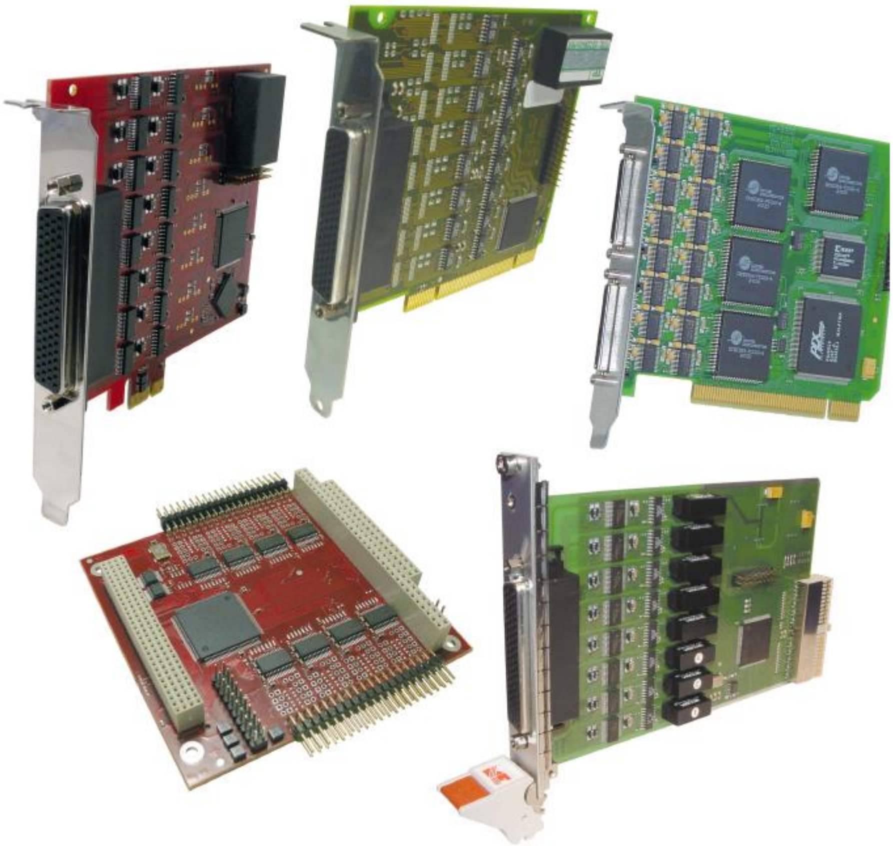
Meilhaus Electronic Manual

ME-90/9000/9100/9300 Series 3.0E

(PCI-, PCI-Express-, PC/104-Plus- and CompactPCI-Versions)



RS-232 or RS-422/485 Interface Boards with up to 16 Ports (optional: Opto-Isolation)

Imprint

Manual ME-9000 Series

Revision 3.0E

Revised: 2019-11-27

Meilhaus Electronic GmbH
Am Sonnenlicht 2
D-82239 Alling bei München
Germany
www.meilhaus.de

Important note:

The information contained in this manual has been reviewed with great care and is believed to be complete and accurate. Meilhaus Electronic assumes no responsibility for its use, any infringements of patents or other rights of third parties which may result from use of this manual or the product. Meilhaus Electronic assumes no responsibility for any problems or damage which may result from errors or omissions. Specifications and instructions are subject to change without notice.

Note the Meilhaus Electronic general terms of business:
www.meilhaus.de/en/infos/my-shop/tob/

Content

1 Introduction

Valued customer,

Thank you for purchasing this device from Meilhaus Electronic. You have chosen an innovative high-technology product that left our premises in a fully functional and new condition.

Please take the time to carefully examine the contents of the package for any loss or damage that may have occurred during shipping. If there are any items missing or if an item is damaged, please contact us immediately.

Before installing the board in your computer, we recommend you read this manual carefully, especially the chapter describing board installation.

The descriptions in this manual concern PCI-, PCI-Express- and CompactPCI-versions of the ME-9000 series, if not otherwise noted.

1.1 Important Notes

1.1.1 Use in Accordance with the Requirements

The PC boards of the ME-series are designed for acquisition and output of analog and digital signals with a PC. Depending on type install the models of the ME-series into:

a free PCI slot or

a free PCI Express slot or

a free CompactPCI slot.

The boards of the ME-90 PC/104-Plus series are serial interface boards (RS-232 or RS-422/485) in the PC/104 format to be integrated in a PC/104-Plus stack.

For information on how to install a plug-in board or connect a USB device, please read the manual of your PC.

Please note the instructions and specifications as presented in this manual (Appendix A, Specifications):

- Please ensure sufficient heat dissipation for the board within the PC housing.

- All unused inputs should be connected to the ground reference of the appropriate functional section. This avoids cross talk between the input lines.
- The opto-isolated inputs and outputs achieve an electrical isolation of the application relative to PC ground.
- Note that the computer must be powered up prior to connecting signals by the external wiring of the board.
- As a basic principle, all connections to the board should only be made or removed in a powered-down state of all components.
- Ensure that no static discharge occurs while handling the board or while connecting/disconnecting the external cable.
- Ensure that the connection cable is properly connected. It must be seated firmly on the D-Sub connector and must be tightened with both screws, otherwise proper operation of the board cannot be guaranteed.

1.1.2 Improper Application

PC plug-in boards for the PCI-, PCI-Express- or CompactPCI-bus may not be taken into operation outside of the PC. PC/104-Plus boards may not be taken into operation outside of an embedded system. Never connect the devices with voltage-carrying parts, especially not with mains voltage. As power supply of the USB models only an authorized power adapter may be used.

Make sure, that no contact with voltage-carrying parts can happen by the external wiring of the device. As a basic principle, all connections should only be made or removed in a powered-down state.

1.1.3 Unforeseeable Misapplications

The device is not suitable to be used as a children’s toy, in the household or under unfavourable environmental conditions (e.g. in the open). Appropriate precautions to avoid any unforeseeable misapplication must be taken by the user.

1.2 Package Contents

We take great care to ensure your delivery is complete. Nonetheless, please check the list enclosed to verify the contents of your delivery. You should find included:

- RS-232 resp. RS-422/485 interface board (depending on version ordered):
    - ME-90 PC/104-Plus or
    - ME-9000 cPCI, PCI or PCIe or
    - ME-9100 cPCI or PCI or
    - ME-9300 PCI
- Manual in PDF format on CD/DVD.
- Driver software on CD/DVD.
- ME-9000/9100: dual-, quad- or octopus-connection cable from 78-pin D-Sub male connector to 2, 4 resp. 8 x 9-pin D-Sub male connector, 1 m
- ME-9000 PCI/PCIe/cPCI: Additional mounting bracket for DIO port with flat ribbon cable to 25-pin D-Sub female connector for PCI-/PCIe- (ME-AK-D25F/S) resp. CompactPCI-slot.
- ME-90 PC/104-Plus: Flat ribbon cable for DIO port to 25-pin D-Sub female connector (ME-AK-D25F)
- ME-90 PC/104-Plus: 2 x flat ribbon cable with each 4 x 9-pin D-Sub male connector (2 x ME-AK-4D9M).

More connectivity option see page 47.

1.3 Features

ME-9000 Models	Serial Ports
ME-9000(i/p)*/2 RS232 PCI/PCIe/cPCI	2 RS-232 ports up to 921,6 kBd
ME-9000(i/p)*/4 RS232 PCI/PCIe/cPCI	4 RS-232 ports up to 921,6 kBd
ME-9000(i/p)*/8 RS232 PCI/PCIe/cPCI	8 RS-232 ports up to 921,6 kBd
ME-9000(i/p)*/2 RS485 PCI/PCIe/cPCI	2 RS-485 ports up to 921,6 kBd
ME-9000(i/p)*/4 RS485 PCI/PCIe/cPCI	4 RS-485 ports up to 921,6 kBd

ME-9000(i/p)*/8 RS485 PCI/PCIe/cPCI	8 RS-485 ports up to 921,6 kBd
ME-9000(i/p)*/2 MIX PCI/PCIe/cPCI	1 RS-232 port and 1 RS-485 ports up to 921,6 kBd
ME-9000(i/p)*/8 MIX PCI/PCIe/cPCI	4 RS-232 ports and 4 RS-485 ports up to 921,6 kBd

Table 1: Model overview ME-9000

* The PCI-Express models, the ME-9000 PCI models (from Rev. 1.4 up) and the ME-9000 CompactPCI models (from Rev. 1.2 up) are provided with optional opto-isolation from PC ground („i“-models) resp. with electrical insulation between the single ports („p“-models).

Note: Further versions on request (sales@meilhaus.com).

The **ME-9000** series is a high integrated serial interface board designed for PCI-, PCI-Express- resp. CompactPCI-bus. It provides a PCI controller with integrated Octo-UART of type EXAR XR17D158IV for extended temperature range. The UARTs are register-compatible with the 16550 and provide an integrated 64-byte transmit and receive FIFO for each port. The board allows transfer rates up to 921,6 kBd. There are versions available with up to 8 ports for RS-232 and/or RS-422/485 standard. The ME-9000 PCI- and cPCI-versions adapt themselves automatically to the level at the PCI bus of +3.3 V or +5 V (Universal PCI).

As a special the ME-9000 offers an 8-bit multi-I/O port for customer-specific extensions. See also chapter 4.1 "ME-9000 Multi-I/O Driver" from page 61 on.

The **ME-9000i** models provide opto-isolation of all serial ports from PC ground. I.e. as seen from the application all ports refer to a common, isolated ground (GND_C). The isolation voltage is 500 V.

ME-9000p models: each serial port is an electrically isolated „island“ with separate ground (GND_x). It is ideal for noise-sensitive environments in industrial control. The isolation voltage is 500 V.

ME-9100 Models	Serial Ports
ME-9100i/4 RS232 PCI/cPCI	4 RS-232 ports up to 921,6 kBd with opto-isolation
ME-9100i/8 RS232 PCI/cPCI	8 RS-232 ports up to 921,6 kBd with opto-isolation
ME-9100i/4 RS485 PCI/cPCI	4 RS-485 ports up to 921,6 kBd with opto-isolation
ME-9100i/8 RS485 PCI/cPCI	8 RS-485 ports up to 921,6 kBd with opto-isolation

Table 2: Model overview ME-9100

The **ME-9100i** is a high-speed serial interface board designed for PCI- resp. CompactPCI-bus. There are versions which are available with 4 or 8 ports for RS-232 or RS-422/485 standard. The UARTs are 16550 compatible and are good for transfer rates up to 921,6 kBd. Each port has an integrated 128-byte transmit and receive FIFO. The ports are opto-isolated by default („i“-versions).

ME-9300 Models	Serial Ports
ME-9300/16 RS232 PCI	16 RS-232 ports up to 921,6 kBd

Table 3: Model overview ME-9300

The **ME-9300** is a multi-port RS-232 interface board designed for the PCI bus. The board is available with 16 ports and is good for transfer rates up to 921,6 kBd. The UARTs are 16550 compatible and each port provides an integrated 128-byte transmit and receive FIFO. Connecting the serial ports we recommend fitting connection panels, which are available in several versions (see chap. 3.6Connection Options).

ME-90 PC/104-Plus Models	Serial Ports
ME-90/8 RS232 PC/104-Plus	8 RS-232 ports up to 921,6 kBd
ME-90/8 RS485 PC/104-Plus	8 RS-422/485 ports up to 21,6 kBd
ME-90/8 MIX PC/104-Plus	4 RS-232 Ports and 4 RS-422/485 ports up to 921,6 kBd

Table 4: Model overview ME-90 PC/104-Plus

Note: Further versions on request (sales@meilhaus.com).

The **ME-90 PC/104-Plus series** is a serial interface board compliant to PC/104-Plus specification 2.3 with PCI interface (3.3 V or 5 V) and a „looped through" ISA bus. I.e. if you have a PC/104-Plus single board computer, which supports PCI- as well as ISA-bus you can combine the ME-90 PC/104-Plus with ISA-based PC/104 boards without problems.

The **ME-90 PC/104-Plus** provides a PCI controller with integrated Octo-UART of type EXAR XR17D158IV for extended temperature range. The UARTs are register compatible with the 16550 and provide an integrated 64-byte transmit and receive FIFO for each port. The board allows transfer rates up to 921,6 kBd. There are versions available either with 8 RS-232, 8 RS-422/485 or mixed with 4 RS-232 and 4 RS-422/485 ports. The routing of the PCI-bus signals CLK, IDSEL and INT0# is done by jumpers. The ME-90 PC/104-Plus adapts itself automatically to the level at the PCI-bus of +3.3 V or +5 V (Universal PCI).

With its **extended temperature range** from -40 °C to +85 °C it is the ideal solution for industrial applications.

As a special the ME-90 PC/104-Plus offers an 8-bit multi-I/O port for customer-specific extensions. See also chapter 4.1 "ME-9000 Multi-I/O Driver" from page 61 on.

1.4 System Requirements

The ME-9000/9100/9300 can be installed into any PC with an Intel® Pentium® or compatible computer with a free standard PCI-, PCI-Express- or CompactPCI-slot (depends on model).

The ME-90 PC/104-Plus can be used as a PCI-based peripheral board in a PC/104 stack.

The board is supported by the Meilhaus Electronic Intelligent Driver System (ME-iDS).

1.5 Available Software

The provided software enables quick integration of the boards under all common operating systems. Using Windows the ports can be accessed as standard COMports.

Note: The ME-9000 and ME-90 PC/104-Plus are not supported under Windows 95/98/Me!

System Drivers: Current driver support see README-files.

High-level language support: Each programming language that supports access to Win32Com-API (e.g. Visual C, Delphi)

Graphical programming languages, i.e.: HP VEE, HP VEE Lab, Agilent VEE Pro, Agilent VEE OneLab, LabView™: no extra driver necessary; ports can be accessed as COM ports under Windows.

For the newest versions and latest software releases, please consult the README-files included with the driver software.

.

2 Installation

Important Note:

If you have got the driver software as an archive file (e.g. by download or the Windows 9x driver coming with CD/DVD), please unpack the software **before installing** the board to a directory of your choice.

Now insert the board into your computer (see chapter „Hardware Installation) and then install the driver software (see chapter „Driver Installation“). This order of operation is important to guarantee the “Plug&Play”-operation under Windows* 95/98/Me/2000/ XP and Vista. Windows NT 4.0 needs an analogous order of operation, however the installation procedure differs slightly.

**If the Windows version is supported by the appropriate board type (see README-files).*

2.1 Hardware Installation

The following chapter also applies to the installation of PC/104 (-Plus) boards.

Please read your computer’s manual instructions on how to install new hardware components.

Basically use the following procedure for installing the board.

- Make sure that the computer is turned off.

Caution: some of the more sensitive components can be damaged by static electricity!

> **That’s why:** Make sure to ground yourself by touching an exposed metal part of the PC case before handling the board.

- Unplug the power cable from your computer.
- Open the computer case.

Pick up your board carefully with both hands. Be careful **not to bend** the board or to damage the edges in any way. This could lead to short-circuits on the board. Do not exert too much pressure when inserting the board into the slot. A small amount of force should be all that is required to seat the board fully and properly into the slot.

- Close the computer case.
- Connect the power cable to your computer.
- Power up your computer and continue with the chapter “Driver Installation”.

2.2 Driver Installation

2.2.1 Unpacking the Driver Software

Proceed with the following steps if you have got the driver update as a self-extracting archive (e.g. by download or the Windows 9x driver coming with the CD/DVD, if supported by the regarding model). Else you can start directly with the driver installation (see the following chapters).

- Navigate to the appropriate archive file (e.g. `<Drive>:\InstallWindows\Serial\me-9x00\win9x\me9x00vxd.exe)` and unpack the driver software by double-clicking the archive file.
- Enable the option „Overwrite Files Without Prompting“ (default)
- Only Windows NT 4.0: enable the option „When Done Unzipping Run: setup.exe“
- Choose a directory and click on „Unzip“. By default the directory `C:\Meilhaus\ME-9x00\install` will be used.
    - o The driver software will be unpacked.
- Click on “Close”.

Continue with the following chapters.

2.2.2 Installation under Windows* 95/98/Me/2000/XP

**If the Windows version is supported by the appropriate board type (see README-files)*

! If you have got the driver as a self-extracting archive (e.g. by download or the Windows 9x driver coming with the CD/ DVD) you must unpack the driver software first. See chap.2.2.1 "Unpacking the Driver Software".

After inserting the board and rebooting, it will be detected automatically by the Windows „Hardware Wizard“ and the message „New Hardware Found“ will be displayed. The dialogs may differ slightly depending on your Windows version.

Note for Windows Vista: For safety reasons the user account control of Windows Vista requires your confirmation to proceed with the installation for several times. If you don’t have the appropriate rights contact your system administrator.

- With the hardware wizard select the option „Specify a location:“ and click the button Browser.
- Select the source for the installation of the driver software and start the installation by OK.
    - The driver will be installed.
    - The currently installed board will be added to the device manager under „Multi-function Adapter“ (Windows 95/98/ Me) resp. „Multi-port Serial Adapter“ (Windows 2000/XP).
    - The ports of the board will be added to the device manager under „Ports (COM & LPT)“.
- Reboot your computer.
- For high-level language support, demos and test programs please install the ME-Software-Developer-Kit (see separate chapter).
- Test the function of the board with the test program.

2.2.3 Checking Installation under Windows

**If the Windows version is supported by the appropriate board type (see README-files)*

Use the Device Manager to check the proper entry of the board.

- Run the Device Manager:
    - **Under Windows 95/98/ME:**
      START-Menu→ Settings → System Control → System → Device Manager
    - **Under Windows 2000:**
      START-Menu → Settings→ System Control → System → System Properties → Hardware → Device Manager
    - **Under Windows XP:**

START-Menu ➜ System Control ➜ Performance and Maintenance ➜ System ➜ Hardware ➜ Device Manager

- Check the entry of your board in the device manager. The entry must not be marked with the symbol „exclamation mark on a yellow circle". This would indicate an installation error.
- By double clicking, open the window „Settings for…".
  - When clicking the property page **Resources**, the following resource settings are shown: interrupt, memory range and I/O range. Under „Device conflicts" the message „No conflicts" must be found.

If one of the above noted entries does not exist, check the following items:

? Your new board was assigned to the device class „? Other Components". This could happen when installing Windows when a new board is plugged in the computer or if the installation was cancelled (depending on the system)

- If this occurs, delete the entry, reboot your computer and run the driver installation once more.

? Is there an address or interrupt conflict?

- Edit the BIOS settings of your computer if necessary (possibly reserving an IRQ channel for ISA boards) or change the interrupt assignment under Windows. Refer to the manuals of the relevant devices.
- The property page **General** should show „Device is ready to run" under „Device Status" and the property page **Resources** must show „No conflicts" under „Device conflicts".
- Reboot your computer if you have changed any settings.

2.2.4 Installation under Windows NT 4.0

! If you have got the driver as a self-extracting archive (e.g. by download or from CD/DVD) you must unpack the driver software first. See chap. 2.2.1 Unpacking the Driver Software.

- If the setup program doesn't start automatically after unpacking the driver, browse to the ME-9x00 driver installation (default: C:\Meilhaus\ME-9x00\install\sys) and start the SETUP.EXE file there by double-click.
    - The driver will be installed.
- Reboot your computer.
- For high-level language support, demos and test programs please install the ME-Software-Developer-Kit (see separate chapter).
- Test the function of the board by the test program.

2.2.5 Checking the Installation under Windows NT 4.0

The proper entries for the memory range and interrupt can be checked in Windows NT Diagnostics under **Resources**.

2.2.6 Installation of the ME Software Developer Kit

The ME Software Developer Kit (ME-SDK) provides programming support for all common high-level languages as well as samples, test programs and tools for all ME PCI boards. The installation is optional.

- Navigate to the directory with the self-extracting archive file of the ME-SDK (mesdk.exe). When installing from the ME-Power-DVD choose `<Drive>:\Legacy\me-boards\me-sdk`. Select the file „mesdk.exe“ and start unpacking by OK.
- Enable the option „Overwrite Files Without Prompting“
- Choose a directory and click on „Unzip“. By default the files are copied to the directory C:\Meilhaus\me-sdk.
- High-level language support, examples, tools and test programs will be copied.
- Click on „Close“

Note: The system driver must be installed separately.

2.2.7 Updating the System Driver

If you have got the driver update as a self-extracting archive (e.g. by download) please unpack the driver software first (see „Unpacking the Driver Software" on page 15).

2.2.7.1 ...under Windows 95/98/Me/2000/XP

- Run the Device Manager:
  - **Under Windows 95/98/ME:**
    START-Menu → Settings → System Control → System → Device Manager
  - **Under Windows 2000:**
    START-Menu → Settings → System Control → System → System Properties → Hardware → Device Manager
  - **Under Windows XP:**
    START-Menu → Settings → Performance and Maintenance → System → Hardware → Device Manager
- Choose the property page **Driver** in the device manager and click on the button **Update Driver**.
- The Device Driver Wizard will be started. Choose the option „Specify a location" and click on **Browse**.
- Choose the source for the driver update, e.g. CD/DVD. If you got the driver update as an archive file (e.g. by download), browse your computer for the directory you unpacked the driver to (default: `C:\Meilhaus\ME-9x00\install\`...). Start the update with OK.
  - The driver will be updated.
- Reboot your computer.

2.2.7.2 ...under Windows NT 4.0

! If you have got the driver as a self-extracting archive (e.g. by download or from CD/DVD) you must unpack the driver software first. See chap. 2.2.1 "Unpacking the Driver Software".

Install the new driver as described in chapter 2.2.4 Installation under Windows NT 4.0.

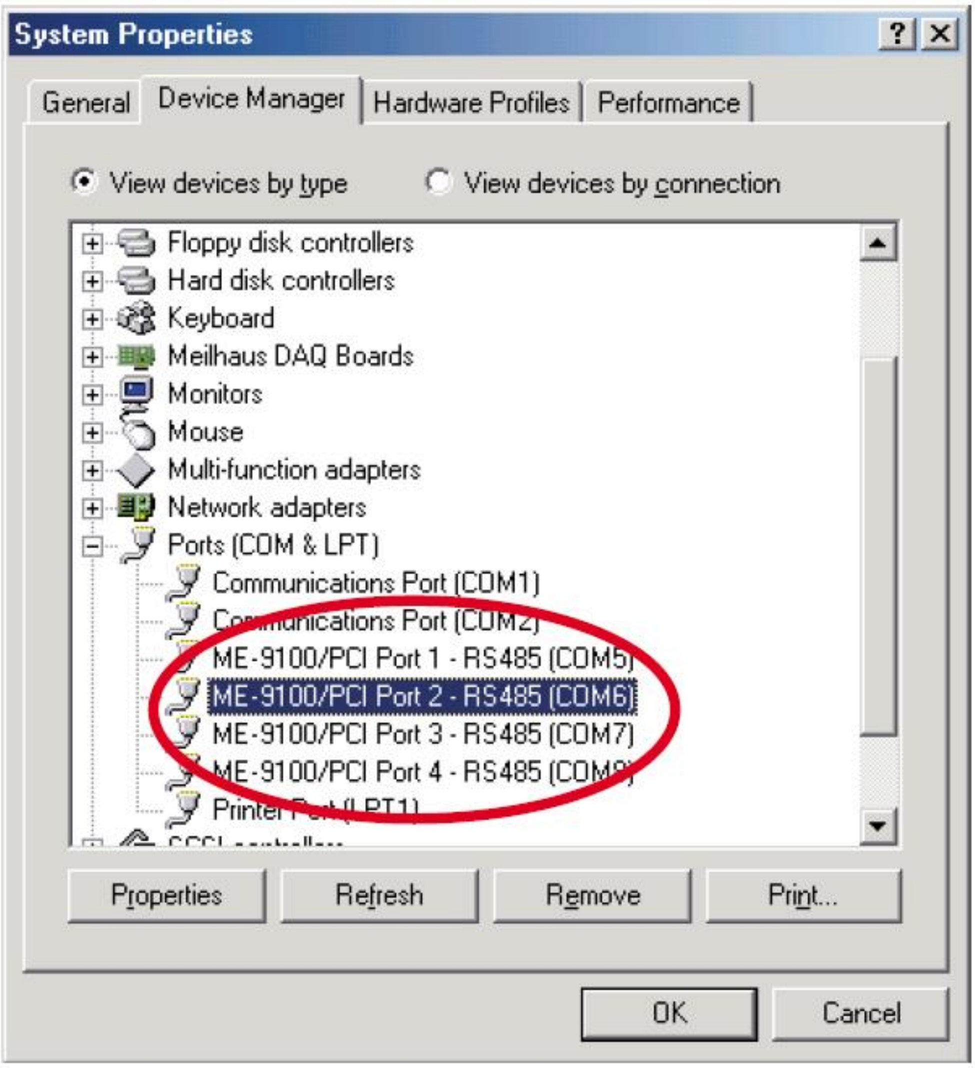
2.3 Port Configuration

2.3.1 ...under Windows 95/98/Me

Use the device manager to check the port assignments and to change the settings. Choose:

> **START-Menu → Settings → System Control → System → Device Manager**

The ports of the ME-9100/9300 will be added under „Ports (COM & LPT)" and the COM ports automatically assigned. In the following picture you see a typical installation of a ME-9100 with 4 ports for RS-485 operation:



● View devices by type ○ View devices by connection

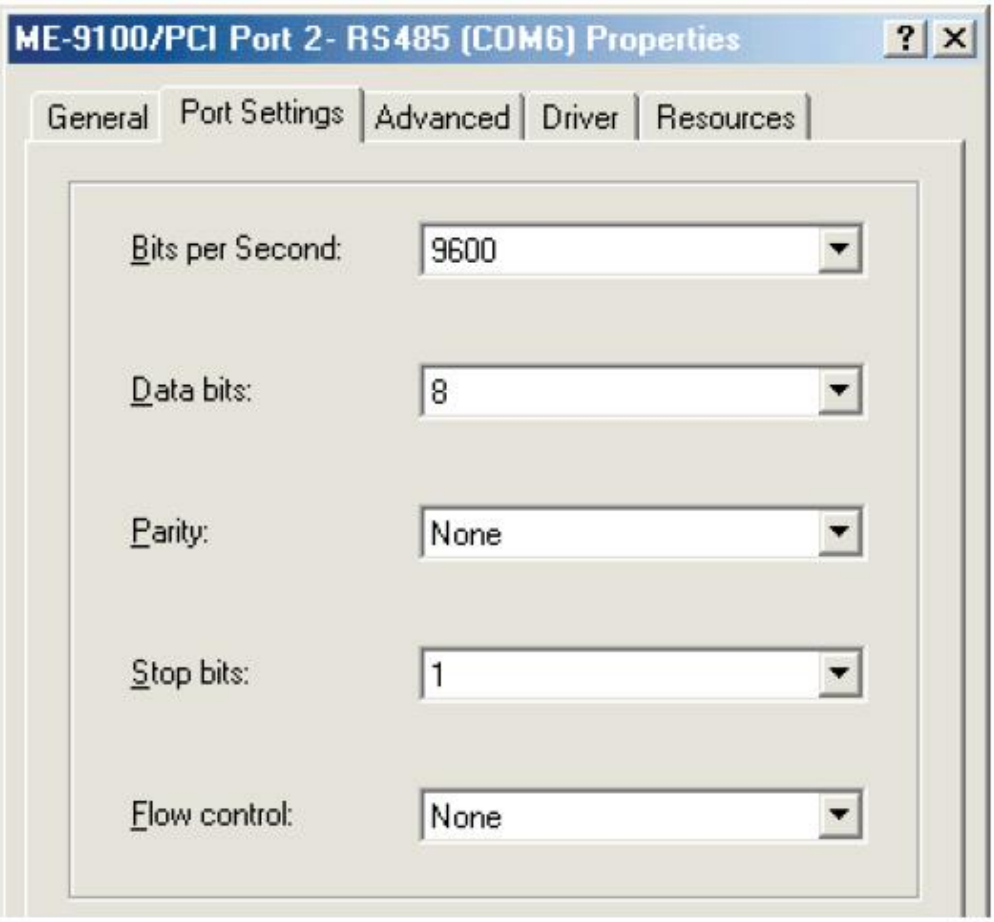
Picture 1: Entry of ports in the device manager

Mark the appropriate port in the device manager under „Ports“(see Picture 1) and click the button „Properties“ in order to display the properties.

