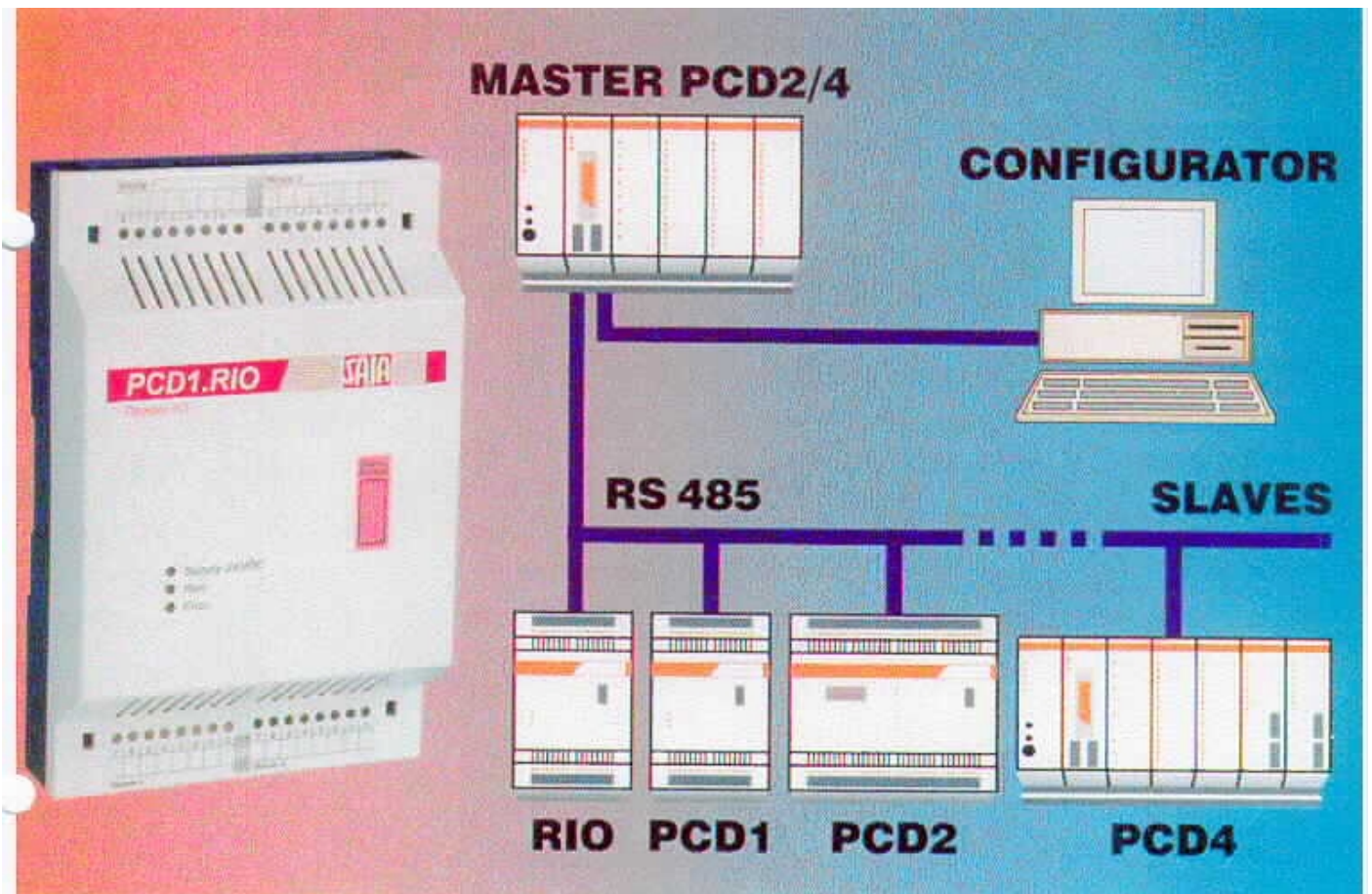


## SAIA® PCD Process Control Devices

## PCD1.T240 SAIA® Remote I/O



---

## SAIA-Burgess Companies

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

---

## Representatives

<b>Great Britain</b>	Canham Controls Ltd. 25 Fenlake Business Centre, Fengate Peterborough PE1 5BQ UK ☎ 01733 89 44 89, Fax 01733 89 44 88	<b>Portugal</b>	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
<b>Denmark</b>	Malthe Winje Automation AS Håndværkerbyen 57 B DK-2670 Greve ☎ 70 20 52 01, Fax 70 20 52 02	<b>Spain</b>	Tecnosistemas Medioambientales, S.L. Poligono Industrial El Cabil, 9 E-28864 Ajalvir, Madrid ☎ 91 884 47 93, Fax 91 884 40 72
<b>Norway</b>	Malthe Winje Automasjon AS Haukelivn 48 N-1415 Oppegård ☎ 66 99 61 00, Fax 66 99 61 01	<b>Czech Republic</b>	ICS Industrie Control Service, s.r.o. Modranská 43 CZ-14700 Praha 4 ☎ 2 44 06 22 79, Fax 2 44 46 08 57
<b>Sweden</b>	Malthe Winje Automation AB Truckvägen 14A S-194 52 Upplands Väsby ☎ 08 795 59 10, Fax 08 795 59 20	<b>Poland</b>	SABUR Ltd. ul. Druzynowa 3A PL-02-590 Warszawa ☎ 22 844 63 70, Fax 22 844 75 20
<b>Suomi/ Finland</b>	ENERGEL OY Atomitie 1 FIN-00370 Helsinki ☎ 09 586 2066, Fax 09 586 2046		
<b>Australia</b>	Siemens Building Technologies Pty. Ltd. Landis & Staefa Division 411 Ferntree Gully Road AUS-Mount Waverley, 3149 Victoria ☎ 3 9544 2322, Fax 3 9543 8106	<b>Argentina</b>	MURTEN S.r.l. Av. del Libertador 184, 4° "A" RA-1001 Buenos Aires ☎ 054 11 4312 0172, Fax 054 11 4312 0172

---

## After sales service

<b>USA</b>	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**

**for the PCD family**

SAIA-Burgess Electronics Ltd. 1996. All rights reserved  
Edition 26/751 E2 - 09.1996

Subject to technical changes

# Updates

---

Manual : SAIA<sup>®</sup> Remote I/O PCD1.T240 - Edition E2

Date	Chapter	Page	Description

# Contents

---

	Page
<b>1. Description and characteristics</b>	<b>1-1</b>
1.1 What is PCD-RIO?	1-1
1.2 Characteristics	1-2
<b>2. SAIA® PCD devices for a PCD-RIO network</b>	<b>2-1</b>
2.1 Which devices can be used as masters?	2-1
2.2 Which devices can be used as slaves?	2-2
<b>3. PCD1.RIO hardware (PCD1.T240)</b>	<b>3-1</b>
3.1 General	3-1
3.2 Technical data for PCD1.RIO	3-2
3.2.1 Specific data	3-2
3.2.2 General technical data	3-3
3.3 Presentation and dimensions	3-4
3.3.1 Presentation of PCD1.RIO	3-4
3.3.2 Assembly, dimension drawing and cabling	3-5
3.4 Architecture and presentation of the main board	3-6
3.4.1 Architecture	3-6
3.4.2 Presentation of the main board	3-7
3.5 The RS 485 interface	3-10
3.5.1 Connection of RS 485 interface	3-10
3.5.2 Connection and placement of RS 485 bus line	3-11
3.6 Addressing the input and output modules	3-13

	Page
<b>4. Concept of the RIO network</b>	<b>4-1</b>
4.1 General description	4-1
4.2 Files for the RIO task	4-3
4.3 Activation of the RIO task	4-6
4.4 Diagnostic possibilities	4-7
4.5 Use of the RIO channel for asynchronous telegram transmission	4-8
4.6 Refresh and reaction times	4-9
4.7 Firmware required	4-10
<b>5. The RIO configurator: PCD8.C3..</b>	<b>5-1</b>
5.1 Introduction	5-1
5.2 Installing the RIO configurator	5-2
5.3 Running the RIO configurator	5-3
5.3.1 'Network' submenu	5-4
5.3.2 'Edit' submenu	5-9
5.3.3 'Station Parameter...' for the master station	5-11
5.3.4 'Station Parameter...' for a non-intelligent slave station	5-14
5.3.5 'Station Parameter...' for an intelligent slave station	5-22
5.3.6 'View' submenu	5-25
5.3.7 'Library' submenu	5-26
5.3.8 'Window' submenu	5-27
5.3.9 'Help' submenu	5-28
5.3.10 Toolbar	5-29

	Page
<b>6. Programming</b>	<b>6-1</b>
6.1 Introductory example	6-1
6.1.1 Task	6-1
6.1.2 Station configuration	6-3
6.1.3 Configuring slave station I/O modules	6-4
6.1.4 Generation of auxiliary files	6-8
6.1.5 Definition file	6-10
6.1.6 Configuration file	6-11
6.1.7 "RIO" file	6-12
6.1.8 User program	6-15
6.1.9 Commissioning	6-17
6.2 Further example	6-19
6.2.1 Task	6-19
6.2.2 Station configuration	6-21
6.2.3 Configuring slave stations	6-22
6.2.4 Generation of auxiliary files	6-28
6.2.5 Definition file	6-29
6.2.6 Definition file of intelligent slave station	6-31
6.2.7 Configuration file	6-32
6.2.8 User program for master station	6-33
6.2.9 User program for intelligent slave station	6-37
6.2.10 Commissioning	6-39
6.3 Causes of common errors	6-41
<b>7. Diagnosis</b>	<b>7-1</b>
7.1 Monitoring presence of master station	7-1
7.2 Monitoring presence of slave stations (XOB 30)	7-2
7.4 Diagnostic flags	7-3
7.4 Diagnostic registers	7-6
7.5 Diagnostic DBs	7-8
7.6 Configuration	7-10
7.6.1 Configuration of PCD1.RIO stations	7-10
7.6.2 Configuration of cyclical messages	7-13
7.7 Diagnosis in practical application	7-15







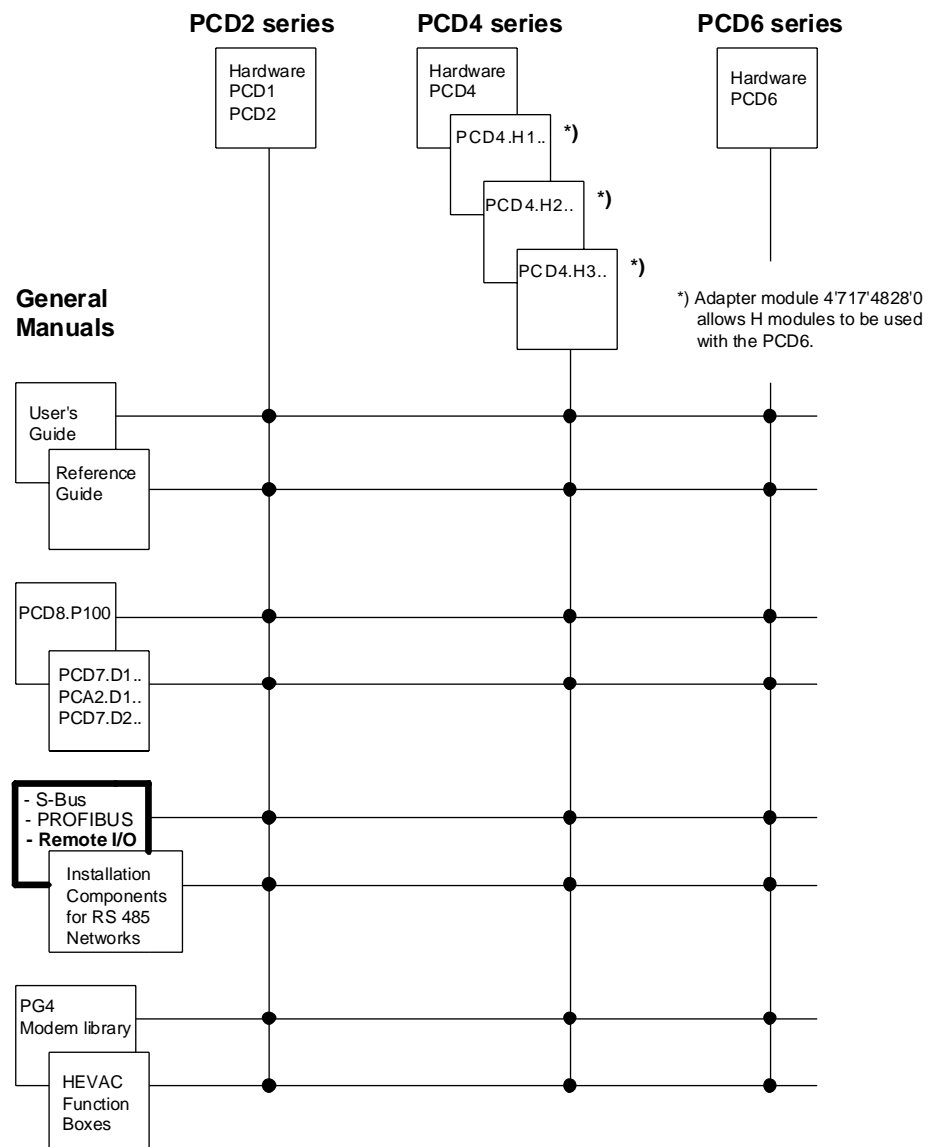
**Please note:**

A number of detailed manuals are available to aid installation and operation of the SAIA PCD. These are for use by technically qualified staff, who may also have successfully completed one of our "workshops".

To obtain the best performance from your SAIA PCD, closely follow the guidelines for assembly, wiring, programming and commissioning given in these manuals. In this way, you will also become one of the many enthusiastic SAIA PCD users.

If you have any technical suggestions or recommendations for improvements to the manuals, please let us know. A form is provided on the last page of this manual for your comments.

**Summary**



## Reliability and safety of electronic controllers

---

SAIA-Burgess Electronics Ltd. is a company which devotes the greatest care to the design, development and manufacture of its products:

- state-of-the-art technology
- compliance with standards
- ISO 9001 certification
- international approvals: e.g. Germanischer Lloyd, UL, Det Norske Veritas, CE mark ...
- choice of high-quality componentry
- quality control checks at various stages of production
- in-circuit tests
- run-in (burn-in at 85°C for 48h)

Despite every care, the excellent quality which results from this does have its limits. It is therefore necessary, for example, to reckon with the natural failure of components. For this reason SAIA-Burgess Electronics Ltd. provides a guarantee according to the "General terms and conditions of supply".

The plant engineer must in turn also contribute his share to the reliable operation of an installation. He is therefore responsible for ensuring that controller use conforms to the technical data and that no excessive stresses are placed on it, e.g. with regard to temperature ranges, overvoltages and noise fields or mechanical stresses.

In addition, the plant engineer is also responsible for ensuring that a faulty product in no case leads to personal injury or even death, nor to the damage or destruction of property. The relevant safety regulations should always be observed. Dangerous faults must be recognized by additional measures and any consequences prevented. For example, outputs which are important for safety should lead back to inputs and be monitored from software. Consistent use should be made of the diagnostic elements of the PCD, such as the watchdog, exception organization blocks (XOB) and test or diagnostic instructions.

If all these points are taken into consideration, the SAIA PCD will provide you with a modern, safe programmable controller to control, regulate and monitor your installation with reliability for many years.

# 1. Description and characteristics

---

## 1.1 What is PCD-RIO?

---

"RIO" is the abbreviation for "Remote Input/Output", and refers to the remote interrogation or control of I/Os. In the context of the PCD series however, a RIO does not only restrict itself to digital I/Os, but also extends to analogue I/Os and, at a further level of design capacity, to the exchange of register, counter and timer values.

A PCD-RIO network is based on the universal SAIA<sup>®</sup> S-Bus, which is included as standard equipment in all PCD stations. A RIO network consists of a master station and 1 to 31 slave stations. The master is always an intelligent station, i.e. a PCD with a CPU and a user program. The slave stations can be in any combination: either non-intelligent (i.e. RIO stations without a CPU or user program) or intelligent PCD stations. Regardless of the RIO mechanism, each intelligent station can process its own user programs.

By connecting up to 3 repeaters in series, the RIO network can be extended to 125 stations. The maximum transmission speed for communications is 76.8 kBaud.

The RIO principle consists of a permanently refreshed map constructed in the master station which reflects all the I/Os of non-intelligent slave stations and of certain I/O groups in the intelligent stations. The communication relationships are executed in the user program of the master station. For example, if station no. 15 wants to map some of the inputs on station no. 7, the user program in the master ensures that the inputs of station no. 15 are interrogated and then sent to the outputs of station no. 7.

Permanently refreshing this map takes time, of course, which means that the low-cost, non-intelligent RIO stations can only be used where longer reaction times ( $\geq 100$  ms) are acceptable.

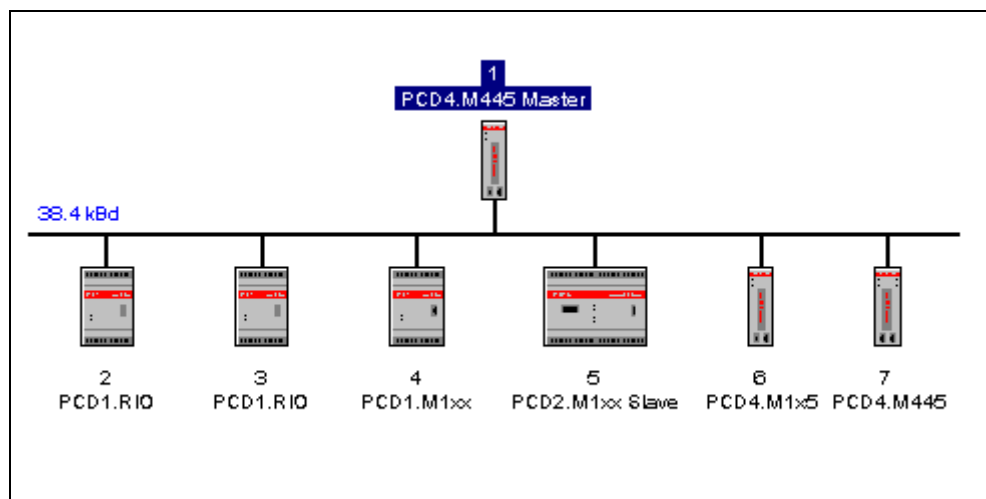
Faster reaction times are achieved by using intelligent stations, if events are triggered immediately, i.e. only when the result is needed. This communication takes place through the same network. The execution instructions, however, are generated in the user program of each individual station and do not pass via the map in the master station.

## 1.2 Characteristics

---

- RS 485 bus with proven S-Bus protocol up to 76.8 kBaud
- Master station with processor module PCD4.M445, PCD4.M1x5 or PCD2.M1xx
- Non-intelligent PCD1.RIO slave stations, for modular insertion of 4 digital or analogue I/O modules
- Intelligent slave stations PCD1, PCD2 and/or PCD4 equipped as desired
- Up to 31 slave stations without repeaters or up to 125 slave stations with 3 PCD7.T100 repeaters
- User friendly configurator under WINDOWS 3.1 or WINDOWS 95 for graphical, menu-driven configuration of the overall network.

### Example of a small PCD-RIO network:



(This small diagram was created with the RIO configurator).

## 2. SAIA<sup>®</sup> PCD devices for a PCD-RIO network

---

### 2.1 Which devices can be used as masters?

---

Usable ports and maximum transmission rates in kBaud of suitable (intelligent) **master stations**.

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	<b>76.8</b>	<b>76.8</b>

With PCD4 devices, port 0 (PGU) always has the RS 232 standard. An RS 232 ↔ RS 485 converter (PCD7.T120) should be used here. For ports 1 to 3, use of the PCD4.C340 bus module is assumed, equipped with RS 485 interface modules (PCD7.F110 or PCD7.F150). With the PCD4.C130 bus module, port 1 can be used directly (RS 485); for other ports and other bus modules, PCD7.T120 converters should also be used.

Descriptions of the PCD4, PCD2 and PCD1 are available in the relevant hardware manuals.

\*) see chapter 4.7: "Firmware required"

## 2.2 Which devices can be used as slaves?

---

For the **slave stations**, use is made primarily of the low-cost (non-intelligent) PCD1.RIO stations. It is possible to choose whether to run these at a transmission speed of 38.4 or 76.8 kBaud. It is also possible to use as slave stations the intelligent (master) devices listed in the previous table, with the same comments applying concerning interface standards. Intelligent devices from the PCD1 series can also be used as slaves.

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

In any one network, the same transmission speed must be selected on all stations. The maximum baud rate of 76.8 kBaud can therefore only operate in a network with a PCD4.M445 at CPU 1 (with ports 2 or 3) as master and the PCD1.RIO devices as slaves.

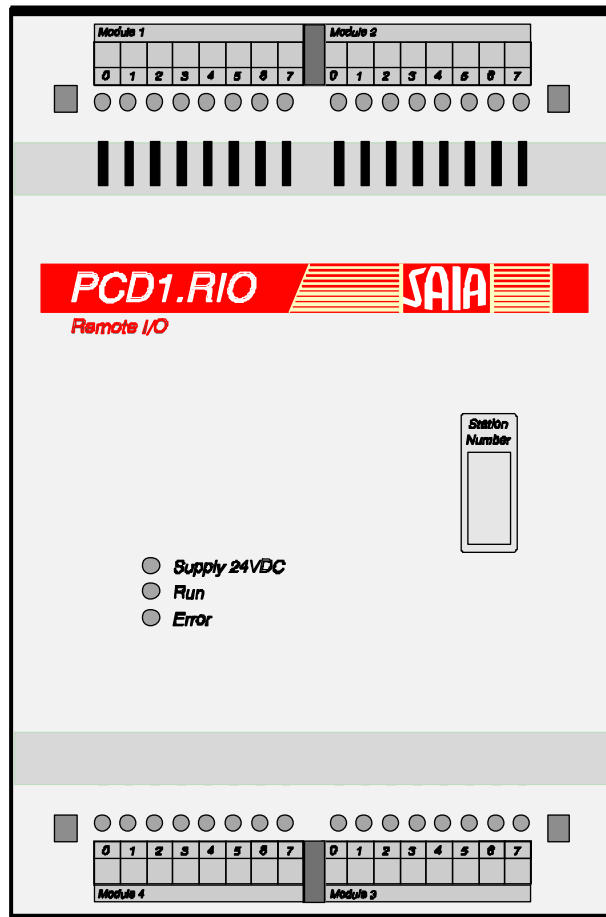
Descriptions of the PCD4, PCD2 and PCD1 are available in the relevant hardware manuals.

A description of the PCD1.RIO follows in the next section.

## 3. PCD1.RIO hardware (PCD1.T240)

### 3.1 General

The PCD1.RIO uses the same housing as the PCD1, but has no PGU connector and is labelled differently. The PCD1.RIO can be equipped with 4 PCD2 I/O modules.



The PCD1.RIO includes an RS 485 interface for connection to the RIO network. An 8-position DIL switch enables the station number and transmission speed to be adjusted. The PCD1.RIO is not programmable. Once the PCD1.RIO has been configured, connected to a supply, linked to the RIO network, and the station address and transmission speed adjusted via the DIL switch, the device is operational and further handling is unnecessary.

