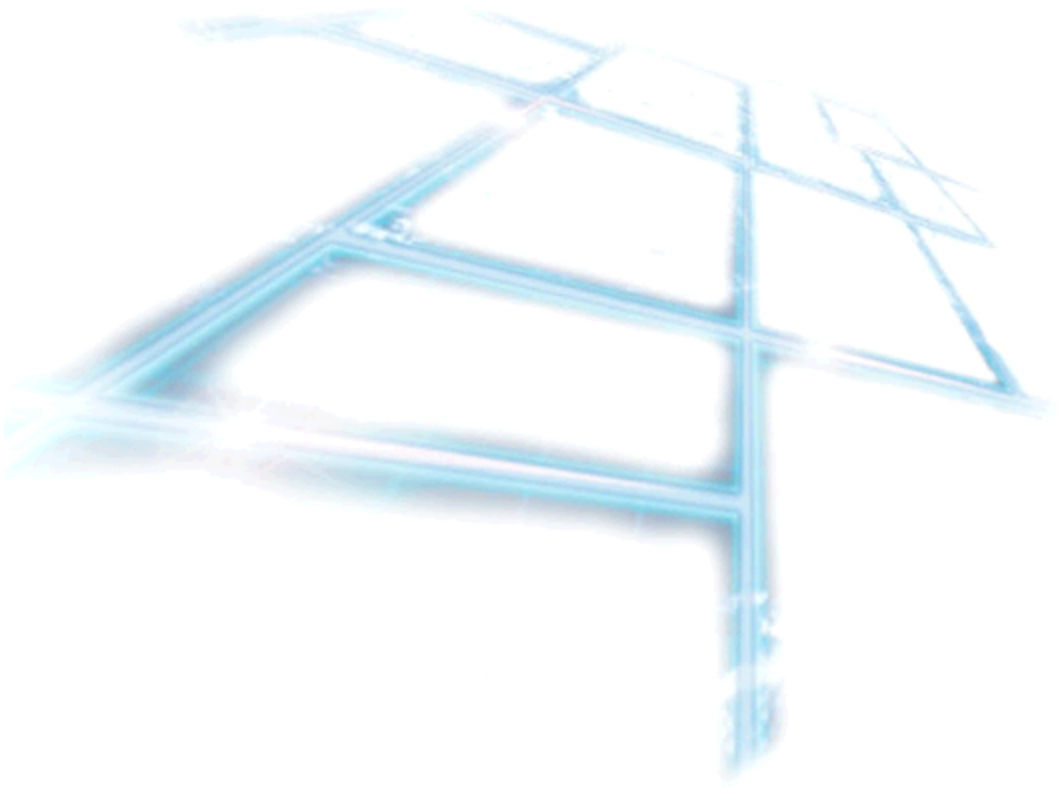
QUANCOM
INFORMATIONSSYSTEME

Serial / Ethernet Protocol



Wesseling, Oktober 2010 Version 4.1.0

Inhaltsverzeichnis

1 Overview

1.1 Introduction

We congratulate you on buying the QUANCOM high quality measurement and automation board. You have chosen a product which attributes and functions show the latest updates of technology.
The following special attributes are included:

Features of the board:

- einfach programmierbar
- diverse Beispielprogramme in verschiedenen Programmiersprachen
- Treiberunterstützung unter Windows XP/2000/NT4 und ME/98/95 mit der **QLIB (QUANCOM Driver Library)**

1.2 Our experience is your profit

QUANCOM is specialised in development of hard- and software. QUANCOM has become one of the leading suppliers of measuring and automation technology in 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 you if you have problems with them. For easy contacting, this manual contains comment and configuration forms for you to complete, which are in chapter ” **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 from our internet website. **www.quancom.de**

2 Register Configuration

In this chapter you will find the tables which contain all registers and their given values, with whom you will be able to program the devices hardware sided.

Please note that the bit width is relative to the used register.

This means:

- **8 Bit Access**

  Only one 8 bit register at a time will be read or written

  Example: Addresses 0x100, 0x101, 0x102, ..

- **16 Bit Access**

  Two 8 bit regsiters at a time will be read or written

  Example: To write the value 0x1234 to register 0x100 it will be spilt as follows
  On address 0x100 0x34 will be written
  On address 0x101 0x12 will be written

- **32 Bit Zugriff**

  Four 8 bit regsiters at a time will be read or written
  Example: To write the value 0x12345678 to register 0x100 it will be spilt as follows
  On address 0x100 0x78 will be written
  On address 0x101 0x56 will be written
  On address 0x102 0x34 will be written
  On address 0x103 0x12 will be written

- **64 Bit Zugriff**

  Eight 8 bit regsiters at a time will be read or written

On the following pages all register configurations will be defined in table form to assure best readability.

