SIEMENS

Application description • 11/2013

Fast Process Value Archiving Using the Function AR_SEND

SIMATIC PCS 7 or STEP 7 with WinCC

http://support.automation.siemens.com/WW/view/en/ 23780904

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.

Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of Siemens Industry Sector.

Caution
The functions and solutions described in this article confine themselves to the realization of the automation task predominantly. Please take into account furthermore that corresponding protective measures have to be taken up in the context of Industrial Security when connecting your equipment to other parts of the plant, the enterprise network or the Internet. Further information can be found under the Item-ID 50203404.

http://support.automation.siemens.com/WW/view/en/50203404

Table of contents

1 Task

1.1 Overview

Introduction

There are some processes which require the archiving of data in rapid cycles.

Since the minimum cycle time of a conventional archiving process with WinCC Tag Logging is 500ms, a better solution should be found.

This solution is offered by the S7-400 / WinAC RTX system function block AR_SEND / SFB 37. It realizes the communication between automation system (AS) and operator station (OS) and sends the data to WinCC where it is stored in the archive “TagLogging Fast”.

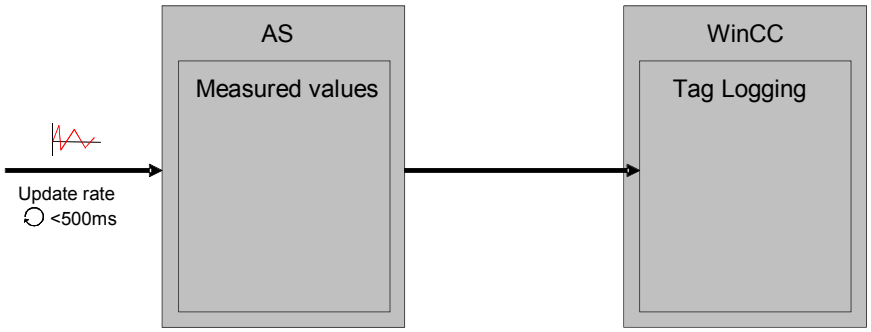
Based upon SFB 37 two function blocks are provided which offer a higher data throughput for the archiving of measured values than can be obtained if data request is initiated by the OS.

In order to keep the load on the system bus as small as possible, the process values are temporarily stored in the AS before they are transmitted blockwise to the OS.

Overview of the automation task

The figure below provides an overview of the automation task.

Figure 1-1



1.2 Requirements

The solution shall be suitable for the following products:

- SIMATIC PCS 7

The solution may also be used in STEP 7 and WinCC, though this approach is not explicitly described in this example.
The solution presented here only requires a configuration and Runtime environment with the corresponding licenses and an automation system supporting the system function block AR_SEND / SFB 37.

2 Solution

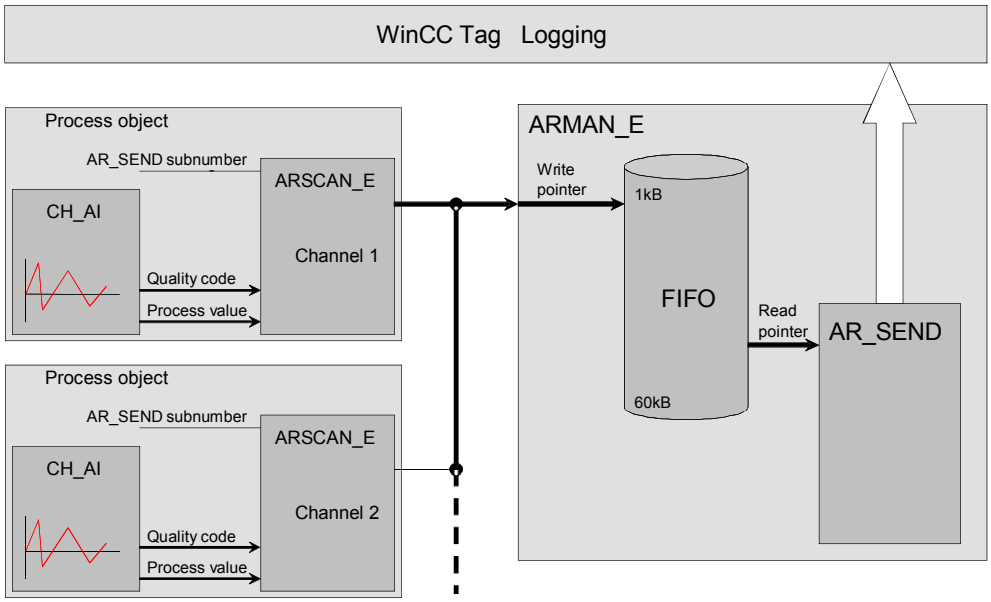
2.1 Overview

Based on the system function block AR_SEND / SFB 37, the function blocks ARSCAN_E and AR_MAN_E are aimed at the most efficient storage of measured values and their blockwise transmission to the OS.

- The block ARSCAN_E is used for the acquisition and storage of an individual process value of data type REAL or DWORD.
  Each process value requires the use of one ARSCAN_E block.
  There are two variants for the acquisition of process values:
  - Cyclic process value acquisition is used for the equidistant archiving of process values (at fixed time intervals).
  - Event-controlled process value acquisition is used, if a process value shall be archived as soon as its value changes. This variant is only suited for process values of data type REAL.
- The block AR_MAN_E includes a FIFO memory for the process values, and it manages the blockwise data transmission to WinCC with the help of the system function block AR_SEND.
  It is also possible to combine several ARSCAN_E blocks with one AR_MAN_E block for writing to the FIFO memory (one ARSCAN_E per process value).

The figure below shows a schematic illustration of the two blocks for the fast archiving of process values.

Figure 2-1



Typical fields of application

- Fast archiving of process values with the data types DWORD and REAL (<500ms)
- Intermediate storage of process values in the AS
- Flexible archiving times for process values (when changed, equidistant, manual)

Advantages

- Easy interconnection of process tags in the CFC Editor
- Easy configuration of the system function block AR_SEND
- Provision of all required header information for the AR_SEND block
- Exact time stamp for the process values from the AS
- Variable trigger options for data transmission to WinCC
- Automatic repetition of telegrams in case of an error

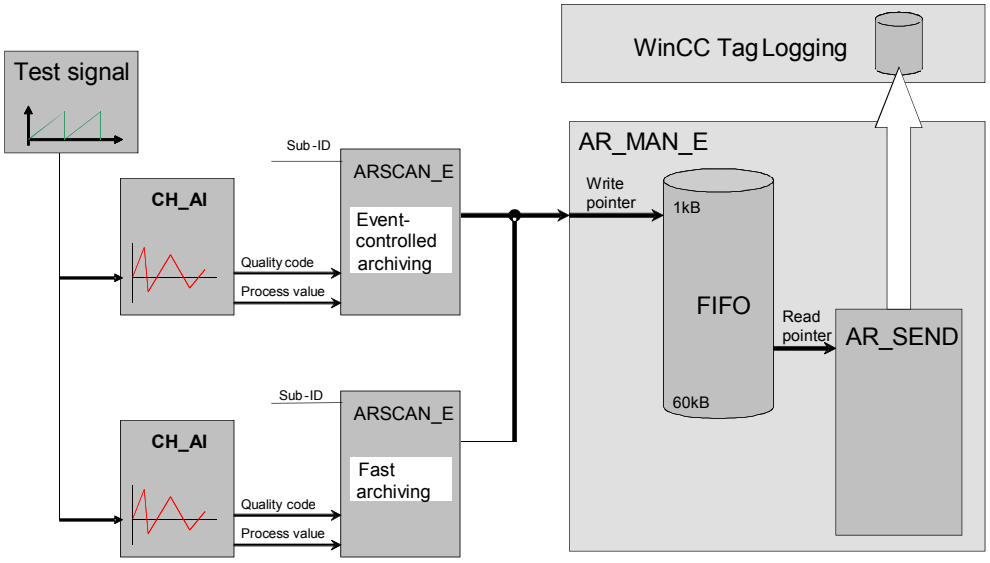
Assumed knowledge

It is assumed that the user has basic knowledge in the field of automation systems and SIMATIC PCS 7.
This application example was created with SIMATIC PCS 7 V8.0 Update 1 (SCL V5.3 + SP6).

2.2 Description of the core functionality

The figure below shows a schematic illustration of the function principle in this example.

Figure 2-2



The test signal is picked up by two ARSCAN_E blocks.
The first block stores the process value event-controlled, the second one at cyclic intervals.
The block AR_MAN_E includes the FIFO memory for the process values and via AR_SEND it sends this data blockwise to WinCC where it is entered in the “TagLogging Fast” archive.

2.3 Hardware and software components

2.3.1 Components

This application example was created using the following components:

Hardware components

Table 2-1

Component	Qty.	Order number	Note
Module rack	1	6ES7400-1JA01-0AA0	Or comparable
Power supply module PS407 10A	1	6ES7407-0KA01-0AA0	Or comparable
SIMATIC S7-416-3 PN/DP	1	6ES7416-3ER05-0AB0	Or comparable S7-400 / S7-400 H WinAC RTX

Software components

Table 2-2

Component	Qty.	Order number	Note
SIMATIC PCS 7 V8.0 Update 1	1	6ES7651-5AA08-0YA0	ES single station V8.0
S7-PLCSIM	1	6ES7841-0CC05-0YA5	If the AS of the sample project shall be simulated with S7-PLCSIM

Sample files and projects

The following list shows all files and projects used in this application example.

Table 2-3

Component	Note
23780904 _FastArch_mp.zip	This file includes the PCS 7 sample project and can be retrieved directly via the SIMATIC Manager.
23780904 _Code.zip	This file includes the SCL sources for the following blocks: • AR_MAN_E • ARSCAN_E

3 Function Principle

3.1 Transmission of archive data with AR_SEND

Limitation of AR_SEND instances

The maximum number of AR_SEND instances per CPU that can be simultaneously registered for transmission is limited and depends on the specific type of CPU.

Table 3-1

CPU Type	Number of AR_SEND Instances
CPU 410	64
CPU 417	64
CPU 416	32
CPU 414	16
CPU 412	4
From WinAC RTX 2010	32

Note Further up-to-date information on the various CPUs and the AR_SEND instances is available in the device manual
“SIMATIC S7-400 Automation System S7-400 CPU Data”.

Function principle of the AR_SEND block

With the AR_SEND block, the archive tags can be supplied with data in different ways by using either

- one archive tag (AR_ID is sufficient) or
- several archive tags (AR_ID and AR_ID-Subnumber)

The AR_ID and the AR_ID-Subnumber are used to manage the allocation between the data in the AS and the archive tags.
Theoretically, up to 4095 AR_ID-Subnumbers are possible for each AR_ID. In practice, however, the number of archive tags per AR_SEND instance is limited by the maximum length of the data range to be transmitted.

