

Application Note

ServoOne / junior / Safety / LeviOne / C-Line PN/DP with STEP 7

Example DP1

Document-No.: APN1126-EN
Language: English
Author: Dipl.-Ing.(FH) Tobias Rühl
Company: KEBA Industrial Automation Germany GmbH
Created: 2017-03-13
Last Change: 2022-01-14
No. of Pages: 21

Table of Contents

1.	Introduction	5
1.1.	Operation Modes STD vs. PPO (ServoOne)	5
1.2.	Easy Drive Operation Modes (C-Line)	6
1.3.	Acyclic Communication via PKW-Channel (ServoOne and C-Line)	6
1.4.	Mapping (ServoOne)	7
1.4.1.	Needed Mapping (ServoOne) for this example	7
2.	KEBA Example Projects	11
2.1.	Hardware Configuration	11
2.2.	Watch List	11
3.	KEBA Function Blocks	12
3.1.	Function Block Access	12
3.2.	Cyclic communication	13
3.3.	SO_ReadPKW	14
3.4.	SO_WritePKW	15
3.5.	SO_Power	16
3.6.	TabSetHandling	17
3.7.	SO_ReadDPV1	20
3.8.	SO_WriteDPV1	21

Document History

Change date	Changed by	Comment
2016-18-11	Dipl.-Ing.(FH) Tobias Rühl	First version 1.0
2017-03-14	Dipl.-Ing.(FH) Tobias Rühl	Prepared for release with new CI/CD layout
2017-09-20	Dipl.-Ing.(FH) Tobias Rühl	Version 1.1 Update for C-Line Drives
2020-03-17	Dipl.-Ing.(FH) Tobias Rühl	Version 1.2, DPV Read/Write added, layout converted to KEBA CI/CD
2022-01-04	Dipl.-Ing.(FH) Tobias Rühl	Version 1.3, Added function for drive positioning and table positioning

Related Files

Change date	Filename	Comment
2022-01-04	KEBA_Example_SO_LEV_DP1_2_2_A.zip	Version 2.2 ServoOne device family (Project A), STEP7 V5.6
2017-09-20	KEBA_Example_CLine_DP1_V1_0_B.zip	Version 1.0 C-Line device family (Project B), STEP7 V5.6

Disclaimer

The recommendations for action on which these application notes are based were produced in the scope of tests completed under the environmental conditions mentioned in the operating instructions. The user is responsible for complying with and reviewing these environmental conditions when applying the recommendations.

These application notes are intended for qualified specialist personnel who commission and maintain drive and automation components. According to IEC 60364 or CENELC HD 384, qualified personnel refers to people who hold an appropriate qualification, who know how to install, assemble, commission and operate the devices, and who are aware of all accident prevention regulations, guidelines and laws applicable at the place of operation.

The references and recommendations for action contained in this application notes only apply to the following products:

- ServoOne
- ServoOne junior
- ServoOne Safety
- C-Line / CM DP-V1
- SIEMENS S7-300 with STEP7 5.6

Please also note that the device documentation of the respective devices also includes safety information. Read these, and ensure that the safety information in the specific device documentation is taken into consideration.

The screenshots included in these application notes are only examples, which are intended to illustrate the individual steps.

We also draw your attention to the fact that the software used in our products may include software that is licensed as Open Source Software (OSS) or Free Software (FOSS). The license conditions of the OSS and/or FOSS included/used in our DriveManager 5.x service tool can be found in the tool itself.

All details are subject to change. Any liability for the correctness of this information is excluded.

1. Introduction

The delivered examples are designed for SIEMENS STEP7 tool. With the example project you are able to do first steps like switch on drive, read/write via PKW channel and cyclic communication. The examples are runnable with Servo One and C-Line device family.

Tab. 1: system test environment

Component	Description
PLC	SIEMENS S7-300 CPU315-2PN/DP FW-V2.6
Slave	ServoOne FS, FW 4.30-18 LeviOne 51LJ008 FW 31.10-91 CM-DPV1 Module V3.0 (FW 2.15) with CDE32.004 FW 5.10-06
Service Tool	KeStudio Drive Manager 5.18.0
GSD-File	LM010A33.GSD (ServoOne) (PROFIBUS) LU030564.GSD (DPV1 C-Line) (PROFIBUS)
GSDML-File	Latest GSDML file for ServoOne
STEP7-Portal	V5.6 + SP1

1.1. Operation Modes STD vs. PPO (ServoOne)

It doesn't matter if you are using PROFIBUS or PROFINET. Depending on your application you have to define your operation mode. The ServoOne acts as I/O slave device and delivers the following modes:

1. Position Mode (PCON)

- Point-to-Point movements
- Target Velocity (endless positioning)
- Traversing blocks
- Jog-Mode (endless positioning)

2. Velocity Mode (SCON)

- Target velocity
- Jog-Mode

3. V/f Mode (VFCON)

- Target velocity

At ServoOne, the modes 1 - 3 are done by the internal profile generator (PG) in the drive. For all modes, you will need a process data mapping for a cyclic communication PLC ↔ Drive. For simple applications using mode 1 ... 3. The PROFIDrive Profile defines so called standard telegrams (**STD**). With it, the mapping is fixed. If you need a user defined mapping, you have to use the parameter process object (**PPO**). The following standard telegrams are supported by ServoOne drives: **STD1**: Velocity Control (VCON) / **STD7**: Traversing Blocks (PCON) / **STD8**: Interpolated Positioning (PCON IRT) / **STD9**: Positioning/Velocity (endless positioning). For detailed information please have a look to the latest PROFIBUS/PROFINET user manual.

