CAN Bus Windows Treiber
Beschreibung für Version 2.3

Autor:
Jürgen Eder
Hermann Hesse Weg 7
73257 Köngen
eMail: Juergen.Eder@gmx.de

Inhaltsverzeichnis

Kapitel 1

Allgemeines

1.1 Support:

Nur per eMail, der Treiber entsteht in meiner Freizeit, bitte nicht anrufen! Aufgrund der vielfältigen PC Hardware haben Sie bitte auch Verständnis, wenn ich in einzelnen Fällen nicht weiterhelfen kann. Der Treiber ist unter anderem auch aus diesem Grund als SHAREWARE veröffentlicht: Sie können die Software testen und so feststellen, ob der Treiber mit Ihrer Hardware und Software kompatibel ist.

1.2 Lizenzvertrag / Copyright

1.2.1 Lizenzvertrag

Lesen Sie die folgenden Zeilen sorgfältig durch, bevor Sie diese Software benützen. Sollten Sie mit dem folgenden nicht einverstanden sein, so benutzen Sie diesen Treiber nicht. Sie haben hiermit das Recht, den Treiber und die damit verbundenen Dateien eingehend zu testen. Sie dürfen den Treiber und die damit verbundene Dateien beliebig weitergeben, solange Sie exakte Kopien ohne Veränderung anfertigen.

1.2.2 Einschränkung der Shareware-Version

Die im Internet verfügbare Version ist eingeschränkt in der Baudratenauswahl. Es können nur Baudraten bis maximal 400 kBit eingestellt werden. Nach der Registrierung der Sharewaregebühr erhalten Sie dann eine Vollversion ohne diese Einschränkung. Der Treiber ist weder Freeware noch Public Domain. Eine regelmässige Benützung benötigt eine Registrierung.

1.2.3 Vollversion (Einzellizenz)

Der Treiber (Einzellizenz) darf auf beliebig vielen Computern installiert werden, solange sie nur von einer Person gleichzeitig benutzt wird. Die Nutzung durch mehrere Personen auf mehreren Computern gleichzeitig erfordert zusätzliche Lizenzen. Eine Einzellizenz kostet EURO 12,50 inkl. MwSt.. Für weitere Lizenzen fragen Sie bitte nach.

1.2.4 Quellkode

Es ist auch möglich, den Quellkode zu erwerben. Sie erhalten damit das Recht, den Treiber beliebig zu ändern und die daraus resultierenden Binärdateien beliebig oft

weiterzugeben. Der Quellkode darf jedoch nicht weitergegeben werden. Falls an dem Treiber Änderungen vorgenommen werden, muss auch folgendes beachtet werden:

1. Alle Texte CANPAR in der Binärdatei und der Registrierdatenbank müssen umbenannt werden, damit es nicht zu einem Konflikt kommt, wenn der Original CANPAR Treiber und eine modifizierte Version installiert sind.

2. Es muss absolut eindeutig zu erkennen sein, dass dies eine modifizierte Version ist. Dies kann z.B. dadurch geschehen, dass alle Original Copyright Hinweise entfernt werden oder durch eigene Copyright Hinweise ersetzt werden.

Der Quellkode kostet EURO 150,00 ohne MwSt. Als Quellkode wird versendet:

- der aktuelle Quellkode der Windows9x / Windows NT4.0 Version oder eine beliebige ältere Version

Um den Quellkode zu verwenden sind ausserdem notwendig (und nicht Bestandteil des Quellkodes):

- Windows SDK
- Windows DDK für Windows 95 (Windows 9x Treiber)
- Windows DDK für Windows NT4.0 (Windows NT Treiber)
- Visual C++ 6.0
- MASM6 (für Windows 9x Treiber)

1.2.5 Haftung

Allgemein gilt: keine Software ist fehlerfrei, und die Anzahl der Fehler steigt mit der Komplexität des Programms. Deshalb kann keine Gewähr dafür übernommen werden, dass diese Software in jeder Umgebung, auf jedem Rechner, und mit jeglichen anderen Anwendungen zusammen fehlerfrei läuft. Jegliche Haftung für direkte wie indirekte Schäden wird hiermit ausgeschlossen, soweit dies gesetzlich zulässig ist. In jedem Fall jedoch ist die Haftung beschränkt auf die Registriergebühr.

