

Handbuch

USB Treiber-Installation



Release

1.1, Oktober 2004

© 10/2004 by Wiesemann & Theis GmbH

Microsoft und Windows sind eingetragene Warenzeichen der Microsoft Corporation

Irrtum und Änderung vorbehalten:

Da wir Fehler machen können, darf keine unserer Aussagen ungeprüft verwendet werden. Bitte melden Sie uns alle Ihnen bekannt gewordenen Irrtümer oder Mißverständlichkeiten, damit wir diese so schnell wie möglich erkennen und beseitigen können.

Führen Sie Arbeiten an bzw. mit W&T Produkten nur aus, wenn Sie hier beschrieben sind und Sie die Anleitung vollständig gelesen und verstanden haben. Eigenmächtiges Handeln kann Gefahren verursachen. Wir haften nicht für die Folgen eigenmächtigen Handelns. Fragen Sie im Zweifel lieber noch einmal bei uns bzw. Ihrem Händler nach!

W&T

Die seriellen W&T USB-Interfaces sind, außer unter Linux, unter den verschiedenen Betriebssystemen nur mittels spezieller Treiber ansprechbar. Diese Treiber unterliegen sowohl hinsichtlich ihrer technischen Features, als auch hinsichtlich der Zahl und Art der unterstützten Betriebssysteme einer ständigen Weiterentwicklung.

Aus diesem Grunde stellt W&T die aktuellen Treiber und Softwareinstallations-Anleitungen auf den Datenblattseiten der USB-Interfaces im Internet unter <http://www.wut.de> zur Verfügung.

Während Linux ab Kernel-Version 2.4 die Interfaces direkt ohne zusätzliche Treiber unterstützt, installiert der Treiber auf Windows 98/ME, Windows 2000 und XP - Systemen virtuelle COM-Schnittstellen, über die auf die seriellen Schnittstellen der Interfaces zugegriffen werden kann.

Detaillierte Informationen zur Installation und zur Konfiguration der Treiber unter den verschiedenen Betriebssystemen finden Sie auf den folgenden Seiten.

Inhalt

Windows 98/ME

1. Installation	5
2. Deinstallation	10

Windows 2000

1. Installation	12
2. Deinstallation	20

Windows XP

1. Installation	22
2. Deinstallation	28

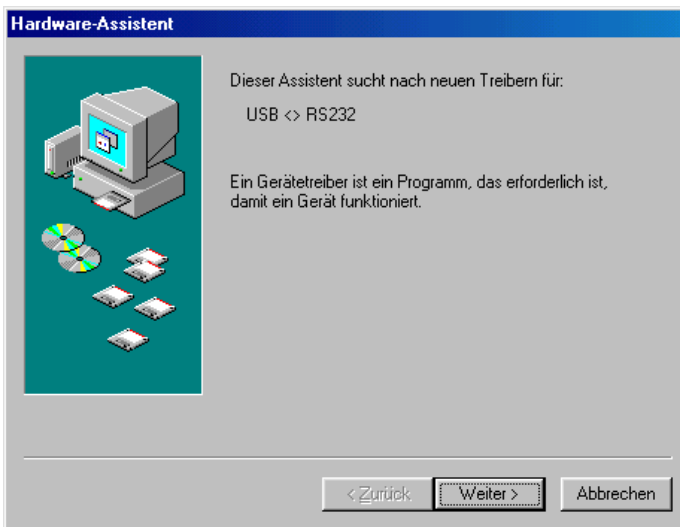
Technische Hintergrundinformationen

Datendurchsatz und Latenzzeit der W&T USB-Interfaces	30
--	----

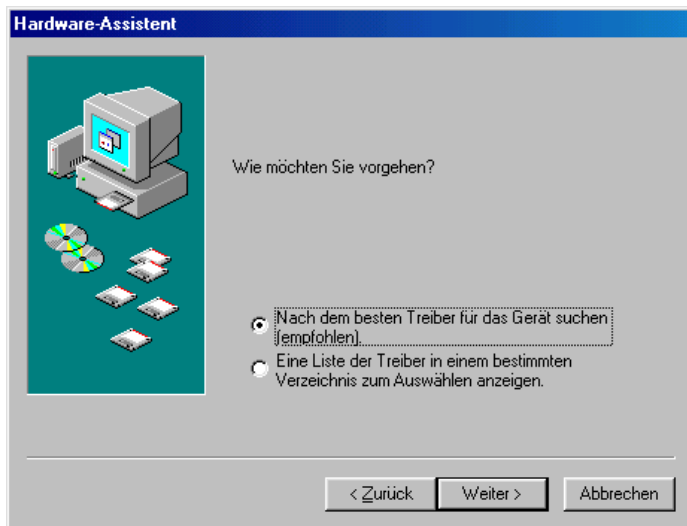
Windows 98/ME**1. Installation**

Die Installation der Treiber geschieht weitgehend automatisiert. Im Verlauf der Installation muss lediglich der Pfad angegeben werden, unter dem die Treiberdateien zu finden sind. In der Regel handelt es sich dabei um das CD-ROM-Laufwerk, wenn die auf der CD mitgelieferten Treiber installiert werden sollen.

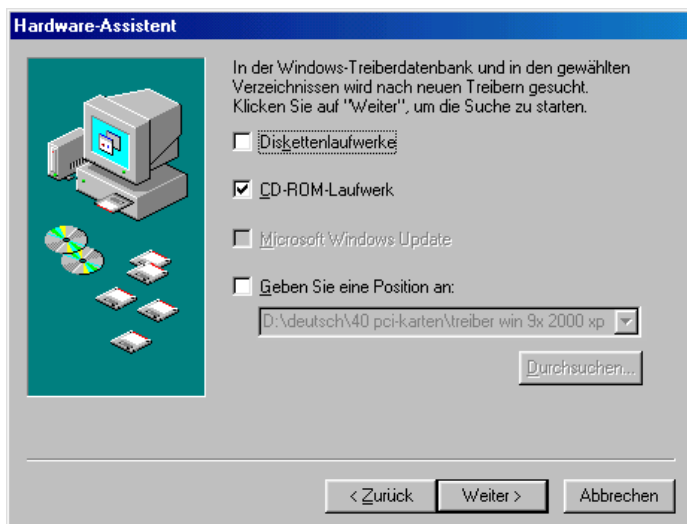
Nach erstmaligem Anschluß des USB-Interfaces an den PC meldet Windows mit dem folgenden Fenster, dass eine neue Hardware-Komponente gefunden wurde und für diese Komponente ein Treiber installiert wird.



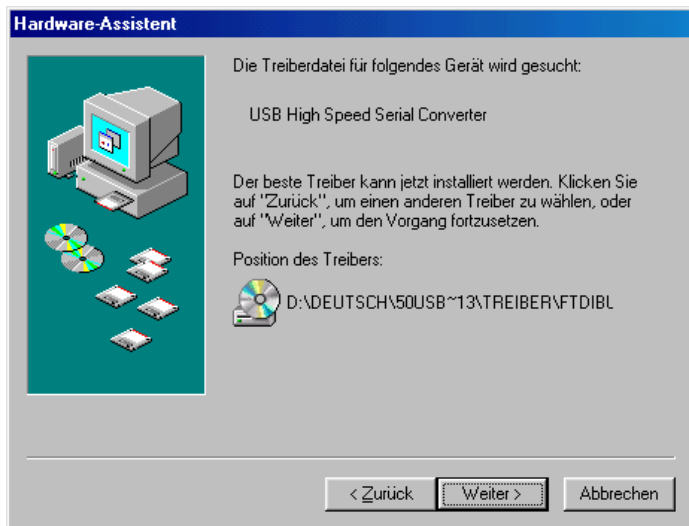
Bestätigen Sie die Treiber-Installation mit *Weiter* und lassen Sie Windows nach einem passenden Treiber für das angeschlossene Gerät suchen:



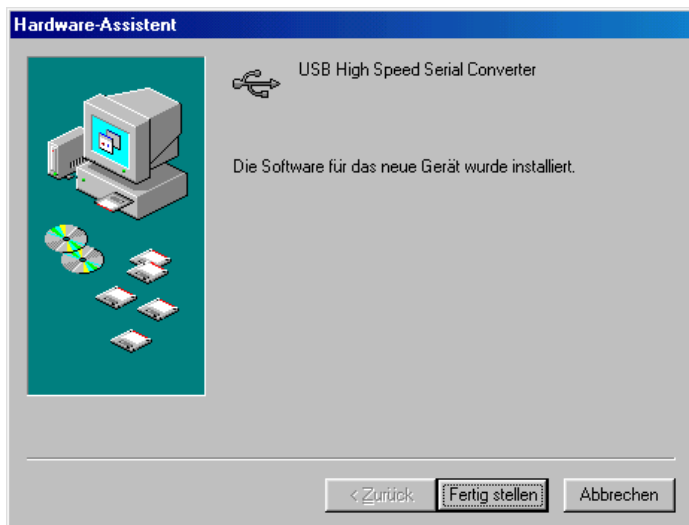
Geben Sie anschließend den Pfad an, unter dem Windows die Treiber findet:



