

Vacuum-Automation

Function Block-Documentation

Table of Contents

1	Function block “FB_SCTMi_IOL“	3
1.1	Image of function block	3
1.2	Brief description.....	4
2	Parameters	5
2.1	Input	5
2.2	Output.....	7
2.3	Additional information.....	9
2.3.1	Structur „stSCTMi_Ejector“	9
3	Appendix.....	10
3.1	List of abbreviations	10
3.2	List of figures	10
3.3	Note.....	10



figure 1: example SCTMi

1 Function block “FB_SCTMi_IOL“

1.1 Image of function block

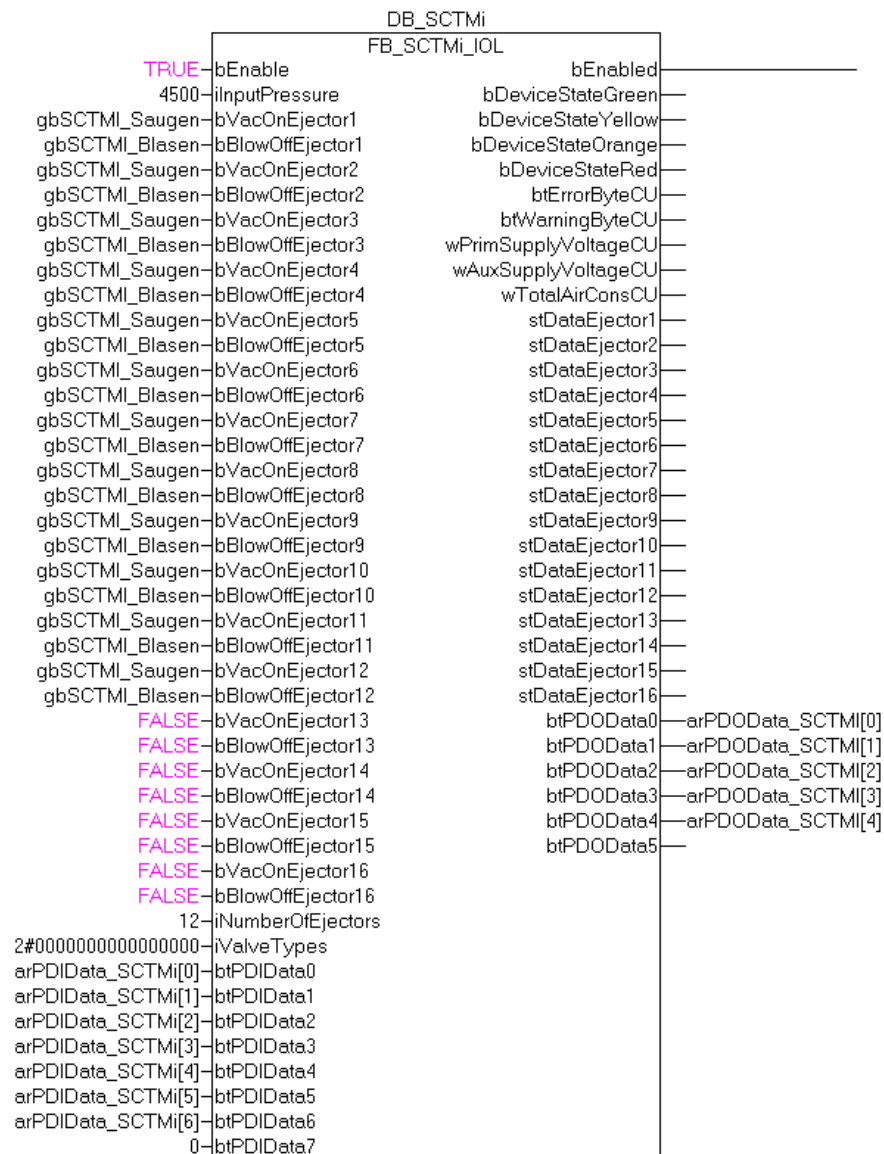


figure 2: example of function block

1.2 Brief description

This function block regulates data communication for following vacuum terminals:

- SCTMi IO-Link 1.1 (with 2- 16 ejectors)

Following jobs are carried out by the function block:

- Querying the epc data for each ejector and control unit.
- Mapping of inputs and outputs to bits of process data.

