

SIEMENS

SIMATIC

S7-1500 Software Controller

Ergänzungen zu ODK 1500S V1.0

Produktinformation

Einleitung

Allgemeine Informationen

Diese Produktinformation enthält wichtige Informationen zu ODK 1500S V1.0. Die Produktinformation ist Bestandteil des gelieferten Produkts und die darin enthaltenen Aussagen sind in Zweifelsfällen in der Verbindlichkeit anderen Aussagen übergeordnet.

Gültigkeitsbereich der Produktinformation

Diese Produktinformation gilt für die Verwendung von ODK 1500S mit folgenden Produkten:

- CPU 1505S
- CPU 1507S

Änderungen beziehen sich auf das Handbuch "Open Development Kit 1500S V1.0" (A5E35253935-AA).

Service & Support im Internet

Einen Wegweiser für die technische Dokumentation zu den verschiedenen SIMATIC Produkten und Systemen finden Sie im Internet (<http://www.siemens.de/simatic-tech-doku-portal>).

Zusätzlich zu unserem Dokumentations-Angebot bieten wir Ihnen im Internet (<http://www.siemens.com/automation/service&support>) unsere vollständige Online-Wissensdatenbank. Dort finden Sie Folgendes:

- Den Newsletter, der Sie ständig mit aktuellen Informationen zu Ihren Produkten versorgt
- Die richtigen Dokumente über unsere Suchfunktion unter Service & Support
- Das Bulletin Board, ein weltweiter Wissensaustausch für Anwender und Experten
- Ihren örtlichen Ansprechpartner für Automation & Drives über unsere Kontaktdatenbank
- Informationen zu unserem Vor-Ort-Service, zu Reparaturen, Ersatzteilen und vielem mehr

Weitere Informationen

Beachten Sie für weitere Informationen und Dokumentationen zum Thema SIMATIC Embedded Bundles den Produkt-Support im Internet (<http://support.automation.siemens.com/WW/view/de/42783973/133300>).

Security-Hinweise

Security-Hinweise

Für den sicheren Betrieb von Produkten und Lösungen von Siemens ist es erforderlich, geeignete Schutzmaßnahmen (z. B. Zellschutzkonzept) zu ergreifen und jede Komponente in ein ganzheitliches Industrial Security-Konzept zu integrieren, das dem aktuellen Stand der Technik entspricht. Dabei sind auch eingesetzte Produkte von anderen Herstellern zu berücksichtigen. Weitergehende Informationen über Industrial Security finden Sie unter (<http://www.siemens.com/industrialsecurity>).

Hinweise zu Open Source Software

Sie können als Ergänzung zum Produkt verschiedene Entwicklungstools des Open Source Projekts MinGW32 (<http://www.mingw.org/>) herunterladen und nachinstallieren. Bitte beachten Sie, dass diese Komponenten optional sind und vom MinGW32-Projekt bzw. weiteren Lizenzgebern unter abweichenden Lizenzen bereitgestellt und vertrieben werden, die Sie auf der Projekt-Homepage oder bei den jeweiligen Paketen einsehen können.

Hinweise

Korrekturen zur Dokumentation

Korrekturen zu den Systemanforderungen

Eine ODK-Anwendung für die Windows-Umgebung erfordert Microsoft Visual Studio 2010 inkl. SP1.

Korrekturen zum Übertragen einer ODK-Anwendung in das Zielsystem für die Windows-Umgebung

Im Kapitel "ODK-Anwendung in das Zielsystem übertragen" ist der vom Registry Key vorgegebene Standardwert, der den Dateipfad für die Ablage der DLL-Dateien beschreibt, angegeben:

```
%ProgramData%\Siemens\Automation\ODK1500S\
```

Das Dateisystems der CPU 1515SP PC blendet den Ordner standardmäßig aus. Sie können sich den Ordner über die Windows-Option "Ausgeblendete Dateien, Ordner und Laufwerke anzeigen" im Explorer-Menü "Organisieren > Ordner- und Suchoptionen > Ansicht" anzeigen lassen.

Hinweis

Administratorrechte

Sie benötigen Administratorrechte, um auf diesen Ordner zuzugreifen. So kann verhindert werden, dass ODK-Anwendungen durch nicht berechtigte Personen eingespielt werden.

Korrektur zum Vorgehen vor dem ersten Debug-Vorgang (Echtzeitanwendung)

Das Kapitel "Debug (Test)" beschreibt die Installation von MinGW mit Hilfe des Kontextmenübefehls "Mit PowerShell ausführen". PowerShell wird für ODK 1500S nicht unterstützt.

