

# **SIEMENS**

**SIMATIC**

**S7ProSim V5.4**

**COM-Objekt**

**Benutzerhandbuch**

**Ausgabe: 01/2007**

# Copyright und Sicherheitsrichtlinien

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden am Produkt und an den angeschlossenen Geräten beachten müssen. Diese Hinweise sind im Handbuch durch ein Warndreieck hervorgehoben und je nach Gefährdungsgrad folgendermaßen dargestellt:



## **Gefahr**

bedeutet, dass Tod oder schwere Körperverletzung eintreten werden, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



## **Warnung**

bedeutet, dass Tod oder schwere Körperverletzung eintreten können, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



## **Vorsicht**

mit Warndreieck bedeutet, dass Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

## **Vorsicht**

ohne Warndreieck bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

## **Achtung**

ohne Warndreieck bedeutet, dass ein unerwünschter Zustand eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

## Qualifiziertes Personal

Inbetriebsetzung und Betrieb des Geräts/Systems dürfen nur zusammen mit diesem Handbuch vorgenommen werden. Nur qualifiziertes Personal darf diese Geräte installieren und bedienen. Qualifiziertes Personal sind Personen, die die Berechtigung haben, Geräte, Systeme und Stromkreise gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu erden und zu kennzeichnen.

## Bestimmungsgemäßer Gebrauch

Beachten Sie Folgendes:



## **Warnung**

Dieses Gerät und seine Komponenten dürfen nur für die im Katalog und in der technischen Beschreibung vorgesehenen Einsatzfälle und nur in Verbindung mit von Siemens empfohlenen bzw. zugelassenen Fremdgeräten und -Komponenten verwendet werden.

Der einwandfreie und sichere Betrieb des Produkts setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung und Montage sowie sorgfältige Bedienung und Instandhaltung voraus.

## Warenzeichen

Siemens® und SIMATIC® sind eingetragene Warenzeichen der SIEMENS AG.

STEP 7™ und S7™ sind Warenzeichen der SIEMENS AG.

Microsoft®, Windows®, Windows 95®, Windows 98®, Windows NT®, Windows ME®, and Windows 2000® sind eingetragene Warenzeichen der Microsoft Corporation.

## Copyright Siemens AG, 2007

### **Alle Rechte vorbehalten.**

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts ist nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte vorbehalten, insbesondere für den Fall der Patenterteilung oder GM-Eintragung.

## Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, und notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten. Für Verbesserungsvorschläge sind wir dankbar.

# Vorwort

S7ProSim bietet programmatischen Zugriff auf die Schnittstelle zum simulierten Zielsystem in S7-PLCSIM. Mit S7ProSim können Sie Software schreiben, mit der Sie Tätigkeiten wie das Umschalten des Schlüsselschalters des simulierten Zielsystems, die zyklusweise Ausführung des Steuerungsprogramms, das Lesen und Schreiben von Steuerungswerten und viele andere Arbeitsschritte ausführen können.

## Leserkreis

Dieses Handbuch wendet sich an Entwickler, Programmierer und Wartungspersonal mit Kenntnissen und Erfahrung in den Bereichen speicherprogrammierbare S7-Steuerungen und Entwicklung von Software in Visual Basic (6.0 oder .NET) bzw. Visual C++ (6.0 oder .NET).

## Umfang des Handbuchs

Dieses Dokument beschreibt die Funktionen und die Bedienung von S7ProSim V5.4.

## Weitere Handbücher

Weitere Informationen finden Sie in der Online-Hilfe von STEP 7 und S7-PLCSIM sowie in den folgenden Handbüchern:

- *Programmieren mit STEP 7, Handbuch.* Dieses Handbuch liefert die grundlegenden Informationen zum Entwerfen und Programmieren von Steuerungsprogrammen. Arbeiten Sie mit diesem Handbuch, wenn Sie mit der Automatisierungssoftware STEP 7 ein Steuerungsprogramm entwerfen.
- *Systemsoftware für S7-300/400 System- und Standardfunktionen, Referenzhandbuch.* In diesem Handbuch finden Sie Beschreibungen von Systemfunktionen, Organisationsbausteinen und Standardfunktionen, mit denen Sie beim Entwickeln eines Steuerungsprogramms arbeiten.
- *Erste Schritte und Übungen mit STEP 7, Getting Started.* In diesem Handbuch wird beschrieben, wie Sie mit der Automatisierungssoftware STEP 7 arbeiten. Das Handbuch verschafft Ihnen einen Überblick über die Vorgehensweise beim Konfigurieren eines Zielsystems und beim Entwickeln von Steuerungsprogrammen.
- *S7-PLCSIM - Testen Ihres S7-CPU Programms.* In diesem Handbuch werden Bedienoberfläche und Funktionsweise von S7-PLCSIM, dem S7-SPS-Simulator, beschrieben.

Diese und andere Handbücher finden Sie, wenn Sie auf dem Computer, auf dem STEP 7 installiert ist, im Windows-Startmenü den Menübefehl **Start > Simatic > Dokumentation** wählen.

## Zusätzliche Unterstützung

Falls Sie technische Fragen haben oder Informationen zum Bestellen dieses Produkts bzw. zu Schulungen benötigen, wenden Sie sich bitte an Ihre Siemens-Vertretung.

### Nord- und Südamerika

Telefon: +1 (800) 333-7421

Fax: +1 (423) 262-2200

simatic.hotline@siemens.com

### Europa und Afrika

Telefon: +49 (0) 180 5050 222

Fax: +49 (0) 180 5050 223

adsupport@siemens.com

### Asien und Pazifik-Region

Telefon: +86 10 64 75 75 75

Fax: +86 10 64 74 74 74

adsupport.asia@siemens.com



# Inhaltsverzeichnis

<b>Übersicht über S7ProSim .....</b>	<b>7</b>
Aufnehmen eines S7ProSim COM-Objekts in Ihr Projekt.....	7
Programmieren einer Schnittstelle zu S7-PLCSIM mit S7ProSim .....	8
<b>Methoden .....</b>	<b>11</b>
BeginScanNotify .....	13
Connect .....	14
Continue .....	15
Disconnect.....	16
EndScanNotify .....	17
ExecuteNmsScan .....	18
ExecuteNScans .....	19
ExecuteSingleScan.....	20
GetPauseState .....	21
GetScanMode.....	22
GetStartUpSwitch .....	23
GetState.....	24
HotStartWithSavedValues .....	25
Pause.....	26
ReadDataBlockValue.....	27
ReadFlagValue .....	28
ReadOutputImage .....	29
ReadOutputPoint .....	30
SavePLC.....	31
SetScanMode .....	32
SetStartUpSwitch.....	33
SetState .....	34
StartPLCSim .....	35
WriteDataBlockValue.....	36
WriteFlagValue .....	37
WriteInputImage .....	38
WriteInputPoint .....	39
<b>Ereignisse .....</b>	<b>41</b>
ConnectionError.....	42
PauseStateChanged.....	43
PLCSimStateChanged.....	44
ScanFinished.....	45
ScanModeChanged .....	46

<b>Typdefinitionen</b> .....	<b>47</b>
CPURunMode.....	48
ImageDataTypeConstants .....	49
PauseStateConstants .....	50
PointDataTypeConstants.....	51
RestartSwitchPosition.....	52
ScanModeConstants .....	53
tagPauseState .....	54
ScanInfo-Konstanten .....	55
Fehlerausgabecodes .....	57
<b>Index</b> .....	<b>59</b>

# Übersicht über S7ProSim

S7ProSim bietet ein COM-Objekt für den programmatischen Zugriff auf die Oberfläche der Prozesssimulation in S7-PCLSIM. Sie können S7ProSim in jeder Anwendung einsetzen, die COM-Objekte für den Anschluss an die S7-PLCSIM Prozesssimulation unterstützt.

In diesem Online-Dokument wird beschrieben, wie Sie S7ProSim in eine Anwendung aufnehmen. Außerdem werden Leistungsmerkmale, Oberfläche und Funktionsweise von S7ProSim beschrieben und die Software-Objektdefinitionen von Methoden und Ereignissen erläutert.

