WinAC

OPC-Client

User documentation

V1.2 • November 2009

Applikationen & Tools

Answers for industry.

SIEMENS

Industry Automation and Drives Technologies Service & Support Portal

This article is taken from the Service Portal of Siemens AG, Industry Automation and Drives Technologies. The following link takes you directly to the download page of this document.

http://support.automation.siemens.com/WW/view/en/48355169

If you have any questions concerning this document please e-mail us to the following address:

online-support.automation@siemens.com

SIEMENS

SIMATIC
WinAC OPC-Client

Warranty and Liability

Note

The Application Examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These application examples do not relieve you of the responsibility to use safe practices in application, installation, operation and maintenance. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time without prior notice.
If there are any deviations between the recommendations provided in these application examples and other Siemens publications – e.g. Catalogs – the contents of the other documents have priority.

We do not accept any liability for the information contained in this document.

Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (“Produkthaftungsgesetz”), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract (“wesentliche Vertragspflichten”). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

Table of Contents

Instruction

Content

The document describes the software **WinAC OPC-Client** for the user.

1 Basic information

1.1 Description of the problem

SIMATIC NET OPC Server allows any OPC client application to access data on a WinAC controller. WinAC RTX is not able to connect to other OPC Servers by default.

This function is now implemented as **WinAC OPC-Client** and enables the WinAC RTX PLC to communicate with any other OPC Server.

The only restriction is that the OPC server has to be a local server on the target system.

1.2 Reference system

The application described in this documentation is based on the following reference system:

Hardware:	SIMATIC Field PG M II
Software:	WinAC RTX-F 2009
	SIMATIC NET 6.3 (only for testing purposes as OPC Server)
	SIMATIC Manager V5.4, SP4

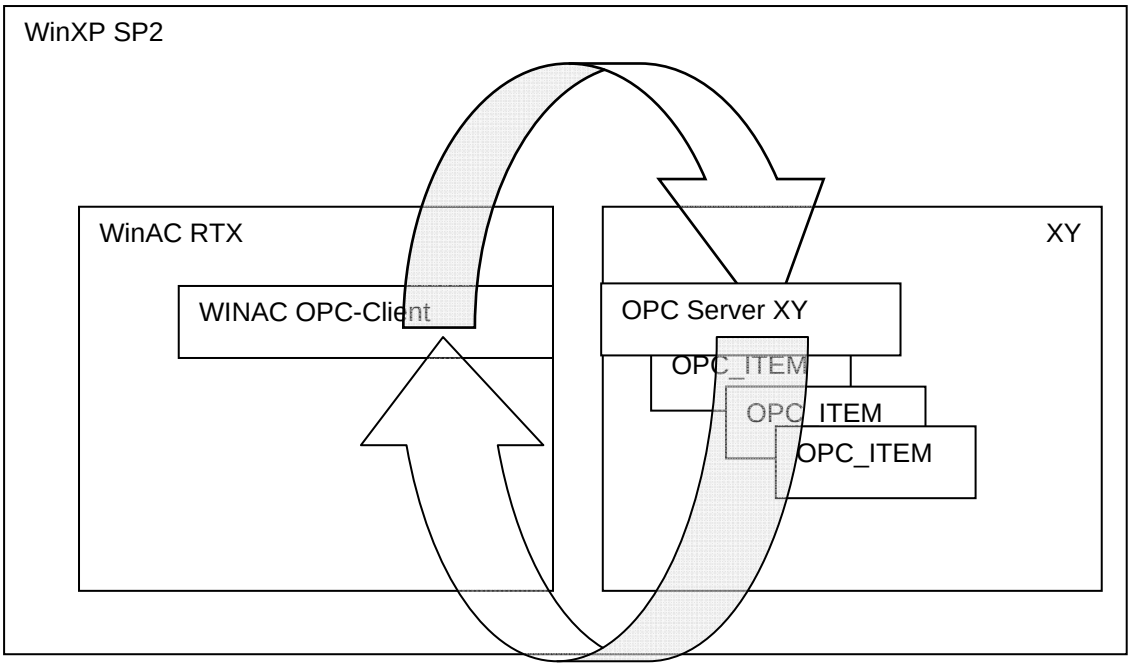
2 Overview

2.1 Functional Range

The WinAC OPC Client allows WinAC to communicate with any other local OPC Server that is installed on the target system. Only local OPC Servers are supported.