Furthermore, the AR_SEND block offers different formats for the transmission of process values to WinCC Tag Logging, e.g. several process values with the same time stamp or with a time different to that of the last value.

Note For detailed information on the AR_SEND block, please refer to the online help in WinCC under
"WinCC Information System > Communication > SIMATIC S7 Protocol Suite > Special Functions > Data exchange with the S7 function block AR_SEND".

AR_SEND performance

When using the S7 functions AR_SEND and BSEND/BRCV for communication with S7-400, the relevant resource restrictions are to be observed.
This means that the maximum data volume that can be simultaneously sent from

the AS to WinCC via the AR_SEND and/or BSEND/BRCV functions is limited to max. 16 Kbytes.

NOTICE

The limitations regarding the data volumes for the WinCC communication channel must be strictly observed. Please also note that the maximum data volume will be halved, if redundant systems are used.

For detailed information on resource restrictions, please read, in any case, the section “Number of process values” in the WinCC online help under "WinCC Information System > Communication > SIMATIC S7 Protocol Suite > Special Functions > Data exchange with the S7 function block AR_SEND > Data block – Structure and Parameters".

3.2 General information on the application example

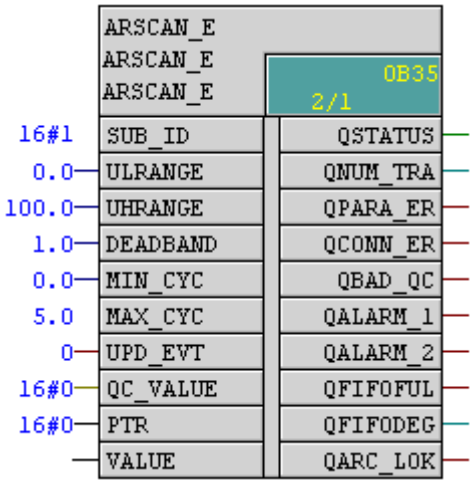
The solution offered here includes two function blocks:

- ARSCAN_E
  - for the acquisition and storage of process values
- AR_MAN_E
  - for the management of the process value memory and data transmission to WinCC via AR_SEND

These two blocks are described in detail in the following chapters.

3.3 The block ARSCAN_E

Figure 3-1



3.3.1 Block connections

Input parameters

Table 3-2

Parameter	Data type	Description
EN (hidden)	BOOL	Activates block processing

Parameter	Data type	Description
		Default value: 1
ENABLE (hidden)	BOOL	Activates the acquisition of process values. *Default value: 1*
RUNUPCYC (hidden)	INT	Number of run-up cycles after CPU startup. *Default value: 5*
SAMPLE_TIME (hidden)	REAL	Cycle time of the cyclic interrupt OB in which the block is called. Defined by the compiler.
SUB_ID	WORD	AR_ID-Subnumber used to assign the process value to an archive tag in a process value archive. *Permissible values: 1- 4095*
ULRANGE	REAL	Lower limit of measuring range (relevant only for process values of data type REAL). *Default value: 0.0*
UHRANGE	REAL	Upper limit of measuring range (relevant only for process values of data type REAL). *Default value: 100.0*
DEADBAND	REAL	Minimum change of process values compared to the previous value in percent, initiating storage of the previous value (relevant only for process values of data type REAL). *Default value: 0.5* **Note:** This percentage refers only to the parameters ULRANGE and UHRANGE
MIN_CYC	REAL	Minimum acquisition time (in seconds). *Default value: 0.0*
MAX_CYC	REAL	Maximum acquisition time (in seconds) of the process value (archives the value even if the change < DEADBAND). *Default value: 5.0*
UPD_EVT	BOOL	Edge-triggered initiation of an archiving process.
UPD_COND (hidden)	BOOL	Status-triggered initiation of an archiving process.
QC_VALUE	BYTE	Quality code of the process value (to be interconnected or configured with **16#80**).
PTR	STRUCT	Interconnected with the block AR_MAN_E. Supplies information for ARSCAN_E and a pointer to the FIFO memory.

In/out parameters

Table 3-3

Parameter	Data type	Description
VALUE	ANY	Process value to be archived. Only values of data types REAL and DWORD are permitted.

Output parameters

Table 3-4

Parameter	Data type	Description
ENO (hidden)	BOOL	Block processing is activated.
QSTATUS	WORD	Status word of the block.
QNUM_TRANS	INT	Number of archived process values.
QUPD_EVT (hidden)	BOOL	Set for a cycle when the block writes to the FIFO memory of the block AR_MAN_E.
QPARA_ER	BOOL	Block parameter error.
QCONN_ER	BOOL	Interconnection error of the input parameter VALUE.
QBAD_QC	BOOL	Bad quality code of the process value. The process value will not be archived.
QALARM_1	BOOL	Alarm stage 1 reached
QALARM_2	BOOL	Alarm stage 2 reached
QFIFOFUL	BOOL	FIFO memory of the block AR_MAN_E is full. No further process values can be archived.
QFIFODEG	INT	Filling level of the FIFO memory in %
QARC_LOK	BOOL	Archiving at block AR_MAN_E is deactivated.

Allocation of the status word QSTATUS

Table 3-5

Bit	Byte	Description
0	Byte 0	QC = 16#00
1		PTR invalid or not connected
2		VALUE not connected
3		VALUE has wrong data type
4		Alarm stage 1 reached
5		Alarm stage 2 reached
6		FIFO memory usage too high (FIFO full)
7		Archiving blocked because AR_MAN_E is active
8	Byte 1	AR_ID-Subnumber < 1
9		Invalid parameter setting for MIN_CYC & MAX_CYC
10		Incorrect parameter setting for DEADBAND (<0% or >100%)
11		ULRANGE >= UHLRANGE
12		Not assigned
13		Not assigned
14		Not assigned

Bit	Byte	Description
15		Not assigned

3.3.2 Function principle

Behavior during overload or if connection failed

If the FIFO memory of the AR_MAN_E block becomes too full, the data volume to be stored will automatically be reduced (alarm stage 1, alarm stage 2). This function is described in detail in chapter 3.4.2.

Process value acquisition

The block ARSCAN_E acquires a process value of data type REAL or DWORD including the associated quality code. For this reason, the block is connected to the corresponding channel block (e.g. Pcs7AnIn) in the CFC-Editor or to another data source.
The ARSCAN_E block is called in the acquisition cycle of the process value (e.g. after the channel block).

Each ARSCAN_E block is interconnected with an AR_MAN_E block, whereby one AR_MAN_E block may be connected to several ARSCAN_E blocks.

For process values of data type REAL, the limits of the process value must be defined at the input parameters ULRANGE and UHRANGE.
The acquired process value will then be checked for validity (quality code “QC_VALUE”) and changes.

A value change for a data type REAL is valid, if comparison with the previously acquired process values shows that the value has changed by a specific percentage. The user can define the percentage of this change value in relation to the measuring range (DEADBAND).

For values of data type DWORD, a binary change monitoring principle is used, i.e. the input parameter DEADBAND will be ignored.

Depending on how the ARSCAN_E block parameters are defined, there are two different criteria for the storage of a process value.

In the following, both types of storage will be described in detail:

- Cyclic process value acquisition
- Event-controlled process value acquisition

Cyclic process value acquisition

Cyclic process value acquisition can be used for both data types REAL and DWROD. The following settings are to be made:

- Input parameters:
  - DEADBAND = 0
  - MIN_CYC to be set accordingly
  - MAX_CYC > MIN_CYC

In this way, the process value will be stored equidistantly at the set cycle time (MIN_CYC).
Please note that the minimum cycle time is limited by the cycle in which the ARSCAN_E block is called.

Note The equidistant storage of process values may lead to very large data volumes (in particular if the cycle time is very small).

Note Alarm stage 1 is deactivated (DEADBAND doubled).

Event-controlled process value acquisition

Event-controlled process value acquisition is suitable only for data type REAL. This requires the following settings:

- Input parameters:
  - ULRANGE / UHRANGE to be defined (measuring range)
  - DEADBAND to be defined (change in percent for archiving)

In this way, the process value will be stored, if the percentage of the value change in comparison with the previous cycle is equal or larger than the value of the input parameter DEADBAND.

Note Compared to cyclic process value acquisition, the event-controlled storage of process values produces less data volume.

Note Alarm stage 2 is deactivated (archiving cycle increased to MAX_CYC).

Storage of process values

The block AR_MAN_E provides the FIFO memory for the process values.

The block ARSCAN_E stores the acquired process values in this memory. To this effect, the two blocks are interconnected so that all required information regarding the memory area is available.

If the conditions for the storage of a process value are fulfilled, the ARSCAN_E block writes this process value, including the current time stamp, to the FIFO memory and then increments the write pointer of the FIFO memory, so as to indicate the next free storage space.

The block AR_MAN_E is described in detail in the next chapter.

Note After a process value has been stored, the output QNUM_TRANS will be incremented by 1.

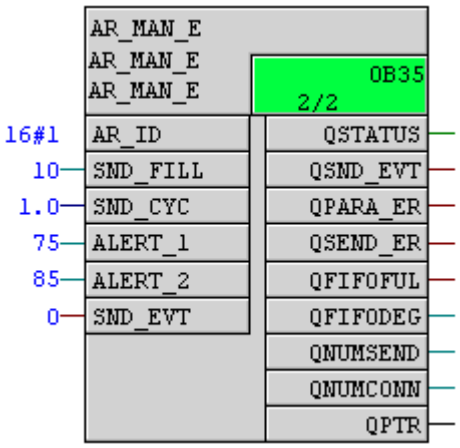
The filling level of the FIFO memory can be seen at the output QFIFODEG.

NOTICE **Since the block ARSCAN_E is interconnected with the block AR_MAN_E with access to its information, it is important that the block ARSCAN_E is always called after the block AR_MAN_E in the run sequence (so as to ensure that this information is always up-to-date).**

This constellation, however, implies that the current process values in the AR_MAN_E block will be processed only in the next cycle.

3.4 The block AR_MAN_E

Figure 3-2



3.4.1 Block connections

Input parameters

Table 3-6

Parameter	Data type	Description
EN (hidden)	BOOL	Activates block processing. *Default value: 1*
ENABLE (hidden)	BOOL	Activates the acquisition of process values. *Default value: 1*
RUNUPCYC (hidden)	INT	Number of run-up cylces after CPU startup. *Default value: 5*
SAMPLE_TIME (hidden)	REAL	Cycle time of the cyclic interrupt OB in which the block is called. Defined by the compiler.
AR_ID	DWORD	Archive number used to allocate an AR_SEND instance to an archive tag. Defined by the compiler.
SND_FILL	INT	Filling level of the FIFO memory in % at which a send process will be triggered. *Default value: 10*
SND_CYC	REAL	Send cycle After this period, a send process with the available data will be triggered. *Default value: 1.0*

Parameter	Data type	Description
ALERT_1	INT	Filling level of the FIFO memory in % at which alarm stage 1 will be activated. *Default value: 75*
ALERT_2	INT	Filling level of the FIFO memory in % at which alarm stage 2 will be activated. *Default value: 85*
SND_EVT	BOOL	Edge-triggered initiation of a send process.

