

User's Guide

GPIB PCIGPIB1

Copyright © QUANCOM Informationssysteme GmbH

All specifications in this manual were compiled after a thorough check, and it is not considered as a warranty for product properties. QUANCOM is not responsible for any errors or omissions contained in this user's manual, and reserves the right to make changes without notice. Passing on and duplication of this manual and the utilisation of its contents as well as the software belonging to the product are permitted only with written permission by QUANCOM.

Table of contents

1	Overview	6
1.1	Introduction	6
1.2	Our experience is your profit.....	6
1.3	Customer Communication	6
1.4	Changes in this manual and software updates	7
1.5	Scope of supply	7
2	Installation procedures.....	8
2.1	System requirements	8
2.2	Safety precautions.....	8
2.3	Installing the board.....	9
3	Technical Hardware Description	10
3.1	General.....	10
3.2	Operation of the GPIB card.....	10
3.3	Installation	11
3.4	Technical data	11
3.4.1	GPIB-1 : Short 8-Bit card for PC's with ISA-BUS:	11
3.4.2	PCIGPIB1 : PCI IEEE-488 Interface with 8 diagnostic LED's.....	11
3.5	Card overview GPIB-1	12
3.5.1	JP1 Base I/O Address setting	13
3.5.2	JP2: IRQ-Jumper	13
3.5.3	JP3: DMA-Jumper.....	13
3.5.4	The LED's.....	13
3.6	Card overview PCIGPIB1	14
3.6.1	Standard Jumpersettings for 5V Use.....	14
3.7	Connector's Pin assignment	16
3.8	Concept.....	17
4	Programming the GPIB-board	18
4.1	Software.....	18

4.1.1	Which software you need ?.....	18
4.1.2	The GPIB Test-program	18
4.2	QLIB: High Level Programming (Windows XP/2000 / NT4.x/ME/98/95).....	19
4.2.1	QLIB (QUANCOM Driver Library).....	19
4.2.2	Installation and general of the QLIB	20
4.2.3	Installation of the drivers and the QLIB for a QUANCOM PCI board or USB-Modul under Windows XP / 2000.....	20
4.2.4	Installation of the QLIB and the Drivers for a QUANCOM ISA board or an PAR-Modul under Windows XP / 2000	22
4.2.5	Installation of the drivers and the QLIB for a QUANCOM PCI board under Windows NT ..	24
4.2.6	Installation of the drivers and the QLIB for a QUANCOM ISA board and PAR-Modul under Windows NT.....	25
4.2.7	Installation of the drivers and the QLIB for a QUANCOM PCI board or USB-Modul under Windows ME / 98 / 95.....	27
4.2.8	Installation of the QLIB and the Drivers for a QUANCOM ISA board or an PAR-Modul under Windows ME / 98 / 95.....	29
5	Excerpt of some QLIB-Instructions for the PCIGPIB1/GPIB-1	32
5.1	Management functions..... Fehler! Textmarke nicht definiert.	
5.2	GPIB - routines..... Fehler! Textmarke nicht definiert.	
5.3	GPIB Visual-C/C++ Example	34
5.4	GPIB VB-Example	39
5.5	GPIB National Instruments LabView Example.....	44
5.6	GPIB Lotus Notes Example	45
5.7	GPIB Borland Delphi Example.....	53
6	The Device Driver (MS-Dos) for the GPIB-1	60
6.1	Installation	60
6.1.1	Command parameter	61
6.1.2	Examples	63
6.2	Programming examples in BASIC:	64
6.3	Programming examples in C:	65
6.4	Programming examples in PASCAL:.....	66
7	Annex	68
7.1	Customer Communication and Help.....	68

Overview

- 7.2 Technical support form..... 71**
- 7.3 Hardware and Software configuration form..... 72**
- 7.4 Documentation Comment Form 73**

1 Overview

1.1 Introduction

Congratulations! You've bought a QUANCOM high quality measurement and automation board, which shows the newest update of technology and whose attributes and functions are able to compete with every other instrument and even beats them. The following special attributes are included:

Features of the board

- Easy programming
- Various sample applications in different programming languages
- Driver support by Windows XP, 2000, NT and Me/98/95 with the **QLIB (QUANCOM Driver Library)**

1.2 Our experience is your profit

We from QUANCOM are specialists for the development of hard- and software. QUANCOM has grown to become one of the leading suppliers of measuring and automation technology to industry. At its design centres QUANCOM has developed an impressive range of products.

1.3 Customer Communication

QUANCOM wants to receive your comments on our products and manuals. We are interested in the applications you develop with our products, and we want to help if you have problems with them. To make it easy for you to contact us, this manual contains comment and configuration forms for you to complete. These forms are in chapter 6.2 "*Customer Communication and Help*" at the end of this manual.

1.4 Changes in this manual and software updates

QUANCOM - products are marked out by their constant further development. You can watch all the actual information of the changes in the README-file on the installation disk or CD. You can always get more information and free software updates on our internet website.

WWW.QUANCOM.DE

1.5 Scope of supply

- Measuring and automation board
- User's manual
- QUANCOM CD

If a component is missing please contact your dealer. QUANCOM reserves the right to change the extent of delivery without a preliminary announcement.

2 Installation procedures



2.1 System requirements

- Personal computer: The QUANCOM boards are assigned to operate in IBM-AT compatible computers with 80X86 or compatible. (i.e. 80386 / 80486 / Pentium)
- Bus: Your computer must have the corresponding bus. (PCI / ISA)

2.2 Safety precautions

For the sake of your security and of a safe function of your new QUANCOM board mind the following advice:

- Before opening the computer please unplug it.
- Computer motherboards and components contain very delicate integrated circuit (IC) chips. To protect them against damage from static electricity, you must follow some precautions whenever you work on your computer. Use a grounded wrist strap before handling computer components. If you don't have one, touch both of your hands to a safely grounded object or to a metal object, such as the power supply case.
- Hold components by the edges and try not to touch the integrated circuit chips, leads or circuitry.
- Place components on a grounded anti-static pad or on the bag that came with the component whenever the components are separated from the system.

! ATTENTION ! Modifications, made at the device without express permission of QUANCOM, lead to the loss

2.3 Installing the board

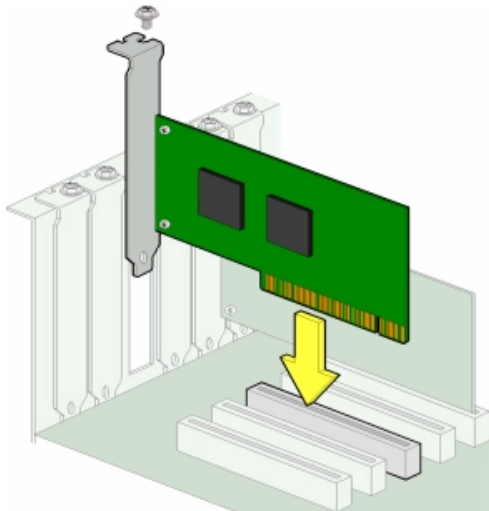
CAUTION:

1. Always turn the system power off and remove the power cord from the wall before installing or removing any device.
2. Always observe static electricity precautions.
See „*Safety precautions*“ in chapter 2.2

1. Switch off the computer and the connected devices and unplug them.

Warning: Static electricity can destroy your computer and the board!

Discharge yourself as described in chapter 3.2 “*Safety precautions*“.



2. To open your PC you have to detach the four safety screws on the back of the case with a screw driver. Then you can pull the cover forwards. If necessary you must remove impeding cables.

3. The slots are positioned at the back side of your computer. The back wall of unused slots is covered by a small metal plates. Search for a free slot, detach its holding screw and remove the small metal plate belonging to it.

4. Position the extension card into a free slot. Pay attention that the card is set firmly in the slot.

5. Fasten the board with the screw of the small metal plate on the back wall.

6. Close the cover of your computer. Cables, that you detached during the installation, should now be reconnected.

7. Connect the cable of the board into the slot belonging to it.

3 Technical Hardware Description

3.1 General

All instruments lie at the IEEE488 bus system parallel to the communication lines. The bus system consists of 16 signal lines: 8 data lines, 3 ,handshake' lines (data-byte-transfer-control) and 5 lines are used for the organization between the instruments.

Some instruments behave as transmitters (,talker') and some as receivers (,listener'). The GPIB controller card can handle both modes.

The QUANCOM GPIB card is used to connect devices with the wide-spread IEEE-488 interface to the PC. The interrupts 2,3,4,5,6,7 and DMA 1,2,3 are available. This gpib board has been sold more than 1000 times and it is the standard and cost effective solution for measurement with the IEEE-488 devices. This GPIB board uses the Standard NEC 7210 register settings and is therefore compatible with all significant applications and operating systems.

Therefore internet downloadable programs or source code, for example, can be used. The card can be connected to the PC with up to 15 peripheral devices having IEEE488 interfaces. All settings are adjusted by means of jumpers. The data exchange is controlled by an intellegent uPD7210 controller from NEC. Therefore, the card's function is compatible with almost all IEEE driver programs.

For a fast and easy implementation, a Terminal-Program and a Device-Driver for plotter- and measuring device control are included in the scope of supply.

3.2 Operation of the GPIB card

The GPIB addresses with it's data lines a specific instrument, determined by a sent address. This address is send to all instruments at the same time. Now every instrument compares his own address with the received one. If the device is addressed by the controller it enters the command mode. The controller card then gives to the initialized instruments the command to work in 'talker' or 'listener' mode. After this the devices receive the command to transfer the data. The controller supervises the devices an controls the data transmission between the devices.

3.3 Installation

Please set the appropriate I/O base address of the GPIB-1 card with the Jumper "JP1". See chapter 3.5.1 for further details.

