GEFRAN

# *CAN OPEN DP404 Digital Output*

## *Operating Handbook*

***code*** *85191B - 07-2017_ENG*

**Summary**

# 1. INTRODUCTION

The MD/WD/KD pressure transducer implements the functions of a CAN BUS network slave device conforming to standard CANopen protocol proposed by C.i.A. (Can in Automation) and described in the document entitled “CANOpen Application Layer and Communication Profile DS 301 v. 4.0” dated 16 June 1999 and in other documents mentioned below.

To enable the application with less recent CANopen networks and devices, we guarantee compatibility with the document entitled “CANopen Communication Profile DS 301 v. 3.0” and “CAL.”

The MD transducer conforms to Standard DP 404 v. 2.0 of 11/05/1998.

This document describes the standard CANopen implementations created. It is addressed to CANopen network system integrators and to CANopen device designers who already know the content of the above-mentioned standards defined by CiA. The details of aspects defined by CANopen do not pertain to the purpose of this text.

For further information on the protocol you can also contact us via e-mail: at www.gefran.com or contact the GEFRAN office nearest to you.

Mercury filled transducers MD series are designed and available in compliance with Directive 2011/65/EU (RoHS II) only for large-scale stationary installation or industrial tools, or for B-to-B laboratory equipments for R&D purposes.

## Definition and Shortening

**CAN**: Controller Area Network.
Describes a serial communication bus that implements the “physical” level 1 and the “data link” level 2 of the ISO/OSI reference model.

**CAL**: CAN Application Layer.
Describes implementation of the CAN in the level 7 “application” of the ISO/OSI reference model from which CANopen derives.

**CMS**: CAN Message Specification.
CAL service element. Defines the CAN Application Layer for the various industrial applications.

**COB**: Communication Object.
Unit of transport of data in a CAN network (a CAN message). A maximum of 2048 COBs may be present in a CAN network, each of which may transport from 0 to a maximum of 8 bytes.

**COB-ID**: COB Identifier.
Identifying element of a CAN message. The identifier determines the priority of a COB in case of multiple messages in the network.

**D1 – D8**: Data from 1 to 8.
Number of bytes in the data field of a CAN message.

**DLC**: Data Length code.
Number of data bytes transmitted in a single frame.

**ISO**: International Standard Organization.
International authority providing standards for various merchandise sectors.

**NMT**: Network Management.
CAL service element. Describes how to configure, initialize, manage errors in a CAN network.

**PDO**: Process Data Object.
Process data communication objects (with high priority).

**RXSDO**: Receive SDO.
SDO objects received from the remote device.

**SDO**: Service Data Object.
Service data communication objects (with low priority). The value of this data is contained in the “Objects Dictionary” of each device in the CAN network.

**TXPDO**: Transmit PDO.
PDO objects transmitted by the remote device.

**TXSDO**: Transmit SDO.
SDO objects transmitted by the remote device.

**N.B.**: The numbers followed by the suffix “h” represent a hexadecimal value, with suffix “b” a binary value, and with suffix “d” a decimal value. The value is decimal unless specified otherwise.

## 2. ELECTRICAL CONNECTIONS

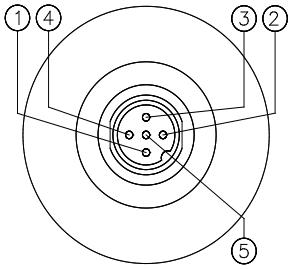
Gefran suggest to use a Belden cable model 3084A or equivalent. For the connections refer to the table below:

| Color | Meaning | Connector M12 |
|---|---|---|
| Red | Supply | 2 |
| Black | Supply GND | 3 |
| White | CAN H | 4 |
| Light Blue | CAN L | 5 |

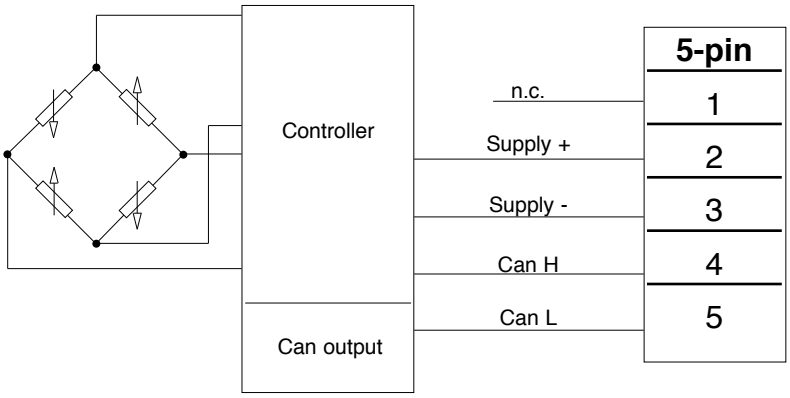
**Note**: please make sure that the CANbus is terminated.

The impedance measured between CAN H and CAN L must be 60 ohm that means the cable must be connected to a 120 ohm resistor on each ends of the bus line.

Internally the transducer is not terminated with the resistor of 120 ohm. Do not confuse the signal lines of the CAN bus, otherwise communication with the transducer is impossible.




**Wiring**




**Note**: the cable shield should be grounded on both ends.

# 3. USING THE TRANSDUCERS

## Baudrate and Node-ID Setting

By default the Baudrate is set to 500 KBps and Node-ID is set to number 127. The selection can be done with a CAN command (see object 2000h and 2001h of the Manufacturer Profile).

The new parameters has effect only after a reset node command or a hardware reset.

In CANopen networks the highest address is limited to 127, also a Node-ID of 0 is not allowed. Every node within a CANopen network must have a unique ID, two nodes with the same ID are not allowed.

The supported baudrates of the transducer are given in the following table. The values are recommended by the CiA.

The Baudrate selection can be done via CAN using the appropriate command (see object 2001h in the Manufacturer Profile); in the table are shown the allowed Baudrate values and the values that have to be transmitted on the Can bus in order to change the Baudrate to the desired value too.

| Baudrate (kBit / s) | Number |
|---|---|
| 1000 | 9 |
| 800 | 8 |
| 500 | 7 |
| 250 | 6 |
| 125 | 5 |
| 100 | 4 |
| 50 | 3 |
| 20 | 2 |
| 10 | 1 |

The maximum cable length depends on the selected baudrate. The following table shows the maximum cable length recommended by CiA.

| Baudrate (kBit / s) | Cable length in m |
|---|---|
| 1000 | 25 |
| 800 | 50 |
| 500 | 100 |
| 250 | 250 |
| 125 | 500 |
| 100 | 650 |
| 50 | 1000 |
| 20 | 2500 |
| 10 | 5000 |

## Diagnosis in Pre-Operational State

After power-on the LED present on the top of the transducer will be turned on and off for a short moment and turned on again.

| Status of the LED | Meaning |
|---|---|
| Quick flashing (2 times per second) | Device is in Pre-Operational state |
| Off | Supply voltage not ok, hardware not ok |
| Short Flashes every second (10ms flashes) | Input is left open or wrong sensor type |
| On | Supply Voltage ok but no CAN connection; maybe the wrong baudrate is selected or any other physical error on the bus. |

## Diagnosis in Operational State

With the help of the LED the communication on the CAN bus as well as the state of the sensor input can be supervised.

| Status of the LED | Meaning |
|---|---|
| Quick flashing (2 times per second) | Device is in Pre-Operational state |
| Slow blinking (1 Hz) | Device is in Operational State and there is communication with the device via CANbus |
| Short Flashes every second (10ms flashes) | Input is left open or wrong sensor type |

## Diagnosis in Stop State

| Status of the LED | Meaning |
|---|---|
| On | Device is in Stop State |

# 4. CANOpen PROTOCOL

## Communication Object

In order to reduce configuration effort for simple networks a default identifier allocation scheme is defined.

