



Rapid archiving of process data in PCS 7 with AR_SEND

SIMATIC PCS 7 or STEP 7 with WinCC

<https://support.industry.siemens.com/cs/ww/en/view/23780904>

Siemens
Industry
Online
Support



Legal Information

Use of the application examples

The application examples illustrate the solution of automation tasks in the interaction of several components in the form of text, graphics and/or software modules. The application examples are a cost-free service provided by Siemens AG and/or a subsidiary company of Siemens AG ("Siemens"). They are not binding and do not claim to be complete regarding completeness and functionality concerning configuration and equipment. The Application Examples do not represent customer-specific solutions; they are intended to provide help for typical applications instead.

You yourself are responsible for the proper and safe operation of the products within the applicable regulations and you must check the function of the respective example application and adapt it individually to your system.

Siemens grants you the non-exclusive, non-sublicensable and non-transferable right to make use of the application examples through specialist, instructed personnel. Any and all modifications to the application examples shall be carried out under your responsibility. Transfer to third parties or duplication of the application examples or extracts thereof is only permitted in combination with your own products. The application examples are not necessarily subject to the usual tests and quality checks of a product that is subject to a charge, may contain functional and performance defects and may be subject to errors. It is your responsibility to structure the use in such a way that any malfunctions do not lead to material damage or personal injury.

Disclaimer of liability

Siemens excludes its liability, irrespective of the legal grounds, in particular for the usability, availability, completeness and faultlessness of the application examples, as well as the associated notes, project planning and performance data and any damage caused thereby.

This does not apply if Siemens is compulsorily liable, e.g. under product liability laws, in cases of malicious intent, gross negligence, culpable injury to life, body or health, failure to comply with an assumed guarantee, fraudulent concealment of a defect or culpable breach of essential contractual obligations. The claims for compensation 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. You shall exempt Siemens from any third-party claims that may exist or arise in this connection, unless Siemens is compulsorily liable by law.

By using the application examples, you agree that Siemens cannot be made liable for any possible damage beyond the liability clause mentioned.

Additional information

We reserve the right to make changes to these Application Examples at any time without prior notice. If there are any discrepancies between the recommendations provided in these Application Examples and other Siemens publications – e.g. Catalogs – the contents of the other documents have priority.

In addition to this, the Siemens terms and conditions of use apply (<https://support.industry.siemens.com>).

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions only form one element of such a concept.

Customer is responsible to prevent unauthorized access to its plants, systems, machines and networks. Systems, machines and components should only be connected to the enterprise network or the Internet where necessary and with appropriate security measures (e.g., use of firewalls and network segmentation) in place.

Additionally, Siemens' guidance on appropriate security measures must be taken into account. For more information about Industrial Security, please visit: <https://www.siemens.com/industrialsecurity>.

Siemens' products and solutions undergo continuous development to make them more secure.

Siemens strongly recommends to apply product updates as soon as available and to always use the latest product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under: <http://www.siemens.com/industrialsecurity>.

Table of Contents

Legal Information	2
1 The task	4
1.1 Overview.....	4
1.2 Requirements	4
2 Solution.....	5
2.1 Overview.....	5
2.2 Hardware and software components	7
2.2.1 Validity	7
2.2.2 Components used	7
3 Basics	8
3.1 Operating principle of the Application Example	8
3.2 Functionality of the block AR_SEND.....	9
3.3 Functionality of the ARSCAN_E block	11
3.3.1 Overview / Function.....	11
3.3.2 Cyclic process value detection	12
3.3.3 Event-driven process value detection	12
3.3.4 Saving process values	13
3.3.5 Responding to overload or connection failure.....	13
3.3.6 Time-of-day synchronization	13
3.3.7 Block I/Os	15
3.4 Functionality of the AR_MAN_E block	18
3.4.1 Overview / Function.....	18
3.4.2 AR_SEND format used	19
3.4.3 Responding to overload or connection failure.....	19
3.4.4 Block I/Os	20
4 Configuration	22
4.1 Preliminary remarks	22
4.2 Important points before configuring.....	22
4.3 Configuring in SIMATIC Manager	24
4.3.1 Integrating module ARSCAN_E and AR_MAN_E in the PCS 7 project.....	24
4.3.2 Configuring AR_MAN_E.....	25
4.3.3 Configuring ARSCAN_E.....	26
4.4 Configuration in WinCC.....	27
4.4.1 Creating process value.....	27
4.4.2 Creating archive tags	28
4.4.3 Further configurations	30
5 Sample project	31
5.1 Adjusting the example project	32
5.2 Description of the example project.....	39
6 Operation of the example project	43
6.1 Operation via the OS runtime.....	43
6.2 Operation via CFC.....	43
6.3 Resetting the archive.....	44
7 Appendix	45
7.1 Service and Support.....	45
7.2 Links and Literature	46
7.3 Change documentation	46

1 The 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.

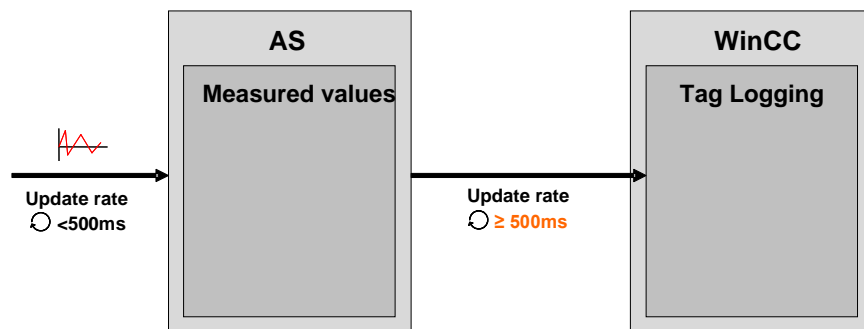
To archive data faster than 500ms, one must find another solution.

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.

- Archiving rates faster 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 module, we make two function modules available to you (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 saved temporarily 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.

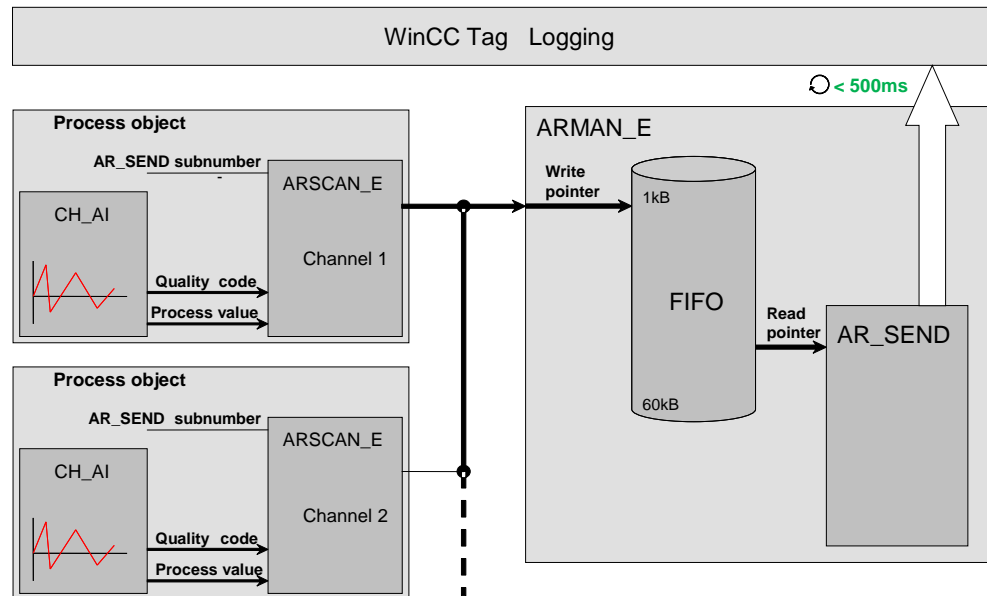
- The block ARSCAN_E is used for detecting and storing a single process value of the data type REAL or DWORD.
An ARSCAN_E block is used for each process value.
There are two variants for acquiring the process values.
 - Cyclical acquisition of process data is used to archive process values in an equidistant 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.