2.3.1.1 Port Settings

For most of the application programs the „Port Settings“ done in the device manager are not relevant. Exception: board-specific parameters like the operation mode (see 2.3.1.2 Settings „Advanced“). Therefore check the transfer parameters in your application program (e.g. Hyper Terminal). The following settings are possible:

- **Bits per Second:** depending on board version all settings selectable in Windows up to 921,6 kbps (see also appendix A "Specifications", page 100)
- **Data bits**: 4; 5; 6; 7; 8
- **Parity:** None, Odd, Even, Mark, Space
- **Stop bits**: 1; 1,5; 2
- **Flow control Xon/Xoff, Hardware, None**


ME-9100/PCI Port 2- RS485 (COM6) Properties

Bits per Second: 9600

Data bits: 8

Parity: None

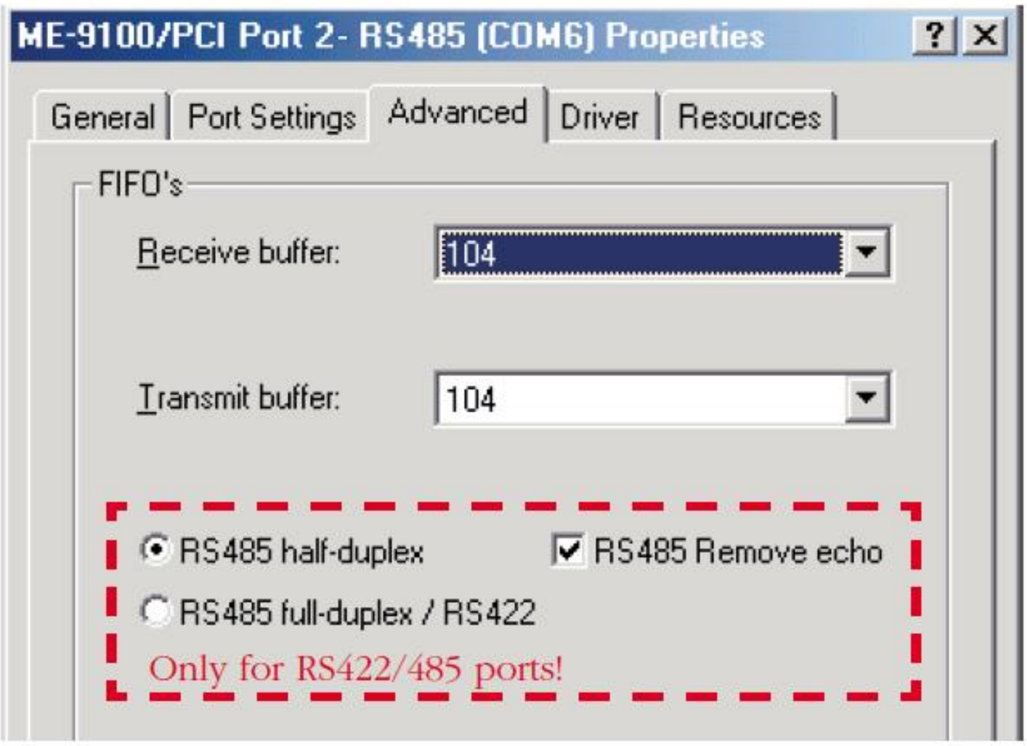
Stop bits: 1

Flow control: None

Picture 2: Port Settings (standard)

2.3.1.2 Settings „Advanced“

The property page „Advanced“ offers you the ability to set the operation mode of RS422/485 ports (see chap 3.7 Operation Modes) and the FIFO trigger level. Every port provides separate transmit and receive FIFOs. Each of them with a size of 128 bytes (ME-9100/9300). For adaption to your system there is the possibility to set the trigger level for reading resp. writing the FIFO. When the trigger level is matched an interrupt occurs. For most applications the default setting is useful.



Receive buffer: 104

Transmit buffer: 104

● RS485 half-duplex ☑ RS485 Remove echo

○ RS485 full-duplex / RS422

Only for RS422/485 ports!

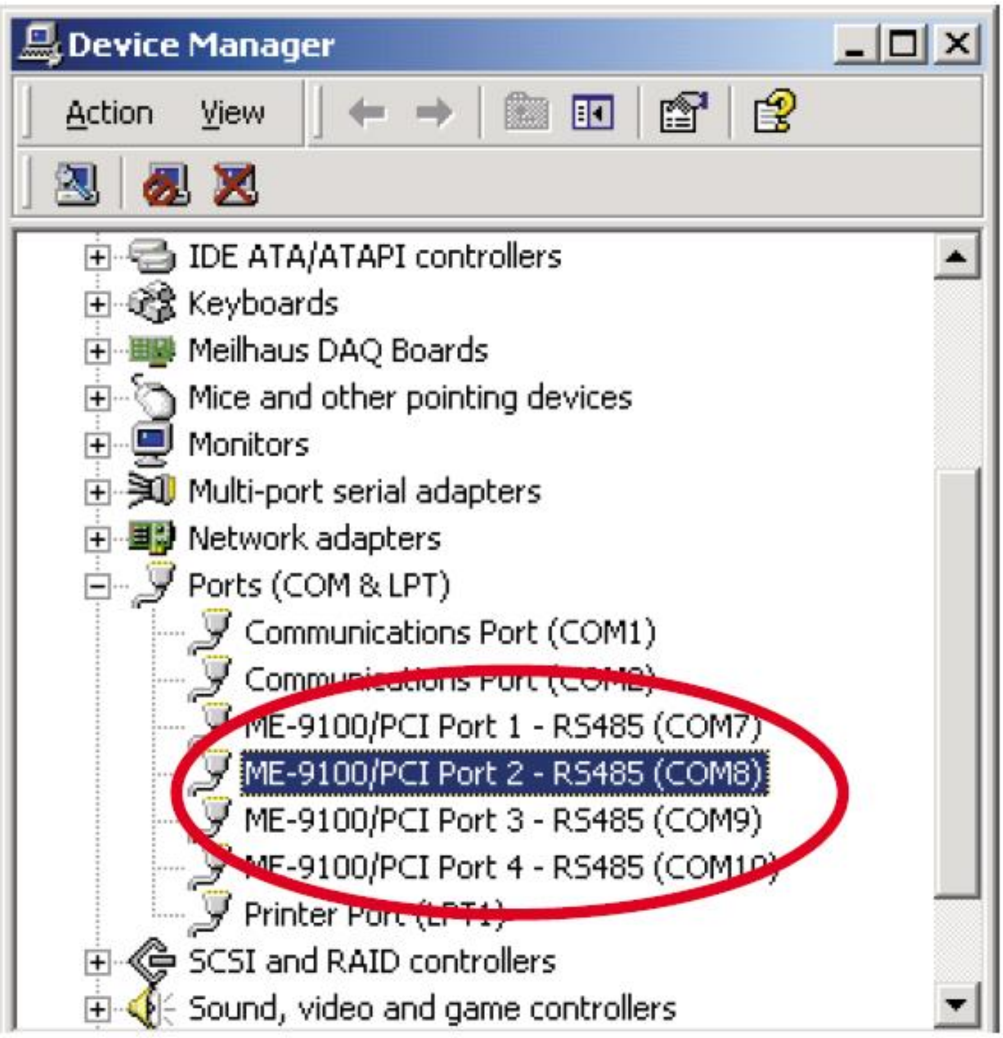
Picture 3: Port Settings "Advanced"

2.3.2 ...under Windows 2000/XP

Use the device manager to check the port assignments and change the settings. Choose:

- **Under Windows 2000:**
  START-Menu → Settings → System Control → System → System Properties → Hardware → Device Manager
- **Under Windows XP:**
  START-Menu → Settings → Performance and Maintenance → System → Hardware → Device Manager

The ports of the ME-9000/9100/9300 as well as the ME-90 PC/104-Plus will be added under „Ports (COM & LPT)“ and the COM ports automatically assigned. In the following picture you see a typical installation of a ME-9100 with 4 ports for RS-485 operation:



- IDE ATA/ATAPI controllers
- Keyboards
- Meilhaus DAQ Boards
- Mice and other pointing devices
- Monitors
- Multi-port serial adapters
- Network adapters
- Ports (COM & LPT)
  - Communications Port (COM1)
  - Communications Port (COM2)
  - ME-9100/PCI Port 1 - RS485 (COM7)
  - ME-9100/PCI Port 2 - RS485 (COM8)
  - ME-9100/PCI Port 3 - RS485 (COM9)
  - ME-9100/PCI Port 4 - RS485 (COM10)
  - Printer Port (LPT1)
- SCSI and RAID controllers
- Sound, video and game controllers

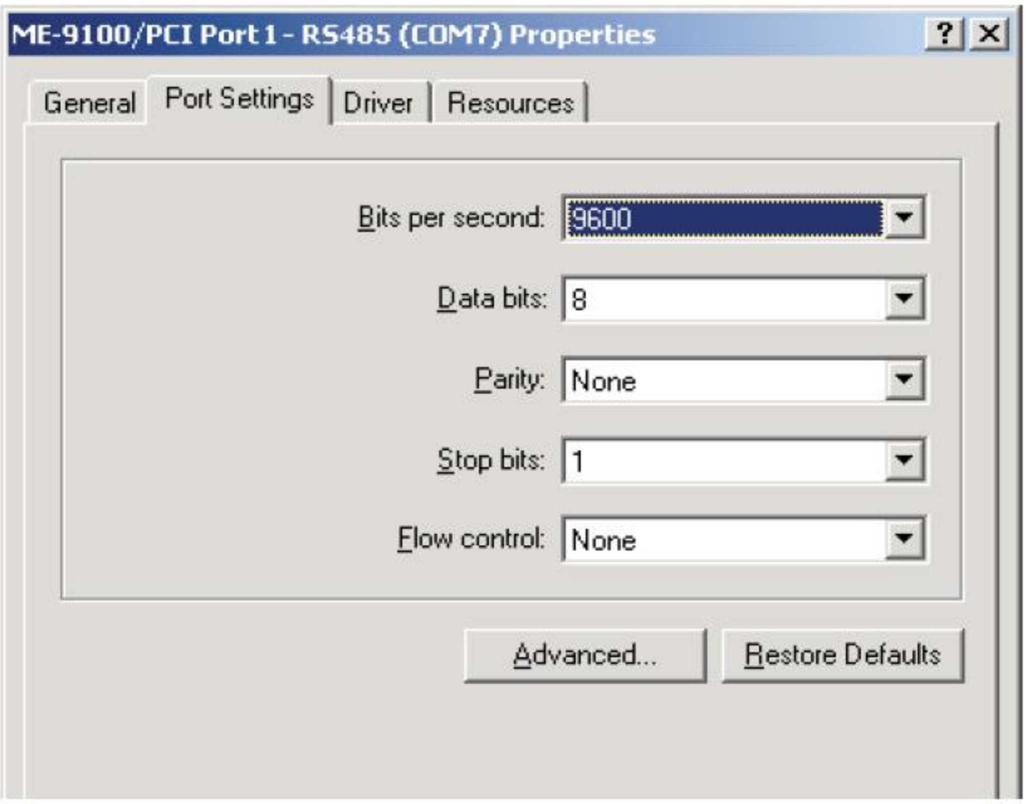
Picture 4: Entry of ports in the device manager

Mark the appropriate port in the device manager under “Ports” (see picture 4) and click the button “Properties” to display the settings.

2.3.2.1 Port Settings

For most of the application programs the „Port Settings“ done in the device manager are not relevant. Exception: board-specific parameters like the operation mode (see chap. 2.3.2.2). Therefore check the transfer parameters in your application program (e.g. Hyper Terminal). The following settings are possible:

- **Bits per Second:** depending on board version all settings selectable in Windows up to 921,6 kbps (see also appendix A "Specifications", page 100)
- **Data bits**: 4; 5; 6; 7; 8
- **Parity:** None, Odd, Even, Mark, Space
- **Stop bits**: 1; 1,5; 2
- **Flow control Xon/Xoff, Hardware, None**



ME-9100/PCI Port 1 - RS485 (COM7) Properties

Bits per second: 9600

Data bits: 8

Parity: None

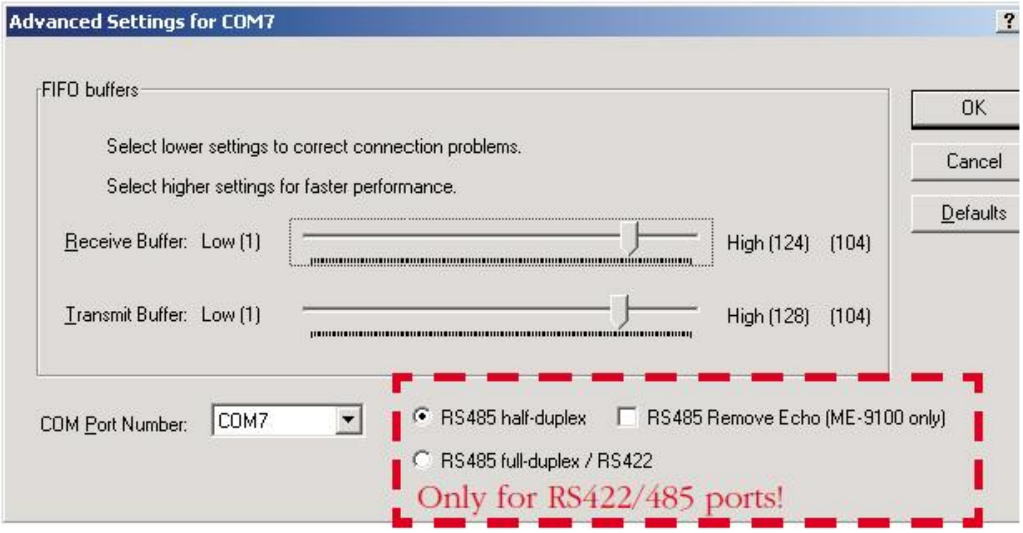
Stop bits: 1

Flow control: None

Picture 5: Port settings (standard)

2.3.2.2 Settings “Advanced”

The property page „Advanced“ (see picture 5) offers you the ability to set the operation mode of RS422/485 ports (see chap 3.7 Operation Modes) and the FIFO trigger level. Every port provides separate transmit and receive FIFOs. Each of them with a size of 64 byte (ME-9000 and ME-90 PC/104-Plus) resp. 128 bytes (ME-9100/9300). For adaption to your system there is the possibility to set the trigger level for reading resp. writing the FIFO. When the trigger level is matched an interrupt occurs. For most applications the default setting is useful.



Advanced Settings for COM7

FIFO buffers

Select lower settings to correct connection problems.

Select higher settings for faster performance.

Receive Buffer: Low (1) High (124) (104)

Transmit Buffer: Low (1) High (128) (104)

COM Port Number: COM7

● RS485 half-duplex □ RS485 Remove Echo (ME-9100 only)

○ RS485 full-duplex / RS422

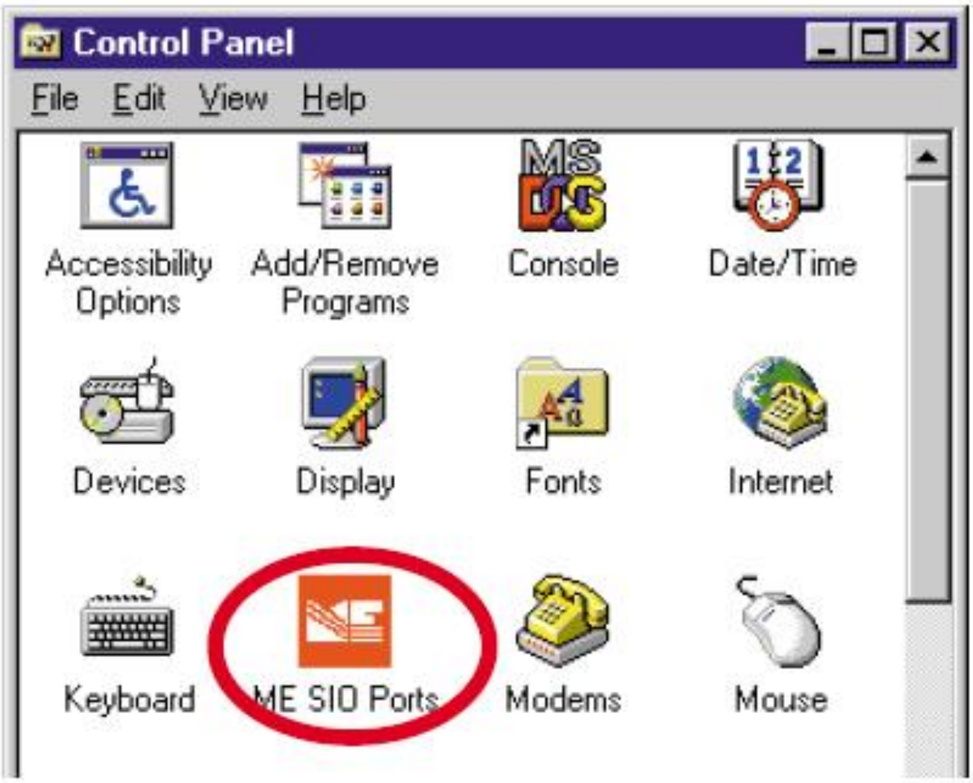
Only for RS422/485 ports!

Picture 6: Port settings "Advanced"

We recommend **not** to change the assignment of the COM ports by the pull-down menu “COM Port Number”.

2.3.3 ...under Windows NT 4.0

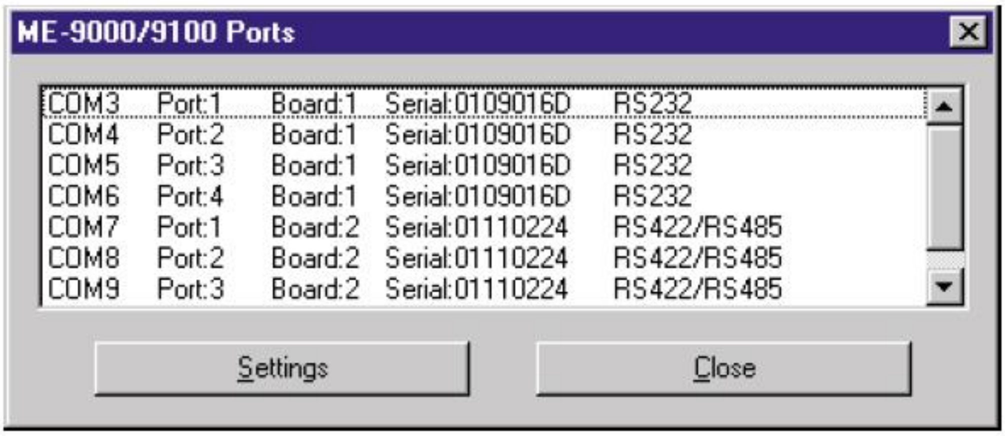
The assignment of COM ports is done automatically by the operation system. Under „Control Panel“ double-click the icon „ME SIO Ports“ to check the port assignments and change the settings.



Picture 7: Control panel

In the following picture you see a typical installation of two boards, the first one with 4 RS-232 ports and the second one with 4 RS-485 ports. You get the following information:

Beginning from the left side the name is displayed which refers to the port, followed by the assignment of the port number to the board, the corresponding serial number and last the interface standard (RS232 or RS485). COM10 is not visible on the picture.

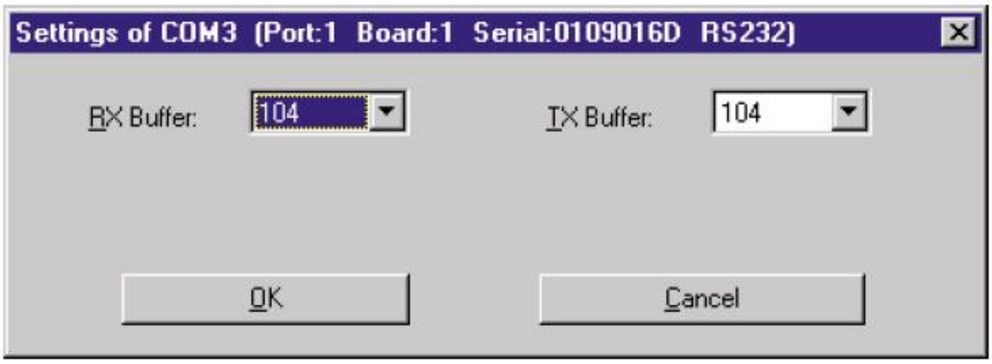


```
COM3   Port:1   Board:1   Serial:0109016D   RS232
COM4   Port:2   Board:1   Serial:0109016D   RS232
COM5   Port:3   Board:1   Serial:0109016D   RS232
COM6   Port:4   Board:1   Serial:0109016D   RS232
COM7   Port:1   Board:2   Serial:01110224   RS422/RS485
COM8   Port:2   Board:2   Serial:01110224   RS422/RS485
COM9   Port:3   Board:2   Serial:01110224   RS422/RS485
```

Picture 8: Control panel "ME SIO Ports"

2.3.3.1 Settings “ME SIO Ports”

Mark a port in the control panel „ME SIO Ports“ (see Picture 8) and click the button „Settings“. The next dialog offers you the ability to set the operation mode of RS422/485 ports (see chap. 3.7Operation Modes) and the FIFO trigger level. Every port provides separate transmit and receive FIFOs. Each of them with a size of 64 bytes (ME-9000 and ME-90 PC/104-Plus) resp.128 bytes (ME-9100/9300). For adaption to your system there is the possibility to set the trigger level for reading resp. writing the FIFO. When the trigger level is matched an interrupt occurs. For most applications the default setting is useful. (RX Buffer = Receive FIFO and TX Buffer = Transmit FIFO.

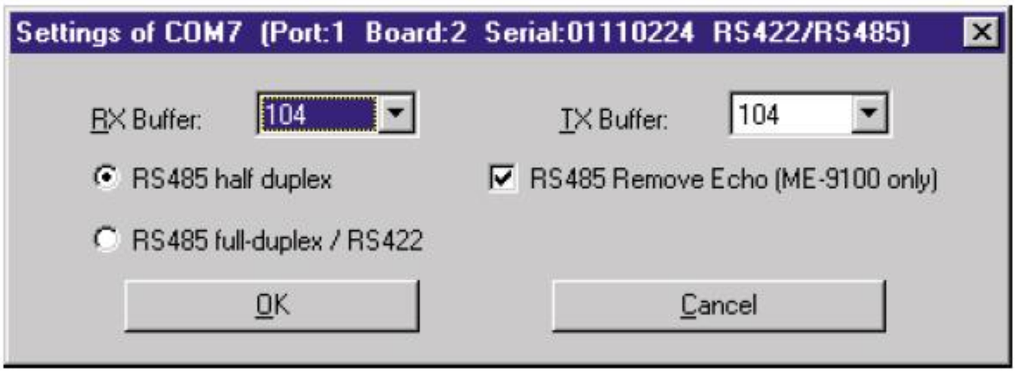


Settings of COM3 (Port:1 Board:1 Serial:0109016D RS232)

RX Buffer: 104

TX Buffer: 104

Picture 9: Settings "ME SIO ports" (RS232)



Settings of COM7 (Port:1 Board:2 Serial:01110224 RS422/RS485)

RX Buffer: 104

TX Buffer: 104

● RS485 half duplex

☑ RS485 Remove Echo (ME-9100 only)

○ RS485 full-duplex / RS422

Picture 10: Settings "ME SIO ports" (RS422/485)

2.3.3.2 Port Settings

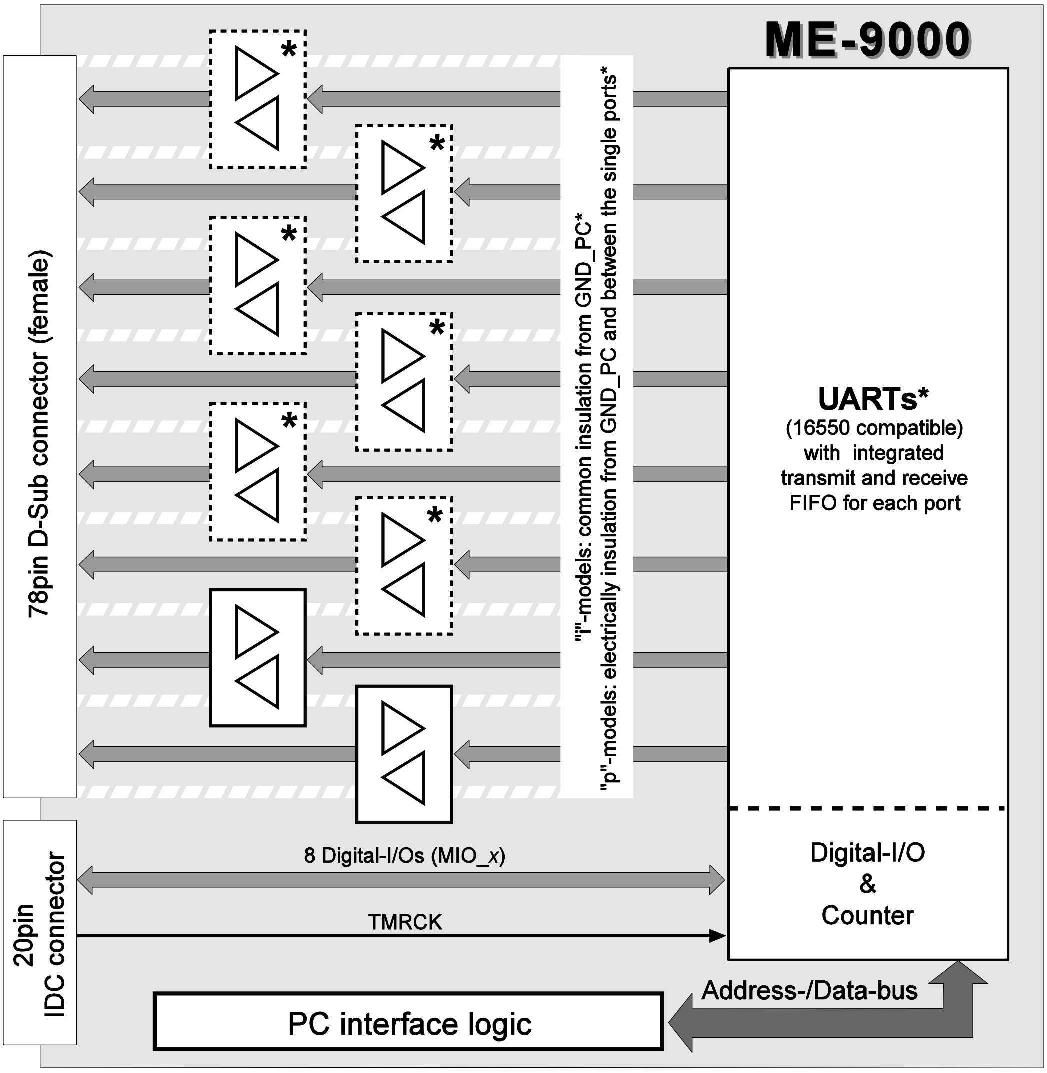
For most of the application programs the settings done in the control-panel „Ports“ are not relevant. Exception: board-specific parameters like the operation mode (see chap. 2.3.3.1 Settings

"ME SIO Ports"). Therefore check the transfer parameters in your application program (e.g. Hyper Terminal). The following settings are possible:

- **Bits per Second:** depending on board version all settings selectable in Windows up to 921,6 kbps (see also appendix A "Specifications", page 101)
- **Data bits**: 4; 5; 6; 7; 8
- **Parity:** None, Odd, Even, Mark, Space
- **Stop bits**: 1; 1,5; 2
- **Flow control Xon/Xoff, Hardware, None**

3 Hardware

3.1 Block Diagram ME-9000



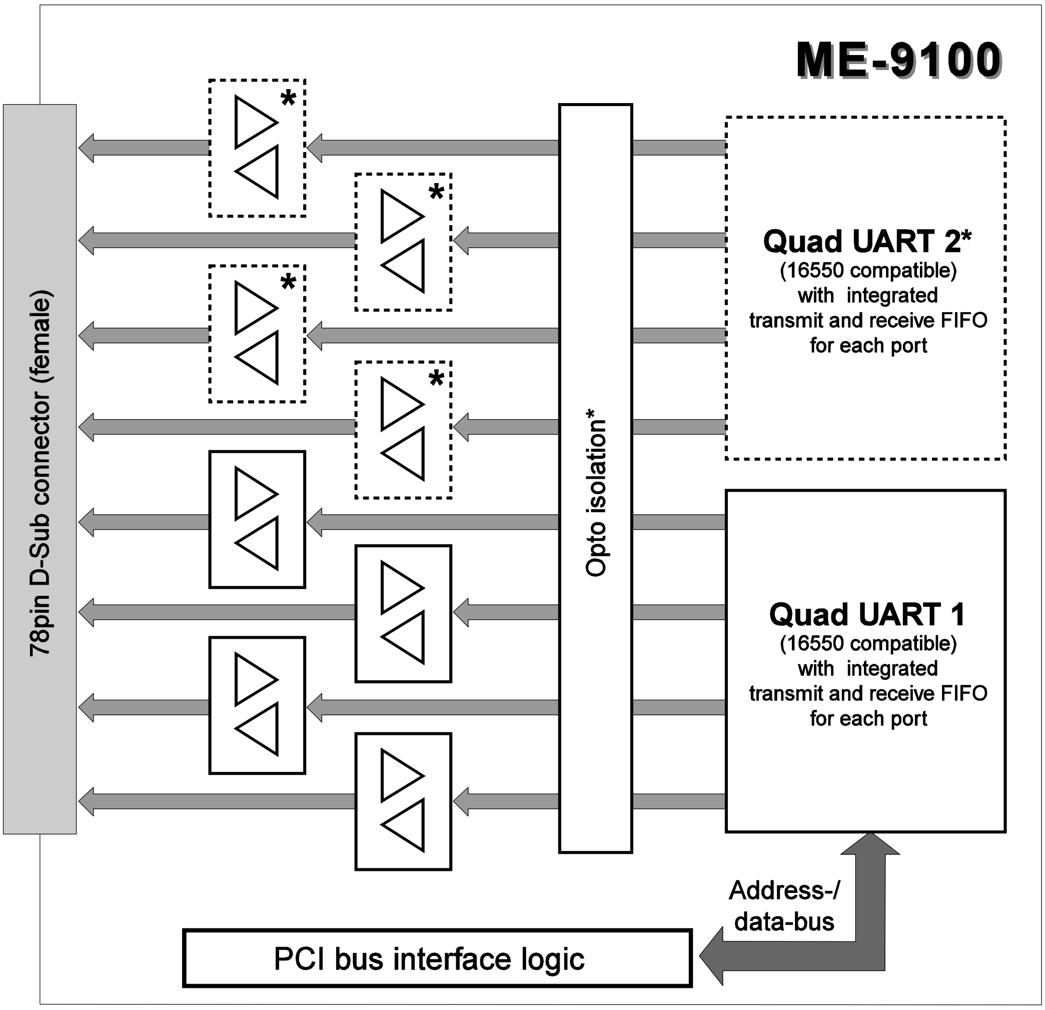
Picture 11: Block diagram of the ME-9000(i/p)

*2, 4 or 8 RS-232 and/or RS-422/485 ports depending on version. Optional:

„i“-models: with a common electrical insulation from PC ground.

