SIEMENS

Application Description • 07/2015

Recipes with SIMATIC S7-1200

SIMATIC STEP 7 (TIA Portal) V13

https://support.industry.siemens.com/cs/ww/en/view/94681612

Warranty and Liability

Note

The Application Examples are not binding and do not claim to be complete with regard to configuration, equipment or any contingencies. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for the correct operation of the described products. These Application Examples do not relieve you of the responsibility of safely and professionally using, installing, operating and servicing equipment. 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 and without prior notice. If there are any deviations between the recommendations provided in this Application Example 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 will be excluded. Such an exclusion will 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 compensation for damages due to a breach of a fundamental contractual obligation is, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

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

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens' products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit http://www.siemens.com/industrialsecurity.

To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit http://support.industry.siemens.com.

Table of Contents

1 Task

Overview of the automation task

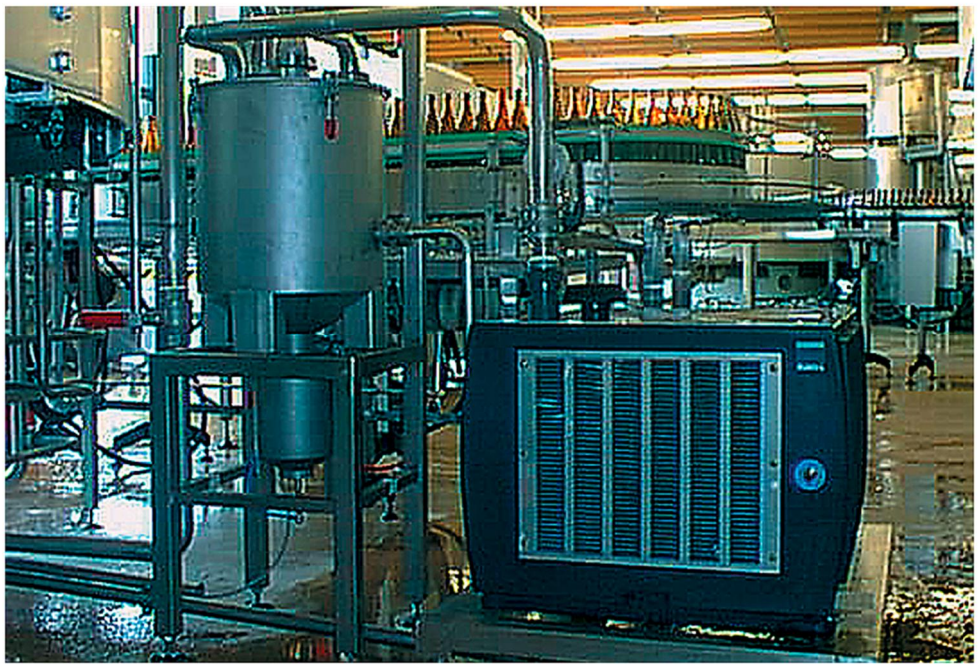
In production or process automation, the application of recipes is useful wherever different variants or compositions of a product shall be produced. The application is therefore not restricted to the different composition of various content materials. Recipes can, for example, also be applied to different production processes. For example, product A is shrink wrapped, strapped and labelled, product B is packaged, printed on and shrink wrapped.

The application shall illustrate a simplified filling process and work with a recipe management. Furthermore, the user shall be able to observe the process.

The integration into an existing infrastructure shall be possible without greater expenses.

The figure below shows a bottling plant:

Figure 1-1



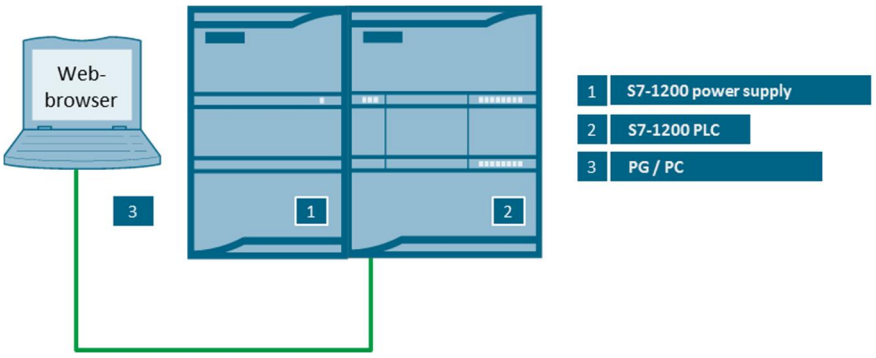
2 Solution

2.1 Overview

Schematic layout

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

Figure 2-1



The automation solution uses an S7-1200 controller. Connecting a PG/PC to the controller enables calling the values of the filling process (number of bottles and elements to be filled) via a user-defined web page (AWP) in the web server of the CPU. In the example, the valves for the individual content materials are represented as “0” for closed or “1” for open. Additionally, the current number of already filled bottles is displayed. Furthermore, it is possible via the AWP to view recipes (export function), change and reload recipes (import function) as well as control the application. Operation is here handled via the bowser of the PG or PC e.g. Internet Explorer.

Implementation via web server makes various access stages available to the user.

Advantages

The solution presented here offers you the following advantages:

- Time and cost saving by simple configuration via TIA Portal and use of already existing hardware
- Expandability
- Easy integration into existing systems
- Comfortable option to gain an overview of the application or even implement control functions
- Access to the CPU via standard mechanisms, especially relating to the plant - each CPU can obtain its own page, if required
- Access protection for the web server by means of user management
- Operating personnel without any automation knowledge is also provided simple access to the CPU

Delimitation

This application does not contain a description of recipes with a visualization system, e.g. WinCC or WinCCFlex

Assumed knowledge

Basic knowledge of S7-1200 and STEP 7 (TIA Portal) is assumed.

2.2 Hardware and software components

2.2.1 Validity

This application is valid for

- STEP 7 (TIA Portal) V13 SP1 \5\
- S7-1200 V4.0 \3\

2.2.2 Components used

The application was created with the following components:

Hardware components

Table 2-1

Component	No.	Article number	Note
Power supply PM1207	1	6EP1332-1SH71	Supplies the components with 24V DC
CPU 1212C DC/DC/DC Firmware V4.0	1	6ES7212-1AE40-0XB0	Alternatively, any other S7-1200 CPU with firmware 4.0 can be used
PG/PC with an Ethernet interface	1	-	-
Top hat rail	1	6ES5710-8MA11	483mm
SIMATIC NET, IND. ETHERNET TP CORD RJ45/RJ45, CAT 6, TP CABLE 4X2, PREASSEMBLED WITH 2 RJ45 CONNECTORS, … 0.5M 1M 2M 6M 10M	1	6XV1870-3Q… …E50 …H10 …H20 …H60 …N10	Ethernet cable

Software components

Table 2-2

Component	No.	Article number	Note
STEP 7 Professional/ Basic (TIA Portal) V13 \6\	1	6ES7822-1..03-..	Configuration and programming of the S7-1200
Microsoft Excel/ Windows Notepad	-	-	Open or change the CSV file

Sample files and projects

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

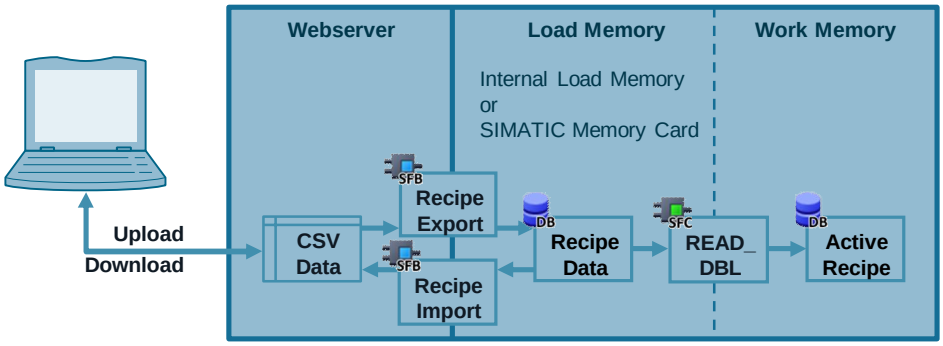
Table 2-3

Component	Note
94681612_S7-1200_Recipe_CODE_v11.zip	This zip file contains the STEP 7 (TIA Portal) project.
94681612_ S7-1200_Recipe_DOKU_v11_en.pdf	This document

3 Basics

3.1 Structure of the memory

Figure 3-1



An S7-1200 is internally equipped with a memory. This is divided into the load and the work memory. With the CPU 1212C used in the application example, it is divided into the 50 Kbyte work memory and the 1 Mbyte load memory. To prevent unnecessary load on the work memory, recipe data are stored in the load memory whenever possible. With a SIMATIC memory card (SMC) it can be expanded by up to 2GB on demand. In the example, 2 data blocks are used. The “ActiveRecipe” DB is located in the work memory (contains only one recipe data record), the “RecipeData” DB on the other hand is located in the load memory and contains several recipe data records. The data is written to a CSV file via the export function and loaded back into the controller via the import function and there copied internally with “READ_DBL”.

Note Apart from the “READ_DBL” function there is also the “WRIT_DBL” function which makes it possible to copy data from a DB in the work memory into a DB in the load memory.

3.2 Creating user-defined web pages (AWP)

The web server of the CPU provides a lot of information on the basic status of the CPU as well as details of the diagnostic buffer. However, to enable a specific view onto the application, the S7-1200 offers user-defined web pages. These must be written by the user in HTM or HTML. Depending on the requirement, data can be read from or written to the S7-1200. This provides the user with a useful means of accessing the application in everyday work without needing to operate special software.

3.2.1 Basic steps

The following basic steps must be run through to be able to call the AWP via the menu of the standard web pages:

- Create HTML pages with an HTML editor as well as Microsoft FrontPage.
- Enter AWP commands into HTML comments in HTML code (“AWP commands” is a default set of commands provided for accessing CPU information).
- Configure the CPU with TIA Portal for reading and processing of HTML pages.
- Create blocks from the HTML pages in TIA Portal.
- Program TIA Portal for controlling the use of the HTML pages.
- Compile blocks and load into the CPU
- Access the user-defined web pages via your PC

Note

STEP 7 does not perform a verification of the HTML source code!

The maximum file size for HTML files with AWP commands is 64 KB. The size of your files must not exceed this limit value.

