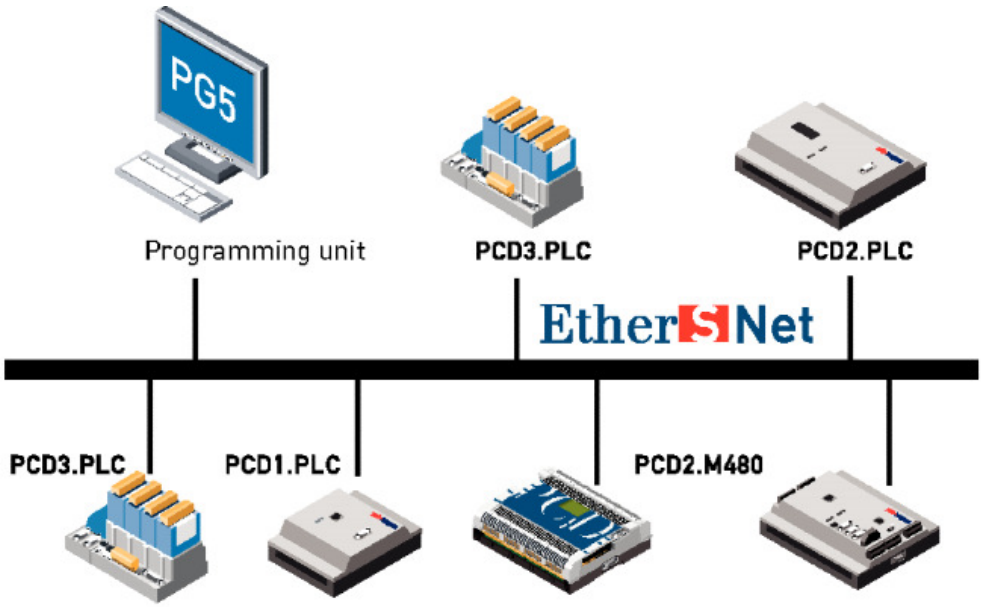
Getting Started:

Ether-S-Bus communication



saia-burgess
Control Systems and Components

Historique du document

Date	Auteur	Modification
31.11.2009	S.Ki, TCS	- Creation of the document (Version 1) Example to make an Ether-S-Bus communication between PCD's - Correction of the text by CD
10.12.2009	S.Ki, TCS	- Restructuration of the document (How to define a MASTER, How to define a Slave)
11.01.2010	S.Ki, TCS	- Adding of chapter 6 et 7
18.01.2010	S.Ki, TCS	- Correction for the release of this document

Contents

1 Introduction

This document explains, how to establish an “Ether-S-Bus” communication between 2 PCD, 3 PCD and more… This “Getting Started” document is accompanied by examples which you can use.

All the examples or explanations in this document will be programmed or done with **PG5 2.0**. (Also there is no problem to program this Ether-S-Bus communication with the PG5 1.4) It is done in the same way, either with PG5 2.0 or PG5 1.4.

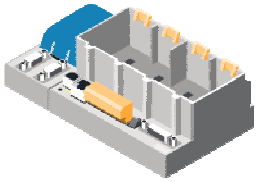
There is only one important difference; it is about the configuration of the ***S-Bus station address*** and the ***IP address*** of your PCD’s.

To do that:
→ on the PG5 2.0, the **Device configurator** is used (see section 3.2)
→ on the PG5 1.4, the **Hardware Setting** “toolwindow” is used

2 Requirements

2.1 Hardware required

- 2 x PCD3.M5540 equipped optionally with a PCD3.S100.



PCD3.Mxxxx

PCD3.S100
(demo module interface):

The examples are realized with two PCD3.M5540. If this HW is not available the same functionality can be realized with:

→ PCD3.Mxxxx (which contains an Ethernet interface or connector)
→ PCD2.M5xxx
→ PCD2.M480 + PCD7.F65x (additional Ethernet interface)
→ PCD2.M170 + PCD7.F65x (additional Ethernet interface)
→ PCD2.M150 + PCD7.F65x (additional Ethernet interface)
→ PCD1.M13x + PCD7.F65x (additional Ethernet interface)

- **either** a crossed Ethernet cable
(only if the network is composed of only 2 stations)

- **or** if your network contains more than 2 stations,
several direct Ethernet cables (CAT5) + one network device as a HUB or a SWITCH

- A PC with USB cable and/or a PGU cable (PCD8.K111) to program your PCD's

2.2 Software required

In this section no special Software tool or something else are required to program the different stations of an Ether-S-Bus networks out of:

- PG5 2.0.110

- The **latest firmware version** of the corresponding PCD's is recommended

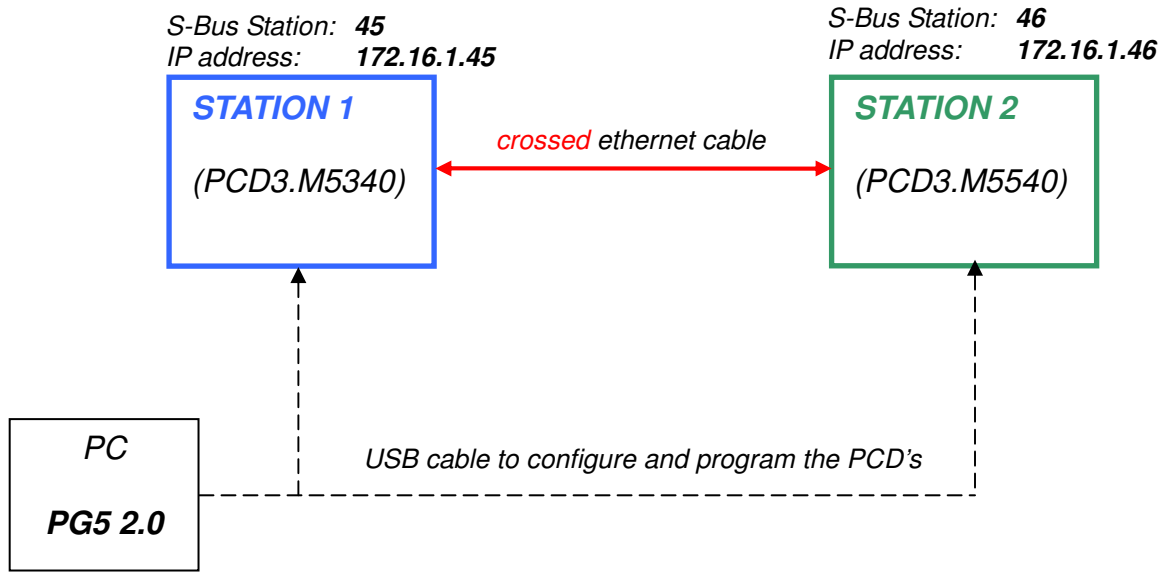
These software and firmware components can be found on our web support site (www.sbc-support.com)

3 Communication between 2 PCD's

3.1 Network connection plan:

There are 2 ways to connect 2 PCD's or stations together. In the following "Network plan" you can see how you could do that and make a choice.

