FTToolbox

Handbuch

Version 2.0

Inhaltsverzeichnis

1 Übersicht

Die FTToolbox ist eine Software für Windows-PCs zur Ansteuerung von fischertechnik Interfaces, die über die parallele Schnittstelle an den Computer angeschlossen werden (Parallelport-Interfaces).

Sowohl ein einzelnes Parallelport-Interface als auch zwei Parallelport-Interfaces kann die FTToolbox ansteuern. Die Ansteuerung von zwei Parallelport-Interfaces ist allerdings nur im sogenannten Master-Slave-Betrieb möglich, d. h., dass die beiden Interfaces gemeinsam an einer einzigen parallelen Schnittstelle des Computers angeschlossen sind. In diesem Fall ist das erste Interface (das sogenannte Master-Interface) mit der parallelen Schnittstelle des Computers verbunden und das zweite Interface (das sogenannte Slave-Interface) ist am ersten Interface angeschlossen.

Der Zweck der FTToolbox besteht darin, in eine eigene, d. h. selbst-erstellte, Applikation eingebunden zu werden, die mittels der FTToolbox Motor-Befehle zu den fischertechnik Interfaces sendet und Schalter-Zustände und Widerstands-Werte von den Interfaces liest. Die Schnittstelle zur eigenen Applikation ist eine .NET-Klassenbibliothek (.NET-Framework Version 2.0).

Die aktuelle Version der FTToolbox kann von der Homepage des Autors Ulrich Keller, www.NachtHacker.de, heruntergeladen werden. Anregungen und Verbesserungsvorschläge können jederzeit an seine eMail-Adresse NachtHacker@gmx.de geschickt werden.

2 Nutzungsbedingungen und Haftungsausschluss

Die FTToolbox ist bei nicht-kommerzieller Nutzung FREEWARE. Jede kommerzielle Nutzung erfordert eine schriftliche Genehmigung durch den Autor Ulrich Keller.

Der Autor der FTToolbox übernimmt keinerlei Haftung für Schäden jeglicher Art, die im Zusammenhang mit der FTToolbox entstehen. Er übernimmt außerdem keinerlei Garantien im Zusammenhang mit der FTToolbox.

3 Hardware-Voraussetzungen

Die FTToolbox unterstützt nur solche fischertechnik Interfaces, die über die parallele Schnittstelle an den Computer angeschlossen werden (Parallelport-Interfaces). Andere fischertechnik Interfaces, die z. B. über die serielle Schnittstelle oder über USB mit dem Computer kommunizieren, werden nicht unterstützt.

Wenn ein einzelnes fischertechnik Interface angesteuert werden soll, dann unterstützt die FTToolbox alle Varianten der Parallelport-Interfaces von fischertechnik.

Wenn zwei fischertechnik Interfaces angesteuert werden sollen, dann unterstützt die FTToolbox nur diejenigen Varianten der Parallelport-Interfaces von fischertechnik, bei denen der Anschluss beider Interfaces an eine einzige parallele Schnittstelle erfolgt (Master-Slave-Betrieb). Bei diesen Varianten der Parallelport-Interfaces wird das erste Interface mit der parallelen Schnittstelle des Computers verbunden und das zweite Interface wird am ersten Interface angeschlossen.

4 Software-Voraussetzungen

Für die FTToolbox sind folgende Software-Voraussetzungen erforderlich:

- Betriebssystem Microsoft Windows
- Microsoft .NET-Framework Version 2.0 oder höher
- Administrator-Rechte bei der Installation und bei der Konfiguration

Die aktuelle Version des .NET-Frameworks kann kostenlos von www.microsoft.de heruntergeladen werden.

5 Installation und Deinstallation

Um die FTToolbox zu installieren, öffnen Sie einfach die Datei FTToolboxSetup.msi. Folgen Sie danach den Anweisungen auf dem Bildschirm.

Die Deinstallation der FTToolbox erfolgt mit den Standard-Mechanismen des Windows-Betriebssystems (Menüpunkt „Software“ bzw. „Programme“ in der Windows-Systemsteuerung).

6 Konfiguration

Zur Konfiguration der FTToolbox können zwei Windows-Umgebungsvariablen angelegt werden. Diese beiden Umgebungsvariablen sind aber kein Muss (seit der Version 2.0 der FTToolbox). Wenn man auf die Umgebungsvariablen verzichtet, dann hat man die Möglichkeit, die entsprechenden Informationen in der Programmierschnittstelle als Parameter anzugeben.

