PPI - Driver of PPI Protocol for SIMATIC S7 200 PLCs
User's Manual

Doc. No. ENP4046
Version: 29-08-2005

1. PPI - Driver of PPI Protocol for SIMATIC S7 200 PLCs

1.1. Driver Use

The communication protocol PPI is used for data exchange between computers with the **asix** application and SIEMENS S7 200 PLCs.

1.2. Declaration of Transmission Channel

The syntax of declaration of transmission channel operating according to the PPI protocol has the following form:

logical_name=PPI,*address*,*COMn*

where:

n	- number of the serial port to which the controller is connected;
address	- address of the PLC.

1.3. Driver Configuration

Each defined channel may have its own section, the name of which is its logical name i.e. [logical_name]. In such section, parameters related only to a given station (logical channel) are placed. You should not place there parameters of serial transmission because they are related to a communication port, so to all controllers connected to a given port. The COMn port may have its own section named **[PPI:n]**. Values defined in such section become default values for all stations connected to a given port. In such section you should place parameters of serial transmission if they differ from the default values. If in the application INI file the section named **[PPI]** is placed, then values placed in such section become default values for all communication ports and stations supported by the driver. Values placed in the section of a given station ([logical_name]) have priority over values placed in the section of a given serial port, and the last ones have priority over the values placed in the section [PPI]. If a parameter is not found in any section, then it assumes its default value following the description below. In particular, the INI file may not include any sections parameterizing stations. Appropriate records defining a logical channel are only required in the section **[ASMEN]**.

Parameters of transmission via a serial interface can not to be placed in sections concerning individual stations.

☑ ***Baud =number***

bps=number

Meaning	- transmission speed; it is not placed in the parameter section of the logical channel.
Default value	- 9600.
Parameter:	
number	- number passed in Bd.

parity =parity_parameter

Meaning - determines parity check type; it is not placed in the parameter section of the logical channel.
Default value - e.

Parameter:
parity_parameter - of parity check type:
- n - no parity bit,
- o - odd parity check,
- e - even parity check,
- m - mark
- s - space.

word =number

word_length=number

Meaning - word length; it is not placed in the parameter section of the logical channel.
Default value - 8.

Parameter:
number - number passed from the range of 5 - 8 bits.

stop_bits =number

Meaning - number of stop bits; it is not placed in the parameter section of the logical channel.
Default value - 1.

Parameter:
number - number of bits.

retries =number

retry =number

Meaning - number of transmission repetitions in case of transmission errors.
Default value - 4.

Parameter:
number - number of repetitions.

time_out =number

timeout =number

Meaning - timeout for the station answer.
Default value - 500.

Parameter:
number - time passed in milliseconds.

Delay =number

Meaning - minimal time interval in milliseconds between transmissions of frames.
Default value - 25.

Parameter:
number - time passed in milliseconds.

AllErrors =yes/no

Meaning - if the parameter has a value of no, the information on *timeout* errors will appear in *'Control Panel'* only when the transmission missed in spite of attempts of its repetition. If it has a value of yes, the information on all errors is transmitted to *'Control Panel'*.
Default value - no.

AsComm =number

Meaning - determines whether the driver has to interoperate with the AsComm communication manager; it is not placed in the parameter section of the logical channel.
Default value - no (from the version 1.1).

Send_Frame =number

Meaning - maximal length of a sending frame.
Default value - 117.

Parameter:
number - number passed in bytes from the range of 10-260.

Receive_Frame =number

Meaning - maximal length of a receiving frame.
Default value - 117.

Parameter:
number - number passed in bytes from the range of 10-260.

Variables =number

Meaning - maximal number of variables transferred once.
Default value - 8.

Simulation =yes/no

Meaning - if yes is given, then data reading/writing from/to a PLC will be simulated.
Default value - no.

PCAdres =number

Meaning - address of a computer.
Default value - no.

Parameter:
number - number passed in bytes from the range of 0-255.

EXAMPLE 1

```
[ASMEN]
.....
S7_212=PPI,5,COM2
....
```

In the example above, the station named S7_212 connected to the COM2 port is defined. The communication with the controller is operating on the basis of default parameters.

EXAMPLE 2

```
[ASMEN]
.....
S7_1=PPI,5,COM2
S7_2=PPI,6,COM2
S7_3=PPI,7,COM2
S7_4=PPI,8,COM3
S7_5=PPI,9,COM3
S7_6=PPI,10,COM4
....

[PPI]
;Default values for all stations
baud=9600

[PPI:3]
;Default values for all stations connected to the COM3 port
baud=19200

[S7_6]
delay = 15
```

In the example above stations with names from S7_1 to S7_6 are defined. The stations S7_1, S7_2 and S7_3 are connected to the COM2 port. The stations S7_4 and

S7_5 are connected to the COM3 port. The station S7_6 is connected to the COM4 port. All serial ports except COM3 work with a speed of 9600 baud. The COM3 port works with a speed of 19200 baud. During data exchange with the station S7_6, a delay between transmissions will be reduced to 15 milliseconds.