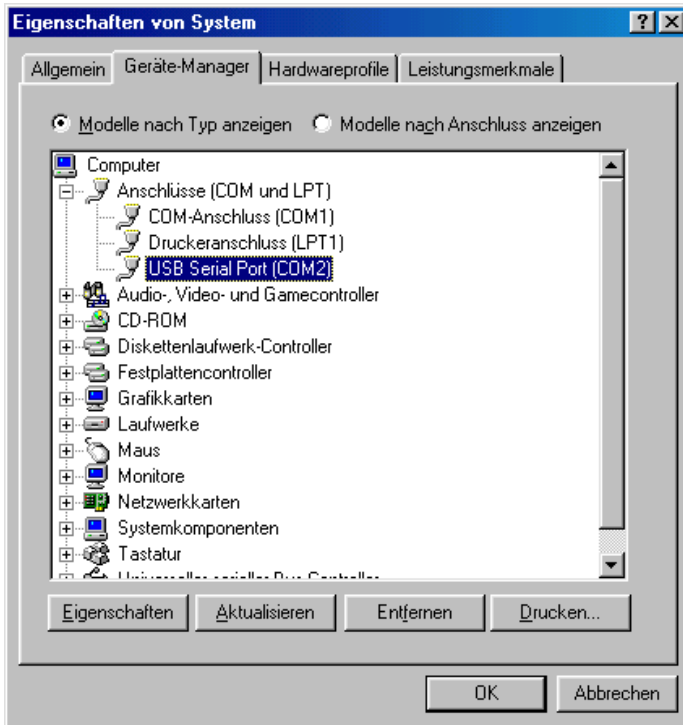
Windows bestätigt anschließend, daß die zum Gerät gehörende Datei *ftdibus.inf* gefunden wurde. Starten Sie die Installation des Treibers mit *Weiter*:



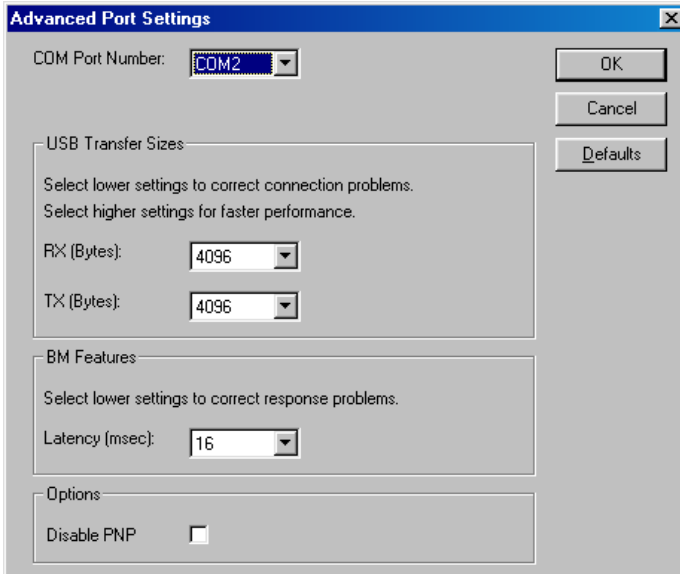
Windows kopiert die erforderlichen Dateien und fordert Sie auf, die Treiber-Installation mit *Fertig stellen* zu beenden:



Nach erfolgter Installation finden Sie unter *Start -> Einstellungen -> Systemsteuerung -> System -> Geräte-Manager -> Anschlüsse (COM und LPT)* einen Eintrag für das eingerichtete USB-Interface - im Beispiel unten als COM2:



Durch einen Doppelklick auf den Eintrag läßt sich das Interface konfigurieren, wobei die unter *Port Settings* gesetzten Werte in der Regel von der in der entsprechenden Anwendung gewählten Konfiguration überschrieben werden. Interessant sind in diesem Zusammenhang die unter *Advanced...* vorgesehenen Möglichkeiten, den dem Interface zugewiesenen COM-Port zu verändern und den Latency Timer anzupassen (siehe technische Hintergrundinformationen).



Warnhinweis:

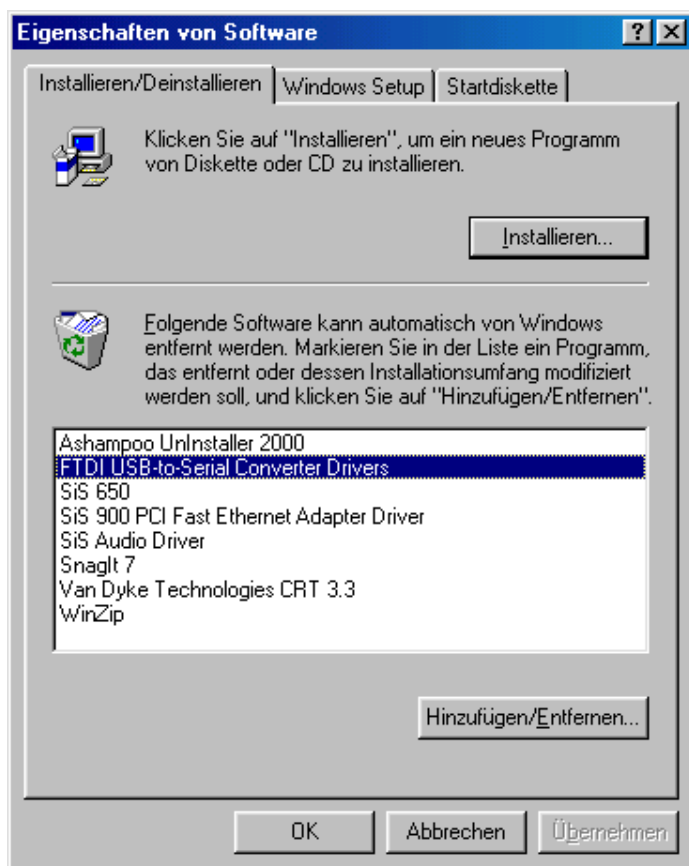
Es sollte unbedingt darauf geachtet werden, daß eine COM-Port-Nummer nicht mehrfach vergeben wird. Diese Möglichkeit besteht in Fällen, in denen nicht permanent alle installierten seriellen Devices im oder am PC in Betrieb sind. Dies ist zum Beispiel der Fall, wenn zusätzliche virtuelle COM-Schnittstellen über ein Netzwerk eingerichtet und zur Zeit nicht mit dem PC verbunden sind. Falls der zugehörige Treiber die Ressourcen wieder freigibt, wenn die Komponente nicht angeschlossen ist, besteht die Möglichkeit eine COM-Port-Nummer auszuwählen, die eigentlich schon vergeben ist und bei Wiederanschluß des Gerätes reaktiviert wird.



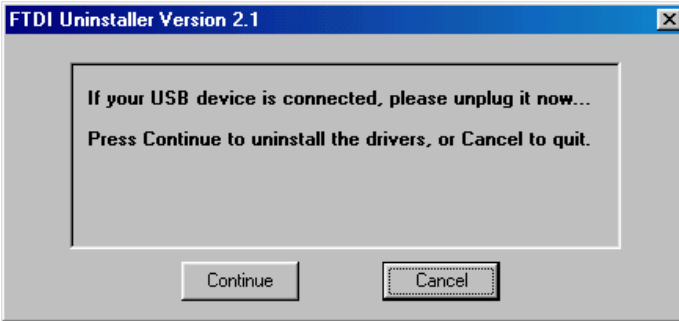
In diesem Fall existiert dann ein COM-Port mehrfach und Windows quittiert den Versuch, die Schnittstelle zu öffnen im günstigsten Fall mit einer Fehlermeldung.

