SIEMENS

Application Example • 02/2015

Fast archiving of process data in PCS 7 with AR_SEND

SIMATIC PCS 7 or STEP 7 with WinCC

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

Warranty and liability

Table of contents

1 Task

1.1 Overview

Introduction

In some processes, there is a requirement for data to be archived in rapid cycles. The cycle time of conventional archiving with WinCC Tag Logging is at least 500ms.

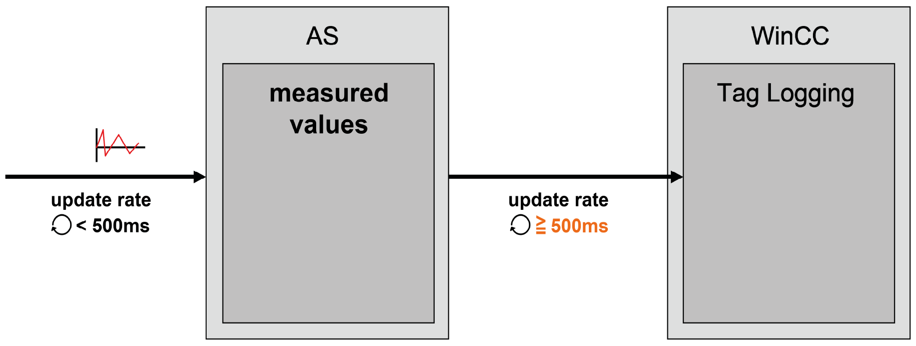
To archive data faster than 500ms, one must find another solution.

To achieve faster archiving of process data, it is also important to have a low load on the system bus.

Overview of the automation task

The following figure provides an overview of the automation task.

Figure 1–1



1.2 Requirements

In order to further improve the existing process data archiving, the following requirements must be met.

- Faster archiving rates than 500ms
- The load on the system bus must be kept as low as possible
- The solution should apply for the SIMATIC PCS 7

2 Solution

2.1 Overview

The S7-400 / WinAC RTX system function block AR_SEND / SFB 37 is the ideal choice for applications requiring faster process data archiving.

This block is responsible for the communication between the automation system (AS) and the operator station (OS) and sends the data to WinCC where it is then added to the archive "TagLogging Fast".

Based on this block, we provide two function blocks (ARSCAN_E and AR_MAN_E), which allow you to achieve a higher data throughput when archiving measured values, compared to requesting the data on the initiative of the OS.

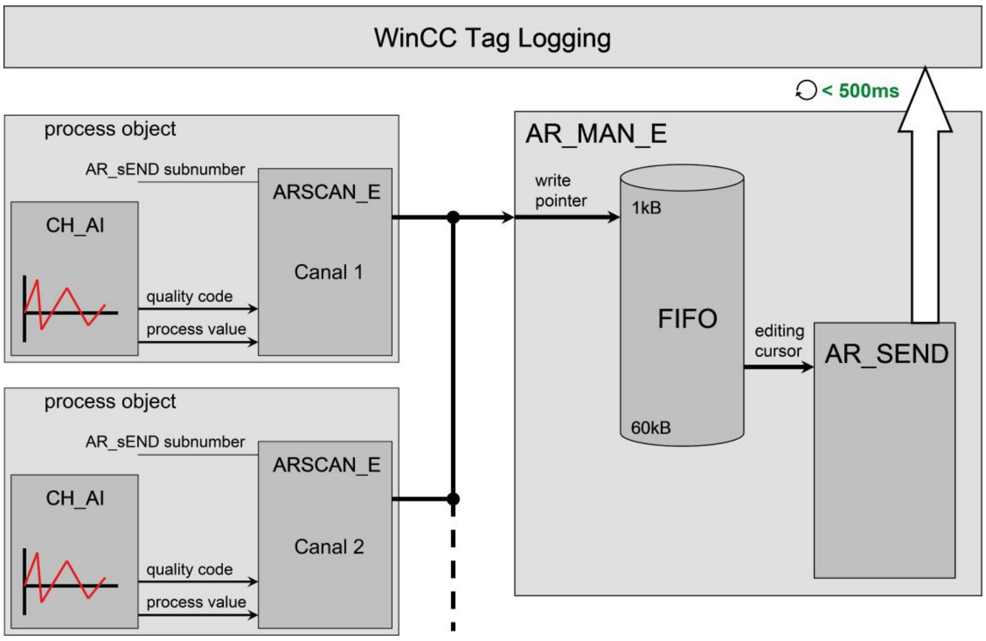
To keep the load on the system bus as low as possible, the process data is stored in the AS and sent in bundles to the OS.

The function blocks ARSCAN_E and AR_MAN_E have been developed based on the system function block AR_SEND / SFB 37 to save measured values as efficiently as possible and to transmit it in bundles to the OS.

- The block ARSCAN_E is used for capturing and saving one single process value of the data type REAL or DWORD.
  An ARSCAN_E block is used for each process value.
  There are two ways of capturing the process data.
  - Cyclical acquisition of process data is used to archive process values in an equidistant manner (at fixed intervals of time).
  - Event-driven acquisition of process data is used if a process value is to be archived as soon as its value changes. The use of this variant is limited to process values of data type REAL.
- The block AR_MAN_E includes FIFO memory for process data and manages the sending of bundled data to WinCC by means of the system function block AR_SEND.
  Several ARSCAN_E blocks can be connected to a single AR_MAN_E block (one ARSCAN_E per process value) and write to the FIFO memory.

The following figure shows a schematic representation of the two blocks used for the fast archiving of process data.

Figure 2–1



Typical areas of application

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

Advantages

- Easy interconnection of process tags in the CFC editor
- Simple parameter assignment of the system function block AR_SEND
- Providing the necessary header information for the AR_SEND block
- Precise time stamp on process data from the AS
- Variable trigger options for sending the data to WinCC
- Automatic frame repetition in case of error

Required knowledge

It is required to have basic knowledge of automation systems and SIMATIC PCS 7. This application example was created with SIMATIC PCS 7 V8.1 (SCL V5.3 SP6 Upd3).

2.2 Hardware and software components

2.2.1 Scope

Valid for SIMATIC PCS 7 V8.1 and WinCC V7.3 update 1

Note The use of this solution is also possible in STEP 7 and WinCC, but it is not explicitly described in this example.

The only prerequisite is the configuration and runtime environment with the appropriate licenses and an automation system that supports the system function block AR_SEND / SFB 37.

2.2.2 Components used

The Application Example has been created with the following components:

Hardware components

Table 2–1

Component	Qty.	Order number	Note
Rack	1	6ES7400-1JA01-0AA0	or equivalent
PS 407 10A power supply	1	6ES7407-0KA01-0AA0	or equivalent
SIMATIC S7-416-3 PN/DP	1	6ES7410-5HX08-0AB0	or equivalent S7-400 / S7-400 H WinAC RTX

Software components

Table 2–2

Component	Qty.	Order number	Note
SIMATIC PCS 7 V8.1	1	6ES7651-5AA18-0YA0	ES Singlestation V8.1
S7-PLCSIM	1	6ES7841-0CC05-0YA5	If the AS of the example project is to be simulated with S7-PLCSIM

Example files and projects

The following list contains all the files and projects used in this example.

Table 2–3

Component	Note
23780904_FastArch_mp_V81.zip	This file complies with the PCS 7 example project and can be retrieved directly via the SIMATIC Manager.
23780904_Code.zip	The file contains the SCL sources of the blocks: • AR_MAN_E • ARSCAN_E

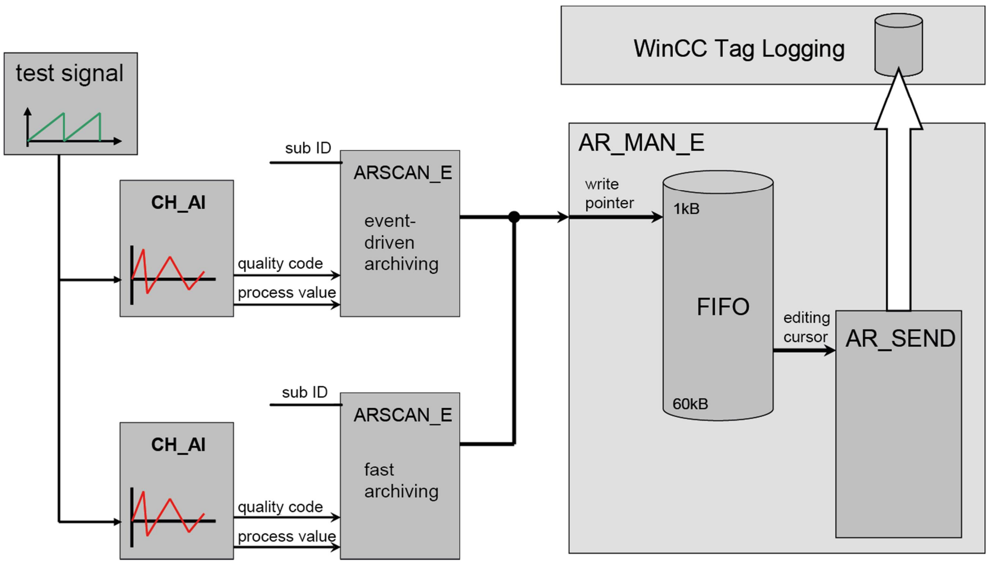
3 Basics

3.1 Operating principle of the Application Example

Two ARSCAN_E blocks detect a test signal.
The first block saves the process value in an event-driven manner, and the second block in a cyclic manner.
The block AR_MAN_E allocates FIFO memory for the process data and by means of AR_SEND, it sends the data in blocks to WinCC, where it is then added to the archive "Tag Logging Fast".

The following figure shows a schematic illustration of the functioning of the Application Example.

Figure 3–1



CAUTION	**Since the block ARSCAN_E is connected to the block AR_MAN_E and it accesses its information, the block ARSCAN_E should be called up in the run sequence following the block AR_MAN_E (to ensure the timeliness of this information).** **However, with this constellation, the current process data in the block AR_MAN_E won't be processed before the next cycle.**

3.2 Functionality of the block AR_SEND

Overview

You can find detailed information about the AR_SEND block in the WinCC Online Help at

"WinCC Information System > Communication > SIMATIC S7 Protocol Suite > Special functions > Data exchange with the S7 function block AR_SEND".

Archive data transmission using AR_SEND

The AR_SEND block can supply various versions of archive tags with data.

