SIEMENS
Ingenuity for life

Data Exchange between S7-1200 and SENTRON PAC via MODBUS TCP (Set 22)

SIMATIC S7-1200, SENTRON PAC3200

https://support.industry.siemens.com/cs/ww/en/view/40614428

Siemens Industry Online Support

Warranty and Liability

Note

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

We do not accept any liability for the information contained in this document.
Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act ("Produkthaftungsgesetz"), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract ("wesentliche Vertragspflichten"). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.
Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of the Siemens AG.

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 if and to the extent necessary and with appropriate security measures (e.g. use of firewalls and network segmentation) in place.
Additionally, Siemens' guidance on appropriate security measures should be taken into account. For more information about industrial security, please visit http://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

1 Task

1.1 Overview

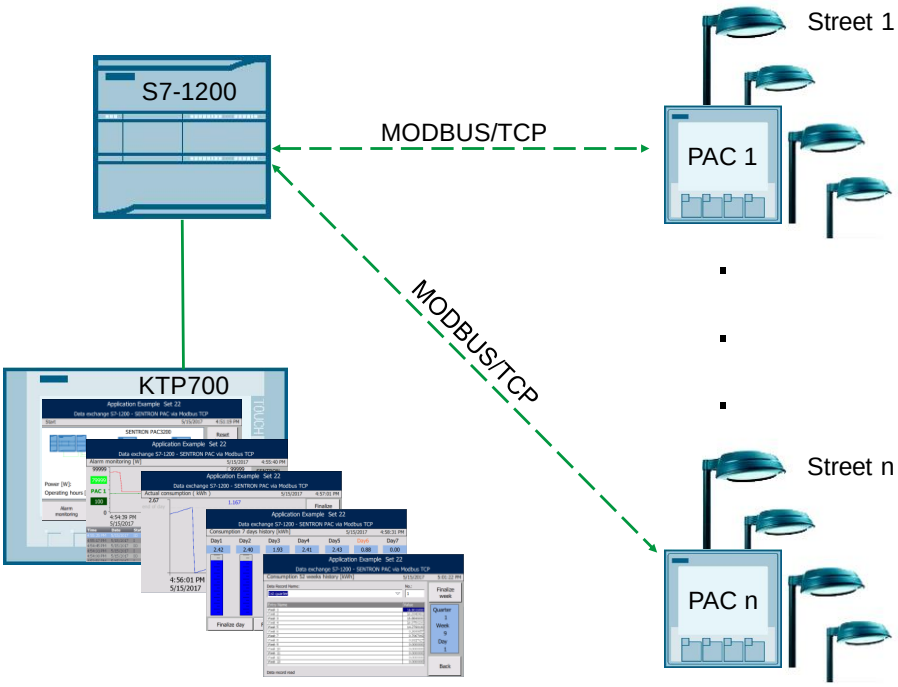
Introduction

The effective power absorbed by several SENTRON PAC 3200 power monitoring devices is to be read out deterministically and the total energy consumption is to be calculated by a S7-1200 controller via Modbus on TCP.

Overview of the automation task

The figure below provides an overview of the automation task.

Figure 1-1



Description of the automation task

The electric power of several street lamps measured by SENTRON PAC3200 devices via the Modbus/TCP protocol is to be transferred to the SIMATIC S7-1200, displayed on an operator panel and processed further to determine the energy consumption.

The power of the street lamps is to be monitored for their upper and lower limits. Specific measuring points are to generate the message “Defective illuminate" when the lower limit falls below the threshold and when the upper limit gets exceeded the message “Short circuit pac x" is to be generated. If an Ethernet connection is interrupted, a message indicating which SENTRON PAC3200 device is inaccessible is to be generated. All messages shall be assigned a time stamp, be displayed on a KTP700 and entered in its message buffer.

The total energy consumption of all streets is calculated with the help of the S7-1200 and displayed as a chronological sequence. The daily energy consumption of a week is to be logged as a bar chart. After a week, the daily total energy consumption is summarized into weekly energy consumption. The weekly energy

consumption throughout the year is then archived in the KTP700 as a recipe of 4 quarterly data records of 13 weeks each.

It shall be possible to record and reset the operating hours of all SENTRON PAC3200 devices. It shall be possible to delete all saved and archived energy consumption values. It shall also be possible to empty the alarm buffer.

For demonstration purposes, it shall be possible to simulate the daily change manually and automatically and the weekly change manually.

2 Solution

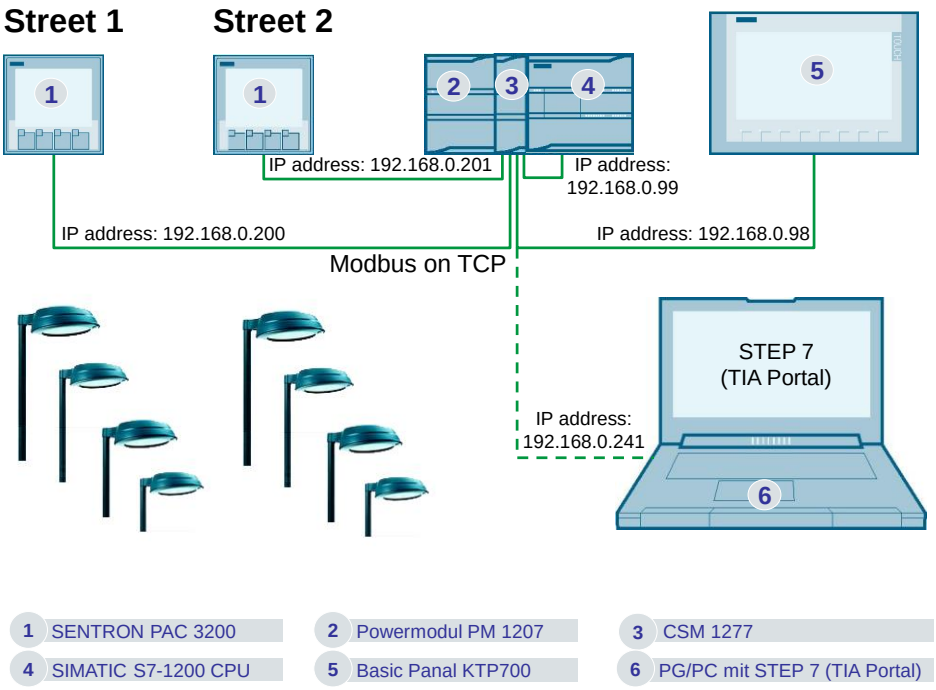
The automation task is demonstrated using the example of data communication with two SENTRON PAC3200 devices.

2.1 Overview

Schematic layout

The figure below shows a schematic overview of the most important components of the solution:

Figure 2-1



Configuration

A SIMATIC S7-1200 is to be linked with two SENTRON PAC 3200 via a CSM 1277 switch.
With the help of the Modbus/TCP protocol, data can be exchanged between the SENTRON PAC3200 power monitoring devices and the S7-1200 CPU. As a result, the output of the street lamps is to be measured. The SIMATIC S7-1200 is to calculate the energy consumption for two streets from the average power demand. The energy consumption shall be visualized in a SIMATIC KTP700. However, to simulate the user interface, the simulation in the TIA Portal can also be used.

Class C network

The IP addresses of the communication nodes only differ in the last octet (192.168.0.x), which makes it a class C network. As a subnet mask, “255.255.255.0” is selected for a class C network.
To expand to several SENTRON PAC3200 devices in the sample program, further IP addresses can be used within this class C network (192.168.0.x).

Advantages

The solution presented here, offers you the following advantages

- Time and cost savings
- Expandability
- Low maintenance effort, since short-circuits and lamp failures are reported
- Since the operating hours are recorded, the time when the lamps must be replaced can be calculated

Topics not covered by this application

This application does not include a description of:

- SIMATIC S7-1200 controller family
- Operator panels series KTP Basic
- SENTRON PAC3200 power monitoring device

Basic knowledge of these topics is assumed.

Assumed knowledge

Basic knowledge in handling the TIA Portal (STEP 7 and WinCC Basic) is assumed.

2.2 Hardware and software components

2.2.1 Validity

This application is valid for
STEP 7 V14 or higher
S7-1200

2.2.2 Components used

The application was created using the following components:

Hardware components

Table 2-1

Component	Qty	Article number	Comment
7KM PAC3200	2	7KM2112-0BA00-3AA0	FW V2.4.2
POWER SUPPLY S7-1200 PM1207	1	6EP1332-1SH71	
COMPACT SWITCH MODULE CSM 1277	1	6GK7277-1AA10-0AA0	
CPU 1215C, DC/DC/DC, 14DI/10DO/2AI/2AO	1	6ES7215-1AG40-0XB0	FW V4.2
SIMATIC HMI KTP700 BASIC	1	6AV2123-2GB03-0AX0	Optional (can also be simulated in TIA Portal)
TP CORD RJ45/RJ45 2M	4	6XV1870-3QH20	

Software components

Table 2-2

Component	Qty	Article number	Comment
STEP 7 Basic V14	1	6ES7822-0AA04-0YA5	With update 2

Example files and projects

The following list includes all files and projects that are used in this example.

Table 2-3

Component	Comment
40614428_Set22_PRJ_v14.zip	This zip file contains the STEP 7 project.
40614428_Set22_DOC_v14_en.pdf	This document.

3 Mode of Operation

The following chapter is to provide a better understanding and for further details on the configuration and programming of the application example. The content of this chapter is not essentially required for commissioning and operation.

3.1 General overview

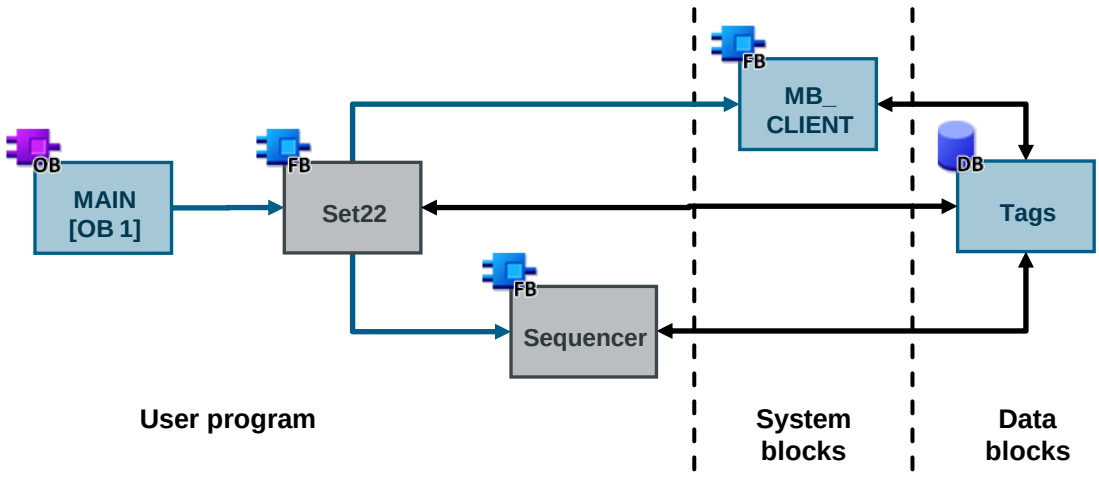
For communication with the SENTRON PAC3200 devices, the MB_CLIENT[1] instruction is used. This function block maps the Modbus/TCP protocol and internally uses the functionality of the communication instructions TSEND, TRCV, TCON and TDISCON.

The resources for the S7-1200 are limited to a maximum of eight parallel connections. To enable communication with more than eight SENTRON PAC3200 devices, the data exchange with the individual devices in this application is handled in series via a single connection.

Program overview

Figure 1-1 shows the call hierarchy of the code blocks and the access to the data blocks.

Figure 3-1: Program structure



The main component of the application is the user-specific function block FB "Set22". This block is called in the cyclic organization block “MAIN” and has the following tasks:

- Controlling the communication with the SENTRON PAC3200 devices in form of a sequencer with periodic calls of FB "MB_CLIENT".
- Preparation and evaluation of communication data according to the requirements of the task description in chap. 1.

Data block "Tags" contains specific information regarding the SENTRON PAC3200 measuring devices, such as their IP addresses, their definable power limit values for short-circuit or defective lamp detection, as well as further stored data (current power values, alarm bits, operating hours, etc.). The data is accessed via FB "Set22".

[1] You can find the instruction in the “Instructions” task card in *Communication > Others > MODBUS TCP*

The FB “Sequencer” controls the communication sequence for sequential data exchange with several PACs.

Block overview

The following table gives you an overview of the blocks used.

Table 3-1: blocks used

Block	Description
OB "Main"	Cyclic organization block: It only contains the call of FB "Set22".
FB "Set22"	Function block for the deterministic data exchange with several SENTRON PAC3200 measuring devices (cyclic program sequence)
DB "InstSet22"	Instance data block of FB "Set22”
FB "MB_CLIENT"	Integrated instruction to MODBUS TCP communication of SIMATIC S7-1200 as client in STEP 7 (TIA Portal)
DB "Tags"	contains the data of all SENTRON PAC3200 measuring devices necessary for the application
FB "Sequencer"	Function block for the deterministic data exchange with several SENTRON PAC3200 measuring devices (communication sequence)

3.2 "Set22" function block

The function block for deterministic data exchange with several SENTRON PAC3200s via Modbus/TCP is called in the cyclic OB.
"Set22" includes the cyclic program part and calls the FB “Sequencer” to control the communication sequence.

