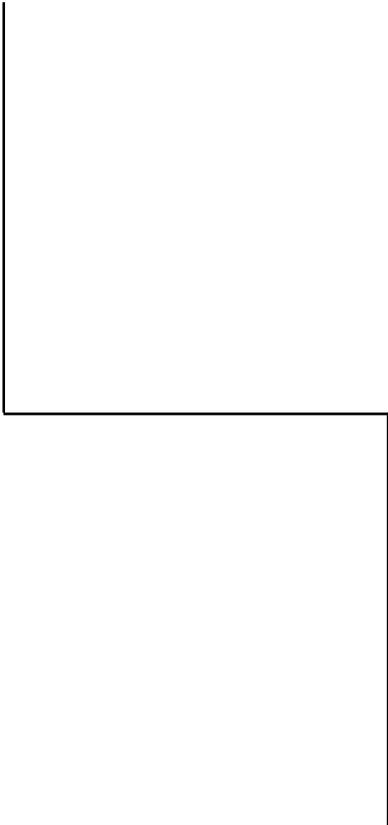


Manual

I/O Com-Server



TCP/IP-Ethernet

Wiring and
Configuration

Rel. 1.0, July 1998

Models: #50xxx

Introduction

The I/O COM Server provides a uniform platform for coupling virtually any I/O to TCP/IP networks.

The PC-like design, with a base board analogous to a motherboard, plus the PC104 bus which allows you to connect all kinds of I/O cards, ensures that the COM Server will be expandable to fit new applications.

To avoid confusion in this guide with all the possible COM Server variations, there are different guide sections for each type.

This basic guide contains information which is irrespective of which I/O cards are installed, including quick installation of the COM Server, arrangement and description of external connections, the configuration tool and menu structure, network configuration of the COM Server, updating the operating software, etc.

© 03/2002 by Wiesemann und Theis GmbH

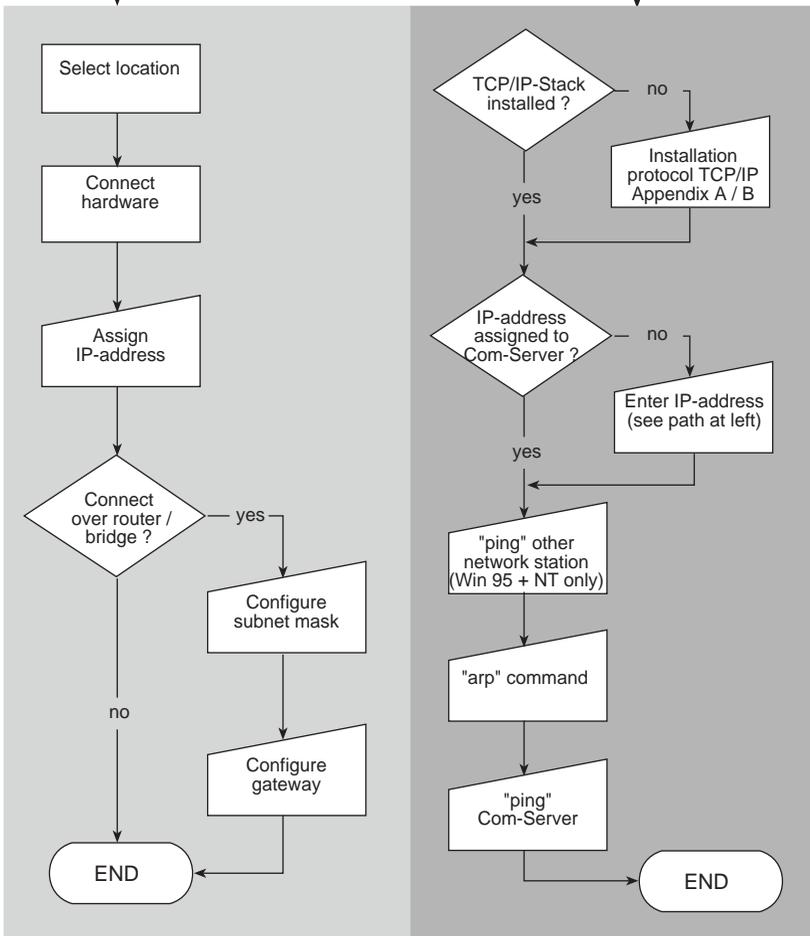
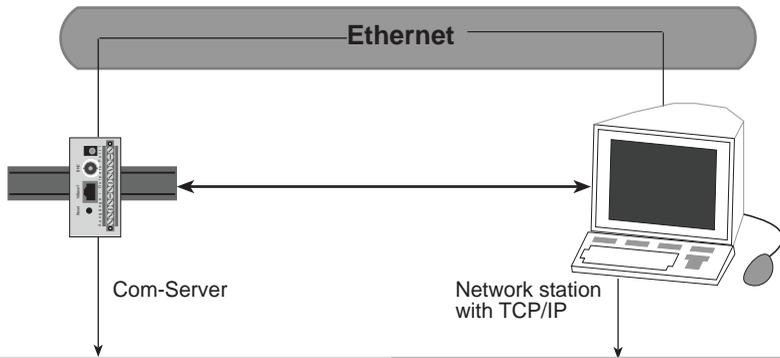
Subject to error and alteration:

Since it is possible that we make mistakes, you mustn't use any of our statements without verification. Please, inform us of any error or misunderstanding you come about, so we can identify and eliminate it as soon as possible.

Carry out your work on or with W&T products only to the extent that they are described here and after you have completely read and understood the manual or guide. We are not liable for unauthorized repairs or tampering. When in doubt, check first with us or with your dealer.

Contents

	Quickstart	4
1	Assigning the IP address	5
2	Tasks of the I/O Com-Server	6
3	Connections, operating elements and displays	7
3.1	Ethernet port	7
3.2	RS232 interface	8
3.3	Supply voltage	10
3.4	Reset	10
3.5	LED displays	10
3.6	Form factors	12
4	Configuring the I/O Com-Server	13
4.1	The I/O Com-Server configuration menu	14
4.2	Menu: INFO Com-Server	15
4.3	Menu: SAVE Setup - Non-volatile memory	15
4.4	Menu: SETUP System	16
4.4.1	Menu: Setup System ▶ Setup TCP/IP	16
4.4.2	Menu: Setup System ▶ Set Password	17
4.4.3	Menu: Setup System ▶ Flash Update	17
4.4.4	Menu: Setup System ▶ Factory Defaults	17
4.4.5	Menu: Setup System ▶ Reset Com-Server	17
5	Software Update for the I/O Com-Server	18
5.1	Network software update under Windows NT / 95	19
5.2	Network software update with TFTP	20
5.3	Software update over the serial interface	21
	APPENDIX A Protocol TCP/IP under Windows 95	22
	APPENDIX B Protocol TCP/IP under Windows NT	23



1. Assigning the IP Address

The Com-Server is factory set to IP address 0.0.0.0. Before you can enter a new one in the Com-Server, you must specify a valid IP address for your network. Obtain this from your system administrator. If you have only a small network where there is no routing, use the IP address of your PC and change only the last position.

Several options are available to you for the first assigning of a valid IP address for the Com-Server.

Note: The IP address must be unique network-wide!

Using the "arp" command (Windows 95, Windows NT, UNIX)

Requirement is a PC which is in the network segment of the Com-Server and which is installed on the TCP/IP protocol. Read out the MAC address of the Com-Server on the device (e.g. EN=00C03D0012FF). Under Windows you then "ping" another network user and then call up "arp":

```

                                arp -s [IP-address]    [MAC-address]
e.g. under Windows:           arp -s 172.16.231.10    00-C0-3D-00-12-FF
e.g. under SCO UNIX:          arp -s 172.16.231.10    00:C0:3D:00:12:FF

```

Now "ping" the Com-Server (e.g. ping 172.16.231.10).

Note: This method can only be used if the Com-Server does not yet have an IP address, i.e., the entry is 0.0.0.0. To change an existing IP address, use a different method or the configuration menu via TELNET.

Entering over the serial interface when starting the Com-Server

Connect a terminal to Port 0 and configure the terminal's serial interface to 9600 baud, no parity, 8 bits, 1 stop bit, no handshake. Interrupt and reconnect the power supply to restart the Com-Server. If the green status LED comes on, press the letter "x" on the terminal at least 3 times until the text "IPno.+<Enter>" appears on your terminal.

Now enter the IP address in the usual format (xxx.xxx.xxx.xxx) and conform your entry with <Enter>. If the entry was correct, this is acknowledged with the assigned IP address, otherwise with "FAIL" and the current IP address. This procedure may be repeated as often as desired.

Using the RARP-Server (UNIX only)

Activate the RARP server and enter the MAC address in the /etc/ethers file and the IP address in the /etc/hosts file. Connect the Com-Server to the network in the same segment as the RARP server.

Example:

Your Com-Server has the MAC address EN= 00C03D0012FF (sticker on the unit).

You want to assign the IP address 172.16.231.10 and the alias WT_1 .

Entry in the file /etc/hosts: 172.16.231.10 WT_1

Entry in the file /etc/ethers: 00:C0:3D:00:12:FF WT_1

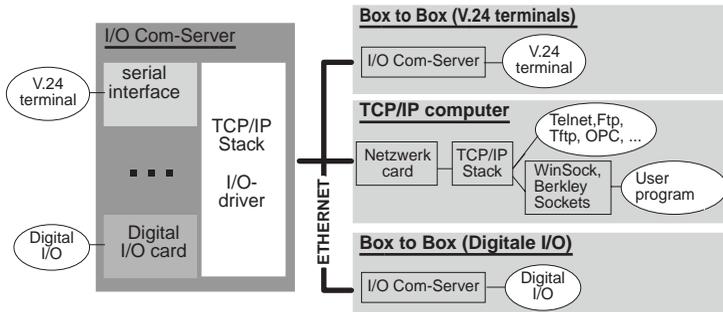
If the RARP demon is not yet active, start it now using the command rarpd -a.

2. I/O COM Server Functions

I/O COM Servers permit direct communication from any computer world with various I/O devices (such as digital I/O, serial devices, ...) over an Ethernet network.

Any device having a serial port – whether digital I/Os or measuring devices, controllers or peripherals of all kinds – can be accessed through an Ethernet LAN without any problem. And you can make the connections without affecting any already present network processes using the existing cable; existing repeaters, bridges or routers can all be used.

The I/O COM Server contains a motherboard which provides the platform for coupling the various I/Os to the network. The number and type of I/O cards depends on the respective I/O COM Server model. As an option, the COM Server can also be ordered with a serial port.



The I/O COM Server has a configuration port which is invoked over the network using the 'telnet' command. The setup menu for the basic configuration is identical for all COM Server models. The setup menus for the various I/O cards vary according to the driver and are described in the corresponding user's guide.

The individual I/Os themselves are accessed using TCP/IP protocol. Which higher protocols for the TCP/IP suite (e.g. Telnet, FTP, ...) are available for data transmission again depends on the type of I/O.

3. Connections, Operating Elements and Displays

The installation location of the I/O COM Server should be selected such that the maximum permissible cable length of 185m for coax cable or 100m for twisted pair cable is not exceeded. If this is not possible, use of Ethernet repeaters and hubs or routers will allow you to extend the length of the network worldwide.