The Predefined Connection Set uses a standard CAN frame with 11 bits identifier (COB-ID) which consists of 4 bit of Function Code (which determines the object priority) and 7 bit of Node-ID (which allows to uniquely identify the node).

If the COB-ID is equal to zero the message is broadcast that means it is applied to all the nodes in the network.

**COB-ID = Function Code (4 Bit) + Module-ID (7 Bit)**

| Object | Function Code (bin) | COB-ID Hex | COB-ID Dec |
|---|---|---|---|
| NMT | 0000 | 000 | 0 |
| SYNC | 0001 | 080 | 128 |
| EMERGENCY | 0001 | 081-0FF | 129-255 |
| TX PDO1 | 0011 | 181-1FF | 385-511 |
| TX PDO2 | 0101 | 281-2FF | 641-767 |
| TX SDO | 1011 | 581-5FF | 1409-1535 |
| RX SDO | 1100 | 601-67F | 1537-1663 |
| HEART BEAT | 1110 | 701-77F | 1793-1919 |

## Network Management

The Network Management (NMT) is node oriented and follows a Master / Slave structure. NMT objects are used for executing NMT services. Through NMT services nodes are initialised, started, monitored, resetted or stopped.

All nodes are regarded as Slaves. Every node is uniquely identified in the network by its Node-ID ( 1 to 127 ). The possible operative states are:

- Initialization
- Pre – Operational
- Operational
- Stopped

The following table shows the relation between communication states and communication objects. Services on the listed communication objects may only be executed if the devices involved in the communication are in the appropriate communication states.

| Object | Initialization | Pre-Operational | Operational | Stop |
|---|---|---|---|---|
| Network Management | | X | X | |
| SYNC | | X | X | |
| EMERGENCY | | X | X | |
| PDO | | | X | |
| SDO | | X | X | |
| Heart Beat Protocol | | X | X | X |
| Boot up | X | | | |

The **Initialization** state is divided into three sub-states:

1. INIT: this is the first sub-state the device enters after a power-on or hardware reset.
   After finishing the basic node initialisation the device executes the write boot-up object service and enters the state Pre-Operational autonomously. Power-on values are the last stored parameters.
2. RESET APPLICATION: In this state the parameters of the manufacturer specific profile area and of the standardised device profile area are set to their power-on values. After setting of the power-on values the state Reset Communication is entered autonomously.
3. RESET COMMUNICATION: In this state the parameters of the communication profile area are set to their power-on values. After this the state Init is entered autonomously.

In the **Pre-Operational** state communication via SDO is possibile. PDO communication is not allowed. The node may be switched into the operational state directly by sending a start remote node.

In the **Operational** state all communication objects are active. Object Dictionary Access via SDO is possible and the node can handle PDO-communication.

In the **Stopped** state the device stop the communication altogether. If active is possible the heart beat trasmission.

To modify the operative state of a node on the network, the Master sends on the CAN network the following message:

| COB-ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | CS | Node-ID | X | X | X | X | X | X |

**CS** = Command Specifier, is the code that corresponds to the operative state.

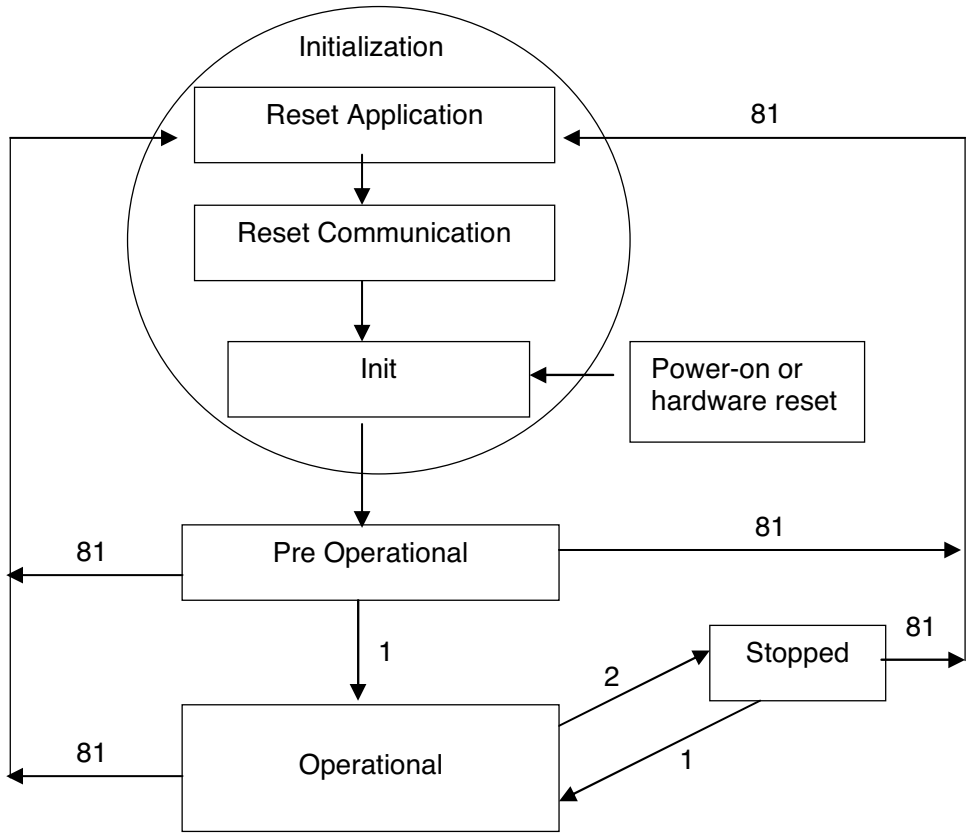
| CS | Operative State |
|---|---|
| 1 | Operational |
| 2 | Stopped |
| 80 | Pre-Operational |
| 81 | Reset Application |
| 82 | Reset Communication |

**Node-ID** = is the number that uniquely identify a device on the network. If the Node-Id is equal to zero the command is sent to all nodes presents on the network.

After power-up the transducer transmits a “Boot-Up Message”. This message does not contain any information except the “Heart Beat”-ID which is by default 700h (1792d) + Node-ID.

| COB-ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 700+Node-ID (1792d) | 1 | 0 | X | X | X | X | X | X | X |

The flow chart shows the states diagram of a device:



## Heart Beat Protocol

On switching on the Heart Beat protocol any other node on the CAN-bus can survey the sending node (Heart Beat producer). This technique is used to monitor the active nodes on the bus for safety reasons.

For example nodes which send data autonomously only every 2 minutes, it can be better for the master to survey the node in the meantime.

The Identifier for the Heart Beat producer has the following structure:

| COB-ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 700+Node-ID (1792d) | 1 | S | X | X | X | X | X | X | X |

The COB-ID can not be changed.

The parameter "**S**" represents the operative state of the Slave:

| S | Operative State |
|---|---|
| 00h | Boot-up |
| 04h | Stop |
| 05h | Operational |
| 7Fh | Pre-Operational |

The delay for sending the Heart Beat message is selected under Index 1017h (see the Communication profile).

## SYNC

Synchronous transmission of a message means that the transmission of the P.D.O is fixed in time with respect to the transmission of the SYNC message.

The Synchronisation Object is broadcasted periodically by the SYNC producer. In order to guarantee timely access to the CAN bus the SYNC is given a very high priority identifier (see object 1005h of the Communication profile).