Figure 3-2: Call of FB "Set22”

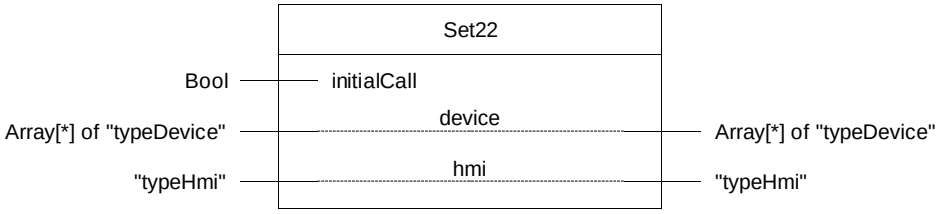


Table 3-2: Parameter of FB "Set22”

Name	P type	Data type	Comment
initialCall	IN	Bool	First call of cyclic OB (interface parameter “Initial_Call”)
device	IN_OUT	Array[*] of "typeDevice" (Table 3-3)	Array with the data of the SENTRON PAC measuring device
hmi	IN_OUT	"typeHmi" (Table 3-6)	PLC data type of HMI tags

Table 3-3: PLC data type “typeDevice”

Name	Data type	Comment
ipOctet4	USInt	Last octet of the IP address (to be specified by user). The application uses the addresses 200 and 201.
pAverage	Real	Current active power resulting as the average value from three double-words read from register 40066 in direct succession. The active power is displayed at the operator panel and used for calculating the total energy consumption.
hoursCounter	UDInt	Word of the bit alarms visualized in the message display on the operator panel.
resetHoursCounter	Bool	The operating hours counter is displayed on the operator panel and used for calculating the total energy consumption.
bitAlarms	UInt	Saved reset request for the operating hours counter. It is triggered on the operator panel by pressing a button and reset by the program after its execution.
commBreakdown	"typeCommBreakdown" (Table 3-4)	Data type with the information regarding communication breakdown
pavLimit	"typePavLimit" (Table 3-5)	Data type with performance limits ($P_{average}$) to diagnose connected consumers

Table 3-4: PLC data type “typeCommBreakdown”

Name	Data type	Comment
started	Bool	Start identifier set in the event of a communication error. If this is the case, the program keeps polling how long the error has already been pending for and sets the “Communication breakdown PAC n” alarm, if the “instSequencer.statCommErrMsgDelay” time (static tag in DB "InstSet22") has been exceeded.
errMsgTime	DTL	Time at which the "Communication breakdown PAC n" error message is output, if “started” = TRUE

Table 3-5: PLC data type “typePavLimit”

Name	Data type	Default value	Comment
high	Real	79999.0	Upper active power limit in W, to be entered by the user, from which a short-circuit in the respective line is reported.
low	Real	100.0	Upper active power limit in W, to be entered by the user, from which on a lamp failure in the respective line is reported.

Table 3-6: PLC data type “typeHmi”

Name	Data type	Default value	Comment
dataRecordNo	Word	1	Data record number (quarter 1..4)
day	DInt	0	Week day number (1..7)
eDay	Array[1..7] of Real		Array of energy data per day (per week)
ekWh	Real	0.0	Current energy consumption (in kWh)
ekWhLimit	Real	0.0	Upper energy limit for the trend curve of the current consumption (in kWh)
eWeek	Array[1..13] of Real		Array of the energy data per week (per quarter)
finalizeDay	Bool	false	Request to finalize day
finalizeWeek	Bool	false	Request to finalize week
globalBitAlarms	UInt	0	Trigger tag for the global HMI bit alarms (“Recipe transmission error”)
reset	Bool	false	Request to reset the operating hours counter of PAC
resetDisplay	Bool	false	Display of enabling the reset of operating hours counter of PAC
resetHistory	Bool	false	Request to reset all energy consumption values
week	DInt	1	Week number (1..13)
jobMailbox	Array[0..3] of Word		“jobMailbox” (control job) area pointer
dataRecord	Array[0..4] of Word		“Data record” area pointer
sysTime	DTL	DTL#1970-01-01-00:00:00	CPU system time

The following static parameters of FB “Set22” can be modified, if required:

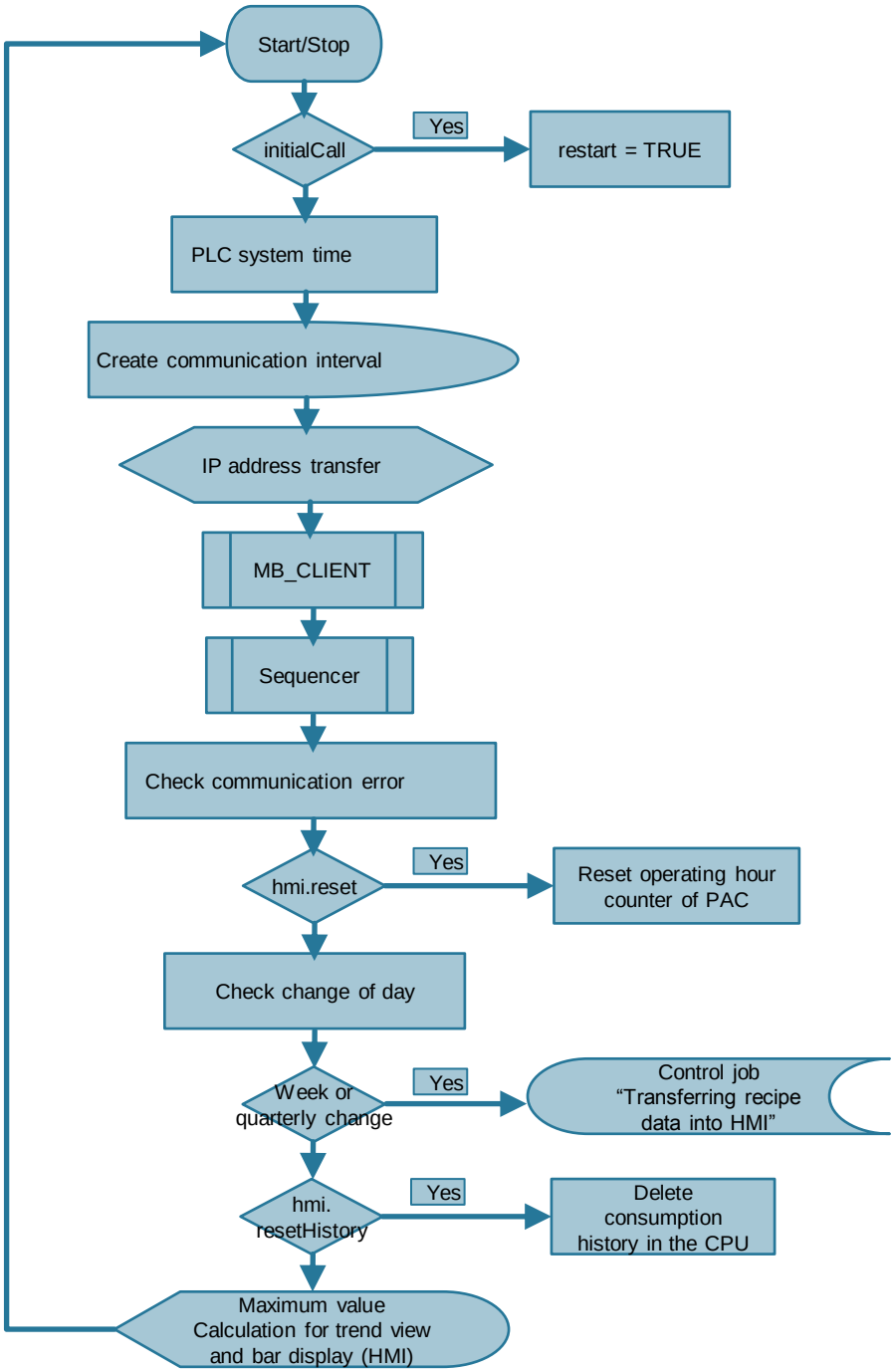
Table 3-7: Parameter of FB “Set22” that can be changed by the user

Name	Data type	Default value	Comment
statScanInterval	Time	T#2s	After this time has elapsed, the connection with the MODBUS slave n is terminated and reestablished with slave n+1. It is assumed that the communication with a slave can be fully processed during this time. If the time is exceeded, the error routines apply.

Note

Do not select the "statScanInterval" time smaller than the static instance parameter "Blocked_Proc_Timeout" and “Rcv_Timeout” of FB "MB_CLIENT" (minimum: 500 ms) -> see Table 3-14.

Figure 3-3: Cyclic program sequence



3.2.1 Program explanation

Restart

During a "initialCall" restart, a "restart" bit is set. It is used for preventing the calculation of the energy consumption value for the first communication to each SENTRON PAC3200 after the restart. For the first communication, only the current

operating hours counter value of each measuring device is entered into the “tags” DB as reference “device[n].hoursCounter”. This is necessary, since calculating an energy quantum requires a time difference, which however, can only be formed after the second communication with each SENTRON PAC3200 device.

PLC system time

The PLC system time is used to specify and store a communication breakdown message time ("Tags".device[n].commBreakdown.errMsgTime) for each SENTRON PAC3200 device with communication breakdown. If the communication breakdown message time is exceeded without reestablishing the connection with the respective SENTRON PAC3200 device in the meantime, a message appears on the HMI (Connection breakdown). The communication breakdown message time is formed by adding the time of detection of a communication breakdown and an editable waiting period "InstSet22“.instSequencer.statCommErrMsgDelay (see last bullet point in chap. 3.6).

Creating communication interval

The communication with the SENTRON PAC3200 devices is serial. First, the connection to device n is established, then communication takes place and afterwards, the communication is terminated in order to continue with device n+1. The time distance ("InstSet22".statScanInterval) with which this happens can be configured. A pulse is generated in this grid, which serves as communication trigger.

IP address transfer / MB_CLIENT / FB “Sequencer”

For the preparation to communication with the appropriate SENTRON PAC, the 4th octet of the IP address ("Tags".device[n].ipOctet4) is copied into the connection data “CONNECT” of FB "MB_CLIENT". FB "MB_CLIENT" is called cyclically. The communication is controlled by its actual parameters, which are continuously adjusted in FB “Sequencer”.

Checking communication error

It is polled whether a current reporting time of the communication breakdown has been entered in DB "Tags" for the currently processed SENTRON PAC3200 device.
If this is the case, and the time has already been exceeded, the “Connection breakdown” (incoming) message is displayed on the operator panel. It marks an uninterrupted communication breakdown with the respective SENTRON PAC3200 device, which is longer then the configured value "InstSet22“.instSequencer.statCommErrMsgDelay.

It is polled whether the MB_CLIENT outputs a DONE pulse for the currently processed SENTRON PAC3200 device. If yes, the messages “Connection error PAC n” and/or “Connection breakdown PAC n” – as far as pending – are reset, or labeled as “outgoing” (“O”) on the operator panel, and an entered communication breakdown message time is disabled and the "Tags".device[n].CommBreakdown.started bit is reset.

Resetting operating hour counter of PACs

If the resetting of all operating hours counter is requested via the operator panel ("Tags".hmi.reset), every measuring point must be informed of this. The respective bits "Tags".device[n].resetHoursCounter are set via loop processing. They are deleted after successfully resetting the respective operating hours counter.

Resetting all operating hour counters takes relatively long, since the communication to all SENTRON PAC3200 devices must be performed in sequence. Therefore, the processing time is displayed for the user on the operator panel ("Tags".hmi.resetDisplay). In a loop, all measuring points are polled, whether the resetting of the operating hours counter is requested for them. If this is the case for at least one SENTRON PAC3200 device, "Tags".hmi.resetDisplay is set. If this is no longer the case for any measuring point, the bit is reset.

Checking change of day

For a change of day, proceed as follows:

- The integrated total daily energy consumption of the day ("Tags".hmi.ekWh) is added to the weekly energy consumption ("InstSet22".statEweekKWh) and set to 0.
- The day count ("Tags".hmi.day) is incremented. If it exceeds the value 7 (weekend), a respective identifier ("InstSet22".statWeekOverflow) is set. The day count is used to enter the daily energy consumption under the correct day of the week in the “04_Consumption 7 day history” screen.

As a day change, the application does not use the transition 24:00 ⇨ 0:00, but for simulation purposes the following criteria:

- Daily energy consumption exceeds a certain limit value (“InstSet22”.statEdayOverflow bit).
- Day is “terminated” at the operator panel by pressing a button ("Tags".hmi.finalizeDay bit).
- Week is “terminated” at the operator panel by pressing a button ("Tags”.hmi.finalizeWeek bit).

Checking change of week

For a change of week proceed as follows:

- The integrated total weekly energy consumption of the last week ("InstSet22_DB".statEweekKWh) is saved specifically for each week ("Tags".hmi.eWeek[n], n=1..13) and set to 0.
- The day count ("Tags".hmi.day, n=1..7) is reset to 1.
- All values of the total daily energy consumption ("Tags".hmi.eDay[n], n=1..7) are set to 0.
- The week count ("Tags".hmi.week, n=1..13) is incremented or is reset to 1 at the end of the quarter (>13 weeks).
- The quarter count ("Tags".hmi.dataRecordNo, n=1..4) is incremented and reset to 1 at the end of the year (>4 quarters).
- The following two bits are set:
  - "InstSet22".statWriteRecipe bit
    The bit is set on each weekend for one cycle. In the subsequent network it triggers the entry of the recipe transfer parameter into the control job (Job

mailbox) and its start. The recipe is also written to the operator panel at the end of each week.

- "InstSet22".statQuarterEnd bit
  The bit is set at the end of each quarter for one cycle. After successful transfer of the recipe, it causes the 13 week values of the total consumption of the last quarter to be overwritten with 0 in the PLC ("Tags".hmi.eWeek[n], n=1..13).

Control job (Job mailbox) “Transferring recipe data into HMI”

Using "InstSet22".statWriteRecipe = TRUE the recipe transfer parameters are entered in the Job mailbox, initiating its start. "InstSet22".statRecipeStarted is set as the execution message. If the job was triggered at the end of the quarter ("InstSet22".statQuarterEnd = TRUE bit), its start is noted by "InstSet22”.statRecipeStartedQuaterEnd bit.

The status ("Tags".hmi.dataRecord[3]) of the synchronous data record transmission is polled.

- If the transmission was completed <u>without errors</u> ("Tags".hmi.dataRecord[3] = 4), a possibly pending recipe transmission error (originating from the last transmission) is reset (“outgoing” message). If the transmission was triggered at the end of the quarter, the 13 week values for active power in the current week ("Tags".hmi.eWeek[n], n=1..13) stored in the PLC are deleted.
- If the transmission was completed <u>with error</u> ("Tags".hmi.dataRecord[3] = 12), or an end status ("Tags".hmi.dataRecord[3] ≠ 4,12) has not yet been output within 5s after triggering the recipe transfer (“RECIPE_ERROR_TIME” constant), a recipe transfer error (“incoming” message) is reported on the operator panel. In real operation, the user has now one week to save the 13 week values for active power of the last quarter values ("Tags".hmi.eWeek[n], n=1..13) that have been saved in the PLC, before they are overwritten.

Afterwards the program will reset the bits "InstSet22".statRecipeStarted or "InstSet22".statQuarterEnd and overwrite the status of the synchronous data record transmission with 0.

Delete consumption history in the CPU

When requested by the operator panel, the entire history stored in the PLC can be deleted. This refers to the following data:

Table 3-8: Deleting the history

Description	Variable in DB “InstSet22”	
Current daily energy consumption	hmi.ekWh	⇨ 0
Current weekly energy consumption	statEweekKWh	⇨ 0
All daily energy consumption values of the current week	hmi.eDay[n], n=1..7	⇨ 0
All weekly energy consumption values of the current quarter	hmi.eWeek[n], n=1..13	⇨ 0
Quarter number	hmi.dataRecordNo	⇨ 1
Week number	hmi.week	⇨ 1
Day number	hmi.day	⇨ 1

Please consider, that the next recipe data record to be transferred to the operator panel is filed after deleting the history for Quarter/Week/Day = 1/1/1 and overwrites the existing data.

