SIEMENS

SIMATIC

Process Control System PCS 7 PCS 7 Advanced Process Library Readme V9.1 (online)

Readme

Version: 2020-07-23 (Online)

V9.1

A5E46730146-AA

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER
indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ WARNING
indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ CAUTION
indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Digital Industries
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E46730146-AA
Ⓟ 03/2021 Subject to change

Table of contents

Security information 1

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

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

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. These systems, machines and components should only be connected to the enterprise network or the Internet if and only to the extent necessary and with appropriate security measures (firewalls and/or network segmentation) in place.

You can find more information on protective measures in the area of industrial security by visiting:
https://www.siemens.com/industrialsecurity.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends performing product updates as soon as they are available and using only the latest product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer's exposure to cyber threats.

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

Overview 2

Note

Read these notes carefully; they contain important information and additional details about the PCS 7 Advanced Process Library .

The statements in this readme file take precedence over all the PCS 7 manuals.

Notes on installation

3

3.1 Scope of delivery

The scope of delivery includes the following library:

- PCS 7 Advanced Process Library V9.1

3.2 Hardware requirements

The installation is subject to the conditions of SIMATIC PCS 7 V9.1.

3.3 Software requirements

The blocks of the library require at least SIMATIC PCS 7 V9.1.

3.4 Installing the PCS 7 Advanced Process Library

Exit all applications before you start setup.

Install the library via the PCS 7 system setup or start the Setup.exe program of the PCS 7 Advanced Process Library. You will receive all further information during the installation process.

The library is copied to the same drive on which the PCS 7 basic software is installed. Installation requires about 1.6 MB of free memory on the hard disk.

Following the installation, you can find the blocks of the PCS 7 Advanced Process Library and templates under "PCS 7 AP Library V9.1".

Practical examples can be found in the example project "APL_Example_*xx*", ("*xx*" stands for one of the installed language variants EU = European, CHS = Chinese, FE = Japanese.

To update your AS project, transfer the blocks to your project and then perform a block type import.

Note

If you do not use CPU 410-5H , the **"Type Change in Run" (TCiR)** function is not available for an upgrade from <V9.1 to ≥V9.1. In order to receive the delta compilation and delta load capability in your project, use the blocks from the previous version of the PCS 7 Advanced Process Library. Note that in this case you can not use the new features and improvements described in the section "New features and changes compared to earlier versions".

3.5 Uninstallation

If you uninstall an older PCS 7 Advanced Process Library using Windows mechanisms, you must afterwards install the current/latest version of the PCS 7 Advanced Process Library V9.1 again. It is not possible to uninstall via the product setup.

New features and changes as compared to previous versions 4

Note

For possible behavior changes in earlier versions, please refer to the readme of the previous version.

NOTICE

The behavior of the channel blocks changes in the following cases:

- **When the CFC function "Generate module driver" is not used**
- **When 3rd party diagnostic blocks are used**
- **When the function block without** `DataXchg1` **connection to PCS 7 Basis Library function blocks is used**

Usually the `DataXchg1` input of the channel block is automatically connected when the module drivers are generated. Users who do not use this function must configure "1" for the `Feature.Bit27` "Initial execution identifier of the diagnostic blocks" at the corresponding channel blocks.

`Feature.Bit27` must also be set if third-party diagnostic blocks are used that do not provide an initial execution identifier, or if the channel block is used without a `DataXchg1` connection.

4.1 Version 9.1

4.1.1 What's New in Version 9.1?

- The blocks of the **Condition Monitoring Library (CML)** have been integrated into the APL. The F-blocks for the fail-safe partial stroke test application (`F_PST` and `S_PST_S`) are **not** included. They are freely available for download at the address <link will be specified at a later date>.
  The following blocks have been added: PumpMon, VlvMon, PrDrpMon, SteadyState, PST, PST_Calc, PST_OrQC, PST_Time.
- The **"Convert"** block family has been extended by the two `StruCmIn` functions (split a structure of several values into discrete outputs) and `StruCmOut` (combine discrete inputs into one structure).