The SYNC has the following structure:

| COB-ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 80 (128d) | 0 | X | X | X | X | X | X | X | X |

As defined by the synchronous transmission, this message affect only the nodes with the parameter "Trasmission Type" (see object 1800h and 1801h of the Communication Profile) between 0d to 240d.

## P.D.O. Communication

The PDO (Process Data Objects) communication service is a method to receive all information from the node in one CAN message. PDO communication with the transducer is only possible in Operational Mode.

The transducer uses a fixed PDO-Mapping, the first PDO contains the pressure value and the input state and the second PDO contains the A/D values, the operating time and the maximum peak measure.

The PDO 1 is by default switched on, the PDO 2 is normally switched off.

**PDO 1 structure: (default = ON)**

| COB-ID | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|
| 180+Node-ID (384d) | X | X | X | X | X | X | X | X |
| | Pressure value (object 7130h) | | Analog Input status (object 6150h) | Not used | Not used | | Not used | |

**PDO 2 structure: (default = OFF)**

| COB-ID | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|
| 280+Node-ID (640d) | X | X | X | X | X | X | X | X |
| | A/D values (object 7100h) | | Operating time (object 2230h sub 1) | | | | Maximum peak (object 2220h sub 1) | |

## S.D.O.

With Service Data Object (S.D.O.) the access to entries of a device Object Dictionary is provided. As these entries may contain data of arbitrary size and data type SDOs can be used to transfer multiple data sets from a client to a server and vice versa.

The contents of the data are defined within the Object Dictionary.

**Structure of SDO-request by the Master**

| COB-ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 600+Node-ID (1536d) | 8 | CMD | Index | | Sub-Index | Data | | | |

**Structure of SDO-answer by the Slave**

| COB-ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 580+Node-ID (1408d) | 8 | CMD | Index | | Sub-Index | Data | | | |

The Command Byte **CMD** has the following meaning:

| CMD | Meaning |
|---|---|
| 40h | Master wants to read from Slave |
| 42h | Slave answers on the read-request |
| 22h | Master wants to write to Slave |
| 60h | Slave answers on the write-request |

The bytes order for the fields "**Index**" and "**Data**" is least significant byte first (Intel format), except for "String" data type (see SDO dictionary).

The minimum time delay between two succeeding SDO-commands should be at least 20ms.

Faster communication might lead to an unpredictible device status.

## S.D.O. Error Messages

The access to an unsupported object (index) leads to an SDO-Error Message. This SDO-Error Message has the following format:

**PDO 1 structure: (default = ON)**

| COB-ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| | 8 | 80h | Index | | Sub-Index | Additional Code | | Error Code | Error Class |

The COB-ID as well as the Index and Sub-Index correspond to the SDO request.

The error messages may have the following contents:

| Error Class | Error Code | Additional Code | Meaning |
|---|---|---|---|
| 05h | 04h | 00h 01h | Client/Server command specifier not valid or unknown |
| 06h | 01h | 00h 01h | Attempt to read a write only object |
| 06h | 01h | 00h 02h | Attempt to write a read only object |
| 06h | 09h | 00h 11h | Sub-index does not exist |
| 06h | 02h | 00h 00h | Object does not exist |

## Emergency Message

Emergency objects are triggered by the occurrence of a device internal error situation and are transmitted from an emergency producer on the device.

An emergency is different from a SDO Error Message. The last one only holds the access error to the object dictionary, whereas an emergency display a severe hardware/software failure.

The emergency message has the following structure:

| COB-ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 80+Node-ID(128d) | 8 | Error Code | | 00h | Manufacturer Specific Error Field | | | | |

Supported Error Codes:

| Error Code | Meaning |
|---|---|
| 00 00h | No Error |
| FF 00h | Device Specific Error |
| 81 00h | Communication Error |

**Device Specific Error**

Sensor break or a short-circuit of sensor

Byte 4 (first byte of the Manufacturer Specific Error Field) holds the information about the specific error type.

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | | Input 1 | | | | Input 1 |
| Short-Circuit of Sensor | | | | Sensor break | Sensor break | Sensor break | Sensor break |

**Communication Error**

The transducer supports the generation of communication errors. For this error type the internal state of the CAN controller is monitored.

Byte 4 (first byte of the Manufacturer Specific Error Field) holds the value BEh in case of a CAN-bus error.

The value of Byte 5 defines the error type.

Description of byte 5:

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| Status | | | | | | | |

**Status**:

**b7** - Bit is set **Bus Off Status**
The CAN controller is in bus-off state because the error counters reached the maximum value.

**b6** - Bit is set **Warning Status**
The CAN controller is in warning status.

## Object Dictionary

The most important part of a device profile is the Object dictionary description. The Object dictionary is essentially a grouping of objects accessible via the network in an ordered pre-defined fashion.

Each object within the dictionary is addressed using a 16 bit index. The overall layout of the standard Object dictionary is shown below.

This layout closely conforms with other industrial serial bus system concepts.

| Index (hex) | Object |
|---|---|
| 0000 | Not used |
| 0001-001F | Static Data Types |
| 0020-003F | Complex Data Types |
| 0040-005F | Manufacturer Specific Complex Data Types |
| 0060-007F | Device Pofile Specific Static Data Types |
| 0080-009F | Device Profile Complex Data Types |
| 00A0-0FFF | Reserved for further use |
| 1000-1FFF | Communication Profile Area |
| 2000-5FFF | Manufacturer Specific Profile Area |
| 6000-9FFF | Standardised Device Profile Area |
| A000-FFFF | Reserved for further use |

## Communication Profile

The parameters which are critical for communication are determined in the Communication profile. This area is common for all CANopen devices.

| Index | Sub Index | Name | Type | Access | Default value | Pag |
|---|---|---|---|---|---|---|
| 1000h | 0 | Device Profile | Unsigned 32 | Ro | Value for DP404 | 13 |
| 1001h | 0 | Error Register | Unsigned 8 | Ro | 0 | 13 |
| 1005h | 0 | COB-ID SYNC-Message | Unsigned 32 | Rw | 80h | 14 |
| 1008h | 0 | Manufacturer Device Name | String | Ro | GSWD | 14 |
| 1009h | 0 | Manufacturer Hardware Version | String | Ro | 1.01 | 15 |
| 100Ah | 0 | Manufacturer Software Version | String | Ro | 2.01 | 15 |
| 1010h | 0 | Store Parameter | Unsigned 8 | Ro | 1 | 16 |
| | 1 | Store all Parameters | Unsigned 32 | Rw | | |
| 1011h | 0 | Restore Default Parameters | Unsigned 8 | Ro | 1 | 16 |
| | 1 | Restore all Parameters | Unsigned 32 | Rw | | |
| 1014h | 0 | Emergency ID | Unsigned 32 | Rw | 80h + Node-ID | 17 |
| 1017h | 0 | Producer Time / Heart Beat | Unsigned 16 | Rw | 0 | 17 |
| 1018h | 0 | Identity Object | Unsigned 8 | Ro | 4 | 18 |
| | 1 | Vendor ID | Unsigned 32 | Ro | 93h | |
| | 2 | Product Code | Unsigned 32 | Ro | WD0 | |
| | 3 | Revision Number | Unsigned 32 | Ro | 00010001h | |
| | 4 | Serial Number | Unsigned 32 | Ro | yy / ww / s.n. h | |
| 1800h | 0 | 1st Transmit PDO Parameter | Unsigned 8 | Ro | 5 | 19 |
| | 1 | COB-ID | Unsigned 32 | Rw | 180h + Node-ID | |
| | 2 | Transmission Type | Unsigned 8 | Rw | FEh | |
| | 3 | Reserved | // | // | | |
| | 4 | Reserved | // | // | | |
| | 5 | Timer | Unsigned 16 | Rw | 14h | |
| 1801h | 0 | 2nd Transmit PDO Parameter | Unsigned 8 | Ro | 5 | 21 |
| | 1 | COB-ID | Unsigned 32 | Rw | 280h + Node-ID | |
| | 2 | Transmission Type | Unsigned 8 | Rw | FEh | |
| | 3 | Reserved | // | // | | |
| | 4 | Reserved | // | // | | |
| | 5 | Timer | Unsigned 16 | Rw | 14h | |
| 1A00h | 0 | 1st Transmit PDO Mapping | Unsigned 8 | Ro | 2 | 23 |
| | 1 | 1st Mapped Object | Unsigned 32 | Ro | 71300110h | |
| | 2 | 2nd Mapped Object | Unsigned 32 | Ro | 61500108h | |
| 1A01h | 0 | 2nd Transmit PDO Mapping | Unsigned 8 | Ro | 3 | 23 |
| | 1 | 1st Mapped Object | Unsigned 32 | Ro | 71000110h | |
| | 2 | 2nd Mapped Object | Unsigned 32 | Ro | 22300120h | |
| | 3 | 3th Mapped Object | Unsigned 32 | Ro | 22200108h | |

