrange Icons': Wurden einige aufgerufene Projekte minimalisiert und auf dem Bildschirm verschoben, können die Ikonen der verschiedenen Projekte mit der vorliegenden Funktion neu ausgerichtet werden.



5.3.9 Das Untermenü 'Help'



'Index':



Interaktives Help in englischer Sprache.

Das "Help" hat etwa den gleichen Inhalt wie das vorliegende Kapitel des Handbuchs.

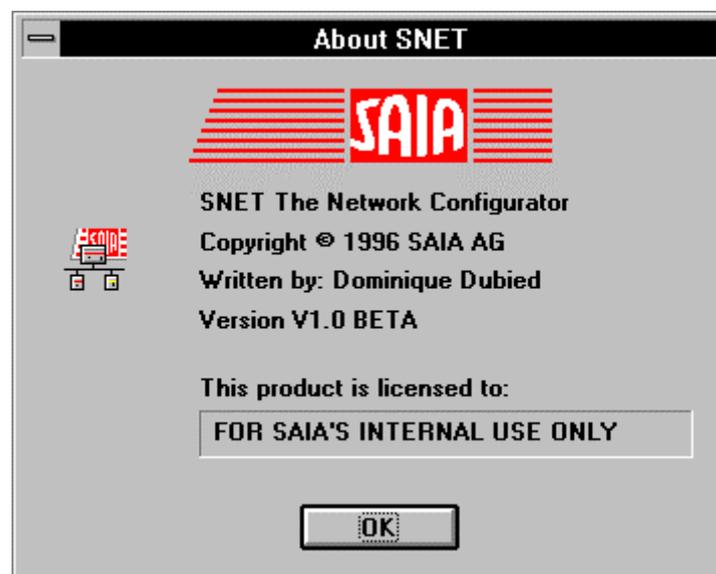
'Using Help':

Anleitung zur Verwendung des "Help". Diese Datei präsentiert sich in der Sprache, welche dem installierten WINDOWS zugrunde liegt.

'About Snet...':



Es erscheint das gleiche Bild wie beim Aufruf des RIO-Konfigurators.



Das Bild zeigt die Version des Konfigurators und den Namen des Lizenznehmers.

5.3.10 Die Funktionsleiste (Toolbar)



Die Funktionsleiste enthält die meist verwendeten Funktionen zum Arbeiten mit dem RIO-Konfigurator. Das Anklicken eines Funktionssymbols führt die Funktion direkt aus oder leitet diese ein.

Symbol	Tasten	Funktion	
	Ctrl + N	'New'	Neues Netzwerk
	Ctrl + O	'Open'	Bestehendes Netzwerk öffnen
	Ctrl + S	'Save'	Speichern des Netzwerks
	Ctrl + P	'Print'	Drucken
	Ctrl + G	'Generate'	Dateien RIO.DEF und RIOCFG.SRC erstellen
	Ctrl + X	'Cut'	Ausschneiden
	Ctrl + C	'Copy'	Kopieren
	Ctrl + V	'Paste'	Einfügen
		'View Zoom In'	Vergössern
		'View Zoom Out'	Verkleinern
		'Help About'	Version, Lizenznehmer
	Shift + F1	'Help Index'	Help

Notizen

6. Programmierung

6.1 Einführendes Beispiel (PCD2 + 2 PCD1.RIO)

Es sollen an einem einfachen Beispiel alle Schritte der Programmierung, von der Planung des Netzwerks bis zur Inbetriebnahme, gezeigt werden.

6.1.1 Aufgabe

Es soll ein RIO-Netzwerk mit 1 Master- und 2 nicht-intelligenten Slave-Stationen aufgebaut, programmiert und in Betrieb genommen werden.

Typen und Bestückung der 3 Stationen:

Master-Station:

Stations-Nummer:	2	
PCD-Typ:	PCD2.M110 oder PCD2.M120 (mit Zusatz-RAM-Speicher)	
Schnittstelle:	PCD7.F110 oder PCD7.F150, Port 1	
Displaymodul:	PCD2.F510 oder PCD2.F530	
I/O-Module:	Adr. 00 - 07	E110
	Adr. 16 - 23	E110 (2 BCD-Digits)
	Adr. 32 - 39	A400
	Adr. 48 - 55	A400

Slave-Stationen:

Stations-Nummer	5	
PCD-Typ	PCD1-RIO	
I/O-Module	Module 1	-
	Module 2	A400
	Module 3	-
	Module 4	-
Stations-Nummer	9	
PCD-Typ	PCD1-RIO	
I/O-Module	Module 1	E100
	Module 2	E100
	Module 3	-
	Module 4	W500

Programmieraufgabe:

- 1) Stellvertretend für ein Steuerprogramm, soll der Ausgang 55 in der Master-Station blinken.
- 2) Die Eingänge 4 - 7 der Master-Station sind an die Ausgänge 0 - 3 des Moduls 2 der Slave-Station Nr. 5 zu übertragen.
- 3) Die Eingänge 4 - 7 des Moduls 1 der Slave-Station Nr. 9 sind an die Ausgänge 32 - 35 der Master-Station zu übertragen.
- 4) Die Eingänge 0 - 3 des Moduls 1 der Slave-Station Nr. 9 sind an die Ausgänge 4 - 7 des Moduls 2 der Slave-Station Nr. 5 zu übertragen.
- 5) Der 2-stellige BCD-Wert an den Eingängen 16 - 23 der Master-Station ist an den analogen Ausgang 0 des W500-Moduls (Modul 4) der Slave-Station Nr. 9 zu übertragen.
- 6) Der analoge Ausgang 0 wird hardwaremässig auf den analogen Eingang 0 des gleichen Moduls geführt. (Brücke zwischen Klemmen 0 und 4 und Klemmen 1 und 9). Das Signal am analogen Eingang 0 soll am Displaymodul der Master-Station angezeigt werden.

(Durch die Hardware bedingt, werden nur die 8 unteren Bit des 12-Bit Analogmoduls verwendet).

Dem Projekt geben wir den Namen TEST1.

Es wird davon ausgegangen, dass die Stationen korrekt bestückt, die Stations-Nummern richtig eingestellt, der RS 485-Bus und die Speisung verdrahtet und die Abschluss- sowie die Pullup- und Pulldown-Widerstände vorschriftsgemäss eingestellt sind.

Weiter wird vorausgesetzt, dass der RIO-Konfigurator und die PG3-Utilities installiert und funktionsfähig sind.

6.1.2 Konfigurierung der Stationen

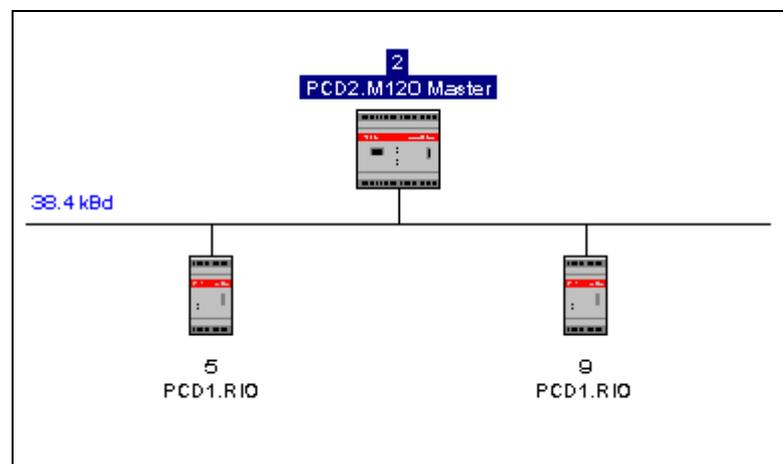
Wir rufen den RIO-Konfigurator auf und schalten diesen auf Vollbild. Dann 'Network' - 'New', Network type: "RIO-Network" - 'OK'.

Wir wählen unsere Master-Station und tragen diese mit der Doppelpfeiltaste ins Konfigurationsfeld. Wir schalten zu den Slaves, wählen PCD1.RIO und betätigen 2 mal die Doppelpfeiltaste. Aus dem Menü 'View' klicken wir 'Zoom to Fit'

Zum Anpassen der Stationsnummern und zum Definieren eines Timeouts klicken wir auf die einzelnen Stationen. Wir beginnen bei der Slave-Station 3. Dieser geben wir die neue Stationsnummer (Address) 9 und wählen ein Timeout von 1000 ms, 'OK'. Der Slave-Station 2 geben wir die Nummer 5, das Timeout belassen wir auf 0, da diese Station keine Ausgänge hat, welche nach dem abgelaufenen Timeout zurück zu schalten sind.

Der Master-Station geben wir die Stationsnummer (Address) 2. Das Timeout belassen wir auf dem Standardwert 0. Die Nummer der Schnittstelle (Port) belassen wir auf 1, der symbolische Name (port_rio) und die CPU-Belastung (50%) belassen wir ebenfalls. 'OK'.

Unsere Konfiguration präsentiert sich nun wie folgt:



Die Baudrate von 38.4 kBd (standardwert) wurde automatisch gewählt. Es ist zugleich die höchste Übertragungsgeschwindigkeit für die eingesetzten Stationen.

Die Master-Station ist hiermit fertig konfiguriert. Bei den Slave-Stationen sind jetzt die I/O-Module zu konfigurieren.

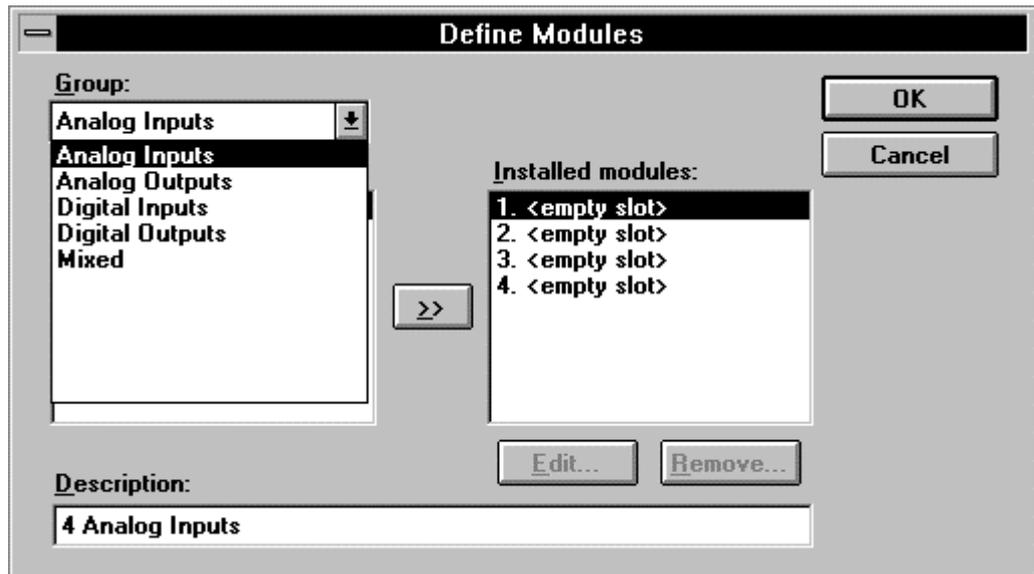
Vorsichtshalber speichern wir das bis hier Erarbeitete mit 'Network' - 'Save As...' unter dem Namen "TEST1.RIO" ab.

6.1.3 Konfigurierung der I/O-Module der Slave-Stationen

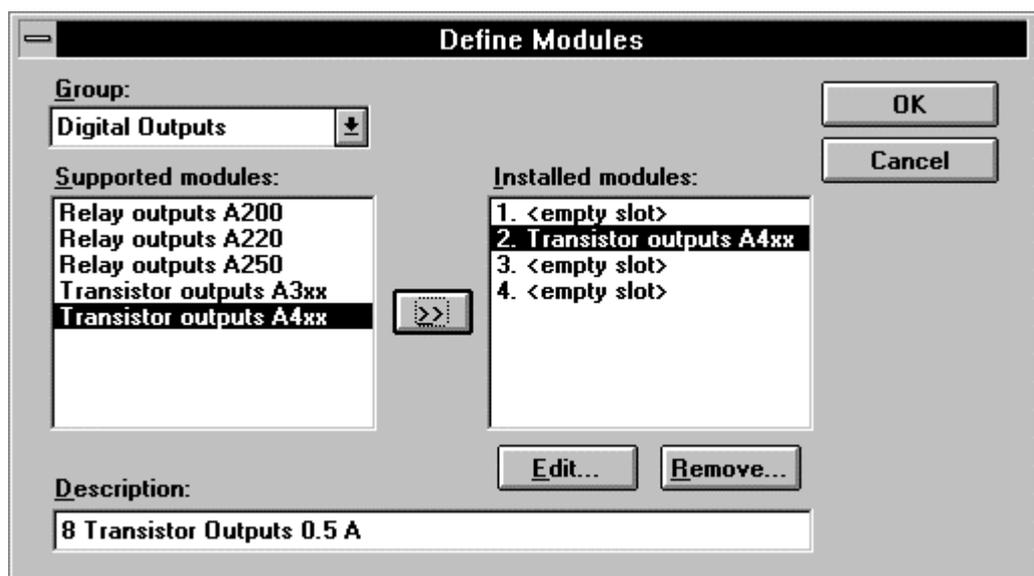
Wir beginnen mit der Slave-Station Nr. 5. Diese hat als Modul 2 eine digitale Ausgangskarte PCD2.A400. Wir definieren alle 8 Ausgänge mit

st05_o08 bis st05_o15

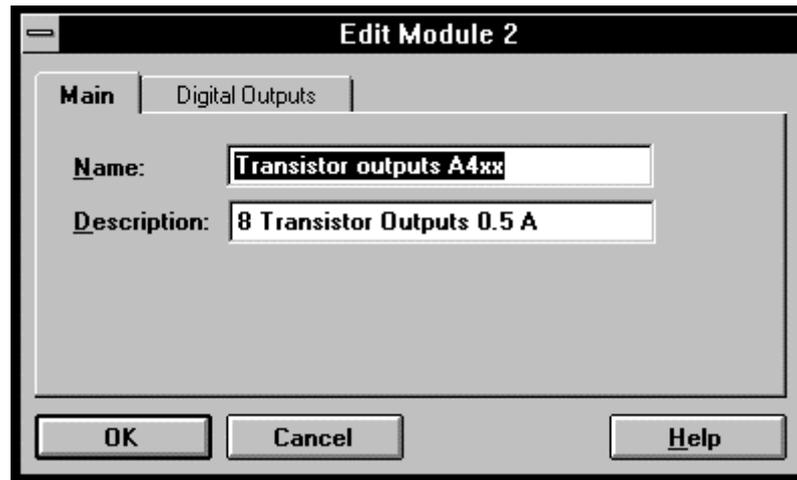
Nach einem Doppelklick auf die Station 5 erscheint das bereits angepasste Bild "Slave Parameters" - 'Station'. Wir klicken auf 'Modules' und dann 'Define Modules...'. und öffnen das Fenster 'Group'.