Testen Sie diese Software gründlich mit unkritischen Daten, für Schäden an Daten wird keinerlei Haftung übernommen. Sollten Sie bis zur Registrierung Fehler entdecken, so akzeptieren Sie diese, sofern Sie sich trotzdem registrieren lassen. Jegliche Fehlerbeschreibung wird gerne entgegengenommen, jedoch kann keine Garantie gegeben werden, dass alle Fehler behoben werden können.

Alle erwähnten Warenzeichen und Copyrights gehören ihren jeweiligen Besitzern.

1.2.6 Bezahlung

Überweisung in Euro; Vorgehensweise: Schicken Sie Ihre komplette Adresse als eMail an Juergen.Eder@gmx.de. Ich schicke Ihnen dann die Kontendaten per eMail zu.

1.3 Unterstützte Hardware

Unterstützte Karten Ab Version 2.2 wird folgende Hardware unterstützt:

- CAN Interface für den Parallelport aus dem ELEKTOR Magazin Juni 2000
- CAN Interface für den Parallelport des CAN200 Projekts (siehe Hardware und Software für Linux: http://private.addcom.de/horo/can200)

- Modifiziertes CAN Interface mit 16MHz Quarz (statt 8 MHz) für den Parallelport des CAN200 Projekts

Die Treibersoftware wurde unabhängig von diesen Projekten und komplett neu programmiert.

Kapitel 2

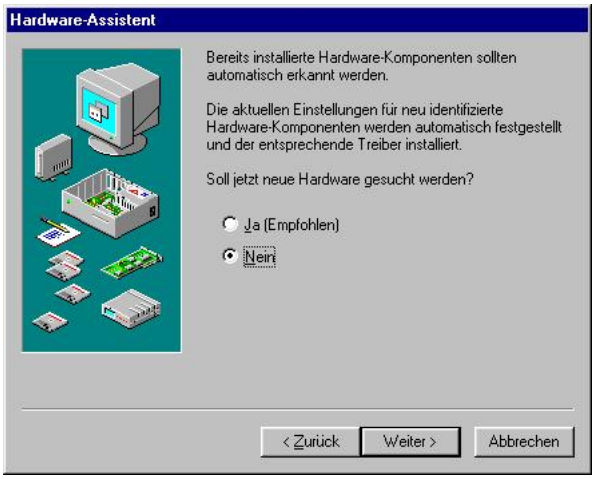
Treiber

2.1 Installation Windows 9x

2.1.1 Erstinstallation:

Die Installation erfolgt üblicherweise über die Systemsteuerung Hardwaremanager. Starten Sie also den Hardwaremanager durch Doppelklick:

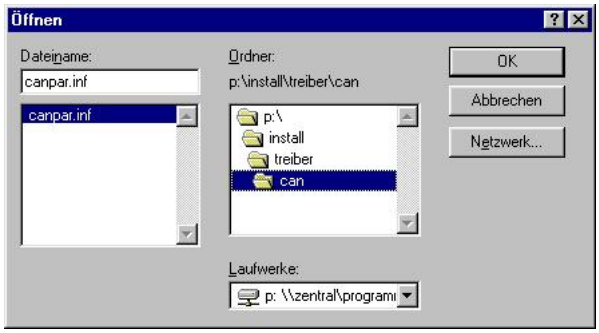
Danach erscheint ein neuer Dialog: Hardware suchen, der mit Nein beantwortet wird:



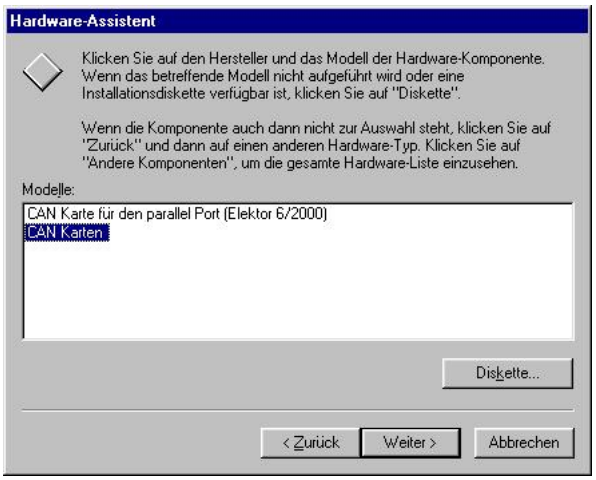
Es erscheint eine Geräteliste, woraufhin das Gerät andere Komponente (bzw. CAN, wenn der CAN Treiber bereits einmal installiert war) ausgewählt wird:

In den danach erscheinendem Dialog wird Diskette angeklickt. Danach muss der Pfad eingegeben werden in dem die Treiberdateien liegen: (z.B. A:\):

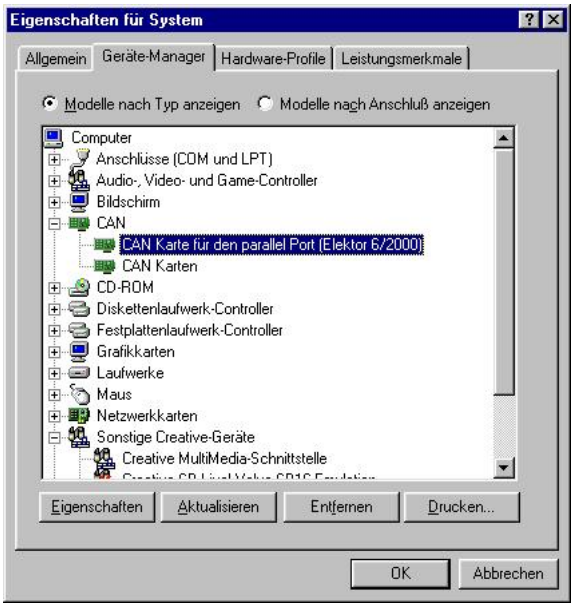
Alternativ kann natürlich auch auf Durchsuchen geklickt werden und die Treiberdateien auf diese Art gesucht werden:



Als Treiber wird dann: CAN Karten ausgewählt:



Nach der Installation kann im Gerätemanager der Anschluss beliebig geändert werden (LPT1 ist die Standardeinstellung). In der Liste werden nur die - dem System bekannten - LPT Schnittstellen aufgelistet:



Anmerkung: Da zusätzliche Schnittstellen auf Zusatzkarten unter Umständen anders programmiert werden müssen als die Schnittstellen auf einem PC Motherboard, kann keine Funktionsgarantie für zusätzliche Karten gegeben werden. Im Zweifel hilft nur ausprobieren!

Der Kartentyp kann jederzeit geändert werden, als Standard ist nach einer Installation immer die ELEKTOR Karte ausgewählt.

2.1.2 Update

Die Installation läuft über die Systemsteuerung ->Treiber aktualisieren

2.1.3 Deinstallation

1. Im Gerätemanager das Gerät: CAN Karten anklicken und auf Entfernen klicken

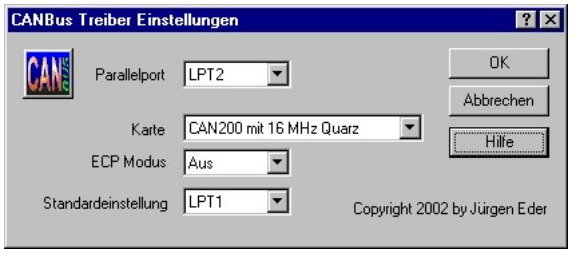
2.2 Installation Windows NT4.0 / 2000 / XP

Grundsätzlich gilt bei diesen Betriebssystemen:

> Zur Installation, zur Deinstallation und um Treibereinstellungen zu ändern, werden *A*dministratorrechte benötigt

2.2.1 Erstinstallation

Die Installation erfolgt über das Installationsprogramm. Über *CAN-Bus Settings* werden die Einstellungen vorgenommen. Für jeden Parallelport kann dazu eine Einstellung gespeichert werden. Ausser wenn das Standardgerät geändert wurde, muss nicht mehr neu gebootet werden, der Treiber wird dann automatisch neu gestartet. Hinweise werden ausserdem in *CANBus Settings* ausgegeben.