3.4 Technical data

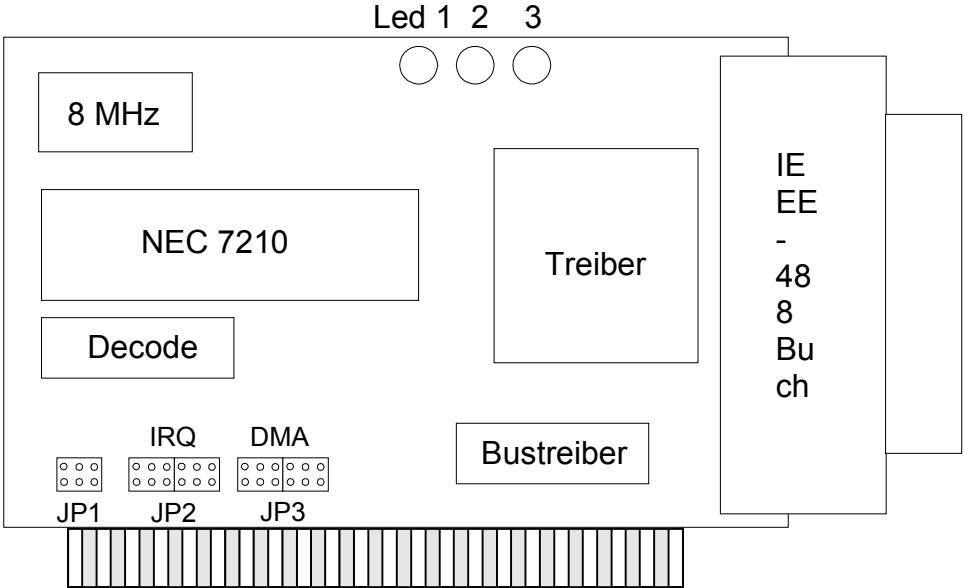
3.4.1 GPIB-1 : Short 8-Bit card for PC's with ISA-BUS:

Controller	NEC μ PD7210
PC-BUS	Interrupts switchable with: IRQ 2, 3, 4, 5, 6, 7
DMA	DMA 1,2 or 3 choice by Jumper.
Address decoding	7 different addresses coice by jumpers: [HEX] 1D0, 2B0, 300, 310, 330, 3B0, 3E0
Data bus	8-Bit Data bus, buffered
Plug connector	24 pol. IEEE-488 socket with M3.5 Thread.
LED:	3 LEDs this one for data and control lines is on the card with direction advertisement.
Software	Example software in C, PASCAL and Basic. Device-Driver for MS-DOS and other programming languages as well as to the direct plotter drive. GPIB Terminalprogram

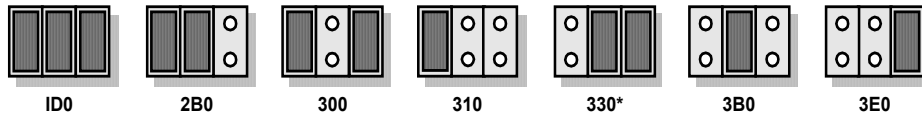
3.4.2 PCIGPIB1 : PCI IEEE-488 Interface with 8 diagnostic LED's

Controller:	μ PD7210 compatibel (with HW FIFO)
Connector:	24 pol. IEEE-488 jack with M3.5 thread.
LED:	8 LEDs are on the card, so show the controll lines ATN, SRQ, NDAC, NRFD, DAV, IFC, REN, EOI. 2 LEDs indicate the access to the board and the PCI status configuration.
Software:	Examplesoftware in C, PASCAL und Basic.

3.5 Card overview GPIB-1

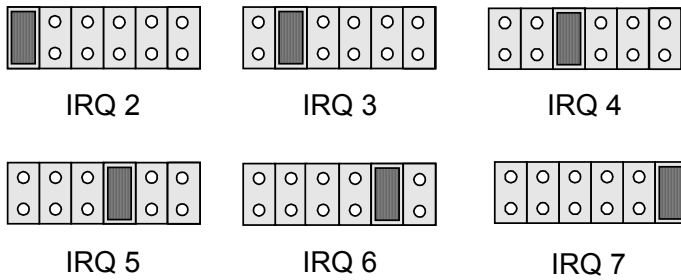


3.5.1 JP1 Base I/O Address setting



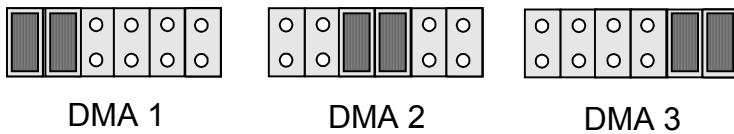
* Standard – Address

3.5.2 JP2: IRQ-Jumper



* Standard: no IRQ

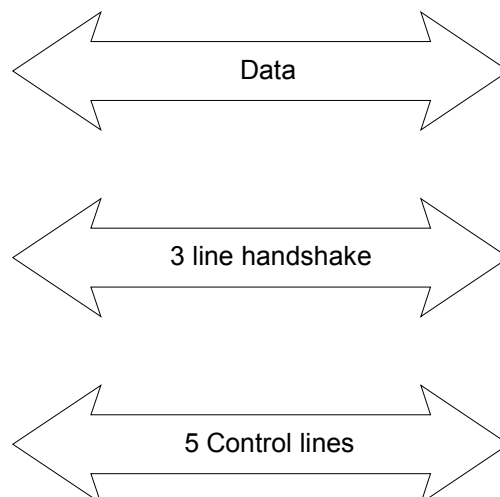
3.5.3 JP3: DMA-Jumper



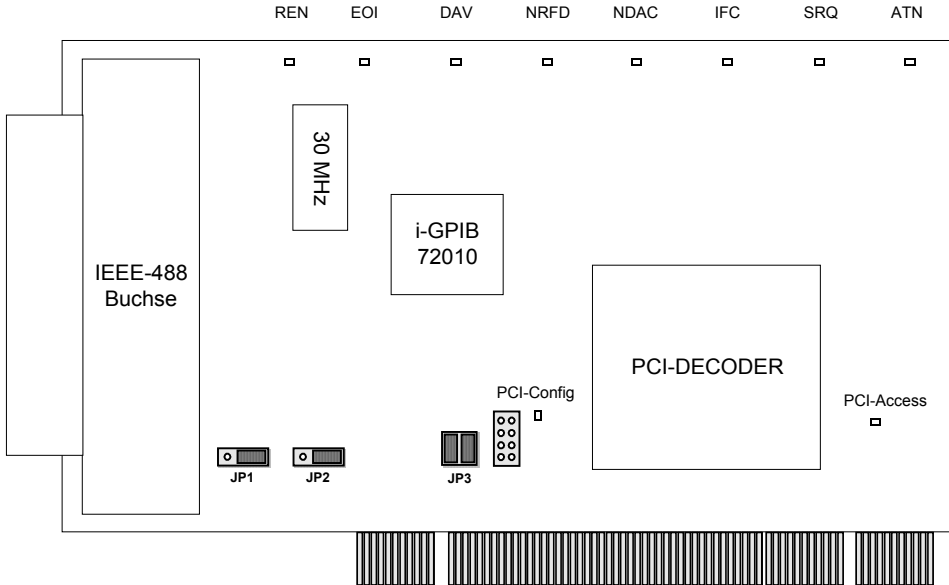
* Standard: no DMA

3.5.4 The LED's

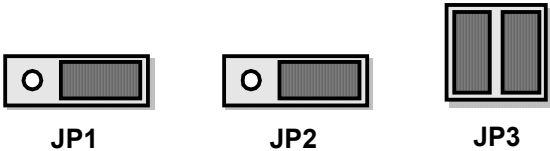
The 3 light-emitting diode (LED 1,2 and 3) signal the status of the following lines:
 (LED ON – GPIB card is sending data; LED OFF – GPIB card is receiving data):



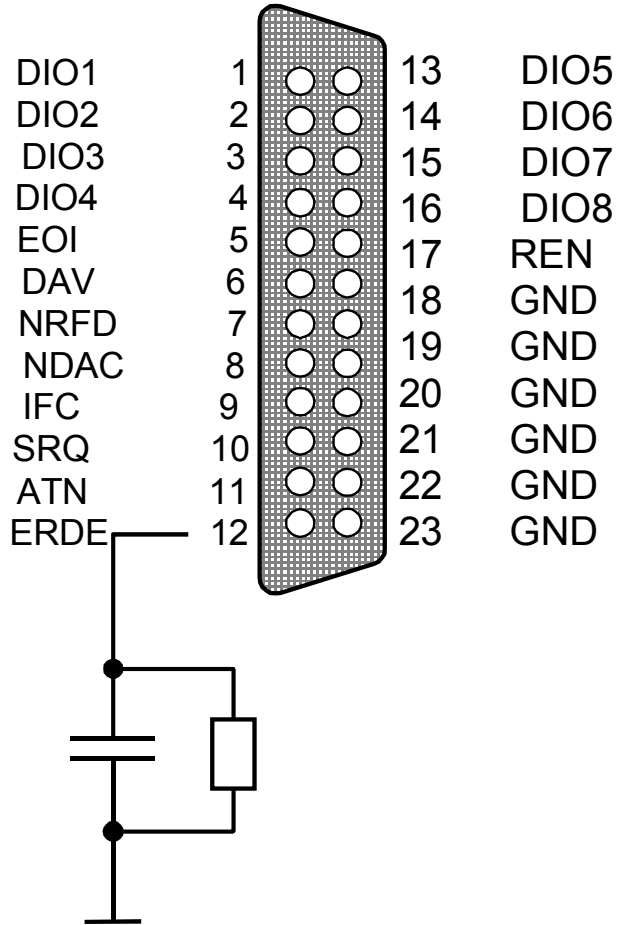
3.6 Card overview PCIGPIB1



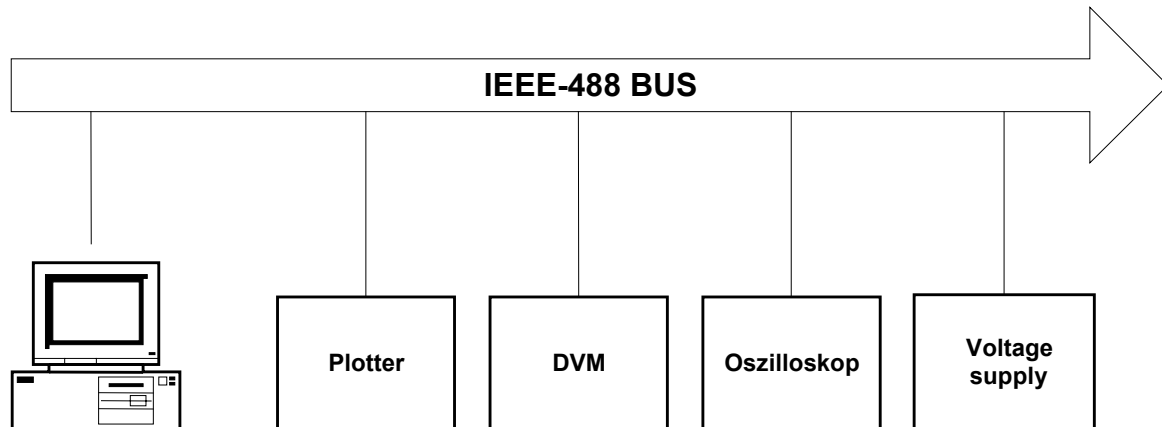
3.6.1 Standard Jumpersettings for 5V Use



3.7 Connector's Pin assignment



3.8 Concept



At most 30 instruments can be connected to the GPIB controller card. The instruments are in the example above Plotter, DVM, Oscilloscope and Voltage source connected to the GPIB interface.

You may define the DVM as 'talker' and the Voltage source as 'listener'. Now it is possible to transfer data to the Voltage source and to read the information from the DVM.

4 Programming the GPIB-board

4.1 Software

4.1.1 Which software you need ?

The software which one is necessary dependent on respective application and on the operating system, the following methods exists:

- **Method 1:** High-Level programming (Access to the board with QLIB) for Windows XP / 2000 / NT / ME / 98 / 95. You can use the card with Visual-C, Visual-Basic, Borland Delphi, Lotus Notes or with an other Compiler and Interpreter.
- **Method 2:** Installation of the QLIB in connection with an other program (examples for the work with National Instruments Labview are at web [http://www.quancom.de/qprod01/deu/files/download.gpib_vi.zip/\\$file/gpib_vi.zip](http://www.quancom.de/qprod01/deu/files/download.gpib_vi.zip/$file/gpib_vi.zip) or after the Qlib installation in the folder d:\programm files\quancom\qlib32\samples\gpib.)
- **Method 3:** To use the DEVICE-Driver for DOS / Windows 3.x / 95 / 98

If you use **method 1**, you need the source code of the application to use the Qlib instructions. If you would like to add instructions into your application, programming knowledge is necessary.

