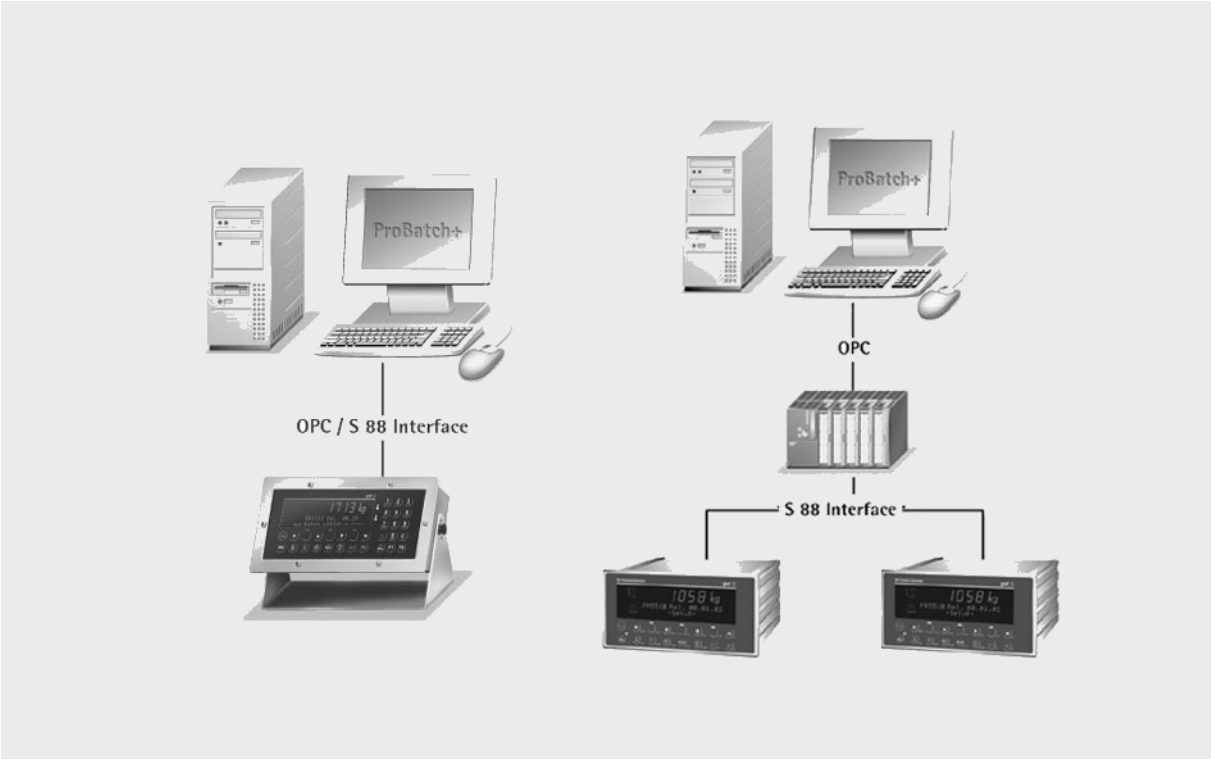
sartorius
mechatronics

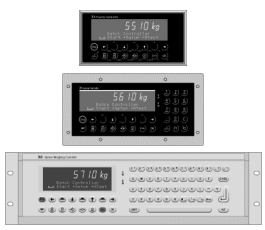
X4, X5, X6 - Application

PHASE Controller

Operating Manual



Operating Manual	9499 050 90001	Edition 1	12.2004
For PR 5510/90	Release 1.10		
For PR 5610/90	Release 1.10		
For PR 5710/90	Release 1.10		



Sartorius Mechatronics T&H GmbH, Meiendorfer Str. 205, 22145 Hamburg, Germany
Tel: +49.40.67960.303 Fax: +49.40.67960.383

Please note
Any information in this document is subject to change without notice and does not represent a commitment on the part of Sartorius. This product should be operated only by trained and qualified personnel. In correspondence concerning this product the type, name and release number as well as all license numbers in relation to the product have to be quoted.

Important

Bitte beachten
Alle Angaben in diesem Dokument sind unverbindlich für Sartorius und stehen unter Änderungsvorbehalt. Die Bedienung des Produktes darf nur von geschultem, fach- und sachkundigem Personal durchgeführt werden. Bei Schriftwechsel über dieses Produkt bitte Typ, Bezeichnung und Versionsnummer sowie alle mit dem Produkt in Zusammenhang stehenden Lizenznummern angeben.

Wichtig

CONTENTS

1 INTRODUCTION

1.1 General

1.1.1 Other manuals

This operating manual describes the operation of Phase-X5 and the differences for Phase-X4 and Phase-X6. Installation and calibration are explained in the installation manuals of instruments PR 5510 for Phase-X4, PR 5610 for Phase-X5 and PR 5710 for Phase-X6.

1.1.2 Delivery condition

The PhaseController standard version does not contain hardware options.
The following software licences are included.

- PR1713/32 S88 Phase Batching
- PR1792/13 OPC Server Licence

1.1.3 PhaseController

The PhaseController is designed for realizing batching applications with several scales, which are controlled via recipe management system PR8400 'Sartorius ProBatch+ for Windows'. The PhaseController can be used also for creation of PLC-based batching applications.
This manual explains how the PhaseController should be used related to the software. You can use this information also for writing a proxy server for a PLC, if PLC control and data guidance is required. A Proxy Server for a Siemens PLC is available. For additional information, refer to IEC 61131 library FBLib-S88 help.
Rough distinction of two different system layouts is as follows:

1. The PC with the recipe manager is connected directly with the PhaseController. Process data handling is done via the built-in inputs and outputs. With this concept, an optional PLC with individual cabling can be connected. See 6.1

2. The recipe manager is connected with a PLC. The PhaseController is connected to this PLC. Control and data must be routed by the PLC. See 6.2.

A PhaseController Phase-X5 or Phase-X6 permits connection of a second external weighing point. A weighingpoint 'C' as sum weighing point (a+B) is not supported.

Survey of other functions
- Tare function
- Configurable digital and analog inputs and outputs

Survey of optional components for the basic instrument versions:
- Digital inputs and outputs
- Serial interfaces
- Analog output (max. 2 cards, fieldbus or Ethernet are not possible)
- Fieldbus in Slot 4 (2nd analog output card and Ethernet are not possible)
- Ethernet in Slot 4 (2nd analog output card and fieldbus are not possible)
- External terminal PR5610/05 as a second operating station
- External weighing point 'B' via DIOS master or XBPI, only Phase-X5 and Phase-X6

1.1.4 Plug-in cards and slots

List of optional retro-fittable modules for PhaseControllers Phase-X4, Phase-X5 and Phase-X6. Max. 3 modules can be built in. Detailed information is given in the relevant installation manuals.

	For Phase-X5 and Phase-X6	Slot 1	Slot 2	Slot 3	Slot 4
PR 1713/04	Serial I/O RS485/422 + RS232	•	•	•	
PR 1713/06	Analog out	• x1	• x1	• x1	
PR 1713/07	1 Analog out / 4 analog in	• x1	• x1	• x1	
PR 1713/08	BCD out / 24 out, 1 in	• x2	• x3	★	
PR 1713/12	Digital I/O 4/4 opto	★	•	•	
PR 1713/13	DIOS master				
PR 1713/14	Ethernet interface				• x1
PR 1713/15	Digital I/O 4/4 relay	★	•	•	
PR 1713/17	Digital I/O 6/8 opto	★	•	•	
PR 1721/11	Profibus interface, PR 1721/21 with Phase-X6				• x1
PR 1721/12	Interbus interface, PR 1721/22 with Phase-X6				• x1

•	= Can be fitted additionally
• x1	= Note restrictions due to high current consumption! With a card fitted in Slot 4, a 2nd analog output card in Slot 1, 2 or 3 is not permissible.
• x2	= Can be fitted additionally, but the upper terminal strip is covered by Slot 2.
• x3	= Can be fitted additionally, but the upper terminal strip is covered by Slot 3.
★	= Preferable position (digital I/O is initialized, BCD card does not cover terminals)

	For Phase-X4	Slot 1	Slot 2	Slot 3	Slot 4
PR 5510/04	Serial I/O RS485/422 + RS232	•	•		
PR 5510/06	Analog out			• x1	
PR 5510/07	1 analog out / 4 analog in	• x1	• x1		
PR 5510/08	BCD out / 24 out, 1 in, CC	•	•		
PR 5510/09	BCD out / 24 out, 1 in, CE	•	•		
PR 5510/12	Digital I/O 6/12 opto	•	•		
PR 5510/13	DIOS master				
PR 5510/14	Ethernet interface				• x1
PR 1721/31	Profibus interface				• x1
PR 1721/32	Interbus interface				• x1

•	= Can be fitted additionally
• x1	= Note restrictions due to high current consumption! With a card fitted in Slot 4, a 2nd analog output card in Slot 1 or Slot 2 is not permissible.

2 OPERATOR INTERFACE

2.1 Display

PHA_TTL5

The **weight Display** allows display of 7 digits for weight plus a decimal point. The weight unit can be selected as tons, kilograms, grams or lbs. In addition to the numeric output, two text lines can be displayed. The other symbols of the display are listed below:

Status indicator	Description
B G	Gross weight is displayed **Gross = Net + Tare** (G only active in NTEP mode)
NET	Net weight is displayed
T	Tare or initial weight is displayed. This is the weight stored in the memory.

Status indicator	Description
→0←	The weight value is within center of zero
◺◿	The weight is in standstill
◇	Dosing activ
✳◇	Flashing indicates alarm

2.2 Keypad

The key functions are almost identical for Phase-X4, Phase-X5 and Phase-X6. Differences are specified. The keys and functions are:

Indicator keys	Description
	Gross weight is displayed whilst this key is pressed (*B – Brutto, German for gross*).
	Tare weight is displayed whilst this key is pressed.
	Set and reset tare. The instantaneous gross weight is stored, provided that: - weight is at standstill - indicator is not in error status

Indicator keys	Description
	Print out
	Toggle key for switching over between weighing point A, B. No WP-'B' with Phase-X4.
	Set gross weight to zero, provided that: - weight at standstill - withing zero set range - not tared

Menu keys	Description
	Exit from menu, continue operation at next higher level.
	Softkey: The function displayed above the key can be selected.
	Scroll down in the menu.
	Scroll up in the menu.
	Display of further menu functions indicated by the double arrow ⁝.

Editing keys	Description
	During editing: move cursor left and select values, when ⁝ is displayed.
	During editing: move cursor right and select values, when ⁝ is displayed.
	Enter / execute / confirm. An additional Enter key is provided for Phase-X6.
	Backspace / delete; as a second function via key  with Phase-X4

Function keys	Description
	Stop a running phase

Function keys	Description
	Programmable function key, without function for PhaseControllers.
	Programmable function, without function for PhaseController.

2.3 Entry of alphanumeric characters

In the alphanumeric input mode, a blinking cursor is displayed in the input field. For access to this mode, press an alphanumeric key. Unless the cursor blinks in the input menu with Phase-X4, the More key must be pressed first.

Several functions are allocated to each alphanumeric key. When pressing a key once, the first character, e.g. 'A', is displayed in the cursor position. After pressing twice, 'B' is displayed in the cursor position, and 'C' is displayed after pressing three times, etc. After the last possible function, the first one is displayed again.
Completing the input of a character is done by pressing another character key or key arrow right.

Press key arrow left to return to the previous character.
Press the delete key C to remove the character from the display.

If only numeric values need to be entered, letters are not enabled. I.e. entry of values like 555 is possible by pressing the key three times without the arrow key.
With Phase-X4, the numbers and letters are given at the bottom right on the keys because of the double allocation of functions..

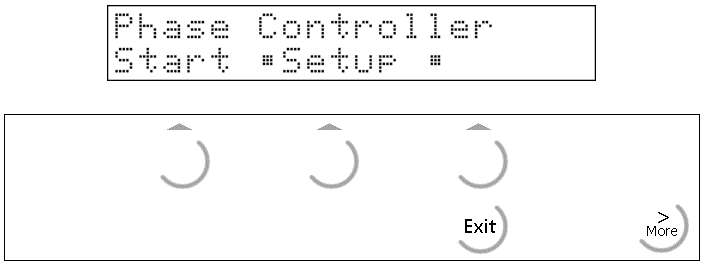
Key X5	Key X4	Character With X4: only via key More	Remark
1 #"()=	1 #"()=	# " ( ) = $? ! % 1	Comma, decimal point or colon can be entered using the dot key.
2 ABC	2 ABC	A B C a b c 2	
3 DEF	3 DEF	D E F d e f 3	
4 GHI	4 GHI	G H I g h i 4	
5 JKL	5 JKL	J K L j k l 5	Values with polarity sign are entered also by pressing the dot key, once for minus and twice for plus.
6 MNO	6 MNO	M N O m n o 6	
7 PQRS	7 PQRS	P Q R S p q r s 7	
8 TUV	8 TUV	T U V t u v 8	Complete each entry with key OK, also with the enter key for Phase-X6.
9 WXYZ	9 WXYZ	W X Y Z w x y z 9	
. -+*/	. -+*/	— + * / : ; _ ' & , < >	A space can be entered using key 0 ÄÖÜ.
0 ÄÖÜ	0 ÄÖÜ	Ä Ö Ü ä ö ü ß 0	

2.4 Operating concept

For simplification, only the operation of PhaseControllers Phase-X5 is explained, because the operation of the other PhaseControllers is nearly identical. Where applicable, differences are explained.

2.4.1 Operation via softkeys

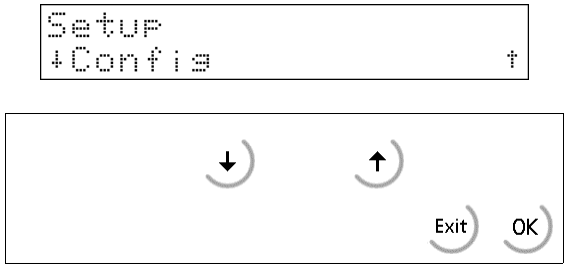
The PhaseController operation is menu-guided. For this purpose, the controllers are provided with a softkey functionality. The function of the three softkeys with the upward arrow below the numerical display is as shown above the keys. For Phase-X6, the key layout is .



Selection of menu items is by pressing .

If more than three functions can be selected, double arrows indicate that further functions can be displayed and called up by means of key or and . In section 2.3.3, the More keys and their function for Phase-X4 and Phase-X6 are also explained.
Exit from a menu is possible by pressing key . After pressing this key, operation is continued at the next higher level.

2.4.2 Selection via scroll keys



Key arrow down can be pressed to go down, key arrow up can be pressed to go up in the menu.

Press key to leave the menu and to continue operation at the next higher level.

Select the menu item displayed between by pressing key .

2.4.3 Selection via key More

If the double arrow is displayed between the function texts on the display line for the softkey functions, additional functions are available after pressing key . Unlike Phase-X5, key is located at the bottom left with Phase-X4. With Phase-X6, key is located at the bottom right of the alphanumeric keypad.

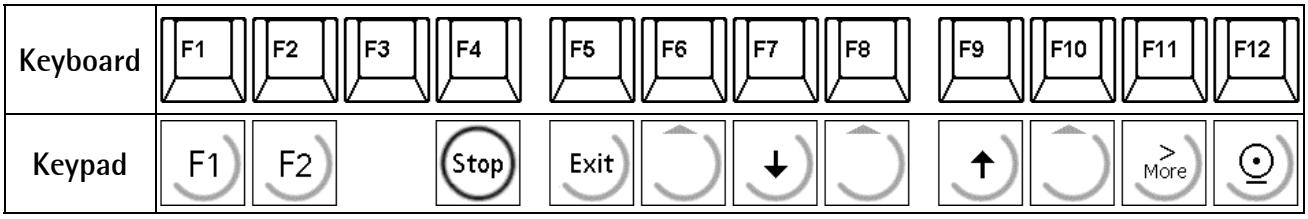
In the case of Phase-X4, another difference of the key function relates to text entry or editing. The cursor does not blink after selecting [New] and entry of a text number for a new text! Press key , until the relevant LED is lit , and the cursor blinks. Now, the text can be entered. When pressing key again, the LED is off and the text can be entered character by character by means of keys or . Press key to complete the entry.

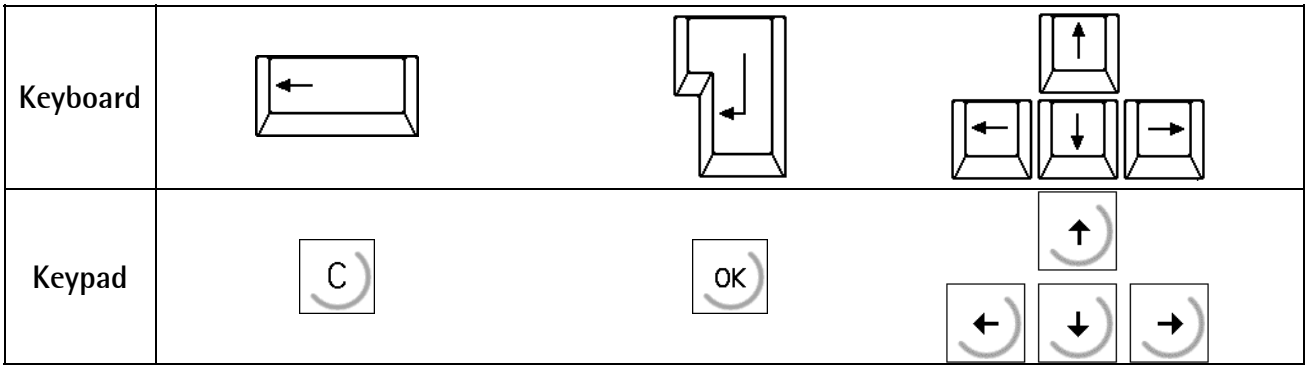
At a text selection, the last selected text is displayed. Scrolling through the existing texts by means of the scroll keys and , selection using and editing by pressing key as described above are possible.

Alternatively, press keys and to enter the number of an existing text. In this case, editing is done also as described above.

2.5 Input over external PC-keyboard

The PhaseController have an alphanumeric key field and a connection for a PC keyboard with DIN-Plug (X5, X6) and PS2-Plug (X4) on the rear side of housing. Thus the operation of the Batch Controller can be made also by an external PC keyboard. Both functions are equivalent and are parallel applicable.



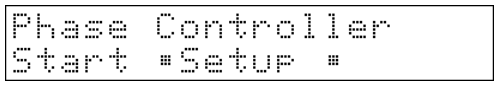


In the delivering condition the external keyboard is adjusted as an US keyboard. If a German keyboard will be used, you have to change the character set with [Strg][F2] into German. With [Strg][F1] you can return again to the delivering condition (US).
The LEDs from the PC keyboard will be not triggered.

For detail informations please refer to the Installation Manual.

3 MAIN MENU

As soon as the instrument is ready for operation, the main menu functions are accessible.



Press the relevant softkey to select the menu items.

Softkey	Function
[Start]	Main program
[Setup]	Configuration, calibration, serial port determination, initial data

3.1 Switching on the PhaseController

After switching on the supply voltage, the name of the PhaseController is shown on the alphanumeric display. The main menu is activated only now.

Installing further options cards, or re-inserting them into another socket must be done before data entry. After installation of an option, a [Cold] start is required.

The boot menu can be reached in the following three ways:
1. Menu [Setup]->[Reboot], [Warm] start possible.
2. Press key Stop when switching on the instrument.
3. Press keys Stop + Exit, ([Setup]->[Software Parameter]->[Reset on stop+exit] = 1 or 5 s.), whereby [Warm] start is not possible. . This key combination should be set to "disabled".

Please, note that a [Cold] start is permissible only in defined cases (e.g. after installing a new options card), because all data which are not stored in EPROM or EAROM will be lost !!

For further details, see the relevant chapter in the installation manual.

4 SETUP

4.1 Setup menu

The [Setup] menu is described in the installation manual. **In this manual**, only the selection of the application-specific configuration parameters is explained.

To prevent the access of unauthorized persons to the set-up menu, disabling by means of a key switch is possible.

Please, note that the calibration data must not be changed any more after entry of weight data. When changing e.g. the weight unit from kg into lb subsequently, a cold start is required. All RAM data (database tables, etc.) are lost.

The structure of the setup menu is:

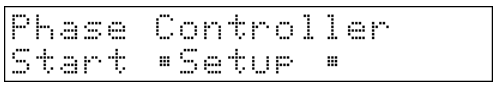
Setup

Configuration	**Follows in this section**
- Weighingpoint(s)	in the installation manual
- Set Clock	in the installation manual
- Serial Ports	in the installation manual
- Software Parameter	in the installation manual
- Show Boardnumber	in the installation manual
- Licence Setup	in the installation manual
- Print Setupdata	in the installation manual
- Print last fault	in the installation manual
- Refresh Display	in the installation manual
- I/O Slots	in the installation manual
- Show Version	in the installation manual
- Enable download	in the installation manual
- Reboot	in the installation manual

4.2 Configuration menu for PhaseController

The specific configuration data are entered in this menu. These data are stored in RAM and a back-up is provided in the EAROM.

Press to select [Setup] .

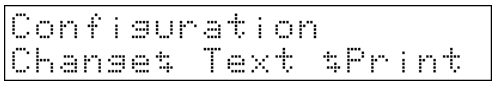


Press or to select [Config] and confirm with .

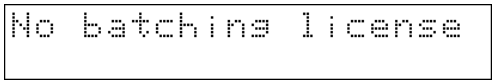
Leave the setup menu with .

Note:
Before handling 'Config', all other configurations should have been handled already. 'Software Parameter' contains a value for 'S88.01 interface'. This parameter for the firmware must be set to 'off', although the PhaseController communicates according to the rules for S88.
Fieldbus parameter 'Scale interface' in the PhaseController must be set to 'disabled'.

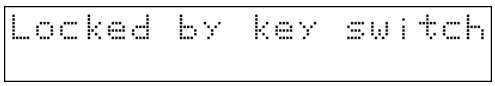
With "Change", the configuration parameters can be changed.
"Text" permits editing of the pre-set messages for the manual and dialogue phases.
Press "Print" for print-out of the configuration parameters on the printer interface.



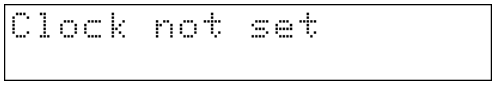
If licence PR1713/32 S88 Phase Batching is not provided, a warning message is displayed during two seconds.



