

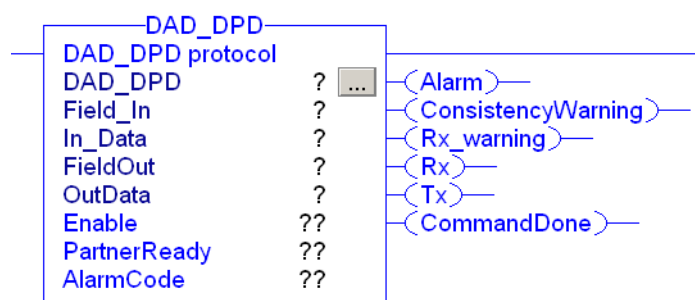
## DAD\_DPD Add-ON

Software implemented on 'RSLogix 5000 Professional/Network Edition' - V17.01.00

The Add\_On implements DAD and DPD protocols. It allows a full communication between Datalogic devices (Laser and Matrix) and Allen Bradley ControlLogix PLC's on the EthernetIP fieldbus (implicit messaging). This implementation has been developed in order to simplify the development of a generic software application on Allen Bradley PLC's.

Communication between PLC and the Datalogic device (partner), made by this Add\_On, uses a string (called the receive buffer - 'Rx\_Buffer') for receiving messages, and a string (called the transmission buffer - 'Tx\_Buffer') for transmission.

When the 'Add\_On' is enabled (Enable Input = 1), is not in alarm (output Alarm = 0 and output AlarmCode = 0), and the signal 'Partner\_ready' = 1, the connection is established, and incoming messages (from signal partner) are copied in the receive buffer (string), as well as messages stored in transmission buffer (string) are sent to partner.

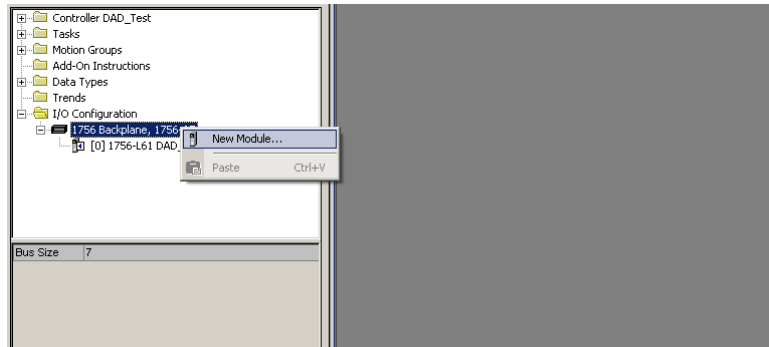


## Hardware Configuration

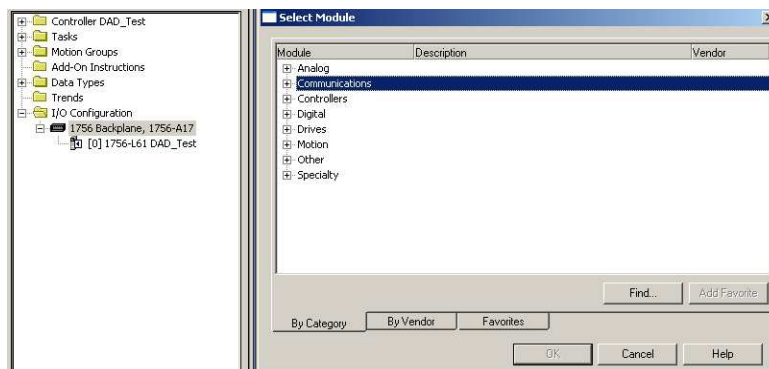
To establish the communication between the PLC and the Datalogic partner (in this example, a Laser device), it is necessary to properly design the hardware.