Output parameters

Table 3-7

Parameter	Data type	Description
ENO (hidden)	BOOL	Block processing is activated.
QSTATUS	WORD	Status word of the block.
QSND_EVT	BOOL	Set for a cycle, if a send process has been triggered.
QPARA_ER	BOOL	Error in block parameter configuration.
QSEND_ER	BOOL	An error occurred during processing of the internal block AR_SEND.
QFIFOFUL	BOOL	FIFO memory is full. The process value cannot be archived.
QFIFODEG	BOOL	Filling level of the FIFO memory in %
QNUMSEND	INT	Number of transmitted data blocks
QNUMCONNECT	INT	Number of connected ARSCAN_E instances
QPTR	STRUCT	Interconnection to ARSCAN_E Supplies information for ARSCAN_E and a pointer to the FIFO memory.

Allocation of the status word QSTATUS

Table 3-8

Bit	Byte	Description
0	**Byte 0**	FIFO is full, archiving is blocked
1		AR_ID < 1
2		ALARM_1 > ALARM_2 or one of both is >= 100 or one of both is <= 0
3		SND_FILL: > ALARM_1 > ALARM_2 >=100 <=0

Bit	Byte	Description
4		SND_CYC < 0
5		Not assigned
6		Not assigned
7		Not assigned
8	Byte 1	Not assigned
…		…
15		Not assigned

3.4.2 Function principle

Realization of the AR_SEND procedure

Owing to the CPU-related limitation of AR_SEND instances, only one instance of the AR_SEND block is used for each AR_MAN_E block instance.

An AR_MAN_E block can manage up to 4095 ARSCAN_E blocks (as 4095 AR_ID-Subnumbers are possible for each AR_ID). In this way, several process values can be acquired and transmitted to WinCC by one AR_MAN_E instance. The actual number of ARSCAN_E blocks, however, will be reduced by the respective data volume and the specific application.

AR_SEND format

For transmission, the AR_SEND format 9 (header type = 9) is used.

This format enables the transmission of values with individual time stamps and different AR_ID-Subnumbers.

A 4-bytes value (REAL /DWORD) typically requires 26 bytes in the telegram.

Table 3-9

22 bytes	Header type = 9			
	Year		Month	
	Day		Hour	
	Minute		Second	
	1/10s	1/100s	1/1000s	Weekday
	Cycle = 0			
	AR_ID-Subnumber			
	Data type of the process data			
	Number of process values			
4 bytes	Process value			

NOTICE **The parameter AR_ID is analyzed only during startup of the program. Consequently, AR_ID cannot be modified later while the program is running.**

Archiving / data transmission to WinCC

A send process is started, when:

- the configured filling level of the FIFO memory has been reached
- the time configured for the send process has elapsed
- a send process has been triggered manually

After initiation of a send process, the process values will be transmitted to WinCC via AR_SEND.

Note

The output QFIFODEG shows the filling level of the FIFO memory.

The output QNUMSEND shows the number of data blocks transmitted.

The output QNUMCONNECT shows the number of connected ARSCAN_E blocks. Counting is performed only when the block starts up, so that this output will be updated only when the controller is started anew.

Behavior during overload or failed connection

If the AR_MAN_E is overloaded or if the connection to WinCC fails, more process values will be written to the memory than can be read out from it for transmission to WinCC. To extend the period until the memory overflows, 2 configurable alarm stages have been implemented.

- Alarm stage 1 (input parameter: ALERT_1 in %)
  - The ARSCAN_E block doubles the input parameter DEADBAND internally. Alarm stage 1 is suitable only for process values of data type REAL.
- Alarm stage 2 (input parameter: ALERT_2 in %)
  - The ARSCAN_E block increases the archiving cycle to MAX_CYC.

Despite these measures, it may occur that the FIFO reaches a filling level of 100%. In this case, no new process values will be stored until the filling level of the memory is reduced again.

Note

Generally should be noted that the quantity structure of the process values should always remain manageable.

4 Configuration

4.1 Preliminary remarks

This chapter describes the configuration steps to be performed for using this solution.

Some basic configurations, such as how to create a project and how to configure stations, hardware and connections are not explicitly described in this document.

The solution can be tested and experimented with the help of a sample project which is available for download. The specific use of this sample project is described in detail in chapters 5 - Sample project and 6 - Operation of the sample project.

All screenshots and descriptions in this chapter are based on the sample project, so as to give a basic outline on how to proceed.

4.2 Important notes before starting configuration

Before you start to configure the 'fast process value archiving' function for your specific application, you should be aware of a couple of points, so as to ensure that the solution will not be stretched to its limits.

- Carefully calculate the quantity structure of the process values to be archived.
- Do not define faster cycles than necessary for archiving, so as to avoid that the data volume will be unnecessarily increased.
- Take note of the maximum number of AR_SEND instances per CPU as described in chapter 3.1 (one AR_MAN_E block instances one AR_SEND block).
- Take note of the limits regarding communication with WinCC as described in chapter 3.1. If you fear to come close to these limits, you should thoroughly think about this issue and carefully calculate the quantity structure of your application.
- Test your quantity structure in advance, e.g. with a project similar to this application example.
- Please note that the use of a CAS (Central Archive Server) / PH (Process Historian) for the archiving of long-term archive tags may reduce the performance of your system and thus constrain the quantity structure for fast process value archiving.
- Please consider that the processing time of the whole S7 program plays a major role.
- All other communications (e.g. Modbus communication, OPC communication on OS level, etc.) should also be taken into consideration.
- Make sure that the clock time synchronization works properly throughout your system; any time differences, e.g. between the AS and the OS, may cause problems.
- Please bear in mind that the communication load of redundant systems may be increased or the limits for communication with WinCC may be reduced.
- We recommend to define a workaround for the handling of stop and start of OS Runtime in connection with fast process value archiving.
  Example: Archiving shall start only, if OS Runtime has been fully started and archiving shall end before OS Runtime is stopped.

NOTICE

Please note that during communication with redundant OS servers the connection to the two servers is not monitored individually. The system block AR_SEND will indicate an error only, if the connection to both servers fails.

If redundant systems are used, we therefore recommend to ensure suitable monitoring of the connections in any case, as well as a suitable workaround.

In summary it can be stated that there are many factors which influence the performance of fast process value archiving. For this reason, the whole environment must always be taken into consideration.

If large-scale quantity structures are present, we also recommend to test their proper function in advance.

NOTICE

If unexpected problems arise in the context of fast process value archiving, the whole system environment should be checked.

Please also take into consideration that negative effects may also result from problems in the network traffic on the plant bus.

4.3 Configuration in SIMATIC Manager

Import blocks

Table 4-1

No.	Action
1.	Import the blocks ARSCAN_E and AR_MAN_E from the sample project into the master data library of your project. Make sure that the AR_SEND block is imported also, since this block is required by the block AR_MAN_E.
2.	If required, adapt the block numbers as necessary for your project.

Note

For the import of blocks, it is recommend to proceed as follows:

1. Use a library as source for the blocks.
2. Create a CFC chart in the project (or library) to which the blocks shall be imported.
3. Open the catalog of the CFC chart, find the library with the blocks you want to import and drag it into the CFC chart.

In this way you make sure that all required blocks will be imported and that the corresponding entries in the symbol table will be generated.

Create blocks from the SCL sources

Proceed as follows to create blocks from the SCL sources:

Table 4-2

No.	Action
3.	Import the SCL sources into your master data library.
4.	Define the symbols for the blocks in the symbol table: FB 672 as ARSCAN_E (or another block number, if required) FB 673 as AR_MAN_E (or another block number, if required)
5.	Import the block AR_SEND into your master data library.
6.	Compile the SCL sources.

Configure AR_MAN_E

Figure 4-1

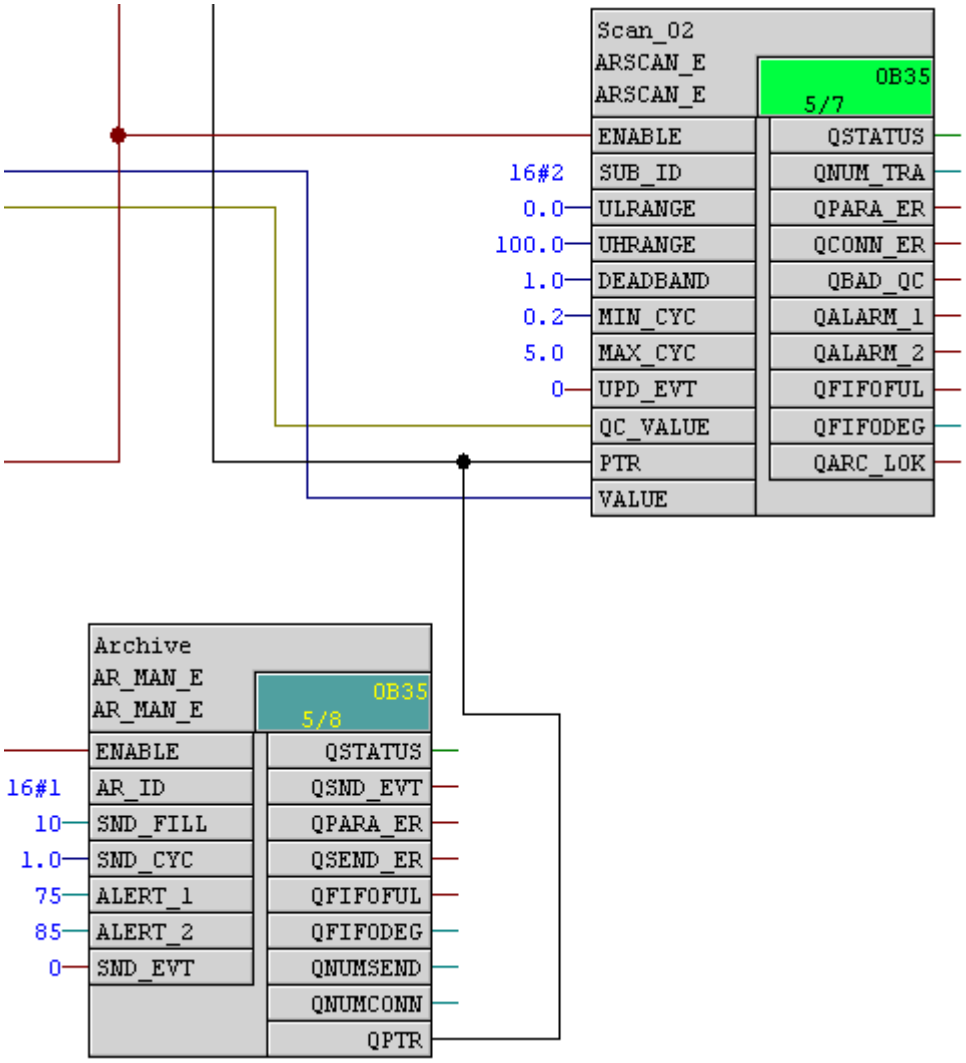


Table 4-3

No.	Action
7.	Place an AR_MAN_E block for each AR_ID you wish to use. You are free to choose whether all process values shall be archived with the same AR_ID or be distributed over several AR_IDs. Conceivably, the process value may be broken down into subsystems, equipment modules or archiving cycles, etc. Please consider the restrictions regarding the AR_ID-Subnumbers (and thus the archive tags) per AR_ID (4095 AR_ID-Subnumbers per AR_ID).
8.	Configure the remaining parameters and interconnections at the block according to your requirements.

