Application Example • 04/2016

Using Recipe Functions for persistent Data

TIA Portal, S7-1200, S7-1500

https://support.industry.siemens.com/cs/ww/en/view/109479727

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

Warranty and liability 2

1 Task 4

1.1 Overview 4
1.2 Requirements of the automation task 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 Fundamentals of power supply system 9

3.1 Main unit SITOP PSU8600 9
3.2 Expansion module SITOP CNX8600 9
3.3 Buffer module SITOP BUF8600 9
3.4 Engineering 10
3.5 Cyclic input and output data 10

4 Mode of Operation 13

4.1 General overview 13
4.2 Global data block “StationData” 16
4.3 Global data block “ActData” 17
4.4 Global data block “ActDataLoad” (only S7-1200) 19
4.5 Function block “ReadWriteCSV” 20
4.6 Function block “DBtoCSV” 23
4.7 Function block “CSVtoDB” 25

5 Installation and Commissioning 27

5.1 Installing the hardware 27
5.2 IP addresses and device names 28
5.3 Installing the software (download) 28
5.4 Loading the project 29

6 Operating the Application Example 30

6.1 Overview 30
6.2 Operation 31

7 Links & Literature 32

8 History 32

1 Task

1.1 Overview

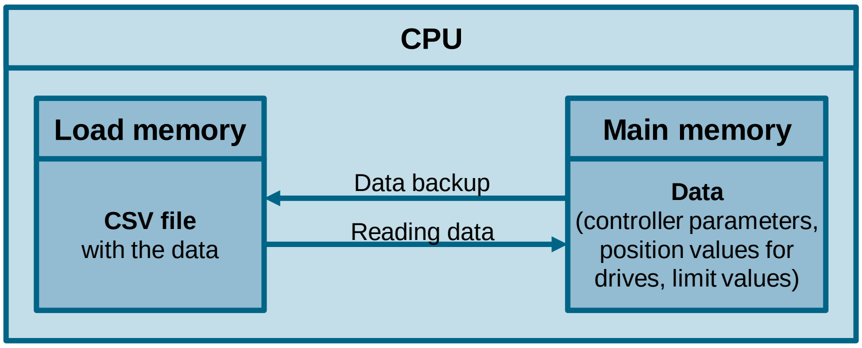
Introduction

In the course of commissioning a machine or plant, various parameters are adapted to the plant. These can be controller parameters, position values for drives, limit values etc.. In order for these parameters to be retained after POWER OFF and restart, they must have been backed up beforehand.

Overview of the automation task

The figure below provides an overview of the automation task.

Figure 1-1



1.2 Requirements of the automation task

- The programmer puts his machine into operation and sets various parameters. These parameters are saved persistently ("non-volatile" or "resistant towards zero") as CSV file on the memory card of the controller and hence backed up prior to power cut and program modifications.
- The user can decide when to save the data. Either automatically in a given time interval, or upon request by the user, for example via HMI.
- The data is backed up automatically as soon as the PSU8600 power supply switches into buffer mode in the event of a power failure.
- The machine is started with the new program. The original parameters shall be restored and further used in the program.
- After power failure and returned current, the machine runs again and reads the persistent data back from the CSV into the user program prior to the start.

Note

To avoid data loss due to a power failure, you may also flag certain data as remanent. This data will be stored in a remanent memory area.
You may also store your data in a DB in the load memory.
When the program is loaded, this data will be deleted. The data in the CSV file is also protected against program downloads.

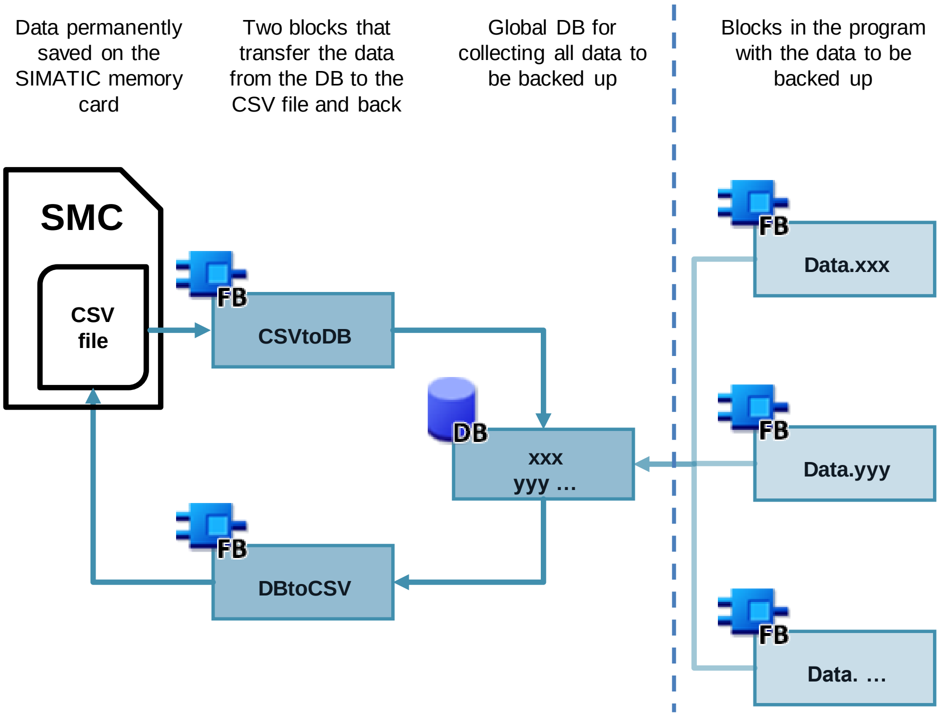
2 Solution

2.1 Overview

Schematic layout

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

Figure 2-1


Data permanently saved on the SIMATIC memory card
Two blocks that transfer the data from the DB to the CSV file and back
Global DB for collecting all data to be backed up
Blocks in the program with the data to be backed up
SMC
CSV file
FB
CSVtoDB
DB
xxx
yyy ...
FB
DBtoCSV
FB
Data.xxx
FB
Data.yyy
FB
Data. …

Setup

The parameters set during commissioning are collected in a global data block.

These settings values are either saved upon request, or cyclically in an adjustable time interval.

CAUTION	**Please note that the permitted number of write processes to the memory card must not be exceeded.** **The number of permitted write processes is available in the technical data of the respective memory card.**

The “DBtoCSV” block writes the collected data of the global data block to a CSV file on the load memory. The “CSVtoDB” block reads the data from the CSV file on demand and writes it back to the global data block.

Note Saving the parameters into the CSV file protects it from power failure and program changes.

Delimitation

- This application example does not contain a detailed description of the recipe function “RecipeImport” and “RecipeExport”.
- This application example does not contain a detailed description of the data block functions.

Note A detailed description of the recipe and data block functions is available in the TIA Portal online help or in the “STEP 7 Professional V13 SP1 system manual”.
https://support.industry.siemens.com/cs/ww/en/view/109011420

Note Further information on the CPU memory assignment is available in the functional manual.
“SIMATIC S7-1500 Structure and Use of the CPU Memory”.
https://support.industry.siemens.com/cs/ww/en/view/59193101

Basic knowledge of these topics is required.

Assumed knowledge

Basic knowledge for the following issues is assumed:

- STEP 7 (TIA Portal)
- WinCC (TIA Portal)
- STEP 7 block architecture and programming

2.2 Hardware and software components

2.2.1 Validity

The application example was tested with

- STEP 7 V13 SP1 Update 7
- S7-1500 V1.8
- S7-1200 V4.1

2.2.2 Components used

This application example was created with the following components:

It is recommended to use the TIA Selection Tool for configuring the hardware:

http://www.siemens.com/tia-selection-tool

Note A list of the hardware and software products used can also be found in the "109479727_Persistente_Daten_PRODUCTS_v10.zip" file in the archive. Use the TIA Selection Tool for this.