Folgende Umgebungsvariablen werden von der FTToolbox verwendet:

- In der Umgebungsvariablen FT_LPT_ANSCHLUSS_NR wird die Anschlussnummer der parallelen Schnittstelle hinterlegt, an der das fischertechnik Interface oder die fischertechnik Interfaces angeschlossen sind (z. B. 1, 2, 3, …).
- In der Umgebungsvariablen FT_MASTER_SLAVE_BETRIEB wird hinterlegt, ob ein einzelnes fischertechnik Interface angesteuert werden soll (Umgebungsvariablen-Wert 0) oder ob zwei fischertechnik Interfaces im Master-Slave-Betrieb angesteuert werden sollen (Umgebungsvariablen-Wert 1).

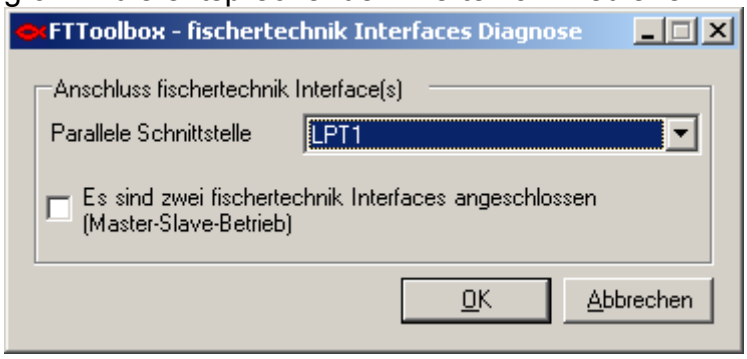
Wenn man diese beiden Umgebungsvariablen verwenden möchte, müssen sie manuell angelegt werden und manuell mit Werten versorgt werden. Dies erfolgt mit den Standard-Mechanismen des Windows-Betriebssystems.

7 Diagnose der Interfaces

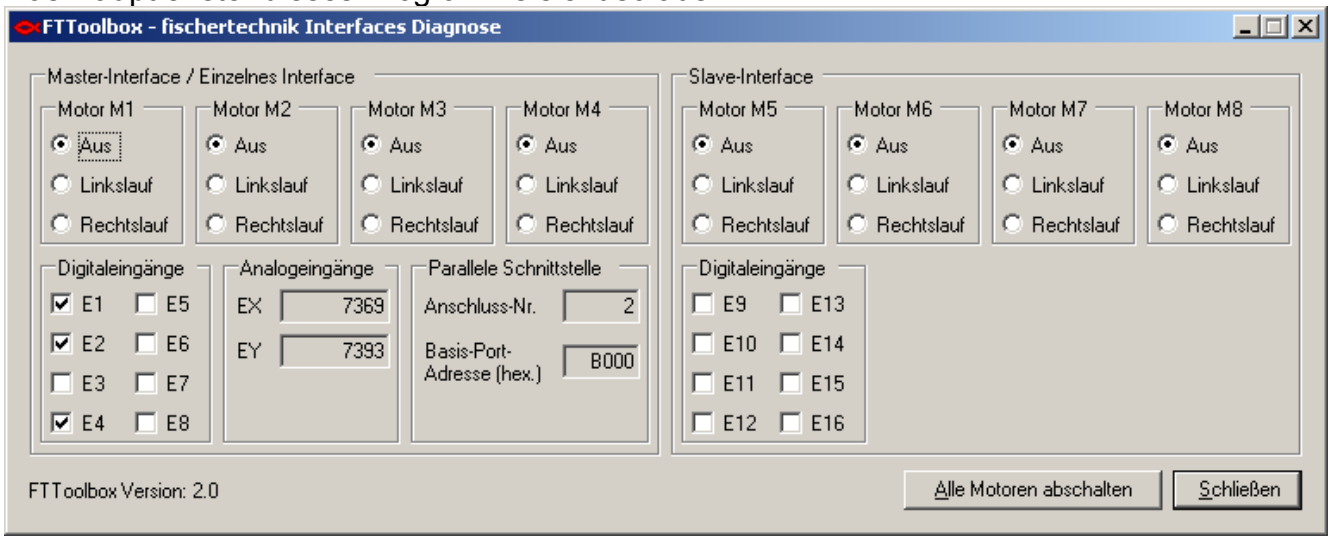
Bei der Installation der FTToolbox wird auch das Programm „fischertechnik Interfaces Diagnose“ installiert; über das Windows-Startmenü kann es gestartet werden.

Das Programm zeigt die Werte aller Eingänge der angeschlossenen fischertechnik Interfaces an. Außerdem können die Motorausgänge per Mausklick geschaltet werden.