NOTE
The PPI protocol driver may act together with the AsComm connections manager. In such case, the driver is registered as a client of the AsComm module with a name PPI:n, where n is a number of a serial interface through which the communication with the PLC is executed. In such case the section [PPI:n] may include driver parameters as well as parameters designated for the AsComm module.

1.4. Defining the Process Variables

Measured Data

The driver executes an access to the following variables (see: ***Table 1***).

Table 1. Variables Serviced by the PPI Driver.

Symbol	Data Length
Mn.m	BYTE
MBn	BYTE
MWn	WORD
MDn	DWORD
In.m	BYTE
Ibn	BYTE
Iwn	WORD
Idn	DWORD
Qn.m	BYTE
BN	BYTE
PWN	WORD
QDn	DWORD
Vn.m	BYTE
VBn	BYTE
VWn	WORD
VDn	DWORD
Sn.m	BYTE
SBn	BYTE
SWn	WORD
SDn	DWORD
SMn.m	BYTE
SMBn	BYTE
SMWn	WORD
SMDn	DWORD
AIWn	WORD
AQWn	WORD
HCn	DWORD
Cn	WORD
Cn.m	BYTE
Tn	WORD
Tn.m	BYTE
RUN	BYTE

Meaning of symbols placed in the left column (except RUN) is described in the documentation of S7 controllers.

The variable RUN assumes a value of 1 if the controller is in the state RUN, and of 0 otherwise. Writing to the variable RUN causes an activation of the controller. Writing the value of 0 to the variable RUN causes a transition of the controller to the STOP mode. The change of the controller state is possible only at a suitable setting of switches on the controller.

The variables Cn and Tn enable to access to an actual value of counters and timers. The variables Cn.m and Tn enable access to the state (1 or 0) of counters and timers. The value m. may be any number of the range 0 to 7.

The present driver version does not allow to write to the variables Q, AQW, AIW. Writing to other variables is limited by the controller (Cm.n, Tm.n).

Access to Pseudo-Variables

The PPI protocol driver enables access to pseudo-variables. The access to pseudo-variables does not cause a physical transmission through a serial interface. Values of pseudo-variables are related with an actual state of a connection with the controller.

See ***Table 2***.

Table 2. Pseudo-Variables Serviced by the PPI Driver.

Symbol	Meaning	Length
SBS	number of sent bytes	DWORD
SBR	number of received bytes	DWORD
SFS	number of sent frames	DWORD
SFR	number of received frames	DWORD
SPE	number of parity errors	DWORD
SFE	number of frame errors	DWORD
SOE	number of overrun errors	DWORD
SLE	number of line errors (sum of parity, frame, overrun and other errors)	DWORD
STE	number of timeout errors	DWORD
SPRE	number of protocol errors	DWORD
SCE	number of checksum errors	DWORD
SFC	number of unsuccessful connections (by means of AsComm module)	DWORD
SBC	number of broken connections (established by AsComm module)	DWORD
SLGE	number of logical errors (no data in the controller, faulty address etc.).	DWORD
ERR	sum of all errors (SLE, STE, SPRE, SCE, SFC, SBC and SLGE). Writing of any value to ERR variable causes zeroing of variables SBS, SBR, SFS, SFR, SPE, SFE, SOE, SLE, STE, SPRE, SCE, SFC, SBC and SLGE.	DWORD
TSBS	number of sent bytes (from the beginning of driver operation)	DWORD
TSBR	number of received bytes (from the beginning of driver operation)	DWORD
TSFS	number of sent frames (from the beginning of driver operation)	DWORD
TSFR	number of received frames (from the beginning of driver operation)	DWORD
TSPE	number of parity errors (from the beginning of driver operation)	DWORD
TSFE	number of frame errors (from the beginning of driver operation)	DWORD
TSOE	number of overrun errors (from the beginning of driver operation)	DWORD
TSLE	number of line errors (sum of parity, frame, overrun and other errors) (from the beginning of driver operation)	DWORD
TSTE	number of timeout errors (from the beginning of driver operation)	DWORD
TSPRE	number of protocol errors (from the beginning of driver operation)	DWORD
TSCE	number of checksum errors (from the beginning of driver operation)	DWORD
TSFC	number of unsuccessful connections (by means of AsComm module) (from the beginning of driver operation)	DWORD
TSBC	number of broken connections (established by AsComm module) (from the beginning of driver operation)	DWORD
TSLGE	number of logical errors (no data in the controller, faulty address etc.). (from the beginning of driver operation)	DWORD
TERR	sum of errors determined by variables TSLE, TSTE, TSPRE, TSCE, TSFC, TSBC and TSLGE.	DWORD
ONLINE	assume a value of 1 if the last attempt to send any frame ended successfully (i.e. an acknowledge from the controller was received) and 0 otherwise.	BYTE

2. List of Tables