## 3.2 Technical data of the PCD1.RIO (PCD1.T240)

---

### 3.2.1 Specific data

Number of inputs/outputs:	4 digital PCD2 I/O modules in any arrangement, or 4 analogue input or output modules, or a combination of digital and analogue I/O modules.
Processormodule:	1 central unit with a 68340 microprocessor
User memory:	none
Real time clock:	none
Communications interfaces:	no PGU interface 1 serial interface: S-Bus/RS 485
Means of programming:	no connection possible for programming unit

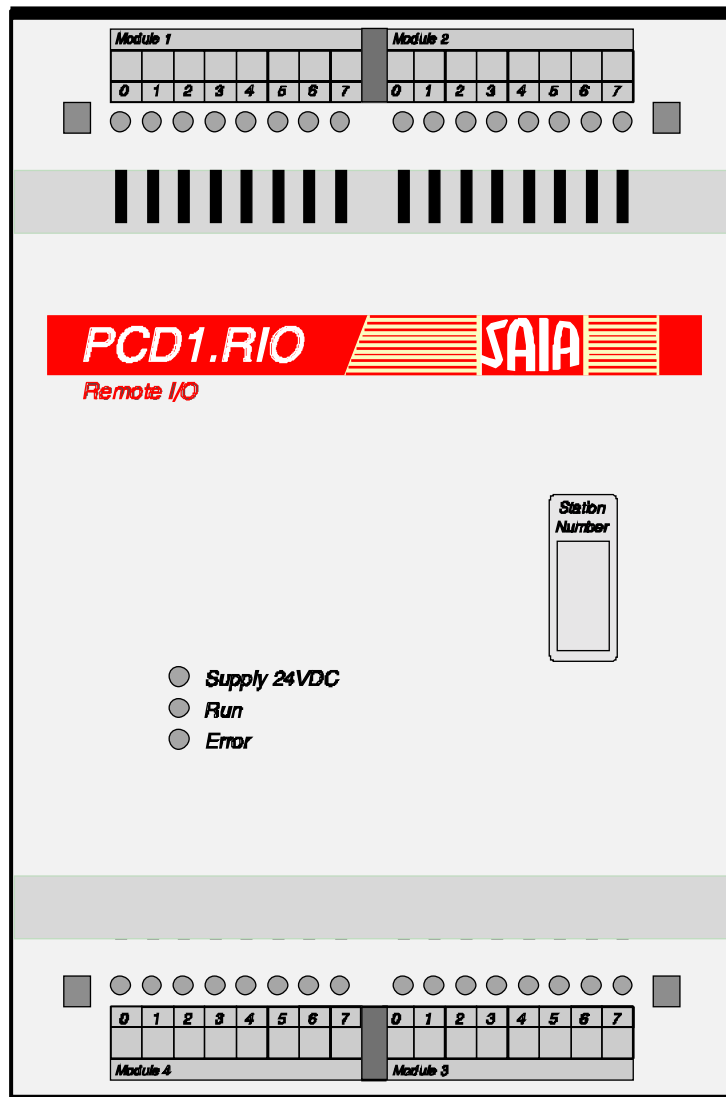


### 3.2.2 General technical data

Supply voltage:	24 VDC $\pm$ 20% smoothed, or 19 VAC $\pm$ 15% full-wave rectified (18 VDC)
Power consumption:	10W for 16 I/Os
Resistance to interference:	burst as IEC 801-4, ESD as IEC 801-2
Security:	according to CE conditions
Ambient temperature:	operating: 0...+55°C or. 0...+40°C (see mounting positions) storage temp.: -20...+85°C
Atmospheric humidity:	95% rel. humidity, without condensation (as DIN 40 040, class F)
Mechanical resistance:	according to IEC 68-2
Mounting positions:	Standard: on vertical surface (wall mounting, clips above and below) Special: on horizontal surface or wall mounting with side clips, for reduced ambient temperature (max. +40°C)
Connections:	I/O modules, add-on modules and supply via plug-in screw terminals for wires 1.5 mm <sup>2</sup> or 2 x 0.5 mm <sup>2</sup> .
For applications according to UL 508:	thermoplastic insulated Cu conductor 50/75°C
Starting torque:	0.5 Nm

### 3.3 Presentation and dimensions

#### 3.3.1 Presentation of PCD1.RIO



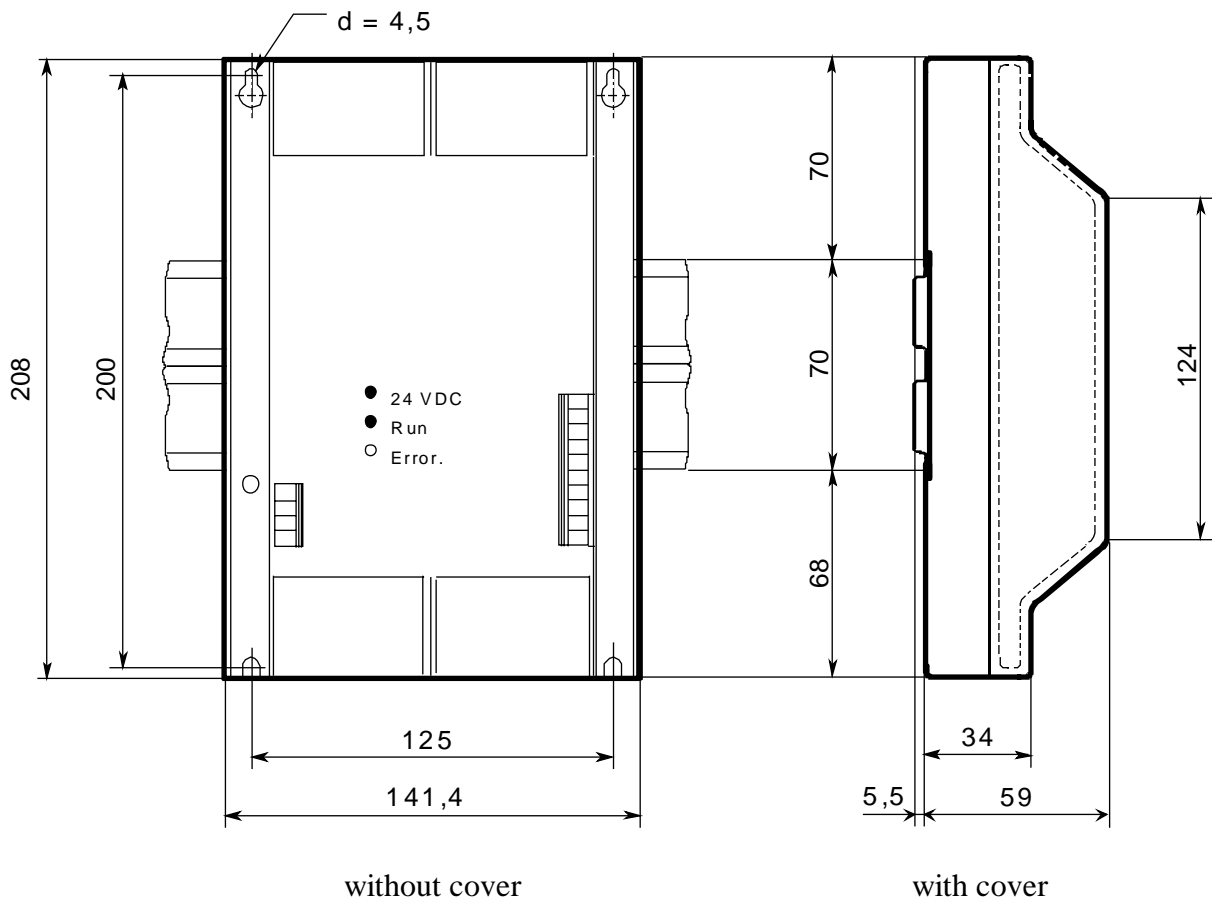
Only one version exists of the PCD1.RIO:

- Front: Plastic cover with signal lamps (LEDs) to display the operating state and position of the digital inputs and outputs, plus a label surface on which to write the station number.
- Beneath the front cover is the main board. This not only contains the electronics and the supply, but also the 8 position DIL switch for selection of station number and transmission speed.
- Beneath the main board is the bus module, containing connectors for the 4 individually inserted input and output modules.

### 3.3.2 Assembly, dimension drawing and cabling

The PCD1.RIO can be snapped onto the same double DIN rail (2x35 mm) as the bus modules of the PCD4 and PCD1 or PCD2. Instructions for assembly, the dimension drawing and proposed cabling can be obtained from the PCD2 manual, section 1.3.2.

#### Dimension drawing:

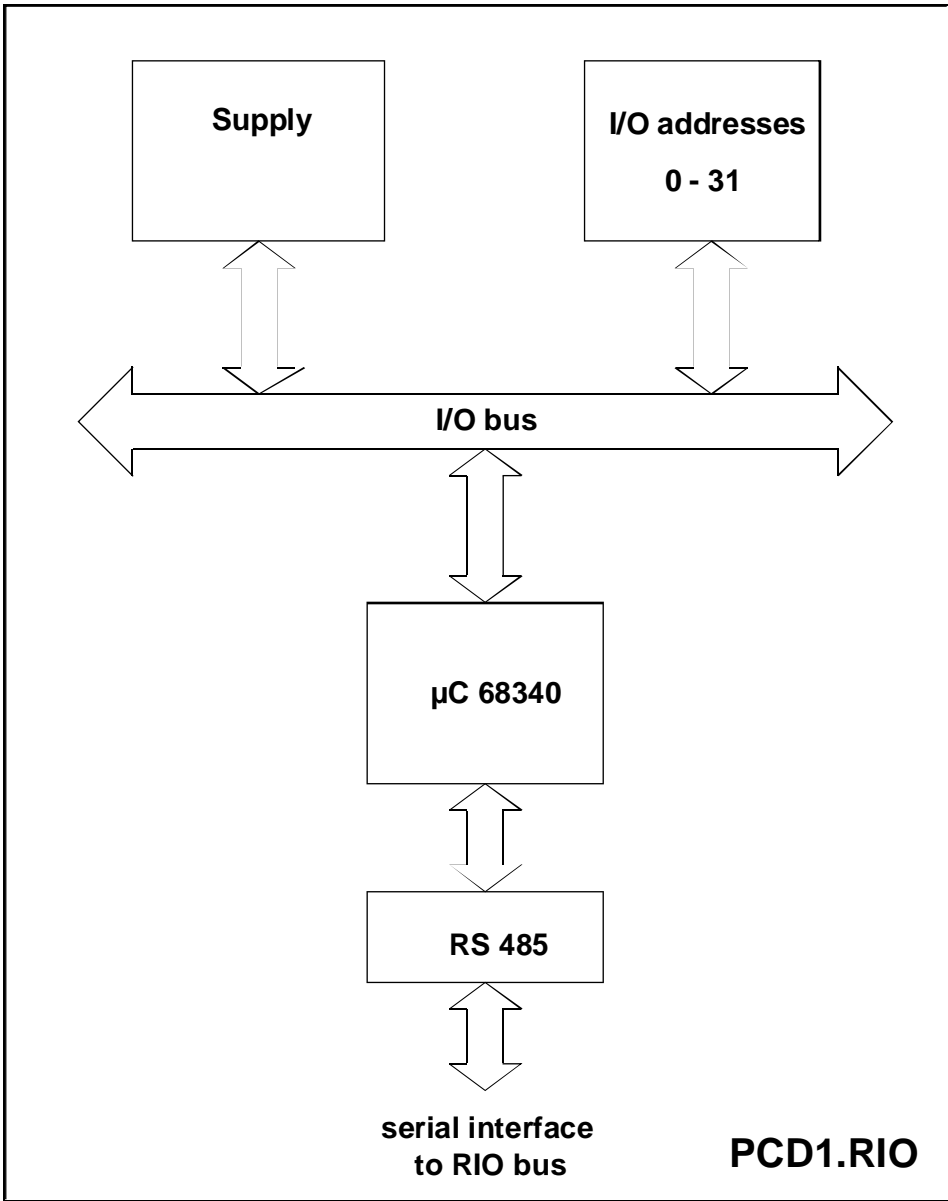


### 3.4 Architecture and Presentation of main board

---

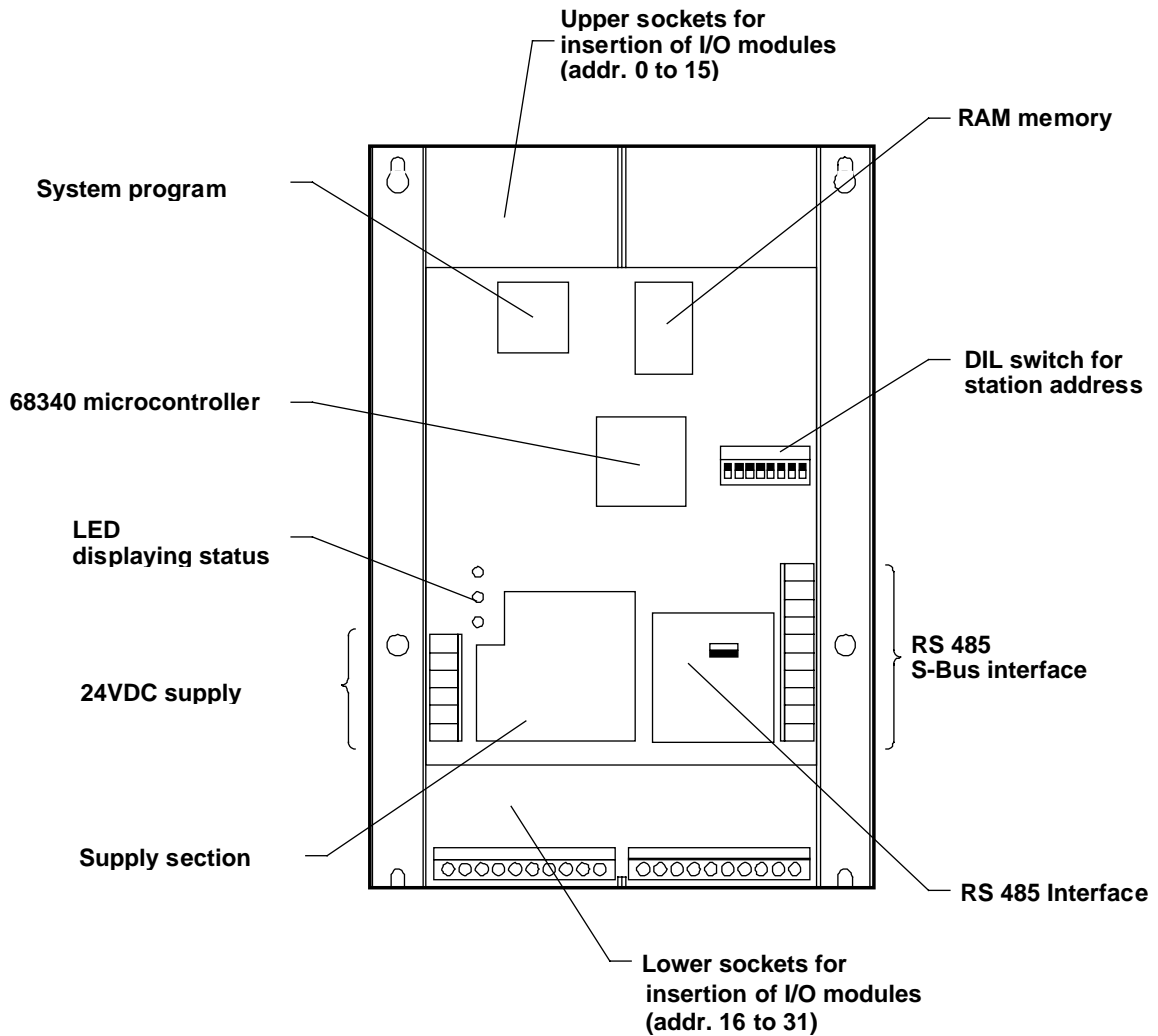
#### 3.4.1 Architecture

The following diagram shows the architecture of the PCD1.RIO:



### 3.4.2 Presentation of the main board

The following drawing shows the PCD1.RIO with its cover removed.

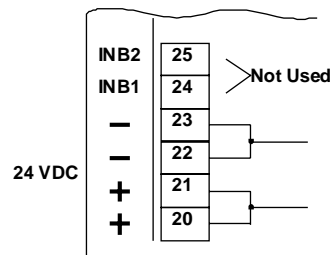


**Caution:** Removal of the cover leaves components accessible to touch which are sensitive to electrostatic discharges.



When the device is switched on, manipulations such as changing the jumpers or inserting/removing I/O modules etc. should also be avoided.

### Connection of the supply, safety cutout, noise suppression



Connection takes place at plug-in terminals 20 to 23 with wires of max. 1.5 mm<sup>2</sup> (with cable end piece)

Either smoothed or pulsed DC voltage can be used (see technical data).

The "24 VDC" LED indicates whether the supply voltage is present.

Reverse battery protection prevents destruction of the circuit by incorrect connection.

The 24V circuit is provided with a PTC resistor, which switches off in the presence of excess current and, in case of error, also protects the circuit from destruction.

Highly effective noise suppression filters protect the electronic circuits from parasitic overvoltages (4 kV according to CEI 801-4).

### Processor operating states

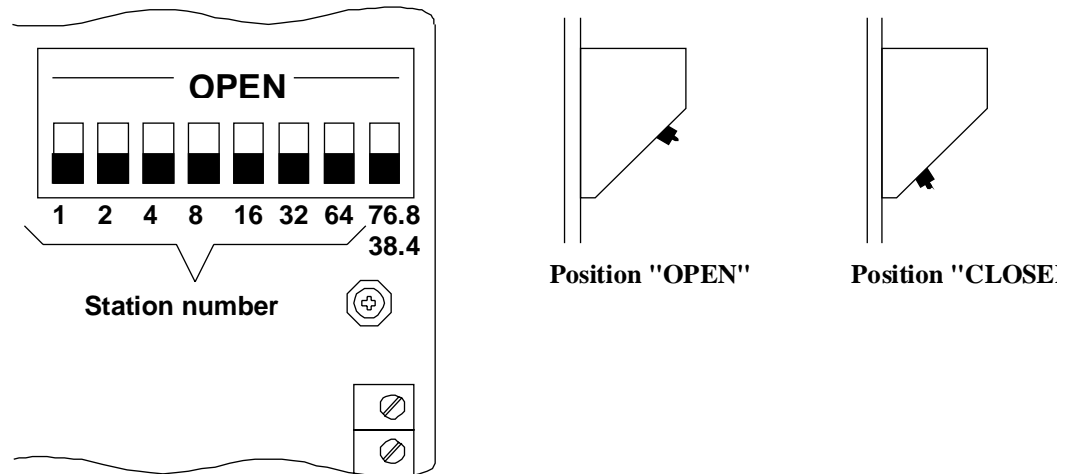
This is displayed by 3 LEDs:

Supply 24 VDC:	yellow LED
Run:	yellow LED
Error:	red LED

The "24 VDC" LED indicates the presence of the external power supply.

When the "Run" LED is on ("Error" LED off) it indicates that the PCD1.RIO is running correctly.

When the "Error" LED is on, it indicates that the PCD1.RIO has an error. This case should never occur in an installation.

**DIL switch**

The 8-position DIL switch on the PCD1.RIO is used to configure the station number and transmission speed.

The transmission speed can be set to either 38.4 or 76.8 kBaud. To select 38.4 kBaud, switch no. 8 (far right) is put in the "CLOSED" position. To select 76.8 kBaud, the same switch is put in the "OPEN" position.

The station number is selected with the first 7 switches. Numbering takes place in binary format. 128 different settings are possible (0 to 127). The number attached to each switch corresponds to the station number, if that switch alone is in the "OPEN" position. For example, if only the 6th of the 7 first switches is in the "OPEN" position, this PCD1.RIO has been configured as station no. 32.

To configure station number 93, as an example, the following switch combination should be made:  $93 = 64 + 16 + 8 + 4 + 1$ . Switches 1, 3, 4, 5 and 7 should therefore be put in the "OPEN" position.

Configuration should take place while the device is powered off.

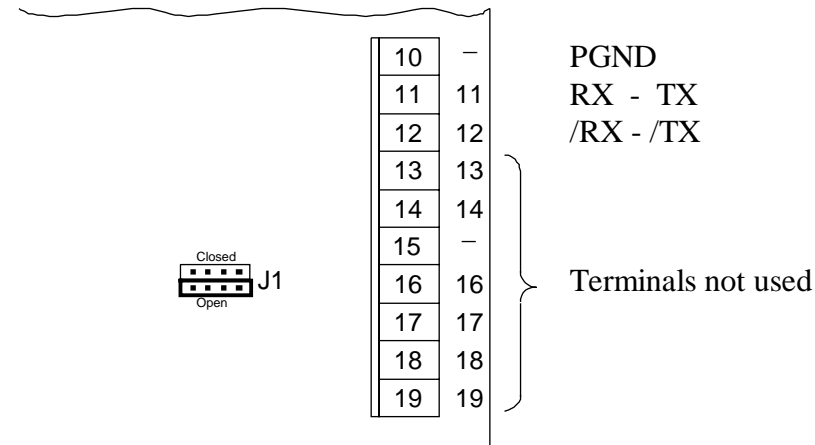


**Caution:** Station numbers 0 and 127 cannot be used. Addressing is limited to station numbers 1 to 126.

## 3.5 The RS 485 interface

### 3.5.1 Connection of the RS 485 interface

The PCD1.RIO is equipped with an interface for connection to the RIO network. S-Bus slave mode is selected automatically, according to the DIL switch settings.



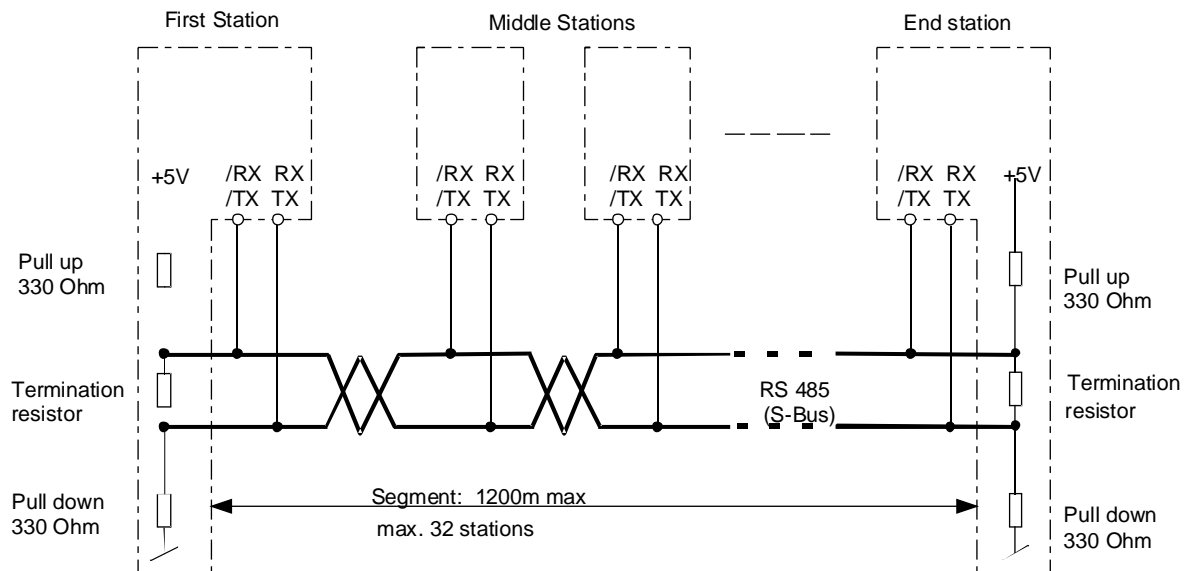
For the correct adjustment of line termination resistors (position of jumper J1: "open" → without termination resistors) and the laying of the bus cable, consult the "Manual of the PCD2 series hardware", "SAIA<sup>®</sup> S-Bus manual" and "Installation components for RS 485 networks".

The main points are shown in the following section. (Extract from the S-Bus manual).



### 3.5.2 Connection and placement of RS 485 bus line

To suppress interference and avoid reflections, pull-up/down line termination resistors must be provided both at the **start and end** of the bus line. These resistors are incorporated in all processor and bus modules and they can be switched on or connected according to choice.



When using the internal resistors of the processor or bus modules to terminate the lines, these stations can not be powered down otherwise the communication over the network is no more possible.

If the network must still continue to work when the first and end stations are not powered, you must use PCD7.T160 termination boxes.

The following points demand special attention:

- When making the bus cable, strict attention is necessary not to mix up the data lines - "RX-TX" must therefore always run to "RX-TX" and "/RX-/TX" to "/RX-/TX".

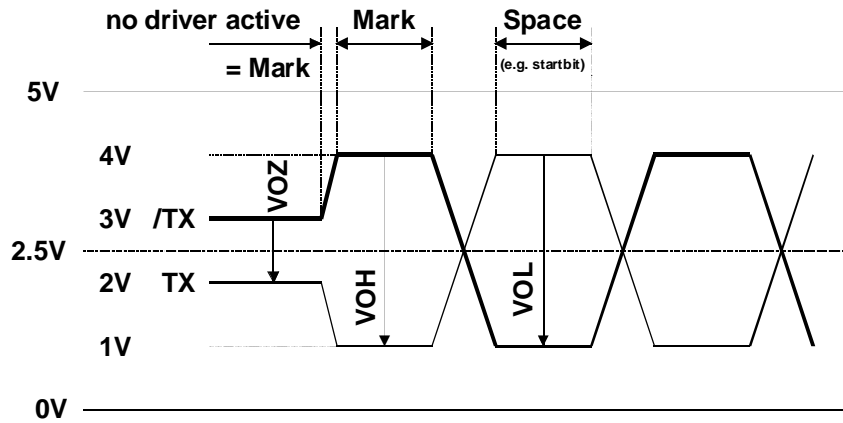
The denominations "RX-TX" and "/RX-/TX" are not always used:

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

- Care should also be taken that the bus line remains continuously connected, even when one or more plugs are pulled out.
- Spur cables (stubs) should not exceed 0.5 m.
- Use stranded cable of at least 0.5 mm<sup>2</sup>, with 2 cores, twisted and shielded.

**Signal levels of the RS 485 interface**

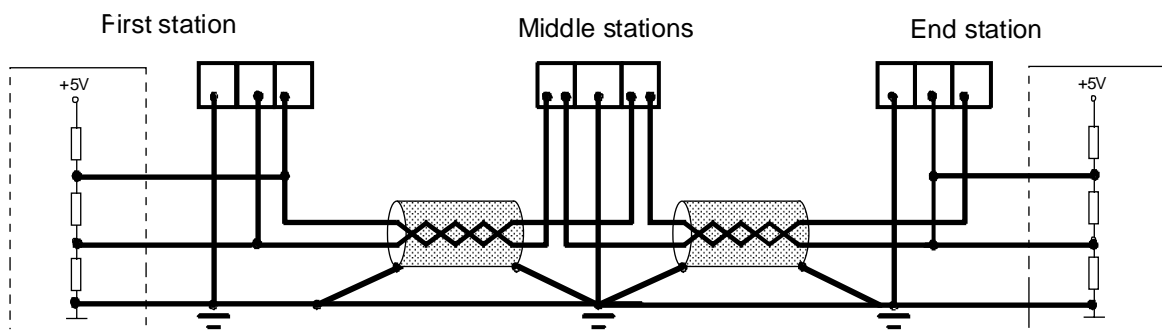
Signal type	Logical state	Polarity
Data signal	0 (space)	RX-TX positive to /RX-/TX
	1 (mark)	/RX-/TX positive to RX-TX



- VOZ = 0,9V min... 1,7V max (no driver active)
- VOH = 2V min (with load) ... 5V max (without load)
- VOL = -2V ... -5V

**Grounding of an RS 485 bus line**

The cable's screening must always be connected at both ends, to produce a continuous, solid earth line, and so reduces potential differences to a minimum.



