

# Serielles / Ethernet Datenübertragungspr otokoll



*Copyright © QUANCOM Informationssysteme GmbH*

*Alle Angaben in diesem Handbuch sind nach sorgfältiger Prüfung zusammengestellt worden, gelten jedoch nicht als Zusicherung von Produkteigenschaften. QUANCOM haftet ausschließlich in dem Umfang, der in den Verkaufs- und Lieferbedingungen festgelegt ist. Weitergabe und Vervielfältigung dieses Handbuches und die Verwertung seines Inhaltes sowie der zum Produkt gehörenden Software sind nur mit schriftlicher Erlaubnis von QUANCOM gestattet. Änderungen, die dem technischen Fortschritt dienen, bleiben vorbehalten.*

*Wesseling, Oktober 2010 Version 4.1.4*

# Inhaltsverzeichnis

<b>Kapitel I</b>	<b>Überblick</b>	<b>1</b>
1	Einführung.....	1
2	Unsere Erfahrung ist ihr Gewinn.....	1
3	Kommunikation mit unseren Kunden.....	1
4	Änderungen zu diesem Handbuch und Softwareupdates.....	2
<b>Kapitel II</b>	<b>Die Registerbelegung</b>	<b>3</b>
1	Ausgänge.....	4
2	Eingänge.....	5
3	Eingangs-Flip-Flops.....	6
4	Zähler / Counter.....	7
5	Dip-Schalter.....	8
6	Kontrollregister.....	8
7	Statusregister.....	9
<b>Kapitel III</b>	<b>Protokoll</b>	<b>10</b>
1	Der gesendete String.....	11
	Beispiel für einen gesendeten String.....	13
	Anmerkungen zum Senden eines Strings.....	14
2	Der Rückgabestring.....	14
	Beispiel für einen Rückgabestring.....	16
3	Der Fehlerfall.....	17
	Beispiel für den Fehlerfall.....	18
4	Zusammenfassung.....	19
<b>Kapitel IV</b>	<b>Anhang</b>	<b>21</b>
1	Kunden Support und Hilfe.....	21
2	Technisches Support Formular.....	24
3	Dokumentations Formular.....	25
4	Hardware und Software Konfigurationsformular.....	26
5	Warenzeichen.....	27

## 1.1 Einführung

Wir beglückwünschen Sie zum Kauf Ihres QUANCOM Moduls. Sie haben sich für ein Produkt entschieden, dessen Eigenschaften und Funktionalität den neuesten Stand der Technik darstellen.

Zu den besonderen Eigenschaften dieser Karte gehören:

### Eigenschaften der Karte

- 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 Unsere Erfahrung ist ihr Gewinn

Wir von QUANCOM sind auf die Entwicklung für Hardware und Software spezialisiert und gehören mittlerweile zu einem der führenden Lieferanten für Meßtechnik und Automatisierung. In ihrem Entwicklungszentrum hat QUANCOM eine eindrucksvolle Produktpalette entwickelt.

## 1.3 Kommunikation mit unseren Kunden

QUANCOM möchte gerne Ihren Kommentar zu unseren Produkten und zu unseren Handbüchern erhalten und ist außerdem an Ihren Anwendungen interessiert, die Sie mit unseren Produkten entwickeln. Wir möchten gleichzeitig helfen, wenn Sie Probleme haben und um dies zu vereinfachen enthält dieses Handbuch Kommentar- und Konfigurationsformulare, mit denen man direkt mit uns in Verbindung treten kann. Diese Formulare befinden sich in dem Kapitel "**Dokumentationsformular**" am Ende dieses Handbuches.

## **1.4 Änderungen zu diesem Handbuch und Softwareupdates**

QUANCOM - Produkte zeichnen sich u.a. durch stetige Weiterentwicklung aus. Aktuelle Informationen über Änderungen können Sie den README - Dateien auf der Installationsdiskette oder CD entnehmen. Weitere Informationen und kostenlose Softwareupdates können Sie jederzeit auf den QUANCOM Internet – WWW-Seiten unter [www.quancom.de](http://www.quancom.de) erhalten.

In dem folgenden Abschnitt befinden sich die Tabellen mit den Registerbelegungen, mit deren Hilfe eine hardwareseitige Programmierung unserer Module ermöglicht wird.

Bitte beachten Sie, dass sich die Bitbreite der Befehle relativ zu den benutzten Registern verhält.

Das heißt, dass bei einem...