„p“-models: with electrical insulation from PC ground **and** between the ports („island-ports“).

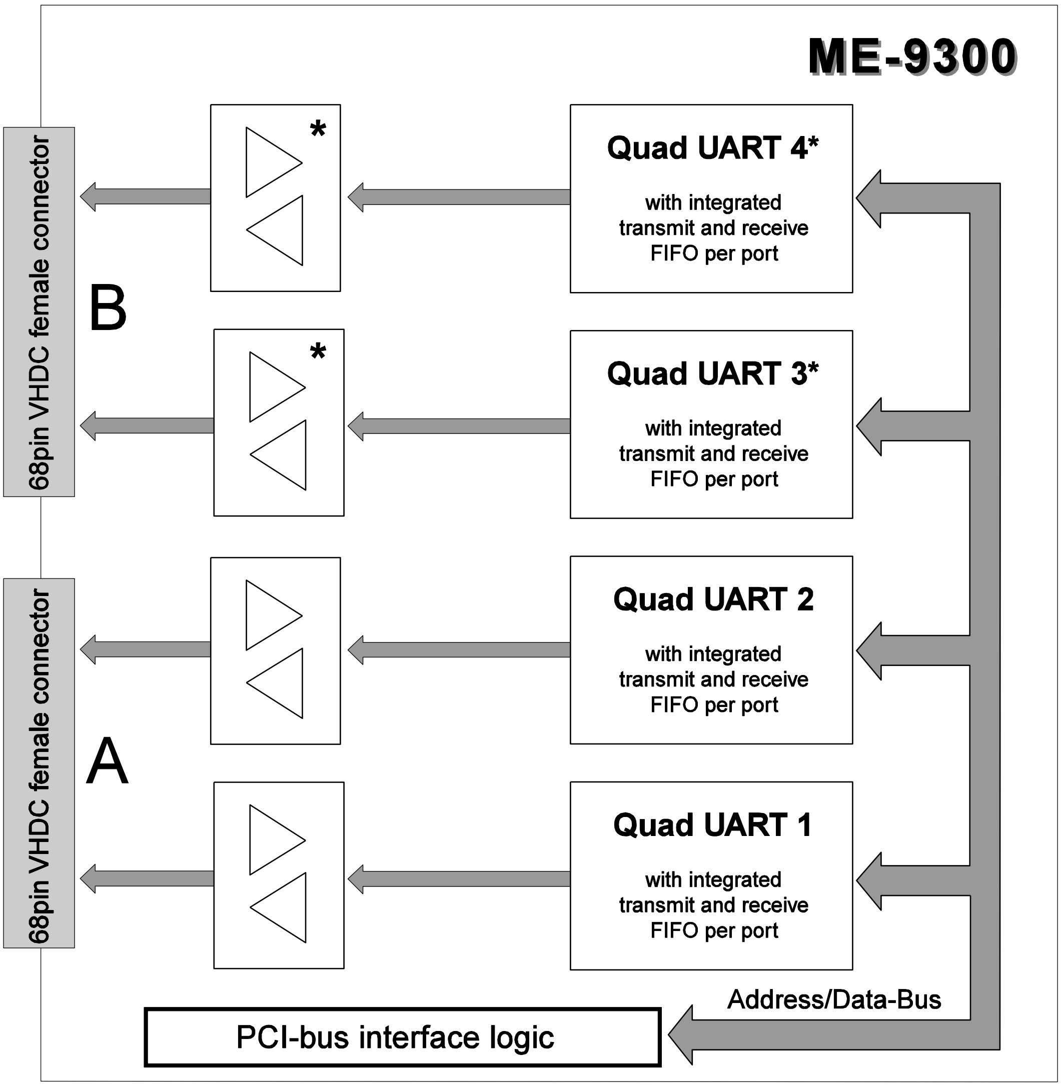
3.2 Block Diagram ME-9100



Picture 12: Block diagram of the ME-9100i

*2, 4 or 8 RS-232 and/or RS-422/485 ports depending on version.

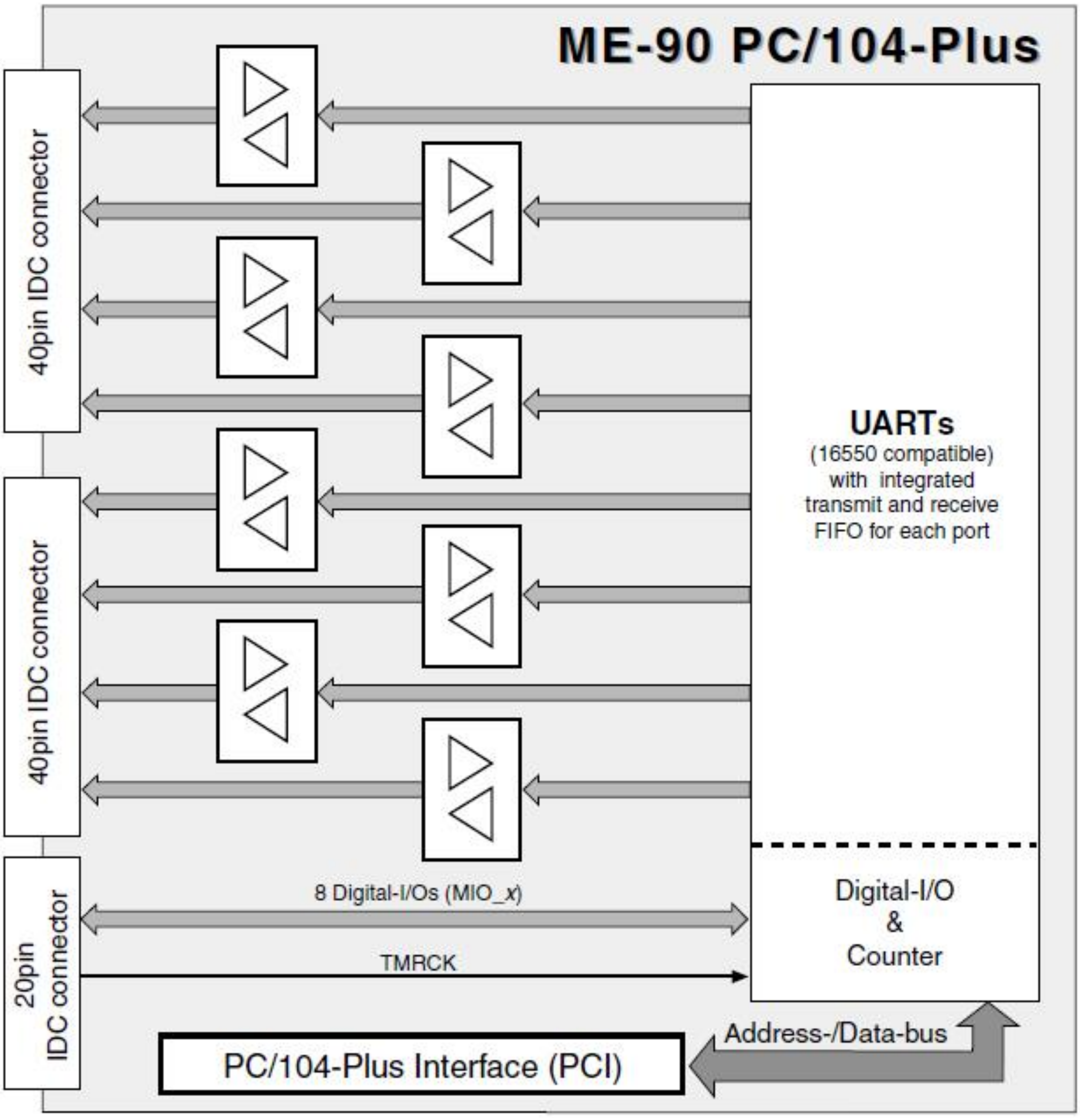
3.3 Block Diagram ME-9300


ME-9300
68pin VHDC female connector
B
68pin VHDC female connector
A
*
*
Quad UART 4*
with integrated transmit and receive FIFO per port
Quad UART 3*
with integrated transmit and receive FIFO per port
Quad UART 2
with integrated transmit and receive FIFO per port
Quad UART 1
with integrated transmit and receive FIFO per port
Address/Data-Bus
PCI-bus interface logic

Picture 13: Block diagram of the ME-9300

*8 or 16 RS-232 ports depending on version.

3.4 Block Diagram ME-90 PC/104-Plus



Picture 14: Block diagram ME-90 PC/104-Plus

*8 RS-232 ports or 8 RS-422/485 ports or mixed 4 RS-232 and 4 RS-422/485 ports depending on version.

3.5 Hardware Options

Depending on the model, the boards differ concerning:

- Number of ports
- Standard-TTL (without opto-isolation), with common opto-isolation ("I"-models) and with "island-ports" ("p"-models).
- RS-232 or RS-422/485 ports or mixed.

The "**MIX**"-versions provide RS-232 as well as RS-422/485 ports. The RS-232 ports occupy always the lower significant ports followed by the RS-485 ports.

With the following diagrams you can determine what version you are using if necessary.

3.5.1 ME-9000 PCI/PCIe/cPCI

3.5.1.1 Multi-I/O Port

On the ME-9000 PCI (from Rev. 1.4 up), ME-9000 cPCI (from Rev. 1.2 up) and ME-9000 PCI-Express the multi-I/O port of the EXAR XR17D158 chip can be used for customer-specific extensions. The 26-pin IDC connector (ST2) provides VCC (depending on model +5 V or +3.3 V), PC ground (GND_PC), 8 multi-I/O pins (MIO_x) and the clock input TMRCK. All the other pads are not connected (n.c.). The pinout of ST2 can be found in appendix B5. A detailed description of the chip can be found in the data-sheet of the manufacturer under www.exar.com.

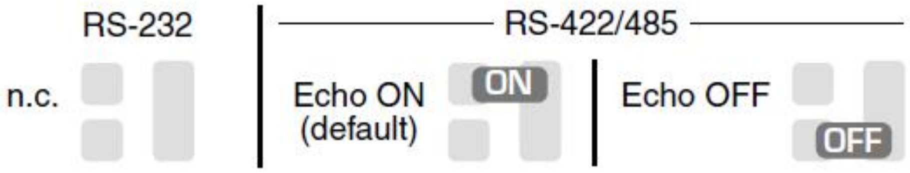
Attention: VCC at ST2 must not exceed 300 mA!

3.5.1.2 Echo ON/OFF

On the RS-422/485 models of the ME-9000 you can determine the receiver's control by soldering bridges at the bottom of the board. With that you can adapt the single ports to the needs of your application individually. See also description of the operation modes from page 56 up.

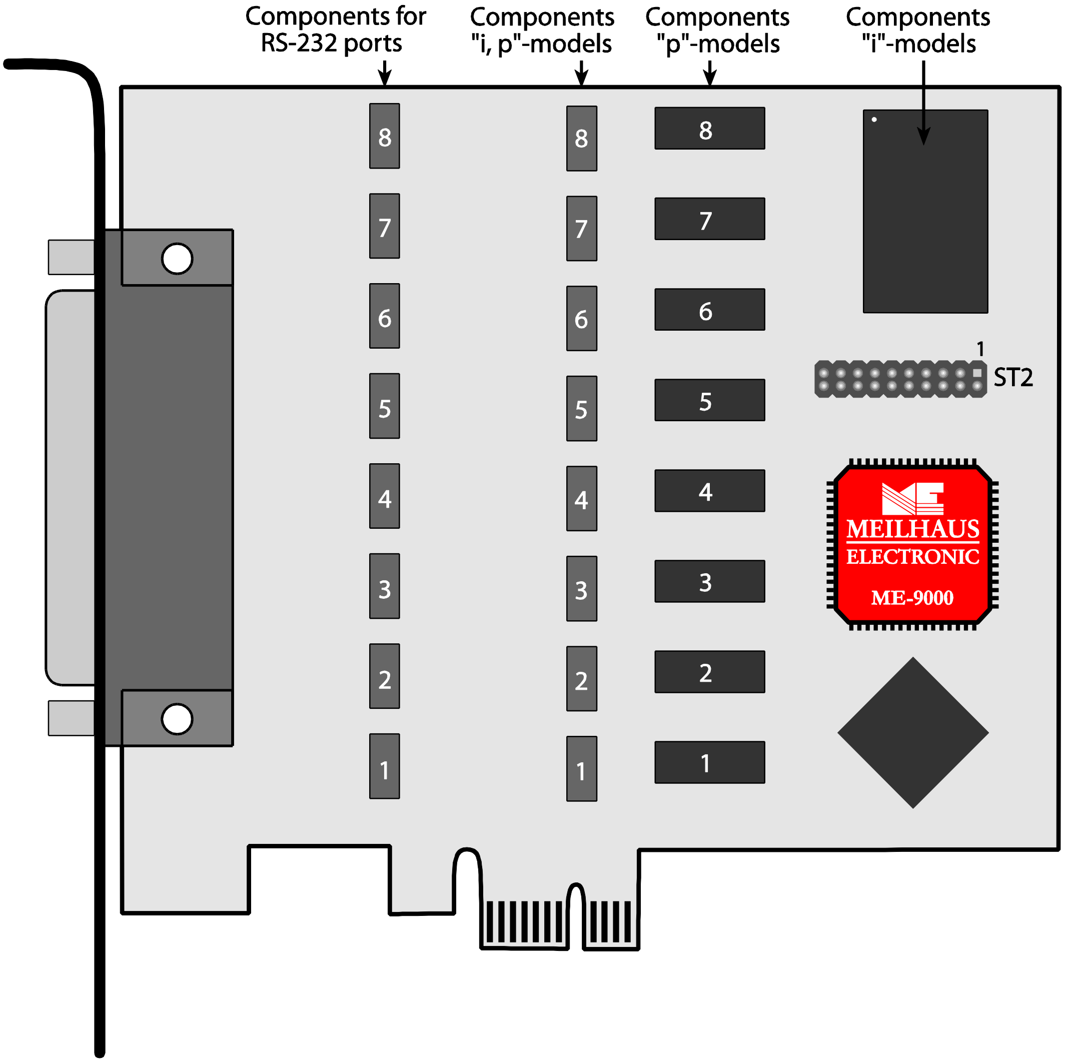
- **Echo ON**: Set the soldering bridge „ON“ and the receiver listens permanently (default setting).

- **Echo OFF**: Set the soldering bridge „OFF“ to control the receiver by the RTS signal. The reception becomes disabled during transmission.

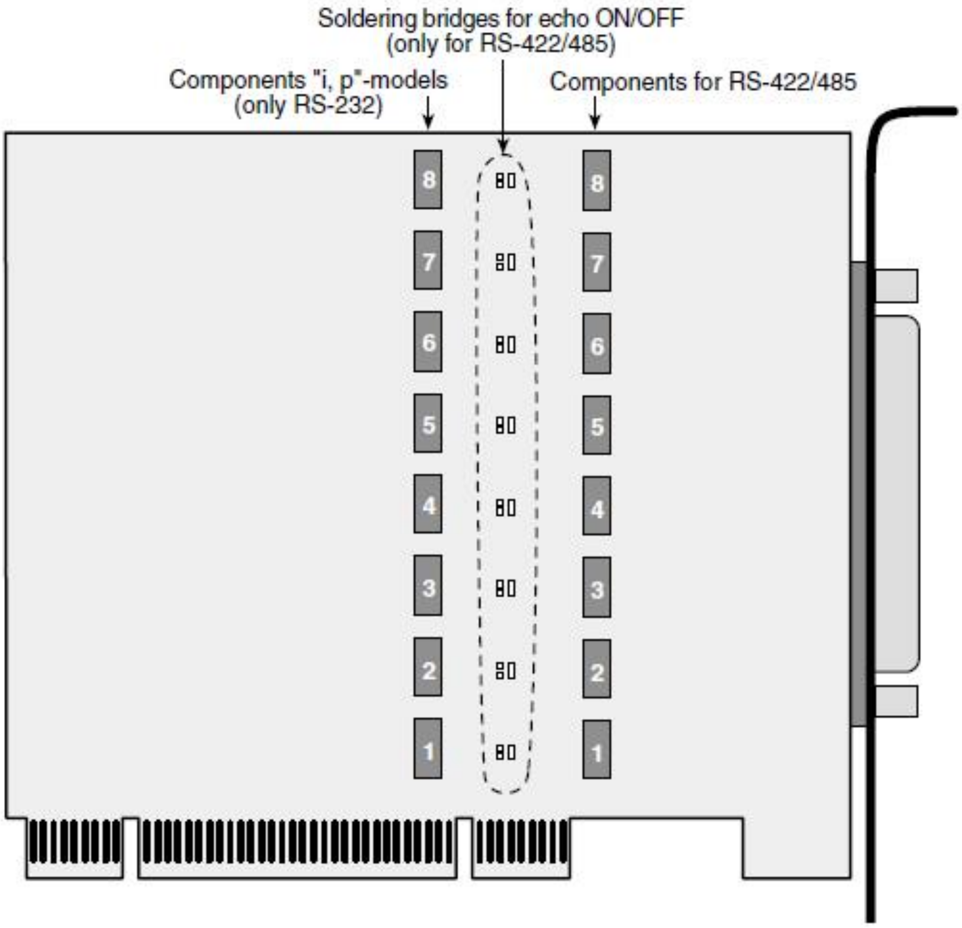


Picture 15: Soldering bridges

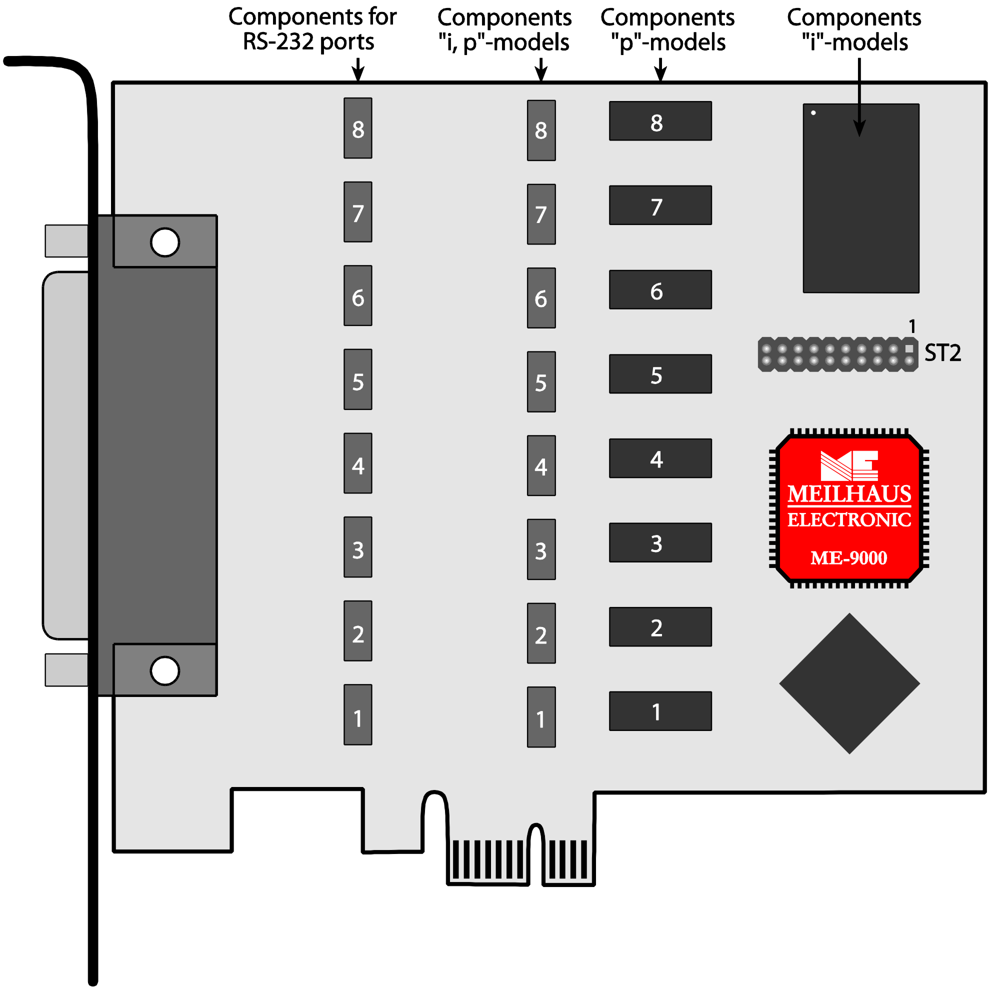
The position of the soldering bridges can be found with picture 17 (PCI), picture 19 (PCIe) resp. picture 21 (cPCI). Alternatively you can send your board to our service department. We will configure your board for your needs.



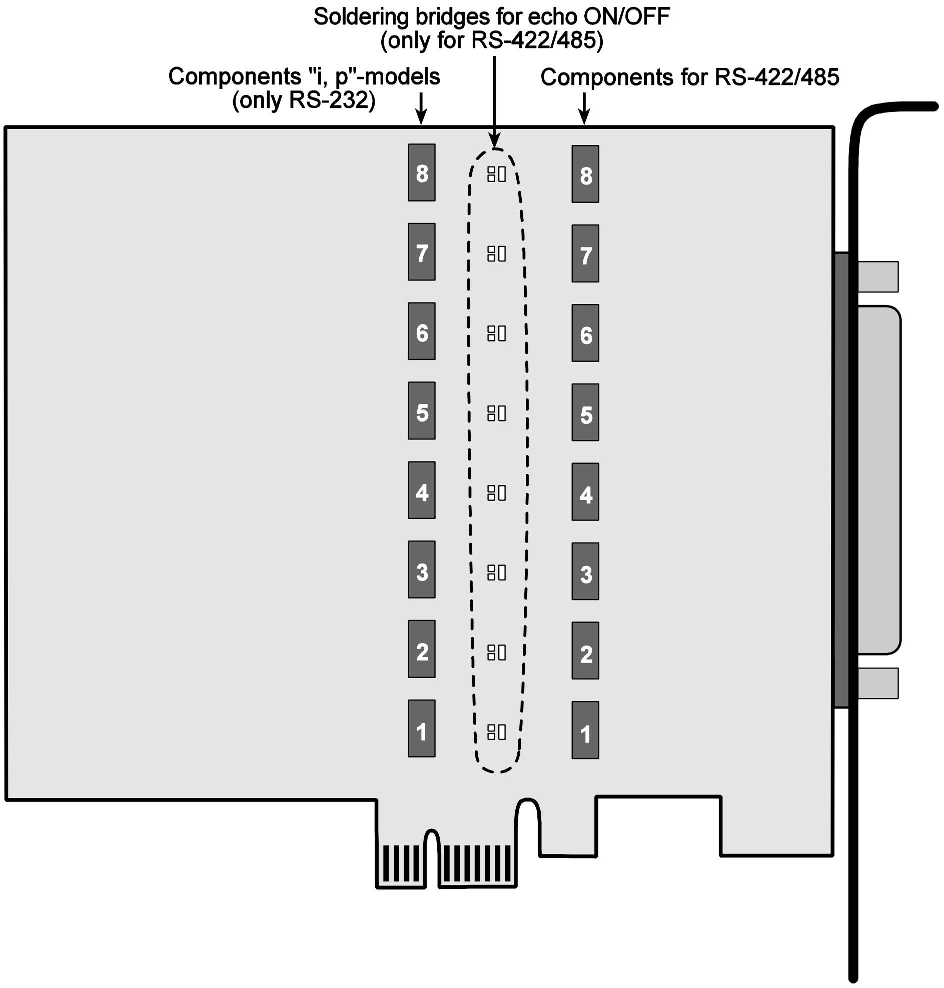
Picture 16: ME-9000 PCI Rev 2.2 (top view)



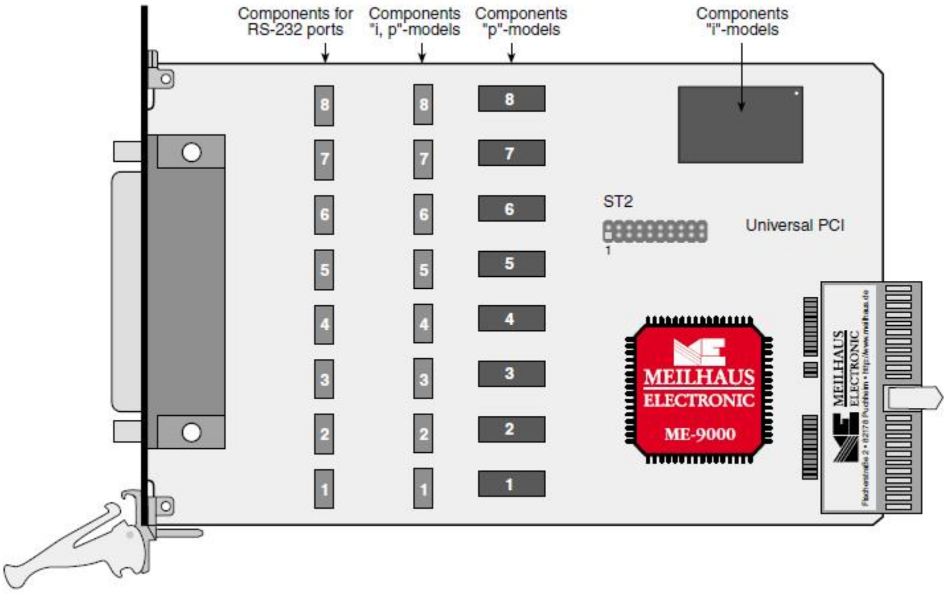
Picture 17: ME-9000 PCI Rev 2.2 (bottom view)



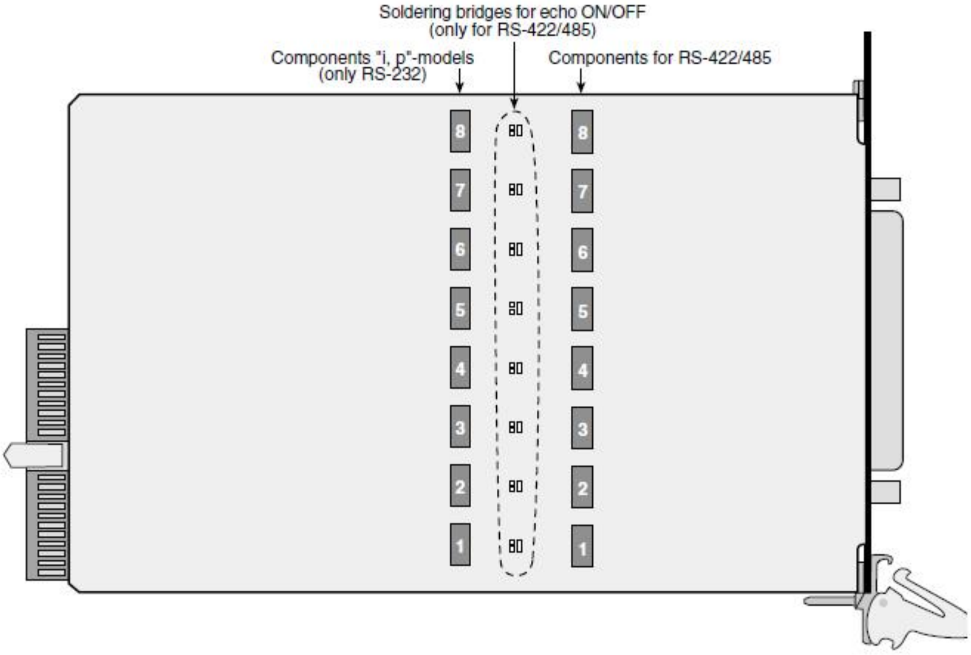
Picture 18: ME-9000 Compact PCI Rev. 2.2 (top view)



Picture 19: ME-9000 PCI-Express (bottom view)