Ro = the parameter can be read only
Rw = the parameter can be read and also written
Wo = the parameter can be written only

**1000h Device Profile**

By a read-access on Index 1000h the Device Profile can be polled.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 1000h | 0 | Device Profile | Ro | Unsigned 32 |

**Example**: Read-Access, Node-ID = 2, Index 1000h, Sub-Index = not supported within this Index

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 00h | 10h | 00h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer is:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 00h | 10h | 00h | 94h | 01h | 22h | 00h |

Byte 5 + Byte 6 = 0194h = 404d (Device Profile Number)
Byte 7 + Byte 8 = 0022h = 22 (Additional Information) - Analog Input + Alarm function

Index 1000h is Read-Only, no Sub-Indices are supported. By writing on this Index (or reading a Sub-Index unequal to “0”) you will receive a SDO-Communication Error (see “SDO-Error Messages” ).

**1001h Error Register**

By a read-access on Index 1001h the state of the error register can be polled

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 1001h | 0 | Error Register | Ro | Unsigned 8 |

**Example**: Read-Access, Node-ID = 2, Index 1001h, Sub-Index = not supported within this Index

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 01h | 10h | 00h | 00h | 00h | 00h | 00h |

As answer you will receive on byte 5 the Error status of the device. The following errors are supported:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 01h | 10h | 00h | 00h | 00h | 00h | 00h |

**No Error: no bits set in Byte 5**

**Generic Error: Bit 1 in Byte 5**
The Generic Error is generated by the signal input of the transducer. This error means: Break or Short-Circuit of the connected sensor or a positive overload.

**Communication Error: Bit 4 in Byte 5**
The Communication Error will be generated by (physical) errors on the CAN-network. A complete list of supported Communication Errors can be seen under “Emergency Message”.

Index 1001h is Read-Only, no Sub-Indices are supported. By writing to this Index (or reading a Sub-Index unequal to “0”) you will receive a SDO-Communication Error (see “SDO-Error Messages” ).

**Index 1005h COB-ID SYNC Message**

The object 1005h defines the identifier (COB) for the Synchronisation Object (SYNC). On reception of a message with this identifier the transmission of PDOs is initiated (refer to “PDO-Communication”).

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 1005h | 0 | COB-ID SYNC Message | Rw | Unsigned 32 |

The COB-ID SYNC identifier has the following structure:

| Bit 31 | Bit 30 | Bit 29 | Bit 11-28 | Bit 10-0 |
|---|---|---|---|---|
| 1 | X | 0 | 0 | 11 bit Identifier |

**Example**: Read-Access, Node-ID = 2, Index 1005h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 05h | 10h | 01h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 05h | 10h | 01h | 80h | 00h | 00h | 80h |

The default identifier is 80h in order to ensure a high priority of the SYNC-message.

A change of this parameter will not be stored inside the EEPROM automatically. This has to be done manually by the user (see “Store All Parameters”).

**1008h Manufacturer Device Name**

By a read-access on Index 1008h the Manufacturer Device Name can be polled.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 1008h | 0 | Manufacturer Device Name | Ro | String |

**Example**: Read-Access, Node-ID = 2, Index 1008h, Sub-Index = not supported within this Index

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 08h | 10h | 00h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer a message with ASCII-coded hardware version:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 08h | 10h | 00h | 47h | 53h | 57h | 44h |

Byte 5 = 47h represents ASCII = G
Byte 6 = 53h represents ASCII = S
Byte 7 = 57h represents ASCII = W
Byte 8 = 44h represents ASCII = D

Index 1008h is Read-Only, no Sub-Indices are supported. By writing on this Index (or reading a Sub-Index unequal to “0”) you will receive a SDO-Communication Error (see “SDO-Error Messages” ).

**1009h Manufacturer Hardware Version**

By a read-access on Index 1009h the Manufacturer Hardware Version can be polled.
The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 1009h | 0 | Manufacturer Hardware Version | Ro | String |

**Example**: Read-Access, Node-ID = 2, Index 1009h, Sub-Index = not supported within this Index

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 09h | 10h | 00h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer a message with ASCII-coded Hardware version.

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 09h | 10h | 00h | 31h | 2Eh | 30h | 31h |

Byte 5 = 31h represents ASCII = 1
Byte 6 = 2Eh represents ASCII = .
Byte 7 = 30h represents ASCII = 0
Byte 8 = 31h represents ASCII = 1

Index 1009h is Read-Only, no Sub-Indices are supported. By writing on this Index (or reading a Sub-Index unequal to “0”) you will receive a SDO-Communication Error (see “SDO-Error Messages” on 8.4.2).

**100Ah Manufacturer Software Version**

By a read-access on Index 100Ah the Manufacturer Software Version can be polled.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 100Ah | 0 | Manufacturer Software Version | Ro | String |

**Example**: Read-Access, Node-ID = 2, Index 100Ah, Sub-Index = not supported within this Index

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 0Ah | 10h | 00 | 00 | 00 | 00 | 00 |

As answer you will receive from the transducer a message with ASCII-coded Software version:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 0Ah | 10h | 00h | 32h | 2Eh | 30h | 31h |

Byte 5 = 31h represents ASCII = 2
Byte 6 = 30h represents ASCII = .
Byte 7 = 2Eh represents ASCII = 0
Byte 8 = 32h represents ASCII = 1

Index 100Ah is Read-Only, no Sub-Indices are supported. By writing on this Index (or reading a Sub-Index equal to 0”) you will receive a SDO-Communication Error (see “SDO-Error Messages”).