Once designed the CPU and acquired in the development system the file '368-0164-EDS\_ABCC\_EIP\_V\_2\_1.eds' (which specifies the characteristics of EthernetIP-CIP device that is intended to connect) supplied by Datalogic, perform the following configuration:

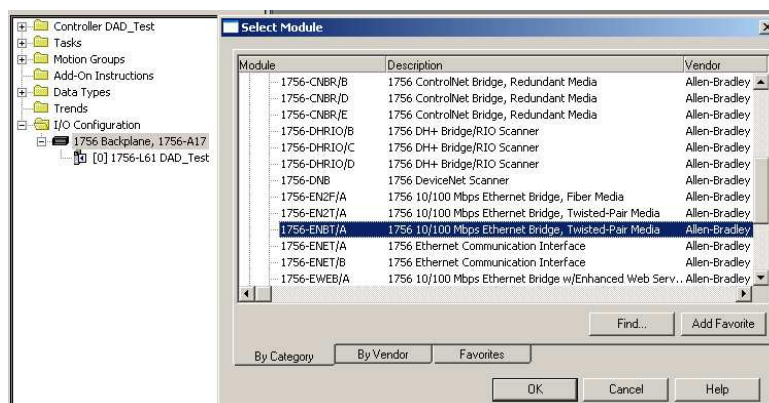
- With the 'New Module ' option that appears by right-clicking on the icon of the Backplane:



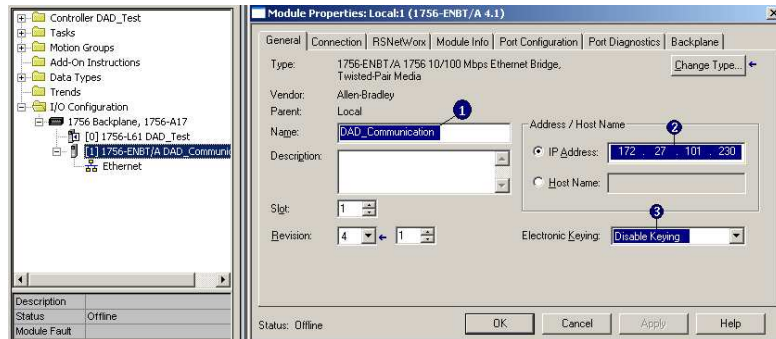
- Select 'Communication':



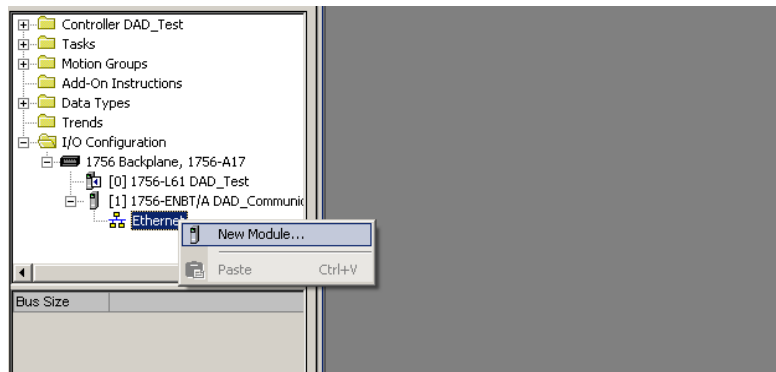
- Select the model of the used module ('1756-ENBT/A' in this example):



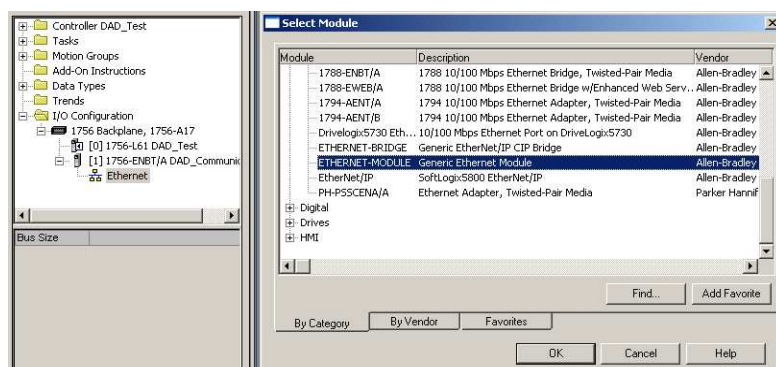
- Then fill in the configuration dialog, in which you specify:
  1. The name that means for the form (in this example 'DAD\_Communication')
  2. Set the IP address that is intended to give the PLC module (eg '1756-ENBT / A')
  3. Select the option 'Disable keying'