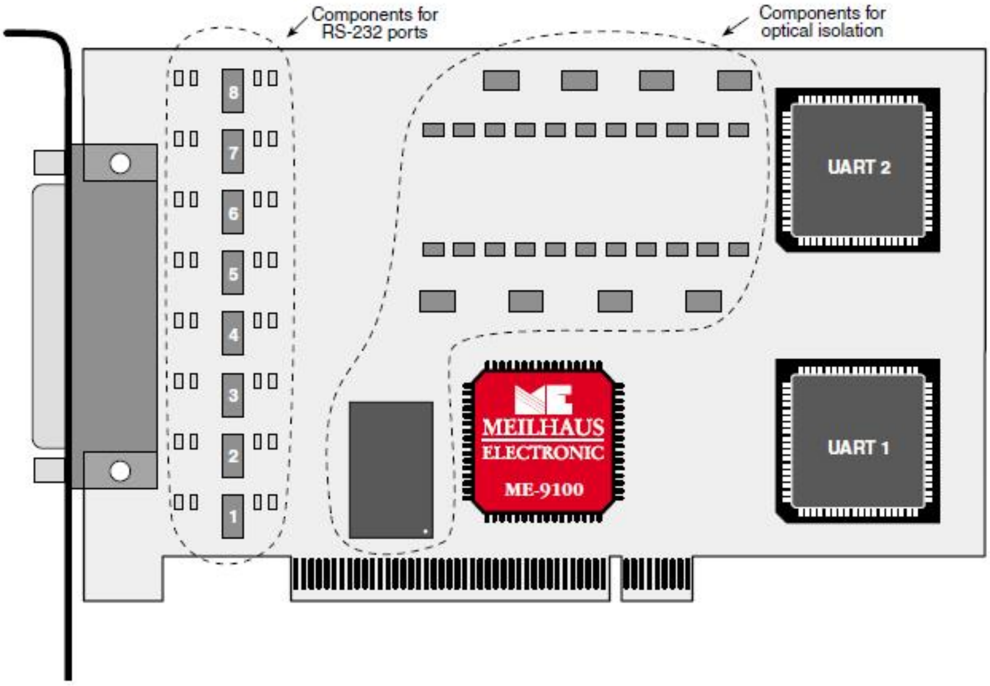


Picture 20: ME-9000 CompactPCI Rev. 2.2 (top view)

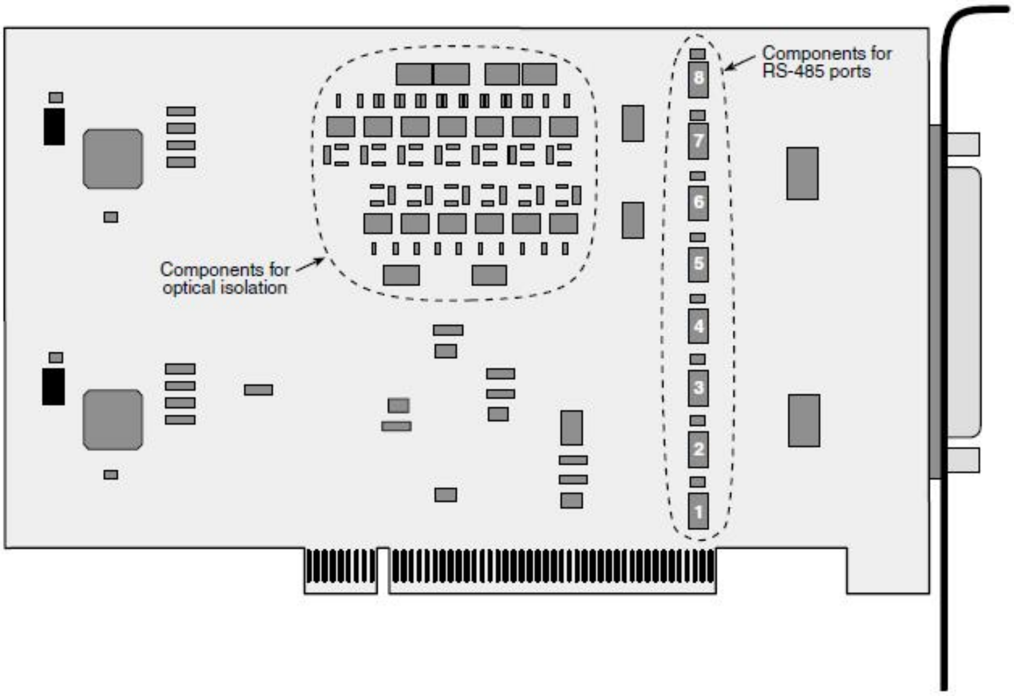

Soldering bridges for echo ON/OFF (only for RS-422/485)
Components "i, p"-models (only RS-232)
Components for RS-422/485
8
7
6
5
4
3
2
1

Picture 21: ME-9000 CompactPCI Rev. 2.2 (bottom view)

3.5.2 ME-9100 PCI

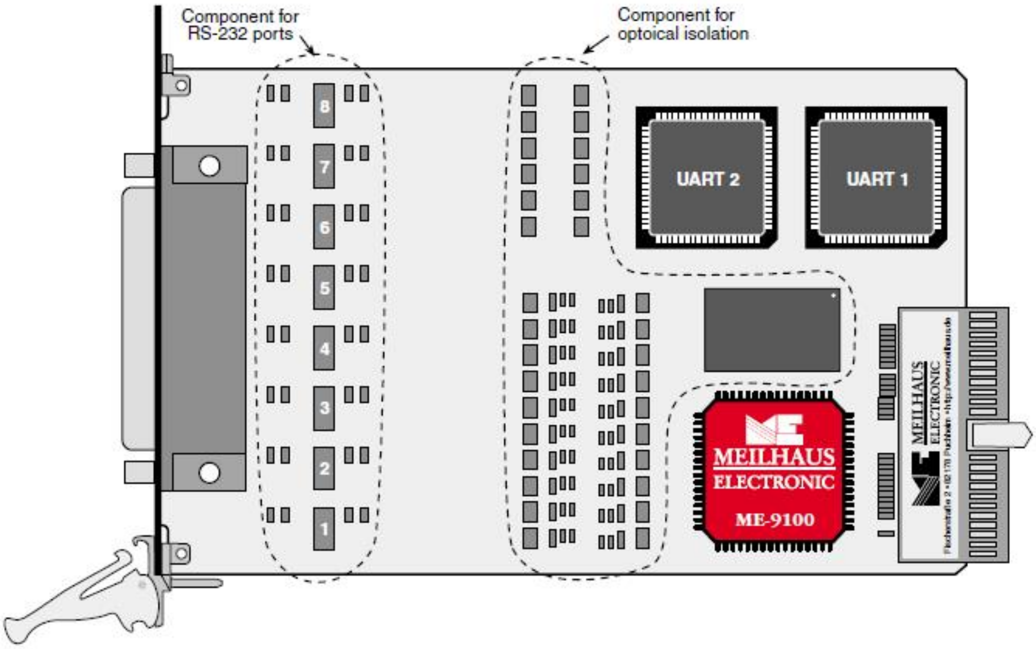


Picture 22: ME-9100 PCI (top view)

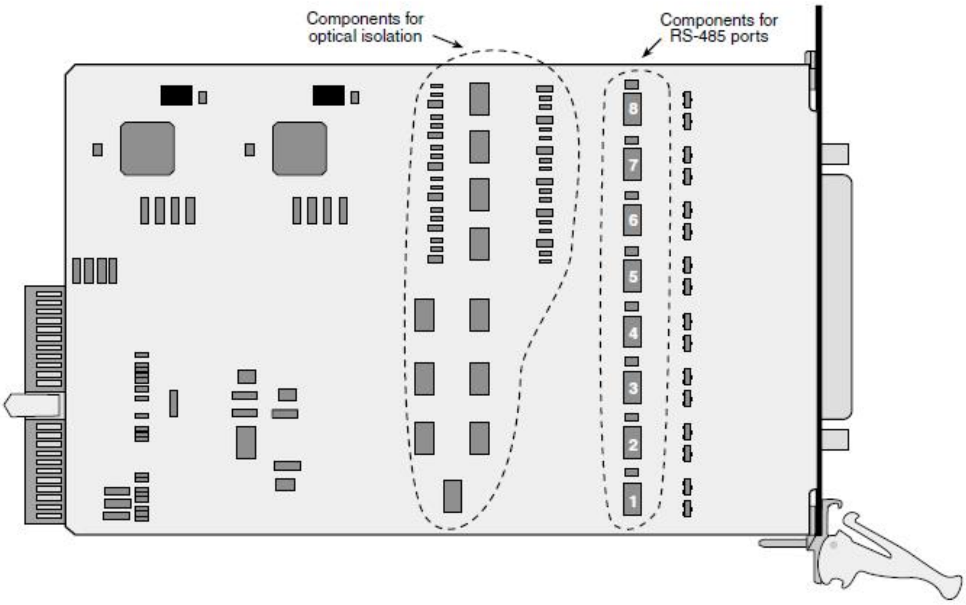


Picture 23: ME-9100 PCI (bottom view)

3.5.3 ME-9100 CompactPCI

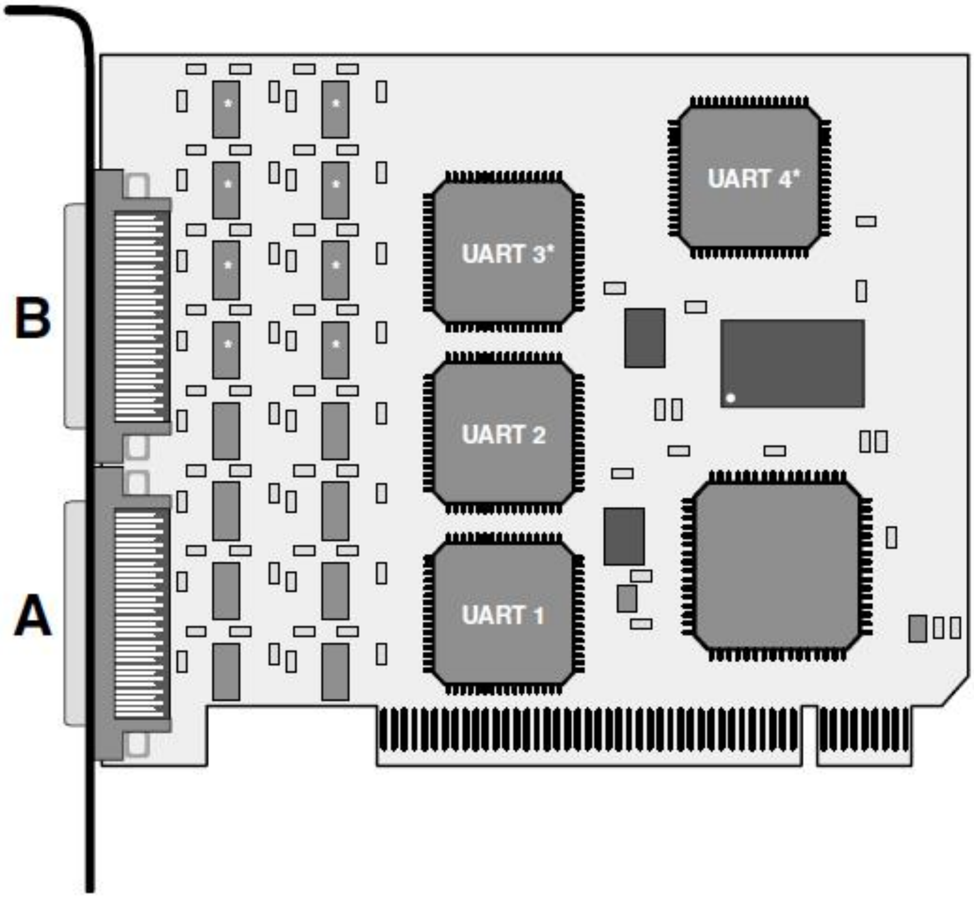


Picture 24: ME-9100 CompactPCI (top view)



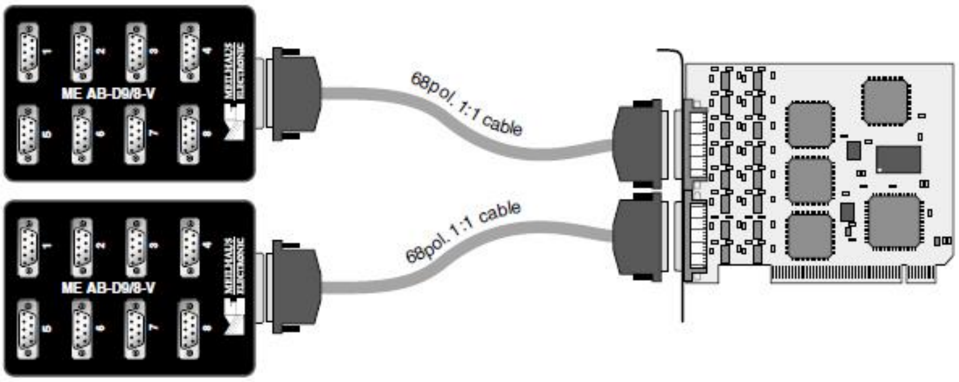
Picture 25: ME-9100 CompactPCI (bottom view)

3.5.4 ME-9300 PCI



Picture 26: ME-9300 PCI

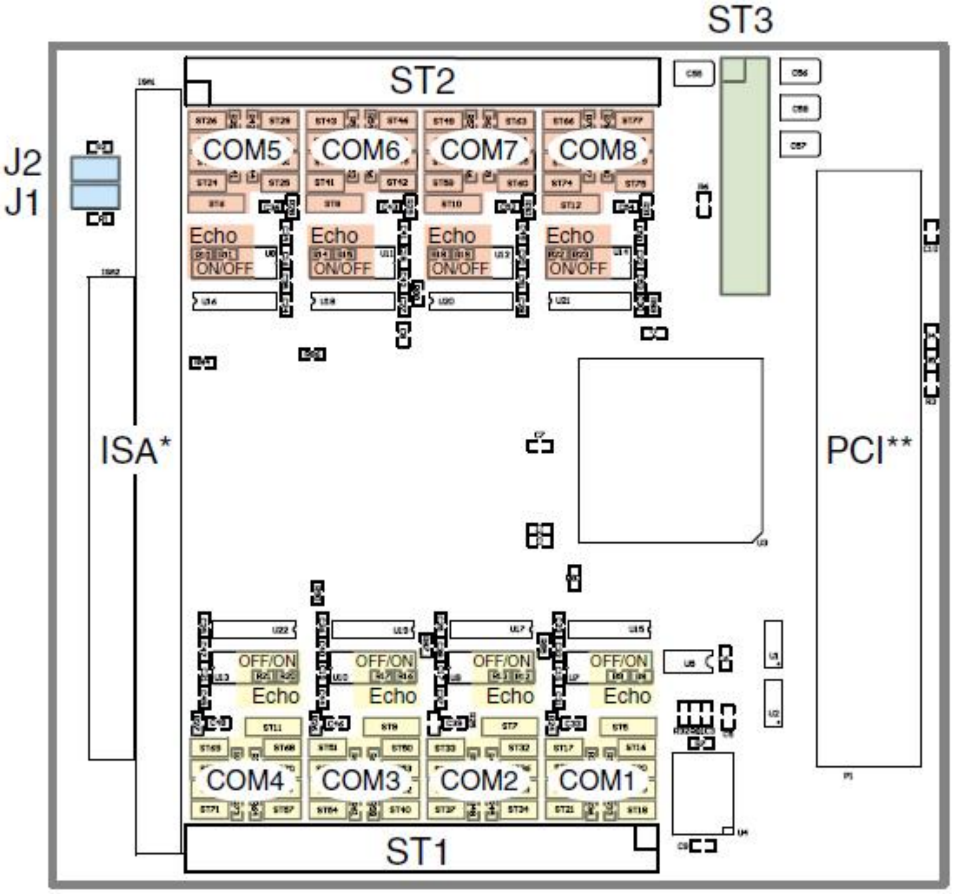
Depending on the number of ports you need one terminal panel for the ME-9300/8 resp. two for the ME-9300/16. They are provided with either 9-pin D-Sub male connectors or 8-pin RJ-45 female connectors (RS-232 Rocket-Port pinout). See also chapter 3.6 Connection Options.



Picture 27: Connection example

3.5.5 ME-90 PC/104-Plus

Picture 28 shows the position of jumpers soldering bridges and connectors.



Picture 28: ME-90 PC/104-Plus Jumper

*ISA bus looped through for other PC/104 boards.

**Automatic level adaption to 3.3 V or 5 V PCI bus (Universal-PCI).

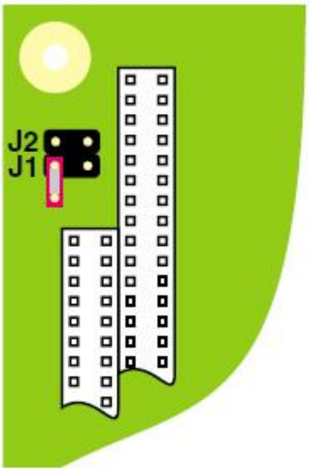
3.5.5.1 Multi-I/O Port

On the ME-90 PC/104-Plus the multi-I/O port of the EXAR XR17D158 chip can be used for customer-specific extensions. The 20-pin IDC connector (ST3) provides VCC (+5 V), PC ground (GND_PC), 8 multi-I/O pins (MIO_x) and the clock input TMRCK. All the other pads are not connected (n.c.). The pinout of ST3 can be found in appendix B6 on page 111. A detailed description of the chip can be found in the datasheet of the manufacturer under www.exar.com.

Attention: VCC at ST3 must not exceed 300 mA!

3.5.5.2 Assignment of the PCI Slots

Because of the „stack architecture“ of PC/104-Plus the signals for all extension boards are routed via one connector (in opposite to the single slots of a motherboard). This means for the ME-90 PC/104-Plus, that the signals CLK, IDSEL and INT0# must be routed to the PCI bus in a way that prevents conflicts with other peripheral boards. The jumpers J1 and J2 control this (see Picture 29).



Assignment of the PCI slots:

J1	J2	CLK	IDSEL	INT0#	Slot*
0	0	CLK0	IDSEL0	INTA#	1 (default)
1	0	CLK1	IDSEL1	INTB#	2
0	1	CLK2	IDSEL2	INTC#	3
1	1	CLK3	IDSEL3	INTD#	4

*Up to 4 boards can be set per "slot".

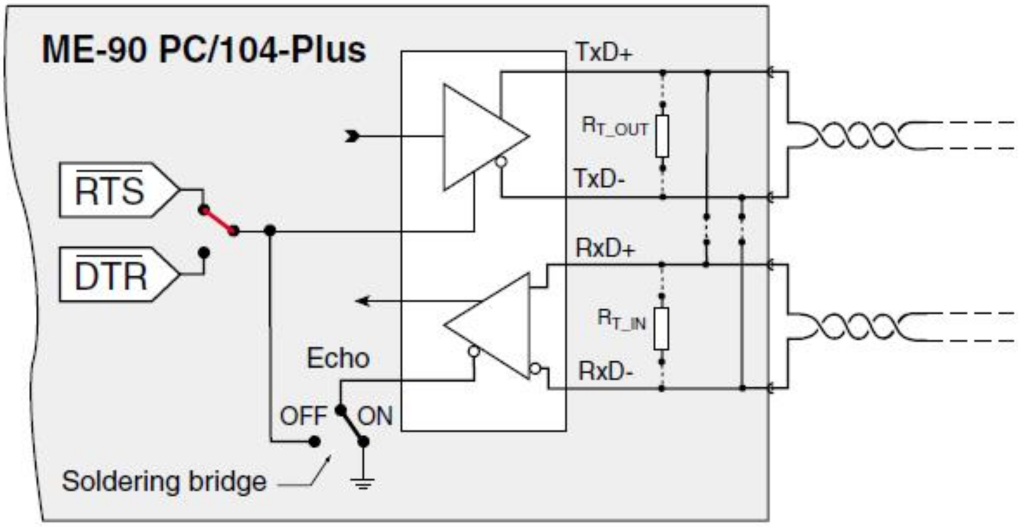
Picture 29: Assignment of PCI slots ME-90 PC/104-Plus

3.5.5.3 Configuration of RS-422/485 Ports

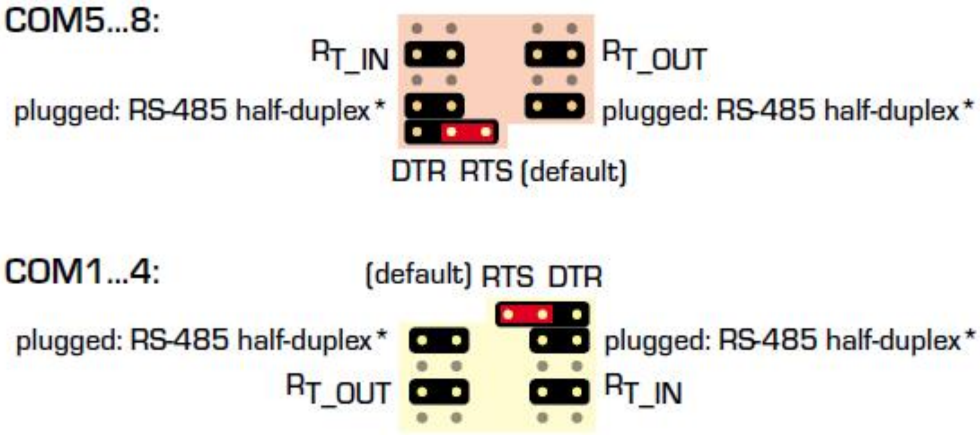
On the RS-422/485 models of the ME-90 PC/104-Plus series the port configuration is done individually by the jumpers shown in picture 31. The assignment to the single ports is shown in picture 28. You have the following configuration possibilities:

1. **Termination** of the input and output buffer with the resistors RT_IN and RT_OUT (default: 120 Ω).
2. **Operation mode RS-485 half-duplex** (see picture 50 on page 58). The connection between TxD+ and RxD+ as well as TxD- and RxD- can be bridged easily by jumpers on the board.
3. **Control of transmitter and receiver change-over** alternatively by the RTS (default) or DTR signal. Note that one of both settings must always be plugged.

Echo ON/OFF see chapter 3.5.5.4 Echo ON/OFF, page 45



Picture 30: Configuration RS-422/485 ports ME-90 PC/104-Plus


COM5...8:
RT_IN
RT_OUT
plugged: RS-485 half-duplex*
plugged: RS-485 half-duplex*
DTR RTS (default)
COM1...4:
(default) RTS DTR
plugged: RS-485 half-duplex*
plugged: RS-485 half-duplex*
RT_OUT
RT_IN

Picture 31: Jumper for port configuration ME-90 PC/104-Plus

Please note that the placement of the soldering bridges of ports COM5...8 is rotated by 180° compared to COM1...4.

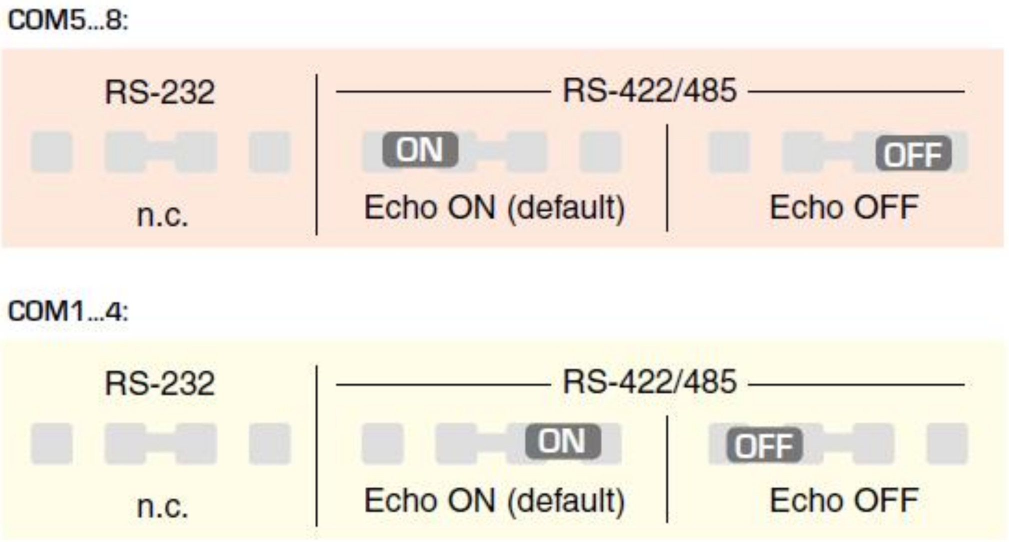
*On demand, these jumpers can be plugged in the operation mode **„RS-485 half-duplex“.** They enable a comfortable connection of RxD and TxD signal without external bridging. In all other operation modes these jumpers are unplugged.

3.5.5.4 Echo ON/OFF

On the RS-422/485 models of the ME-90 PC/104-Plus you can determine the receiver's control by soldering bridges at the top of the board (see Picture 28). With that you can adapt the single ports to the needs of your application individually. See also description of the operation modes from page 56 up.

- **Echo ON**: Set the soldering bridge „ON“ and the receiver listens permanently (default setting).

- **Echo OFF**: Set the soldering bridge „OFF“ to control the receiver by the RTS resp. DTR signal. The reception becomes disabled during transmission.



Picture 32: Soldering bridges ME-90 PC/104-Plus

Please note that the placement of the soldering bridges of ports COM5…8 is rotated by 180° compared to COM1…4.

The position of the soldering bridges can be found with picture 28. Alternatively you can send your board to our service department. We will configure your board for your needs.

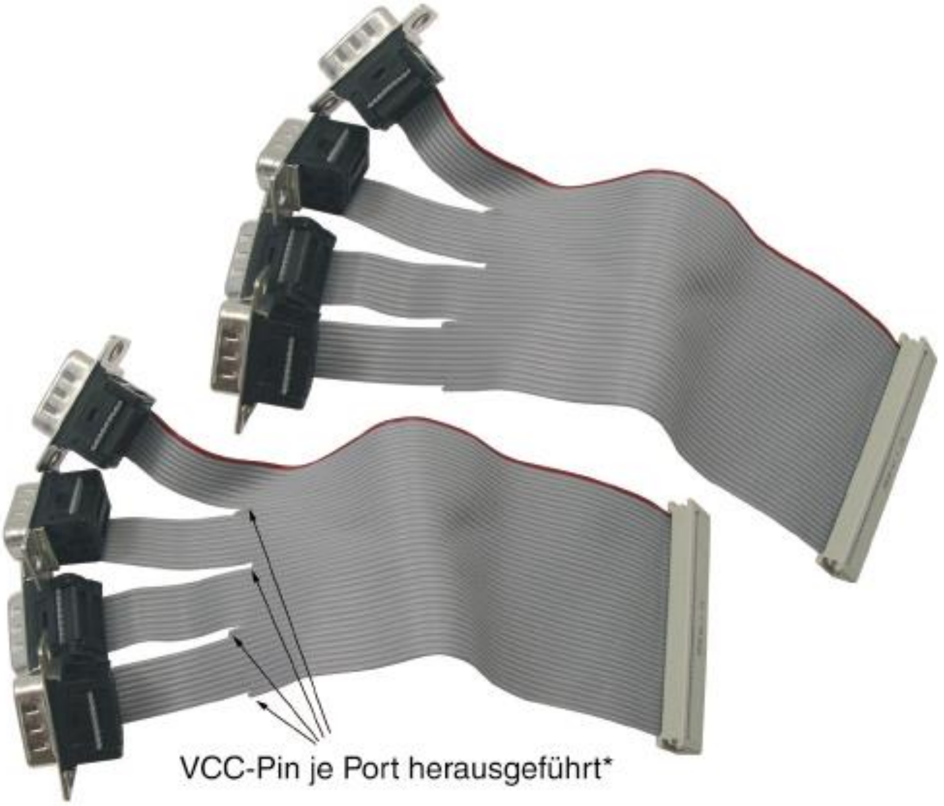
3.6 Connection Options

3.6.1 Dual/Quad/Octopus Cables

With the ME-9000 and ME-9100 a dual, quad or octopus cable is included with the board depending on the number of ports. The pinout is defined as a standard, see B Pinout in Appendix B.

3.6.2 Flat Ribbon Cable to 9-pin D-Sub Connectors

Two flat ribbon cables each with 4 x 9-pin D-Sub male connectors (ME-AK 4D9M) are included with the ME-90 PC/104-Plus. The pinout for RS-232 and RS-422/485 can be found on page 104, 9-pin D-Sub male connector of ME-90 PC/104-Plus. With RS-232 ports VCC (+5 V) is attached to an open end of the flat ribbon cable.