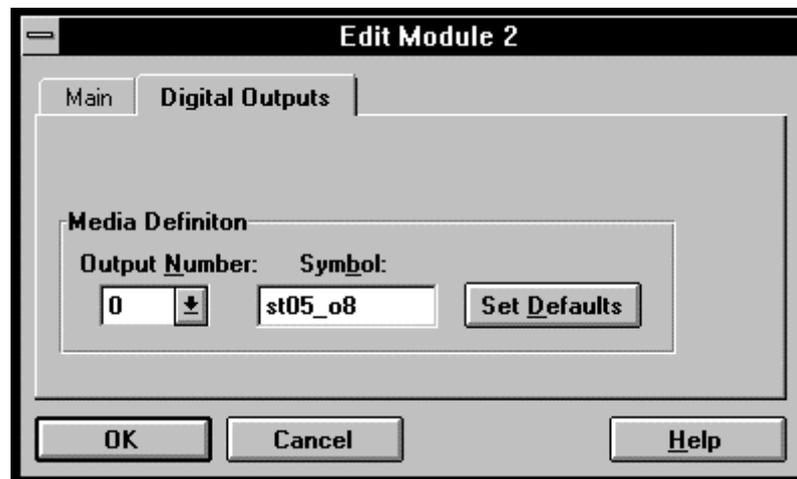
Wir wählen 'Digital Outputs' und daraus 'Transistor outputs A4xx'. Im Fenster "Installed modules" wählen wir das 2. Modul und betätigen die Doppelpfeiltaste. Unser gewünschtes Modul ist am gewählten Steckplatz eingetragen und kann editiert werden.



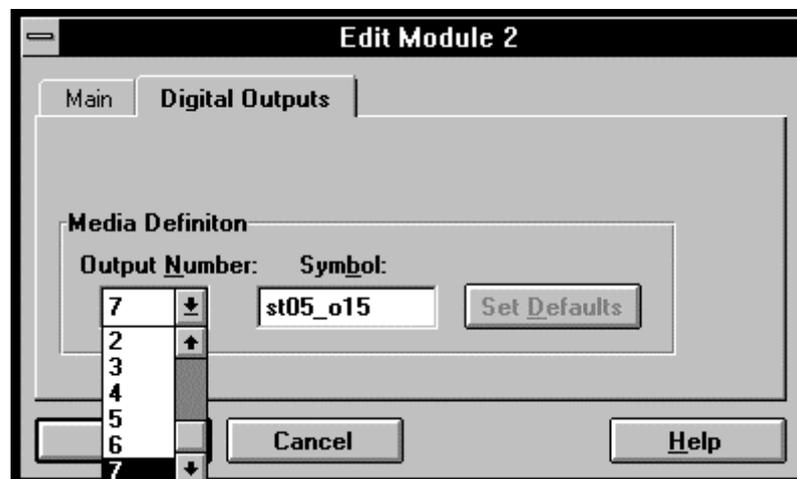
Es ist nun 'Edit...' zu betätigen.



Nach dem Betätigen von 'Digital Outputs' kann mit dem Editieren des ersten symbolischen Namen <st05_o08> für den Ausgang 0 des Moduls 2 begonnen werden.



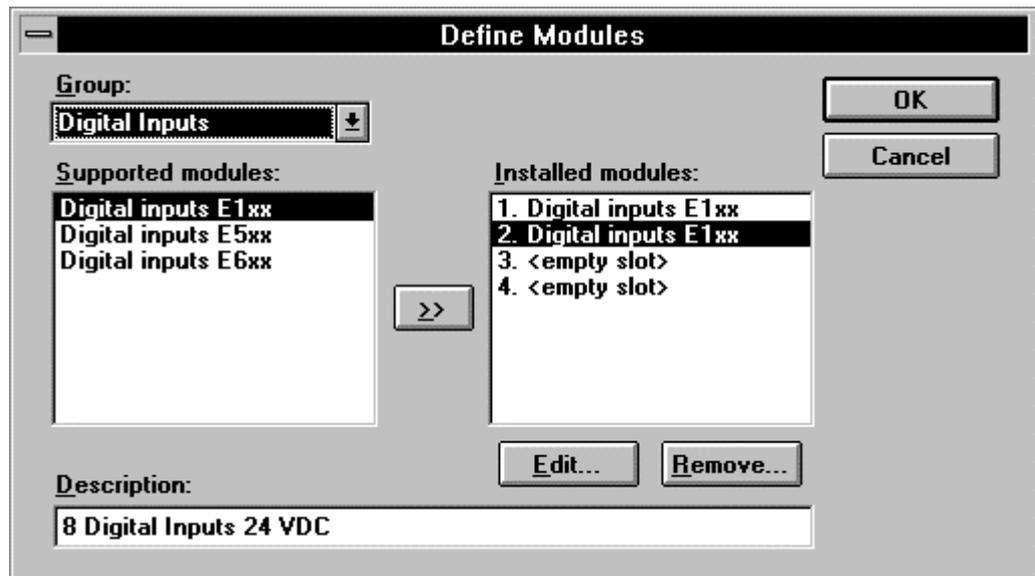
Nach dem 1. Symbol werden ohne 'OK', die nächsten Elemente "von Hand" editiert oder es werden mit 'Set Defaults' die nächsten 7 Ausgänge automatisch bezeichnet.



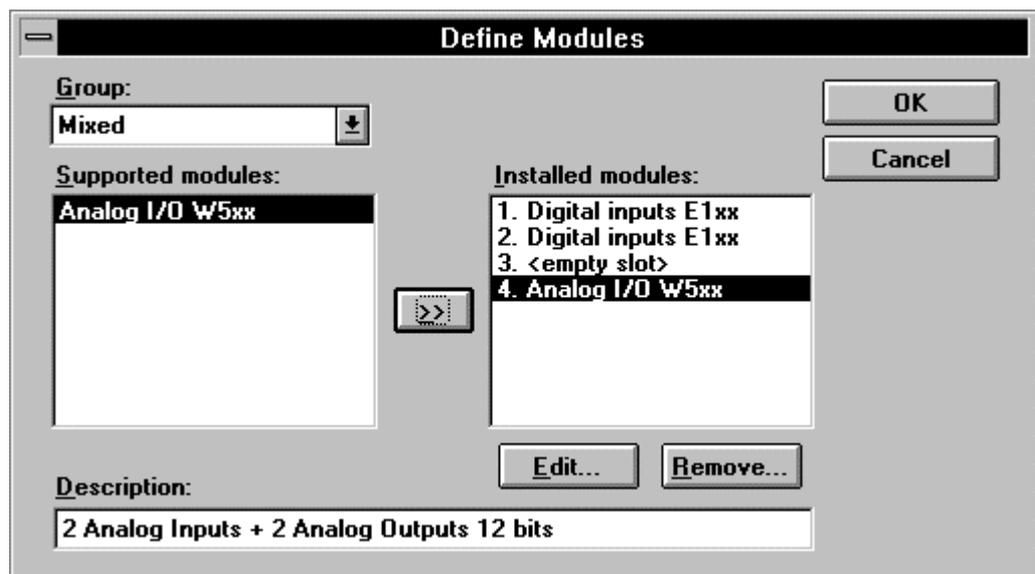
Sind alle 8 Ausgänge bezeichnet, wird mit 'OK' bestätigt.

Die Slave-Station Nr. 5 hat nur dieses einzige Ausgangsmodul. Die Editierung ist für diese Station somit abgeschlossen. Es werden alle Fenster mit 'OK' verlassen, bis wieder das Konfigurierungsfenster zum Vorschein kommt.

Wir editieren in gleicher Weise die Station Nr. 9 mit 2 E110 Modulen auf den Plätzen 1 und 2 ...

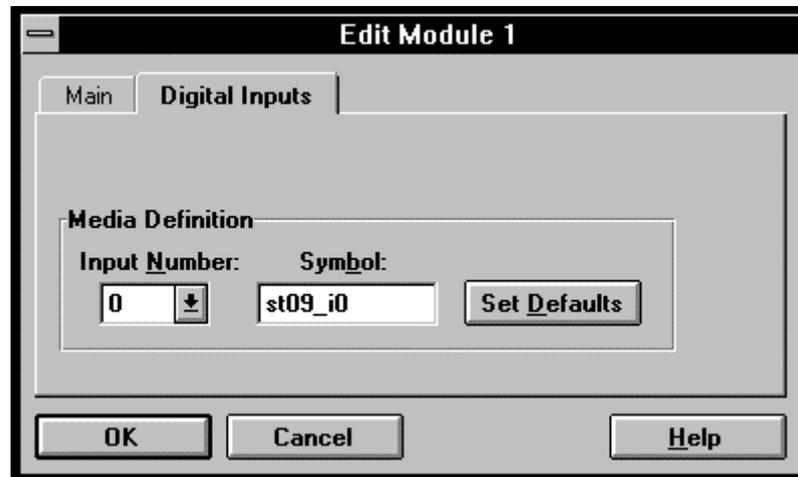


... und einem kombinierten analogen Ein-/Ausgangsmodul (mixed) auf Platz 4.

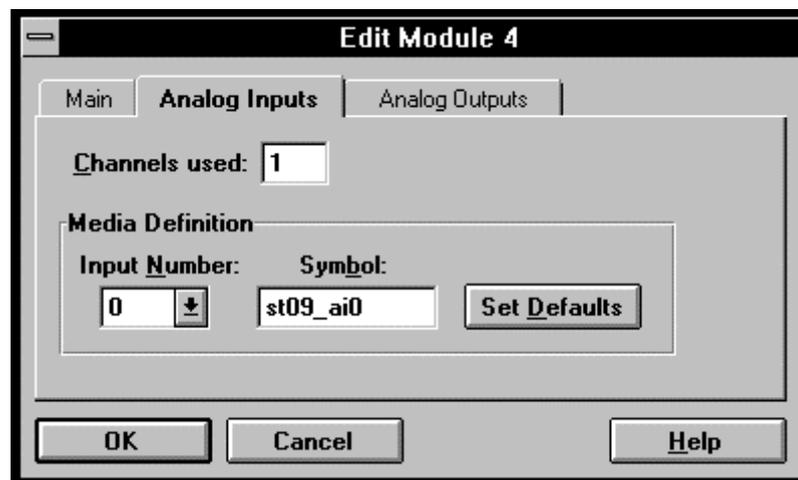


Wir editieren die Eingangsmodule 1 und 2 (16 Eingänge) mit

st09_i0 bis st09_i15 ...



den analogen Eingang mit st09_ai0 und
den analogen Ausgang mit st09_ao0 wobei
sowohl beim Ein- wie beim Ausgang nur je 1 Kanal verwendet wird.
(Channels used: 1)

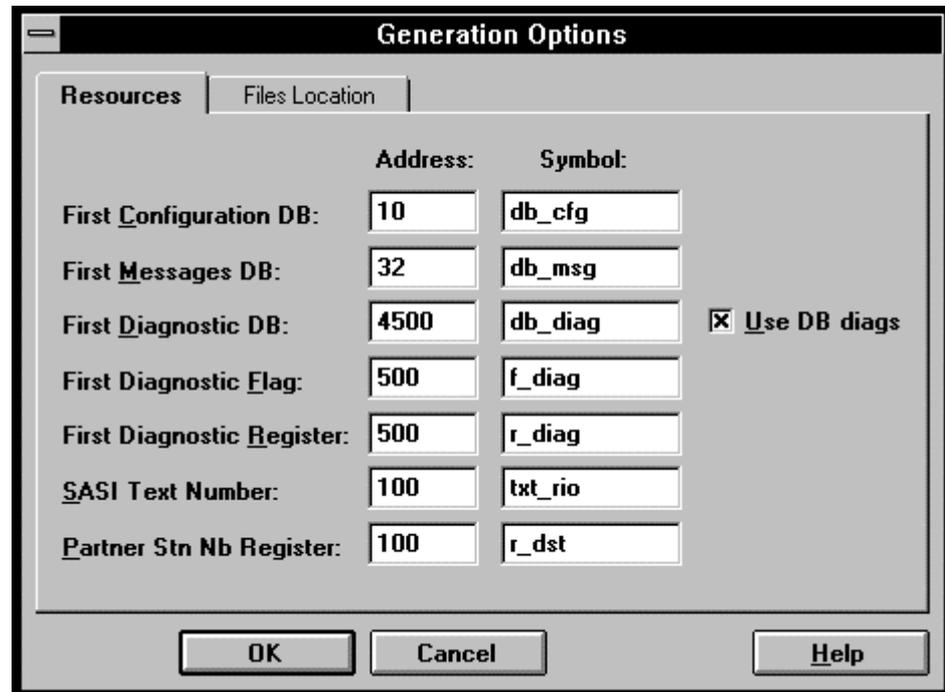


und schliessen die Editierung der Station Nr. 9 ab.

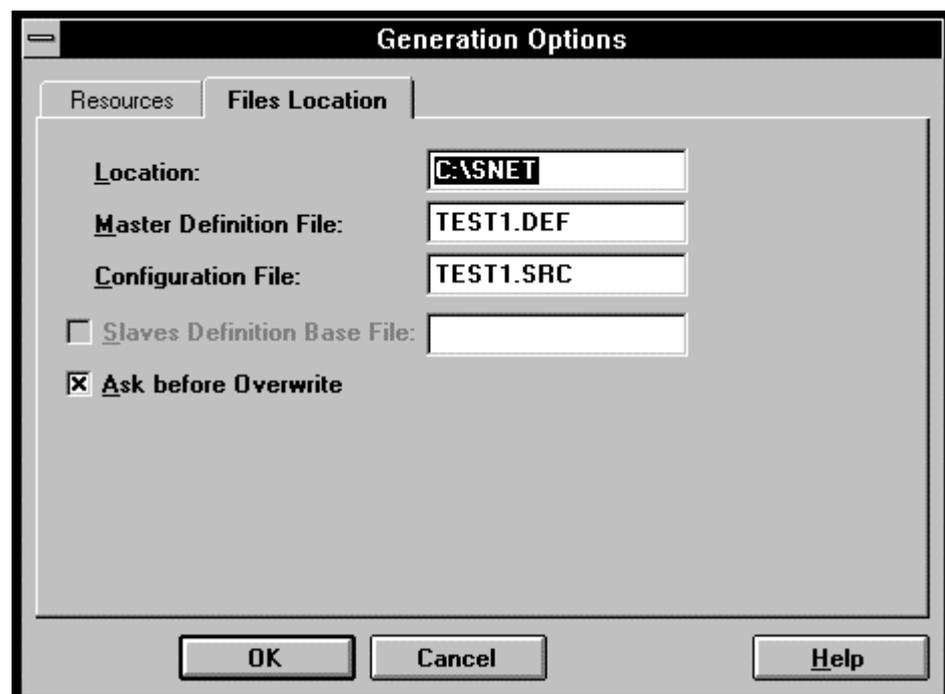
6.1.4 Generieren der Hilfsdateien

Nachdem alle Stationen konfiguriert und editiert sind, können die beiden Hilfsdateien "TEST1.DEF" und "TEST1.SRC" erzeugt werden.

Die aus 'Network' - 'Generate Options...' angegebenen Voreinstellungen belassen wir, wie diese sind:



Nach dem Anklicken von 'Files Location' erscheint das folgende Bild:



Dies belassen wir auch und betätigen mit 'OK'.

Wir klicken auf 'Generate...' aus dem Menü 'Network'.

Es werden die 3 Dateien

- TEST1.RIO (Konfigurierung)
- TEST1.DEF (Definitions-Datei mit alle EQU)
- TEST1.SRC (Konfigurations-Datei)

erzeugt.

Es muss jetzt noch das Anwenderprogramm geschrieben werden, welches neben dem Prozess-Steuerprogramm auch die Kommunikationsbeziehungen des RIO-Netzwerks enthält.

Bevor wir mit der Editierung des Anwenderprogramms beginnen, drucken wir die oben erwähnten Dateien aus, um einen Überblick über die verwendeten Symbole zu bekommen und allfällige Kollisionen von Medienbereichen erkennen zu können.