Activate the "key switch locking" with the SPM-Bit MX08 to prevent access to the configuration. A warning message is displayed during two seconds.



Unless the system clock is not running (clock not set), an error message is displayed.
The configuration is finished. With Phase-X4, the clock runs immediately after switching on.

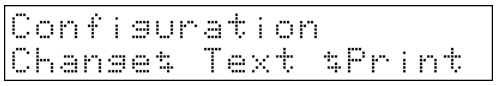


4.2.1 Config menu tree

Config	Configuration for PhaseController
- Input config.	Configuration of inputs
- Output config.	Configuration of outputs
- Limit config.	Configuration of two limit values for WP A and WP B
- Parameter config.	Parameters for local start of a recipe at the instrument (Remote operation of the PC).
- Simulation	Weighing point simulation for commissioning

4.2.2 Changing the configuration

Press [Change] to change the configuration parameters.

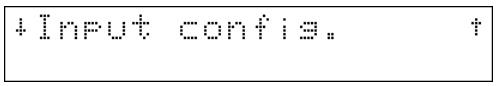


4.2.2.1 Digital input configuration

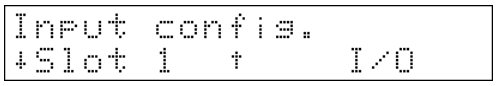
An address can be allocated to the individual inputs so that the input status is copied continuously into this address. All addresses of the virtual SPM and the related functions are possible. See chapter Virtual SPM.

Generally, more than one input can be allocated to an address. In this case, the input with a higher card number and / or input number will overrule. Inputs without function are given address '-1' or '0'. The card type and the available inputs and outputs are detected automatically. Functions for a second weighing point can be selected, but without a second weighing point, they are not allocated.

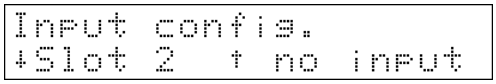
Select the parameter [Input config.] by pressing OK.



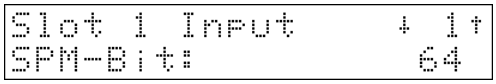
A configurable card with digital inputs fitted in this position is indicated with "I/O". Select the card position and press OK.
For Phase-X4, the menu indicates Slot 1 instead of card 1.



Unless a card with digital inputs is fitted in this position, "no input" is displayed. The card is not usable for an input configuration.



Select the input by means of the cursor keys. Enter the address for this input.
Exit finishes the entries for this card position.



Configure other plug-in cards as described above. Press Exit to finish the input configuration.

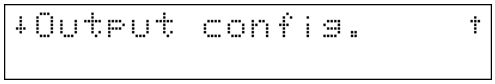
4.2.2.2 Input configuration of analog card PR1713/07 (PR 5510/07 for Phase-X4)

With an analog input card PR1713/07, no values need to be configured. The four analog values of the card with the lowest slot number are copied into word addresses MW 4 to MW 7 automatically.
See chapter Virtual SPM.

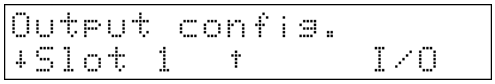
4.2.2.3 Digital output configuration

An address can be allocated to the individual outputs so that the state of this address is copied to the output continuously. All addresses of the virtual SPM and the related functions are permitted.
Outputs without function are given address '-1' or '0'.
Card type and available inputs/outputs are detected automatically. Functions for a second weighing point can be selected, but without a second weighing point, they are not allocated.

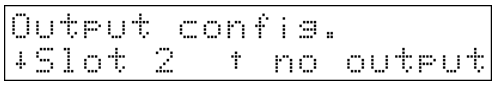
Select parameter [Output config.] by pressing OK.



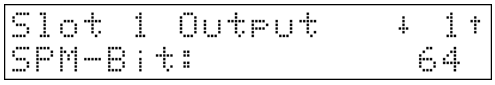
A configurable card with digital outputs fitted in this position is indicated by "I/O". Select the card position and press OK. For Phase-X4, the menu indicates Slot 1 instead of card 1.



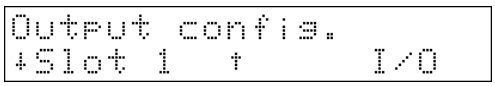
Unless a card with digital outputs is fitted in this position, "no output" is displayed. The card is not selectable for an output configuration.



Select the output by means of the cursor keys. Enter the address for this output. Configure further cards as described above.
Exit finishes the entries for this plug-in card position.



Exit finishes the output configuration.



4.2.2.4 Output configuration of BCD card PR1713/08 (PR 5510/08 for Phase-X4)

The BCD card can be inserted into any slot, however, Slot 3 is preferable for Phase-X5 due to mechanical reasons. The mechanical limitations for mounting adjacent cards are described in the installation manual.
See also chapter 1.1.4. Switch-over from BCD to digital outputs deletes all output functions for this slot.
Card PR1713/08 is configurable as

1) digital output card with 24 outputs and one input. Configurable like a digital I/O card.

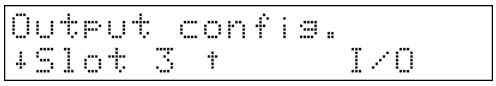
2) scale BCD output for
- gross weight,
- net weight,
- tare or
- following the display.

5 decades are displayed. On scales with more than 5 digits, only the first 5 digits are displayed. All data relate to the weighing point selected in parameter "Source of data".

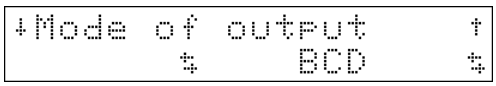
Data output:

Bit 0 to 19:	5-digit weight value
Bit 20:	negative
Bit 21:	standstill
Bit 22:	valid value
Bit 23:	tared

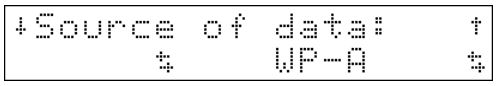
Select the slot on which the PR1713/08 card is fitted and press OK.
The mode selected last (digital or BCD) is displayed.



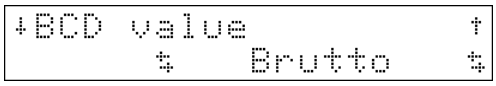
Select mode "digital" and press OK. Continue operation as with normal digital I/Os (see above)
or
select mode "BCD" and press OK.



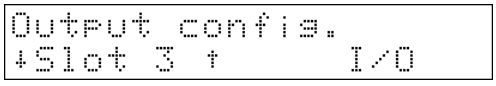
Selection is from
- WP A
- WP B , only with Phase-X5 and Phase-X6



If a weighing point was selected (WP-A, WP-B), the weight mode can be displayed:
- gross weight
- net weight
- tare
- following the display



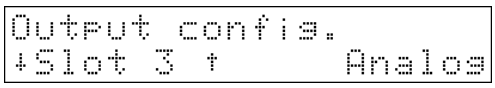
Press Exit to finish the slot configuration.



4.2.2.5 Output configuration of analog card PR 1713/07 (PR 5510/07 for Phase-X4)

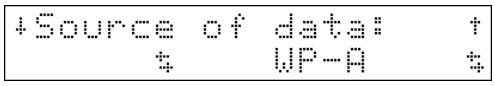
The analog output can be inserted on Slot 1, 2 or 3 (only Slot 3 for X4). Generally, selection is between weighing data and SPM values. Dependent on data source, further parameters are configurable. See also chapter Plug-in cards and slots.

Select the slot on which the PR1713/07 card is fitted.



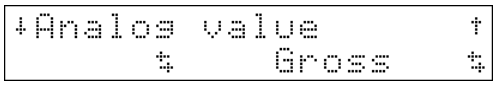
Select parameter "Source of data":

A	weighing point A
B	weighing point B , only with Phase-X5 and Phase-X6
External	from the value in SPM address MW3 (DWORD) in the virtual SPM. [1]



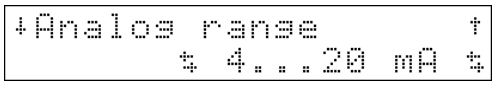
Only WP: Select the value from

Gross	always gross weight
Net / gross	net weight -> in tared condition, gross weight -> in non-tared condition
Net / 0mA	net weight -> in tared condit., 0mA -> in non-tared condition
Net / 4mA	net weight - > in tared condit., 4mA -> in non-tared condition
Net / 20mA	net weight in tared condit., 20 mA in non tared condition-> 20mA



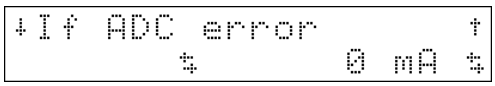
Select parameter "Analog range"
Select the following options using the scroll keys:

4..20mA:	0 to FSD results in output 4 ... 20mA
0..20mA	0 to FSD results in output 0 to 20mA



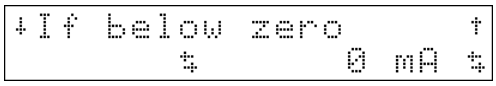
Only WP: Select parameter "If ADC error"
Press the scroll keys to select the following options:

0mA	analog error results in output 0mA
4mA	analog error results in output 4mA
20mA	analog error results in output 20mA
hold	The last value is held



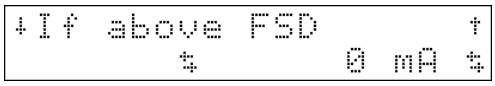
Only WP: Select parameter "If below zero"
Press the scroll keys to select the following options:

0mA	below zero results in output 0mA
4mA	below zero results in output 4mA
20mA	below zero results in output 20mA
hold	The last value is held



Only WP: Select parameter "If above FSD"
Press the scroll keys to select the following options:

0mA	above FSD results in output 0mA
4mA	above FSD results in output 4mA
20mA	above FSD results in output 20mA
hold	The last value is held



"Exit" finishes the slot configuration.

[1] Scaling: output current = *value in MW3* * 1 µA (within 0 and 20 mA)

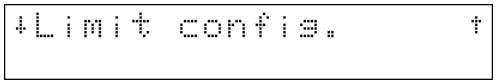
4.2.2.6 Entry of limit values

For each weighing point, two limit values can be entered. The evaluation signal of each limit value can be used e.g. as a digital output. The limit values are stored as a sequence of numbers (as on the weight display). If the scale of the relevant WP is changed, the weight value might change. After changing the scale, the limit values should be checked. Each limit value comprises two parameters: switch-on point and switch-off point for definition of a hysteresis. All limit values are entered according to the same schematics.

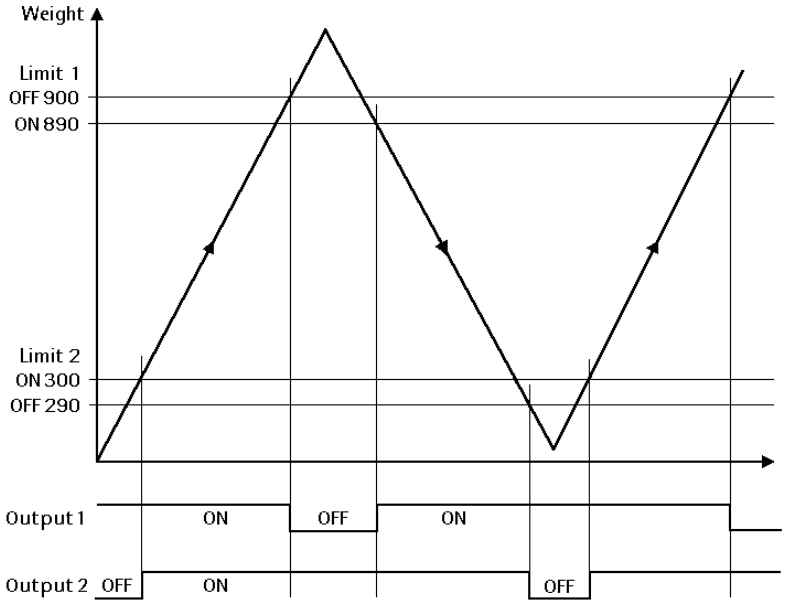
The limit values for WP-A are:
- "WP-A: limit 1 on"
- "WP-A: limit 1 off"
- "WP-A: limit 2 on"
- "WP-A: limit 2 off"

With a second weighing point, another 4 values must be configured (omitted for Phase-X4).

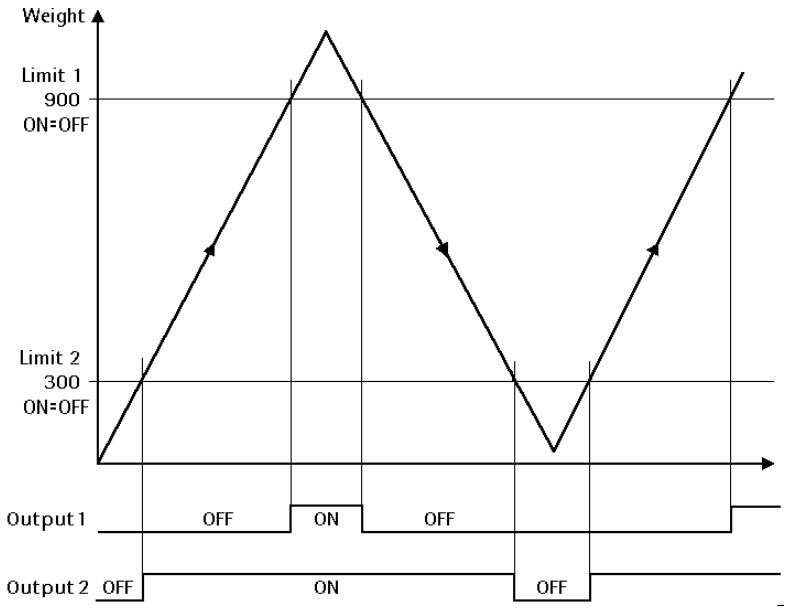
Select parameter [Limit config.] by pressing OK.



Position the cursor keys on the limit value to be configured and enter the required value.



Example:
The output signal of limit value switch 1 switches OFF above 900 kg, limit value switch 2 switches OFF below 290 kg. The two limit value switches have a hysteresis of 10 kg. With power failure (the two limit value switches are OFF), the switches indicate underfill and overfill simultaneously.

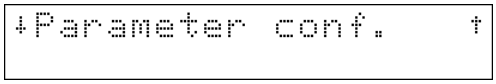


If the limit values for ON and OFF are equal, the limit value switch switches ON, when the weight exceeds the value, and OFF, when the weight drops below the value.

4.2.2.7 Parameters for recipe start

The recipe start dialogue is used for starting a recipe locally at the instrument (Remote PC operation). Configuration of the recipe start dialogue permits definition which parameters should be used and how they should be used. The dialogue is very similar for all parameters.

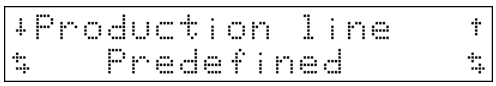
Select parameter with OK.



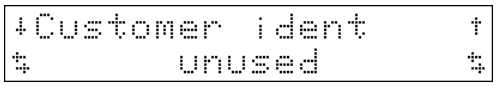
The start parameter adjustment can be different.

Unused: the parameter remains empty
Input: the operator must enter the parameter during recipe start.
Predefined: a fixed value is used as a parameter. In this case, the parameter must be entered now.

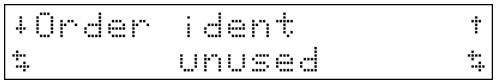
Number of the production line to be used (input, predefined)



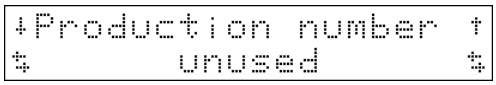
Customer ident (unused, input, predefined). An input or predefined parameter is included additionally in the batch report.



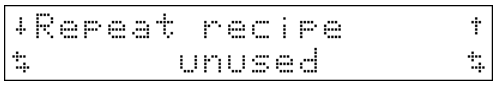
Order ident: Name or number (unused, input, predefined)
An input or predefined parameter is included additionally in the batch report.



Production number: Enter the production number (unused, input, predefined)
An input or predefined parameter is displayed additionally in the batch report.



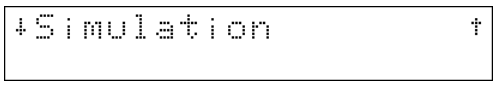
Repeat recipe: Number of productions of the same recipe (unused (1 times), input, predefined)



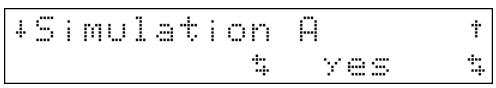
4.2.2.8 Simulation

This function is used only during commissioning, for automatic batch without load cells. With fine flow, the speed is 0.2 times as high as the coarse flow. With discharge, the speed is 5 times as high as the coarse flow. Simulation is used for phases B1 ... B6 and B8.

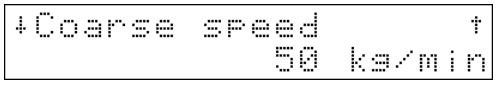
Select parameter by pressing OK.



Select either 'yes' or 'no'.



If 'yes' was selected, the coarse flow speed can be determined.



4.2.2.9 Default settings

Inputs

only Slot 1

Input number	Function
1	MX 130
2	MX 131
3	MX 132
4	MX 133
5	MX 134
6	MX 135

Outputs

only Slot 1

Output number	Function
1	MX 36 coarse flow
2	MX 37 fine flow
3	MX 38 discharge
4	
5	MX 140
6	MX 141
7	MX 142
8	MX 143

Analog output

Analog output	gross weight of WP-A
Analog range	4..20mA
Analog error 1 ADC	0=0mA
Analog error 2 <0	0=0mA
Analog error 3 >FSD	0=0mA

Parameter

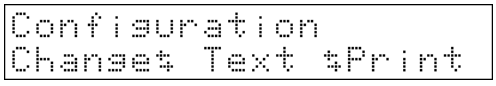
Parameter	Value
Limit values	0 in the format of the relevant scale
Start parameters	
Production line	predefined
Customer ident	unused
Order ident	unused
Production number	unused
Repeat recipe	unused
Simulation	
WP-A	No
WP-B	No

4.2.3 Entry of predefined texts

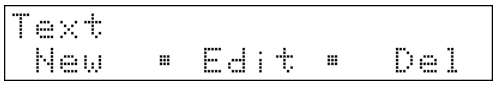
Manual and dialogue phase can work with predefined messages. In this case, the text field for 'Dialogue' contains the parameter 'dialog', 'dsp1' and 'dsp2'. The appearance and behaviour of the dialog is defined by a numeric code:
'dialog' = 01...09: the texts are set in the text field. dsp1='Text line 1' und dsp2='Text line 2'
'dialog' > 10: the texts will be taken from the database and need not be transmitted.

The texts are stored in a database. A text is identified by the text for line 1 and a unique number within 1 ... 999. For line 2, any text is possible (can also remain empty). The maximum length of the two texts is 20 characters. (See dialogue phase).

Press [Text] for editing of predefined texts.

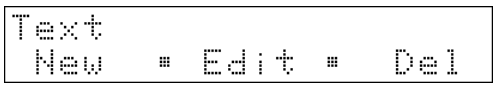


The database can be edited with [New], [Edit] and [Del].

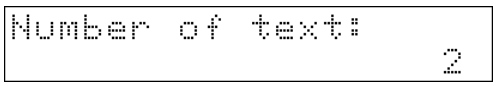


4.2.3.1 Entry of a new text

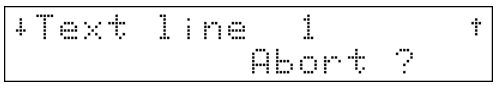
Press [New], see also the hints given below for Phase-X4.



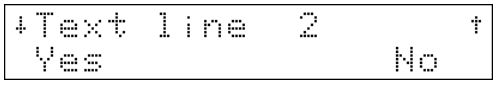
Enter a number within 1 and 999 and press OK. This number is used to address the text.



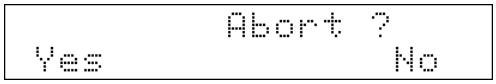
Enter the text for line 1 and press OK. This text is used also for identification of the database entry, i.e. it must not be empty.



Enter the text for line 2 and press OK. The text must correspond exactly to what it should be like subsequently. When using the text with a function which uses softkeys, the typical icons are inserted in the display (by replacing the character in the text). This text may remain empty and can contain the text of a dimension for numeric entries.

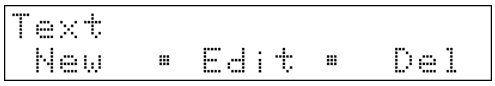


In the dialog, the two text entries would result in the following display:

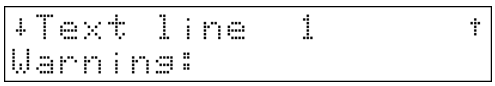


4.2.3.2 Editing a text

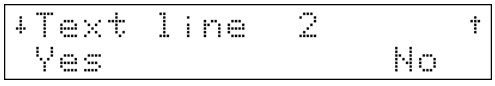
Press [Edit], for Phase-X4, see also hints in section 2.3.3.
Select by pressing key More.



Select text and press OK. With Phase-X4, press key More first, select the text and press OK.
This text is also content of the first line and can be edited. Text line editing is similar to entry. Press OK.



Edit the text of line 2 and press OK.

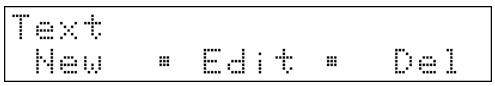


Press Exit to leave the edit mode.

After selecting [Edit], the text selected last is displayed. Scroll through the existing texts by means of scroll keys ↑ 4 GHI and ↓ 2 ABC, select with OK F2 and edit by means of key More as described above. Alternatively, an existing text number can be entered after pressing keys More and ← BACK. In this case, handling is done also as described above.

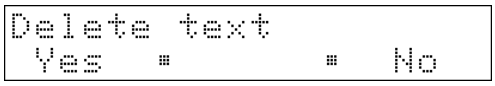
4.2.3.3 Deleting a text

Press [Del].



Select text and press OK.

Reply [Yes] to delete the text.



Press Exit to leave the delete mode.

4.2.4 Exit from the configuration

From the configuration main menu
Press key "Exit"
During parameter selection
Press key 

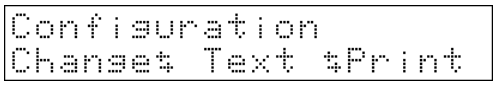
When parameters were changed, the following menu is displayed:

- Yes The changed parameters are stored in EAROM.
- No All changes are cancelled.
- Exit handling can be continued

4.2.5 Printing out the configuration data

When printing the first line, checking, if printing was possible is done. In case of printer failure during printing, there is a timeout of 2 sec for every print line. The print-out cannot be changed by means of 'Nice Lable Express' .