Note In a real application it makes sense not to reset the above time tag to 1, but to the current calendar values. If necessary, modify the program in this sense.

Maximum calculation for trend and bar display in HMI

The ordinate scale in the trend view of the current daily energy consumption on the "03_Actual consumption" HMI screen, automatically adjusts to the power level of the connected consumers. The upper ordinate limit ("Tags".hmi.ekWhLimit) is defined as the energy which would accumulate if all consumers were at their upper power limit for 24 hours (limit from when the short-circuit is detected). "Tags".hmi.ekWhLimit is calculated via all SENTRON PAC3200 devices by means of loop processing.

Depending on this maximum value, the “statEdayOverflow” bit is also set. If the daily energy consumption exceeds 90% of the ordinate upper limit, a day change will take place:

```
#statEdayOverflow := (#hmi.ekWh >= #E_DAY_OVERFLOW_FACTOR *
#hmi.ekWhLimit);
```

3.3 "Sequencer" function block

The FB “Sequencer” controls the communication sequence for deterministic data exchange with several SENTRON PAC3200s via Modbus/TCP and is called in FB "Set22".

Figure 3-4Sequencer

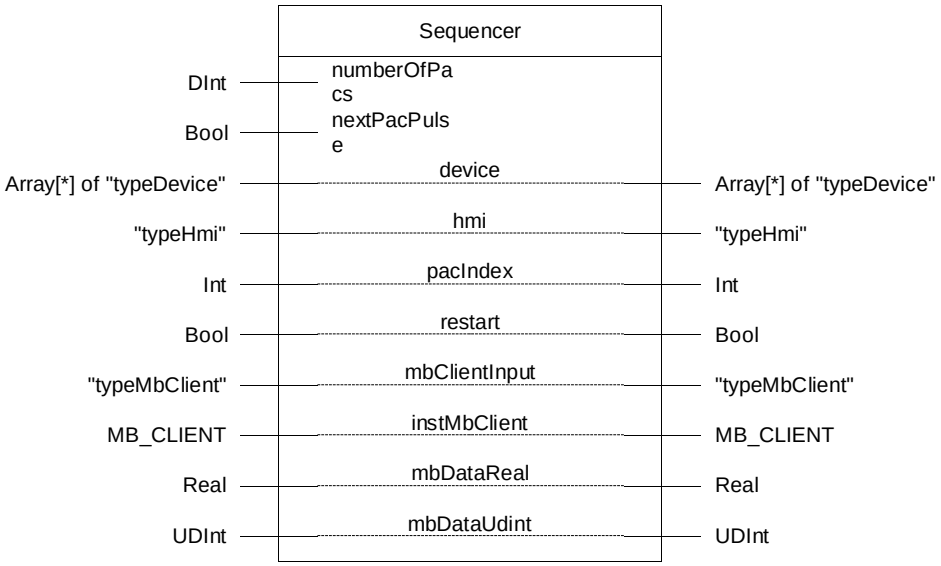


Table 3-9: Parameters of "Sequencer"

Name	P type	Data type	Comment
numberOfPacs	IN	DInt	Number of SENTRON PACs
nextPacPulse	IN	Bool	Communication trigger pulse
device	IN_OUT	Array[*] of "typeDevice" (Table 3-3)	Array with the data of the SENTRON PAC measuring device
hmi	IN_OUT	"typeHmi" (Table 3-6)	Array with the HMI tags
pacIndex	IN_OUT	Int	Index, which SENTRON PAC is currently addressed
restart	IN_OUT	Bool	first communication run to all SENTRON PAC
mbClientInput	IN_OUT	"typeMbClient" (Table 3-10)	Input parameters for calling “MB_CLIENT”
instMbClient	IN_OUT	MB_CLIENT	Instance data of “MB_CLIENT”
mbDataReal	IN_OUT	Real	Data buffer of “MB_CLIENT” in floating-point number format
mbDataUdint	IN_OUT	UDInt	Data buffer of “MB_CLIENT” in integer format

Table 3-10: PLC data type “typeMbClient”

Name	Data type	Default value	Comment
req	Bool	false	Executing the requested data transmission if TRUE
disconnect	Bool	false	Disconnecting a connection
mode	USInt	0	Access type of request (0 = read, 1 = write)
mbDataAddr	UDInt	40066	Start address of the access data of Modbus client
connect	TCON_IP_v4		Data type with connection data

The following static parameters of FB "Sequencer” can be modified if required:

Table 3-11: Static parameters of FB "Sequencer" that can be changed by the user

Name	Data type	Default value	Comment
statCommErrMsgDelay	Time	T#5m	The time a communication error with a SENTRON PAC3200 device must be continuously pending until the communication breakdown is reported on the operator panel. “Continuously” means, that during that time not a single communication request for the respective device is satisfied.
statSampleMax	UInt	3	The active power is read several times in direct sequence from the SENTRON PAC3200 measuring device and the average value is formed. How often the reading takes place is defined by "statSampleMax”.

Figure 3-5: Communication sequence (1)

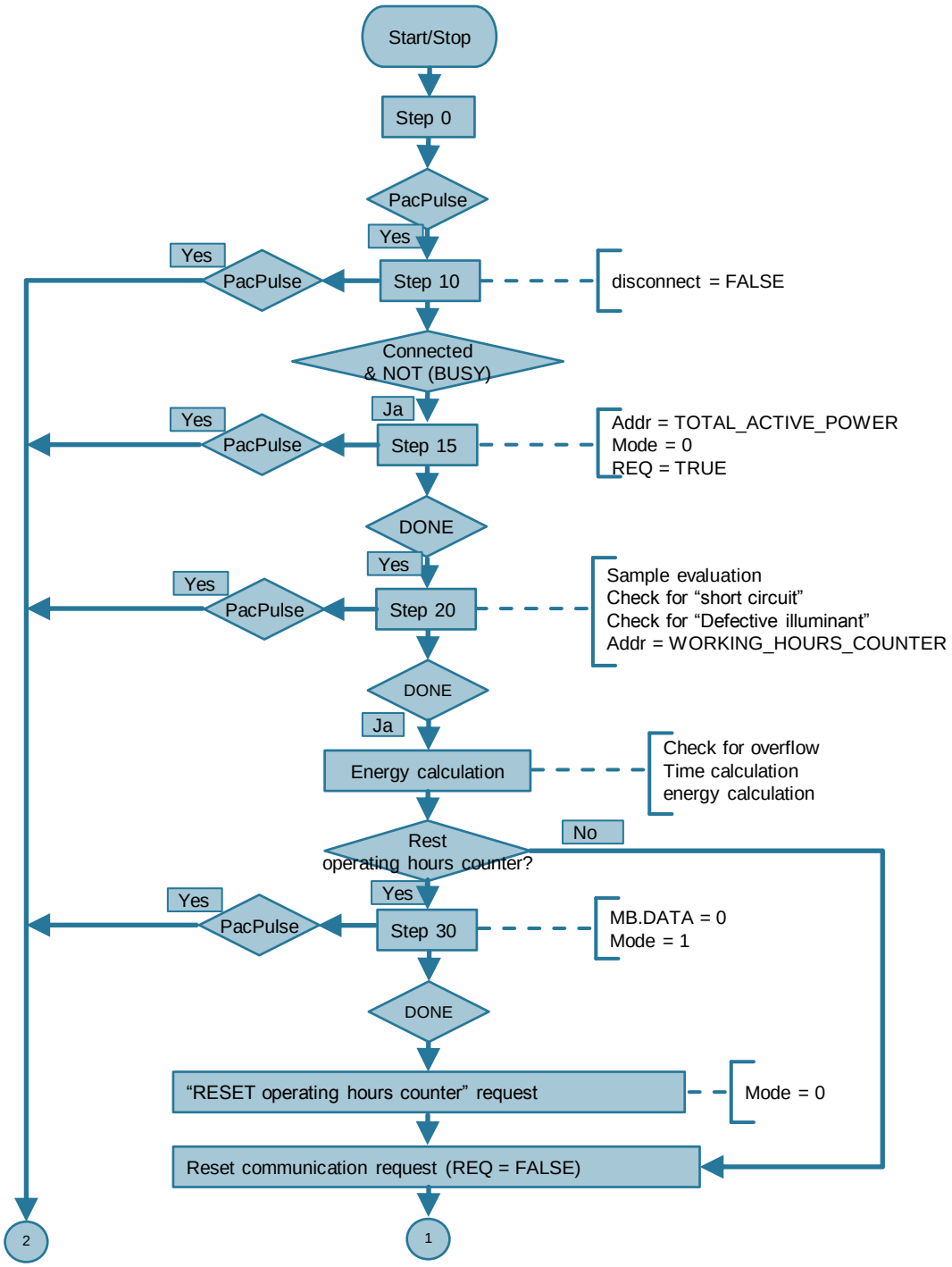
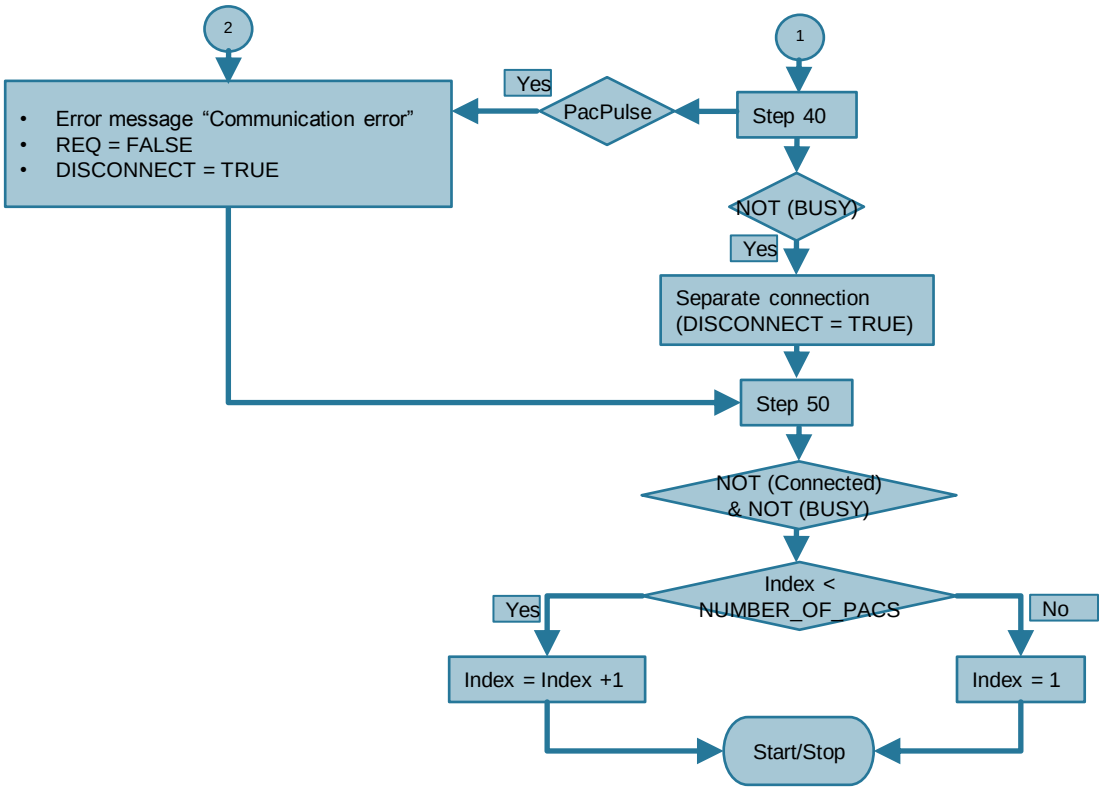


Figure 3-6: Communication sequence (2)



3.3.1 Program explanation

Step 0

In step 0, it is waited until the “Trigger pulse” event. After it has occurred, the MB_CLIENT parameter “DISCONNECT” is reset and proceeded to step 10.

Step 10

If the “CONNECTED and NOT BUSY” conditions of MB_CLIENT have been met, it is configured for reading the electrical power[2]. The Modbus register address and the mode (read or write) are specified. Subsequently, the communication request is started by setting the MB_CLIENT parameter REQ and branched to step 15.

If a renewed communication trigger pulse arrives, before the above conditions are fulfilled, it can be assumed, that an error is present and an error routine is branched off which ends in step 50.

Step 15

It waits for the “DONE” message of “MB_CLIENT”, which indicates that the electrical power was transferred without errors from SENTRON PAC3200 to the receive buffer of S7-1200. After “DONE” has occurred, the step is further processed.

[2] The active power is read from the three phases.

Figure 3-7: Transferred active power value

Since the power value shall be averaged over n random samples, it is measured several times in direct succession. This is performed without exiting step 15. The number of random samples can be configured ("statSampleMax"). After completing n-fold reading, averaging the value and device-related storing in DB "Tags".device[n].pAverage, the active power value is checked for defective illuminants (power too low) and short-circuit (power too high). Error cases are displayed on the control panel. Before branching off to step 20, the Modbus register address for reading the operating hour counter, which is pending next, is transferred to the “MB_CLIENT”.

If a communication trigger pulse arrives in step 15, it can be assumed, that an error is present and an error routine branch is followed that leads to step 50.

Step 20

It waits for the “DONE” message of “MB_CLIENT”, which indicates, that the operating hours counter was transferred without faults from SENTRON PAC3200 to the receive buffer of S7-1200. After “DONE” has occurred, the step is further processed.

Figure 3-8: Transferred operating hours counter value

The device-related energy quantum, which is formed by multiplying the averaged active power from step 15 and the time difference between the last two operating hours counter values (currently read value and value of the last send request for this measuring device), is in this step always added back to the total energy consumption value of all SENTRON PAC ("Tags".hmi.ekWh). It is then saved day-specifically ("Tags".hmi.eDay[n], n=1..7).

When calculating the energy consumption, the following must be considered in this step:

- Overflow of the operating hours counter in the SENTRON PAC3200.
- At the first run of step 20 for all SENTRON PAC3200 measuring devices after the restart, neither a device-related energy quantum can be calculated, nor can the total energy consumption value be updated. At this point in time, no time difference for the operating hours counter is available yet for calculating the energy quanta. For the first respective communication cycle after the restart, only the current operating hours counter values are entered into "Tags".device[n].hoursCounter as reference. In the respective second communication cycle from the restart on, the time differences of the operating hour counters are then available for calculating the energy quanta.

At last, step 20 polls whether the reset of the operating hours counters of all SENTRON PAC3200 devices was requested via the operator panel. If yes, the default value 0 of the operating hours counter is written to the send buffer, the “write” mode for MB_CLIENT is set, branching off to step 30. If no, the communication request is cancelled by resetting the MB_CLIENT parameter “REQ”, branching off to step 40.