- **Block-internal error states**, which are output to `ErrorNum` as error numbers, are displayed in the **standard view**.
  The status bits of the following blocks have been adapted for this: ShrdResL, ShrdResS, MotL, MotS, MotSpdL, MotRevL, MotSpdCL, VlvL, VlvS, Vlv2WayL, VlvAnl, VlvMotL, VlvPosL, Intlk02, Intlk04, Intlk08, Intlk16, MonAnL, MonAns, MonDiL, MonDi08, OpAnl, OpAns, OpDi01, OpDi03, PIDConL, PIDConR, PIDConS, PIDStepL, FmCont, FmTemp, ModPreCon, ConPerMon, Ratio, MPC10x10, KalFilt, CountScL, CntOhSc, CountOh, TotalL, DoseL, SelA16In, SelAn16L, TimeTrig, SteadyState, PST
- At the **channel block Pcs7AnOu**, `Feature.Bit5 =1` can be used to invert the effect of the `PV_In.Value` bzw. `SimPV_In.Value` input parameter. This saves a separate block for inverting when using a fail-open valve, for example.
- The **channel blocks** have been given a **"Channel active" identifier**. The identification is provided via the `DataXchg1` input of the upstream diagnostic block of the PCS 7 Basis Library and evaluated in the channel block.
  The following channel blocks evaluate this "Channel active" identifier: FbAnIn, FbAnOu, FbAnTot, FbDiIn, FbDiOu, FbDrive, FbEnMe, FbSwtMMS, FmCont, FmTemp, Pcs7AITC, Pcs7AnIn, Pcs7AnOu, Pcs7Cnt1, Pcs7Cnt2, Pcs7Cnt3, Pcs7DiCnt, Pcs7DiIn, Pcs7DiIT, Pcs7DiOu, Pcs7HaAI, Pcs7HaAO.
  The "Channel active" identifier is reset for channels that have been deactivated for HW Config. The channel blocks react to this with signal status 16#68 (uncertain, maintenance demanded) and `Bad = false`.

Note

Behavior change for channels deactivated in HW Config.

There has been a change in behavior in the channel blocks for channels deactivated in HW Config. Up to now, these reacted with signal status 16#00 (bad) and Bad=true. The behavior changes to signal status 16#68 (uncertain, maintenance request) and Bad=false.

Behavior change when the "Generate module drivers" CFC function is not used or when 3rd-party diagnostic blocks are used

Normally the DataXchg1 input of the channel blocks is automatically switched when the module driver is generated. DataXchg1 now also contains the new "Channel active" identifier. Therefore ensure that the 3rd party diagnostic blocks provide this identifier or configure the identifier for "Channel active" yourself if necessary: 2#xxxxxxxxxxxx1xxx (Byte0 Bit 3 = 1).

- The information on **interlocks, motor protection function, external error and process control error** are now also available as additional outputs (`PermitOut, IntlockOut, ProtectOut,.., TripOut, FaultExtOut, CSF_Out`) for further processing in your user program.
  The following blocks have been extended for this: FmCont, FmTemp, PIDConL, PIDConR, PIDConS, PIDStepL, DoseL, MotL, MotRevL, MotS, MotSpdCL, MotSpdL, Vlv2WayL, VlvAnL, VlvMotL, VlvPosL, VlvL, VlvS, MonAnL, MonAnS, MonDiL, MonDiS, OpDi01, OpDi03
- The information "**Internal simulation**" is now also available at the new `SimAct`output.
  The following blocks have been extended for this: DoseL, FmCont, FmTemp, ModPreCon, PIDConL, PIDConR, PIDConS, PIDStepL, VlvPosL, VlvMotL, VlvAnL, VlvS, Vlv2WayL, VlvL, MonAnS, MonDiL, MonDiS, MonAnL, MPC10x10, TotalL, Ratio, MotL, MotRevL, MotSpdCL, MotSpdL, MotS, KalFilt.