- **8 Bit Zugriff** jeweils nur ein 8 Bit Register geschrieben oder gelesen wird
  - Z.B. Adresse 0x100, 0x101, 0x102, ..
- **16 Bit Zugriff** 2 hintereinanderliegende 8 Bit Register geschrieben bzw. gelesen werden
  - Um z.B. den Wert 0x1234 auf Reg. 0x100 zu schreiben wird dies folgendermaßen gesplittet
    - Auf Adresse 0x100 wird 0x34 geschrieben
    - Auf Adresse 0x101 wird 0x12 geschrieben
- **32 Bit Zugriff** 4 hintereinanderliegende 8 Bit Register geschrieben bzw. gelesen werden
  - Um z.B. den Wert 0x12345678 auf Reg. 0x100 zu schreiben wird dies folgendermaßen ausgeführt
    - Auf Adresse 0x100 wird 0x78 geschrieben
    - Auf Adresse 0x101 wird 0x56 geschrieben
    - Auf Adresse 0x102 wird 0x34 geschrieben
    - Auf Adresse 0x103 wird 0x12 geschrieben
- **64 Bit Zugriff** 8 hintereinanderliegende 8 Bit Register geschrieben bzw. gelesen werden

Auf den folgenden Seiten sind die genauen Registerbelegungen in Tabellenform ausgearbeitet worden um eine möglichst große Übersichtlichkeit zu gewährleisten.

## 2.1 Ausgänge

### Registerbelegung für das Schreiben der Ausgänge

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

### Registerbelegung für das Zurücklesen der Ausgänge

Offset	R / W	Bit	Beschreibung
0x100	Read	0-7	Zurücklesen der Ausgänge 1 - 8
0x101	Read	0-7	Zurücklesen der Ausgänge 9 - 16
0x102	Read	0-7	Zurücklesen der Ausgänge 17 - 24
0x103	Read	0-7	Zurücklesen der Ausgänge 25 - 32
0x104	Read	0-7	Zurücklesen der Ausgänge 33 - 40
0x105	Read	0-7	Zurücklesen der Ausgänge 41 - 48
0x106	Read	0-7	Zurücklesen der Ausgänge 49 - 56
0x107	Read	0-7	Zurücklesen der Ausgänge 57 - 64

## 2.2 Eingänge

### Einlesen der Eingänge

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



## 2.3 Eingangs-Flip-Flops

### Einlesen der Flip-Flops

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

## 2.4 Zähler / Counter

### Counter 8 Bit Register

Offset	R / W	Bit	Wertigkeit (0..255)	Beschreibung
0xF8	Write	0	1	Reset Counter 0 (1 = aktiv)
0xF8	Write	1	2	Reset Counter 1 (1 = aktiv)
0xF9	Write	0	1	0 = Counter 0 ohne HW-Gate 1 = Counter 0 mit HW-Gate (nur Zählen wenn externes Gate=1)

### Counter lesen (nur 32 Bit Zugriffe)

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

Bsp:

Um Counter1 wieder zu resetten müssen folgende Befehle ausgeführt werden:

```
write 0xF8,2 // Reset Counter 1 = aktiv
```

```
write 0xF8,0 // Reset Counter 1 = inaktiv
```

## 2.5 Dip-Schalter

Die Dip-Schalter werden mit Hilfe des **Statusregisters** ausgelesen.

## 2.6 Kontrollregister

### Kontrollregister

Offset	R / W	Bit	Wertigkeit (0..255)	Beschreibung
0x07	Write	0	1	Reset (1 = aktiv)
0x07	-	1	-	unbenutzt
0x07	-	2	-	unbenutzt
0x07	Write	3,4, 5	8,16,32	Timeout Selection (siehe unten)
0x07	Write	6	64	TOUT_EN Timeout für die Ausgänge einschalten (1 = aktiv)
0x07	-	7	-	unbenutzt

### Timeout Selection

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

## 2.7 Statusregister

### Statusregister

Offset	R / W	Bit	Wertigkeit (0..255)	Beschreibung
0x07	Read	0	1	Zurücklesen des Timeout Flip/Flop (1=Timeout)
0x07	-	1	-	unbenutzt
0x07	-	2	-	unbenutzt
0x07	-	3	-	unbenutzt
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)

Offset	R / W	Bit	Wertigkeit (0..255)	Beschreibung
0x02	Write	-	-	Zurücksetzen des Timeout Flip/Flop



Das Timeout Flip/Flop wird automatisch nach einer definierten Zeit ohne Zugriff auf das Modul gesetzt.