Hardware components

Table 2-1

Component	Qty	Article number	Note
CPU 1516-3 PN/DP	1	6ES7516-3AN00-0AB0	Alternatively, any other CPU of the S7-1500 product family can also be used.
Memory Card, 12 Mbytes	1	6ES7954-8LE02-0AA0	
CPU 1214C DC/DC/DC	1	6ES7214-1AG40-0XB0	Alternatively, any other CPU of the S7-1200 product family can also be used.
TP1200 Comfort	1	6AV2124-0MC01-0AX0	Optional: Simulation in TIA Portal is also possible.
SITOP PSU8600	1	6EP3437-8MP00-2CY0	Alternatively, any other PSU8600 may also be used.
SITOP BUF8600	1	6EP4297-8HB00-0XY0	If longer buffer times are required, other buffer modules may also be used alternatively. Per PSU8600, two buffer modules can be used.

Software components

Table 2-2

Component	Qty	Article number	Note
STEP 7 Professional V13 SP1 Update 7	1	6ES7822-1..03-..	
WinCC Advanced V13 SP1 Update 7	1	6AV210.-....3-0	

Example files and projects

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

Table 2-3

Component	Note
109479727_Persistente_Daten_DOKU_v10_en.pdf	This document.
109479727_Persistente_Daten_CODE_v10.zip	This zip file contains the STEP 7 project.
109479727_Persistente_Daten_PRODUCTS_v10.zip	The zip file contains: TIA Selection Tool file with hardware and software products.

3 Fundamentals of power supply system

3.1 Main unit SITOP PSU8600

The SITOP PSU8600 main unit is a primary switched power supply for connection to a 3-phase AC voltage grid. The outputs of the device are supplied with an electronically-controlled DC voltage which for each output can be adjusted through a potentiometer. The outputs of the device are potential-free, idling- and short-circuit proof. They have an electronic overload switch which can be adjusted through another potentiometer for each output. The outputs of the device can be switched on or off individually via a button for each output.

3.2 Expansion module SITOP CNX8600

Through a SITOP CNX8600 expansion module, the number of outputs of the system can be increased, but not the maximum output power of the main unit. Just like the main unit, the expansion modules have potentiometers to adjust each output and to switch them on or off via a button.

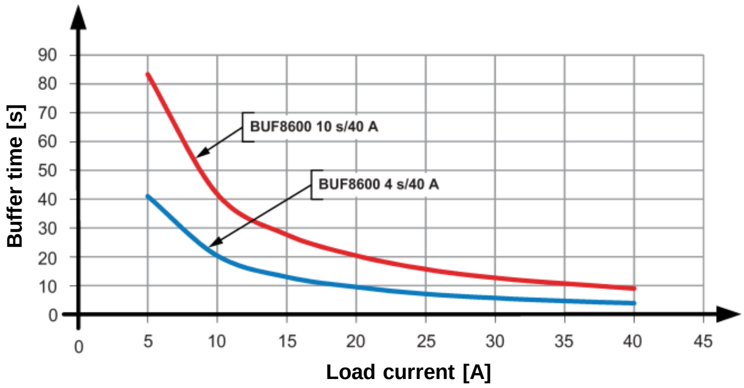
3.3 Buffer module SITOP BUF8600

A SITOP BUF8600 buffer module allows the buffer time to be extended in the event of a power failure. The operating status of the buffer module is indicated by an LED pilot lamp. Per main unit, a maximum of two buffer modules can be used. The connection to the main unit or to the adjacent module is established by an integrated connection plug.

In the event of a brief power failure, the outputs of the power supply system will still be supplied without interruption via the energy saved in the buffer module.

When the buffer module with an electrolytic capacitor is fully loaded, it can buffer 100 ms or 300 ms power failures at a load current of 40 A. With a lower load current, the buffer time increases correspondingly.
When the buffer module with a double layer capacitor is fully loaded, it can buffer 4 ms or 10 ms power failures at a load current of 40 A. With a lower load current, the buffer time increases correspondingly.

Figure 3-1 Typical buffer time depending on the load current



3.4 Engineering

The SITOP PSU8600 power supply system is fully integrated into the TIA Portal. The SITOP PSU8600 main unit is integrated into the network view and is connected to a control via an integrated PROFINET interface. The SITOP CNX8600 and SITOP BUF8600 additional modules are added in the device view. The parameters of the main unit and its additional modules are set in the inspector panel under properties.

Note The integration into the TIA Portal also enables a comprehensive evaluation of operating and diagnostic data as well as energy management functions.

Note Further information can be found in the technical manual of the SITOP PSU8600
https://support.industry.siemens.com/cs/ww/en/view/105867947
or on the SITOP-Homepage
http://www.siemens.com/sitop

3.5 Cyclic input and output data

The SITOP PSU8600 and its additional modules communicate with the control system via cyclic input and output data. The default addresses of the input and output data can be changed in the properties of the devices and modules.

Note A description of the input and output data of the SITOP PSU8600 and its additional modules can be found in the technical manual of the SITOP PSU8600
https://support.industry.siemens.com/cs/ww/en/view/105867947

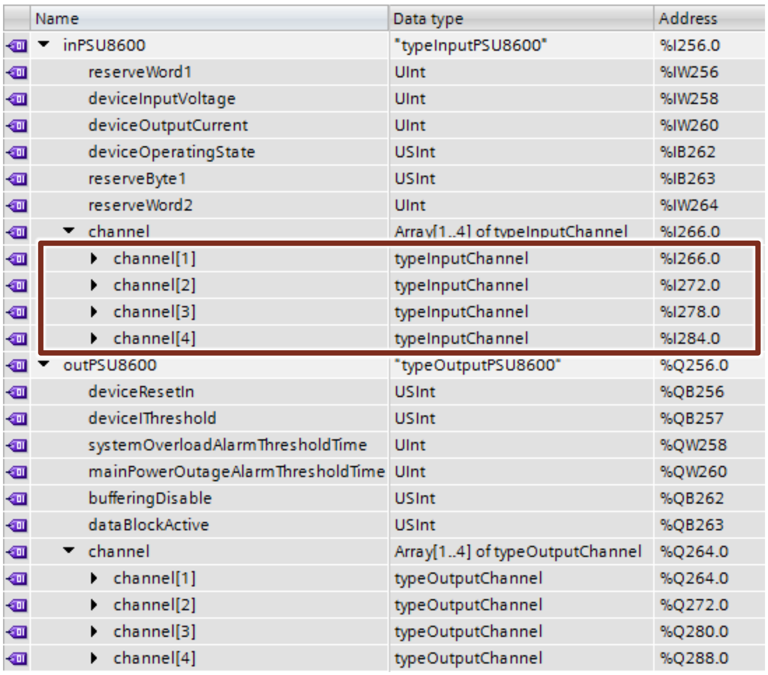
In this application example, the SIMATIC S7-1500 evaluates the data of the SITOP PSU8600.

In order to structurally and symbolically access the input and output data in the application program, you must observe the following points:

- Create a data type for both, input and output data.
- Create a PLC variable for both, input and output data. As a data type, use the newly created data types. As address, enter the initial address of the I/O data area of the device.

The following figure shows the PLC variables for the input and output data.