2. Deinstallation

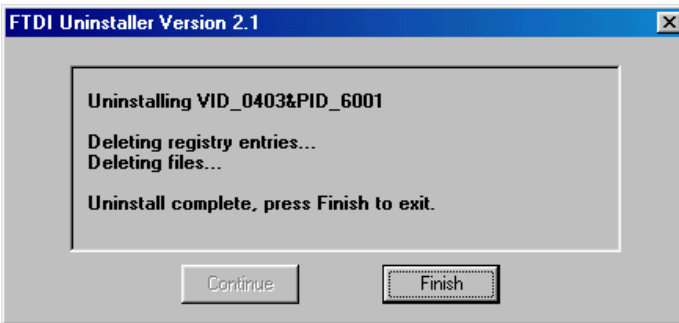
Die Deinstallation der Treibersoftware geschieht unter *Start -> Einstellungen -> Systemsteuerung -> Software*. Wählen Sie den *FTDI USB-to-Serial Converter Drivers* Punkt aus und bestätigen Sie die Auswahl mit *Hinzufügen/Entfernen...*



Falls das Interface noch nicht vom PC getrennt wurde, so sollten Sie dies vor Deinstallation des Treibers vornehmen und das folgende Fenster quittieren:



Bestätigen Sie im folgenden Fenster das Löschen der Treiberdateien:



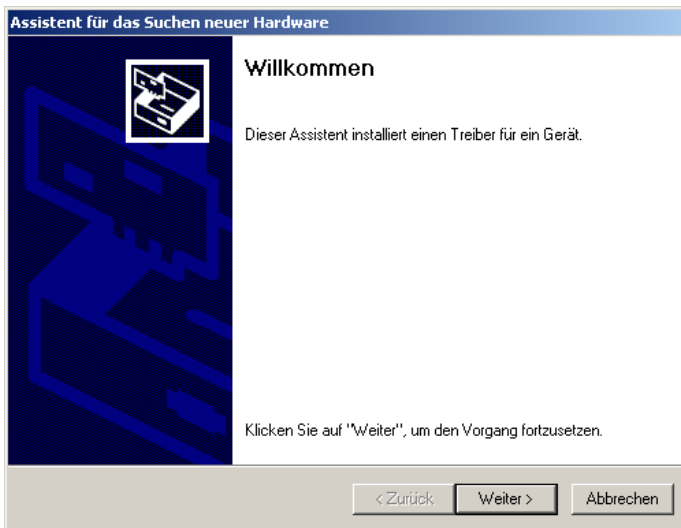
Damit ist die Deinstallation der Treiber beendet.

Windows 2000

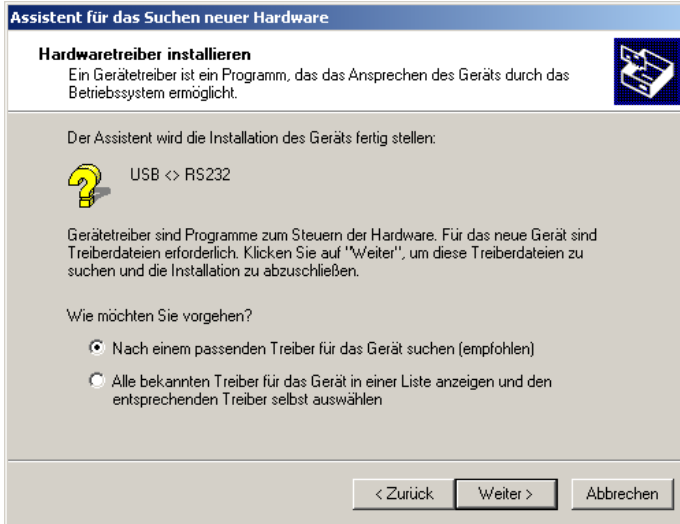
1. Installation

Die Installation der Treiber geschieht weitgehend automatisiert. Im Verlauf der Installation muß lediglich der Pfad angegeben werden, unter dem die Treiberdateien zu finden sind. In der Regel handelt es sich dabei um das CD-ROM-Laufwerk, wenn die auf der CD mitgelieferten Treiber installiert werden sollen.

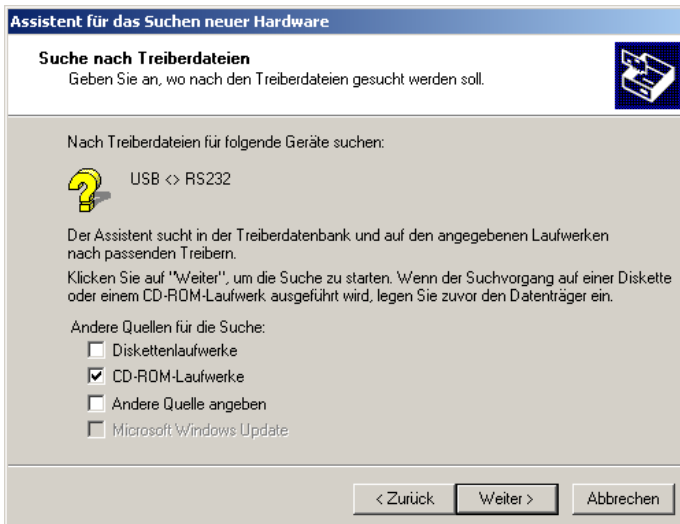
Nach erstmaligem Anschluß des USB-Interfaces an den PC meldet Windows mit dem folgenden Fenster, daß eine neue Hardware-Komponente gefunden wurde und für diese Komponente ein Treiber installiert wird.



Bestätigen Sie die Treiber-Installation mit *Weiter* und lassen Sie Windows nach einem passenden Treiber für das angeschlossene Gerät suchen:



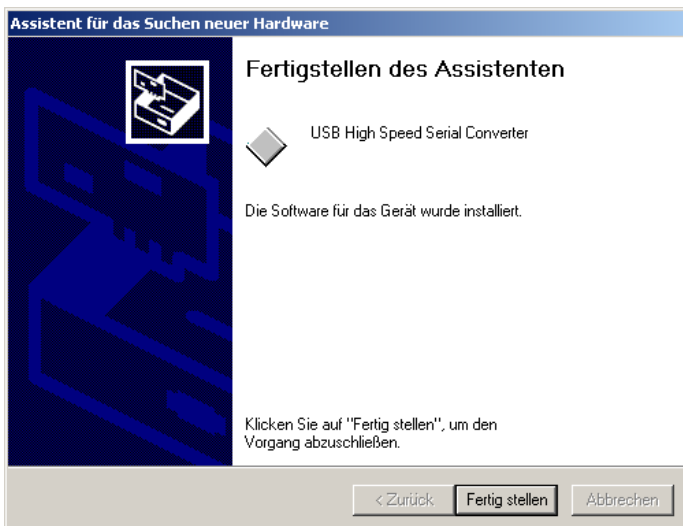
Geben Sie anschließend den Pfad an, unter dem Windows die Treiber findet:



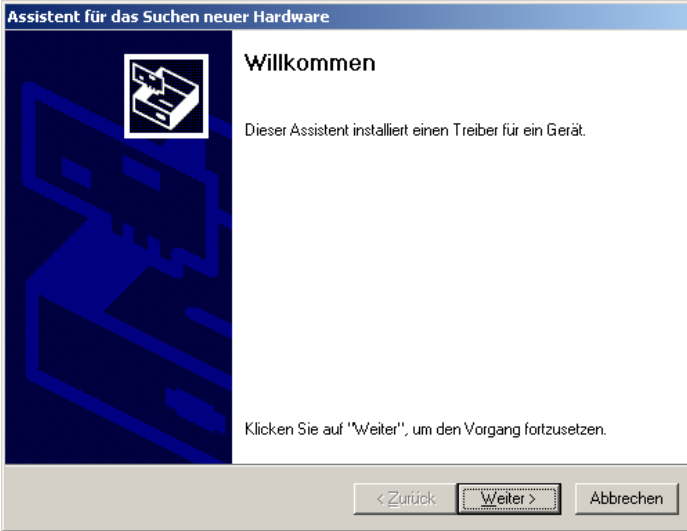
Windows bestätigt anschließend, daß die zum Gerät gehörende Datei *ftdibus.inf* gefunden wurde. Starten Sie die Installation des Treibers mit *Weiter*.



