ABB Industry Oy
P.O.Box 184
FIN–00381 HELSINKI
Finland

APPLICATION NOTE

Handled by	Date	Pages
Tommi Saarinen Tel. +358 10 22 22231	11 November 1999	1 (5)

APPLICATION NOTE

1 PPO Types in the Profibus

Five PPO Types are defined in Profibus DP. All types are supported by the NPBA-02 Profibus adapter module. Which PPO Type is required, is determined from the need of the parameter read/write service and from the length of the required process data frame.

	Parameter Identification				Process Data									
	ID	IND	VALUE		CW SW	REF1 ACT1	REF2 ACT2	PD1 PD1	PD2 PD2	PD3 PD3	PD4 PD4	PD5 PD5	PD6 PD6	PD7 PD7
PPO Type 1														
PPO Type 2														
PPO Type 3														
PPO Type 4														
PPO Type 5														

In the drive the process data appears as data sets. The following example is valid for the standard ACS600.

Data sets to the drive	CW	REF1	REF2	PD1	PD2	PD3
	Data set 1			Data set 3		
	DW 1.1	DW 1.2	DW 1.3	DW3.1	DW 3.2	DW3.3

Data sets from the drive	SW	ACT1	ACT2	PD1	PD2	PD3
	Data set 2			Data set 4		
	DW 2.1	DW 2.2	DW 2.3	DW4.1	DW 4.2	DW4.3

The process data can be configured on the drive with the parameter groups 90 D SET REC ADDR and 92 D SET TR ADDR.

2 The configuration on the PLC

NOTE: In this example a SIEMENS S7 PLC is used as master.

When a PPO Type is selected, the corresponding periphery area is reserved for the slave. For example if PPO Type 2 is chosen, the reserved area is 20 bytes long. Eight bytes for the parameter identification and 12 bytes for the process data.

When a PPO Type, that supports parameter services, is selected the reserved periphery area is divided into two modules. One module for the parameter identification and one for the process data area. In the S7 hardware configuration these modules are named as universal modules and have to be handled separately.

The peripherydata can be handled in the S7 in two different ways. In the Simatic S7 PLC the Profibus data can be handled as consistent data with the standard system functions SFC14 and SFC15. These SFC's are described in the S7 manuals. The SFC's are required when the handled data is longer than four bytes. Up to four bytes of data is handled with the load instruction.

When PPO Types 1 or 3 are used, the process data can and also must be accessed with the load and transfer instructions. This is because the process data module is only 4 bytes long.

```
L    MW 10       // Memory word for the control word
T    PAW 264     // Periphery output word for the control word,
                 // i.e. first word of the process data output module

L    MW 12       // Memory word for the reference
T    PAW 266     // Periphery output word for the reference,
                 // i.e. second word of the process data output module

L    PEW 264     // Periphery input word for the status word,
                 // i.e. first word of the process data input module
T    MW 20       // Memory word for the status word

L    PEW 266     // Periphery input word for the actual value,
                 // i.e. second word of the process data input module
T    MW 22       // Memory word for the actual value
```

NOTE: *The periphery addresses and the memory words are only examples. The corresponding periphery addresses must be checked from the hardware configuration window in the S7.*

With the other PPO Types the process data modules are longer and SFC's must be used. For example PPO Type 2.

```
CALL  "DPWR_DAT"                          // Calls the SFC 15, write consistent data to the slave.

    LADDR  :=W#16#108                     // First word address of the process data output
                                          // module 108 Hex = 264 Dec.

    RECORD :=P#DB10.DBX 80.0 BYTE 12      // Pointer to the first bit of the data that is sent
                                          // Also the length of the data frame has to be
                                          // specified.

    RET_VAL:=DB10.DBW2                    // Return value from the SFC


CALL  "DPRD_DAT"                          // Calls the SFC 14, read consistent data from the slave.

    LADDR  :=W#16#108                     // First word address of the process data input
                                          // module 108 Hex = 264 Dec.

    RET_VAL:=DB10.DBW4                    // Return value from the SFC

    RECORD :=P#DB10.DBX 150.0 BYTE 12     // Pointer to the first bit of the data area, where
                                          // the received data is stored. Also the length of
                                          // the data frame has to be specified.
```

It is important, that the data frame has the same length as the module. That means 20 bytes when PPO Type 5 is used and 12 bytes when PPO Type 2 or 4 is used.