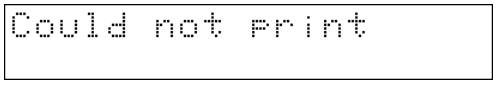
From the configuration main menu



Select [Print].

All data and predefined texts are printed out.

If printing is not possible, because no printer interface was selected in [Setup]-[Serial Ports], the following message is displayed during 2 s:



Print-out example:

```
Configuration data  PhaseController - Rel 1.00
Date: 2004.10.28 12:48
-------------------------------------------------

Input configuration
 Slot 1:            Digital inputs
  Input 1:          130
  Input 2:          131
  Input 3:          132
  Input 4:          133
  Input 5:          134
  Input 6:          135
 Slot 2:            No function
 Slot 3:            No function

Output configuration
 Slot 1:            Digital output
  Output 1:         36
  Output 2:         37
  Output 3:         38
  Output 4:         0
  Output 5:         140
  Output 6:         141
  Output 7:         142
  Output 8:         143
 Slot 2:            No function
 Slot 3:            Analog output
  Source of data:   WP-A
  Analog range:     4..20mA
  Analog value:     Gross
  ADU error:        0mA
  Below zero:       4mA
  Above FSD:        20mA

Limits
 WP-A: Limit 1 on   1.000 kg
 WP-A: Limit 1 off  0.900 kg
 WP-A: Limit 2 on   4.500 kg
 WP-A: Limit 2 off  4.600 kg
 WP-B: Limit 1 on   0.0 kg
 WP-B: Limit 1 off  0.0 kg
 WP-B: Limit 2 on   0.0 kg
 WP-B: Limit 2 off  0.0 kg

Parameter
 Production line    1
 Customer ident     unused
 Order ident        unused
 Production number  unused
 Repeat recipe      unused

Text #    Line 1               Line 2
-------------------------------------------------
     1    Warning:             Mixer is not running
     2    Setpoint:
    21    Take a sample                  OK
    47    Product code:
```

5 MAIN PROGRAM

'Start' activates the operator interface. For using phases **manual addition** and **dialogue**, the operator interface ('Start'-Programm) must have been activated. Production is also possible, when the operator interface is not active. The display remains dark, as long as no phase is running. For a local operation (e.g. phase stopping or tolerance alarm acknowledgement), the operator interface must have been started previously.

5.1 "Start" program

Select the main program from the initial status:

Press key [Start]:

```
PhaseController
Start  ▪Setup  ▪
```

As long as the display remains dark, no phase is running. In this state the status of an activ phase is visualised. A local operator may send a recipe start request to the host system (ProBatch+)

The 2 text lines are deleted.

Key Exit can be pressed at any time to call up a dialogue.

```
Start recipe
 Yes   ▪   No   ▪   End
```

The operator has to press [End]
Now the program is in the initial status again.

```
PhaseController
Start  ▪Setup  ▪
```

Press [No]
The program returns to the operator interface.

Press [Yes]
The start menu is displayed. See section 'Recipe start'.

If licence PR1713/32 S88 Phase Batching is missing, a warning message is displayed during two seconds.

```
No batching license
```

5.2 Recipe start

The PhaseController does not know recipes, but may send a recipe start request to the PC (ProBatch+). The recipe start is done on the PC.

Key Exit can always be pressed to call up a dialogue from the operator interface. This could also be done during a running phase visualisation.

```
Start recipe
 Yes   ▪   No   ▪   End
```

When replying 'Yes' , the recipe start dialogue is displayed. The dialogue is dependent on the configured recipe start parameters.
Before a recipe can be started, the name and several parameters must be stored in a database.

```
↓Shampoo                ↑
Start  ▪ New  ▪Delete
```

If the recipe name is still not stored in the database, it can be created by pressing 'New'.

```
Recipe
concrete
```

Subsequently, the unit for the setpoint must be entered.

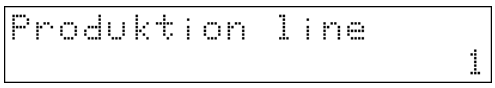
```
Weight dimension
                 % kg %
```

When pressing key 'Del' , the name of the recipe is removed from the database without further prompts.

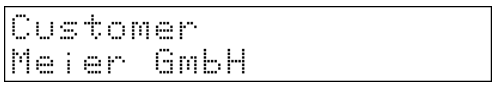
If the recipe exists in the datbase, it can be selected by means of the scroll keys and started by pressing key 'Start'.

Dependent on the start parameter configuration, a different number of parameters are prompted before starting.

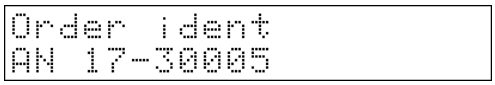
Enter the number of the production line to be used.



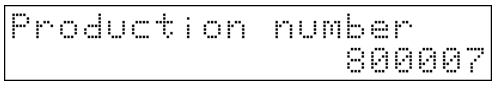
Name of the customer



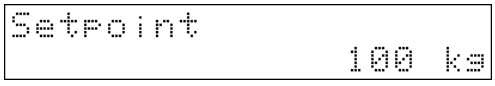
Order ident



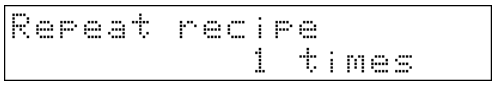
Production number



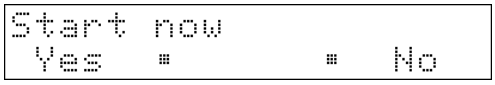
Setpoint



Number of cycles required for this recipe.



Now, the recipe can be started:



During recipe start, the PhaseController communicates with the PC. The communication status is displayed:

'Start ...' if the recipe start request will be send

'Started' if the PC has accepted the start request

'Could not start', if the recipe could not be started. In case of error, a message is sent to the PC alarm table.

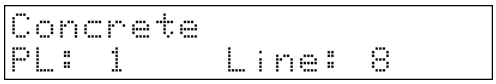
Starting a recipe takes some time.

5.3 Status display

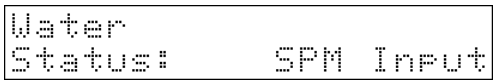
During an activ batch the status of the phase on the PhaseController and the status of the recipe on the PC is displayed.
Between the weighing point 'A' and 'B' the visualisation could be switched with the 'WP'-key.
The status display can indicate either the recipe status or the status of the current phase.

Press **'More'** for switch-over between the following two displays.

Status display for the **recipe**:
Line 1 contains the recipe name. Line 2 shows the production line and the recipe line number.
The status display for the recipe has only one display level.

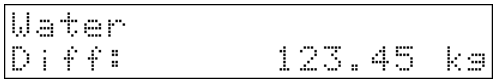


Status display for the current **phase**:
Line 1 contains the phase name (e.g. the material). Line 2 contains the current status, a value or an alarm.



If the status display is set to phase status, more information than fitting into a display level can be displayed. For this reason, a phase has at least two display levels. The various levels are accessible via  and  . The display is dependent on the current phase type.

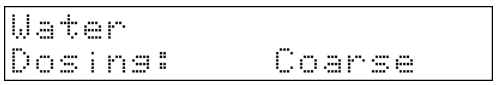
Status display for the **phases**:
'Diff:' indicates that further batching is required to reach the setpoint.



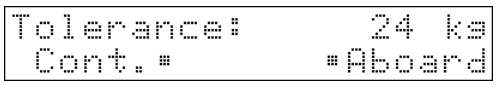
Status display for the **phases**:
'Setpoint:' is the material weight which is batched according to recipe by this phase.

Status display for the **phases**:
'±Tol:' indicates the permissible tolerance.

Status display for the **phases**:
'Dosing:', 'Status:' or 'Alarm:' informs you on the phase status: (see table below)



Status display for the **phases**:
Tolerance alarm with a manual component.



No phase is activ.

Phase status during display: 'Dosing:', 'Status:' or 'Alarm:'

Lozenge	Display	Status	Phase type
on	Status: SPM input	The phase is waiting for enable signal, coming from hardware. The input at address SPMin (configuration parameter) is used.	all with enable input
on	Dosing: Coarse	The phase is transporting material in coarse flow.	dosing
on	Dosing: Fein	The phase is transporting material in fine flow.	dosing
on	Differenz:	Display difference value	dosing
on	Setpoint:	Display setpoint value	dosing
on	Tolerance	Display tolerance values	dosing

on	Time:	Display the time of a timer component	dosing
on	Status: Discharge	The phase is discharging.	discharge
on	Status: Calming	The phase is waiting some time (configuration parameter).	dosing, discharge
blinking	Status: Stopped	The phase was stopped. Pressing 'Stop' allows to terminate or continue the phase.	all
on	Status: Done	The phase waits for reset after finishing.	all
on	Status: Wait	Wait-component activ	dosing
blinking	Alarm: Stopped	The phase was not started due to a wrong parameter in the text field.	dosing
blinking	Alarm: Aborted	Waiting for reset after aborting the phase.	all
blinking	Alarm: Tolerance	A tolerance alarm was detected. The phase is in held state. Pressing 'Stop' allows to terminate or continue the phase.	dosing
blinking	Alarm: Parameter	parameter of the phase is wrong	
blinking	Alarm: License	The batching license is missing	

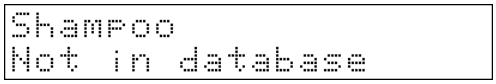
A flow warning is not displayed.

5.4 Stopping a phase or recipe

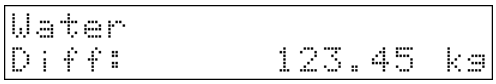
Press the red 'Stop' key to set a running phase into status HELD. A menu instead of the status is displayed.

Press keys → and ← to switch over between the menu for phases and recipe
In the phase menu, the stopped phase can be restarted or aborted.
In the recipe menu, the recipe can be stopped or aborted.

Only recipes from the local database are stopped or aborted. Unless the recipe can be found, a message is displayed.



Press key [Back] to return to the status display.
The current phase or recipe status remains unchanged.



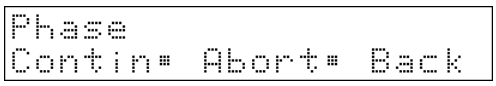
5.5 Restarting a phase

The phase is in status HELD.

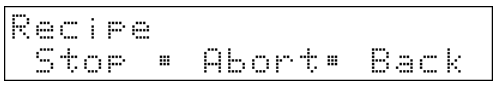
Press 'Contin' to continue the phase.

5.6 Aborting a phase or a recipe

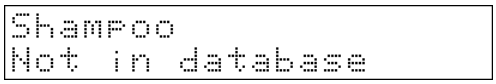
The phase is in status HELD.



Press  and  for switch-over between the menu for phases and recipe.



In the recipe menu, the recipe can be stopped.
'Abort' terminates the phase or recipe without further prompts.
Whilst the recipe is aborted, the PhaseController communicates with the PC. Aborting the recipe will lead to recipe stopping. During stopping, the communication status is displayed:
'Stopping ...'
and during aborting:
'Aborting ...'
'Aborted'
The execution of this commands is done on the PR8400 on the PC and may take some time during communication.
If this command could not be accepted by the PC, 'Could not stop' or 'Could not abort' is displayed.
Only recipes from the local database are stopped or aborted (recipes that have been started from this device). Unless the recipe can be found, a message is displayed.

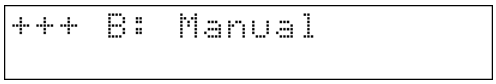


5.7 Manual additions

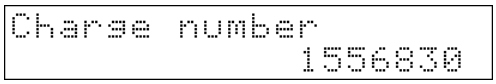
Manual batch modes (D1 or D2) can be acknowledged:
- by activating the bit addressed with SPMin or
- by means of key 'OK'.

In a configuration with two weighing points, manual batching on the non-displayed weighing point may be required.

When weighing point 'A' is selected for display and manual operation is to be done on weighing point 'B', the following message is displayed at regular intervals: For manual batching, the display must have been set to B by means of key 'WP' previously in this example.



Example of a dialog during a manual batching:



6 APPLICATIONS

6.1 PhaseController directly at the PC

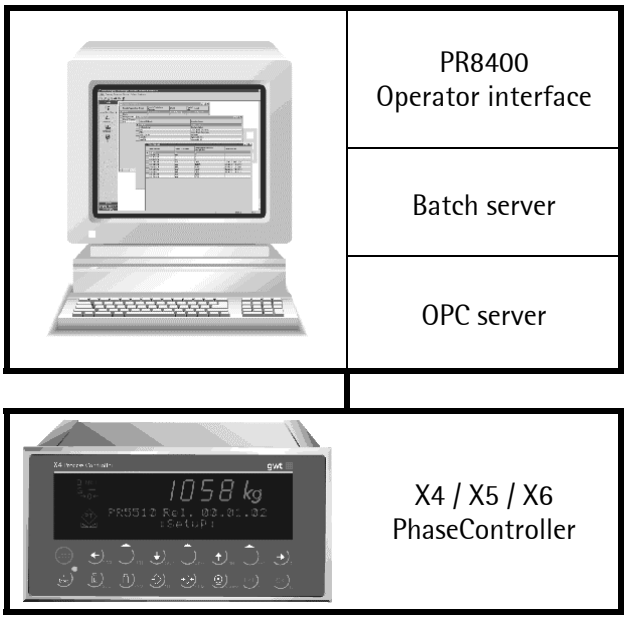
6.1.1 System description

In the simplest case,, the PhaseController is connected to the PC directly via a serial line or via Ethernet. Recipe management with the PR8400 ProBatch+ program runs on the PC.

The ProBatch+ is configured in open mode for the PhaseController. Recipe control is in the batch server.
The batching devices with PhaseController X4/X5/X6 are connected via serial line or via Ethernet.
The batch server can communicate with several batching devices via the PR1792 OPC server.

At batch start, the recipe remains in the PC. Only the data of the instantaneously active batch phase is downloaded into the batching device, and started.

The batching device handles the batch phase independently. ProBatch+ and the PhaseController visualize the status.



Please, note that the (coarse and fine) hopper valve(s) are connected with the batching device. There is no need to use an external PLC for further processing of these output signals.

SPM output and SPM input address must be specified in the component parameters of PR8400 ProBatch+. The addresses are grouped. The analog input and output components excepted, the SPM addresses are individual bits. Analog components A1 and A2 address words.
The following table describes the various **virtual SPM** groups. :

SPM address		Function
Word %MW	Bit %MX	
	0...47	System data, e.g. coarse flow or fine flow WP-A or WP-B, tared, etc. (for details, see virtual SPM)
3		Data for an analog output card
4...7		Data from an analog input card
	128...255	SPM addresses which can be used freely by the user for I/O functions with the digital hardware interface.
	256...511	Fieldbus interface. Not relevant in this application.
	512 ... 575	AND-combinations

6.1.2 Examples with local I/O

In this examples, all inputs and outputs are switched directly by the PhaseController. There is no PLC control intervention. The PLC does not use data of this weighing point, which could be made available by a possible proxy.

Example 1:
In the ProBatch+ program, the following components for WP-A were defined:
Component 'Flour' with batch mode B1 uses SPM output address: 128
Component 'Sugar' with batch mode B1 uses SPM output address: 129

When a recipe uses these components, the actual component data are transmitted to the PhaseController, and started.

In the PhaseController, the valves should be switched directly by the digital output card.
Output 1 is configured with address 36 and is active, when the coarse flow is activated by batching.
Output 2 is configured with address 37 and is active, when the fine flow is activated by batching.
Output 3 is configured with address 128 and is active, as long as component 'Flour' is active.
Output 4 is configured with address 129 and is active, as long as component 'Sugar' is active.

Example 2:
In the ProBatch+ program, the following component for WP-A was defined:
Component 'Controller' with batch mode A1 uses SPM Word output address: 3

In the PhaseController, the analog value should be output directly by the analog output card.
The analog output card is defined as 'external'. The value of component 'Controller' is output as an analog signal.

Example 3:
In the ProBatch+ program, the following component for WP-A was defined:
Manual component 'Manual' with batch mode D1 uses SPM input address: 31

In the PhaseController, the manual addition should be acknowledged with a digital input.
Input 1 is configured with address 131. This adress is TRUE when the input is activated. The component is also acknowledged by pressing the 'OK'-key.

Example 4:
In the ProBatch+ program, the following component for WP-A was defined:
Manual component 'Dialogue' with batch mode D4_DIALOG is used.
The additional parameter is set with: "dialog=11". '1x' is used to read the dialogue texts from the local PhaseController database (entry number 1). The parameter dsp1 and dsp2 are ignored.
'x1' should be used for local read-in of an integer value by the operator.

In the PhaseController, a dialogue should be held and an integer value should be read in. The dialog is enabled by means of a digital input.
With 'Setup' -> 'Config' -> 'Text', a database entry with number 1 containing the display texts should be done. When the component is active, the operator can enter a value and complete the input with 'OK'.

Block diagram
In the program ProBatch+ the components 'Flour', 'Sugar' and 'Controller' were defined as above:

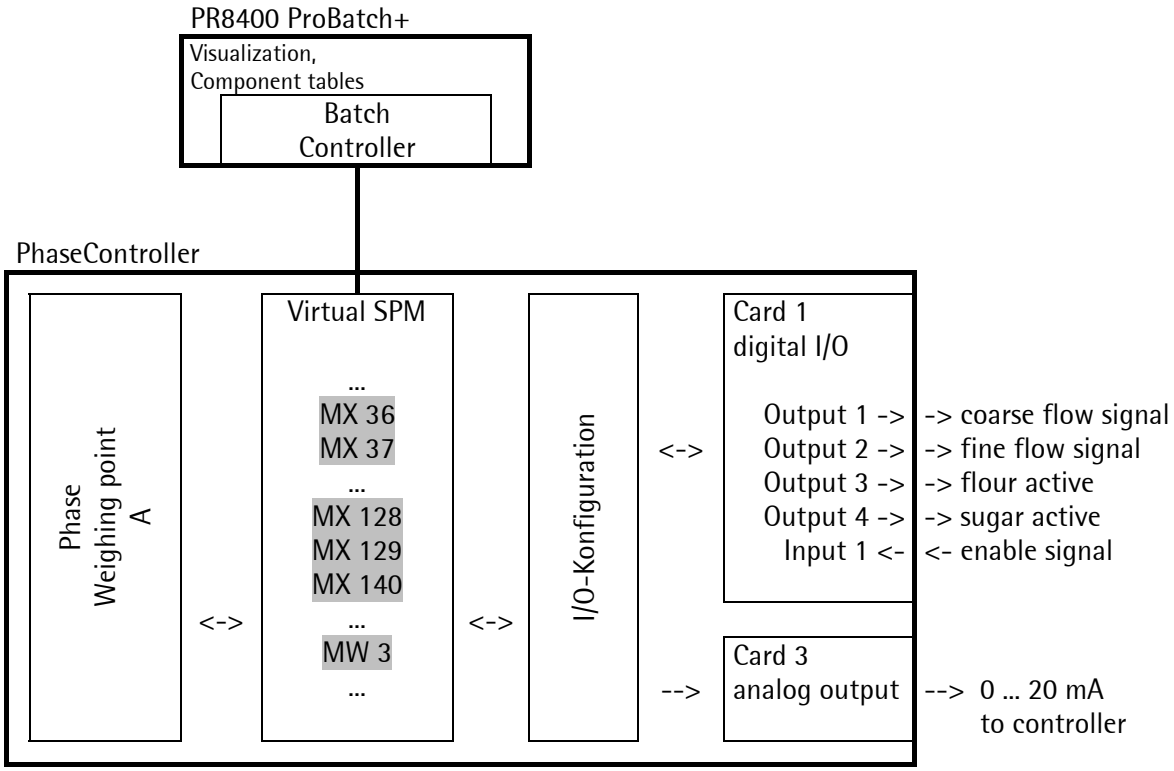
When a recipe in the PC (ProBatch+) uses a component, the data neccessary for batching of the activ component are transmitted to the PhaseController.

In the PhaseController, the valves should be switched directly by the digital output card. A mixer should be controlled with an aanalog output value. For this purpose, the inputs and outputs must be configured.

Input configuration:
Input 1: 140 (enable signal)

Output configuration:
Card 1:
Output 1: 36 (active, when the coarse flow is activated by batching)
Output 2: 37 (active, when the fine flow is activated by batching)
Output 3: 128 (active, as long as component 'flour' is active)
Output 4: 129 (active, as long as component 'sugar' is active)

Card 3:
The analog output card is configured as 'external. The value of component 'Controller' is output as an analog signal.



6.2 PhaseController connected to a PLC with proxy server

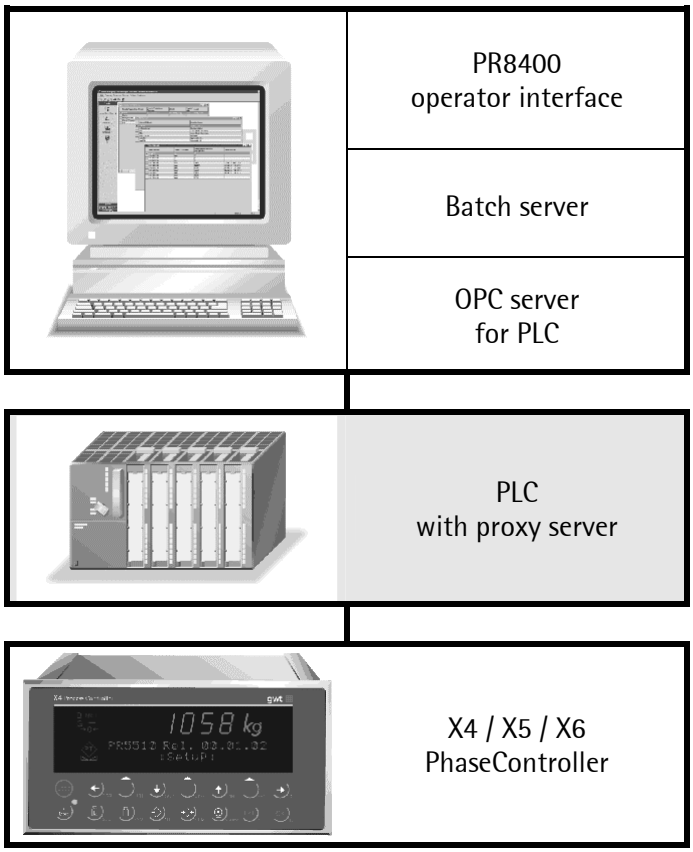
6.2.1 System description

For further processing of coarse and fine flow signals as well as component signals from the various weighing points by means of logic functions with process data from the production line, connecting of a PLC between batching controller and PC is purposeful.