Das Timeout Flip/Flop bleibt gesetzt, bis dieses wieder zurückgesetzt wird.

Ein "Write" auf Registeradresse 0x02 setzt das Timeout Flip/Flop zurück.

Im folgenden wird beschrieben, wie eine Datenübertragung über die serielle Schnittstelle von Statten geht.

Die Datenübertragung über die serielle Schnittstelle erfolgt mit einer Geschwindigkeit von 115200 Baud (8,n,1).

Bei der Datenübertragung über die serielle Schnittstelle wird immer ein String gesendet.

Zu jedem gesendeten String erhält man eine Antwort, welche durch die jeweilig übermittelte **<job\_id>** eindeutig zugeordnet werden kann.

In diesen Antworten sind je nach übertragenen Parametern (Read oder Write) die gelesenen Daten, eine Erfolgsmeldung für erfolgreich geschriebene Daten, oder ein Fehlercode enthalten.

Der Aufruf um beispielsweise Ausgang 32 einzuschalten sieht folgendermaßen aus:

Kommando: **<Register Set><Adresse = 0x100><Daten = 0x80000000>**

Um mit dem Einsatz von Hex-Werten alle Ausgänge abzuschalten gehen Sie wie folgt vor:

Kommando: **<Register Write><Adresse = 0x100><Daten = 0x00000000>**

### 3.1 Der gesendete String

Der Aufbau eines solchen zu sendenden Strings, welcher vom PC zum jeweiligen Device gesendet wird sieht folgendermaßen aus:

**<STX><job\_id><Kommando><Width><Adresse><Daten><Checksumme><CR>**

**<STX>** ist das Startzeichen (0x02), welches den String einleitet.

**<job\_id>** wird verwendet um die gesendeten Strings eindeutig zuzuordnen.

Länge: 2 ASCII Zeichen (Hex-Wert 0x00.. 0xFF)

Bsp: Der Hex-Wert 0x1F wird in ASCII als "1" und "F" übertragen. Dies sind die Werte 0x31 ("1") und 0x46("F").

**<Kommando>** besagt, folgendes:

- 'R' = Register lesen
- 'W' = Register schreiben
- 'S' = Bit setzen
- 'C' = Bit zurücksetzen

Länge: 1 ASCII Zeichen

**<width>** definiert die Länge der Daten, welche gesendet ("W"), bzw. beim Lesebefehl ('R') empfangen werden sollen

Es kann folgende Werte aufnehmen:

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

Länge: 1 ASCII Zeichen

**<Adresse>** die Adresse auf welche geschrieben werden bzw. von der gelesen werden soll.

Länge: 4 ASCII Zeichen (0x0000..0xFFFF)

**<Daten>** die zu schreibenden Daten können 0-16 ASCII-Zeichen betragen. Dies ist abhängig von der in **<width>** gewählten Grösse.

**<Checksumme>** wird aus den vorangegangenen Teilen des gesendeten Strings errechnet.

Länge: 2 ASCII-Zeichen

**<CR>** Carriage Return (0x0D) schließt den String ab

Länge: 1 ASCII Zeichen

Hiernach wird vom Gerät eine Antwort erwartet.

Wie diese aussieht wird auf den folgenden Seiten beschrieben.



Der Aufbau der Antworten variiert, je nach eingesetztem Kommando (W/R)

Die Kommandos 'S' und 'C' können nur in Verbindung mit einer Datenlänge von 32 und 64 Bit ('L' / 'X') verwendet werden.

### 3.1.1 Beispiel für einen gesendeten String

Beispiel für einen gesendeten String:

**ASCII: <STX>48WB0100FF54<CR>**

Erläuterung:

Zeichen	Bedeutung	ASCII Werte	Übertragene Werte
<STX>	Start Text Initialisiert den String	'STX'	0x02
48	In diesem Falle die Job_ID 0x48	'4' '8'	0x34 0x38
W	W für Write	'W'	0x57
B	B für Byte	'B'	0x42
0100	Adresse = 0x100	'0' '1' '0' '0'	0x30 0x31 0x30 0x30
FF	Daten = 0xFF	'F' 'F'	0x46 0x46
	Berechnete Checksumme (Addition aller "Übertragenen Werte" von STX bis Daten)		Summe der oben genannten Hex-Werte = 0x254
54	Steht für die Checksumme (Hex 0x254)	'5' '4'	0x35 0x34
<CR>	Carriage return schließt den String ab (ASCII 0x0D)	<CR>	0x0D



Bei der Checksumme werden nur die letzten beiden Zeichen des errechneten Hex-Wertes verwendet. Das heißt, der Wert 0x254 wird zur Checksumme 54



### 3.1.2 Anmerkungen zum Senden eines Strings

Es muss darauf geachtet werden, dass die Job\_Id bei jedem Lese, oder Schreibzugriff auf das Modul wechselt.

