

The Matrix_Esc Add-On

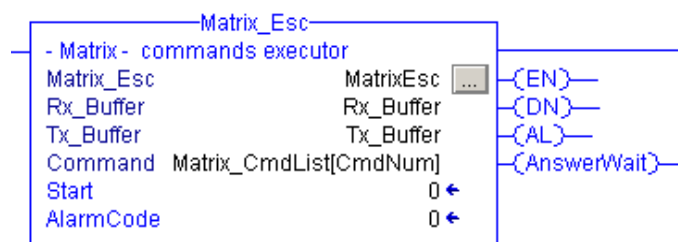
This Add-On has been implemented in order to provide basic help to PLC's application developers on the command configuration side with a Datalogic Matrix reader device.

This function processes the data set by the developer and organized according to the pre-stored format ('Matrix_Command'), and produces the corresponding command in the format readable by the Datalogic device. These commands (stored in the 'Tx_Buffer' string) are automatically sent to the device through the 'DAD_DPD' Add-On. Later, every response generated by the device itself (also read through 'DAD_DPD') will be stored in the 'Rx_Buffer' string and analyzed.

If the answer is correct compared to the command, the function considers the job done and sets TRUE (1) its output variable DN, indicating that the procedure was successful. Otherwise, i.e. if the answer is wrong, or it specifies a refusal by the device, the function generates an error signal (AL) and a relatively specific error code (AlarmCode).

Some commands are given in different strings, so, despite the function produces the appropriate string in a single scan of the program, it may take several scans to give a complete command. Responses in some cases are composed of multiple strings in a row, so it is necessary to test the signals 'DN' or 'AL' to consider or not the task done.

This function does not handle the command 'SET_String' because the size of its parameter 'Data' has a potentially large amount of memory. For this particular command has been implemented the Add-On 'Matrix_String'. This Add-On does exactly the same way as 'Matrix_Esc', but provides a string (the parameter 'Data' in the variable 'SetStringPar') to 256 characters (compared to 16 characters of the function 'Matrix_Esc').



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
Rx_Buffer	InOut	DAD_RxTx			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Reception buffer (for partner answers)
Tx_Buffer	InOut	DAD_RxTx			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Transmission buffer (for commands to partner)
Command	InOut	Matrix_Command			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Command with parameters
Start	Input	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Function command start
EN	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Function working
DN	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Function successful executed
AL	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alarm active
AlarmCode	Output	INT	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Code of Alarm occurred
AnswerWait	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Waiting For Answer flag

- **EnableIn:** BOOL.
 This input enables the Add-On function processing. Set this input to 0 is equivalent to 'jump' the function code. The Add-On does not perform any function if its input 'Start' is enabled, so it is permissible to put FALSE (0) this input when the function does not work.

- **Start:** BOOL.
 This input signal runs the command specified in the input variable 'Command' (IN / OUT). When the function detects the status of 'Start ' at TRUE (1), it begins executing the specified command until it is completely processed. If 'Start' is set to FALSE (0) while the command is running (e.g. while waiting for response from the partner), then the execution is deleted. The execution of a command is not interruptible by acting on the input 'Start'.

Output 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
Rx_Buffer	InOut	DAD_RxTx			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Reception buffer (for partner answers)
Tx_Buffer	InOut	DAD_RxTx			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Transmission buffer (for commands to partner)
Command	InOut	Matrix_Command			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Command with parameters
Start	Input	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Function command start
EN	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Function working
DN	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Function successful executed
AL	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alarm active
AlarmCode	Output	INT	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Code of Alarm occurred
AnswerWait	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Waiting For Answer flag

- **EnableOut:** BOOL.
This output only copies the status of 'EnableIn' input, so it do not communicate any information about the processing status of the function.

- **EN:** BOOL.
This output only copies the state of the 'Start' input, so it communicates only that the execution of the function has been controlled, but not the eventual completion of the processing or interruption due to an alarm.

- **DN:** BOOL.
This output is set to TRUE (1) by the function itself when the execution of a command is successfully completed. The output status remains TRUE (1) until the 'Start' input is TRUE (1).

- **AL:** BOOL.
If during execution of a command a inconsistent situation occurs (e.g. the command given is not recognized by the function, or one or more of its parameters are invalid, or the response is not recognized, etc. ...) then the function stops and sets the output 'AL ' TRUE (1) to indicate the fault (alarm). The output status remains TRUE (1) until the 'Start' input is TRUE (1).

- **AlarmCode** INT.

