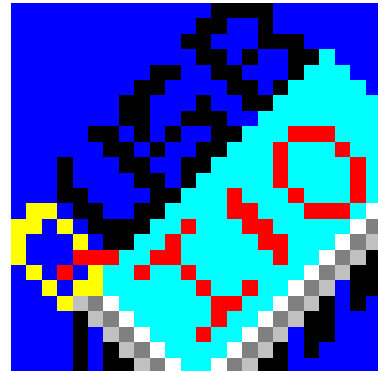




Komponente QPortUSB für Delphi 2006



INHALTSVERZEICHNIS

Komponente.....	1
QPortUSB	1
für Delphi 2006.....	1
Allgemeines	
Nutzung und Gewährleistung.....	5
Installation.....	6
Start vom Setup.....	6
Zielordner festlegen.....	7
Startmenü auswählen.....	8
Installation durchführen.....	9
Installiere	10
Fertige Installation.....	11
Komponente QPortUSB.....	12
Private - Funktion.....	13
Protected - Funktion.....	14
Port - Routinen.....	14
Die Eigenschaft-Property Port, WPort und DPort.....	14
Port einlesen.....	14
Port schreiben	14
Public - Funktion.....	15
Die Methode erzeugt und initialisiert ein QPortUSB Objekt.....	15
Die Methode Destroy entledigt sich der Komponente und der weiteren	
Komponenten, die zu dieser gehören.....	15
Die Methode SetName legt den Wert der Eigenschaft Name fest.....	15
Loaded ermöglicht einer Komponente, sich selbst zu initialisieren,	
nachdem alle ihre Teile aus	
einem Stream geladen wurden.....	15
Eigenschafts - Routinen.....	15
USB Karte aktivieren.....	15
USB Port Karte prüfen.....	15
Byte - Port X.....	15
Word - Port X.....	15
Dword - Port X.....	15
Hilfsfunktion - Fehlermeldung.....	15
Fehlermeldung per Dialogbox ausgeben.....	15
Property-Eigenschaft für den jeweiligen Port. Alle internen Routinen	
sind Public, wie oben schon erwähnt.....	16
Byte - Port X.....	16
Word - Port X.....	16
Dword - Port X.....	16



<u>Published - Funktion.....</u>	<u>17</u>
<u>Info über die Komponente.....</u>	<u>17</u>
<u>Sprache für Fehlermeldung (z.Zt. Deutsch, Englisch).....</u>	<u>17</u>
<u>USB Karte aktivieren, die Karten-Nr. Muss vorher festgelegt werden..</u>	<u>17</u>
<u>USB - Karten ermitteln, entweder als User oder Auto.....</u>	<u>17</u>
<u>Portadresse festlegen.....</u>	<u>17</u>
<u>Port Ausrichtung festlegen (z.Zt. Alles Bit als Eingabe oder Ausgabe)..</u>	<u>17</u>
<u>Test der USB Karte durchführen.....</u>	<u>17</u>
<u>USB Process (Wenn TURE, dann muss man halt warten).....</u>	<u>17</u>
<u>Komponenten - Fehler.....</u>	<u>18</u>
<u>NT kennt halt noch keine USB - Geräte.....</u>	<u>18</u>
<u>Der Treiber konnte nicht gestartet werden. Bitte überprüfen sie die</u>	
<u>USBportactive Variable.....</u>	<u>18</u>
<u>Der Treiber konnte nicht gestartet werden, da diese sich nicht im</u>	
<u>Aktuellen Verzeichnis befindet.....</u>	<u>18</u>
<u>Beispielanwendung.....</u>	<u>19</u>



Microsoft, Windows, Windows NT, Windows 2000, Windows XP sind Marken oder eingetragene Marken der Microsoft Corporation.

Delphi 5.0, BDS 2006 sind Produkte der Inprise Corporation.

USB -Treiber sind Produkte der Quamcom.de.

Inno-Setup sind Produkte von © 1997-2005 Jordan Russell.

All rights reserved. Portions Copyright © 2000-2005 Martijn Laan.

Die Nennung weitere Namen erfolgt in diesem Handbuch in der Regel ohne Erwähnung bestehender Patente, Gebrauchsmuster oder Warenzeichen. Das Fehlen eines entsprechenden Vermerks begründet nicht die Annahme, die Namen seien frei nutzbar. Alle Warenzeichen werden anerkannt.



