MICRONAS

Application Note

Magnetic Sensor Programmer V1.x

Edition June 4, 2020
APN000136_003EN

Copyright, Warranty, and Limitation of Liability

Due to the open construction of the evaluation board/ kit, it is the user's responsibility to take any and all appropriate precautions with regard to safe and proper handling and use. The user assumes all responsibly and liability for proper and safe handling of the evaluation boards/ kits. Further, the user indemnifies TDK-Micronas GmbH from third party claims arising from the handling or use of the goods to the extent the user would also be directly liable.

EXCEPT FOR TDK-MICRONAS GMBH INTENTIONAL MISBEHAVIOR OR GROSS NEGLIGENCE TDK-MICRONAS GMBH REJECTS ANY LIABILITY FOR IMPROPER USE. CLAIMS FROM PRODUCT LIABILITY REMAIN UNAFFECTED.

By providing the evaluation board/ kit, no license is granted under any patent right or other intellectual property right whatsoever for any use other than the limited use described above.

TDK-Micronas Trademarks

HAL, HAC, HAR, 3D HAL, masterHAL, curSENS

Third-Party Trademarks

All other brand and product names or company names may be trademarks of their respective companies.

License Note

HAL 36xy, HAL 37xy, HAL 38xy, and HAL 39xy use licenses of Fraunhofer Institute for Integrated Circuits IIS.

Contents

Page	Section	Title
7	**1.**	**General Information**
7	1.1.	Certification
8	1.2.	Support
8	1.3.	Introduction
8	1.3.1.	Supported Sensors
9	1.3.2.	Sensor-Specific PC Software
10	1.4.	Block Diagram of the MSP
11	**2.**	**Getting Started**
11	2.1.	USB Driver Installation
12	2.2.	First Steps
12	2.2.1.	Connect MSP V1.x
12	2.2.2.	Check Communication with PC and Sensor
13	**3.**	**Magnetic Sensor Programmer V1.x Configuration**
15	3.1.	Firmware Update
17	**4.**	**Specification**
17	4.1.	Absolute Maximum Ratings
17	4.2.	Recommended Wiring
18	4.3.	Maintenance and Calibration
19	4.4.	Recommended Operating Conditions
20	**5.**	**Functions**
20	5.1.	Serial Command Interpreter
20	5.1.1.	Serial Interface Configuration
20	5.1.2.	Definition of the COMMAND Frame
21	5.1.3.	Definition of the RESPONSE Frame
21	5.1.4.	Analog Measurements
22	5.1.5.	Error Codes
22	5.2.	Biphase Interface
24	**6.**	**Configuration Commands**
29	**7.**	**Operation Mode 8**
29	7.1.	Operation Mode 8 – Configuration Commands
30	7.2.	HAL/ HAR 3900
30	7.2.1.	Available Command Frames
31	7.2.2.	Available Sensor Commands
31	7.2.3.	Read
31	7.2.4.	Write
32	7.2.5.	CRC
34	7.3.	CUR 42xy
34	7.3.1.	General SPI Frame
34	7.3.2.	Available Commands
34	7.3.3.	Read
35	7.3.4.	Write
35	7.3.5.	CRC
36	7.4.	Protocol Error Handling
36	7.5.	Mode 8 – Communication commands

Contents, continued

Page	Section	Title
38	**8.**	**Operation Mode 9**
38	8.1.	Operation Mode 9 – Configuration Commands
38	8.2.	Available Command Frames
39	8.3.	Available Sensor Commands
39	8.3.1.	Set Base Address
39	8.3.2.	Read with Absolute Address
39	8.3.3.	Read with Base Address
40	8.3.4.	Write Byte with Base Address
40	8.3.5.	Write Word with Base Address
40	8.3.6.	Special Cases
40	8.3.7.	CRC
41	8.3.8.	Protocol Error Handling
42	8.4.	Mode 9 – Commands
44	**9.**	**Operation Mode A**
44	9.1.	Operation Mode A – Configuration Commands
45	9.2.	Available Command Frames
46	9.3.	Available Sensor Commands
46	9.3.1.	Set Base Address
46	9.3.2.	Read
46	9.3.3.	Write
46	9.3.4.	Protocol Error Handling
46	9.3.5.	Data Check
47	9.3.6.	CRC Check
47	9.3.7.	Parity Check
48	9.4.	Mode A – Commands
49	**10.**	**Operation Mode B**
49	10.1.	Operation Mode B – Configuration Commands
49	10.2.	Available Command Frames
50	10.3.	Available Sensor Commands
50	10.3.1.	Read
51	10.3.2.	Write
51	10.3.3.	Protocol Error Handling
52	10.4.	Mode B – Commands
53	**11.**	**Operation Mode C**
53	11.1.	Operation Mode C – Configuration Commands
53	11.2.	Available Command Frames
54	11.3.	Available Sensor Commands
54	11.3.1.	Set Base Address
54	11.3.2.	Read
54	11.3.3.	Write
55	11.3.4.	CRC
55	11.3.5.	Parity Check
55	11.3.6.	Protocol Error Handling
56	11.4.	Mode C – Commands

Contents, continued

Page	Section	Title
57	**12.**	**Operation Mode D**
57	12.1.	Operation Mode D – Configuration Commands
58	12.2.	Available Command Frames
58	12.3.	Available Sensor Commands
59	12.3.1.	Read
59	12.3.2.	Write
60	12.3.3.	CRC
61	12.3.4.	Protocol Error Handling
62	12.4.	Mode D – Commands
63	**13.**	**Document History**

Release Note:	Revision bars indicate significant changes to the previous document

1. General Information

The hardware and software description in this document is valid for the **Magnetic Sensor Programmer V1.x** (**MSP** for short).

The MSP can be ordered via the TDK-Micronas' Customer Service or via distributors. Ordering code: 999-000-50.

Fig. 1–1: Magnetic Sensor Programmer V1.x

	MSP Interface	Function
1	DB-25 Interface	Communication with sensors over SPI or Biphase via VSUP/ OUT
2	HAL 1/ 2 Interface	Communication with sensors over Biphase via VSUP/ OUT
3	Status LEDs	Status of the MSP
4	24V DC Jack	24V power supply
5	Reset	Reset the MSP
6	USB Interface	Communication with host computer
7	RS-232 Interface	Communication with host computer

1.1. Certification

TDK-Micronas GmbH fulfills the requirements of the international automotive standard IATF 16949 and is certified according to ISO 9001:2015. This ISO standard is a world-wide accepted quality standard.

1.2. Support

We kindly ask you to register on https://service.micronas.com in order to obtain access to the workgroups for our various product families. Here you can request for support by opening a support ticket in the customer support system.

TDK-Micronas GmbH - Application Engineering

Hans-Bunte-Strasse 19

D-79108 Freiburg im Breisgau

1.3. Introduction

The Magnetic Sensor Programmer V1.x is the programmer board for TDK-Micronas' magnetic field sensors with analog and digital output formats. It provides application software supporting a command interface for the communication with a PC. This allows the implementation of specific PC software for engineering purposes.

The MSP can be used in laboratories for engineering purposes.

Note: The MSP is not recommended for production purpose (or environment).

1.3.1. Supported Sensors

The MSP supports the sensors listed in Table 1–1.

Table 1–1: Supported sensors

Sensor	MSP mode	Description
HAL 18xy		
HAL 1820	A	Linear sensor with analog output
HAL 1880	A, C	Linear sensor with analog output
HAL 1890	A, C	Linear sensor with SENT output
HAL/ HAR 24xy[1)]		
HAL 2420, HAL 2421	A, C	Linear sensor with analog output
HAL/ HAR 2425	A, C	Linear sensor with analog output, 16 setpoints
HAL/ HAR 2455	A, C	Linear sensor with PWM output, 16 setpoints
HAL 28xy		
HAL 283x	9	Linear sensor with SENT output
HAL 2850	9	Linear sensor with fast PWM output

Table 1–1: Supported sensors, continued

Sensor	MSP mode	Description
HAL 36xy		
HAL 3625	A, C	Direct angle sensor with analog output (B_X/ B_Y)
HAL 3675	A, C	Direct angle sensor with PWM output (B_X/ B_Y)
HAL/ HAR/ HAC 37xy[1)]		
HAL/ HAR/ HAC 3715	A, C	Direct angle sensor with analog/ modulo output (B_X/ B_Y)
HAL/ HAR/ HAC 3725	A, C	Direct angle sensor with analog output (B_X/ B_Y)
HAL/ HAR/ HAC 3726	A, C	2D position sensor with analog output (B_Y/ B_Z)
HAL/ HAR/ HAC 3727	A, C	2D position sensor with analog output (B_X/ B_Z)
HAL/ HAR/ HAC 3735	A, C	Direct angle sensor with PWM and SENT output (B_X/ B_Y)
HAL/ HAR/ HAC 3736	A, C	2D position sensor with PWM and SENT output (B_Y/ B_Z)
HAL/ HAR/ HAC 3737	A, C	2D position sensor with PWM and SENT output (B_X/ B_Z)
HAL 38xy		
HAL 3855	A, C	2D position sensor with analog output (B_Y/ B_Z)
HAL 3856	A, C	2D position sensor with analog output (B_X/ B_Z)
HAL 3875	A, C	2D position sensor with PWM output (B_Y/ B_Z)
HAL 3876	A, C	2D position sensor with PWM output (B_X/ B_Z)
HAL 39xy[1)]		
HAL 393x	D	3D position sensor with PWM and SENT output
HAL 3900	8	3D position sensor with SPI interface
CUR 42xy		
CUR 42xy	8, D	Tunnel Magneto Resistance (TMR) based current sensor with SENT/ SPI output
1) HAL: single-die, HAR: dual-die, HAC: single-die with integrated caps		

Please visit https://service.micronas.com to access the corresponding Data Sheets, User Manuals and Programming Guides for detailed information on the listed sensors.

1.3.2. Sensor-Specific PC Software

TDK-Micronas GmbH provides easy-to-use PC software (based on LabVIEW™) for each supported sensor, which can be downloaded from the TDK-Micronas Service Portal upon registration (https://service.micronas.com).

1.4. Block Diagram of the MSP

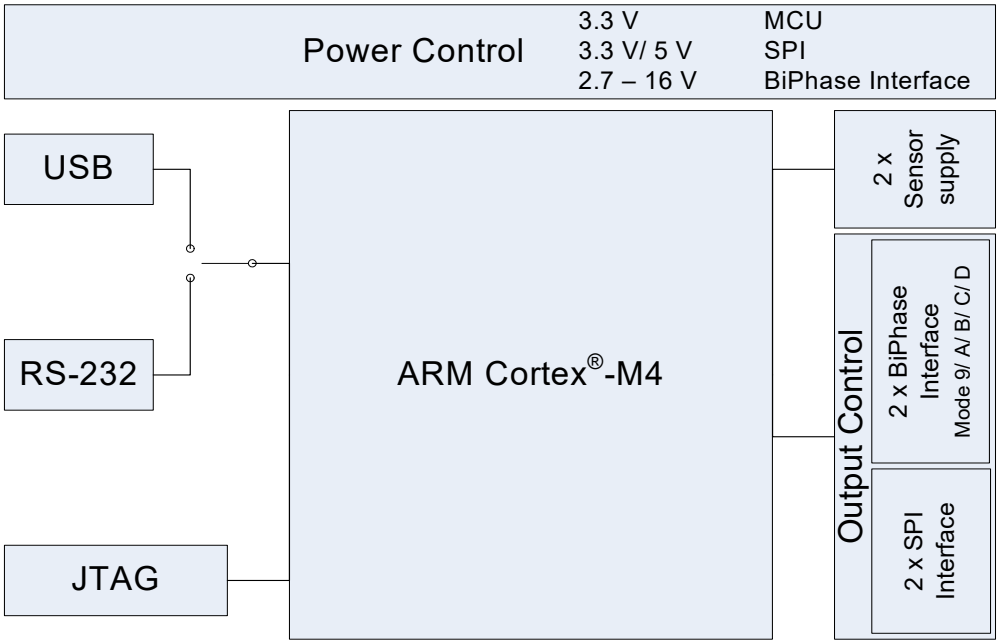


Fig. 1–2: MSP V1.x block diagram

2. Getting Started

2.1. USB Driver Installation

Note: The USB drivers do not need to be installed when using the RS-232 interface. They are only necessary when connecting the MSP to the PC via a USB cable.

Installing the USB VCP Drivers

Connect the MSP to a USB port.

Windows® 10 will automatically search for the latest driver if the PC is connected to internet. For previous Windows versions: If there are problems with the installation of FTDI Drivers, application notes are available on the web or in the “TDK-Micronas Service Portal (https://service.micronas.com)”[1].

More information can be found on the FTDI homepage as well.

Sometimes the installer repeats the whole installation procedure. You can open the device manager of your PC system to check if the system has configured the MSP correctly as shown in Fig. 2–3 as COM4.