Falls die Umgebungsvariablen FT_MASTER_SLAVE_BETRIEB und/oder FT_LPT_ANSCHLUSS_NR nicht angelegt sind oder nicht korrekt mit Werten versorgt sind, dann erfragt das Programm die entsprechenden Werte vom Bediener:



Das Hauptfenster dieses Programms sieht so aus:



8 Programmierschnittstelle

8.1 Allgemeines

Um die Programmierschnittstelle der FTToolbox nutzen zu können, muss in der eigenen Applikation ein Verweis auf die .NET-Klassenbibliothek FTToolbox.dll hinzugefügt werden.

Dieser Verweis kann auf verschiedene Arten realisiert werden:

- Als Verweis auf die Datei FTToolbox.dll im Anwendungsordner der FTToolbox-Software. Dieser Anwendungsordner wurde bei der Installation der FTToolbox durch den Bediener festgelegt und lautet standardmäßig „C:\Programme\FTToolbox“. Während des Hinzufügens des Verweises muss bei der Entwicklungsumgebung Microsoft Visual Studio (und ggf. auch bei anderen Entwicklungsumgebungen) der vollständige Pfad zur .NET-Klassenbibliothek FTToolbox.dll gar nicht angegeben werden. Beim Hinzufügen des Verweises wird die .NET-Klassenbibliothek FTToolbox.dll (unter dem Komponentennamen „FTToolbox“) bereits in der Liste der verfügbaren .NET-Framework-Komponenten angezeigt.
- Als Verweis auf eine Kopie der Datei FTToolbox.dll aus dem Anwendungsordner der FTToolbox-Software. Die Kopie kann z. B. im Arbeitsordner der eigenen Applikation liegen.

Die FTToolbox kann sowohl auf einem 32bit-Windows-Betriebssystem (x86-Architektur) als auch auf einem 64bit-Windows-Betriebssystem (x64-Architektur) verwendet werden. Beim 64bit-Betriebssystem ist zu beachten, dass die .NET-Klassenbibliothek FTToolbox.dll nur mit 32bit ausgeführt werden kann. Dies bedeutet, dass die eigene Applikation auch als 32bit-Applikation realisiert werden muss, die auf einem 64bit-Windows-Betriebssystem dann über das WOW64-Subsystem ausgeführt wird.

Die .NET-Klassenbibliothek FTToolbox.dll stellt mehrere Klassen und Module bereit, die in den folgenden Unterkapiteln beschrieben werden.

8.2 Klasse FTToolboxException

Exception-Klasse. Beim Auftreten von Fehlern in der FTToolbox werden Exceptions dieses Typs geworfen.

```
Public Class FTToolboxException
    Inherits Exception
```

8.3 Modul FTToolboxInfo

Modul mit Methoden zum Ermitteln von Informationen über die FTToolbox.

```
Public Module FTToolboxInfo
```

8.3.1 Methoden

VersionHauptNr

Gibt die Hauptversionsnummer der FTToolbox zurück.

Rückgabewert:
Hauptversionsnummer der FTToolbox.

Bei einem Fehler wird eine FTToolbox-Exception geworfen.

```
Public Function VersionHauptNr() As Integer
```

VersionNebenNr

Gibt die Nebenversionsnummer der FTToolbox zurück.

Rückgabewert:
Nebenversionsnummer der FTToolbox.

Bei einem Fehler wird eine FTToolbox-Exception geworfen.

```
Public Function VersionNebenNr() As Integer
```

8.4 Modul FTInterfaces

Modul mit Methoden zum Ansteuern von fischertechnik Interfaces, die über parallele Schnittstelle an den Computer angeschlossen sind (Parallelport-Interfaces).

Es kann sowohl ein einzelnes Parallelport-Interface angesteuert werden als auch zwei Parallelport-Interfaces, wobei das zweite Interface dann als „Slave“-Interface an das erste „Master“-Interface angeschlossen sein muss.

Wenn das Modul FTInterfaces eine Exception wirft, gilt es danach als "nicht-initialisiert". Bevor dann also an den angeschlossenen Interfaces Motorausgänge (wieder) geschaltet werden können oder Eingänge (wieder) gelesen werden können oder die Diagnose (wieder) angezeigt werden kann, muss zuerst die Methode InitialisiereInterfaces (ggf. erneut) aufgerufen werden.

```
Public Module FTInterfaces
```

8.4.1 Typen

MotorAusgangZustand

Datentyp für die Zustände eines Motorausgangs.

```
Public Enum MotorAusgangZustand
  Aus = 0
  Linkslauf = 1
  Rechtslauf = 2
End Enum
```