Configure ARSCAN_E

Figure 4-2

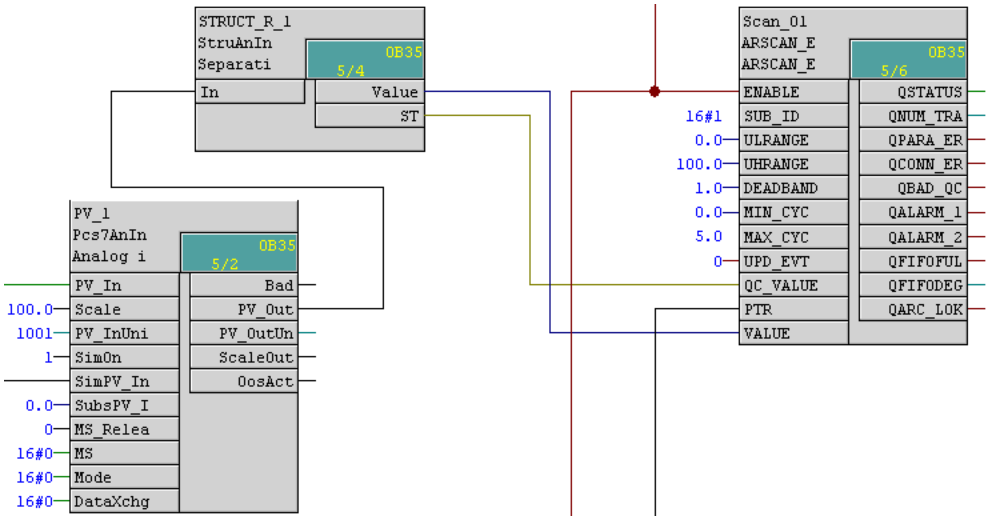


Table 4-4

No.	Action
9.	Place an ARSCAN_E block for each process value (archive value) you wish to archive.
10.	Connect the input parameter VALUE with the source of your process value. Make sure that only the data types DWORD and REAL are used. If your process value uses another data type, ensure correct data type conversion.
11.	Connect the input QC_VALUE with the quality code of the process value, or define the value **16#80** (good quality) for it.
12.	Connect the input PTR with the output QPTR of the AR_MAN_E block which is used to produce a connection to the WinCC process value archive.
13.	Make sure that the run sequence of the blocks is correct. The ARSCAN_E blocks should, by standard, be arranged after the blocks where the process values are picked up and after the AR_MAN_E block to which they are connected.
14.	Configure the remaining parameters and connections at the block according to your requirements.

NOTICE

All AR_ID-Subnumbers must be configured in WinCC.

If WinCC identifies a non-configured AR_ID-Subnumber, the interpretation of user data will be stopped.

Note

In order to check the proper function of your configuration, e.g. to ensure completeness of the archives in case of large quantity structures, we recommend to use a singular signal (e.g. sawtooth function) and to connect it optionally with the ARSCAN_E block (e.g. as a simulation value at the channel block, as indicated in Figure 4-2).

In this way you can archive the test signal for commissioning, export the archive data via WinCC and analyze it in Microsoft Excel, for example.

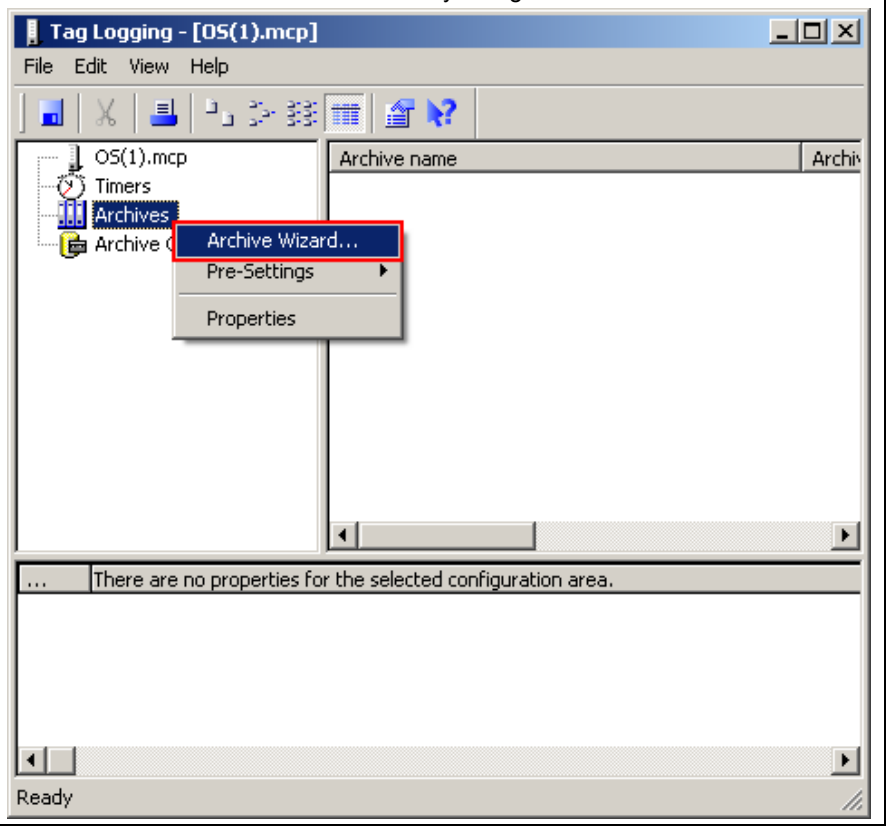
4.4 Configuration in WinCC

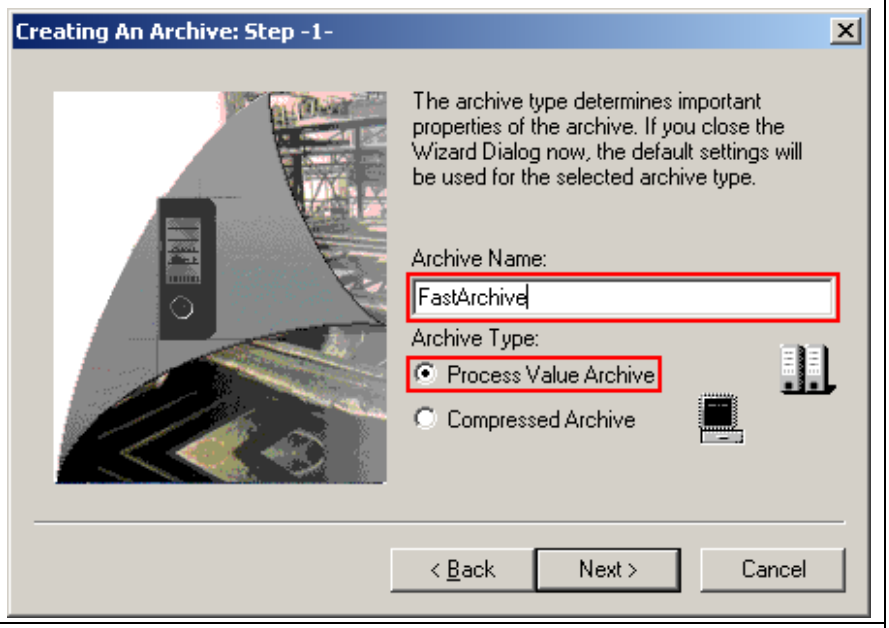
The storage of archive tags requires the creation of at least one process value archive.

You are free to choose whether to create a separate process value archive for each configured AR_ID and each configured AR_MAN_E block. In some cases this may be reasonable and depends on the allocation of process values to be archived, e.g. to enable the separate storage of database segments or to provide a better overview within the project.

Create a process value archive

Table 4-5

No.	Action
15.	Open the WinCC Tag Logging window in the WinCC Explorer.
16.	Select the Archive Wizard for the creation of archives. Click “Archives > Archive Wizard…” with your right mouse button. 

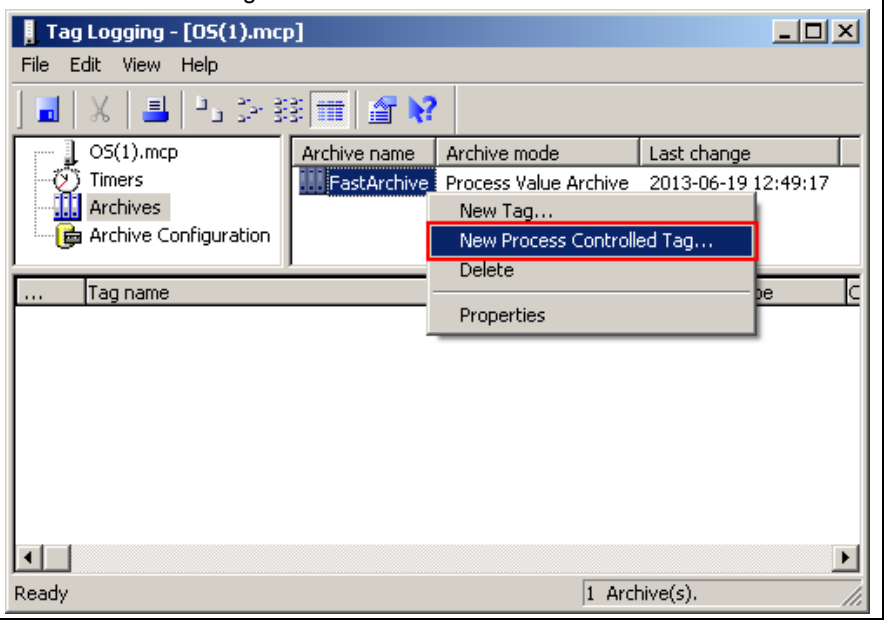
No.	Action
17.	Select the archive type “Process Value Archive”, enter a suitable name and then click the “Next” button. 
18.	In the next dialog, just click the “Apply” button without selecting any tag.

