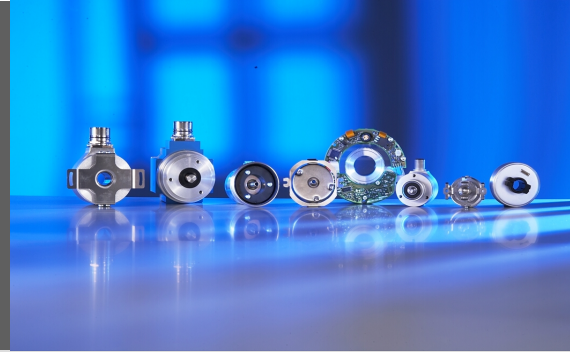


HIPERFACE DSL



HIPERFACE DSL INTERFACE

E

HIPERFACE[®]
DSL

©SICK STEGMANN GmbH

All rights reserved. No part of the description must be reproduced or further processed in any form without the company's consent.

This documentation refers to HIPERFACE DSL® version 3.13, release date 2008-05-28.

Subject to change without prior notice.

SICK STEGMANN GmbH accepts no warranty for non-infringement of patent rights e.g. in case of suggestions for circuitry or processes.

Trade good names listed are the property of the companies concerned.

HIPERFACE® and HIPERFACE DSL® are registered trademarks by SICK STEGMANN GmbH.

SICK|STEGMANN GmbH
Dürrheimer Straße 36
D-78166 Donaueschingen
Fon: (49) 771 / 807 – 0
Fax: (49) 771 / 807 – 100
Web: <http://www.sick.com>
E-mail: Info@sick-stegmann.de

Made in Germany, 2010.

Table of Contents

1.	Scope of this Document	4
1.1.	Used symbols	4
1.2.	HIPERFACE DSL® for motor-feedback	5
1.3.	HIPERFACE DSL® features	6
2.	Protocol Overview	8
2.1.	Horizontal Channel	11
2.2.	Vertical Channel	11
2.3.	Vertical Channel 2	12
2.4.	Parameter Channel	12
2.5.	Pipeline Channel	13
3.	Hardware Implementation	14
3.1.	Interface Circuit	14
3.2.	FPGA IP-Core	17
3.3.	Cable Specification	21
4.	Interfaces	23
4.1.	SPI1 Interface	24
4.2.	SPI2 Interface	29
4.3.	SPI-PIPE Interface	30
4.4.	SYNC Interface	32
4.5.	INT Interface	32
4.6.	LINK Interface	32
4.7.	Reset	33
5.	Register map	34
5.1.	Register Legend	36
5.2.	Online DSL-Master Status	37
5.3.	DSL Master Function Registers Channel 1	41
5.4.	DSL Master Function Registers Channel 2	66
5.5.	Function Registers DSL Slave	71
6.	Central Functions	75
6.1.	System Start	75
6.2.	System Diagnostics	76
6.3.	Incremental Position	78
6.4.	Absolute Position Channel 1	81
6.5.	Absolute Position Channel 2	82
6.6.	Status and Error Messages	83
6.7.	Parameter Channel	90
7.	Resources Motor-feedback System	99
7.1.	Resource Access	99
7.2.	Resource List	102
	Glossary	150

1. Scope of this Document

1.1. Used symbols



Note/Tip

Notes inform you about particularities of the device. Please keep them in mind, they often contain important information.

Tips are offering extra information, which relieve the work with the documentation.



Safety Instructions!

Safety instructions advise you of concrete or potential dangers, or wrong treatment of the application. This should protect you from harm.

Read and obey the safety instructions accurately.

1.2. HIPERFACE DSL® for motor-feedback

This document describes the use and implementation of the data protocol HIPERFACE DSL® as implemented in motor-feedback systems in the servo drive market.

HIPERFACE DSL® is a purely digital protocol needing a minimum of connection lines between drive and motor-feedback system. The robustness of the protocol allows the connection to the motor-feedback system signals within the motor cable.

Motor-feedback systems with the interface HIPERFACE DSL® can be used for all performance ranges and simplify the implementation of an encoder system within a drive significantly:

- Uniform digital interface (RS485).
- Analog components for encoder interface become obsolete.
- Standardized interface between drive application and protocol logic.

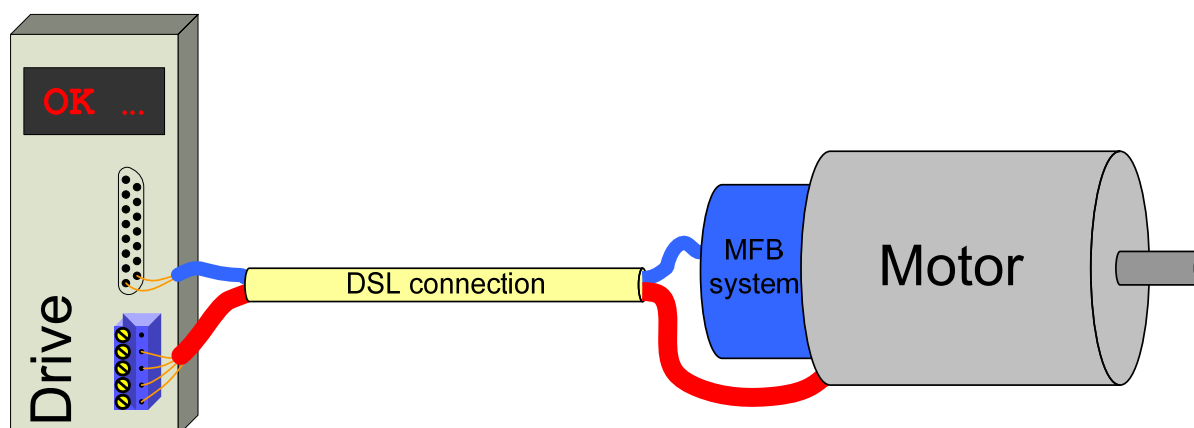


Figure 1. Drive system with HIPERFACE DSL®.

Based on the name of the predecessor protocol HIPERFACE® from SICK STEGMANN the name HIPERFACE DSL® stands for High PERFORMANCE InterFACE Digital Servo Link.

This interface provides for all current requirements on digital motor-feedback systems and additionally features future extensions for drive manufacturers.

1.3. HIPERFACE DSL[®] features

Some of the main advantages of HIPERFACE DSL[®] result from the possibilities to connecting the encoder:

- One digital interface on the drive for the complete communication to the motor-feedback system. The interface conforms to the RS485 standard with a transmission rate of 9.216 MBaud.
- Communication to the encoder on a single line pair
- Encoder power supply and communication can be placed on the same line pair. This is enabled by adding one pulse transformer component to the drive.
- Connection lines to the encoder can be placed as a shielded twisted pair line within the motor supply cable. This eliminates the encoder connector on both motor and drive.
- Cable length between drive and motor-feedback system can be up to 100 m without derating of performance.

The digital protocol HIPERFACE DSL[®] can be used in a variety of drive applications:

- Cyclical data transmission synchronized to the feedback cycle of the drive. This allows synchronous processing of encoder position and speed.
- Cycle times down to 12.15 μ s can be supported.
- Transmission of absolute position of the motor-feedback system with a maximum cycle time of 195 μ s.
- Redundant transmission of absolute position with a maximum cycle time of 195 μ s for the use of suitable motor-feedback systems in SIL2 applications (according to IEC 61508).
- Transmission of second channel absolute position with a maximum cycle time of 195 μ s for the use of suitable motor-feedback systems in SIL3 applications (according to IEC 61508).
- Parameter channel for bidirectional general data communication with a bandwidth of up to 340 kBaud. This data includes an electronic type label for identification of the motor-feedback system and for storage of drive-related data in the motor-feedback system.
- Pipeline channel for transmission of data from external sensors related to the motor that are linked to the motor-feedback system via the Hiperface DSL[®] SensorHub protocol.

The protocol is integrated into the drive as a hardware logic design. This logic design is supplied as an IP core for Field Programmable Gate Array components (FPGA) of various manufacturers.

- Net list for protocol logic allows free routing of the implementation of the HIPERFACE DSL® IP core. The protocol design can be implemented alongside the drive application on the same FPGA.
- Fast full duplex Serial Peripheral Interface (SPI) between protocol logic and drive application for uniform access to process data (position, speed) and parameters.
- Redundant fast full duplex Serial Peripheral Interface (SPI) between protocol logic and drive application for access to secondary position data.
- Additionally configurable SPI interface for output of data of external sensors.
- Configurable interrupt output.

2. Protocol Overview

Hiperface DSL® is a fast digital protocol for motor-feedback systems that connects servo drives with a motor-feedback encoder. The protocol is implemented in the drive up to the Transport Layer by the use of a digital logic design, the “DSL Master IP core”.

Position data in Hiperface DSL® is generated by one of two ways. Either in the free-running mode where position values are sampled and transmitted as fast as possible or in SYNC mode where position data is sampled and transmitted synchronous to a supplied clock signal. In a servo drive application this clock signal usually is the feedback clock of the drive.

In SYNC mode the protocol will adjust time points for sampling of data without jitter to the supplied drive cycle.

During each drive cycle at least one position value will be sampled and transmitted to the DSL Master with constant latency. Since the protocol adjusts the internal packet transmission rate to the drive cycle the overall transmission rate of Hiperface DSL® depends on the drive cycle.

The protocol packet is adjusted for varying lengths as seen in Figure 2. If the drive cycle is long enough additional sampling points within the drive cycle can be appended, so-called „Extra“ packets. The number of additional packets is programmed by the user with a divider value.



The number of transmitted packets per drive cycle cannot be chosen arbitrarily as the length of a protocol packet must follow minimum and maximum boundaries. This should be considered during setting of the divider value.

If you use the free-running mode the drive cycle is not considered for sampling and transmission and the minimum packet length is used by the protocol.



Please note that the minimum packet length in free-running mode is shorter than the minimum packet length in SYNC mode.

Table 1 shows the dependency of protocol packet lengths for exemplary drive cycle lengths.

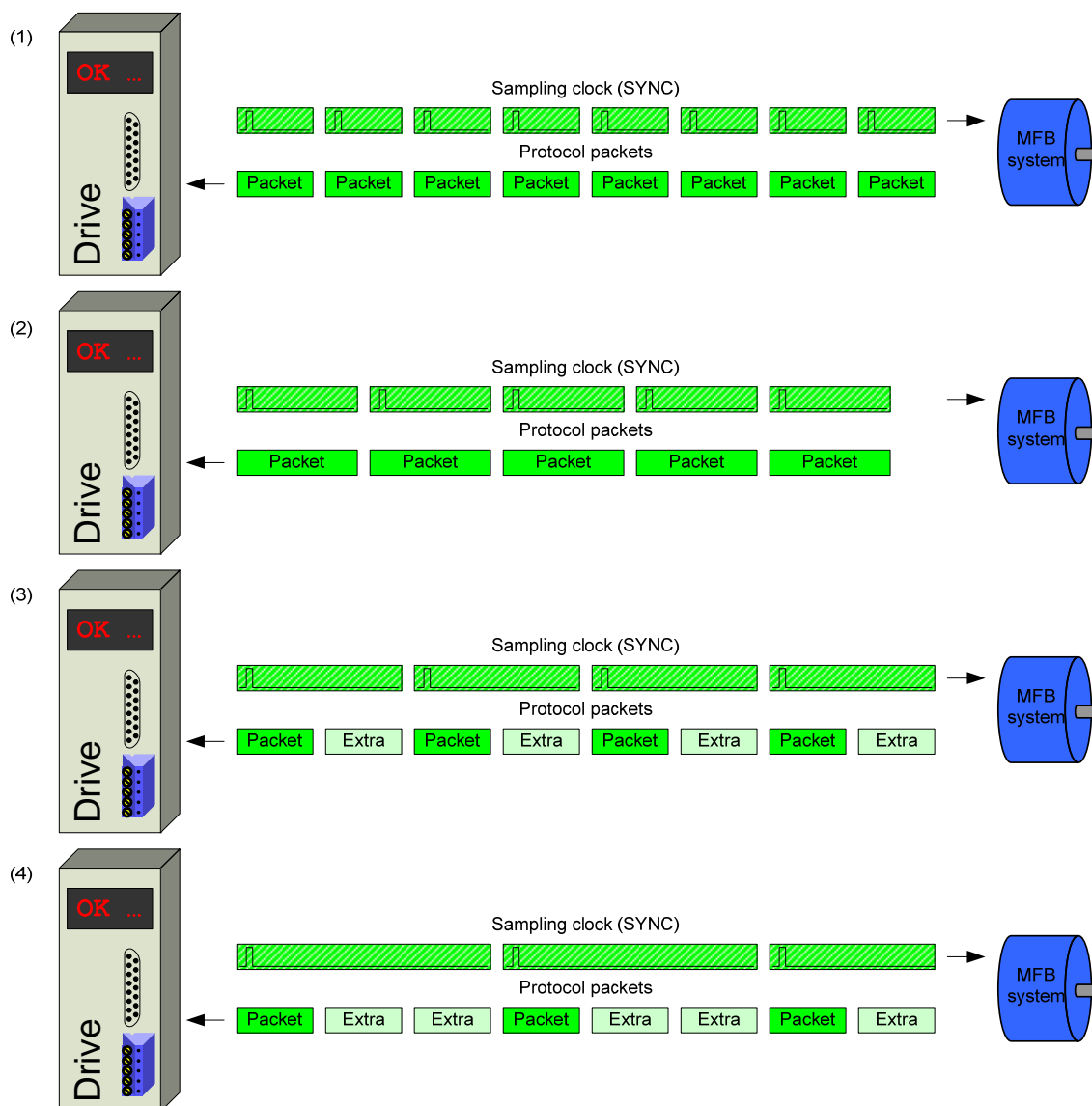


Figure 2. Protocol packet lengths.

Frequency drive cycle[kHz]	Length drive cycle [μs]	Length protocol packet [μs]	Protocol packets per drive 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
41..82	24.3 ... 12.15	24.3 ... 12.15	1
Free-running	--	11.72	--

Table 1. Drive cycle and protocol packet lengths.

HIPERFACE DSL®

With Hiperface DSL® data is transmitted on several channels. Each channel is adapted to different requirements according to its content. The cycle time of each channel varies with the length of the underlying protocol packets.

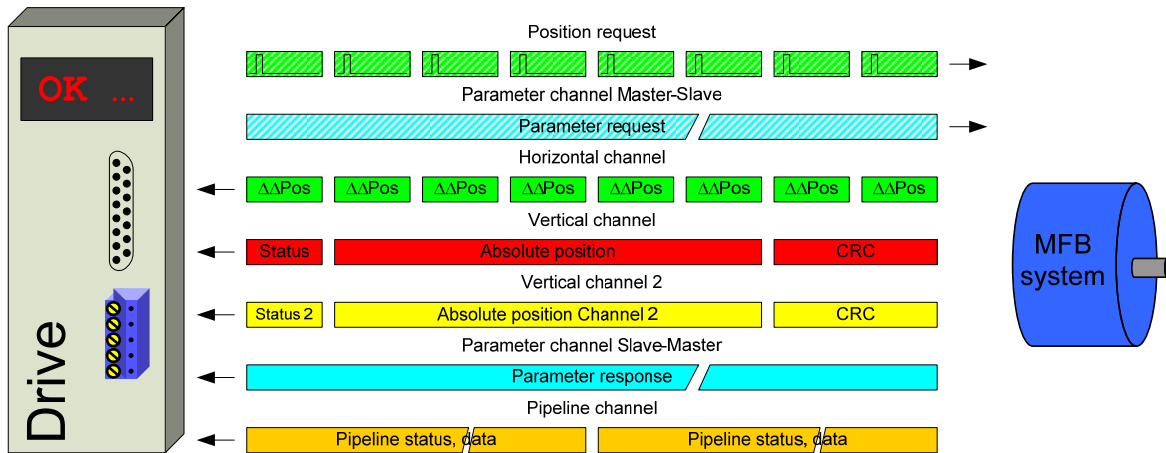


Figure 3. Data channels of Hiperface DSL®.

Table 2 gives an overview of characteristics of the different channels.



Please note that minimum cycle time and maximum bandwidth are only valid if the maximum possible number of sampling points per drive cycle are programmed (see register synchronization control, chapter 5.3.2).

Channel in Hiperface DSL®	Function	Cycle time [μs]	Band width [kBit/s]
Horizontal	Incremental position, speed	12.15 ... 24.3	1316 ... 658
Vertical	Absolute position, Status of channel 1	97.2 ... 194.4	658 ... 329
Vertical 2	Absolute position, Status of channel 2	97.2 ... 194.4	658 ... 329
Parameter	General data, parameters	Varies	329 ... 164
Pipeline	Externally supplied data	12.15 ... 24.3	658 ... 329

Table 2. Protocol data channels.

2.1. Horizontal Channel

In the Horizontal Channel the position value of the motor-feedback system is transmitted synchronous to position requests in line with the drive cycle supplied to the SYNC input.

The Horizontal Channel is the fastest channel of Hiperface DSL® protocol. Each transmitted protocol packet contains a full update of the contents of this channel.

Within this channel the position value is not transmitted absolutely but as an increment. This increment contains information about the second order of difference between the previous and current position values (so-called $\Delta\Delta$ Position).

In the protocol logic this incremental position transmission is integrated after each update to allow access to speed and absolute position values at any time (see chapters 5.3.11, 5.3.12).

These values form the feedback parameters to the control loop of the motor drive.



The transmission of the position as an increment implicates a maximum boundary to the encoder for both speed and acceleration of the motor. These boundaries are listed in Table 6 and must be respected.

2.2. Vertical Channel

In the Vertical Channel the position value of the motor-feedback system is transmitted as an absolute value. Additionally, the status of the encoder in the form of errors and warnings is reported in this channel.

For transmitting a full data packet of the Vertical Channel eight underlying protocol packets are needed.



The position value transmitted on the Vertical Channel is not synchronous to the drive cycle supplied to the SYNC input.

The absolute position value is used by the DSL Master IP core for verifying the incremental position value and can be used for the same purpose by the drive application.

If the absolute and incremental position values are not equal an error message is raised (see chapter 6.6). The protocol will replace the incremental position with the absolute position in this case.

In each packet of the Vertical Channel a summary of status bits is transmitted representing the error and warning states of the motor-feedback system.



Please note that each bit of the status byte of the Vertical Channel is a summary of eight individual states of the motor-feedback system. Each individual state of the encoder can be read out with a „Short Message“ (see chapter 6.7.1).

2.3. Vertical Channel 2

In the Vertical Channel 2 the absolute position value and status of the secondary channel of the motor-feedback system is transmitted. The channel is used for diagnostics in a safety application.

For transmitting a full data packet of the Vertical Channel 2 eight underlying protocol packets are needed.



The position value transmitted on the Vertical Channel 2 is not synchronous to the drive cycle supplied to the SYNC input.

The position value of Vertical Channel 2 is either derived by a second processing channel within the DSL encoder (SIL2 system) or from a separate, second sensor source (SIL3 system).

The position value of Vertical Channel 2 must be used by the user application for cross-checks with the position value of the primary Vertical Channel to detect transmission or sensor processing errors. Safety requirements are detailed in the Hiperface DSL® Safety Implementation Manual.

In each packet of the Vertical Channel 2 a status byte is transmitted representing the error and warning states of channel 2 of the motor-feedback system.

2.4. Parameter Channel

The Parameter Channel serves as interface for reading and writing of parameters of the motor-feedback system by the drive application.

Motor-feedback systems equipped with the Hiperface DSL® interface implement various internal resources apart from the main position measurement. These resources can be accessed through the Parameter Channel.

Examples for these resources are temperature measurements, monitoring mechanisms for correct function, product data (so-called “electronic type label”), or freely programmable data fields.



Please note that actual implemented resources differ between DSL products and are detailed in the respective product data sheet.

There are two ways of communicating on the Parameter Channel:

- “Short Message“ transaction
- “Long Message“ transaction

A “Short Message“ transaction enables access to resources that influence or monitor the Hiperface DSL® protocol interface. This includes detailed status and error messages of the motor-feedback system or indicators of signal strength on the DSL link. Since a “Short Message“ is processed directly by the interface logic of the motor-feedback system this transaction is finished within comparably short time.

A “Long Message“ transaction enables access to all other resources of the motor-feedback system. In contrast to a “Short Message“ a “Long Message“ usually

requires handling by the processor of the motor-feedback system and therefore has a response time that can not be determined beforehand.



Please note that Hiperface DSL® supports processing of a maximum of one “Short Message“ and one “Long Message“ at the same time.

2.5. Pipeline Channel

The Pipeline Channel allows transmission of additional external sensor data that may be used in the drive system. External sensors must be linked to the motor-feedback system via the Hiperface DSL® SensorHub interface. Over this interface various sensors or sensor networks can be accessed and read out through Hiperface DSL®.

Parameterization of external sensor components is conducted over the Parameter Channel while the Pipeline Channel enables reading of the data. Transmission of protocol packets in the Pipeline Channel is synchronous to the DSL transmission and in extension to the drive cycle supplied to the SYNC input of the DSL Master. Depending on the utilization of the SensorHub interface external data therefore can be sampled and transmitted synchronous.

The protocol within the Pipeline Channel is not governed by Hiperface DSL®. Apart from data transmission quality monitoring there are no protocol mechanisms running in this channel.

For a description of the SensorHub interface please consult the manual “Hiperface DSL® SensorHub“.

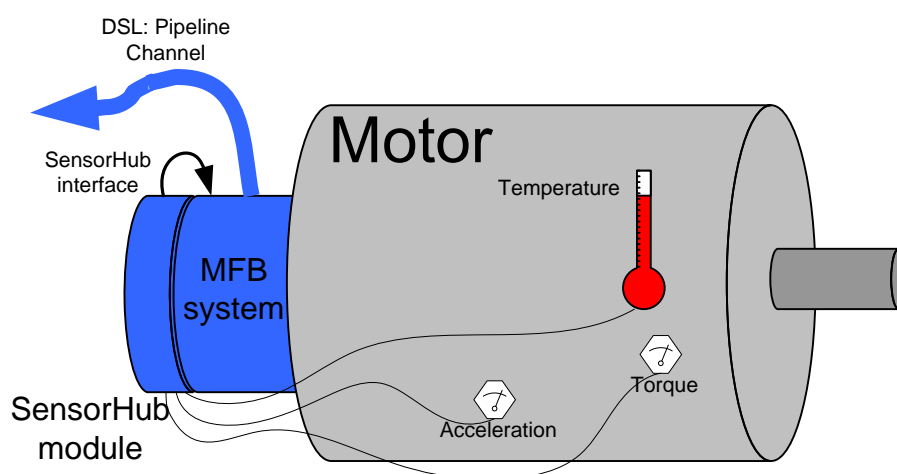


Figure 4. Hiperface DSL® SensorHub interface.

3. Hardware Implementation

The implementation of Hiperface DSL® within a drive system requires the inclusion of an interface circuit with specific components as well as the implementation of a digital logic core for a field programmable gate array device (FPGA).

The interface circuit is specified in this chapter along with suggestions for component choice.

The digital logic core (IP-core) will be supplied by SICK STEGMANN for specified FPGA types.

Additionally, this chapter specifies the type of cable that is suggested to be used as connection between the drive and the motor-feedback system.



The use of other cables might be possible but should be validated before deployment.

3.1. Interface Circuit

Hiperface DSL® allows the use of three different interface circuit scenarios. Each scenario requires a different kind of connection cable (see chapter 3.3).

3.1.1. Separate Encoder Cable

If a separate encoder cable is used (see chapter 3.3.1), the smallest interface circuit can be used. The separate encoder cable scenario requires the use of a four wire connection.

The following Figure 5 together with Table 3 shows the interface circuit specification.

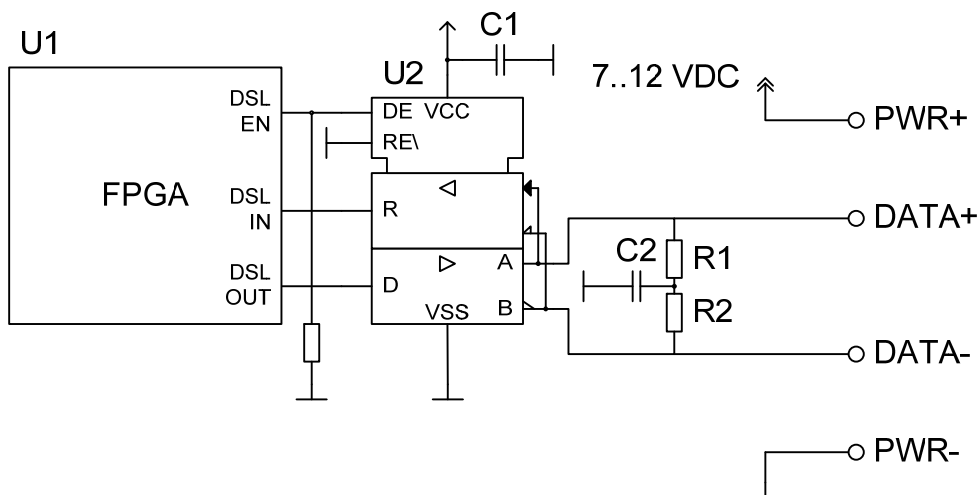


Figure 5. Separate encoder cable interface circuit.

Suggested components for the interface circuit are listed in Table 3.

Component		Part	Manufacturer
C1	Ceramic Capacitor	100 nF	
C2	Ceramic Capacitor	2.2 μ F 16V	
R1, R2	Resistor	56R	
U1	FPGA	XC3S250E (or larger)	Xilinx
U2	RS485 Transceiver	SN65LBC176 MAX13431E	Texas Instruments Maxim IC

Table 3. Separate encoder cable interface circuit components.

3.1.2. Integrated Cable – 4-Wire

If a four-wire connection is used that is integrated into the motor cable (see chapter 3.3.2) a pulse transformer must be added to the data lines to increase the common mode rejection ratio.

The following Figure 6 together with Table 4 shows the interface circuit specification.

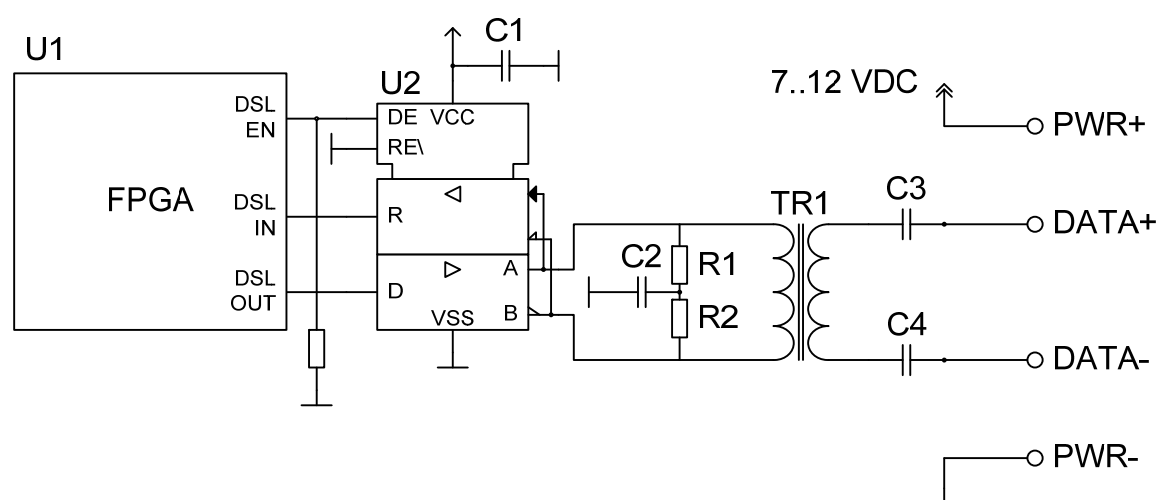


Figure 6. Integrated cable 4-wire interface circuit.

Suggested components for the interface circuit are listed in Table 4.

Component		Part	Manufacturer
C1	Ceramic Capacitor	100 nF	
C2	Ceramic Capacitor	2.2 μ F 16V	
C3, C4	Ceramic Capacitor	470 nF 50V	
R1, R2	Resistor	56R	
U1	FPGA	XC3S250E (or larger)	Xilinx
U2	RS485 Transceiver	SN65LBC176 MAX13431E	Texas Instruments Maxim IC
TR1	Pulse Transformer	B78416A8607A003 B78304B1030A003 78602/1C	Epcos Epcos Murata

Table 4. Integrated cable 4-wire interface circuit components.

3.1.3. Integrated Cable – 2-Wire

If a two-wire connection is used that is integrated into the motor cable (see chapter 3.3.3) a pulse transformer must be added to the data lines to increase the common mode rejection ratio. Also supply voltage injection on the data lines necessitates the addition of choke coils.

The following Figure 6 together with Table 5 shows the interface circuit specification.

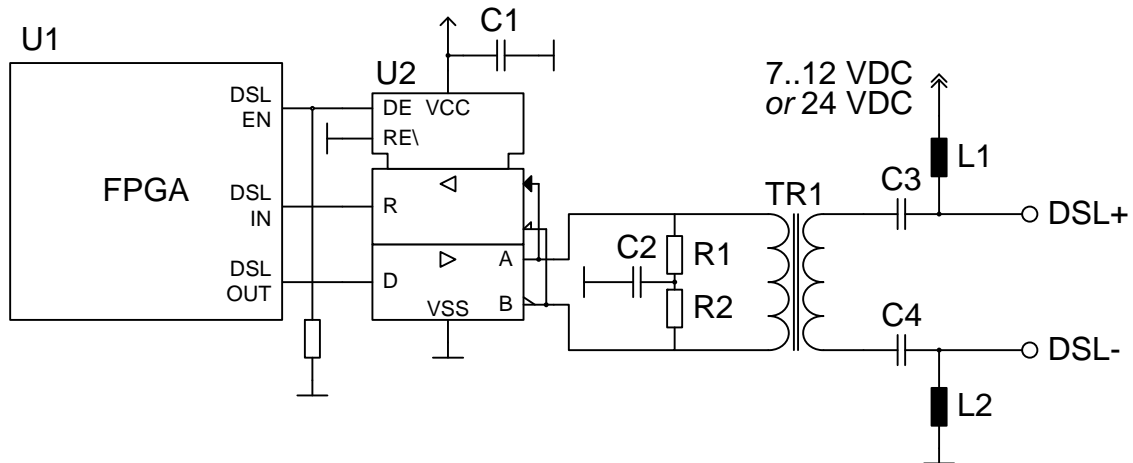
HIPERFACE DSL®

Figure 7. Integrated cable 2-wire interface circuit.

Suggested components for the interface circuit are listed in Table 5.

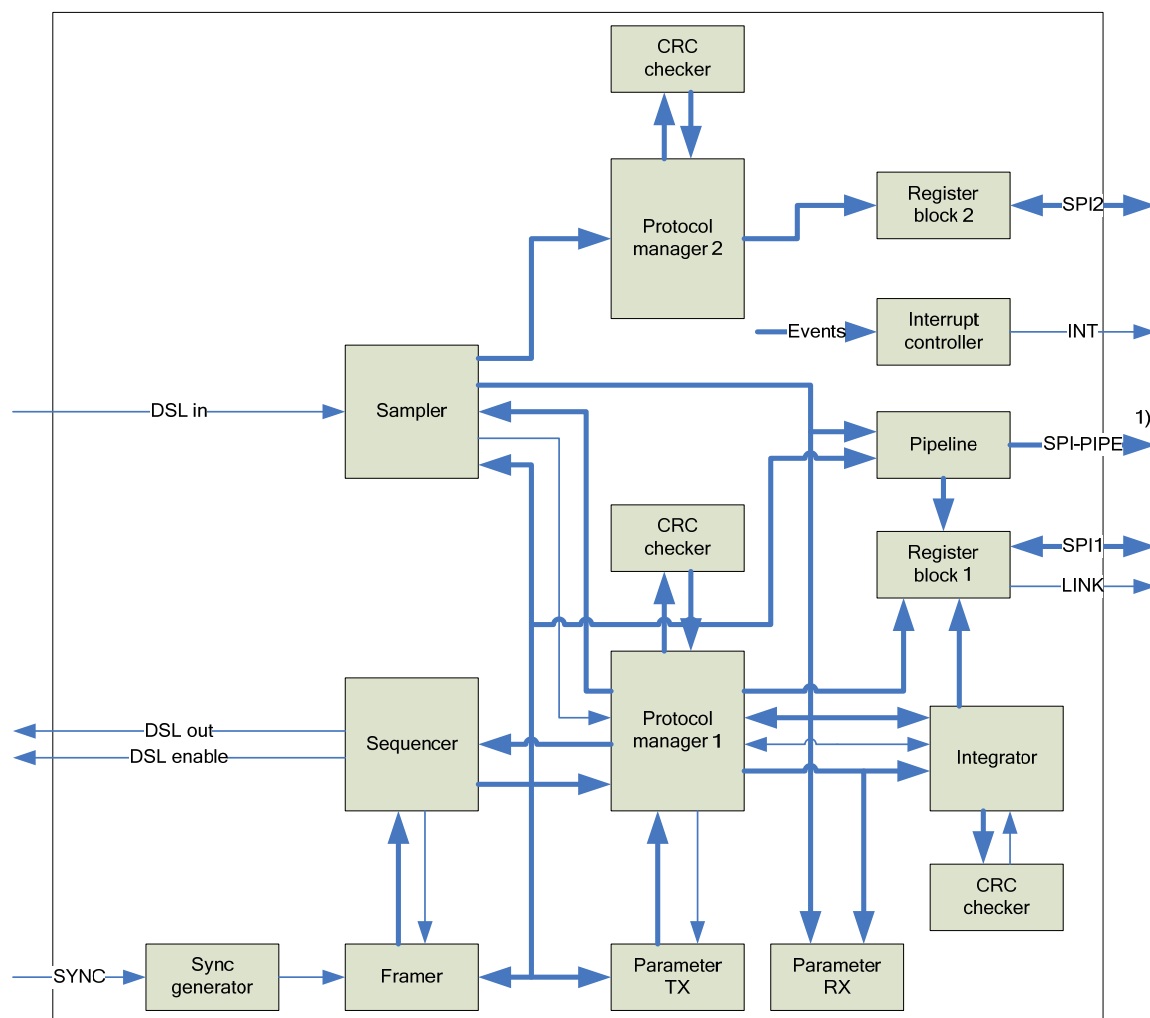
Component		Part	Manufacturer
C1	Ceramic Capacitor	100 nF	
C2	Ceramic Capacitor	2.2 μ F 16V	
C3, C4	Ceramic Capacitor	470 nF 50V	
L1, L2	Choke Coils	B82462A2104K000 100 μ H B82422H1473K000 47 μ H ELL6SH101M 100 μ H	Epcos Epcos Panasonic
R1, R2	Resistor	56R	
U1	FPGA	XC3S250E (or larger)	Xilinx
U2	RS485 Transceiver	SN65LBC176 MAX13431E	Texas Instruments Maxim IC
TR1	Pulse Transformer	B78416A8607A003 B78304B1030A003 78602/1C	Epcos Epcos Murata

Table 5. Integrated cable 2-wire interface circuit components.

3.2. FPGA IP-Core

The drive system communicates to the DSL motor-feedback system through a dedicated protocol logic design, the so-called “DSL Master”. The design is implemented in a Field Programmable Gate Array component (FPGA) and is supplied by SICK STEGMANN. The design is delivered as an Intellectual Property core (IP core) together with a test design (test bench). The DSL Master IP core comes in the form of a net list that can be routed freely within the FPGA. If the used FPGA has enough space it is possible to implement the DSL Master on the same component as the drive application.

Figure 8 shows the block diagram of the DSL Master design. Basic characteristics are listed in Table 6.



1) Optional interface

Figure 8. Block diagram of the DSL Master IP core.

HIPERFACE DSL®

Parameter	Value			Unit	Notes
	min.	Typ.	max.		
Peripheral characteristics					
Clock frequency	73.721	73.728	73.735	MHz	Tolerance: +/- 100 ppm
Supported FPGA family	Xilinx Spartan 3E				
Required „Speed Grade“	tbd				
Required FPGA resources: LUTs			tbd		
Required FPGA resources: FFs			tbd		
FPGA power consumption		tbd		mW	At 25 °C
Line transmission rate		9.216		MBd	
Reset duration	0.02	0.06		µs	Reset low-active
Start-up time		tbd		µs	
Recovery time after communication errors	281		562	µs	
Motor-feedback characteristics					
Position resolution per turn		24	40	Bit	Sum can not exceed 40 bits
Number of resolved turns		16	40	Bit	
Speed			258 000	rad/s	24 bits/turn
Acceleration			648 000	rad/s ²	24 bits/turn
Sampling latency		9		µs	Trigger edge to valid position
Host interface characteristics					
Drive cycle time	12.15		2 000	µs	In SYNC mode
Packet cycle time	12.15		24.30	µs	In SYNC mode
Packet cycle time		11.72		µs	In free-running mode
SYNC signal duration	0.02			µs	SYNC signal must be inactive for at least 0.02 µs per cycle
SYNC signal frequency jitter			2	%	
Resynchronization duration	240		480	µs	For changes of SYNC signal frequency outside of specified jitter
Clock SPI 1			10	MHz	
Clock SPI 2			10	MHz	
Clock SPI-PIPE			10	MHz	
Parameter channel characteristics					
Theoretical transmission rate	164		329	kBd	
Duration of access to communication resource	170		1120	µs	„Short Message“
Duration of access to encoder resource	267		777	µs	„Long Message“
Number of encoder resources		1 024			
Address space per encoder resource		32 768			
Pipeline channel characteristics					
Transmission rate	329		658	kBd	

Table 6. DSL Master IP core properties.

3.2.1. Pin-out DSL Master

The following Table 7 describes the pin functions of the DSL Master IP core.

Pin Name	Type	Function
/RST	Input	Master reset (low-active)
CLK	Input	Clock input
SYNC	Input	Trigger position sampling
INT	Output	Configurable interrupt
LINK	Output	Link indicator
SPI1 SS	Input	SPI 1 slave select
SPI1 CLK	Input	SPI 1 serial clock
SPI1 MOSI	Input	SPI 1 master out / slave in data
SPI1 MISO	Output	SPI 1 master in / slave out data
SPI2 SS	Input	SPI 2 slave select
SPI2 CLK	Input	SPI 2 serial clock
SPI2 MOSI	Input	SPI 2 master out / slave in data
SPI2 MISO	Output	SPI 2 master in / slave out data
SPIPIPE SS	Input	Pipeline SPI slave select
SPIPIPE CLK	Input	Pipeline SPI serial clock
SPIPIPE MISO	Output	Pipeline SPI master out / slave in data
DSL IN	Input	DSL line data in
DSL OUT	Output	DSL line data out
DSL EN	Output	DSL line transceiver enable

Table 7. Pin functions of IP core interface.

3.2.2. IP Core Implementation – Xilinx ISE

The DSL Master IP core is distributed by SICK STEGMANN as a fixed netlist in NGC format.

The NGC netlist must be copied directly into the project directory of the FPGA project.



Please note that an NGC netlist can not be added with the menu commands „Project > Add Source“ or „Project > Add Copy of Source“.

Instantiation of the NGC netlist is made as a „black_box“ within the source code of the design.

In VHDL this instantiation follows the following template:

```
component DSLMASTER
  port (...);
end component;

...
attribute box_type : string;
attribute box_type of DSLMASTER : component is „black_box“;
...
begin
...
<component_name> : DSLMASTER
  port map (...);
...

```

In Verilog this instantiation follows the following template:

```
module DSLMASTER(...);
endmodule

...
DSLMASTER <module_name> (...);
//synthesis attribute box_type of DSLMASTER is „black_box“
...

```

3.3. Cable Specification

The recommended cable for connecting the drive to the Hiperface DSL® motor-feedback system is specified by parameters in Table 8. This specification is valid for all scenarios.

For integrated cables (see chapters 3.3.2, 3.3.3) motor wires are not specified.

Property	Min	Typical	Max	Unit
Length			100	m
Characteristic impedance @ 10 MHz	100	110	120	Ω
DC-loop resistance			0.1	Ω/m
Velocity ratio		0.66		c
Propagation delay		5		ns/m
Delay skew		irrelevant		
Capacitance @ 1 kHz		tbd		pF
Inductance		tbd		nH
Cutoff frequency	25			MHz
Wire size		tbd		
Maximum current per conductor	0.2			A
Temperature operating	-40		125	$^{\circ}\text{C}$

Table 8. Hiperface DSL® cable specification.

3.3.1. Separate Encoder Cable

A suggested cross-section of the separate encoder cable with four encoder wires is shown in Figure 9.