Um einen Test zu einer Echtzeit-Anwendung in einer Windows-Umgebung durchzuführen, gehen Sie einmalig folgendermaßen vor:

1. Schließen Sie Eclipse.
2. Öffnen Sie den Ordner "bin" Ihrer ODK-Installation.
3. Führen Sie die Datei "MinGW32_Install.cmd" mit dem Kontextmenübefehl "Als Administrator ausführen" aus.
Ein Textausgabedialog wird geöffnet. Die Windows-Eingabeaufforderung installiert alle notwendigen Komponenten.
4. Folgen Sie den weiteren Schritten im Kapitel "Debug (Test)".

Korrekturen zum Entladen einer Funktion

Im Kapitel "Funktionen entladen" für die Entwicklung einer ODK-Anwendung für die Echtzeit-Umgebung sind im Absatz "Einführung" Gründe für das automatische Entladen aufgelistet. Folgende Punkte sind fälschlicherweise vermerkt:

- Neustart von Windows
- Abmelden des Windows-Benutzers (im Kontext eines Windows-Benutzers)

Im Kapitel "Funktionen entladen" für die Entwicklung einer ODK-Anwendung für die Windows-Umgebung sind im Absatz "Einführung" Gründe für das automatische Entladen aufgelistet. Folgender Punkt ist nicht vermerkt:

- Abmelden des Windows-Benutzers (im Kontext eines Windows-Benutzers)

Projektnamen mit Leerzeichen

Der Name Ihres ODK-Projektes darf keine Leerzeichen enthalten.

Übersicht über Mengengerüste

Mengengerüste für ODK-Anwendungen

ODK-Anwendung für die Windows-Umgebung:

- bis zu 32 parallele ODK-Anwendungen
- bis zu 32 parallele Funktionsaufrufe (gesamt)
- bis zu 32 KB Eingangsdaten pro Funktionsaufruf
- bis zu 32 KB Ausgangsdaten pro Funktionsaufruf

ODK-Anwendung für die Echtzeit-Umgebung:

- bis zu 32 parallele ODK-Anwendungen (abhängig vom verfügbaren Speicher)
- bis zu 3 parallele Funktionsaufrufe in eine ODK-Anwendung
- bis zu 26 parallele Funktionsaufrufe (gesamt)
- bis zu 32 KB Eingangsdaten pro Funktionsaufruf
- bis zu 32 KB Ausgangsdaten pro Funktionsaufruf

Link zu den ODK-Beispielprojekten

Um den Einstieg in die ODK-Thematik zu erleichtern, bietet Ihnen ODK 1500S Beispielprojekte für beide Entwicklungsumgebungen.

Die Beispielprojekte finden Sie zum Download im Internet (<http://support.automation.siemens.com/WW/view/de/106192387>).

Reparaturinstallation bei aktivem Eclipse

Führen Sie eine Reparaturinstallation nur durch, wenn die Entwicklungsumgebung Eclipse beendet ist.

Wenn Sie eine Reparaturinstallation durchführen, während Eclipse geöffnet ist, können Sie Eclipse danach nicht mehr nutzen. Schließen Sie Eclipse und führen Sie eine erneute Reparaturinstallation durch.

Verwendung von ODK_CLASSIC_DB

Verwendung von ODK_CLASSIC_DB als Parameter

Der Datentyp ODK_CLASSIC_DB darf nur mit dem InOut-Identifer [IN] und [INOUT] benutzt werden. Wird ein Parameter des der Datentyps ODK_CLASSIC_DB mit dem InOut-Identifer [IN] oder [INOUT] verwendet, darf kein weiterer Parameter, egal welchen Datentyps, mit dem gleichen InOut-Identifer verwendet werden.

Beispiel

```
// INTERFACE
...
// OK:
ODK_RESULT MyFunc1 ([IN] ODK_CLASSIC_DB myDB);
ODK_RESULT MyFunc2 ([IN] ODK_CLASSIC_DB myDB1, [INOUT] ODK_CLASSIC_DB myDB2);
//
// NOT OK (Code Generator will throw an error):
// ODK_CLASSIC_DB nicht für [OUT] zulässig
ODK_RESULT MyFunc3 ([OUT] ODK_CLASSIC_DB myDB);
// wenn ODK_CLASSIC_DB für [IN] verwendet, darf kein weiterer [IN] Parameter in dieser
// Funktion definiert werden
ODK_RESULT MyFunc4 ([IN] ODK_CLASSIC_DB myDB, [IN] ODK_INT32 myint);
```

Anwendungsbeispiel für C++

```
#include "ODK_CpuReadData.h"
...
ODK_RESULT MyFunc1 (const ODK_CLASSIC_DB& myDB)
{
    CODK_CpuReadData myReader (&myDB);
    ODK_INT32 myInt1, myInt2;

    myReader.ReadS7DINT(0, myInt1);
    myReader.ReadS7DINT(4, myInt2);

    return myInt1 + myInt2;
}
```

Um innerhalb einer Anwenderfunktion auf den Datentyp ODK_CLASSIC_DB zugreifen zu können, stehen Ihnen die Helfer-Funktionen der folgenden Klassen zur Verfügung:

- Klasse "CODK_CpuReadData"
- Klasse "CODK_CpuReadWriteData"

ODK_CLASSIC_DB mit Identifier [IN]

Wenn Sie den ODK-Datentyp "ODK_CLASSIC_DB" in einer ODK-Anwendung für die Echtzeitumgebung mit dem InOut-Identifer [IN] nutzen, werden die Daten nicht korrekt an die ODK-Anwendung übermittelt.