- The **alarm limits** of the setpoint/actual value difference have been made available at the PIDConL, PIDStepL, PIDConR, MotSpdCL, VlvAnL, VlvPosL blocks as the **output** `ER_AH_LimOut, ER_AL_LimOut` . The alarm limits for overdosing and underdosing (relative to the dosing setpoint) have been made available for DoseL as `ER_AH_TolOut, ER_AL_TolOut` output.
- The range limits and units are also available at the CountScL, MonAnL, MonAnS, MotSpdCL, OpAnL, OpAnS, Ratio, TotalL, PIDConL, PIDConS, PIDConR, PIDStepL, SteadyState, DoseL blocks as `xxxOpScaleOut` and `xxxUnitOut` outputs.
- At the **CntOhSc counter block**, the `CntHi1LimOut, CntHi2LimOut, HrsHi1LimOut, HrsHi2LimOut` outputs have been added for further interconnection of the limits.
  The `OutAH_LimOut, OutWH_LimOut, OutTH_LimOut, OutTL_LimOut, OutWL_LimOut, OutAL_LimOut` outputs have been added to the **CountScL** block accordingly.
- The **analog auxiliary values** together with their units are also available as output (`UserAna1Out, UserAna2Out, UA1unitOut, UA2unitOut`) for further interconnection. This affects the following blocks: **MotL, MotRevL, MotSpdL, MotSpdCL, VlvL, VlvMotL, VlvPosL, VlvAnL, Vlv2wayL, DoseL, MonAnL, MonDiL, PIDConL, PIDConR, PIDStepL**
- In order to increase the number of free bits for OS information, **all technological blocks** except the Small blocks have a second `UserStatus2` input. This affects the following blocks: OpAnL, OpDi01, OpDi03, OpStations, OpTrig, MonAnL, MonDiL, MonDi08, ConperMon, FmCont, FmTemp, ModPreCon, PIDConL, PIDConR, PIDStepL, Ratio, MPC10X10, KalFilt, DoseL, MotL, MotRevL, MotSpdCL, MotSpdL, ShrdResL, Vlv2WayL, VlvMotL, VlvAnL, VlvPosL, Intlk02, Intlk04, Intlk08, Intlk16, Event, EventNck, EventTs, Event16Ts, SelA16In, SelAn16L, TimeTrig, VlvMon, PumpMon, PressPropMon
- The behavior of the `xx_xx_Act` outputs "**Limit active**" with deactivated limit message (`xx_xx_MsgEn =0`) can be configured via `Feature Bit` . When the deactivated limit message and `Feature Bit = 1` is the output for "**Limit active**" is also reset.
  The new `Feature Bit` is available for the following blocks:
  `Feature Bit3`: **MonAnL, MonAnS, PIDConS, AV, CountOh, CountScL, TotalL, PumpMon**
  `Feature2 Bit12`: **PIDConL, PIDConR, PIDStepL, ConPerMon, DoseL, VlvPosL, MotSpdCL, FmCont, FmTemp, VlvAnL**
- The **PIDConL, PIDConR, PIDStepL controller blocks** have been extended by the **"Display auxiliary values"** function. Two analog auxiliary values `UserAna1, UserAna2` can now be displayed together with their `UA1unit, UA2unit` units in the **standard view**.
- The process input signals can now also be evaluated in inverted form at the **MonDiL, MonDiS, MonDi08 monitoring blocks**. This can be configured at the newly added `InvIn, InvIn1..8` inputs.
- With `Feature2 Bit13 = 1`, an error can also be reset at the MotL, MotRevL, MotSpdcL, MotSpdL, VlvMotL, VlvPosL, Vlv2WayL, VlvL **motor and valve blocks** in **local mode** (`LocalSetting = 1,3`) via the **reset button** in the standard view.
- The analog **position feedback** Rbk at the **VlvPosL** block can now also be monitored in the **intermediate position** when **position control is switched off**.
  After a stop control, the manipulated variable is tracked for the time from `V_MonTiStatic` to the position feedback (`MV_Out = RbkOut`). The manipulated variable is then held and can be monitored.

- The Stop output at the **VlvAnL** block. Stop is set when the analog valve has reached its position and no further control is pending.
- The **MotSpdCL** block was extended by a **directional interlock**.
- At the **DoseL** block, the automatically determined **dribbling quantity** `DribbOut` can be applied for a complete download using Feature2 Bit7 `=1`. The `OldDribbOut` input has been added for this.
- The **TimeTrig** block was extended by a **external time input** and by **local time**. The calculation of the local time is performed via the `FC296 LOC_TIME_BAS` function from the PCS 7 Basis Library.
- At the **TimeTrig** block, the periodic output of a trigger was extended by the **minute output**.
- The **FlipFlop** block was extended by a **JK function**.

4.1.2 Changes in version 9.1

- Suitable `SP_LiBy`, `SP_LiST` inputs were added in byte format to the **FbDiOu** block.
- The `SP_ChnST` output for the return connection to a technological block was added to the **FbDiOu** block.
- The signaling behavior for high-precision time stamping with toggled signal sequences has been improved at the **Pcs7DiIT** block.
- If the signaling is 0-active (`Feature.Bit29 =1`) and "last stored value" is selected as startup behavior (`Feature.Bit0 =1`), the limit messages are no longer triggered after a complete download of the user program. This has been improved for the following blocks: **AV, CntOhSc, ConPerMon, CountOh, CountScL, DoseL, FmCont, FmTemp, MonAnL, MotSpdCL, PIDConL, PIDConR, PIDStepL, TotalL, VlvMon, VlvAnL, VlvPosL.**
- The function "**Bumpless switchover to 'Automatic' mode for valves, motors, and dosers**" at the **MotL**, **MotRevL**, **MotSpdCL**, **MotSpdL**, **VlvL**, **Vlv2WayL**, **VlvMotL**, **VlvPosL**, **VlvAnL**, **DoseL** blocks also take into account the implausible values of the automatic commands with `Feature2.Bit3` and a bad status with `Feature2.Bit10` .
- Default values of parameters have changed at the following blocks:
  MotSpdCL: `SP_Int 0.0`
  VlvPosL: `V_MonTiDynamic 30.0`
  Check these parameter values after an upgrade.
- The function of the `Stop` output has changed at the **VlvPosL** and **VlvMotL** blocks. The calculations of the `Status3.Bits24,25` have been adapted for this.
  Previously, `Stop =1`: Motor is stopped and the valve is in an intermediate position.
  New behavior `Stop =1`: Motor is stopped regardless of the valve position.
- Am outgoing motor trip error (`Trip =0->1`) at the **VlvPosL, VlvMotL** blocks must first be reset before it can move to the "Closed" or "Opened" idle position due to a monitoring error of the feedback signals.
- With **VlvAnL**, an interlock now has higher priority than the tracking via `MV_TrkOn =1` with `MV_TrkRbk =0`. Thus this block now behaves like all controller blocks and the **VlvPosL**.