Figure 3-2



Name	Data type	Address
inPSU8600	"typeInputPSU8600"	%I256.0
reserveWord1	UInt	%IW256
deviceInputVoltage	UInt	%IW258
deviceOutputCurrent	UInt	%IW260
deviceOperatingState	USInt	%IB262
reserveByte1	USInt	%IB263
reserveWord2	UInt	%IW264
channel	Array[1..4] of typeInputChannel	%I266.0
channel[1]	typeInputChannel	%I266.0
channel[2]	typeInputChannel	%I272.0
channel[3]	typeInputChannel	%I278.0
channel[4]	typeInputChannel	%I284.0
outPSU8600	"typeOutputPSU8600"	%Q256.0
deviceResetIn	USInt	%QB256
deviceIThreshold	USInt	%QB257
systemOverloadAlarmThresholdTime	UInt	%QW258
mainPowerOutageAlarmThresholdTime	UInt	%QW260
bufferingDisable	USInt	%QB262
dataBlockActive	USInt	%QB263
channel	Array[1..4] of typeOutputChannel	%Q264.0
channel[1]	typeOutputChannel	%Q264.0
channel[2]	typeOutputChannel	%Q272.0
channel[3]	typeOutputChannel	%Q280.0
channel[4]	typeOutputChannel	%Q288.0

Table 3-1

Variable name	Data type	Meaning
inPSU8600	“typeInputPSU8600”	Data from SITOP PSU8600
outPSU8600	“typeOutputPSU8600”	Data to SITOP PSU8600

CAUTION

PLC data types always end at WORD limits. This means that for the PLC data type “typeInputChannel” with 5 bytes of user data, 6 Bytes are actually taken up.

In order to use the PLC data type “typeInputChannel”, the start addresses of the input channels must be adjusted to match the marked addresses in figure 3-1. This can be done in the properties of the SITOP PSU8600.

inPSU8600 [typeInputPSU8600]

Table 3-2

Variable name	Data type	Meaning
reserveWord1	UInt	Filling bytes
deviceInputVoltage	UInt	Input voltage main unit [10mV]
deviceOutputCurrent	UInt	System load power of power supply [10mA]
deviceOperatingState	USInt	Operating status of the power supply system. (**9 = buffer mode**)
reserveByte1	USInt	Filling byte
reserveWord2	UInt	Filling bytes
channel	Array[1..4] of „typeInputChannel“	Current values of output channel 1 to 4

channel [typeInputChannel]

Table 3-3

Tag name	Data type	Meaning
outputVoltage	UInt	Current voltage at output x
outputCurrent	UInt	Current load current at output x
operationState	USInt	Operating status at output x

outPSU8600 [typeOutputPSU8600]

Table 3-4

Tag name	Data type	Meaning
deviceResetIn	USInt	Status of input "Reset_In"
deviceThreshold	USInt	Prewarning threshold of total current
systemOverload AlarmThresholdTime	UInt	Time delay for alarm message after system overload has been detected.
mainPowerOutage AlarmThresholdTime	UInt	Time delay for alarm message after supply voltage failure.
bufferingDisable	USInt	Buffer module has been deactivated by control command.
dataBlockActive	USInt	Indicator: Data block contains valid data.
channel	Array[1..4] of „typeOutputChannel“	Current set values of output channel 1 to 4

channel [typeOutputChannel]

Table 3-5

Tag name	Data type	Meaning
UTarget	UInt	Current set value of the output current for output x
ILimit	UInt	Response threshold of output current for output x
IThreshold	USInt	Prewarning threshold of the configured response threshold for the output current in %
outputEnable	USInt	Switches output x on or off
resetIn	USInt	Reset of overload shutoff for this output
dataBlockActive	USInt	Indicator: Data block contains valid data.

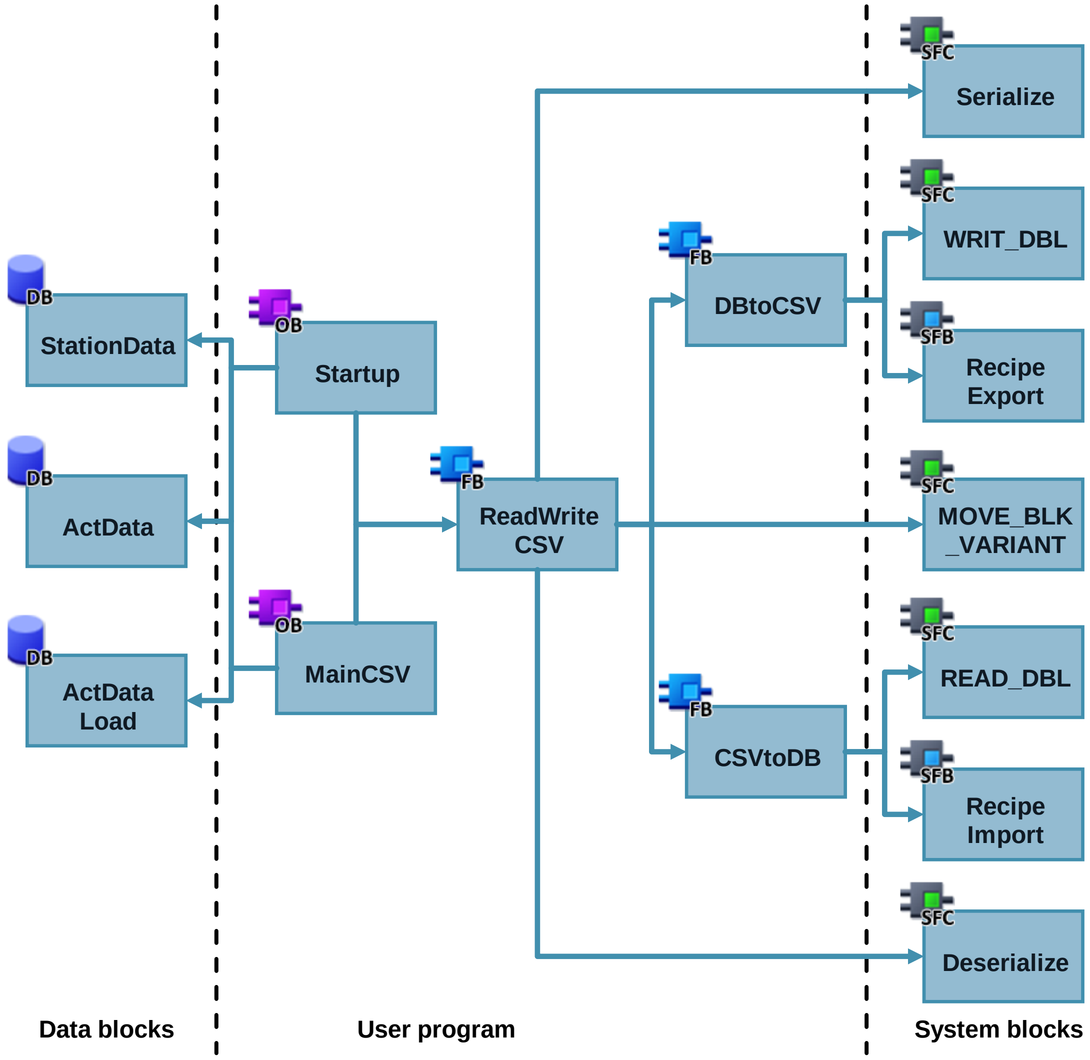
4 Mode of Operation

4.1 General overview

Program overview SIMATIC S7-1200

The figure below shows the program structure of the STEP 7 project.

Figure 4-1



Program overview SIMATIC S7-1500

The figure below shows the program structure of the STEP 7 project.

