TECHNICAL INFORMATION

HIPERFACE DSL®

Implementation

SICK
Sensor Intelligence.

HIPERFACE® DSL

This documentation applies to the HIPERFACE DSL® release version 1.07, release date July 29, 2016. Subject to modification without notice.

SICK STEGMANN GmbH accepts no responsibility for the non-infringement of patent rights, e.g. in the case of recommendations for circuit designs or processes.

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

Made in Germany, 2016.

Contents

1 List of figures

2 Scope of application of the document

This document is for a standard HIPERFACE DSL® application. For safety applications, please only refer to the document “HIPERFACE DSL® safety manual (8017596)".

2.1 Symbols used

NOTE
Notes refer to special features of the device. Please pay attention to these notes. They often contain important information.

Tips provide additional information that facilitates using the documentation.

CAUTION
Safety notes contain information about specific or potential dangers, and misuse of the application. This information is to prevent injury.

Read and follow the safety notes carefully.

2.2 Associated documents

Along with this manual, the following documents are relevant for the use of the HIPERFACE DSL® interface:

Document number	Title	Status
8017596	HIPERFACE DSL® safety manual	2018-01-15

Table 1: Associated documents

Individual encoder types with the HIPERFACE DSL® interface are described with the following documents:

- Data sheet
- Operating instructions
- Errata document

2.3 HIPERFACE DSL® for Motor Feedback Systems

This document describes the use and implementation of the HIPERFACE DSL® data protocol installed in motor feedback systems of servo drives.

HIPERFACE DSL® is a purely digital protocol that requires a minimum of connection cables between frequency inverter and motor feedback system. The robustness of the protocol enables the connection to the motor feedback system via the motor connection cable.

Motor feedback systems with the HIPERFACE DSL® interface can be used across all performance ranges and substantially simplify the installation of an encoder system in the drive:

- Standardized digital interface (RS485)
- Analog components for the encoder interface are not required
- Standardized interface between the frequency inverter application and the protocol logic

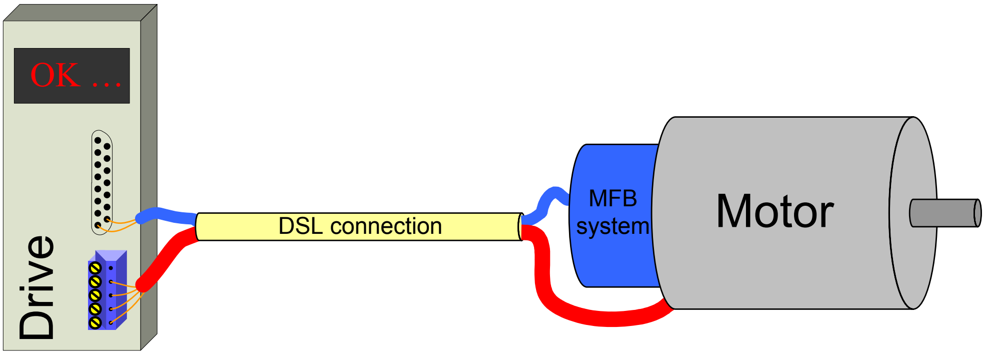


Figure 1: Drive system with HIPERFACE DSL®

Based on the name for the predecessor protocol, the SICK HIPERFACE®, the name HIPERFACE DSL® stands for HIgh PERformance InterFACE Digital Servo Link.

This interface takes into account all the current requirements of digital motor feedback systems and also contains future enhancements for the manufacturers of frequency inverters.

2.4 Features of HIPERFACE DSL®

Some of the main advantages of HIPERFACE DSL® are based on the opportunity for connection of the encoder:

- A digital interface on the frequency inverter for all communication with the motor feedback system. The interface complies with the RS485 standard with a transfer rate of 9.375 MBaud.
- Communication with the encoder via a twisted pair
- Power supply and communication with the encoder can be carried out using the same dual cable. This is possible by the enhancement of the frequency - inverter with a transformer.
- The connection cables to the encoder can be routed as a shielded, twisted- pair cable in the power supply cable to the motor. This means that no encoder plug connector to the motor and to the frequency inverter is necessary.
- The cable length between the frequency inverter and the motor feedback system can be up to 100 m, without degradation of the operating performance.

The digital HIPERFACE DSL® protocol can be used for a variety of frequency inverter applications:

- For the feedback cycle of the frequency inverter's synchronous cyclic data that enables synchronous processing of position and rotation speed of the encoder.
- Shortest possible cycle time: 12.2 µs.
- Transmission of the safe position of the motor feedback system with a maximum cycle time of 216 µs.
- Redundant transmission of the safe position of the motor feedback system with a maximum cycle time of 216 µs, so that suitable motor feedback systems can be used in SIL2 applications (in accordance with IEC 61508).
- Transmission of the safe position of the motor feedback system on a second channel with a maximum cycle time of 216 µs, so that suitable motor feedback systems can be used in SIL3 applications (in accordance with IEC 61508).

- Parameter data channel for bi-directional general data transfer with a band width of up to 340 kBaud. This data includes an electronic type label for designation of the motor feedback system and for storage of frequency inverter data in the motor feedback system.
- SensorHub channel via which motor data from external sensors is transmitted, that are connected by the HIPERFACE DSL® SensorHub protocol to the motor feedback system.

The protocol is integrated into the frequency inverter in the form of hardware logic. This logic circuit is supplied by several manufacturers as an IP Core for FPGA components (FPGA = Field Programmable Gate Array).

- The available protocol logic enables free routing when installing the HIPERFACE DSL® IP Core. The protocol circuit can be installed along with the frequency inverter application on the same FPGA.
- Choice between full-duplex SPI (SPI = serial peripheral interface) or parallel interface between protocol logic and frequency inverter applications for standardized access to process data (position, rotation speed) and parameters.
- Fast additional full-duplex SPI between protocol logic and frequency inverter applications for standardized access to secondary position data
- Additional configurable SPI for output of the data from external sensors.
- Configurable interrupt output

3 Protocol overview

HIPERFACE DSL® is a fast digital protocol for motor feedback systems for the connection between servo drive and motor feedback system. The protocol is installed in the transport layer in the frequency inverter using a digital logic circuit (DSL Master IP Core).

The position data are generated in two different ways in HIPERFACE DSL®, either in free running mode, in which the position values are sampled and transmitted as quickly as possible, or in SYNC mode, in which the position data are sampled and transmitted synchronously with a defined clock signal. With a frequency inverter application, this clock signal is normally the clock feedback of the frequency inverter.

In SYNC mode the protocol matches the time points for the sampling of the data without time fluctuations with the clock coming from the frequency inverter.

For each frequency inverter cycle at least one position value is sampled and transmitted with constant latency to the DSL Master. As the protocol matches the internal data transfer speed to the frequency inverter cycle, the overall transfer rate of the HIPERFACE DSL® depends on the frequency inverter clock.

The protocol package is matched to the various lengths, see figure 2. Provided the frequency inverter cycle is long enough, additional sampling points can be positioned in the frequency inverter cycle, known as "Extra" packages. The number of additional packages is programmed by the user with a divider value.

The number of packages transmitted per frequency inverter cycle cannot be selected at random, as the lower and upper range length of a protocol package must be adhered to. This must be taken into account when setting the divider value.

In free running mode, the frequency inverter cycle is not taken into account for sampling and transmission and the protocol uses the minimum package length.

It must be noted that the minimum package length in free running mode is shorter than the minimum package length in SYNC mode.

table 1 shows the dependency of the lengths of the protocol packages using examples for the length of the frequency inverter cycle.

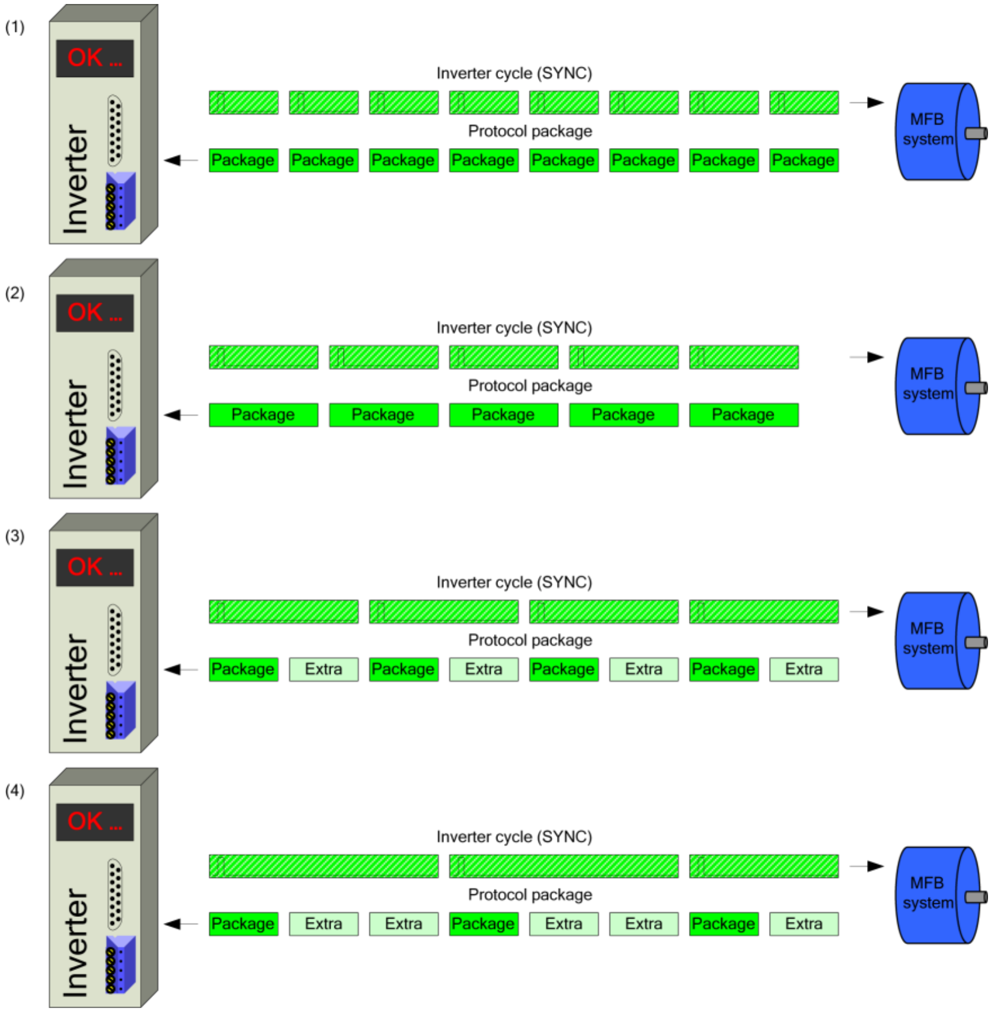


Figure 2: Length of protocol packages

Table 1: Frequency inverter cycle and length of protocol packages

Inverter cycle frequency (kHz)	Length of the frequency inverter cycle (µs)	Length of the protocol package (µs)	Protocol packages per frequency inverter cycle
2	500	12.50	40
4	250	12.50	20
6.25	160	13.33	12
8	125	12.50	10
16	62.5	12.50	5
40	25	12.50	2
37 to 84	27 to 12.2	27 to 12.2	1
Free running	--	11.52	--

In HIPERFACE DSL®, the data are transmitted over several channels. Each individual channel is adapted to different requirements according to its content. The cycle time of each individual channel varies with the length of the basic protocol package.

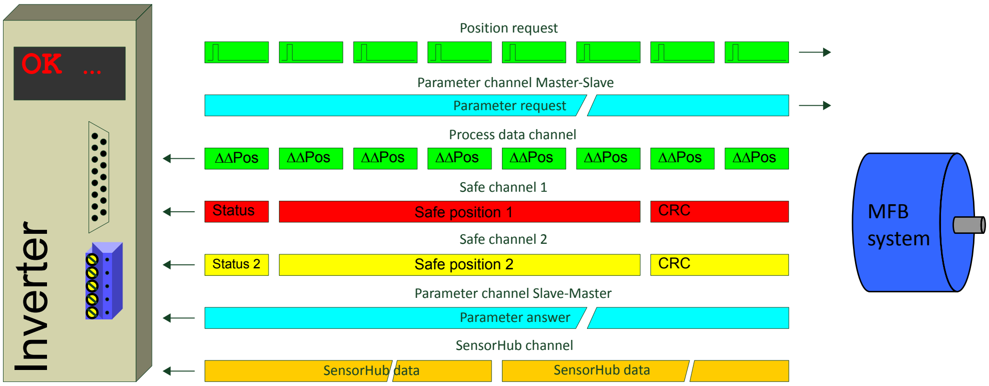


Figure 3: Data channels in HIPERFACE DSL®

table 2 gives an overview of the characteristics of the various channels.

NOTE
It should be noted that the minimum cycle time and the maximum band width only apply if the maximum number of sample points per frequency inverter cycle was programmed (refer to "Register synchronization control", chapter 6.3.2).

Table 2: Channels for protocol data

Channel in HIPERFACE DSL®	Function	Cycle time (µs)	Band width (kBaud)
Process data channel	Fast position, rotation speed	12.2 to 27.0	1321 to 669
Safe Channel 1	Absolute / safe position, status of Channel 1	96.8 to 216.0	660 to 334
Safe Channel 2	Absolute / safe position, status of Channel 2	96.8 to 216.0	660 to 334
Parameter channel	General data, parameters	Variable	330 to 167
SensorHub channel	External data	12.2 to 27.0	660 to 334

3.1 Process data channel

The fast position value of the motor feedback system is transferred on the process data channel synchronously with the position requests that are controlled by the signal at the SYNC input of the frequency inverter cycle.

The process data channel is the fastest channel of the HIPERFACE DSL® protocol. Every protocol package transferred contains a complete update of the content of this channel.

This content consists of increments to rotation speed that is used as feedback parameters for the control loop of the motor drive (see chapter 6.3.12 and chapter 6.3.13).

If the fast position from the process data channel cannot be calculated (either due to transmission or due to sensor errors), estimation is made by the DSL Master based on the last two available position values of Safe Channel 1. The worst case deviation from the actual mechanical position is also provided.

NOTE
For reliable position estimation, the user needs to provide application specific information about maximum speed and maximum acceleration. Please see chapter 7.3.1 for details.

3.1.1 Sampling Time

The fast position value sampling time is based on the transmission of a SYNC edge in a protocol package (where the SYNC edge can be user-commanded or belong to an EXTRA package, see above).

The duration from SYNC edge to sampling time point is based on the following formula:

$t_{sample} = t_{latency} + t_{delay} \pm t_{jitter}$

where

$t_{latency} < 100$ ns

$t_{delay} = 5$ ns/m * l_cable [m]

$t_{jitter} = 6.5$ ns + 13.33 ns * EDGES

EDGES refers to the number of set bits in the EDGES register, see chapter 6.3.7. Sampling latency will always be less than 1 µs. Note that position values will only be available after a longer duration (around 10 µs after SYNC edge) due to data serialization and transmission to the drive controller.

3.2 Safe Channel 1

The safe position value of the motor feedback system is transferred on the Safe Channel as an absolute value. In addition, the status of the encoder is reported on this channel in the form of errors and warnings.

NOTE
The safe position value transferred on the Safe Channel is not synchronous with the frequency inverter cycle signal at the SYNC input.

The safe position is used by the DSL Master IP Core to check the fast position value of the process data channel and can be used by the frequency inverter application for the same purpose.

Where there are deviations between the safe and the fast position values, an error message is generated (see chapter 5.4.2). In this case, the protocol replaces the fast position with the estimated position. Please see chapter 7.3.1 for details.

In each package of the safe channel, a collection of status bits is transferred that reflects the error and warning condition of the motor feedback system.

NOTE
It should be noted that each bit of the summary byte of the Safe Channel refers to one status byte the motor feedback system. Each status byte of the encoder can be read with a "short message" (see chapter 7.5.1).

3.2.1 Sampling Time

The safe position value (both channel 1 and 2) is not synchronous to the drive controller cycle signal at the SYNC input. The safe position value is transmitted in eight protocol packages. The sampling point of the safe position is based upon the SYNC edge of the first of these eight protocol packages (keeping in mind that the SYNC edge might be user-commanded or belong to a DSL Master-generated EXTRA package, see above). Depending on the actual position of the last user-generated SYNC edge the safe position value will be 1 to 9 protocol packages old. Depending on the timer settings for SYNC to EXTRA packages the sampling time of the safe position value will change between measurements.

3.3 Safe Channel 2

In Safe Channel 2, copies of the absolute position value and the status of the motor feedback system are transferred. This information can be discarded in non-safe applications.

NOTE
The Safe Channel 2 is only accessible in the safety variants of the DSL Master IP Core.

For sampling time of safe channel 2 see chapter 3.2.1.

3.4 Parameters Channel

The Parameters Channel is the interface, over which the frequency inverter application reads and writes parameters of the motor feedback system.

In addition to the main task of position measurement, motor feedback systems with the HIPERFACE DSL® interface also have various internal resources installed. These resources are accessible via the Parameters Channel.

Examples of these resources are temperature measurements, monitoring - mechanisms for correct functioning, product data (the "electronic type label") or freely programmable data fields.

NOTE
It should be noted that the resources actually installed for DSL products differ and are listed in the relevant product data sheet.

There are two types of communication on the Parameters Channel:

- "Short message" transaction
- "Long message" transaction

A "short message” transaction allows access to resources that have an influence on the HIPERFACE DSL® protocol interface and are used for monitoring them. This includes detailed status and error messages for the motor feedback system and indications of the signal strength on the DSL connection. As a "short message" transaction is processed directly by the interface logic of the motor feedback system, this transaction is completed in a comparatively short time.

A "long message" transaction allows access to all the other resources of the motor feedback system. Unlike a "short message" transaction, a "long message" normally requires processing by the motor feedback system processor and therefore has does not have a response time that can be defined in advance.

NOTE
It should be noted that in HIPERFACE DSL®, a maximum of one "short message" and one "long message" are processed at any time.

3.5 SensorHub Channel

Data from additional external sensors can be transferred on the SensorHub Channel that can be used in the frequency inverter system. External sensors must be connected to the motor feedback system via the HIPERFACE DSL® SensorHub interface. Various sensors or sensor networks are accessible via this interface and can be selected using HIPERFACE DSL®.

The configuration of external sensors is carried out via the Parameters Channel, whilst the data are transferred via the SensorHub Channel. The transfer of protocol packages in the SensorHub Channel takes place synchronously with the DSL transfer and as an extension of the frequency inverter cycle signal that is present at the DSL Master SYNC input. Depending on the use of the SensorHub interface, external data can therefore be sampled and transferred synchronously.

The protocol in the SensorHub Channel is not monitored by HIPERFACE DSL®. Apart from the monitoring of the data transfer quality, there are no protocol mechanisms on this channel.

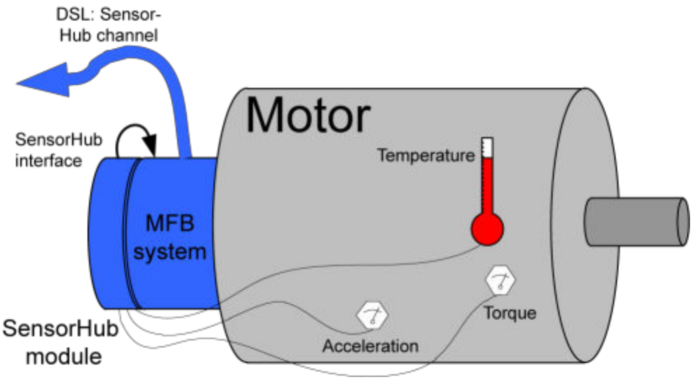


Figure 4: HIPERFACE DSL® SensorHub interface

4 Hardware installation

The installation of HIPERFACE DSL® in a drive system requires an interface circuit with specific components as well as the installation of a digital logic core for an FPGA component.

The interface circuit is described thoroughly in this chapter. The chapter also contains recommendations for the selection of components.

The digital logic core (IP Core) is supplied by SICK for prescribed FPGA types.

In addition, the type of cable recommended for the connection between the frequency inverter and the motor feedback system is described thoroughly in this chapter.

NOTE
It may also be possible to use other sorts of cable. These must be tested before use, however.

As a physical layer, HIPERFACE DSL® uses a transfer in accordance with EIA-485 (RS-485).

4.1 Interface circuit

In most cases a transceiver for more than 20 MBaud is suitable. Nevertheless the timing parameters of the transceiver have to fulfill the requirements of the following table 3 under worst case conditions of the application.

Table 3: Interface circuit

Characteristic	Value	Units
Transfer rate	>20	MBaud
Permitted common mode voltage	-7 to +12	V
Receiver: Differential threshold voltage	< 200	mV
Load resistance	< 55	Ohm
Receiver running time delay	< 60	ns
Sender running time delay	< 60	ns
Sender power-up delay	< 80	ns
Sender power-down delay	< 80	ns
Sender rise time	< 10	ns
Sender dropout time	< 10	ns
Switch over time of 1 bit	< 106.7	ns
Protection against short-circuit		
Protection against bus conflict		

HIPERFACE DSL® can be used in connection with two different interface circuit configurations. Each configuration requires a different sort of connection cable (see chapter 4.3).

4.1.1 Separate encoder cable - four core cable

When using a separate encoder cable, the smallest interface circuit can be used. The separate encoder cable allows a four core connection.

In connection with the associated table, figure 5 below gives the specification of the interface circuit.

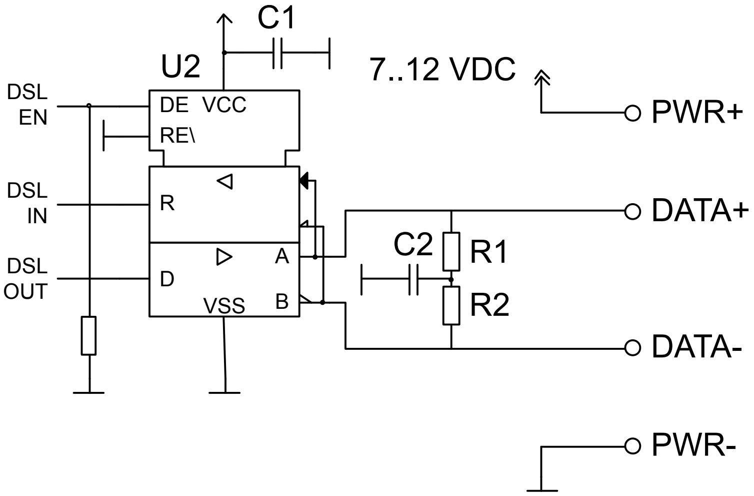


Figure 5: Interface circuit with separate encoder cable

Recommended components for the interface circuit are set out in table 4.

Table 4: Components for the interface circuit with separate encoder cable

Component		Part	Manufacturer
C1	Ceramic capacitor	100 nF	
C2	Ceramic capacitor	2.2 µF, 16 V	
R1, R2	Resistors	56R	
U2	RS485 transceiver	SN65LBC176A SN75LBC176A	Texas Instruments Texas Instruments

NOTE

The use of four core cable is no longer recommended for the motor cable.

4.1.2 Integrated cable - two core cable

For a connection via a two core cable integrated in the motor cable, (see chapter 4.3), the data cables must be provided with a transformer to raise the common mode rejection ratio. To feed the supply voltage into the data cables choke coils are also required.

In connection with the associated table, figure 6 below gives the specification of the interface circuit.

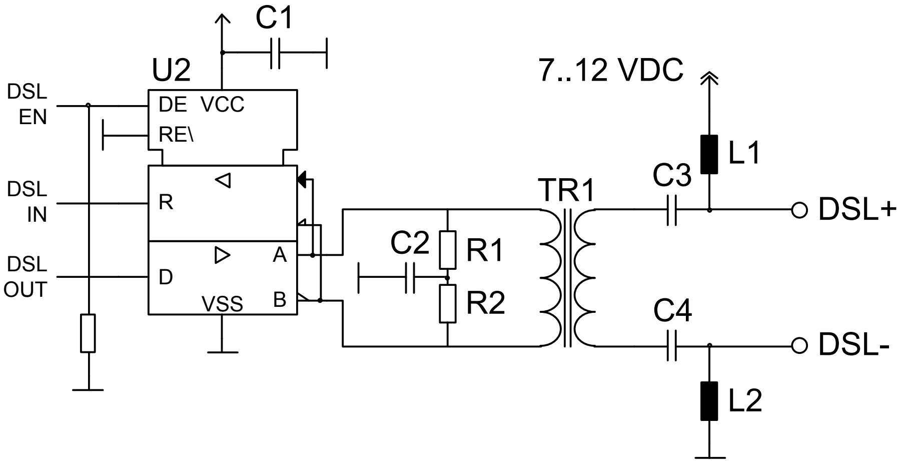


Figure 6: Interface circuit with two core cable (integrated in cable)

Recommended components for the interface circuit are set out in table 5.

Table 5: Components of the interface circuit with two core cable (integrated in cable)

Component		Part	Manufacturer
C1	Ceramic capacitor	100 nF	
C2	Ceramic capacitor	2.2 µF, 16 V	
C3, C4	Ceramic capacitor	470 nF, 50 V	
L1, L2	Choke coils	744043101, 100 µH ELL6SH101M, 100 µH	Würth Elektronik Panasonic
R1, R2	Resistors	56R	
U2	RS485 transceiver	SN65LBC176A SN75LBC176A	Texas Instruments Texas Instruments
TR1	Transformer	PE-68386NL 78602/1C B78304B1030A003 78602/1C	Pulse Engineering Murata Epcos Epcos

4.1.3 Motor feedback voltage supply

Motor feedback systems with HIPERFACE DSL® have been developed for operation with a supply voltage of 7 to 12 V. The voltage supply is measured at the encoder plug connector.

table 6 below describes the specification for the power supply.

Table 6: Voltage supply

Parameter	Value
Switch-on voltage ramp	Max. 180 ms from 0 to 7 V
Inrush current	Max. 3.5 A (0 to100 µs) Max. 1 A (100 µs to 400 µs)
Operating current	Max. 250 mA at 7 V

4.1.4 Interface circuit design recommendations

figure 5 and figure 6 show the two different interface circuits depending on the chosen system configuration. The following recommendations help in attaining a system design optimized for transmission robustness.

- During PCB design a good RF isolation for the interface circuit shall be achieved against the motor power circuit.
- The two sides of the transformer TR1 have to be well separated from each other to avoid crosstalk.
- Inside the servo controller the DSL-signal lines shall be routed as short as possible and with good symmetry in the differential part. To avoid or reduce signal disturbances by EMC-noise it is recommended to place this circuit as close as possible to the connection point of the DSL-lines.
- During PCB layout design also assess and avoid potential EMC-noise coupling from brake lines as well as the brake power supply circuit.
- For the encoder power supply via L1/L2 a star connection to a very low impedance point is important. Both inductances shall be well matched to each other to avoid differential mode noise. Self-resonance frequency should be of at least 10 MHz. A common mode filter between L1/L2 and the supply voltage can improve robustness.
- The DSL-line impedance is matched balanced by 2 x 56 Ohm. C2 grounds remaining common mode noise after the transformer; RF parts shall be used or different types paralleled to get low impedance on a broader frequency range. PCB design

at this area needs to consider RF requirements for the actual components selection and PCB layout.

- DSL signal transmission is done with about 10 MHz frequency but square signal harmonics can reach frequencies beyond (to 60 MHz) which should be considered for layout design.
- The used motor cable shall meet the impedance requirements of (110 +/-10) Ohm to avoid signal reflections.
- DSL line connection to the servo controller shall be separated from the motor power connection point.
- A good main shielding connection to a low inductance path shall allow draining motor power residual current. For the DSL-line shielding a separate connection point is recommended. For the connection unshielded DSL lines shall be avoided or kept as short as possible (<20 mm).

4.2 FPGA IP Core

The frequency inverter system communicates with the DSL motor feedback system via a special protocol logic circuit that is designated as the DSL Master. The circuit is supplied by SICK and must be installed in an FPGA component. It is supplied as an Intellectual Property Core (IP Core). The DSL Master IP Core is supplied in different forms, depending on the FPGA vendor preferred by the user (compiled netlist or encrypted VHDL). If there is sufficient space in the FPGA being used, the DSL Master can be installed in the same component as the frequency inverter application.

⚠ **CAUTION**
There are two different IP Cores available, one for standard and one for safety applications. This manual only describes the standard variant. Please choose according to the desired system.

For interfacing the IP Core, several options are available. For details of those interface blocks see chapter 9.1.

The following figure show the possible combinations of IP Core and interface block variants.

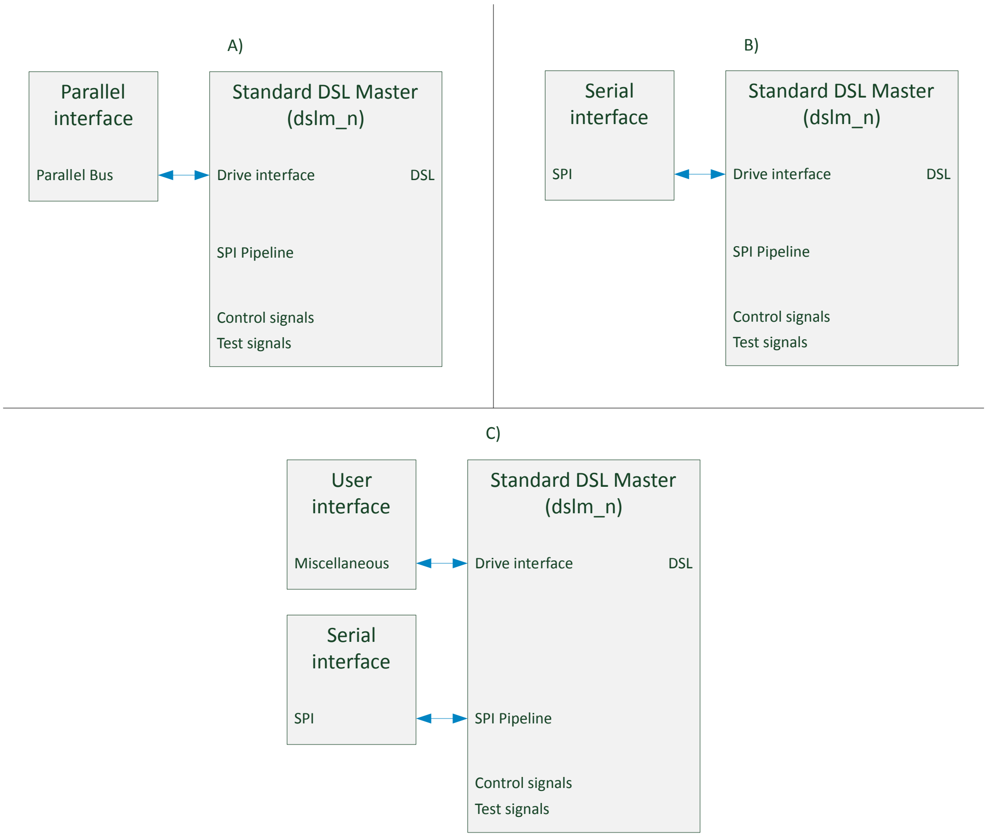


Figure 7: Block diagrams of the "standard" DSL Master IP Core with interfaces

4.2.1 DSL Master inputs / outputs

Table 7: Pin functions of the IP Core interface

Signal name	Type	Function
rst*	Input	Master reset (High active)
clk*	Input	Clock input
sync*	Input	Position sampling resolution
interrupt	Output	Configurable interrupt
link	Output	Connection indication
pos_ready	Output	Position data availability indication
sync_locked	Output	Position sampling resolution locked
bigend	Input	Byte sequence choice
fast_pos_rdy	Output	Fast position update indication
sample	Output	DSL bit sampling information
estimator_on	Output	Postion Estimator activated
safe_channel_err	Output	Transmission error in safe channel 1
safe_pos_err	Output	Safe position not valid
acceleration_err	Output	Fast channel / position error
acc_thr_err	Output	Fast channel / position threshold error
encoding_err	Output	DSL message encoding error
dev_thr_err	Output	Estimator deviation threshold reached
aux_signals	Output (12)	Auxiliary signals
dsl_in*	Input	DSL cable, input data

Signal name	Type	Function
dsl_out*	Output	DSL cable, output data
dsl_en*	Output	DSL cable transceiver, activation
spipipe_ss	Input	SensorHub SPI slave select
spipipe_clk	Input	Serial clock for SPI SensorHub
spipipe_miso	Output	SPI SensorHub, master output data/slave input data
online_status_d	Output (16)	IP Core status information
hostd_a	Input (7)	Host interface address
hostd_di	Input (8)	Host interface data in
hostd_do	Output (8)	Host interface data out
hostd_r	Input	Host interface data read
hostd_w	Input	Host interface data write
hostd_f	Input	Host interface register freeze

* these signals must be assigned to physical pins of the FPGA.

4.2.2 SYNC signal

The HIPERFACE DSL® communication can be established in “SYNC mode” or “free running mode”. In free running mode, the IP-Core will use the fastest possible transmission timing and this input should be low (0). Please note that the IP-Core is not bound to any timing of the frequency inverter in this mode.

In SYNC mode the frequency inverter clock must be supplied to this input/pin. Please refer to table 8 for the signal specification. This signal triggers position sampling of the DSL encoder. The polarity of the edge can be programmed using the SPOL bit in the SYS_CTRL register.

As the frame cycle time must always be within a limited range, a divider for the SYNC frequency has to be chosen accordingly. The divider value needs to be written to the SYNC_CTRL register.

NOTE

In case of the SYNC frequency changing, the IP-Core will synchronize automatically. During this synchronization the former sampling frequency is used. Please note that this synchronization takes a few SYNC periods.

4.2.3 Reset signal

`rst` is the reset input (high active) of the DSL Master IP Core.

After start-up (switching on) of the frequency inverter, a reset procedure is mandatory to return the DSL Master IP Core to its initialization condition.

The reset procedure is established by the parameters listed in table 8 and quoted in figure 8.

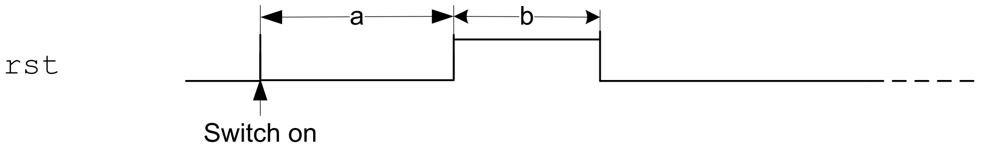


Figure 8: Reset procedure

Table 8: Reset time sequence

Diagram reference	Parameters in figure 8	Value (cf.figure 8)
a	Reset delay	Variable
b	Duration of the reset signal	>60 ns

Additional pin functions are described in detail in chapter 5.

4.3 Cable specification

The cable recommended for connecting the frequency inverter to the HIPERFACE DSL® motor feedback system is specified by the parameters set out in table 8. These technical data apply to all configurations.

In the case of integrated cables (see chapter 4.1.2), the motor cables are not listed.

Table 9: ***Technical data for the HIPERFACE DSL® cable***

Characteristic	Minimum	Typical	Maximum	Units
Length			100	m
Impedance at 10 MHz	100	110	120	Ω
DC loop resistance			0.1	Ω/m
Velocity ratio	0.66			c
Propagation delay		5		ns/m
Limit frequency	25			MHz
Maximum current per cable	0.25			A
Operating temperature	-40		125	°C

More information relating cable construction and installation are available on the Whitepaper “**Cable and Connector for HIPERFACE DSL® Motor and Drive Applications**”.

5 Interfaces

The IP Core of the DSL Master includes interfaces to the motor feedback system (DSL Slave) and to the frequency inverter application (see figure 9).

The motor feedback system communicates via a DSL connection with the DSL Master. All data channels between the DSL Master and DSL Slave are routed via this connection.

The user application is connected via one interface (choice of SPI or parallel bus) and several control signals . In addition, the frequency inverter provides a clock signal (CLK) and a reset signal (RST) to the DSL Master IP Core. By means of these signals, a defined start-up performance is achieved.

According to the requirements of the particular application, an optional serial interface (SPI-PIPE) can be employed to use the SensorHub Channel (see chapter 3.5).

The various interfaces correspond to the tasks described in table 10.

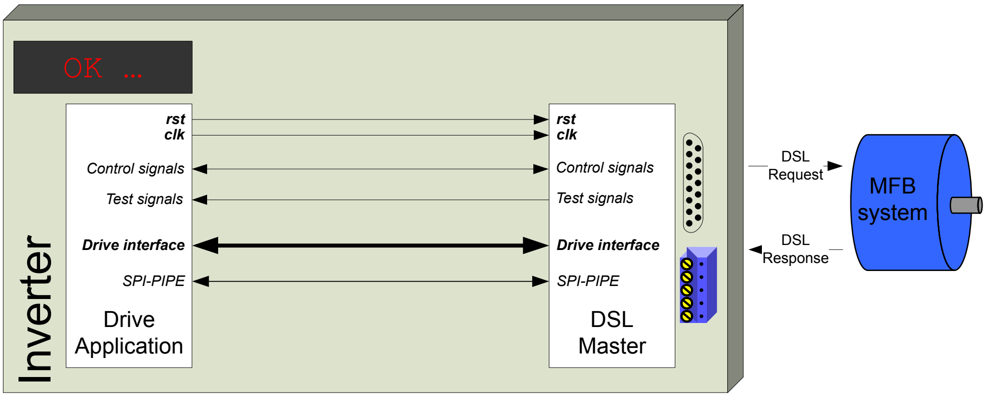


Figure 9: DSL system interfaces

Table 10: Interface functions

Interface	Function
Drive interface	Register-based access to all DSL Master and DSL Slave functions relevant for the core frequency inverter application
SPI PIPE	Optional register-based access to SensorHub Channel data
Control signals	DSL Master indication and control signals
Test signals	Test signals for development or fault-finding for a DSL controller
CLK	Clock signal for the IP Core circuit
RST	Reset signal for the IP Core circuit
DSL	Connection to the motor feedback system

5.1 Drive interface

The drive interface forms the central communications interface between the frequency inverter application and the DSL Master IP Core. Absolute and fast position data can be read via this interface. The functions of the motor feedback system are also accessible via this interface.

The following signals are used for Drive interface:

Table 11: Drive interface signals

Pin name	Type	Function
online_status_d(0:15)	Output	IP Core status (see chapter 6.2)

Pin name	Type	Function
hostd_a(0:6)	Input	Register address bus
hostd_di(0:7)	Input	Register input data bus
hostd_do(0:7)	Output	Register output data bus
hostd_r	Input	Read signal
hostd_w	Input	Write signal
hostd_f	Input	Freeze signal

Example installations of interface blocks for the Drive interface of the DSL Master are supplied together with the IP Core. These examples include a serial SPI interface and a parallel Texas Instruments EMIFA interface. For more information please see chapter 9.1.

5.2 SPI PIPE Interface

The SPI PIPE is a read-only Serial Peripheral Interface (SPI). SPI PIPE is an optional communication channel between the frequency inverter application and the DSL Master IP Core. Read processes on the SensorHub Channel can be carried out via this interface. Alternatively, this data can also be read from the registers by standard transactions via Drive interface.

The type of access to the SensorHub Channel is selected by setting or deleting the **SPPE** bits in the **SYS_CTRL** register (see chapter 6.3.1). If the **SPPE** bit is deleted, the data and the status of the SensorHub channel are accessible via the DSL Master **PIPE_S** (2Dh) and **PIPE_D** (2Eh) registers. If the **SPPE** bit is set, the SensorHub Channel is read using the SPI PIPE "Read Pipeline" transaction.

SPI PIPE should be activated if, at a fast frequency inverter cycle, the bandwidth of Drive interface is insufficient to access position and pipeline data, or if the pipeline data is being processed by another frequency inverter application resource.

NOTE
It should be noted that in every case, the configuration of external sensor components at the sHub® is carried out via the DSL Master Parameters Channel. The SPI PIPE provides only one read access to the SensorHub Channel (see chapter 3.5).

The SensorHub Channel data is kept in a FIFO (First In First Out) buffer that can hold 8 bytes. In addition, for each data byte, status information is also stored in the FIFO buffer (see chapter 6.3.22 and chapter 6.3.23).

NOTE
It should be noted that the FIFO buffer can only store 8 bytes of SensorHub Channel data. If the buffer is not read quickly enough, old data will be overwritten. This is indicated by a flag in the FIFO buffer status information.

The SPI Master for the SPI PIPE is the frequency inverter application. The SPI functions "Slave Selection" (Pin: `spipipe_ss`) and "clock" (Pin: `spipipe_clk`) are controlled by the frequency inverter application. The SPI function "Data, Master input Slave output" (Pin: `spipipe_miso`) is controlled by the DSL Master.

SPI PIPE has the following SPI characteristics:

- PHA = 1 (Sampling for clock trailing edge, data changes for clock leading edge)
- POL = 0 (Basic clock value)

The data with the highest value bit (MSB) is given first.

When accessing the SensorHub Channel via the SPI PIPE, the first four bits of the status buffer (0101) show a different value for each transaction, in order to check the correct function of the interface.

5.2.1 SPI-PIPE timing

The time sequence for SPI PIPE is shown in the time sequence diagram (figure 10) below and in table 12.

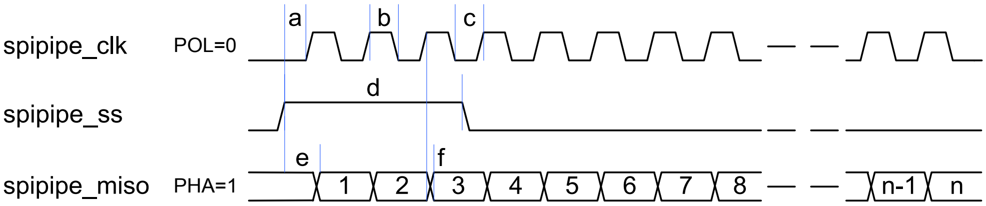


Figure 10: SPI-PIPE interface time control

Table 12: SPI-PIPE time control

Diagram reference	Description	Minimum	Maximum	Units
A	Assertion of `spipipe_ss` before `spipipe_clk`	30		ns
B	Time for `spipipe_clk` high	30		ns
C	Time for `spipipe_clk` low	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

5.2.2 Read pipeline

The SPI PIPE transaction "Read Pipeline" is used for access to the FIFO buffer values that contain the data and status of the SensorHub Channel.

Table 13: "Read Pipeline" transaction

Symbol	Meaning
PIPE STATUS	SensorHub Channel status (see chapter 6.3.21)
PIPE DATA	SensorHub Channel data (see chapter 6.3.22)

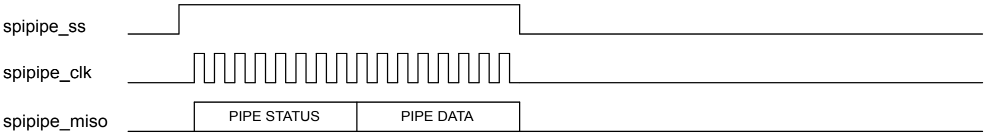


Figure 11: "Read Pipeline" transaction

5.3 Control signals

Various control signals are available between the DSL Master and the frequency inverter application to configure the performance of the IP Core or to carry out fast monitoring of the IP Core status.

5.3.1 SYNC signal

`sync` is a DSL Master digital input.

One edge on this pin triggers a position sampling. The polarity of the edge can be programmed using the **SPOL** bit in the **SYS_CTRL** (00h) register. The protocol requires a constant frequency of the signal at this pin, with deviations permitted within a set tolerance band. Continuously, the protocol synchronizes the protocol frame with the signal frequency at `sync`.

NOTE
If the `sync` signal frequency is outside the tolerance range, re-synchronization of the protocol is triggered. During the time that the re-synchronization is taking place, sampling is carried out with the former sync frequency until the re-synchronization is complete. For more details on the sync signal also see chapter 7.3.3.

5.3.2 INTERRUPT signal

`interrupt` is a DSL Master digital output.

`interrupt` is set to "1" if an interrupt condition has been fulfilled in the DSL-Master. The interrupt conditions are set using the registers **MASK_H**, **MASK_L** and **MASK_SUM** (see chapter 6.3.5 and chapter 6.3.6).