**1010h Store All Parameters**

By writing on Index 1010h all parameters changed are stored on the EEPROM
The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 1010h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Store all Parameters | Rw | Unsigned 32 |

The storing will be generated by transmitting the code “save” as ASCII in the data area of Index 1010h. The message has the following format:

**Example**: Write-Access, Node-ID = 2, Index 1010h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 22h | 10h | 10h | 01h | 73h | 61h | 76h | 65h |

The answer after successful storing you will receive is:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 60h | 10h | 10h | 01h | 00 | 00 | 00 | 00 |

**1011h Restore Default Parameter**

By writing on Index 1011h the default parameters can be loaded.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 1011h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Restore Default Parameters | Rw | Unsigned 32 |

The default parameters values are shown in the list below.

| Index | Name | Default |
|---|---|---|
| 6112h | Operating Mode | 1 (Channel On) |
| 6144h | Analog Input Lower limit | 0 |
| 6145h | Analog Input Upper limit | 0 |
| 2000h | Node – ID | 127d (or according to customer order) |
| 2001h | Baudrate setting | 7d (500KB) (or according to customer order) |
| 6509h | Alarm 1 Action | 0 |
| 6519h | Alarm 2 Action | 0 |
| 750Ah | Alarm 1 Level | 0 |
| 750Bh | Alarm 1 Hysteresis | 0 |
| 751Ah | Alarm 2 Level | 0 |
| 751Bh | Alarm 2 Hysteresis | 0 |
| 7124h | Input Offset | 0 |
| 1014h | Emergency ID | 80h + ID |
| 1017h | Producer Time / Heart Beat | 0 ms |
| 1005h | SYNC-ID | 80 h |
| 1800h | PDO 1 – Parameter | PDO active, ID 180h + Node-ID,Transmission type FEh, Timer 14h (or according to customer order) |
| 1801h | PDO 2 – Parameter | PDO not active, ID 280h + Node-ID,Transmission type FEh, Timer 14h (or according to customer order) |

The loading will be generated by transmitting the code “load” as ASCII in the data-area of Index 1011h.

The message has to look like the following example:

**Example**: Write-Access, Node-ID = 2, Index 1011h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 22h | 11h | 10h | 01h | 6Ch | 6Fh | 61h | 64h |

The answer after successful loading you will receive is:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 60h | 11h | 10h | 01h | 00 | 00 | 00 | 00 |

**Index 1014h Emergency ID**

By accessing this object the ID of the Emergency message can be changed.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 1014h | 0 | Emergency ID (32 bit) | Rw | Unsigned 32 |

The Emergency-ID has the following structure:

| Bit 31 | Bit 30 | Bit 29 | Bit 11-28 | Bit 10-0 |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 11 bit Identifier |

**Example**: Read-Access, Node-ID = 2, Index 1014h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 14h | 10h | 01h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 14h | 10h | 01h | 82h | 00h | 00h | 00h |

Byte 5 holds the Emergency ID. By default the ID is set to 80h + module address.
This ID reserves high priority for the emergency messages.

A change of this parameter will not be stored inside the EEPROM automatically. This has to be done manually by the user (see “Store All Parameters”).

**Index 1017h Producer Time / Heart Beat**

Within this Index the timer for the transmission of Heart Beat messages can be changed.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 1017h | 0 | Timer in ms (16 bit) | Rw | Unsigned 16 |

By default the timer is set to 0. When changing the timer to a value greater 5 ms the transducer will begin to send Heart Beat messages autonomously.

The timer is a ms-timer with multiples of 5 ms. Any value unequal to a multiple of 5 written to the module will be rounded. The value is a 16-bit value.

If for example written to the module a value of 112 ms, this value will be changed to 110 ms.

**Example**: Write parameter, Module-ID = 2, Index 1017h, Sub-Index = not supported, Timer = 1000 ms

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 22h | 17h | 10h | E8h | 03h | 00h | 00h | 00h |

Data bytes 4+5 contain the value 03E8h which represents a producer time of 1000 ms.

The answer you will receive from the module is:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 60h | 17h | 10h | 00 | 00 | 00 | 00 | 00 |

A change of these parameters will not be stored inside the EEPROM automatically. This has to be done manually by the user (see “Store All Parameters”).

**1018h Identity Object**

Index 1018h contains the so called “Identity Object” which includes several parameters.

The Index has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 1018h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Vendor ID | Ro | Unsigned 32 |
| | 2 | Product Code | Ro | Unsigned 32 |
| | 3 | Revision Number | Ro | Unsigned 32 |
| | 4 | Serial Number | Ro | Unsigned 32 |

**Sub-Index 1 Vendor ID**
The vendor ID is an unique number which can be used to identify the manufacturer of the device.
The numbers are managed by the CiA (http://www.can-cia.de) worldwide. Gefran Sensori has the ID = 93h.

**Example**: Read-Access, Node-ID = 2, Index 1018h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 18h | 10h | 01h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 18h | 10h | 01h | 93h | 00h | 0h | 00h |

The Byte five holds the Vendor ID code.

**Sub-Index 2 Product Code**

The product code is a manufacturer specific code, which in our case represents the model of the transducer (ex. WD0).

**Example**: Read-Access, Node-ID = 2, Index 1018h, Sub-Index = 02h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 18h | 10h | 02h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer a message with ASCII-coded hardware version:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 18h | 10h | 02h | 00h | 30h | 44h | 57h |

Byte 6 = 30h represents ASCII = 0
Byte 7 = 44h represents ASCII = D
Byte 8 = 57h represents ASCII = W

The transducer is the model WD0.

**Sub-Index 3 Revision Number**

The revision number consits of two 16bit values. The higher 16 bits represents the revision of the CANopen parts of the software and the lower 16 bits represents the general firmware release.

**Example**: Read-Access, Node-ID = 2, Index 1018h, Sub-Index = 03h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 18h | 10h | 03h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 18h | 10h | 03h | 02h | 00h | 02h | 00h |

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 1017h | 0 | Timer in ms (16 bit) | Rw | Unsigned 16 |

**Sub-Index 4 Serial Number**

The serial number is also manufacturer specific and represents the coded date, on which the module was checked and calibrated.

**Example**: Read-Access, Node-ID = 2, Index 1018h, Sub-Index = 04h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 18h | 10h | 04h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 18h | 10h | 04h | 02h | 15h | 10h | 02h |

The transducer serial number is: 02165378

Index 1018h is Read-Only. By writing on this Index (or reading an unsupported Sub-Index) you will receive a SDO - Communication Error (see “SDO-Error Messages”).

**Index 1800h PDO 1 Parameters**

The object 1800h defines communication parameters for the first transmit PDO. The PDO can only be used in Operational Mode.

The PDO may not be requested faster than 1ms by the SYNC-service, because new pressure values for the input is available every 1ms.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 1800h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Identifier of PDO 1 (COB-ID) | Rw | Unsigned 32 |
| | 2 | Transmission Type | Rw | Unsigned 8 |
| | 3 | Reserved | | |
| | 4 | Reserved | | |
| | 5 | Timer | Rw | Unsigned 16 |

**Sub-Index 1 Identifier of PDO 1**

The default identifier is 180h + module-address. This value defines the identifier that is used for PDO 1.

The 32-bit value has the following structure:

| Bit 31 | Bit 30 – 11 | Bit 10 – 0 |
|---|---|---|
| 0 / 1 | 0 | ID 11 bit |

In order to enable the PDO the most significant bit (Bit 31) must be set to 0. In order to disable the PDO the most significant bit must be set to 1.

In the default setting the PDO is active (Bit 31 = 0).

**Example**: Read-Access, Node-ID = 2, Index 1800h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 00h | 18h | 01h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 00h | 18h | 01h | 82h | 01h | 00h | 00h |

The identifier of the PDO 1 is 00000182h (PDO active)

**Sub-Index 2 Transmission Type**

The transmission type defines the transmission type of the PDO. The device has three different types of transmission.

By writing to Sub-Index 02 the transmission type can be selected.