Please be sure that terminal devices are turned off before plugging or unplugging connectors. The location of the individual connections can be found in the illustrations in section 3.6.

A hat rail attachment on the back side of the I/O COM Servers allows you to directly rail mount them.

3.1 Ethernet Connections

One IEEE 802.3-compatible 10Base2- (=BNC) and 10BaseT port (=RJ45) each is provided for making the network connection.

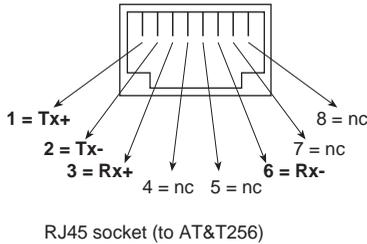
Switching...

between the two network connections is done automatically. Please note here that the cable type will only be detected if the cable is actually connected when the I/O COM server is turned on. If you want to change the cable type, replace the cable and then press the RESET button on the bottom of the unit.

10BaseT, Cable Type: Twisted Pair

The shielded UJ45 female on the bottom of the unit is used to connect the I/O COM Server to a 10BaseT hub (#55603). Pin configuration is per a standard MDI interface (AT&T258), so that a 1:1 cable can be used here.

The current link status is indicated by the Error LED's on the front panel. If all are flashing simultaneously at a 1-2 second interval, this means the connection to the hub was not properly established.



10Base2, Cable Type: Coaxial

The BNC connection is used to integrate the I/O COM Server into a 10Base2 segment. When using standard wiring techniques with BNC T-splitters, please note that any separation of the coax cable which might be necessary could cause problems for other users. Consult with your network administrator before making these connections.

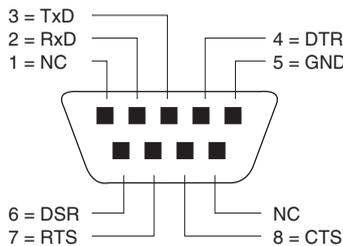
Caution: If the I/O COM Server is located at the end of a segment, the cable must be correctly terminated with a 50W terminating resistor.

3.2 Serial Port

All I/O COM Servers come standard with an RS232 module, since this is needed among other things for configuration. Whether the serial port is also available via the network depends on the COM Server version.

The pin configuration of the RS232 interface is identical to that of a PC, so that standard cables can be used. Be sure the I/O COM Server port and the port on the serial terminal device are configured for the identical transmission parameters and handshake procedure.

Please see the next page for a detailed pin description along with the W&T cables for the main areas of application.

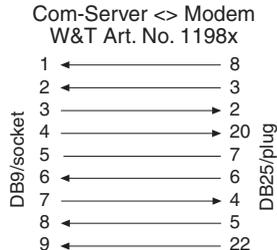
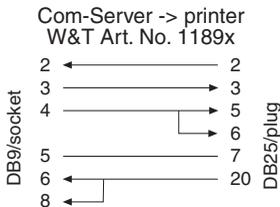
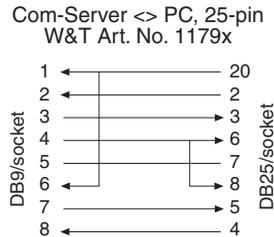
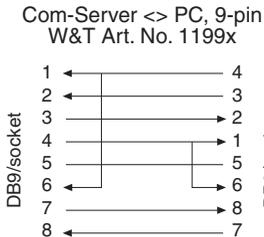


RS232 pin assignments, DB9 plug

3.2.1 Pin configuration and function, RS232

Pin	Direction	Signal	Description	Default function ⁽¹⁾
1	-	NC	Not Connected	-
2	IN	RxD	Receive Data	Data input
3	OUT	TxD	Transmit Data	Data output
4	OUT	DTR	Data Terminal Ready	+12V with active connection to PC (LOCK/UNLOCK)
5	---	GND	Signal Ground	---
6	IN	DSR	Data Set Ready	Data sent only at +3...12V
7	OUT	RTS	Ready To Send	Handshake output +12V = Ready to receive -12V = Not ready to receive
8	IN	CTS	Clear To Send	Data sent only at +3...12V
9	-	NC	Not Connected	-

⁽¹⁾ Can be temporarily modified by the respective software



3.3 Supply Voltage

Supply voltage is introduced through two clamp terminals (24V / GND) on the top of the unit. The feed voltage must be between 12V DC -10% and 24V DC +20%. Current draw with no load depends on the I/O COM Server model, the supply voltage level and the type of network connection (see Technical Data in the appendix to the I/O card description).

3.4 Reset Button

Pressing this button resets the internal microprocessor, thus restarting the COM Server. Since this results in loss of all data in the COM Server, use this function with caution.

3.5 LED displays

→ Power LED

Indicates the presence of supply voltage. If the LED is not on, please check for correct wiring of the supply voltage.

→ Status LEDs

turn on whenever there is any network activity involving the corresponding port. The 'Status Port 0' LED indicates network traffic for serial Port 0 only. The 'Status Port 1..n' indicates network traffic for all other ports.

The LED function for receiving refers only to data and control packets which are addressed directly to the I/O COM Server.

Periodic flashing indicates that the port has a valid connection to another network station. The 'Status Port 1..n' LED indicates the connection status for Port 1. If there are additional ports present (Port 2, ..), the LED flashes when at least one of the ports has a connection to a network station. The status of each individual port can be read in the I/O COM Server configuration menu.

Arriving user data are indicated by momentary flashing of the LED.

→ Fehler LEDs

The Error LEDs use various flashing codes to indicate error conditions on the unit or on the corresponding port.

1 x flashing of all Error LEDs = Check network connection

If the 10BaseT port is being used, the I/O COM Server cannot receive a link pulse from a hub. Check the cable or the hub port. When using the BNC terminal make sure the coax cable is properly terminated and that there are no cable breaks or short circuit.

All Error LEDs on = Self-test error

The self-test which is carried out after every start or reset of the I/O COM Server could not be finished correctly.

This error can occur if you have prematurely cancelled a software update without loading the complete operating software. The I/O COM Server is no longer operational in this condition. Repeat the software update over the network (see section 5). Address the I/O COM Server using the IP address which is assigned to it.

If the error cannot be cleared or if it occurs unrelated to a prior software update, please return the unit for evaluation.

→ Error LED Port 0

The Error LED for Port 0 uses various flashing codes to indicate error conditions on the serial port. I/O COM Servers with the RS232 option allow the Telnet configuration tool to be used for reading out the last 5 errors as error text including the system time (time since the last I/O COM Server restart).

2 x blinking of the error LED = Check serial data format

At least 1 character with a parity or frame error was received at the serial port, or the data register of the serial receiver chip was written to even though the previous character was not yet read out. Check for correctly configured serial parameters, handshaking and connection cable.

3 x blinking of the error LED = Check serial handshake

The serial connected device is not responding to the handshake stop signal set by the Com-Server and is continuing to send data. The result can be an overflow of the serial ring memory and a loss of data. Check the handshake configuration of the devices and correct wiring of the connection cable.

Error LED Port 1 flashes = port-specific error

This flashing code indicates error conditions on one or more I/O ports. Please consult the manual for the corresponding card drivers.

3.6 Form factor

4. Configuring the Com-Server

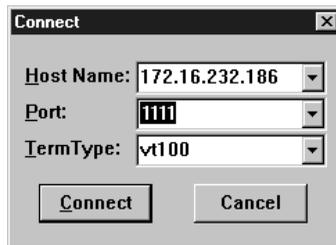
The Com-Server is configured with the help of "Telnet" network protocol. The requirement is that a valid IP address has been assigned to the Com-Server in this network which can now be used to access it (see section 1.2). Configuring can be done with virtually any computer with a network connection and TCP/IP protocol (see Manual for TCP/IP-Ethernet, Appendix A/B). The file *telnet.exe* is included standard with Windows 95 and NT. Find it in the Windows master directory.

To get to the configuration menu, run telnet.exe:

Telnet [IP address] 1111

[Address] = IP address of the Com-Server
1111 = Configuration port of the Com-Server

In Windows NT or Windows 95 you simply call up "telnet" and enter the IP address of the Com-Server as the host name and port number "1111" in the "Connect" "Remote system..."



If the connection was established, you will see the following on your screen:

```
*****
*      W&T - E/A Com-Server      *
*****
1. INFO  System
2. SETUP System
3. SETUP Port ... (card type)
... SAVE Setup
```

The number of ports for which a setup is displayed depends on the I/O COM Server model. "SETUP: Port 0" is always the configuration menu for the serial port and is only accessible when using I/O COM Servers having the RS232 option. For all others the setup menus begin with number 1.

4.1 The I/O Com-Server configuration menu

The following is an overview of the Com-Server configuration menu. You will only see one level of the selected menu on the monitor at one time. To see the next level, simply enter the number of the desired menu and press ENTER. Use 'q' or just the ENTER key to go back.

The last configured value is always shown in parentheses. If you change this value, the new value will appear in parentheses the next time. The value only becomes valid in the Com-Server however if you have saved it under **SAVE Setup**.

As long as you have not called up this menu option, you can move through the entire menu and change values without really changing anything.

*Note: If you exit the configuration menu by closing the telnet connection without selecting **SAVE Setup**, the old configuration is retained.*

4.2 Menu: INFO Com-Server

This menu allows you to call up device-specific parameters such as version number and create date for the software, the MAC address for the device, etc.

→ **Cable Type**

Displays the Ethernet connection of the Com-Server (Twisted Pair).

→ **MAC-Adresse**

Displays the Ethernet address of the Com-Server. This number is factory set and registered, and may not be changed.

→ **SOFTW Date/Rev.**

Shows the create date and version number of the operating software in flash memory.

→ **HARDW Rev.**

Displays the revision status of the Com-Server hardware.

→ **Run Time**

Displays the time in hours and minutes since the last I/O Com-Server restart.

4.3 Menu: SAVE Setup - non-volatile memory

At the prompt "Save Changes?", respond with 'y'. If a correct entry was made, "Saving..." appears on the monitor. The Com-Server saves all the changes you have made in a non-volatile memory (EEPROM). Once the data have been saved, they are activated again each time the Com-Server is turned on or reset.

If an incorrect entry has been made or by pressing the ENTER key, you are returned to the main menu without saving the values.4.4 Menü: SETUP System

4.4 Menu: Setup system

This menu is used to configure all the parameters which affect the Com-Server's operating system and which are independent of the serial or I/O ports

4.4.1 Menü: SETUP System P Setup TCP/IP

→ **IP address**

Here enter the IP address if you would like to change it. Please note that this number is not freely selectable, but rather depends on the network address of the TCP/IP network. The entry form complies with the usual syntax (e.g. 172.16.231.5).

Note: The Com-Server is not accessible under the new IP address until you have established the telnet connection.

→ **Subnet Mask**

The subnet mask only needs to be entered if the Com-Server is supposed to route connections to another subnetwork. Enter the subnet mask of the subnetwork in which the Com-Server is located (e.g. 255.255.255.0). Please note that the subnet mask depends on the class of the IP address (Class A, B or C). Incorrect entries are automatically corrected when saving.