This output (numerical) has been introduced to provide specific information about the status of a fault when it occurs. The table below lists all possible anomalies detected by the function, with a brief description.

The numerical value of output 'AlarmCode' when a problem occurs, is composed in order to show both the code of the command that was running and the fault event. In the decimal encoding value, the most significant digit (left) expresses the code of the running command in accordance with the following table (field IN/OUT), while the least significant (right) identifies the alarm in the table below.

When 'AlarmCode' = 0, then no anomaly was detected. If 'AlarmCode' takes on a nonzero value, this value will remain unchanged until the input 'Start' will be set TRUE (1). When 'Start' is FALSE (0), it is not possible (if the code is processed) to have 'AlarmCode' different than 0.

Value	Description
1	command refused from partner
2	Command refused from partner (in a multi-command string at 2nd string)
3	Command refused from partner (in a multi-command string at 3th string)
4	Unknow partner answer
5	Unknow command code
6	Bad [Data] parameter lenght
7	Bad [address] parameter
8	Bad [mode] parameter
9	Bad [Depth] parameter

- **AnswerWait** Bool.

Note that the execution of a command by this Add-On function is to send the partner a string that specifies an action to perform (with specification parameters), the expectation of the response from partner and its processing to evaluate partner's reaction.

Well, in the interval of time that elapses between the writing of the command and reading the response, the output 'AnswerWait' remains TRUE (1). Only when the response will be processed by the function, the output 'AnswerWait' will be set to FALSE (0). The waiting time for a response from the partner is undefined. To stop waiting, set to FALSE the input 'Start ' (this will set 'AnswerWait' to FALSE).

Input/output 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
Rx_Buffer	InOut	DAD_RxTx			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Reception buffer (for partner answers)
Tx_Buffer	InOut	DAD_RxTx			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Transmission buffer (for commands to partner)
Command	InOut	Matrix_Command			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Command with parameters
Start	Input	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Function command start
EN	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Function working
DN	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Function successful executed
AL	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alarm active
AlarmCode	Output	INT	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Code of Alarm occurred
AnswerWait	Output	BOOL	0	Decimal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Waiting For Answer flag

- **Rx_Buffer:** DAD_RxTx.
This variable refers to the string which will be read from the responses to commands generated by the partners. This string is shared with the Add-On 'DAD_DPD' that is responsible for receiving messages from the partners.

- **Tx_Buffer:** DAD_RxTx.
This variable refers to the string on which the commands for the communication partner are written. This string is shared with the Add-On 'DAD_DPD' that is meant to send messages (commands) to the partner.

- **Command:** Matrix_Command.
This variable is structured as above:

Name: Matrix_Command

Description: Command parameters

Members: Data Type Size: 32 byte(s)

Name	Data Type	Style	Description
Code	SINT	Decimal	Command code
Par	Matrix_Param		Command parameters
Address	SINT	ASCII	Address parameter
Chr1	SINT	ASCII	Chr1 parameter
Chr2	SINT	ASCII	Chr2 parameter
Depth	INT	Decimal	Depth parameter
Data	MatrixData		Data string parameter

It includes the 'Code' byte, which is the identifier of the command, and the structured variable 'Par' which includes all the parameters that complete the command to be given to the partner. This

variable therefore represents a real command, and processed by the function 'Matrix_Esc', it produces the command string interpretable by the Matrix device.

Below is a list of possible commands, identified by code, and the syntax of all the parameters associated with each command. The following tables are intended as a brief guide to the compilation of configuration commands. For their meaning, please refer to specific instructions in the "Host Programming Mode" manual supplied by Datalogic.

Note: Parameters not described, are not taken into account by the function 'Matrix_Esc', their value is therefore irrelevant.

List of commands

Start_Config				
				Command Code
				1
Parameter	Format	Range		Comment
		min.	max.	
Address	SINT	30 _h	4F _h	<i>If out of range: Bad [address] parameter' (Code 7)</i>
Chr1				---
Chr2				---
Depth				---
Data				---

SAVE_Config				
				Command Code
				2
Parameter	Format	Range		Comment
		min.	max.	
Address	SINT	30 _h	4F _h	<i>If out of range: Bad [address] parameter' (Code 7)</i>
Chr1	SINT	'!' or ' (space)		Rappresenta 'Mode' – se fuori range: 'Bad [mode] parameter'(8)
Chr2				---
Depth				---
Data				---