The following types are supported:

| Transmission Type | Description |
|---|---|
| 0<br>(0h) | Acyclic synchron<br>The module sends on every SYNC-Message |
| 1 – 240d<br>(1- F0h) | Cyclic synchron<br>The module sends on every n-th SYNC-Message (with n = 1 .. 240) |
| 254d<br>(FEh) | Manufacturer specific<br>The module sends autonomously every x ms.<br>The value for x can be selected in Sub-Index 05. |

If selected Transmission Type 254d (manufacturer specific) and the value for transmission delay (Sub-Index 05) is greater than 5ms, the module will begin to transmit PDOs after receiving a “Start Node”-command from the master.

**Example**: Read-Access, Node-ID = 2, Index 1800h, Sub-Index = 02h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 00h | 18h | 02h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 00h | 18h | 02h | FEh | 00h | 00h | 00h |

The transmission type of the PDO 1 is FEh = 254d

**Sub-Index 3 Reserved**
**Sub-Index 4 Reserved**

**Sub-Index 5 Timer**

As mentioned before, the transmission timer for PDOs can be selected under Sub-Index 05. The timer is a ms-timer with multiples of 5ms.

Any value unequal to a multiple of 5 written to the module will be rounded. The value is a 16-bit value.

If for example written to the module a value of 112(ms), this value will be changed to 110(ms).

Only in Operational Mode the module will send PDOs with timer delay.

**Example**: Read-Access, Node-ID = 2, Index 1800h, Sub-Index = 05h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 00h | 18h | 05h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 00h | 18h | 05h | 14h | 00h | 00h | 00h |

The transmission timer of the PDO 1 is 14h = 20 ms

A change of these parameters will not be stored inside the EEPROM automatically. This has to be done manually by the user (see “Store All Parameters”).

**Index 1801h PDO 2 Parameters**

The object 1801h defines communication parameters for the second transmit PDO. The PDO can only be used in Operational Mode.

The PDO may not be requested faster than 1ms by the SYNC-service. It is recommended to request the PDO not faster than 1ms, because new A/D converter values for the input is available every 1ms.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 1801h | 0 | Largest Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Identifier of PDO 2 (COB-ID) | Rw | Unsigned 32 |
| | 2 | Transmission Type | Rw | Unsigned 8 |
| | 3 | Reserved | | |
| | 4 | Reserved | | |
| | 5 | Timer | Rw | Unsigned 16 |

**Sub-Index 1 Identifier of PDO 2**

The default identifier is 280h + module-address. The value on Sub-Index 1 defines the identifier that is used for PDO 2.

The 32-bit value has the following structure:

| Bit 31 | Bit 30 – 11 | Bit 10 – 0 |
|---|---|---|
| 0 / 1 | 0 | ID 11 bit |

In order to enable the PDO the most significant bit (Bit 31) must be set to 0. In order to disable the PDO the most significant bit must be set to 1.

In the default setting of the PDO is **not active** (Bit 31 = 1).

**Example**: Read-Access, Node-ID = 2, Index 1801h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 01h | 18h | 01h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 01h | 18h | 01h | 82h | 02h | 00h | 80h |

The identifier of the PDO 2 is 80000282h (PDO not active)

**Sub-Index 2 Transmission Type**

The transmission type (Sub-Index 2) defines the transmission type of the PDO. The device has three different types of transmission.

By writing to Sub-Index 02 the transmission type can be selected.

The following types are supported:

| Transmission Type | Description |
|---|---|
| 0<br>(0h) | Acyclic synchron<br>The module sends on every SYNC-Message |
| 1 – 240d<br>(1- F0h) | Cyclic synchron<br>The module sends on every n-th SYNC-Message (with n = 1 .. 240) |
| 254d<br>(FEh) | Manufacturer specific<br>The module sends autonomously every x ms.<br>The value for x can be selected in Sub-Index 05. |

If selected Transmission Type 254d (manufacturer specific) and the value for transmission delay (Sub-Index 05) is greater 5ms, the module will begin to transmitt PDOs after receiving a “Start Node”-command from the master.

**Example**: Read-Access, Node-ID = 2, Index 1801h, Sub-Index = 02h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 01h | 18h | 02h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 01h | 18h | 02h | FEh | 00h | 00h | 00h |

The transmission type of the PDO 2 is FEh = 254d

**Sub-Index 3 Reserved**
**Sub-Index 4 Reserved**

**Sub-Index 5 Timer**

As mentioned before, the transmission timer for PDOs can be selected under Sub-Index 05. The timer is a ms-timer with multiples of 5ms.

Any value unequal to a multiple of 5 written to the module will be rounded. The value is a 16-bit value. If for example written to the module a value of 112(ms), this value will be changed to 110(ms).

Only in Operational Mode the module will send PDOs with timer delay.

**Example**: Read-Access, Node-ID = 2, Index 1801h, Sub-Index = 05h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 01h | 18h | 05h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 01h | 18h | 05h | 14h | 00h | 00h | 00h |

The transmission timer of the PDO 2 is 14h = 20 ms

A change of these parameters will not be stored inside the EE¬PROM automatically. This has to be done manually by the user (see “Store All Parameters”).

**Index 1A00h PDO 1 Mapping**

The object 1A00h defines the PDO mapping for the first PDO. The transducer uses a fixed PDO-Mapping.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 1A00h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | 1st object | Ro | Unsigned 32 |
| | 2 | 2st object | Ro | Unsigned 32 |

The first object mapped is the Process Values (object 7130h)
The second object mapped is the Analog Input Status (object 6150h)

**Index 1A01h PDO 2 Mapping**

The object 1A01h defines the PDO mapping for the second PDO. The transducer uses a fixed PDO-Mapping.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 1A01h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | 1st object | Ro | Unsigned 32 |
| | 2 | 2st object | Ro | Unsigned 32 |
| | 3 | 3st object | Ro | Unsigned 32 |

The CAN message for the first PDO has the following structure:

The first object mapped is the **A/D value (object 7100h)**
The second object mapped is the **Operating Time (object 2230h)**
The third object mapped is the **Maximum Peak (object 2220h)**

## Device Profile Objects

In this section you will find all device profile specific indices for the transducer. These indices are implemented according to the DS-404 device profile.