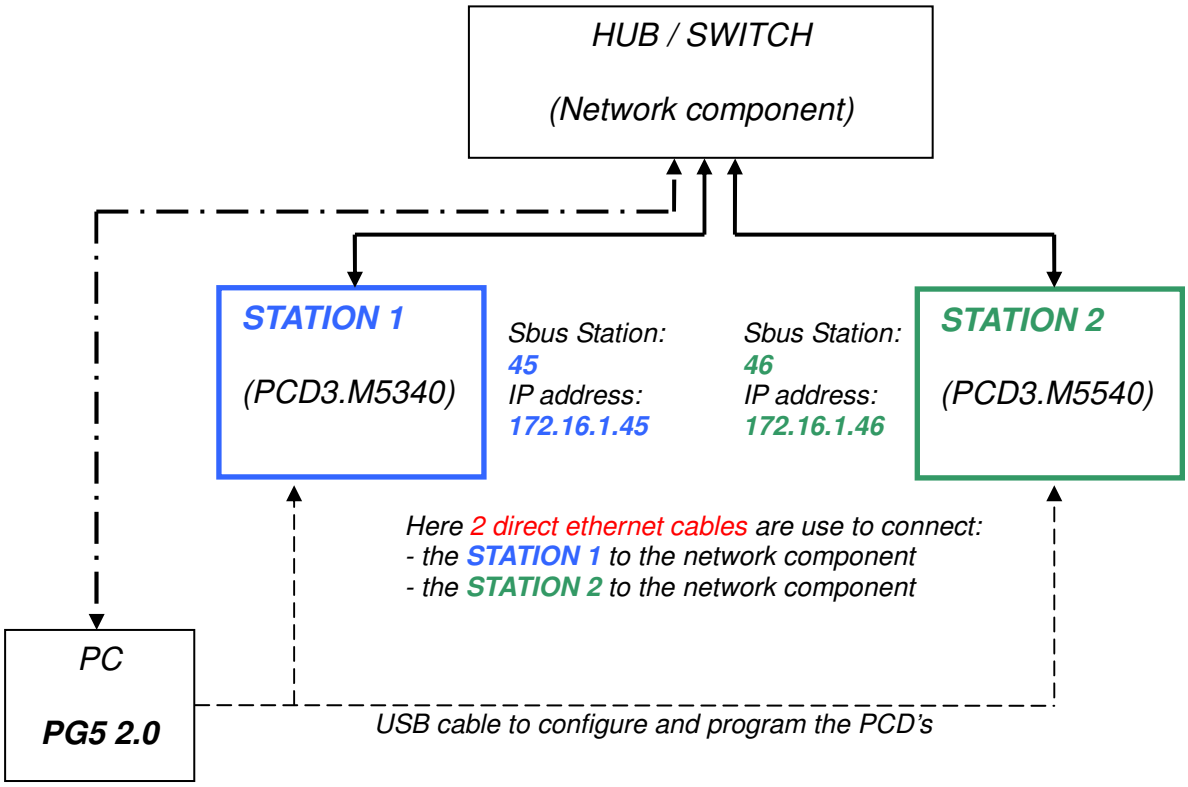
3.1.1 With only one crossed Ethernet cable



In the above network plan, you can just use ***one crossed ethernet cable*** to connect the both PCD's together.

The connection between two Ethernet devices (PCD, PC, …) without using a hub or switch has to be done with a crossed Ethernet cable

3.1.2 With a network component as one HUB or SWITCH



If there is a network component as a HUB or a SWITCH (see section: 7.1) at disposal, you can use 2 direct Ethernet cables to connect the both PCD's to this network component.

A 3rd direct Ethernet cable could be used to connect the PC to the network component. In this way you could program the PCD's per Ethernet and go online with the both PCD's at the same time.

Of course it works only if the PCD's contain already an S-Bus address, an IP address and the same subnet mask. (see chapter 7.2)