## Aufnehmen eines S7ProSim COM-Objekts in Ihr Projekt

Wenn Sie ein S7ProSim COM-Objekt in Ihrem Projekt verwenden möchten, fügen Sie im Projekt eine Referenz auf das Objekt hinzu. Die Vorgehensweise zum Ergänzen einer Projektreferenz richtet sich nach Ihrer Programmierumgebung. In Microsoft Visual Basic (6.0 oder .NET) beispielsweise gehen Sie folgendermaßen vor, um eine Referenz auf ein S7ProSim COM-Objekt zu ergänzen:

1. Wählen Sie den Menübefehl **Projekt > Referenzen** oder **Projekt > Referenz hinzufügen**.
2. Wählen Sie im Dialogfeld "Referenzen" das Kontrollkästchen für das Siemens S7ProSim COM-Objekt. (In Visual Basic .NET befindet sich diese Auswahl im Register "COM" des Dialogfelds "Referenzen".)
3. Bestätigen Sie mit "OK".

Nachdem Sie die Projektreferenz aufgenommen haben, können Sie sich im Objekt-Browser die Methoden und Ereignisse des S7ProSim COM-Objekts ansehen. Wählen Sie im Objekt-Browser im aufklappbaren Listenfeld der Bibliotheken die Option S7PROSIMLib. Die Klasse S7ProSim enthält die Methoden und Ereignisse, mit denen Sie eine Schnittstelle zu S7-PLCSIM programmieren können.

In Microsoft Visual Studio C++ V6.0 und in Microsoft Visual C++ .NET befolgen Sie die für die jeweilige Programmierumgebung entsprechende Vorgehensweise zum Hinzufügen eines COM-Objekts.

## Programmieren einer Schnittstelle zu S7-PLCSIM mit S7ProSim

Damit Sie mit S7ProSim die in S7-PLCSIM simulierte Steuerung programmatisch bedienen können, gehen Sie folgendermaßen vor:

- Nehmen Sie das Siemens S7ProSim COM-Objekt in das Projekt auf.
- Fügen Sie zu Ihrem Projekt eine Deklaration für S7ProSim hinzu.

### Beispiel: Visual Basic 6.0

```
Option Explicit
Private WithEvents S7ProSim As S7PROSIMLib.S7ProSim
...
Private Sub Form_Load()
Set S7ProSim = New S7PROSIMLIB.S7ProSim
...
End Sub
```

### Beispiel: Visual Basic .NET

```
Private WithEvents S7ProSim As New S7PROSIMLib.S7ProSim
```

### Beispiel: Visual C++ 6.0

```
// die ProSim-Bibliothek/tlb ist in der dll

#import <S7wspmx.dll> named_guids, no_namespace//, raw_interfaces_only

class ProSimWrapper
{
public:
    ProSimWrapper() : m_pProSim(OLESTR("S7wspmx.S7ProSim"), NULL,
CLSCTX_INPROC_SERVER)
    {}; // smartptr wird automatisch im Stack erstellt, wenn die Anwendung
        // gestartet wird

    virtual ~ProSimWrapper()
    {}; // keine Implementierung, smartptr wird automatisch freigegeben,
        // wenn die Anwendung geschlossen wird

    IS7ProSim * GetPtr()
    {
        return m_pProSim;
    };

// Attribute
protected:
    // IProSimPtr ist ein CComPtr (smart ptr) für die IProSim-Schnittstelle
    // kommt aus der dll-Datei von #import
    // CoCreateInstance wird automatisch für das Objekt ptr im Konstruktor
    // dieser Klasse aufgerufen
    // Freigabe von ptr wird automatisch vom Destruktor dieser Klasse
    // aufgerufen
    IS7ProSimPtr    m_pProSim;
};
```

### Beispiel: C#

```
using S7PROSIMLib;
...
private S7ProSim ps;
```



- Programmieren Sie in Visual Basic Ereignis-Handler für die S7ProSim Ereignisse. Ereignis-Handler sind in Visual C++ nicht erforderlich. In jeden Ereignis-Handler können Sie beliebigen Code für Ihre Anwendung einfügen.

**Beispiel: Visual Basic 6.0**

```
Private Sub S7ProSim_PauseStateChanged(ByVal NewState As String)
    DoEvents
    ...
End Sub

Private Sub S7ProSim_ScanFinished(ByVal ScanInfo As Variant)
    DoEvents
    ...
End Sub

Private Sub S7ProSim_PLCSimStateChanged(ByVal NewState As String)
    DoEvents
    ...
End Sub

Private Sub S7ProSim_ConnectionError(ByVal ControlEngine As String, ByVal
error As Long)
    DoEvents
    MsgBox "Verbindungsfehler"
End Sub

Private Sub S7ProSim_ScanModeChanged(ByVal NewState As String)
    DoEvents
    ...
End Sub
```

**Hinweis**

In Visual Basic .NET ist der Aufruf "DoEvents" nicht erforderlich.

- Nehmen Sie in Ihre Anwendung Befehlsschaltflächen, Textfelder und andere Objekte auf, die für den Zugriff auf verschiedene S7ProSim Methoden erforderlich sind. Programmieren Sie den Code für jeden Befehlsschaltflächen-Handler, um S7ProSim Methoden aufzurufen und entsprechende Werte für Textfelder gemäß Ihrer Anwendung einzustellen.



# Methoden

◆ <b>BeginScanNotify</b>	Registriert S7ProSim für Rückrufe von der Steuerung. Die Ereignisse ScanFinished und PLCSimStateChanged werden gesendet, wenn diese Ereignisse auftreten.
◆ <b>Connect</b>	Verbindet S7ProSim mit S7-PLCSIM.
◆ <b>Continue</b>	Setzt eine Simulation fort, die angehalten wurde.
◆ <b>Disconnect</b>	Trennt die Verbindung zwischen S7ProSim und S7-PLCSIM.
◆ <b>EndScanNotify</b>	Hebt die Registrierung von S7ProSim für Rückrufe von der Steuerung auf. Die Ereignisse ScanFinished und PLCSimStateChanged werden nicht gesendet.
◆ <b>ExecuteNmsScan</b>	Zwingt S7-PLCSIM, für eine bestimmte Zeitdauer (Nms) Zyklen auszuführen, wobei nicht auf das Ende des aktuellen Zyklus gewartet wird. Ist die Zyklusbenachrichtigung aktiviert, wird dem Programm mitgeteilt, wenn S7-PLCSIM die Zyklen beendet hat.
◆ <b>ExecuteNScans</b>	Zwingt S7-PLCSIM, eine bestimmte Anzahl Zyklen auszuführen, wobei nicht auf das Ende des aktuellen Zyklus gewartet wird. Ist die Zyklusbenachrichtigung aktiviert, wird dem Programm mitgeteilt, wenn S7-PLCSIM die Zyklen beendet hat.
◆ <b>ExecuteSingleScan</b>	Zwingt S7-PLCSIM, einen Zyklus auszuführen, wobei nicht auf das Ende des aktuellen Zyklus gewartet wird. Ist die Zyklusbenachrichtigung aktiviert, wird dem Programm mitgeteilt, wenn S7-PLCSIM den Zyklus beendet hat.
◆ <b>GetPauseState</b>	Gibt den aktuellen Pausenzustand von S7-PLCSIM aus.
◆ <b>GetScanMode</b>	Meldet den Bearbeitungsmodus von S7-PLCSIM.
◆ <b>GetStartUpSwitch</b>	Holt die AnlaufEinstellung (Wiederanlauf, Warmstart oder Kaltstart) für S7-PLCSIM.
◆ <b>GetState</b>	Gibt einen String mit der aktuellen Schüsselschalterposition von S7-PLCSIM (RUN, RUN-P oder STOP) aus.
◆ <b>HotStartWithSavedValues</b>	Setzt einen booleschen Wert, um anzugeben, ob S7-PLCSIM bei einem Start im Zustand HotStart gespeicherte E/A-Werte der Peripherie laden soll. Damit S7-PLCSIM beim Start Peripherie-E/A lädt, muss der Anwender HotStartWithSavedValues mit dem Wert TRUE aufrufen, das SPS-Programm speichern ( SavePLC) und den Anlaufzustand für S7-PLCSIM setzen auf HotStart ( SetStartUpSwitch). Beim Neustart von S7-PLCSIM werden die Peripherie-E/A geladen.
◆ <b>Pause</b>	Hält eine Simulation an.
◆ <b>ReadDataBlockValue</b>	Liest ein bestimmtes Bit, Byte, Wort oder Doppelwort aus dem Speicherbereich DB von S7-PLCSIM.
◆ <b>ReadFlagValue</b>	Liest ein bestimmtes Bit, Byte, Wort oder Doppelwort aus dem Speicherbereich der Merker (M) von S7-PLCSIM.
◆ <b>ReadOutputImage</b>	Liest Elemente aus dem Peripheriebereich der Ausgänge (Speicherbereich PA) von S7-PLCSIM.
◆ <b>ReadOutputPoint</b>	Liest ein bestimmtes Bit (Boolean), ein Byte (Byte), ein 2-Byte-Wort (Integer) oder ein 4-Byte-Wort (Long) aus dem Peripheriebereich der Ausgänge (Speicherbereich PA).