6.1.5 Die Definitions-Datei (TEST1.DEF)

```

; Master 2 (PCD2.M120 Master)
;-----

port_rio    EQU 1            ;RIO: Port number

; Slave 5 (PCD1.RIO)
;-----

; Module 2 (Transistor outputs A4xx)
st05_o8     EQU F 2064      ;Slave 5, mod 2, dig output 0
st05_o9     EQU F 2065      ;Slave 5, mod 2, dig output 1
st05_o10    EQU F 2066      ;Slave 5, mod 2, dig output 2
st05_o11    EQU F 2067      ;Slave 5, mod 2, dig output 3
st05_o12    EQU F 2068      ;Slave 5, mod 2, dig output 4
st05_o13    EQU F 2069      ;Slave 5, mod 2, dig output 5
st05_o14    EQU F 2070      ;Slave 5, mod 2, dig output 6
st05_o15    EQU F 2071      ;Slave 5, mod 2, dig output 7

; Slave 9 (PCD1.RIO)
;-----

; Module 1 (Digital inputs E1xx)
st09_i0     EQU F 96        ;Slave 9, mod 1, dig input 0
st09_i1     EQU F 97        ;Slave 9, mod 1, dig input 1
st09_i2     EQU F 98        ;Slave 9, mod 1, dig input 2
st09_i3     EQU F 99        ;Slave 9, mod 1, dig input 3
st09_i4     EQU F 100       ;Slave 9, mod 1, dig input 4
st09_i5     EQU F 101       ;Slave 9, mod 1, dig input 5
st09_i6     EQU F 102       ;Slave 9, mod 1, dig input 6
st09_i7     EQU F 103       ;Slave 9, mod 1, dig input 7

; Module 2 (Digital inputs E1xx)
st09_i8     EQU F 104       ;Slave 9, mod 2, dig input 0
st09_i9     EQU F 105       ;Slave 9, mod 2, dig input 1
st09_i10    EQU F 106       ;Slave 9, mod 2, dig input 2
st09_i11    EQU F 107       ;Slave 9, mod 2, dig input 3
st09_i12    EQU F 108       ;Slave 9, mod 2, dig input 4
st09_i13    EQU F 109       ;Slave 9, mod 2, dig input 5
st09_i14    EQU F 110       ;Slave 9, mod 2, dig input 6
st09_i15    EQU F 111       ;Slave 9, mod 2, dig input 7

; Module 4 (Analog I/O W5xx)
st09_ai0    EQU R 96        ;Slave 9, mod 4, alg input 0

st09_ao0    EQU R 2096      ;Slave 9, mod 4, alg output 0

; Generation Options
db_cfg      EQU 10          ;RIO: First Configuration DB
db_msg      EQU 32          ;RIO: First Messages DB
db_diag     EQU 4500       ;RIO: First Diagnostic DB
f_diag      EQU 500        ;RIO: First Diagnostic Flag
r_diag      EQU 500        ;RIO: First Diagnostic Register
txt_rio     EQU 100        ;RIO: SASI Text Number
r_dst       EQU 100        ;RIO: Partner Stn Nb Register

```

6.1.6 Die Konfigurationsdatei (TEST1.SRC)

```

$INCLUDE TEST1.DEF

; Configuration of the stations

DB db_cfg+0 [8] 5, 0, 0, 2000H, 0, 0, 0, 1
DB db_cfg+1 [8] 9, 1000, 1000H,1000H, 0, 6110H,0, 1
DB db_cfg+2 [8] -1, -1, -1, -1, -1, -1, -1, -1

; Messages of the stations

DB db_msg+0 [7] 5, 8, 8, 2064, 64
DB db_msg+1 [7] 9, 1, 16, 0, 96
DB db_msg+2 [7] 9, 3, 1, 0, 96
DB db_msg+3 [7] 9, 10, 1, 2096, 32
DB db_msg+4 [7] -1, -1, -1, -1, -1

; Diagnostics

DB db_diag+0 [127]
DB db_diag+1 [127]

; SASI Text

TEXT txt_rio "UART:38400,0,,0;"
             "MODE:RM1,R",r_dst," ,DB",db_msg,";"
             "CONF:DB",db_cfg,";"
             "DIAG:F",f_diag," ,R",r_diag," ,DB",db_diag,";"

```

Die Bedeutung der einzelnen Codes in den DB sind im Kapitel 7 in den Abschnitten 7.6.1 und 7.6.2 beschrieben.

Diese vorliegende Datei "TEST1.SRC" sowie die Datei "TEST1.DEF" auf der vorangehenden Seite, sind so wiedergegeben, wie diese vom Konfigurator erzeugt wurden.

6.1.7 Die "RIO"-Datei (TEST1.RIO)

```

*****
USER:          FOR SAIA'S INTERNAL USE ONLY
EDITOR:       SNET V1.0 BETA
FILENAME:    C:\SNET\TEST1.RIO
DATE:        6.08.96
*****

*****
* Network Overview *
*****

(2) PCD2.M120 Master (PCD2.M120 Master)
    (5) PCD1.RIO (PCD1.RIO)
    (9) PCD1.RIO (PCD1.RIO)

*****
* Bus Description *
*****
BaudRate = 38400 Baud

*****
* Station Parameters *
*****

Master 2 (PCD2.M120 Master)
=====
Vendor = SAIA
Model = PCD2.M120 Master
Revision = 1.0
Identification Number = 1
Protocol = 16
Type = 18
Hardware Release = v1.0
Software Release = v1.0
Port = 1 (port_rio)
Timeout = 0 [ms]
CPU Charge = 50
    
```

```
Simple Slave 5 (PCD1.RIO)
=====
Vendor = SAIA
Model = PCD1.RIO
Revision = 1.0
Identification Number = 1
Protocol = 16
Type = 16
Hardware Release = 1.0
Software Release = 1.0
Timeout = 0
Modular = Yes
Maximum Number of Modules = 4

Modules
-----
Module(s) 2: Transistor outputs A4xx
Description : 8 Transistor Outputs 0.5 A
Number of Digital Outputs Used = 8
Digital Output 0 : st05_o8 (F 2064)
Digital Output 1 : st05_o9 (F 2065)
Digital Output 2 : st05_o10 (F 2066)
Digital Output 3 : st05_o11 (F 2067)
Digital Output 4 : st05_o12 (F 2068)
Digital Output 5 : st05_o13 (F 2069)
Digital Output 6 : st05_o14 (F 2070)
Digital Output 7 : st05_o15 (F 2071)
```

```
Simple Slave 9 (PCD1.RIO)
=====
Vendor = SAIA
Model = PCD1.RIO
Revision = 1.0
Identification Number = 1
Protocol = 16
Type = 16
Hardware Release = 1.0
Software Release = 1.0
Timeout = 1000
Modular = Yes
Maximum Number of Modules = 4

Modules
-----
Module(s) 1: Digital inputs Elxx
Description : 8 Digital Inputs 24 VDC
Number of Digital Inputs Used = 8
Digital Input 0 : st09_i0 (F 96)
Digital Input 1 : st09_i1 (F 97)
Digital Input 2 : st09_i2 (F 98)
Digital Input 3 : st09_i3 (F 99)
Digital Input 4 : st09_i4 (F 100)
Digital Input 5 : st09_i5 (F 101)
Digital Input 6 : st09_i6 (F 102)
Digital Input 7 : st09_i7 (F 103)

Module(s) 2: Digital inputs Elxx
Description : 8 Digital Inputs 24 VDC
Number of Digital Inputs Used = 8
Digital Input 0 : st09_i8 (F 104)
Digital Input 1 : st09_i9 (F 105)
Digital Input 2 : st09_i10 (F 106)
Digital Input 3 : st09_i11 (F 107)
Digital Input 4 : st09_i12 (F 108)
Digital Input 5 : st09_i13 (F 109)
Digital Input 6 : st09_i14 (F 110)
Digital Input 7 : st09_i15 (F 111)

Module(s) 4: Analog I/O W5xx
Description : 2 Analog Inputs + 2 Analog Outputs 12 bits
Number of Analog Inputs Used = 1
Analog Input 0 : st09_ai0 (R 96)
Number of Analog Outputs Used = 1
Analog Output 0 : st09_ao0 (R 2096)
```

6.1.8 Das Anwenderprogramm (TEST1_X.SRC)

```

#include test1.def

        xob          16
        sasi         port_rio
                   txt_rio

        exob
; -----
        xob          30
        inc          c    100
        exob
; -----
        cob          0          ; Steuerprogramm für Prozess
                   0
        stl          t    0          ; Aufgabe Punkt 1)
        ldl          t    0          ; Blinker
                   5
        com          o    55
        ecob
; -----

        cob          1          ; RIO-Kommunikationsbeziehungen
                   0

        sth          i    4          ; Aufgabe Punkt 2)
        out          st05_o8
        sth          i    5
        out          st05_o9
        sth          i    6
        out          st05_o10
        sth          i    7
        out          st05_o11

        sth          st09_i4      ; Aufgabe Punkt 3)
        out          o    32
        sth          st09_i5
        out          o    33
        sth          st09_i6
        out          o    34
        sth          st09_i7
        out          o    35

        sth          st09_i0      ; Aufgabe Punkt 4)
        out          st05_o12
        sth          st09_i1
        out          st05_o13
        sth          st09_i2
        out          st05_o14
        sth          st09_i3
        out          st05_o15

        digir        2          ; Aufgabe Punkt 5)
                   i    16
                   st09_ao0

        dsp          st09_ai0      ; Aufgabe Punkt 6)

        ecob

```

Dies ist das Anwenderprogramm, welches der Programmierer für eine einfache RIO-Anlage zu editieren hat. Die Dateien mit den spezifischen RIO-Belangen wurde bereits durch den Konfigurator erstellt und wurden bereits gezeigt..

Die Datei "TEST1.DEF" wird mit "\$include" ins Anwenderprogramm eingefügt, die andere Datei, welche der Konfigurator erzeugt hat, "TEST1.SRC", wird zum Anwenderprogramm gelinkt. Das Anwenderprogramm muss deshalb einen andern Namen als das Konfigurationsprogramm haben. In unserem Fall nennen wir das Anwenderprogramm "TEST1_X.SRC".

Einige Erläuterungen zum Anwenderprogramm:

"\$include test1.def" dürfte klar sein. Falls "test1.def" in einem andern Verzeichnis liegt, muss der Pfad angegeben werden.

"XOB 16". In der Kaltstartroutine liegt der SASI-Befehl der RIO-Aufgabe. Das Symbol für die Nummer der Schnittstelle wurde bei der Konfigurierung der Master-Station im Fenster "Master Parameters" definiert. Das Symbol für den SASI-Text ist im Fenster "Generation Options" definiert. Dieses Fenster wird aus 'Network' - 'Generate Options...' erreicht.

"XOB 30": Dieser XOB ist optional, d.h. die Anlage läuft auch ohne diesen XOB 30. Der XOB wird hauptsächlich dann aufgerufen, wenn online Slave-Stationen zu- und weggeschaltet werden. Auch bei Syntax-Fehlern in den Konfigurations- und Definitions-DB wird dieser XOB aufgerufen. Für unser einfaches Beispiel zählen wir bloss wie oft der XOB 30 aufgerufen wurde. Wird der XOB 30 weggelassen, wird statt dem Aufruf des XOB das Error-Flag gesetzt und die Error-LED eingeschaltet. Weitergehende Anwendungen des XOB 13 werden im Kapitel 7: Diagnose, aufgezeigt.

"COB 0": Stellvertretend für ein Steuerprogramm wird in diesem Beispiel der Ausgang 55 der Master-Station zum Blinken gebracht. Dieser COB 0 hat mit der RIO-Aufgabe nichts zu tun und könnte auch weggelassen werden.

"COB 1": In diesem Programmteil werden die Kommunikationsbeziehungen der RIO-Aufgabe programmiert. Die 5 Befehlspakete entsprechen den Punkten 2 bis 6 der Aufgabenstellung und bedürfen kaum weitere Erklärungen, was die Programmierung betrifft. Die im Konfigurator editierten Ein- und Ausgänge, z.B. "st05_o8" werden in der eingebundenen TEST1.DEF-Datei Flags zugeordnet. Die Analogwerte werden in Register abgelegt. Diese Flags und Register sind das Medienabbild der RIO-Stationen. Die Auffrischung geschieht, für den Anwender unsichtbar, in der Firmware. Die Konfiguration und der SASI-Text werden der Konfigurationsdatei TEST1.SRC entnommen, welche für den Anwender auch nicht sichtbar ist. Diese Datei wird dazu gelinkt.

6.1.9 Inbetriebnahme

Die Konfigurierung des Programmspeichers der PCD2 muss Code, Text und Extension-Memory haben, z.B. die Standardkonfiguration mit:

7 k-Linien Code, 4 kBytes Text und 24 kBytes Extension-Memory.

Der Zusatzspeicher muss bestückt sein, da DBs im Adressbereich 4000 - 5999 zu adressieren sind.

Das vorliegende Programm belegt 47 Programmzeilen, 576 Textcharakter und 1032 Bytes Externer Speicher.

Nach dem Linken der beiden Dateien "TEST1_X.SRC" und "TEST1.SRC" und dem Laden in die PCD, kann in RUN geschaltet werden und das Netzwerk läuft. Alle programmierten Kommunikationsbeziehungen können durchgespielt werden.

Obwohl wir uns die Diagnosemöglichkeiten erst später genauer ansehen wollen, können im Debugger bereits jetzt einige interessante Diagnosepunkte hervorgehoben werden.

Es ist zum Ersten der Counter, welchen wir im XOB 30 verwenden um die Aufrufe dieses XOB zu zählen, aufgefrischt (refreshed) anzuzeigen. Wird online eine Station vom Bus getrennt oder deren Speisung ausgeschaltet, kann dies am Counter 100 verfolgt werden.

Ein weiterer interessanter Punkt ist der SASI-Text, welcher mit 'Display' - 'Text' - <100> im Klartext, d.h. mit den absoluten Adressen gezeigt wird, für unser Beispiel:

```
TEXT 100: "UART:38400,500,,0;MODE:RM1,R100,DB32;"
          "CONF:DB10;DIAG:F500,R500,DB4500"
```

Diese absoluten Adressen sind für die Verfolgung im Debugger wichtig. Für unsere Test-Anlage schauen wir uns den DB 4500 an:

```
'Display' - 'data-Block' - <4500> <SP> 'Element' <0> <SP> 'To' <14> <CR>.
```

Es erscheinen die 15 ersten Elemente des DB 4500. Es kann eingesehen werden, dass die Elemente 5 und 9 einen Wert enthalten. Diese Werte sind die Anzahl Telegramme, welche bis jetzt zwischen dem Master und den Slave-Stationen Nr. 5 und Nr. 9 ausgetauscht wurden. Die beiden für uns interessanten Elemente können mit

```
'Display' - 'data-Block' - <4500> <SP> 'Element' <5> (bzw. <9>) <SP>
<Refresh> <CR>
```

aufgefrischt angezeigt und verfolgt werden.

Ist die Master-Station in 'STOP' geht der Austausch der Telegramme weiter.

In der Liste der Definitionsdatei (Abschnitt 6.1.5) kann die Beziehung der Slave-Elemente zu den Flags des Medienabbildes eingesehen werden. In 'STOP' können die Flags für die Ausgänge mit dem Debugger ein- und ausgeschaltet werden. Da die Auffrischung auch in STOP weiter läuft, werden die entsprechenden Ausgänge den forcierten Flags folgen.