→ **Gateway**

Here enter the IP address of the gateway if routing to other subnetworks is required.

→ **MTU - Maximum Transfer Unit (default: 512)**

This value determines the maximum packet size of a TCP/IP packet. It refers to the number of bytes (excluding header) which can be sent in a packet. The smaller the MTU is made, the more total network buffers are available in the I/O COM-Server. The selectable range begins at 512 and ends at 1024 bytes. The values can be set in steps of 128 bytes (automatic correction).

Note: For an explanation of the basic concepts of Internet addressing, see the appendix of our manual "Ready in 1 Day for TCP/IP Sockets".

4.4.2 Menü: SETUP System P Set Password

Here you may specify an 8-place, hexadecimal password which protects the configuration menu from unauthorized access. The factory setting is 00000000, which permits unlimited access to the Com-Server's configuration menus.

If a valid password is entered, it is asked for starting immediately after the telnet connection to the configuration port is opened. If an incorrect password is then entered, access to the configuration port is denied.

Note: *Resetting or changing any parameter, including the password itself, can only be done with knowledge of the old password. Therefore the password should always be stored in a safe place.*

4.4.3 Menu: **SETUP System** → **Flash Update**

Before you activate the two modes, quit any active network connections. Then confirm with 'y'. The mode is displayed by illumination of all status LED's on the Com-Server.

→ **Net Update (TFTP)**

Activate this mode if you want to perform an update to the Com-Server operating software over the network using TFTP (see section 4.....).

→ **Serial Update (Port 0)**

Activate the mode if you want to send the files for the Com-Server operating software update serially (see section 4.....).

Note: *You can only leave the update mode by carrying out the update or by pressing the Reset button!*

4.4.4 Menu: **SETUP System** → **Factory Defaults**

Enter 'y' to reset all settings. The configuration is then returned to the factory default settings. The Com-Server closes the telnet connection. Then it carries out a software reset to activate the new configuration.

Note: *Resetting the non-volatile memory causes **all** settings you have made to be lost, including the IP address.*

4.4.5 Menu: **SETUP System** ⇨ **Reset Com-Server**

Select this menu option to carry out a Com-Server software reset. First your telnet connection is properly closed.

Note: *All the data from any still open network connections are lost!*

5. Software-Update for the Com Server

Because we are continuously developing the Com-Server operating software, we have also made it possible to perform a software update on this device. The operating software for the Com-Server is available in two different forms, either as a file with a *bin* suffix or split into 4 files. The files are then named according to the following system:

There are two ways to perform the update itself. If you have access to a network computer with TCP/IP protocol, you can perform the software update using TFTP protocol. Windows 95 does not offer TFTP protocol. Instead we have created a small program (50flash.exe) for performing the update over the network.

The second method is updating over the serial interface.

The network method is significantly faster than using the serial interface and can be done from virtually any computer which has an activated TCPIP stack.

Note: *Never forcibly interrupt the update process by pulling the network plug or pressing the Reset button. The Com-Server cannot be operated after an incomplete update.*

Never mix files with different version numbers in the file names. This will result in a loss of function in the unit.

Send all the files one after the other. The Com-Server automatically recognizes when all the files have been sent and the new operating software is complete. It then automatically carries out a Reset.

The software update has no effect on the Com-Server configuration.

5.1 Network Software Update under Windows NT / 95

Required is a PC having a network port and activated TCP/IP protocol (see Appendix A and B). The two files needed for the update process:

1. The program 50flash.exe (32-bit application for WindowsNT, Windows95), which executes the update, and
2. the file having the suffix *.bin, e.g. C4r1_0.bin, which contains the new operating software for the Com-Server. This file is sent to the Com-Server.

The update process is described step-by-step below. Please adhere to the instructions. An incomplete update will cause the Com-Server to lose functionality!

1. Close **all** connections which may still be active on the Com-Server. All buffers and therefore all data are deleted before the update process begins!
2. Use telnet to start the Com-Server remote configuration tool.

```
telnet [IP-Adresse] 1111
```

Select „SETUP: System Flash Update ▶ Net Update“ from the menu and confirm with 'y'. The Com-Server closes the telnet connection. The green status LED indicates that you are currently in update mode.

3. Run the 50flash.exe program. Enter the IP address of the target Com-Server and the name of the update file (e.g. C4r1_1.bin).

4. Click on the "Start Update" button. The update takes several seconds. It is not complete until the value 100% is displayed in the process display and a message window indicates the end of the update process.

5. Check the Com-Server configuration menu to see whether it has taken the new operating software. The "INFO Com-Server SOFTW Date / Rev" menu must contain the new version number for the software, which is also encoded in the file name (e.g. "C4r1_2.bin" means Version 1.2).

If only the previous version is present, the file with the new operating software is corrupt. Please get in touch with the W&T Hotline.

5.2 Network Software Update with TFTP

Required is a computer having a network port and a TCP/IP stack which provides the network protocols TELNET and TFTP. For the update process you need the update files in split form (e.g. C4r1_1.4_1 ... C4r1_1.4_b).

1. Start the Com-Server remote configuration tool over telnet.

```
telnet [IP-Adresse] 1111
```

Select „SETUP: System → Flash Update → Net Update“ from the menu and confirm with 'y'. The Com-Server closes the telnet connection. The green status LED indicates that it is currently in update mode.

2. Use the TFTP command to send the first file to the Com-Server (C4r*.4_1). **You must use binary mode!** While the data are being sent over the network the Status LED flashes. Then the Com-Server switches over to programming mode and the Error LED comes on. This process may take several seconds. Wait until the Error LED goes off and the Status LED comes on again.

Repeat this process for all four files.

3. The Com-Server recognizes when all the files have been sent and **automatically** performs a Restart. Should the green Status LED come on again after all the files have been sent, repeat step 2 completely. Sending the same file twice does not cause an error in the update itself. The Com-Server waits until it has all the necessary files.

Example: SCO UNIX

Enter the following commands after the respective prompt:

```
# tftp
tftp> connect [ip_number | host_name]
tftp> binary
tftp> put C4r1_1.4_1 [remote filename]
(remote filename = any letter)
```

Now wait until the green Status LED comes on again, and use the *put* command to send the remaining three files.

```
tftp> quit
#
```

4. Check the Com-Server configuration menu to see whether it has taken the new operating software. The "INFO Com-Server SOFTW Date / Rev" menu must contain the new version number for the software, which is also encoded in the file name (e.g. "C4r1_2.bin" means Version 1.2).

If only the previous version is present, the file with the new operating software is corrupt. Please get in touch with the W&T Hotline.

5.3 Software update over the serial interface

Required is a computer having a configurable serial port. The update process is described step-by-step below. Please follow the instructions exactly. An incomplete update will cause the device to lose functionality!

1. Connect the Com-Server's serial port to the serial port of the computer from where you want to perform the update.
2. Configure the computer's serial port as follows:

9600 baud, no parity, 8 bits, 1 stop bit

3. Use telnet to start the Com-Server's remote configuration tool.

```
telnet [IP-Adresse] 1111
```

Select „SETUP System ▶ Flash Update ▶ Serial Update" from the configuration menu and confirm with "y". The Com-Server closes the telnet connection. The green Status LED indicates that you are now in update mode.

4. Use the COPY command to send the first file to the Com-Server. While the data are being sent over the serial interface, the Status LED blinks rhythmically. This may take up to 3 minutes. Then the Com-Server switches over to programming mode, and the Error LED comes on. Wait until the Status LED is again constantly on. Repeat this procedure for all four files.

5. The Com-Server recognizes when all the files have been sent and **automatically** performs a Restart. Should the green Status LED come on again after all the files have been sent, repeat step 3 completely. Sending the same file twice does not cause an error in the update itself. The Com-Server waits until it has all the necessary files.

6. Check the Com-Server configuration menu to see whether it has taken the new operating software. The "INFO Com-Server SOFTW Date / Rev" menu must contain the new version number for the software, which is also encoded in the file name (e.g. "C4r1_2.bin" means Version 1.2).

If only the previous version is present, the file with the new operating software is corrupt. Please get in touch with the W&T Hotline.

APPENDIX A

TCP/IP Protocol under Windows95

To activate TCP/IP, a computer having a network card and Windows 95 is required. TCP/IP is included standard and does not need to be configured for the local network.

- Obtain a valid IP address from your network administrator.
- In the **Start** menu first select **Settings**, then **Control Panel**. Then double-click **Network**.
- Select the **Configuration** tab and click on **Add**, then select **Protocol** and then in this window **Manufacturers: Microsoft** and **Network Protocol: TCP/IP**.
- Click on OK. This returns you to the **Configuration** tab, and now TCP/IP also appears in the list of network components.
- Select **TCP/IP** and click on **Properties**. Enter the IP address for this computer and the subnet mask of the network in which the computer is located on the **IP-Address** tab.

If you also want to establish connections to other networks, enter the IP address of the gateway(s) over which you wish to route your connections on the Gateway tab.

If you have only a local network and do not use gateways or routers, the following rules apply to the subnet mask. Note the first digit in the IP address and select the subnet mask accordingly:

- 1–126: 255.0.0.0
- 128–191: 255.255.0.0
- 192–254: 255.255.255.0

- Confirm again with OK. Now you are prompted to insert the Windows 95 CD-ROM. The drivers will be updated and then the computer must be rebooted to activate the new configuration.

APPENDIX B

TCP/IP protocol under Windows NT

To activate TCP/IP, a computer having a network card and Windows NT is required. TCP/IP is included standard and does not need to be configured for the local network.

- Obtain a valid IP address from your network administrator.
- In the **Start** menu first select **Settings**, then **Control Panel**. Then double-click **Network**.
- Select the **Configuration** tab and click on **Add**, then select **TCP/IP Protocol** from this list.
- Click OK. You are asked whether you would like to use a DHCP server for assigning the IP address. Ask your network administrator, or answer with NO. Now insert your Windows NT CD-ROM in order to install the required drivers.

You are then returned to the Protocols tab, and will now see TCP/IP protocol in the list of network components. Now click on **Close**!

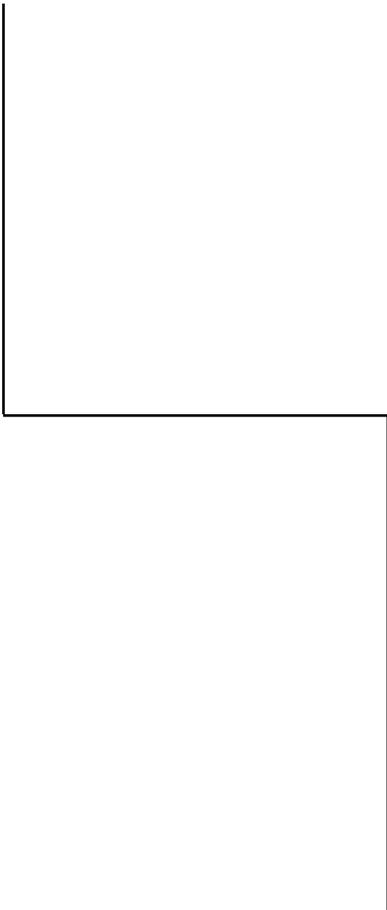
- The connections are updated, and then the TCP/IP properties field appears. Enter the valid IP address for this network. The subnet mask is automatically calculated based on the IP address.

