TECHNICAL INFORMATION

Implementation of sHub® functionality in HIPERFACE DSL® servo system

Integration Manual

SICK
Sensor Intelligence.

This documentation applies to the HIPERFACE DSL® MASTER IP Core release version 1.07.

SICK STEGMANN GmbH
Dürrheimer Strasse 36
78166 Donaueschingen, Germany
Tel.: +(49) 771 / 807 – 0
Fax: +(49) 771 / 807 – 100
Internet: www.sick.com
E-mail: info@sick.com

Contents

1 List of figures

2 List of tables

3 Introduction

This document is intended for supporting the integration of the sHub® - Core version in a servo-drive system adopting the HIPERFACE DSL® technology.

Article number: 1107310, device: SHUB10-0ZA87010

The sHub® is used in combination with the following motor feedback systems:

Article number: 1106846, device: EDM35-0VF0A024A (non safe)

Article number: 1106851, device: EDM35-2VF0A024A (safe)

To be able to read out the additional data provided via the sHub® in the servo controller, the servo controller firmware must first have implemented this function. The servo controller application can access the sHub® data via the same interface (HDSL Master Interface) used for accessing the HDSL motor feedback system.

The following documents, detailing the HDSL Master Interface functionality, are related to this document:

- HIPERFACE DSL® Master Integration Manual 8017595
- HIPERFACE DSL® Master Safety Integration Manual 8017596
- Operating Instruction sHub® 8025330
- Product information sHub® 8023528

The steps to successfully implement the sHub® functionality in the servo motor and the servo controller are described below.

4 Integration in the servo motor

The sHub® shall be integrated in a servomotor by assembling it, as close as possible to the HDSL motor feedback system. The sHub® housing-shell is constituted by a top-cover carrying the electronic-card, assembled onto a base-plate, to be fixed to the motor-housing B-flange.

The mounting instructions 8025330 describe the correct assembly of the sHub® in the motor. The instructions in the mounting instructions must be followed.

5 Integration in the servo controller

The sHub® is a device specifically intended for acquiring and providing the servo controller application with additional sensor data synchronized with the position data of the Hiperface DSL® motor feedback system.

The Hiperface DSL® communication protocol allows the servo drive application for achieving a synchronous acquisition of an accelerometer-sensor data along with the measured shaft-position and the motor feedback system relevant status. Specifically, the sHub® acquires and processes the accelerometer-signal according to the HDSL basic framing (i.e: on H-Frame timing basis).

Based on that timing also the signal of the temperature-sensor of the motor-windings is acquired, but the subsequent data processing is not synchronously performed upon that framing; as it is not actually required. The temperature-sensor (shortly named "thermistor" throughout this manual) related data shall be provided just on-request basis.

NOTE

The sHub® functionality is normally (i.e.: upon system power-on) disabled; no sensor-data can be received; as well as, any access to the sHub® specific resources makes the servo-drive application stay hanging over a response, being the request not at all served.

The servo-drive application shall enable the sHub® functionality, by setting the relevant value (21h) onto the sHub® specific register at the motor feedback system; which is remotely accessible at address 50h, by using the "Short Message" transaction via the HDSL Master Interface.

The sHub® device is fully integrated into the Hiperface DSL® system, as shown hereafter:

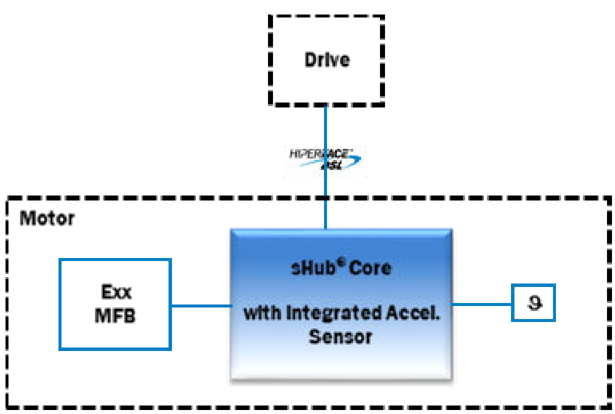


Figure 1: sHub® Functional Block Diagram

5.1 sHub®-Core functional interfaces

The basic sHub® version (i.e: Core) supports the following ports suitable for interfacing:

- HDSL-based servo-drive.
- HDSL-based Motor-feedback (sHub® -Ready).
- MEMS Accelerometer (1-axis) analog signal.
- Motor-winding Thermistor (KTY84-130, PT1000).

The power-supply is fed by the servo-drive via the HDSL link:

- Supply voltage: 7...12 VDC (measured at sHub® female connector of MFB)
- Power consumption (sHub®- Core, MFB and external-sensor inclusive): 2 W

5.1.1 Connectors

The sHub®- Core is interconnected via the connectors shown hereafter; each connector is suitable for a specific functional interface:

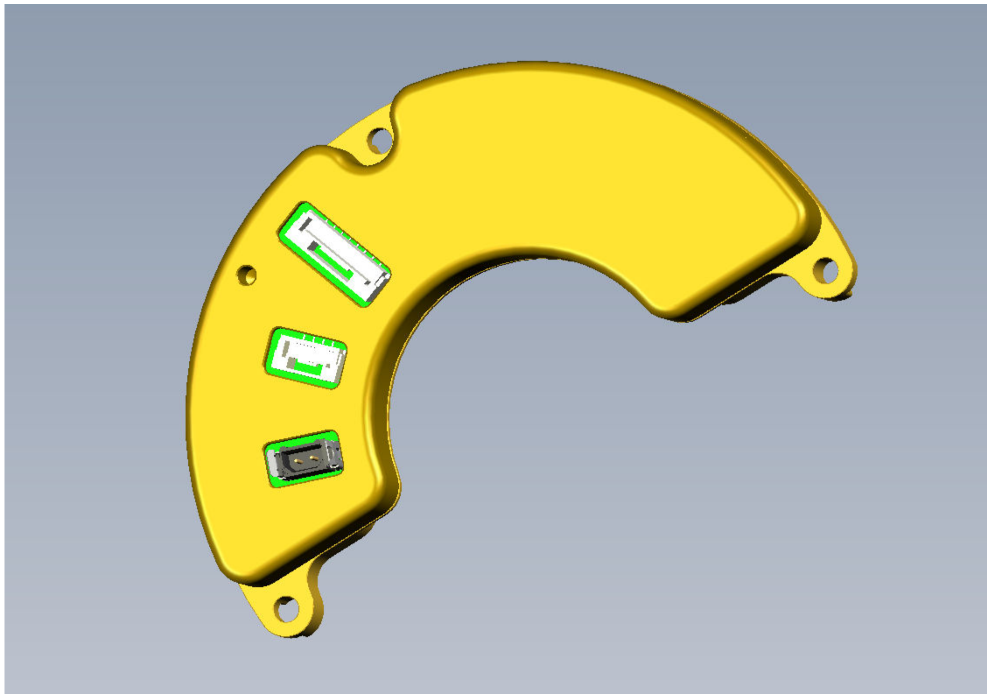


Figure 2: sHub® Connectors.

J1 1x8 MALE (JST BM08B-GHS-TBT)

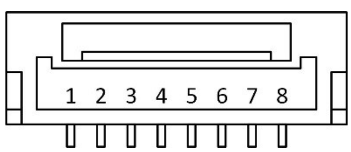
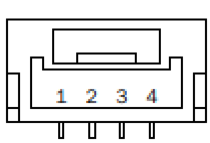


Table 1: Pin assignment JST BM08B-GHS-TBT

PIN	Signal	Explanation
1	MFB_US	7 ... 12 V supply of the connected MFB
2	MFB_GND	Ground of the connected MFB
3	MFB_DSLN	DSLN
4	MFB_DSLP	DSLP
5	MFB_RxD+	Differential communication between sHub®- Core and MFB
6	MFB_RxD-	Differential communication between sHub®- Core and MFB
7	MFB_TxD-	Differential communication between sHub®- Core and MFB
8	MFB_TxD+	Differential communication between sHub®- Core and MFB

J1 is for connecting the sHub® to the suitable HDSL Motor-feedback (EDx35 pro).

J2 1x4 MALE (JST BM04B-GHS-GB-TBT)



HDSL 2-wires:

Table 2: Pin assignment JST BM04B-GHS-GB-TBT

PIN	Signal	Explanation
1		Not connected
2	+Us/DSL+	7 ... 12 V supply
3	GND/DSL-	Ground
4		Not connected

Recommended outer diameter for set of stranded wires: 2.8 mm ± 0.3 mm
Recommended mating plug: JST (GHR-04V-S)
J2 is for connecting the HDSL-based servo-drive to the sHub®.

J3 1x2 MALE (HARWIN M80-8820242)

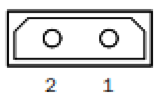


Table 3: Pin assignment HARWIN M80-8820242

PIN	Signal	Explanation
1	T+	Thermistor connection
2	T-	Thermistor connection (ground)

Recommended outer diameter for set of stranded wires: 2.2 mm ± 0.1 mm
Recommended mating plug: Harwin M80-8990205
J3 is for connecting the Motor-winding Thermistor to the sHub®.

NOTE
Both the pinout and the functionality of the same port as the one supported by the standard HDSL Motor-feedback fully apply.

5.2 sHub®-Core Ports for Sensors Acquisition

The sHub®- Core has an input port for specific sensors acquisition.

The sHub®- Core provides user with two acquisition channels, specifically suitable for acquiring and processing the analog-signals provided by the following sensors:

- MEMS Accelerometer integrated in sHub®- Core
- Motor-winding Thermistor.

The two acquisition channels are not suitable for acquiring any sensor else; they are neither interchangeable, being designed according to the specific sensor characteristics.

The following specifications strictly apply:

Table 4: sHub®- Core ports for External Sensors.

Connector	Function	Specification	Sensor
	Internal Acceleration Sensor	- Bandwidth: 0.5Hz...10kHz - Acceleration measuring range ±50g - Acceleration resolution 12.51 mg/digit - Digital data: 14bit (signed, 2's complement)	MEMS Accelerometer (1 axis, radial to motor axis)
J3	Analog Input	Digital data: 32bit (unsigned)	Motor Thermistor (KTY84-130, PT1000)

The acquisition channel for the MEMS Accelerometer shows a pass-band frequency response:

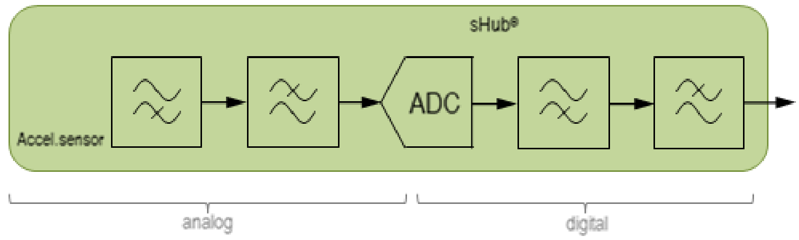


Figure 3: sHub®- Core acquisition channel for MEMS Accelerometer.

NOTE
The digital filtering parameters shall properly be adapted to the HDSL framing (sampling time). The servo-drive application can setup that as described herein at chapter 5.3.1.3.

5.3 sHub®-Core Functional Resources

The sHub® is a component of the HDSL System specifically intended for providing the servo-drive application with some remote I/O ports suitable for external devices (sensors and actuators), making them accessible via HDSL link

The sHub®- Core provides the servo-drive with two remote input ports.

The servo-drive can hence exploit the HDSL link also for remotely acquiring some specifc sensor-data. The sHub®- Core supports the following Input-only ports:

Table 5: sHub®- Core Input-only ports for External Sensor

Connector	Port #	Function	Sensor
		internal sensor	MEMS Accelerometer
J3	0	Analog Input	Thermistor

For details about the data acquisition relevant to the above listed sensors, refer to the chapter 5.3.2 and the chapter 5.3.3 respectively.

