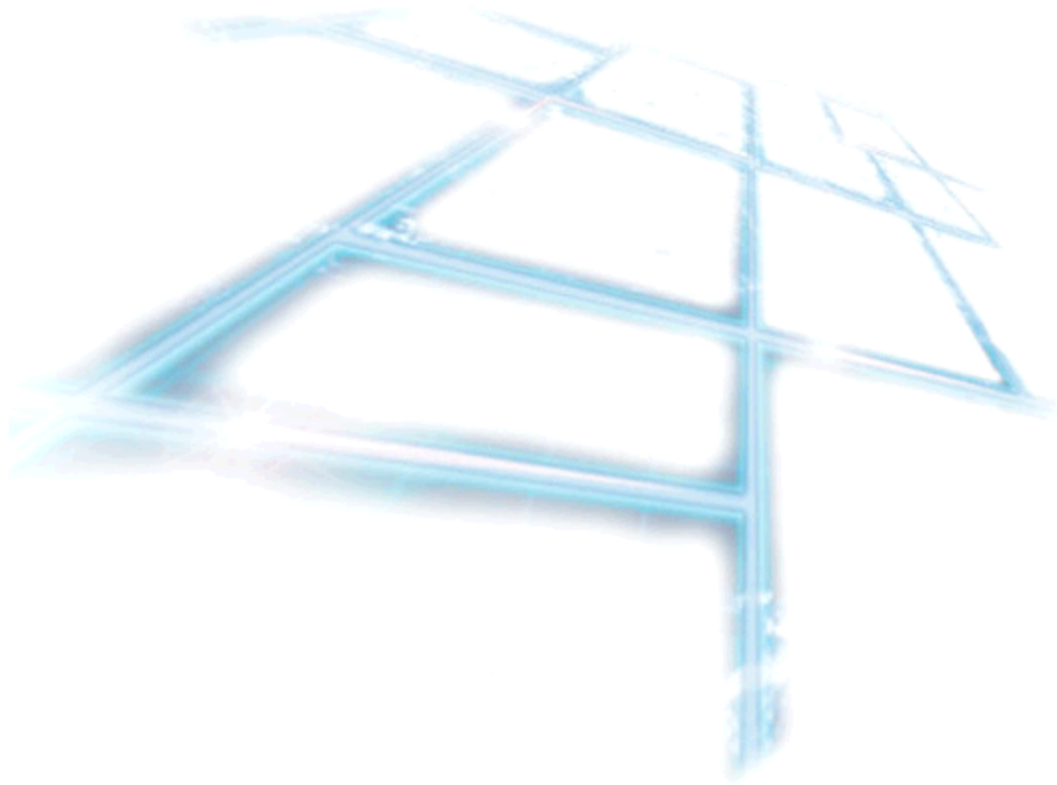


# PAR8R / PAR80



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 Vervielfältigung 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, März 2007 Version 4.1.0

# 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
5	Lieferumfang.....	2
<b>Kapitel II</b>	<b>Technische Hardware Beschreibung</b>	<b>3</b>
1	Modulinformationen PAR8R.....	3
	Kartenübersicht PAR8R.....	3
2	Modulinformationen PAR8O.....	4
	Kartenübersicht PAR8O.....	4
	Pinbelegung der Centronics-Buchse.....	5
	Einstellung des DIP-Schalter.....	6
	Datenwort für Adresse und Daten.....	7
3	Systemvoraussetzungen.....	8
4	Sicherheitsanweisungen.....	8
<b>Kapitel III</b>	<b>Programmierung der PAR8R, PAR8O</b>	<b>9</b>
1	Welche Software brauche ich ?.....	9
2	QLIB: High Level Programmierung (Windows XP / 2000 / NT4.x / ME / 98 / 95).....	10
	QLIB ( QUANCOM Driver Library ).....	10
	Schnellinstallation der Software für ISA und PAR.....	12
<b>Kapitel IV</b>	<b>QLIB Befehle</b>	<b>13</b>
1	Einfache QLIB Befehle.....	13
	Verwaltungsfunktionen.....	14
	Digitale Schreib- / Lesefunktionen.....	15
	Sonstige Funktionen.....	17
2	Erweiterte QLIB Befehle.....	20
	Verwaltungsfunktionen (Erweitert).....	20
	Digitale Lesefunktionen (Erweitert).....	24
	Digitale Schreibroutinen (Erweitert).....	26
3	Programmbeispiele in C für QLIB.....	28
	Programmierung der Ein-/Ausgänge, so wie diverser Spezialfunktionen.....	28
	Ansteuerung der Relais mit der QLIB in C/C++.....	30
	Lesen der Optokoppler Eingänge mit der QLIB in C/C++.....	31
<b>Kapitel V</b>	<b>Anhang</b>	<b>32</b>
1	Kunden Support und Hilfe .....	32
2	Technisches Support Formular .....	35
3	Dokumentations Formular .....	36
4	Hardware und Software Konfigurationsformular .....	37
5	Warenzeichen.....	38