Verwenden Sie den InOut-Identifer [INOUT] mit dem ODK-Datentyp "ODK_CLASSIC_DB" in einer ODK-Anwendung für die Echtzeitumgebung.

Hinweis

Nutzung des InOut-Identifer [IN] ab CPU-Version V1.8

Ab der CPU-Version V1.8 können Sie den ODK-Datentyp "ODK_CLASSIC_DB" auch mit dem InOut-Identifer [IN] nutzen.

Verwendung von ODK_DTL

Nutzen Sie den ODK-Datentyp "ODK_DTL" **nur als letzten Parameter** je InOut-Identifizier in einer Funktion:

```
ODK_RESULT Sample1([IN] ODK_INT32 inInt, [IN] ODK_DTL inDTL
, [OUT] ODK_INT32 outInt, [OUT] ODK_DTL outDTL
, [INOUT] ODK_INT32 inoutInt, [INOUT] ODK_DTL inoutDTL);
```

Laden und Aufruf einer ODK-Anwendung

Um die Ausführung einer ODK-Anwendung in der Echtzeit-Umgebung zu gewährleisten, beachten Sie eine der folgenden Empfehlungen:

- das Laden der ODK-Anwendung sollte in einem StartupOB erfolgen
- während eines Ladevorgangs sollte keine ODK-Funktion ausgeführt werden

Paralleles Ausführen von ODK-Aufrufen in der Echtzeit-Umgebung

Wenn Sie die CPU-Version V1.7 nutzen, dürfen Sie nicht mehr als einen ODK-Aufruf gleichzeitig ausführen.

Code-Generator gibt keine Meldungen aus

Der Code-Generator erzeugt Fehlermeldungen für Ihr ODK-Projekt.

Wenn in der Datei <Projekt>.odk ein Array-Bereich nach dem Datentyp (z. B.: ODK_DOUBLE[10] myReal) anstatt nach dem Variablennamen definiert wird, dann kann der Code-Generator diese Datei nicht mehr verarbeiten.

STATUS-Meldungen

STATUS 0x8099 für Anweisung "Load" (Echtzeit-Umgebung)

Eine ODK-Anwendung wird durch den Aufruf der Anweisung "<STEP7Prefix>_Load" im STEP 7-Anwenderprogramm geladen. Über den Ausgangsparameter werden Informationen nach dem Ladevorgang zurückgeliefert.

Die folgende Tabelle zeigt die Informationen zum Wert STATUS 0x8099 für die Entwicklung einer ODK-Anwendung für die Echtzeit-Umgebung:

DONE	BUSY	ERROR	STATUS	Bedeutung
0	0	1	0x8099	CPU-Version V1.7: Die ODK-Anwendung konnte nicht geladen werden, weil der Aufruf der Anweisung nicht vom OB1 bzw. OB100 erfolgte. CPU-Version V1.8: Die ODK-Anwendung konnte nicht geladen werden, weil der Aufruf der Anweisung nicht in einem OB mit niedrigster Priorität erfolgt ist. Verwenden Sie einen Startup OB (z. B. OB100) oder einen Program cycle OB (z. B. OB1).

STATUS 0x80C2" für Anweisung "Load", "Execute" und "Unload" (Echtzeit-Umgebung)

Der Wert STATUS 0x80C2 existiert nicht für die Entwicklung einer ODK-Anwendung für die Echtzeit-Umgebung.

STATUS 0x80A4" für Anweisung "Unload" (Echtzeit-Umgebung)

Die folgende Tabelle zeigt die Informationen zum Wert STATUS 0x80A4 für die Entwicklung einer ODK-Anwendung für die Echtzeit-Umgebung:

DONE	BUSY	ERROR	STATUS	Bedeutung
0	0	1	0x80A4	ODK-Anwendung konnte nicht entladen werden. Bei der Ausführung der Funktion "OnUnload()" ist ein Kommunikationsfehler zwischen der CPU und ODK aufgetreten.

Helper-Funktionen

String-Helper-Funktionen für ODK-Anwendung für die Windows- und Echtzeit-Umgebung

Folgende Helper-Funktionen geben Zugriff auf S7Strings:

Helper-Funktion	Beschreibung
Convert_S7STRING_to_SZSTR	Konvertiert PLC-String-Typen in C/C++ String-Typen ("char" Array, Null-terminiert)
Convert_SZSTR_to_S7STRING	Konvertiert C/C++ String-Typen ("char" Array, Null-terminiert) in PLC-String-Typen.
Get_S7STRING_Length	Gibt die aktuelle Länge eines PLC-String-Typs zurück.
Get_S7STRING_MaxLength	Gibt die maximale Länge eines PLC-String-Typs zurück.

String-Helper-Funktionen für ODK-Anwendung für die Windows-Umgebung

Folgende Helper-Funktionen geben Zugriff auf S7WStrings:

Helper-Funktion	Beschreibung
Convert_S7WSTRING_to_SZWSTR	Konvertiert PLC-WString-Typen in C/C++ WString-Typen ("wchar_t" Array, Null-terminiert)
Convert_SZWSTR_to_S7WSTRING	Konvertiert C/C++ WString-Typen ("wchar_t" Array, Null-terminiert) in PLC-WString-Typen.
Get_S7WSTRING_Length	Gibt die aktuelle Länge eines PLC-WString-Typs zurück.
Get_S7WSTRING_MaxLength	Gibt die maximale Länge eines PLC-WString-Typs zurück.

Klasse "CODK_CpuReadData" (Windows- und Echtzeit-Umgebung)

Die Klasse "CODK_CpuReadData" erlaubt lesenden Zugriff auf Classic-DBs:

Wert	Beschreibung
CODK_CpuReadData	Klassen-Konstruktor: Initialisiert den Input-Daten-Bereich und die Datengröße.
ReadS7BYTE	Liest ein "byte" (1 Byte) aus dem Datenbereich.
ReadS7WORD	Liest ein "word" (2 Bytes) aus dem Datenbereich.
ReadS7DWORD	Liest ein "double word" (4 Bytes) aus dem Datenbereich.
ReadS7LWORD	Liest ein "long word" (8 Bytes) aus dem Datenbereich.
ReadS7S5TIME	Liest einen 16-Bit (2 Bytes) Zeitwert aus dem Datenbereich.
ReadS7DATE	Liest einen Datumswert (2 Bytes) aus dem Datenbereich.
ReadS7TIME_OF_DAY	Liest die Tageszeit (4 Bytes) aus dem Datenbereich.
ReadS7SINT	Liest ein "short integer" (1 Byte) aus dem Datenbereich.
ReadS7INT	Liest ein "integer" (2 Bytes) aus dem Datenbereich.
ReadS7DINT	Liest ein "double integer" (4 Bytes) aus dem Datenbereich.
ReadS7USINT	Liest ein "unsigned short integer" (1 Byte) aus dem Datenbereich.
ReadS7UINT	Liest ein "unsigned integer" (2 Bytes) aus dem Datenbereich.
ReadS7UDINT	Liest ein "unsigned double integer" (4 Bytes) aus dem Datenbereich.
ReadS7REAL	Liest ein "real number" (4 Bytes) aus dem Datenbereich.
ReadS7LREAL	Liest ein "long real number" (8 Bytes) aus dem Datenbereich.
ReadS7LINT	Liest ein "long integer" (8 Bytes) aus dem Datenbereich.
ReadS7ULINT	Liest ein "unsigned long integer" (8 Bytes) aus dem Datenbereich.
ReadS7TIME	Liest einen Zeitwert (4 Bytes) aus dem Datenbereich.
ReadS7CHAR	Liest ein "char" (1 Byte) aus dem Datenbereich.
ReadS7BOOL	Liest ein "bool" (1 Bit) aus dem Datenbereich.
ReadS7STRING_LEN	Liest die Information der Sting-Länge für einen S7-String im Datenbereich.
ReadS7STRING	List einen S7-String aus dem Datenbereich und gibt ihn als C++-Zeichen-String zurück.
ReadS7DATE_AND_TIME	Liest einen allgemeinen Datums- und Zeitbereich.

Klasse "CODK_CpuReadWriteData" (Windows- und Echtzeit-Umgebung)

Die Klasse "CODK_CpuReadWriteData" erlaubt zusätzlich zu allen lesenden Zugriffen von "CODK_CpuReadData" auf Classic-DBs auch die folgenden schreibenden Zugriffe:

Wert	Beschreibung
CODK_CpuReadWriteData	Klassen-Konstruktor: Initialisiert den Output-Daten-Bereich und die Datengröße.
WriteS7BYTE	Schreibt ein "byte" (1 Byte) in den Datenbereich.
WriteS7WORD	Schreibt ein "word" (2 Bytes) in den Datenbereich.
WriteS7DWORD	Schreibt ein "double word" (4 Bytes) in den Datenbereich.
WriteS7LWORD	Schreibt ein "long word" (8 Bytes) in den Datenbereich.
WriteS7SINT	Schreibt ein "short integer" (1 Byte) in den Datenbereich.
WriteS7INT	Schreibt ein "integer" (2 Bytes) in den Datenbereich.
WriteS7DINT	Schreibt ein "double integer" (4 Bytes) in den Datenbereich.
WriteS7USINT	Schreibt ein "unsigned short integer" (1 Byte) in den Datenbereich.
WriteS7UINT	Schreibt ein "unsigned integer" (2 Bytes) in den Datenbereich.
WriteS7UDINT	Schreibt ein "unsigned double integer" (4 Bytes) in den Datenbereich.
WriteS7S5TIME	Schreibt einen 16-Bit (2 Bytes) Zeitwert in den Datenbereich.
WriteS7TIME	Schreibt einen Zeitwert (4 Bytes) in den Datenbereich.
WriteS7DATE	Schreibt einen Datumswert (2 Bytes) in den Datenbereich.
WriteS7TIME_OF_DAY	Schreibt die Tageszeit (4 Bytes) in den Datenbereich.
WriteS7CHAR	Schreibt ein "char" (1 Byte) in den Datenbereich.
WriteS7REAL	Schreibt ein "real number" (4 Bytes) in den Datenbereich.
WriteS7LREAL	Schreibt ein "long real number" (8 Bytes) in den Datenbereich.
WriteS7LINT	Schreibt ein "long integer" (8 Bytes) in den Datenbereich.
WriteS7ULINT	Schreibt ein "unsigned long integer" (2 Bytes) in den Datenbereich.
WriteS7BOOL	Schreibt ein "bool" (1 Bit) in den Datenbereich.
WriteS7STRING	Schreibt einen S7-String in den Datenbereich.
WriteS7DATE_AND_TIME	Schreibt Datums- und Zeitdaten in den Datums- und Zeitbereich.

Mögliche Meldungen

Programm "g++" / "gcc" wurde nicht gefunden

Hinweis

Nutzung von Eclipse nur als "Administrator"

Wenn Sie die Entwicklungsumgebung Eclipse nutzen möchten, müssen Sie als Administrator bei Windows angemeldet sein. Wenn Sie als Standard-Nutzer angemeldet sind, sind bestimmte Bereiche für Sie schreibgeschützt. Damit kann es bei Nutzung von Eclipse zu Fehlern kommen.

Die Meldungen "Program g++ not found in PATH" und "Program gcc not found in PATH" können folgende Gründe haben:

- **MinGW noch nicht installiert**
Die Meldung erscheint, wenn MinGW noch nicht installiert wurde.
 - **Nutzung von Eclipse mit neuem Workspace**
Die Meldung erscheint, wenn Sie die Entwicklungsumgebung Eclipse nutzen und einen neuen Workspace anlegen, der nicht dem vorgeschlagenen Workspace entspricht.
Verwenden Sie das von ODK 1500S installierte Eclipse mit voreingestelltem Workspace, um diese Meldungen zu vermeiden.
Falls Sie den Workspace an einen anderen Ablageort legen müssen, achten Sie darauf, dass Sie den gesamten Workspace umkopieren.
-

Hinweis

Meldungen haben keine Auswirkung auf "Release"-Version

Wenn Sie eine "Release"-Version bauen, haben die Meldungen "Program g++ not found in PATH" und "Program gcc not found in PATH" keine Auswirkungen. Sie können diese Meldungen ignorieren.

Programm "rm" wurde nicht gefunden

Die Meldung "Program rm not found in PATH" hat folgenden Grund.

Projekt säubern

Die Meldung erscheint, wenn Sie die Entwicklungsumgebung Eclipse nutzen und Ihr Projekt mit Hilfe des Befehls "Clean..." im Menü "Project" säubern möchten.

Die Meldung hat keine Auswirkung auf die Durchführung des Vorgangs. Ihr Projekt wird trotz der Meldung gesäubert.

SIEMENS

SIMATIC

S7-1500 Software Controller

Amendments to ODK 1500S V.10

Product Information

Introduction

General information

This product information contains important information about the ODK 1500S V1.0. The product information is part of the delivered product and the contained statements are when in doubt more binding than other statements.

Scope of validity of the product information

This product information applies to the use of ODK 1500S with the following products:

- CPU 1505S
- CPU 1507S

Changes refer to the manual "Open Development Kit 1500S V1.0" (A5E35253935-AA).

Service & Support on the Internet

A guide to the technical documentation offered for the various SIMATIC products and systems is available on the Internet (<http://www.siemens.com/simatic-tech-doku-portal>).

In addition to our documentation, we offer our complete online knowledge base on the Internet (<http://www.siemens.com/automation/service&support>). There you will find:

- The newsletter, which is continuously updated to provide you with the latest information about your products
- The right documents via our search function under Service & Support
- The bulletin board, worldwide knowledge exchange for users and experts
- Your local representative for Automation & Drives in our contact database
- Information about on-site services, repairs, spare parts, and lots more

Additional information

For additional information and documentation on SIMATIC Embedded Bundles, please see the product support on the Internet (<http://support.automation.siemens.com/WWW/view/en/42783973/133300>).

Security information

Security information

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. You can find more information about industrial security on the Internet (<http://www.siemens.com/industrialsecurity>).

Notes on Open Source software

You can download and install various development tools of the Open Source Project MinGW32 (<http://www.mingw.org/>) as add-ons to the product. Please note that these components are optional and provided and marketed by MinGW32 project or other licensors subject to various licenses that can be consulted on the project homepage or in the respective packages.

Notes

Amendments to the documentation

System requirements - Corrections

An ODK application for the Windows environments requires Microsoft Visual Studio 2010 incl. SP1.

Corrections for transferring an ODK application to the target system for Windows environment

The default value specified by the registry key, which describes the data path for storing the DLL files, is specified in the section "Transferring an ODK application to the target system":

```
%ProgramData%\Siemens\Automation\ODK1500S\
```

The file system of the CPU 1515SP PC hides the folder by default. You can display the folder using the Windows option "Show hidden files, folders and drives" in the Explorer menu "Organize > Folders and search options > View".

Note

Administrator rights

You need administrator rights in order to access this folder. This prevents ODK applications from being loaded by unauthorized persons.

Amendment to procedure before the first debug process (real-time application)

The "Debug (Test)" chapter describes the installation of MinGW with the help of the shortcut menu command "Execute with PowerShell". PowerShell is not supported for ODK 1500S.

To perform a test on a real-time application in a Windows environment, perform the following once:

1. Close Eclipse.
2. Open the "bin" folder of your ODK installation.
3. Run the "MinGW32_Install.cmd" file with the shortcut menu command "Run as administrator".
A text editing dialog opens. The Windows command prompt installs all required components.
4. Follow the additional steps in the "Debug (Test)" chapter.

Unloading a function - Corrections

For the development of an ODK application for the real-time environment, reasons for automatic unloading are listed in the "Introduction" section of the "Unload functions" chapter. The following items are incorrectly noted:

- Windows restart
- Windows user logs off (in the context of a Windows user)

For the development of an ODK application for the Windows environment, reasons for automatic unloading are listed in the "Introduction" section of the "Unload functions" chapter. The following item is not noted:

- Windows user logs off (in the context of a Windows user)

Project name with blanks

The name of your ODK project must not contain any blanks.

Overview of the quantity frameworks

Quantity frameworks for ODK applications

ODK application for the Windows environment:

- Up to 32 parallel ODK applications
- Up to 32 parallel function calls (in total)
- Up to 32 KB input data per function call
- Up to 32 KB output data per function call

ODK application for the real-time environment:

- Up to 32 parallel ODK applications (depending on the available memory)
- Up to 3 parallel function calls in an ODK application
- Up to 26 parallel function calls (in total)
- Up to 32 KB input data per function call
- Up to 32 KB output data per function call

Link to the ODK sample projects

To facilitate your introduction to the topic of ODK, ODK 1500S offers example projects for both development environments.

To download the sample projects, go to on the Internet (<http://support.automation.siemens.com/WW/view/en/106192387>).

Repair installation with active Eclipse

Only perform the repair installation after the Eclipse development environment has been closed.

If you perform the repair installation while Eclipse is open you will no longer be able to use Eclipse after this. Close Eclipse and run the repair installation again.

Using ODK_Classic_DB

Using ODK_Classic_DB as parameter

The data type ODK_CLASSIC_DB may be used only with the InOut-Identifier [IN] and [INOUT]. When a parameter of the data type ODK_CLASSIC_DB is used with the InOut-Identifier [IN] or [INOUT], it is not permitted to use other parameters with the same InOut-Identifier, regardless of data type.

Example

```
// INTERFACE
...
// OK:
ODK_RESULT MyFunc1 ([IN] ODK_CLASSIC_DB myDB);
ODK_RESULT MyFunc2 ([IN] ODK_CLASSIC_DB myDB1, [INOUT] ODK_CLASSIC_DB myDB2);
//
// NOT OK (Code Generator will throw an error):
// ODK_CLASSIC_DB not permitted for [OUT]
ODK_RESULT MyFunc3 ([OUT] ODK_CLASSIC_DB myDB);
// If ODK_CLASSIC_DB is used for [IN], no other [IN] parameter may be defined in this
// function
ODK_RESULT MyFunc4 ([IN] ODK_CLASSIC_DB myDB, [IN] ODK_INT32 myint);
```

Application example for C++

```
#include "ODK_CpuReadData.h"
...
ODK_RESULT MyFunc1 (const ODK_CLASSIC_DB& myDB)
{
    CODK_CpuReadData myReader (&myDB);
    ODK_INT32 myInt1, myInt2;

    myReader.ReadS7DINT(0, myInt1);
    myReader.ReadS7DINT(4, myInt2);

    return myInt1 + myInt2;
}
```

Helper functions of the following classes are available to help you access the data type ODK_CLASSIC_DB inside a user function:

- Class "CODK_CpuReadData"
- Class "CODK_CpuReadWriteData"

ODK_CLASSIC_DB with identifier [IN]

When you use the ODK data type "ODK_CLASSIC_DB" in an ODK application for the real-time environment with the InOut-Identifier [IN], the data is not transferred correctly to the ODK application.

Use the InOut identifier [INOUT] with the ODK data type "ODK_CLASSIC_DB" in an ODK application for the real-time environment.

Note

Using the InOut identifier [IN], CPU version V1.8 and higher

As of CPU version V1.8, you can also use the ODK data type "ODK_CLASSIC_DB" with the InOut identifier [IN].

Using ODK_DTL

Use the ODK data type "ODK_DTL" **only as last parameter** je InOut-Identifier in a function:

```
ODK_RESULT Sample1([IN] ODK_INT32 inInt, [IN] ODK_DTL inDTL
                  , [OUT] ODK_INT32 outInt, [OUT] ODK_DTL outDTL
                  , [INOUT] ODK_INT32 inoutInt, [INOUT] ODK_DTL inoutDTL);
```

Loading and calling an ODK application

To ensure the execution of an ODK application in the real-time environment, observe the following recommendations:

- The ODK application should be loaded in a StartupOB.
- Do not execute any ODK function during a loading process.

Executing ODK calls parallel in the real-time-environment

When using CPU version V1.7, you must not execute more than one ODK call at the same time.

Code generator does not output any messages

The code generator generates error messages for your ODK project.

If an array range is defined after the data type (for example: ODK_DOUBLE[10] myReal) instead of after the tag name, the code generator can no longer process this file.

Status messages

Status 0x8099 for "Load" instruction (real-time-environment)

An ODK application is loaded by calling the "<STEP7Prefix>_Load" instruction in the STEP 7 user program. Information is returned through the output parameter after the load process.

The following table shows information on the value STATUS 0x8099 for the development of an ODK application for the real-time environment:

DONE	BUSY	ERROR	STATUS	Meaning
0	0	1	0x8099	CPU version V1.7: ODK application could not be loaded because the instruction was not called from OB1 or OB100. CPU version V1.8: The ODK application could not be loaded because the instruction was not called in an OB with lowest priority. Use a startup OB (such as OB100) or a program cycle OB (for example, OB1).

Status 0x80C2" for "Load", "Execute" and "Unload" instruction (real-time environment)

The value STATUS 0x80C2 does not exist for the development of an ODK application for the real-time environment.

Status 0x80A4" for "Unload" instruction (real-time-environment)

The table below shows the information on the value STATUS 0x80A4 for the development of an ODK application for the real-time environment:

DONE	BUSY	ERROR	STATUS	Meaning
0	0	1	0x80A4	ODK application could not be unloaded. A communication error between the CPU and ODK occurred during execution of the "OnUnload()" function.

Helper functions

String-helper functions for ODB application for Windows and real-time environment

The following helper functions provide access to S7 strings:

Helper functions	Description
Convert_S7STRING_to_SZSTR	Convert PLC string types to C/C++ string types ("char" array, null-terminated)
Convert_SZSTR_to_S7STRING	Convert C/C++ string types ("char" array, null-terminated) to PLC string types.
Get_S7STRING_Length	Returns the current length of a PLC string type.
Get_S7STRING_MaxLength	Returns the maximum length of a PLC string type.

String-helper functions for ODB application for the Windows environment

The following helper functions provide access to S7WStrings:

Helper functions	Description
Convert_S7WSTRING_to_SZWSTR	Convert PLC WString types to C/C++ WString types ("wchar_t" array, null-terminated)
Convert_SZWSTR_to_S7WSTRING	Convert C/C++ WString types ("wchar_t" array, null-terminated) to PLC WString types.
Get_S7WSTRING_Length	Returns the current length of a PLC Wstring type.
Get_S7WSTRING_MaxLength	Returns the maximum length of a PLC WString type.

Class "CODK_CpuReadData" (Windows and real-time environment)

The class "CODK_CpuReadData" allows read access to Classic DBs:

Value	Description
CODK_CpuReadData	Class constructor: Initializes the input data area and the data size.
ReadS7BYTE	Reads a "byte" (1 byte) from the data area.
ReadS7WORD	Reads a "word" (2 bytes) from the data area.
ReadS7DWORD	Reads a "double word" (4 bytes) from the data area.
ReadS7LWORD	Reads a "long word" (8 bytes) from the data area.
ReadS7S5TIME	Reads a "16-bit" (2 bytes) time value from the data area.
ReadS7DATE	Reads a date value (2 bytes) from the data area.
ReadS7TIME_OF_DAY	Reads the time of day (4 bytes) from the data area.
ReadS7SINT	Reads a "short integer" (1 byte) from the data area.
ReadS7INT	Reads a "integer" (2 bytes) from the data area.
ReadS7DINT	Reads a "double integer" (4 bytes) from the data area.
ReadS7USINT	Reads a "unsigned short integer" (1 byte) from the data area.
ReadS7UINT	Reads a "unsigned integer" (2 bytes) from the data area.
ReadS7UDINT	Reads a "unsigned double integer" (4 bytes) from the data area.
ReadS7REAL	Reads a "real number" (4 bytes) from the data area.
ReadS7LREAL	Reads a "long real number" (8 bytes) from the data area.
ReadS7LINT	Reads a "long integer" (8 bytes) from the data area.
ReadS7ULINT	Reads a "unsigned long integer" (8 bytes) from the data area.
ReadS7TIME	Reads a time value (4 bytes) from the data area.
ReadS7CHAR	Reads a "char" (1 byte) from the data area.
ReadS7BOOL	Reads a "bool" (1 byte) from the data area.
ReadS7STRING_LEN	Reads the information of string length from an S7 string in the data area.
ReadS7STRING	Reads an S7 string from the data area and returns it as C++ character string.
ReadS7DATE_AND_TIME	Reads the general data and time area.

Class "CODK_CpuReadWriteData" (Windows and real-time environment)

The class "CODK_CpuReadWriteData" also allows all access of "CODK_CpuReadData" to Classic-DBs, as well as the following write access:

Value	Description
CODK_CpuReadWriteData	Class constructor: Initializes the output data area and the data size.
WriteS7BYTE	Writes a "byte" (1 byte) to the data area.
WriteS7WORD	Writes a "word" (2 bytes) to the data area.
WriteS7DWORD	Writes a "double word" (4 bytes) to the data area.
WriteS7LWORD	Writes a "long word" (8 bytes) to the data area.
WriteS7SINT	Writes a "short integer" (1 byte) to the data area.
WriteS7INT	Writes a "integer" (2 bytes) to the data area.
WriteS7DINT	Writes a "double integer" (4 bytes) to the data area.
WriteS7USINT	Writes a "unsigned short integer" (1 byte) to the data area.
WriteS7UINT	Writes a "unsigned integer" (2 bytes) to the data area.
WriteS7UDINT	Writes a "unsigned double integer" (4 bytes) to the data area.
WriteS7S5TIME	Writes a 16-bit (2 bytes) time value to the data area.
WriteS7TIME	Writes a time value (4 bytes) to the data area.
WriteS7DATE	Writes a date value (2 bytes) to the data area.
WriteS7TIME_OF_DAY	Writes a time of day (4 bytes) to the data area.
WriteS7CHAR	Writes a "char" (1 byte) to the data area.
WriteS7REAL	Writes a "real number" (4 bytes) to the data area.
WriteS7LREAL	Writes a "long real number" (8 bytes) to the data area.
WriteS7LINT	Writes a "long integer" (8 bytes) to the data area.
WriteS7ULINT	Writes a "unsigned long integer" (2 bytes) to the data area.
WriteS7BOOL	Writes a "bool" (1 byte) to the data area.
WriteS7STRING	Writes a S7 string to the data area.
WriteS7DATE_AND_TIME	Write data and time data to the date and time area.

Possible messages

Program "g++" / "gcc" was not found

Note

Use of Eclipse only as "Administrator"

If you want to use the Eclipse development environment, you must be logged on as administrator. If you are logged on as standard user, certain areas are read-only. This can lead to errors when using Eclipse.

The messages "Program g++ not found in PATH" and "Program gcc not found in PATH" can have following reasons:

- **MinGW not yet installed**
The message appears if MinGW has not been installed yet.
 - **Use of Eclipse with new workspace**
This message appears if you use the Eclipse development environment and create a new workspace that does not match the proposed workspace.
To avoid these messages, use the Eclipse with preset workspace installed by ODK 1500S.
If you have to place the workspace at a different storage location, make sure that you copy over the entire workspace.
-

Note

Messages have no effect on "Release" version

When you build a "Release" version, the messages "Program g++ not found in PATH" and "Program gcc not found in PATH" have no effect. You can ignore these messages.

Program "rm" was not found

The message "Program rm not found in PATH" has following reason.

Clean project

The message appears if you use the Eclipse development environment and want to clean your project with the help of the command "Clean..." in the "Project" menu.

The message does not affect the execution of the process. Your project will be cleaned despite the message.