- ◆ **SavePLC** Speichert die aktuellen simulierten SPS-Daten in einer Datei.  
Die Daten, die gespeichert werden, bestehen aus dem Programm, der Hardware-Konfiguration, der Schlüsselschalterposition wie im Unterfenster "CPU", der Art der Programmbearbeitung (automatisch oder zyklusweise), dem E/A-Zustand, Zeitwerten, symbolischen Adressen und der Netzeinstellung (ein oder aus).
- ◆ **SetScanMode** Stellt den Bearbeitungsmodus für S7-PLCSIM ein.
- ◆ **SetStartupSwitch** Setzt die Anlaufart (Wiederanlauf, Warmstart oder Kaltstart) für den Anlauf von S7-PLCSIM.
- ◆ **SetState** Stellt die aktuelle Schlüsselschalterposition von S7-PLCSIM (RUN, RUN-P oder STOP) ein.
- ◆ **StartPLCSim** Startet S7-PLCSIM mit der angegebenen SPS-Simulationsdatei (gespeichert von einem früheren Aufruf von SavePLC).
- ◆ **WriteDataBlockValue** Schreibt ein bestimmtes Bit, Byte, Wort oder Doppelwort in den Speicherbereich DB von S7-PLCSIM.
- ◆ **WriteFlagValue** Schreibt ein bestimmtes Bit, Byte, Wort oder Doppelwort in den Speicherbereich der Merker (M) von S7-PLCSIM.
- ◆ **WriteInputImage** Schreibt Elemente mit Beginn am StartIndex der Daten, auf die pData zeigt, in den Peripheriebereich der Eingänge (Speicherbereich PE) von S7-PLCSIM.
- ◆ **WriteInputPoint** Schreibt ein bestimmtes Bit (Boolean), ein Byte (Byte), ein 2-Byte-Wort (Integer) oder ein 4-Byte-Wort (Long) aus der Variante Data in den Peripheriebereich der Eingänge (Speicherbereich PE).



## BeginScanNotify

STDMETHOD( CS7ProSim::BeginScanNotify ) ( )

### Beschreibung

Registriert S7ProSim für Rückrufe von der Steuerung. Die Ereignisse ScanFinished und PLCSimStateChanged werden gesendet, wenn diese Ereignisse auftreten.

### Parameter

Keine

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
E_FAIL	0x80004005 : Nicht angegebener Fehler
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden
PS_E_POWEROFF	0x80040212 : S7-PLCSIM ist ausgeschaltet

### Verwendung in Visual Basic

Function **BeginScanNotify()** As Long

 **Connect**

STDMETHOD( **CS7ProSim::Connect** ) ( )

 **Beschreibung**

Verbindet S7ProSim mit S7-PLCSIM.

 **Parameter**

Keine

 **Fehlerbehandlung**

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

 **Rückgabewert**

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
E_FAIL	0x80004005 : Nicht angegebener Fehler
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden
PS_E_POWEROFF	0x80040212 : S7-PLCSIM ist ausgeschaltet

 **Verwendung in Visual Basic**

Function **Connect()** As Long

 **Continue**

```
STDMETHOD( CS7ProSim::Continue ) ( )
```

 **Beschreibung**

Setzt eine Simulation fort, die angehalten wurde.

 **Parameter**

Keine

 **Fehlerbehandlung**

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

 **Rückgabewert**

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden

 **Verwendung in Visual Basic**

```
Sub Continue()
```

## Disconnect

STDMETHOD(**CS7ProSim::Disconnect**)()

### Beschreibung

Trennt die Verbindung zwischen S7ProSim und S7-PLCSIM.

### Parameter

Keine

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
E_FAIL	0x80004005 : Nicht angegebener Fehler
PS_E_POWEROFF	0x80040212 : S7-PLCSIM ist ausgeschaltet

### Verwendung in Visual Basic

Function **Disconnect()** As Long



## EndScanNotify

STDMETHOD(**CS7ProSim::EndScanNotify**) ( )

### Beschreibung

Hebt die Registrierung von S7ProSim für Rückrufe von der Steuerung auf. Die Ereignisse ScanFinished und PLCSimStateChanged werden nicht gesendet.

### Parameter

Keine

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
E_FAIL	0x80004005 : Nicht angegebener Fehler
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden
PS_E_POWEROFF	0x80040212 : S7-PLCSIM ist ausgeschaltet
PS_E_NOTREGISTERED	0x80040209 : S7ProSim ist nicht für Rückrufe von S7-PLCSIM registriert

### Verwendung in Visual Basic

Function **EndScanNotify**() As Long


## ExecuteNmsScan

STDMETHOD(**CS7ProSim::ExecuteNmsScan**)( long MsNumber)

### Beschreibung

Zwingt S7-PLCSIM, für eine bestimmte Zeitdauer (Nms) Zyklen auszuführen, wobei nicht auf das Ende des aktuellen Zyklus gewartet wird. Ist die Zyklusbenachrichtigung aktiviert, wird dem Programm mitgeteilt, wenn S7-PLCSIM die Zyklen beendet hat. S7-PLCSIM muss sich für diese Methode in der zyklusweisen Programmbearbeitung befinden.

### Parameter

 MsNumber Zeitdauer (in Millisekunden), während der Zyklen ausgeführt werden sollen.

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
E_FAIL	0x80004005 : Nicht angegebener Fehler
PS_E_NOTSINGLESCAN	0x8004020A : In S7-PLCSIM ist nicht die zyklusweise Programmbearbeitung eingestellt
PS_E_PLCONOTRUNNING	0x8004020E : S7-PLCSIM läuft nicht
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden

### Verwendung in Visual Basic

Function **ExecuteNmsScan**(MsNumber As Long) As Long

## ExecuteNScans

STDMETHOD(**CS7ProSim::ExecuteNScans**)( long NScanNumber)

### Beschreibung

Zwingt S7-PLCSIM, eine bestimmte Anzahl Zyklen auszuführen, wobei nicht auf das Ende des aktuellen Zyklus gewartet wird. Ist die Zyklusbenachrichtigung aktiviert, wird dem Programm mitgeteilt, wenn S7-PLCSIM die Zyklen beendet hat. S7-PLCSIM muss sich für diese Methode in der zyklusweisen Programmbearbeitung befinden.

### Parameter

 NScanNumber Anzahl der Zyklen, die ausgeführt werden soll.

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
E_FAIL	0x80004005 : Nicht angegebener Fehler
PS_E_NOTSINGLESCAN	0x8004020A : In S7-PLCSIM ist nicht die zyklusweise Programmbearbeitung eingestellt
PS_E_PLCONOTRUNNING	0x8004020E : S7-PLCSIM läuft nicht
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden

### Verwendung in Visual Basic

Function **ExecuteNScans**(NScanNumber As Long) As Long

## ExecuteSingleScan

STDMETHOD( CS7ProSim::ExecuteSingleScan ) ( )

### Beschreibung

Zwingt S7-PLCSIM, einen Zyklus auszuführen, wobei nicht auf das Ende des aktuellen Zyklus gewartet wird. Ist die Zyklusbenachrichtigung aktiviert, wird dem Programm mitgeteilt, wenn S7-PLCSIM den Zyklus beendet hat. S7-PLCSIM muss sich für diese Methode in der zyklusweisen Programmbearbeitung befinden.

### Parameter

Keine

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
E_FAIL	0x80004005 : Nicht angegebener Fehler
PS_E_PLCNOTRUNNING	0x8004020E : S7-PLCSIM läuft nicht
PS_E_NOTSINGLESCAN	0x8004020A : In S7-PLCSIM ist nicht die zyklusweise Programmbearbeitung eingestellt
PS_E_MODENOTPOSSIBLE	0x8004020C : S7-PLCSIM konnte den angegebenen Bearbeitungsmodus nicht einstellen

### Verwendung in Visual Basic

Function **ExecuteSingleScan()** As Long


## GetPauseState

STDMETHOD(**CS7ProSim::GetPauseState**)( PauseStateConstants \*pVal)

### Beschreibung

Gibt den aktuellen Pausenzustand von S7-PLCSIM aus.