Figure 4-2

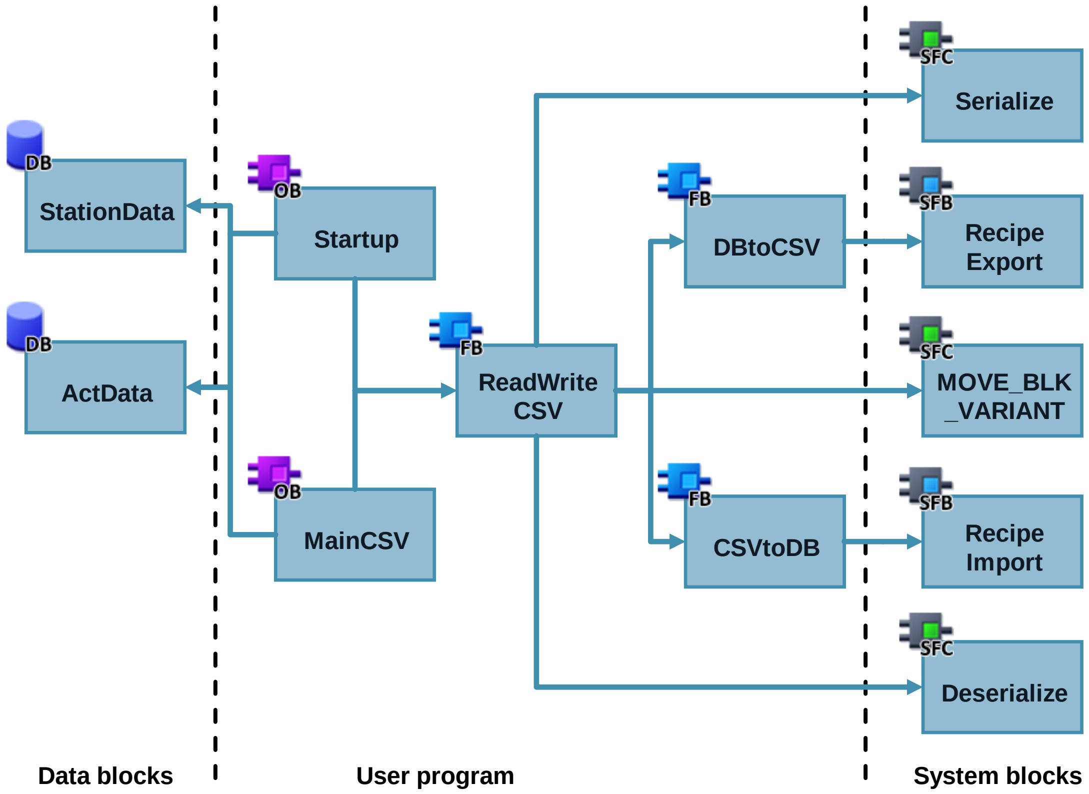


Table 4-1

Symbolic name	Description
StationData [DB1]	The global data block (DB) “ActData” in the main memory contains the data to be backed up.
ActData [DB2001]	The global data block (DB) “ActData” in the main memory contains the data to be backed up (Array of Byte) and the tags for communicating with the HMI.
ActDataLoad [DB2002] (only for S7-1200)	The global data block (DB) “ActDataLoad” in the load memory contains the data to be backed up (Array of Byte).
Startup [OB100]	The startup organization block (OB) calls the function block “ReadWriteCSV” to read the data from the CSV file.
MainCSV [OB123]	The cyclic organization block (OB) calls the function block “ReadWriteCSV”.
ReadWriteCSV [FB2000]	The function block (FB) calls the function blocks “DBtoCSV” and “CSVtoDB”.
DBtoCSV [FB2001]	The function block (FB) takes on the saving of data in the global data block into the CSV file using the instructions “WRIT_DBL” (only for S7-1200) and “ExportRecipe”.
CSVtoDB [FB2001]	The function block (FB) takes on the reading of data from the CSV file into the global data block using the instructions “RecipeImport” (only for S7-1200) and “READ_DBL”.

Note The recipe functions of the SIMATIC S7-1500 enable writing the data directly from the main memory (DB “ActData”) into a CSV file or reading it directly from the main memory into the CSV file. Intermediate storage of data in the load memory (DB “ActDataLoad”) is therefore not necessary.

For SIMATIC S7-1200, the following data from the main memory (DB “ActData”) must first be intermediately stored in the load memory (DB “ActDataLoad”) before writing it to a CSV file. If the data is read from the CSV file, it must first be intermediately stored in the load memory (DB “ActDataLoad”) before transferring it to the main memory (DB “ActData”)

Saving the parameters into the CSV file protects it from power failure and program changes.

CAUTION **Please note that the permitted number of write processes to the memory card must not be exceeded.**

The number of permitted write processes is available in the technical data of the respective memory card.

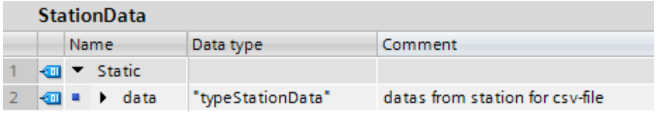
4.2 Global data block “StationData”

Overview

The data block (DB) “StationData” in the main memory contains all data to be backed up on your device.

The figure below shows the structure of the “StationData” global data block.

Figure 4-3



StationData

	Name	Data type	Comment
1	▼ Static		
2	▶ data	"typeStationData"	datas from station for csv-file

Table 4-2

Tag name	Data type	Meaning
data	typeStationData	System-specific data to be backed up.

data [typeStationData]

Data type “typeStationData” describes the structure of the data to be saved. An arbitrary structure was defined in this application example.

This data type must be adapted to your project.

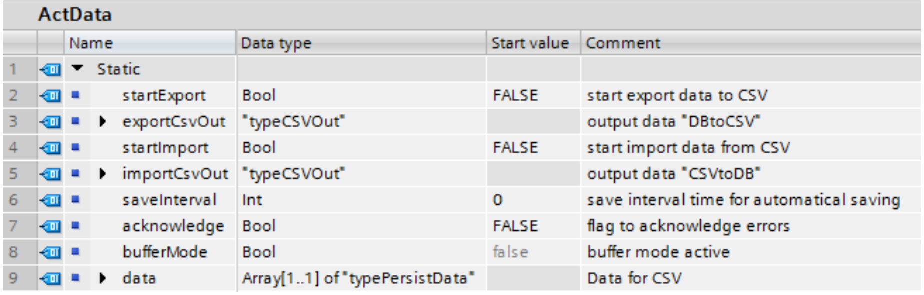
4.3 Global data block “ActData”

Overview

The global data block (DB) “ActData” in the main memory contains the data to be backed up (Array of Byte), the tags for controlling the application example and the variable „bufferMode“, which will be set, if the PSU8600 is switched into the buffer mode.

The figure below shows the structure of global data block “ActData”.