The **method 2** permits that the GPIB-board works with an existing software together. In this case, you must only install the QLIB of the QUANCOM CD.

4.1.2 The GPIB Test-program

We developed a tool to test the GPIB-board. This tool is depending upon kind of the QLIB (QUANCOM Driver LIBRARY) installation in your starting menu at **Start | Program | QLIB 32-Bit | Program | PCIGPIB, ISA GPIB-1 and USBGPIB Controller**. You find the source code of the test-program at "d:\programm files\quancom\qlib32\samples\gpib\vb".

4.2 QLIB: High Level Programming (Windows XP/2000 / NT4.x/ME/98/95)

4.2.1 QLIB (QUANCOM Driver Library)

The QLIB (the abbreviation for **QUANCOM LIB**rary) offers the possibility of addressing all QUANCOM-Boards under the operating systems Windows XP/2000/NT and ME/98/95 and the programming languages C/C++/Delphi/Visual Basic. It is provided and permitted to all QUANCOM-Boards by the simplicity of the instructions the user to merge the QLIB into own applications. The instructions and functions apply to all operating systems.

Supported operating systems:

- Microsoft Windows XP / 2000 / NT 4.0
- Microsoft Windows ME / 98 / 95

Supported compiler:

C / C++

- Borland C++ 3.1, 4.x, 5.x
- Microsoft® Visual C++ 1.x, 2.x, 4.x, 5.x

Pascal

- Borland Turbo Pascal

Delphi

- Borland Delphi

Basic

- Microsoft® Visual Basic 3.x, 4.x, 5.x; 6.x

Graphic programming language

- Agilent VEE von Agilent
- LabView® von National Instruments

4.2.2 Installation and general of the QLIB

Programming all QUANCOM-Boards independently what for a operating system you use. The QLIB (**QUANCOM LIB**rary) offers the possibility of addressing all QUANCOM-Boards under the operating systems Windows XP/2000/NT and ME/98/95 and the programming languages C/C++/Delphi/Visual Basic with simple instructions. The QLIB is provided to all QUANCOM-Boards and facilitates for the user the integration into own applications.

Attention ! There are for the different Operating Systems and Bussystems (PCI, ISA, USB, PAR) special Installation instructions. Please choose the Installation for your board and your operating system at the following chapters.

4.2.3 Installation of the drivers and the QLIB for a QUANCOM PCI board or USB-Modul under Windows XP / 2000

If you downloaded the QLIB of our Internet page, please read first the information of the Readme before you install the drivers.

1. Step : Driver installation:

After you put the board into a free Slot, please start your system. Windows would recognize automatically the new board. Windows will ask you for a driver for the new founded board. Please put the **QLIB Installation CD** in your CD Drive (f.e. "D".)

- If Windows the dialogue box "new hardware found" when starting indicates, selects you "Next".
- After it, you select please "search for a suitable driver for the equipment (recommended)", then confirm with "Next".
- As source for the search for the drivers you select please "Choose the destination" and press "Next".
- Windows opens now an dialog to chose an file. Please click on the button „Choose“. Change to the CD-Drive and choose the folder Win2000 or WinXP, it depends on for what for an Operating Systems you want to install the QLIB. Windows would find automatically the file QLIBXDRV.INF. Please click on „OK“ to go ahead with the Installation of the driver.

- Confirming it in the next window with “next” that you want to install the driver.
- In the following window you close the installation of the QUANCOM driver with click on the button “finishing”.

2. Step : QLIB installation:

After the driver installation of the QUANCOM board must you still install the **QLIB Software**.

- Please click on **Start | Implement** and choose the programm drive:\qlib32.exe, click on „OK“. (Please use for **drive** the driveletter of your CD-ROM-Device)
- If after starting the programm Qlib32.exe a messagebox appears, which states that your system must restart to install the QLIB, please click on „YES“ to restart your system. (*After the reboot, the installation would continued automatically*)
(This step are only available if you had no version of the Installer on your system)
- In the following window click on „Next“ to continue the installation. (Please consider before beginning the installation that all current windows programm are terminated)
- After it, please confirm the Licenseagreement of the QLIB-Software with click on „I accept the license agreement“ and lead the installation with press the „Next“ Button.
- Please enter now your personal user data (name; organisation;). Select aferwards whether you want to install the QLIB-Software only for the current user or for all users of this computer (only with administrator rights possible). Confirm your choice with click on „Next“.
- To change the installation path of the QLIB-Software, please click on „Browse“. The installation continued with clicking on „Next“.
- Please select now the kind of installation of the software and confirm these with „Next“. Three possibilities are available :
- Typical (without examples and helpfiles)
- Complete (with examples and helpfiles)
- Custom (free selectabel)
- The installation starts immediately with a click on „Next“.
- After the succesfull installation of the QLIB-Software, the installation programm closed with a click on the „Finishing“ Button.
- To take over the changes at your computer, you will be requested to “Restart“ your system, this happens when you press the „YES“ Button.

4.2.4 Installation of the QLIB and the Drivers for a QUANCOM ISA board or an PAR-Modul under Windows XP / 2000

If you downloaded the QLIB of our Internet page, please read first the information of the Readme before you install the drivers.

1. Step : QLIB Installation :

After you put the board into a free Slot (please note the Jumpersettings on the GPIB-board), please start your system. Please put the **QLIB Installation CD** in your CD Drive (f.e. "D".)

- Please click on **Start | Implement** and choose the programm d:\qlib32.exe, click on „OK“.
- If after starting the programm Qlib32.exe a messagebox appears, which states that your system must restart to install the QLIB, please click on „YES“ to restart your system. *(After the reboot, the installation would continued automatically)* (This step are only available if you had no version of the Installer on your system)
- In the following window click on „Next“ to continue the installation. (Please consider before beginning the installation that all current windows programm are terminated)
- After it, please confirm the Licenseagreement of the QLIB-Software with click on „I accept the license agreement“ and lead the installation with press the „Next“ Button.
- Please enter now your personal user data (name; organisation;). Select aferwards whether you want to install the QLIB-Software only for the current user or for all users of this computer (only with administrator rights possible). Confirm your choice with click on „Next“.
- To change the installation path of the QLIB-Software, please click on „Browse“. The installation continued with clicking on „Next“.
- Please select now the kind of installation of the software and confirm these with „Next“. Three possibilities are available :
 - Typical (without examples and helpfiles)
 - Complete (with examples and helpfiles)
 - Custom (free selectabel)
- The installation starts immediatly with a click on „Next“.
- After the succesfull installation of the QLIB-Software, the installation programm closed with a click on the „Finishing“ Button.

- To take over the changes at your computer, you will be requested to “Restart” your system, this happens when you press the „YES“ Button.

2. Step: Driver installation for ISA board and Parallel Port Module:

Step 2 installs the standard driver for the ISA boards. You install the driver at the system control. Please click on **Start | Settings | System control**.

- Doubleclick on the icon „Hardware“
- Click “Next”
- Please choose “Add Device” and click on “Next”.
- Select in the following window “No, select the hardware components themselves in the list” and click on “Next”.
- In the now appearing list, please select the QLIB Devices and confirm your selection with click on “Next”
- Now select your bought board in the list of the Quancom boards and confirm with a click on “Next” and “OK”
- Please adjust now the I/O range, which is adjusted by Jumper on your ISA board and confirm your selection with twice click on “OK”.
- To install the board click on “Next”.
- After you had clicked on “Finishing”, you must restart your system with clicking on “Restart”.

4.2.5 Installation of the drivers and the QLIB for a QUANCOM PCI board under Windows NT

If you downloaded the QLIB of our Internet page, please read first the information of the Readme before you install the drivers.

1. Step : QLIB installation:

After you put the board into a free Slot, please start your system. Windows would recognize automatically the new board. Windows will ask you for a driver for the new founded board. Please put the **QLIB Installation CD** in your CD Drive (f.e. "D".)

- Please click on **Start | Implement** and choose the programm drive:\qlib32.exe, click on „OK“. (Please use for **drive** the driveletter of your CD-ROM-Device)
- If after starting the programm Qlib32.exe a messagebox appears, which states that your system must restart to install the QLIB, please click on „YES“ to restart your system. (*After the reboot, the installation would continued automatically*) (This step are only available if you had no version of the Installer on your system)
- In the following window click on „Next“ to continue the installation. (Please consider before beginning the installation that all current windows programm are terminated)
- After it, please confirm the Licenseagreement of the QLIB-Software with click on „I accept the license agreement“ and lead the installation with press the „Next“ Button.
- Please enter now your personal user data (name; organisation;). Select aferwards whether you want to install the QLIB-Software only for the current user or for all users of this computer (only with administrator rights possible). Confirm your choice with click on „Next“.
- To change the installation path of the QLIB-Software, please click on „Browse“. The installation continued with clicking on „Next“.
- Please select now the kind of installation of the software and confirm these with „Next“. Three possibilities are available :
- Typical (without examples and helpfiles)
- Complete (with examples and helpfiles)
- Custom (free selectabel)
- The installation starts immediatly with a click on „Next“.
- After the succesfull installation of the QLIB-Software, the installation programm closed with a click on the „Finishing“ Button.

- To take over the changes at your computer, you will be requested to “Restart” your system, this happens when you press the „YES“ Button.

4.2.6 Installation of the drivers and the QLIB for a QUANCOM ISA board and PAR-Modul under Windows NT

If you downloaded the QLIB of our Internet page, please read first the information of the Readme before you install the drivers.

1. Step : QLIB installation:

After you put the board into a free Slot, please start your system. Windows would recognize automatically the new board. Windows will ask you for a driver for the new founded board. Please put the **QLIB Installation CD** in your CD Drive (f.e. “D”).

- Please click on **Start | Implement** and choose the programm drive:\qlib32.exe, click on „OK“. (Please use for **drive** the driveletter of your CD-ROM-Device)
- If after starting the programm Qlib32.exe a messagebox appears, which states that your system must restart to install the QLIB, please click on „YES“ to restart your system. (*After the reboot, the installation would continued automatically*) (This step are only available if you had no version of the Installer on your system)
- In the following window click on „Next“ to continue the installation. (Please consider before beginning the installation that all current windows programm are terminated)
- After it, please confirm the Licenseagreement of the QLIB-Software with click on „I accept the license agreement“ and lead the installation with press the „Next“ Button.
- Please enter now your personal user data (name; organisation;). Select aferwards whether you want to install the QLIB-Software only for the current user or for all users of this computer (only with administrator rights possible). Confirm your choice with click on „Next“.
- To change the installation path of the QLIB-Software, please click on „Browse“. The installation continued with clicking on „Next“.
- Please select now the kind of installation of the software and confirm these with „Next“. Three possibilities are available :
- Typical (without examples and helpfiles)
- Complete (with examples and helpfiles)
- Custom (free selectabel)

- The installation starts immediately with a click on „Next“.
- After the successful installation of the QLIB-Software, the installation program closed with a click on the „Finishing“ Button.
- To take over the changes at your computer, you will be requested to “Restart” your system, this happens when you press the „YES“ Button.