Further notes on the subject of “user-defined web pages” are available in the “Automation System S7-1200” \3\ manual in chapter 11.6 as well as in the application example “Creating and using user-defined web pages for S7-1200” \4\ as well in application example “Creating and using user-defined web pages on S7-1500” \10\

3.2.2 Preconditions

The following requirements must be met for accessing variables on the web page:

- Each variable must be assigned a symbolic name. The variable can only be displayed on the web page or written to via symbolic names.
- The “WWW” (SFC99) instruction must be called (if variables are pre-processed in the S7 program, a cyclic call is possible)
- For variables the standard data types (“DTL” is not displayed), user-created PLC data types, and structures are permitted.

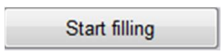
In the STEP 7 program no further programming is necessary except for the call of block “WWW”.

3.2.3 Instruction in HTML code / AWP commands

To realize reading as well as writing to variables in S7-1200, certain commands are necessary.

Table 3-1

Command	Remarks
Reading a variable	General syntax: **:="<Variable>":** Example in the AWP: **:="ActiveRecipe".product.water:** ("ActiveRecipe" is the DB, "product" the structure and "water" the variable) Reading a variable on the AWP does not require an AWP command.

Writing und sending a variable	General syntax:
	<!-- AWP_In_Variable Name='"Variable"' -->
	
	For writing a variable requires an AWP command, best at the start of the HTM(L) page. In addition, a POST method must be called to transfer the data to the CPU.
	
	Example from AWP:
	Definition:
	
	`<!-- AWP_In_Variable Name='"startSeq"' -->`
	
	Call:
	
	`<form method="post" action="">`
	`<input type="submit" value="Start filling"`
	`style="height: 30px; width: 150px">`
	`<input type="hidden" name='"startSeq"' size="30"`
	`value="1">`
	`</form>`
	
	Appearance on the AWP: 

3.2.4 Instruction in HTML code / typical instructions

The following table shows the typical HMTL commands

Table 3-2

Command	Remarks
`<!-- … -->`	Comments or AWP command
`<form> … </form>`	Defines a form
`<h1> … </h1>`	Text heading
`<input>`	Creates a form element
`<p> … </p>`	Text paragraph
`<script> … </script>`	Defines an area for scripts (e.g. JavaScript)
`<table> … </table>`	Table
`<tr> … </tr>`	Table row
`<td> … </td>`	Table column
` `	Line break

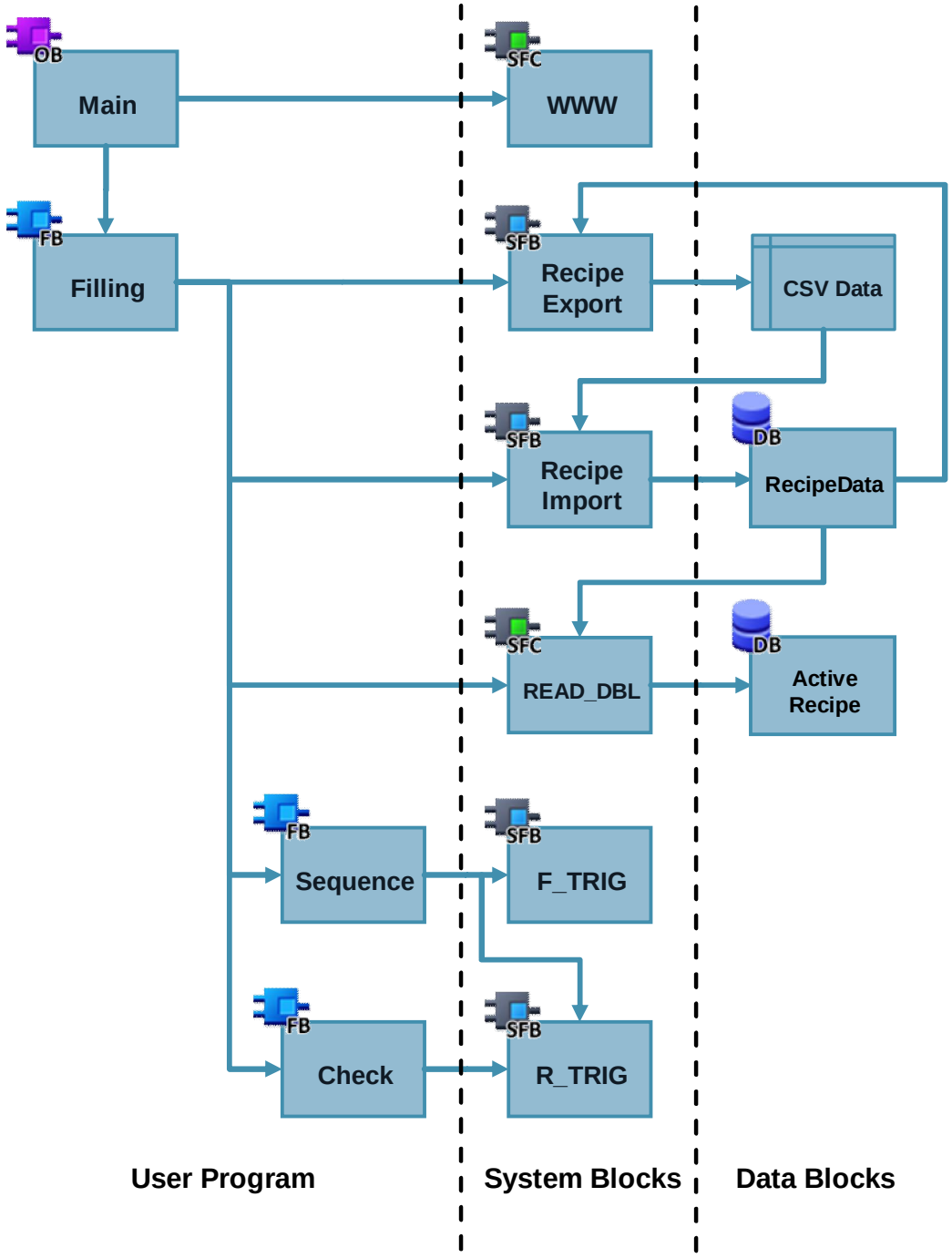
Note A previously created HMT(L) page exists for the project in the "Webpage_v11" folder.

4 Function Principle

4.1 Program overview

The call structure for the application example looks as follows:

Figure 4-1



4.1.1 PLC data type „typeBeverage“

The data type „typeBeverage“ is a self-created data type and consists the following structure:

Figure 4-2

typeBeverage

	Name	Data type	Default value	Comment
1	productname	String[20]	'Beverage_Recipe'	name of product
2	quantity	UInt	0	number of bottles to produce
3	water	UInt	0	water quantity in per cent
4	concentrateCola	UInt	0	cola quantity in per cent
5	concentrateLemonade	UInt	0	lemonade quantity in per cent
6	concentrateOrangejuice	UInt	0	orange juice quantity in per cent
7	concentrateApplejuice	UInt	0	apple juice quantity in per cent
8	spritzer	UInt	0	spritzer quantity in per cent
9	flavour	UInt	0	flavour quantity in per cent

4.1.2 PLC data type „typeProductsOfBeverage“

The data type „typeProductsOfBeverage“ is a self-created data type and consists the following structure:

Figure 4-3

typeProductsOfBeverage

	Name	Data type	Comment
1	products	Array[1..10] of "typeBeverage"	recipe data sets
2	products[1]	"typeBeverage"	

4.1.3 Data block „RecipeData“

The data block „RecipeData“ is the basis for this application. This data block contains 10 recipe data sets (PLC data type „typeBeverage“). Starting this application only the first four recipe data sets are filled up. All other recipe data sets contain zero values. All recipe data sets can be arbitrarily filled up by the user.
For each element of recipe data set initial values have to be set.

Figure 4-4

RecipeData

	Name	Data type	Start value	Comment
1	Static			
2	products	Array[1..10] of "typeBeverage"		recipe data sets
3	products[1]	"typeBeverage"		
4	products[2]	"typeBeverage"		
5	productname	String[20]	'lemonade'	name of product
6	quantity	UInt	500	number of bottles to produce
7	water	UInt	60	water quantity in per cent
8	concentrateCola	UInt	0	cola quantity in per cent
9	concentrateLemonade	UInt	40	lemonade quantity in per cent
10	concentrateOrangejuice	UInt	0	orange juice quantity in per cent
11	concentrateApplejuice	UInt	0	apple juice quantity in per cent
12	spritzer	UInt	0	spritzer quantity in per cent
13	flavour	UInt	0	flavour quantity in per cent
14	products[3]	"typeBeverage"		
15	products[4]	"typeBeverage"		

4.1.4 Data block „ActiveRecipe“

The data block „ActiveRecipe“ contains a recipe data set of PLC data typ „typeBeverage“ which can be accessed by the user (a swell as the user defined webpage).

Figure 4-5

ActiveRecipe

	Name	Data type	Start value	Comment
1	Static			
2	product	"typeBeverage"		active recipe
3	productname	String[20]	'Beverage_Recipe'	name of product
4	quantity	UInt	0	number of bottles to produce
5	water	UInt	0	water quantity in per cent
6	concentrateCola	UInt	0	cola quantity in per cent
7	concentrateLemonade	UInt	0	lemonade quantity in per cent
8	concentrateOrangejuice	UInt	0	orange juice quantity in per cent
9	concentrateApplejuice	UInt	0	apple juice quantity in per cent
10	spritzer	UInt	0	spritzer quantity in per cent
11	flavour	UInt	0	flavour quantity in per cent

4.1.5 System function “WWW”

The SFC “WWW” is called at the beginning of the user program in OB1 and is used for initializing the web server. The data blocks DB333 - DB335 are automatically created by TIA Portal transforming information of the HTML files.

Figure 4-6

Table 4-1

	Name	Data type	Description
Input	CTRL_DB	DB_WWW	Data block, which writes to user-defined web pages (web control DB)
Output	RET_VAL	Int	Error information

4.1.6 Function block “Filling“

FB “Filling” is the processing block of the actual application and is called in OB1.