NOTE
During each write process in one of the registers **EVENT_H** or **EVENT_L**, the `interrupt` output is masked until the current SPI transaction has ended.

5.3.3 LINK signal

`link` is a DSL Master digital output.

`link` is effected by the content of the LINK bit in the **MASTER_QM** register (see chapter 6.3.3) and therefore indicates whether the DSL Master has produced a communications link to a connected HIPERFACE DSL® motor feedback system.

`link` is intended to be a control signal for an LED display, but can also be used to control the start-up performance (see chapter 7.1) or for global error handling.

`link` is reset if communication faults are detected.

5.3.4 FAST_POS_RDY signal

`fast_pos_rdy` is a DSL Master digital output.

`fast_pos_rdy` signals that a new fast position value is available and permits an event-based reading of the position for incorporating latency reduction.

`fast_pos_rdy` is always available, even if the position value is invalid or no connection to the encoder has been established.

Dependent upon the configuration in the register system control (see chapter 6.3.1), `fast_pos_rdy` displays either only the availability of positions based on user requirements (edge at `sync` input) or all transmitted positions.

5.3.5 SYNC_LOCKED signal

`sync_locked` is a DSL Master digital output.

`sync_locked` indicates whether the `sync` signal was correctly passed to the encoder, or whether the IP Core is still in a synchronization phase. `sync_locked` drops to “0” when the SYNC edge supplied by the application has been transported with more than 2 clock cycles of distortion.

5.3.6 BIGEND signal

bigend is a DSL Master digital input.

The byte sequence of the address allocation for registers can be influenced via bigend (see chapter 9.5). The byte sequence is based on 32 bit-wide data words. The selection influences the allocation independently of the interface block used.

Table 16 below lists the selection options for bigend.

Table 14: bigend selection

Value	Address allocation byte sequence
0	Little endian
1	Big endian

5.4 Test signals

To support development or fault-finding for controllers that have a DSL interface integrated, the DSL Master supplies some test signals.

5.4.1 SAMPLE signal

sample is a DSL Master digital output.

The sample signal is set at the sampling time point of each bit that is transmitted from the DSL motor feedback system. It consists of 50 pulses from Channel 1 followed by a bit pause and 10 pulses from Channel 2 of the motor feedback system.

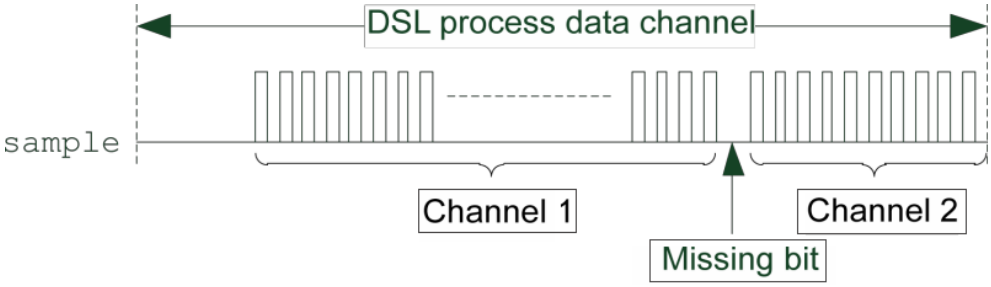


Figure 12: Sample signal

The sample signal can be used for eye diagrams to measure time and voltage margins during signal transmission.

When making the evaluation, signal delays in the DSL Master must be taken into account. The rising edge of the sample signal is offset by 40 ns from the line driver signal. The time delay of the line driver must also be taken into account. Typically this is 13 ns.

5.4.2 ESTIMATOR_ON signal

estimator_on is a DSL Master digital output.

The estimator_on signal is set if some event leads to the transmitted fast position (see chapter 6.3.12) being invalid and the position estimator supplying the values. Such events are:

- The DSL motor feedback system reporting a position error
- A coding error in transmission of the fast position
- A check-sum error in transmission of the fast position
- Realignment from safe to fast - fast position is forced to the safe position value
- The protocol is re-synchronizing following a break in the link

The estimator_on signal can be used to carry out a statistical analysis of the incidence of errors in the DSL system. For more information, please refer to chapter 7.3.1. It should be noted that the POS flag and estimator_on signal information is redundant.

5.4.3 DEV_THR_ERR signal

dev_thr_err is a DSL Master digital output.

If the fast position from the process data channel cannot be calculated (either due to transmission or due to sensor errors), estimation is made by the DSL Master based on the last two available position values of the Safe Channel. The worst case deviation from the actual mechanical position is also provided, referring to a user- defined parameter for the maximum possible acceleration in the application (fast position acceleration boundary, see chapter 6.3.25).

A threshold can be set up for a worst case deviation to raise the dev_thr_err output. The threshold is a user defined parameter for the maximum tolerable deviation in the application (fast position estimator deviation, see chapter 6.3.26).

dev_thr_err indicates whether the maximum tolerable deviation is violated (‘1’) or kept (‘0’).

5.4.4 SAFE_CHANNEL_ERR signal

safe_channel_err is a DSL Master digital output.

The safe_channel_err signal is set if some event leads to the safe position or status (see chapter 6.3.15) being invalid. Such events are:

- A coding error in transmission of the safe position
- A check-sum error in transmission of the safe position

The safe_channel_err signal can be used to carry out a statistical analysis of the incidence of errors in the DSL system.

5.4.5 SAFE_POS_ERR signal

safe_pos_err is a DSL Master digital output.

safe_pos_err is a DSL Master digital output. The safe_pos_err signal is set if the safe position of Safe Channel 1 is not updated or it has never been written since the startup. Another possible cause can be the DSL motor feedback system reporting a position error.

5.4.6 ACCELERATION_ERR signal

acceleration_err is a DSL Master digital output.

The acceleration_err signal is set if an encoding error was detected after transmission of a fast position value, or if the encoder has transmitted an invalid acceleration cause of internal errors.

If the acceleration_err signal is set the error counter acc_err_cnt will be incremented with each transmission (see chapter 6.3.24). As soon as the acceleration_err signal is reset the error counter acc_err_cnt will be set to “0” again.

5.4.7 ACC_THR_ERR signal

acc_thr_err is a DSL Master digital output.

The acc_thr_err signal is set if the threshold programmed in register

acc_err_cnt is exceeded.

The acc_thr_err signal can be used to implement a fault-tolerant evaluation in the drive. For this the maximum position deviation should be calculated from the number of transmission errors.

5.4.8 ENCODING_ERR signal

encoding_err is a DSL Master digital output.

The encoding_err signal is set if the underlying 8B/10B encoding of a DSL frame transmission is disturbed.

The encoding_err signal can be used to make a statistical analysis of the bit error rate of a DSL system.

6 Register map

The DSL Master is accessible via register in three different register blocks. Each register block has its own address area (see table 15).

Table 15: Register blocks address area

Register block	Address area	Functions
Drive	00h to 3Fh 60h to 6Fh	Process data Channel, position/status Parameters Channel (long messages) SensorHub Channel
Safe 1	00h to 7Fh	Safe Channel 1, position/status Parameters Channel (short messages)
Safe 2	00h to 3Fh	Safe Channel 2, position/status

All IP Core registers and functions can be accessed via drive interface. As an option, the SensorHub Channel data is accessible via the SPI PIPE interface.

In addition, the DSL Slave interface registers are mirrored as decentralized registers. The address area 40h to 7Fh is intended for this. The addressing of these registers is identical to the addressing of the registers in the DSL Master. The answer to the transaction is, however, delayed and must be read individually (see under "Short message", in chapter 7.5.1).

figure 13 below shows via which interface a connection to which register block is established.

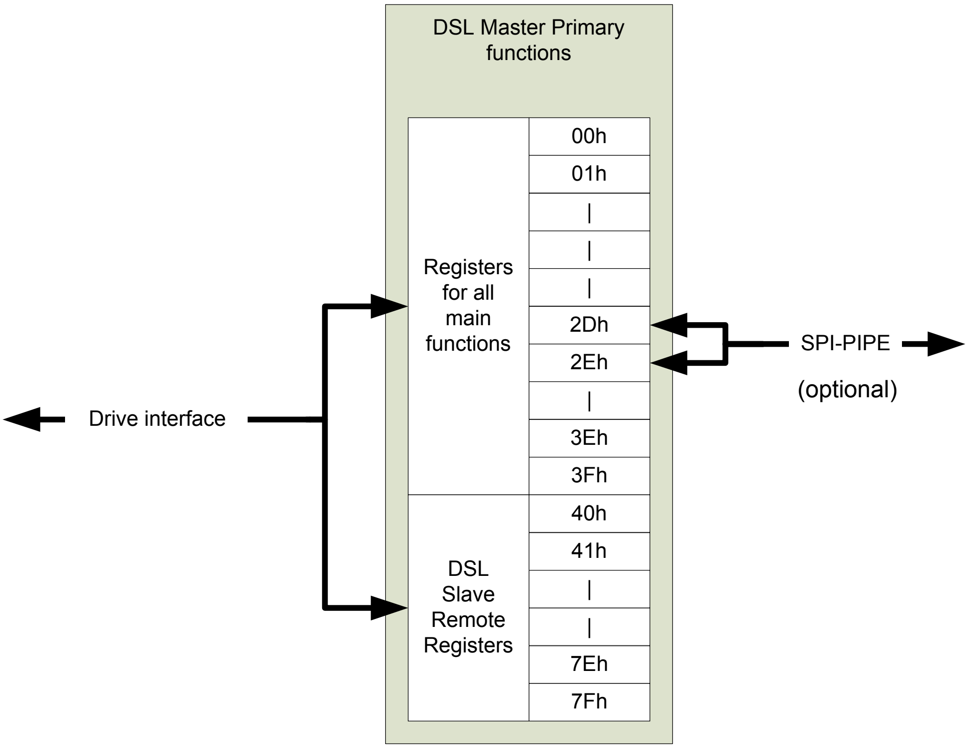

DSL Master Primary functions
Registers for all main functions
DSL Slave Remote Registers
00h
01h
|
|
|
2Dh
2Eh
|
3Eh
3Fh
40h
41h
|
|
7Eh
7Fh
Drive interface
SPI-PIPE
(optional)

Figure 13: Register block overview

6.1 Explanation of the registers

In the following description of the registers, symbols are used to describe the standard value of a bit following a reset. Additional symbols are used to describe the functions provided to the frequency inverter application for this bit.

The bit is described according to the following example: "Function" "Reset value", e.g. "R/W-0"

Table 16: Function symbols for bits

Function symbol	Meaning
R	Bit can be read.
W	Bit can be set and deleted.
C	Bit can only be deleted.
X	Bit is not installed and will always be read as "0".

Table 17: Symbols for bit reset values

Reset value	Meaning
0	The bit is deleted after a reset.
1	The bit is set after a reset.
x	After a reset, the bit has no defined value.
-	In the register diagram: The bit is not installed and will always be read as "0".

NOTE
It should be noted that read access to a bit that can only be written ("W") always returns the value "0". If a register address that is not used is read, the result will be “0” as well.

6.2 Online Status D

The Online Status D is a non-storing copy of registers **EVENT_H** and **EVENT_L**. The static information in these registers must be deleted by the user after the read process, by writing the value "0" to the corresponding bit in the register, whilst the Online Status D only shows the current status without storing previous indications. The signal name of the Online Status is `online_status_d` (with d indicating drive).

`online_status_d` is given in two bytes. If an SPI block is used for interfacing the IP Core, `online_status_d` is transmitted in each transaction in the first two bytes via the `spi_miso` output. When a parallel bus interface is used for drive interface, `online_status_d` has 16 dedicated output signals available.

NOTE
It should be noted that when the parallel bus interface is used the 16 signals of the Online Status D are not frozen during a read access. If required, the user can insert a Latch (e.g. using the hostd_fcode.inlinesignal).

*Table 18: **Online Status D, High Byte***

R-0	R-0	R-1	R-1	R-1	R-1	R-1	R-1
INT	SUM	SCE	FIX1	POS	VPOS	DTE	PRST
Bit 7							Bit 0

Bit 7 **INT**: Status of the Interrupt output

This bit represents an exception to the Online Status D, as this bit does not relate to an event indication. INT provides the value of the physical INT output so that request management (polling) can be established. The importance of this flag depends on the Interrupt sources monitored.

1 = interrupt output on "High" level
0 = `interrupt` output on "Low" level

Bit 6 **SUM**: Summary byte

1 = The last valid value from **SUMMARY** was not zero. The importance of this flag depends on the particular error source that leads to a set **SUMMARY** (see chapter 6.3.14).

0 = The last valid value from **SUMMARY** was zero.

Bit 5 **SCE:** CRC error on the Safe Channel

1 = The last Safe Channel CRC received was wrong. It is expected that the last safe position transmitted (see chapter 6.3.16) is invalid.

0 = The last Safe Channel CRC received was correct.

Bit 4 **FIX1**: This bit always gives a "1". For SPI interfaces, this is used for checking the `spi_miso` pin for stuck-at- '0' faults.

Bit 3 **POS**: Estimator turned on (functionality based on `estimator_on`, see chapter 5.4.2)

1 = A source of an error in the fast position was identified or an alignment procedure is currently being carried out. It is probable that the last fast position is invalid. Be aware that the fast position read through drive interface is provided by the estimator.

0 = No fast position error.

Bit 2 **VPOS**: Safe position invalid

1 = An error in the safe position was identified. It is expected that the safe position transmitted from the encoder is invalid.

0 = The last safe position received was correct.

Bit 1 **DTE**: Deviation Threshold Error (see chapter 5.4.3)

1 = Current value of deviation greater than the specified maximum.

0 = Current value of deviation smaller than the specified maximum.

Bit 0 **PRST**: Protocol reset

1 = IP-Core has restarted the protocol.

0 = IP-Core running.

Table 19: ***Online Status D, Low Byte***

X-0	X-0	R-0	R-0	R-0	R-1	R-0	R-0
POSTX		MIN	ANS	FIX0	QMLW	FREL	FRES
Bit 7							Bit 0

Bit 7-6 **POSTX1:POSTX0**: Position transmission status

00: Position request is transmitted to the DSL encoder.

01: Safe Channel was correctly transmitted.

10: Fast position present (see chapter 6.3.11).

11: Safe Channel 2 was correctly transmitted.

Recommended to trigger on rising edge of a POSTX bit change (is set). A new state is indicated by a set bit. It is cleared/maintained until a new frame/event arrives.

Bit 5 **MIN**: Acknowledgment of message initialization

1 = The DSL encoder sends a figure by which the initialization of the Parameter Channel is acknowledged.

0 = Parameter Channel not functioning.

Bit 4 **ANS**: Incorrect answer detected.

1 = The last answer to a long message was damaged.

0 = No error detected in the last answer to a long message.

Bit 3 **FIX0**: This bit always gives a "0". For SPI interfaces, this is used for checking the `spi_miso` pin for stuck-at-'1' faults.

Bit 2 **QMLW**: Quality monitoring at Low level (see chapter 6.3.3)

1 = Current value of quality monitoring less than 14.

0 = Current value of quality monitoring greater than or equal to 14.

Bit 1 **FREL**: Channel status for “long message”.

1 = The channel for the “long message” is free.

0 = The channel for the “long message” is in use.

Bit 0 **FRES**: Channel status for the "short message". 1 = The channel for the "short message" is free.

0 = The channel for the "short message" is in use.

6.3 DSL Master function register

The protocol logic controls the performance of the DSL Master via the registers in the DSL Master IP Core on drive interface. These registers are also used for accessing the position values.

The table below contains a list of all the function registers available in the IP Core.

NOTE
The addresses given below are referencing a big-endian addressing. For a table stating the register addresses depending on the endianness, see chapter 9.5.

Table 20: Description of the registers in DSL Master, drive interface

Addr	Designation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value at reset
00h	SYS_CTRL	PRST	MRST	FRST	LOOP	PRDY	SPPE	SPOL	OEN	0000 0000
01h	SYNC_CTRL	ES								0000 0001
03h	MASTER_QM	LINK	-	-	-	Quality monitoring				0--- 0000
04h	EVENT_H	INT	SUM	SCE	-	POS	VPOS	DTE	PRST	000- 0000
05h	EVENT_L	-	-	MIN	ANS	-	QMLW	FREL	FRES	--00 -000
06h	MASK_H	-	MSUM	MSCE	-	MPOS	MVPOS	MDTE	MPRST	-00- 0000
07h	MASK_L	-	-	MMIN	MANS	-	MQMLW	MFREL	MFRES	--00 -000
08h	MASK_SUM	MSUM7:0								0000 0000
09h	EDGES	Bit sampling pattern								0000 0000
0Ah	DELAY	RSSI				Cable delay				0000 0000
0Bh	VERSION	Coding		IP Core version number						0101 0111
0Ch	RELEASE	Release date FIFO								0000 0000
0Dh	ENC_ID2	-	SCI			ENC_ID19:16				-000 0000
0Eh	ENC_ID1	ENC_ID15:8								0000 0000
0Fh	ENC_ID0	ENC_ID7:0								0000 0000
10h	POS4	Fast position, byte 4								0000 0000
11h	POS3	Fast position, byte 3								0000 0000
12h	POS2	Fast position, byte 2								0000 0000

Addr	Designation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value at reset
13h	POS1	Fast position, byte 1								0000 0000
14h	POS0	Fast position, byte 0								0000 0000
15h	VEL2	Speed, byte 2								0000 0000
16h	VEL1	Speed, byte 1								0000 0000
17h	VEL0	Speed, byte 0								0000 0000
18h	SUMMARY	SUM7:0								0000 0000
19h	VPOS4	Safe position, byte 4								0000 0000
1Ah	VPOS3	Safe position, byte 3								0000 0000
1Bh	VPOS2	Safe position, byte 2								0000 0000
1Ch	VPOS1	Safe position, byte 1								0000 0000
1Dh	VPOS0	Safe position, byte 0								0000 0000
1Eh	VPOSCRC_H	CRC of the safe position, byte1								0000 0000
1Fh	VPOSCRC_L	CRC of the safe position, byte0								0000 0000
20h	PC_BUFFER0	Parameters Channel, byte0								0000 0000
21h	PC_BUFFER1	Parameters Channel, byte1								0000 0000
22h	PC_BUFFER2	Parameters Channel, byte2								0000 0000
23h	PC_BUFFER3	Parameters Channel, byte3								0000 0000
24h	PC_BUFFER4	Parameters Channel, byte4								0000 0000
25h	PC_BUFFER5	Parameters Channel, byte5								0000 0000
26h	PC_BUFFER6	Parameters Channel, byte6								0000 0000
27h	PC_BUFFER7	Parameters Channel, byte7								0000 0000
28h	PC_ADD_H	LID	LRW	LOFF	LIND	LLEN		LADD9:8		1000 0000
29h	PC_ADD_L	LADD7:0								0000 0000
2Ah	PC_OFF_H	LID	LOFFADD14:8							1000 0000
2Bh	PC_OFF_L	LOFFADD7:0								0000 0000
2Ch	PC_CTRL	-	-	-	-	-	-	-	LSTA	---- ---0
2Dh	PIPE_S	-	-	-	-	POVR	PEMP	PERR	PSCI	---- 0000
2Eh	PIPE_D	SensorHub FIFO, output								0000 0000
2Fh	PC_DATA	"Short message" data								0000 0000
38h	ACC_EE_CNT	-	-	-	Acc. errorthreshold/counter					---0 0000
39h	MAXACC	Acc. Res.		Acc. Mantissa						1111 1111
3Ah	MAXDEV_H	Max. position deviation, byte1								0000 0000
3Bh	MAXDEV_L	Max. position deviation, byte0								0000 0000
3Fh	DUMMY	No data								---- ----

NOTE

It should be noted, that some registers are shared (e.g. 09h **EDGES**) and others are just used for accessing the same data (**SYS_CTRL** and **SAFE_CTRL**, for instance). In most cases though, both interfaces use registers that are exclusively available to them.

6.3.1 System control

The system control register `SYS_CTRL` contains the main control bits of the DSL Master.

NOTE
It should be noted that apart from a reset of the Master, all system control bits can only be set and deleted by the user.

Register 00h: **System control**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PRST	MRST	FRST	LOOP	PRDY	SPPE	SPOL	OEN
Bit 7							Bit 0

Bit 7 **PRST**: Protocol reset

1 = A forced reset of the protocol status will be initiated. If the bit is deleted, a restart of the connection is triggered.

0 = Normal protocol action

Bit 6 **MRST**: Messages reset

1 = The Parameters Channel is reset. Current short and long messages are discarded.

0 = Normal Parameters Channel action

Bit 5 **FRST**: Pipeline FIFO, reset

1 = The FIFO is reset. Data is not stored and cannot be read.

0 = Normal FIFO access

Bit 4 **LOOP**: Test drive interface

Value for the read back test for drive interface. This value has no other purpose.

Bit 3 **PRDY**: POS_READY mode.

1 = `pos_ready` shows time of receipt of all position transmissions.

0 = `pos_ready` shows only the time of receipt of position transmissions following a control clock (`sync` input).

Bit 2 **SPPE**: SPI-PIPE activation

1 = SPI-PIPE activated. Access to pipeline status and data via SPI-PIPE. The registers **PIPE_S** and **PIPE_D** are read as "0".

0 = SPI-PIPE deactivated. Access to pipeline status and data via the registers **PIPE_S** and **PIPE_D**.

Bit 1 **SPOL**: Polarity of the synchronization pulse

1 = The `sync` trailing edge is used.

0 = The `sync` leading edge is used.

Bit 0 **OEN**: Activation of the output

1 = The DSL cables are activated for output to the DSL Slave.

0 = The impedance of the DSL cable is high.

6.3.2 Synchronization control

The `SYNC_CTRL` register for control of the synchronization contains the bit with which the synchronization source for position sampling is controlled.

Register 01h: **Synchronization control**

W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-1
ES							

Bit 7							Bit 0

Bit 7-0 **ES**: External synchronization

00000000 = Position sampling during free running at the shortest cycle time.

All other values = Position sampling with the `sync` signal synchronized. The value from **ES** determines the number of position samplings carried out in one `sync` cycle. The user must match the number of samplings per cycle to the shortest frame length (see chapter 7.3.3).

6.3.3 Quality monitoring

The `MASTER_QM` quality monitoring register contains the quality monitoring value for the data connection.

As soon as the DSL Master detects events that indicate an improvement or degradation of the quality of the data connection, these events are indicated as values higher or lower than the quality monitoring value (see table 21).

Table 21: Quality monitoring events

Protocol event	**Value change in quality monitoring**
Wrong synchronization in Safe Channel (last byte)	-4
Wrong synchronization in Safe Channel (1st ... 7th bytes)	-6
RSSI <2	-4
Wrong encoding in parameter or SensorHub channels	-1
Wrong encoding in process data channel	-2
Unknown special characters in the protocol package	-2
Any identified error in Safe Channel 1	-6
Any identified error in Safe Channel 2	-8
Correct synchronization in Safe Channel	+1
Correct CRC value in Safe Channel	+1

Quality monitoring is initiated with the value "8".

The maximum quality monitoring value is "15". This is the standard value during operation.

NOTE
Particular attention must be paid to the quality monitoring value during the development of a DSL drive controller. If a value lower than "15" is indicated, the cause may be a problem with the connection circuit, particularly if the value is continuously displayed.

If the quality monitoring value falls below "14", **QMLW** information is indicated in Online Status and in the **EVENT_L** register.

If the quality monitoring value falls to "0", a forced reset of the protocol is carried out. This is indicated by the **PRST** error bit in Online Status and in the **EVENT_H** register.

The `MASTER_QM` register is write protected.

Register 03h:

Quality monitoring register

R-0	X-0	X-0	X-0	R-0	R-0	R-0	R-0
LINK				QM			
Bit 7							Bit 0

Bit 7 **LINK**: DSL protocol connection status

1 = Protocol connection between DSL Master and Slave was established.

0 = No connection present or connection error due to a communications error.

It should be noted that **LINK** is also represented at the `link` interface output (see chapter 5.3).

Bit 6-4 **Not implemented**: Read as "0".

Bit 3-0 **QM3:QM0**: Quality monitoring bits

0000 to 1111: Quality monitoring value. Higher values indicate a better connection. If the quality monitoring reaches the value "0000", a forced reset of the protocol is carried out.

6.3.4 Events

The `EVENT_H/EVENT_L` registers contain the messaging bits for all warning and error modes of the DSL system.

All messaging bits are set by the DSL Master if a corresponding status is determined.

The following bit description lists the effects of warning and error conditions as well as the reactions to errors that must be installed in the frequency inverter application.

An event bit that has been set is not reset by the DSL Master. The frequency inverter application must delete bits that have been set.

Both edge and level-sensitive flags are present in the `EVENT` registers. Edge- sensitive bits are set when the corresponding status arises. They are only set again if the corresponding status disappears and then arises once more. This is the standard action. The level-sensitive bits set a bit as long as the corresponding status exists.

NOTE
It should be noted that all event register bits are also transferred to Online Status D (see chapter 6.2 and chapter 7.6.2). The event bits are not static there and contain the actual status of each individual event.

Register 04h: **High Byte events**

R-0	R/C-0	X-0	X-0	R/C-0	X-0	R/C-0	R/C-0
INT	SUM			POS		DTE	PRST
Bit 7							Bit 0

Bit 7 **INT**: Interrupt status

This bit reflects the status of the `interrupt` signal (see chapter 5.3.2).

Bit 6 **SUM**: Remote event monitoring

1 = The DSL Slave has signaled an event and the summary mask is set accordingly (see registers `MASK_SUM` and `SUMMARY`).

0 = All DSL Slave events are deleted.

When the **SUM** bit is set, an error or a warning has been transmitted from the DSL Slave. The frequency inverter application must check the `SUMMARY` register (see chapter 6.3.14) to obtain a detailed description. This bit is level sensitive.

Bit 5 **SCE:** Error on the Safe Channel

1 = Data consistency error on the Safe Channel.

0 = Safe Channel data was correctly transmitted.

This error usually indicates a transmission error on the DSL connection. If this error occurs frequently, the wiring of the DSL connection should be checked. If this error occurs continuously, there is probably an error in the motor feedback system.

This error affects quality monitoring and produces the **QMLW** warning or a protocol reset.

: Read as "0".

⚠ **CAUTION**
When this error occurs, the last valid value of the safe position is retained (see chapter 6.3.15). Only if a fresh value has been transmitted will the safe position be updated and correspond to the actual position of the motor feedback system. The fast position is not affected by this.

Bit 4 **Not implemented**: Read as "0".

Bit 3 **POS**: Estimator turned on (functionality based on `estimator_on`, see chapter 5.4.2)

1 = Fast position data consistency error. The fast position read through drive interface is supplied by the estimator.

0 = The data for the fast position was correctly transmitted.

This error usually indicates a transmission error on the DSL connection. If this error occurs frequently, the wiring of the DSL connection should be checked. If this error occurs continuously, there is probably an error in the motor feedback system.

⚠ **CAUTION**
When this error occurs, the fast position is updated by the estimator (see chapter 7.3.1).

Bit 2 **VPOS**: Safe position error 1 = Sensor error.

0 = The safe position is correct.

This error usually indicates an encoder sensor error. If this error occurs continuously, there is probably an error in the motor feedback system.

⚠ **CAUTION**
When this error occurs, the error value FD FD FD FD FDh is displayed instead of the safe and fast position.

Bit 1 **DTE**: Estimator Deviation Threshold Error (see chapter 5.4.3)

1 = Current value of deviation greater than the specified maximum.

0 = Current value of deviation smaller than the specified maximum.

⚠ **CAUTION**
This error message is relevant when the estimator deviation threshold is used. See chapter 7.3.1 for details of this function.

Bit 0 **PRST**: Protocol reset warning

1 = The forced protocol reset was triggered.

0 = Normal protocol action

This error message indicates that the protocol connection to the DSL Slave has been re-initialized. This error message can be caused by a frequency inverter application request (**PRST** bit in `SYS_CTRL`), generated by the DSL Master itself, or activated via the `rst` input.

The DSL Master causes a protocol reset if too many transmission errors indicate a connection problem (see chapter 6.3.3). A protocol reset causes a re-synchronization with the DSL Slave that can improve the connection quality.

Register 05h: **Low Byte events**

X-0	X-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	X-0
		MIN	ANS		QMLW	FREL	
Bit 7							Bit 0

Bit 7-6 **Not implemented**: Read as "0".

Bit 5 **MIN**: Message initialization

1 = An acknowledgment was received from the Slave for the initialization of a message.

0 = No acknowledgment for the initialization received.

When this warning is displayed, the Parameters Channel is still in the initialization status and no "short message" or "long message" can be triggered.

This bit is level sensitive.

Bit 4 **ANS**: Erroneous answer to "long message"

1 = An error occurred during the answer to a long message. The effectiveness of the previous transaction is not known.

0 = The last answers to "long messages" were error free.

This error indicates that the transmission of an answer from the DSL Slave to the last "long message" failed. The frequency inverter application must send the "long message" again.

Bit 3 **Not implemented**: Read as "0".

Bit 2 **QMLW**: Quality monitoring low value warning

1 = Quality monitoring value (see register 03h) below "14"

0 = Quality monitoring value greater than or equal to "14"

This warning indicates that a transmission error occurred at bit level for one of the CRC values. If this error occurs frequently, the wiring of the DSL connection should be checked (also see chapter 6.3.3).

Bit 1 **FREL**: Channel free for "long message"

1 = A "long message" can be sent on the Parameters Channel.

0 = No "long message" can be sent.

If the bit is set, the frequency inverter application can trigger a "long message". Provided no answer has been received from the DSL Slave, this bit remains deleted. As the processing duration of a "long message" in the motor feedback system is not specified, a user time limit condition should be installed via the frequency inverter application. When a time limit is exceeded, the **MRST** bit in the **SYS_CTRL** register is set, which causes the Parameters Channel to be reset.

Bit 0 **FRES**: Channel free for "short message"

1 = A "short message" can be sent on the Parameters Channel.

0 = No "short message" can be sent.

If the bit is set, the frequency inverter application can trigger a "short message". Provided no answer has been received from the DSL Slave, this bit remains deleted. As the processing duration of a "long message" in the motor feedback system is not specified, a time limit condition is installed in the DSL Master. If the time limit is exceeded, attempts are made again automatically (see under **RET** bit).

6.3.5 Event mask

In the event mask registers `MASK_H/MASK_L`, the events are set with which the `interrupt` signal is set.

Several events can be masked to trigger an interrupt. In addition, events from the DSL Slave summary can be combined with these events (see chapter 6.3.6). This is explained in figure 14 .

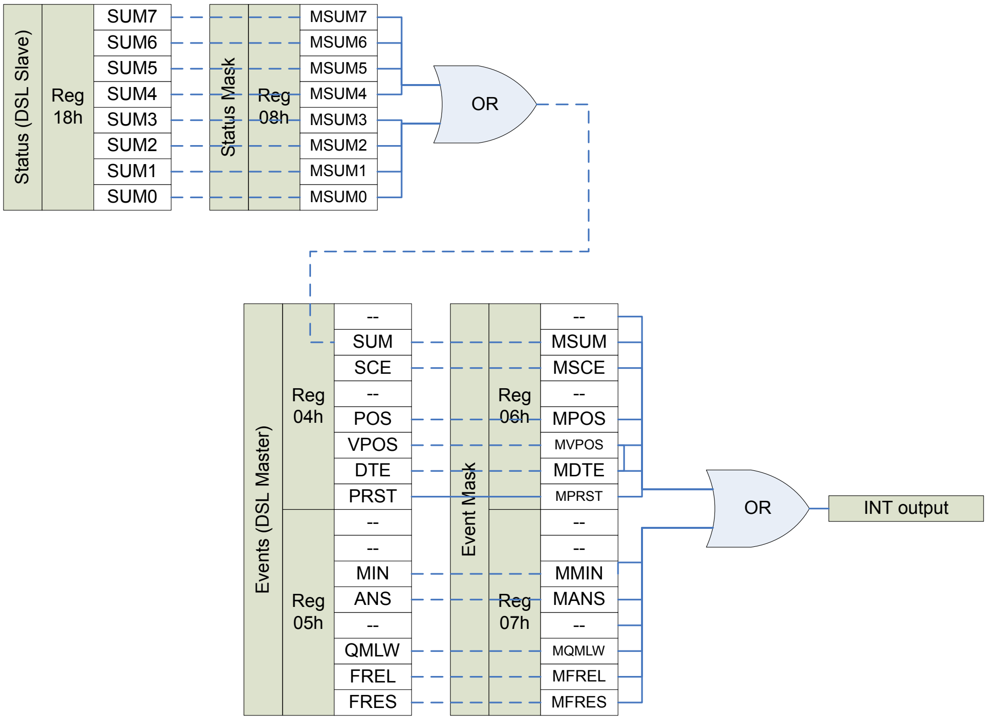


Figure 14: Interrupt masking

NOTE
It should be noted that the **SUM** bit is an OR connection of all bits of the status bit is an OR connection of all bits of the status summary (`SUMMARY` register).

Register 06h: **High Byte event mask**

X-0	W-0	X-0	X-0	W-0	X-0	W-0	W-0
	MSUM			MPOS		MDTE	MPRST
Bit 7							Bit 0

Bit 7 **Not implemented**: Read as "0".

Bit 6 **MSUM**: Mask for remote event monitoring

1 = DSL Slave events that are masked in the `SUMMARY` register set the `interrupt` signal.

0 = DSL Slave events that are masked in the `SUMMARY` register do not set the `interrupt` signal.

Bit 5 **MSCE**: Mask for transmission errors on the Safe Channel

1 = A transmission error on the Safe Channel sets the `interrupt` signal.

0 = A transmission error on the Safe Channel does not set the `interrupt` signal.

Bit 4 **Not implemented**: Read as "0".

Bit 3 **MPOS**: Mask for fast position error

1 = An error in the fast position sets the `interrupt` signal.

0 = An error in the fast position does not set the `interrupt` signal.

Bit 2 **MVPOS**: Mask for safe position error

1 = An error in the safe position sets the `interrupt` signal.

0 = An error in the safe position does not set the `interrupt` signal.

Bit 1 **MDTE**: Mask for estimator deviation threshold error warning

1 = A high estimator deviation threshold error sets the `interrupt` signal.

0 = A high deviation threshold error value does not set the `interrupt` signal.

Bit 0 **MPRST**: Mask for protocol reset warning

1 = A protocol reset sets the `interrupt` signal.

0 = A protocol reset does not set the `interrupt` signal.

Register 07h: **Low Byte event mask**

X-0	X-0	W-0	W-0	X-0	W-0	W-0	X-0
		MMIN	MANS		MQMLW	MFREL	
Bit 7							Bit 0

Bit 7-6 **Not implemented**: Read as "0".

Bit 5 **MMIN**: Mask for message initialization confirmation

1 = The acknowledgment for the initialization of a DSL Slave message sets the `interrupt` signal.

0 = The acknowledgment for the initialization of a DSL Slave message does not set the `interrupt` signal.

Bit 4 **MANS**: Mask for erroneous answer to long message

1 = A transmission error during the answer to a long message sets the `interrupt` signal.

0 = A transmission error during the answer to a long message does not set the `interrupt` signal.

Bit 3 **Not implemented**: Read as "0".

Bit 2 **MQMLW**: Mask for low quality monitoring value warning

1 = A low quality monitoring value (see registers 03h and 05h) sets the `interrupt` signal.

0 = A low quality monitoring value does not set the `interrupt` signal

Bit 1 **MFREL**: Mask for "channel free for "long message""

1 = If a "long message" can be sent on the Parameters Channel, the `interrupt` signal is set.

0 = If a "long message" can be sent on the Parameters Channel, the `interrupt` signal is not set.

Bit 0 **MFRES**: Mask for "channel free for "short message"'

1 = If a "short message" can be sent on the Parameters Channel, the `interrupt` signal is set.

0 = If a "short message" can be sent on the Parameters Channel, the `interrupt` signal is not set.

6.3.6 Summary mask

In the `MASK_SUM` summary mask register, the DSL Slave collective events are determined with which the **SUM** event monitoring in the event register as well as the signal to the interrupt pin are set (interrupt).

Several events can be masked to trigger an interrupt. In addition, events from the DSL Master can be combined with these events (see chapter 6.3.4).

NOTE
It should be noted that the **MSUM** bit from the `MASK_H` register is an OR connection of all bits of the summary mask register.

Register 08h: **Summary mask**

W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
MSUM7	MSUM6	MSUM5	MSUM4	MSUM3	MSUM2	MSUM1	MSUM0
Bit 7							Bit 0

Bit 7-0 **MSUM7:MSUM0**: Mask for status summary bits

1 = In the set status, the corresponding status summary bit sets the **SUM** event monitoring and the signal at the `interrupt` pin.

0 = In the set status, the corresponding status summary bit does not set the **SUM** event monitoring and the signal at the `interrupt` pin.

6.3.7 Edges

The `EDGES` edge register contains the time control for the DSL cable bit sampling and can be used to monitor the connection quality.

Each individual edge register bit is set if, at system start-up, an edge of the test signal is detected during the time period of the corresponding bit. An edge is defined as a change in cable value between successive detections. The sampling is carried out eight times as fast as the cable bit rate.

Clean cable signals mean that only a few bits are set in the edge register, whilst noisy cable signals set a large number of bits.

CAUTION
If all bits in the edge register are set, this is an indication of excessive interference in the cable in which no connection can be established.

The register is write protected. The content of this register does not change after the start-up phase. A new bit sampling pattern is only generated after a forced reset of the protocol.

Register 09h: **Edges**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Bit sampling pattern							

Bit 7							Bit 0

Bit 7-0 **Bit sampling pattern**: Identification of edges in the cable signal

1 = An edge was detected in the time period of the corresponding bit.

0 = No edge was detected in the time period of the corresponding bit.

6.3.8 Delay / RSSI

The `DELAY` run time register stores information about the run time delay of the system cable and the signal strength. The register can be used to monitor the connection quality.

The register is write protected.

Register 0Ah: **Run time delay**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
RSSI				Cable delay			
Bit 7							Bit 0

Bit 7-4 **RSSI**: Indication of the received signal strength

4 bit value for the cable signal strength, from "0" to "12". Higher values indicate better connection quality. Low connection (<2) quality affects QM.

RSSI is continuously updated during operation and used for signal monitoring during run time.

Bit 3-0 **Cable delay:**

4 bit value for cable delay. This value gives the cable signal round trip delay of cable and transceivers in bits. This value enables a rough estimate of cable length to be made.

The value for **Line Delay** does not change after the start-up phase. A fresh value for **Line Delay** is only measured after a forced reset of the protocol.

table 22 below shows the relationship between the value in **Line Delay** and the cable length of the DSL connection.

Table 22: Cable delay

Cable delay	DSL connection cable delay	Cable length DSL connection
0	<100 ns	< 10 m
1	100 to 200 ns	10 to 20 m
2	200 to 300 ns	20 to 30 m
3	300 to 400 ns	30 to 40 m
4	400 to 500 ns	40 to 50 m
5	500 to 600 ns	50 to 60 m
6	600 to 700 ns	60 to 70 m
7	700 to 800 ns	70 to 80 m
8	800 to 900 ns	80 to 90 m
9	900 to 1000 ns	90 to 100 m

CAUTION
A value above "9" indicates a delay of greater than 1 μs. Such a value will lead to a violation of the specification for cycle time. In this case, a check should be made of whether the cable complies with the cable specification.

6.3.9 Version

The `VERSION` register contains the release version of the DSL Master IP Core. The register is write protected.

Register 0Bh: **Version**

R-0	R-1	R-0	R-1	R-0	R-1	R-1	R-0
Coding		Major Release		Minor Release			
Bit 7							Bit 0

Bit 7-6 **Coding:** Type of IP Core 01= DSL Master IP Core

Bit 5-4 **IP Core Major release number:**

The current version is 1 (01).

Bit 3-0 **IP Core Minor release number:**

The current version is 7 (0111).

6.3.10 Release Date

The RELEASE register contains the release date of the DSL Master IP Core. The release date value can be read by sequential access to a three byte structure with arbitration rules to avoid possible conflicts between both interfaces "drive" and "safe 1" where the value can be accessed. To read the release date value the following steps are required:

- During one host interface transaction the VERSION register (see 5.3.9) is read first; this enables access to the release date register.
- The first returned byte from the release date register encodes the release year as binary number in excess of 2000 (10h = 2016, ... 63h = 2099).
- The second returned byte from the release date register encodes the release month as binary number (01h = January, ... 0Ch = December).
- The third returned byte from the release date register encodes the release day as binary number (01h = 1st, ... 1Fh = 31st).

Every reading to the RELEASE register after the third byte repeats the third returned byte. To reset the RELEASE register a new reading of the VERSION register is required. While one of the interfaces accesses the RELEASE register ("drive" or "safe 1") the other returns 00h. When using the Basic Interface (see chapter 9.4) the access right to the RELEASE register is released when the freeze signal ("hostx_f") is lowered.

NOTE
Access from Safe 2 Interface is not possible.

6.3.11 Encoder ID

The `ENC_ID` registers contain the designation code of the motor feedback system connected to the DSL Master. In the current protocol specification, the designation code is 20 bits long. With later enhancements, the free bits in the encoder ID registers are used to indicate special characters.

These registers are write protected.

Register 0Dh: **Encoder ID, byte 2**

X-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	SCI			ENC_ID19:16			
Bit 23							Bit 16

Register 0Eh: **Encoder ID, byte 1**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
ENC_ID15:8							
Bit 15							Bit 8

Register 0Fh: **Encoder ID, byte 0**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
ENC_ID7:0							
Bit 7							Bit 0

Bit 23 **Not implemented**: Read as "0".

Bit 22-20 **SCI**: Indication of special characters

3 bit special character for later enhancements of the encoder designation code. Not allocated.

Bit 19-0 **ENC_ID**: Encoder designation code

Designation of the motor feedback system (length: 20 bits)

The individual **ENC_ID** register bits are allocated as follows:

Bit 19 **Continue**: In High status, **Continue** indicates that ENC_ID is longer than 20 bits (for future use).

Bit 18 to 16 **Reserved**: Read as "0".

Bit 15 to 12 **User defined encoder index**: 4 bit value (0 to 15) for user- defined encoder index (see chapter 8.8.7).

Bit 11 **Reserved**: Read as "0".

Bit 10 **Sign**: In High status, **Sign** indicates that the position value is signed, in Low status, **Sign** indicates that the position value is not signed.

Bit 9 to 4 **#Pos-#Acc**: Length of position information (standard value: 40 bits) minus length of the acceleration value transmitted (see chapter 6.3.12, standard value: 11 bits).

Bit 3 to 0 **#Acc-8**: Length of the acceleration value transmitted (standard value: 11 bits) minus 8.

6.3.12 Fast position

The `POS` registers for the fast position contain the value of the motor feedback system connected. This position is generated incrementally from the safe position at start-up and is updated with every protocol frame.

After every eight protocol frames, the fast position is checked against the safe position (see under registers 18h to 1Ch).

The position sampling point is determined by the **ES** value of the synchronization control register.

Only those `POS` bits are activated that lie within the range that the motor feedback system has actually measured. All other higher value bits are read as "0". The number of measurable bits can be taken from **ENC_ID** bits 9 to 0 in the `ENC_ID0 to 2` registers.

If **Sign** is set in the **ENC_ID** register, the value of the fast position is given signed in the two's complement.

The units of the position value are (steps). These registers are write protected.

CAUTION
The fast position must not be used for safety functions.

Register 10h: **Fast position, byte 4**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Fast position, byte 4							
Bit 39							Bit 32

Register 11h: **Fast position, byte 3**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Fast position, byte 3							
Bit 31							Bit 24

Register 12h: **Fast position, byte 2**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Fast position, byte 2							
Bit 23							Bit 16

Register 13h: **Fast position, byte 1**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Fast position, byte 1							
Bit 15							Bit 8

Register 14h: **Fast position, byte 0**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Fast position, byte 0							
Bit 7							Bit 0

Bit 39-0 **Fast position, byte 4/3/2/1/0**:

Position value of the motor feedback system (length: 40 bits), incrementally generated.

6.3.13 Speed

The `VEL` speed registers contain the speed values of the connected motor feedback system. This value is calculated as a Δ position from the acceleration value (ΔΔposition) transmitted on the process data channel and the currently updated protocol frame (see chapter 3).