Stop_config				
				Command Code
				3
Parameter	Format	Range		Comment
		min.	max.	
Address	SINT	30 _h	4F _h	<i>If out of range: Bad [address] parameter' (Code 7)</i>
Chr1				---
Chr2				---
Depth				---
Data				---

SET_RestoreDefault				
				Command Code 4
Parameter	Format	Range		Comment
		min.	max.	
Address	SINT	30 _h	4F _h	<i>If out of range: Bad [address] parameter' (Code 7)</i>
Chr1				---
Chr2				---
Depth				---
Data				---

SET_DisableAllSymbologies				
				Command Code 5
Parameter	Format	Range		Comment
		min.	max.	
Address	SINT	30 _h	4F _h	<i>If out of range: Bad [address] parameter' (Code 7)</i>
Chr1				---
Chr2				---
Depth				---
Data				---

TOGGLE_ButtonFunction1				
				Command Code 6
Parameter	Format	Range		Comment
		min.	max.	
Address	SINT	30 _h	4F _h	<i>If out of range: Bad [address] parameter' (Code 7)</i>
Chr1				---
Chr2				---
Depth				---
Data				---

TOGGLE_ButtonFunction2				
				Command Code 7
Parameter	Format	Range		Comment
		min.	max.	
Address	SINT	30 _h	4F _h	<i>If out of range: Bad [address] parameter' (Code 7)</i>
Chr1				---
Chr2				---
Depth				---
Data				---

TOGGLE_ButtonFunction3				
				Command Code 8
Parameter	Format	Range		Comment
		min.	max.	
Address	SINT	30 _h	4F _h	<i>If out of range: Bad [address] parameter' (Code 7)</i>
Chr1				---
Chr2				---
Depth				---
Data				---

TOGGLE_ButtonFunction4				
				Command Code 9
Parameter	Format	Range		Comment
		min.	max.	
Address	SINT	30 _h	4F _h	<i>If out of range: 'Bad [address] parameter' (Code 7)</i>
Chr1				---
Chr2				---
Depth				---
Data				---
SET_Integer				
				Command Code 10
Parameter	Format	Range		Comment
		min.	max.	
Address	SINT	30 _h	4F _h	<i>If out of range: 'Bad [address] parameter' (Code 7)</i>
Chr1	SINT	<i>See command specifications</i>		
Chr2	SINT	<i>See command specifications</i>		
Depth	INT	0	---	<i>If out of range: 'Bad [Depth] parameter' (Code 9)</i>
Data	STRING	Max. 16		<i>If out of range: 'Bad [Data] parameter length' (Code 6)</i>
SET_Enumeration				
				Command Code 11
Parameter	Format	Range		Comment
		min.	Max.	
Address	SINT	30 _h	4F _h	<i>If out of range: 'Bad [address] parameter' (Code 7)</i>
Chr1	SINT	<i>See command specifications</i>		
Chr2	SINT	<i>See command specifications</i>		
Depth	INT	0	---	<i>If out of range: 'Bad [Depth] parameter' (Code 9)</i>
Data	STRING	Max. 4		<i>If out of range: 'Bad [Data] parameter length' (Code 6)</i>
SET_BitField				
				Command Code 12
Parameter	Format	Range		Comment
		min.	max.	
Address	SINT	30 _h	4F _h	<i>If out of range: 'Bad [address] parameter' (Code 7)</i>
Chr1	SINT	<i>See command specifications</i>		
Chr2	SINT	<i>See command specifications</i>		
Depth	INT	0	---	<i>If out of range: 'Bad [Depth] parameter' (Code 9)</i>
Data	STRING	Max. 16		<i>If out of range: 'Bad [Data] parameter length' (Code 6)</i>
SET_AtLeastOne				
				Command Code 13
Parameter	Format	Range		Comment
		min.	max.	
Address	SINT	30 _h	4F _h	<i>If out of range: 'Bad [address] parameter' (Code 7)</i>
Chr1	SINT	<i>See command specifications</i>		
Chr2	SINT	<i>See command specifications</i>		
Depth	INT	0	---	<i>If out of range: 'Bad [Depth] parameter' (Code 9)</i>
Data	STRING	Max. 16		<i>If out of range: 'Bad [Data] parameter length' (Code 6)</i>