### Parameter

 pVal Pointer auf den ausgegebenen Zustand von S7-PLCSIM; ist eine der Einstellungen PauseStateConstants

### Hinweise

Bei Aufruf aus Visual Basic wird der Pausenzustand im Rückgabewert der Funktion ausgegeben, und es gibt keinen pVal.

Bei Aufruf aus C++ wird der Zustand in dem Wert ausgegeben, auf den pVal zeigt.

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden

### Verwendung in Visual Basic

Function **GetPauseState()** As [PauseStateConstants](#)


## GetScanMode

STDMETHOD(**CS7ProSim::GetScanMode**)( ScanModeConstants \*pVal)

### Beschreibung

Meldet den Bearbeitungsmodus von S7-PLCSIM.

### Parameter

 pVal Pointer auf den ausgegebenen Bearbeitungsmodus. Der ausgegebene Bearbeitungsmodus ist eine der ScanModeConstants

### Hinweise

Bei Aufruf aus Visual Basic wird der Bearbeitungsmodus im Rückgabewert der Funktion ausgegeben, und es gibt keinen pVal.

Bei Aufruf aus C++ wird der Zustand in dem Wert ausgegeben, auf den pVal zeigt.

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden

### Verwendung in Visual Basic

Function **GetScanMode()** As [ScanModeConstants](#)


## GetStartUpSwitch

STDMETHOD(**CS7ProSim::GetStartUpSwitch**)( RestartSwitchPosition \*pPos)

### Beschreibung

Holt die AnlaufEinstellung (Wiederanlauf, Warmstart oder Kaltstart) für S7-PLCSIM.

### Parameter

 pPos Pointer auf den S7-PLCSIM AnlaufpositionsWert, der eine der Einstellungen RestartSwitchPosition ist

### Hinweise

Bei Aufruf aus Visual Basic wird die Schalterposition im Rückgabewert der Funktion ausgegeben, und es gibt keinen pPos .

Bei Aufruf aus C++ wird der Zustand in dem Wert ausgegeben, auf den pPos zeigt.

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden

### Verwendung in Visual Basic

Function **GetStartUpSwitch()** As **RestartSwitchPosition**


 **GetState**

```
STDMETHOD( CS7ProSim::GetState ) ( BSTR *pVal )
```

 **Beschreibung**

Gibt einen String mit der aktuellen Schlüsselschalterposition von S7-PLCSIM (RUN, RUN-P oder STOP) aus.

 **Parameter**

 pVal Pointer auf den ausgegebenen Wert für die S7-PLCSIM Schlüsselschalterposition.

 **Hinweise**

Bei Aufruf aus Visual Basic wird der Zustand im Rückgabewert der Funktion ausgegeben, und es gibt keinen pVal.

Bei Aufruf aus C++ wird der Zustand in dem Wert ausgegeben, auf den pVal zeigt.

 **Fehlerbehandlung**

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

 **Rückgabewert**

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
E_FAIL	0x80004005 : Nicht angegebener Fehler
E_INVALID_STATE	0x00008002 : Ungültiger Zustand
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden

 **Verwendung in Visual Basic**

```
Function GetState() As String
```



## HotStartWithSavedValues


STDMETHOD(**CS7ProSim::HotStartWithSavedValues**)( BOOL val)

### Beschreibung

Setzt einen booleschen Wert, um anzugeben, ob S7-PLCSIM bei einem Start im Zustand HotStart gespeicherte E/A-Werte der Peripherie laden soll.

Damit S7-PLCSIM beim Start Peripherie-E/A lädt, muss der Anwender **HotStartWithSavedValues** mit dem Wert TRUE aufrufen, das SPS-Programm speichern (SavePLC) und den Anlaufzustand für S7-PLCSIM auf HotStart (SetStartUpSwitch) setzen. Beim Neustart von S7-PLCSIM werden die Peripherie-E/A geladen.

### Parameter

 **val** Der Wert TRUE zeigt an, dass S7-PLCSIM bei einem Wiederanlauf gespeicherte E/A-Daten der Peripherie laden soll. Der Wert FALSE zeigt an, dass dies nicht geschehen soll.

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden

### Verwendung in Visual Basic

```
Sub HotStartWithSavedValues(val As Long)
```

## Pause

STDMETHOD( CS7ProSim::Pause ) ( )

### Beschreibung

Hält eine Simulation an.

### Parameter

Keine

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscod
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden

### Verwendung in Visual Basic

Sub Pause()






## ReadDataBlockValue

```
STDMETHOD( CS7ProSim::ReadDataBlockValue ) (
    long BlockNumber,
    long ByteIndex,
    long BitIndex,
    PointDataTypeConstants DataType,
    VARIANT* pData)
```

### Beschreibung

Liest ein bestimmtes Bit, Byte, Wort oder Doppelwort aus dem Speicherbereich DB von S7-PLCSIM.

### Parameter

-  **BlockNumber** Nummer des Datenbausteins, der gelesen werden soll. Gültige Werte für BlockNumber richten sich nach der CPU.
-  **ByteIndex** Anfangsbyteposition im zu lesenden Datenbaustein. Gültige Werte für ByteIndex richten sich nach der CPU.
-  **BitIndex** Anfangsbitposition im zu lesenden Datenbaustein, wenn ein boolescher Wert (Bit) gelesen wird. Gültige Werte für BitIndex sind 0 bis 7.
-  **DataType** Zu lesender Datentyp. DataType muss eine der PointDataTypeConstants sein.
-  **pData** Pointer auf die Stelle für den Rückgabewert. Sie müssen diesen Speicherbereich in Ihrer Anwendung zuordnen und leer machen.

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
E_FAIL	0x80004005 : Nicht angegebener Fehler
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden
PS_E_POWEROFF	0x80040212 : S7-PLCSIM ist ausgeschaltet
PS_E_BADTYPE	0x80040206 : Ungültiger Datentyp
PS_E_BADBYTENDX	0x80040201 : Byteindex ist ungültig
PS_E_BADBYTECOUNT	0x80040202 : Größe des Daten-Array ist für den angegebenen Anfangs-Byteindex ungültig
PS_E_READFAILED	0x80040203 : Leseoperation fehlgeschlagen

### Verwendung in Visual Basic

```
Sub ReadDataBlockValue(BlockNum As Long, ByteIndex As Long, BitIndex As Long, DataType As PointDataTypeConstants, pData)
```





## ReadFlagValue

```
STDMETHOD(CS7ProSim::ReadFlagValue) ( long ByteIndex,
                                        long BitIndex,
                                        PointDataTypeConstants DataType,
                                        VARIANT* pData)
```

### Beschreibung

Liest ein bestimmtes Bit, Byte, Wort oder Doppelwort aus dem Speicherbereich der Merker (M) von S7-PLCSIM.

### Parameter

-  **ByteIndex**    Stellt die zu lesende Anfangsbyteposition im Speicherbereich der Merker dar. Gültige Werte für ByteIndex richten sich nach der CPU.
-  **BitIndex**    Stellt die zu lesende Anfangsbitposition im Speicherbereich der Merker dar, wenn ein boolescher Wert (Bit) gelesen werden soll. Gültige Werte für BitIndex sind 0 bis 7.
-  **DataType**    Stellt den zu lesenden Datentyp dar. DataType muss eine der PointDataTypeConstants sein.
-  **pData**        Pointer auf die Stelle für den Rückgabewert. Sie müssen diesen Speicherbereich in Ihrer Anwendung zuordnen und leer machen.

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscod
E_FAIL	0x80004005 : Nicht angegebener Fehler
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden
PS_E_POWEROFF	0x80040212 : S7-PLCSIM ist ausgeschaltet
PS_E_BADTYPE	0x80040206 : Ungültiger Datentyp
PS_E_BADBYTENDX	0x80040201 : Byteindex ist ungültig
PS_E_BADBYTECOUNT	0x80040202 : Größe des Daten-Array ist für den angegebenen Anfangs-Byteindex ungültig
PS_E_READFAILED	0x80040203 : Leseoperation fehlgeschlagen

### Verwendung in Visual Basic

```
Sub ReadFlagValue(ByteIndex As Long, BitIndex As Long, DataType As PointDataTypeConstants, pData)
```





## ReadOutputImage

```
STDMETHOD( CS7ProSim::ReadOutputImage ) ( long StartIndex,
                                           long ElementsToRead,
                                           ImageDataTypeConstants DataType,
                                           VARIANT* pData )
```

### Beschreibung

Liest Elemente aus dem Peripheriebereich der Ausgänge (Speicherbereich PA) von S7-PLCSIM.