The use of the Load and Transfer instructions are also possible with these PPO Types, but then the default configuration on the Simatic has to be changed in the *Hardware Configuration* window. By clicking the corresponding DP-Slave, the universal modules are shown with the address areas on the same window. In this case, one universal module in slot 0 and one in slot 1. Because the process data area is in the module in the slot 1, that has to be handled. By double clicking this module, the *DP Slave Properties* window will appear. In this window, the consistency can be selected. If the consistency is chosen to be via unit the Load and Transfer instructions can be used instead of SFCs.

When accessing the parameter identification area, that means the universal module in the slot 0, the SFC's must always be used and the consistency must be via total length.

NOTE: *When drive generic GSD file (abb_drv.gsd) is used there are three extra options in the window where the PPO Type is selected. When S7 is used as a master system, there is no need to select any of these three options.*

3 Parameter handling

When PPO Types 1, 2 or 5 are used, the drive parameters can be handled through the Profibus. As mentioned earlier, the parameter handling is done in the first universal module in slot 0 and the periphery area of this module has to be configured as data consistent area via total length. Therefore, the parameter services must always be handled with SFC14 and SFC15.

EXAMPLE 1

Writing a drive parameter *12.02 CONST SPEED 1* in *ACS600 Standard Software.*

Profibus parameter number for this parameter is 152 and it can be found from the drive manual, *APPENDIX A – Complete Parameter Settings*.

```
L     W#16#2098          // Write_request label '2' + Parameter number '98' (152 Dec)
T     MW 10

L     0
T     MW 12
T     MW 14

L     W#16#64            // Parameter value '64' (100 Dec)
T     MW 16

CALL  "DPWR_DAT"                 // Calls the SFC 15, write consistent data to the slave.

   LADDR  :=W#16#100             // First word address of the process data output
                                 // module 100 Hex = 256 Dec.

   RECORD :=P#M10.0 BYTE 8       // Pointer to the first bit of the data that is sent
                                 // Also the length of the data frame has to be
                                 // specified.

   RET_VAL:=MW 18                // Return value from the SFC
```

EXAMPLE 2

Reading an actual value *1.06 POWER* from *ACS600 Standard Software.*

Profibus parameter number for this parameter is 6 and it can be found from the drive manual, *APPENDIX A – Complete Parameter Settings*.

```
L       W#16#1006           // Read_request label '1' + Parameter number '6'
T       MW 20

L       0
T       MW 22
T       MW 24
T       MW 26

// First the read request is sent to the slave

CALL  "DPWR_DAT"                    // Calls the SFC 15, write consistent data to the slave.

    LADDR  :=W#16#100               // First word address of the process data output
                                    // module 100 Hex = 256 Dec.

    RECORD :=P#M20.0 BYTE 8         // Pointer to the first bit of the data that is sent
                                    // Also the length of the data frame has to be
                                    // specified.

    RET_VAL:=MW28                   // Return value from the SFC

// The response is read with the SFC 14

CALL  "DPRD_DAT"                    // Calls the SFC 14, read consistent data from the slave.

    LADDR  :=W#16#100               // First word address of the process data input module
                                    // 100 Hex = 256 Dec.

    RET_VAL:=MW38                   // Return value from the SFC

    RECORD :=P#M30.0 BYTE 8         // Pointer to the first bit of the data area, where the
                                    // received data is stored. Also the length of the
                                    // data frame has to be specified.
```

NOTE: *See also the example on the manual of PROFIBUS Adapter Module NPBA-02 (Page 6-14).*