# 1. Überblick

## 1.1 Einführung

Herzlichen Glückwunsch zum Kauf Ihrer QUANCOM Meßtechnik Karte. Sie sind nun stolzer Besitzer einer modernen Meß- Steuer- Regelkarte, die den neusten Stand der Technik darstellt und deren Eigenschaften und Funktionalität sich mit jedem anderem Gerät dieser Klasse messen kann und sie sogar noch übertrifft. Zu den besonderen Eigenschaften dieser Karte gehören:

### Eigenschaften der Karte

- einfach programmierbar
- diverse Beispielprogramme in verschiedenen Programmiersprachen
- Treiberunterstützung unter Windows Vista, XP, 2000, NT und 98/95

## 1.2 Unsere Erfahrung ist ihr Gewinn

Wir von QUANCOM sind auf die Entwicklung für Hardware und Software spezialisiert. QUANCOM gehört 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. Wir sind an Ihren Anwendungen interessiert, die Sie mit unseren Produkten entwickeln. Wir möchten gleichzeitig helfen, wenn Sie Probleme haben. Um das 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 **”Dokumentations Formular”** 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.

## 1.5 Lieferumfang

- Meßtechnik-Modul
- QUANCOM CD mit Benutzerhandbuch (PDF)

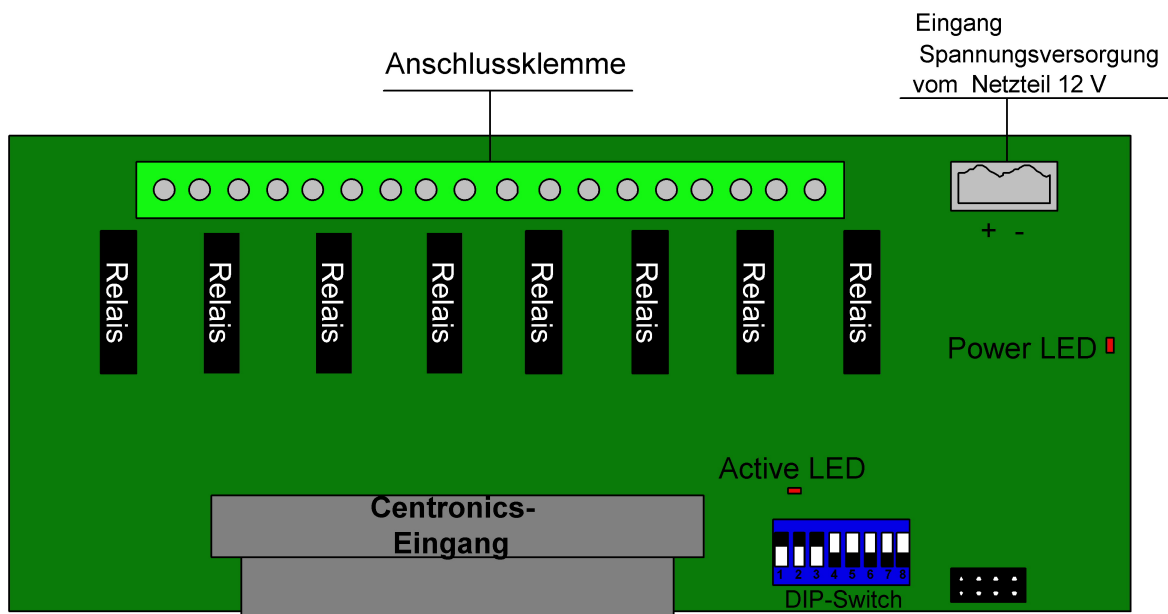
Sollten eine oder mehrere Komponenten fehlen wenden Sie sich bitte an Ihren Händler. QUANCOM behält sich das Recht vor, Änderungen im Lieferumfang ohne Vorankündigung vorzunehmen.

## 2. Technische Hardware Beschreibung

### 2.1 Modulinformationen PAR8R

- 8 Relais-entkoppelte Ausgänge
- 8 LED zur Funktionskontrolle aller Ausgänge
- Einfache Programmierung durch die QLIB
- Anschluss über LPT
- Spannungsversorgung über 12V Spannungsquelle