2.1 Outputs

Register configuration for writing the outputs

Offset	R / W	Bit	Description
0x100	Write	0-7	Writing the outputs 1 - 8
0x101	Write	0-7	Writing the outputs 9 - 16
0x102	Write	0-7	Writing the outputs 17 - 24
0x103	Write	0-7	Writing the outputs 25 - 32
0x104	Write	0-7	Writing the outputs 33 - 40
0x105	Write	0-7	Writing the outputs 41 - 48
0x106	Write	0-7	Writing the outputs 49 - 56
0x107	Write	0-7	Writing the outputs 57 - 64

Register configuration for read back the outputs

Offset	R / W	Bit	Description
0x100	Read	0-7	read back the outputs 1 - 8
0x101	Read	0-7	read back the outputs 9 - 16
0x102	Read	0-7	read back the outputs 17 - 24
0x103	Read	0-7	read back the outputs 25 - 32
0x104	Read	0-7	read back the outputs 33 - 40
0x105	Read	0-7	read back the outputs 41 - 48
0x106	Read	0-7	read back the outputs 49 - 56
0x107	Read	0-7	read back the outputs 57 - 64

2.2 Inputs

Read the Inputs

Offset	R / W	Bit	Description
0x120	Read	0-7	Read the Inputs 1 - 8
0x121	Read	0-7	Read the Inputs 9 - 16
0x122	Read	0-7	Read the Inputs 17 - 24
0x123	Read	0-7	Read the Inputs 25 - 32
0x124	Read	0-7	Read the Inputs 33 - 40
0x125	Read	0-7	Read the Inputs 41 - 48
0x126	Read	0-7	Read the Inputs 49 - 56
0x127	Read	0-7	Read the Inputs 57 - 64

2.3 Input-Flip-Flops

Read the Flip-Flops

Offset	R / W	Bit	Description
0x140	Read	0-7	Read the FlipFlops 1 - 8
0x141	Read	0-7	Read the FlipFlops 9 - 16
0x142	Read	0-7	Read the FlipFlops 17 - 24
0x143	Read	0-7	Read the FlipFlops 25 - 32
0x144	Read	0-7	Read the FlipFlops 33 - 40
0x145	Read	0-7	Read the FlipFlops 41 - 48
0x146	Read	0-7	Read the FlipFlops 49 - 56
0x147	Read	0-7	Read the FlipFlops 57 - 64

2.4 Counter

Counter 8 Bit Register

Offset	R / W	Bit	Value (0..255)	Description
0xF8	Write	0	1	Reset Counter 0 (1 = active)
0xF8	Write	1	2	Reset Counter 1 (1 = active)
0xF9	Write	0	1	0 = Counter 0 without HW-Gate 1 = Counter 0 with HW-Gate (only count if external Gate=1)

Counter read (only 32 Bit Requests)

Offset	R / W	Bit	Description
0xF0	Read	0-31	Counter 0 read
0xF4	Read	0-31	Counter 1 read
0xF8	Read	0-31	Counter 0 read + Reset
0xFC	Read	0-31	Counter 1 read + Reset

Example:

To reset the Counter1 the following Commands are needed:

write 0xF8,2 // Reset Counter 1 = active

write 0xF8,0 // Reset Counter 1 = inactive

2.5 Dip switch

You can read The Dip switches with help of the **status registers**.

2.6 Control register

Control Register

Offset	R / W	Bit	Value (0..255)	Description
0x07	Write	0	1	Reset (1 = active)
0x07	Write	1	2	IRQ-0 enable Timeout is active (1 = active)
0x07	Write	2	4	IRQ-1 enable Input change is active (1 = active)
0x07	Write	3,4,5	8,16,32	Timeout Selection
0x07	Write	6	64	TOUT_EN enable Timeout for the outputs (1 = active)
0x07	-	7	-	unused

Timeout Selection

Worth	Bit 5,4,3	Description
0	000	Timeout Worth = 0,35s
1	001	Timeout Worth = 0,7s
2	010	Timeout Worth = 1,398s
3	011	Timeout Worth = 2,796s
4	100	Timeout Worth = 5,59s
5	101	Timeout Worth = 11,18s
6	110	Timeout Worth = 22,36s
7	111	Timeout Worth = 44,73s

2.7 Status register

Status register

Offset	R / W	Bit	Value (0..255)	Description
0x07	Read	0	1	IRQ-0 enable (read back the Control register Bit 1)
0x07	Read	1	2	IRQ-1 enable (read back the Control register Bit 2)
0x07	-	2	-	unused
0x07	-	3	-	unused
0x07	Read	4	16	read Dip switch 1 (0 = On / 1 = Off)
0x07	Read	5	32	read Dip switch 2 (0 = On / 1 = Off)
0x07	Read	6	64	read Dip switch 3 (0 = On / 1 = Off)
0x07	Read	7	128	read Dip switch 4 (0 = On / 1 = Off)

3 Protocoll

This chapter shows how communication through our USBADPETH or USBADPRS232 works.