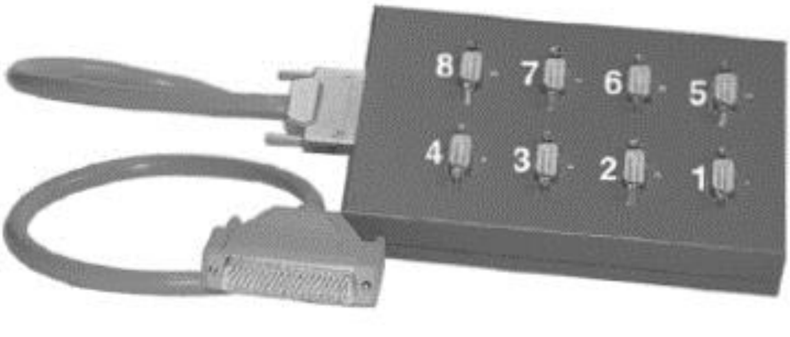


Picture 33: Flat ribbon cable to 4 x 9-pin D-Sub male connector

3.6.3 Terminal Panel for ME-9000/9100

- **ME AB-D9/8-78 + ME AK-D78/1:**

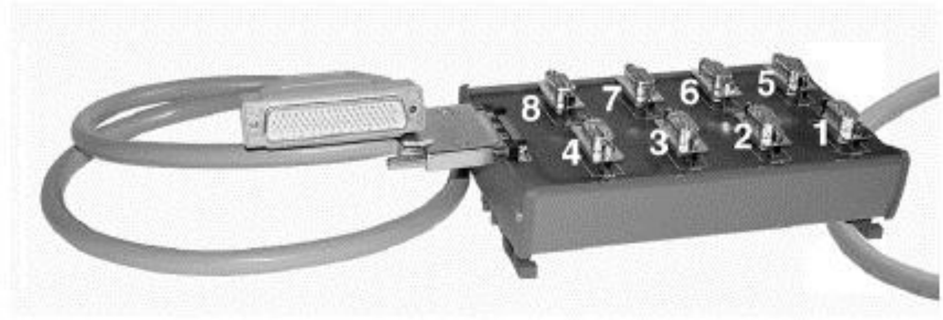
Octo terminal panel in a closed box from 78-pin D-Sub male connector to 8 x 9-pin D-Sub male connectors. Suitable connection cable ME AK-D78/1 (1 m) from 78-pin D-Sub male connector to 78-pin D-Sub female connector.



Picture 34: ME AB-D9/8-78 with ME AK-D78/1 (diagram similar)

- **ME AB-D9/8-78-H + ME AK-D78/1:**

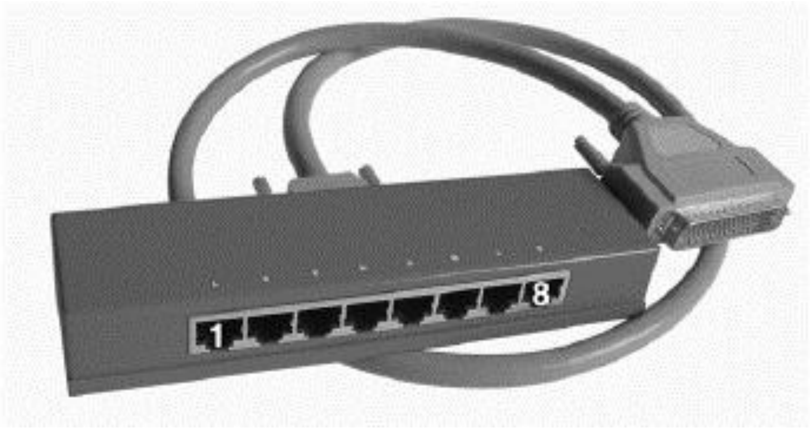
Octo terminal panel for DIN rail mounting from 78-pin D-Sub male connector to 8 x 9-pin D-Sub male connectors. Suitable connection cable ME AK-D78/1 (1 m) from 78-pin D-Sub male connector to 78-pin D-Sub female connector.



Picture 35: ME AB-D9/8-78-H with ME AD-D78/1 (diagram similar)

- **ME AB-RJ45/8x8-V + ME AK-D78/VHDCI:**

Octo terminal panel in a closed box from 78-pin D-Sub male connector to 8 x 8-pin RJ-45 female connectors. Suitable connection cable ME AK-D78/VHDCI (1 m) from 78-pin D-Sub male connector to 68-pin VHDCI female connector.



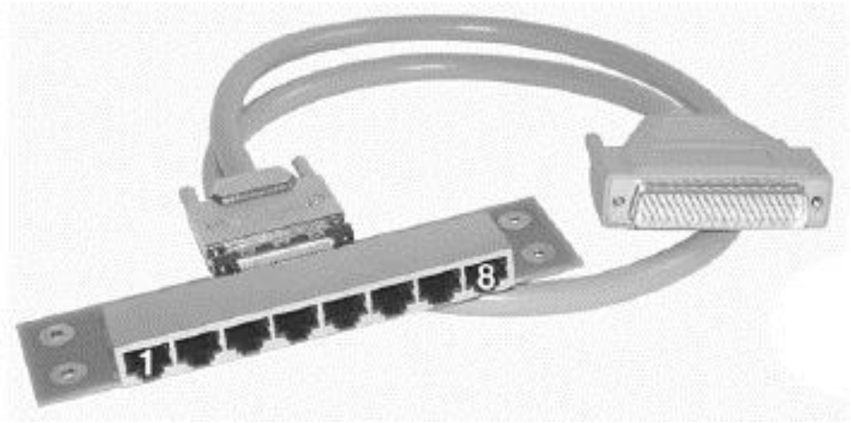
Picture 36: ME AB-RJ45/8x8-V with ME AK-D78/VHDCI

Note: Not suitable for ME-9000p!

- **ME AB-RJ45/8x8-PV + ME AK-D78/VHDCI:**

Octo terminal panel without case for integration in user specific systems from 78-pin D-Sub male connector to 8 x 8-pin RJ-

45 female connectors. Suitable connection cable ME AK-D78/ VHDCI (1 m) from 78-pin D-Sub male connector to 68-pin VHDCI female connector.



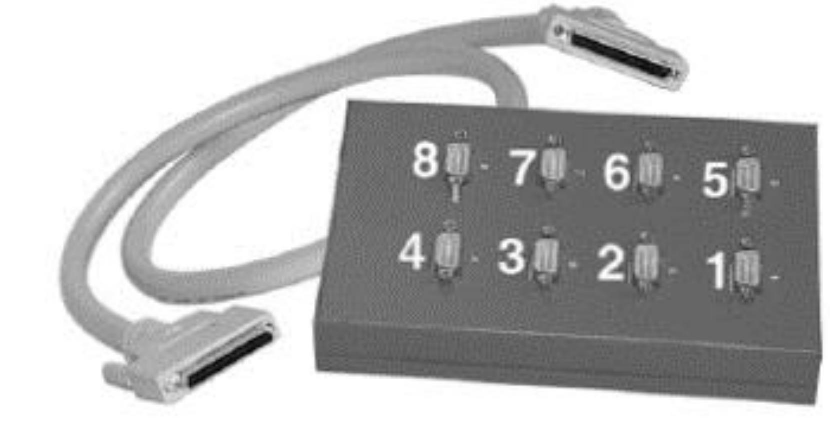
Picture 37: ME AB-RJ45/8x8-PV with ME AK-D78/VHDCI

Note: Not suitable for ME-9000p!

3.6.4 Terminal Panel for ME-9300

- **2 x ME AB-D9/8-V + 2 x VHDCI cable:**

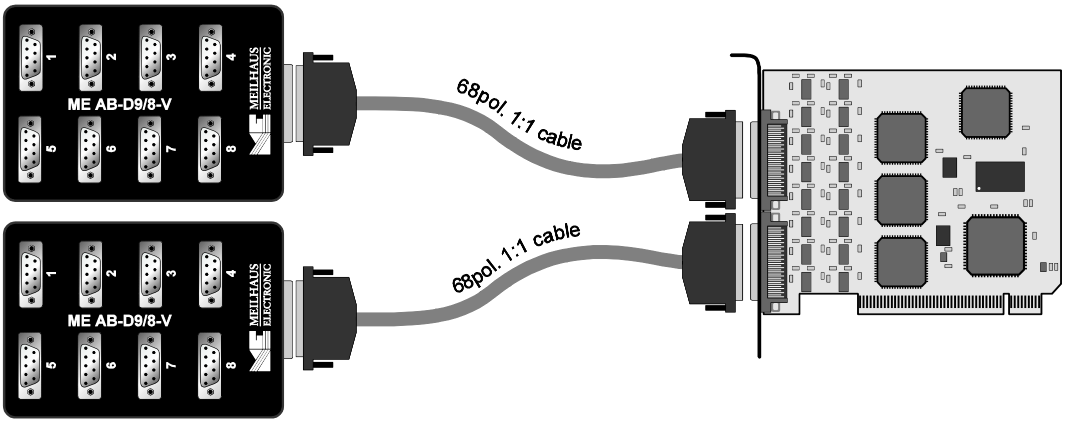
Octo terminal panel in a closed box from 68-pin VHDCI connector to 8 x 9-pin D-Sub male connectors. Connection via 1:1 VHDCI cable (1 m). Terminal panel and cable are required twice.



Picture 38: ME AB-D9/8-V with VHDCI cable

- **2 x ME AB-D9/8-HV + 2 x VHDCI cable:**

Octo terminal panel for DIN rail mounting from 68-pin VHDCI-connector to 8 x 9-pin D-Sub male connectors. Connection via 1:1 VHDCI cable (1 m). Terminal panel and cable are required twice.



Picture 39: ME AB-D9/8-HV with VHDCI cable

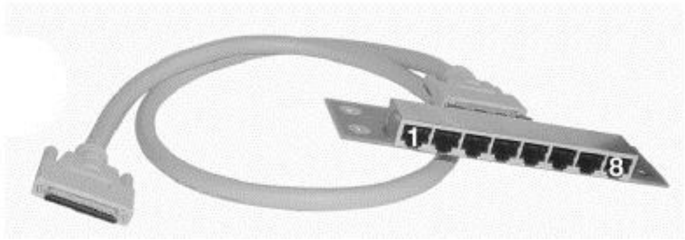
- **2 x ME AB-RJ45/8x8-V + 2 x VHDC cable:**

Octo terminal panel in a closed box from 68-pin VHDCI connector to 8 x 8-pin RJ-45 female connectors. Connection via 1:1 VHDCI cable (1 m). Terminal panel and cable are required twice.

Picture 40: ME AB-RJ45/8x8-V with VHDCI cable

- **2 x ME AB-RJ45/8x8-PV + 2 x VHDCI cable:**

Octo-terminal panel without case for integration in user specific systems from 68-pin VHDCI connector to 8 x 8-pin RJ-45 female connectors. Connection via 1:1 VHDCI cable (1 m). Terminal panel and cable are required twice.



Picture 41: ME AB-RJ45/8x8-PV with ME AK-D78/VHDCI

- **ME AB-D9/16-V + 2 x VHDCI cable:**

16-port terminal panel for 19"-rack-mount from VHDCI connector (2 x 68-pin) to 16 x 9-pin D-Sub male connectors. Connection via two 1:1 VHDCI cable (1 m).

Picture 42: ME AB-D9/16-V

3.7 Operation Modes

Both the RS-232 and the RS-422/485 ports can be accessed like standard COM ports under Windows. The RS-422/485 ports can be used in the operation modes „RS-422“, „RS-485 half-duplex“ and „RS-485 full-duplex“. Each port of the board is configurable independently. The configuration has to be done by the system control (see chap. 2.3 Port Configuration. However, check the transfer parameters in your application program (e. g. Hyper Terminal).

Caution:

Never use more than one driver on an RS-422 bus, and in an RS-485 system do not use an RS-422 driver. The electrical signals are not compatible and may cause damage to the hardware.

3.7.1 The RS-232 Standard

The RS-232 interface used on the ME-9000/9100/9300 and ME-90.

PC/104-Plus is an asynchronous serial standard interface e. g. for data transfer between computers and terminals. A distinction is drawn between DTE (Data Terminal Equipment) and DCE (Data Communication Equipment). Terminals, printers and computers belong to DTEs, modems belong to DCEs. The max. transfer rate depends on the cable length and should not exceed 19,2 kBd for a distance of 15 m.

The following signals are available:

TxD (Transmitted Data): This line carries serial data from the DTE to the corresponding pin on the DCE. The line is held at a negative voltage during periods of line idle.

RxD (Received Data): This line carries serial data from the DCE to the corresponding pin on the DTE.

$\overline{\textbf{RTS}}$ (Request to Send): This line requests data from the DCE.

$\overline{\textbf{CTS}}$ (Clear to Send): When a half-duplex modem is receiving, the DTE keeps RTS inhibited. When it becomes the DTE’s turn to transmit, it advises the modem by asserting the RTS pin. When the modem asserts the CTS, it informs the DTE that it is now safe to send data. The procedure is reversed when switching from transmit to receive.

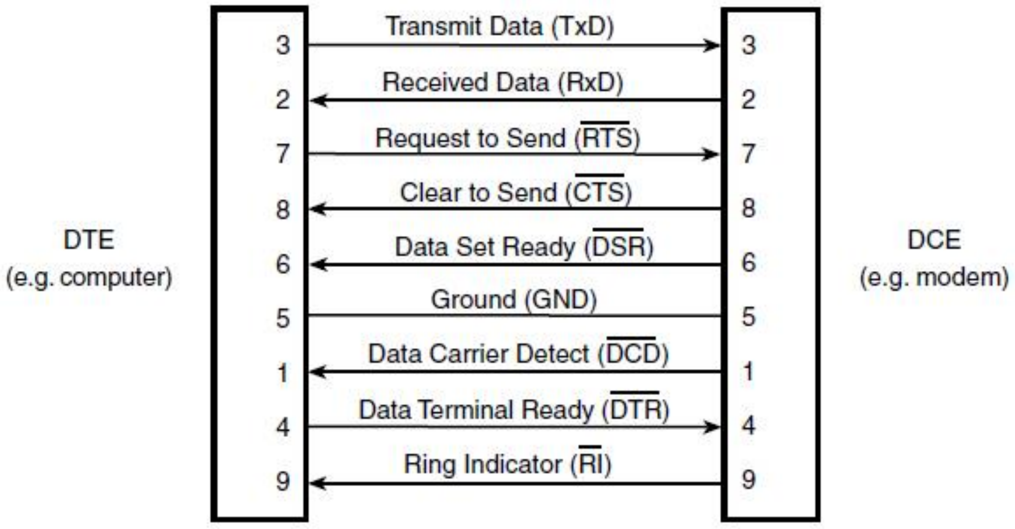
$\overline{\text{DSR}}$ (Data Set Ready): This is also called DCE Ready. In the answer mode, the answer tone and the Data Set Ready are asserted two seconds after the telephone goes off hook.

$\overline{\text{GND}}$ (Signal Ground): This is the common return line for the data Transmit and Receive signals. The connection between the two ends is always made.

DCD (Data Carrier Detect): This is also called the Received Line Signal Detector. It is asserted by the modem when it receives a remote carrier and remains asserted for the duration of the link.

$\overline{\text{DTR}}$ (Data Terminal Ready): This line enables, but does not cause, the modem to switch onto the line. In originate mode, Data Terminal Ready must be asserted in order to auto dial. In answer mode, Data Terminal Ready must be asserted to auto answer.

$\overline{\text{RI}}$ (Ring Indicator): This pin is asserted during a ring on the line.


DTE
(e.g. computer)
DCE
(e.g. modem)
3 Transmit Data (TxD) 3
2 Received Data (RxD) 2
7 Request to Send ($\overline{\text{RTS}}$) 7
8 Clear to Send ($\overline{\text{CTS}}$) 8
6 Data Set Ready ($\overline{\text{DSR}}$) 6
5 Ground (GND) 5
1 Data Carrier Detect ($\overline{\text{DCD}}$) 1
4 Data Terminal Ready ($\overline{\text{DTR}}$) 4
9 Ring Indicator ($\overline{\text{RI}}$) 9

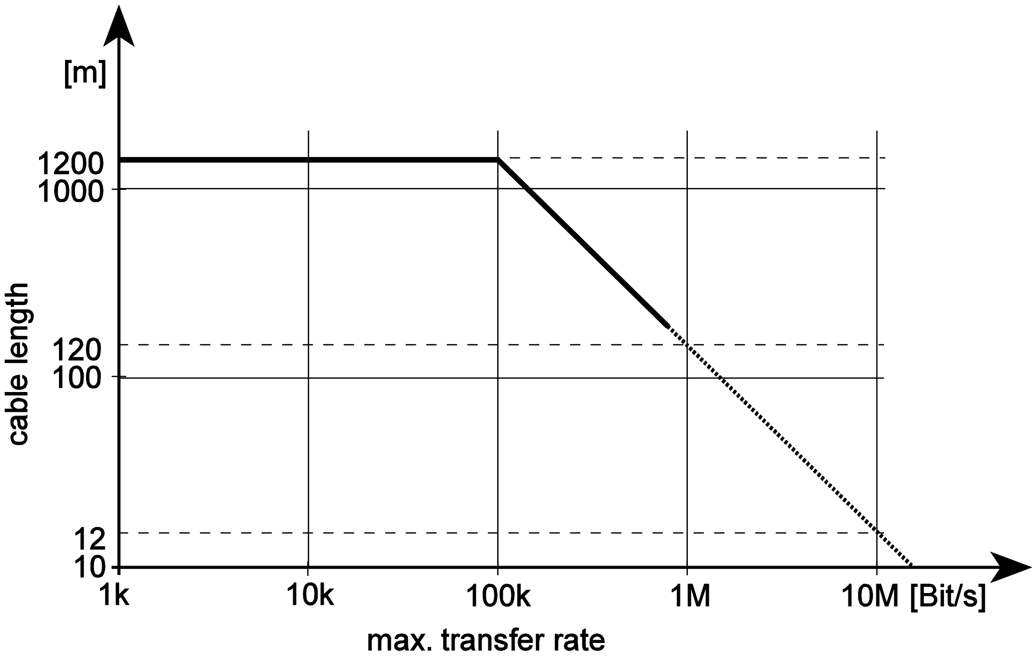
Picture 43: Operation mode RS-232

Note: Keep the signal levels (see appendix “A Specification”).

3.7.2 The RS-422/485 Standard

In comparison to the RS-232 standard the RS-422 and RS-485 standard allow reliable data transfer over greater distances. For optimum noise immunity twisted pair wires will be used. The data transfer is done in a differential way. A voltage difference of more than +200 mV at the receiver input is defined as high-level and a voltage difference of less than -200 mV as low-level. The RS-485 standard is an extension of the RS-422 standard and allows up to 32 transmitters/receivers, which can be in half-duplex or full-duplex mode. Every transmitter/receiver must have a unique address.

Note that the transfer rate decreases with increasing cable length. See the following Picture to calculate the max. transfer rate (max. 921,6 kBd, depending on board version) in dependency of cable length (max 1200 m).



Picture 44: Dependency of cable length to max. transfer rate with terminated lines

Note: Keep the signal levels (see specifications on page 101).

3.7.2.1 RS-422

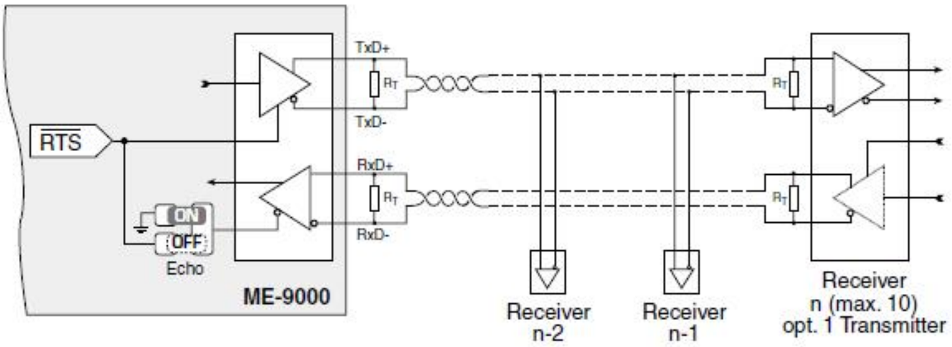
The RS-422 standard provides serial data transfer up to 1200 m by two twisted pair wires for optimum noise immunity. Beside the ME-9000/9100 resp. ME-90 PC/104-Plus maximum one more transmitter and up to 10 receivers are allowed. Usage of termination resistors R_T (50...500 Ω, depending on characteristic resistance) is recommended.

Note:

Under Windows the configuration is done per port in the system settings (see chapter 2.3 Port Configuration). Flow control is handled automatically by the driver.

Under Linux the transmitter must be controlled by the user with the RTS resp. DTR signal as wanted (logical „0“ sets the transmitter driving).

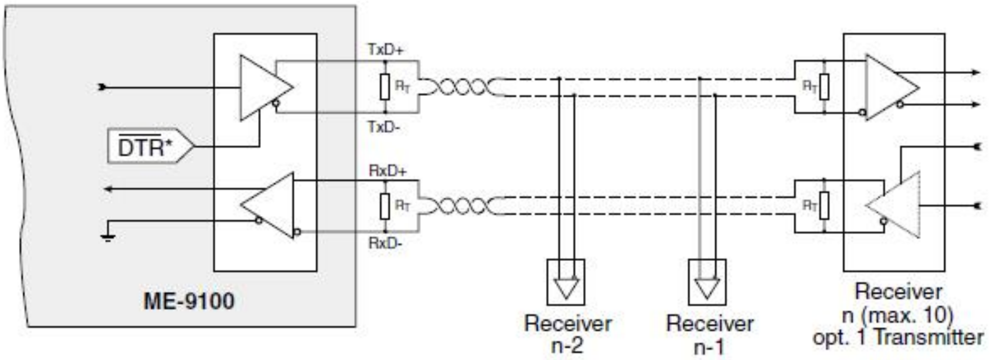
3.7.2.1.1 ME-9000



Picture 45: Operation mode "RS-422" ME-9000

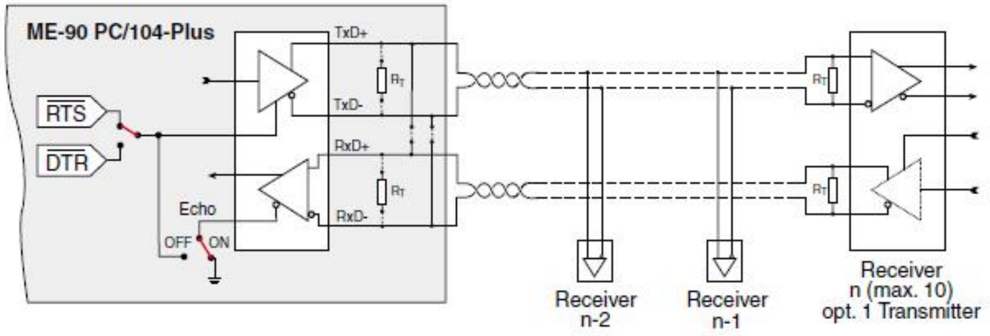
Controlling the receiver (echo ON/OFF) can be adapted by a soldering bridge to the requirements or your application if necessary (see also page 33). Factory setting: echo ON

3.7.2.1.2 ME-9100



Picture 46: Operation mode "RS-422" ME_9100

3.7.2.1.3 ME-90 PC/104-Plus



Picture 47: Operation mode "RS-422" ME-90 PC/104-Plus

Flow control can be changed from RTS (default) to DTR signal by jumper (see picture 31 on page 45). Controlling the receiver (echo ON/OFF) can be adapted by a soldering bridge to the requirements or your application if necessary (see picture 32 on page 45). Factory setting: echo ON. The termination resistors RT have 120 Ω and are not jumpered by default.

3.7.2.2 RS-485 half-duplex

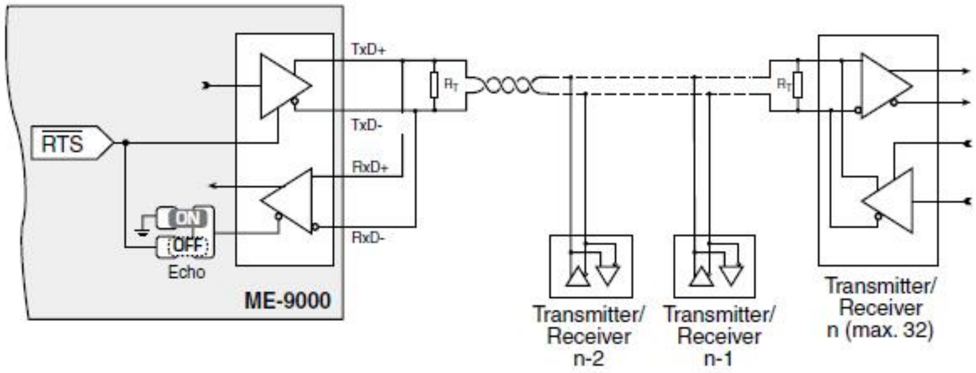
In „RS-485 half-duplex“ operation mode, data can be **either** sent or received over one twisted-pair wire. This feature is provided by the ability of the line drivers to be set in high-impedance state besides low and high. Note that only one transmitter can be active at any given time. Up to 32 transmitters are allowed and the cable length should not exceed 1200 m. Usage of termination resistors R_T(50…500 Ω, depending on characteristic resistance) is recommended.

Note:

Under Windows the configuration is done per port in the system settings (see chapter 2.3 "Port Configuration"). Flow control is handled automatically by the driver.

Under Linux the transmitter must be controlled by the user with the RTS resp. DTR signal as wanted (logical „0“ sets the transmitter driving).

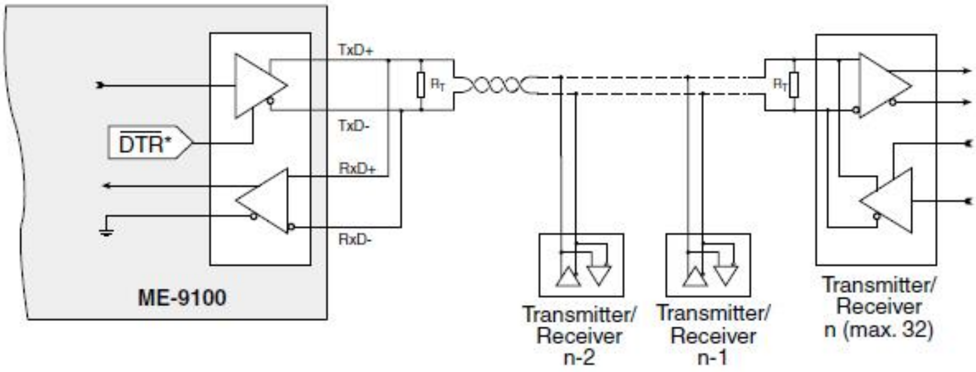
3.7.2.2.1 ME-9000


TxD+
TxD-
RxD+
RxD-
RTS
ON
OFF
Echo
ME-9000
R_T
Transmitter/ Receiver n-2
Transmitter/ Receiver n-1
Transmitter/ Receiver n (max. 32)

Picture 48: Operation mode "RS-485 half-duplex" ME-9000

Controlling the receiver (echo ON/OFF) can be adapted by a soldering bridge to the requirements of your application if necessary (see also page 33). Factory setting: echo ON.

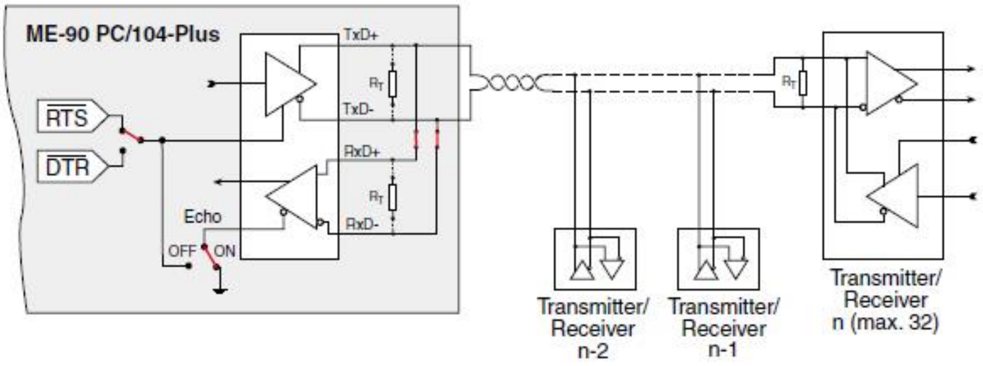
3.7.2.2.2 ME-9100


TxD+
TxD-
RxD+
RxD-
DTR*
ME-9100
R_T
Transmitter/ Receiver n-2
Transmitter/ Receiver n-1
Transmitter/ Receiver n (max. 32)

Picture 49: Operation mode "RS-485 half-duplex" ME-9100

Depending on your needs you have the possibility to remove the echo of the transmitter in the system settings (see chapter 2.3 Port Configuration) of Windows.

3.7.2.2.3 ME-90 PC/104-Plus


ME-90 PC/104-Plus
RTS
DTR
Echo
OFF
ON
TxD+
TxD-
RxD+
RxD-
R_T
Transmitter/ Receiver n-2
Transmitter/ Receiver n-1
Transmitter/ Receiver n (max. 32)

Picture 50: Operation mode "RS-485 half-duplex" ME-90 PC/104-Plus

Flow control can be changed from RTS (default) to DTR signal by jumper (see picture 31 on page 45). Controlling the receiver (echo ON/OFF) can be adapted by a soldering bridge to the requirements or your application if necessary (see picture 32 on page 46). Factory setting: echo ON. The termination resistors R_T have 120 Ω and are not jumpered by default.

In this operation mode it is easy to connect the signals TxD+ with RxD+ and TxD- with RxD- by the appropriate jumpers (see picture 31 on page 45) – no external bridging required.