Anmerkung:

Optimale Informationen liefert die .LST-Datei, welche beim Assemblieren erzeugt werden kann. Die Datei ist zwar recht umfangreich, bietet aber den Vorteil, dass sämtliche Definitionen im symbolischen und absoluten Format sowie das Anwenderprogramm mit allen Kommentare in ein- und derselben Datei enthalten sind. (Leider fehlt der SASI-Text. Dieser muss separat ausgedruckt werden). Das Beispiel "TEST2_XA.LST" ist im Anhang gezeigt.

6.2 Weiteres Beispiel (PCD4.M445 + PCD2 + 2 PCD1.RIO)

Es soll an diesem erweiterten Beispiel der Einsatz einer Master-Station mit 2 CPU sowie der Einsatz von nicht-intelligenten und intelligenten Slave-Stationen gezeigt werden. Im weiteren soll auch das Übertragen von azyklischen Telegrammen kurz gezeigt werden.

6.2.1 Aufgabe

Es soll ein RIO-Netzwerk mit einer Master-, 2 nicht-intelligenten und einer intelligenten Slave-Station aufgebaut, programmiert und in Betrieb genommen werden.

Typen und Bestückung der 4 Stationen:

Master-Station:

Stations-Nummer:	1	
PCD-Typ:	PCD4.M445	
Bus-Modul:	PCD4.C340	
Anwenderprogramm:	CPU 0	
RIO:	CPU 1	
RIO-Schnittstelle:	Port 3 mit PCD7.F150	
I/O-Module:	Adr. 00 - 15	E100
	Adr. 16 - 31	E100 (4 BCD-Digits)
	Adr. 32 - 47	A400
	Adr. 48 - 63	A400

Slave-Stationen:

Stations-Nummer	5	(nicht-intelligente Slave-Station)
PCD-Typ	PCD1-RIO	
I/O-Module	Module 1	-
	Module 2	A400
	Module 3	A400
	Module 4	-

Stations-Nummer	9	(nicht-intelligente Slave-Station)
PCD-Typ	PCD1-RIO	
I/O-Module	Module 1	E100
	Module 2	E100 (2 BCD-Digits)
	Module 3	-
	Module 4	-

Stations-Nummer	12	(intelligente Slave-Station)
PCD-Typ	PCD2.M110 oder PCD2.M120	
Schnittstelle:	PCD7.F110 oder PCD7.F150, Port 1	
Displaymodul:	PCD2.F510 oder PCD2.F530	
I/O-Module:	Adr. 00 - 07	E110
	Adr. 16 - 23	E110 (2 BCD-Digits)
	Adr. 32 - 39	A400
	Adr. 48 - 55	A400
	Adr. 64 - 71	A400

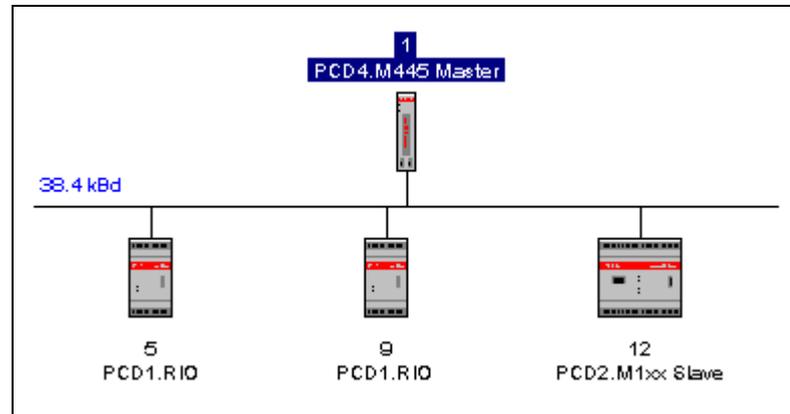
Programmieraufgabe:

- 1) Stellvertretend für ein Steuerprogramm, soll der Ausgang 63 in der Master-Station blinken.
- 2) Die Eingänge 4 - 7 der Master-Station sind an die Ausgänge 0 - 3 des Moduls 2 der Slave-Station Nr. 5 zu übertragen.
- 3) Die Eingänge 4 - 7 des Moduls 1 der Slave-Station Nr. 9 sind an die Ausgänge 32 - 35 der Master-Station zu übertragen.
- 4) Die Eingänge 0 - 3 des Moduls 1 der Slave-Station Nr. 9 sind an die Ausgänge 4 - 7 des Moduls 2 der Slave-Station Nr. 5 zu übertragen.
- 5) Die Eingänge 0 - 3 der Master-Station sind an die Ausgänge 32 - 35 der Slave-Station Nr. 12 zu übertragen.
- 6) Die Eingänge 16 - 23 der Slave-Station Nr. 12 sind an die Ausgänge 48 - 55 der Master-Station zu übertragen.
- 7) Die Eingänge 0 - 7 des Moduls 2 der Slave-Station Nr. 9 sind an die Ausgänge 64 -71 der Slave-Station Nr. 12 zu übertragen.
- 8) Die Eingänge 20 - 31 (3 BCD-Digits) der Master-Station sind am Displaymodul der Slave-Station Nr. 12 anzuzeigen.
- 9) Geht an der Master-Station der Eingang 15 von L → H, sollen die Eingänge 16 - 19 (1 BCD-Digit) an die Ausgänge 48 - 51 der Slave-Station Nr. 12 azyklisch übertragen werden.
- 10) Als Anwenderprogramm in der intelligenten Slave-Station Nr. 12, soll der Ausgang 55 blinken. (Hat mit der RIO-Aufgabe nichts zu tun, nur optischer Watchdog).

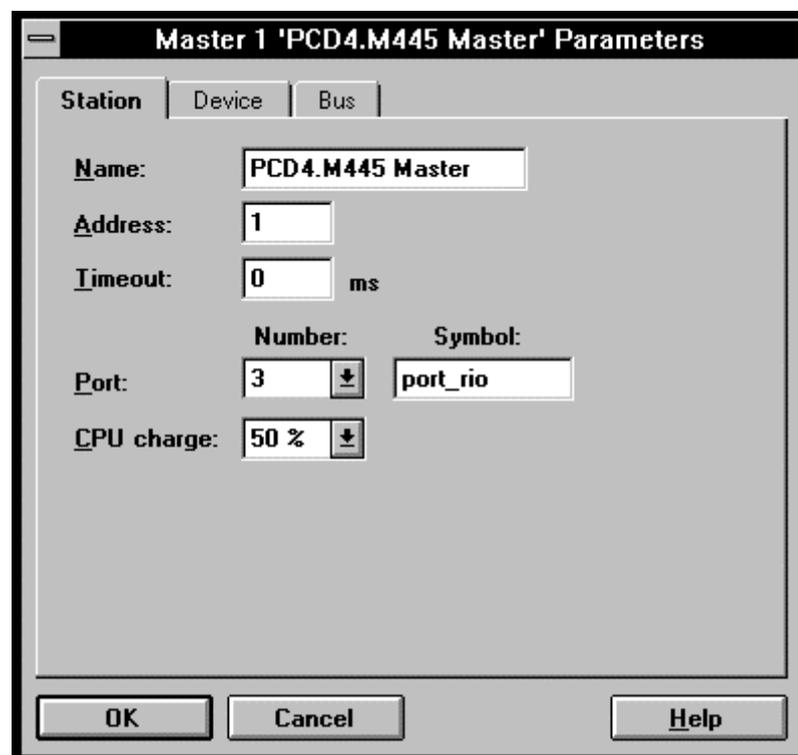
Dem Projekt geben wir den Namen TEST2.

6.2.2 Konfigurierung der Stationen

Die Konfigurierung der Stationen erfolgt in bekannter Weise. Weil die PCD2 72.8 kBd nicht unterstützt, kann das Netzwerk nur mit 38.4 kBd betrieben werden.



In der Master-Station läuft die gesamte RIO-Aufgabe auf CPU 1. Diese CPU 1 hat Zugang zu den Schnittstellen 2 und 3. Wir wählen Port 3. Dieses Port muss auf dem Busmodul PCD4.C340 mit einem RS 485 Schnittstellenmodul bestückt sein. (PCD7.F110 oder PCD7.F150). Das Timeout belassen wir auf dem Standardwert 0.



6.2.3 Konfigurierung der Slave-Stationen

Die Konfigurierung der nicht-intelligenten Slave-Stationen geschieht nach dem gleichen Muster, wie beim vorangehenden Beispiel.

Es geht nun darum, die intelligente Slave-Station zu editieren. Mit einem Doppelklick auf die Station Nr. 12 öffnen wir das Fenster "Slave Parameters". Nach dem Anklicken von 'Messages' erscheint ein leeres Feld mit dem Titel "Defined Messages". Nach dem Betätigen von 'Add...' erscheint das folgende Fenster:



Die Punkte 5), 6), 7) und 8) der Programmieraufgabe betreffen die Slave-Station Nr. 12. Diese 4 Aufgabenteile sind jetzt in 4 "Messages" zu formulieren.

Beginnen wir mit Punkt 5):

- 5) Die Eingänge 0 - 3 der Master-Station sind an die Ausgänge 32 - 35 der Slave-Station Nr. 12 zu übertragen.

Eine Message "Master Inputs → Slave Outputs" steht nicht zur Verfügung. Es können nur die Ein- und Ausgänge der Slave-Stationen direkt erreicht werden. Wir wählen

"Master Flags → Slave Outputs"

und tragen diese Message mit 'OK' ins Feld ein.

Edit Message

Function:

- Master Flags -> slave Flags
- Master Flags -> slave Outputs**
- Master Registers -> slave Registers
- Slave Flags -> master Flags
- Slave Inputs -> master Flags
- Slave Registers -> master Registers

Description:

Copy the master flags to the slave outputs

Media Definition

Number of Media to Transfer: 4

Position: 0 **Symbol:** st12_o32

Media Map

First Master Media Type:	Flag
First Master Media Address:	40
First Slave Media Type:	Output
First Slave Media Address:	32

Bei "Number of Media" geben <4> ein, da wir auf 4 aufeinanderfolgende Ausgänge übertragen wollen.

Bei "First Master Media Address" müssen wir 4 aufeinanderfolgende Flags der Master-Station angeben. Dort werden die 4 Eingänge abgebildet. Es ist strikte darauf zu achten, dass sich die Bereiche der Flags nicht überschneiden! Wir wählen Flag 40 (bis 43).

Bei "First Slave Media Address" geben wir <32> ein. Dieses ist die unterste der 4 Ausgangsadressen.

Bei "Symbol" bezeichnen wir alle Ausgänge mit einem symbolischen Namen. Ist alles ausgefüllt, bestätigen wir mit 'OK'.

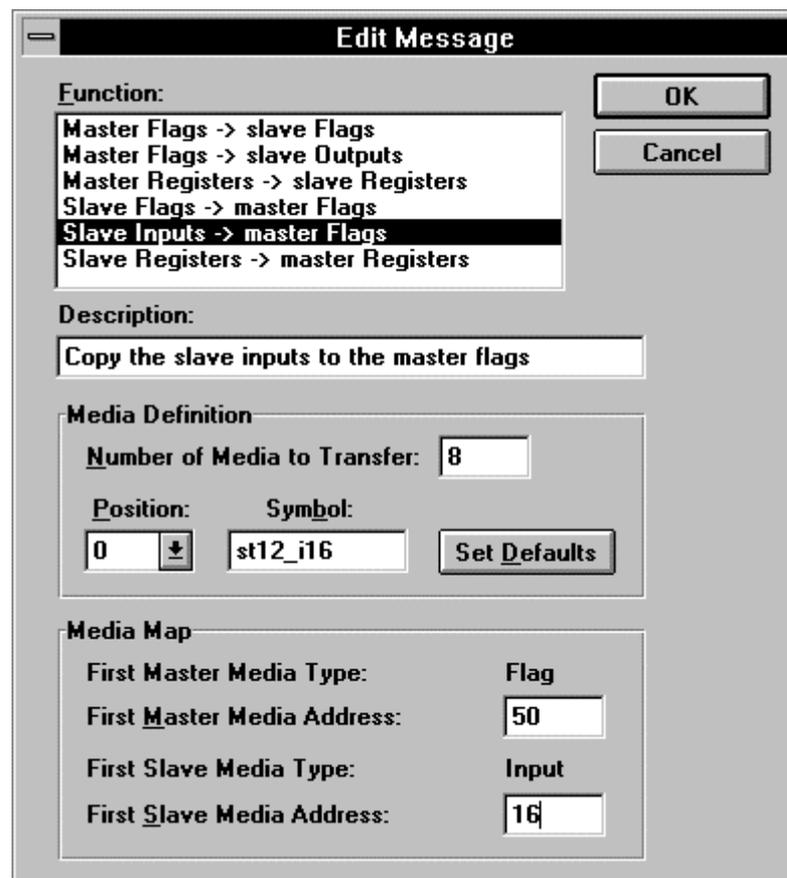
Es folgt Punkt 6):

- 6) Die Eingänge 16 - 23 der Slave-Station Nr. 12 sind an die Ausgänge 48 - 55 der Master-Station zu übertragen.

Eine Message "Slave inputs → Master outputs" steht nicht zur Verfügung. Es können nur die Ein- und Ausgänge der Slave-Stationen direkt erreicht werden. Wir wählen

"Slave inputs → Master flags"

und tragen diese Message mit 'OK' ins Feld ein.



Bei "Number of Media" geben <8> ein, da wir 8 aufeinanderfolgende Eingänge übertragen wollen.

Bei "First Slave Media Address" geben wir <16> ein. Dieses ist die unterste der 8 Eingangsadressen.

Bei "First Master Media Address" müssen wir 8 aufeinanderfolgende Flags der Master-Station angeben. Dort werden die 8 Eingänge abgebildet. Es ist strikte darauf zu achten, dass sich die Bereiche der Flags nicht überschneiden! Wir wählen Flag 50 (bis 57).

Bei "Symbol" bezeichnen wir alle Eingänge mit einem symbolischen Namen. Ist alles ausgefüllt, bestätigen wir mit 'OK'.

Wir machen mit Punkt 7) weiter:

- 7) Die Eingänge 0 - 7 des Moduls 2 der Slave-Station Nr. 9 sind an die Ausgänge 64 -71 der Slave-Station Nr. 12 zu übertragen.

Eine Message zum direkten Übertragen von Elementen einer Slave-Station an eine andere Slave-Station ist nicht vorgesehen. Die Eingänge der Slave-Station Nr. 9 sind bei der Konfigurierung dieser Station bereits im Medienabbild enthalten. Es muss jetzt noch die Beziehung zwischen den Ausgängen der intelligenten Slave-Station und dem Medienabbild hergestellt werden. Wir wählen dazu die Flags 60 - 67. Bei "First Master Media Address" tragen wir <60> ein. Bei "First Slave Media Address" tragen wir <64> für die tiefste der Ausgangsadressen ein.

Das Fenster füllen wir folgendermassen aus:

Edit Message

Function:
 Master Flags -> slave Flags
Master Flags -> slave Outputs
 Master Registers -> slave Registers
 Slave Flags -> master Flags
 Slave Inputs -> master Flags
 Slave Registers -> master Registers

Description:
 Copy the master flags to the slave outputs

Media Definition

Number of Media to Transfer: 8

Position: 0 **Symbol:** st12_o64 **Set Defaults**

Media Map

First Master Media Type:	Flag
First Master Media Address:	60
First Slave Media Type:	Output
First Slave Media Address:	64

... und bestätigen mit 'OK'.