If you want to establish connections for additional networks, the subnet mask specified for your network and a standard gateway must be entered. Obtain this information likewise from your systems administrator.

- Confirm again by clicking OK, and reboot the computer to activate the new configuration.

User's Guide

I/O COM Server



Digital I/O
per TCP/IP

(Wiring, Configuration and
Programming)

Rel. 1.0, July1998

Models: 50210, 50211,
50310, 50311

Introduction

The I/O COM Server provides a uniform platform for coupling a wide variety of I/O modules to the network.

The PC-type design, with a motherboard-like main board and a PC104 bus for connecting all kinds of I/O cards, ensures future expandability of the I/O COM Server for new or growing applications.

In order to avoid confusion with all the possible variations of the I/O COM Server, there are different sections of the User's Guide depending on model.

This Guide contains all information pertaining to the Digital I/O Card, including physical layout drawings, configuration for the various TCP/IP modes, configuration and programming of the Digital I/O Card, etc.

© 03/2002 by Wiesemann und Theis GmbH

Subject to error and alteration:

Since it is possible that we make mistakes, you mustn't use any of our statements without verification. Please, inform us of any error or misunderstanding you come about, so we can identify and eliminate it as soon as possible.

Carry out your work on or with W&T products only to the extent that they are described here and after you have completely read and understood the manual or guide. We are not liable for unauthorized repairs or tampering. When in doubt, check first with us or with your dealer.

Contents

Part 1	Functions of the Digital I/O COM Server	4
Part 2	Addressing Digital I/O Cares in the TCP/IP network	5
Part 3	Physical layout of the Digital I/O COM Server	6
3.1	Pin configurations	7
3.1.1	Digital inputs	7
3.1.2	Digital outputs	7
3.2	Displays (LED's)	7
Part 4	The TCP/IP modes of the Digital I/O COM Server port	8
4.1	Data transfer using TCP/IP sockets	10
4.1.1	"TCP Server" Mode (Standard Mode)	11
4.1.2	"TCP Client" Mode (Menu: TCP Client)	11
4.1.3	"UDP Client" Mode (Menu: UDP Client)	13
4.1.4	"Box to Box" Mode (Menu: Box to Box)	14
4.1.5	Menu: System Options	16
4.2	Menu: SETUP Port.. -> Port State	16
Part 5	Controlling the Digital I/O Card per TCP/IP	18
5.1	Definition of I/O structures	19
5.1.1	Writing and reading the I/O register	20
5.1.2	Checking I/O Card status (models 50211, 50311, 50611 only)	23
5.1.3	Alarms (models 50211, 50311, 50611 only)	25
5.2	The Acknowledge procedure for UDP (optional)	28
5.2.1	Synchronizing the sequence numbers	29
5.2.2	Examples	30
Appendix		
	Technical Data	32

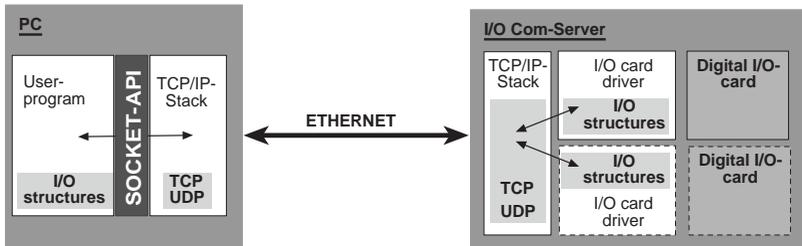
1. Functions of the Digital I/O COM Server

The Digital I/O COM Server consists of a motherboard and one or more digital I/O cards. Each card contains 12 digital inputs which are current sourcing per IEC 1131-2 Type 1, and 12 short-circuit protected digital outputs whose output current is monitored by A/D converters. All in- and outputs can be supplied by a voltage between 12V DC -10% and 24V DC +20%.

The drivers for the I/O cards are a component of the operating software for the I/O COM Server. These drivers handle the I/O cards, write and read the registers for the in- and outputs and carry out all necessary monitoring operations.

The data itself are exchanged between the user program and the card driver in the I/O COM Server using TCP/IP protocol. The data format for transmission is determined by defined structures.

In addition to the usual transmission of the contents of card input and output registers, the I/O COM Server models 50211, 50311 and 50611 also support measuring and monitoring functions for current limiting and for temperature and input voltage monitoring.

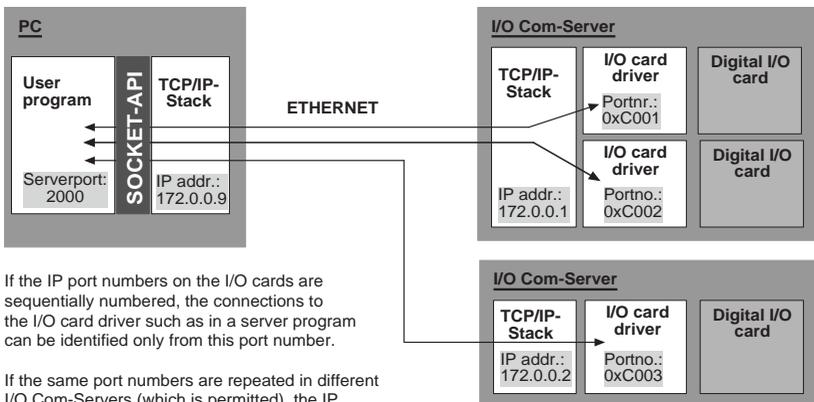


Both TCP client/server and UDP client/server are supported for connection control and data transmission over the network. Each digital I/O card occupies one port in the I/O COM Server and is addressed through a freely selectable port number.

2. Addressing digital I/O cards in the TCP/IP network

Addressing in the TCP/IP network is done in two steps. First the network station itself is addressed with the IP address, then the services of these network stations are addressed with port numbers. Each IP address must be unique throughout the network and each port number must be unique to the network station.

The I/O cards are addressed analogously. At installation an IP address is assigned to the I/O COM Server (see chapter 1 in the Guide "TCP/IP per Ethernet"). This address is used to access all the services of the I/O COM Server. The services themselves are addressed using the port numbers.

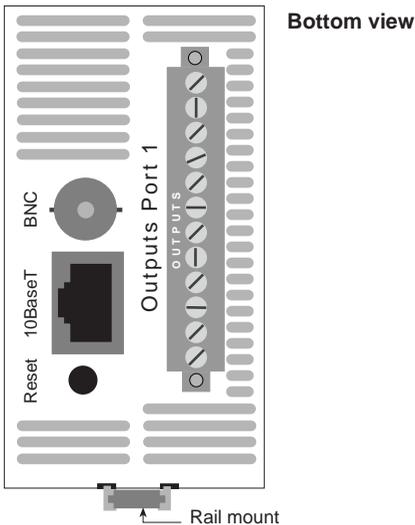
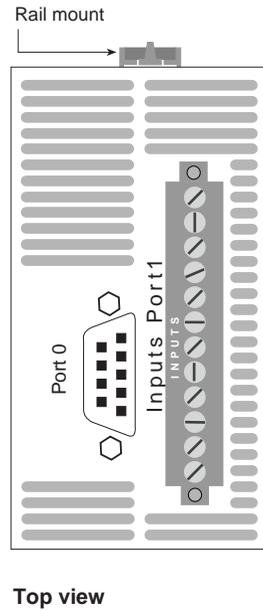
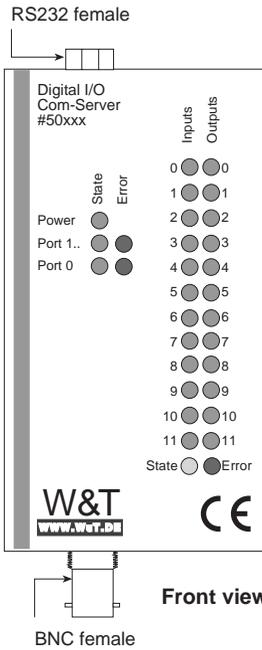


If the IP port numbers on the I/O cards are sequentially numbered, the connections to the I/O card driver such as in a server program can be identified only from this port number.

If the same port numbers are repeated in different I/O Com-Servers (which is permitted), the IP address must always be compared as well.

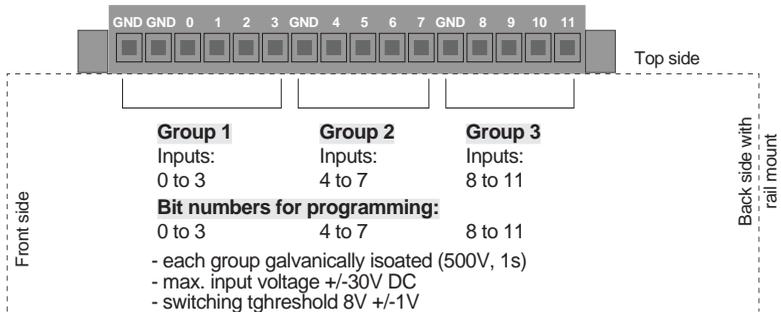
Each I/O card in the I/O COM Server occupies a port and is addressed through a port number. Whereas all the other services have fixed port numbers assigned (e.g. remote configuration via Telnet = 1111), the port numbers for the I/O cards are freely selectable in a range between 49152 - 65535 (hex: 0xC000 and 0xFFFF) and can be configured in the I/O COM Server. In this way all I/O cards "present in the network" are numbered consecutively even if they are located in different I/O COM Servers.

3. Physical layout of the Digital I/O COM Server

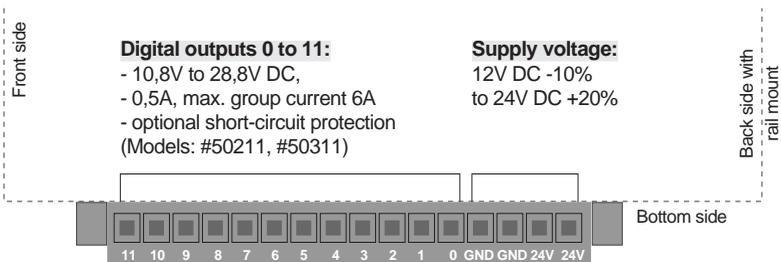


3.1 Pin configurations

3.1.1 Digital inputs



3.1.2 Digital outputs



3.2 Displays (LED's)

Inputs:

Error: These LED's are on if a High level signal was detected on the digital I/O card.

Outputs:

The output LED's are on if the I/O card driver has logically activated the output, i.e. the application program has set the bit for the output.

Error:

The Error LED is on if the measuring system for the digital I/O card is not recognized by the driver, or the measuring system has turned off the outputs due to a short circuit.

Note: Even if the outputs are turned off due to a short circuit, the LED associated with the output remains on if it is logically set!

4. The TCP/IP modes of the Digital I/O COM Server port

The various TCP/IP modes are configured over the network with the help of the Telnet protocol. Configuration is possible from any computer having a network connection and TCP/IP protocol (â Guide for TCP/IP-Ethernet Appendix A / B). The file *telnet.exe* comes standard with Windows 95 and NT. It can be found in the Windows master directory.