To use one of the above mentioned modules they have to be installed as described in our manual_english_usbadpeth.

This one can be found in the Documentation Area of our homepage or on the driver CD.

The communication always runs with a speed of 115200 baud (8,n,1).

To transfer data a string is send or received depending on the used parameters.

The QUANCOM protocol sends back a return string to every send string which is clearly identifiable through the given <job_id>.

Those return strings could contain the read data, a success message for the written data or an error code, depending on the used parameters (read or write).

3.1 Sending a String

Now we will show you how to setup a string that should be send to the device.

<STX><job_id><Command><Width><Adress><Data><Checksum><CR>

<STX> Starting character which leads in the string.

<job_id>

Is used to make the send string2 clearly identifiable.

Length: 2 ASCII Chars (Hex-Value 0x00.. 0xFF)

Sample: The Hex-Value 0x1F will be transported as ASCII "1" (0x31) and "F" (0x46).

<Command>

- 'R' = Register write
- 'W' = Register read
- 'S' = Bit set
- 'C' = Bit reset

Length: 1 ASCII Chars

<width>

This defines the length of the data which is expected ("R") or send ("W").

It can contain the following values.

- 'B' = Byte (8 Bit)
- 'W' = Word (16 Bit)
- 'L' = Long (32 Bit)
- 'X' = eXtra Long (64 Bit)

Length: 1 ASCII Chars

<Adress>

The adress where you want to write to or read from.

Length: 4 ASCII Chars (0x0000..0xFFFF)

<Data>

The data that should be written may have a length of 0..16ASCII Chars. This is dependet by the chosen width **<width>**

<Checksum>

Has to be calculated by adding all parts of the string up to this point

Length: 2 ASCII Chars

<CR>

Carriage Return (0x0D) ends the string

Length: 1 ASCII Chars

After that a return string is expected by the device. How those return strings look like will be described on the following pages

The setup of the return string depends on the chosen command (W/R)

3.1.1 Send String Sample

Sample for a send string:

<STX>48WB0100FF54<CR>

Description:

Char	Meaning	ASCII Value	Transported Values
<STX>	Starts the string	'STX'	0x02
48	Job_ID	'4' '8'	0x34 0x38
W	W for Write	'W'	0x57
B	B for Byte	'B'	0x42
0100	Adress = 0x100	'0' '1' '0' '0'	0x30 0x31 0x30 0x30
FF	Data = 0xFF	'F' 'F'	0x46 0x46
	Checksum (Addition of all transported values from STX to DATA)		Sum of all Hex-Values = 0x254
54	Checksumme (Hex 0x254)	'5' '4'	0x35 0x34
<CR>	Carriage return ends the string (ASCII 0x0D)	<CR>	0x0D

Only the last two digits of the calculated checksum are used to build the checksum.

3.2 Return String

If the data transfer was successfully the return strings looks like the following:

<Startsign><job_id><Data><Checksum><CR>

<Startsign>

Starting sign which may contain the values "D" or "O". This depends on the chosen sending command.

"D" = Transfer OK incl. data

"O" = Transfer OK

Length: 1 ASCII Char

<job_id>

Used to identify the string

Length: 2 ASCII Chars

<Data>

Received data (only if **<Startsign>** = "D")

Length: If "D": 2-16 ASCII Chars

If "O": 0 ASCII Chars

<Checksum>

Has to be calculated by adding all parts of the string up to this point

Length: 2 ASCII Chars

<CR>

Carriage Return (0x0D) ends the string

Length: 1 ASCII Chars

3.2.1 Return String

Sample for a return string on successfully send data:

D48FF3C<CR>

Explanation:

Char	Meaning	ASCII Values	Transported Values
D	Starts the string OK inkl. send data	'D'	0x44
48	Job_ID 0x48	'4' '8'	0x34 0x38
FF	Data = 0xFF	'F' 'F'	0x46 0x46
	Checksum (Addition of all transported values from STX to DATA)		Sum of all Hex-Values = 0x13C
3C	Checksumme (Hex 0x13C)	'3' 'C'	0x33 0x43
<CR>	Carriage return ends the string (ASCII 0x0D)	<CR>	0x0D

The Job ID has to be changed in any consecutive read command
otherwise the old state of the register is read back.
This behavior only applies to read commands.

For example:

The first eight inputs are read from address 0x120 with the job ID "48". After that the inputs 9-15 are read back with the same Job-ID. In this case, the value from the second read command, is still the value of the first read register.
To avoid this situation the Job-IDs should change for each read command.