Figure 4-7

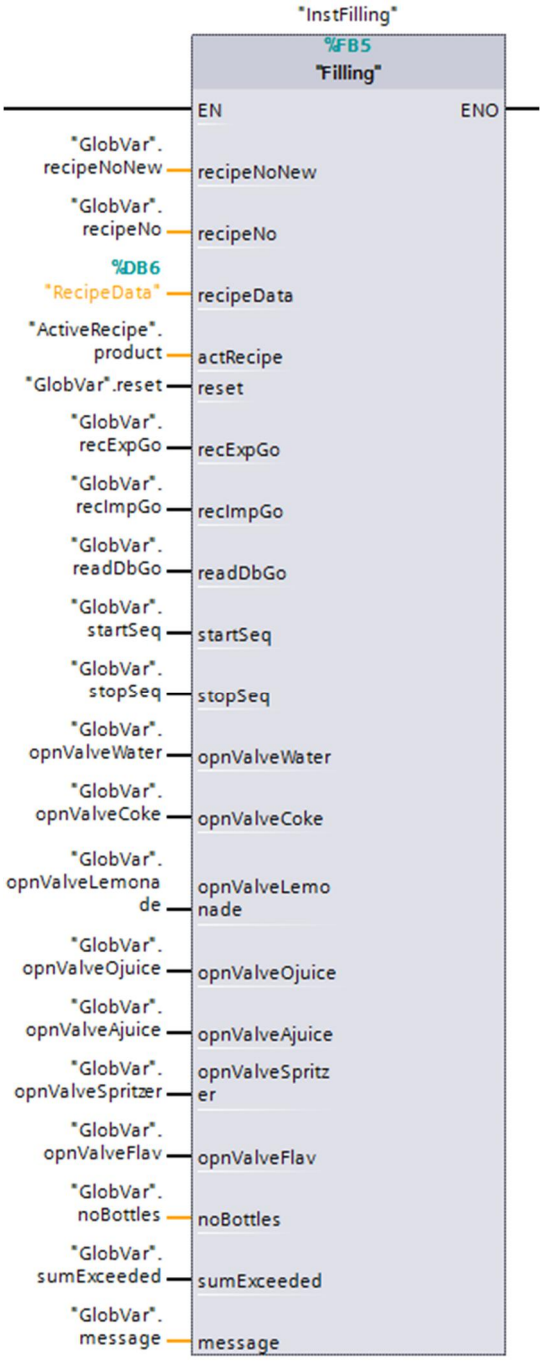


Table 4-2

	Name	Data type	Description
Input	recipeNoNew	Int	new recipe number
InOut	recipeNo	Int	number of recipe
	RecipeData	"typeProducts OfBeverage"	recipe data sets
	actRecipe	"typeBeverage"	active recipe
	reset	Bool	reset filling process
	recExpGo	Bool	start export recipe
	recImpGo	Bool	start import recipe
	readDbGo	Bool	start copy data from RecipeData to ActiveRecipe
	startSeq	Bool	start sequence
	stopSeq	Bool	stop sequence
	opnValveWater	Bool	TRUE = valve water open
	opnValveCoke	Bool	TRUE = valve coke open
	opnValveLemonade	Bool	TRUE = valve lemonade open
	opnValveOjuice	Bool	TRUE = valve orange juice open
	opnValveAjuice	Bool	TRUE = valve apple juice open
	opnValveSpritzer	Bool	TRUE = valve spritzer open
	opnValveFlav	Bool	TRUE = valve flavour open
	noBottles	UInt	actual number of bottles
	sumExceeded	Bool	sum of quantities exceeds 100%
	message	String	active filling state

4.1.7 System function block “RecipeExport”

SFB “RecipeExport” can be used for copying the recipe data block into a CSV file via the user program. This file is then located on the web server and available for further use.

Figure 4-8

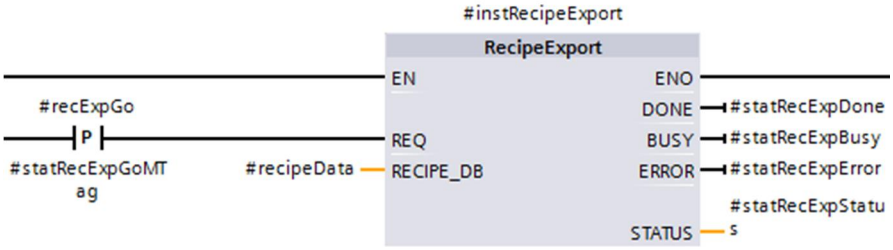


Table 4-3

	Name	Data type	Description
Input	REQ	Bool	Control parameter REQUEST: Activates the export on a rising edge.
Output	DONE	Bool	Status parameter 0: Job not yet started or still executing. 1: Job executed without errors.
	BUSY	Bool	Status parameter 0: The instruction is not executed. 1: The instruction is executed.
	ERROR	Bool	Status parameter 0: Neither warning nor error. 1: An error has occurred. STATUS supplies detailed information on the type of error.
	STATUS	WORD	Status parameter See the "STATUS" parameter table of this block in online help
InOut	RECIPEDATA	VARIANT	Pointer to the recipe data block.

4.1.8 System function block “RecipeImport”

SFB “RecipeImport” enables importing recipes located in a SCV file (on the web server). The recipes are imported into a DB which is only located in the load memory of the CPU (internal or SIMATIC memory card).

Figure 4-9

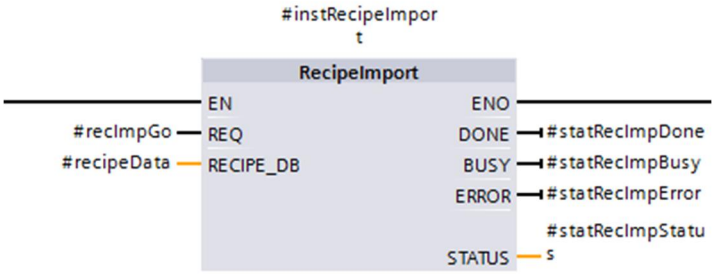


Table 4-4

	Name	Data type	Description
Input	REQ	Bool	Control parameter REQUEST: Activates the import on a positive edge.
Output	DONE	Bool	Status parameter 0: Job not yet started or still executing. 1: Job executed without errors.
	BUSY	Bool	Status parameter 0: The instruction is not executed. 1: The instruction is executed.
	ERROR	Bool	Status parameter 0: Neither warning nor error. 1: An error has occurred. STATUS supplies detailed information on the type of error.
	STATUS	WORD	Status parameter See the "STATUS" parameter table of this block in online help
InOut	RECIPEDATA	VARIANT	Pointer to the recipe data block.

4.1.9 System function “READ_DBL”

After the import process, the recipes are located in a DB in the load memory. The user program cannot access these elements. It is therefore necessary to copy the required information into the main memory. SFC “READ_DBL” realizes this very process. Only one recipe set can be copied with one call. In this application example only one active recipe set each is used. This set is then located in DB “ActiveRecipe”. The „recipeData.products[#recipeNo]“ tag at input “SRCBLK” is an indirect assignment of the respective recipe set. This notation enables selecting “recipeNo” via the HTML page.

Figure 4-10

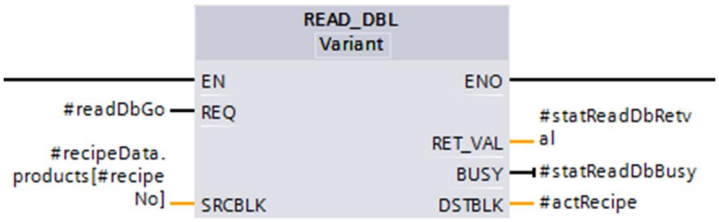


Table 4-5

	Name	Data type	Description
Input	REQ	Bool	REQ = 1: Read request
	SRCBLK	VARIANT	Pointer to data block in the load memory that is to be read from
Output	RET_VAL	WORD	Error information
	BUSY	Bool	BUSY = 1: The reading process is not yet complete.
	DSTBLK	VARIANT	Pointer to the data block in the work memory that is to be written to

4.1.10 Function block “SEQUENCE“

Function block “SEQUENCE” represents a simplified filling process. Each element in the active recipe set is polled and processed accordingly here. On the one hand, it is checked whether the total of the elements amounts to maximal 100. The result of the verification can then be evaluated at the “sumExceeded” output. If the total is in the main memory (0-100) the process will be run through. The respective current status is displayed via the “opnValve_...” outputs. If the element is on “1”, it is currently filled up. Only one element at a time can be active.
The “noBottles” output shows the number of bottles already filled up.

Figure 4-11

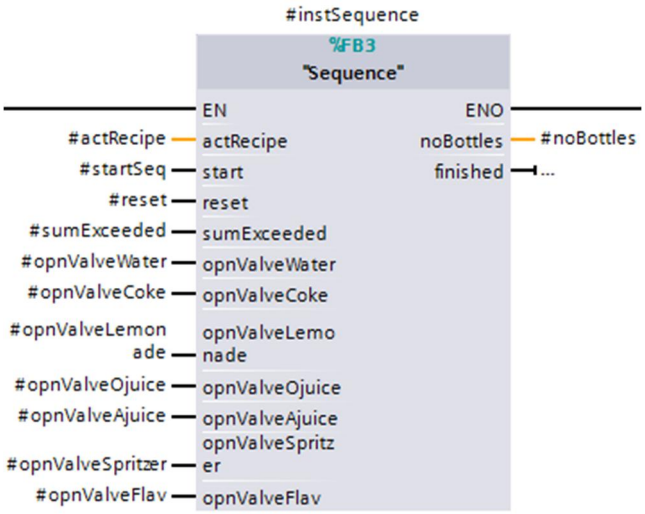


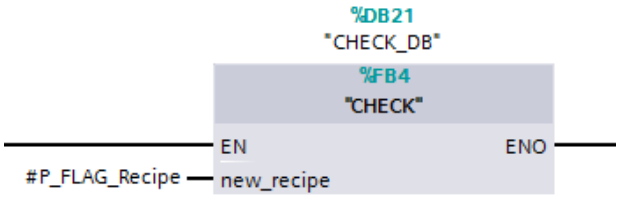
Table 4-6

	Name	Data type	Description
Input	activeRecipe	Beverage	Active recipe
	start	Bool	Start flag sequence
Output	noBottle	Uint	Actual number of bottles
	finished	Bool	Sequence of one bottle finished
InOut	sumExceeded	Bool	Sum of quantities exceeds 100%
	opnValveWater	Bool	TRUE = valve water open
	opnValveCoke	Bool	TRUE = valve coke open
	opnValveLemonade	Bool	TRUE = valve lemonade open
	opnValveOjuice	Bool	TRUE = valve orange juice open
	opnValveAjuice	Bool	TRUE = valve apple juice open
	opnValveSpritzer	Bool	TRUE = valve spritzer open
	opnValveFlav	Bool	TRUE = valve flavour open

4.1.11 Function block “CHECK”

Function block “CHECK” checks whether a new recipe set was added on the HTML page and then automatically triggers the copying process. Copying is necessary to receive the correct recipe set in DB “ActiveRecipe”. The IN/OUT parameter “newRecipe” shows with “1”, that a new recipe set has been selected. This bit is processed further in the program.