The speed sampling point is determined by the **ES** value of the `SYNC_CTRL` register. The units of the speed value are (steps/frame cycle time).

These registers are write protected.

⚠ **CAUTION**
The speed value must not be used for safety functions.

Register 15h: **Speed, byte 2**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Speed, byte 2							
Bit 23							Bit 16

Register 16h: **Speed, byte 1**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Speed, byte 1							
Bit 15							Bit 8

Register 17h: **Speed, byte 0**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Speed, byte 0							
Bit 7							Bit 0

Bit 23-0 **Speed, byte 2/1/0:**

Speed of the motor feedback system (length: 24 bits)

6.3.14 Status summary

The SUMMARY status summary register contains the summarized DSL Slave status information. Each status summary bit contains the summarized information from 8 error, warning and event modes of the DSL Slave. The bits in the status summary register can be read only. figure 15 shows the relationship between the encoder status registers, the status summary register and the **SUM** bit in the EVENT registers.

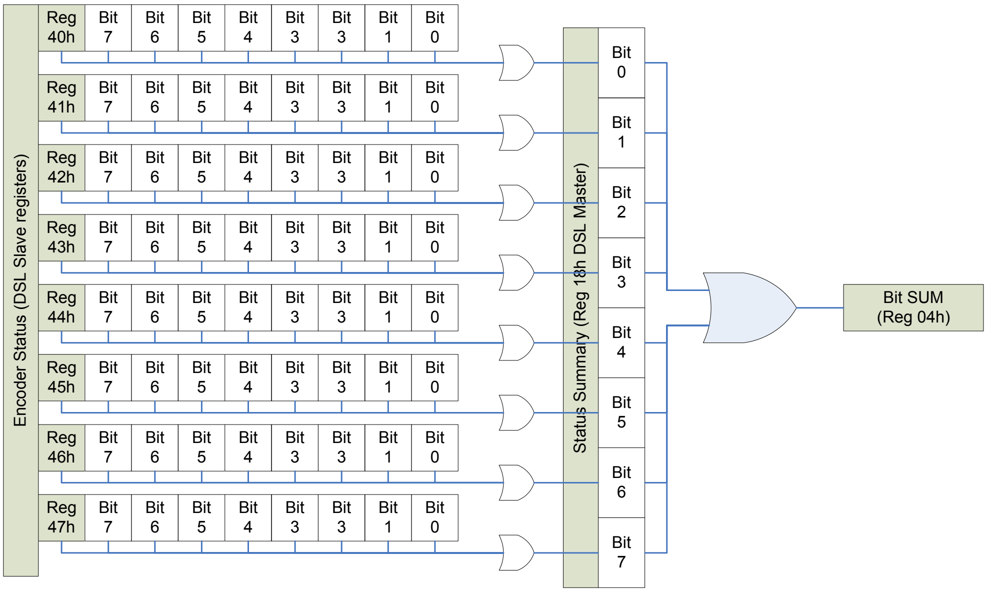


Figure 15: DSL Slave status and summary

A bit that has been set in the SUMMARY register is not automatically deleted by the DSL System. To delete, the frequency inverter application must read the corresponding DSL Slave encoder status register (see chapter 6.4.1) and acknowledge the status message, by individually deleting each set bit.

Register 18h: **Status summary**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SUM7	SUM6	SUM5	SUM4	SUM3	SUM2	SUM1	SUM0
Bit 7							Bit 0

Bit 7-1 **SUM7:SUM1**: Status summary bit (external resource)

1 = An error, a warning or an event associated with DSL Slave external resources was triggered.

0 = The corresponding error, warning or event is not active.

Bit 0 **SUMO**: Status summary bit (interface)

1 = An error, a warning or an event associated with the DSL Slave protocol interface was triggered.

0 = The DSL Slave protocol has not triggered an error, warning or event.

NOTE
A bit that has been set in summary register is not automatically deleted by the DSL System. To delete, the user application must read the corresponding DSL Slave encoder status register and acknowledge the status message, by individually deleting each set bit.

6.3.15 Safe position

The value for each safe position transmitted is compared with the incrementally generated position value (fast position) (see registers 10h to 14h).

The safe position is not synchronized with the `sync` signal.

Only those `vpos` bits are activated that lie within the range that the motor feedback system has actually measured. All other higher value bits are read as "0". The number of measurable bits can be taken from **ENC_ID** bits 9 to 0 in the `ENC_ID0 to 2` registers.

If **Sign** is set in the **ENC_ID** register, the value of the safe position is given signed in the two's complement.

The units of the position value are [steps]. These registers are write protected.

Register 19h: **Safe position, byte 4**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Safe position, byte 4							
Bit 39							Bit 32

Register 1Ah: **Safe position, byte 3**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Safe position, byte 3							
Bit 31							Bit 24

Register 1Bh: **Safe position, byte 2**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Safe position, byte 2							
Bit 23							Bit 16

Register 1Ch: **Safe position, byte 1**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Safe position, byte 1							
Bit 15							Bit 8

Register 1Dh: **Safe position, byte 0**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Safe position, byte 0							

Bit 7							Bit 0

Bit 39-0 **Safe position, byte 4/3/2/1/0:**

Position value transmitted through Safe Channel 1 (length: 40 bits), absolute value.

6.3.16 Position checksum

The POSCRC registers for the position checksum contain the CRC checksum of the safe position VPOS (see chapter 6.3.15) and the SUMMARY status summary (see chapter 6.3.14).

The CRC is checked in the DSL Master IP Core. These registers can be checked with an external cross check by the drive application.

The CRC is generated with the following CRC parameters:

*Table 23: **POSCRC parameters***

Parameter	Value
CRC sequence	16 Bit
CRC polynomial	C86Ch ($x^{16} + x^{15} + x^{12} + x^7 + x^6 + x^4 + x^3 + 1$) Normal representation: 90D9h
Starting value	0000h
Closing XOR value	00FFh
Reverse data bytes	No
Reverse CRC before closing XOR	No

The sequence of the bytes to calculate the CRC is shown in the following figure:

SAFE_SUM	VPOS[39:32]	VPOS[31:24]	VPOS[23:16]	VPOS[15:8]	VPOS[7:0]

Figure 16: Sequence of the bytes to calculate the CRC

Register 1Eh: **CRC of the safe position, byte 1**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
CRC of the safe position, byte 1							
Bit 15							Bit 8

Register 1Fh: **CRC of the safe position, byte 0**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
CRC of the safe position, byte 0							
Bit 7							Bit 0

Bit 15-0 **CRC of the safe position:**

16 bit CRC checksum (CRC 16) of the safe position and status summary in Safe Channel 1.

6.3.17 Parameters Channel buffer

The eight PC_BUFFER registers of the Parameters Channel buffer contain the answer to the last "long message" request or the data for a "long message" write operation.

NOTE
Access to these registers may only take place if the "long message" channel is free (**FREL** in the EVENT_L register).

Depending on the length of the "long message" answer, the registers are used as follows:

Table 24: Data length of the "long message"

Length of the "long message"	Register used
8 bytes	20h to 27h
4 bytes	20h to 23h
2 bytes	20h to 21h
0 bytes	None

These registers are also for the reporting of error conditions arising from a "long message" operation. If, when accessing a resource, an error due to a "long message" arises (e.g. invalid data, error in the A/D conversion, time overrun), after the answering message has been received the **LOFF** bit in the `PC_ADD_H` register (28h) is set. In this case the Parameters Channel buffer bytes 0 and 1 contain an error code.

The meaning of the error code depends on the particular HIPERFACE DSL® encoder and is described in detail in the data sheet.

Register 20h: Parameters Channel buffer, byte 0

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Parameters Channel, byte 0							
Bit 63							Bit 56

Register 21h: Parameters Channel buffer, byte 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Parameters Channel, byte 1							
Bit 55							Bit 48

Register 22h: Parameters Channel buffer, byte 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Parameters Channel, byte 2							
Bit 47							Bit 40

Register 23h: Parameters Channel buffer, byte 3

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Parameters Channel, byte 3							
Bit 39							Bit 32

Register 24h: Parameters Channel buffer, byte 4

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Parameters Channel, byte 4							
Bit 31							Bit 24

Register 25h: Parameters Channel buffer, byte 5

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Parameters Channel, byte 5							
Bit 23							Bit 16

Register 26h: Parameters Channel buffer, byte 6

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Parameters Channel, byte 6							
Bit 15							Bit 8

Register 27h: **Parameters Channel buffer, byte 7**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Parameters Channel, byte 7							
Bit 7							Bit 0

Bit 63-0 **Parameters Channel buffer, byte 0-7**:

8 bytes for the answer to a long message (read operation) or for a "long message" write operation.

Bit 63-48 **Error report for a long message, byte 0-1**:

2 bytes for reports about errors in encoder resources arising from the previous "long message" operation.

6.3.18 Long message address

The addresses and the addressing mode for "long messages" sent over the Parameters Channel are determined in the `PC_ADD_H/PC_ADD_L` long message address registers.

In addition, the long message address register 28h (`PC_ADD_H`) contains indications of errors arising from "long message" operations. For this sort of error, the Parameters Channel buffer contains the error code in bytes 0 and 1 associated with this status (see chapter 7.6.6).

Register 28h: **Long message address, byte 1**

R-1	W-0	R/W-0	W-0	W-0	W-0	W-0	W-0
LID	LRW	LOFF	LIND	LLEN		LADD9	LADD8
Bit 15							Bit 8

Register 29h: **Long message address, byte 0**

W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
LADD7:0							
Bit 7							Bit 0

Bit 15 **LID**: Long message identification. This is a read only bit that will always return ‘1’ when read.

Bit 14 **LRW**: Long message, read/write mode

1 = "long message" read operation

0 = "Long message" write operation

Bit 13 **LOFF**: Long message addressing mode/long message error Write access:

1 = Offset addressing of "long messages". The offset value from the `PC_OFF_H/PC_OFF_L` registers is used in the resource of the selected database entry as a sub-address.

0 = Addressing of "long messages" without offset. The offset value from the `PC_OFF_H/PC_OFF_L`registers is not used.

Read access:

1 = The last "long message" caused an error.

0 = The last "long message" was correctly processed.

Bit 12 **LIND**: Indirect addressing of long messages

1 = Indirect addressing of "long messages". During this operation, the stored address content in the given database entry is evaluated.

0 = Direct addressing of "long messages". The operation affects the database entry given in the current address.

Bit 11-10 **LLEN**: Data length of the "long message"

11 = 8 data bytes

10 = 4 data bytes

01 = 2 data bytes

00 = No data bytes

Bit 9-0 **LADD**: Long message address

Database entry with 10 bit address for a "long message" operation.

6.3.19 Long message address offset

The `PC_OFF_H/PC_OFF_L` address offset registers for long messages are used in "long message" operations, if **LOFF** is set in the register 28h. In this case the LOFFADD value from these registers is used to communicate with the sub-address of a multiple byte encoder resource.

Only write access is possible for these registers.

Register 2Ah: **Long message address offset, byte 1**

R-1	W-0	W-0	W-0	W-0	W-0	W-0	W-0
LID	LOFFADD14:8						
Bit 15							Bit 8

Register 2Bh: **Long message address offset, byte 0**

W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
LOFFADD7:0							
Bit 7							Bit 0

Bit 15 **LID**: Long message identification. The value must be "1".

Bit 14-0 **LOFFADD14:0**: Long message offset value

The 15 bit offset value of the "long message" address offset is stored in these bits.

6.3.20 Parameters Channel control

The `PC_CTRL` control register for the Parameters Channel handles the start of "long message" transactions. After setting all "long message" registers (registers `PC_BUFFER0 to 7`, `PC_ADD_H/PC_ADD_L` and `PC_OFF_H/L`), the "long message" is transmitted to the DSL Slave by setting the **LSTA** bit.

Register 2Ch: **Parameters Channel control**

X-0	X-0	X-0	X-0	X-0	X-0	X-0	W-0
							LSTA
Bit 7							Bit 0

Bit 7-1 **Not implemented**: Read as "0".

Bit 0 **LSTA**: Control of the long message start

1 = A "long message" transaction is started with the values currently stored in the "long message" registers.

0 = No effect.

6.3.21 SensorHub Channel status

The SensorHub Channel status register `PIPE_S` provides information about the current status of the SensorHub Channel (see chapter 3.5).

`PIPE_S` is only accessible as a register of the DSL Master if SPI-PIPE is deactivated (**SPPE** in the `SYS_CTRL` register is deleted).

Otherwise the value of `PIPE_S` is transmitted via SPI-PIPE as the first byte of each read request (see chapter 5.2). In this case, the first four bits are transmitted as "0101" to check the SPI-PIPE interface for errors due to unchanged values.

When this register is read, the current data from the FIFO buffer is read and stored in an intermediate register so that a subsequent read process in the `PIPE_D` register can be considered to be completed at the same time as the `PIPE_S` register is read. Using this mechanism, any deviation between status and data information in this instance will prevent new data entering the SensorHub Channel during access to the FIFO buffer.

PIPE_S is a write protected register.

Register 2Dh: **SensorHub Channel status**

X-0	X-0	X-0	X-0	R-0	R-0	R-0	R-0
				POVR	PEMP	PERR	PSCI
Bit 7							Bit 0

Bit 7-4 **Not implemented**: Read as "0".

Bit 3 **POVR**: SensorHub Channel overflow

1 = The capacity of the 8 byte FIFO buffer for SensorHub Channel data was exhausted and since the last read process, values have been discarded.

0 = The capacity of the FIFO buffer for SensorHub Channel data is not yet exhausted.

This bit is deleted after the read process.

Bit 2 **PEMP**: The SensorHub channel buffer is empty.

1 = A read request was issued, but the FIFO buffer for SensorHub Channel data is empty. In this case, `PIPE_D` contains the value 00h.

0 = No "buffer empty" error.

This bit is updated after every access to the FIFO buffer.

Bit 1 **PERR**: Coding error of the bits in the SensorHub Channel.

1 = The bit level coding of the data currently in the SensorHub Channel is erroneous.

0 = No error in bit coding.

This bit is stored together with the pipeline data byte in question in the FIFO buffer.

Bit 0 **PSCI**: Indication for special characters in the SensorHub Channel.

1 = A special character was received in the SensorHub Channel.

0 = Indication for "no special character".

This bit is stored together with the pipeline data byte in question in the FIFO buffer.

Special characters are normally used as data separators or to indicate special events. To obtain information about which special character was received, the PIPE_D register must be read. All 8b10b special characters can be used on the SensorHub channel. An exception is the "K30.7" symbol that

is used in HIPERFACE DSL® to indicate "no data" and is not stored in the FIFO buffer.

table 25 below contains the supported 8b10b special characters.

Table 25: 8b10b special characters supported in the SensorHub Channel

Special characters	Coding in register PIPE_D
K28.0	1Ch
K28.1	3Ch
K28.2	5Ch
K28.3	7Ch
K28.4	9Ch
K28.5	BCh
K28.6	DCh
K28.7	FCh
K23.7	F7h
K27.7	FBh
K29.7	FDh

6.3.22 SensorHub Channel data

The PIPE_D SensorHub Channel data register contains the SensorHub Channel data that is stored in an 8 byte FIFO buffer.

If new data arrives at the buffer when it is full, before PIPE_D is read, the oldest value is discarded and the **POVR** bit in PIPE_S is set.

If a read request is issued when the buffer is empty, the **PEMP** bit in PIPE_S is set and the value 00h is transmitted.

PIPE_D is only accessible as a register of the DSL Master if SPI-PIPE is deactivated (**SPPE** in the SYS_CTRLregister is deleted).

Otherwise the value of PIPE_D is transmitted via SPI-PIPE as the second byte of each read request (see chapter 5.2).

At the moment that the PIPE_S register is accessed, the corresponding PIPE_D value is frozen to guarantee synchronization between status and data information.

PIPE_Dis a write protected register.

Register 2Eh: **Sensor Hub Channel data**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SensorHub Channel data							
Bit 7							Bit 0

Bit 7-0 **SensorHub Channel data**

8 bit value of the FIFO buffer for SensorHub Channel data.

6.3.23 Parameters Channel short message

The PC_DATA register for the Parameters Channel short message contains the results of "short message" transactions.

"Short message" transactions are generated if operations are carried out with remote registers (DSL Slave). Generally, **FRES** (in the `EVENT_S` register) must be set after a transaction is started. Only then will `PC_DATA` contain valid information.

Register 2Fh: **"Short message" Parameters Channel data**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
"Short message" Parameters Channel data							
Bit 7							Bit 0

Bit 7-0 **"Short message" Parameters Channel data**

8 bit value of the requested remote register.

6.3.24 Fast position error counter

The `ACC_ERR_CNT` register returns the count of transmitted fast position values with consecutive transmission errors. The value is clamped to a maximum of 31 (1Fh).

With a writing access the error threshold for the test signal `acc_thr_err` can be set. This value is also clamped to a maximum of 31 (1Fh). If the count of transmitted fast position values with consecutive transmission errors exceeds this threshold `acc_thr_err` will be set to ‘1’.

Register 38h: **Fast position error counter**

X-0	X-0	X-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	-	-	CNT4	CNT3	CNT2	CNT1	CNT0
Bit 7							Bit 0

Bit 7-5 **Not implemented**: Read as "0".

Bit 4-0 **CNT4:CNT0**: Position error count/threshold for `acc_thr_err`

Read: 5 bit value of count of transmitted fast position values with consecutive transmission errors

Write: 5 bit value for threshold of `acc_thr_err`

6.3.25 Fast position acceleration boundary

The `MAXACC` register allows setting an acceleration threshold for a given application. This threshold is used by the fast position estimator to clamp the acceleration of the estimated position during communication or sensor failures of the fast position channel.

Register 39h: **Fast position acceleration boundary**

W-1	W-1	W-1	W-1	W-1	W-1	W-1	W-1
RES1	RES0	MNT5	MNT4	MNT3	MNT2	MNT1	MNT0
Bit 7							Bit 0

Bit 7-6 **RES1:RES0**: Resolution of fast position acceleration boundary

Bit 5-0 **MNT5:MNT0**: Mantissa of fast position acceleration boundary

6.3.26 Fast position estimator deviation

The `MAXDEV` registers return the maximum absolute position deviation while the position estimator is active. The returned 16 bit value has the same format (resolution) as the fast position channel and is clamped to a maximum of 65535 steps (0xFFFF). The registers are set to the maximum value 0xFFFF at reset.

These registers also allow setting a deviation threshold value for triggering the output signal `dev_thr_err` (see chapter 5.4.3). The threshold value can be written with the same format as the deviation (unsigned 16 bit, same resolution as the fast position channel).

Register 3Ah: **Fast position estimator deviation high byte**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DEV15	DEV14	DEV13	DEV12	DEV11	DEV10	DEV09	DEV08
Bit 15							Bit 8

Register 3Bh: **Fast position estimator deviation low byte**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DEV07	DEV06	DEV05	DEV04	DEV03	DEV02	DEV01	DEV00
Bit 7							Bit 0

Bit 15-0 **DEV15:DEV00**: Position deviation/Deviation threshold

Read: 16 bit value of position deviation

Write: 16 bit value for deviation threshold for dev_thr_err

6.4 Function register for the DSL Slave

The remote registers of the DSL Slave (encoder) are mirrored in the DSL Master under the addresses 40h to 7Fh. These registers are accessible using "short message" transactions (see chapter 7.5.1).

NOTE
It should be noted that the DSL Slave register can only be accessed via 8 bit addressing. The `bigend` option does not affect the Slave register address allocation.

The minimum number of remote registers present in the DSL Slave is set out in table 26. For real DSL Slave installations, more remote registers can be installed than are set out in the table.

Table 26: Remote slave register

Address	Designation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value at reset
40h	ENC_ST0	ST07	ST06	ST05	ST04	ST03	ST02	ST01	ST00	0000 0000
41h	ENC_ST1	ST17	ST16	ST15	ST14	ST13	ST12	ST11	ST10	0000 0000
42h	ENC_ST2	ST27	ST26	ST25	ST24	ST23	ST22	ST21	ST20	0000 0000
43h	ENC_ST3	ST37	ST36	ST35	ST34	ST33	ST32	ST31	ST30	0000 0000
44h	ENC_ST4	ST47	ST46	ST45	ST44	ST43	ST42	ST41	ST40	0000 0000
45h	ENC_ST5	ST57	ST56	ST55	ST54	ST53	ST52	ST51	ST50	0000 0000
46h	ENC_ST6	ST67	ST66	ST65	ST64	ST63	ST62	ST61	ST60	0000 0000
47h	ENC_ST7	ST77	ST76	ST75	ST74	ST73	ST72	ST71	ST70	0000 0000
7Ch	SRSSI	-	-	-	-	-	SRSSI2:0			---- -000

Address	Designation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value at reset
7Eh	MAIL	Slave-Mail								0000 0000
7Fh	PING	Slave-Ping[1]								0000 0000

[1] After a protocol reset, the PING register contains the slave interface version (see chapter 6.4).

6.4.1 Encoder status

The ENC_ST encoder status registers contain all slave system errors, events and warnings from the DSL encoder.

The allocation between the individual bits and the slave system statuses is determined when the DSL Slave is installed and set out in the associated data sheet. The general application of the status register follows the list in chapter 7.6.3.

NOTE
It should be noted that all bits of an encoder status register are OR linked and mirror bits in the SAFE_SUM DSL Master register (36h) (see figure 14). In this way the appropriate groups can react rapidly to slave statuses.

NOTE
Bits in the encoder status register can only be set by the DSL Slave and only deleted by the frequency inverter application (acknowledgment).

Table 27: Encoder status and summary register

Encoder status	SAFE_SUM bit (DSL Master 36h)
ENC_ST0 (40h)	SUM0
ENC_ST1 (41h)	SUM1
ENC_ST2 (42h)	SUM2
ENC_ST3 (43h)	SUM3
ENC_ST4 (44h)	SUM4
ENC_ST5 (45h)	SUM5
ENC_ST6 (46h)	SUM6
ENC_ST7 (47h)	SUM7

Register 40h: **Encoder status, byte 0**

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
Encoder status							
Bit 7							Bit 0

Register 41h: **Encoder status, byte 1**

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
Encoder status							
Bit 15							Bit 8

Register 42h: **Encoder status, byte 2**

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
Encoder status							
Bit 23							Bit 16

Register 43h: Encoder status, **byte 3**

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
Encoder status							
Bit 31							Bit 24

Register 44h: **Encoder status, byte 4**

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
Encoder status							
Bit 39							Bit 32

Register 45h: **Encoder status, byte 5**

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
Encoder status							
Bit 47							Bit 40

Register 46h: **Encoder status, byte 6**

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
Encoder status							
Bit 55							Bit 48

Register 47h: Encoder status, **byte 7**

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
Encoder status							
Bit 63							Bit 56

Bit 63-0 **Encoder status**

The individual bits indicate different errors, events and warnings. The meaning of each individual bit is determined by the particular DSL Slave installation. Generally the specification in chapter 7.6.3 applies.

1 = Error, event or warning status.

0 = Encoder in normal status.

6.4.2 Slave RSSI

The `SRSSI` register for indicating the received signal strength at the slave (Slave Received Signal Strength Indicator, RSSI) provides an indication of the strength of the signal arriving at the slave.

The value of the register is only updated from frame to frame if the measurement result deteriorates. After a read access to this register, the register is reset to the value "7" (maximum signal strength).

The register is write protected.

Register 7Ch: **Slave RSSI**

X-0	X-0	X-0	X-0	X-0	R-0	R-0	R-0
					SRSSI		
Bit 7							Bit 0

Bit 7-3 **Not implemented:** Read as "0".

Bit 2-0 **Value of the Slave RSSI**

The values for the Slave RSSI range from "0" (poorest signal strength) to "7" (best signal strength).

6.4.3 Slave-Mail

The `MAIL` multi-purpose register of the slave is used for fast communication with the DSL motor feedback system processor. The content of the slave mail register is transmitted to the encoder processor by the most rapid route possible.

Register 7Eh: **Slave Mail**

W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
Slave-Mail							
Bit 7							Bit 0

Bit 7-0 **Slave-Mail**

8 bit slave mail data for multiple applications.

6.4.4 Slave-Ping

The `PING` register of the slave is used to carry out connection tests on behalf of the DSL Slave. The register can be written to and read externally, without this affecting the DSL interface.

On start-up, the register is initialized with the DSL Slave interface hardware version.

NOTE
This register can also be used for multi axis implementation. For more information please see chapter .

Register 7Fh: **Slave-Ping**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Slave-Ping							
Bit 7							Bit 0

Bit 7-0 **Slave-Ping**

8 bit hardware version of the DSL encoder at start-up. On first reading, the Ping value of the slave can be used as a connection test on behalf of the slave.

7 Central functions

In this chapter, access to the central sensor functionality via interfaces and registers is described.

7.1 System start

As soon as the motor feedback system is supplied with power, a forced reset ensures that a defined system start status is produced in the DSL Master IP Core.

figure 17 shows the status table for system start.

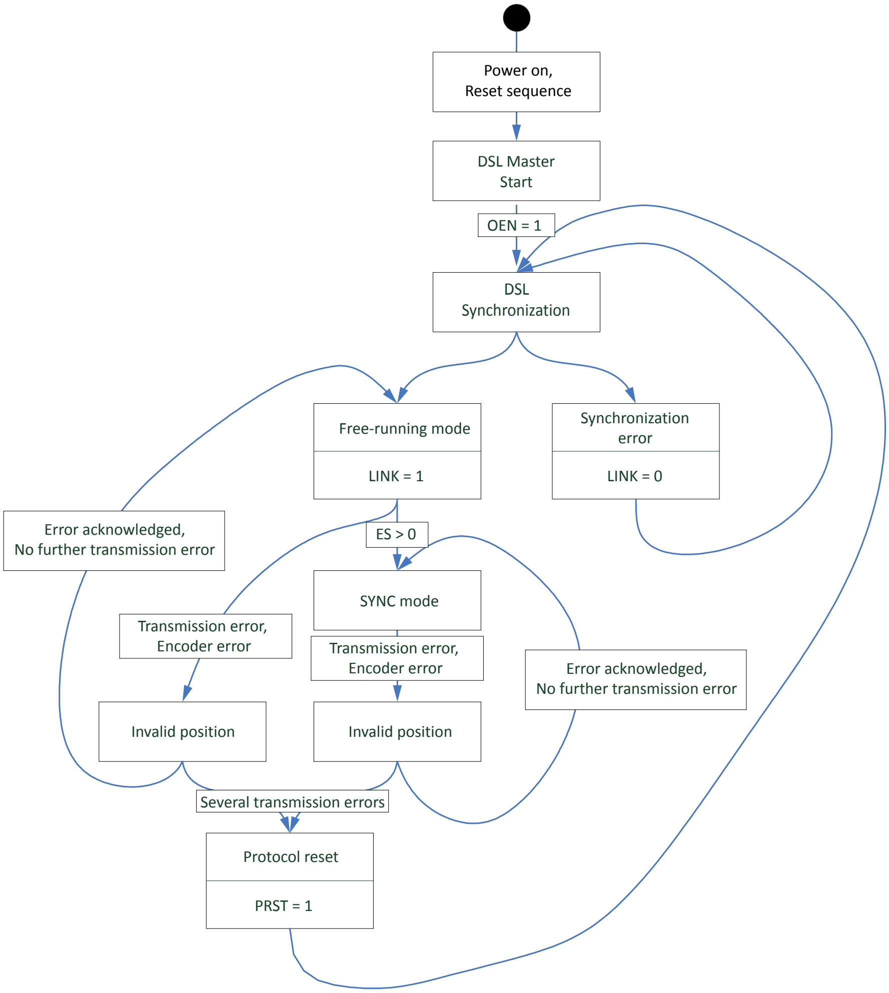


Figure 17: Status table for DSL system start.

Individual conditions are described in table 28.

Table 28: Conditions at DSL system start

Status	Prerequisite	Indication
DSL Master start	Switching on supply voltage. Reset process (Duration: 500 ms)	Communication via drive interface is possible
DSL synchronization	**OEN** = 1 (**SYS_CTRL** register)	None

Status	Prerequisite	Indication
Synchronization error	Time overrun (typ. after 5 ms) during the DSL synchronization	**LINK** = 0 (**MASTER_QM** register)
Free running mode	Successful DSL synchronization	**LINK** = 1 (**MASTER_QM** register)
SYNC mode	**ES** > 0 (**SYNC_CTRL** register), Cyclic signal to SYNC input	Synchronous encoder position in the POS0 to POS4 registers
Invalid position	External transmission or encoder error	Error bit set in **EVENT_H**, **EVENT_L** or in Online Status
Protocol reset	Two successive transmission errors	**PRST** = 1 (**EVENT_H** register or Online Status)

7.2 System diagnostics

HIPERFACE DSL® provides comprehensive system diagnostics in relation to communications quality both during the development of a DSL system as well as during normal operation.

7.2.1 System diagnostics during development

During the development of a DSL system, several registers are involved in the diagnostics of correct use and operation. These include:

- Quality monitoring `MASTER_QM`
- Edge register `EDGES`
- Run time register `DELAY`

After the DSL connection has been activated (**OEN** bit, see chapter 6.3.1), the **LINK** flag in the `MASTER_QM` register must be checked for the set value "1". This indicates that the connection to the motor feedback system was successfully established.

If this bit remains deleted for longer than the expected start-up time (see encoder datasheet), there is a fundamental problem in the connection between the frequency inverter and the motor feedback system.

Check whether the encoder is supplied with power.

Using an oscilloscope, also check whether any level changes in the transmission frequency range can be identified over the data cables between the frequency inverter and the encoder.

Using the run time register (see chapter 6.3.8), it is possible to identify whether the DSL signal **cable delay** complies with the specification. The run time is mainly a result of the length of the cable between the frequency inverter and the motor

feedback system. In addition, the selection of the interface drive (RS485 transceiver) has an effect on the signal run time.

The value of the `EDGES` register (see chapter 6.3.7) indicates how well or badly the DSL Master can sample the communication signal coming from the motor feedback system.

Start the check of the **bit sampling pattern** with the motor switched off. If several bits have been set in the sampling pattern (more than four), the encoder shielding design should be checked. The aim should be that, during interference-free operation, the minimum number of bits is set in the sampling pattern.

In the second step, check the sampling pattern with the motor switched on, preferably in the target application. In such cases a maximum of seven bits may be set in the `EDGES` register.

CAUTION
If under certain circumstances, however, eight bits are set in the `EDGES` register, the operation of the DSL motor feedback system cannot be guaranteed.

7.2.2 System diagnostics during operation

When operating the DSL system, system diagnostics are indicated in the following registers:

- Run time register `DELAY`
- Quality monitoring `MASTER_QM`
- Indication of the received signal strength at the `SRSSI` slave

The run time register (see chapter 6.3.8) contains the **RSSI** value that lies in the range between "0" and "12". The register indicates the quality of the connection during operation with regard to the signal strength.

The quality monitoring (see chapter 6.3.3) contains the **QM** value that lies in the range between "0" and "15". **QM** indicates the quality of the connection during operation with regard to transmission errors.

For continuous monitoring of the connection quality it is recommended that these two values are polled cyclically.

Event-oriented monitoring is also possible. For this, the event bits **QMLW** and **PRST** must be polled. These bits indicate **QM** sinking below a value of "14" (poor quality) or a broken connection if **QM** has a value of "0" or **RSSI** has a maximum value of "1". The following table contains the possible conditions:

*Table 29: **Values for quality monitoring and RSSI***

Quality monitoring value	QMLW	PRST	Connection status	LINK	
15	0	0	Good connection quality	1	🟢
14 to 1	1	0	Poor connection quality	0	🟡
0	1	1	Connection broken	0	🔴

Frequent errors can indicate that the shielding design of the DSL connection is inadequate or that the cable does not comply with the specification.

The slave RSSI register (see chapter 6.4.2) contains the **SRSSI** value that lies in the range between "0" and "7". **SRSSI** indicates the quality of the run time connection as the signal strength of the data transmitted to the DSL encoder.

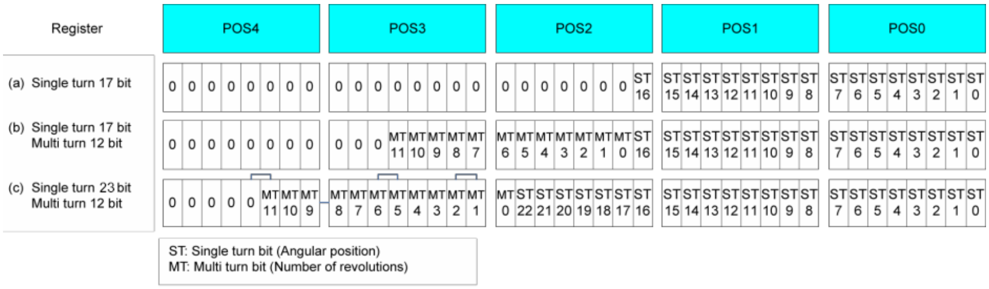
7.3 Fast position

The fast position and the rotation speed of the encoder shaft are transmitted on the DSL motor feedback system process data channel. These values are the main process values for the drive application control circuit.

HIPERFACE DSL® stores the fast position in the `POS0…4` DSL Master registers and the rotation speed in the `VEL0…2` registers.

The position is given as a 40 bit value that includes the angle setting ("single turn" value) and the number of rotations ("multi-turn" value). Only the position bits actually measured by the motor feedback system are accessible and are stored in the registers as a right-justified value. The other (higher value) bits are constantly set at "0" (see examples "a" to "c" in figure 18).

The fast position is automatically aligned to the current safe position of the motor feedback system. This mechanism is automatically checked by the DSL Master. For this purpose, the DSL Master compares the fast with the safe position (see chapter 7.4).



Register	POS4	POS3	POS2	POS1	POS0
(a) Single turn 17 bit	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 ST 16	ST 15 ST 14 ST 13 ST 12 ST 11 ST 10 ST 9 ST 8	ST 7 ST 6 ST 5 ST 4 ST 3 ST 2 ST 1 ST 0
(b) Single turn 17 bit Multi turn 12 bit	0 0 0 0 0 0 0 0	0 0 0 MT 11 MT 10 MT 9 MT 8 MT 7	MT 6 MT 5 MT 4 MT 3 MT 2 MT 1 MT 0 ST 16	ST 15 ST 14 ST 13 ST 12 ST 11 ST 10 ST 9 ST 8	ST 7 ST 6 ST 5 ST 4 ST 3 ST 2 ST 1 ST 0
(c) Single turn 23 bit Multi turn 12 bit	0 0 0 0 0 MT 11 MT 10 MT 9	MT 8 MT 7 MT 6 MT 5 MT 4 MT 3 MT 2 MT 1	MT 0 ST 22 ST 21 ST 20 ST 19 ST 18 ST 17 ST 16	ST 15 ST 14 ST 13 ST 12 ST 11 ST 10 ST 9 ST 8	ST 7 ST 6 ST 5 ST 4 ST 3 ST 2 ST 1 ST 0

ST: Single turn bit (Angular position)
MT: Multi turn bit (Number of revolutions)

Figure 18: Position value format

The motor feedback system fast position is sampled and transmitted if the DSL Master receives a SYNC signal. This SYNC signal can be created in two different ways (see chapter 7.3.2 and chapter 7.3.3).

If the encoder detects faults in the fast position sensor or if a transmission fault of the fast position value occurs, the fast position registers `POS0…4` and the rotation speed registers `VEL0…2` are automatically loaded with estimator values to allow for a ride-through of non-permanent fault conditions. This state is indicated by a non-zero value in the `MAXDEV` registers and a raised `estimator_on` signal and POS flag.

7.3.1 Estimator

The estimator is implemented for providing an estimated fast position when the regular process provides wrong or no position at all.

The estimator is turned on because of the following reasons:

- While operating under harsh conditions, sometimes the encoder position cannot be sampled correctly (e. g. mechanical shock). In these cases the encoder will transmit a “position not valid” character instead of a valid position.
- The fast position CRC or the fast position encoding received by the DSL master is wrong (e. g. EMC problems).
- Interrupted link to encoder (missing acceleration data)

In these cases the value in the POS4 … POS0 registers is supplied by an estimator calculation instead of a real transmitted position.

NOTE

Whenever the estimator is active the `estimator_on` signal as well as the POS flags are set to “1”. Therefore it is highly recommended to monitor at least one of these flags during operation.

As there is no limit how long the estimator is running, the user application has to decide how long the estimated values are accepted. There are two use cases foreseen for the estimator.

In one case the user measures the time of the `estimator_on` signal or POS flag being raised to “1”. This is the easiest approach to use the estimator and recommended for most applications. No further setup is required.

In the other case the user must define the maximum deviation accepted along with the maximum acceleration of the system. Whenever the estimator uncertainty exceeds the maximum deviation, the `dev_thr_err` signal will be raised to "1". The remainder of this chapter will provide the necessary information to set up these values.

To set the maximum position deviation allowed for the estimator calculation the `MAXDEV` registers are used (see chapter 6.3.26). The resolution of this value is the same as the resolution of the fast position.

The maximum acceleration needs to be calculated for each individual application. Some margin should also be given to the maximum limit in order to account for noise, inaccuracy and so on.

The maximum absolute value of the acceleration must be written to the `MAXACC` register in a floating point format (see chapter 6.3.25). The resolution is stored in bits 7 and 6, and a positive integer mantissa on the remaining part of the register. The resulting value can be calculated as follows:

$$|acc_{\max}| = mantissa \cdot resolution$$

Here, mantissa is the value stored in bits 5...0 and resolution is defined as per the following table:

Table 30: Resolution of fast position acceleration boundary

Bit 7,6	Resolution
00	256
01	64
10	16
11	4

The ACC_{LSB} value is the resolution of the DSL fast position channel which can be calculated as:

$$ACC_{LSB} = \frac{2\pi}{2^{Pres} \cdot T^2_{hframe}} [rad/s^2]$$

with Pres as position resolution per turn in number of bits and T_{hframe} as DSL frame duration in s (see chapter 3)

Example:

If the DSL frame lasts 15.625 µs, an 18 bit resolution encoder is used, and a limit of 10.000 rad/s^2 is foreseen:

$$ACC_{LSB} = \frac{2\pi}{2^{18} \cdot (15.625 \cdot 10^{-6})^2} = 98174.77\ rad/s^2$$

The finest possible resolution for ACC_{max} is ACC_{LSB}/256 = 383.5 rad/s^2. Accordingly, ACC_{max} can be set to

$$acc_{max} = 27 \cdot 383.5\ rad/s^2 = 10355\ rad/s^2$$

with a setting of 0x1B in the `MAXACC` register.

Values greater than 24.000 rad /s^2 can be achieved by using different resolutions: on the same system as above, a limit of 30.000 rad /s^2 can be set using ACC_{LSB}/64 = 1534 rad /s^2.

Therefore:

$$acc_{max} = 20 \cdot 1534\ rad/s^2 = 30680\ rad/s^2$$

In this case the data to be written to the `MAXACC` register would be 0x54.

NOTE
When the estimator is turned on and MAXACC and MAXDEV are not configured, dev_thr_err signal and DTE flag are set to "1".

7.3.2 Free running mode

In free running mode, the SYNC signal is automatically created by the DSL Master, for which the maximum frame transmission frequency is used (see chapter 4.2.1). The free running mode is the standard DSL Master operating mode at start-up.

This operating mode can also be selected manually, by setting the **ES** value in the `SYNC_CTRL` register to "0".

NOTE
It should be noted that in free running mode, no account is taken of the signals at the SYNC input.
The polling of the position and rotation speed values is explained in figure 19 and figure 20.

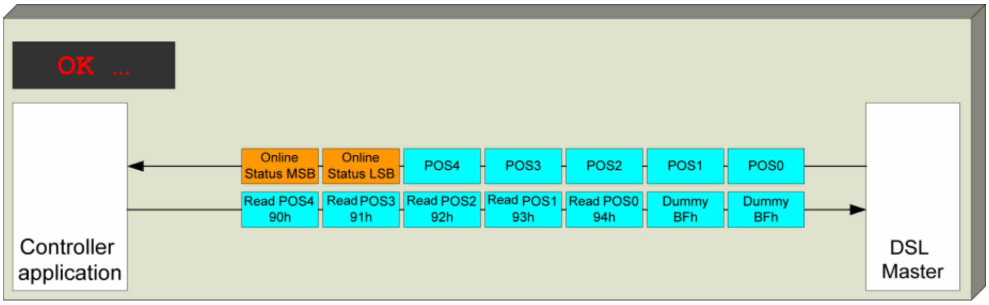


Figure 19: Polling of position registers in free running mode

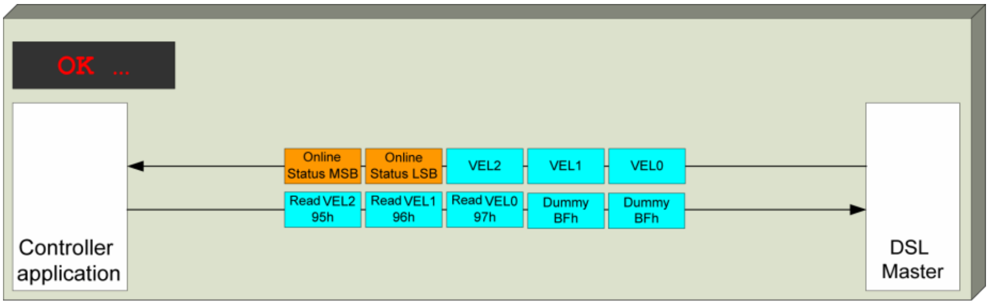


Figure 20: Polling of rotation speed registers in free running mode

7.3.3 SYNC mode

In SYNC mode, the DSL Master depends on a prepared cyclic control signal. This control signal triggers position measurements and enables polling of position and rotation speed values synchronously with the control signal. The control signal must be applied to the SYNC input and have the characteristics prescribed for the DSL Master (see chapter 4.2).

The position is available after a set delay in relation to the leading edge of the control signal.

When SYNC mode is used the following points must be noted:

1 A correct control signal must be applied at the `sync` input. The signal must correspond with the specifications for pulse width and cycle time.

2 Setting or deleting the **SPOL** bit in the `SYS_CTRL` register determines whether the position measurements are to be triggered by the leading of the trailing edge of the control signal. The set latency of the DSL system is measured from this edge.

3 The correct **ES** divider must be set in the `SYNC_CTRL` register. This divider determines how many position samplings and transmissions will be undertaken for each control signal. Do not change setting of ES divider during start up or operation.

CAUTION
The ES divider must be selected so that the cycle time between the two position samplings corresponds to the prescribed range limits (package cycle time) in see table 1.

The range limits for the **ES** divider can be calculated as follows:

$\mathbf{ES} \leq t_{Sync} / t_{Min}$

$\mathbf{ES} \geq t_{Sync} / t_{Max}$

The symbols used in the formulae are explained as follows:

Table 31: Symbols used in the formulae

Symbol	Description
tSync	Cycle time of the pulse signal at the SYNC input
tMin	Minimum cycle time for the transmission of DSL frames (11.95 µs)
tMax	Maximum cycle time for the transmission of DSL frames (27,00 µs)

table 32 below contains typical cycle times for the control signal and the valid ranges of **ES** divider values.

Table 32: Cycle times for SYNC signals and valid ES values

Frequency of the SYNC signal (kHz)	Cycle time of the SYNC signal (µs)	Minimum value ES	Maximum value ES
2	500	19	41
4	250	10	20
6.25	160	6	13
8	125	5	10
16	62.5	3	5
40	25	1	2
38 to 82	26.3 to 12.2	1	1

After the sequence described above, SYNC mode is activated. In the specified "start-up time” the protocol is synchronized with the applied `sync` signal. Following this period, the position value is available with constant latency after the data package has been transmitted (see figure 21).

The time profile of the relevant signals in SYNC mode is shown in the following graphic. This shows the `sync` signal, the `cycle` signal generated from the **ES** divider and the `dsl_out` DSL output signal.

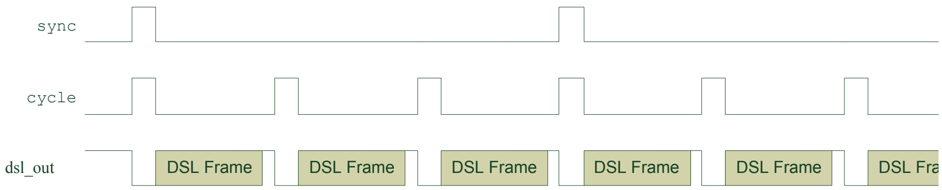


Figure 21: SYNC mode signals

The arrival of a requested fast position is indicated by the POSTX Online Status bits of drive interface (see chapter 6.2).