The external PLC can be used for programming any logic data functions. For instance, a batch start request can be delayed, until all required material paths are enabled. Or the resources used in common, e.g. product paths, hoppers and valves, can be switched.
During PLC programming, the PLC logic can be coordinated with the ProBatch+ recipe procedure. This mode can be used also, when special production steps which are controlled exclusively by a PLC are required. Then a phase is running inside the PLC inself.

The proxy server for the PLC is a program, which only routes all data without hinderance. The batching device function can be checked. In practice, this program is extended by the actual PLC working program.

ProBatch+ is configured in open mode for the PhaseController. The recipe control runs in the batch server.



The batching devices with the X4/X5/X6 PhaseController are connected with a Siemens PLC e.g. via Profibus. The batch server can communicate via the OPC server to one or more PLC with several batching device each. For each PLC-type a specific OPC-Server is needed (e.g. Siemens).

When starting a batch, the recipe remains in the PC. Only the data of the instantaneously active batch phase is downloaded into the batching device via the proxy server in the PLC, and started.
Any logic data processing functions with these data can be programmed in the PLC.

The batching device handles the batching phase independently. ProBatch+ and PhaseController visualize the current status.

The SPM output and SPM input addresses must be specified in the PR8400 ProBatch+ component parameters. The addresses are grouped. The analog input and analog output component excepted, the SPM addresses are individual bits. Analog components A1 and A2 address words. The following table describes the various groups.

The following table describes the various **virtual SPM** groups:

SPM address		Function
Word %MW	Bit %MX	
	0 ... 47	System data, e.g. WP-A or WP-B coarse flow or fine flow, tared, etc. (for details, see virtual SPM)
3		Data for an analog output card
4...7		Data from an analog input card
	128 ... 255	SPM addresses to be used freely by the user for I/O functions with the digital hardware interface.
16 ... 19	256 ... 319	This data area (8 bytes for **weighing point A**) is read cyclically via the fieldbus and transmitted to the PLC by the PhaseController. The data area can be addressed by words or individual bits.
20 ... 23	320 ... 383	In this data area (8 bytes for **weighing point A**), data are written into the PhaseController cyclically by the PLC via the fieldbus. The data area can be addressed by words or individual bits.
24 ... 27	384 .. 447	This data area (8 bytes for **weighing point B**) is read cyclically via the fieldbus and transmitted to the PLC by the PhaseController. The data area can be addressed by words or by individual bits.
28 ... 31	448 ... 511	In this data area (8 bytes for **weighing point B**), data are written into the PhaseController cyclically by the PLC via the fieldbus. The data area can be addressed by words or by individual bits.
	512 ... 575	AND-combination of 128...159 and coarse/fine signal of WP-A/B

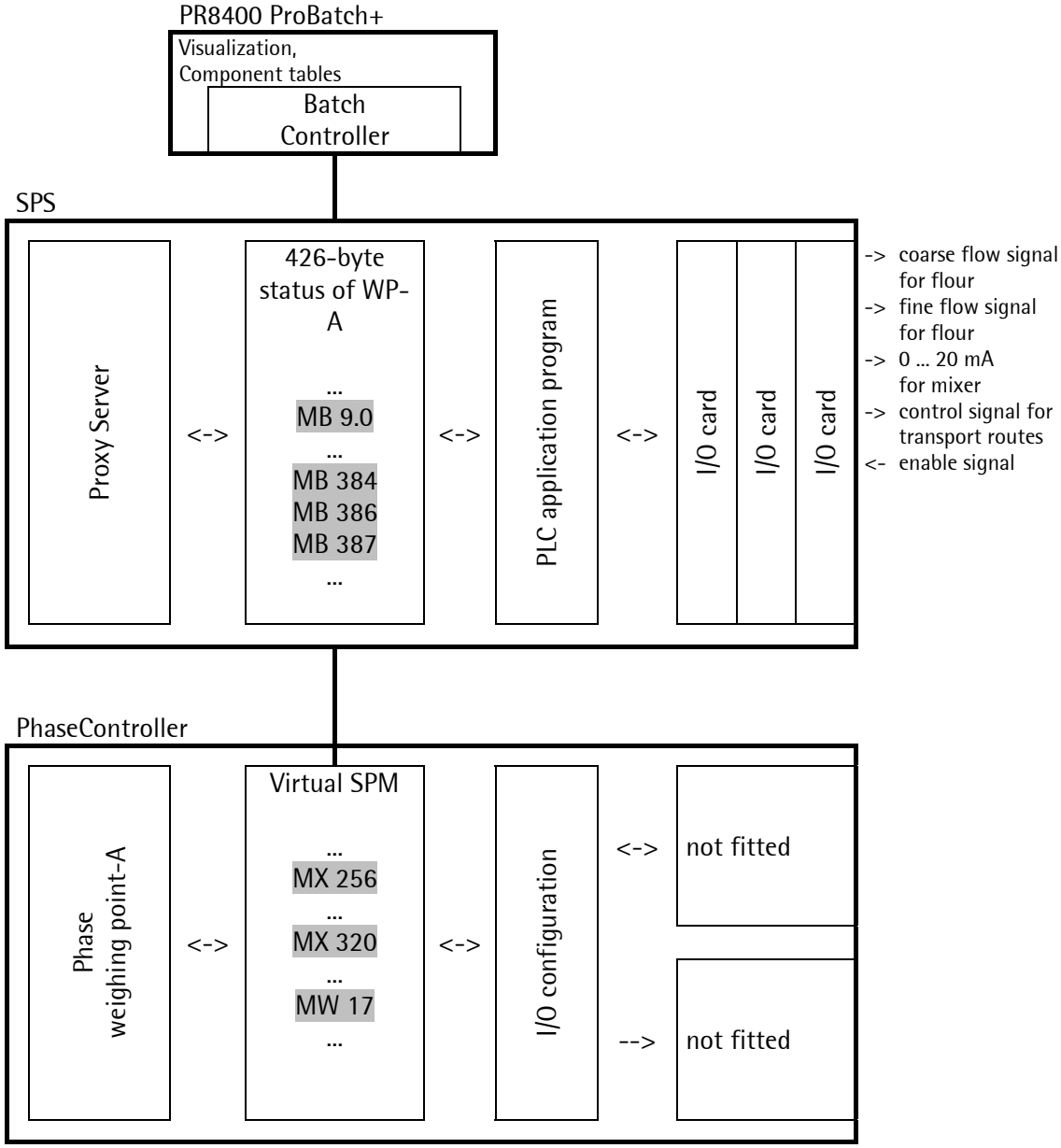
6.2.2 Example with I/O via PLC

In this example , PLC control intervention is demonstrated. Using the weighing point data made available by the proxy, the control signals are switched by the PLC.

Example 1:
In the ProBatch+ program, the following components for WP-A were defined:
component 'Flour' with batch mode B1 uses SPM output address: 256.
The SPM input address was set to 320. For starting the phase, the batch controller waits, until the bit on address 320 is set.

When a recipe uses this component, the data are transmitted to the PhaseController via the PLC.
When component 'Flour' is active, the bit on address 256 in the PhaseController is set. This data area (MX256 ... 319) is transmitted to the PLC cyclically by the PhaseController.
An application program in the PLC detects bit 256 256 (DB201.DBX384.7 in a Siemens PLC) for component 'Flour' and prepares the transport route for the material. During this time, the PhaseController waits for bit 320 (enable signal). When the transport route is running, the PLC sets bit 320 (DB201.DBX392.77 in a Siemens PLC). This data area (MX320 ... 383) is transmitted cyclically from the PLC to the PhaseController.
The PhaseController detects the set enable signal and starts batching. The phase status with coarse and fine flow batching is connected with the component signal (256) in the PLC aplication program and sent to a digital PLC output card. Valve control switching is done by the digital PLC output cards. After completing the PhaseController reset the component bit 256.

Example 2:
The PLC controls the r.p.m. of a mixer. The recipe contains a line with an A1 component (analog output).
SPM output is set to word address 17 in the PhaseController. This data area is transmitted cyclically from the PhaseController to the PLC.
The PLC reads the value of this address and adjusts the mixer r.p.m., which was defined in the recipe setpoint. This r.p.m. is sent to an analog PLC output card.



PhaseController	corresponds to	SPS (Siemens S7)
MX 256		DB201.DBX384.7
MX 257		DB201.DBX384.6
MW 17 (MX 272...287)		DB201.DBX386.7 ... DB201.DBX387.0
MX 320		DB201.DBX392.7

Both bit 256 and word address 17 are within MX 256 ... MX 319 (length 8 bytes) and transmitted from the PhaseController to the PLC cyclically. In the PLC, these data in the structure for WP-A are within MB 384 and MB 391.
Bit 320 is within MX 320 ... MX 383 (length 8 bytes) and is transmitted cyclically from the PLC to the PhaseController.

6.3 Virtual SPM MEMORY

To avoid conflicts between system variables and process interface, a 'virtual SPM' as part of the overall SPM was introduced. Thus the remaining SPM is protected against access by PC and PLC. This is an area of 64 bytes. The basic address is %MB 400 -> %MB 0.

All SPM addresses defined in components (SPMIn / SPMOut) with ProBatch+ (SPMIn / SPMOut) refer exclusively to the addresses of the virtual SPM. A phase is set to HELD automatically, when the address is beyond this range. Exception: -1 and 0 are ignored, the 'input' is always set TRUE.

Relative address	Type		Function
MX 0	BOOL		Reserved for the system
MX 1	BOOL	out	TRUE when the phase is running
MX 2	BOOL	in	The rising flank stops all phases
MX 8	BOOL	in	TRUE blocks 'Setup'
MX 9	BOOL	in	TRUE blocks closing of the operator interface
MX 10	BOOL	out	Limit value 1 WP-A
MX 11	BOOL	out	Limit value 2 WP-A
MX 12	BOOL	out	Limit value 1 WP-B
MX 13	BOOL	out	Limit value 2 WP-B
MX 32	BOOL	out	Data for WP-A are valid
MX 16	BOOL	out	Tolerance alarm WP-A
MX 17	BOOL	out	Flow-warning WP-A
MX 18	BOOL	out	Held WP-A
MX 24	BOOL	out	Tolerance alarm WP-B
MX 25	BOOL	out	Flow-warning WP-B
MX 26	BOOL	out	Held WP-B
MX 33	BOOL	out	¼ d WP-A
MX 34	BOOL	out	WP-A standstill
MX 35	BOOL	out	WP-A is tared
MX 36	BOOL	out	WP-A coarse flow
MX 37	BOOL	out	WP-A fine flow
MX 38	BOOL	out	WP-A discharge
MX 39	BOOL	out	WP-A is filled during simulation
MX 40	BOOL	out	Data for WP-B are valid

MX 41	BOOL	out	¼ d WP-B
MX 42	BOOL	out	WP-B standstill
MX 43	BOOL	out	WP-B is tared
MX 44	BOOL	out	WP-B coarse flow
MX 45	BOOL	out	WP-B fine flow
MX 46	BOOL	out	WP-B discharge
MX 47	BOOL	out	WP-B is filled during simulation
MW 3	WORD	out	Analog output, no function without interface
MW 4	WORD	in	Analog input 1, no function without interface
MW 5	WORD	in	Analog input 2, no function without interface
MW 6	WORD	in	Analog input 3, no function without interface
MW 7	WORD	in	Analog input 4, no function without interface
MX 128 ... 255		in and out	Addressed by SPMIn and SPMOut, defined in ProBatch+ and I/O configuration. For components and control signals for use with local I/O.
MX 256 ... 319		out	Adressed by SPMIn and SPMOut, is copied to the fieldbus (weighing point A, if any), defined in ProBatch+ and IO configuration. For components and control signals, which are switched on the PLC (e.g. Siemens S7).
MX 320 ... 383		in	Adressed by SPMIn and SPMOut, is copied by the fieldbus (weighing point A, if any), defined in ProBatch+ and IO configuration. Signal from the PLC (e.g. Siemens S7), which are required for the phase or local IO.
MX 384 ... 447		out	Addressed by SPMIn and SPMOut, is copied to the fieldbus (weighing point B, if any), defined in ProBatch+ and IO configuration. For components and control signals, which are switched on the PLC (e.g. Siemens S7).
MX 448 ... 511		in	Addressed by SPMIn and SPMOut, is copied by the fieldbus (weighing point B, if any), defined in ProBatch+ and IO configuration. Signals from the PLC (e.g. Siemens S7), which are required for the phase or local IO.
MX 512 ... 527		out	MX 512 ⇦ MX 128 **AND** Coarse WP-A ... MX 527 ⇦ MX 143 **AND** Grobstrom WP-A
MX 528 ... 543		out	MX 528 ⇦ MX 128 **AND** Feinstrom WP-A ... MX 543 ⇦ MX 143 **AND** Feinstrom WP-A
MX 544 ... 559		out	MX 544 ⇦ MX 144 **AND** Grobstrom WP-B ... MX 559 ⇦ MX 159 **AND** Grobstrom WP-B
MX 560 ... 575		out	MX 560 ⇦ MX 144 **AND** Feinstrom WP-B ... MX 575 ⇦ MX 159 **AND** Feinstrom WP-B

%ML	%MD	%MW	%MB	%MX 0 $80	1 $40	2 $20	3 $10	4 $08	5 $04	6 $02	7 $01	
0	0	0	0	0	1	2	3	4	5	6	7	System
			1	8	9	10	11	12	13	14	15	System, Limits
		1	2	16	17	18	19	20	21	22	23	Status WP-A
			3	24	25	26	27	28	29	30	31	Status WP-B
	1	2	4	32	33	34	35	36	37	38	39	Status WP-A
			5	40	41	42	43	44	45	46	47	Status WP-B
		3	6	48	49	50	51	52	53	54	55	*anaout*
			7	56	57	58	59	60	61	62	63	
1	2	4	8	64	65	66	67	68	69	70	71	*anain1*
			9	72	73	74	75	76	77	78	79	
		5	10	80	81	82	83	84	85	86	87	*anain2*
			11	88	89	90	91	92	93	94	95	
	3	6	12	96	97	98	99	100	101	102	103	*anain3*
			13	104	105	106	107	108	109	110	111	
		7	14	112	113	114	115	116	117	118	119	*anain4*
			15	120	121	122	123	124	125	126	127	
2	4	8	16	128	129	130	131	132	133	134	135	*SPMout* prepared
			17	136	137	138	139	140	141	142	143	for WP-A
		9	18	144	145	146	147	148	149	150	151	*SPMout* prepared
			19	152	153	154	155	156	157	158	159	for WP-B
	5	10	20	160	161	162	163	164	165	166	167	*SPMin / SPMout*
			21	168	169	170	171	172	173	174	175	
		11	22	176	177	178	179	180	181	182	183	
			23	184	185	186	187	188	189	190	191	
3	6	12	24	192	193	194	195	196	197	198	199	
			25	200	201	202	203	204	205	206	207	
		13	26	208	209	210	211	212	213	214	215	
			27	216	217	218	219	220	221	222	223	
	7	14	28	224	225	226	227	228	229	230	231	
			29	232	233	234	235	236	237	238	239	
		15	30	240	241	242	243	244	245	246	247	
			31	248	249	250	251	252	253	254	255	
4	8	16	32	256	257	258	259	260	261	262	263	SPMin / SPMout
			33	264	265	266	267	268	269	270	271	Fieldbus read
		17	34	272	273	274	275	276	277	278	279	WP-A
			35	280	281	282	283	284	285	286	287	
	9	18	36	288	289	290	291	292	293	294	295	
			37	296	297	298	299	300	301	302	303	
		19	38	304	305	306	307	308	309	310	311	
			39	312	313	314	315	316	317	318	319	
5	10	20	40	320	321	322	323	324	325	326	327	SPMin / SPMout
			41	328	329	330	331	332	333	334	335	Fieldbus write
		21	42	336	337	338	339	340	341	342	343	WP-A
			43	344	345	346	347	348	349	350	351	
	11	22	44	352	353	354	355	356	357	358	359	
			45	360	361	362	363	364	365	366	367	
		23	46	368	369	370	371	372	373	374	375	
			47	376	377	378	379	380	381	382	383	
6	12	24	48	384	385	386	387	388	389	390	391	SPMin / SPMout
			49	392	393	394	395	396	397	398	399	Fieldbus read
		25	50	400	401	402	403	404	405	406	407	WP-B
			51	408	409	410	411	412	413	414	415	
	13	26	52	416	417	418	419	420	421	422	423	
			53	424	425	426	427	428	429	430	431	
		27	54	432	433	434	435	436	437	438	439	
			55	440	441	442	443	444	445	446	447	
7	14	28	56	448	449	450	451	452	453	454	455	SPMin / SPMout
			57	456	457	458	459	460	461	462	463	Fieldbus write
		29	58	464	465	466	467	468	469	470	471	WP-B
			59	472	473	474	475	476	477	478	479	
	15	30	60	480	481	482	483	484	485	486	487	
			61	488	489	490	491	492	493	494	495	
		31	62	496	497	498	499	500	501	502	503	
			63	504	505	506	507	508	509	510	511	
8	16	32	64	512	513	514	515	516	517	518	519	Word 8 **AND** coarse
			65	520	521	522	523	524	525	526	527	of WP-A
		33	66	528	529	530	531	532	533	534	535	Word 8 **AND** fine
			67	536	537	538	539	540	541	542	543	of WP-A
	17	34	68	544	545	546	547	548	549	550	551	Word 9 **AND** coarse
			69	552	553	554	555	556	557	558	559	of WP-B
		35	70	560	561	562	563	564	565	566	567	Word 9 **AND** fine
			71	568	569	570	571	572	573	574	575	of WP-B

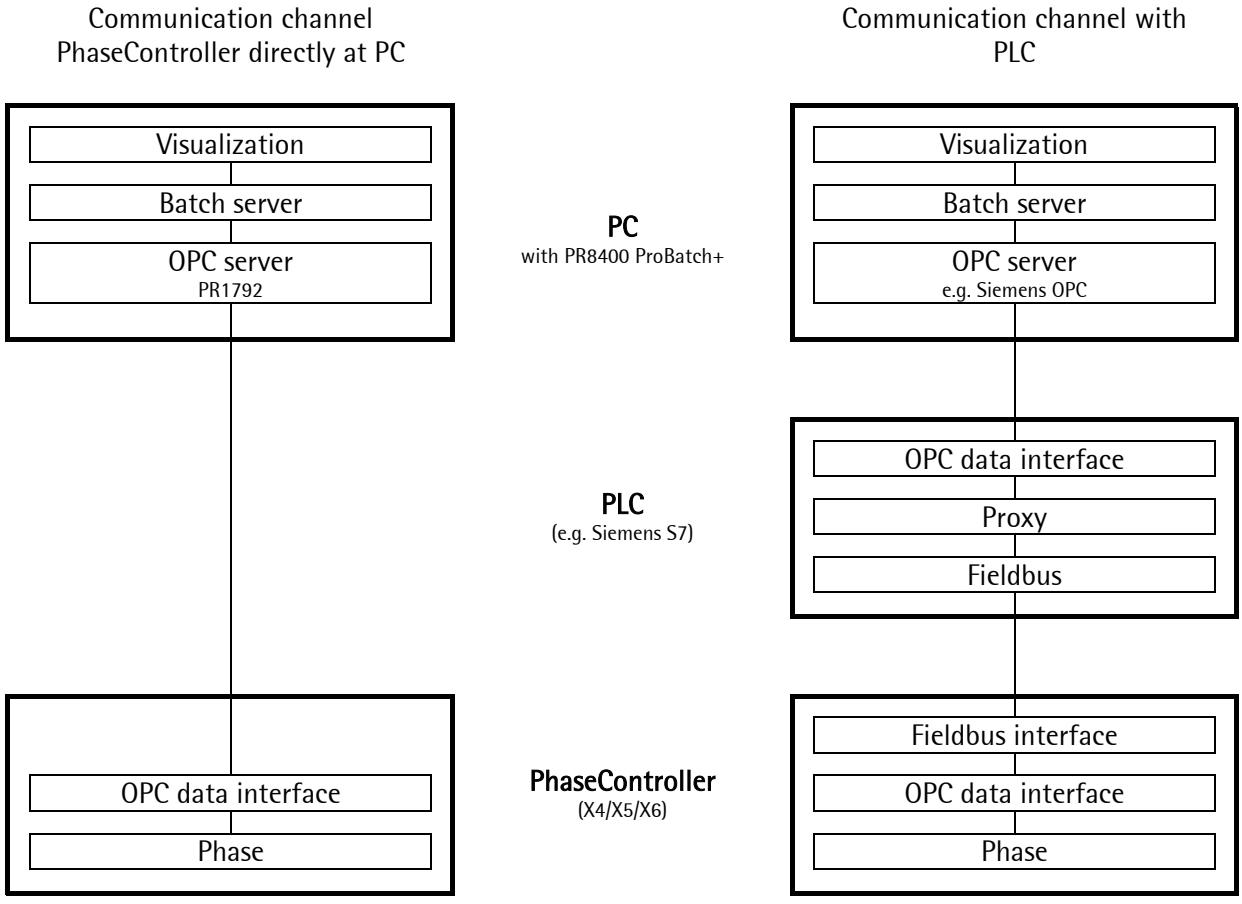
7 INTERFACE

This chapter contains detailed information for the installation and for development of a proxy. See also the manual 'Proxy Server'.

Each PhaseController is responsible for the batch process with one or two weighing points. The PhaseController interface structure is 320 bytes for each weighing point.

This structures are accessible for the PR1792 OPC server.
Frequently, the incoming and outgoing signals are also taken via a PLC. In this case, the PC communicates via the PLC and is not connected with the PhaseController. For connecting PC and PLC, a specific OPC server is installed for the PLC. OPC server, protocol and interface hardware are dependent on the PLC. For data transfer from the PLC to the PhaseController and vice versa, a proxy program is required. The proxy must transfer the data via fieldbus through a limited window.

The PLC can detect and modify all incoming and outgoing signals. The PLC application program connects the process data (e.g. enable, coarse, fine, discharge, material,...) with the PhaseController. Additional cables between PhaseController and PLC for process signals are not required.