8.4.2 Konstanten

UmgebungsVariableFTLPTAnschlussNr

Name der Windows-Umgebungsvariablen, in der die Nummer der parallelen Schnittstelle abgelegt werden kann, an der die fischertechnik Interfaces angeschlossen sind (einzelnes Interface oder Master-Interface mit Slave-Interface).

```
Public Const UmgebungsVariableFTLPTAnschlussNr As String = _
  "FT_LPT_ANSCHLUSS_NR"
```

UmgebungsVariableFTMasterSlaveBetrieb

Name der Windows-Umgebungsvariablen, in der abgelegt werden kann, ob ein einzelnes Interface (Umgebungsvariablen-Wert 0) oder ein Master-Interface mit einem Slave-Interface (Umgebungsvariablen-Wert 1) angeschlossen ist.

```
Public Const UmgebungsVariableFTMasterSlaveBetrieb As String = _
  "FT_MASTER_SLAVE_BETRIEB"
```

MinWertAnalogEingang

Minimaler Wert, der von einem Analogeingang eines fischertechnik Interfaces gelesen werden kann.

```
Public Const MinWertAnalogEingang As Integer = 0
```

MaxWertAnalogEingang

Maximaler Wert, der von einem Analogeingang eines fischertechnik Interfaces gelesen werden kann.

```
Public Const MaxWertAnalogEingang As Integer = 20000
```

8.4.3 Initialisierungs-Methoden

InitialisiereInterfaces (ohne Parameter)

Diese Methode initialisiert den Kommunikationskanal zu den angeschlossenen fischertechnik Interfaces (einzelnes Interface oder Master-Interface mit Slave-Interface).

Die Nummer der parallelen Schnittstelle, an der die fischertechnik Interfaces angeschlossen sind, wird in dieser Methode aus der Windows-Umgebungsvariablen FT_LPT_ANSCHLUSS_NR gelesen. Die Information, ob ein einzelnes Interface oder ein Master-Interface mit einem Slave-Interface angeschlossen ist, wird in dieser Methode aus der Windows-Umgebungsvariablen FT_MASTER_SLAVE_BETRIEB gelesen.

Alle Motorausgänge der angeschlossenen Interfaces werden ausgeschaltet.

Bevor an den angeschlossenen Interfaces Motorausgänge geschaltet werden können oder Eingänge gelesen werden können, muss zuerst diese Methode (oder eine Überladung davon) aufgerufen werden. Außerdem muss diese Methode (oder eine Überladung davon) aufgerufen werden, bevor die Diagnose angezeigt werden kann.

Bei einem Fehler wird eine FTToolbox-Exception geworfen.

```
Public Sub InitialisiereInterfaces()
```

InitialisiereInterfaces (UShort, Boolean)

Diese Methode initialisiert den Kommunikationskanal zu den angeschlossenen fischertechnik Interfaces (einzelnes Interface oder Master-Interface mit Slave-Interface).

Parameter (Eingang):

- pv_lptAnschlussNr — Anschlussnummer der parallelen Schnittstelle, an der die fischertechnik Interfaces angeschlossen sind (einzelnes Interface oder Master-Interface mit Slave-Interface).
- pv_masterSlaveBetrieb — Gibt an, ob ein einzelnes Interface (Wert False) oder ein Master-Interface mit einem Slave-Interface (Wert True) angeschlossen ist.

Alle Motorausgänge der angeschlossenen Interfaces werden in dieser Methode ausgeschaltet.

Bevor an den angeschlossenen Interfaces Motorausgänge geschaltet werden können oder Eingänge gelesen werden können, muss zuerst diese Methode (oder eine Überladung davon) aufgerufen werden. Außerdem muss diese Methode (oder eine Überladung davon) aufgerufen werden, bevor die Diagnose angezeigt werden kann.

Bei einem Fehler wird eine FTToolbox-Exception geworfen.

```
Public Sub InitialisiereInterfaces( _
  ByVal pv_lptAnschlussNr As UShort, _
  ByVal pv_masterSlaveBetrieb As Boolean _
)
```

8.4.4 Diagnose-Methoden

LeseUmgebungsVariableFTLPTAnschlussNr

Ermitteln des Wertes der Windows-Umgebungsvariable FT_LPT_ANSCHLUSS_NR.

Parameter (Ausgang):

- pr_umgebungsVariablenWert: Ermittelter Wert der Windows-Umgebungsvariable FT_LPT_ANSCHLUSS_NR (als Datentyp UShort).