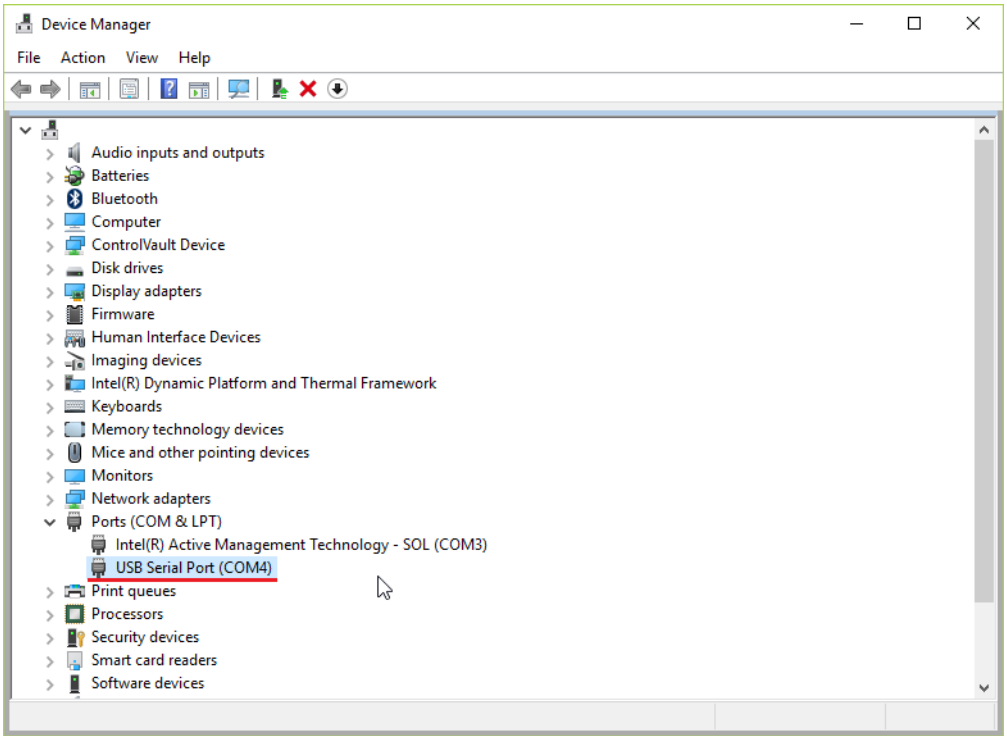


Fig. 2–3: Device manager

1) Application note “AN_119_FTDI_Drivers_Installation_Guide_for_Windows7.pdf” and “AN_104_FTDI_Drivers_Installation_Guide_for_WindowsXP.pdf” can be used to install the drivers on a Windows 7 or Windows XP operating system.

2.2. First Steps

2.2.1. Connect MSP V1.x

– Connect MSP via the USB/ RS-232 to the host computer and then connect MSP to the external supply voltage. Please follow this connection order in order to avoid communication problems.
– The MSP performs initialization and self test after power on. This is indicated by the BUSY LED. The MSP is ready for operation once the ERROR LED and BUSY LED are turned off.

2.2.2. Check Communication with PC and Sensor

Connect a TDK-Micronas sensor to one of the sockets of the MSP extension board.

Note: For the first communication check it is recommended to use the LabVIEW Programming Environment software provided by TDK-Micronas for the specific sensor.

Alternatively:

– set up a hyperterminal connection (see Section 5 on page 20),
– switch V_{S_SUP} on using the “vho1” command (see Section 6 on page 24),
– try to read out a register (see Operation Mode X of the used sensor type).

3. Magnetic Sensor Programmer V1.x Configuration

The description of the different status LEDs of the MSP is as shown in Table 3–2

Table 3–2: LEDs description

LED Name	Function
ERROR	On, in case of communication error
POWER	On, after power-on of the MSP
BUSY	On, when a command is sent to the MSP via the PC and the MSP is executing the command.

HAL Interface Connector

The HAL Interface connector on the MSP offers Biphase interface to the sensors. Depending on the sensor type, up to two sensors can be connected to the MSP. For this purpose, a 6-pin connector HAL 1/ 2 is provided.

The RJ25 - MMJ (part number: 940-SP-3066R-OST) corresponding to the fawn connector HAL 1/ 2 can be ordered from all leading electronic stores. The pinning of the interface is described in Table 3–3.

Table 3–3: Pin description of HAL 1/ 2 pin connector

Pin No.	Description
1	Sensor input V_{S_SUP} sensor 1
2	Common GND
3	Sensor output V_{S_OUT}/ DIO sensor 1
4	Sensor input V_{S_SUP} sensor 2
5	Common GND
6	Sensor output V_{S_OUT}/ DIO sensor 2

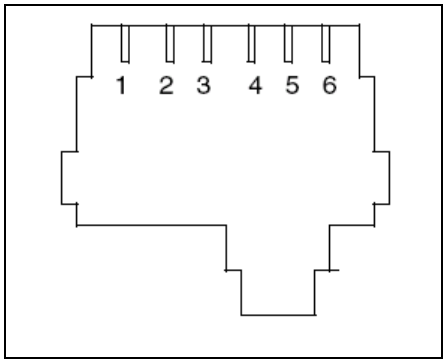


Fig. 3–4: Modular connector HAL 1/ 2, front view

DB-25 Connector

The DB-25 connector on the MSP offers both Biphase interface and SPI interface to the sensors. Depending on the sensor type up to two sensors which support SPI interface can be connected to the MSP. Additionally the connections from HAL interface connector are also made available on the DB-25 connector.

The DB-25 plug (part number: DB25P064HTXLF) corresponding to the DB-25 connector can be ordered from all leading electronic stores. The pin description of the DB-25 connector on MSP is shown in Table 3–4.

Table 3–4: Pin description of DB-25 Connector

Pin No.	Description
1	V_{EXT_OUT}
2	V_{S_SUP} sensor 1
3	Sensor output V_{S_OUT}/ DIO sensor 1
4	Sensor output V_{S_OUT}/ DIO sensor 2
5	V_{S_SUP} sensor 2
6	Not connected
7	Common GND
8	MOSI sensor 1
9	CS sensor 1
10	Common GND
11	MOSI sensor 2
12	MISO sensor 2
13	V_{EXT_OUT}
14	Not connected
15	Common GND
16	Not connected
17	Common GND
18	Not connected
19	Not connected
20	Not connected
21	CLK sensor 1
22	MISO sensor 1
23	CLK sensor 2
24	CS sensor 2
25	Common GND

3.1. Firmware Update

Regular firmware updates for TDK MSP V1.x are made available on the TDK-Micronas Service Portal. Please follow the below procedure to update the firmware:

1. Download the MSP V1.x firmware update zip file from the TDK-Micronas Service Portal. Extract the contents of the zip file to a folder. It contains an executable application for the Firmware update.
2. Replace the COM port number in the file name with the number of the COM port the MSP is connected to. Example: The application file name MSP_FW-v1.01MSP-COM1.exe is replaced by MSP_FW-v1.01MSP-COM28.exe as the MSP is connected to COM port number 28.
3. Run the executable application. A dialog box as shown in Fig. 3–5 appears informing the user about the current firmware version on the MSP and the inherent firmware version to which it will be updated. Clicking on Yes button starts the firmware update process.

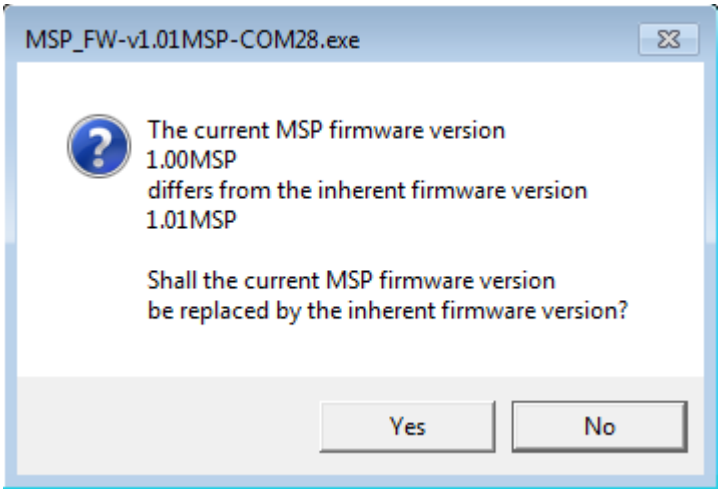


Fig. 3–5: Firmware update dialog box

Note: Do not disconnect the MSP or turn off the computer during the update process as this may lead to a permanent damage to the MSP.

4. During update the process' status is displayed and the user is informed once the update is successfully completed as shown in Fig. 3–6. After the update the busy LED on MSP is blinking. Reset the MSP to complete the firmware update procedure.

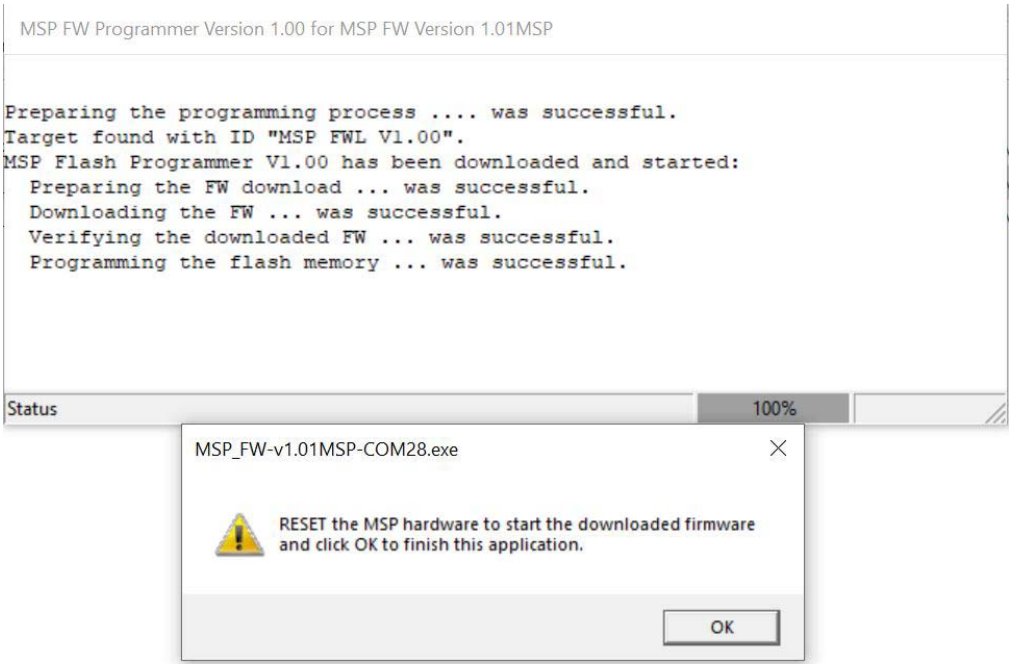


Fig. 3–6: Dialog box informing completion of Firmware update

4. Specification

4.1. Absolute Maximum Ratings

Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. All voltages are referenced to GND.

Table 4–5: Absolute maximum ratings

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_S	Supply voltage (external input)	17	24	32	V
I_S	Supply current (external input)	0.6	–	1	A
V_{S_SUP}	Output voltage	–	5	16	V
I_{S_SUP}	Output current (peak)	–	–	70	mA
T_A	Operating free-air temperature	0	25	55	°C
V_{ESD} [1)]	ESD protection on V_{S_SUP} & V_{S_OUT}				
	MSP V1.0	–	–	2	kV
	MSP V1.1	–	–	4	
	MSP V1.2	–	–	12	
1) According to IEC 61000-4-2 Standard					

4.2. Recommended Wiring

It is recommended to connect the customer's application to the MSP using shielded wires.

In order to minimize the risk of electromagnetic disturbances, the cable shall be as short as possible.

Note: Especially in noisy environments close to power switches and electromagnetic actuators, EMI-compliant layout of the wiring is mandatory.

For recommended cable parameters, please refer to Table 4–6.

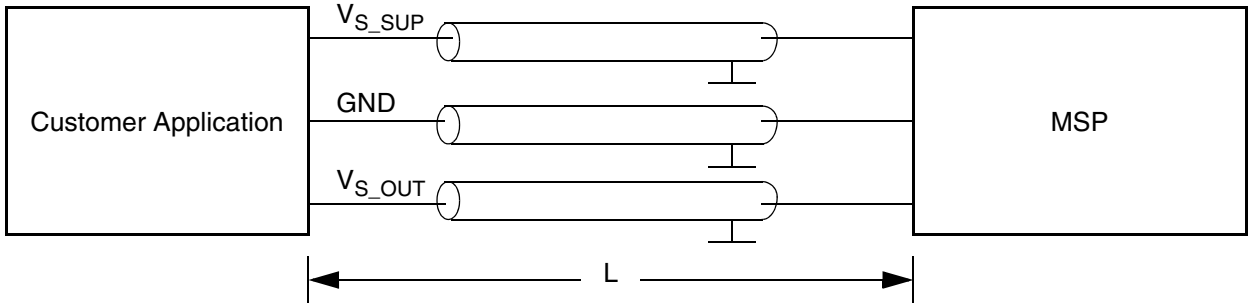


Fig. 4–7: Recommended wiring – schematic drawing

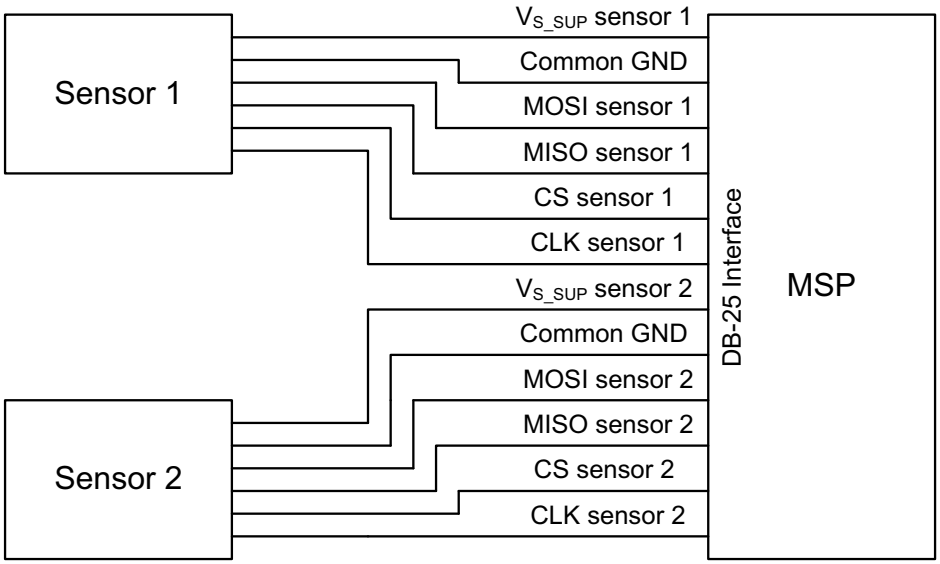


Fig. 4–8: Recommended wiring - SPI sensor

Table 4–6: Recommended cable parameters

Symbol	Parameter	Min.	Typ.	Max.	Unit	Conditions
R_0	Ohmic resistance per wire	–	1	5	Ω	I ≤ 10 mA
C_0	Capacitance	–	80	120	pF	
Z	Impedance	–	50	–	Ω	
L	Length	–	–	1[1)]	m	

1) SPI communication speeds up to 400 kHz have been successfully tested over a 1m cable. For higher speeds the cable length needs to be reduced accordingly.