2 Parameters

2.1 Input

TC2, TC3, S7, TIA

name	data type	description
bEnable	BOOL	Activates the function block
iInputPressure	INT	With this parameter, the terminal gets the input pressure in mbar to be able to create EPC analysis.
bVacOnEjector1 ... bVacOnEjector16	BOOL	Request for suction of the corresponding ejector
bBlowOffEjector1 ... bBlowOffEjector16	BOOL	Request for blow-off of the corresponding ejector
iNumberOfEjectors	INT	The number of how many ejectors are operated with the terminal. (2-16)
iValveTypes	INT	This parameter specifies the valve type. (NO = Normally Open or NC = Normally Closed) The value should be given in binary notation. So, for each ejector the valve type can be specified (NO = 1 and NC = 0). For example we have a terminal with 16 ejectors, whereas ejectors 1 to 8 have NO valves and ejectors 9 to 16 have NC valves. The value that should be written, would be: 2 # 0000000011111111
btPDIData0	BYTE	Process data byte 0 from SCTMi
btPDIData1	BYTE	Process data byte 1 from SCTMi
btPDIData2	BYTE	Process data byte 2 from SCTMi
btPDIData3	BYTE	Process data byte 3 from SCTMi
btPDIData4	BYTE	Process data byte 4 from SCTMi
btPDIData5	BYTE	Process data byte 5 from SCTMi
btPDIData6	BYTE	Process data byte 6 from SCTMi
btPDIData7	BYTE	Process data byte 7 from SCTMi

AB

name	data type	description
EnableIn	BOOL	Activates the function block
ilInputPressure	INT	With this parameter, the terminal gets the input pressure in mbar to be able to create EPC analysis.
bVacOnEjector1 ...	BOOL	Request for suction of the corresponding ejector
bVacOnEjector16		
bBlowOffEjector1 ...	BOOL	Request for blow-off of the corresponding ejector
bBlowOffEjector16		
iNumberOfEjectors	INT	The number of how many ejectors are operated with the terminal. (2-16)
iValveTypes	int	This parameter specifies the valve type. (NO = Normaly Open or NC = Normaly Closed) The value should be given in binary notation. So, for each ejector the valve type can be specified (NO = 1 and NC = 0). For example we have a terminal with 16 ejectors, whereas ejectors 1 to 8 have NO valves and ejectors 9 to 16 have NC valves. The value that should be written, would be: 2 # 0000000011111111
siPDIData0	SINT	Process data byte 0 from SCTMi
siPDIData1	SINT	Process data byte 1 from SCTMi
siPDIData2	SINT	Process data byte 2 from SCTMi
siPDIData3	SINT	Process data byte 3 from SCTMi
siPDIData4	SINT	Process data byte 4 from SCTMi
siPDIData5	SINT	Process data byte 5 from SCTMi
siPDIData6	SINT	Process data byte 6 from SCTMi
siPDIData7	SINT	Process data byte 7 from SCTMi
IN/OUT Parameter:		
stDataEjector1To16	ARRAY [0..15] OF stSCTMi_Ejector	This In / OUT parameter returns a structure for each ejector. In this structure, all relevant data of each ejector are stored. In chapter 2.3 the structure is described in more detail.

2.2 Output

TC2, TC3, S7, TIA

name	data type	description
bEnabled	BOOL	Feedback about status of Enable
bDeviceStateGreen	BOOL	Status of the terminal is green
bDeviceStateYellow	BOOL	Status of the terminal is yellow
bDeviceStateOrange	BOOL	Status of the terminal is orange
bDeviceStateRed	BOOL	Status of the terminal is red
btErrorByteCU	BYTE	Current error code (see operating manual)
btWarningByteCU	BYTE	Current warnings code (see operating manual)
wPrimSupplyVoltageCU	WORD	Current primary voltage (V)
wAuxSupplyVoltageCU	WORD	Current auxillary voltage (V)
wTotalAirConsCU	WORD	Total air consumption of the last handling cycle (0.1NL)
stDataEjector1 ... stDataEjector16	stSCTMi_Ejector	These outputs returns a structure for each ejector. In this structure, all relevant data of each ejector are stored. In chapter 2.3 the structure is described in more detail.
btPDOData0	BYTE	Processdata byte 0 to SCTMi
btPDOData1	BYTE	Processdata byte 1 to SCTMi
btPDOData2	BYTE	Processdata byte 2 to SCTMi
btPDOData3	BYTE	Processdata byte 3 to SCTMi
btPDOData4	BYTE	Processdata byte 4 to SCTMi
btPDOData5	BYTE	Processdata byte 5 to SCTMi