1.2. Easy Drive Operation Modes (C-Line)

The C-Line drive doesn't support Standard-Telegrams and PPO-Telegrams. There are several modes offered for different application demands (velocity, positioning,...). For C-line Drives the Easy Drive Modes have to be used with the communication module. They are similar to standard telegrams because the mapping is fixed, too. For the description of each mode please have a look to the operation manual of the communication module.

1.3. Acyclic Communication via PKW-Channel (ServoOne and C-Line)

ServoOne: Some PPO telegrams deliver an additional PKW-Channel (Parameter Configuration Channel). This channel/service is used to set or read parameters in drives. The communication is acyclic, means that the processing time varies (no real time). It is a confirmed service. When you want to use the PKW-Channel beside a cyclic communication, you have to map the right telegram, which supports both:

Slot	Module	Order number	I Address	Q address	Diagnostic Address
0	DRIVE	1xxx.xx7x.xxxx.x			2043*
X1	Interface				2042*
P1 B	Port 1 (x47)				2041*
P2 B	Port 2 (x48)				2040*
1	Profil PPO 5, PKW + 20~				2039*
1.1	Parameter Access Point				2039*
1.2	Profil PPO 5, PKW + 20~		256...263	256...263	

Fig. 1.1: User defined Telegram with PKW channel (PPO5) for ServoOne with PROFINET

For an acyclic communication via DP-V1 service, the telegram type doesn't matter anymore. DP-V1 service is only supported by ServoOne.

C-Line: The acyclic communication channel PKW is always optional for C-Line. If you will need this, you can add a PKW-Slot in the Hardware-Configuration (8 Byte). The I/O address will automatically assigned by the tool:

Slot	DP ID	...	Order Number / Designation	I Address	Q Address	Comment
0	4AX		PKW Parameterdaten	256...263	256...263	
1	193		PZD EasyDrive Basic	264...271	264...271	

Fig. 1.2: Additional PKW Slot in the hardware configuration (PROFIBUS)

The PKW channel has the same telegram structure like for ServoOne device family. The only difference is the offset = 1 for Sub-Index value. For detailed information please have a look to the CM-DPV1 manual.



Fig. 1.2: Structure of PKW telegram for C-Line, LeviOne and ServoOne

1.4. Mapping (ServoOne)

When using the standard telegrams, the master will write down the telegram selection value in P922[0]. Depending on that value, the drive will map the necessary process data automatically in P195/P916. You can't change this mapping.

When you are using a user defined mapping, the master will write down a value > 100 in P922[0] (depending on the telegram type). With it, you have to map your parameters in P915/P916 manually (and save in the device). P915 means "write from PLC to drive" and P916 means "read from drive to PLC". Then, You can find the same sequence in you PLC data input.

1.4.1. Needed Mapping (ServoOne) for this example


You can find the map able parameters in the signal list P1284 and P1285. The parameters P967 (*controlword*) and P968 (*statusword*) are always necessary. In Tab. 2 you can find an example for a user defined mapping using a PPO telegram. With this mapping table you can estimate the minimum amount of process data read/write. Depending on this amount, you have to select the PPO-telegram type (PPO1 – PPO5). If you need an additional acyclic transfer, you can choose a PPO telegram (see *manual*) with PKW channel (parameter configuration channel).


Tab. 3: User defined mapping PPO5 for this example

P915 with PCON			P916 with PCON		
Object	Parameter	Datatype	Object	Parameter	Datatype
Controlword	P967[0]	uint16	Statusword	P968	uint16
Target Position [user units]	P1275[0]	int32	Act. Position [user units]	P1276	int32
Target Velocity [user units]	P1277[0]	Int32	Act. Velocity [user units]	P1271	Int32
Acceleration	P1278[0]	uint16			

Deceleration	P1279[0]	uint16		
Σ [Bit]		112	Σ [Bit]	80
Σ [Words]		7	Σ [Words]	5
Σ [Bytes]		14	Σ [Bytes]	10

Tab. 2: User Defined Mapping (example)

	Id	Sub id	Name	Value
	915		COM_DP_PZDSelecti...	
.....	915 0		COM_DP_PZDSelecti...	967
.....	915 1		COM_DP_PZDSelecti...	1275
.....	915 2		COM_DP_PZDSelecti...	1275
.....	915 3		COM_DP_PZDSelecti...	1277
.....	915 4		COM_DP_PZDSelecti...	1277
.....	915 5		COM_DP_PZDSelecti...	1278
.....	915 6		COM_DP_PZDSelecti...	1279

	Id	Sub id	Name	Value
	916		COM_DP_PZDSelecti...	
.....	916 0		COM_DP_PZDSelecti...	968
.....	916 1		COM_DP_PZDSelecti...	1276
.....	916 2		COM_DP_PZDSelecti...	1276
.....	916 3		COM_DP_PZDSelecti...	1271
.....	916 4		COM_DP_PZDSelecti...	1271

For this example mapping in Tab. 3 you can use PPO5 telegram in your hardware configuration (20 Byte I/O + PKW channel). **The control mode PCON for Positioning has to be set manually in the DriveManager or acyclic service.**

You have to import the GSD/GSDML device file into the STEP7 tool. After that, the ServoOne/LeviOne/C-Line is imported in the hardware catalogue with all supported telegrams. STEP7 will handle the ServoOne/LeviOne/C-Line as external periphery so STEP7 will generate a start/end address, when the periphery will be used (*Figure 2*).