If in step 20 a communication trigger pulse arrives, it can be assumed, that an error is present and an error routine branch is followed that leads to step 50.

Step 30

It waits for the “DONE” message of “MB_CLIENT”, which indicates, that the operating hours counter of the SENTRON PAC3200 measuring device, to which the communication is currently directed, was overwritten with 0. After “DONE” has occurred, the further processing follows.

The reset request of the operating hours counter for the currently processed SENTRON PAC3200 measuring device ("Tags”.device[n].resetHoursCounter) is reset, as well as the operating hours counter ("Tags".device[n].hoursCounter).

Then the “read” mode for “MB_CLIENT” is set again, the communication request is cancelled by resetting the “MB_CLIENT” parameter “REQ”, and it is branched off to step 40.

If in step 30 a communication trigger pulse arrives, it can be assumed, that an error is present and an error routine branch is followed that leads to step 50.

Step 40

If the “NOT BUSY” condition of “MB_CLIENT” is true, “DISCONNECT” is set for the next “MB_CLIENT” call, branching off to step 50.

If in step 40 a communication trigger pulse arrives, it can be assumed, that an error is present and an error routine branch is followed that also leads to step 50.

Step 50

If the “NOT CONNECTED and NOT BUSY” condition of MB_CLIENT is fulfilled, the count index of the SENTRON PAC3200 devices is incremented or reset to 1, if all have already been processed. Subsequently, the branch to step 0 is followed again.

If in step 50 a communication trigger pulse arrives, it can be assumed, that an error is present and an error routine branch is followed that in return leads to step 50.

Error routine

If a communication error is reported in steps 10, 15, 20, 30, 40 or 50, this error routine is next. Sequence as follows:

For the currently processed SENTRON PAC3200 device a “Connection error PAC n” (incoming) is output, the communication request cancelled by resetting the MB_CLIENT parameter REQ, and the MB_CLIENT parameter DISCONNECT is set.

Provided no new communication breakdown message time ("Tags”.device[n].commBreakdown.errMsgTime) has yet been entered for the measuring device with communication breakdown, the message time is calculated and saved (see also PLC system time).

Subsequently, the branch to step 50 is followed.

3.4 System function block “MB_CLIENT”

FB “MB_CLIENT” can be found in STEP 7 (TIA Portal) in the instructions in “Communication > Others > MODBUS TCP”. In this application example version 4.1 is used.

Figure 3-9

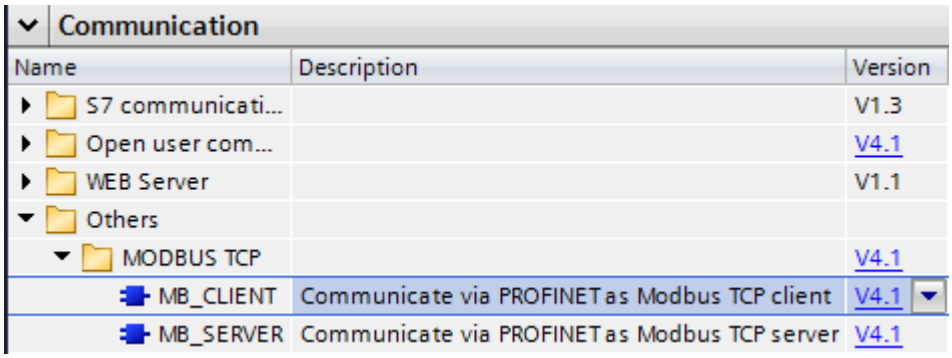


Figure 3-10: Call of system block “MB_CLIENT”

```
REGION 5 MB_CLIENT call
  #instMbClient(REQ := #statMbClientInput.req,
                DISCONNECT := #statMbClientInput.disconnect,
                MB_MODE := #statMbClientInput.mode,
                MB_DATA_ADDR := #statMbClientInput.mbDataAddr,
                MB_DATA_LEN := #MB_DATA_LEN,
                MB_DATA_PTR := #statMbDataReal,
                CONNECT := #statMbClientInput.connect);
END_REGION
```

FB “MB_CLIENT” is generated by STEP 7, as soon as you drag the identically named instruction into your user program when editing the program. In the user program FB "MB_CLIENT" is called once per cycle in FB "Set22". Successively calling with different actual parameters determines whether to read or write (“MB_MODE”), which data is transferred (“MD_DATA_*”), and which server (SENTRON PAC3200) to communicate with (“CONNECT”). The instance data of FB "MB_CLIENT" is saved as multi-instance in DB "InstSet22".

Note

Further information on the “MB_CLIENT” instruction is available in the STEP 7 online help or the S7-1200 system manual (\3\).

Register access

The number of registers you need to access during the data exchange with SENTRON PAC3200 is available in chapter 3.9.3 “Modbus measured variables with the function codes 0x03 and 0x04” of the device manual (\9\). Table 3-6 of the device manual lists all available measured variables of SENTRON PAC3200 and the decisive register offsets.

The register addresses of the measured variables used, result from the MODBUS function codes as follows:

Table 3-12: Measured variables

Operation	Function code	Length (Words)	MB_MODE	MB_DATA_ADDR (Req.adr. + offset)
Read total of the active power	0x03	2	0	40001 + 65 = 40066
Read operating hours counter	0x03	2	0	40001 + 213 = 40214
Write operating hours counter	0x10	2	1	40001 + 213 = 40214

Send/receive buffer

The send/receive buffer simultaneously used for the source and destination of the read or written data only consists of a double-word. It is made accessible through data superimposition (key word ‘AT’, see \4\) as ‘Real’ type (“mbDataReal” tag for active power [W]) as well as ‘UDInt’ type (mbDataUdInt tag for the operating hours counter [s]). The FB “MB_CLIENT” symbolically accesses the data source or the data destination via the actual parameter “statMbDataReal”, using its parameters” MB_DATA_PTR” of the ‘VARIANT’ type in FB “Set22”.

Figure 3-11: Send/receive buffer

```
REGION 5 MB_CLIENT call
  #instMbClient(REQ := #statMbClientInput.req,
                DISCONNECT := #statMbClientInput.disconnect,
                MB_MODE := #statMbClientInput.mode,
                MB_DATA_ADDR := #statMbClientInput.mbDataAddr,
                MB_DATA_PTR := #statMbDataReal,
                CONNECT := #statMbClientInput.connect);
END_REGION

REGION 6 Sequencer
  #instSequencer(numberOfPacs := #statNumberOfPacs,
                 nextPacPulse := #statNextPacPulse,
                 device := #device,
                 hmi := #hmi,
                 pacIndex := #statPacIndex,
                 restart := #statRestart,
                 instMbClient:=#instMbClient,
                 mbClientInput:=#statMbClientInput,
                 mbDataReal:=#statMbDataReal);
END_REGION
```

Sequencer

	Name		Data type	...	Retain
	Input				
	Output				
	InOut				
	device		Array[*...		
	hmi		"typeH...		
	pacIndex		Int	1	Non-retain
	restart		Bool	fal:	Non-retain
	mbClientInput		"typeM...		
	instMbClient		MB_CLI...		
	mbDataReal		Real	0.0	Set in IDB
	mbDataUdint	AT "mbDataReal"	UDInt		Set in IDB

Connection data

Via the parameters “CONNECT” of data type "TCON_IP_v4" the connection data to the SENTRON PACs is stored:

Table 3-13: Data type “TCON_IP_v4"

Name	Data type	Default value	Comment
InterfaceId	HW_ANY	64	Hardware ID of the Ethernet interface of the CPU
ID	CONN_OUC	1	Connection ID
ConnectionType	Byte	16#0B	Connection log (16#0B = TCP/IP)
ActiveEstablished	Bool	true	Active connection establishment (if client “true”)
RemoteAddress	IP_V4		IP address of SENTRON PACs as array of byte
ADDR	Array[1..4] of Byte		
ADDR[1]	Byte	192	Octet of IP address
ADDR[2]	Byte	168	Octet of IP address
ADDR[3]	Byte	0	Octet of IP address
ADDR[4]	Byte	16#0	Octet of IP address
RemotePort	UInt	502	IP port number of Modbus server
LocalPort	UInt	0	The value has to be 0 for a MB_CLIENT connection.

The connection data is identical for all SENTRON PACs with the exception of 4. Octets of IP address that are read before establishing a connection (“Tags”.device[n].ipOctet4).

For the differentiation of IP addresses in the first to third octet, you have to copy the octets of the appropriate SENTRON PACs before calling "MB_CLIENT" into the connection data (see IP address transfer) and adjust the subnet masks of the nodes to the network class (see Class C network).

Instance data of MB_CLIENT

The instance data of FB "MB_CLIENT" is stored as multi-instance in DB "InstSet22". The following default settings were made:

Table 3-14: Default value adjustment of the static instance data “instMbClient” in FB "Set22"

Name	Data type	Default value	Comment
Blocked_Proc_Timeout	Real	0.5	Duration (in seconds) for which it should be waited for a blocked Modbus client instance, before this instance is removed as ACTIVE (minimum 0.5 s).
Rcv_Timeout	Real	0.5	Time in seconds, which MB_CLIENT waits for the answer of a server to a request (minimum 0.5 s).
MB_Unit_ID	Byte	1	Modbus device identification (1..247 for SENTRON PACs) see \10\

Select the times “Blocked_Proc_Timeout” and “Rcv_Timeout” smaller than the interval between two SENTRON PACs connection calls (“InstSet22”.statScanInterval).

Note Further information on the "MB_CLIENT" instruction is available in the STEP 7 online help or the S7-1200 system manual (\3\).

3.5 "Tags" data block

The DB "Tags" includes the specific data of the SENTRON PAC3200 measuring devices and the interface tags for the HMI connection.

Table 3-15: "Tags" data block

Name	Data type	Comment
device	Array[1..2] of "typeDevice" (Table 3-3)	Array with the data of the SENTRON PAC measuring device (Upper limit of the array specifies the number of SENTRON PACs)
hmi	"typeHmi" (Table 3-6)	PLC data type of HMI tags

3.6 Error messages

Error messages are displayed on the operator panel in the "02_Alarm monitoring" screen in the form of a message buffer. The messages are not configured to require acknowledgement. A message is entered into the message buffer as "incoming" (I = incoming) and as "outgoing" (O = outgoing). The following messages are configured:

- "Defective illuminant PAC 3" ("Tags".device[n].bitAlarms.%X2)
  The "incoming" message is generated if the active power measured ("Tags".device[n].pAverage) by a SENTRON PAC 3200 falls below a configurable limit ("Tags".device[n].pavLimit.low), which indicates a defective lamp.

  If the limit is exceeded again, the program enters a respective "outgoing" message into the message buffer.
- "Short circuit PAC n" ("Tags".device[n].bitAlarms.%X3)
  The "incoming" message is generated if the active power measured ("Tags".device[n].pAverage) by a SENTRON PAC 3200 exceeds a configurable limit ("Tags".device[n].pavLimit.high), which indicates a short-circuit in the monitored branch.

  If the limit is fallen short again, the program enters a respective "outgoing" message into the message buffer.
- "Connection error PAC n" ("Tags".device[n].bitAlarms.%X0)
  The "incoming" message is generated in those program steps in which the communication sequence waits for a response of MB_CLIENT (e.g. DONE, CONNECTED or NOT BUSY) and instead, however, a communication request pulse ("nextPacPulse") for the next SENTRON PAC3200 device arrives. Specify the interval for this pulse via the static "InstSet22".statScanInterval tag.

  The respective "outgoing" message is created as soon as the MB_CLIENT signals DONE for the respective measuring device regarding a communication request.
- "Connection breakdown PAC n" ("Tags".device[n].bitAlarms.%X1)
  While each communication breakdown is reported for "Connection error PAC

n" – even if only a single value could not be transferred, the program only generates an "incoming" message for "Connection breakdown PAC n" if not a single communication request to the appropriate SENTRON PAC3200 device could be satisfied over a configurable period of ("InstSet22".instSequencer.statCommErrMsgDelay). In this application example, 5min have been set as the default period of time. The purpose of this message is discussed below.

- For the first successfully completed communication after a communication breakdown, the program determines the last consumed energy quantum at the just processed measuring point. It results from the product of the current actual power value and the difference between the current operating hours counter value and its reference value. The reference value is the last counter value before communication breakdown. For the time of communication breakdown an energy quantum is assumed which would result if the active power would be constant during the communication breakdown. For a short[3] communication breakdown, the substitute value of the energy quantum falsifies the consumed total energy only marginally, while the impact of a long[3] communication breakdown can considerably affect the precision of the total energy.

The respective "outgoing" message is created as soon as the MB_CLIENT signals DONE for the respective measuring device regarding a communication request.

- "Recipe transmission error" ("Tags".hmi.globalBitAlarms.%X0)
  For the recipe transmission with synchronization, a status information ("Tags".hmi.dataRecord[3]) is stored in the data area to which the "Data record" area pointer points. If it has value 4, the transmission was completed successfully without errors. If it has the value 12, a transmission error has occurred which is reported ("incoming" message). If no status message 4 or 12 is generated, the program also assumes a transmission error after 5 seconds ("RECIPE_ERROR_TIME" constant in FB "Set22").

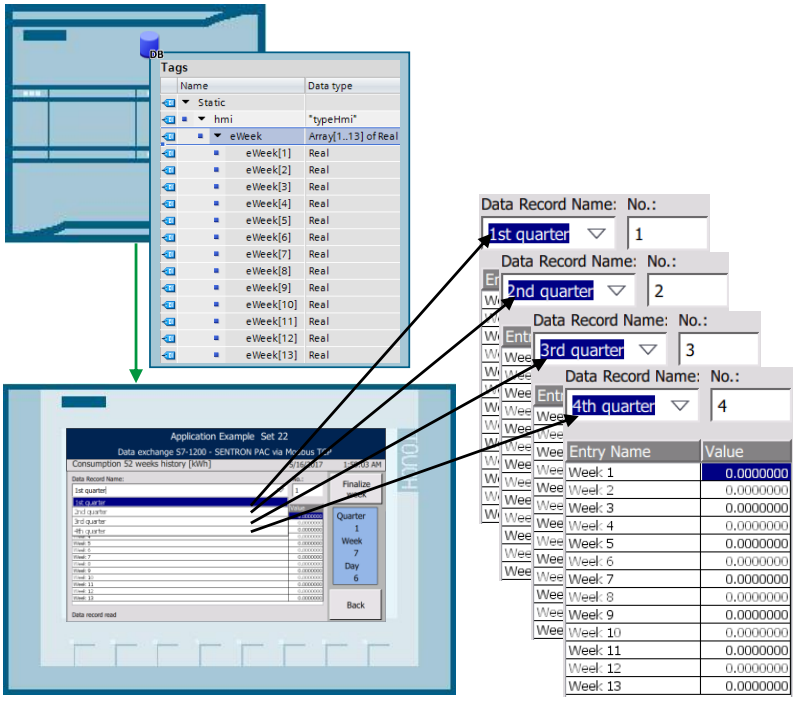
  The respective "ingoing" message is created if the transmission was completed without errors (status = 4).