Windows kopiert die erforderlichen Dateien und fordert Sie auf, die Treiber-Installation mit *Fertig stellen* zu beenden:



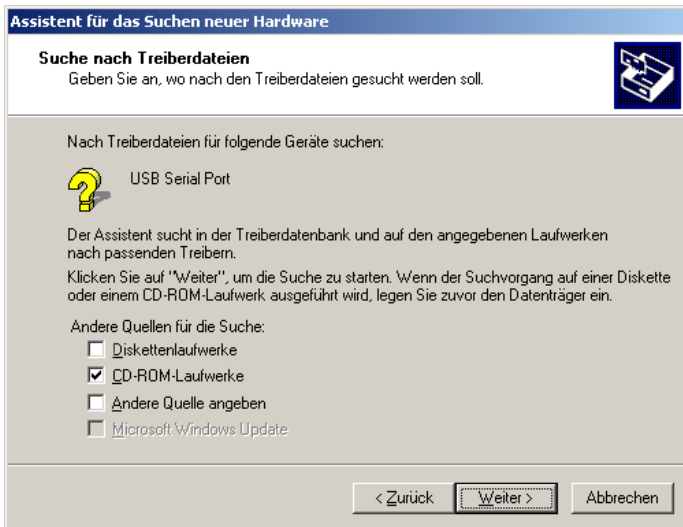
Nach Installation des Treibers ist zusätzlich noch die Installation der COM-Umlenkung erforderlich, die für das serielle Interface einen virtuellen COM-Port einrichtet. Dieser Vorgang wird automatisch nach Installation des Treibers gestartet und erfordert dieselben Eingaben, wie bei der Treiber-Installation.



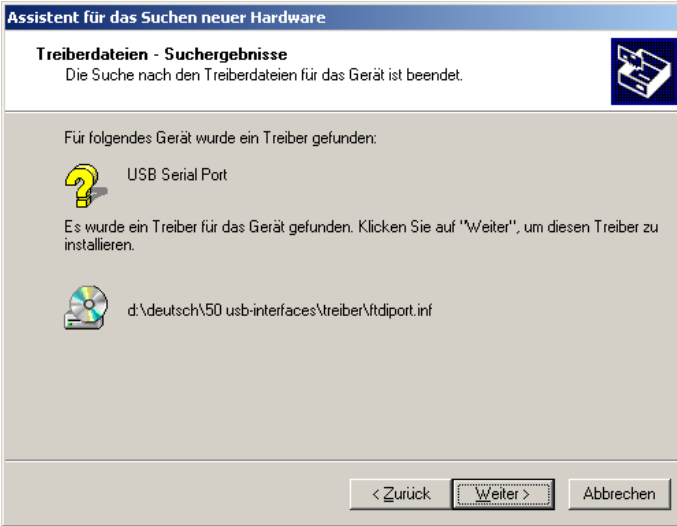
Lassen Sie Windows nach einem passenden Treiber für das angeschlossene Gerät suchen:



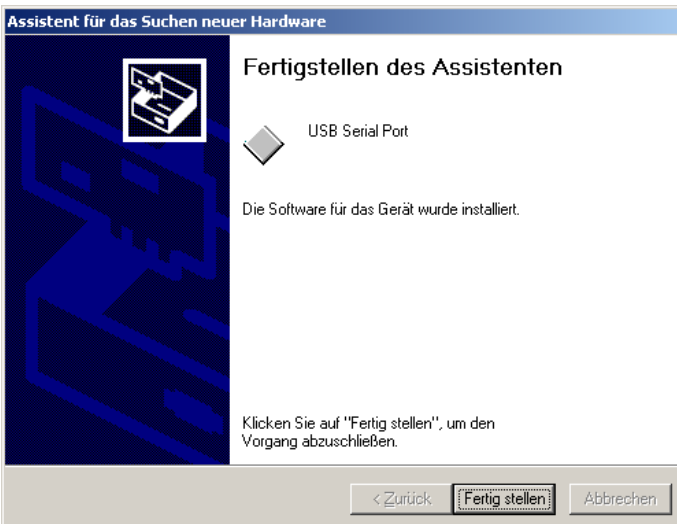
Geben Sie anschließend den Pfad an, unter dem Windows die Treiber findet:



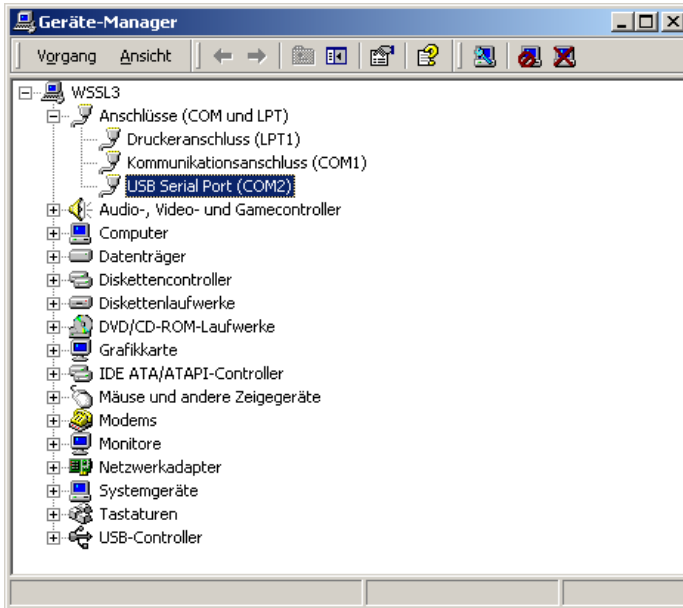
Windows bestätigt anschließend, daß die zum Gerät gehörende Datei *ftdiport.inf* gefunden wurde. Starten Sie die Installation des Treibers mit *Weiter*:



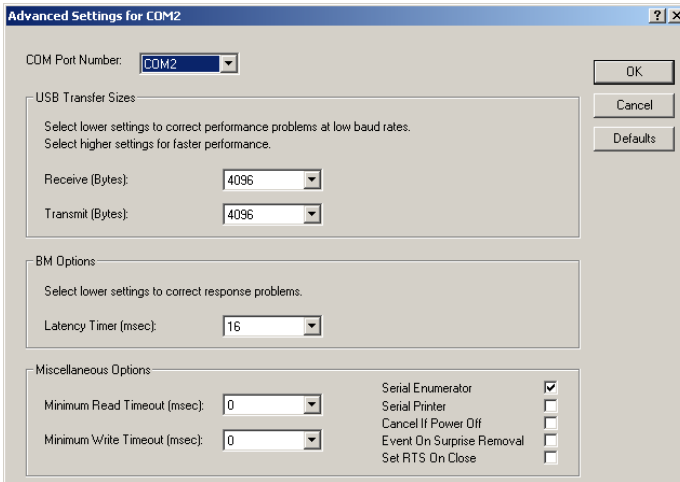
Windows fordert Sie anschließend wiederum auf, die Treiber-Installation mit *Fertig stellen* zu beenden:



Nach erfolgter Installation finden Sie unter *Start -> Einstellungen -> Systemsteuerung -> System -> Hardware -> Geräte-Manager -> Anschlüsse (COM und LPT)* einen Eintrag für das eingerichtete USB-Interface - im Beispiel unten als COM2:



Durch einen Doppelklick auf den Eintrag läßt sich das Interface konfigurieren, wobei die unter *Port Settings* gesetzten Werte in der Regel von der in der entsprechenden Anwendung gewählten Konfiguration überschrieben werden. Interessant sind in diesem Zusammenhang die unter *Advanced...* vorgesehenen Möglichkeiten, den dem Interface zugewiesenen COM-Port zu verändern und den Latency Timer anzupassen (siehe technische Hintergrundinformationen).



Warnhinweis:

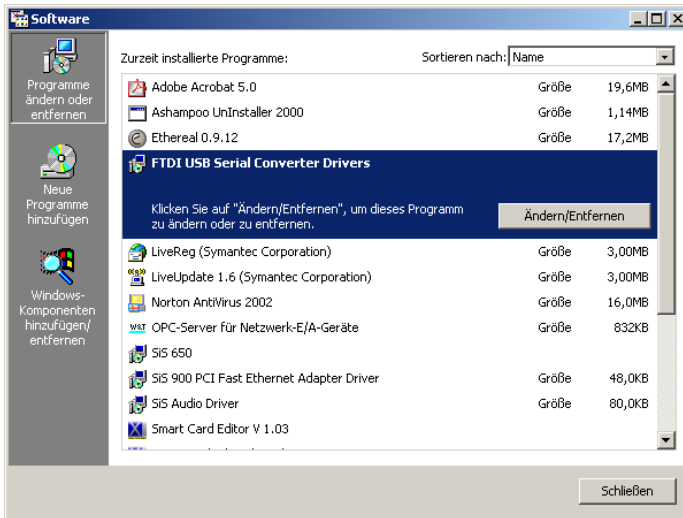
Es sollte unbedingt darauf geachtet werden, daß eine COM-Port-Nummer nicht mehrfach vergeben wird. Diese Möglichkeit besteht in Fällen, in denen nicht permanent alle installierten seriellen Devices im oder am PC in Betrieb sind. Dies ist zum Beispiel der Fall, wenn zusätzliche virtuelle COM-Schnittstellen über ein Netzwerk eingerichtet und zur Zeit nicht mit dem PC verbunden sind. Falls der zugehörige Treiber die Ressourcen wieder freigibt, wenn die Komponente nicht angeschlossen ist, besteht die Möglichkeit eine COM-Port-Nummer auszuwählen, die eigentlich schon vergeben ist und bei Wiederanschluß des Gerätes reaktiviert wird.