Figure 4-12



	Name	Data type	Description
InOut	RecipeNo	Int	Recipe number
Output	newRecipe	Bool	Flag new recipe

4.2 Functionality of the HMT(L) file

The user-defined web page provides an overview of the currently active recipe sets with the contained elements as well as the filling status. Furthermore, it is possible to perform various functions of the application via buttons.

Figure 4-13

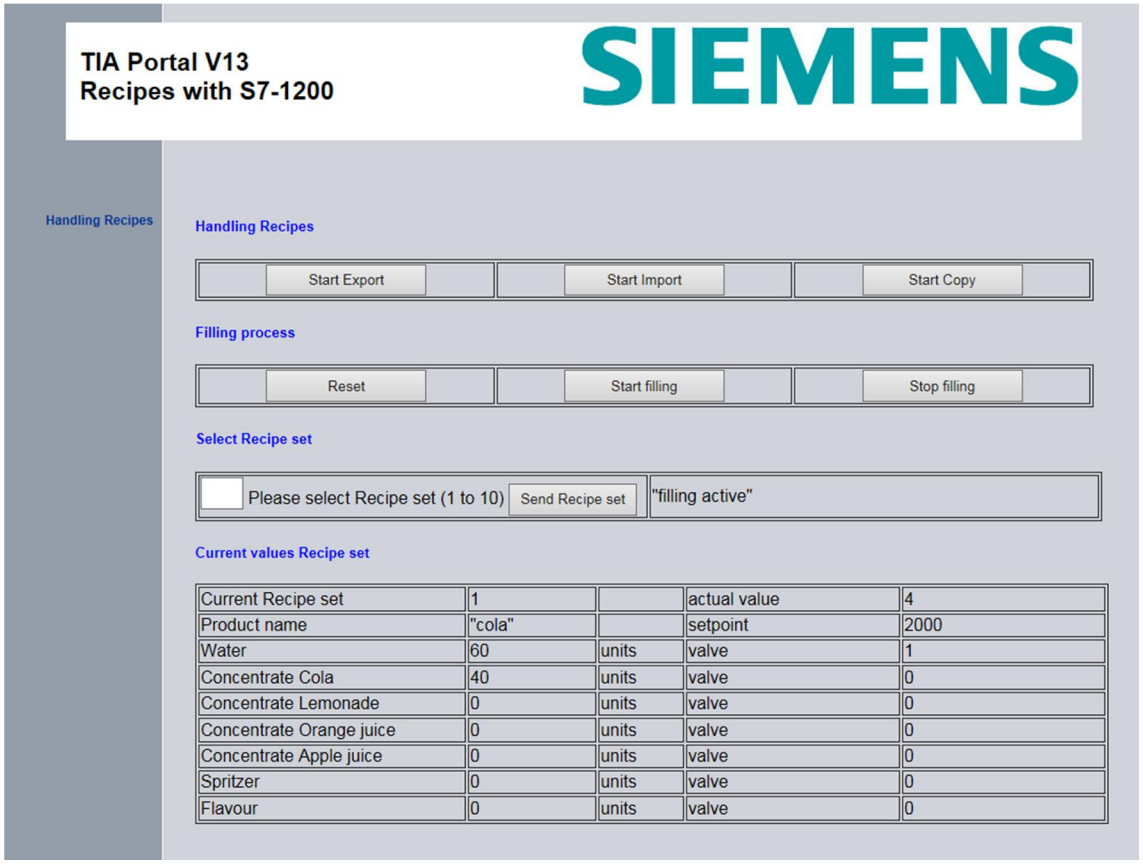


Table 4-7

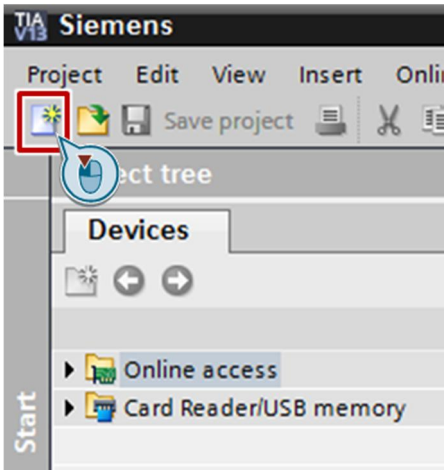
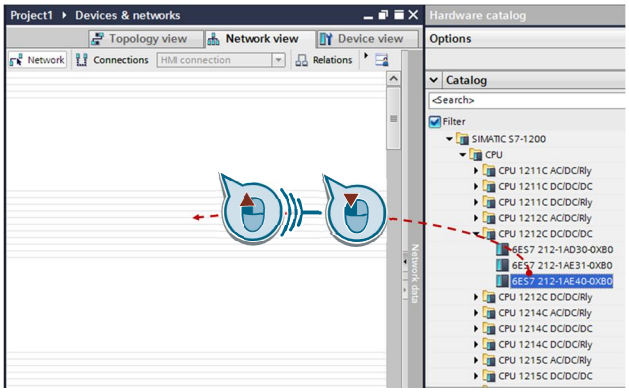
Button	Description
Start export	Clicking this button, the “RecipeExport” function is started in the user program. This process exports the content of the “RecipeData” into a CSV file. This file can be downloaded via built-in webpages of the CPU.
Start import	Clicking this button starts the “RecipeImport” function. The entire content of the CSV file is imported into the “RecipeData”. When changing the SCV file you need to make sure that not more recipe sets exist than available in the DB.
Start copy	Clicking this button copies recipe sets from load memory to work memory for instance in cases of changes of recipe sets.
Reset	This button resets the current filling process.
Start filling	Clicking this button the filling process will be started. The values of the HTML page will be updated by means of HTML page “Update_Start.html”. You can hence track how the number of filled bottles increases until it hits the target number. It is also possible to update the page manually with “F5”.
Stop filling	This button stops the current filling process.
Send recipe set	Select the suitable recipe set to be copied from DB “RecipeData” into DB “ActiveRecipe”. To do this, enter the appropriate number in the field before the button. Pressing the button then sends the value.

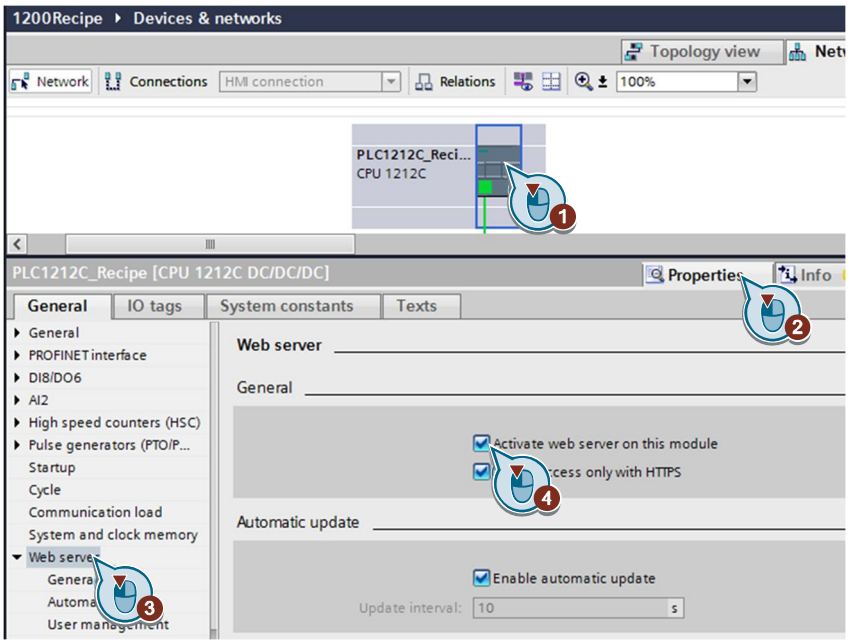
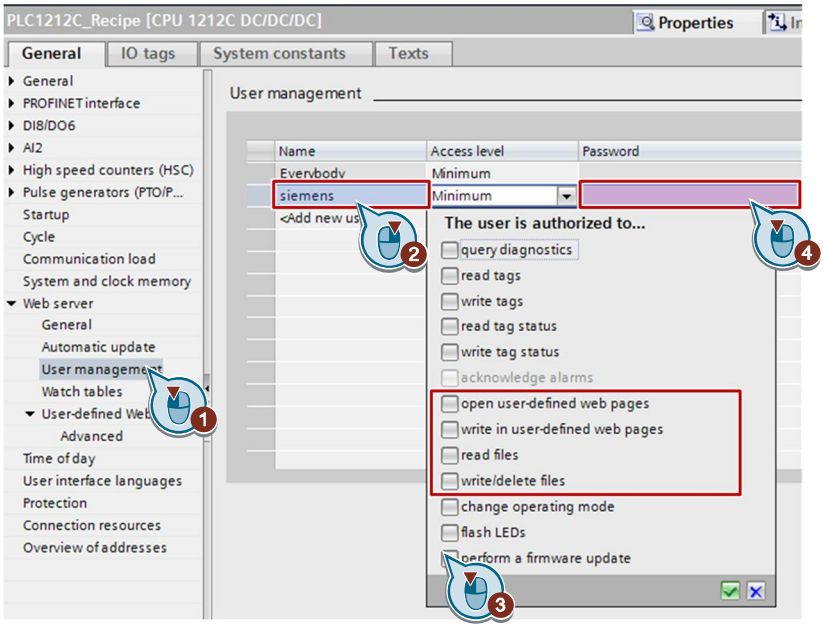
For more Information please refer to the following applications
“Creating and using user-defined web pages on S7-1200” \4\and
“Creating and using user-defined web pages on S7-1500” \10\ .