It is possible to register up to 4000 variables of different types with read and/or write access at the OPC Server

Figure 2-1 Overview



3 Installation

3.1 Quickstart

The following steps will guide you through the installation of **WinAC OPC-Client.**

- Copy ‘WinAC_OPCClient.dll’ to **system32**-folder. (use ‘setup.bat’)
- Copy the following components from the demo project to the user project:
  - FBs 65000-65007 including their-Instance-DBs UDT 4
  - SFBs 65001 and 65002
- Create an OPC Client parameter DB and parameterize it creating OPC Server Name and OPC Items (UDT 4) or adapt the given example DB (DB2000)
- Execute FB65000 OPC_INIT
- Call FB65003 OPC_CONN and FB65004 OPC_ADD_ITEM with the values stored in parameter DB
- As soon as the connection is established and all items are registered, proceed with cyclic reading or writing items with FB65006 OPC_READ and FB65007 OPC_WRITE.

4 The user interface

The following S7-Function blocks serve as user interface

- FB65000 OPC_INIT → Initialize WinAC OPC Client
- FB65003 OPC_CONN → Connect to OPC-Server
- FB65004 OPC_ADD_ITEM → Register OPC Items
- FB65006 OPC_READ → Read all 'READ' Values
- FB65007 OPC_WRITE → Write all 'WRITE' Values

All OPC Items belong to a user defined data type:

- UDT4 OPC_ITEM_DATA → Parameter for a OPC Value

The respective number of any FB or UDT can be adapted according to your requirements.

4.1 Initialize FB65000 OPC_INIT

This FB initializes ODK-DLL 'WinAC_OPCClient.dll' and has to be executed once only, e.g. in OB100. All other FBs of the WinAC OPC-Client do not work correctly until the execution of **OPC_INIT**.

Interface

Table 4-1 Parameters of FBs OPC_INIT

Parameter	In/Out	Type	Description
DB_OPC_CONN	In	Block_DB	Instance-DB of OPC_CONN
DB_OPC_ADD_ITEM	In	Block_DB	Instance-DB of OPC_ADD_ITEM
DB_OPC_READ	In	Block_DB	Instance-DB of OPC_READ
DB_OPC_WRITE	In	Block_DB	Instance-DB of OPC_WRITE
ERROR	Out	Bool	Error occurred
STATUS	Out	Word	State of call (to be evaluated if error is set)

All instance DBs of the remaining OPC-Client FBs need to be listed here for initialization.

Return information

In case of an error (e.g. ODK DLL does not exist on system) a negative value of state is returned and the error flag is set.

For more detailed information please refer to chapter 6 "Error messages".

4.2 Establish connection FB65003 – OPC_CONN

This FB establishes a connection between WinAC RTX and any OPC Server.

Interface

Table 4-2 Parameters of OPC_CONN

Parameter	In/Out	Type	Description
REQ	In	BOOL	Request to connect
OPC_SERVER	In	STRING	OPC-Server name (e.g. „SIMATICNET.OPC“)
DONE	Out	BOOL	Done Flag
BUSY	Out	BOOL	Busy Flag
ERROR	Out	BOOL	Error Flag

The connection will only be established until the “REG” flag is set. During connection establishment the “BUSY” flag is set. After the function has been completed, the “DONE” flag is set. Additionally the “ERROR” flag is set in case of an error.

Return information

In case of an error (“ERROR” flag is set) the error codes are saved in the instance DB of the particular FB (e.g. DB06503).

Status	→	ODK-errors, refer to chapter 6.1
ErrorCode	→	User- or system error, see chapter 6.2
OPC_Error_Code	→	OPC error codes are defined through the OPC Foundation. To find out more details on error codes go to: http://www.advosol.us/OpcErrorLookup.aspx

4.3 Register OPC Items FB65004 – OPC_ADD_ITEMS

Use this FB to register items with the OPC Server. The individual items are described in the corresponding UDT “OPC_ITEM_DATA”

Data type UDT4 – OPC_ITEM_DATA

OPC_ITEM_DATA is structured according to table 4-3.