3.7.2.3 RS-485 full duplex

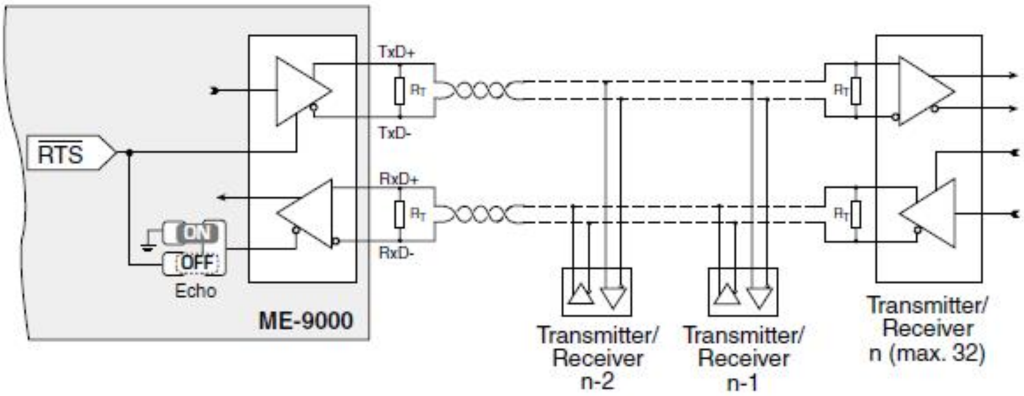
In „RS-485 full-duplex“ operation mode, data can be sent and received over two twisted pair wires at the same time. Up to 32 transmitters are allowed and the cable length should not exceed 1200 m. Usage of termination resistors R_T (50…500 Ω, depending on characteristic resistance) is recommended.

Note:

Under Windows the configuration is done per port in the system settings (see chapter 2.3 Port Configuration). Flow control is handled automatically by the driver.

Under Linux the transmitter must be controlled by the user with the $\overline{\text{RTS}}$ resp. $\overline{\text{DTR}}$ signal as wanted (logical „0“ sets the transmitter driving).

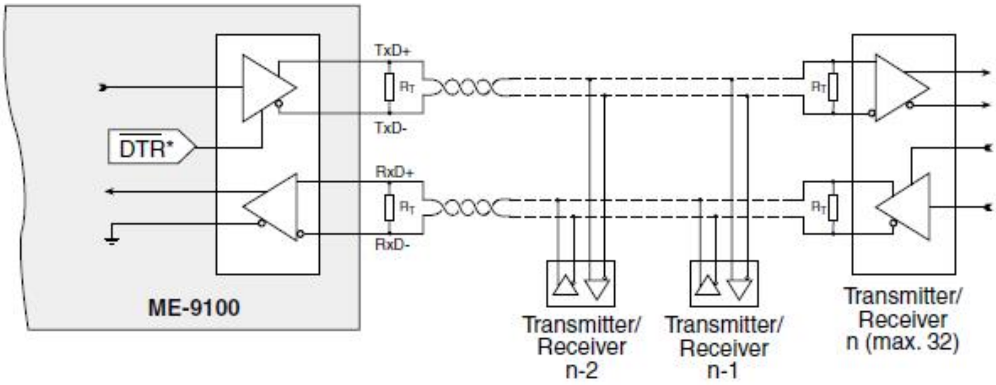
3.7.2.3.1 ME-9000



Picture 51: Operation mode "RS-485 full-duplex" ME-9000

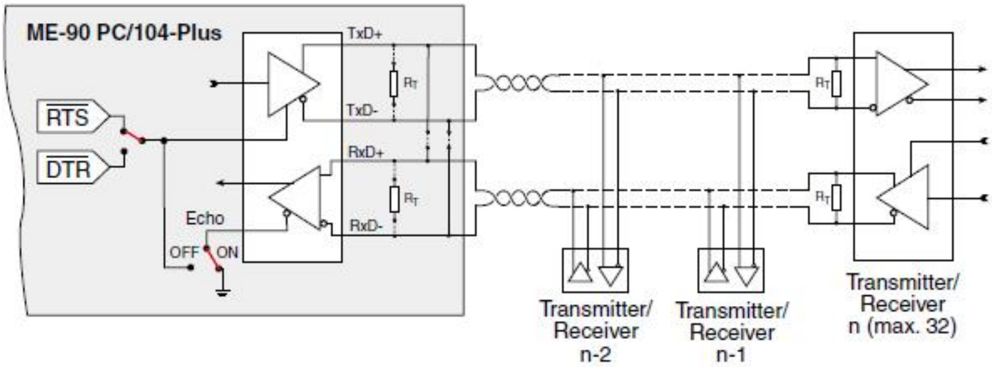
Controlling the receiver (echo ON/OFF) can be adapted by a soldering bridge to the requirements of your application if necessary (see also page 33). Factory setting: echo ON.

3.7.2.3.2 ME-9100



Picture 52: Operation mode "RS-485 full-duplex" ME-9100

3.7.2.3.3 ME-90 PC/104-Plus



Picture 53: Operation mode "RS-485 full-duplex" ME-90 PC/104-Plus

Flow control can be changed from $\overline{\text{RTS}}$ (default) to $\overline{\text{DTR}}$ signal by jumper (see picture 31 on page 44). Controlling the receiver (echo ON/OFF) can be adapted by a soldering bridge to the requirements or your application if necessary (see picture 32 on page 45). Factory setting: echo ON. The termination resistors R_T have 120 Ω and are not jumpered by default.

4 Programming

For programming the serial boards of type ME-9000/9100/9300 and ME-90 PC/104-Plus no board specific programming interfaces are required. The ports can be accessed as COM ports by the standard API under Windows.

For programming the digital I/O and counter section of the ME-9000 and ME-90 PC/104-Plus a special ME-9000 multi-I/O driver is available for Windows 98/Me/2000/XP and NT4.0. See the following chapters.

4.1 ME-9000 Multi-I/O Driver

The ME-9000 multi-I/O driver (ME9000MIO) provides an extensive function library for programming the digitalI/O lines and the counter of the ME-9000 and ME-90 PC/104-Plus. The function reference can be found in chapter 5 on page 69.

The system driver consists of the following components:

- WDM driver (me9000mio.sys) for
- Windows 98/Me/2000/XP.
- Kernel driver (me9000mio.sys) for Windows NT.
- API-DLL (me9000mio.dll) for Visual C++ and Delphi.
- API-DLL (me9000mio.dll) for Visual Basic.

To make it easy for you we provide simple demo programs and small projects with source code to help understanding of the functions and how to include them into your project. These demo programs can be found within the ME Software Developer Kit (ME-SDK), which is installed to the directory `C:Meilhaus\ me-sdk` by default. Please read the notes in the corresponding README-files.

4.1.1 Visual C++

API-DLL	me9000mio.dll	System driver
Function prototypes	me9000miodll.h	ME-SDK
Constant definitions	me9000miodefs.h	ME-SDK
Function prefix	me9000mio...	

Table 5: Visual C++

Visual C++ support for your board is included with the ME-SDK on CD/DVD or under http://www.meilhaus.de/en/download.

4.1.2 Visual Basic

API-DLL	me9000mioEx.dll	System driver
Function prototypes	me9000mio.bas	ME-SDK
Constant definitions	me9000mio.bas	ME-SDK
Function prefix	me9000mioVB_...	

Table 6: Visual Basic

Visual Basic support for your board is included with the ME-SDK on CD/DVD or under www.meilhaus.de/download.

Important Notes: Partly the function prototypes for Visual Basic differ in the number of parameters and the datatype of single parameters. Please note the file `me9000mio.bas`, included with the ME-SDK. Instead of the standard API `me9000mio.dll` you have to use the specific API `me9000mioEx.dll`. „Missing“-parameters are marked with the symbol „~~**VB**~~“ in the function reference.

Because of the threading model was changed in Visual Basic 6.0, the usage of callback functions is not possible there. However, it is possible in Visual Basic 5.0.

4.1.3 Delphi

API-DLL	me9000mio.dll	System driver
Function prototypes	me9000miodll.pas	ME-SDK
Constant definitions	me9000miodefs.pas	ME-SDK
Function prefix	me9000mio...	

Table 7: Delphi

Delphi support for your board is included with the ME-SDK on CD/DVD or under www.meilhaus.de/download.

4.1.4 Agilent VEE

The serial ports of the ME-9000/9100/9300 and ME-90 PC/104-Plus can be accessed by the menu „I/O Instrument Manager...“. The digital I/O and counter functions are not supported in VEE. For basics of VEE programming please use your VEE documentation and the VEE online help index.

4.1.5 LabVIEW

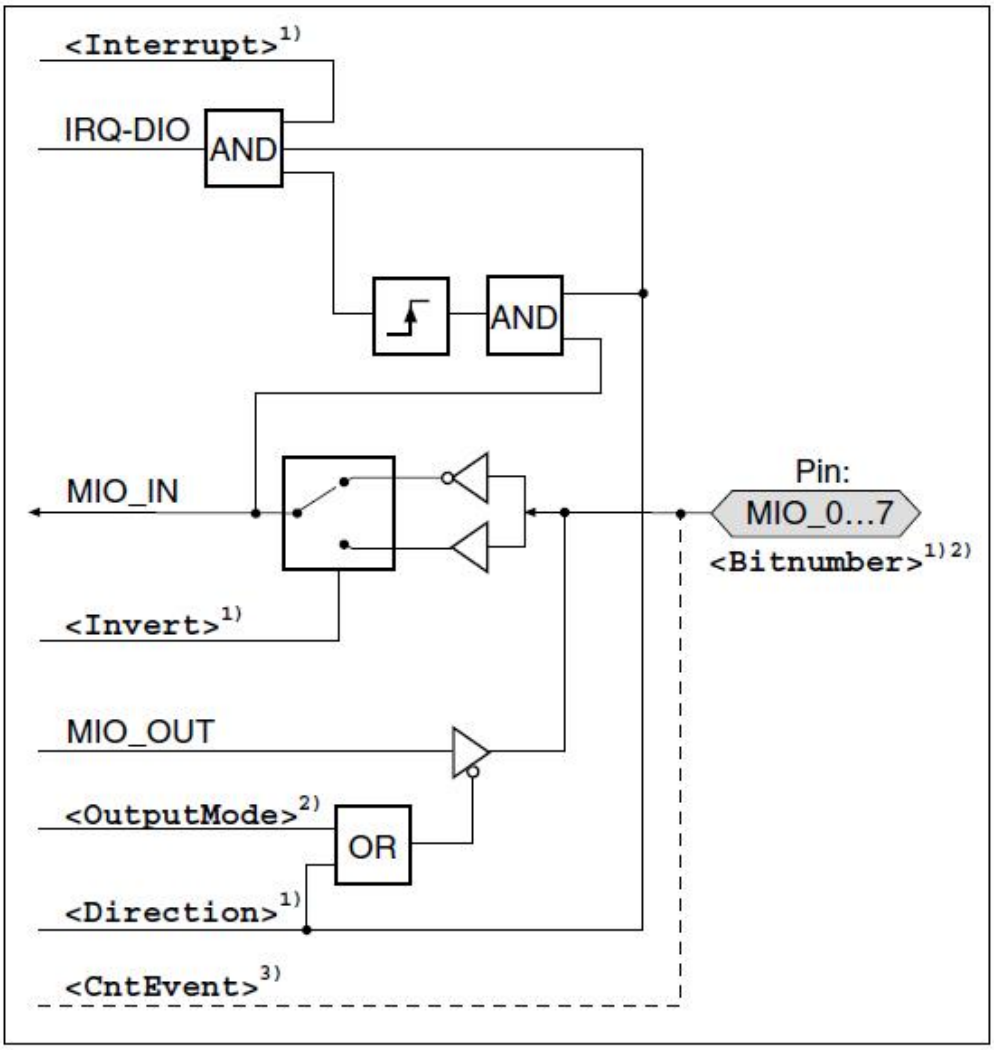
The ports of the ME-9000/9100/9300 and ME-90 PC/104-Plus can be accessed as serial standard interfaces. The digital I/O and counter functions are not supported in VEE. For basics of LabVIEW™ programming please use your LabVIEW™ documentation and the LabVIEW™ online help index.

4.2 Digital-I/O Section

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

The ME-9000 provides 8 bidirectional digital-I/O lines. The multi-I/O bits (MIO_0...7) can be configured very flexible (In/Out, interrupt function, inverting inputs, high-impedance outputs).

Use the functions...*DIOConfig* and ...*DIOOutpu*t for configuration and note the following picture



Picture 54: Configuration of the multi-I/O bits

Note: A bit configured for output can be read back!

1) Parameter of the function *...DIOConfig*:

- `<BitNumber>` Select the multi-I/O bit wanted.
- `<Direction>` Choose between input or output.
- `<Invert>` The polarity of an input can be inverted.
- `<Interrupt>` Bit becomes an edge triggered interrupt input. Condition is that the bit will be configured as input. The interrupt is directly forwarded to the system. All enabled interrupt bits initiate one common interrupt of the digital-I/O section (IRQ-DIO). The single bit cannot be determined as the interrupt source.

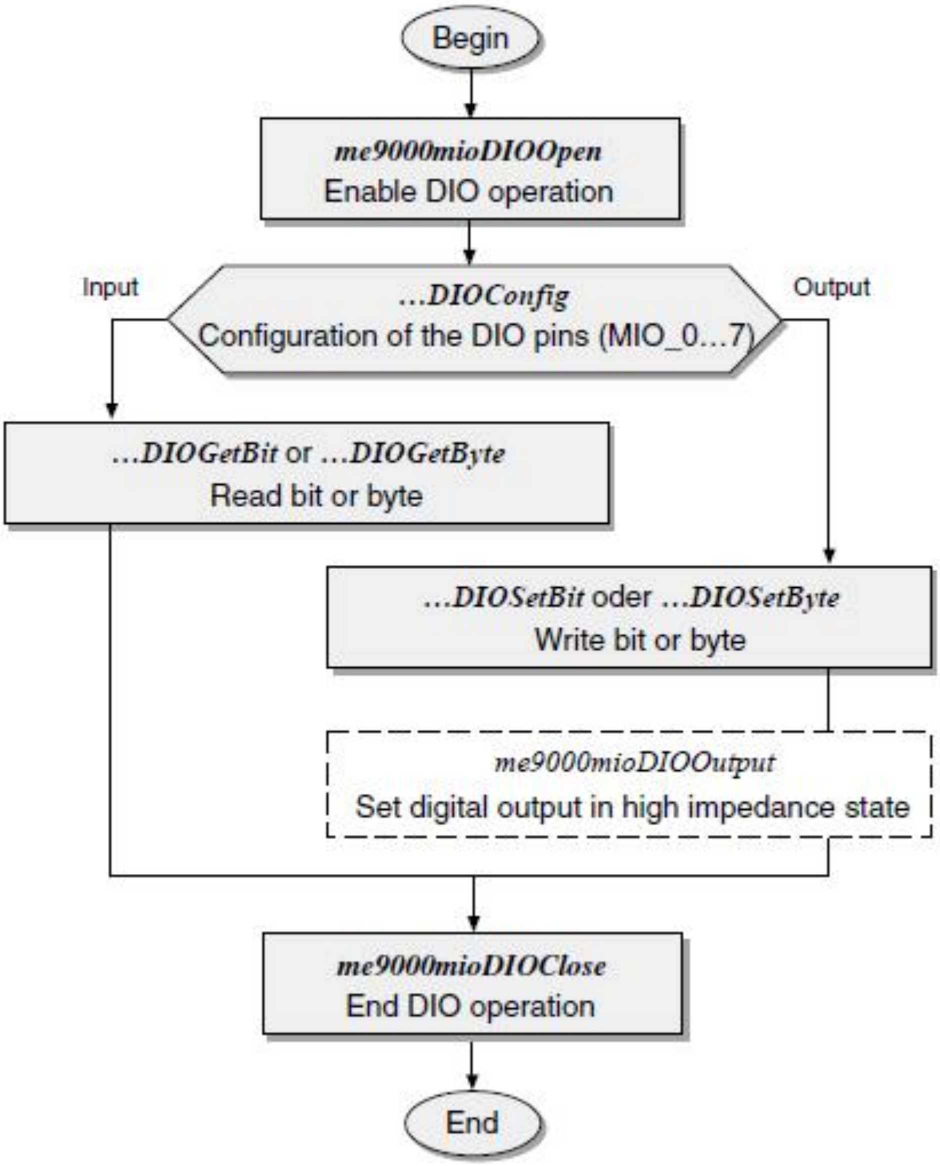
2) Parameter of the function *...DIOOutput:*

- `<OutputMode>` A bit configured for output can also set to a high-impedance state (tristate).

3) Parameter `<CntEvent>` of the function …*CntConfig*:

- **Please note**, in combination with the counter the pin MIO_0 provides a special function. If wanted you can use it to output a counter event. In that case the bit may not be switched as an input because of it becomes an output automatically. If the board is damaged, no guarantee can be given.

The picture below shows the basic program flow:



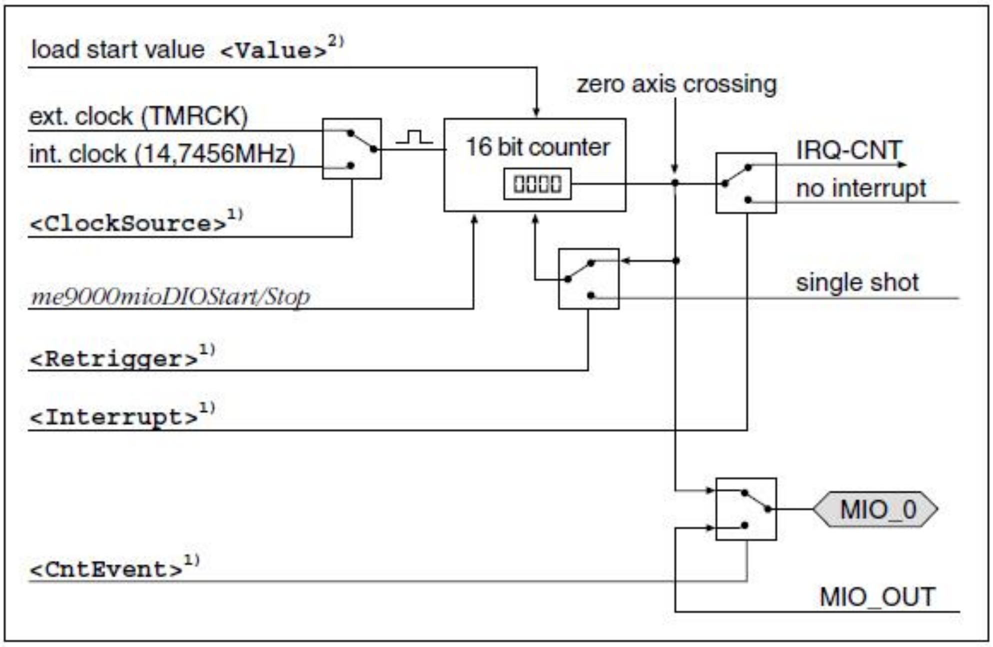
Picture 55: Programming Digital-I/O Section

4.3 Counter

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

The ME-9000 provides a 16-bit counter. A start value must be loaded into the counter by the function ...*CntWrite*. Calling the function ...*CntStart* starts counting down to „0". On zero axis crossing either an interrupt can be triggered and/or the multi-I/ O bit MIO_0 can be driven. You can choose between „single shot" and „retrigger" mode. In „single shot" mode counting will be ended on reaching „0", in the „retrigger" mode the start value will be reloaded into the counter. In that case the counting operation will be ended by the function...*CntStop*.

For configuration use the functions ...CntConfig and ...CntWrite and note the following Picture:



Picture 56: Configuration counter

1) Parameter of the function ...CntConfig:

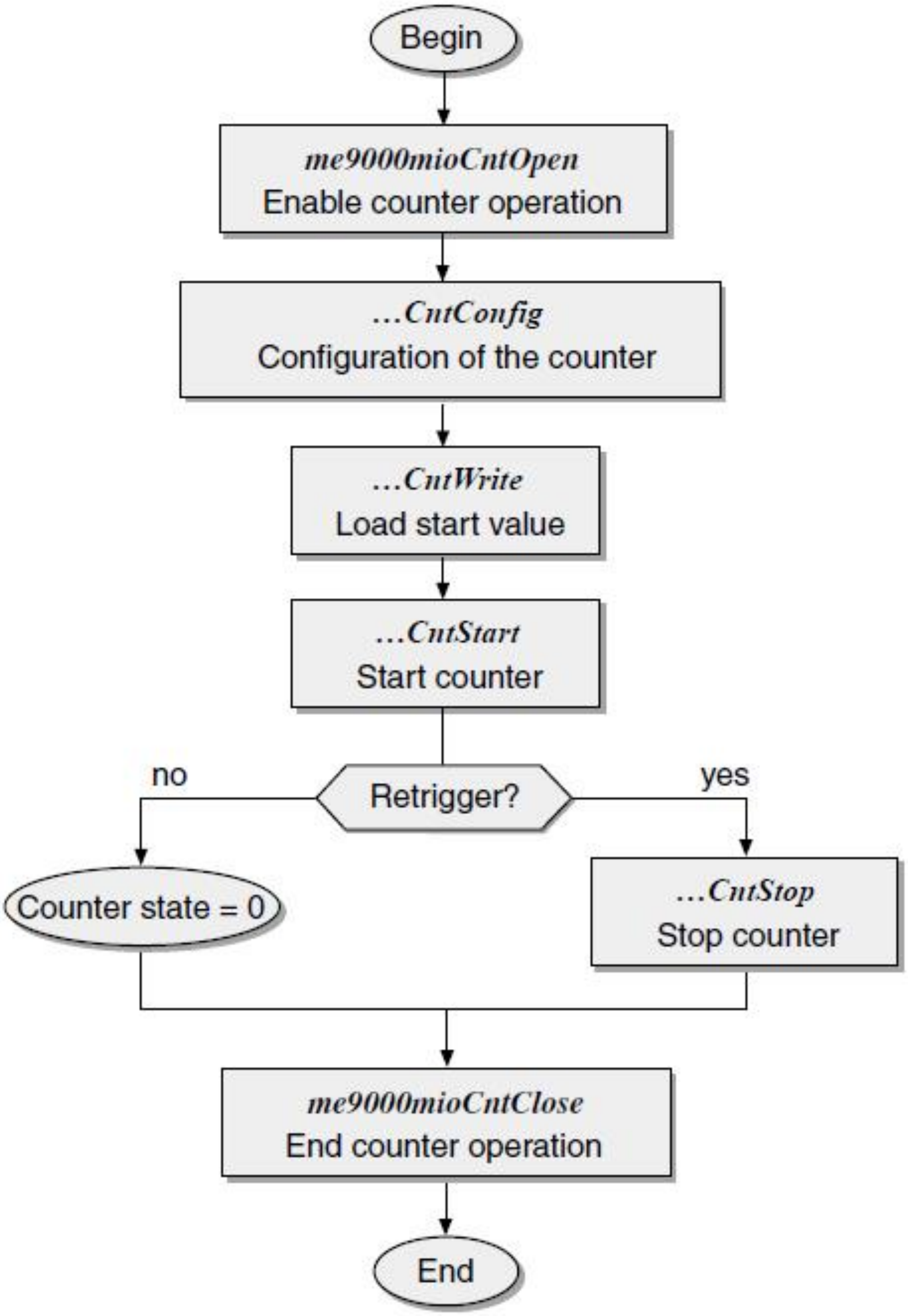
- `<Retrigger>` Choose between „Single Shot" (counting once to „0") or „Retrigger" (reloading the start value on zero axis crossing).
- `<CntEvent>` Enable this option to use the multi-I/O pin

- MIO_0 to output a counter event (on zero axis crossing). **Please note**: In that case the pin MIO_0 may not be switched as an input because the pin becomes an output automatically. If the board will be damaged no guarantee can be given!
- <ClockSource> Choose between internal clock (14,7456MHz) or external clock by pin TMRCK.
- <Interrupt> On zero axis crossing the counter interrupt (IRQ-CNT) will be triggered. The interrupt is directly forwarded to the system.

2) Parameter of the function ...CntWrite:

- <Value> A 16-bit start value will be loaded into the counter.

The following picture shows the basic program flow:



Picture 57: Programming the counter

5 Function Reference

5.1 General Notes

- **Function Prototypes:**

In the following function description the generic function prototypes for Visual C++ are used. The definitions for other supported programming languages which are partly using different data types can be found in the appropriate definition files included with the ME-SDK.

- **Parameter „Board Number“**

When using a single board of a board family, the board number is always „0“ (integer value). In systems running several boards from the same board family the computer assigns the board number. Use this number to access the board. Determine the assignment of board numbers after installation of the boards.

Tip: Verify the assignment of „Board Number“ and serial number at the beginning of your program (see function *Get- Serial Number*).

5.2 Naming Conventions

The API functions of the “ME9000MIO” function library concern boards of type ME-9000 with hardware revision 1.4 or higher and for ME-90 PC/104-Plus. The function names consist of the prefix “*me9000mi”* and several components representing the respective function as descriptive as possible. (e.g. “DIO” for “Digital I/O”.

For Visual C++ and Delphi no language specific identification is given, i.e. the function name follows the prefix directly e.g. *me9000mioDIOConfig.* However for Agilent VEE the characters “VEE_“ (e.g. *me9000mioVEE_DIOConfig*), for LabVIEW the characters „“LV_“ (e.g. *me9000mioLV_DIOConfig*) and for Visual Basic the characters „VB_“ (e.g. *me9000mioVB_DIOConfig*) are inserted.

For the description of the functions, the following standards will be used:

function name will be italic in body text e.g. *me9000mioGetDLLVersion*.

<parameters> will be in brackets as shown and in font Courier.

[square brackets] will indicate physical units.

`main(...)` parts of programs will be in `Courier` type

5.3 Description of the API Functions

The functions will be described by functional groups as listed below. Within each functional group the individual functions will be described in alphabetical order:

„5.3.1 “Error Handling“ on page 72

„5.3.2 “General Functions“ on page 76

„5.3.3 “Digital Input/Output“ on page 80

„5.3.4 “Counter Functions“ on page 91

Function	Short Description	Page
Error Handling		
me9000mioErrorGetMessage	Assign error string to an error number	72
me9000mioErrorGetLastMessage	Assign error string to the last error occurred	73
me9000mioErrorSetDefaultProc	Install predefined global error routine for API	73
me9000mioErrorSetUserProc	Install user defined global error routine for API	74
General Functions		
me9000mioGetBoardCount	Number of boards of type ME-9000 installed in the system	76
me9000mioGetBoardVersion	Determine device ID	76

me9000mioGetDLLVersion	Determine DLL version number	77
me9000mioGetDriverVersion	Determine driver version number	78
me9000mioGetSerialNumber	Determine serial number	79
Digital Input/Output		
me9000mioDIOClose	Close the „Digital-I/O“ operation	80
me9000mioDIOConfig	Configuring the digital-I/O lines	80
me9000mioDIOGetBit	Getting one bit	82
me9000mioDIOGetByte	Getting a byte	83
me9000mioDIOGetIrqCount	Determine number of IRQs	84
me9000mioDIOOpen	Open „Digital-I/O“ operation	85

Table 8: Overview library function

Function	Short Description	Page
me9000mioDIOOutput	Set output in a high-impedance state	86
me9000mioDIOReset	Reset the digital-I/O section	87
me9000mioDIOSetBit	Setting one bit	88
me9000mioDIOSetByte	Setting a byte	89
me9000mioDIOSetCallback	Install callback function	90
Counter Functions		
me9000mioCntClose	Close counter operation	91
me9000mioCntConfig	Configuring the counter	92
me9000mioCntGetIrqCount	Determine number of IRQs	94
me9000mioCntOpen	Open counter operation	95
me9000mioCntRead	Read back start value	95
me9000mioCntReset	Reset counter	96
me9000mioCntStart	Start counter	97
me9000mioCntStop	Stop counter	98
me9000mioCntWrite	Write start value into the counter	98

Table 9: Overview library function

5.3.1 Error Handling

me9000mioErrorGetMessage

✎**Description**

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

This function can be used to determine the error text from an error number returned from the API functions.

● **Definitions**

VC: me9000mioErrorGetMessage (int iErrorCode, char* pcBuffer, unsigned int uiBufferSize);

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ **Parameter**

`<ErrorCode>`

The error number caused by the API function

`<Buffer>`

Pointer to the error description text.

`<BufferSize>`

Buffer size in bytes for the error description text (max. 256 characters)

❬**Return value**

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioErrorGetLastMessage

✎Description

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

This function returns the last error caused by a „me9000mio...“ API function and retrieves the error description text.

● Definitions

VC: me9000mioErrorGetMessage (int iErrorCode, char* pcBuffer, unsigned int uiBufferSize);

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ Parameter

`<Buffer>`

Pointer to the error description text.

`<BufferSize>`

Buffer size in bytes for the error description text (max. 256 characters)

〈Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioErrorSetDefaultProc

✎Description

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

This function can be used to install a predefined global error routine for the entire API. This global error routine is automatically called if an API function call returns an error. The following information is re- turned in the form of a message box:

- Name of the function that returned an error
- Short error description
- Error code

☞ **Note**:

Only one global error routine can be installed (...ErrorSetDefaultProc or ...ErrorSetUserProc)

● **Definitions**

VC: me9000mioErrorSetDefaultProc (int iDefaultProcStatus);

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ **Parameter**

`<DefaultProcStatus>`

ME9000MIO_ERROR_DEFAULT_PROC_ENABLE Installing the predefined error routine.