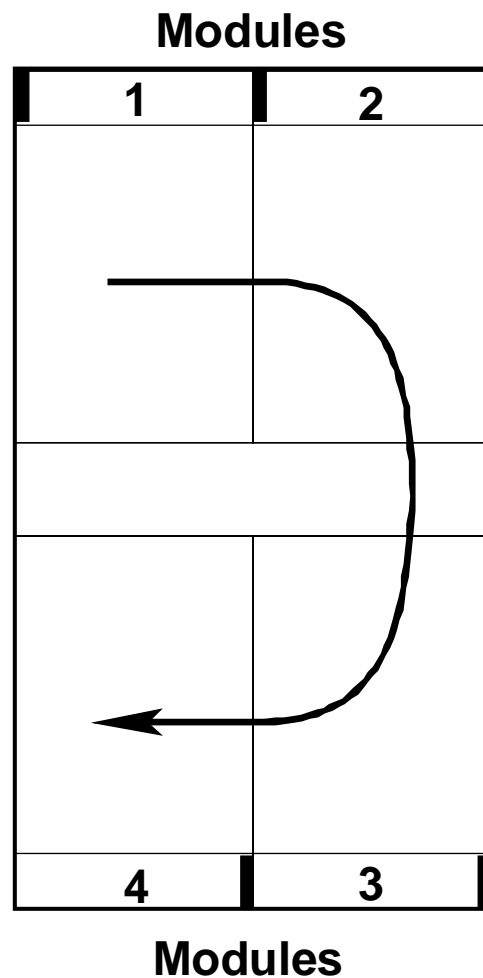
It is recommended that the RS 485 cable is not laid in direct proximity to motor cables which may produce interference, unless these cables are also well screened.

### 3.6 Addressing the input and output modules

---

With the PCD1.RIO module **positions** are designated, not the addresses of individual inputs and outputs, as with the PCD1 or PCD2. The 4 sockets are identified as "Module 1", "Module 2", "Module 3" and "Module 4" and are addressed in the same way during programming or configuration.

Each occupied socket must be assigned a PCD2 series type of module, e.g. A400. This information is passed on by the configurator of the master station in the RIO network (see chapters 4 and 5).



Notes

## 4. Concept of the RIO network

---

### 4.1 General description

---

The fundamental principle of the PCD-RIO is that all slave station media (status of digital inputs/outputs and values of analogue inputs/outputs) are mapped in the master station and transmitted back to the slave stations. The user program in the master station will then handle the data as local media. The task of the RIO processor is then to poll the slaves cyclically in the background in order to refresh the media map and, after processing by the user program, to retransmit to the slaves. The definitions for communication between master and slaves are stored in the data blocks (DB) of the master station.

Either a PCD4.M445, PCD4.Mxx5 or PCD2.M1xx can be used as the master station. The list with the messages to be cyclically exchanged with the slaves is located in the configuration file "RIOCFG.SRC". Refreshing the media map takes place in the background (in firmware) and is unseen by the user.

In time-sensitive installations, a PCD4.M445 as master station offers the advantage over a PCD4.Mxx5 or PCD2 that the user program for the actual process and the user program for communication in the RIO network can each be kept in their own CPU, so that the process under control is not slowed down by the RIO procedure. In this case, the process is handled by CPU 0 and all RIO requirements by CPU 1.

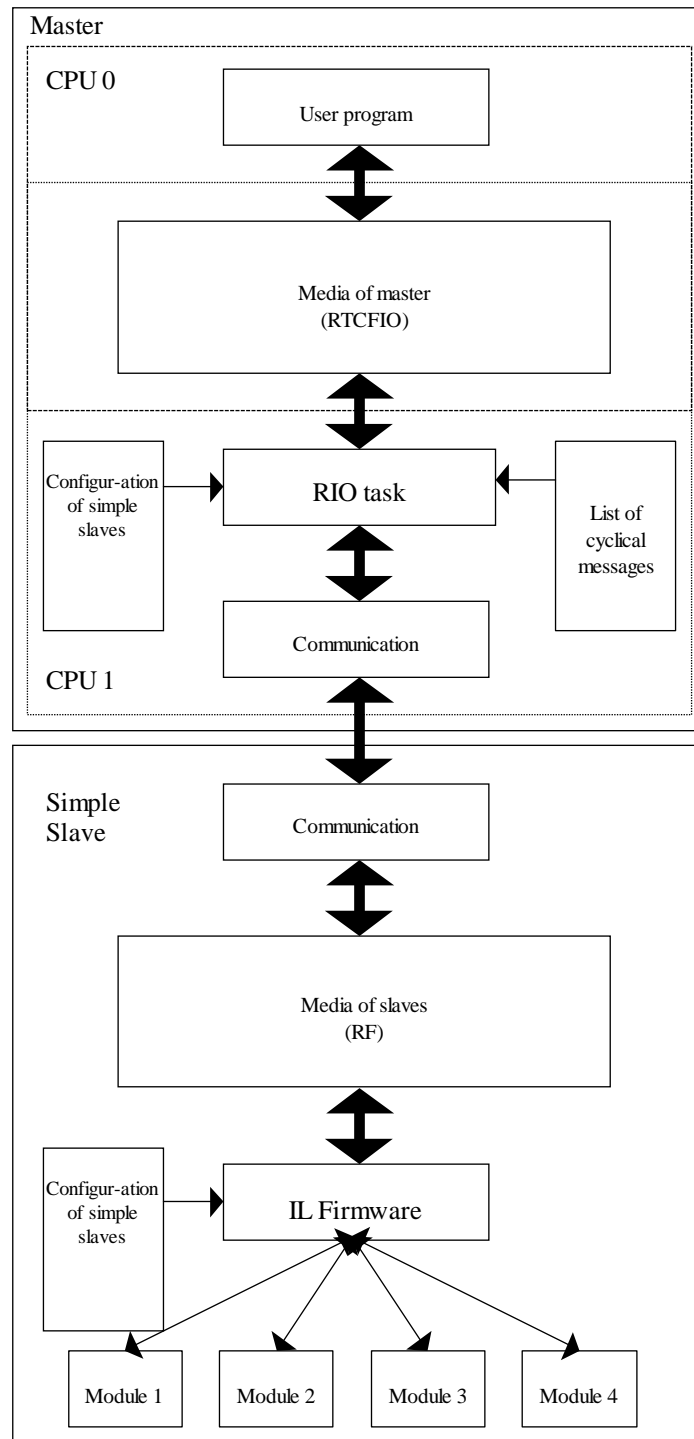
With the help of the configurator (detailed description in chapters 4 and 5) a number of files are created. The first file, "RIO.DEF" (or other selected name xxx.DEF) contains all definitions of slave station resources (EQU). The second file, "RIOCFG.SRC" (or other selected name xxx.SRC) contains the configuration of the whole network.

File "RIOCFG.SRC" will ultimately be linked to the user program which contains the communications relationships of the RIO network. In the case of a PCD4.M445 this is the user program in CPU1, in other cases it is the whole user program.

All user programs which have anything to do with the RIO network must integrate the "RIO.DEF" program with \$INCLUDE.

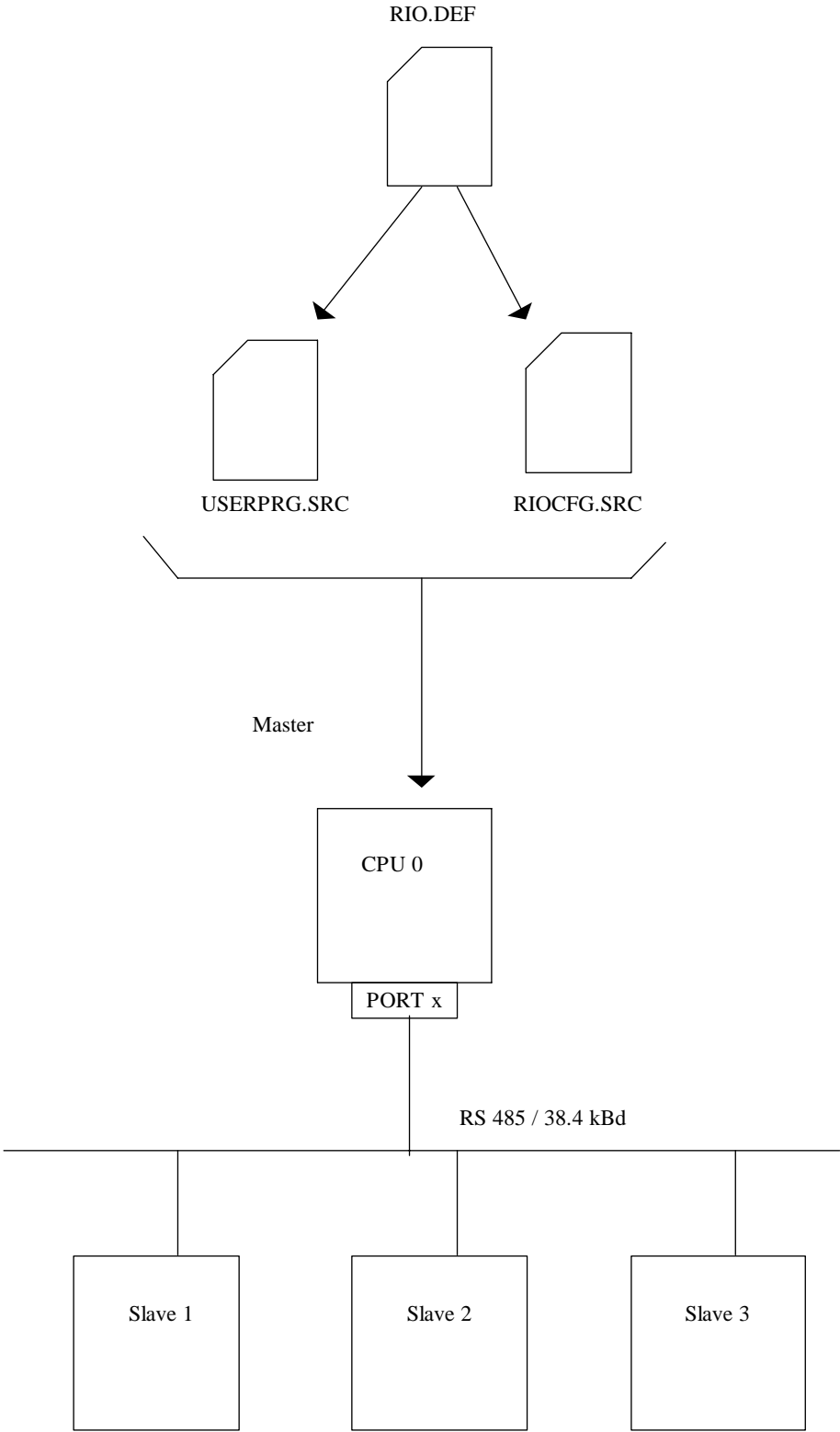
The non-intelligent PCD1.RIO stations are not user programmable.

The following figure shows the operation of a PCD-RIO with a multi-processor CPU (PCD4.445) as master station and a single PCD1.RIO as slave.

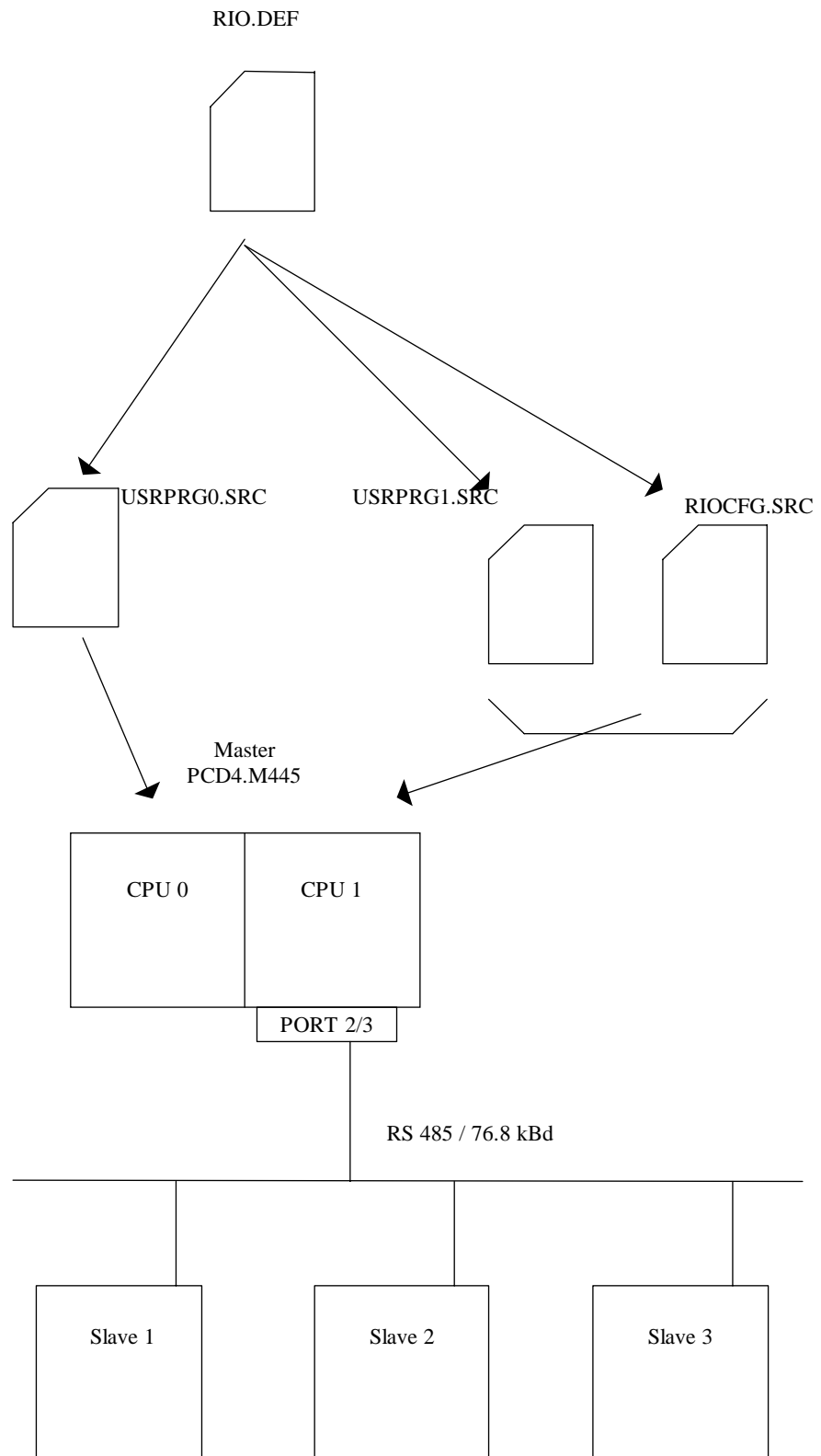


## 4.2 Files for the RIO task

The following drawing shows the various files which must be linked and loaded if a PCD2 or PCD4.Mxx5 is used as the master PCD (single processor CPU).



The following drawing shows the various files which must be linked and loaded if a PCD4.M445 is used as the master PCD (dual processor CPU).





### **Files for RIO requirements**

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

This file contains all declarations (EQU) of all slave station resources needed by the master to complete the RIO task. This file will ultimately be integrated with the directive \$INCLUDE into the user program which handles the communications relationships.

The user must define all resources of all slave stations (digital and analogue I/Os) with a symbolic name. Absolute addressing is not possible in a PCD.RIO network. These symbols are used in the application program and the configuration file "RIOCFG.SRC".

**This file is created automatically in the RIO configurator.**

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

This file contains the configuration of slaves and the definitions needed for cyclically refreshing the media map in the master station. The file also contains the assignment text for the SASI instruction. The file should be linked to the user program of whichever CPU contains the program for RIO requirements.

**This file is also created automatically in the RIO configurator.**

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

This is the user program which, as its name says, is created by the user. The user program contains both the communication relationships of the RIO task and the control of the process. The part of the program with the communication relationships must be edited in IL (Instruction List). The part of the program for process control can be written in the PG3 or PG4. In this case, the RIO task must be linked in the PG4 project manager.

## 4.3 Activation of the RIO task

---

Activation of the RIO task takes place, like all SAIA® PCD communication tasks, with the SASI instruction. The appropriate SASI text is the following:

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

The "UART" assignment is the same as for standard S-Bus. Entry of the timeout and Ts and Tn delays is optional.

The DB number specified at "MODE" indicates the address of the first in a series of DBs, which contain the list of cyclical telegrams. (Rxxx serves to select the S-Bus station, when telegrams are to be exchanged asynchronously, as with SASI S-Bus).

The DB number specified at "CONF" indicated the address of the first in a series of DBs, which contain the configuration of the simple (non-intelligent) slaves. If this heading is undefined, it means that there are no simple slaves in the network. In this case, the processor immediately starts sending the cyclical telegrams.

At "DIAG" the base addresses are defined of 24 diagnostic flags and 4 diagnostic registers. The base addresses of 2 diagnostic DBs can also be specified. Both these DBs contain statistical information about each station in the network. If no diagnostic DB address is indicated, no statistics are drawn up. The addresses of both diagnostic DBs must be equal to or greater than 4000. Further details are in section 4.4.

**This SASI text is created automatically in the RIO configurator.**

After processing the SASI instruction, the RIO interface is assigned and the simple slave stations are configured. After the configuration phase, the cyclical telegrams are transmitted continuously, even when the CPU is in STOP, COND.-RUN or TRACE.

Assignment is "RM1" mode also allows acyclical telegrams to be sent with the instructions STXM and SRXM (see section 4.5).

Before RIO communication is activated, the user can assign this (RIO) interface in conventional S-Bus mode, so that certain preliminary adjustments can be made with STXM and SRXM.

## 4.4 Diagnostic possibilities

---

The many diagnostic possibilities are summarized in chapter 7 and explained with the help of a practical example.

Diagnosis can be divided into the following groups:

- Monitoring the presence of the master station
- Monitoring the presence of the slave stations (XOB 30)
- Diagnostic flags for rough diagnosis of communications
- Diagnostic register for fine diagnosis of communications
- Diagnostic data blocks for RIO task monitoring and statistics
- Data blocks for configuring the stations
- Data blocks for configuring the cyclical telegrams

## 4.5 Use of the RIO channel for asynchronous telegram transmission

---

By means of the user program, the RIO communication channel can also be used to transmit S-Bus telegrams with the instructions STXM and SRXM. These telegrams for the asynchronous transmission of information to intelligent slaves are sent between two cyclical messages.

As soon as the communication interface is assigned in "RM1" mode, asynchronous telegrams can be transmitted in normal S-Bus mode. The overlaying of both communications modes is not visible to the user. This technique is indicated briefly in example "TEST2", task point 9), section 6.2.

**The transmission of acyclical telegrams by the user program slows down the refreshing of the media map.**

## 4.6 Refresh and reaction times

---

The reaction time ( $T_r$ ) depends on 4 different factors:

- Delay produced by the input filter ( $T_{dio}$ )
- Refresh time ( $T_{rfr}$ )
- Program cycle time ( $T_{plc}$ )
- Delay depending on the output switching time ( $T_{dio}$ )

The delay which arises from the input filter and the output switching time should be ignored in any initial consideration.

The refresh time " $T_{rfr}$ " is the time for transmitting all messages to the slave stations or, in other words, the time which elapses between 2 similar telegrams.

The cycle time of the PCD user program " $T_{plc}$ " is the time necessary for all instructions in the user program to be processed once.

The reaction time of an installation will be as follows:

$$T_{plc} < \text{Reaction time of installation} < (2 * T_{rfr} + 2 * T_{plc})$$

The following table shows the refresh time ( $T_{rfr}$ ) for various PCD systems with a varying number of media to transmit.

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 Firmware required

---

To make the most of the possibilities offered by RIO communication, a firmware must be used which can cope with RIO outputs.

The first firmware versions are:

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

## 5. The RIO configurator: PCD8.C3..

---

### 5.1 Introduction

---

The configurator is an indispensable tool for the successful creation of a PCD.RIO application. It is used to configure the entire RIO task:

- Configuration of the network with one master and a number of non-intelligent or, possibly, intelligent slaves
- Definition of PCD types for each RIO station
- Definition of station numbers
- Definition of transmission speed
- Definition of equipment on each individual non-intelligent slave station
- Definition of messages for communication with intelligent slaves
- Definition of base addresses of all diagnostic and assignment resources
- Creation of the RIO.DEF file with all EQU declarations for the media map in the master station
- Creation of the SLAVExx.DEF file with the EQU declarations for each individual intelligent slave station
- Creation of the RIOCFG.SRC file with the configuration of slaves
- Preparation of the SASI text for assigning the task

The configurator runs under WINDOWS 3.1 or WINDOWS 95.

The present initial version of the RIO configurator is a stand-alone version, i.e. the configurator is not integrated into the PG3 or PG4. Management of the resources therefore only concerns those used by the configurator itself. It is the responsibility of the programmer to prevent conflicts between the RIO task's data and other parts of the program. The aim is to incorporate a later version of the RIO configurator into the PG4 and to manage all resources in common.

The present version includes comprehensive Help according to the WINDOWS philosophy, in English. All items are listed in the present manual.

## 5.2 Installing the RIO configurator

---

No install program exists for the present version, so installation must be done manually. The configurator is supplied on 2 diskettes and takes up approx. 4 MBytes on the hard disk.

To install the configurator, proceed as follows:

Installation in C:\SNET

1. Insert diskette 1 in drive A: and load the file SNET.INI from DOS into an ASCII editor. Edit the path and directory:

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

(or according to the installation desired)

2. Copy both files CTL3DV2.DLL and SNET.INI into the \WINDOWS directory.
3. Unzip the compressed installation files with:

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

Insert the appropriate diskettes according to the instructions on the screen.

4. Select the WINDOWS program group in which the RIO configurator is to be located.

With 'File' - 'New...' - 'Program' - 'OK'

At 'Description' enter the title of the RIO configurator,

e.g. "PCD-RIO"

At the command line

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

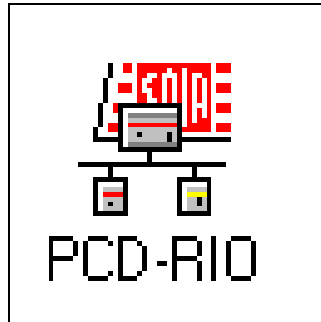
enter (or 'Browse...') and 'OK'.

The icon for the RIO configurator appears. Installation is complete.

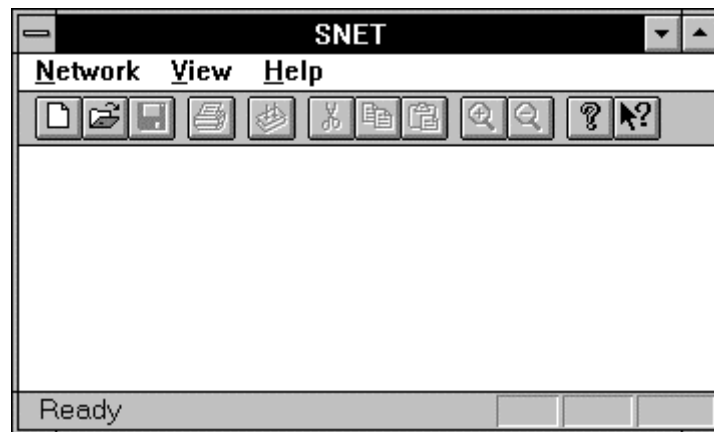


## 5.3 Running the RIO configurator

---



After double-clicking on the "PCD-RIO" icon, the main menu of the RIO configurator appears:

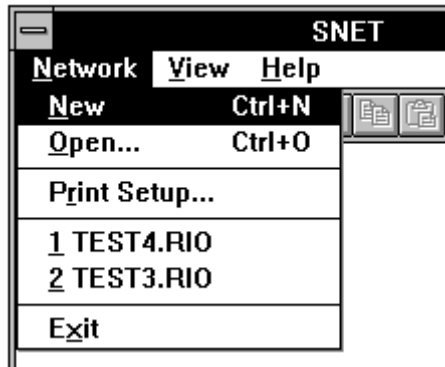


The following is a brief description of all menus and submenus in the order in which they appear on the screen.

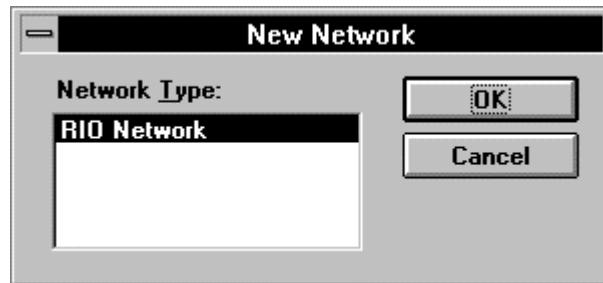
A description of the menus as they are used in practice is given in chapter 6: "Programming".

The toolbar is described at the end of this chapter, in sub-section 3.3.10. The individual function keys are shown at the appropriate menu items.

### 5.3.1 'Network' submenu



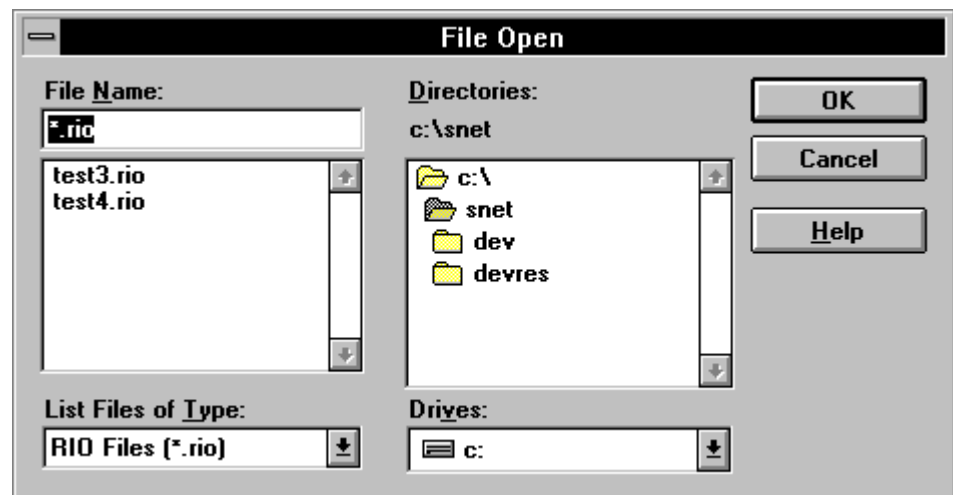
'New': Clicking here displays the following window:



As present it is only possible to configure a RIO network. In a later version, a PROFIBUS DP network may also be available for selection.

