FB general description

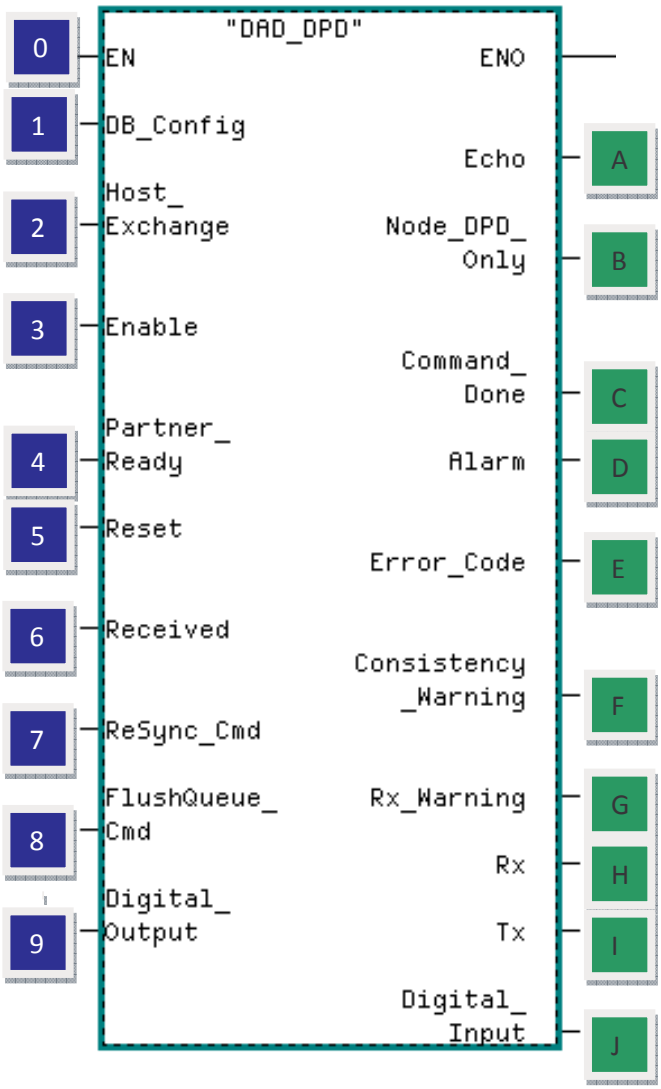
The "DAD_DPD Function Block" implements Datalogic DAD and DPD protocols for the communication between Datalogic identification devices and a Siemens PLC (S7-300/400), using Profibus or Profinet. This block has been developed in order to simplify the development of application software for Siemens STEP7 PLC.

The FB is configurable to use all the DAD-DPD functions; moreover some code has been developed to control the configuration and connections.

Communication between PLC and Datalogic identification devices (partner), made by this FB, uses a buffer (in this example a DB) for receiving messages, and a buffer for the transmission.

When the FB is enabled (input 3 **Enable = TRUE**), if it is not in alarm (output D **Alarm = FALSE**) and the signal "**Partner_ready" = TRUE**, the connection is established and the incoming messages (received from the partner) are stored in the receive buffer; as well as messages stored in the send buffer are sent to the partner up control of the PLC.

Below is the DAD-DPD FB symbol:



Following the operations and the proper use of FB, it specified for each input and output.

FB Input description

0. EN (BOOL)

This command enables the execution of the block. For proper operation, you should not stop this power line, because the block performs functions (reset) even when its enabling control (input 3 Enable) is FALSE. Stopping this line can cause incorrect behaviour of the function block on its activation.

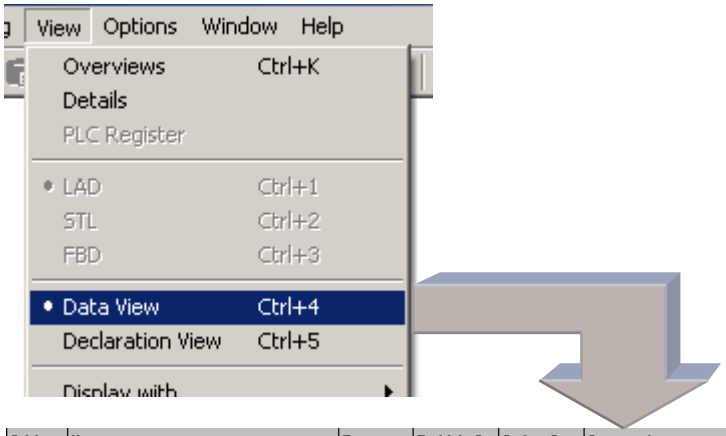
1. DB_Config (Block_DB)