- With `SafePos =2` (stop), the manipulated variable `MV` of the `RbkOut` position feedback is tracked at the **VlvAnL** block. **VlvAnL** thus behaves like VlvPosL.
- The manipulated variable limit outputs `MV_HiAct, MV_LoAct` at the **VlvAnL** block are now set when the manipulated variable limit is reached.
- At the **VlvAnL** block, when `Feature2.Bit10 = 1` (*inclusion of bad quality of automatic commands* **...**) a bad signal status at the `OpenAut.ST, CloseAut.ST, MV_Ext.ST` inputs becomes effective only after switching over to **automatic mode** and not already in **manual mode**. With bad signal status, the block sets the idle position configured at `SafePos`.
- At the **VlvPosL** block, the `OpenForce`, `CloseForce`, `StopForce` inputs have a higher priority than `MV_ForOn`. An error number at the `ErrorNum` output is not output if `MV_ForOn` or one of the other force inputs is set.
- At the **VlvPosL** block, the activation enable for position control and direction-dependent interlocking has been improved.
- The manipulated variable limit outputs `MV_HiAct, MV_LoAct` at the **VlvPosL** block are now also set if the manipulated variable limits are reached when the position control is switched off.
- The **controller blocks**, **PIDConL, PIDConR, PIDStepL, PIDConS, FmCont, FmTemp** can now continue to be operated even if a #NaN process value is present. The block acts as a sink, thus ensuring that no #NaN values are passed on to outputs.
- The behavior of the during setpoint tracking `SP_TrkPV = 1` has been improved at **PIDConL, PIDConR, PIDStepL, PIDConS** blocks.
- At the **PIDStepL** block, messages for the position feedback `RbkWH_Act, RbkWL_Act` are no longer output for the "**Step controller without position feedback**" parameter assignment `WithRbk = 0`.
- The switchable inputs `OpenLi, StopLi, CloseLi` can be activated at the **PIDStepL** block via the `CtrlLiOp` parameter and `Feature.Bit7` in manual mode. This means that the `OpenLi, StopLi, CloseLi` inputs can now be used independently of `ModLiOp` and therefore independently of the operating mode switchover.
- At the **TimeTrig** block, it is now also possible to change the delay time of the single trigger while the single trigger is running. This means that the single trigger behaves in the same way as the periodic trigger.
- At the **FlowCorr** block, the `Flow.ST` status "Bad, device related" `16#68` is output if one of the relevant inputs for the formation of `Flow.ST` is not defined.

4.1.3 Update information for version 9.1

The following applies for the software update from version 9.0 SP 3 >=Update 4 to version 9.1:

1.	Make Textual Interconnections	**Yes**
2.	Compile AS	**Complete compilation**
3.	Download AS program	**Complete download**

4.	AS STOP necessary	**Yes (AS STOP is not necessary if a CPU 410-5H with TCiR is used)**
5.	Compile OS	**Complete compilation**

Note

Updating the blocks from the PCS 7 Advanced Process Library with TCiR

"Type import - Changed default value" dialog box

- If information on changed internal local default values in the static area is displayed during type import in the dialog box, these changes are not relevant. A complete download in STOP mode is not necessary.
- If information on changed default values is displayed at block I/Os during type import in the dialog box, these new default values must be applied in the "Update Block Types" dialog.

List of changed blocks

Name	Number	Block version	Interface change (AS STOP is not necessary if a CPU 410-5H with TCiR is used)	Code change
AssetM	FB1840	9.5	Yes	No
AV	FB1903	5.5	Yes	Yes
CntOhSc	FB1803	5.5	Yes	Yes
ConPerMon	FB1805	5.5	Yes	Yes
CountScL	FB1806	5.5	Yes	Yes
CountOh	FB1864	5.5	Yes	Yes
DoseL	FB1809	5.5	Yes	Yes
Event	FB1811	5.5	Yes	Yes
Event16Ts	FB1887	5.5	Yes	Yes
EventNck	FB1904	5.5	Yes	Yes
EventTs	FB1812	5.5	Yes	Yes
FbAnIn	FB1813	9.5	Yes	Yes
FbAnOu	FB1814	9.5	Yes	Yes
FbAnTot	FB1817	9.5	Yes	Yes
FbDiIn	FB1815	9.5	Yes	Yes
FbDiOu	FB1816	9.5	Yes	Yes
FbDrive	FB1905	9.5	Yes	Yes
FbEnMe	FB1908	9.5	Yes	Yes
FbSwtMMS	FB1907	9.5	Yes	Yes
FlipFlop	FC389	5.5	Yes	Yes
FlowCorr	FB1916	9.5	Yes	Yes
FmCont	FB1818	9.5	Yes	Yes
FmTemp	FB1819	9.5	Yes	Yes
Integral	FB1823	5.5	Yes	No