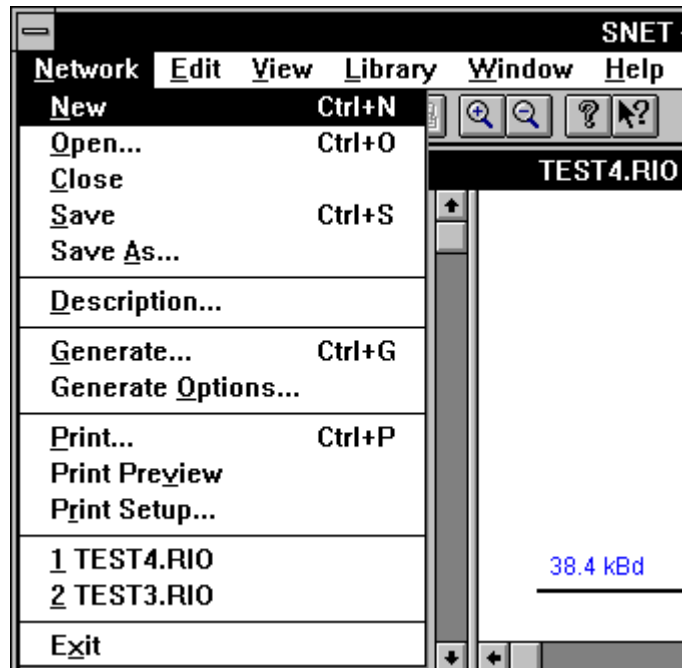
After 'OK', actual configuration begins. This takes place in the following section.

'Open...': If configurations have already been made and saved, the following window appears:



Configurations created earlier can be displayed.

Once work is in progress on a configuration, the 'Network' menu appears as follows:



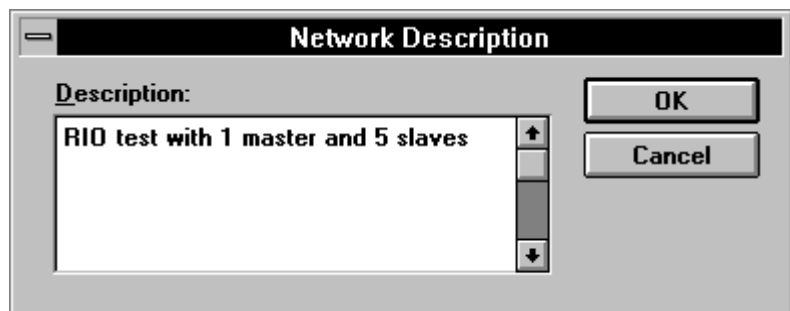
**'Close':** Closes the active network document.

**'Save':** Saves the network document, if it already has a name



**'Save as...':** Saves for the first time, or saves under a different name.

**'Description':** Field for a description of the network.



**'Generate':** Generates the files RIO.DEF, SLAVExx.DEF and RIOCFG.SRC at the end of configuration (later)



**'Generate Options...':**

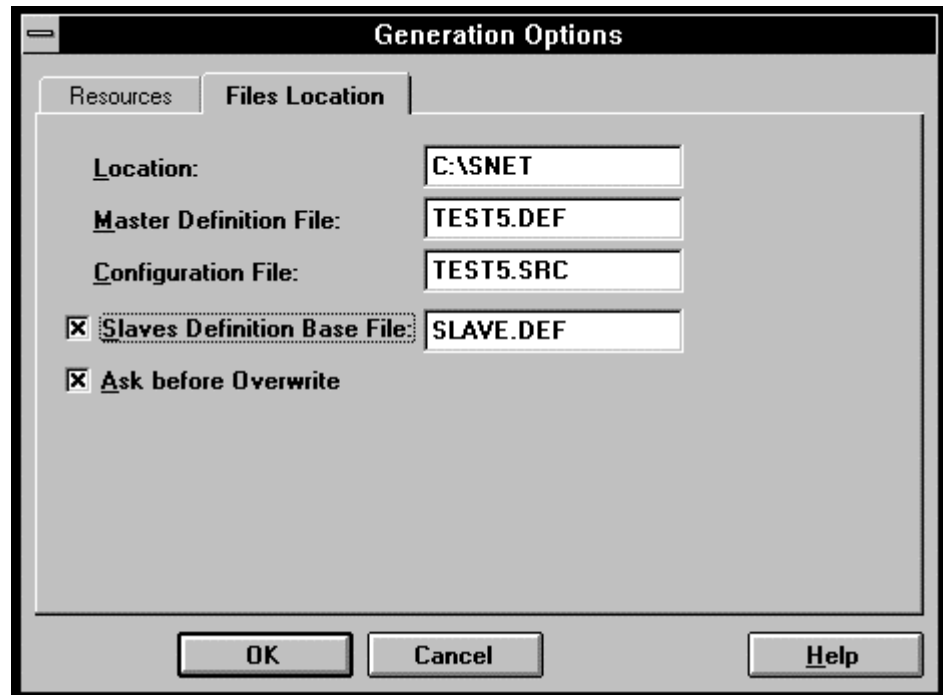
The base addresses and symbolic names for the base resources of various configuration and diagnostic media are defined here. A standard setting is indicated and can be accepted.

	Address:	Symbol:	
First <u>C</u> onfiguration DB:	10	db_cfg	
First <u>M</u> essages DB:	32	db_msg	
First <u>D</u> iagnostic DB:	4500	db_diag	<input checked="" type="checkbox"/> <u>U</u> se DB diags
First Diagnostic <u>F</u> lag:	500	f_diag	
First Diagnostic <u>R</u> egister:	500	r_diag	
<u>S</u> ASI Text Number:	100	txt_rio	
<u>P</u> artner Stn Nb Register:	100	r_dst	

Buttons: OK, Cancel, Help

'Resource': The significance of the individual media groups will be explained later. It should be stressed here again that the programmer has to take care that no conflicts arise between the media groups.

'File Location': The names and directory of RIO files is specified here.



Location: Name of the directory in which the files are stored for the RIO task.

Master Definition File:

Name of the definition file with the EQU declarations for the media map in the master station. With "\$include" this file is incorporated into all files relevant to the RIO task.

Configuration File:

Name of the configuration file with the slave configurations. This file is linked with the user program.

Slaves Definition Base File:

This file is optional. It contains the EQU declarations for each individual intelligent slave station. With "\$include" it is incorporated into all files relevant to the intelligent slave stations of the RIO task. The station name appears in the file name, e.g. for station 15: "SLAVE15.DEF".

Ask before Overwrite:

When this is switched on, a query is returned before overwriting a file.

**'Print':** Prints out the network and the configuration.  
(see ...)



**'Print Preview...':**

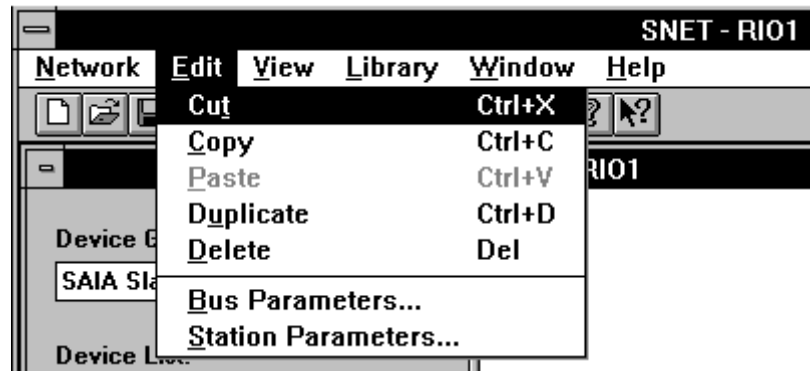
Shows a preview of the printed page.

**'Print Setup...':**Configures the printer

**'Recent Files':** Shows the last 4 configuration files (if they already exist).

**'Exit':** Exits the RIO configurator.

5.3.2 'Edit' submenu



'Cut': 'Cut out'



'Copy': 'Copy'



'Paste': 'Insert'



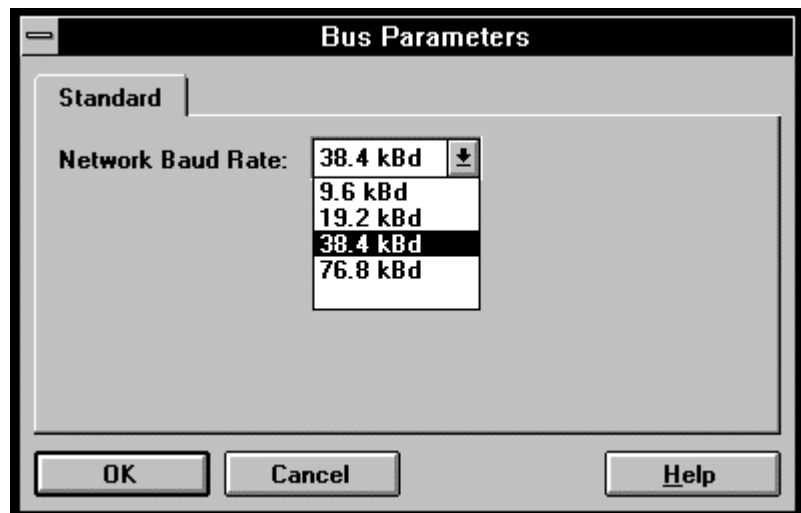
'Duplicate': 'Duplicate'

'Delete': 'Delete'

'Bus Parameters': 'Standard':

The standard baud rate is 38.4 kBd. Another baud rate can be selected. If the baud rate chosen is too high, a message is displayed, then the highest permissible baud rate is set automatically.

Standard  
WINDOWS  
functions

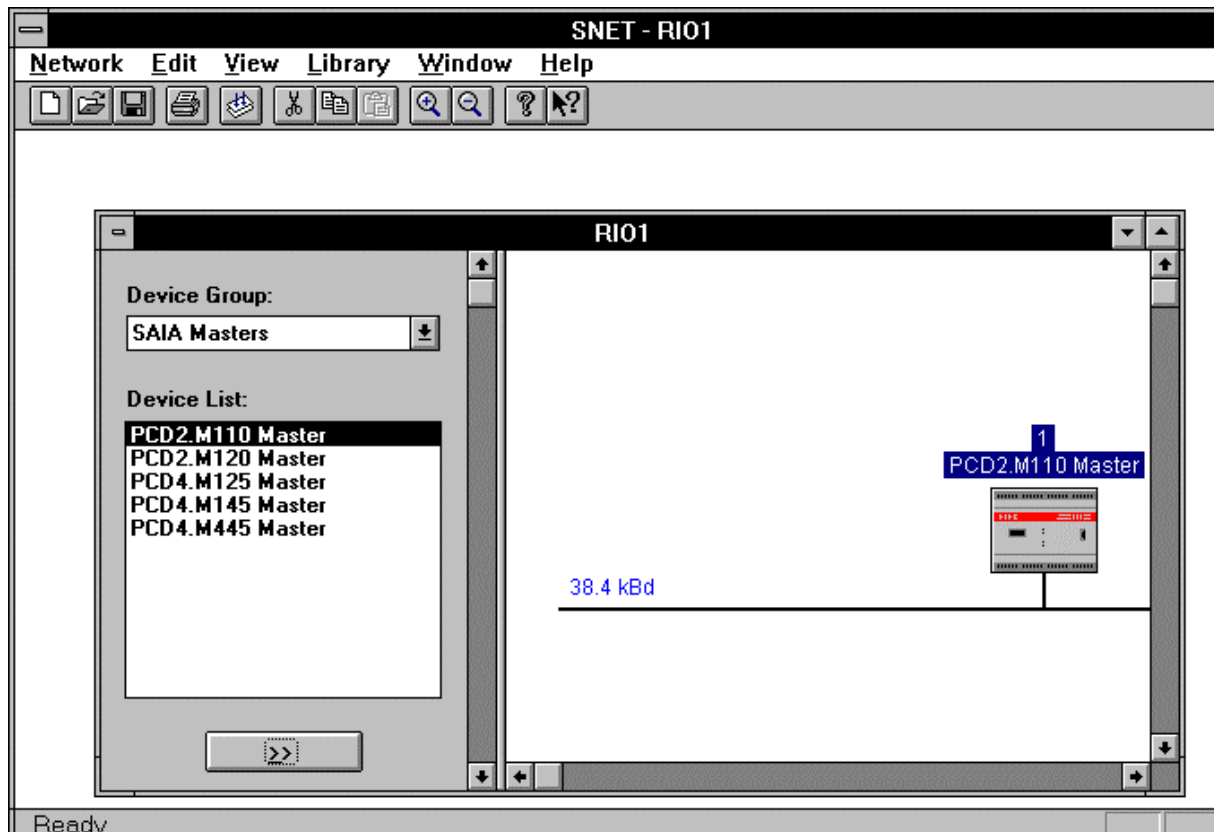


The previous window can also be opened by double-clicking on the bus line (see next figure).

When a PCD type has been selected from the list of possible master stations and inserted into the configuration field by activating the button:



it is then possible to define the station parameters for the master station.



### 'Station Parameters...':

This important, comprehensive menu item is divided here into 3 independent subsections:

- 5.3.3 'Station Parameters...': (for the master station)
- 5.3.4 'Station Parameters...': (for a non-intelligent slave station)
- 5.3.5 'Station Parameters...': (for an intelligent slave station)



### 5.3.3 'Station Parameters...': (for the master station)

'Station': It is possible to define the S-Bus timeout, the port and the CPU charge.

This window can also be opened by double clicking on the master station in the configuration field.

The screenshot shows a dialog box titled "Master 1 'PCD2.M110 Master' Parameters". It has three tabs: "Station", "Device", and "Bus". The "Station" tab is selected. The fields are as follows:

- Name:** PCD2.M110 Master
- Address:** 1
- Timeout:** 0 ms
- Port:** 1 (with a dropdown arrow)
- Symbol:** port\_rio
- CPU charge:** 50% (with a dropdown arrow)

Buttons for "OK", "Cancel", and "Help" are located at the bottom of the dialog.

**Name:** Name or type of master station  
(max 32 characters)

**Address:** Address (station number) in RIO network.  
Address range: 1 - 254

**Timeout:** When = 0, S-Bus standard values are used

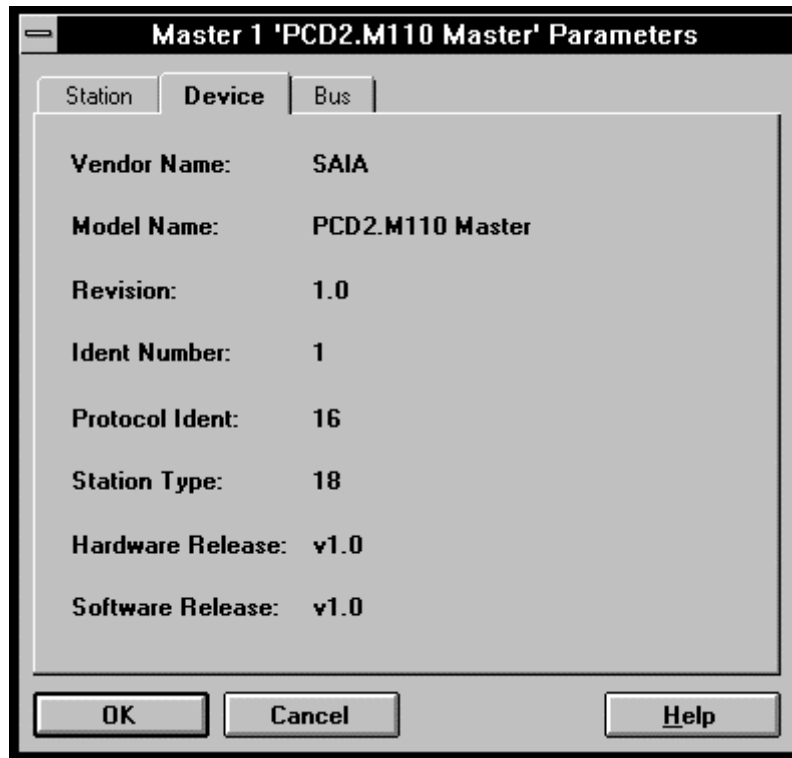
for 9600 Bd	→	250 ms
for 19,200 Bd	→	200 ms
for 38,400 Bd	→	100 ms
for 76,000 Bd	→	50 ms

**Port:** Port used for the RIO task.

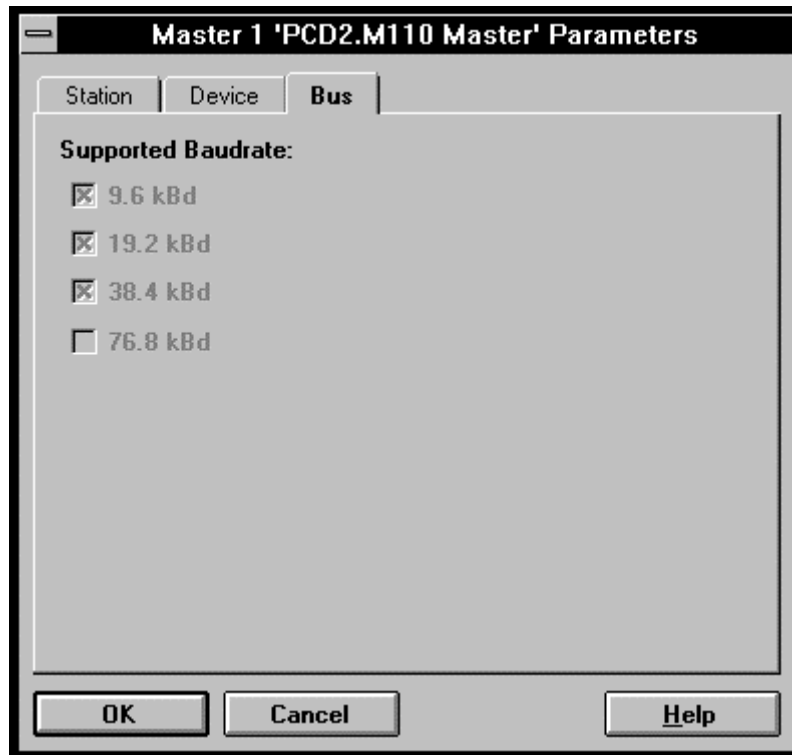
**CPU charge:** 25%, 50% or 75%

Defines the CPU overhead for communications. The higher the value, the slower the user program. (Not yet implemented).

'Device': All values refer to the selected master station.  
Nothing can be modified.



'Bus': All details are dependent on the station selected  
and cannot be modified.

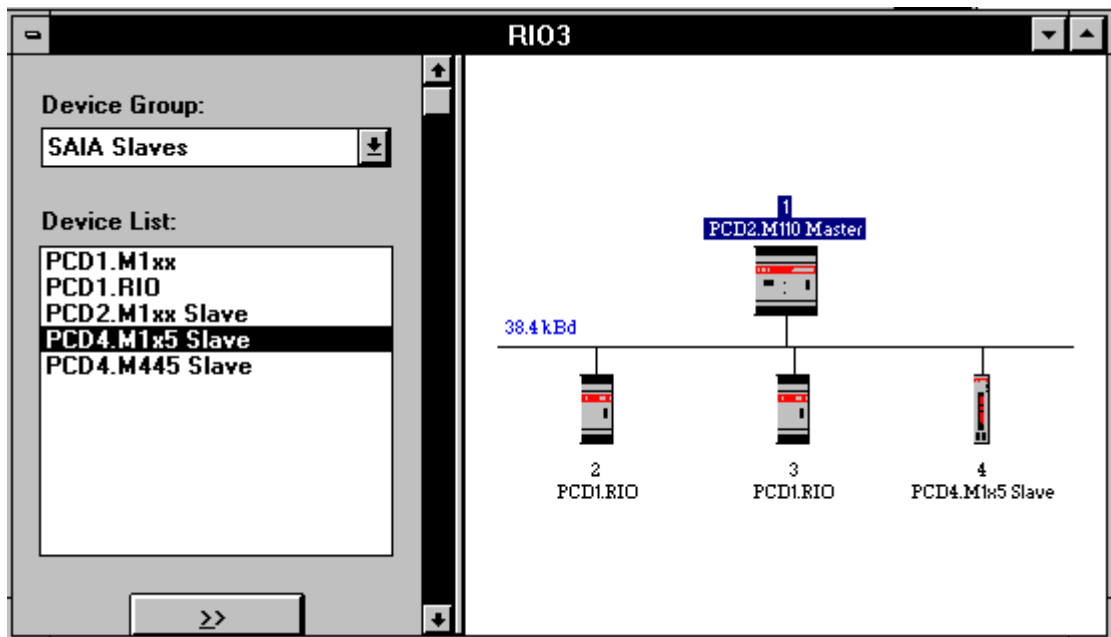


When one or more PCD types from the list of possible slave stations have been selected and each inserted into the configuration field with the button:



the station parameters for the slaves can then be defined.

From the 'View' menu, select 'Zoom to Fit' to obtain the screen shown below (explanation follows in section 5.3.6).



The configuration of the PCD1.RIO non-intelligent slave stations is different from that of the intelligent slave stations, such as PCD1.M1xx, PCD2.M1xx, PCD4.M1x5 or PCD4.M445.

For the non-intelligent slaves, the precise equipment must be specified, so that the master station can configure the media mapping and prepare cyclical communications.

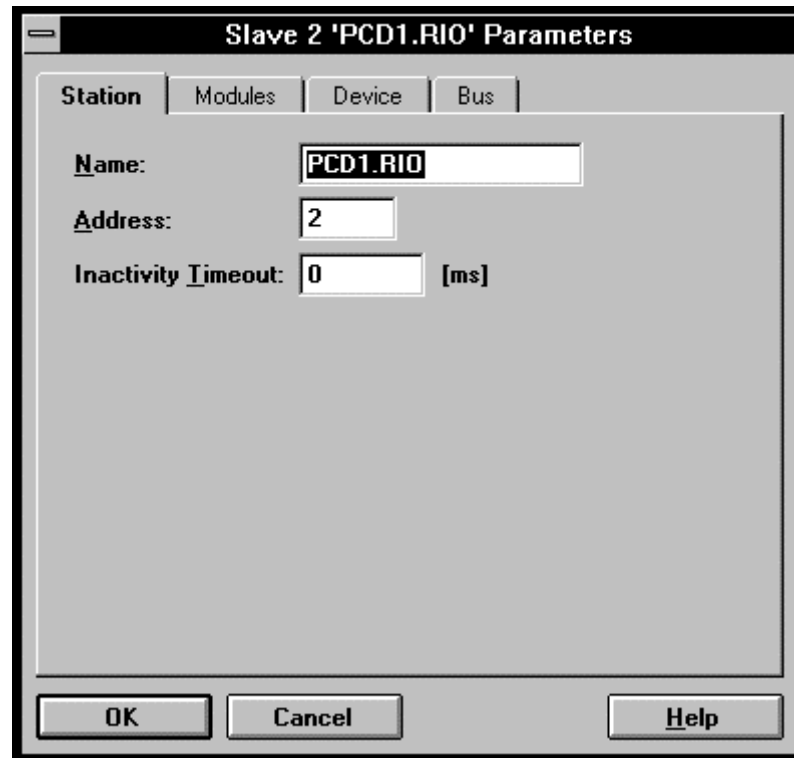
For the intelligent slaves, it is necessary to define which resources are to have a RIO communications relationship.

We begin with the non-intelligent stations, since in practice these will usually be the standard case.

### 5.3.4 'Station Parameters...': (for a non-intelligent slave station)

'Station': For the present station, the address and timeout can be modified.

This window can also be opened by double-clicking on the corresponding slave station in the configuration field.



Name: Name or type of station  
(max. 32 characters)

Address: Address (station number) in RIO network.  
Address range: 1 - 126

Timeout: It is possible to define the resetting of outputs and zeroing of analogue value registers after a communications error.

Values allowed: 50 - 3,600,000 ms (1 hour) in increments of 10.

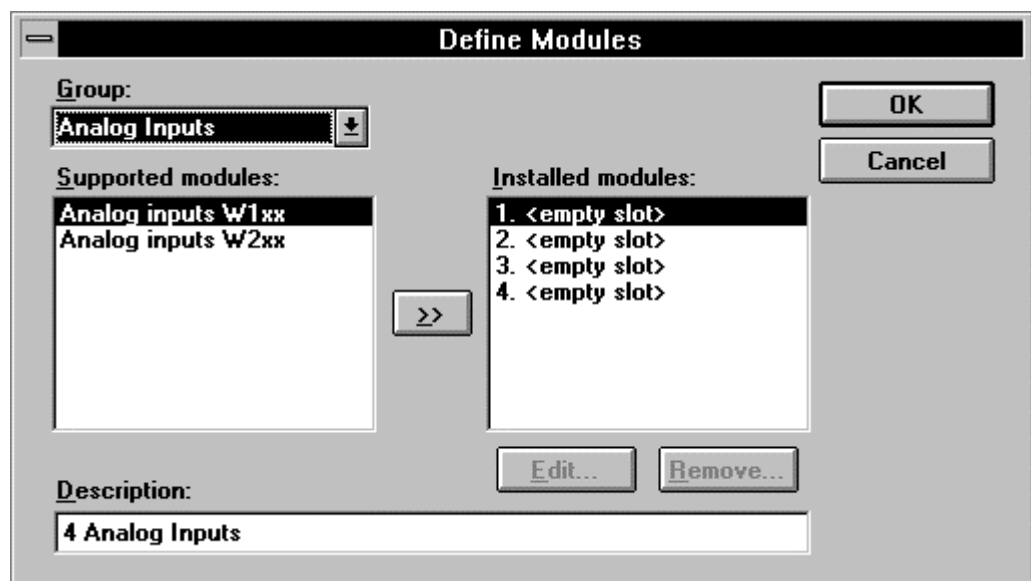
If the value zero is selected, the switch-off function is not enabled.