| Index | Sub Index | Name | Type | Access | Default value | Pag |
|---|---|---|---|---|---|---|
| 6110h | 0 | Sensor Type (SubIndex SUPP.) | Unsigned 8 | Ro | 1 | 25 |
| | 1 | Parameter of the sensor type | Unsigned 16 | Ro | 5Ah = 90d = Pressure transducer | |
| 6112h | 0 | Op. Mode (SubIndex SUPP.) | Unsigned 8 | Ro | 1 | 25 |
| | 1 | Operating Mode | Unsigned 8 | Rw | 1 | |
| 6125h | 0 | Autozero (SubIndex SUPP.) | Unsigned 8 | Ro | 1 | 26 |
| | 1 | Autozero | Unsigned 32 | Wo | | |
| 6131h | 0 | Unit (SubIndex SUPP.) | Unsigned 8 | Ro | 1 | 26 |
| | 1 | Physical Unit Process Value | Unsigned 32 | Rw | 2310h (Bar) | |
| 6132h | 0 | Digits (SubIndex SUPP.) | Unsigned 8 | Ro | | 27 |
| | 1 | Decimal Digits Process Value | Unsigned 8 | Rw | Depends on FS (0,1 or 2) | |
| 6144h | 0 | Lower Limit (SubIndex SUPP.) | Unsigned 8 | Ro | 1 | 27 |
| | 1 | Analog Input Lower limit | Unsigned 16 | Rw | 0 | |
| 6145h | 0 | Upper Limit (SubIndex SUPP.) | Unsigned 8 | Ro | 1 | 27 |
| | 1 | Analog Input Upper limit | Unsigned 16 | Rw | 0 | |
| 6150h | 0 | Status (SubIndex SUPP.) | Unsigned 8 | Ro | 1 | 28 |
| | 1 | Analog Input Status | Unsigned 8 | Ro | 0 | |
| 6508h | 0 | Alarm 1 Type(SubIndex SUPP.) | Unsigned 8 | Ro | 1 | 29 |
| | 1 | Alarm 1 Type | Unsigned 16 | Ro | 2h (above or equal) | |
| 6509h | 0 | Alarm 1 Act. (SubIndex SUPP.) | Unsigned 8 | Ro | 1 | 29 |
| | 1 | Alarm 1 Action | Unsigned 16 | Rw | 0 | |
| 6518h | 0 | Alarm 2 Type(SubIndex SUPP.) | Unsigned 8 | Ro | 1 | 29 |
| | 1 | Alarm 2 Type | Unsigned 16 | Ro | 3h (below) | |
| 6519h | 0 | Alarm 2 Act. (SubIndex SUPP.) | Unsigned 8 | Ro | 1 | 29 |
| | 1 | Alarm 2 Action | Unsigned 16 | Rw | 0 | |
| 7100h | 0 | ADC value (SubIndex SUPP.) | Unsigned 8 | Ro | 1 | 30 |
| | 1 | ADC value | Unsigned 16 | Ro | | |
| 7124h | 0 | Input Offset (SubIndex SUPP.) | Unsigned 8 | Ro | 1 | 30 |
| | 1 | Input Offset | Unsigned 16 | Rw | 0 | |
| 7130h | 0 | ProcessValue(SubIndex SUPP.) | Unsigned 8 | Ro | 1 | 31 |
| | 1 | Analog Input Process Value | Unsigned 16 | Ro | | |
| 7500h | 0 | Alarm1 val (SubIndex SUPP.) | Unsigned 8 | Ro | 1 | 31 |
| | 1 | Alarm 1 Input value | Unsigned 16 | Ro | | |
| 750Ah | 0 | Alarm1 Lev (SubIndex SUPP.) | Unsigned 8 | Ro | 1 | 32 |
| | 1 | Alarm 1 Level | Unsigned 16 | Rw | 0 | |
| 750Bh | 0 | Alarm1 Hyst(SubIndex SUPP.) | Unsigned 8 | Ro | 1 | 33 |
| | 1 | Alarm 1 Hysteresis | Unsigned 16 | Rw | 0 | |
| 7510h | 0 | Alarm2 val (SubIndex SUPP.) | Unsigned 8 | Ro | 1 | 31 |
| | 1 | Alarm 2 Input value | Unsigned 16 | Ro | | |
| 751Ah | 0 | Alarm2 Lev (SubIndex SUPP.) | Unsigned 8 | Ro | 1 | 32 |
| | 1 | Alarm 2 Level | Unsigned 16 | Rw | 0 | |
| 751Bh | 0 | Alarm2 Hyst(SubIndex SUPP.) | Unsigned 8 | Ro | 1 | 33 |
| | 1 | Alarm 2 Hysteresis | Unsigned 16 | Rw | 0 | |

Ro = the parameter can be read only
Rw = the parameter can be read and also written
Wo = the parameter can be written only

**Index 6110h Sensor Type**

By a read-access on Index 6110h the sensor type can be polled.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 6110h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Parameter channel 1 | Ro | Unsigned 16 |

**Example**: Read-Access, Node-ID = 2, Index 6110h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 10h | 61h | 01h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 10h | 61h | 01h | 5Ah | 00h | 00h | 00h |

| Parameter (Byte 5) | Sensor Type |
|---|---|
| 5Ah = 90d | Pressure transducer |

Index 6110h is Read-Only. By writing on this Index you will receive a SDO-Communication Error (see “SDO-Error Messages”).

**Index 6112h Operating Mode**

Writing and Reading of Operating Mode from the different channels is done with Index 6112h.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 6112h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Parameter channel 1 | Rw | Unsigned 8 |

The different Operating Modes which are supported are shown in the following list:

| Parameter | Status |
|---|---|
| 00 h | Channel Off |
| 01 h | Normal Operation |

By default the channel is turned on. That means the pressure can be measured just after the initial power-up.

**Example**: You do not want to use the input for your application. To turn the input off, you have to send the following message:

Write access, Module-ID = 2, Index = 6112h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 22h | 12h | 61h | 01h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 60h | 12h | 61h | 01h | 00h | 00h | 00h | 00h |

The input is now turned off. When requesting data from that input you will always receive the value 0.

A change of this parameter will not be stored inside the EEPROM automatically. This has to be done manually by the user (see “Store All Parameters”).

**Index 6125h Autozero**

By writing on Index 6125h the actual measuring value will be interpreted as zero. Internally the actual measuring value will be monitored to the manufacturer threshhold value.

If the threshhold is exceeded, the autozero-access will lead to an error-message.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 6125h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Autozero | Wo | Unsigned 32 |

The autozero will be generated by transmitting the code “zero” as ASCII in the data area of Index 6125h sub1.
The message has the following format:

**Example**: Write-Access, Node-ID = 2, Index 6125h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 22h | 25h | 61h | 01h | 7Ah | 65h | 72h | 6Fh |

The answer (after successful storing of the message) you will receive is:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 60h | 25h | 61h | 01h | 00h | 00h | 00h | 00h |

A change of this parameter will be stored inside the EEPROM automatically.

**Index 6131h Physical Unit Process Value**

Index 6131h has read and write access. By selecting this index the physical unit of the process value can be changed.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 6131h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Physical Unit for Input 1 | Rw | Unsigned 32 |

**Example**: Read Access, Module-ID = 2, Index = 6131h, Sub-Index=1

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 31h | 61h | 01h | 00 | 00 | 00 | 00 |

The answer to this request is:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 31h | 61h | 01h | 10h | 23h | 00h | 00h |

The data bytes 5 + 6 hold the value of the physical unit. Here in the example you see 2310h – Bar.

The physical units available are the following:

| Unit | Value (Hex) |
|---|---|
| Bar | 2310 |
| Kpa | 2323 |
| Psi | 2330 |
| Kg/cm² | 2340 |

By changing the unit by the operator, the internal calculation will be changed according to the selected physical unit. A change of this parameter will be stored inside the EEPROM automatically.

The parameters affected by the changing of the physical unit are: full scale, alarm 1 level, alarm 2 level, alarm 1 hysteresis, alarm 2 hysteresis, maximum peak, 80 % full scale, input offset, lower input limit, upper input limit.

The physical unit could not be changed if one of these parameters will lead to an overflow condition.

**Index 6132h  Decimal Digits Process Value**

Index 6132h has read and write access. By selecting this index the decimal digit of the process value can be changed.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 6132h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Number Decimal Digits for Input 1 | Rw | Unsigned 8 |

**Example**: Read-Access, Node-ID = 2, Index 6132h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 32h | 61h | 01h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 32h | 61h | 01h | 01h | 00h | 00h | 00h |

Data byte 5 holds the value 01h, i.e. the process values are represented with one decimal digit. The possible number of decimal digits are: 0,1 or 2.

By changing the decimal digit, the internal calculation will be changed according to the selected decimal point. A change of this parameter will be stored inside the EEPROM automatically.

The parameters affected by the changing of the physical unit are: full scale, alarm 1 level, alarm 2 level, alarm 1 hysteresis, alarm 2 hysteresis, maximum peak, 80 % full scale, input offset, lower input limit, upper input limit.

The decimal digit could not be changed if one of these parameters will lead to an overflow condition.