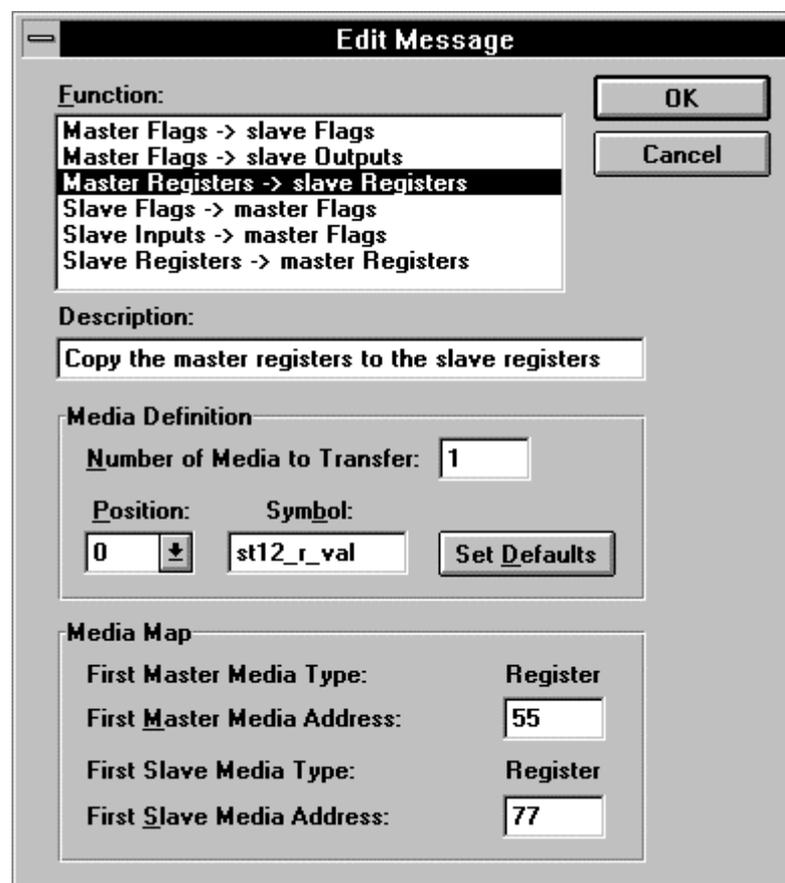
Es bleibt noch Punkt 8):

- 8) Die Eingänge 20 - 31 (3 BCD-Digits) der Master-Station sind am Displaymodul der Slave-Station Nr. 12 anzuzeigen.

Es könnten nun die 12 Eingänge von der Master- zur Slave-Station übertragen und im Anwenderprogramm der (intelligenten) Slave-Station in ein Register geladen und zur Anzeige gebracht werden.

Wir wählen den andern Weg und laden den 3-Digitwert in der Master-Station in ein Register und übertragen diesen via das RIO-Netzwerk zur Slave-Station.

Die Konfigurierung ist die folgende:



Mit 'OK' werden die Eingaben bestätigt.

Das Fenster "Slave Parameters" sieht nun folgendermassen aus:



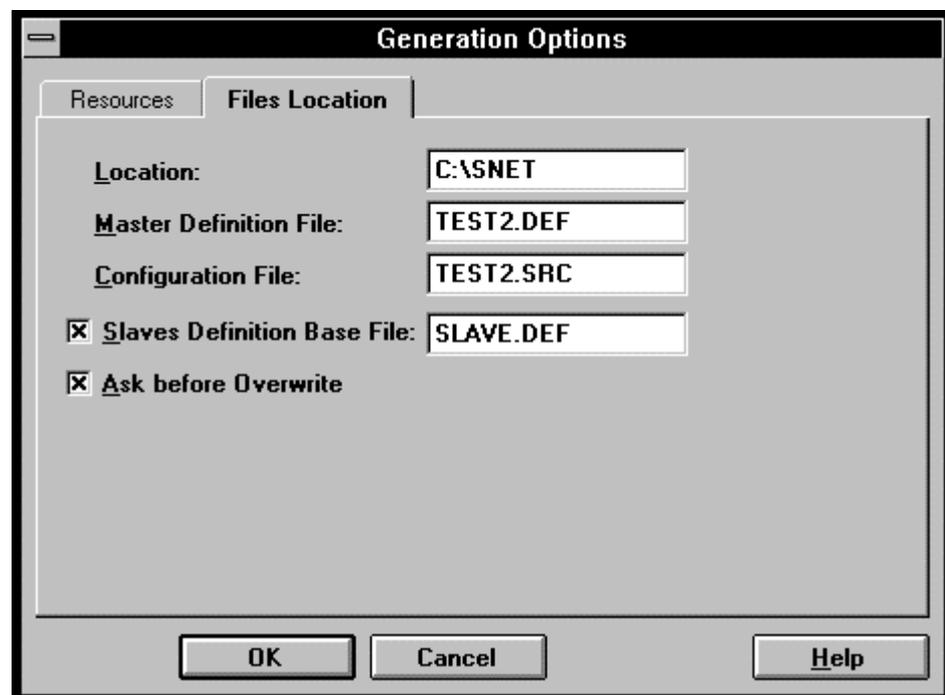
Wir bestätigen mit 'OK'.

6.2.4 Generieren der Hilfsdateien

Dies geschieht genau so, wie dies in Abschnitt 6.1.4 beschrieben ist. Es ist darauf zu achten, dass die Diagnose-Elemente (Flags und Register) nicht mit den Elementen des Medienabbildes, sowohl der nicht-intelligenten als auch der intelligenten Stationen kollidieren.

Wir rufen 'Generate Options...' auf. Die Voreinstellungen im Fenster 'Resources' belassen wir.

Im nächsten Fenster 'Files Location' geben wir das Verzeichnis und die Dateinamen für .DEF und .SRC ein. Da eine intelligente Station im Netzwerk editiert wurde, ist die Eintragung "SLAVES.DEF" bereits gemacht. Bestätigen mit 'OK'.



Mit 'Network' - 'Generate...' erstellen wir in bekannter Weise die Dateien "TEST1.DEF" und "TEST1.SRC".

Im vorliegenden Fall, d.h. wenn im Netzwerk eine oder mehrere intelligente Slave-Stationen liegen, wird für jede dieser Stationen eine "SLAVE<Stat. Nr.>. DEF" erstellt. Für unser Beispiel mit einer intelligenten Slave-Station Nr. 12, wird die Datei "SLAVE12.DEF" erzeugt. Die Datei enthält alle Definitionen (EQU) aller Elemente dieser Station.

Das Ganze speichern wir mit 'Network' - 'Save'.

Bevor wir mit der Editierung des Anwenderprogramms beginnen, drucken wir die 3 Dateien TEST2.DEF, SLAVE12.DEF und TEST2.SRC aus.

6.2.5 Die Definitionsdatei (TEST2.DEF)

```

; Master 1 (PCD4.M445 Master)
;-----

port_rio EQU    3           ;RIO: Port number

; Slave 5 (PCD1.RIO)
;-----

; Module 2 (Transistor outputs A4xx)

st05_o8 EQU    F 2064      ;Slave 5, mod 2, dig output 0
st05_o9 EQU    F 2065      ;Slave 5, mod 2, dig output 1
st05_o10 EQU   F 2066      ;Slave 5, mod 2, dig output 2
st05_o11 EQU   F 2067      ;Slave 5, mod 2, dig output 3
st05_o12 EQU   F 2068      ;Slave 5, mod 2, dig output 4
st05_o13 EQU   F 2069      ;Slave 5, mod 2, dig output 5
st05_o14 EQU   F 2070      ;Slave 5, mod 2, dig output 6
st05_o15 EQU   F 2071      ;Slave 5, mod 2, dig output 7

; Module 3 (Transistor outputs A4xx)

st05_o16 EQU   F 2072      ;Slave 5, mod 3, dig output 0
st05_o17 EQU   F 2073      ;Slave 5, mod 3, dig output 1
st05_o18 EQU   F 2074      ;Slave 5, mod 3, dig output 2
st05_o19 EQU   F 2075      ;Slave 5, mod 3, dig output 3
st05_o20 EQU   F 2076      ;Slave 5, mod 3, dig output 4
st05_o21 EQU   F 2077      ;Slave 5, mod 3, dig output 5
st05_o22 EQU   F 2078      ;Slave 5, mod 3, dig output 6
st05_o23 EQU   F 2079      ;Slave 5, mod 3, dig output 7

; Slave 9 (PCD1.RIO)
;-----

; Module 1 (Digital inputs E1xx)

st09_i0 EQU    F 96        ;Slave 9, mod 1, dig input 0
st09_i1 EQU    F 97        ;Slave 9, mod 1, dig input 1
st09_i2 EQU    F 98        ;Slave 9, mod 1, dig input 2
st09_i3 EQU    F 99        ;Slave 9, mod 1, dig input 3
st09_i4 EQU    F 100       ;Slave 9, mod 1, dig input 4
st09_i5 EQU    F 101       ;Slave 9, mod 1, dig input 5
st09_i6 EQU    F 102       ;Slave 9, mod 1, dig input 6
st09_i7 EQU    F 103       ;Slave 9, mod 1, dig input 7

; Module 2 (Digital inputs E1xx)

st09_i8 EQU    F 104       ;Slave 9, mod 2, dig input 0
st09_i9 EQU    F 105       ;Slave 9, mod 2, dig input 1
st09_i10 EQU   F 106       ;Slave 9, mod 2, dig input 2
st09_i11 EQU   F 107       ;Slave 9, mod 2, dig input 3
st09_i12 EQU   F 108       ;Slave 9, mod 2, dig input 4
st09_i13 EQU   F 109       ;Slave 9, mod 2, dig input 5
st09_i14 EQU   F 110       ;Slave 9, mod 2, dig input 6
st09_i15 EQU   F 111       ;Slave 9, mod 2, dig input 7

```

```

; Slave 12 (PCD2.Mlxx Slave)
;-----

st12_o32      EQU    F 40  ;Slave 12, message media
st12_o33      EQU    F 41  ;Slave 12, message media
st12_o34      EQU    F 42  ;Slave 12, message media
st12_o35      EQU    F 43  ;Slave 12, message media

st12_i16      EQU    F 50  ;Slave 12, message media
st12_i17      EQU    F 51  ;Slave 12, message media
st12_i18      EQU    F 52  ;Slave 12, message media
st12_i19      EQU    F 53  ;Slave 12, message media
st12_i20      EQU    F 54  ;Slave 12, message media
st12_i21      EQU    F 55  ;Slave 12, message media
st12_i22      EQU    F 56  ;Slave 12, message media
st12_i23      EQU    F 57  ;Slave 12, message media

st12_o64      EQU    F 60  ;Slave 12, message media
st12_o65      EQU    F 61  ;Slave 12, message media
st12_o66      EQU    F 62  ;Slave 12, message media
st12_o67      EQU    F 63  ;Slave 12, message media
st12_o68      EQU    F 64  ;Slave 12, message media
st12_o69      EQU    F 65  ;Slave 12, message media
st12_o70      EQU    F 66  ;Slave 12, message media
st12_o71      EQU    F 67  ;Slave 12, message media

st12_r_val    EQU    R 55  ;Slave 12, message media

; Generation Options

db_cfg        EQU    10    ;RIO: First Configuration DB
db_msg        EQU    32    ;RIO: First Messages DB
db_diag       EQU    4500  ;RIO: First Diagnostic DB
f_diag        EQU    500   ;RIO: First Diagnostic Flag
r_diag        EQU    500   ;RIO: First Diagnostic Reg.
txt_rio       EQU    100   ;RIO: SASI Text Number
r_dst         EQU    100   ;RIO: Partner Stn Nb Register
    
```

6.2.6 Die Definitionsdatei der intelligenten Slave-Station (SLAVE12.DEF)

```

; Slave 12 (PCD2.M1xx Slave)
;-----

st12_o32      EQU    O 32      ;Slave 12, message media
st12_o33      EQU    O 33      ;Slave 12, message media
st12_o34      EQU    O 34      ;Slave 12, message media
st12_o35      EQU    O 35      ;Slave 12, message media

st12_i16      EQU    I 16      ;Slave 12, message media
st12_i17      EQU    I 17      ;Slave 12, message media
st12_i18      EQU    I 18      ;Slave 12, message media
st12_i19      EQU    I 19      ;Slave 12, message media
st12_i20      EQU    I 20      ;Slave 12, message media
st12_i21      EQU    I 21      ;Slave 12, message media
st12_i22      EQU    I 22      ;Slave 12, message media
st12_i23      EQU    I 23      ;Slave 12, message media

st12_o64      EQU    O 64      ;Slave 12, message media
st12_o65      EQU    O 65      ;Slave 12, message media
st12_o66      EQU    O 66      ;Slave 12, message media
st12_o67      EQU    O 67      ;Slave 12, message media
st12_o68      EQU    O 68      ;Slave 12, message media
st12_o69      EQU    O 69      ;Slave 12, message media
st12_o70      EQU    O 70      ;Slave 12, message media
st12_o71      EQU    O 71      ;Slave 12, message media

st12_r_val    EQU    R 77      ;Slave 12, message media

```

6.2.7 Die Konfigurationsdatei (TEST2.SRC)

```
$INCLUDE TEST2.DEF

; Configuration of the stations

DB db_cfg+0 [8] 5, 1000, 0, 2000H, 2000H, 0, 0, 1
DB db_cfg+1 [8] 9, 0, 1000H, 1000H, 0, 0, 0, 1
DB db_cfg+2 [8] -1, -1, -1, -1, -1, -1, -1, -1

; Messages of the stations

DB db_msg+0 [7] 5, 8, 16, 2064, 64
DB db_msg+1 [7] 9, 1, 16, 0, 96
DB db_msg+2 [7] 12, 9, 4, 40, 32
DB db_msg+3 [7] 12, 2, 8, 16, 50
DB db_msg+4 [7] 12, 9, 8, 60, 64
DB db_msg+5 [7] 12, 10, 1, 55, 77
DB db_msg+6 [7] -1, -1, -1, -1, -1

; Diagnostics

DB db_diag+0 [127]
DB db_diag+1 [127]

; SASI Text

TEXT txt_rio "UART:38400,500,,0"
             "MODE:RM1,R",r_dst,"DB",db_msg,";"
             "CONF:DB",db_cfg,";"
             "DIAG:F",f_diag,"R",r_diag,"DB",db_diag,";"
```

6.2.8 Das Anwenderprogramm für die Master-Station (TEST2_X.SRC für CPU 1, TEST_Y.SRC für CPU 0)

```

;      Programm für CPU 1      (TEST2_X)
;      *****

$include test2.def

      xob          16
      sasi         port_rio
                   txt_rio
      exob

;      -----

      xob          30
      inc          c      100
      exob

;      -----

      cob          0      ; RIO-Kommunikationsbeziehungen
                   0

      sth          i      4      ; Aufgabe Punkt 2)
      out          st05_o8
      sth          i      5
      out          st05_o9
      sth          i      6
      out          st05_o10
      sth          i      7
      out          st05_o11

      sth          st09_i4   ; Aufgabe Punkt 3)
      out          o      32
      sth          st09_i5
      out          o      33
      sth          st09_i6
      out          o      34
      sth          st09_i7
      out          o      35

      sth          st09_i0   ; Aufgabe Punkt 4)
      out          st05_o12
      sth          st09_i1
      out          st05_o13
      sth          st09_i2
      out          st05_o14
      sth          st09_i3
      out          st05_o15

      sth          i      0      ; Aufgabe Punkt 5)
      out          st12_o32
      sth          i      1
      out          st12_o33
      sth          i      2
      out          st12_o34
      sth          i      3
      out          st12_o35

```

```

sth          st12_i16      ; Aufgabe Punkt 6)
out          o          48
sth          st12_i17
out          o          49
sth          st12_i18
out          o          50
sth          st12_i19
out          o          51
sth          st12_i20
out          o          52
sth          st12_i21
out          o          53
sth          st12_i22
out          o          54
sth          st12_i23
out          o          55

```

```

sth          st09_i8      ; Aufgabe Punkt 7)
out          st12_o64
sth          st09_i9
out          st12_o65
sth          st09_i10
out          st12_o66
sth          st09_i11
out          st12_o67
sth          st09_i12
out          st12_o68
sth          st09_i13
out          st12_o69
sth          st09_i14
out          st12_o70
sth          st09_i15
out          st12_o71

```

```

digi          3          ; Aufgabe Punkt 8)
              i          20
              st12_r_val

```

```

sth          i          15      ; Aufgabe Punkt 9)
anl          f          f_diag+3 ; Transmitter Busy
dyn          f          0
cpb          h          0

```

ecob

;

```

pb          0          ; (Aufgabe Punkt 9))

```

```

ldl          r          r_dst
              12          ; Stations-Nummer

```

```

stxm          3          ; Port
              4          ; Anzahl Medien
              i          16      ; Quelle
              o          48      ; Ziel

```

epb

```

;      Programm für CPU 0      (TEST2_Y)
;      *****

      cob          0          ; Anwenderprogramm
                        0          ; für Prozess

      stl      t      0          ; Aufgabe Punkt 1)
      ldl      t      0
                        5
      com      o      63

      ecob

```

Einige Erläuterungen zu den Anwenderprogrammen.