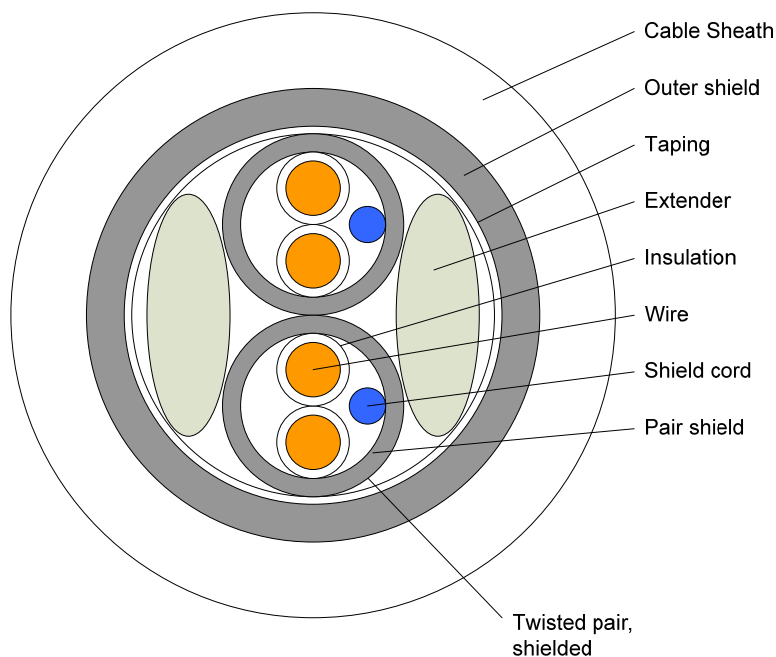


Figure 9. Cross section of separate encoder cable with 4 encoder wires.

3.3.2. Integrated Cable – 4-Wire

A suggested cross-section of the integrated cable with four encoder wires is shown in Figure 10.

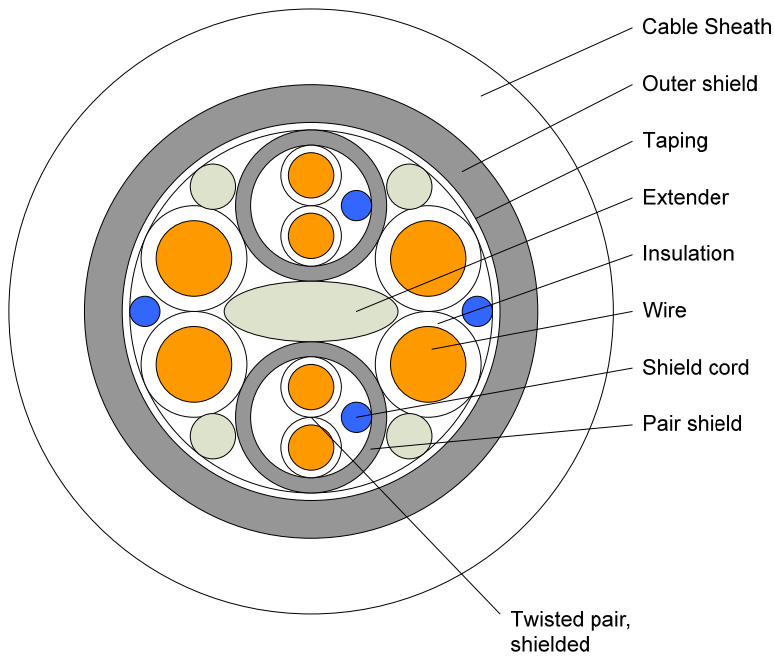


Figure 10. Cross section of integrated cable with 4 encoder wires.

3.3.3. Integrated Cable – 2-Wire

A suggested cross-section of the integrated cable with two encoder wires is shown in Figure 11.

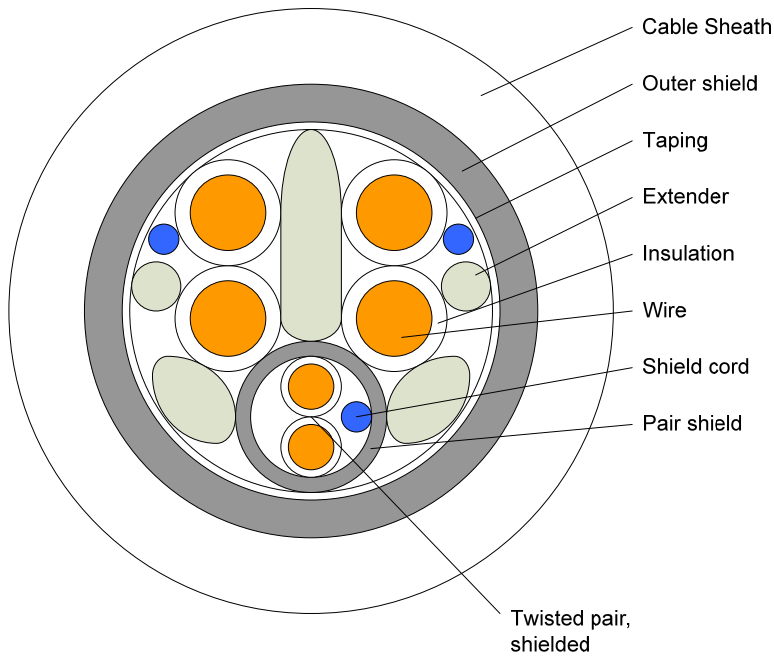


Figure 11. Cross section of integrated cable with 2 encoder wires.

4.1. SPI1 Interface

SPI1 is a full duplex serial interface according to the SPI standard (Serial Peripheral Interface). The SPI1 interface is the central communication interface between drive application and DSL Master IP core. Over this interface primary channel position data can be read out and functions of the motor-feedback system can be accessed.

The SPI1 interface is the only interface to the register block 1 (see chapter 5.3).

SPI master of SPI1 is the drive application. The SPI functions Slave Select (PIN: SPI1 SS), Clock (PIN: SPI1 CLK) and Data Master-Out-Slave-In (PIN: SPI1 MOSI) are controlled by the drive application. SPI function Data Master-In-Slave-Out (PIN: SPI1 MISO) is controlled by the DSL Master.

SPI1 is used with the following SPI parameters:

- PHA = 1 (Sampling during falling clock edge, data changes during rising clock edge)
- POL = 0 (Default level of clock)

Data is transmitted with most significant bit (MSB) first.

SPI1 implements the following register based transactions:

- Read single register
- Read multiple registers with random access
- Write single register
- Write multiple registers (auto-increment)
- Read/write sequence



Please note that in a read/write sequence the write operation must always be the last transaction in the sequence.

At the start of each transaction the DSL Master transmits on SPI1 MISO the Online Master Status in two bytes (see chapter 5.2).

4.1.1. Timing SPI1

For accessing the SPI1 interface the following timing specifications have to be kept.



Each transaction over SPI1 is framed by a '1' level on SPI1 SS. Each time SPI1 SS is reset a new transaction is started, leading to a renewed transmission of the online master status values as the first two bytes (see chapter 5.2).

SPI1 is specified according to the following timing chart (Figure 13) and Table 10.

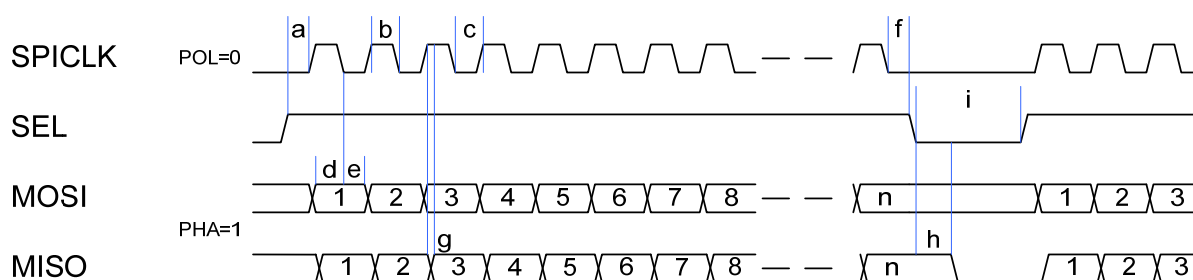


Figure 13. SPI1 interface timing.

The timing of the SPI is specified by the following table:

Reference	Description	Min	Max	Unit
a	Set up SS before CLK	25		Ns
b	CLK high time	50		ns
c	CLK low time	50		ns
d	Set up MOSI before CLK low	10		ns
e	Hold MOSI after CLK low	25		ns
f	Hold SS after CLK low	260		ns
g	Delay MISO after CLK high	40	60	ns
h	Delay MISO after SS low	40	60	ns
i	SS low time	50		ns

Table 10. SPI1 timing.

4.1.2. Dummy Read

Read transactions take a shorter time to transmit on SPI1 MISO than receiving on SPI1 MOSI. Therefore, during reception on SPI1 MOSI, dummy read transactions must be inserted on SPI1 MISO to avoid unwanted additional transactions.

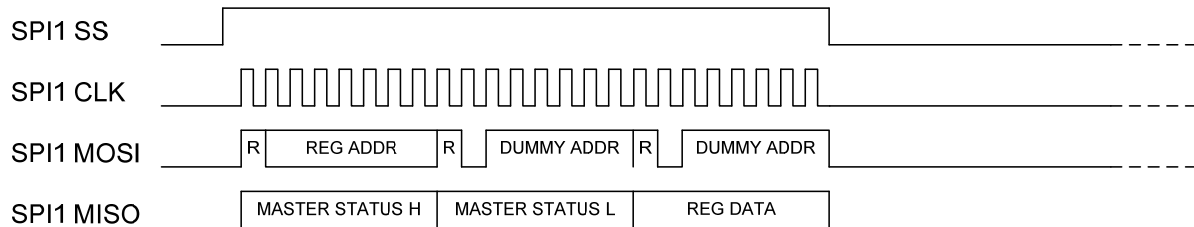
A read access to register 3Fh has no effect and has to be used for this purpose.

4.1.3. Read Single Register

The SPI1 transaction “Read Single Register” allows the reading of a single register of the DSL Master IP-core.

HIPERFACE DSL®

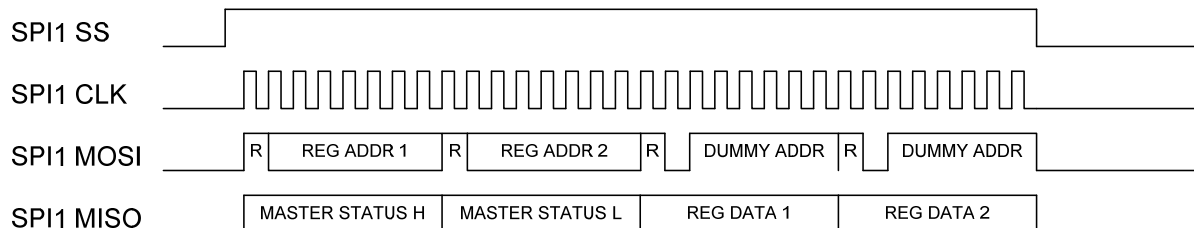
Symbol	Meaning
R	Access bit: Read ('1')
REG ADDR	Register address (00h – 7Fh)
DUMMY ADDR	Register address for "Dummy Read" (3Fh)
MASTER STATUS H	Online DSL Master Status – high byte
MASTER STATUS L	Online DSL Master Status – low byte
REG DATA	Register contents



4.1.4. Read Multiple Registers

The SPI1 transaction "Read Multiple Registers" allows the reading of several registers of the DSL Master IP-core. Registers can be read out in arbitrary order.

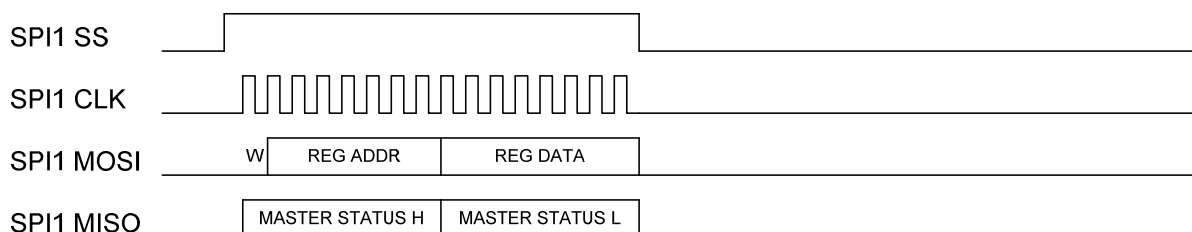
Symbol	Meaning
R	Access bit: Read ('1')
REG ADDR x	Register address (00h – 7Fh) No. x
DUMMY ADDR	Register address for "Dummy Read" (3Fh)
MASTER STATUS H	Online DSL Master Status – high byte
MASTER STATUS L	Online DSL Master Status – low byte
REG DATA x	Contents of register No. x



4.1.5. Write Single Register

The SPI1 transaction "Write Single Register" allows the writing to a single register of the DSL Master IP-core.

Symbol	Meaning
W	Access bit: Write ('0')
REG ADDR	Register address (00h – 7Fh)
REG DATA	Register contents
MASTER STATUS H	Online DSL Master Status – high byte
MASTER STATUS L	Online DSL Master Status – low byte



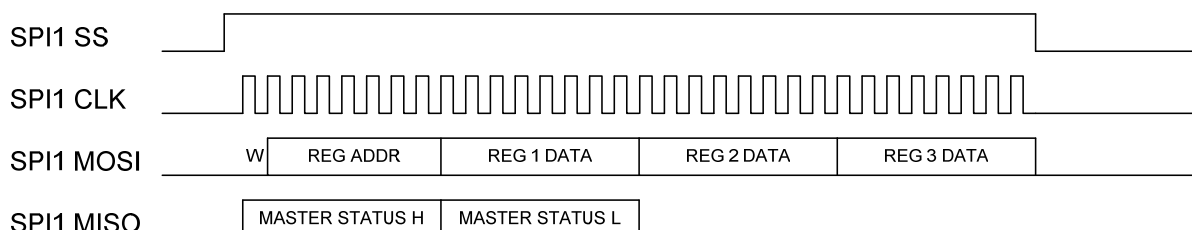
4.1.6. Write Multiple Registers (Auto-increment)

The SPI1 transaction “Write Multiple Registers” allows the writing to several subsequent registers of the DSL Master IP-core. Within the transaction only the starting register address is transmitted, the IP-core auto-increments addresses for each received register data byte.



Please note that in order to write to several registers in arbitrary order, new SPI1-transactions must be started for each new sequence.

Symbol	Meaning
W	Access bit: Write ('0')
REG ADDR	Starting register address (00h – 7Fh)
REG DATA x	Contents for register No. x starting at REG ADDR
MASTER STATUS H	Online DSL Master Status – high byte
MASTER STATUS L	Online DSL Master Status – low byte



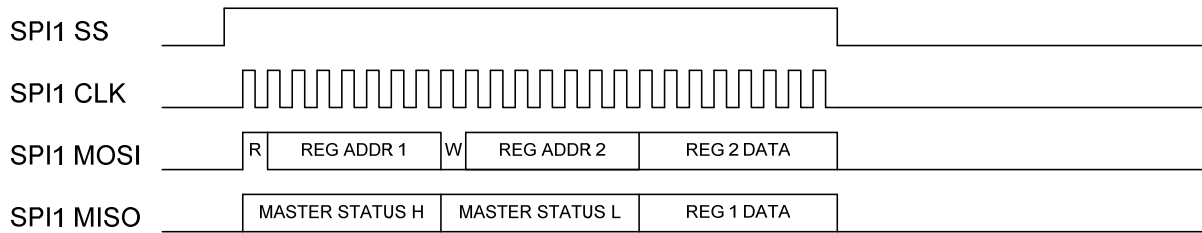
4.1.7. Read/Write Sequence

The SPI1 transaction “Read/Write Sequence” enables a fast sequence for reading from one or more registers and then writing to several subsequent registers of the DSL Master IP-core. SPI1 SS is not reset within the transaction and the online DSL master status is not transmitted twice.



The write operation must always be the last part of the “Read/Write Sequence”. Multiple and single read or write access can be combined.

Symbol	Meaning
R	Access bit: Read ('1')
W	Access bit: Write ('0')
REG ADDR 1	Register address for read access (00h – 7Fh)
REG ADDR 2	Starting register address for write access (00h – 7Fh)
REG DATA 2	Register contents for write access
MASTER STATUS H	Online DSL Master Status – high byte
MASTER STATUS L	Online DSL Master Status – low byte
REG DATA 1	Register contents of read access

HIPERFACE DSL®**4.1.8. SPI1 Errors**

If the SPI1 interface is addressed in a wrong way, error indications will be given by the SPI1 MISO line.

The error indication is shown by a '1' signal on SPI1 MISO after SPI1 SS is reset by the drive application.

Table 11 shows the list of error conditions that lead to the error indication.

SPI1 error condition	Error indication
Wrong number of CLK pulses	SPI1 MISO high
Write command without data	SPI1 MISO high

Table 11. SPI1 errors.

4.2. SPI2 Interface

SPI2 is a full duplex read-only serial interface according to the SPI standard (Serial Peripheral Interface). The SPI2 interface is the communication interface for the secondary channel between drive application and DSL Master IP core. Over this interface position data from a second source can be read out for safety-relevant diagnostics.

The SPI2 interface is the only interface to the register block 2.

SPI master of SPI2 is the drive application. The SPI functions Slave Select (PIN: SPI2 SS), Clock (PIN: SPI2 CLK) and Data Master-Out-Slave-In (PIN: SPI2 MOSI) are controlled by the drive application. SPI function Data Master-In-Slave-Out (PIN: SPI2 MISO) is controlled by the DSL Master.

SPI2 uses identical SPI parameters, timing, register access transactions, and error indications as specified for SPI1 in chapter 4.1.

4.3. SPI-PIPE Interface

SPI-PIPE is a read-only serial peripheral interface (SPI). SPI-PIPE is an optional communication channel between drive application and DSL Master IP-core. This interface allows reading of the pipeline channel. Alternatively it is also possible to read this data register-based with standard transactions over SPI1.

The selection of access to the pipeline channel is made by setting or clearing bit **SPPE** of the register **SYS_CTRL** (see chapter 5.3.1). If **SPPE** is cleared, data and status of the pipeline channel is accessed through DSL Master registers **PIPE_S** (2Dh) and **PIPE_D** (2Eh). If **SPPE** is set, reading of the pipeline channel is conducted through the SPI-PIPE transaction “Read Pipeline”.

SPI-PIPE should be enabled if the bandwidth of SPI1 is not enough to accommodate accessing both position data and pipeline data in a fast drive cycle application or if the pipeline data is processed by a different resource of the drive application.



Please note that the parameterization of external sensor components on the SensorHub is carried out via the Parameter channel of the DSL Master in any case. SPI-PIPE only grants read-access to the Pipeline channel (see chapter 2.5).

Data of the Pipeline channel is stored in a first-in-first-out buffer (FIFO) that can hold 8 bytes. Additionally for each data byte in the FIFO status information is stored (see chapters 5.3.21, 5.3.22).



Please note that the FIFO buffer can only store 8 bytes of Pipeline channel data. If the buffer is not read out fast enough old data will be overwritten. This is indicated by a flag in the status information of the FIFO buffer.

The SPI master of SPI-PIPE is the drive application. The SPI functions “slave select” (pin: SPIPIPE SS) and “clock” (pin: SPIPIPE CLK) are controlled by the drive application. The SPI function “master-in-slave-out data” (pin: SPIPIPE MISO) is controlled by the DSL Master.

SPI-PIPE has the following SPI characteristics:

- PHA = 1 (falling clock edge sampling, data changes during rising clock edge)
- POL = 0 (base clock value)

Data is clocked out with most significant bit (MSB) first.

If SPI-PIPE is used for access to the Pipeline channel, the first 4 bits of the status buffer show a toggling (0101) value in each transaction in order to check the integrity of the interface.

4.3.1. Timing SPI-PIPE

SPI-PIPE timing is specified according to the timing chart (Figure 14) and Table 12.

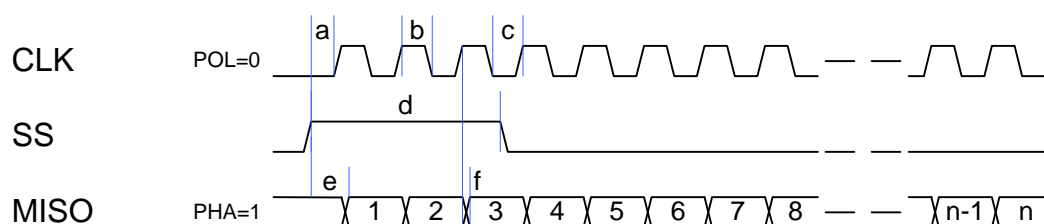


Figure 14. SPI-PIPE interface timing.

Reference	Description	min	max	unit
a	Set up SS before CLK	30		ns
b	CLK high time	30		ns
c	CLK low time	30		ns
d	SS pulse width	30		ns
e	Delay MISO after SS high	25	70	ns
f	Delay MISO after CLK high	25	70	ns

Table 12. SPI-PIPE timing.

4.3.2. Pipeline Read

The SPI-PIPE transaction “Read Pipeline” allows read-access to values of the FIFO buffer containing data and status of the Pipeline channel.

Symbol	Meaning
PIPE STATUS	Pipeline channel status (see chapter 5.3.20)
PIPE DATA	Pipeline channel data (see chapter 5.3.21)

Table 13. Pipeline read transaction.

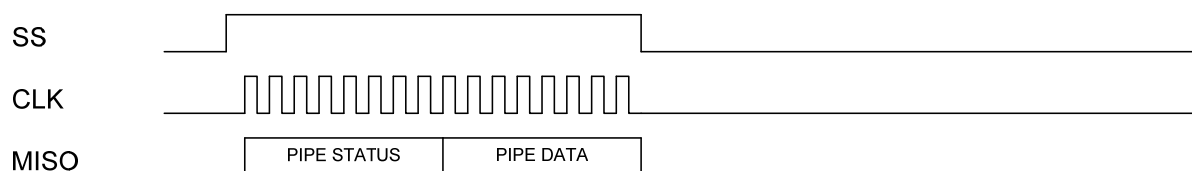


Figure 15. Pipeline read transaction.

4.4. SYNC Interface

SYNC is a single digital input to the DSL Master.

An edge on this pin triggers a position sampling. The polarity of the edge can be programmed via the **SPOL** bit of the register **SYS_CTRL** (00h). The protocol relies on a constant frequency of the signal on this pin within the specified jitter tolerance. During start-up the protocol synchronizes the protocol framing to the SYNC signal frequency.



If the frequency of the SYNC signal leaves the jitter window a protocol re-synchronization will be initiated. During the time of re-synchronization sampling is performed with a signal generated by the DSL Master IP-core until re-synchronization has finished.

Incremental position data is guaranteed to be not older than 30 μ s and absolute position data not older than 240 μ s in this case.

4.5. INT Interface

INT is a single digital output from the DSL Master.

INT is raised if an interrupt condition has been met within the DSL Master. The interrupt conditions are set by registers **MASK_H**, **MASK_L**, **MASK_SUM** (see chapters 5.3.5, 5.3.15).



Each time one of **EVENT_H** or **EVENT_L** registers is written to the INT output is cleared until the current SPI transaction ends.

4.6. LINK Interface

LINK is a single digital output from the DSL Master.

LINK represents the bit LINK of the register **MASTER_QM** (see chapter 5.3.3) and therefore shows if the DSL Master has acquired a communication link to a connected Hiperface DSL® motor-feedback system.

LINK can be used for start-up behavior control (see chapter 6.1) or global error handling.

4.7. Reset

/RST is the reset input of the DSL Master IP core.

After start-up (power-on) of the drive a reset sequence is mandatory to force the DSL Master IP core into its initialization state.

The reset sequence is determined by parameters detailed in Table 14 and explained in Figure 16.

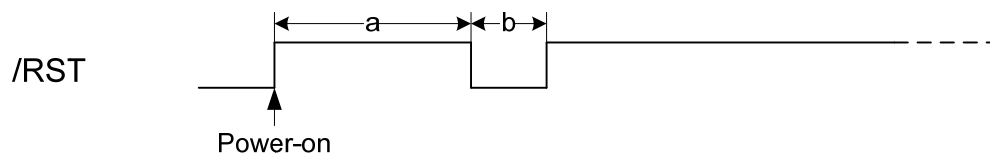


Figure 16. Reset sequence.

Reference	Parameter in Table 6	Value (cf. Table 6)
a	Reset delay	Arbitrary
b	Reset signal duration	>60 ns

Table 14. Reset timing.

5. Register map

The DSL Master is accessed via registers in two different register blocks. Each register block has its own address space as detailed in Table 15.

Register Block	Address Space	Functions
1	00h – 7Fh	Channel 1 position/status Parameter channel Pipeline channel
2	00h – 3Fh	Channel 2 position/status

Table 15. Register block address space.

Transactions with the DSL Master registers are carried out via either SPI1 or SPI2 by corresponding read and write operations. Optionally, SPI-PIPE allows direct access to the Pipeline Channel data.

In Register Block 1 additionally DSL Slave interface registers are mirrored as remote registers in the address range 40h to 7Fh. Addressing of these registers is carried out like with DSL Master registers. The response to the transaction is delayed though and has to be read out individually (see „Short message“, chapter 6.7.1).

The following Figure 17 shows which interface connects to which register block.

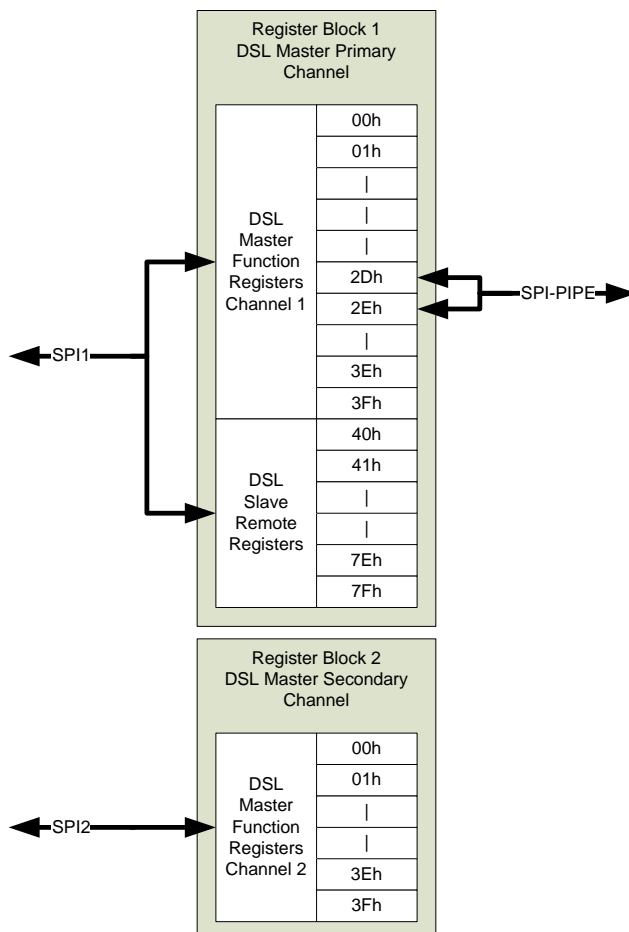


Figure 17. Register block interfaces.

Function	Register Address	Function	Register Address	Function	Register Address	Function	Register Address
SYS_CTRL	00h	PC_BUFFER0	20h	ENC_ST0	40h		60h
SYNC_CTRL	01h	PC_BUFFER1	21h	ENC_ST1	41h		61h
	02h	PC_BUFFER2	22h	ENC_ST2	42h		62h
MASTER_QM	03h	PC_BUFFER3	23h	ENC_ST3	43h		63h
EVENT_H	04h	PC_BUFFER4	24h	ENC_ST4	44h		64h
EVENT_L	05h	PC_BUFFER5	25h	ENC_ST5	45h		65h
MASK_H	06h	PC_BUFFER6	26h	ENC_ST6	46h		66h
MASK_L	07h	PC_BUFFER7	27h	ENC_ST7	47h		67h
MASK_SUM	08h	PC_ADD_H	28h		48h		68h
EDGES	09h	PC_ADD_L	29h		49h		69h
DELAY	0Ah	PC_OFF_H	2Ah		4Ah		6Ah
VERSION	0Bh	PC_OFF_L	2Bh		4Bh		6Bh
	0Ch	PC_CTRL	2Ch		4Ch		6Ch
ENC_ID2	0Dh	PIPE_S	2Dh		4Dh		6Dh
ENC_ID1	0Eh	PIPE_D	2Eh		4Eh		6Eh
ENC_ID0	0Fh	PC_DATA	2Fh		4Fh		6Fh
POS4	10h		30h		50h		70h
POS3	11h		31h		51h		71h
POS2	12h		32h		52h		72h
POS1	13h		33h		53h		73h
POS0	14h		34h		54h		74h
VEL2	15h		35h		55h		75h
VEL1	16h		36h		56h		76h
VELO	17h		37h		57h		77h
SUMMARY	18h		38h		58h		78h
VPOS4	19h		39h		59h		79h
VPOS3	1Ah		3Ah		5Ah		7Ah
VPOS2	1Bh		3Bh		5Bh		7Bh
VPOS1	1Ch		3Ch		5Ch	SRSSI	7Ch
VPOS0	1Dh		3Dh		5Dh		7Dh
POSCRC1	1Eh		3Eh		5Eh	MAIL	7Eh
POSCRC0	1Fh	DUMMY	3Fh		5Fh	PING	7Fh

Table 16. DSL Master Register Block 1 Map.

Function	Register Address	Function	Register Address	Function	Register Address	Function	Register Address
	00h		10h		20h		30h
	01h		11h		21h		31h
	02h		12h		22h		32h
	03h		13h		23h		33h
	04h		14h		24h		34h
	05h		15h		25h		35h
	06h		16h		26h		36h
	07h		17h		27h		37h
	08h	STATUS2	18h		28h		38h
	09h	VPOS24	19h		29h		39h
	0Ah	VPOS23	1Ah		2Ah		3Ah
	0Bh	VPOS22	1Bh		2Bh		3Bh
	0Ch	VPOS21	1Ch		2Ch		3Ch
	0Dh	VPOS20	1Dh		2Dh		3Dh
	0Eh	POSCRC21	1Eh		2Eh		3Eh
ENC2_ID0	0Fh	POSCRC20	1Fh		2Fh	DUMMY2	3Fh

Table 17. DSL Master Register Block 2 Map.

5.1. Register Legend

The following register description uses symbols to describe the default bit value after a reset. Other symbols are used to define the functions that are granted to the drive applications for this bit.

The bit is described with the pattern

“Function”-“Reset value”, e. g. “R/W-0”

Function symbol	Meaning
R	Bit can be read.
W	Bit can be set and cleared.
C	Bit can only be cleared.
X	Bit is not implemented and will always be read as '0'.

Table 18. Bit function symbols.

Reset value	Meaning
0	Bit is cleared after reset.
1	Bit is set after reset.
x	Bit has no defined value after reset.
-	In register map: Bit is not implemented and will always be read as '0'.

Table 19. Bit reset value symbols.



Please note that a read access to any write-only bit (“W”) will always return a ‘0’ value.

5.2. Online DSL-Master Status

A set of event indicators is transferred through either SPI1 and SPI2 interfaces each time a transaction is performed over the respective interface.

5.2.1. SPI1 Online DSL-Master Status

The status information transmitted at the start of each SPI1 transaction is a non-static copy of the registers **EVENT_H** and **EVENT_L**. The static information in these registers has to be cleared by the user after reading it by writing the value '0' to the corresponding bit of the registers while the Online Status only shows the current state without storing past indications.

The SPI1 Online Status is represented in two bytes.



WARNING

Any error indication in the High Byte is potentially critical and/or safety-relevant. Appropriate measures must be implemented by the user application.

Any error indication in the Low Byte is an indication of an error or a warning that is not safety-relevant.

SPI1 Online Status, High Byte

R-0	R-0	R-0	R-1	R-0	R-0	R-0	R-0
INT	SUM	VRT	FIX1	POS	FIX0	QMLW	PRST
Bit 7							Bit 0

- Bit 7 **INT:** Interrupt line status
 This bit is an exception regarding online status as it is no event indicator. INT reflects the value of the physical line INT so that a polling management can be set up. The criticality of this flag depends on the monitored interrupt sources.
 1 = INT line is high.
 0 = INT line is low.
- Bit 6 **SUM:** Summary byte
 1 = The last valid value of **SUMMARY** was different from zero. The criticality of this flag depends on the individual error source that leads to a set **SUMMARY** (see chapter 5.3.15).
 0 = The last valid value of **SUMMARY** was zero.
- Bit 5 **VRT:** Vertical CRC error
 1 = The last received vertical CRC in channel 1 was wrong. It has to be assumed that the transmitted absolute position of channel 1 (see chapter 5.3.13) is invalid. **Appropriate measures must be taken by the user application.**
 0 = The last received vertical CRC in channel 1 was right.
- Bit 4 **FIX1:** This bit is used to check the SPI1 MISO line against stuck-at-0 errors, forcing a high logic state each SPI1 access.
- Bit 3 **POS:** Position not valid
 1 = A cause of incremental position error in channel 1 has been detected or a realignment is in progress. It must be assumed that the last incremental position value is invalid. This error is not

safety-relevant within the boundaries defined for the safety functions of the DSL system. Appropriate measures for handling an incremental position error should be considered in the user application anyway.

0 = No errors of incremental position in channel 1 detected.

- Bit 2 **FIX0:** This bit is used to check the SPI1 MISO line against stuck-at-1 errors, forcing a low logic state each SPI1 access.
- Bit 1 **QMLW:** Quality monitor low (see chapter 5.3.3)
1 = Current quality monitor is less than 14.
0 = Current quality monitor is greater than or equal to 14.
- Bit 0 **PRST:** Protocol reset
1 = Protocol control unit is performing a stack rebuild.
0 = Protocol control unit is working.

SPI1 Online Status, Low Byte

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	
		MIN	ANS	RETL	RETS	FREL	FRES	
Bit 7								Bit 0

- Bit 7-6 **Unimplemented.** Read as '0'.
- Bit 5 **MIN:** Message initialization acknowledge
1 = The DSL encoder is sending a parameter channel initialization acknowledgement character.
0 = Parameter channel is working.
- Bit 4 **ANS:** Bad answer detected
1 = The last answer to a long message was corrupted.
0 = No errors detected on the last answer to a long message.
- Bit 3 **RETL:** Long message retry
1 = A long message retry is in progress.
0 = No long message retry in progress.
- Bit 2 **RETS:** „Short message“ retry
1 = A „Short message“ retry is in progress.
0 = No „Short message“ retry in progress.
- Bit 1 **FREL:** Long message channel status
1 = The long message channel is free.
0 = The long message channel is busy.
- Bit 0 **FRES:** „Short message“ channel status
1 = The „Short message“ channel is free.
0 = The „Short message“ channel is busy.

5.2.2. SPI2 Online DSL-Master Status

The status information transmitted at the start of each SPI2 transaction is a copy of the register **STATUS2**.



Please note that the channel 2 event register is non-static as opposed to those of channel 1 where the user application has to clear all indications upon occurrence.

The SPI2 Online Status is represented in two bytes.



WARNING

Any error indication in the SPI2 Online Status is potentially critical and/or safety-relevant. Appropriate measures must be implemented by the user application.

SPI2 Online Status, High Byte

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TOG2	TEST2	ERR2	FIX2				
Bit 7							Bit 0

Bit 7 **TOG2**: Channel 2 toggling bit.
TOG2 must always toggle between '0' and '1' between subsequent position transmissions in channel 2. The initial value of **TOG2** is '0'. It has to be assumed that a transmission error occurred if the toggling bit didn't change and that the transmitted absolute position of channel 2 is invalid. **Appropriate measures must be taken by the user application.**

Bit 6 **TEST2**: Channel 2 test in progress.
TEST2 is set if a test is in progress during the currently available status and position values of channel 2.
TEST2 can only be valid if the user application requested a test previously (see chapter 5.5.3).

Corresponding error indications to **TEST2** are either the **ERR2** bit or a mismatch between position and CRC of channel 2.

The following Table 20 shows the possible scenarios and appropriate responses by the user application.

5.3. DSL Master Function Registers Channel 1

The registers of the DSL Master IP-core in channel 1 are used by the protocol logic to control the non-safety behavior of the DSL Master and to access position values of channel 1. Table 21 gives a list of these registers for channel 1.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Reset	
00h	SYS_CTRL	PRST	MRST	FRST	SPI1TEST		SPPE	SPOL	OEN	0000 0000	
01h	SYNC_CTRL	ES								0000 0001	
03h	MASTER_QM	LINK	-	-	-	Quality monitor				0--- 0000	
04h	EVENT_H	-	SUM	VRT	-	POS	-	QMLW	PRST	-00- 0-00	
05h	EVENT_L	-	-	MIN	ANS	RETL	RETS	FREL	FRES	--00 0000	
06h	MASK_H	-	MSUM	MVRT	-	MPOS	-	MQMLW	MPRST	-00- 0-00	
07h	MASK_L	-	-	MMIN	MANS	MRETL	MRETS	MFREL	MFRES	--00 0000	
08h	MASK_SUM	MSUM7:0								0000 0000	
09h	EDGES	Bit sampling pattern								0000 0000	
0Ah	DELAY	RSSI				Line delay				0000 0000	
0Bh	VERSION	Encoding			Secondary Release No					0100 1101	
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	Incremental position byte 4								0000 0000	
11h	POS3	Incremental position byte 3								0000 0000	
12h	POS2	Incremental position byte 2								0000 0000	
13h	POS1	Incremental position byte 1								0000 0000	
14h	POS0	Incremental position byte 0								0000 0000	
15h	VEL2	Velocity byte 2								0000 0000	
16h	VEL1	Velocity byte 1								0000 0000	
17h	VEL0	Velocity byte 0								0000 0000	
18h	SUMMARY	SUM7:0								0000 0000	
19h	VPOS4	Absolute position byte 4								0000 0000	
1Ah	VPOS3	Absolute position byte 3								0000 0000	
1Bh	VPOS2	Absolute position byte 2								0000 0000	
1Ch	VPOS1	Absolute position byte 1								0000 0000	
1Dh	VPOS0	Absolute position byte 0								0000 0000	
1Eh	VPOSCRC_H	Absolute position CRC byte 1								0000 0000	
1Fh	VPOSCRC_L	Absolute position CRC byte 0								0000 0000	
20h	PC_BUFFER0	Parameter channel buffer byte 0								0000 0000	
21h	PC_BUFFER1	Parameter channel buffer byte 1								0000 0000	
22h	PC_BUFFER2	Parameter channel buffer byte 2								0000 0000	
23h	PC_BUFFER3	Parameter channel buffer byte 3								0000 0000	
24h	PC_BUFFER4	Parameter channel buffer byte 4								0000 0000	
25h	PC_BUFFER5	Parameter channel buffer byte 5								0000 0000	
26h	PC_BUFFER6	Parameter channel buffer byte 6								0000 0000	
27h	PC_BUFFER7	Parameter channel buffer byte 7								0000 0000	
28h	PC_ADD_H	LID	LDIR	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	Pipeline FIFO output								0000 0000	
2Fh	PC_DATA	Parameter channel data "Short message"								0000 0000	
3Fh	DUMMY	No data								---- ----	

Table 21. DSL Master Register Description channel 1.

5.3.1. System Control SYS_CTRL

The System Control register **SYS_CTRL** contains the main controlling bits of the DSL Master.



Note that apart from a master reset, all System Control bits are set and cleared by the user only.

Register 00h:

System Control

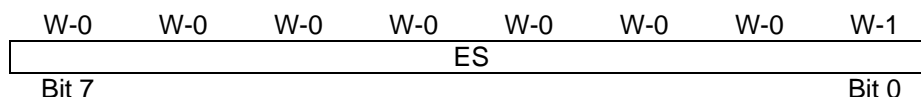
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
PRST	MRST	FRST	SPI1TEST		SPPE	SPOL	OEN	
Bit 7								Bit 0

- Bit 7 **PRST**: Protocol reset
 1 = The protocol is forced to a reset state; a link start-up is re-initiated when this bit is cleared.
 0 = Regular protocol execution.
- Bit 6 **MRST**: Message reset
 1 = The parameter channel is reset, current short and long messages are discarded.
 0 = Regular execution of parameter channel.
- Bit 5 **FRST**: Pipeline FIFO reset.
 1 = FIFO is reset, no data is stored or readable.
 0 = Regular FIFO access.
- Bit 4-3 **SPI1TEST**: Test for SPI1 function
 2 bit value for read-back test of SPI1. This value has no further use.
- Bit 2 **SPPE**: SPI-PIPE enable
 1 = SPI-PIPE enabled. Pipeline status and data accessed via SPI-PIPE, registers **PIPE_S** and **PIPE_D** are read as '0'.
 0 = SPI-PIPE disabled. Pipeline status and data accessed through registers **PIPE_S** and **PIPE_D**.
- Bit 1 **SPOL**: Sync pulse polarity
 1 = The falling edge of SYNC is used
 0 = The rising edge of SYNC is used.
- Bit 0 **OEN**: Output enable
 1 = The DSL line is enabled for output to the DSL Slave.
 0 = The DSL line is in high impedance.