Intlk02	FB1824	5.5	Yes	Yes
Intlk04	FB1825	5.5	Yes	Yes
Intlk08	FB1826	5.5	Yes	Yes
Intlk16	FB1827	5.5	Yes	Yes
KalFilt	FB1925	9.5	Yes	Yes
LPOptim	FB1844	5.5	Yes	No
ModPreCon	FB1843	5.5	Yes	Yes
MonAnL	FB1845	5.5	Yes	Yes
MonAnS	FB1912	5.5	Yes	Yes
MonDi08	FB1847	5.5	Yes	Yes
MonDiL	FB1848	5.5	Yes	Yes
MonDiS	FB1913	5.5	Yes	Yes
MotL	FB1850	5.5	Yes	Yes
MotRevL	FB1851	5.5	Yes	Yes
MotS	FB1910	5.5	Yes	Yes
MotSpdCL	FB1854	5.5	Yes	Yes
MotSpdL	FB1856	5.5	Yes	Yes
MPC10x10	FB1920	5.5	Yes	Yes
MPC_OpFB	FB1922	5.5	Yes	No
OpAnL	FB1865	5.5	Yes	Yes
OpAnS	FB1915	5.5	Yes	Yes
OpDi01	FB1866	5.5	Yes	Yes
OpDi03	FB1867	5.5	Yes	Yes
OpStations	FB1901	5.5	Yes	No
OpTrig	FB1868	5.5	Yes	Yes
Pcs7AITC	FB1933	9.5	Yes	Yes
Pcs7AnIn	FB1869	9.5	Yes	Yes
Pcs7AnOu	FB1870	9.5	Yes	Yes
Pcs7Cnt1	FB1833	9.5	Yes	Yes
Pcs7Cnt2	FB1834	9.5	Yes	Yes
Pcs7Cnt3	FB1835	9.5	Yes	Yes
Pcs7DiCn	FB1935	9.5	Yes	Yes
Pcs7DiIn	FB1871	9.5	Yes	Yes
Pcs7DiIT	FB1872	9.5	Yes	Yes
Pcs7DiOu	FB1873	9.5	Yes	Yes
Pcs7HaAI	FB1931	9.5	Yes	Yes
Pcs7HaAO	FB1932	9.5	Yes	Yes
PIDConL	FB1874	5.5	Yes	Yes
PIDConR	FB1875	5.5	Yes	Yes
PIDConS	FB1830	5.5	Yes	Yes
PIDKernR	FB1877	5.5	Yes	Yes
PIDStepL	FB1878	5.5	Yes	Yes
Polygon	FB1881	5.5	Yes	Yes
RateLim	FB1882	5.5	Yes	Yes

Ratio	FB1883	5.5	Yes	No
SelA16In	FB1888	5.5	Yes	Yes
SelAn16L	FB1889	5.5	Yes	Yes
Snd_AnaVal	FB1893	9.5	Yes	No
Snd_DigVal	FB1891	9.5	Yes	No
ShrdResL	FB1917	5.5	Yes	Yes
ShrdResS	FB1914	5.5	Yes	Yes
TimerP	FB1810	5.5	Yes	Yes
TimeTrig	FB1802	5.5	Yes	Yes
TotalL	FB1906	5.5	Yes	Yes
Vlv2WayL	FB1897	5.5	Yes	Yes
VlvAnL	FB1896	5.5	Yes	Yes
VlvL	FB1899	5.5	Yes	Yes
VlvMotL	FB1900	5.5	Yes	Yes
VlvPosL	FB1918	5.5	Yes	Yes
VlvS	FB1911	5.5	Yes	Yes

Note

- Blocks for which only attributes, message texts, and/or version number were changed, are not listed.
- If "Yes" is listed in the column "Interface change" and "No" in the column "Code change", the interfaces were prepared for a future function or the order of the static variables (VAR area) were optimized.

Information on configuration and operation 5

- The **blocks in the library** have been designed exclusively for **S7-400** automation systems and are not suitable for the configuration of failsafe applications.
- For **AS blocks** that include **ALARM_8P blocks**:
  The BOOL data type is allowed as an **message associated value** (not allowed are: Bit array), BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME, COUNTER, TIMER. Copy a STRING data type into an ARRAY OF BYTE . Structures must be interconnected with the message associated values via a converter block; otherwise, these message associated values are always displayed with "0".

  Note

  When an associated value is interconnected for the first time or the value of the data type is changed, this takes effect after a CPU restart.