3.7 Recipe management

Usually, recipe data are managed by the user in the operator panel and supplied to the control program on demand. This application uses the "Recipes" mechanisms to export data from the PLC to the operator panel and archive them to create storage space in the PLC. The energy consumption values are written to the HMI device for the entire year in weekly intervals via control job (Job mailbox).

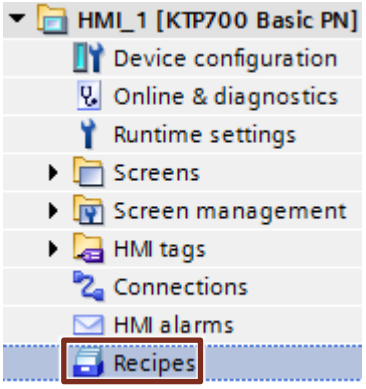
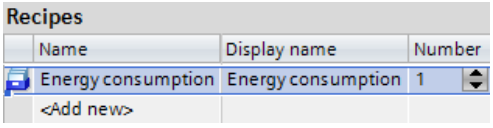
[3] The terms "short" and "long" must be considered in the context of the dynamics of the measured variable.

Figure 3-12: Storing energy consumption as recipes in the HMI device

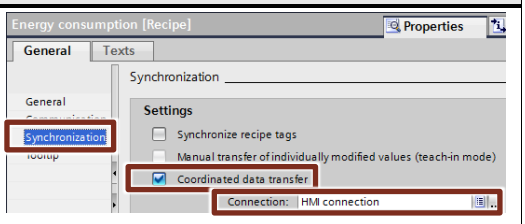
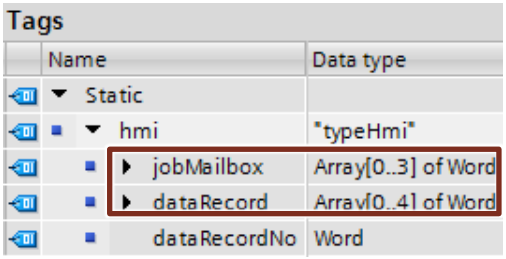
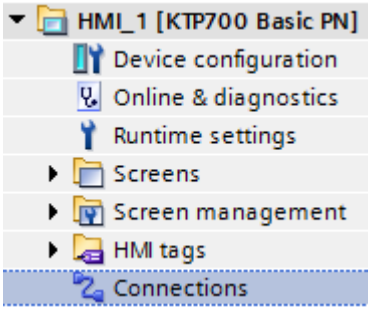
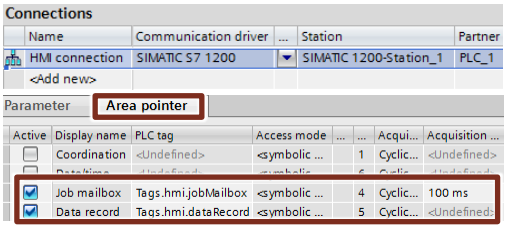


The following instruction[4] shows how to transfer synchronized recipe data records from the controller into the operator panel.

Table 3-16: Synchronized transfer of recipe data records from the PLC into the operator panel

No.	Instruction	Comment/screen
1.	Open the “Recipes” editor in the operator device folder of your project.	
2.	Select the recipe to be synchronized and open their properties, using the key combination ALT+ ENTER.	

[4] The individual steps have already been implemented in the example project.

No.	Instruction	Comment/screen
3.	• In menu item “Synchronization” tick “coordinated transfer of data records”. • Enter the connection to the controller with which the operator panel synchronizes the transfer in “Connection”. For the transfer with synchronization, you avoid uncontrolled mutual overwriting of data in your control program.	
4.	Create two arrays in one DB of the CPU (here in “Tags”.hmi): • jobMailbox Array[0..3] of Word (for “**jobMailbox**” area pointer) • dataRecord Array[0..4] of Word (for "**dataRecord**" area pointer)	
5.	Open the connections of the operator panel.	
6.	• Select the connection via which the control job is to be executed. • Open the “Area pointers” menu item. • Enable the “Job mailbox” area pointer and select the previously created job mailbox array as tag. • Enable the “Data record” area pointer and select the previously created data record array as tag.	

<table>
<tr><th>No.</th><th>Instruction</th><th>Comment/screen</th></tr>
<tr><td>7.</td><td>

- The “**Job mailbox**” array consists of 4 words (0 to 3). The least significant byte of the 0 word contains the job number.
- Job number 69 is the function for reading a data record from the controller.
- The 1st word (parameter 1) contains the recipe number. Since there is only one recipe in the application, this parameter is set to 1 in the program code (jobMailbox[1] = 1).
- The 2nd word (parameter 2) contains the data record number. The tag which determines the quarter, is transferred in the program code. (jobMailbox[2] = #dataRecordNo).

</td><td>

<table>
<tr><th>Word</th><th>Most significant byte</th><th>Least significant byte</th></tr>
<tr><td>n+0</td><td>0</td><td>Job number</td></tr>
<tr><td>n+1</td><td colspan="2">Parameter 1</td></tr>
<tr><td>n+2</td><td colspan="2">Parameter 2</td></tr>
<tr><td>n+3</td><td colspan="2">Parameter 3</td></tr>
</table>

<table>
<tr><th>No.</th><th>Function</th><th></th></tr>
<tr><td>69</td><td colspan="2">Reading data record from PLC</td></tr>
<tr><td></td><td>Parameter 1</td><td>Recipe number (1-999)</td></tr>
<tr><td></td><td>Parameter 2</td><td>Data record number (1-65535)</td></tr>
<tr><td></td><td>Parameter 3</td><td>0: Do not overwrite existing data record
1: Overwrite existing data record</td></tr>
</table>

</td></tr>
<tr><td></td><td colspan="2">

- The 3rd word (parameter 3) specifies whether an existing data record is overwritten or not. Existing data records are overwritten in the program (jobMailbox[3] = 1).
- The overwriting of the respective recipe data record is in the example project triggered for each change of the week (see SCL code in FB “Set22”).

```
REGION 11 Transmit recipe to operator panel
  IF (#statWriteRecipe = TRUE) THEN

      // recipe number
      #hmi.jobMailbox[1] := 1;

      // quarter
      #hmi.jobMailbox[2] := #hmi.dataRecordNo;

      // 1=overwrite
      #hmi.jobMailbox[3] := 1;

      // reading data record from PLC
      #hmi.jobMailbox[0] := 69;
```

</td></tr>
</table>

<table>
<tr><th>No.</th><th>Instruction</th><th>Comment/screen</th></tr>
<tr><td>5.</td><td>

The “Data record” array consists of 5 words (0 to 4). The 4th word (#dataRecord[3]) specifies the current status of the recipe transfer.
The provides information of whether the recipe transfer was completed without error. More details and the further procedure can be found in “Transferring recipe data into HMI” control job.
Without this synchronization there would be no confirmation that the data record was really transferred by the controller into the recipe of the operator panel.

</td><td>
<table>
<tr><th></th><th>15</th><th>0</th></tr>
<tr><td>1. Word</td><td colspan="2">Current recipe number (1 - 999)</td></tr>
<tr><td>2. Word</td><td colspan="2">Current data record number (0 - 65535)</td></tr>
<tr><td>3. Word</td><td colspan="2">Reserved</td></tr>
<tr><td>4. Word</td><td colspan="2">Status (0, 2, 4, 12)</td></tr>
<tr><td>5. Word</td><td colspan="2">Reserved</td></tr>
</table>
Status
The status word (word 4) can adopt the following values:
<table>
<tr><th colspan="2">Value</th><th rowspan="2">Meaning</th></tr>
<tr><th>Decimal</th><th>Binary</th></tr>
<tr><td>0</td><td>0000 0000</td><td>Transfer permitted, data record free</td></tr>
<tr><td>2</td><td>0000 0010</td><td>Transfer is busy</td></tr>
<tr><td>4</td><td>0000 0100</td><td>Transfer completed without error</td></tr>
<tr><td>12</td><td>0000 1100</td><td>Transfer completed with error</td></tr>
</table>
</td></tr>
</table>

Note

Further information on the topic Data exchange with area pointer (Basic Panels) can be found in \5\.

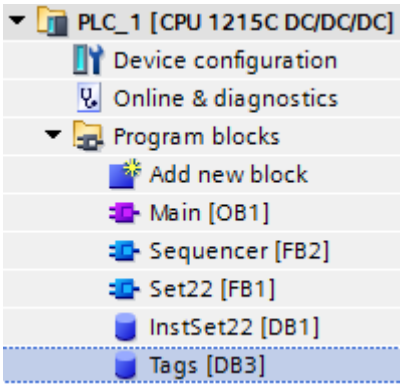
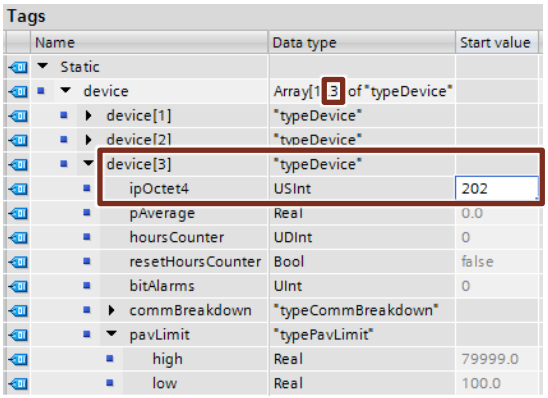
4 Expanding the Number of Measuring Points

To be able to run the example program with more than two SENTRON PAC3200 measuring devices, you require a switch with more Ethernet ports to which you can connect additional measuring devices. Configure them according to chap. 5.1. Use those values as 4th octet of the IP address which you enter in "Tags".device[n].ipOctet4 in chap. 4.1.

The following example shows the expansion to three SENTRON PAC3200 devices.

4.1 Adjusting the control program part

Table 4-1

No.	Instruction	Comment/screen
1.	Open the DB "Tags" by double-clicking the block in the project tree.	
2.	Expand the “device” array to three elements.	
3.	Assign the 4th octet of the IP address to the newly added SENTRON PAC3200 (here: “202”).	
4.	optional: Enter the desired power limits of the added measuring point (pavLimit) as start values. See also block "Specifying the power limits in the configuration (optional)" in chap. 6.4.	
5.	To apply the changes, save, compile and load the DB “Tags".	

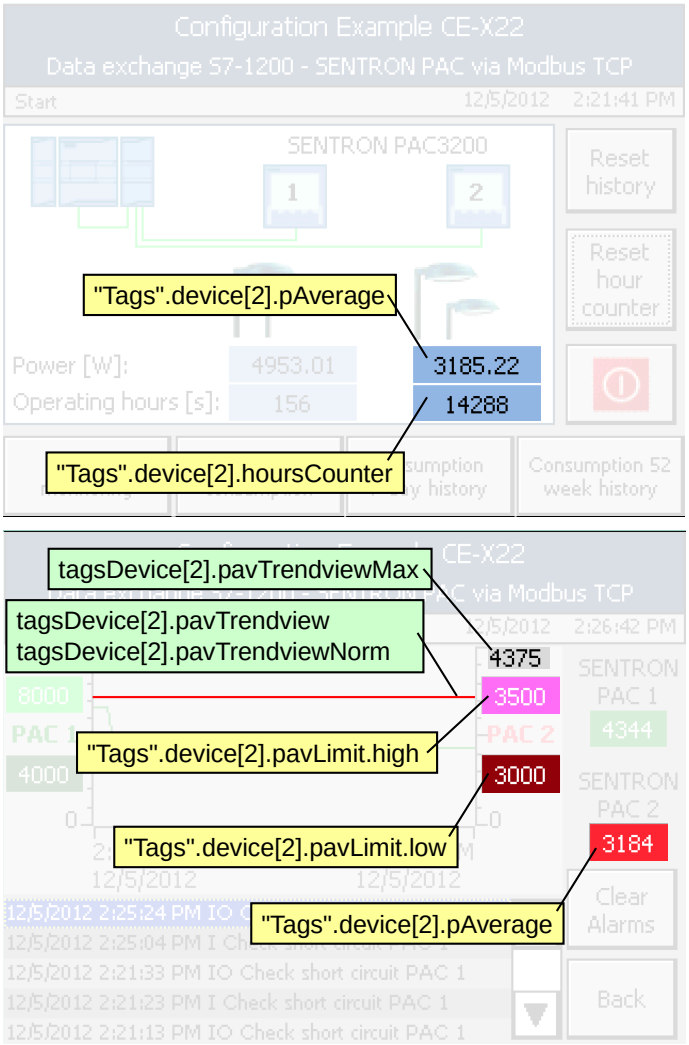
4.2 Adjusting the HMI program part

A clear representation of more than two SENTRON PAC3200 devices would require a larger operator panel screen and/or a completely different screen configuration. Since this application does not focus on the operator panel, it is only explained here how to access further tags required for the display on a further measuring device.

Tags in the HMI screens

Figure 4-1 lists the tags of a SENTRON PAC3200 required for the HMI display (using the example of device 2). The tags with yellow names are PLC tags. For those, you need to create respective HMI tags for device 3 (also with index 3). The tags with green names are internal HMI tags, which you copy and, if necessary, must also label with index 3. At a value change, some HMI tags to be copied come with respective events, which in return are described by further HMI tags (through “linear scaling” in the “Events” tab, for example). Please make sure you also adjust the indices to 3 for these HMI tags.