The position value can be polled via drive interface from the `POS0 to POS4` registers of the DSL Master (see chapter 6.3.12).

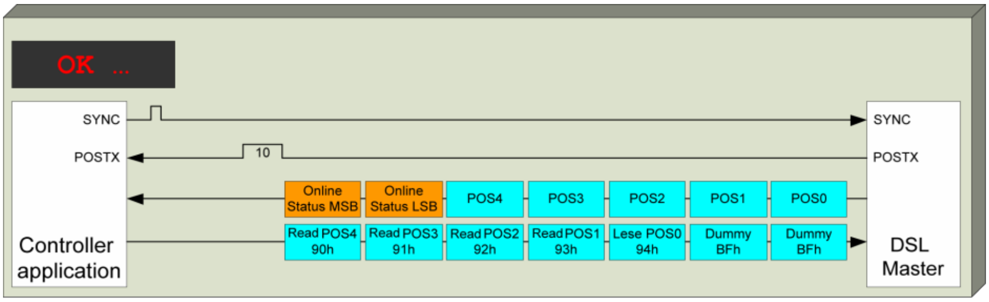


Figure 22: Polling registers for the fast position in SYNC mode

NOTE
It should be noted that polling of fewer than the five full position registers may be appropriate dependent upon the application. This enables fast reading of the position.

The rotation speed of the motor feedback system can be read in the same way. The rotation speed is also measured and transmitted synchronously with the `sync` signal. This is explained in figure 23.

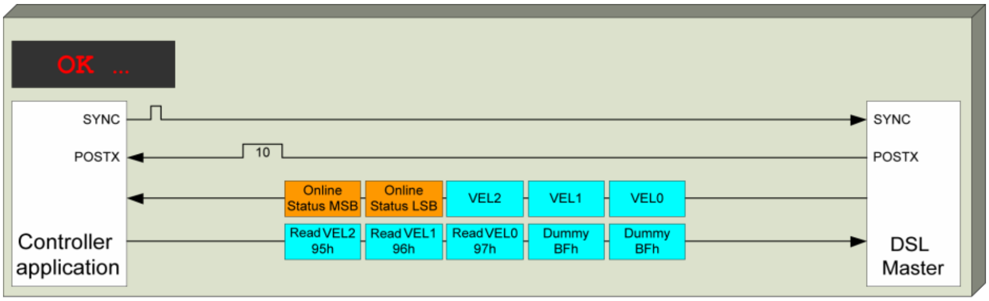


Figure 23: Polling of rotation speed registers in SYNC mode

7.4 Safe position, Channel 1

The motor feedback system safe position is transmitted as a complete absolute position. This makes internal validation of the data transfer possible.

The complete transmission is available every eighth protocol frame. Therefore the position values are older than the fast position values received in the same frame.

CAUTION
The safe position is not synchronized with the last control cycle present at the DSL Master IP Core. The safe position should not be used in the control circuit for frequency inverter position or speed.

The safe position is stored in the `VPOS0…4` registers and can be polled via drive interface (see figure 24).

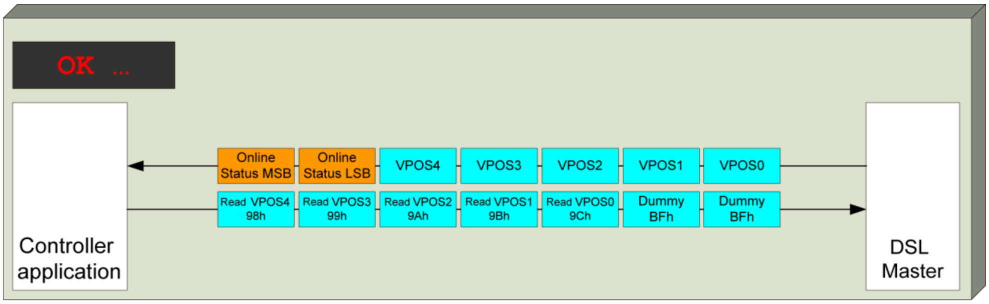


Figure 24: Polling the safe position

As soon as the DSL master identifies a difference between the transmitted safe position and the integrated fast position, the **POS** error bit is set in the `EVENT_H` register (see chapter 7.5).

7.5 Parameters Channel

The HIPERFACE DSL® Parameters Channel is for access to the motor feedback system parameters.

Using two separate access mechanisms, the Parameters Channel distinguishes between two separate data areas:

- Interface information is polled via "short messages".
- Information on the motor feedback system is polled via "long messages".

7.5.1 Short message

Remote (DSL motor feedback system) registers that indicate interface information are mirrored in the DSL Master under register addresses 40h to 7Fh. These remote registers are addressed in the same way as DSL Master registers.

As the values of remote registers are transmitted via the Parameters Channel and hence via the DSL cables, the delay between polling and answer for "short message" transactions depends on the connection cables of the systems in question. Unlike DSL Master registers, the frequency inverter application must wait for the answer to arrive.

Although remote registers are addressed and written in the same way as DSL Master registers, the answer is recorded in a special DSL Master register (`PC_DATA`, 2Fh).

The value of the direct answer that reaches SPI1 MISO during reading or writing is a dummy value.

In the `EVENT_L` DSL Master register, **FRES** indicates whether the "short message" channel is busy or whether the answer has reached the DSL Master. **FRES** can be evaluated for all SPI1 operations as the register content is a component of every SPI1 transmission (bit 0 in **ONLINE STATUS L**, see chapter 6.2).

The Parameters Channel can only transmit one "short message" at a time. Several remote registers can only be polled in sequence, i.e. after the previous answer has been received.

NOTE
It should be noted that a "short message" can be triggered during a running "long message" transaction (see chapter 7.5.2) and vice versa.

The following figure gives an example of reading from the remote register `ENC_ST0` (40h).

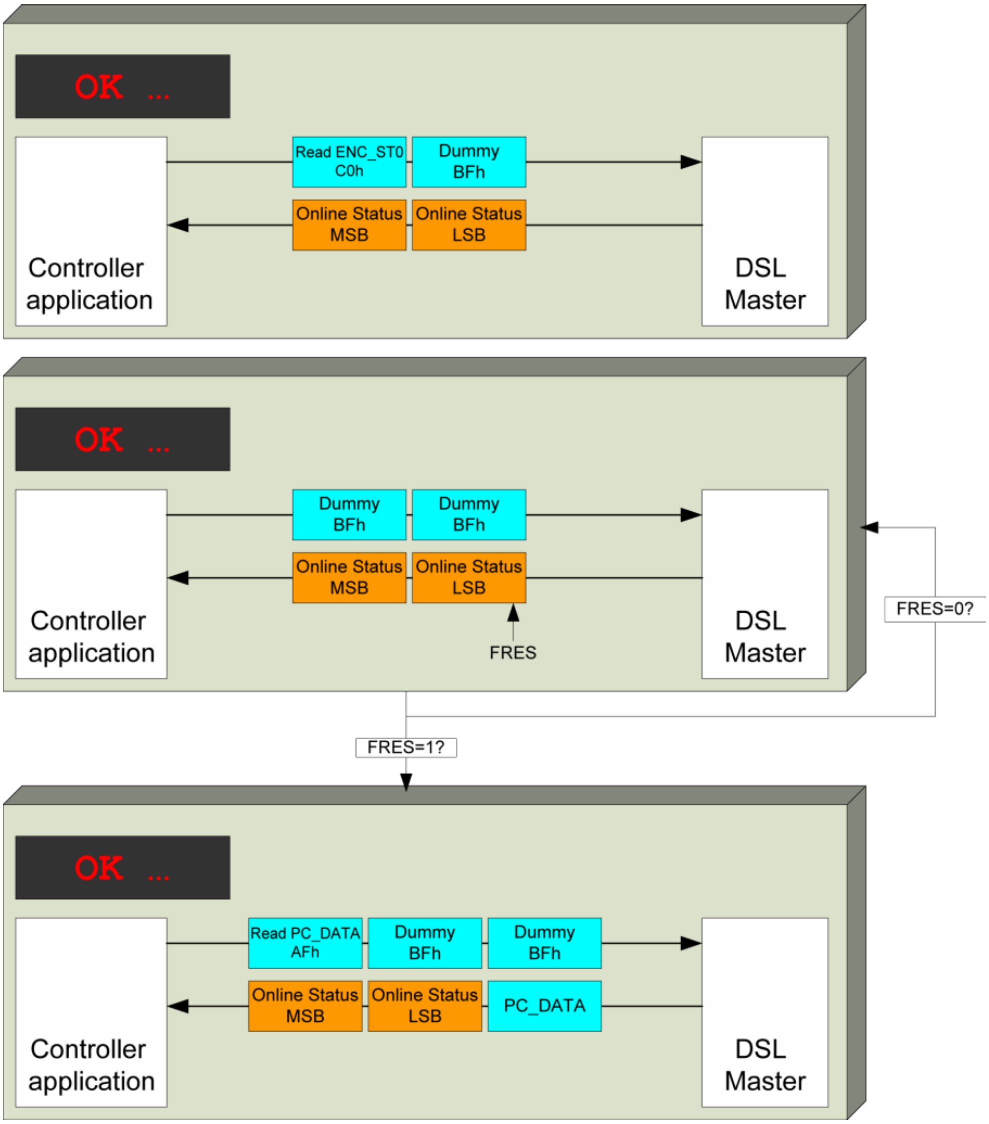


Figure 25: Reading from remote register

7.5.2 Long message

Apart from the interface registers (see chapter 6.4), access to resources of the motor feedback system takes place by "long message" transactions on the Parameters Channel.

The organization and scope of the resources depend on the particular DSL Slave and DSL encoder installation.

A "long message" is triggered by setting the corresponding "long message" registers (`PC_ADD_H/L`, `PC_OFF_H/L`, `PC_CTRL` and – for write operations – `PC_BUFFER0:7`). The result, where present, is recorded in the `PC_BUFFER0:7` registers.

When carrying out a long message transaction, **FREL** is deleted in the `EVENT_L` register. When the transaction has completed, **FREL** is set again.

NOTE
It should be noted that a "long message" can be triggered during a running "short message" transaction (see chapter 7.5.1) and vice versa.

A "long message" transaction enables the exchange of general parameter data between the frequency inverter and the motor feedback system. These parameters can contain information on the status of the motor feedback system, control data for the motor feedback system or user-defined data.

Individual parameters are defined as resources of the motor feedback system.

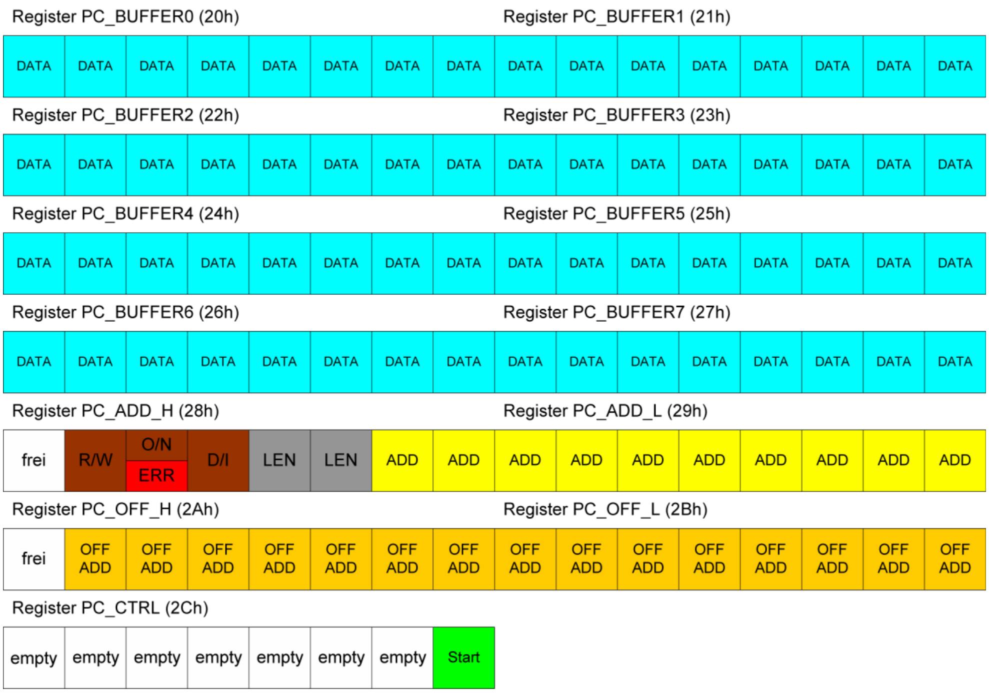
Chapter chapter 8 lists the usual resources of a DSL encoder. Resources that have actually been installed are specified in the data sheet for individual DSL encoders.

A "long message" is triggered by the setting of the corresponding `PC_BUFFER`, `PC_ADD`, `PC_OFF`and `PC_CTRL` (20h to 2Ch) registers in the DSL Master.

Whilst the motor feedback system is processing a "long message", the **FREL** flag in the `EVENT_L` (05h) events register is deleted. Once the processing is finished, this flag is set once more to indicate readiness to process a fresh "long message".

After the setting of a **FREL** flag has been indicated, the data returned from a read access can be polled in the `PC_BUFFER` registers (see chapter 6.3.17).

NOTE
It should be noted that only one "long message" can be processed at a time. Access to resources with more than 8 bytes must be done using successive "long messages".



Register PC_BUFFER0 (20h)								Register PC_BUFFER1 (21h)							
DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA
Register PC_BUFFER2 (22h)								Register PC_BUFFER3 (23h)							
DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA
Register PC_BUFFER4 (24h)								Register PC_BUFFER5 (25h)							
DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA
Register PC_BUFFER6 (26h)								Register PC_BUFFER7 (27h)							
DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA
Register PC_ADD_H (28h)								Register PC_ADD_L (29h)							
frei	R/W	O/N ERR	D/I	LEN	LEN	ADD	ADD	ADD	ADD	ADD	ADD	ADD	ADD	ADD	ADD
Register PC_OFF_H (2Ah)								Register PC_OFF_L (2Bh)							
frei	OFF ADD	OFF ADD	OFF ADD	OFF ADD	OFF ADD	OFF ADD	OFF ADD	OFF ADD	OFF ADD	OFF ADD	OFF ADD	OFF ADD	OFF ADD	OFF ADD	OFF ADD
Register PC_CTRL (2Ch)															
empty	empty	empty	empty	empty	empty	empty	Start								

Figure 26: "Long message" characteristics

The meaning of each characteristic is described in the table below.

Table 33: "Long message" characteristics

Characteristic	Description
DATA	Content of the "long message"
R/W	Direction of the "long message" (read/write)
O/N	Only for triggering the message: "Long message" mode (with offset/without offset)
ERR	Only for answer to the message: Error indication
D/I	"Long message" mode (direct/indirect addressing)
LEN	Data length of the "long message" (0/2/4/8 bytes)
ADD	Identification/address of the resource for a "long message"
OFF ADD	Offset address of the resource for a "long message"
Start	Trigger for the transmission of the "long message"

DATA contains all the data to be transmitted during write access to the motor feedback system. After a read access, **DATA** contains all the data from the motor feedback system.

Dependent upon the **LEN** characteristic, separate areas of the **DATA** register are used.

Table 34: DATA register areas

LEN value	Data length	DATA register used
0 (00b)	0 bytes	No data transfer
1 (01b)	2 bytes	PC_BUFFER0 PC_BUFFER1
2 (10b)	4 bytes	PC_BUFFER0 PC_BUFFER1 PC_BUFFER2 PC_BUFFER3
3 (11b)	8 bytes	PC_BUFFER0 PC_BUFFER1 PC_BUFFER2 PC_BUFFER3 PC_BUFFER4 PC_BUFFER5 PC_BUFFER6 PC_BUFFER7

The **R/W** "long message" characteristic is used to determine whether a read or write access is programmed.

Table 35: R/W value for the "long message"

R/W value	Direction of the "long message"
0	Write
1	Read

For programmed write access, the data to be transferred must be present in the **DATA** characteristic.

The "long message" characteristic **O/N** determines whether the message is transmitted with or without an offset address.

Table 36: O/N value for the "long message"

O/N value	"Long message" mode
0	No offset addressing
1	Offset addressing

The resource description in chapter 8.2 contains an explanation of the purpose for which offset addressing is used. Using offset addressing, an additional "long message" parameter can be transmitted to the motor feedback system as well as the address (**ADD**) and the message data (**DATA**).

If the **O/N** characteristic is set to "1", then the **OFF ADD** characteristic must contain the value for the offset address characteristic.

The same **O/N** bit in the **PC_ADD_H** register can be read after receipt of the "long message" answer to determine the **ERR** error characteristic.

Table 37: ERR value for the "long message"

ERR value	Error during resource access
0	No error
1	An error was identified

If the motor feedback system discovers an error during a resource access, the **ERR** bit is set and the **LEN** characteristic is set to 2 bytes (01b).

In this case the `PC_BUFFER0` and `PC_BUFFER 1` **DATA** registers will contain an error code as detailed in chapter 7.6.6. This error code enables precise error handling for "long messages".

D/I determines whether direct or indirect addressing is used for a "long message".

Table 38: D/I value for the "long message"

D/I value	"Long message" addressing
0	Direct addressing
1	Indirect addressing

The resource description in chapter 8.2 contains an explanation of the purpose for which direct or indirect addressing is used.

The **LEN** characteristic determines the data length of the "long message".table 34 describes the use of this characteristic.

LEN must correspond to the permitted values applicable to the resource addressed (see chapter 8.2). If these values are not observed, the "long message" in the motor feedback system will be ended and a corresponding error message indicated.

The **ADD** characteristic determines the target resource of the "long message". The **ADD** value corresponds to the RID resource index.

Table 39: ADD value for the "long message"

ADD value	Resource index (RID)
000h to 3FFh	000h to 3FFh

Access to resources not installed in the motor feedback system is ended with a corresponding error message.

The **OFF ADD** "long message" characteristic contains the offset address, provided offset addressing is used (see above under the **O/N** characteristic). The resource description in chapter 8.2 contains an explanation of the permitted scope and purpose of each individual resource.

Table 40: OFF ADD value for the "long message"

OFF ADD value	Register used
0000h to 7FFFh	PCR_ADD_H/PCR_ADD_L

Access to a resource with an invalid **OFF ADD** value, or one that is too high, will cause the "long message" in the motor feedback system to be ended and a corres- ponding error message will be indicated.

The table below gives an example of a "long message" read command.

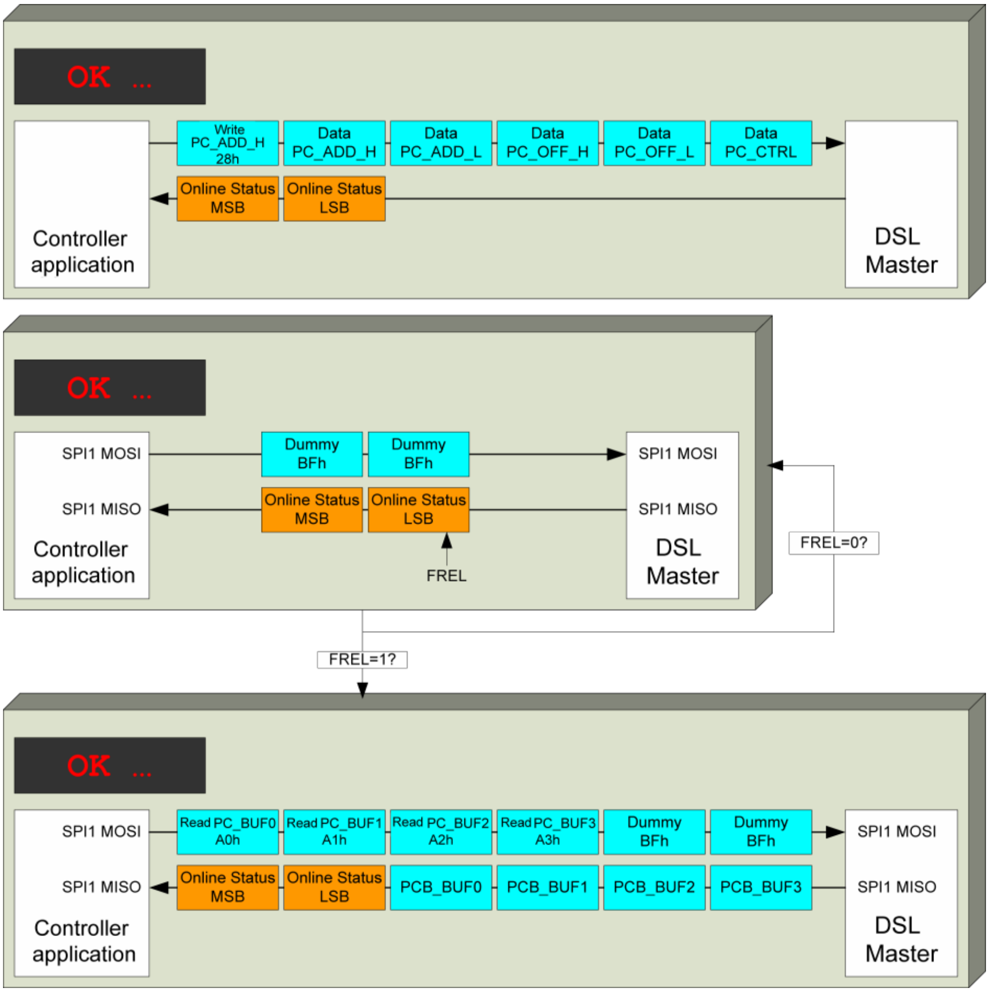


Figure 27: Example of a "long message" read command

7.5.3 Error handling in the Parameters Channel

Errors in a "short message" are handled differently than for a "long message".

If a "short message" is transmitted to the motor feedback system with an error, the protocol sends the message again automatically until an acknowledgment of correct transmission is received. This is not explicitly indicated to the frequency inverter. The **FRES** flag remains deleted until correct receipt of the answer to the "short message".

If the DSL Master receives no acknowledgment of the transmission of a "short message", the protocol automatically begins cyclic repetition of the transmission.

There is no limit to the number of repetitions and the user must decide when to issue a message reset.

If the DSL Master receives no acknowledgment of the successful reception of a "long message" from the DSL slave, the protocol automatically begins a cyclic repetition of the transmission until such an acknowledgment is received. This has no internal timeout.

If a "long message" is transmitted correctly to the motor feedback system but the answer received from the DSL slave is not valid, the **ANS** flag in the `EVENT_L` register will be set. In that case, "the long message" will not be repeated. The user needs to perform the same "long message" action once again.

In case of a "long message" being correctly transmitted and an acknowledgement received, the DSL Master will wait for an answer from the DSL Slave. As the processing time of a long message cannot be reliably predicted, there is no timeout implemented in the DSL Master. To determine the time overrun for a "long message" transaction, the user must refer to the time overrun characteristic in each individual resource of the DSL motor feedback system (see chapter 8.2).

To be able to use the Parameters Channel again in case of a pending "short message" or "long message" that is blocking the corresponding message channel, the user application must trigger a Message reset of the Parameters Channel (see chapter 6.3.1).

This reset does not affect position measuring or the transmission of position data. The reset sequence for the Parameters Channel is specified in figure 28.

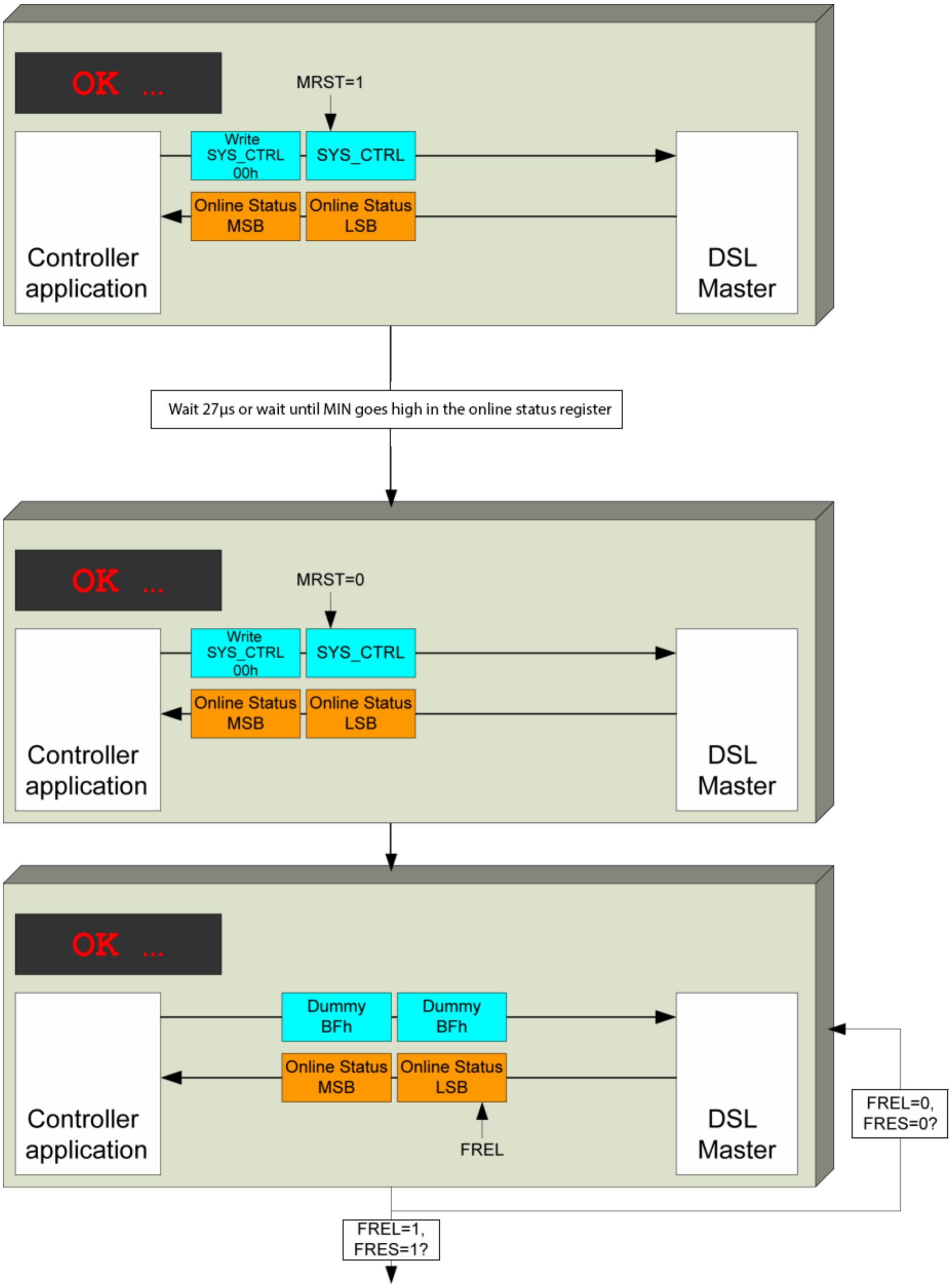


Figure 28: Reset of the Parameters Channel

7.6 Status and error messages

HIPERFACE DSL® can be used to monitor the status of the motor feedback system in various ways.

Dependent upon the importance of the status or error message, different indication mechanisms are used to inform the frequency inverter application.

7.6.1 Event register

The `EVENT_H` and `EVENT_L` registers (see chapter 6.3.4) contain all important error and status indications for the DSL Master. All events are updated after 200 µs at the latest.

More specifically, the `EVENT_H` register contains all the critical motor feedback system error messages. Recommendations for error handling can be found in chapter 6.3.4.

The `EVENT_L` register contains all motor feedback system warning and status messages. Recommendations for error handling can be found in chapter 6.3.4.

All errors and warning conditions indicated in the event registers must be acknowledged by deletion of the corresponding error bits. The DSL Master does not automatically reset these bits.

This mechanism is explained below using an example (error in the transmitted fast position, **POS** bit).

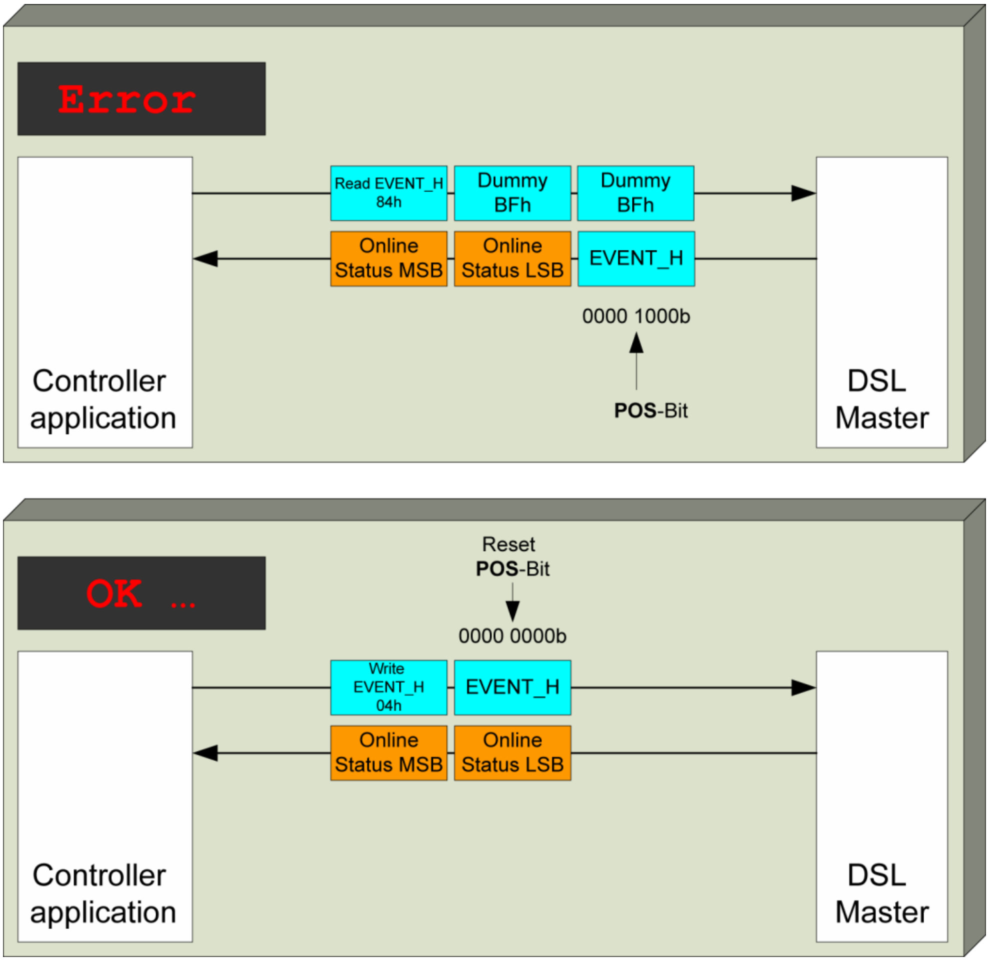


Figure 29: Acknowledgment of event bits

In the frequency inverter application, three mechanisms can be installed to allow timely reaction to reports in the event registers.

- These registers are polled cyclically.
- The Online Status is polled cyclically. Event registers are mirrored here (see chapter 7.6.2).
- Either all, or individual event register events can be masked in the event mask registers (registers `MASK_H` and `MASK_L`, see chapter 6.3.5), in order to issue events via the interrupt interface (see chapter 5.3.2).

7.6.2 Online Status

The Online Status (see chapter 6.2) is transmitted during every SPI communication via drive interface between the frequency inverter application and the DSL Master. The status contains the error and status reports from the event registers.

Unlike with direct polling of the event registers, the Online Status only shows the current status values. As soon as the error status of the motor feedback system becomes unavailable, the error is no longer indicated in the Online Status.

The event registers retain the error statuses until the registers are acknowledged. After acknowledgment, the event registers are reset (see chapter 6.3.4).

The Online Status is updated after 200 µs at the latest.

7.6.3 Status summary of the motor feedback system

In addition, detailed motor feedback system errors and warnings are indicated in the `SUMMARY` status summary register (18h, see chapter 6.3.14).

Each individual bit of the register indicates an error status of a functionality in the motor feedback system (see table 41). The safety relevance of all of these error groups is precisely described in this table.

It should be noted that the reading of detailed motor feedback system error messages enables a more precise reaction to all fault indications in the status summary. In any event a position-relevant fault will always also be signaled directly through the corresponding position fault indicators POS,
VPOS, or VPOS2.

Table 41: Motor feedback system error groups

Bit no.	Error group
0	1 = Fast position error
1	Safe position error
2	Installation error
3	Monitoring error
4	Error when accessing a resource
5	Reserved
6	Reserved
7	User-defined warnings

A bit set in the status summary register definitely indicates that one or more individual errors in the motor feedback system have been recognized. The individual errors can be determined by polling the remote encoder status register `ENC_ST` (see chapter 6.4.1).

NOTE
It should be noted that the **SUM** error bit in the `EVENT_H` event register represents an aggregated summary of all error groups (see chapter 6.3.5).

NOTE
It should be noted that the reading of detailed motor feedback system error messages enables a more precise reaction to all fault indications in the status summary.

7.6.4 Motor feedback system error messages

Errors recognized in the motor feedback system are indicated in the `ENC_ST` remote encoder status registers.

Access to a remote encoder status register can last up to 75 ms.

A summary of error groups is indicated at the appropriate time in the status summary register in the DSL Master (see chapter 7.6.3).

To simplify error handling , the motor feedback system errors are grouped.

table 42 below contains all error messages and recommendations for error handling . Depending on product family only some of the following errors might be reported. Please consult individual product data sheets for availability of error messages.

Within the column "Severity" the different information is classified for general drive controller reactions. The column "Required response" provides a proposal for a first drive controller response to each particular message. In this case SW reset means to perform an encoder reset via RID 100h. For HW reset power have to be taken off for at least 1 sec.

Table 42: Motor feedback system error messages

Error register	Error bit	Product compatibility	Notification	Severity	Persistence p-permanent, np-not permanent)	Description	Required error handling
ENC_ST0 (adr. 40h)	7	EKx36, EFx50, EDx35	Position Synchronization Error	Minor	np	Position sensor subsystems could not match their signals due to shocks, hardware faults or too high speed during power-up. Synchronization of the counted position and a new absolute position was not possible.	Attention! If this error is accompanied by the following (position synchronization related) critical errors, the system must be placed in a safe state: • Counter Error (ENC_ST0) • Position Cross-check Error (ENC_ST1) Meaning, it becomes a critical severity error. In any other case, it can be handled as a minor severity error and no action required.
		EEx37	Cross-check Error	Critical	p	Cross-check of the two safety channels failed.	The drive system must be placed in a safe condition. Action: SW reset of the encoder.
	6	EKx36, EFx50, EDx35	Counter Error	Critical	p	Position quadrant counter has detected an invalid sequence of signals Occurrence during operation leads to a permanent error.	The drive system must be placed in a safe condition. Action: SW reset of the encoder.
		EEx37	Improper Configuration Error			Encoder has detected an invalid configuration. Possible cause: allocation misalignment between encoder and motor shaft or general encoder hardware problem. Occurrence during operation leads to a permanent error.	Bit will be set after start-up, so clearing is needed. Action: SW reset of the encoder

Error register	Error bit	Product compatibility	Notification	Severity	Persistence p-permanent, np-not permanent)	Description	Required error handling
ENC_ST0 (adr. 40h)	5	All	Position Vector Length Error	Critical	np	The Vector Length of the sampled and measured Sin-Cos signals has exceeded its boundaries. Possible causes: HW or supply faults, mechanical shocks.	When this error occurs, the IP Core advances the fast position in linear fashion, by turning its estimator on (estimator_on = 1), until valid values are present again. Dependent upon the application, one or more vector length errors in succession can be tolerated. The tolerable number of errors can be determined from a calculation of the maximum deviations in each occurring error. Recommendation: monitor (count) the occurrence of this error. If the maximum tolerable number of errors is exceeded, the drive system must be placed in a safe state, meaning a critical severity. If necessary, the safe position can be used as an alternative to position measurement, if this error arises for the fast position. In this case the significantly slower refresh cycle of the safe position must be considered. If the error persists, there is probably a general hardware or mechanical failure. Inform customer service.
	4	EKx36, EFx50, EDx35	Position Tracking Filter Error	Minor	np	The sine/cosine-signals got a big distortion and the signal tracking (interpolation) got lost. Occurs, when check is enabled and tracking is out of limit. Not considered safety relevant. Affecting only Safety Channel 1.	No action required. Encoder recovers automatically.
		EEx37	Plausibility Error	Critical	p	Potential cause: faults affecting the sensor, the analog input chain, the excitation-signals. Also possible: sensor-signal distortion due to electrical-burst or mechanical shock. The sensor-signal behavior is not as expected: plausibility result is greater than the allowed threshold. Will lead to a VPOS-error.	The drive system must be placed in a safe condition. Action: SW reset of the encoder.

Error register	Error bit	Product compatibility	Notification	Severity	Persistence p-permanent, np-not permanent)	Description	Required error handling
ENC_ST0 (adr. 40h)	3	EKx36, EFx50, EDx35	Not implemented. Always ‘0’	-	-	-	-
		EEx37	Drift Compensation Error	Critical	p	The position drift was not properly compensated. Will lead to a VPOS-error.	The drive system must be placed in a safe condition. Action: SW reset of the encoder.
	2	EKx36, EFx50, EDx35	Test Running	Critical	np	Available only for safety encoders. Indicates that a diagnostic test has been requested by application.	Action required as detailed in the safety implementation manual see table 56.
		EEx37	Not implemented. Always ‘0’	-	-	-	-
	1	All	Acceleration Overflow Error	Minor	np	Two subsequent fast position values were too far apart for valid DSL transmission; too high shaft acceleration or shaft was turned while the link was not active.	No action required. Fault is not permanent and encoder recovers automatically. This error will likely will lead to other errors. Possible causes: mechanical shocks or hardware faults. If the error persists, there is probably a general hardware or mechanical failure. Inform customer service.
	0	All	Protocol Reset Indication	Minor	np	Indicates that the DSL protocol has been reset and a new connection was established. Serves a base indication for the start of the communication (like LINK). If not set after an encoder reset, no status registers can be relied upon!	Action: successful clearing is required. If this indication persists (bit is set), it is likely that the encoder was again resetted. Check connection or supply of the encoder. Before the successful clearing of this indication, no other ENC_ST information can be relied upon!

Error register	Error bit	Product compatibility	Notification	Severity	Persistence p-permanent, np-not permanent)	Description	Required error handling
ENC_ST1 (adr. 41h)	4	EKx36, EFx50, EDx35	Position Cross-check Error	Critical	p	Only for safety encoders and HF2DSL coupler. Internal cross-check of safe absolute position has failed. In case of non-safe encoders, this bit is always ‘0’. Potential cause(s): hardware faults; for linear encoders: alignment fault read-head/ measure.	The drive system must be placed in a safe condition. Action: SW-reset of the encoder.
		EEx37	Not implemented. Always ‘0’	-	-	-	-
	3	All	Multi-turn Vector Length Error	Critical	p	Encoder has detected an invalid multi-turn sensor signal vector length. Potential cause(s): excessive mechanical stress	The drive system must be placed in a safe condition. Action: SW-reset of the encoder
	2	All	Multi-turn Sync Error	Critical	p	Multi-turn sensor synchronization was erroneous. Potential cause(s): encoder reached end of lifetime, hardware faults, magnet (clip) faults or gear wear-out. NOTE: A multi-turn synchronization diagnostic cannot reliably detect gear synchronization faults over extended time. After first indication of this fault gear synchronization can wear out even more and result in undetected false position output.	The drive system must be placed in a safe condition. Action: immediate replacement of encoder
	1	All	Multi-turn Amplitude Error	Critical	p	Invalid multi-turn sensor signal amplitude. Potential cause(s): magnetic disturbances, multi-turn sensor defects, loss of multi-turn magnets or errors in multi turn parameters.	The drive system must be placed in a safe condition. Action: immediate replacement of encoder
	0	All	Single-turn Error	Critical	p	Initialization of single-turn sensor was erroneous, or the encoder could not measure absolute single-turn position. This results in an overall invalid encoder position. Potential causes: hardware failures or external shocks.	The drive system must be placed in a safe condition. Action: SW-reset of the encoder

Error register	Error bit	Product compatibility	Notification	Severity	Persistence p-permanent, np-not permanent)	Description	Required error handling
ENC_ST2 (adr. 42h)	6	All	Internal System Error	Critical	p	After startup, an internal electronic error was identified. (Encoder initialization timeout; FW CRC error; internal transmission faults (SPI) of safety-relevant parameters). Potential cause(s): uC faults, hardware faults; EMC noise injection	The drive system must be placed in a safe condition. Action: SW-reset of the encoder
	5	All	Internal Communication Error 2	Minor	np	During startup an internal communication error has occurred (I^2C).	No action required. If error persists, potential EEPROM hardware failures are present.
	4	All	Internal Communication Error 1	Minor	np	During or after startup an internal communication error has occurred (SPI).	No action required. Encoder recovers automatically. If error persists, potential hardware failures are present.
	3	All	Standard Parameter Error	Critical	p	There were errors in the internal encoder EEPROM parameter or diagnosis that could not be rectified.	The drive system must be placed in a safe condition. Action: SW-reset of the encoder
	2	All	Safety Parameter Error	Critical	p		The drive system must be placed in a safe condition. Action: SW-reset of the encoder If error persists, potential EEPROM hardware failures are present. Encoder replacement is required!
	1	All	Safety Parameter Warning	Minor	np	There were errors found in the safety parameters, which were rectified.	No action required.
	0	All	Power-on Self-test Conducted	Critical	np	Only for safety encoders. Mandatory to read this bit at startup for all safety encoders.	Action: successful clearing is required. If this indication persists (bit is set), it is likely that the encoder was again resetted. Check connection or supply of the encoder. Before the successful clearing of this indication, no other ENC_ST information can be relied upon!

Error register	Error bit	Product compatibility	Notification	Severity	Persistence p-permanent, np-not permanent)	Description	Required error handling
ENC_ST3 (adr. 43h)	6	All	Internal Monitoring Error	Minor	np	During monitoring of an internal electronic, an error was identified.	No action required. Encoder might recover automatically. If error persists even after clearing, potential hardware failures are present and immediate replacement of the encoder is required!
	5	All	Counter Overflow	Minor	np	User counter has overrun.	Depending on counter use case.
	4	All	Critical Acceleration	Minor	np	Acceleration was out of specification.	No action required; check application, reduce acceleration or check for shock.
	3	All	Critical Rotation Speed	Minor	np	Rotation speed was out of specification.	No action required. Check application for higher speeds than specified. Possibly reduce speed.
	2	All	Critical Supply Voltage	Minor	np	Supply voltage was out of specification.	No action required. Check encoder power supply conditions . If error persists after ensuring the input power supply, potential hardware failures are present and immediate replacement of the encoder is required.
	1	EKx36, EFx50, EDx35	Critical LED Current	Minor	p	Monitored sensor behavior or code disc position out of specification. Potential cause: encoder sensor near end of lifetime, sensor is polluted, encoder was used outside of its specification.	Action: SW-reset of the encoder Check application for potential pollution (dust/brake dust). If error persists even with correctly set rotor position, replacement of the encoder is required.