- The **templates** contain CFC charts with ready-made technological functions. These charts include the blocks and interconnections that are required as default for each of the functions. You can continue to use these charts as a template for individual automation solutions in the future. The templates are described in detail in the online help.
- You are not permitted to set the **minimum cycle time** of OB1 to a value other than 0. For **CPU 410 PA SMART** , the minimum cycle time of OB1 is permanently set to 200 ms.
- Note that the **signal status** of the individual block I/Os is not evaluated by the **SFC type**, since the SFC type does not provide any features of the Struct type.
- **Interconnecting inputs and outputs with APL structures to inputs and outputs of blocks without structures.**
  This CFC function is available from PCS 7 V7.1 SP1 and enables the **interconnection of APL block parameters with blocks of other libraries without converter block**. This simplifies the engineering. However, this function may result in an interruption of the Flow signal status.
  Example:
  A BOOL output with status information can be connected directly to a BOOL input without status information and vice versa. If the signal status information is to be maintained, the signal status must be routed around the "non-APL logic" using converter blocks.
- The user-defined data types (**UDT**) from 1 to 50 are reserved for APL.
- The **FbAnIn**, **FbAnOu**, **FbDiIn** and **FbDiOu** channel blocks support **FF field devices** with analog or digital input and output values.
  The first implementation stage of using FF field devices behind **Profibus Slave AB7000** (introduced for PCS 7 V7.0) is no longer supported.
- The use of different channel block types for the same drive or frequency converter device type is not permitted.
- If the **EventTS/Event16TS** block is interconnected with the `EventTsIn` input in a technological block, the **EventTs/Event16TS** and the technological block must be installed in the **same** cyclic interrupt OB of the **same** CPU.

- Connection ManModLi "**Manual mode" via interconnection or SFC**"
  The attribute S7_contact is missing on the ManModLi and must be set manually if needed at the block interface. You can reach the interface by double-clicking the appropriate block in the block folder. To retain backward compatibility with existing projects, the attribute can no longer be set because otherwise the existing connections between the SFC and the block would be changed.
- The use of different channel block types for the same drive device type or frequency converter is not permitted.
- **Feature "Separate evaluation of excluded and simulated interlock signals"**
  If the feature is enabled using Feature2.Bit2 = 1, the signal may **not** be inverted per CFC for an interconnection to the inputs, otherwise the **bypass display** cannot be refreshed for system related reasons.
  This affects the following interlock inputs:
  Intlk02, Intlk04, Intlk08, Intlk16: In01, In02, ...
  DoseL, MotL, MotRevL, MotSpdL, MotSpdCL, VlvL, Vlv2WayL, VlvAnL, VlvMotL, VlvPosL: Permit, Interlock, Protect
  MotRevL, MotSpdCL: Permit, Interlock, Protect, PermRev, IntlRev, ProtRev
  MotS, VlvS, OpDi01, OpDi03, PIDConL, PIDConR: Interlock
  For the **inversion of the interlock signals**, the inputs InvIn01, InvIn02... can be used at the interlock blocks.
- **LAG** block
  A change to the SampleTime first takes effect after a change to the delay time LagTime or after resetting the block (Reset = 1).
- Before using the **PCS 7 example project Advanced Process Control** (APL_Example_xx, xx refers to the language version), it recommended to update the function blocks of the PCS 7 Advanced Process Library and the basic data contained within it using the OS Project Editor.
- Channel block **Pcs7DiOu**:
  When using the digital outputs of modules with the order numbers **6DL1 133-6EW00-0PH1** and **6DL1 132-6HD50-0PK0** and configuring substitute values in the HW configuration, the configured substitute value must be entered at the StartVal input parameter at the associated channel blocks **Pcs7DiOu** in the CFC Editor and Feature Bit 0 must be set to the value "1" for the substitute value to be applied as the startup value.
- Channel block **Pcs7Cnt1**:
  For the acyclic data traffic to an **FM350-2 counter module** to function correctly, at least one **Pcs7Cnt1** channel block must be configured in the user program for each FM350-2 counter module configured in the HW configuration.
- Channel blocks **FbDrive**, **FbSwtMMs**:
  Feature Bit 9 (as of CFC version 9.0.1.2) is used to select whether the CFC installs the standard diagnostic blocks of the PCS 7 Basis Library or the diagnostic blocks of an AddOn Library for "Generate module drivers".
  0 = Diagnostic blocks from PCS 7 Basis Library, 1 = Diagnostic blocks from an AddOn Library.
- **MonDiL, MonDiS** blocks:
  A change of the Color parameter does not take effect until after a picture change.