Allgemeines

Nutzung und Gewährleistung

Endbenutzer-Lizenzvertrag für die Digital – Analyser Anwendung

1. Installation und Verwendung der Software

Sie sind berechtigt, eine Kopie der SOFTWARE auf einem oder mehreren Computern zu installieren, zu nutzen, darauf zuzugreifen, sie auszuführen oder anderweitig damit zu interagieren ("Ausführen"). Die SOFTWARE darf gleichzeitig auf oder von verschiedenen Computern installiert, angezeigt, ausgeführt, gemeinsam genutzt oder verwendet werden. Dieses Recht bezieht sich auf den privaten und kommerziellen Verwendungszweck der SOFTWARE.

2. Ausschluss von zufälligen, Folge- und bestimmten anderen Schäden

Im größtmöglichen durch das anwendbare Recht gestatteten Umfang ist der Autor in keinem Fall haftbar für irgendwelche speziellen, zufälligen oder Folgeschäden welcher Art auch immer (einschließlich, aber nicht beschränkt auf Schäden aus entgangenem Gewinn, Geschäftsunterbrechung, Personenschäden, Verlust der Privatsphäre, Verletzung von Vertragspflichten einschließlich Pflichten nach Treu und Glauben oder Sorgfaltspflichten, Fahrlässigkeit sowie Vermögens- oder sonstige Schäden), die aus der Verwendung der SOFTWARE oder aus der Tatsache, dass die SOFTWARE nicht verwendet werden können, resultieren oder in irgendeinem Zusammenhang damit stehen, selbst wenn der Autor auf die Möglichkeit solcher Schäden hingewiesen wurde.

4. Urheberrecht

Alle Eigentumsrechte und gewerblichen Schutzrechte an der SOFTWARE liegen beim Autor.

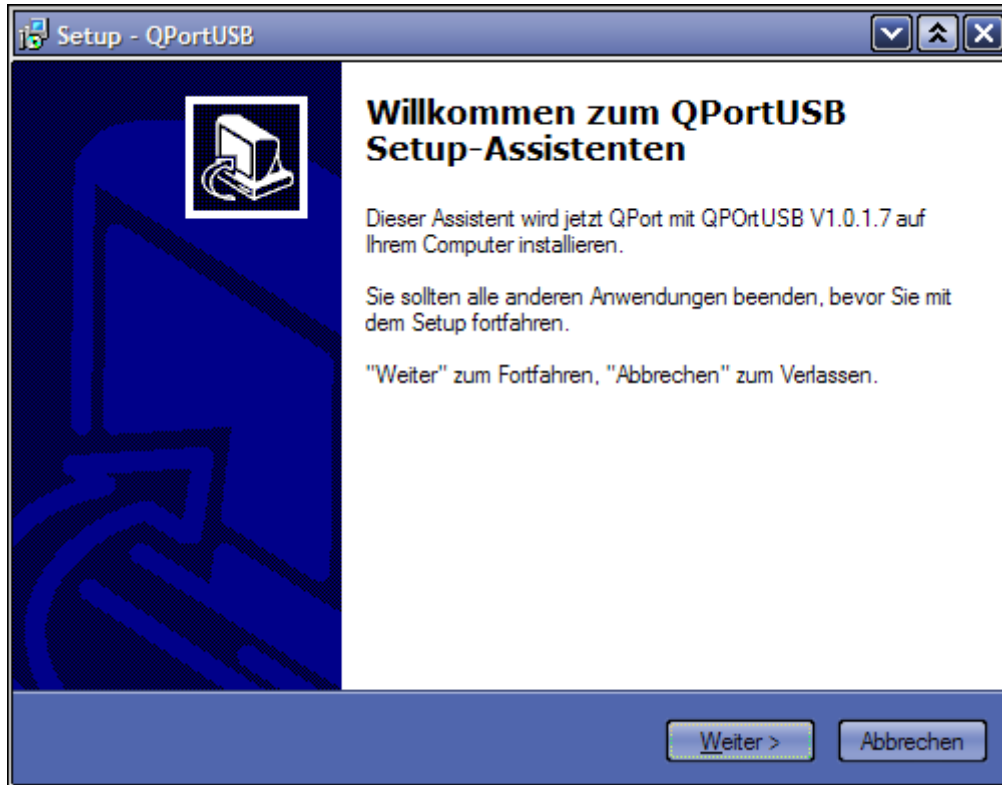
5. Trennung von Komponenten

Die SOFTWARE wird als einheitliches Produkt lizenziert für hmk-and-software.de und Sie sind nicht berechtigt, ihre Komponenten voneinander zu trennen.

Installation

Start vom Setup

Die Installation vom Digital Analyserprogramm erfolgt jetzt über ein Setup - Assistenten. Die Anwendung heißt `Analyser_setup.exe`. Diese muss von Ihnen mit dem Explorer gestartet werden. Nach dem Start dieser Anwendung sehen Sie folgende Dialogbox.