- One archive tag (AR_ID sufficient)
- Multiple archive tags (AR_ID and AR_ID subnumber)

AR_ID and AR_ID subnumber establish the assignment between the data in the AS and the archive tags.
Each AR_ID can have theoretically up to 4095 AR_ID subnumbers. However in practice, the number of archive tags per AR_SEND instance is limited by the maximum length of the data area to be transmitted.

The AR_SEND block also offers a variety of formats to transfer process data in the WinCC Tag Logging, e.g. more process values with the same time stamp or process values with time difference to the last value.

AR_SEND Performance

It is to be noted that there is a resource restriction when using the S7 functions AR_SEND and BSEND/BRCV for communication with S7-400.
That is, the maximum amount of data that can be transmitted simultaneously from AS to WinCC with the functions AR_SEND and/or BSEND/BRCV is limited to max. 16 kByte.

CAUTION	**The data amount limit in the WinCC communication channel must be always observed. Please also note that in case of redundant systems, the maximum amount of data is halved.** **For detailed information about the resource limit, please read the paragraph "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> Structure and parameters of a data block".**

Limitation of AR_SEND instances

The maximum number of AR_SEND instances that can be connected simultaneously to each CPU is limited and depends on the CPU type.

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

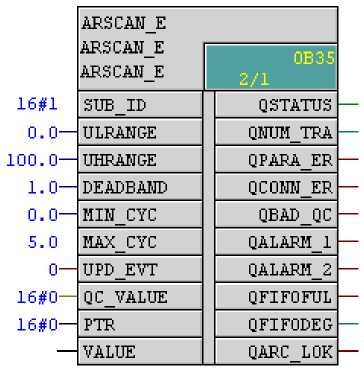
Note For current information about the various CPUs and the AR_SEND instances, please refer to the "SIMATIC S7-400 Automation System S7-400 CPU Specifications" manual.

3.3 Functionality of the block ARSCAN_E

3.3.1 Overview / Function

This chapter describes the block ARSCAN_E in detail.

Figure 3–2



The block ARSCAN_E detects a process value of the data type REAL or DWORD and the associated quality code. To do this, the block is connected in the CFC editor to the corresponding channel block (e.g. Pcs7AnIn) or another data source. The ARSCAN_E block is called up in the acquisition cycle of the process value (e.g. according to the channel block).

Each ARSCAN_E block is connected to an AR_MAN_E block, whereby an AR_MAN_E block can connect to several ARSCAN_E blocks.

Process values of the REAL data type require process value limits to be set at the input parameters ULRANGE and UHRANGE.
The acquired process value is checked for validity ("QC_VALUE" quality code) and modification.
Value changes to REAL data types are identified when the process value has changed by a certain percentage when compared to the last measured process value. This change can be set by the user by means of a percentage value based on the measurement range (DEADBAND).
With respect to the DWORD data type, binary change monitoring is achieved, i.e. the input parameter DEADBAND is ignored.

Depending on the parameter assignment of the ARSCAN_E block, one can distinguish between two different criteria for storing a process value.

The following sections describe the two types in detail:

- Cyclic process value detection
- Event-driven process value detection

3.3.2 Cyclic process value detection

The cyclic process value detection can be used for the data types REAL and DWROD. The following settings are required.

- Input parameters:
  - DEADBAND = 0
  - MIN_CYC adjust accordingly
  - MAX_CYC adjust accordingly

Note It is important to ensure that MAX_CYC > MIN_CYC, otherwise an error is generated.

Output (of the block ARSCAN_E) QPARA_ER = TRUE

This enables the process value to be saved as a constant bus cycle time (MIN_CYC).
Note that the minimum cycle time is limited by the cycle in which the ARSCAN_E block is called up.

Note By constantly saving process values, the amount of data can become very large (especially when the cycle time is low).

Note The security level 1 is disabled: Double DEADBAND.

3.3.3 Event-driven process value detection

The event-driven process value detection can only be used for the data type REAL. The following settings are required.

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

Therefore, the process value is stored if the value change in percent for the last cycle is greater than or equal to the value of the input parameter DEADBAND.

Note The event-driven storage of the process values results in a reduction of data traffic, when compared to the cyclic process value detection.

Note The security level 2 is disabled: Increase the archiving cycle to MAX_CYC.

3.3.4 Saving process values

The block AR_MAN_E allocates FIFO memory for the process values.

The block ARSCAN_E stores the collected process data in this memory.
To do this, these two blocks are connected to each other via an interconnection that provides all the necessary information about the memory area.

If the conditions for storing a process value are met, the block ARSCAN_E writes the process value with the current timestamp in the FIFO memory and then increments the write pointer of the FIFO memory so that it points to the next free space.

You can find a detailed description of the block AR_MAN_E in the next chapter.

Note After saving a process value, the output QNUM_TRANS is incremented by the value 1.

The allocation of the FIFO memory can be read from the output QFIFODEG.

3.3.5 Responding to overload or connection failure

If the FIFO of the block AR_MAN_E is too full, the amount of data to be stored is automatically reduced (security level 1, security level 2). This function is described in detail in chapter 3.4.3.

3.3.6 Block I/Os

Input parameters

Table 3–2

Parameter	Data type	Description
EN (hidden)	BOOL	Activates block processing *Standard value: 1*
ENABLE (hidden)	BOOL	Activates the detection of the process values. *Standard value: 1*
RUNUPCYC (hidden)	INT	Number of block cycles before starting. *Standard value: 5*
SAMPLE_TIME (hidden)	REAL	Cycle time of the cyclic interrupt OB in which the block is called. It is set by the compiler.
SUB_ID	WORD	AR_ID subnumber for the assignment of the process value to an archive tag in a process value archive. *Allowed values: 1 - 4095*
ULRANGE	REAL	Lower limit of the measuring range (only relevant for process values of the data type REAL). *Standard value: 0.0*
UHRANGE	REAL	Upper limit of the measuring range (only relevant for process values of the data type REAL). *Standard value: 100.0*
DEADBAND	REAL	Minimum change of the process value (in percent) from the previous value, for it to be stored (only relevant for process values of the data type REAL). *Standard value: 0.5* **Remark:** The percentage refers to the parameters ULRANGE and UHRANGE
MIN_CYC	REAL	Minimum detection time (in seconds). *Standard value: 0.0*
MAX_CYC	REAL	Maximum detection time (in seconds) of the process value (archived, even if change < DEADBAND). *Standard value: 5.0* **Note:** MAX_CYC > MIN_CYC otherwise an error is generated -> Output QPARA_ER = TRUE
UPD_EVT	BOOL	Edge triggering of an archiving process.

Parameter	Data type	Description
UPD_COND (hidden)	BOOL	Condition triggering of an archiving process.
QC_VALUE	BYTE	Quality code of the process value (must be interconnected or have its parameter assigned with 16#80).
PTR	STRUCT	Interconnection with AR_MAN_E block. Provides information for ARSCAN_E and a pointer to the FIFO memory.

In-out parameter

Table 3–3

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

Output parameters

Table 3–4

Parameter	Data type	Description
ENO (hidden)	BOOL	Block processing is activated.
QSTATUS	WORD	Status word for the block.
QNUM_TRANS	INT	Number of archived process values.
QUPD_EVT (hidden)	BOOL	It is set for one cycle if the block writes to the FIFO memory of the AR_MAN_E block.
QPARA_ER	BOOL	Wrong parameter assignment of the block.
QCONN_ER	BOOL	Incorrect interconnection of the input parameter VALUE.
QBAD_QC	BOOL	Quality code of the process value is poor. No archiving of the process value.
QALARM_1	BOOL	Security Level 1 has been reached
QALARM_2	BOOL	Security Level 2 has been reached
QFIFOFUL	BOOL	FIFO memory of the module AR_MAN_E is full. It is not possible to archive the process value.
QFIFODEG	INT	FIFO memory allocation in %
QARC_LOK	BOOL	Archiving at the block AR_MAN_E is deactivated.

Allocation of the status word QSTATUS

Table 3–5

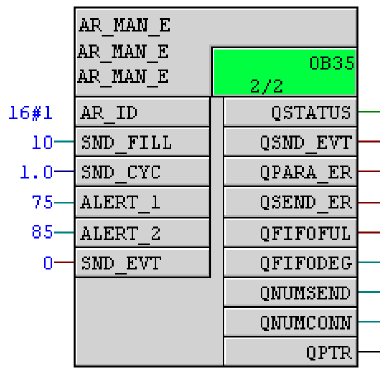
Bit	Byte	Description
0	Byte 0	QC = 16#00
1		PTR invalid or not connected
2		VALUE not connected
3		VALUE has a wrong data type
4		Security Level 1 reached
5		Security Level 2 reached
6		FIFO memory allocation is too high (FIFO full)
7		Archiving lock of AR_MAN_E active
8	Byte 1	AR_ID-Subnummer < 1
9		Invalid parameter assignment of MIN_CYC & MAX_CYC
10		Parameter assignment of DEADBAND is incorrect (< 0% or > 100%)
11		ULRANGE ≥ UHLRANGE
12		Not used
13		Not used
14		Not used
15		Not used

3.4 Functionality of the block AR_MAN_E

3.4.1 Overview / Function

This chapter describes the block AR_MAN_E in detail.

Figure 3–3



The block AR_MAN_E manages the memory area of the process values and sends the data to WinCC via AR_SEND.

Due to the CPU-dependent limitation of AR_SEND instances, only one instance of the AR_SEND block is used per AR_MAN_E block instance.
An AR_MAN_E block can manage up to 4095 ARSCAN_E blocks (4095 AR_ID subnumbers are possible per AR_ID). This allows multiple process values to be detected and sent to WinCC with an AR_MAN_E instance. However, the actual number of ARSCAN_E blocks is reduced due to the resulting data traffic and the individual application.

A transmission is started when:

- The FIFO has reached the configured memory allocation
- The configured time for the transmission has elapsed
- A send operation has been manually triggered

If a send operation is triggered, the process values are sent to WinCC using an AR_SEND call.

Note

The FIFO memory allocation is displayed at the output QFIFODEG.

The output QNUMSEND indicates the number of data blocks sent.

The output QNUMCONNECT indicates the number of connected ARSCAN_E blocks. The counting takes place only in the startup characteristics of the block and therefore, the control system must be restarted for the output to be updated.

3.4.2 Used AR_SEND format

The AR_SEND format 9 (Header Type = 9) is used for the transfer.
This format allows the transmission of values with individual time stamps and different AR_ID subnumbers.

26 bytes are needed in the frame for a 4-byte value (REAL / DWORD).