ME9000MIO_ERROR_DEFAULT_PROC_DISABLE Uninstall the redefined error routine.

Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioErrorSetUserProc

✎**Description**

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

This function is used to install a global user defined error routine for the API. This function is automatically called when an API function returns an error. The function ...ErrorGetMessage is used to assign an error description to the error code.

☞ **Note**:

Only one global error routine can be installed (*...ErrorSetDefaultProc* or *...ErrorSetUserProc*)

● Definitions

Type definition for ME9000MIO_P_ERROR_PROC:

```
typedef void (*ME9000MIO_P_ERROR_PROC) (char
*pcFunctionName, int iErrorCode)
```

VC:me9000mioErrorSetUserProc(ME9000MIO_P_ERRORP ROC pErrorProc)

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ Parameter

`<ErrorProc>`

Pointer to an error routine. The name of the faulty function and the error code will be passed to the function installed there. Passing a NULL will uninstall a previously installed error routine.

❬Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

5.3.2 General Functions

me9000mioGetBoardCount

✎**Description**

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

Number of boards of type ME-9000 installed in your system will be determined.

● **Definitions**

VC: me9000mioGetBoardCount(void);:

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ **Parameter** **none**

Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioGetBoardVersion

✎**Description**

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

This function determines the board version for an installed ME-4600 series board.

● Definitions

VC:me9000mioGetBoardVersion(unsigned int uiBoardNumber,unsigned short* pus Version);

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ Parameters

`<BoardNumber>`

Number of the board to be accessed of type ME-9000.

`<Version>`

Pointer to the device ID.

⟨Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioGetDLLVersion

✎Description

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

Determines the version number of the driver DLL.

● Definitions

VC:me9000mioGetDLLVersion (unsigned long *pulVersion);

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ Parameters

`<Version>`

Version number. The 32-bit value contains the main version (higher 16 bits) and the sub version (lower 16 bits). Example: 0x00020001 is the version 2.01

‹Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioGetDriverVersion

✎Description

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

Determines the version number of the driver.

● Definitions

VC:me9000mioGetDriverVersion (unsigned long *pulVersion);

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ Parameters

`<Version>`

Pointer to a value containing the driver version (hex coded).

‹Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioGetSerialNumber

✎Description

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

Determines the serial number of the selected board.

● Definitions

VC:me9000mioGetSerialNumer (unsigned int uiBoardNumber; unsigned long* pulSerialNumber);

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ Parameters

`<Version>`

Number of the board to be accessed of type ME-9000.

`<Serial Number>`

Pointer to a value containing the serial number.

‹Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

5.3.3 Digital Input/Output

me9000mioDIOClose

✎**Description**

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

This function closes the digital-I/O operation opened with ...*DIO-Open*. Hardware resources which were reserved are released again.

● **Definitions**

VC:me9000mioDIOClose(unsigned int uiBoardNumber);

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ **Parameters**

`<BoardNumber>`

Number of the board to be accessed of type ME-9000.

‹Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioDIOConfig

✎**Description**

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

This function configures the multi-I/O bits (MIO_0...7). They can be configured very flexible (In/Out, interrupt function, inverting the inputs). See chapter 4.2 on page 63.

Call this function separately for each bit. A bit configured as output can also be read back!

To set a bit configured as output in the high-impedance state use the function …*DIOOutput.*

● Definitions

VC:me9000mioDIOConfig (unsigned int uiBoardNumber, unsigned int uiBitNumber, inti Direction, inti Invert, inti Interrupt);

VB: me9000mioVB_…(see `me9000mio.bas`)

➔ Parameters

`<BoardNumber>`

Number of the board to be accessed of type ME-9000.

`<BitNumber>`

Choose multi-I/O bit:

- ME9000MIO_DIO_LINE_0: MIO_0
- ME9000MIO_DIO_LINE_1: MIO_1
- ME9000MIO_DIO_LINE_2: MIO_2
- ME9000MIO_DIO_LINE_3: MIO_3
- ME9000MIO_DIO_LINE_4: MIO_4
- ME9000MIO_DIO_LINE_5: MIO_5
- ME9000MIO_DIO_LINE_6: MIO_6
- ME9000MIO_DIO_LINE_7: MIO_7

`<Direction>`

Direction of the bit:

- ME9000MIO_DIO_OUTPUT: Output bit
- ME9000MIO_DIO_INPUT: Input bit

`<Invert>`

Polarity of an input bit:

- ME9000MIO_DIO_INV_DISABLE: Polarity not inverted
- ME9000MIO_DIO_INV_ENABLE: Polarity inverted

`<Interrupt>`

Enable interrupt functions fort he selected bit. Condition: Bit must be configured as input:

- ME9000MIO_DIO_INT_DISABLE: Interrupt disabled
- ME9000MIO_DIO_INT_ENABLE: Interrupt enabled

Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioDIOGetBit

✎Description

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

This function returns the state of the specified bit. Outputs can be read back with this function.

☞ **Note**:

The ports must be configured with the function …DIOConfig first. The bit has to be configured as input.

● Definitions

VC:me9000mioDIOGetBit (unsigned int uiBoardNumber, unsigned int uiBitNumber, int *piBitValue);

VB: me9000mioVB_…(see `me9000mio.bas`)

➔ Parameters

`<BoardNumber>`

Number of the board to be accessed of type ME-9000.

`<BitNumber>`

Choose multi-I/O bit:

- ME9000MIO_DIO_LINE_0: MIO_0
- ME9000MIO_DIO_LINE_1: MIO_1
- ME9000MIO_DIO_LINE_2: MIO_2
- ME9000MIO_DIO_LINE_3: MIO_3
- ME9000MIO_DIO_LINE_4: MIO_4
- ME9000MIO_DIO_LINE_5: MIO_5
- ME9000MIO_DIO_LINE_6: MIO_6
- ME9000MIO_DIO_LINE_7: MIO_7

<Value>

Pointer to an integer value which returns the state of the bit.

„0“: input line is low

„1“: input line is high

❬Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioDIOGetByte

✎Description

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

This function reads a byte from the 8 digital-I/O lines (MIO_0…7). Outputs can be read back with this function.

☞ **Note**:

The ports must be configured with the function *…DIOConfig* first. All bits bust be configured as inputs.

● Definitions

VC:me9000mioDIOGetByte (unsigned int uiBoardNumber, unsigned char*pucByteValue);

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ Parameters

`<BoardNumber>`

Number of the board to be accessed of type ME-9000.

`<ByteValue>`

Number of the board to be accessed of type ME-9000.

`<BitValue>`

Pointer to an “unsigned char” value, which returns the read byte.

❬Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioDIOGetIrqCount

✎Description

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

This function determines the number of interrupts from all multi-I/O bits which are configured as interrupt input. Condition: interrupt operation enabled by function me9000mioDIOConfig. By call up the value of parameter Count it is possible to determine relative to a previous call up whether an interrupt occurred or not.

● **Definitions**

VC:me9000mioDIOGetIrqCount (unsigned int uiBoardNumber, unsigned long*pulCount);

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ **Parameters**

`<BoardNumber>`

Number of the board to be accessed of type ME-9000.

`<Count>`

Total number of interrupts from all multi-I/O bits, which are configured as interrupt input (IRQ-DIO).

❬**Return value**

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioDIOOpen

✎**Description**

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

This function opens programing of the digital-I/O section.

● **Definitions**

VC: me9000mioDIOOpen (unsigned int uiBoardNumber,

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ **Parameters**

`<BoardNumber>`

Number of the board to be accessed of type ME-9000.

‹Return value

If the function is successfully executed, an '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioDIOOutput

✎Description

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

Use this function to set a multi-I/O bit in the high-impedance state which was configured as output by the function *...DIOConfig*.

● Definitions

VC: me9000mioDIOOutput (unsigned int uiBoardNumber, unsigned int uiBitNumber, int iOutputMode);

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ Parameters

`<BoardNumber>`

Number of the board to be accessed of type ME-9000.

`<BitNumber>`

Choose multi-I/O bit:

- ME9000MIO_DIO_LINE_0: MIO_0
- ME9000MIO_DIO_LINE_1: MIO_1
- ME9000MIO_DIO_LINE_2: MIO_2
- ME9000MIO_DIO_LINE_3: MIO_3
- ME9000MIO_DIO_LINE_4: MIO_4
- ME9000MIO_DIO_LINE_5: MIO_5
- ME9000MIO_DIO_LINE_6: MIO_6
- ME9000MIO_DIO_LINE_7: MIO_7

<OutputMode>

Set the output bit in a driving or high-impedance state:

- ME9000MIO_DIO_OUTPUT_ENABLE: Set output bit driving
- ME9000MIO_DIO_OUTPUT_DISABLE: Set output bit in a high-impedance state

‹Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioDIOReset

✎Description

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

All the multi-I/O bits (MIO_0...7) are reset, i.e.:

- All bits are set to input
- Inputs not inverting
- Interrupt function disabled
- Output driving

● Definitions

VC: int me9000mioDIOReset (unsigned int uiBoardNumber);

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ Parameters

<BoardNumber>

Number of the board to be accessed of type ME-9000.

‹Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioDIOSetBit

✎Description

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

This function sets the state of the specified output bit.

☞ **Note**:

The ports must be configured with the function *...DIOConfig* first.

● Definitions

VC: me9000mioDIOSetBit (unsigned int uiBoardNumber, unsigned int uiBitNumber, int iBitValue);

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ Parameters

`<BoardNumber>`

Number of the board to be accessed of type ME-9000.

`<BitNumber>`

Choose multi-I/O bit:

- ME9000MIO_DIO_LINE_0: MIO_0
- ME9000MIO_DIO_LINE_1: MIO_1
- ME9000MIO_DIO_LINE_2: MIO_2
- ME9000MIO_DIO_LINE_3: MIO_3
- ME9000MIO_DIO_LINE_4: MIO_4

- ME9000MIO_DIO_LINE_5: MIO_5
- ME9000MIO_DIO_LINE_6: MIO_6
- ME9000MIO_DIO_LINE_7: MIO_7

<BitValue>

Pointer values are:

„0“: Bit is set low level

„1“: Bit is set to high level

‹Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioDIOSetByte

✎Description

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

Writes a byte to all of the 8 digital-I/O lines (MIO_0...7).

☞ **Note**:

The ports must be configured with the function *...DIOConfig* first. All bits must be configured as output bit.

● Definitions

VC:me9000mioDIOSetByte (unsigned int uiBoardNumber, unsigned char ucByteValue);

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ Parameters

`<BoardNumber>`

Number of the board to be accessed of type ME-9000.

`<ByteValue>`

Value to be output, possible values are: 0...255 (00Hex...FFHex)

❬Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioDIOSetCallback

✎Description

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

Function to install a callback routine.

● Definitions

VC: me9000mioDIOSetCallback (unsigned int uiBoardNumber, ME9000MIO_P_DIO_PROC pDIOIrqProc, void*pDIOIrqContext);

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ Parameters

`<BoardNumber>`

Number of the board to be accessed of type ME-9000.

<DIOIrqProc>

Callback routine to be called repeatedly.

<DIOIrqContext>

User defined pointer to be passed to the callback routine.

❬Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

5.3.4 Counter Functions

me9000mioDIOCntClose

✎Description

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

This function closes the operation opened with the function *...CntOpen*. Hardware resources which were reserved are released again.

● Definitions

VC: int me9000mioCntClose (unsigned int uiBoardNumber);

VB: me9000mioVB_...(see me9000mio.bas)

➔ Parameters

<BoardNumber>

Number of the board to be accessed of type ME-9000.

‹Return value

If the function is successfully executed, an '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioDIOCntClose

✎Description

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

This function is for configuration of the 16-bit counter. After loading the start value by the function *...CntWrite* and calling the function*...CntStart* it counts down to „0“. On zero axis crossing either an interrupt can be triggered and/or the multi-I/O bit MIO_0 can output a counter event. You can choose between „Single Shot“ and „Retrigger“ mode. In „Single Shot“ mode the counter stops at „0“, in „Retrigger“ mode the counter becomes reloaded with the start value. In that case the counter operation will be ended with the function *...Cnt- Stop*. See chapter 4.3 on page 68.

● Definitions

VC me9000mioCntConfig (unsigned int uiBoardNumber, int iRetrigger, int iCntEvent, int iClockSource, int iInterrupt, ME9000MIO_P_CNT_PROC pCntIrqProc, void *pCntIrqContext);

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ Parameters

`<BoardNumber>`

Number of the board to be accessed of type ME-9000.

`<Retrigger>`

Choose counter mode:

- ME9000MIO_CNT_RETRIGGER_DISABLE: „Single Shot“ (counting once to „0“)
- ME9000MIO_CNT_RETRIGGER_ENABLE: „Retrigger“ (reloading the start value on zero axis crossing).

<CntEvent>

Multi-I/O pin MIO_0 to output a counter event (on zero axis crossing):

- ME9000MIO_CNT_EVENT_DISABLE: Disable counter event
- ME9000MIO_CNT_EVENT_ENABLE: Enable counter event

Please note: Pin MIO_0 may not be switched as an input because the pin becomes an output automatically. If the board will be damaged no guarantee can be given!

<ClockSource>

Choose between internal clock (14,7456MHz) or external clock by pin TMRCK.

- ME9000MIO_CNT_CLOCK_INTERNAL: internal clock
- ME9000MIO_CNT_CLOCK_EXTERNAL: external clock

<Interrupt>

On zero axis crossing an interrupt (IRQ-CNT) can be triggered. The interrupt is directly forwarded to the system.

- ME9000MIO_CNT_INT_DISABLE: Disable interrupt
- ME9000MIO_CNT_INT_ENABLE: Enable interrupt

<CntIrqProc>

Callback function, which can be called on zero axis crossing. If no callback function is required, pass the constant. ME9000MIO_POINTER_NOT_USED.

<CntIrqContext>

This is a user defined pointer that is passed to the callback function. If no callback function is required, pass the constant ME9000MIO_POINTER_NOT_USED.

‹Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioCntGetIrqCount

✎Description

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

This function determines the number of interrupts from the counter. Condition: interrupt operation enabled by function *me9000mioCntConfig* . By call up the value of parameter Count it is possible to determine relative to a previous call up whether an interrupt occurred or not.

● Definitions

VC: me9000mioCntGetIrqCount (unsigned int uiBoardNumber, unsigned long *pulCount);

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ Parameters

<BoardNumber>

Number of the board to be accessed of type ME-9000.

<Count>

Total number of interrupts from the counter (IRQ-CNT).

⟨Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioCntOpen

✎**Description**

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

This function opens programming the counter functions.

● **Definitions**

VC int me9000mioCntOpen (unsigned int uiBoardNumber);;

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ **Parameters**

`<BoardNumber>`

Number of the board to be accessed of type ME-9000.

❬**Return value**

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioCntRead

✎**Description**

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

With this function the start value of the counter can be read back.

● **Definitions**

VC me9000mioCntRead (unsigned int uiBoardNumber, unsigned short *pusValue);

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ Parameters

`<BoardNumber>`

Number of the board to be accessed of type ME-9000.

`<Value>`

Start value of the counter (16 bit).

❬Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioCntReset

✎Description

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

The counter will be stopped and reset:

- Mode „Single Shot“
- Counter event disabled
- Internal clock
- Interrupt disabled

● Definitions

VC me9000mioCntReset (unsigned int uiBoardNumber);

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ Parameters

`<BoardNumber>`

Number of the board to be accessed of type ME-9000.

Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioCntStart

✎Description

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

Starts counting down in the chosen mode (see ...*CntConfig*).

● Definitions

VC me9000mioCntStart (unsigned int uiBoardNumber);

VB: me9000mioVB_...(see `me9000mio.bas`)

→ Parameters

`<BoardNumber>`

Number of the board to be accessed of type ME-9000.

Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioCntStop

✎Description

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

Stop the counter in mode “Retrigger”.

● Definitions

VC me9000mioCntStop (unsigned int uiBoardNumber);

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ Parameters

`<BoardNumber>`

Number of the board to be accessed of type ME-9000.

‹Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

me9000mioCntStop

✎Description

ME-9000	ME-9100	ME-9300	ME-90 PC/104-Plus
✓	–	–	✓

With this function you can load a 16-bit start value into the counter. Start the counter operation by calling the function *me9000mioCntStart.*

● Definitions

VC: me9000mioCntWrite (unsigned int uiBoardNumber, unsigned short usValue);

VB: me9000mioVB_...(see `me9000mio.bas`)

➔ Parameters

`<BoardNumber>`

Number of the board to be accessed of type ME-9000.

`<Value>`

16-bit start value for counter; value range: 0...65535 (0000Hex...FFFFHex)

❬Return value

If the function is successfully executed, a '0' ME9000MIO_NO_ERROR) is returned. If an error occurs, an error code unequal to '0' is returned. The cause of the error can be determined with the functions for error handling.

Appendix

A Specification

PC Interface

Resources are assigned automatically (Plug&Play)	
ME-9000 PCI/cPCI (Rev. 2.2)	PCI local bus specification version 2.2 (32 bit, 33 MHz, Universal-PCI: 5 V/3,3 V)
ME-9000 PCI-Express	PCI-Express x1 specification version 2.0 (32 bit, 33 MHz, 3,3 V)
ME-9100/9300 PCI/cPCI	PCI local bus specification version 2.1 (32 bit, 33 MHz, 5 V)
ME-90 PC/104-Plus	PC/104-Plus specification version 2.3 (32 bit, 33 MHz, Universal-PCI: 5 V/3,3 V)

Note: One „**Mix**“-versions RS-232 and RS-422/485 ports can be combined on one board.

UARTs (ME-9000/9100/9300, ME-90 PC/104-Plus)

Number of ports (RS-232 + RS-422/485)	ME-9000: 2, 4 or 8 (RS-232, RS-422/485) ME-9100: 4 or 8 (RS-232, RS-422/485) ME-9300: 16 (RS-232) ME-90 PC/104-Plus: 8 (RS-232, RS-422/485)
Type ME-9000 and ME-90 PC/104-Plus	Octo-UART integrated in the PCI controller of type EXAR XR17D158IV; register compatible to the 16550 with integrated transmit and receive FIFO for each port.
Type ME-9100	1 resp. 2 Quad-UARTs of type OX16C954 or compatibles (depends on number of ports); register compatible to the 16550 with integrated transmit and receive FIFO for each port.
Type ME-9300	4 Quad-UARTs of type OX16C954 or compatibles; register compatible to the 16550 with integrated transmit and receive FIFO for each port.
FIFO capacity	ME-9000/ME-90 PC/104-Plus: each 64 bytes ME-9100/9300: each 128 bytes

Transfer rates	75/110/134/150/300/600/1200/1800/2400/4800/7200/9600/14.400/19.200/38.400/57.600/115.200/128.000/230.400/460.800/921.600 Bd
Parity	none, odd, even, mark, space
Data bits	4; 5; 6; 7; 8
Stop bits	1; 1,5; 2
Flow control	Xon/Xoff, hardware, none

RS-232 Ports (ME-9000/9100/9300, ME-90 PC/104-Plus)

Signals	RxD, TxD, DCD, DTR, DSR, RTS, CTS, RI
Transfer distance	max. 15 m
Voltage level for output signals (TxD)	logical: „0“: typ. +5,4 V (+5 V < U < +13,2 V) logical: „1“: typ. -5,4 V (-13,2 V < U < -5 V)
Voltage level for input signals (RxD)	logical: „0“: +3 V < U < +25 V logical: „1“: -25 V < U < -3 V
ESD protection	up 15 kV (IEC 1000)

Electrical isolation	for „i“-versions up to 500 V
	for „p“-versions up to 500 V

RS-422/485 Ports (ME-9000/9100, ME-90 PC/104-Plus)

Operation modes	- RS-422 - RS-485 half-duplex (automatic flow-control) - RS-485 full-duplex
Transfer distance	max. 1200 m
Differential output voltage of the buffer:	2…4.8 V
Differential input voltage for high level:	min. 200 mV
Differential input voltage for low level:	max. 200 mV
Output offset voltage relating to GND	2…3 V
Electrical isolation	for „i“- und „p“-versions:

	- to PC ground: 500 V - between the ports (only „p“-models): 500 V

Multi-I/O Pins (ME-9000, ME-90 PC/104-Plus)

Number	8-bit bidirectional
Type	TTL ports
Interrupt	IRQ-DIO (must be enabled)
Output level (Vcc = 5 V±10 %)	U_{OL}: max. 0,55 V at 6 mA U_{OH}: min. 2,4 V at -2 mA
Input level (Vcc = 5 V±10 %)	U_{IL}: -0,5 < 0,8 V U_{IH}: 2,0 V < 6,0 V
Reference to GND	PC ground (GND_PC)

Counter (ME-9000, ME-90 PC/104-Plus)

Number	1 x 16 bit
Type	down-counter
Modes	„Single-Shot“ or „Retrigger“
Clock source	internal/external
Internal Oscillator	crystal oscillator (14,7456 MHz/100 ppm)
Ext. clock input	TMRCK
Interrupt	IRQ-CNT (must be enabled)
Reference to GND	PC-ground (GND_PC)

General Information

Power consumption	typ. 2,3 A at +5 V
Max. load of VCC on the customer design area (DA) resp. ST2 of the ME-9000:	max. 300 mA @ VCC (+5 V resp. +3,3 V)
Max. load of VCC via ST3 of the ME-90 PC/104-Plus:	max. 300 mA @ VCC (+5 V)
Physical size (without mounting bracket and connectors)	ME-9000 PCIe: 124 x 99 mm ME-9000 PCI: 124 x 99 mm ME-9100 PCI: 136 x 99 mm ME-9300 PCI: 129 x 99 mm ME-90 PC/104-Plus: 90 x 96 mm CompactPCI models: 3U CompactPCI

Connection ME-9000	78-pin D-Sub female connector with dual, quad or octopus cable to 9-pin D-Sub male connectors 20-pin IDC connector (ST2) for MIO pins incl. flat ribbon cable to 25-pin.
Connection ME-9100	78-pin D-Sub female connector with dual, quad or octopus cable to 9pin D-Sub male connectors (ST2) for MIO pins incl. flat ribbon cable to 25-pin D-Sub female connector.
Connection ME-9300	2 x 68-pin VHDCI female connectors
Connection ME-90 PC/104-Plus	Bus connectors for PCI and ISA bus (ISA bus looped through only), 2 x 40-pin IDC connectors (with 90° angle) incl. 2 flat ribbon cables with each 4 x 9-pin. D-Sub male connector, 20-pin. IDC connector (ST3) for MIO pins incl. flat ribbon cable to 25-pin D-Sub female connector.
Operating temperature	ME-9000: -40…+71 °C ME-9100/9300: 0…70 °C ME-90 PC/104-Plus: -40…+85 °C
Storage temperature	ME-9000/9100: -40…100 °C ME-9300: -40…100 °C ME-90 PC/104-Plus: -40…100 °C
Relative humidity	20…55 % (non-condensing)

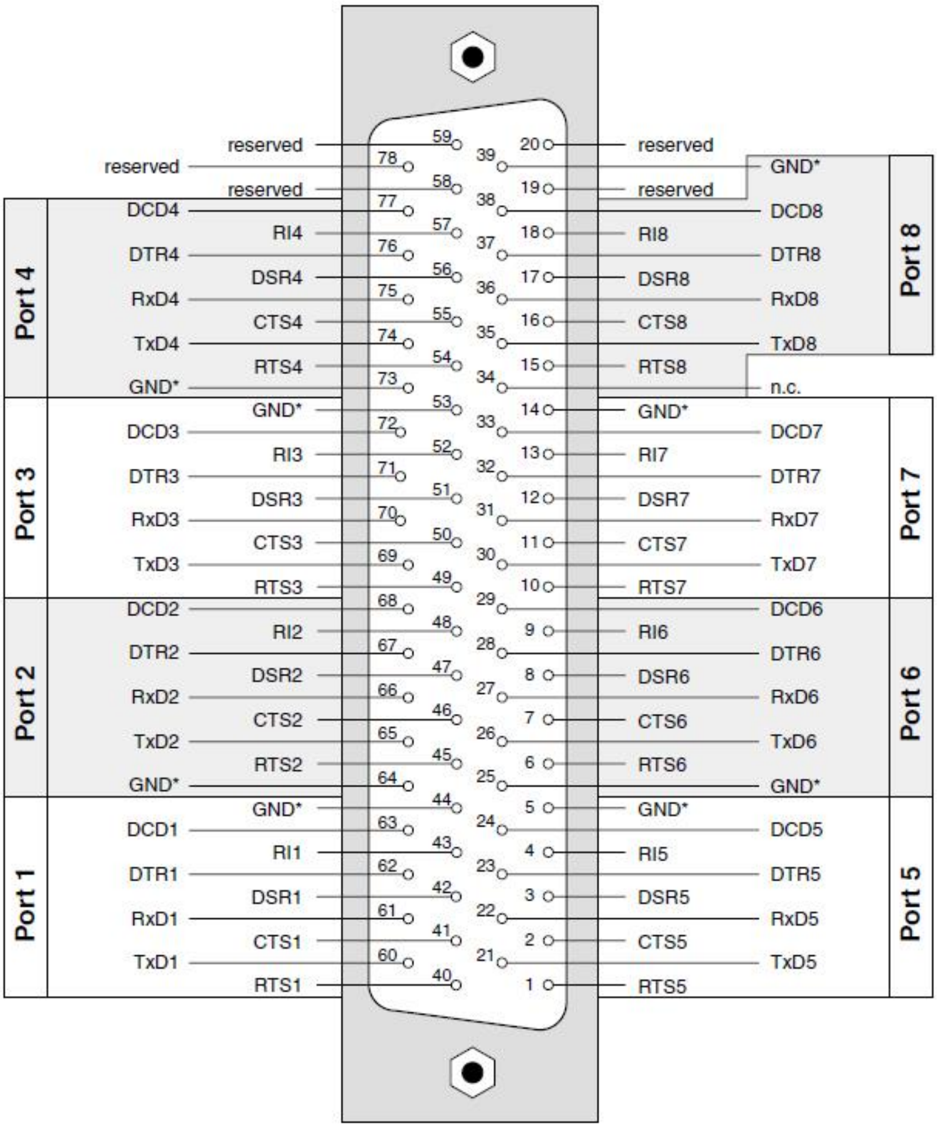
CE Certification

EMC-directive	89/336/EMC
Emission	EN 55022
Noise immunity	EN 50082-2

B Pinout

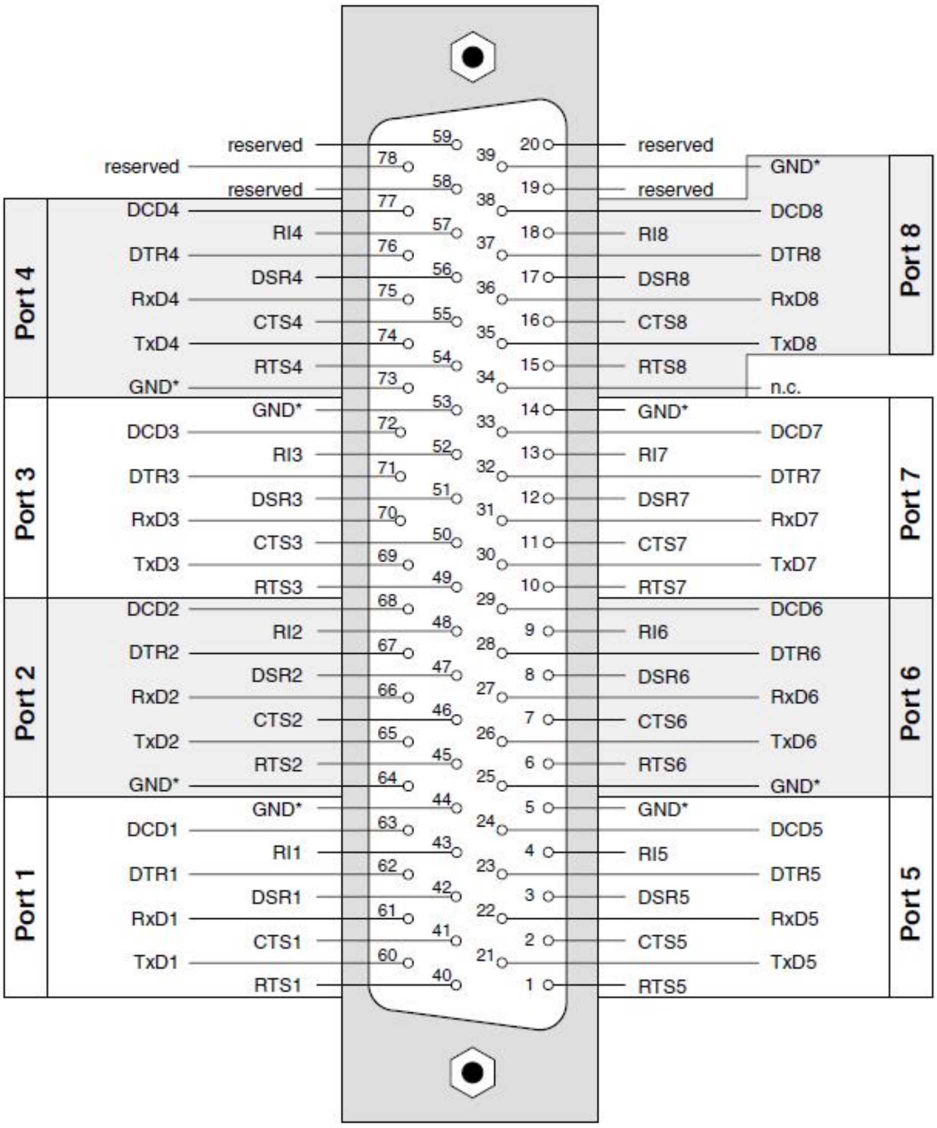
Note the different pinout of the RS-232 and RS-422/485 ports. The „**MIX**- versions provide RS-232 as well as RS-422/485 ports. The RS-232 ports occupy always the lower significant ports followed by the RS-485 ports.