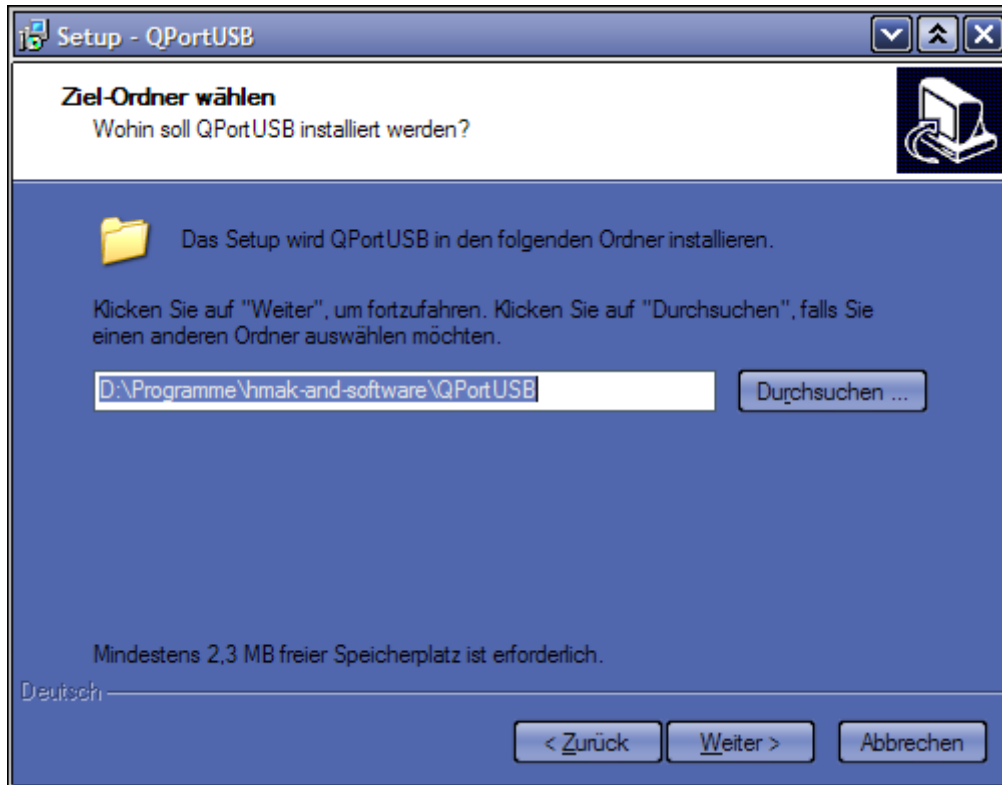
Bitte betätigen Sie den Weiter - Button, um das Setup - Programm fort zu führen.



Zielordner festlegen

Nach dem betätigen vom Weiter - Button können Sie jetzt festlegen, wo die Anwendung Digital Analyser installiert werden soll. Sie haben hier die Möglichkeit das Laufwerk oder die Verzeichnisstruktur zu ändern.

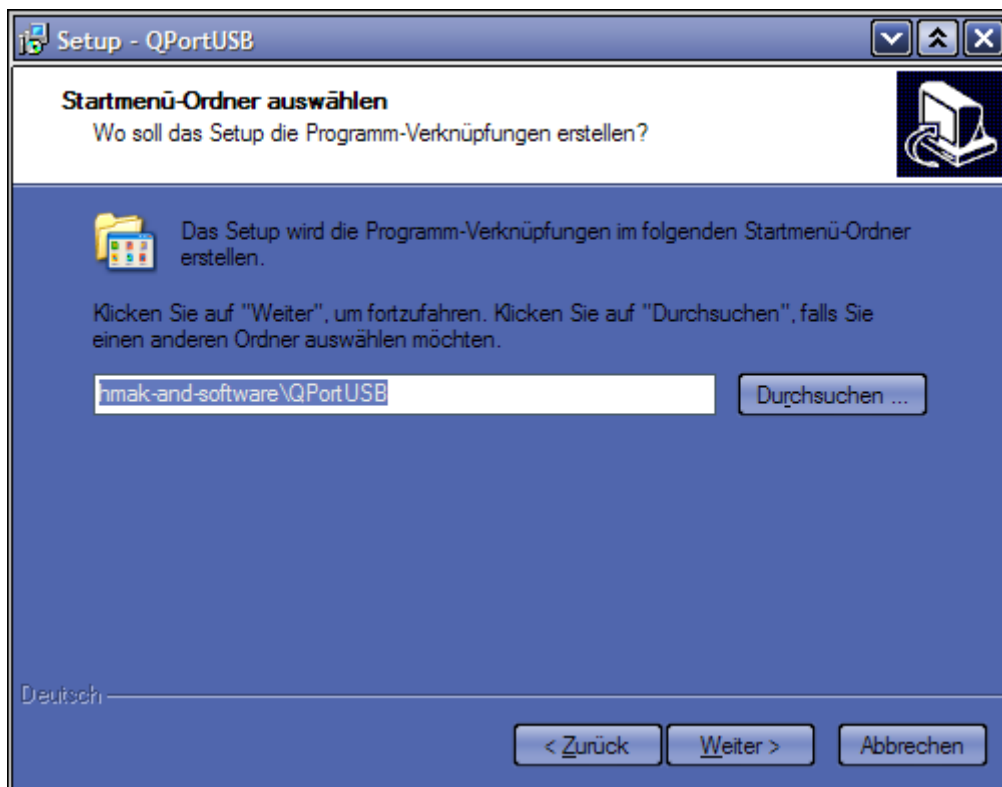
Weiter wird der benötigte Speicherplatz für die Anwendung angezeigt.