Hinweis zum ECP Modus: Die Windows Treiber-API kennt keinen ECP Modus, so dass nur über direkte Zugriffe auf die Hardwareregister dieser Modus verwendet werden kann. Dieser Modus ist dadurch sehr problematisch und sollte, wenn möglich, nicht verwendet werden. Falls dies auf dem jeweiligen PC eingestellt werden kann, sollte der Modus EPP oder Bidirektional verwendet werden. Nach einem Treiberneustart bzw. ein Reboot ist die Einstellung übernommen

2.2.2 Update

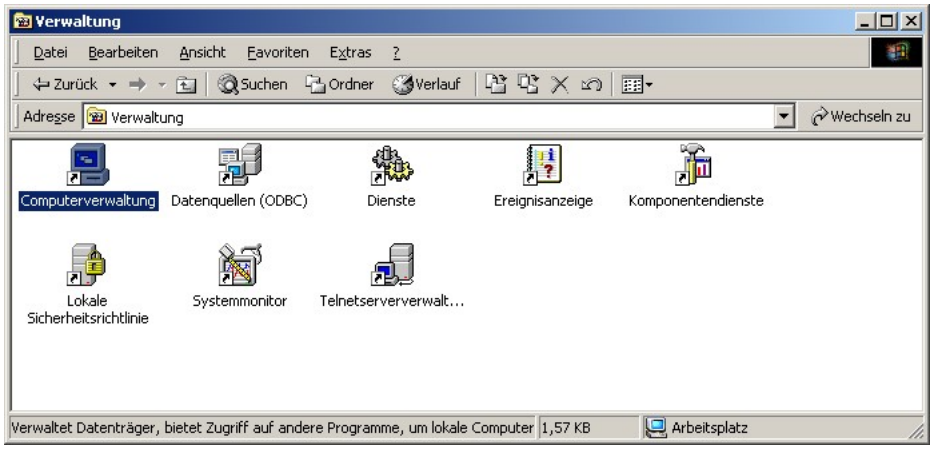
Installationsprogramm verwenden, wie bei einer Neuinstallation. Die Einstellungen müssen danach ausserdem neu angegeben werden.