For handling the sensors specific acquisition, the sHub® provides the servo-drive application with a collection of specific Resources; to be accessed via the HDSL Long Message protocol; on the same way as for accessing the HDSL Motor-feedback specific Resources.

The sHub® specific collection includes also further Resources related to the identification, the diagnostics and the control of the sHub® itself, similarly to what the HDSL Motor-feedback provides.

5.3.1 sHub®- Core specific Resources database

The sHub®- Core is a component of the HDSL System providing the servo-drive application with a specific collection of Resources, which constitutes a branch extending the Resources database implemented by the HDSL Motor-feedback. Those sHub® specific Resources are implemented as leaves rooted to the sHub® specific node

The sHub®- Core specific collection of Resources is listed hereafter:

Table 6: sHub®- Core specific Resource Identifiers (RID's)

RID	Function	Reference into HDSL Implementation Manual (8017595)
006h	sHub® node	Chapter 8.3.7
200h	Access I/O	Chapter 8.11.1
201h	Manage I/O	Chapter 8.11.2
202h	Identify I/O	Missing reference
210h	Reset	Similar as the MFB RID 100h at chapter 8.8.1
218h	Reset to factory setting	Similar as the MFB RID 108h at chapter 8.8.6
280h	Device type	Similar as the MFB RID 080h at chapter 8.4.1
283h	Type code name	Similar as the MFB RID 083h at chapter 8.4.4
284h	Serial number	Similar as the MFB RID 084h at chapter 8.4.5
285h	Firmware revision number	Similar as the MFB RID 085h at chapter 8.4.5
286h	Firmware date	Similar as the MFB RID 086h at chapter 8.4.6
2C0h	Temperature range	Similar as the MFB RID 0C0h at chapter 8.5.1
2C4h	Supply Voltage range	Similar as the MFB RID 0C4h at chapter 8.5.5
2CBh	Lifetime	Similar as the MFB RID 0CBh at chapter 8.5.10
2CCh	Diary: error log	Similar as the MFB RID 0CCh at chapter 8.5.11

NOTE
The sHub® specific Resources can only be accessed on condition that the sHub® functionality is enabled; as described at chapter 5

5.3.1.1 RID 006h - sHub® Node

The sHub®- Core implements (in place of the sHub® -Ready motor-feedback) the sHub® specific node collecting all Resources relevant to the sHub® functionality. The sHub Node access reports the RID value of the first resource implemented in the sHub®-Core.

Read access by a Long Message direct transaction reports the resource definition, whose qualifiers are listed hereafter; the offset value of the Long Message shall be used for selecting the qualifier to be read:

Table 7: RID 006h: sHub® Node Definition

Defining qualifier	Offset	Value
RID		006h
Resource name	0	"SENSHUB"
Data size	1	2
Read access level	2	0
Write access level	3	15
Time overrun	4	70 ... 255
Data type	5	00h – Node indicator

Write access by a Long Message direct transaction shall not be granted. An attempt shall lead to an error indication

Read and/or write access by a Long Message indirect transaction depends on the access rights. Any denied or wrong access shall lead to error indications as detailed later on in the resource access description table

The offset value of the Long Message transaction shall be used by this resource in indirect transactions to identify the specifically addressed leaf. Specification of a leaf that is not implemented shall lead to an error indication.

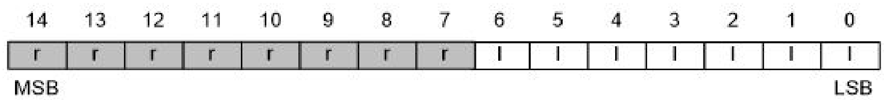


Figure 4: RID 006h: Offset Value Structure

Table 8: RID 006h: Offset Value Definition (Direct Read Access)

Signal	Value	Definition
r	-	Reserved
l	0 … 5	Return the qualifiers of the RID definition

Table 9: RID 006h: Offset Value Definition (Indirect Read Access)

Signal	Value	Definition
r	-	Reserved
l	0	Return the number of attached leaves
	1 … 127	Return the RID of 1st … 127th leaf

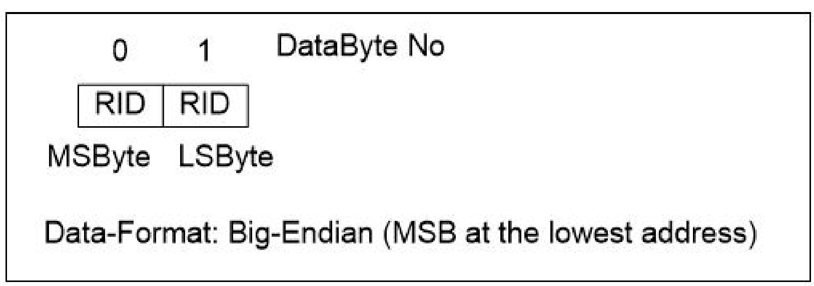


Figure 5: RID 006h: Data structure according to Data type (Indirect Read Access)

Table 10: RID 006h: Value Definition (Indirect Read Access)

Signal	Offset	Value	Definition
RID	0000h	0000h…007Fh	Number of attached leaves
	0001h…007Fh	0200h…02FFh	RID of requested leaf

Table 11: RID 006h: sHub® Node - Access Detailed Description

Offset	Length	Mode	Data-Bytes	Response	Description
0000h	8	Direct Read	-	“SENSHUB”	Resource name
0001h	2	Direct Read	-	2	Data size
0002h	2	Direct Read	-	0	Read access level

Offset	Length	Mode	Data-Bytes	Response	Description
0003h	2	Direct Read	-	15	Write access level
0004h	2	Direct Read	-	xxx	Time-out
0005h	2	Direct Read	-	0	Data type – node indicator
0006h... 7FFFh	0...8	Direct Read	-	Error: 13 40	Not allowed
0000h... 7FFFh	0...8	Direct Write	xx ...	Error: 14 41	Not allowed
0000h	2	Indirect Read	-	000Bh	Number of attached leaves (Access Level < 2)
0000h	2	Indirect Read	-	000Ch	Number of attached leaves (Access Level = 2)
0001h... 000Bh	2	Indirect Read	-	200h...2CCh	Resource ID of 1st... 11th leaf (Access Level < 2)
0001h... 000Ch	2	Indirect Read	-	200h...2CCh	Resource ID of 1st... 12th leaf (Access Level = 2)
000Ch... 7FFFh	2	Indirect Read	-	Error: 14 40	Not allowed
0000h... 7FFFh	0...8	Indirect Write	xx xx	Error: 10 41	Not allowed

5.3.1.2 RID 200h Access I, O

This resource allows the servo-drive application for accessing the remote I/O ports suitable for interfacing the external sensors.

Read access by a Long Message **direct transaction** reports the resource definition, whose qualifiers are listed hereafter; the offset value of the Long Message shall be used for selecting the qualifier to be read:

Defining qualifier	Offset	Value
RID		200H
Resource name	0	"ACCESSIO"
Data size	1	4
Read access level	2	0
Write access level	3	0
Time overrun	4	70 ... 255
Data type	5	05h – unsigned 32bit

RID 200h: Access I/O Definition

Write access by a Long Message direct transaction shall not be granted. An attempt shall lead to an error indication.

Read and/or write access by a Long Message **indirect transaction** shall instead be granted. However any wrong access parameter shall lead to error indications as detailed later on in the resource access description table.

NOTE
The write access shall be indicated as a wrong access, being only input-ports supported by the sHub®- Core

The sHub®- Core supports the following ports selection:

Table 12: sHub®- Core ports selection via Long Message Offset

Connector	Port #	LM Offset	Function	Sensor
Internal	1	0001h	Internal	MEMS Accelerometer
J3	0	0000h	Analog Input	Motor Thermistor

The offset value of the Long Message transaction shall be used by this resource in indirect transactions to select the supported input port number. Specification of a port that is not supported shall lead to an error indication.

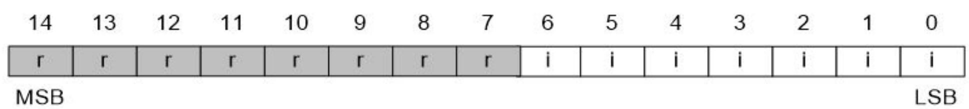


Figure 6: RID 200h: Offset Value Structure

Table 13: RID 200h: Offset Value Definition (Direct Read Access)

Signal	Value	Definition
r	-	Reserved
i	0...5	Return the qualifiers of the RID definition

Table 14: RID 200h: Offset Value Definition (Indirect Read Access)

Signal	Value	Definition
r	-	Reserved
i	0...1	Input port number

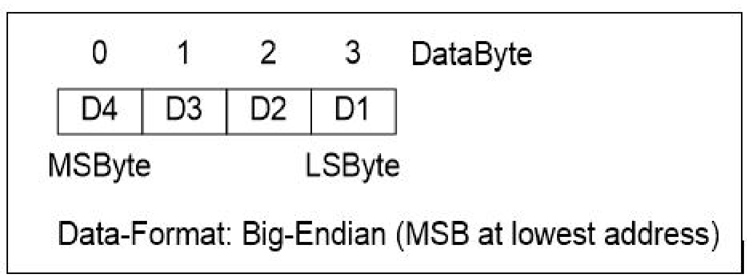


Figure 7: RID 200h: Data structure according to Data type (Indirect Read Access)

Table 15: RID 200h: Value Definition (Indirect Read Access)

Signal	Offset	Value	Definition
D4/D3	0001h	-	
D2/D1	0001h	E000h...1FFFh	Internal acceleration sensor - signed 16bit (2's complement), while only 14 bit hold value-able information - MEMS accelerome-ter processing output (-8192...+8191)

Signal	Offset	Value	Definition
D4/D3/D2/D1	0000h	00000000h... 000332C0h FFFFFFFFh	Analog Input [J3] - unsigned 32bit - winding-temperature sensor resistance (0... 209600 Ω) - Invalid measurement

Table 16: RID 200h: Access I/O - Access Detailed Description

Offset	Length	Mode	DataBytes	Response	Description
0000h	8	Direct Read	-	“ACCESSIO”	Resource name
0001h	2	Direct Read	-	4	Data size
0002h	2	Direct Read	-	0	Read access level
0003h	2	Direct Read	-	0	Write access level
0004h	2	Direct Read	-	xxx	Time-out
0005h	2	Direct Read	-	5	Data type - 32bit, unsigned
0006h... 7FFFh	0...8	Direct Read	-	Error: 13 40	Not allowed
0000h... 7FFFh	0...8	Direct Write	xx ...	Error: 14 41	Not allowed
0000h	4	Indirect Read	-	xx xx xx xxh	32bit digital value (ther-mistor)
0001h	4	Indirect Read	-	xx xx xx xxh	14bit digital value (accelerome-ter)
0002h... 7FFFh	0...8	Indirect Read	-	Error: 10 41	Not allowed
0000h... 7FFFh	0...8	Indirect Write	xx xx xx xx	Error: 10 41	Not allowed

5.3.1.3 RID 201h - Manage I, O

This resource allows the servo-drive application for configuring the remote I/O ports suitable for interfacing the external sensors.

The following configuration functions apply:

- Enable/disable thermistor readout filtering (low pass).
- Enable/disable thermistor readout conversion (PT1000 to KTY84-130).
- Accelerometer signal digital filtering parameters setting, according to the HDSL framing (sampling time).