### Parameter

-  **StartIndex**      Stellt die zu lesende Anfangsbyteposition im Peripheriebereich der Ausgänge dar. Gültige Werte für StartIndex richten sich nach der CPU.
-  **ElementsToRead**      Stellt die Anzahl der aus dem Abbild zu lesenden Bytes, Wörter oder Doppelwörter dar. Gültige Werte für ElementsToRead richten sich nach der CPU.
-  **DataType**      Stellt den zu lesenden Datentyp dar. Der DataType muss eine der ImageDataTypeConstants sein.
-  **pData**      Pointer auf die Stelle für ausgegebene Elemente. Die gültigen Werte für Daten richten sich nach ElementsToRead. Sie müssen diesen Speicherbereich in Ihrer Anwendung zuordnen und leer machen.

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
E_FAIL	0x80004005 : Nicht angegebener Fehler
PS_E_BADBYTENDX	0x80040201 : Byteindex ist ungültig
PS_E_BADBYTECOUNT	0x80040202 : Größe des Daten-Array ist für den angegebenen Anfangs-Byteindex ungültig
PS_E_READFAILED	0x80040203 : Leseoperation fehlgeschlagen
PS_E_BADTYPE	0x80040206 : Ungültiger Datentyp
PS_E_NOTALLREADSWORKED	0x8004020F : Nicht alle Leseoperationen waren erfolgreich
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden
PS_E_POWEROFF	0x80040212 : S7-PLCSIM ist ausgeschaltet

### Verwendung in Visual Basic

Function **ReadOutputImage**(StartIndex As Long, ElementsToRead As Long, DataType As **ImageDataTypeConstants**, pData) As Long





## ReadOutputPoint

```
STDMETHOD(CS7ProSim::ReadOutputPoint)( long ByteIndex,
                                         long BitIndex,
                                         PointDataTypeConstants DataType,
                                         VARIANT* pData)
```

### Beschreibung

Liest ein bestimmtes Bit (Boolean), ein Byte (Byte), ein 2-Byte-Wort (Integer) oder ein 4-Byte-Wort (Long) aus dem Peripheriebereich der Ausgänge (Speicherbereich PA).

### Parameter

-  **ByteIndex** Stellt die zu lesende Anfangsbyteposition im Peripheriebereich dar. Gültige Werte für ByteIndex richten sich nach der CPU.
-  **BitIndex** Stellt die zu lesende Bitposition (in Bytes) im Peripheriebereich dar. Gültig sind die Werte 0 bis 7.
-  **DataType** Eine der PointDataTypeConstants
-  **pData** Pointer auf die zu lesenden Daten. Die gültigen Werte für Data richten sich nach dem Datentyp.

### Hinweise

Wenn der DataType ein S7\_Bit ist, dann müssen ByteIndex und BitIndex beide auf gültige Indizes gesetzt werden. Bei Erfolg gibt die Methode das angegebene Bit in pData aus, und der Variant-Datentyp ist Boolean.

Wenn der DataType ein S7\_Byte, S7\_Word oder S7\_DoubleWord ist, dann muss ByteIndex auf einen gültigen Index gesetzt werden (BitIndex wird ignoriert). Bei Erfolg meldet die Methode den Wert in pData. Der Variant-Datentyp ist Byte, Integer oder Long, je nach DataType.

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
E_FAIL	0x80004005 : Nicht angegebener Fehler
PS_E_BADBYTENDX	0x80040201 : Byteindex ist ungültig
PS_E_BADBYTECOUNT	0x80040202 : Größe des Daten-Array ist für den angegebenen Anfangs-Byteindex ungültig
PS_E_READFAILED	0x80040203 : Leseoperation fehlgeschlagen
PS_E_BADBITNDX	0x80040205 : Bitindex ist ungültig
PS_E_BADTYPE	0x80040206 : Ungültiger Datentyp
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden
PS_E_POWEROFF	0x80040212 : S7-PLCSIM ist ausgeschaltet

### Verwendung in Visual Basic

Function **ReadOutputPoint**(ByteIndex As Long, BitIndex As Long, DataType As **PointDataTypeConstants**, pData As Long

## SavePLC


STDMETHOD(**CS7ProSim::SavePLC**)( BSTR FileName)

### Beschreibung

Speichert die aktuellen simulierten SPS-Daten in einer Datei.

Die Daten, die gespeichert werden, bestehen aus dem Programm, der Hardware-Konfiguration, der Schlüsselschalterposition wie im Unterfenster "CPU", der Art der Programmbearbeitung (automatisch oder zyklusweise), dem E/A-Zustand, Zeitwerten, symbolischen Adressen und der Netzeinstellung (ein oder aus).

### Parameter

 **FileName** Name der Datei, in der die Daten des simulierten Zielsystems gespeichert werden sollen

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscod
STG_E_CANTSAVE	0x80030103 : Speichern nicht möglich
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden

### Verwendung in Visual Basic

Sub **SavePLC**(*FileName As String*)


## SetScanMode

STDMETHOD(**CS7ProSim::SetScanMode**)( ScanModeConstants newVal)

### Beschreibung

Stellt den Bearbeitungsmodus für S7-PLCSIM ein.

### Parameter

 **newVal** Einzustellender Bearbeitungsmodus für S7-PLCSIM. Der Bearbeitungsmodus muss eine der ScanModeConstants sein.

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden

### Verwendung in Visual Basic

Sub **SetScanMode**(*newVal* As **ScanModeConstants**)




## SetStartUpSwitch

STDMETHOD(**CS7ProSim::SetStartUpSwitch**)( RestartSwitchPosition position)

### Beschreibung

Setzt die Anlaufart (Wiederanlauf, Warmstart oder Kaltstart) für den Anlauf von S7-PLCSIM.

### Parameter

 position Einzustellender Wert für die Anlaufposition in S7-PLCSIM

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden

### Verwendung in Visual Basic

Sub **SetStartUpSwitch**(*position* As **RestartSwitchPosition**)

## SetState

STDMETHOD(**CS7ProSim::SetState**)( BSTR newVal)

### Beschreibung

Stellt die aktuelle Schlüsselschalterposition von S7-PLCSIM (RUN, RUN-P oder STOP) ein.

### Parameter

 newVal Einzustellender Wert für die Schlüsselschalterposition in S7-PLCSIM

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
E_FAIL	0x80004005 : Nicht angegebener Fehler
E_INVALID_STATE	0x00008002 : Ungültiger Zustand
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden

### Verwendung in Visual Basic

Sub **SetState**(newVal As String)


## StartPLCSim

STDMETHOD(**CS7ProSim::StartPLCSim**)( BSTR plcFile)

### Beschreibung

Startet S7-PLCSIM mit der angegebenen SPS-Simulationsdatei (gespeichert von einem früheren Aufruf von SavePLC).

### Parameter

 plcFile Name der Datei, mit der S7-PLCSIM gestartet werden soll

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
E_FAIL	0x80004005 : Nicht angegebener Fehler

### Verwendung in Visual Basic

Sub **StartPLCSim**(*plcFile As String*)





## WriteDataBlockValue

```
STDMETHOD( CS7ProSim::WriteDataBlockValue ) (
    long BlockNumber,
    long ByteIndex,
    long BitIndex,
    const VARIANT* pData)
```

### Beschreibung

Schreibt ein bestimmtes Bit, Byte, Wort oder Doppelwort in den Speicherbereich DB von S7-PLCSIM.

### Parameter

-  **BlockNumber**    Stellt die Nummer des Datenbausteins dar, der geschrieben werden soll. Gültige Werte für BlockNumber richten sich nach der CPU.
-  **ByteIndex**    Stellt die zu schreibende Anfangsbyteposition im Datenbaustein dar. Gültige Werte für ByteIndex richten sich nach der CPU.
-  **BitIndex**      Stellt die zu schreibende Anfangsbitposition im Datenbaustein dar, wenn ein boolescher Wert (Bit) geschrieben werden soll. Gültige Werte für BitIndex sind 0 bis 7.
-  **pData**         Pointer auf die Stelle, die die zu schreibenden Daten enthält. Sie müssen diesen Speicherbereich in Ihrer Anwendung zuordnen und leer machen.