4.3. Maintenance and Calibration

MSP maintenance or repair should not be carried out by the customer. In case of any problems or defects, please contact your supplier.

WARNING: Do not modify any part of the MSP. Otherwise, the MSP may be damaged, causing programming to the sensors to be inadequate and rendering the sensors unreliable.

4.4. Recommended Operating Conditions

All voltages are referenced to GND.

Table 4–7: Recommended operating conditions

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_S	Supply voltage (external input)	17.5	24	26	V
I_S	Supply current	0.8	–	1	A
T_A	Operating free-air temperature	–	25	–	°C
HAL 1/ 2					
V_{S_SUP}	Output voltage	2.7	5	14	V
I_{S_SUP}	Output current	–	20	60	mA
DB-25					
V_{EXT_OUT}	Supply voltage for external devices	–	18	–	V
I_{EXT_OUT}	Maximal output current for external devices	–	–	100	mA
V_{S_SUP}	Output voltage	2.7	5	14	V
I_{S_SUP}	Output current	–	20	60	mA
V_{SPI_X}	SPI operating voltage	–	3.3	5	V

5. Functions

5.1. Serial Command Interpreter

The MSP provides a serial command interpreter for interaction with a PC, connected via USB or RS-232.

The serial communication protocol applies a software handshake:

– The PC acts as a master, the MSP V1.x as a slave.

– The MSP V1.x responds to each master **COMMAND** frame with a **RESPONSE** frame.

Note: This document uses the following symbolization:
“==>” to denote the commands being sent by the PC to the MSP.
“<==” to denote the response sent by the MSP.

5.1.1. Serial Interface Configuration

When using hyperterminal communication please set the following parameters.

Table 5–8: Parameter settings of the serial interface

Parameter	Value
Bits per second	38400
Data bits	8
Parity	Even
Stop bits	1
Flow control	None

5.1.2. Definition of the COMMAND Frame

The COMMAND frame is of variable length. There are basically two types of commands:

1. Configuration of the MSP.

Example: to request firmware version from the MSP.
command ==> ?v LF

2. Communication with the connected Hall device.

Example: to read data from the sensor.
command ==> xxr08 LF

The command string has to end with a line feed - LF (ASCII code: 0x0A).

5.1.3. Definition of the RESPONSE Frame

Based on the COMMAND the RESPONSE frame consists of a status character followed by ':' and varying number of data characters (minimum 5) plus finishing carriage return (CR) and line feed (LF).

Example: <ST>:<Rn><Rn-1>....<R2><R1><R0> CR LF

ST (status character) = 0, if the command was successful.

ST != 0, in case of an error (see Table 5–9)

The Rx-characters contain the received data depending on the command (see mode dependent tables of commands in sections 7 to 12).

5.1.4. Analog Measurements

The MSP is equipped with a 9-bit resolution ADC that allows to measure voltages, such as the V_{S_SUP} or V_{S_OUT}. The V_{S_SUP} can be set to 5 or 6 V using the ftvdl command (see Table 6–11). The analog output voltage of a sensor is ratiometric to the sensor supply voltage V_{S_SUP}.

Example:

ftsad1 (enable the MSP ADC)
ftvdl0 (set V_{S_SUP} to 5 V)
ftana1 (measure V_{S_SUP})
Store received data to DATA1
ftana2 (measure V_{S_OUT})
Store received data to DATA2
ftsad0 (disable the MSP ADC)

V_{S_SUP} = DATA1 / 1024 x 3 x 5 V

V_{S_OUT} = DATA2 / 1024 x 5 V

5.1.5. Error Codes

Table 5–9: MSP error codes

STATUS[3:0]	Error
0	No error
1	Acknowledge error
2	2nd acknowledge error
3	Invalid command for selected mode
4	Reserved
5	Reserved
6	Reserved
7	No PWM detected (read PWM duty cycle command)
8	Reserved
9	Reserved
10	Reserved
11	No SENT detected (read SENT data command)
12	Reserved
13	Data read error
14	Invalid command parameter
15	Invalid command

5.2. Biphase Interface

HAL interface connector and DB-25 connector enable the MSP to be connected to magnetic sensors which support Biphase interface. The BiPhase signal is a digital signal which has binary data encoded in its phase.

A logical “0” is coded as no level change within the bit time. A logical “1” is coded as a level change at typically 50% of the bit time. After each bit time, a level change occurs (see Fig. 5–9).

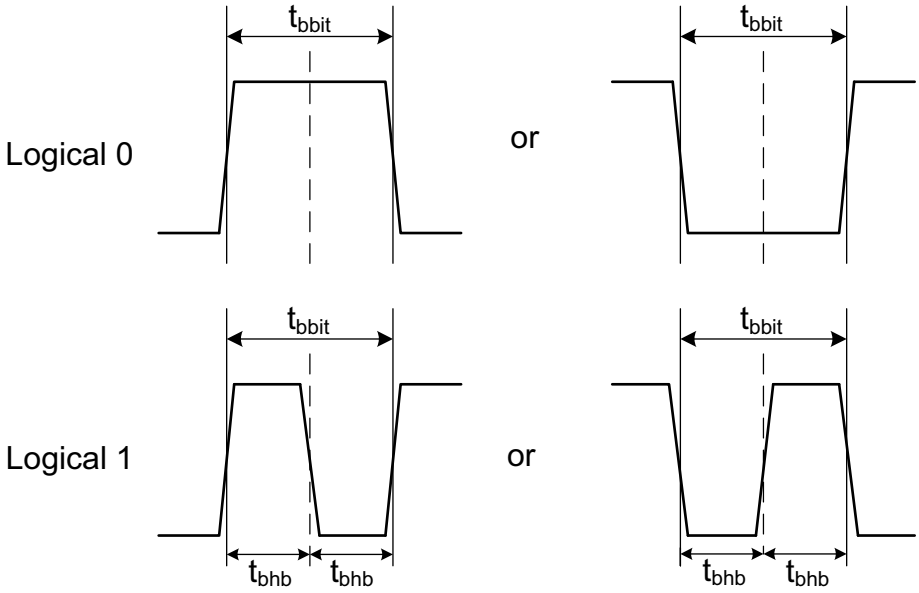


Fig. 5–9: Definition of logical 0 and 1 bit

Table 5–10: Biphase frame characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit
Biphase frame characteristic of the MSP					
t_{m_bbit}	MSP biphase bit time	100	1000	3400	µs
t_{m_bhb}	MSP biphase half bit time	0.45	0.5	0.55	t_{m_bbit}
Biphase frame characteristic of the sensor					
t_{s_bbit}	Sensor biphase bit time	0.75	1.0	1.25	t_{m_bbit}
t_{s_bhb}	Sensor biphase half bit time	0.25	0.5	0.75	t_{m_bbit}

6. Configuration Commands

The configuration commands shall be used to

– select the MSP operation mode based on the connected sensor
– read the MSP firmware version
– control the power supply V_{S_SUP} to the connected sensor

Note: Please refer to the respective MSP Modes for mode specific commands.

Table 6–11: Configuration commands

Action	Command	Parameter	Remarks
Get firmware version	?v	Return value: 0:v<V3><V2><V1><V0> MSP CR LF	V = firmware release version Example: ==> ?v <== 0:v1.00MSP
Get MSP hardware version[2)]	?hwv	Return value: 0:HWv<V5><V4><V3><V2><V1><V0> CR LF	V = MSP hardware version Example: ==> ?hwv <== 0:HWv1.0000
Show the available modes	?m	–	Returns a list of the available modes and respective commands
Set mode	sm<N>	N = 8 N = 9 N = A N = B N = C N = D Return value: <ST>:0000<N> CR LF[1)]	Switch MSP to device specific mode Example: ==> smA <== 0:0000A
Switch V_{S_SUP} on/ off	vho<X>	X = 0: Switch V_{S_SUP} off X = 1: Switch V_{S_SUP} on Return value: <ST>:0000X CR LF[1)]	Supply voltage on (default 5V; see ftvdl command below) Example: ==> vho1 <== 0:00001 Supply voltage off Example: ==> vho0 <== 0:00000

Table 6–11: Configuration commands, continued

Action	Command	Parameter	Remarks
For factory tests only			
Set low voltage level	ftvdl<X>	X = 0: V_{S_SUPL} = 5V X = 1: V_{S_SUPL} = 6V	Normal V_{S_SUP} Low level for Biphase-M protocol, if programming via VSUP Pin
Set high voltage level	ftvdh<X>	X = 0: V_{S_SUPH} = 5V X = 1: V_{S_SUPH} = 7.5V (if ftvdl0) X = 1: V_{S_SUPH} = 8V (if ftvdl1)	High level for Biphase-M protocol, if programming via VSUP Pin
Set programming voltage	ftvdp0	V_{S_SUP} = 5V (if ftvdl0 and ftvdh0)	
Force output voltage	ftsme<X>	X = 1: force output voltage to high state X = 0: release output	
Switch output pull-up resistor on/ off	ftpon<X>	X = 0: pull-up off X = 1: pull-up on	For open-drain devices a pull-up resistor is necessary
Select I/ O channel	ftses<N>	N = 1: sensor 1 N = 2: sensor 2 Return value: <ST>:00000<N> CR LF[1)]	Select the HAL I/O channel Example: ==> ftses1 <== 0:000001
Set bit time	sbt<BT>	BT = 000A... 0D48 (10 µs...3.4 ms) Return value: <ST>:00000 CR LF[1)]	Bit time in µs (4-digit hexadecimal number) 1 ms by default (BT = 0x03E8) Example: ==> sbt0064 <== 0:00000 0x0064 = 100 µs
Enable/ disable the MSP's ADC	ftsad<X>	X = 1: enable X = 0: disable Return value: <ST>:00000X CR LF[1)]	Enable ADC Example: ==> ftsad1 <== 0:000001
Measure V_{S_SUP} and V_{S_OUT}	ftana<X>	X = 1: measure V_{S_SUP} X = 2: measure V_{S_OUT} Return value: <ST>:<R4><R3><R2><R1><R0> CR LF[1)]	Measure the supply voltage with MSP's ADC R = Received ADC data (5-digit hexadecimal number) Example: ==> ftana1 <== 0:0019A 0x0019A = 410 410 / 1024 * 15V = 6V Measure the analog output voltage of sensor 1 with MSP's ADC Example: ==> ftana2 <== 0:00200 0x00200 = 512 512 / 1024 * 5V = 2.5V

Table 6–11: Configuration commands, continued

Action	Command	Parameter	Remarks
Read PWM period and pulse width	pr<N>	N = 0: falling edge N = 1: rising edge Return value: <ST>:<P4><P3><P2><P1><P0><W4> <W3><W2><W1><W0> CR LF[1)]	Read PWM period and pulse width with trigger on falling/ rising PWM edge P = period in 10^{-7}s (5-digit hexadecimal number) W = pulse width in 10^{-7}s (5-digit hexadecimal number) Example: ==> pr1 <== 0:013AE00A00 – Conversion of PWM period: 0x013AE = 5038 5038 * 10^{-7} s ≈ 0.50 ms – Conversion of pulse width: 0x00A00 = 2560 2560 * 10^{-7} s ≈ 0.26 ms – Calculation of duty-cycle: 2560 / 5038 ≈ 0.508 = 50.8%
Read asynchronous SENT fast channel	xxsf<STR>	STR = <T1><T0><S2><S1><S0> <N2><N1><N0><B0> T = SENT Tick time in x50 ns (2-digit hexadecimal number) S = reserved = 000 N = Number of consecutive SENT frames to be read (2-digit hexadecimal number). The internal MSP buffer can read/ store up to 250 SENT nibbles. Hence when SENT with two high speed channels (Mode H1/ H4) is being read then a maximum of 28 (0x1C) frames can be read and when SENT with single high speed channel (Mode H2) is being read then a maximum of 42 (0x2A) frames can be read. B = Number of nibbles (Status + Data + CRC) following the SENT sync pulse in a SENT frame (1-digit hexadecimal number) Return value: <ST>:<F1>:<F2>: .....:<Fn> CR LF[1)]	Read SENT fast channel data frames Fn = n^{th} fast channel SENT data frame, n ∈ {1,2,... N} Example: Read 5 consecutive SENT fast channel data frames with tick time 2.0 µs configured with two high speed channels Tick time = 2.0 µs : 2000 ns / 50 ns = 40 = 0x28 No. of frames to read = 0x005 No. of nibbles = 0x8 ==> xxsf280000058 <== 0:0C0EBB34:0C0EBC3A:0C0EBD38 :0C0EBE3E:0C0EBF3C 5 consecutive SENT frames are read. The SENT frames are separated by ":" character. Frame 1 = 0C0EBB34 – Consists of high speed data nibbles that follow the SYNC pulse in the SENT frame