Figure 4-1: Device-specific tags



Tags in the message display

Supplement the message display by three bit alarms

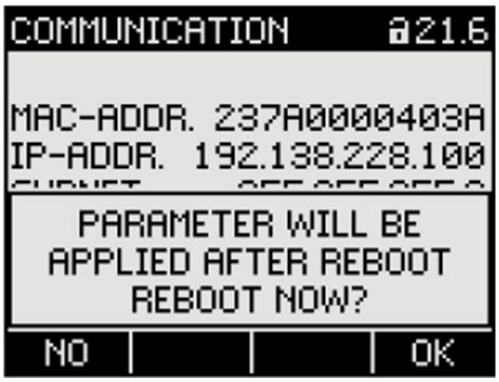
- "Connection error PAC 3" (trigger bit 0),
- "Connection breakdown PAC 3" (trigger bit 1)
- "Defective illuminant PAC 3" (trigger bit 2),
- "Short circuit PAC 3" (trigger bit 3),

In HMI devices in the project tree go to “HMI alarms > Discrete alarms”. Create the respective trigger tag with index 3 and connect it to PLC tag Tags.device[3].bitAlarms

5 Configuration and Settings

5.1 Configuring the SENTRON PAC3200 devices

Table 5-1: Configuring the SENTRON PAC3200

No.	Instruction	Comment/screen
1.	Wire the power side of the SENTRON PAC3200 devices according to their instruction manual (\9\) and the power level of the measuring point. Do not operate the devices without current transformer.	
2.	Supply the voltage	
3.	Configure the devices for the power unit according to the instruction manual (\9\) and the power level of the measuring point.	
4.	Configure the devices as follows with regards to communication: • Open the main menu of the SENTRON PAC3200 device via F4. • Navigate to “Settings”, using F2 or F3 and open it with F4. • Navigate to the “Communication” menu item, using F2 or F3 and open it with F4. • Make the following settings: - IP address: 192.168.0.xxx - Subnet mask: 255.255.255.0 - Gateway: 0.0.0.0 - Protocol: MODBUS TCP For the 4th octet of the IP address please use 200 and 201. • After completing all settings, click F1 and confirm the new start request with F4.	 

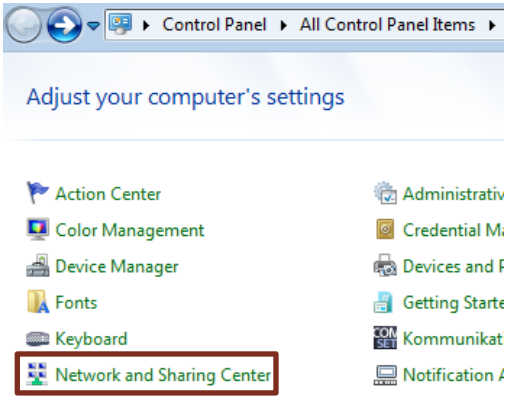
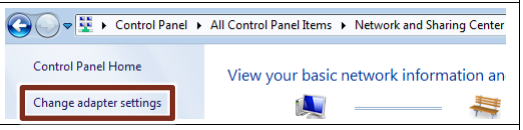
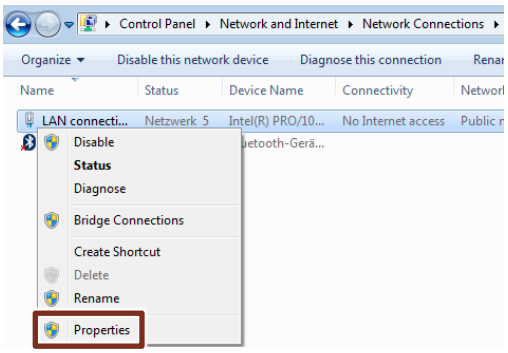
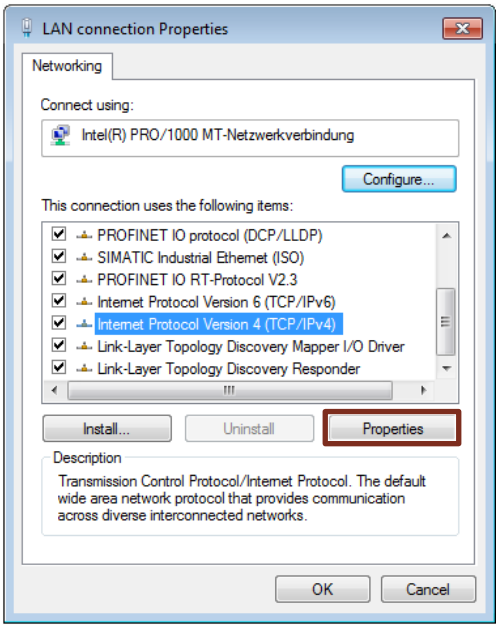
5.2 PG/PC settings on Windows level

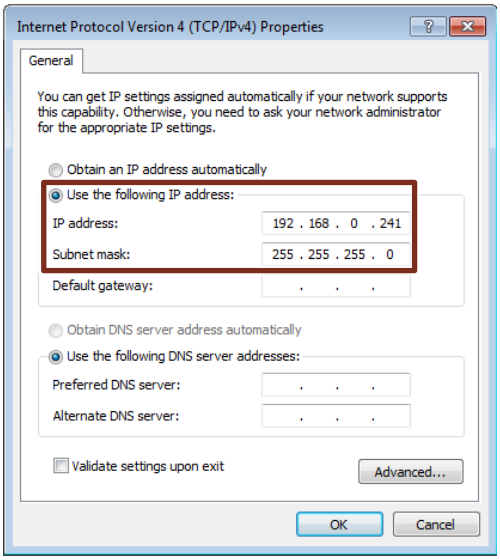
5.2.1 Assigning the IP address of the PG/PC

You have to assign your PG/PC an IP address in the same subnet as the CPUs and the optional HMI. The IP addresses of the individual nodes can be viewed in Figure 6-1.

In order to assign the IP address for your network card in the Windows 7 operating system, please proceed as follows:

Table 5-2: Assigning the IP address of the PG/PC

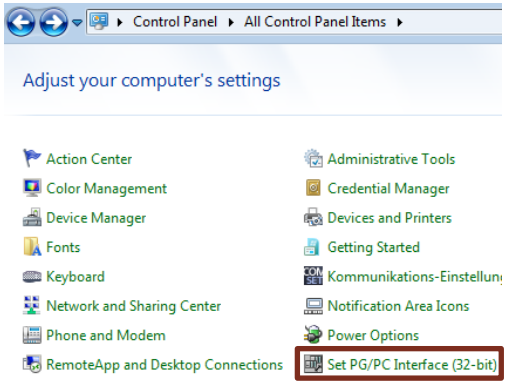
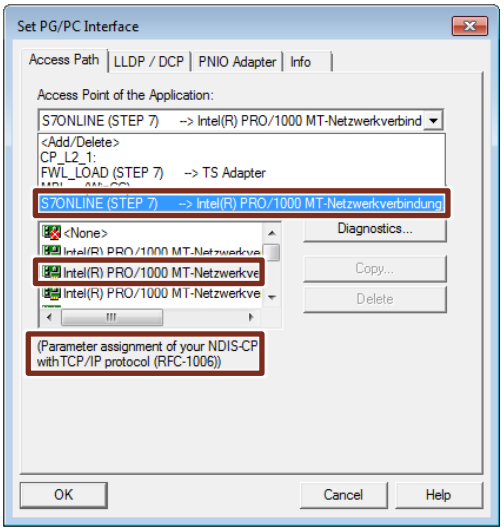
No.	Instruction	Comment/screen
1.	Open the “Network and Sharing Center” in the Windows control panel.	
2.	Select the “Change adapter settings” menu item.	
3.	Select the network card to be used and open the properties by right clicking them.	
4.	Select the “Internet Protocol Version 4 (TCP/IPv4)” item and open its properties.	

No.	Instruction	Comment/screen
5.	• Select “Use the following IP address” • Enter “192.168.0.241”, for example, as IP address (see Figure 6-1). (However, you can use any free address in the respective subnet) • Enter “255.255.255.0” in Subnet mask. • Click on "OK" to confirm the settings.	

5.2.2 Setting PG/PC interface

To simulate the KTP700 in the TIA Portal, the PG/PC interface must be set accordingly. Proceed as follows:

Table 5-3: Setting PG/PC interface

No.	Instruction	Comment/screen
1.	Open the “Set PG/PC Interface” dialog box in the Windows Control Panel.	
2.	• At the access point of the “S7ONLINE (STEP 7)” application, select the interface parameter assignment used for your network card ("…TCP/IP protocol (RFC-1006)") used. • Click on "OK" to confirm the settings.	

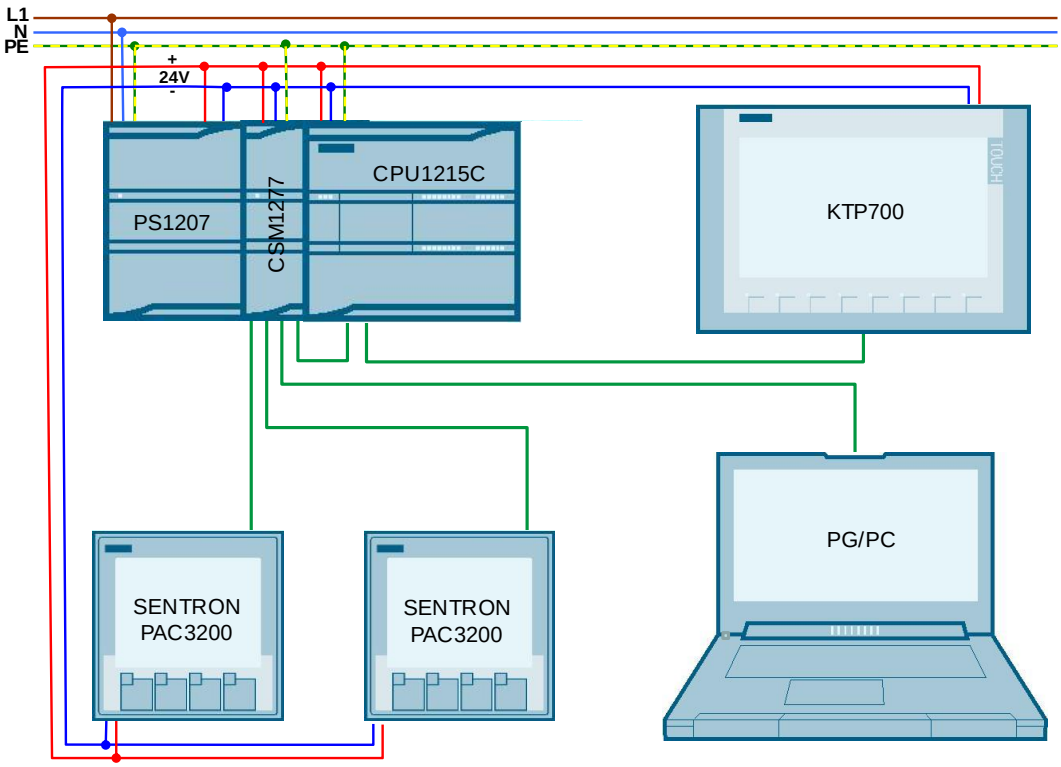
6 Installation and Commissioning

6.1 Installing the hardware

Install and wire the components (see chapter Table 2-1) according to the specifications in manuals \3\, \6\, \7\, \8\, \9\, any enclosed instruction sheets and the wiring plan (Figure 6-1). Earth all of the devices.

The figure below shows the hardware configuration of the application:

Figure 6-1



You can use the KTP700[5] as the operator panel or the simulation in the TIA Portal.

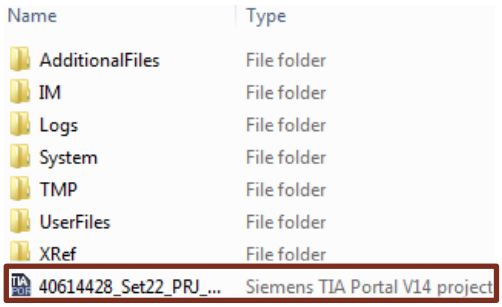
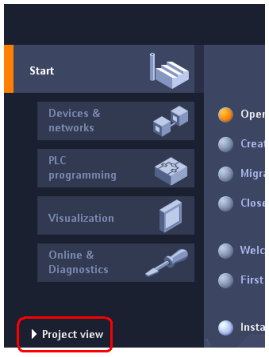
Note

The various connection types for the power side of the SENTRON power monitoring device PAC3200 can be found in the device manual \9\.

[5] When using a CPU with only one LAN connection, only connect the KTP700 when you no longer require the development system for configuration and testing.

6.2 Opening the project in the TIA portal

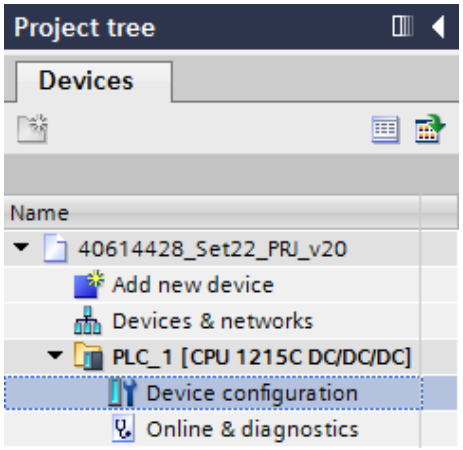
Table 6-1: Opening the project in the TIA portal

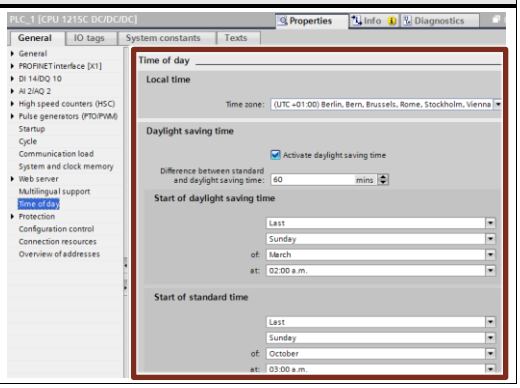
No.	Instruction	Comment/screen
1.	Unzip the archived STEP 7 project (.zip) into any directory on your PG/PC.	
2.	Navigate to your unzipped project folder in the Windows Explorer, open the folder and start TIA Portal by double-clicking on the contained *.ap1x file.	
3.	The project opens in the TIA portal. If the TIA Portal has opened in the portal view, go to the portal view (depending on the default setting in the TIA Portal).	

6.3 Configuring the PLC local time

The system time of the KTP700 shall be synchronized with the PLC local time every minute. It must be derived from the PLC system time regarding summer/winter time. Make the respective settings in the “Properties” of the CPU – as explained below.

Table 6-2: Configuring the PLC local time

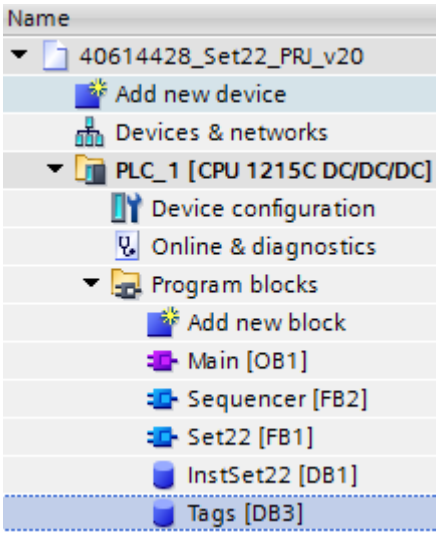
No.	Instruction	Comment/screen
1.	In the project tree, open the “Device configuration” in “PLC_1” with a double-click.	