Bitte betätigen Sie den Weiter - Button, um das Setup - Programm fort zu führen.

Startmenü auswählen

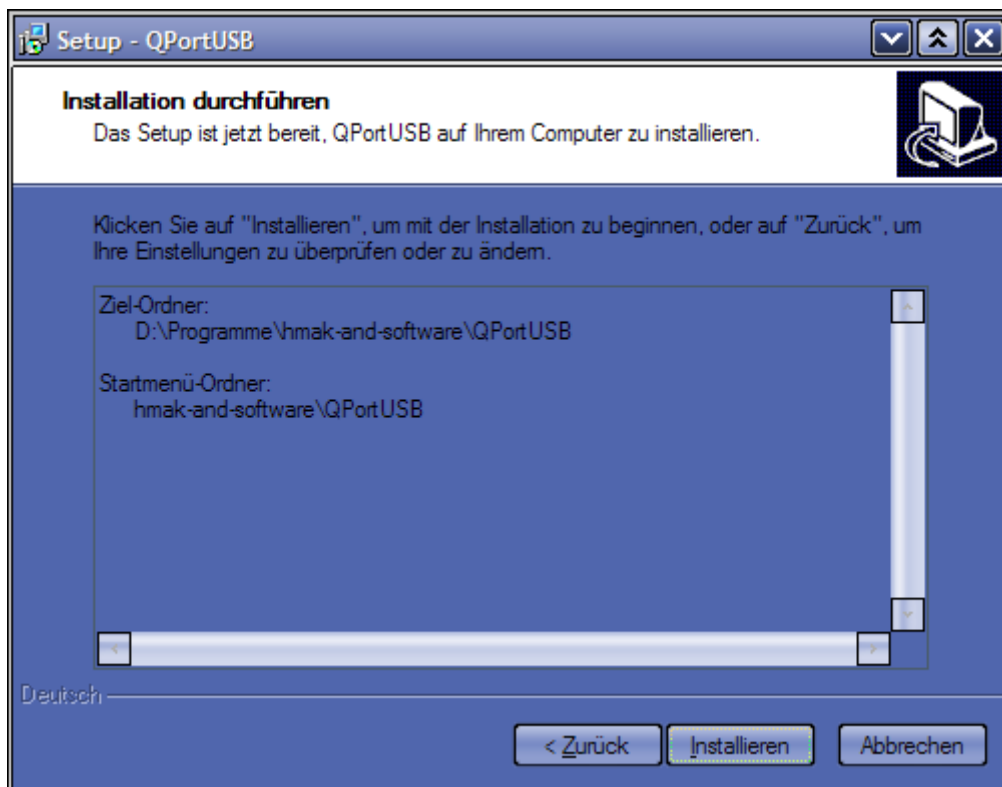
Nach nochmaligen betätigen vom Weiter - Button, können Sie jetzt das Startmenü festlegen oder aber Sie lassen die Standardeinstellung.



Bitte betätigen Sie den Weiter - Button, um das Setup - Programm fort zu führen.