3.3 On Error

If an error occurs the return string will look as follows:

<Startsign><Errorcode><CR>

<Startsign>

"E" for error

Length: 1 ASCII Sign

<Errorcode>

The errorcode depending on the occured failure

Meaning of the error codes

Length: 1 ASCII Char

<CR> Carriage Return (0x0D) end the string

Length: 1 ASCII Char

3.3.1 Sample Error String

Sample for a return string on checksumerror:

E0<CR>

Erläuterung:

Char	Meaning	ASCII Values	Transported Values
E	Starting text for error code (ASCII 0x02)	'E'	0x45
0	Errorcode for Checksumfailure	'0'	0x30
<CR>	Carriage return ends the string (ASCII 0x0D)	<CR>	0x0D

3.4 Compendium

Parts of the data transfer when sending a string

<STX> 0x02	<job_id> 1 Byte	<Command> 1 Byte	<Length>	<Adress> 2 Byte	<Data> 0..8 Byte	<Checksum> 1 Byte	<CR>
0x02	0x00..0xFF	'W' Register write 'R' Register read 'S' Register set 'C' Register clear	'B' Byte (8Bit) 'W' Word(16Bit) 'L' Long (32Bit) 'X' XLong (64Bit)	xx	xx	xx	0x0D
1 Chars	2 Chars	1 Chars	1 Chars	4 Chars	0..16 Chars	2 Chars	1 Chars

Parts of the return string after sending a string

On Error:

<Identifier> 0x45	<error_code> 1 Byte	<CR> 1 Byte
'E'	0 = Checksum error 1 = Character error 2 = Command invallid 3 = Width invalid	0x0D
1 Chars	1 Chars	1 Chars

Transfer OK:

<Identifier> 0x4F	<job_id> 1 Byte	<Checksum> 1 Byte	<CR> 1 Byte
'O'	'0'..'0'	xx	0x0D
1 Chars	2 Chars	2 Chars	2 Chars

Transfer OK incl. Data:

<Identifier> 0x44	<job_id> 1 Byte	<Daten> 1 Byte	<Checksumme> 1 Byte	<CR> 1 Byte
'D'		'0'..'0'	xx	0x0D
1 Chars	2 Chars	0..16 Chars	2 Chars	2 Chars

4.1 Customer Communication and Help

Did you need help?

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

! Tip !

You can find an ASCII – text – file README.TXT, which includes changes made after printing of this user's manual on the QUANCOM installation CD.

! Important !

Informationen bereit: If you have further questions please contact our support team. For this case please prepare the following information:

- Exact type of the board.
- Version of the driver
- Version of the QLIB
- Operating system, hardware equipment and bus - system
- Name and version of the program, which reports the failure
- A detailed failure description. To make sure, please try to reproduce the failure, and describe exactly, which steps led to this failure.

Contact?

Die QUANCOM Internet Webseite
www.quancom.deThe QUANCOM internet website
http://www.quancom.de/

Per Fax
+49 22 36 / 89 92 - 49

Per E-Mail:

support@quancom.de

Adress:
QUANCOM INFORMATIONSSYSTEME GmbH
In der Flecht 14
50389 Wesseling

Wenn Sie Hilfe brauchen, erreichen Sie uns unter:
QUANCOM Hotline Deutschland If you need urgent help call:QUANCOM Hotline Germany
0 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 latest 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)" there., please check if you are using the latest software version of the QUANCOM software before contacting the QUANCOM support.

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 sending us the QUANCOM board to be repaired call:

Tel.: **+49 22 36 / 89 92 – 20**

If you send your QUANCOM board to us, 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.

4.2 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 help 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 will reproduce the problem

_____

_____

_____

_____

_____

4.3 Documentation comment form

QUANCOM Informationsysteme 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.

Titel: Serial / Ethernet Protocol

Edition date: 19.10.2010

Name: _____

Firma: _____

Adresse: _____

Telefon: _____

Fax: _____

Kommentar: _____

_____

_____

_____

_____

Email to: **support@quancom.de**

Fax to: **+49 2236 89 92 49**

Adress: **QUANCOM Informationssysteme GmbH**
In der Flecht 14
50389 Wesseling

4.4 Hardware and software configuration form

This form allows you to record the settings of your hardware and software. Complete this form each time you revise your software or hardware configuration, and use this form as a reference for your current configuration. Completing this form accurately before contacting QUANCOM Informationssysteme GmbH for technical support helps our application engineers 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: _____

4.5 Trademarks

Linux is registered trade-mark of Linus Torvalds.

MS, MS-DOS, Microsoft, Visual Basic, Windows, Windows Vista/XP/2000/NT/ME/98/95 is registered trade-mark of 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 is registered trade-mark of PCI Special Interest Group.

PCI Express is registered trade-mark of PCI-SIG.

National Instruments, LABVIEW is registered trade-mark of National Instruments 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.