Sollten Sie beispielsweise vier mal die Eingänge 1 - 8 einlesen wollen so müssen die entsprechenden Strings folgendermaßen zusammengesetzt werden.

Der Wert der Job\_Id kann frei gewählt werden.

1. Lesen <STX>30RB0120BC<CR>
2. Lesen <STX>31RB0120BD<CR>
3. Lesen <STX>32RB0120BE<CR>
4. Lesen <STX>33RB0120BF<CR>

### 3.2 Der Rückgabestring

Bei einer erfolgreichen Übertragung sieht der Rückgabestring folgendermaßen aus:

**<Startzeichen><job\_id><Daten><Checksumme><CR>**

**<Startzeichen>** Einleitendes Zeichen welches die Werte 'D' und 'O' enthalten kann. Dies ist abhängig von der Sendart.

'D' = Übertragung OK inklusive übertragenden Daten

'O' = Übertragung OK

Länge: 1 ASCII Zeichen

**<job\_id>** Zur Identifizierung des Strings (Wird immer übertragen)

Länge: 2 ASCII Zeichen

**<Daten>** Empfangene Daten (Wird nur übertragen, wenn **<Startzeichen>** = 'D')

Länge bei Startzeichen 'D': 2-16 ASCII Zeichen

Länge bei Startzeichen 'O': 0 ASCII Zeichen

**<Checksumme>** wird aus den vorangegangenen Teilen des gesendeten Strings errechnet  
(Wird immer übertragen)  
Länge: 2 ASCII-Zeichen

**<CR>** Carriage Return (0x0D) schließt den String ab (Wird immer übertragen)  
Länge: 1 ASCII Zeichen

### 3.2.1 Beispiel für einen Rückgabestring

Beispiel für einen Rückgabestring bei erfolgreicher Sendung:

D48FF3C<CR>

Erläuterung:

Zeichen	Bedeutung	ASCII Werte	Übertragene Werte
D	Startzeichen Übertragung OK inklusive übertragenden Daten	'D'	0x44
48	In diesem Falle die Job_ID 0x48	'4' '8'	0x34 0x38
FF	Daten = 0xFF	'F' 'F'	0x46 0x46
	Berechnete Checksumme (Addition aller "Übertragenen Werte" von STX bis Daten)		Summe der oben genannten Hex-Werte = 0x13C
3C	Steht für die Checksumme (Hex 0x13C)	'3' 'C'	0x33 0x43
<CR>	Carriage return schließt den String ab (ASCII 0x0D)	<CR>	0x0D

Die Job-ID muss bei jedem aufeinander folgenden Lesebefehl gewechselt werden, da sonst immer der alte Zustand des Registers zurückgelesen wird. Dieses Verhalten gilt nur für Lesebefehle.



**Beispiel:** Es werden von Adresse 0x120 mit der Job-ID "48" die ersten acht Eingänge eingelesen. Danach werden mit gleicher Job-ID die Eingänge 9-15 eingelesen. In diesem Fall liefert das Modul beim zweiten Lesebefehl den Wert des zuerst zurück gelesenen Registers.

Um dies zu vermeiden sollten die Job-IDs bei den einzelnen Lesebefehlen wechseln.