Figure 4-4



ActData

	Name	Data type	Start value	Comment
1	▼ Static			
2	startExport	Bool	FALSE	start export data to CSV
3	▸ exportCsvOut	"typeCSVOut"		output data "DBtoCSV"
4	startImport	Bool	FALSE	start import data from CSV
5	▸ importCsvOut	"typeCSVOut"		output data "CSVtoDB"
6	saveInterval	Int	0	save interval time for automatical saving
7	acknowledge	Bool	FALSE	flag to acknowledge errors
8	bufferMode	Bool	false	buffer mode active
9	▸ data	Array[1..1] of "typePersistData"		Data for CSV

Table 4-3

Tag name	Data type	Meaning
startExport	Bool	Activates writing the data into the CSV file.
exportCsvOut	“typeCSVOut”	Output parameter of function block “DBtoCSV”.
startImport	Bool	Activates reading the data from the CSV file.
importCsvOut	“typeCSVOut”	Output parameter of function block “CSVtoDB”.
saveIntervall	Int	Time interval for the automatic backup in hours Default: 0 (no automatic backup)
acknowledge	Bool	Error acknowledgement
bufferMode	Bool	Activates writing the data into the CSV file when the PSU8600 switches to buffer mode.
data	Array[1..1] of “typePersistData”	Data to be backed up (see data [typePersistData])

exportCsvOut / importCsvOut [typeCSVOut]

Table 4-4

Tag name	Data type	Meaning
busy	Bool	busy = 1, if block currently executed
done	Bool	done = 1, if block executed without errors
error	Bool	error = 1, if an error has occurred
statusID	UInt	ID of the error source
status	Word	Error code that uniquely identifies the error
elapsedTime	Time	Execution duration of the last call
lastExecute	DTL	Outputs the time, when the block was last started.

data [typePersistData]

Table 4-5

Tag name	Data type	Meaning
b	Array[0..254] of Byte	Data to be backed up

Note PLC data types and structures are not supported by the recipe functions. Therefore, the data to be backed up is converted into “Array of Byte” in this application example.

Note The tag names of the recipe data block are also written to the CSV file. Please note, that the number of characters of all tag names must not exceed 5000 characters. Therefore, please use the shortest tag names possible.

Note The data structure of the recipe data block at “RecipeExport” must not exceed 5000 bytes.

Note Please ensure that the parameters to be backed up are consistent.

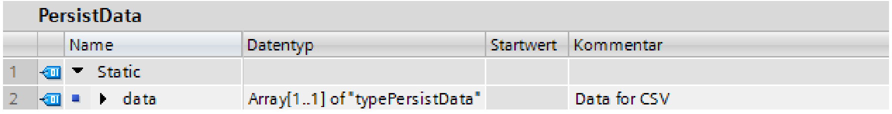
4.4 Global data block “ActDataLoad” (only S7-1200)

Overview

The data block (DB) “ActDataLoad” in the load memory contains all of the data to be backed up (Array of Byte).

The figure below shows the structure of global data block “ActDataLoad”.

Figure 4-5



PersistData

	Name	Datentyp	Startwert	Kommentar
1	▼ Static			
2	■ ▸ data	Array[1..1] of "typePersistData"		Data for CSV

Table 4-6

Tag name	Data type	Meaning
data	Array[1..1] of “typePersistData”	Data to be backed up (see data [typePersistData])

Note

For the S7-1200, the data block of the data to be saved must be located in the load memory of the controller. This requires that the field “Only store in load memory” be activated in the properties of DB “ActDataLoad” in attributes.

If the data block is located in the main memory, error code “80B6” is output at “RecipeExport”.

4.5 Function block “ReadWriteCSV”

The block serves as central block for writing the data into the CSV file and reading the data from the CSV file.

Interfaces

Figure 4-6: Call in “MainCSV” for S7-1200

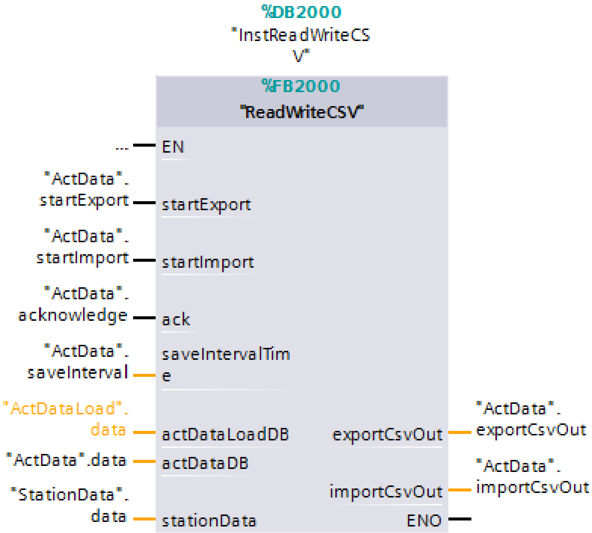


Figure 4-7: Call in “MainCSV” for S7-1500

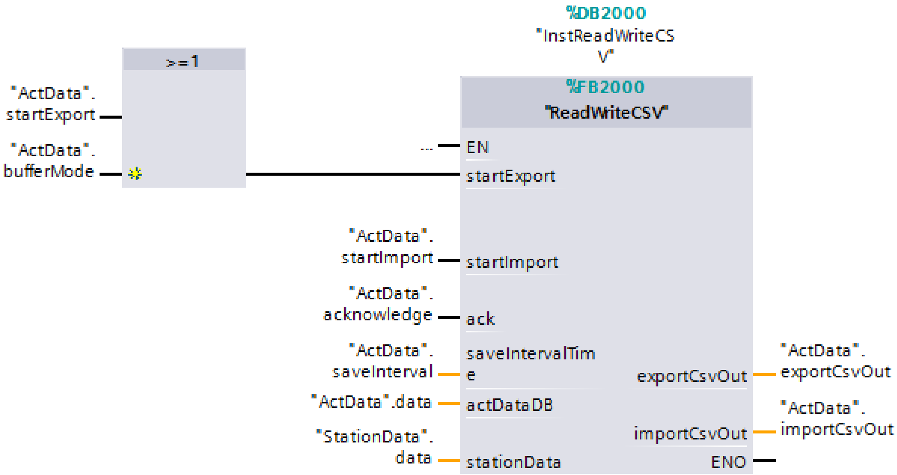


Table 4-7

Type	Parameter	Data type	Description
Input	startExport	Bool	Activates writing the data into the CSV file.
	startImport	Bool	Activates reading the data from the CSV file.
	ack	Bool	Error acknowledgement
	saveIntervalTime	Int	Interval for the automatic backup
Output	exportCsvOut	"typeCSVOut"	Output parameter of function block "DBtoCSV".
	importCsvOut	"typeCSVOut"	Output parameter of function block "CSVtoDB".
InOut	actDataLoadDB	Variant	Data to be saved in the load memory (Array of Byte) (only for S7-1200)
	actDataDB	Variant	Data to be saved in the main memory (Array of Byte)
	stationData	typeStationData	System-specific data to be stored.

Function description

Function block "ReadWriteCSV" is the higher-level block for writing and reading the data. It is called in startup OB "Startup" and in cyclic OB "MainCSV". During startup, the block reads the saved data from the CSV file and copies it into the global data block „StationData". In cyclic operation, it takes on the automatic backup of data or saving and reading data on request.

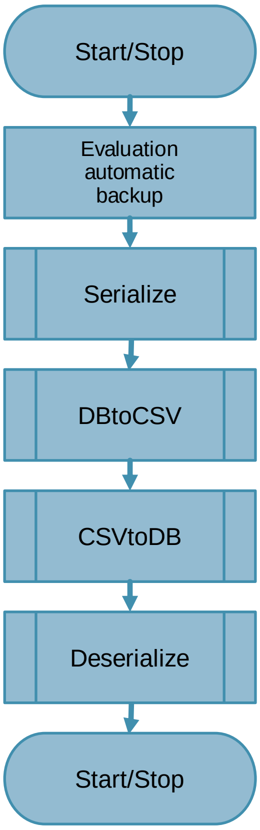
The block calls the function blocks "DBtoCSV" for writing the data, and "CSVtoDB" for reading the data.

In order to store complex data structures in the data block "StationData", the function "Serialize" converts the data into an "Array of Byte". While reading the data, the function "Deserialize" converts the data from "Array of Byte" into the original structure.