Table 4-3 UDT4 OPC_ITEM_DATA structure:

Parameter	In/Out	Type	Comment
OPC_ITEM_NAME	In	STRING[254]	OPC-Item name
dataTypee	In	BYTE	For more information on data types see table 4-4
quantity	In	INT	Number of items (Array>1)
dbNumber	In	INT	DB number (if memoryArea type is a DB)
memoryArea	In	BYTE	Type of memory: DB, M , E, A,... see table 4-5
areaOffset	In	INT	Memory byte offset
bitNumber	In	BYTE	Memory bit offset

Parameter	In/Out	Type	Comment

Read_Write_Mask	In	BYTE	Direction of communication
			read = 0;
			write = 1;
			read and write = 2

Status	Out	DWORD	State of item<>0 → error in item.

"OPC_ITEM_NAME" describes the name of the particular OPC Item. The remaining parameters describe a specific memory of the WinAC RTX PLC, the state of item or the direction of communication.

Syntax of "OPC_ITEM_NAME" depends on the OPC Server. Type and length of the item described with "OPC_ITEM_NAME" have to match the values of dataTypee, quantity etc.

Parameter **'dataTypee'** supports the following list of data types:

Table 4-4 possible data types of dataTypee:

Hex	Data type
0x01	Bit
0x02	Byte
0x03	Char
0x04	Word
0x05	Integer
0x06	Double Word
0x07	Double Integer
0x08	Real
0x09	Date
0x0A	Tod
0x0B	Time
0x13	S7-String

Parameter **'memoryArea'** supports the following types of memory:

Table 4-5 possible memory types of memoryArea:

Hex	Memory Types
0x80	(P) Peripheral
0x81	(E) Process image
0x82	(A) Process image
0x83	(M) Flag
0x84	(DB) Data Block
0x01	(DS) Data Set
0x03	(SD) AS200:SYS-AREA
0x04	(S) AS200:stage bits
0x05	(SF) AS200:special flag

Hex	Memory Types
0x06	(AI) AS200:analog input
0x07	(AQ) AS200:analog output
0x08	(SZL) System State List

Parameter “Read_Write_Mask” defines whether an item can be read and written or both:

0 → Item values are read from OPC-Server and written to specified WinAC memory area.

1 → Item values are read from specified WinAC memory area and written to OPC-Server.

2 → Item values are both read and written

In case of failed registration of an item with the OPC-Server its parameter **Status** contains the corresponding error code.

When reading and writing, items are processed only when their status flag is zero, i.e. when they are registered successfully with the OPC Server.

Interface

Table 4-6 Parameters of FBs OPC_ADD_ITEMS

Parameter	In/Out	Type	Description
REQ	In	BOOL	Request for OPC_ADD_ITEM
OPC_ITEMS	In/Out	ANY	ANY-Pointer to array of UDT4 structure (OPC_ITEM_DATA)
DONE	Out	BOOL	Done Flag
BUSY	Out	BOOL	Busy Flag
ERROR	Out	BOOL	Error Flag

Items are not registered with the OPC Server until “REQ” flag is set. Length of ANY-Pointer (OPC_ITEMS) is checked. If the length is not a multiple of the UDT4 structure (OPC_ITEM_DATA) the function is aborted with error.

On registration all incorrect items are “marked” with corresponding error codes in the Status parameter. Registration of the other items, however, is still continued. At the end of the registration, the “BUSY” flag is reset, “DONE” flag and if applicable “ERROR” flag is set.

Return information

In case of an error, i.e. when the “ERROR” flag is set, the error codes are stored in the instance-DB of FB (e.g. DB6504):

Status	→	ODK-errors, refer to chapter 6.1
ErrorCode	→	User- or system errors, see chapter 6.2
OPC_Error_Code	→	OPC error codes are defined through the OPC Foundation. To find out more details on error codes go to: http://www.advosol.us/OpcErrorLookup.aspx

4.4 Read OPC-Items FB65006 OPC_READ

Use this FB to read all items which are declared as Read-Items or Read/Write-Items from the OPC Server and write them into the specified memory area of WinAC.