- **Channel blocks** - Input `Feature`:
  To prevent duplicate configurations, specific Feature bits are automatically configured by the **PCS 7 driver generator** during CFC compiling with the "**Generate module drivers**" function based on the settings made in the hardware configuration. For this reason, the `Feature` parameter **must not** be interconnected. Otherwise, any existing interconnections could be deleted.
- When configuring `Feature2 Bit3=1` **"Control priority for an invalid and implausible input command"** on the **MotRevL** and **MotSpdCL** motor blocks, note that in automatic and local mode (LocalSetting =1/3) a direct change of direction is also possible with `Feature Bit7 =0`. If a command is given in the opposite direction, the motor first goes to stop and thereby returns to a plausible state. Then, the motor changes to the opposite direction. The configured enable time `IdleTime` is maintained in any case.
- For the controllers **PIDConL**, **PIDConR** and the motor block **MotSpdCL**, the function **"Delay alarm for system deviation with setpoint steps"** `(ER_AL_DFac, ER_AH_DFac >0)` only works for the limit in the direction of the setpoint step. If the error signal exceeds the opposite limit when the process value overshoots, undershoots, the delay time at the `ER_A_DCOut` output is reset and the corresponding `ER_A_DC` or `ER_AH_DC.` on-delay goes into effect.
- The `Gain` parameter must not be set at `>100` at the **PIDConR** block. Using the scaling factors `NormPV, NormMV`, it is possible to set the effective controller gain `GainEff` so that a controller gain at `Gain >100` is not necessary.
- The **PST** and **PST_Calc** blocks and for the **partial stroke test** uses the following IEC functions AD_DT_TM (FC1), CONCAT (FC2), DT_DATE (FC6) DT_TOD (FC8), I_STRING (FC16, NE_DT (FC28) and RIGHT (FC32) from the STEP 7 Standard Library. Therefore, before using the two blocks **PST** and **PST_Calc** , the **reserved area for other applications** must be adapted in the **CFC settings**. Details can be found in the manual in section "**Compile and load logic of the partial stroke test**".
- If the **time stamp of the I/O** is to be used with `TimeStampOn = 1` set for an **EventTS instance**, it must be ensured that `Feature.Bit5 = 0` for **all** connected **Pcs7DiIT** channel blocks. The time stamping will not work reliably for this EventTS instance.

Notes on the documentation 6

- The terms or message classes Alarm, Warning, Tolerance or the corresponding acronyms in icons and colors used in the documentation are not valid when using user-configured message classes.
  These terms and colors depend on the project-specific setting.
  The **user-configured message classes** are not supported by block icons from the old template pictures „@PCS7TypicalsAPLV7.PDL and „@TemplateAPLV7.PDL".
- The following has to be replaced in the section "**Controlling a device infeed or inverter enable**" of the "**MotSpdCL functions**".
  - OLD:
    When the function is activated, the motor can also be started directly. When starting forwards or backwards the device infeed or inverter enable is also activated. The frequency converter automatically determines the correct sequence. The motor can now be stopped and restarted without having to switch off the device infeed or inverter enable. The activation of the device infeed or inverter enable for starting the motor can be performed in the automatic, manual and local operating modes with local setting 1 or 3, or by forcing.
  - NEW:
    When the "device infeed" function is activated, the motor can also be started directly: The device infeed is also activated when starting in forward or reverse. The frequency converter automatically determines the correct sequence. The activation of the device infeed for starting the motor can be performed in the automatic, manual and local operating modes with local setting 1 or 3, or by forcing.
- The following must be replaced in the "**Redundancy**" section of the "**Description of Pcs7AnIn**" chapter.
  - OLD:
    If both signals deliver a "bad" quality code, the substitute value is passed on to the process with a "bad" quality code
  - NEW:
    If both signals deliver a "bad" quality code, output `Bad` *= 1 is set and the last valid value, a substitute value or the invalid value takes effect depending on the configuration of the* `Feature` *parameter.*
- The following information is missing in the **Area of application for Pcs7AnIn** section of "**Description of Pcs7AnIn**" chapter:
  Note:
  You can use the block to record temperatures in °C and °F. Kelvin is not supported.
- The following paragraph must be amended in the "**Configuration**" section of the "**Description of Pcs7AnIn**" chapter.
  If you use the block for **temperature measurement** and the `Mode` input has been connected by the driver generator, the unit at the `PV_InUnit` input is automatically set to the code 1001 (°C) or 1002 (°F) by the block depending on the hardware configuration.
- In the **I/Os of Pcs7AnIn** chapter, there is no note for the `PV_InUnit` input parameter that this input can be written back by the block algorithm.

- The following must be replaced in the "**Redundancy**" section of the "**Description of Pcs7AnOu**" chapter.
  - OLD:
    If both signals deliver a "bad" quality code and an "E/E redundancy warning" (`Mode` *= 16#x5xxxxxx) is issued, 50 % of the value of the application program is passed on to the process.*
    If both signals deliver a "bad" quality code, the last process value is retained.
  - NEW:
    If both signals deliver "good" quality code and an "E/E redundancy warning" `(Mode` *= 16#x5xxxxxx) is issued, 50 % of the* `PV_In` *value takes effect at each of the outputs* `PV_Out` *and* `PV_OutSlv`*.*
    If both signals deliver "bad" quality code, 100 % of the `PV_In` *value takes effect at each of the outputs* `PV_Out` *and* `PV_OutSlv`*.*
- "**Description of TimeTrig**"
  - The `DailyOn`, `WeeklyOn`, `MonthlyOn` and `MinuteOn` inputs mentioned in the text are not available on the block for the selection of the **periodic trigger**. The selection of the periodic trigger is made with the `PerMode` input.