7.1 OPC interface

The PC interface communicates with the PhaseController via a serial line or via Ethernet. When selecting the Ethernet, a fieldbus for the PhaseController is not available. PC access to the PhaseController by a PLC is either direct or via fieldbus. The interface data description is identical for PC, PLC and PhaseController.

OPC server
The T_OPC structure provides all required variables which are required for control of the PhaseController. The PR1792 OPC server is used for communication via a serial line or Ethernet and has direct access to the interface structure.

Modbus / TCP
A PR 5510/14 interface card is required for using the Modbus/TCP. This interface could be used with Ethernet from the PC or from the PLC. Writing is done in word address 1024, reading is from word address 0. None of the two addresses is in the PhaseController SPM! Instead, the PhaseController uses the multiplex IO area on socket 4. Although the Ethernet is used, configuration and use are similar to the fieldbus interface!

7.2 Fieldbus interface

The interface can be used with various fieldbus systems. The fieldbus interface requires 20 or 32 bytes for each weighing point. With an interface width of 20 bytes, only one weighing point is possible. The width can be configured in the X4, X5 and X6 'setup'. By means of the interface, the PLC can write and read the configured number of bytes. The size of a complete interface parameter set is 320 bytes. Therefore, the fieldbus window must be used in a multiplex mode. The fieldbus program in the PhaseController distributes the incoming and outgoing information from/to the T_OPC interface structure.

> **Important:**
> Fieldbus parameter 'Scale-interface' in the PhaseController must be set to 'disabled'. The data width has to be set to 20 or 32 bytes.

For further details related to the data exchange – see chapter 8.2.

Profibus:
A 32 byte interface is used for ProfiBus . Per weighing point, 32 bytes for the input and 32 bytes for the output are available. Optionally, a 20-byte interface can be used. However, using a 20-byte interface is much more time-consuming than the use of the 32-byte interface. For using two weighing points, the 32-byte interface is required.

Interbus-S:
For the Interbus-S , a 20-byte interface is used. Only one weighing point per instrument can be used (restriction due to hardware). 20 bytes for the input and 20 bytes for the output are used.

7.2.1 20-byte FB interface

Normally, this interface is used with Interbus-S installations. The PhaseController definition is used.

Messages and answers:
<Name>< [data format] >< comment>
The gray areas are the same for all telegrams.
The contens of the parameter "iReqCode" is in the example "iReqCode [USINT] **message 00"** equal to 00 or 00 + 1 = 01.

iReqCode	[USINT]	**message 00**
iCommand	[SINT]	phase status command

oAnsCode	[USINT]	**answer 00**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight
oActual	[REAL]	actual result
oWPError	[USINT]	WP error, like on display
oPhaseError	[USINT]	last error code
oMsgActive	[BYTE]	message transfer S7

iReqCode	[USINT]	**message 02**
iCommand	[SINT]	phase status command
iSPM	[LWORD]	part of virtual SPM

oAnsCode	[USINT]	**answer 02**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight
oSPM	[LWORD]	part of virtual SPM

iReqCode	[USINT]	**message 04**
iCommand	[SINT]	phase status command

oAnsCode	[USINT]	**answer 04**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight
oFSD	[REAL]	FSD weight
oUnit	[SINT]	WP dimension t, g, kg, ...
oExpo	[SINT]	WP decimals
oStep	[SINT]	WP step width8
oLicense	[BYTE]	WP license information

iReqCode	[USINT]	**message 06**
iCommand	[SINT]	phase status command
offset	[USINT]	offsetiniRecipeID
Filler	[BYTE]	due to S7 string format
iRecipeID	[STR15]	name of the recipe for visualization

oAnsCode	[USINT]	**answer 06**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight

iReqCode	[USINT]	**message 08**
iCommand	[SINT]	phase status command
iRecipeLine	[UINT]	line in the recipe for visualization
iSPMin	[INT]	SPM address of enable bit
iSPMout	[INT]	SPM address of 'component'
iRstMode	[UINT]	restart of dosing
iNegTol	[REAL]	negative tolerance limit
iPosTol	[REAL]	positive tolerance limit
iPLineID	[SINT]	production line ID

oAnsCode	[USINT]	**answer 08**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight

iReqCode	[USINT]	**message 10**
iCommand	[SINT]	phase status command
offset	[USINT]	offset in iMatID
Filler	[USINT]	due to S7 string format
iMatID	[STR15]	name of the material visualization

oAnsCode	[USINT]	**answer 10**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight

iReqCode	[USINT]	**message 12**
iCommand	[SINT]	phase status command
offset	[USINT]	offset in iBatchMode
Filler	[USINT]	due to S7 string format
iBatchMode	[STR15]	mode of dosing

oAnsCode	[USINT]	**answer 12**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight

iReqCode	[USINT]	**message 14**
iCommand	[SINT]	phase status command
iCalTime	[UINT]	calming time
iPreset	[REAL]	preset coarse to fine switching
iOvershoot	[REAL]	overshoot fine stream
iFlowRate	[REAL]	warning limit for flow
iSetpoint	[REAL]	target

oAnsCode	[USINT]	**answer 14**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight

iReqCode	[USINT]	**message 16**
iCommand	[SINT]	phase status command
iMinScale	[REAL]	scaling factor for analog I/O
iMaxScale	[REAL]	scaling factor for analog I/O

oAnsCode	[USINT]	**answer 16**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight

iReqCode	[USINT]	**message 18**
iCommand	[SINT]	phase status command
offset	[USINT]	offset in iTextpar
Filler	[USINT]	due to S7 string format
iTextLen	[USINT]	total length of iTextpar
iTextpar	[STR14]	extra optional text parameters

oAnsCode	[USINT]	**answer 18**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight

iReqCode	[USINT]	**message 20**
iCommand	[SINT]	phase status command

oAnsCode	[USINT]	**answer 20**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight
oSetpoint	[REAL]	
oActual	[REAL]	actual result / report
oOvershoot	[REAL]	new overshoot as report

iReqCode	[USINT]	**message 22**
iCommand	[SINT]	phase status command

oAnsCode	[USINT]	**answer 22**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight
oFlowRate	[REAL]	actual flow rate
oSig	[DINT]	scale signature
oPhaseError	[USINT]	last error code
oWPError	[USINT]	last error code
oTextStatus	[BYTE]	status of message text transfer

iReqCode	[USINT]	**message 24**
iCommand	[SINT]	phase status command
offset	[USINT]	offset in bMessage

oAnsCode	[USINT]	**answer 24**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight
Filler	[BYTE]	S/-string format
Length	[SINT]	total length of oTextPar
oTextLen	[USINT]	due to S7 string format
oTextPar	[STR8]	option report parameter

iReqCode	[USINT]	**message 26**
iCommand	[SINT]	phase status command
offset	[USINT]	offset in bMessage
iMsgStatus	[BOOL]	TRUE if last transmission of bMessage
Filler	[USINT]	due to S7 string format
iMessage	[STR14]	portion of bMessage

oAnsCode	[USINT]	**answer 26**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight

iReqCode	[USINT]	**message 28**
iCommand	[SINT]	phase status command
offset	[USINT]	offset in bMessage
iMsgStatus	[BOOL]	status of message text transfer

oAnsCode	[USINT]	**answer 28**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight
Filler	[USINT]	due to S7 string format
Length	[SINT]	total length of oMessage
oMsgLen	[USINT]	due to S7 string format
oMessage	[STR8]	portion of bMessage

7.2.2 32-byte FB interface

The PhaseController definition is used. See IO layout.
For serial Modbus communication: All incoming messages are stored in the same memory. Analogously, one memory is used for the output messages. The basic address is %MB 480 for the input (T_FBIN) and %MB 544 for the output (T_FBOUT).

Messages and answers:
<Name>< [data format] >< comment>
The gray areas are the same for all telegrams.
The contens of the parameter "iReqCode" is in the example "iReqCode [USINT] **message 00"** equal to 00 or 00 + 1 = 01.

iReqCode	[USINT]	**message 00**
iCommand	[SINT]	phase status command
iSPM	[LWORD]	part of virtual SPM

oAnsCode	[USINT]	**answer 00**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight
oActual	[REAL]	actual result
oWPError	[USINT]	WP error, like on display
oPhaseError	[USINT]	last error code
oSPM	[LWORD]	part of virtual SPM
oMsgActive	[BYTE]	message transfer S7

iReqCode	[USINT]	**message 02**
iCommand	[SINT]	phase status command

oAnsCode	[USINT]	**answer 02**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight
oFSD	[REAL]	FSD weight
oUnit	[SINT]	WP dimension t, g, kg, ...
oExpo	[SINT]	WP decimals
oStep	[SINT]	WP step width
oLicense	[BYTE]	WP license information

iReqCode	[USINT]	**message 04**
iCommand	[SINT]	phase status command
Filler	[BYTE]	due to S7 string format
iRecipeID	[STR20]	name of the recipe for visualization
iRecipeLine	[UINT]	line in the recipe for visualization
iSPMin	[INT]	SPM address of enable bit
iSPMout	[INT]	SPM address of 'component'
iRstMode	[UINT]	restart of dosing

oAnsCode	[USINT]	**answer 04**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight

iReqCode	[USINT]	**message 06**
iCommand	[SINT]	phase status command
Filler	[USINT]	due to S7 string format
iMatID	[STR20]	name of the material visualization
iSetpoint	[REAL]	target
iPLineID	[SINT]	production line ID

oAnsCode	[USINT]	**answer 06**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight

iReqCode	[USINT]	**message 08**
iCommand	[SINT]	phase status command
Filler	[USINT]	due to S7 string format
iBatchMode	[STR20]	mode of dosing
iNegTol	[REAL]	negative tolerance limit
iPosTol	[REAL]	positive tolerance limit

oAnsCode	[USINT]	**answer 08**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight

iReqCode	[USINT]	**message 10**
iCommand	[SINT]	phase status command
iCalTime	[UINT]	calming time
iPreset	[REAL]	preset coarse to fine switching
iOvershoot	[REAL]	overshoot fine stream
iFlowRate	[REAL]	warning limit for flow
iMinScale	[REAL]	scaling factor for analog I/O
iMaxScale	[REAL]	scaling factor for analog I/O

oAnsCode	[USINT]	**answer 10**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight

iReqCode	[USINT]	**message 12**
iCommand	[SINT]	phase status command
offset	[USINT]	offset in bTextPar
Filler	[USINT]	due to S7 string format
iTextLen	[USINT]	total length of bTextPar
iTextpar	[STR26]	portion of bTextPar

oAnsCode	[USINT]	**answer 12**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight

iReqCode	[USINT]	**message 14**
iCommand	[SINT]	phase status command

oAnsCode	[USINT]	**answer 14**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight
oSetpoint	[REAL]	
oActual	[REAL]	actual result / report
oOvershoot	[REAL]	new overshoot as report
oFlowRate	[REAL]	actual flow rate
oSig	[DINT]	scale signature
oPhaseError	[USINT]	last error code
oWPError	[USINT]	last error code
oTextStatus	[BYTE]	status of message text transfer

iReqCode	[USINT]	**message 16**
iCommand	[SINT]	phase status command
offset	[USINT]	offset in bTextPar

oAnsCode	[USINT]	**answer 16**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight
Filler	[BYTE]	S/-string format
Length	[SINT]	total length of bTextPar
oTextLen	[USINT]	due to S7 string format
oTextPar	[STR20]	option report parameter

iReqCode	[USINT]	**message 18**
iCommand	[SINT]	phase status command
offset	[USINT]	offset in bMessage
iMsgStatus	[BOOL]	TRUE if last transmission of bMessage
Filler	[BYTE]	due to S7 string format
iMessage	[STR26]	portion of bMessage

oAnsCode	[USINT]	**answer 18**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight

iReqCode	[USINT]	**message 20**
iCommand	[SINT]	phase status command
offset	[USINT]	offset in bMessage
iMsgStatus	[BOOL]	status of message text transfer

oAnsCode	[USINT]	**answer 20**
oStatus	[SINT]	actual phase status
oWPFlags	[BYTE]	WP status
oPhaseFlags	[BYTE]	phase status
oGross	[REAL]	gross weight
Filler	[BYTE]	due to S7 string format
Length	[SINT]	total length of bMessage
oMsgLen	[USINT]	due to S7 string format
oMessage	[STR20]	portion of bMessage

7.3 Serial interface

A serial interface cannot be used with a fieldbus. Unless a fieldbus is connected, the addresses starting with %MB480 can be used to control the PhaseController as with the fieldbus. For detailed information, see Fieldbus interface and Memory and IO layout.

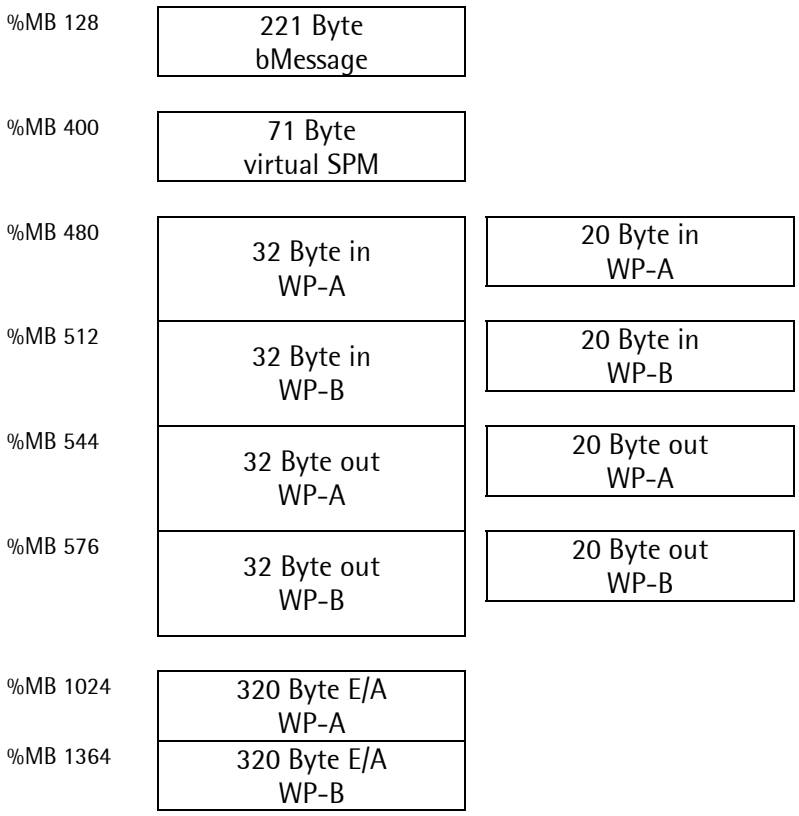
Modbus and DUST3964:
The ModBus has direct access to the specified SPM areas (starting with %MB 480). The fieldbus interface control logic is used. In addition to the interface provided for the fieldbus, the SPM output area, starting with %MB 432, can be used directly. %MB 432 is byte 32 (= bit 256) in the virtual SPM (see 'Virtual SPM').

7.4 Memory and I/O layout

Origin is structure T_OPC, as used by the OPC server. The contents and data types of this structure must be used fo all interfaces to the PhaseController. The format (structure or individual variable) is not decisive. In the PhaseController, the interface is on a fixed SPM address. For connection via fieldbus, a transfer program – the proxy – is required. The PhaseController is provided with a special fieldbus interface with a limited width of 20 or 32 bytes- see chapter I/O. Additionally, a copy is stored on a fixed SPM address.

7.4.1 Memory

The T_OPC basic address in the PhaseController is MB 1024. Each weighing point uses 320 bytes. As the variables are not sorted, every item must be transferred separately. The proxy uses the same interface definition. Basic address and type of addressing are dependent on PLC type. Alike, the data storage mode (as a structure, as a set of single local variables or as global variables) is dependent on the PLC characteristics and the relevant OPC server.
With %MB 400, a 64-byte 'virtual SPM' IO area for the process interface is used for data exchange with the process or the PLC. This area is addressed by ProBatch+ and by the PhaseController configuration.



%MD 89 (MB 356...359) is reserved for the counter of user-defined weighing points(for projects and test).

The fieldbus interface writes up to 15 different telegrams. The variable storage method is dependent on the PLC characteristics. For the PhaseController, it is as follows:
Every message is defined as a type T_FBIN and T_FBOUT structure (see fieldbus interface).
All incoming messages are stored in the same memory. The outgoing messages are also stored in one memory.
Input and output area form an array, which comprises the interface for the various weighing points.
The basic address is %MB 480 for the input (T_FBIN) and %MB 544 for the output (T_FBOUT).
For several ProBatch+ recipe management functions, remote control is possible. Remote control is via a memory part which is not part of this T_OPC interface structure. This buffer is available per device rather than per weighing point. Character string bMessage of up to 220 characters starts at MB 128 in the SPM. This buffer is controlled via T_OPC.

7.4.2 I/O

The 20 or 32 fieldbus interface bytes are copied from/into the SPM working area starting with MB480. Dependent on requested function code (iReqCode), the FB_PLC task in the PhaseController copies from and into the original interface T_OPC structure. The message layout consists of a fixed and a variable part. All incoming messages have a fixed header of 2 bytes:
1. request code in 'iReqCode' and
2. the command to the 'PhaseController' in 'iCommand'.

The content of the remaining bytes is dependent on iReqCode. All outgoing messages have a fixed 8-byte header:
1. answer code in 'oAnsCode',
2. phase status in 'oStatus',
3. weighing point status in 'oWPFlags',
4. status flags of the presently running phase in 'oPhaseFlags' and
5. the actual gross weight in 'oGross'.

The content of the remaining bytes is dependent on the answer code in 'oAnsCode'.
The PhaseController automatically uses the configured fieldbus interface card width. For every incoming message, a defined counterpart for the output exists.

See also chapter 'Fieldbus interface'.

8 PROXY (SELF PROGRAMMER)

8.1 Phase

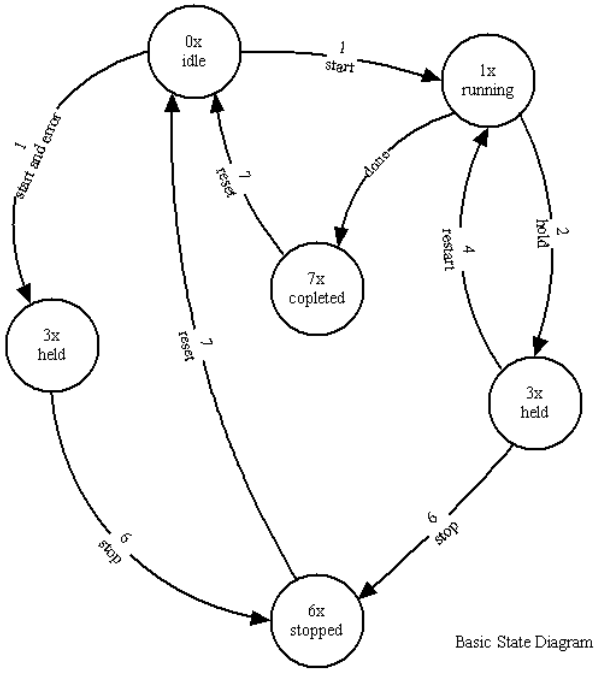
The proxy follows the basic state diagram of the phases (see chapter Phases). The proxy analyses the incoming commands and the current phase status:

1. Idle state: Before giving the START command to the PhaseController, all parameters required for a phase must have been sent to the PhaseController.

2. During operation, sending the phase status to the PC and the PC commands to the PhaseController is required.

3. Before sending states STOPPED and COMPLETED to the PC, all protocol data must have been sent to the PC.

4. Subsequently, the RESET command is given. Proxy and phase are idle again.



Basic State Diagram

The following table identifies the commands (in iReqCode):

	20-byte width	32-byte width
Initialization, after start:	4	2
In IDLE, before the START command is sent to the PhaseController:	6, 8, 10, 12, 14, 16, 18	4, 6, 8, 10, 12
When the phase is running or in idle state:	0, (2)	0
Before STOPPED or COMPLETED state signalling to the PC:	20, 22, 24	14, 16
Spontaneous messages:	26, 28	18, 20

(2) only when the PLC application uses the optional SPM transfer.

Each PLC requires an appropriate OPC server. As the characteristics of the various PLCs are different, an individual solution for every proxy must be found.

8.2 Data exchange

8.2.1 General

This chapter describes the data exchange between **PLC and PhaseController**. The 20 or 32-byte interface is used. For handling up to 15 different messages, a proxy is required for the PLC. The general data exchange is written in 'data'. Information for special handling of character strings is given in 'String'. The recipe management offers the possibility of remote control.

8.2.2 Data

The data exchange is controlled via the PLC fieldbus master. Control variables are iReqCode and oAnsCode. For multiple transmission, every message has 2 codes: n and n+1. As the data on the fieldbus are transmitted cyclically, code iReqCode must be changed, in order to be detected as a new request. The PLC must e.g. send n and n+1 alternately to handle the same function successively several times. When the PhaseController receives the same iReqCode, it will be ignored. The PhaseController reflects the iReqCode in oAnsCode. A detailed description of the method is given using an application example (for telegram content, see Fieldbus interface):

	Fieldbus / PLC	PhaseController
1	The PLC prepares a data set, which is sent to the PhaseController and changes iReqCode, if necessary.	
2	The PLC sends the data set to the PhaseController via the field bus.	The message is copied into the memory by the IO of the fieldbus interface card. The PhaseController detects a changed iReqCode.
3		Dependent on iReqCode, the content is transferred to the common interface structure T_OPC. Dependent on iReqCode, data are copied from the common interface T_OPC to the fieldbus interface.
4		iReqCode is copied into the oAnsCode. The message is copied into the fieldbus interface card IO.
5	The PLC detects the reflected answer in oAnsCode and handles the received data according to oAnsCode.	
6	Continue with step 1	

Example for data exchange (20-byte interface):

PLC	iReqCode	2		3		0		4		0		2		0		...
PhaseController	oAnsCode		2		3		0		4		0		2		0	

Messages which contain long characrter strings require special handling. Due to the limited number of bytes on the fieldbus, long character strings must be transmitted in several parts. In the case of the 20-byte interface, all character strings can be too long for a single message. With the 32-byte interface, the optional text parameter bTextPar and the optional message bMessage can be too long.

The PhaseController can handle a message in a PLC cycle at intervals of 50ms.