2. Step: Driver installation for ISA board and Parallel Port Module:

Step 2 installs the standard driver for the ISA boards. You install the driver with the utility QLIBCNFG.EXE. You find the Utility at **Start | Programs | QLIB 32-Bit | QLIB 32-Bit Configuration Utility**.

- Please click on the icon „Add“
- Now select your bought board in the list of the Quancom boards and confirm with a click on “OK“
- In the following automatically dialog can you select the I/O Address of the board. Please check the displayed I/O Address of the board with the jumpered Address on your board. If the Address is not the same, please click double on “I/O Basisadresse”. Adjust now with the Pageup and down arrow the the I/O range, which is adjusted by Jumper on your ISA board.

4.2.7 Installation of the drivers and the QLIB for a QUANCOM PCI board or USB-Modul under Windows ME / 98 / 95

If you downloaded the QLIB of our Internet page, please read first the information of the Readme before you install the drivers.

1. Step : Driver installation:

After you put the board into a free Slot, please start your system. Windows would recognize automatically the new board. Windows will ask you for a driver for the new founded board. Please put the **QLIB Installation CD** in your CD Drive (f.e. "D".)

- If Windows the dialogue box "new hardware found" when starting indicates, selects you "Next".
- After it, you select please "search for a suitable driver for the equipment (recommended)", then confirm with "Next".
- As source for the search for the drivers you select please "Choose the destination" and press "Next".
- Windows opens now an dialog to chose an file. Please click on the button „Choose“. Change to the CD-Drive and choose the folder WinME, Win98 or Win95, it depends on for what for an Operating Systems you want to install the QLIB. Windows would find automatically the file QLIBXDRV.INF. Please click on „OK“ to go ahead with the Installation of the driver.
- Confirming it in the next window with “next” that you want to install the driver.
- In the following window you close the installation of the QUANCOM driver with click on the button “finishing”.

2. Step : QLIB installation:

After the driver installation of the QUANCOM board must you still install the **QLIB Software**.

- Please click on **Start | Implement** and choose the programm drive:\qlib32.exe, click on „OK“. (Please use for **drive** the driveletter of your CD-ROM-Device)
- If after starting the programm Qlib32.exe a messagebox appears, which states that your system must restart to install the QLIB, please click on „YES“ to restart your system. *(After the reboot, the installation would continued automatically)*
(This step are only available if you had no version of the Installer on your system)

- In the following window click on „Next“ to continue the installation. (Please consider before beginning the installation that all current windows programmes are terminated)
- After it, please confirm the Licenseagreement of the QLIB-Software with click on „I accept the license agreement“ and lead the installation with press the „Next“ Button.
- Please enter now your personal user data (name; organisation;). Select afterwards whether you want to install the QLIB-Software only for the current user or for all users of this computer (only with administrator rights possible). Confirm your choice with click on „Next“.
- To change the installation path of the QLIB-Software, please click on „Browse“. The installation continued with clicking on „Next“.
- Please select now the kind of installation of the software and confirm these with „Next“. Three possibilities are available :
 - Typical (without examples and helpfiles)
 - Complete (with examples and helpfiles)
 - Custom (free selectabel)
- The installation starts immediately with a click on „Next“.
- After the succesfull installation of the QLIB-Software, the installation programm closed with a click on the „Finishing“ Button.
- To take over the changes at your computer, you will be requested to “Restart“ your system, this happens when you press the „YES“ Button.

4.2.8 Installation of the QLIB and the Drivers for a QUANCOM ISA board or an PAR-Modul under Windows ME / 98 / 95

If you downloaded the QLIB of our Internet page, please read first the information of the Readme before you install the drivers.

1. Step : QLIB Installation :

After you put the board into a free Slot (please note the Jumpersettings on the GPIB-board), please start your system. Please put the **QLIB Installation CD** in your CD Drive (f.e. "D".)

- Please click on **Start | Implement** and choose the programm d:\qlib32.exe, click on „OK“.
- If after starting the programm Qlib32.exe a messagebox appears, which states that your system must restart to install the QLIB, please click on „YES“ to restart your system. *(After the reboot, the installation would continued automatically)*
(This step are only available if you had no version of the Installer on your system)
- In the following window click on „Next“ to continue the installation. (Please consider before beginning the installation that all current windows programm are terminated)
- After it, please confirm the Licenseagreement of the QLIB-Software with click on „I accept the license agreement“ and lead the installation with press the „Next“ Button.
- Please enter now your personal user data (name; organisation;). Select aferwards whether you want to install the QLIB-Software only for the current user or for all users of this computer (only with administrator rights possible). Confirm your choice with click on „Next“.
- To change the installation path of the QLIB-Software, please click on „Browse“ . The installation continued with clicking on „Next“.
- Please select now the kind of installation of the software and confirm these with „Next“. Three possibilities are available :
- Typical (without examples and helpfiles)
- Complete (with examples and helpfiles)
- Custom (free selectabel)
- The installation starts immediately with a click on „Next“.
- After the succesfull installation of the QLIB-Software, the installation programm closed with a click on the „Finishing“ Button.

- To take over the changes at your computer, you will be requested to “Restart” your system, this happens when you press the „YES“ Button.

2. Step: Driver installation for ISA board and Parallel Port Module:

Step 2 installs the standard driver for the ISA boards. You install the driver at the system control. Please click on **Start | Settings | System control**.

- Doubleclick on the icon „Hardware“
- Click “Next”
- Please choose “Add Device” and click on “Next”.
- Select in the following window “No, select the hardware components themselves in the list” and click on “Next“.
- In the now appearing list, please select the “Unknown Device”, “Other Devices” or QLIB Devices if a QLIB board is installed and confirm your selection with click on “Next“
- Now select your bought board in the list of the Quamcom boards and confirm with a click on “Next“ and “OK“
- Please adjust now the I/O range, which is adjusted by Jumper on your ISA board and confirm your selection with twice click on “OK“.
- To install the board click on “Next“.
- After you had clicked on “Finishing”, you must restart your system with clicking on “Restart”.

5 Excerpt of some QLIB-Instructions for the PCIGPIB1 / GPIB-1

Enclosed you see a excerpt of the QLIB instructions for the QUANCOM GPIB board. You find a detailed explanation of all QLIB instructions with appropriate installation in the QLIB Windows assistance in your starting menu.

5.1 Simple QLIB-Instructions

5.1.1 Management functions

QAPINumOfCards

ULONG QAPINumOfCards (void);

With the function **QAPINumOfCards** it is possible to ask , which used cards are supported by the QLIB

5.1.2 GPIB - routines

QAPIWriteString

ULONG QAPIWriteString (ULONG cardid ULONG device char* buffer ULONG chars ULONG mode);

With the function **QAPIWriteString** a string will be sent to the GPIB board

QAPIReadString

ULONG QAPIReadString (ULONG cardid ULONG device char* buffer ULONG chars ULONG mode);

With the function **QAPIReadString** a string will be receive from a device.

QAPISpecial

ULONG QAPISpecial (ULONG cardid ULONG jobcode ULONG para1 ULONG para2);

With this function **QAPISpecial** it is possible to run card specific functions

5.2 Extended QLIB-Instructions

5.2.1 Management functions

QAPIExtOpenCard

ULONG QAPIExtOpenCard (ULONG cardid, ULONG devnum);

Use the function QAPIExtOpenCard to open a board and retrieve the board handle

QAPIExtCloseCard

void QAPIExtCloseCard(ULONG cardhandle);

With the function QAPIExtCloseCard the board is closed

QAPIExtNumOfCards

ULONG QAPIExtNumOfCards (void);

With the function QAPIExtNumOfCards it is possible to ask , which used cards are supported by the QLIB

5.2.2 GPIB - routines

QAPIExtWriteString

ULONG QAPIExtWriteString (ULONG cardhandle ULONG device char* buffer ULONG chars ULONG mode);

With the function **QAPIExtWriteString** a string will be sent to the GPIB board.

QAPIExtReadString

ULONG QAPIExtReadString (ULONG cardhandle ULONG device char* buffer ULONG chars ULONG mode);

With the function **QAPIExtReadString** a string will be receive from a device.

QAPIExtSpecial

ULONG QAPIExtSpecial (ULONG cardhandle ULONG jobcode ULONG para1 ULONG para2);

With this function QAPIExtSpecial it is possible to run card specific functions.