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscod
E_FAIL	0x80004005 : Nicht angegebener Fehler
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden
PS_E_POWEROFF	0x80040212 : S7-PLCSIM ist ausgeschaltet
PS_E_BADTYPE	0x80040206 : Ungültiger Datentyp
PS_E_BADBYTENDX	0x80040201 : Byteindex ist ungültig
PS_E_BADBYTECOUNT	0x80040202 : Größe des Daten-Array ist für den angegebenen Anfangs-Byteindex ungültig
PS_E_WRITEFAILED	0x80040204 : Schreiboperation fehlgeschlagen

### Verwendung in Visual Basic

```
Sub WriteDataBlockValue(BlockNum As Long, ByteIndex As Long, BitIndex As Long, pData)
```




## WriteFlagValue

```
STDMETHOD(CS7ProSim::WriteFlagValue) ( long ByteIndex,
                                         long BitIndex,
                                         const VARIANT* pData)
```

### Beschreibung

Schreibt ein bestimmtes Bit, Byte, Wort oder Doppelwort in den Speicherbereich der Merker (M) von S7-PLCSIM.

### Parameter

-  **ByteIndex**    Stellt die zu schreibende Anfangsbyteposition im Speicherbereich der Merker dar, in den geschrieben werden soll. Gültige Werte für ByteIndex richten sich nach der CPU.
-  **BitIndex**    Stellt die zu schreibende Anfangsbitposition im Speicherbereich der Merker dar, wenn ein boolescher Wert (Bit) geschrieben werden soll. Gültige Werte für BitIndex sind 0 bis 7.
-  **pData**        Pointer auf die Stelle, die die zu schreibenden Daten enthält. Sie müssen diesen Speicherbereich in Ihrer Anwendung zuordnen und leer machen.

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscod
E_FAIL	0x80004005 : Nicht angegebener Fehler
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden
PS_E_POWEROFF	0x80040212 : S7-PLCSIM ist ausgeschaltet
PS_E_BADTYPE	0x80040206 : Ungültiger Datentyp
PS_E_BADBYTENDX	0x80040201 : Byteindex ist ungültig
PS_E_BADBYTECOUNT	0x80040202 : Größe des Daten-Array ist für den angegebenen Anfangs-Byteindex ungültig
PS_E_WRITEFAILED	0x80040204 : Schreiboperation fehlgeschlagen

### Verwendung in Visual Basic

```
Sub WriteFlagValue(ByteIndex As Long, BitIndex As Long, pData)
```


## WriteInputImage


```
STDMETHOD(CS7ProSim::WriteInputImage)( long StartIndex, const
VARIANT* pData)
```

### Beschreibung

Schreibt Elemente in den Peripheriebereich der Eingänge (Speicherbereich PE) von S7-PLCSIM, mit Beginn an StartIndex der Daten, auf die pData zeigt.

### Parameter

 StartIndex Stellt die zu schreibende Anfangsbyteposition im Peripheriebereich der Eingänge dar. Gültige Werte für StartIndex richten sich nach der CPU.

 pData Pointer auf die Daten, die S7-PLCSIM schreiben soll. Die gültigen Werte für die Daten richten sich nach der CPU. Sie müssen diesen Speicherbereich in Ihrer Anwendung zuordnen und leer machen.

### Hinweise

Die Art der Elemente, die geschrieben werden soll, wird durch die Art der Elemente in Data angegeben. Alle Elemente müssen vom gleichen Datentyp sein. Ein Byte-Array schreibt Bytes, ein Integer-Array schreibt Wörter und ein Long-Array schreibt Doppelwörter. Die geschriebenen Werte sind "roh" und werden von der Methode in keinsten Weise ausgewertet bzw. umgewandelt. Die Anzahl der geschriebenen Elemente wird von der Größe des Array festgelegt, auf das Data zeigt.

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
E_FAIL	0x80004005 : Nicht angegebener Fehler
PS_E_BADBYTENDX	0x80040201 : Byteindex ist ungültig
PS_E_BADBYTECOUNT	0x80040202 : Größe des Daten-Array ist für den angegebenen Anfangs-Byteindex ungültig
PS_E_WRITEFAILED	0x80040204 : Schreiboperation fehlgeschlagen
PS_E_BADTYPE	0x80040206 : Ungültiger Datentyp
PS_E_NOTALLWRITESWORKED	0x80040210 : Nicht alle Schreiboperationen waren erfolgreich
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden
PS_E_POWEROFF	0x80040212 : S7-PLCSIM ist ausgeschaltet

### Verwendung in Visual Basic

```
Function WriteInputImage(StartIndex As Long, Data As Long)
```




## WriteInputPoint

```
STDMETHOD(CS7ProSim::WriteInputPoint)( long ByteIndex,
                                         long BitIndex,
                                         const VARIANT* pData)
```

### Beschreibung

Schreibt ein bestimmtes Bit (Boolean), ein Byte (Byte), ein 2-Byte-Wort (Integer) oder ein 4-Byte-Wort (Long) aus der Variante Data in den Peripheriebereich der Eingänge (Speicherbereich PE).

### Parameter

-  **ByteIndex**    Stellt die zu schreibende Anfangsbyteposition im Peripheriebereich der Eingänge dar. Gültige Werte für ByteIndex richten sich nach der CPU.
-  **BitIndex**    Stellt die zu schreibende Bitposition (in Bytes) im Peripheriebereich dar. Gültig sind die Werte 0 bis 7.
-  **pData**        Pointer auf die zu schreibenden Daten. Die gültigen Werte für Data richten sich nach dem Datentyp.

### Hinweise

Wenn Boolean als Datentyp angegeben wird, dann müssen ByteIndex und BitIndex beide auf gültige Indizes gesetzt werden. Bei Erfolg schreibt die Methode das angegebene Bit in pData.

Wird Byte, Integer oder Long als Datentyp angegeben, dann muss ByteIndex auf einen gültigen Index gesetzt werden (BitIndex wird ignoriert). Bei Erfolg schreibt die Methode die Elemente in pData.

### Fehlerbehandlung

Fehler werden im Ereignis ConnectionError gemeldet, nicht vom Funktionsaufruf.

### Rückgabewert

Wert	Bedeutung
S_OK	0x00000000 : Erfolgscode
E_FAIL	0x80004005 : Nicht angegebener Fehler
PS_E_BADBYTENDX	0x80040201 : Byteindex ist ungültig
PS_E_BADBYTECOUNT	0x80040202 : Größe des Daten-Array ist für den angegebenen Anfangs-Byteindex ungültig
PS_E_WRITEFAILED	0x80040204 : Schreiboperation fehlgeschlagen
PS_E_BADBITNDX	0x80040205 : Bitindex ist ungültig
PS_E_BADTYPE	0x80040206 : Ungültiger Datentyp
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden
PS_E_POWEROFF	0x80040212 : S7-PLCSIM ist ausgeschaltet

### Verwendung in Visual Basic

Function **WriteInputPoint**(ByteIndex As Long, BitIndex As Long, Data) As Long





# Ereignisse

- ◆ **ConnectionError** Wird erzeugt, wenn keine Verbindung zur Control Engine ("S7-PLCSIM") hergestellt werden kann oder wenn ein Fehler beim Aufruf einer S7ProSim Methode auftritt.
- ◆ **PauseStateChanged** Wird erzeugt, wenn ein Zustandswechsel Anhalten/Fortfahren erkannt wird. NewState ist eine Zeichenkette, die eine der PauseStateConstants darstellt.
- ◆ **PLCSimStateChanged** Wird erzeugt, wenn ein neuer Schalterzustand des Zielsystems erkannt wird. NewState ist der neue Betriebszustand: "RUN", "RUN\_P" oder "STOP".
- ◆ **ScanFinished** Wird erzeugt, wenn ein einzelner Zyklus ausgeführt wurde. ScanInfo bietet indizierte Informationen zum Zyklus.
- ◆ **ScanModeChanged** Wird erzeugt, wenn ein Wechsel des Bearbeitungsmodus erkannt wird. NewState ist eine Zeichenkette, die eine der ScanModeConstants darstellt.