The DB specified by this input represents the configuration of function block. Its form (the variables in it) is detailed in 'UDT103 (DAD_DPD_Config), and the DB should not contain other declarations (its size is checked).
This configuration DB is read and evaluated only on the FB activation, and sets internal variables (STAT filed in the DB of FB instance) based on the corresponding values. Changing the values of this DB when the FB input 'Enable' is TRUE, will have no effect, and any changes will be evaluated and applied only after the subsequent activation of the FB.

Below is the correct setup of this DB declaration:

Addres	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	Cfg	"DAD_DPD_ConfigForm"		DAD_DPD function block configuration
=26.0		END_STRUCT		

After declaring the structure of the DB, you must initialize the variables:



Addres	Name	Type	Initial	Actual v	Comment
0.0	Cfg.In_FieldArea.FirstByte	INT	64	64	Input field area first byte number
2.0	Cfg.In_FieldArea.Size	INT	32	8	Input field area size (num. of bytes)
4.0	Cfg.In_FieldArea.Area	INT	0	0	Input field area type: 0=E / 1=A / 2=M / 3=DB (other codes not supported)
6.0	Cfg.In_FieldArea.DB_Num	INT	0	0	Input field area Number of DataBlock (if Area = 3)
8.0	Cfg.Rx_Buffer	INT	0	101	Number of DataBlock where messages (from Slave) will be stored (starting at 0)
10.0	Cfg.Out_FieldArea.FirstByte	INT	64	64	Output field area first byte number
12.0	Cfg.Out_FieldArea.Size	INT	32	8	Output field area size (num. of bytes)
14.0	Cfg.Out_FieldArea.Area	INT	1	1	Output field area type: 1=A / 2=M / 3=DB (other codes not supported)
16.0	Cfg.Out_FieldArea.Db_Num	INT	0	0	Output field area Number of DataBlock (if Area = 3)
18.0	Cfg.Tx_Buffer	INT	0	102	Number of DataBlock where messages to send (at Slave) will read (starting at 0)
20.0	Cfg.DAD	BOOL	TRUE	TRUE	Set protocol mode: TRUE = DAD / FALSE = DPD
20.1	Cfg.Consistency	BOOL	FALSE	TRUE	Set protocol consistency mode: TRUE = enable
20.2	Cfg.Digital_IO	BOOL	FALSE	FALSE	Digital Input/Output via FieldBus mode: TRUE = enable
20.3	Cfg.Spare_1	BOOL	FALSE	FALSE	Reserved for future use
20.4	Cfg.OverwriteProtect	BOOL	TRUE	TRUE	This option disable data reception while `DataReady` is TRUE
20.5	Cfg.Read_En	BOOL	TRUE	TRUE	Read from partner function enable: TRUE = enable
20.6	Cfg.Write_En	BOOL	TRUE	TRUE	Write to parnter function enable: TRUE = enable
20.7	Cfg.Spare_2	BOOL	FALSE	FALSE	Reserved for future use
22.0	Cfg.Sync_T_Out	TIME	T#10S	T#10S	Maximum delay for handshake (from partner) during ReSyncrhronization procedure

Following the datailed list of the configuration DB items:

I. DAD_DPD_Config.In_FieldArea.FirstByte (INT)
It represents the address of the first byte of the input area assigned to the fieldbus (Profibus / Profinet)

II. DAD_DPD_Config.In_FieldArea.Size (INT)

Sizing the input area (Fieldbus) to FB. This parameter indicates to the FB how many bytes to read in the input signal field. The minimum value is 8, but in any case, this value has to be agreed with the number of bytes declared in the hardware configuration as inputs (or if you use a different area, that area must have size (num. of bytes) greater than or equal to the value set in this variable).

III. DAD_DPD_Config.In_FieldArea.Area (INT)

Defines the area of the input signals to the function block. Possible values: 'E', 'A', 'M', 'DB'. Normally, that is for actual use of the function block, this variable is assigned the value 0 (default), so that we inform the FB to take the input signals from the inputs (fieldbus). To test or debug functions, we can force the function block to take the input signals from different memory areas (e.g. M or DB). This will make it possible to manipulate (or simulate) the signals (including chars), which would be supplied by partner.