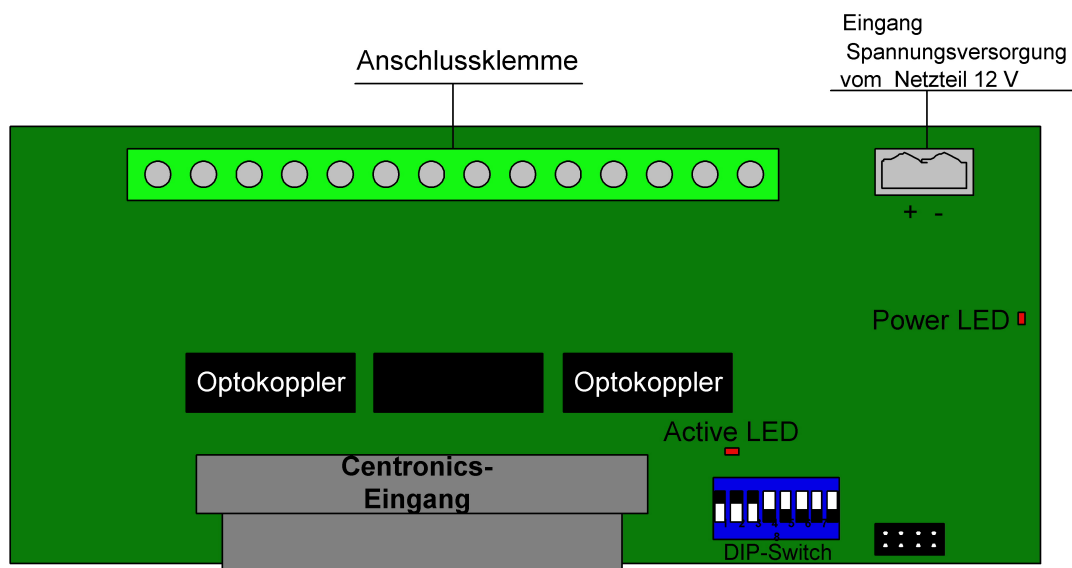
#### 2.1.1 Kartenübersicht PAR8R



## 2.2 Modulinformationen PAR80

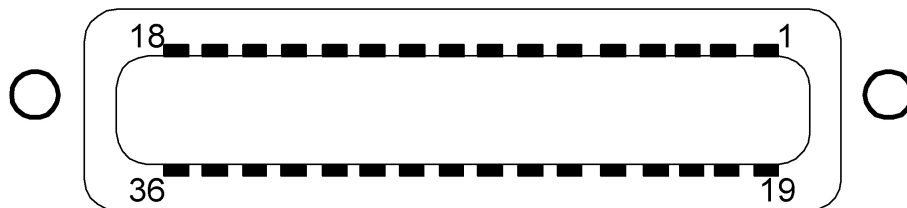
- 8 optisch getrennte Eingänge zur Erfassung von Eingangssignalen (12 - 30V veränderbar auf 5 - 12V)
- 8 LED zur Funktionskontrolle aller Eingänge
- Einfache Programmierung durch die QLIB
- Anschluss über LPT
- Spannungsversorgung über 12V Spannungsquelle

### 2.2.1 Kartenübersicht PAR80



## 2.2.2 Pinbelegung der Centronics-Buchse

### Centronics-Buchse (Draufsicht)



Pinbelegung:			
Pin	Signal	Pin	Signal
1	Strobe	19	GND
2	Data 0	20	.
3	Data 1	21	.
4	Data 2	22	.
5	Data 3	23	.
6	Data 4	24	.
7	Data 5	25	.
8	Data 6	26	.
9	Data 7	27	.
10	ACK	28	.
11	BUSY	29	.
12	PE (Paper END)	30	GND
13	SELKT	31	Nc.
14	Nc.	32	Nc.
15	Nc.	33	Nc.
16	GND Signal	34	Nc.
17	Nc.	35	Nc.
18	Nc.	36	Nc.

### 2.2.3 Einstellung des DIP-Schalter

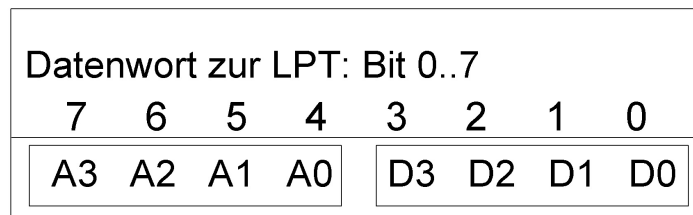
Dip-Switch Einstellungen								
1	2	3	4	5	6	7	8	Wirkung
On	On	On	x	x	x	x	x	Adresse0
Off	On	On	x	x	x	x	x	Adresse1
On	Off	On	x	x	x	x	x	Adresse2
Off	Off	On	x	x	x	x	x	Adresse3
On	On	Off	x	x	x	x	x	Adresse4
Off	On	Off	x	x	x	x	x	Adresse5
On	On	On	x	x	x	x	x	Adresse6
Off	Off	Off	x	x	x	x	x	Adresse7
x	x	x	On	Off	x	x	x	ACK= 1
x	x	x	Off	On	x	x	x	ACK= 0
x	x	x	x	x	On	x	x	Busy= 0
x	x	x	x	x	Off	x	x	Busy= X
x	x	x	x	x	x	On	x	PE= 0
x	x	x	x	x	x	Off	x	PE= X
x	x	x	x	x	x	X	On	SELKT= 0
x	x	x	x	x	x	X	Off	SELKT= x

DIP-Switch 6 (Busy) wird bei Ausgabekarten auf 0 gesetzt. Wird eine Eingabekarte an der gleichen Schnittstelle verwendet, so ist der Schalter 6 nur bei einem Modul auf 1 zu stellen. "1" Bedeutet hier: Busy = PullDown. Die anderen Handshakesignale können je nach Anwendung auf 1 oder 0 (0=offen) gesetzt werden.