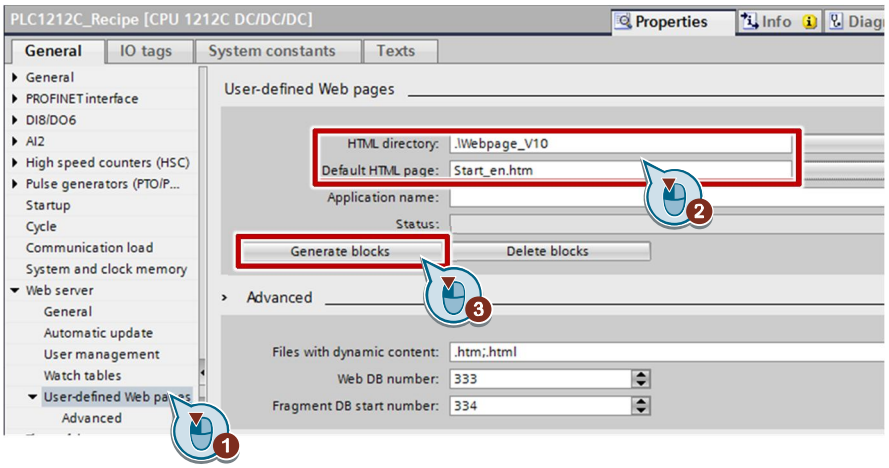
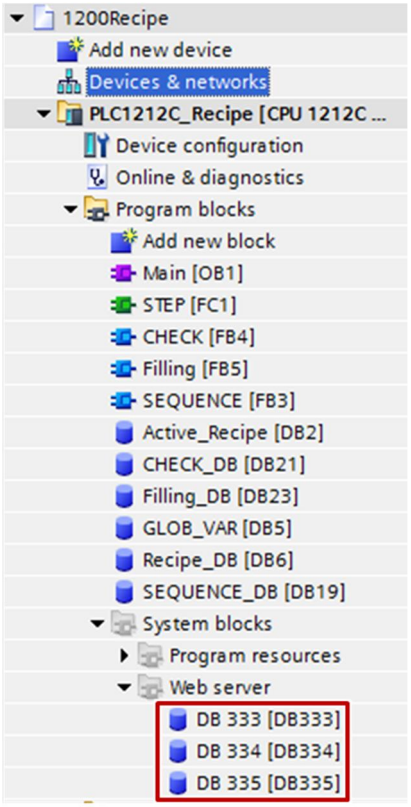
5 Configuration and Settings

5.1 Configuring the S7-1200

Table 5-1

No.	Action
1.	Open STEP 7 (TIA Portal) V13 UPD3.
2.	Create a new project. 
3.	Open “Devices & Networks”. 
4.	Select the suitable CPU and drag it into your project via “drag & drop”.  Please note, that this application example requires a CPU firmware equal to or higher than V4.0 to be able to use the described functions.

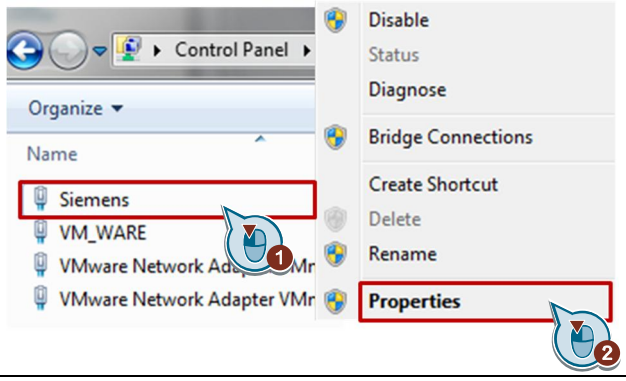
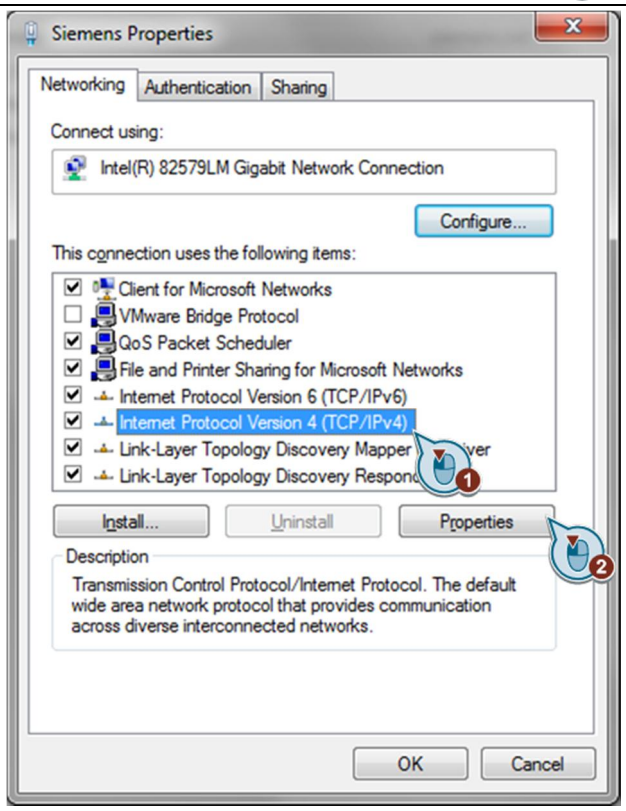
No.	Action
5.	Go to CPU properties and select the settings of the web server and activate the web server with via the check box “Activate web server on this module”.  Note: if necessary (depending on your security concept) activate/deactivate the check box “Permit access only with HTTPS”.
6.	Go to “User management” and create a user which has the required rights to call and describe an AWP. Assign the password for the user as well.  Note: you can define several users with different rights to implement different access stages. The user “Everybody” is created automatically and has the default access level “Minimum”.

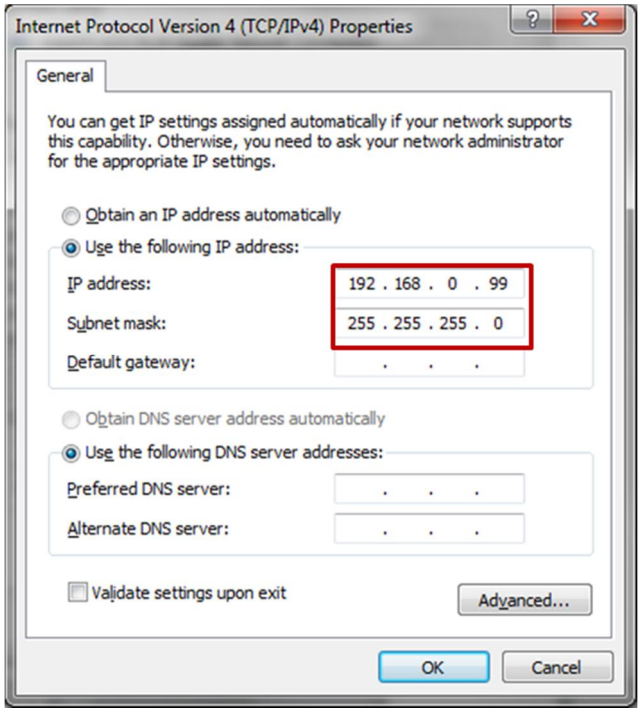
No.	Action
7.	Go to menu item “User-defined Web pages” and create the “HTML directory” in which the AWP is located and in “Default HTML page” the file itself.  Important: for the language “German” you have to select the HTML “Start_de.htm”, for English the “Start_en.htm”. Press the “Generate blocks” button to create the data blocks required for the control. These are available in the project under the system blocks in the “Webserver” folder. 
8.	Your configuration is now prepared for the download to the CPU.

5.2 Configuring the network connection

The LAN interface of the programming device requires a static IP address to configure the controller. Configuration of the LAN connection is described in the following.

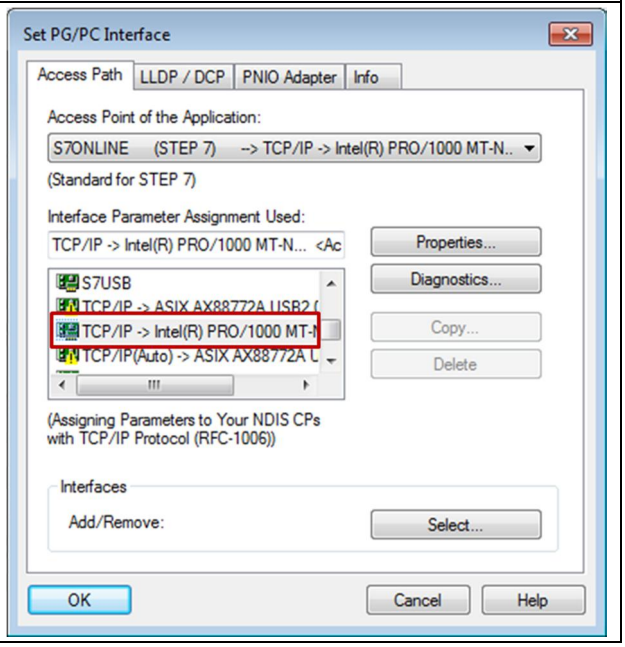
Table 5-2

No.	Action	Note
1.	Open the network connections via “Start > Control Panel> Network and Sharing > Change adapter settings”.	
2.	Select the suitable network adapter and right-click to call its properties.	
3.	In the Properties you select the internet protocol V4.	

No.	Action	Note
4.	Enter a fixed IP address including subnet mask.	

5.3 Setting PG/PC interface

Table 5-3

No.	Action	Note
1.	Open “Start > Control Panel > Set PG/PC Interface”.	
2.	Select “S7ONLINE (STEP 7)” as access point. Select your network card as “Interface Parameter Assignment Used”.	

6 Installation and Commissioning

6.1 Installing the hardware

The figure below shows the hardware setup of the application.