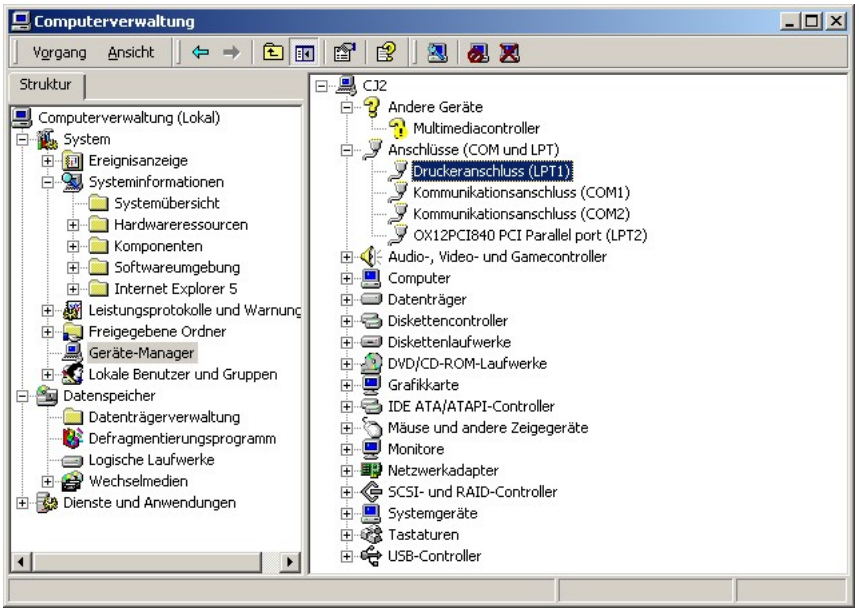
2.2.3 Deinstallation

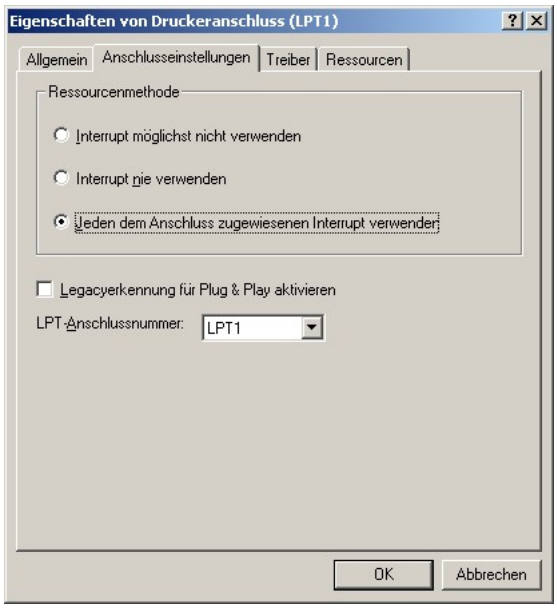
Der Treiber wird über die *Systemsteuerung->Software* deinstalliert. Nach einem Reboot ist der Treiber vollständig gelöscht

2.2.4 Installationshinweise für Windows 2000

Unter Windows 2000 wurde nur ein (erfolgreicher) Kurztest durchgeführt. Zur Installation, Update oder Deinstallation geht man so vor wie oben für NT4.0 beschrieben. Danach müssen aber noch in *CANBus Settings* die Interrupts für die Parallelports aktiviert werden. Dazu wird einmal auf den *A*ktivieren Button geklickt. Zusätzlich muss aber auch noch im Geräte-Manager die Verwendung der Interrupts eingeschaltet werden:







Damit die neuen Einstellungen wirksam werden, muss auf jeden Fall ein Reboot erfolgen.

2.2.5 Installationshinweise für Windows XP

Unter XP wurde kein Test durchgeführt. Versuchsweise kann so vorgegangen werden, wie bei Windows 2000 beschrieben.

Kapitel 3

Beschreibung der zusätzlichen Software

3.1 CanParActiveX.OCX

CanParActiveX.ocx ist Public Domain, Weitergabe, Veränderung und Verwendung wird hiermit ohne Einschränkung erlaubt.

3.1.1 Allgemein

Ein ActiveX zur Verwendung in C und C++ Programme. Da die Rückgabe einer CAN Botschaft als BYTE-Array erfolgt (und dies im ActiveX nicht entsprechend berücksichtigt wird, d.h. die Rückgabe der Daten an die Applikation erfolgt zur Zeit nur als Pointer und nicht als SAFEARRAY) kann das ActiveX nicht in BASIC Programme verwendet werden.

Damit das ActiveX verwendet werden kann, muss es zunächst im System registriert werden. Dies geschieht am einfachsten über die Kommandozeile:

c:\windows\system\regsvr32 canparactivex.ocx

3.1.2 Eigenschaften des ActiveX:

CANStatus

Statusregister des SJA1000 auslesen

AcceptanceCode

Acceptanz-Code-Register des SJA1000 auslesen / setzen (Verwendbar für 11 Bit Frames), anstelle dieser Eigenschaft, sollte ExtAcceptance-Code verwendet werden.

AcceptanceMask

Acceptanz-Mask-Register des SJA1000 auslesen / setzen (Verwendbar für 11 Bit Frames), anstelle dieser Eigenschaft, sollte ExtAcceptance-Mask verwendet werden.

ExtAcceptanceCode

Acceptanz-Code-Register des SJA1000 auslesen / setzen (Verwendbar für 11 und 29 Bit Frames)

ExtAcceptanceMask

Acceptanz-Mask-Register des SJA1000 auslesen / setzen (Verwendbar für 11 und 29 Bit Frames)

Baudrate

Baudrate setzen. Die Werte für die Baudratenregister des SJA1000 werden im Treiber berechnet

LostDataMsg

Anzahl der verlorenen Nachrichten ermitteln. Beim Schreibzugriff wird dieser Zähler immer auf 0 gesetzt

Time

Zeit auslesen in ms, der Treiber liefert die Zeit als 64-Bit Wert mit einer Auflösung von 800 ns. Die Auflösung wird dann im ActiveX auf 1 ms heruntergerechnet.