5.3.2. Synchronization Control SYNC_CTRL

The Synchronization Control register **SYNC_CTRL** contains the bit controlling the synchronization source for position sampling.

Register 01h: **Synchronization Control**



Bit 7-0

ES: External synchronization

00000000 = Position sampling is free-running at minimum cycle time.

Any other value = Position sampling is synchronized to SYNC signal. Value of **ES** determines the number of position samplings carried out during one SYNC cycle. The user must match the number of samplings per cycle to the minimum frame length (see chapter 6.3.2).

5.3.3. Quality Monitor MASTER_QM

The Quality Monitor register **MASTER_QM** contains the value of the data-link quality monitor.

If the DSL Master registers events that indicate an improvement or deterioration of the data-link quality these are indicated as a raised or lowered value of the quality monitor (see Table 22).

Protocol event	Quality monitor variation
Indication of last byte of vertical channel during first 7 bytes.	-4
Indication of last byte of vertical channel during last byte.	+1
Indication of first 7 bytes of vertical channel during last byte.	-8
RSSI < 1	-8
Wrong CRC value in vertical channel	-8
Correct CRC value in vertical channel	+1
Bit encoding error in DSL frame 1 st byte	-1
Bit encoding error in DSL frame 2 nd byte	-1
Bit encoding error in DSL frame 3 rd byte	-1
Bit encoding error in DSL frame 4 th and/or 5 th byte	-1

Table 22. Quality monitor events.

The quality monitor is initialized with the value “8”.

The maximum value of the quality monitor is “15”. This value constitutes the default value during operation.



Please note the value of the quality monitor especially during engineering of a DSL motor-feedback system. If a lower value than “15” is indicated – especially if continuously – a problem with the connection link design could be the cause.

If the value of the quality monitor falls below “14” the indication **QMLW** is shown in the online DSL Master status and in the register **EVENT_H**.

HIPERFACE DSL®

If the value of the quality monitor drops to “0” a protocol reset is enforced. This is indicated by the error bit **PRST** in the online DSL Master status and in the register **EVENT_H**.

The register **MASTER_QM** is read-only.

Register 03h: **Quality Monitor Register**

R-0	X-0	X-0	X-0	R-0	R-0	R-0	R-0
LINK				QM3	QM2	QM1	QM0
Bit 7				Bit 0			

- Bit 0 **LINK**: Link status of DSL protocol
 1 = Protocol link between DSL Master and Slave has been engaged.
 0 = No link present.
 Please note that **LINK** is also represented in the LINK interface output (see chapter 4.6).
- Bit 6-4 **Unimplemented**: Read as ‘0’.
- Bit 3-0 **QM3:QM0**: Quality monitor bits.
 0000 – 1111 quality monitor value. Higher values indicate a better link. If the quality monitor reaches 0000, a protocol reset is enforced.

5.3.4. Events

The Event registers **EVENT_H/EVENT_L** contain the notification bits for all warning and error modes of the DSL system.

All notification bits are set by the DSL Master if a corresponding state was diagnosed.

The following bit description lists the consequences of warning and error states and which error reaction should be implemented in the drive application.

If an event bit is set it will not be reset by the DSL Master. The drive application has to clear asserted bits.

In the **EVENT** registers there are both edge sensitive and level sensitive flags. Edge sensitive bits are set whenever their condition occurs and they will be set again only after the condition has disappeared and then become true again. This is the default behaviour. The level sensitive bits cause a flag to be set as long as the condition is true.



Please note that all bits of the event registers are also transmitted in the online DSL Master status (see chapter 5.2). There the event bits are non-static and contain the actual state of each event.

Register 04h: **Events, High Byte**

X-0	R/C-0	R/C-0	X-0	R/C-0	X-0	R/C-0	R/C-0
	SUM	VRT		POS		QMLW	PRST
Bit 7				Bit 0			

- Bit 7 **Unimplemented**: Read as ‘0’.
- Bit 6 **SUM**: Remote Event Summary

1 = The DSL Slave has signaled an event (see register **SUMMARY**).

0 = The DSL Slave events are all cleared.

If the bit **SUM** is set an error or warning of the DSL Slave was transmitted. The drive application should check the register **SUMMARY** (see chapter 5.3.15) to receive a detailed description. This bit is level sensitive.

Bit 5

VRT: Vertical channel error

1 = Data consistency error in vertical channel.

0 = Vertical channel data correctly transmitted.

This error usually shows a transmission error on the DSL link. If this error happens often the DSL connection design should be checked. If this error happens continuously, a fault of the motor-feedback system is likely.

This error influences the quality monitor and leads to the warning **QMLW** or to a protocol reset.



If this error has occurred the last valid absolute position value is retained (see chapter 5.3.13). Only if a new valid value was transmitted will the absolute position be updated and equals the actual position of the motor-feedback system. The incremental position is not influenced by this.

Bit 4

Unimplemented: Read as '0'.

Bit 3

POS: Incremental position error

1 = Data consistency error in incremental position.

0 = Incremental position data correctly transmitted.

This error usually shows a transmission error on the DSL link. If this error happens often the DSL connection design should be checked. If this error happens continuously, a fault of the motor-feedback system is likely.



If this error has occurred the incremental position is updated by an extrapolation with the last valid speed value. Only if a new valid absolute position was transmitted (see chapter 5.3.13) will the incremental position be updated to the actual position of the motor-feedback system. This can lead to temporary (<200 µs) differences between actual and transmitted position, especially during accelerated movements of the drive.

Bit 2

Unimplemented: Read as '0'.

Bit 1

QMLW: Quality monitor low warning

1 = Quality monitor value (see Register 03h) less than 14.

0 = Quality monitor value 14 or higher.

This warning shows that a transmission error on bit level or in one of the CRC values happened. If this error happens often the DSL connection design should be checked (see also chapter 5.3.3).

Bit 0 **PRST:** Protocol reset warning
 1 = Protocol reset has been enforced.
 0 = Regular protocol execution.
 This error message shows that the protocol link to the DSL Slave is re-initialized. This could be caused by a request of the drive application (bit **PRST** in **SYS_CTRL**), self-generated by the DSL Master or activated through the RST input.
 The DSL Master causes a protocol reset if too many transmission errors indicate a link problem (see chapter 5.3.3). A protocol reset causes a re-synchronization with the DSL Slave that can lead to an improvement in link quality.

Register 05h: **Events, Low Byte**

X-0	X-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
		MIN	ANS	RETL	RETS	FREL	FRES
Bit 7						Bit 0	

Bit 7-6 **Unimplemented:** Read as '0'.

Bit 5 **MIN:** Message Init
 1 = A message initialization acknowledge has been received from the slave.
 0 = No initialization acknowledge received.
 If this warning is shown the Parameter channel is still initializing and no „Short message“ or “Long Message” can be initiated.
 This bit is level sensitive.

Bit 4 **ANS:** “Long Message” Bad Answer Error
 1 = A transmission error has occurred during long message response. The effectiveness of the previous transaction is unknown.
 0 = Last “Long message” responses transmission were ok.
 This error shows that the transmission of a response of the DSL Slave to the last “Long message” failed. The drive application should re-issue the “Long message”.

Bit 3 **RETL:** “Long Message” Retry
 1 = “Long message” response has been queried again due to transmission error.
 0 = “Long message” response transmission is ok.
 This warning indicates that the transmission of a request to the DSL Slave from the last “Long message” failed and is retried automatically. If this happens often the DSL connection design should be checked. If this warning is indicated continuously, a fault of the motor-feedback system is likely.

Bit 2 **RETS:** „Short message“ Retry
 1 = „Short message“ response has been queried again due to transmission error.
 0 = „Short message“ response transmission is ok.

This warning indicates that the transmission of a request to the DSL Slave or the response from the DSL Slave from the last „Short message“ failed and is retried automatically. If this happens often the DSL connection design should be checked. If this warning is indicated continuously, a fault of the motor-feedback system is likely.

Bit 1

FREL: „Long Message“ Free Channel

1 = A „Long message“ can be sent over the parameter channel.

0 = No „Long message“ can be sent.

If this bit is set a „Long message“ can be initiated by the drive application. As long as no response was received from the DSL Slave this bit remains cleared. Since the processing duration of a „Long message“ in the motor-feedback system is not determined the drive application should implement an own time-out condition. If a time-out happens setting bit **MRST** in register **SYS_CTRL** leads to a reset of the Parameter channel.

Bit 0

FRES: „Short message“ Free Channel

1 = A „Short message“ can be sent over the parameter channel.

0 = No „Short message“ can be sent.

If this bit is set a „Short message“ can be initiated by the drive application. As long as no response was received from the DSL Slave this bit remains cleared. Since the processing duration of a „Short message“ in the motor-feedback system is determined, a time-out condition is implemented in the DSL Master. If a time-out happens automatic retries are conducted (see bit **RETS**).

5.3.5. Event Mask MASK

In the Event Mask registers **MASK_H/MASK_L** the events are set that cause the interrupt pin assertion (INT).

Multiple events can be masked to raise the interrupt. Additionally, events from the DSL Slave summary can be combined with these events (see chapter 5.3.15). This is illustrated in Figure 18.

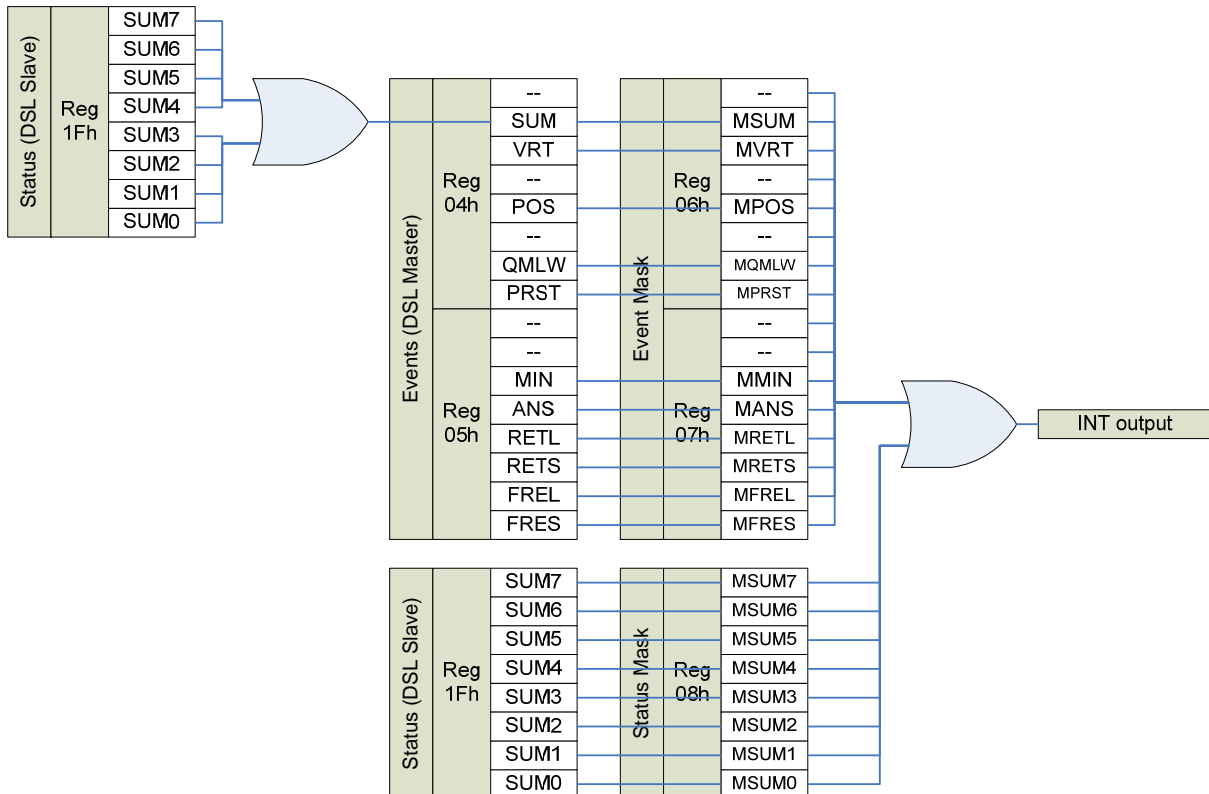


Figure 18. Interrupt masking.



Please note that bit **SUM** is an OR-combination of all bits of the status summary (register **SUMMARY**).

Register 06h:

Event Mask, High Byte

X-0	W-0	W-0	X-0	W-0	W-0	W-0	W-0
MSUM	MVRT		MPOS		MQMLW	MPRST	
Bit 7							Bit 0

Bit 7

Unimplemented: Read as '0'.

Bit 6

MSUM: Remote event summary mask

1 = DSL Slave events masked by register **SUMMARY** will cause assertion of INT pin.

0 = DSL Slave events masked by register **SUMMARY** will not cause INT pin assertion.

- Bit 5 **MVRT**: Vertical channel error mask
 1 = Vertical channel error will cause assertion of INT pin.
 0 = Vertical channel error will not cause INT pin assertion.
- Bit 4 **Unimplemented**: Read as '0'.
- Bit 3 **MPOS**: Incremental position error mask
 1 = Incremental position error will cause assertion of INT pin.
 0 = Incremental position error will not cause INT pin assertion.
- Bit 2 **Unimplemented**: Read as '0'.
- Bit 1 **MQMLW**: Quality monitor low warning mask
 1 = Low quality monitor value (see Registers 03h, 05h) will cause assertion of INT pin.
 0 = Low quality monitor value will not cause INT pin assertion.
- Bit 0 **MPRST**: Protocol reset warning mask
 1 = Protocol reset will cause assertion of INT pin.
 0 = Protocol reset will not cause INT pin assertion.

Register 07h: **Event Mask, Low Byte**

X-0	X-0	W-0	W-0	W-0	W-0	W-0	W-0	
		MMIN	MANS	MRETL	MRETS	MFREL	MFRES	
Bit 7								Bit 0

- Bit 7-6 **Unimplemented**: Read as '0'.
- Bit 5 **MMIN**: Message initialization mask
 1 = DSL Slave message initialization acknowledgement will cause assertion of INT pin.
 0 = DSL Slave message initialization acknowledgement will not cause assertion of INT pin.
- Bit 4 **MANS**: Long message bad answer error mask
 1 = Transmission error during long message response will cause assertion of INT pin.
 0 = Transmission error during long message response will not cause INT pin assertion.
- Bit 3 **MRETL**: Long message retry mask
 1 = Long message re-querying will cause assertion of INT pin.
 0 = Long message re-querying will not cause INT pin assertion.
- Bit 2 **MRETS**: „Short message“ retry mask
 1 = „Short message“ re-querying will cause assertion of INT pin.
 0 = „Short message“ re-querying will not cause INT pin assertion.
- Bit 1 **MFREL**: Long message free channel mask
 1 = If a long message can be sent over the parameter channel INT pin will be asserted.
 0 = If a long message can be sent over the parameter channel INT pin will not be asserted.
- Bit 0 **MFRES**: „Short message“ free channel mask

1 = If a „Short message“ can be sent over the parameter channel INT pin will be asserted.

0 = If a „Short message“ can be sent over the parameter channel INT pin will not be asserted.

5.3.6. Summary Mask MASK_SUM

In the Summary Mask register **MASK_SUM** the DSL Slave summary events are set that cause the interrupt pin assertion (INT).

Multiple events can be masked to raise the interrupt. Additionally, DSL Master events can be combined with these events (see chapter 5.3.4).



Note that bit **MSUM** from register **MASK_H** is an OR combination 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: Status summary bit mask

1 = The corresponding status summary bit will cause an assertion of INT pin if set.

0 = The corresponding status summary bit will not cause an assertion of INT pin if set.

5.3.7. EDGES

The Edges register **EDGES** contains the bit sampling timing of the DSL line and can be used to monitor the link quality.

Each bit of the Edges register is set if an edge of the test signal upon system start-up is detected during the corresponding bit period. An edge is defined as a line value variation between subsequent acquisitions. Sampling is carried out eight times faster than the line bit rate.

Clear line signals will result in only few set bits in the Edges register whereas noisy line signals will show more set bits.



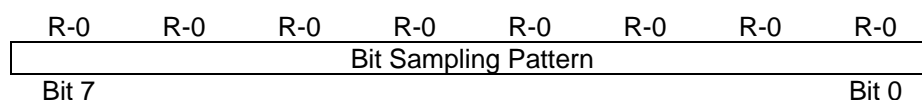
WARNING

If all bits in the Edges register are set this is a sign of a too noisy line where no link can be established.

This register is read-only. The contents of this register do not change after the start-up phase. A new bit sampling pattern is only generated if a protocol reset was enforced.

Register 09h:

Edges



Bit 7-0

Bit Sampling Pattern: Edge detection of line signal.

1 = Edge found in corresponding bit period.

0 = No edge found in corresponding bit period.

5.3.8. DELAY

The Delay register **DELAY** holds information on system line delay and signal strength. It can be used to monitor link quality.

This register is read-only.

Register 0Ah:

Delay

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
RSSI				Line Delay			
Bit 7				Bit 0			

Bit 7-4

RSSI: Received Signal Strength Indication

4 bit line signal strength value ranging from “0” to “12”. Higher values indicate better link quality. If the value sinks below “1”, a protocol reset will be enforced.

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

Bit 3-0

Line Delay:

4 bit line delay value. This value is a representation of line and transceivers round trip delay expressed in bits of the line signal. It can give an estimation of line length.

The value **Line Delay** does not change after the start-up phase. A new **Line Delay** is measured only if a protocol reset was enforced.

The following Table 23 shows the connection between the value **Line Delay** and the cable length of the DSL connection.

Line Delay	Cable length DSL connection
0	< 10 m
1	10 ... 20 m
2	20 ... 30 m
3	30 ... 40 m
4	40 ... 50 m
5	50 ... 60 m
6	60 ... 70 m
7	70 ... 80 m
8	80 ... 90 m
9	90 ... 100 m

Table 23. Line Delay.



WARNING

A value of greater than “9” indicates a delay of more than 1 μ s which will lead to a violation of cycle time specification. You should check that the cable being used adheres to the cable specification.

5.3.9. VERSION

The Version register **VERSION** holds the release revision of the DSL Master IP-core.

This register is read-only.

Register 0Bh: **Version**

R-0	R-1	R-0	R-0	R-1	R-0	R-1	R-1
Encoding				Secondary Release No			
Bit 7				Bit 0			

Bit 7-6 **Encoding:** IP-core type
01 = DSL Master IP-core

Bit 5-0 **Secondary Release No:** Release revision number
The current revision is 0Dh (001101).

5.3.10. Encoder ID ENC_ID

The Encoder ID registers **ENC_ID** hold the identification code of the motor-feedback system connected to the DSL Master. In the current protocol specification the identification code is 20 bit long. For future expansions the free bits in the Encoder ID registers are used to indicate special characters.

These registers are read-only.

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 **Unimplemented:** Read as '0'.

Bit 22-20 **SCI:** Special character indication
3 bit special character for future encoder identification code expansions. Not in use.

Bit 19-0 **ENC_ID:** Encoder identification code
20 bit motor-feedback system identification.

The individual bits of **ENC_ID** have the following usage:

Bit 19 **Continue:** If high, **Continue** indicates that ENC_ID is wider than 20 bit (for future implementation).

Bit 18..10 **Reserved**

HIPERFACE DSL®

- Bit 9..4 **#Pos-#Acc**: Length of position information (default: 40 bits) minus the width of transmitted acceleration value ($\Delta\Delta$ position, see chapter 5.3.11) value (default: 11 bits).
- Bit 3..0 **#Acc-8**: Length of transmitted acceleration value ($\Delta\Delta$ position) (default: 11 bits) minus 8.

5.3.11. Incremental Position POS

The Incremental Position registers **POS** hold the position value of the attached motor-feedback system. This position is incrementally built from a start-up absolute position and updated each protocol frame.

Every eight protocol frames the incremental position is checked against the absolute position (see registers 18h – 1Ch).

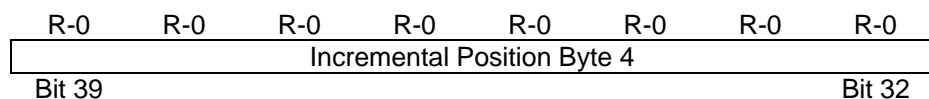
The sampling point of the position is determined by the Synchronization Control register value **ES**.

Only those bits of POS are enabled that lie in the range actually measured by the motor-feedback system. All other, more significant bits read as '0'. The number of measurable bits can be determined from bits 9..0 of **ENC_ID** in registers **ENC_ID0...2**.

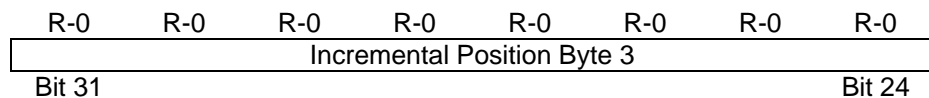
The unit of the position value is [steps].

These registers are read-only.

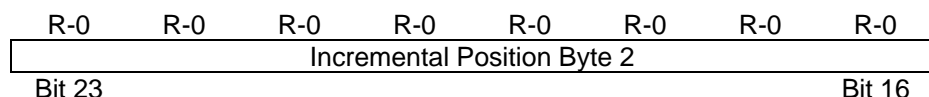
Register 10h: **Incremental Position Byte 4**



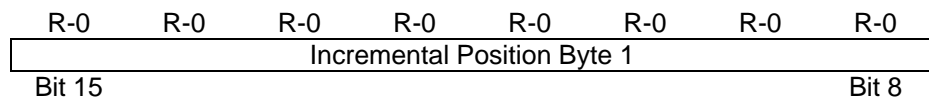
Register 11h: **Incremental Position Byte 3**



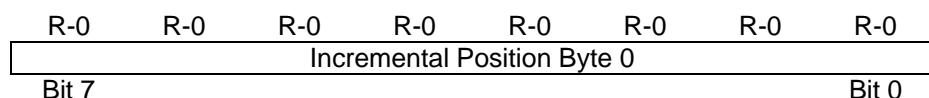
Register 12h: **Incremental Position Byte 2**



Register 13h: **Incremental Position Byte 1**



Register 14h: **Incremental Position Byte 0**



- Bit 39-0 **Incremental Position Byte 4/3/2/1/0**:
40 bit motor-feedback system position value, incrementally built.

5.3.12. Velocity VEL

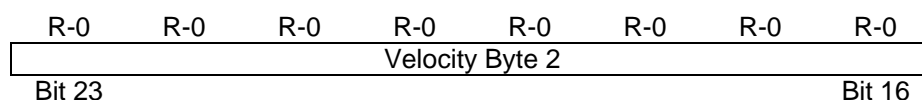
The Velocity registers **VEL** hold the velocity value of the attached motor-feedback system. This value is calculated as Δ position from the acceleration value ($\Delta\Delta$ position) transmitted in the horizontal channel and updated each protocol frame (see chapter 2.1).

The sampling point of the velocity value is determined by the value **ES** of the register **SYNC_CTRL**.

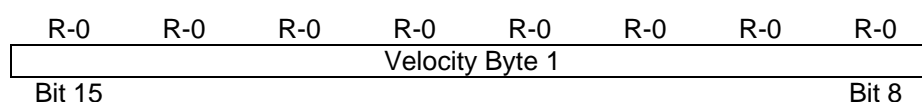
The unit of the velocity value is [steps/frame cycle time].

These registers are read-only.

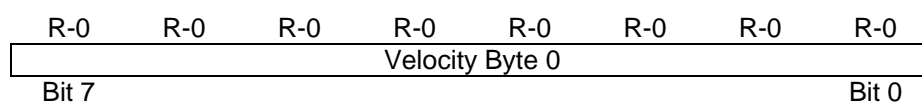
Register 15h: **Velocity Byte 2**



Register 16h: **Velocity Byte 1**



Register 17h: **Velocity Byte 0**



Bit 23-0

Velocity Byte 2/1/0:

24 bit motor-feedback system velocity value

5.3.13. Status Summary SUMMARY

The Status Summary register **SUMMARY** contains the summarized DSL Slave status information. Each bit from the Status Summary contains the aggregated information of 8 error/warning/event modes of the DSL Slave. Bits in the Status Summary register are readable and can be cleared if set. Figure 19 shows the connection between Encoder Status registers, the Status Summary register, and the bit **SUM** in the **EVENT** registers.

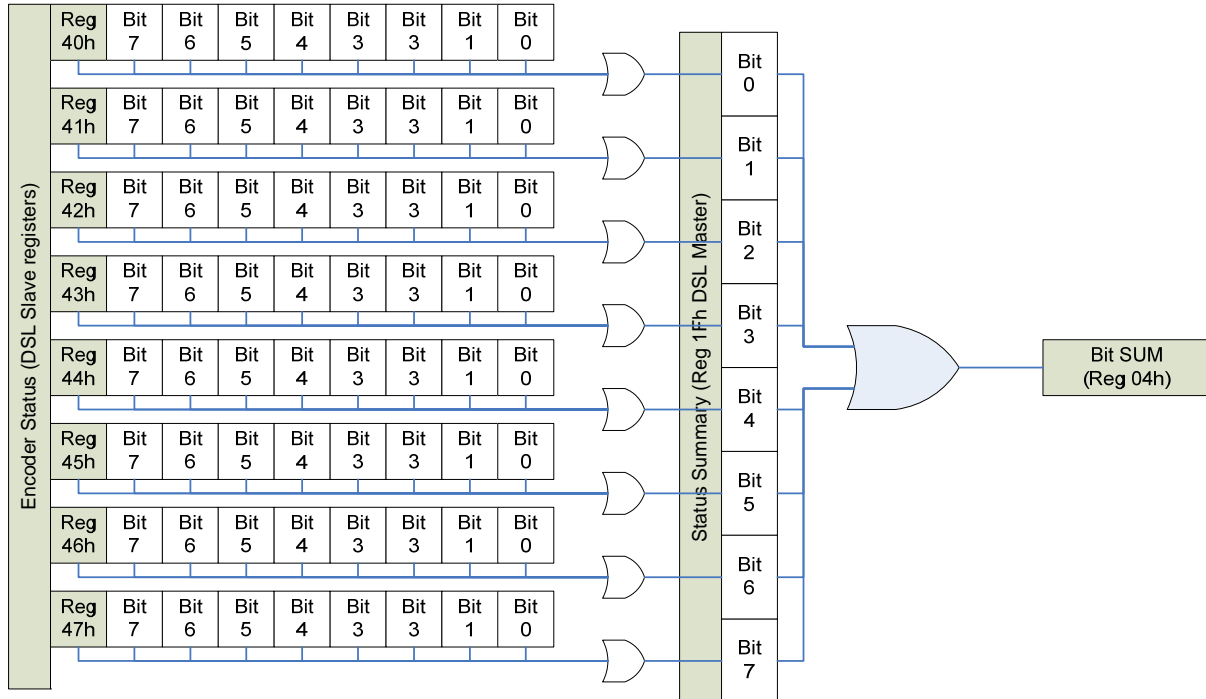


Figure 19. DSL Slave status and summary.

If a bit of **SUMMARY** is set it will not be cleared by the DSL system automatically. For clearing the drive application must read the applicable Encoder Status register of the DSL Slave (see chapter 5.5.1) and acknowledge the status message by clearing any 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/warning/event pertaining to DSL Slave external resources has been raised.

0 = The corresponding error/warning/event is not active.

Bit 0

SUM0: Status summary bit (interface)

1 = An error/warning/event pertaining to the DSL Slave protocol interface has been raised.

0 = The DSL Slave protocol has not raised an error/warning/event.

5.3.14. Absolute Position VPOS

The Absolute Position registers **VPOS** hold the position value of the primary channel of the attached motor-feedback system. This absolute position is transmitted every eight protocol frames if the validity of the data transmission has been verified.

Each time the absolute position is transmitted its value is checked against the incrementally built position value (see registers 10h – 14h).

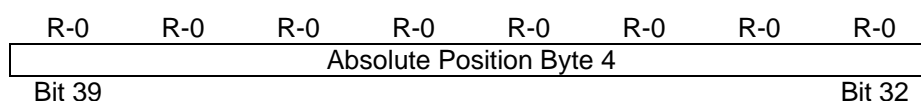
The absolute position is not synchronized to the SYNC signal.

Only those bits of VPOS are enabled that lie in the range actually measured by the motor-feedback system. All other, more significant bits read as '0'. The number of measurable bits can be determined from bits 9..0 of **ENC_ID** in registers **ENC_ID0...2**.

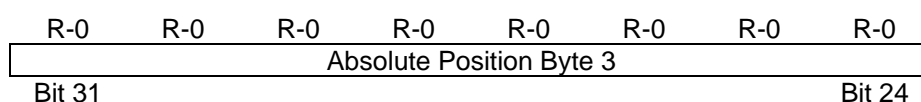
The unit of the position value is [steps].

These registers are read-only.

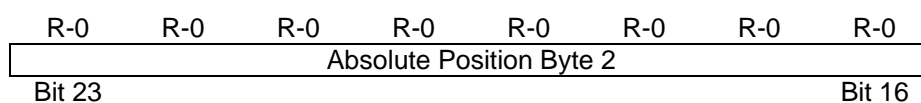
Register 19h: **Absolute Position Byte 4**



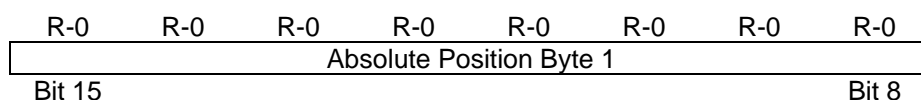
Register 1Ah: **Absolute Position Byte 3**



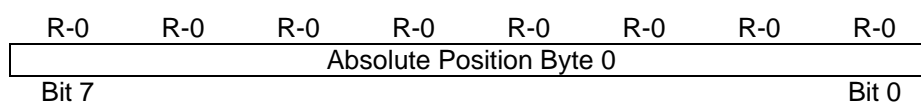
Register 1Bh: **Absolute Position Byte 2**



Register 1Ch: **Absolute Position Byte 1**



Register 1Dh: **Absolute Position Byte 0**



Bit 39-0

Absolute Position Byte 4/3/2/1/0:

40 bit motor-feedback system channel 1 position value, absolute value transmitted.

5.3.15. Position Checksum POSCRC

The Position Checksum registers **POSCRC** hold the cyclic redundancy checksum (CRC) of the absolute position VPOS (see chapter 5.3.13) and the status summary SUMMARY (see chapter 5.3.13).

The CRC is checked within the DSL Master IP-core. To ensure in a safety application that the CRC machine is working within the IP-core these registers allow external cross-checking within the diagnostic test interval defined in the chapter on Safety Implementation (see Hiperface DSL® Safety Implementation Manual).

The CRC is built with the following CRC parameters:

Parameter	Value
CRC order	16 bits
CRC polynomial	C86Ch
Initial value	0000h
Final XOR value	00FFh
Reverse data bytes	No
Reverse CRC before final XOR	No

Table 24. POSCRC parameters.

The order of bytes for calculating the CRC is shown in the following figure:

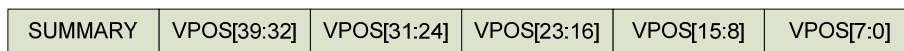
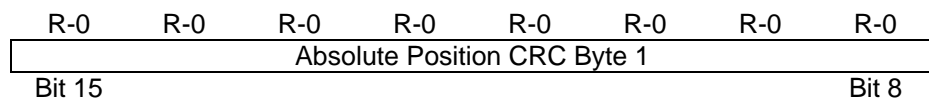
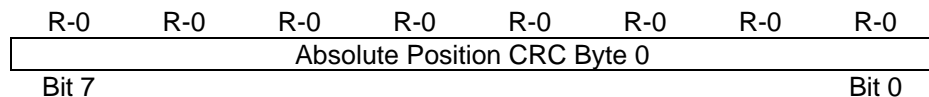


Figure 20. Byte order for CRC calculation.

Register 1Eh: **Absolute Position CRC Byte 1**



Register 1Fh: **Absolute Position CRC Byte 0**



Bit 15-0

Absolute Position CRC:

16 bit cyclic redundancy checksum (CRC16) of absolute position and status summary in Channel 1.

5.3.16. Parameter Channel Buffer PC_BUFFER

The eight Parameter Channel Buffer registers **PC_BUFFER** contain the response of the last “Long message” request or the data for a “Long message” write operation.



These registers are to be accessed only if the “Long message” channel is free (**FREL** in register **EVENT_L**).

According to the length of the “Long message” response the registers are used as follows:

„Long message“ length	Used registers
8 bytes	20h – 27h
4 bytes	20h – 23h
2 bytes	20h – 21h
0 bytes	none

Table 25. Data length “Long message”.

These registers are also used to report error conditions resulting from a “Long message” operation. If an error occurs during accessing a resource due to a “Long message” (e.g. invalid data, ADC conversion errors, time-out conditions), the bit **LOFF** in the **PC_ADD_H** register (28h) will be set after receiving the reply message. In this case, the Parameter Channel Buffer bytes 0 and 1 will hold an error code.

The meaning of the error code depends on the individual Hiperface DSL® encoder and will be detailed in the product manual.

Register 20h: **Parameter 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
Parameter Channel Buffer Byte 0							
Bit 63				Bit 56			

Register 21h: **Parameter 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
Parameter Channel Buffer Byte 1							
Bit 55				Bit 48			

Register 22h: **Parameter 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
Parameter Channel Buffer Byte 2							
Bit 47				Bit 40			

Register 23h: **Parameter 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
Parameter Channel Buffer Byte 3							
Bit 39				Bit 32			

Register 24h: **Parameter Channel Buffer Byte 4**

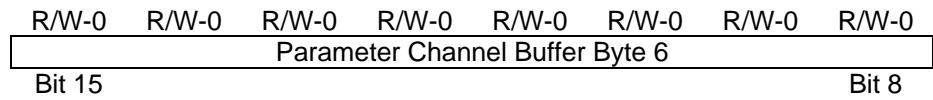
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Parameter Channel Buffer Byte 4							
Bit 31				Bit 24			

Register 25h: **Parameter Channel Buffer Byte 5**

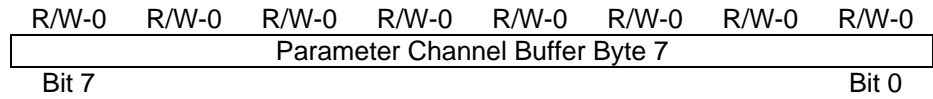
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Parameter Channel Buffer Byte 5							
Bit 23				Bit 16			

HIPERFACE DSL®

Register 26h:

Parameter Channel Buffer Byte 6

Register 27h:

Parameter Channel Buffer Byte 7

Bit 63-0

Parameter Channel Buffer Byte 0-7:

8 bytes for long message response (read operation) or for “Long message” write operation.

Bit 63-48

Long Message Error Report Byte 0-1:

2 bytes for reporting encoder resource errors resulting from the previous “Long message” operation.

5.3.17. Long Message Address PC_ADD

In the Long Message Address registers **PC_ADD_H/PC_ADD_L** the address and addressing mode for “Long messages” over the Parameter channel are defined.

Additionally, the Long Message Address register 28h (**PC_ADD_H**) contains error indications resulting from “Long message” operations. In the case of such an error the Parameter Channel Buffer bytes 0 and 1 will report the error code corresponding to this condition (see chapter 6.7.2).

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 Identifier. Must be ‘1’.
- 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 = “Long message” offset addressing mode. The offset value from registers **PC_OFF_H/PC_OFF_L** is used as sub address in the selected database entry resource.
 0 = “Long message” offset-free addressing mode. The offset value from registers **PC_OFF_H/PC_OFF_L** is not used.
 Read access:
 1 = Last “Long message” caused an error.
 0 = Last “Long message” processed correctly.
- Bit 12 **LIND**: Long Message Indirect Mode
 1 = “Long message” indirect addressing. The address contents stored at the indicated database entry will be evaluated by this operation.
 0 = “Long message” direct addressing. The operation acts on the database entry indicated by the current address.
- Bit 11-10 **LLEN**: Long Message Data Length
 11 = 8 data bytes
 10 = 4 data bytes
 01 = 2 data bytes
 00 = no data bytes
- Bit 9-0 **LADD**: Long Message Address
 10 bit database entry address for “Long message” operation.

5.3.18. Long Message Address Offset PC_OFF

The Long Message Address Offset registers **PC_OFF_H/PC_OFF_L** are used for “Long message” operations if **LOFF** in register 28h is set. In this case the value **LOFFADD** from these registers used to address the sub-space address of a multi-byte encoder resource.

These registers are write-only.

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						W-0
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							W-0
Bit 7							Bit 0

Bit 15 **LID**: Long Message Identifier. Must be ‘1’.

Bit 14-0 **LOFFADD14:0**: Long Message Offset Value
These bits will hold the 15 bit offset value for a “Long message” address offset.

5.3.19. Parameter Channel Control PC_CTRL

The Parameter Channel Control register **PC_CTRL** manages the start of “Long message” transactions. After setting all “Long message” registers (registers **PC_BUFFER0...7**, **PC_ADD_H/PC_ADD_L**, **PC_OFF_H/L**) the “Long message” is transmitted to the DSL Slave by setting bit **LSTA**.

Register 2Ch: **Parameter 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 **Unimplemented**: Read as ‘0’.

Bit 0 **LSTA**: Long Message Start Control
1 = A “Long message” transaction will be started with the currently stored values in the “Long message” registers.
0 = No effect.

5.3.20. Pipeline Channel Status PIPE_S

The Pipeline Channel Status register **PIPE_S** gives information on the current status of the pipeline channel (see chapter 2.5).

PIPE_S is only accessible as a DSL Master register if SPI-PIPE is disabled (**SPPE** in register **SYS_CTRL** cleared).

Else, the value of **PIPE_S** is transmitted as the first byte of each read request via SPI-PIPE (see chapter 4.3); in that case the first 4 bits are sent as '0101' in order to check the SPI-PIPE interface against stuck-at errors.

Reading this register causes the current data to be extracted from the FIFO buffer and stored in a temporary register, so a subsequent reading to the **PIPE_D** register can be considered as done at the same time as reading the **PIPE_S** register. Through this mechanism misalignment between status and data information is avoided in case new data is received in the Pipeline channel while the FIFO buffer is accessed.

PIPE_S is a read-only register.

Register 2Dh: **Pipeline 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 **Unimplemented:** Read as '0'.

Bit 3 **POVR:** Pipeline channel buffer overrun
 1 = The 8 byte FIFO buffer for the pipeline channel data was filled and values have been discarded since the last reading.
 0 = FIFO buffer for pipeline channel data is not filled.
 This bit is cleared after reading.

Bit 2 **PEMP:** Pipeline channel buffer empty
 1 = A read request was issued but the FIFO buffer for the pipeline channel data is empty. **PIPE_D** will hold the value 00h in this case.
 0 = No buffer empty error.
 This bit is updated following each FIFO buffer access.

Bit 1 **PERR:** Pipeline channel bit encoding error
 1 = The bit level encoding of the current pipeline channel data is erroneous.
 0 = No bit encoding error.
 This bit is stored in the FIFO buffer along the respective pipeline data byte.

Bit 0 **PSCI:** Pipeline channel special character indication
 1 = A special character was received in the Pipeline channel.
 0 = No special character indication.
 This bit is stored in the FIFO buffer along the respective pipeline data byte.
 Special characters are typically used as data separator or for special events signaling. In order to get information about which special character was received register **PIPE_D** has to be read.

All 8b10b special characters can be used over the pipeline, except the symbol “K30.7” which is used by Hiperface DSL[®] to indicate “no data” and is not stored in the FIFO buffer.

The following Table 26 shows which 8b10b special characters are supported.

Special character	Encoding 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

Table 26. Supported 8b 10b special characters in the Pipeline channel.

5.3.21. Pipeline Channel Data PIPE_D

The Pipeline Channel Data register **PIPE_D** holds the pipeline channel data that is stored in an 8 byte FIFO buffer.

If the buffer is filled and a new value arrives before **PIPE_D** is read out, the oldest value will be discarded and bit **POVR** in **PIPE_S** will be raised.

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

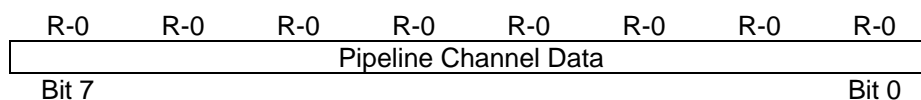
PIPE_D is only accessible as a DSL Master register if SPI-PIPE is disabled (**SPPE** in register **SYS_CTRL** cleared).