Installation durchführen

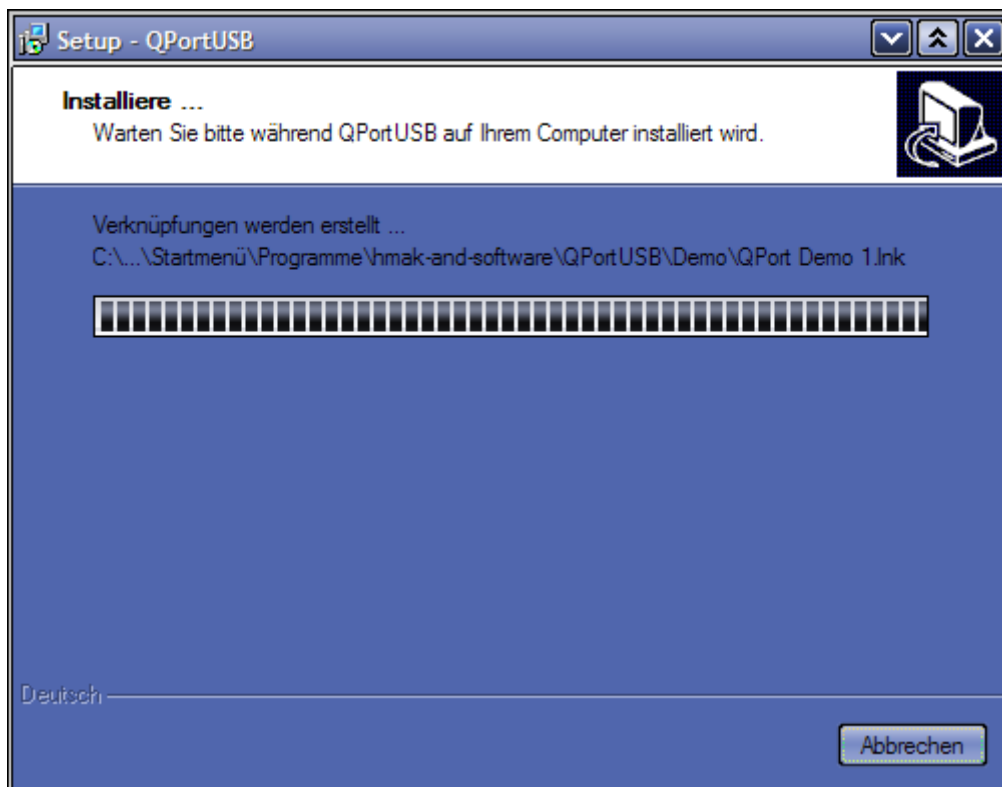
Haben Sie alle notwendigen Anpassung für die Installation durchgeführt, so zeigt das Setup noch einmal, alle Änderungen in einer Übersicht an.



Bitte betätigen Sie den Weiter - Button, um die Installation jetzt durch zu führen.

Installiere ...

Die Installation wird jetzt durchgeführt und die Progress bar zeigt den Fortschritt der Installation an.



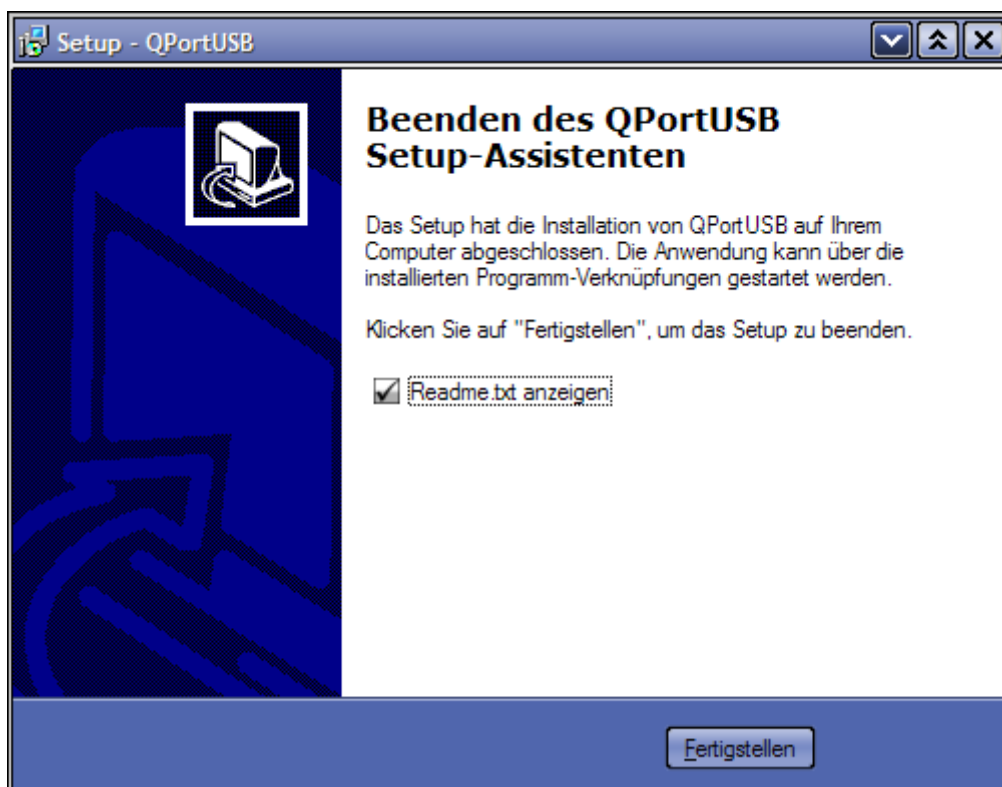
Hier haben Sie noch die Möglichkeit, die Installation abzubrechen. Sobald die Installation abgeschlossen ist, erhalten Sie ein weiteres Dialogfenster.