AB

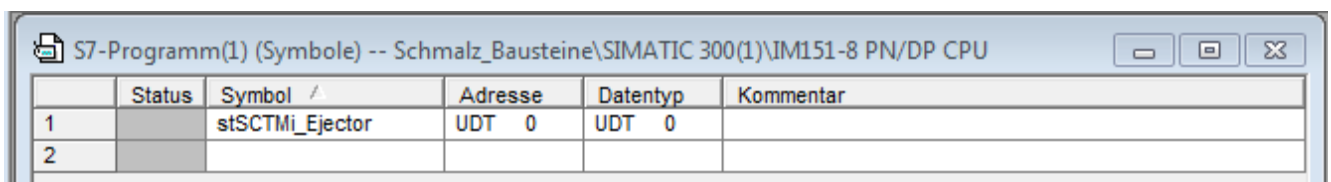
name	data type	description
EnableOut		Feedback about status of Enable
bDeviceStateGreen	BOOL	Status of the terminal is green
bDeviceStateYellow	BOOL	Status of the terminal is yellow
bDeviceStateOrange	BOOL	Status of the terminal is orange
bDeviceStateRed	BOOL	Status of the terminal is red
siErrorByteCU	SINT	Current error code (see operating manual)
siWarningByteCU	SINT	Current warnings code (see operating manual)
iPrimSupplyVoltageCU	INT	Current primary voltage (V)
iAuxSupplyVoltageCU	INT	Current auxillary voltage (V)
iTotalAirConsCU	INT	Total air consumption of the last handling cycle (0.1NL)
siPDOData0	SINT	Processdata byte 0 to SCTMi
siPDOData1	SINT	Processdata byte 1 to SCTMi
siPDOData2	SINT	Processdata byte 2 to SCTMi
siPDOData3	SINT	Processdata byte 3 to SCTMi
siPDOData4	SINT	Processdata byte 4 to SCTMi
siPDOData5	SINT	Processdata byte 5 to SCTMi

2.3 Additional information

In addition to the function block, the structure "stSCTMi_Ejector" must also be imported into the respective plc system. Here you can proceed exactly in the same way as when importing a function block. As already mentioned in chapter 2.2, the block returns a structure for each ejector which contains all relevant data. Without importing / creating the structure, errors will occur when the plc program is compiled.

2.3.1 Additional notes for Siemens S7

To compile the source of the structure in Siemens S7 a data type of type UDT must be created in the symbol table. This data type must be assigned to the name of the structure ("stSCTMi_Ejector"). The following picture shows how the corresponding line in the symbol table should look like.



Status	Symbol	Adresse	Datentyp	Kommentar
1	stSCTMi_Ejector	UDT 0	UDT 0	
2				

If the symbol table has been adapted, both sources (structure and function block) can be compiled without errors.

2.3.2 Structur „stSCTMi_Ejector“

TC2, TC3, S7, TIA

name	data type	description
btErrorByte	BYTE	Error Code of the ejector
btWarningByte	BYTE	Code for pending warnings of the ejector
btLeakageOfLastCycle	BYTE	Measured leakage of the last handling cycle (mbar/s)
wSystemVacuum	WORD	Current vacuum value of the ejector (mbar)
wEvacuationTimeT1	WORD	Measured evacuation time T1 (ms)
wLastFreeFlowVacuum	WORD	Measured free flow vacuum (mbar)
wAirConsumptionOfLastCyc	WORD	Air consumption of the last handling cycle (0.1NL)
bAirSavingFunktion_H1	BOOL	Status of H1
bPartPresent_H2	BOOL	Status H2

AB

name	data type	description
siErrorByte	SINT	Error Code of the ejector
siWarningByte	SINT	Code for pending warnings of the ejector
siLeakageOfLastCycle	SINT	Measured leakage of the last handling cycle (mbar/s)
iSystemVacuum	INT	Current vacuum value of the ejector (mbar)
iEvacuationTimeT1	INT	Measured evacuation time T1 (ms)
iLastFreeFlowVacuum	INT	Measured free flow vacuum (mbar)
iAirConsumptionOfLastCyc	INT	Air consumption of the last handling cycle (0.1NL)
bAirSavingFunktion_H1	BOOL	Status of H1
bPartPresent_H2	BOOL	Status H2