`PerMode=0`	No periodic trigger
`PerMode=1`	Daily periodic trigger
`PerMode=2`	Weekly periodic trigger
`PerMode=3`	Monthly periodic trigger
`PerMode=4`	Minute periodic trigger, the trigger is always released at the same minute of an hour.

- Chapter "Status word of **FbSwtMMS"**"
  The status word assignment is as follows:

```
Bit  0 FdkRev
Bit  1 FdkOff
Bit  2 FdkFwd
Bit  3 OverlAct
Bit  4 LockTmAct
Bit  5 FdkAuto
Bit  6 ErrAct
Bit  7 WarnAct
Bit  8 Status8
Bit  9 Status9
Bit 10 ManSpcO5
Bit 11 ManSpcO6
Bit 12 ManSpcO1
Bit 13 ManSpcO2
Bit 14 ManSpcO3
Bit 15 ManSpcO4
```

- In the chapter "**Configurable response using the Feature I/O**" - "**Calculation of the time stamp**", the **note** relating to `Feature bit5` must be replaced.
  - OLD:
    Changes are made in the **Pcs7DiIT** *block and prevent invalid error messages regarding time stamp.*
  - NEW:
    With `Feature.BitB5 = 0` *the* `TimeStampOn` *input at the connected* **EventTS** *instance must be configured with "1" (use I/O time stamp).*

- The following sections must be added or replaced in the "**Time stamp**" section:
  "**Time stamp**" section:
  The two blocks **EventTS** and **Event16TS** can be used for processing of time stamped signals. If the **EventTS** block is mentioned in this chapter, the **Event16TS** block is always meant accordingly.
  "**Forming the time information**" section: The section must be replaced as follows:
  - OLD:
    The time information is generated by one of the following methods and is specified at the block by means of the input parameter TimeStampOn:
    `TimeStampOn = 0`: Use time stamp of the CPU (default)
    `TimeStampOn = 1`: Use time stamp of the I/O.
  - NEW:
    The time information is generated in one of the following two ways and is specified at the EventTS block via the `TimeStampOn` input parameter for **all** channels of an **EventTS** instance:
    `TimeStampOn = 0`: Use time stamp of the CPU (default)
    `TimeStampOn = 1`: Use time stamp of the I/O.

  Time stamping in the EventTS block - Use time stamp of the CPU
  Connect the binary output parameter of another block (e.g. **Pcs7DiIn**) to an `Inx` message input (x = 1 ... 8) of the **EventTS**, (**Event16TS**, x = 1 ... 16) block.
  The message time stamp is formed in the EventTS from the CPU time. If this mode is to be selected for a signal, you need to disable time stamping for the corresponding channel in the HW configuration. If you use the **Pcs7DiIT** block instead of the **Pcs7DiIn** channel block for signal acquisition in your user program and if you have connected this block to one of the `Inx` inputs with the EventTS, you also need to ensure that `Feature.Bit5` at **Pcs7DiIT** is configured with "1" (CPU time stamp).

  If the **EventTS** block detects a change of the signal state at a signal input, it reads the CPU time and uses this as time stamp. Only the signal changes that are slower than the cycle time of the block can be detected.
  High-precision time stamp in the process I/O - Use time stamps of the I/O
  You have configured the hardware of your system for high-precision time stamping as explained in the "**PCS 7 - High-Precision Time Stamping Function Manual**". Signal changes are recognized in the I/O and provided with the time stamp of the I/O. This data is available at the output parameter `TS_Out` of the **Pcs7DiIT** block. To generate a time-stamped message, connect the `TS_Out` output parameter of **Pcs7DiIT** to an `InTSx` (x = 1 ... 8) message input of the **EventTS**, (**Event16TS**, x = 1 ... 16) block. Ensure that `Feature.Bit5=0` (I/O time stamp) is at the **Pcs7DiIT** block.

  The high-precision time stamp is independent of the cycle time of the blocks. The actual time resolution for two different status changes depends on your plant configuration and the hardware you are using.

- In the chapter "**Functions of VlvPosL**", the following note must be inserted in the section "**Motor valve control without position control**":
  After a stop control, the manipulated variable is tracked for the position feedback (`MV_Out = RbkOut`) for the time from `V_MonTiStatic`. The manipulated variable is then held and can be monitored to ensure that the position is maintained.

Change history of PCS 7 Advanced Process Library 7

Changes since delivery release PCS 7 Advanced Process Library Readme (online)

Version	Edition	Change
2021-01-21 (ONLINE)	01/2021	Delivery status PCS 7 Advanced Process Library V9.1