Table 6–11: Configuration commands, continued

Action	Command	Parameter	Remarks
Read asynchronous SENT slow channel	xxss<STR>	STR = <T1><T0><N1><N0><S> T = SENT Tick time in x50 ns (2-digit hexadecimal number) N = Number of consecutive SENT slow channel frames to be read (2-digit hexadecimal number). The number of frames are restricted to a maximum value of 30 (0x1E) S = 0: Enhanced serial channel message format S = 1: Short serial message format Return value: <ST>:<F1>:<F2>: .....:<Fn> CR LF[1)]	Read SENT slow channel data frames Fn = n^{th} slow channel SENT data frame, n ∈ {1,2,... N} Example: Read 5 consecutive SENT enhanced serial channel data frames with tick time 2.0 µs Tick time = 2.0 µs : 2000 ns / 50 ns = 40 = 0x28 No. of enhanced serial frames to read = 5 ==> xxss28050\n <== 0:2902001:2A0241D:018000D :2B00121:2C0FA33 5 consecutive enhanced serial frames are read. The serial frames are separated by ":" character. Frame 1 = 2902001– Consists of enhanced serial channel message. 8 bit msg ID = 0x29 12 bit data = 0x020 6 bit CRC = 0x01
Get bit time	?bt	Return value: <ST>:<BT4><BT3><BT2><BT1><BT0> CR LF[1)]	BT = bit time in µs (5-digit hexadecimal number) Example: ==> ?bt <== 0:00064 0x00064 = 100 µs.
Get last acknowledge time	?ack	Return value: <ST>:<ACK4><ACK3><ACK2><ACK1> <ACK0> CR LF[1)]	ACK = width of last acknowledge pulse in µs (5-digit hexadecimal number) Example: ==> ?ack <== 0:00064 0x00064 = 100 µs

1) <ST> MSP status (see Table 5–9 on page 22 for details)

2) Available from MSP Firmware version v1.09 onwards.

In order to meet the different requirements of the various Hall devices, the MSP can be run in different operation modes. When a particular device is used, the corresponding mode has to be selected first. The mode list can be displayed by sending the command “?m”.

Table 6–12: MSP operating modes

Mode	Description	
8	HAL/ HAR 3900, CUR 42xy	– SPI
9	HAL 28xy	– Biphase via OUT Pin
A	HAL 1820 HAL 1880/ HAL 1890 HAL/ HAR 24xy HAL 36xy HAL/ HAR/ HAC 37xy HAL 38xy	– Biphase via VSUP Pin
B	HAL/ HAC 3980	– Biphase via VSUP Pin/ Biphase via current modulation
C	HAL 1880/ HAL 1890 HAL/ HAR 24xy HAL 36xy HAL/ HAR/ HAC 37xy HAL 38xy	– Biphase via OUT Pin
D	HAL/ HAR/ HAC 393x HAL/ HAR 392x CUR 42xy	– Biphase via OUT Pin

The operation modes are explained in the following chapters in detail.

7. Operation Mode 8

Operation mode 8 allows to communicate with sensor via SPI protocol. Detailed features and specifications are described in the respective sensor data sheet.

7.1. Operation Mode 8 – Configuration Commands

Table 7–13: Operation mode 8 – MSP configuration commands

Action	Command	Parameter	Remarks
Set board mode	sm8	Return value: <ST>:00008 CR LF[1)]	Switch board to Operation Mode 8
Set board mode for respective sensor	spisw<N>	N = 0: HAL/ HAR 3900; no CRC check by MSP. MSP board returns {Status, data and CRC} N = 1: reserved N = 2: reserved N = 3: CUR 42xy N = 4: HAL/ HAR 3900; Address byte for CRC is defined as {Last address[6:0], RWN_bit} Return value: <ST>:000000 CR LF[1)]	Sensors supported by Mode 8 have different communication protocols. This command sets the board to program respective sensors.
Set SPI voltage supply	spivs<N>	N = 0: Set V_{S_SUP} and SPI voltage level (CS, CLK, MOSI) to 3.3V N = 1: Set V_{S_SUP} and SPI voltage level (CS, CLK, MOSI) to 5V Return value: <ST>:0000<N> CR LF[1)]	V_{S_SUP} and SPI voltage levels (CS, CLK, MOSI) are default 3.3V when vho1 command is sent in Mode 8. Command “spivs” has to be used after switching V_{S_SUP} on.
Set voltage supply	svs<N>	N = 0: Set V_{S_SUP} to 5V N = 1: Set V_{S_SUP} to 8.3V N = 2: Set V_{S_SUP} to 3.3V Return value: <ST>:0000<N> CR LF[1)]	Change V_{S_SUP} level
Switch sensor to programming mode	pms	Return value: <ST>:000000 CR LF[1)]	This command switches HAL 3900 sensor from application to programming mode.
Set SPI clock frequency	spif<CLK>	CLK is a 4 digit hexadecimal number specifying SPI clock frequency in kHz. Return value: <ST>:000000 CR LF[1)]	MSP can be configured with following SPI clock frequencies using this command 10, 20, 30 . . . . . 90 kHz 100, 200, 300 . . . . . 900 kHz 1 MHz Example: To set SPI clock frequency as 1 MHz 1 MHz = 1000 kHz 1000 = 0x03E8 ==> spif03E8 <== 0:000000
1) <ST> MSP status (see Table 5–9 on page 22 for details)			

7.2. HAL/ HAR 3900

7.2.1. Available Command Frames

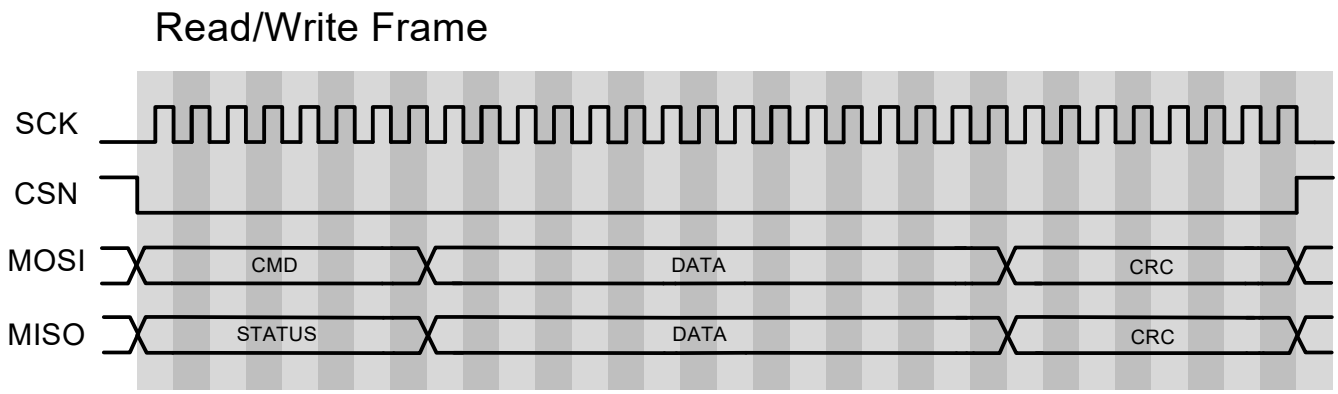


Fig. 7–10: HAL/ HAR 3900 communication frame structure via SPI

The HAL/ HAR 3900 SPI frame format is as shown in Fig. 7–10:

1. MSP pulls CSN low.
2. MSP sends one command byte followed by two master data bytes.
3. MSP sends a CRC byte.
4. HAL/ HAR 3900 replies in the next frame with one status byte, two slave data bytes and a slave CRC byte.

Note: Details on voltage levels and timing are explained in the programming guide for HAL/ HAR 3900 (SPI version).

The command byte (CMD) comprises of a 7-bit address (ADR) and a RWN bit. The RWN bit is the least significant bit (LSB) of the CMD byte and indicates if it is a read or a write command. In case of read command RWN is 1, the provided data on MOSI is ignored. In case of a write command RWN is 0, the provided data on MOSI is written to the register at address ADR after CRC check. If a read command is successfully received by the sensor then register data at address ADR is sent on MISO in the next frame.

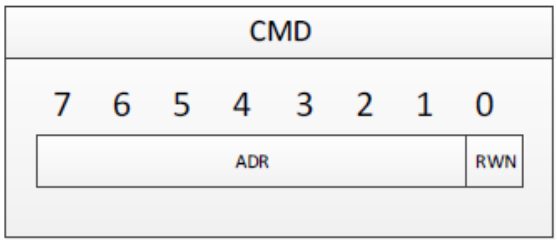


Fig. 7–11: Command (CMD) byte

7.2.2. Available Sensor Commands

The MSP supports 2 commands which provide read and write access to the sensor's memory (ROM, RAM register).

The write data frame and read data frame contain 7 address bits.

Table 7–14: Available commands

Command	CMD[0]	CMD[7:1]	DATA[15:0] (RD/ WD)
Read	1	address (0 to 127)	Dummy data (0x0000) is sent on MOSI.
Write	0	address (0 to 127)	Data which has to be written at the address is sent on MOSI.

7.2.3. Read

Each communication frame sent to/ received from the sensor consists of 32 bits. A read command frame starts with the MSP transmitting the CMD byte (7-bit address, RWN bit set to 1), followed by 16-bit master data and 8-bit master CRC. Next the slave sensor responds by transmitting a status byte, 16-bit slave data and 8-bit slave CRC. The structure of a read command frame and its response is shown in Fig. 7–12.

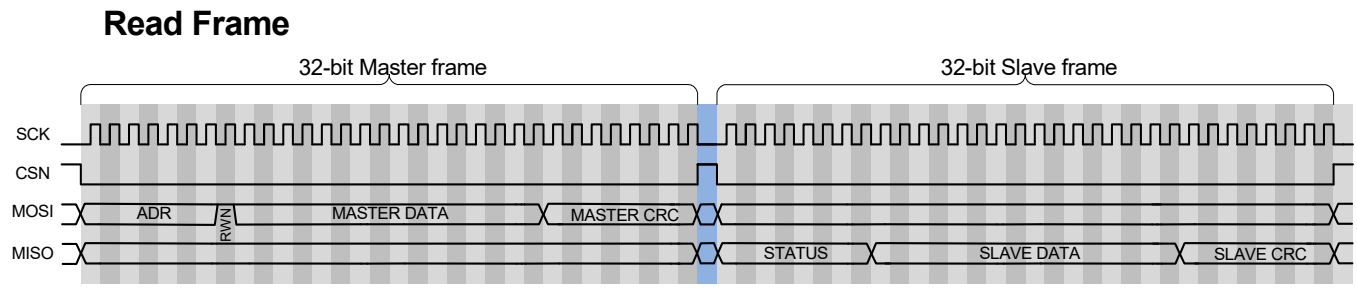


Fig. 7–12: Read command frame structure

7.2.4. Write

A successful switch into programming mode is required to send one or more write command frames to registers 0x00 to 0x6F. It is always possible to write to registers 0x70 to 0x7F. Each communication frame sent to/ received from the sensor consists of 32 bits. A write communication frame starts with the MSP transmitting the CMD byte (7-bit ADDRESS, RWN bit set to 0), followed by 16-bit master data and 8-bit master CRC. In the next frame the slave sensor responds with a status byte, 16-bit slave data, 8-bit slave CRC. The structure of a write command frame and its response is shown in Fig. 7–13.

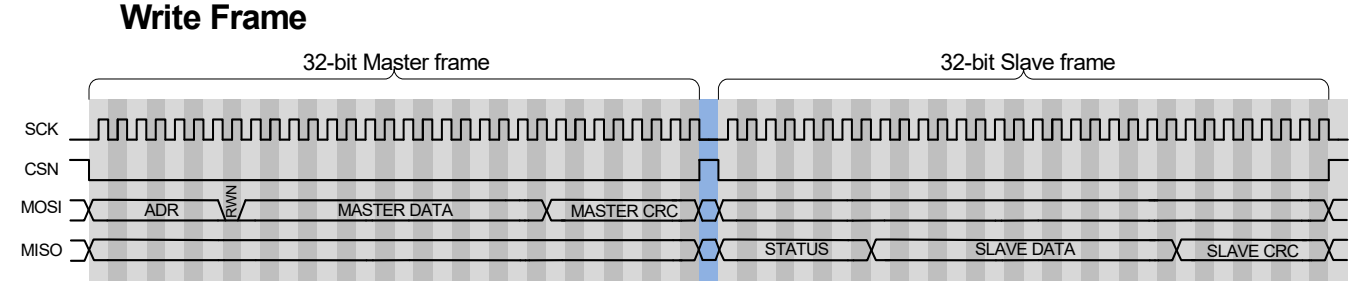


Fig. 7–13: Write command frame structure

7.2.5. CRC

The DATA bits are always followed by 8 CRC bits. There are two different cases for the calculation of the CRC byte:

1. Master CRC: For the read and the write command on MOSI, the CRC is calculated based on the following three bytes:

CMD[7:0]	MASTER DATA[15:8]	MASTER DATA[7:0]

2. Slave CRC: The byte-order (of four bytes) on which the CRC calculation for MISO is based on is shown below:

STATUS[7:0]	COMMAND[7:0]	DATA[15:8]	DATA[7:0]

The second byte is the COMMAND byte. This byte is structured as follows (last read address = ADR):

spisw2:

ADR[6] XOR ADR[5]	ADR[4] XOR RWN	ADR[3]	ADR[2]	ADR[1]	ADR[0]	0	0

spisw4:

ADR[6:0]	RWN

The polynomial for the CRC calculation is $X^8 + X^4 + X^3 + X^2 + 1$ (0x1D), with a seed value of 0xFF (CRC-8-SAE-J1850).