'Modules': In this window, the equipment of the RIO station is defined with 'Define Modules...'

With 'Media Map...', it is possible to view the elements of the media map and, if necessary, adjust them.

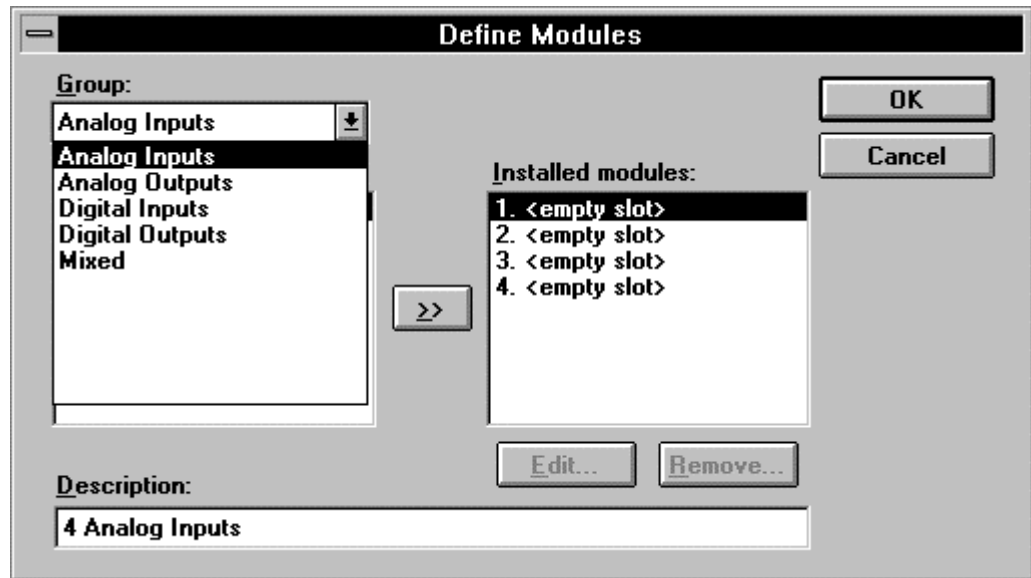


After clicking on the 'Define Modules...' switch, the following window appears:

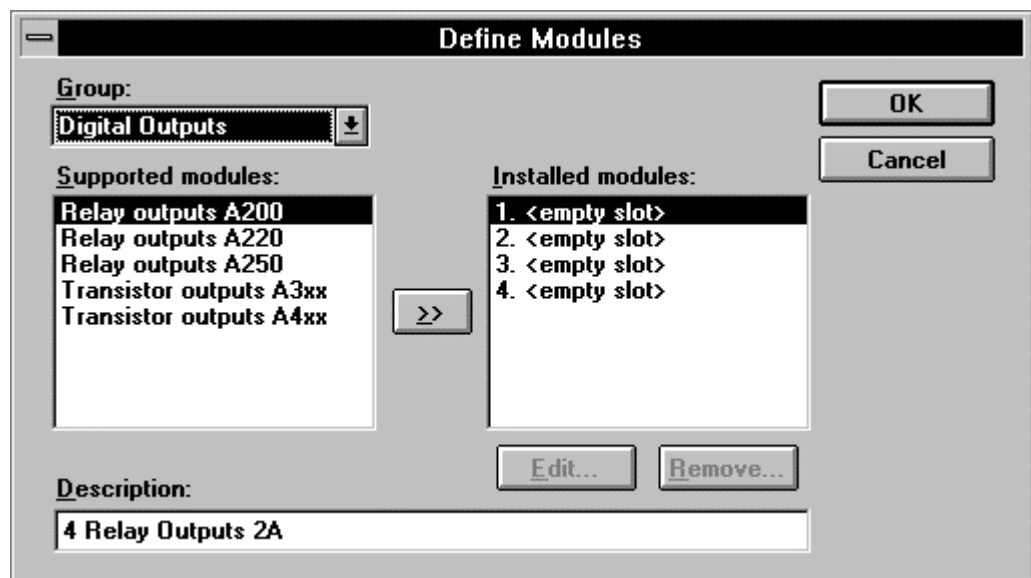


'Installed modules': The selected module has 4 sockets, all of which are still empty <empty slot>

'Group': It is possible to select the range of modules which can be used (in alphabetical order, starting with 'Analog Inputs')



For example, if 'Digital Outputs' is selected, all usable digital output modules appear under the heading 'Supported modules'.

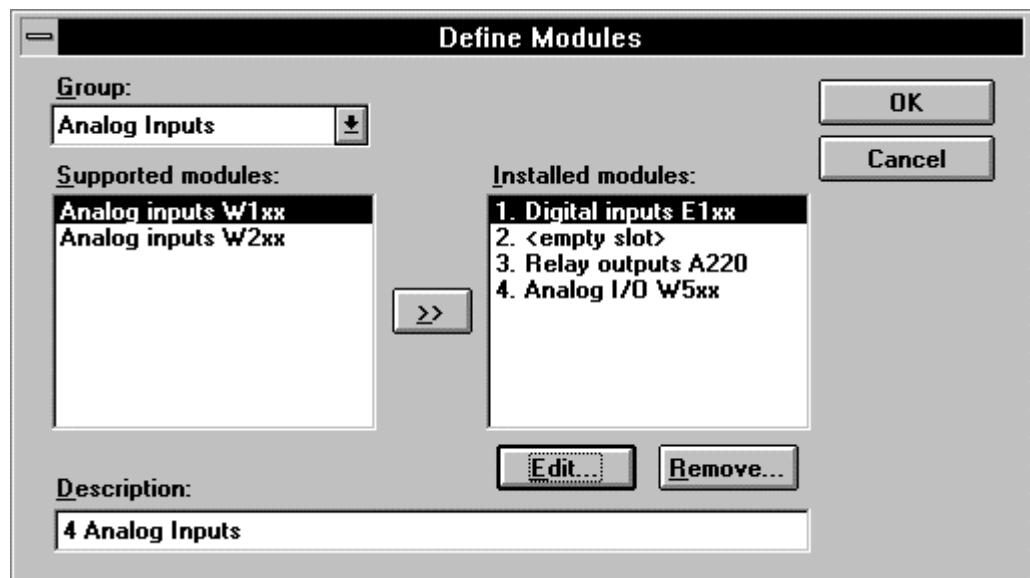


Let us define the following equipment as an example:

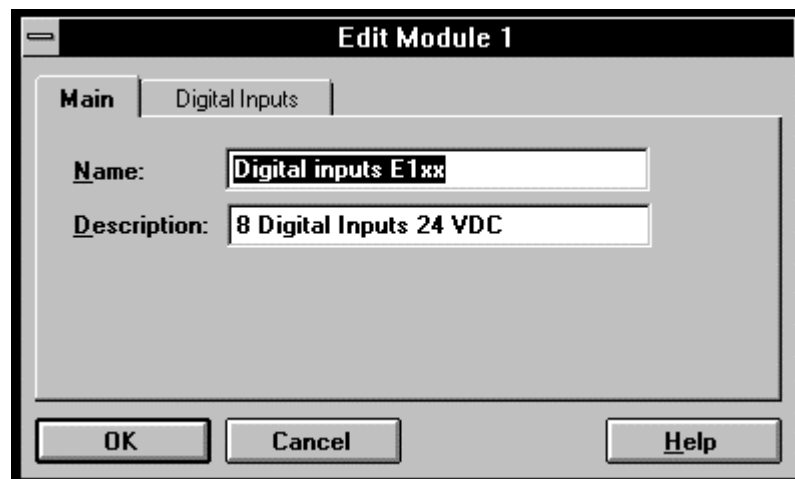
- Slot 1: 8 digital inputs PCD2.E110
- Slot 2: empty
- Slot 3: 6 digital relay outputs PCD2.A220
- Slot 4: analogue I/O module PCD2.W5xx

From 'Group': select 'Digital Inputs', then from 'Supported modules': select 'Digital inputs E1xx'. At 'Installed modules' the first slot is selected and, using the double-arrow button, the module defined is inserted.

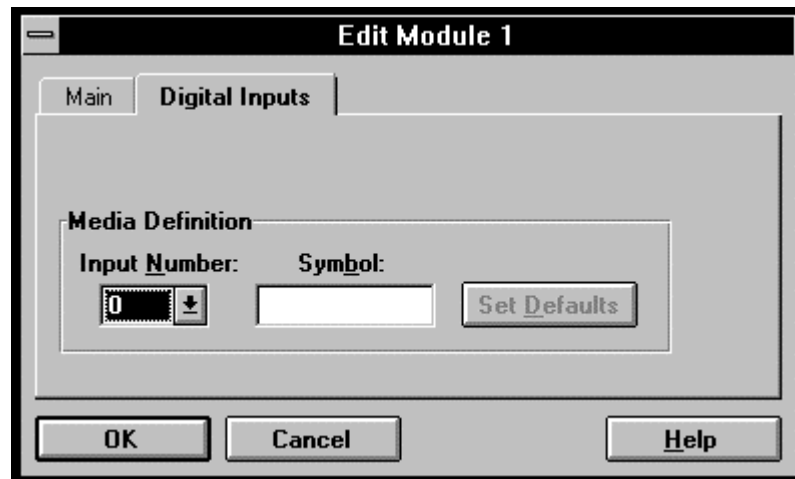
The same procedure is followed for the other modules, so that the following appears under 'Installed modules' after 'OK'.



The individual modules must now be edited. Starting with the digital inputs, after 'Edit...' the following appears for 'Main':



Click on 'Digital Inputs'. The following window appears:



It is now necessary to assign a symbol to each resource used in the installation, i.e. all resources of all stations, with the same symbol only allowed to appear once in the whole network. Direct addressing, such as is often used for small tasks on a single PCD, is not possible here even for testing purposes.

Syntax for symbols:

- Symbols must start with a letter (a-z, A-Z).
- A symbol must comprise at least 2 and at most 10 characters.
- The underline character "\_" is allowed, but not as the first character.
- .,:;\ etc. are not allowed.
- The "ß" (German double-S) must not be used.
- No assembler declarations (PUBL, AUTO, EQU...) and no mnemonics (STH, COB, BITL...) can be used as symbolic names.

To retain clarity even with the larger networks, good organization of symbol names is imperative.

For our examples, the symbols have been chosen so that both the station number and the resource addresses are immediately recognizable. For example, the digital outputs on the 3rd slot of station no. 25 are identified with

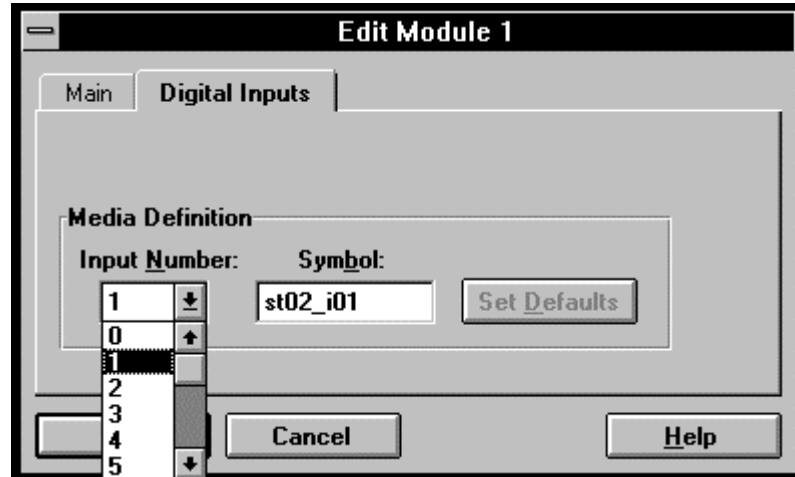
```
st25_o16 ... st25_o23
```

and the 1st analogue input (input 0) on the 2nd slot of station 13 with

```
st13_ai08
```



It is preferable to provide all the elements of a module with a symbol. Although it might only be necessary to define those elements which are actually used in the process, it is still probably more convenient, for later use also, if all elements are defined.



If the elements are defined consecutively, as suggested, <Ctrl> + <C> can be used to call the first element into the temporary store and <Ctrl> + <V> to insert it again in the adjacent field and verify it.

Since editing the modules of even quite a small RIO task is a very laborious business, a simplified process has been implemented. This simplification consists of specifying the symbolic name of the first element of the module, subsequent I/Os are then automatically assigned the same name followed by consecutive numbers.

Example:      Element 0:      Symbol:      st02\_i16

After clicking on 'Set Defaults' (only active if "Input Number" <0> is selected) the following elements are automatically defined as

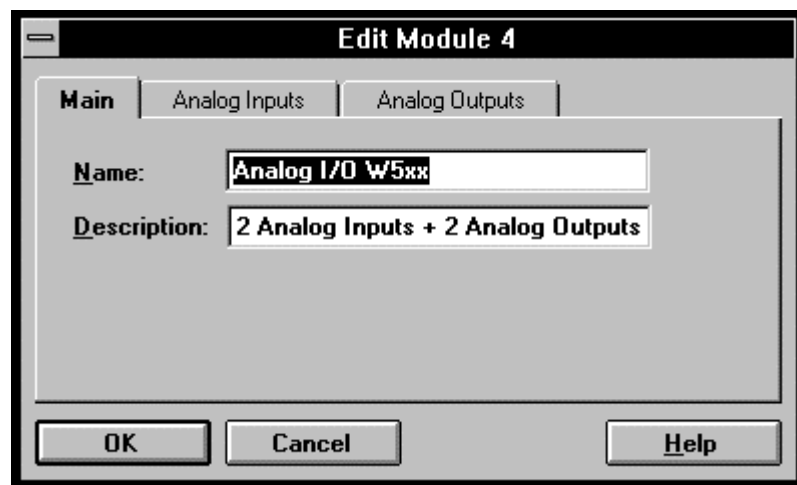
Element 1:	Symbol:	st02_i17
Element 2:	Symbol:	st02_i18
Element 3:	Symbol:	st02_i19
Element 4:	Symbol:	st02_i20
Element 5:	Symbol:	st02_i21
Element 6:	Symbol:	st02_i22
Element 7:	Symbol:	st02_i23

If the symbolic name does not end with a number, numbering starts from 0.

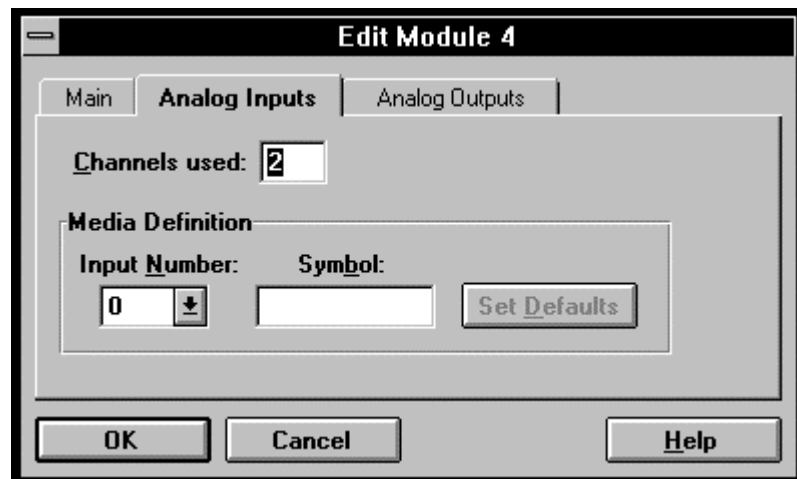
For example, if the symbolic name "Motor" is defined, after 'Set Default' → "Motor0" to "Motor7" or with "ABC\_7" → "ABC\_7" to "ABC\_14" etc. (for modules with 8 elements).

Other digital I/O modules are edited in the same way.

Analogue modules are edited in a similar way. Let us edit module no. 4, a mixed module with inputs and outputs. After clicking on the 4th module in the list of 'Installed modules' the following window is displayed:



After clicking on 'Analog Inputs' the following window appears:



It is possible to specify whether one or both channels are to be used. Afterwards, the symbolic names are entered in the usual way. Each symbol defines a register in the analogue modules.

The procedure is the same for the analogue outputs.

In the 'Slave Parameters' window, clicking on the 'Media Map...' button displays the following window:

Digital Inputs		Digital Outputs	
Base Media Type:	Flag	Base Media Type:	Flag
Base Media Address:	64	Base Media Address:	2064
Used Media:	8	Used Media:	8

Analog Inputs		Analog Outputs	
Base Media Type:	Register	Base Media Type:	Register
Base Media Address:	64	Base Media Address:	2064
Used Media:	2	Used Media:	2

This defines the base addresses of the flags and registers which are used for the media map in the master station. The base addresses can be modified here if necessary.

Let us return to the main window: 'Station Parameters...'

'Device': Clicking on 'Device' displays a list of station-specific data, which cannot be modified.

'Bus': Clicking on 'Bus' displays a list of possible baud rates. Nothing can be modified here either.

### 5.3.5 'Station Parameters...': (for an intelligent slave station)

'Station': The address of the present station can be adjusted.

This window can also be opened by double-clicking on the appropriate slave station in the configuration field.



In contrast to the configuration window for non-intelligent slave stations, the modules are not configured here. Instead, under the heading 'Messages', groups of media are defined which are handled cyclically, i.e. included in the media map.

'Messages': The following window appears:



To define media groups, click on 'Add...'. The following window appears:



'Function': The possible relationships between the master and the intelligent slave are specified and can be selected.

'Description': The selected relationship is displayed.

'Media definition':

Number of media to Transfer:

Number of related media to be transferred in a single group

Position, Symbol:  
(Set Defaults)

Definition of the individual media with symbolic names. The same rules apply as for non-intelligent slaves.

'Media Map':

First Master media Type/Address:

Media type and base address of media group in master

First Slave media Type/Address:

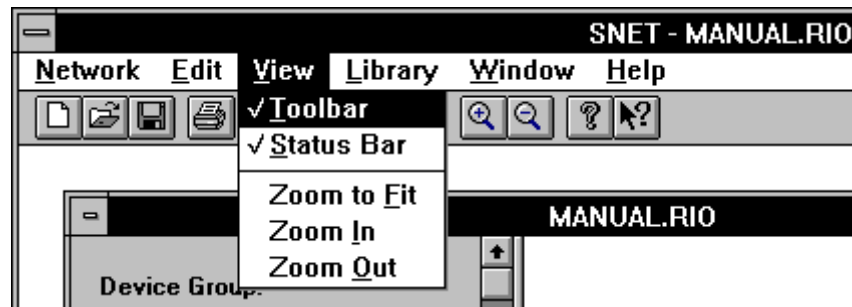
Media type and base address of media group in intelligent slave station

An example of this is given in chapter 6.2: "Further example".

'Device': Clicking on 'Device' displays a list of station-specific data, which cannot be modified.

'Bus': Clicking on 'Bus' displays a list of possible baud rates. Nothing can be modified here either.

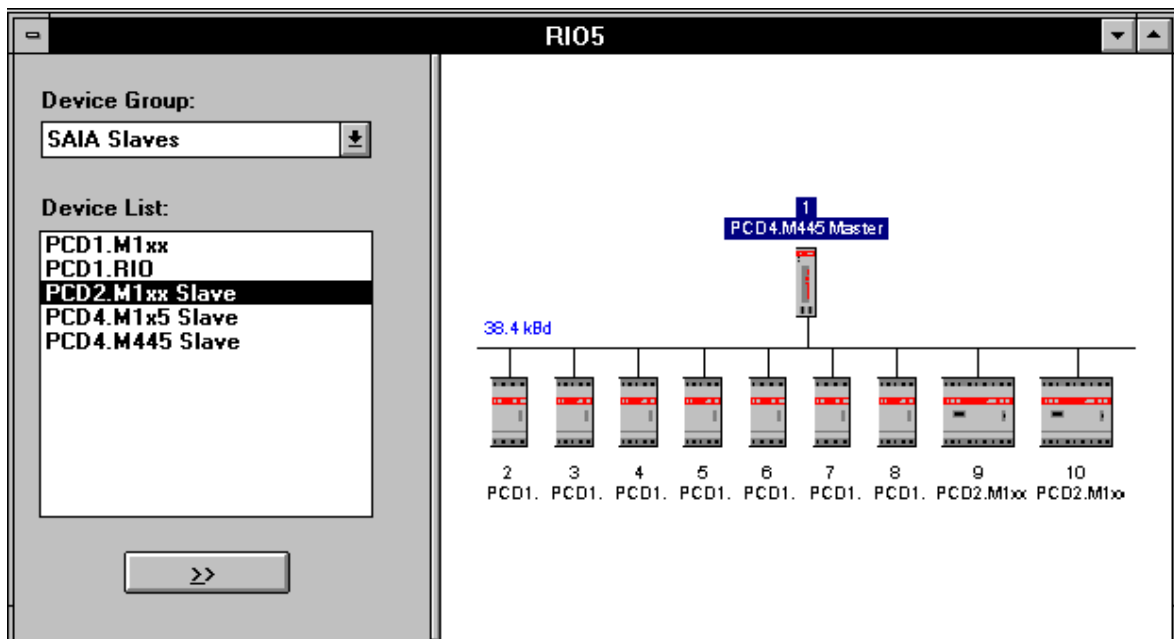
### 5.3.6 'View' submenu



**'Toolbar':** The toolbar can be switched on or off. The toolbar itself is described in section 5.3.10.

**'Status Bar':** The status bar can be switched on or off. (The status bar is located on the bottom screen margin of the main menu).

**'Zoom to Fit':** The RIO configuration in the configuration field can be adjusted to fit the window size.



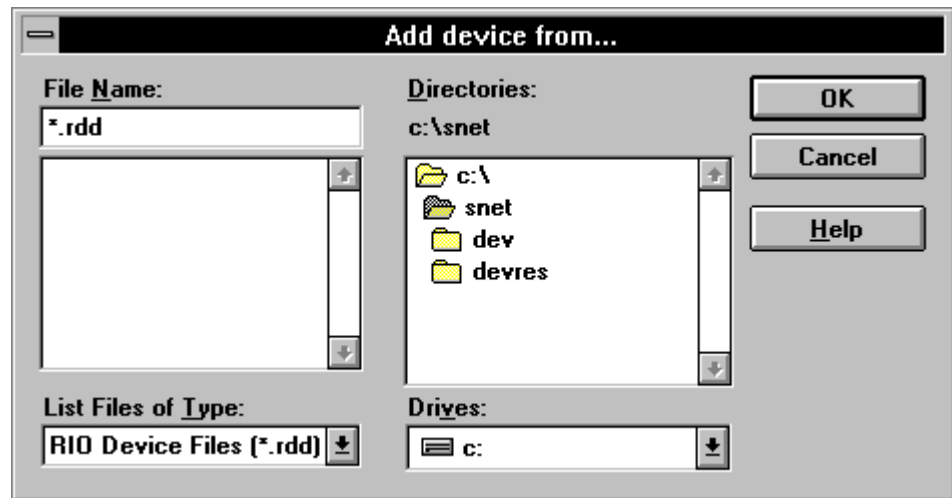
**'Zoom In':** The configuration can be enlarged and scrolled with the scroll bars. (Useful for the larger configurations).



**'Zoom Out':** The enlarged configuration is returned to the size it had during editing.



### 5.3.7 'Library' submenu



**'Add Device...':** Further RIO master or slave types can be added to the existing library.

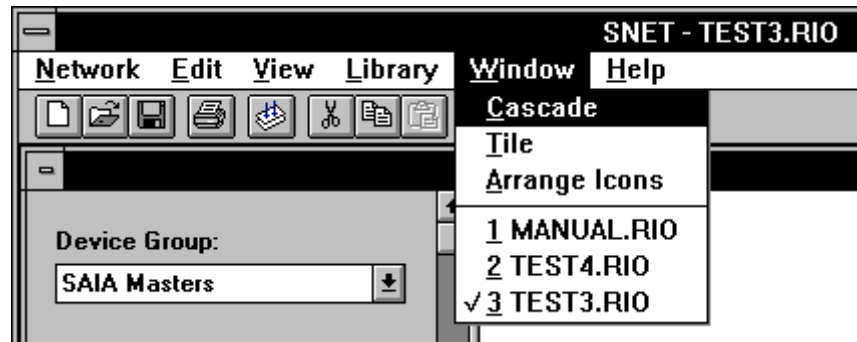
**'Remove Device...':** RIO master or slave types can be removed from the library. The deleted types are subsequently no longer available for any configuration.

Any attempt to remove a "Device" results first in a warning, which must be acknowledged.