With time-of-day synchronization, several ARSCAN_E modules receive the same CPU time stamp.

- The block AR_MAN_E includes a FIFO memory for process values and manages the bundled sending of the data to WinCC with the system function block AR_SEND.
It is possible to connect to an AR_MAN_E block (one ARSCAN_E per process value), multiple ARSCAN_E blocks which 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)

Benefits

- 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

Basic knowledge of automation systems and SIMATIC PCS 7 are a prerequisite. This application example has been created with SIMATIC PCS 7 V9.0 (SCL V5.6.0.0).

2.2 Hardware and software components

2.2.1 Validity

Valid for SIMATIC PCS 7 V9.0

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	6ES7400-1JA11-0AA1	or equivalent
PS 407 10A power supply	1	6ES7407-0KA02-0AA0	or equivalent
Controller SIMATIC S7-410-5H	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 V9.0	1	6ES7651-5AA28-0YA0	ES Single station V9.0
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_V90.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: <ul style="list-style-type: none"> AR_MAN_E ARSCAN_E

3 Basics

3.1 Operating principle of the Application Example

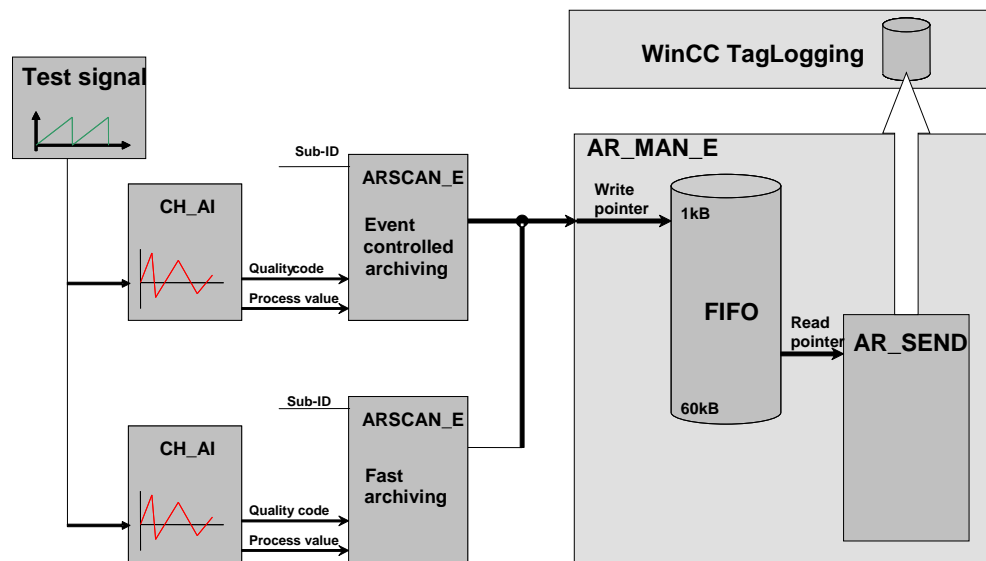
Two ARSCAN_E blocks acquire a test signal.

The first block saves the process data event-controlled, the second cyclically.

The AR_MAN_E block 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

The resource limitations when using the S7 functions AR_SEND and BSEND/BRCV for communication with S7-400 are to be taken into account. This means that the maximum data volume that can be sent simultaneously using AR_SEND and/or BSEND/BRCV functions from AS to WinCC is limited to a maximum of 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

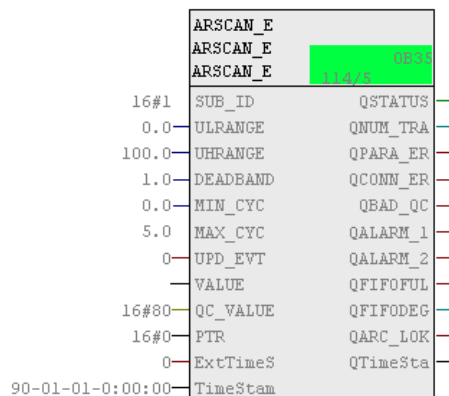
For current information about the various CPUs and the AR_SEND instances, please refer to the manual "SIMATIC S7-400 Automation System S7-400 CPU data".

3.3 Functionality of the ARSCAN_E block

3.3.1 Overview / Function

This chapter describes the ARSCAN_E block in detail.

Figure 3-2



The ARSCAN_E block 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
- Time-of-day synchronization

3.3.2 Cyclic process value detection

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

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

Note

It is important that $\text{MAX_CYC} > \text{MIN_CYC}$, as otherwise an error will be generated.

Output (of the ARSCAN_E block) 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).

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

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 time stamp 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. To prevent an overflow of the integer variables, the counter is automatically reset to 0 after 10,000 storage operations.

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 Responding to overload or connection failure](#).

3.3.6 Time-of-day synchronization

The time-of-day synchronization always makes sense if several variables are to be archived together in a cycle with the same time stamp.

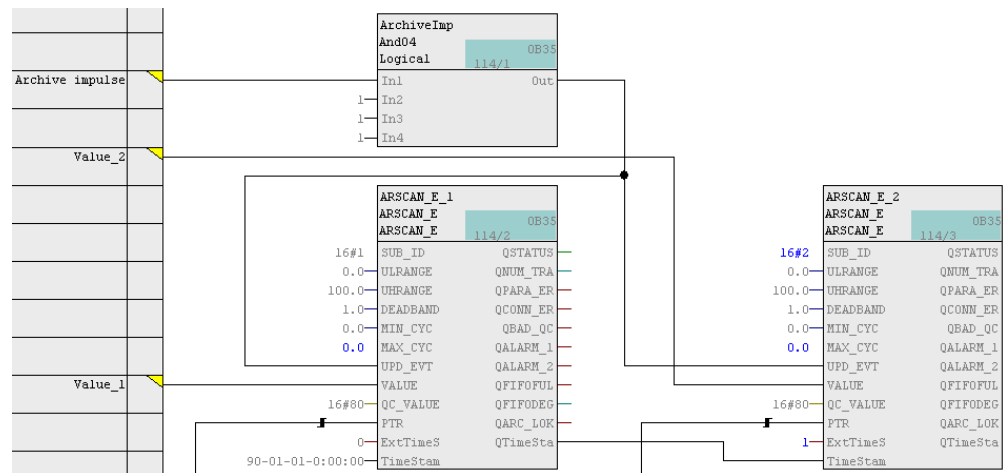
If the point in time for archiving has to be the same for all relevant blocks, an implementation variant is event-controlled process value recording with the following parameterization of ARSCAN_E module:

- Input parameters:
 - RUNUPCYC = 0
 - MIN_CYC = 0.0
 - MAX_CYC = 0.0
 - UPD_EVT = archiving impulse (edge change from 0 => 1)

An ARSCAN_E module is required for each process value. The first block called up uses the CPU time stamp and transfers it to all the following blocks.

The time stamp is forwarded via the QTimeStamp output to the TimeStampIn input. In addition, the received time stamp must be activated via the bit ExtTimeStampMode = 1. A possible wiring and parameterization can be found in Figure 3-3.

Figure 3-3

**Note**

To prevent an outdated time stamp, the first ARSCAN_E block must be called in the sequence **before** all subsequent blocks. In addition, all the blocks in the same OB cycle must be called up.