B1 ME-9000/9100 RS-232



Picture 58: Pinout of the 78-pin D-Sub female connector for RS-232 ports

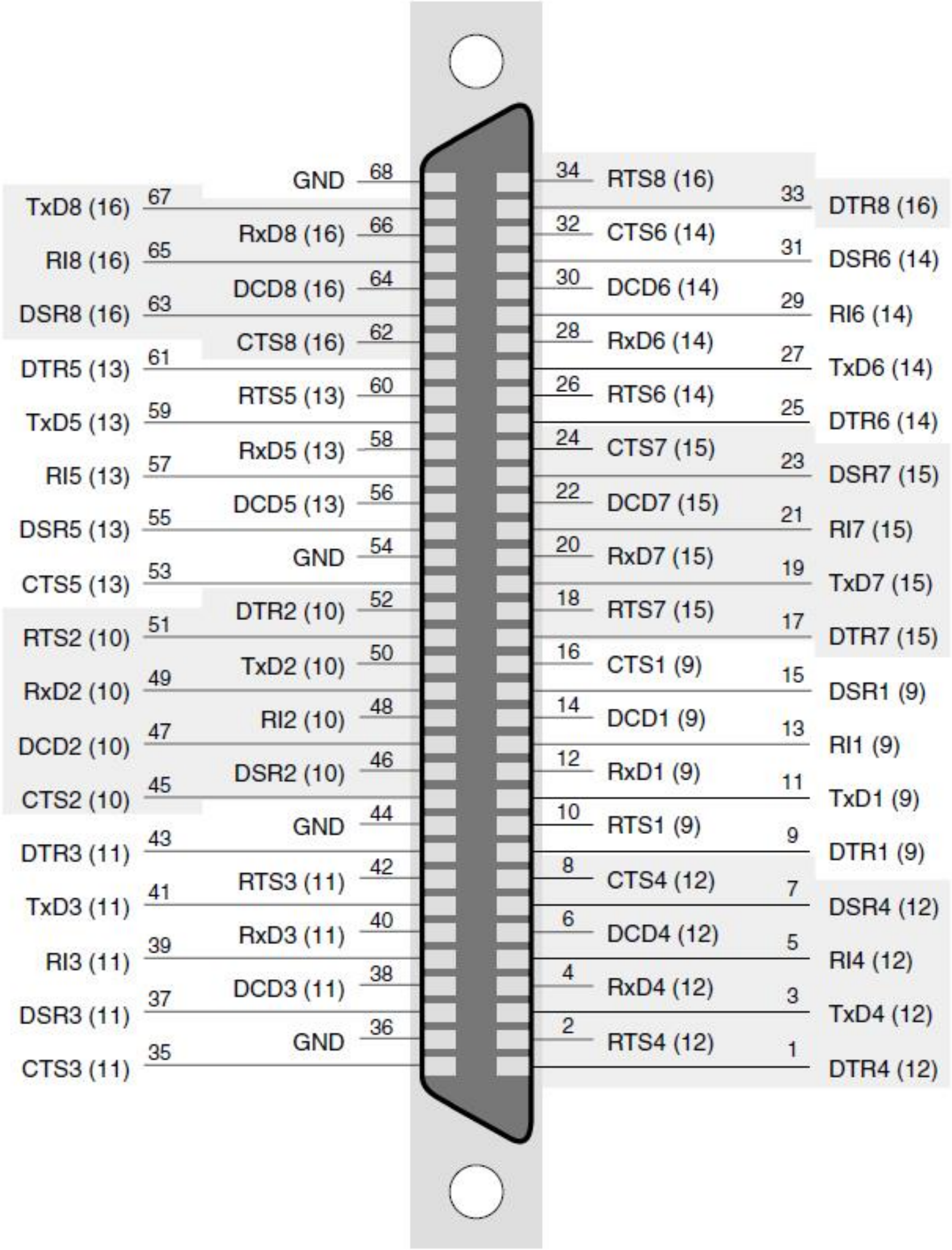
B2 ME-9000/9100 RS-422/485



Picture 59: Pinout of the 78-pin D-Sub female connector for RS-422-485 ports

*see note on page 111!

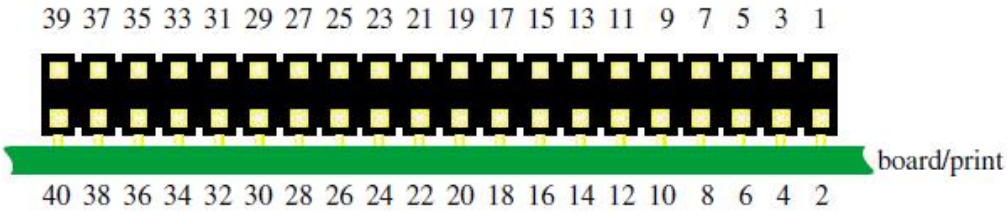
B3 ME-9300 RS-232



Picture 60: 68-pin VHDC female connector of the ME-9300

Pinout is valid for both VHDC connectors (connector A: lower, connector B: upper). The numbers in brackets describe the port numbers of connector B.

B4 ME-90PC/104-Plus



Picture 61: 40-pin IDC connector of the ME-90 PC/104-Plus

Both of the 40-pin IDC connectors for COM1…4 and COM5…8 are allocated identically (see the following table).

Pin	COM	RS-232	Direction	RS-422/485	Direction
1	1 or 5	DCD	Input	RxD+	Input
2	1 or 5	DSR	Input	reserved	–
3	1 or 5	RxD	Input	RxD-	Input
4	1 or 5	RTS	Output	reserved	–
5	1 or 5	TxD	Output	TxD+	Output
6	1 or 5	CTS	Input	reserved	–
7	1 or 5	DTR	Output	TxD-	Output
8	1 or 5	RI	Input	reserved	–
9	1 or 5	GND	Mass	GND	Mass
10	1 or 5	+5 V	VCC	n. c.	–
11	2 or 6	DCD	Input	RxD+	Input
12	2 or 6	DSR	Input	reserved	–
13	2 or 6	RxD	Input	RxD-	Input
14	2 or 6	RTS	Output	reserved	–
15	2 or 6	TxD	Output	TxD+	Output
16	2 or 6	CTS	Input	reserved	–
17	2 or 6	DTR	Output	TxD-	Output
18	2 or 6	RI	Input	reserved	–
19	2 or 6	GND	Mass	GND	Mass

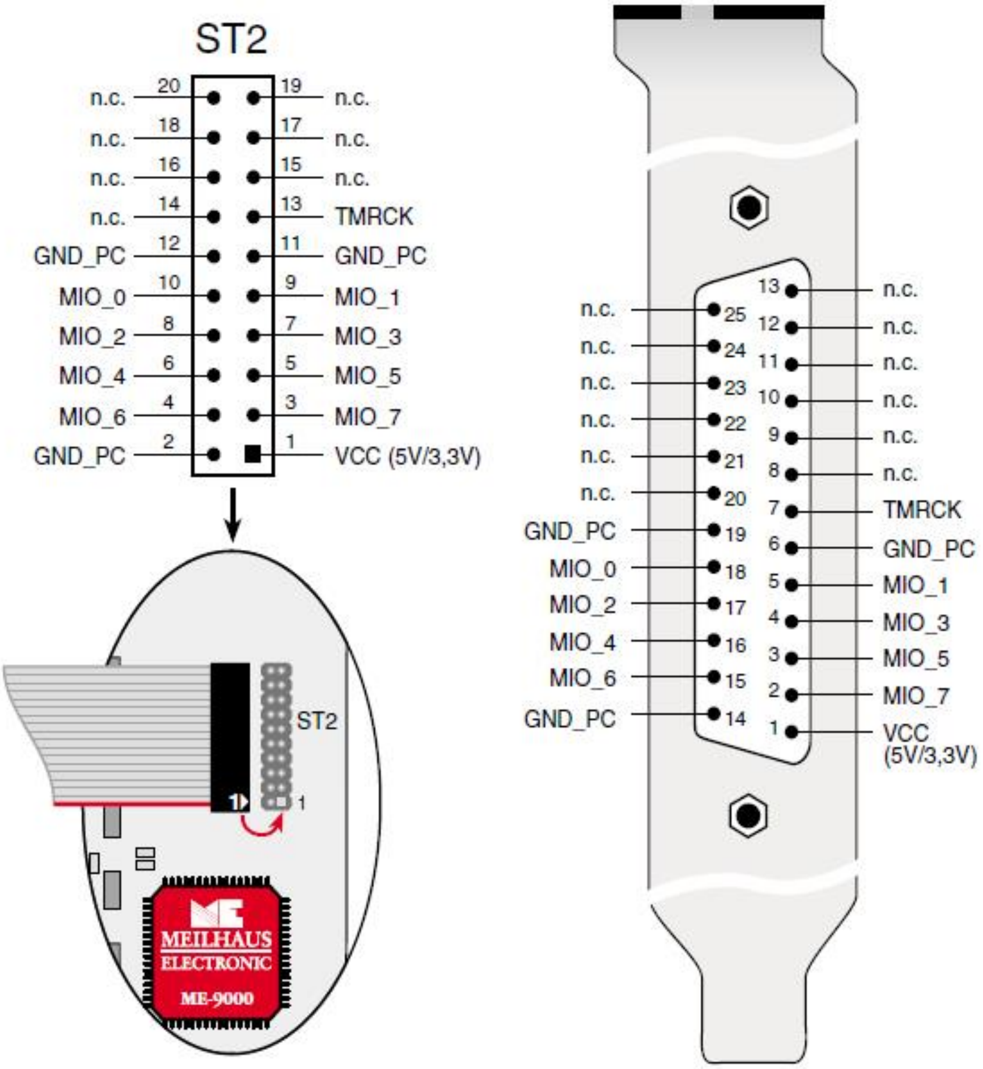
Table 9: Pinout of the 40-pin IDC connector

Pin	COM	RS-232	Direction	RS-422/485	Direction
20	2 or 6	+5 V	VCC	n. c.	–
21	3 or 7	DCD	Input	RxD+	Input
22	3 or 7	DSR	Input	reserved	–
23	3 or 7	RxD	Input	RxD-	Input
24	3 or 7	RTS	Output	reserved	–
25	3 or 7	TxD	Output	TxD+	Output
26	3 or 7	CTS	Input	reserved	–
27	3 or 7	DTR	Output	TxD-	Output
28	3 or 7	RI	Input	reserved	–
29	3 or 7	GND	Mass	GND	Mass
30	3 or 7	+5 V	VCC	n. c.	–
31	4 or 8	DCD	Input	RxD+	Input
32	4 or 8	DSR	Input	reserved	–
33	4 or 8	RxD	Input	RxD-	Input
34	4 or 8	RTS	Output	reserved	–
35	4 or 8	TxD	Output	TxD+	Output
36	4 or 8	CTS	Input	reserved	–
37	4 or 8	DTR	Output	TxD-	Output
38	4 or 8	RI	Input	reserved	–
39	4 or 8	GND	Mass	GND	Mass
40	4 or 8	+5 V	VCC	n. c.	–

Table 9: Pinout of the 40-pin IDC connector

B5 Auxiliary Connector ST2 (ME-9000)

ME-AK-D25F/S: Adapter cable from 20-pin IDC connector to mounting bracket with 25-pin D-Sub female connector (comes with the board).

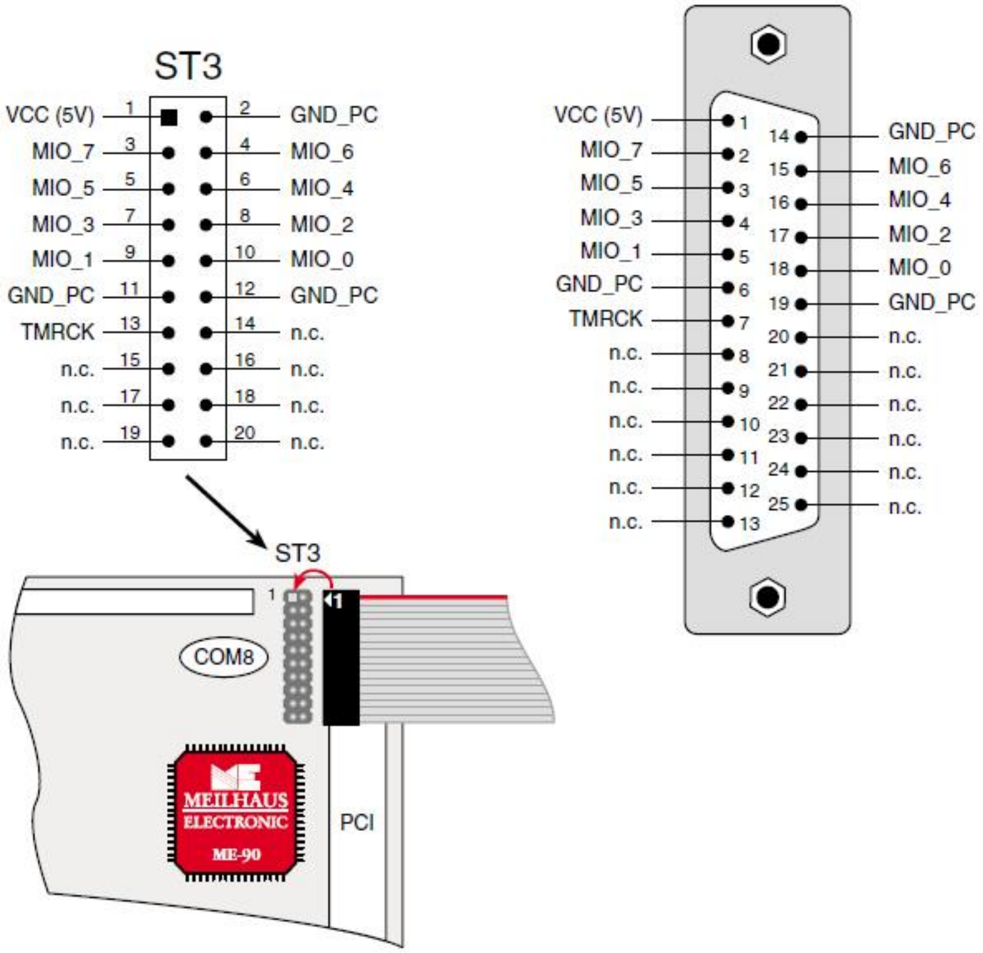


Picture 62: Pinout of ST2 on ME-9000 (i)

Note: Connect the mounting bracket pin 1 of the flat ribbon cable (red marked line) as shown above to the IDC connector ST2.

B6 Auxiliary Connector ST3 (ME-90 PC/104-Plus)

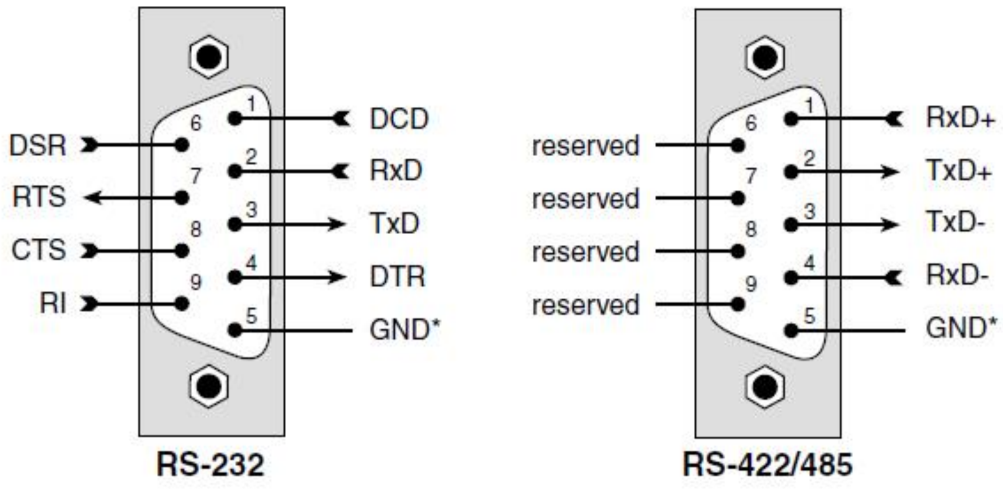
ME-AK-D25F: Adapter cable for multi-I/O port of the ME-90 PC/104-Plus from 20-pin IDC connector to 25-pin D-Sub female connector (comes with the board).



Picture 63: Pinout of ST3 of the ME-90 PC/104-Plus

Note: Connect pin 1 of the flat ribbon cable (red marked line) as shown above to pin 1 of the IDC connector ST3.

B7 9-pin male connector ME-9000/9100/9300



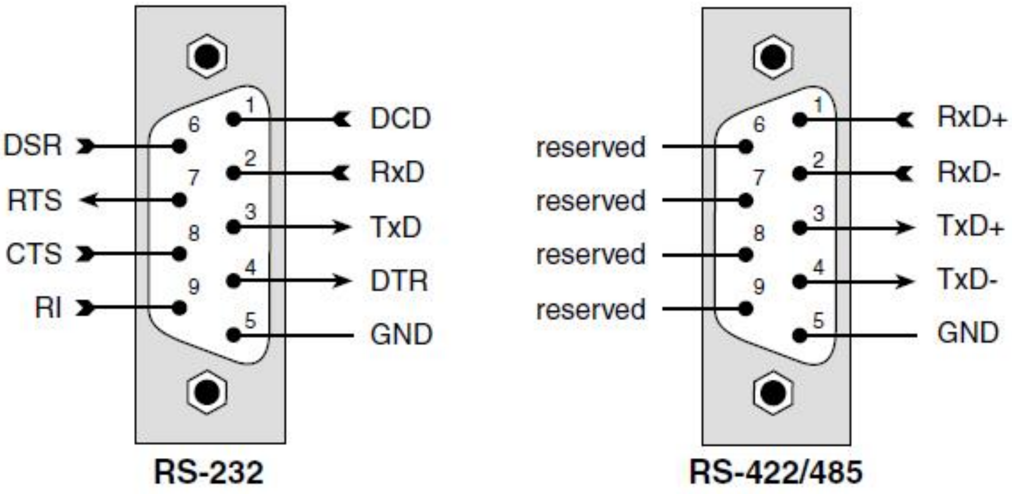
Picture 64: 9-pin D-Sub male connector ME-9x00

***Ground Reference ME-9000 Series**

Note the different ground reference at the GND pins of the connectors of the ME-9000 series:

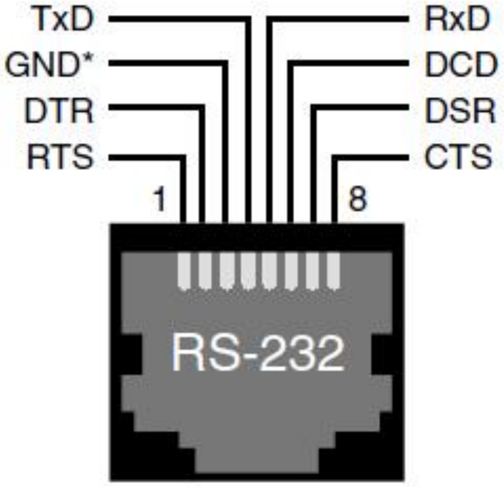
- TTL models (without opto-isolation): PC ground (GND_PC).
- „i“-models: from application view one common ground (GND_C) isolated to PC ground.
- „p“-models: ground of the single ports isolated from one another and to PC ground, so called „island-grounds“ (GND_x).

B8 9-pin male connector ME-90 PC/104-Plus

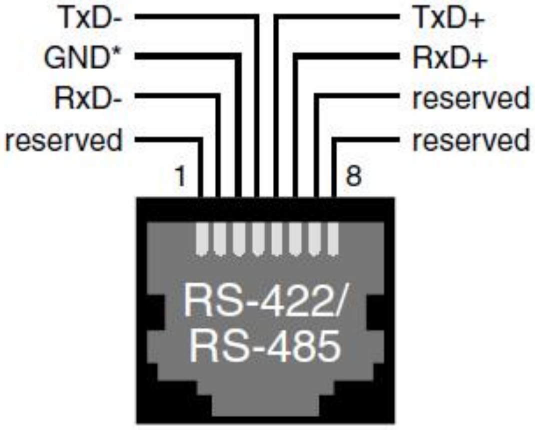


Picture 65: 9-pin D-Sub male connector of ME-90 PC/104-Plus

B9 8-pin RJ-45 female connectors

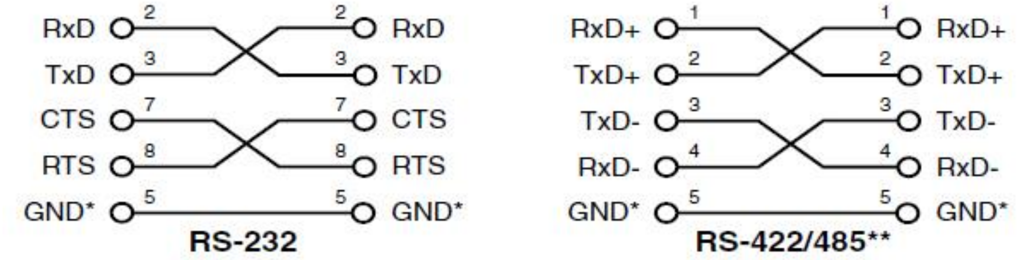


Picture 66: 8-pin RJ-45 female connector for RS_232 ports (Rocket-Port pinout)



Picture 67: 8-pin RJ-45 female connector for RS-422/485 ports (not Rocket-Port compatible)

B10 Null modem cable



Picture 68: Null modem cable RS-232 (left), RS-422/485 (right)

* see note on page 111! **not for ME-90 PC/104-Plus

C Accessories

We recommend to use high-quality connector cables with single-shielded lines per channel.

For further accessories please refer to the current Meilhaus Electronic catalog and the internet:
www.meilhaus.de/en/pc-boards/accessories/

Connectivity options for ME-9000/9100:

- **ME AK-DDual** Connection cable from 78-pin D-Sub male connector to 2 x 9-pin D-Sub male connectors (1 m).
- **ME AK-DQuad** Connection cable from 78-pin D-Sub male connector to 4 x 9-pin D-Sub male connectors (1 m).
- **ME AK-DOcto** Connection cable from 78-pin D-Sub male connector to 8 x 9-pin D-Sub male connectors (1 m).
- **ME AB-D9/8-78 + ME AK-D78/1:**
  Octo terminal panel in a closed box from 78-pin D-Sub male connector to 8 x 9-pin D-Sub male connectors. Suitable connection cable ME AK-D78/1 (1 m) from 78-pin D-Sub male connector to 78-pin D-Sub female connector.
- **ME AB-D9/8-78-H + ME AK-D78/1:**
  Octo terminal panel for DIN rail mounting from 78-pin D-Sub male connector to 8 x 9-pin D-Sub male connectors. Suitable connection cable ME AK-D78/1 (1 m) from 78-pin D-Sub male connector to 78-pin D-Sub female connector.
- **ME AB-RJ45/8x8-V + ME AK-D78/VHDCI** (not for ME-9000p):
  Octo terminal panel in a closed box from 78-pin D-Sub male connector to 8 x 8-pin RJ-45 female connectors. Suitable connection cable ME AK-D78/VHDCI (1 m) from 78-pin D-Sub male connector to 68-pin VHDCI female connector.
- **ME AB-RJ45/8x8-PV + ME AK-D78/VHDCI** (not for ME-9000p):
  Octo terminal panel without case for integration in user specific systems from 78-pin D-Sub male connector to 8 x 8-pin RJ-45 female connectors. Suitable connection cable ME AK-D78/VHDCI (1 m) from 78-pin D-Sub male connector to 68-pin VHDCI female connector.

Connectivity options for ME-9300:

- **2 x ME AB-D9/8-V + 2 x VHDCI cable:**
  Octo terminal panel in a closed box from 68-pin VHDCI connector to 8 x 9-pin D-Sub male connectors. Connection via 1:1 VHDCI cable (1 m). Terminal panel and cable are required twice.
- **2 x ME AB-D9/8-HV + 2 x VHDCI cable:**
  Octo terminal panel for DIN rail mounting from 68-pin VHDCI connector to 8 x 9-pin D-Sub male connectors. Connection via 1:1 VHDCI cable (1 m). Terminal panel and cable are required twice.
- **ME AB-D9/16-V + 2 x VHDCI cable:**
  16 port terminal panel for 19“-rack-mount from VHDCI connector (2 x 68-pin) to 16 x 9-pin D-Sub male connectors. Connection via two 1:1 VHDCI cable (1 m).
- **2 x ME AB-RJ45/8x8-V + 2 x VHDC cable:**
  Octo terminal panel in a closed box from 68-pin VHDCI connector to 8 x 8-pin RJ-45 female connectors. Connection via 1:1 VHDCI cable (1 m). Terminal panel and cable are required twice.
- **2 x ME AB-RJ45/8x8-PV + 2 x VHDCI cable:**
  Octo terminal panel without case for integration in user specific systems from 68-pin VHDCI connector to 8 x 8-pin RJ-45 female connectors. Connection via 1:1 VHDCI cable (1 m). Terminal panel and cable are required twice.

Connectivity options for ME-90 PC/104-Plus

- **ME AK 4D9M** flat ribbon cable from 40-pin IDC connector to 4 x 9-pin D-Sub male connector.

D Technical Questions

D1 Hotline

Should you have questions or inquiries concerning your Meilhaus device, please contact us:

Meilhaus Electronic GmbH
Repair & Service
Am Sonnenlicht 2
D-82239 Alling

Sales:		**Support:**	
Tel.:	(08141) 52 71 – 0	Tel.:	(08141) 52 71 – 188
Fax:	(08141) 52 71 – 129	Fax:	(08141) 52 71 – 169
eMail:	sales@meilhaus.de	eMail:	support@meilhaus.de

Download-Server and Driver Update:
To download current driver versions for Meilhaus Electronic devices as well as manuals in PDF format, please go to:
www.meilhaus.org/driver

Service Department with RMA Process:
In case you need to return a board for repair purposes, we strongly ask you attach a detailed description of the error as well as information regarding your computer/system and the software used. Please register online using our RMA process:
www.meilhaus.de/en/infos/service/rma.htm.

E Constant Definitions

Note: The following constant definitions are valid for Windows. Please note also the current definition file (me9000miodefs.h) included with the Meilhaus Electronic Developer Kit (ME-SDK). The Linux driver uses its own constant definitions (see Linux driver).

Constant	Value
General	
ME9000MIO_BOARD_0	0x0
ME9000MIO_BOARD_1	0x1
ME9000MIO_BOARD_2	0x2
ME9000MIO_BOARD_3	0x3
ME9000MIO_VALUE_NOT_USED	0
ME9000MIO_POINTER_NOT_USED	NULL
ME9000MIO_NO_ERROR	0x00000000
Error Handling	
ME9000MIO_ERROR_DEFAULT_PROC_DISABLE	0x0
ME9000MIO_ERROR_DEFAULT_PROC_ENABLE	0x1
Digital Input/Output	
ME9000MIO_DIO_LINE_0	0x0
ME9000MIO_DIO_LINE_1	0x1
ME9000MIO_DIO_LINE_2	0x2
ME9000MIO_DIO_LINE_3	0x3
ME9000MIO_DIO_LINE_4	0x4
ME9000MIO_DIO_LINE_5	0x5
ME9000MIO_DIO_LINE_6	0x6
ME9000MIO_DIO_LINE_7	0x7

ME9000MIO_DIO_OUTPUT	0x0
ME9000MIO_DIO_INPUT	0x1
ME9000MIO_DIO_LINE_ENABLE	0x0
ME9000MIO_DIO_LINE_DISABLE	0x1
ME9000MIO_DIO_INT_DISABLE	0x0
ME9000MIO_DIO_INT_ENABLE	0x1
ME9000MIO_DIO_INV_DISABLE	0x0
ME9000MIO_DIO_INV_ENABLE	0x1

Table 10: Constant definitions

Constant	Value
Counter	
ME9000MIO_CNT_INT_DISABLE	0x0
ME9000MIO_CNT_INT_ENABLE	0x1
ME9000MIO_CNT_RETRIGGER_DISABLE	0x0
ME9000MIO_CNT_RETRIGGER_ENABLE	0x1
ME9000MIO_CNT_EXT_CLOCK_DISABLE	0x0
ME9000MIO_CNT_EXT_CLOCK_ENABLE	0x1
ME9000MIO_CNT_EXT_EVENT_DISABLE	0x0
ME9000MIO_CNT_EXT_EVENT_ENABLE	0x1

Table 11: Constant definitions

F Index