Else, the value of **PIPE_D** is transmitted as the second byte of each read request via SPI-PIPE (see chapter 4.3).

The instant that the **PIPE_S** register is accessed the corresponding value of **PIPE_D** will be frozen to allow synchronicity between both status and data information.

PIPE_D is a read-only register.

Register 2Eh: **Pipeline Channel Data**



Bit 7-0

Pipeline Channel Data

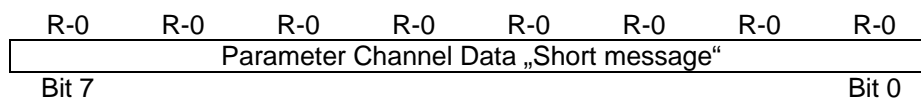
8 bit value of pipeline channel data FIFO buffer.

5.3.22. Parameter Channel Short Message PC_DATA

The Parameter Channel Data „Short message“ register **PC_DATA** holds the result of „Short message“ transactions.

„Short message“ transactions are generated when remote (DSL Slave) register operations are carried out. Generally, **FRES** (in register **EVENT_L**) must be asserted after a transaction was started before **PC_DATA** holds valid information.

Register 2Fh: **Parameter Channel Data „Short message“**



Bit 7-0

Parameter Channel Data „Short message“

8 bit value of requested remote register.

5.4. DSL Master Function Registers Channel 2

The registers of the DSL Master IP-core in channel 2 are used by the protocol logic to access position values of channel 2. Table 27 gives a list of these registers for channel 1.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Reset
0Fh	ENC2_ID	ENC2_ID7:0								0000 0000
18h	VPOS24	Absolute position channel 2 byte 4								0000 0000
19h	VPOS23	Absolute position channel 2 byte 3								0000 0000
1Ah	VPOS22	Absolute position channel 2 byte 2								0000 0000
1Bh	VPOS21	Absolute position channel 2 byte 1								0000 0000
1Ch	VPOS20	Absolute position channel 2 byte 0								0000 0000
1Dh	VPOSCRC2_H	Absolute position CRC 2 byte 1								0000 0000
1Eh	VPOSCRC2_L	Absolute position CRC 2 byte 0								0000 0000
1Fh	STATUS2	Status channel 2								0000 0000
3Fh	DUMMY2	No data								---- ----

Table 27. DSL Master Register Description channel 2.

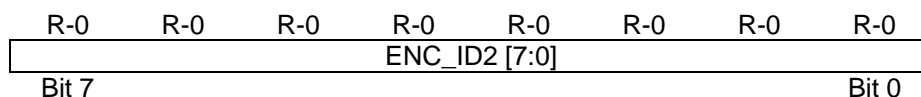
5.4.1. Channel 2 Encoder ID ENC2_ID

The Channel 2 Encoder ID register **ENC2_ID** holds the identification code of the secondary channel of the DSL encoder.

The contents of **ENC2_ID** are defined in the actual motor-feedback implementation and can be found in the appropriate data sheet.

This register is read-only.

Register 0Fh: **Channel 2 Encoder ID**



Bit 7-0

ENC_ID2: Secondary Channel Identification

02h: SIL2 arrangement. Secondary channel transports same position data as primary channel.

03h: SIL3 arrangement. Secondary channel transports position data from secondary sensor source within Hiperface DSL® motor-feedback system.

5.4.2. Channel 2 Status STATUS2

The Channel 2 Status register **STATUS2** contains the status information of channel 2 of the DSL encoder. The contents are also made available in the Online Status of SPI2 (see chapter 5.2.2)



Please note that the **STATUS2** bits are not latched and therefore behave different from the status summary of channel 1 (see chapter 5.3.15). It is not necessary (nor possible) to acknowledge **STATUS2** bits.



WARNING

Any error indication in the SPI2 Online Status is potentially critical and/or safety-relevant. Appropriate measures must be implemented by the user application.

Register 18h: **Channel 2 Status**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TOG2	TEST2	ERR2	FIX2				
Bit 7							Bit 0

Details of the register contents can be found in chapter 5.2.2.

5.4.3. Absolute Position Channel 2 VPOS2

The Absolute Position Channel 2 registers **VPOS2** hold the position value of the secondary channel of the attached motor-feedback system. This absolute position is transmitted every eight protocol frames if the validity of the data transmission has been verified.

The absolute position is not synchronized to the SYNC signal.

Only those bits of VPOS2 are enabled that lie in the range actually measured by the motor-feedback system. All other, more significant bits read as '0'. The number of measurable bits can be determined from bits 9..0 of **ENC2_ID** in registers **ENC2_ID0...2**.

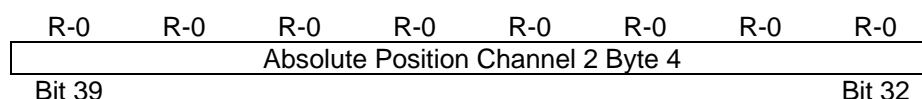


The position VPOS2 is output as a complementary value to VPOS from Absolute Position Channel 1.

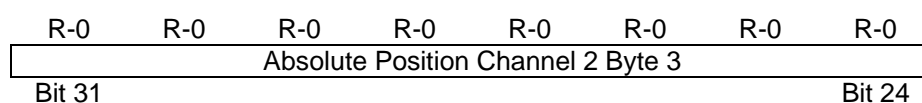
The unit of the position value is [steps].

These registers are read-only.

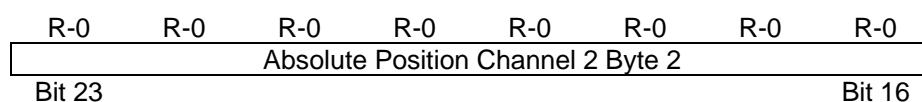
Register 19h: **Absolute Position Channel 2 Byte 4**



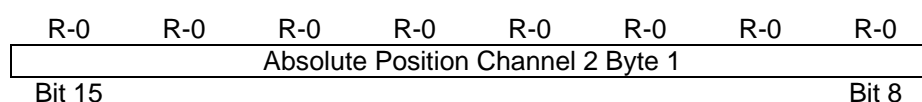
Register 1Ah: **Absolute Position Channel 2 Byte 3**



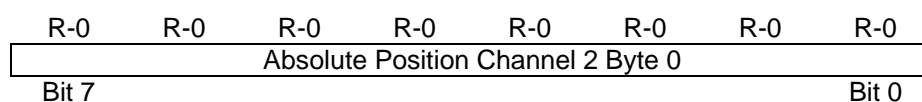
Register 1Bh: **Absolute Position Channel 2 Byte 2**



Register 1Ch: **Absolute Position Channel 2 Byte 1**



Register 1Dh: **Absolute Position Channel 2 Byte 0**



Bit 39-0

Absolute Position Channel 2Byte 4/3/2/1/0:

40 bit motor-feedback system channel 2 position value, absolute value transmitted as complement.

5.4.4. Position Checksum Channel 2 POSCRC2

The Position Checksum Channel 2 registers **POSCRC2** hold the cyclic redundancy checksum (CRC) of the absolute position VPOS2 (see chapter 5.4.2).

The CRC is checked within the DSL Master IP-core. To ensure in a safety application that the CRC machine is working within the IP-core these registers allow external cross-checking within the diagnostic test interval defined in the chapter on Safety Implementation (see Hiperface DSL[®] Safety Implementation Manual).

The CRC is built with the following CRC parameters:

Parameter	Value
CRC order	16 bits
CRC polynomial	C86Ch
Initial value	0000h
Final XOR value	00FFh
Reverse data bytes	No
Reverse CRC before final XOR	No

Table 28. POSCRC parameters.

The order of bytes for calculating the CRC is shown in the following figure:

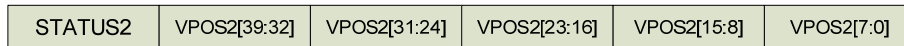
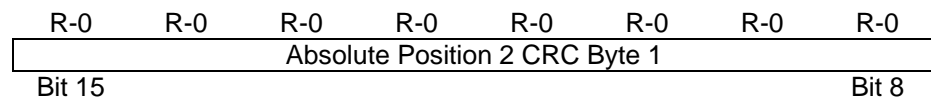
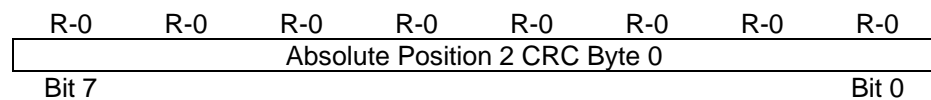


Figure 21. Byte order for CRC calculation.

Register 1Eh: **Absolute Position CRC Byte 1**



Register 1Fh: **Absolute Position CRC Byte 0**



Bit 15-0

Absolute Position 2 CRC:

16 bit cyclic redundancy checksum (CRC16) of absolute position and status summary in Channel 2.

5.5. Function Registers DSL Slave

The remote registers of the DSL Slave are mirrored inside the DSL Master at register addresses 40h to 7Fh. They are accessed via „Short message“ transactions (see chapter 6.7.1).

The minimum amount of remote registers present in any DSL Slave is detailed in Table 29. Actual DSL Slave implementations can implement more remote registers.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on 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
7Eh	MAIL	Slave Mail								0000 0000
7Fh	PING	Slave Ping ¹								0000 0000

Table 29. DSL Slave remote registers.

5.5.1. Encoder Status ENC_ST

The Encoder Status registers **ENC_ST** hold all possible slave system errors, events, and warnings of channel 1 of the DSL encoder.

The assignment of individual bits to slave system states is defined in the actual DSL Slave implementation.

Hiperface DSL® defines that **ENC_ST0** (register 40h) states pertain to DSL Slave interface errors/events/warnings. All other Encoder Status registers (41h – 47h) relate to encoder resources.



Note that all bits of a single Encoder Status register are OR-combined together and mirrored to DSL Master register bits in **SUMMARY** (1Fh) (see Table 30 and Figure 19 in chapter 5.3.15). This allows for a fast response to slave states from corresponding groups.



Bits in the Encoder Status registers can only be set by the DSL Slave and they can only be cleared by the drive application (acknowledgment).

¹ After a protocol reset the **PING** register holds the slave interface version, see chapter 5.5.4).

Encoder Status	SUMMARY bit (DSL Master 1Fh)	Status group
ENC_ST0 (40h)	SUM0	DSL Slave interface
ENC_ST1 (41h)	SUM1	General purpose
ENC_ST2 (42h)	SUM2	General purpose
ENC_ST3 (43h)	SUM3	General purpose
ENC_ST4 (44h)	SUM4	General purpose
ENC_ST5 (45h)	SUM5	General purpose
ENC_ST6 (46h)	SUM6	General purpose
ENC_ST7 (47h)	SUM7	General purpose

Table 30. Encoder Status and Summary registers dependence.

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
DSL Slave Interface 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 7-0 **DSL Slave Interface Status**
 Individual bits show different errors/events/warnings. The meaning of each bit is determined by the individual DSL Slave implementation.
 1 = Error/Event/Warning state.
 0 = DSL Slave interface in normal operation.
- Bit 63-8 **Encoder Status**
 Individual bits show different errors/events/warnings. The meaning of each bit is determined by the individual DSL Slave implementation.
 1 = Error/Event/Warning state.
 0 = Encoder in normal operation.

5.5.2. Slave RSSI SRSSI

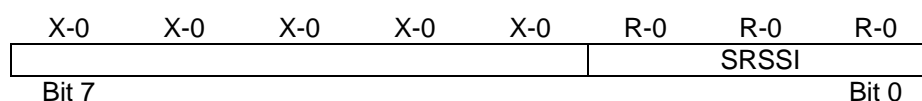
The Slave Received Signal Strength Indicator (RSSI) register **SRSSI** gives an indication of the signal received by the slave.

The value of the register is updated from frame to frame with worse results only. After a read access to this register was performed it is reset to value 5 (maximum strength).

This register is read-only.

Register 7Ch:

Slave RSSI



Bit 7-3 **Unimplemented:** Read as '0'.

Bit 2-0 **Slave RSSI Value**

The Slave RSSI value ranges from "0" (worst signal strength) to "5" (best signal strength).

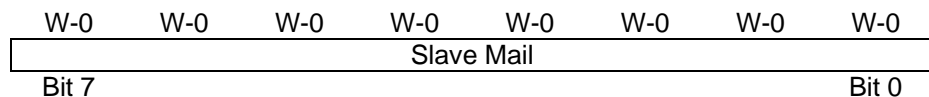
5.5.3. Slave Mail MAIL

The Slave Mail register **MAIL** is a general purpose register that allows for fast communication with the DSL motor-feedback system processor. The content of the Slave Mail register is transmitted to the encoder processor in the fastest way possible.

Slave Mail is used to transmit test messages to the DSL motor-feedback system in safety applications. Please refer to the Hiperface DSL® Safety Implementation Manual for correct usage in safety applications.

This register is write-only.

Register 7Eh: **Slave Mail**



Bit 7-0

Slave Mail

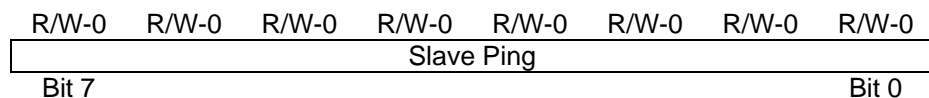
8 bit slave mail data for general purpose.

5.5.4. Slave Ping PING

The Slave Ping register **PING** allows to conduct a DSL Slave side link test as it can be written to and read from remotely without any effect on the DSL Slave interface.

At start-up it is initialized with the hardware version of the DSL Slave interface.

Register 7Fh: **Slave Ping**



Bit 7-0

Slave Ping

8 bit DSL encoder hardware version at start-up. This value is "0001 0001". After first read-out, the Slave ping value can be used for slave side link test.

6. Central Functions

In this chapter an overview is given how central sensor functionality is accessed through interfaces and registers (see chapter 2).

6.1. System Start

If the motor-feedback system is supplied with power a reset sequence is mandatory to achieve a defined system start state of the DSL Master IP-core (see chapter 4.6).

Figure 22 shows the state chart of the system start.

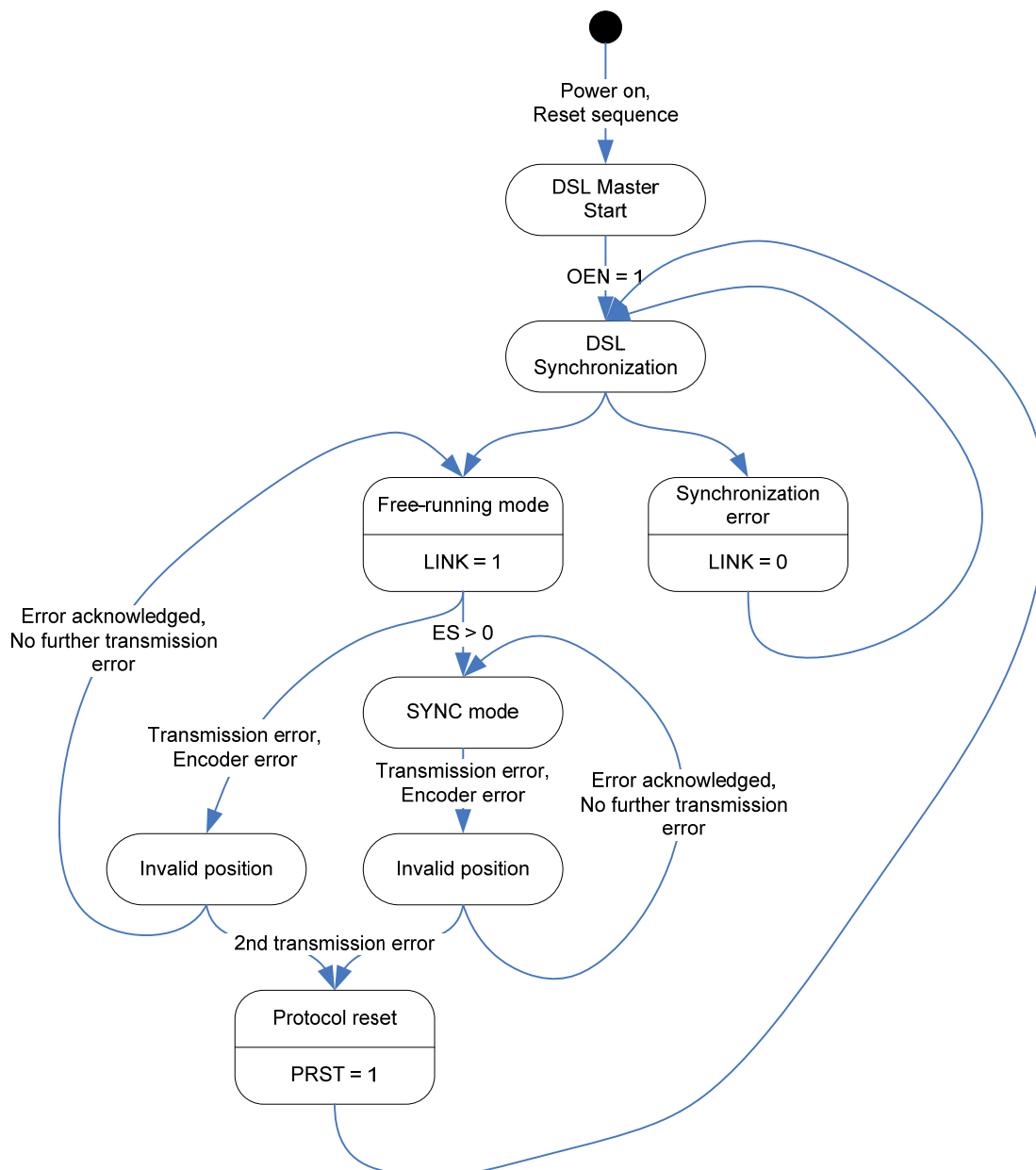


Figure 22. State chart DSL system start.

Individual states are characterized in Table 31.

State	Precondition	Indication
DSL Master Start	Power on, Reset sequence	Communication via SPI1 possible
DSL Synchronization	OEN = 1 (Register SYS_CTRL)	None
Synchronization error	Time-out during DSL synchronization	LINK = 0 (Register MASTER_QM)
Free-running mode	Successful DSL synchronization	LINK = 1 (Register MASTER_QM)
SYNC mode	ES > 0 (Register SYNC_CTRL), Cyclic signal to SYNC input	Synchronous encoder position in registers POS0...4
Invalid position	Single transmission error or encoder error	Error bit set in EVENT_H , EVENT_L or in Online DSL Master Status
Protocol reset	Two successive transmission errors	PRST = 1 (Register EVENT_H or in Online DSL Master Status)

Table 31. States during DSL system start.

6.2. System Diagnostics

Hiperface DSL® allows for comprehensive system diagnostics regarding communication quality both during engineering of a DSL system as well as during normal operation.

6.2.1. System Diagnostics during Engineering

During engineering of a DSL system several registers help diagnose correct usage and operation. This includes:

- Quality monitor **MASTER_QM**
- Edges register **EDGES**
- Delay register **DELAY**

After enabling the DSL link (bit **OE**, see chapter 6.1) the flag **LINK** in register **MASTER_QM** must be checked for set '1' value. This indicates that the link to the motor-feedback system was successfully established.

If this bit remains cleared for more than the specified start-up time (see Table 6) a fundamental problem exists in the connection between drive and motor-feedback system.

Please check if the encoder is supplied with power.

Please also check with an oscilloscope if the data lines between drive and encoder show level transitions in a frequency range of the transmission frequency (see Table 6).

The delay register (see chapter 5.3.8) helps you in determining if the **Line Delay** of the DSL signal lies within the specification. The delay mainly results from the cable length between drive and motor-feedback system. Additionally, the choice of interface driver (RS485 transceiver) can influence the signal delay.

The value of the **EDGES** register (see chapter 5.3.7) indicates how well or bad the DSL Master can sample the communication signal coming from the motor-feedback system.

Please start with checking the **Bit Sampling Pattern** with the motor switched off. If several bits in the sampling pattern are set (more than 4) the shielding concept of the encoder cable should be checked. The target should be to have only one set bit in the sampling pattern under undisturbed conditions.

In the second step please check the sampling pattern with enabled motor and if possible in the target application. In these cases a maximum of seven bits must be set only in the **EDGES** register.



If eight bits of the EDGES register are set under any circumstances the operation of the DSL motor-feedback system is not guaranteed.

6.2.2. System Diagnostics during Operation

During operation of the DSL system the following registers display system diagnostics:

- Delay register **DELAY**
- Quality monitor **MASTER_QM**
- Slave received signal strength indicator **SRSSI**

The Delay register (see chapter 5.3.8) contains the value **RSSI** that lies in the range from 0 to 12. It indicates the run-time link quality regarding signal strength.

The quality monitor (see chapter 5.3.3) contains the value **QM** that lies in the range from 0 to 15. **QM** indicates the run-time link quality regarding transmission errors.

For continuous monitoring of link quality it is advisable to query both of these values cyclically.

An event-based monitoring is also possible by querying the event bits **QMLW** or **PRST**. These bits indicate a lowering of **QM** below the value of 14 (quality low) or a broken link due to a **QM** value of 0 or a low **RSSI** value of 2 or less. The following table shows the possible states:

Quality Monitor value	RSSI value	QMLW	PRST	Link state	
15	12 ... 2	0	0	Good link quality	●
14 ... 1		1	0	Low link quality	●
0		1	1	Link broken	●
15	1 ... 0	0	1	Link broken	●
14 ... 0		1	1	Link broken	●

Table 32. Quality monitor and RSSI values.

Frequent errors can indicate that the shielding concept of the DSL connection is not sufficient or that the cable does not conform to the specification.

The Slave RSSI register (see chapter 5.5.2) contains the value **SRSSI** that lies in the range from 0 to 5. **SRSSI** indicates the run-time link quality regarding signal strength of data transmitted to the DSL encoder.

6.3. Incremental Position

Within the horizontal channel of the DSL motor-feedback system the incremental position and speed of the encoder shaft are transmitted. These values are the core process values for the control loop of the drive application.

Hiperface DSL® stores the incremental position in the DSL Master **POS0...4** and the speed value in the registers **VELO...2**.

The position is presented as a 40 bit value including the angular position (single-turn value) as well as the number of revolutions (multi-turn value). Only position bits actually measured by the motor-feedback systems are made available and stored as a right-justified value in the registers. The other bits (most significant bits) will be constantly cleared to '0' (see examples a-c in Figure 23).

The incremental position is automatically added up to present the current absolute position of the motor-feedback system. This mechanism is checked automatically by the DSL Master by comparing the incremental position to the absolute position (see chapter 6.4).

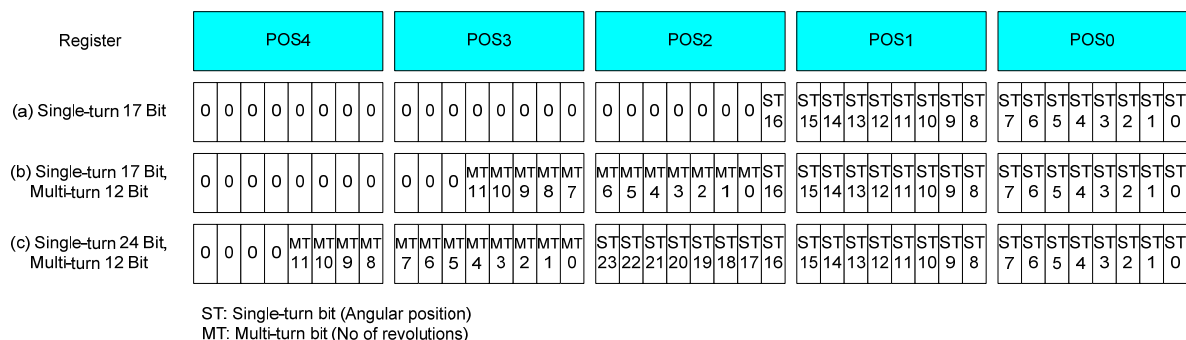


Figure 23. Position value format.

The incremental position of the motor-feedback system is sampled and transmitted if a SYNC signal is supplied to the DSL Master. This SYNC signal can be generated in two different ways (see chapters 4.4 and 5.3.2).

6.3.1. Free Running Mode

In free running mode the SYNC signal is generated automatically by the DSL Master with the maximum frame transmission frequency (see Table 6). The free running mode is the default of the DSL Master upon start-up.

The mode can also be chosen manually by setting the value **ES** = "0" in register **SYNC_CTRL**.



Please note that in free running mode signals on the SYNC input are not regarded.

Querying the position or speed value is illustrated in Figure 24 and Figure 25.

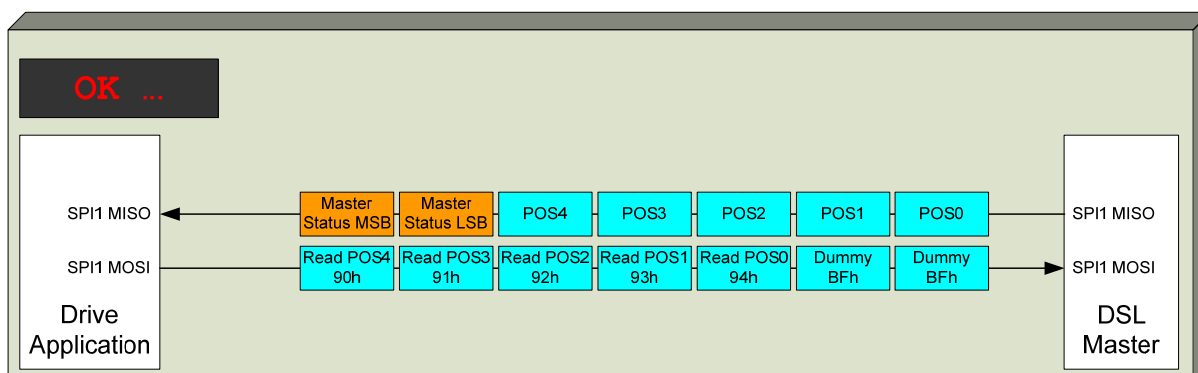


Figure 24. Querying of position registers in free running mode.

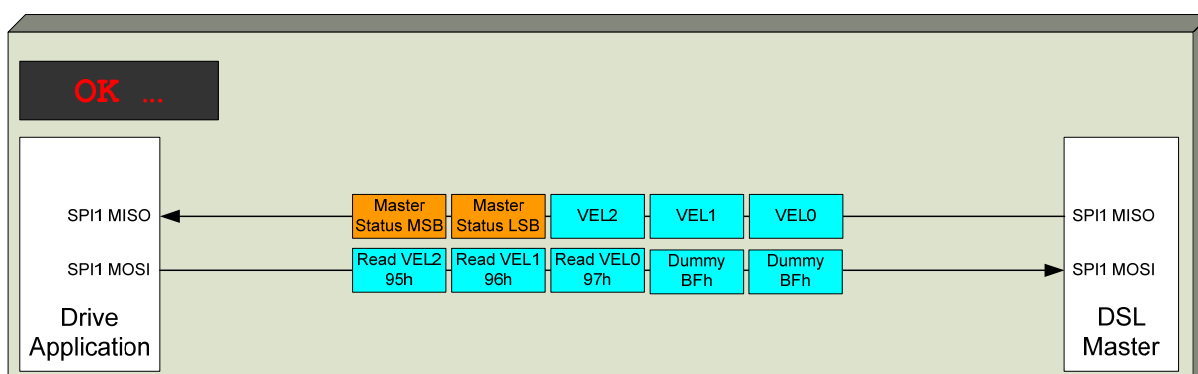


Figure 25. Querying of speed registers in free running mode.

6.3.2. SYNC Mode

In SYNC mode the DSL Master depends on a supplied cyclic drive signal. This drive signal initiates position measurements and allows querying position and speed values synchronous to the drive signal. The drive signal must be supplied to the SYNC input with characteristics specified for the DSL Master (see chapter 4.4).

The position is available after a fixed delay to the drive signal leading edge.

The following steps must be observed to use the SYNC mode:

- 1.) Supply a correct drive signal to the SYNC input. This signal must conform to pulse width and cycle time specifications.
- 2.) By setting or clearing the bit **SPOL** in register **SYS_CTRL** choose whether the rising or falling edge of the drive signal is used to trigger position measurements. The specified latency of the DSL system is measured from this edge.
- 3.) Set the correct divider **ES** in register **SYNC_CTRL**. This divider determines how many position samplings and transmissions are made for each drive signal.



The divider ES must be chosen so that the cycle time between two position samplings conforms to the specified boundaries (packet cycle time) in Table 6.

The boundaries for the divider **ES** can be calculated from:

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

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

These formulas use the following symbols:

Symbol	Description
t_{Sync}	Cycle time of pulse signal on SYNC input
t_{Min}	Minimum cycle time for DSL frame transmission (12.15 μs)
t_{Max}	Maximum cycle time for DSL frame transmission (24.30 μs)

The following Table 33 lists typical cycle times of the drive signal and the valid range of divider values **ES**.

Frequency of SYNC signal [kHz]	Cycle time of SYNC signal [μs]	Minimum Value ES	Maximum Value ES
2	500	21	41
4	250	11	20
6.25	160	7	13
8	125	6	10
16	62.5	3	5
40	25	2	2
41..82	24.3 ... 12.15	1	1

Table 33. SYNC cycle times and valid ES values.

After following this sequence the SYNC mode is enabled. Within the specified “Start-up time” (see Table 6) the protocol synchronizes to the supplied SYNC signal. After this time the position value is available with constant latency after each packet transmission (see Figure 26).

The position value can be queried from the registers **POS0...4** of the DSL Master via SPI1 (see chapter 0).

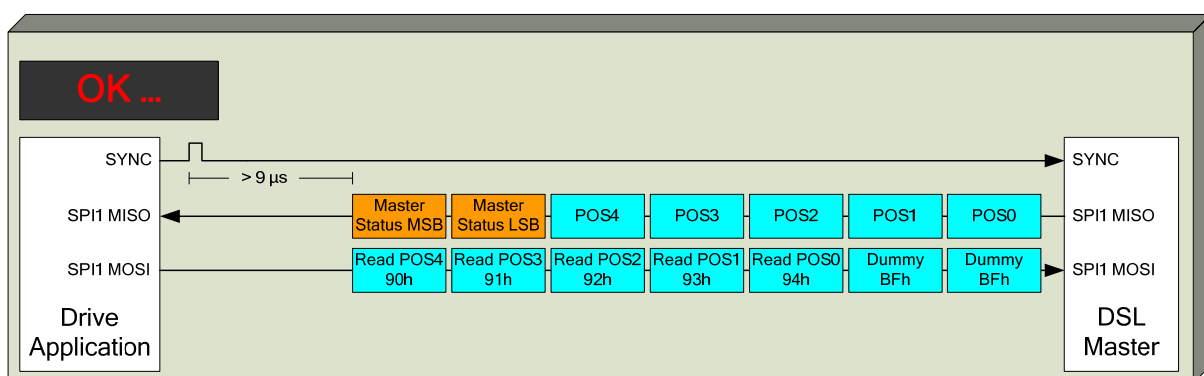


Figure 26. Querying of Incremental Position registers in SYNC mode.



Please note that querying of less than the full five position registers might be appropriate according to the application. This allows faster read-out of the position.

Similarly, the speed value of the motor-feedback system can be read out. The speed value is also measured and transmitted synchronously to the SYNC signal. This is shown in Figure 27.

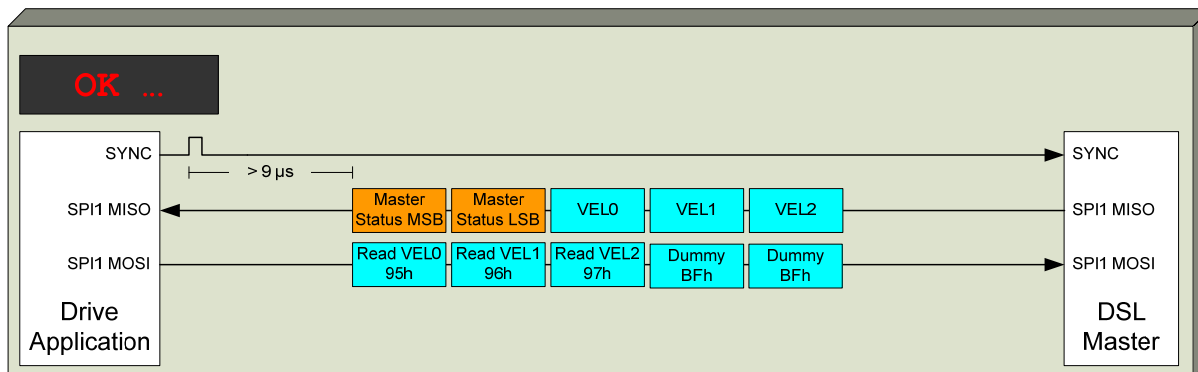


Figure 27. Querying of Speed registers in SYNC mode.

6.4. Absolute Position Channel 1

The absolute position of the motor-feedback system is not only transmitted incrementally (see chapter 6.3) but also as complete absolute position. This allows internal validation of the data transmission.

The complete absolute position is transmitted every eight protocol frames and therefore is slower than the incremental position.



The absolute position is not synchronized to the last drive cycle supplied to the DSL Master IP-core. You should not use the absolute position for the drive position or velocity loop.

The absolute position is stored in the registers **VPOS0...4** and can be queried over the SPI1 interface, see Figure 28.

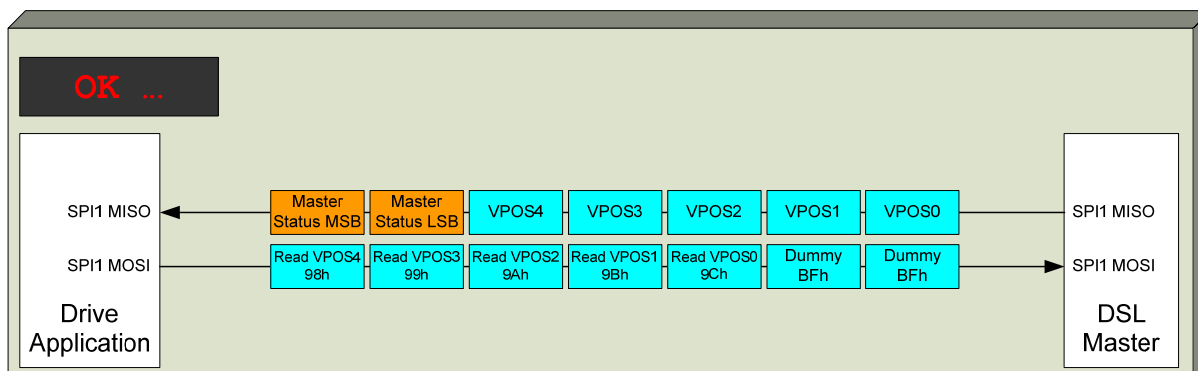


Figure 28. Absolute position query.

If the DSL Master detects a difference between transmitted absolute position and integrated incremental position the error bit **POS** in the register **EVENT_H** is raised (see chapter 6.5).

6.5. Absolute Position Channel 2

For safety applications a second channel for absolute position exists. This channel 2 serves as a redundancy to detect errors of the DSL motor-feedback system.

Depending on the individual DSL motor-feedback system the absolute position of channel 2 has different origins as shown in Table 34.

DSL encoder safety architecture	Channel 2 absolute position origin	Absolute position channel 2 resolution
Category 2 (SIL 2) (One sensor, internal diagnostics)	Same as channel 1, separate diverse transmission of position data	Same as channel 1
Category 3 (SIL 3) (Two sensors)	Separate sensor to channel 1	Depending on separate sensor (less or equal to channel 1)

Table 34. Channel 2 absolute position origin.

The resolution of absolute position channel 2 can be less than the resolution of channel 1 if a second sensor is used in the DSL motor-feedback system with a lower performance than the main sensor.



Please note that in this case, cross-checking both absolute positions of channel 1 and 2 must be limited to an amount of most significant bits of positions as indicated in the product datasheet.



WARNING

Cross-checking of absolute position of channel 1 and 2 is not performed internally by the DSL Master IP-core. Due to the safety architecture of an FPGA this check has to be performed by electronic components external to the FPGA containing the IP-core.

The complete absolute position channel 2 is transmitted every eight protocol frames, synchronous to the absolute position channel 1.

The absolute position channel 2 is stored in the registers **VPOS2_0...4** and can be queried over the SPI2 interface, see Figure 29.

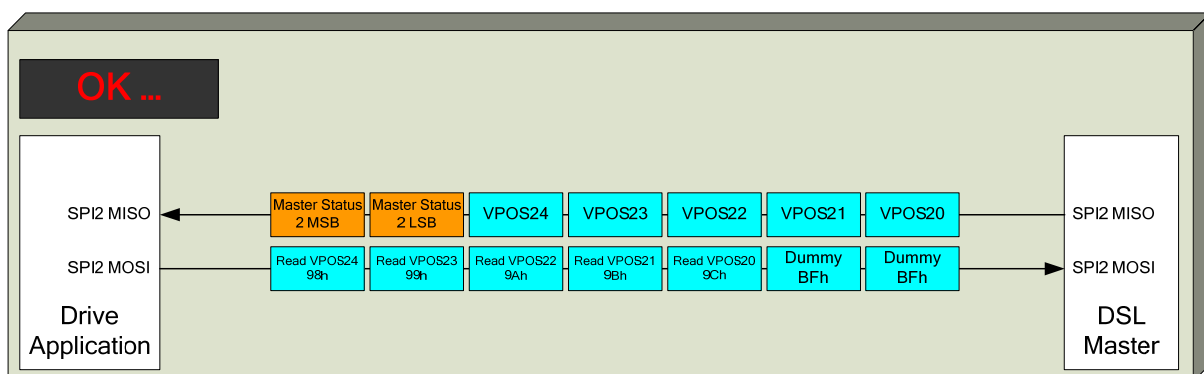


Figure 29. Absolute position channel 2 query.

6.6. Status and Error Messages

Hiperface DSL® allows status monitoring of the motor-feedback system in several ways.

Depending on criticality of a status or error message different indication mechanisms are used to inform the drive application.

6.6.1. Event Registers

The event registers **EVENT_H**, **EVENT_L** (see chapter 5.3.4) contain all core error and status indicators of the DSL Master. All events are updated at least every 200 µs.

In detail, the register **EVENT_H** contains all critical error messages of the motor-feedback system. Recommended error handling is described in chapter 5.3.4.

The register **EVENT_L** contains all warnings and status messages of the motor-feedback system. Recommended error handling is described in chapter 5.3.4.

Each error or warning state indicated in the event registers must be acknowledged by clearing the corresponding error bit. The DSL Master will not reset these bits automatically.

The following shows this mechanism with an example (error in transmitted incremental position, bit **POS**).

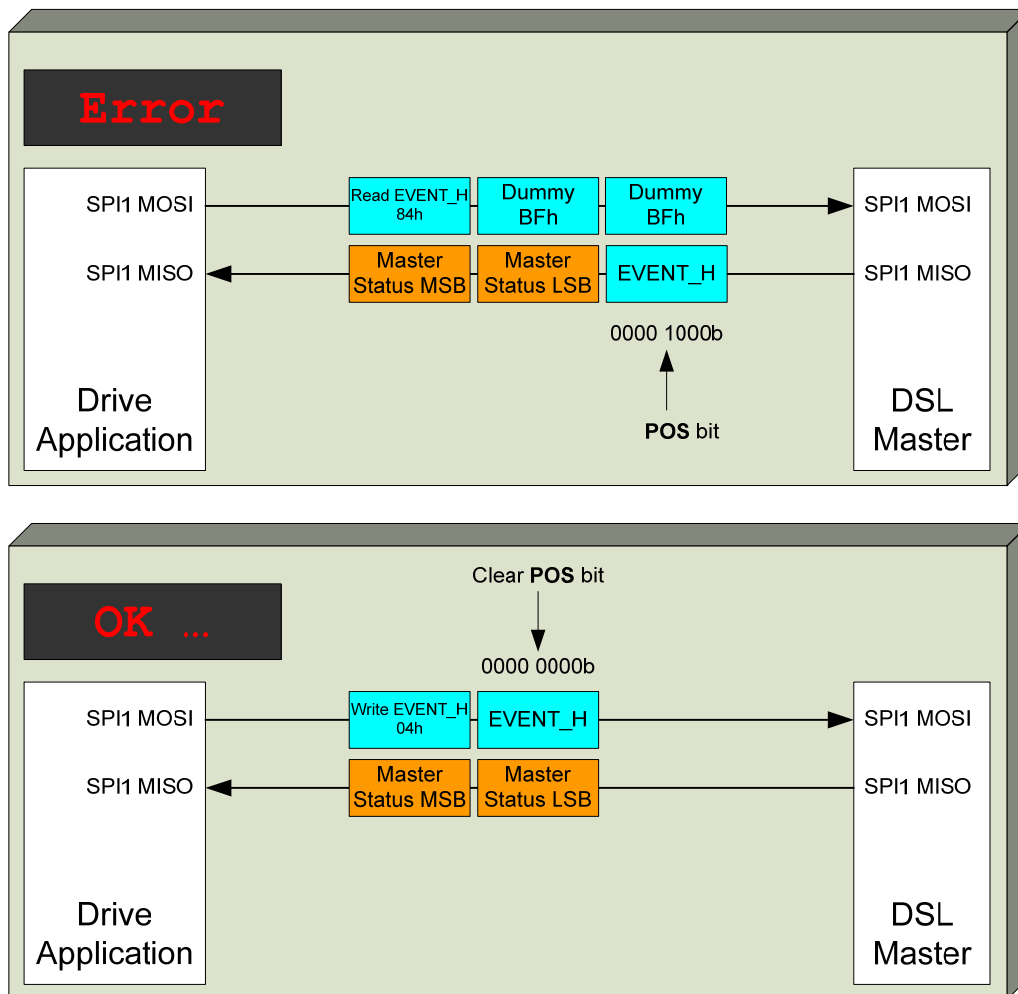


Figure 30. Acknowledgment of event bits.