IV. DAD_DPD_Config.In_FieldArea.DB_Num (INT)

This parameter indicates the number of DB from which the FB will read the input signals, if the variable 'Cfg.In_FieldArea.Area' = 3. Otherwise, the value of this variable is not taken into account. If the input area has been defined on a DB ('Cfg.In_FieldArea.Area' = 3), this DB must exist and must be sized to contain the entire defined area (No. of bytes set in variable 'Cfg.In_FieldArea.Size', starting from the address specified in the 'Cfg.In_FieldArea.FirstByte'). The correctness of this approach is verified at the first activation of the FB.

V. DAD_DPD_Config.Rx_Buffer (INT)

This is the number of DB that the FB will use as an Rx buffer. The received data, sent by their partners, will be 'defragmented' (if necessary) and stored in DB indicated by the value of this parameter, always starting from the address DBx0.0. This means that each incoming message will overwrite the previous one, this is a mechanism designed to have overwrite protection (see below). The size of this DB should be proportionate to the size of the messages that you expect to receive, but in case of overflow (greater than the length of the received data buffer size-DB), reception stops (while continuing to respond to the partner to avoid block), and an alarm is generated to indicate that the message stored in the receive buffer is not complete. The DB should not be 'Unlinked' and the attribute 'ReadOnly' must not be set, otherwise an alarm is generated to the activation of the FB, as shown if the DB is not existing in CPU memory.

VI. DAD_DPD_Config.Out_FieldArea.FirstByte (INT)

Represents the address of the first byte of the output area assigned to the fieldbus (Profibus / Profinet)

VII. DAD_DPD_Config.Out_FieldArea.Size (INT)

It sizes the FB output area (Fieldbus). This parameter indicates how many bytes to read in the FB output signal field. The minimum value is 8, but in any case, this value has to be agreed with the number of bytes declared in the hardware configuration as outputs (if you use a different area, that area must have size (no. of bytes) more or equal to the value set in this variable).

VIII. DAD_DPD_Config.Out_FieldArea.Area (INT)

Defines the area of the output signals to the function block. Possible values are: 'A', 'M', 'DB'. Normally, that is for actual use of the function block, this variable is assigned the value 0 (default), that you inform the FB to take the output signals from the output area (fieldbus). To test or debug functions, you can force the function block to take the input signals from different memory areas (e.g. M or DB). This will make it possible to manipulate (or simulate) the signals (including characters), which would be supplied by partner.

IX. DAD_DPD_Config.Out_FieldArea.Db_Num (INT)

This parameter indicates the number of DB in which the FB should deposit the output signals, if the variable 'Cfg.Out_FieldArea.Area' = 3. Otherwise, the value of this variable is not taken into account. If the output area was defined on the DB ('Cfg.Out_FieldArea.Area' = 3), this DB must exist and must be sized to contain the entire defined area (No. of bytes set in variable 'Cfg.Out_FieldArea.Size', starting at the address specified by the variable 'Cfg.Out_FieldArea.FirstByte'. The correctness of this approach is tested at the first activation of the FB.

X. DAD_DPD_Config.Tx_Buffer (INT)

This is the number of DB that the FB will use as a TX buffer. The data sent to the partner will be taken from the DB indicated by the value of this parameter, always starting from the address DBx0.0. The size of this DB should be proportionate to the size of the messages you are sending. The DB should not be 'Unlinked' otherwise an alarm is generated on the FB activation, as if the DB is not existing in CPU memory.

XI. DAD_DPD_Config.DAD (BOOL)

Definition of the communication protocol

- TRUE: DAD is running
- FALSE: DPD is running
  - Default = TRUE

XII. DAD_DPD_Config.Consistency (BOOL)

Enable / disable data consistency control (Default = FALSE)

- TRUE: consistency control enabled
- FALSE: consistency control disabled
  - Default = FALSE

XIII. DAD_DPD_Config.Digital_IO (BOOL)

Enable / disable Digital I/O by FieldBus (Default = FALSE)

- TRUE: Digital I/O on Fieldbus enabled
- FALSE: Digital I/O on Fieldbus disabled
  - Default = FALSE

XIV. DAD_DPD_Config.OverwriteProtect (BOOL)

Overwrite protection activation. When the FB detects a write request from the partner, it activates a control to enable or disable the partner to send the message. If the attribute 'Cfg.OverwriteProtect' is TRUE and the variable 'DataReady' (see below) is TRUE, then the received text is rejected.