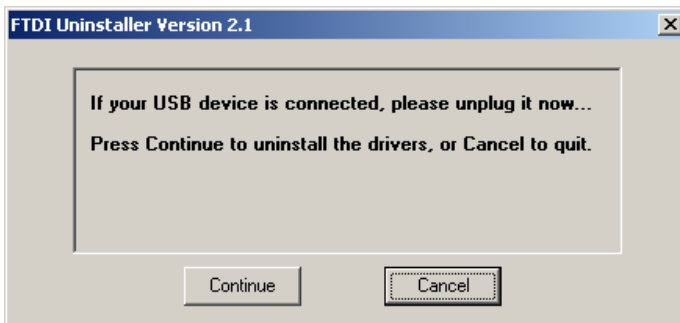
In diesem Fall existiert dann ein COM-Port mehrfach und Windows quittiert den Versuch, die Schnittstelle zu öffnen im günstigsten Fall mit einer Fehlermeldung.

2. Deinstallation

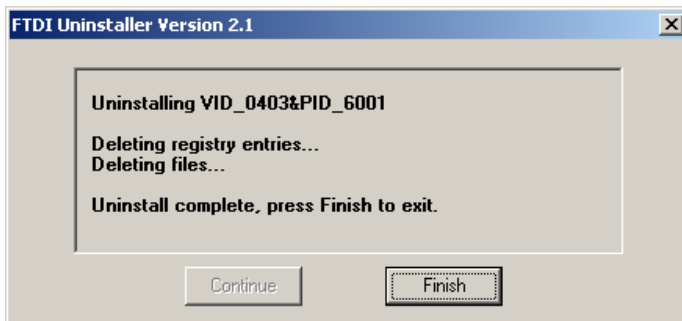
Die Deinstallation der Treibersoftware geschieht unter *Start -> Einstellungen -> Systemsteuerung -> Software*. Wählen Sie den *FTDI USB-to-Serial Converter Drivers* Punkt aus und bestätigen Sie die Auswahl mit *Ändern/Entfernen...*



Falls das Interface noch nicht vom PC getrennt wurde, so sollten Sie dies vor Deinstallation des Treibers vornehmen und das folgende Fenster quittieren:



Bestätigen Sie im folgenden Fenster das Löschen der Treiberdateien:



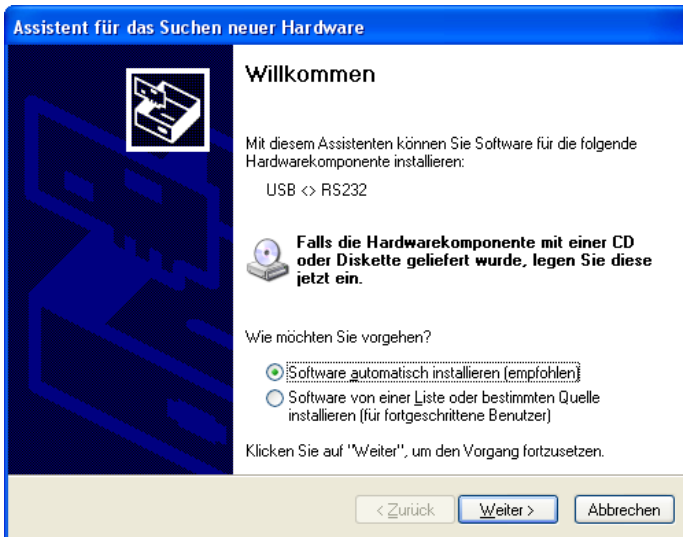
Damit ist die Deinstallation der Treiber beendet.

Windows XP

1. Installation

Die Installation der Treiber geschieht weitgehend automatisiert. Im Verlauf der Installation muß lediglich der Pfad angegeben werden, unter dem die Treiberdateien zu finden sind, wenn nicht die Treiber von der mitgelieferten CD-ROM installiert werden sollen.

Nach erstmaligem Anschluß des USB-Interfaces an den PC meldet Windows mit dem folgenden Fenster, daß eine neue Hardwarekomponente gefunden wurde und für diese Komponente ein Treiber installiert wird. Bestätigen Sie die Installation mit *Weiter* und lassen Sie Windows nach einem passenden Treiber für das angeschlossene Gerät suchen:



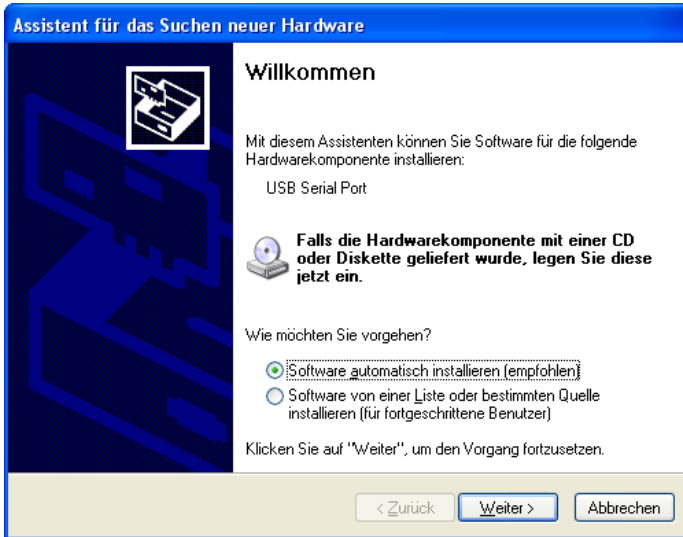
Windows fordert nun eine Bestätigung, dass der Treiber installiert werden soll. Bestätigen Sie dies mit *Installation fortsetzen*.



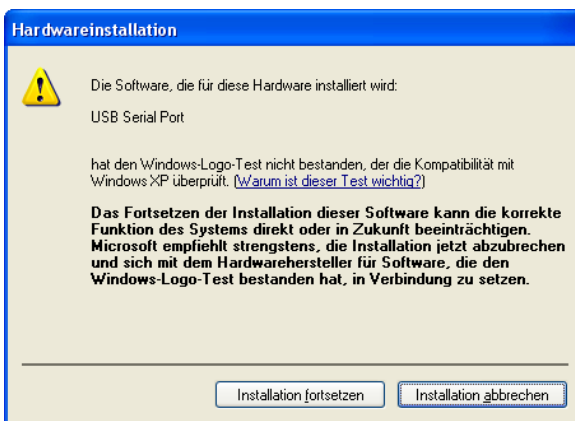
Windows bestätigt die abgeschlossene Installation. Bestätigen Sie dies mit *Fertig stellen*.



Nach Installation des Treibers ist zusätzlich noch die Installation der COM-Umlenkung erforderlich, die für das serielle Interface einen virtuellen COM-Port einrichtet. Dieser Vorgang wird automatisch nach Installation des Treibers gestartet und erfordert dieselben Eingaben, wie bei der Treiber-Installation.



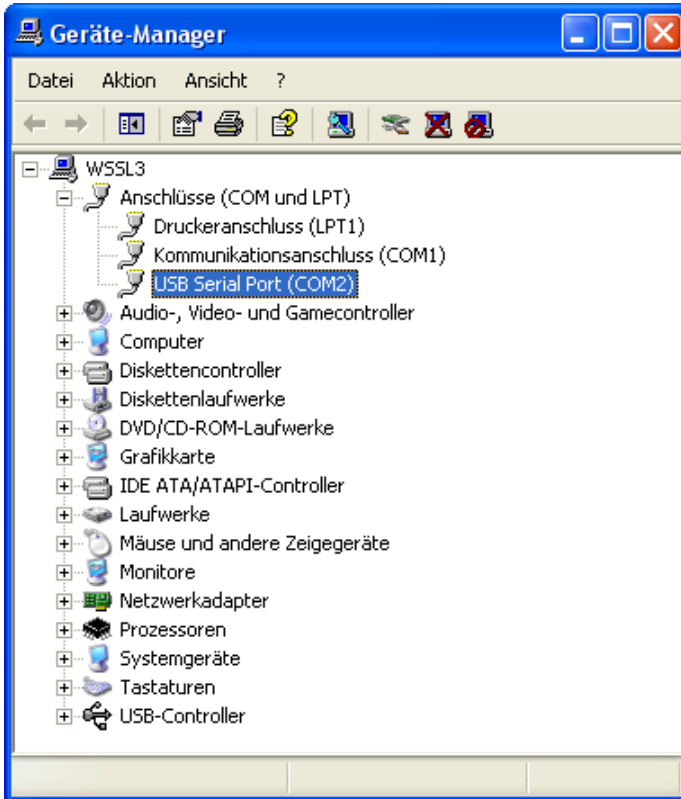
Windows fordert nun eine Bestätigung, dass der Treiber installiert werden soll. Bestätigen Sie dies mit *Installation fortsetzen*.