Three mechanisms can be implemented in the drive application to react timely to event register messages:

- The registers are polled cyclically.
- The online DSL-Master status is polled cyclically. Event registers are mirrored here (see chapter 6.6.2).
- All or individual events of the event registers can be masked in the event mask registers (registers **MASK_H**, **MASK_L**, see chapter 5.3.5) for output of events on the **INT** interface (see chapter 4.5).

6.6.2. Online DSL-Master Status

The Online DSL-Master status (see chapter 5.2) is transmitted during each SPI1 communication between drive application and DSL-Master and contains the error and status messages of the event registers.

Differently to direct querying of the event registers the Online DSL-Master status only shows the current status values. As soon as the error state of the motor-feedback is over the online status will no longer indicate the error.

The event registers latch error states until they are acknowledged. After acknowledgment the event registers will be reset (see chapter 5.3.4)

The Online DSL-Master status is updated at least every 200 µs.

6.6.3. Status Summary Motor-feedback System

Additional, more detailed errors and warnings of the motor-feedback system are indicated in the status summary register **SUMMARY** (18h, see chapter 5.3.15).

Each bit of the register indicates an error state of a motor-feedback system functionality as detailed in Table 35. This table also details the safety relevancy of any of these error groups.



An error indication with critical safety relevancy must always be acted upon. If no further diagnostics are performed, the indication of a critical safety error must be followed by the establishment of a safe state of the drive system (e.g. safe operating stop of the motor).

Bit No	Error Group	Safety Relevancy
0	Incremental position error	Critical for safe speed functions
1	Absolute position error	Critical for safe position functions
2	Initialization error	Critical for any safety function
3	Monitoring error	Warning for any safety function
4	Resource access error	Not safety-critical
5	Reserved	
6	Reserved	
7	Reserved	

Table 35. Motor-feedback system error groups.

A set bit in the status summary register actually indicates that one or more of eight individual motor-feedback system errors were detected. The individual error can be determined by querying the remote encoder status register **ENC_ST** (see chapters 5.5.1, 6.6.4).



Please note that the error bit **SUM** in the event register **EVENT_H** indicates an aggregation of the summary of all error groups (see chapter 5.3.5).



Please note that reading of the detailed error messages of the motor-feedback system allow a more appropriate response to the indication of any error in the status summary.

6.6.4. Error Messages Motor-feedback System

Errors detected within the motor-feedback system are indicated in the remote encoder status registers **ENC_ST**.

Access to a single remote encoder status register can take up to 1.2 ms.

A summary over error groups is indicated more timely in the status summary register within the DSL Master (see chapter 6.6.3).

Errors of the motor-feedback system are logically grouped for simplified error handling.

The following Table 36 lists all error messages and recommended error handling.

Error Group	Error No	Description	Error Handling
0	0	Protocol reset indication	<p>The position and speed of the DSL encoder must be assumed to be wrong after this error has been indicated.</p> <p>Automatic restart of communication If error is repeated frequently, please check cable connection. If error persists, a general hardware or mechanic error must be assumed. Please call customer service.</p>
	1	<p>Test in progress</p> <p>A test was requested by the drive system. Any error indication caused by this test will be indicated along with "Test in progress" to differentiate from true errors.</p>	<p>If "Test in progress" is accompanied by an error indication caused by a previous test message no action must be taken but clearing this bit and the corresponding error bit. If "Test in progress" is shown without the expected error indication, a safety-relevant error must be assumed in the DSL encoder. A safe state must be entered by the drive system in this case. Restart of encoder. If condition persists, a general hardware or mechanic error must be assumed. Please call customer service.</p>
	4	Position error: Tracking filter error	<p>The position and speed of the DSL encoder must be assumed to be wrong after this error has been indicated.</p> <p>A safe state must be entered by the drive system.</p> <p>Check encoder/motor assignment. If error persists, a general hardware or mechanic error must be assumed. Please call customer service.</p>
	5	Position error: Vector length error	<p>The position and speed of the DSL encoder must be assumed to be wrong after this error has been indicated.</p> <p>A safe state must be entered by the drive system.</p> <p>Check encoder/motor assignment. If error persists, a general hardware or mechanic error must be assumed. Please call customer service.</p>
	6	Position error: Counter error	<p>The position and speed of the DSL encoder must be assumed to be wrong after this error has been indicated.</p> <p>A safe state must be entered by the drive system.</p> <p>Check encoder/motor assignment. If error persists, a general hardware or mechanic error must be assumed. Please call customer service.</p>

Error Group	Error No	Description	Error Handling
	7	Position error: Synchronization error	The position and speed of the DSL encoder must be assumed to be wrong after this error has been indicated. A safe state must be entered by the drive system. Check encoder/motor assignment. If error persists, a general hardware or mechanic error must be assumed. Please call customer service.
1	0	Single-turn error	The position of the DSL encoder must be assumed to be wrong after this error has been indicated. A safe state must be entered by the drive system if a safe position function is being used. Restart of encoder Check encoder/motor assignment. If error persists, a general hardware or mechanic error must be assumed. Please call customer service.
	1	Multi-turn error gear stage 1	The position of the DSL encoder must be assumed to be wrong after this error has been indicated. A safe state must be entered by the drive system if a safe position function is being used. Restart of encoder Check encoder/motor assignment. If error persists, a general hardware or mechanic error must be assumed. Please call customer service.
	2	Multi-turn error gear stage 2	The position of the DSL encoder must be assumed to be wrong after this error has been indicated. A safe state must be entered by the drive system if a safe position function is being used. Restart of encoder Check encoder/motor assignment. If error persists, a general hardware or mechanic error must be assumed. Please call customer service.
	3	Multi-turn error gear stage 3	The position of the DSL encoder must be assumed to be wrong after this error has been indicated. A safe state must be entered by the drive system if a safe position function is being used. Restart of encoder Check encoder/motor assignment. If error persists, a general hardware or mechanic error must be assumed. Please call customer service.

Error Group	Error No	Description	Error Handling
	5	Synchronization mismatch	<p>The position of the DSL encoder must be assumed to be wrong after this error has been indicated.</p> <p>A safe state must be entered by the drive system if a safe position function is being used.</p> <p>Restart of encoder Check encoder/motor assignment. If error persists, a general hardware or mechanic error must be assumed. Please call customer service.</p>
2	0	Wrong calibration data	<p>The position and speed of the DSL encoder must be assumed to be wrong after this error has been indicated.</p> <p>A safe state must be entered by the drive system.</p> <p>Restart of encoder If error persists, a memory error or general hardware fault has to be assumed. Please call customer service.</p>
	1	Wrong internal angle offset	<p>The position and speed of the DSL encoder must be assumed to be wrong after this error has been indicated.</p> <p>A safe state must be entered by the drive system.</p> <p>Restart of encoder If error persists, a memory error or general hardware fault has to be assumed. Please call customer service.</p>
	3	Internal checksum error	<p>The position and speed of the DSL encoder must be assumed to be wrong after this error has been indicated.</p> <p>A safe state must be entered by the drive system.</p> <p>Restart of encoder If error persists, a memory error or general hardware fault has to be assumed. Please call customer service.</p>
	4	Internal processor error	<p>The position and speed of the DSL encoder must be assumed to be wrong after this error has been indicated.</p> <p>A safe state must be entered by the drive system.</p> <p>Restart of encoder If error persists, a memory error or general hardware fault has to be assumed. Please call customer service.</p>

Error Group	Error No	Description	Error Handling
	5	Internal communication error	The position and speed of the DSL encoder must be assumed to be wrong after this error has been indicated. A safe state must be entered by the drive system. Restart of encoder If error persists, a memory error or general hardware fault has to be assumed. Please call customer service.
	6	Sensor error	The position and speed of the DSL encoder must be assumed to be wrong after this error has been indicated. A safe state must be entered by the drive system. Restart of encoder If error persists, a memory error or general hardware fault has to be assumed. Please call customer service.
	7	Memory error	The position and speed of the DSL encoder must be assumed to be wrong after this error has been indicated. A safe state must be entered by the drive system. Restart of encoder If error persists, a memory error or general hardware fault has to be assumed. Please call customer service.
3	0	Critical temperature	Cool down/warm up encoder Check mounting environment of encoder
	1	Critical LED current	LED of encoder defect or fatigued Internal mechanical damage Please call customer service
	2	Critical supply voltage	Hardware fault of encoder. Please call customer service.
	3	Critical speed	Data sheet supplied shaft speed range exceeded. Check application boundaries.
	4	Critical acceleration	Data sheet supplied shaft acceleration range exceeded. Check application boundaries.
4	5	Counter overflow	Reset counter
	0	Invalid resource access argument	Check programming of drive application
	1	Resource access denied	Check programming of drive application Set correct access code
	2	Internal resource access error	Restart of encoder If error persists, a memory error or general hardware fault has to be assumed. Please call customer service.
	3	File access error	Check programming of drive application

Table 36. Error messages of the motor-feedback system.

6.7. Parameter Channel

The parameter channel of Hiperface DSL® allows accessing parameters of the motor-feedback system.

The parameter channel distinguishes between two separate areas of data with separate access mechanisms:

- Interface information is queried via “Short Messages“.
- Motor-feedback system information is queried via “Long Messages“.

6.7.1. Short Message

Remote (DSL motor-feedback system) registers indicating interface information are mirrored as register addresses 40h to 7Fh within the DSL Master. These remote registers are addressed like DSL Master registers.

Since the transmission of the remote register values is carried out over the Parameter Channel and therefore over the DSL line the delay between request and response for “Short message” transactions varies between systems with different connection lines. Unlike for DSL Master registers it is necessary that the drive application waits for the response to arrive.

While the addressing of remote registers is carried out like with DSL Master registers, the response will be placed in a special DSL Master register (**PC_DATA**, 2Fh).

The direct response value received on SPI1 MISO during reading or writing is a dummy value.

In the DSL Master register **EVENT_L**, **FRES** shows if the “Short message“ channel is busy or if the response has arrived in the DSL Master. **FRES** can be evaluated with every SPI1 operation since it is part of every SPI1 transmission (bit 0 in **MASTER STATUS L**, see chapter 5.2).

The parameter channel can only carry one “Short message“ at a time. Multiple remote registers can only be queried sequentially, i.e. after the previous response has arrived.



Please note that a “Short message“ can be initiated during an ongoing “Long message” transaction (see chapter 6.7.2) and vice versa.

The following Figure 31 shows an example of reading the remote register **ENC_STO** (40h).

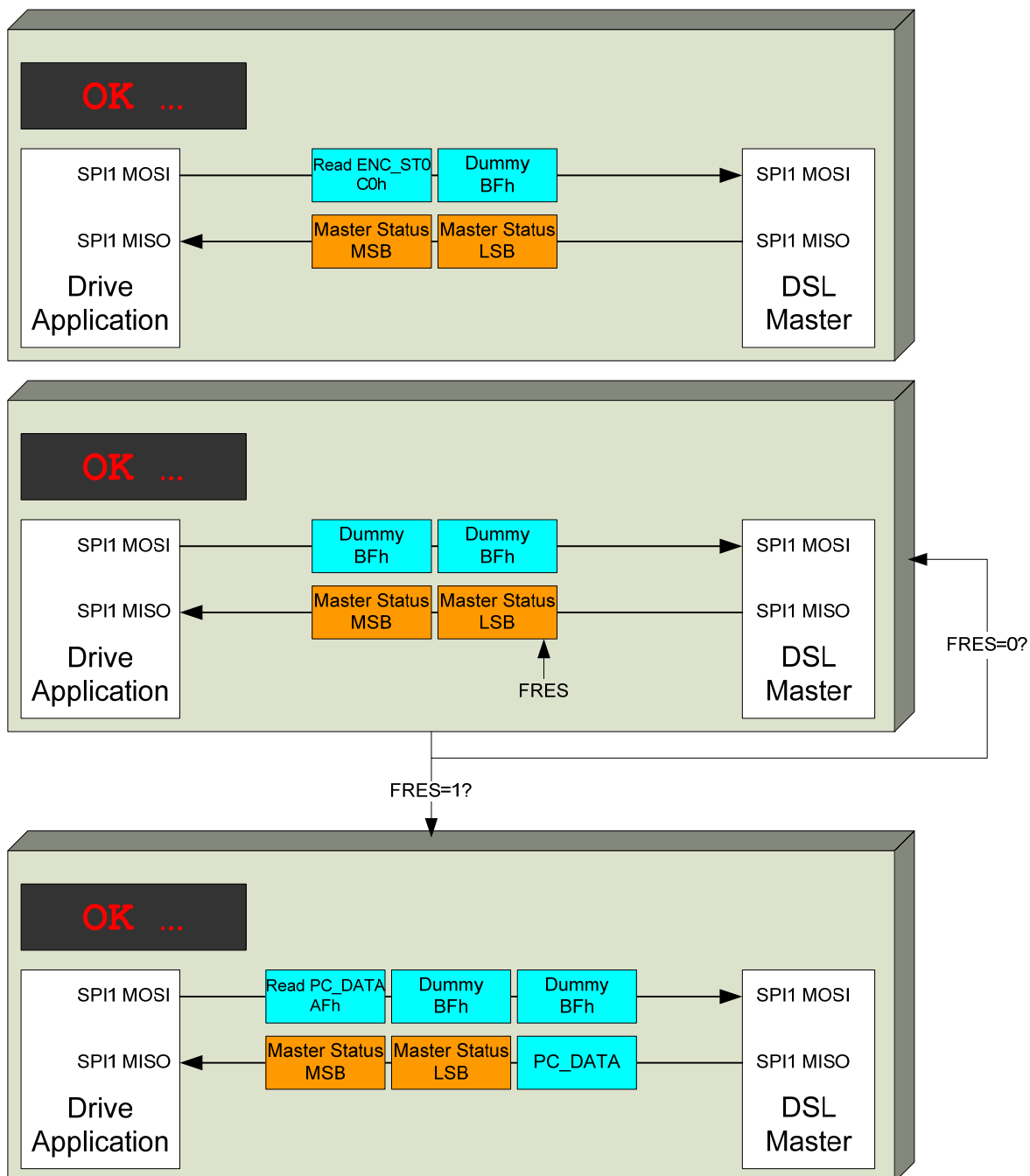


Figure 31. Reading Remote Registers.

6.7.2. Long Message

Access to motor-feedback system resources apart from interface registers (see chapter 5.4) is conducted via “Long message” transactions on the parameter channel.

The organization and scope of resources depends on the individual implementation of the DSL Slave and DSL encoder.

A “Long message” is initiated 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 – if any – is placed in registers **PC_BUFFER0:7**.

During execution of the long message transaction **FREL** in register **EVENT_L** is cleared. When the transaction is finished **FREL** is set again.



Please note that a “Long message” can be initiated during an ongoing “Short message” transaction (see chapter 6.7.1) and vice versa.

A “Long Message“ allows exchange of general parameter data between drive and motor-feedback system. These parameters can contain information on the motor-feedback system status, control data for the motor-feedback system or user-defined data.

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

In chapter 7 usual resources are listed for DSL encoders. Actually implemented resources are specified in the data sheet of individual DSL encoders.

A “Long Message” is initiated by setting the corresponding DSL Master registers **PC_BUFFER**, **PC_ADD**, **PC_OFF**, and **PC_CTRL** (20h – 2Ch).

During processing of a “Long Message” by the motor-feedback system the flag **FREL** in the event register **EVENT_L** (05h) is cleared. After completion this flag will be set again to indicate readiness for a new “Long Message”.

After indication of a set **FREL** data returned by a read access can be queried in the registers **PC_BUFFER** (see chapter 5.3.16).



Please note that only one “Long Message” can be processed at a time. Access to resources with more than 8 bytes of data must be conducted through subsequent “Long Messages”.

A “Long Message“ is defined through several qualifiers placed in the registers named above. The following Figure 32 gives an overview of these qualifiers.

Value of LEN	Data length	Used DATA registers
0 (00b)	0 bytes	No data transmission
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

Table 38. DATA register ranges.

If the motor-feedback system encounters an error during resource access this is indicated as an error message (see chapter 6.6.4). Additionally the flag **ERR** will be set, the qualifier **LEN** will be set to 2 bytes (01B) and the **DATA** registers **PC_BUFFER0** and **PC_BUFFER1** will contain an error code.

This error code allows a detailed error tracking of a “Long Message” transaction.

The following lists error codes and their meaning.



Please note that the value of register **PC_BUFFER1** corresponds to the error code in the encoder status register **ENC_ST** (see chapter 5.5.1).

PC_BUFFER1	PC_BUFFER0	Meaning of error code
40h	10h	Resource address not implemented in encoder
	11h	Wrong length indicated for resource access
	12h	Wrong length indicated for direct resource access
	13h	Offset address too high
	14h	Invalid offset address
	15h	Invalid “Long Message” qualifier
	16h	Missing offset value
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 corrupt
	11h	Time-out during resource access
	12h	Internal processing error during resource access
43h	10h	File number not found
	11h	File name not found
	12h	Invalid address for file access
	13h	File size not changeable
	14h	File storage space full
	15h	File allocation table corrupt
	16h	No file loaded for action
17h	File with same name exists	

Table 39. “Long Message” error codes.

The “Long Message” qualifier **R/W** determines if a read or write access is programmed.

Value of R/W	“Long Message“ direction
0	Write
1	Read

Table 40. Value of R/W for "Long Message".

If a write access is programmed the data to be transmitted must be placed in the qualifier **DATA**.

The “Long Message” qualifier **O/N** determines whether the message is transmitted with an offset address or without.

Value of O/N	“Long Message” mode
0	No offset addressing
1	Offset addressing

Table 41. Value of O/N for "Long Message".

The resource description in chapter 7 details for which purpose offset addressing is used. Offset addressing allows transmission of an additional “Long Message” parameter to the motor-feedback system apart from address (**ADD**) and the message data proper (**DATA**).

If the qualifier **O/N** is set to ‘1’ the qualifier **OFF ADD** must contain the value for the offset address qualifier.

The same bit **O/N** in the register **PC_ADD_H** can be read out after reception of the “Long Message” response to determine the error qualifier **ERR**.

Value of ERR	Error during resource access
0	No error
1	Error encountered

Table 42. Value of ERR for "Long Message".

ERR is set if processing of the “Long Message” led to an error in the motor-feedback system.

D/I determines whether a “Long Message” uses direct or indirect addressing.

Value of D/I	“Long Message” addressing
0	Direct addressing
1	Indirect addressing

Table 43. Value of D/I for "Long Message".

The resource description in chapter 7 details for which purpose direct or indirect addressing is used.

The qualifier **LEN** sets the data length of the “Long Message”. Table 38 specifies the usage of this qualifier.

LEN must conform to the allowed values of the addressed resource (see chapter 7). Failure to do so will lead to a termination of the “Long Message” within the motor-feedback system and indication of a corresponding error message.

The qualifier **ADD** determines the targeted resource of the “Long Message”. The value of **ADD** corresponds to the resource index RID.

Value of ADD	Resource index (RID)
000h – 3FFh	000h – 3FFh

Table 44. Value of ADD for "Long Message".

Access to resources not implemented in the motor-feedback system will be terminated with a corresponding error message.

HIPERFACE DSL®

The “Long Message” qualifier **OFF ADD** contains the offset address if offset addressing is used (see above on qualifier **O/N**). The resource description in chapter 7 details the allowed range and purpose for each individual resource.

Value of OFF ADD	Used registers
0000h – 7FFFh	PCR_ADD_H/PCR_ADD_L

Table 45. Value of OFF ADD for "Long Message".

Access to a resource with an invalid or too high **OFF ADD** value will lead to a termination of the “Long Message” within the motor-feedback system and indication of a corresponding error message.

The following Figure 33 shows an exemplary “Long Message” read command.

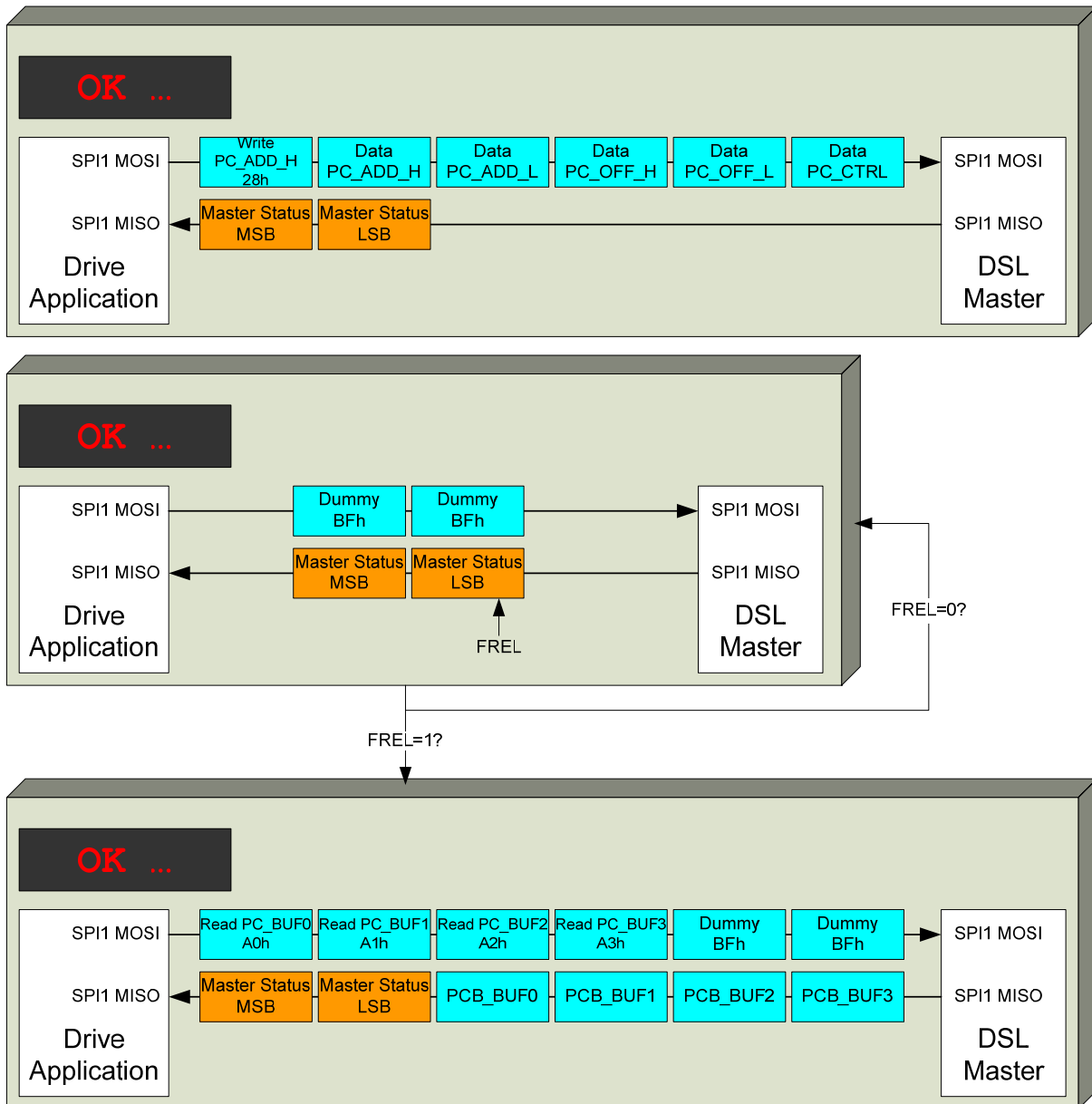


Figure 33. Example: "Long Message" read command.

6.7.3. Error Handling Parameter Channel

Error handling of a “Short Message” and a “Long Message” differs.

If a “Short Message” to the motor-feedback system is transmitted with an error the protocol will automatically retransmit the message until an acknowledgment for correct transmission is received. This is not indicated to the drive explicitly. The flag **FRES** will remain cleared until the response to the “Short Message” is received correctly.

If the DSL Master receives no acknowledgment to the transmission of a “Short Message” the protocol automatically starts cyclic retransmissions. This time-out of a “Short Message” will be indicated by the flag **RETS** in the event register **EVENT_L** and in the online DSL Master status. **RETS** will only be cleared automatically after the response to the “Short Message” was received correctly.

If a “Long Message” to the motor-feedback system is transmitted with an error this will be indicated by the flag **ANS** in the event register **EVENT_L**. The “Long Message” will not be retransmitted automatically..

If the DSL Master receives no acknowledgment to the transmission of a “Long Message” the protocol automatically starts cyclic retransmissions. This time-out of a “Long Message” will be indicated by the flag **RETL** in the event register **EVENT_L** and in the online DSL Master status. **RETL** will only be cleared automatically after the acknowledgment to the transmission of the “Long Message” was received correctly.

If a “Long Message” was transmitted correctly to the motor-feedback system but the response to the drive application is missing or too slow there will be no automatic retries by the DSL Master. This is necessary since the processing time of a “Long Message” is not predictable. To keep track of the time-out of an ongoing “Long Message”, the drive application can resort to the time-out qualifier of each DSL motor-feedback system resource (see chapter 7).

In order to use the parameter channel again even though an unanswered “Short Message” or “Long Message” blocks the respective message channel, the drive application has to trigger a parameter channel reset.

This reset has no influence on position measurement or position data transmission.

The reset sequence for the parameter channel is shown in Figure 34.

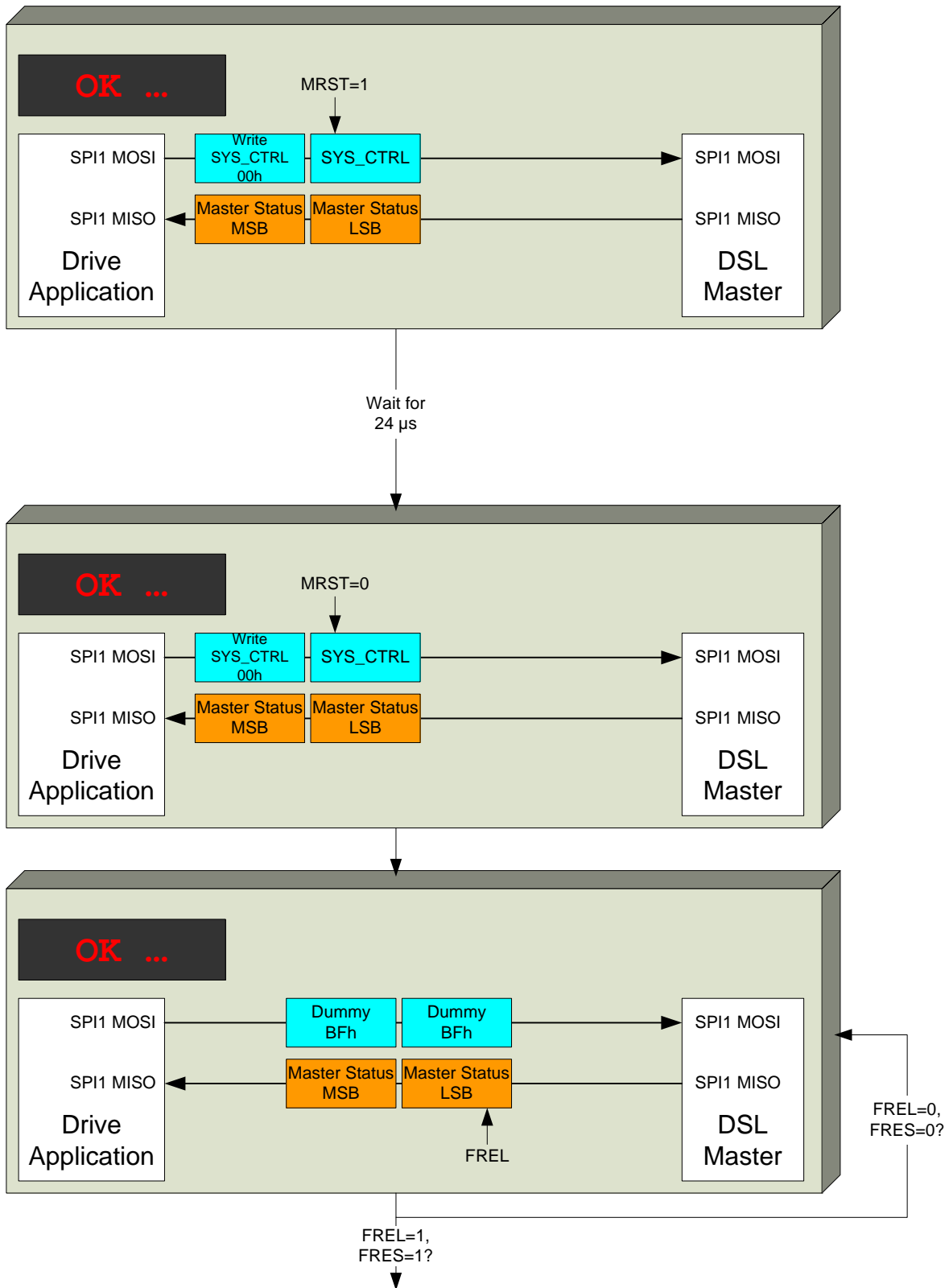


Figure 34. Reset of parameter channel.

7. Resources Motor-feedback System

Resources of a DSL motor-feedback system represent most functions of the sensor.

“Long Message” transactions allow access to all resources implemented within a DSL motor-feedback system. Examples for resources are encoder identification values, function and failure monitoring, sensor administration, or user-defined data storage.



Please note that process values of the motor-feedback system, i.e. position and speed values, have separate access mechanisms (see chapters 6.3, 6.4).

The resources implemented within a DSL motor-feedback system are made accessible through the Resource DataBase (RDB). A “Long Message” always targets a single RDB entry.

The resources listed in this chapter show usual functions of a DSL motor-feedback system. Actually implemented resources of individual DSL motor-feedback systems are specified in their respective data sheets.

7.1. Resource Access

Resources of a DSL motor-feedback system can be accessed in two ways. Additionally, this chapter describes how resource definition values can be read out by the so-called “Direct Access”.

7.1.1. Access-by-Address

Each resource is defined by a unique resource index (RID).

A “Long Message” can target the associated resource by using the RID as its address qualifier (see chapter 6.7.2).

If the resource is accessed with a “Direct Access” the resource definition values will be returned. See chapter 7.1.3.

7.1.2. Access-by-Tree

The Resource DataBase (RDB) is structured in the form of a tree. This allows accessing a resource by reference, starting at a root resource pointing to other resources. Figure 35 shows a representation of this tree structure.

Starting at the “root node” resource with the resource index RID=000h a read access will return the address of attached nodes. Traversing through further nodes recursively allows access to any tier of the tree.

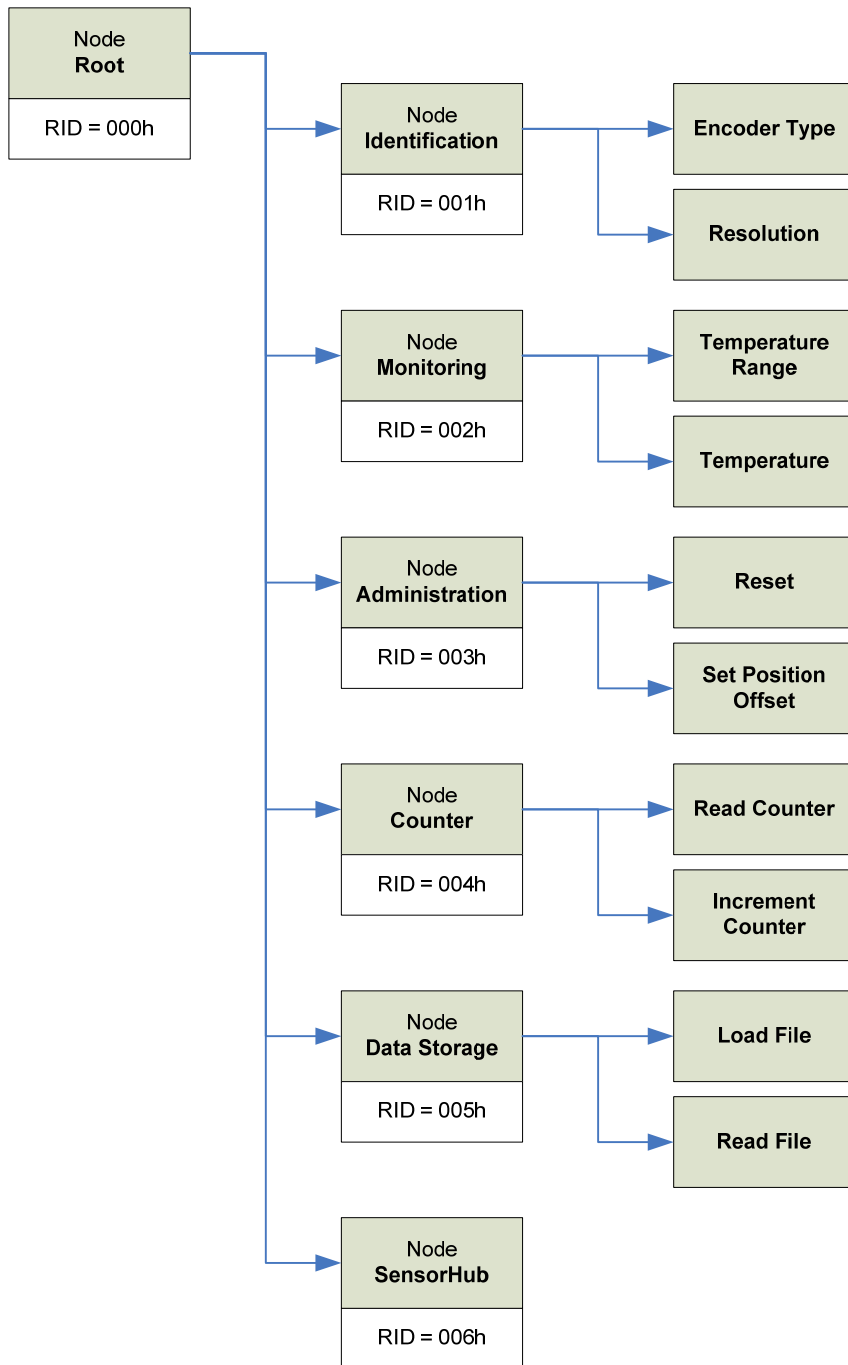


Figure 35. Tree structure of Resource DataBase.

The qualifiers of a “Long Message” for reading the address of an attached node are listed in Table 46.

Qualifier	Value	Description
DATA	-	-
R/W	1	Read
O/N	1	Offset
D/I	1	Indirekt
LEN	1	2 bytes
ADD	Variable	Calling node
OFF ADD	Variable	Ordinal number of attached node

Table 46. Parameters for node access .

The value stored in **OFF ADD** indicates the ordinal number of the attached node whose resource identifier is to be returned. This ordinal number is indicated in the following list of all resources.

The result of this “Long Message” transaction is the resource index (RID) of the requested resource.

7.1.3. Direct Access

Using “Direct Access” (see chapter 6.7.2) the definition values of any resource can be read out by the user.

These definition values consist of a clear-text name (max. 8 characters) describing the resource, the resource size, access rights, and resource access time-out value.

The desired definition value is chosen by the user by setting an appropriate offset value.

The following Table 47 lists the possible access modes (direct, indirect) and associated resource definition values.

Resource Type (see chapter 7.2)	Access	Offset Value	Data	Note
Node	Direct Read	0/none	Resource Name	e.g. “ROOT”
		1	Resource Size	e.g. 05h for 5 sub-entries
		2	Resource Read Access	e.g. 0 for access level 0
		3	Resource Write Access	e.g. 2 for access level 2
		4	Resource Time-out	e.g. 0Ah for 10 ms
	Indirect Read	0	RID of sub-entry 1	e.g. 001h
		...		
All others	Direct Read	0/none	Resource Name	e.g. “ENCTYPE”
		1	Resource Size	e.g. 02h for 2 bytes
		2	Resource Read Access	e.g. 0 for access level 0
		3	Resource Write Access	e.g. 2 for access level 2
		4	Resource Time-out	
	Indirect Read/Write	Any	Resource Value	See chapter 7.2

Table 47. Resource access modes.

7.2. Resource List

This chapter lists all resources implemented in a DSL motor-feedback system.



Please note that position and speed values of the motor-feedback system are process values and are accessed differently to general resources (see chapters 6.3, 6.4).

All resources are specified with the “Long Message” qualifiers valid for access (see chapter 6.7.2).

Also, the Resource DataBase (RDB) definitions are detailed for each resource. These definitions specify the following resource characteristics:

RDB definition	Data range	Description
RID	0 – 1023 000h – 3FFh	Resource index: Used as address qualifier in a “Long Message”
Size	0 – 32767 0 – 7FFFh	Length of resource data in bytes. Defines range usable by offset-based access in a “Long Message”. For nodes: Number of attached nodes. Offset address may be in the range between 0 and Size-1.
R	0 1 2 3 4 15	Read access: Reading possible for anyone Access level “Operator” required for reading Access level “Maintenance” required for reading Access level “Authorized Client” required for reading Access level “Service” required for reading No read access possible
W	0 1 2 3 4 15	Write access: Writing possible for anyone Access level “Operator” required for writing Access level “Maintenance” required for writing Access level “Authorized Client” required for writing Access level “Service” required for writing No write access possible
Timeout	0 – 254 255	Time-out of resource access in msec. If the DSL system does not respond to a “Long Message” within this time, a processing error has to be assumed (see chapter 6.7.3) The resource needs more than 254 msec for processing or has a non-deterministic time-out.

Table 48. Resource DataBase definitions.

7.2.1. Nodes

All resources of a DSL motor-feedback system are logically structured in a tree (see chapter 7.1.2). This arrangement is structured through node resources.

A read access to a node returns the address of an attached resource.

The following table shows an overview over all node resource definitions (explanation of the columns see Table 48).

Resource	RID	Size	R	W	Timeout
Root Node	000h	5	0	15	10
Identification Node	001h	*	0	15	10
Monitoring Node	002h	*	0	15	10
Administration Node	003h	*	0	15	10
Counter Node	004h	*	0	15	10
Data Storage Node	005h	*	0	15	10

Table 49. Identification resource definitions.

* The size of these nodes depends on the number of actually implemented resources in an individual DSL encoder.

7.2.1.1. Root Node

The root node is the top most resource of the tree structure placed at address (RID) 000h.

From the root node all nodes can be accessed representing different resource groups.

A direct read access to the root node returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"ROOT"
Resource Size	1	5
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	10

Table 50. Root node definition values.

An indirect read access to the root node returns the Resource DataBase definition of the n th attached node where n is specified as offset address.

The offset address must fall in the range between 0 and Size-1.

7.2.1.2. Identification Node

The identification node contains pointers to all resources related to motor-feedback system identification ("Electronic Type Label").

A direct read access to the identification node returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"IDENT"
Resource Size	1	Depends on implemented resources
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	10

Table 51. Identification node definition values.

HIPERFACE DSL®

An indirect read access to the identification node returns the Resource DataBase definition of the n th attached node where n is specified as offset address. Each attached node is an actual resource.

The offset address must fall in the range between 0 and Size-1.

7.2.1.3. Monitoring Node

The monitoring node contains pointers to all resources related to motor-feedback system monitoring (e.g. temperature check).

A direct read access to the monitoring node returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"MONITOR"
Resource Size	1	Depends on implemented resources
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	10

Table 52. Monitoring node definition values.