- Default = TRUE

XV. DAD_DPD_Config.Read_En (BOOL)

Enable the reception of messages: the logic for receiving messages and their associated parameters are disabled (not done) if this parameter is FALSE. In this case you can choose to not initialize the variables related to the RX messages (as they are not checked). This parameter can also be changed with the entry 'Enable' = TRUE, and its variation will take effect immediately (on first activation, reception configuration parameters will still be controlled).

- Default = TRUE

XVI. DAD_DPD_Config.Write_En (BOOL)

Enable the transmission of messages: the logic for transmitting messages and their associated parameters are disabled (not done) if this parameter is FALSE. In this case you can choose to not initialize the variables related to the TX messages (as they are not checked). This parameter can also be changed with the entry 'Enable' = TRUE, and its variation will take effect immediately (on first activation, transmission configuration parameters will still be controlled).

- Default = TRUE

XVII. DAD_DPD_Config.Sync_T_Out (TIME)

This variable (32 bit) defines the maximum response time allowed to a partner for the 'Resync' command or 'FlushQueue' command. If the partner does not respond within the time-out, an alarm will be generated.

- Default: 10s

2. Host_Exchange (POINTER)

FB communicates with the program, as well as through the receive buffer (Rx_Buffer) and transmission buffer (Tx_Buffer), with variables indicating a completed reception ('DataReady'), how many characters have been received ('Rx_Chars'), a TX to be done ('Wr_Chars'), etc..

All of these signals (numeric or Boolean) are 'included' in a variable structure, whose 'type' is defined in the UDT104 ('DAD_DPD_HostExgForm'). The size of this variable is 10bytes and is therefore not possible to 'pass' FB directly. Because of this, it was defined as a pointer: it represent. The address of the beginning of the 10bytes variable. The indication of this address in the connection of the FB can NOT be omitted because, in case of failure, the software will detect an invalid address and the CPU will go in Stop State. Note that we don't provide a default address to prevent accidental writes to addresses already used by the application in which the FB is integrated.
This structure includes two numeric variables ('Wr_Chars'and 'Rx_Chars') and a Boolean variable ('DataReady') of interest for the application that uses this protocol.

- 'Wr_Chars': is a numeric value (INT) that indicates how many characters should be sent to the partner via the DAD / DPD protocol. This value must be set by the application that includes this FB, and in fact this is the command to transmit a message (if 'Tx_Chars' <> 0). When the FB notes 'Tx_Chars'> 0, it triggers a TX sequence for the message contained in the TX DB buffer. During transmission, the FB will decrease the value of 'Tx_Chars' continuously indicating how many characters are still being sent, until it reaches a value of 0 indicating that the transmission is completed.
- 'Rx_Chars': a numeric value (INT) which represents the number of characters received in the last receipt. If 'DataReady' = FALSE then 'Rx_Chars'= 0.
- 'DataReady': Indicates that a reception has been completed, and the message is copied to the RX DB is complete and valid. If 'OverWriteProtect' is TRUE, the next message reception will be rejected until 'DataReady' is not set to FALSE.

The structure of the variable 'HostExchange' is shown below:

Addres	Name	Type	Initial v	Comment
0.0		STRUCT		
+0.0	Rx_Buffer	INT	0	Data recive buffer (of DAD_DPD FB) DataBlock number
+2.0	Tx_Buffer	INT	0	Data write buffer (of DAD_DPD FB) DataBlock number
+4.0	Wr_Chars	INT	0	Num of chars (remaining) to write via DAD_DPD FB (Write command > 0)
+6.0	Rx_Chars	INT	0	Num of recived chars in the last reception
+8.0	DataReady	BOOL	FALSE	String from partner is ready to be read (reception complete) via DAD_DPD FB
=10.0		END_STRUCT		

3. Enable (BOOL)
Enable FB processing. At the rising edge (transition from FALSE to TRUE) of this variable, an Interface Control (variables related to) of the FB is executed, and then the execution of the protocol code will be enabled. When 'Enable' = FALSE, the values of variable 'HostExchange' are not updated.