5.3 GPIB Visual-C/C++ Example

The fat-printed text describes, how you can include the QLIB in a MFC / C++ / C project. After the installation of the QLIB, you find the immediately compilable Example project at the folder *d:\program files\quancom\qlib32\samples\gplib\vc* (only if you install at the standard folder).

```
// gplib.cpp : Sample project for the GPIB-1, PCIGPIB and USB GPIB Modules
//
// Author: Michael Reimer, QUANCOM Informationssysteme GmbH, Germany
//
// Website: http://www.quancom.de
//
// Information:
//
// To use the QLIB Commands in your source, do the following:
//
// (1) Add statement #include "qlib.h" to your source file.
// (2) Add in the Dialog Menu->Project->Settings->C/C++>Preprocessor
// "$(QLIB_INC)" to the additional include directories entry.
// (3) Add in the Dialog Menu->Project->Settings->Linker->General
// "$(QLIB_LIB)\qlib32.lib" to the additional library and object
// modules directories entry.

#include "stdafx.h"
#include "windows.h"
#include "qlib.h"

int main(int argc, char* argv[])
{
    ULONG result;
    ULONG listener;
    ULONG talker;
    char buffer[1024];

    // The following sequence tries to find the
    // Bus Type ( PCI, ISA or USB ) of the
    // installed GPIB Controller

    ULONG handle = QAPIExtOpenCard(PCIGPIB,0);
    if ( handle == NULL )
    {
        handle = QAPIExtOpenCard(USBGPIB,0);
        if ( handle == NULL )
        {
            handle = QAPIExtOpenCard(GPIB,0);
        }
    }

    // The handle is != NULL if there is a GPIB Controller
    // installed

    if ( handle == NULL )
    {
        MessageBox(NULL, "No QUANCOM GPIB Controller found!", "Error", MB_OK);
    }
}
```

```
        return FALSE;
    }

    // Ok, we found a QUANCOM GPIB Controller Card
    // Now we can send a string to the listner with address 3
    // Change the address to the appropriate address for your
    // device ( normally set by DIP-Switches on the back side )
    // -----
    // PART 1a: Writing a string to the DMM ( The DMM is a the // listener )
    //
    // Listener: A device capable of receiving data over the
    // interface
    // when addressed to Listen by the active controller.
    // Examples of such
    // devices are printers, programmable power supplies, or any
    // other programmable instrument. There can be up to 14
    // Listeners on the GPIB Bus at one time.
    // -----
    // Select the listener address, which is set by a DIP-Switch
    // on the back side of your instrument.

    listener = 3;

    // Send "z" to the listner 3, which resets the DMM to the
    // initial settings.

    char s1[] = "z";
    if ( QAPIExtWriteString(handle, listener, (char*)&s1, strlen(s1),0)
        {
            printf("Writing to device %u was successful\n", listener);
        }
    else
        {
            printf("Writing to device %u failed\n", listener);
        }

    // -----
    // PART 1b: Reading a DMM ( The DMM is a talker )
    //
    // Talker: A device capable of transmitting data over the
    // interface when addressed to talk by the active
    // controller. Examples of such devices are voltmeters,
    // data-acquisition systems, or any other programmable
    // instrument. There can be only one addressed talker on the // GPIB Bus at one time.
    // -----
    // Select the talker address, which is set by a DIP-Switch
    // on the back side of your instrument.

    talker = 3;

    // Read value from DMM with talker address 3

    result = QAPIExtReadString(handle, talker, (char*)&buffer, sizeof(buffer), 0);
    if (result)
        {
            printf("Reading from device %u was: %s\n",talker,  buffer);
        }
    else
        {
            printf("Reading from device %u failed\n", talker);
        }
}
```

```

// -----
// PART2: Checking whether a device has requested service
// (SRQ Service Request)
// A device can interrupt the active controller by asserting
// the SRQ line. The SRQ is a single line, and if there are
// multiple devices on the GPIB Bus that have been
// configured to assert an SRQ, the active controller will // have to "poll" the devices to
// figure out which one
// actually asserted the SRQ.
// More than one device could in principle assert an SRQ at // the same time. The active
// controller can poll the devices
// in one of two ways: serial poll or parallel poll.
// // -----

result = QAPIExtSpecial(handle, JOB_READSRQ, 0, 0);
if (result)
    {
        printf("SRQ was asserted\n");
    }
else
    {
        printf("SRQ was not asserted\n");
    }

// -----
// PART3: Reading Serial Poll Status from DMM
//
// In a serial poll, the active controller asks each device
// in turn to send back a status byte that indicates whether // the device has asserted the
// SRQ. Bit 6 of this
// byte (where the bits are numbered 0 through 7) is set
// if the device is requesting service. The definition of
// the other bits is device dependent (under 488.1 at least;
// 488.2 provides a much more concise definition of the
// status byte).
// The program has to perform this same sequence with every
// device it needs to poll.
// -----
// We poll all devices from 1 to 15 here. This is normally
// not necessary and very time consuming. Poll only valid // devices on the GPIB Bus.

ULONG serial_poll_byte = 0;
for ( ULONG device=1;device<15; device++ )
    {
        result = QAPIExtSpecial(handle, JOB_SERIALPOLL, device, (ULONG)&serial_poll_byte);
        if ( result )
            {
                if (serial_poll_byte & 0x40)

                    // check for bit 6 = device requested service

                    {
                        printf("Device %u requested service
and returned status byte %u\n",
device, serial_poll_byte);
                    }
            }
        else
    }

```

```

        {
            printf("Device %u no SRQ requested.
                Status is %u \n", device,
                serial_poll_byte);
        }
    else
    {
        printf("No answer from device %u\n",
            device);
    }
}

// -----
// PART4: Send a command to DMM
//
// The following commands are accepted by all devices on the
// GPIB Bus simultaneously. The address part will be
// ignored.
//
// - JOB_DCL (Device Clear): The DCL command causes all
// devices to return to a device dependent initial state.
//
// - JOB_LLO (Local Lockout): The LLO command disables the
// return-to-local front
// panel key on the device. The user can no longer change
// the device settings from its front panel.
//
// The following commands need an address and are only
// accepted by addressed devices. Whether the devices are
// the listeners or the talkers depends on the command. The
// three commands are as follows:
//
// - JOB_GET (Group Execute Trigger): The GET command tells
// all the addressed
// listeners to perform some device-dependent function,
// like take a measurement.
// GET allows for synchronizing a measurement function
// between multiple devices.
// This is only used in specialized cases.
//
// - JOB_SDC (Selected Device Clear): The SDC command resets
// the addressed listeners to a device-dependent state. It
// performs the same function as DCL,
// but only resets the addressed listeners, not all the
// devices.
//
// - JOB_GTL (Go To Local): The GTL command sets the
// addressed listeners back to
// local mode.
// -----
// Send DCL to all devices

result = QAPIExtSpecial(handle, JOB_DCL, 1, NULL);
if ( !result )
{
    printf("Command DCL failed\n");
}

// Send LLO to all devices

```

```
result = QAPIExtSpecial(handle, JOB_LLO, 1, NULL);
if ( !result )
{
    printf("Command LLO failed\n");
}

// Send GET to device 3

result = QAPIExtSpecial(handle, JOB_GET, 3, NULL);
if ( !result )
{
    printf("Command GET failed\n");
}

// Send SDC to device 3

result = QAPIExtSpecial(handle, JOB_SDC, 3, NULL);
if ( !result )
{
    printf("Command SDC failed\n");
}

// Send GTL to device 3

result = QAPIExtSpecial(handle, JOB_GTL, 3, NULL);
if ( !result )
{
    printf("Command GTL failed\n");
}
QAPIExtCloseCard(handle);
return 0;
```

5.4 GPIB VB-Example

The fat-printed text describes, how you can include the QLIB in a Visual-Basic project. After the installation of the QLIB, you find the immediately compilable Example project at the folder *d:\program files\quancom\qlib32\samples\gplib\vb* (only if you install at the standard folder).

```
' gplib.cpp : Sample project for the GPIB-1, PCIGPIB and USB GPIB Modules
'
' Author: Michael Reimer, QUANCOM Informationssysteme GmbH, Germany
'
' Website: http://www.quancom.de
'
' Information:
'
' To use the QLIB Commands in your source, do the following:
' (1) Install the QLIB ( QUANCOM Driver Library )
' (2) Add module "qlib.bas" to your project.
'

Sub Main()

Dim s As String
Dim handle As Long
Dim result As Long
Dim register As Integer
Dim nCardID As Integer
Dim nListener As Long
Dim serial_poll_byte As Long

' The following sequence tries to find the
' Bus Type ( PCI, ISA or USB ) of the
' installed GPIB Controller

handle = QAPIExtOpenCard(PCIGPIB, 0)

If (handle = 0) Then

    handle = QAPIExtOpenCard(GPIB, 0)

    If (handle = 0) Then

        handle = QAPIExtOpenCard(USBGPIB, 0)

    End If

End If

If (handle = 0) Then
    MsgBox "Unable to find GPIB Controller Card!", 16, "Error"
    Exit Sub
End If

' Ok, we found a QUANCOM GPIB Controller Card
'
' Now we can send a string to the listner with address 3
' Change the address to the appropriate address for your
```

```
' device ( normally set by DIP-Switches on the back side )
'
'-----
' PART 1a: Writing a string to the DMM ( The DMM is a the listener )
'
' Listener: A device capable of receiving data over the interface
' when addressed to Listen by the active controller. Examples of
' such devices are printers, programmable power supplies, or any
' other programmable instrument. There can be up to 14 Listeners on
' the GPIB Bus at one time.
'-----
'
' Select the listener address, which is set by a DIP-Switch on the
' back side of your instrument.
'
nListener = 3
s = "z"

' Send "z" to the listner 3, which resets the DMM to the initial settings.

result = QAPIExtWriteString(handle, nListener, s, Len(s), 0)

If (result) Then
    ' data send to gpib device
Else
    ' failure
    MsgBox "Failure sending data!", vbInformation, "Information"
End If

'-----
' PART 1b: Reading a DMM ( The DMM is a talker )
'
' Talker: A device capable of transmitting data over the interface
' when addressed to talk by the active controller. Examples of such
' devices are voltmeters, data-acquisition systems, or any other
' programmable instrument. There can be only one addressed talker on
' the GPIB Bus at one time.
'-----
'
' Select the talker address, which is set by a DIP-Switch on the
' back side of your instrument.
'
nListener = 3
s = Space$(1024) ' create buffer 1024 chars
'
' Read value from DMM with talker address 3

result = QAPIExtReadString(handle, nListener, s, Len(s), 0)

If (result) Then
    ' data successfully read from gpib device

    MsgBox "Read from device " & nListener & " String " & s

Else
    ' failure
    MsgBox "Failure reading data!", vbInformation, "Information"
End If

'-----
```

```
' PART2: Checking whether a device has requested service ( SRQ
' Service Request)
'
' A device can interrupt the active controller by asserting the SRQ
' line. The SRQ is a single line, and if there are multiple devices
' on the GPIB Bus that have been configured to assert an SRQ, the
' active controller will have to "poll" the devices to figure out
' which one actually asserted the SRQ. More than one device could in
' principle assert an SRQ at the same time. The active controller
' can poll the devices in one of two ways: serial poll or parallel
' poll.
' -----

result = QAPIExtSpecial(handle, JOB_READSRQ, 0, 0)

If (result = 0) Then
    MsgBox "No service requested", 16, "Information"
Else
    MsgBox "Device has requested service ( SRQ )", 16, "Information"
End If

' -----

' PART3: Reading Serial Poll Status from DMM
'
' In a serial poll, the active controller asks each device in turn
' to send back a status byte that indicates whether the device has
' asserted the SRQ. Bit 6 of this byte (where the bits are numbered
' 0 through 7) is set if the device is requesting service. The
' definition of the other bits is device dependent (under 488.1 at
' least; 488.2 provides a much more concise definition of the status
' byte).
'
' The program has to perform this same sequence with every device
' it needs to poll.
' -----

' We poll all devices from 1 to 15 here. This is normally not
' necessary and very time consuming. Poll only valid devices on the
' GPIB Bus.

serial_poll_byte = 0

For nListener = 1 To 15

result = QAPIExtSpecialSP(handle, JOB_SERIALPOLL, nListener, serial_poll_byte)

    If (result) Then
        ' data successfully read from gpib device
        MsgBox "Serial poll from device " & nListener & " returns = " & Val(serial_poll_byte)
    Else
        ' failure
        MsgBox "Failure reading data!", vbInformation, "Information"
    End If

Next nListener

' -----

' PART4: Send a command to DMM
'
' The following commands are accepted by all devices on the GPIB Bus
' simultaneously. The address part will be ignored.
```

```
'
' - JOB_DCL (Device Clear): The DCL command causes all devices to
' return to a device dependent initial state.
'
' - JOB_LLO (Local Lockout): The LLO command disables the return-to-
' local front panel key on the device. The user can no longer
' change the device settings from its front panel.
'
' The following commands need an address and are only accepted by
' addressed devices. Whether the devices are the listeners or the
' talkers depends on the command. The three commands are as follows:
'
' - JOB_GET (Group Execute Trigger): The GET command tells all the
' addressed listeners to perform some device-dependent function,
' like take a measurement. GET allows for synchronizing a
' measurement function between multiple devices. This is only used
' in specialized cases.
'
' - JOB_SDC (Selected Device Clear): The SDC command resets the
' addressed listeners to a device-dependent state. It performs the
' same function as DCL, but only resets the addressed listeners,
' not all the devices.
' - JOB_GTL (Go To Local): The GTL command sets the addressed
' listeners back to local mode.
'
```

```
' Send DCL to all devices
```

```
result = QAPIExtSpecial(handle, JOB_DCL, 1, 0)
```

```
If (Not result) Then
  MsgBox "Command DCL failed!", 16, "Information"
End If
```

```
' Send LLO to all devices
```

```
result = QAPIExtSpecial(handle, JOB_LLO, 1, 0)
```

```
If (Not result) Then
  MsgBox "Command LLO failed!", 16, "Information"
End If
```

```
' Send GET to device 3
```

```
result = QAPIExtSpecial(handle, JOB_GET, 3, 0)
```

```
If (Not result) Then
  MsgBox "Command GET failed!", 16, "Information"
End If
```

```
' Send SDC to device 3
```

```
result = QAPIExtSpecial(handle, JOB_SDC, 3, 0)
```

```
If (Not result) Then
  MsgBox "Command SDC failed!", 16, "Information"
End If
```

```
' Send GTL to device 3
```

```
result = QAPIExtSpecial(handle, JOB_GTL, 3, 0)
```

```
If (Not result) Then
  MsgBox "Command GTL failed!", 16, "Information"
End If

QAPIExtCloseCard (handle)

End Sub

Private Sub Form_Load()
  Call Main
End Sub
```