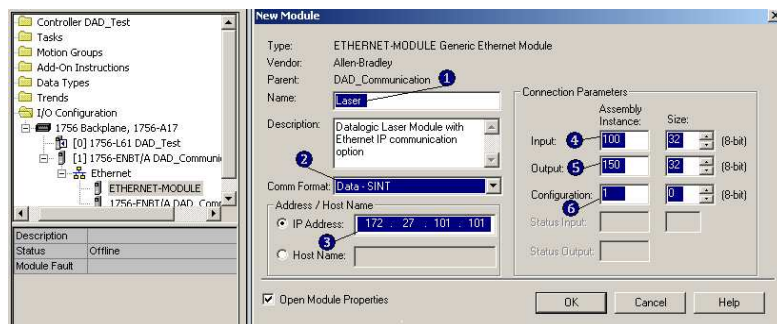
- Now the main features on the communication module mounted on the backplane of the PLC are set. In the next step you can configure the characteristics of the communication partner: right-click on the 'Ethernet' which was created by the environment when the PLC module has been configured.



- And select the module shown in the picture below. Please note that the voice selected here refers to the characteristics of the module that were previously 'loaded' into the development environment through the inclusion of the file '368-0164-EDS\_ABCC\_EIP\_V\_2\_1.eds' provided by Datalogic. If the uploading process of the file 'xxx.eds' was not executed properly, this item does not appear!



- Then fill in the configuration, in which you must specify:
  1. The name of the device (here, 'Laser')
  2. The type of exchanged data. The Add-On is written to process the input data type SINT. The selection of a different type of data will cause the failure of the function.
  3. The IP address that is intended to assign to the communication partner
  4. The size (in bytes) of receive buffer (here = 32). This value can be assigned freely, but must be  $\geq 8$  and the sum of the receiving buffer size (this one) and that of the transmission buffer must be  $\leq 152$ . The address of the instance (here = 100)
  5. The size (in bytes) of transmit buffer (here = 32). This value can be assigned freely, but must be  $\geq 8$  and the sum of the size of the transmission buffer (this) and that of the receive buffer must be  $\leq 152$ . The address of the instance (here = 150)
  6. The size of configuration buffer (service). The value set here is not important because it is set by the system. The address of the instance (here = 1)



## Input description

Name	Usage	Data Type	Default	Style	Req	Vis	Description
EnableIn	Input	BOOL	1	Decimal	<input type="checkbox"/>	<input type="checkbox"/>	Enable Input - System Defined Parameter
EnableOut	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input type="checkbox"/>	Enable Output - System Defined Parameter
± Config	Input	SINT	2#0111_0001	Binary	<input checked="" type="checkbox"/>	<input type="checkbox"/>	DAD_DPD work mode configuration flags
± Field_In	InOut	SINT[16]		ASCII	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Field input area
± In_Data	InOut	DAD_RxTx			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Data coming from Partner (Slave) - Data read
± FieldOut	InOut	SINT[16]		ASCII	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Field output area
± OutData	InOut	DAD_RxTx			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Data to send to Partner (Slave) - Data write
Enable	Input	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Function execution enable
PartnerReady	Input	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Communication partner ready
Resync_Cmd	Input	BOOL	0	Decimal	<input type="checkbox"/>	<input type="checkbox"/>	Resync command
FlushQueue_Cmd	Input	BOOL	0	Decimal	<input type="checkbox"/>	<input type="checkbox"/>	Flush queue command
Reset	Input	BOOL	0	Decimal	<input type="checkbox"/>	<input type="checkbox"/>	Alarm reset command
Alarm	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alarm active
ConsistencyWarning	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Consistency warning
Rx_warning	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Rx Buffer overwrite warning
Rx	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Data receiving
Tx	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Data sending
CommandDone	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Resync command or FlushQueue command successful done
± AlarmCode	Output	INT	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alarm code (error specification)
± ProcTimeOut	Input	DINT	10000	Decimal	<input type="checkbox"/>	<input type="checkbox"/>	Procedure time-out preset (ms) - (used for Resync and FlushQueue procedures)
± Digital_Outputs	Input	SINT	2#0000_0000	Binary	<input type="checkbox"/>	<input type="checkbox"/>	Digital outputs command (bit by bit)
± Digital_Inputs	Output	SINT	2#0000_0000	Binary	<input type="checkbox"/>	<input type="checkbox"/>	Digital inputs status (bit by bit)
± DPD_Node	Output	SINT	16#00	Hex	<input type="checkbox"/>	<input type="checkbox"/>	Number of node (PDP mode only)