Note If you change the data type "typeStationData", you first have to back up the data before reading it out again.

The following figure shows the principle program sequence of the function block.

Figure 4-8

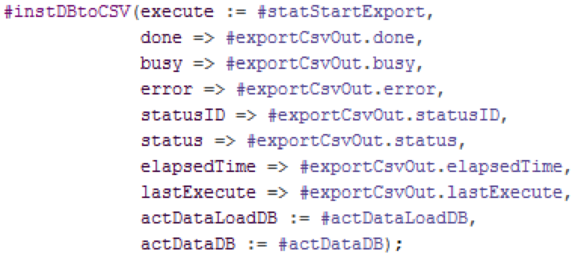


4.6 Function block “DBtoCSV”

The function block takes on writing the data into the CSV file.

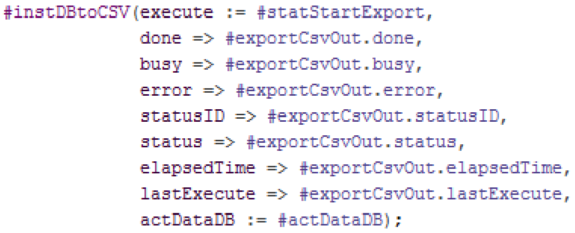
Interfaces

Figure 4-9: Call in “ReadWriteCSV” for S7-1200



```
#instDBtoCSV(execute := #statStartExport,
             done => #exportCsvOut.done,
             busy => #exportCsvOut.busy,
             error => #exportCsvOut.error,
             statusID => #exportCsvOut.statusID,
             status => #exportCsvOut.status,
             elapsedTime => #exportCsvOut.elapsedTime,
             lastExecute => #exportCsvOut.lastExecute,
             actDataLoadDB := #actDataLoadDB,
             actDataDB := #actDataDB);
```

Figure 4-10: Call in “ReadWriteCSV” for S7-1500



```
#instDBtoCSV(execute := #statStartExport,
             done => #exportCsvOut.done,
             busy => #exportCsvOut.busy,
             error => #exportCsvOut.error,
             statusID => #exportCsvOut.statusID,
             status => #exportCsvOut.status,
             elapsedTime => #exportCsvOut.elapsedTime,
             lastExecute => #exportCsvOut.lastExecute,
             actDataDB := #actDataDB);
```

Table 4-8

Type	Parameter	Data type	Description
Input	execute	Bool	Start writing
Output	busy	Bool	busy = 1, if block currently executed
	done	Bool	done = 1, if block executed without errors
	error	Bool	error = 1, if an error has occurred
	statusID	UInt	ID of the error source 1: local FB 2: “RecipeExport” error 3: “WRIT_DBL” error (only for S7-1200)
	status	Word	Error code that uniquely identifies the error. The error codes of the instructions “RecipeExport” and “WRIT_DBL” are described in the online help.
	elapsedTime	Time	Execution duration of the last call
	lastExecute	DTL	Outputs the time, when the block was last started.
InOut	actDataLoadDB	Variant	Data to be saved in the load memory (only for S7-1200)
	actDataDB	Variant	Data to be saved in the main memory

Function description

The block “DBtoCSV” is used for writing the data into the CSV file in data block “ActData”.

S7-1200

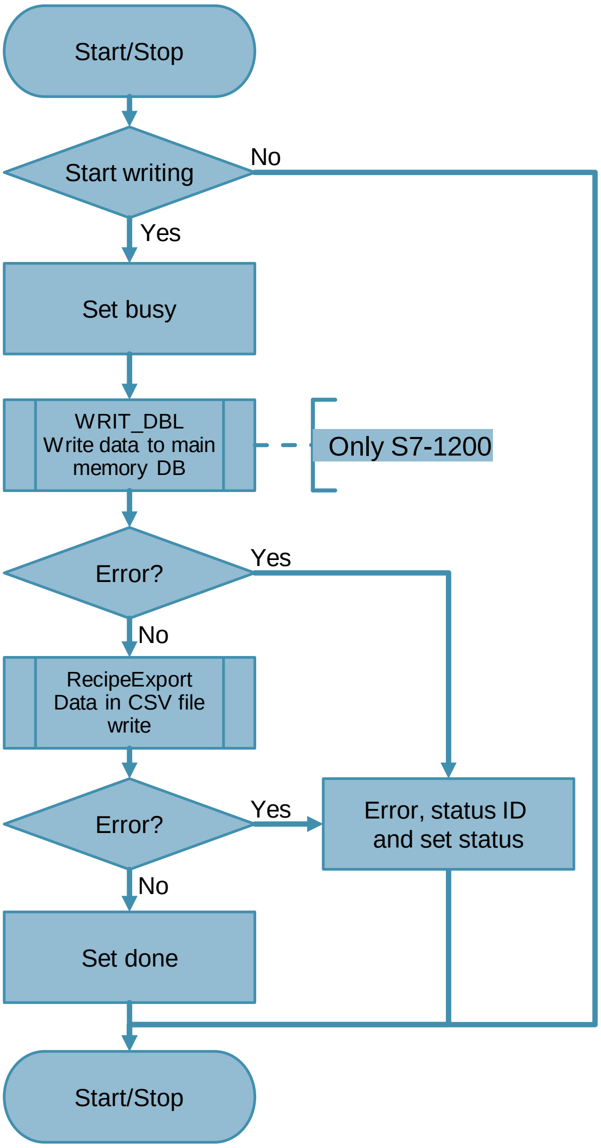
The “WRIT_DBL” instruction writes the data in data block “ActData” into DB “ActDataLoad”, which is stored in the load memory. The “RecipeExport” instruction then writes the data from DB “ActDataLoad” into the CSV file.

S7-1500

The “RecipeExport” instruction writes the data directly from DB “ActData” stored in the main memory into the CSV file.

The following figure shows the principle program sequence of the function block.

Figure 4-11

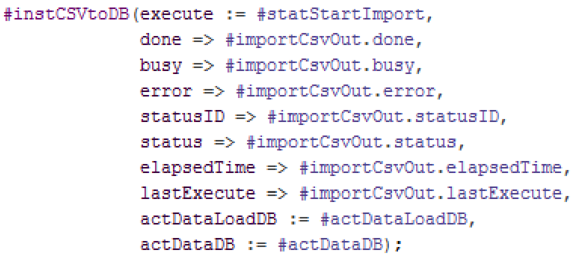


4.7 Function block “CSVtoDB”

The function block takes on reading the data from the CSV file.