### 3.3 Der Fehlerfall

Im Falle eines Fehlers sieht der Rückgabestring wie folgt aus:

**<Startzeichen><Fehlercode><CR>**

**<Startzeichen>** In diesem Falle ein 'E' für Error

Länge: 1 ASCII Zeichen

**<Fehlercode>** Der dem Fehler entsprechende Fehlercode

[Bedeutung der Fehlercodes](#)

Länge: 1 ASCII Zeichen

**<CR>** Carriage Return (0x0D) schließt den String ab

Länge: 1 ASCII Zeichen

### 3.3.1 Beispiel für den Fehlerfall

Beispiel für einen Rückgabestring bei einem Checksummenfehler:

E0<CR>

Erläuterung:

Zeichen	Bedeutung	ASCII Werte	Übertragene Werte
E	Start Text Initialisiert den String (ASCII 0x02)	'E'	0x45
0	Fehlercode für Checksummenfehler	'0'	0x30
<CR>	Carriage return schließt den String ab (ASCII 0x0D)	<CR>	0x0D

### 3.4 Zusammenfassung

Die Bestandteile der Datenübertragung beim Senden eines Befehls

<STX> 0x02	<job_id> 1 Byte	<Kommando> 1 Byte	<Länge>	<Adresse> 2 Byte	<Daten> 0..8 Byte	<Checksumme> 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 Zeichen	2 Zeichen	1 Zeichen	1 Zeichen	4 Zeichen	0..16 Zeichen	2 Zeichen	1 Zeichen

Bestandteile der Rückgabe nach dem Senden eines Befehls

**Bei einem Fehler:**

<Identifizier> 0x45	<error_code> 1 Byte	<CR> 1 Byte
'E'	0 = Checksum Fehler 1 = Zeichen Fehler 2 = Command ungültig 3 = Width ungültig	0x0D
1 Zeichen	1 Zeichen	1 Zeichen

## Übertragung OK:

<b>&lt;Identifizier&gt; 0x4F</b>	<b>&lt;job_id&gt; 1 Byte</b>	<b>&lt;Checksumme&gt; 1 Byte</b>	<b>&lt;CR&gt; 1 Byte</b>
'O'	'0'..'0'	xx	0x0D
1 Zeichen	2 Zeichen	2 Zeichen	2 Zeichen

## Übertragung OK inklusive Daten:

<b>&lt;Identifizier&gt; 0x44</b>	<b>&lt;job_id&gt; 1 Byte</b>	<b>&lt;Daten&gt; 1 Byte</b>	<b>&lt;Checksumme&gt; 1 Byte</b>	<b>&lt;CR&gt; 1 Byte</b>
'D'		'0'..'0'	xx	0x0D
1 Zeichen	2 Zeichen	0..16 Zeichen	2 Zeichen	2 Zeichen

## 4.1 Kunden Support und Hilfe



### Sie benötigen Hilfe?

Wenn Sie nicht wissen was Sie während einer Installation tun müssen, oder wie die Karte in Betrieb genommen wird, lesen Sie bitte dieses Handbuch.

### ! Tip !

Im Kapitel "Frequently asked questions" (Häufig gestellte Fragen) sind einige Antworten auf häufig gestellte Fragen. Sie können Ihnen bei der Problemlösung behilflich sein. Auf der QUANCOM Installations CD finden Sie in Textform die Datei README.TXT, welche alle wichtigen Änderungen beinhaltet.

### ! Wichtig !

Wenn Sie weitere Fragen haben, kontaktieren Sie unser Support-Team. Für diesen Fall halten Sie bitte folgende Informationen bereit:

- Genauer Karten-Typ
- Version der Treiber
- Version der QLIB
- Betriebssystem, Hardware-Ausstattung und Bussystem
- Name und Version von dem Programm, welches den Fehler ausgibt
- Eine genaue Fehlerbeschreibung (versuchen Sie den Fehler zu wiederholen, um diesen besser beschreiben zu können)



## Wen kann ich erreichen?

Die QUANCOM Internet Webseite  
[www.quancom.de](http://www.quancom.de)

Per Fax  
**+49 22 36 / 89 92 - 49**

Per E-Mail:  
**support@quancom.de**

Adresse:  
QUANCOM INFORMATIONSSYSTEME GmbH  
In der Flecht 14  
50389 Wesseling

Wenn Sie Hilfe brauchen, erreichen Sie uns unter:  
QUANCOM Hotline Deutschland  
0 22 36 / 89 92 - 20

Montags - Donnerstag  
von 9:00 bis 18:00  
Freitags  
von 9:00 bis 17:00

## Aktuelle Treiber

Auf unserer Internetseite <http://www.quancom.de> können sie immer die neusten Treiber Versionen und Updates finden. Zudem finden Sie ebenfalls viele andere Informationen und die "Frequently asked questions (FAQ's)". Bevor Sie uns kontaktieren, überprüfen Sie ob die neueste Version der QUANCOM Software installiert ist.