## 2.2.4 Datenwort für Adresse und Daten

Die Adressierung des Moduls erfolgt jeweils über die 4 höchsten Datenbits. Insgesamt sind somit 16 Adressierungen möglich. Da aber die meisten Anwendungen 8 Bit verlangen, werden je Modul zwei Moduladressen benötigt. Mit einem Schreibbefehl werden sowohl die Datenbits 0..3 als auch die Adressen 4..7 übertragen. Die Übertragung erfolgt mit einem OUT-Befehl auf die Adresse, an der sich die LPT-Schnittstelle befindet, gefolgt von einem Low-Impuls, der den Strobe (Übernahme) erzeugt. Der Strobeimpuls wird auf einer zweiten LPT-Adresse mit einfachen OUT-Befehlen generiert. Im folgenden Beispiel liegt der Strobe (Pin 1) auf Adresse 037A hex (entsprechend LPT1:), Bit 0.

**Beispiel 10**    OUT &H037A,0: REM H  
:  
          20    OUT &H037A,1: REM L  
          30    OUT &H037A,0 REM H



Durch das Schreiben von zwei Bytes (zu je 4Bit) auf zwei Moduladressen wird ein Byte (8 Bit) am GAL-Ausgang erzeugt. Die erste Adresse erzeugt das LSB, die Zweite das MSB.

## 2.3 Systemvoraussetzungen

- Personal Computer: Die QUANCOM Karten laufen in einem IBM-AT Computer mit 80X86 oder kompatibel (z.B. 80386 / 80486 / Pentium).
- Bus: Ihr Computer muß den entsprechenden Bus haben (PCI).



Mehr Informationen hierzu finden Sie im Abschnitt Leitfaden zur "**Technische Hardware Beschreibung**".

## 2.4 Sicherheitsanweisungen

Im Interesse Ihrer eigenen Sicherheit und einer einwandfreien Funktion Ihrer neuen QUANCOM-Karte beachten Sie bitte die folgenden Hinweise:

- Ziehen Sie vor dem Öffnen des Rechners den Netzstecker um sicherzustellen, dass das Gerät nicht unter Spannung steht.
- Da PC-Karten empfindlich gegen elektrostatische Aufladungen sind ist es wichtig, sich vorher zu entladen bevor die Karte mit den Händen oder dem Werkzeug berührt wird. Dies geschieht am einfachsten, wenn Sie vorher ein metallisches Gehäuseteil berühren.
- Halten Sie die Karte immer am Rand fest und vermeiden Sie ein Anfassen der IC's.
- Legen Sie die Karten immer auf einen elektrogeschützten Gegenstand solange sich die Karten noch nicht im PC Gehäuse befinden.



Veränderungen, die ohne ausdrückliche Genehmigung der QUANCOM Informationssysteme GmbH an dem Gerät vorgenommen werden, führen zum Erlöschen der Betriebserlaubnis und der CE Zertifizierung.

## 3. Programmierung der PAR8R, PAR8O

### 3.1 Welche Software brauche ich ?

Die Software die ich brauche ist abhängig von der jeweiligen Anwendung und Betriebssystem. Um Zugriff per Programm auf die Karte zu bekommen sind bestehen folgende Möglichkeiten:

- **Methode 1:** Direkter I/O Zugriff ( Direkter Zugriff auf die Hardware Register des Betriebssystems)
- **Methode 2:** High-Level Programmierung (Zugriff auf die Karte über die QLIB) unter Windows XP / 2000 / NT / ME / 98 / 95. Hiermit ist es möglich die Karte z.B. über Visual-C, Visual-Basic, Borland Delphi, Lotus Notes u.a. Compilern und Interpretern anzusprechen.
- **Methode 3:** Installation der QLIB in Zusammenhang mit einem anderen Programm.
- **Methode 4:** Ansprechen der Karte unter Windows 3.x und Dos

**Methode 3** erlaubt einem das man die QUANCOM PAR48IO mit einer bestehenden Software laufen lassen kann z.B. LabView oder HP VEE . Dafür müssen Sie als erstes die QLIB von der Installations CD installieren. Hinweise rund um die QLIB und Installation entnehmen Sie bitte dem QLIB Handbuch, welches sich ebenfalls auf der CD befindet. Auf der CD finden Sie gleichzeitig einige Beispielprogramme für LabView und HP VEE.

## 3.2 QLIB: High Level Programmierung (Windows XP / 2000 / NT4.x / ME / 98 / 95)

### 3.2.1 QLIB ( QUANCOM Driver Library )

Die QLIB (die Abkürzung für **QUANCOM LIBrary**) bietet die Möglichkeit, alle QUANCOM-Karten unter den Betriebssystemen Windows XP/2000/NT und ME/98/95 und den Programmiersprachen C/C++/Delphi/Visual Basic anzusprechen. Sie wird zu allen QUANCOM-Karten mitgeliefert und gestattet dem Anwender durch die Einfachheit der Befehle, die QLIB in eigene Applikationen einzubinden. Die Befehle und Funktionen gelten für alle Betriebssysteme.