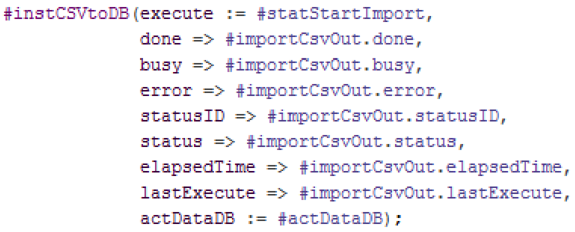
Interfaces

Figure 4-12: Call in “ReadWriteCSV” for S7-1200



```
#instCSVtoDB(execute := #statStartImport,
             done => #importCsvOut.done,
             busy => #importCsvOut.busy,
             error => #importCsvOut.error,
             statusID => #importCsvOut.statusID,
             status => #importCsvOut.status,
             elapsedTime => #importCsvOut.elapsedTime,
             lastExecute => #importCsvOut.lastExecute,
             actDataLoadDB := #actDataLoadDB,
             actDataDB := #actDataDB);
```

Figure 4-13: Call in “ReadWriteCSV” for S7-1500



```
#instCSVtoDB(execute := #statStartImport,
             done => #importCsvOut.done,
             busy => #importCsvOut.busy,
             error => #importCsvOut.error,
             statusID => #importCsvOut.statusID,
             status => #importCsvOut.status,
             elapsedTime => #importCsvOut.elapsedTime,
             lastExecute => #importCsvOut.lastExecute,
             actDataDB := #actDataDB);
```

Table 4-9

Type	Parameter	Data type	Description
Input	execute	Bool	Start reading
Output	busy	Bool	busy = 1, if block currently executed
	done	Bool	done = 1, if block executed without errors
	error	Bool	error = 1, if an error has occurred
	statusID	UInt	ID of the error source 1: local FB 2: “RecipeImport” error 3: “READ_DBL” error (only for S7-1200)
	status	Word	Error code that uniquely identifies the error The error codes of the instructions “RecipeImport” and “READ_DBL” are described in the online help.
	elapsedTime	Time	Execution duration of the last call
	lastExecute	DTL	Outputs the time, when the block was last started.
InOut	actDataLoadDB	Variant	Data to be saved in the load memory (only for S7-1200)
	actDataDB	Variant	Data to be saved in the main memory

Function description

The block “DBtoCSV” is used for reading the data from the CSV file and transferring it to data block “ActData”.

S7-1200

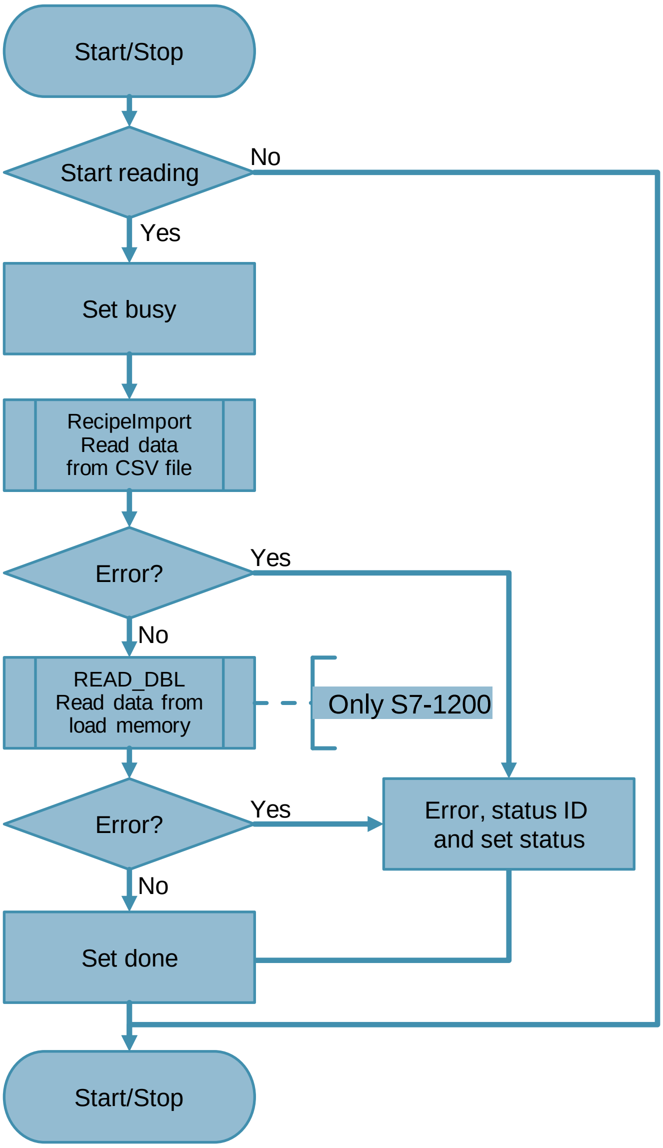
The “RecipeImport” instruction imports the data from the CSV file into DB “ActDataLoad” in the load memory. The “READ_DBL” instruction transfers the data from DB “ActDataLoad” into DB “ActData” in the main memory.

S7-1500

The “RecipeImport” instruction imports the data directly from the CSV file into DB “ActData” which is stored in the main memory.

The following figure shows the principle program sequence of the function block.

Figure 4-14

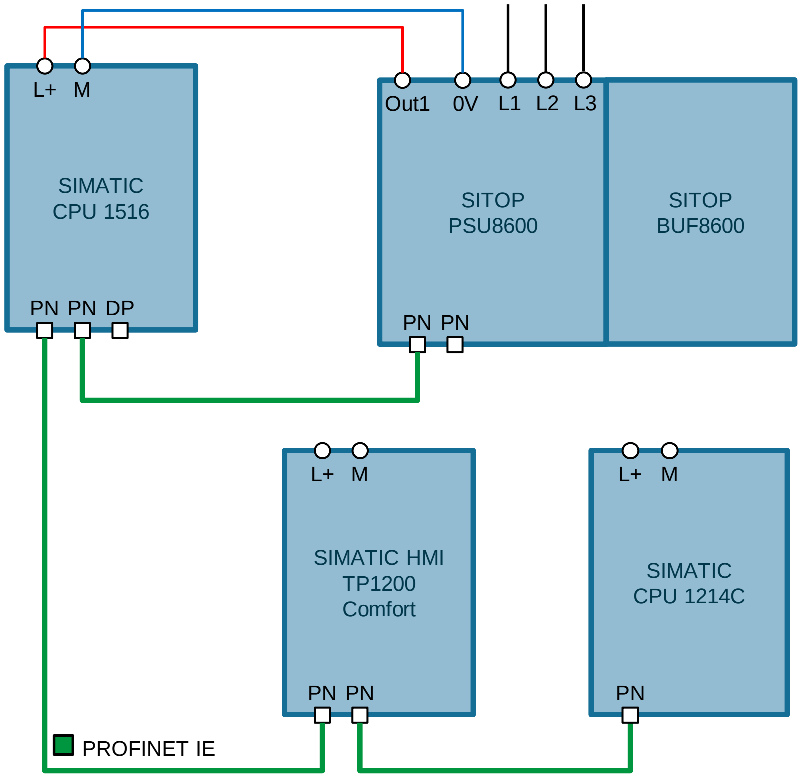


5 Installation and Commissioning

5.1 Installing the hardware

The figure below shows the hardware configuration of the application.

Figure 5-1



The SIMATIC CPU 1516 is supplied through channel 1 of the SITOP PSU8600 power supply, thereby ensuring an uninterruptable supply of power (buffering) for a certain amount of time in the event of a power failure.

Note The setup guidelines of the devices must generally be followed.