To select the configuration menu, run telnet.exe:

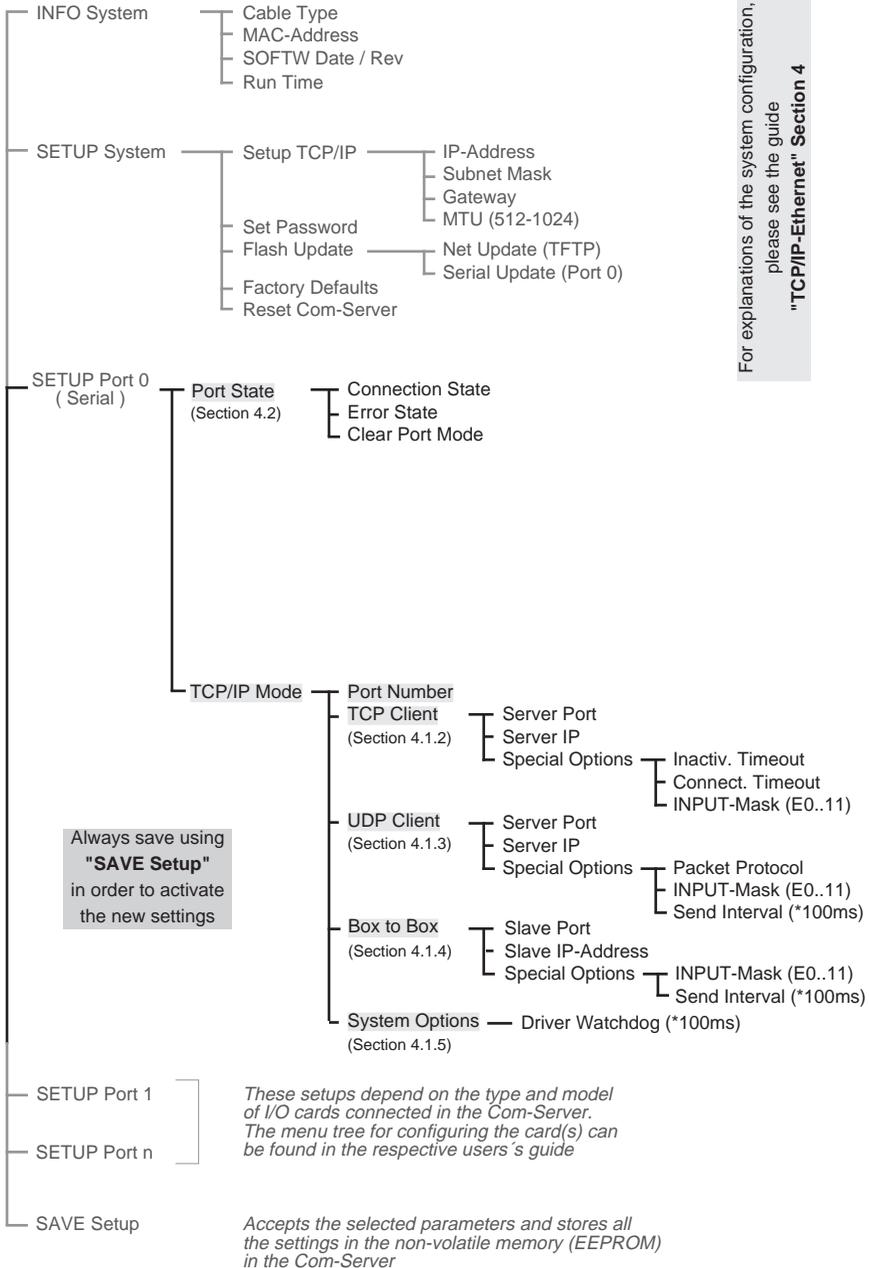
```
Telnet [IP-Address] 1111
```

The overview on the next page shows the entire menu. One level will always be visible on the monitor. Enter a number to page forward or press ENTER to page back.

Any settings you make are only activated in the I/O COM Server after you save using the **SAVE Setup** menu. This means you can move through the menu as you like, change values, and set various modes. If you close the Telnet connection without saving, the I/O COM Server retains all the settings which it had before the configuration menu was opened.

The Digital I/O COM Server has the following four basic configurations:

	Menu	Section
"TCP-Server" Mode (Standard Mode)	—	4.1.1
"TCP-Client" Mode	TCP/IP Mode → TCP-Client	4.1.2
"UDP-Client" Mode	TCP/IP Mode → UDP-Client	4.1.3
"Box to Box" Mode	TCP/IP Mode → Box to Box	4.1.4



For explanations of the system configuration, please see the guide "TCP/IP-Ethernet" Section 4

Always save using "SAVE Setup" in order to activate the new settings

These setups depend on the type and model of I/O cards connected in the Com-Server. The menu tree for configuring the card(s) can be found in the respective users's guide

Accepts the selected parameters and stores all the settings in the non-volatile memory (EEPROM) in the Com-Server

4.1 Data transfer using TCP/IP sockets

The socket API (Windows: "WinSock", UNIX: "Berkley Sockets", ...) makes it possible to implement the various application cases in the form of "Client" or "Server" processes on TCP/IP computers. The API provides the entire functionality for transporting data over the network. You can construct your application program for final processing and evaluating the data to match the requirements.

Caution: Regardless of which mode the I/O COM Server port is configured in, the address always consists of the **port number of the port** (Menu: SETUP Port..→ TCP/IP Mode → Port Number) and the **IP address of the I/O COM Server** (Menu: SETUP System → Setup TCP/IP → IP-Address) .

The port number range is freely selectable between 49152 - 65535 (hex: 0xC000 - 0xFFFF) (see section 2). The factory setting is 49153 (0xC001) for Port 1, 49154 (0xC002) for Port 2, etc.

The I/O COM Server provides both possible socket-level protocols for sending data: Client/server with TCP sockets (streams) or UDP sockets (datagrams). The following comparison shows the essential features of both protocols at a glance:

Feature	UDP Protocol	TCP Protocol
Data integrity (checksum)	yes	yes
Connection control using the TCP/IP stack (opening and closing the connection)	no	yes
Guarantee of correct packet sequence using TCP/IP stack	no, may be implemented separately	yes
Limit to number of simultaneously active connections on the computer	no (multiple I/O cards can be polled with one socket)	One socket required per connection to an I/O card
Number of user programs which can access an I/O card at the same time	unlimited	1
Number of user programs which can receive alarms from a card driver.	1	1

TCP protocol is connection-based, i.e. during data transmission there is a fixed connection between client and server. TCP has all the mechanisms for opening a connection and closing it as well as ensuring error-free data transfer over the network.

UDP has none of these mechanisms. It can however be faster in some cases, since there is no time required for connection building and breaking and there can be no timeout situations. This means if a packet gets lost, data transmission continues uninterrupted unless a higher level protocol is taking care of resends.

4.1.1 "TCP Server" Mode (Standard Mode)

The standard configuration for the I/O COM Server is TCP server mode. If this is the mode you wish to use, you need to make no additional settings. A connection can be established with the COM server port from any TCP/IP station in the network. If the port is in use by a client process, the connection request is rejected until the active connection is closed.

The connection is controlled by the user program (client process), i.e. the user program opens and closes the connection.

Note: *In connectionless state the entry "FREE" must be visible in the COM server port menu "SETUP Port.. P Port State P Connection State", and "In Use ... " if there is an active connection !*

4.1.2 "TCP Client" Mode (Menu: TCP Client)

The I/O COM Server port is configured as a TCP client in the submenu "TCP Client" (data transport using streams).

In this mode the I/O COM Server is able depending on the input register to open a connection to the configured TCP server and after a timeout has expired to close it again. If there is no connection to the TCP server currently active, it can also be the TCP server itself and accept connections.

Menu settings: SETUP Port..→ TCP/IP Mode → TCP Client

→ **Server Port**

Port number which the user program (TCP server process) addresses.

Format: decimal

→ **Server IP**

IP address of the computer on which your user program (server process) is active.

Format: Dot-Notation

→ **Special Options P Inactivity Timeout (default: 30)**

Configures the timer for closing the I/O COM Server port. The value is decimal and given in seconds. The timer is reset for an active network connection when data are exchanged.

1 Tick: 1 second

Format: decimal

Inactivity Timeout = 0 deactivates the connection after a timeout.

→ **Special Options P Connection Timeout (default: 300)**

This value is also a connection timeout. It only becomes active if there is no activity between server and cli→ Special Options → INPUT-Mask E0..11 (default: E0 to E11 = 1)

Here is where the inputs are configured which are polled by the card driver. If there is a level change at one or more inputs, the TCP connection for the configured TCP server process (application program) is opened and the I/O structure "Write Register" sent.

Format: bit-by-bit comma separated (0,0,0....)

E0 to E11 = 1 activates all inputs

Caution: The parameter "INPUT Mask E0..11" must be a number other than zero, otherwise there is no condition for sending a packet!

→ Special Options → INPUT-Mask E0..11 (default: E0 to E11 = 1)

Here is where the inputs are configured which are polled by the card driver. If there is a level change at one or more inputs, the TCP connection for the configured TCP server process (application program) is opened and the I/O structure "Write Register" sent.

Format: bit-by-bit comma separated (0,0,0....)

E0 to E11 = 1 activates all inputs

Note: The parameter "INPUT Mask E0..11" must be a number other than zero, otherwise there is no condition for sending a packet!

After you have entered all the parameters in the COM Server port menu, repeatedly press ENTER to return to the main menu and save the entries using **SAVE Setup**. The message "**Saving...**" appears, after which the "TCP Client" mode is activated. In the menu "SETUP Port..P Port State P Connection State" the entry "TCP-Client" appears. The current status of the connection can be read in this menu whenever desired.

Deactivating "TCP Client" mode

In the "SETUP Port..P TCP/IP Mode P TCP Client" menu set the entry "Server Port" to zero or use the menu "SETUP Port..P Port State P Clear Port Mode".

4.1.3 "UDP-Client" Mode (Menu: UDP Client)

The name "UDP Client" is somewhat misleading, since there are actually no mechanisms in UDP protocol for opening or closing a connection. Whether a process behaves like a server or a client is determined by the layer above the UDP (Application Layer).

Note: An I/O COM Server port must in either case (client or server) be configured for data exchange using UDP datagrams in order to ensure clear classification for independent alarm and data transmissions. Essentially all UDP datagrams which are addressed to the respective port are accepted. Alarms and status changes in the input register on the other hand are only sent to the configured application (UDP server process).

Menu settings: SETUP Port..→ TCP/IP Mode → UDP Client

→ Server Port

Port number which the user program (UDP server process) on the computer addresses.

→ Server IP

IP address of the computer on which the user program is active.

→ Special Options → Packet Protocol (default: 0)

This switch activates (1) or deactivates (0) the control procedure of the packet sequence in UDP mode (see section 5.2).

→ Special Options → INPUT-Mask E0..11 (default: E0 to E11 = 1)

This menu is used for specifying the inputs which are to be monitored for a change in level of the input signal. If the signal changes at one of the "activated" inputs, a data gram with the I/O structure "Write Register" is sent to the configured user program (UDP server process).