4. Partner_Ready (BOOL)
This signal enables the communication with the partner. When 'Partner_Ready' is TRUE, the control byte of the protocol is evaluated in order to generate an alarm if there is no congruence between signals. When 'Partner_Ready' is FALSE, the FB execution is stopped until 'Partner_Ready' becomes TRUE. During this time no signal from the partner will be evaluated and no message are sent. This input is useful for debugging functions or to prevent unexpected writing or reading.

5. Reset (BOOL)
Alarm delete command. If an alarm is active (shown from the 'Alarm' and specified by the variable 'Error_Code'), the FB execution is stopped until the alarm will be cancelled by this command. If the situation that caused the alarm remains active, the alarm will not be cancelled. It suffice just a rising edge of this command to delete an alarm, 'Reset 'always TRUE does not inhibit the generation of alarms.

6. Received (BOOL)
   This command inform the FB the last received message was 'taken over' by the application, and so the receive buffer may receive the next message overwriting the existing one.
   On 'Received' activation (rising edge), the FB will set 'DataReady' = FALSE (and hence 'Rx_Chars' = 0) - (variables included in the structure 'HostExchange)
   - If 'OverWriteProtect' = TRUE, this command will enable receiving an other message from the partner
   - If 'OverWriteProtect' = FALSE, resetting the 'DataReady'and 'Rx_Chars' variables after the 'Received' activation, will have no effect.

7. ReSync_Cmd (BOOL)
   This command starts the synchronization procedure of the communication protocol with the partner. At the end of the procedure the signal (output) 'Done' will be set, and the command 'ReSync_Cmd' = TRUE will have no effect. When 'ReSync_ Cmd'' changes from TRUE to FALSE, the signal 'Done' will be set to FALSE. If the synchronization process should not be successful (the partner does not respond), it sets an alarm when the monitoring-time of the procedure expires (shown in the configuration variable 'Cfg.Sync_T_Out').

8. FlushQueue_Cmd (BOOL)
   This command requires a cancellation of the message queue of the partner. At the end of the procedure the signal (output) 'Done' will set, and the command 'FlushQueue_Cmd' = TRUE has no effect. When 'FlushQueue_Cmd' changes from TRUE to FALSE, the signal 'Done' is set to FALSE. If the cancellation request is refused by the partners (or the partner does not respond), an alarm will be set on the monitoring-time of the procedure expiration (shown in the configuration variable 'Cfg.Sync_T_Out')

9. Digital_Output (BYTE)
   This byte represents (bit per bit) the command of the digital outputs that could be configured via fieldbus using the variable 'Cfg.Digital_IO'. If 'Cfg.Digital_IO' is FALSE, this entry has no meaning.

FB Output description

A. Echo (BYTE)

This byte represents (bit for bit) the configuration assigned to the FB after its activation. The states shown here refer only to digital settings evaluated after the first enabling. If the FB is not enabled (input 'Enable' = FALSE), the byte 'Echo' is not significant. The following describes the sequence and the relative significance of the bits of the byte 'Echo' (also shown into UDT105 'DAD_DPD_EchoForm'):

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	DAD_DPD	BOOL	FALSE	TRUE = DAD protocol mode / FALSE = DPD protocol mode
+0.1	Digital_IO	BOOL	FALSE	Digital Input/Output via FielBus: TRUE = enabled / FALSE = disabled
+0.2	Consistency	BOOL	FALSE	Consistency mode: TRUE = enabled / FALSE = disabled
+0.3	Spare_1	BOOL	TRUE	Reserved for future use
+0.4	OverwriteProtect	BOOL	TRUE	TRUE = disable reception while `DataReady` is TRUE / FALSE = reception enabled
+0.5	Read_En	BOOL	TRUE	Read from partner function enabled
+0.6	Write_En	BOOL	TRUE	Write to parnter function enabled
+0.7	Spare_2	BOOL	FALSE	Reserved for future use
=2.0		END_STRUCT		

B. Node_DPD_Only (INT)

Displaying the node number of partners who sent the last message. This parameter makes sense only if the used protocol is 'DPD'. If 'DAD' was set, the value of this output is forced = 0.

C. Command_Done (BOOL)

Signal (BOOL) indicating the end of the 'Resync'or 'FlushQueue' procedure.
When set to TRUE the entry 'ReSync_Cmd', or set to TRUE the entry 'FlushQueue_Cmd', FB stop any carrying out the procedures of writing and / or reading, and follow the commanded procedure. In this procedure, the partner must respond (in the manner and with the signals specified in the protocol) within a preset time (set in the DB configuration parameter 'DAD_DPD_Config.Sync_T_Out'). If not, the corresponding alarm is activated (see table of alarms), while if the partner responds by completing the procedure within the set time, the signal 'Command_Done' is set to TRUE. This signal remains TRUE until the corresponding command is TRUE. If the entire procedure is completed without errors (i.e. no alarm is set), on the 'Command_Done' signal activation, any broken procedure will be completed.