Rückgabewert:

- True: Umgebungsvariablen-Wert wurde erfolgreich ermittelt.
- False: Umgebungsvariablen-Wert konnte nicht ermittelt werden, z. B. weil die Umgebungsvariable nicht existiert oder einen ungültigen Wert enthält.

```
Public Function LeseUmgebungsVariableFTLPTAnschlussNr( _
  ByRef pr_umgebungsVariablenWert As UShort _
) As Boolean
```

LeseUmgebungsVariableFTMasterSlaveBetrieb

Ermitteln des Wertes der Windows-Umgebungsvariable FT_MASTER_SLAVE_BETRIEB.

Parameter (Ausgang):

- pr_umgebungsVariablenWert: Ermittelter Wert der Windows-Umgebungsvariable FT_MASTER_SLAVE_BETRIEB (als Datentyp Boolean).

Rückgabewert:

- True: Umgebungsvariablen-Wert wurde erfolgreich ermittelt.
- False: Umgebungsvariablen-Wert konnte nicht ermittelt werden, z. B. weil die Umgebungsvariable nicht existiert oder einen ungültigen Wert enthält.

```
Public Function LeseUmgebungsVariableFTMasterSlaveBetrieb( _
  ByRef pr_umgebungsVariablenWert As Boolean _
) As Boolean
```

LPTAnschlussNr

Liefert die Anschlussnummer der parallelen Schnittstelle zurück, an der die fischertechnik Interfaces angeschlossen sind (einzelnes Interface oder Master-Interface mit Slave-Interface).
Diese Information wurde beim letzten Aufruf der Methode InitialisiereInterfaces entweder

- aus der Umgebungsvariablen FT_LPT_ANSCHLUSS_NR gelesen

oder

- als Parameter übergeben.

Rückgabewert:
Anschlussnummer der parallelen Schnittstelle, an der die fischertechnik Interfaces angeschlossen sind (einzelnes Interface oder Master-Interface mit Slave-Interface).

Bevor diese Methode verwendet werden kann, muss die Methode InitialisiereInterfaces aufgerufen werden!

Bei einem Fehler wird eine FTToolbox-Exception geworfen.

```
Public Function LPTAnschlussNr() As UShort
```

MasterSlaveBetrieb

Liefert zurück, ob ein einzelnes Interface oder ein Master-Interface mit einem Slave-Interface angeschlossen ist.
Diese Information wurde beim letzten Aufruf der Methode InitialisiereInterfaces entweder

- aus der Umgebungsvariablen FT_MASTER_SLAVE_BETRIEB ermittelt

oder

- als Parameter übergeben.

Rückgabewert:

- True: Ein Master-Interface mit einem Slave-Interface ist angeschlossen.
- False: Ein einzelnes Interface ist angeschlossen.

Bevor diese Methode verwendet werden kann, muss die Methode InitialisiereInterfaces aufgerufen werden!

Bei einem Fehler wird eine FTToolbox-Exception geworfen.

```
Public Function MasterSlaveBetrieb() As Boolean
```

LPTAnschlussBasisPortAdresse

Liefert die Basis-Adresse der Computer-internen Schnittstellenports zurück, über die mit der parallelen Schnittstelle kommuniziert werden, an der die fischertechnik Interfaces angeschlossen sind (einzelnes Interface oder Master-Interface mit Slave-Interface).

Rückgabewert:
Basis-Adresse der Computer-internen Schnittstellenports, über die mit der parallelen Schnittstelle kommuniziert werden.

Bevor diese Methode verwendet werden kann, muss die Methode InitialisiereInterfaces aufgerufen werden!

Bei einem Fehler wird eine FTToolbox-Exception geworfen.

```
Public Function LPTAnschlussBasisPortAdresse() As UShort
```

DiagnoseAnzeigen

Modales Anzeigen eines Fensters zur Diagnose der fischertechnik Interfaces (einzelnes Interface oder Master-Interface mit Slave-Interface).

Parameter (Eingang):

- pv_ownerForm — Owner-Form für das anzuzeigende modale Fenster.
- pv_iconAnzeigen — Gibt an, ob im Fenstertitel ein Fisch-Icon angezeigt werden soll (True) oder nicht (False).
- pv_fensterTitel — Fenstertitel, der verwendet werden soll.
- pv_minimierenErlaubt — Gibt an, ob das Fenster durch den Bediener minimiert werden darf (True) oder nicht (False).
- pv_inTaskleisteAnzeigen — Gibt an, ob das Fenster in der Task-Leiste erscheinen soll (True) oder nicht (False).