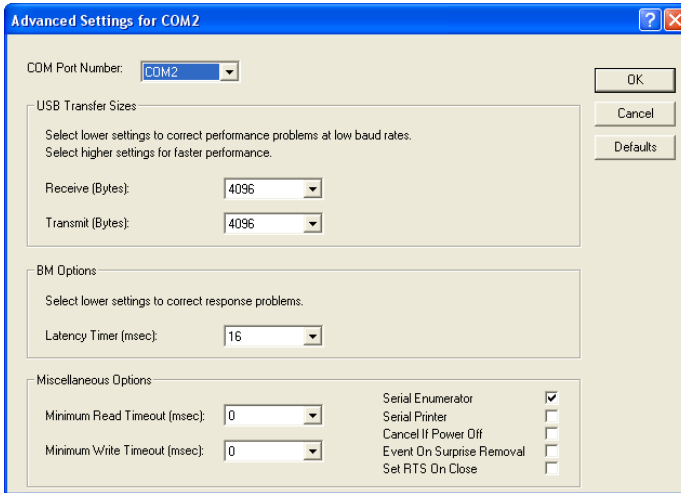
Windows bestätigt die abgeschlossene Installation. Bestätigen Sie dies mit *Fertig stellen*.



Nach erfolgter Installation finden Sie unter *Start -> Einstellungen -> Systemsteuerung -> System -> Hardware -> Geräte-Manager -> Anschlüsse (COM und LPT)* einen Eintrag für das einggerichtete USB-Interface - im Beispiel unten als COM2:



Durch einen Doppelklick auf den Eintrag läßt sich das Interface konfigurieren, wobei die unter *Port Settings* gesetzten Werte in der Regel von der in der entsprechenden Anwendung gewählten Konfiguration überschrieben werden. Interessant sind in diesem Zusammenhang die unter *Advanced...* vorgesehenen Möglichkeiten, den dem Interface zugewiesenen COM-Port zu verändern und den Latency Timer anzupassen (siehe technische Hintergrundinformationen).



Warnhinweis:

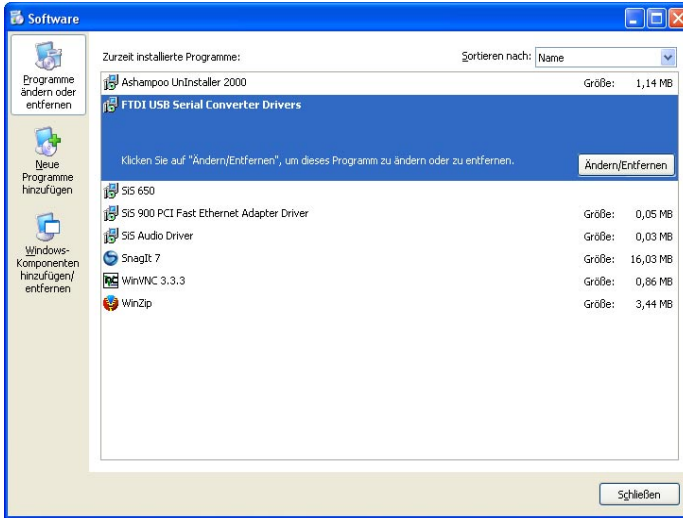
Es sollte unbedingt darauf geachtet werden, daß eine COM-Port-Nummer nicht mehrfach vergeben wird. Diese Möglichkeit besteht in Fällen, in denen nicht permanent alle installierten seriellen Devices im oder am PC in Betrieb sind. Dies ist zum Beispiel der Fall, wenn zusätzliche virtuelle COM-Schnittstellen über ein Netzwerk eingerichtet und zur Zeit nicht mit dem PC verbunden sind. Falls der zugehörige Treiber die Ressourcen wieder freigibt, wenn die Komponente nicht angeschlossen ist, besteht die Möglichkeit eine COM-Port-Nummer auszuwählen, die eigentlich schon vergeben ist und bei Wiederanschluß des Gerätes reaktiviert wird.



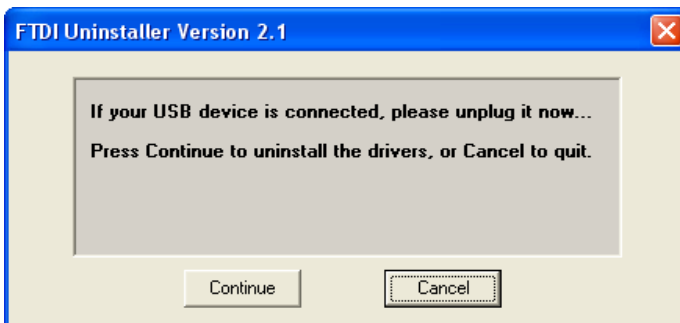
In diesem Fall existiert dann ein COM-Port mehrfach und Windows quittiert den Versuch, die Schnittstelle zu öffnen im günstigsten Fall mit einer Fehlermeldung.

2. Deinstallation

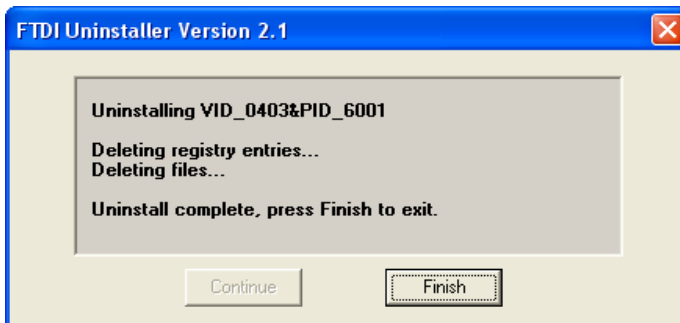
Die Deinstallation der Treibersoftware geschieht unter *Start -> Einstellungen -> Systemsteuerung -> Software*. Wählen Sie den *FTDI USB-to-Serial Converter Drivers* Punkt aus und bestätigen Sie die Auswahl mit *Ändern/Entfernen...*



Falls das Interface noch nicht vom PC getrennt wurde, so sollten Sie dies vor Deinstallation des Treibers vornehmen und das folgende Fenster quittieren:



Bestätigen Sie im folgenden Fenster das Löschen der Treiberdateien:



Damit ist die Deinstallation der Treiber beendet.

Technische Hintergrundinformationen

Datendurchsatz und Latenzzeit der W&T USB-Interfaces

Der folgende Text beschreibt die grundsätzlichen Architekturunterschiede, die von Anwendern der seriellen W&T USB-Interfaces berücksichtigt werden müssen, wenn sie von einer herkömmlichen seriellen Verbindung eines Peripheriegerätes mit einem PC zu einer Verbindung auf USB-Basis überwechseln wollen.

Ein Vergleich: Datenübertragung via serieller Schnittstelle und USB

Prinzipbedingt erzeugt die Datenübertragung via USB Verzögerungen, die beim direkten Anschluß eines Peripheriegerätes an die lokale COM-Schnittstelle eines PC nicht auftreten:

Die ursprünglichen seriellen Schnittstellen sind lokal im Rechner installiert und werden durch Interrupts gesteuert. Nach Senden oder Empfangen eines oder mehrerer Zeichen unterbricht die CPU ihren Betrieb und wechselt zu einer Routine über, die die empfangenen Daten umgehend weiterverarbeitet bzw. neue Zeichen in den Sendebuffer des UARTs schreibt. Dies bedeutet, dass sich ein Anwender bei einer bestimmten Übertragungsgeschwindigkeit im Allgemeinen darauf verlassen kann, dass die Datenübertragung ohne Flusskontrolle durchgeführt werden kann. Der Hardware-Interrupt stellt sicher, dass die Anforderung ohne größere Zeitverluste bearbeitet wird. Daten können deshalb ohne Handshake übertragen werden und trotzdem ohne Datenverlust im PC ankommen.