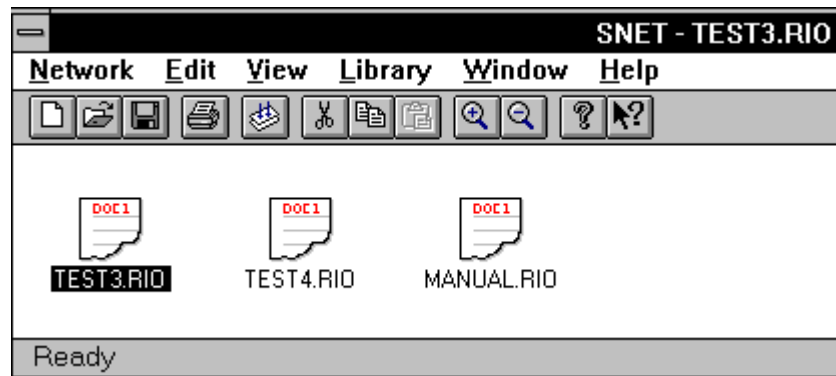
### 5.3.8 'Window' submenu



**'Cascade':** If several projects are open, the individual windows are arranged so that each header bar is visible.

**'Tile':** If several projects are open, the individual windows are arranged from top to bottom so that the start of each project remains visible.

**'Arrange Icons':** If some of the projects called have been minimized and repositioned on the screen, the projects' icons can be rearranged with this function.



### 5.3.9 'Help' submenu



**'Index':**

Interactive Help in English.



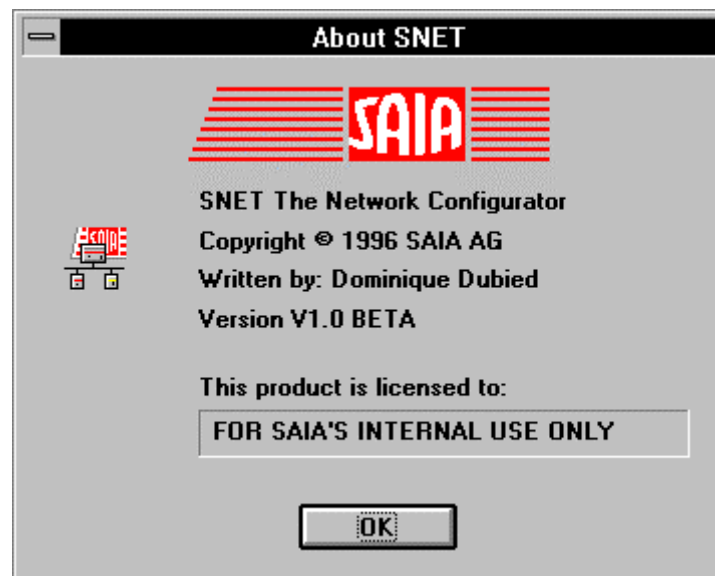
The "Help" contents are similar to the current chapter of the manual.

**'Using Help':**

Instructions for using "Help". This file presents itself in the language of the installed WINDOWS version.

**'About Snet...':**

The same screen appears as when the RIO configurator is called.















The screen shows the configurator version and the name of the licensee.

### 5.3.10 The toolbar



The toolbar contains the functions most frequently used for working with the RIO configurator. Clicking on a function symbol executes that function directly, or begins an operation.

Symbol	Keys	Function	
	Ctrl + N	'New'	New network
	Ctrl + O	'Open'	Open existing network
	Ctrl + S	'Save'	Save the network
	Ctrl + P	'Print'	Print
	Ctrl + G	'Generate'	Generate files RIO.DEF and RIOCFG.SRC
	Ctrl + X	'Cut'	Cut out
	Ctrl + C	'Copy'	Copy
	Ctrl + V	'Paste'	Insert
		'View Zoom In'	Enlarge
		'View Zoom Out'	Reduce
		'Help About'	Version, licensee name
	Shift + F1	'Help Index'	Help

Notes

## 6. Programming

---

### 6.1 Introductory example (PCD2 + 2 PCD1.RIO)

---

A simple example shows all stages of programming, from network planning to commissioning.

#### 6.1.1 Task

The task is to build a RIO network with 1 master and 2 non-intelligent slave stations, which is then programmed and commissioned.

#### Types and equipment of the 3 stations:

Master station:

Station number:	2
PCD type:	PCD2.M110 or PCD2.M120 (with additional RAM memory)
Interface:	PCD7.F110 or PCD7.F150, Port 1
Display module:	PCD2.F510 or PCD2.F530
I/O modules:	Addr. 00 - 07      E110
	Addr. 16 - 23      E110 (2 BCD digits)
	Addr. 32 - 39      A400
	Addr. 48 - 55      A400

Slave stations:

Station number:	5
PCD type:	PCD1-RIO
I/O modules:	Module 1      -
	Module 2      A400
	Module 3      -
	Module 4      -
Station number:	9
PCD type:	PCD1-RIO
I/O modules:	Module 1      E100
	Module 2      E100
	Module 3      -
	Module 4      W500

**Programming task:**

- 1) In place of a control program, output 55 in the master station is to blink.
- 2) Inputs 4 - 7 of the master station are to be transferred to outputs 0 - 3 of module 2, slave station no. 5.
- 3) Inputs 4 - 7 of module 1, slave station no. 9, are to be transferred to outputs 32 - 35 of the master station.
- 4) Inputs 0 - 3 of module 1, slave station no. 9, are to be transferred to outputs 4 - 7 of module 2, slave station no. 5.
- 5) The 2-digit BCD value at inputs 16 - 23 of the master station is to be transferred to analogue output 0 of the W500 module (module 4), slave station no. 9.
- 6) Analogue output 0 is carried by hardware to analogue input 0 of the same module. (Jumper between terminals 0 and 4 and terminals 1 and 9). The signal at analogue input 0 is to be displayed on the master station display module.

(Depending on the hardware, only the lower 8 bits of the 12-bit analogue module are used).

We will call this project TEST1.

It is assumed that the stations are correctly equipped, station numbers are correctly set, RS 485 bus and supply are wired and that the line termination, pull-up and pull-down resistors are adjusted as specified.

It is further assumed that the RIO configurator and PG3 utilities have been installed and are operational.

### 6.1.2 Station configuration

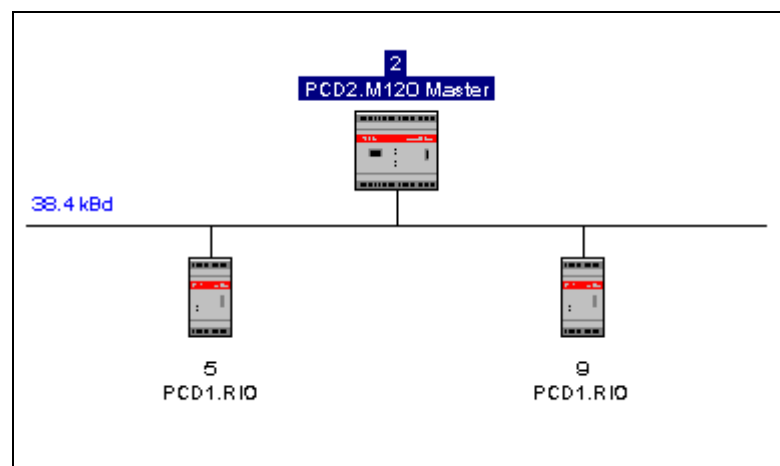
We call the RIO configurator and switch it to full-screen. Then 'Network' - 'New', Network type: "RIO-Network" - 'OK'.

We select our master station and move it into the configuration field with the double-arrow button. We switch to the slaves, select PCD1.RIO and activate the double-arrow button twice. From the 'View' menu, we click on 'Zoom to Fit'

To adjust the station numbers and define a timeout, we click on the separate stations. We start with slave station 3. We give it a new station number (address) 9 and select a timeout of 1000 ms, 'OK'. We give slave station 2 the number 5, leaving the timeout at 0, since this station has no outputs to reset after the timeout has passed.

We give the master station number (address) 2. We leave the timeout as standard value 0. We leave the port number at 1 and also leave the symbolic name (port\_rio) and the CPU charge (50%). 'OK'.

Our configuration now appears as follows:



The baud rate of 38.4 kBd (standard value) has been selected automatically. It is also the highest transmission speed for the stations used.

This completes configuration of the master station. The I/O modules on slave stations must now be configured.

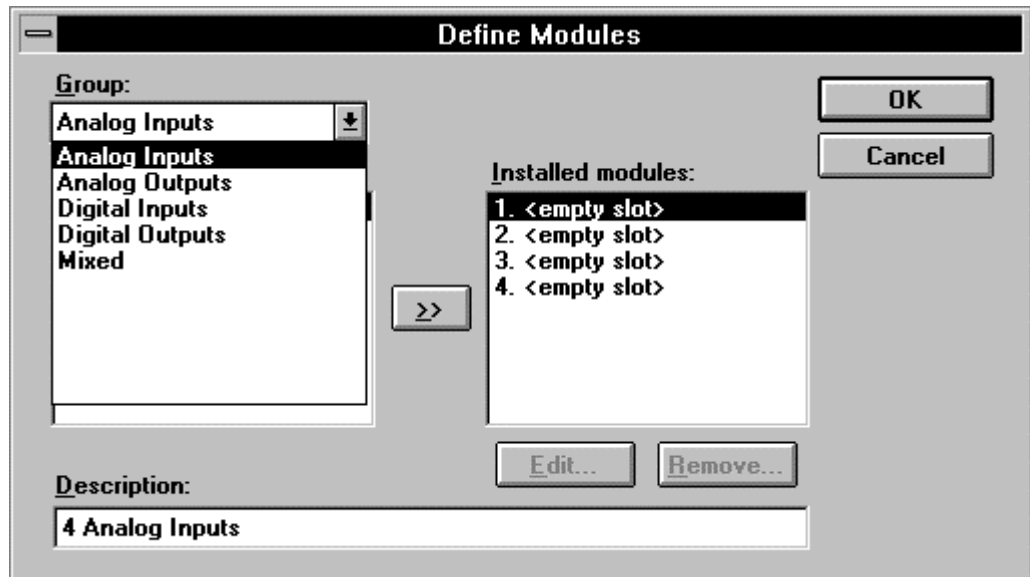
As a precaution, we save the work done up to now with 'Network' - 'Save As...', using the name "TEST1.RIO".

### 6.1.3 Configuring slave station I/O modules

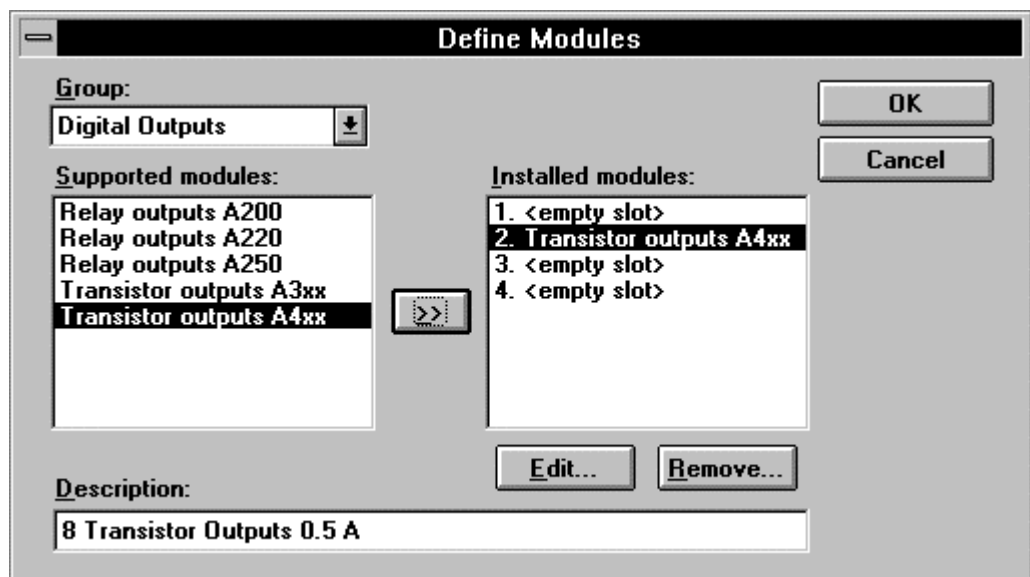
We begin with slave station no. 5, which has as its second module a PCD2.A400 digital output card. We define all 8 outputs with

```
st05_o08 to st05_o15
```

After double-clicking on station 5, the previously adjusted screen "Slave Parameters" - 'Station' appears. We click on 'Modules', then 'Define Modules...' and open the 'Group' window.

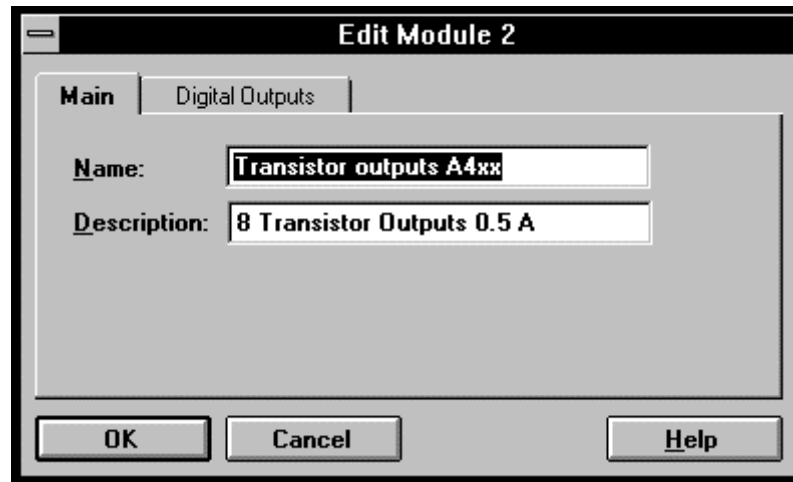


We select 'Digital Outputs', from which we choose 'Transistor outputs A4xx'. In the "Installed modules" window, we select the second module and activate the double-arrow button. Our chosen module is entered in the selected slot and can be edited.

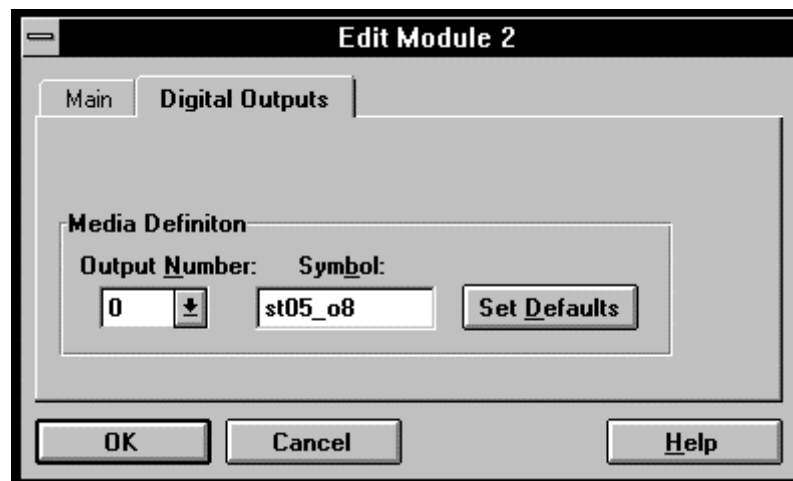


The 'Edit...' button should now be activated.

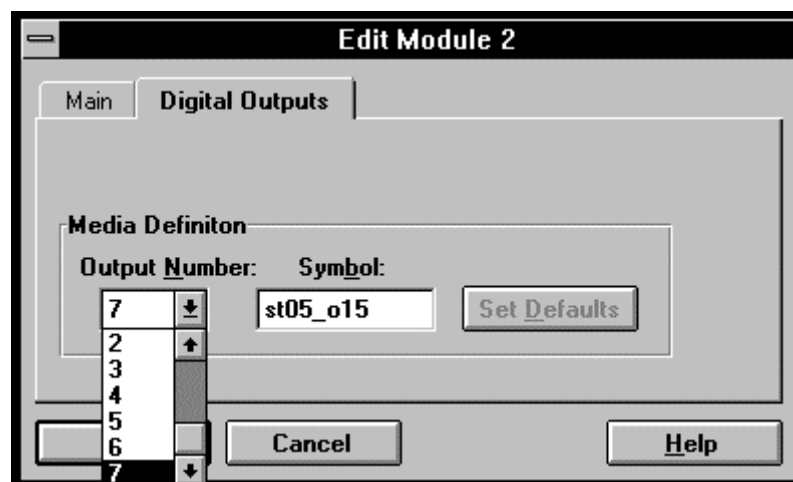




After clicking on 'Digital Outputs' it is possible to start editing the first symbolic name, <st05\_o08>, for output 0 of module 2.



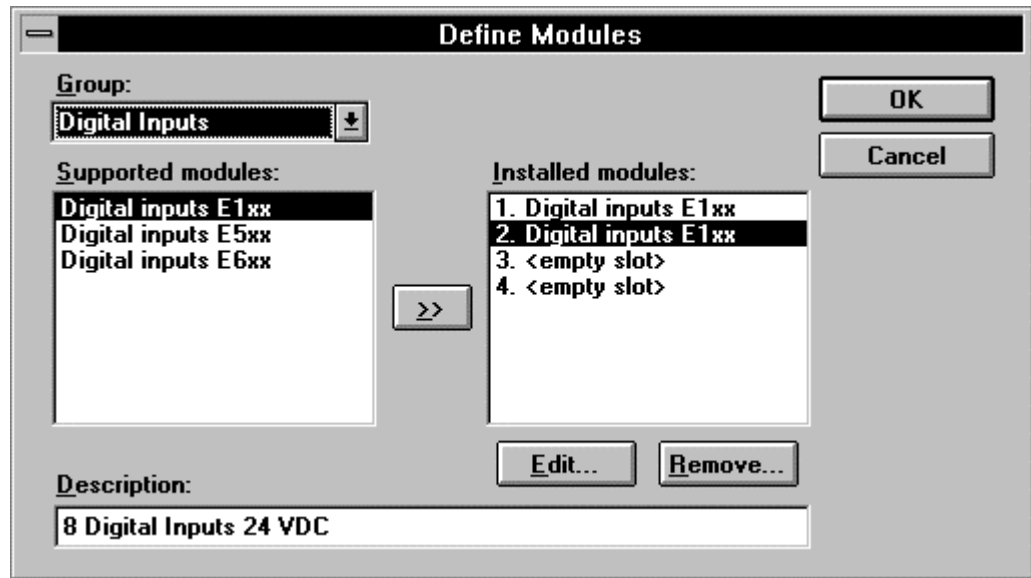
After the first symbol and **without** 'OK', subsequent elements are edited "by hand", otherwise the next 7 outputs are defined automatically with 'Set Defaults'.



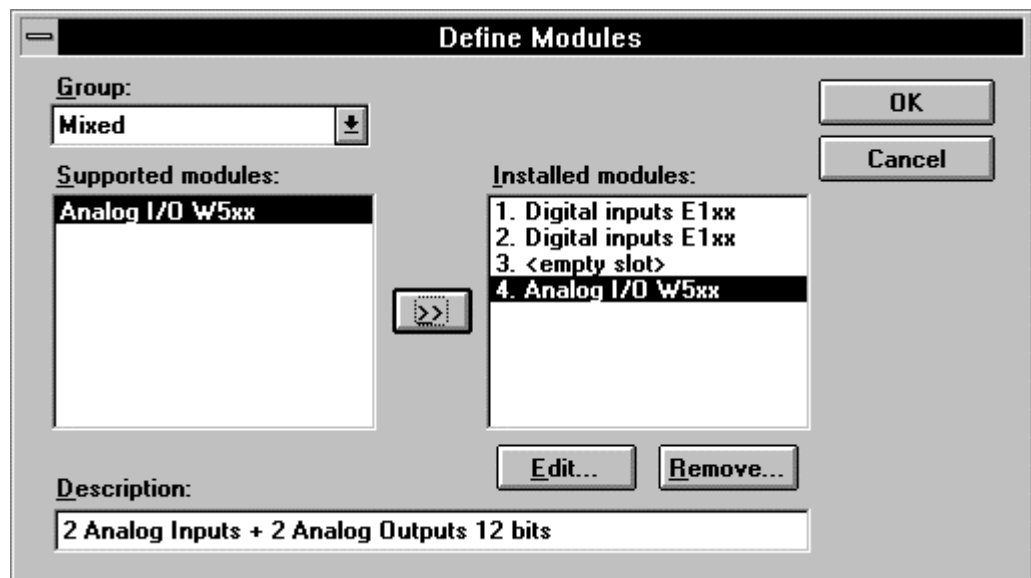
When all 8 outputs have been defined, confirm with 'OK'.

Slave station no. 5 only has this single output module. For this station, therefore, editing is complete. All windows are exited with 'OK', until the configuration window is reached again.

In the same way, we edit station no. 9, with 2 x E110 modules in slots 1 and 2 ...

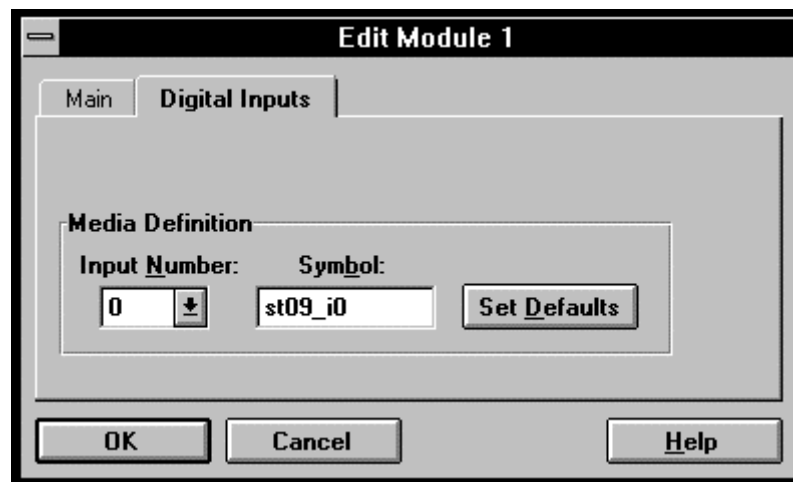


... and a combined analogue I/O module (mixed) in slot 4.

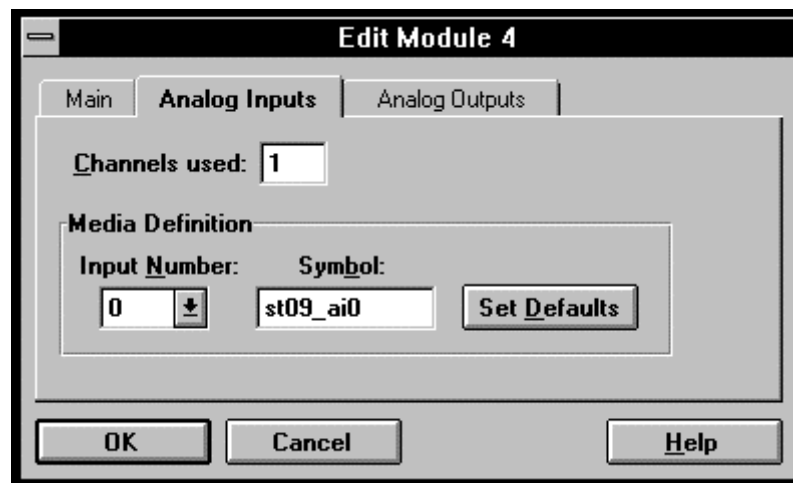


We edit input modules 1 and 2 (16 inputs) with

st09\_i0 to st09\_i15 ...



the analogue input with st09\_ai0 and  
the analogue output with st09\_ao0  
using only 1 channel each for the input and output. (Channels used: 1)

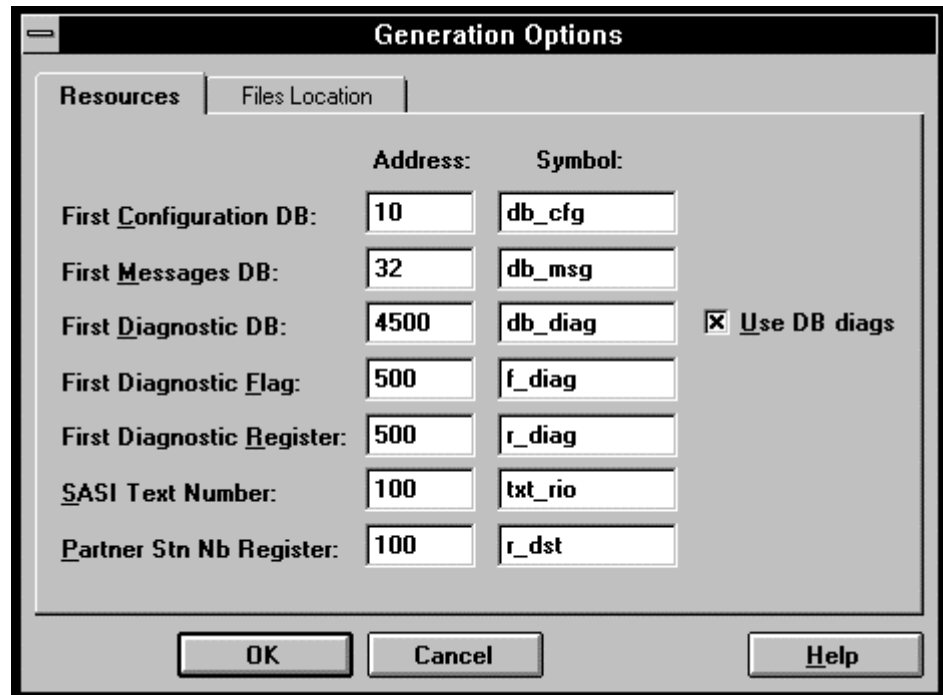


This concludes the editing of station no. 9.