3.3.7 Block I/Os

Input parameters

Table 3-2

Parameters	Data type	Description
EN (hidden)	BOOL	Activates block processing <i>Standard value: 1</i>
ENABLE (hidden)	BOOL	Activates the detection of the process values. <i>Standard value: 1</i>
RUNUPCYC (hidden)	INT	Number of block cycles before starting. <i>Standard value: 5</i>
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. <i>Permissible values: 1 - 4095</i>
ULRANGE	REAL	Lower limit of the measuring range (only relevant for process values of the data type REAL). <i>Standard value: 0.0</i>
UHRANGE	REAL	Upper limit of the measuring range (only relevant for process values of the data type REAL). <i>Standard value: 100.0</i>
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). <i>Standard value: 0.5</i> Remark: The percentage refers to the parameters ULRANGE and UHRANGE
MIN_CYC	REAL	Minimum detection time (in seconds). <i>Standard value: 0.0</i>
MAX_CYC	REAL	Maximum detection time (in seconds) of the process value (archived, even if change < DEADBAND). <i>Standard value: 5.0</i> Note: MAX_CYC > MIN_CYC otherwise an error is generated -> Output QPARA_ER = TRUE

Parameters	Data type	Description
UPD_EVT	BOOL	Edge triggering of an archiving process.
UPD_COND (hidden)	BOOL	Condition triggering of an archiving process.
VALUE	ANY	Process value to be archived. Only values of data type REAL and DWORD are allowed.
QC_VALUE	BYTE	Quality code of the process value (must be interconnected or have its parameter assigned with 16#80).
PTR	STRUCT	Interconnection to AR_MAN_E block. Provides information for ARSCAN_E and a pointer to the FIFO memory.
ExtTimeStamp Mode	BOOL	With value 1, the time stamp from TimeStampIn input is adopted. With value 0, the time stamp from TimeStampIn input is ignored. <i>Standard value: 0</i>
TimeStampIn	DATE_AND_TIME	Input for time stamping. If required, is interconnected with QTimeStamp output. <i>Standard value: 1990-01-01-0:00:00</i>

Output parameters

Table 3-3

Parameters	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. When 10,000 is reached the value is reset to 0.
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 AR_MAN_E block is full. It is not possible to archive the process value.
QFIFODEG	INT	FIFO memory allocation in %
QARC_LOK	BOOL	Archiving at block AR_MAN_E is deactivated.
QTimeStamp	DATE_AND_TIME	Provides the time stamp for subsequent blocks. If required, is interconnected with TimeStampIn input. <i>Standard value: 1990-01-01-0:00:00</i>

Allocation of the status word QSTATUS

Table 3-4

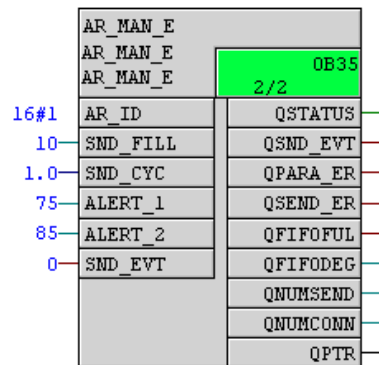
bit	Byte	Description
0	Byte 0	QC = 16#00
1		PTR invalid or not connected
2		VALUE not connected
3		VALUE has a wrong data type
4		Security Level 1 reached
5		Security Level 2 reached
6		FIFO memory allocation is too high (FIFO full)
7		Archiving lock of AR_MAN_E active
8	Byte 1	AR_ID Subnumber < 1
9		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 AR_MAN_E block

3.4.1 Overview / Function

This chapter describes the block AR_MAN_E in detail.

Figure 3-4



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. To prevent an overflow of the integer variables, the counter is automatically reset to 0 after 10,000 storage operations.

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 AR_SEND format used

The AR_SEND format 9 (Header Type = 9) is used for the transfer. This format allows the transfer 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 3-5

22 Byte	Header type = 9			
	Year		Month	
	Day		Hour	
	Minute		Second	
	1/10s	1/100s	1/1000s	Day of the week
	Cycle = 0			
	AR_ID subnumber			
	Data type of the process data			
	Number of process values			
4 bytes	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, two 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 3-6

Parameters	Data type	Description
EN (hidden)	BOOL	Activates block processing. <i>Standard value: 1</i>
ENABLE (hidden)	BOOL	Activates the detection of the process values. <i>Standard value: 1</i>
RUNUPCYC (hidden)	INT	Number of block cycles before starting. <i>Standard value: 5</i>
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. <i>Standard value: 10</i>
SND_CYC	REAL	Transmission cycle After this time, the send operation is initiated with the available data. <i>Standard value: 1.0</i>
ALERT_1	INT	Allocation of FIFO memory required in % to activate Security Level 1. <i>Standard value: 75</i>
ALERT_2	INT	Allocation of FIFO memory required in % to activate Security Level 2. <i>Standard value: 85</i>
SND_EVT	BOOL	Edge triggering of a sending process.

Output parameters

Table 3-7

Parameters	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.
QFIFODEG	BOOL	FIFO memory allocation in %
QNUMSEND	INT	Number of data blocks transmitted. When 10,000 is reached the value is reset to 0.
QNUMCONNECT	INT	Number of connected ARSCAN_E instances
QPTR	STRUCT	Interconnection to ARSCAN_E Provides 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 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. Chapters [5 Sample 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.
- Observe the maximum usable number of AR_SEND instances per CPU, as described in Chapter [3.2 Functionality of the block AR_SEND](#) (an AR_MAN_E block instantiates an AR_SEND block).
- Observe the limits of the communication to WinCC, as described in Chapter [3.2 Functionality of the block AR_SEND](#).
If it is likely that you will come close to these limits, study the topic in detail and calculate your quantity structure carefully.
- Test your quantity structure in advance. For example, with a project like the one shown in 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 blocks

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

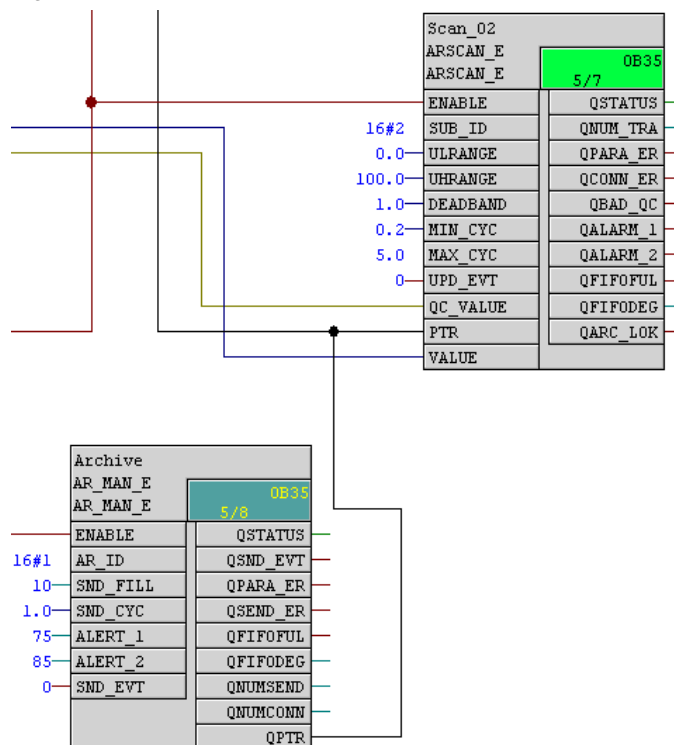
To create the blocks from the SCL source files, proceed as follows:

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. For each AR_ID that you want to use, place an AR_MAN_E module.
It is up to you whether you want to archive all your process values with the same AR_ID or divide them among 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.