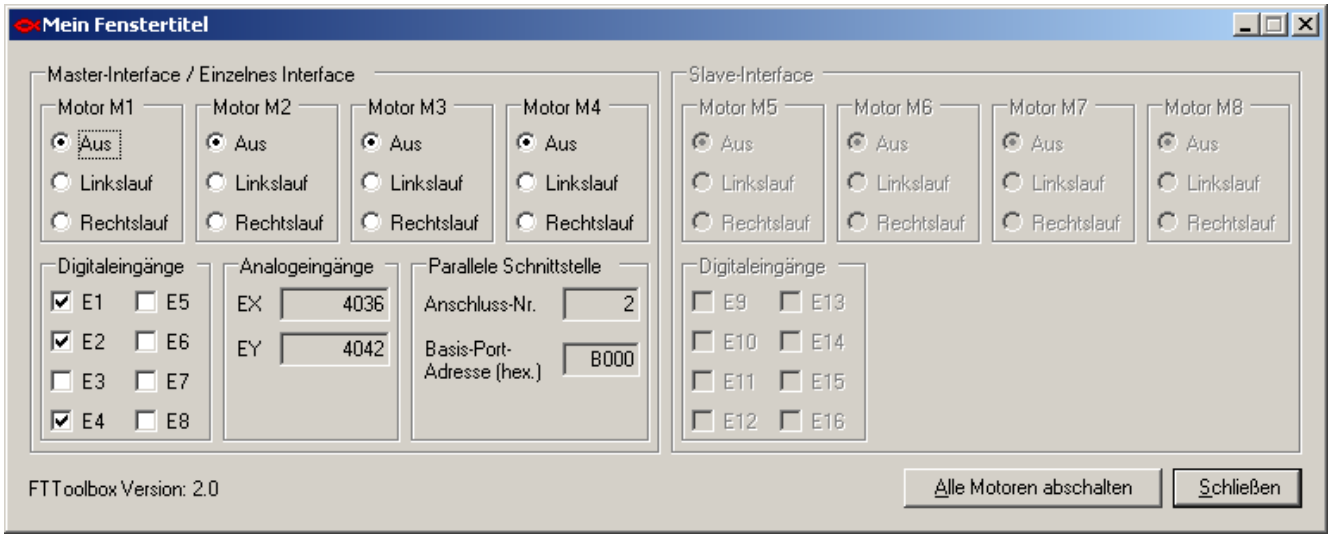
In diesem Fenster werden die Werte aller Eingänge der angeschlossenen fischertechnik Interfaces angezeigt. Außerdem können in diesem Fenster die Motorausgänge per Mausklick geschaltet werden.

Bevor diese Methode verwendet werden kann, muss die Methode InitialisiereInterfaces aufgerufen werden!

Bei einem Fehler wird eine FTToolbox-Exception geworfen.

```
Public Sub DiagnoseAnzeigen( _
  ByVal pv_ownerForm As Windows.Forms.IWin32Window, _
  ByVal pv_iconAnzeigen As Boolean, _
  ByVal pv_fensterTitel As String, _
  ByVal pv_minimierenErlaubt As Boolean, _
  ByVal pv_inTaskleisteAnzeigen As Boolean _
)
```

Screenshot:



8.4.5 Methoden zum Schalten der Motorausgänge

SchalteMotor

Schalten eines Motors bzw. eines Motorausgangs an den fischertechnik Interfaces (einzelnes Interface oder Master-Interface mit Slave-Interface).

Parameter (Eingang):

- pv_motorNr — Nummer des Motors (1..4 bei einem einzelnen Interface, 1..8 bei einem Master-Interface mit Slave-Interface)
- pv_motorZustand — Zustand, in den der Motor geschaltet werden soll

Bevor diese Methode verwendet werden kann, muss die Methode InitialisiereInterfaces aufgerufen werden!

Bei einem Fehler wird eine FTToolbox-Exception geworfen.

```
Public Sub SchalteMotor( _
  ByVal pv_motorNr As Integer, _
  ByVal pv_motorZustand As MotorAusgangZustand _
)
```

SchalteAlleMotoren

Schalten aller Motoren bzw. aller Motorausgänge an den fischertechnik Interfaces (einzelnes Interface oder Master-Interface mit Slave-Interface).

Parameter (Eingang):

- pv_motorZustand — Eindimensionaler Array mit allen Zuständen, in die die Motoren geschaltet werden sollen.
  Wenn ein einzelnes Interface angeschlossen ist, dann muss der Array 4 Elemente enthalten (Indizierung: 0..3).
  Wenn ein Master-Interface mit einem Slave-Interface angeschlossen ist, dann muss der Array 8 Elemente enthalten (Indizierung: 0..7).

Bevor diese Methode verwendet werden kann, muss die Methode InitialisiereInterfaces aufgerufen werden!

