

VCI Wrapper

COM Objekt für VCI

IXXAT

Hauptsitz

IXXAT Automation GmbH
Leibnizstr. 15
D-88250 Weingarten

Tel.: +49 (0)7 51 / 5 61 46-0
Fax: +49 (0)7 51 / 5 61 46-29
Internet: www.ixxat.de
e-Mail: info@ixxat.de

Geschäftsbereich USA

IXXAT Inc.
120 Bedford Center Road
USA-Bedford, NH 03110

Phone: +1-603-471-0800
Fax: +1-603-471-0880
Internet: www.ixxat.com
e-Mail: sales@ixxat.com

Support

Sollten Sie zu diesem, oder einem unserer anderen Produkte Support benötigen, wenden Sie sich bitte schriftlich an:

Fax: +49 (0)7 51 / 5 61 46-29
e-Mail: support@ixxat.de

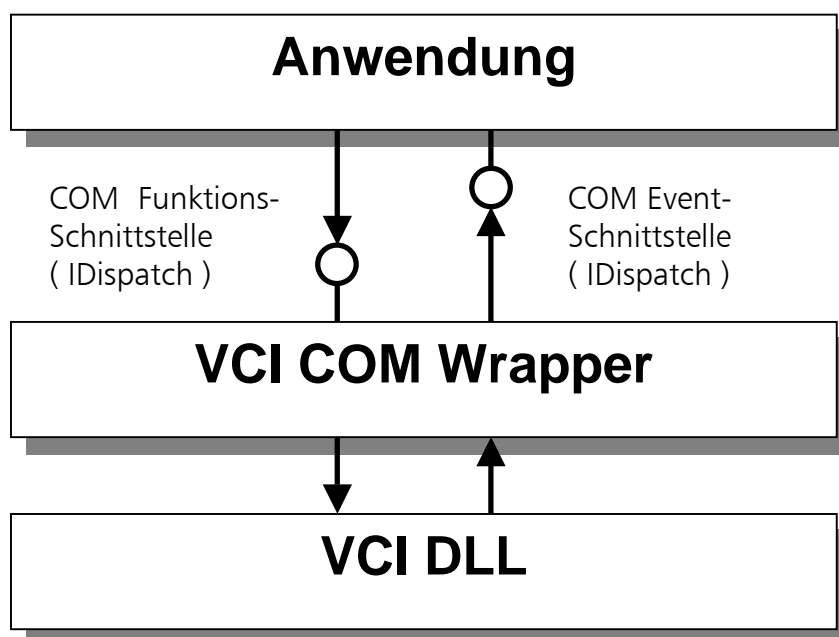
Copyright

Die Vervielfältigung (Kopie, Druck, Mikrofilm oder in anderer Form) sowie die elektronische Verbreitung dieses Dokuments ist nur mit ausdrücklicher, schriftlicher Genehmigung von IXXAT Automation erlaubt. IXXAT Automation behält sich das Recht zur Änderung technischer Daten ohne vorherige Ankündigung vor. Es gelten die allgemeinen Geschäftsbedingungen sowie die Bestimmungen des Lizenzvertrags. Alle Rechte vorbehalten.

1	EINLEITUNG	5
1.1	Anwendungsgebiet	5
1.2	Installation.....	5
2	PROGRAMMIERHINWEISE	6
2.1	Funktionsbeschreibung	6
2.2	Initialisierung (Visual Basic).....	7
2.3	Event Handler (Visual Basic)	9
2.4	Einschränkungen	10
2.5	Beispielprogramme.....	12

1 Einleitung

Die VCI Wrapper Softwarekomponente basiert auf der COM (Component Object Model) Technologie und legt über automatisierungskompatible Schnittstellen die Funktionalität der VCI DLL (siehe VCI-Handbuch) offen.