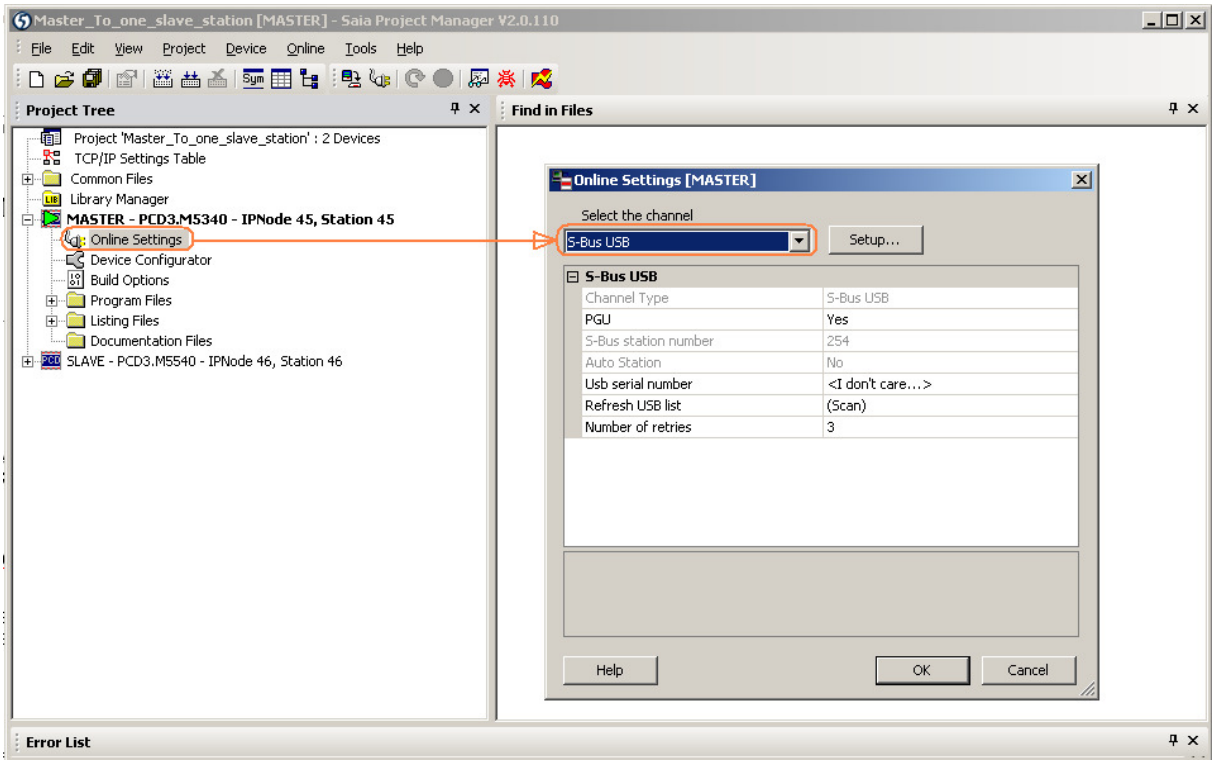
3.2 Configuration of the S-Bus address, IP address and IP Node

As you can see on the two “network plan” (section 3.1.1 or 3.1.2), the STATION 1 and 2 need a **S-Bus address** (= S-Bus station number) and an **IP address** (referenced to the **IP Node** , explained later in this document).

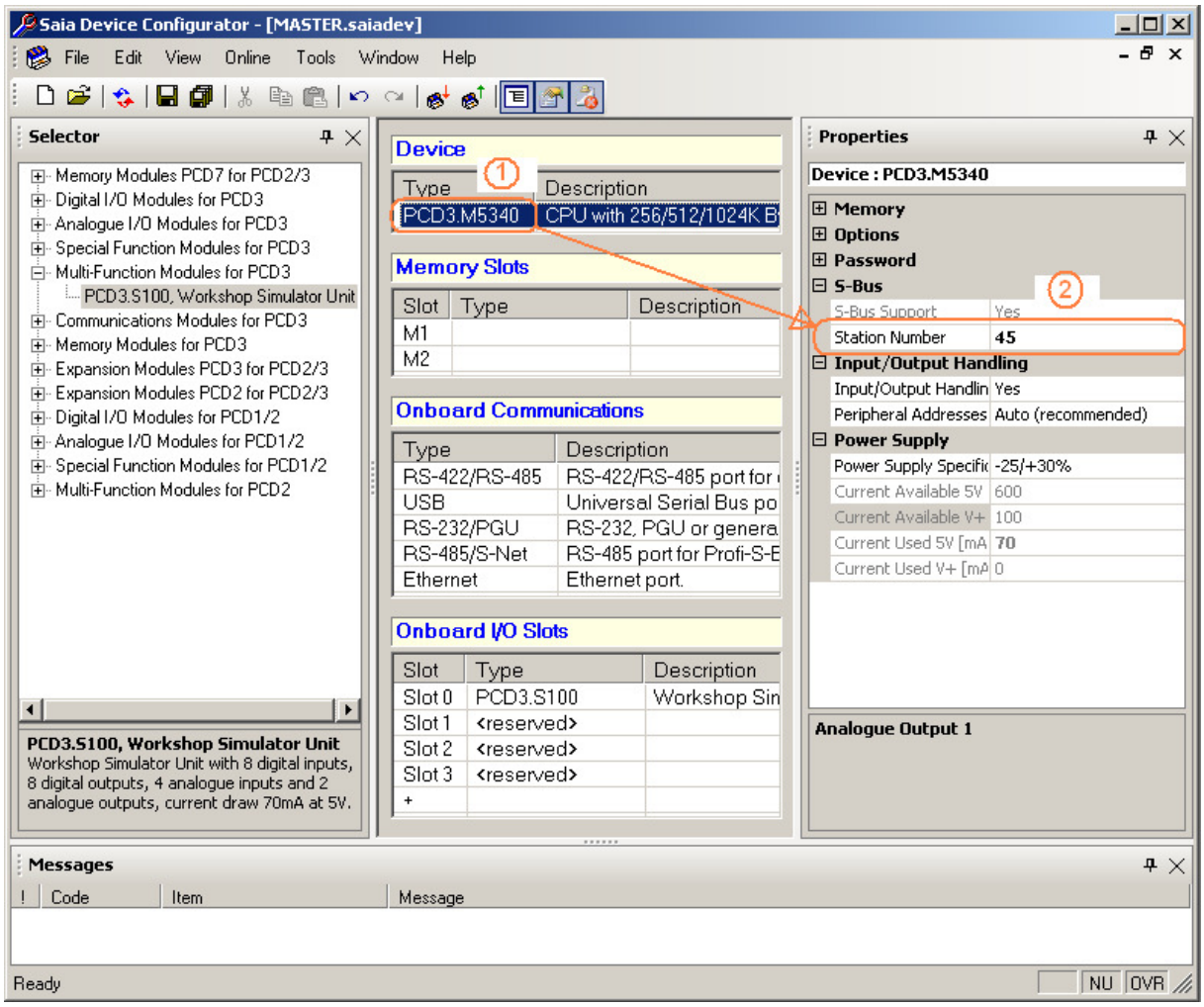
The configuration of the S-Bus, the IP address and the IP Node is always the first thing to do, before to program the communication between the 2 PCD’s.