Table 3–6

22 bytes	Headertype = 9			
	Year		Month	
	Day		Hour	
	Minute		Second	
	1/10s	1/100s	1/1000s	Day of the week
	Cylce = 0			
	AR_ID subnumber			
	Data type of the process data			
	Number of process values			
4 Byte	Process value			

CAUTION	**The parameter AR_ID is only evaluated when the program starts. You cannot change the AR_ID subsequently during operation.**

3.4.3 Responding to overload or connection failure

When the AR_MAN_E block is overloaded or in case of connection problems with WinCC, more process values are written to the memory than can be read from it and sent to WinCC. To delay the filling up of the memory, 2 programmable security levels have been created.

- Security Level 1 (input parameter: ALERT_1 in %)
  - The ARSCAN_E block doubles the input parameter DEADBAND internally. Security Level 1 only functions with process values of the data type REAL.
- Security Level 2 (input parameter: ALERT_2 in %)
  - The ARSCAN_E block increases the logging cycle to MAX_CYC.

Despite these measures, it may happen that the allocation of the FIFO memory increases to 100%. In this case, no new process values are stored until the memory usage drops again.

Note In general, one must ensure that the quantity structure of the process values remains manageable.

3.4.4 Block I/Os

Input parameters

Table 3-7

Parameter	Data type	Description
EN (hidden)	BOOL	Activates block processing. *Standard value: 1*
ENABLE (hidden)	BOOL	Activates the detection of the process values. *Standard value: 1*
RUNUPCYC (hidden)	INT	Number of block cycles before starting. *Standard value: 5*
SAMPLE_TIME (hidden)	REAL	Cycle time of the cyclic interrupt OB in which the block is called. It is set by the compiler.
AR_ID	DWORD	Archive number for the assignment of an AR_SEND instance to an archive tag. It is set by the compiler.
SND_FILL	INT	Allocation of FIFO memory required in % to initiate a transmission. *Standard value: 10*
SND_CYC	REAL	Transmission cycle After this time, the send operation is initiated with the available data. *Standard value: 1.0*
ALERT_1	INT	Allocation of FIFO memory required in % to activate the security level 1. *Standard value: 75*
ALERT_2	INT	Allocation of FIFO memory required in % to activate the security level 2. *Standard value: 85*
SND_EVT	BOOL	Edge triggering of a sending process.

Output parameters

Table 3–8

Parameter	Data type	Description
ENO (hidden)	BOOL	Block processing is activated.
QSTATUS	WORD	Status word for the block.
QSND_EVT	BOOL	Is set for one cycle if a sending process was triggered.
QPARA_ER	BOOL	Wrong parameter assignment of the block.
QSEND_ER	BOOL	An error has occurred when processing the internal block AR_SEND.
QFIFOFUL	BOOL	The FIFO memory is full. It is not possible to archive the process value.

Parameter	Data type	Description
QFIFODEG	BOOL	FIFO memory allocation in %
QNUMSEND	INT	Number of data blocks transmitted
QNUMCONNECT	INT	Number of connected ARSCAN_E instances
QPTR	STRUCT	Interconnection with ARSCAN_E Provides information for ARSCAN_E and a pointer to the FIFO memory.

Allocation of the status word QSTATUS

Table 3–9

Bit	Byte	Description
0	Byte 0	FIFO is full, Archiving is locked
1		AR_ID < 1
2		ALARM_1 > ALARM_2 Or one of both ≥ 100 Or one of both ≤ 0
3		SND_FILL: > ALARM_1 > ALARM_2 ≥ 100 ≤ 0
4		SND_CYC < 0
5		Not used
6		Not used
7		Not used
8	Byte 1	Not used
…		…
15		Not used

4 Configuration

4.1 Preliminary remarks

The following section describes the configuration steps necessary to deploy the solution.
Basic configuration steps, such as creating a project, configuring the stations and their hardware, configuring the connections, etc. are not described in this Application Example.

An example project has been created and can be downloaded for testing purposes and to experiment with the solution. The chapters 5 "Example project" and 6 "Operation of the example project" contain information about the concrete use of the example project.

The screenshots and descriptions in this chapter have been prepared on the basis of the example project and serve to illustrate the basic procedure.

4.2 Important points before configuring

In order to ensure that the solution does not reach its limits, you must take some points into consideration before you start configuring the fast process value archiving for your application.

- Calculate the quantity structure of the process values to be archived carefully.
- Do not archive the values in cycles that are faster than necessary as this would increase the data traffic unnecessarily.
- Take note of the maximum usable number of AR_SEND instances per CPU, as described in Chapter 3.2 (an AR_MAN_E block instantiates an AR_SEND block).
- Take note of the limits of communication to WinCC as described in chapter 3.2. Should it be likely to come close to these limits, you should look at the issue in detail and calculate your quantity structure carefully.
- Test your quantity structure in advance. For example, with a project like the one shown this Application Example.
- Please note that, for example, the use of a CAS (Central Archive Server) / PH (Process Historian) for archiving long-term relevant archive tags can restrict the performance of the system and thus the quantity structure for fast archiving of process values.
- Consider that even the processing time of the complete S7 program plays a role.
- Other communications (e.g. Modbus communication, OPC communication at the OS level, etc.) must be also taken into consideration.
- Make sure that the time synchronization works properly in your system, as for example, time differences between the AS and the OS can lead to problems.
- Make sure that the communication load increases with redundant systems, or that the limits of communication to WinCC decrease.
- It is recommended to define a workaround on how to handle the stopping and starting of the OS runtime in combination with the fast process value archiving. For example, start archiving only when the OS Runtime is fully started and end the archiving before the OS runtime is stopped.

CAUTION	**Please note that no monitoring takes place during communication with redundant OS servers, whether there is connection to both servers or not. The AR_SEND system block only reports an error when the connection to both servers is interrupted.** **As a general rule, it is therefore recommended that when using redundant OS servers you define connection monitoring and a corresponding workaround.**

In summary, it can be said that there are many factors that affect the performance of the fast process value archiving. One must always consider the entire environment.

For larger quantity structures, it is advisable to test in advance to see whether they work on a general scale.

CAUTION	**Should there be any unexpected difficulties with the fast process value archiving, check the entire environment.** **Consider that there may also be problems with the network traffic on the system bus that affect negatively.**

4.3 Configuring in SIMATIC Manager

To use the solution you need the blocks ARSCAN_E and AR_MAN_E in your project. There are two ways of doing this, as described below.

Besides this, you must then configure the modules according to their requirements.

4.3.1 Integrating module ARSCAN_E and AR_MAN_E in the PCS 7 project

Importing the block

1. Import the blocks ARSCAN_E and AR_MAN_E from the example project into the master data library of your project.

   Be sure to also import the AR_SEND block, because it is used by the AR_MAN_E block.
2. Adjust the block numbers (e.g. FB672 to FBxxx), if this is required by your project.

Note To import blocks, it is recommended to do the following:

1. A library serves as a source for the block.
2. Place a Continuous Function Chart into the project (or library) where you want to import the blocks.
3. Search the Continuous Function Chart catalog for the library with the blocks that you want to import and drag it into the Continuous Function Chart.

This way you ensure that all required blocks are also imported and the entries are created in the symbol table.

Generating blocks from the SCL sources

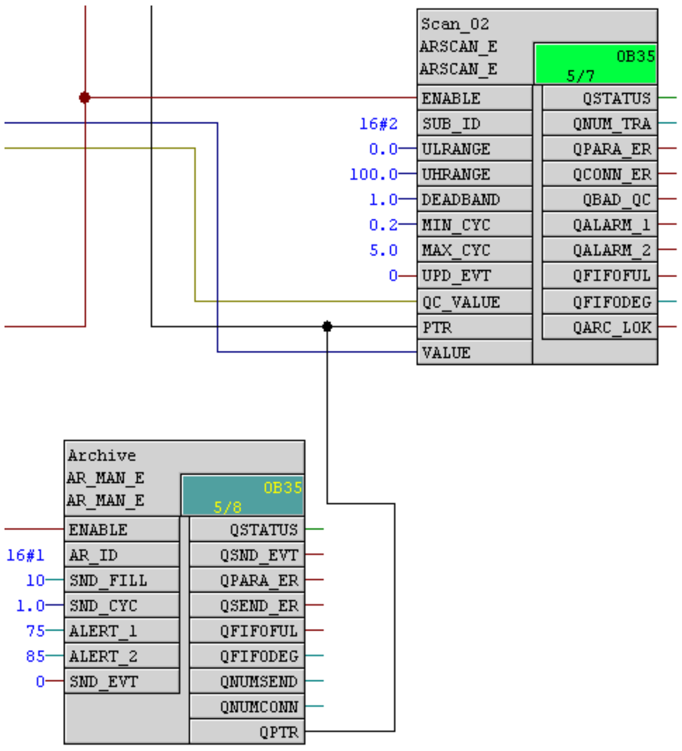
To create the blocks from the SCL source files, proceed as follows:

1. Import the SCL source files into your master data library.
2. Create the icons for the blocks in the symbol table:
   - FB 672 as ARSCAN_E (or another block number if necessary)
   - FB 673 as AR_MAN_E (or another block number if necessary)
3. Import the block AR_SEND into your master data library.
4. Compile the SCL source files.

4.3.2 Configuring AR_MAN_E

1. Place an AR_MAN_E block for each AR_ID that you want to use.
   It is up to you whether you want to archive all your process values with the same AR_ID or split them into several different ones.
   It would be conceivable to divide the process values according to subsystem, technological functions, archiving cycles, etc.
   Consider limiting the AR_ID subnumbers (and thus the archive tags) per AR_ID (4095 AR_ID subnumbers per AR_ID).
2. Set the remaining parameters and interconnections to the block according to your requirements.