D. Alarm (BOOL)

Alarm active signal. When this digital output is TRUE execution of any active procedure is interrupted, and the FB no longer performs any action until the 'Reset' input is activated. If the situation (or event) that caused the alarm has not changed, the alarm will be immediately re-set (the state of the 'Alarm' output is TRUE, no transition to FALSE). When an alarm situation will be recovered, procedures that were eventually withdrawn will be terminated if possible. The specific active alarm signal is given by 'Error_Code' signal.

E. Error_Code (INT)

Specification for the active alarm. The value of this output is set to a value other than 0 when the output signal 'Alarm' is TRUE. The value set indicates the cause for alarm in accordance with the attached table.

F. Consistency_Warning (BOOL)

This signal indicates an inconsistency detected during the consistency check in the Input Area of the fieldbus. This check is performed continuously when the FB is:

- On (‘Enable’ = TRUE)
- Active (‘Parnter_Ready’ = TRUE)
- No Alarm present (‘Alarm’ = FALSE)

TRUE state of this output may indicate a disturbance in the communication (usually in this case we observe an oscillation of the signal) or a difference between the FB configuration and the partner configuration.

- If 'DAD_DPD_Config.Consistency' = TRUE and we are currently receiving a message from the partner, on the activation of ‘Consistency_Warning’ the corresponding alarm will be activated and the reception cancelled. In this case the message received from the partner will be restarted on the alarm reset (in order not to block the partner that is performing a procedure for transmission), but will be considered as a new message that will overwrite the first part of the previous received message in the buffer.

  NOTE: when a warning 'Data Consistency Error' (AlarmCode = 400) occurs, the message stored in the receive buffer will not be complete (in particular, will be present only the part received after the consistency alarm reset).

G. Rx_Warning (BOOL)

This signal indicates that a communication from the partner is waiting to be received but not accepted as:

- 'DataReady' = TRUE (e 'Rx_Chars' <> 0)
- DAD_DPD_Config.OverwriteProtect = TURE

That is the last message received is still in the receive buffer (not yet 'taken over'), and the overwrite protection option has been enabled.

H. Rx (BOOL)

Indicates a data reception in progress.

I. Tx (BOOL)

Indicates a data transmission in progress.

J. Digital_Input (BYTE)

This byte represents (bit per bit) the digital input state when activated on fieldbus protocol by using the variable 'Cfg.Digital_IO'. If 'Cfg.Digital_IO' is FALSE, this output has zero value.

Error table

Action	Error code	Description
Interface control	1	Config DB error: not present or 'UNLINKED' attribute
	2	Config DB error: bad size (may be a wrong DB)
	6	Unsupported Host interface area code
	7	Host interface DataBlock not present or 'UNLINKED'
	8	Host interface DataBlock is 'WriteProtect'
	11	Unsupported Input field area code
	12	Input field area = 3 (DB) but DB not present or 'UNLINKED'
	13	
	14	Input field area size error
	15	Rx_Buffer error: not present or 'UNLINKED'
	16	Rx_Buffer DB is 'WriteProtect'
	21	Unsupported Output field area code
	22	Output field area = 3 (DB) but DB not present or 'UNLINKED'
	23	Output field area = 3 (DB) but DB is 'WriteProtect'
	24	Output area size error
	25	Tx_Buffer error: not present or 'UNLINKED'
Partner wait	102	No match between configured DAD/DPD protocol and partner DAD/DPD protocol
Resync	202	Synchronization procedure error: time out of partner (slave) handshake
Flush Queue	302	FlushQueue' service time out error: Partner didn't answer in preset time
	304	Flush queue command rejected
	306	Unknown Slave answer at 'FlushQueue' command

Read	400	Data consistency error
	402	Data protocol error: Num of char to receive > num of char in Input field area (May be a wrong configuration)
	404	Rx_Buffer overflow READ INCOMPLETE
Write	502	Tx_Buffer overflow (num of required char to send > num of char in Tx_Buffer)
General	-1	Software error: Bad function index