Anwendungen die direkt auf der VCI-DLL aufsetzen benutzen für die Signalisierung von asynchronen Ereignissen sog. Callbackhandler. Bei Verwendung der VCI Wrapper Softwarekomponente werden derartige Ereignisse über COM Events an die Anwendung weitergegeben.

1.1 Anwendungsgebiet

Da die VCI-DLL asynchrone Ereignisse über Callbackhandler signalisiert, müssen der VCI DLL Funktionsadressen übergeben werden. Da dies z.B. in Visual Basic 6.0 nicht möglich bzw. eingeschränkt möglich ist, wurde mit der VCI Wrapper Komponente eine Möglichkeit geschaffen, VCI Anwendungen unter Visual Basic 6.0 zu schreiben.

Wie bereits erwähnt, legt die VCI Wrapper Komponente seine Funktionalität über automatisierungskompatible Schnittstellen offen und ist somit ideal für Visual Basic geeignet.

1.2 Installation

Die VCI Wrapper Komponente wird mit der VCI V2.14 (oder höher) mitinstalliert. Es ist daher keine separate Installation erforderlich.

2 Programmierhinweise

Die VCI Wrapper Komponente bietet nahezu die gleiche Funktionalität wie die VCI DLL. In diesem Abschnitt wird daher nur auf die Unterschiede bzw. auf die Besonderheiten bei der Programmierung unter Visual Basic eingegangen. Bei den nachfolgenden Beschreibungen wird daher vorausgesetzt, daß Sie das Programmierhandbuch des Virtual CAN Interface bereits durchgearbeitet haben.

2.1 Funktionsbeschreibung

Die VCI Wrapper Komponente unterstützt sämtliche VCI Funktionen mit den folgenden Ausnahmen bzw. Einschränkungen:

- Über die Automatisierungsschnittstelle (IDispatch) können keine Strukturen zurückgegeben werden. Bei VCI-Funktionen die in ihrer ursprünglichen Form Strukturen als Ausgabeparameter besitzen, wird jedes Strukturelement zu einem eigenen Ausgabeparameter. Zu dieser Kategorie zählen die folgenden Funktionen:

VCI_ReadBoardInfo
VCI_ReadBoardStatus
VCI_ReadCanInfo
VCI_ReadCanStatus
VCI_ReadQueObj

Beispiel:

```
typedef struct{
    UINT8 sts;
    UINT8 cpu_load;
}VCI_BRD_STS;

int VCI_ReadBoardStatus( UINT16 board_hdl , VCI_BRD_STS* p_sts );
```



```
HRESULT VCI_ReadBoardStatus( [in] long board_hdl ,
                             [out] unsigned char* status ,
                             [out] unsigned char* cpu_load ,
                             [out,retval] int* ret_code) ;
```

- Die VCI Wrapper Komponente unterstützt keine Callbackhandler. Bei der Funktion **VCI_PrepareBoard** fehlen daher die Parameter zur Angabe von Callback-Handlern.
- Die folgenden Funktionen werden von der VCI Komponente nicht unterstützt:

VCI_PrepareBoardMsg
VCI_PrepareBoardVisBas
VCI_TestBoard

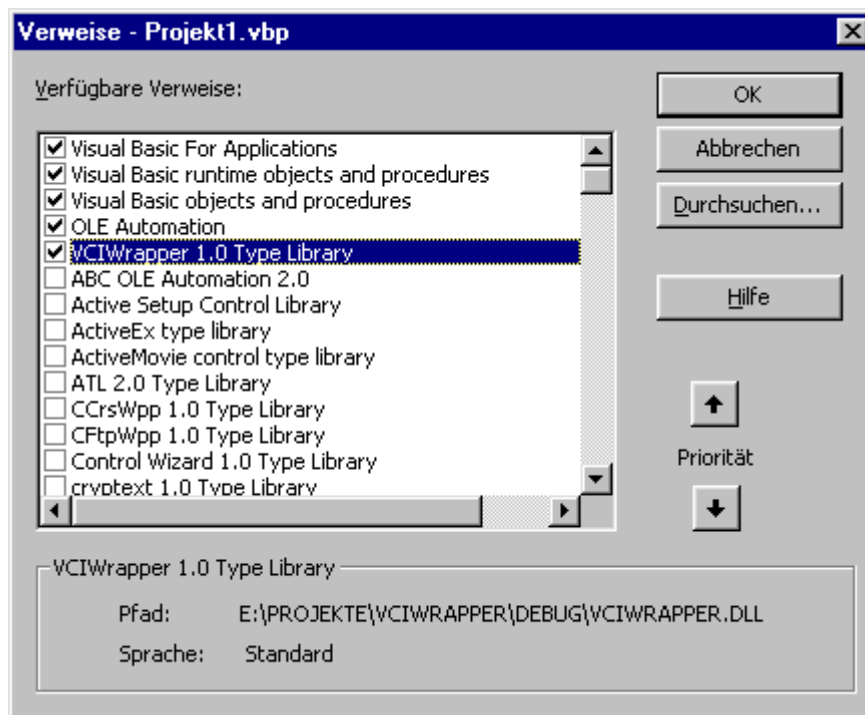
- Der VCI-Wrapper unterstützt zusätzlich ein Objekt mit dem Informationen über die installierte Hardware abgefragt werden kann (ab VCI Version 2.16).

2.2 Initialisierung (Visual Basic)

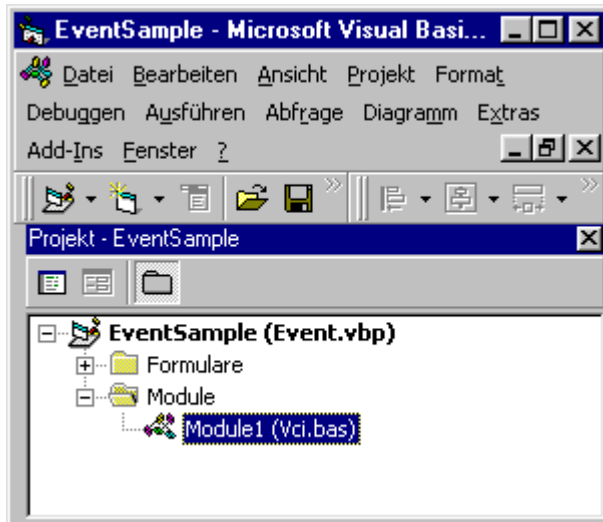
In diesem Abschnitt wird Schritt für Schritt gezeigt wie Sie die VCI Wrapper Komponente in Ihr Visual Basic 6.0 Projekt einbinden und initialisieren.



1. Erstellen Sie das Visual Basic 6.0 Projekt und wählen Sie im **Verweise-Dialogfeld** die Type Library der VCI Wrapper Komponente. Das Verweise-Dialogfeld ist über den Menüpunkt **Projekt/Verweise...** erreichbar.



2. Fügen Sie das Modul **Vci2.bas** in ihr Visual Basic Projekt ein. Vci2.bas enthält symbolische Konstanten für die Verbesserung der Lesbarkeit des Codes.



3. Im zweiten Schritt muß die Definition einer Objekt-Variable für die VCI Wrapper Komponente in ihr Visual Basic Modul eingefügt werden. Diese Deklaration sollte das folgende Aussehen besitzen:

```
Dim WithEvents objVCI As VCIWRAPPERLib.CVCIWrap
```

Hierbei sollte das Schlüsselwort `WithEvents` nicht vergessen werden, da sonst keine COM-Events empfangen werden können. Bei obigem Beispiel ist `objVCI` ein frei wählbarer Variablenname – jedoch muß die Referenz `VCIWRAPPERLib.CVCIWrap` in genau dieser Form angegeben werden.