An indirect read access to the monitoring node returns the Resource DataBase definition of the n th attached node where n is specified as offset address. Each attached node is an actual resource.

The offset address must fall in the range between 0 and Size-1.

7.2.1.4. Administration Node

The administration node contains pointers to all resources related to motor-feedback system administration (e.g. reset, set access level).

A direct read access to the administration node returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"ADMIN"
Resource Size	1	Depends on implemented resources
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	10

Table 53. Administration node definition values.

An indirect read access to the administration node returns the Resource DataBase definition of the n th attached node where n is specified as offset address. Each attached node is an actual resource.

The offset address must fall in the range between 0 and Size-1.

7.2.1.5. Counter Node

The counter node contains pointers to all resources related to the user-defined counter.

A direct read access to the counter node returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"COUNTER"
Resource Size	1	Depends on implemented resources
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	10

Table 54. Counter node definition values.

An indirect read access to the counter node returns the Resource DataBase definition of the n th attached node where n is specified as offset address. Each attached node is an actual resource.

The offset address must fall in the range between 0 and Size-1.

7.2.1.6. Data Storage Node

The data storage node contains pointers to all resources related to user-defined data storage.

A direct read access to the data storage node returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"DATA"
Resource Size	1	Depends on implemented resources
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	10

Table 55. Data storage node definition values.

An indirect read access to the data storage node returns the Resource DataBase definition of the n th attached node where n is specified as offset address. Each attached node is an actual resource.

The offset address must fall in the range between 0 and Size-1.

7.2.2.2. Resolution

The resolution value defines the amount of steps per single turn of the encoder (Rotary Encoder) or length of one measuring step in multiples of 1 nm (Linear Encoder).

A direct read access to Resolution returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"RESOLUTN"
Resource Size	1	4
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	5

Table 59. Resolution definition values.

The resolution value is output as a 32 bit unsigned value.

Offset based access is not useful for this resource as the size of the resource data is less than the maximum for one "Long Message" transaction.

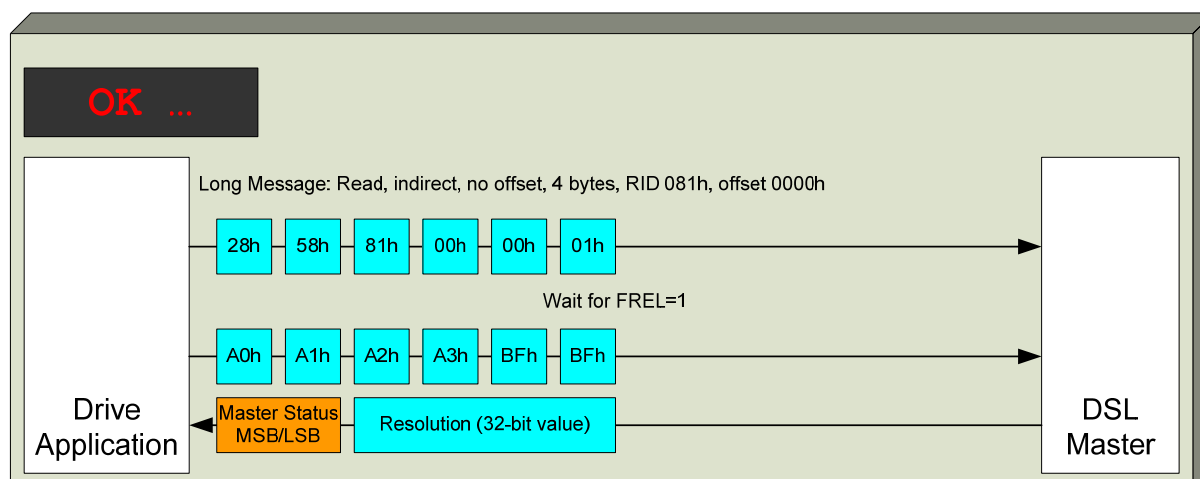


Figure 37. Read Resolution.

7.2.2.3. Range

The range value defines the amount of coded revolutions of the encoder (Rotary Encoder) or the coded measuring range in multiples of measuring steps (Linear Encoder).

A direct read access to Range returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"RANGE"
Resource Size	1	4
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	5

Table 60. Range definition values.

The range value is output as a 32 bit unsigned value.

Offset based access is not useful for this resource as the size of the resource data is less than the maximum for one "Long Message" transaction.

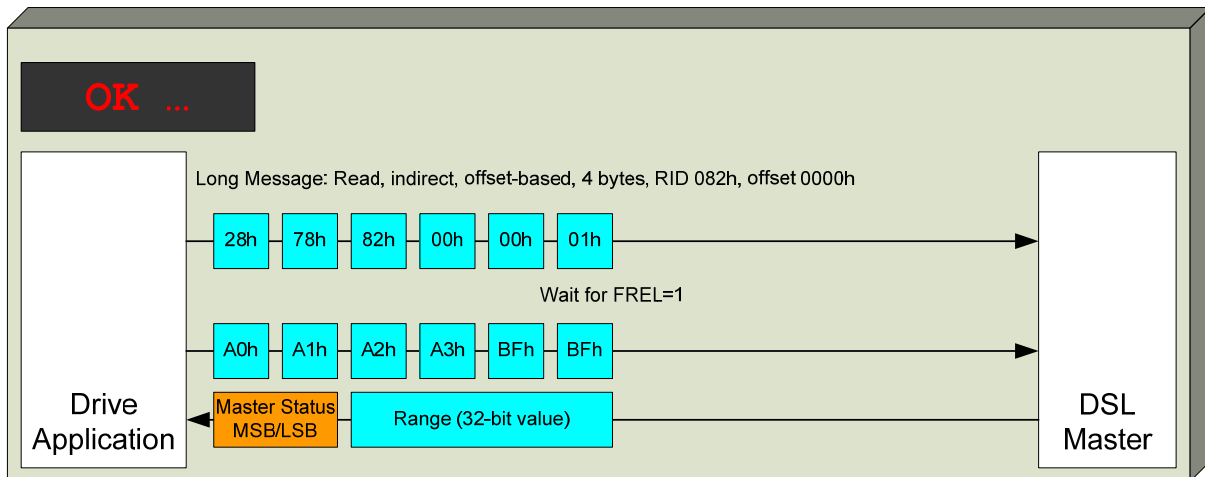


Figure 38. Read Range.

7.2.2.4. Type Code Name

This resource indicates the type code name of the encoder. The name is stored with a maximum of 18 characters in ASCII format. Unused characters are stored with the ASCII code 00h.

A direct read access to Type Code Name returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"TYPECODE"
Resource Size	1	18
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	5

Table 61. Type Code Name definition values.

Please note that reading of the full type code name makes the use of up to three "Long Message" transactions necessary since one "Long Message" can only carry 8 bytes of data.

With offset based access the qualifier OFF ADD indicates the first character of the type code name that should be transmitted in the "Long Message".

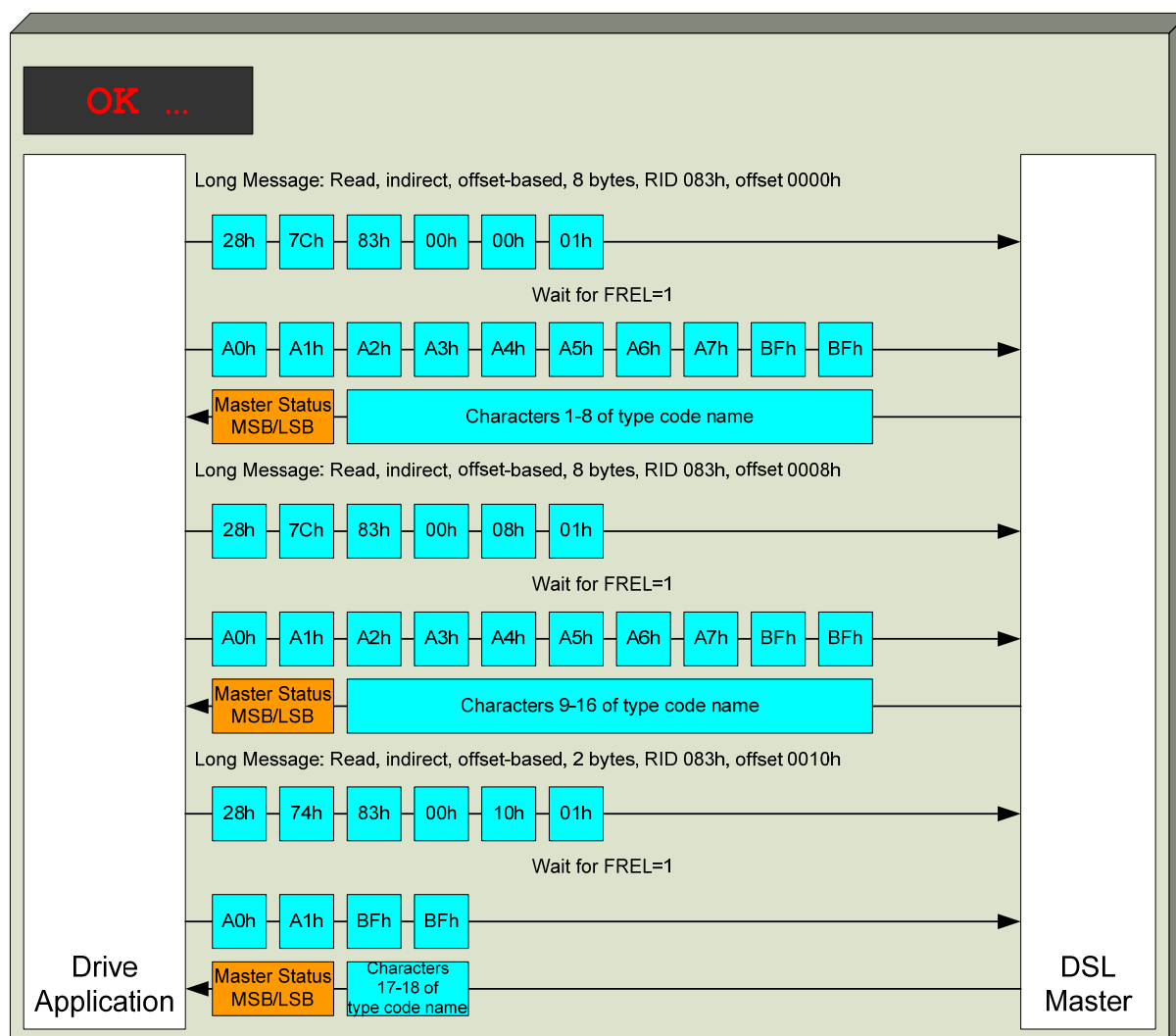


Figure 39. Read Type Code Name.

7.2.2.5. Serial No

This resource indicates the serial number of the encoder. The serial number is stored with a maximum of 10 characters in ASCII format. Unused characters are stored with the ASCII code 00h.

A direct read access to Resolution returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"SERIALNO"
Resource Size	1	10
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	5

Table 62. Serial No definition values.

Please note that reading of the full serial number name makes the use of up to two "Long Message" transactions necessary since one "Long Message" can only carry 8 bytes of data.

With offset based access the qualifier OFF ADD indicates the first character of the serial number that should be transmitted in the "Long Message".

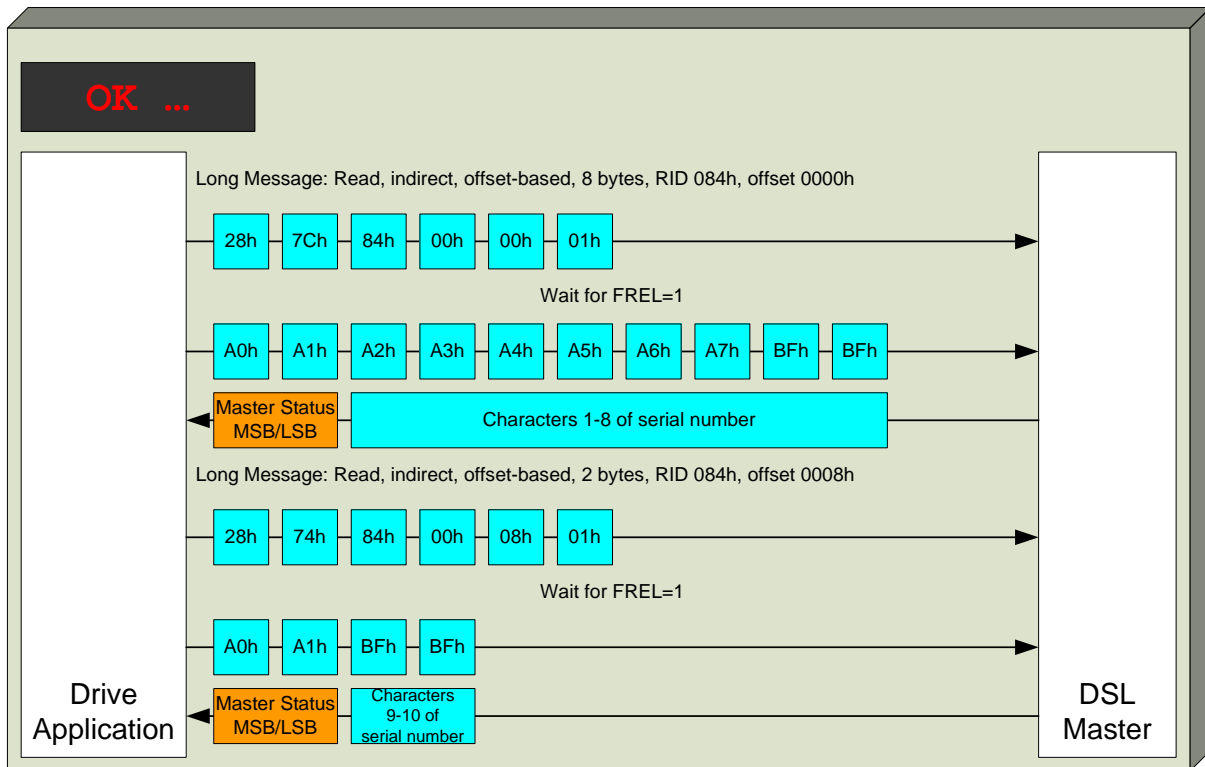


Figure 40. Read Serial No.

7.2.2.6. Firmware Revision No

This resource indicates the firmware revision number of the encoder. The firmware revision number is stored with a maximum of 20 characters in ASCII format. Unused characters are stored with the ASCII code 00h.

A direct read access to Resolution returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"FWREVNO"
Resource Size	1	20
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	5

Table 63. Firmware Revision No definition values.

Please note that reading of the full firmware revision number makes the use of up to three "Long Message" transactions necessary since one "Long Message" can only carry 8 bytes of data.

With offset based access the qualifier OFF ADD indicates the first character of the firmware revision number that should be transmitted in the "Long Message".

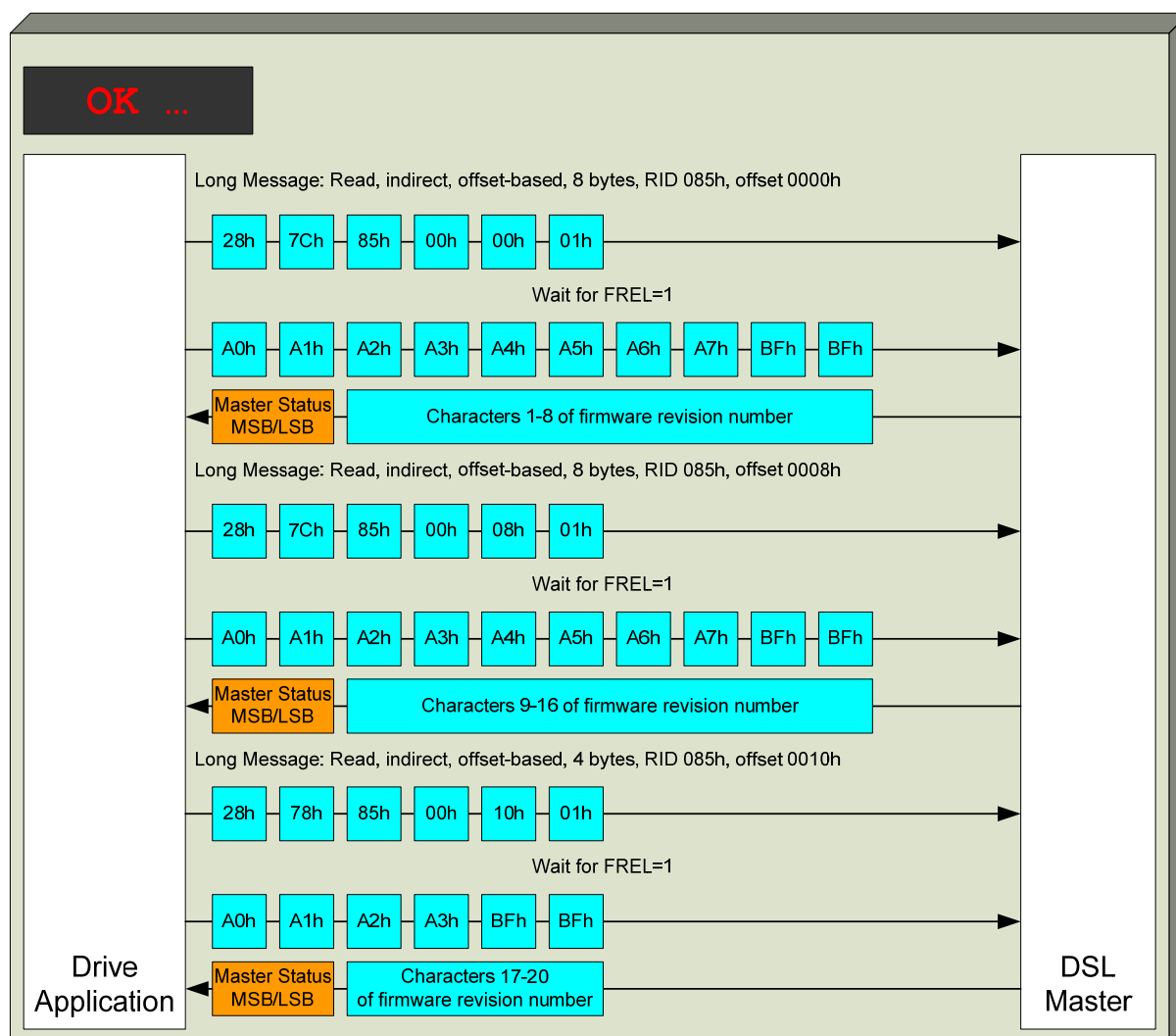


Figure 41. Read Firmware Revision No.

7.2.2.7. Firmware Date

This resource indicates the firmware date of the encoder. The firmware date is stored in 8 characters in ASCII format.

A direct read access to Firmware Date returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"FWDATE"
Resource Size	1	8
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	5

Table 64. Firmware Date definition values.

The firmware date is returned in the following format:

Byte	Value	Description
7/6	'00'...'99'	Year of firmware date meaning 20yy
5	'.'	Decimal point separator
4/3	'01'...'12'	Month of firmware date
2	'.'	Decimal point separator
1/0	'01'...'31'	Day of firmware date

Table 65. Firmware Date definition.

Offset based access is not useful for this resource as the resource data is readable with one “Long Message” transaction.

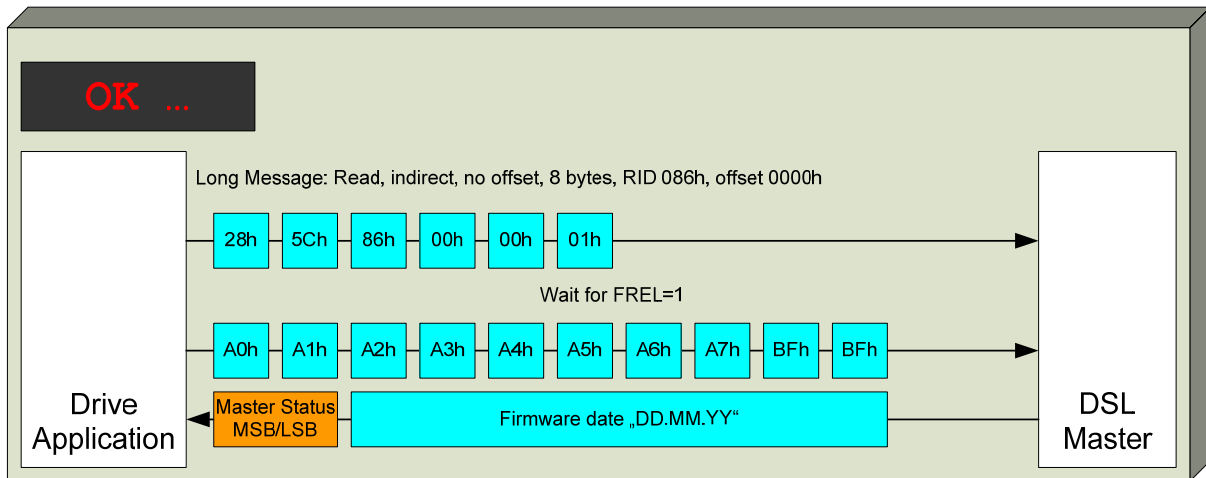


Figure 42. Read Firmware Date.

7.2.2.8. EEPROM Size

This resource indicates the total amount of non-volatile memory in the encoder available for storage of user data. The EEPROM size is returned as an unsigned 16 bit value representing the number of bytes.

A direct read access to EEPROM Size returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	“EESIZE”
Resource Size	1	2
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	5

Table 66. EEPROM Size definition values.

Offset based access is not useful for this resource as the resource data is readable with one “Long Message” transaction.

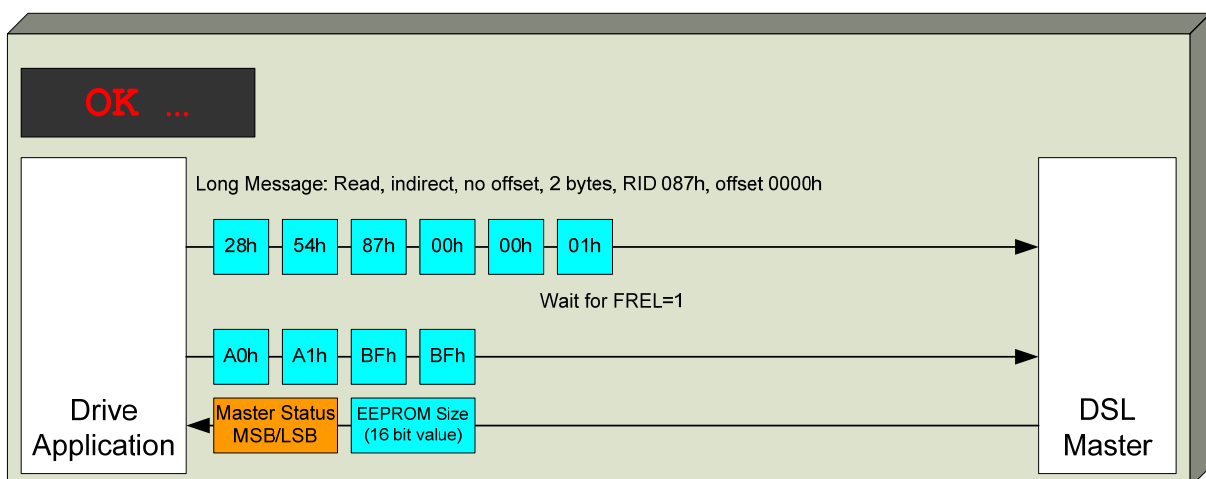


Figure 43. Read EEPROM Size.

7.2.3. Monitoring Resources

The monitoring resources of the DSL motor-feedback system indicate current environment values and their boundaries as well as usage statistics and an error stack.

The following table shows an overview over all identification resource definitions (explanation of the columns see Table 48).

Resource	RID	Size	R	W	Timeout
Temperature Range	0C0h	4	0	15	5
Temperature	0C1h	2	0	15	5
LED Current Range	0C2h	4	0	15	5
LED Current	0C3h	2	0	15	5
Supply Voltage Range	0C4h	4	0	15	5
Supply Voltage	0C5h	2	0	15	5
Speed Range	0C6h	2	0	15	5
Speed	0C7h	2	0	15	5
Acceleration Range	0C8h	2	0	15	5
Acceleration	0C9h	2	0	15	5
Life-time	0CBh	8	0	15	5
Error Log	0CCh	16	0	15	30
Usage Histogram	0CDh	4	0	15	30

Table 67. Monitoring resource definitions.

7.2.3.1. Temperature Range

This resource indicates the minimum and maximum allowed temperature value of the DSL motor-feedback system as stated in the product data sheet.

A direct read access to Temperature Range returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"TEMPRNG"
Resource Size	1	4
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	5

Table 68. Temperature Range definition values.

The Temperature Range values are stored as 16 bit signed values with two's complement arithmetic. The unit of the temperature values is tenths of °C (0.1 °C).

Examples for Temperature Range values are:

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

Table 69. Temperature Range examples.

The Temperature Range values are returned in the following format:

Byte	Value	Description
3/2	-2730...10000	Maximum allowed temperature of encoder in 0.1°C
1/0	-2730...10000	Minimum allowed temperature of encoder in 0.1°C

Table 70. Temperature Range definition.

Offset based access can be used for returning only one of two temperature range values.

Offset value	Message Length	Return values
0000h	4	Minimum and maximum temperature range
0000h	2	Minimum temperature
0002h	2	Maximum temperature

Table 71. Temperature Range offset choices.

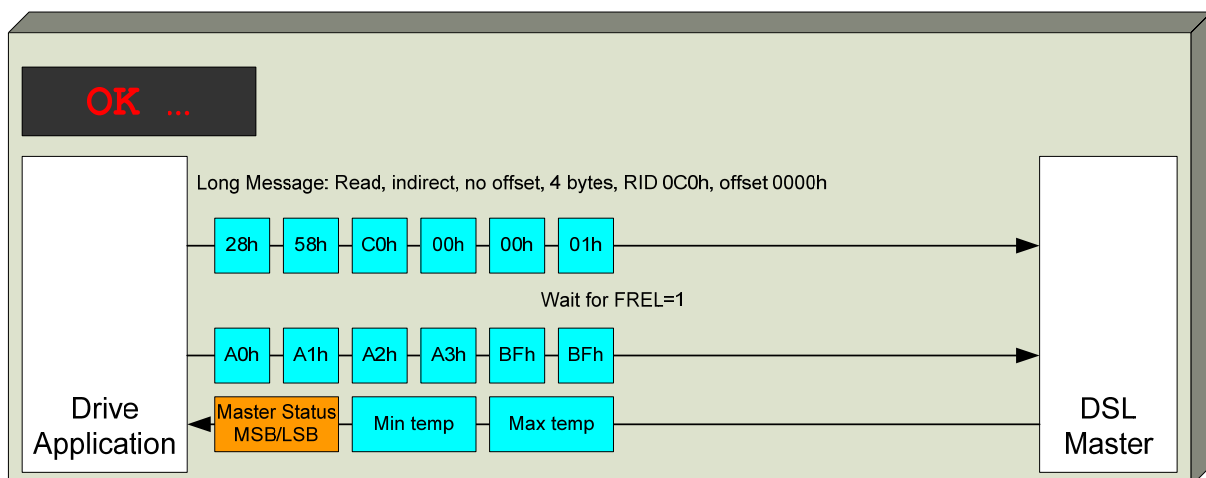


Figure 44. Read Temperature Range.

7.2.3.2. Temperature

This resource indicates a temperature measurement, either the current DSL motor-feedback system temperature or the value of an external temperature sensor.

In the case of the DSL motor-feedback system temperature, if the measured value lies beyond one of the boundaries an error will be indicated (see chapter 6.6.4, error group 3, error no 0).

A direct read access to Temperature returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"TEMPRTUR"
Resource Size	1	2
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	5

Table 72. Temperature definition values.

The temperature value is stored as a 16 bit signed value with two's complement arithmetic. The unit of the temperature value is tenths of °C (0.1 °C).

The temperature value is returned in the following format:

Byte	Value	Description
1/0	-2730...10000	Current temperature value in 0.1°C

Table 73. Temperature definition.

Offset based access is used for indicating the required temperature measurement.

Offset value	Message Length	Return values
0000h	2	Motor-feedback system temperature
0001h	2	External temperature sensor 1
0002h	2	External temperature sensor 2

Table 74. Temperature offset choices.

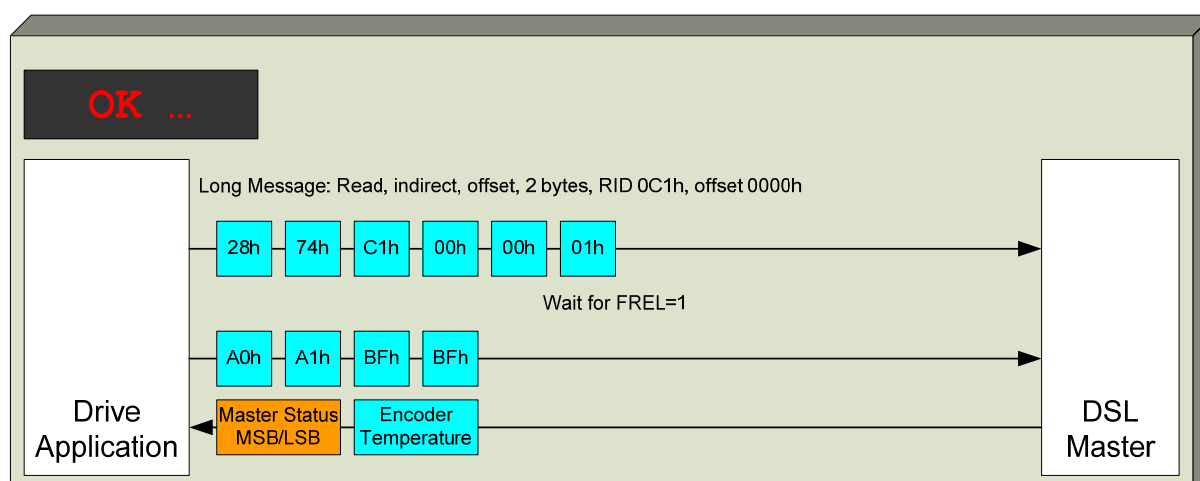


Figure 45. Read Temperature.

7.2.3.3. LED Current Range

This resource indicates the minimum and maximum allowed LED current values of optical DSL motor-feedback system as stated in the product data sheet.

A direct read access to LED Current Range returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"LEDRANGE"
Resource Size	1	4
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	5

Table 75. LED Current Range definition values.

The LED Current Range values are stored as 16 bit unsigned values. The unit of the LED current values is tenths of mA (0.1 mA).

Examples for LED current values are:

Current	Resource Value (bin)	Resource Value (hex)
5 mA	0000 0000 0011 0010b	0032h
20 mA	0000 0000 1100 1000b	00C8h

Table 76. LED Current Range examples.

The LED Current Range values are returned in the following format:

Byte	Value	Description
3/2	0...65535	Maximum allowed LED current of encoder in 0.1 mA
1/0	0...65535	Minimum allowed LED current of encoder in 0.1 mA

Table 77. LED Current Range definition.

Offset based access can be used for returning only one of two LED Current Range values.

Offset value	Message Length	Return values
0000h	4	Minimum and maximum LED current
0000h	2	Minimum LED current
0002h	2	Maximum LED current

Table 78. LED Current Range offset choices.

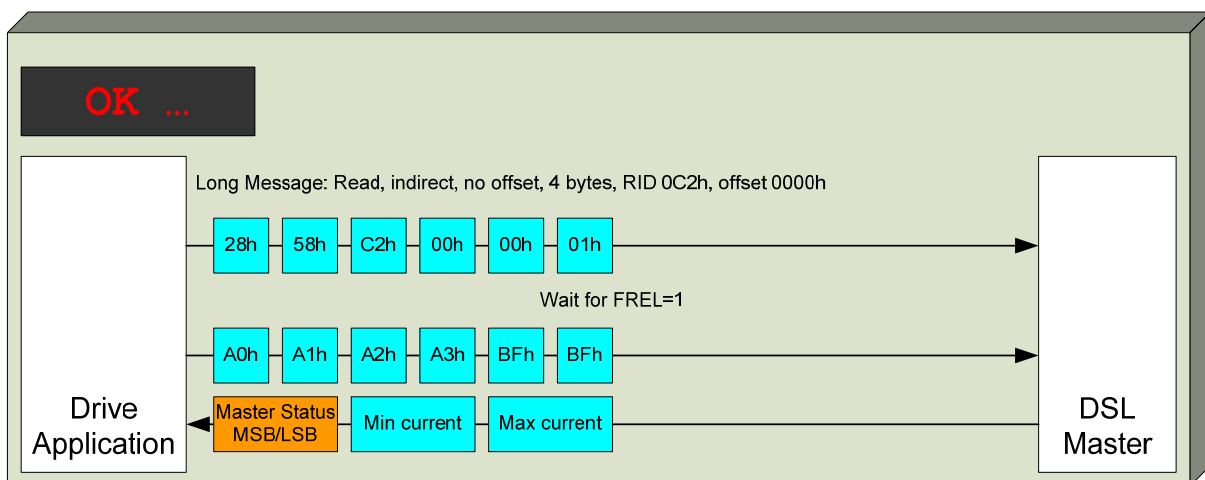


Figure 46. Read LED Current Range.

7.2.3.4. LED Current

This resource indicates the LED current of an optical DSL motor-feedback system.

If the measured LED Current lies near the maximum boundary the product could be at the end of its life-time.

If the measured LED Current lies beyond one of the boundaries an error will be indicated (see chapter 6.6.4, error group 3, error no 1).

A direct read access to LED Current returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"LEDCURR"
Resource Size	1	2
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	5

Table 79. LED Current definition values.

The LED Current value is stored as a 16 bit unsigned value. The unit of the LED Current value is tenths of mA (0.1 mA).

The LED Current value is returned in the following format:

Byte	Value	Description
1/0	0...65535	Current LED current of encoder in 0.1 mA

Table 80. LED Current definition.

Offset based access is not useful for this resource as the resource data is readable with one "Long Message" transaction.

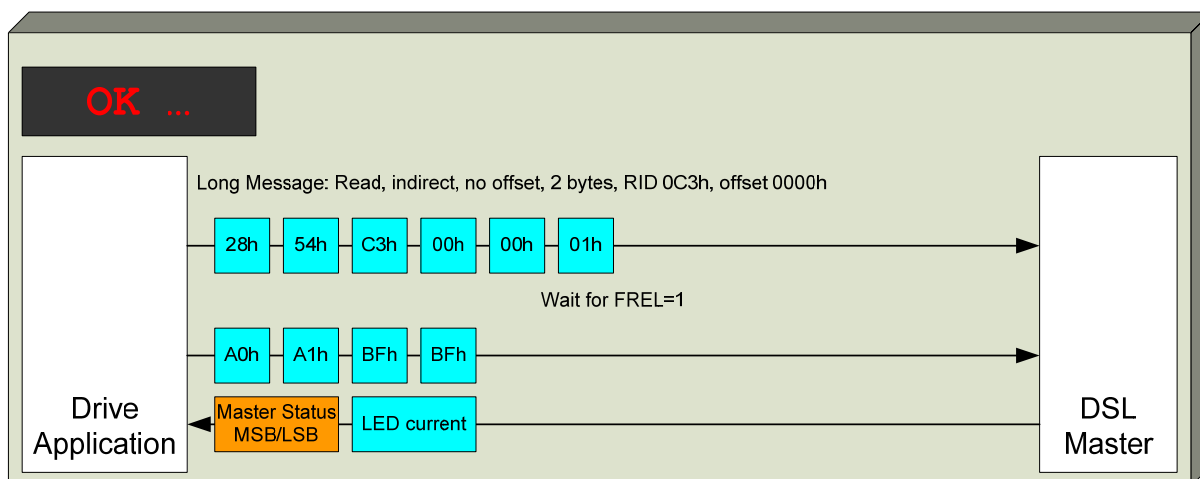


Figure 47. Read LED Current.

7.2.3.5. Supply Voltage Range

This resource indicates the minimum and maximum allowed supply voltage values of DSL motor-feedback system as stated in the product data sheet.

A direct read access to Supply Voltage Range returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"SUPRANGE"
Resource Size	1	4
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	5

Table 81. Supply Voltage Range definition values.

The Supply Voltage Range values are stored as 16 bit unsigned values. The unit of the supply voltage values is 1 mV.

The Supply Voltage Range values are returned in the following format:

Byte	Value	Description
3/2	0...65535	Maximum allowed supply voltage of encoder in mV
1/0	0...65535	Minimum allowed supply voltage of encoder in mV

Table 82. Supply Voltage Range definition.

Offset based access can be used for returning only one of two supply voltage range values.

Offset value	Message Length	Return values
0000h	4	Minimum and maximum supply voltage
0000h	2	Minimum supply voltage
0002h	2	Maximum supply voltage

Table 83. Supply Voltage Range offset choices.

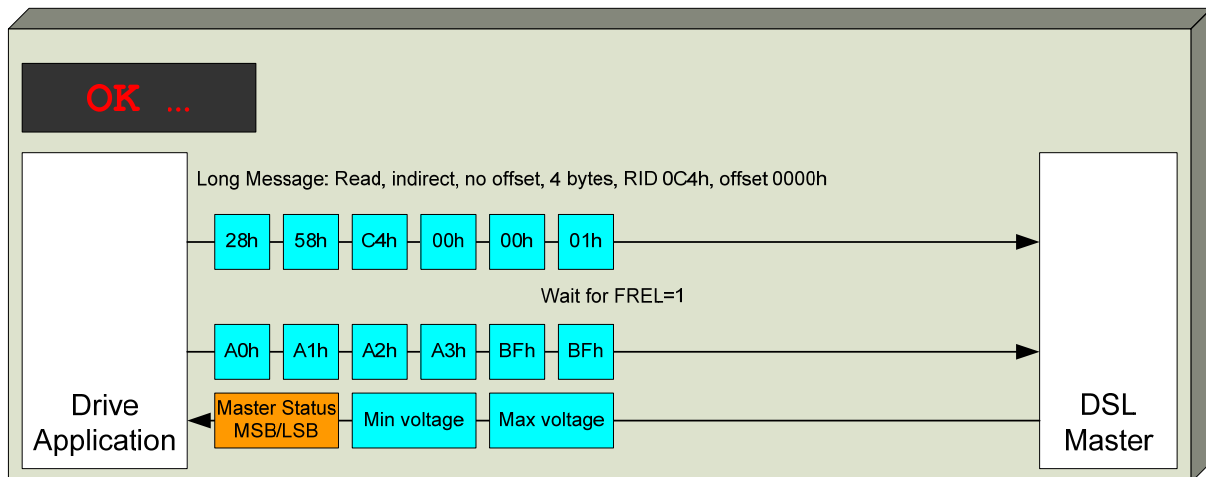


Figure 48. Read Supply Voltage Range.

7.2.3.6. Supply Voltage

This resource indicates the supply voltage of a DSL motor-feedback system.

If the measured supply voltage lies beyond one of the boundaries an error will be indicated (see chapter 6.6.4, error group 3, error no 2).

A direct read access to Supply Voltage returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"SUPVOLT"
Resource Size	1	2
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	5

Table 84. Supply Voltage definition values.

The Supply Voltage value is stored as a 16 bit unsigned value. The unit of the Supply Voltage value is 1 mV.

The Supply Voltage value is returned in the following format:

Byte	Value	Description
1/0	0...65535	Current supply voltage of encoder in mV

Table 85. Supply Voltage definition.

Offset based access is not useful for this resource as the resource data is readable with one "Long Message" transaction.

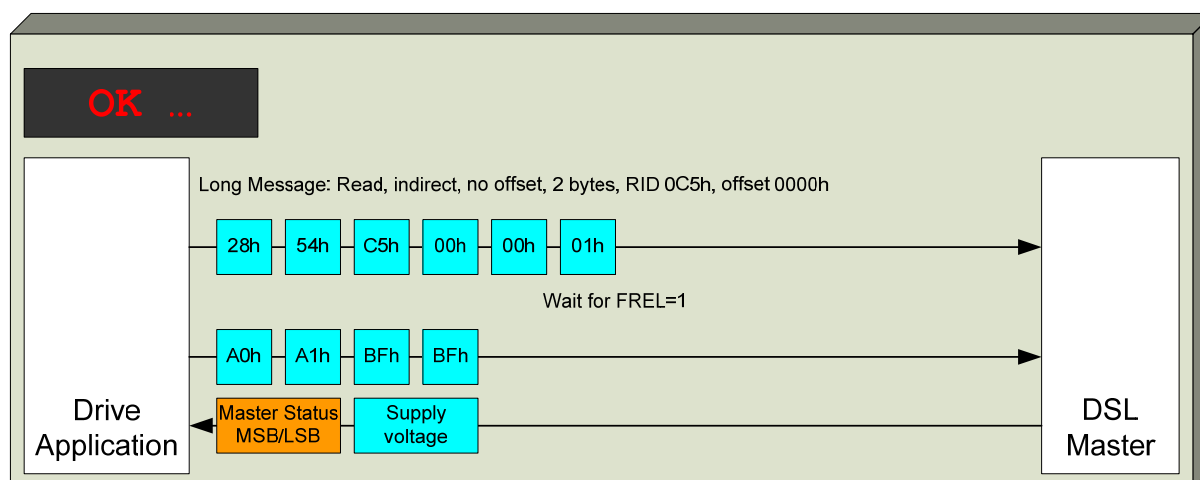


Figure 49. Read Supply Voltage.