### Unterstützte Betriebssysteme:

- Microsoft Windows Vista / XP / 2000 / NT 4.0 / ME / 98 / 95 und Linux

### Unterstützte Compiler:

#### C / C++

- Borland C++ 3.1, 4.x, 5.x
- Microsoft® Visual C++ 1.x, 2.x, 4.x, 5.x, 6.x

#### Pascal

- Borland Turbo Pascal

#### Delphi

- Borland Delphi

#### Basic

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

## Grafische Programmiersprache:

- Agilent VEE von Agilent
- LabView® von National Instruments

### 3.2.2 Schnellinstallation der Software für ISA und PAR



Für die Installation der Treiber und Laufzeitumgebung sind Administratorenrechte erforderlich. Ohne die entsprechenden Rechte kann der Treiber und die Laufzeitumgebung nicht korrekt installiert werden. Eine detailliertere Installationsanleitung finden Sie in unserem Manual QLIB.

#### Windows XP/2000/95/98/ME/NT4.x

PAR-Modul anschließen / ISA-Karte einstecken und die Jumpereinstellungen notieren.

Rechner starten

Starten Sie die Datei QUANCOM.EXE im Hauptverzeichnis der Installations-CD und folgen Sie den Anweisungen des Programmes.

Rechner neu starten

Öffnen Sie nun die Konfiguration, die sich unter START | Programme | QLIB32 | QLIB 32-Bit Configuration Utility befindet. Klicken Sie hier auf Hinzufügen und wählen nun Ihre Karte aus.

Es öffnet sich jetzt automatisch der Dialog mit dem Sie die I/O Adresse der Karte auswählen können. Überprüfen Sie ob die Einstellung der E/A Adresse mit der Jumperstellung auf Ihrer Karte übereinstimmt. Wenn nicht, öffnen Sie bitte durch doppelklicken auf "E/A Basisadresse" den Dialog zur Einstellung E/A Adresse.

Rechner neu starten

PAR-Modul / ISA-Karte mit dem mitgelieferten Testprogramm auf Funktionstüchtigkeit prüfen.

## 4. QLIB Befehle

Stellen Sie sicher, dass die QLIB (QUANCOM Library ) installiert ist.

Für weitere Informationen über die Installation und wie man die entsprechenden Dateien in die Anwendung einbindet entnehmen Sie bitte der "QLIB " Dokumentation.

Diese Kapitel beschreibt die speziellen Kommandos, um die PAR48IO mit der QLIB benutzen zu können.

### 4.1 Einfache QLIB Befehle

Die folgende Auflistung enthält alle von dieser Karte verwendeten QLIB-Befehle. Diese unterscheiden sich in erweiterte (QAPIExt...) und einfache (QAPI...) Funktionen. Bei den einfachen Funktionen werden keine Kartenhandle's (Kartenparameter) übergeben. Dadurch wird auch nur die erste vom System erkannte Karte eines Typs angesprochen und verwendet. Sollen mehrere Karten eines Types in einem System verwendet werden, so müssen die erweiterten Funktionen verwendet werden.

### 4.1.1 Verwaltungsfunktionen

#### QAPINumOfCards

Mit der Funktion **QAPIExtNumOfCards** wird abgefragt, wieviele unterschiedliche Karten- bzw. Modultypen von der QLIB unterstützt werden.

ULONG **QAPINumOfCards** (void);

#### QAPIGetLastError

Die **QAPIGetLastError** Funktion liefert den letzten Fehlercode des aufrufenden Threads.

Der letzte Fehlercode wird dabei für jeden Thread gesondert gespeichert. Mehrere Threads überschreiben ihre Fehlercodes nicht gegenseitig.

ULONG **QAPIGetLastError** (void);

#### QAPIGetLastErrorCode

Die **QAPIGetLastErrorCode** Funktion liefert den letzten erweiterten Fehlercode eines vorher aufgerufenen **QAPIGetLastError** Befehls.

ULONG **QAPIGetLastErrorCode** (void);

## 4.1.2 Digitale Schreib- / Lesefunktionen

### QAPIGetDI

Mit der Funktion **QAPIGetDI** wird der Zustand eines 32 Bit breiten Digitalkanals einer DI-Karte ausgegeben.

ULONG **QAPIGetDI** (ULONG **cardhandle**,ULONG **channel**);

Funktionsprototypen für Delphi und Vb

#### Parameter

##### **cardhandle**

Gibt das Handle einer geöffneten Karte an.

##### **channel**

Gibt den Kanal an, von dem der Digitalwert eingelesen werden soll.

## QAPIPutDO

Mit der Funktion **QAPIGetDO** wird der Zustand eines 32 Bit breiten Digitalkanals einer DI-Karte ausgegeben.