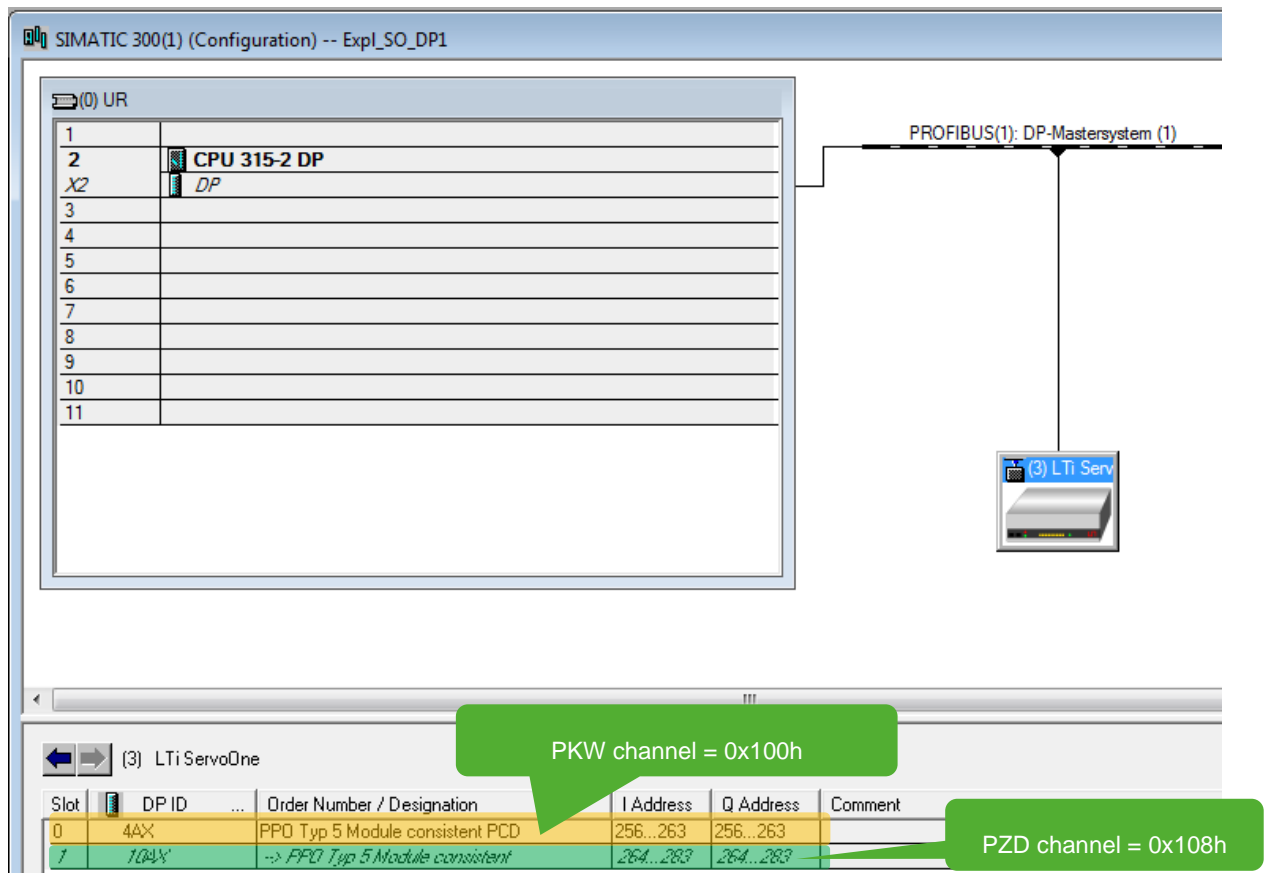
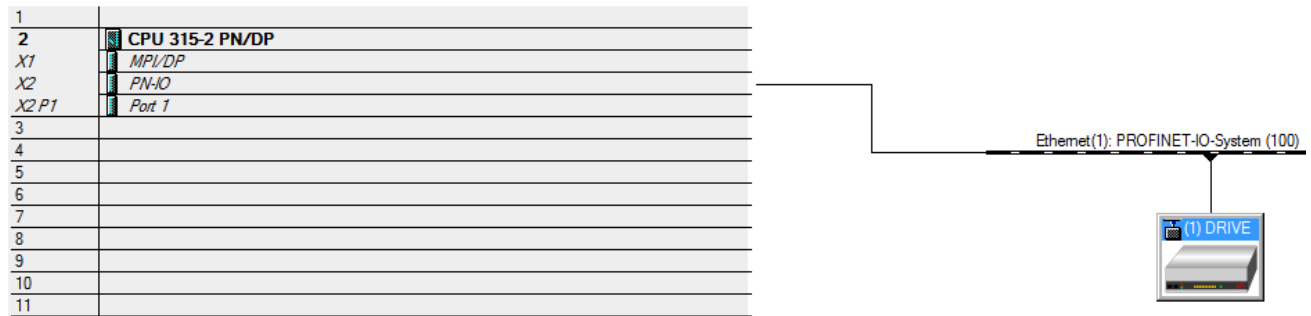