Table 17: Accelerometer Digital Filtering Setup according to the HDSL Framing

HDSL framing (µs)	25	22.2	20	17.8	15.625	14	12.75	11.52 (free-run)
Digital Filter Setup	1	2	3	4	5 (Factory-Settings)	6	7	8

Read access by a Long Message **direct transaction** reports the resource definition, whose qualifiers are listed hereafter; the offset value of the Long Message shall be used for selecting the qualifier to be read:

Table 18: RID 201h: Manage I/O Definition

Defining qualifier	Offset	Value
RID		201h
Resource name	0	"MANAGEIO"
Data size	1	4
Read access level	2	0
Write access level	3	2
Time overrun	4	180 ... 255
Data type	5	14h - 4 byte structure

Write access by a Long Message direct transaction shall not be granted. An attempt shall lead to an error indication.

Read and/or write access by a Long Message **indirect transaction** shall be granted depending on the access rights. However any denied or wrong access parameter shall lead to error indications as detailed later on in the resource access description table.

The offset value of the Long Message indirect transaction shall be used for two purposes:

- Selection of the input port number;
  specifying a port that is not supported shall lead to an error indication.
- Selection of a special function (e.g.: KTY84-130);
  for configuring it (e.g.: KTY84-130 conversion on/off).

14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
r	r	r	r	r	r	r	r	i	i	i	i	i	i	i
MSB														LSB

Figure 8: RID 201h: Offset Value Structure (Direct Read Access)

Table 19: RID 201h: Offset Value Definition (Direct Read Access)

Signal	Value	Definition
r	-	Reserved
i	0...5	Return the qualifiers of the RID definition

14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
r	r	r	r	r	r	F	r	i	i	i	i	i	i	i
MSB														LSB

Figure 9: RID 201h: Offset Value Structure (Indirect Read/Write Access)

Table 20: RID 201h: Offset Value Definition (Indirect Read/Write Access)

Signal	Value	Definition
r	-	Reserved
F	1	KTY84-130 conversion selection
i	0...1	Input port number

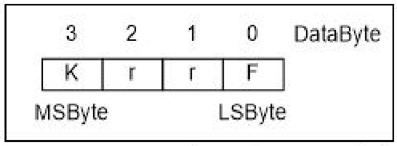


Figure 10: RID 201h: Data structure according to Data type (Indirect Read/Write Access)

Table 21: RID 201h: Value Definition (Indirect Read/Write Access)

Signal	Offset	Value	Definition
K	0001h	00h...FFh	Reserved
	0000h	1	KTY84-130 conversion enabled
		0	KTY84-130 conversion disabled
r	-	-	Reserved
F	0001h	1...8	Digital filtering setup
	0000h	65h...FFh	Reserved
		1 ... 100	Low-pass effect (100 = no effect)

Table 22: RID 201h: Manage I/O - Access Detailed Description

Offset	Length	Mode	DataBytes	Response	Description
0000h	8	Direct Read	-	"MANAGEIO"	Resource name
0001h	2	Direct Read	-	4	Data size
0002h	2	Direct Read	-	0	Read access level
0003h	2	Direct Read	-	2	Write access level
0004h	2	Direct Read	-	xxx	Time-out
0005h	2	Direct Read	-	20	Data type(14h) – structure with 4 bytes
0006h... 7FFFh	0...8	Direct Read	-	Error: 13 40	Not allowed
0000h... 7FFFh	0...8	Direct Write	xx ...	Error: 14 41	Not allowed
0000h	4	Indirect Read	-	xx 00 00 00	Read filter settings (temperature sensor)
0001h	4	Indirect Read	-	xx 00 00 00	Read filter settings (accelerom-eter)

Offset	Length	Mode	DataBytes	Response	Description
0002h... 00FFh	0...8	Indirect Read	-	Error: 14 40	Not allowed
0100h	4	Indirect Read	-	00 00 00 xx	Read status of KTY conversion
0101h...7FFFh	4	Indirect Read	-	Error: 14 40	Not allowed
0000h	4	Indirect Write	xx 00 00 00	-	Write filter settings (temperature sensor)
0001h	4	Indirect Write	xx 00 00 00	-	Write filter settings (accelerometer)
0002h... 00FFh	0...8	Indirect Write	xx xx xx xx	Error: 14 40	Not allowed
0100h	4	Indirect Write	xx 00 00 01 xx 00 00 00	-	Enable KTY emulation Disable KTY emulation
0101h...7FFFh	0...8	Indirect Write	xx xx xx xx	Error: 14 40	Not allowed

5.3.1.4 RID 202h - Identify I, O

This resource allows the servo-drive application for identifying the available remote I/O ports suitable for interfacing the external sensors.

Read access by a Long Message **direct transaction** reports the resource definition, whose qualifiers are listed hereafter; the offset value of the Long Message shall be used for selecting the qualifier to be read:

Table 23: RID 202h: Access I/O Definition

Defining qualifier	Offset	Value
RID		202h
Resource name	0	"IDENTIO"
Data size	1	8
Read access level	2	0
Write access level	3	15
Time overrun	4	70 ... 255
Data type	5	18h - 8 byte structure

Write access by a Long Message either direct or indirect transaction shall not be granted. An attempt shall lead to an error indication.

Read access by a Long Message **indirect transaction** shall instead be granted. However any wrong access parameter shall lead to error indications as detailed later on in the resource access description table.

The offset value of the Long Message indirect transaction shall be used for two purposes:

- for getting the amount of available I/O's.
- for selecting the I/O port number; specifying a port that is not supported shall lead to an error indication.

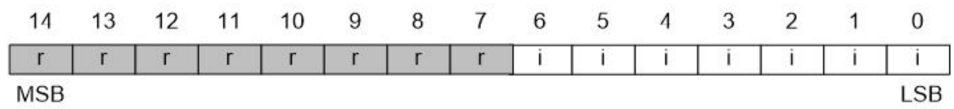


Figure 11: RID 202h: Offset Value Structure (Direct Read Access)

Table 24: RID 202h: Offset Value Definition (Direct Read Access)

Signal	Value	Definition
r	-	Reserved
i	0…5	Return the qualifiers of the RID definition

Table 25: RID 202h: Offset Value Definition (Indirect Read Access)

Signal	Value	Definition
r	-	Reserved
I	0	The amount of available I/O's is returned
	1…2	Selection of the 1st or the 2nd input port

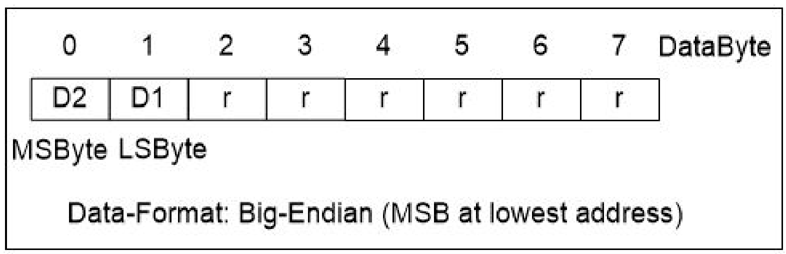


Figure 12: RID 202h: Data structure according to Data type (Indirect Read/Write Access @ Offset = 0)

Table 26: RID 202h: Value Definition (Indirect Read Access @ Offset = 0)

Signal	Value	Definition
D2/D1	2	The amount of input ports provided by sHub®-Core
r	-	Reserved

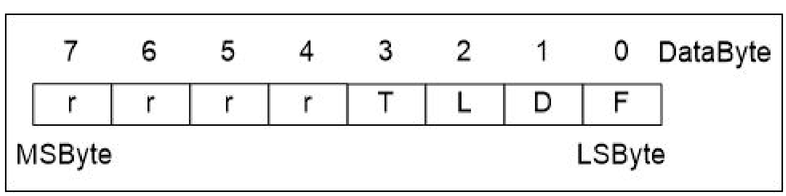


Figure 13: RID 202h: Data structure according to Data type (Indirect Read/Write Access @ Offset > 0)

Table 27: RID 202h: Value Definition (Indirect Read Access @ Offset > 0)

Signal	Value	Definition
r	-	Reserved
T	05h / 08h	Data type (5) -> unsigned 32bit; (8) -> signed 16bit
L	4…2	Data length (reported value range in bytes)
D	02h	Resistance (~210 kΩ full-scale)
	00h	Raw analog voltage conversion

Signal	Value	Definition
F	01h	Analog signal input

Table 28: RID 202h: Identify I/O - Access Detailed Description

Offset	Length	Mode	DataBytes	Response	Description
0000h	8	Direct Read	-	“IDENTIO”	Resource name
0001h	2	Direct Read	-	8	Data size
0002h	2	Direct Read	-	0	Read access level
0003h	2	Direct Read	-	15	Write access level
0004h	2	Direct Read	-	xxx	Time-out
0005h	2	Direct Read	-	24	Data type (18h) – structure with 8 bytes
0006h... 7FFFh	0...8	Direct Read	-	Error: 13 40	Not allowed
0000h... 7FFFh	0...8	Direct Write	xx ...	Error: 14 41	Not allowed
0000h	2	Indirect Read	-	00 02	Number of available I/ OsData type (18h) – structure with 8 bytes
0001h	4	Indirect Read	-	01 02 04 05	1st input specification (i.e.: Ther- mistor)
0002h	4	Indirect Read	-	01 00 02 08	2nd input specification (i.e.: Accelerome- ter)
0003h... 7FFFh	0...8	Indirect Read	-	Error: 14 40	Not allowed
0000h... 7FFFh	0...8	Indirect Write	xx ...	Error: 10 41	Not allowed

5.3.1.5 RID 210h - Reset

This resource shall trigger the same initialization sequence as on power-on. Termination of the reset sequence is not directly indicated.

NOTE
Upon the reception of the Reset command (RID 100h), the Motor-feedback triggers a reset also onto the sHub®.

Read access by a Long Message **direct transaction** reports the resource definition, whose qualifiers are listed hereafter; the offset value of the Long Message shall be used for selecting the qualifier to be read:

Table 29: RID 210h: sHub® Reset Definition

Defining qualifier	Offset	Value
RID		210h
Resource name	0	"SH_RESET"
Data size	1	0
Read access level	2	15
Write access level	3	0
Time overrun	4	180 … 255
Data type	5	01h – void (no data)

Write access by a Long Message direct transaction shall not be granted. An attempt shall lead to an error indication.

Read and/or write access by a Long Message **indirect transaction** depends on the access rights. Any denied or wrong access shall lead to error indications as detailed later on in the resource access description table.

The sHub® reset is triggered by a Long Message **indirect write transaction without data;** neither the offset value of the Long Message indirect transaction shall be relevant.

The offset value of the Long Message is just relevant for the Long Message direct transaction.

14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
r	r	r	r	r	r	r	r	i	i	i	i	i	i	i
MSB														LSB

Figure 14: RID 210h: Offset Value Structure (Direct Read Access)

Table 30: RID 210h: Offset Value Definition (Direct Read Access)

Signal	Value	Definition
r	-	Reserved
i	0…5	Return the qualifiers of the RID definition

Table 31: RID 210h: sHub® Reset - Access Detailed Description

Offset	Length	Mode	Data-Bytes	Response	Description
0000h	8	Direct Read	-	“SH_RESET”	Resource name
0001h	2	Direct Read	-	0	Data size
0002h	2	Direct Read	-	15	Read access level
0003h	2	Direct Read	-	0	Write access level
0004h	2	Direct Read	-	xxx	Time-out
0005h	2	Direct Read	-	1	Data type (01h) – void (no data)
0006h… 7FFFh	0…8	Direct Read	-	Error: 13 40	Not allowed
0000h… 7FFFh	0…8	Direct Write	xx …	Error: 14 41	Not allowed
0000h… 7FFFh	0…8	Indirect Read	-	Error: 11 41	Not allowed

Offset	Length	Mode	Data-Bytes	Response	Description
0000h... 7FFFh	0	Indirect Write	-	-	sHub® reset

5.3.1.6 RID 218h - Reset to factory setting

This resource allows all sHub® user-defined settings to be reset to the factory settings. Specifically, the following values are reset by this command:

- sHub® Input ports Parameters.
- Error Log.