5.5 GPIB National Instruments LabView Example

After the installation of the QLIB, you find the LabView VI's in the folder *d:\program files\quancom\qlib32\samples\gplib\labview* (only if you install at the standard folder)

The file GPIB.VI is a example program for National Instruments LabView. You can use this program to test the functions of the QUANCOM GPIB under Labview. The example Das Beispiel loads the files *gplib_receive.vi* and *gplib_send.vi*, which call the QLIB functions.

The following VI's are available at the moment :

<i>gplib_receive.vi</i>	Read a String of a device at the bus
<i>gplib_send.vi</i>	Send a String to a device at the bus

The following VI's are available soon :

<i>gplib_spoll.vi</i>	Serialpoll the devices at the bus
<i>gplib_checksrq.vi</i>	Read the SRQ-Status at the bus
<i>gplib_command.vi</i>	Send a command at the bus

Please check our homepage for Updates. Updates for the QLIB are available under <http://www.quancom.de/qlib> .

To use the GPIB Controllercard with LabView, you must du the following steps :

Step 0: Install the QLIB (QUANCOM Driver Library)

Step 1: Add the makros "*gplib_receive.vi*" and "*gplib_send.vi*" to your project.

5.6 GPIB Lotus Notes Example

The fat-printed text contains Script Examples for the integration of the QLIB in Lotus Notes. The following example represents only the scripts, which also refer of fields in masks. For details, please check the example data base. After the installation of the QLIB, you find this example data base at the folder *d:\program files\quancom\qlib32\samples\gplib\lotus* (only if you install at the standard folder)

The steps to add the QLIB to your own projects are at the header of the sourcetext.

```
' gplib.bas : Sample project for the GPIB-1, PCIGPIB and USB GPIB Modules
'
' Author: Michael Reimer, QUANCOM Informationssysteme GmbH, Germany
'
' Website: http://www.quancom.de
'
' Information:
'
' To use the QLIB Commands in your source, do the following:
' (Step 0: Install the QLIB ( QUANCOM Driver Library )
Step 1: Copy Script Library "qlib" to your database
Step 2: Add Statement Use "qlib" to section Globals->Options
'
'
' Change the address to the appropriate address for your
' device ( normally set by DIP-Switches on the back side )
'
'-----
' PART 1: Writing a string to a GPIB / HPIB / IEEE488 DMM ( The DMM
' is a listener ) from Lotus Notes.
' Listener: A device capable of receiving data over the interface
' when addressed to Listen by the active controller. Examples of
' such devices are printers, programmable power supplies, or any
' other programmable instrument. There can be up to 14 Listeners on
' the GPIB Bus at one time.
'-----

Sub SendString()

    Dim ws As New NotesUIWorkspace
    Dim uidoc As NotesUiDocument

    Dim nDevice As Long
    Dim nListener As Long
    Dim s As String
    Dim handle As Long
    Dim result As Long
    Dim nCardID As Integer
    Dim CardName As String
    Dim Listener As String

    Set uidoc = ws.CurrentDocument

    CardName = uidoc.FieldGetText("SelectedCard")
```

```

    If ( CardName = "" ) Then
        MsgBox "Please select the GPIB Card!" , 16, "Error"
        Exit Sub
    End If

' QLIB supports up to 8 GPIB Controller

    nDevice = Val(Right$(CardName, 1))

    If (Instr(CardName, "PCIGPIB")) Then
        nCardID = PCIGPIB
    ElseIf (Instr(CardName, "USBGPIB")) Then
        nCardID = USBGPIB
    Else
        nCardID = GPIB
    End If

' Retrieve Listener Address

    Listener = uidoc.FieldGetText("Listener")

    nListener = Val(Right$(Listener, 2))

' Retrieve Text to send to the GPIB Device

    s = uidoc.FieldGetText("StringToSend")

    handle = QAPIExtOpenCard(nCardID, nDevice)

    If (handle <> 0) Then

        result = QAPIExtWriteString(handle, nListener, s, Len(s), 0)

        If (result) Then
' data send to gpib device
        Else
' failure
            MsgBox "Failure sending data!", 16, "Information"
        End If

    End If

End Sub

' -----
' PART 2: Reading a GPIB / IEEE-488 / HPIB Device from Lotus Notes
'
' Talker: A device capable of transmitting data over the interface
' when addressed to talk by the active controller. Examples of such
' devices are voltmeters, data-acquisition systems, or any other
' programmable instrument. There can be only one addressed talker on
' the GPIB Bus at one time.
' -----
'
' Select the talker address, which is set by a DIP-Switch on the
' back side of your instrument.
'
Sub ReadString()

```

```
Dim ws As New NotesUIWorkspace
Dim uidoc As NotesUiDocument

Dim nDevice As Long
Dim nListener As Long
Dim s As String
Dim handle As Long
Dim result As Long
Dim nCardID As Integer

Dim CardName As String
Dim Listener As String

Set uidoc = ws.CurrentDocument

CardName = uidoc.FieldGetText("SelectedCard")

If ( CardName = "" ) Then
    MsgBox "Please select the GPIB Card!" ,16,"Error"
    Exit Sub
End If

' QLIB supports up to 8 GPIB Controller

nDevice = Val(Right$(CardName, 1))

If (Instr(CardName, "PCIGPIB")) Then
    nCardID = PCIGPIB
Elseif (Instr(CardName, "USBGPIB")) Then
    nCardID = USBGPIB
Else
    nCardID = GPIB
End If

' Retrieve Listener Address

Listener = uidoc.FieldGetText("Talker")

nListener = Val(Right$(Listener, 2))

' Create string buffer

s = Space$(256)

handle = QAPIExtOpenCard(nCardID, nDevice)

If (handle <> 0) Then

    result = QAPIExtReadString(handle, nListener, s, Len(s), 0)

    If (result) Then
        ' data successfully read from gpib device

        Call uidoc.FieldSetText("String", s)

    Else
        ' failure

        MsgBox "Failure reading data!", vbInformation, "Information"
    End If
End If
```

```

        End If

End Sub

'-----
' PART3: Checking whether a device has requested service ( SRQ
' Service Request)
'
' A device can interrupt the active controller by asserting the SRQ
' line. The SRQ is a single line, and if there are multiple devices
' on the GPIB Bus that have been configured to assert an SRQ, the
' active controller will have to "poll" the devices to figure out
' which one actually asserted the SRQ. More than one device could in
' principle assert an SRQ at the same time. The active controller
' can poll the devices in one of two ways: serial poll or parallel
' poll.
'-----

Sub ReadSRQ()

    Dim ws As New NotesUIWorkspace
    Dim uidoc As NotesUiDocument

    Dim nDevice As Long
    Dim nListener As Long
    Dim s As String
    Dim handle As Long
    Dim result As Long
    Dim nCmd As Long
    Dim nCardID As Integer
    Dim CardName As String
    Dim Listener As String

    Set uidoc = ws.CurrentDocument

    CardName = uidoc.FieldGetText("SelectedCard")

    If ( CardName = "" ) Then
        MsgBox "Please select the GPIB Card!" , 16, "Error"
        Exit Sub
    End If

' QLIB supports up to 8 GPIB Controller

    nDevice = Val(Right$(CardName, 1))

    If (Instr(CardName, "PCIGPIB")) Then
        nCardID = PCIGPIB
    ElseIf (Instr(CardName, "USBGPIB")) Then
        nCardID = USBGPIB
    Else
        nCardID = GPIB
    End If

    handle = QAPIExtOpenCard(nCardID, nDevice)

    If (handle <> 0) Then

        result = QAPIExtSpecial(handle, JOB_READSRQ, 0, 0)

```

```
        If (result = 0) Then
            MsgBox "No service requested", 16, "Info"
        Else
            MsgBox "Device has requested service ( SRQ )", 16, "Info"
        End If

    End If

End Sub

'-----
' PART4: Reading Serial Poll Status from DMM
'
' In a serial poll, the active controller asks each device in turn
' to send back a status byte that indicates whether the device has
' asserted the SRQ. Bit 6 of this byte (where the bits are numbered
' 0 through 7) is set if the device is requesting service. The
' definition of the other bits is device dependent (under 488.1 at
' least; 488.2 provides a much more concise definition of the status
' byte).
'
' The program has to perform this same sequence with every device
' it needs to poll.
'-----
Sub SerialPoll()
    Dim ws As New NotesUIWorkspace
    Dim uidoc As NotesUIDocument

    Dim nDevice As Long
    Dim nListener As Long
    Dim s As String
    Dim handle As Long
    Dim result As Long
    Dim pollbyte As Long
    Dim nCardID As Integer
    Dim CardName As String
    Dim Listener As String

    Set uidoc = ws.CurrentDocument

    CardName = uidoc.FieldGetText("SelectedCard")

    If ( CardName = "" ) Then
        MsgBox "Please select the GPIB Card!" , 16, "Error"
        Exit Sub
    End If

' QLIB supports up to 8 GPIB Controller

    nDevice = Val(Right$(CardName, 1))

    If (Instr(CardName, "PCIGPIB")) Then
        nCardID = PCIGPIB
    ElseIf (Instr(CardName, "USBGPIB")) Then
        nCardID = USBGPIB
    Else
        nCardID = GPIB
    End If

' Retrieve Listener Address
```

```

    Listener = uidoc.FieldGetText("Listener2")
    nListener = Val(Right$(Listener, 2))
' Create string buffer
    s = Space$(256)
    handle = QAPIExtOpenCard(nCardID, nDevice)
    If (handle <> 0) Then
        result = QAPIExtSpecialSP(handle, JOB_SERIALPOLL, nListener, pollbyte)
        If (result) Then
' data successfully read from gpib device
            Call uidoc.FieldSetText("String2", "Hex: " & Hex(Val(pollbyte)) & " Dec: " &
(Val(pollbyte)))
        Else
' failure
            MsgBox "Failure reading data!", vbInformation, "Information"
        End If
    End If
End Sub