An invalid CRC indicates a detected transmission error. The code for the CRC calculation is shown on the next page.

```
 const uint8_t CRC_POLY = 0x1D; // x^8+x^4+x^3+x^2+1
uint8_t crc8 (unint8_t *data, int8_t length){
   int32_t i, bit;
   uint32_t crc = 0xFF;
   for (i = 0; i < length; i++)
   {
        crc = crc ^ data[i];
        for (bit = 0; bit < 8; bit++)
        {
             if (crc & 0x80)
             {
                  crc = crc << 1;
                  crc = crc ^ CRC_POLY;
                  crc = 0xFF & crc;
             }
             else
                  crc = crc << 1;
        }
   }
   return crc = (~crc) & 0xFF;
}
```

7.3. CUR 42xy

7.3.1. General SPI Frame

1. MSP pulls CSN low.
2. MSP sends one command byte followed by a number of master data bytes.
3. MSP sends a CRC byte calculated over the command and master data bytes.
4. If required, CUR 42xy replies with a number of slave data bytes followed by a slave CRC byte calculated over the slave data bytes.

Note: Details on voltage levels and timing are explained in the programming guide for CUR 42xy.

7.3.2. Available Commands

The MSP supports 2 commands which provide read and write access to the sensor's memory (ROM, RAM register).

Table 7–15: Available commands

Command	COMMAND (1 byte)	ADDRESS (1 byte)	DATA (2 bytes)
Read	0x3C	0x00 to 0x7F	MISO: Data read from address
Write	0x33	0x00 to 0x7F	MOSI: Data written to address

7.3.3. Read

A read command starts with the MSP sending command 0x3C, followed by the 8-bit address of the register and the CRC byte. The sensor responds with 16 bits of slave data followed by the slave CRC byte. The read frame is shown in Fig. 7–14.

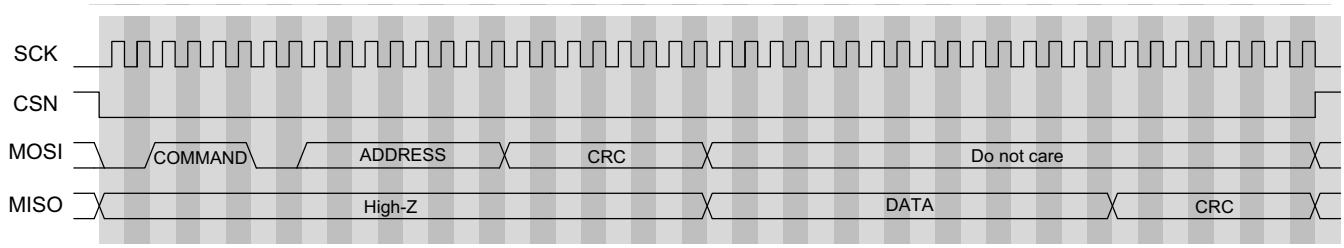


Fig. 7–14: Read command frame structure

7.3.4. Write

A write command starts with the MSP sending command 0x33, followed by the 8-bit address of the register, 16 bits of write data and the CRC byte. The sensor provides no reply data. The write frame is shown in Fig. 7–15.

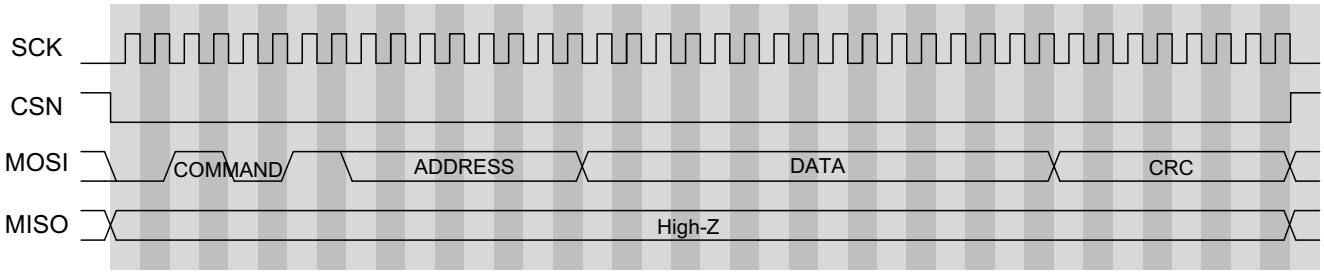


Fig. 7–15: Write command frame structure

7.3.5. CRC

The CRC is calculated over the command and data bytes (shifted MSB first) with polynomial $x^8 + x^2 + x + 1$ and initialized to 0xFF. The CRC is calculated in the same way on the returned slave data bytes. The c-code for the CRC calculation is shown below where “data” is an array of all bytes and “length” is the total number of bytes.

```
const uint8_t CRC_POLY = 0x07; // x^8+x^2+x+1
uint8_t crc8_spi (unint8_t *data, int8_t length){
  int32_t i, bit;
  uint32_t crc = 0xFF;
  for (i = 0; i < length; i++)
  {
      crc = crc ^ data[i];
      for (bit = 0; bit < 8; bit++)
      {
          if (crc & 0x80)
          {
              crc = ( crc & 0x7F ) << 1;
              crc = crc ^ CRC_POLY;
          }
          else
              crc = crc << 1;
      }
  }
  return crc;
}
```

7.4. Protocol Error Handling

In case of communication error with the sensor, the MSP indicates errors as listed in Table 5–9.

7.5. Mode 8 – Communication commands

Table 7–16: Mode 8 – MSP commands

Action	Command	Address	Data
Write data Sub-mode 0, 2 or 4 (spisw0/ spisw2/ spisw4)	xxw<STR>	STR = <A1><A0><D3><D2><D1><D0><CRC1><CRC0> LF Return value: <ST>:000000 CR LF[1)]	A = Address (2-digit hexadecimal number) D = Data (4-digit hexadecimal number) CRC = Checksum (2-digit hexadecimal number) Example: Write 0x0001 to address 0x49 ==> xxw49000137 <== 0:000000
Write data Sub-mode 3 (spisw3)	xxw<STR>	STR = <C1><C0><A1><A0><D3><D2><D1> <D0><CRC1><CRC0> LF Return value: <ST>:000000 CR LF[1)]	C = Command (2-digit hexadecimal number) A = Address (2-digit hexadecimal number) D = Data (4-digit hexadecimal number) CRC = Checksum (2-digit hexadecimal number) Example: Write 0x0001 to address 0x49 ==> xxw3349000137 <== 0:000000
Read data Sub-mode 2 or 4 (spisw2/ spisw4)	xxr<STR>	STR = <A1><A0> LF Return value: <ST>:<R3><R2><R1><R0><CRC1><CRC0> CR LF[1)]	A = Address (2-digit hexadecimal number) CRC = Checksum (2-digit hexadecimal number) R = Received data (4-digit hexadecimal number) Example: Read address 0x49 ==> xxr49 <== 0:0001F3

Table 7–16: Mode 8 – MSP commands, continued

Action	Command	Address	Data
Read data Sub-mode 0 (spisw0)	xxr<STR>	STR = <A1><A0> LF Return value: <ST>:<S1><S0><R3><R2><R1><R0><CRC1><CRC0> CR LF[1)] MSP returns the status byte, received data bytes and CRC byte from the slave MISO frame	A = Address (2-digit hexadecimal number) S = Status (2-digit hexadecimal number) R = Received data (4-digit hexadecimal number) CRC = Checksum (2-digit hexadecimal number) Example: Read address 0x49 ==> xxr49 <== 0:110001F3
Read data Sub-mode 3 (spisw3)	xxr<STR>	STR = <C1><C0><A1><A0><CRC1><CRC0> LF Return value: <ST>:<R3><R2><R1><R0><CRC1><CRC0> CR LF[1)] MSP returns the received data bytes and CRC byte on the MISO line	C = Command (2-digit hexadecimal number) A = Address (2-digit hexadecimal number) CRC = Checksum (2-digit hexadecimal number) R = Received data (4-digit hexadecimal number) CRC = Checksum (2-digit hexadecimal number) Example: Read address 0x49 ==> xxr3C4912 <== 0:0001F3
1) <ST> MSP status (see Table 5–9 on page 22 for details)			

8. Operation Mode 9

Operation Mode 9 allows to communicate with the HAL 28xy family. Detailed features and specifications are described in the respective sensor data sheet.

8.1. Operation Mode 9 – Configuration Commands

Table 8–17: Operation mode 9 – MSP configuration commands

Action	Command	Parameter	Remarks
Set MSP Mode	sm9	Return value: <ST>:00009 CR LF[1)]	Switch MSP to Operation Mode 9
1) <ST> MSP status (see Table 5–9 on page 22 for details)			

8.2. Available Command Frames

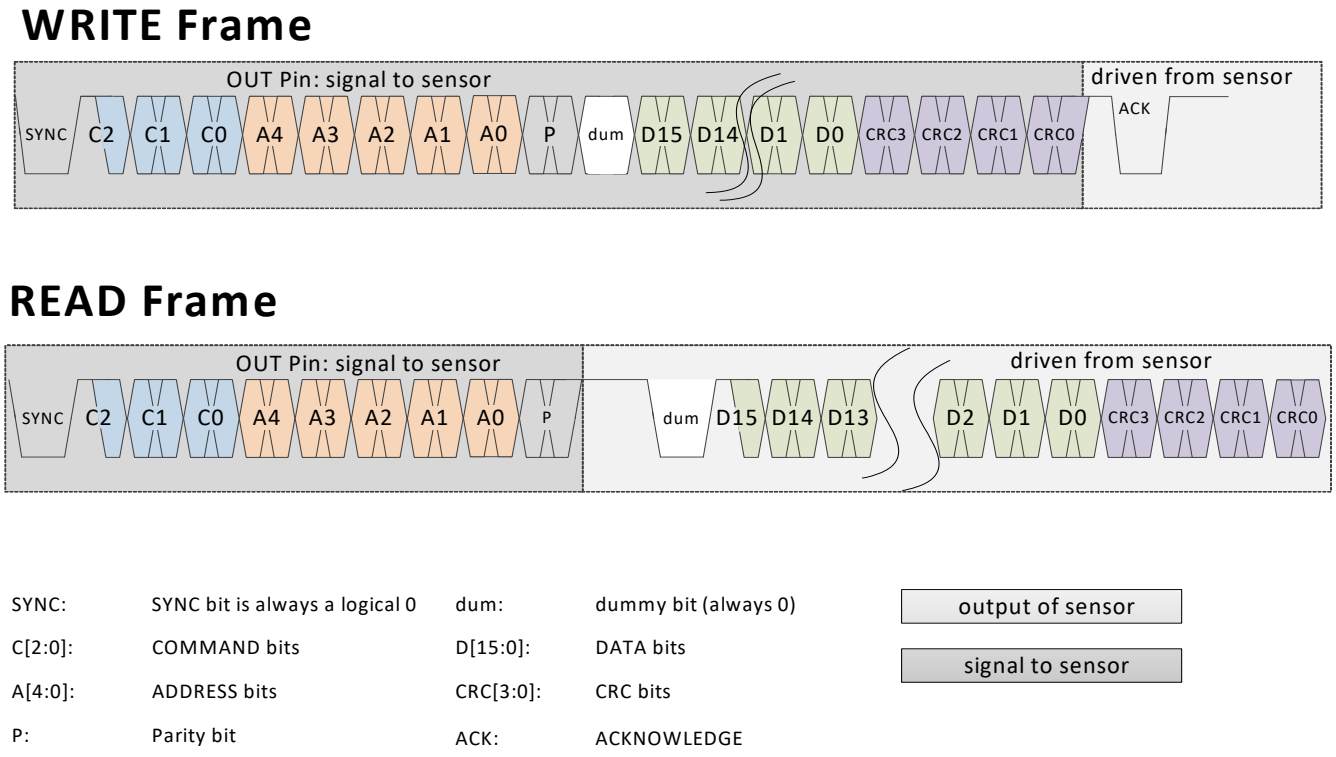


Fig. 8–16: Write and read frames

Note: Details on voltage levels and timing are explained in the programming guide of the HAL 28xy.

8.3. Available Sensor Commands

The MSP supports 5 commands which provide read and write access to the whole memory (ROM, RAM register). The mentioned commands allow for example to read the Hall value, the temperature value, and to program the EEPROM.

The write and read data frames contain 5 address bits only. A “set base address” command, which defines the base address, expands the accessible address range to 16 bits.

In case of an unknown command, the sensor transmits neither an acknowledge nor a body. (Each message consists of two parts: the message header, which contains information about the message body so that the recipient can interpret the message correctly, and the message body, which finally contains the payload.)