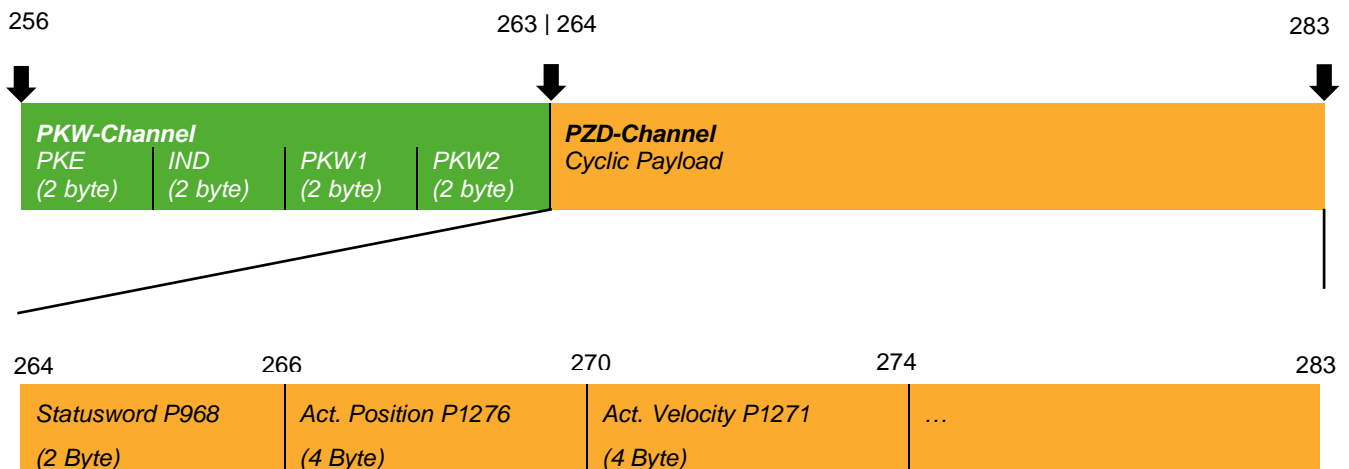
Figure 2.1: Hardware configuration ServoOne with start/end address (PROFIBUS)



(1) DRIVE						
Slot	Module	Order number	I Address	Q address	Diagnostic Address	Comment
0	DRIVE	1 XXXX.XXX.XXX.XXX.X			2043*	
X1	Interface				2042**	
P1 R	Port 1 (X47)				2041**	
P2 R	Port 2 (X48)				2040**	
1	Profil PPO 5, PKW + 20**				2039*	
1.1	Parameter Access Point				2038**	
1.2	Profil PPO 5, PKW + 20**		256...263	256...263		

Figure 2.2: Hardware configuration ServoOne with start/end address (PROFINET)

You can find your process data in the input/output process image of the PLC. The process data starts at the configured address and have the same order like in the ServoOne mapping. The SIEMENS S7 PLC handles the data with Little Endian bit order.



264	266	270	274	276	283
<i>Statusword P967</i> (2 Byte)	<i>Target Position P1275</i> (4 Byte)	<i>Target Velocity P1277</i> (4 Bytes)	<i>Accel. P1278</i> (2 Byte)	<i>Decel. P1279</i> (2 Byte)	

Process image PLC output

2. KEBA Example Projects

There is one example project for C-Line PROFIBUS and ServoOne PROFINET. The programs and function blocks are programmed in language IL/AWL and FUP.

Tab. 3: Overview of implemented programs

Category	Description
PKW Read/Write	The function blocks <i>SO_ReadPKW</i> and <i>SO_WritePKW</i> read and write parameter values via acyclic channel (ServoOne, LeviOne and C-Line)
PowerOn	Project A: The function <i>SO_Power</i> offers a drive handling for positioning with device feedback. (Only ServoOne)
ReadDPV1, WriteDPV1	The function blocks <i>SO_ReadDPV1</i> and <i>SO_WriteDPV1</i> read and write parameter values via acyclic DP-V1 service (Records) (ServoOne, LeviOne)
Main	Main program includes program calls and handles cyclic process data

2.1. Hardware Configuration

ServoOne (Project A): In example project A the PPO5 telegram with PKW-channel + 20 Byte I/O is configured because of the required amount of process data in Tab. 2.

C-Line (Project B): In example project B the EasyDrive Basic Mode / Telegram (8 Byte) with optional PKW channel (8 Byte) is configured.

2.2. Watch List

The example project contains a variable list to show actual values of program and function blocks. The variable list "VAT1/Parameter" shows the actual parameters of all implemented functions. You can force some values for testing.

Parameter -- @APN1126_SO_PN\SIMATIC 300\CPU 315-2 PN\DP\S7-Programm(11) ONLINE					
	Address	Symbol	Display format	Status value	Modify value
3	DB3.DBW 2	"PKWParameter".Read ParaID	DEC	700	700
4	DB3.DBB 4	"PKWParameter".Read SubID	DEC	0	0
5	DB3.DBD 6	"PKWParameter".Read Value	FLOATING_POINT	0.0	
6	DB3.DBX 10.0	"PKWParameter".Read Error	BOOL	false	
7	DB3.DBW 12	"PKWParameter".Read ErrorID	DEC	0	
8					
9	//2: PKW Write				
10	DB3.DBX 14.0	"PKWParameter".Write Enable	BOOL	false	false
11	DB3.DBW 16	"PKWParameter".Write ParaID	DEC	1270	1270
12	DB3.DBB 18	"PKWParameter".Write SubID	DEC	0	0
13	DB3.DBD 20	"PKWParameter".Write Value	DEC	L#400	L#400
14	DB3.DBX 24.0	"PKWParameter".Write Error	BOOL	false	
15	DB3.DBW 26	"PKWParameter".Write ErrorID	DEC	0	
16					
17	//3: SO_Power				
18	DB200.DBX 0.0	"Handle".PwrOn	BOOL	false	
19	DB200.DBX 0.1	"Handle".Reset	BOOL	false	
20	DB200.DBX 0.2	"Handle".Jog1	BOOL	false	
21	DB200.DBX 0.3	"Handle".RunHome	BOOL	false	
22	DB200.DBX 0.4	"Handle".RunAbsPos	BOOL	false	
23	DB200.DBX 0.5	"Handle".Quickstop	BOOL	false	
24	DB200.DBX 0.6	"Handle".PosVelocityMode	BOOL	false	
25	DB200.DBX 0.7	"Handle".PwrStatus	BOOL	false	
26	DB200.DBX 1.0	"Handle".DriveError	BOOL	false	
27	DB200.DBX 1.1	"Handle".HomeDone	BOOL	false	
28	DB200.DBX 1.2	"Handle".Standstill	BOOL	true	
29	DB200.DBX 1.3	"Handle".TargetPosReached	BOOL	true	
30	DB200.DBX 1.4	"Handle".NewTargetAck	BOOL	false	
31	DB108.DBX 0	"TabSetHandle".Index	DEC	2	2
32	DB108.DBX 14.0	"TabSetHandle".Next	BOOL	true	
33					
34	//Set Positions from table DB				