Bei einem Fehler wird eine FTToolbox-Exception geworfen.

```
Public Sub SchalteAlleMotoren( _
  ByVal pv_motorZustand As MotorAusgangZustand() _
)
```

AbschaltenAlleMotoren

Abschalten aller Motoren bzw. aller Motorausgänge an den fischertechnik Interfaces (einzelnes Interface oder Master-Interface mit Slave-Interface).

Bevor diese Methode verwendet werden kann, muss die Methode InitialisiereInterfaces aufgerufen werden!

Bei einem Fehler wird eine FTToolbox-Exception geworfen.

```
Public Sub AbschaltenAlleMotoren()
```

8.4.6 Methoden zum Lesen der Digitaleingänge

LeseDigitalEingang

Lesen des Zustandes eines Digitaleingangs von den fischertechnik Interfaces (einzelnes Interface oder Master-Interface mit Slave-Interface).

Parameter (Eingang):

- pv_digitalEingangNr — Nummer des Digitaleingangs (1..8 bei einem einzelnen Interface, 1..16 bei einem Master-Interface mit Slave-Interface)

Rückgabewert:
Zustand des Digitaleingangs (True = Schalter ist geschlossen, False = Schalter ist geöffnet)

Bevor diese Methode verwendet werden kann, muss die Methode InitialisiereInterfaces aufgerufen werden!

Bei einem Fehler wird eine FTToolbox-Exception geworfen.

```
Public Function LeseDigitalEingang( _
  ByVal pv_digitalEingangNr As Integer _
) As Boolean
```

LeseAlleDigitalEingaenge

Lesen des Zustandes aller Digitaleingänge von den fischertechnik Interfaces (einzelnes Interface oder Master-Interface mit Slave-Interface).

Rückgabewert:
Eindimensionaler Array mit den Zuständen aller Digitaleingänge jeweils als Boolscher Wert (True = Schalter ist geschlossen, False = Schalter ist geöffnet).
Wenn ein einzelnes Interface angeschlossen ist, enthält der Array 8 Elemente (Indizierung: 0..7).
Wenn ein Master-Interface mit einem Slave-Interface angeschlossen ist, enthält der Array 16 Elemente (Indizierung: 0..15).

Bevor diese Methode verwendet werden kann, muss die Methode InitialisiereInterfaces aufgerufen werden!

Bei einem Fehler wird eine FTToolbox-Exception geworfen.

```
Public Function LeseAlleDigitalEingaenge() As Boolean()
```

8.4.7 Methoden zum Lesen der Analogeingänge

LeseAnalogEingang

Lesen des Wertes eines Analogeingangs von den fischertechnik Interfaces (einzelnes Interface oder Master-Interface mit Slave-Interface).

Parameter (Eingang):

- pv_analogEingangNr — Nummer des Analogeingangs, dessen Wert gelesen werden soll (1 für Analogeingang EX oder 2 für Analogeingang EY).
- pv_anzahlMessungen — Anzahl der Messungen des Analogeingang-Wertes (1 bis max. 10).
  Je höher die Anzahl, desto genauer ist der Wert des Analog-Eingangs, den diese Methode zurückgibt.
  Allerdings wächst die Laufzeit dieser Methode mit steigender Anzahl der Messungen.

Rückgabewert:
Wert des Analog-Eingangs. Der Wert hat mindestens den Wert der Konstanten MinWertAnalogEingang und maximal den Wert der Konstanten MaxWertAnalogEingang.

Wenn ein einzelnes Interface angeschlossen ist, können die beiden Analogeingänge dieses Interfaces gelesen werden.

Wenn ein Master-Interface mit einem Slave-Interface angeschlossen ist, können nur die beiden Analogeingänge des Master-Interfaces gelesen werden.

Bevor diese Methode verwendet werden kann, muss die Methode InitialisiereInterfaces aufgerufen werden!

Bei einem Fehler wird eine FTToolbox-Exception geworfen.

```
Public Function LeseAnalogEingang( _
  ByVal pv_analogEingangNr As Integer, _
  ByVal pv_anzahlMessungen As Integer _
) As Integer
```

8.5 Änderungen an der Programmierschnittstelle seit Version 1.x

Die Version 2.0 der FTToolbox ist *nicht* voll abwärtskompatibel zu den Versionen 1.x der FTToolbox. Wenn ein Programm, das bisher eine Version 1.x der FTToolbox verwendet hat, nun die FTToolbox 2.0 verwenden soll, dann sind kleinere Änderungen an dem Programm notwendig.