3.2.1 On STATION 1: (in this example PCD3.M5340)

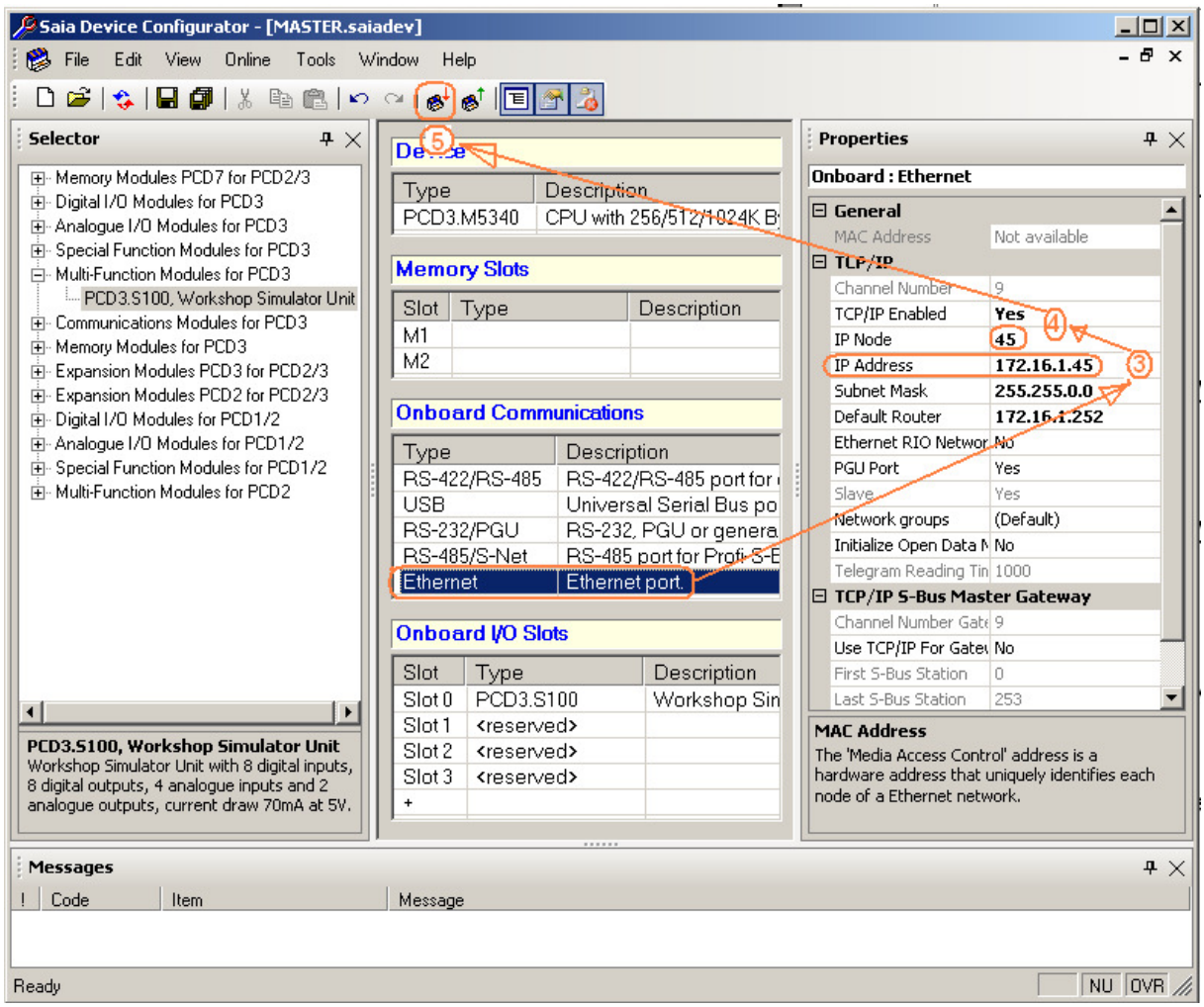
- Create a new project.
- Create a new CPU (device) in PG5 for the (master) station
- Choose in the **Online Setting** the **S-Bus USB** connection (image below)
- Confirm by clicking “OK”



- Open the **Device Configurator**, choose the type of your station
- Set the **SBus Station number → 45** (image below)



- Set the **IP address → 172.16.1.45** (with the corresponding Subnet Mask and Default Router)
- Set the **IP node → 45** (is used in the user program, it corresponds to the IP address reference)
- Plug your USB cable on the STATION 1.
- Then download this hardware setting on the STATION 1 by clicking the button “Download configuration” (image below)



➔ Now the **S-Bus station number**, **IP address** and the **IP Node** are configured on the STATION 1

Make sure that every IP address is only used 1 time and is unique on the network

3.2.2 On STATION 2: (in this example a PCD3.M5540)

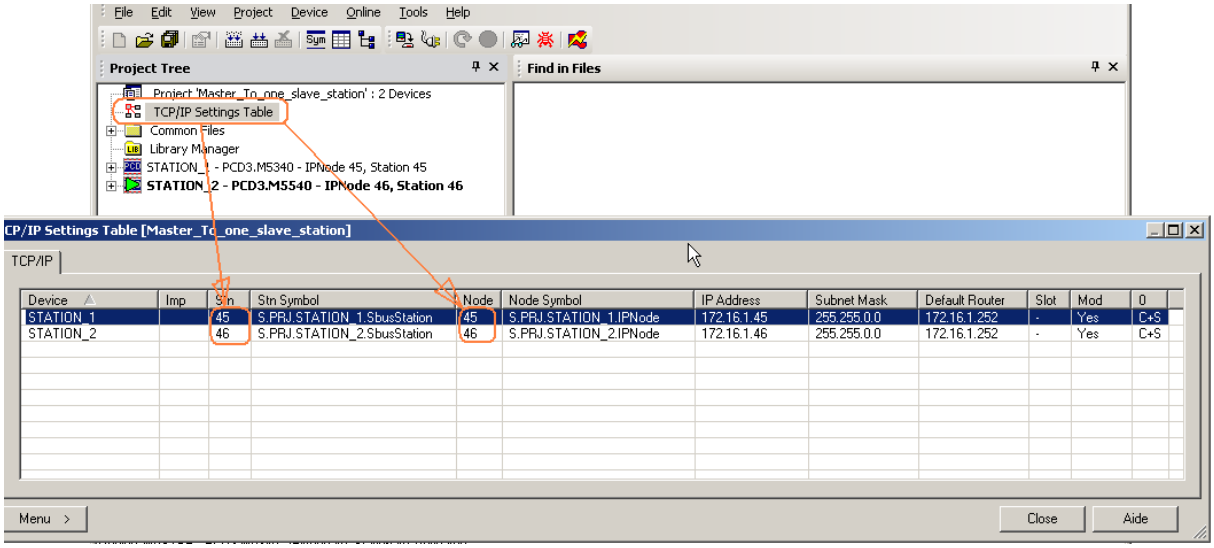
- Do the same steps as the section 3.2.1 but for this (slave) station but adapt the network parameters
- Just change: S-Bus station number → **46**
  IP Address → **172.16.1.46**
  IP Node → **46**

3.3 TCP/IP Setting Table

Before to start the program, you have to know what is this ***TCP/IP Setting Table***.