```

```

'-----
' PART5: Send a command to DMM
'
' The following commands are accepted by all devices on the GPIB Bus
' simultaneously. The address part will be ignored.
'
' - JOB_DCL (Device Clear): The DCL command causes all devices to
' return to a device dependent initial state.
'
' - JOB_LLO (Local Lockout): The LLO command disables the return-to-
' local front panel key on the device. The user can no longer
' change the device settings from its front panel.
'
' The following commands need an address and are only accepted by
' addressed devices. Whether the devices are the listeners or the
' talkers depends on the command. The three commands are as follows:
'
' - JOB_GET (Group Execute Trigger): The GET command tells all the
' addressed listeners to perform some device-dependent function,
' like take a measurement. GET allows for synchronizing a
' measurement function between multiple devices. This is only used
' in specialized cases.
'
' - JOB_SDC (Selected Device Clear): The SDC command resets the
' addressed listeners to a device-dependent state. It performs the
' same function as DCL, but only resets the addressed listeners,
' not all the devices.
'
' - JOB_GTL (Go To Local): The GTL command sets the addressed
' listeners back to local mode.
'-----

```

```
Sub SendCommand()

    Dim ws As New NotesUIWorkspace
    Dim uidoc As NotesUiDocument

    Dim nDevice As Long
    Dim nListener As Long
    Dim s As String
    Dim handle As Long
    Dim result As Long
    Dim nCmd As Long
    Dim nCardID As Integer
    Dim CardName As String
    Dim Listener As String
    Dim CommandName As String

    Set uidoc = ws.CurrentDocument

    CardName = uidoc.FieldGetText("SelectedCard")
    CommandName = uidoc.FieldGetText("CommandList")

    If ( CardName = "" ) Then
        MsgBox "Please select the GPIB Card!" , 16, "Error"
        Exit Sub
    End If

' QLIB supports up to 8 GPIB Controller

    nDevice = Val(Right$(CardName, 1))

    If (Instr(CardName, "PCIGPIB")) Then
        nCardID = PCIGPIB
    ElseIf (Instr(CardName, "USBGPIB")) Then
        nCardID = USBGPIB
    Else
        nCardID = GPIB
    End If

' Retrieve Listener Address

    Listener = uidoc.FieldGetText("Listener3")

    nListener = Val(Right$(Listener, 2))

    handle = QAPIExtOpenCard(nCardID, nDevice)

    If (handle <> 0) Then

        Select Case Left$( CommandName,3)
            Case "GTL": nCmd = JOB_GTL
            Case "SDC": nCmd = JOB_SDC
            Case "GET": nCmd = JOB_GET
            Case "LLO": nCmd = JOB_LLO
            Case "DCL": nCmd = JOB_DCL
        End Select

        result = QAPIExtSpecial(handle, nCmd, nListener, 0)

        If (result) Then
            ' cmd successfully send to gpib device
        End If
    End If
End Sub
```

```
    Else
' failure
        MsgBox "Failure sending command!", vbInformation, "Information"
    End If
End If
End Sub
```

5.7 GPIB Borland Delphi Example

The fat-printed text describes, how you can include the QLIB in a Delphi project. After the installation of the QLIB, you find the immediately compilable Example project at the folder *d:\program files\quancom\qlib32\samples\gplib\delphi* (only if you install at the standard folder).

The steps to add the QLIB to your own projects are at the header of the sourcetext.

```
program GPIB_CONSOLE_APP;

{*
// gplib.pas : Sample project for Borland Delphi shows how to
// program the GPIB-1, PCIGPIB and USB GPIB Modules
//
// Author: Michael Reimer, QUANCOM Informationssysteme GmbH, Germany
//
// Website: http://www.quancom.de
// Product:
// GPIB PCI Controller http://www.quancom.de/qprod01/deu/pb/pcigpib_1.htm
// GPIB ISA Controller http://www.quancom.de/qprod01/deu/pb/GPIB_1.htm
// GPIB USB Controller http://www.quancom.de/qprod01/deu/pb/usb_gpib_1.htm
// Information:
//
// To use the QLIB Commands in your source, do the following:
//
// (1) Add statement #include "qlib.pas" to your source file.
// (2) Copy QLIB.PAS from QLIB Installation Directory
//     d:\program files\quancom\qlib32\include to your
//     working directory
*}

{$APPTYPE CONSOLE}

{$INCLUDE QLIB.pas}

{$X+}

var handle: longint;
    result: longint;
    listener: longint;
    talker: longint;
    buffer: string;
    s: string;
    serial_poll_byte: longint;
    device: longint;

begin

{*
// The following sequence tries to find the
// Bus Type ( PCI, ISA or USB ) of the
// installed GPIB Controller
*}

serial_poll_byte := 3;
```

```

result := LongInt(@serial_poll_byte);

result := result + 1;

writeln(result);

handle := QAPIExtOpenCard(PCIGPIB,0);
if ( handle = 0 ) then
  begin
    handle := QAPIExtOpenCard(USBGPIB,0);
    if ( handle = 0 ) then
      begin
        handle := QAPIExtOpenCard(GPIB,0);
      end;
    end;

  {*
// The handle is <> NULL if there is a GPIB Controller installed
*}

if ( handle = 0 ) then
  begin
    s := 'No QUANCOM GPIB Controller found!';
    writeln(s);
    halt(0);
  end;

  {*
// Ok, we found a QUANCOM GPIB Controller Card

// Now we can send a string to the listner with address 3
// Change the address to the appropriate address for your
// device ( normally set by DIP-Switches on the back side )

// -----
// PART 1a: Writing a string to a IEEE 488, GPIB, HPIB Device
//
// Listener: A device capable of receiving data over the interface
// when addressed to Listen by the active controller. Examples of such
// devices are printers, programmable power supplies, or any other
// programmable instrument. There can be up to 14 Listeners on the GPIB Bus
// at one time.
// -----

// Select the listener address, which is set by a DIP-Switch on the
// back side of your instrument.
*}

listener := 3;

// Send "z" to the listener 3, which resets the DMM to the initial settings.

s := 'z'#0;

if ( QAPIExtWriteString(handle, listener, Pchar(s), Length(s),0) = 0) then
  begin
    writeln('Writing to device ', listener, ' was successful. ');
  end
else
  begin

```

```
writeln('Writing to device ', listener, ' failed.');
```

```
end;
```

```
{*  
// -----  
// PART 1b: Reading a DMM or any other IEEE-488 device  
//  
// Talker: A device capable of transmitting data over the interface when  
// addressed to talk by the active controller. Examples of such devices  
// are voltmeters, data-acquisition systems, or any other programmable  
// instrument. There can be only one addressed talker on the GPIB Bus at one  
// time.  
// -----  
  
// Select the talker address, which is set by a DIP-Switch on the  
// back side of your instrument.  
*}
```

```
talker := 3;  
SetLength(buffer,1024);
```

```
{*  
// Read value from DMM with talker address 3  
*}
```

```
result := QAPIExtReadString(handle, talker, PChar(buffer), Length(buffer), 0);
```

```
if ( result <> 0 ) then  
begin  
writeln('Reading from device', talker, ' was ', buffer);  
end  
else  
begin  
writeln('Reading from device ', talker, ' failed.');
```

```
end;
```

```
{*  
// -----  
// PART2: Checking whether a device has requested service ( SRQ Service Request)  
//  
// A device can interrupt the active controller by asserting the SRQ line. The  
// SRQ is a single line, and if there are multiple devices on the GPIB Bus that  
// have been configured to assert an SRQ, the active controller will have  
// to "poll" the devices to figure out which one actually asserted the SRQ.  
// More than one device could in principle assert an SRQ at the same time. The  
// active controller can poll the devices in one of two ways: serial poll  
// or parallel poll.  
// -----  
*}
```

```
result := QAPIExtSpecial(handle, JOB_READSRQ, 0, 0);
```

```
if (result <> 0) then  
begin  
writeln('SRQ was asserted.');
```

```
end  
else  
begin  
writeln('SRQ was not asserted.');
```

```
end;
```

```

{*
// -----
// PART3: Reading Serial Poll Status from DMM
//
// In a serial poll, the active controller asks each device in turn to
// send back a status byte that indicates whether the device has asserted
// the SRQ. Bit 6 of this byte (where the bits are numbered 0 through 7) is set
// if the device is requesting service. The definition of the other bits
// is device dependent (under 488.1 at least; 488.2 provides a much more
// concise definition of the status byte).
//
// The program has to perform this same sequence with every device
// it needs to poll.
// -----

// We poll all devices from 1 to 15 here. This is normally not necessary and
// very time consuming. Poll only valid devices on the GPIB Bus.
*}

serial_poll_byte := 0;

for device := 1 to 15 do
  begin
    result := QAPIExtSpecial(handle, JOB_SERIALPOLL, device, LongInt(@serial_poll_byte));

    if ( result <> 0 ) then
      begin
        if ((serial_poll_byte and $40) = 0) then
          begin
            {* check for bit 6 = device requested service *}
            writeln('Device ', device, ' requested service and returned status byte ',
              serial_poll_byte);
          end
        else
          begin
            writeln('Device ', device, ' no SRQ requested. Status is ', serial_poll_byte);
          end;
        end
      else
        begin
          writeln('No answer from device ', device);
        end;
      end;
  end;

{*
// -----
// PART4: Send a command to DMM
//
// The following commands are accepted by all devices on the GPIB Bus
// simultaneously. The address part will be ignored.
//
// - JOB_DCL (Device Clear): The DCL command causes all devices to return to a device
// dependent initial state.
//
// - JOB_LLO (Local Lockout): The LLO command disables the return-to-local front
// panel key on the device. The user can no longer change the device settings
// from its front panel.
//
// The following commands need an address and are only accepted by addressed
// devices. Whether the devices are the listeners or the talkers depends on the

```

```
// command. The three commands are as follows:
//
// - JOB_GET (Group Execute Trigger): The GET command tells all the addressed
// listeners to perform some device-dependent function, like take a measurement.
// GET allows for synchronizing a measurement function between multiple devices.
// This is only used in specialized cases.
//
// - JOB_SDC (Selected Device Clear): The SDC command resets the addressed listeners
// to a device-dependent state. It performs the same function as DCL,
// but only resets the addressed listeners, not all the devices.
//
// - JOB_GTL (Go To Local): The GTL command sets the addressed listeners back to
// local mode.
// -----

// Send DCL to all devices
*}

result := QAPIExtSpecial(handle, JOB_DCL, 1, NULL);

if ( result = 0 ) then
  begin
    writeln('Command DCL failed\n');
  end;

{* Send LLO to all devices *}

result := QAPIExtSpecial(handle, JOB_LLO, 1, NULL);

if ( result = 0 ) then
  begin
    writeln('Command LLO failed\n');
  end;

{* Send GET to device 3 *}

result := QAPIExtSpecial(handle, JOB_GET, 3, NULL);

if ( result = 0 ) then
  begin
    writeln('Command GET failed\n');
  end;

{* Send SDC to device 3 *}

result := QAPIExtSpecial(handle, JOB_SDC, 3, NULL);

if ( result = 0 ) then
  begin
    writeln('Command SDC failed\n');
  end;

{* Send GTL to device 3 *}

result := QAPIExtSpecial(handle, JOB_GTL, 3, NULL);

if ( result = 0 ) then
  begin
```

```
writeln('Command GTL failed\n');  
end;  
  
QAPIExtCloseCard(handle);  
  
end.
```


6 The Device Driver (MS-Dos) for the GPIB-1

6.1 Installation

If you are using MS-DOS or Windows 95 you have to install the device driver "GPIB_DRV.SYS". This allows you to communicate to the GPIB devices through a file interface. To use this option, you must install the driver on your computer:

1. Insert the enclosed program disk in disc drive A:.
2. Enter at the MS-DOS prompt:

```
COPY A:\DEVDRV\GPIB_DRV.SYS C:\
```

If you should have problems, then please look up in the MS DOS user manual.