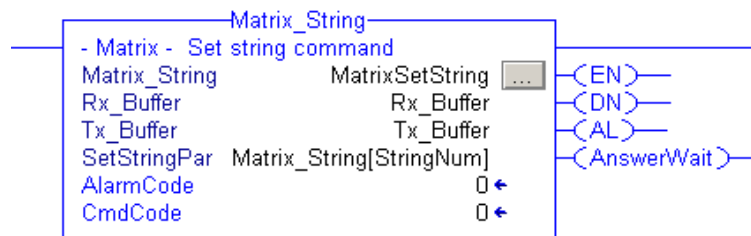
All commands code > 13 are mening 'Do nothing'

The Matrix_string Add-On

This function represents a completion of the Add-On 'Matrix_Esc'. It has been implemented in order to save system resources. In this function there is a specific control variable (of type 'Matrix_StringSet') that differs from that of 'Matrix_Esc' (type 'Matrix_Command') for only three details:

- The variable 'Data' in 'Matrix_Esc' now is called 'String' and it is a 256 char string ('Data' in 'Matrix_Esc' was a 16 char string)
- Does not have the variable 'Code' because this function only processes the command 'Set_string' (so you do not need to specify a command code).
- Structured variable of type 'Matrix_SetString' includes the 'SaveMode' parameter (*see below*).
-

The rules that apply to function 'Matrix_Esc' also apply to function 'Matrix_String'.



Datalogic device accepts one configuration command if the previous command 'Start_Config' has been given, and it assumes the final configuration to be completed after the command 'Stop_config'. Unlike 'Matrix_Esc', the function 'Matrix_String' automatically performs these functions, so it is not necessary to specify a sequence. In particular, this function executes the following command sequence:

1. 'Start_Config' to start configuration mode
2. 'Set_String' by using data specified into parameter 'String'
3. 'SAVE_Config' with the specified mode (parameter 'SaveMode')
4. 'Stop_config' to stop the configuration mode.

Therefore, this function can be used independently to run a complete sequence of configuration (only for the command 'Set_String') each time it runs.

The structure of the variable 'Matrix_StringSet' is as follows:

Name: Matrix_StringSet

Description: Command parameters

Members: Data Type Size: 268 byte(s)

Name	Data Type	Style	Description
Address	SINT	ASCII	
Chr1	SINT	ASCII	
Chr2	SINT	ASCII	
SaveMode	SINT	ASCII	Save mode: 'N' or 'n' = No save / 'Y' = RAM only / ' ' (space) = RAM + FLASH
Depth	INT	Decimal	
String	MatrixString		
LEN	DINT	Decimal	
DATA	SINT[256]	ASCII	

Below are listed the rules for compilation:

Parameter	Format	Range		Comment
		min.	max.	
Address	SINT	30 _h	4F _h	<i>Se fuori range:</i> Bad [address] parameter' (Code 7)
Chr1	SINT	<i>Vedere le specifiche del comando</i>		
Chr2	SINT	<i>Vedere le specifiche del comando</i>		
SaveMode	SINT	'!' or ' '(space)		<i>Se fuori range:</i> 'Bad [mode] parameter'(Code 8)
Depth	INT	0	---	<i>Se fuori range:</i> 'Bad [Depth] parameter' (Code 9)
Data	String	Max. 256		<i>Se fuori range:</i> 'Bad [Data] parameter length' (Code 6)

'AlarmCode' output variable

This variable takes on particular significance since the function 'Matrix_String' does in fact multiple commands in a row.

This output (numerical) has been introduced to provide specific information about the status of a fault when it occurs. The table below lists all possible anomalies detected by the function, with a brief description.

The numerical value of output 'AlarmCode' when a problem occurs, is composed in order to show both the code of the command that was running and the fault event. In the decimal encoding value, the most significant digit (left) expresses the code of the running command in accordance with the following table, while the least significant (right) identifies the alarm in the table below. When 'AlarmCode' = 0, then no anomaly was detected. If 'AlarmCode' takes on a nonzero value, this value will remain unchanged until the input 'Start' will be set TRUE (1). When 'Start' is FALSE (0), it is not possible (if the code is processed) to have 'AlarmCode' different than 0.

Command Code	
Value	Description
1	START_Config (1St command string)
2	START_Config (2Nd command string)
3	SET_String
4	SET_String (2nd command string)
5	SAVE_Config
6	STOP_Config

AlarmCode	
Value	Description
1	command refused from partner
2	Command refused from partner (in a multi-command string at 2nd string)
3	Command refused from partner (in a multi-command string at 3th string)
4	Unknow partner answer
5	---
6	Bad [Data] parameter lenght
7	Bad [address] parameter
8	Bad [mode] parameter
9	Bad [Depth] parameter