After you have configured the S-Bus station number, the IP Address and the IP Node in the STATION 1 and STATION 2, you can observe, that the **TCP/IP Settings table** is appeared.

This table (image below) contains the **SBus address**, the **IP address** and the **IP Node** of the STATION's which are configured and which have to be present physically on the Ether-S-Bus network.



In the image above:

- → **Stn**: is identical to S-Bus address
- → (IP) **NODE:** USED WHILE THE PROGRAMMING (reference the IP address of each station)
- → **IP Address:** NOT USED WHILE THE PROGRAMMING (only stored in each PCD configuration)

→ In this case (example) the two STATION 1 and 2 are present in this network

These 2 bit of information (Stn and IP NODE) are used in the user program (in the FBox's), to establish the Ether-S-Bus communication.

This table is downloaded automatically on each PCD together with the user program

In this way each station knows the S-Bus and IP address of the other station which is present on the network and can communicate easily with the others by using these settings (Stn and IP NODE).

3.4 How to define a SLAVE station

There are no additional configurations necessary in order to configure a station (PCD) as slave.

As soon you have configured and downloaded the **S-Bus address**, the **IP address** and the **IP Node** (see section: 3.2), the station works already a Ether-S-Bus SLAVE station.

3.5 How to define a MASTER station

You have to configure the **Sbus address**, the **IP address** and the **IP Node** on a station (see section 3.2), after that this station works as Ether-S-Bus SLAVE station.

To transform it into a MASTER station, you have to do the following (here below).

- create FUPLA file

- add a **SASI SBus Master IP** Fbox: it allows to initialize the Ethernet interface for the usage as master (the slave configuration remains active)

In the "***Properties***" of the Fbox:

→ Then choose the "***name***" of this FBox which will be used as "***reference name***" for the others communication FBox's

→ Choose the corresponding "***channel***" number of the Ethernet interface of the PCD

→ This FBox (above) **defines the STATION as MASTER station**

The Ether-S-Bus has the advantage that a PCD can be Ether-S-Bus *MASTER* and *SLAVE* simultaneously. Thus a PCD can be monitored by e.g. PG5 and at the same time act as master for reading media (flags, registers, …) from another PCD over Ether-S-Bus

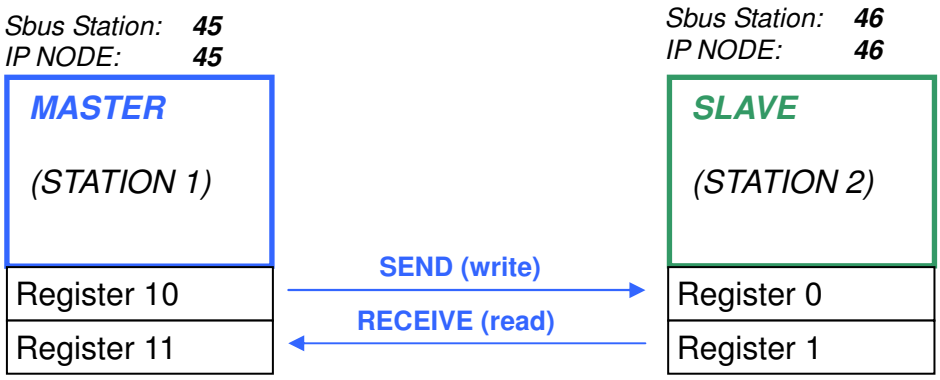
4 Example

In this section you can use any PCD (described in the section 2.1).

4.1 Program example 1: MASTER to SLAVE communication

In this example, we are using 2 PCD3.Mxxxx and we have already configured the network parameters (see section 3.2) of both stations. We have to program now (with the FUPLA editor) our communication application between the STATION 1 and the STATION 2.
Therefore we need to create a program for **STATION 1** to transform it into a **MASTER** station

4.1.1 Data exchange diagram (Master to Slave)



The Slave will automatically answer the requests from any master station for reading/writing its resources (register, flags, …).

The MASTER will send a SEND request: (programmed in Fupla)
→so he will **write** the value of Register 10 into the Register 0 of the SLAVE.

The MASTER will send a RECEIVE request: (programmed in Fupla)
→so he will **read** the value of the Register 1 of the SLAVE and will copy it in his Register 11

KEEP IN MIND WHICH SLAVE VARIABLES ARE USED, WHICH MASTER VARIABLE ARE USED AND WHAT IS EXCHANGED EXACTLY BETWEEN THEM

4.1.2 Configure the SLAVE

In PG5 on the side of the SLAVE (STATION 2) no more actions are required.
The STATION 2 is configured as **SLAVE** station on this Ethernet-S-Bus network as soon the network parameters are downloaded to the device (see section 3.2).

To be a SLAVE station, a PCD need only:

1) SBus Address
2) IP address
3) IP Node (used in the user program instead of the IP address)
4) User program is optional

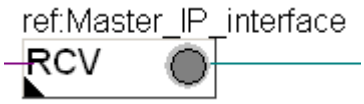
Of course you can insert additionally a user program, as you wish for your application.