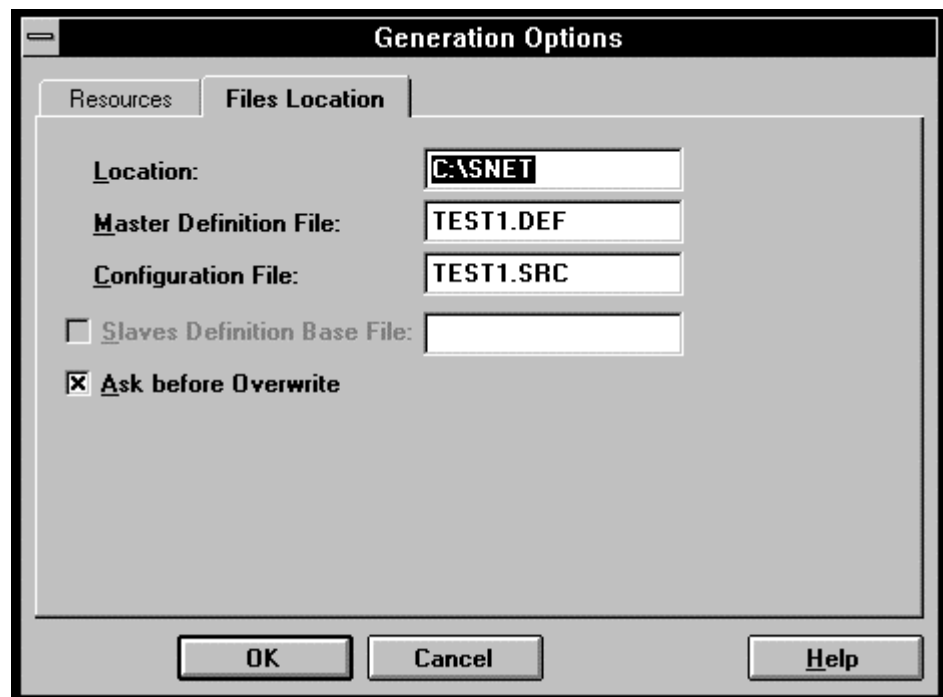
### 6.1.4 Generation of auxiliary files

When all stations have been configured and edited, the two auxiliary files "TEST1.DEF" and "TEST1.SRC" can be generated.

From 'Network' - 'Generate Options...' we leave the preset values indicated as they are, i.e.:



After clicking on 'Files Location', the following screen appears:



We leave this also and confirm with 'OK'.

We click on 'Generate...' from the 'Network' menu.

The following 3 files are generated:

- TEST1.RIO (configuration)
- TEST1.DEF (definition file with all EQUs)
- TEST1.SRC (configuration file)

The user program must now be written, which not only contains the process control program but also the communications relationships of the RIO network.

Before we start editing the user program, let us print out the files listed above to obtain an overview of the symbols used and to enable us to identify any collisions between media ranges.

**6.1.5 Definition file (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 Configuration file (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,";"

```

The significance of individual codes in the DBs is described in chapter 7, sections 7.6.1 and 7.6.2.

The present file, "TEST1.SRC" and file "TEST1.DEF" on the preceding page are reproduced in the form as generated by the configurator.

**6.1.7 "RIO" file (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 User program (TEST1\_X.SRC)

```

#include test1.def

      xob          16
      sasi         port_rio
                   txt_rio

      exob

; -----
      xob          30
      inc         c    100
      exob

; -----
      cob          0          ; Control program for process
                   0
      stl         t    0          ; Task item 1)
      ldl         t    0          ; Blinker
                   5
      com         o    55
      ecob

; -----

      cob          1          ; RIO comms relationships
                   0

      sth         i    4          ; Task item 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  ; Task item 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  ; Task item 4)
      out                st05_o12
      sth                st09_i1
      out                st05_o13
      sth                st09_i2
      out                st05_o14
      sth                st09_i3
      out                st05_o15

      digir        2          ; Task item 5)
                   i    16
                   st09_ao0

      dsp          st09_ai0  ; Task item 6)

      ecob

```

This is the user program which the programmer must edit for a simple RIO installation. The files with the specific RIO requirements have already been created by the configurator and have already been shown..

File "TEST1.DEF" is inserted in the user program with "\$include". The other file generated by the configurator, "TEST1.SRC", is linked to the user program. The user program must therefore have a different name from the configuration program. In our example, we have called the user program "TEST1\_X.SRC".

### **Some notes explaining the user program:**

**"\$include test1.def"** This should be clear. If "test1.def" is located in a different directory the path must be specified.

**"XOB 16"**. The cold-start routine contains the SASI instruction for the RIO task. The symbol for the port number was defined during configuration of the master station in the "Master Parameters" window. The symbol for the SASI text is defined in the "Generation Options" window. This window is reached from 'Network' - 'Generate Options...'.

**"XOB 30"**: This XOB is optional, i.e. the installation will also run without XOB 30. It is mainly called when online slave stations are hooked up or disconnected. This XOB is also called when there are syntax errors in the configuration and definition DBs. For our simple example, we merely count how often XOB 30 has been called. If XOB 30 is omitted, instead of calling it the error flag is set and the error LED comes on. More comprehensive applications of XOB 13 are shown in chapter 7: Diagnosis.

**"COB 0"**: In place of a control program, in this example output 55 of the master station is made to blink. This COB 0 has nothing to do with the RIO task and could also be omitted.

**"COB 1"**: In this part of the program, the communications relationships of the RIO task are programmed. The 5 instruction packages correspond to items 2 to 6 of the task specification and hardly require further explanation as far as the programming is concerned. The I/Os edited in the configurator, e.g. "st05\_o8" are assigned flags in the integrated TEST1.DEF file. The analogue values are stored in registers. These flags and registers are the media map of the RIO stations. It is refreshed in the firmware, unseen by the user. The configuration and SASI text are taken from the configuration file TEST1.SRC, which cannot be viewed by the user either. This file is linked to it..

### 6.1.9 Commissioning

The configuration of the PCD2 program memory must have code, text and extension memory, e.g. standard configuration with:

7 k lines of code, 4 kBytes text and 24 kBytes extension memory.

The additional memory must be fitted, as there are DBs to be addressed in the range 4000 - 5999.

The present program takes up 47 program lines, 576 text characters and 1032 Bytes external memory.

After the two files "TEST1\_X.SRC" and "TEST1.SRC" have been linked and downloaded into the PCD, it can be switched into RUN and the network is functional. All programmed communications relationships can be run through.

Although we will only be looking in detail at the diagnostic possibilities later, some interesting diagnostic points can already be emphasised in the debugger.

First, the counter which we use in XOB 30 to count the calls of this XOB is to be shown refreshed. The online separation of a station from the bus, or powering off its supply, can be viewed in counter 100.

Another interesting point is the SASI text which, by selecting 'Display' - 'Text' - <100>, is shown in plain text i.e. the absolute addresses, for our example:

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

These absolute addresses are important for viewing with the debugger. Let us look at DB 4500 for our test installation:

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

The first 15 elements of DB 4500 are displayed. It can be seen that elements 5 and 9 contain values. These values are the number of telegrams which, until now, have been exchanged between the master and slave stations 5 and 9. The two elements we are interested in can be displayed and viewed as refreshed values with

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

If the master station is in 'STOP' the exchange of telegrams continues.

In the definition file list (section 6.1.5) the relationship of the slave elements to the media map flags can be seen. In 'STOP' it is possible to switch on and off the flags for outputs with the debugger. Since refreshing also continues to run in STOP, the appropriate outputs will follow the forced flags.

Comment:

The best information is supplied by the .LST file, which can be generated during assembly. Although this file is very comprehensive, it offers the advantage of containing all definitions in their symbolic and absolute formats, as well as the user program with all its comments, in one and the same file. (Unfortunately, the SASI text is missing. This must be printed out separately). The example "TEST2\_XA.LST" is shown in the appendix.

## 6.2 Further example (PCD4.M445 + PCD2 + 2 PCD1.RIO)

---

This expanded example is intended to show the use of a master station with 2 CPUs plus the use of non-intelligent and intelligent slave stations. It will also briefly demonstrate the transmission of acyclical telegrams.

### 6.2.1 Task

The task is to build a RIO network with 1 master, 1 intelligent and 2 non-intelligent slave stations, which is then programmed and commissioned.

#### Types and equipment of the 4 stations:

Master station:

Station number:	1	
PCD type:	PCD4.M445	
Bus module:	PCD4.C340	
User program:	CPU 0	
RIO:	CPU 1	
RIO port:	Port 3 with PCD7.F150	
I/O modules:	Addr. 00 - 15	E100
	Addr. 16 - 31	E100 (4 BCD digits)
	Addr. 32 - 47	A400
	Addr. 48 - 63	A400

Slave stations:

Station number:	5 (non-intelligent slave station)	
PCD type:	PCD1-RIO	
I/O modules:	Module 1	-
	Module 2	A400
	Module 3	A400
	Module 4	-

Station number:	9 (non-intelligent slave station)	
PCD type:	PCD1-RIO	
I/O modules:	Module 1	E100
	Module 2	E100 (2 BCD digits)
	Module 3	-
	Module 4	-

Station number:	12	(intelligent slave station)
PCD type:	PCD2.M110 or PCD2.M120	
Port:	PCD7.F110 or PCD7.F150, Port 1	
Display module:	PCD2.F510 or PCD2.F530	
I/O modules:	Addr. 00 - 07	E110
	Addr. 16 - 23	E110 (2 BCD digits)
	Addr. 32 - 39	A400
	Addr. 48 - 55	A400
	Addr. 64 - 71	A400

### Programming task:

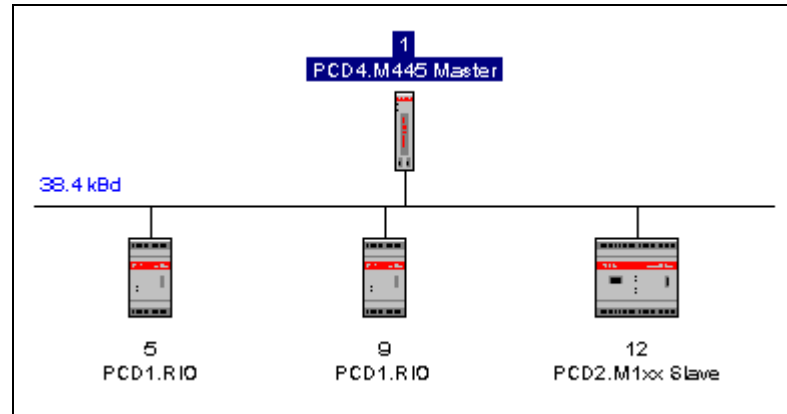
- 1) In place of a control program, output 63 in the master station is to blink.
- 2) Inputs 4 - 7 of the master station are to be transferred to outputs 0 - 3 of module 2, slave station no. 5.
- 3) Inputs 4 - 7 of module 1, slave station no. 9, are to be transferred to outputs 32 - 35 of the master station.
- 4) Inputs 0 - 3 of module 1, slave station no. 9, are to be transferred to outputs 4 - 7 of module 2, slave station no. 5.
- 5) Inputs 0 - 3 of the master station are to be transferred to outputs 32 - 35 of slave station no. 12.
- 6) Inputs 16 - 23 of slave station no. 12 are to be transferred to outputs 48 to 55 of the master station.
- 7) Inputs 0 - 7 of module 2, slave station no. 9 are to be transferred to outputs 64 - 71 of slave station no. 12.
- 8) Inputs 20 - 31 (3 BCD digits) of the master station are to be displayed on the display module of slave station no. 12
- 9) If input 15 of the master station passes from low to high, inputs 16 to 19 (1 BCD digit) should be transmitted acyclically to outputs 48 - 51 of slave station no. 12.
- 10) Output 55 is to blink as user program in intelligent slave station no. 12. (This has nothing to do with the RIO task: visual watchdog only).

We will call this project TEST2.

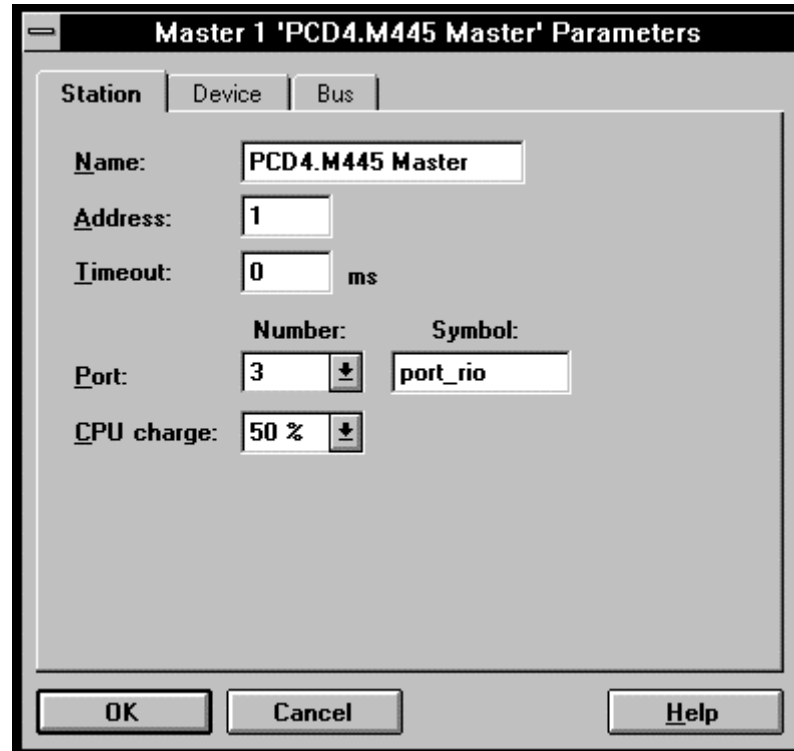


### 6.2.2 Station configuration

Station configuration proceeds in the usual way. Since the PCD2 does not support 72.8 kBd, the network can only be run at 38.4 kBd.



In the master station the whole RIO task runs on CPU 1. This CPU has access to ports 2 and 3. We select port 3. This port must be equipped on bus module PCD4.C340 with an RS 485 interface module. (PCD7.F110 or PCD7.F150). We leave the timeout at its standard value of 0.



### 6.2.3 Configuring slave stations

Configuration of the non-intelligent slave stations follows the same pattern as in the preceding example.

It is now necessary to edit the intelligent slave station. By double-clicking on station no. 12, we open the "Slave Parameters" window. After clicking on 'Messages' an empty field appears with the header "Defined Messages". When 'Add...' is clicked on, the following window appears:

**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 flags

**Media Definition**

**Number of Media to Transfer:** 8

**Position:** 0 **Symbol:**

**Media Map**

<b>First Master Media Type:</b>	Flag
<b>First Master Media Address:</b>	<input type="text" value="0"/>
<b>First Slave Media Type:</b>	Flag
<b>First Slave Media Address:</b>	<input type="text" value="0"/>

Items 5), 6), 7) and 8) of the programming task concern slave station no. 12. These 4 parts of the task are now to be formulated in 4 "Messages".

Let us start with item 5):

- 5) Inputs 0 - 3 of the master station are to be transferred to outputs 32 - 35 of slave station no. 12.

There is no available message "Master Inputs → Slave Outputs". It is only possible to reach slave station I/Os directly. We select

"Master Flags → Slave Outputs"

and enter this message in the field with 'OK'.

**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 **Set Defaults**

**Media Map**

<b>First Master Media Type:</b>	<b>Flag</b>
<b>First Master Media Address:</b>	40
<b>First Slave Media Type:</b>	<b>Output</b>
<b>First Slave Media Address:</b>	32

OK  
Cancel

At "Number of Media" we enter <4>, since we want to transfer to 4 consecutive outputs.

At "First Master Media Address" we must specify 4 consecutive master station flags. The 4 inputs will be mapped here. Strict care should be taken to ensure that the flag ranges do not intersect! We select flag 40 (to 43).

At "First Slave Media Address" we enter <32>. This is the lowest of the 4 output addresses.

At "Symbol" we define all outputs with a symbolic name. When everything has been completed, we confirm with 'OK'.

Item 6) follows:

- 6) Inputs 16 - 23 of slave station no. 12 are to be transferred to outputs 48 - 55 of the master station.

There is no available message "Master Inputs → Slave Outputs". It is only possible to reach slave station I/Os directly. We select

"Slave inputs → Master flags"

and enter this message in the field with 'OK'.

**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 slave inputs to the master flags

**Media Definition**

**Number of Media to Transfer:** 8

**Position:** 0

**Symbol:** st12\_i16

**Media Map**

**First Master Media Type:** Flag

**First Master Media Address:** 50

**First Slave Media Type:** Input

**First Slave Media Address:** 16

At "Number of Media" we enter <8>, since we want to transfer 8 consecutive inputs.

At "First Slave Media Address" we enter <16>. This is the lowest of the 8 input addresses.

At "First Master Media Address" we must specify 8 consecutive master station flags. The 8 inputs will be mapped here. Strict care must be taken to ensure that the ranges of flags do not intersect! We select flag 50 (to 57).

At "Symbol" we define all inputs with a symbolic name. When everything has been completed, we confirm with 'OK'.

Let us move on to item 7):

- 7) Inputs 0 - 7 of module 2, slave station no. 9, are to be transferred to outputs 64 -71 of slave station no. 12.

No message is provided for the direct transfer of elements from one slave station to another slave station. The inputs of slave station no. 9 are already included in the media map when this station is configured. It is now only necessary to produce the relationship between the outputs of the intelligent slave station and the media map. For this we select flags 60 - 67. At "First Master Media Address" we enter <60>. At "First Slave Media Address" we enter <64> for the lowest output address.

The window is completed as follows:

**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**

<b>First Master Media Type:</b>	<b>Flag</b>
<b>First Master Media Address:</b>	60
<b>First Slave Media Type:</b>	<b>Output</b>
<b>First Slave Media Address:</b>	64

OK  
Cancel

... and confirmed with 'OK'.

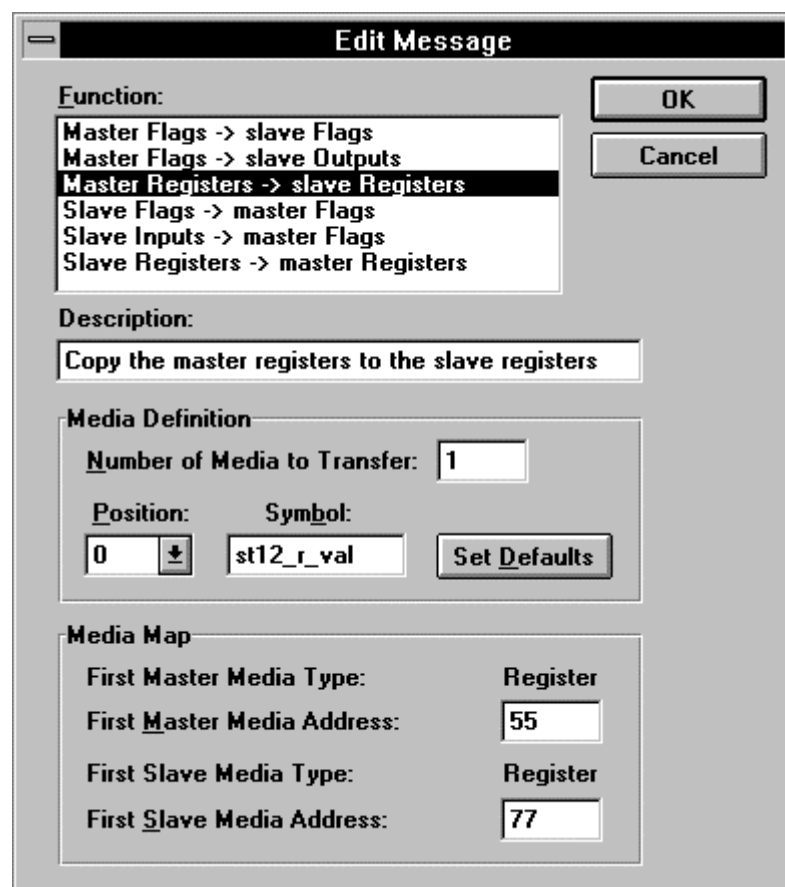
This still leaves item 8):

- 8) Inputs 20 - 31 (3 BCD digits) of the master station are to be displayed on the display module of slave station no. 12.

The 12 inputs could now be transferred from the master to the slave station and, in the user program of the (intelligent) slave station, downloaded to a register and displayed.

We select the other route and download the 3-digit value in the master station into a register, then transfer it via the RIO network to the slave station.

The configuration is as follows:



The entries are confirmed with 'OK'.

The "Slave Parameters" window now appears as follows:



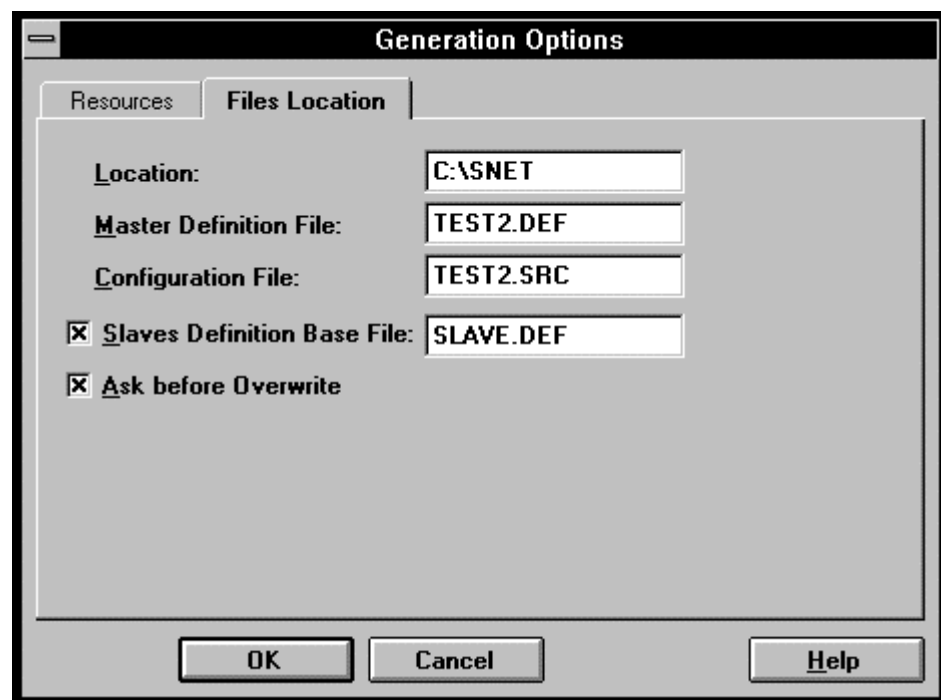
We confirm with 'OK'.

### 6.2.4 Generation of auxiliary files

This takes place exactly as described in section 6.1.4. Care should be taken to ensure that the diagnostic resources (flags and registers) do not collide with the elements of the media map, both for the non-intelligent and the intelligent stations.

Select 'Generate Options...'. The default settings in the 'Resources' window are left as they are.

In the next window, 'Files Location', we enter the directory and filenames for .DEF and .SRC. Since an intelligent station has been edited in the network, the entry "SLAVES.DEF" has already been made. Confirm with 'OK'.



With 'Network' - 'Generate...' we create the files "TEST1.DEF" and "TEST1.SRC" in the usual way.

In the present case, i.e. when one or more intelligent slaves are present in the network, a "SLAVE<Stat. Nr.>. DEF" file is created for each of these stations. For our example with one intelligent slave station no. 12, the file "SLAVE12.DEF" is generated. This file contains all definitions (EQU) for all elements in that station.

We save everything with 'Network' - 'Save'.

Before we begin editing the user program, let us print out the 3 files TEST2.DEF, SLAVE12.DEF and TEST2.SRC.



**6.2.5 Definition file (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 Definition file of intelligent 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 Configuration file (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 User program for master station (TEST2\_X.SRC for CPU 1, TEST\_Y.SRC for 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 comms relationships
                  0

      sth          i      4      ; Task item 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  ; Task item 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  ; Task item 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      ; Task item 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      ; Task item 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       ; Task item 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            ; Task item 8)
           i          20
           st12_r_val

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

ecob
; -----

pb          0            ; (Task item 9))

ldl         r          r_dst
           12          ; Station number

stxm        3            ; Port
           4            ; Number of media
           i          16      ; Source
           o          48      ; Destination

epb

```

```

;      Program for CPU 0      (TEST2_Y)
;      *****

      cob          0          ; User program
                        0          ; for process

      stl      t      0          ; Task item 1)
      ldl      t      0
                        5
      com      o      63

      ecob

```

### Some notes explaining the user programs

#### Program for CPU 1:

The whole RIO task is handled in CPU 1, i.e. assignment of the RIO task and definition of the communications relationships between the master and the non-intelligent or intelligent slave stations. Any acyclical telegrams are also dealt with in this user program.

Individual programs must be written for each of the intelligent slave stations. To meet RIO requirements, these programs must at least contain the assignment of the RIO interface. The station number must also be configured for all these stations.

The files with the specific RIO requirements have already been generated by the configurator and have already been shown. The file "TEST2.DEF" is inserted in the user program with "\$include". The other file generated by the configurator, "TEST2.SRC", is linked to the user program. The user program must therefore have a different name from the configuration program. In our example, we call the user program for CPU 1 "TEST2\_X.SRC".

**"\$include test2.def"** This should be clear. If "test2.def" is located in a different directory the path must be specified.

**"XOB 16"**. The cold-start routine contains the SASI instruction for the RIO task. The symbol for the port number was defined during configuration of the master station in the "Master Parameters" window. The symbol for the SASI text is defined in the "Generation Options" window. This window is reached from 'Network' - 'Generate Options...'.

**"XOB 30":** This XOB is optional, i.e. the installation will also run without XOB 30. It is mainly called when online slave stations are hooked up or disconnected. This XOB is also called when there are syntax errors in the configuration and definition DBs. For our simple example, we merely count how often XOB 30 has been called. If XOB 30 is omitted, instead of calling it the error flag is set and the error LED comes on. More comprehensive applications of XOB 13 are shown in chapter 7: Diagnosis.

