Documentation of the library
WagoAppSiemensS7Protocol

Release 1.6.1.0
Description

This document is automatically generated. Because of this, the chapter 30 Visualization is not shown in this document. If you are interested in getting to know more about visualization, we refer to the library manager of e!Cockpit.

Subject to Changes

WAGO Kontakttechnik GmbH Co. KG reserves the right to provide for any alterations or modifications. WAGO Kontakttechnik GmbH Co. KG owns all rights arising from the granting of patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

Personnel Qualification

All tasks that are carried out with libraries made for the e!COCKPIT software must only be performed by qualified electrical specialists instructed in PLC programming according to IEC 61131-3.

All tasks that have an effect on the properties or the behavior of automation hardware or software products must only be performed by qualified employees with a thorough knowledge of handling the products concerned.

Intended Use of e!COCKPIT Libraries

Libraries created for the e!COCKPIT software are used to simplify the development of application projects in the IEC 61131-3 programming languages.

For automation tasks, WAGO offers programmable logic controllers in a wide variety of performance classes. In combination with a wide range of I/O modules, the controllers can process standard types of field signals. Controllers can be implemented centrally or in decentralized configurations. The controllers offer interfaces for the most commonly used fieldbuses for use in decentralized configurations. Fieldbus independent I/O modules are then linked via fieldbus couplers. WAGO controllers offer a runtime environment for user programs called e!RUNTIME. Software projects for implementation in e!RUNTIME environments can be created in e!COCKPIT. The programming environment in e!COCKPIT is based on the established CODESYS 3 industrial standard. Users with a previous knowledge of CODESYS 3 will thus find this environment largely familiar. The following programming languages of the IEC 61131-3 standard are available:

- Structured Text (ST)
- Ladder Diagram (LD)
- Function Block Diagram (FBD)
- Instruction List (IL)
- Sequential Function Chart (SFC)
- Continuous Function Chart (CFC)

The individual programming languages can also be combined as required during the development of the software. A portfolio of prepared libraries can be accessed for many frequently used functions in order to make software development more efficient. This document provides an overview of the WagoAppSiemensS7Protocol that WAGO offers for e!COCKPIT.
Data exchange with a Siemens plc

Further library information are summerized here:

- **Company**  WAGO
- **Title**  WagoAppSiemensS7Protocol
- **Version**  1.6.1.0
- **Categories**  WAGO LayerView|App
- **Author**  Wago/u010663
- **Placeholder**  WagoAppSiemensS7Protocol

---

1 Based on WagoAppSiemensS7Protocol.library, last modified 13.08.2019, 20:59:16.
The content of this file was automatically generated with None on 13.08.2019, 20:59:18
2.1 DataConversion

2.1.1 Fu_DB_DWORD_READ_S7_TO_CoDeSys (FUN)

Interface variables

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return</td>
<td>Fu_DB_DWORD_READ_S7_TO_CoDeSys</td>
<td>DWORD</td>
</tr>
<tr>
<td>Input</td>
<td>pIn</td>
<td>POINTER TO ARRAY [0..3] OF BYTE</td>
</tr>
</tbody>
</table>

2.1.2 Fu_DB_DWORD_WRITE_CoDeSys_TO_S7 (FUN)

Interface variables

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return</td>
<td>Fu_DB_DWORD_WRITE_CoDeSys_TO_S7</td>
<td>DWORD</td>
</tr>
<tr>
<td>Input</td>
<td>dwIn</td>
<td>DWORD</td>
</tr>
</tbody>
</table>

2.1.3 Fu_DB_REAL_READ_S7_TO_CoDeSys (FUN)

Interface variables

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return</td>
<td>Fu_DB_REAL_READ_S7_TO_CoDeSys</td>
<td>REAL</td>
</tr>
<tr>
<td>Input</td>
<td>pIn</td>
<td>POINTER TO ARRAY [0..3] OF BYTE</td>
</tr>
</tbody>
</table>

2.1.4 Fu_DB_REAL_WRITE_CoDeSys_TO_S7 (FUN)

Interface variables

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return</td>
<td>Fu_DB_REAL_WRITE_CoDeSys_TO_S7</td>
<td>REAL</td>
</tr>
<tr>
<td>Input</td>
<td>rIn</td>
<td>REAL</td>
</tr>
</tbody>
</table>
2.1.5 Fu_DB_WORD_READ_S7_TO_CoDeSys (FUN)

Interface variables

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return</td>
<td>Fu_DB_WORD_READ_S7_TO_CoDeSys</td>
<td>WORD</td>
</tr>
<tr>
<td>Input</td>
<td>pIn</td>
<td>POINTER TO ARRAY [0..1] OF BYTE</td>
</tr>
</tbody>
</table>

2.1.6 Fu_DB_WORD_WRITE_CoDeSys_TO_S7 (FUN)

Interface variables

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return</td>
<td>Fu_DB_WORD_WRITE_CoDeSys_TO_S7</td>
<td>WORD</td>
</tr>
<tr>
<td>Input</td>
<td>wIn</td>
<td>WORD</td>
</tr>
</tbody>
</table>

2.2 Fb_DB_ReadWrite (FB)

Interface variables

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>xOpen</td>
<td>BOOL</td>
<td>Open connection to S7 plc</td>
</tr>
<tr>
<td></td>
<td>sIP</td>
<td>STRING</td>
<td>IP address of the S7 plc</td>
</tr>
<tr>
<td></td>
<td>wDB_Number</td>
<td>WORD</td>
<td>DB number in S7 plc, e.g. DB2-&gt;2</td>
</tr>
<tr>
<td></td>
<td>bType</td>
<td>BYTE</td>
<td>Access type: 2=Byte, 4=Word, 6=DWord, 8=REAL</td>
</tr>
<tr>
<td></td>
<td>wOffset</td>
<td>WORD</td>
<td>Offset within the DB, e.g. DBW4-&gt;4</td>
</tr>
<tr>
<td></td>
<td>wCount</td>
<td>WORD</td>
<td>Amount of data for read or write command</td>
</tr>
<tr>
<td></td>
<td>pWriteData</td>
<td>POINTER TO ARRAY [0..65535] OF BYTE</td>
<td>Pointer to the data which should be written in case of write commands</td>
</tr>
<tr>
<td></td>
<td>xWriteDB</td>
<td>BOOL</td>
<td>Differentiates between read and write commands, False: read, True: write</td>
</tr>
<tr>
<td></td>
<td>bTsap</td>
<td>BYTE</td>
<td>0:=CPU, 2=CP on slot 4</td>
</tr>
<tr>
<td>Inout</td>
<td>xTrigger</td>
<td>BOOL</td>
<td>Execute command. Will be reset by function block</td>
</tr>
<tr>
<td></td>
<td>aReadData</td>
<td>ARRAY [0..DB_MAX_DBSIZE] OF BYTE</td>
<td>Variable to store received data in case of read commands</td>
</tr>
<tr>
<td>Output</td>
<td>xConnected</td>
<td>BOOL</td>
<td>Connection to the S7 is established</td>
</tr>
<tr>
<td></td>
<td>xDone</td>
<td>BOOL</td>
<td>Job successful executed</td>
</tr>
<tr>
<td></td>
<td>xError</td>
<td>BOOL</td>
<td>Job with error executed</td>
</tr>
<tr>
<td></td>
<td>oStatus</td>
<td>WagoSysErrorBase.FbResult</td>
<td>Detailed Status information</td>
</tr>
</tbody>
</table>

Function

Data transfer according to S7 protocol

Graphical Illustration
Function Description

This function block allows to read and write data from a Siemens plc. The maximum amount of data is 200 Byte. The received data will be stored in aReadData. The outputs display either successful execution of the command or error details.

-oStatus: aReadData array too small->Increase the global constant DB_MAX_DBSIZE in the parameter list
Server response error.DB data invalid->the DB in the plc is either not available or the area within the DB is too large(DB size is 10 byte and a read command asks for 15 bytes)
MAX_SEND_TCP_CLIENT_ToSmall->reduce wDB_Length or increase global constant DB_MAX_DBSIZE
Write command failed ->the DB in the plc is either not available or the area within the DB is too large

..note:

Using Siemens TIA Portal may need the following settings

In the DB preferences the ‘Optimized block access’ has to be disabled.
Depending on the PLC, e.g. a S7-1200, the ‘Permit access with PUT/GET communication from remote partner’ has to be enabled in the PLC preferences under ‘Protection’.

Examples

Example 1: Read 3 word from DB10.DBW8
-wDB_Number=10
-bType=4
-wOffset=8
-wCount=3
-xWriteDB=FALSE

The data from the Siemens plc will be copied to the variable aReadData. The function Fu_DB_WORD_READ_S7_TO_CoDeSys may be used to convert the value from S7 to CoDeSys.

Example 2: Write 5 bytes to DB7.DBB4
-wDB_Number=7
-bType=2
-wOffset=4
-wCount=5
-xWriteDB=TRUE

2.2. Fb_DB_ReadWrite (FB)
The data which should be written has to be placed at the input pWriteData=adr(WriteData) (Write-
Data:ARRAY[0..4] OF Byte).

Example 3: Write 2 real to DB7.DBD0 and DB7.DBD4
-wDB_Number=7
-bType=6 (handle real like dword)
-wOffset=0
-wCount=2
-xWriteDB=TRUE

The data which should be written has to be placed at the input pWriteData=adr(WriteData) (Write-
Data:ARRAY[0..1] OF REAL). Please use Fu_DB_REAL_WRITE_CoDeSys_TO_S7 to convert the
real variable: WriteData[0]:= Fu_DB_REAL_WRITE_CoDeSys_TO_S7(myReal1) WriteData[1]:= Fu_DB_REAL_WRITE_CoDeSys_TO_S7(myReal2)
3.1 Status (GVL)

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
</tr>
</thead>
</table>

3.2 eStatus (ENUM)

<table>
<thead>
<tr>
<th>InOut:</th>
<th>Name</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OK</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WriteCommandFailed</td>
<td>1</td>
<td>DB invalid or DB to small</td>
</tr>
<tr>
<td></td>
<td>MAX_SEND_TCP_CLIENT_ToSmall</td>
<td>2</td>
<td>Transmit data to large, increase DB_MAX_DBSIZE in parameter list</td>
</tr>
<tr>
<td></td>
<td>ServerResponseError</td>
<td>3</td>
<td>DB invalid or DB to small</td>
</tr>
<tr>
<td></td>
<td>aReadData_ToSmall</td>
<td>4</td>
<td>Input aReadData to small to allow receiving all data which has been requested</td>
</tr>
</tbody>
</table>

Description: status codes
## ParameterList (PARAMS)

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>DB_MAX,DBSIZE</td>
<td>UINT</td>
<td>100</td>
<td>1..200</td>
</tr>
</tbody>
</table>
### VersionHistory (GVL)

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info</td>
<td>ProjectInfo</td>
<td>Version 1.6.0.0</td>
</tr>
</tbody>
</table>

**WagoAppSiemensS7Protocol.library**

<table>
<thead>
<tr>
<th>date</th>
<th>version</th>
<th>author</th>
<th>change</th>
</tr>
</thead>
<tbody>
<tr>
<td>08.01.2019</td>
<td>1.6.1.0</td>
<td>u015842</td>
<td>Properties: free placeholder added</td>
</tr>
<tr>
<td>04.03.2016</td>
<td>1.6.0.0</td>
<td>u010663</td>
<td>WagoSysErrorBase integrated</td>
</tr>
<tr>
<td>02.10.2015</td>
<td>1.0.0.1</td>
<td>u010663</td>
<td>Improvement, placeholder</td>
</tr>
<tr>
<td>16.09.2015</td>
<td>1.0.0.0</td>
<td>u010663</td>
<td>RC</td>
</tr>
<tr>
<td>26.06.2015</td>
<td>0.0.0.1</td>
<td>u010663</td>
<td>init</td>
</tr>
</tbody>
</table>

**Release Notes:**
This is a dictionary of all referenced libraries and their name spaces.

**Standard**

*Library Identification:*

Placeholder: Standard

Default Resolution: Standard, * (System)

Namespace: Standard

*Library Properties:*

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: False
- Optional: False

**WagoAppSocket**

*Library Identification:*

Placeholder: WagoAppSocket

Default Resolution: WagoAppSocket, * (WAGO)

Namespace: WagoAppSocket

*Library Properties:*

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: False
- Optional: False

*Library Parameter:*

Parameter: UIMULTICONNECT_NINSTANCELISTSIZE = 20

Parameter: UIBACKLOGDEFAULT = 5

Parameter: TMULTICONNECT_TIMEOUT = TIME#0ms

**WagoSysErrorBase**

*Library Identification:*

---
Placeholder: WagoSysErrorBase
Default Resolution: WagoSysErrorBase, * (WAGO)
Namespace: WagoSysErrorBase

**Library Properties:**

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: False
- Optional: False

**Library Parameter:**

Parameter: RES_LOG_MAX_FILESIZE = 2000
Parameter: RES_LOG_MAX_FILES = 1
Parameter: RES_LOG_MAX_ENTRIES = 200
Parameter: RES_LOG_NAME = ‘WagoAppResultLogger’

**WagoSysVersion**

**Library Identification:**
Name: WagoSysVersion
Version: 1.0.0.0
Company: WAGO
Namespace: WagoSysVersion

**Library Properties:**

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: False
- Optional: False

**WagoTypesErrorBase**

**Library Identification:**
Placeholder: WagoTypesErrorBase
Default Resolution: WagoTypesErrorBase, * (WAGO)
Namespace: WagoTypesErrorBase

**Library Properties:**

- LinkAllContent: False
- Optional: False
- QualifiedOnly: True
- SystemLibrary: False
- PublishSymbolsInContainer: True

© WAGO Kontaktechnik GmbH & Co. KG, Germany 2018 – All rights reserved. For the avoidance of doubt, this copyright notice does not only apply to the information above but also and primarily to the described library itself. Please note that third-party products are always mentioned without reference to intellectual property rights,
including patents, utility models, designs and trademarks, accordingly the existence of such rights cannot be excluded. WAGO is a registered trademark of WAGO Verwaltungsgesellschaft mbH.