- **EnableIn:** BOOL.  
This system input enables the Add-On function. To set this input to 0, equals to 'jump' the function code. This function performs some set-to-zero operations even when not 'working' in order to have clean data when it is called to perform the procedures. Disabling the function by setting to zero this input may cause a malfunction the next time the procedure will be performed. We recommend to keep always enabled this input.
- **Config:** SINT.  
The function (Add-On) DAD-DPD, allows you to configure the protocol used for communication (DAD or DPD), the 'mode' of communication (Digial I / O, Consistency), and some other optional features.  
To this end a variable input have been introduced that, in view bit by bit, enables or disables the options. This variable is deliberately not been represented in the interface connecting the Add-On as they generally do not have to drive it dynamically. A default configuration is supplied: it includes the most common settings, so that the final application developer does not need to change any setting.

± Config	Input	SINT	2#0111_0001	Binary	<input type="checkbox"/>	<input type="checkbox"/>	DAD_DPD work mode configuration flags
± Config.0		BOOL	1	Decimal			DAD or DPD mode: 0 = DPD mode 1 = DAD mode
± Config.1		BOOL	0	Decimal			Data Consistency check (1 = enable)
± Config.2		BOOL	0	Decimal			Digital Inputs / Outputs enable (1 = Enable)
± Config.3		BOOL	0	Decimal			Spare: Reserved for future use
± Config.4		BOOL	1	Decimal			Overwrite protection: if enabled (= 1) receiving message is disabled while 'DataReady' is TRUE (dynamic set)
± Config.5		BOOL	1	Decimal			Read from partner function enable: TRUE = enable (dynamic set)
± Config.6		BOOL	1	Decimal			Write to partner function enable: TRUE = enable (dynamic set)
± Config.7		BOOL	0	Decimal			Spare: Reserved for future use

Here's the complete description.

Please note the 'Dynamic' column: it specifies if the option is dynamically changed.

- An option that is NOT dynamically modifiable, it is considered only on the function (Add-On) activation, which will be set accordingly. Any subsequent change in option will not produce any effect, and the function will continue to work as configured on the Add-On activation.
- An option that is dynamically modifiable, causes the change in the behavior of the function (Add-On) every time its value is changed, even if the function is working.