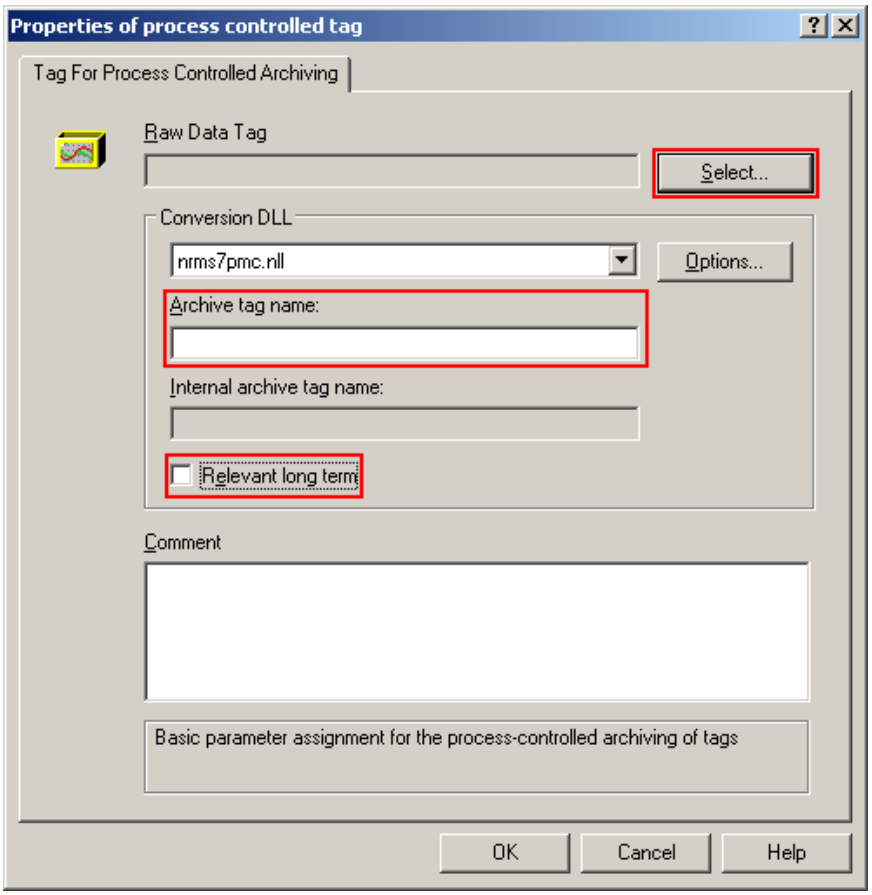
The process value archive has now been created. Repeat the steps described above for each further process value archive you want to create.

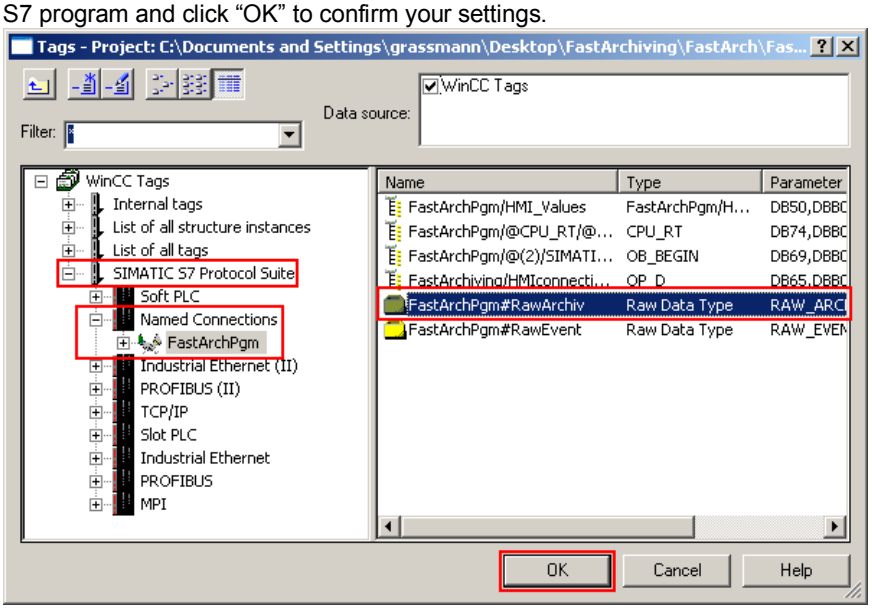
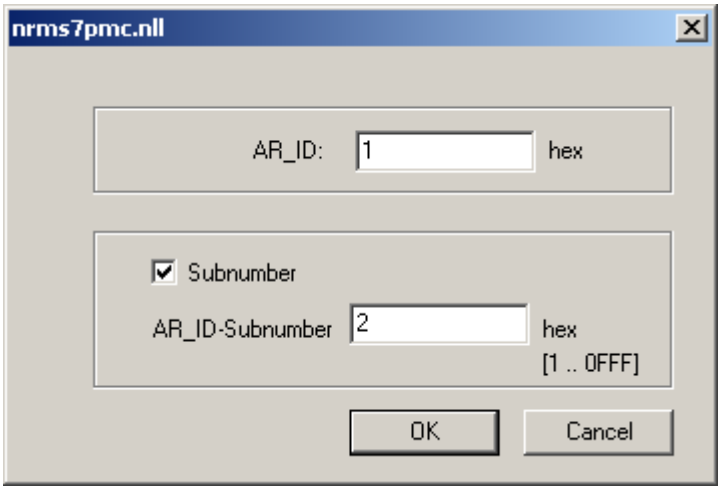
Create new archive tags

Now you have to create an archive tag for each process value to be archived.
If you have decided to use more than one process value archive, please perform these steps in the relevant process archive to which the tag shall be allocated.

Table 4-6

No.	Action
19.	Create a new archive tag. Click the process value archive with your right mouse button and select “New Process Controlled Tag…”. 

No.	Action
20.	• If your archive tags shall be stored for the long-term archiving, enable the “Relevant long term” checkbox. Please note that, for example, the use of a CAS (Central Archive Server) for the archiving of long-term archive tags may affect the system performance and thus reduce the quantity structure for fast process data archiving. • Naming of the archive tag is optional. Sometimes it may be reasonable to not define a name. In this case the system will create a generic name. The advantages of automatic name generation are explained at the end of this table. • Open the dialog window to select the raw data tag for process value archiving. 

No.	Action
21.	Select the raw data type for archives “[Name of program]#RawArchive” in your S7 program and click “OK” to confirm your settings. 
22.	Assign the AR_ID and the AR_ID-Subnumber in compliance with your configurations in the CFC.  If you have decided to use more than one process value archive, make sure to select the corresponding archive when creating an archive tag and to allocate the AR_ID and AR_ID-Subnumber correctly.
23.	Click “OK” to confirm your settings made in the dialog window as shown in point 2.
24.	Repeat the last few steps for the creation of further archive tags.
25.	Save and close the WinCC Tag Logging.

In this example two archive tags were created in the process value archive “FastArchive”. The first one was created without defining a name, for the second one the name “Value02” was chosen.

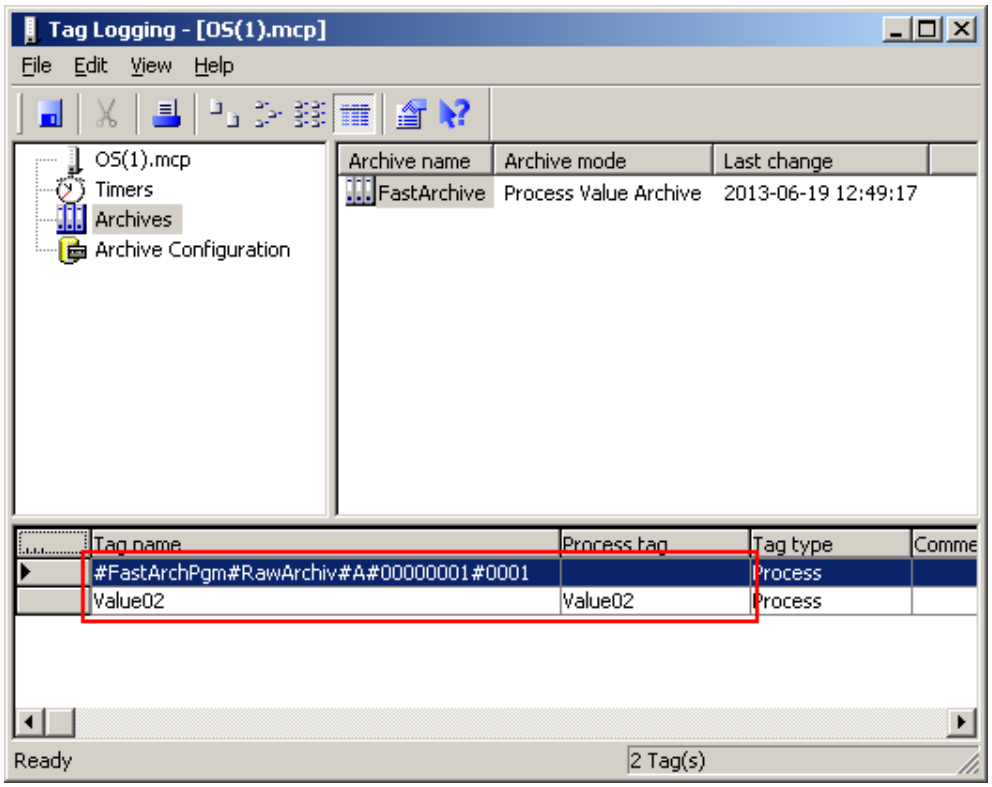
In the following screenshot you can see that a generic name was created automatically for the first tag. This name includes the designations from the S7 program, the raw data type, the AR_ID and the AR_ID-Subnumber.

The use of this generic name is favorable, if a large number of archive tags shall be configured, as you can see at a glance whether AR_ID and AR_ID-Subnumber are correctly allocated.

You may also change the name of archive tags at a later point by clicking the archive tag with your left mouse button.

Please make sure that there is an archive tag for each AR_ID and AR_ID-Subnumber configured in the AS program.

Figure 4-3



Further configuration settings

All steps required for configuration are now completed.

In the context of fast process data archiving you may also specify further configuration settings, as for example:

- Adding trend controls and table controls in WinCC displays for visualization and easy export of the archived process values.
- Configuration of archive backups, if long-term relevance has been selected.
- Configuration for deactivation of the fast process value archiving function before WinCC Runtime is closed, or activation of archiving after the start of WinCC Runtime.
  This configuration helps to increase the plausibility of archived process values, since it provides a clear cut with reference to the start and stop of WinCC Runtime.
  Furthermore, this measure prevents an overflow of the process value buffer.