Interface

Table 4-7 Parameters of FBs OPC_READ

Parameter	In/Out	Type	Description
REQ	In	BOOL	Request of OPC_READ
DONE	Out	BOOL	Done Flag
BUSY	Out	BOOL	Busy Flag
ERROR	Out	BOOL	Error Flag

All “Read” OPC Items are read when “REQ” flag is set. Afterwards “REQ” flag is reset, the “DONE” flag and, if applicable, the “ERROR” flag is also set.

Return information

In case of an error, i.e. when the “ERROR” flag is set, error codes are stored in the instance-DB of FB (e.g. DB6506):

Status	→	ODK-errors, refer to chapter 6.1
ErrorCode	→	User- or system errors, see chapter 6.2
OPC_Error_Code	→	OPC error codes are defined through the OPC Foundation. To find out more details on error codes go to: http://www.advosol.us/OpcErrorLookup.aspx

4.5 Write OPC-Items FB65007 OPC_WRITE

Use this FB to read all items which are declared as Write-Items or Read/Write-Items from WinAC and write them to the OPC-Server

Interface

Table 4-8 Parameters of FBs OPC_WRITE

Parameter	In/Out	Type	Description
REQ	In	BOOL	Request for OPC_WRITE
DONE	Out	BOOL	Done Flag
BUSY	Out	BOOL	Busy Flag
ERROR	Out	BOOL	Error Flag

All “Write” OPC Items are written when “REQ” flag is set. Afterwards “REQ” flag is reset, “DONE” flag and, if applicable, “ERROR” flag is set

Return information

In case of an error, i.e. when the “ERROR” flag is set, error codes are stored in the instance-DB of FB (e.g. DB6507):

Status	→	ODK-errors, refer to chapter 6.1
ErrorCode	→	User- or system error, see chapter 6.2
OPC_Error_Code	→	OPC error codes are defined through the OPC Foundation. To find out more details on error codes go to: http://www.advosol.us/OpcErrorLookup.aspx

5 Program example

The supplied Step7 example program demonstrates correct usage of the WinAC OPC-Client FBs

Step1

OPC-Client Paramater DB defines the OPC Items and the OPC Server:

Table 5-1 OPC_ClientDB. OPC_SERVER_NAME:

DB Variable	Type	Value
OPC_SERVER_NAME	STRING [254]	'OPC.SIMATICNET'

→ **“OPC.SIMATICNET” identifies current OPC Server**

Table 5-2 OPC_ClientDB.OPC_ITEM1:

DB Variable	Type	Value
OPC_ITEM_NAME	STRING]	'S7:[Tst]DB10,X0.0,16'
dataTypee	BYTE	B#16#1 → BOOL
quantity	INT	16
dbNumber	INT	0
memoryArea	BYTE	B#16#83 → flag area
areaOffset	INT	2000
bitNumber	BYTE	B#16#0
Read_Write_Mask	BYTE	B#16#1 → WRITE
Status	DWORD	DW#16#0

→Write **16 bits** from **MX2000.0,16** to OPC item **'S7:[Tst]DB10,X0.0,16'**

Table 5-3 OPC_ClientDB.OPC_ITEM2:

DB Variable	Type	Value
OPC_ITEM_NAME	STRING	'S7:[Tst]DB10,CHAR2,2'
dataTypee	BYTE	B#16#3 → Char
quantity	INT	2
dbNumber	INT	0
memoryArea	BYTE	B#16#83 → flag area
areaOffset	INT	2002
bitNumber	BYTE	B#16#0
Read_Write_Mask	BYTE	B#16#1 → WRITE
Status	DWORD	DW#16#0

→Write **2 CHAR** from **MB2002** to OPC Item **'S7:[Tst] DB10,CHAR2,2'**

Table 5-4 OPC_ClientDB.OPC_ITEM3:

DB Variable	Type	Value
OPC_ITEM_NAME	STRING	'S7:[Tst]MX2022.0,7'
dataTypee	BYTE	B#16#1 → BOOL
quantity	INT	7
dbNumber	INT	10
memoryArea	BYTE	B#16#84 → Data Block
areaOffset	INT	22
bitNumber	BYTE	B#16#0
Read_Write_Mask	BYTE	B#16#0 → READ
Status	DWORD	DW#16#0