Read access by a Long Message **direct transaction** reports the resource definition, whose qualifiers are listed hereafter; the offset value of the Long Message shall be used for selecting the qualifier to be read:

Table 32: RID 218h: sHub® Reset-to-Factory Definition

Definign qualifier	Offset	Value
RID		218h
Resource name	0	"SH_FACSE"
Data size	1	8
Read access level	2	15
Write access level	3	2
Time overrun	4	255
Data type	5	0Bh - ASCII string

Write access by a Long Message direct transaction shall not be granted. An attempt shall lead to an error indication.

Read and/or write access by a Long Message **indirect transaction** depends on the access rights. Any denied or wrong access shall lead to error indications as detailed later on in the resource access description table.

The offset value of the Long Message is just relevant for the Long Message direct transaction.

14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
r	r	r	r	r	r	r	r	i	i	i	i	i	i	i
MSB														LSB

Figure 15: RID 218h: Offset Value Structure (Direct Read Access)

Table 33: RID 218h: Offset Value Definition (Direct Read Access)

Signal	Value	Definition
r	-	Reserved
i	0...5	Return the qualifiers of the RID definition

The sHub® Reset-to-Factory is triggered by a Long Message **indirect write transaction with specific data pattern (code word)**, as described hereafter:

0	1	2	3	4	5	6	7	DataByte
D1	D2	D3	D4	D5	D6	D7	D8	

Figure 16: RID 218h: Data structure according to Data type (Indirect Write Access)

Table 34: RID 218h: Value Definition (Indirect Write Access)

Signal	Value	Definition
D1...D8	"RESE-TALL"	Code word to reset the sHub® user-defined settings to factory settings

Table 35: RID 218h: sHub® Reset-to-Factory - Access Detailed Description

Offset	Length	Mode	DataBytes	Response	Description
0000h	8	Direct Read	-	“SH_FACSE”	Resource name
0001h	2	Direct Read	-	8	Data size
0002h	2	Direct Read	-	15	Read access level
0003h	2	Direct Read	-	2	Write access level
0004h	2	Direct Read	-	xxx	Time-out
0005h	2	Direct Read	-	11	Data type (0Bh) -ASCII string.
0006h... 7FFFh	0 ... 8	Direct Read	-	Error: 13 40	Not allowed
0000h... 7FFFh	0 ... 8	Direct Write	xx ...	Error: 14 41	Not allowed
0000h... 7FFFh	0 ... 8	Indirect Read	-	Error: 11 41	Not allowed
0000h... 7FFFh	8	Indirect Write	52 45 53 45 54 41 4C 4C	Error: 12 41	Not allowed with access level < 2
0000h... 7FFFh	8	Indirect Write	52 45 53 45 54 41 4C 4C	-	User data reset (access level ≥ 2)
0000h... 7FFFh	8	Indirect Write	xx xx xx xx xx xx xx xx	Error: 15 40	Not allowed (data not equal to code-word)

5.3.1.7 RID 280h - Device Type

This resource describes the basic type of functionality provided by the sHub® device.

Read access by a Long Message **direct transaction** reports the resource definition, whose qualifiers are listed hereafter; the offset value of the Long Message shall be used for selecting the qualifier to be read:

Table 36: RID 280h: sHub® Device Type Definition

Defining qualifier	Offset	Value
RID		218h

Defining qualifier	Offset	Value
Resource name	0	"SH_TYPE"
Data size	1	2
Read access level	2	0
Write access level	3	15
Time overrun	4	180...255
Data type	5	04h - unsigned 16bit

Write access by a Long Message direct transaction shall not be granted. An attempt shall lead to an error indication.

Read and/or write access by a Long Message **indirect transaction** depends on the access rights. Any denied or wrong access shall lead to error indications as detailed later on in the resource access description table.

The offset value of the Long Message is just relevant for the Long Message direct transaction.

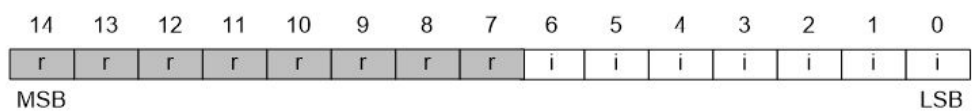


Figure 17: RID 280h: Offset Value Structure (Direct Read Access)

Table 37: RID 280h: Offset Value Definition (Direct Read Access)

Signal	Value	Definition
r	-	Reserved
i	0...5	Return the qualifiers of the RID definition

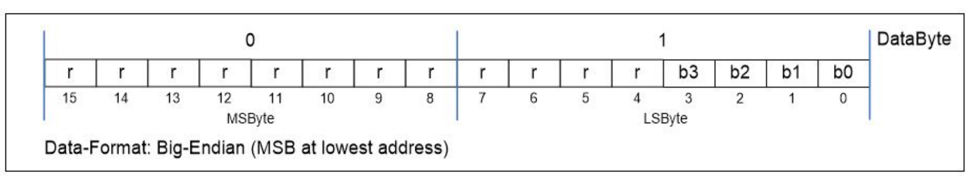


Figure 18: RID 280h: Data structure according to Data type (Indirect Read Access)

Table 38: RID 280h: Value Definition (Indirect Read Access)

Signal	Value	Definition
r	-	Reserved
b3...b0	0002h	Functionality type of sHub® device

Table 39: RID 280h: sHub® Device Type - Access Detailed Description

Offset	Length	Mode	Data Bytes	Response	Description
0000h	8	Direct Read	-	“SH_TYPE”	Resource name
0001h	2	Direct Read	-	2	Data size
0002h	2	Direct Read	-	0	Read access level
0003h	2	Direct Read	-	15	Write access level
0004h	2	Direct Read	-	xxx	Time-out

Offset	Length	Mode	Data Bytes	Response	Description
0005h	2	Direct Read	-	4	Data type (04h) – 16bit, unsigned
0006h…7FFFh	0…8	Direct Read	-	Error: 13 40	Not allowed
0000h…7FFFh	0…8	Direct Write	xx …	Error: 14 41	Not allowed
0000h	2	Indirect Read	-	00 02	Functionality type of sHub® device
0001h…7FFFh	2	Indirect Read	-	Error: 13 40	Not allowed
0000h…7FFFh	8	Indirect Write	xxxxh	Error: 10 41	Not allowed

5.3.1.8 RID 283h - Type Code Name

This resource indicates the type code name (i.e.: the coded name identifying a SICK product) specific of the sHub® device.

The type code name shall consist of 18 characters at most; being each 8bit-char ASCII-coded. The type code name shall be stored left justified; unused trailing bytes are filled by the code 00h.

Read access by a Long Message **direct transaction** reports the resource definition, whose qualifiers are listed hereafter; the offset value of the Long Message shall be used for selecting the qualifier to be read:

Table 40: RID 283h: sHub® Type Code Name Definition

Defining qualifier	Offset	Value
RID		283h
Resource name	0	"SH_TYPCO"
Data size	1	18
Read access level	2	0
Write access level	3	15
Time overrun	4	180…255
Data type	5	0Bh - ASCII string

Write access by a Long Message direct transaction shall not be granted. An attempt shall lead to an error indication.

Read and/or write access by a Long Message **indirect transaction** depends on the access rights. Any denied or wrong access shall lead to error indications as detailed later on in the resource access description table.

The offset value of the Long Message indirect transaction shall be used as an index for addressing the specific byte of the “Type Code Name” characters string.

It should be noted that three Long Message transactions are required for reading the whole string, 8 bytes of data at time.

14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
r	r	r	r	r	r	r	r	i	i	i	i	i	i	i
MSB														LSB

Figure 19: RID 283h: Offset Value Structure

Table 41: RID 283h: Offset Value Definition (Direct Read Access)

Signal	Value	Definition
r	-	Reserved
i	0 … 5	Return the qualifiers of the RID definiton

Table 42: RID 283h: Offset Value Definition (Indirect Read Access)

Signal	Value	Definition
r	-	Reserved
i	0…16	Index to the “Type Code Name” characters

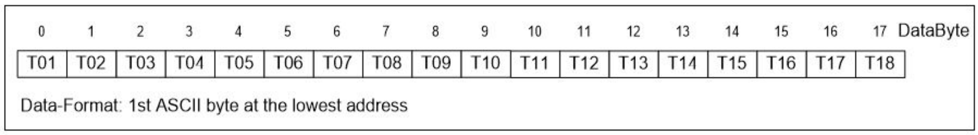


Figure 20: RID 283h: Data structure according to Data type (Indirect Read Access)

Table 43: RID 283h: Value Definition (Indirect Read Access)

Signal	Value	Definiton
T01…T18	00h…FFh	“Type Code Name” characters (ASCII codes)

Table 44: RID 283h: sHub® Device Type Code Name - Access Detailed Description

Offset	Length	Mode	Data Bytes	Response	Description
0000h	8	Direct Read	-	“SH_TYPCO”	Resource name
0001h	2	Direct Read	-	18	Data size
0002h	2	Direct Read	-	0	Read access level
0003h	2	Direct Read	-	15	Write access level
0004h	2	Direct Read	-	xxx	Time-out
0005h	2	Direct Read	-	11	Data type (0Bh) – string
0006h…7FFFh	0…8	Direct Read	-	Error: 13 40	Not allowed
0000h…7FFFh	0…8	Direct Write	xx …	Error: 14 41	Not allowed
0000h	8	Indirect Read	-	53 48 55 42 31 30 2D 30	“SHUB10-0”
0008h	8	Indirect Read	-	5A 41 38 37 30 31 30 00	“ZA87010”
0010h	2	Indirect Read	-	00 00	""
0012h…7FFFh	0…8	Indirect Read	-	Error: 13 40	Not allowed
0000h…7FFFh	8	Indirect Write	xx …	Error: 10 41	Not allowed

5.3.1.9 RID 284h - Serial Number

This resource indicates the serial number of each sHub® device manufactured.

The serial number shall consist of 10 characters at most; being each 8bit-char ASCII-coded. The type code name shall be stored left justified; unused trailing bytes are filled by the code 00h.

Read access by a Long Message direct transaction reports the resource definition, whose qualifiers are listed hereafter; the offset value of the Long Message shall be used for selecting the qualifier to be read:

Table 45: RID 284h: sHub® Serial Number Definition

Defining qualifier	Offset	Value
RID		284h
Resource name	0	"SH_SERNO"
Data size	1	10
Read access level	2	0
Write access level	3	15
Time overrun	4	180…255
Data type	5	0Bh – ASCII string

Write access by a Long Message direct transaction shall not be granted. An attempt shall lead to an error indication.

Read and/or write access by a Long Message indirect transaction depends on the access rights. Any denied or wrong access shall lead to error indications as detailed later on in the resource access description table.

The offset value of the Long Message indirect transaction shall be used as an index for addressing the specific byte of the “Serial Number” characters string.

It should be noted that two Long Message transactions are required for reading the whole string, 8 bytes of data at time.