Table 8–18: Supported commands

Command	C[2:0]	Frame type	A[4:0]	D[15:8] (RD/WD)	D[7:0] (RD/WD)
Read with absolute address	0	Read	Absolute address (0 to 31, byte aligned)	Data read from address = A + 1	Data read from address = A
Read with base address	1	Read	Address offset (0 to 31, byte aligned)	Data read from address = base address + A + 1	Data read from address = base address + A
Set base address	3	Write	Do not care	Base address (16 bits, byte aligned)	
Write byte with base address	5	Write	Address offset (0 to 31, byte aligned)	Do not care	Data which is written to address = base address + A
Write word (16-bit) with base address	6	Write	Address offset (0 to 31, byte aligned)	Data which is written to address = base address + A +1	Data which is written to address = base address + A

8.3.1. Set Base Address

The “set base address” telegram functions as preparation for write telegrams and the “read with base address” telegram. It uses the write data frame. The base address is defined by D[15:0]. The sensor transmits an acknowledge if a communication error has not been detected.

8.3.2. Read with Absolute Address

The “read with absolute address” telegram uses the read data frame. The sensor transmits the data of the effective address after the header has been successfully received and the effective address is permitted. Otherwise, the sensor does not respond.

The effective address is defined by the address bits of the header (A[4:0]). Thus, this telegram can be used for reading the lower 32 byte only.

8.3.3. Read with Base Address

The “read with base address” telegram uses the read data frame. The sensor transmits the data of the effective address after the header has been successfully received and the effective address is permitted. Otherwise, the sensor does not respond. The effective address is calculated by the base address plus offset address. The offset address is defined by the address bits of the header (A[4:0]).

8.3.4. Write Byte with Base Address

The “write byte with base address” telegram uses the write data frame. The sensor saves the received byte (D[7:0]) to the calculated effective address and transmits an acknowledge after the header and body have been successfully received and the effective address is permitted. Otherwise, the command is discarded and the sensor does not transmit an acknowledge.

A “write byte with base address” telegram is also discarded while EEPROM programming.

8.3.5. Write Word with Base Address

This telegram is similar to the “write byte with base address” telegram. Unlike the “write byte with base address” telegram, this telegram is used for writing 16-bit data to the effective address and the effective address+1.

8.3.6. Special Cases

After reset, the IC does not execute a “read with base address”, “write byte with base address” or “write word with base address” command till a “set base address” telegram has been received.

During the EEPROM programming sequence (clear and set), a “write byte with base address” command or “write word with base address” command is discarded. It is recommended to pause the communication while the clear and set sequence is operated.

8.3.7. CRC

The data bits are always followed by 4 CRC bits. The CRC is calculated from the 16 data bits. The polynomial for the CRC calculation is $X^4 + X + 1$.

The code for the CRC calculation is shown on the next page.

```
int __stdcall calcCRC(int nData, int nSize)
{
        unsigned short bit_in, bit_out, bit_comp, crc;
        int i;
        crc = 0; /* initialize crc */
        for ( i = nSize - 1 ; i > -1 ; i-- )
        {
                bit_in = ( nData >> i ) & 0x1; /* extract input bit */
                bit_out = ( crc >> 3 ) & 0x1; /* extract bit b3 of crc */
                bit_comp = ( bit_out ^ bit_in ) & 0x1;
                crc = ( crc << 1 ) + bit_comp; /* calculate interim value of crc */
                crc = crc ^ ( bit_comp << 1 );
        }
        crc &= 0xF;
        return crc;
}
```

8.3.8. Protocol Error Handling

In case of a communication error with the sensor, the MSP indicates errors as listed in Table 5–9.

8.4. Mode 9 – Commands

Note: For general MSP configuration commands see Table 6–11 on page 24.

Table 8–19: Mode 9 – MSP commands

Action	Command	Parameter	Remarks
Mode switch[1)]	pcms	Return value: <ST>:00000 CR LF[2)]	Switch HAL 283x/HAL 2850 to programming mode Example: ==> pcms <== 0:00000
Read with absolute address Returns data read from address to address+1	pxr0<STR>	STR = <A1><A0> LF Return value: <ST>:<R3><R2><R1><R0><CRC> CR LF[2)]	A = Address (2-digit hexadecimal number) R = Received data (4-digit hexa-decimal number) CRC = Checksum (1-digit hexa-decimal number) Example: ==> pxr002 <== 0:0FFB2
Set base address	pxsb<STR>	STR = <A3><A2><A1><A0><CRC> LF Return value: <ST>:000000 CR LF[2)]	A = Base address (4-digit hexa-decimal number) CRC = Checksum (1-digit hexa-decimal number) Example: ==> pxsb30808 <== 0:000000
Read with base address Returns data from address to address+1	pxrb<STR>	STR = <A1><A0> LF Return value: <ST>:<R3><R2><R1><R0><CRC> CR LF[2)]	A = Address (2-digit hexadecimal number) R = Received data (4-digit hexa-decimal number) CRC = checksum (1-digit hexa-decimal number) Example ==> pxrb00 <== 0:D4537

Table 8–19: Mode 9 – MSP commands, continued

Action	Command	Parameter	Remarks
Write byte with base address	pxwb<STR>	STR = <A1><A0><D1><D0><CRC> LF Return value: <ST>:000000 CR LF[2)]	A = Address (2-digit hexadecimal number) D = Data (2-digit hexadecimal number) CRC = Checksum (1-digit hexa-decimal number) Example: ==> pxwb001E4 <== 0:000000
Write word with base address	pxww<STR>	STR = <A1><A0><D3><D2><D1><D0><CRC> LF Return value: <ST>:000000 CR LF[2)]	A = Address (2-digit hexadecimal number) D = Data (4-digit hexadecimal number) CRC = Checksum (1-digit hexa-decimal number) Example: ==> pxww00031E6 <== 0:000000

1) To switch back the sensor to application mode please switch V_{SUP} off (command vho0) and then switch VSUP on (command vho1)

2) <ST> MSP status (see Table 5–9 on page 22 for details)

9. Operation Mode A

Operation Mode A allows to communicate by means of the Biphase-M protocol via the VSUP Pin. Details on features and specification of the relevant devices are described in the respective sensor's data sheet.

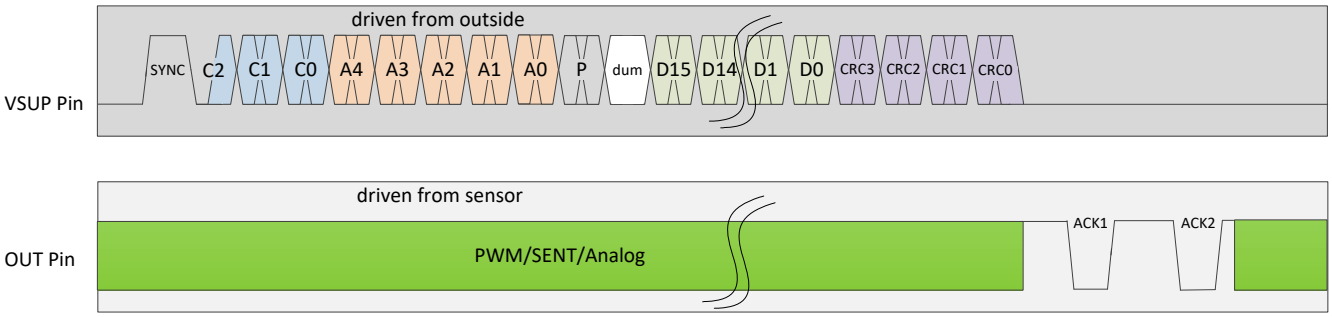
9.1. Operation Mode A – Configuration Commands

Table 9–20: Operation mode A – MSP configuration commands

Action	Command	Parameter	Remarks
Set MSP mode	smA	Return value: <ST>:0000A CR LF[1)]	Switch MSP to Operation Mode A
1) <ST> MSP status (see Table 5–9 on page 22 for details)			

9.2. Available Command Frames

WRITE Frame



READ Frame

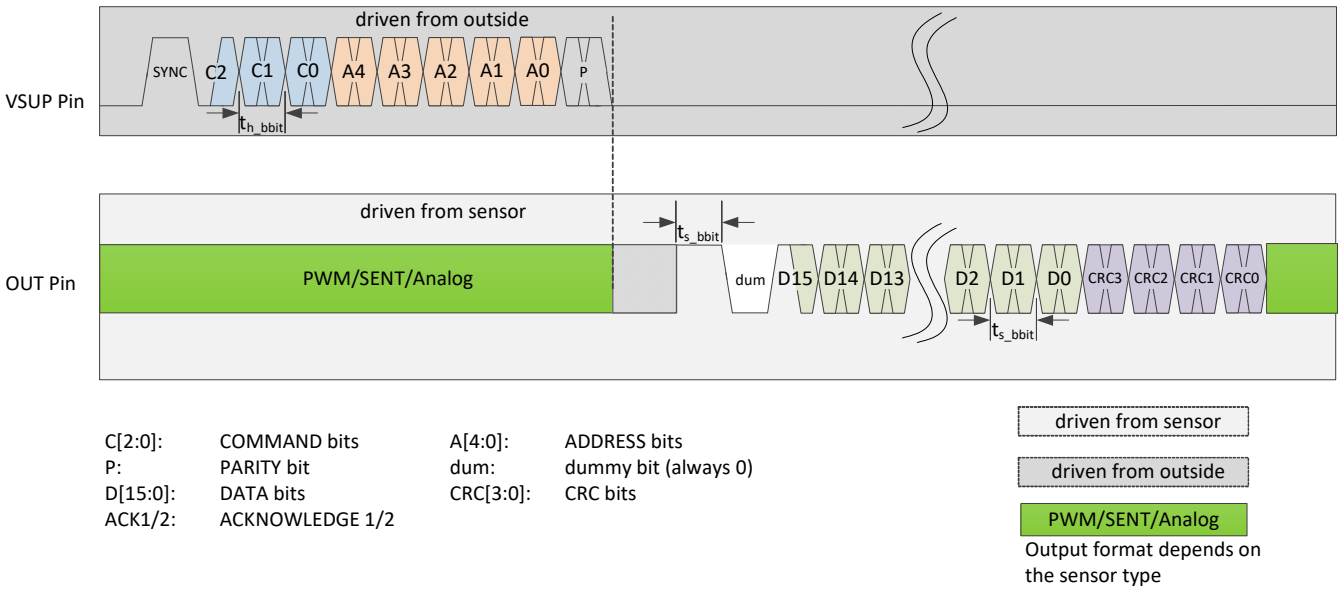


Fig. 9–17: Write and read frames

Note: Details on voltage levels and timing are explained in the programming guides respective to each product family.

9.3. Available Sensor Commands

The MSP supports 3 commands which provide read and write access to the sensor's memory. The mentioned commands allow for example to read the register value and to program the NVRAM.

The write data frame and read data frame contain 5 address bits.

Table 9–21: Available commands

Command	C[2:0]	Frame type	A[4:0]	D[15:0] (RD/WD)
Read	1	Read	Offset address (0 to 31)	Data read from address = ADR
Set base address	3	Write	Do not care	Base address 0,1,2,3 Not valid for HAL 18xy
Write	6	Write	Offset address (0 to 31)	Data which is written to address = ADR

9.3.1. Set Base Address

The set base address telegram functions as preparation for the write telegram and the read telegram. It uses the write data frame. Bits [15:2] are "do not care" bit 0 and bit 1 are concatenated to the address. MSP receives an Acknowledge from the sensor if the communication is successful.

9.3.2. Read

The read telegram uses the read data frame. The MSP receives the data of the address (A[4:0]) after the header has been successfully sent and the address is permitted. Otherwise, the sensor does not respond and the MSP indicates a data read error.

9.3.3. Write

The write telegram uses the write data frame. The sensor saves the received data to the calculated effective address and transmits an acknowledge after the header and body have been successfully received and the effective address is permitted. Otherwise, the command is discarded and the sensor does not transmit any acknowledge.

A write telegram is also discarded while NVRAM programming.

9.3.4. Protocol Error Handling

In case of communication error with the sensor, the MSP indicates errors as listed in Table 5–9.

9.3.5. Data Check

To allow data transmission in rough environments, two separate check mechanisms are implemented.

The command bits and the address bits are followed by a common parity bit as per description in the respective sensor's programming guide.

9.3.6. CRC Check

The data bits are always followed by 4 CRC bits. For all commands except read the CRC is calculated based on all protocol bits, including command, address, parity and data bits.

For the read command, the CRC is calculated based on dummy bit and data bits only D[15:0].

The polynomial for the CRC calculation is always $X^4 + X + 1$.

In case of correct command detection (parity, CRC and command address if applicable), ACK is sent.

Disrupted transfers can be retried by the master.

The code for the CRC calculation is shown below.

```
int __stdcall calcCRC( int nData, int nSize );
/* nData: 16 Bit ... 25 Bit binary */
/* nSize: 16 ... 25 */

int __stdcall calcCRC( int nData, int nSize )
{
        unsigned short bit_in, bit_out, bit_comp, crc;
        int i;
        crc = 0; /* initialize crc */
        for ( i = nSize ; i > -1 ; i-- )
        {
                bit_in = ( nData >> i ) & 0x1; /* extract input bit */
                bit_out = ( crc >> 3 ) & 0x1; /* extract bit b3 of crc */
                bit_comp = ( bit_out ^ bit_in ) & 0x1;
                crc = ( crc << 1 ) + bit_comp; /* calculate interim value of crc */
                crc = crc ^ ( bit_comp << 1 );
        }
        crc &= 0xF;
        return crc;
}
```

9.3.7. Parity Check

For the command and address bits, an “odd” parity check is used, i.e. the total number of 1's in the data plus parity bit must be an odd number. In the case of an even number of “1”s, the parity bit has to be “1”. In the case of on odd number of “1”s, the parity bit has to be “0”.