→Write **7 Bits** from OPC Item **'S7:[Tst]MX2022.0,7'** to **DB10.DBX22.0**

Table 5-5 OPC_ClientDB.OPC_ITEM9:

DB Variable	Type	Value
OPC_ITEM_NAME	STRING	'S7:[Tst]MREAL2028,3'
dataTypee	BYTE	B#16#8 →REAL
quantity	INT	3
dbNumber	INT	10
memoryArea	BYTE	B#16#84 → Data Block
areaOffset	INT	28
bitNumber	BYTE	B#16#0
Read_Write_Mask	BYTE	B#16#0 → READ
Status	DWORD	DW#16#0

→Write **3 REAL** from OPC Item **'S7:[Tst]MREAL2028,3'** to **DB10.DBD28**

etc. (Refer to DB2000 in program example)

Step2

OPC Client is initialized in OB100.

Table 5-6 OB100 Initialization

```
CALL  "OPC_INIT" , "DB_INIT_OPC"                    //Init OPC Client
     DB_OPC_CONN    :="DB_OPC_CONN"
     DB_OPC_ADD_ITEM:="DB_OPC_ADD_ITEMS"
     DB_OPC_READ    :="DB_OPC_READ"
     DB_OPC_WRITE   :="DB_OPC_WRITE"
     ERROR          :="Init_Error"
     STATUS         :=

     L     0                                        //Reset Status Flags
     T     "Flags"

     UN    "Init_Error"                             // Init error ?
     SPB   Ende                                     // N -->

     CALL  "STP"                                    // Stop PLC


Ende: BE
```

In the example shown above “OPC_INIT” is executed with all instance DBs of the remaining WinAC OPC-Client FBs as parameters.

Furthermore all status flags (Connected, Added,...) are reset. The CPU stops when an error occurs.

Step3:

In OB1 the connection to the OPC Server is established, and all items from parameter DB are registered and read/written in a cycle:

Table 5-7 Connect OB1 to OPC Server, register and read/write items:

```
      U     "Connected"                                          //If connected jump to mrk0
      SPB   mrk0

      CALL  "OPC_CONN" , "DB_OPC_CONN"                           //Connect OPC Server
       REQ       :="Connect_OPC"
       OPC_SERVER:="OPCClient_DB".OPC_SERVER_NAME
       DONE      :="Connected"
       BUSY      :="Connect_Busy"
       ERROR     :="Conn_Error"

mrk0: U     "Add_Item_Done"                                     //If items added jump to mrk1
      SPB   mrk1

      CALL  "OPC_ADD_ITEM" , "DB_OPC_ADD_ITEMS"                  //Else add items
       REQ      :="Connected"
       DONE     :="Add_Item_Done"
       BUSY     :="Add_Item_Busy"
       ERROR    :="Add_Item_Error"
       OPC_ITEMS:=P#DB2000.DBX256.0 BYTE 2720

mrk1: CALL  "OPC_READ" , "DB_OPC_READ"                           //Read all "READ" items
      REQ  :="Add_Item_Done"
      DONE :="Read_Done"
      BUSY :="Read_Busy"
      ERROR:="Read_Error"

      UN    "Read_Done"                                          // Read counter
      SPB   mrk2
      L     "Read_Counter"
      L     1
      +I
      T     "Read_Counter"

mrk2: CALL  "OPC_WRITE" , "DB_OPC_WRITE"                         // Write all "WRITE" Items
      REQ  :="Add_Item_Done"
      DONE :="Write_Done"
      BUSY :="Write_Busy"
      ERROR:="Write_Error"

      UN    "Write_Done"                                         //Write counter
      SPB   ende
      L     "Write_Counter"
      L     1
      +I
      T     "Write_Counter"

ende: BE
```

Step4

All functions of the OPC Client can be monitored and controlled through Variable Table VAT1.

6 Error Messages

WinAC OPC-Client supplies three different types of error messages. All error messages are stored in the instance DB.