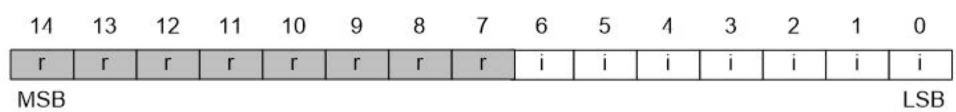


Figure 21: RID 284h: Offset Value Structure

Table 46: RID 284h: Offset Value Definition (Direct Read Access)

Signal	Value	Definition
r	-	Reserved
i	0…5	Return the qualifiers of the RID definition

Table 47: RID 284h: Offset Value Definition (Indirect Read Access)

Signal	Value	Definition
r	-	Reserved
i	0…8	Index to the “Serial Number” chararacters

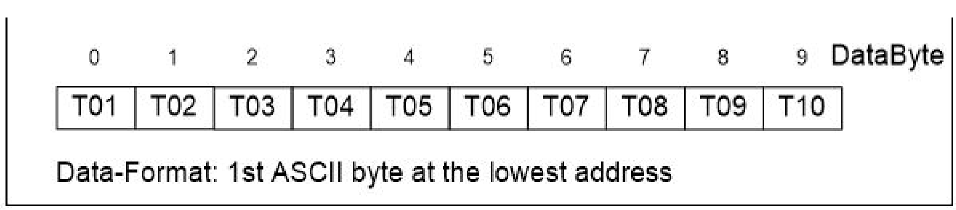


Figure 22: RID 284h: Data structure according to Data type (Indirect Read Access)

Table 48: RID 284h: Value Definition (Indirect Read Access)

Signal	Value	Definition
T01…T10	00h…FFh	“Serial Number” characters (ASCII codes)

Table 49: RID 284h: sHub® Serial Number - Access Detailed Description

Offset	Length	Mode	Data Bytes	Response	Description
0000h	8	Direct Read	-	“SH_SERNO”	Resource name
0001h	2	Direct Read	-	10	Data size
0002h	2	Direct Read	-	0	Read access level
0003h	2	Direct Read	-	15	Write access level
0004h	2	Direct Read	-	xxx	Time-out
0005h	2	Direct Read	-	11	Data type (0Bh) – string
0006h…7FFFh	0…8	Direct Read	-	Error: 13 40	Not allowed
0000h…7FFFh	0…8	Direct Write	xx …	Error: 14 41	Not allowed
0000h	8	Indirect Read	-	xx xx xx xx xx xx xx xx	
0008h	2	Indirect Read	-	xx xx	
000Ah…7FFFh	0…8	Indirect Read	-	Error: 13 40	Not allowed
0000h…7FFFh	8	Indirect Write	xx …	Error: 10 41	Not allowed

5.3.1.10 RID 285h - Firmware Revision Number

This resource indicates the firmware and hardware revision number of the sHub® device.

The firmware and hardware revision number shall consist of 20 characters at most; being each 8bit-char ASCII-coded. The firmware revision number shall be stored left justified; unused trailing bytes shall be filled by the code 20h.

The hardware revision shall be indicated by a 4-characters string starting with the identifier “HW_” followed by the revision character ‘A’ through ‘Z’. The hardware revision number shall occupy the last 4 bytes of the characters string, starting at offset 16.

Read access by a Long Message **direct transaction** reports the resource definition, whose qualifiers are listed hereafter; the offset value of the Long Message shall be used for selecting the qualifier to be read:

Table 50: RID 285h: sHub® Firmware Revision Number Definition

Defining qualifier	Offset	Value
RID		285h
Resource name	0	"SH_FWREV"
Data size	1	20
Read access level	2	0
Write access level	3	15
Time overrun	4	70…255
Data type	5	0Bh – ASCII string

Write access by a Long Message direct transaction shall not be granted. An attempt shall lead to an error indication.

Read and/or write access by a Long Message **indirect transaction** depends on the access rights. Any denied or wrong access shall lead to error indications as detailed later on in the resource access description table.

The offset value of the Long Message indirect transaction shall be used as an index for addressing the specific byte of the “Firmware Revision Number” characters string.

It should be noted that three Long Message transactions are required for reading the whole string, 8 bytes of data at time.

14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
r	r	r	r	r	r	r	r	i	i	i	i	i	i	i
MSB														LSB

Figure 23: RID 285h: Offset Value Structure

Table 51: RID 285h: Offset Value Definition (Direct Read Access)

Signal	Value	Definition
r	-	Reserved
i	0...5	Return the qualifiers of the RID definition

Table 52: RID 285h: Offset Value Definition (Indirect Read Access)

Signal	Value	Definition
r	-	Reserved
i	0...16	Index to the “Firmware Revision Number” chararacters

17	16	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	DataByte
T01	T02	T03	T04	T05	T06	T07	T08	T09	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	

Data-Format: 1st ASCII byte at the lowest address

Figure 24: RID 285h: Data structure according to Data type (Indirect Read Access)

Table 53: RID 285h: Value Definition (Indirect Read Access)

Signal	Value	Definiton
T01...T16	01h...FFh	“Firmware Revision Number” characters (ASCII codes)
T17...T20	01h...FFh	“Hardware Revision Number” characters (ASCII codes)

Table 54: RID 285h: sHub® Firmware Revision Number - Access Detailed Description

Offset	Length	Mode	Data Bytes	Response	Description
0000h	8	Direct Read	-	“SH_FWREV”	Resource name
0001h	2	Direct Read	-	20	Data size
0002h	2	Direct Read	-	0	Read access level
0003h	2	Direct Read	-	15	Write access level
0004h	2	Direct Read	-	xxx	Time-out

Offset	Length	Mode	Data Bytes	Response	Description
0005h	2	Direct Read	-	11	Data type (0Bh) – string
0006h…7FFFh	0…8	Direct Read	-	Error: 13 40	Not allowed
0000h…7FFFh	0…8	Direct Write	xx …	Error: 14 41	Not allowed
0000h	8	Indirect Read	-	xx xx xx xx xx xx xx xx	
0008h	2	Indirect Read	-	xx xx	
0010h	4	Indirect Read	-	48 57 5F xx	HW_x (e.g.: “HW_A”)
0014Ah… 7FFFh	0…8	Indirect Read	-	Error: 13 40	Not allowed
0000h…7FFFh	8	Indirect Write	xx …	Error: 10 41	Not allowed

5.3.1.11 RID 286h - Firmware Date

This resource indicates the firmware date of the sHub® device.

The firmware date shall consist of 8 characters string, formatted as “$D_2D_1.M_2M_1.Y_2Y_1$”; being each 8bit-char ASCII-coded.

Read access by a Long Message **direct transaction** reports the resource definition, whose qualifiers are listed hereafter; the offset value of the Long Message shall be used for selecting the qualifier to be read:

Table 55: RID 286h: sHub® Firmware Date Definition

Defining qualifier	Offset	Value
RID		286h
Resource name	0	"SH_FWDAT"
Data size	1	8
Read access level	2	0
Write access level	3	15
Time overrun	4	70…255
Data type	5	0Bh – ASCII string

Write access by a Long Message direct transaction shall not be granted. An attempt shall lead to an error indication.

Read and/or write access by a Long Message **indirect transaction** depends on the access rights. Any denied or wrong access shall lead to error indications as detailed later on in the resource access description table.

The offset value of the Long Message is just relevant for the Long Message direct transaction.

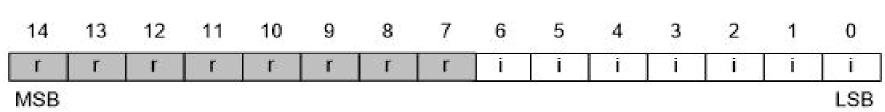


Figure 25: RID 286h: Offset Value Structure (Direct Read Access)

Table 56: RID 286h: Offset Value Definition (Direct Read Access)

Signal	Value	Definition
r	-	Reserved

Signal	Value	Definition
i	0…5	Return the qualifiers of the RID definition

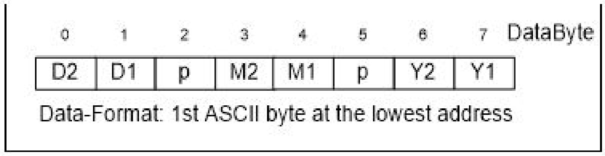


Figure 26: RID 286h: Data structure according to Data type (Indirect Read Access)

Table 57: RID 286h: Value Definition (Indirect Read Access)

Signal	Value	Definition
Y1	30h…39h	1st decimal-digit of the short-format of the Year (full-format “20y_2y_1”)
Y2	30h…39h	2nd decimal-digit of the short-format of the Year (full-format “20y_2y_1”)
p	2E	Decimal point as separator (“.”)
M1	30h…39h	1st decimal-digit of the Month (0…9)
M2	30h…3!h	2nd decimal-digit of the Month (0…1)
p	2E	Decimal point as separator (“.”)
D1	30h…39h	1st decimal-digit of the Day (0…9)
D2	30h…39h	2nd decimal-digit of the Day (0..3)

Table 58: RID 286h: sHub® Firmware Date - Access Detailed Description

Offset	Length	Mode	Data Bytes	Response	Description
0000h	8	Direct Read	-	“SH_FWDAT”	Resource name
0001h	2	Direct Read	-	10	Data size
0002h	2	Direct Read	-	0	Read access level
0003h	2	Direct Read	-	15	Write access level
0004h	2	Direct Read	-	xxx	Time-out
0005h	2	Direct Read	-	11	Data type (0Bh) – string
0006h…7FFFh	0…8	Direct Read	-	Error: 13 40	Not allowed
0000h…7FFFh	0…8	Direct Write	xx …	Error: 14 41	Not allowed
0000h	8	Indirect Read	-	xx xx 2E xx xx 2E xx xx	xx xx xx
0008h…7FFFh	0…8	Indirect Read	-	Error: 13 40	Not allowed
0000h…7FFFh	8	Indirect Write	xx …	Error: 10 41	Not allowed

5.3.1.12 RID 2C0h - Temperature Range

This resource indicates the minimum and maximum permitted values for the operating temperature of the sHub® device, as specified in the product data sheet.

The temperature range values are stored as an array of two signed 16 bit values in the form of 2's complements. The temperature value units are tenths of degrees Celsius (0.1 °C).

NOTE
It is for information only; no temperature measurement and monitoring is done inside the sHub® device.

Read access by a Long Message **direct transaction** reports the resource definition, whose qualifiers are listed hereafter; the offset value of the Long Message shall be used for selecting the qualifier to be read:

Table 59: RID 2C0h: sHub® Operating Temperature Range Definition

Defining qualifier	Offset	Value
RID		2C0h
Resource name	0	"SH_TEMPR"
Data size	1	4
Read access level	2	0
Write access level	3	15
Time overrun	4	180...255
Data type	5	08h – signed 16bit

Write access by a Long Message direct transaction shall not be granted. An attempt shall lead to an error indication.

Read and/or write access by a Long Message **indirect transaction** depends on the access rights. Any denied or wrong access shall lead to error indications as detailed later on in the resource access description table.

The offset value of the Long Message is just relevant for the Long Message direct transaction.