5 Sample project

The sample project shows the basic configuration required for fast process value archiving.

The project has been realized as a single-station system. Configuration of the automation hardware corresponds to a real environment and has been tested with this system.

The sample project is suitable for use with S7-PLCSSIM, so as to enable a quick introduction.

Consequently, the following descriptions refer exclusively to using the sample project with S7-PLCSIM. If real hardware components are used, the following changes have to be made:

- Adjust the access point in the SIMATIC Manager
- Adjust HW-Config of the PC station
- Adjust HW-Config of the AS
  - If you want to use another type of CPU, please make sure to use the “Replace Object…” function. If you just delete the CPU and insert a new one, the S7 program will be lost.
- If required, adjust the network adapter in the system properties of the WinCC channel driver.

Note

This sample project has been created with PCS 7 V8.0 Update 1.

If you use a later PCS 7 version, the project must be updated.

The sample project cannot be used, if a version older than PCS 7 V8.0 Update 1 is installed.
In this case you may draw on the SCL source files of the blocks ARSCAN_E and AR_MAN_E and reproduce the sample project configuration as described in the following.
However, not every aspect of project creation will be mentioned. In case of doubt please refer to the online help and the documentation.

5.1 Installation of the project

5.1.1 Retrieval

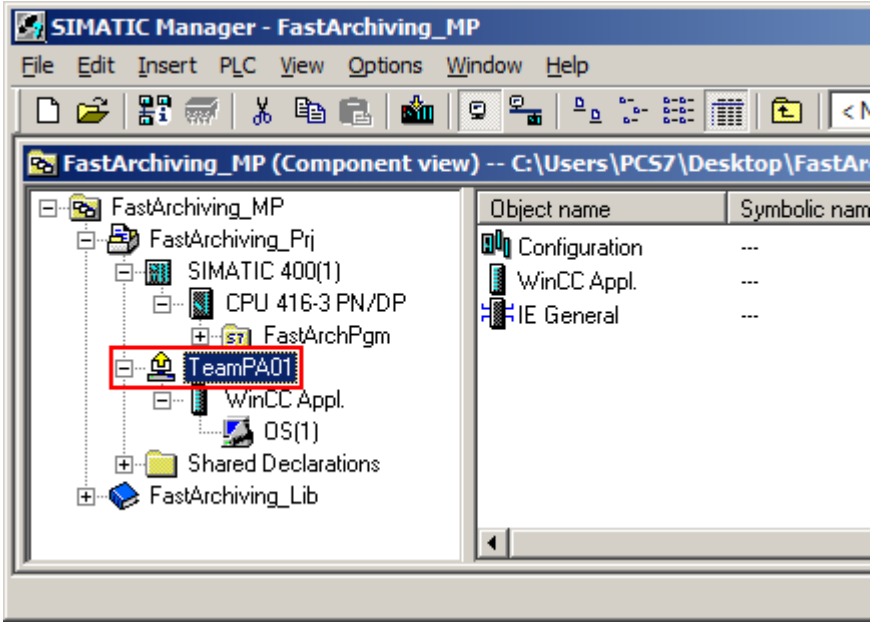
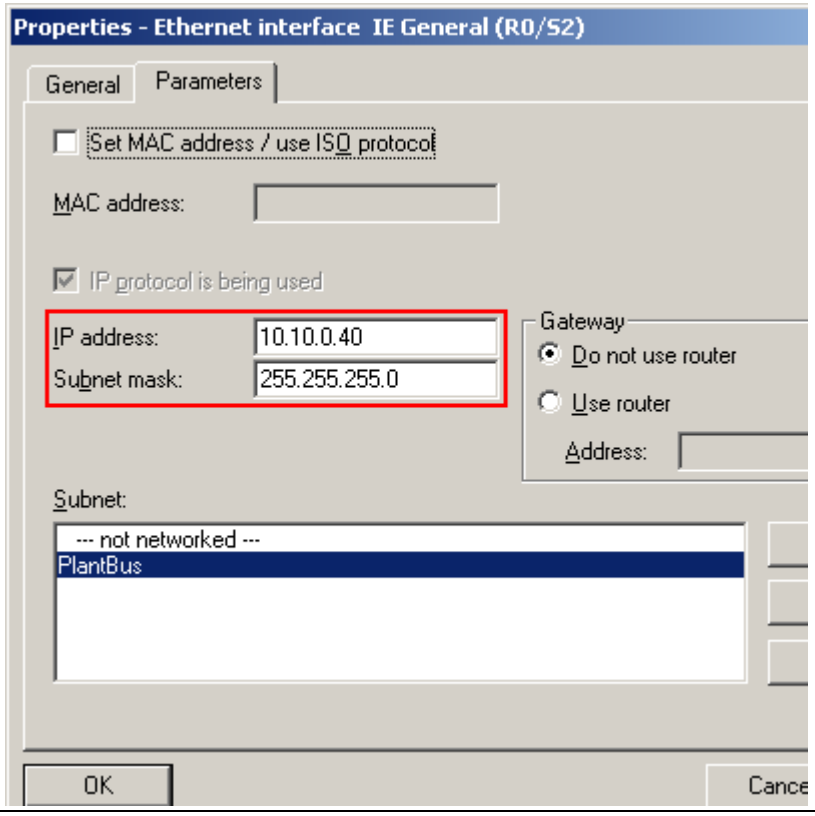
Table 5-1

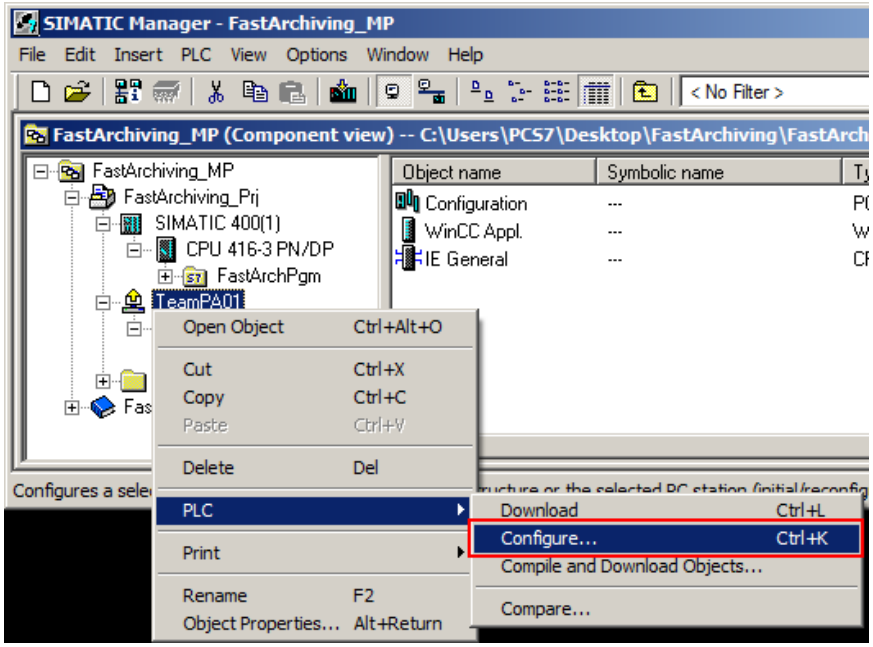
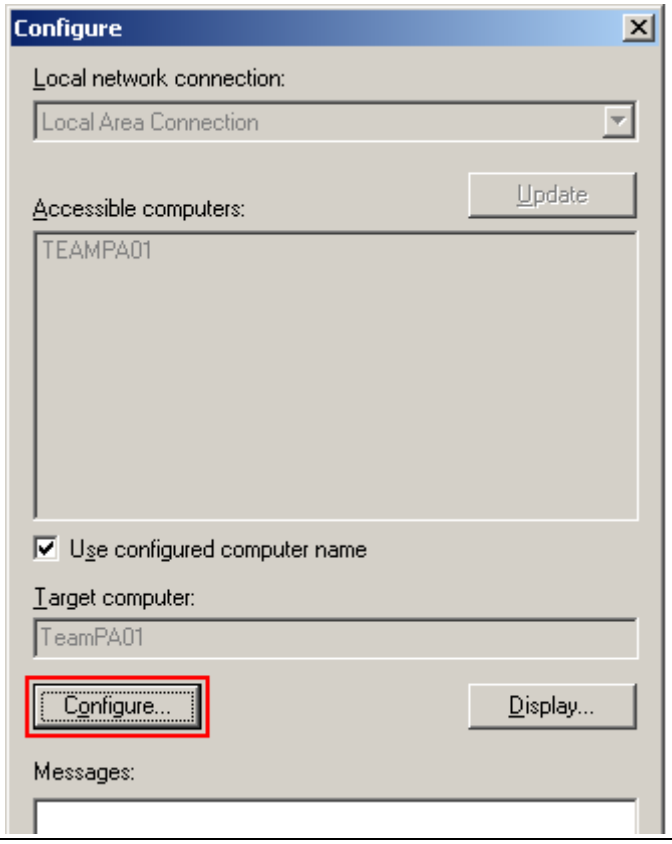
No.	Action
26.	Download the sample project and copy it to your Engineering Station.
27.	Open the SIMATIC Manager, retrieve the project and open it.

5.1.2 Required adjustments

Adapt the single station

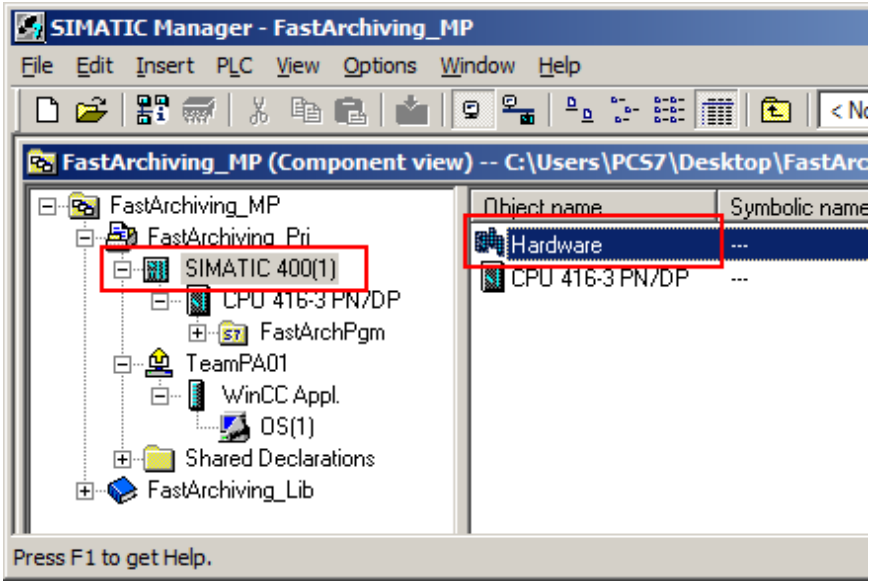
Table 5-2

No.	Action
28.	Adapt the name of the PC station to your computer name. 
29.	Open HW-Config of the PC station and specify an IP address in compliance with the address of your computer. 