Fertige Installation

Wenn die Installation abgeschlossen ist, erhalten Sie folgende Dialogbox.

Bitte lesen Sie die WICHTIG.txt und Liesmich.txt, da sonst keine Funktionen von der Komponente QPortUSB genutzt werden können.

Bitte betätigen Sie den Fertigstellen - Button, um die Installation abzuschließen. Sie haben jetzt alle notwendigen Programmteile für die Installation ausgeführt.



Viel Spaß bei der Verwendung vom QPortUSB



Komponente QPortUSB

Die Komponente QPortUSB ist von mir für Delphi 2006 umgesetzt worden. Der Quellcode kann aber auch für Delphi 5.0 und co. Genutzt werden. Die passenden Treiber werden von in der Demo - Installation mit eingebunden.

Die neusten Treiber findet man bei Quancom.de. Ein Komponente existiert hierfür bis heute nicht. Die Delphi Komponente kapselt den QLIB.pas mit meiner Komponente

ACHTUNG es gibt jetzt unterschiedliche Treiber für VISTA, XP und Windows 98. Von mir wird die Version 1.98s genutzt. Diese funktioniert mit folgenden Betriebssystemen Windows 98, 2000 und XP. Es muss nichts weiter gemacht werden, außer das die Installation ausgeführt wird. Hier bei werden dann die unterschiedlichen Treiber installiert.

Weiter müssen Sie die Komponente in das jeweilige Delphi System einbinden und übersetzen.

Die meisten Funktionen sind in der Protected Struktur versteckt und sollten auch nicht sichtbar gemacht werden durch eine Ableitung. Man muss hier eigentlich nichts weitere ändern.

Der Aufbau von der Komponente wird in den noch folgenden Seiten erklärt.



Private - Funktion

Wie das Wort Private schon ausdrückt, sind hier alle Variablen hinterlegt, die man tunlichst nicht direkt ändern sollte.

FQAbout	: string;	Info über die Komponente
FQUSBLanguage	: TQUSBLanguage;	Sprache bei Fehlermeldungen
FCmpName	: string;	Interner Komponenten Name
FErrtxt	: string;	Art der Fehlermeldung
Fmsgtxt	: string;	Fehlermeldung
FErrNumber	: integer;	Fehlernummer
FQUSBportactive	: boolean;	
FQUSBarray	: array[0..7] of TQUSBstruct;	Struktur der QPortUSB - Karten
FQUSBportConnect	: TQUSBConnect;	Verbindung zur USB - Karte aktivieren
FQUSBportAddr	: integer;	USB Karten Nummer
FQUSBportTest	: boolean;	Selbsttest von USB- Karte und Komponente
FQUSBprocess	: boolean;	Komponente zeigt an, ob der Selbsttest Prozess noch läuft.
FQUSBstopProcess	: boolean;	Anwender erzwingt den Abbruch vom Selbsttest. Hier durch kann es schon passieren, dass man die USB- Karte einmal schließen und neu aktivieren muss.



Protected - Funktion

Port - Routinen

Eigentliche Routinen, für den QportUSB - Treiber ansprechen

Digit USB - Port Kommando

```
function    GetQUSBcommand :TQUSBcommand;
procedure   SetQUSBcommand( avalue: TQUSBcommand );
```

Digi USB - Port einlesen

```
function    GetQPort :byte;
function    GetQWPort:word;
function    GetQDPort:dword;
```

Digi USB - Port schreiben

```
procedure   SetQPort (Data:byte);
procedure   SetQWPort(Data:word);
procedure   SetQDPort(Data:dword);
```

Fehlermeldung

```
property    ErrorNumber : integer;
```

Die Eigenschaft-Property Port, WPort und DPort

Port einlesen

```
function GetQPort      :byte;
function GetQWPort     :word;
function GetQDPort     :dword;
```

Port schreiben

```
procedure SetQPort      (Value: dword; Data:byte);
procedure SetQWPort     (Value: dword; Data:word);
procedure SetQDPort     (Value: dword; Data:dword);
```

Ein direkter Zugriff auf die Routinen nicht Empfehlenswert, da weitere interne Änderungen noch durch geführt werden müssen.