Figure 6-1

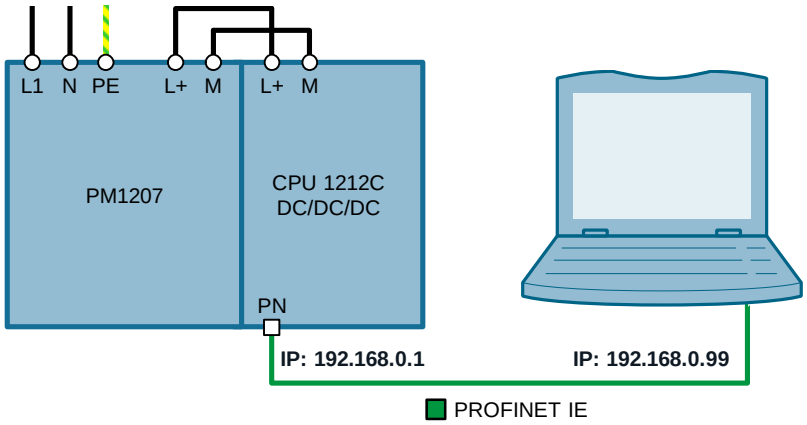


Table 6-1

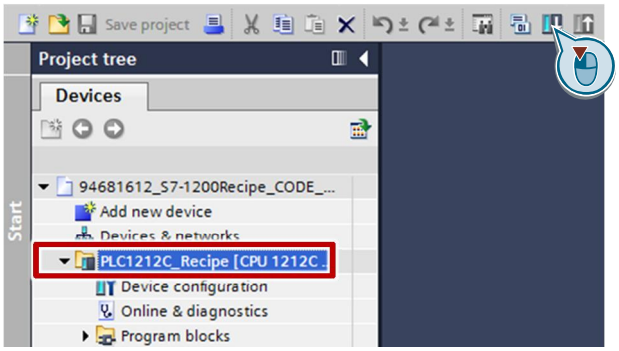
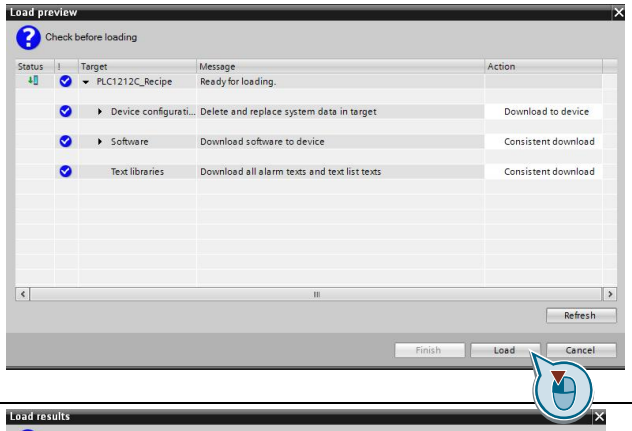
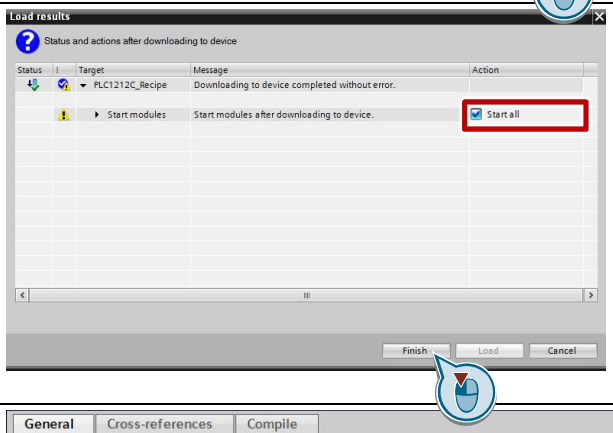
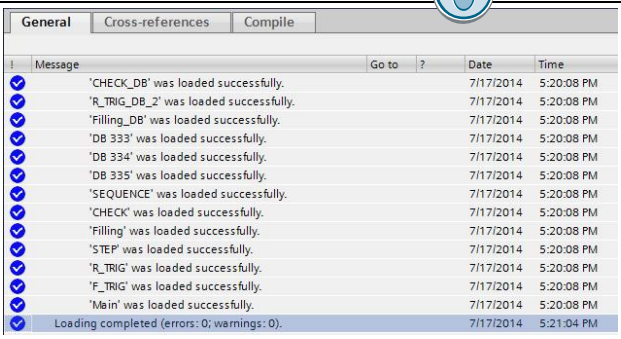
No.	Action
1.	Mount the power supply and the CPU on a top hat rail.
2.	Connect the CPU to the 24 V DC supply voltage of the power supply.
3.	Establish the bus connection (Ethernet) between CPU and PG / PC.
4.	Connect all the protective earth connections (PE) with the protective conductor.
5.	Connect the voltage supply (L1, N) with the power supply (230V~).

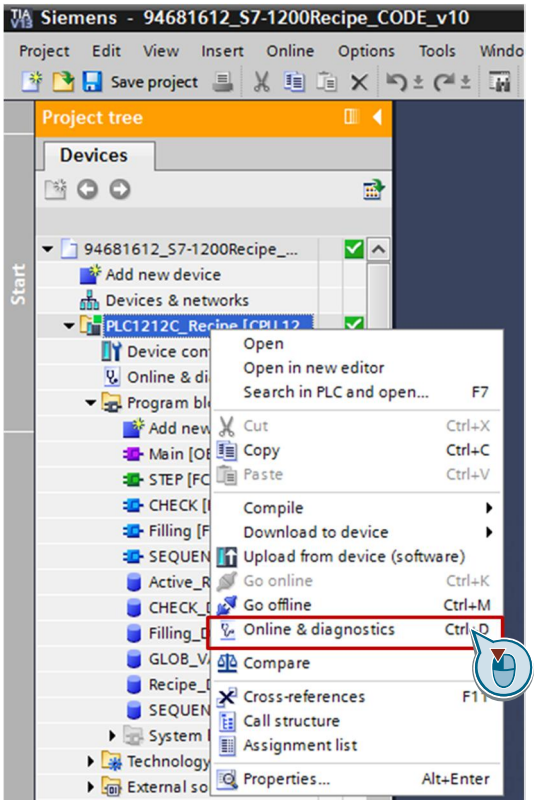
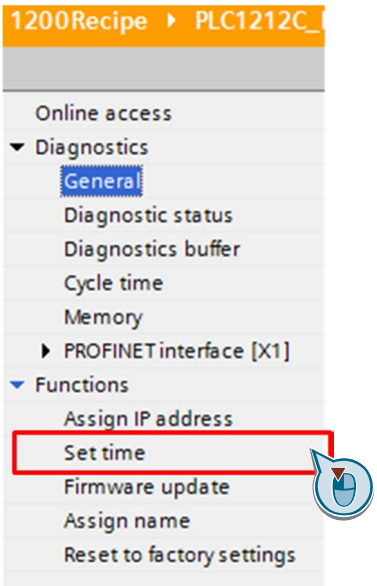
6.2 Installing and commissioning the software

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

Table 6-2

No.	Action	Remarks
1.	Download the project “94681612_S7-1200Recipe_ CODE_v11.zip” for this documentation and unzip it.	
2.	Open the contained project “94681612_S7-1200Recipe_ CODE_v11.ap13” with STEP 7 (TIA Portal) V13.	TIA V13

No.	Action	Remarks
3.	Starting from Project tree select the CPU “PLC1212C_Recipe” and click on the “Download to Device” button for download into the CPU.	 When downloading, the hardware configuration as well as the blocks (software) are transferred.
4.	The “Load preview” window shows whether all of the download conditions are met. Click on the “Load” button	
5.	The “Load results” window shows you a summary of the load process. Activate the “Start all” check box and clock on the “Finish” button.	
6.	After the end of the download process in the inspector window you check whether all blocks have been transferred correctly.	
7.	For checking the clock setting of the CPU, go online.	

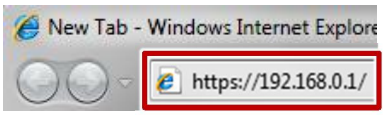
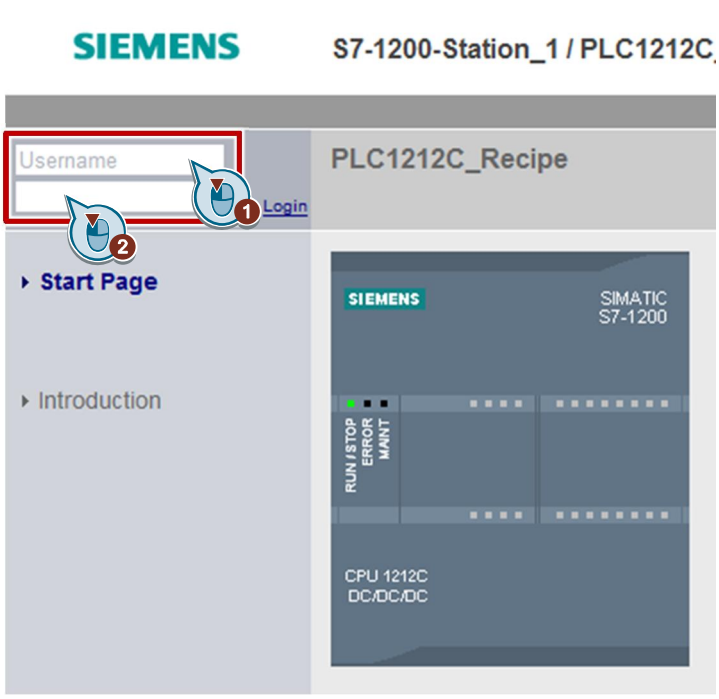
No.	Action	Remarks
8.	Go to “online” status and select the “Online & diagnostics” menu.	 You will recognize the online mode at the orange coloring.
9.	Go to “Online access” > “Functions” and select the “Set time” submenu.	
10.	Check the time and change it, if necessary.	

7 Operating the Application

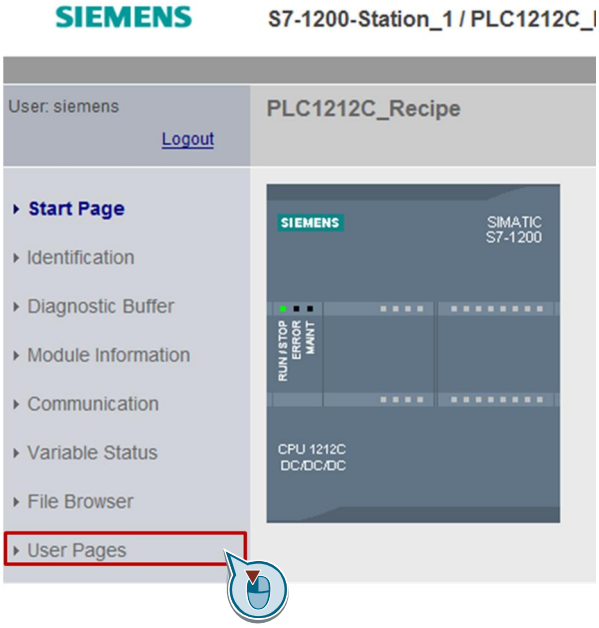
This chapter describes the operation of the application regarding exporting and importing a recipe as well as handling the filling process.