7.2.3.7. Speed Range

This resource indicates the maximum allowed shaft speed of a rotary DSL motor-feedback system as stated in the product data sheet.

A direct read access to Speed Range returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"SPEEDRNG"
Resource Size	1	2
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	5

Table 86. Speed Range definition values.

The Speed Range value is stored as a 16 bit signed value. The unit of the speed value is 1 turn per minute (min^{-1}).

Please note that the actual speed value is returned as a process value in the "Horizontal Channel" (see chapter 6.3).

The Speed Range value is returned in the following format:

Byte	Value	Description
1/0	0...32767	Maximum allowed rotational speed of encoder in min^{-1}

Table 87. Speed Range definition.

Offset based access is not useful for this resource as the resource data is readable with one "Long Message" transaction.

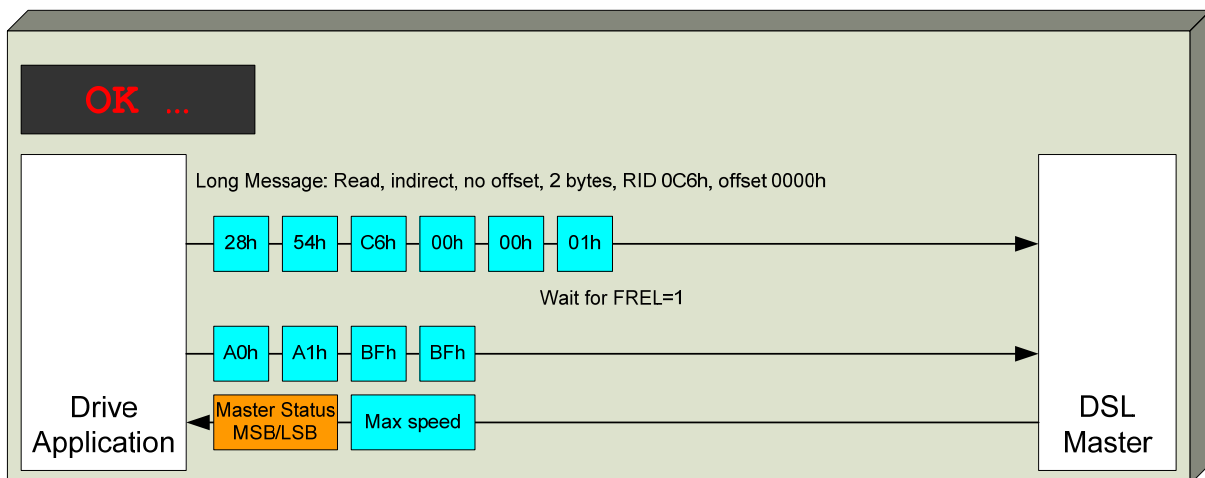


Figure 50. Read Speed Range.

7.2.3.8. Speed

This resource indicates the shaft speed of a rotary DSL motor-feedback system.

If the measured speed lies beyond the boundary an error will be indicated (see chapter 6.6.4, error group 3, error no 3).

A direct read access to Speed returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"SPEED"
Resource Size	1	2
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	5

Table 88. Speed definition values.

The Speed value is stored as a 16 bit signed value. The unit of the speed value is 1 turn per minute (min^{-1}).

Please note that a speed value synchronous to the DSL measurement trigger is returned as a process value in the "Horizontal Channel" (see chapter 6.3).

The Speed value is returned in the following format:

Byte	Value	Description
1/0	0...32767	Actual speed of encoder in min^{-1}

Table 89. Speed definition.

Offset based access is not useful for this resource as the resource data is readable with one "Long Message" transaction.

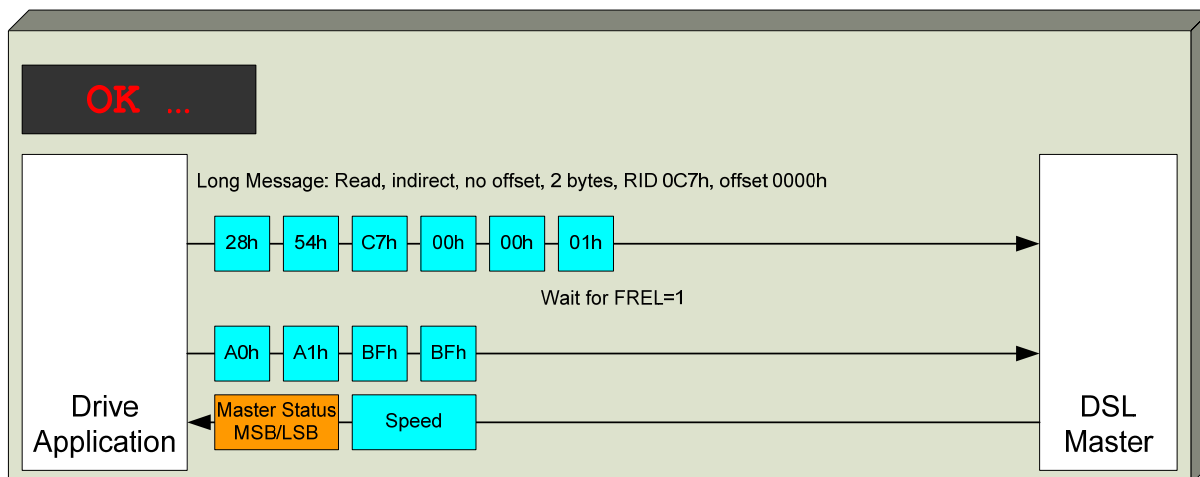


Figure 51. Read Speed.

7.2.3.9. Acceleration Range

This resource indicates the maximum allowed shaft acceleration of a rotary DSL motor-feedback system as stated in the product data sheet.

A direct read access to Acceleration Range returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"ACCRANGE"
Resource Size	1	2
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	5

Table 90. Acceleration Range definition values.

The Acceleration Range value is stored as a 16 bit signed value. The unit of the acceleration value is 1000 rad/s².

Please note that the actual acceleration value can be determined from the speed process value in the "Horizontal Channel" (see chapter 6.3).

The Acceleration Range value is returned in the following format:

Byte	Value	Description
1/0	0...32767	Maximum allowed rotational acceleration of encoder in 1000 rad/s ²

Table 91. Acceleration Range definition.

Offset based access is not useful for this resource as the resource data is readable with one "Long Message" transaction.

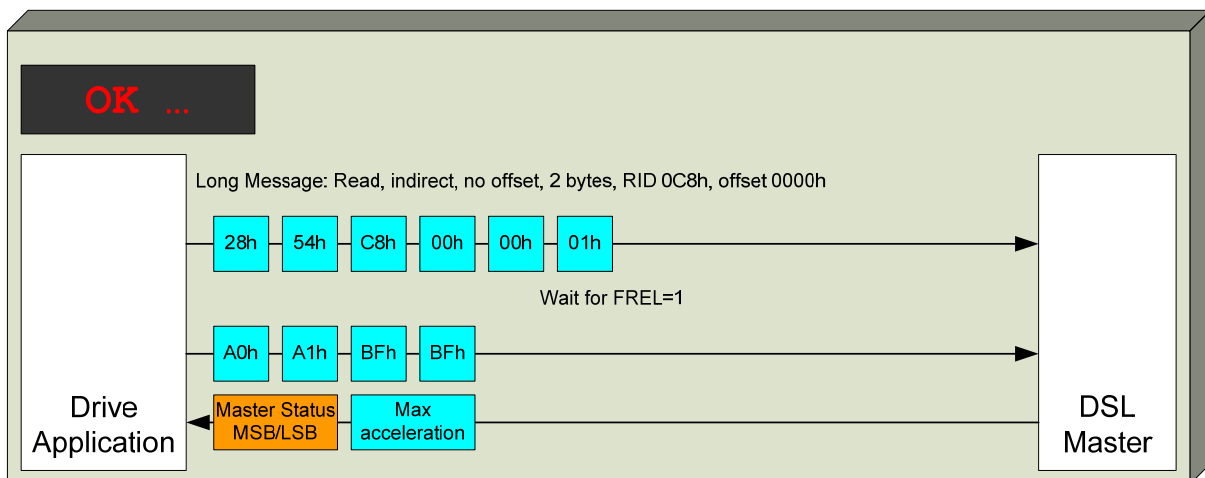


Figure 52. Read Acceleration Range.

7.2.3.10. Acceleration

This resource indicates the current shaft acceleration of a rotary DSL motor-feedback system.

If the measured acceleration lies beyond the boundary an error will be indicated (see chapter 6.6.4, error group 3, error no 4).

A direct read access to Acceleration returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"ACC"
Resource Size	1	2
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	5

Table 92. Acceleration definition values.

The Acceleration value is stored as a 16 bit signed value. The unit of the acceleration value is 1000 rad/s².

Please note that a acceleration value synchronous to the DSL measurement trigger can be determined from the speed process value in the "Horizontal Channel" (see chapter 6.3).

The Acceleration value is returned in the following format:

Byte	Value	Description
1/0	0...32767	Actual rotational acceleration of encoder in 1000 rad/s ²

Table 93. Acceleration definition.

Offset based access is not useful for this resource as the resource data is readable with one "Long Message" transaction.

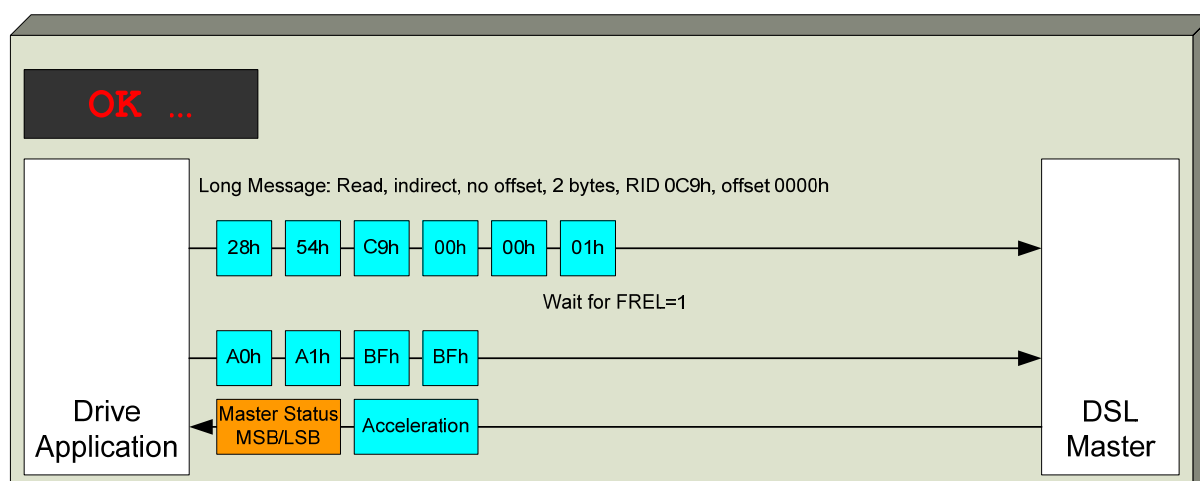


Figure 53. Read Acceleration.

7.2.3.11. Life-time

This resource indicates the operating time and the expected remaining life-time of the DSL motor-feedback system.

A direct read access to Life-time returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"LIFETIME"
Resource Size	1	8
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	5

Table 94. Life-time definition values.

The expected remaining life-time of the product is determined from the statistical mean life-time. For estimating this value the operating time and operating temperature are taken into account.



Please note that the expected remaining life-time only gives a statistical estimate based on standard failure models of electronic components. The value can not reflect other failure models due to mechanical faults or environmental conditions.

Both operating time and remaining life-time values are stored as unsigned 32 bit values. The unit of both values is 1 minute.

Examples for time values are:

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

Table 95. Life-time examples.

The Life-time values are returned in the following format:

Byte	Value	Description
7...4	0...4294967295	Expected remaining life-time in minutes
3...0	0...4294967295	Operating time in minutes

Table 96. Life-time definition.

Offset based access can be used for returning only one of two Life-time values.

Offset value	Message Length	Return values
0000h	8	Operating and remaining life-time
0000h	4	Operating time
0004h	4	Remaining life-time

Table 97. Life-time offset choices.

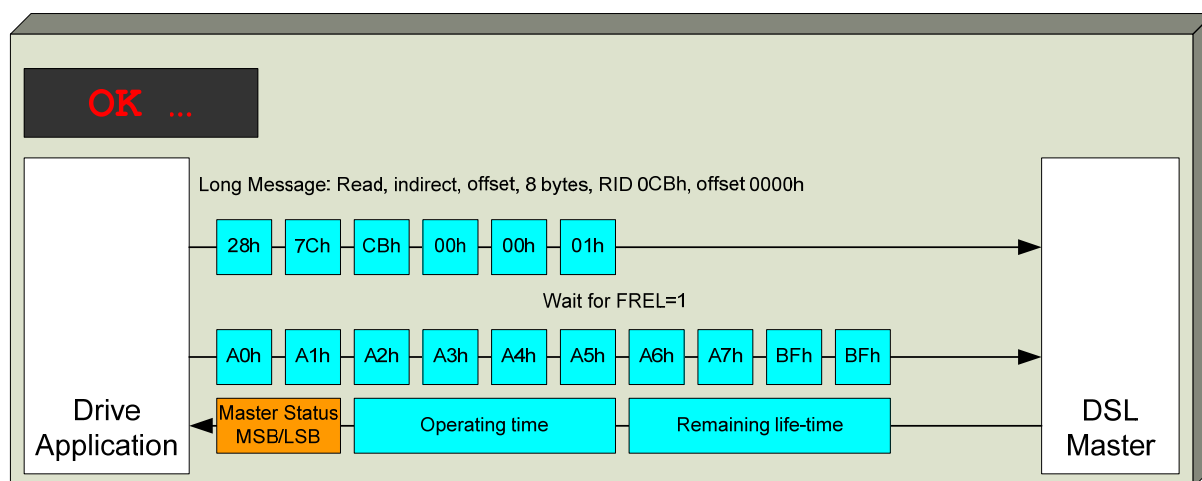


Figure 54. Read Life-time.

7.2.3.12. Error Log

This resource returns stored error messages of the DSL motor-feedback system.

A direct read access to Error Log returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"ERRORLOG"
Resource Size	1	16 per error log entry
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	30

Table 98. Error Log definition values.

If the encoder detects an error this is indicated to the drive application as explained in chapter 6.5. Additionally, errors are stored in the non-volatile memory of the DSL motor-feedback system.

The Error Log also stores those errors that were detected while a DSL link could not be established to the drive application. This resource allows review of these errors.

A DSL motor-feedback system can store a fixed amount of errors in its error log. If more errors are recorded than indicated by this number, the oldest error entries are overwritten. Please refer to the product datasheet to find out about the maximum number of error log entries.

Each error is recorded with a time indication and along several process and environment values from the time of the error occurrence.

The administration resource Diary Options (see chapter 7.2.4.6) allows to set the format of the time indication for an error.

Error Log entries are stored in 16 bytes each.

The Error Log entry values are returned in the following format:

Byte	Value	Description
15	00...FFh	Error code (see chapter 6.6.4)
14	-	Reserved
13/12	0 ... 65535	Acceleration during error in 1000 rad/s ²
11/10	0 ... 65535	Speed during error in min ⁻¹
9/8	0...65535	Internal supply voltage during error in mV
7/6	0...65535	LED current during error in 0.1 mA
5/4	-2730...10000	Temperature during error in 0.1 °C
3...0	0...4294967295	Error time indication (operating time or real time stamp)

Table 99. Error Log entry definition.

Each Error Log entry is stored subsequently and can be accessed by specifying appropriate offset address values.

To return the full error entry, two “Long Message” transactions have to be conducted since one message can only carry 8 bytes.



Please note that the highest offset address that can be specified depends on the maximum number of error log entries for a given product.

Offset value	Message Length	Return values
0000h	8	First part of most recent error
0008h	8	Second part of most recent error
0010h	8	First part of error log entry #2
0018h	8	Second part of error log entry #2
...	8	...
xxx0h	8	First part of oldest error log entry
xxx8h	8	Second part of oldest error log entry

Table 100. Error Log offset choices.

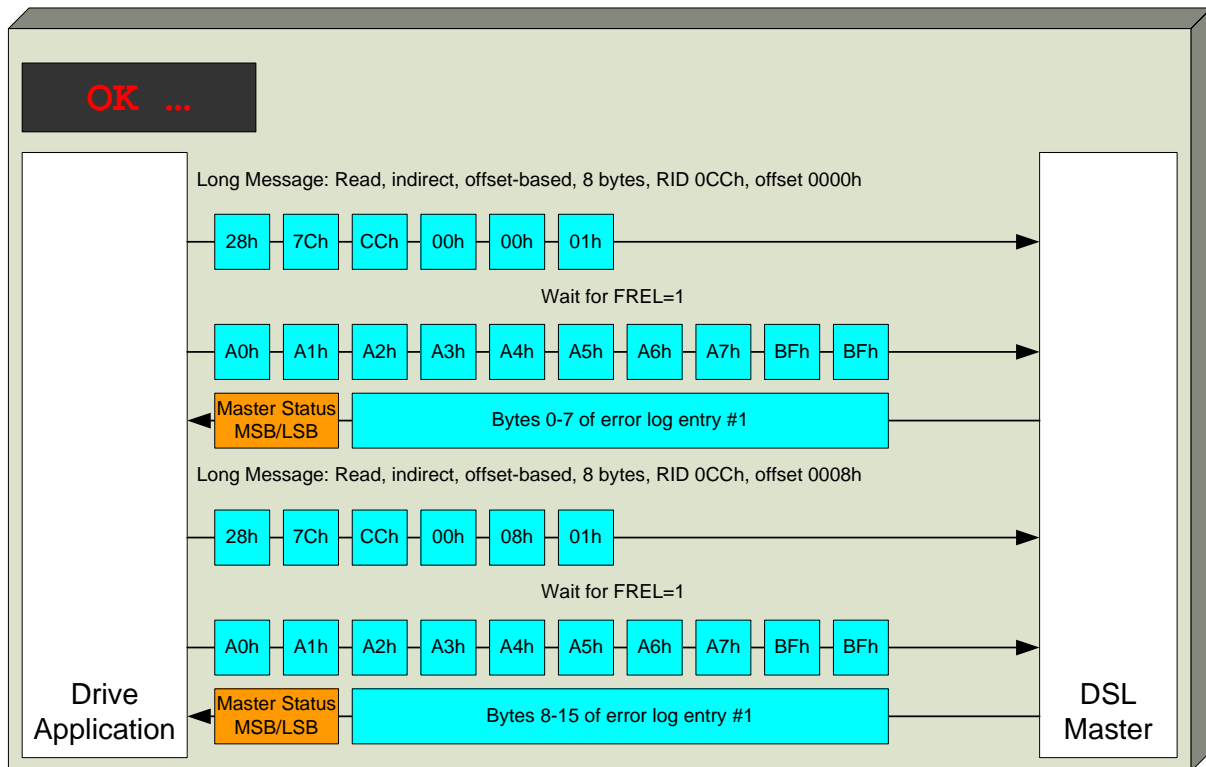


Figure 55. Read Error Log entry.

7.2.3.13. Usage Histogram

This resource returns histogram values of encoder parameters. The histogram values show how often a parameter value was measured during the life-time of the encoder.

A direct read access to Usage Histogram returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"USAGEHIS"
Resource Size	1	4
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	30

Table 101. Usage Histogram definition values.

Within the diary options (see chapter 7.2.4.6) it can be specified which encoder parameter shall be logged by the usage histogram.

Logged encoder parameters are measured and stored every 1 minute.

The following table specifies which encoder parameters can be logged and stored in a histogram. Also, the boundaries and resolution of the histogram classes is indicated.

Encoder Parameter	Min class	Max class	Histogram class width
Temperature	< -40 °C	>= 120 °C	10 °C
LED Current	0 ... 10 mA	>= 100 mA	10 mA
Supply Voltage	< 2.0 V	>= 30.0 V	1.0 V
Speed	0 ... 500 rpm	>= 10 000 rpm	500 rpm
Acceleration	0 ... 50 krad/s ²	>= 500 krad/s ²	50 krad/s ²
External Temperature	< -40 °C	>= 300 °C	10 °C

Table 102. Encoder parameter histogram definitions.

Usage Histogram values are stored in 4 bytes each.

The Usage Histogram values are returned in the following format:

Byte	Value	Description
3/2/1	00 00 00h - FF FF FFh	Amount of parameter values in histogram class
0	00h - FFh	Histogram class identifier

Table 103. Usage Histogram value definition.

The histogram class identifier is transmitted in the offset value along with an identifier for the requested encoder parameter. The histogram class identifier depends on the chosen encoder parameter as detailed in the following table.

HIPERFACE DSL®

Encoder Parameter	Histogram Class	Histogram class identifier
Temperature	< -40 °C	00h
	-40 ... -30 °C	01h
	-30 ... -20 °C	02h
	-20 ... -10 °C	03h
	-10 ... 0 °C	04h
	0 ... 10 °C	05h
	10 ... 20 °C	06h
	20 ... 30 °C	07h
	30 ... 40 °C	08h
	40 ... 50 °C	09h
	50 ... 60 °C	0Ah
	60 ... 70 °C	0Bh
	70 ... 80 °C	0Ch
	80 ... 90 °C	0Dh
	90 ... 100 °C	0Eh
	100 ... 110 °C	0Fh
110 ... 120 °C	10h	
>= 120 °C	11h	
LED Current	0 ... 10 mA	00h
	10 ... 20 mA	01h
	20 ... 30 mA	02h
	30 ... 40 mA	03h
	40 ... 50 mA	04h
	50 ... 60 mA	05h
	60 ... 70 mA	06h
	70 ... 80 mA	07h
	80 ... 90 mA	08h
	90 ... 100 mA	09h
	> 100 mA	0Ah
Supply Voltage	< 2.0 V	00h
	2.0 ... 3.0 V	01h
	3.0 ... 4.0 V	02h
	4.0 ... 5.0 V	03h
	5.0 ... 6.0 V	04h
	6.0 ... 7.0 V	05h
	7.0 ... 8.0 V	06h
	8.0 ... 9.0 V	07h
	9.0 ... 10.0 V	08h
	10.0 ... 11.0 V	09h
	11.0 ... 12.0 V	0Ah
	12.0 ... 13.0 V	0Bh
	13.0 ... 14.0 V	0Ch
	14.0 ... 15.0 V	0Dh
	15.0 ... 16.0 V	0Eh
	16.0 ... 17.0 V	0Fh
	17.0 ... 18.0 V	10h
	18.0 ... 19.0 V	11h
	19.0 ... 20.0 V	12h
	20.0 ... 21.0 V	13h
	21.0 ... 22.0 V	14h
	22.0 ... 23.0 V	15h
23.0 ... 24.0 V	16h	
24.0 ... 25.0 V	17h	
25.0 ... 26.0 V	18h	
26.0 ... 27.0 V	19h	
27.0 ... 28.0 V	1Ah	
28.0 ... 29.0 V	1Bh	

	29.0 ... 30.0 V	1Ch
	> 30.0 V	1Dh
Speed	0 ... 500 rpm	00h
	500 ... 1000 rpm	01h
	1000 ... 1500 rpm	02h
	1500 ... 2000 rpm	03h
	2000 ... 2500 rpm	04h
	2500 ... 3000 rpm	05h
	3000 ... 3500 rpm	06h
	3500 ... 4000 rpm	07h
	4000 ... 4500 rpm	08h
	4500 ... 5000 rpm	09h
	5000 ... 5500 rpm	0Ah
	5500 ... 6000 rpm	0Bh
	6000 ... 6500 rpm	0Ch
	6500 ... 7000 rpm	0Dh
	7000 ... 7500 rpm	0Eh
	7500 ... 8000 rpm	0Fh
	8000 ... 8500 rpm	10h
	8500 ... 9000 rpm	11h
	9000 ... 9500 rpm	12h
	9500 ... 10000 rpm	13h
> 10000 rpm	14h	
Acceleration	0 ... 50 krad/s ²	00h
	50 ... 100 krad/s ²	01h
	100 ... 150 krad/s ²	02h
	150 ... 200 krad/s ²	03h
	200 ... 250 krad/s ²	04h
	250 ... 300 krad/s ²	05h
	300 ... 350 krad/s ²	06h
	350 ... 400 krad/s ²	07h
	400 ... 450 krad/s ²	08h
	450 ... 500 krad/s ²	09h
	> 500 krad/s ²	0Ah
	External Temperature	< -40 °C
-40 ... -30 °C		01h
-30 ... -20 °C		02h
-20 ... -10 °C		03h
-10 ... 0 °C		04h
0 ... 10 °C		05h
10 ... 20 °C		06h
20 ... 30 °C		07h
30 ... 40 °C		08h
40 ... 50 °C		09h
50 ... 60 °C		0Ah
60 ... 70 °C		0Bh
70 ... 80 °C		0Ch
80 ... 90 °C		0Dh
90 ... 100 °C		0Eh
100 ... 110 °C		0Fh
110 ... 120 °C		10h
120 ... 130 °C		11h
130 ... 140 °C		12h
140 ... 150 °C		13h
150 ... 160 °C	14h	
160 ... 170 °C	15h	
170 ... 180 °C	16h	
180 ... 190 °C	17h	

HIPERFACE DSL®

	190 ... 200 °C	18h
	200 ... 210 °C	19h
	210 ... 220 °C	1Ah
	220 ... 230 °C	1Bh
	230 ... 240 °C	1Ch
	240 ... 250 °C	1Dh
	250 ... 260 °C	1Eh
	260 ... 270 °C	1Fh
	270 ... 280 °C	20h
	280 ... 290 °C	21h
	290 ... 300 °C	22h
	>= 300 °C	23h

Table 104. Histogram classes.

The offset value must be specified in the following format:

Bits	Value	Definition
0 ... 7	00h – FFh	Histogram class identifier
8 ... 11	0h	Request Temperature
	1h	Request LED Current
	2h	Request Supply Voltage
	3h	Request Speed
	4h	Request Acceleration
	5h	Request External Temperature

Table 105. Usage Histogram offset choices.

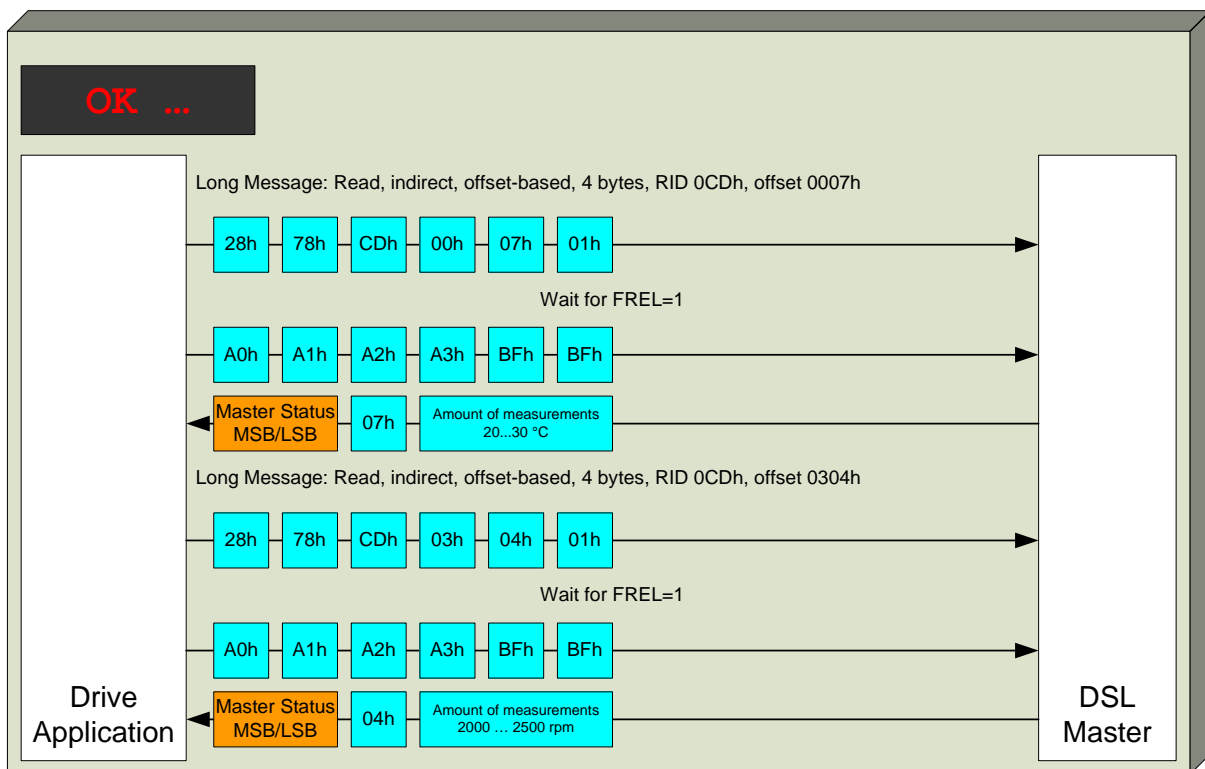


Figure 56. Read Usage Histogram entry.

7.2.4. Administration Resources

The administration resources of the DSL motor-feedback system give access to setting encoder options.

The following table shows an overview over all administration resource definitions (explanation of the columns see Table 48).

Resource	RID	Size	R	W	Timeout
Reset	100h	0	15	0	n/a
Set Position	101h	8	15	1	40
Time/Date	103h	16	0	0	5
Set Access Level	104h	8	0	0	5
Change Access Key	105h	8	15	0	40
Diary Options	106h	2	1	1	40

Table 106. Administration resource definitions.

7.2.4.1. Reset

A write access to this resource leads to a reset of the DSL motor-feedback system.

A direct read access to Reset returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"RESET"
Resource Size	1	0
Resource Read Access	2	15
Resource Write Access	3	0
Resource Time-out	4	0

Table 107. Reset definition values.

The encoder will initialize in the same way as after a power-on (see chapter 6.1). There will be no return message to the Reset command.

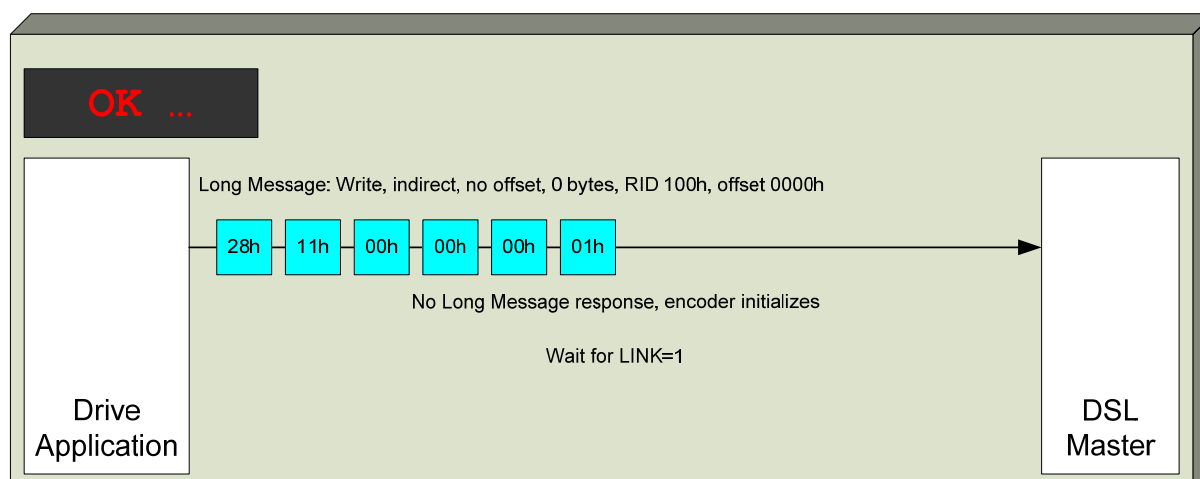


Figure 57. Write Reset command.

7.2.4.2. Set Position

This resource triggers an assignment of an arbitrary position value to the current mechanical shaft position.



In synchronous servo drives the position information is used for motor commutation. Improper use of this resource may result in an impairment of the motor. This function should only be called by motor manufacturers.

A direct read access to Set Position returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"SETPOS"
Resource Size	1	8
Resource Read Access	2	15
Resource Write Access	3	1
Resource Time-out	4	40

Table 108. Set Position definition values.



No synchronization for on-going movements is foreseen. This function must only be used during stand-still of the encoder shaft.

The position value that is to be assigned to the current shaft position is transmitted as an unsigned 40 bit value. Only values within the measurement range of the DSL motor-feedback system are valid.

The position value for this command is specified in the following format:

Byte	Value	Description
4...0	00 0000 0000 ... FF FFFF FFFFh	New position value for current mechanical shaft position. Position value is right-aligned.

Table 109. Set Position definition.

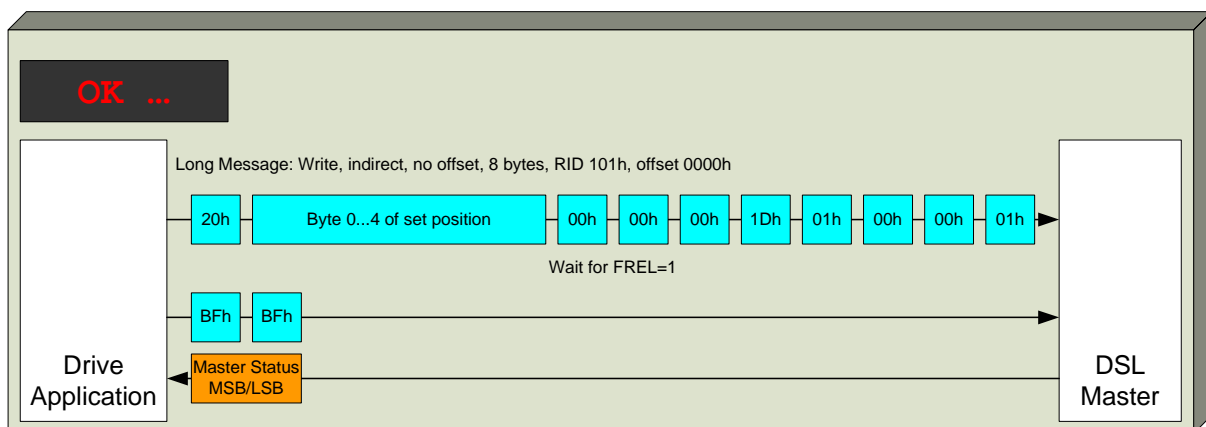


Figure 58. Write Set Position command.

7.2.4.3. Time/Date

This resource indicates the real-time and date that the motor-feedback system keeps track of. The current time and date can be set by a write access to this resource.



Unless otherwise specified in the encoder datasheet, please note that a DSL motor-feedback system can only keep track of time during power-on. For using correct time and date, the values must be set after each power-on.

A direct read access to Time/Date returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"TIMEDATE"
Resource Size	1	16
Resource Read Access	2	0
Resource Write Access	3	0
Resource Time-out	4	5

Table 110. Time/Date definition values.

The time and date values are stored as ASCII strings in 8 bytes each.

Examples for time and date strings are:

Time/Date	Resource Value (hex)
11:15:23	3131 3A31 353A 3233h
20090424	3230 3039 3034 3234h

Table 111. Time/Date examples.

The Time/Date values are returned in the following format:

Byte	Value	Description
15...8	yyyymmdd	String representation of date: yyyy four ASCII characters: year mm two ASCII characters: month dd two ASCII characters: day
7...0	hh:mm:ss	String representation of time: hh two ASCII characters: hour mm two ASCII characters: minute ss two ASCII characters: second : ASCII constant (3Ah)

Table 112. Time/Date definition.

Offset based access can be used for returning either date or time values.

Offset value	Message Length	Return values
0000h	8	Current time
0008h	8	Current date

Table 113. Time/Date offset choices.

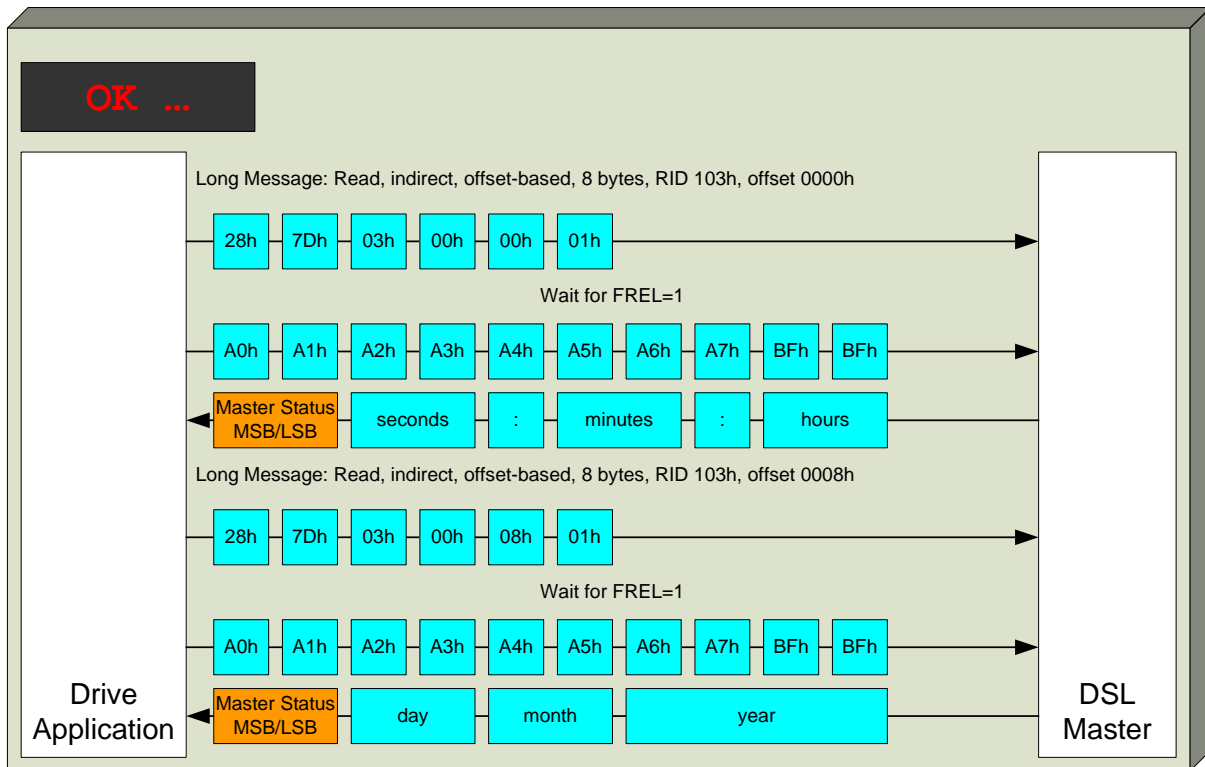


Figure 59. Read Time/Date.