4. Bevor auf die VCI Wrapper Komponente zugegriffen werden kann, muß die zuvor angesprochene Objekt-Variable initialisiert werden. Dies könnte beispielsweise im Load-Handler des Visual Basic Formulars erfolgen. Die folgende Anweisung ist hierzu notwendig:

```
Set objVCI = New VCIWRAPPERLib.CVCIWrap
```

5. Nach erfolgter Initialisierung der Objekt-Variablen sind die Methoden der VCI Wrapper Komponente verfügbar:

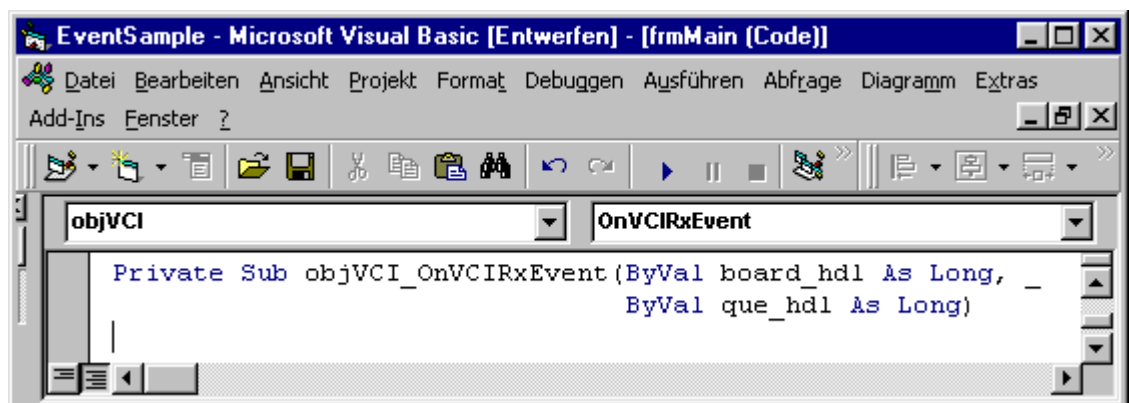
z.B: `intBoardHandle = objVCI.PrepareBoard(VCI_IPCI165, &HDC00, 9)`

2.3 Event Handler (Visual Basic)

Die VCI Wrapper Komponente meldet über COM Events den Empfang von CAN-Nachrichten.



Damit diese COM Events in der Anwendung bearbeitet werden können muß ein entsprechender Event-Handler vorhanden sein. Diesen Event-Handler fügen Sie in Ihre Anwendung ein, indem Sie in der Objekt-Liste die VCI Wrapper Komponente (in diesem Beispiel objVCI) selektieren und anschließend das Element OnVCIRxEvent wählen.



Durch dieses Vorgehen wird automatisch der passende Event-Handler in Ihre Anwendung eingefügt:

```
Private Sub objVCI_OnVCIRxEvent (    ByVal board_hdl As Long,
                                   ByVal que_hdl As Long )
End Sub
```

Mit den Informationen, die von der Event-Prozedur geliefert werden (Board Handle und Queue Handle) können die Empfangenen CAN-Nachrichten mit der Funktion VCI_ReadQueObj aus der jeweiligen Queue ausgelesen werden.



Innerhalb des Event Handlers sollte die Queue so schnell wie möglich leer gelesen werden.

2.4 Lesen von Hardware-Informationen

Ab der VCI 2.16 unterstützt der VCI-Wrapper ein Object zum Auslesen von Informationen über die installierte CAN-Hardware. Um diesen Objekttyp zu verwenden, fügen sie folgende Deklaration in ihr VB-Programm ein:

```
Dim objBoardEnum As VCIWRAPPERLib.BoardCollection
```

Dieses Objekt implementiert eine Sammlung von BoardInfo-Objekten. Diese Sammlung kann mit der „for each“-Anweisung durchlaufen werden:

```
Dim board As VCIWRAPPERLib.IBoardInfo
For Each board In objBoardEnum
    lbBoards.AddItem board.Name & vbTab & board.Info
Next
```