9.4. Mode A – Commands

Note: For general MSP configuration commands see Table 6–11 on page 24.

Table 9–22: Mode A – MSP commands

Action	Command	Parameter	Data
Set base address[1)]	xxsb<STR>	STR = <A1><A0><D3><D2><D1><D0><CRC> LF Return value: <ST>:000000 CR LF[2)]	A = Address (2-digit hexadecimal number) D = Data (4-digit hexadecimal number) CRC = Checksum (1-digit hexadecimal number) Example: Set base address 0x1 ==> xxsb000001D <== 0:000000
Write data to address	xxw<STR>	STR = <A1><A0><D3><D2><D1><D0><CRC> LF Return value: <ST>:000000 CR LF[2)]	A = Address (2-digit hexadecimal number) D = Data (4-digit hexadecimal number) CRC = Checksum (1-digit hexadecimal number) Example: Write 0xC000 to register 0x08 ==> xxw08C0008 <== 0:000000
Read data from address	xxr<STR>	STR = <A1><A0> LF Return value: <ST>:<R3><R2><R1><R0><CRC> CR LF[2)]	A = Address (2-digit hexadecimal number) CRC = Checksum (1-digit hexadecimal number) R = Received data (4-digit hexadecimal number) Example: Read address 0x08 ==> xxr08 <== 0:C000B
1) Not valid for HAL 18xy 2) <ST> MSP status (see Table 5–9 on page 22 for details)			

10. Operation Mode B

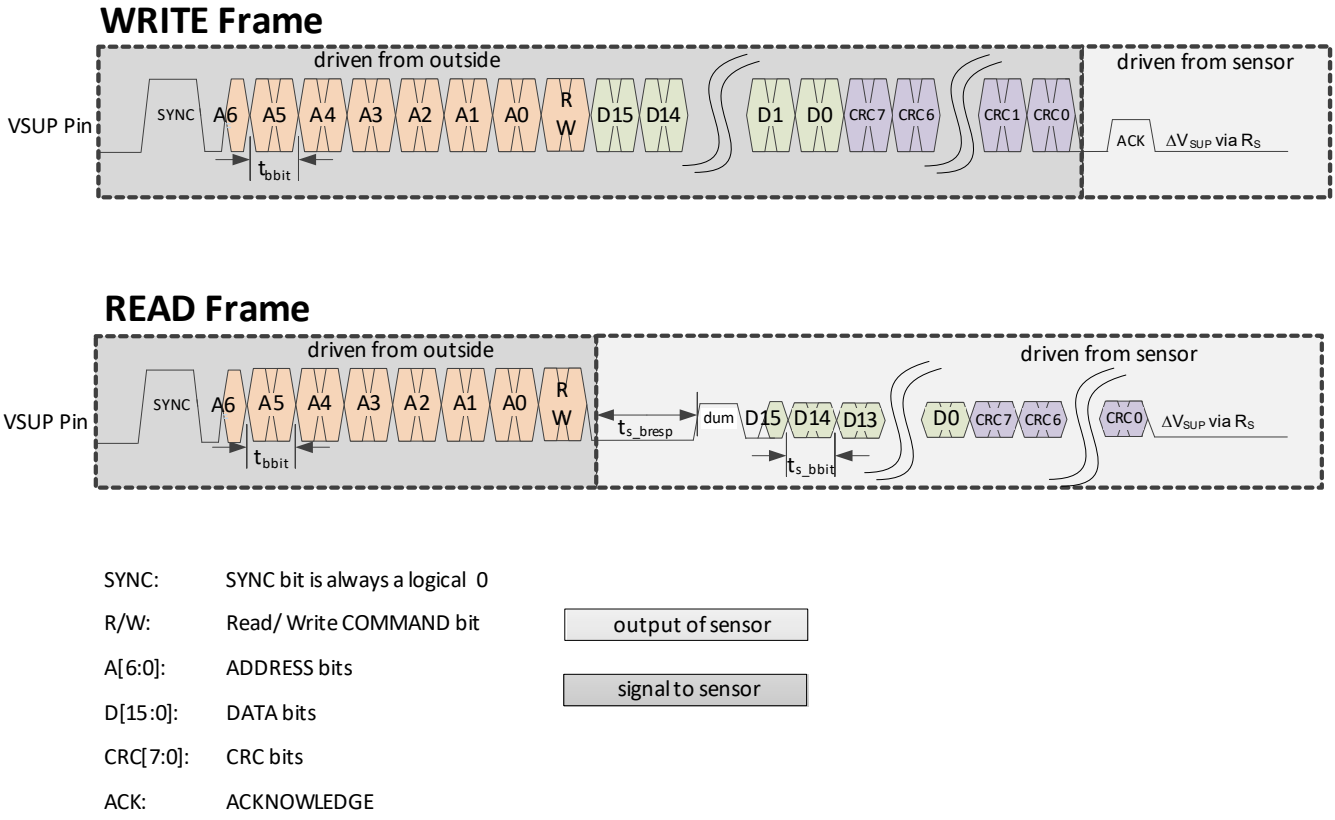
Operation Mode B allows to communicate with sensor via Vsup modulation and read sensor response sent by current modulation. Detailed features and specifications are described in the sensor data sheet.

10.1. Operation Mode B – Configuration Commands

Table 10–23: Operation mode B – MSP configuration commands

Action	Command	Parameter	Remarks
Set MSP mode	smB	Return <ST>:0000B CR LF[1)]	Switch MSP to operation Mode B
Switch to programming mode	pms	Return <ST>:000000 CR LF[1)]	
1) <ST> MSP status (see Table 5–9 on page 22 for details)			

10.2. Available Command Frames



SYNC:	SYNC bit is always a logical 0
R/W:	Read/ Write COMMAND bit
A[6:0]:	ADDRESS bits
D[15:0]:	DATA bits
CRC[7:0]:	CRC bits
ACK:	ACKNOWLEDGE

output of sensor

signal to sensor

Fig. 10–18: Write and read frames

The MSP writes the command frames via V_{SUP} voltage modulation and the sensor responds via current modulation over VSUP Pin. The current modulation can be detected as ΔV_{SUP} across the series resistor R_S.

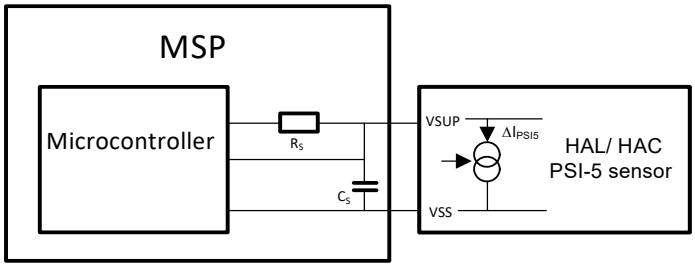


Fig. 10–19: PSI-5 Sensor interfacing with the MSP programmer

Note: Details on voltage levels and timing are explained in the programming guide of the sensor.

10.3. Available Sensor Commands

The MSP supports 2 commands which provide read and write access to the PSI-5 sensor memory.

The write data frame and read data frame contain 7 address bits.

Table 10–24: Available commands

Command	C[0]	A[6:0]	D[15:0] (RD/ WD)
Read	1	Address (0 to 127)	Data read from address A
Write	0	Address (0 to 127)	Data which is written to address A

10.3.1. Read

The read telegram uses the read data frame. The MSP receives the data of the effective address after the header has been successfully sent and the effective address is permitted. Otherwise, the sensor does not respond and the MSP indicates an error as listed in Table 5–9.

The address is defined by the address bits of the header (A[6:0]).

10.3.2. Write

The write telegram uses the write data frame. After successfully receiving the command header and the body, the sensor checks CRC, and if the address is permitted to be written, the data is written to the address and the MSP receives an acknowledge (ACK).

CRC Check

The data bits are always followed by 8 CRC bits. For the write command the CRC is calculated based on all protocol bits, including command, address, and data bits.

For the read command, CRC is calculated based on the following data:

A[6] XOR A[5]	A[4:0]	0	0	D[15:0]

The polynomial for the CRC calculation is $X^8 + X^4 + X^3 + X^2 + 1$ (CRC-8-SAE-J1850) with a seed value of 0xFF.

The code for the CRC calculation is shown below.

```
const unsigned char CRC_POLY = 0x1D; // X^8+X^4+X^3+X^2+1
int calcCRC (int data, int size)
{
   int i;
   unsigned char crc = 0xFF;
   for ( i = 0 ; i < size ; i++ )
   {
      if ( ( crc >> 7 & 0x1 ) != ( data >> ( size - 1 - i ) & 1 ) )
      {
         crc = crc << 1;
         crc ^= CRC_POLY;
         crc &= 0xFF;
      }
      else
         crc <<= 1;
   }
   return crc = ( ~crc ) & 0xFF;
}
```

10.3.3. Protocol Error Handling

In case of communication error with the sensor, the MSP indicates errors as listed in Table 5–9.

10.4. Mode B – Commands

Note: For general MSP configuration commands see Table 6–11 on page 24.

Table 10–25: Mode B – MSP commands

Action	Command	Address	Data
Write data	xxw<STR>	STR = <A1><A0><D3><D2><D1><D0><CRC1><CRC0> LF Return value: <ST>:00000 CR LF[1)]	A = Address (2-digit hexadecimal number) D = Data (4-digit hexadecimal number) CRC = Checksum (2-digit hexadecimal number) Example: Write 0x37B7 to address 0x08 ==> xxw0837B7EE <== 0:00000
Read data	xxr<STR>	STR = <A1><A0> LF Return value: <ST>:<R3><R2><R1><R0><CRC1><CRC0> CR LF[1)]	A = Address (2-digit hexadecimal number) CRC = Checksum (2-digit hexadecimal number) R = Received data (4-digit hexadecimal number) Example: Read address 0x08 ==> xxr08 <== 0:37B7C6
1) <ST> MSP status (see Table 5–9 on page 22 for details)			

11. Operation Mode C

Operation Mode C allows to communicate with the sensor by means of the Biphase-M protocol via the OUT Pin. Details on features and specification of the relevant sensors are described in the respective sensor data sheets.

11.1. Operation Mode C – Configuration Commands

Table 11–26: Operation mode C – MSP configuration commands

Action	Command	Parameter	Remarks
Set MSP mode	smC	Return value: <ST>:0000C CR LF[1)]	Switch MSP to Operation Mode C
Switch to listen mode	pgm	Return value: <ST>:000000 CR LF[1)]	Only valid for HAC 37xy and HAR 379x
1) <ST> MSP status (see Table 5–9 on page 22 for details)			

11.2. Available Command Frames

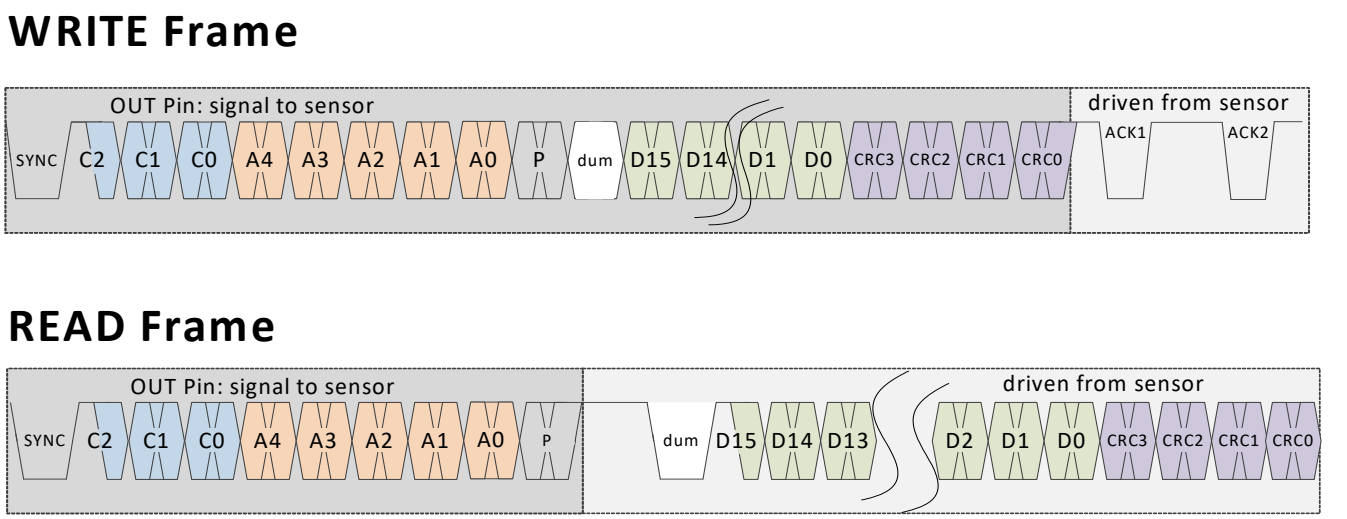


Fig. 11–20: Write and read frames

Note: Details on voltage levels and timing are explained in the programming guides respective to each product family.

11.3. Available Sensor Commands

The MSP supports 3 commands which provide read and write access to the sensor's memory.

The write data frame and read data frame contain 5 address bits. A set base address command, which defines a base address, expands the accessible address range to 8 bits.

Table 11–27: Available commands

Command	C[2:0]	Frame type	A[4:0]	D[15:0] (RD/WD)
Read	1	Read	Offset address (0 to 31)	Data read from address A
Set base address	3	Write	Do not care	Base address 0,1,2,3 Not valid for HAL 18xy
Write	6	Write	Offset address (0 to 31)	Data which is written to address A

11.3.1. Set Base Address

The set base address telegram functions as preparation for the write telegram and the read telegram. It uses the write data frame. Bits [15:2] are "do not care" bit 0 and bit 1 are concatenated to the address.The MSP receives an acknowledge if the communication is successful.