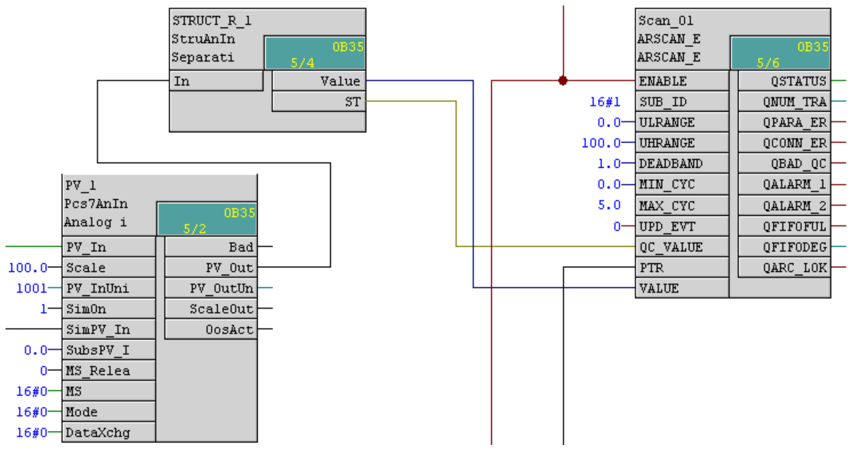
Figure 4–1



4.3.3 Configuring ARSCAN_E

1. Place an ARSCAN_E block for each process value (archive tag) that you want to archive.
2. Connect the input parameter VALUE to the source file of your process value. Please note that only the data types DWORD and REAL are permitted. If your process value is not of the correct data type, make sure that you perform a correct data type conversion.
3. Connect the input QC_VALUE with the quality code of the process value or assign it the value **16#80** (valid value).
4. Connect the input PTR to the output QPTR of the AR_MAN_E block, which creates a connection to the WinCC process value archive.
5. Ensure that the blocks have the correct execution sequence.
   By default, the ARSCAN_E blocks should run after other blocks from which you access process values and after the AR_MAN_E block, to which they are connected.
6. Set the remaining parameters and interconnections to the block according to your requirements.

Figure 4–2



CAUTION	**All configured AR_ID subnumbers must be configured in WinCC.** **If WinCC detects an AR_ID subnumber, which is not configured, it stops the interpreting of user data.**

Note

To see if your configuration works, for example, in order to ensure the completeness of the archives in case of large quantity structures, it is advisable to provide a simple signal (e.g. "sawtooth" function) and to optionally interconnect this with the ARSCAN_E blocks (e.g. as a simulation value at the channel block, such as indicated in Figure 4–2).

Therefore, for commissioning you can archive the test signal. Export the archive data via WinCC and, for example, analyze it in Microsoft Excel.

4.4 Configuring in WinCC

To save archive tags, you must create at least one process value archive.

It is up to you, whether you want to create your own process value archive for each configured AR_ID and each configured AR_MAN_E block.
Depending on how the process values to be archived are split, this could be quite useful, for example, to achieve a separate storage of the database segments or to ensure a clear arrangement of the project.

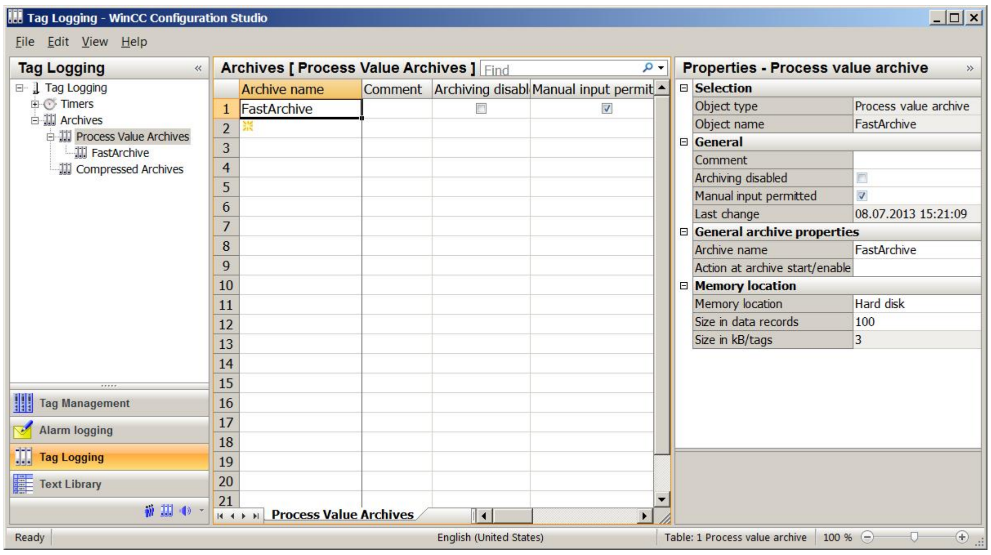
4.4.1 Creating a process value archive

1. Open the WinCC Tag Logging in WinCC Explorer.
2. Click on "Archives > Process Value Archives" and create a new archive.
3. Click inside the field marked with the asterisk in the column "Archive name" and enter an appropriate name.
4. Configure this archive to your needs.
5. The process value archive has been created.
   To create more process value archives, repeat the mentioned steps.

Example project

In the example project, a process value archive was created with the name "FastArchive".

Figure 4–3



	Archive name	Comment	Archiving disabl	Manual input permit
1	FastArchive		☐	☑
2				

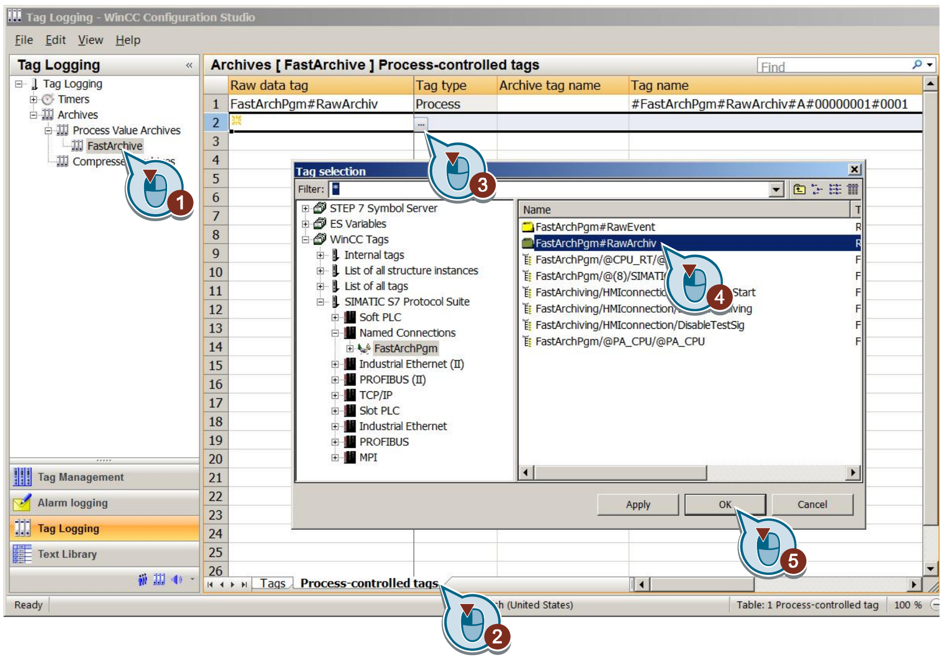
Selection	
Object type	Process value archive
Object name	FastArchive
General	
Comment	
Archiving disabled	☐
Manual input permitted	☑
Last change	08.07.2013 15:21:09
General archive properties	
Archive name	FastArchive
Action at archive start/enable	
Memory location	
Memory location	Hard disk
Size in data records	100
Size in kB/tags	3

4.4.2 Applying archive tags

Create an archive tag for each process value you would like to archive.
Should you have chosen to use several process value archives, perform the steps in the process archive, to which the tag is to be assigned.

6. Create a new archive tag.
   Click on the archive (Example: "FastArchive") and then on the "Process Controlled Tags" tab.
7. Click inside the field marked with the asterisk in the column "Raw data tags" and click on the button ... .
8. In your S7 program, select the raw data type for archives "[Program name]#RawArchive" and confirm with "OK".

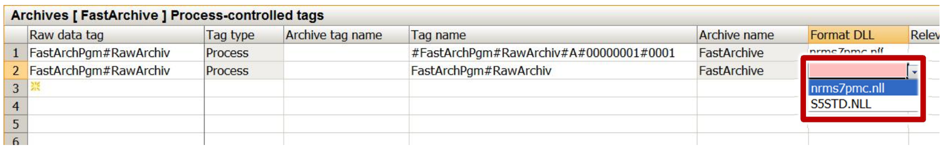
Figure 4–4



	Raw data tag	Tag type	Archive tag name	Tag name
1	FastArchPgm#RawArchiv	Process		#FastArchPgm#RawArchiv#A#00000001#0001

9. Select an appropriate DLL normalization format for the process tag to be archived.

Figure 4–5

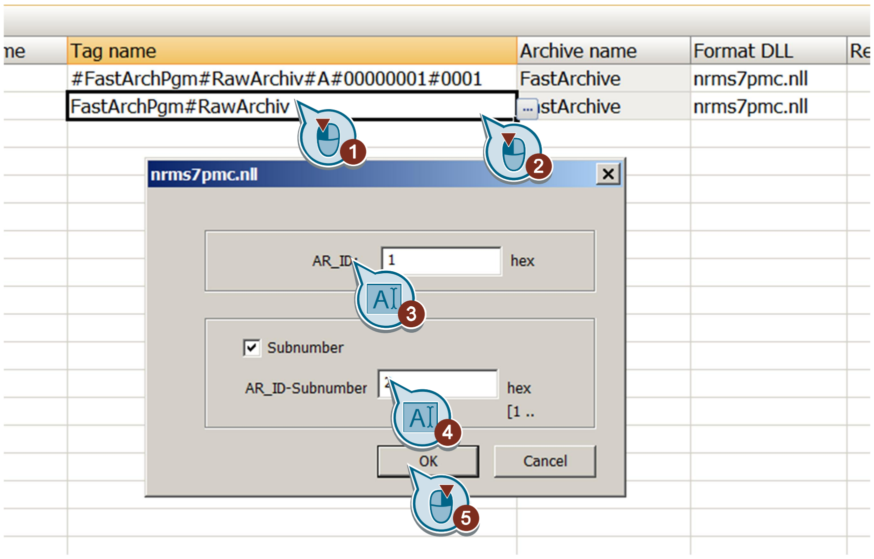


Archives [FastArchive] Process-controlled tags

	Raw data tag	Tag type	Archive tag name	Tag name	Archive name	Format DLL	Relev
1	FastArchPgm#RawArchiv	Process		#FastArchPgm#RawArchiv#A#00000001#0001	FastArchive	nrms7pmc.nll	
2	FastArchPgm#RawArchiv	Process		FastArchPgm#RawArchiv	FastArchive		
3							
4							
5							
6							

10. Select the cell in the "Tag name" column and click on the button ⋯.
11. Assign the AR_ID and the AR_ID subnumber according to your configuration in the Continuous Function Chart and click the "OK" button.