**"COB 1":** In this part of the program, the communications relationships of the RIO task are programmed. The 8 instruction packages correspond to items 2 to 9 of the task specification and hardly require further explanation as far as the programming is concerned. The I/Os edited in the configurator, e.g. "st05\_o10" are assigned flags in the integrated TEST2.DEF file. The analogue values are stored in registers. These flags and registers are the media map of the RIO stations. It is refreshed in the firmware, unseen by the user. The configuration and SASI text are taken from the configuration file TEST2.SRC, which cannot be viewed by the user either. This file is linked to it..

#### **Program for CPU 0:**

Actual user program for controlling the process. Access to ports 0 and 1, but not 2 or 3. In this example port 3 is in any case occupied by the RIO task.

The user program in CPU 0 is not in any way affected by the RIO task, neither by the cyclical refreshing of the media map, nor by the cyclical or acyclical execution of communications relationships.



### 6.2.9 User program for intelligent slave station (TEST2\_12.SRC)

```

;      Programm for slave station 12 (PCD2)      (TEST2_12)
;      *****
xob          16

sasi          1
              1

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

exob
; -----
cob          0
              0

stl      t      0      ; Task item 10)
ldl      t      0
              5
com      o      55

dsp      r      77      ; Partial task item 8)

ecob

```

An individual program must be written for each intelligent slave station. To meet RIO requirements, these programs must at least contain the assignment of the RIO port. Assignment of the RIO port must take place in "SS1" mode (slave S-Bus), e.g.:

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

The baud rate must match the RIO network.

For all these stations, the station number must also be configured.

This is done in the PG3 with "SAIA PCD CONFIGURATION" - "S-Bus communications":

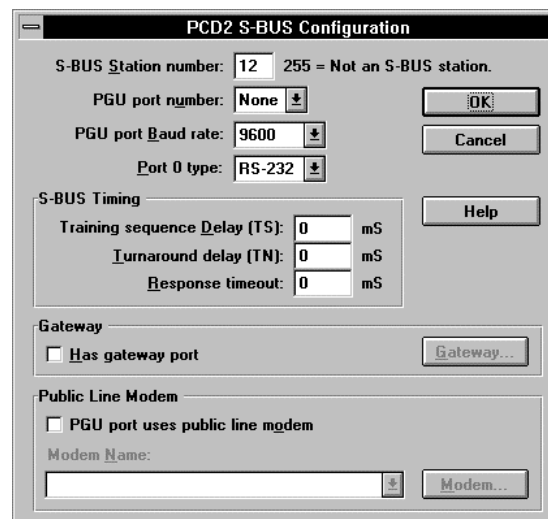
```

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
    
```

The configuration must then be downloaded in the menu "Up/download" - "CONFIGURE S-BUS".

Configuration can also be carried out using the PG4 online configurator:



### 6.2.10 Commissioning

The configuration of the CPU 1 program memory in the PCD4 must have code, text and extension memory. CPU 0 should be configured according to the demands of the user program.

After the two files "TEST2\_X.SRC" and "TEST2.SRC" have been linked and downloaded into CPU 1 of the PCD4, and when the user program has been downloaded into CPU 0, both CPUs can be switched into RUN. If the program is also downloaded to intelligent slave station no. 12 and if this station is also in RUN, the user programs and the RIO network are operational. All programmed communications relationships can be run through.

Although we will only be looking in detail at the diagnostic possibilities later, some interesting diagnostic points can already be emphasised in the debugger.

First, the counter which we use in XOB 30 to count the calls of this XOB is to be shown refreshed. The online separation of a station from the bus, or powering off its supply, can be viewed in counter 100.

Another interesting point is the SASI text which, by selecting 'Display' - 'Text' - <100>, is shown in plain text i.e. the absolute addresses, for our example:

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

These absolute addresses are important for viewing with the debugger. Let us look at DB 4500 for our test installation:

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

The first 21 elements of DB 4500 are displayed. It can be seen that elements 5, 9 and 12 contain values. These values are the number of telegrams which, until now, have been exchanged between the master and slave stations 5, 9 and 12. The three elements we are interested in can be displayed and viewed as refreshed values with

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

If the master station is in 'STOP', the exchange of telegrams continues.

In the definition file list (section 6.2.6) the relationship of the slave elements to the media map flags can be seen. In 'STOP' it is possible to switch on and off the flags for outputs with the debugger. Since refreshing also continues during STOP, the appropriate outputs will follow the forced flags.

Via the debugger, it is also possible, for example, to write a decimal value of max. 6 digits into register 55. This value appears directly after confirmation with <CR> on the display module of slave no. 12 (PCD2), as configured registers in the media map are also refreshed cyclically. (The PCD2 must be in RUN for this, as the value in the user program is sent to the display). If just the receive register in the PCD 2 is considered (for our example, R 77), it is refreshed even when the PCD 2 CPU is in STOP. For intelligent slave stations, the SASI instruction only has to be processed once (also possible in TRACE) to keep the RIO channel open and maintained.

#### Comment:

The best information is supplied by the .LST file, which can be generated during assembly. Although this file is very comprehensive, it offers the advantage of containing all definitions in their symbolic and absolute formats, as well as the user program with all its comments, in one and the same file. (Unfortunately, the SASI text is missing. This must be printed out separately). The example "TEST2\_XA.LST" is shown in the appendix.

## 6.3 Causes of common errors

---

Apart from difficulties with the RS 485 bus line (which must be very carefully laid out and connected up according to the information in section 3.5 and the manuals on "SAIA<sup>®</sup> S-Bus" and "Installation components for RS 485 networks") many small errors and lapses can adversely affect or prevent the RIO network from working.

The following list contains errors which occurred during preparation of the two examples in this manual. The list is probably not complete and can be expanded.

<b>Station</b>	<b>Possible error</b>
Master	<ul style="list-style-type: none"> <li>• Wrong port assigned</li> <li>• RIO bus cable not connected or connected to wrong port</li> <li>• Wrong firmware</li> <li>• Program downloaded to wrong CPU (PCD4.M445)</li> <li>• CPU not in RUN (PCD switched on with programming unit connected)</li> <li>• Wrong type of interface (not RS 485)</li> </ul>
Slave (non-intelligent)	<ul style="list-style-type: none"> <li>• Bus cable not plugged in or cut</li> <li>• Supply absent from individual stations, or cut (looping pins ++ and -/-!)</li> <li>• Wrong station number set</li> <li>• Wrong baud rate set</li> </ul>
Slave (intelligent)	<ul style="list-style-type: none"> <li>• Station number not configured</li> <li>• Wrong station number</li> <li>• SASI instruction not processed</li> <li>• Wrong mode assigned</li> <li>• Wrong port assigned</li> <li>• Wrong port connected</li> <li>• Wrong interface module (not RS 485)</li> <li>• Wrong firmware</li> </ul>



## 7. Diagnosis

---

### 7.1 Monitoring presence of master station

---

In the configurator a timeout of, for example, 500 ms can be entered during configuration of the master station (see section 5.3.3). This timeout is continuously monitored by the slaves. If the timeout expires, i.e. if the master is absent for longer than the timeout, a flag is reset (F\_diag+19) and can be evaluated in the user program (see section 7.3).

Another function of this timeout, and probably a more important one in practice, is that when it expires all outputs of non-intelligent slave stations are reset and the registers with analogue module values are zeroed. However, this only occurs if a timeout has been defined for the slave stations (see next section)

When the master returns, the media map is transmitted again.

If monitoring of the master is not wanted, 0 is entered during configuration. In this case, S-Bus standard values are used:

for 9600 Bd	→	250 ms
for 19,200 Bd	→	200 ms
for 38,400 Bd	→	100 ms
for 76,800 Bd	→	50 ms

The timeout values are 0 or 50 - 3,600,000 ms (1 hour).

If an intelligent slave station is to determine the presence of the master, this is programmed via the user program of that station, e.g. with the evaluation of the Receiver Busy flag (RBSY). It would also be feasible to complement a flag in the slave station continuously from the master station and evaluate it in the user program (watchdog). A reset of some or all outputs must also take place in the user program.

## **7.2 Monitoring presence of slave stations (XOB 30)**

---

In the configurator, a timeout of, for example, 500 ms can be entered during configuration of the slave stations (see section 5.3.4). If no RIO communication occurs between the master and the slave during a period which exceeds the defined timeout, all that station's outputs are switched off and the registers with the analogue module values are zeroed.

This behaviour can be followed in the two examples TEST1 and TEST2, by having the outputs of slave station no. 5 switched on and then pulling out the bus connector. After expiry of the timeout, the outputs will switch off and, after reconnection, return to their previous state.

In addition, after every message sent from the master to a slave, the connection is tested. If the test is not answered positively by the slave, the master CPU calls XOB 30. This is essentially the case when, online, a station is removed from the network or closed down. This can be viewed in the two examples, TEST1 and TEST2.

Even when the master station is in STOP, RIO communication goes on and the media map continues to be refreshed. If during this time stations are added to or removed from the network, after returning to RUN or COND.RUN, XOB 30 is called and processed for each addition or removal.

If XOB 30 is not programmed, the error flag is set and the error LED is switched on instead of calling XOB 30.

In our two examples, TEST1 and TEST2, XOB 30 merely contains a counter which is incremented. The number of additions or removals can therefore be read.

More detailed diagnosis with XOB 30 will be shown later (section 7.7), as the diagnostic resources should be discussed first.



## 7.3 Diagnostic flags

---

In the master station 24 flags are provided for rough communications diagnostics. The address of the first of these 24 flags is included in the SASI text under the heading "DIAG".

The **first 8 flags** (flags 0..7) contain the usual rough communications diagnostics for detecting communications features with the user program.

Address	Name	Description
xxxx	<b>RBSY</b>	Receiver busy
xxxx + 1	<b>RFUL</b>	Receive buffer full
xxxx + 2	<b>RDIA</b>	Receiver diagnostic
xxxx + 3	<b>TBSY</b>	Transmitter busy
xxxx + 4		Not used
xxxx + 5	<b>TDIA</b>	Transmitter diagnostic
xxxx + 6	<b>XBSY</b>	SASI permission
xxxx + 7	<b>NEXE</b>	Not executed

Die **next 8 flags** (flags 8...15) contain the usual communications diagnostics for detecting communications features of the RIO task.

Address	Name	Description
xxxx + 8	<b>RBSY</b>	Receiver busy
xxxx + 9	<b>RFUL</b>	Receive buffer full
xxxx + 10	<b>RDIA</b>	Receiver diagnostic
xxxx + 11	<b>TBSY</b>	Transmitter busy
xxxx + 12		Not used
xxxx + 13	<b>TDIA</b>	Transmitter diagnostic
xxxx + 14	<b>XBSY</b>	SASI permission
xxxx + 15	<b>NEXE</b>	Not executed

In general, the user has nothing to do with the standard flags. If an error occurs, however, the specialist can obtain interesting pointers towards error recognition from these diagnostic flags. More detailed explanations are available in the "SAIA<sup>®</sup> S-Bus" manual.

**Flag 16** (fcfg) indicates that the master is sending configuration messages to the slaves.

**Flag 17** (frdy) indicates that media map refreshing is underway.

**Flag 19** (fall) indicates that all messages in a cycle have been transmitted to the slaves. The flag is set high at the end of each successfully concluded cycle. For monitoring purposes, this flag can be polled in the user program after each cycle and reset. The flag is reset by the RIO task, if a transmission error has been detected.

**Flag 21** (festn) shows that a station has been added to or separated from the network online.

**Flag 22** (fecfg) shows that the contents of a DB, which is used to configure a station, are garbled. The DB number can be obtained from the 4th diagnostic register.

**Flag 23** (femsg) shows that the contents of a DB, which defines a cyclical telegram, are garbled. The DB number can be obtained from the 4th diagnostic register.

## Summary of the diagnostic flags

Position	Function	Description
F_diag+0.. F_diag+7	Usual diagnosis if communication is via user program (STXM/SRXM)	The meaning of this flag corresponds to S-Bus communication
F_diag+8.. F_diag+15	Communications diagnostic of the RIO task (cyclical telegrams)	The meaning of this flag corresponds to S-Bus communication
F_diag+16: <b>fcfg</b>	Configuration underway	The master is busy transmitting configuration instructions.
F_diag+17: <b>frdy</b>	Refreshing underway	Media map refreshing is being executed.
F_diag+19: <b>fall</b>	All messages in a cycle correctly transmitted	Indicates that all cyclical telegrams have been received by all slave stations. The flag can be reset by the user. It is then reset high in the next cycle. The flag is automatically set low if a telegram is acknowledged negatively.
F_diag+21: <b>festn</b>	Station error	Indicates that a station has been added to or removed from the network.
F_diag+22: <b>fecfg</b>	Configuration error	Indicates that the syntax of a configuration DB is incorrect. The RIO task is aborted when this kind of error occurs. This flag must be reset by the user.
F_diag+23: <b>femsg</b>	Transmission error	Indicates that the syntax of a transmission DB is incorrect. This flag must be reset by the user.

The basic resource "F\_diag" is defined automatically in the configurator. The absolute base addresses of the diagnostic flags can most easily be discovered in the debugger, with 'Display' - 'teXt' - <Text number of SASI text>.

## 7.4 Diagnostic registers

Apart from the 24 flags discussed, there are also 4 registers and 2 data blocks provided in the master station for fine diagnosis. The **1st register** contains standard information for locating errors using the communications instructions STXM and SRXM (see SASI instruction in the relevant manuals). The address of the first of these 4 registers is contained in the SASI text, under the heading "DIAG".

	Bit	Definition	Description
<b>R E C E I V E R</b>	0	Overflow error	Overflow of internal receive buffer
	1		
	2	Framing error	Character with format error (stop bit)
	3	Break error	Data line broken *
	4	CRC-16 error	CRC-16 error
	5	S-Bus PGU status	S-Bus PGU with PLM
	6	SASI OFF permission	SASI OFF permission
	7		
	8	Length error	Invalid telegram length
	9		
	10	Telegram error	Invalid telegram
	11	Status error	PCD with wrong status, impossible to execute instruction
	12	Range error	Invalid resource address
	13	Value error	Invalid data value
	14	Missing media error	Media address not defined or invalid
15	Program error	Program error: station number not assigned	
<b>T R A N S M I T T E R</b>	16	Retry count	Number of telegram repetitions in binary
	17		
	18		
	19		
	20	Negative response	Negative response (NAK)
	21	Missing response	No response after timeout
	22	Invalid response	Invalid response received
	23	CTS timeout	CTS not set after expiry TS delay
	24		
	25		
	26		
	27		
	28	Range error	Invalid resource address
	29		
	30		
	31	Program error	Program error: STXM/SRXM instruction not permitted

The **2nd register** also contains standard information for error location, but in this case for the RIO task (cyclical telegrams). The same table applies.

The **3rd register** contains the number of the last station to cause XOB 30 to be called. This register is initialized with the value 0 after the system is powered up or after a cold start. If the station is in the network, the number is positive; if the station is not in the network, the number appears negative.

This can be tried out with the examples TEST1 and TEST2.

The **4th register** contains the DB numbers in which messages are located when the error occurs. If the station comes onto the network, the DB number is indicated with 0. A DB number is only indicated, therefore, if a station is removed from the network. The base addresses of these message DBs are specified during configuration (Message DB).

Summary of the diagnostic registers:

Position	Function	Description
R_diag: <b>rcom1</b>	Usual diagnosis if communication is via user program (STXM/SRXM)	Usual diagnostic bits as in S-Bus mode
R_diag+1: <b>rcom2</b>	Communications diagnostic of the RIO task (cyclical telegrams)	Usual diagnostic bits as in S-Bus mode
R_diag+2: <b>rstn</b>	Addresse of last station	Number of the last station to cause XOB 30 to be called. 0:Initialization value, no error >0:Station back in network <0:Station outside network
R_diag+3: <b>rdb</b>	Number of last DB	Number of DB which is active during an error (0, if station is back in network).

The basic resource "R\_diag" is defined automatically in the configurator. The absolute base addresses of the diagnostic registers can most easily be discovered in the debugger, with 'Display' - 'teXt' - <Text number of SASI text>.

## 7.5 Diagnostic DBs

---

The 24 diagnostic flags and 4 diagnostic registers are joined by another 2 diagnostic data blocks (DB). Both DBs each consist of 127 elements (values) and are used as diagnostic counters. The number of the first of these two DBs is contained in the SASI text, under the heading "DIAG".

The first DB contains the total number of telegrams transmitted to each station. Each station is assigned 1 element in the DB. The element 0 is not used. Element 1 contains the counter for station no. 1, element 2 the counter for station no. 2, etc.

The use of these DBs is shown in the commissioning sections 6.1.9 and 6.2.10 of the examples TEST1 and TEST2.

The second DB contains for each station the number of transmission errors and the status of the station. The status of the station is encoded in bits 0 to 7 and the number of transmission errors in bits 8 to 31. The number of transmission errors is calculated by subtracting 255 from the value received. (The first error switches bit 8 = H (1), signifying the decimal value 256. If 255 is deducted from this, we obtain the value 1 for 1 error). Element 0 is not used. Element 1 contains the counter for station no. 1, element 2 the counter for station no. 2, etc.

The number (address) of the first DB must be selected at 4000 or higher.

Meaning of one element of the second diagnostic DB.

31	8	7..4	3..0
Error counter		Resv.	Status

If bit no. 0 = H (1),      the station in the network is active.  
 If bit no. 1 = H (1),      the station is to be configured.  
 If bit no. 2 = H (1),      the station has been removed from the network.  
 If bit no. 3 = H (1),      the station remains outside the network until the end of the cycle.

Examples :

00000000      The station is not in the network  
 00000001      Active slave in network  
 0000x110      Non-intelligent slave is no longer in network  
 0000x100      Intelligent slave is no longer in network

These counters are used to determine the transmission quality of the network. The DBs are viewed normally in the debugger and deleted. With the TFR instruction (Transfer Data) the results can, if necessary, also be integrated into the application.

This DB can also be tried out in connection with the two examples, TEST1 and TEST2.

#### Summary of the diagnostic DBs

Position	Function	Description
DB_diag: <b>dbcnt</b>	Number of telegrams transmitted	Each element of the DB belongs to a station in the network
DB_diag+1: <b>dberr</b>	Number of transmission errors and status of each station	Each element of the DB belongs to a station in the network

The basic element "DB\_diag" is defined automatically in the configurator. The absolute base addresses of the diagnostic flags can most easily be discovered in the debugger, with 'Display' - 'teXt' - <Text number of SASI text>.

## 7.6 Configuration

---

As we have seen in chapters 5 "RIO configurator" and 6 "Programming", configuration of the non-intelligent slave stations and message editing for the intelligent slave stations take place in the configurator. The definition and configuration files, as reproduced in chapter 6, are then generated at the touch of a button. However, a series of data blocks (DB) are also generated, which the configuration contains to benefit the firmware.

This section is intended to throw some light on the content of these DBs, although they are not important for the user.

### 7.6.1 Configuration of PCD1.RIO stations

Each station has its own configuration DB with the following elements:

1. Station address in network
2. Timeout for monitoring the presence of the master
3. Code for identification of module 1
4. Code for identification of module 2
5. Code for identification of module 3
6. Code for identification of module 4
7. Reserve
8. Synchronization

Let us view this as a whole, taking the example of TEST2.

At 'Generation Options...' in the 'Network' menu, we have accepted against "First Configuration DB" the address <10> [db\_cfg] and for First Message Address <32> [db\_msg].

In the debugger we select <Display> <data-Block> <10> To <15> <CR> and obtain the following:

```

>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 and DB 11 are used. Each DB consists of 8 elements (values) according to the following list:

## DB 10:

1. Element:	5	Station number
2. Element	1000	Timeout in ms
3. Element	0	Module no. 1 empty
4. Element	8192	Module no. 2 digital output module
5. Element	8192	Module no. 3 digital output module
6. Element	0	Module no. 4 empty
7. Element	0	Reserve
8. Element	1	Synchronization

## DB 11:

1. Element:	9	Station number
2. Element	1000	Timeout in ms
3. Element	4096	Module no. 1 digital input module
4. Element	4096	Module no. 2 digital input module
5. Element	0	Module no. 3 empty
6. Element	0	Module no. 4 empty
7. Element	0	Reserve
8. Element	1	Synchronization

## DB 12:

1. Element:	-1	:	
2. Element	-1	:	
3. Element	-1	:	End of
4. Element	-1	:	configuration
5. Element	-1	:	
6. Element	-1	:	(no other DB)
7. Element	-1	:	
8. Element	-1	:	

The codes for the individual modules are given in the following table (next page).

The codes for the modules are as follows:

Code	Hex.	Dec.	Description
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 Configuration of cyclical messages

The list with the messages for refreshing the media map is located in the configuration file, for our example in "TEST2.SRC" (see section 6.2.7). The messages have the following format:

1. Address of station in network
2. Function to be executed
3. Number of media to be transmitted
4. First source medium
5. First target medium
6. Reserve (always = 0)
7. Reserve (always = 0)

This information is also located in a series of data blocks (DB).

Let us view this as a whole, taking the example of TEST2.

At 'Generation Options...' from the 'Network' menu, we have accepted for "First Configuration DB" the address <10> [db\_cfg] and for First Message Address <32> [db\_msg].

In the debugger, we select <Display> <data-Block> <32> To <40> <CR> and obtain the following:

```

>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

```

DBs 32 to 37 are used. Each DB consists of 7 elements (values) according to the following list:

DB 32:

1. Element:	5	Station number
2. Element	8	Code for function to be executed (see table)
3. Element	16	Number of media to be transmitted
4. Element.	2064	First source medium
5. Element	64	First target medium
6. Element	0	Reserve
7. Element	0	Reserve

Codes for the function to be executed:

Code 1:	Copy the flags of a slave to the flags of the master
Code 2	Copy the inputs of a slave to the flags of the master
Code 3:	Copy registers of a slave to registers of the master
Code 8:	Copy flags of the master to flags of the slave
Code 9:	Copy flags of the master to outputs of the slave
Code 10:	Copy registers of the master to registers of the slave

Station no. 5 therefore has only 1 job:

- Copy flags of the master to flags of the slave
- 16 elements, comprising the 2 x PCD4.A400 digital output modules
- The source elements are flags 2064 - 2079 of the master
- The target elements are flags 64 - 79 of the slave \*)

\*) In the slaves, the digital input and output information is put on flags and from there, unseen by the user, put on the inputs and outputs.

DB 33 concerns station no. 9. This station has only 1 job: the 16 inputs of the PCD4.E110 input modules are transferred to flags 0 - 15 of the master station.

DBs 34 - 37 concern the 4 jobs of station 15.

DB 38 shows that the series of configuration DBs ends here.

## 7.7 Diagnosis in practical application

---

As seen, extensive information is offered in the diagnostic flags, diagnostic registers and diagnostic DBs. From this diversity, a logical, practical application is to be extracted.

A diagnostic is to be added to our example, TEST2 from section 6.2, which will show the operator of the installation which slave stations are in the network and which are not.

At the master station (PCD4) instead of the first PCD4.E100 input module a PCD4.B900 I/O module is used. At outputs 0 - 15, the status of slave stations 0 - 15 is to be shown:

output = L: station in network  
output = H: station not in network

This diagnostic is to take place in XOB 30.

The diagnostic flag "F\_diag+21" indicates that a station has been added to or removed from the network. This flag can be evaluated and reset in the user program, so that the next event, addition or removal of another station can be registered again.

The diagnostic register "R\_diag+2" contains the number of the last station to cause XOB 30 to be called. The station number is negative if the station is not in the network and positive if the station is in the network.

With these two messages, and some knowledge of the PCD instruction set, the diagnostic task we have set ourselves can be accomplished.

The program part is to be written in pure BLOCTEC structure, i.e. without jumps. (See next page).

It would of course be possible to make the diagnostic even more precise, for example by identifying and reacting to a change in the configuration DB. (F\_diag+22 und F\_diag+23). However, this seems rather far-fetched, as if this DB had actually been modified, other data in the user program ought also to have changed, which would then compromise the correct function of the CPU.

```

xob          30

inc          c          100          ; Event counter

mul          r          r_diag+2    ;| Generate absolute
           r          r_diag+2    ;| value of station
           r          0           ;| number.
sqr          r          0           ;| Auxiliary register for
           r          0           ;| St. no. absolute value

copy         r          r_diag+2    ;| Status flag
           r          r_diag+2    ;| Create n or p

acc          n          ; when st.no. negative
anh          f          f_diag+21  ; and flag = H

cfb          h          0           ; when negative: Call FB 0
           r          0           ; Parameter 1

cfb          l          1           ; when positive: Call FB 1
           r          0           ; Parameter 1

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

exob
;-----

fb           0           ; Station off network

sei          =          1           ; Load index reg. with st.no.
setx         o          0           ; Set corresponding output

efb
;-----

fb           1           ; Station back again

sei          =          1           ; Load index reg. with st.no.
resx         o          0           ; Reset corresponding output

efb
;-----

```

This routine can be integrated into user program "TEST2" and tested.  
(TEST2\_XA)

From :

Company :

Department :

Name :

Address :

Tel. :

Date :

Send back to :

SAIA-Burgess Electronics Ltd.

Bahnhofstrasse 18

CH-3280 Murten (Switzerland)

<http://www.saia-burgess.com>

BA : Electronic Controllers

SAIA<sup>®</sup> Remote I/O PCD1.T240

If you have any suggestions concerning the SAIA<sup>®</sup> PCD, or have found any errors in this manual, brief details would be appreciated.

**Your suggestions :**