No.	Action
30.	Save and compile HW-Config.
31.	Open the dialog window for configuration of the PC station. 
32.	Configure your PC station. 

Adjust the AS

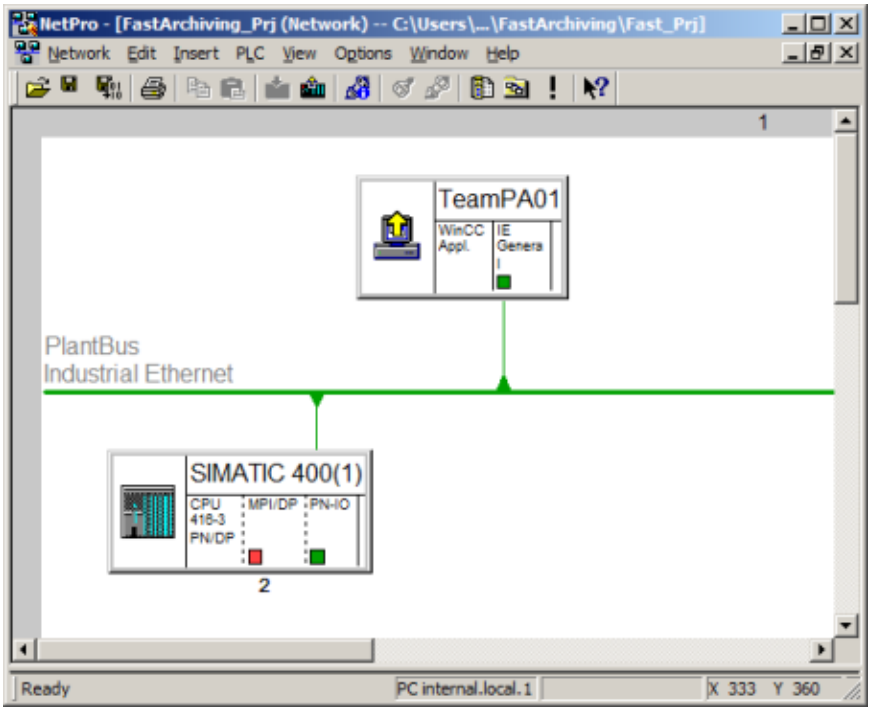
Table 5-3

No.	Action
33.	Open HW-Config of the AS and adjust the IP address of the Ethernet interface of the CPU, so that it belongs to the same network as your PC station. 
34.	Save and compile HW-Config.

Compile and load changes

After having made the necessary adjustments, the specified changes must be compiled and loaded.

Table 5-4

No.	Action
35.	Open NetPro by selecting the project "FastArchiving_Prj" and then “Options > Configure Network”. 
36.	Use the option “Compile and Check Everything” to compile the NetPro configuration.
37.	Select the PC station and then “PLC > Download to Current Project > Selected Station” to load the configuration into the PC station and then follow the dialog instructions.
38.	Click “Options > Simulate Modules” to open PLCSIM.
39.	Load the configuration into the AS.
40.	Load the program into the AS and start PLCSIM.

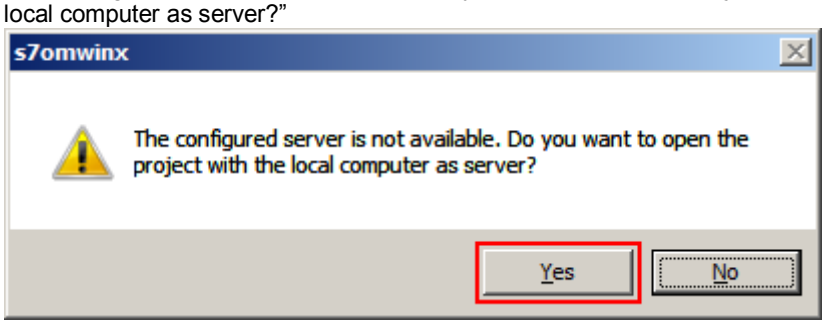
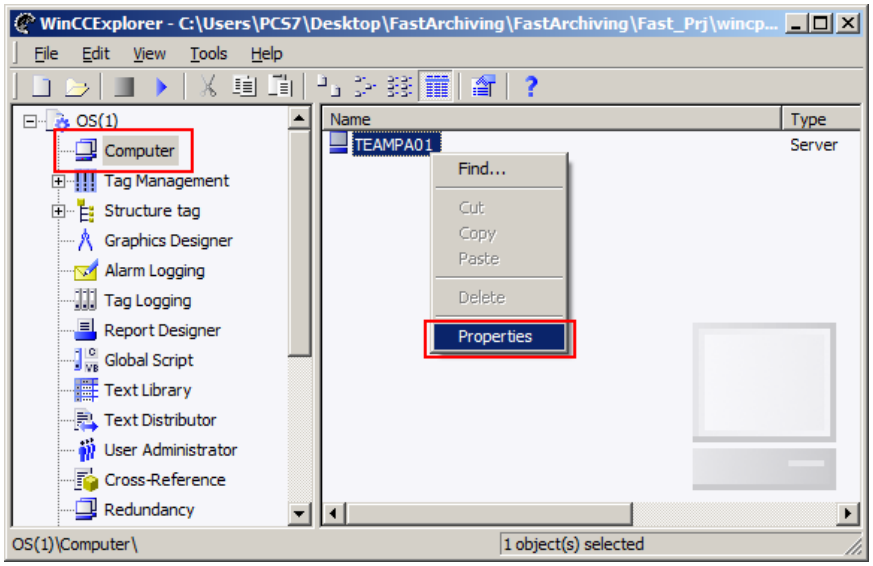
Note

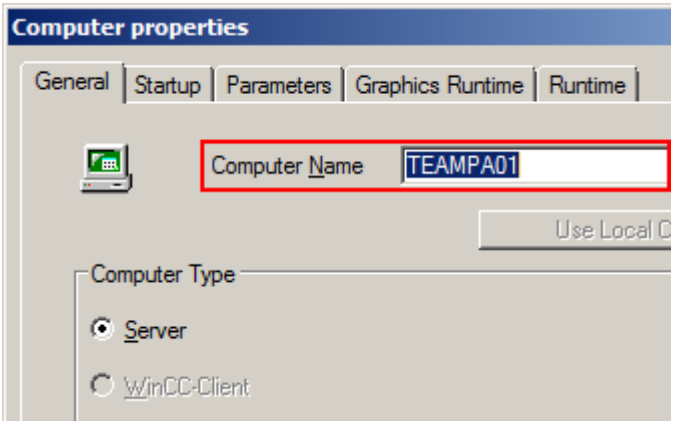
If downloading to the stations is not possible, please check your configurations:

- Network configuration (network card, IP addresses, firewall, switches, etc.)
- Check your configuration in the project.
- Perform the steps for configuring the PC station again.

Adjust the operator station

Table 5-5

No.	Action
41.	Open the WinCC project.
42.	If you use another computer name than defined for the sample project , the following message will appear: "The configured server is not available. Do you want to open the project with the local computer as server?"  Click "Yes" to confirm this message.
43.	Now, adjust the computer name in the WinCC project. To do so, select "Computer" on the left side and open the Properties dialog for the configured computer. 

No.	Action
44.	Enter the computer name. 
45.	Start the WinCC Explorer anew, so that the modification will become effective.
46.	Compile the OS. One change compilation is sufficient.

With these steps, all necessary adaptations are completed. You can now start WinCC Runtime and test the fast process value archiving function.

Note

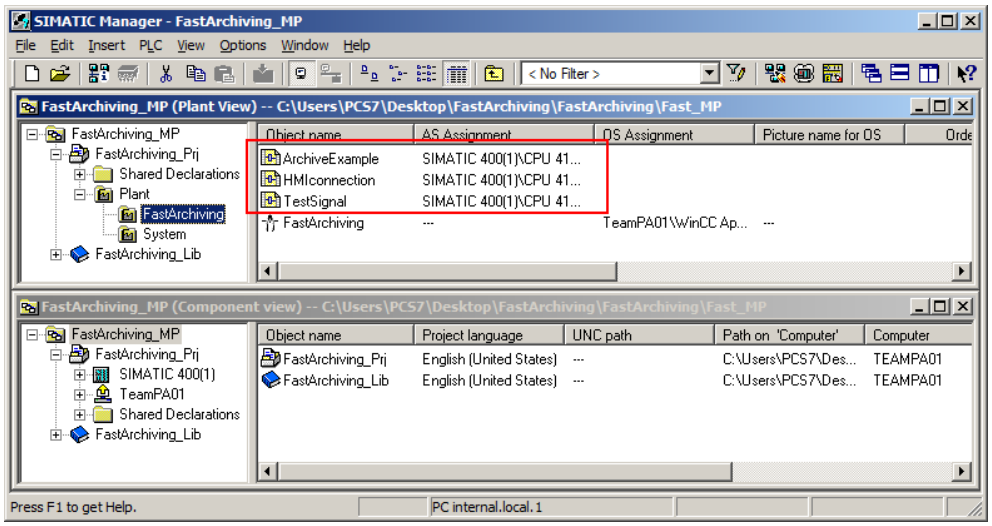
If WinCC Runtime cannot establish a connection to PLCSIM, please check your configuration.
Also check whether the correct network adapter has been selected in the system parameters of the WinCC channel driver.

5.2 Description of the sample project

AS section

The AS project includes three hierarchy folders. The OS section starts on the second hierarchy level.

Figure 5-1



- Plant
  - This is the central folder and it is used as a central point to allocate subordinate folders to the AS.
- System
  - This folder includes system-relevant program parts which are not required directly for fast process value archiving.
- FastArchiving
  - This folder includes the program for fast process value archiving and the picture for the OS project.

The following descriptions only refer to the CFC charts in the hierarchy folder “FastArchiving”.

Table 5-6

CFC name	Function
TestSignal	In this chart, a simple sawtooth signal is generated in order to visualize the function of fast process value archiving. This signal is generated by the incrementation and reset of a value. The chart is processed in OB35 (100ms cycle). The initial value 1 is incremented by 1 and will be reset at 100, resulting in a signal with a period length of 10 seconds and amplitude of 100. In this way, this singular signal is optimal for the analysis of the archived process values.
HMIconnection	This chart includes three blocks of type “OpDi01” which enable the binary operation of WinCC. • Block “DisableTestSig” - If the interlinked output of this block shows a logic 1, the test signal will be deactivated and reset to 1.