Format: bit-by-bit comma delineated (0,0,0,...)

E0 to E11 = 1 activates all inputs

E0 to E11 = 0 deactivates the mode

→ Special Options → Send Interval (default: 0)

Configurable interval with a 100ms basis at which the card driver sends the I/O structure "Write Register" with the contents of the input register to the configured server process. The value is given in decimal format. E0 to E11 = 0 deactivates the mode.

1 Tick: 100ms

Format: decimal

After you have entered all the parameters in the COM Server port menu, repeatedly press ENTER to return to the main menu and save the entries using **SAVE Setup**. The message "Saving..." appears, after which "UDP-Client" mode is activated. The message "UDP-Client" appears in the "SETUP Port..→ Port State → Connection State" menu. The current connection status can always be read in this menu.

Deactivating "UDP-Client" Mode

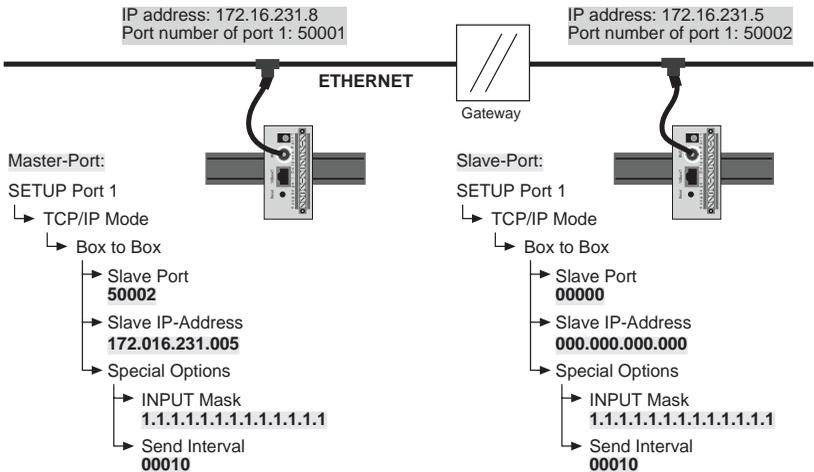
Open the "SETUP Port..P TCP/IP Mode P UDP Client" menu and set the entry "Server Port" to zero or use the "SETUP Port..P Port State P Clear Port Mode" menu.

4.1.4 "Box to Box" Mode (Menu: Box to Box)

"Box to Box" mode allows any two I/O COM Server ports to be logically connected through the network without an additional control computer. In this mode the inputs on both I/O cards are in constant online contact with the outputs of the other ports. Any additional data transmission or other network protocols have no effect on the connection.

In "Box to Box" mode a permanent TCP connection is established between the Master port and the Slave port. It makes no difference which port is the Master and which the Slave. The Master port functions like a TCP client and is thereby responsible for opening (after configuration or reset) and closing (after cancelling "Box to Box" mode).

Example configuration for "Box to Box" Mode:



The Slave address is only configured on one port. The conditions for data exchange are set however at both ports. Both send when the status at an input changes or every second (10*100ms).

Menu settings: SETUP Port..→ TCP/IP Mode → Box to Box

Note: *In Box to Box mode only the Master port is configured. "Slave IP-Address" and "Slave Port Number" are configured **only on the Master port!** Settings in the "Special Options" submenu can be made on both ports.*

→ **Slave Port** (set for Master port only)

Port Number of the I/O COM Server Slave port.

→ **Slave IP-Address** (set for Master port only)

IP address of the I/O COM Server in which the Slave port is located.

→ **Special Options → INPUT-Mask E0..11 (default: E0 to E11 = 1)**

This mask indicates at which I/O card inputs a level change generates a data transmission to the other port. If the level at one of the "activated" inputs changes, the entire input register is sent over the network and mapped in the output register of the other port.

Note: *The precise time space between level changes cannot be mapped and changes which take place at intervals shorter than 20ms cannot be captured individually. This would create an enormous network load on the Ethernet.*

→ **Send Interval (default: 0)**

Configurable interval with a 100ms time base during which the card driver maps the input register to the output register of the other I/O card. Send Interval = 0 deactivates the mode.

1 Tick: 100ms

Format: decimal

Send Interval = 0 deactivates sending at defined intervals.

Note: *One of the two parameters INPUT-Mask and Send Interval must be greater than ZERO, otherwise there is no condition for data exchange.*

Deactivating "Box to Box" Mode

Just as in the case of configuration, "Box to Box" mode is deactivated only on the Master port. The I/O COM Server must also be on the Ethernet so that the Master port can properly close the TCP connection.

Set "Slave Port Number" in the "SETUP Port..→ TCP/IP Mode → Box to Box" menu to zero or use the "SETUP Port..→ Port State → Clear Port Mode" menu.

If for some reason the Slave port is not released, you can also use the "SETUP Port..→ Port State → Clear Port Mode" command in the Slave port menu.

4.1.5 Menu: System Options

The parameters configurable in this menu branch are always valid regardless of which mode the COM Server port is in.

→ Driver Watchdog (*100ms) (default: 0)

This parameter allows monitoring of all entities which are active between the card driver in the COM server and the control program, including the card driver itself (e.g. TCP/IP stack in the computer, network card, network itself, ...)> If one of these entities fails and communication between the control program and the card driver is no longer ensured, the card driver turn all the outputs off after the Driver Watchdog time has expired.

*Note: If this parameter is activated the control program must regularly poll the card driver by sending an I/O structure (e.g. READ_REG). The value of the "Driver Watchdog" parameter should be a multiple of the polling rate. Monitoring is **always** active. If a TCP connection is closed without deactivating the outputs, this will be done after the Driver Watchdog time has elapsed.*

1 Tick: 100ms

Format: decimal

Driver Watchdog = 0 deactivates turning off the outputs after a timeout.

4.2 Menu: SETUP Port .. → Port State

This menu shows you information about the configured TCP/IP mode of the I/O COM Server port, the status of the network connection, and a listing of any errors which may have occurred.

→ Error State

This list shows error which have occurred on this port. If more than 5 errors have occurred since the last restart of the COM server or the last time the error table was deleted, the oldest entry is always overwritten. The most recent entry is always in first position, and the oldest in last position. Before each entry is the time of the event in hours and minutes since the last restart of the COM server. You can delete the error table in order to have all entries available again.

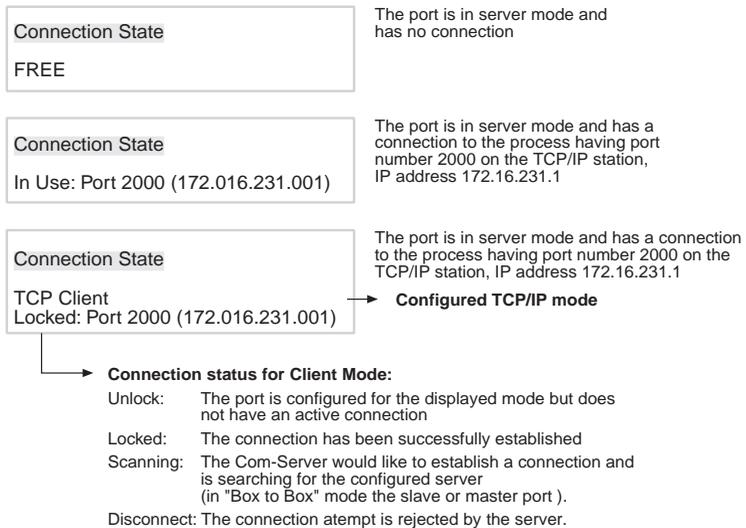
→ Clear Port Mode

This command represents a simple and easy way to return the port to standard TCP/IP mode. The configured TCP/IP mode (Socket Client, UDP-Mode, Box to Box, ...) is cancelled.

*Note: The changes made by "Clear Port Mode" are still in effect after exiting the menu **without SAVE Setup**, i.e. they are backed up directly in a non-volatile memory (EEPROM).*

→ **Connection State**

This menu allows online connection monitoring. The display is structured as follows:



The display is updated by exiting and then reselecting the "Connection State" menu.

5. Controlling the Digital I/O Card using TCP/IP

To enable simple communication between the user program on the computer and the card driver in the I/O COM Server there are a number of structures which define the format and the contents of the data exchanged between the user program and card driver. These I/O structures allow you to exchange pure data, i.e. the contents of the input and output registers of the I/O cards, as well as to configure the various methods of automatic messaging (for models 50211, 50311 and 50611 including alarm and monitoring modes). actual handling of the I/O cards, such as polling the registers, status monitoring etc., is taken over by the card driver in the I/O COM Server.

The user program uses the convenient socket interface (Windows: WinSock, UNIX: Berkley Sockets) for exchanging data in the form of these I/O structures with the card driver in the I/O COM Server over the network via TCP/IP.

The I/O structures are independent of the selected network protocol (TCP or UDP).

Socket Interface		I/O Structures	
		UDP-/TCP-Header	UDP-/TCP-Datagram
	IP-Header	IP-Datagram	
Ethernet-Header	Ethernet-Datagram		

For each digital I/O card present in the I/O COM Server a TCP or UDP connection must be established and configured in the I/O COM Server (see section 4) depending on the selected mode (Client or Server). The type of application determines which of the two protocols - UDP or TCP - should be selected. Both protocols have their pros and cons which you must consider depending on the application (see section 4.1).

Note: For help in socket programming including the basics of TCP/IP, see our short and concise manual "Ready in 1 Day for TCP/IP Sockets". Program examples for client and server applications under TCP/IP can be found in the download area of our Homepage at <http://www.wut.de>.

5.1 Definition of I/O Structures

In order to be able to clearly identify and process the contents of a packet, all the data must be sent to the I/O COM Server in the form of these I/O structures. All structures begin with the same header, which consists of the following 4 WORDs (16bit_Integer).

send_sequence and *rec_sequence* are only filled in if you want to implement packet sequence control in UDP mode yourself (see section 5.2). Otherwise these values are always 0 and may not be omitted!

The value *struct_typ* identifies the contents of the structure after the first 4 WORDs and *length* indicates the total length of the structure in bytes, i.e. including the first 4 WORDs. The following general packet structure results:

```
typedef struct _EA_DRIVER
{
    word send_sequence;    // UDP: Transmit-Packetcounter or 0, TCP: 0
    word rec_sequence;    // UDP: Receive-Packetcounter or 0, TCP: 0
    word struct_typ;      // Structure type name
    word length;          // Packet length in Bytes (8 + sizeof(name))

    typedef struct _name
    {
        ...                // Structure corresponding to
                           // struct_typ
    } name ;              // variable part, see next
                           // section
} EA_DRIVER;
```

Caution: The following applies to all I/O structures:

A **word** corresponds to one 16bit_Integer.

A **char** corresponds to one byte (8bit).

Hexadecimal notation: **0x** in front of the value.

The definition of the structures (also called Type Definition) and the examples are shown in "C" programming language. In the following sections the individual structures which are used for *name* are declared and explained and the corresponding values of the variables *send_sequence*, *rec_sequence*, *struct_typ* and *length*, with which each packet begins, will be indicated.