7.1 Calling the web page via the browser

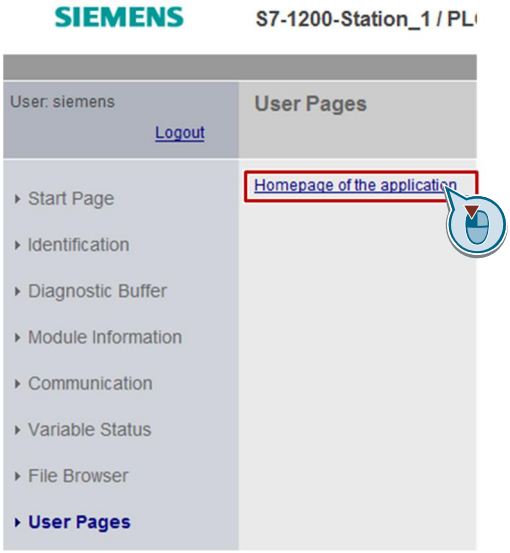
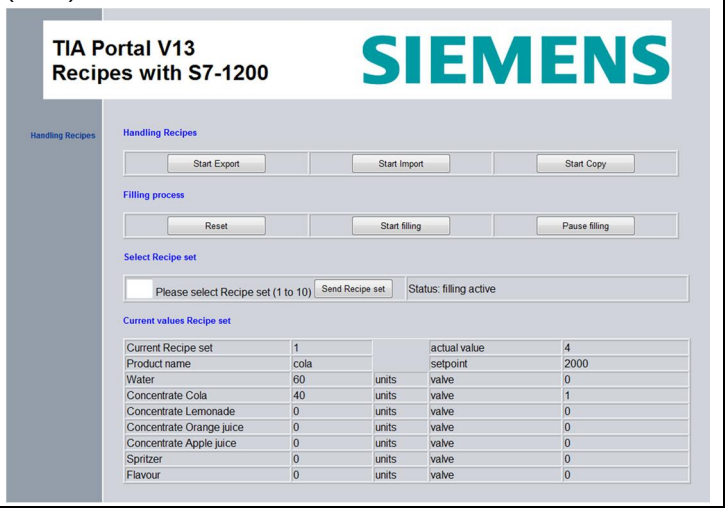
Table 7-1

No.	Action	Remarks
1.	Start your browser (e.g. the Internet Explorer).	
2.	Enter the IP address of the CPU in the address bar: Note: if an access error appears when loading the web page, follow the steps in chapter 8.2.	 The start page of the web server of the CPU follows. Here you can read only the basic information (operating state of the CPU) without logging in. Important: Calling the user-defined web page is not possible without logging in.
3.	Log in at the web server of the CPU. Several users are available. For access to the user-defined web page as well, type in the user “siemens” with the password “siemens”. “user1”, “user2” and “user3” were created as alternative users with different access stages. The respective password is identical with the user.	 The web server menu now appears completely

7.1 Calling the web page via the browser

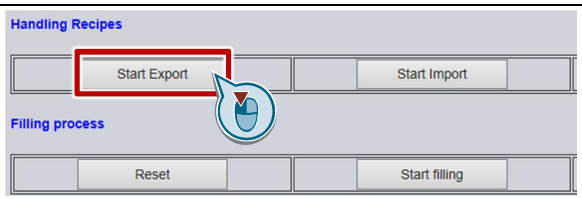
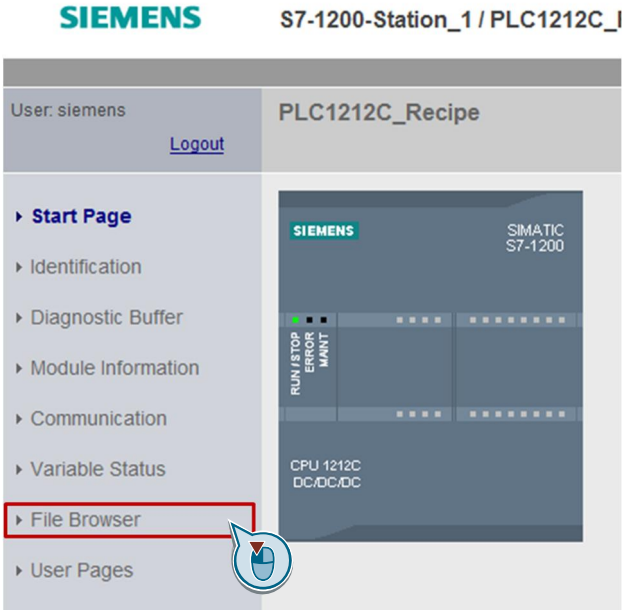
No.	Action	Remarks
4.	Select the menu item “User Pages” with a double-click.	

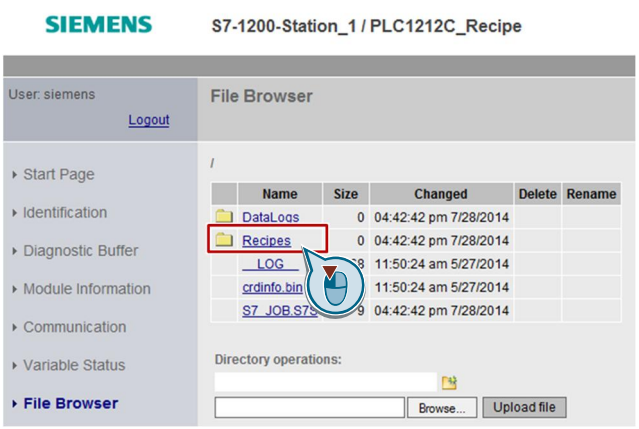
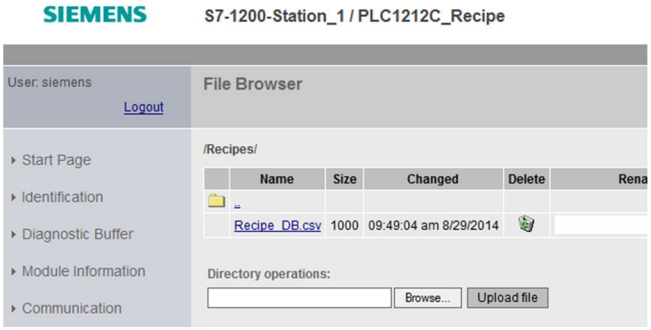
7.1 Calling the web page via the browser

No.	Action	Remarks
5.	Start the application with a left-click.	 A new browser window opens – the user-defined web page (AWP). 

7.2 Exporting the recipes in a CSV file

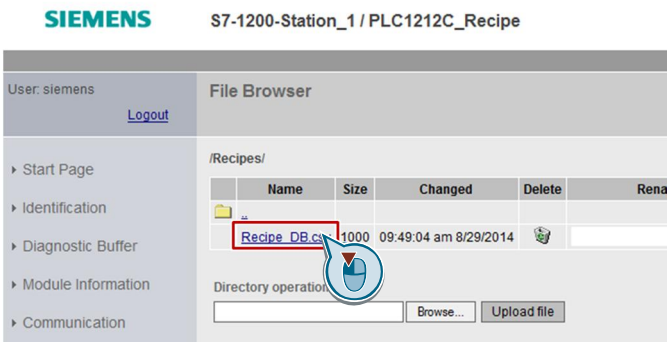
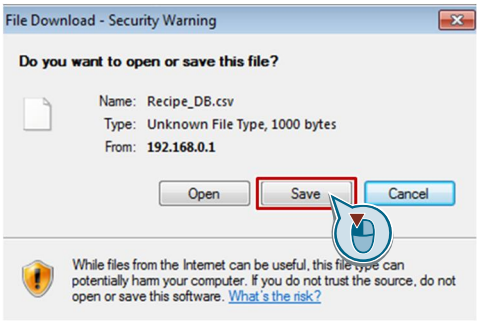
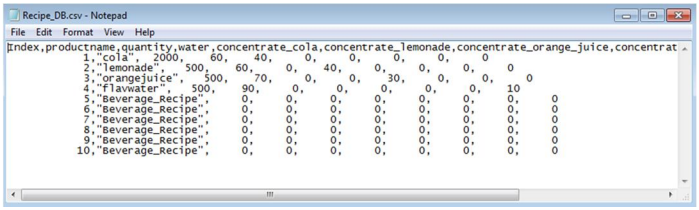
Table 7-2

No.	Action	Remarks
1.	Follow steps 1-3 in Table 7-1.	
2.	Press the “Start Export” button for exporting the data block “recipeData” into a CSV file.	 Pressing the button performs the export and starts the “RecipeExport” system function. See chapter 4.1.3.
3.	A “RecipeData.csv” file was created. Follow the next steps to be able to use this file.	
4.	Select the entry “File Browser” in the web server.	 An overview of the file structure pops up.

No.	Action	Remarks
5.	Left-click to open the “Recipes” folder.	 Now you see the content of the recipe folder.  During an export, the system always files the CSV file with the same name on the web server.

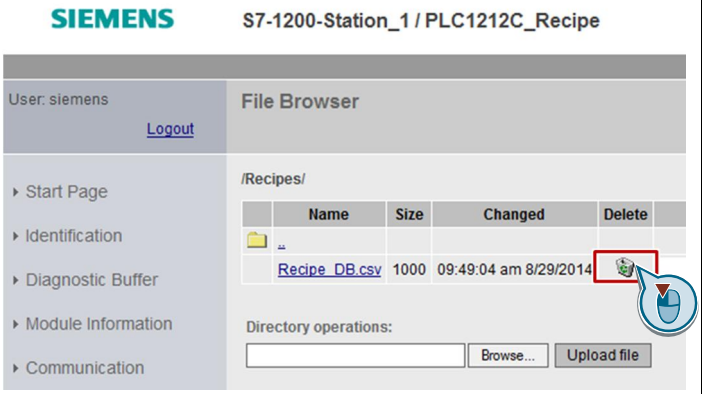
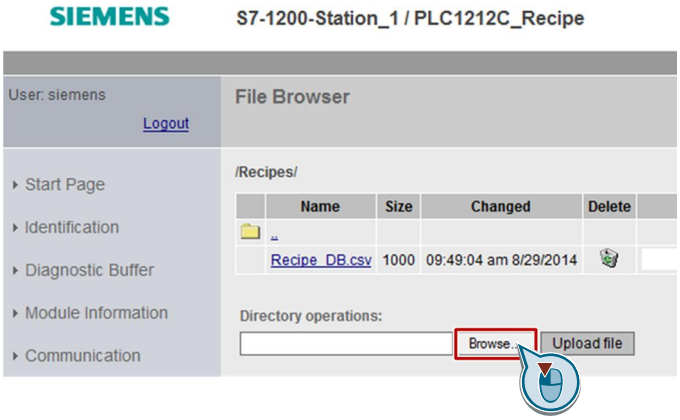
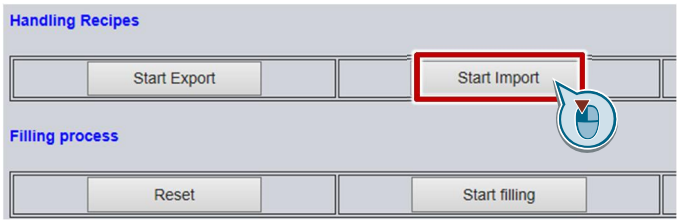
7.3 Changing the CSV file

Table 7-3

No.	Action	Remarks
1.	Save the CSV file locally on your PG/PC to be able to change the content of the recipe DB.	
2.	Acknowledge the following dialog field with “Save” and then select the suitable path.	
3.	Open the file with a suitable program (e.g. Notepad/Excel), correct the file according to your wishes and save it. For each element of the recipe set a value must be stored:	 Note: A recipe set corresponds to a line in the CSV file.