8.2.3 String

These messages must be divided into several parts, for transmission of long character strings:

	20 bytes wide	32 bytes wide
Telegram to PhaseController	6, 10, 12, 18	12
Telegram from PhaseController	24, 28	16, 20

Example:
The recipe name may be up to 20 characters long. With a 20-byte interface, only up to 15 characters of the recipe name in message number 6 can be transmitted. With a 32-byte wide interface, all 20 characters in message 4 can be transmitted.

8.2.3.1 String from PLC to PhaseController

The messages with potentially too long character strings contain variable 'offset' for decision, which part of the character string is sent.

	Fieldbus / PLC	PhaseController
1	The first message is sent with an offset = 0. The length of the transmitted character string is correct for the first part.	The PhaseController clears the corresponding character string and writes the received part into the variable.
2		The PhaseController copies iReqCode into oAnsCode and returns the defined answer telegram.
3	If the character string is already complete with this first message, transmission is completed, otherwise, the next part is sent (4).	
4	If transmission still isn't complete, the next part of the character string is sent. The offset is not decisive, but must be <> 0. The length of the transmitted character string is correct.	The PhaseController appends this part to the already received character string. The PhaseController returns the defined answer telegram.
5	If the character string is now complete, no further transmission will follow, otherwise, the next part is transmitted (4).	

As the character string is built up by appending, the format is always correct. In order to finish the transmission of a character string, a completion message is not required. The master simply stops the transmission, however, the parts must be sent in ascending order. Incoming character strings are not used before starting a phase. Consequently, the PLC must take care that the character string is complete, before giving the start command.

8.2.3.2 String from PhaseController to PLC

Output of character strings from the PhaseController is similar. Transmission is controlled by the master.

	Fieldbus / PLC	PhaseController
1	The first message is sent with offset = 0.	The PhaseController writes the overall length of the requested character string into the answer telegram (length). Additionally, the first part of the character string is copied.
2		The PhaseController copies iReqCode into oAnsCode and returns the specified answer telegram.
3	If the character string with this first message is already complete (actual length = length), no further transmission will follow, otherwise, the second part is recalled. (4).	
4	If the character string still isn't complete, the next part of the character string is requested and an offset > 0 is sent. The offset corresponds to the position of the next character which has to be sent.	The PhaseController writes the overall length of the requested character string into the answer telegram (length). The PhaseController writes a sub-string with the maximum possible number of characters – starting at the offset – into the answer telegram. If the remaining length is below the available space, the rest of the string is sent.
		The PhaseController copies the iReqCode into oAnsCode and returns the defined answer telegram.
5	The received character string is appended to the previously received character string. If the length of the stored character string corresponds with this length, no further transmission will follow, otherwise, the next part is requested. (4).	

As the character string is built up by appending, the format is correct at any time. No completion message is required to finish transmission of a character string. Outgoing character strings are used only, after the phase has reached the STOPPED status or the COMPLETED status, i.e. the PLC must only take care that the character string is complete, before sending the RESET command. RESET deletes the character string.

8.2.4 Remote control

Character string bMessage can be used for limited remote control of the PC (ProBatch+). As these messages are synchronous to all phases, handling is slightly different from the transmission of other character strings. In this case, the initiative is taken by the PhaseController.

The request code iReqCode in the following table are identical to those of the 32-bit model. The variables, e.g. oWPFlags, are part of the common PLC and PC IO (dependent on PLC characteristics, T_OPC is of equal or similar type). bMessage is a local buffer of up to 220 characters. The data block number in S7 installations is written into dMessage in T_OPC. This value is not used by the PhaseController.

Time nesting of these messages is not possible.

PhaseController	PLC	PC
Unless message transmission occurs: write message into local buffer bMessage, set oMsgStatus := TRUE, message available in oWPFlags := TRUE, iMsgStatus := FALSE. oMsgActive bit 2 := TRUE	PLC discovers request for message transmission by PhaseController in oWPFlags. This flag still isn't transmitted to the PC at this time. In OPC structure: iMsgStatus := FALSE.	
	Read character string with number of characters in cmd20. Length (iReqCode := 20 / 21, starting with cmd20.Offset := 0). After reading all characters in several cycles, if necessary, the signal that a message is available is transmitted to the PC in oWPFlags := TRUE .	PC discovers that a message is available in oWPFlags = TRUE, reads the local bMessage. The PC resets oMsgStatus.
		PC proceeds according to instruction in bMessage (e.g. start recipe) and writes the answer into bMessage. The PC sets iMsgStatus := TRUE.
	The PLC discovers iMsgStatus = TRUE and sends all characters from bMessage to the PhaseController (iReqCode := 18 / 19, starting with cmd18.Offset := 0). During the last substring transfer, cmd18.iMsgStatus becomes = TRUE.	
The PhaseController waits for cmd18.iMsgStatus = TRUE. bMessage is now valid and is handled, oMsgStatus := FALSE, the PhaseController sets signal 'message available' in oWPFlags := FALSE, oMsgActive bit 2 := FALSE		

The transfer time is monitored (timeout approx. 15 s).

8.3 Debug

With a terminal a debug-programm may be called, to debug status and variables.
Setup -> Serial Ports -> Operator device at
Chosse in the main menue 'L' for 'Comand Line'.
The debugger may be called with 'DBG'.

```
Version =       01.09.00                FSD =           10.000 kg
WP =            A                       Gross =         0.773

Command =          - 1  4 2 6 7 1  4  4  4  7 1   7 1  6 7 1
Status =        13 -  13 1 3 6 0 13 13 13 17 0 137 0 13 6 0 1
BatchMode =     D1                      RestartMode =   0
PLine ID =      1                       Material ID =   X5-A-D1
Recipe ID =     Manual                  Line =          1
Setpoint =      1                       Actual =        0.774
Preset =        0                       Overshoot =     0
Flowrate =      0                       Calmingtime =   0 s
PosTolerance =  0.05                    NegTolerance =  0.02
MinScale =      0                       MaxScale =      0
SPM in =        130                     SPM out =       140
WPError =       0                       WPFlags =       $01
PhaseError =    0                       PhaseFlags =    $20
License =       $00
TextPar =       OK:37.4 C
Message =       (0)
```

In the example above a phase is started by the Command '1:START'.
The phase is running: Status '1:RUNNING' and goes after a while into the state '3:HELD'. After that the phase is restarted with the command '4:RESTART'. The newest commands are displaied right in the line.
Below, all parameters of the actual phase are displaied: 'BatchMode', 'PLine ID', 'Recipe ID',...

9 PHASES

9.1 General

The information given in this section is intended primarily for project-based proxy development or PhaseController handling by the software project engineer. The normal user will find background information.

The phase specification is the result of long years of experience. This collection of control functions can be used to handle the process control functions of most batching applications. The phases were developed as the main part of process control. The phase behaviour follows the rules of S88.
The interface is designed for OPC server.

For general information, see Basic state diagram, and Phase types.

The common interface is described in Interface and Interface parameters.

A description of the various phase functions is given in Phase function blocks.

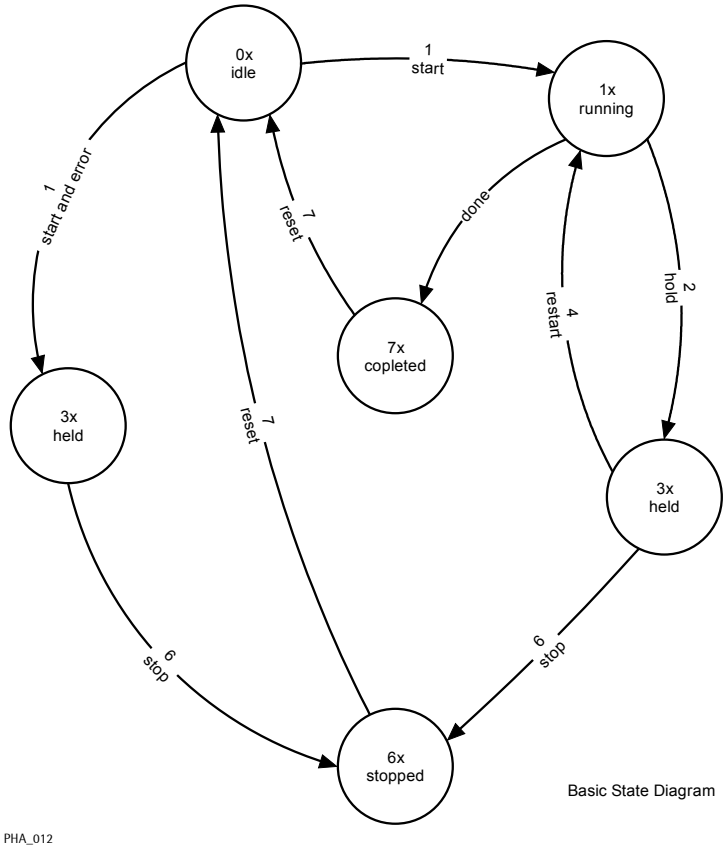
The PhaseController application program uses the FBLib-S88 library batching phases, which is available for building up similar applications. The application code can be used as an example for programming an operator interface for dialogues and visualization and for using the function blocks from the library. The application code is available on request and can be used as a basis for development of new applications.

9.2 Basic state diagram

9.2.1 Survey

Control information (iCommand) and status information (oStatus) are of utmost importance. Each phase follows the basic state diagram:

The arrows show the commands in iCommand, the circles show the phase states. As a state can have sub-states, the state in oStatus is multiplied by ten and the number of sub-state x is added. The result in lState is used for indication of a detailed state. Other state transitions than those which are shown in the diagram are not permissible.
A state in this diagram may have sub-states, which are used intern.



9.2.2 Commands

The commands for the state transitions:

commandl	Name	Description
11	TEST	Starts a phase for checking the licence validity/availability and the parameters. A process is still not started by this command.
1	START	This command starts the phase from the idle state. The command may be given only after all parameters were sent to the PhaseController.
2	HOLD	This command halts a running phase.
4	RESTART	This command restarts a phase, which is in HELD state.
6	STOP	This command is given, in order to stop a phase, which is in HELD state.
7	RESET	This command resets the phase to IDLE state. This command may be given only after recalling all phase parameters. The batch report is not available any more.

Commands 3 and 5 are not used by the phases in the library. After accepting a command, iCommand is reset to 0. Commands which are not listed above, and commands which are not permitted (not included in state diagram) are ignored.

9.2.3 States

The basic states are :

oStatus	Name	Description
0	IDLE	The phase is ready for starting, after all required data were written into the interface structure.
1	RUNNING	The phase is being handled presently.
3	HELD	The phase is not handled due to a HOLD command or due to internal problems (e.g. batching out of tolerance).
6	STOPPED	The phase was stopped due to a STOP command. This kind of termination is an exception. The batch report is available, until a RESET command is given.
7	FERTIG	The phase was finished regularly. The batch report is available, until a RESET command is given.

Other states are not used by the phase. Due to very fast handling, e.g. 50 ms per transition, a state may be not detected by the control program. A parameter error causes direct phase transition into the HELD state.

9.2.4 Order of phase handling

Phase handling order:

1. In IDLE state, all required parameters must be transferred to the interface. The number of parameters is dependent on the phase type, see Interface parameters and Phase function blocks.

2. Send the START command. The phase changes from IDLE into RUNNING (or into HELD if an error occurred).

3. If the phase is in HELD state, it can be re-activated with the RESTART command, or aborted using the STOP command.

4. When the phase reaches state COMPLETED (or STOPPED due to STOP command), read all phase-related return values. Important outputs are e.g. oActual, bOvershoot and sometimes bTextPar (for this, see also Interface parameters and Phase function blocks.).

5. Send RESET. The phase is reset to the IDLE state and waits for the next START command.

9.3 Phase types

The phases in the library (FBlib-S88) are implementations of the well-known firmware functions, now written in IEC 61131 with extended interface. The following batch modes are included:

Type	Phase name	Function
B1 ... B6	FDOSING	Filling or discharging in various mode
B8	FDISCHARGE	Discharges the scale completely
D1, D2	FMANUAL	Manual additions
D4_DIALOG	FDIALOG	Operator dialogue
A1, A2	FANALOG	Analog value reading or output
D4	FSTOP	Stops handling until a command is given
D5, D8	FWAITBIT	Waits for a specific bit
D6, D7	FSETRESET	Sets or resets a specific bit
D3	FTIMER	Waits a defined time

9.4 Phase interface

Structure T_OPC was developed to connect a complete process controller, but to a certain extent, also as a local interface for all phases.

The first character in a variable name is of special signification in this structure:

	Signification		Example
i	Phase inputs	These variables are only written by the control program (PC/SPS). There is one exception: iCommand is reset, after the command was detected to avoid handling it twice.	iCommand
o	Phase outputs	These variables are output by the phase. The values can change at any time.	oStatus
b	Bi-directional input / output	These variables are used in the two directions. For parameter setting before phase starting and for protocol sending after phase handling.	bOvershoot
l	Local, not transmitted	These variables are used for internal purposes, e.g. for communication between the PhaseController resources. The content can be read by a PC.	lStatus
d		Not used in the PhaseController.	dMessage

The interface structure is used by the following phases and special functions (e.g. visualization):

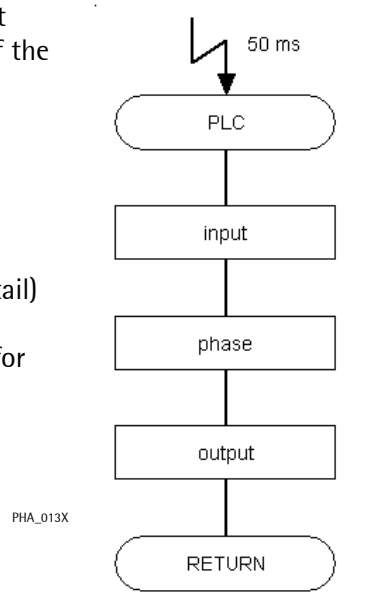
	FANALOG	FDIALOG	FDISCHARGE	FDOSING	FMANUAL	FSETRES	FSTOP	FTIMER	FWAITBIT
bOvershoot				R W					
bSetpoint	R			R	R			R	
bTextStatus		W		W	W				
bTextpar		R W		R	R W				
iBatchMode	R	R	R	R	R	R	R	R	R
iCalTime			R	R					
iCommand	R (W)	R (W)	R (W)	R (W)	R (W)	R (W)	R (W)	R (W)	R (W)
iFlowRate				R					
iMatID	Used for visualization								
iMaxScale	R								
iMinScale	R								
iMsgStatus	Used for visualization in ProBatch+								
iNegTol				R	R				
iPLineID	Used for visualization								
iPosTol				R	R				
iPreset				R					
iRecipeID	Used for visualization								
iRecipeLine	Used for visualization								
iRstMode				R					
iSPMin	R W								
iSPMout	R W								
iSPMinBit		R	R	R	R	R	R	R	R
oSPMoutBit		W	W	W	W			W	W
lCont		R W			R W				
lDirection				W					
lPowerFail	R W	R W	R W	R W	R W	R W	R W	R W	R W
lStatus	W	W	W	W	W	W	W	W	W
oActual	W	W	W	W	W	W	W	W	W
oExpo	Weighing point scaling for ProBatch+								
oFSD									
oGross	Used for visualization in ProBatch+								
oLicense	Weighing point scaling for ProBatch+								
oMsgActive	Used for PLC information on the message transfer								
oMsgStatus	Used for ProBatch+ remote control								
oPhaseError	W	W	W	W	W	W	W	W	W
oPhaseFlags	R W	R W	R W	R W	R W	R W	R W	R W	R W
oSig	Not used by the library								
oStatus	W	W	W	W	W	W	W	W	W
oStep	Weighing point scaling for ProBatch+								
oUnit									
oWPError									
oWPFlags									

R		Phase reads this item
W		Phase writes this item
(W)		Phase is set to 0, if accepted; not used by the application

9.4.1 Code

The code of all phases is called up in a cyclical PLC program. For best batching results, we recommend using each individual conversion of the scale ADC. Only one state transition per call is possible. None of the phases uses WAIT() and will return as quickly as possible, in order to handle the IO statements of the internal PLC.

Optionally, you can use the state in oStatus (or in lState in more detail) for display of the current phase status. Apart from FDIALOG and FANALOG with optional dialogue, no operator interface is required for handling a batch phase.



PHA_013X

9.5 Interface parameters (T_OPC)

In the following table, the 'master' is the control function in a system, the 'phase' is the executive instance, including all software. As structure T_OPC was developed for control of a complete batching device, some items are not used by the FBs in the library (FBlib-S88). The main program can use the complete interface. As data type WEIGHT is not used, the relevant REAL value has unit kg or lb – dependent on the actual configuration.

Convention for names: iXyz = data to PhaseController, oXyz = data from PhaseController, bXyz = for the two directions and lXyz =used locally in the PhaseController.

Variable	Type	Update	Use
bOvershoot	REAL	Bi-directional: by the master before phase start and updated by the phase in states STOPPED or COMPLETED.	Overshoot in kg or lb. Not changed whilst the phase is running.
bSetpoint	REAL	Bi-directional: by the master before phase start, or in HELD state and by the phase before changing to the STOPPED or COMPLETED state.	Setpoint in kg, lb, s or without unit – dependent on phase type: • Setpoint is sent to the phase before start. • read in STOPPED or COMPLETED state.
bTextStatus	BOOL	Bi-directional: by master or phase in STOPPED or COMPLETED state, when text parameters must be transmitted. After transmission by master or phase.	Used by the master, if text parameters for the phase are available. Used by the phase, when a parameter is returned in bTextPar. See also String.
bTextpar	STR160	Bi-directional: by master before phase start and by the phase before changing to STOPPED or COMPLETE state.	• send by the master before start, the format is 'name1=value2 name2=value2 ....' • not relevant whilst the phase is running • in STOPPED or COMPLETED state, if an optional protocol is available.
iBatchMode	STR20	By the master before phase start	Dependent on the phase type. 2 characters for standard phases. New phases must start with a standard batch mode (e.g. D4_DIALOG). iBatchMode is used twice: for further routing and selecting a function within the FB.
iCalTime	UINT	By the master before phase start.	Time in steps of 100 ms: • Calming time for batching. • still open time after completing a discharge.
iCommand	SINT	By the master, restart by the phase	Controls the phase - see basic state diagram. The application writes the command. iCommand is set to 0, when the phase has detected the command.
iFlowRate	REAL	By the master before phase start.	Min. flow rate in kg/min or lb/min. used for flow warnings.
iMatID	STR20	By the master before phase start.	Name of the actual material; only for visualization.
iMaxScale	REAL	By the master before phase start.	Value of iSetpoint for 0/4 mA on analog inputs and outputs.
iMinScale	REAL	By the master before phase start.	Value of iSetpoint for 20 mA on analog inputs and outputs.
dMessage	UINT	Not used by the PhaseController.	S7 data block number for bMessage. Information from S7 to PC.
iMsgStatus	BOOL	By the master, restart by the phase.	TRUE when an asynchronous message (answer) was transmitted by the master.
iNegTol	REAL	By the master before phase start.	Lower tolerance in kg or lb.
iPLineID	SINT	By the master before phase start.	Production line number; only for visualization.
iPosTol	REAL	By the master before phase start.	Upper tolerance in kg or lb.

iPreset	REAL	By the master before phase start.	Preset point, for switch-over from coarse to fine flow (in kg or lb)
iRecipeID	STR20	By the master before phase start.	Name of the currently running recipe; only for visualization.
iRecipeLine	UINT	By the master before phase start.	Number of the recipe line, only for visualization.
iRstMode	UINT	By the master before phase start.	Restart mode batching.
iSPMin	UINT	By the master before phase start.	Address im SPM. The application program copies the content from this address into iSPMinBit.
iSPMout	UINT	By the master before phase start.	Address in SPM. The application program copies oSPMoutBit into this address.
iSPMinBit	BOOL	By the application.	Activates the phase, set and reset by the application. oPhaseFlags contains of copy of this bit.
oSPMoutBit	BOOL	By the phase.	Signals: 'Phase running' (select component) or used for process control – dependent on the phase type. Set and reset by the phase. oPhaseFlags contains a copy of this bit.
lCont	BOOL	By the phase.	Used internally for initialization of manual phase restart.
lDirection	BOOL	By the phase.	Only for weight simulation, is TRUE when filling
lPowerFail	BOOL	By the application, restart by the phase.	Used internally, to stop the running phase after power recovery.
lStatus	BOOL	By the phase.	Detailed state. See state diagram of phase function blocks.
oActual	REAL	Actual value whilst the phase is running. Becomes the report in state STOPPED or COMPLETED.	Actual phase value. The unit (kg, lb, s, ...) is dependent on scale and phase type.
oBModeLen	USINT	Constant = 20	Max. length of character string iBatchMode
oExpo	SINT	By the application program during starting.	Digits behind the decimal point, exactly like WEIGHT_EXPO().
oFSD	REAL	By the application program during starting.	Full scale value in kg or lb.
oGross	REAL	By the phase.	Gross weight in kg or lb.
oLicense	BYTE	By the application program during starting.	See table below.
oMatLen	USINT	Constant = 20	Max. length of character string iMatID
oMsgActive	BYTE	By the phase.	Bit 0 is TRUE as long as message transfer is going on (only for fieldbus interface).
oMsgStatus	BOOL	By the phase, restart by the master.	TRUE when an asynchronous message (answer) must be transmitted.
oPhaseError	USINT	By the phase.	Updated whilst the phase is running. Codes with a higher number are not overwritten. 0: No error 1: Flow alarm, also coded in oPhaseFlags 2: Tolerance alarm, coded also in oPhaseFlags 3: Power failure set by PLC task. At

			power recovery, the phase must be reset. 4: Phase was aborted. 5: Faulty parameter, aborted. 6: Recipe stopped with red 'Stop' key. 7: An inadmissible state transition was requested 8: General weighing point error, also coded in oWPFlags: - 'Error X' on the display - Scale could not be tared - Switch-over to 'Single Conversion' was not possible - analog I/O value out of limits 9: Fatal error, e.g. no licence
oRecipeLen	USINT	Constant = 20	Max. length of character string iRecipeID
oPhaseFlags	BYTE	By the phase.	See table below
oSig	UDINT	By the application program during starting.	Signature from software and hardware parameters to ensure that the instrument was not replaced. The upper word contains a CRC16 on several hardware and software parameters. The lower word contains the same value + 100 * software version.
oStatus	SINT	By the phase.	Actual state - see basic state diagram.
oStep	SINT	By the application program during starting.	Display stepwidth, exactly like WEIGHT_VALUE(WGT_TYPE#STEP).
oTextLen	USINT	Constant = 160	Max. length of character string bTextPar
oUnit	SINT	By the application program during starting.	Weight unit, exactly like ENUM_TO_INT(WGT_UNIT).
oWPError	USINT	By the phase.	Exactly like 'Error x' on the display.
oWPFlags	BYTE	By the phase.	See table below.