5.1.1 Writing and reading the I/O registers

The 12 digital in- and outputs are represented respectively in a WORD, i.e. Bit 0 corresponds for example to Input or Output 0 (see section 3.1.1).

The I/O Structure Read Register

Sending this structure to the I/O card driver causes the latter to read the input register and send it to the user program. The packet consists only of these four WORDs. This structure is used only by the user program, and the I/O card driver always responds by sending the structure WRITE_REG.

```
send_sequence   = 0;           // UDP-Protocol see section 5.2
rec_sequence    = 0;           // UDP-Protocol see section 5.2
struct_typ     = 0x0001;      // Structure type READ_REG
length         = 8;           // Packet length in Bytes
```

The I/O Structure Write Register

This structure is used to send the contents of the input or output register. If the user program sends this structure to the I/O card driver, the value *word_anz* must always be 1. The I/O card driver writes the value register[0] to the output register of the I/O card.

If the card driver sends this structure to the user program, register[0] contains the value of the input register.

```
send_sequence   = 0;           // UDP-Protocol see section 5.2
rec_sequence    = 0;           // UDP-Protocol see section 5.2
struct_typ     = 0x0008;      // Structure type WRITE_REG
length         = 10 + 2*word_anz; // Packet length in Bytes
```

```
typedef struct _WRITE_REG
{
    word word_anz;           // Number of 16-bit registers
    word register[word_anz]; // Array of registers
} WRITE_REG;
```

Depending on the "Send Mode" of the I/O card driver, several register contents can be send in this structure. If for example you want to send each status change in the input register, these can take place faster than network packsts can be generated. In this case all the register contents are contained in chronological order in one packet.

The I/O Structure Set Bit

This structure lets you set individual bits in tghе output register. If for example less than the entire process status is mapped in the user program, individual I/O bits can be set without changing the value of the others. The bis 0...11 in *set_bits* and *value* correspond to the corresponding digital outputs. This structure is used only by the user program.

```
send_sequence= 0;           // UDP-Protocol see section 5.2
rec_sequence  = 0;           // UDP-Protocol see section 5.2
struct_typ   = 0x0009;      // Structure type SET_BIT
    length      = 12;        // Packet length in Bytes

typedef struct _SET_BIT
{   word set_bits;           // Bit = 1:           | Bitposition = 0:
    // Output is set         | is not set
    word value;             // Level of bits to be set
}   SET_BIT;
```

Example:

```
set_bits = 0x0124 / value = 0x0104
```

The digital outputs A2 and A8 (numbering from A0...11) are set to HIGH and output A5 to LOW. All other outputs remain unchanged.

The I/O Structure Send Mode

This structure is used to specify the trigger conditions for the I/O card driver to send the contents of the input register to the user program. There are essentially three possibilities, all of which can be combined:

1. The user program polls the input register by sending the READ structure
2. The I/O card driver sends the WRITE structure with the contents of the input register at a configurable interval, or
3. after a state change of the configured inputs

```
send_sequence= 0;           // UDP-Protocol see section 5.2
rec_sequence  = 0;           // UDP-Protocol see section 5.2
struct_typ   = 0x0010;      // Structure type SEND_MODE
    length      = 12;        // Packet length in Bytes

typedef struct _SEND_MODE
{   word EA_bits;           // Bit mask of the inputs which are to be
    // .. tested for a state change
    word interval;         // Interval for data update, Factor: 100ms
    // .. 0=deactive, 1=100ms, 2=200ms, etc.
}   SEND_MODE;
```

The I/O Structure Register Request

The user program sends this structure to the I/O card driver in order to be able to read (overview) the contents of input and output registers. The I/O card driver always responds with the I/O structure REG_STATE.

```
send_sequence= 0;           // UDP-Protocol see section 5.2
rec_sequence  = 0;           // UDP-Protocol see section 5.2
struct_typ   = 0x0021;      // Structure type REG_REQUEST
length       = 8;           // Packet length in Bytes
```

The I/O Structure Register State

The I/O card driver uses this structure to send the contents of the input and output register. This structure is only sent if the user program has sent the Register Request structure to the card driver.

```
send_sequence= 0;           // UDP-Protocol see section 5.2
rec_sequence  = 0;           // UDP-Protocol see section 5.2
struct_typ   = 0x0031;      // Structure type REG_STATE
length       = 14;          // Packet length in Bytes

typedef struct _REG_STATE
{
    word Driver_id;          // Card driver ID
    word INPUT_REG;          // Input register E0...11 ,1 = HIGH, 0 = LOW
    word OUTPUT_REG;        // Output register A0...11 ,1 = HIGH, 0 = LOW
} REG_STATE;
```

Variable	
Driver_ID	For digital I/O card drivers with the measuring function (#'s 50211, 50311, 50611) this value is always 1. For digital I/O card driver without measuring function (#'s 50210, 50310) this value is always 2.
INPUT_REG	Contents of the input register, e.g. IBit1 = 1 -> Input E1 = HIGH
OUTPUT_REG	Contents of the output register, e.g. BIt3 = 1 -> Output A3 = HIGH

5.1.2 Status check of the I/O Card (Models 50211, 50311, 50611 only)

The I/O structures described in this section are only supported by the I/O COM Servers which have an integrated measuring and monitoring unit. These are models 50211, 50311 and 50611.

On one hand the user program can specifically read the status of the I/O card. On the other hand it is possible to configure error states which, when they occur, the user program will be automatically notified by the I/O card driver (see section 5.1.3).

The I/O Structure Status Request

The user program sends this structure to the I/O card driver in order to request status information. The I/O card driver always responds with the I/O structure DRV_STATE.

```
send_sequence = 0;           // UDP-Protocol see section 5.2
rec_sequence  = 0;           // UDP-Protocol see section 5.2
struct_typ    = 0x0020;     // Structure type STATE_REQUEST
length        = 8;          // Packet length in Bytes
```

The I/O Structure Driver State

The I/O card driver fills this structure with all the information which is available to this digital I/O card. This structure is only sent if the user program has sent the Status Request structure to the card driver.

```
send_sequence = 0;           // UDP-Protocol see section 5.2
rec_sequence  = 0;           // UDP-Protocol see section 5.2
struct_typ    = 0x0030;     // Structure type DRV_STATE
length        = 70;         // Packet length in Bytes

typedef struct _DRV_STATE
{
    word Driver_id;          // Card driver ID
    word INPUT_REG;         // Input regs E0...11 ,1 = HIGH, 0 = LOW
    word OUTPUT_REG;        // Output regs A0...11 ,1 = HIGH, 0 = LOW
    word RESET;             // the I/O card was reset
    char Temperature;       // Temperature in housing
    char VCC;               // supply voltage present
    word Short_circuit;     // Short circuit-Bits A0..11, Bit=1:
                            // Current>0.5A
    word AD_lim_exceed;     // A/D Limit exceed A0...11
    char AD_mess[12];       // current A/D measured value A0...11
    char AD_low_limit[12];  // A/D low limits A0...11
    char AD_high_limit[12]; // A/D high limits A0...11
} DRV_STATE;
```

Variable				
Driver_ID	For digital I/O card drivers with measuring function (#'s 50211, 50311, 50611) this value is always 1.			
INPUT_REG	Contents of the input register, e.g. Bit1 = 1 -> Input E1 = HIGH			
OUTPUT_REG	Contents of the output register, e.g. Bit3 = 1 -> A3 = HIGH			
RESET	If RESET is not zero, the I/O card was reset. The card driver configures the last status before the reset.			
Short circuit	If this value is not zero, the I/O card driver has turned off the outputs because one or more outputs exceeded the current limit of 0.5A (per output). The set bits correspond to the associated outputs.			
AD_lim_exceed	The set bits correspond to the outputs where the current value was outside the set limit.			
	Range	Resolution	Units	Conversion factor
Temperature	-10...100	1°C	CC	-
If the temperature value at the warmest location of the housing interior. The difference between outside and housing interior temperature may be up to 36° C.				
VCC	0...60	1V	V	-
Applied supply voltage in Volts				
AD_meas[12]	0...255	3.3 mA	mA	*3.3
Measured current values for the 12 outputs. If the corresponding bit in AD_lim_exceed is set, AD_meas[] contains the value of the current over- or undervalue, otherwise the actual measured value. The value AD_meas must be multiplied by 3.3 in order to obtain the current in mA.				
AD_low_limit[12] AD_high_limit[1-2]	0...15	30 mA	mA	*30
Indicates the current limits configured for each output. If AD_low_limit = 0 or AD_high_limit = 15, no comparison is made with the respective limit. Multiply these values by 30 to obtain the corresponding current value in mA.				

5.1.3 Alarms (Models 50211, 50311, 50611 only)

The I/O card driver supports alarming the user program when limit values on the I/O card are not reached or exceeded. Default is for alarm mode deactivated. The user program can configure the events which it wants to be informed about by using the SET_ALARM structure. If such an event occurs, the card driver sends the structure ALARM_STATE. Not until the user program has cleared the alarm using the CLEAR_ALARM structure and the event reoccurs on the I/O card does the card driver generate the next alarm.

The I/O Structure Set Channel Alarm

The control program can use this structure to specify which events it wants to be informed about by the card driver.

```

send_sequence = 0;           // UDP-Protocol see section 5.2
rec_sequence  = 0;           // UDP-Protocol see section 5.2
struct_typ    = 0x0080;     // Structure type SET_ALARM
length        = 40;         // Packet length in Bytes

typedef struct _SET_ALARM    // Alarm if:
{
  word RESET;                // - I/O card was reset
  char Temperature;          // - Temperature < 0 or > 60 degrees
  char VCC;                  // - VCC < 9V or > 31V
  word Short_circuit;        // - Turn off outputs A0..11
  word AD_lim_exceed;         // - Current exceeded A0..11
  char AD_low_limit[12];      // - Low Limit not reached
                                // (0..15, current value: *30mA)
  char AD_high_limit[12];     // - High Limit exceeded
                                // (0..15, current value: *30mA)
} SET_ALARM;

```

To activate the monitoring functions RESET, Temperature, VCC and Short Circuit, simply set to a value other than zero. To configure current limits for the individual channels, set the corresponding bit in *AD_lim_exceed* and enter the corresponding limits in *AD_low_limit* and/or *AD_high_limit*. The same value ranges and conversion factors apply as for the DRV_STATE structure.

Caution: Sending this structure also clears all active alarms at the same time, and so has the same effect as the CLEAR_ALARM structure.