14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
r	r	r	r	r	r	r	r	i	i	i	i	i	i	i
MSB														LSB

Figure 27: RID 2C0h: Offset Value Structure (Direct Read Access)

Table 60: RID 2C0h: Offset Value Definition (Direct Read Access)

Signal	Value	Definition
r	-	Reserved
i	0...5	Return the qualifiers of the RID definition

Table 61: RID 2C0h: Offset Value Definition (Indirect Read Access)

Signal	Value	Definition
r	-	Reserved
i	0	Return both min / max temperatures (data length = 4) Return the min. temperature only (data length = 2)
	2	Return the max. temperature only (data length = 2)

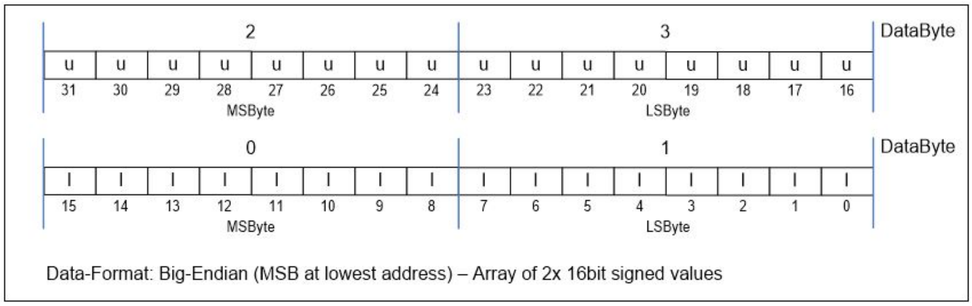


Figure 28: RID 2C0h: Data structure according to Data type (Indirect Read Access)

Table 62: RID 2C0h: Value Definition (Indirect Read Access)

Signal	Value	Definition
u	-2730...+10000	Upper operating temperature value (tenths of °C; 2’s complement)
l	-2730...+10000	Lower operating temperature value (tenths of °C; 2’s complement)

Table 63: RID 2C0h: Operating Temperature Range Example

Values	Data Bytes [0...3]
-40°C...+115 °C	FE 70 04 7E

Table 64: RID 2C0h: sHub® Operating Temperature Range - Access Detailed Description

Offset	Length	Mode	Data Bytes	Response	Description
0000h	8	Direct Read	-	“SH_TEMPR”	Resource name
0001h	2	Direct Read	-	4	Data size
0002h	2	Direct Read	-	0	Read access level
0003h	2	Direct Read	-	15	Write access level
0004h	2	Direct Read	-	xxx	Time-out
0005h	2	Direct Read	-	8	Data type (08h) – string
0006h...7FFFh	0...8	Direct Read	-	Error: 13 40	Not allowed
0000h...7FFFh	0...8	Direct Write	xx ...	Error: 14 41	Not allowed
0000h	4	Indirect Read	-	xx xx xx xx xx xx xx xx	Min. (-40°C) and max. (115°C) operating temperature
0000h	2	Indirect Read	-	xx xx	Min. operat-ing tempera-ture (-40°C)
0002h	2	Indirect Read	-	04 7E	Max.operat-ing tempera-ture (115°C)
0008h...7FFFh	0...8	Indirect Read	-	Error: 13 40	Not allowed
0000h...7FFFh	8	Indirect Write	xx ...	Error: 10 41	Not allowed

5.3.1.13 RID 2C4h - Supply Voltage Range

This resource indicates the minimum and maximum permitted values for the supply voltage of the sHub® device, as specified in the product data sheet.

The supply voltage range values are stored as an array of two unsigned 16 bit values. The supply voltage value units are 1 mV.

NOTE
It is for information only; no voltage measurement and monitoring is done inside the sHub® device.

Read access by a Long Message direct transaction reports the resource definition, whose qualifiers are listed hereafter; the offset value of the Long Message shall be used for selecting the qualifier to be read:

Table 65: RID 2C4h: sHub® Supply Voltage Range Definition

Defining qualifier	Offset	Value
RID		2C4h
Resource name	0	"SH_SUPR"
Data size	1	4
Read access level	2	0
Write access level	3	15
Time overrun	4	180…255
Data type	5	04h – unsigned 16bit

Write access by a Long Message direct transaction shall not be granted. An attempt shall lead to an error indication.

Read and/or write access by a Long Message indirect transaction depends on the access rights. Any denied or wrong access shall lead to error indications as detailed later on in the resource access description table.

The offset value of the Long Message is just relevant for the Long Message direct transaction.

14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
r	r	r	r	r	r	r	r	i	i	i	i	i	i	i
MSB														LSB

Figure 29: RID 2C4h: Offset Value Structure (Direct Read Access)

Table 66: RID 2C4h: Offset Value Definition (Direct Read Access)

Signal	Value	Definition
r	-	Reserved
i	0…5	Return the qualifiers of the RID definition

Table 67: RID 2C4h: Offset Value Definition (Indirect Read Access)

Signal	Value	Definition
r	-	Reserved
i	0	Return both min / max supply voltage (data length = 4) Return the min. supply voltage only (data length = 2)
	2	Return the max. supply voltage only (data length = 2)

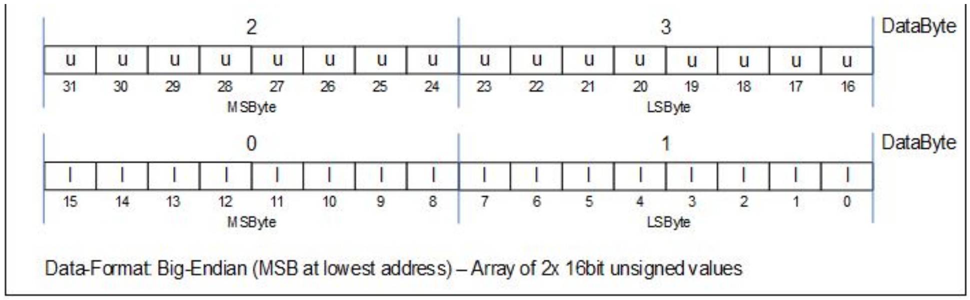


Figure 30: RID 2C4h: Data structure according to Data type (Indirect Read Access)

Table 68: RID 2C4h: Value Definition (Indirect Read Access)

Signal	Value	Definition
u	0...15000	Upper supply voltage value (mV)
l	0...15000	Lower supply voltage value (mV)

Table 69: RID 2C4h: Supply Voltage Range Example

Values	Data Bytes [0...3]
7 V...12 V	1B 58 2E E0

Table 70: RID 2C4h: sHub® Supply Voltage Range - Access Detailed Description

Offset	Length	Mode	Data Bytes	Response	Description
0000h	8	Direct Read	-	“SH_SUPR”	Resource name
0001h	2	Direct Read	-	4	Data size
0002h	2	Direct Read	-	0	Read access level
0003h	2	Direct Read	-	15	Write access level
0004h	2	Direct Read	-	xxx	Time-out
0005h	2	Direct Read	-	4	Data type (04h) – unsigned 16bit
0006h...7FFFh	0...8	Direct Read	-	Error: 13 40	Not allowed
0000h...7FFFh	0...8	Direct Write	xx ...	Error: 14 41	Not allowed
0000h	8	Indirect Read	-	1B 58 2E E0	Min. (7V) and max. (12V) supply voltage
0000h	2	Indirect Read	-	1B 58	Min. supply voltage (7V)
0002h	2	Indirect Read	-	2E E0	Max. supply voltage (12V)
0008h...7FFFh	0...8	Indirect Read	-	Error: 13 40	Not allowed
0000h...7FFFh	8	Indirect Write	xx ...	Error: 10 41	Not allowed

5.3.1.14 RID 2CBh - Lifetime

This resource returns the operating time of the sHub® device.

The operating time value is stored as an unsigned 32 bit value. The operating time unit is 1 min.

Read access by a Long Message direct transaction reports the resource definition, whose qualifiers are listed hereafter; the offset value of the Long Message shall be used for selecting the qualifier to be read:

Table 71: RID 2CBh: sHub® Lifetime Definition

Defining qualifier	Offset	Value
RID		2CBh
Resource name	0	"SH_LIFET"
Data size	1	4
Read access level	2	0
Write access level	3	15
Time overrun	4	70…255
Data type	5	05h – unsigned 32bit

Write access by a Long Message direct transaction shall not be granted. An attempt shall lead to an error indication.

Read and/or write access by a Long Message indirect transaction depends on the access rights. Any denied or wrong access shall lead to error indications as detailed later on in the resource access description table.

The offset value of the Long Message is just relevant for the Long Message direct transaction.

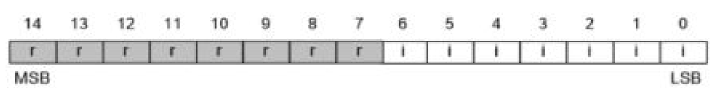


Figure 31: RID 2CBh: Offset Value Structure (Direct Read Access)

Table 72: RID 2CBh: Offset Value Definition (Direct Read Access)

Signal	Value	Definition
r	-	Reserved
i	0…5	Return the qualifiers of the RID definition

Table 73: RID 2CBh: Offset Value Definition (Indirect Read Access)

Signal	Value	Definition
r	-	Reserved
i	0	Return operating time (data length = 4, [min])

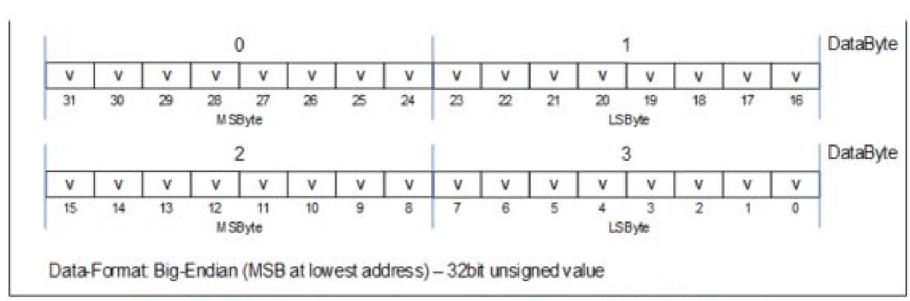


Figure 32: RID 2CBh: Data structure according to Data type (Indirect Read Access)

Table 74: RID 2CBh: Value Definition (Indirect Read Access)

Signal	Value	Definition
v	0…4,294,967,295	Operating time value (min)

Table 75: RID 2CBh: Lifetime Example

Values	Data Bytes [0…3]
15,000 min	00 00 3A 98

Table 76: RID 2CBh: sHub® Lifetime - Access Detailed Description

Offset	Length	Mode	Data Bytes	Response	Description
0000h	8	Direct Read	-	“SH_LIFET”	Resource name
0001h	2	Direct Read	-	4	Data size
0002h	2	Direct Read	-	0	Read access level
0003h	2	Direct Read	-	15	Write access level
0004h	2	Direct Read	-	xxx	Time-out
0005h	2	Direct Read	-	5	Data type (05h) – string
0006h…7FFFh	0…8	Direct Read	-	Error: 13 40	Not allowed
0000h…7FFFh	0…8	Direct Write	xx …	Error: 14 41	Not allowed
0000h	4	Indirect Read	-	00 00 3A 98	Operating time (e.g. 15,000 min)
0004h…7FFFh	0…8	Indirect Read	-	Error: 13 40	Not allowed
0000h…7FFFh	8	Indirect Write	xx …	Error: 10 41	Not allowed

5.3.1.15 RID 2CCh - Error Log

This resource returns the logged error-messages at the sHub® device.

At most 16 error-messages can be stored; once the memory is full, further error-messages are recorded by overwritting the oldest log-entries (circular-buffering).

For each error-message a time-stamp is logged; that is a tag based on the operating time of the sHub® device.

Read access by a Long Message **direct transaction** reports the resource definition, whose qualifiers are listed hereafter; the offset value of the Long Message shall be used for selecting the qualifier to be read:

Table 77: RID 2CCh: sHub® Error Log Definition

Defining qualifier	Offset	Value
RID		2CCh
Resource name	0	"SH_ERRLG"
Data size	1	256 (16 entries x 16 bytes)
Read access level	2	0
Write access level	3	15
Time overrun	4	220…255
Data type	5	20h – 16 byte structure

Write access by a Long Message direct transaction shall not be granted. An attempt shall lead to an error indication.

Read and/or write access by a Long Message **indirect transaction** depends on the access rights. Any denied or wrong access shall lead to error indications as detailed later on in the resource access description table.

The offset value of the Long Message indirect transaction shall be used as an index for addressing the specific error-message to be read out.

It should be noted that two Long Message transactions are required for reading each erro-message entry, 8 bytes of data at time.