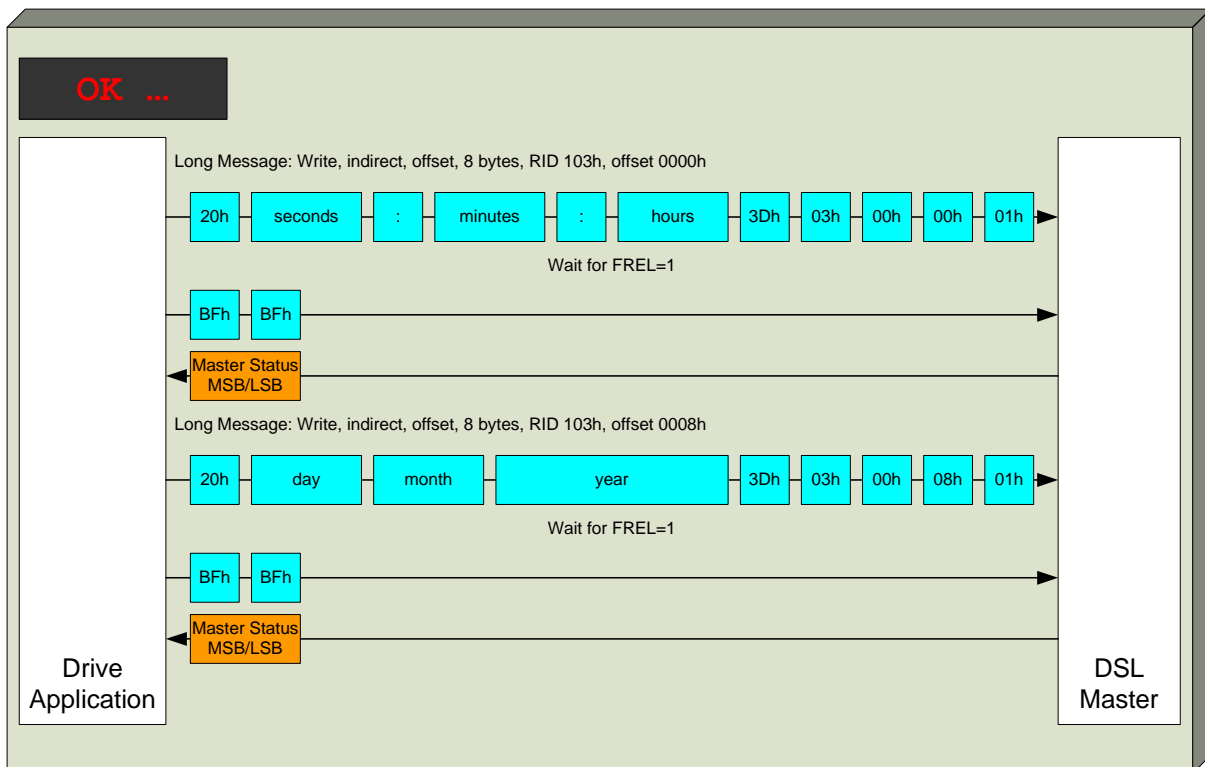


Figure 60. Write Time/Date.

7.2.4.4. Set Access Level

This resource allows to set or read the encoder access level. The access level determines which functions can be accessed by the user application. The needed access level is specified for each function in the resource list (this chapter 7.2).

A direct read access to Set Access Level returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"SETACCES"
Resource Size	1	8
Resource Read Access	2	0
Resource Write Access	3	0
Resource Time-out	4	5

Table 114. Set Access Level definition values.

After power-on or after a reset the access level will always be set to "0", indicating lowest (public) access privileges.

To change the access level the corresponding access key must be transmitted to the DSL encoder. The access level will be kept until another level is set with this resource.

The following Table 115 specifies the available access levels and the default access keys for setting the corresponding level.

Access Level	Default Access Key	Usage
0	No access key needed	Public 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

Table 115. Access levels and default access keys.

The access level is set in the following format:

Byte	Value	Description
7/6/5/4	0000 0000 ... FFFF FFFFh	Access key for requested access level
3/2/1		Reserved for future use
0	00h – 04h	Requested access level

Table 116. Set Access Level definition.

With a read access the currently set access level can be determined. It is returned in byte 0 of the Long Message.

Offset based access is not useful for this resource as the size of the resource data is less than the maximum for one "Long Message" transaction.

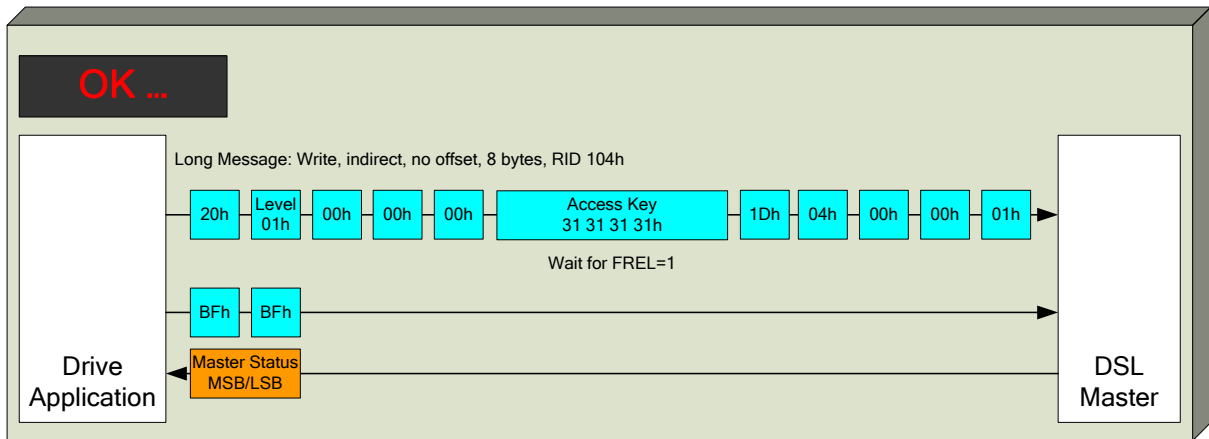


Figure 61. Set Access Level (in this example: 01h with access key 31 31 31 31h).

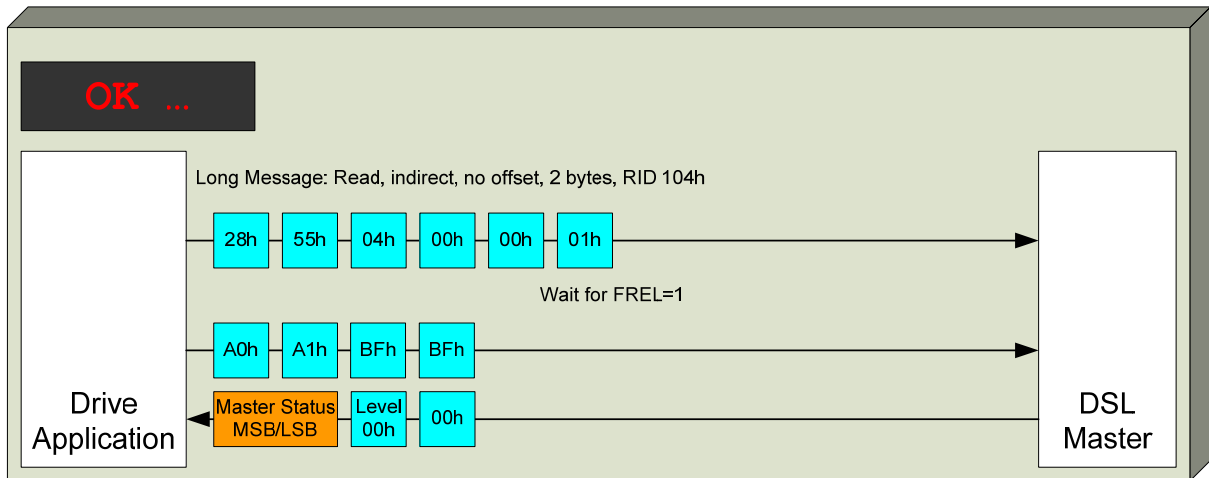


Figure 62. Read current access level (in this example: 00h).

7.2.4.5. Change Access Key

This resource allows to change the access key required for setting the appropriate access level. The access level determines which functions can be accessed by the user application. The needed access level is specified for each function in the resource list (this chapter 7.2).

A direct read access to Change Access Key returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"CHNGEKEY"
Resource Size	1	8
Resource Read Access	2	15
Resource Write Access	3	0
Resource Time-out	4	40

Table 117. Change Access Key definition values.

To change the access key the old and new access key of the target level as well as the access level itself must be transmitted to the DSL encoder.

Please note that the access key for any level can be changed regardless of currently chosen access level.

The access key is changed by transmitting data in the following format:

Byte	Value	Description
7/6/5/4	0000 0000 ... FFFF FFFFh	Old access key
3/2/1/0	0000 0000 ... FFFF FFFFh	New access key

Table 118. Change Access Key definition.

A read access to this resource is not possible.

The offset value indicates the target access level for the key change.

Offset value	Description
0 – 4	Target access level

Table 119. Access Level choices.

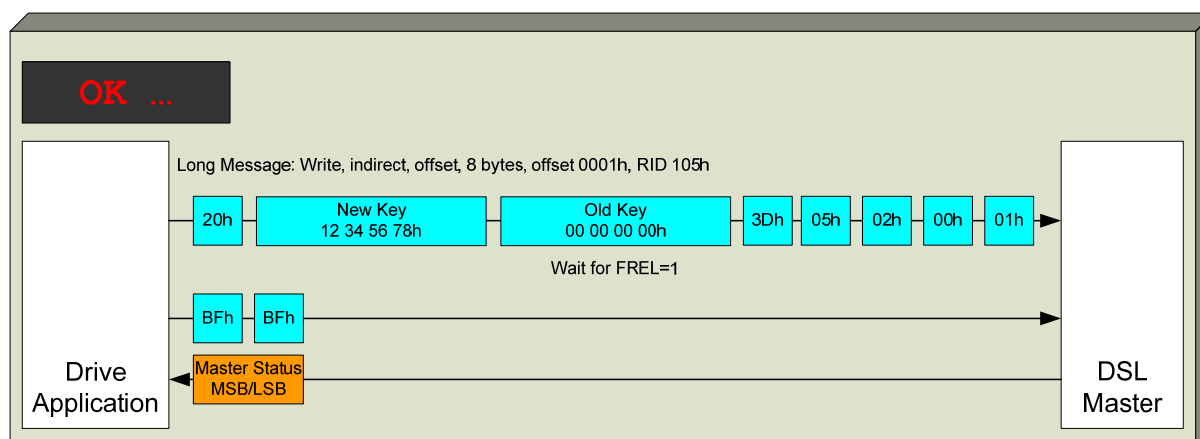


Figure 63. Change Access Key (in this example for Access Level 02h, from 0000 0000h to 1234 5678h).

7.2.4.6. Diary Options

This resource allows to set options regarding logging of encoder errors and usage history (see chapters 7.2.3.12, 7.2.3.13).

A direct read access to Diary Options returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"DIARYOPT"
Resource Size	1	2
Resource Read Access	2	1
Resource Write Access	3	1
Resource Time-out	4	40

Table 120. Diary Options definition values.

By writing 2 data bytes to this resource, these functions are enabled and configured as detailed in the following table:

Byte	Bit	Value	Description
0	0 (LSB)	0	Error Log switched off
		1	Error Log switched on (default)
	1	0	Usage Histogram switched off (default)
		1	Usage Histogram switched on
	2	0	Time Format for Error Log: Operating Time
1		Time Format for Error Log: Real Time	
	3 ... 7		Reserved for future use
1	0 (LSB)	0	Encoder Temperature Histogram Off
		1	Encoder Temperature Histogram On
	1	0	LED Current Histogram Off
		1	LED Current Histogram On
	2	0	Supply Voltage Histogram Off
		1	Supply Voltage Histogram On
	3	0	Speed Histogram Off
1		Speed Histogram On	
4	0	Acceleration Histogram Off	
	1	Acceleration Histogram On	
5	0	External Temperature Histogram Off	
	1	External Temperature Histogram On	
	6 ... 7		Reserved for future use

Table 121. Diary Options definition.

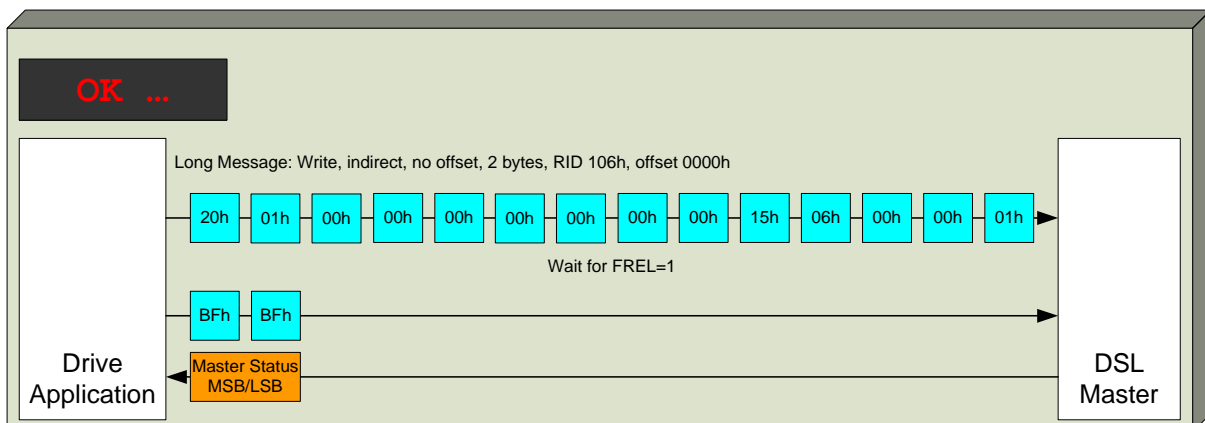


Figure 64. Write Diary Options command (here: enable Error Log with operating time, disable rest).

7.2.5. Counter Resources

The counter implemented in the Hiperface DSL® motor-feedback system is a freely incrementable 32 bit counter for user purposes. It can be read, incremented, or reset.

The following table shows an overview over all counter resource definitions (explanation of the columns see Table 48).

Resource	RID	Size	R	W	Timeout
Read Counter	120h	4	0	15	5
Increment Counter	121h	0	15	0	40
Reset Counter	122h	0	15	1	40

Table 122. Counter resource definitions.

7.2.5.1. Read Counter

This resource indicates the value of a user-defined counter. The counter value is returned as an unsigned 32 bit value.

A direct read access to Read Counter returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"READCNT"
Resource Size	1	4
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	5

Table 123. Read Counter definition values.

The counter value is returned in the following format:

Byte	Value	Description
3/2/1/0	0000 0000 ... FFFF FFFFh	User counter value

Table 124. Read Counter definition.

Offset based access is not useful for this resource as the resource data is readable with one "Long Message" transaction.

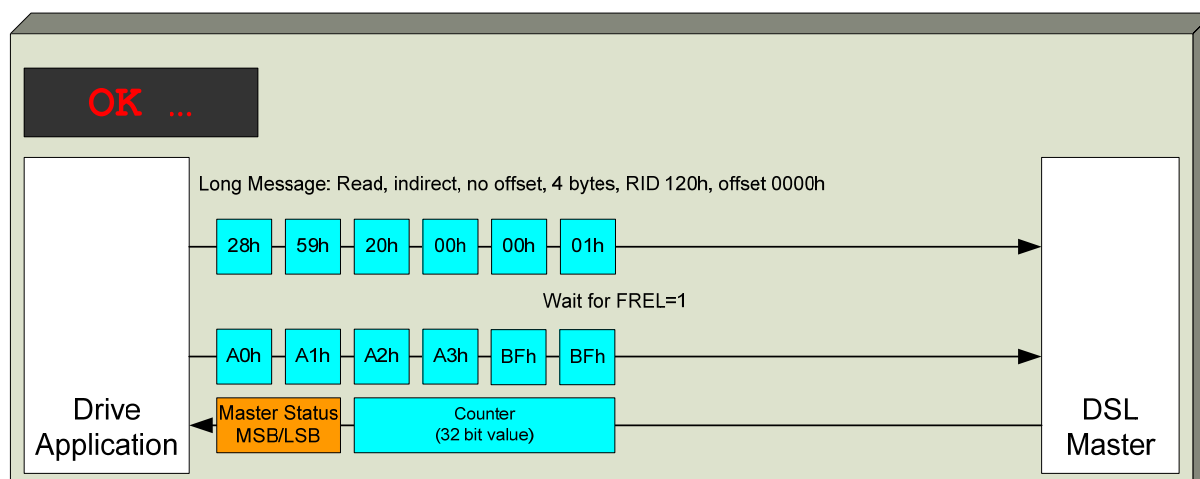


Figure 65. Read Counter.

7.2.5.2. Increment Counter

This resource causes an increment of the 32 bit user counter. If the increment causes an overflow of the counter, the error message 35 is returned (see chapter 6.6.4).

A direct read access to Increment Counter returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"INCCOUNT"
Resource Size	1	0
Resource Read Access	2	15
Resource Write Access	3	0
Resource Time-out	4	40

Table 125. Increment Counter definition values.

The increment is performed through a write command to this resource without any data (Long Message length = 0).

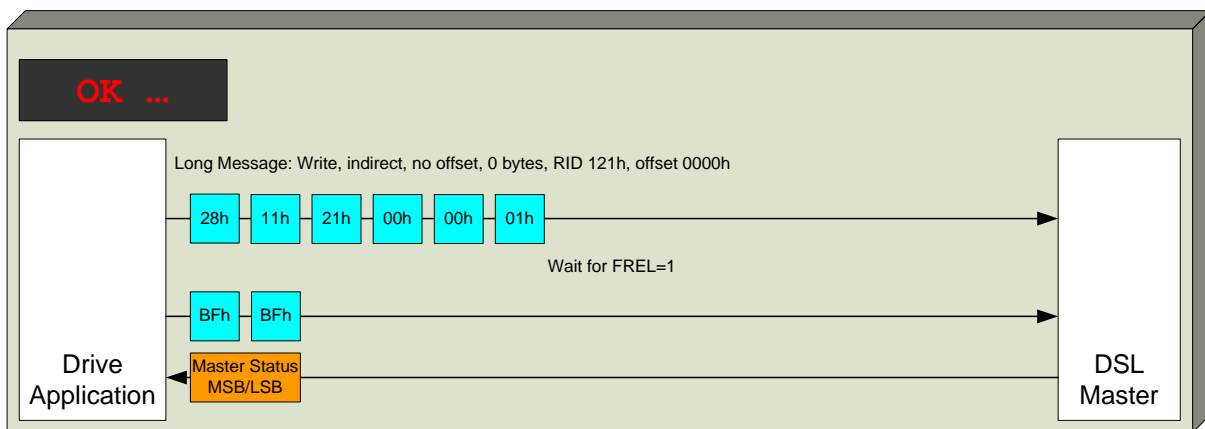


Figure 66. Increment Counter command.

7.2.5.3. Reset Counter

This resource causes a reset of the 32 bit user counter.

A direct read access to Reset Counter returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"RESETCNT"
Resource Size	1	0
Resource Read Access	2	15
Resource Write Access	3	1
Resource Time-out	4	40

Table 126. Reset Counter definition values.

The reset is performed through a write command to this resource without any data (Long Message length = 0).

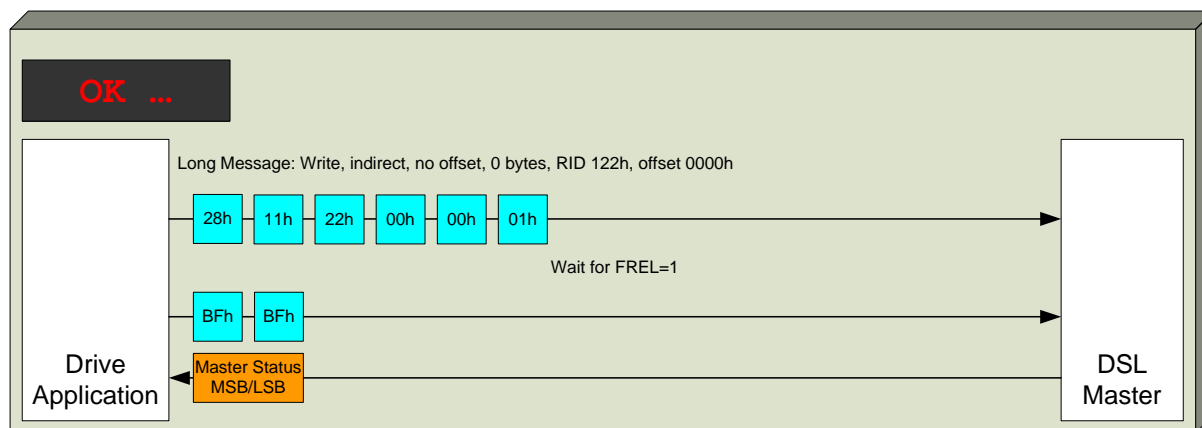


Figure 67. Reset Counter command.

7.2.6. Data Storage Resources

The data storage resources of the DSL motor-feedback system allow the user to access user-defined files for general purpose data storage.

User data is stored in non-volatile memory (EEPROM) and automatically protected by cyclic redundancy checksums (CRC). The CRC mechanism gives the user very high reliability in detecting storage errors of user data.

The following table shows an overview over all data storage resource definitions (explanation of the columns see Table 48).

Resource	RID	Size	R	W	Timeout
Load File	130h	8	15	1	40
Read/Write File	131h	8	0...4	0...4	40
File Status	132h	4	0	15	5
Create/Delete/Modify	133h	8	15	0...4	70
Directory	134h	8	0	15	40

Table 127. Data storage resource definitions.

The following figure shows workflows for handling data storage. Each step is a single resource access (Long Message).

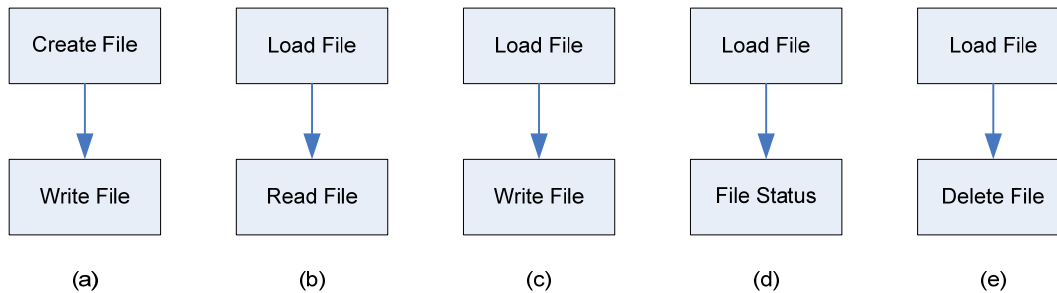


Figure 68. Data storage workflows: (a) Writing to a new file. (b) Reading from a file. (c) Writing to an existing file. (d) Querying the status of an existing file. (e) Deleting a file.

* Password protected access to user data files can be configured by the user during file creation.

7.2.6.1. Load File

In order to access an existing user file, it must be loaded first with this resource.

A direct read access to Load File returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"LOADFILE"
Resource Size	1	8
Resource Read Access	2	15
Resource Write Access	3	1
Resource Time-out	4	40

Table 128. Load File definition values.

Please note that only one file can be loaded at one time. If a second file is loaded the first is unloaded automatically.

A file remains loaded until another is loaded or the DSL motor-feedback system is reset or powered down.

A file is indicated by its file name transmitted in the Long Message data buffer. If the file name is unknown the resource "Directory" (see chapter 7.2.6.5) can be used to find existing files.

The file name can be up to 8 bytes long. Each byte represents an ASCII character. The end of the file name (if less than 8 bytes) is indicated by the character "\0" (00h). Please note that file names are case-sensitive.

A file can only be loaded if the currently set access level (see chapter 0) allows either reading or writing to the file. Access rights are set during creation or modification of a file (see chapter 7.2.6.4).

Byte	Value	Description
0 ... 7	Any	Name of file to be loaded

Table 129. Load File definition.

Offset based access is not useful for this resource as the resource data is readable with one "Long Message" transaction.

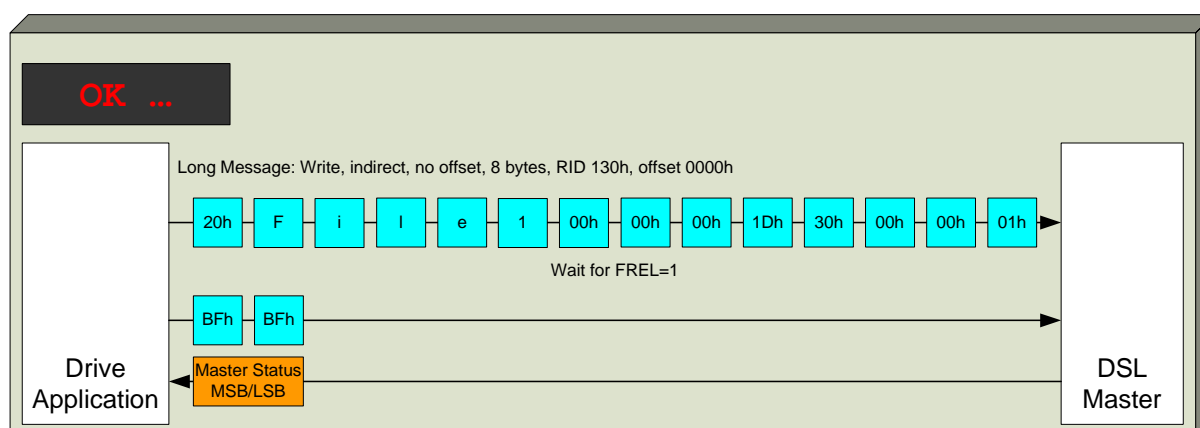


Figure 69. Write Load File command (here: load file with name "File1").

7.2.6.2. Read/Write File

This resource allows read and write access to a user file.

A direct read access to Read/Write File returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"RWFIL"
Resource Size	1	8 (actual size depends on file size)
Resource Read Access	2	0 (actual access level depends on user settings)
Resource Write Access	3	0 (actual access level depends on user settings)
Resource Time-out	4	40

Table 130. Read/Write File definition values.

Prior to reading or writing from a file it must be loaded (see chapter 7.2.6.1).

Reading or writing can be performed with random access to any address within the file. If an address is specified for reading that is higher than the file size, an error message will be returned. If an address is specified for writing that is higher than the file size, the file will be automatically enlarged accordingly. If the remaining EEPROM storage space is not large enough to contain the enlarged file, the access will be stopped and an error message will be returned.

By setting the length value for a Long Message, 2, 4, or 8 bytes can be read/written in one Long Message.

The data from a read access or the data for a write access is stored in the Long Message buffer:

Byte	Value	Description
7 ... 0	Any	Data from file or for file

Table 131. Read/Write File definition.

The offset value indicates the target address for read/write access. Please note that files have a maximum size of 32.768 bytes.

Offset value	Description
0 ... 32767	Address for read/write access

Table 132. Offset value for Read/Write File.

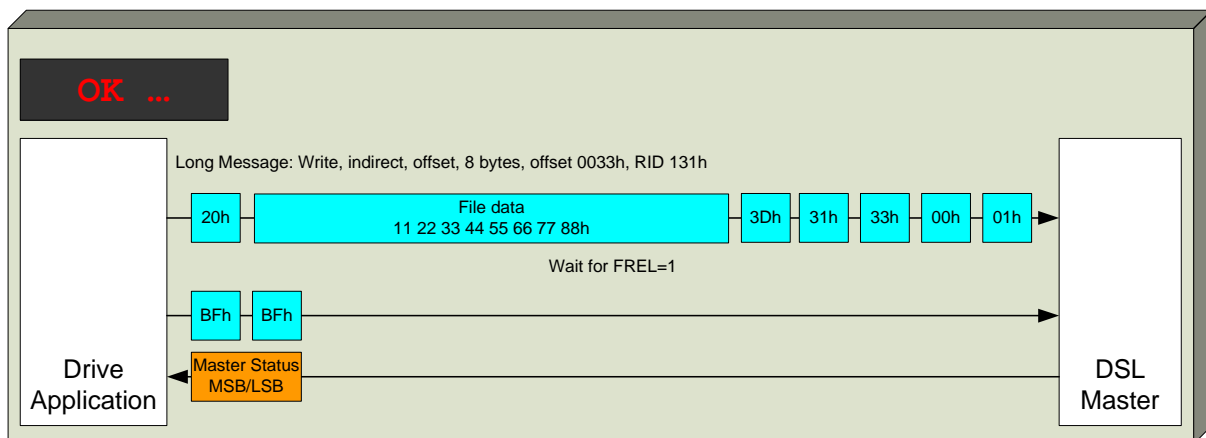


Figure 70. Read/Write File (in this example: Write 8 bytes to address 0033h).

7.2.6.3. File Status

This resource returns the status of the file currently loaded (see chapter 7.2.6.1).

A direct read access to File Status returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"FILESTAT"
Resource Size	1	4
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	5

Table 133. File Status definition values.

A read access to File Status returns the file access rights and the size of the file.

The File Status is returned in the following format:

Byte	Value	Description
3/2	0000 – FFFFh	File size in bytes
1		Reserved for future use
0, bits 7 – 4		Write access rights
	0	Public
	1	Operator
	2	Maintenance
	3	Authorized Client
	4	Service
	5 – 14	Reserved for future use
	15	No writing allowed
0, bits 3 – 0		Read access rights
	0	Public
	1	Operator
	2	Maintenance
	3	Authorized Client
	4	Service
	5 – 14	Reserved for future use
	15	No reading allowed

Table 134. Read/Write File definition.

Offset based access is not useful for this resource as the resource data is readable with one "Long Message" transaction.

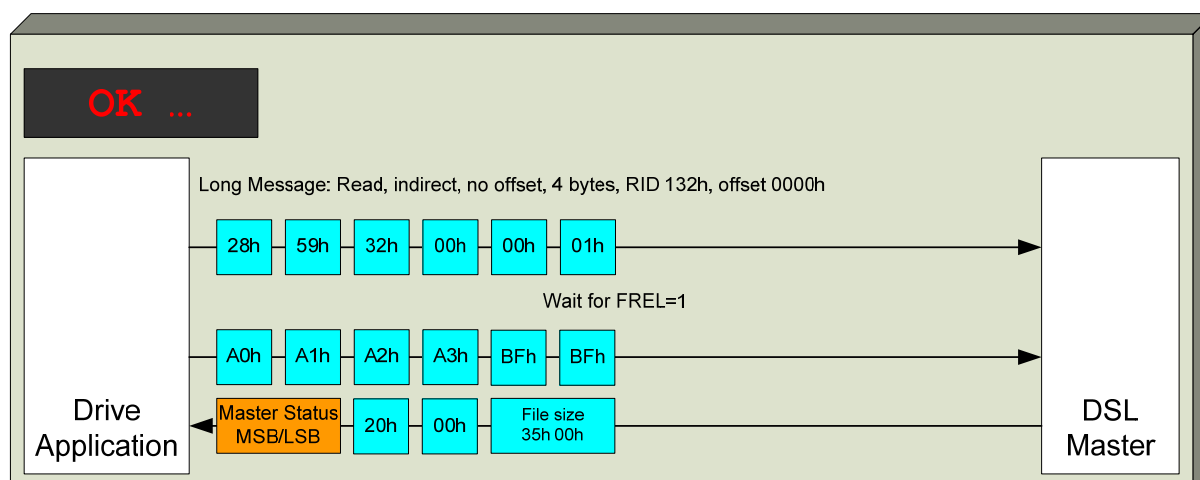


Figure 71. File Status (in this example: File with read access 0, write access 2, file size 53 bytes).

7.2.6.4. Create/Delete/Modify File

This resource allows either creation, modification, or deletion of a user file.

A direct read access to Create/Delete/Modify File returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"MAKEFILE"
Resource Size	1	8
Resource Read Access	2	15
Resource Write Access	3	0 (actual access level depends on user settings)
Resource Time-out	4	70

Table 135. Create/Delete/Modify File definition values.

A user file must be create before the file can be loaded, written to, or read from (see chapter 7.2.6).

Prior to modifying or deleting a file it must be loaded (see chapter 7.2.6.1).

To **create a file**, the name is set in the Long Message buffer. Unused characters of the file are set to "00h". If there is already a user file with the indicated name, the creation will be aborted with an error message.

Byte	Value	Description
7 ... 0	Any	Name of file to be created

Table 136. Create File definition.

Further, for creating a file, the offset value is used:

Bits	Value	Definition
14 – 10		Reserved for future use
9 – 8	11b	Create file
7 – 4	0	Write access rights Public
	1	Operator
	2	Maintenance
	3	Authorized Client
	4	Service
	5 – 14	Reserved for future use
3 – 0	15	No writing allowed
	0	Read access rights Public
	1	Operator
	2	Maintenance
	3	Authorized Client
	4	Service
5 – 14	Reserved for future use	
15	No reading allowed	

Table 137. Offset value for Create File.

To **modify a file**, only the offset value is used:

Bits	Value	Definition
14 – 10		Reserved for future use
9 – 8	01b	Modify file
7 – 4	0 1 2 3 4 5 – 14 15	Write access rights Public Operator Maintenance Authorized Client Service Reserved for future use No writing allowed
3 – 0	0 1 2 3 4 5 – 14 15	Read access rights Public Operator Maintenance Authorized Client Service Reserved for future use No reading allowed

Table 138. Offset value for Modify File.

To **delete a file**, the file name (of the currently loaded file) is set in the Long Message buffer.

Byte	Value	Description
7 ... 0	Any	Name of file to be deleted

Table 139. Delete File definition.

Further, for deleting a file, the offset value is used:

Bits	Value	Definition
14 – 10		Reserved for future use
9 – 8	00b	Delete file
7 – 0		Reserved for future use

Table 140. Offset value for Delete File.

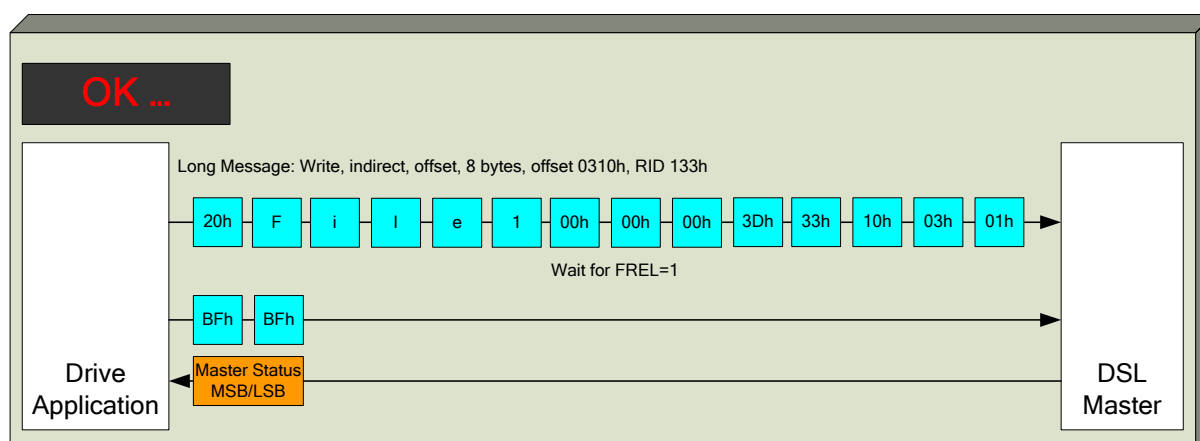


Figure 72. Create File (here: create file with name "File1", read access 0, write access 1).

7.2.6.5. Directory

By accessing this resource a list of existing user files is returned.

A direct read access to Directory returns the resource definition values:

Definition value	Offset	Value
Resource Name	0	"DIR"
Resource Size	1	8
Resource Read Access	2	0
Resource Write Access	3	15
Resource Time-out	4	40

Table 141. Directory definition values.

The directory only lists those files that are accessible with the currently set access level (read or write).

Furthermore, by accessing the Directory, the current amount of used and free user memory can be read.

Please note that due to file headers, user files will usually occupy more physical memory than raw data contents.

The type of data required by the user is set in the offset value during a read access to this resource:

Bits	Value	Definition
14 – 8		Reserved for future use
7 – 0	00h	Return number of files, used and free user memory
	01h	Return name of first user file
	02h	Return name of second user file

	FFh	Return name of 255 th user file

Table 142. Offset value for Directory.

The basic directory data (offset = 00h) is returned in the Long Message buffer as follows:

Byte	Value	Description
7/6		Reserved for future use
5/4	0 – 65535	Number of free bytes in user memory
3/2	0 – 65535	Number of used bytes in user memory
1		Reserved for future use
0	0 – 255	Number of user files

Table 143. Directory definition (basic directory data).

The user file data (offset > 00h) is returned in the Long Message buffer as follows:

Byte	Value	Description
7 – 0	Any	File name

Table 144. Directory definition (user file data).

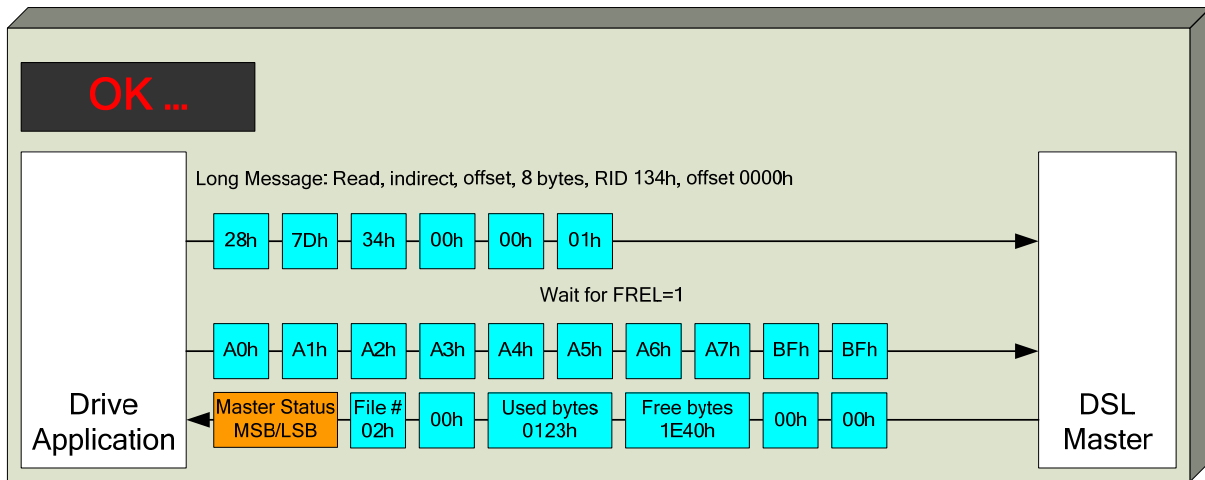


Figure 73. Directory (in this example: Read basic directory data – 2 user files, 123h bytes used, 1E40h bytes free).

Glossary

8B/10B	8 bit / 10 bit code (Line code for transmitting 8 bits of data in 10 bit lengths to achieve DC-balance)
CRC	Cyclic Redundancy Check (Algorithm to determine checksum of data)
DSL	Digital Servo Link, complete name: Hiperface DSL®
FIFO	First in – First out (Storage method where elements stored first are removed first)
FPGA	Field Programmable Gate Array (Programmable digital logic component)
IP-core	Intellectual Property core (Chip design for integration in an IC or FPGA)
Motor-feedback system	Rotary (or linear) encoder for the use in servo drive applications
RS485	Radio Sector Norm 485 (also EIA-485 or TIA-485-A, standard for two-wire differential serial data connection)
RSSI	Received Signal Strength Indicator
SensorHub	Interface from a DSL motor-feedback system to an external sensor component within a drive system
SPI	Serial Peripheral Interface (serial bus system for digital circuits)
VHDL	Very high speed integrated circuit Hardware Description Language (Hardware abstraction language for FPGAs)

Australia

Phone +61 3 9497 4100
1800 33 48 02 – tollfree
E-Mail sales@sick.com.au

Belgium/Luxembourg

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

Brasil

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

Ceská Republika

Phone +420 2 57 91 18 50
E-Mail sick@sick.cz

China

Phone +852-2763 6966
E-Mail ghk@sick.com.hk

Danmark

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

Deutschland

Phone +49 211 5301-301
E-Mail kundenservice@sick.de

España

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

France

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

Great Britain

Phone +44 (0)1727 831121
E-Mail info@sick.co.uk

India

Phone +91-22-4033 8333
E-Mail info@sick-india.com

Israel

Phone +972-4-999-0590
E-Mail info@sick-sensors.com

Italia

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

Japan

Phone +81 (0)3 3358 1341
E-Mail support@sick.jp

Nederlands

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

Norge

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

Österreich

Phone +43 (0)22 36 62 28 8-0
E-Mail office@sick.at

Polska

Phone +48 22 837 40 50
E-Mail info@sick.pl

Republic of Korea

Phone +82-2 786 6321/4
E-Mail info@sickkorea.net

Republika Slovenija

Phone +386 (0)1-47 69 990
E-Mail office@sick.si

România

Phone +40 356 171 120
E-Mail office@sick.ro

Russia

Phone +7 495 775 05 34
E-Mail info@sick-automation.ru

Schweiz

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

Singapore

Phone +65 6744 3732
E-Mail admin@sicksgp.com.sg

Suomi

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

Sverige

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

Taiwan

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

Türkiye

Phone +90 216 587 74 00
E-Mail info@sick.com.tr

United Arab Emirates

Phone +971 4 8865 878
E-Mail info@sick.ae

USA/Canada/México

Phone +1(952) 941-6780
1 800-325-7425 – tollfree
E-Mail info@sickusa.com

More representatives and agencies
in all major industrial nations at
www.sick.com