- Error Code within instance DB **ODK-Status** according to WinAC-ODK (refer to chapter 6.1 in this document)
- Error Code within instance DB **ODK_OUT. ErrorCode** as user-/system error (refer to chapter 6.2 in this document)
- Error Code within instance DB **ODK_OUT. OPC_ErrorCode** as OPC error. OPC error codes are defined through the OPC- Foundation. To find out more details on OPC error codes go to: http://www.advosol.us/OpcErrorLookup.aspx

All errors are signaled through the **error flag**.

6.1 Error Codes of WinAC ODK 4.1

WinAC OPC-Client was developed with WinAC ODK (Open Development Kit). ODK itself also generates error codes, which are returned in the status flag of the FB. These are not described in the documentation of the FBs.

6.1.1 Error Codes of SFB65001 CREA_COM

The following error codes can be returned only by **FB TINIT**.

Table 6-1 WinAC ODK error codes of CREA_COM

Error Code	Symbol	Description
0	NO_ERRORS	Success
0x807F	ERROR_INTERNAL	An internal error occurred.
0x8001	E_EXCEPTION	An exception occurred.
0x8102	E_CLSID_FAILED	The call to CLSIDFromProgID failed.
0x8103	E_COINITIALIZE_FAILED	The call to CoInitializeEx failed.
0x8104	E_CREATE_INSTANCE_FAILED	The call to CoCreateInstance failed.
0x8105	E_LOAD_LIBRARY_FAILED	The library failed to load.
0x8106	E_NT_RESPONSE_TIMEOUT	A Windows response timeout occurred.
0x8107	E_INVALID_OB_STATE	Controller is in an invalid state for scheduling an OB.
0x8108	E_INVALID_OB_SCHEDULE	Schedule information for OB is invalid.
0x8109	E_INVALID_INSTANCEID	Instance ID for SFB65001 call is invalid.
0x810A	E_START_ODKPROXY_FAILED	Controller could not load proxy DLL.
0x810B	E_CREATE_SHAREMEM_FAILED	The WinAC controller could not create or initialize shared memory area.
0x810C	E_OPTION_NOT_AVAILABLE	Attempt to access unavailable option ocurred.

6.1.2 Error Codes of SFB65002 EXEC_COM

The following error codes can be returned by all FBs.

Table 6-2 WinAC ODK error codes of EXEC_COM

Error Code	Symbol	Description
0	NO_ERRORS	Success
0x807F	ERROR_INTERNAL	An internal error occurred.
0x8001	E_EXCEPTION	An exception occurred.
0x8002	E_NO_VALID_INPUT	Input: the ANY pointer is invalid.
0x8003	E_INPUT_RANGE_INVALID	Input: the ANY pointer range is invalid.
0x8004	E_NO_VALID_OUTPUT	Output: the ANY pointer is invalid.
0x8005	E_OUTPUT_RANGE_INVALID	Output: the ANY pointer range is invalid.
0x8006	E_OUTPUT_OVERFLOW	More bytes were written into the output buffer by the extension object than were allocated.
0x8007	E_NOT_INITIALIZED	ODK system has not been initialized: no previous call to SFB65001 (CREA_COM).
0x8008	E_HANDLE_OUT_OF_RANGE	The supplied handle value does not correspond to a valid extension object.
0x8009	E_INPUT_OVERFLOW	More bytes were written into the input buffer by the extension object than were allocated.

6.2 User-/System Errors

The following error codes can be returned by the driver of OPC Client.

Table 6-3

Error Code	Description
0x80000001	Establishing of connection to OPC server failed
0x80000002	Error on registering OPC items
0x80000008	Error on reading OPC items
0x80000010	Error on writing OPC items
0x80000100	Error on ODK input/output
0x80000200	Error during call of asynchronous function (ODK internal)
0x80001000	Wrong length of ANY at OPC_ADD_ITEM function or maximum quantity of items reached (4000 items max)
0x80002000	Connection to server has not been established yet.
0x80004000	Length of selected memory area does not match length of ‘READ’ OPC item
0x80008000	Length of ‘WRITE’ OPC does not match length of selected memory area
0x80020000	There is no registered item to be written/read

7 History

Table 7-1

Version	Date	Comment
V1.0	18.04.08	First version for delivery
V1.1	02.04.09	Tested with WinAC RTX 2008
V1.2	02.11.09	Tested with WinAC RTX-F 2009