Error register	Error bit	Product compatibility	Notification	Severity	Persistence p-permanent, np-not permanent)	Description	Required error handling
ENC_ST3 (adr. 43h)	1	EEx37	Critical Rotor Position	Critical	p	Only for capacitive encoders: Rotor position out of specified tolerance. Attention! A proper and correct mounting of the encoder has to be ensured, otherwise proper functionality can't be guaranteed! Potential cause: mechanical shocks, improper encoder mounting	The drive system must be placed in a safe condition, because with distorted signals no proper functionality can be guaranteed Action: SW reset of the encoder. Upon reset, the Rotor Position will be checked again. If error bit is set, check application and encoder mounting. If error persists even with correctly set Rotor Position, replacement of the encoder is required!
	0	All	Critical Temperature	Minor	np	Encoder temperature was out of the specification	No action required; check application for potential motor/encoder cool down
ENC_ST4 (adr.44h)	3	All	File access error	Minor	np	Error when accessing a file in EEPROM. Potential cause: may indicate EMC problems or corrupted file system due to previous power-down during write access or just simply incorrect access parameters.	Check and retry command.
	2	All	Resource access error	Minor	np	Error when accessing an internal resource. Potential cause: may indicate EMC problems, excessive number of EEPROM write cycles or wrong timing of power cycle vs. EEPROM write access or just simply incorrect access parameters.	Check and retry command; check resource access sequence.
	1	All	Access denied	Minor	np	Access to a resource was denied.	Check and retry command; check access user level.
	0	All	Invalid access	Minor	np	Invalid argument/command of a resource access. Potential cause: drive firmware fault or internal signal transmission fault (due to EMC) or just simply incorrect access parameters.	Check and retry command; check drive firmware.
ENC_ST7 (adr. 47h)	0-7	All	User defined warning	Minor	np	Depending on user customization (see chapter 9.8.5).	Depending on configuration.

P – bit is set again in slave, after every clearing by master (acknowledgement)

Np- bit is cleared in slave after clearing by master (acknowledgement)

For all Critical errors the following rule applies: following a critical error indication, it is probable that the position and rotation speed of the encoder are wrong. The drive system must be placed in a safe condition.

7.6.5 Recommend fault handling

Within the previous chapters a summary is shown of all the different information which the motor feedback system provides to monitor and analyze operation and error conditions of the drive system.

For non-safety applications it is recommended to watch the fast position information only. As long as the fast position is valid there is no loss of accuracy in motor controlling and the system can ride through any error situation within the set tolerances.

The simplest fault handling in this case would only watch the `DEV_THR_ERR` signal (see chapter 5.4.3). When set to “1” the maximum tolerable deviation between the mechanical and the calculated position value is violated and the allowed fast position error is exceeded. In this case the application has to be stopped and brought into safe conditions, respectively. The tolerance for this signal can be adjusted by the user via fast position acceleration boundary (MAXACC register; chapter 6.3.25) and the fast position estimator deviation (MAXDEV register; chapter 6.3.26).

This information is also available within the EVENT_H register Bit 1 (DTE; chapter 6.3.4) as well as the online_status_d/high byte at Bit 1 (chapter 6.2) which is a copy of the EVENT_H-register. The online_status is non-persisting information while the EVENT_H register content is latched. Online Status

For root cause investigations of an error situation further encoder messages can be read.

It is further recommended to monitor the `estimator_on` signal and/or POS flag, which are provided as a further digital output of the DSL-master (see also chapter 5.4.2). It indicates that the fast position value is invalid and the current value is supplied by the position estimator (see chapter 7.3.1). For statistical analyses of the stability of the position reading this signal can be monitored in reference to a defined time base. In this case it is possible to identify changes of the position reading behavior of the system.

This information is also available within the EVENT_H-register Bit 3 (POS; chapter 6.3.4) as well as the online_status_d/high byte at Bit 3 (chapter 6.2), which is a copy of the EVENT_H register. The online_status is a non-persisting information, but the EVENT_H register content needs to be cleared after reading.

Information like SUM (high byte/Bit 6) or even QMLW (low byte/Bit 2) at the online_status_d or EVENT_H/L registers shall not be solely used for fault handling. These are summary information where several different contents are combined by OR, which sometimes only report a line or link quality status (i.e. quality monitoring at MASTER_QM; chapter 6.3.3).

During start-up of the motor feedback system (either due to power-on or due to a reset command) internal checks are carried out and some notifications are set that need to be treated different to normal operation conditions.

While the motor feedback system is powered-down or in reset multiple notifications and errors will be shown in the online status. These are set by default and they are cleared after the motor feedback system starts sending correct information to the DSL-master. If the online status still contains transmission or position fault indications after the specified initialization time has elapsed reactions have to follow the recommendations of the previous chapters.

Any notifications stored in the Events registers (see chapter 6.3.4) or in the encoder status registers (see chapter 6.4.1) during reset need to be acknowledged once after initialization. If a notification in one of these registers persists after acknowledgment the problem handling needs to follow the recommendations as shown in the previous chapters.

7.6.6 Long message error code

Due to the complexity of "long messages", errors occurring here are reported in detail to the user.

If the motor feedback system establishes an error when accessing a resource, this error is displayed as an error message (see chapter 7.6.4). In addition the **ERR** flag is set, the **LEN** characteristic is set to 2 bytes (01B) and the `PC_BUFFER0` and `PC_BUFFER1` DATA registers contain an error code.

By means of this error code, the errors in a "long message" transaction can be understood in detail.

The table below contains the error codes and their meaning.

NOTE
It should be noted that the value of the `PC_BUFFER1` register corresponds to the error code in the `ENC_ST` encoder status register (see chapter 6.4.1).

Table 43: "Long message" error codes

PC_BUFFER1	PC_BUFFER0	Meaning of the error code
40h	10h	Resource address not installed in the encoder
	11h	Incorrect length for resource access given
	12h	Incorrect length for direct resource access given
	13h	Offset address too high
	14h	Invalid offset address
	15h	Invalid "long message" characteristic
	16h	Missing offset address
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
42h	10h	Resource database entry damaged
	11h	Time overrun during resource access
	12h	Internal processing error during resource access
43h	11h	File name was not found
	12h	Invalid address for file access
	13h	File size may not be altered
	14h	Memory location for files full
	15h	File allocation table damaged
	16h	No file loaded for action
	17h	File exists with the same name

8 Motor feedback system resources

The resources of a DSL motor feedback system make up most of the functions of the sensor.

"Long message" transactions enable access to all resources installed in a DSL motor feedback system. Examples of resources are the values for encoder designation, function and fault monitoring, sensor administration or the storage of user-defined data (i.e. electronic type label).

NOTE

It should be noted that for motor feedback system process values, i.e. position and rotation speed values, separate access mechanisms apply (see chapter 7.3 and chapter 7.4).

The resources installed in a DSL motor feedback system are accessible via the resources database (RDB). A "long message" is always aimed at an individual RDB entry.

The resources set out in this section describe the normal functions of a DSL motor feedback system. The actual resources installed in individual DSL motor feedback systems are given in the appropriate data sheets.

8.1 Access to resources

Access to the resources of a DSL motor feedback system is possible in two ways. This section also describes how resource definitions can be read by "direct access".

8.1.1 Access by means of an index

Each individual resource is defined by a unique resource index (RID).

A "long message" can be directed at the associated resource by using the RID as the address characteristic (see chapter 6.3.18).

If a resource is accessed via direct access, the resource definition is returned (see chapter 8.1.3).

8.1.2 Access using the tree

The resources database (RDB) is structured in the form of a tree. This enables access to a resource by referencing, in which the access begins with a root resource that returns an indicator to other resources. figure 30 shows this tree structure.

Starting at the "root node" resource with the resource index RID=000h, a read access returns the addresses of the linked nodes. By progressing recursively through further nodes, it is possible to access all levels of the tree.

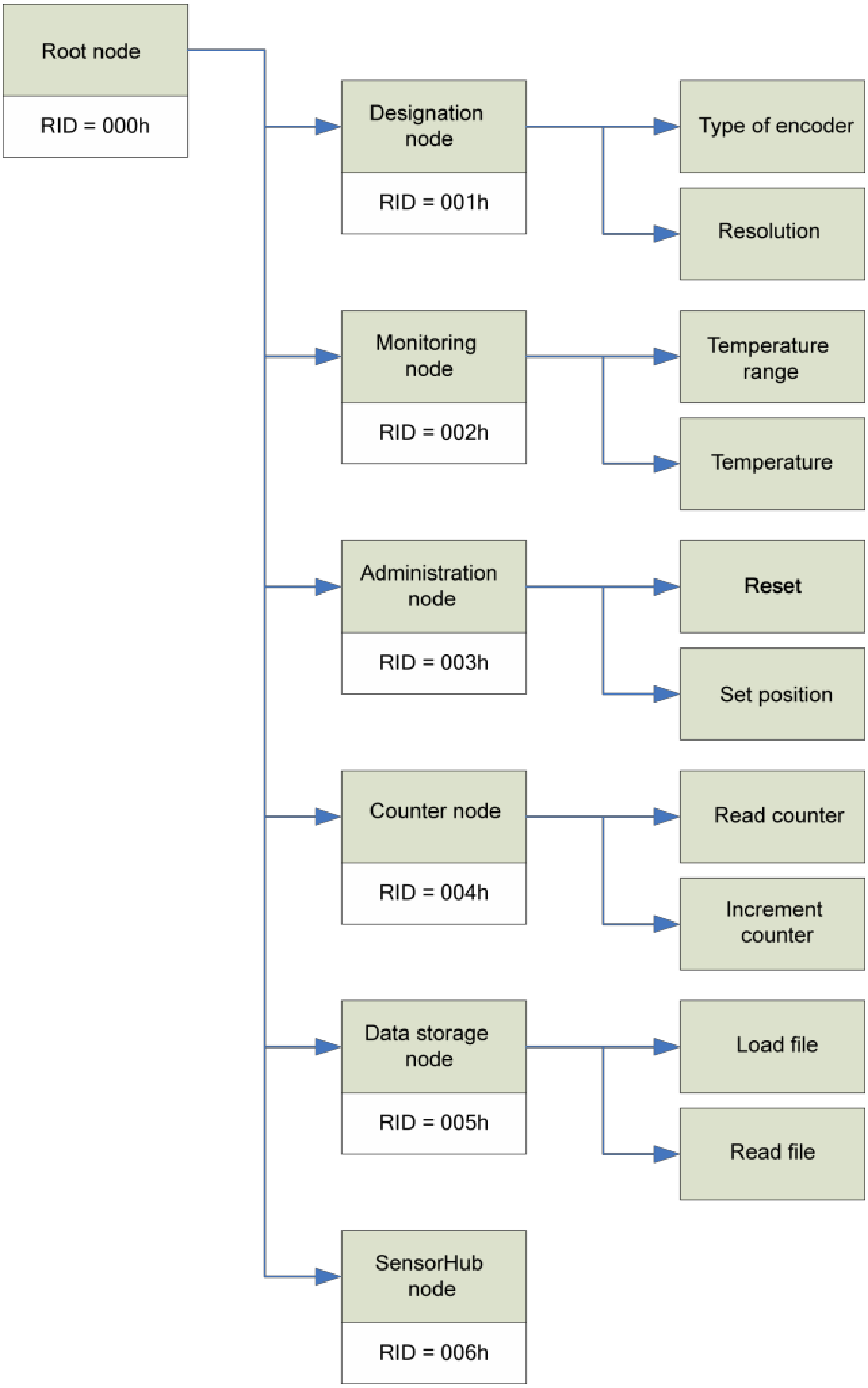


Figure 30: Tree structure of the resources database

The characteristics of a "long message" for reading a linked node are listed in table 44

Table 44: Parameters for node access.

Characteristic	Value	Description
DATA	-	-

Characteristic	Value	Description
R/W	1	Read
O/N	1	Offset
D/I	1	Indirect
LEN	1	2 bytes
ADD	Variable	Calling node
OFF ADD	Variable	Ordinal number of the linked node

The value stored in **OFF ADD** shows the ordinal number of the linked node, the resource identification of which should be returned. The ordinal number is given in the following list of all resources.

The result of this "long message" transaction is the resource index (RID) of the resource requested.

The resources data is described in detail in the resources list (see chapter 8.1.3).

8.1.3 Direct access

The defining values of a resource can be read from the DSL encoder by direct access (see chapter 7.5.2).

These defining values consist of a readable description of the resource (max. 8 characters), the data length, the access rights, a value for time overrun and the data type of the resource.

The desired value is selected by the user by setting a corresponding offset address.

Please note that for different encoders the time overrun values can be different. Therefore it is recommended to check the time overrun values prior to reading. The values provided in the resources list are only examples.

table 45 below sets out all possible access methods (direct and indirect) and their associated values.

Table 45: Different methods of resource access

Resource type (see chapter 8.2)	Access	Offset address	Data	Note
Node	Direct reading	0/none	Resource name	e.g. "ROOT"
		1	Resource data length	e.g. 05h for 5 Subentries
		2	Read access level	e.g. 0 for access level 0
		3	Write access level	e.g. 2 for access level 2
		4	Time overrun	e.g. 46h for 70 ms
		5	Data type	00h for node indicator
	Indirect reading	0	Number of linked nodes	e.g. 5
		1	RID of the 1st linked node	e.g. 001h
		...		

Resource type (see chapter 8.2)	Access	Offset address	Data	Note
All remaining	Direct reading	0/none	Resource name	e.g. “ENCTYPE”
		1	Resource data length	e.g. 02h for 2 bytes
		2	Read access level	e.g. 0 for access level 0
		3	Write access level	e.g. 2 for access level 2
		4	Time overrun	e.g. 46h for 70 ms
		5	Data type	e.g. 04h for 16 bits, unsigned
	Indirect read/ write	Variable	Resource value	See chapter 8.2

NOTE

It should be noted that individual device families of the DSL motor feedback systems can contain different resources. The list of available resources is published in the device data sheet.

In the following resources list all those resources that are mandatory features of all existing and future DSL motor feedback systems are indicated as such. For maximum compatibility between DSL components it is recommended that use of optional resources is based on an architecture where the presence of the optional resource is determined before actually accessing the resource. The presence query can be made with "direct access".

NOTE

It should be noted that the defining values of the resources that have been laid down in a motor feedback system have priority over the values published in this manual.

8.2 Resources list

The following sections contain possible resources installed in a DSL motor feedback system.

NOTE

It should be noted that the motor feedback system position and rotation speed values are process values and access to these values is different from access to general resources (see chapter 7.3 and chapter 7.4).

All resources are indicated using the "long message" characteristics that are valid for access (see chapter 7.5.2).

In addition, the definitions from the resources database (RDB) for each resource are described. These definitions are used to indicate the following resource properties:

Table 46: Definitions of the resources database.

RDB definition	Data area	Description
RID	0 – 1023 000h to 3FFh	Resources index: Is used as an address characteristic in a "long message".
Size	0 – 32767 0 to 7FFFh	Length of the resources data in bytes. Defines the area that can be used when accessing offset basis in a "long message".
R		Read access:

RDB definition	Data area	Description
	0	Read possible for all.
	1	For read, the "operator" access level is required.
	2	For read, the "maintenance" access level is required.
	3	For read, the "authorized client" access level is required.
	4	For read, the "service" access level is required.
	15	No read access possible.
W		Write access:
	0	Write possible for all.
	1	For write, the "operator" access level is required.
	2	For write, the "maintenance" access level is required.
	3	For write, the "authorized client" access level is required.
	4	For write, the "service" access level is required.
	15	No write access possible.
Time overrun	0 – 254	Resources access time overrun in milliseconds. If the DSL
		system does not react to a "long message" within this period,
		then there is probably a processing error.
	255	The resource needs more than 254 ms for processing or the
		time overrun is not deterministic.
Resource	00h	Node indicator (index, 16 bit)
data type	01h 02h	Void (no data) Bit (1 = true/0 = false)
	03h	8 bit, unsigned
	04h	16 bit, unsigned
	05h	32 bit, unsigned
	06h	64 bit, unsigned
	07h	8 bit, with sign
	08h	16 bit, with sign
	09h	32 bit, with sign
	0Ah	64 bit, with sign
	0Bh	String (character chain)
	10h to 4Fh	Data structure with data length 0 to 63 bytes

If the size of a resource gives a higher byte total than the data type needs, then it is an array of the data type given.

8.3 Node

All resources of a DSL motor feedback system have a logical tree structure (see chapter 8.1.2). This arrangement is structured with node resources.

An indirect read access to a node returns the address of a linked node or a linked resource. For this, an offset must be given to determine the type of information:

Table 47: Indirect read access to nodes.

Offset	Value
0	Number of linked nodes n
1	RID of the first linked node
...	

Offset	Value
n	RID of the n-th linked node

8.3.1 Root node

The root node is the uppermost resource of the tree structure for the address (RID) 000h.

All nodes representing different resource groups are accessible from the root node. Direct read access to the root node returns the defining values:

Table 48: Root node defining values

Defining value	Offset	Value
RID		000h
Resource name	0	"ROOT"
Data size	1	2
Read access level	2	0
Write access level	3	15
Time overrun	4	75
Data type	5	00h – node indicator
Mandatory		yes

Indirect read access to root nodes returns information on linked nodes (see table 47).

8.3.2 Identification node

The identification node contains indicators to all resources associated with designations in the motor feedback system ("electronic type label").

Direct read access to the designation node returns the defining values:

Table 49: Identification node defining values.

Defining value	Offset	Value
RID		001h
Resource name	0	"IDENT"
Data size	1	2
Read access level	2	0
Write access level	3	15
Time overrun	4	75
Data type	5	00h – node indicator
Mandatory		yes

Indirect read access to the identification node returns information on linked nodes (see table 47).

8.3.3 Monitoring node

The monitoring node contains indicators to all resources associated with monitoring in the motor feedback system (e.g. temperature control).

Direct read access to the monitoring node returns the defining values:

Table 50: Monitoring node defining values.

Defining value	Offset	Value
RID		002h
Resource name	0	"MONITOR"

Defining value	Offset	Value
Data size	1	2
Read access level	2	0
Write access level	3	15
Time overrun	4	75
Data type	5	00h – node indicator
Mandatory		yes

Indirect read access to the monitoring node returns information on linked nodes (see table 47).

8.3.4 Administration node

The administration node contains indicators to all resources associated with administration in the motor feedback system (e.g. reset, determining access level).

Direct read access to the administration node returns the defining values:

Table 51: Administration node defining values.

Defining value	Offset	Value
RID		003h
Resource name	0	"ADMIN"
Data size	1	2
Read access level	2	0
Write access level	3	15
Time overrun	4	75
Data type	5	00h – node indicator
Mandatory		yes

Indirect read access to the administration node returns information on linked nodes (see table 47).

8.3.5 Counter node

The counter node contains indicators to all resources associated with the user- defined counter.

Direct read access to the counter node returns the defining values:

Table 52: Counter node defining values.

Defining value	Offset	Value
RID		004h
Resource name	0	"COUNTER"
Data size	1	2
Read access level	2	0
Write access level	3	15
Time overrun	4	75
Data type	5	00h – node indicator
Mandatory		yes

Indirect read access to the counter node returns information on linked nodes (see table 47).

8.3.6 Data storage node

The data storage node contains indicators to all resources associated with the user defined data storage.

Direct read access to the data storage node returns the defining values:

Table 53: Data storage node defining values.

Defining value	Offset	Value
RID		005h
Resource name	0	"DATA"
Data size	1	2
Read access level	2	0
Write access level	3	15
Time overrun	4	75
Data type	5	00h – node indicator
Mandatory		yes

Indirect read access to the data storage node returns information on linked nodes (see table 47).

8.3.7 SensorHub node

The SensorHub node contains indicators to all resources associated with the identification and actuation of external sensors.

Direct read access to the SensorHub node returns the defining values:

Table 54: Data storage node defining values.

Defining value	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	75
Data type	5	00h – node indicator
Mandatory		yes

Indirect read access to the SensorHub node returns information on linked nodes (see see table 47, page 90).

8.4 Identification resources

The identification resources of the DSL motor feedback system contain the encoder electronic type label.

8.4.1 Type of encoder

The type of encoder describes the basic functionality of the motor feedback system. Direct read access to the type of encoder returns the defining values:

Table 55: Defining values for type of encoder.

Defining value	Offset	Value
RID		080h
Resource name	0	"ENCTYPE"

Defining value	Offset	Value
Data size	1	2
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	04h – 16 Bit, unsigned
Mandatory		yes

The following table contains the possible values for the type of encoder and their meaning.

Table 56: Definition of the type of encoder.

Value (dec.)	Value (hex.)	Type of encoder
0	00 00h	Rotary encoder, bipolar counting
1	00 01h	Linear encoder, bipolar counting
2	00 02h	Rotary encoder, unipolar counting
3	00 03h	Linear encoder, unipolar counting

For this resource, access to the offset base is not meaningful as the size of the resource data is smaller than the maximum for a "long message" transaction.

Table 57: Read type of encoder.

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									54	80	00	00	01
Wait for FREL = 1													
Read	Type of encoder												

8.4.2 Resolution

The resolution value defines the number of steps per rotation of the encoder (rotary encoder) or the length of a measurement step in multiples of 1 nm (linear encoder).

Direct read access to resolution returns the defining values:

Table 58: Resolution defining values

Defining value	Offset	Value
RID		081h
Resource name	0	"RESOLUTN"
Data size	1	4
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	05h – 32 bit, unsigned
Mandatory		yes

The resolution value is given as a 32 bit unsigned value.

For this resource, access to the offset base is not meaningful as the size of the resource data is smaller than the maximum for a "long message" transaction.

Table 59: Reading the resolution.

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									58	81	00	00	01
Wait for FREL = 1													
Read	Resolution (32 bit)												

8.4.3 Measurement range

The measurement range defines the number of coded rotations of the encoder (rotary encoders), or the coded measurement range in multiples of measurement steps (linear encoders).

Direct read access to measurement range returns the defining values:

Table 60: Resolution defining values

Defining value	Offset	Value
RID		082h
Resource name	0	"RANGE"
Data size	1	4
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	05h – 32 bit, unsigned
Mandatory		yes

The measurement range is given as a 32 bit unsigned value.

For this resource, access to the offset base is not meaningful as the size of the resource data is smaller than the maximum for a "long message" transaction.

Table 61: Reading the measurement range.

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									58	82	00	00	01
Wait for FREL = 1													
Read	Measurement range (32 bit)												

8.4.4 Type name

This resource indicates the type name of the encoder. The designation is stored in ASCII format with a maximum length of 18 characters. Unallocated characters are stored with the ASCII code 00h.

Direct read access to type name returns the defining values:

Table 62: Type name defining values

Defining value	Offset	Value
RID		083h
Resource name	0	"TYPECODE"
Data size	1	18
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	0Bh - string
Mandatory		yes

It should be noted that to read the whole code designation requires up to three "long message" transactions, as a "long message" can only contain 8 bytes of data.

When accessing offset basis, the OFF ADD characteristic gives the first character of the type name to be returned in the "long message".

Table 63: Read type name.

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									7C	83	00	00	01
Wait for FREL = 1													
Read	Characters 1 to 8 of the type name												
Write									7C	83	00	08	01
Wait for FREL = 1													
Read	Characters 9 to 16 of the type name												
Write									74	83	00	10	01
Wait for FREL = 1													
Read	Characters 17 to 18 of the type name												

8.4.5 Serial number

This resource indicates the serial number of the encoder. The serial number is stored in ASCII format with a maximum length of 10 characters. Unallocated characters are stored with the ASCII code 00h.

Direct read access to serial number returns the defining values:

Table 64: Serial number defining values

Defining value	Offset	Value
RID		084h
Resource name	0	"SERIALNO"
Data size	1	10
Read access level	2	0
Write access level	3	15
Time overrun	4	70

Defining value	Offset	Value
Data type	5	0Bh - string
Mandatory		yes

It should be noted that to read the whole serial number requires up to two "long message" transactions, as a "long message" can only contain 8 bytes of data.

When accessing offset basis, the OFF ADD characteristic gives the first character of the serial number to be returned in the "long message".

Table 65: Reading the serial number

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									7C	84	00	00	01
Wait for FREL = 1													
Read	Characters 1 to 8 of the serial number												
Write									74	84	00	08	01
Wait for FREL = 1													
Read	Characters 9 to 10 of the serial number												

8.4.6 Device version

This resource indicates the firmware and hardware version of the encoder. The firmware version is stored in ASCII format with a maximum length of 16 characters, the hardware version is in the same format with a maximum of 4 characters. Unallocated characters are stored with the ASCII code 00h.

Direct read access to device version returns the defining values:

Table 66: Device version defining values

Defining value	Offset	Value
RID		085h
Resource name	0	"FWREVNO"
Data size	1	20
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	0Bh - string
Mandatory		yes

It should be noted that to read the whole device version data requires up to three "long message" transactions, as a "long message" can only contain 8 bytes of data.

When accessing offset basis, the OFF ADD characteristic gives the first character of the device version to be returned in the "long message".

The device version is given in the following format:

Table 67: Definition of the device version

Byte	Description
0 to 15	ASCII characters of the firmware version
16 to 19	ASCII characters of the hardware version

Table 68: Reading the device version

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									7C	85	00	00	01
Wait for FREL = 1													
Read	Characters 1 to 8 of the firmware version												
Write									7C	85	00	08	01
Wait for FREL = 1													
Read	Characters 9 to 16 of the firmware version												
Write									78	85	00	10	01
Wait for FREL = 1													
Read	Characters 1 to 4 of the hardware version												

8.4.7 Firmware date

This resource indicates the firmware date of the encoder. The firmware date is stored in ASCII format with a maximum length of 8 characters.

Direct read access to firmware date returns the defining values:

Table 69: Firmware date defining values

Defining value	Offset	Value
RID		086h
Resource name	0	"FWDATE"
Data size	1	8
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	0Bh - string
Mandatory		yes

The firmware date is given in the following format:

Table 70: Firmware date definition

Byte	Value	Description
7/6	'00' to '99'	Firmware date year, i.e 20yy
5	'.'	Decimal point as separator
4/3	'01' to '12'	Firmware date month
2	'.'	Decimal point as separator
1/0	'01' to '31'	Firmware date day

For this resource, access to the offset basis is not meaningful as the resource data can be read using a "long message" transaction.

Table 71: Reading the firmware date

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									5C	86	00	00	01
Wait for FREL = 1													
Read	Firmware date "DD.MM.YY"												

8.4.8 EEPROM size

This resource indicates the total size of the non-volatile memory in the encoder available for storage of user data. The size of the EEPROM is given as an unsigned 16 bit value, which shows the number of bytes.

Direct read access to the EEPROM size returns the defining values:

Table 72: EEPROM size defining values

Defining value	Offset	Value
RID		087h
Resource name	0	"EESIZE"
Data size	1	2
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	04h – 16 Bit, unsigned
Mandatory		yes

For this resource, access to the offset basis is not meaningful as the resource data can be read using a "long message" transaction.

Table 73: Reading the EEPROM size

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									54	87	00	00	01
Wait for FREL = 1													
Read	EEPROM size												

8.5 Monitoring resources

The DSL motor feedback system monitoring resources indicate the current ambient values and their range limits as well as usage statistics and an error stack.

8.5.1 Temperature range

This resource indicates the minimum and maximum permitted values for the temperature of the DSL motor feedback system given in the product data sheet.

Direct read access to temperature range returns the defining values:

Table 74: Temperature range defining values

Defining value	Offset	Value
RID		0C0h
Resource name	0	"TEMPRNG"
Data size	1	4
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	08h – 16 bit, with sign
Mandatory		yes

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

Examples of temperature range values:

Table 75: Examples of temperature ranges

Temperature	Resource value (bin.)	Resource value (hex.)
20.0 °C	0000 0000 1100 1000b	00C8h
115.0 °C	0000 0100 0111 1110b	047Eh
-40.0 °C	1111 1110 0111 0000b	FE70h

The temperature range values are given in the following format:

Table 76: Temperature range definition

Byte	Value	Description
3/2	-2730 to 10000	Maximum permitted encoder temperature in 0.1 °C
1/0	-2730 to 10000	Minimum permitted encoder temperature in 0.1 °C

By accessing offset basis, only one of two temperature range values can be given.

Table 77: Selection of the temperature range offset

Offset value	Length of the message	Return values
0000h	4	Temperature range minimum and maximum values
0000h	2	Minimum temperature
0002h	2	Maximum temperature

Table 78: Reading the temperature range

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									58	C0	00	00	01
Wait for FREL = 1													
Read	Min. temp.		Max. temp.										

8.5.2 Temperature

This resource indicates the current temperature of the DSL motor feedback system. The temperature is measured once per second.

If this relates to the temperature of the DSL motor feedback system, an error is indicated if the measured value is outside one or other of the range limits (see chapter 7.6.3, error group 3, error number 0).

Direct read access to temperature returns the defining values:

Table 79: Temperature defining values

Defining value	Offset	Value
RID		0C1h
Resource name	0	"TEMPRTUR"
Data size	1	2
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	08h – 16 bit, with sign
Mandatory		yes

The temperature value is stored as a signed 16 bit two's complement. The temperature value units are tenths of degrees Celsius (0.1 °C).The temperature value is given in the following format:

Table 80: ***Temperature definition***

Byte	Value	Description
1/0	-2730 to 10000	Current temperature value in 0.1 °C

For this resource, access to the offset basis is not meaningful as the resource data can be read using a "long message" transaction.

Table 81: ***Reading the encoder temperature***

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									54	C1	00	00	01
Wait for FREL = 1													
Read	Encoder temperature												

8.5.3 Sensor monitor range

This resource indicates the minimum and maximum permitted values for product-specific sensor monitoring. The availability and type of sensor monitoring will be documented in the product data sheet.

Typical sensor monitoring includes measurement of the sender current for optical encoders.

Direct read access to sensor monitor range returns the defining values:

Defining value	Offset	Value
RID		0C2h

Defining value	Offset	Value
Resource name	0	product-specific
Data size	1	product-specific
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	product-specific
Mandatory		no

Resource name, data size, data type, and data format are specified in the product data sheet.

8.5.4 Sensor monitor

This resource indicates the current value of a product-specific sensor monitoring. The availability and type of sensor monitoring will be documented in the product data sheet.

Typical sensor monitoring includes measurement of the sender current for optical encoders.

The frequency of sensor monitoring is specified in the product data sheet. It is recommended to monitor long-term trends of the sensor monitor value against permitted minimum and maximum values. This allows determining if a product nears end of life-time and needs to be replaced.

If the measured sensor monitor value is outside either one of the range limits, an error is indicated (see chapter , error group 3, error number 1).

Direct read access to sensor monitor returns the defining values:

Table 82: Sensor monitor defining values

Defining value	Offset	Value
RID		0C3h
Resource name	0	product-specific
Data size	1	product-specific
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	product-specific
Mandatory		no

Resource name, data size, data type, and data format are specified in the product data sheet.

8.5.5 Supply voltage range

This resource indicates the minimum and maximum permitted values for the internal supply voltage for the DSL motor feedback system given in the product data sheet.

Direct read access to supply voltage range returns the defining values:

Table 83: Supply voltage range defining values

Defining value	Offset	Value
RID		0C4h
Resource name	0	"SUPRANGE"
Data size	1	4

Defining value	Offset	Value
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	04h – 16 Bit, unsigned
Mandatory		no

The values of the supply voltage range are stored as unsigned 16 bit values. The supply voltage units are 1 mV.

The values for the supply voltage range are given in the following format:

Table 84: Supply voltage range definition

Byte	Value	Description
3/2	0 to 65535	Maximum permitted internal supply voltage for the encoder in mV
1/0	0 to 65535	Minimum permitted internal supply voltage for the encoder in mV

By accessing offset basis, only one of two supply voltage range values can be given.

Table 85: Offset selection for the supply voltage range

Offset value	Length of the message	Return values
0000h	4	Minimum and maximum value of the supply voltage
0000h	2	Minimum supply voltage
0002h	2	Maximum supply voltage

Table 86: Reading the supply voltage range

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									58	C4	00	00	01
Wait for FREL = 1													
Read	Min. voltage		Max. voltage										

8.5.6 Supply voltage

This resource indicates the supply voltage of a DSL motor feedback system. The supply voltage is measured every 10 msec.

If the measured supply voltage is outside either one of the range limits, an error is indicated (see chapter , error group 3, error number 2).

Direct read access to supply voltage returns the defining values:

Table 87: Supply voltage defining values

Defining value	Offset	Value
RID		0C5h
Resource name	0	"SUPVOLT"
Data size	1	2
Read access level	2	0

Defining value	Offset	Value
Write access level	3	15
Time overrun	4	70
Data type	5	04h – 16 Bit, unsigned
Mandatory		no

The value of the supply voltage is stored as an unsigned 16 bit value. The supply voltage units are 1 mV.

The supply voltage value is given in the following format:

Table 88: Supply voltage definition

Byte	Value	Description
1/0	0 to 65535	Current encoder supply voltage in mV

For this resource, access to the offset basis is not meaningful as the resource data can be read using a "long message" transaction.

Table 89: Reading the supply voltage

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									54	C5	00	00	01
Wait for FREL = 1													
Read	Supply voltage												

8.5.7 Rotation speed range

This resource indicates the permitted maximum shaft rotation speed for rotary DSL motor feedback systems given in the product data sheet.

Direct read access to rotation speed range returns the defining values:

Table 90: Rotation speed range defining values

Defining value	Offset	Value
RID		0C6h
Resource name	0	"SPEEDRNG"
Data size	1	2
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	04h – 16 Bit, unsigned
Mandatory		no

The rotation speed range value is stored as a 16 bit unsigned value. The rotation speed value units are 1 rotation per minute (min^{-1}).

It should be noted that the current rotation speed value is given as a process value in the "process data channel" (see chapter 7.3).

The rotation speed range value is given in the following format:

Table 91: Rotation speed range definition

Byte	Value	Description
1/0	0 to 65535	Maximum permitted rotation speed of the encoder in min^{-1}

For this resource, access to the offset basis is not meaningful as the resource data can be read using a "long message" transaction.

Table 92: Reading the rotation speed range

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									54	C6	00	00	01
Wait for FREL = 1													
Read	Max. rotation speed												

8.5.8 Rotation speed

This resource indicates the shaft rotation speed of a rotary DSL motor feedback system. The rotation speed is measured once per second.

If the measured shaft rotation speed is outside either one of the range limits, an error is indicated (see table 41, error group 3, error number 3).

Direct read access to rotation speed returns the defining values:

Table 93: Rotation speed defining values

Defining value	Offset	Value
RID		0C7h
Resource name	0	"SPEED"
Data size	1	2
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	04h – 16 Bit, unsigned
Mandatory		no

The rotation speed value is stored as a 16 bit unsigned value. The rotation speed value units are 1 rotation per minute (min^{-1}).

It should be noted that the rotation speed value is given synchronously with the DSL measurements trigger signal as a process value in the "process data channel" (see chapter 7.3). Process data

The rotation speed value is given in the following format:

Table 94: Rotation speed definition

Byte	Value	Description
1/0	0 to 65535	Current rotation speed of the encoder in min^{-1}

For this resource, access to the offset basis is not meaningful as the resource data can be read using a "long message" transaction.

Table 95: Reading the rotation speed

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									54	C7	00	00	01
Wait for FREL = 1													
Read	Rotation speed												

8.5.9 Acceleration range

This resource indicates the permitted minimum and maximum shaft acceleration for rotary DSL motor feedback systems given in the product data sheet.

Direct read access to acceleration range returns the defining values:

Table 96: Acceleration range defining values

Defining value	Offset	Value
RID		0C8h
Resource name	0	"ACCRANGE"
Data size	1	2
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	04h – 16 Bit, unsigned
Mandatory		no

The value of the acceleration range is stored as an unsigned 16 bit value. The acceleration value units are 1000 rad/s^2.

It should be noted that the current acceleration value can be derived from the rotation speed process value in the "process data channel" (see chapter 7.3).

The acceleration range value is given in the following format:

Table 97: Acceleration range definition

Byte	Value	Description
1/0	0 to 65535	Maximum permitted rotational acceleration of the encoder in 1000 rad/s^2

For this resource, access to the offset basis is not meaningful as the resource data can be read using a "long message" transaction.

Table 98: Reading the acceleration range.

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									54	C8	00	00	01
Wait for FREL = 1													

Transaction	Register											
Read	Max. acceleration											

8.5.10 Lifetime

This resource indicates the operating time and the observed number of shaft rotations for a DSL motor feedback system. For safety variants, the remaining encoder task lifetime is also indicated.

Direct read access to lifetime returns the defining values:

Table 99: Lifetime defining values

Defining value	Offset	Value
RID		0CBh
Resource name	0	"LIFETIME"
Data size	1	8 (12 for safety variants)
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	05h – 32 bit, unsigned
Mandatory		no

The number of shaft rotations of the product is determined based on an average rotation speed. The integration time lasts one second.

The values for operating time, remaining task lifetime and number of shaft rotations are all stored as unsigned 32- bit values. The operating time units and the remaining task lifetime are 1 minute. The values are stored every 20 minutes in a non-volatile memory.

CAUTION
If the remaining task lifetime falls to 0, the encoder continuously issues error message 22 (safety error). If this is the case, the encoder must be replaced.

Examples of time values:

Table 100: Examples of lifetime

Duration	Resource value (bin.)	Resource value (hex.)
10 min	0000 0000 0000 0000 0000 0000 0000 1010b	0000 000Ah
200 hours	0000 0000 0000 0000 0010 1110 1110 0000b	0000 2EE0h
5 years	0000 0000 0010 1000 0001 1001 1010 0000b	0028 19A0h

The lifetime values are given in the following format:

Table 101: Lifetime definition

Byte	Value	Description
11 to 8	0 to 4294967295	Remaining task lifetime in minutes
7 to 4	0 to 4294967295	Number of shaft rotations
3 to 0	0 to 4294967295	Operating time in minutes

By accessing offset basis, only one of the lifetime values can be given.

Table 102: Selection of the lifetime offset

Offset value	Length of the message	Return values
0000h	4	Operating time

Offset value	Length of the message	Return values
0004h	4	Number of shaft rotations
0008h	4	Remaining task lifetime

Table 103: Reading the lifetime

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									7C	CB	00	00	01
Wait for FREL = 1													
Read	Operating time				Number of shaft rotations								

8.5.11 Error log

This resource returns stored DSL motor feedback system error messages. Direct read access to error protocol returns the defining values:

Table 104: Error protocol defining values

Defining value	Offset	Value
RID		0CCh
Resource name	0	"ERRORLOG"
Data size	1	16 per error protocol entry
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	20h – structure with 16 bytes
Mandatory		no

As soon as the encoder identifies an error, this error is indicated to the frequency inverter application (see chapter 7.5). In addition, errors are stored in the non-volatile memory of the DSL motor feedback system.

In addition, those errors identified when the establishment of a connection to the frequency inverter application failed are also stored in the error protocol. This resource provides an overview of these errors.

A DSL motor feedback system can store a set number of errors in its error protocol. If the total of errors recorded exceeds this number, the oldest entries are overwritten. The maximum number of error protocol entries can be found in the product data sheet.

NOTE
Depending on motor feedback system some faults might not be available for error log storage. This applies for faults that are diagnosed in hard realtime and directly transmitted on the DSL interface only.

NOTE
It is recommended to read out the additional error code in byte 14. This additional error code is product specific and supports the SICK diagnostic capabilities

All errors are recorded with time information, and several process and condition values from the time at which the error occurred.

Error protocol entries are each stored in 16 bytes.

The values for the error protocol are given in the following format:

Table 105: Error protocol entries definition

Byte	Value	Description
15	00 to FFh	Error code (see table 41)
14	00 to FFh	Additional error code
13/12	0	Reserved
11/10	0 to 65535	Rotation speed in min^{-1} during the error
9/8	0 to 65535	Internal supply voltage in mV during the error
7/6	0 to 65535	LED current in 0.1 mA during the error
5/4	-2730 to 10000	Temperature in 0.1 °C during the error
3 to 0	0 to 4294967295	Error time information (operating time or real-time stamp)

All error protocol entries are stored sequentially and are accessible by giving the individual values for the offset address.

To give a complete error entry, two "long message" transactions must be carried out as a message can only contain 8 bytes.

NOTE
It should be noted that the highest offset address that can be given depends on the maximum number of error protocol entries for the particular product.

Table 106: Offset selection for the error protocol

Offset value	Length of the message	Return values
00 00h	8	Number of stored error messages
00 01h	8	First part of the most recently occurring error
01 01h	8	Second part of the most recently occurring error
00 02h	8	First part of the error protocol entry no. 2
01 02h	8	Second part of the error protocol entry no. 2
...	8	...
00 xxh	8	First part of the oldest error protocol entry
01 xxh	8	Second part of the oldest error protocol entry

The higher value offset byte indicates whether the first part of the error message (00h) or the second part (01h) will be given.

The lower value offset byte indicates the desired error number.

Table 107: Reading the error protocol entries

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									7C	CC	00	01	01
Wait for FREL = 1													

Transaction	Register												
Read	Bytes 0 - 7 of the error protocol entry #1												
Write									7C	CC	01	01	01
Wait for FREL = 1													
Read	Bytes 8 - 15 of the error protocol entry #1												

8.5.12 Usage histogram

This resource gives histogram values of encoder parameters. The histogram values indicate how often a parameter value was measured during the lifetime of the encoder.

Direct read access to usage histogram returns the defining values:

Table 108: Usage histogram defining values

Defining value	Offset	Value
RID		0CDh
Resource name	0	"USAGEHIS"
Data size	1	4
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	20h – structure with 16 bytes
Mandatory		no

Recorded encoder parameters are measured at one minute intervals and stored every 20 minutes in a non-volatile memory.

The following table contains the encoder parameters that can be recorded in a histogram and stored. In addition, the range limits and the resolution of the histogram are given.

Table 109: Encoder parameter histograms definitions

Encoder parameters	Min. class	Max. class	Width of histogram class
Temperature	< -40 °C	>= 120 °C	10 °C
Sensor monitor	product-specific	product-specific	product-specific
Supply voltage	< 6.0 V	>= 14.0 V	1.0 V
Rotation speed	0 to 500 min^{-1}	>= 10,000 min^{-1}	500 min^{-1}

The values of the usage histograms are each stored in 4 bytes.

The values for the usage histograms are given in the following format:

Table 110: Value definitions in usage histograms

Byte	Value	Description
3/2/1	00 00 00h - FF FF FFh	Number of parameter values in the histogram class
0	00h - FFh	Identification of histogram class

The identification of the histogram class is transmitted in the offset value with an identification for the requested encoder parameter. The identification of the histogram class depends on the encoder parameter selected (see the following table).

Table 111: Histogram classes

Encoder parameters	Histogram class	Identification of histogram class
Temperature	< -40 °C	00h
	-40 to -30 °C	01h
	-30 to -20 °C	02h
	-20 to -10 °C	03h
	-10 to 0 °C	04h
	0 to 10 °C	05h
	10 to 20 °C	06h
	20 to 30 °C	07h
	30 to 40 °C	08h
	40 to 50 °C	09h
	50 to 60 °C	0Ah
	60 to 70 °C	0Bh
	70 to 80 °C	0Ch
	80 to 90 °C	0Dh
	90 to 100 °C	0Eh
	100 to 110 °C	0Fh
	110 to 120 °C	10h
	>= 120 °C	11h
Sensor monitor	product-specific, see product data sheet	product-specific, see product data sheet
Supply voltage	< 6.0 V	00h
	6.0 to 7.0 V	01h
	7.0 to 8.0 V	02h
	8.0 to 9.0 V	03h
	9.0 to 10.0 V	04h
	10.0 to 11.0 V	05h
	11.0 to 12.0 V	06h
	12.0 to 13.0 V	07h
	13.0 to 14.0 V	08h
	> 14.0 V	09h

Encoder parameters	Histogram class	Identification of histogram class
Rotation speed	0 to 500 min^{-1}	00h
	500 to 1000 min-1	01h
	1000 to 1500 min-1	02h
	1500 to 2000 min-1	03h
	2000 to 2500 min-1	04h
	2500 to 3000 min-1	05h
	3000 to 3500 min-1	06h
	3500 to 4000 min-1	07h
	4000 to 4500 min-1	08h
	4500 to 5000 min-1	09h
	5000 to 5500 min-1	0Ah
	5500 to 6000 min-1	0Bh
	6000 to 6500 min-1	0Ch
	6500 to 7000 min-1	0Dh
	7000 to 7500 min-1	0Eh
	7500 to 8000 min-1	0Fh
	8000 to 8500 min-1	10h
	8500 to 9000 min-1	11h
	9000 to 9500 min-1	12h
	9500 to 10,000 min-1	13h
	> 10,000 min-1	14
Code disk position	product-specific, see product data sheet	product-specific, see product data sheet

The offset value must be given in the following format:

Table 112: Selection of the histogram offset

Bits	Value	Definition
0 to 7	00h to FFh	Identification of histogram class
8 to 11	0h 1h 2h 3h 4h	Request temperature Request sensor monitor Request supply voltage Request rotation speed Request code disk position

Table 113: Reading histogram entries

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									78	CD	00	07	01
Wait for FREL = 1													
Read	07	Number of measurements 20 to 30 °C											
Write									78	CD	03	04	01
Wait for FREL = 1													

Transaction	Register										
Read	04	Number of measurements 2000 to 2500 rpm									

8.6 Code disk position range

This resource indicates the minimum and maximum permitted values for code disc position of bearing-less or kit encoders.The availability and type of code disc position monitoring will be documented in the product data sheet.