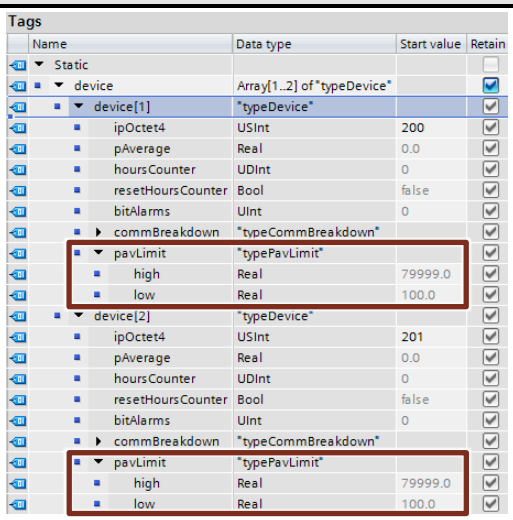
No.	Instruction	Comment/screen
2.	• Open the menu item “Time of day” in “Properties”. • Select your “Time zone” for calculating the local time. • If necessary, enable the daylight saving time with all required settings.	

6.4 Defining the power limits in the configuration (optional)

The power limit values for the short-circuit and defective illuminate detection can be entered conveniently on the operator panel and are then saved retentively in the PLC. If you do not need this functionality, but want the correct power limit values to be written retentively to the controller during program download already, enter these as start values into the DB “Tags” prior to loading the program. This has the advantage, that the power limit values are available again after exchanging a CPU, for example.

Table 6-3: Defining the power limits in the configuration

No.	Instruction	Comment/screen
1.	Open the DB "Tags" by double-clicking the block in the project tree.	

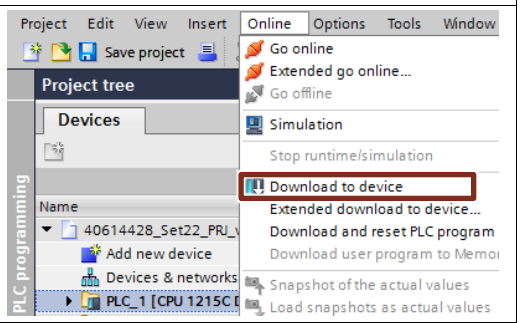
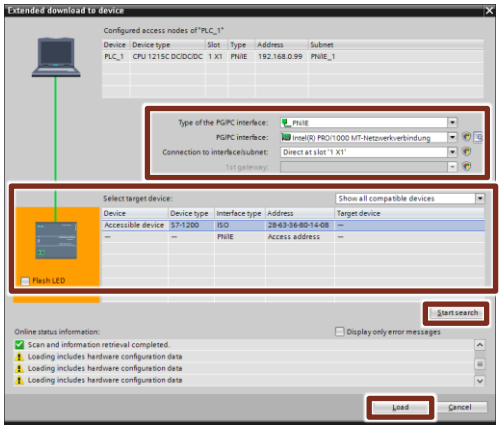
No.	Instruction	Comment/screen
2.	Enter the power limits for the configured measuring points (individually for each SENTRON PAC as start value DB or identical for all, via the PLC data type “typePavLimit”).	

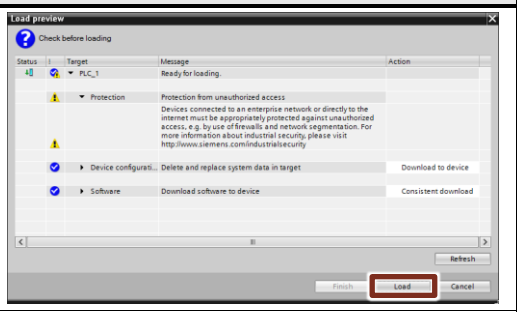
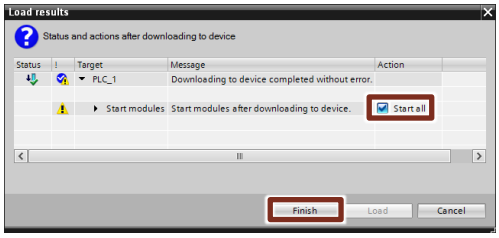
Tags

Name	Data type	Start value	Retain
Static			
device	Array[1..2] of "typeDevice"		☑
device[1]	"typeDevice"		☑
ipOctet4	USInt	200	☑
pAverage	Real	0.0	☑
hoursCounter	UDInt	0	☑
resetHoursCounter	Bool	false	☑
bitAlarms	UInt	0	☑
commBreakdown	"typeCommBreakdown"		☑
pavLimit	"typePavLimit"		☑
high	Real	79999.0	☑
low	Real	100.0	☑
device[2]	"typeDevice"		☑
ipOctet4	USInt	201	☑
pAverage	Real	0.0	☑
hoursCounter	UDInt	0	☑
resetHoursCounter	Bool	false	☑
bitAlarms	UInt	0	☑
commBreakdown	"typeCommBreakdown"		☑
pavLimit	"typePavLimit"		☑
high	Real	79999.0	☑
low	Real	100.0	☑

6.5 Loading the control unit into the CPU

Table 6-4: Loading the control unit into the CPU

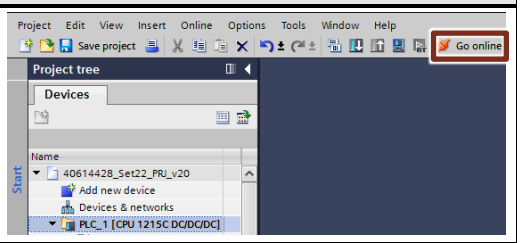
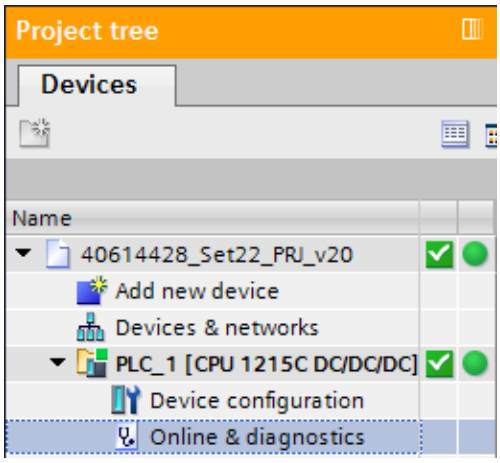
No.	Instruction	Comment/screen
1.	Connect your PG/PC to the CPU directly or using the CSM1277 switch. (see Figure 6-1).	
2.	• Select the “PLC_1 [CPU 1215C DC/DC/DC]” PLC folder. • Download the controller project part into the CPU via “Online > Download to device”.	
3.	• Select the type of the PG/PC interface used, the network card and the subnet. • Enable “Show all compatible devices”. • Use the “Start search” button to start the search. • Identify your controller from the list of accessible devices using the MAC address or the “Flash LED” option. • Select the selected controller and click the “Load” button.	

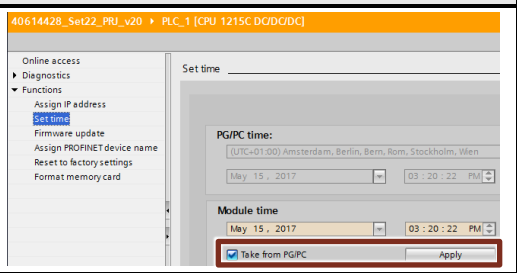
No.	Instruction	Comment/screen
4.	• Select the “Load” button in the “Load preview” screen.	
5.	After transferring all program blocks into the controller, the “Load result” window appears. • Select “Start all” to set the controller to “Run” mode. • Select the “Finish” button to finish the entire download.	

6.6 Setting the time of the controller

To synchronize the KTP700 or its simulation in TIA Portal with the local time of the controller, the time of the CPU has to be set. Proceed as follows:

Table 6-5: Setting the time of the controller

No.	Instruction	Comment/screen
1.	Select the controller in the project tree and press the “Go online” button.	
2.	In the project tree of the controller, double-click the “Online & diagnostics” menu item.	

No.	Instruction	Comment/screen
3.	• Select the menu item “Functions > Set time”. • Adopt the PG/PC time to your controller (tick “Take from PG/PC”) or enter the date and time manually (no tick at “Take from PG/PC”). • Press the “Apply” button.	

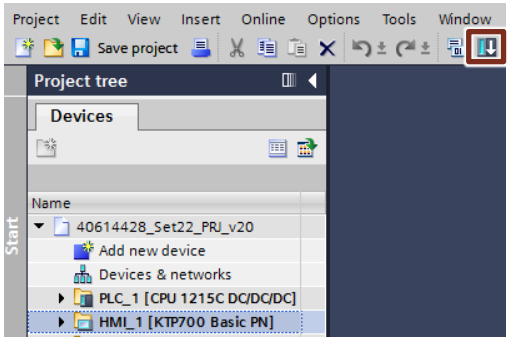
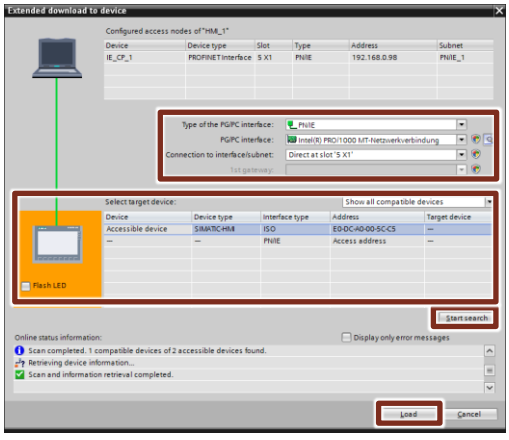
Note

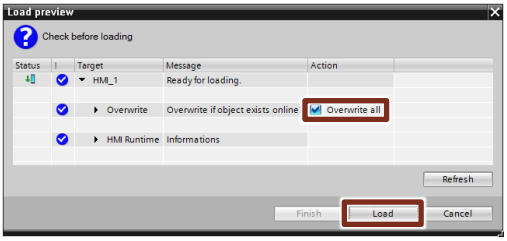
Please note that the times displayed are the local times of the PG and the CPU. The CPU system time arises from the resulting offset of the configured time settings (Configuring the PLC local time).

6.7 Loading the HMI project part into the KTP700

If the KTP700 is used as operator panel, then the HMI project part has to be loaded to the KTP700.

Table 6-6: Loading the HMI project part into the KTP700

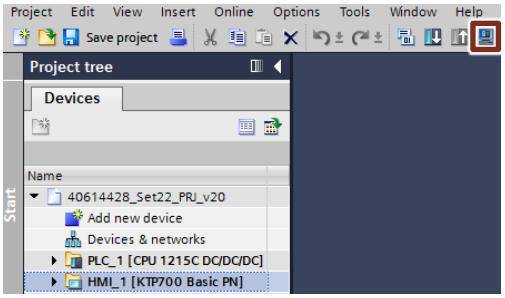
No.	Instruction	Comment/screen
1.	For the transfer, connect your PG/PC to the KTP700 either directly or using the CSM1277 switch.	
2.	• Select the operator panel folder “HMI_1 [KTP700 Basic PN]”. • Select the “Download to device” button to download the HMI project part to the KTP700.	
3.	• Select the type of the PG/PC interface used, the network card and the subnet. • Enable “Show all compatible devices”. • Use the “Start search” button to start the search. • Identify your operator panel from the list of accessible devices using the MAC address or the “Flash LED” option. • Select the selected operator panel and click the “Load” button.	

No.	Instruction	Comment/screen
4.	• If necessary, enable the “Overwrite all” option in the “Load preview” screen. • Press the “Load” button.	

6.8 Starting the operator panel simulation on the PG/PC

If the PG/PC is to be used as operator panel, start the operator panel simulation as follows:

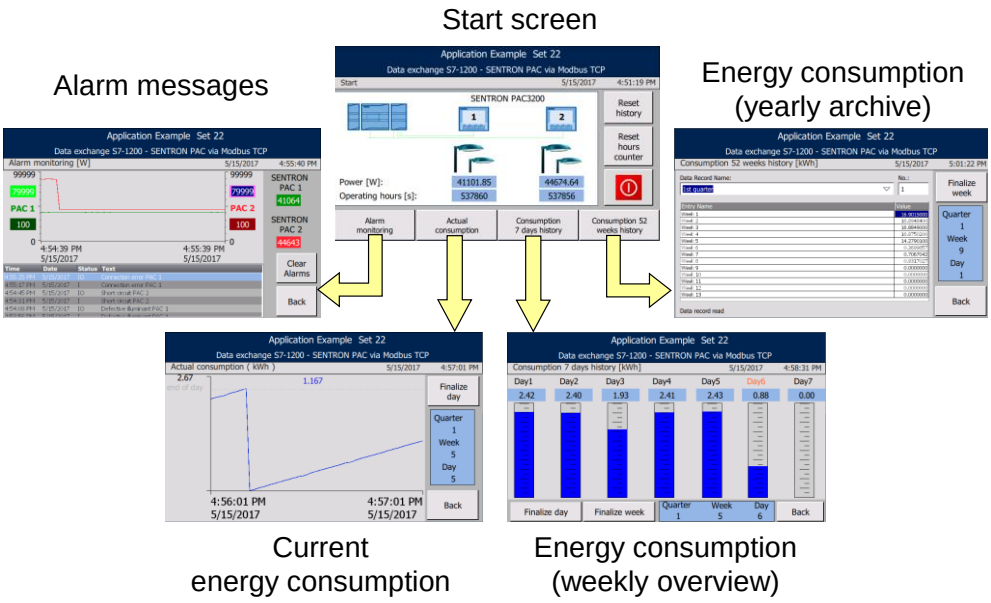
Table 6-7: Starting the operator panel simulation on the PG/PC

No.	Instruction	Comment/screen
1.	• Select the operator panel folder “HMI_1 [KTP700 Basic PN]”. • Click the “Start simulation” button.	

7 Operating the Application

7.1 Overview

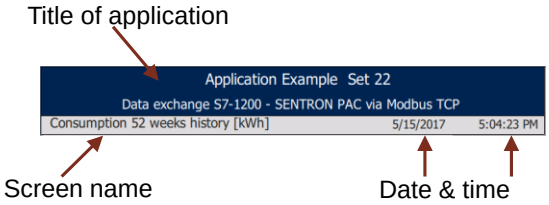
Figure 7-1: Menu navigation



The “Back” button takes you back from the respective screen to the start screen.