Figure 4–6

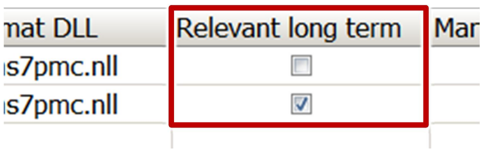


ne	Tag name	Archive name	Format DLL	Re
	#FastArchPgm#RawArchiv#A#00000001#0001	FastArchive	nrms7pmc.nll	
	FastArchPgm#RawArchiv	⋯ stArchive	nrms7pmc.nll	

Note If you have decided to use multiple process value archives, be careful to make the correct selection when creating an archive tag and the correct allocation of AR_ID and AR_ID subnumbers.

12. If your archive tags are to be stored in a long-term archive, activate the "long-term relevant" option.

Figure 4–7

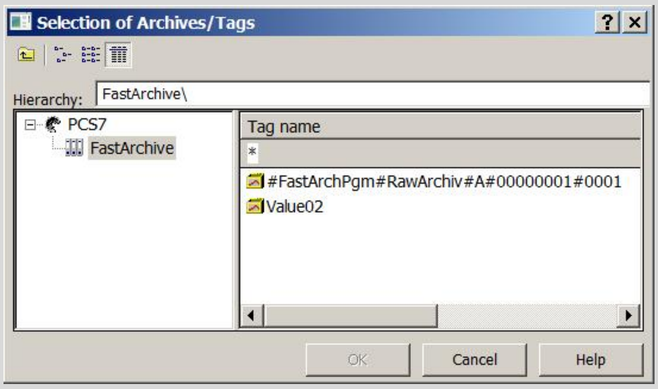


nat DLL	Relevant long term	Mar
s7pmc.nll	☐	
s7pmc.nll	☑	

Note Please note that, for example, the use of a CAS (Central Archive Server) for archiving long-term relevant archive tags can restrict the performance of the system and thus the quantity structure for fast archiving of process values.

13. (Optional) Allocate an archive tag name.

Note If you leave this field blank, it is used as an archive tag name for the tag name. This is, for example, clear in WinCC when selecting tags for a trend control.



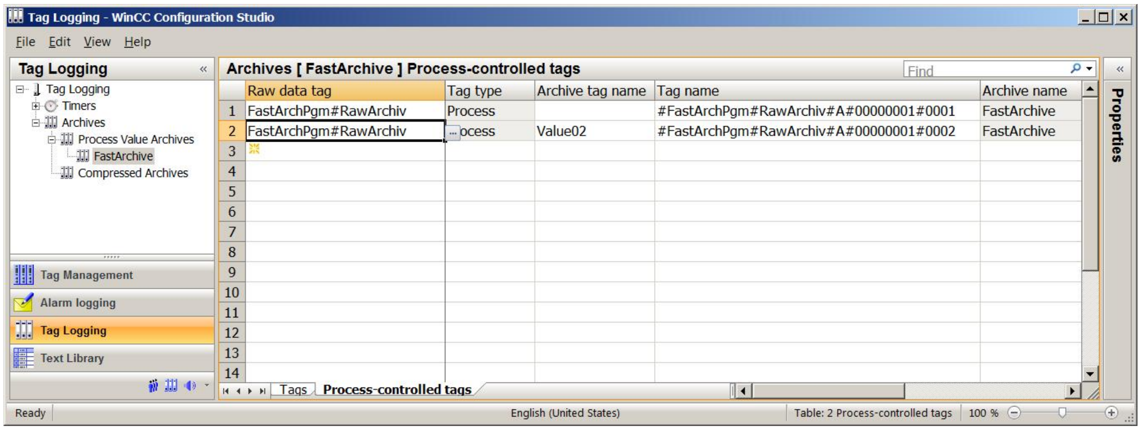
Hierarchy: FastArchive\

Tag name
*
#FastArchPgm#RawArchiv#A#00000001#0001
Value02

Example project

In the bundled Application Example, two archive tags have been created in the process value archive "FastArchive". No archive variable name has been assigned to the first one, whereas Value02" was chosen as the name for the second one.

Figure 4-8



Archives [FastArchive] Process-controlled tags

	Raw data tag	Tag type	Archive tag name	Tag name	Archive name
1	FastArchPgm#RawArchiv	Process		#FastArchPgm#RawArchiv#A#00000001#0001	FastArchive
2	FastArchPgm#RawArchiv	ocess	Value02	#FastArchPgm#RawArchiv#A#00000001#0002	FastArchive
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					

Note Make sure that a corresponding archive tag is created for every AR_ID and AR_ID subnumber that was configured in the AS program.

14. Repeat the previous steps to create additional archive tags.
15. Save and exit the WinCC Tag Logging.

4.4.3 Further configurations

The necessary configuration steps are now done.

Additional configuration steps in connection with the fast process value archiving could include the following:

- Add trend controls and table controls in WinCC pictures for the visualization and simple export of archived process values.
- Configuring the backup configuration if the long-term relevance has been configured.
- Configuring a disconnection of the fast process value archiving before the WinCC runtime is exited or an activation of the archiving after starting the WinCC runtime.
  This can increase the plausibility of the archived process values, as there is a clear division and therefore an assignment to stop and start the WinCC runtime.
  In addition, it also ensures that the buffering of process values does not fill up.

5 Example project

The example project shows the basic configuration of the fast process value archiving.

The project was created as a single-user system. The configuration of the automation hardware corresponds to a real system and has been tested with it.

As a quick introduction, the example project was prepared for use with S7-PLCSSIM.

The following descriptions refer exclusively to the use of the example project with S7-PLCSIM. If you want to use real hardware, the following changes must be also carried out:

- Adapt the access point in the SIMATIC Manager
- Adapt the hardware configuration of the PC station
- Adapt the hardware configuration of the AS
  - If you want to use a different CPU, make sure to use the function "Replace object...". If you delete the CPU and paste another, you would lose the S7 program.
- You might need to adjust the network adapter in the system properties of the WinCC channel driver.

Note

The example project was created with PCS 7 V8.1.

If you are using a newer version of PCS 7, you must update the project.

If you are using a version prior to PCS 7 V8.1, you cannot use this example project.

In this case, you can fall back to the SCL source files of the blocks ARSCAN_E and AR_MAN_E and reproduce the sample configuration from the following descriptions.

However, not every aspect of the project creation is addressed.

If in doubt, refer to the online help and documentation.

5.1 Adjusting the example project

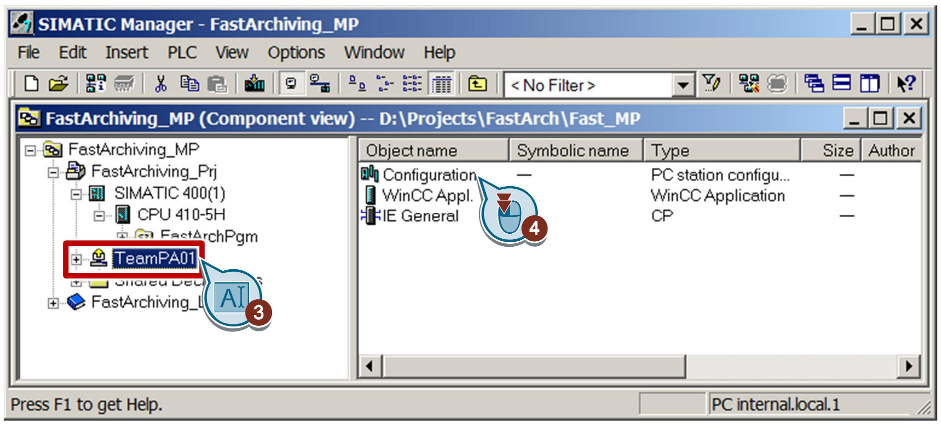
Retrieving the SIMATIC PCS 7 project

1. Download the example project and copy it to your Engineering Station.
2. Open the SIMATIC Manager, retrieve the project and open it.

Adjusting the Single Station

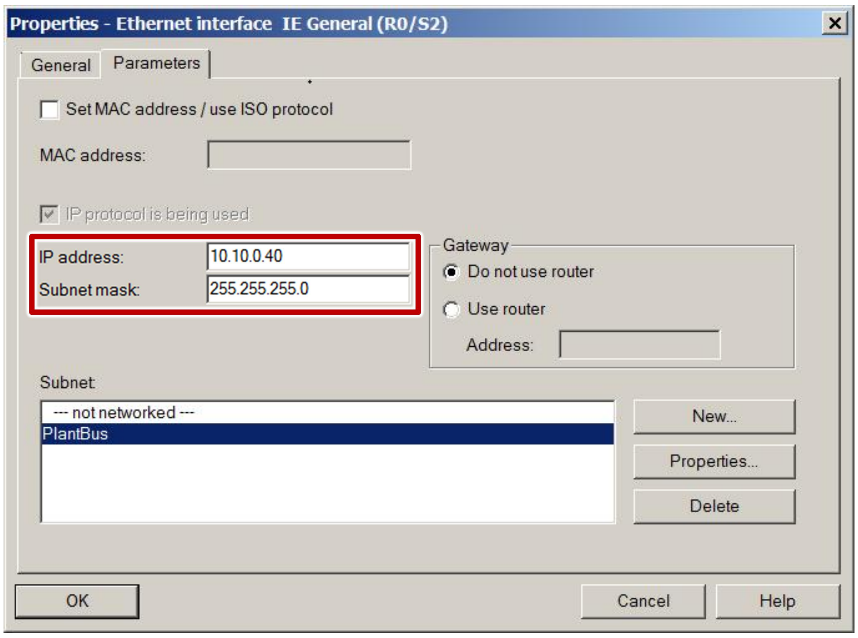
3. Match the name of the PC station to your computer name.
4. Open the hardware configuration of the PC station and match the IP address to that of your computer.