Figure 3.1: Watch List Project A (ServoOne)

Parameter -- @Expl_SO_DP1\SIMATIC 300(I)\CPU 315-2 DP\Programs ONLINE					
	Address	Symbol	Display format	Status value	Modify value
1	//1: PKW Read				
2	DB3.DBX 0.0	"PKWParameter".Read.Enable	BOOL	true	true
3	DB3.DBW 2	"PKWParameter".Read ParaID	DEC	405	405
4	DB3.DBB 4	"PKWParameter".Read SubID	DEC	-1	-1
5	DB3.DBD 6	"PKWParameter".Read Value	FLOATING_P...	315.5	
6	DB3.DBX 10.0	"PKWParameter".Read Error	BOOL	false	
7	DB3.DBW 12	"PKWParameter".Read ErrorID	DEC	0	
8					
9	//2: PKW Write				
10	DB3.DBX 14.0	"PKWParameter".Write.Enable	BOOL	false	false
11	DB3.DBW 16	"PKWParameter".Write ParaID	DEC	1270	1270
12	DB3.DBB 18	"PKWParameter".Write.SubID	DEC	0	0
13	DB3.DBD 20	"PKWParameter".Write.Value	DEC	L#400	L#400
14	DB3.DBX 24.0	"PKWParameter".Write.Error	BOOL	false	
15	DB3.DBW 26	"PKWParameter".Write.ErrorID	DEC	0	
16					
17	DB1.DBW 0	"DPInput".Data.STW1	HEX	W#16#0C20	
18	DB2.DBW 0	"DPOutput".Data.CTR1	HEX	W#16#0000	
19					
20	//Function Selection: 1 = Read, 2 = Write, 3 = SwitchOnServoOne				
21	MW 10		DEC	1	1
22					

Figure 3.2: Watch List Project B (C-Line)

3. KEBA Function Blocks

Each instance needs a data block. Data blocks which are named with numbers >100 are instance data blocks.

The function blocks are only examples to show how it can be handled. KEBA recommend to build up your own project with your own solution without adopt this examples.

3.1. Function Block Access

If you want to use more function blocks in a program which are modifying the output process data (controlword, target values, etc. ...) you have to handle the access of this function blocks. Otherwise, each function block will modify the same output data in the same time. In this example project(s) you have to handle the access manually through a selector (compare function).

With the parameter "Selector/MW 10" in the variable list "Parameter" you can handle the function block access.

- Selector = 1: Read via PKW channel (for ServoOne, LeviOne, C-Line)
- Selector = 2: Write via PKW channel (for ServoOne, LeviOne, C-Line)
- Selector = 3: Drive Handling for power and movement (for ServoOne)
- ~~Selector = 4: Read PKW String (for ServoOne, LeviOne)~~ → under construction
- Selector = 5: Read DPV1 (for ServoOne, LeviOne)
- Selector = 6: Write DPV1 (for ServoOne, LeviOne)

3.2. Cyclic communication

In this example, the system functions *SFC14* "DPRD_DAT" and *SFC15* "DPWR_DAT" handle the cyclic input/output data for PROFIBUS and/or PROFINET. The incoming process data will be copied to the data block *DB1* "DPInput", the outgoing process data will be taken from the data block *DB2* "DPOutput".

@DB1 -- "DPInput" -- APN1126_SO_PN\SIMATIC 300\CPU 315-2 PN/DP\...DB1 ONLINE

Address	Name	Type	W#16#0	W#16#0000	payload
0.0	Data.STW1	WORD	W#16#0		
2.0	Data.ActPos	DWORD	DW#16#		
6.0	Data.ActVelo	DWORD	DW#16#		
10.0	Data.data4	WORD	W#16#0	W#16#0000	payload
12.0	Data.data5	WORD	W#16#0	W#16#0000	payload
14.0	Data.data6	WORD	W#16#0	W#16#0000	payload
16.0	Data.data7	WORD	W#16#0	W#16#0000	payload
18.0	Data.data8	WORD	W#16#0	W#16#0000	payload
20.0	Data.data9	WORD	W#16#0	W#16#0000	payload

Figure 3a: Data Block for PPO5 telegram 20 byte input data

@DB2 -- "DPOutput" -- APN1126_SO_PN\SIMATIC 300\CPU 315-2 PN/DP\...DB2 ONLINE

Address	Name	Type	W#16#0	W#16#0000	payload
0.0	Data.CTRL	WORD	W#16#0		
2.0	Data.TargetPos	DINT	L#0	L#	
6.0	Data.TargetSpeed	DINT	L#0	L#	
10.0	Data.Acceleration	WORD	W#16#0	W#16#0000	Target Deceleration
12.0	Data.Deceleration	WORD	W#16#0	W#16#0000	payload
14.0	Data.data5	WORD	W#16#0	W#16#0000	payload
16.0	Data.data6	WORD	W#16#0	W#16#0000	payload
18.0	Data.data7	WORD	W#16#0	W#16#0000	payload

Figure 3b: Data Block for PPO5 telegram 20 byte output data

Note: The SFC14 / SFC15 can only handle 4 byte of consistent data. For more, you can use the functions several times (see OB1)