5.2 IP addresses and device names

In the example, the following device numbers, IP addresses and device names are used:

Table 5-1

Component	IP Address	Device name
SIMATIC CPU 1516	192.168.0.1	PLC_1
SIMATIC CPU 1214C	192.168.0.2	PLC_2
SIMATIC HMI TP1200	192.168.0.10	TP1200
SITOP PSU8600	192.168.0.11	PSU_1

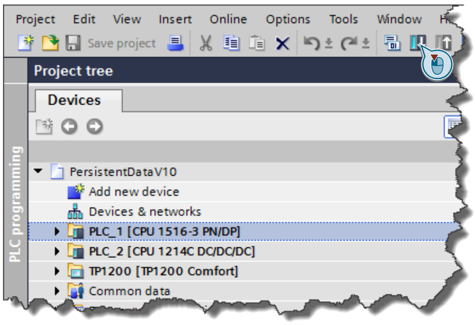
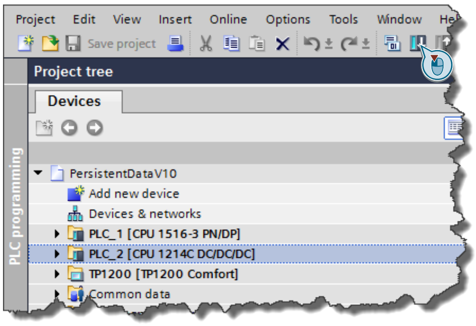
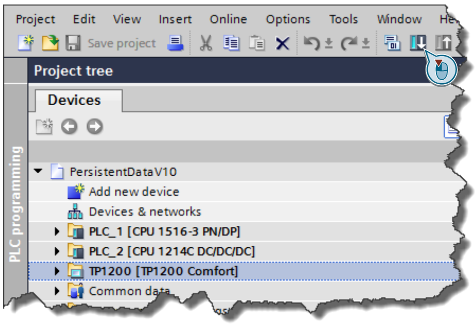
5.3 Installing the software (download)

Note At this point, it is assumed that the necessary software has been installed on your computer and that you are familiar with the software.

5.4 Loading the project

The software example is available on the HTML page from which you downloaded this document.

Table 5-2

No.	Action	Notes
1.	Unzip the zipped code folder “109479727_Persistente_Daten_CODE_V10.zip” into a directory of your choice.	
2.	Open the “PersistentDataV10.ap13” project with the TIA Portal.	
3.	In the project tree you select the folder “PLC_1” (S7-1500) or “PLC_2” (S7-1200), depending on the controller you use, and in the toolbar you click on the “Download to device” button.	 Project Edit View Insert Online Options Tools Window Save project Project tree Devices PLC programming PersistentDataV10 Add new device Devices & networks PLC_1 [CPU 1516-3 PN/DP] PLC_2 [CPU 1214C DC/DC/DC] TP1200 [TP1200 Comfort] Common data  Project Edit View Insert Online Options Tools Window Save project Project tree Devices PLC programming PersistentDataV10 Add new device Devices & networks PLC_1 [CPU 1516-3 PN/DP] PLC_2 [CPU 1214C DC/DC/DC] TP1200 [TP1200 Comfort] Common data
4.	Select the “TP1200” folder of the operator panel in the project tree and click the “Download to device” button in the toolbar.	 Project Edit View Insert Online Options Tools Window Save project Project tree Devices PLC programming PersistentDataV10 Add new device Devices & networks PLC_1 [CPU 1516-3 PN/DP] PLC_2 [CPU 1214C DC/DC/DC] TP1200 [TP1200 Comfort] Common data

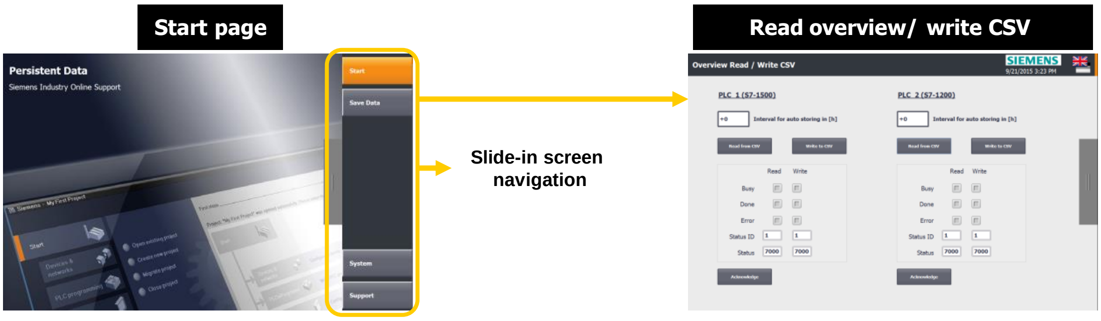
Note When starting the controller for the first time, a CSV file does not yet exist and can therefore not be read during startup. Status “8092” is therefore displayed on the HMI when reading. The error status must be acknowledged when clicking on the “Acknowledge” button.

6 Operating the Application Example

6.1 Overview

The figure below shows the user interface of the operator panel.

Figure 6-1



The “Persistent Data” screen shows the start screen of the application example.

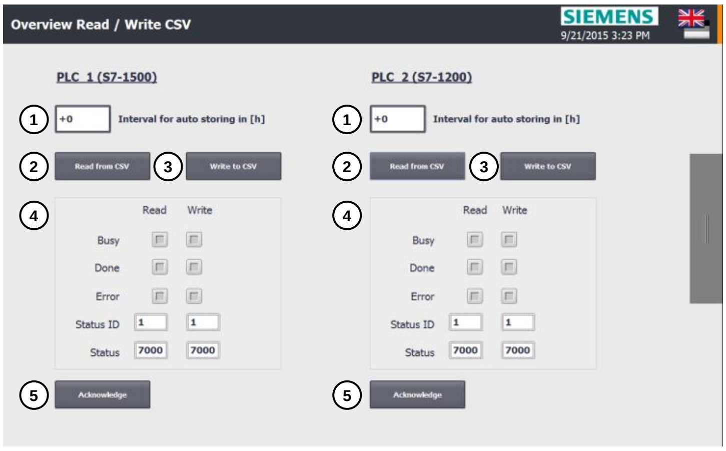
The slide-in screen is used for screen navigation.

In the “Overview Read / Write CSV” screen you start reading the data from the CSV file or writing it to the CSV file.

6.2 Operation

Proceed as follows when operating the application example.

Table 6-1

Action	Notes
Open the “Overview Read / Write CSV” screen at the operator panel. 1. Here you enter the time interval in which the automatic backup shall be performed. “0” means: no automatic backup 2. Clicking on the button activates reading from the CSV file. 3. Clicking on the button activates writing to the CSV file. 4. This is where you can monitor the current read / write status. 5. In the event of errors, a renewed action “Read from CSV” / “Write to CSV” is only possible after “Acknowledge”. Clicking on the button acknowledges the error status.	 Overview Read / Write CSV SIEMENS 9/21/2015 3:23 PM PLC_1 (S7-1500) PLC_2 (S7-1200) 1 Interval for auto storing in [h] 2 Read from CSV 3 Write to CSV 4 Read Write Busy Done Error 5 Acknowledge PLC_1 (S7-1500): Interval for auto storing in [h] +0; Status ID 1 1; Status 7000 7000 PLC_2 (S7-1200): Interval for auto storing in [h] +0; Status ID 1 1; Status 7000 7000

7 Links & Literature

Table 7-1

	Topic	Title
\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/109479727
\3\	STEP 7 Professional V13 SP1 System Manual	https://support.industry.siemens.com/cs/ww/en/view/109011420
\4\	S7-1500 System Manual	https://support.industry.siemens.com/cs/ww/en/view/59191792
\5\	S7-1200 System Manual	https://support.industry.siemens.com/cs/ww/en/view/109478121
\6\	Using recipes with SIMATIC S7-1200	https://support.industry.siemens.com/cs/ww/en/view/94681612
\7\	Device manual SITOP PSU8600	https://support.industry.siemens.com/cs/ww/en/view/105867947
\8\	Homepage SITOP Power Supply	http://www.siemens.com/sitop
\9\	HSP for SITOP PSU8600	https://support.industry.siemens.com/cs/ww/en/view/102254062 or https://support.industry.siemens.com/cs/ww/en/view/72341852
\10\	PSU8600 Firmware V1.1	https://support.industry.siemens.com/cs/ww/en/view/102295547

8 History

Table 8-1

Version	Date	Modifications
V1.0	04/2016	First version