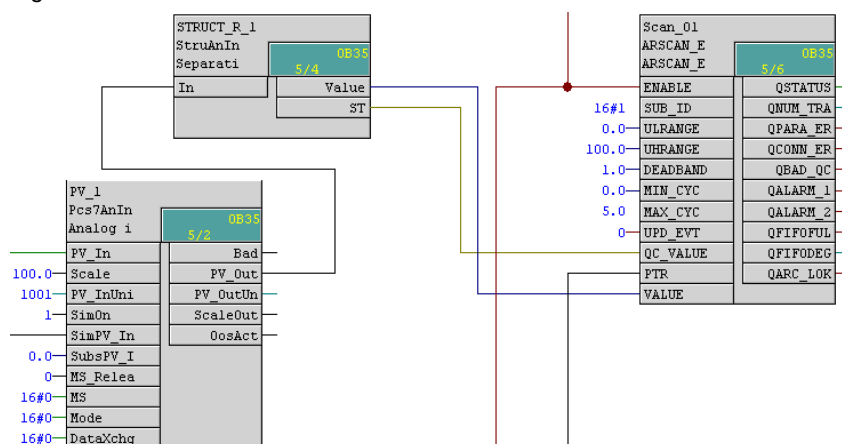
Figure 4-1



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 to 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 are in 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](#)).

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

4.4 Configuration 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.

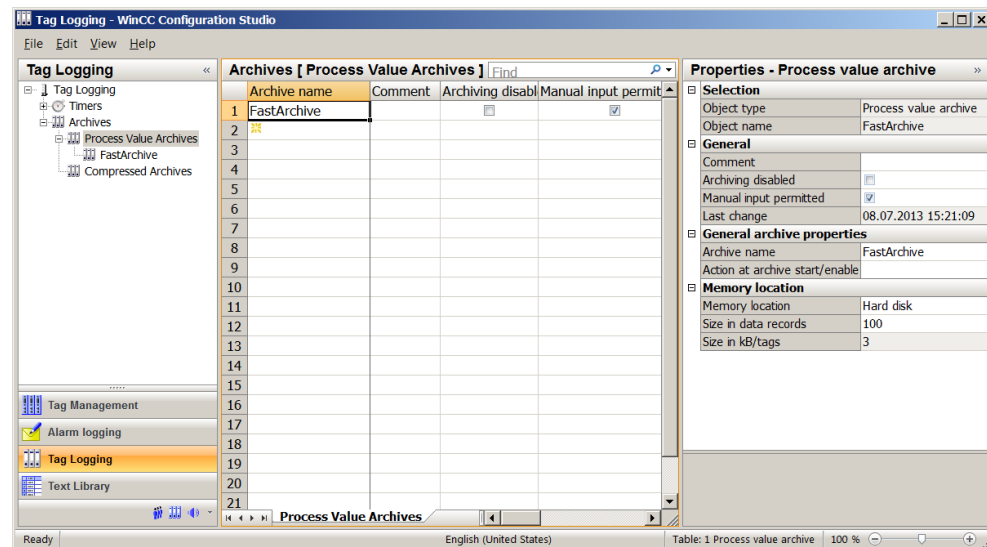
4.4.1 Creating process value

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.

Sample project

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

Figure 4-3



4.4.2 Creating 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.


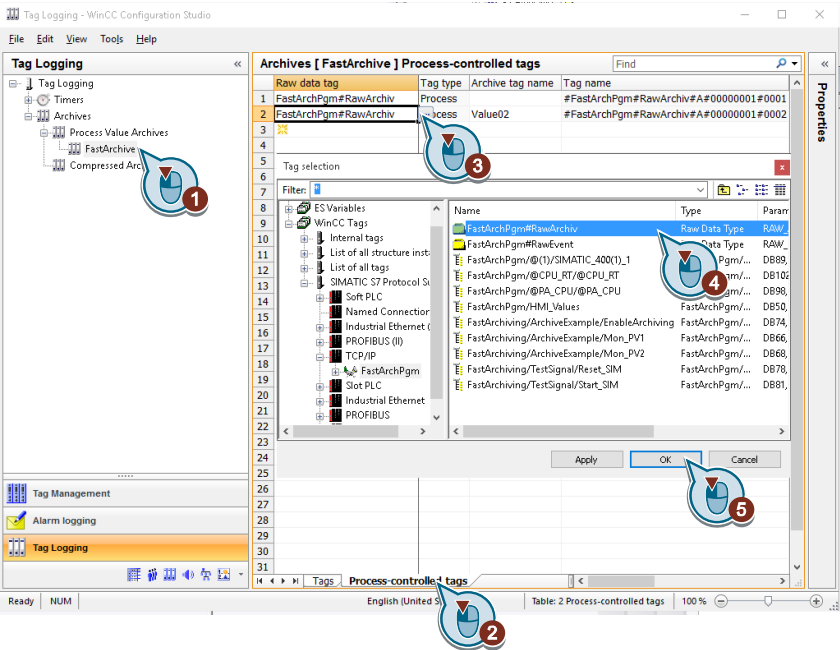
- 1. Create a new archive tag.
Click on the archive (Example: "FastArchive") and then on the "Process Controlled Tags" tab.
- 2. Click inside the field marked with the asterisk in the column "Raw data tags" and click on the button .
- 3. In your S7 program, select the raw data type for archives "[Program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

Figure 4-4



- 4. For the process tag to be archived, an appropriate DLL normalization format must be selected.

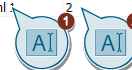
Figure 4-5

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	SSSTD.NLL		
3							
4							
5							
6							

- Assign the AR_ID and the AR_ID subnumber according to your configuration in the Continuous Function Chart and click on the "OK" button.

Figure 4-6

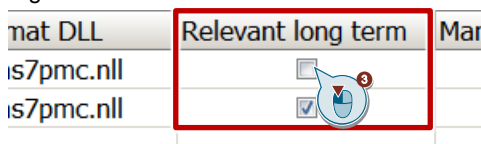
Archives [FastArchive] Process-controlled tags					
Raw data tag	Tag name	Archive name	Format DLL	Block ID	Sub-number
1 FastArchPgm#RawArchiv	#FastArchPgm#RawArchiv#A#00000001#0001	FastArchive	nrms7pmc.nl 1	1	
2 FastArchPgm#RawArchiv	#FastArchPgm#RawArchiv#A#00000001#0002	FastArchive	nrms7pmc.nl 1	2	

**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.

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

Figure 4-7

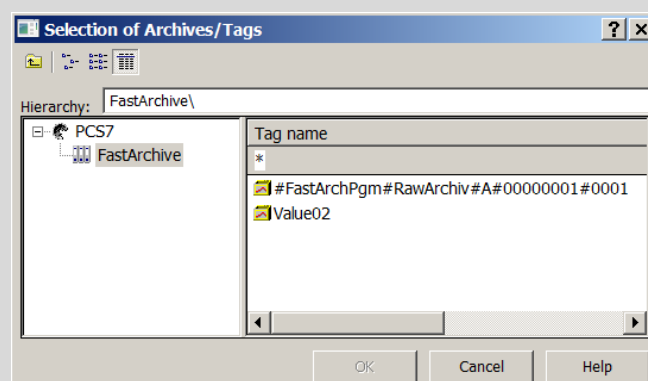
**Note**

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.

- (Optional) Allocate an archive tag name.