4.1.3 Program the MASTER

In PG5 on the side of the MASTER (= STATION 1):

- See **Section 3.5** : "***name***" = "*Master_IP_interface*" (in this example)
"***channel" = 9*** (in this example)
to transform Station 1 into **MASTER** Station

- add a **Receive Integer** FBox : (**read** the value of the remote/**SLAVE station**)



In the "***Properties"*** of the FBox:

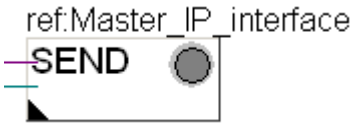
→ ***reference name =*** "*Master_IP_interface*" (it is the name of the **SASI S-Bus Master IP** FBox)

→ ***IP-Node* = 46** (Slave IP address)
→ ***Source station* = 46** (SLAVE S-Bus address you want to access)
These information are also contained in the **TCP/IP setting table** (section 3.3)

→ ***Source element* = Register**
→***Source address* = 0** (address of the slave media, you want to **read**.)

→ The output of the Fbox must be connected to the **Register 10** of the local/master station (for this example)

- add a **Transmit Integer** Fbox: (**write** a value of the remote/**slave station**)



In the "***Properties"*** of the FBox:

→ ***reference name*** = "*Master_IP_interface*"
→ ***IP-Node*** **=** **46**
→ ***Source station*** **=** **46** (S-Bus address of the SLAVE station, you want to access)

→ ***Source element*** **= Register**
→ ***Source address*** **=** **1** (address of the slave media, you want to **write**.)

→ The input of the FBox must be connected to the **Register** **11** of the local/master station (for this example)

- Make a REBUILD ALL
- And download the user program to the PCD.

4.1.4 Test this MASTER to SLAVE communication

Make sure the first input (**enable** input) of the "SEND" and "RCV" FBox are high.
If so, the communication works in case the LED of the FBox's is green.

Check also which resources are written or read with the **data exchange diagram** (see section: 4.1.1).

Otherwise you have at disposition a project, ready to be use about this example.
Name: ***2_PCD_with_MasterToSlave_com.zip***

You can find it in the same compressed folder where you have found this **Getting Started**.

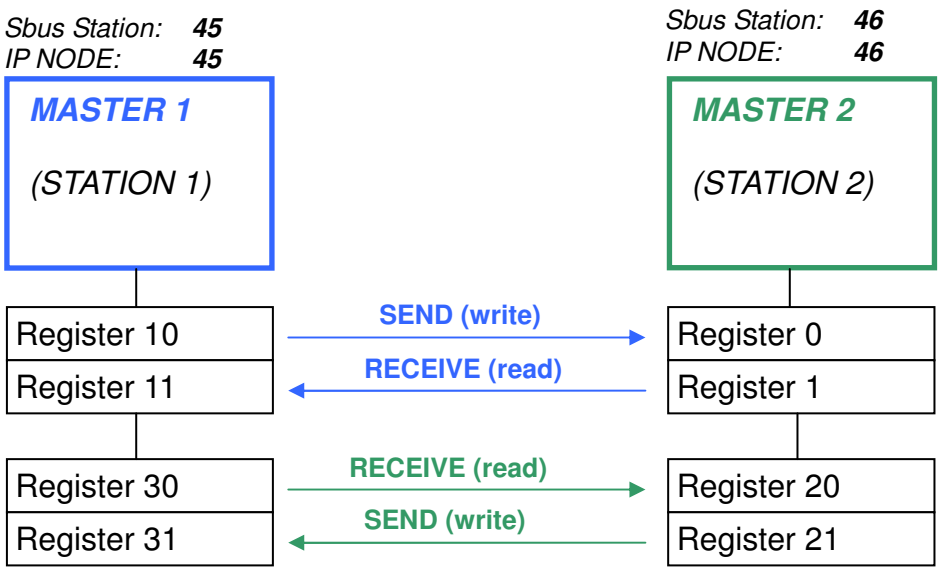
4.2 Program example 2: MASTER to MASTER communication

In this example, we are using 2 PCD3.Mxxxx and we have already configured the network parameters (see section 3.2) of both stations.
The essential communication program process was explained in the section 4.1 .
The only difference is that now in this case both stations are working as master (and in parallel as slave, too).

Now we will program:
- **STATION 1** as **MASTER 1**
- **STATION 2** as **MASTER 2**

As it was explained, the **SASI SBus Master IP** FBox adds to a station the capability to read/write data from/to another station.
We will do that for the STATION 2, to transform it into a **MASTER 2** station. Therefore we need to create a program for **STATION 2.**

4.2.1 Data exchange diagram (Master to Master)



KEEP IN MIND WHICH MASTER 1 VARIABLES ARE USED , WHICH MASTER 2 VARIABLE ARE USED AND WHAT IS EXCHANGED EXACTLY BETWEEN THEM

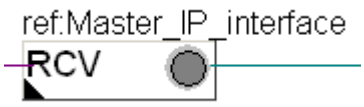
4.2.2 Configure the MASTER 2

Similar to the section 4.1.3 , but adapt the network parameters for the MASTER 2 station.

In PG5 on the side of the MASTER 2 (= STATION 2),

- See **Section 3.5** : "***name***" = "*Master_IP_interface*" (in this example)
  "***channel*" = *9*** (in this example)
  to transform Station 2 into **MASTER 2** Station

- add a **Receive Integer** Fbox : (**read** the value of the remote station / **MASTER 1**)

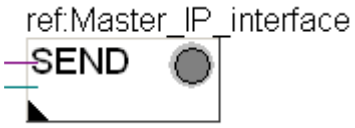


→ ***reference name*** = "*Master_IP_interface*"
→ ***IP-Node*** = **45** (**MASTER 1** IP address)
→ ***Source station*** = **45** (**MASTER 1** S-Bus address)

→ ***Source element*** = **Register**
→***Source address*** = **30** (address of the **MASTER 1** media, you want to **read**.)

→ The output of the FBox must be connected to the **Register 20** of the local/ **MASTER 2** station (for this example)

- add a **Tranmit Integer** Fbox : (**write** the value of the remote station / **MASTER 1**)



→ ***reference name*** = "*Master_IP_interface*"
→ ***IP-Node*** = **45** (**MASTER 1** IP address)
→ ***Source station*** = **45** (**MASTER 1** Sbus address)

→ ***Source element*** = **Register**
→***Source address*** = **31** (address of the **MASTER 1** media, you want to **write**.)