Beim USB erfolgt die Datenübertragung dagegen nicht mittels Interrupts. Er verwendet ein Scheduling-System, d. h. es können Zeitintervalle vorhanden sein, in denen eine USB-Anforderung nicht abgearbeitet werden kann, weil das System gerade mit anderen Aufgaben beschäftigt ist. Wenn in diesem Fall auf der seriellen Seite des Interfaces kein Handshake verwendet wird, kommt es unweigerlich zum Datenverlust. Eine solche Scheduling-Verzögerung können Sie zum Beispiel beobachten,

wenn eine geöffnetes Anwendungsfenster mit der Maus umherbewegt wird.

Bei USB-Geräten erfolgt die Datenübertragung in Paketen. Wenn Daten aus dem PC gesendet werden sollen, wird ein Datenpaket vom Gerätetreiber erstellt und an den USB-Scheduler übergeben. Dieser Scheduler setzt die Anforderung auf die Liste der von der USB-Hoststeuerung auszuführenden Aufgaben. Dieser Vorgang nimmt in der Regel mindestens eine Millisekunde in Anspruch, da die neue Anforderung erst beim nächsten USB-Datenübertragungsblock verarbeitet wird. Das Datenübertragungsblock-Intervall beträgt somit 1 ms. Es ist deshalb, je nach erforderlichem Durchsatz, ein beträchtlicher Overhead mit der Übertragung der Daten von der Anwendung zum USB-Interface verbunden. Würde deshalb zum Beispiel eine Anwendung Daten byteweise übertragen, hätte dies für das System einen stark eingeschränkten Gesamtdurchsatz zur Folge.

Fortlaufende Daten

Daten vom USB werden grundsätzlich per Polling vom PC empfangen. Der Treiber fordert eine bestimmte Menge an Daten in Vielfachen von 64 Byte vom USB-Scheduler an, was gleichzeitig der maximalen Paketgröße beim USB entspricht. Die Hoststeuerung liest Daten aus dem Gerät, bis die angeforderte Datenmenge erreicht ist oder bis ein Paket mit weniger als 64 Byte Inhalt empfangen wird.

Die vom Gerätetreiber angeforderten Datenmengen liegen zwischen 64 Byte und 4 Kilobyte. Die Größe des Datenblocks beeinflusst erheblich die Leistung der Interface-Lösung und ist von der seriellen Übertragungsgeschwindigkeit abhängig: bei extrem hohen Geschwindigkeiten ist die höchste Paketgröße erforderlich.

Bei Echtzeit-Anwendungen, die zum Beispiel Audiodaten mit einer Geschwindigkeit von 115.200 Baud übertragen, ist dagegen die kleinstmögliche Paketgröße wünschenswert, da das Gerät sonst jeweils 4 KB Daten zwischenspeichert.

Dies kann den Effekt einer ‚sprungartigen‘ Datenübertragung bewirken, wenn die USB-Anforderungsgröße zu hoch und die Übertragungsgeschwindigkeit (relativ) zu niedrig ist. Die neueste Treiber-Version variiert die angeforderte Paketgröße automatisch je nach ausgewählter Baudrate.

Kleine Datenmengen und Pufferende

Bei der Übertragung von Daten aus dem Interface an den PC sendet das Gerät die Daten, wenn eine der folgenden Bedingungen eintritt: -

1. Der Puffer ist voll (64 Byte – 2 Statusbyte und 62 Benutzerbyte).
2. Eine der RS232-Statusleitungen hat sich geändert (nur FT232BM Chip). Eine Änderung des Pegels (hoch oder niedrig) auf CTS# / DSR# / DCD# oder RI# bewirkt, dass der aktuelle Puffer zurückgeschickt wird, auch wenn er leer ist oder weniger als 64 Byte enthält.
3. Ein Ereigniszeichen wurde aktiviert und im eingehenden Datenstrom erkannt.
4. Ein im Chip eingebauter Zeitgeber ist abgelaufen. Sowohl der FT232BM als auch der FT245BM Chip enthält einen Zeitgeber, der die Zeitspanne seit der letzten Datenübertragung an den PC misst. Die Standardeinstellung des Zeitgebers beträgt 16 ms. Der Zeitgeber wird jedes Mal dann zurückgesetzt, wenn Daten wieder an den PC geschickt werden. Wenn der Zeitgeber abläuft, sendet der Chip die 2 Statusbyte und alle anderen im Puffer befindlichen Daten zurück.

Hieraus ist ersichtlich, dass kleine Datenmengen (bzw. das Ende großer Datenmengen) bei der Übertragung an den PC einer Verzögerung von 16 ms unterliegen. Diese Verzögerung kommt zu den mit der USB-Anforderungsgröße verbundenen Verzögerungen (siehe Beschreibung im vorigen Abschnitt) hinzu. Der Zeitgeberwert wurde so gewählt, dass 64-Byte-Pakete große Puffer im Hochgeschwindigkeitsmodus füllen. Darüber hinaus können auch einzelne Zeichen durchgelassen werden.

Der gewählte Wert beträgt 16 ms. Dies bedeutet, dass der Empfang eines einzelnen Zeichens 16 ms zusätzlich zur Übertragungszeit im seriellen oder parallelen Verbund in Anspruch nimmt.

Der Zeitgeber wird bei großen Datenmengen und hohen Übertragungsgeschwindigkeiten nicht verwendet. Er wird u. U. zum Senden des letzten Pakets eines Blocks eingesetzt, falls dieses weniger als 64 Byte enthält. Die ersten 2 Byte jedes Pakets dienen als Statusbyte für den Treiber. Dieser Status wird alle 16 ms gesendet, auch wenn keine Daten im Gerät vorhanden sind.

Wenn 62 Byte Daten in 16 ms empfangen werden, könnte ein Problem auftreten. Dies würde zwar keine Zeitsperre bewirken, die 64 Byte (2 Statusbyte + 62 Benutzer-Datenbyte) würden jedoch alle 16 ms per USB zurückgesendet werden. Wenn der USB-D-Systemtreiber die 64 Byte empfängt, würde er diese beibehalten und eine weitere ‚EIN‘ Transaktion anfordern. Dieser Vorgang würde in weiteren 16 ms abgeschlossen sein usw., bis USB-D alle der erforderlichen 4 KB Daten empfangen hat. Die Gesamtzeit würde $(4096 / 64) * 16 \text{ ms} = 1,024 \text{ s}$ zwischen von der Anwendung empfangenen Datenpaketen betragen. Damit die Daten nicht in Paketen von 4 KB ankommen, sollten diese in kleineren Mengen angefordert werden. Ein kleines Paket (< 64 Byte) bewirkt natürlich, dass die Daten für den Gebrauch durch die Anwendung vom USB-D zurück an unseren Treiber geschickt werden.

*** Hinweis für Anwendungsprogrammierer: Daten sollten mit Puffern und nicht anhand einzelner Zeichen übertragen bzw. empfangen werden.

Justierung des Latenzzeitgebers für den Empfangspuffer

Mit der zweiten Generation der BM Chips von FTDI kann der Latenzzeitgeber von 16 ms auf einen beliebigen Wert zwischen 1 und 255 ms in Inkrementen von 1 ms geändert werden. Bei Verwendung des FTDI-Treibers ‚Virtual Com Port‘ kann der Latenzzeitgeber auf der Seite ‚Anschlusseigenschaften‘ eingestellt werden. In Windows kann die Seite ‚Anschlusseigenschaften‘ über den Geräte-Manager aufgerufen werden. Der Anfangswert

des Latenzzeitgebers kann auch in `ftdiport.inf` vorkonfiguriert werden, indem der Wert der letzten Zahl in der folgenden Zeile geändert wird –

```
HKR,,“LatencyTimer“,0x00010001,16
```

Wobei 16 ms wiederum die Standardeinstellung ist.

Bei Verwendung des FTDI D2XX Direkttreibers kann der Wert des Latenzzeitgebers über die Funktion `FT_SetLatencyTimer` justiert werden.

Auswirkung von USB-Puffergröße und Latenzzeit auf den Datendurchsatz

Die Größe der USB-Gesamtpaketanforderung hat eine nicht sofort erkennbare Auswirkung auf den Datenfluss. Wenn eine Leseanforderung an USB gesendet wird, liest die USB-Hoststeuerung weiterhin 64-Byte-Pakete, bis eine der folgenden Bedingungen erfüllt ist: -

- a) Sie hat die angeforderte Größe gelesen (Standardwert: 4 KB).
- b) Sie hat ein Paket mit weniger als 64 Byte vom Chip empfangen.
- c) Sie wurde unterbrochen.