ULONG **QAPIGetDI** (ULONG **cardhandle**,ULONG **channel**);

Funktionsprototypen für Delphi und Vb

### Parameter

#### **cardhandle**

Gibt das Handle einer geöffneten Karte an.

#### **channel**

Gibt den Kanal an, von dem der Digitalwert eingelesen werden soll.

### 4.1.3 Sonstige Funktionen

#### QAPIExtGetCardInfo

Mit der Funktion **QAPIExtGetCardInfo** lassen sich Informationen über eine Karte einlesen.

```
LPCARDDATAS QAPIExtGetCardInfo(ULONG cardid);
```

Funktionsprototypen für Delphi und VB

#### Parameter

##### **cardid**

Gibt die ID der Karte an, über welche Informationen abgefragt werden sollen. Der Feld

SizeOf muss mit der Grösse der Struktur initialisiert werden, sonst schlägt die Funktion fehl.

## QAPIExtGetCardInfoEx

Mit der Funktion **QAPIExtGetCardInfoEx** können Informationen einer Karte abgerufen werden. Diese werden in dem vom der Anwendung übergebenem Speicher geschrieben.

```
ULONG QAPIExtGetCardInfoEx(ULONG cardid, LPCARDDATAS lpcd);
```

Funktionsprototypen für Delphi und VB

### Parameter

#### **cardnum**

Gibt die ID der Karte an, über welche Informationen abgefragt werden sollen

#### **lpcd**

Ist ein Zeiger auf eine Struktur vom Typ CARDDATAS

## QAPIExtReleaseCardInfo

Mit der Funktion **QAPIExtReleaseCardInfo** werden die mit **QAPIExtGetCardInfo** abgefragten Karteninformationen freigegeben.

```
void QAPIExtReleaseCardInfo(LPCARDDATAS lpcarddatas);
```

Funktionsprototypen für Delphi und VB

### Parameter

#### **lpcarddatas**

Zeigt auf eine Struktur vom Typ CARDDATAS.

## 4.2 Erweiterte QLIB Befehle

### 4.2.1 Verwaltungsfunktionen (Erweitert)

#### QAPIExtOpenCard

Mit der Funktion **QAPIExtOpenCard** wird eine Karte geöffnet.

```
ULONG QAPIExtOpenCard (ULONG cardid,ULONG devnum);
```

Funktionsprototypen für Delphi und VB

#### Parameter

##### **cardid**

Gibt die ID der Karte an, welche geöffnet werden soll.

##### **devnum**

Gibt die Device-Nummer der Karte an, die geöffnet werden soll.

## QAPIExtCloseCard

Mit der Funktion **QAPIExtCloseCard** wird eine Karte geschlossen.

```
void QAPIExtCloseCard(ULONG cardhandle);
```

Funktionsprototypen für Delphi und VB

### Parameter

#### **cardhandle**

Gibt das Handle einer geöffneten Karte an.

## QAPIExtNumOfCards

Mit der Funktion **QAPIExtNumOfCards** wird abgefragt, wieviele unterschiedliche Karten- bzw. Modultypen von der QLIB unterstützt werden.

```
ULONG QAPIExtNumOfCards (void);
```

Funktionsprototypen für Delphi und VB

### Parameter

Diese Funktion benötigt keine Parameter.

## QAPIGetLastErrorStringEx

Die QAPIGetLastErrorStringEx dient dazu einen String zu erzeugen, der eine lesbare Fehlermeldung aus den QLIB Fehlercodes erzeugt, die von der Funktion QAPIGetLastError bzw. QAPIGetLastErrorCode zurückgeliefert werden. Ein Beispiel finden Sie unter Fehler codes .

```
ULONG QAPIGetLastErrorStringEx(char* buffer, ULONG bufferize);
```

Funktionsprototypen für Delphi und Vb

### Parameter

#### **buffer**

Zeiger auf einen Puffer der den nullterminierten Fehlerstring erhält.

#### **bufferize**

Dieser Parameter gibt die Grösse des übergebenen Puffers in Bytes an.

## 4.2.2 Digitale Lesefunktionen (Erweitert)

### QAPIExtReadDI1

Mit den Funktion **QAPIExtReadDI1** wird der Zustand eines 1 Bit breiten Digitalkanals einer DI-Karte eingelesen.

```
ULONG QAPIExtReadDI1 (ULONG cardhandle,ULONG channel,ULONG mode,);
```

Funktionsprototypen für Delphi und VB

#### Parameter

##### **cardhandle**

Gibt das Handle einer geöffneten Karte an.

##### **channel**

Gibt den Kanal an, von dem der Digitalwert eingelesen werden soll.

##### **mode**

Dieser Wert ist für zukünftige Erweiterungen reserviert. Es muß daher mit Wert 0 angegeben werden.

Ausnahme: Beim PAR48IO Modul kann mit einer "1" das automatische "Latches" der Eingänge deaktiviert werden.