## **ConnectionError**

HRESULT **ConnectionError**(BSTR *ControlEngine*, long *Error*)

### **Beschreibung**

Wird erzeugt, wenn keine Verbindung zur Control Engine ("S7-PLCSIM") hergestellt werden kann oder wenn ein Fehler beim Aufruf einer S7ProSim Methode auftritt.

### **Verwendung in Visual Basic**

Event **ConnectionError**(*ControlEngine As String, Error As Long*)

 **PauseStateChanged**

HRESULT **PauseStateChanged**(BSTR NewState)

 **Beschreibung**

Wird erzeugt, wenn ein Zustandswechsel Anhalten/Fortfahren erkannt wird. NewState ist eine Zeichenkette, die eine der PauseStateConstants darstellt.

 **Verwendung in Visual Basic**

Event **PauseStateChanged**(*NewState As String*)

## **PLCSimStateChanged**

HRESULT **PLCSimStateChanged**(BSTR NewState)

### **Beschreibung**

Wird erzeugt, wenn ein neuer Schalterzustand des Zielsystems erkannt wird. NewState ist der neue Betriebszustand: "RUN", "RUN\_P" oder "STOP".

### **Verwendung in Visual Basic**

Event **PLCSimStateChanged**(*NewState As String*)

 **ScanFinished**

HRESULT **ScanFinished**(VARIANT ScanInfo)

 **Beschreibung**

Wird erzeugt, wenn ein einzelner Zyklus ausgeführt wurde. ScanInfo bietet indizierte Informationen zum Zyklus.

 **Verwendung in Visual Basic**

Event **ScanFinished**(*ScanInfo*)

## ScanModeChanged

HRESULT `ScanModeChanged`(BSTR *NewState*)









### Beschreibung

Wird erzeugt, wenn ein Wechsel des Bearbeitungsmodus erkannt wird. *NewState* ist eine Zeichenkette, die eine der `ScanModeConstants` darstellt.

### Verwendung in Visual Basic

Event `ScanModeChanged`(*NewState As String*)

# Typdefinitionen

 <b>CPURunMode</b>	Konstanten für den Bearbeitungszustand des Betriebszustands RUN der CPU
 <b>ImageDataTypeConstants</b>	Konstanten für die Methode ReadOutputImage
 <b>PauseStateConstants</b>	Konstanten für den Pausenzustand
 <b>PointDataTypeConstants</b>	Konstanten für die Methode ReadOutputPoint
 <b>RestartSwitchPosition</b>	Konstanten für die Position des Anlaufartenschalters auf der Frontplatte
 <b>ScanModeConstants</b>	Konstanten für den Bearbeitungsmodus
 <b>tagPauseState</b>	Konstanten für den Pausenzustand
 <b>ScanInfo-Konstanten</b>	Konstanten für Informationen zum Zyklus

## **CPURunMode**

```
enum CPURunMode { CONTINUOUS_SCAN, SINGLE_SCAN, SINGLE_STEP }
```

### **Beschreibung**

Konstanten für den Bearbeitungszustand des Betriebszustands RUN der CPU

### **Elemente**

*CONTINUOUS\_SCAN*

*SINGLE\_SCAN*

*SINGLE\_STEP*



 **ImageDataTypeConstants**

```
enum {  
    S7Byte = 2,  
    S7Word = 3,  
    S7DoubleWord = 4  
}
```

 **Beschreibung**

Konstanten für die Methode ReadOutputImage

 **Elemente**

*S7Byte*

*S7DoubleWord*

*S7Word*

## **PauseStateConstants**

```
enum {  
    Running = 0,  
    Paused = 1,  
    Disabled = 2  
}
```

### **Beschreibung**

Konstanten für den Pausenzustand

### **Elemente**

*DISABLED*

*Paused*

*Running*

 **PointDataTypeConstants**

```
enum {  
    S7_Bit = 1,  
    S7_Byte = 2,  
    S7_Word = 3,  
    S7_DoubleWord = 4  
}
```

 **Beschreibung**

Konstanten für die Methode ReadOutputPoint

 **Elemente**

*S7\_Bit*  
*S7\_Byte*  
*S7\_DoubleWord*  
*S7\_Word*

## RestartSwitchPosition

```
enum {  
    WarmStart = 0,  
    HotStart = 1,  
    ColdStart = 2  
}
```

### Beschreibung

Konstanten für die Position des Anlaufartenschalters auf der Frontplatte

### Elemente

*ColdStart* Wiederanlaufposition OB102

*HotStart* Wiederanlaufposition OB101

*WarmStart* Wiederanlaufposition OB100

 **ScanModeConstants**

```
enum {  
    SingleScan = 0,  
    ContinuousScan = 1  
}
```

 **Beschreibung**

Konstanten für den Bearbeitungsmodus

 **Elemente**

*ContinuousScan*

*SingleScan*

## tagPauseState

```
enum tagPauseState { ENABLED_RUNNING, ENABLED_PAUSED, DISABLED }
```

### Beschreibung

Konstanten für den Pausenzustand

### Elemente

*DISABLED*


*ENABLED\_PAUSED*

*ENABLED\_RUNNING*

## ScanInfo-Konstanten

### ScanInfo-Konstanten


 ScanInfo


 NUM\_OF\_SCANINFO\_ELEMENTS Anzahl der Elemente im ausgegebenen Array ScanInfo.

 EXECUTION\_TIME\_NDX Index 0: Ausführungszeit in ms

 MIN\_CYCLE\_TIME\_NDX Index 1: kürzeste Ausführungszeit in ms

 LARGEST\_CYCLE\_TIME\_NDX Index 2: größte Ausführungszeit in ms

 AVERAGE\_CYCLE\_TIME\_NDX Index 3: durchschnittliche Zykluszeit in ms

 IS\_PLC\_RUNNING\_NDX Index 4: Merker: 1 = Zielsystem läuft; 0 = Zielsystem läuft nicht

### ScanInfo

#### ScanInfo

Der Variant-Datentyp **ScanInfo** stellt ein Array aus dem Datentyp Long dar. Jeder Datentyp Long im Array definiert einige Informationen zum Zyklus, entsprechend der Definition der ScanInfo-Konstanten.

#### NUM\_OF\_SCANINFO\_ELEMENTS

```
#define NUM_OF_SCANINFO_ELEMENTS 5
```

#### Beschreibung

Anzahl der Elemente im ausgegebenen Array ScanInfo.