Direct read access to Code disc position range returns the defining values:

Table 114: Code disc position range defining values

Defining value	Offset	Value
RID		0D3h
Resource name	0	product-specific
Data size	1	product-specific
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	product-specific
Mandatory		no

Resource name, data size, data type, and data format are specified in the product data sheet.

8.7 Code disk position

This resource indicates the current value of the code disc position of bearing-less or kit encoders.

The resource can be used for monitoring correct assembly during mounting of the encoder to a motor. During life-time it also helps monitoring correct usage of the encoder, e.g. due to axial motor shaft movement over temperature.

Actual axial position of encoder. The axial position is categorized in 5 values:

- -2: rotor position in -Z axis exceeds boundary
- -1: rotor position in -Z axis tolerable
- 0: rotor position in Z axis ideal
- 1: rotor position in +Z axis tolerable
- 2: rotor position in +Z axis exceeds boundary

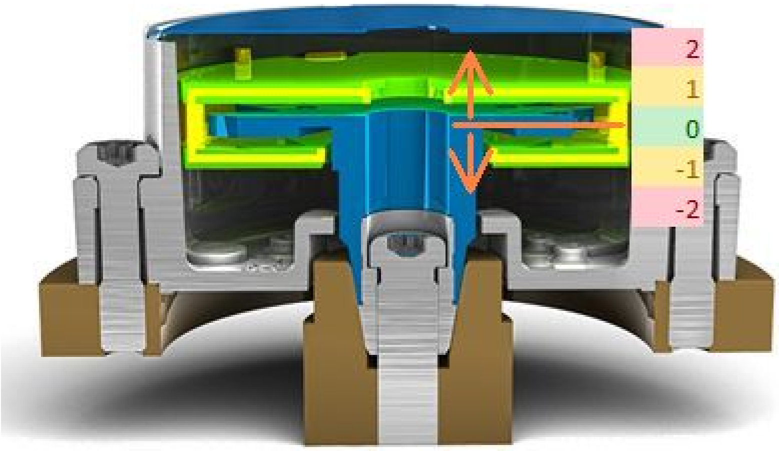


Figure 31: Code disc position

The availability and type of code disc position monitoring will be documented in the product data sheet.If the measured code disc position is outside either of the range limits, an error is indicated (see table 42, error group 3, error number 1).

Direct read access to Code disc position returns the defining values:

Table 115: Code disc position defining values

Defining value	Offset	Value
RID		0D4h
Resource name	0	product-specific
Data size	1	product-specific
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	product-specific
Mandatory		no

Resource name, data size, data type, and data format are specified in the product data sheet.

8.8 Administration resources

The administration resources of the DSL motor feedback system provide access to the encoder options settings.

8.8.1 Reset/shut-down

With this resource an encoder reset or shut-down can be executed. These sequences are required to validate certain changes, save encoder operation relevant data before shutting down or just simply reset the encoder.

Direct read access to reset/shut-down returns the defining values:

Table 116: Reset defining values

Defining value	Offset	Value
RID		100h
Resource name	0	"RESET"

Defining value	Offset	Value
Data size	1	0
Read access level	2	15
Write access level	3	0
Time overrun	4	150
Data type	5	01h - empty
Mandatory		yes

An indirect write access to this resource is required to trigger a reset or a shut-down of the DSL motor feedback system. The offset value allows selection of the desired function:

Table 117: Function selection

Value	Definition
0	Motor feedback system reset
1	Motor feedback system shut-down

In case of a reset, the firmware of the encoder is reset by initializing the encoder the same way as at switch-on (see chapter 7.1). Following a reset command, no report is returned. In case of a shut-down, the firmware of the encoder discontinues all communication and enters into an endless loop state. The encoder can be taken out from this state only by a power cycle. In both cases, all encoder operation relevant data are saved (e.g. lifetime information, usage histograms).

NOTE
It should be noted that in case of a reset, the initializing time of the encoder as well as the maximum resource time overrun need to be taken into account. This means, that the encoder will start communicating only after this time has passed.

NOTE
It should be noted that after shut-down, the encoder no longer reacts to any command until it has been switched off and on again.

NOTE
It should be noted that the shut-down function can be used to reliably store all lifetime and usage histogram data before a planned shut-down.

Table 118: Reset write command.

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									11	00	00	00	01
Wait for FREL = 1													

NOTE
This funtion must be finalized by a reset of the encoder (RID 100h).

8.8.2 Set position

Using this resource, an arbitrary position value can be allocated to the current mechanical shaft position and the current position offset can be read.

WARNING
With synchronous servo drives, the position information is used for motor commutation. The incorrect use of this resource can adversely affect the motor. This function should only be called up by the motor manufacturers.

Direct read access to set position returns the defining values:

Table 119: Set position defining values

Defining value	Offset	Value
RID		101h
Resource name	0	"SETPOS"
Data size	1	8
Read access level	2	0
Write access level	3	2
Time overrun	4	100
Data type	5	06h – 64 bit, unsigned
Mandatory		yes

WARNING
There is no synchronization for movements in progress. This function may only be used when the encoder shaft is stationary.

During a write access, the current position is set for the transmitted value. The position value to be allocated to the current shaft position is transmitted as an unsigned 40 bit value. Only values in the measurement range of the DSL motor feedback system are valid. During a read access, the offset value currently being used is transmitted in the same format.

The position value for this command must be given in the following format:

Table 120: Set position definition

Byte	Value	Description
4 to 0	00 0000 0000 to FF FFFF FFFFh	New position value for the current mechanical shaft position. The position value is right justified.

Table 121: Set write command position.

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write	Byte 0 to 4 of the target position					00	00	00	1D	01	00	00	01
Wait for FREL = 1													

WARNING
An encoder reset has to be carried out afterwards in order to validate the changes.

8.8.3 Set access level

This resource is used to set or read the encoder access level. The access level determines which functions are accessible for the user application. The access level required for each function is set out in the resources list (chapter 8.2).

Direct read access to Set access level returns the defining values:

Table 122: Set access level defining values

Defining value	Offset	Value
RID		104h
Resource name	0	"SETACCESS"
Data size	1	8
Read access level	2	0
Write access level	3	0
Time overrun	4	70
Data type	5	18h – structure with 8 bytes
Mandatory		yes

After switch-on or after a reset, the access level is always set to "0", i.e. to the lowest (public) access rights.

To alter the access level, the corresponding access key must be transmitted to the DSL encoder. The access level is retained until another level is set using this resource.

The table below sets out the access levels and the standard access keys available with which the appropriate level can be set.

Table 123: Access levels and standard access keys

Access level	Standard access key	Usage
0	No access key necessary	Publicly accessible system functions
1	31 31 31 31h	Protected system functions - "operator" level
2	32 32 32 32h	Protected system functions - "maintenance" level
3	33 33 33 33h	Protected system functions - "authorized client" level
4	34 34 34 34h	Protected system functions - "service" level

The access level is given in the following format:

Table 124: Set access level definition

Byte	Value	Description
7/6/5/4	0000 0000 to FFFF FFFFh	Access keys for the requested access level
3/2/1		Reserved for later use
0	00h to 04h	Requested access level

The currently set access level can be determined using a read access. This access level is returned in byte 0 of the long message.

For this resource, access to the offset base is not meaningful as the size of the resource data is smaller than the maximum for a "long message" transaction.

Table 125: Set access level (in this example: 01h with access key 31313131h)

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write	01	00	00	00	31	31	31	31	1D	04	00	00	01
Wait for FREL = 1													

Table 126: Reading the current access level (in this example: 00h).

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									55	04	00	00	01
Wait for FREL = 1													
Read	00	00											

8.8.4 Change access key

This resource is used to change the access key required to set the appropriate access level. The access level determines which functions are accessible for the user application. The access level required for each function is set out in the resources list (chapter 8.2).

Direct read access to change access key returns the defining values:

Table 127: Change access key defining values

Defining value	Offset	Value
RID		105h
Resource name	0	"CHNGKEY"
Data size	1	8
Read access level	2	15
Write access level	3	0
Time overrun	4	90
Data type	5	18h – structure with 8 bytes
Mandatory		yes

To change the access key, both the old and the new access keys for the target access level as well as the access level itself must be transmitted to the DSL encoder.

It should be noted that the access key for any level can be changed irrespective of the access level currently selected.

The access key is changed when data in the following format is transmitted:

Table 128: Change access key definition

Byte	Value	Description
7/6/5/4	0000 0000 to FFFF FFFFh	Old access key
3/2/1/0	0000 0000 to FFFF FFFFh	New access key

A read access to this resource is not possible.

The offset value indicates the target access level of the key change.

Table 129: Selection of access level

Offset value	Description
0 – 4	Target access level

Table 130: Changing the access key (in this example for access level 02h, changing from 31313131h to 12345678h)

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write	12	34	56	78	31	31	31	31	3D	05	02	00	01
Wait for FREL = 1													

8.8.5 User-defined warnings

This resource allows the user to program warnings to be set when thresholds are exceeded or certain bits of the DSL motor feedback system parameters change.

All user-defined thresholds or bit masks are checked once per second. The number of available user-defined warnings is set out in the product data sheet.

If a user-defined warning is triggered, this appears as a motor feedback system error message (see chapter) and is recorded in the error protocol (see chapter 8.5.11).

Direct read access to user-defined warnings returns the defining values:

Table 131: User-defined warnings defining values

Defining value	Offset	Value
RID		107h
Resource name	0	"UWARNING"
Data size	1	8
Read access level	2	0
Write access level	3	2
Time overrun	4	90
Data type	5	18h – structure with 8 bytes
Mandatory		no

A user-defined warning is configured by selecting the offset value for configuring the warning (see below):

Table 132: User-defined warning configuration bits

Byte	Value	Meaning
7/6/5/4		Reserved
3/2	0000 to FFFFh	Offset value of the monitored resource
0 (bit 1/0) / 1	000 to 3FFh	Resources index of the monitored resource
0 (bit 7/6/5)		Type of warning:
	0	Warning switched off
	1	Warning if monitored resource below threshold
	2	Warning if monitored resource above threshold
	3	Warning if monitored resource bit is deleted
	4	Warning if monitored resource bit is set

Byte	Value	Meaning
	5 to 7	Reserved
0 (bit 4/3/2)		Threshold data format:
	0	Not applicable
	1	16 bit, unsigned
	2	32 bit, unsigned
	3	64 bit, unsigned
	4	Not applicable
	5	16 bit, with sign
	6	32 bit, with sign
	7	64 bit, with sign

The threshold or the bit mask for the user-defined warning is set by selecting the offset value for thresholds (see below) and transmitting the threshold in the following format:

Table 133: Setting value of user-defined warning

Byte	Value	Meaning
7 to 0	Any	Threshold or bit mask for 64 bit value
3 to 0	Any	Threshold or bit mask for 32 bit value
1/0	Any	Threshold or bit mask for 16 bit value

Alternatively, the currently set values of a user-defined warning can be read in the same format.

The offset value indicates which user-defined warning is to be processed and whether the configuration bits or the value are affected.

Table 134: User-defined warning offset value

Offset value	Meaning
00h	Configuration bits for user-defined warning 1
01h	Configuration bits for user-defined warning 2
...	
0Fh	Configuration bits for user-defined warning 16
10h	Threshold/bit mask for user-defined warning 1
11h	Threshold/bit mask for user-defined warning 2
...	
1Fh	Threshold/bit mask for user-defined warning 16

Table 135: Writing a user-defined warning (here: switch on warning 1 if encoder temperature above 100 °C)

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write	34	C1	00	00	00	00	00	00	39	07	00	00	01
Wait for FREL = 1													
Transaction	**Register**												

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write	03	E8	00	00	00	00	00	00	39	07	00	10	01
Wait for FREL = 1													

8.8.6 Factory settings

This resource allows all DSL motor feedback system user-defined settings to be reset to the factory settings.

The following values are reset by this command:

- Position offset (see chapter 8.8.2)
- Changed access key (see chapter 8.8.4)
- User-defined warnings (see chapter 8.8.5)
- Counter (see chapter 8.9.1)
- All user files (see chapter 8.10)
- User settings for SHub® I/Os (see chapter 8.11.2)

WARNING

The "factory settings" command deletes all user-defined settings and data. Use this function with care.

NOTE

It should be noted that lifetime information, the error protocol and the usage histogram are not affected by this command.

Direct read access to factory settings returns the defining values:

Table 136: Factory settings defining values

Defining value	Offset	Value
RID		108h
Resource name	0	"FACRESET"
Data size	1	8
Read access level	2	15
Write access level	3	2
Time overrun	4	255
Data type	5	0Bh - string
Mandatory		no

To revert to factory settings, a write command with specified code word must be sent to this resource.

Table 137: Factory settings definition

Byte	Value	Meaning
0 to 7	"RESETALL"	Code word to revert to factory settings, including error protocol and usage histogram.

A read access to this resource is not possible.

Table 138: ***Factory settings***

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write	52	45	53	45	54	41	4C	4C	1D	08	00	00	01
Wait for FREL = 1													

8.8.7 User-defined encoder index

This resource is used to set or read the user-defined encoder index. This index can be freely programmed by the user and is indicated when the encoder is switched on in the `ENC_ID` register (see chapter 6.3.11).

The user-defined encoder index can be used to distinguish between several DSL Master instances occurring in an FPGA. This is required for safety-related use.

Direct read access to user-defined encoder index returns the defining values:

Table 139: User-defined encoder index defining values

Defining value	Offset	Value
RID		109h
Resource name	0	"ENCIDENT"
Data size	1	2
Read access level	2	0
Write access level	3	3
Time overrun	4	90
Data type	5	04h – 16 Bit, unsigned
Mandatory		yes

The user-defined encoder index can be between 0 and 15. Inputting a higher value will cause an error message.

The user-defined encoder index is given in the following format:

Table 140: User-defined encoder index definition

Byte	Value	Description
7 to 2		Reserved for later use
1/0	0 to 15 Other values	Requested user-defined encoder index Reserved for later use

A previously set user-defined encoder index can be identified using a read access. The default value is "0".

For this resource, access to the offset base is not meaningful as the size of the resource data is smaller than the maximum for a "long message" transaction.

Table 141: Reading the current user-defined encoder index (in this example: 00h)

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									55	09	00	00	01

Transaction	Register												
Wait for FREL = 1													
Read	00	00											

NOTE
This funtion must be finalized by a reset of the encoder (RID 100h).

8.8.8 Position filter setting

This resource is used to adjust or read the DSL motor feedback system position filter setting.

NOTE
Per default all DSL motor feedback systems are set to a maximum dynamic response. Changing the filter value can change the drive feedback loop characteristic significantly. Please consult SICK regarding possible use and format of this functionality.

The position filter characteristic depends on individual encoder types and is specified in the related data sheet.

Direct read access to position filter setting returns the defining values:

Table 142: Position filter setting defining values

Defining value	Offset	Value
RID		10Ah
Resource name	0	"POSFILT"
Data size	1	4
Read access level	2	0
Write access level	3	3
Time overrun	4	90
Data type	5	05h – 32 bit, unsigned
Mandatory		no

The position filter is set in the following format:

Table 143: Position filter definition

Byte	Value	Description
3 to 0	3000 to 37500	Mechanical filter limit frequency, measured in rotations per minute (rpm)

A previously set position filter can be identified using a read access.

For this resource, access to the offset base is not meaningful as the size of the resource data is smaller than the maximum for a "long message" transaction.

Table 144: Reading the current position filter (in this example: 35000)

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									59	0A	00	00	01
Wait for FREL = 1													
Read	00	00	88	B8									

8.9 Counter resources

The counter installed in the HIPERFACE DSL® motor feedback system is a 32 bit counter for user purposes that can be incremented as required. The counter can be read, incremented and reset.

8.9.1 Read counter

This resource indicates the value of a user-defined counter. The counter value is given as a 32 bit unsigned value.

Direct read access to read counter returns the defining values:

Table 145: Read counter defining values

Defining value	Offset	Value
RID		120h
Resource name	0	"READCNT"
Data size	1	4
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	05h – 32 bit, unsigned
Mandatory		no

The counter value is given in the following format:

Table 146: Read counter definition

Byte	Value	Description
3/2/1/0	0000 0000 to FFFF FFFFh	Value of the user-defined counter

For this resource, access to the offset basis is not meaningful as the resource data can be read using a "long message" transaction.

Table 147: Reading the counter

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									59	20	00	00	01
Wait for FREL = 1													
Read	Counter (32 bit)												

8.9.2 Increment counter

This resource increments the user-defined 32 bit counter. If the incrementation causes an overrun of the counter, error message 35 appears (see chapter) and the value of the counter remains at the maximum value.

Direct read access to increment counter returns the defining values:

Table 148: Increment counter defining values

Defining value	Offset	Value
RID		121h
Resource name	0	"INCCOUNT"

Defining value	Offset	Value
Data size	1	0
Read access level	2	15
Write access level	3	0
Time overrun	4	90
Data type	5	01h - empty
Mandatory		no

The incrementation is carried out using a write command to this resource that contains no data (length of the long message = 0).

Table 149: Command to increment the counter

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									11	21	00	00	01
Wait for FREL = 1													

8.9.3 Reset counter

This resource carries out a reset of the user-defined 32 bit counter. Direct read access to reset counter returns the defining values:

Table 150: Reset counter defining values

Defining value	Offset	Value
RID		122h
Resource name	0	"RESETCNT"
Data size	1	0
Read access level	2	15
Write access level	3	1
Time overrun	4	105
Data type	5	01h - empty
Mandatory		no

The reset is carried out using a write command to this resource that contains no data (length of the long message = 0).

Table 151: Command to reset the counter

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									11	22	00	00	01
Wait for FREL = 1													
Read	00	00											

8.10 Data storage resources

The user has access to user-defined files to be stored for miscellaneous purposes via the DSL motor feedback system data storage resources.

User data is stored in the non-volatile memory (EEPROM) and protected automatically by CRC checksums. The CRC mechanism provides the user with a very high level of reliability for error detection in relation to the storage of user data.

The following figure contains workflows for handling data storage. Each step represents an individual resource access (long message).

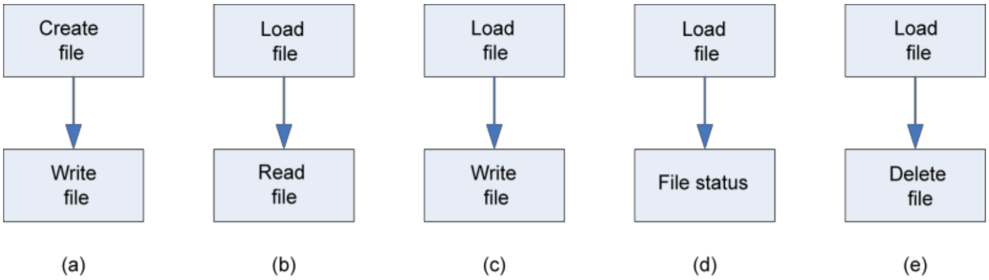


Figure 32: Workflows for data storage

(a) Writing to a new file, (b) Reading from a file, (c) Writing to an existing file, Polling the status of an existing file, (e) Deleting a file

8.10.1 Load file

To be able to access an existing file, it must first be loaded using this resource. Direct read access to load file returns the defining values:

Table 152: Load file defined values

Defining value	Offset	Value
RID		130h
Resource name	0	"LOADFILE"
Data size	1	8
Read access level	2	15
Write access level	3	1
Time overrun	4	130
Data type	5	0Bh - string
Mandatory		yes

It should be noted that only one file can be loaded at a time. When loading a new file, any hitherto loaded file is discarded.

A file remains loaded until another file is loaded or the DSL motor feedback system is reset or shut down.

A file is specified with its file name that is transmitted to the long message data buffer. If the file name is unknown, the "directory" resource (see chapter 8.10.5) can be used to search for existing files.

The file name can be up to 8 bytes long. Each byte represents one ASCII character. The end of the file name (when less than 8 bytes) is indicated by the character "\0" (00h). It should be noted that upper or lower case characters are valid.

A file can only be loaded if the currently set access level (see chapter 8.8.3) permits the reading or writing of a file. Access rights are determined when a file is created or changed (see chapter 8.10.4).

Table 153: Load file definition

Byte	Value	Description
0 to 7	Variable	Name of the file to be loaded

Offset-based access enables it to be determined whether or not a verification of the checksums is carried out by the DSL motor feedback system when a file is loaded.

If an error is detected when the verification is carried out, the motor feedback system responds to the "long message" with an error message (4315h, see chapter 7.6.6). If the verification is not carried out, it is not certain that subsequent read access to the loaded file will produce valid data.

Table 154: Selection of the offset for load file

Offset value	Description
0000h	Checksums are verified
0010h	Checksums are not verified
Other values	Reserved

Table 155: Example of loading a file (in this example: Loading a file with the name "FILE1")

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write	F	I	L	E	1	00	00	00	1D	30	00	00	01
Wait for FREL = 1													
Read													

8.10.2 Read/write file

Read and write access to a user file is possible via this resource. Direct read access to read/write file returns the defining values:

Table 156: Read/write file defining values

Defining value	Offset	Value
RID		131h
Resource name	0	"RWFILE"
Data size	1	8 (Total size depends on file size)
Read access level	2	0 (User determines actual access level)
Write access level	3	0 (User determines actual access level)
Time overrun	4	140
Data type	5	0Bh - string
Mandatory		yes

Before a file can be read or written to, the file must be loaded (see chapter 8.10.1).

Read or write procedures can be carried out by access to any addresses within the file. If an address given for reading causes the file size to be exceeded, an error message is returned. If an address given for writing causes the file size to be exceeded, the file is automatically enlarged. The largest address allowed for this attachment of data is the size of the file.

If the remaining EEPROM memory space is insufficient to accept the enlarged file, access is stopped and an error message is returned (4314h, see chapter 7.6.6).

By setting the length value for a long message, 2, 4 or 8 bytes can be read, or written in a long message.

The data from a read access or the data for a write access is stored in the long message buffer.

Table 157: Definition of reading and writing a file

Byte	Value	Description
7 to 0	Variable	Data from, or for a file

The offset value indicates the target address for the read or write access. It should be noted that the files may be a maximum of 32768 bytes in size.

Table 158: Offset value for reading or writing a file

Offset value	Description
0 to 32767	Address for the read or write access

Table 159: Reading or writing to a file (in this example: write 8 bytes to address 0033h)

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write	11	22	33	44	55	66	77	88	3D	31	00	33	01
Wait for FREL = 1													
Read													

8.10.3 File status

This resource returns the status of the currently loaded file (see chapter 8.10.1). Direct read access to file status returns the defining values:

Table 160: File status defining values

Defining value	Offset	Value
RID		132h
Resource name	0	"FILESTAT"
Data size	1	4
Read access level	2	0
Write access level	3	15
Time overrun	4	70
Data type	5	14h – structure with 4 bytes
Mandatory		yes

A read access to "file status" returns the file access rights and the size of the file.

The file status is given in the following format:

Table 161: Definition of reading and writing a file

Byte	Value	Description
3/2	0000 to FFFFh	File size in bytes
1		Reserved for later use
0, bits 7 to 4		Write access rights

Byte	Value	Description
	0	Public
	1	Operator
	2	Maintenance
	3	Authorized client
	4	Service
	5 – 14	Reserved for later use
	15	No write operation permitted
0, bits 3 to 0		Read access rights
	0	Public
	1	Operator
	2	Maintenance
	3	Authorized client
	4	Service
	5 – 14	Reserved for later use
	15	No read operation permitted

For this resource, access to the offset basis is not meaningful as the resource data can be read using a "long message" transaction.

Table 162: File status (in this example: File with read access 0, write access 2, file size 53 bytes)

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									59	32	00	00	01
Wait for FREL = 1													
Read	20	00	Buffer2 = 00h	Buffer3 = 35h									

8.10.4 Create/delete/change file

This resource is used for the creation, changing or deletion of a user file. Direct read access to create/delete/change file returns the defining values:

Table 163: Create/delete/change file defining values

Defining value	Offset	Value
RID		133h
Resource name	0	"MAKEFILE"
Data size	1	8
Read access level	2	15
Write access level	3	0 (User determines actual access level)
Time overrun	4	130
Data type	5	0Bh - string
Mandatory		yes

A user file must have been previously created before it can be loaded, written to or read from.

Before a file can be changed or deleted, the file must be loaded (see chapter 8.10.1).

To **create a file**, the name must be set in the long message buffer. Unallocated characters of the file name are set to "00h". If there is a user file with the name given already present, the procedure to create the file is canceled with an error message. The name may only consist of printable ASCII characters (20h - 7Eh). The name must be at least one character long.

Table 164: Create file definition

Byte	Value	Description
7 to 0	Variable	Name of the file to be created

The offset value is also used to create a file.

Table 165: Offset value for creating a file

Bits	Value	Definition
14 – 10		Reserved for later use
9 – 8	11b	Create file
7 – 4		Write access rights
	0	Public
	1	Operator
	2	Maintenance
	3	Authorized client
	4	Service
	5 – 14	Reserved for later use
	15	No write operation permitted
3 – 0		Read access rights
	0	Public
	1	Operator
	2	Maintenance
	3	Authorized client
	4	Service
	5 – 14	Reserved for later use
	15	No read operation permitted

To **change a file**, only the offset value is used.

Table 166: Offset value for changing a file

Bits	Value	Definition
14 – 10		Reserved for later use
9 – 8	01b	Change file
7 – 4		Write access rights
	0	Public
	1	Operator
	2	Maintenance
	3	Authorized client
	4	Service
	5 – 14	Reserved for later use
	15	No write operation permitted

Bits	Value	Definition
3 – 0		Read access rights
	0	Public
	1	Operator
	2	Maintenance
	3	Authorized client
	4	Service
	5 – 14	Reserved for later use
	15	No read operation permitted

To **delete a file**, the file name (of the currently loaded file) must be set in the long message buffer.

Table 167: Delete file definition

Byte	Value	Description
7 to 0	Variable	Name of the file to be deleted

The offset value is also used to delete a file.

Table 168: Offset value for deleting a file

Bits	Value	Definition
14 – 10		Reserved for later use
9 – 8	00b	Delete file
7 – 0		Reserved for later use

Table 169: Creating a file (in the example: Creation of a file with the name "FILE1", read access 0, write access 1)

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write	F	I	L	E	1	00	00	00	3D	33	03	10	01
Wait for FREL = 1													

8.10.5 Directory

When this resource is accessed, a list of the existing user files is returned. Direct read access to directory returns the defining values:

Table 170: Directory defining values

Defining value	Offset	Value
RID		134h
Resource name	0	"DIR"
Data size	1	8
Read access level	2	0
Write access level	3	15
Time overrun	4	130
Data type	5	18h – structure with 8 bytes
Mandatory		yes

In "directory", only those files are listed that are accessible at the access level set (read or write access).

In addition, by accessing "directory", the current size of the filled and empty user stores can be read.

It should be noted that due to the file header, the user files normally fill more physical stores than their pure data content.

The type of data required by the user is set in the offset value during read access to this resource.

Table 171: Offset value for "directory"

Bits	Value	Definition
14 – 8		Reserved for later use
7 – 0	00h	Return number of files as well as filled and empty
	01h	user stores
	02h	Return name of first user file
		Return name of second user file
	…	…
	FFh	Return name of 255th user file

The "directory" basic data (offset = 00h) is returned in the long message buffer as follows:

Table 172: Definition of "directory" ("directory" basic data)

Byte	Value	Description
7/6		Reserved for later use
5/4	0 – 65535	Number of filled bytes in the user store
3/2	0 – 65535	Number of empty bytes in the user store
1		Reserved for later use
0	0 – 255	Number of user files

The data from the user files (offset > 00h) is returned in the long message buffer as follows:

Table 173: Definition of "directory" (data from user files)

Byte	Value	Description
7 – 0	Variable	File name

Table 174: "Directory" (in this example: Read directory basic data - 2 user files, 123h bytes filled, 1E40h bytes empty)

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write									7D	34	00	00	01
Wait for FREL = 1													
Read	# file 02h	00	Empty store 1E40h		Used store 0123h		00	00					

8.11 SensorHub resources

SensorHub resources refer to additional external sensors that are connected to the motor feedback system.

The actual connectivity for external sensors depends on the individual product variant and is specified in the product data sheet.

Connectivity for external sensors is divided into two categories:

- Simple I/Os (inputs, outputs) are connected directly to a suitable DSL motor feedback system. Included in this category, for example, are temperature sensors, temperature switches or digital I/Os. Motor feedback systems with simple I/O connections are generally standard products.
- Enhanced sensors are connected to an external SensorHub component, which itself has a defined interface to the DSL motor feedback system. This architecture is used if several external sensors, or sensors with complex interfaces, are to be connected (e.g. torque or acceleration sensors). SensorHub components are normally customer-specific and are developed in collaboration with SICK.

The following figures give block diagrams for these scenarios:

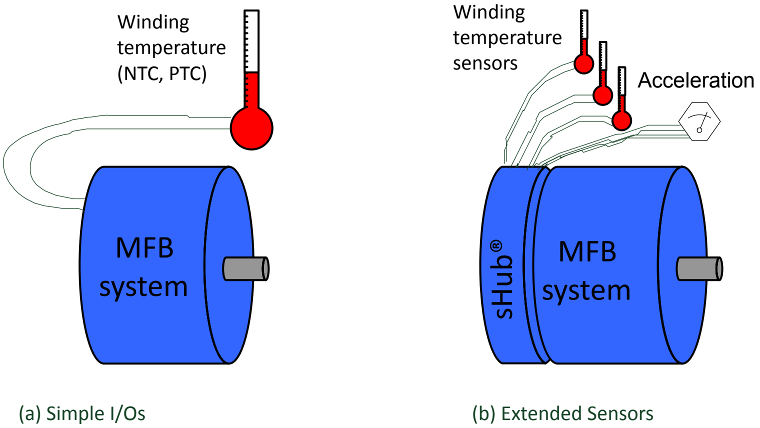


Figure 33: sHub® categories

8.11.1 Access simple I/Os

This resource enables access to simple I/Os connected directly to the motor feedback system.

Direct read access to access simple I/Os returns the defining values:

Table 175: Access simple I/Os defining values

Defining value	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
Data type	5	14h – 4 byte structure
Mandatory		no

The availability of read or write access depends on the product variant. Access to an input or output in a non-specific direction produces an error message (see chapter 7.6.4, error 41h).

In general, simple I/Os can carry out one of the following functions:

Table 176: Functions of simple I/Os

Direction	Signal type	Example
Input	Digital	Switch
Output	Digital	Braking control
Input	Analog	Temperature sensor

The model name, number and measurement characteristics of simple I/Os for any product variant are specified in the product data sheet. For analog inputs, this also lists the data format and the units of the values measured.

Signals for digital I/Os are specified as follows:

Table 177: Definition of access simple I/Os

Byte	Value	Description
3	00h 01h 02h to FFh	Set to 0 (output) / Value = 0 (input) Set to 1 (output) / Value = 1 (input) Reserved
0 to 2		Reserved

The offset value gives the I/O number to be accessed. It should be noted that the number of I/Os and the associated I/O numbers are specified in the product data sheet.

Table 178: Offset value for access simple I/Os

Offset value	Description
0 to 127	I/O number

Table 179: "Simple access I/Os" (here: Set digital output with I/O number #0)

Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write	00	00	00	01	00	00	00	00	3A	00	00	00	01
Wait for FREL = 1													

8.11.2 Manage simple I/Os

This resource enables access to the management functions for simple I/Os connected to the motor feedback system.

Direct read access to manage simple I/Os returns the defining values:

Table 180: Manage simple I/Os defining values

Defining value	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	90
Data type	5	14h – 4 byte structure
Mandatory		no

The availability of this function depends on the product variant and is specified in the product data sheet.

The offset value determines the requested management function. It should be noted that any additional management functions are specified in the product data sheet.

Table 181: Offset value for manage simple I/Os

Offset value	Description
0	Input filter I/O 0
1	Input filter I/O 1
2 to 32767	Reserved

The input filter function enables the user to set the low pass characteristic of an analog input. The value of 1 to 100 specifies as a percentage (%) the weighting of new measured values to previously averaged measurements. Examples:

- 100 indicates that there was no filtering.
- 50 indicates that each new measurement will be calculated with a weighting of 50:50 against previous measurements.
- 1 indicates that each new measurement will be calculated with a weighting of 1% against previous measurements.

The values for the input filter functions are determined as follows:

Table 182: Definition of manage simple I/Os (input filter)

Byte	Value	Description
0	0 1 to 100 101 to 255	Reserved Filter characteristics of the analog input Reserved
1 to 3		Reserved

Table 183: "Manage simple I/Os" (here: Set filter value 50 for analog input with I/O number #0)

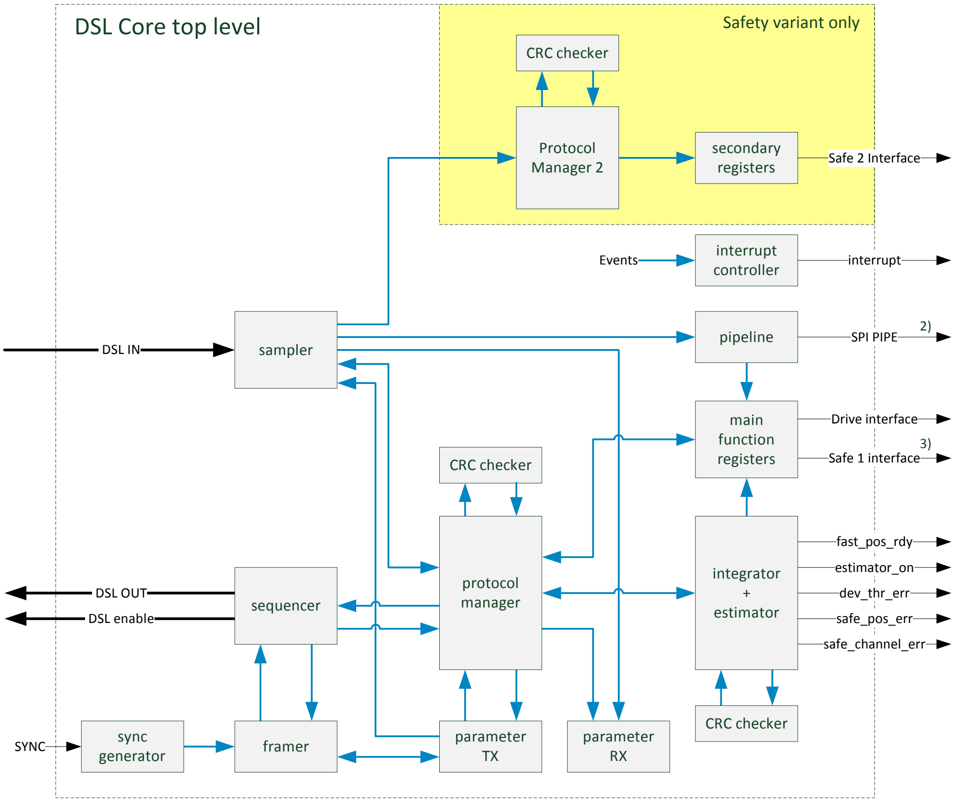
Transaction	Register												
	PC_BUFFER0	PC_BUFFER1	PC_BUFFER2	PC_BUFFER3	PC_BUFFER4	PC_BUFFER5	PC_BUFFER6	PC_BUFFER7	PC_ADD_H	PC_ADD_L	PC_OFF_H	PC_OFF_L	PC_CTRL
Write	32	00	00	00	00	00	00	00	3A	01	00	00	01
Wait for FREL = 1													

9 FPGA IP-Core

The HIPERFACE DSL® interface is installed in the frequency inverter system via a special protocol logic circuit, known as the DSL Master. The circuit is installed in an FPGA component and is supplied as an Intellectual Property Core (IP Core). The IP Core of the DSL master is supplied in a form such that it can be freely connected within the FPGA. If there is sufficient space within the FPGA used, the DSL Master can be installed in the same component as the frequency inverter application.

To be able to join different components to the IP Core, e.g. internal FPGA buses, various open-source interface blocks are supplied with the IP Core.

figure 34 shows the block circuit diagram of the DSL Master circuit without the interface blocks. Signal characteristics are listed in table 184, functional characteristics in table 185.



2) Optional interface, SPI only

3) Interface only available in Safety variant

Figure 34: Block circuit diagram of the DSL Master IP Core

Table 184: Signal characteristics of the DSL Master IP Core

Signal characteristics	
IP Core clock – "clk"	
Frequency	75.0 MHz
Frequency tolerance	±100 ppm
IP Core reset – "rst"	High active
Minimum reset duration after switch on/loading	20 ns
HIPERFACE DSL® interface	

"dsl_out" "dsl_in" "dsl_en"	Based on RS-485 specification, see application schematics
Typical signal transmission rate	9,375,000 baud
Drive interface	
"online_status_d(15:0)" "hostd_a(6:0)" "hostd_di(7:0)" "hostd_do(7:0)" "hostd_r" "hostd_w" "hostd_f"	drive interface signals
SPI PIPE interface - optional, configurable by the user	
"pipipe_clk" "pipipe_miso" "pipipe_ss"	Based on SPI specification
Maximum SPI clock	10 MHz
Control signals	
Digital input – "sync"	Synchronization to drive clock
Sync signal cycle time	12.2 … 1950 µs
Minimum sync signal duration	40 ns
Maximum jitter sync frequency	± 2 system clock cycles, 26 ns
Digital output – "sync_locked"	Drive clock synchronization indicator
Digital output – "interrupt"	Interrupt configurable by the user, high active
Digital output – "link"	DSL interface indicator, high active
Digital output – "fast_pos_rdy"	Indicator fast position value availability
Digital output – "dev_thr_err"	Indicator position estimator deviation threshold crossed
Digital input – "bigend"	Byte sequence selection for register addresses
Test signals	
Digital output “estimator_on”	Indicator for active position estimator
Digital output “safe_channel_err”	Indicator for safety frame transmission errors
Digital output “safe_pos_err”	Indicator for safe position update errors
Digital output “acceleration_err”	Indicator for transmission error of fast position
Digital output “acc_thr_err”	Indicator for crossing of fast position error counter threshold
Digital output “encoding_err”	Indicator for 8b/10b encoding transmission fault

Table 185: Functional characteristics of the DSL Master IP Core

Parameter	Value			Units	Remarks
	Minimum	Typical	Maximum		
System clock	74.9925	75.0000	75.0075	MHz	± 100 ppm
Characteristics of the interface					
Wire transmission rate		9.375		MBd	

Parameter	Value			Units	Remarks
	Minimum	Typical	Maximum		
Reset duration	0.02	0.06		µs	Reset is High active
Recovery time following communications failure			727	µs	
Characteristics of the motor feedback system					
Position resolution per revolution		23	40	Bit	The total can be a maximum of 40 bit
Number of resolved revolutions		16	40	Bit	
Rotation speed			262,000	rad/s	24 bit/rotation
Acceleration			670,000	rad/s^2	24 bit/rotation
Dead time		10.5		µs	SYNC trigger to position register update
Sampling latency			1	µs	SYNC trigger to position sampling time
Characteristics of the host interface					
Cycle time of the frequency inverter	12.1		1,950	µs	In SYNC mode
Packet cycle time	12.1		27	µs	In SYNC mode
Packet cycle time		11.52		µs	Infree-running mode
Duration of the SYNC signal	0.04			µs	The SYNC signal must be inactive for at least 0.04 µs per cycle.
SYNC signal jitter			26	ns	±2 System clock cycles
Characteristics of the SPI PIPE interface					
Clock of SPI PIPE			10	MHz	
Characteristics of the parameter channel					
Theoretical transmission rate	166		334	kBd	
Duration of access to the communications resource	167		1,100	µs	“short message”
Duration of access to the encoder resource		75		ms	“long message”, including encoder processing time
Characteristics of the SensorHub channel					
Transmission rate	334		669	kBd	

The following table contains a description of the pin functions of the DSL Master IP Core. Pin functions

Table 186: Pin functions of the IP Core

Pin description	Type	Function	Note
rst	Input	Master reset (High active)	
clk	Input	Clock input	
sync	Input	Drive cycle for position sampling trigger	

bigend	Input	Byte sequence selection register addresses	
interrupt	Output	Configurable interrupt	
link	Output	Connection display	
fast_pos_rdy	Output	Indicator fast position value availability	
sync_locked	Output	Drive cycle indicator	
online_status_d(15:0)	Output	IP-Core status bits	
hostd_a(6:0)	Input	Address bus	
hostd_di(7:0)	Input	Databus input	
hostd_do(7:0)	Output	Databus output	
hostd_r	Input	Selection read access	
hostd_w	Input	Selection write access	
hostd_f	Input	Selection freeze register	
aux_signals(4:0)	Output	Interface relevant internal signals	
sample	Output	Test signal line sampler	
estimator_on	Output	Position estimator indicator	
safe_channel_err	Output	Safe position channel error indicator	
safe_pos_error	Output	Safe position update error indicator	
acceleration_err	Output	Fast position transmission fault indicator	
acc_thr_err	Output	Fast position error counter indicator	
encoding_err	Output	Encoding fault indicator	
dev_thr_err	Output	Error signal “Max. estimated position deviation”	
spipipe_ss	Input	Selection SensorHub-SPI	
spipipe_clk	Input	Clock for SensorHub SPI	
spipipe_miso	Output	SensorHub SPI, master input data/ slave output data	
dsl_in	Input	DSL link, input data	
dsl_out	Output	DSL link, output data	
dsl_en	Output	DSL link transceiver, activation	

9.1 Interface blocks

Various interface blocks for the IP Core allow simpler access for differing drive architectures. SICK provides two different interface blocks as open-source VHDL modules. This enables individual modifications and adaptations. There are examples for a parallel or serial interface.

This section describes the connections between the interface blocks and the IP Core.

CAUTION
If interface blocks are altered or self-created, the installer must pay attention to the safety measures and processes when doing so. It is recommended that the interface blocks for Safe 1 and Safe 2 interface should be realized by using the SPI interface.

The figure below shows possible combinations of interface blocks.

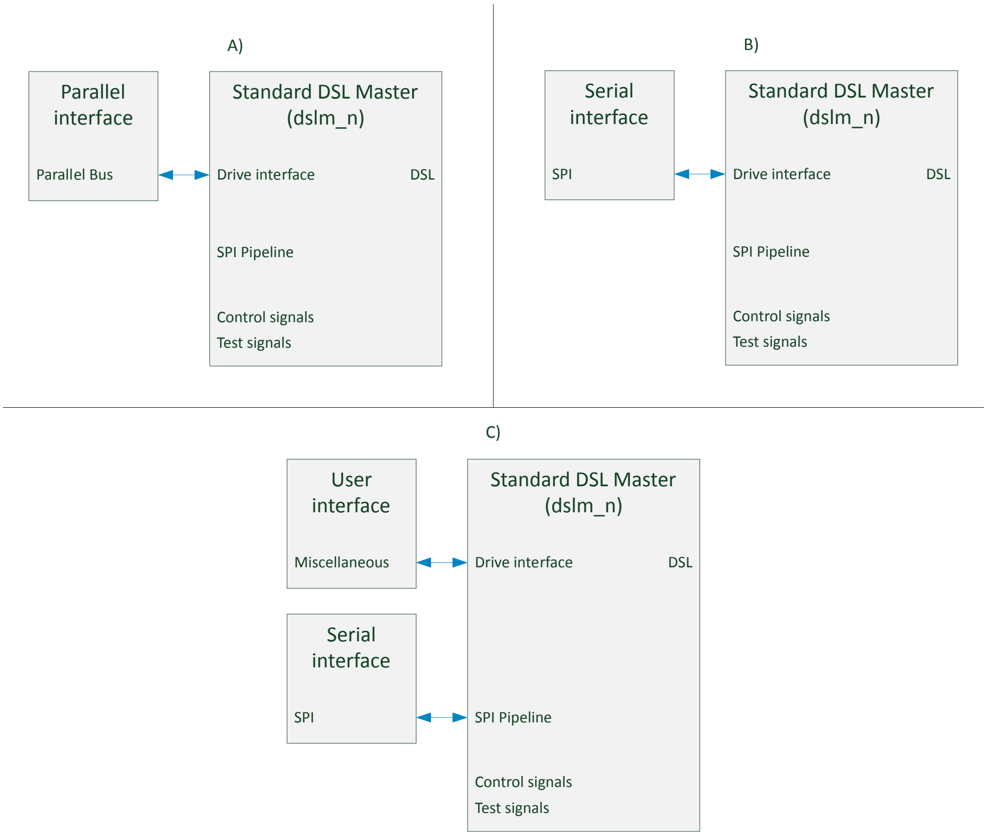


Figure 35: Combination examples of interface blocks

9.2 Serial interface block

As an example, a serial interface block is supplied together with the IP Core. In this example, a Full Duplex "Serial Peripheral Interface (SPI)" is installed.