11.3.2. Read

The read telegram uses the read data frame. The MSP receives the data of the effective address after the header has been successfully sent and the effective address is permitted. Otherwise, the sensor does not respond and the MSP indicates this by error.

The effective address is calculated by the base address plus offset address. The offset address is defined by the address bits of the header (A[4:0]).

11.3.3. Write

The write telegram uses the write data frame. The sensor saves the received data to the calculated effective address and transmits an acknowledge after the header and body have been successfully received and the effective address is permitted. Otherwise, the command is discarded and the sensor transmits no acknowledge.

An incorrect write telegram can also be discarded while EEPROM or NVRAM programming.

11.3.4. CRC

The data bits are always followed by 4 CRC bits. For all commands except read, the CRC is calculated based on all protocol bits, including command, address, parity and data bits.

For the read command, the CRC is calculated based on the data bits only D[15:0].

The polynomial for the CRC calculation is always $X^4 + X + 1$.

In case of correct command detection (parity, CRC and command address if applicable), an ACK is sent as an answer.

The code for the CRC calculation is shown below.

```
int __stdcall calcCRC( int nData, int nSize );
/* nData: 16 Bit ... 25 Bit binary */
/* nSize: 16 ... 25 */

int __stdcall calcCRC( int nData, int nSize )
{
    unsigned short bit_in, bit_out, bit_comp, crc;
    int i;
    crc = 0; /* initialize crc */
    for ( i = nSize ; i > -1 ; i-- )
    {
       bit_in = ( nData >> i ) & 0x1; /* extract input bit */
       bit_out = ( crc >> 3 ) & 0x1; /* extract bit b3 of crc */
       bit_comp = ( bit_out ^ bit_in ) & 0x1;
       crc = ( crc << 1 ) + bit_comp; /* calculate interim value of crc */
       crc = crc ^ ( bit_comp << 1 );
    }
    crc &= 0xF;
    return crc;
}
```

11.3.5. Parity Check

For the command and address bits, an “odd” parity check is used. In the case of an even number of “1”s, the parity bit has to be “1”. In the case of an odd number of “1”s, the parity bit has to be “0”.

11.3.6. Protocol Error Handling

In case of communication error with the sensor, the MSP indicates errors as listed in Table 5–9.

11.4. Mode C – Commands

Note: For general MSP configuration commands see Table 6–11 on page 24.

Table 11–28: Mode C – MSP commands

Action	Command	Address	Data
Set base address[1)]	xxsb<STR>	STR = <A1><A0><D3><D2><D1> <D0><CRC> LF Return value: <ST>:000000 CR LF[2)]	A = Address (2-digit hexadecimal number) D = Data (4-digit hexadecimal number) CRC = Checksum (1-digit hexadecimal number) Example: Set base address 0x1 ==> xxsb000001D <== 0:000000
Write data	xxw<STR>	STR = <A1><A0><D3><D2><D1> <D0><CRC> LF Return value: <ST>:000000 CR LF[2)]	A = Address (2-digit hexadecimal number) D = Data (4-digit hexadecimal number) CRC = Checksum (1-digit hexadecimal number) Example: Write 0x37B7 to address 0x08 ==> xxw0837B76 <== 0:000000
Read data	xxr<STR>	STR = <A1><A0> LF Return value: <ST>:<R3><R2><R1><R0><CRC> CR LF[2)]	A = Address (2-digit hexadecimal number) CRC = Checksum (1-digit hexadecimal number) R = Received data (4-digit hexadecimal number) Example: Read address 0x08 ==> xxr08 <== 0:37B75

1) Not valid for HAL 18xy
2) <ST> MSP status (see Table 5–9 on page 22 for details)

12. Operation Mode D

Operation Mode D allows to communicate with the sensor by means of the Biphase-M protocol via the OUT Pin. Details on features and specifications are described in the respective sensor data sheet.

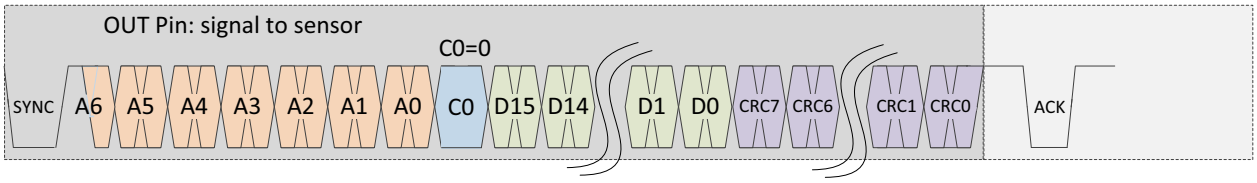
12.1. Operation Mode D – Configuration Commands

Table 12–29: Operation mode D – MSP configuration commands

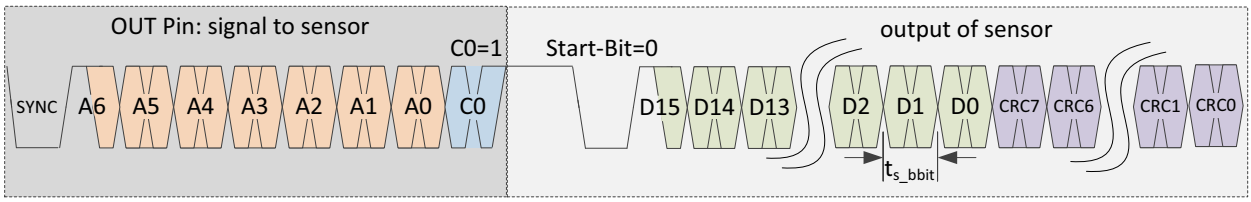
Action	Command	Parameter	Remarks
Set board mode	smD	Return value: <ST>:0000D CR LF[1)]	Switch board to Operation Mode D
Switch to listen mode	pgm	Return value: <ST>:000000 CR LF[1)]	Switch HAL/ HAR/ HAC 393x sensor to listen mode
Switch to programming mode	pms	Return value: <ST>:000000 CR LF[1)]	Switch HAL/ HAR/ HAC 393x sensor to programming mode
Change the over-current pulse polarity	ovcp<P>	P = 0: High pulse followed by low pulse (default) P = 1: Low pulse followed by high pulse Return value: <ST>:000000 CR LF[1)]	Change the polarity for the over-current pulse which is generated by pmsf command
Change the over-current pulse-width for pgm, pms pmsf commands	ovct<T>	T = 000A .... EA60 (10µs to 60ms) Return value: <ST>:000000 CR LF[1)]	T = Pulse-width in µs (4-digit hexadecimal number) Default Pulse-width for pgm/ pmsf command is 2ms Default Pulse-width for pms command is 30ms Example: Set pulse-width to 4ms ==> ovct0FA0 <== 0:000000 0x0FA0 = 4000µs = 4ms
Switch to programming mode	pmsf	Return value: <ST>:000000 CR LF[1)]	Switch HAL/ HAR 392x sensor to programming mode
Set voltage supply	svs<N>	N = 0: Set V_{S_SUP} to 5V N = 1: Set V_{S_SUP} to 8.3V N = 2: Set V_{S_SUP} to 3.3V Return value: <ST>:0000<N> CR LF[1)]	Change V_{S_SUP} level
Switch to programming mode	pmsc	Return value: <ST>:000000 CR LF[1)]	Switch CUR 42xy sensor to biphase programming mode
1) <ST> MSP status (see Table 5–9 on page 22 for details)			

12.2. Available Command Frames

WRITE Frame



READ Frame



SYNC:	SYNC bit is always a logical 0
C0:	COMMAND bit
A[6:0]:	ADDRESS bits
D[15:0]:	DATA bits
CRC[7:0]:	CRC bits
ACK:	ACKNOWLEDGE

output of sensor

signal to sensor

Fig. 12–21: Write and read frames

Note: Details on voltage levels and timing are explained in the programming guides of respective sensors.

12.3. Available Sensor Commands

The MSP supports 2 commands which provide read and write access to the sensor's memory.

The write data frame and read data frame contain 7 address bits.

Table 12–30: Available commands

Command	C[0]	A[6:0]	D[15:0] (RD/WD)
Read	1	Address (0 to 127)	Data read from address A
Write	0	Address (0 to 127)	Data which is written to address A

12.3.1. Read

The read telegram uses the read data frame. The MSP receives the data of the effective address after the header has been successfully sent and the effective address is permitted. Otherwise, the sensor does not respond and the MSP indicates error as per Table 5–9.

The address is defined by the address bits of the header (A[6:0]).

12.3.2. Write

The write telegram uses the write data frame. The sensor saves the received data and transmits an acknowledge (ACK) after the command and the CRC have been checked, the header and the body have been successfully received, and the effective address is permitted.

12.3.3. CRC

a) HAL/ HAR/ HAC 393x and HAL/ HAR 392x

The data bits are always followed by 8 CRC bits. For the write command the CRC is calculated based on all protocol bits, including command, address, and data bits.

For the read command, CRC is calculated based on the following data:

A[6] XOR A[5]	A[4:0]	0	0	D[15:0]

The polynomial for the CRC calculation is $X^8 + X^4 + X^3 + X^2 + 1$ (CRC-8-SAE-J1850), with a seed value of 0xFF.

The code for the CRC calculation is shown below.

```
const unsigned char CRC_POLY = 0x1D; // X^8+X^4+X^3+X^2+1
int calcCRC ( int data, int size )
{
   int i;
   unsigned char crc = 0xFF;
   for ( i = 0 ; i < size ; i++ )
   {
      if ( ( crc >> 7 & 0x1 ) != ( data >> ( size - 1 - i ) & 1 ) )
      {
         crc = crc << 1;
         crc ^= CRC_POLY;
         crc &= 0xFF;
      }
      else
         crc <<= 1;
   }
   return crc = ( ~crc ) & 0xFF;
}
```

b) CUR 42xy

The CRC is calculated with the polynomial $X^8 + X^4 + X^3 + X^2 + 1$ (0x1D), an initial value of 0xFF and result inversion (CRC-8-SAE-J1850). For write and read frames the CRC is calculated over the address, command and data bits {address[6:0], rwn[0], data[15:0]}, shifted MSB first. An example of the CRC calculation in c-code is shown on the next page, where “data” is an array of all bytes and “size” is the total number of bytes.

```
const uint8_t CRC_POLY = 0x1D; // x^8+x^4+x^3+x^2+1
uint8_t crc8_bp (uint8_t *data, uint8_t length)
{
   uint8_t byte, bit, crc = 0xFF;
   for (byte = 0; byte < length; byte++)
   {
      crc = crc ^ data[byte];
        for (bit = 0; bit < 8; bit++)
        {
           if (crc & 0x80)
           {
                crc = (crc & 0x7F) << 1;
                crc = crc ^ CRC_POLY;
           }
           else {
                   crc = crc << 1;
           }
        }
   }
   return crc ^ 0xFF;
}
```

12.3.4. Protocol Error Handling

In case of communication error with the sensor, the MSP indicates errors as listed in Table 5–9.

12.4. Mode D – Commands

Table 12–31: Mode D – MSP commands

Action	Command	Address	Data
Write data	xxw<STR>	STR = <A1><A0><D3><D2><D1><D0><CRC1><CRC0> LF Return value: <ST>:00000 CR LF[1)]	A = Address (2-digit hexadecimal number) D = Data (4-digit hexadecimal number) CRC = Checksum (2-digit hexadecimal number) Example: Write 0x37B7 to address 0x08 ==> xxw0837B7EE§ <== 0:00000
Read data	xxr<STR>	STR = <A1><A0> LF Return value: <ST>:<R3><R2><R1><R0><CRC1><CRC0> CR LF[1)]	A = Address (2-digit hexadecimal number) CRC = Checksum (2-digit hexadecimal number) R = Received data (4-digit hexadecimal number) Example: Read address 0x08 ==> xxr08 <== 0:37B7C6
1) <ST> MSP status (see Table 5–9 on page 22 for details)			

Note: For general MSP configuration commands see Table 6–11 on page 24.

13. Document History

1. Magnetic Sensor Programmer V1.0, Sept. 24, 2018; APN000136_001EN.
   First release of the application note.

2. Magnetic Sensor Programmer V1.0, May 22, 2019; APN000136_002EN.
   Second release of the application note.

   Major Changes:

   – Section 1.1 Certification added

   – Section 1.2 Support added

   – List of supported sensors updated

   – Table 6–11 MSP Configuration commands updated

   – Section 7 Operation Mode 8 added

   – HAL 1890 added

3. Magnetic Sensor Programmer V1.x, June 4, 2020; APN000136_003EN.
   Third release of the application note.

   Major Changes:

   – Document title changed to Magnetic Sensor Programmer V1.x

   – Table 6–11 Updated: the read asynchronous SENT fast channel command

   – Table 7–13 Updated: operation Mode 8 configuration commands spisw and svs

   – Section 7.2.5 Updated: parameters for CRC calculation

   – Section 7.3 Added: support for CUR 42xy in Mode 8

   – Table 7–16 Updated: operation Mode 8 read and write commands

   – Table 12–29 Updated: operation mode D configuration commands pms, pmsf, ovcp, ovct, svs and pmsc

   – Added HAL/ HAR 392x

TDK-Micronas GmbH
Hans-Bunte-Strasse 19 · D-79108 Freiburg · P.O. Box 840 · D-79008 Freiburg, Germany
Tel. +49-761-517-0 · Fax +49-761-517-2174 · www.micronas.com