Bit	Default	Description		Dynamic
0	1	This bit specifies whether the protocol should be used for communication with DPD or DAD.	0 = DPD 1 = DAD	No
1	0	Enables the consistency check of the communication.	0 = No consistency 1 = Consistency	No
2	0	Enables the digital I/O exchange on the communication buffer.	0 = I/O disabled 1 = I/O enabled	No
3	0	Reserved for future use.		--
4	1	Disable the reception of messages from the partners if the receive buffer is not empty. This option is seen as an overwrite protection to a received message. If this option is selected, you must reset the value of the variable 'Rx_Buffer.LEN' to receive a new message from the partner. Until this action, the reception of messages from the partner will be rejected. Otherwise, i.e. if the protection is disabled, no protection will be interposed against overwriting (and potential loss) of messages received by the partner, and the reception will be exclusively entrusted to the request from the partner.	0 = Protection disabled.  1 = Protection enabled.	Yes
5	1	When this option is disabled, any message received from the communication partner is rejected regardless of the status of the receive buffer. It can be used to avoid to acquire any disturb when the partner does not make writings. In the case of deactivation during a reception (output Rx = 1), the flow will stop in receiving data thus effectively blocking the communication without having completed the incoming message.	0 = Rx-Disabled 1 = Rx-Enabled	Yes
6	1	This option enables the writing of messages to the partner. If not set, the function (Add-On) will not trigger the process of writing to the partner, and no message will be sent. In the case of deactivation during a transmission (output Tx = 1) the text will be stopped and the communication will not be completed thereby blocking actually writing to the partner.	0 = Tx-Disabled 1 = Tx-Enabled	Yes
7	0	Reserved for future use		--

- **Enable:** BOOL.  
Added to inactivate the function. This is the correct signal on which work to disable the functionality of the Add-On. The logic of this input does not prevent the execution of all code, but it allows the execution of a small part which keeps the dynamic variables updated. If you want to temporarily inactivate this function, without causing malfunction when it needs its execution, you should act on this input signal.

- **PartnerReady:** BOOL.  
Indicates the availability status of the communication partner. When this signal is set to 0 (FALSE), the function stops processing the input buffer area (Field\_In) and sets to 0 all the bytes of the output buffer (FieldOut). This signal can be used in case of partner off (or loss of communication in general) to avoid generation of the 102 Alarm (No match between configured DAD/DPD protocol and partner DAD/DPD protocol).
  
- **ReSync\_Cmd:** BOOL.  
The change in the status of this variable from FALSE to TRUE (from 0 to 1), causes the execution of the synchronization procedure described in the DAD / DPD Protocol. If the procedure will be successful, the Add-On will set to TRUE (1) the output signal 'CommandDone', and the 'ReSync\_Cmd' = 1 will no longer have any effect. When 'Cmd ReSync\_' is switched from TRUE (1) to FALSE (0), the signal 'CommandDone' is set to FALSE (0).  
If the synchronization process should not be successful (the partner does not respond), it sets an alarm when the time monitoring of the procedure expires (detected in the variable 'ProcTimeOut').
  
- **FlushQueue\_Cmd:** BOOL.  
This command controls the deletion of the message queue of the partner. At the end of the procedure the output signal 'CommandDone' will be set to TRUE (1), and the 'FlushQueue\_Cmd'=1 will no longer have any effect. When 'FlushQueue\_Cmd' is switched from TRUE (1) to FALSE (0), the signal 'CommandDone' is set to FALSE (0).
  - If the command is rejected by their partner (with a specific message described by DAD / DPD protocol), the 304 alarm is set (Flush command queue Rejected).
  - If the partner's response to the command is not comprehensible, the 306 alarm is set (Unknown Slave answer at 'FlushQueue' command).
  - If partner does not respond within the time specified by the variable 'ProcTimeOut', the 302 alarm is set (FlushQueue service time out error: partner did not answer at preset time).
  
- **Reset:** BOOL.  
Delete command for current active alarm. If an alarm is activated (shown from the 'Alarm' output and specified by the variable 'AlarmCode'), the function (Add-On) is interdicted, waiting for the alarm is canceled by this command. If the situation that caused the alarm remains triggered, the alarm will not be erased. It suffices just a rising edge of this command to delete an alarm. 'Reset' always TRUE does not inhibit the generation of alarms.
  
- **ProcTimeOut:** DINT.  
Specifies the maximum time (ms) to carry out processes controlled by the boolean inputs 'ReSync\_Cmd' and 'FlushQueue\_Cmd'. The time specified here is the maximum execution time of the procedure performed, and when it expires the corresponding alarm is set.
  