→ The input of the Fbox must be connected to the **Regiser 21** of the local/ **MASTER 2** station (for this example)

As usual, make a REBUILD ALL and then download.

4.2.3 Configure the MASTER 1

THE **SAME** CONFIGURATION ARE DONE IN THE **SECTION 4.1.3**

4.2.4 Test this MASTER to MASTER communication.

Make sure the first input (**enable** input) of the FBox “SEND” and “RCV” are high.
If so, the communication works in case the LED of the FBox is green.

Check also which resources are written or read with the **data exchange diagram** (see section 4.2.1)

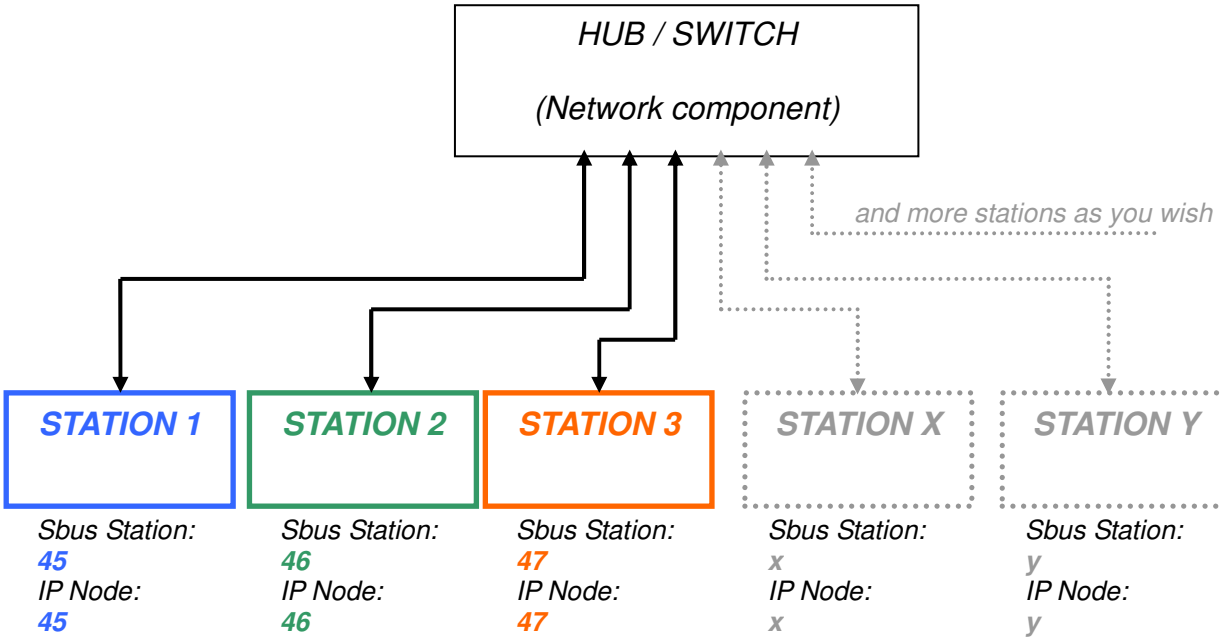
Otherwise you have at disposition a project, ready to be use about this example.
Name: ***2_PCD_with_MasterToMaster_com.zip***

You can find it in the same compressed folder where you have found this **Getting Started**.

5 Communication between more than 2 PCD's

5.1 Network plan

Here below you can see an example of network with several stations. Each has their own network parameter. You can modify their parameters as you wish.



5.2 Programming with additional stations

- As you can see in the above network plan, the networking of more than 2 stations is only possible with an **HUB** or a **SWITCH** (a network component)
- The crossed Ethernet cable is only used when the communication is only done between 2 PCD's.
- In your PG5 project, create the same number of CPU's than the number of station which will be present on your Ether-S-Bus network
- As already explained in this document (**section 3.2**), the first thing to do to integrate a STATION (PCD) in a network is the configuration of a) a **S-Bus address**, b) an **IP Address** and c) an **IP node** of each STATION.

- As soon the setting of these 3 parameter are done, each station is to considered as SLAVE station (**section 3.4**)

- Now choose which station(s) has(have) to be a MASTER station and program it/them (**section 3.5**)

- Program the data exchange between the different master and slave stations (**section 4.1.3**)

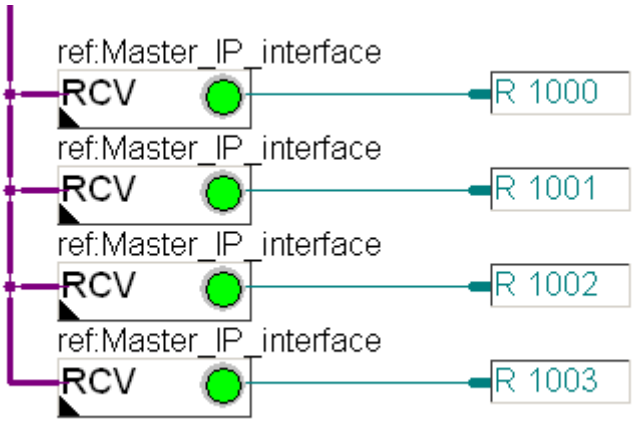
6 Performance improvement possibilities

6.1 Exchange the most data as possible per FBox "SEND" or "RCV"

In the programming of an Ether-S-Bus communication you use eiter an **SEND** FBox or an **RCV** Fbox to write/read resources to/from the remote station

1st possibility: (not the best)

For example, if you have to read 4 register of the remote station, it could be possible to put into the Fupla file 4 corresponding **RCV Integer** FBox's.

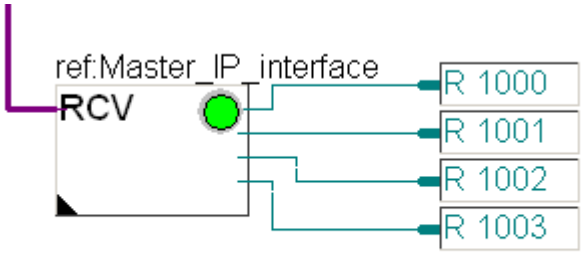


That mean's that the PCD (PLC) will send 4 request telegrams on the Ether-S-Bus network.