Public - Funktion

Hier sind alle Funktionen und Variablen die für die Öffentlichkeit zugänglich sind.

Die Methode erzeugt und initialisiert ein QPortUSB Objekt.

constructor Create(AOwner: TComponent); override;

Die Methode Destroy entledigt sich der Komponente und der weiteren Komponenten, die zu dieser gehören.

destructor Destroy; override;

Die Methode SetName legt den Wert der Eigenschaft Name fest.

procedure SetName(const NewName: TComponentName); override;

Loaded ermöglicht einer Komponente, sich selbst zu initialisieren, nachdem alle ihre Teile aus einem Stream geladen wurden.

procedure Loaded; override;

Eigenschafts - Routinen

USB Karte aktivieren

procedure SetQUSBportactive(avalue : boolean);
function GetQUSBportactive: boolean;

USB Port Karte prüfen

procedure SetQUSBportTest(avalue : boolean);
procedure StopQUSBprocess;

Byte - Port X

property BPort: byte
 read GetQPort write SetQPort;

Word - Port X

property WPort : word
 read GetQWPort write SetQWPort;

Dword - Port X

property DPort : dword
 read GetQDPort write SetQDPort;

Hilfsfunktion - Fehlermeldung

function QUSBCheckactive : boolean;
function QUSBCheckerror : boolean;
function QUSBReaderrnumber : integer;
function QUSBReaderrorstr (avalue : integer; svalue : string): string;

Fehlermeldung per Dialogbox ausgeben

procedure SetQerrmessage (Value : integer; Svalue : string; Flags: integer);



Property-Eigenschaft für den jeweiligen Port. Alle internen Routinen sind Public, wie oben schon erwähnt.

Byte - Port X

```
property BPort[Value: dword]: byte read GetQPort write SetQPort;
```

Word - Port X

```
property WPort[Value: dword] : word read GetQWPort write SetQWPort;
```

Dword - Port X

```
property DPort[Value: dword] : dword read GetQDPort write SetQDPort;
```



Published - Funktion

Endlich kommt der Objektinspektor zum Zug. Alles was sichtbar sein soll, in der Komponenten wird hier angezeigt.

Info über die Komponente

property About : string read Fabout;

Sprache für Fehlermeldung (z.Zt. Deutsch, Englisch)

property USBLanguage : TQUSBLanguage
read FQUSBLanguage write FQUSBLanguage;

USB Karte aktivieren, die Karten-Nr. Muss vorher festgelegt werden.

property USBportactive : boolean
read GetQUSBportactive write SetQUSBportactive;

USB - Karten ermitteln, entweder als User oder Auto

property USBportconnect : TQUSBConnect
read FQUSBportConnect write FQUSBportConnect default auto;

Portadresse festlegen

property USBportAddr : integer
read FQUSBportAddr write SetQUSBportAddr;

Port Ausrichtung festlegen (z.Zt. Alles Bit als Eingabe oder Ausgabe)

property USBPortCMD : TQUSBcommand
read GetQUSBcommand write SetQUSBcommand;

Test der USB Karte durchführen

property USBportTest : boolean
read FQUSBportTest write SetQUSBportTest;

USB Process (Wenn TURE, dann muss man halt warten)

property USBprocess : boolean
read fQusbprocess;



Komponenten - Fehler

Folgende Fehlerarten können bei der Verwendung der Komponente auftreten.

Fehlernummer

01. System konnte die USB - Karte nicht starten. Bitte überprüfen Sie deshalb ihr USB-Kabel oder die QUSB-TLL-24 Karte.

02. Der Treiber ist nicht erlaubt für dieses Windows-System:
Windows NT 3.51 oder NT 4.0. Bitte überprüfen Sie ihr Windows-System.

NT kennt halt noch keine USB - Geräte.

03. System konnte die USB - Karte nicht starten. Bitte überprüfen Sie in der Komponente, ob die USBportactive Variable gesetzt wurde!

Der Treiber konnte nicht gestartet werden. Bitte überprüfen sie die USBportactive Variable.

04. Leider fehlt die QUSB.DLL Sie befindet sich nicht im aktuellen System32 - Verzeichnis.
Bitte überprüfen Sie ihr System.