3.3. SO_ReadPKW

Overview:

```
CALL "SO_ReadPKW" , DB101
Enable      := "PKWParameter".Read.Enable
ParaID      := "PKWParameter".Read.ParaID
SubID       := "PKWParameter".Read.SubID
PKEin       := "PKWInput".In.PKE
INDin       := "PKWInput".In.IND
DataIn      := "PKWInput".In.Payload
PKEout      := "PKWOutput".Out.PKE
INDout      := "PKWOutput".Out.IND
DataOut     := "PKWOutput".Out.Payload
ValueDWORD := "PKWParameter".Read.Value
Error       := "PKWParameter".Read.Error
ErrorId     := "PKWParameter".Read.ErrorID
```

- ServoOne, LeviOne and C-Line
- Acyclic communication
- Single Parameters or Array-Parameters possible
- It doesn't matter which operation mode is configured.
- A telegram with PKW channel is necessary in the S7 hardware configuration
- Use only one time per drive or add an access control.
- Parameter-ID and Sub-ID are online changeable while FB is enabled.
- You have to convert the output value to your target data type.

Inputs	Datatype	Description
Enable	BOOL	While TRUE, the value will be read out continuously. When FALSE, all outputs are frozen
ParaID	WORD	ServoOne Parameter which have to be read out, Range: 0 ... 4095
SubID	BYTE	Array sub index; the FB automatically compensates the offset (+1 = ServoOne), for C-Line: -1 means 0, Range: -1 ... 254
PKEin	WORD	Input data of identifier PKE (2 bytes)
INDin	WORD	Input data of sub index value IND (2 bytes)
DataIn	DWORD	Input data of payload PKW1+PKW2 (4 bytes)
Outputs	Datatype	Description
PKEout	WORD	Output data of identifier PKE (2 bytes)
INDout	WORD	Output data of sub index value IND (2 bytes)
DataOut	DWORD	Output data of payload PKW1+PKW2 (4 bytes)
ValueDWORD	DWORD	Value read from drive of type DWORD
Error	BOOL	When TRUE an error occurred

ErrorID	STRING	ErrorID by the device, 88 = no response, ErrorID = error list in manual (CM-DPV1 / SO PN-Manual)
---------	--------	--

3.4. SO_WritePKW

Overview:

```
CALL "SO_WritePKW" , DB102
  Enable := "PKWParameter".Write.Enable
  ParaID := "PKWParameter".Write.ParaID
  SubID := "PKWParameter".Write.SubID
  Value := "PKWParameter".Write.Value
  PKEin := "PKWInput".In.PKE
  INDin := "PKWInput".In.IND
  DataIn := "PKWInput".In.Payload
  PKEout := "PKWOutput".Out.PKE
  INDout := "PKWOutput".Out.IND
  DataOut := "PKWOutput".Out.Payload
  Error := "PKWParameter".Write.Error
  ErrorID := "PKWParameter".Write.ErrorID
```

- ServoOne , LeviOne and C-Line
- Acyclic communication
- Single Parameters or Array-Parameters possible
- It doesn't matter which operation mode is configured.
- A telegram with PKW channel is necessary in the S7 hardware configuration
- Use only one time per drive or add an access control.
- Parameter-ID and Sub-ID are online changeable while FB is enabled.
- You have to convert the output value to your target data type.

Inputs	Datatype	Description
Enable	BOOL	While TRUE, the value will be written continuously. When FALSE, all outputs are frozen
ParaID	WORD	ServoOne Parameter which have to be read out, Range: 0 ... 4095
SubID	BYTE	Array sub index; the FB automatically compensates the offset (+1 = ServoOne), for C-Line: -1 means 0, Range: -1 ... 254
Value	DWORD	Value which has to be written to the device
PKEin	WORD	Input data of identifier PKE (2 bytes)
INDin	WORD	Input data of sub index value IND (2 bytes)
DataIn	DWORD	Input data of payload PKW1+PKW2 (4 bytes)
Outputs	Datatype	Description
PKEout	WORD	Output data of identifier PKE (2 bytes)
INDout	WORD	Output data of sub index value IND (2 bytes)

DataOut	DWORD	Output data of payload PKW1+PKW2 (4 bytes)
Error	BOOL	When TRUE an error occurred, Value = 0
ErrorID	STRING	ErrorID by the device, 88 = no response, ErrorID = error list in manual (CM-DPV1 / SO PN-Manual)

3.5. SO_Power

Overview:

```
CALL "SO_Power" , DB103
  Statusword      := "DPInput".Data.STW1
  ErrReset        := "Handle".Reset
  Enable          := "Handle".PwrOn
  Jog1            := "Handle".Jog1
  RunHome         := "Handle".RunHome
  RunAbsPos       := "Handle".RunAbsPos
  Quickstop       := "Handle".Quickstop
  PosVelocityMode := "Handle".PosVelocityMode
  Controlword     := "DPOutput".Data.CTRL
  Status          := "Handle".PwrStatus
  DriveError      := "Handle".DriveError
  HomingDone      := "Handle".HomeDone
  Standstill      := "Handle".Standstill
  TargetPosReached := "Handle".TargetPosReached
  NewTargetAck    := "Handle".NewTargetAck
```

- For ServoOne
- PCON, SCON, VFCON
- Cyclic communication
- Switch on a single axis
- Run homing procedure
- Set new target value (single axis movement with feedback TargetReached)
- Get motion status bits

Inputs	Datatype	Description
Statusword	WORD	Statusword 1 of the drive as input, read by process data input SFC
ErrReset		Drive error reset while value is true (always possible)
Enable	BOOL	When TRUE the FB will run the power-on sequence. When FALSE, the drive will be disabled
Jog1	BOOL	Run drive with given jog speed 1 in P1268 while value is true
RunHome	BOOL	While true, the given homing mode will be active, check for HomingDone! Run homing before positioning (if no absolute encoder)
RunAbsPos	BOOL	Edge sensitive according to P1267, default = rising + falling edge, the drive will run to the given target within given dynamics.
Quickstop	BOOL	While true, drive will run given quick stop ramp immediately (always possible)