3. The driver must be entered in the CONFIG.SYS now, to be loaded at system startup. You can add a line of text to the config.sys file with the MS-DOS editor.
4. Type the following command at the operating system prompt:

```
EDIT C:\CONFIG.SYS.
```

5. Move with the arrow buttons the cursor to the end of the file and insert:

```
DEVICE=C:\GPIB_DRV.SYS
```

Please see chapter 6.1.1 for additional arguments. I.e. if you have selected a non default I/O Address you may write a modified line as follows:

```
DEVICE=C:\GPIB_DRV.SYS /p200
```

(setup for a GPIB board at I/O Address 200 hex)

6. Save the file with the and exit the editor.
7. To be able to use the driver, you must restart your computer now.

6.1.1 Command parameter

The following parameters are optional commands to initialize the device driver:

/pxxx Configures the port address xxx (200..FFF) in the hexadecimal format.
(Standard: 330 HEX)

/yxx Select the command termination character with the number xx (00..FF) (default is the semicolon “;”). The command sequence is ended by sending a semicolon.

Example: After opening the file GPIB for the output, you may send the command “;/d0D;” this sets the current listener to address 0D (command /dxx see below)

/tx The timeout period in seconds (0..9). Default timeout is 1 second. This parameter determines how long the device waits for data input from the devices on the bus. After this time the controller terminates current transfer.

/ux Enables or disables the possibility to terminate the data transfer by the user.
(0: No termination allowed , 1: Enable user break, **Default :1**)

Note: Terminating the data transfer by pressing a key always results in a „general error“ from the GPIB driver !

/ex The GPIB device driver searches the data for the termination character. When it receives the termination character it assumes the previously received data as complete command string and initiates the data transfer.

1: CR+EOI

2: CR (Default)

3: LF+EOI

4: LF

5: CR+LF+EOI

6: CR+LF

7: LF+CR

8: EOI (only received)

/sxx Sets the talkeraddress to xx (1..1E).

/dxx Sets the listeneraddress to xx (1..1E).

6.1.2 Examples

DEVICE = C:\GPIB_DRV.SYS p330

The GPIB driver is loaded and expects the GPIB board to be configured with JP1 at the I/O Address 330 hex.

Settings:

The port address is 330 hex,

The command character is the semicolon,

The timeout period is set to one second,

User break not allowed

The string termination character is CR.

DEVICE = C:\GPIB_DRV.SYS /p300 /e4 /t2 /u0 /y07

The port address is 300 hex,

The GPIB driver expects every command terminated by a "LF" (Line feed charcter = 10),

The timeout period is set to two seconds.

No user keyboard break allowed .

Command delimiter is set to 07 hex (bell charcter).

6.2 Programming examples in BASIC:

Also during the enterprise can you change the parameters of the board.

For this serves the above described command symbol. its pre-setting is the semicolon. To change the parameters, you must directly after opening transmitted the command symbol. The following parameters, which you must changed, afterwards an ; (semicolon) and then the data which can be transmitted.

```
10 OPEN "GPIB" FOR OUTPUT AS #1
20 PRINT #1,";/d0e; T0V1T1"
30 CLOSE #1
```

Line 10 assigns the GPIB Controllerboard to the outputstream 1. The board can be addressed now like a file. The output instruction in line 20 addressed now the equipment with the address 0e as Listener and gives T0V1T1 as data out.

The instruction PRINT spends following the data automatically still another CR, which releases the transmission of the data as preset end identifier. Line 30 closed the outputstream 1. also reading in of data is possible over the DEVICE driver

Also reading in of data is possible over the DEVICE driver:

```
10 OPEN "GPIB" FOR INPUT AS #1
20 INPUT #1,A$
30 PRINT A$
40 CLOSE #1
```

Line 10 assigns the GPIB-Controllerboard to the inputstream 1.

Line 20 reads the data in and spent in line 30 on the screen.

In line 40 the inputstream 1 is closed.

6.3 Programming examples in C:

```
/* Send with the GPIB */

#include <stdio.h>
#include <fcntl.h>

void main(void)
{
    int handle;
    char string[20];
    unsigned length;

    handle=open("GPIB",O_WRONLY);
    strcpy(string, "/d0e;T0V1T1");
    length=strlen(string);
    write(handle,string,length);
    close(handle);
}
```

```
/* Received with the GPIB */

#include <stdio.h>
#include <fcntl.h>

void main(void)
{
    int handle;
    char string[255];

    handle=open("GPIB",O_RDONLY);
    read(handle,string,255);
    printf("%s",string);

    close(handle);
}
```

6.4 Programming examples in PASCAL:

```
program Send_with_GPIB;

uses crt;

var handle : text;

begin
  assign(handle,'GPIB');
  reset(handle);
  writeln(handle,;/d0e;TOV1T1');
  close(handle);
end.
```

```
program Received_with_GPIB;

uses crt;

var handle: text;
    eingabe  : string;

begin
  assign(handle,'GPIB');
  reset(handle);
  readln(handle,eingabe);
  writeln(eingabe);
  close(handle);
end.
```


7 Annex



7.1 Customer Communication and Help

You need help?

If you don't know how to go on during the installation or operation of your QUANCOM board please first read this user's guide.

! Tip !

On the QUANCOM installation CD you can find a ASCII – Text – file README.TXT, which includes changes made after printing this user's manual.

! IMPORTANT !

If you have further questions please contact our support team and have the following information handy:

- Exact type of board
- Operating system, hardware equipment and Bus - System
- Name and Version of the program, where the error is reported.
- A detailed failure description. To make sure, please try to reproduce the failure, describe it as exact as possible, and which steps led to this failure.

Who can you contact?

The QUANCOM internet website

WWW.QUANCOM.DE

Per Fax

+49 22 36 / 89 92 - 49

Per e-mail:

support@quancom.de

Address:

**QUANCOM INFORMATIONSSYSTEME
GmbH**

In der Flecht 14

D-50389 Wesseling

Germany

If you need urgent help call:

QUANCOM Hotline Germany

+49 22 36 / 89 92 - 20

Monday-Thursday

from 9:00 to 18:00

Friday

from 9:00 to 17:00

Actual drivers

You can find the newest Version of QUANCOM software on our internet website <http://www.quancom.de>. You can also find a lot of information and „Frequently asked questions (FAQ's)”. Before you contact the QUANCOM support, please check if you are using the newest software version of the QUANCOM software.

Repair

If you are not sure whether your QUANCOM board is defective please call the QUANCOM Hotline:

Tel.: **+49 22 36 / 89 92 – 20**

Before you send us the QUANCOM board to be repaired call:

Tel.: **+49 22 36 / 89 92 – 20**

If you send us your QUANCOM board, please use original package or any other suitable package, to protect the contents against transport damage. You also need to send us a copy of the original bill and the RMA number.

You can shorten the repair time by sending us an exact failure description, so that a faster failure search is possible. Send your QUANCOM board directly to the service department of QUANCOM Informations-systeme GmbH.

7.2 Technical support form

If you have internet access please enter the following URL in your browser: <http://www.quancom.de/quancom/qshop.nsf/techniksupport?OpenForm&eng> else photocopy this form and use the copy of this form as a reference for your current configuration. Complete this form before contacting QUANCOM Informationssysteme GmbH for technical support and our applications engineers may answer your questions more efficiently. If you are using any other QUANCOM hardware or software products please add them to this configuration form. Include additional pages if necessary.

Name: _____

Company: _____

Address: _____

Phone: _____

Fax: _____

Computer brand / Processor: _____

Operating system: _____

Display adapter: _____

Mouse: _____

QUANCOM board _____

Other adapters installed: _____

Hard disk (capacity, free): _____

The problem is: _____

List any error messages: _____

The following steps cause the problem to recur:

7.3 Hardware and Software configuration form

This form helps you to list your hardware and software settings. Complete this form each time you change your software or hardware configuration, and use this form as a reference for your current configuration. Complete this form accurately before contacting QUANCOM Informationssysteme GmbH for technical support, so that our applications engineers can answer your questions more efficiently.

• **QUANCOM Product:**

Name / Name of board _____

Interrupt Level _____

DMA Channel _____

Base I/O Address _____

Operating system _____

• **Other information**

Computer brand and Model _____

Processor _____

Clock Frequency _____

Type of Video Board Installed _____

DOS Version _____

Programming Language _____

Programming Language Version _____

• **Other Boards in System**

Base I/O Address of other Boards _____

DMA Channels of other Boards _____

Interrupt Level of other Boards _____

7.4 Documentation Comment Form

QUANCOM Informationssysteme GmbH would like you to comment on the documentation supplied with our products. This information helps us to provide you with quality products to meet your needs. Include additional pages if necessary.

Title: **PCIGPIB1 / GPIB-1**

Edition Date: **18.07.03**

Please comment on the completeness, clarity, and organisation of the manual. If you find errors in the manual, please record the page numbers and describe the errors. Thank you for your help.

Name: _____

Company: _____

Address: _____

Phone: _____

Fax: _____

Comment: _____

Mail to: support@quancom.de

Fax to: +49 2236 89 92 49

Address: **QUANCOM Informationssysteme GmbH
In der Flecht 14,
50389 Wesseling
Germany**

registered trade-mark:

Linux is registered trade-mark of Linus Torvalds.

MS, MS-DOS, Microsoft, Visual Basic, Windows, Windows XP/2000/NT/ME/98/95 is registered trade-mark of von Microsoft Corporation.

XT and PS/2 are trade-marks and IBM, OS/2 and AT are registered trade-mark of International Business Machines Corporation.

Intel, Pentium is registered trade-mark of Intel Corporation.

USB is registered trade-mark of USB Implementers Forum Inc.

JAVA is registered trade-mark of Sun Microsystems.

DELPHI and Pascal are registered trade-mark of Borland Corporation.

PCI ist eingetragenes Warenzeichen von PCI Special Interest Group.

Nationalinstruments, LABVIEW is registered trade-mark of Nationalinstruments Corporation.

Agilent VEE is registered trade-mark of Agilent Technologies.

By other product- and company names, that are mentioned in this manual, it may deal with trademarks of the respective owners.