- **Digital\_Outputs:** SINT.  
This byte represents (bit per bit) the command of the fieldbus digital outputs configured using the variable 'Config.2' (digital inputs / outputs enable). If bit 2 of variable 'Config' is FALSE (0), this input has no meaning.



## Output description

Add-On Instruction Definition - DAD_DPD v5.0									
General Parameters Local Tags Scan Modes Change History Help									
Name	Usage	Data Type	Default	Style	Req	Vis	Description		
EnableIn	Input	BOOL	1	Decimal	<input type="checkbox"/>	<input type="checkbox"/>	Enable Input - System Defined Parameter		
EnableOut	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input type="checkbox"/>	Enable Output - System Defined Parameter		
Config	Input	SINT	2#0111_0001	Binary	<input type="checkbox"/>	<input type="checkbox"/>	DAD_DPD work mode configuration flags		
Field_In	InOut	SINT[16]		ASCII	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Field input area		
In_Data	InOut	DAD_RxTx			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Data coming from Partner (Slave) - Data read		
FieldOut	InOut	SINT[16]		ASCII	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Field output area		
OutData	InOut	DAD_RxTx			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Data to send to Partner (Slave) - Data write		
Enable	Input	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Function execution enable		
PartnerReady	Input	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Communication partner ready		
ReSync_Cmd	Input	BOOL	0	Decimal	<input type="checkbox"/>	<input type="checkbox"/>	Resync command		
FlushQueue_Cmd	Input	BOOL	0	Decimal	<input type="checkbox"/>	<input type="checkbox"/>	Flush queue command		
Reset	Input	BOOL	0	Decimal	<input type="checkbox"/>	<input type="checkbox"/>	Alarm reset command		
Alarm	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alarm active		
ConsistencyWarning	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Consistency warning		
Rx_warning	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Rx_Buffer overwrite warning		
Rx	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Data receiving		
Tx	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Data sending		
CommandDone	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Resync command or FlushQueue command successful done		
AlarmCode	Output	INT	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alarm code (error specification)		
ProcTimeOut	Input	DINT	10000	Decimal	<input type="checkbox"/>	<input type="checkbox"/>	Procedure time-out preset (ms) - (used for Resync and FlushQueue procedures)		
Digital_Outputs	Input	SINT	2#0000_0000	Binary	<input type="checkbox"/>	<input type="checkbox"/>	Digital outputs command (bit by bit)		
Digital_Inputs	Output	SINT	2#0000_0000	Binary	<input type="checkbox"/>	<input type="checkbox"/>	Digital inputs status (bit by bit)		
DPD_Node	Output	SINT	16#00	Hex	<input type="checkbox"/>	<input type="checkbox"/>	Number of node (PDP mode only)		

- **Rx:** BOOL.  
Indicates a data RX in progress.
- **Tx:** BOOL.  
Indicates a data TX in progress
- **Command\_Done:** BOOL.  
Digital signal (BOOL) indicating the end of the 'Resync' or 'FlushQueue' procedure. When input 'ReSync\_Cmd' or 'FlushQueue\_Cmd' are set to TRUE (1), the Add-On stops the eventual execution of the procedures of writing and / or reading, and follows the commanded procedure. The partner must respond to this procedure (in the manner and with the signals specified in the protocol) within a preset time (set in the input variable 'ProcTimeOut'). 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 (1). This signal remains TRUE until the corresponding command is TRUE (1). If the whole procedure is completed without errors (i.e. no alarm set), on the activation of the signal 'Command\_Done' all interrupted procedures are completed.
- **Rx\_Warning:** BOOL. This signal indicates that a communication from the partner is waiting to be received but not accepted because:
  - 'In\_Data.LEN' > 0
  - Config.4 (Overwrite Protection) = TRUE (1)

i.e. the last received message is still in the RX buffer (not yet 'taken over'), and overwrite protection is activated (Config.4 OverwriteProtect = TRUE).



- **ConsistencyWarning** 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 add-on is:

- Enabled ('Enable' = TRUE)
- Active ('PartnerReady' = TRUE)
- No Alarm present ('Alarm' = FALSE)

The TRUE (1) 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 configuration of the Add-On and of the communication partner.