The figure and table below show the interface signals.

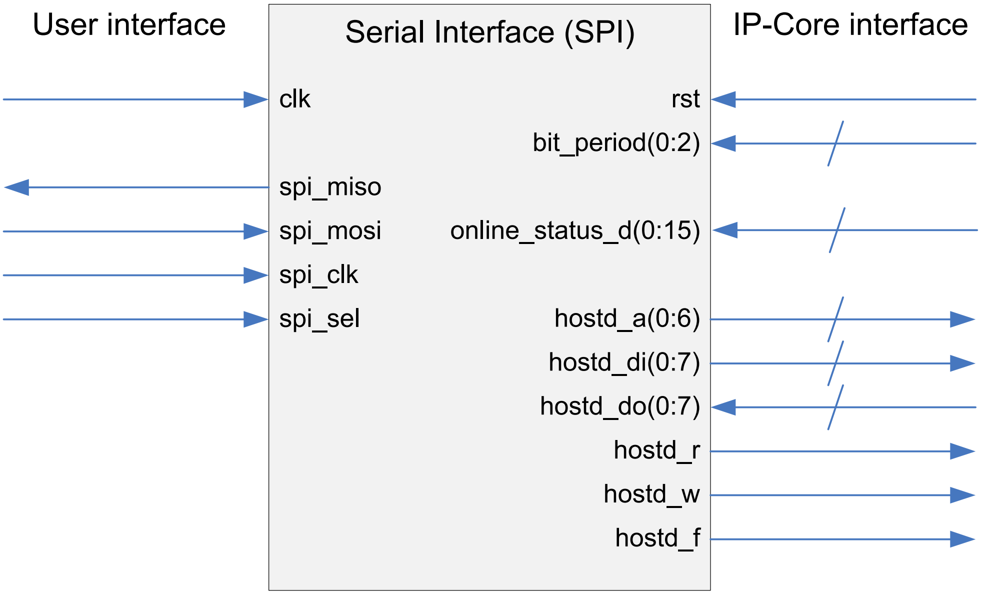


Figure 36: Serial interface block signals

Table 187: Serial interface block signals

Pin name	Model name	Function	Note
Host interface			
clk	Input	Clock input	
spi_miso	output	SPI data output	
spi_mosi	Input	SPI data input	
spi_clk	Input	SPI clock	
spi_sel	Input	SPI selection	
IP Core interface			
rst	Input	Internal reset	Connect to IUO(1) of the IP Core
bit_period(2:0)	Input	Internal state machine	Connect to IUO(4:2) of the IP Core
online_status_d(15:0)	Input	Internal status IP Core	
hostd_a(6:0)	output	Register block address bus	
hostd_di(7:0)	output	Data bus interface to core	
hostd_do(7:0)	Input	Data bus core to interface	
hostd_r	output	Read access requirement	
hostd_w	output	Write access requirement	
hostd_f	output	Freeze register selection	

The signal characteristics of the serial interface block are set out in the table below:

Table 188: SPI interface characteristics

Parameter	Value			Units	Comments
	Minimum	Typical	Maximum		
Clock spi_clk			10	MHz	
Clock phase (PHA)	PHA = 1, scanning during falling clock edge				
Clock polarity (POL)	POL = 0, base value of the clock is 0				
Data endianness	MSB is clocked out first				

The SPI interface block implements the following register based transactions:

- Read individual register
- Read several registers with random access
- Write to individual register
- Write to several registers (automatic increment)
- Read/write sequence

NOTE

It must be noted that during a read/write sequence, the write operation must always be the final transaction.

At the beginning of each transaction, the DSL Master transmits the online-status via `spi_miso` in two bytes.

9.2.1 Time control of the SPI

When accessing the SPI interface, the following specifications for time control must be adhered to:

NOTE
Each transaction via SPI is included in `spi_sel` by a "1" level. With each `spi_sel` reset, a new transaction is started. This causes the online-status to be retransmitted in the first two bytes.

The time sequence is shown in the following time sequence diagram and in table 189.

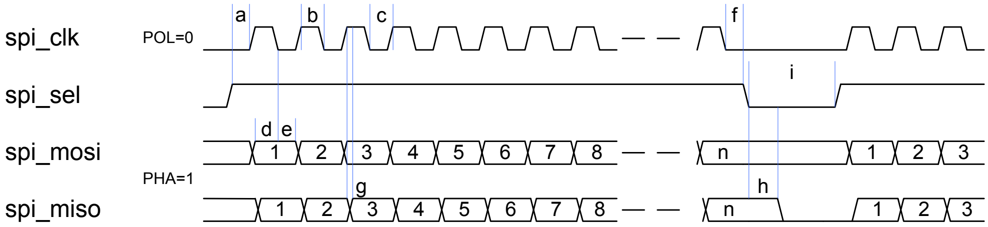


Figure 37: Time control of the SPI

The time control is given in the table below:

Table 189: Time control of the SPI

Diagram position	Description	Minimum	Maximum	Units
a	Setting `spi_sel` before `spi_clk`	25		ns
b	Time for `spi_clk` high	50		ns
c	Time for `spi_clk` low	50		ns
d	Setting `spi_mosi` before `spi_clk` low	10		ns
e	Keep `spi_mosi` at `spi_clk` low	25		ns
f	Keep `spi_sel` at `spi_clk` low	260		ns
g	Delay `spi_miso` at `spi_clk` high	25	60	ns
h	Delay `spi_miso` at `spi_sel` low	25	60	ns
i	Time for `spi_sel` low	50		ns

9.2.2 Dummy read process

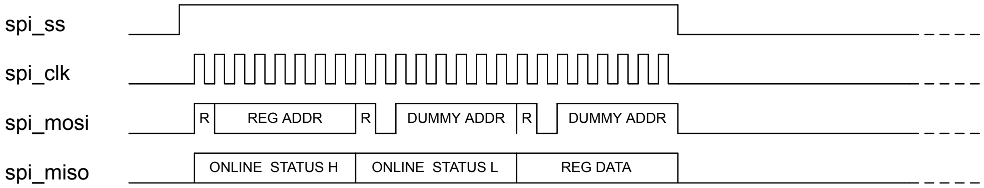
Due to the transmission of the online-status, Online Status read transactions need less time for transmission via `spi_mosi` than when receiving via `spi_miso`. Therefore, when receiving via `spi_miso`, dummy read transactions must be inserted into `spi_mosi` to avoid unwanted extra transactions.

A read access to register 3Fh has no effect and must be used for this purpose.

9.2.3 Read individual register

Using the SPI transaction "Read individual register", an individual register can be read in the IP Core of the DSL Master.

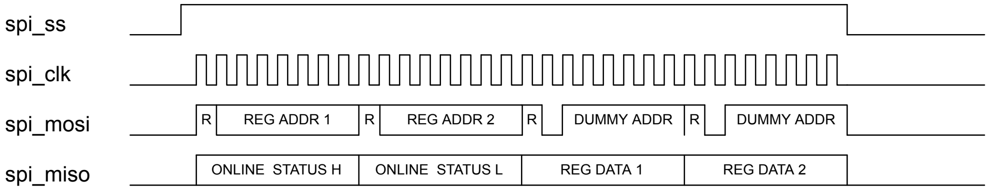
Symbol	Meaning
R	Access bit: Read ("1")
REG ADDR	Register address (00h to 7Fh)
DUMMY ADDR	Register address for the dummy read process (3Fh)
ONLINE STATUS H	Online-status – High byte
ONLINE STATUS L	Online-status – Low byte
REG DATA	Register content



9.2.4 Read several registers

Using the SPI transaction "Read several registers", several registers can be read in the IP Core of the DSL Master. Registers can be selected for reading in any sequence desired.

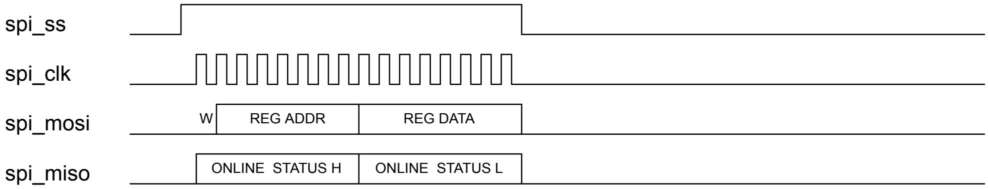
Symbol	Meaning
R	Access bit: Read ("1")
REG ADDR x	Register address (00h to 7Fh), no. x
DUMMY ADDR	Register address for the dummy read process (3Fh)
ONLINE STATUS H	Online-status – High byte
ONLINE STATUS L	Online-status – Low byte
REG DATA x	Content of register x



9.2.5 Write to individual register

Using the SPI transaction "Write to individual register", an individual register can be written to in the IP Core of the DSL Master.

Symbol	Meaning
W	Access bit: Write ("0")
REG ADDR	Register address (00h to 7Fh)
REG DATA	Register content
ONLINE STATUS H	Online-status – High byte
ONLINE STATUS L	Online-status – Low byte

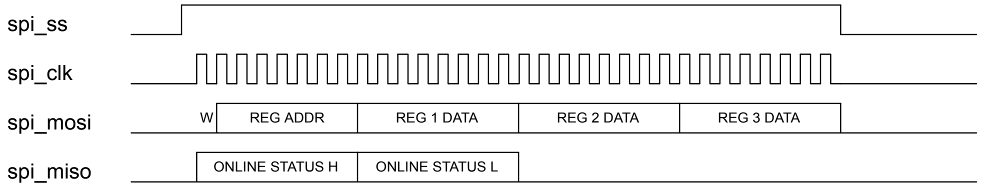


9.2.6 Write to several registers (automatic increment)

Using the SPI transaction "Write to several registers", several registers can be written to in the IP Core of the DSL Master. During the transaction only the address of the starting register is transmitted. With each register data byte, the IP Core raises the addresses automatically.

NOTE
If several registers are written to in any desired order, care must be taken that for each new sequence new SPI transactions are started.

Symbol	Meaning
W	Access bit: Write ("0")
REG ADDR	Address of the starting register (00h to 7Fh)
REG DATA x	Content of register x, beginning at REG ADDR
ONLINE STATUS H	Online-status – High byte
ONLINE STATUS L	Online-status – Low byte

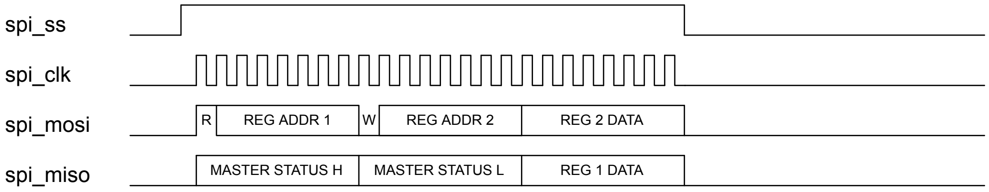


9.2.7 Read/write sequence

The SPI transaction "Read/write sequence" permits a rapid sequence of actions that consist of read processes at one or more registers and related write processes to several registers in sequence in the IP Core of the DSL Master. The SPI SS is not reset during the transaction and the online-status is not transmitted twice.

NOTE
The write operation must always form the final part of the "Read/write sequence". Multiple and single read or write accesses cannot be combined.

Symbol	Meaning
R	Access bit: Read ("1")
W	Access bit: Write ("0")
REG ADDR 1	Register address for read access (00h to 7Fh)
REG ADDR 2	Register start address for write access (00h to 7Fh)
REG DATA 2	Register content for write access
MASTER STATUS H	Online-status – High byte
MASTER STATUS L	Online-status – Low byte
REG DATA 1	Register content for read access



9.2.8 SPI errors

If the address of the SPI interface is wrong, fault indications are issued via `spi_miso`.

The fault indication is shown by a "1" signal to `spi_miso`, after which the `spi_sel` is reset by the frequency inverter application.

table 190 contains a list of the fault conditions that lead to a fault indication.

Table 190: SPI errors

Fault condition of the SPI	Fault indication
Incorrect number of CLK impulses	`spi_miso` at high level
Write command without data	`spi_miso` at high level

9.3 Parallel interface block

As an example, a parallel interface block is supplied together with the IP Core.

The parallel interface block follows the Texas Instruments Asynchronous External Memory Interface A (EMIFA). The reference document for this interface is the User's Guide SPRUFL6E, dated April 2010.

The figure and table below show the interface signals.

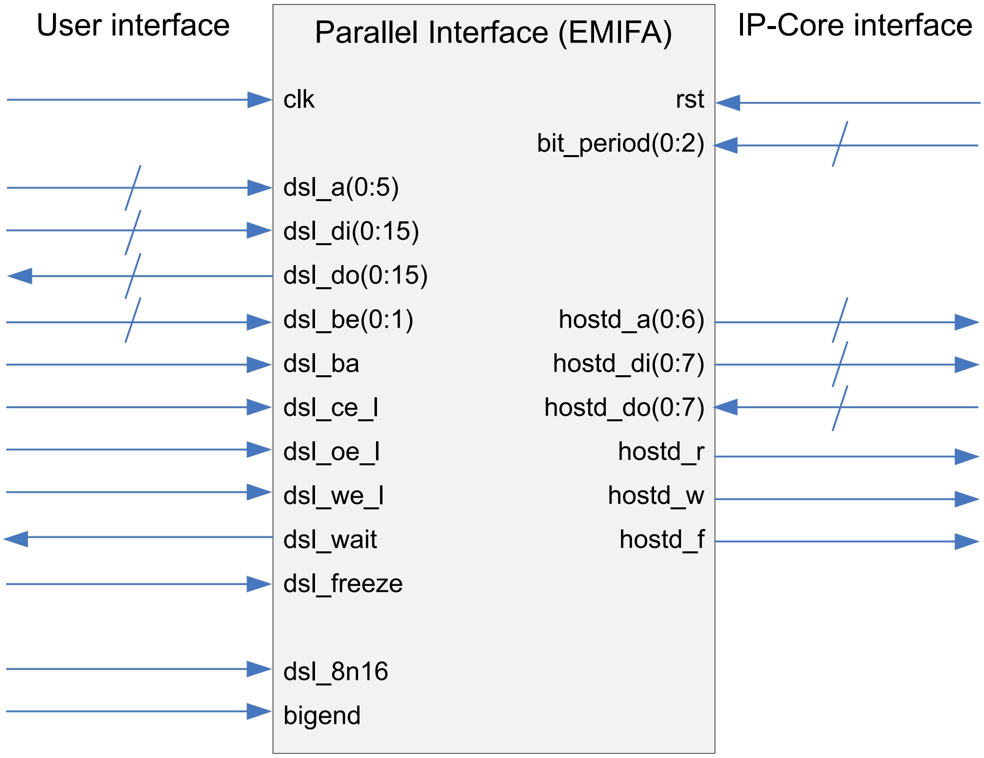


Figure 38: Parallel interface block signals

NOTE
Please note that the EMIFA block supplied with the HIPERFACE DSL® IPCore is only fit for the usage with drive interface. It can be adapted to safe 1 and safe 2 interfaces if needed, but it is not suggested.

Table 191: Parallel interface block signals

Pin name	Model name	Function	Note
Host interface			
ema_clk	Input	Clock input	Separate clock domain to the IP Core
dsl_a(5:0)	Input	EMIFA: Address bus	
dsl_di(15:0)	Input	EMIFA: Data bus input	

Pin name	Model name	Function	Note
dsl_do(15:0)	output	EMIFA: Data bus output	
dsl_be(1:0)	Input	EMIFA: Byte switch on	
dsl_ba	Input	EMIFA: Memory bank	
dsl_ce_l	Input	EMIFA: Slave selection	
dsl_oe_l	Input	EMIFA: Output switch on	
dsl_we_l	Input	EMIFA: Write access	
dsl_wait	output	EMIFA: Maintenance display	
dsl_freeze	Input	Freeze register	
dsl_8n16	Input	Data bus width selection	
bigend	Input	Byte sequence selection	Connect to the bigend input of the IP Core as well
IP Core interface			
rst	Input	Internal reset	Connect to IUO(1) of the IP Core
bit_period(2:0)	Input	Internal state machine	Connect to IUO(4:2) of the IP-Core
online_status_d(15:0)	Input	Internal status IP Core	
hostd_a(6:0)	output	Register block address bus	
hostd_di(7:0)	output	Data bus interface to core	
hostd_do(7:0)	Input	Data bus core to interface	
hostd_r	output	Read access requirement	
hostd_w	output	Write access requirement	
hostd_f	output	Freeze register selection	Should be set 1 cycle before reading starts

NOTE
Note that the parallel interface does not implement the `online_status` signals. These signals must be recorded by the user separately to the parallel interface.

The signal characteristics of the parallel interface block are set out in the table below:

Table 192: Characteristics of the EMIFA interface

Parameter	Value			Units
	Min.	Typical	Max.	
Clock (ema_clk)			100	MHz
Setup time	2			EMA_CLK cycles
Strobe time	7			EMA_CLK-cycles
Hold time	1			EMA_CLK-cycles
Turnaround time	1			EMA_CLK-cycles

9.3.1 Assignment to the host

The assignment of the signals between Host (interfaces master with EMIFA signals) and DSL Master (interfaces slave) should appear as follows:

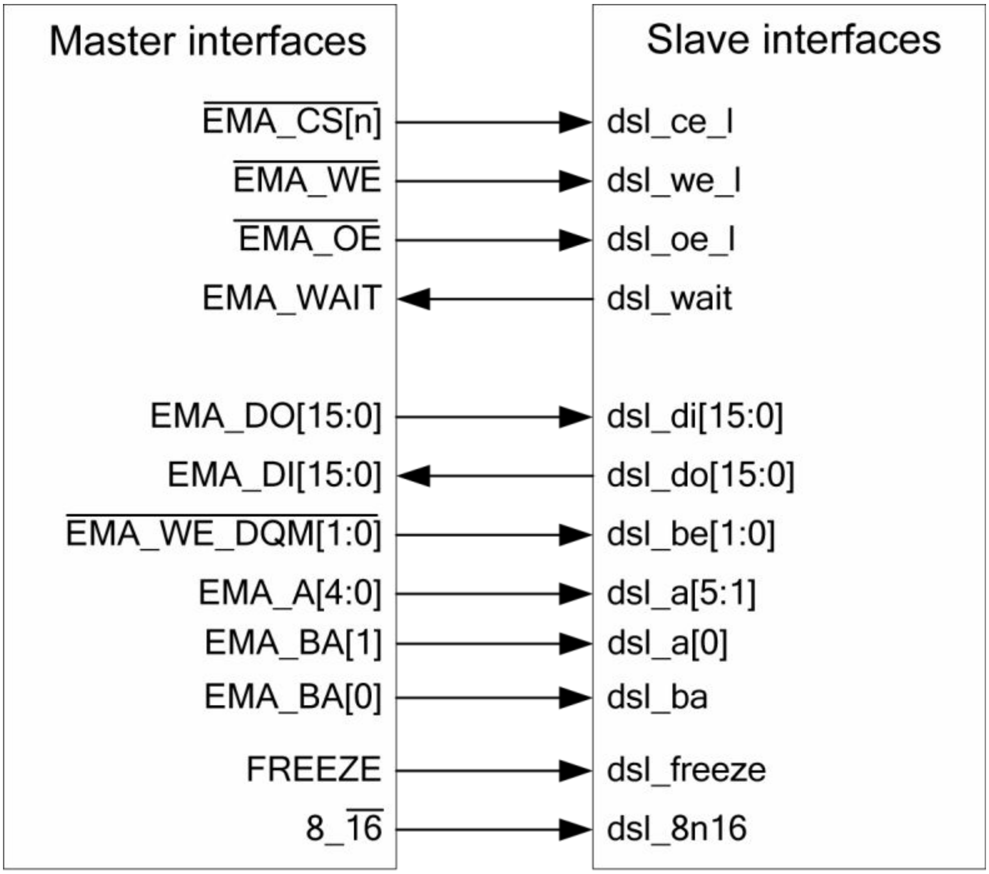


Figure 39: Allocation of parallel interface block to host

The table below provides details of the interface installation:

Table 193: Details of the parallel interface blocks

Interfaces signal	Installation detail
dsl_di, dsl_do	16 bit wide data bus with separate input and output direction.
dsl_a	Address bus for addressing 16 bit registers.
dsl_ba	Sub-address for 8 bit interface. If the parallel bus is used with a 16 bit wide EMIFA interface, only even addresses are used and this input is not used.
dsl_be_l	"Switch on single byte" inputs. These inputs are used to access individual bytes of a 16 bit wide EMIFA interface.
dsl_ce_l	Slave parallel bus selection. This signal can be an internally generated selection signal (chip enable) or part of the address decoding. If this signal is deactivated ('1'), no access to the DSL Master is possible.
dsl_oe_l	Input "Output switch on". This signal gives the time control for a direction switch of the bi-directional bus.
dsl_we_l	Input "Switch on write access". This signal gives the time control for a write access to the DSL Master.
dsl_wait	Waiting time Host direction The minimum requirement for the bus states of the parallel bus are specified in Table 9.

In addition to the EMIFA signals, the `dsl_freeze` and `dsl_8n16` signals are implemented.

Table 194: Additional parallel interface block signals

Interfaces signal	Value	Function
dsl_freeze		Use for consistent multi-byte access
	0	Multi-byte registers are refreshed
	1	Multi-byte registers are frozen
dsl_8n16		Use for selection of the data bus width
	0	16 bit width
	1	8 bit width

Also, because of the selection of `bigend`, the register assignment of the DSL Master should be taken into account.

9.4 Basic interface specification

If a fast connection of the IP-Core registers is needed the user can develop an own interface block directly connected to the basic interface. The following signals are available for interfacing the IP Core directly:

Table 195: Direct interfacing signals

Pin name	IP Core signal type	Function	Note
hostd_a(6:0)	input	Register block address bus	
hostd_di(7:0)	input	Data in bus interface to core	
hostd_do(7:0)	output	Data out bus core to interface	
hostd_r	input	Read access identifier	
hostd_w	input	Write access identifier	
hostd_f	input	Register block freeze indicator	
bit_period(2:0)	output	Internal state machine timing	aux_signals (4:2) top level output

9.4.1 General information

The main signals for accessing the basic interface are the address bus (`hostd_a`), the data in (`hostd_di`) and data out (`hostd_do`) lines as well as their respective read and write flags (`hostd_r` and `hostd_w`). Apart from these access signals, there are two other important signals:

1 The register freeze flag (`hostd_f`)
This flag controls the updating of multi-byte registers inside the IP Core. When it is raised to ‘1’, no further updates are done for those multi-byte registers. Update values are stored separately, but will not be available in their corresponding registers until the freeze flag drops to ‘0’ again. This procedure should be used whenever a multi-byte register is read, to avoid updates in between read operations of a single set of data.

2 The internal timing reference (`bit_period`)
Some registers care for being read by the user, and there is a specific timing when the data from the data in line is actually written to the register on which the write command is issued. Therefore, when implementing the fastest possible communications, the user needs to be aware of these timings.

NOTE
It should be noted that some commands can only be issued once per eight clock cycles of the IP Core (which translates to roughly 110 ns). If the user interface is designed in a way that guarantees the read/write flags to be always set for a significantly longer time, then the timing reference doesn't need to be monitored.

9.4.2 Read access timing

The read access through the basic interface must be implemented according to the diagram below.

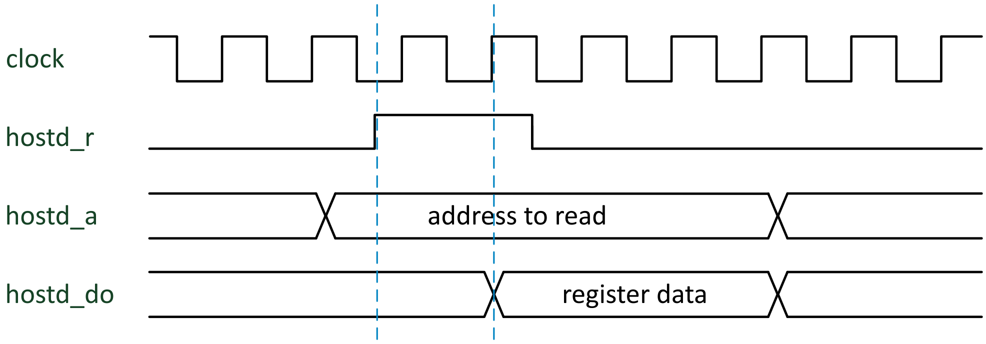


Figure 40: Read access basic interface

The register address to be read must be set at the address bus (`hostd_a`). Then the read flag (`hostd_r`) must be set to “1”. After two clock cycles (i.e. at the second rising edge of the `CLK` signal, max. 27 ns) the register data will be available at the data out bus (`hostd_do`). It will be kept available for as long as the address is not changed. This is the fastest reading procedure possible for the IP Core.

9.4.3 Write access timing

The write access through the basic interface must be implemented according to the diagram below.

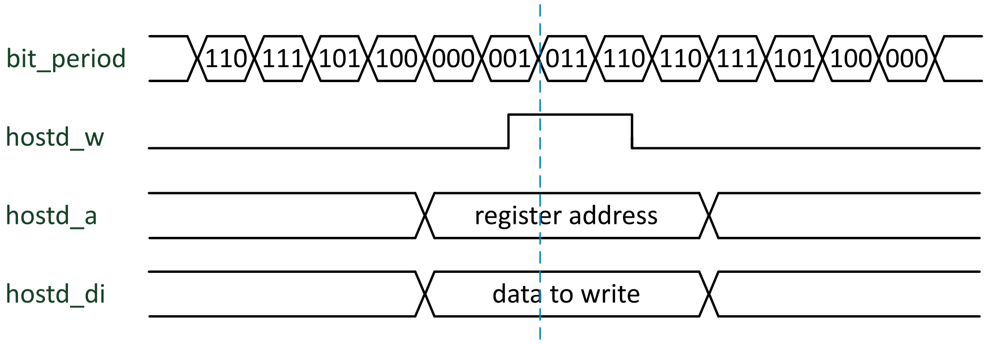


Figure 41: Write access basic interface

For write operations it is important to monitor the `bit_period` signal when implementing the fastest possible write commands. Register address and data to write to the register must be set on their corresponding busses (`hostd_a and hostd_di`). Then the write flag (`hostd_w`) must be set to “1” and stay active while `bit_period` is transitioning from “000” to “001”. At the time of this transition, the data in the register will be updated with the data of the data in bus. The write flag should be kept high until `bit_period` has reached “011”.

`bit_period` is a three bit gray counter, incrementing every clock cycle (around 13 ns). Thus, it is only possible to execute one write command every eight clock cycles, which takes about 110 ns. Of course, if the customer interface connected will issue commands significantly slower, and it is known that the write flag will be active for a time greater than 110 ns, `bit_period` doesn't need to be monitored. It is only needed when aiming for the most rapid interfacing.

9.5 Register assignment

The address assignment of the registers depends on the `bigend` control signal and likewise on the `dsl_8n16` control signal for 16 bit wide parallel bus interfaces.

NOTE
It should be noted that only 8 bit addressing is available when using the SPI interface block.

When using the parallel bus interface block (EMIFA) with 16 bit addressing, the register addresses must be assigned to the `dsl_a` and `dsl_ba` signals as follows:

Table 196: Parallel bus register address assignment

Signal	Register address
dsl_a[5:0]	Bit 6:1
dsl_ba	Bit 0

The table below specifies the relevant address assignment.

Table 197: Address assignment for the DSL Master

Designation	8 bit, Big-Endian	8 bit, Little-Endian	16 bit, Big-Endian	16 bit, Little-Endian
SYS_CTRL	00h	03h	00h (15:8)	02h (15:8)
SYNC_CTRL	01h	02h	00h (7:0)	02h (7:0)
MASTER_QM	03h	00h	02h (7:0)	00h (7:0)
EVENT_H	04h	07h	04h (15:8)	06h (15:8)
EVENT_L	05h	06h	04h (7:0)	06h (7:0)
MASK_H	06h	05h	06h (15:8)	04h (15:8)
MASK_L	07h	04h	06h (7:0)	04h (7:0)
MASK_SUM	08h	0Bh	08h (15:8)	0Ah (15:8)
EDGES	09h	0Ah	08h (7:0)	0Ah (7:0)
DELAY	0Ah	09h	0Ah (15:8)	08h (15:8)
VERSION	0Bh	08h	0Ah (7:0)	08h (7:0)
ENC_ID2	0Dh	0Eh	0Ch (7:0)	0Eh (7:0)
ENC_ID1	0Eh	0Dh	0Eh (15:8)	0Ch (15:8)
ENC_ID0	0Fh	0Ch	0Eh (7:0)	0Ch (7:0)
POS4	10h	17h	10h (15:8)	16h (15:8)
POS3	11h	13h	10h (7:0)	12h (15:8)
POS2	12h	12h	12h (15:8)	12h (7:0)
POS1	13h	11h	12h (7:0)	10h (15:8)
POS0	14h	10h	14h (15:8)	10h (7:0)
VEL2	15h	16h	14h (7:0)	16h (7:0)
VEL1	16h	15h	16h (15:8)	14h (15:8)
VEL0	17h	14h	16h (7:0)	14h (7:0)

Designation	8 bit, Big-Endian	8 bit, Little-Endian	16 bit, Big-Endian	16 bit, Little-Endian
SUMMARY	18h	1Eh	18h (15:8)	1Eh (7:0)
VPOS4	19h	1Fh	18h (7:0)	1Eh (15:8)
VPOS3	1Ah	1Bh	1Ah (15:8)	1Ah (15:8)
VPOS2	1Bh	1Ah	1Ah (7:0)	1Ah (7:0)
VPOS1	1Ch	19h	1Ch (15:8)	18h (15:8)
VPOS0	1Dh	18h	1Ch (7:0)	18h (7:0)
VPOSCRC_H	1Eh	1Dh	1Eh (15:8)	1Ch (15:8)
VPOSCRC_L	1Fh	1Ch	1Eh (7:0)	1Ch (7:0)
PC_BUFFER0	20h	20h	20h (15:8)	20h (7:0)
PC_BUFFER1	21h	21h	20h (7:0)	20h (15:8)
PC_BUFFER2	22h	22h	22h (15:8)	22h (7:0)
PC_BUFFER3	23h	23h	22h (7:0)	22h (15:8)
PC_BUFFER4	24h	24h	24h (15:8)	24h (7:0)
PC_BUFFER5	25h	25h	24h (7:0)	24h (15:8)
PC_BUFFER6	26h	26h	26h (15:8)	26h (7:0)
PC_BUFFER7	27h	27h	26h (7:0)	26h (15:8)
PC_ADD_H	28h	2Bh	28h (15:8)	2Ah (15:8)
PC_ADD_L	29h	2Ah	28h (7:0)	2Ah (7:0)
PC_OFF_H	2Ah	29h	2Ah (15:8)	28h (15:8)
PC_OFF_L	2Bh	28h	2Ah (7:0)	28h (7:0)
PC_CTRL	2Ch	2Dh	2Ch (15:8)	2Ch (15:8)
PIPE_S	2Dh	2Fh	2Ch (7:0)	2Eh (15:8)
PIPE_D	2Eh	2Eh	2Eh (15:8)	2Eh (7:0)
PC_DATA	2Fh	2Ch	2Eh (7:0)	2Ch (7:0)
ACC_ERR_CNT	38h	38h	38h (15:8)	38h (7:0)
MAXACC	39h	39h	38h (7:0)	38h (15:8)
MAXDEV_H	3Ah	3Bh	3Ah (15:8)	3Ah (15:8)
MAXDEV_L	3Bh	3Ah	3Ah (7:0)	3Ah (7:0)
DUMMY	3Fh	3Fh	n/v	n/v

9.6 Implementation of the IP Core for Xilinx Spartan-3E/6

The DSL Master IP Core is provided by SICK for Xilinx Spartan-3E and Spartan-6 FPGA components.

table 196 lists the requirements that must be fulfilled for the FPGA components selected.

Table 198: Requirements of Xilinx Spartan-3E/6 FPGAs

FPGA requirements	IP Core variant	Spartan-3E	Spartan-6
Supported "Speed Grades"		-4 -5	-3 -2
Number of "Slices" used / "Slice register"	Standard, serial	2,522	1,882
Number of "BUFG/BUFGMUX" used		1	1
Timing requirements (clk periods)		13 ns	13 ns

The implementation of the IP Core is based on an installed tool-chain, that is provided by Xilinx.

Table 199: Xilinx tool-chain

Tool	Version
Xilinx ISE Design Suite	14.7

9.6.1 Design variants

The DSL Master IP Core is supplied in two variants.

Standard

The "Standard" variant (dslm_n) is a small IP Core, but does not support any safety functions and diagnostics of the protocol and the motor feedback systems connected. Use of the "Standard" variant does **not** permit a safety relevant application to be supported using SICK motor feedback systems certified for safety applications.

Safe

The "Safe" variant (dslm_s) is a larger IP Core and supports safety functions and diagnostics in accordance with the requirements which are described in this manual and which are required for specific motor feedback systems (see the corresponding data sheet).

Both variants can be combined with various interface blocks (see chapter 9.1). Example projects are provided on the accompanying CD-ROM.

All IP Cores with all interface blocks are packaged as ZIP files that can be found on the accompanying CD-ROM in the "IP Core\Xilinx" folder.

Table 200: IP Core Xilinx Spartan-3E/6

Zip file
yymmdd.dslmaster_ xilinx.zip

yymmdd defines the release date of the IP Core in the format year (yy), month (mm) and day (dd).

9.6.2 Design resources

The DSL Master IP Core is supplied as a net list in NGC format. Additional design resources contain a "constraint" file that specifies the time and environmental conditions as well as a VHDL template ("wrapper") for a top level circuit that integrates the IP Core.

Table 201: Design resources

Path in ZIP archive	File	dslm_n_bus	dslm_n_spi	Resource
\ISE-sp3\...	dslm_n.ngc	x	x	IP Core, wrapper, timing constraints Spartan-3E
	top.ucf	x	x	
	rapper_n_bus.vhd wrap-	x		
	per_n_bus.xise wrap-	x		
	per_n_spi.vhd wrap-		x	
	per_n_spi.xise		x	
\ISE-sp6\...	dslm_n.ngc	x	x	IP core, wrapper, timing constraints Spartan-6
	top.ucf	x	x	
	wrapper_n_bus.vhd	x		
	wrapper_n_bus.xise	x		
	wrapper_n_spi.vhd		x	
	wrapper_n_spi.xise		x	

Path in ZIP archive	File	dslm_n_bus	dslm_n_spi	Resource
\External_Blocks	bus_ctrl.vhd spi_ctrl.vhd	x	 x	Open source interface blocks

If a Xilinx ISE project is loaded, the NGC net list must be copied directly into the project folder.

NOTE
Please note that the NGC net lists cannot be added via the menu commands "Project > Add Source" or "Project > Add Copy of Source".

The instantiation of the NGC net list is by a "black_box" in the source code of the top level design.

In VHDL, this instantiation is in accordance with the following template:

```
component dslm_n

port (...);

endcomponent;

...

attribute box_type : string;

attribute box_type of dslm_n: component is „black_box";

...

begin

...

<component_name> : dslm_n

port map (...);

...
```

In Verilog, this instantiation is in accordance with the following template:

```
module dslm_n (...);

endmodule

...

dslm_n <module_name> (...);

//synthesis attribute box_typeofdslm_n is„black_box"

...
```

9.6.3 Demo project

To get started quickly, each IP Core is supplied with a demo project for Xilinx ISE. This demo project contains all the settings to begin the installation of the IP Core with the accompanying top level design ("wrapper").

Prerequisites for the demo project are:

- All the files from the ZIP archive for the IP Core have been extracted into a folder. Any preferred location may be used for this.
- All the tools listed have been installed on the development computer.

Demo project resources

The ZIP archive contains the project file for Xilinx ISE, as well as help files that are automatically generated by Xilinx ISE.

The project files for the variants are set out in the table below:

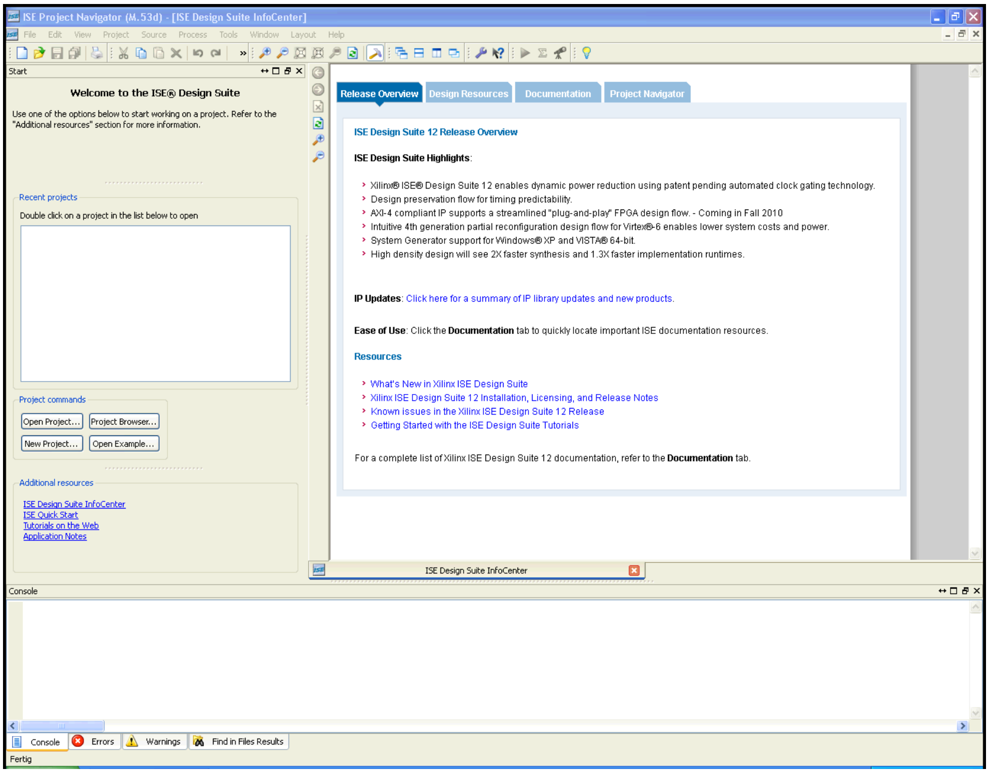
Table 202: Demo project resources

Path in ZIP archive	File	dslm_n_bus	dslm_n_spi	Resource
\ISE_sp3\...	wrapper_n_bus.xise wrapper_n_spi.xise	x	 x	Xilinx ISE project file Spartan-3E
\ISE_sp6\...	wrapper_n_bus.xise wrapper_n_spi.xise	x	 x	Xilinx ISE project file Spartan-6

Sequence of the demo project

This chapter lists the steps that are required to carry out the demo project for the "Standard" variant with SPI interface in Xilinx ISE 12.1.

Start "ISE Project Navigator".



Welcome to the ISE® Design Suite

Use one of the options below to start working on a project. Refer to the "Additional resources" section for more information.

Double click on a project in the list below to open

ISE Design Suite 12 Release Overview

ISE Design Suite Highlights:

- Xilinx® ISE® Design Suite 12 enables dynamic power reduction using patent pending automated clock gating technology.
- Design preservation flow for timing predictability.
- AXI-4 compliant IP supports a streamlined "plug-and-play" FPGA design flow. - Coming in Fall 2010
- Intuitive 4th generation partial reconfiguration design flow for Virtex®-6 enables lower system costs and power.
- System Generator support for Windows® XP and VISTA® 64-bit.
- High density design will see 2X faster synthesis and 1.3X faster implementation runtimes.

IP Updates: Click here for a summary of IP library updates and new products.

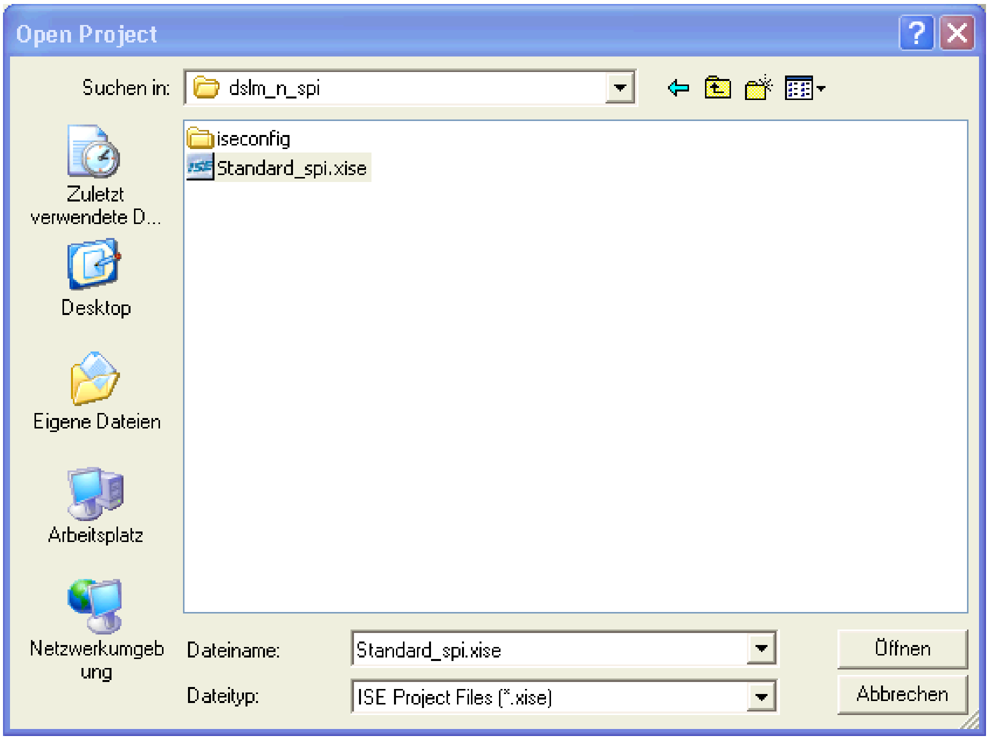
Ease of Use: Click the **Documentation** tab to quickly locate important ISE documentation resources.

Resources

- What's New in Xilinx ISE Design Suite
- Xilinx ISE Design Suite 12 Installation, Licensing, and Release Notes
- Known issues in the Xilinx ISE Design Suite 12 Release
- Getting Started with the ISE Design Suite Tutorials

For a complete list of Xilinx ISE Design Suite 12 documentation, refer to the **Documentation** tab.

Open the "Standard" project file.