Note

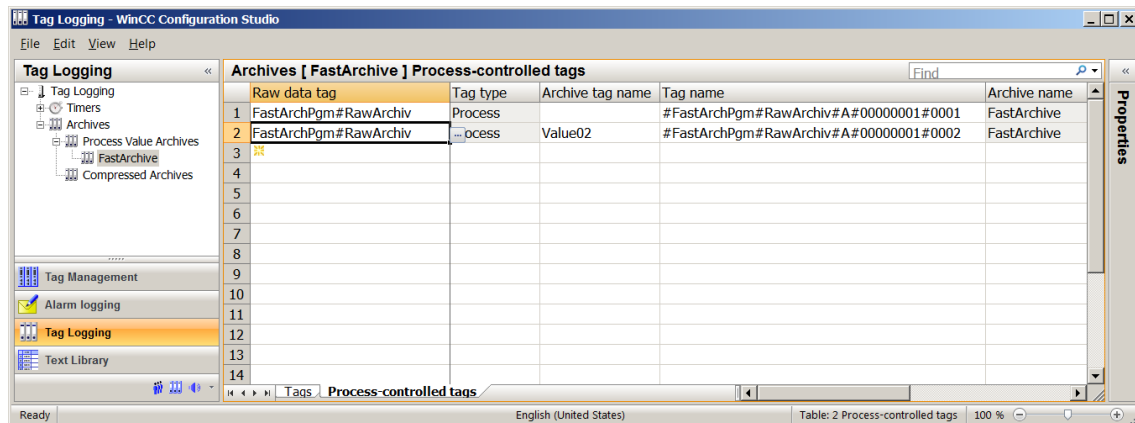
If you leave this field blank, it is used as an archive tag name for the tag name. This is, for example, clear in WinCC when selecting tags for a trend control.



Sample 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



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.

8. Repeat the previous steps to create additional archive tags.
9. 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 Sample 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-PLC SIM.

The following descriptions refer exclusively to the use of the example project with S7-PLC SIM. 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 V9.0.

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 V9.0, 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. Copy file "23780904_FastArch_PROJ_PCS7V90.zip" to any folder on the configuration PC and then open SIMATIC Manager.
2. Click on "File > Retrieve" in the menu bar and select the file "23780904_FastArch_PROJ_PCS7V90.zip". Then click on "Open" to confirm.
3. Choose the folder where you want to save the project and click on "OK" to confirm.
The project is retrieved.
4. Confirm the "Retrieve" dialog by clicking on "OK" and then click on "Yes" in the dialog to open the project.

Adjusting the Single Station

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

Figure 5-1

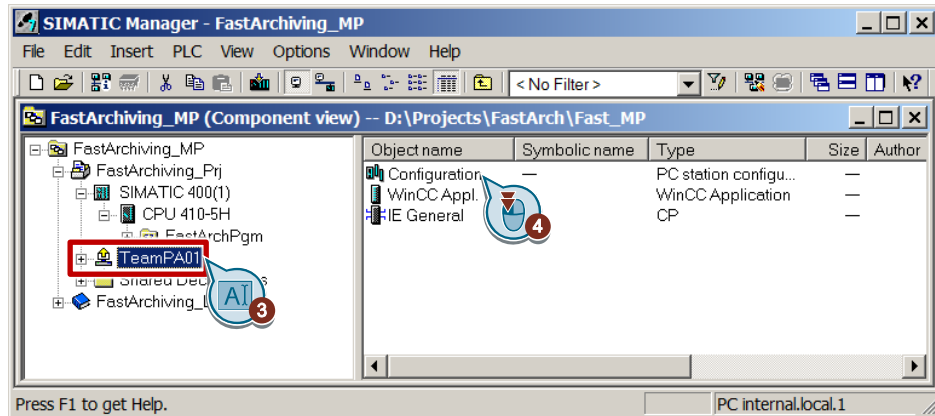
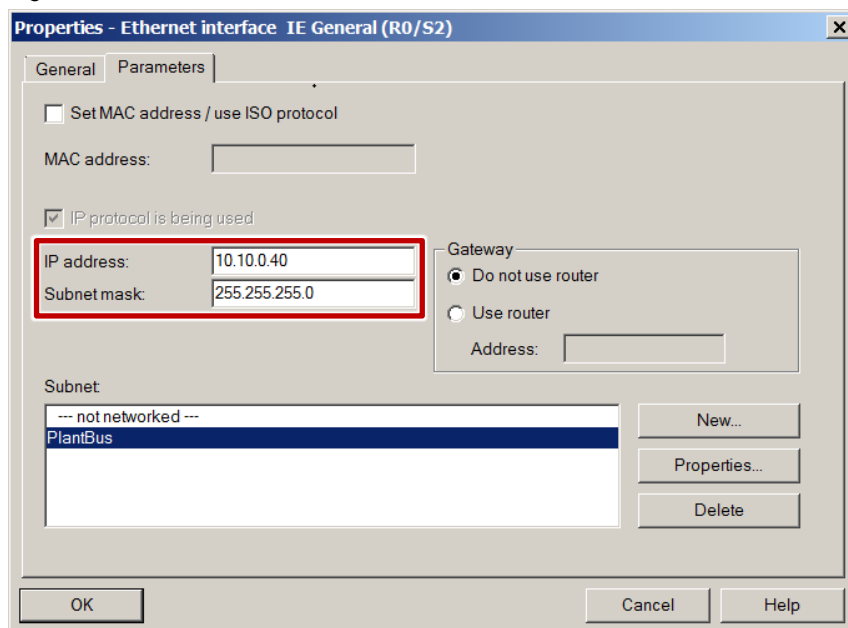


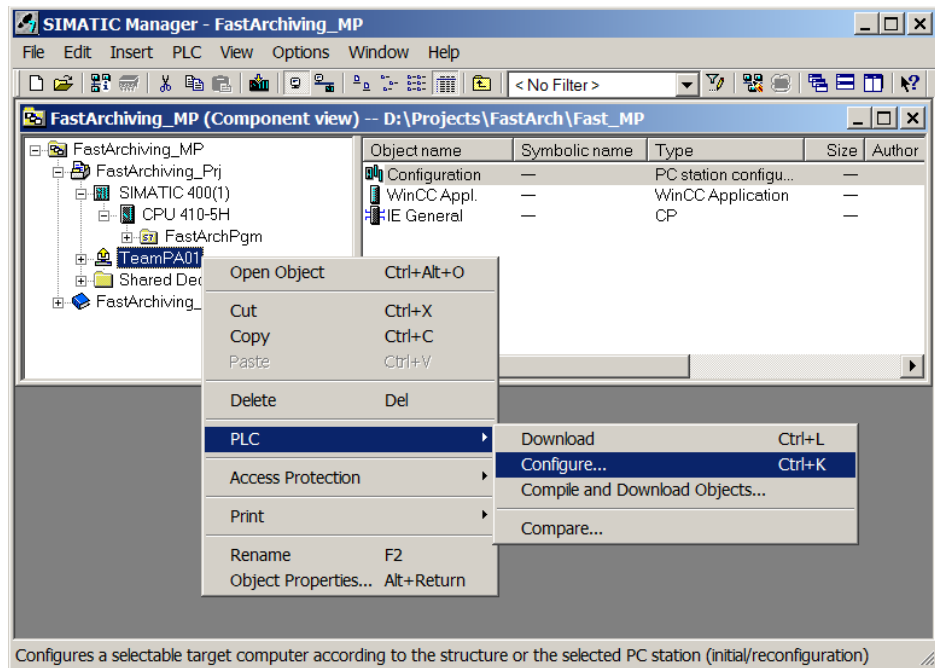
Figure 5-2



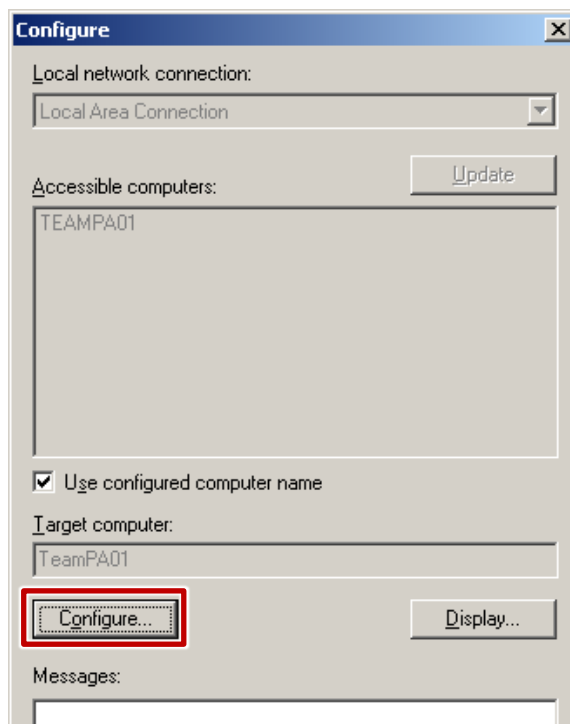
3. Save and compile the hardware configuration.