ShowMessages

1 = Nachrichten an die Applikation senden
0 = keine Nachricht beim Empfang von Daten an die Applikation senden

3.1.3 Funktionen des ActiveX

OpenCan

OpenCAN();
CAN-Kanal öffnen, erst nach diesem Aufruf kann auf die CAN Karte zugegriffen werden.

CloseCan

CloseCan();
CAN-Kanal schliessen, der CAN Kanal wird ausserdem zwangsweise spätestens beim Beenden der ActiveX-Instanz geschlossen

GetRegister und SetRegister

GetRegister(short nIndex);
SetRegister(short nIndex, short nNewValue);
Ein Register des SJA1000 auslesen oder beschreiben

GetRegisterRR1 und SetRegisterRR1

GetRegisterRR1(short nIndex);
SetRegisterRR1(short nIndex, short nNewValue);
Ein Register im RESET-Mode des SJA1000 auslesen oder beschreiben

Transmit

Transmit(long id, short rtr, short dlc, short* data);
Can Botschaft senden. Die Daten werden als short-Array übergeben

TransmitData

TransmitData(long id, short rtr, BSTR data);
Can Botschaft senden. Die Daten werden hier als String übergeben: z.B. data = 12 34 56 af

GoToSleep

GoToSleep();
SJA1000 in den Power-Down Modus schalten

ClearDataOverrun

ClearDataOverrun();
Overrun-Bit löschen

AbortTransmission

AbortTransmission();
Eine gerade gesendete CAN Botschaft abbrechen

ReleaseReceiveBuffer

ReleaseReceiveBuffer();
Receive-Bit im SJA1000 rücksetzen

ResetTime

ResetTime();
Zeit rücksetzen (es wird nur die interne Referenz neu berechnet, der WINDOWS-Timer kann nicht rückgesetzt werden)

BaudrateDialog

BaudrateDialog();
Dialog anzeigen, in dem die Baudrate eingestellt wird

AcceptanceDialog

AcceptanceDialog();
Dialog anzeigen, in dem Acceptancecode und Acceptancemask eingestellt wird

AddTransmitThread

AddTransmitThread(long id, short rtr, short dlc, short* data, long delay);
Hintergrund-Thread anlegen, der ständig die übergebene CAN Botschaft im Zeitraster delay sendet. Die Daten werden als short-Array übergeben. Es kann nur ein Hintergrundthread angelegt werden!

AddTransmitDataThread

AddTransmitDataThread(long id, short rtr, BSTR data, long delay);
Hintergrund-Thread anlegen, der ständig die übergebene CAN Botschaft im Zeitraster delay sendet. Die Daten werden hier als String übergeben: z.B. data= 34 a2 fc Es nur ein Hintergrundthread angelegt werden!

DeleteThread

DeleteThread(short thread_ID);
Hintergrundthread beenden, zur Zeit wird die Id nicht berücksichtigt

3.1.4 Nachrichten des ActiveX

Hinweise:

- Die Zeit wird als STRING ausgegeben *"[absolut seit Start] relativ"*
- In status steht der Wert des Statusregisters des SJA1000
- In lost steht die Anzahl der verlorengegangen Botschaften, die empfangen wurden.
- Alle Nachrichten werden per Interrupt vom SJA1000 gemeldet und dann über diese Nachrichten an die Applikation weiter gemeldet. Die Nachrichten können jederzeit eintreffen, sobald der CAN Kanal geöffnet wurde!

CanReceived

CanReceived(BSTR time, short status, long lost, long id, short rtr, short dlc, short* msg);
Eine Botschaft wurde über den CAN Bus empfangen.

CANTransmitted

CANTransmitted(BSTR time, short status, long id, short rtr, short dlc, short* msg);
Eine Botschaft wurde soeben über den CAN Bus gesendet

CANError

CANError(BSTR time, short status);
Der SJA1000 meldet einen BUS-Error

CANWakeUp

CANWakeUp(BSTR time, short status);
Der Power-Down Modus wurde durch eine BUS-Aktivität beendet.

CANOverrun

CANOverrun(BSTR time, short status);
Der SJA1000 meldet einen Overrun