CFC name	Function
	- Operation can be effected via HMI or CFC. The input parameter “LiOp” is used to determine whether the block shall be operated via HMI or CFC. • Block “DisableSigLink” - With the input parameter “In”, the block “DisableTestSig” can be controlled via CFC (if the parameters have been defined accordingly). • Block “EnableArchiving” - This block is used to activate/deactivate archiving at the blocks ARSCAN_E and AR_MAN_E. • Block “EnableArchLink” - With the input parameter “In”, the block “EnableArchiving” can be controlled via CFC. • Block “ResetSigOnStart” - If the interlinked output of this block shows a logic 1, the test signal will be reset to 1 when activated (start from the beginning). - Operation can be effected via HMI or CFC. The input parameter “LiOp” is used to determine whether the block shall be operated via HMI or CFC. • Block “RstSigOnStartLnk” - With the input parameter “In”, the “EnableArchiving” block can be controlled via CFC.
ArchiveExample	This chart includes the example for fast process value archiving. • Two channel blocks (Pcs7AnIn) have been placed. The test signal is interconnected with these blocks as a simulation value and simulation at the channel block is activated. • After the channel blocks, the output structure is broken down into its elements (process value and quality) each of which is then interconnected with an ARSCAN_E block. • Block “Scan01” - This block is configured for event-controlled acquisition of the process value, but after 5 seconds at the latest. - SUB_ID = 1 DEADBAND = 1 MIN_CYC = 0 MAX_CYC = 5 - Since the test signal may show values between 1 and 100 and changes by 1 in each cycle, every single value will be archived by the test signal. If the test signal is not active, the value will be archived at 5-second intervals. • Block “SCAN_02” - This block is configured for cyclic process value acquisition. - SUB_ID = 2 DEADBAND = 0 MIN_CYC = 0.2 MAX_CYC = 0 - The parameter UPD_EVT is interconnected with the signal for starting the archiving process, so that the first value of the data series will be archived also. - Through the cyclic archiving at 200ms-intervals, every second value of the test signal will be archived, for example: 1, 3, 5, 7, 9, 11, etc. • Block “Archive01” - This block is interconnected with the two ARSCAN_E blocks with the parameter configuration AR_ID = 1. The remaining

CFC name	Function
	parameters remain in their default setting.

OS section

Configuration of WinCC Tag Logging is performed as described in chapter 4.4.

The “FastArchiving” screen representation in the hierarchy folder of the same name includes some elements used to operate the sample program and to display, and possibly export, the archived values.

The individual display elements are described in the following sections.

Figure 5-2

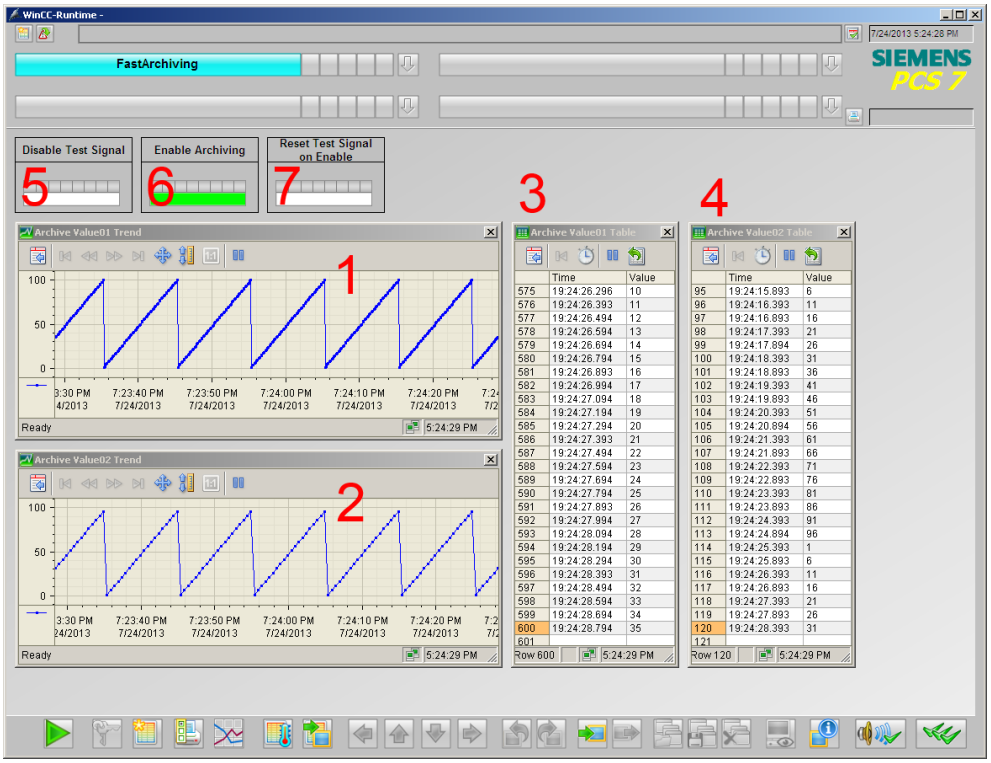


Table 5-7

No.	Object type	Description
1	WinCC OnlineTrendControl	This trend shows the values of the archive tag “#FastArchPgm#RawArchiv#A#00000001#0001” acquired by the CFC block Scan01.
2	WinCC OnlineTrendControl	This trend shows the values of the archive tag “Value02” acquired by the CFC block Scan02.
3	WinCC OnlineTableControl	This table shows the values of the archive tag “#FastArchPgm#RawArchiv#A#00000001#0001”. With WinCC OnlineTableControl you can export the archived values to a CSV file. Please note that the export only includes the values of the defined time period.
4	WinCC OnlineTableControl	This table shows the values of the archive tag “Value02”.

No.	Object type	Description
5	APL OpDi01 block icon	This binary process operation is used to deactivate the test signal. If this operation is active, the test signal will be reset to 1 and hold.
6	APL OpDi01 block icon	This binary process operation is used to start or stop the archiving process.
7	APL OpDi01 block icon	If this process operation is activated, the test signal will be reset to 1 each time when the archiving process starts.

6 Operation of the sample project

6.1 Operation via OS Runtime

The easiest way to operate the application example is using OS Runtime. This is effected with the configured elements in the OS area “FastArchiving” as described in the previous chapter.

6.2 Operation via CFC

As an alternative to OS Runtime, you may also operate all configured functions directly via CFC.

- Start / stop archiving
  - Open the CFC chart “HMIconnection”.
  - Activate the test mode in CFC
  - At the block “EnableArchiving”, set the input parameter „LiOp“ = 1.
    The block will then be controlled via the inputs “SetLi” and “RstLi”.
  - Start the archiving process by setting the input parameter “In” at the block “„EnableArchLink” to 1
  - Stop the archiving process by setting the input parameter “In” at the block “„EnableArchLink” to 0
- Deactivate test signal
  - Deactivate the test signal in the same way as described above.
- Reset test signal when starting the archiving process
  - Reset the test signal when starting the archiving process in the same way as described above.

6.3 Reset of archives

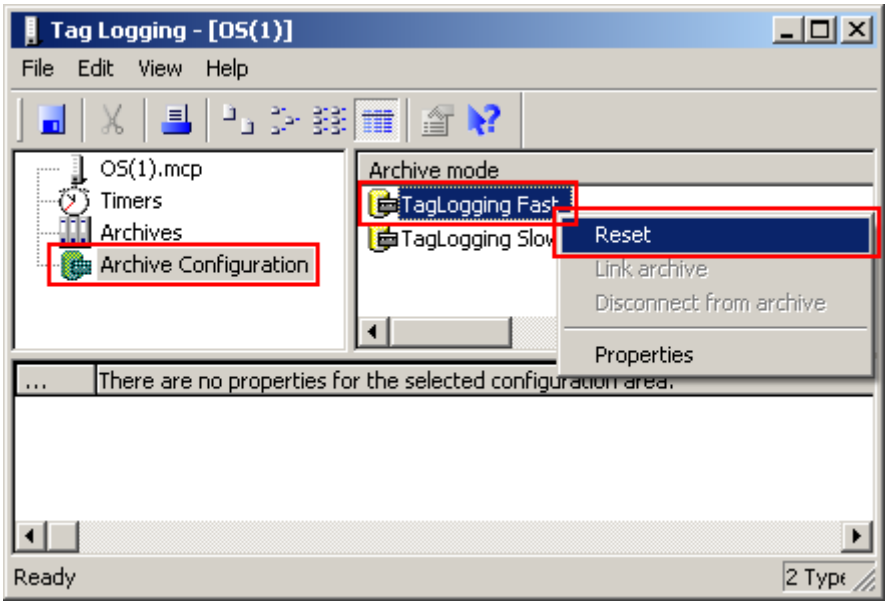
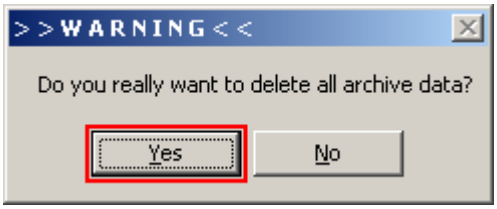
Testing of the fast process value archiving function may involve a large amount of data which is not necessarily needed and only occupies a lot of memory space.

You can delete this data by resetting the WinCC archive “Tag Logging Fast“.

NOTICE **Please note that a reset of this archive will also delete all data included in this archive for the project.**

To reset the archive, proceed as follows:

Table 6-1

No.	Description
1	Open the WinCC-Tag Logging window in the WinCC Explorer.
2	Select “Archive Configuration” on the left side of this window.
3	Click “TagLoggingFast” on the right side with your right mouse button and select “Reset”. 
4	If you are sure that you really want to delete all data included in this archive, confirm the following warning message with “Yes”. 
5	Save and close the WinCC Tag Logging window.

7 Related literature

Table 7-1

	Topic	Title / Link
\1\	Siemens Industry Online Support	http://support.automation.siemens.com
\2\	Download page of this entry	http://support.automation.siemens.com/WW/view/en/ 23780904
\3\		

8 History

Table 8-1

Version	Date	Modifications
V1.0	12/2006	First issue
V1.1	03/2009	• Changing the time calculation from TIME to REAL • Supplementing the following block connections/functions - SAMPLE_TIME / cycle time of the OB - QFIFODEG / filling level of the memory - QNUM_TRANS / archived process value - QNUMSEND / sent data blocks - QNUMCONNECT / ARSCAN_E connected
V2.0	11/2013	• Complete revision of the document - New structure of the chapters - Validity for PCS 7 V8.0 Update 1 - Engineering details for quantity structures / limitations • New sample project based upon PCS 7 V8.0 Update 1 • SCL sources in version 1.2 are in inofficial / project specific released • Update of SCL sources to version 1.3 (for a complete list of changes please have a look into the SCL sources) - **ATTENTION Change of interface** - New structure of version history - Revision of comments - Cleanup of internal data structure - Improvement of run-up behavior