Example 1:

You want an alarm when the output current on Output A5 exceeds 300mA and when the output current on A10 falls below 80mA or is greater than 200mA.

```
AD_lim_exceed= 0x0420;
AD_low_limit[5] = 0;          // No comparison
AD_high_limit[5] = 10;       // 300mA / 30mA = 10
AD_low_limit[10] = 3;        // 3 * 30mA = 90mA -> Alarm at < 90mA
AD_high_limit[10] = 7;       // 7 * 30mA = 210mA -> Alarm at > 210mA
```

Example 2:

You want an alarm when the output current on A11 falls below 30mA. The alarm for falling below should only be given if the output is also in fact activated (HIGH level).

```
AD_lim_exceed      = 0x0800;
AD_low_limit[5]    = 1;          // 300mA / 30mA = 10
AD_high_limit[5]   = 15;        // No comparison
```

The I/O Structure Alarm Status

The card driver uses this structure to notify the user program when one or more events has occurred which were configured with the SET_ALARM structure. The ALARM_STATE structure is similar to the SET_ALARM structure. All variables whose contents is not equal to 0 contain a valid alarm.

```
send_sequence= 0;          // UDP-Protocol see section 5.2
rec_sequence = 0;          // UDP-Protocol see section 5.2
struct_typ   = 0x0090;    // Structure type ALARM_STATE
length       = 28;        // Packet length in Bytes

typedef struct _ALARM_STATE
{
    word RESET;           // Digital I/O card was reset
    char Temperature;    // Temperature value < 0 or > 60 degrees
    char VCC;            // Supply voltage < 9V or > 31V
    word Short circuit;  // Output current longer than 250ms > 0.5A
    A0..11
    word AD_lim_exceed;  // Over/under current limit A0..11
    char AD_meas[12];    // AD values corresponding to AD_lim_exceed
} ALARM_STATE;
```

If the value "short circuit" is set, the entire structure is filled in even if there are values not outside the valid ranges. This allows for a complete process image at the time when the outputs were turned off.

The I/O Structure Clear Alarm

The alarm variables Temperature, VCC, Short Circuit and AD_lim_exceed require specific clearing of the alarm by the control program. These values represent long-lasting error states. In order to prevent alarm messages from being continuously sent over the network, a new alarm is sent only once the control program has cleared the alarm in the PC104 card driver.

As in the case of the SET_ALARM and ALARM_STATE structures, the corresponding bits and values must be set in order to clear the corresponding alarms.

```

send_sequence = 0;           // UDP-Protocol see section 5.2
rec_sequence = 0;           // UDP-Protocol see section 5.2
struct_typ    = 0x00A0;     // Structure type CLEAR_ALARM
length        = 14;         // Packet length in Bytes

typedef struct _CLEAR_ALARM
{
    char Temperature;        // != 0 -> clear Alarm
    char VCC;                // != 0 -> clear Alarm
    word short_circuit;      // != 0 -> Clear all outputs
    word AD_lim_exceed;      // Reactivate current control for the
} CLEAR_ALARM;              // .. set channels (A0..11)

```



Clearing the "Short Circuit" alarm causes the outputs to be turned on again. However, this is allowed to happen no sooner than 15 seconds after the alarm was generated. The reason why the outputs were turned off must be investigated.

Constantly turning the outputs on in error conditions may result in permanent damage to the unit!

5.2 The Acknowledge procedure for UDP optional)

To assure proper sequencing of the datagrams and provide a mechanism for ready confirmation when using UDP protocol, two additional values precede the actual I/O structures in addition to length and the structure type: *send_sequence* and *rec_sequence*.

Important:

Packet sequence control is only supported by the I/O COM Server if the option "Packet Protocol" is set to active in the "SETUP: Port .. -> Setup TCP/IP -> UDP-Mode -> Special Options" menu (see section ...).

If TCP protocol is used or no sequence monitoring is required, these values may be set to zero. They may not however be omitted!

Definitions:

An Acknowledge Packet

is a packet which only contains the first four words (*send_sequence*, *rec_sequence*, *struct_typ*, *length*) with the value *struct_typ* = 0.

A Synchronization Packet

is a packet which only contains the first four words (*send_sequence*, *rec_sequence*, *struct_typ*, *length*) with the value *struct_typ* = 0xFFFF.

A Data Packet

is a packet which contains an I/O structure having the value *struct_typ* != 0 and *struct_typ* != 0xFFFF.

The *send_sequence*

is the number of the last sent data packet and is incremented by one for each data packet.

The *rec_sequence*

is the number of the last received data packet.

Every data packet

requires an acknowledge packet with a *rec_sequence* identical to its own *send_sequence*.

Timeout ...

The next data packet may not be sent until the previous one was acknowledged. If there is no acknowledgement, the packet is repeated after a timeout (the timeout is doubled after each repetition up to some limit value). If the *rec_sequence* in a received packet does not agree with the actual *send_sequence*, the data packet with the last *send_sequence* is immediately repeated.

5.2.1 Synchronization of the sequence numbers

If the user program is started on the computer, it sends a UDP datagram containing the first four words of an I/O structure to the I/O COM Server port:

```
send_sequence = n;           // Enter desired send sequence
rec_sequence  = x;           // Undefined, determined by the COM-Server
struct_ttyp   = 0xFFFF;     // Synchronization packet
length        = 8;
```

The I/O COM Server accepts the send_sequence as a rec_sequence and enters its first send_sequence:

```
send_sequence = m;           // first send_sequence of COM Server port
rec_sequence  = n;           // accepted send_sequence of user program
struct_ttyp   = 0xFFFF;     // Synchronization packet
length        = 8;
```

Das Anwenderprogramm übernimmt nun seinerseits die send_sequence des E/A Com-Servers als rec_sequence. Somit ist jeweils die send_sequence der einen Seite mit der rec_sequence der anderen identisch.

Whenever the user program sends a synchronization packet, the sequences are synchronized all over again. The I/O COM Server always sends the sequences 0 after a reset, i.e. in "unsynchronized state".

5.2.2 Examples

The following page shows a schematic representation of the cases defined below:

Example 1 shows error-free exchange of data packets between the user program and the I/O card driver. The user program sends "Read Register". The card driver acknowledges receipt of the packet and sends the I/O structure "Write Register". The user program likewise acknowledges receipt of the data.

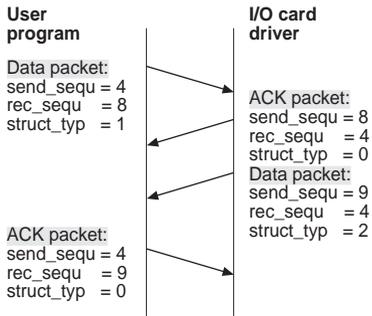
Example 2 is identical to the first, except that in this case the I/O card driver sends the acknowledgement of the Read Request and the I/O structure "Write Register" in a single packet.

Example 3 shows an error situation. The Read Request from the user program does not reach the I/O COM Server. After a timeout the user program repeats the packet since it has received no acknowledgement. The I/O card driver acknowledges receipt.

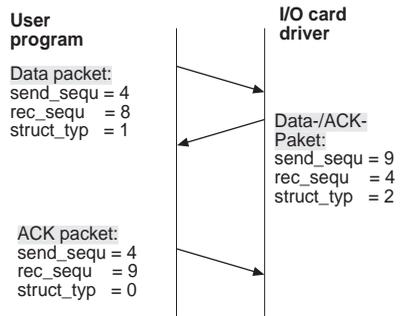
Example 4 shows a similar error situation. Before the timeout expires on the user program side, the I/O card driver sends the I/O structure "Write Register". The user program recognizes that the *rec_sequence* is not identical to the *send_sequence* and immediately repeats the last data packet. The acknowledgement for the packet received by the I/O card driver is already contained in the repetition.

Example 5 shows the error case in which the acknowledgement is lost. After a timeout the user program repeats the data packet. The I/O card driver recognizes that it has already received this data packet. It rejects the packet, but does acknowledge it again.

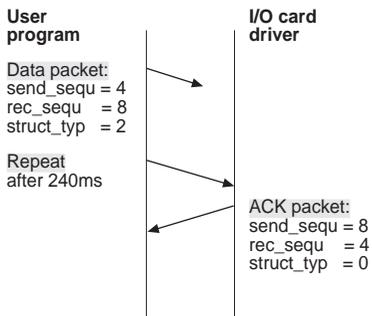
Example 1:



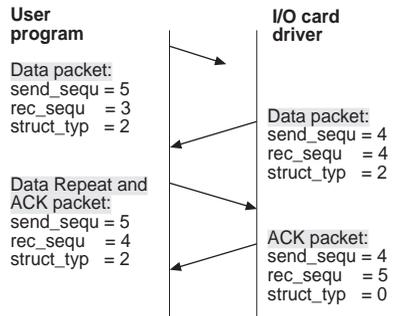
Example 2:



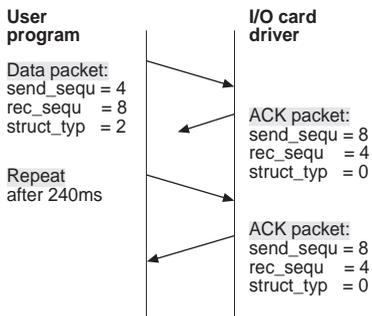
Example 3:



Example 4:



Example 5:



APPENDIX

Technical Data

Supply voltage:	at the screw terminals 24V and GND on the terminal strip	12V DC -10% to 24V DC +20% (3.4W)
Current draw: (Models: 50210, 50211 50310, 50311)	Supply voltage 24V Mains supply Coax-/TP Supply voltage 12V Mains supply Coax-/TP	140mA / 116mA 230mA / 180mA
with expansion box: (Model: 50611)	Supply voltage 24V Mains supply Coax-/TP Supply voltage 12V Mains supply Coax-/TP	240mA / 216mA 330mA / 380mA

**Current draw values above do not consider
switched output loads!**

Digital outputs:	1 module 12 x Digital Out: Output voltage: Output current: max. module current: Connection: Isolation outputs-to-network	10.8 to 28.8V DC 0.5A 6A 16x screw terminals 500V
Digital Inputs:	3 modules 4 x each Digital In: max. input voltage: Switching threshold: "On" Current: Connection: Isolation inputs-to-network Module galvanic isolation:	+/-30V DC 8V +/-1V 2.2mA 16x screw terminals 500V 500V, 1s
Measuring functions: (Models 50211, 50311, 50611 only)	Output current: Internal temperature: Supply voltage:	0 to 500mA 0°C to 100°C 0V to 60V
Safety functions: (Models 50211, 50311, 50611 only)	Current limiter: Turn-off delay: Peak load tolerant for Short-circuit protected	0.65A / bit 250ms t < 250ms

Serial Port: (Models 50310, 50311 only)	DB9 male: Baud rate: Data bits: Stop bits: Parity: Flow control:	PC-COM configured 300 to 57600 7 and 8 1 and 2 NO, EVEN, ODD NO, Hardware, Software (XON,XOFF)
---	---	--

Ambient temperature:	Storage: Operating:	-25°C to 70°C 0°C to 50°C
Housing:	Plastic housing for hat rail mount	100 x 100 x 50
Weight:	approx. 320g	

Network timing:

The times for I/O COM Server response over the network are typical values for a normally loaded network (<40% network load). The response times increase with network load as well as loading of the serial port in the I/O COM Server (Port 0 only for models 50310 and 50311) and depend on the I/O structure being requested. The following table shows the typical times for all I/O structures which the I/O COM Server sends.

E/A Structure Type	Event	Response time using UDP Protocol	Response time using TCP Protocol
WRITE_REG	after receiving the I/O structure READ_REG	4ms	6ms
DRV_STATE	After receiving the I/O structure STATE_REQUEST	47ms	55ms
ALARM_STATE	After the alarm occurs	42ms	50ms
ACK_PACKET (UDP Acknowledge only)	After receipt of the packet to be acknowledged	4ms	