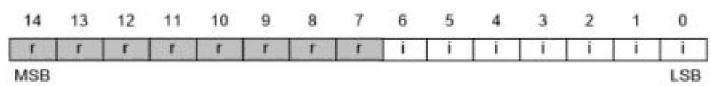


Figure 33: RID 2CCh: Offset Value Structure (Direct Read Access)

Table 78: RID 2CCh: Offset Value Definition (Direct Read Access)

Signal	Value	Definition
r	-	Reserved
i	0…5	Return the qualifiers of the RID definition

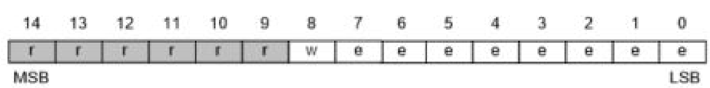


Figure 34: RID 2CCh: Offset Value Structure (Indirect Read Access)

Table 79: RID 2CCh: Offset Value Definition (Indirect Read Access)

Signal	Value	Definition
r	-	Reserved
w	1	MSByte of error log entry returned
	0	LSByte of error log entry returned
e	1…255	Index to address the specific log-entry
	0	Amount of error log entries

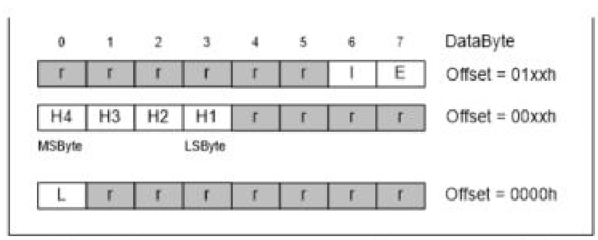


Figure 35: RID 2CCh: Data structure according to Data type (Indirect Read Access)

Table 80: RID 2CCh: Value Definition (Indirect Read Access)

Signal	Value	Definition
r	-	Reserved
L	00h…FFh	Amount of error log entries
H4/H3/H2/H1	00000000h…FFFFFFFFh	Operating time in minutes (Big-endian format)
I	00h…FFh	Additional Error code specification (refer to chapter 5.4)

Signal	Value	Definition
E	10h...17h 20h...27h 30h... 37h 40h...47h 50h...57h 60h...67h 70h...77h	Error code (refer to chapter 5.4)

Table 81: RID 2CCh: sHub® Error Log - Access Detailed Description

Offset	Length	Mode	Data Bytes	Response	Description
0000h	8	Direct Read	-	"SH_ERRLG"	Resource name
0001h	2	Direct Read	-	256	Data size
0002h	2	Direct Read	-	0	Read access level
0003h	2	Direct Read	-	15	Write access level
0004h	2	Direct Read	-	xxx	Time-out
0005h	2	Direct Read	-	32	Data type (0Bh) – string
0006h...7FFFh	0...8	Direct Read	-	Error: 13 40	Not allowed
0000h...7FFFh	0...8	Direct Write	xx ...	Error: 14 41	Not allowed
0000h	2	Indirect Read	-	xx 00	Amount of error log entries
0001h	8	Indirect Read		xx xx xx xx xx xx xx xx	1st part of the most recently logged error
0101h	8	Indirect Read		xx xx xx xx xx xx xx xx	2nd part of the most recently logged error
000xh	8	Indirect Read		xx xx xx xx xx xx xx xx	1st part of the xth logged error
010xh	8	Indirect Read		xx xx xx xx xx xx xx xx	2nd part of the xth logged error
0008h...7FFFh	0...8	Indirect Read	-	Error: 13 40	Not allowed
0000h...7FFFh	8	Indirect Write	xx ...	Error: 10 41	Not allowed

5.3.2 MEMS Accelerometer Acquisition

The sHub®-Core is specifically intended for acquiring and providing the servo-drive application with the MEMS accelerometer data along with the position-data via Hiperface DSL® link.

The sHub®- Core provides the user with an integrated acceleration sensor.

The Hiperface DSL® communication protocol allows the servo-drive application for achieving a synchronous acquisition of the accelerometer-sensor data along with the measured shaft-position and the motor-feedback relevant status. Specifically, the sHub®- Core acquires and processes the MEMS accelerometer output signal according to the HDSL basic framing:

- The sensor's output signal is sampled and processed on every H-Frame.
- The processing output data (signed 14bit, 2's complement) are transmitted over HDSL link (sHub® 8bit data slot) in two H-Frames; 7bit at time.
- The processing output data can also be transmitted over HDSL link (Long Messages related data slot) upon a specific request.

The servo-drive application can get the MEMS accelerometer data through the HDSL Master Interface, which allows for fulfilling a true real-time acquisition of those data (i.e.: every processing output data can be available for the servo-drive application purpose).

NOTE
The MEMS accelerometer data are buffered (FIFO) at the HDSL Master Interface; the buffer capacity is limited to 64bit. Empty or overrun status is notified to the application.

The servo-drive application can rely on the following diverse reading-out modes, for retrieving the MEMS accelerometer data:

- SPI-Pipeline based (the fastest access to the sHub® buffered data).
- Register-based (for accessing the sHub® buffered data via the Drive interface).
- Request-based (the slowest access to the most-recently available sHub® data).

The first mode and the second mode are suitable for a high-bandwidth data acquisition, synchronized with the HDSL framing; note that they are mutually-exclusive options. The third mode is just suitable for a low-bandwidth, asynchronous, data acquisition; this mode, relying on the Long Message protocol, is just for on-request basis data acquisition.

NOTE
The MEMS accelerometer data shall be acquired strictly synchronously with the shaft-position measurement provided by the HDSL motor-feedback, for making the data post-processing effective. Therefore both transactions for reading the sHub® data and the position data shall be triggered by the rising-edge of the FAST_POS_RDY signal provided by the HDSL Master Interface; further details are documented in the Hiperface DSL® Master Implementation manual (8017595).

Table 82: MEMS Accelerometer Acquisition

Scaling factor	1 LSB = 12.51 m g
Range	[E000h...1FFFh] = [-8192...+8191]

5.3.2.1 MEMS Accelerometer Data via SPI-PIPE

The MEMS accelerometer output signal is sampled and processed on every H-Frame. The processing output data (signed 14bit) are transmitted over HDSL link (sHub® 8bit data slot) in two chunks (7bit at time) over two consecutive H-Frames.

The HDSL Master Interface allows for forwarding those data along with a leading status (8bit) to the servo-drive application (data consumer) via the sHub® related interface (SPI-PIPE interface); as documented by the Hiperface DSL® Master Implementation manual (8017595).

NOTE
The MEMS accelerometer data are forwarded via the SPI-PIPE interface on condition that the **SPPE** bit of **SYS_CTRL** register at HDSL Master Interface is **set**.

SPI-PIPE Interface Timings

The SPI PIPE is a **read-only** Serial Peripheral Interface (SPI); whose master is the application at servo-drive, consuming the MEMS accelerometer data.

According to the original Motorola SPI specification, the following basic operating modes apply to the SPI-PIPE:

- Data changing on the SPI-clock rising-edge; data sampling on the SPI-clock falling-edge (PHA = 1).
- Basic SPI-clock level is low (POL = 0).
- Data transmitted starting from the most significant bit.

The following timing diagram applies:

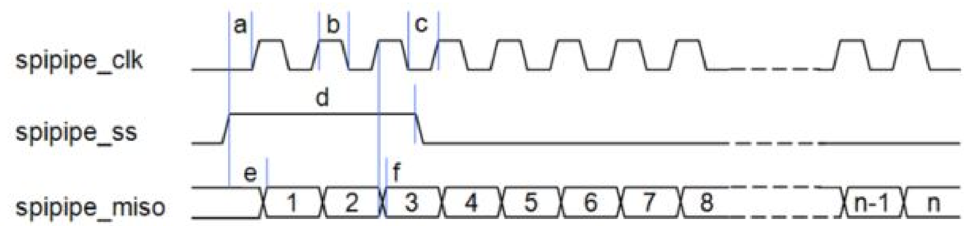


Figure 36: SPI PIPE Timing diagram

Table 83: SPI PIPE Timings

Timing reference	Description	Min	Max	Unit
a	Assertion of spipipe_ss before spipipe_clk	30		ns
b	spipipe_clk high pulse-width	30		ns
c	spipipe_clk low pulse-width	30		ns
d	spipipe_ss pulse width	30		ns
e	Delay spipipe_miso after spipipe_ss high	25	70	ns
f	Delay spipipe_miso after spipipe_clk high	25	70	ns

MEMS accelerometer data readout via SPI-PIPE

The MEMS accelerometer data can be read via the SPI-PIPE according to the following transaction, to be triggered on every H-Frame:

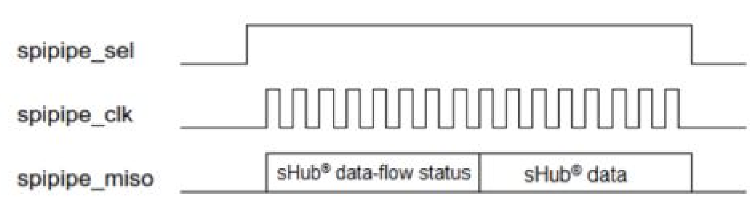


Figure 37: sHub® data along status readout via SPI-PIPE.

Note that the "spipipe_sel" is shown as extending over full clock-pulses train; according to the usual operating conditon for the standard SPI; although, specifically for the SPI-PIPE, the "spipipe_sel" pulse width could be shortened (see the SPI-PIPE timings above).

Two subsequent H-Frames are required for transferring the whole MEMS acceleration data (signed 14bit); consequently two subsequent SPI-PIPE transactions are required for retrieving those data 8bit at time.

The description at chapter 5.3.2.2 applies; specifically regarding the way of retrieving the MEMS accelerometer data from the sHub® data buffer.

5.3.2.2 MEMS Accelerometer Data via Register

The MEMS accelerometer output signal is sampled and processed on every H-Frame. The processing output data (signed 14bit) are transmitted over HDSL link (sHub® 8bit data slot) in two chunks (7bit at time) over two consecutive H-Frames.

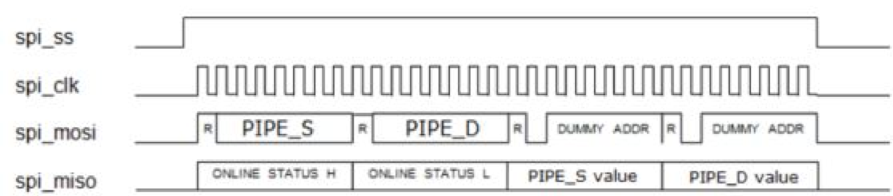
The HDSL Master Interface allows for forwarding those data data along with a leading status (8bit) to the servo-drive application (data consumer) via the sHub® related registers (**PIPE_S** and **PIPE_D**); which are accessible thru the Drive interface, by addressing them at **2Dh** and **2Eh** respectively; as documented by the Hiperface DSL® Master Implementation manual (8017595).

NOTE
The MEMS accelerometer data are forwarded via the Drive interface on condition that the **SPPE** bit of **SYS_CTRL** register at HDSL Master Interface is **reset.**

The MEMS accelerometer data can be retrieved by means of two subsequent register-read transaction; by reading on each transaction both PIPE_S and PIPE_D registers.

The MEMS accelerometer data (signed 14bit) are reported splitted in the PIPE_D; note that the most-significant part (MS-7bit) is reported in the PIPE_D value (8bit) showing the bit7 set („1“); for distinguishing it from the least-significant part (LS-7bit) reported in the PIPE_D value showing the bit7 reset („0“).

For reading both PIPE_S and PIPE_D registers the „Multi Register Read“ is recommended:



NOTE
The PIPE_S value shall be checked to validate the associated PIPE_D.

NOTE
The most-significant bit of PIPE_D shall be checked to detect the most-significant part of the MEMS accelerometer data

---

The whole MEMS accelerometer data are retrieved by combining two subsequend PIPE_D values:

Table 84: MEMS Accelerometer Data from PIPE_D register

MEMS Accelerometer data														
1st PIPE_D value							2nd PIPE_D value							
1	A13	A12	A11	A9	A8	A7	0	A6	A5	A4	A3	A2	A1	A0

PIPE_S and PIPE_D registers

The HDSL Master Interface reports the status of the sHub® relevant data-flow into the register PIPE_S, which is accessible via the Drive interface.

NOTE
The same status information is reported via SPI-PIPE interface as the first byte of each read transaction (refer to chapter 5.3.2.1).