Der Treiber konnte nicht gestartet werden, da diese sich nicht im Aktuellen Verzeichnis befindet.

05. Selbsttest von der USB-Karte: XX wurde gestartet. Wenn hier ein Fehler auftritt, so überprüfen Sie bitte die USB-Karte.
ACHTUNG
Die Ports werden auf Ausgang geschaltet!!!

06. ACHTUNG:, der Selbsttest prüft jetzt die INPUT-Ports. Legen Sie bitte die Leitung Bit XX auf 0 Volt, an ihrer USB-Karte

07. Der Selbsttest wird gerade geschlossen, etwas Geduld bitte. Wenn hier ein Fehler auftritt, so überprüfen Sie bitte die USB-Karte.

08. Der USB Port ist leider nicht offen. Bitte überprüfen Sie ihre Parameter.

09. Bitte warten, da gerade der USB-Treiber die Karte initialisiert. Zum deaktivieren dieser Funktion, siehe Systemvariable USBProcess.

10. Keine Function.

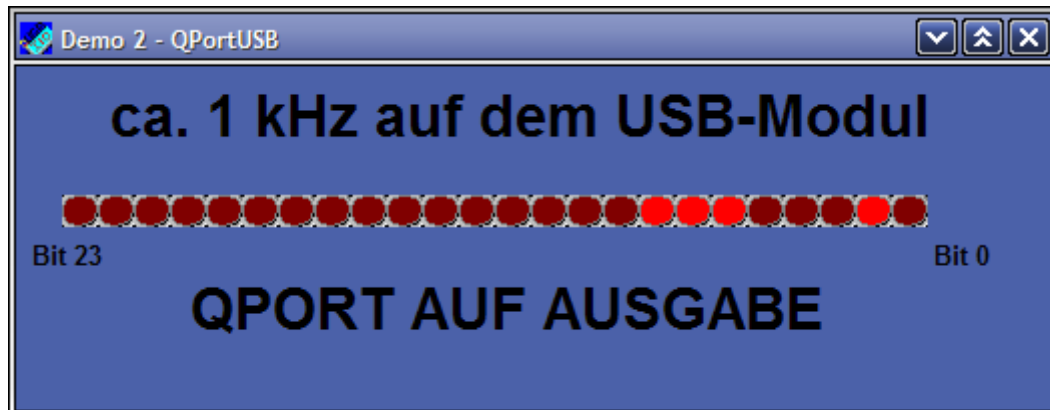
11. Die QPortUSB Komponente hat den USB- Treiber oder die USB- Karte nicht gefunden.
Bitte überprüfen Sie ihr Windows-System.

12. ACHTUNG:, der Selbsttest prüft jetzt die OUTPUT-Ports.
Bitte prüfen Sie die Leitungen Bit 0 - Bit 32 auf 5 Volt an ihrer USB-Karte XX.
ACHTUNG
Ein Wert von 3.6x Volt wäre in Ordnung.

13. ACHTUNG:, der Selbsttest prüft jetzt die OUTPUT-Ports.
Bitte prüfen Sie die Leitungen Bit 0 - Bit 32 auf 0 Volt an ihrer USB-Karte XX.
ACHTUNG
Ein Wert von 0.7x Volt wäre in Ordnung.



Beispielanwendung



```
unit Demo_2;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, ExtCtrls, LED, QPortUSB, StdCtrls;
```

```
type
```

```
TForm2 = class(TForm)  
  QPortUSB1: TQPortUSB;  
  Led1: TLed;  
  Timer1: TTimer;  
  Label1: TLabel;  
  Label2: TLabel;  
  Label3: TLabel;  
  Label4: TLabel;  
  procedure Timer1Timer(Sender: TObject);  
  procedure FormCreate(Sender: TObject);  
  procedure FormDestroy(Sender: TObject);  
private  
  { Private-Deklarationen }  
public  
  { Public-Deklarationen }  
end;
```

```
var
```

```
Form2: TForm2;  
counter : integer;
```

```
implementation
```

```
{ $R *.dfm }
```



```
procedure TForm2.FormCreate(Sender: TObject);
begin
    counter := 0;
    QPortUSB1.USBportactive := true;
    if QPortUSB1.USBportactive then
    begin
        QportUSB1.USBPortAllCMD := qoutput;
        Timer1.Enabled := true;
    end;
end;

procedure TForm2.FormDestroy(Sender: TObject);
begin
    Timer1.Enabled := false;
end;

procedure TForm2.Timer1Timer(Sender: TObject);
begin
    Led1.LEDHexValue := counter;
    qportusb1.DPort := counter;
    inc( counter, 1 );
end;
end.
```