Programm für CPU 1:

In der CPU 1 wird die gesamte RIO-Aufgabe behandelt. d.h. es wird die Assignierung der RIO-Aufgabe und die Kommunikationsbeziehungen zwischen dem Master und den nicht-intelligenten und den intelligenten Slave-Stationen editiert. Auch eventuelle azyklische Telegramme werden in diesem Anwenderprogramm behandelt.

Für die intelligenten Slave-Stationen sind je eigene Programme zu schreiben. Diese Programme müssen zur Erfüllung der RIO-Belange zum Mindesten die Assignierung der RIO-Schnittstelle enthalten. Auch muss bei allen diesen Stationen die Stations-Nummer konfiguriert werden.

Die Dateien mit den spezifischen RIO-Belangen wurde bereits durch den Konfigurator erstellt und wurden bereits gezeigt. Die Datei "TEST2.DEF" wird mit "\$include" ins Anwenderprogramm eingefügt, die andere Datei, welche der Konfigurator erzeugt hat, "TEST2.SRC", wird zum Anwenderprogramm gelinkt. Das Anwenderprogramm muss deshalb einen andern Namen als das Konfigurationsprogramm haben. In unserem Fall nennen wir das Anwenderprogramm für die CPU 1 "TEST2_X.SRC".

"\$include test2.def" dürfte klar sein. Falls "test2.def" in einem andern Verzeichnis liegt, muss der Pfad angegeben werden.

"XOB 16". In der Kaltstartroutine liegt der SASI-Befehl der RIO-Aufgabe. Das Symbol für die Nummer der Schnittstelle wurde bei der Konfigurierung der Master-Station im Fenster "Master Parameters" definiert. Das Symbol für den SASI-Text ist im Fenster "Generation Options" definiert. Dieses Fenster wird aus 'Network' - 'Generate Options...' erreicht.

"XOB 30": Dieser XOB ist optional, d.h. die Anlage läuft auch ohne diesen XOB 30. Der XOB wird hauptsächlich aufgerufen, wenn online Slave-Stationen zu- und weggeschaltet werden. Auch bei Syntax-Fehlern in den Konfigurations- und Definitions-DB wird dieser XOB aufgerufen. Für unser einfaches Beispiel zählen wir bloss wie oft der XOB 30 aufgerufen wurde. Wird der XOB 30 weggelassen, wird statt dem Aufruf des XOB das Error-Flag gesetzt und die Error-LED eingeschaltet. Weitergehende Anwendungen des XOB 13 werden im Kapitel 7: Diagnose, aufgezeigt.

"COB 1": In diesem Programmteil werden die Kommunikationsbeziehungen der RIO-Aufgabe programmiert. Die 8 Befehlspakete entsprechen den Punkten 2 bis 9 der Aufgabenstellung und bedürfen kaum weitere Erklärungen, was die Programmierung betrifft. Die im Konfigurator editierten Ein- und Ausgänge, z.B. "st05_o10" werden in der eingebundenen TEST2.DEF-Datei Flags zugeordnet. Die Analogwerte werden in Register abgelegt. Diese Flags und Register sind das Medienabbild der RIO-Stationen. Die Auffrischung geschieht, für den Anwender unsichtbar, in der Firmware. Die Konfiguration und der SASI-Text werden der Konfigurationsdatei TEST2.SRC entnommen, welche für den Anwender auch nicht sichtbar ist. Diese Datei wird dazu gelinkt.

Programm für CPU 0:

Eigentliches Anwenderprogramm für die Prozess-Steuerung. Zugang zu den Schnittstellen 0 und 1, nicht jedoch 2 und 3, wobei Port 3 im vorliegenden Fall sowieso durch die RIO-Aufgabe belegt ist.

Das Anwenderprogramm in der CPU 0 wird in keiner Weise durch die RIO-Aufgabe in CPU 1 beeinflusst, weder durch das zyklische Auffrischen des Medienabbildes, noch durch die zyklisch und azyklisch durchgeführten Kommunikationsbeziehungen.

6.2.9 Das Anwenderprogramm für die intelligente Slave-Station (TEST2_12.SRC)

```

;      Programm für Slave-Station 12 (PCD2)      (TEST2_12)
;      *****
      xob          16

      sasi         1
                  1

      $sasi
      text 1 "uart:38400;mode:ss1;diag:f100,r100"
      $endsasi

      exob
;      -----
      cob          0
                  0

      stl   t      0      ; Aufgabe Punkt 10)
      ldl   t      0
                  5
      com   o      55

      dsp   r      77      ; Teil-Aufgabe Punkt 8)

      ecob

```

Für jede intelligenten Slave-Station ist ein eigenes Programm zu schreiben. Diese Programme müssen zur Erfüllung der RIO-Belange zum Mindesten die Assignierung der RIO-Schnittstelle enthalten. Die Assignierung des RIO-Ports hat im Modus "SS1" (Slave S-Bus) zu erfolgen: z.B.:

"UART:38400;MODE:SS1;DIAG:F100,R100"

Die Baudrate muss mit dem RIO-Netzwerk übereinstimmen.

Auch muss bei allen diesen Stationen die Stations-Nummer konfiguriert werden.

Dies geschieht im PG3 mit "SAIA PCD CONFIGURATION" - "S-Bus communications":

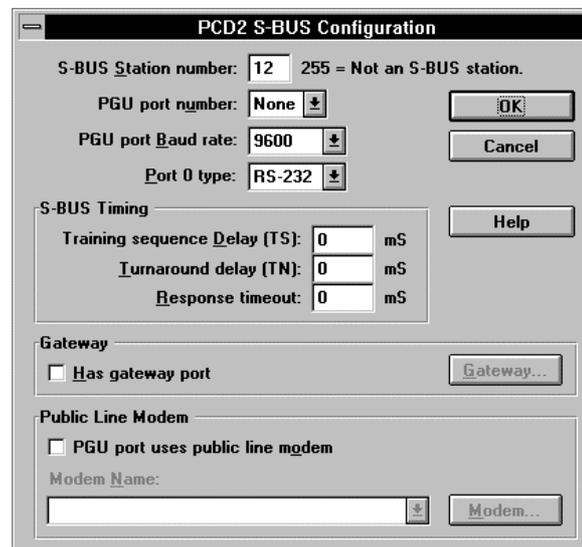
```

SAIA PCD OFFLINE CONFIGURATOR V2.0                                S-BUS COMMUNICATIONS
  Station number . . . . . (0..254)  12_      (255 = No S-BUS support)
+ S-BUS PGU PORT CONFIGURATION - FOR CPU TYPE PCD2 -----+
+
+-----+
PGU Port Allocation:      | CPU | PGU PORT | TYPE |
+-----+-----+-----+
PCD2: Port 0 is software | 0 | None | RS 232 |
selectable RS 232/RS 485, +-----+
for ports 1..3 the TYPE
is hardware dependent.

PGU port baud rate . (110..38400)  9600      +----- NOTE -----+
S-BUS mode . . . . (Break/Parity)  Parity    | Changes do not take effect |
PGU via Public Line Modem . . . .  No       | until a "CONFIGURE S-BUS" |
                                       | operation is done from the |
S-BUS TIMING (0=default):              | "Up/download" menu, or until|
TS delay in mS   (0, or 1..15000)  0____   | "SDNLD /S" is executed from |
Timeout in mS   (0, or 1..15000)  0____   | the DOS prompt.            |
TN delay in mS   (0, or 1..15000)  0____   +-----+
+-----+
+
Enter decimal value, ARROW moves cursor, ESC or ENTER accepts.
F1=Help
    
```

Die Konfiguration muss anschliessend im Menü "Up/download" - "CONFIGURE S-BUS" geladen werden.

Die Konfiguration kann auch im PG4 im Online-Konfigurator vorgenommen werden:



6.2.10 Inbetriebnahme

Die Konfigurierung des Programmspeichers der CPU 1 der PCD4 muss Code, Text und Extension-Memory haben. Die CPU 0 ist gemäss den Anforderungen des Anwenderprogramms zu konfigurieren.

Nach dem Linken der beiden Dateien "TEST2_X.SRC" und "TEST2.SRC" und dem Laden in die CPU 1 der PCD4 und dem Laden des Anwenderprogramms in die CPU 0, können beide CPU in RUN geschaltet werden. Ist auch das Programm in die intelligente Slave-Station Nr. 12 geladen und auch diese Station in RUN, laufen sowohl die Anwenderprogramme und auch das RIO-Netzwerk. Alle programmierten Kommunikationsbeziehungen können durchgespielt werden.

Obwohl wir uns die Diagnosemöglichkeiten erst später genauer ansehen wollen, können im Debugger bereits jetzt einige interessante Diagnosepunkte hervorgehoben werden.

Es ist zum Ersten der Counter, welchen wir im XOB 30 verwenden um die Aufrufe dieses XOB zu zählen, aufgefrischt (refreshed) anzuzeigen. Wird online eine Station vom Bus getrennt oder die Speisung ausgeschaltet, kann dies am Counter 100 verfolgt werden.

Ein weiterer interessanter Punkt ist der SASI-Text, welcher mit 'Display' - 'Text' - <100> im Klartext, d.h. mit den absoluten Adressen gezeigt wird, für unser Beispiel:

```
TEXT 100:    "UART:38400,500,,0;MODE:RM1,R100,DB32;"
             "CONF:DB10;DIAG:F500,R500,DB4500"
```

Diese absoluten Adressen sind für die Verfolgung im Debugger wichtig. Für unsere Test-Anlage schauen wir uns den DB 4500 an:

```
'Display' - 'data-Block' - <4500> <SP> 'Element' <0> <SP> 'To' <20> <CR>.
```

Es erscheinen die 21 ersten Elemente des DB 4500. Es kann eingesehen werden, dass die Elemente 5, 9 und 12 einen Wert enthalten. Diese Werte sind die Anzahl Telegramme, welche bis jetzt zwischen dem Master und den Slave-Stationen Nr. 5, Nr. 9 und Nr. 12 ausgetauscht wurden. Die 3 für uns interessanten Elemente können mit

```
'Display' - 'data-Block' - <4500> <SP> 'Element' <5> (bzw. <9> oder <12>) <SP> <Refresh> <CR>
```

aufgefrischt angezeigt und verfolgt werden.

Ist die Master-Station in 'STOP' geht der Austausch der Telegramme weiter.

In der Liste der Definitionsdatei (Abschnitt 6.2.6) kann die Beziehung der Slave-Elemente zu den Flags des Medienabbildes eingesehen werden. In 'STOP' können die Flags für die Ausgänge mit dem Debugger ein- und ausgeschaltet werden. Da die Auffrischung auch in STOP weiter läuft, werden die entsprechenden Ausgänge den forcierten Flags folgen.

Es kann z.B. auch via den Debugger ein max. 6-stelliger dezimaler Wert ins Register 55 eingeschrieben werden. Dieser Wert erscheint unmittelbar nach der Bestätigung mit <CR> am Displaymodul des Slaves Nr. 12 (PCD2), da auch die konfigurierten Register im Medienabbild zyklisch aufgefrischt werden. (Die PCD2 muss dazu in RUN sein, da der Wert im Anwenderprogramm zum Display geschickt wird). Wird nur das Empfangsregister in PCD 2 betrachtet (für unser Beispiel R 77), so wird dieses aufgefrischt, auch wenn die CPU der PCD 2 in STOP ist. Es muss bei den intelligenten Slave-Stationen nur einmal der SASI-Befehl abgearbeitet werden (auch in TRACE möglich), damit der RIO-Kanal geöffnet und aufrecht erhalten bleibt.

Anmerkung:

Optimale Informationen liefert die .LST-Datei, welche beim Assemblieren erzeugt werden kann. Die Datei ist zwar recht umfangreich, bietet aber den Vorteil, dass sämtliche Definitionen im symbolischen und absoluten Format sowie das Anwenderprogramm mit allen Kommentare in ein- und derselben Datei enthalten sind. (Leider fehlt der SASI-Text. Dieser muss separat ausgedruckt werden). Das Beispiel "TEST2_XA.LST" ist im Anhang gezeigt.

6.3 Häufige Fehlerursachen

Abgesehen von Schwierigkeiten mit der RS 485-Busleitung, welche sehr sorgfältig gemäss den Angaben im Abschnitt 3.5 und den Handbüchern zum "SAIA® S-Bus" und den "Installations-Komponenten für RS 485-Netzwerke" verlegt und angeschlossen werden muss, können viele kleinere Fehler und Unachtsamkeiten die Funktion des RIO-Netzwerk verunmöglichen oder beeinträchtigen.

Die nachfolgende Liste enthält Fehler, welche bei der Erstellung der beiden Beispiele im vorliegenden Handbuch aufgetreten sind. Die Liste ist sicher nicht vollständig und kann erweitert werden.

Station	Fehlermöglichkeit
Master	<ul style="list-style-type: none"> ● Falsche Schnittstelle assigniert ● RIO-Buskabel nicht oder auf dem falschen Port angeschlossen ● Falsche Firmware ● Programm auf die falsche CPU geladen (PCD4.M445) ● CPU nicht in RUN (PCD eingeschaltet mit angeschlossenem Programmiergerät) ● falscher Schnittstellen-Typ (nicht RS 485)
Slave (nicht intelligent)	<ul style="list-style-type: none"> ● Buskabel nicht eingesteckt bzw. unterbrochen ● Speisung einzelner Stationen nicht da oder unterbrochen (Schlaufung auf Pins +/+ und -/!) ● Falsche Stationsnummer eingestellt ● Falsche Baudrate eingestellt
Slave (intelligent)	<ul style="list-style-type: none"> ● Stationsnummer nicht konfiguriert ● Falsche Stationsnummer ● SASI-Befehl nicht abgearbeitet ● falscher Modus assigniert ● falsche Schnittstelle assigniert ● falsche Schnittstelle angeschlossen ● falsches Schnittstellenmodul (nicht RS 485) ● falsche Firmware

7. Diagnose

7.1 Überwachung der Anwesenheit der Master-Station

Im Konfigurator kann bei der Konfigurierung der Master-Station eine Überwachungszeit (Timeout) von z.B. 500 ms eingegeben werden (siehe Abschnitt 5.3.3). Diese Überwachungszeit wird von den Slaves dauernd getestet. Sollte die Überwachungszeit abgelaufen sein, d.h. der Master ist länger als die Überwachungszeit nicht da, wird ein Flag (F_diag+19) zurückgesetzt und kann im Anwenderprogramm ausgewertet werden. (siehe Abschnitt 7.3).