3 Appendix

3.1 List of abbreviations

abbreviation	description
TC2	Beckhoff TwinCAT 2
TC3	Beckhoff TwinCAT 3
S7	Siemens Step 7
TIA	Siemens Step 7 TIA
AB	Allen Bradley
FB	Function module
EPC	Energy- and Process control
CM	Condition Monitoring
EM	Energy Monitoring
PM	Predictive Maintenance

3.2 List of figures

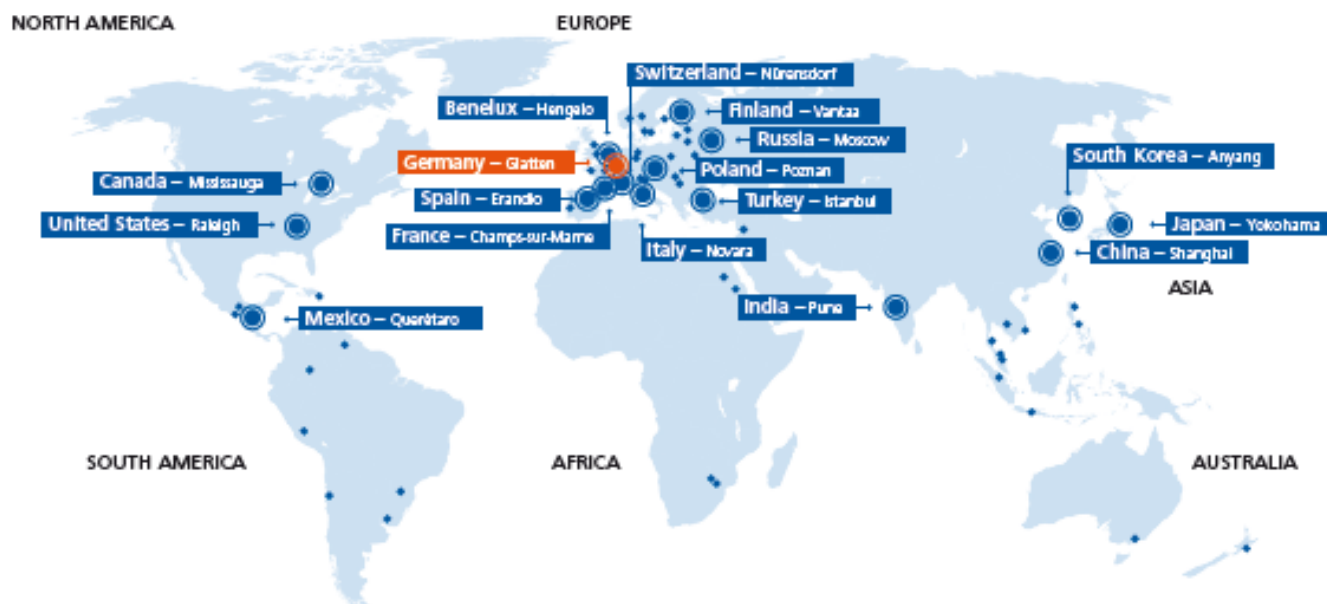
figure 1: example SCTMi2

figure 2: example of function block3

3.3 Note

The byte order of the product is represented as big endian.

At your side worldwide



Subsidiaries

Schmalz Benelux – Hengelo (NL)
 Schmalz Canada – Mississauga
 Schmalz China – Shanghai
 Schmalz Finland – Vantaa
 Schmalz France – Champs-sur-Marne
 Schmalz India – Pune
 Schmalz Italy – Novara
 Schmalz Japan – Yokohama

Schmalz Mexico – Querétaro
 Schmalz Poland – Suchy Las (Poznań)
 Schmalz Russia – Moscow
 Schmalz South Korea – Anyang
 Schmalz Spain – Erandio (Vizcaya)
 Schmalz Switzerland – Nürensdorf
 Schmalz Turkey – Istanbul
 Schmalz United States – Raleigh (NC)

Headquarters

Schmalz Germany – Glatten

Sales Partners

You can find the Schmalz sales partner in your country at: WWW.SCHMALZ.COM/SALESNETWORK

Vacuum Automation

T: +49 7443 2403-102

Handling Systems

T: +49 7443 2403-108

J. Schmalz GmbH
 Johannes-Schmalz-Str. 1
 72293 Glatten, Germany
 T: +49 7443 2403-0
schmalz@schmalz.de
WWW.SCHMALZ.COM