7.2 Header

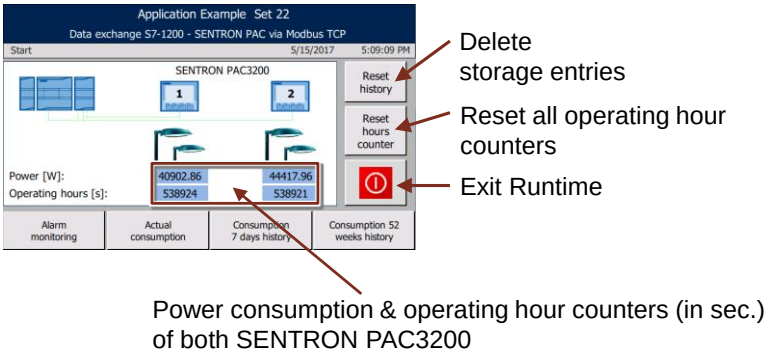
Figure 7-2: Header



Date and time of the operator panel are synchronized once every minute by the S7-1200 CPU

7.3 Start screen

Figure 7-3: Start screen

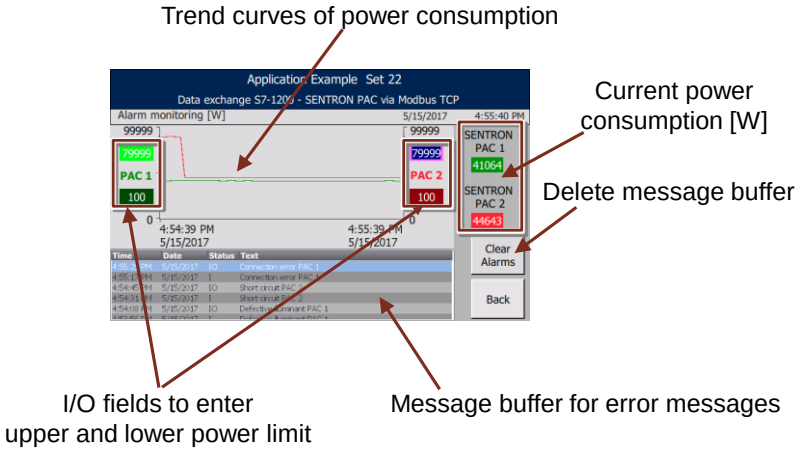


- ”Reset history” button
  Delete the following storage entries in the PLC:
  - Deleting the daily energy consumption values of the current week
  - Deleting the weekly energy consumption values of the current quarter
  - Resetting the archiving start to the first day of the first week in the first quarter.

  Any energy consumption already archived in the operator panel as recipes are not affected by the delete process. Use the function if you wish to start a new data archiving at the operator panel.
- "Reset hours counter” button
  Resetting the operating hour counters of all connected SENTRON PAC3200 devices. For measuring devices that cannot be reached, the reset request is stored and once they can be reached, the request is executed. As long as the reset process is carried out, i.e. as long as not all of the operating hour counters have been reset, the button lights up in yellow.
- Button
  Stop runtime. This will get you to the loader menu of the KTP700.
- Display of power consumption and operating hours counter
  The update of power consumption and operating hours counter occurs in 2s intervals ("InstSet22”.statScanInterval) in this application. This means that the communication with the next SENTRON PAC3200 measuring device takes place after 2 seconds. After all devices have been communicated with, communication starts again with device 1.

7.4 Alarm messages

Figure 7-4: “Alarm messages” screen



- **Coloration**
  Curves and values of the measuring point 1 (PAC 1) are displayed in shades of green, those of measuring point 2 (PAC 2) in shades of red.
- **Trend curves of power consumption**
  The measured active power values of both SENTRON PAC3200 devices are displayed. The left ordinate axis is the measure for device 1, the right one for device 2. The specified upper ordinate limits correspond each to 1.25 times the value of the upper power limit of the affected measuring point.
- **Current power consumption [W]**
  The values correspond to the trend curves at the right end of the x-coordinate.
- **I/O fields to input power limits**
  When exceeding the upper power limit, a short-circuit is identified for the affected measuring point, when the power limit falls below the lower power limit, a defective illuminant is assumed. Input values are stored retentively and are available again after controller restart or after a power cut.
- **Alarm buffer**
  The following messages are displayed with timestamp and identifier “incoming/outgoing”:
  - Short circuit PAC n (n=1..2)
    For simulating a short-circuit and outputting a respective error message, you can, for example, reduce the upper power limit for a measuring point during runtime, so the current active power value is above this limit.
  - Defective illuminant PAC n (n=1..2)
    For simulating a lamp failure and outputting a respective error message, you can, for example, reduce the lower power limit for a measuring point during runtime, so the current active power value falls below this limit.

- Connection error PAC n (n=1..2)[6]
  For simulating a communication error, pull the RJ45 connector from a SENTRON PAC3200 device, for example.

Connection breakdown PAC n (n=1..2)[6]
For simulating a longer connection failure that may lead to errors in determining the total energy consumption, pull the RJ45 connector from a SENTRON PAC3200 device for >5 minutes ("InstSet22".instSequencer.statCommErrMsgDelay), for example.

Note

For a communication breakdown with a SENTRON PAC3200 device, data is no longer transferred to the PLC. The data last received by the PLC is retained and hence also displayed at the operating panel in the trend view (horizontal bar) and as a value. The data is not displayed as zero.

Depending on the system, all above messages have a delay time. Since the messages are generated when the affected device is next in communication, there can be a maximum delay time of

```
"InstSet22".statNumberOfPacs * "InstSet22".statScanInterval
```

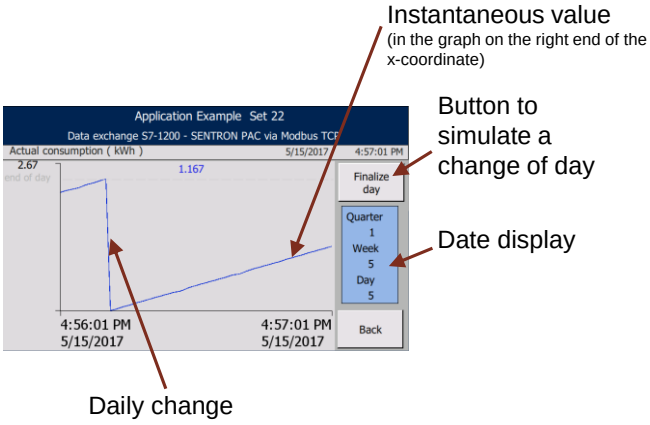
- i.e. 2s in this application.

- Recipe transmission error
  A recipe transfer error is, for example, generated if there is no communication connection with the operator panel at the time of the recipe transfer. If the communication is possible again, the recipe is then transferred to the operator panel. The “incoming” and “outgoing” messages are entered into the message buffer at the same time.

The content of the message buffer can be deleted with the “Clear Alarms” button.

7.5 Current energy consumption

Figure 7-5: Current energy consumption



[6] The difference between “Connection error” and “Connection breakdown” is explained in chap. 3.6.

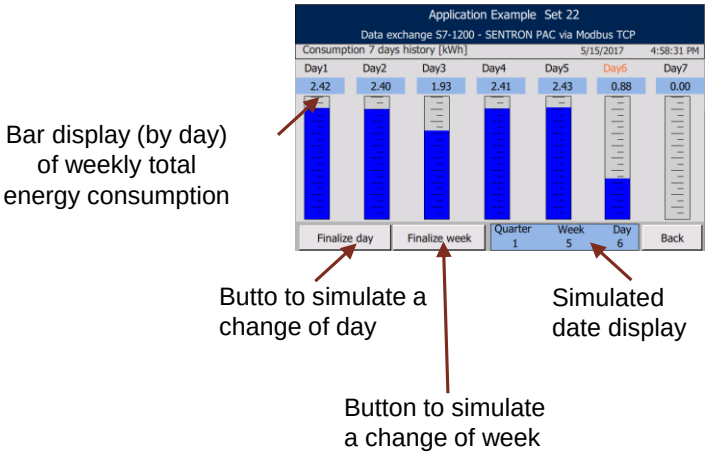
- Trend view
  Displays the total energy consumption of all consumers connected via the SENTRON PAC3200 devices. The upper ordinate limit automatically adjusts to the energy level. It corresponds to the energy quantity which would accumulate if all consumers (in the application example measuring points 1 and 2) were at their upper power limit for 24 hours (limit from when the short-circuit is detected).

  For an energy consumption which exceeds 90% of the upper ordinate limit, a daily change is automatically performed where the energy value is reset to 0.

  A change in the power consumption (e.g. adding or removing a consumer) can be immediately detected at the inclination of the trend curve.
- "Finalize day" button
  It enables the manual simulation of a change of day at any time.
- Date display
  The current date (related to the start of the recordings), to which the energy consumption trend relates, is specified as quarter (1..4), week (1..13) within the quarter and day (1..7) within the week. The date display is incremented as the day changes.

7.6 Weekly energy consumption

Figure 7-6: Weekly energy consumption

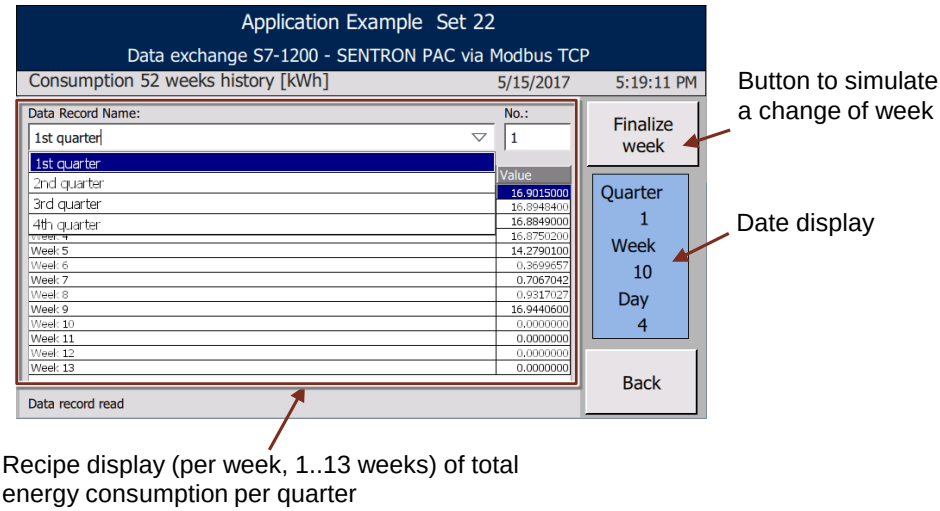


- Bar display
  Displays the daily total weekly energy consumption of all connected SENTRON PAC3200 devices in bar chart format. The current weekday is shaded in red. The upper bar limit is identical with the upper ordinate limit in Figure 7-5. The automatically daily change produced for the simulation only occurs according to the criteria in chap. 7.5.
- "Finalize day" button
  It enables the manual simulation of a daily change. The addition of the energy consumption for the current day is terminated and continued in the bar of the subsequent day.
- "Finalize week" button
  It enables the manual simulation of a weekly change. The addition of the energy consumption for the current week is terminated and continued on day 1 of the subsequent day. Furthermore, the total energy consumption (weekly values) of the current quarter is archived in the operator panel as a recipe.

- Date display
  As shown in chap. 7.5.

7.7 Recipe display of the total energy consumption

Figure 7-7: Recipe display of the total energy consumption



- Recipe display
  In contrast to the other screens, the recipe display shows the energy consumption values already transferred into the operator panel. These are subdivided into four quarters. Each quarter shows the total energy consumption values of 13 weeks. In the first level of the recipe display, select the quarter ("Data Record Name:") and this takes you to the desired weekly values. The controller writes the recipe entries into the operator panel in the event of a change of week. The display is updated if a quarter is selected.
- "Finalize week" button
  It enables the manual simulation of a weekly change. When clicked, the quarter selection is updated. However, since the recipe transmission will occur time-delayed, the display will always be one week behind.
  Execute the quarter selection independently, to update the recipe display.
- Date display
  As shown in chap. 7.5.

8 Links & Literature

Table 8-1

No.	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Download page of the entry https://support.industry.siemens.com/cs/ww/en/view/40614428
\3\	Manual "SIMATIC S7 S7-1200 Automation system" https://support.industry.siemens.com/cs/ww/en/view/109741593
\4\	FAQ “How do you program the overlaying of tags with the keyword "AT" in the STEP 7 (TIA Portal)?” https://support.industry.siemens.com/cs/ww/en/view/57132240
\5\	Manual "STEP 7 Basic V14" https://support.industry.siemens.com/cs/ww/en/view/109742266
\6\	Operating Instructions S7-1200 Compact Switch Module CSM 1277 https://support.industry.siemens.com/cs/ww/en/view/36087313
\7\	Operating Instructions PM1207 https://support.industry.siemens.com/cs/ww/en/view/37316256
\8\	Operating instructions “SIMATIC HMI HMI devices Basic Panels 2nd Generation” https://support.industry.siemens.com/cs/ww/en/view/90114350
\9\	Manual SENTRON Power Monitoring Device PAC3200 https://support.industry.siemens.com/cs/ww/en/view/26504150
\10\	FAQ "How do you clear the status 16#8382 in the case of a Modbus/TCP connection between SIMATIC S7-1500 / S7-1200 and SENTRON PAC devices?" https://support.industry.siemens.com/cs/ww/en/view/109736516
\11\	Application example “Using recipes with S7-1200” https://support.industry.siemens.com/cs/ww/en/view/94681612
\12\	Application Example “Energy Data Acquisition with the Energy Meter of the S7-1200” https://support.industry.siemens.com/cs/ww/en/view/109739414
\13\	Manual “SIMATIC WinCC WinCC Basic V14” https://support.industry.siemens.com/cs/ww/en/view/109742284

9 History

Table 9-1

Version	Date	Modifications
V10.5_1.5_0	10.06.2011	First publication for • STEP 7 Basic V10.5 SP2 with CPU firmware V1.0.2, MODBUS/TCP library V1.5 for SENTRON PAC3200
V11.0_1.5_0	10.06.2011	First publication for • STEP 7 Basic V11.0 update 1 with CPU firmware V2.0, MODBUS/TCP library V1.5 for SENTRON PAC3200
V11.0_1.6_0	23.09.2011	Revised publication for • STEP 7 Basic V11.0 SP1 update 2 with CPU firmware V2.1.2, MODBUS/TCP library V1.6 for SENTRON PAC3200
V11.0_1.7_0	02.01.2013	Revised publication for • STEP 7 Basic V11 SP2 update 5 as of CPU firmware V2.1.2, with system FB MB_CLIENT, without MODBUS/TCP library
V12.0_1.7_0	23.05.2013	Revised publication • Supplemented by project in TIA Portal V12.
V14.0	12.06.2017	Revised publication for • STEP 7 Basic V14.0 update 2, MODBUS/TCP library V4.1, S7-1200 CPU firmware V4.2, SENTRON PAC3200 firmware V2.4.2