Eine weitere, für die Praxis wohl wichtigere Funktion dieses Timeouts ist, dass nach Ablauf dieses Timeouts alle Ausgänge der nicht-intelligenten Slave-Stationen zurückgeschaltet und die Register mit den Werten der Analogmodule auf 0 gesetzt werden, aber nur dann, wenn auch ein Timeout bei den Slave-Stationen definiert wurde. (siehe nächster Abschnitt)

Beim Wiederkommen des Masters, wird das Abbild der Medien wieder übertragen.

Wird keine Überwachung des Masters gewünscht, wird bei der Konfigurierung 0 eingeschrieben. Es werden in diesem Fall die S-Bus Standardwerte eingesetzt:

für 9'600 Bd	→	250 ms
für 19'200 Bd	→	200 ms
für 38'400 Bd	→	100 ms
für 76'800 Bd	→	50 ms

Die Timeout-Werte sind 0 bzw. 50 - 3'600'000 ms (1 Stunde).

Soll die Anwesenheit des Masters von einer intelligenten Slave-Station aus festgestellt werden, ist dies via das Anwenderprogramm dieser Station zu programmieren, z.B. durch die Auswertung des Receiver Busy Flags (RBSY). Es wäre auch denkbar, von der Master-Station aus ein Flag in der Slave-Station dauernd zu komplementieren und im Anwenderprogramm auszuwerten. (Watchdog). Ein Zurückschalten aller oder bestimmter Ausgänge hat auch im Anwenderprogramm zu erfolgen.

7.2 Überwachung der Anwesenheit der Slave-Stationen (XOB 30)

Im Konfigurator kann bei der Konfigurierung der Slave-Stationen eine Überwachungszeit (Timeout) von z.B. 500 ms eingegeben werden (siehe Abschnitt 5.3.4). Findet für eine Zeit länger als die definierte Überwachungszeit keine RIO-Kommunikation zwischen dem Master und diesem Slave statt, werden alle Ausgänge dieser Station ausgeschaltet und die Register mit den Werten der Analogmodule auf 0 gesetzt.

Dieses Verhalten kann an den beiden Beispielen TEST1 und TEST2 nachvollzogen werden, indem an der Slave-Station Nr. 5 die Ausgänge eingeschaltet und dann der Busstecker heraus gezogen wird. Nach dem Ablauf des Timeouts werden die Ausgänge ausschalten und nach dem Wiedereinstecken den Zustand von vorher einnehmen.

Im weiteren wird nach jeder vom Master zu einem Slave gesendeten Meldung die Verbindung zum Slave getestet. Wird der Test vom Slave nicht positiv beantwortet, ruft die CPU des Masters den XOB 30 auf. Dies ist im wesentlichen dann der Fall, wenn online eine Station vom Netzwerk entfernt oder zugeschaltet wird. Dies kann an den beiden Beispielen TEST1 und TEST2 verfolgt werden.

Auch wenn sich die Master-Station im Status STOP befindet, läuft die RIO-Kommunikation weiter und das Abbild der Medien wird weiter aufgefrischt. Werden während dieser Zeit Stationen dazu- oder weggeschaltet, wird nach der Rückkehr in RUN oder COND. RUN der XOB 30 für jedes Dazu- und Wegschalten aufgerufen und abgearbeitet.

Ist der XOB 30 nicht programmiert, wird anstelle des Aufrufs des XOB das Error-Flag gesetzt und die Error-LED eingeschaltet.

In unseren beiden Beispielen TEST1 und TEST2 wurde in diesem XOB 30 bloss ein Counter inkrementiert. Die Anzahl Zu- und Wegschaltungen kann so abgelesen werden.

Eine weitergehende Diagnose mit dem XOB 30 wird später gezeigt (Abschnitt 7.7), da zuvor die Diagnose-Elemente besprochen werden sollen.

7.3 Diagnose-Flags

In der Masterstation sind 24 Flags für die Grob-Diagnose der Kommunikation vorgesehen. Die Adresse des 1. der 24 Flags ist in der Rubrik "DIAG" des SASI-Textes enthalten.

Die **8 ersten Flags** (Flags 0..7) enthalten die übliche Kommunikations-Grobdiagnose zur Erkennung von Kommunikationsmerkmalen durch das Anwenderprogramm.

Adresse	Name	Beschreibung
xxxx	RBSY	Receiver busy Empfänger busy
xxxx + 1	RFUL	Receive buffer full Empfangsbuffer voll
xxxx + 2	RDIA	Receiver diagnostic Empfänger-Diagnose
xxxx + 3	TBSY	Transmitter busy Sender busy
xxxx + 4		Not used Nicht benützt
xxxx + 5	TDIA	Transmitter diagnostic Sender-Diagnose
xxxx + 6	XBSY	SASI permission SASI Erlaubnis
xxxx + 7	NEXE	Not executed Nicht ausgeführt

Die **8 folgenden Flags** (Flags 8...15) enthalten die übliche Kommunikations-Diagnose zur Erkennen von Kommunikationsmerkmalen der RIO-Aufgabe.

Adresse	Name	Beschreibung
xxxx + 8	RBSY	Receiver busy Empfänger busy
xxxx + 9	RFUL	Receive buffer full Empfangsbuffer voll
xxxx + 10	RDIA	Receiver diagnostic Empfänger-Diagnose
xxxx + 11	TBSY	Transmitter busy Sender busy
xxxx + 12		Not used Nicht benützt
xxxx + 13	TDIA	Transmitter diagnostic Sender-Diagnose
xxxx + 14	XBSY	SASI permission SASI Erlaubnis
xxxx + 15	NEXE	Not executed Nicht ausgeführt

Der Anwender hat im allgemeinen mit den Standard-Flags nichts zu tun. Im Fehlerfall kann jedoch der Spezialist diesen Diagnose-Flags interessante Hinweise zur Fehlererkennung entnehmen. Weitergehende Erläuterungen sind dem Handbuch "SAIA[®] S-Bus" zu entnehmen.

Das **Flag 16** (fcfg) zeigt an, dass der Master Konfigurations-Meldungen zu den Slaves schickt.

Das **Flag 17** (frdy) zeigt an, dass die Auffrischung des Medianabbildes durchgeführt wird.

Das **Flag 19** (fall) zeigt an, dass alle Meldungen eines Zyklus zu den Slaves übermittelt wurden. Das Flag wird = H am Ende jedes erfolgreich verlaufenen Zyklus. Zu Überwachungszwecken kann dieses Flag im Anwenderprogramm nach jedem Zyklus abgefragt und zurückgesetzt werden. Das Flag wird durch die RIO-Aufgabe zurückgesetzt, wenn ein Übertragungsfehler festgestellt wurde.

Das **Flag 21** (festn) zeigt, dass eine Station online ans Netz zugeschaltet oder vom Netz getrennt wird.

Das **Flag 22** (fecfg) zeigt, dass der Inhalt eines DB, welcher der Konfiguration einer Station dient, verstümmelt ist. Die Nummer des DB kann dem 4. Diagnoseregister entnommen werden.

Das **Flag 23** (femsg) zeigt, dass der Inhalt eines DB, welcher eine zyklische Telegramm definiert, verstümmelt ist. Die Nummer des DB kann dem 4. Diagnoseregister entnommen werden.

Zusammenfassung der Diagnose-Flags

Position	Funktion	Beschreibung
F_diag+0.. F_diag+7	Übliche Diagnose, wenn Kommunikation via Anwenderprogramm (STXM/SRXM)	Die Bedeutung dieser Flags entspricht der S-Bus Kommunikation
F_diag+8.. F_diag+15	Kommunikations-Diagnose der RIO-Aufgabe (zyklische Telegramme)	Die Bedeutung dieser Flags entspricht der S-Bus Kommunikation
F_diag+16: fcfg	Konfigurierung unterwegs	Der Master ist damit beschäftigt, Konfigurations-Anweisungen zu übermitteln.
F_diag+17: frdy	Auffrischung unterwegs	Die Auffrischung des Medienabbildes wird durchgeführt.
F_diag+19: fall	Alle Meldungen eines Zyklus korrekt übertragen	Zeigt an, dass alle zyklischen Telegramme von allen Slave-Stationen empfangen wurden. Das Flag kann vom Anwender zurückgeschaltet werden. Es wird dann im nächsten Zyklus wieder = H. Das Flag wird automatisch = L, wenn ein Telegramm negativ quittiert wurde.
F_diag+21: festn	Fehler einer Station	Zeigt an, dass eine Station zugeschaltet oder vom Netz getrennt wurde.
F_diag+22: fecfg	Konfigurationsfehler	Zeigt an, dass die Syntax eines Konfigurations-DB fehlerhaft ist. Die RIO-Aufgabe wird beim Auftreten eines solchen Fehlers abgebrochen. Dieses Flag muss vom Anwender zurückgesetzt werden.
F_diag+23: femsg	Übertragungsfehler	Zeigt an, dass die Syntax eines Übertragungs-DB fehlerhaft ist. Dieses Flag muss vom Anwender zurückgeschaltet werden.

Das Basiselement "F_diag" wird im Konfigurator automatisch definiert. Die absolute Basis-Adresse der Diagnose-Flags kann am einfachsten im Debugger mit 'Display' - 'teXt' - <Text-Nummer des SASI-Text> in Erfahrung gebracht werden.

7.4 Diagnose-Register

In der Masterstation sind neben den besprochenen 24 Flags auch 4 Register und 2 Datenblocks für die Fein-Diagnose vorgesehen. Das **1. Register** enthält die Standardinformationen für die Fehlerlokalisierung bei Verwendung der Kommunikations-Befehle STXM und SRXM (siehe SASI-Befehl in den entsprechenden Handbüchern). Die Adresse des 1. der 4 Register ist in der Rubrik "DIAG" des SASI-Textes enthalten.

	Bit	Bezeichnung	Beschreibung
E M P F Ä N G E R	0	Overrun error	Überlauf des internen Empfangsbuffers
	1		
	2	Framing error	Zeichen mit Format Fehler (Stopbit)
	3	Break error	Datenleitung unterbrochen *
	4	CRC-16 error	CRC-16 Fehler
	5	S-Bus PGU status	S-Bus PGU mit PLM
	6	SASI OFF permission	SASI OFF Erlaubnis
	7		
	8	Length error	Ungültige Telegrammlänge
	9		
	10	Telegram error	Ungültiges Telegramm
	11	Status error	PCD mit falschem Status, Unmöglichkeit den Befehl auszuführen
	12	Range error	Ungültige Elementadresse
	13	Value error	Ungültiger Datenwert
	14	Missing media error	Media-Adresse nicht definiert oder ungültig
15	Program error	Programm Fehler: Stationsnummer nicht zugewiesen	
S E N D E R	16	Retry count	Anzahl Telegrammwiederholungen in binärer Darstellung
	17		
	18		
	19		
	20	Negative response	Negative Antwort (NAK)
	21	Missing response	Keine Antwort nach Timeout
	22	Invalid response	Ungültige Antwort empfangen
	23	CTS-Timeout	CTS nach Ablauf TS-Delay nicht gesetzt
	24		
	25		
	26		
	27		
	28	Range error	Ungültige Elementadresse
	29		
	30		
	31	Program error	Programm Fehler: STXM/SRXM-Befehl nicht erlaubt

Das **2. Register** enthält auch die Standard-Informationen für die Fehlerlokalisierung, hier aber für die RIO-Aufgabe (zyklische Telegramme). Es gilt die gleiche Tabelle.

Das **3. Register** enthält die Nummer der Station, welche zuletzt einen Aufruf des XOB 30 verursacht hat. Das Register wird nach dem Einschalten der Anlage oder nach einem Kaltstart mit dem Wert 0 initialisiert. Ist die Station im Netz, ist die Nummer positiv, ist die Station nicht im Netz, erscheint die Nummer negativ.

Dies kann an den Beispielen TEST1 und TEST2 ausprobiert werden.

Das **4. Register** enthält die DB-Nummer in welchem Meldungen während dem Auftreten des Fehlers liegen. Wenn die Station ans Netz kommt, wird die DB-Nummer mit 0 angegeben. Es wird also nur eine DB-Nummer angegeben, wenn eine Station vom Netz wegegeschaltet ist. Die Basisadresse dieser Meldungs-DBs wird bei der Konfiguration angegeben (Message DB).

Zusammenfassung der Diagnose-Register:

Position	Funktion	Beschreibung
R_diag: rcom1	Übliche Diagnose, wenn Kommunikation via Anwenderprogramm (STXM/SRXM)	Übliche Diagnose-Bits wie im Modus S-Bus
R_diag+1: rcom2	Kommunikations-Diagnose der RIO-Aufgabe (zyklische Telegramme)	Übliche Diagnose-Bits wie im Modus S-Bus
R_diag+2: rstn	Adresse der letzten Station	Nummer der letzten Station, welche einen Aufruf des XOB 30 verursacht hat. 0: Initialisierungswert, kein Error >0: Station wieder im Netz <0: Station aus dem Netz
R_diag+3: rdb	Nummer des letzten DB	Nummer des DB, welcher während eines Errors aktiv ist (0, wenn Station wieder im Netz).

Das Basiselement "R_diag" wird im Konfigurator automatisch definiert. Die absolute Basis-Adresse der Diagnose-Register kann am einfachsten im Debugger mit 'Display' - 'teXt' - <Text-Nummer des SASI-Text> in Erfahrung gebracht werden.

7.5 Diagnose-DB

Zu den 24 Diagnose-Flags und den 4 Diagnose-Registern gesellen sich noch 2 Diagnose Daten-Blocks (DB). Beide DBs bestehen aus je 127 Elementen (Werten) und werden als Diagnose-Zähler verwendet. Die Nummer des 1. der beiden DBs ist in der Rubrik "DIAG" des SASI-Textes enthalten.

Der 1. DB enthält die totale Anzahl der Telegramme, welche an jede Station übermittelt wurde. Jeder Station ist 1 Element aus dem DB zugewiesen. Das Element 0 wird nicht verwendet. Das Element 1 enthält den Zähler der Station Nr. 1, das Element 2 den Zähler der Station Nr. 2 usw.

Die Verwendung dieser DBs ist in den Inbetriebnahme-Abschnitten 6.1.9 und 6.2.10 der Beispiele TEST1 und TEST2 gezeigt.

Der 2. DB enthält für jede Station die Anzahl Übertragungsfehler und den Status der Station. Der Status der Station ist auf den Bits 0 bis 7 und die Anzahl Übertragungsfehler sind in den Bits 8 bis 31 codiert enthalten. Die Anzahl Übertragungsfehler wird durch Subtraktion von 255 vom erhaltenen Wert ermittelt. (Der erste Fehler schaltet das Bit 8 = H (1), was dem dezimalen Wert 256 bedeutet. Wird davon 255 abgezählt, erhalten wir den Wert 1 für 1 Fehler). Das Element 0 wird nicht verwendet. Das Element 1 enthält den Zähler der Station Nr. 1, das Element 2 den Zähler der Station Nr. 2 usw.