4. Open the dialog to configure the PC station.

Figure 5-3



5. Configure your PC station.



Adjusting the AS

1. 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

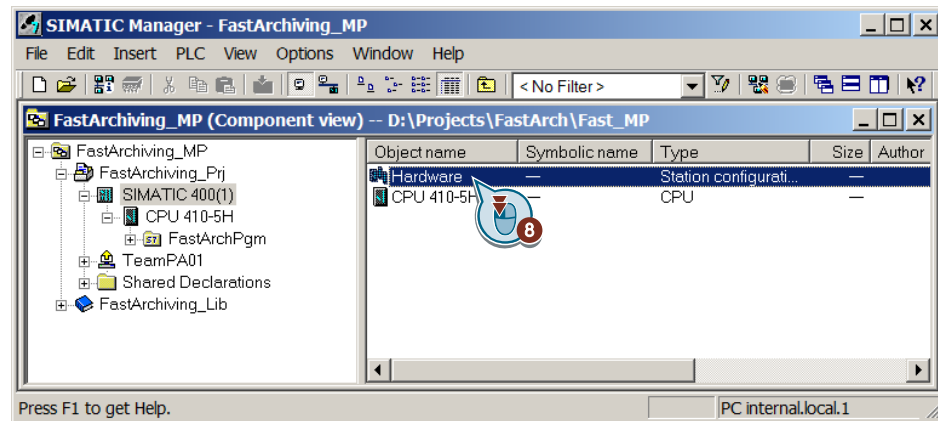
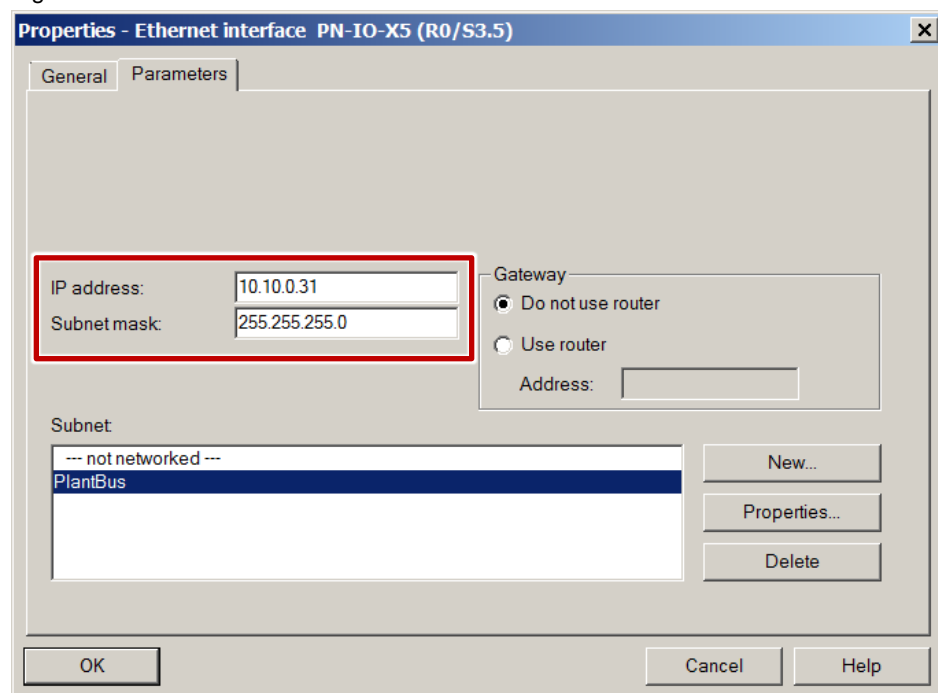


Figure 5-5



2. Save and compile the hardware configuration.

Compiling and loading the changes

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

Figure 5-6

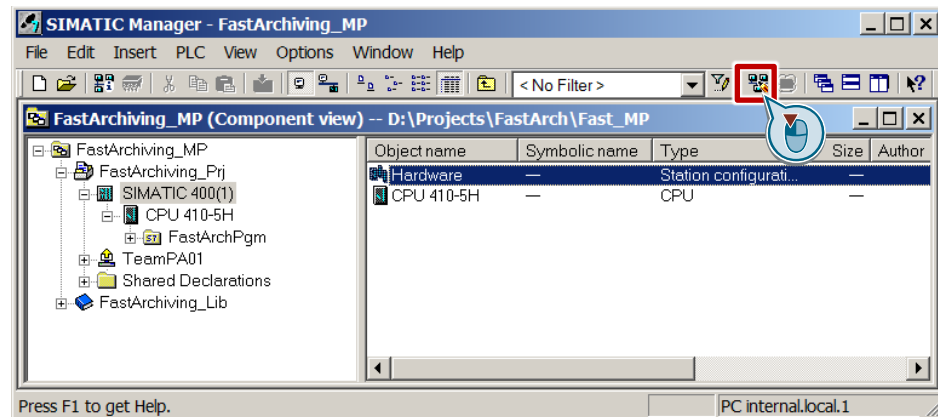
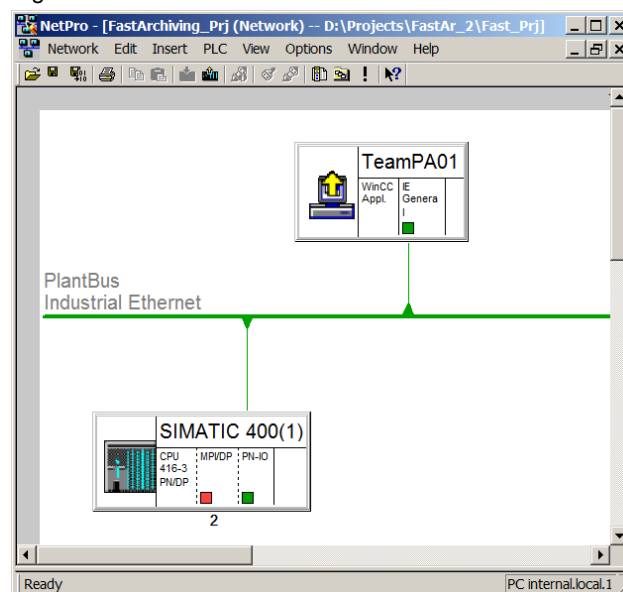


Figure 5-7



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

Note

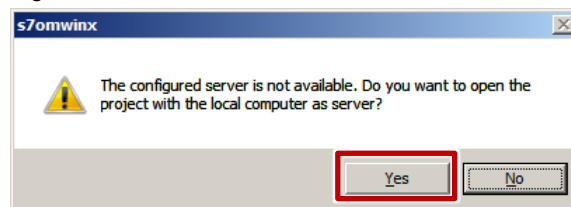
If you can't load the stations, check your configuration:

- 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

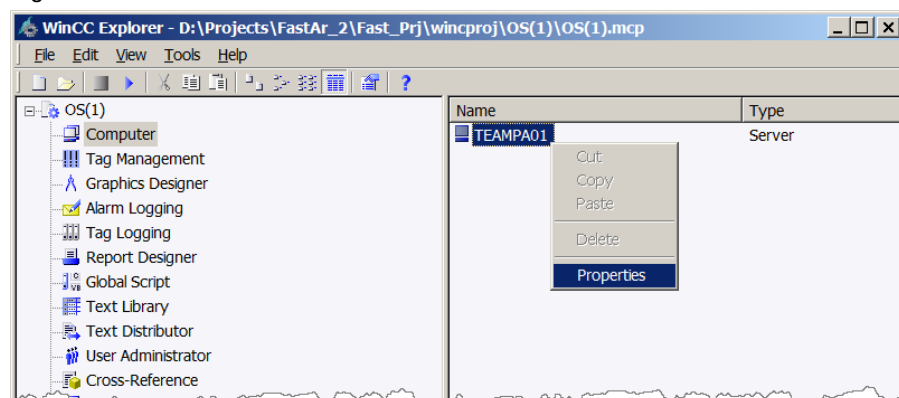
1. Open the WinCC project.
2. If your computer name is different than that of the example project, you will get the following message:
"The configured server is not available. Would you like to open the project with the local computer as the server?"
Confirm this message by clicking on "Yes".

Figure 5-8



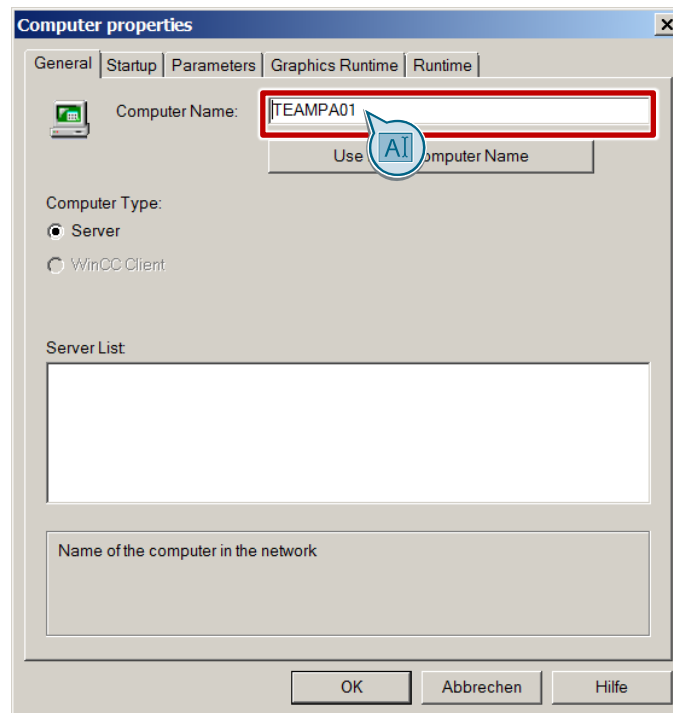
3. Adjust the computer name in the WinCC project.
For this, select the "Computer" option on the left side and open the properties dialog of the configured computer.

Figure 5-9



4. Set the computer name.

Figure 5-10



5. Restart WinCC Explorer for the change to take effect.
6. 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 two hierarchy folders. 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.
- 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

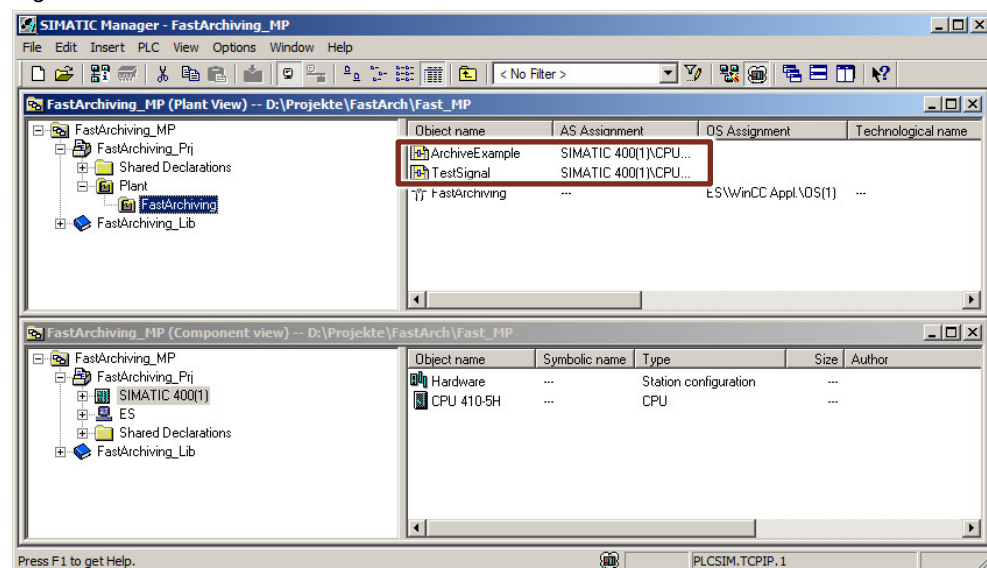


Table 5-1

Continuous Function Chart name	Function
Test signal	<p>The CFC plan contains the modules for generating and controlling two simulated values and is processed in the OB35 (100ms cycle). The chart contains the following blocks:</p> <ul style="list-style-type: none"> • 2x "OpDi01" to start/stop and reset real value simulation on the OS • 1x "TimerP" for automatic resetting of the RESET signal • 2x "SIM_REAL_VAL" for the simulation of the REAL values <p>The simulation blocks for the REAL values can be set with the following parameters:</p> <ul style="list-style-type: none"> • Type of simulation: Sawtooth or sinusoidal wave • Minimum and maximum values • Frequency (rate of repetitions per second)
ArchiveExample	<p>This chart includes the example for fast process value archiving and is processed in OB35 (100ms cycle). The chart contains the following blocks:</p> <ul style="list-style-type: none"> • 2x "Pcs7AnIn" for process value acquisition • 2x "MonAnS" for the monitoring of the process values at the OS • 2x "StruAnIn" for the separation of the APL signal structure • 2x "StruAnIn" for the separation of the APL scale structure • 1x "OpDi01" to start/stop archiving at the OS • 2x "ARSCAN_E" for acquisition of the values to be archived • 1x "AR_MAN_E" for high-speed process value archiving <p>The blocks for the high-speed archiving are parameterized as follows:</p> <ul style="list-style-type: none"> • "Scan01" block <ul style="list-style-type: none"> - 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 <ul style="list-style-type: none"> - 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 <ul style="list-style-type: none"> - 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 [4.4 Configuration in WinCC](#).