Content of error bytes and status bytes:

IEC 61131	"C"	oLicense	oPhaseFlags	oWPFlags
7 ($01)	0	TRUE = batching PR1713/32	Discharge	Tared
6 ($02)	1		Fine flow	Scale error
5 ($04)	2		Coarse flow	Licence PR1713/32 missing
4 ($08)	3		Flow alarm	
3 ($10)	4		Tolerance alarm	
2 ($20)	5		Copy of oSPMOutBit	Content of oMsgStatus bit
1 ($40)	6		Copy of iSPMInBit	reserved
0 ($80)	7		TRUE whilst the phase is active	reserved

9.6 Offsets in T_OPC

					Base	1024	1344
Bit	Byte	Length	Type	Variable	Client	WP-A	WP-B
0	0	1	SINT	iCommand	MB	1024	1344
8	1	1	SINT	oStatus	MB	1025	1345
16	2		BOOL	iMsgStatus	MX	8208	1346
17	2		BOOL	oMsgStatus	MX	8209	10768
18	2		BOOL	bTextStatus	MX	8210	10769
19	2		BOOL	lPowerFail	MX	8211	10770
20	2		BOOL	lDirection	MX	8212	10771
21	2		BOOL	lCont	MX	8213	10772
22	2		BOOL	iSPMinBit	MX	8214	10773
23	2		BOOL	oSPMoutBit	MX	8215	10774
24	3	1	SINT	oUnit	MB	1027	1347
32	4	1	SINT	oExpo	MB	1028	1348
40	5	1	SINT	oStep	MB	1029	1349
48	6	1	USINT	oPhaseError	MB	1030	1350
56	7	1	USINT	oWPError	MB	1031	1351
64	8	1	BYTE	oWPFlags	MB	1032	1352
72	9	1	BYTE	oPhaseFlags	MB	1033	1353
80	10	2	UINT	iRecipeLine	MW	517	677
96	12	2	INT	iSPMin	MW	518	678
112	14	2	INT	iSPMout	MW	519	679
128	16	2	UINT	iRstMode	MW	520	680
144	18	2	UINT	iCalTime	MW	521	681
160	20	4	DINT	oSig	MD	261	341
192	24	4	REAL	oFSD	MR	262	342
224	28	4	REAL	oGross	MR	263	343
256	32	4	REAL	oActual	MR	264	344
288	36	4	REAL	bSetpoint	MR	265	345
320	40	4	REAL	iPreset	MR	266	346
352	44	4	REAL	bOvershoot	MR	267	347
384	48	4	REAL	iNegTol	MR	268	348
416	52	4	REAL	iPosTol	MR	269	349
448	56	4	REAL	iFlowRate	MR	270	350
480	60	4	REAL	iMinScale	MR	271	351
512	64	4	REAL	iMaxScale	MR	272	352
544	68	1	SINT	lStatus	MB	1092	1412
552	69	1	SINT	iPLineID	MB	1093	1413
560	70	1	USINT	oRecipeLen	MB	1094	1414
568	71	21	STR20	iRecipeID	MB	1095	1415
736	92	1	USINT	oMatLen	MB	1116	1436
744	93	21	STR20	iMatID	MB	1117	1437
912	114	1	USINT	oBModeLen	MB	1138	1458
920	115	21	STR20	iBatchMode	MB	1139	1459
1088	136	1	USINT	oTextLen	MB	1160	1480
1096	137	161	STR160	bTextpar	MB	1161	1481
2384	298	2	UINT	dMessage	MW	661	821
2400	300	1	BYTE	oLicense	MB	1324	1644
2408	301	1	BYTE	oMsgActive	MB	1325	1645
2416	302	18	STR17	lFiller	MB	1326	1646
	Länge	320	Bytes				

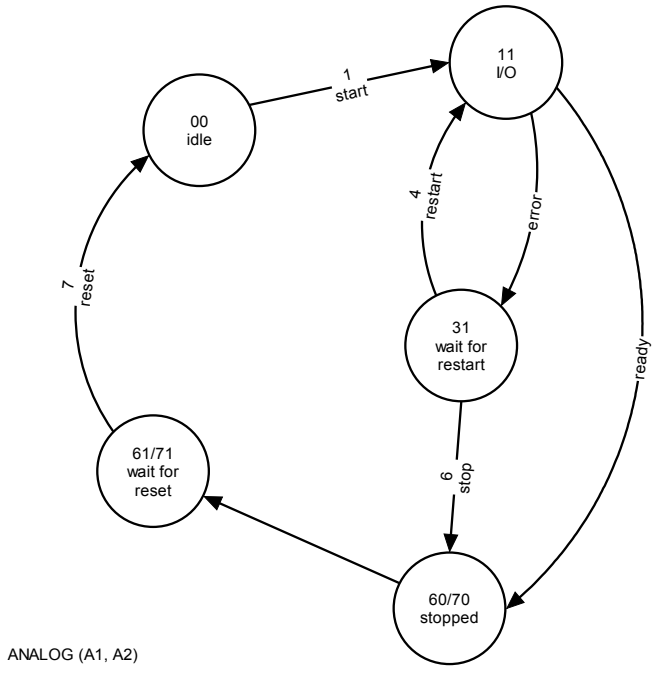
9.7 Phase function blocks

The description of the functionblocks is a description of the library FBlib-S88. This could be used for bachground information for the use of the parameter.

9.7.1 Analog on / off

The module is designed for input and output of analog values. Input and output values in the SPM are standardized to 0 ... 20000.

Phase	FANALOG	
Type	A1, A2	
Parameter	iBatchMode	Direction is defined by A1 (output) or A2 (input).
	iSPMin	Address for input. PLC program must transform the address by an array of type WORD for use as VAR_IN_OUT. See also Interface parameters.
	iSPMout	Address for output. PLC program must transform the address by an array of type WORD for use as VAR_IN_OUT. See also Interface parameters.
	bSetpoint	Output value for A1, not used for A2.
	oActual	Returns the standard value of the analog input – for A2, unused for A1.
	iMinScale	Setpoint for 0/4 mA
	iMaxScale	Setpoint for 20 mA
	SPMin	Not in T_OPC: direct address for word in SPM, internal PLC copies the analog input in scaled form into this address.
	SPMout	Not in T_OPC: direct address for word in SPM, internal PLC copies the scaled content of this address into the analog output.
Extended:	The phase goes to HELD automatically, when the input value is out of the permissible limits -2 mA and 22 mA.	



ANALOG (A1, A2)

Pha_001

Standardization for input and output 0 mA = 0 counts and 20 mA = 20000 counts.

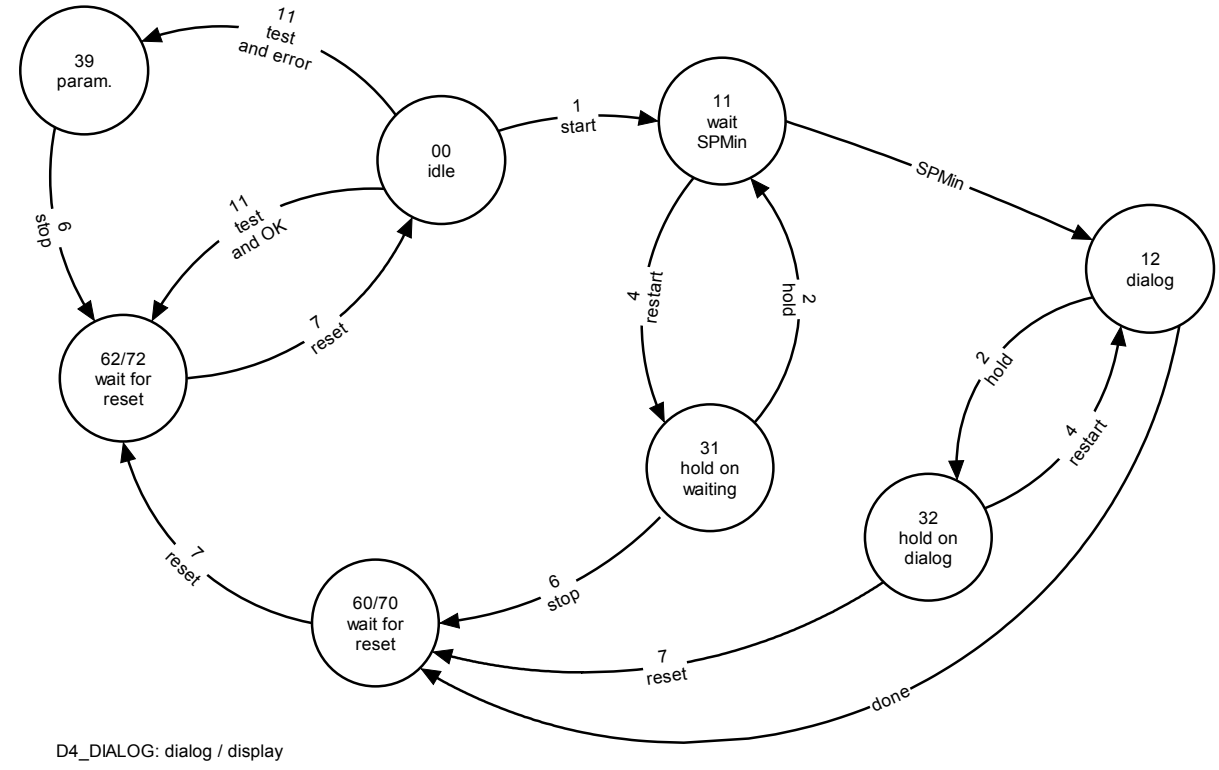
9.7.2 Dialogue

The dialogue phase provides only the state machine to program a dialogue for the operator interface. The dialogue itself must be programmed in the operating program, because access to DEVICE in resource operating is required.

Phase	F DIALOG	
Type	D4_DIALOG	
Parameter	WP	Weighing point identification ('A', 'B', ...)
	cmd	Copy of iCommand
	iBatchMode	Phase type is D4_DIALOG.
	iSPMin	Address to activate the dialogue, for more details, see interface parameters.
	iSPMout	Address: phase running
	bTextPara	Parameter : dialogue=(type of dialogue), optional dsp1 and dsp2 (see further down in the table)

The dialogue entry by the user is returned in bTextPar. The returned value is dependent on the dialogue type (see further down in the table).

For using this material in ProBatch+, file PHASECNT.DAT must be copied into the database directory of the PC.




```
FUNCTION_BLOCK D4_DIALOG
```

Parameters:
The parameters are written into the text field of this phase in ProBatch+. The three parameters are:

1.	dialog=xy	x > 0: prompt text from database, y: dialogue type
2.	dsp1='abc'	Optional: 'abc' is the message in the first display line
3.	dsp2='cde'	Optional: 'cde' is the unit for numeric entries in line 2

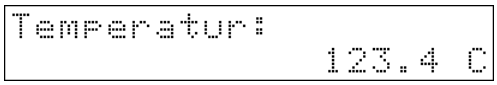
The phase checks parameter 'dialog' (MOD 10 >= 1 AND MOD 10 <= 9 = TRUE). x can be any number and is used to mark a special dialogue from the text database. y is the dialogue type. The text for the unit dsp2 should not be longer than 16 characters. '---' is returned when aborting the phase..

Dialog	Type	Function	Return
x1	DINT	Enter integer	'OK:<integer>' or '---'
x2	REAL	Enter decimal number	'OK:<real>' or '---'
x3	WEIGHT	Enter weight in the format of the actual unit	OK:<weight>' or '---'
x4	STR20	Text max. 20 characters	'OK:<text>' or '---'
x5		Softkey dialogue: Ok	'OK'
x6		Softkey dialogue: Ok + abort	'OK' or '---'
x7		Softkey dialogue: YES + No	'Yes', 'NO' or '---'
x8		Softkey dialogue: YES + abort + No	'Yes', 'NO' or '---'
x9		Softkey dialogue: softkeys of dsp2	'KEY1', 'KEY2' or 'KEY3' or '---'

This example shows how the temperature from the keyboard is read with a fixed input prompt and unit. Dialogue phase text parameters:

```
dsp1='temperature:'
dsp2='C'
dialog=2
```

The dialogue is as follows:

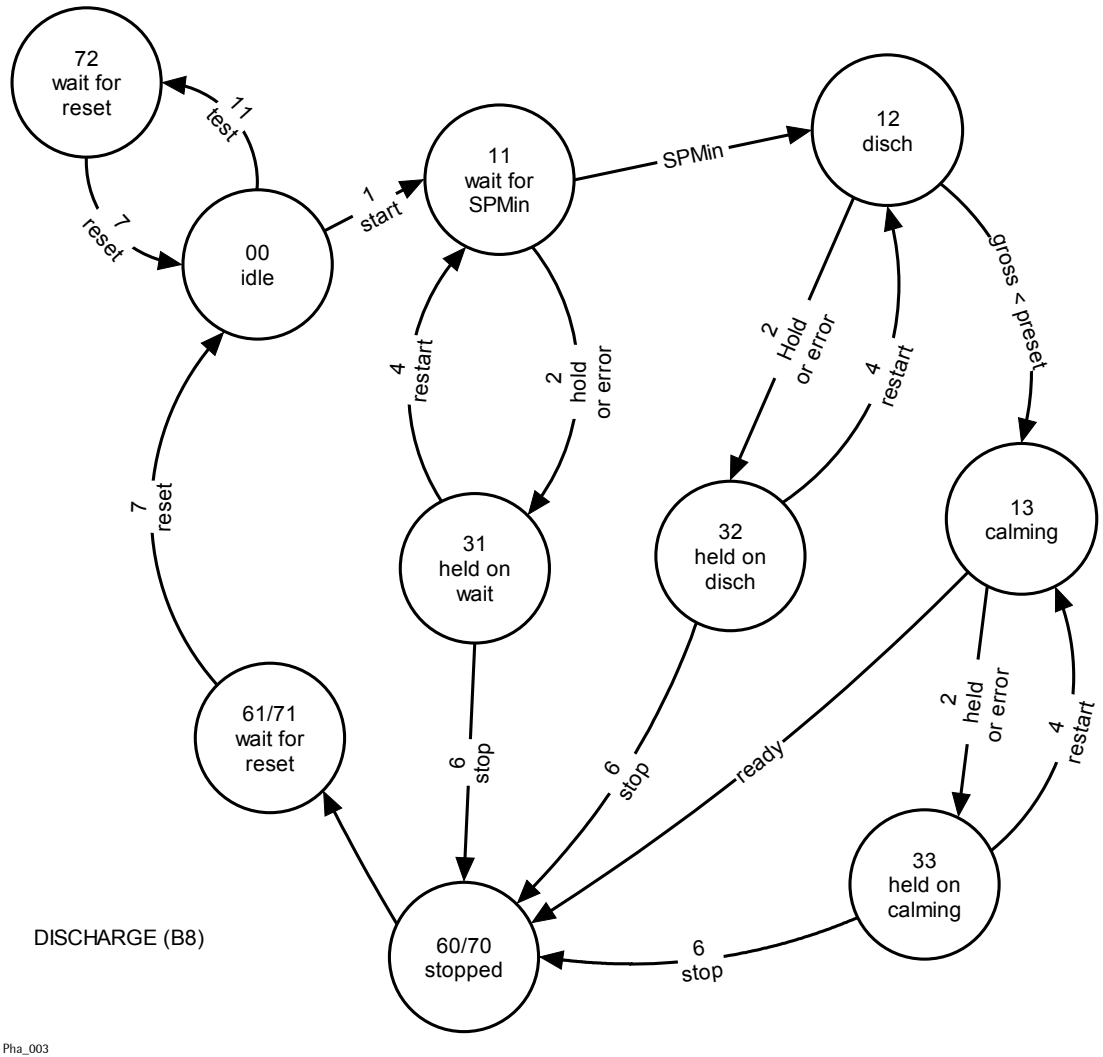


The operator has typed in '123.4' . The return value in bTextPar is 'OK:123.4 C'. It is shown in the report of ProBatch+.

9.7.3 Discharge

This block is used for complete discharge of the scale.

Phase	F DISCHARGE	
Parameter	WP	Weighing point identification ('A', 'B', ...)
	cmd	Copy of iCommand
	iSPMin	Address: activate discharge, for detailed information, see Interface parametersr
	iSPMout	Address: discharge is active (discharge signal in oWPFlags)
	iCalmTime	Waiting time after reaching iPreset
	bSetpoint	Ignored; target is always 0.0 kg or lb
	iPreset	Stops discharging, when gross weight < iPreset, then wait for discharging as much as possible
	oActual	Actual gross weight

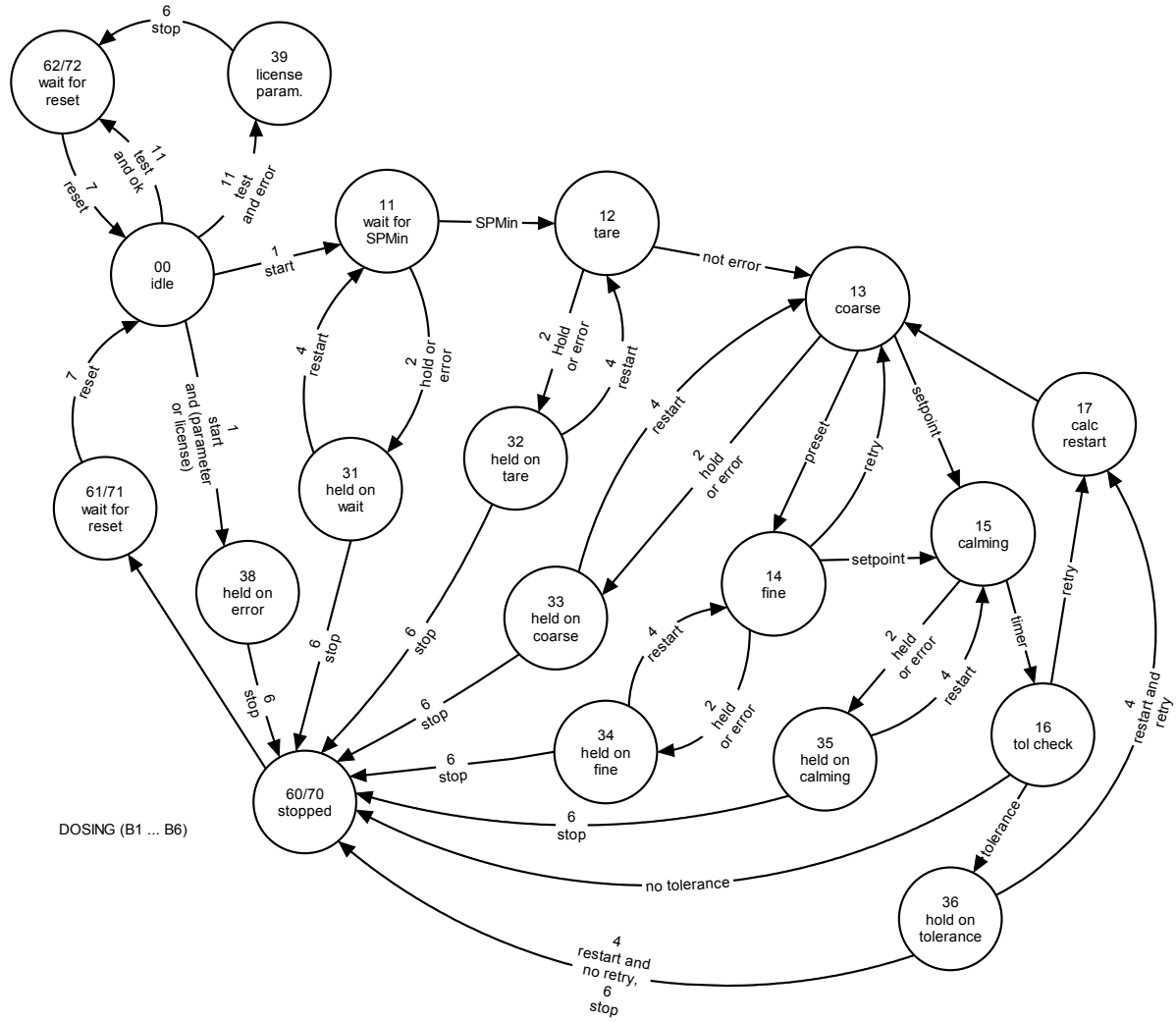


9.7.4 Batching

This block contains 6 batch modes. The behaviour is controlled by several parameters.

Phase	FDOSING	
Parameter	WP	Weighing point identification ('A', 'B', ...)
	cmd	Copy of iCommand
	iBatchMode	Batch mode (B1 ... B6), for more details, see Interface parameters
	iRstMode	Restart mode (0 ... 4), see flow diagram in manual of ProBatch+
	iSPMin	Address for activating, for further details, see Interface parameters
	iSPMout	Address: phase running = material selection, coarse and fine then in oWPFlags
	iCalTime	Calming time in steps of 100 ms
	iPreset	Preset point coarse/fine in kg or lb
	bOvershoot	Overshoot in kg or lb
	iFlowrate	Limit for flowrate monitoring (0 = no monitoring) in kg/min or lb/min.
	iSetpoint	Setpoint (gross or net weight) in kg or lb
	oActual	Actual net weight, then batch report in kg or lb
	iNegTol	Lower tolerance in kg or lb
	iPosTol	Opper tolerance in kg or lb
	iTextPara	Optional parameter: • Flowdelay = switch-on delay of flowrate monitoring default: 4s; • Adjust = fine / coarse ratio default: 0.125 • D = density default: not used =1

For using FDOSING with B1 ... B6, licence PR1713/32 is required.