The advantage of this method is that no previous work for organizing "blocks" of resources is required

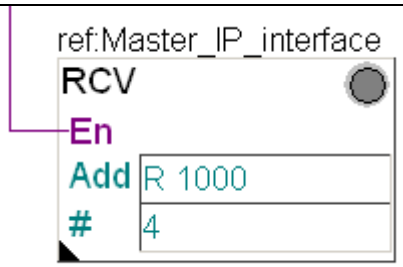
2nd possibility: (better way)

To improve the reading of 4 register of a remote station, use only 1 **RCV Integer** FBox (instead 4 **RCV Integer** FBox's).



It allows discharging the Ether-S-Bus network because in this case only 1 request telegram will be sent to read simultaneously the 4 register of the remote station.

Hint:

With the "**Read Integer quick**" FBox up to 32 Register (corresponding to 128 flags also) can be read per time. This is the biggest amount of data possible per (Ether) S-bus telegram and thus the most efficient.	

Remark:
On a network where few frame "surfed", the way to read or write resources of a remote station, has not a really importance, if you use 4 Fbox's to read/write 4 registers or if you use only 1 FBox to read 4 registers.

But if the network is very (over-)loaded, the 2nd way to read/write is better than the 1st. You will notice it.

6.2 "SASI-Station IP" FBox

This **SASI Station IP** Fbox is very usefull, in the explained following situation:

In a network which is composed of several slave stations and at least 1 MASTER station, it can happen that several slaves lose their connection with the master for diverse reasons (bad address settings, broken cable, etc…).

As result if the enable inputs of each RCV/SEND FBox are always high, these FBox's will always send read/write request to the slave stations (also to the disconnected) and that will strongly increase the time to access of each connected station.

Example of a long access time
- You have programmed the communication for 10 slave station
- 9 slaves are defected (because of wrong addressing, broken cables, etc…)
- Just 1 slave communicate good
- Response timeout: 250 [ms] (applied for each slave station)
- Number of retry: 3 times (applied for each slave station)

The master needs about **6.75 [s]** to access the only slave which works.

Why ?
Every time that the master doesn't succeed to reach a slave station by sending a request, the master will retry to send 2 other request with a default *Response Timeout* of 250ms. Then he tries to access the next station (which also doesn't work) and so on…

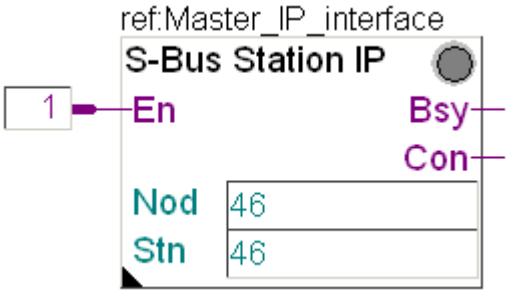
That means the time to access a connected slave station is equal to a time resulting of the sum of the number of retry per not connected slave station, multiplied by the response timeout.

→ **6.75** [s] = **3** (retry) x **0.25** (timeout) x **9** (defected station)

This is problematic if the required resource (of the worked slave) change every second. Because you could access it approximately only each 6.75 [s]
And the effect is greater with 100 disconnect stations and more…

Solution:

To avoid this long data exchange time, add in the program of the master an **S-Bus Station IP** FBox for every slave communication.



This Fbox must be placed before the SEND/RCV Fbox (and of course after the **SASI S-Bus Master IP** Fbox).

For further information, see in the help of the FBox

7 Appendix

7.1 What is the difference between an HUB and a SWITCH ?

	HUB	SWITCH
Connectivity	- Possess several Ethernet Port - Connect together different devices containing an Ethernet interface, to build a network	
Behavior		- Learn the MAC address of a remote station connected on its Ethernet port. - Store this MAC address and map it with the corresponding Ethernet port where this remote station are connected
Data exchange process	A received frame on 1 port is always duplicated and transmitted on all other ports	Every received frame contains a destination MAC address. This destination MAC address is read and the received frame is lead to the correct Ethernet port

7.2 Which IP Address and subnet mask to use ???

For these informations we recommend you to read the “word” document which is contained in the same folder where you have opened this “Getting Started”.

Name: ***IP_in_general_application_V2.0.doc***

7.3 Debugging: Meaning of the color of the Fbox LED

LED of the FBox:			
SASI Master IP	**Station S-Bus IP**	**SEND / RCV**	**Description of the probable LED error**
●	can be green or red	can be green or red	1) The concerned LED are red, but the PCD's communicate good → Make a clear of ***SASI Master IP*** Fbox with the corresponding input and the LED becomes green → Why? The slave/s is/are switched OFF. Your master station was switched ON, before 1 or more slaves was/were switch ON, so the master could not communicate with the slave/s, but now it can.
●	can be green or red	can be green or red	2) The wrong port are Configured in the ***SASI S-BUS master IP*** →By being Online: "***SASI error !***" are mentioned in the Fbox. → Configure the correct **port number** of the TCP/IP interface available on your PCD
●	can be green or red	can be green or red	Otherwise this led can become red consecutively to the following case: 3) , 4) , 5)
can be green or red	●	can be green or red	3) The corresponding slave station is not connected: → Check if the slave station contain the correct IP settings (which are used by the master) → Check if the slave are physically connected (to same network than the master)
can be green or red	can be green or red	●	4) Wrong **IP Node**(3.3) is configured in the ***SEND / RCV*** Fbox → The IP Node doesn't exist in the **TCP/IP settings table** (3.3). → The slave containing this IP Node is not present on the network
can be green or red	can be green or red	●	5) Wrong **Source Station** (3.3). number is configured in the ***SEND or RCV*** Fbox or the slave with this Source Station number are not present on the network

8 Contact us

If you notice something wrong in this document let it us know.

For further information you can ask us per mail or per telephone.

- **Saia-Burgess Controls AG**
  Rue de la Gare 18
  CH-3280 Morat
  Switzerland

- **Technical Support**
  Phone: 026 / 672 71 11
  Fax: 026 / 672 74 99
  Website: www.sbc-support.ch
  E-mail : pcdsupport@saia-burgess.com