Upon the PIPE_S register reading access, the current data from the sHub® relevant data FIFO buffer is read and prepared for the subsequent read access to the PIPE_D register, in order to report the status coherently referring to the associated data value.

Table 85: PIPE_S register

PIPE_S	0	0	0	0	POVR	PEMP	PERR	PSCI
	bit7							bit0

POVR	sHub® data FIFO full	8-bytes buffer exhausted
PEMP	sHub® data FIFO empty	No data buffered
PERR	sHub® data corruption	Invalid data (bit-encoding error)
PSCI	sHub® “special” data	Special character (8b10b coding) buffered

Table 86: PIPE_D register

PIPE_D	sHub® relevant data from the buffer (00h reported if empty)	
	bit7	bit0

5.3.2.3 MEMS Accelerometer Data via Long Message

The MEMS accelerometer data can also be retrieved by a specific request, exploiting the Resource access protocol, Long Message based.

NOTE
This method of reading the acceleration data is not suitable for productive use of the acceleration data. It is only meant for development and validation purposes. For productive use of acceleration values, use one of the two other methods described directly above this subchapter.

The MEMS accelerometer data can be read out by means of the RID 200h; refer to the chapter 5.3.2.1 for further details.

5.3.3 Motor-winding Temperature Acquisition

The sHub®- Core provides user with an input port for connecting a motor-winding thermistor (KTY84-130, PT1000); which is functionally the same as what the HDSL Motor-feedback provides.

The motor-thermistor’s signal is acquired (converted from analog to digital) based on the HDSL link framing, but the subsequent digital data processing is not synchronous with that framing, being not actually required.

NOTE
The PT1000 specific resistance measurement data can optionally be adjusted to get the measurement data as the thermistor was instead the KTY84-130 type.

The motor-thermistor’s resistance measurement data are provided just on-request basis; that is, upon a HDSL Long Message read transaction addressing the RID 200h (ref. to chapter 5.3.1.2).

The motor-thermistor's resistance measurement data processing can also be conditioned by selecting some specific options by means of a HDSL Long Message write transaction addressing the RID 201h (ref. to chapter 5.3.1.3); the following independent options can be selected:

- Raw data or Filtered data
- KTY84-130 conversion

The thermistor's resistance measurement data are reported as an unsigned 32 bit integer value, whose scaling-factor and the range are hereafter specified:

Table 87: Thermistor's resistance measurement data

Scaling Factor	1 LSB = 1 Ω
Range	[00000000h…000332C0h] = [0…209600 Ω] FFFFFFFFh = Invalid measurement

5.4 sHub®-Core Functional Diagnostics

The sHub® internal diagnostics is implemented based on the same concept applied to the HDSL Motor-feedback diagnostics. A fault specific error-code is defined for reporting as many as possible details about the fault condition. Upon the fault detection, the specific error-code is stored in the error-log and the sHub® user shall access the error-log for retrieving the error-codes for diagnostic purpose.

How to access the error-log is described at chapter 5.3.1.13.

5.4.1 Error Code Groups

The error codes are grouped according to the specific category, as defined for the HDSL Motor-feedback. Eight error groups (EG0…EG7) are specified, but just few apply to the sHub® internal diagnostics, as listed in the table hereafter:

Table 88: Error Code Groups

EG #	Error Category
0	MFB reserved
1	MFB reserved
2	**Initialization**
3	MFB reserved
4	**Resource Access**
5	sHub® reserved
6	Reserved
7	MFB reserved

Error Codes Assignment

Each detected fault information is reported by a unique error code value (8bit), whose format is hereafter specified:

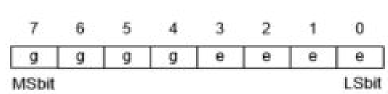


Figure 38: sHub® Error Code Value Format.

Table 89: sHub® Error Code Value Definition

Signal	Value	Definition
g	0…7	Error Code Group number (EG #)

Signal	Value	Definition
e	0...7	Error Code number

Table 90: sHub® Error Codes Assignment

EG #	Error category	Error Code number	Error definition
0			
1			
2	Initialization	3	Standard parameter error
		5	Communication error (I^2C)
3			
4	Resource access	0	Invalid argument
		1	Permission error (RID access denied)
		2	Internal error
6			
7			

5.4.2 Resource Access Errors

A specific error code is notified whenever the sHub® user performs a Long Message transaction for accessing the Resources database (ref. chapter 5.3.1) and a fault is detected.

The fault detection is notified directly in the specific Long Message response by means of:

“ERR” qualifier (ref. to the register PC_ADD at the HDSL Master Interface). “Resource Access Error” (RAE) code in the response data-field (ref. to registers PC_BUFFER0 and PC_BUFFER1 at the HDSL Master Interface).

The sHub® applicable RAE codes are listed in the following table:

Table 91: sHub® Resources Access Error Codes Assignment

EG #	Error Code	PC_BUFFER 1	PC_BUFFER 0	Meaning of the error code
4	0	40h	10h	Resource idendifier (RID) not implemented
			11h	Incorrect length for resource access given
			13h	Offset address too high
			14h	Invalid offset address
			15h	Invalid data value
	1	41h	10h	Write access not possible
			11h	Read access not possible
			12h	Write access denied
			13h	Read access denied
			14h	Write access for direct resource access denied
	2	42h	10h	Resource database entry damaged
			12h	Internal processing error during resource access

5.4.3 Troubleshooting

The sHub®-Core user shall handle the notified fault conditions as specified in the following table:

Table 92: sHub® Troubleshooting

EG #	Error Code	Notification	Severity	Add. Error Code	Description	Error handling
2	5	EEPROM access: Internal Communication Error 2	Minor / Critical	97	I^2C communication error on accessing RID 218h.	The error might be temporary. If the error persists, after a SW-reset (RID 210h), a hardware failure has most-likely occurred. A device replacement is required.
				96		
				95		
				94		
				93	I^2C communication error on accessing RID 2CCh.	
				92		
				91		
				90	I^2C communication error on startup (power-on or reset).	
				0Bh		
				0Ah		
	3	EEPROM access: Standard Parameter Error		82h	Wrong CRC related to the error-log storage memory (while writing).	A reset to factory setting (RID 218h) might fix. If the error persists, after a SW-reset (RID 210h), a hardware failure has most-likely occurred. A device replacement is required.
				81h		
				1Bh	Wrong CRC related to the parameters and error-log storage memory (while accessing it on power-on or reset).	
				19h		
				17h		
				15h		
				13h		
				11h		
				0Bh		
				09h		
				0Ah	Internal error on reading data related to the RID 201h.	
				08h	Wrong data related to the parameters storage memory (while accessing it on power-on or reset).	
4	2	RID access error	Minor	8Eh	Internal error when accessing a resource.	Check first the command format and the access arguments; then retry.
			Critical	51h	Resources database entry corruption.	If the error persists, after a SW-reset (RID 210h), a hardware failure has most-likely occurred. A device replacement is required.

EG #	Error Code	Notifi-cation	Severity	Add. Error Code	Description	Error handling
4	1	RID access denied	Minor	8Fh	Write access to RID 200h not allowed.	Check the access cre-dentials and retry.
				85h	Write access to RID 2CCh not allowed.	
				57h	Indirect write access denied	
				56h	Indirect write access invalid	
				55h	Indirect read access denied	
				54h	Indirect read access invalid	
				53h	Direct read access denied	
				52h	Direct read access invalid	
				51h	Direct write access not allowed	
4	0	RID access invalid argu-ment	Minor	D1h	RID 201h invalid setup (KTY).	Check first the access arguments; then retry.
				CDh	RID 202h invalid off-set value.	
				C2h	RID 201h invalid setup (Accel.)	
				C1h	RID 201h invalid setup (Therm.).	
				C0h	RID 201h invalid off-set value.	
				BEh	RID 200h invalid off-set value.	
				9Fh	RID 218h invalid data value.	
				85h	RID 2CCh invalid off-set value.	
				59h	Invalid offset value for Node.	
				58h	Offset too high (indi-rect access).	
				57h	Size mismatch (indi-rect access).	
				56h	Invalid RID (indirect access).	
				55h	Offset too high (direct access).	
				54h	Invalid RID (direct access).	
				53h	Missing RID.	

8026039//2020-09_30

Australia
Phone +61 (3) 9457 0600
1800 33 48 02 – tollfree
E-Mail sales@sick.com.au

Austria
Phone +43 (0) 2236 62288-0
E-Mail office@sick.at

Belgium/Luxembourg
Phone +32 (0) 2 466 55 66
E-Mail info@sick.be

Brazil
Phone +55 11 3215-4900
E-Mail comercial@sick.com.br

Canada
Phone +1 905.771.1444
E-Mail cs.canada@sick.com

Czech Republic
Phone +420 234 719 500
E-Mail sick@sick.cz

Chile
Phone +56 (2) 2274 7430
E-Mail chile@sick.com

China
Phone +86 20 2882 3600
E-Mail info.china@sick.net.cn

Denmark
Phone +45 45 82 64 00
E-Mail sick@sick.dk

Finland
Phone +358-9-25 15 800
E-Mail sick@sick.fi

France
Phone +33 1 64 62 35 00
E-Mail info@sick.fr

Germany
Phone +49 (0) 2 11 53 010
E-Mail info@sick.de

Greece
Phone +30 210 6825100
E-Mail office@sick.com.gr

Hong Kong
Phone +852 2153 6300
E-Mail ghk@sick.com.hk

Hungary
Phone +36 1 371 2680
E-Mail ertekesites@sick.hu

India
Phone +91-22-6119 8900
E-Mail info@sick-india.com

Israel
Phone +972 97110 11
E-Mail info@sick-sensors.com

Italy
Phone +39 02 27 43 41
E-Mail info@sick.it

Japan
Phone +81 3 5309 2112
E-Mail support@sick.jp

Malaysia
Phone +603-8080 7425
E-Mail enquiry.my@sick.com

Mexico
Phone +52 (472) 748 9451
E-Mail mexico@sick.com

Netherlands
Phone +31 (0) 30 229 25 44
E-Mail info@sick.nl

New Zealand
Phone +64 9 415 0459
0800 222 278 – tollfree
E-Mail sales@sick.co.nz

Norway
Phone +47 67 81 50 00
E-Mail sick@sick.no

Poland
Phone +48 22 539 41 00
E-Mail info@sick.pl

Romania
Phone +40 356-17 11 20
E-Mail office@sick.ro

Russia
Phone +7 495 283 09 90
E-Mail info@sick.ru

Singapore
Phone +65 6744 3732
E-Mail sales.gsg@sick.com

Slovakia
Phone +421 482 901 201
E-Mail mail@sick-sk.sk

Slovenia
Phone +386 591 78849
E-Mail office@sick.si

South Africa
Phone +27 10 060 0550
E-Mail info@sickautomation.co.za

South Korea
Phone +82 2 786 6321/4
E-Mail infokorea@sick.com

Spain
Phone +34 93 480 31 00
E-Mail info@sick.es

Sweden
Phone +46 10 110 10 00
E-Mail info@sick.se

Switzerland
Phone +41 41 619 29 39
E-Mail contact@sick.ch

Taiwan
Phone +886-2-2375-6288
E-Mail sales@sick.com.tw

Thailand
Phone +66 2 645 0009
E-Mail marcom.th@sick.com

Turkey
Phone +90 (216) 528 50 00
E-Mail info@sick.com.tr

United Arab Emirates
Phone +971 (0) 4 88 65 878
E-Mail contact@sick.ae

United Kingdom
Phone +44 (0)17278 31121
E-Mail info@sick.co.uk

USA
Phone +1 800.325.7425
E-Mail info@sick.com

Vietnam
Phone +65 6744 3732
E-Mail sales.gsg@sick.com

Detailed addresses and further locations at **www.sick.com**