Die Unterschiede zwischen den Programmierschnittstellen der FTToolbox Versionen 1.x und der Version 2.0 sind im Folgenden aufgelistet:

FTToolbox Versionen 1.x	**FTToolbox Version 2.0**	**Änderung**	**Änderungsgrund**
Modul FTToolbox	Modul FTInterfaces	Umbenennung	Vermeiden von Namens-konflikten zwischen Namespace FTToolbox und Modul FTToolbox
Unter-Klasse FTToolboxException im Modul FTToolbox	Klasse FTToolboxException	Verschiebung bzgl. Hierarchieebene	Vereinfachung der Hierarchien
Methoden VersionHauptNr und VersionNebenNr im Modul FTToolbox	*Methoden VersionHauptNr und VersionNebenNr im Modul FTToolboxInfo*	*Verschieben der Methoden von Modul FTToolbox (neu: FTInterfaces) in Modul FTToolboxInfo*	*Dezentralisieren des Klassen-Modells*
Methode FormFTInterfaceDiagnoseAnzeigen	*Methode DiagnoseAnzeigen*	*Umbenennung und neuer Parameter pv_minimierenErlaubt*	*Methodenname war sehr lang, Funktions-Erweiterung*

Für die kursiv markierten Änderungen existieren in der FTToolbox Version 2.0 Vorkehrungen zur Wahrung der Abwärtskompatibilität. Das bedeutet, dass Programme, die die FTToolbox Version 1.x verwenden, bzgl. der kursiv markierten Änderungen nicht zwingend umgestellt werden müssen.

9 Beispiele zur Programmierung mit der FTToolbox

Die FTToolbox kann mit jeder .NET-Programmiersprache verwendet werden. Im Folgenden ein kleines Beispiel zur Verwendung der FTToolbox, einmal in der Programmiersprache Visual Basic .NET und einmal in der Programmiersprache C#.

9.1 Visual Basic .NET

```
Imports System.Threading
Imports FTToolbox

' ...

' Allgemeine Fehlerbehandlung
Try

  ' fischertechnik Interfaces initialisieren
  FTInterfaces.InitialisiereInterfaces()

  ' Sicherheitsabfrage
  If MessageBox.Show("M1 läuft gleich los...", _
                     "Sicherheitsabfrage", _
                     MessageBoxButtons.OKCancel, _
                     MessageBoxIcon.Question, _
                     MessageBoxDefaultButton.Button2 _
                    ) <> Windows.Forms.DialogResult.OK Then
    Return
  End If

  ' M1 auf Linkslauf stellen
  FTInterfaces.SchalteMotor(1, FTInterfaces.MotorAusgangZustand.Linkslauf)

  ' Warten, bis E1 auf "an" geht
  While Not FTInterfaces.LeseDigitalEingang(1)
    Thread.Sleep(50)  ' Damit die CPU-Auslastung nicht auf 100% geht
  End While

  ' M1 wieder abschalten
  FTInterfaces.SchalteMotor(1, FTInterfaces.MotorAusgangZustand.Aus)

Catch ex As Exception  ' Allgemeine Fehlerbehandlung

  ' Fehlermeldung ausgeben
  MessageBox.Show(ex.Message, _
                  "Fehler", _
                  MessageBoxButtons.OK, _
                  MessageBoxIcon.Stop _
                 )

End Try  ' Allgemeine Fehlerbehandlung
```

9.2 C#

```
using System;
using System.Windows.Forms;
using System.Threading;
using FTToolbox;

// ...

try  // Allgemeine Fehlerbehandlung
{
  // fischertechnik Interfaces initialisieren
  FTInterfaces.InitialisiereInterfaces();

  // Sicherheitsabfrage
  if (MessageBox.Show("M1 läuft gleich los...",
                      "Sicherheitsabfrage",
                      MessageBoxButtons.OKCancel,
                      MessageBoxIcon.Question,
                      MessageBoxDefaultButton.Button2
                     ) != DialogResult.OK
     )
  {
      return;
  }

  // M1 auf Linkslauf stellen
  FTInterfaces.SchalteMotor(1, FTInterfaces.MotorAusgangZustand.Linkslauf);

  // Warten, bis E1 auf "an" geht
  while (!FTInterfaces.LeseDigitalEingang(1))
  {
    Thread.Sleep(50);  // Damit die CPU-Auslastung nicht auf 100% geht
  }

  // M1 wieder abschalten
  FTInterfaces.SchalteMotor(1, FTInterfaces.MotorAusgangZustand.Aus);
}
catch (Exception ex)  // Allgemeine Fehlerbehandlung
{
  // Fehlermeldung ausgeben
  MessageBox.Show(ex.Message,
                  "Fehler",
                  MessageBoxButtons.OK,
                  MessageBoxIcon.Stop
                 );
}  // Allgemeine Fehlerbehandlung
```