Figure 5–1



Object name	Symbolic name	Type	Size	Author
Configuration	—	PC station configu...	—	
WinCC Appl.		WinCC Application	—	
IE General		CP	—	

Figure 5–2



Set MAC address / use ISO protocol ☐

MAC address:

IP protocol is being used ☑

IP address: 10.10.0.40

Subnet mask: 255.255.255.0

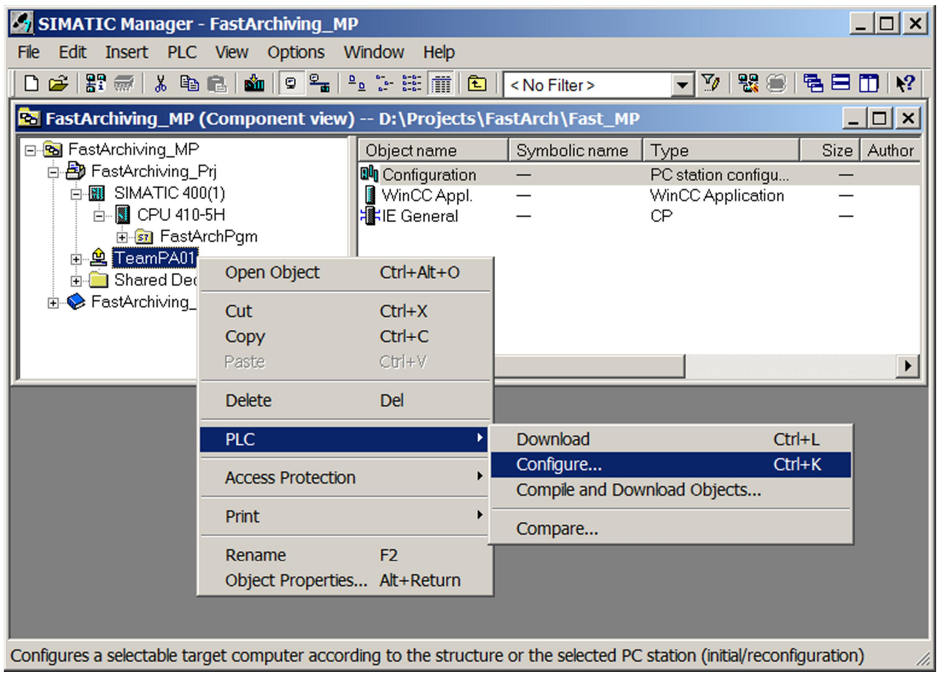
Gateway: [✓ Do not use router] Use router

Address:

Subnet:
--- not networked ---
PlantBus

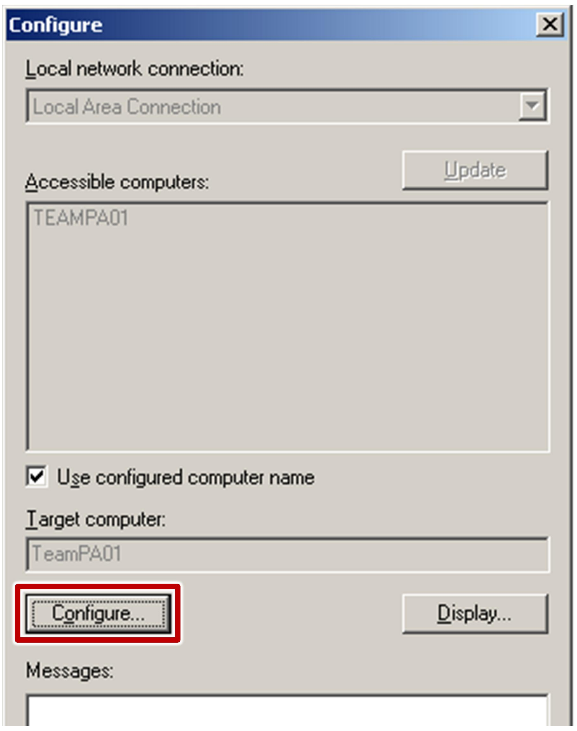
5. Save and compile the hardware configuration.
6. Open the dialog to configure the PC station.

Figure 5–3



Configures a selectable target computer according to the structure or the selected PC station (initial/reconfiguration)

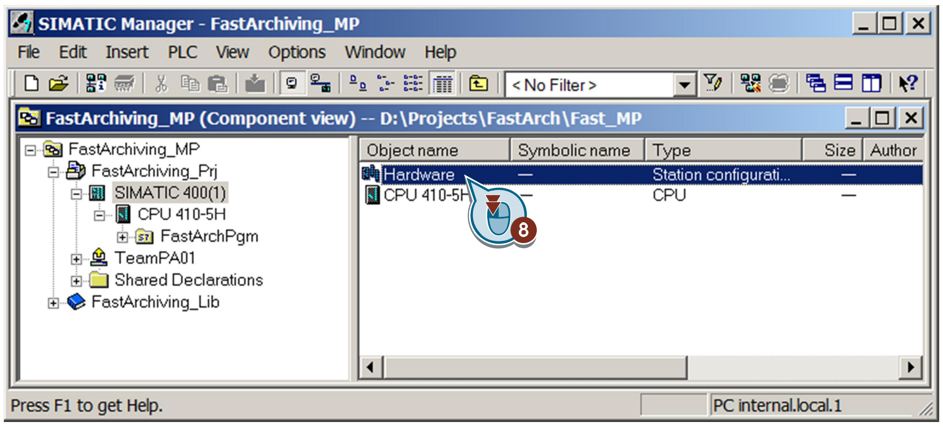
7. Configure your PC station.



Adjusting the AS

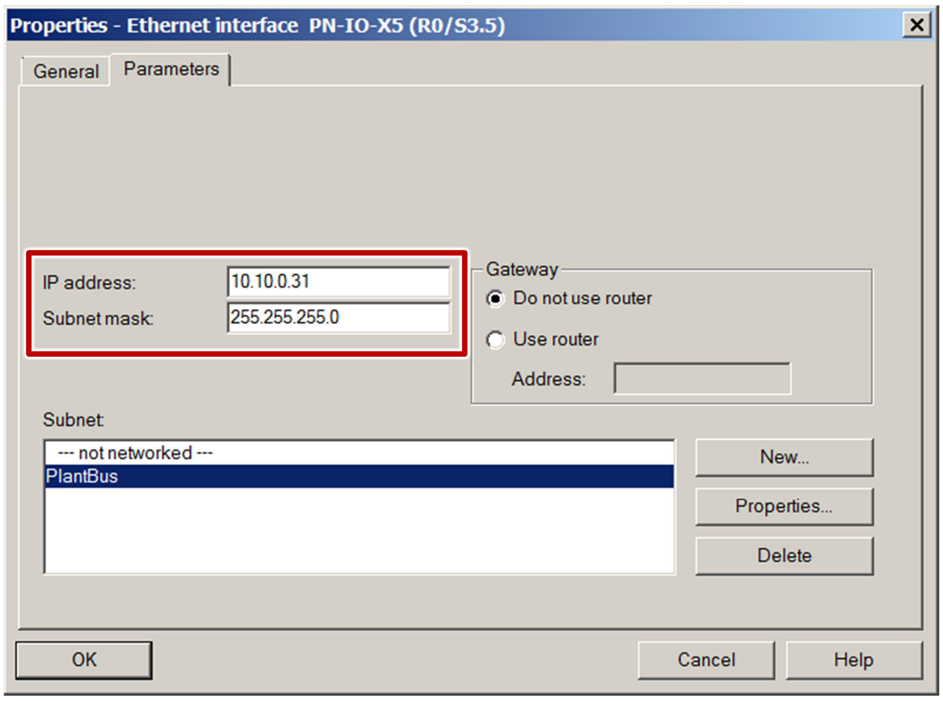
8. Open the hardware configuration of the AS and modify the IP address of the Ethernet interface of the CPU so that it connects to the same network as your PC station.

Figure 5–4



Object name	Symbolic name	Type	Size	Author
Hardware	—	Station configurati...	—	
CPU 410-5H	—	CPU	—	

Figure 5–5



IP address: 10.10.0.31
Subnet mask: 255.255.255.0

Gateway: [● Do not use router] ○ Use router
Address:

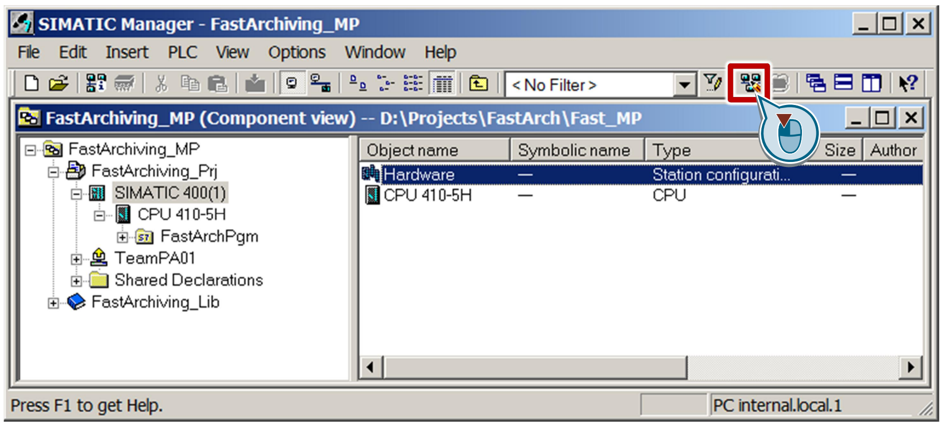
Subnet:
--- not networked ---
PlantBus

9. Save and compile the hardware configuration.

Compiling and loading the changes

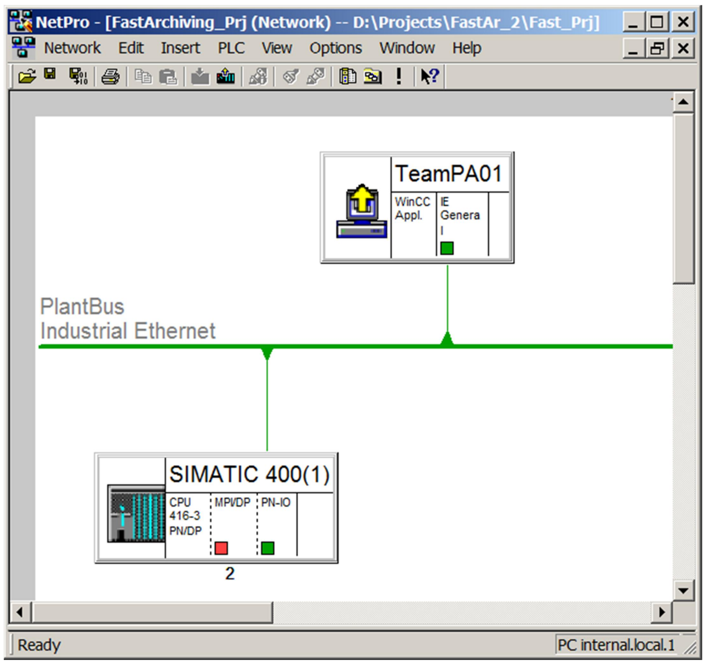
10. After the necessary adjustments have been made, they must be compiled and loaded.
Open NetPro.

Figure 5–6



Object name	Symbolic name	Type	Size	Author
Hardware	—	Station configurati...	—	
CPU 410-5H	—	CPU	—	

Figure 5–7



11. Compile the NetPro configuration with the option "Compile and check everything".
12. Load the configuration to the PC station, by highlighting it and selecting "Target System > Download to Current Project > Selected Station" and follow the dialog.
13. Open PLCSIM by clicking on "Extras > Simulate Modules".
14. Load the configuration in the AS.
15. Load the program in the AS and start PLCSIM.