Pha_004

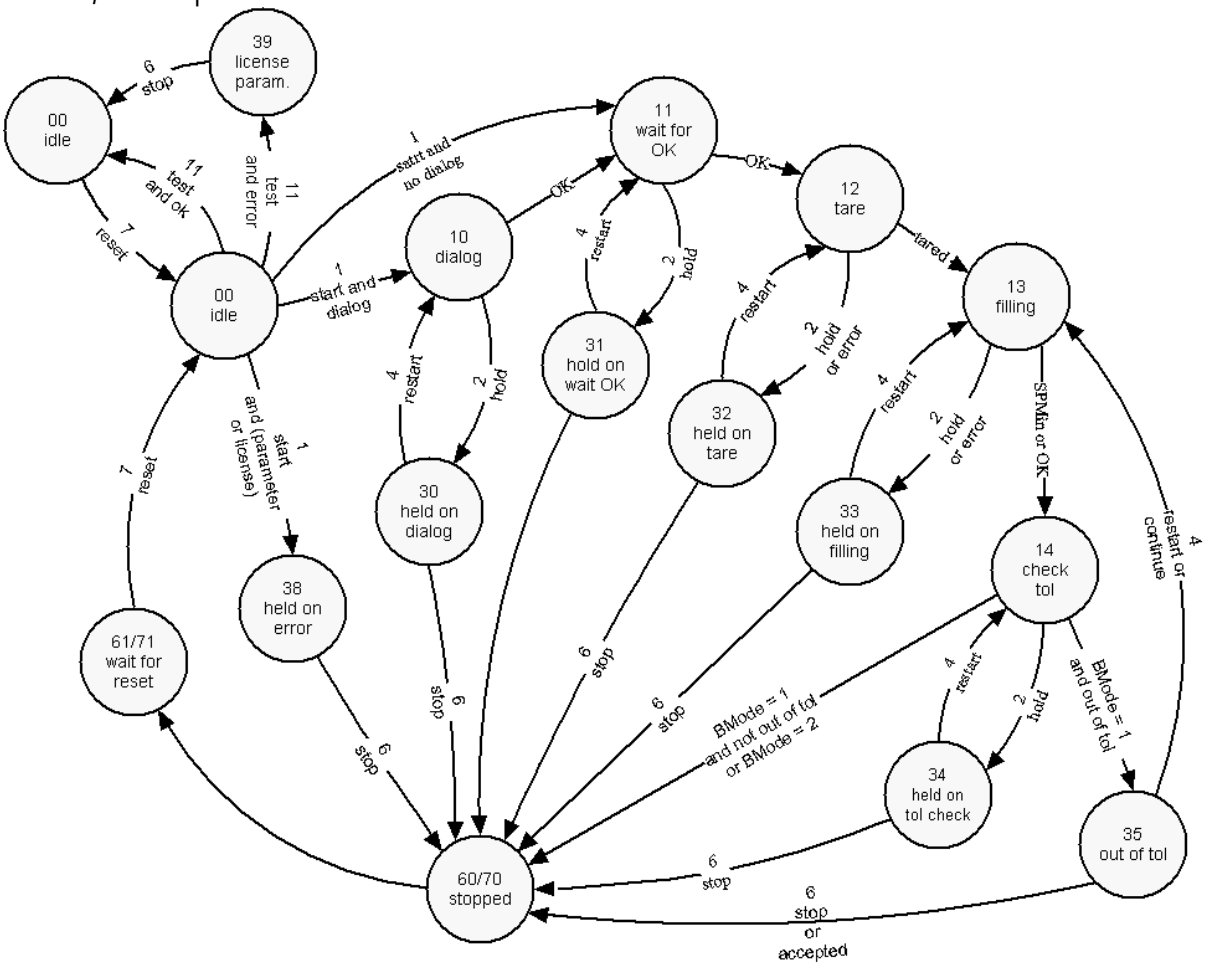
iBatchMode	Description	New tare weight	Gross setpoint
B1	Net batching	gross weight	tare + setpoint
B2	Charge batchingo	unchanged	old tare + setpoint
B3	Gross batching	0.0	setpoint
B4	Discharge batching	gross weight	tare – setpoint
B5	Charge/discharge batching	unchanged	old tare- setpoint
B6	Gross discharge batching	gross weight	setpoint

9.7.5 Manual addition

The phase for manual additions requests the operator to batch material manually.

Phase	FMANUAL	
Parameter	WP	Weighing point identification ('A', 'B', ...)
	cmd	Copy of iCommand
	iBatchMode	D1: with tolerance check, D2: without tolerance check and protocol. oActual = bSetpoint
	iSPMin	Address: batch completed (e.g. key), for more detailed information, see Interface parameters.
	iSPMout	Address: batching requested (e.g. lamp)
	bSetpoint	Setpoint
	oActual	Actual weight, added to report after completion
	iNegTol	Lower tolerance in kg or lb (0: no tolerance check)
	iPosTol	Uper tolerance in kg or lb (0: no tolerance check)
	bTextPara	Optional parameters: dialogue = ... (is checked – see dialogue), Default: no dialogue.

The phase waits for the activation bit iSPMin or for pressing the 'OK' key. iSPMin can be -1 or 0. In this case, manual batching can be terminated only by 'OK' at the instrument. The phase can be extended by means of the text parameters for entry of additional data after dosing. For using FMANUAL, licence PR1713/32 is required.



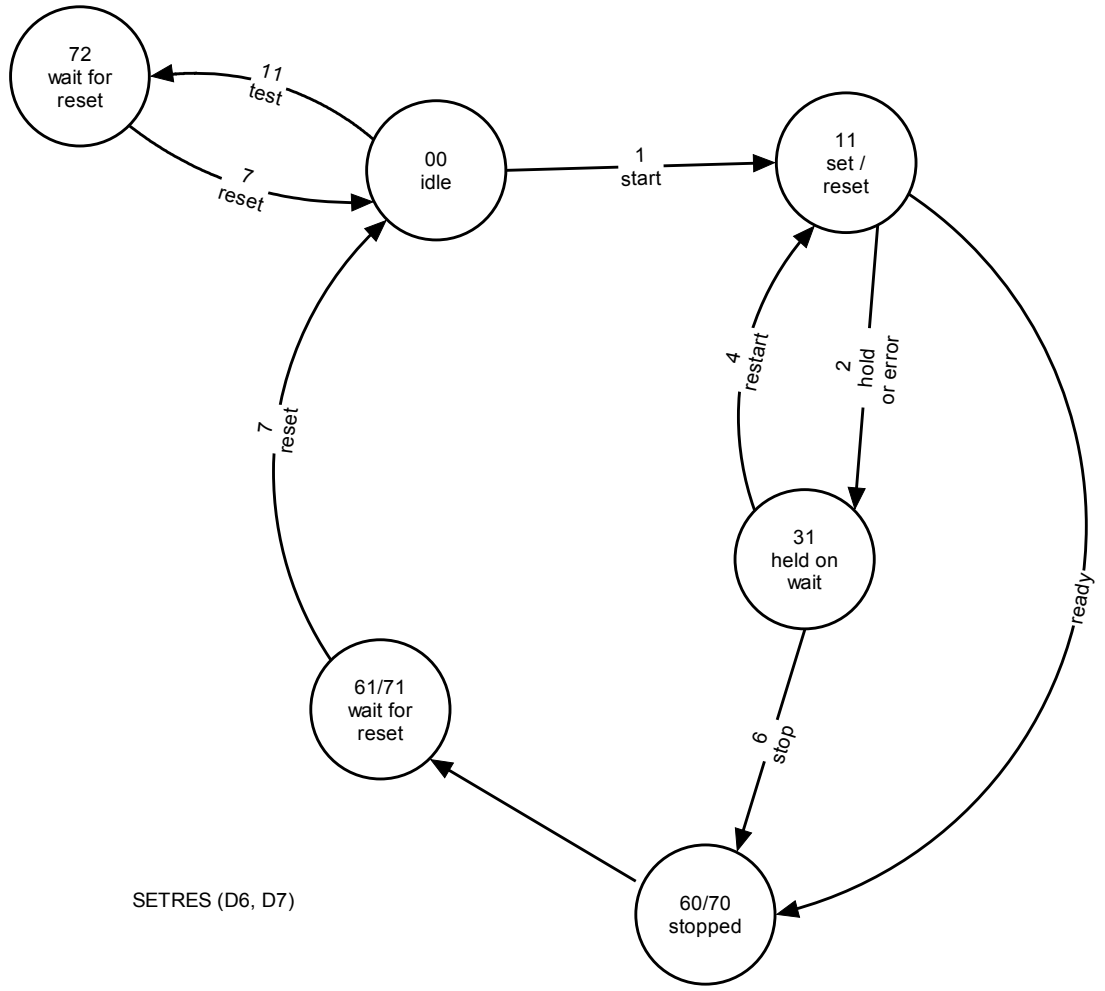
MANUAL (D1, D2)

Pha_005

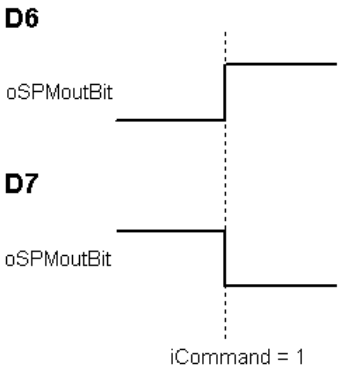
9.7.6 Set and reset bits

This phase is used for setting or resetting single SPM bits

Phase	FSETRES	
Parameter	WP	Weighing point identification ('A', 'B', ...)
	cmd	Copy of iCommand
	iSPMin	Not used
	iSPMout	Address of the SPM bit, which must be set or reset.



Pha_006

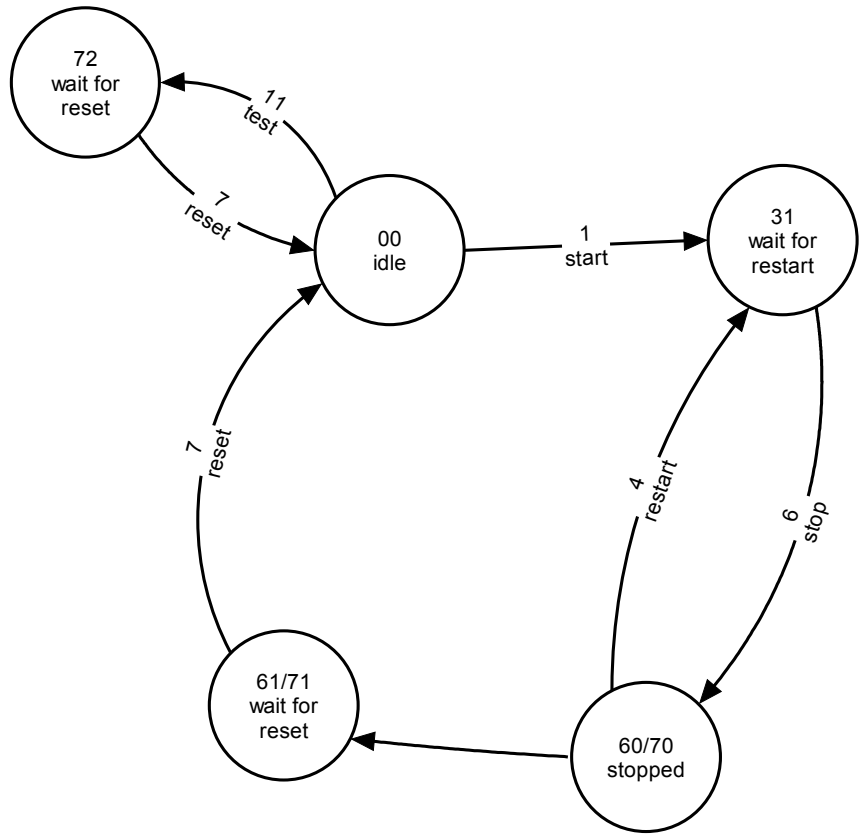


Pha_007

9.7.7 Stop

FSTOP changes into the HELD state directly and waits for the restart command.

Phase	FSTOP	
Parameter	WP	Weighing point identification ('A', 'B', ...)
	cmd	Copy of command iCommand
	iSPMout	Address: the phase waits for the addressed bit

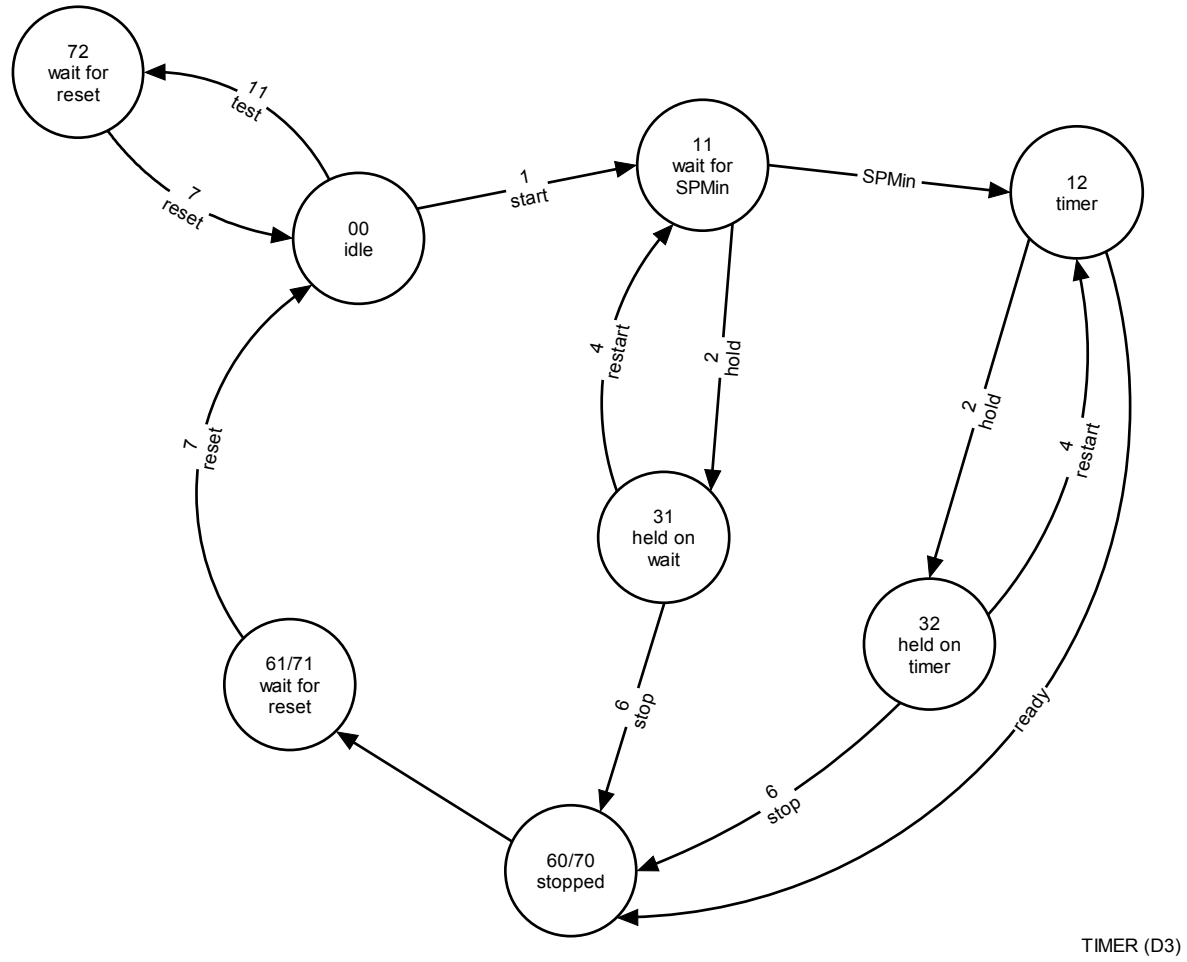


STOP (D4)

Pha_008

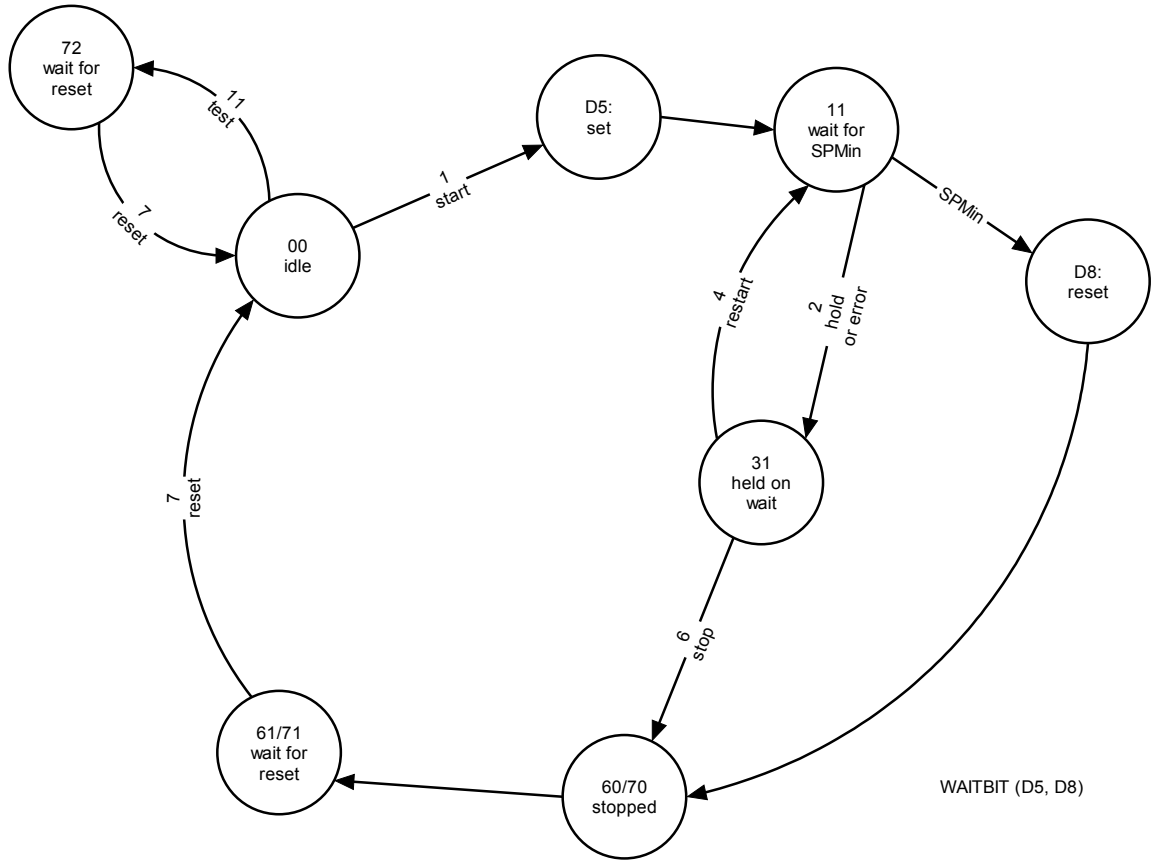
9.7.8 Timer

Phase	FTIMER	
Parameter	WP	Weighing point identification ('A', 'B', ...)
	cmd	Copy of iCommand
	iSPMin	Address: activates the timer, for further information, see Interface parameters
	iSPMout	Address: timer running
	bSetpoint	Time in seconds (resolution: PLC clock, e.g. 50 ms)
	oActual	Elapsed time in seconds

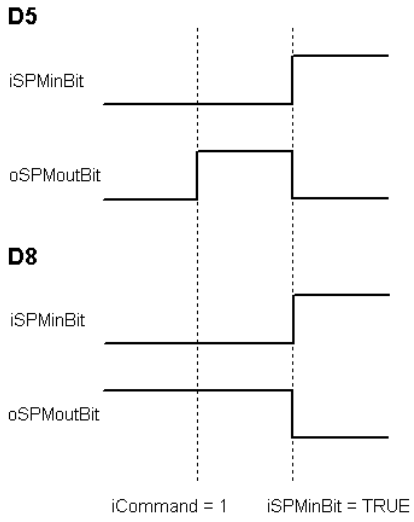


9.7.9 Waiting for a bit

Phase	FWAITBIT	
Parameter	WP	Weighing point identification ('A', 'B', ...)
	cmd	Copy of iCommand
	iBatchMode	Operating mode (for more details, see Interface parameters): • D5: oSPMoutBit is set, wait for iSPMinBit and reset oSPMoutBit • D8: wait for iSPMinBit, then reset oSPMoutBit
	iSPMin	Address: wait, until this bit is set
	iSPMout	Address: bit set or reset



Pha_010

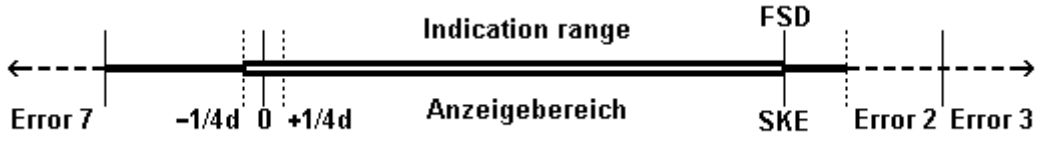


Pha_011

10 ERROR MESSAGES ON THE WEIGHT DISPLAY

The error statuses of the analog part are output on the weight display. They are displayed in code as 'Error X' .

Error messages on the weight display	
Error 1	Internal processor overflow (faulty calibration values)
Error 2	The input voltage is above f.s.d. + overload.
Error 3	The input voltage is above the permissible value of 36mV. However, the message can also be due to an error in the analog part or load cell, or a cable break.
Error 4	Weight value exceeds the display number of digits behind the decimal point.
Error 5	Weight value not available, e.g. weighing point busy
Error 6	Sense voltage out of tolerance
Error 7	Negative input voltage or faulty load cell connection
Error 8	ADC error, e.g. hardware defective or overload.
Error 9	No communication with the weighing point
Error 11	No weight value
Error 15	Different serial number



11 USED ABBREVIATIONS

Abbr.	Signification / comment
ASCII	Standard for description of characters during data transmission
BOOL	Data type, 1 single bit
CRC	Method for safeguarding a data transmission
d	Scale stepwidth
DINT	Data type, 32-bit integer value with polarity sign
EAROM	Erasable memory only for reading during operation
FSD	Fullscale value of the scale
ID	Identification number e.g. of a material
MD ...	Double integer address (32-bit) in the scratchpad memory
MW ...	Word address (16-bit) in the scratchpad memory
MX ...	Bit address in the scratchpad memory
NLE	NiceLabelExpress, Sartorius PC program
OPC	Interface in MS-windows operationg system
SPM-Address	Scratchpad-Memory: free addressible memory space
SPS	Programmable controller
WP	Weighing point, i.e. this scale

12 INDEX

R

S

T

U

V

W

Z

Sartorius Mechatronics T&H GmbH
Meiendorfer Straße 205
22145 Hamburg, Germany
Tel +49.40.67960.303
Fax: +49.40.67960.383
www.sartorius-mechatronics.com