## QAPIExtReadDI8

Mit der Funktion **QAPIExtReadDI8** wird der Zustand eines 8 Bit breiten Digitalkanals einer DI-Karte eingelesen.

ULONG **QAPIExtReadDI8** (ULONG **cardhandle**,ULONG **channel**,ULONG **mode**);

Funktionsprototypen für Delphi und Vb

### Parameter

#### **cardhandle**

Gibt das Handle einer geöffneten Karte an.

#### **channel**

Gibt den Kanal an, von dem der Digitalwert eingelesen werden soll.

#### **mode**

Dieser Wert ist für zukünftige Erweiterungen reserviert. Es muß daher mit Wert 0 angegeben werden. Ausnahme: Beim PAR48IO Modul kann mit einer "1" das automatische "Latches" der Eingänge deaktiviert werden.

### 4.2.3 Digitale Schreibroutinen (Erweitert)

#### QAPIExtWriteDO1

Mit der Funktion **QAPIExtWriteDO1** wird ein 1 Bit breiter Digitalwert auf den Kanal einer DO-Karte ausgegeben.

```
void QAPIExtWriteDO1 (ULONG cardhandle,ULONG channel,ULONG value,ULONG mode);
```

Funktionsprototypen für Delphi und Vb

#### Parameter

##### **cardhandle**

Gibt das Handle einer geöffneten Karte an.

##### **channel**

Gibt den Kanal an, auf dem der Digitalwert ausgegeben werden soll.

##### **value**

Gibt den Digitalwert an, welcher ausgegeben werden soll.

##### **mode**

Dieser Wert ist für zukünftige Erweiterungen reserviert. Es muß daher mit Wert 0 angegeben werden. Ausnahme: Beim PAR48IO Modul kann mit einer "1" das automatische "Latches" der Ausgänge deaktiviert werden.

## QAPIExtWriteDO8

Mit der Funktion **QAPIExtWriteDO8** wird ein 8 Bit breiter Digitalwert auf den Kanal einer DO-Karte ausgegeben.

```
void QAPIExtWriteDO8 (ULONG cardhandle,ULONG channel,ULONG value,ULONG mode);
```

Funktionsprototypen für Delphi und Vb

### Parameter

#### **cardhandle**

Gibt das Handle einer geöffneten Karte an.

#### **channel**

Gibt den Kanal an, auf dem der Digitalwert ausgegeben werden soll.

#### **value**

Gibt den Digitalwert an, welcher ausgegeben werden soll.

#### **mode**

Dieser Wert ist für zukünftige Erweiterungen reserviert. Es muß daher mit Wert 0 angegeben werden. Ausnahme: Beim PAR48IO Modul kann mit einer "1" das automatische "Latches" der Ausgänge deaktiviert werden.

## 4.3 Programmbeispiele in C für QLIB

Die folgenden Programmbeispiele zeigen, wie einfach Programme für die QUANCOM-Karten mit Hilfe der QLIB, geschrieben werden können.

### 4.3.1 Programmierung der Ein-/Ausgänge, so wie diverser Spezialfunktionen.

Der fettgedruckte Text beschreibt, wie man die QLIB in ein MFC / C++ / C Projekt einbindet. Nach der Installation der QLIB befindet sich das sofort kompilierbare Beispielprojekt im Ordner `\samples\usbopto\vc` des Installationsverzeichnis.