Die Nummer (Adresse) des 1. DB muss 4000 oder höher gewählt werden.

Bedeutung eines Elementes des 2. Diagnose-DB

31	8	7..4	3..0
Fehler-Zähler		Resv.	Status

Ist Bit Nr. 0 = H (1), ist die Station im Netz aktiv.
 Ist Bit Nr. 1 = H (1), ist die Station zu konfigurieren.
 Ist Bit Nr. 2 = H (1), wurde die Station vom Netz entfernt.
 Ist Bit Nr. 3 = H (1), bleibt die Station bis ans Ende des Zyklus ausserhalb des Netzes

Beispiele :

00000000 Die Station ist nicht im Netz
 00000001 Aktiver Slave im Netz
 0000x110 Unintelligenter Slave ist nicht mehr im Netz
 0000x100 Intelligenter Slave ist nicht mehr im Netz

Diese Zähler dienen zur Ermittlung der Übertragungsqualität des Netzwerks. Die DB werden normalerweise im Debugger eingesehen und gelöscht. Mit dem Befehl TFR (Transfer Data) können die Resultate im Bedarfsfall auch in die Anwendung eingebunden werden.

Auch dieser DB kann im Zusammenhang mit den beiden Beispielen TEST1 und TEST2 ausprobiert werden.

Zusammenfassung der Diagnose DB

Position	Funktion	Beschreibung
DB_diag: dbcnt	Anzahl der übermittelten Telegramme	Jedes Element des DB gehört zu einer Station im Netzwerk
DB_diag+1: dberr	Anzahl Übertragungsfehler und Status jeder Station	Jedes Element des DB gehört zu einer Station im Netzwerk

Das Basiselement "DB_diag" wird im Konfigurator automatisch definiert. Die absolute Basis-Adresse der Diagnose-Flags kann am einfachsten im Debugger mit 'Display' - 'teXt' - <Text-Nummer des SASI-Text> in Erfahrung gebracht werden.

7.6 Konfigurierung

Wie wir in den Kapiteln 5 "Der RIO-Konfigurator" und 6. "Programmierung" gesehen haben, geschieht die Konfigurierung der nicht-intelligenten Slave-Stationen und das Editieren der Messages der intelligenten Slave-Stationen im Konfigurator. Es werden danach auf Knopfdruck die Definitions- und die Konfigurationsdateien erstellt, welche im Kapitel 6. wiedergegeben sind. Es werden aber auch noch eine Reihe Data-Blocks (DB) erzeugt, welche die Konfiguration zu Gunsten der Firmware enthält.

Dieser Abschnitt soll, obwohl für den Anwender nicht von Wichtigkeit, den Inhalt dieser DB etwas erhellen.

7.6.1 Konfiguration der PCD1.RIO-Stationen

Jede Station hat einen eigenen Konfigurations-DB mit den folgenden Elementen:

1. Adresse der Station im Netzwerk
2. Timeout für die Überwachung der Anwesenheit des Masters
3. Code für die Identifikation des Moduls 1
4. Code für die Identifikation des Moduls 2
5. Code für die Identifikation des Moduls 3
6. Code für die Identifikation des Moduls 4
7. Reserve
8. Synchronisation

Wir schauen uns das Ganze am Beispiel TEST2 an.

Bei 'Generation Options...' aus dem Menü 'Network' haben wir für "First Configuration DB" die Address <10> [db_cfg] und für First Message Address <32> [db_msg] übernommen.

Im Debugger wählen wir <Display> <data-Block> <10> To <15> <CR> und erhalten folgendes:

```
>Display data-Block 10 To 15
10 [0]:    5      1000    0      8192      8192
10 [5]:    0        0      1
11 [0]:    9      1000   4096   4096      0
11 [5]:    0        0      1
12 [0]:   -1      -1     -1     -1      -1
12 [5]:   -1      -1     -1
13:  DATA BLOCK NOT DEFINED
14:  DATA BLOCK NOT DEFINED
15:  DATA BLOCK NOT DEFINED
```

DB 10 und DB 11 sind verwendet. Jeder DB besteht aus 8 Elementen (Werten) gemäss nachfolgender Aufstellung:

DB 10:

1. Element:	5	Stationsnummer
2. Element	1000	Timeout in ms
3. Element	0	Modul Nr. 1 leer
4. Element	8192	Modul Nr. 2 digitales Ausgangsmodul
5. Element	8192	Modul Nr. 3 digitales Ausgangsmodul
6. Element	0	Modul Nr. 4 leer
7. Element	0	Reserve
8. Element	1	Synchronisation

DB 11:

1. Element:	9	Stationsnummer
2. Element	1000	Timeout in ms
3. Element	4096	Modul Nr. 1 digitales Eingangsmodul
4. Element	4096	Modul Nr. 2 digitales Eingangsmodul
5. Element	0	Modul Nr. 3 leer
6. Element	0	Modul Nr. 4 leer
7. Element	0	Reserve
8. Element	1	Synchronisation

DB 12:

1. Element:	-1	⋮
2. Element	-1	⋮
3. Element	-1	⋮ Ende der
4. Element	-1	⋮ Konfigurierung
5. Element	-1	⋮
6. Element	-1	⋮ (kein weiterer DB)
7. Element	-1	⋮
8. Element	-1	⋮

Die Codes für die einzelnen Module ist der nachfolgenden Tabelle zu entnehmen (nächste Seite).

Die Codes für die Module sind die folgenden:

Code	Hex.	Dez.	Beschreibung
leer	equ 0	(0)	;Code used when no module configured
exxx	equ 1000H	(4096)	;Code for exxx modules
axxx	equ 2000H	(8192)	;Code for axxx modules
w1x1	equ 3010H	(12304)	;Code for w1xx modules, 1 input used
w1x2	equ 3020H	(12320)	;Code for w1xx modules, 2 inputs used
w1x3	equ 3030H	(12336)	;Code for w1xx modules, 3 inputs used
w1x4	equ 3040H	(12352)	;Code for w1xx modules, 4 inputs used
w2x1	equ 4010H	(16040)	;Code for w2xx modules, 1 input used
w2x2	equ 4020H	(16416)	;Code for w2xx modules, 2 inputs used
w2x3	equ 4030H	(16432)	;Code for w2xx modules, 3 inputs used
w2x4	equ 4040H	(16448)	;Code for w2xx modules, 4 inputs used
w2x5	equ 4050H	(16464)	;Code for w2xx modules, 5 inputs used
w2x6	equ 4060H	(16480)	;Code for w2xx modules, 6 inputs used
w2x7	equ 4070H	(16496)	;Code for w2xx modules, 7 inputs used
w2x8	equ 4080H	(16512)	;Code for w2xx modules, 8 inputs used
w4x1	equ 5100H	(20436)	;Code for w4xx modules, 1 output used
w4x2	equ 5200H	(20992)	;Code for w4xx modules, 2 outputs used
w4x3	equ 5300H	(21248)	;Code for w4xx modules, 3 outputs used
w4x4	equ 5400H	(21504)	;Code for w4xx modules, 4 outputs used
w501	equ 6010H	(24592)	;Code for w5xx modules, 0 out. 1 in. used
w502	equ 6020H	(24608)	;Code for w5xx modules, 0 out. 2 in. used
w510	equ 6100H	(24832)	;Code for w5xx modules, 1 out. 0 in. used
w511	equ 6110H	(24848)	;Code for w5xx modules, 1 out. 1 in. used
w512	equ 6120H	(24864)	;Code for w5xx modules, 1 out. 2 in. used
w520	equ 6200H	(25088)	;Code for w5xx modules, 2 out. 0 in. used
w521	equ 6210H	(25104)	;Code for w5xx modules, 2 out. 1 in. used
w522	equ 6220H	(25120)	;Code for w5xx modules, 2 out. 2 in. used

7.6.2 Konfigurierung der zyklischen "Messages"

Die Liste mit den "Messages" zur Auffrischung des Medienabbildes liegt in der Konfigurationsdatei, für unser Beispiel in "TEST2.SRC" (siehe Abschnitt 6.2.7). Die "Messages" haben folgendes Format:

1. Adresse der Station im Netzwerk
2. Auszuführende Funktion
3. Anzahl zu übertragender Medien
4. Erstes Quell-Medium
5. Erstes Ziel-Medium
6. Reserve (immer = 0)
7. Reserve (immer = 0)

Diese Informationen liegen auch in einer Reihe von Data-Blocks (DB).

Wir schauen uns dies am Beispiel TEST2 an.

Bei 'Generation Options...' aus dem Menü 'Network' haben wir für "First Configuration DB" die Address <10> [db_cfg] und für First Message Address <32> [db_msg] übernommen.

Im Debugger wählen wir <Display> <data-Block> <32> To <40> <CR> und erhalten folgendes:

```
>Display data-Block 32 To 40
```

32 [0]:	5	8	16	2064	64
32 [5]:	0	0			
33 [0]:	9	1	16	0	96
33 [5]:	0	0			
34 [0]:	15	2	8	16	50
34 [5]:	0	0			
35 [0]:	15	9	8	60	64
35 [5]:	0	0			
36 [0]:	15	10	1	55	77
36 [5]:	0	0			
37 [0]:	15	9	4	40	32
37 [5]:	0	0			
38 [0]:	-1	-1	-1	-1	-1
38 [5]:	0	0			
39:	DATA BLOCK NOT DEFINED				
40:	DATA BLOCK NOT DEFINED				

Es sind die DB 32 bis 37 verwendet. Jeder DB besteht aus 7 Elementen (Werten) gemäss nachfolgender Aufstellung:

DB 32:

1. Element:	5	Stationsnummer
2. Element	8	Code für auszuführende Funktion (siehe Tabelle)
3. Element	16	Anzahl zu übertragender Medien
4. Element.	2064	Erstes Quell-Medium
5. Element	64	Erstes Ziel Medium
6. Element	0	Reserve
7. Element	0	Reserve

Codes für die auszuführende Funktion:

Code 1:	Kopieren der Flags eines Slaves zu Flags des Masters
Code 2	Kopieren der Eingänge eines Slaves zu Flags des Masters
Code 3:	Kopieren von Registern eines Slaves zu Reg. des Masters
Code 8:	Kopieren von Flags des Masters zu Flags des Slaves
Code 9:	Kopieren von Flags des Masters zu Ausgängen des Slaves
Code 10:	Kopieren von Registern des Masters zu Reg. des Slaves

Die Station Nr. 5 hat demnach nur 1 Job:

- Kopieren von Flags des Masters zu Flags des Slaves
- 16 Elemente, es sind dies die 2 digitalen Ausgangsmodule PCD4.A400
- Die Quellelemente sind die Flags 2064 - 2079 des Masters
- Die Zielelemente sind die Flags 64 - 79 des Slaves *)

*) In den Slaves wird die Information der digitalen Ein- und Ausgänge auf Flags und von dort, für den Anwender nicht sichtbar, auf die Ein- und Ausgänge gebracht.

DB 33 betrifft die Station Nr. 9. Diese hat auch nur 1 Job: es werden die 16 Eingänge der PCD4.E110 Eingangsmodule auf die Flags 0 - 15 der Master-Station übertragen.

Die DB 34 - 37 betreffen die 4 Jobs der Station 15.

DB 38 zeigt, dass hier die Reihe der Konfigurations-DB zu Ende ist.

7.7 Diagnose in der praktischen Anwendung

Wie gesehen, werden in den Diagnose-Flags, den Diagnose-Registern und den Diagnose-DB umfangreiche Informationen angeboten. Aus dieser Vielfalt soll eine sinnvolle praktische Anwendung herausgegriffen werden.

Es soll in unserem Beispiel TEST2 aus dem Abschnitt 6.2 ein Diagnose angefügt werden, welche dem Betreiber der Anlage anzeigt, welche Slave-Stationen im Netz sind bzw. welche nicht.

Es wird an der Master-Station (PCD4) anstelle des 1. Eingangsmoduls PCD4.E100 ein E/A-Modul vom Typ PCD4.B900 eingesetzt. An den Ausgängen 0 - 15 sollen die Status der Slave-Stationen 0 - 15 gezeigt werden:

Ausgang = L: Station im Netz

Ausgang = H: Station nicht im Netz

Diese Diagnose soll im XOB 30 erfolgen.

Das Diagnose-Flag "F_diag+21" zeigt an, dass eine Station zugeschaltet oder vom Netz getrennt wurde. Dieses Flag kann ausgewertet und im Anwenderprogramm wieder zurückgeschaltet werden, damit das nächste Ereignis, Zu- oder Wegschalten einer andern Station, wieder erfasst werden kann.

Das Diagnose-Register "R_diag+2" enthält die Nummer der letzten Station, welche einen Aufruf des XOB 30 verursacht hat. Die Stationsnummer ist negativ, wenn die Station nicht im Netz und positiv, wenn die Station im Netz ist.

Mit diesen beiden Meldungen und etwas Kenntnis des Befehlssatzes der PCD, lässt sich die Diagnose-Aufgabe, welche wir uns gestellt haben, lösen.

Der Programmteil soll in reiner BLOCTEC-Struktur, also ohne Jumps, erstellt werden. (Siehe nächste Seite)

Die Diagnose könnte natürlich noch verfeinert werden, indem z.B. eine Veränderung in den Konfigurations-DB festgestellt und dann reagiert werden könnte. (F_diag+22 und F_diag+23). Dies scheint aber doch von recht weit hergeholt, denn wenn diese DB tatsächlich verändert wären, dürften im Anwenderprogramm noch andere Daten verändert worden sein, was dann die korrekte Funktion der CPU sowieso in Frage stellen würde.

```

xob          30

inc    c      100      ; Ereignis-Zähler

mul    r      r_diag+2 ; | Absolutwert
       r      r_diag+2 ; | der Stationsnummer
       r      0        ; | erstellen.
sqr    r      0        ; | Hilfsregister für
       r      0        ; | Absolutwert der St.Nr.

copy   r      r_diag+2 ; | Statusflag
       r      r_diag+2 ; | n oder p erfassen

acc    n
anh    f      f_diag+21 ; wenn St.Nr. negativ
                        ; und Flag = H

cfb    h      0        ; wenn negativ: Call FB 0
       r      0        ; Parameter 1

cfb    l      1        ; wenn positiv: Call FB 1
       r      0        ; Parameter 1

acc    h
res    f      f_diag+21 ; Rücksetzen Diag-Flag

exob
;-----

fb          0          ; Station vom Netz weg

sei    =      1          ; Lade Indexreg. mit St.Nr.
setx   o      0          ; Setze entsprechenden Ausgang

efb
;-----

fb          1          ; Station wieder da

sei    =      1          ; Lade Indexreg. mit St.Nr.
resx   o      0          ; Rücksetze entspr. Ausgang

efb
;-----

```

Dieser Programmteil kann ins Anwenderprogramm "TEST2" eingebaut und getestet werden. (TEST2_XA)

Absender:

Firma
Abteilung
Name
Adresse

Tel.

Datum

An:

SAIA-Burgess Electronics AG
Bahnhofstrasse 18
CH-3280 Murten (Schweiz)
<http://www.saia-burgess.com>

GB: Electronic Controllers

SAIA[®] Remote I/O PCD1.T240

Falls Sie Vorschläge zu SAIA[®] PCD zu machen oder Fehler in diesem Handbuch gefunden haben, sind wir Ihnen für einen kurzen Bericht dankbar.

Ihre Vorschläge: