

SDI-12 Master to MODBUS Slave Converter

The TBS09DR is a converter to connect MODBUS sensors to an SDI-12 master device such as a data logger or telemetry unit. It can connect multiple MODBUS sensors in parallel by setting the corresponding address of the individual MODBUS sensors using an extended SDI-12 command, upfront to issuing the measurement commands.

The TBS09DR provides a rich command set to offer maximum flexibility for configuring the device to a sensor with Modbus interface.



TBS09 SDI-12 Master to MODBUS Slave Converter

Features

- SDI-12 Master to MODBUS Slave Converter
- Multiple sensors can be connected
- SDI-12 Standard V1.3
- Highly configurable
- Switched sensor supply voltage output
- 6 - 16V supply voltage

- 7mA current consumption when active
- Less than 100µA idle current
- Operating Temperature Range:
- 40°C ... + 80°C

Target Applications

- SDI-12 sensor networks

SDI 12 Master to MODBUS Slave Converter

Contents

1	INTRODUCTION	3
2	PRODUCT SPECIFICATION	3
3	CALIBRATION AND SETTINGS	4
4	CONNECTIONS	5
5	SDI-12 CONFIGURATION	6
5.1	OVERVIEW	6
5.2	CONFIGURATION TOOL	7
5.3	TBS09 SDI-12 MODES	7
5.4	SDI-12 VIRTUAL ADDRESSING	7
5.5	LIMITATIONS	8
6	CONFIGURATION OF SDI-12 TO MODBUS COMMUNICATION	9
6.1	EXTENDED SDI-12 COMMANDS LOOK-UP TABLE	9
6.2	HW CONFIGURATION	10
6.2.1	Operation modes	10
6.2.2	RS485 bus configuration	11
6.2.3	Power mode	13
6.2.4	Warm-up time	14
6.3	VIRTUAL MAPPING OF MODBUS SENSORS TO SDI-12	14
6.3.1	Assigning a virtual SDI-12 address and identification to a MODBUS sensor	15
6.3.2	Managing the MODBUS / virtual SDI-12 sensors configuration list	16
6.4	CONFIGURING SDI-12 MEASUREMENT COMMANDS	17
6.4.1	Mapping MODBUS sensor function code to SDI-12 measurement command	17
6.4.2	Read and write operations	21
6.4.3	Measurements conversion	26
6.4.4	MODBUS sensors wake-up	28
7	TBS09 CONFIGURATION AND COMMUNICATION EXAMPLES	31
8	COMMUNICATION PROTOCOLS	35
8.1	SDI-12	35
8.2	MODBUS	36
9	MECHANICAL INFORMATION	36
10	ENVIRONMENTAL SPECIFICATION	36
11	ORDERING INFORMATION	36
12	HISTORY	36

SDI 12 Master to MODBUS Slave Converter

1 Introduction

The TBS09DR is a converter to connect one or multiple Modbus sensors to an SDI-12 master device such as a data logger or telemetry unit.

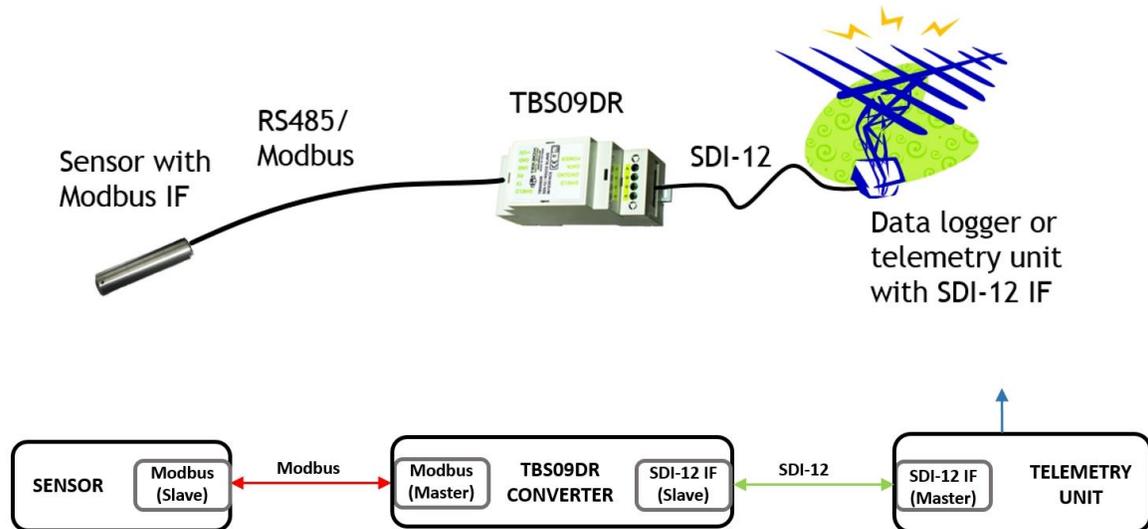


Figure 1 – TBS09DR adding a Modbus interface to a telemetry system with SDI-12 interface

2 Product specification

APPLICABLE FW: 18.00.01.01 and above

- Application: converter used to interface MODBUS master devices (e.g., RTU) with SDI-12 slave devices (e.g., sensors)
 - The converter embeds MODBUS master and SDI-12 slave modules
 - Maximum of 4 MODBUS sensors connected to TBS09
- SDI-12 compatibility:
 - Version: v1.3
- MODBUS compatibility:
 - Protocol: MODBUS RTU - Supported function codes by TBS09 MODBUS master:
 - 0x01 (01) – Read Coils
 - 0x02 (02) – Read Discrete Inputs
 - 0x03 (03) – Read Holding Registers
 - 0x04 (04) – Read Input Registers
 - 0x05 (05) – Write Single Coil
 - 0x06 (06) – Write Single Register
 - 0x0F (15) – Write Multiple Coils
 - 0x10 (16) – Write Multiple Registers
 - 0x17 (23) – Read/Write Multiple Registers
 - RS485 configuration:

SDI 12 Master to MODBUS Slave Converter

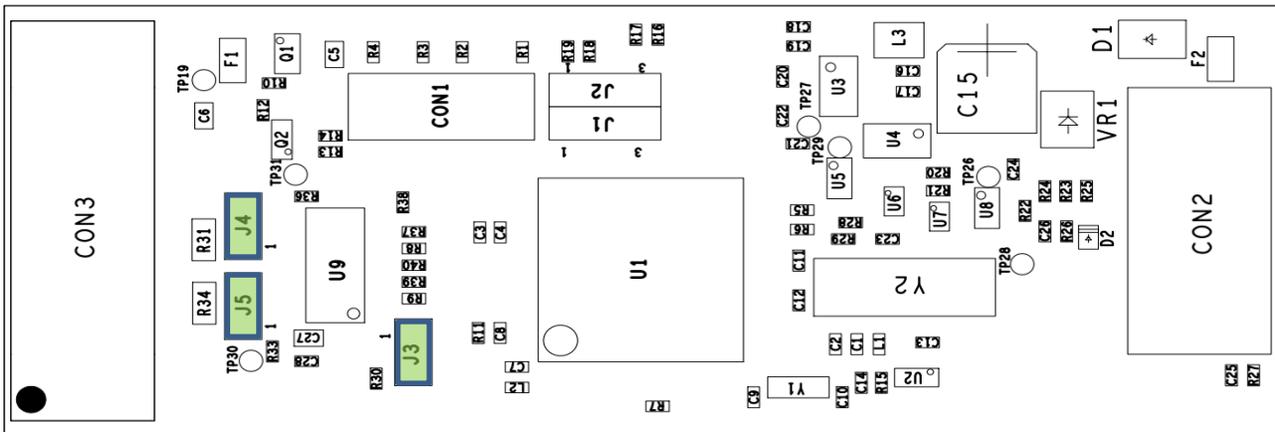
- Baud rate: 4800, 9600, 14400, 19200, 28800 and 38400
- Data length: 7 or 8 bits
- Parity: odd, even, none
- Stop bits: 1 or 2 bits
 - Half or full duplex
 - 120 Ω termination (configurable)
- Supply voltage: 5 – 16 V
- Power consumption
 - Active mode: 4 mA + connected SDI-12 sensors power consumption
 - Idle mode: 20µA
- Form factor:
 - DIN RAIL
 - FIBOX (IP67 housing suitable for outdoor use)

3 Calibration and settings

TBS09 doesn't require any calibration.

It comes factory-configured to operate by default in half duplex with termination.

This configuration can be changed by the user by setting related jumpers J3/J4/J5 after lifting the housing:



Communication mode:

Configuration	J3	Default configuration
Half Duplex	✓	✓
Full Duplex	x	

SDI 12 Master to MODBUS Slave Converter

MODBUS termination:

Configuration	J4 – J5	Default configuration
120 Ω termination	✓	✓
No termination	x	

4 Connections

TBS09 provides one 4 slots connector (SDI-12) and one 6 slots connector (RS485):



SDI-12 terminal assignment, from top to bottom:

Slot name	Description	Comment
Shield	Cable shield	Connect to sensor's cable shield
Ground	Ground	Connect to ground
Data	SDI-12 data line	Connect to SDI-12 sensor data line
Power	TBS09 supply voltage input	Connect to +12V external power supply

RS485 terminal assignment, from top to bottom:

Slot name	Description	Comment
T+	TXD+ output	Connect to MODBUS master RXD+
T-	TXD- output	Connect to MODBUS master RXD-
R+	RXD+ input	Connect to MODBUS master TXD+ (full duplex operation only – must be left)

SDI 12 Master to MODBUS Slave Converter

		<i>unconnected in half duplex)</i>
R-	RXD- input	Connect to MODBUS master TXD- (full duplex operation only – must be left unconnected in half duplex))
G	Ground	Connect to ground
P	TBS09 power supply output used to supply connected MODBUS sensors. Output voltage matching with TBS09 power input voltage.	Connect to MODBUS sensors power input. Leave unconnected if an external power supply is used to power MODBUS sensors (ensure grounds are tied to each other's)

5 SDI-12 configuration

5.1 Overview

TBS09 shall be seen as a fully compatible SDI-12 v1.3 sensor that is able to respond to following commands:

- Identification (a!)
- Query Address (?!)
- Acknowledge Active (a!)
- Change Address (aAb!)

TBS09 response to identification command a!: a13TEKBOXVNTBS09rvA 1234561**800100**

Where:

- 13: SDI-12 standard v1.3
- TEKBOXVN: manufacturer, Tekbox Vietnam
- TBS09: product name
- rvA: HW revision A
- 123456: serial number
- **1800100**: FW version 18.0.01.00

TBS09 default SDI-12 address is set to 0 and can be changed with SDI-12 Change Address command to any supported SDI-12 address:

- 0-9
- A-Z
- a-z

Unlike a regular SDI-12 sensor, TBS09 support of SDI-12 measurement commands (aMx!, aMCx!, aCx!, aCCx!, aRx!, aRCx! and aV!) is flexible and fully configurable through extended SDI-12 commands.

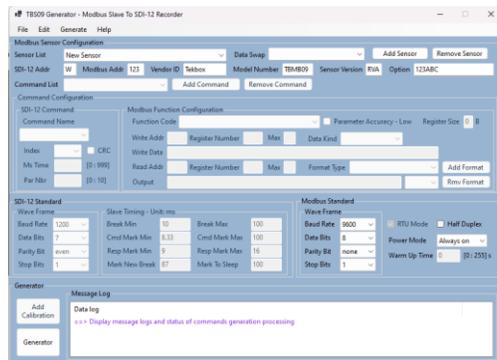
SDI 12 Master to MODBUS Slave Converter

This configuration is totally dependent on the use case and depends on the MODBUS sensors connected to TBS09. As the module is a bridge between both protocols, the binding of MODBUS function codes and SDI-12 measurement commands is performed through SDI-12 extended commands.

5.2 Configuration tool

A configuration PC application is also provided to ease the generation of these commands.

Refer to the configuration tool embedded user manual for further information.



5.3 TBS09 SDI-12 modes

TBS09 operates in 2 different SDI-12 modes, one used for configuration and one used for interacting with connected MODBUS sensors:

- Configuration mode:
 - Communication between SDI-12 data recorder and TBS09
 - Used to send required extended SDI-12 commands to configure the SDI-12 to MODBUS communication and other internal parameters.
- Measurement mode:
 - Communication between SDI-12 data recorder and virtual SDI-12 sensors (MODBUS sensors which have been virtually remapped as SDI-12 sensors)
 - Data recorder sends measurements SDI-12 commands to retrieve measurement values from the virtual SDI-12 sensors (i.e., the physical MODBUS sensors)

5.4 SDI-12 virtual addressing

From the SDI-12 master device perspective, TBS09 is an SDI-12 sensor with a defined slave address.

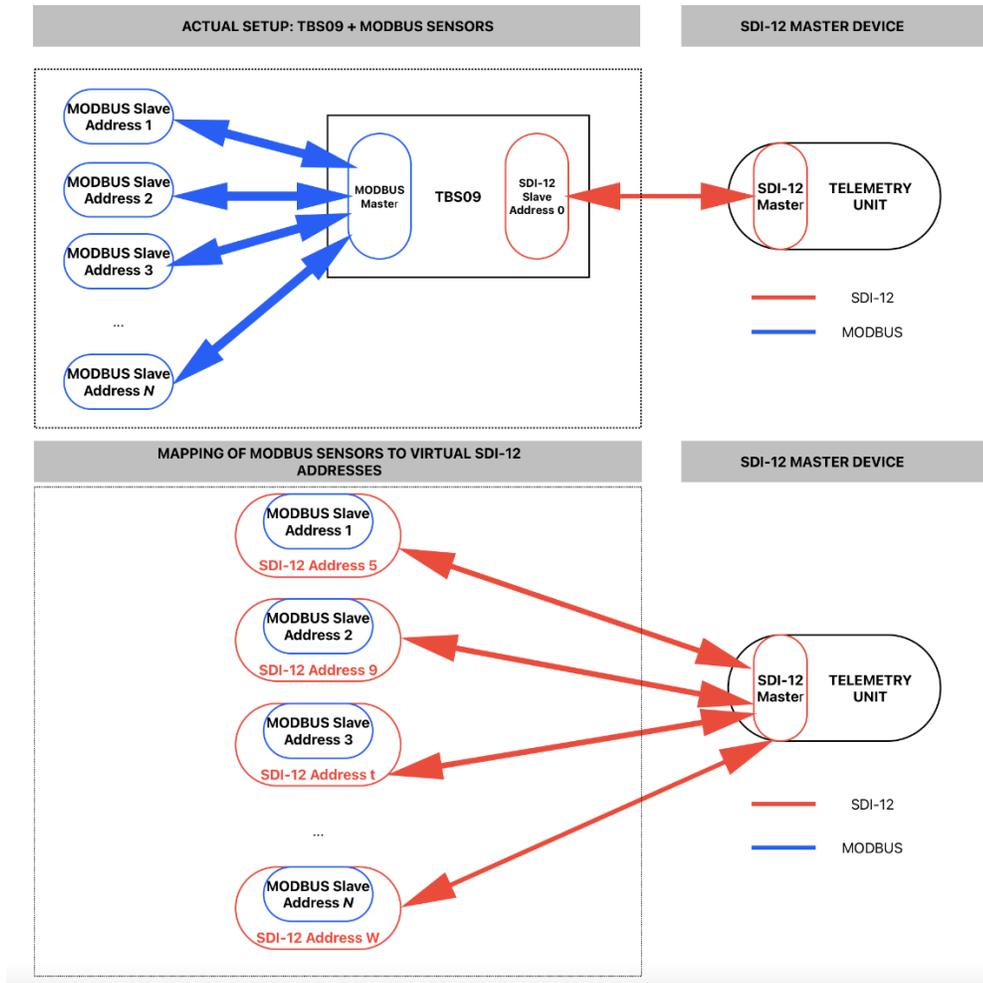
While in configuration mode, the data recorder needs to send a set of extended SDI-12 commands to map MODBUS sensors addresses to virtual SDI-12 addresses.

This process results in turning MODBUS sensors into virtual SDI-12 sensors, each with a specific and unique SDI-12 address.

When switching TBS09 to measurement mode, the data recorder can send SDI-12 measurement commands to each remapped MODBUS sensor and collect the measurement results through aDx! command like any SDI-12 sensor.

The following diagram highlights at the upper side of the diagram the actual setup of TBS09, bridging MODBUS sensors with the SDI-12 data recorder (TBS09 in configuration mode) and at the bottom side the data recorder communicating with the virtual SDI-12 sensors (TBS09 in measurement mode).

SDI 12 Master to MODBUS Slave Converter



For instance, MODBUS sensor with address 2 is seen as an SDI-12 sensor with SDI-12 address 9 by the data recorder.

5.5 Limitations

The following limitations apply to TBS09:

- TBS09 SDI-12 address must be different from those used for the virtual SDI-12 sensors
- Concurrent measurement commands aCx! are supported however they deviate from the SDI-12 standard by being sequentially executed.
- Only 1 aR(C)x! command can be assigned to each MODBUS sensors
- Each MODBUS sensors can be assigned up to 6 different SDI-12 commands, each of them returning up to 9 measurement values.
- The maximum data size that can be read at once from a MODBUS sensor is limited to 40 bytes.
- Up to 4 MODBUS sensors can be connected to TBS09

SDI 12 Master to MODBUS Slave Converter

6 Configuration of SDI-12 to MODBUS communication

In the following tables, parameter **a** preceding any extended SDI-12 command represents the SDI-12 address.

6.1 Extended SDI-12 commands look-up table

Category	Commands	Description
Operation mode	XSTM (Table 2)	Switch to MODBUS mode (Measurement mode)
	XSPM (Table 3)	Stop MODBUS mode (configuration mode)
RS485 configuration	XSMF (Table 4)	Configure MODBUS RS485 bus
	XGMF (Table 5)	Get MODBUS RS485 bus configuration
	XSMI (Table 6)	RS485 mode (half/full duplex)
	XGMI (Table 7)	Get RS485 mode
Power	XSMP (Table 8)	Set TBS09 power mode
	XGMP (Table 9)	Get TBS09 power mode
	XSPW (Table 10)	Set TBS09 warm-up time
	XGPW (Table 11)	Get TBS09 warm-up time
Virtual SDI-12 sensors address mapping	XASS (Table 12)	Add virtual SDI-12 sensor
	XGSS (Table 13)	Get virtual SDI-12 sensor mapping
	XRSS (Table 14)	Remove virtual SDI-12 sensor
	XCAS (Table 15)	Clear virtual SDI-12 sensors list
	XLAS (Table 16)	List all virtual SDI-12 sensors mapping
SDI-12 command / MODBUS function code mapping	XAMC (Table 17)	Assign SDI-12 command to a MODBUS sensor function code
	XGMC (Table 18)	Retrieve MODBUS sensor function code/SDI-12 command mapping
	XRMC (Table 19)	Remove MODBUS sensor function code/SDI-12 command assignment
	XCAC (Table 20)	Clear all MODBUS sensor function code/SDI-12 command assignments
	XGCN (Table 21)	Get the number of SDI-12 commands configured for a MODBUS sensor
MODBUS sensor read/write operations over SDI-12	XSRD (Table 22)	Configure MODBUS read operations
	XGRD (Table 23)	Retrieve MODBUS read configuration

SDI 12 Master to MODBUS Slave Converter

	XSWD (Table 24)	Configure MODBUS write operations
	XGWD (Table 25)	Retrieve MODBUS write configuration
	XSIC (Table 26)	Direct MODBUS request
	XGIR (Table 27)	Retrieve MODBUS response
Parameters conversion	XAPC (Table 28)	Assign a conversion method
	XSMT (Table 29)	Set coefficients of a conversion method
	XGPC (Table 30)	Get conversion method parameters
	XRPC (Table 31)	Remove a conversion method
	XGMT (Table 32)	Get coefficients of a conversion method
MODBUS sensors wake-up	XSWM (Table 33)	Set MODBUS wake-up mode
	XGWM (Table 34)	Get MODBUS wake-up mode
	XAWC (Table 35)	Add MODBUS wake-up command
	XRWC (Table 36)	Remove MODBUS wake-up command
	XGWC (Table 37)	Get MODBUS wake-up command

Table 1 Extended SDI-12 commands look-up table

6.2 HW configuration

6.2.1 Operation modes

To proceed with TBS09 configuration, it is required to have it set in configuration mode using the extended SDI-12 command **aXSTM!**.

Command format	Description	
aXSTM!	Start MODBUS mode, TBS09 in SDI-12 measurement mode.	
Parameters description	None	
Parameters values	None	
Response format	aOK<CR><LF>	aFAIL<CR><LF>
Description	Success	Fail
SDI-12 Command Example	aXSTM! aOK<CR><LF>	

Table 2 Extended SDI-12 command aXSTM

SDI 12 Master to MODBUS Slave Converter

Command format	Description	
aXSPM!	Stop MODBUS mode, TBS09 in SDI-12 configuration mode.	
Parameters description	None	
Parameters values	None	
Response format	aOK<CR><LF>	aFAIL<CR><LF>
Description	Success	Fail
SDI-12 Command Example		
aXSPM! aOK<CR><LF>		

Table 3 Extended SDI-12 command aXSPM

6.2.2 RS485 bus configuration

RS485 bus must be correctly configured to ensure the MODBUS communication with the MODBUS sensors is functional. Obviously, all connected MODBUS sensors must be compliant with the programmed configuration so it's important to double check parameters like baud rate and parity.

Command format	Description			
aXSMF,Baud,Data,Parity,Stop!	Configure RS485 parameters: baud rate, data length, parity and stop bits			
Parameters description	Baud Baud rate	Data Number of data bits	Parity Parity bit	Stop Number of stop bits
Parameters values	<ul style="list-style-type: none"> • 4800 • 9600 • 14400 • 19200 • 28800 • 38400 	<ul style="list-style-type: none"> • 7 • 8 	<ul style="list-style-type: none"> • 0 (Odd) • 1 (Even) • 2 (None) 	<ul style="list-style-type: none"> • 1 • 2
Response format	aOK<CR><LF>		aFAIL<CR><LF>	
Description	Success		Fail	
SDI-12 Command Example				
aXSMF,38400,8,0,1! aOK<CR><LF>				

Table 4 Extended SDI-12 command aXSMF

Command format	Description	
aXGMF!	Get RS485 parameters: baud rate, data length, parity and stop bits	
Parameters description	None	
Parameters values	None	
Response format	a Baud,Data,Parity,Stop <CR><LF>	aFAIL<CR><LF>
Description	Success	Fail

SDI 12 Master to MODBUS Slave Converter

Parameters description	Baud Baud rate	Data Number of data bits	Parity Parity bit	Stop Number of stop bits
Parameters values	<ul style="list-style-type: none"> • 4800 • 9600 • 14400 • 19200 • 28800 • 38400 	<ul style="list-style-type: none"> • 7 • 8 	<ul style="list-style-type: none"> • 0 (Odd) • 1 (Even) • 2 (None) 	<ul style="list-style-type: none"> • 1 • 2
SDI-12 Command Example	aXGMF! a38400,8,0,1<CR><LF>			

Table 5 Extended SDI-12 command aXGMF

Command format	Description	
aXSMI, BusMode!	Configure RS485 in half or full duplex	
Parameters description	BusMode Half or full duplex communication	
Parameters values	<ul style="list-style-type: none"> • 0 (half duplex) • 1 (full duplex) 	
Response format	aOK<CR><LF>	aFAIL<CR><LF>
Description	Success	Fail
SDI-12 Command Example	aXSMI, 1! aOK<CR><LF>	

Table 6 Extended SDI-12 command aXSMI

Command format	Description	
aXGMI!	Get RS485 communication mode	
Parameters description	None	
Parameters values	None	
Response format	aBusMode<CR><LF>	aFAIL<CR><LF>
Description	Success	Fail
Parameters description	BusMode Half or full duplex communication	
Parameters values	<ul style="list-style-type: none"> • 0 (half duplex) • 1 (full duplex) 	
SDI-12 Command Example	aXGMI! a1<CR><LF>	

Table 7 Extended SDI-12 command aXGMI

SDI 12 Master to MODBUS Slave Converter

6.2.3 Power mode

These extended commands control how TBS09 supplies connected MODBUS sensors.

Command format	Description	
aXSMP,PowerMode!	Set TBS09 MODBUS power mode	
Parameters description	PowerMode TBS09 MODBUS power mode	
Parameters values	<ul style="list-style-type: none"> • 0 (TBS09 continuously delivers power to MODBUS sensors) • 1 (TBS09 delivers power to MODBUS sensors only when measuring) • 2 (TBS09 does not deliver any power – MODBUS sensors externally powered) 	
Response format	aOK<CR><LF>	aFAIL<CR><LF>
Description	Success	Fail
SDI-12 Command Example		
aXSMP,1! aOK<CR><LF>		

Table 8 Extended SDI-12 command aXSMP

Command format	Description	
aXGMP!	Get TBS09 MODBUS power mode	
Parameters description	None	
Parameters values	None	
Response format	a PowerMode <CR><LF>	aFAIL<CR><LF>
Description	Success	Fail
Parameters description	PowerMode TBS09 MODBUS power mode	
Parameters values	<ul style="list-style-type: none"> • 0 (TBS09 continuously delivers power to MODBUS sensors) • 1 (TBS09 delivers power to MODBUS sensors only when measuring, automatically turned off after receiving SDI-12 data command or after executing continuous measurement command) • 2 (TBS09 does not deliver any power – MODBUS sensors externally powered) 	
SDI-12 Command Example		
aXGMP! a1<CR><LF>		

Table 9 Extended SDI-12 command aXGMP

SDI 12 Master to MODBUS Slave Converter

6.2.4 Warm-up time

Some MODBUS sensors require a warm-up time before any measurement can be performed.

Below command allows setting such warm-up time which is globally applied to all connected MODBUS sensors even though some of them would not require it.

Command format	Description	
aXSPW, WarmUpTime!	Set TBS09 MODBUS warm-up time	
Parameters description	WarmUpTime TBS09 MODBUS warm-up time	
Parameters values	<ul style="list-style-type: none"> • 0 (no warm-up time) • 1 – 255 (warm-up time in seconds, applied to all MODBUS sensors) 	
Response format	aOK<CR><LF>	aFAIL<CR><LF>
Description	Success	Fail
SDI-12 Command Example		
aXSMP,5! aOK<CR><LF>		

Table 10 Extended SDI-12 command aXSPW

Command format	Description	
aXGPW!	Get MODBUS sensors warm-up time	
Parameters description	None	
Parameters values	None	
Response format	a WarmUpTime <CR><LF>	aFAIL<CR><LF>
Description	Success	Fail
Parameters description	WarmUpTime MODBUS sensors warm-up time	
Parameters values	<ul style="list-style-type: none"> • 0 (no warm-up time) • 1 – 255 (warm-up time in seconds, applied to all MODBUS sensors) 	
SDI-12 Command Example		
aXGPW! a5<CR><LF>		

Table 11 Extended SDI-12 command aXGPW

6.3 Virtual mapping of MODBUS sensors to SDI-12

Each MODBUS sensor connected to TBS09 must be virtually mapped to a SDI-12 sensor, which results in a specific SDI-12 address being assigned to the MODBUS sensor along with an identification string which can be retrieved with SDI-12 identification command *aI!*.

SDI 12 Master to MODBUS Slave Converter

Internally, TBS09 builds then a mapping table between MODBUS sensors and virtual SDI-12 sensors:

MAPPING TABLE FORMAT		
MODBUS sensor address	Virtual SDI-12 address	SDI-12 identification
EXAMPLE		
4	T	TekboxvnTBSWS1RVA2808231000103

Below extended SDI-12 commands are used to manage the SDI-12 virtual sensors list stored in TBS09 internal memory.

6.3.1 Assigning a virtual SDI-12 address and identification to a MODBUS sensor

A maximum of 4 MODBUS sensors can be configured and mapped as virtual SDI-12 sensors.

Extended command **XASS** is used to create a virtual SDI-12 sensor assigned to a given MODBUS sensor. At this step, it is also required to indicate how the MODBUS sensor data shall be read by TBS09.

MODBUS RTU standard uses big-endian data format to store 16-bits data, which is the default format used by TBS09. However, some MODBUS sensors manufacturers implementation differs from the standard by applying bytes swap and the same problem also occurs with 32-bits data where bytes swap, words swap or even both can be applied.

To cope with this situation, it is needed to indicate which swap type shall be applied when reading data from the MODBUS sensor using the field **Swp** from **XASS** extended command.

Below example shows how 8, 16 and 32-bits data types are read depending on the selected swap type:

Swap type	Data type (hexadecimal format)		
	8-bits data	16-bits data	32-bits data
No swap (default)	42	42 ED	42 ED 21 CB
Bytes swap		ED 42	ED 42 CB 21
Words swap			21 CB 42 ED
Bytes & Words swap			CB 21 ED 42

Command format	Description						
aXASS,ModA,SdiA,Swp, Vendor,Model, Ver, Opt!	Add MODBUS slave sensor to virtual SDI-12 sensors list						
Parameters description	<i>ModA</i> MODBUS address	<i>SdiA</i> SDI-12 address	<i>Swp</i> Data swap	<i>Vendor</i> Vendor ID	<i>Model</i> Sensor model	<i>Ver</i> Sensor version	<i>Opt</i> Optional field
Parameters values	1 to 247	0 – 9 A – Z a - z	0 (none) 1 (byte) 2 (word) 3 (both)	Custom string	Custom string	Custom string	Custom string
Response format	aOK<CR><LF>			aFAIL<CR><LF>			
Description	Success			Fail			
SDI-12 Command Example	aXASS,4,T,0,Tekbox,SMOD04,RVA,2PARAM! aOK<CR><LF>						

Table 12 Extended SDI-12 command aXASS

SDI 12 Master to MODBUS Slave Converter

Command format	Description			
aXGSS,ModA!	Retrieve virtual SDI-12 address and information for MODBUS sensor <i>ModA</i>			
Parameters description	ModA MODBUS address			
Parameters values	1 to 247			
Response format	aSdiA,Swp, Vendor,Model, Ver,Opt!<CR><LF>			aFAIL<CR><LF>
Description	Success			Fail
Parameters description	Vendor Vendor ID	Model Sensor model	Ver Sensor version	Opt Optional field
Parameters values	Custom string	Custom string	Custom string	Custom string
SDI-12 Command Example	aXGSS,4! aTekboxvn,TBSWS1,RVA,2401241000103<CR><LF>			

Table 13 Extended SDI-12 command aXGSS

6.3.2 Managing the MODBUS / virtual SDI-12 sensors configuration list

Mapped MODBUS sensors can be removed from the configuration list:

Command format	Description	
aXRSS,ModA!	Remove MODBUS sensor address <i>ModA</i> from the configuration list	
Parameters description	ModA MODBUS sensor address to be removed	
Parameters values	• 1 – 247	
Response format	aOK<CR><LF>	aFAIL<CR><LF>
Description	Success	Fail
SDI-12 Command Example	aXRSS,54! aOK<CR><LF>	

Table 14 Extended SDI-12 command aXRSS

Command format	Description	
aXCAS!	Clear the configuration list from all MODBUS sensors	
Parameters description	None	
Parameters values	None	
Response format	aOK<CR><LF>	aFAIL<CR><LF>
Description	Success	Fail
SDI-12 Command Example	aXCAS! aOK<CR><LF>	

Table 15 Extended SDI-12 command aXCAS

SDI 12 Master to MODBUS Slave Converter

Command format	Description	
aXLAS!	Retrieve MODBUS sensor and virtual SDI-12 addresses pairs	
Parameters description	None	
Parameters values	None	
Response format	aModA,SdiA,...<CR><LF>	aFAIL<CR><LF>
Description	Success – up to 4 pairs of MODBUS/SDI-12 addresses	Fail
Parameters description	ModA MODBUS address	SdiA SDI-12 address
Parameters values	<ul style="list-style-type: none"> • 1 to 247 	<ul style="list-style-type: none"> • 0 – 9 • A – Z • a -z
SDI-12 Command Example	aXLAS! a4,T<CR><LF>	

Table 16 Extended SDI-12 command aXLAS

6.4 Configuring SDI-12 measurement commands

Once each connected MODBUS sensor has been assigned to a virtual SDI-12 address, it is needed to configure the measurement commands for each sensor:

- Map MODBUS function code to a SDI-12 measurement command
- Define mapped SDI-12 measurement command time and number of returned parameters
- Define the data format of each returned measurement value

Note:

Each MODBUS sensor can be assigned with up to 8 different SDI-12 measurement commands and each measurement command can return up to 6 values.

6.4.1 Mapping MODBUS sensor function code to SDI-12 measurement command

For any MODBUS sensor connected to TBS09, it is required to build a table assigning a given MODBUS function code to a specific SDI-12 measurement command:

MAPPING TABLE FORMAT					
Index	MODBUS address	SDI-12 measurement command	SDI-12 measurement time	MODBUS function code	Returned values accuracy mode
EXAMPLE					
0	54	M2!	3s	0x04	Low

This mechanism gives full flexibility to the user to control MODBUS sensors measurements over SDI-12: for instance, some MODBUS sensors can have their measurements read right away by sending MODBUS function

SDI 12 Master to MODBUS Slave Converter

code (0x04) whereas some others would require to write to a specific coil or register to trigger the measurement before reading back the results.

Note:

Whereas MODBUS allows for storing values in various formats, SDI-12 standard limits the representation of measurement values to up to 7 digits (plus polarity sign and optional decimal point).

Therefore, TBS09 allows defining 2 different accuracy modes for the returned measurement values:

- Low accuracy mode: the measurement value is represented as a float coded over 7 digits with decimal point as per SDI-12 standard (so the measurement value will be rounded if exceeded 7 digits).
- High accuracy mode: the MODBUS sensor measurement value is converted to hexadecimal floating point (4 bytes – B3 B2 B1 B0) and each 2 bytes pair (B3B2 and B1B0) is reported as an SDI-12 measurement. In this case, after having been collected by the SDI-12 data logger the measurement value must be reconstructed by the backend application.

Below example shows how the MODBUS sensor measurement value +45236.36818 is reported over SDI-12 in both accuracy modes.

Low accuracy mode				
MODBUS sensor measurement (decimal float value)	+45236.36718			
aD0! Response	a+45236.37 Value rounded to be represented over 7 digits as per SDI-12 standard			
High accuracy mode				
MODBUS sensor measurement (decimal float format)	+45236.36718			
MODBUS sensor measurement (hexadecimal floating-point format)	0x4730B45E			
	Byte3	Byte2	Byte1	Byte0
	47	30	B4	5E
	Hexadecimal: 4730 Decimal: 18224		Hexadecimal: B45E Decimal: 46174	
aD0! Response	a+18224+46174			

The user shall therefore be cautious when selecting the accuracy mode, keeping in mind that the low accuracy mode is only suitable for values ranging from -9999999/+9999999 to comply with SDI-12 standard.

Would larger numbers be required or having decimal numbers whose total number of digits is more than 7 then the high accuracy mode shall be selected.

This choice has also to be made depending on the MODBUS measurements formats (cf extended command Table 22 Extended SDI-12 command aXSRD).

In case of wrong configuration, TBS09 will return +999.9999 if the measurements have more than 7 digits.

SDI 12 Master to MODBUS Slave Converter

Extended command **XAMC** is used to map SDI-12 commands to function codes for a specific MODBUS address device:

Command format	Description					
aXAMC,ModA,SdiCmd,ttt,FC,Acc,Tmo!	Assign SDI-12 command to MODBUS function code					
Parameters description	ModA MODBUS address	SdiCmd SDI-12 command	ttt SDI-12 response time	FC MODBUS Function Code	Acc Accuracy mode	Tmo Modbus response timeout
Parameters values	1 to 247	<ul style="list-style-type: none"> • M, M1 to M9 • MC, MC1 to MC9 • C, C1 to C9 • CC, CC1 to CC9 • R0 to R9 • RC0 to RC9 • V 	1 to 999 seconds	1, 2, 3, 4, 5, 6, 15, 16 and 23	<ul style="list-style-type: none"> • 0 (low) • 1 (high) 	1 to 65535 ms
Response format	aIndex <CR><LF>			aFAIL <CR><LF>		
Description	Success			Fail		
Parameters description	Index Index of the configured SDI-12 command for a specific MODBUS address					
Parameters values	0 to 7					
SDI-12 Command Example	aXAMC,4, M,2, 3,0,500! a0<CR><LF>					

Table 17 Extended SDI-12 command aXAMC

Command format	Description					
aXGMC,ModA,Index!	Retrieve SDI-12 command/MODBUS function code assignment					
Parameters description	ModA MODBUS address		Index Index of programmed SDI-12 command.			
Parameters values	1 to 247		0 to 7			
Response format	aSdiCmd,ttt,FC,Acc,Tmo <CR><LF>			aFAIL <CR><LF>		
Description	Success			Fail		
Parameters description	SdiCmd SDI-12 command	ttt SDI-12 response time	FC MODBUS Function Code	Acc Accuracy mode	Tmo Modbus response timeout	
Parameters values	<ul style="list-style-type: none"> • M, M1 to M9 • MC, MC1 to MC9 • C, C1 to C9 • CC, CC1 to CC9 • R0 to R9 • RC0 to RC9 • V 	1 to 999 seconds	1, 2, 3, 4, 5, 6, 15, 16 and 23	<ul style="list-style-type: none"> • 0 (low) • 1 (high) 	1 to 65535 ms	

SDI 12 Master to MODBUS Slave Converter

SDI-12 Command Example	aXGMC,4,0! aM,2,3,0,500<CR><LF>
-------------------------------	------------------------------------

Table 18 Extended SDI-12 command aXGMC

This results in TBS09 building a mapping table between each MODBUS sensors address (maximum 4 MODBUS sensors) and SDI-12 commands to be executed, each configuration being identified by the index returned by **XAMC** command.

Index	MODBUS address	SDI-12 command	MODBUS function code	SDI-12 measurement time	Accuracy mode	Tmo
0	MODBUS Address N	SDI_Cmd_0	MOD_FC_0	ttt_0	Acc_0	Tmo_0
1		SDI_Cmd_1	MOD_FC_1	ttt_1	Acc_1	Tmo_1
...	
7		SDI_Cmd_7	MOD_FC_7	ttt_7	Acc_7	Tmo_7

A total of up to 8 pairs can be configured for each MODBUS sensor and any combination between MODBUS addresses and SDI-12 commands are allowed, for instance with 2 different MODBUS sensors:

Index	MODBUS Address	SDI-12 Command
0	32	M1
1	32	MC3
2	32	C
3	32	C4
0	51	M1
1	51	M2

This index is then used by extended commands **XSRD** and **XSWD** to format the data to be read/written for a specific MODBUS function code / SDI-12 command pair to be executed.

A set of extended commands allows to handle the list of configured SDI-12 measurement commands:

Command format	Description	
aXRMC,ModA,Index!	Remove SDI-12 command from the list for a specific MODBUS address	
Parameters description	ModA MODBUS slave address	Index Index of programmed SDI-12 command to be removed
Parameters values	1 to 247	0 to 7
Response format	aOK<CR><LF>	aFAIL<CR><LF>
Description	Success	Fail
SDI-12 Command Example	aXRMC,4,0! aOK<CR><LF>	

Table 19 Extended SDI-12 command aXRMC

SDI 12 Master to MODBUS Slave Converter

Command format	Description	
aXCAC,ModA!	Clear all SDI-12 commands from the list for a specific MODBUS address	
Parameters description	ModA MODBUS slave address	
Parameters values	1 to 247	
Response format	aOK<CR><LF>	aFAIL<CR><LF>
Description	Success	Fail
SDI-12 Command Example		
	aXCAC,4! aOK<CR><LF>	

Table 20 Extended SDI-12 command aXCAC

Note:

A command configuration can't be directly updated. It must first be removed using *XRMC* and then created again using *XAMC*.

Doing so the commands indexes are automatically reordered without any other configuration required for the commands that have not been changed.

Command format	Description	
aXGCN,ModA!	Get number of configured SDI-12 commands for a specific MODBUS address	
Parameters description	ModA MODBUS slave address	
Parameters values	1 to 247	
Response format	aNumCmd<CR><LF>	aFAIL<CR><LF>
Description	Success	Fail
Parameters description	NumCmd Number of configured SDI-12 commands.	
Parameters values	0 to 8	
SDI-12 Command Example		
	aXGCN,4! a2<CR><LF>	

Table 21 Extended SDI-12 command aXGCN

6.4.2 Read and write operations

It is crucial to ensure the MODBUS register addresses used for read and write operations are matching with the defined function code, like accessing holding registers and not input registers while using MODBUS function code 0x03.

Furthermore, TBS09 allows MODBUS sensors to respond to MODBUS commands within 0-65535ms time frame as defined by *XAMXC* command. Outside this window, a MODBUS communication error will be returned.

SDI 12 Master to MODBUS Slave Converter

6.4.2.1 Read command configuration

For the function codes performing a read operation, it is needed to define the expected data format so the read value can be correctly reported over SDI-12. Considering SDI-12 standard allows only for up to 7 digits along with a sign and decimal point, it is important to carefully choose which [accuracy mode](#) shall be used to report the measurements.

The command allows to retrieve multiple parameters as long as they're stored in continuous registers, otherwise subsequent commands will have to be used.

Command format	Description				
aXSRD,ModA,Index,RdA,NbR,Format!	Define MODBUS parameters address and format for MODBUS/SDI-12 configuration index				
Parameters description	ModA MODBUS slave address	Index Index returned by XAMC	RdA MODBUS register address	NbR Number of MODBUS registers to read	Format List of parameters formats to be read
Parameters values	1 to 247	0 - 7	0 - 65535	(*)	(*)
Response format	aValNb<CR><LF>			aFAIL<CR><LF>	
Description	Success			Fail	
Parameters description	ValNb Number of values that will be returned by SDI-12 data command aDx! after executing the configured SDI-12 measurement command.				
Parameters values	1 - 6				
SDI-12 Command Example	aXSRD,4,0,0,4,6,6! a2<CR><LF>				

Table 22 Extended SDI-12 command aXSRD

Format (*): while reading MODBUS registers content, it is required to define the desired data format so it can be correctly reported over SDI-12 when SDI-12 data command aDx! is executed. The format type is encoded as a number as follows:

- 0 char (1 byte)
- 1 unsigned char (1 byte)
- 2 short (2 bytes)
- 3 unsigned short (2 bytes)
- 4 integer (4 bytes)
- 5 unsigned integer (4 bytes)
- 6 float (4 bytes)
- 7 dummy parameter (1 byte)
- 8 dummy parameter (2 bytes)
- 9 dummy parameter (4 bytes)

SDI 12 Master to MODBUS Slave Converter

XSRD response includes the number of measurement values that will be returned by the executed SDI-12 command as indicated in *attn* for Measurement commands aMx!/aMCx! and *attnn* for Concurrent measurement commands aCx!/aC(C)x!.

The total size that can be read is 40 bytes (so up to 20 2-bytes registers) and the maximum number of returned parameters is limited to 9.

In some cases, especially when reading multiple registers, not all bytes are relevant and those bytes must be ignored by the measurement command. In that case it is required to use the dummy parameter format codes *a*, *b*, *c*, *d* so TBS09 ignores them. This case is detailed further in this document by [a comprehensive example](#).

Command format	Description	
aXGRD,ModA,Index!	Define MODBUS parameters address and format for MODBUS/SDI-12 configuration <i>index</i>	
Parameters description	ModA MODBUS slave address	Index Index returned by XAMC
Parameters values	1 to 247	0 - 7
Response format	aValNb,<ParListFmt><CR><LF>	
Description	Success	Fail
Parameters description	ValNb Number of values that will be returned by SDI-12 data command aDx! after executing the configured SDI-12 measurement command.	ParListFmt List of each parameter data format (refer to Table 22 for details)
Parameters values	1 - 6	0 - 9
SDI-12 Command Example	aXGRD,4,0! a2,4,6,6<CR><LF>	

Table 23 Extended SDI-12 command aXGRD

6.4.2.2 Write command configuration

For function codes performing a write operation, it is required to define the MODBUS register to be written and its content. Multiple continuous registers can be written using one single SDI-12 command **XSWD**.

Command format	Description				
aXSWD,ModA,Index,WrA,NbR,Content!	Define MODBUS register address and content to be written for MODBUS/SDI-12 configuration <i>index</i>				
Parameters description	ModA MODBUS slave address	Index Index returned by XAMC	WrA MODBUS register address	NbR Number of MODBUS registers to be written	Content Hexadecimal data to be written to the registers

SDI 12 Master to MODBUS Slave Converter

Parameters values	1 to 247	0 - 7	0 - 65535	0 - 65535	Hexadecimal string (32 bytes max.)
Response format	aOK<CR><LF>		aFAIL<CR><LF>		
Description	Success		Fail		
SDI-12 Command Example					
aXSWD,15,5,29,1,1a1a! aOK<CR><LF>					

Table 24 Extended SDI-12 command aXSWD

Command format	Description	
aXGWD,ModA,Index!	Get the hexadecimal string to be written to MODBUS sensor ModA by command XSWD with index Index	
Parameters description	ModA MODBUS slave address	Index Index returned by XAMC
Parameters values	1 to 247	0 - 7
Response format	a NbR,Content <CR><LF>	aFAIL<CR><LF>
Description	Success	Fail
Parameters description	NbR Number of bytes to be written	Content Hexadecimal data to be written to the MODBUS register displayed as a list of bytes.
Parameters values	1 - 32	Hexadecimal bytes list
SDI-12 Command Example		
aXGWD,15,5! a2,1A,1A<CR><LF>		

Table 25 Extended SDI-12 command aXGWD

In case of write operation, TBS09 automatically sets the number of returned parameters to 1.

So, when executing the data command aDx! after executing an SDI-12 measurement command assigned to a MODBUS write function code (ex 0x06), the returned value is either 0 (fail) or 1 (success).

6.4.2.3 Direct access to MODBUS registers

It is also possible to directly access connected MODBUS sensors registers without using SDI-12 commands virtual mapping.

Extended SDI-12 command **XSIC** is used to send the desired MODBUS request while **XGIR** retrieves the response.

SDI 12 Master to MODBUS Slave Converter

Command format	Description						
aXSIC,ModA,Func,RdA,NbR,WrA,NbW,W_str!	Directly sends a MODBUS function code Func request to address ModA to read or write a defined number of registers.						
Parameters description	ModA MODBUS slave address	Func MODBUS function code	RdA Register read address	NbR Number of registers to read	WrA Register write address	NbW Number of registers to write	W_str Hexadecimal string to write
Parameters values	1 - 247	1-6, 15, 16, 23	(*)	(*)	(*)	(*)	(*)
Response format	aOK<CR><LF>				aFAIL<CR><LF>		
Description	Success				Fail		
SDI-12 Command Example	aXSIC,4,3,11,1,0,0,00! aOK<CR><LF>						

Table 26 Extended SDI-12 command aXSIC

Register addresses and number of bytes to read/write (*): registers address range depends on the selected MODBUS function code and **RdA/WrA** shall be set accordingly.

The extended command XSIC can buffer up to 20 bytes to be read or written depending on the operation. Therefore, the maximum number of registers to be read (**NbR**) or written (**NbW**) depends on the selected function code: for instance, when using function code 3, 2 bytes are read for each register so the maximum total number of registers that can be read at once is 10.

Command format	Description	
aXGIR!	Get MODBUS response following the MODBUS request executed by XSIC	
Parameters description	None	
Parameters values	None	
Response format	aHex_string<CR><LF>	aFAIL<CR><LF>
Description	Success	Fail
Parameters description	Hex_string MODBUS response returned as a hexadecimal string	
Parameters values	Dependent on MODBUS sensor and executed function code, hexadecimal string.	
SDI-12 Command Example	aXGIR! a002C<CR><LF>	

Table 27 Extended SDI-12 command aXGIR

SDI 12 Master to MODBUS Slave Converter

6.4.3 Measurements conversion

TBS09 allows applying a custom conversion on one measurement value for each MODBUS sensor.

The selected SDI-12 measurement is identified through 3 parameters:

- The physical MODBUS address corresponding to the virtual SDI-12 address
- The SDI-12 measurement command index returned by **XAMC** extended command
- The index of the measurement value returned by the SDI-12 measurement command

4 different conversion methods are available for the measurement value V :

- Linear conversion: scaling factor and offset are applied to the measurement value.
- Conversion as a percentage: $V = \frac{V-Min}{Max-Min} * 100$
- Polynomial conversion: $V = a * V^4 + b * V^3 + c * V^2 + d * V + e$
- Multi-polynomial conversion: up to 3 different 4th order polynomials can be used on up to 3 continuous ranges - $Poly_0[V_{Min0} ; V_{Max0}]$, $Poly_1[V_{Min1} ; V_{Max1}]$ and $Poly_2[V_{Min2} ; V_{Max2}]$

The configuration of the conversion for a specific measurement value is made in two steps:

- Assign the conversion method to the measurement value to be modified (**XAPC** command). TBS09 builds then a list of conversion profiles that are identified by an index.
- Configure the parameters of the selected conversion method (**XSMT** command)

Command format	Description			
aXAPC,ModA,CmdIdx,MeasIdx,MID!	Assign conversion method MID to the measurement value MeasIdx returned by SDI-12 command CmdIdx for MODBUS sensor ModA .			
Parameters description	ModA MODBUS slave address	CmdIdx Index returned by XAMC	MeasIdx Measurement value index	MID Conversion method
Parameters values	1 to 247	0 - 7	0 - 5	<ul style="list-style-type: none"> • 0 (linear) • 1 (percentage) • 2 (polynomial) • 3 (multi-polynomials)
Response format	aCidx <CR><LF>		aFAIL <CR><LF>	
Description	Success		Fail	
Parameters description	Cidx Conversion profile index			
Parameters values	0 - 3			
SDI-12 Command Example	aXAPC,4,2,0,2! a0<CR><LF>			

Table 28 Extended SDI-12 command aXAPC

SDI 12 Master to MODBUS Slave Converter

Command format	Description	
aXSMT,Cldx,Coeff_list!	Set the coefficients of the conversion method Cldx defined by XAPC .	
Parameters description	Cldx Conversion method index returned by XAPC .	Coeff_list List of coefficients to be applied depending on the conversion method that is selected.
Parameters values	0 - 3	8 characters for each coefficient maximum including sign and decimal point.
Response format	aOK<CR><LF>	aFAIL<CR><LF>
Description	Success	Fail
SDI-12 Command Example	aXSMT,0,0,0.00100,3.2,-0.2,-5! (polynomial method) aOK<CR><LF>	

Table 29 Extended SDI-12 command aXSMT

Command format	Description				
aXGPC,Cldx!	Retrieve configuration for conversion profile index Cldx .				
Parameters description	Cldx Conversion profile index				
Parameters values	0 – 3				
Response format	aModA,Cmdldx,Measldx,MID<CR><LF>			aFAIL<CR><LF>	
Description	Success			Fail	
Parameters description	ModA MODBUS slave address	Cmdldx Index returned by XAMC	Measldx Measurement value index	MID Conversion method	
Parameters values	1 to 247	0 - 7	0 - 5	<ul style="list-style-type: none"> • 0 (linear) • 1 (percentage) • 2 (polynomial) • 3 (multi-polynomials) 	
SDI-12 Command Example	aXGPC,0! a4,2,0,2<CR><LF>				

Table 30 Extended SDI-12 command aXGPC

SDI 12 Master to MODBUS Slave Converter

Command format	Description		
aXRPC,ModA,CmdIdx,MeasIdx!	Remove conversion profile of measurement value MeasIdx returned by SDI-12 command CmdIdx for MODBUS sensor ModA .		
Parameters description	ModA MODBUS slave address	CmdIdx Index returned by XAMC	MeasIdx Measurement value index
Parameters values	1 to 247	0 - 7	0 - 5
Response format	aOK<CR><LF>		aFAIL<CR><LF>
Description	Success		Fail
SDI-12 Command Example	aXRPC,4,2,0,2! aOK<CR><LF>		

Table 31 Extended SDI-12 command aXRPC

Command format	Description			
aXGMT,CIdx,<Idx>!	Retrieve configuration for conversion profile index CIdx .			
Parameters description	CIdx Conversion profile index			
Parameters values	0 – 3			
Response format	a ModA,CmdIdx,MeasIdx,MID <CR><LF>			aFAIL<CR><LF>
Description	Success			Fail
Parameters description	ModA MODBUS slave address	CmdIdx Index returned by XAMC	MeasIdx Measurement value index	MID Conversion method
Parameters values	1 to 247	0 - 7	0 - 5	<ul style="list-style-type: none"> • 0 (linear) • 1 (percentage) • 2 (polynomial) • 3 (multi-polynomials)
SDI-12 Command Example	aXGMT,2,00! a+0.00000,+0.00100,+3.20000,-0.20000,-5.00000<CR><LF>			

Table 32 Extended SDI-12 command aXGMT

6.4.4 MODBUS sensors wake-up

Some MODBUS sensors require a wake-up signal to start operating. This can be done by sending a specific character or hexadecimal string.

Therefore, TBS09 supports 3 wake-up modes:

- No wake-up: this is the default configuration.
 - This mode has to be used with any sensor that does not require any wake-up signal.

SDI 12 Master to MODBUS Slave Converter

- Custom: in this mode, TBS09 sends the wake-up command only to a specific MODBUS sensor.
 - Only one connected MODBUS sensor can be configured to receive a wake-up command (so it's not possible to have 2 different sensors having 2 different wake-up commands).
- Broadcast: in this mode, TBS09 sends the wake-up command to all configured MODBUS sensors.
 - To be used only if all connected MODBUS sensors are the same.

Command format	Description	
aXSWM, WakeupMode!	Configure MODBUS sensor wake-up mode.	
Parameters description	WakeupMode Wake-up mode (none, custom, broadcast)	
Parameters values	<ul style="list-style-type: none"> • 0: none • 1: custom • 2: broadcast 	
Response format	aOK<CR><LF>	aFAIL<CR><LF>
Description	Success	Fail
SDI-12 Command Example	aXSWM,1! aOK<CR><LF>	

Table 33 Extended SDI-12 command aXSWM

Command format	Description	
aXGWM!	Get MODBUS sensors wake-up mode	
Parameters description	None	
Parameters values	None	
Response format	a WakeupMode <CR><LF>	aFAIL<CR><LF>
Description	Success	Fail
Parameters description	WakeupMode Wake-up mode (none, custom, broadcast)	
Parameters values	<ul style="list-style-type: none"> • 0: none • 1: custom • 2: broadcast 	
SDI-12 Command Example	aXGWM! a1<CR><LF>	

Table 34 Extended SDI-12 command aXGWM

SDI 12 Master to MODBUS Slave Converter

Command format	Description			
aXAWC, Cmd, Tdur, Tact, ModA!	Configure custom wake-up command for MODBUS sensor address ModA			
Parameters description	Cmd Wake-up string	Tdur Wake-up period before any measurement can be taken	Tact Period during which the sensor remains active after wake-up	ModA MODBUS address
Parameters values	Hexadecimal string	0-65535 ms	0-255 s	1 to 247 (0 if broadcast mode is used)
Response format	aOK<CR><LF>		aFAIL<CR><LF>	
Description	Success		Fail	
SDI-12 Command Example	aXAWC,0D,1200,5,1! aOK<CR><LF>			

Table 35 Extended SDI-12 command aXAWC

Command format	Description	
aXRWC, ModA!	Remove wake-up configuration for MODBUS sensor address ModA .	
Parameters description	ModA Sensor MODBUS address	
Parameters values	1 - 247	
Response format	aOK<CR><LF>	aFAIL<CR><LF>
Description	Success	Fail
SDI-12 Command Example	aXRWC,1! aOK<CR><LF>	

Table 36 Extended SDI-12 command aXRWC

Command format	Description			
aXGWC, ModA!	Retrieve wake-up configuration for MODBUS sensor address ModA .			
Parameters description	ModA Sensor MODBUS address			
Parameters values	1 – 247 (Note: in case of broadcast wake-up mode, the command is XGWC!)			
Response format	a Cmd, Tdur, Tact, ModA <CR><LF>			aFAIL<CR><LF>
Description	Success			Fail
Parameters description	Cmd Wake-up string	Tdur Wake-up period before any	Tact Period during which the sensor remains active after wake-up	ModA MODBUS address

SDI 12 Master to MODBUS Slave Converter

		measurement can be taken		
Parameters values	Hexadecimal string	0-65535 ms	0-255 s	1 to 247 (not reported if broadcast mode is used)
SDI-12 Command Example	aXGWC,1! a0D,1200,5,1<CR><LF>			

Table 37 Extended SDI-12 command aXGWC

7 TBS09 configuration and communication examples

In the following example, TBS09 communicates with 2 MODBUS RTU sensors

The initial setup is as follows:

- TBS09 SDI-12 address: 0
- MODBUS sensors:
 - SMOD04, MODBUS address: 4, measures 2 parameters.
 - SMOD15, MODBUS address: 15, measures 4 parameters.
 - Data format: big endian
 - RS485 configuration: baud rate 9600, 8 data bits, 1 stop bit, no parity, half duplex

In this example, it is considered the following parameters will be retrieved based on below mapping and conditions:

SMOD04 sensor (MODBUS address: 4)										
Data type	Memory mapping	MODBUS register address	MODBUS function code	Comments						
Float	Param_1	0	(0x03) Read Holding Registers	-						
		1								
Float	Dummy_1	2		(0x03) Read Holding Registers	Dummy data is ignored.					
		3								
Float	Dummy_2	4				(0x03) Read Holding Registers	Dummy data is ignored.			
		5								
Float	Dummy_3	6						(0x03) Read Holding Registers	Dummy data is ignored.	
		7								
Float	Param_2	8			(0x03) Read Holding Registers					-
		9								

SDI 12 Master to MODBUS Slave Converter

SMOD15 sensor (MODBUS address: 15)				
Data type	Memory mapping	MODBUS register address	MODBUS function code	Comments
Float	Param_1	14	(0x03) Read Holding Registers	-
		15		
Unsigned Char	Dummy_1 Param_2	16		Other half of the register is dummy data
Short	Param_3	17		-
Unsigned Int	Param_4	18		-
		19		
-	Trigger	29	(0x06) Write Single Register	Need to write 0x1A1A (6682) to this register to trigger the measurement before reading above registers.

Switch TBS09 to configuration mode

```
0XSPM!  
0OK
```

Configure RS485

9600, 8 bits, 1 stop bit, no parity

```
0XSMF,9600,8,2,1!  
0OK  
0XGMF!  
09600,8,2,1
```

Half duplex

```
0XSMI,0!  
0OK
```

Power configuration

Power mode: TBS09 always active

```
0XSMP,0!  
0OK
```

No warm-up delay

```
0XSPW,0!  
0OK
```

SDI 12 Master to MODBUS Slave Converter

Define virtual SDI-12 sensors address

SMOD04 (MODBUS address 4) is assigned virtual SDI-12 address T.
SMOD15 (MODBUS address 15) is assigned virtual SDI-12 address U.

Clear sensors list

0XCAS!

00K

Map virtual SDI-12 address and information

0XASS,4,T,0,Tekbox,SMOD04,RVA,2PARAM!

00K

0XASS,15,U,0,Tekbox,SMOD04,RVA,4PARAM!

00K

Build SDI-12 measurement commands and configure returned values format

The following commands are implemented:

SMOD04							
SDI-12 command	Parameters					ttt	n
M!	Param_1					2	1
M1!	Param_2					2	1
M2!	Param_1	Dummy_1	Dummy_2	Dummy_3	Param_2	3	2

Note:

M2! shows how to retrieve both *Param_1* and *Param_2* with one single command. However due to the fact these parameters are not contiguous, it is needed to include the 3 dummy float data as well so TBS09 can correctly return both parameters' values *Param_1* and *Param_2*.

SMOD15							
SDI-12 command	Returned parameters					ttt	nn
C!	Param_1					2	01
C1!		Dummy_1	Param_2			1	01
C2!				Param_3		4	01
C3!					Param_4	2	01
C4!	Param_1	Dummy_1	Param_2	Param_3	Param_4	7	04
C5!	<i>This is MODBUS write operation (0x06) so the number of returned parameters is automatically assigned to 1 (possible values: 0- fail, 1- success)</i>					1	01

SDI 12 Master to MODBUS Slave Converter

Note:

Param_2 is defined as a char and since MODBUS function code (0x03) is used, it can only be retrieved by reading the whole register content (16 bits).

Therefore C1! will be configured to return 2 char parameters, one dummy value corresponding the 8 most significant bits and the second parameter being *Param_2*. It will be implemented likewise for C4!.

TM!:

0XAMC, 4, M,2, 3,0,0!

00

0XSRD, 4, 0, 0,2, 6!

01

TM1!:

0XAMC, 4, M1,2, 3,0,0!

01

0XSRD, 4, 1, 8,2, 6!

01

TM2!:

0XAMC, 4, M2,3, 3,0,0!

02

0XSRD, 4, 2, 0,10, 6, 9, 8, 8, 7, 7, 8, 6!

02

UC!:

. 0XAMC, 15, C,2, 3,0,0!

. 00

! 0XSRD, 15, 0, 14,2, 6!

. 01

UC1!:

0XAMC, 15, C1,1, 3,0,0!

01

0XSRD, 15, 1, 16,1, 7,1!

01

UC2!:

0XAMC, 15, C2,4, 3,0,0!

02

0XSRD, 15, 2, 17,1, 2!

01

SDI 12 Master to MODBUS Slave Converter

UC3!:

0XAMC, 15, C3,2, 3,0,0!

03

0XSRD, 15, 3, 18,2, 5!

01

UC4!:

0XAMC, 15, C4,7, 3,0,0!

04

0XSRD, 15, 4, 14,6, 6, 7, 1, 2, 5!

04

UC5!:

0XAMC, 15, C5,1, 6,0,0!

05

0XSWD, 15, 5, 29,1,1a1a!

00K

Switch TBS09 to measurement mode

0XSTM!

00K

8 Communication protocols

8.1 SDI-12

SDI-12 is a standard for interfacing data recorders with microprocessor-based sensors. SDI-12 stands for serial/digital interface at 1200 baud. It can connect multiple sensors with a single data recorder on one cable. It supports up to 60-meter cable between a sensor and a data logger.

The SDI-12 standard is prepared by

**SDI-12 Support Group
(Technical Committee)**
165 East 500 South
River Heights, Utah
435-752-4200
435-752-1691 (FAX)
<http://www.sdi-12.org>

The standard is available on the website of the SDI-12 Support Group.

SDI 12 Master to MODBUS Slave Converter

8.2 MODBUS

The MODBUS standard is managed by MODBUS Organization and the reference MODBUS Application Protocol is available on their website: <https://www.modbus.org/>

9 Mechanical information

Housing	Length (mm)	Width (mm)	Height (mm)
DIN rail	90.2	36.3	41.9
FIBOX	120	80	55

10 Environmental specification

Symbol	Parameter	Conditions	Min	Max	Unit
T _A	Operating Ambient Temperature Range	DIN rail: standard room humidity levels.	+20	+30	°C
T _{STG}	Storage Temperature Range	DIN rail housing	-20	+80	°C
		FIBOX housing	-40	+140	
-	Waterproofness	DIN rail: no, indoor use only	-		
		FIBOX: yes, IP67			

11 Ordering information

Part Number	Description
TBS09	MODBUS master to SDI-12 slave converter – FIBOX housing
TBS09DR	MODBUS master to SDI-12 slave converter – DIN rail housing
TBS09J	7 ports MODBUS junction box (half/full duplex) – FIBOX housing

12 History

Version	Date	Author	Changes
V1.0	4.11.2014	Mayerhofer	Creation of the document
V1.1	4.12.2015	Mayerhofer	formatting
V1.2	2.11.2019	PhuThinh	Updated baud rate
V1.3	24.06.2020	Hoa Hoang	Updated document name from SDI 12 to MODBUS converter to SDI 12 Master to Modbus Slave Converter Update ordering information
V1.4	2.04.2021	Mayerhofer	Updated figure 1
V1.5	08.03.2022	Philippe Hervieu	Updates related to default values and XSMVS parameters.

SDI 12 Master to MODBUS Slave Converter

V2.0	24.01.2024	Philippe Hervieu	User manual reworked following FW upgrade.
V2.1	11.03.2024	Philippe Hervieu	Update chapter 4 – power supply for MODBUS sensors
V2.2	29.03.2024	Philippe Hervieu	Add sensor wake-up feature and update commands limitations.
V2.3	13.12.2024	Philippe Hervieu	Update 5.5 Limitations