## Reparatur

Wenn Sie nicht genau wissen, ob die QUANCOM Karte defekt ist, rufen Sie unsere QUANCOM Hotline an:

**Tel.: +49 22 36 / 89 92 – 20**

Bevor Sie uns die Karte zur Reparatur schicken, rufen Sie unsere Hotline an:

**Tel.: +49 22 36 / 89 92 – 20**

Wenn Sie uns die Karte zurückschicken, legen Sie diese bitte in die Originalverpackung oder eine adäquate Verpackung, um einen Transportschaden zu verhindern. Zusätzlich bitten wir Sie, uns eine Kopie der Originalrechnung mitzuschicken.

## 4.2 Technisches Support Formular

Wenn Sie einen Internetzugang haben, öffnen Sie folgende URL in Ihrem Browser:

<http://www.quancom.de/quancom/qshop.nsf/techniksupport?OpenForm&deu>

Füllen Sie das Formular komplett aus bevor Sie sich an QUANCOM Informationssysteme GmbH wenden. Wenn Sie andere QUANCOM Hardware oder Software nutzen, fügen Sie das bitte dem Formular hinzu.

Name: \_\_\_\_\_

Firma: \_\_\_\_\_

Adresse: \_\_\_\_\_

Telefon: \_\_\_\_\_

Fax: \_\_\_\_\_

Computer / Prozessor: \_\_\_\_\_

Betriebssystem: \_\_\_\_\_

Grafikkarte: \_\_\_\_\_

Maus: \_\_\_\_\_

QUANCOM Karte \_\_\_\_\_

Andere installierte Karten: \_\_\_\_\_

Festplatte (Kapazität, frei): \_\_\_\_\_

Das Problem ist: \_\_\_\_\_

Auflistung der Fehlermeldung: \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Folgende Schritte führen zur Wiederholung des Problems:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_



## 4.4 Hardware und Software Konfigurationsformular

Dieses Formular hilft Ihnen die Einstellungen der Hardware und Software aufzulisten. Füllen Sie das Formular komplett aus bevor Sie sich an QUANCOM Informationssysteme GmbH wenden und nutzen Sie das Formular ebenfalls um die aktuelle Konfiguration nachzuschlagen.

### • QUANCOM Produkt:

Name / Name der Karte \_\_\_\_\_  
Interrupt Level \_\_\_\_\_  
DMA Kanal \_\_\_\_\_  
Basis I/O Adresse \_\_\_\_\_  
Betriebssystem \_\_\_\_\_

### • Andere Informationen

Computer Model \_\_\_\_\_  
Prozessor \_\_\_\_\_  
Taktfrequenz \_\_\_\_\_  
Grafikkarte \_\_\_\_\_  
Betriebssystem \_\_\_\_\_  
Programmiersprache \_\_\_\_\_  
Programmiersprachen-Version \_\_\_\_\_

### • Andere Karten im System

Basis I/O-Adresse anderer Karten \_\_\_\_\_  
DMA Kanäle anderer Karten \_\_\_\_\_  
Interrupt Level anderer Karten \_\_\_\_\_

## 4.5 Warenzeichen

Linux ist ein eingetragenes Warenzeichen von Linus Torvalds.

MS, MS-DOS, Microsoft, Visual Basic, Windows, Windows Vista/XP/2000/NT/ME/98/95 sind eingetragene Warenzeichen von Microsoft Corporation.

XT und PS/2 sind Warenzeichen und IBM, OS/2 und AT sind eingetragene Warenzeichen der International Business Machines Corporation.

Intel, Pentium ist ein eingetragenes Warenzeichen von Intel Corporation.

USB ist ein eingetragenes Warenzeichen von USB Implementers Forum Inc.

JAVA ist ein eingetragenes Warenzeichen von Sun Microsystems.

DELPHI und Pascal sind ein eingetragene Warenzeichen von Borland Corporation.

PCI ist ein eingetragenes Warenzeichen von PCI Special Interest Group.

PCI Express ist ein eingetragenes Warenzeichen der PCI-SIG.

Nationalinstruments, LABVIEW ist ein eingetragenes Warenzeichen von Nationalinstruments Corporation.

Agilent VEE ist ein eingetragenes Warenzeichen von Agilent Technologies.

Ethernet ist ein eingetragenes Warenzeichen der Xerox Corporation.

Bei anderen Produkt- und Firmennamen, die in dieser Anleitung erwähnt werden, könnte es sich um Marken ihrer jeweiligen Eigentümer handeln.