Während die Hoststeuerung auf die Erfüllung einer der obigen Bedingungen wartet, werden KEINE Daten von unserem Treiber, d. h. der Benutzeranwendung, empfangen. Die Daten (sofern vorhanden) werden erst dann übertragen, nachdem eine der obigen Bedingungen erfüllt wurde.

Normalerweise tritt Bedingung (c) nicht auf, weshalb wir uns Fall (a) und (b) ansehen werden. Wenn 64-Byte-Pakete ständig an den Host zurückgeschickt werden, wird er weiterhin die der angeforderten Blockgröße entsprechenden Daten lesen, bevor er den Block wieder an den Treiber sendet. Wenn eine kleine Datenmenge gesendet wird oder die Daten sehr langsam gesendet werden, übernimmt der Latenzzeitgeber die Kontrolle und

sendet ein kurzes Paket an den Host zurück, wodurch die Leseanforderung beendet wird. Die bisher gelesenen Daten werden dann über den FTDI-Treiber an die Benutzeranwendung weitergeleitet. Dies weist auf eine Beziehung zwischen Latenzzeitgeber, Übertragungsgeschwindigkeit und Verfügbarmachung der Daten hin. Wenn Daten mit einer solchen Geschwindigkeit an den FTDI-Chip übertragen werden, dass ein Abfließen des Latenzzeitgebers vermieden wird, kann zwischen dem Empfangen einzelner Datenblöcke eine lange Zeit vergehen. Diese Bedingung tritt ein, da die Hoststeuerung 64-Byte-Pakete genau vor Ablauf der Latenzzeit feststellen und deshalb weiterhin die Daten lesen wird, bis die Blockgröße erreicht ist. Die Daten werden anschließend wieder an die Benutzeranwendung geschickt.

Die entsprechende Geschwindigkeit wäre:

62 / Latenzzeitgeber Byte/Sekunde

(Wir verwenden 2 Byte pro 64-Byte-Paket)

Standardwerte:

$62 / 0,016 \sim = 3875 \text{ Byte/Sekunde} \sim = 38,75 \text{ kBaud}$

Wenn Daten also mit einer Geschwindigkeit von 3875 Byte pro Sekunde (38,75 kBaud) empfangen werden, unterliegen die Daten je nach angeforderter USB-Blocklänge Verzögerungen. Wenn Daten mit einer langsameren Geschwindigkeit empfangen werden, stehen nach 16 ms weniger als 62 Byte (64 einschließlich unserer 2 Statusbyte) zur Verfügung. Es kommt deshalb zu einem kurzen Paket, wodurch die USB-Anforderung beendet wird und die Daten zurückgeschickt werden. Bei einer Übertragungsgeschwindigkeit von 38,75 kBaud verstreichen etwa 1,06 s vom Datenpuffern bis in die Benutzeranwendung (vorausgesetzt, die angeforderte USB-Blockgröße beträgt 4 KB).

Um diese Verzögerung zu umgehen, können Sie entweder den Wert des Latenzzeitgebers erhöhen oder die angeforderte USB-Blocklänge reduzieren. Die Reduzierung der angeforderten USB-Blocklänge stellt die bevorzugte Methode dar, obwohl ein

Gleichgewicht zwischen den zwei Optionen für eine optimale Systemreaktion erforderlich sein kann.

Justierung der USB-Übertragungsgröße

Bei Verwendung des FTDI Virtual COM-Port-Treibers kann auch die USB-Übertragungs-/Puffergröße auf der Seite ‚Anschlusseigenschaften‘ eingestellt werden. Die anfängliche Puffergröße wird aus Einträgen in der Datei ftdiport.inf berechnet – wobei die Größe des zugewiesenen Puffers dem Eintrag in der „inf“-Datei plus 1, multipliziert mit 64 (Byte) entspricht.

Dies bedeutet: 0 ist 64 Byte, und 3F ist $(63+1)*64 = 4096$.

Es sind zwei Einträge in der INF-Datei vorhanden – der erste Eintrag gehört zum Sendepuffer, der zweite Eintrag zum Empfangspuffer.

```
HKR,,ConfigData,1,01,00,3F,3F,10,27,88,13,C4,09,E2,04,71,02,38,  
41,9c,80,4E,C0,34,00,1A,00,0D,00,06,40,03,80,00,00,00,00
```

Im obigen Beispiel sind die zwei 3F die entsprechenden Einträge. Mit diesen wird die Puffergröße auf 4 KB eingestellt.

```
HKR,,ConfigData,1,01,00,00,00,10,27,88,13,C4,09,E2,04,71,02,38,  
41,9c,80,4E,C0,34,00,1A,00,0D,00,06,40,03,80,00,00,00,00
```

Stellt eine Puffergröße von 64 Byte ein.

Bei Verwendung des FTDI D2XX Direkttreibers kann die Größe des angeforderten USB-Blocks über die Funktion FT_SetUSBParameters justiert werden. Siehe separate FTDI-Anwendungsrichtlinie in Bezug auf Hinweise zur Verwendung von FT_SetUSBParameters für optimierten Datendurchsatz.

Ereigniszeichen

Ereigniszeichen können sowohl mit FT232BM als auch mit FT245BM Geräten verwendet werden. Wenn das Ereigniszeichen aktiviert ist und im Datenstrom erkannt wird, dann wird der Inhalt des Gerätepuffers sofort übertragen. Das Ereigniszeichen

wird weder vom Gerät noch von den Treibern aus dem Datenstrom entfernt. Diese Aufgabe obliegt der Anwendung. Ereigniszeichen können ein- oder ausgeschaltet werden – je nachdem, ob große Mengen an Zufallsdaten oder kleine Befehlsfolgen gesendet werden sollen. Das Ereigniszeichen darf nicht das erste Zeichen im Puffer sein, da es sonst nicht funktioniert. Es muss Zeichen 2 oder höher sein. Anwendungen zum Beispiel, die das Internet verwenden, programmieren das Ereigniszeichen als ‚\$7E‘. Es werden dann alle Daten in Paketen, die mit ‚\$7E‘ beginnen und enden, gesendet und empfangen. Das Ereigniszeichen wird nicht an der ersten Position ausgelöst, damit der Durchsatz maximiert und ein nur mit ‚\$7E‘ beginnendes Paket vermieden wird.

Leeren des Empfangspuffers mit den Modem-Statusleitungen

Der Puffer im USB-UART-Chip der W&T USB-Interfaces kann auch mit Hilfe von Handshake-Signalen geleert werden. Dies kann durch Änderung einer der Modem-Statusleitungen erzielt werden. Die Modem-Statusleitungen können von einem externen Gerät oder vom Host-PC selbst gesteuert werden. Wenn eine unbelegte Ausgangsleitung (DTR) an einen der unbelegten Eingänge (DSR) angeschlossen ist, und wenn der Zustand der DTR-Leitung durch das Anwendungsprogramm geändert wird, bewirkt dies eine Änderung auf DSR und der Puffer wird geleert.

Flusskontrolle

Die W&T USB-Interfaces beherrschen sowohl RTS/CTS- bzw. DTR/DSR-Hardware-Handshake, als auch XON-/XOFF-Softwarehandshake. Es sollte unbedingt eine der unterstützten Handshake-Arten verwendet werden, um Datenverluste zu vermeiden.

Es sind 4 Methoden zur Flusskontrolle vorhanden, die für die Interfaces programmiert werden können:

Kein Handshake: Dies kann bei hohen Baudrates zu Datenverlusten führen

RTS-/CTS-Handshake: Das Interface sendet, wenn CTS aktiv ist, und deaktiviert RTS, wenn der interne Buffer überzulaufen droht.

DTR-/DSR-Handshake: Das Interface sendet, wenn DSR aktiv ist, und deaktiviert DTR, wenn der interne Buffer überzulaufen droht.

XON/XOFF-Handshake: Die Flusskontrolle erfolgt über das Senden bzw. Empfangen spezieller Zeichen. Das eine Zeichen ist XON (Übertragung ein), das andere ist XOFF (Übertragung aus). Sie können einzeln auf einen beliebigen Wert programmiert werden.

Generell wird die Verwendung einer Handshake-Steuerung dringend empfohlen, da das Auftreten von Scheduling-Verzögerungen nicht verhindert werden kann und der USB-UART selbst lediglich bis zu 384 Byte Daten intern zwischenspeichern kann. Es kann lediglich dann auf eine Flusskontrolle verzichtet werden, wenn die Übertragungsgeschwindigkeit niedrig bzw. ein Verlust gesendeter und empfangener Daten akzeptabel ist.

Technische Hintergrundinformationen:

© Future Technology Devices International Limited, 2002/2003

<http://www.ftdichip.com>