Note

If you can't load the stations, check your configuration:

- Network configuration (network cards, IP addresses, firewall, switches, etc.)
- Check the configuration in the project.
- Carry out the steps for configuring the PC station once again.

Adjusting the Operator Station

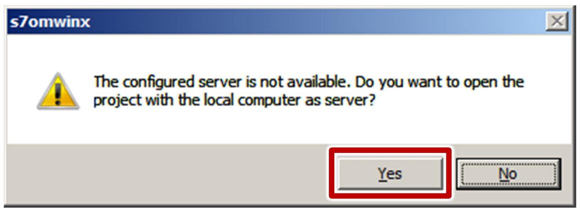
16. Open the WinCC project.

17. If your computer name is different than that of the example project, you will get the following message:

"The configured server is not available. Would you like to open the project with the local computer as the server?"

Confirm this message by clicking "Yes".

Figure 5–8



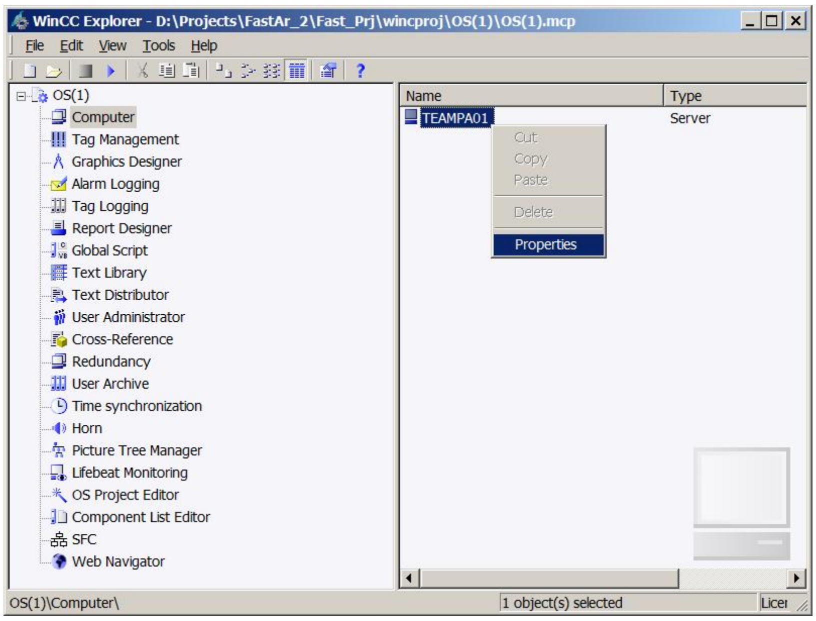
s7omwinx

The configured server is not available. Do you want to open the project with the local computer as server?

18. Adjust the computer name in the WinCC project.

For this, select the "Computer" option on the left side and open the properties dialog of the configured computer.

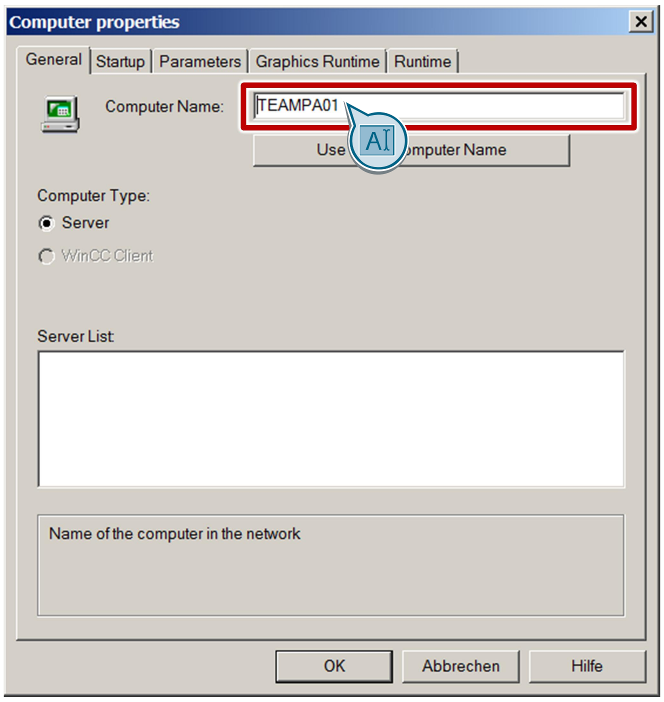
Figure 5–9



WinCC Explorer - D:\Projects\FastAr_2\Fast_Prj\wincproj\OS(1)\OS(1).mcp

19. Set the computer name.

Figure 5–10



Computer properties

General | Startup | Parameters | Graphics Runtime | Runtime

Computer Name: TEAMPA01

Use Computer Name

Computer Type:

(● Server) WinCC Client

Server List

Name of the computer in the network

OK | Abbrechen | Hilfe

20. Restart WinCC Explorer for the change to take effect.
21. Compile the OS. A change compilation is sufficient.

The necessary changes are now in place. You can now start the WinCC runtime and test the fast process value archiving.

Note If the WinCC runtime cannot connect to PLCSIM, please check your configuration.

Also check whether the correct network adapter is selected in the system parameters of the WinCC channel driver.

5.2 Description of the example project

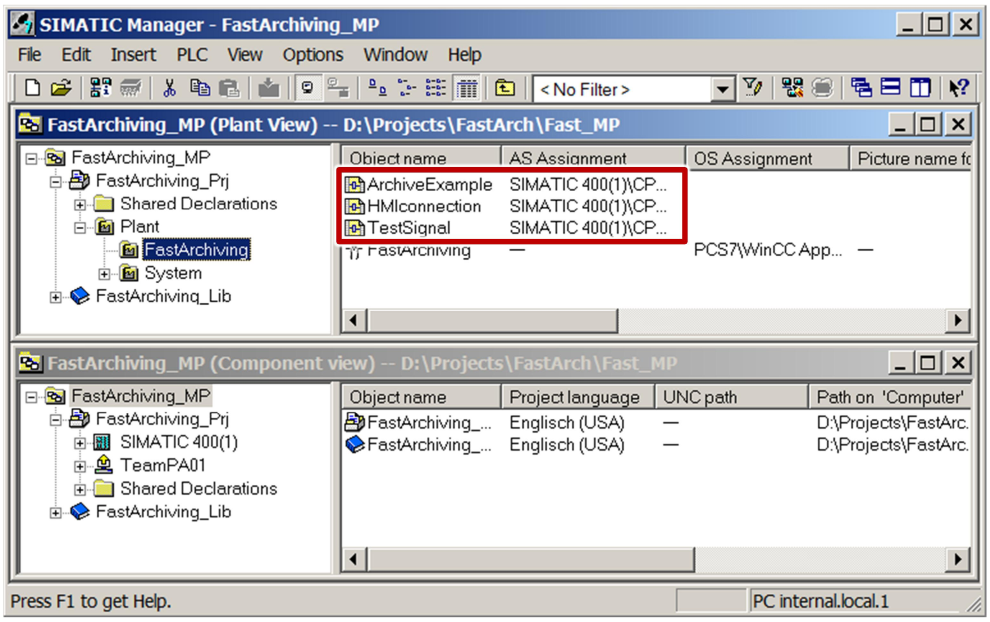
AS section

The AS project contains a three-hierarchy folder. The OS area begins at the second hierarchical level.

- Plant
  - This folder serves as a central point to establish the assignment of the lower-level folder to the AS.
- System
  - This folder contains system-relevant program parts that are not immediately required for fast process value archiving.
- FastArchiving
  - This folder contains the program for fast process value archiving and the picture for the OS project.

The following sections only discuss the Continuous Function Charts in the hierarchy folder "FastArchiving".

Figure 5–11



Object name	AS Assignment	OS Assignment	Picture name fc
ArchiveExample	SIMATIC 400(1)\CP...		
HMIconnection	SIMATIC 400(1)\CP...		
TestSignal	SIMATIC 400(1)\CP...		
FastArchiving	—	PCS7\WinCC App...	—

Object name	Project language	UNC path	Path on 'Computer'
FastArchiving_...	Englisch (USA)	—	D:\Projects\FastArc.
FastArchiving_...	Englisch (USA)	—	D:\Projects\FastArc.

Table 5–1

Continuous Function Chart name	Functions
TestSignal	In this chart, a simple sawtooth signal is generated to illustrate the way fast process value archiving functions. The signal is formed by incrementing and resetting a value. The chart is edited in OB35 (100ms cycle). By incrementing the initial value 1 by the value 1 and resetting at 100, this results in a signal with a time period of 10 seconds and an amplitude of 100. This simple signal is optimally suited for the analysis of archived process values.
HMIconnection	This chart includes three blocks of the type "OpDi01" to allow binary operation of WinCC. • "DisableTestSig" block - If the linked output of this block is logically 1, the test signal is deactivated and reset to the value 1. - Operation can be done via HMI or CFC. The input parameter "LiOp" determines whether the block is operated via HMI or CFC. • "DisableSigLink" block - The input parameter "In" enables the block "DisableTestSig" to be controlled from CFC (if it has been configured accordingly). • "EnableArchiving" block - This block enables the activation/deactivation of archiving at the blocks ARSCAN_E and AR_MAN_E. • "EnableArchLink" block - The input parameter "In" enables the block "EnableArchiving" to be controlled from CFC. • Block: "ResetSigOnStart" - If the linked output of this block is logically 1, the test signal is reset to 1 when activated (starts again). - Operation can be done via HMI or CFC. The input parameter "LiOp" determines whether the block is operated via HMI or CFC. • "RstSigOnStartLnk" block - The input parameter "In" enables the block "EnableArchiving" to be controlled from CFC.

ArchiveExample	This chart includes the example for fast process value archiving. • Two channel blocks (Pcs7AnIn) were placed. In these blocks, the test signal has been connected as a simulation value and the simulation has been activated at the channel block. • The initial structure is split into its components (process value and quality) after the channel blocks and each is connected to an ARSCAN_E block. • "Scan01" block - This block has been configured to detect the process value in an event-driven manner but at least after 5 seconds. - SUB_ID = 1 DEADBAND = 1 MIN_CYC = 0 MAX_CYC = 5 - Since the test signal can accept the values 1 - 100 and each cycle changes by the value 1, each individual value from the test signal is also archived. If the test signal is not active, the value is archived every 5 seconds. • "SCAN_02" block - This block has been configured to detect the process value cyclically. - SUB_ID = 2 DEADBAND = 0 MIN_CYC = 0.2 MAX_CYC = 5.0 - The UPD_EVT parameter has been connected to the signal to start archiving, so that even the first value in the data series is archived. - Thanks to the cyclic archiving every 200ms, every second value of the test signal is archived. • "Archive01" block - This block is connected to the two ARSCAN_E blocks and parameterized with AR_ID = 1. The other parameters were left at their default settings.