PosVelocityMode	BOOL	When true, the drive will switch to endless positioning = velocity mode in PCON. Target Velocity is then the active target value.
Outputs	Datatype	Description
Controlword	WORD	Controlword to the drive, written by process data output SFC
Status	BOOL	When TRUE, the axis is switched on
DriveError	BOOL	When TRUE, the axis has an error
HomingDone	BOOL	When TRUE, the homing sequence is done
Standstill	BOOL	When TRUE, the speed is 0 according to the standstill window in the drive
TargetPosReached	BOOL	When TRUE, the commanded target value was reached
NewTargetAck	BOOL	When state is alternating, the commanded target value was acknowledged by drive

3.6. TabSetHandling

```
CALL "TabSetHandling" , "TabSetHandle"
Index      :=
SetPosition:= "DPOutput".Data.TargetPos
SetVelocity:= "DPOutput".Data.TargetSpeed
SetAcc      := "DPOutput".Data.TargetAcceleration
SetDec      := "DPOutput".Data.TargetDeceleration
Next       :=
```

Overview:

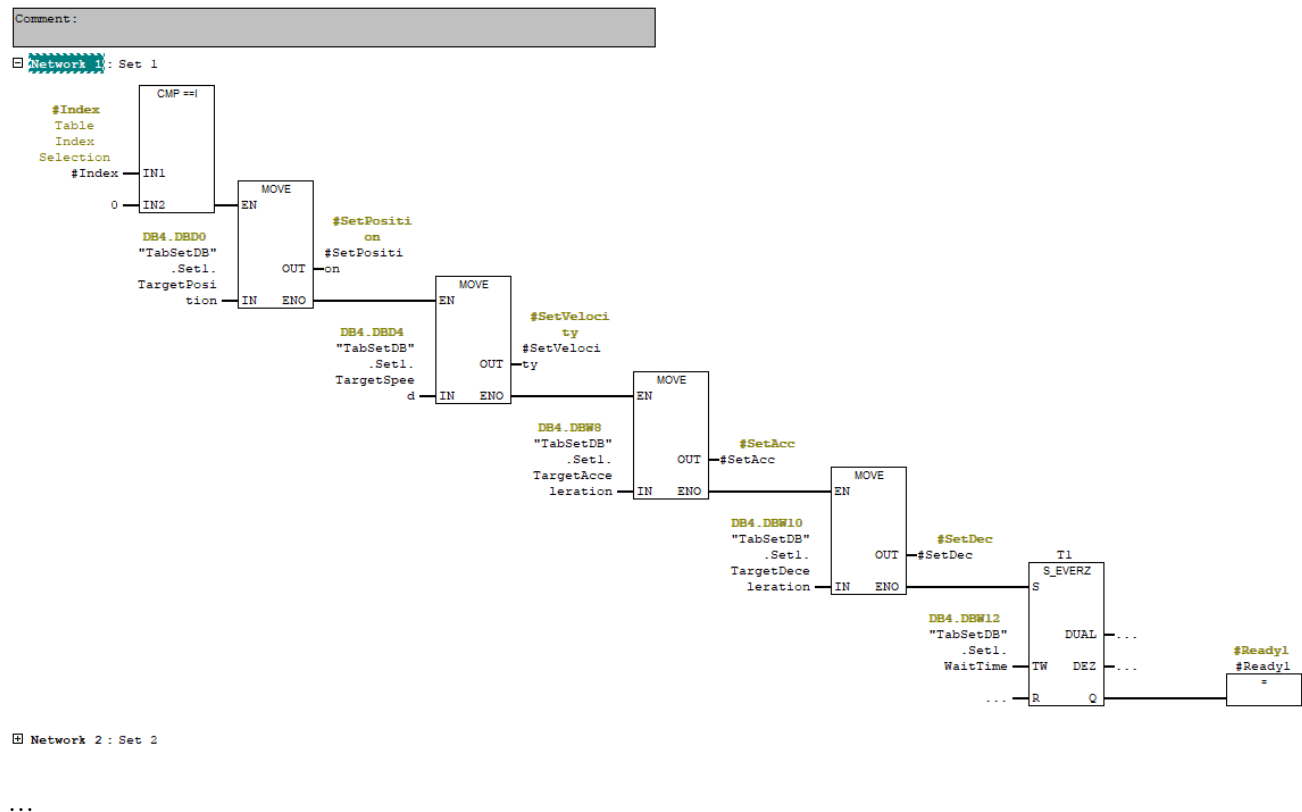
- Universal
- Can be used for a table positioning by PLC
- Values from data block
- 1 entry = 1 network in this function block

Inputs	Datatype	Description
Index	INT	Table Index Selection
Outputs	Datatype	Description
SetPosition	DINT	Resulting position value to drive
SetVelocity	DINT	Resulting velocity value to drive
SetAcc	WORD	Resulting acceleration value to drive
SetDec	WORD	Resulting deceleration value to drive
Next	BOOL	True, when table set wait time has expired

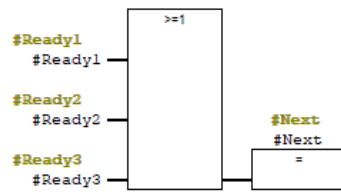
This function block contains several networks. One network for one table entry + one Timer + one *Ready*-Variable and has to be extended to the application demands. The number of entries has to be consistent to the Data Block, where all positioning sets values are contained (here DB4 "TabSetDB")