7.4 Importing the recipes from the CSV file

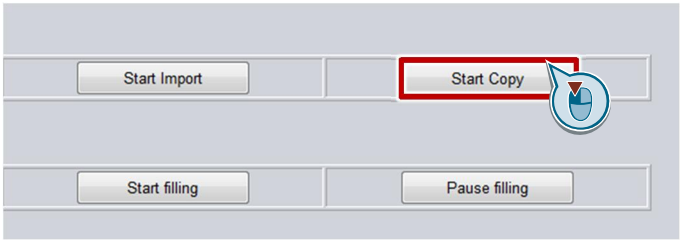
Table 7-4

No.	Action	Remarks
1.	Open the file browser and the “Recipes” folder via the menu in the web server.	
2.	Delete the previous file before starting the import.	
3.	Select the storage path of the CSV file by pressing the “Browse” button.	
4.	Click on “Upload file”.	This process loads the CSV file into the web server.
5.	Open the AWP and click on the “Start Import” button.	 The program uses SFB “Recipe_Import” for this. The values of the CSV file are now loaded to the “RecipeData”. See chapter 4.1.4. Note: the new values are currently still located in the load memory and can therefore still not be accessed in the program.

7.5 Copy a recipe set from “RecipeData” DB in “ActiveRecipe” DB

To be able to access the new values, these must first be copied from the load memory into the main memory, i.e. in the application example from the “RecipeData” into the “ActiveRecipe” DB.

Table 7-5

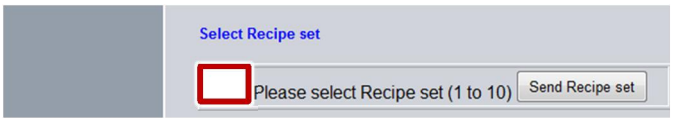
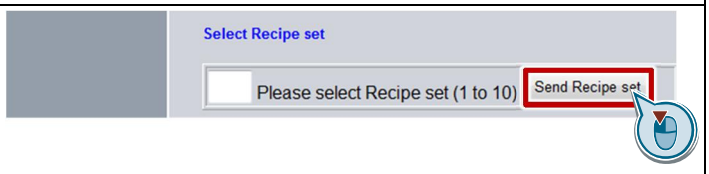
No.	Action	Remarks
1.	Open the AWP and click on “Start Copy”.	 The program uses SFB “READ_DBL” for this. With this function, only one recipe set from the “RecipeData” is copied to “ActiveRecipe” DB. See chapter 4.1.5. Now you can view the changed values in the bottom part of the AWP.

Note In connection with the “READ_DBL” function, note the following entry in the Support Portal: http://support.automation.siemens.com/WW/view/en/51434747.

7.6 Selecting the recipe set

Since for the “READ_DBL” function it is only possible to copy one recipe set, it must be selected by the user.

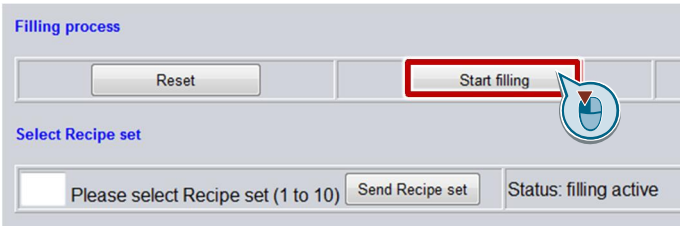
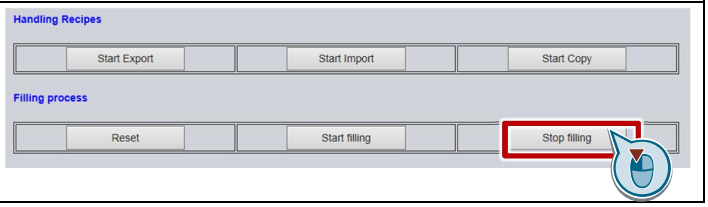
Table 7-6

No.	Action	Remarks
1.	Open the AWP. Click in the labelled field and enter a value between 1 and 10 to select the respective recipe set.	
2.	Press the “Send Recipe set” button.	 The current value is available in the “Current Recipe set” field. Note: it is only possible to change the recipe set when the filling process is not active.

7.7 Starting and running the filling process

Note The filling process shown in the application is simplified and not represented correctly.

Table 7-7

No.	Action	Remarks
1.	Select the recipe set as described in chapter 7.6.	The respective recipe set contains a “setpoint” which specifies how many bottles shall be filled. Note: at the beginning, recipe set “1” is selected.
2.	Start the filling process by pressing the “Start filling” button.	 Since the AWP is updated automatically every 6 seconds, you find the value of the already filled bottles in the “actual value” field.
3.	You can view the current status in the “Status” field.	
4.	To stop the process, press the “Pause filling” button.	

The filling process runs automatically after the start. Each recipe set consists of several elements. For each of these elements a maximum of 100 units is considered. 10 units correspond to 1 second. In total, the total value of the recipe set must not exceed 100 (verified in the program). For one bottle, the filling process corresponds to maximal 10 seconds. The user can “monitor” each valve – an opened valve corresponds to the value “1”.

Example of recipe set 1:

The elements used here are “water” and “cola”. “Water” has 60 units assigned to it (corresponds to 6 seconds) and “cola” 40 units (corresponds to 4 seconds). The total passage would then be 10 seconds. After a passage the number increases by 1 up to the “setpoint”.

7.8 End of the filling process

The end of the filling process is reached if the actual value is equal to the setpoint in the current recipe set. Or if the user stops the filling process (see chapter 7.7.). To start a new filling process, the previous one must first be reset with the “Reset” command via the AWP.

8 Further Notes

8.1 Updating the web page

The browser updates web pages automatically. This does not apply to user-defined web pages.

The refresh time of the web pages depends on the number of tags used.

A manual refreshing can be performed by pressing the “F5” button.

The optimized actualization is performed by HTML file “Update_Start.html” which is embedded as a frame (iframe) in the main page.

Figure 8-1

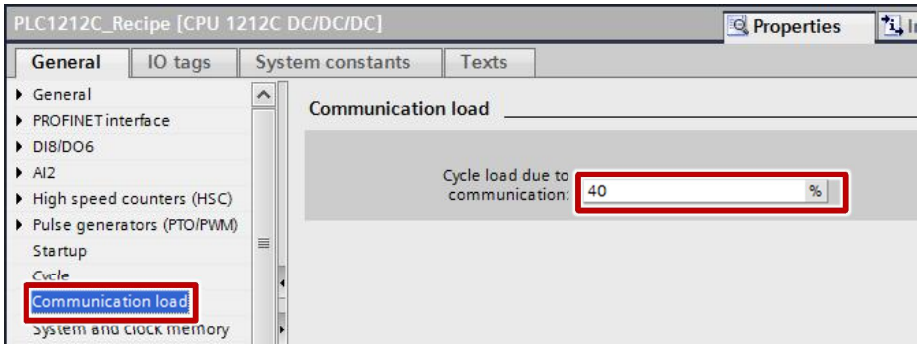
```
<iframe src="Update_Start.html" style="display:none;" />
```

Increasing the resources for the communication in the settings of the CPU may cause a reduction of the update time.

Note Further information for creating time optimized HTML pages is to be found under “Creating and using user-defined web pages on S7-1500” Basics Document in chapter1.7.

Increasing of resources for the communication in the settings of the CPU can cause a shorter actualization time.

Figure 8-2



Note The refresh time must not be selected to small since otherwise write commands to tags might not be executed.

8.2 Installing the certificate for web server

If after entering the IP address of the CPU the message “There is a problem with this website's security certificate” appears, go to the "Introduction" page and download the Siemens security certificate, for example for the Internet Explorer, as follows:

Table 8-1

No.	Action
1.	Click the "download certificate" link on the introductory page. The dialog window “File Download – Security Warning” opens.
2.	Click the “Open” button in the dialog window “File Download – Security Warning” to open the file. The dialog window “Certificate” opens.
3.	Click the "Install Certificate" button in the "Certificate" dialog, to call the assistant for importing the certificate.
4.	Follow the dialog instructions of the “Certificate Import Wizard” to import the certificate. Select the "Trustworthy root certificate authorities" certification storage.

8.3 Comparing recipes in an S7-1200 and in the HMI system

The differences between the S7-1200 and the HMI systems are given below.

- When using recipes in an S7-1200 and an AWP, is not necessary to use a panel.
- The quantity framework of the recipes is for S7-1200 based on the size of the data blocks
- When using recipes in the S7-1200 all data types possible in the controller can be accessed
- When using recipes in the S7-1200 the recipes are only processed in the CPU (recipe memory in the CPU), when using HMI systems, however, the data must be synchronized between CPU and panel.
- Simple integration of recipes into an existing program or project in the CPU
- Flexible integration of recipes into an S7-1200 since no specific visualization system is used

9 Related Literature

Table 9-1

	Topic	Title
\1\	Siemens Industry Online Support	http://support.industry.siemens.com
\2\	Download page of the entry	https://support.industry.siemens.com/cs/ww/en/view/94681612
\3\	S7-1200 manual Automation system	https://support.industry.siemens.com/cs/ww/en/view/107623221
\4\	Application “Creating and using user-defined web pages for S7-1200”	https://support.industry.siemens.com/cs/ww/en/view/58862931
\5\	TIA Portal STEP7 V13 Trial	https://support.industry.siemens.com/cs/ww/en/view/78793685
\6\	STEP 7 (TIA Portal) V13 SP1und WinCC (TIA Portal) V13 SP1	https://support.industry.siemens.com/cs/ww/en/view/109311724
\7\	Application “Quality Assurance by means of Weighing, Control and Logging with the SIMATIC S7-1200” (set 6)	https://support.industry.siemens.com/cs/ww/en/view/82454336
\8\	STEP7 SIMATIC S7-1200	Automating with SIMATIC S7-1200 Author: Hans Berger Publicis Corporate Publishing ISBN: 978-3-89578-404-0
\9\	SELF HTML (German website)	http://de.selfhtml.org http://wiki.selfhtml.org/wiki/startseite
\10\	Application “Creating and using user-defined web pages on S7-1500”	https://support.industry.siemens.com/cs/ww/en/view/68011496

10 History

Table 10-1

Version	Date	Modifications
V1.0	09/2014	First version
V1.1	07/2015	Update software and dokumentation