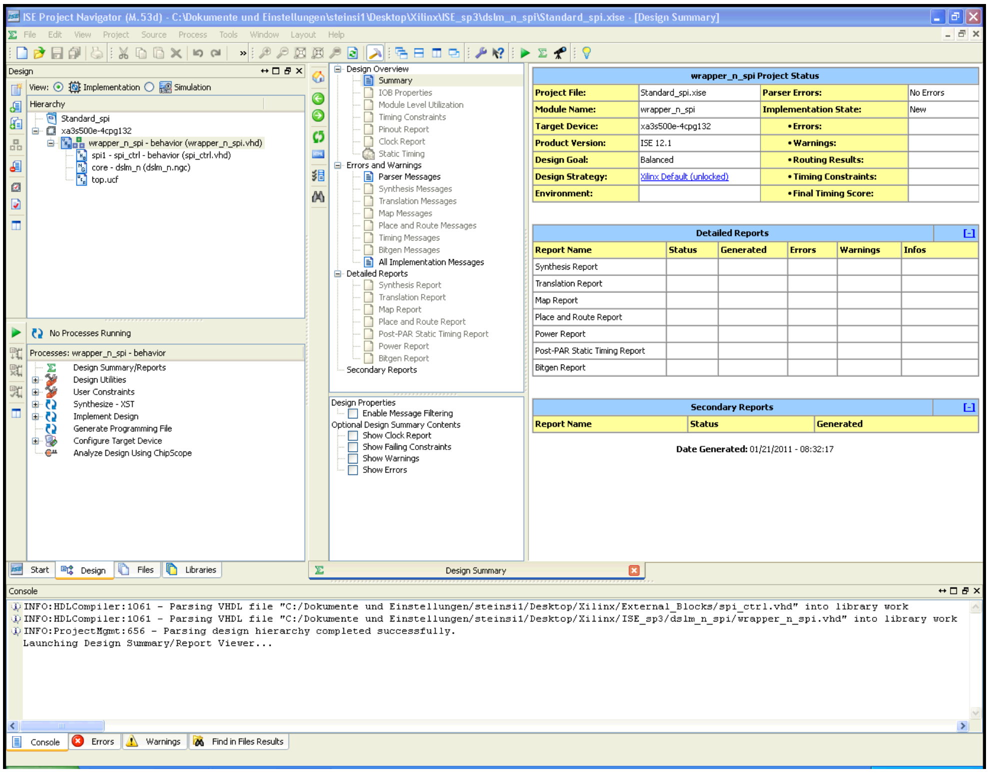


Suchen in: dslm_n_spi

Dateiname: Standard_spi.xise

Dateityp: ISE Project Files (*.xise)

The "ISE Project Navigator" should appear as follows:



ISE Project Navigator (M.53d) - C:\Dokumente und Einstellungen\steinsi1\Desktop\Xilinx\ISE_sp3\dslm_n_spi\Standard_spi.xise - [Design Summary]

View: ● Implementation ○ Simulation

□ Enable Message Filtering

□ Show Clock Report

□ Show Failing Constraints

□ Show Warnings

□ Show Errors

wrapper_n_spi Project Status			
Project File:	Standard_spi.xise	**Parser Errors:**	No Errors
Module Name:	wrapper_n_spi	**Implementation State:**	New
Target Device:	xa3s500e-4cpg132	**• Errors:**	
Product Version:	ISE 12.1	**• Warnings:**	
Design Goal:	Balanced	**• Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:		**• Final Timing Score:**	

Detailed Reports [-]

Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report					
Translation Report					
Map Report					
Place and Route Report					
Power Report					
Post-PAR Static Timing Report					
Bitgen Report					

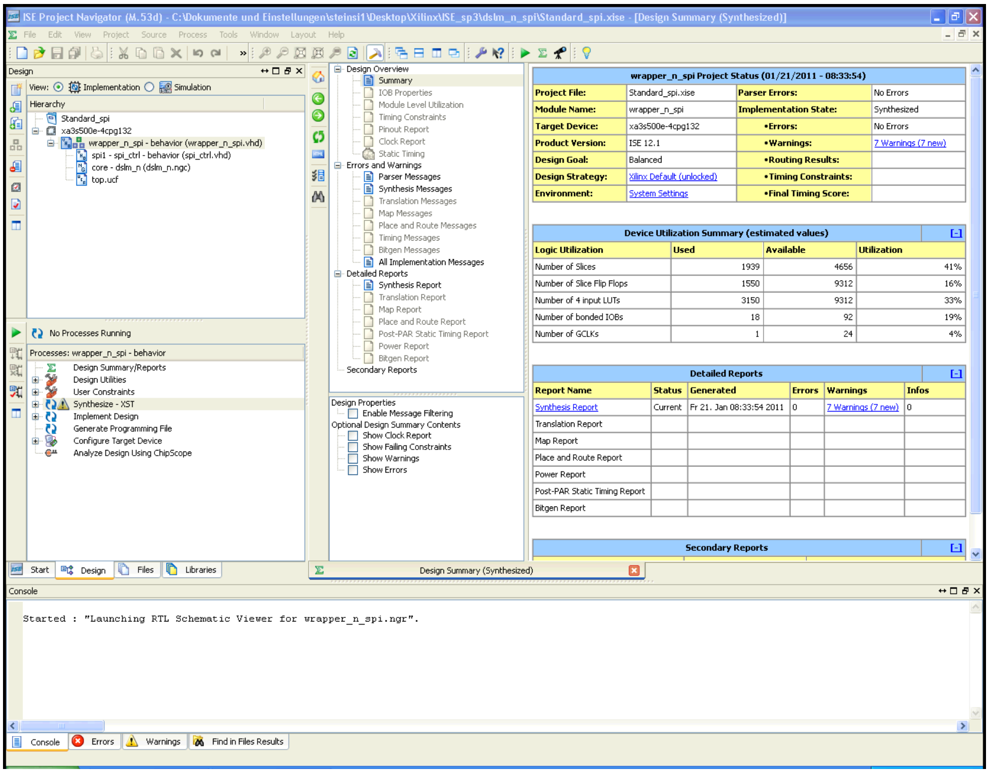
Secondary Reports [-]

Report Name	Status	Generated

Date Generated: 01/21/2011 - 08:32:17

```
INFO:HDLCompiler:1061 - Parsing VHDL file "C:/Dokumente und Einstellungen/steinsi1/Desktop/Xilinx/External_Blocks/spi_ctrl.vhd" into library work
INFO:HDLCompiler:1061 - Parsing VHDL file "C:/Dokumente und Einstellungen/steinsi1/Desktop/Xilinx/ISE_sp3/dslm_n_spi/wrapper_n_spi.vhd" into library work
INFO:ProjectMgmt:656 - Parsing design hierarchy completed successfully.
Launching Design Summary/Report Viewer...
```

Start the circuit synthesis:



wrapper_n_spi Project Status (01/21/2011 - 08:33:54)			
Project File:	Standard_spi.xise	**Parser Errors:**	No Errors
Module Name:	wrapper_n_spi	**Implementation State:**	Synthesized
Target Device:	xa3s500e-4cpg132	**•Errors:**	No Errors
Product Version:	ISE 12.1	**•Warnings:**	7 Warnings (7 new)
Design Goal:	Balanced	**•Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**•Timing Constraints:**	
Environment:	System Settings	**•Final Timing Score:**	

Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slices	1939	4656	41%
Number of Slice Flip Flops	1550	9312	16%
Number of 4 input LUTs	3150	9312	33%
Number of bonded IOBs	18	92	19%
Number of GCLKs	1	24	4%

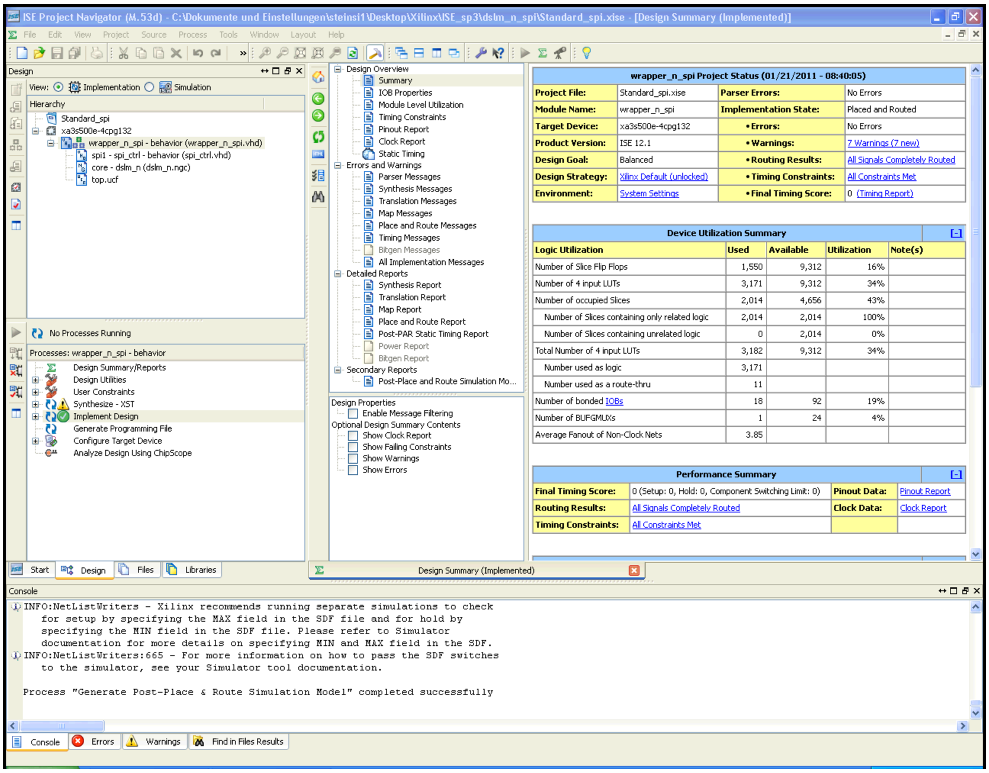
Detailed Reports

Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Fr 21. Jan 08:33:54 2011	0	7 Warnings (7 new)	0
Translation Report					
Map Report					
Place and Route Report					
Power Report					
Post-PAR Static Timing Report					
Bitgen Report					

Secondary Reports

```
Started : "Launching RTL Schematic Viewer for wrapper_n_spi.ngr".
```

Start the circuit layout:



wrapper_n_spi Project Status (01/21/2011 - 08:40:05)			
Project File:	Standard_spi.xise	**Parser Errors:**	No Errors
Module Name:	wrapper_n_spi	**Implementation State:**	Placed and Routed
Target Device:	xa3s500e-4cpg132	**•Errors:**	No Errors
Product Version:	ISE 12.1	**•Warnings:**	7 Warnings (7 new)
Design Goal:	Balanced	**•Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	**•Timing Constraints:**	All Constraints Met
Environment:	System Settings	**•Final Timing Score:**	0 (Timing Report)

Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	1,550	9,312	16%	
Number of 4 input LUTs	3,171	9,312	34%	
Number of occupied Slices	2,014	4,656	43%	
Number of Slices containing only related logic	2,014	2,014	100%	
Number of Slices containing unrelated logic	0	2,014	0%	
Total Number of 4 input LUTs	3,182	9,312	34%	
Number used as logic	3,171			
Number used as a route-thru	11			
Number of bonded IOBs	18	92	19%	
Number of BUFGMUXs	1	24	4%	
Average Fanout of Non-Clock Nets	3.85			

Performance Summary

Final Timing Score:	0 (Setup: 0, Hold: 0, Component Switching Limit: 0)	**Pinout Data:**	Pinout Report
Routing Results:	All Signals Completely Routed	**Clock Data:**	Clock Report
Timing Constraints:	All Constraints Met		

```
INFO:NetListWriters - Xilinx recommends running separate simulations to check
   for setup by specifying the MAX field in the SDF file and for hold by
   specifying the MIN field in the SDF file. Please refer to Simulator
   documentation for more details on specifying MIN and MAX field in the SDF.
INFO:NetListWriters:665 - For more information on how to pass the SDF switches
   to the simulator, see your Simulator tool documentation.

Process "Generate Post-Place & Route Simulation Model" completed successfully
```

9.7 Installation of the IP Core for Altera FPGAs

The DSL Master IP Core is provided by SICK for Altera FPGA components. The IP Core is available in the form of encrypted VHDL files. These can be used for synthesis on all Altera FPGAs of sufficient size.

table 203 lists the requirements that must be fulfilled for the FPGA components selected.

Table 203: Requirements for Altera FPGAs

FPGA requirements	IP Core variant	i.e for Cyclone III
Supported "Speed Grades"		-7 -8
Number of "Logic Elements" used	Standard, serial	4,614
Timing requirements (clk)		Period: 13 ns

The installation of the IP Core is based on an installed Toolchain that is provided by Altera.

Table 204: Altera toolchain

Tool	Version
Quartus® II	v13.0

9.7.1 Design variants

The DSL Master IP Core is supplied in two variants.

Standard

The "Standard" variant (dslm_n) is a small IP Core, but does not support any safety functions and diagnostics of the log and the motor feedback systems connected. Use of the "Standard" variant does not permit a safety relevant application to be supported using SICK motor feedback systems certified for safety applications.

Safe

The "Safe" variant (dslm_s) is a larger IP Core and supports safety functions and diagnostics in accordance with the requirements which are described in this manual and which are required for specific motor feedback systems (see the corresponding data sheet).

Both variants can be combined with various interface blocks (see chapter 9.1). Example projects are provided on the accompanying CD-ROM.

All IP Cores with interface blocks are packaged in a ZIP file and can be found on the accompanying CD-ROM in the "IP Core\Altera" folder

Table 205: Altera IP Core variants

Altera toolchain
yymmdd.dslmaster_ altera.zip

yymmdd defines the release date of the IP Core in the format year (yy), month (mm) and day (dd).

9.7.2 Design resources

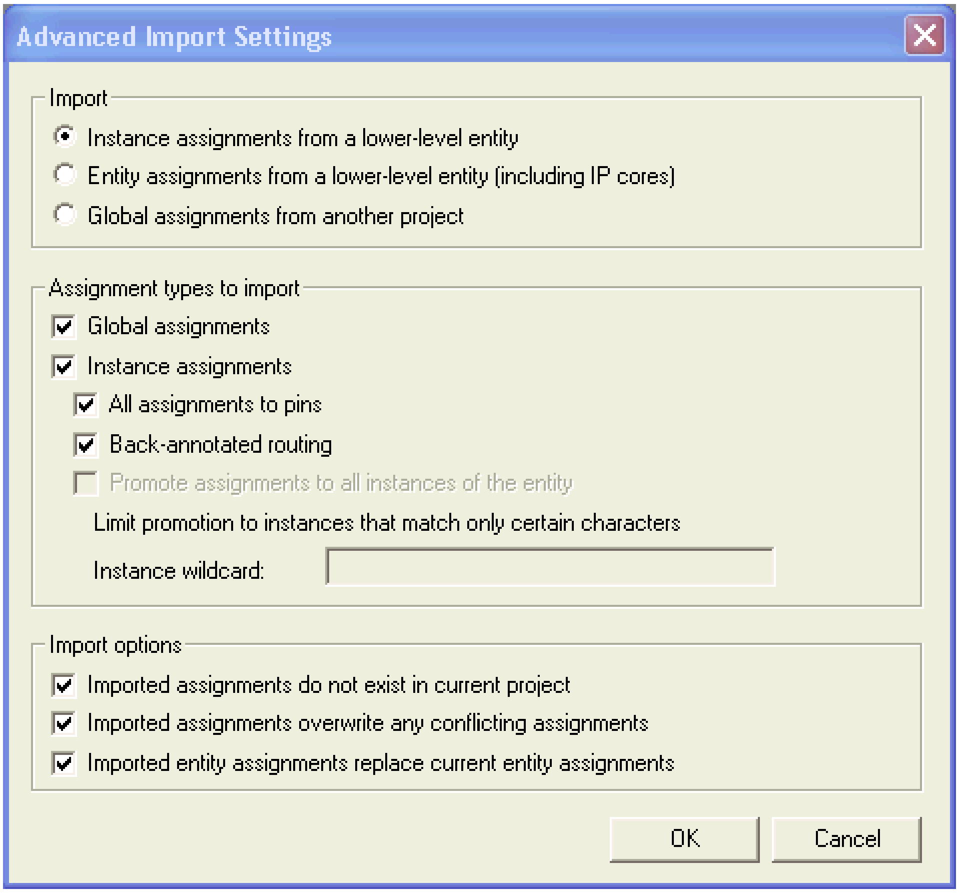
The DSL Master IP Core is supplied as an encrypted VHDL project. The interface blocks are supplied as open source VHDL files. Additional design resources contain a file that specifies the time and environmental conditions as well as a VHDL template for a top level circuit that integrates the IP Core and the interface blocks.

Table 206: Design resources

Path in ZIP archive	File	dslm_n_spi	dslm_n_bus	Resource
\ (root)	dslm_n_spi.qsf dslm_n_bus.qsf	x	x	Project settings and time conditions
\wrapper	wrapper_n_spi.vhd wrapper_n_bus.vhd	x	x	VHDL template for the top level circuit "wrapper" for Altera Quartus II
\source	e_dslm_n.vhd e_auxiliary.vhd e_check8b10b.vhd e_crc5.vhd e_crc16.vhd e_dec8b10b.vhd e_dsl_core_top.vhd e_dslmaster.vhd e_dualport.vhd e_enc8b10b.vhd e_flag.vhd e_framer.vhd e_globals.vhd e_int_ctrl.vhd e_intg.vhd e_master_pm.vhd e_par_mrx.vhd e_par_mtx.vhd e_par_pm.vhd e_pipeline.vhd e_reset_sync.vhd e_sampler.vhd e_sequencer.vhd e_sync_gen.vhd	x x x x x x x x x x x x x x x x x x x x x x x x	x x x x x x x x x x x x x x x x x x x x x x x x	Encrypted VHDL modules
\source\ External_Blocks	spi_ctrl.vhd bus_ctrl.vhd	x	 x	Open VHDL modules
\license	DSL_encr_license.dat	x	x	License file for use of the encrypted VHDL project

NOTE

Please note: If you import the time lines from the associated qsf file into Quartus II, set the following options in the "Advanced Import Settings" dialog box:



Advanced Import Settings

Import

- ● Instance assignments from a lower-level entity
- ○ Entity assignments from a lower-level entity (including IP cores)
- ○ Global assignments from another project

Assignment types to import

- ☑ Global assignments
- ☑ Instance assignments
  - ☑ All assignments to pins
  - ☑ Back-annotated routing
  - □ Promote assignments to all instances of the entity

Limit promotion to instances that match only certain characters

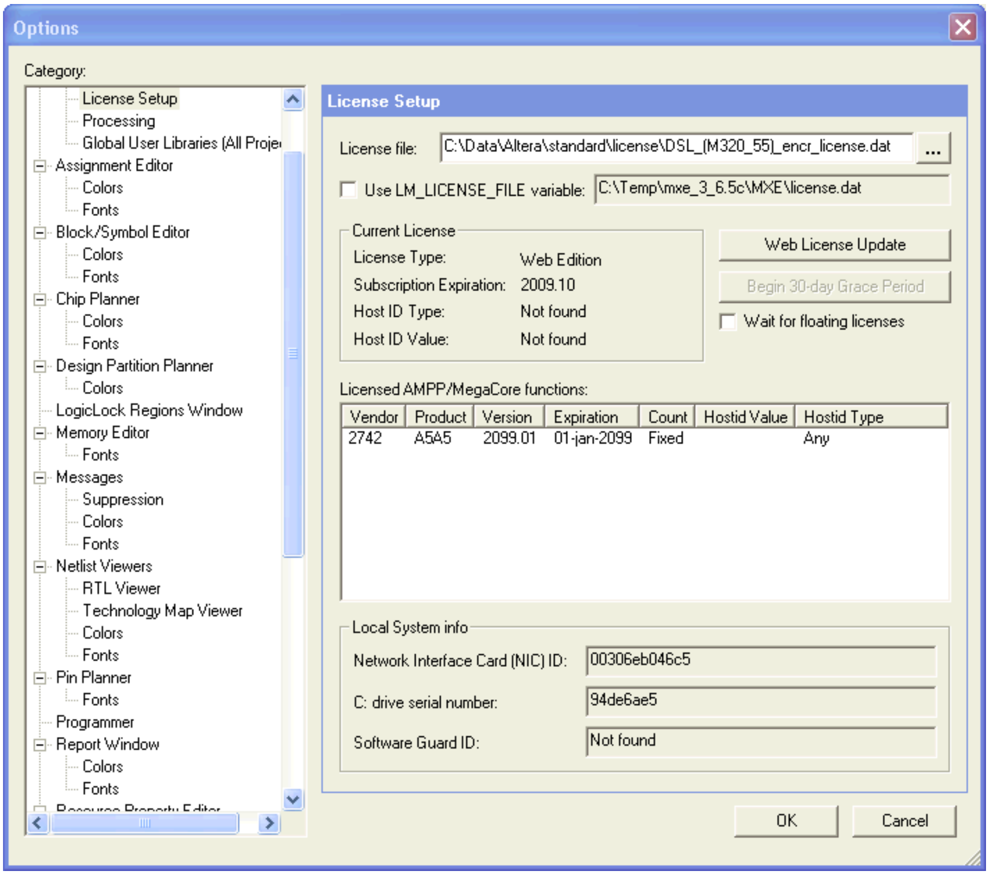
Instance wildcard:

Import options

- ☑ Imported assignments do not exist in current project
- ☑ Imported assignments overwrite any conflicting assignments
- ☑ Imported entity assignments replace current entity assignments

NOTE
Please note: To compile a circuit with encrypted VHDL modules, the associated license file must be installed.

In the "Tools" menu in Quartus II, click on the "License Setup" entry. The "Options" dialog box is displayed. Under the entry "License file", select the associated license file:



License file: C:\Data\Altera\standard\license\DSL_(M320_55)_encr_license.dat

□ Use LM_LICENSE_FILE variable: C:\Temp\mxe_3_6.5c\MXE\license.dat

Current License

License Type: Web Edition

Subscription Expiration: 2009.10

Host ID Type: Not found

Host ID Value: Not found

□ Wait for floating licenses

Licensed AMPP/MegaCore functions:

Vendor	Product	Version	Expiration	Count	Hostid Value	Hostid Type
2742	A5A5	2099.01	01-jan-2099	Fixed		Any

Local System info

Network Interface Card (NIC) ID: 00306eb046c5

C: drive serial number: 94de6ae5

Software Guard ID: Not found

10 DSL component interoperability

To support future implementations of this technology especially with interchangeable components SICK provides a description of a preferred HDSLsystem configuration. This contains information of components, communication as well as installation conditions for a HIPERFACE DSL® Motor feedback system.

By reason of the wide range of different application conditions, e.g. for machine tooling, automated handling or robots, not all particular requirements can be foreseen and considered within one system configuration. It is within the obligation of the user to verify the presented preferred system configuration for particular application requirements and carry out necessary changes. In case of questions or concern SICK will provide further support for implementation and application assessment of the HDSL-technology.

The HDSL-system consists of “internal” and “external” components. The “internalHDSL-chain” contains the following main parts:

- Servo controller (with IP-core)
- Connection line (with cables and connectors)
- Servo motor (with encoder)

These three parts belong together and for a reliable running system all of them need to be considered and optimized as one system during design and realization.

The “external” contributor which impacts a stable and reliable function of the HDSLsystem is the Installation site (application & environment).

Conditions can vary significant between different installation sites worldwide, especially for power supply and grounding conditions but not limited to them. Within (Whitepaper Doc. 8018857) information are provided about potential subjects which need to be considered and checked for a successful installation.

10.1 Servo controller recommendations

The servo controller part within the system configuration contains software routines as well as hardware design and components for interface and connections. Within the servo controller the master part of the HDSL system – the IP-core – needs to be integrated.

10.1.1 IP-Core

Based on field experiences and customer feedback SICK continuously improves the HDSL technology. The information presented within this document are based on an IP-core version of 1.06 at least.

10.1.2 Position Estimator

For an increased system stability and robustness against short time fast position losses caused by multiple reasons the system contains the position estimator functionality. As soon as the system recognizes invalid position information the fast position is provided by the position estimator, which is indicated to the user. For a stable and optimized system function the SICK-position estimator shall be used which is integrated within the IP-core. Based on system features and application requirements the position estimator can be configured by the help of the parameter “max system acceleration” (MAXACC-register) and the “max acceptable position deviation” (MAXDEV-register/dev_thr_errsignal). For further information please refer to chapter 7.3.1.

10.1.3 Event handling

Beside the position data the HDSL-system provides a lot of information to the user [servo controller]. This contains information of the operation conditions, connection performances as well as error information. These information are available as bit settings within different registers as well as digital output signals of the IP-core. Within (SICK-HIPERFACE DSL_event_list; Doc.E_134759) the different information are classified into critical, major or minor events. Furthermore a first root cause information is provided and a first servo controller response. For a stable system during normal operation and start-up the event monitoring shall be done as described within (SICK-HIPERFACE DSL_event_list; Doc.E_134759) and in chapter 7.2. For non-safety operation applications the fast position information needs to be watched only. As long as the fast position is valid (within the limits of the position estimator set limits) there is no loss of accuracy in motor controlling and the system can ride through any error situation within the set tolerances (see position estimator at chapter 7.3.1).

10.1.4 Resources

Encoder functions and information are defined as resources. Each individual resource is identified by a unique resource index (RID). “Long message” transactions enable access to all resources installed in a HDSL motor feedback system. For further information of the detailed resources handling please refer to chapter 8. There are mandatory resources which are available at all of the HDSL-encoder types. A few of the resources vary for the different types of the encoders. For the available resources of each encoder family please refer to the according product information document. At the initial start-up of the system the servo controller needs to read out the actual encoder configuration to identify available encoder functionality (see also chapter 10.1.5).

10.1.5 Encoder identification

Encoders can have different configurations (resolution, single/multi-turn, motor temperature reading) or different internal information based on their type (optical or capacitive). When starting communication initially the electronic type label of the encoder need to be read to configure the servo controller for the particular encoder:

- Resolution (steps per turn)
- Range (revolutions)
- Motor temperature reading function
- Encoder type (to differentiate between standard and Safety version)

Based on the actual encoder configuration the according controller settings need to be done.

10.1.6 Motor commutation

For reading or setting motor commutation no other procedure than laid out for motor manufacturers in chapter 10.1.6 must be used by the drive controller.

10.1.7 Interface circuit

In chapter 4.1 an electrical schematic for the interface circuit is provided with a BOM of the needed components. Furthermore a description is given with layout design assessments and advices when designing the PCB-layout and choosing the different components therefore.

10.1.8 DSL line connection

With the HDSL-technology data wires are routed together with motor power lines within one hybrid cable type. Within the cable certain measures are taken to prevent the data lines from disturbing electrical noise. At the connection point the data wires need to be separated from the motor lines and connected to the controller in such a way that potential EMC-noise coupling is avoided. Depending on various experiences and needs different styles of connector were developed and are in use successfully. figure 42 below shows the principle connection configuration at the servo controller with a separate & shielded data line connection.

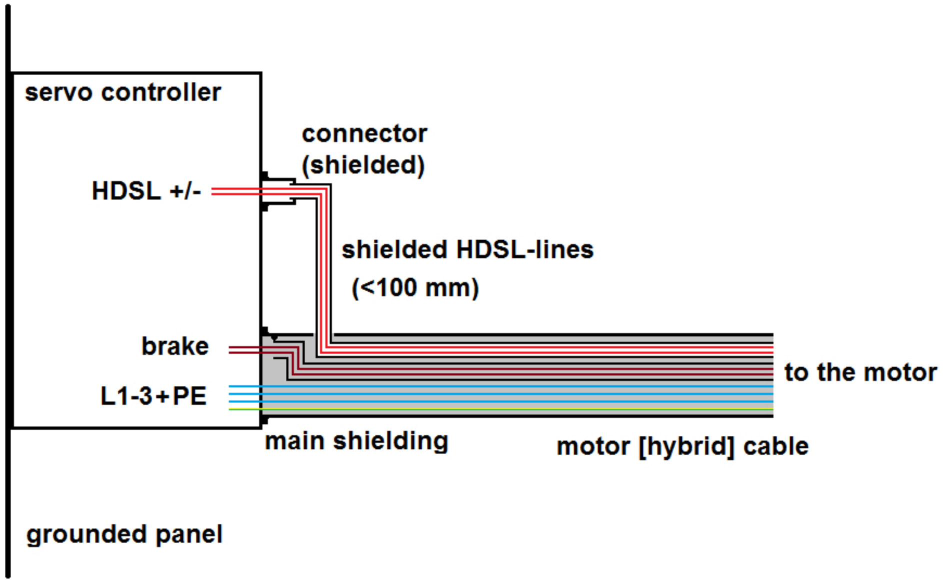


Figure 42: Connection of the hybrid motor cable to the servo controller

The shielded data wires are separated from the other two cable shielding. The data line shielding is connected to a metallized connector housing. Through the connector housing this shielding is connected to the servo controller ground connection. Within this connector the two data wires are routed to the servo controller. So the data lines outside the cable are well shielding, length should be <100 mm. The cable main shielding is connected together with the brake line shielding to the servo controller via large areal contact. It needs to be considered that this ground connection has a low impedance to the controller ground connection and further. It needs to be assured that EMC noise is well drained and not bounced back into the cable. For installation grounding condition please refer also to chapter 10.4.

10.1.9 Verification

To check and verify the successful integration of the HDSL-technology into the servo controller different tests can be performed.

Event handling:

- During IP-core integration and programming the event handling can be checked by simulating encoder events by the help of the PGT-11-S up to a certain extent. Then the desired system responses can be observed and verified. For more details please see PGT-11-S Doc. 8017723.
- By generating error situations (mechanical shock by hitting the motor shaft e.g.) the functionality of the position estimator and its parameter settings can be checked. Depending on the parameter settings for the position estimator the system must be able to cover a certain period of time with invalid fast position information.

EMC noise impact:

- With an oscilloscope and differential probes the EMC noise level on the HDSL-lines can be checked. Within (HIPERFACE DSL® field test & analysis – Possibilities and limits Application Note Vers. 12-00; Doc. E_148498) some test possibilities are described. This can be done during different servo controller operation modes like acceleration, deceleration, stand still, reverse cycling at free running mode or with motor load.
- By modifying connections and/or shielding performances changes can be checked for optimization purposes.

10.2 Motor recommendations

The servo motor part within the system configuration contains the motor connector, the encoder connecting set and the encoder as well.

10.2.1 Motor connector

In principle the information provided at chapter 10.3.3 apply here, too. Depending on the motor power size currently the two different sizes are in use: M17 and M23, the last one is the mainly used size. For interchangeable components the 9-pin layout version should be used as shown at chapter 10.3.3 (e.g. Intercontec connector series 923)

For both the connector sizes there are the same criteria applicable:

- Keep open (= unshielded) wire lengths as short as possible, especially for the data wires (< 20 mm).
- Keep data wires well separated from power and brake wires, no cross overs between data and other wires.
- Connect the shielding via areal contact to the connector (no pigtails).
- Data line shielding gets its own contact through the connector.

For connecting and assembling the particular connector follow the according instructions of the manufacturer.

10.2.2 Encoder connection set

With the encoder connecting set the encoder is connected to the motor connector inside the motor. SICK provides this configured cable set as an accessory. Here the wires are already assembled to the connector for the HDSL-connection to the encoder. For the connection between the encoder to the motor connector the length shall be minimized without any loops, unused wire lengths shall be cut-off. By using a connection set without shielding which is typical for small and midsize servo motors (frame sizes up to 130 mm) the data line shielding at pin F needs to be connected to PE at the motor housing. When using a shielded connection set (for connections lengths >200 mm) this shielding is connected to pin F. In this case the connector series 723 (of Intercontec e.g.) is used for the motor connection (at the motor and cable as well) to optimize data line shielding performance. At the encoder side this shielding is connected to the encoder housing and by special measures inside the encoder to PE finally.

10.2.3 Encoder mounting

For a proper and long reliable function the encoder needs to be assembled according the information within its operation instructions. If necessary the required special tools shall be used and the listed torque setting. Tools with adjustable torque setting shall be used for repeatable mounting conditions. Motor temperature sensor leads shall be cut to a reasonable length before configured with the connector to connect to the encoder. Long loops of these leads shall be avoided.

10.2.4 Motor commutation

For an optimized motor power control a defined rotor angle needs to be set to certain encoder position information. For interchangeable components this position is always set to “0”. To commutate the motor the encoder is mounted within the motor and connected to the PGT or a comparable IP-core system within a production system (in Master Mode). A DC voltage is supplied to the motor phase connections (V – +24 VDC; U & W – -24 VDC), which forces the rotor of the motor (and the encoder) into a certain position. The power level may vary depending on the motor size. This encoder position is set to “0” by the help of the “SETPOS”-command (RID 101h). Therefore please follow the instruction within [2]. If the command was carried out correct the position reading from the encoder must be “0”. With a position jitter the actual value can fluctuate and numbers of around 262144 (18 bit singleturn-type) or 1073741824 (18 bit multiturn-type) can appear, too.

10.2.5 Motor type label

For an optimized motor controlling and operation the servo controller needs particular information about the connected motor. For interchangeable configurations (different motors brands at a controller) a default set of data is stored at the encoder within an electronic motor type label. The use of additional manufacturer-specific data files is not limited by this requirement. Within an electronic motor type label the following data could be listed (extracted from existing electronic motor type labels) as subject for further discussions. This example data file has the following specification:

File name: INTEROP

File size: t.b.d.

File content:

Table 207: Example for motor type label

Information	Unit	Range	Offset [bytes]	Length [bytes]
Number of pole pairs	-	0 ... 255	0	1
Rate speed nN	min^{-1}	0 ... 65535	1	2
Standstill current I0rms	A	0 ... 255	3	1
Peak current I0max	A	0 ... 255	4	1
Rated voltage Vn	V	0 ... 65535	5	2
Rated torque Mn	Nm	0 ... 65535	7	2
Rotor moment of inertia J	kgcm^2	0 ... 65535	9	2
Thermal time constant tTH	-	0 ... 255	11	1
Max. brake voltage Vbkmax	V	0 ... 255	12	1
Brake release delay time tBRH	ms	0 ... 255	13	1
Temperature sensor coefficient ΔR/Δδ	Ω/K	0 ... 255	14	1

By reading these information the servo controller can automatically configure a standard motor controlling performance with the actual connected motor. The HDSL encoders provided an EEPROM storage area of 8 kByte for external usage organized by file system. In this case several files can be stored there. Access to this storage area is provided by different resources.

- “MAKEFILE” – RID 133h – create, delete, change file
- “LOADFILE” – RID 130h – load an existing file for access
- “RWFILE” – RID 131h – read or write access
- “FILESTAT” – RID 132h – information about access rights and file size
- “DIR” – RID 134h – provides the list of existing files

For further information of the file handling please refer to section chapter 8.10.

10.2.6 Verification

For the proper encoder assembling a visual inspection is done. In case of problems (vibrations, e.g.) acceleration sensors can be placed on the encoder and the vibration level can be checked (in reference to the limits shown within the product information). To check the proper mounting of the encoder to the motor shaft an external reference encoder is used. Its position information is compared to them of the encoder within the motor. If a potential deviation between the two positions data exceeds a certain limit the encoder mounting needs to be checked and reworked if necessary. The proper shielding and grounding is tested with an oscilloscope and differential probes during different drive modes and observing the EMC noise coupling or disturbances on the data lines.

10.3 Recommendations for connection line

The connection line part within the system configuration contains the cable between the servo controller connection point and the motor connector as well as potential couplings within the line.

10.3.1 Cable

The combination of data lines and motor power lines within one cable is done with the so-called hybrid cable type. Herein the data lines are protected against electromagneticdisturbances by different measures as good as possible. To be suitable for a stable and reliable HDSL-communication the cable need to fulfill certain criteria.

Within [Whitepaper Vers. 2-03; Doc. 8018816] the basic cable performance requirements are listed. Nowadays more or less all bigger cable manufacturer and supplier have one or morehybrid cable types within their product portfolio. When choosing a particular cable the manufacturer data need to be compared with the requirements, listed within [Whitepaper Vers. 2-03; Doc. 8018816].

Further criteria for the cable selection are listed within [Whitepaper Vers. 1-04; Doc. 8018817]. These criteria are mainly driven by requirements and conditions of the different installation sites (environmental and legal issues, e.g.).

10.3.2 Couplings

Couplings are installed to separate different equipment sections from each other mainly for shipment and installation purposes. At coupling points the cable needs to be opened. Thereby all the measures to protect the data line have no longer any effect. The general recommendation is to avoid couplings within the connection line. If coupling points are necessary the number shall be kept as low as possible – one should be suitable for most of the cases. Requirements for couplings:

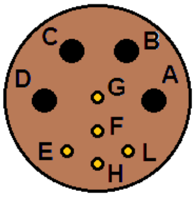
- Keep open (= unshielded) wire lengths as short as possible, especially for the data wires (<20 mm)
- Keep data wires well separated from power and brake wires, no wire crossing
- The shielding of the data lines needs to be separated from the other shielding
- no pigtails for any shielding connection
- A motor connector type coupling shall be used instead of terminal boxes.

10.3.3 Connector to motor

Up to some extent the motor [power] size defines the type [size] of connector used for motor connection to the hybrid cable. Currently two different sizes are in use: M17 and M23, the last one is the mainly used size. The M17 connector size is used for very small motors only (typical frame size 55 mm and smaller). For both the connector sizes there are the same requirements applicable:

- Keep open (= unshielded) wire lengths as short as possible, especially for the data wires (<20 mm)
- Keep data wires well separated from power and brake wires, no cross overs between data and other wires
- Connect the shielding via areal contact to the connector (no pigtails)
- Data line shielding gets its own contact through the connector. For connecting and assembling of the particular connector follow the according instructions of the manufacturer. For interchangeable components (motor/cable) the pin layout as shown at the figure below shall be used (9-pin-M23 size motor connector).

pin orientation:
view on the cable connector



pin code: A - U
B - V
C - W
D - PE
E - data+
H - data-
F - data shielding
G - brake+
L - brake-

Figure 43: Pin layout M23

Pin layout for a M23 size motor connector. Main & brake line shielding is connected together to the connector housing; within the connector also PE is connected to the connector housing; there is a separate feedthrough of the data line shielding. When using a shielded configured encoder connecting set the shielding of this set is connected to the according pin on the motor part of the connector.

10.3.4 Verification

Within [HIPERFACE DSL field test & analysis – Possibilities and limits Application Note Vers. 12-00; Doc. E_148498] there are tests listed to check the performance of the connection line, especially during field inspection. These tests are also applicable to check and verify a system configuration initially. Signal transmission performance:

- Evaluation of the contents for registers RSSI, EDGE, DELAY, QM.
- Monitor the digital IP-core output signals `estimator_on` and `encoding_err`
- Monitor the online_status_d register bit for POS, DTE and QMLW

For the exact meaning for the content of these register and threshold values please refer to chapter 6.3.3, chapter 6.3.7, chapter 6.3.8. The signal propagation delay within the connection line will be adjusted within the HDSL-protocol frame in steps of approx. 106 ns (~10m-steps). Intended connection lengths shall be configured and tested. Results can be compared with the content of the DELAY-register.

EMC noise impact:

With an oscilloscope and differential probes the EMC noise level on the HDSLlines can be checked. The probes are initially connected at the HDSL-inlet point at the servo controller. Tests can be done during different servo controller operation modes like acceleration, deceleration, stand still, reverse cycling at free running or with motor load.

10.4 Recommendations on installation site

As mentioned within chapter 10.1 also external conditions influence the proper function of the HDSL-system. Depending on foreseeable target applications known conditions can be considered during the system design process. Nevertheless the conditions can vary significantly between different installations sites. By reason of the great variety of different conditions there are no single HDSLsystem configuration which will fit all needs. In such cases the differences need to be assessed and the HDSL-system configuration can be adjusted to particular requirements. The two important external impacts to the HDSL-system performance are

- The power supply and grounding conditions
- Application changes and field modifications

10.4.1 Power supply and grounding conditions

Most power supply ratings for current HDSL-servo motor applications are either 400 VAC/3~ or 480 VAC/3~. The difference between these two level leads to a different level of DC voltage for PWM controlling of the motors of either approx. 570 VDC or approx. 700 VDC. In combination with different rising edges for the PWM this leads to significant different EMC conditions especially within the connection line between the servo controller and the motor. This needs to be considered when adapting a HDSLsystem configuration for one power supply system to the other one. The impact of the grounding condition is caused either by insufficient contact qualities or wire sizes. It leads to noise reflections instead of noise draining. Depending on noise frequencies and cable lengths it can amplify noise amplitudes up to a disturbing level. Large distances between controller and motor location or different ground connections points can cause ground loops.

11 Index

12 Glossary

8B/10B	8 bit/10 bit code (line code for transmission of 8 bits with data in 10 bit lengths to achieve DC balance)
CRC	Cyclic Redundancy Check (algorithm to determine data checksum) DSL Digital Servo Link, complete name: HIPERFACE DSL®
DSL	Digital Servo Link
EDIF	Electronic Design Interchange Format (format for electronic exchange of FPGA netlists)
FIFO	First in – First out (storage method in which the first stored elements are the first to be discarded)
FPGA	Field Programmable Gate Array (programmable digital logic component)
IP Core	Intellectual Property Core, for integration into ICs or chip provided for FPGAs
Long Message	Protocol component for polling parameter data of an encoder that must first be processed by the encoder.
Motor feedback system	Rotary or linear encoder for use in servo drives
RS485	Radio Sector Standard 485 (also designated as EIA-485 or TIA- 485-A standard for serial data transmission over symmetric pair cables)
RSSI	Received Signal Strength Indicator
SensorHub	Interface between a motor feedback system and an external sensor component in a drive system
Short Message	Protocol component for polling directly transmitted parameter data of an encoder
SPI	Serial Peripheral Interface (serial bus system for digital switching)
VHDL	Very high speed integrated circuit Hardware Description Language (hardware abstraction language for FPGAs)

13 Versions

Table 208: Document versions

Date	Version	Change
6/27/2014	00	First issue (replacement of HIPERFACE DSL® manual 8013607 and IPCore DSL Master Manual 8013736)
8/21/2014	01	Adapted parameter channel error handling description to version 1.06 of the Hiperface DSL® IP Core (RET flag no longer available). Part MAX13431E no longer suggested due to decreased availability.
5/18/2015	02	Update of safety requirements (removal of diagnostic tests on position vector length check) Clarifications on position usage for safety functions Clarifications on usage of POSTX Addition of documentation on function to reset user file system. Addition and clarification of functions in regard to new DSL products EFx50 and EEx37 (LED current, position filter).
6/6/2016	03	Update to DSL Master version 1.07: Bugfix related Test messages and POSTX signals. Addition of RELEASE register. Addition of interface circuit design recommendation Adjustments of resources and errors for different encoder types EKx/EFx/EEx (LED Current, Rotor Position) Addition of recommended error handling for different encoder types EKx/EFx/EEx Revision on Safety requirements Addition of the feature "Mandatory yes/no" for each resource
1/15/2018	04	Improvement of error handling list see table 42

8017595/ZTW6/2018-01-15

Australia
Phone +61 3 9457 0600
1800 334 802 – tollfree
E-Mail sales@sick.com.au

Austria
Phone +43 22 36 62 28 8-0
E-Mail office@sick.at

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

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

Canada
Phone +1 905 771 14 44
E-Mail information@sick.com

Czech Republic
Phone +420 2 57 91 18 50
E-Mail sick@sick.cz

Chile
Phone +56 2 2274 7430
E-Mail info@schadler.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 +358-9-2515 800
E-Mail sick@sick.fi

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

Germany
Phone +49 211 5301-301
E-Mail info@sick.de

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

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

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

Israel
Phone +972 4 6881000
E-Mail info@sick-sensors.com

Italy
Phone +39 02 274341
E-Mail info@sick.it

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

Malaysia
Phone +6 03 8080 7425
E-Mail enquiry.my@sick.com

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

Netherlands
Phone +31 30 2044 000
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 +40 356 171 120
E-Mail office@sick.ro

Russia
Phone +7 495 775 05 30
E-Mail info@sick.ru

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

Slovakia
Phone +421 482 901201
E-Mail mail@sick-sk.sk

Slovenia
Phone +386 591 788 49
E-Mail office@sick.si

South Africa
Phone +27 11 472 3733
E-Mail info@sickautomation.co.za

South Korea
Phone +82 2 786 6321
E-Mail info@sickkorea.net

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 2645 0009
E-Mail Ronnie.Lim@sick.com

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

United Arab Emirates
Phone +971 4 88 65 878
E-Mail info@sick.ae

United Kingdom
Phone +44 1727 831121
E-Mail info@sick.co.uk

USA
Phone +1 800 325 7425
E-Mail info@sick.com

Vietnam
Phone +84 945452999
E-Mail Ngo.Duy.Linh@sick.com

Further locations at **www.sick.com**

SICK AG | Waldkirch | Germany | www.sick.com