#### EXECUTION\_TIME\_NDX

```
#define EXECUTION_TIME_NDX 0
```

#### Beschreibung

Index 0: Ausführungszeit in ms

#### MIN\_CYCLE\_TIME\_NDX

```
#define MIN_CYCLE_TIME_NDX 1
```

#### Beschreibung

Index 1: kürzeste Ausführungszeit in ms

## LARGEST\_CYCLE\_TIME\_NDX

```
#define LARGEST_CYCLE_TIME_NDX 2
```

### Beschreibung

Index 2: größte Ausführungszeit in ms

## AVERAGE\_CYCLE\_TIME\_NDX

```
#define AVERAGE_CYCLE_TIME_NDX 3
```

### Beschreibung

Index 3: durchschnittliche Zykluszeit in ms

## IS\_PLC\_RUNNING\_NDX

```
#define IS_PLC_RUNNING_NDX 4
```

### Beschreibung

Index 4: Merker: 1 = Zielsystem läuft; 0 = Zielsystem läuft nicht



## Fehlerausgabecodes

PS_E_BADBITNDX	0x80040205 : Bitindex ist ungültig
PS_E_BADBYTECOUNT	0x80040202 : Größe des Daten-Array ist für den angegebenen Anfangs-Byteindex ungültig
PS_E_BADBYTENDX	0x80040201 : Byteindex ist ungültig
PS_E_BADTYPE	0x80040206 : Ungültiger Datentyp
PS_E_INVALIDCALLBACK	0x80040207 : Ungültiger Rückruf
PS_E_INVALIDDISPATCH	0x80040208 : Ungültige Verteilung
PS_E_INVALIDINPUT	0x80040213 : Ungültiger Eingang
PS_E_INVALIDSCANTYPE	0x8004020B : Ungültiger Zyklustyp, muss eine der ScanModeConstants sein.
PS_E_MODENOTPOSSIBLE	0x8004020C : S7-PLCSIM konnte den angegebenen Bearbeitungsmodus nicht einstellen
PS_E_NOTALLREADSWORKED	0x8004020F : Nicht alle Leseoperationen waren erfolgreich
PS_E_NOTALLWRITESWORKED	0x80040210 : Nicht alle Schreiboperationen waren erfolgreich
PS_E_NOTCONNECTED	0x80040211 : S7ProSim ist nicht mit S7-PLCSIM verbunden
PS_E_NOTIFICATION_EXIST	0x8004020D : S7ProSim ist bereits für die Benachrichtigung registriert
PS_E_NOTREGISTERED	0x80040209 : S7ProSim ist nicht für Rückrufe von S7-PLCSIM registriert
PS_E_NOTSINGLESCAN	0x8004020A : In S7-PLCSIM ist nicht die zyklusweise Programmbearbeitung eingestellt
PS_E_PLCSIMNOTRUNNING	0x8004020E : S7-PLCSIM läuft nicht
PS_E_POWEROFF	0x80040212 : S7-PLCSIM ist ausgeschaltet
PS_E_READFAILED	0x80040203 : Leseoperation fehlgeschlagen
PS_E_WRITEFAILED	0x80040204 : Schreiboperation fehlgeschlagen
E_FAIL	0x80004005 : Nicht angegebener Fehler
E_INVALID_STATE	0x00008002 : Ungültiger Zustand
S_OK	0x00000000 : Erfolgscode
STG_E_CANTSAVE	0x80030103 : Speichern nicht möglich



# Index

## A

Aufzählungstypen, 47  
  CPURunMode, 48  
  ImageDataTypeConstants, 49  
  PauseStateConstants, 50  
  PointDataTypeConstants, 51  
  RestartSwitchPosition, 52  
  ScanModeConstants, 53  
  tagPauseState, 54  
Automatische Programmbearbeitung  
  GetScanMode, 22  
  ScanModeConstants, 53  
  SetScanMode, 32  
AVERAGE\_CYCLE\_TIME\_NDX, 55, 56

## B

Bearbeitungsmodus  
  Einstellung, 32  
  Holen, 22

## C

ColdStart, 52  
  Einstellung, 33  
CPURunMode, 48

## D

Datenbausteinwerte (DB)  
  Lesen, 27  
  Schreiben, 36  
Definierte Konstanten, 47  
  CPURunMode, 48  
  ImageDataTypeConstants, 49  
  PauseStateConstants, 50  
  PointDataTypeConstants, 51  
  RestartSwitchPosition, 52  
  ScanModeConstants, 53  
  tagPauseState, 54

## E

Einführung, 7  
Ereignis ConnectionError, 42  
Ereignis PauseStateChanged, 43

Ereignis PLCSimStateChanged, 44  
Ereignis ScanFinished, 45  
Ereignis ScanModeChanged, 46  
Ereignis-Handler, 8  
Ereignisse, 41  
  ConnectionError, 42  
  PauseStateChanged, 43  
  PLCSimStateChanged, 44  
  ScanFinished, 45  
  ScanModeChanged, 46  
EXECUTION\_TIME\_NDX, 55

## F

Fehlerausgabecodes, 57

## H

HotStart, 52  
  Einstellung, 25, 33

## I

ImageDataTypeConstants, 49  
IS\_PLC\_RUNNING\_NDX, 55, 56

## K

Konstanten, 47  
  CPURunMode, 48  
  ImageDataTypeConstants, 49  
  PauseStateConstants, 50  
  PointDataTypeConstants, 51  
  RestartSwitchPosition, 52  
  ScanModeConstants, 53  
  tagPauseState, 54

## L

Laden gespeicherter \*.PLC-Dateien, 35  
Laden von Peripherie-E/A bei Wiederanlauf, 25  
LARGEST\_CYCLE\_TIME\_NDX, 55, 56  
Lesen  
  Ausgang, 30  
  Ausgangsabbild, 29  
  Datenbausteinwerte (DB), 27  
  Merkerwert (M), 28

**M**

## Merker

- Lesen, 28
- Schreiben, 37

## Merkerwert

- Lesen, 28
- Schreiben, 37

## Methode BeginScanNotify, 13

## Methode Connect, 14

## Methode Continue, 15

## Methode Disconnect, 16

## Methode EndScanNotify, 17

## Methode ExecuteNmsScan, 18

## Methode ExecuteNScans, 19

## Methode ExecuteSingleScan, 20

## Methode GetPauseState, 21

## Methode GetScanMode, 22

## Methode GetStartUpSwitch, 23

## Methode GetState, 24

## Methode HotStartWithSavedValues, 25

## Methode Pause, 26

## Methode ReadDataBlockValue, 27

## Methode ReadFlagValue, 28

## Methode ReadOutputImage, 29

## Methode ReadOutputPoint, 30

## Methode SavePLC, 31

## Methode SetScanMode, 32

## Methode SetStartUpSwitch, 33

## Methode SetState, 34

## Methode StartPLCSim, 35

## Methode WriteDataBlockValue, 36

## Methode WriteFlagValue, 37

## Methode WriteInputImage, 38

## Methode WriteInputPoint, 39

## Methoden, 11

- BeginScanNotify, 13
- Connect, 14
- Continue, 15
- Disconnect, 16
- EndScanNotify, 17
- ExecuteNmsScan, 18

## ExecuteNScans, 19

## ExecuteSingleScan, 20

## GetPauseState, 21

## GetScanMode, 22

## GetStartUpSwitch, 23

## GetState, 24

## HotStartWithSavedValues, 25

## Pause, 26

## ReadDataBlockValue, 27

## ReadFlagValue, 28

## ReadOutputImage, 29

## ReadOutputPoint, 30

## SavePLC, 31

## SetScanMode, 32

## SetStartUpSwitch, 33

## SetState, 34

## StartPLCSim, 35

## WriteDataBlockValue, 36

## WriteFlagValue, 37

## WriteInputImage, 38

## WriteInputPoint, 39

## MIN\_CYCLE\_TIME\_NDX, 55

**N**

## NUM\_OF\_SCANINFO\_ELEMENTS, 55

**P**

## Pausenzustand

- Einstellung, 15, 26
- Holen, 21

## PauseStateConstants, 50

## PLCSIM, Starten, 35

## PointDataTypeConstants, 51

## Position Anlaufartenschalter, 52

- Einstellung, 33
- Holen, 23

## Position Schlüsselschalter

- Einstellung, 34
- Holen, 24

## Programmbearbeitungsmethoden

- ExecuteNmsScan, 18
- ExecuteNScans, 19

- ExecuteSingleScan, 20
- Programmieren einer S7ProSim Schnittstelle zu S7-PLCSIM, 8
- Projektverweise, 7
- R**
- Referenzen, 7
- RestartSwitchPosition, 52
- Rückgabewerte, 57
- S**
- S7-PLCSIM, Starten, 25, 35
- S7ProSim
  - Aufnehmen in VB-Projekt, 7
  - Schnittstelle zu S7-PLCSIM, Programmieren, 8
  - Übersicht, 7
- S7ProSim Pro Methoden
  - ReadDataBlockValue, 27
  - ReadFlagValue, 28
  - WriteDataBlockValue, 36
  - WriteFlagValue, 37
- ScanInfo-Konstanten, 55
- ScanModeConstants, 53
- Schreiben
  - Datenbausteinwerte (DB), 36
  - Eingang, 39
  - Eingangsabbild, 38
  - Merkerwert (M), 37
- Siemens S7ProSim COM-Objekt, Aufnehmen in Projekt, 7
- Speichern von \*.PLC-Dateien, 31
- T**
- tagPauseState, 54
- Typdefinitionen, 47
  - CPURunMode, 48
  - ImageDataTypeConstants, 49
  - PauseStateConstants, 50
  - PointDataTypeConstants, 51
  - RestartSwitchPosition, 52
  - ScanModeConstants, 53
  - tagPauseState, 54
- U**
- Übersicht, 7
- V**
- Visual Basic Projekt, Aufnehmen von S7ProSim, 7
- W**
- WarmStart, 52
  - Einstellung, 33
- Z**
- Zyklusbenachrichtigung, 13, 17
- Zyklusweise Programmbearbeitung
  - ExecuteNmsScan, 18
  - ExecuteNScans, 19
  - ExecuteSingleScan, 20
  - GetScanMode, 22
  - ScanModeConstants, 53
  - SetScanMode, 32