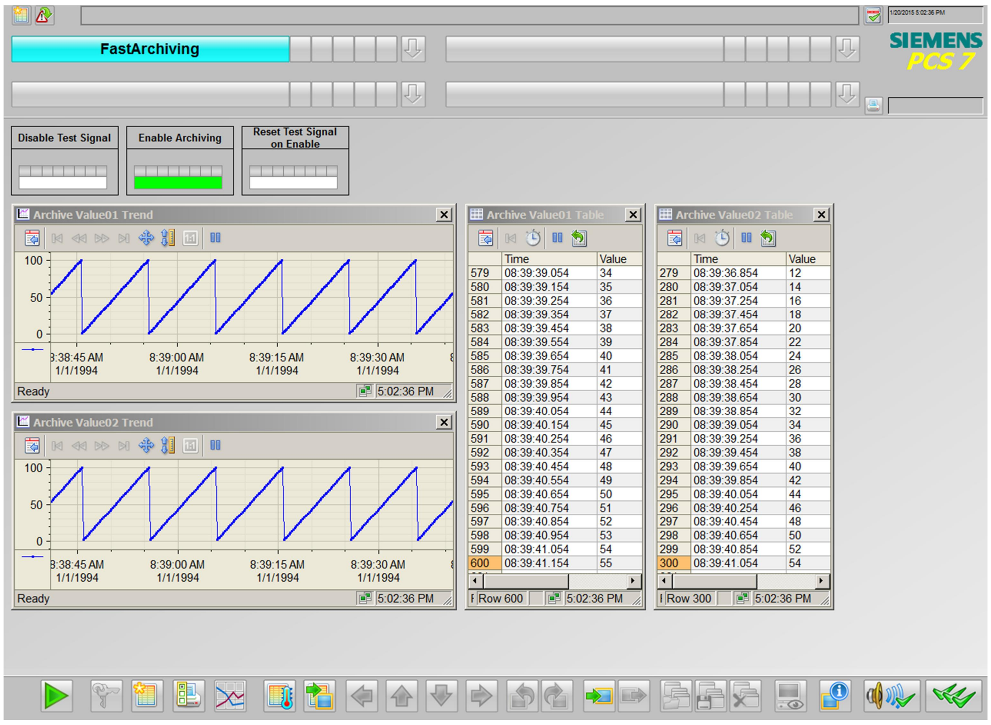
OS section

The WinCC Tag Logging has been configured as described in Chapter 0.

The picture "FastArchiving" in the hierarchy folder that carries the same name contains some elements to control the sample program, and to display the archived values and export them if required.

The individual picture elements are described in the following section.

Figure 5–12



Archive Value01 Table

	Time	Value
579	08:39:39.054	34
580	08:39:39.154	35
581	08:39:39.254	36
582	08:39:39.354	37
583	08:39:39.454	38
584	08:39:39.554	39
585	08:39:39.654	40
586	08:39:39.754	41
587	08:39:39.854	42
588	08:39:39.954	43
589	08:39:40.054	44
590	08:39:40.154	45
591	08:39:40.254	46
592	08:39:40.354	47
593	08:39:40.454	48
594	08:39:40.554	49
595	08:39:40.654	50
596	08:39:40.754	51
597	08:39:40.854	52
598	08:39:40.954	53
599	08:39:41.054	54
600	08:39:41.154	55

Archive Value02 Table

	Time	Value
279	08:39:36.854	12
280	08:39:37.054	14
281	08:39:37.254	16
282	08:39:37.454	18
283	08:39:37.654	20
284	08:39:37.854	22
285	08:39:38.054	24
286	08:39:38.254	26
287	08:39:38.454	28
288	08:39:38.654	30
289	08:39:38.854	32
290	08:39:39.054	34
291	08:39:39.254	36
292	08:39:39.454	38
293	08:39:39.654	40
294	08:39:39.854	42
295	08:39:40.054	44
296	08:39:40.254	46
297	08:39:40.454	48
298	08:39:40.654	50
299	08:39:40.854	52
300	08:39:41.054	54

Table 5–2

No.	Object type	Description
1	WinCC OnlineTrendControl	This trend shows the values of the archive tag "#FastArchPgm#RawArchiv#A#00000001#0001", which is detected by the CFC block Scan01.
2	WinCC OnlineTrendControl	This trend shows the values of the archive tag "Value02", which is detected by the CFC block Scan02.
3	WinCC OnlineTableControl	This trend shows the values of the archive tag "#FastArchPgm#RawArchiv#A#00000001#0001" The WinCC Online TableControl allows you to export the archived values to a CSV file. Please note that the exported file will only contain the values of the selected time range.
4	WinCC OnlineTableControl	This trend shows the values of the archive tag "#Value02".

5	APL OpDi01 block icon	This binary process operation allows you to deactivate the test signal. When operation is active, the test signal is reset and held at the value 1.
6	APL OpDi01 block icon	This binary process operation allows you to start / stop the archiving process.
7	APL OpDi01 block icon	When this process operation is activated, the test signal is reset to the value 1 each time you start archiving.

6 Operation of the example project

6.1 Operation via the OS runtime

The easiest way of operating the Application Example is via the OS runtime. To do this, use the configured elements in the OS area "FastArchiving" as described in the previous section.

6.2 Operation via CFC

Besides operating via the OS runtime one can also operate all configured functions directly from the CFC.

- Start / stop archiving
  - Open the CFC chart "HMIconnection".
  - Activate the test mode in CFC
  - Set the input parameter
    "LiOp" = 1 at the block "EnableArchiving".
    Now the operation of the block is done via the inputs "SetLi" and "RstLi".
  - To start archiving, set the input parameter
    "In" = 1 at the block "EnableArchLink"
  - To stop archiving, set the input parameter
    "In" = 0 at the block "EnableArchLink"
- Deactivate test signal
  - To deactivate the test signal, proceed in accordance with the above description.
- Reset the test signal when starting the archiving process
  - To automatically reset the test signal when starting the archiving process, proceed in accordance with the above description.

6.3 Resetting the archive

When testing the fast process value archiving, there might be large amounts of data that are not necessarily needed and which take up space.

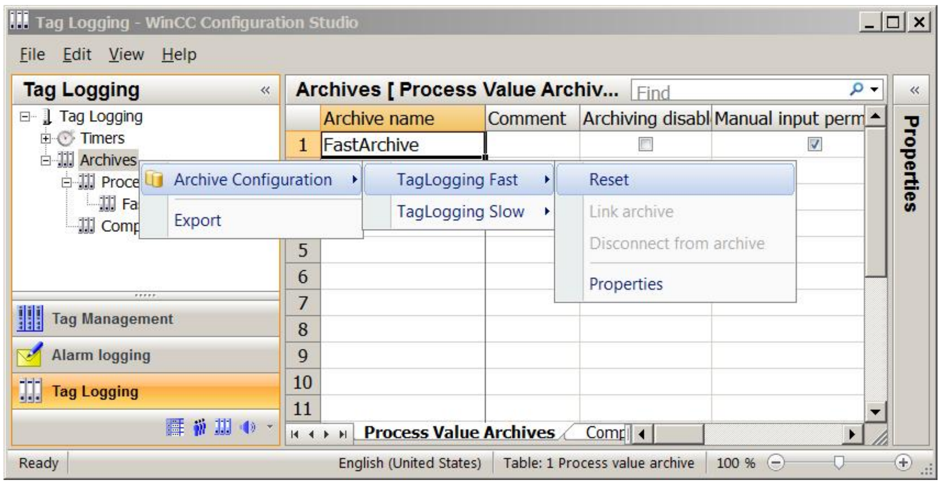
If you want to delete this data, you can reset the WinCC archive "Tag Logging Fast".

CAUTION	**Note that when you reset this archive, all data in it will be deleted from the project.**

To reset the archive, proceed as follows:

1. Open the WinCC Tag Logging in WinCC Explorer.
2. Right click on "Archives" to open the associated context menu.
   Navigate to "Archive Configuration> TagLoggingFast" and click on "Reset".

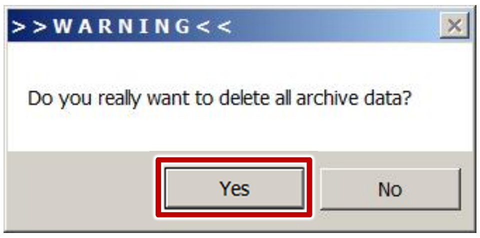
Figure 6–1



	Archive name	Comment	Archiving disabl	Manual input perm
1	FastArchive		☐	☑

3. If you are sure that you want all data in this archive to be deleted, confirm the following dialog with "Yes".

Figure 6–2


>>WARNING<<

Do you really want to delete all archive data?

4. Save and close the WinCC Tag Logging.

7 Literature

Table 7–1

	Subject area	Title
\1\	Siemens Industry Online Support	http://support.automation.siemens.com
\2\	Integration of S7-300 Package Units and Operator Panels in SIMATIC PCS 7 with PCS 7 Industry Library	http://support.automation.siemens.com/WW/view/en/23780904

8 History

Table 8–1

Version	Date	Change
V1.0	12/2006	First edition
V1.1	03/2009	• Changing the time calculation from TIME to REAL • Extend the following block interfaces / functions - SAMPLE_TIME / cycle time OB - QFIFODEG / storage level - QNUM_TRANS / archived process values - QNUMSEND / data blocks sent - QNUMCONNECT / connected ARSCAN_E
V2.0	11/2013	• Complete revision of the document - Restructuring of the chapter - Validity for PCS 7 V8.0 Update 1 - Project guidelines for quantity structures / limitations • New example project based on PCS 7 V8.0 Update 1 • SCL source files from version 1.2 are unofficial / released on a project-specific basis • SCL source file update to version 1.3 (for a complete list of changes, please check in the SCL source files) - **CAUTION Interface change** - Restructuring of the version history - Revision of comments - Streamlining of internal data management - Improvement of startup characteristics
V2.1	02/2015	• Documentation update to SIMATIC PCS 7 V8.1 • Example project update