If 'Config.1' = TRUE (1) and we are currently receiving a message from the partner, on the activation of 'Consistency\_Warning' the corresponding alarm is turned on and the receiving of the message stopped. In this case the receiving procedure from the partner will be restarted on the alarm reset event (in order not to block the execution of a transmission procedure by the partner), but will be considered as a new message and will overwrite the first part already stored in the receiving buffer. Therefore, due to a 'Data Consistency Error' (AlarmCode = 400) the message stored in the receive buffer will not be complete (in particular, there will be only the receipt after the resetting of consistency alarm).

- **Alarm** BOOL.

Alarm signal active. When this digital output is TRUE (1) execution of any active procedure is interrupted, and the Add-On no longer performs any action until input 'Reset' 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' is TRUE, no transition to FALSE). When the alarm situation is recovered, all the procedures that were eventually withdrawn, are terminated if possible. The specific active alarm signal is given by 'AlarmCode'.

- **AlarmCode:** INT.  
Active alarm. The value of this output is set to a value other than 0 when the output signal 'Alarm' is TRUE (1). The value indicates the cause for alarm in the table below.

AlarmCode	Description
<i>Data Input configuration Alarms</i>	
13	Rx_Buffer too small to contain a segment
14	Fieldbus Input area size error (< 8 bytes)
15	Fieldbus Input area size error (> 144 bytes)
<i>Data Output configuration Alarms</i>	
24	Fieldbus Output area size error (< 8 bytes)
25	Fieldbus Output area size error (> 144 bytes)
<i>I/O area size Alarm</i>	
35	Fieldbus total area size error (Input area + Output area) (global size > 152 bytes)
<i>Partner communication establishment Alarms</i>	
102	No match between configured DAD/DPD protocol and partner DAD/DPD protocol
<i>Synchronization procedure Alarms</i>	
202	Synchronization procedure error: time out of partner (slave) handshake
<i>Flush-Queue procedure Alarms</i>	
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
<i>Data Receiving Alarms</i>	
400	Data consistency error
402	Data protocol error: Num of char to receive > num of char in Input field area
404	Rx_Buffer_ overflow READ INCOMPLETE
<i>Data Transmitting Alarms</i>	
502	Tx_Buffer overflow (num of required char to send > num of char in Tx_Buffer )

- **Digital\_Input:** SINT.  
This byte represents (bit per bit) the state of digital inputs activated on by fieldbus protocol using the variable 'Config.2' (digital inputs / outputs enable). If bit 2 in variable 'Config' is FALSE (0), this output has 0 value.
- **DPD\_Node:** SINT.  
Displays the node number of the partner who have sent the last message. This parameter makes sense only if the protocol 'DPD' is in use (Config.0 = FALSE). If DAD protocol is set, the value of this output is forced to 0.