Sie verfügt zusätzlich über die Standard-Eigenschaften „Item“ und „Count“ und kann daher auch folgendermaßen durchlaufen werden:

```
Dim iCount As Integer
For iCount = 1 To objBoardEnum.Count
    lbBoards.AddItem objBoardEnum.Item(iCount).Name
Next
```

Die Eigenschaft „DefaultBoard“ liefert die im ControlPanel eingestellt Default-Hardware oder falls keine Default-Hardware festgelegt wurde, das erste Board in der Sammlung.

Ein BoardInfo-Objekt legt die folgenden Eigenschaften offen:

Type	VCI-interner Board-Typ, wird für die Initialisierung via VCI2_Prepareboard benötigt.
Index	Board-Index (Hardware-Key), wird für die Initialisierung via VCI2_Prepareboard benötigt.
Position	Boardtyp-abhängige Position (z.B. zweite von insgesamt drei installierten iPCI 165 PCI)
Name	Name des Boards (z.B. iPCI 165 PCI)
Manufacturer	Hersteller (z.B. IXXAT Automation GmbH)
Info	Board Informationen (z.B. IRQ, Basisadresse)
AdditionalInfo	zusätzliche Informationen, wird für die Initialisierung via VCI2_Prepareboard benötigt.

Ein BoardInfo-Objekt kann folgendermaßen zur Initialisierung eines Boards verwendet werden:

```
Dim board As IVCIWrap_BoardInfo
Set board = objBoardEnum.Item(iSelected)
iRet = objVCI.VCI2_PrepareBoard(board.Type, _
                                board.Index, _
                                board.AdditionalInfo)
```

2.5 Einschränkungen

Grundsätzlich sollten performancekritische Anwendungen auf die VCI DLL zurückgreifen (mit Visual Basic 6.0 ist dies nicht möglich!), da durch die VCI Wrapper Komponente eine zusätzliche Softwareschicht zwischen Hardware und Anwendung durchlaufen werden muß.

Die VCI Wrapper Komponente ist eine sog. Apartment Threaded Komponente und fügt sich daher ideal in die Architektur von Visual Basic ein. Dies bedeutet für die Programmierung, daß Receive COM Events über Windows Nachrichten an die Anwendung weitergeleitet werden. Für die Visual Basic Anwendung ist es daher äußerst wichtig, daß die Windows Message Queue kontinuierlich abgearbeitet wird.



Bei der Aktivierung eines modalen Dialogs ist die zuvor genannte Bedingung nicht mehr erfüllt. Während der modale Dialog geöffnet ist können keine Receive Events bearbeitet werden und CAN-Nachrichten können verlorengehen. Außerdem sollten in sämtlichen Event Handlern Ihrer Visual Basic Anwendung längere Berechnungen oder dergleichen vermieden werden da dies ebenfalls die Bearbeitung der Windows Message Queue verzögert.

Innerhalb der VCI Wrapper Komponente befinden sich zusätzliche Queues zur Zwischenspeicherung von empfangenen CAN-Nachrichten. Die Größe dieser Queues beträgt 3000 Nachrichten. Dies bedeutet, daß bei verzögertem Auslesen 3000 Nachrichten zwischengespeichert werden können, ohne daß Nachrichten verlorengehen. Ist die Queue gefüllt und eine weitere Nachricht findet keinen Platz mehr, wird die letzte Nachricht der Queue mit einem Queue Overrun Status-Code 0x80 markiert (siehe auch VCI Programmierhandbuch)

2.6 Beispielprogramme

Die VCI Wrapper Komponente enthält Visual Basic Beispiele, die das Senden und Empfangen von CAN-Nachrichten demonstrieren. Es wird sowohl ereignisgesteuertes Auslesen der Queue als auch das Auslesen mittels Polling aufgezeigt.