#### Beispiel zu den Funktionen PAR80 od. PAR8R

```
// usbmodul.cpp : Sample project for the PAR8R and PAR80 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"
#include <conio.h>
#include <stdio.h>

#define OUT1    0x1
#define OUT2    0x2
#define OUT3    0x4
#define OUT4    0x8
#define OUT5    0x10
#define OUT6    0x20
#define OUT7    0x40
#define OUT8    0x80

int main(int argc, char* argv[])
{
    ULONG handle;
    char ch;
    char relmod=1;

    //
    // Open the PAR Module (PAR8R or PAR80 )
    //
    handle = QAPIExtOpenCard(PAR8R,0);
    if ( handle == 0 ) {
        handle = QAPIExtOpenCard(PAR8R,0);
        relmod=0;
    }

    //
    // If there are no usb modules terminate application
    //
    if ( handle == 0 ) {
        MessageBox(NULL, "No PAR Modules found!","Error", MB_OK);
    }
}
```

```

        return FALSE;
    }

    // Ok, we found a QUANCOM PAR Module
if (relmod) {
    // -----
    // PART 1: Setting the outputs
    // -----
    ULONG lines = 0;

    //
    // Reset all lines to "Low"
    //
    printf("Reset all lines to 'Low' ( Press return to continue ):\n");
    QAPIExtWriteD08(handle, 0, lines, 0);
    ch = getchar();

    //
    // Set the outputs OUT2,OUT4 and OUT7 to "High" ( 8-Bit )
    //
    printf("Set OUT2,OUT4 and OUT7 to 'High' ( Press return to continue ):\n");
    lines = OUT2 + OUT4 + OUT7;
    QAPIExtWriteD016(handle, 0, lines, 0);
    ch = getchar();
    //
    // Set the output OUT1, OUT3 and OUT5 to "High" ( 16-Bit )
    //
    printf("Set OUT1, OUT3 and OUT5 to 'High' ( Press return to continue ):\n");

    lines = OUT1 + OUT3 + OUT5;
    QAPIExtWriteD08(handle, 0, lines, 0);
    ch = getchar();

    //
    // Reset line OUT3 to "Low"
    //
    printf("Reset line OUT3 to 'Low' ( Press return to continue ):\n");
    QAPIExtWriteD01(handle, 3 - 1, FALSE, 0);
    ch = getchar();

    //
    // Set line OUT3 to "High"
    //
    printf("Set line OUT3 to 'High' ( Press return to continue ):\n");
    QAPIExtWriteD01(handle, 3 - 1, TRUE, 0);
    ch = getchar();

    //
    // Reset all lines to "Low"
    //
    printf("Reset all to 'Low' ( Press return to continue ):\n");
    lines = 0;
    QAPIExtWriteD08(handle, 0, lines, 0);
    ch = getchar();
} else {
    // -----
    // PART 2: Reading the inputs ( and detect changed inputs )
    //
    // This part reads the state of the input lines.
    // -----
    int i;
    while (!kbhit()) {
        // read the current state from the inputs
        lines = QAPIExtReadDI16(handle, 0, 0);
        printf("\n-----\n");
        printf("Current input states\n");
        printf("IN1 IN2 IN3 IN4 IN5 IN6 IN7 IN8\n");
        for (i=0;i<7;i++) {
            if ( lines & 1<<i) {
                printf(" 1 ");
            } else {
                printf(" 0 ");
            }
        }
        printf("\n-----\n");
        printf("Press return to stop reading the inputs every 5 seconds ... \n");
        for (int j=0;(j<10) && !kbhit();j++) {
            Sleep(500);
        }
    }
    ch = getchar();
}

    QAPIExtCloseCard(handle);

    return 0;
}

```

## 4.3.2 Ansteuerung der Relais mit der QLIB in C/C++

### Beispiel zur Ansteuerung der Relais

```
#include <windows.h>
#include <stdio.h>
#include <conio.h>

#include "qlib.h"

/*=====
Main program
=====*/

void main () {
    ULONG handle;    /*Handle of the PAR8R

    if ((handle=QAPIExtOpenCard(PAR8R,0L)) == 0L) {
        printf("Couldn't open PAR8R \n");
        return;
    }

    for (;;) {
        if (kbhit() != 0 && getch() == 27) break;

        QAPIExtWriteD08(handle,0L,0x00L,0L);
        Sleep(500);
        QAPIExtWriteD08(handle,0L,0xFFL,0L);
        Sleep(500);
        QAPIExtWriteD08(handle,0L,0x55L,0L);
        Sleep(500);
        QAPIExtWriteD08(handle,0L,0xAAL,0L);
        Sleep(500);
    }

    QAPIExtCloseCard(handle);
}
```

### 4.3.3 Lesen der Optokoppler Eingänge mit der QLIB in C/C++

#### Beispiel zum Lesen der Optokoppler

```
#include <windows.h>
#include <stdio.h>
#include <conio.h>

#include "qlib.h"

/*=====
Main program
=====*/

void main () {
    ULONG handle;    /* Handle of the PAR8O */

    if ((handle = QAPIExtOpenCard(PAR8O,0L)) == 0L) {
        printf("Couldn't open PAR8O \n");
        return;
    }

    for (;;) {
        if (kbhit() != 0 && getch() == 27) break;

        printf("%02lx\n",QAPIExtReadDI8(handle,0L,0L));
        Sleep(500);
        printf("%02lx\n",QAPIExtReadDI8(handle,0L,0L));
        Sleep(500);
        printf("%02lx\n",QAPIExtReadDI8(handle,0L,0L));
        Sleep(500);
        printf("%02lx\n",QAPIExtReadDI8(handle,0L,0L));
        Sleep(500);
    }

    QAPIExtCloseCard(handle);
}
```

## 5. Anhang

### 5.1 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.

## 5.2 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:

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

### 5.3 Dokumentations Formular

QUANCOM Informationssysteme GmbH möchte Ihren Kommentar zum Produkt und zu der über diese Dokumentation oder eines Produktes. Diese Informationen helfen uns unsere Qualität zu verbessern.

Titel: PAR8R / PAR8O

Erstellungsdatum: 07.03.2007

Nehmen Sie Stellung zur Kompetenz, Übersichtlichkeit und Inhalt dieses Handbuchs. Wenn Sie Fehler im Handbuch entdecken notieren Sie sich bitte die Seitenzahl.

Vielen Dank für Ihre Hilfe.

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

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

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

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

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

Kommentar: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Email an: support@quancom.de

Fax an: +49 2236 89 92 49

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

## 5.4 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 \_\_\_\_\_

## 5.5 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.