## Input/output description

Add-On Instruction Definition - DAD_DPD v5.0									
General Parameters Local Tags Scan Modes Change History Help									
Name	Usage	Data Type	Default	Style	Req	Vis	Description		
EnableIn	Input	BOOL	1	Decimal	<input type="checkbox"/>	<input type="checkbox"/>	Enable Input - System Defined Parameter		
EnableOut	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input type="checkbox"/>	Enable Output - System Defined Parameter		
⊕ Config	Input	SINT	2#0111_0001	Binary	<input type="checkbox"/>	<input type="checkbox"/>	DAD_DPD work mode configuration flags		
⊕ Field_In	InOut	SINT[16]		ASCII	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Field input area		
⊕ In_Data	InOut	DAD_RxTx			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Data coming from Partner (Slave) - Data read		
⊕ FieldOut	InOut	SINT[16]		ASCII	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Field output area		
⊕ OutData	InOut	DAD_RxTx			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Data to send to Partner (Slave) - Data write		
Enable	Input	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Function execution enable		
PartnerReady	Input	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Communication partner ready		
ReSync_Cmd	Input	BOOL	0	Decimal	<input type="checkbox"/>	<input type="checkbox"/>	Resync command		
FlushQueue_Cmd	Input	BOOL	0	Decimal	<input type="checkbox"/>	<input type="checkbox"/>	Flush queue command		
Reset	Input	BOOL	0	Decimal	<input type="checkbox"/>	<input type="checkbox"/>	Alarm reset command		
Alarm	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alarm active		
ConsistencyWarning	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Consistency warning		
Rx_warning	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Rx_Buffer overwrite warning		
Rx	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Data receiving		
Tx	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Data sending		
CommandDone	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Resync command or FlushQueue command successful done		
⊕ AlarmCode	Output	INT	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alarm code (error specification)		
⊕ ProcTimeOut	Input	DINT	10000	Decimal	<input type="checkbox"/>	<input type="checkbox"/>	Procedure time-out preset (ms) - (used for Resync and FlushQueue procedures)		
⊕ Digital_Outputs	Input	SINT	2#0000_0000	Binary	<input type="checkbox"/>	<input type="checkbox"/>	Digital outputs command (bit by bit)		
⊕ Digital_Inputs	Output	SINT	2#0000_0000	Binary	<input type="checkbox"/>	<input type="checkbox"/>	Digital inputs status (bit by bit)		
⊕ DPD_Node	Output	SINT	16#00	Hex	<input type="checkbox"/>	<input type="checkbox"/>	Number of node (PDP mode only)		

- **Field\_In** SINT[xx].  
In this ARRAY are written the 'characters' coming from the communication partner, i.e. it represents the fieldbus in reception. Its size can vary with the variation of hardware resources available and according to the settings of the developer. In any case, the size of this ARRAY (in this example, 32 SINT) must be modified by the developer to match the value that was assigned to the receive buffer size in the hardware configuration settings (Input for the configuration of the partner communication).
- **FieldOut** SINT[xx].  
In this ARRAY are written the 'characters' to send to the communication partner, i.e. it represents the fieldbus in transmission. Its size can vary with the variation of hardware resources available and according to the settings of the developer. In any case, the size of this ARRAY (in this example, 32 SINT) must be modified by the developer to match the value that was assigned to the transmit buffer size in the hardware configuration settings (Output for the configuration of the partner communication).

- **In\_Data:** DAD\_RxTx (STRING[256]).

In this string are stored messages coming from the communication partner. When the size of the string (see the DWord header in the string) is > 0, then a message was received from the communication partner. Its size is arbitrary (in this example = 256 characters), and may be freely modified by the developer in order to save resources (if necessary).

Receiving messages is done automatically, i.e. whenever the partner becomes a transmission, the message is received, reconstructed according to the rules of the protocol and stored in the string connected to this variable. The value of 'In\_Data.LEN' > 0 specifies that a message was indeed received.

Note: the reception is 'refused' by the Add-On only if 'In\_Data.LEN' > 0 (i.e. in the receive buffer there is already a string) and you have set 'Overwrite Protection' (config.4) = TRUE (1).

The value of the variable 'In\_Data.LEN' is updated from the Add-On only when the reception is completed. The application using this Add-On will take charge of zero-setting the value of 'In\_Data.LEN' ('In\_Data.LEN' := 0) to indicate the function that the message has been taken over, and that the reception buffer is available again. This practice is not necessary if you set 'Config.4' = 0 (no overwrite protection) because in this case the function does not check the availability of the receive buffer, but overwrite the message.

- **OutData:** DAD\_RxTx (STRING[256]).

In this string are stored messages to be sent to the communication partner. When the size of the string (see the DWord header in the string) is > 0, then the message written in this string is broken down according to the rules of the Protocol and sent to the communication partner to complete the entire string. At this point the value of 'OutData.LEN' is cleared by the function to indicate that the transmission is complete. Its size is arbitrary (in this example = 256 characters), and may be freely modified by the developer in order to save resources (if necessary).