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

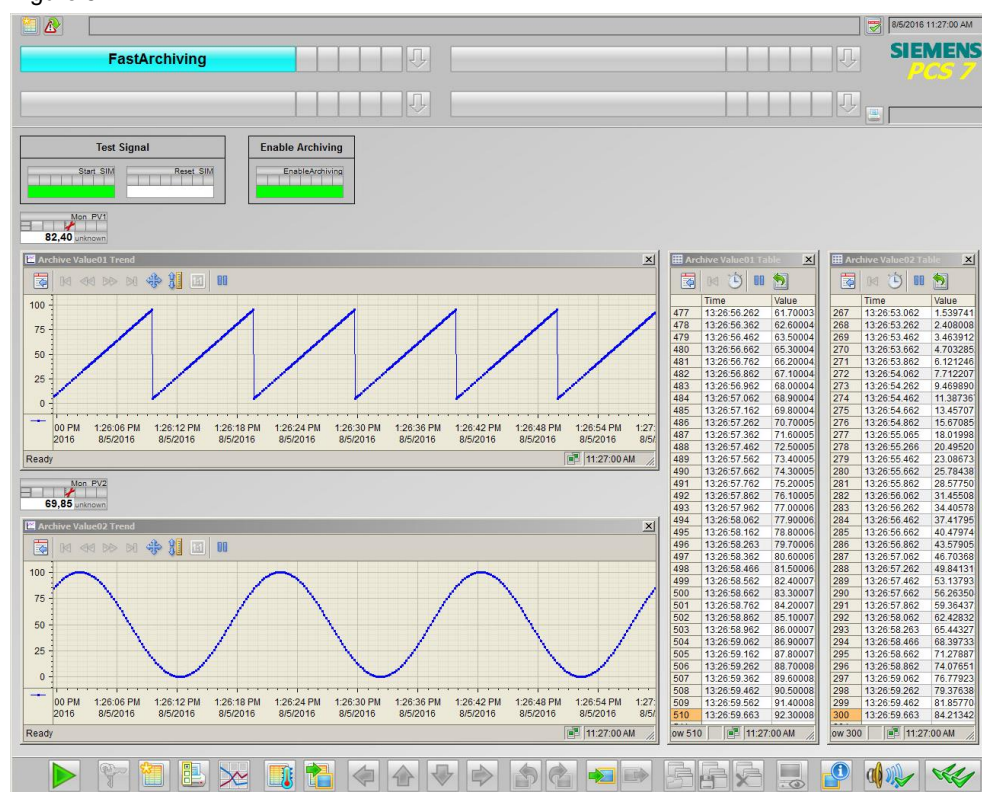


Table 5-2

No.	Object type	Description
1	WinCC OnlineTrendControl "Archives Value01 Trend"	This trend shows the values of the archive variables. "#FastArchPgm#RawArchiv#A#00000001#0001" , which is detected by the CFC block Scan01.
2	WinCC OnlineTrendControl "Archives Value02 Trend"	This trend shows the values of the archive variables. "Value02" , which is detected by the CFC block Scan02.
3	WinCC OnlineTableControl "Archive Value01 Table"	This table 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 "Archive Value02 Table"	This table shows the values of the archive tag "Value02"

No.	Object type	Description
5	APL OpDi01 block icon "Start_SIM"	This binary process operation allows you to activate / deactivate the test signal.
6	APL OpDi01 block icon "Reset_SIM"	This binary process operation allows you to reset the simulated process values.
7	APL OpDi01 block icon "EnableArchiving"	This binary process operation allows you to start / stop the archiving process.
8	APL MonAnS block icon "Mon_PV1"	Monitoring the process value PV1
9	APL MonAnS block icon "Mon_PV2"	Monitoring process value PV2

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
 - At the block "EnableArchiving", set the input parameter "LiOp" = 1.
Now the operation of the block is done via the inputs "SetLi" and "RstLi".
 - To start archiving, at the block "EnableArchLink", set the input parameter "In" = 1
 - To stop archiving, at the block "EnableArchLink" set the input parameter "In" = 0
- 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.

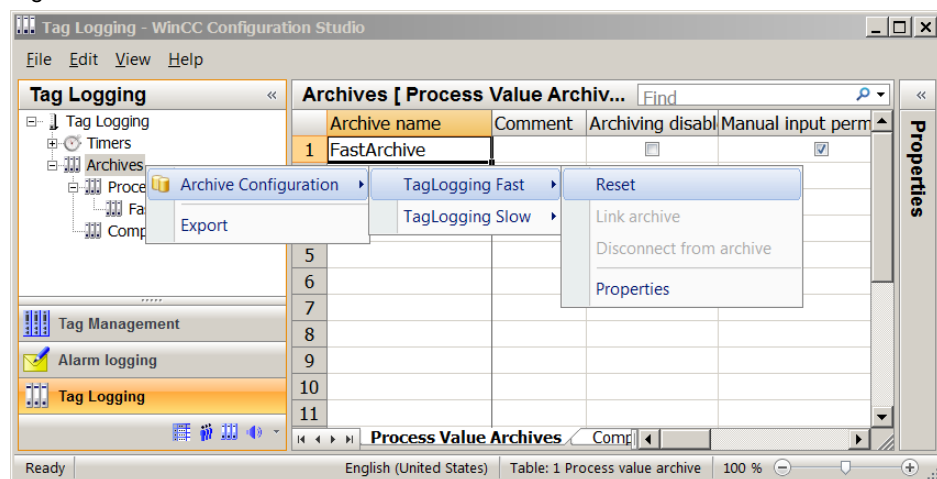
If you want to delete this data, you can reset the WinCC archive "Tag Logging Fast".

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:

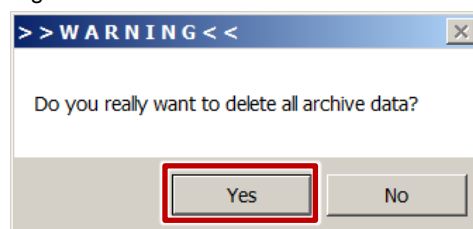
1. Open the WinCC Tag Logging in WinCC Explorer.
2. Right-click on "Archives" to open the associated context menu. Navigate to "Archive Configuration> TagLoggingFast" and click on "Reset".

Figure 6-1



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



4. Save and close the WinCC Tag Logging.

7 Appendix

7.1 Service and Support

Industry Online Support

Do you have any questions or need assistance?

Siemens Industry Online Support offers round the clock access to our entire service and support know-how and portfolio.

The Industry Online Support is the central address for information about our products, solutions and services.

Product information, manuals, downloads, FAQs, and application examples – all the information you need is accessible with just a few mouse clicks at:

<https://support.industry.siemens.com>

Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-made offers – ranging from basic support to individual support contracts.

You send queries to Technical Support via Web form:

www.siemens.com/industry/supportrequest

SITRAIN – Training for Industry

With our globally available training courses for our products and solutions, we help you with innovative learning methods.

You can find out more about the trainings and courses offered as well as their locations and dates under:

www.sitrain-learning.siemens.com/DE/en/index.do

Service offer

Our range of services includes, inter alia, the following:

- Plant data services
- Spare parts services
- Repair services
- On-site and maintenance services
- Retrofitting and modernization services
- Service programs and contracts

You can find detailed information on our range of services in the service catalog:

<https://support.industry.siemens.com/cs/sc>

Industry Online Support app

You will receive optimum support wherever you are with the "Siemens Industry Online Support" app. The app is available for Apple iOS, Android and Windows Phone:

<https://support.industry.siemens.com/cs/ww/en/sc/2067>

7.2 Links and Literature

Table 7-1

	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Download page for this entry https://support.industry.siemens.com/cs/ww/en/view/23780904

7.3 Change documentation

Table 7-2

Version	Date	Modification
V1.0	12/2006	First version
V1.1	03/2009	<ul style="list-style-type: none"> Changing the time calculation from TIME to REAL Extend the following block interfaces / functions <ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Complete revision of the document <ul style="list-style-type: none"> 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 the SCL source files) <ul style="list-style-type: none"> CAUTION Interface change Restructuring of the version history Revision of comments Streamlining of internal data management Improvement of startup characteristics
V2.1	01/2015	<ul style="list-style-type: none"> Documentation update to SIMATIC PCS 7 V8.1 Example project update
V2.2	08/2016	<ul style="list-style-type: none"> Documentation update to SIMATIC PCS 7 V8.2 Example project update Adaptations
V2.3	11/2017	<ul style="list-style-type: none"> Documentation update to SIMATIC PCS 7 V9.0 Example project update New feature: Time-of-day synchronization 3.3.6 AR_MAN_E and ARSCAN_E update to version 1.5 CAUTION Interface change