DB4 -- "TabSetDB" -- APN1126_SO_PN\SIMATIC 300\CPU 315-2 PN\DP\...\DB4

Address	Name	Type	Initial val	Actual value
0.0	Set1.TargetPositi	DINT	L#0	L#11
4.0	Set1.TargetSpeed	DINT	L#0	L#22
8.0	Set1.TargetAcce	WORD	W#16#0	W#16#21
10.0	Set1.TargetDecel	WORD	W#16#0	W#16#2C
12.0	Set1.WaitTime	S5TIME	S5T#0MS	S5T#1S
14.0	Set2.TargetPositi	DINT	L#0	L#55
18.0	Set2.TargetSpeed	DINT	L#0	L#66
22.0	Set2.TargetAcce	WORD	W#16#0	W#16#4D
24.0	Set2.TargetDecel	WORD	W#16#0	W#16#58
26.0	Set2.WaitTime	S5TIME	S5T#0MS	S5T#2S
28.0	Set3.TargetPositi	DINT	L#0	L#9000
32.0	Set3.TargetSpeed	DINT	L#0	L#1000
36.0	Set3.TargetAcce	WORD	W#16#0	W#16#1F4
38.0	Set3.TargetDecel	WORD	W#16#0	W#16#258
40.0	Set3.WaitTime	S5TIME	S5T#0MS	S5T#1S



□ Network 4 : Title:



3.7. SO_ReadDPV1

```
CALL "SO_ReadDPV1" , DB105
Enable      := "Handle".EnReadDPV1
ParaId      := "Handle".ParaID
SubId       := "Handle".SubID
StartAdr    := "Handle".SlaveAdr
Datatype    := "Handle".DPV1RWDatatype
WRREC_Status := "Handle".WRRECStatus
RDREC_Status := "Handle".RDRECStatus
Busy        := "Handle".Busy
Done        := "Handle".Done
Error       := "Handle".Error
ReadDWORD   := "Handle".DPV1ReadDWORD
ReadREAL    := "Handle".DPV1ReadREAL
ReadBYTE    := "Handle".DPV1ReadBYTE
ReadWORD    := "Handle".DPV1ReadWORD
```

Overview:

- ServoOne / LeviOne
- Acyclic communication
- Reading structured parameter types in the device not possible (e.g. P33[SubIdx])
- **Only possible datatypes: WORD, DWORD, REAL, BYTE, in case of other datatypes (e.g. UINT) note the possible value range of the target parameter (overflow)**

Inputs	Datatype	Description
Enable	BOOL	When TRUE the FB will read the parameter acyclic via DPV1, but is called cyclically. A single call has to be managed by the user.
ParaId	WORD	Parameter-ID (from DriveManager)
SubId	BYTE	Parameter-SubID (from DriveManager)
StartAdr	DWORD	Logical start address of the slave (e.g. W#16#108, see Fig. 2.2)
Datatype	BYTE	Target data type has to be set before FB Enable = TRUE!, 0 = WORD, 1 = DWORD, 2 = REAL, 3 = BYTE
Outputs	Datatype	Description
WRREC_Status	DWORD	Status of SFB53
RDREC_Status	DWORD	Status of SFB52
Busy	BOOL	While TRUE, the transmission process is active
Done	BOOL	When TRUE, the transmission process is done
Error	BOOL	When TRUE, the transmission process has an error, see WR/RDREC_Status
ReadDWORD	DWORD	Read value of type DWORD
ReadREAL	REAL	Read value of type REAL
ReadBYTE	BYTE	Read value of type BYTE
ReadWORD	Word	Read value of type WORD

3.8. SO_WriteDPV1

```
CALL "SO_WriteDPV1" , DB107
Enable      := "Handle".ENWriteDPV1
ParaId      := "Handle".ParaID
SubId       := "Handle".SubID
StartAdr    := "Handle".SlaveAdr
WriteDWORD  := "Handle".DPV1WriteDWORD
WriteREAL   := "Handle".DPV1WriteREAL
Datatype    := "Handle".DPV1RWDatatype
WRREC_Status := "Handle".WRRECStatus
RDREC_Status := "Handle".RDRECStatus
Busy        := "Handle".Busy
Done        := "Handle".Done
Error       := "Handle".Error
```

Overview:

- ServoOne / LeviOne
- Acyclic communication
- Writing structured parameter types in the device not possible (e.g. P33[SubIdx])
- **Only possible datatypes: WORD, DWORD, REAL, BYTE, in case of other datatypes (e.g. UINT) note the possible value range of the target parameter (overflow)**

Inputs	Datatype	Description
Enable	BOOL	When TRUE the FB will write the parameter acyclic via DPV1, but is called cyclically. A single call has to be managed by the user.
ParaId	WORD	Parameter-ID (from DriveManager)
SubId	BYTE	Parameter-SubID (from DriveManager)
StartAdr	DWORD	Logical start address of the slave (e.g. W#16#108, see Fig. 2.2)
WriteDWORD	DWORD	Value of type DWORD (8 ... 32 Bit), which has to be written to the device, use if data type = 0,1,3
WriteREAL	REAL	Value of type REAL (32 Bit), which has to be written to the device, use if data type = 2
Datatype	BYTE	Target data type has to be set before FB Enable = TRUE!, 0 = WORD, 1 = DWORD, 2 = REAL, 3 = BYTE
Outputs	Datatype	Description
WRREC_Status	DWORD	Status of SFB53
RDREC_Status	DWORD	Status of SFB52
Busy	BOOL	While TRUE, the transmission process is active
Done	BOOL	When TRUE, the transmission process is done
Error	BOOL	When TRUE, the transmission process has an error, see WR/RDREC_Status