**Index 6144 h  Analog Input Lower Limit**
**Index 6145 h  Analog Input Upper Limit**

The objects 6144h and 6145h are specified as “fixed” lower and upper limit for PDO-sending. If only the lower limit (object 6144h) is specified on the transducer, PDO 1 will only be sent if the input value is higher then the lower limit.

If only the higher limit (object 6145h) is specified, PDO 1 will be sent while values are in the range from zero to the specified upper limit.

If both limits are specified, PDO 1 will be sent while values are between lower and upper limit.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 6144h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Analog Input Lower Limit | Rw | Unsigned 16 |

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 6145h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Analog Input Upper Limit | Rw | Unsigned 16 |

**Example**: Write-Access, Node-ID = 2, Index 6144h, Sub-Index = 01h , Lower Threshold=C8h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 22h | 44h | 61h | 01h | C8h | 00h | 00h | 00h |

The answer (after succesful storing) you will receive is:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 60h | 44h | 61h | 01h | 00h | 00h | 00h | 00h |

**Index 6150h Analog Input Status**

By a read-access on Index 6150h the Input State of the device can be requested.

This object is read-only and has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 6150h | 0 | Number of Inputs | Ro | Unsigned 8 |
| | 1 | Input Status | Ro | Unsigned 8 |

**Example**: Read-Access, Node-ID = 2, Index 6150h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 50h | 61h | 01h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 50h | 61h | 01h | 03h | 00h | 00h | 00h |

Data byte 5 holds the value 03h, i.e. there is a positive overload.

The data byte 5 of the CAN message represents the input state in the following manner:

| Parameter | Status |
|---|---|
| 00 h | Measuring Value valid |
| 01 h | Measuring Value not valid |
| 02 h | Full scale exceeded |
| 03 h | Positive Overload (input >10%FS) |

If the measured value is within the input range, the request will return the value 00h, i.e. process value is valid.

**Index 6508 h Alarm 1 Type**
**Index 6518 h Alarm 2 Type**

The objects 6508h and 6518h specifies the alarm type. The alarm type could not be changed by the customer.

The default values are:

| Alarm | Code | Type |
|---|---|---|
| Alarm 1 | 02h | Above or Equal |
| Alarm 2 | 03h | Below |

These objects are read-only and have the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 6508h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Alarm 1 type | Ro | Unsigned 16 |

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 6518h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Alarm 2 type | Ro | Unsigned 16 |

**Example**: Read-Access, Node-ID = 2, Index 6508h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 08h | 65h | 01h | 00h | 00h | 00h | 00h |

The answer (after succesful storing) you will receive is:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 08h | 65h | 01h | 02h | 00h | 00h | 00h |

Byte 5 holds the value 02h.

**Example**: Read-Access, Node-ID = 2, Index 6518h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 18h | 65h | 01h | 00h | 00h | 00h | 00h |

The answer you will receive is:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 18h | 65h | 01h | 03h | 00h | 00h | 00h |

Byte 5 holds the value 03h

**Index 6509 h Alarm 1 Action**
**Index 6519 h Alarm 2 Action**

The objects 6509h and 6519h specifies if the alarm is active or not. By default they are inactive.

The possible values are:

| Value | Meaning |
|---|---|
| 00h | Alarm not active |
| 01h | Alarm active |

These objects have the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 6509h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Alarm 1 Action | Rw | Unsigned 16 |

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 6519h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Alarm 2 Action | Rw | Unsigned 16 |

**Example**: Read-Access, Node-ID = 2, Index 6509h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 09h | 65h | 01h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 09h | 65h | 01h | 01h | 00h | 00h | 00h |

Byte 5 holds the value 01h, that means the Alarm is active.

**Index 7100h A/D value**

By a read-access on Index 7100h the value of the A/D-converter (field value) can be requested.

This object is read-only and has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 7100h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | A/D-value | Ro | Unsigned 16 |

**Example**: Read-Access, Node-ID = 2, Index 7100h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 00h | 71h | 01h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 00h | 71h | 01h | LBh | HBh | 00h | 00h |

Byte 4 holds the number of the input channel, i.e. 1 in this example. In byte 5 and 6 the value of the A/D-converter is given in Intel-Format (Low-Byte first).

**Index 7124h Input Offset**

Index 7124h has read and write access. By selecting this index the offset to the process value can be changed. The offset always has the actual physical unit.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 7124h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Offset (16 bit) | Rw | Unsigned 16 |

**Example**: Read-Access, Node-ID = 2, Index 7124h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 24h | 71h | 01h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 24h | 71h | 01h | LBh | HBh | 00 | 00 |

Byte 5+6 contains the offset data as 16 bit.

**Index 7130h Analog Input Process value**

By a read-access on Index 7130h the process value can be requested. This is the pressure value measured in the selected physical unit.

This object is read-only and has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 7130h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Process Value | Ro | Unsigned 16 |

**Example**: Read-Access, Node-ID = 2, Index 7130h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 30h | 71h | 01h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 30h | 71h | 01h | LB | HB | 00h | 00h |

Byte 4 holds the number of the input channel, i.e. 1 in this example. In byte 5 and 6 the process value given in Intel-Format (Low-Byte first, 16-Bit Signed Integer).

In case of an input signal error the value will be EEEEh. At the same time the objects 1001h (“Error-Register”) and 6150h (“Analog Input State”) will hold an appropriate error code.

Also an emergency message will be sent by the device in that case (refer to “Emergency Message”).

**Index 7500 h Alarm 1 Input Value**
**Index 7510 h Alarm 2 Input Value**

The objects 7500h and 7510h represents the process value which is the input to the alarm function block.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 7500h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Alarm 1 Input value | Ro | Unsigned 16 |

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 7510h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Alarm 2 Input value | Ro | Unsigned 16 |

**Index 750Ah Alarm 1 Level**
**Index 751Ah Alarm 2 Level**

The objects 750Ah and 751Ah represents the level which the Alarm Input Value is compared with Alarm 1 represents the lower limit and Alarm 2 the upper limit. By default they are set to zero.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 750Ah | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Alarm 1 Level | Rw | Unsigned 16 |

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 751Ah | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Alarm 2 Level | Rw | Unsigned 16 |

**Example**: Node-ID = 2, Alarm 1 active , Alarm 2 not active

Alarm 1 level = 1C8h (object 750Ah)

When reaching the level of 1C8 (Input value greater then Alarm 1 Level) the device sends the EMCY:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 82 | 8 | FFh | 00h | 00h | 00h | 10h | 00h | 00h | 00h |

When the input value becomes less then Alarm 1 Level (minus Alarm 1 Hysteresis ) the EMCY is “reset”. That means you will get all “00h” in the according EMCY-message (-> no faults detected).

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 82 | 8 | 00h | 00h | 00h | 00h | 00h | 00h | 00h | 00h |

**Example**: Node-ID = 2, Alarm 1 active , Alarm 2 active

Alarm 1 level = 1C8h (object 750Ah)
Alarm 2 level = 5F0h (object 750Ah)

Alarm 1 works as described above. Alarm 2 is an alarm which works as “Input value less then Alarm 2 Level”.
When reaching first time the value of 5F0 (plus Alarm 1 Hysteresis ) the Alarm 2 is reset.

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 82 | 8 | 00h | 00h | 00h | 00h | 00h | 00h | 00h | 00h |

When falling the input value again under the value of 5F0 the alarm becomes active again

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 82 | 8 | FFh | 00h | 00h | 00h | 20h | 00h | 00h | 00h |

**Index 750Bh Alarm 1 Hysteresis**
**Index 751Bh Alarm 2 Hysteresis**

The objects 750Bh and 751Bh represents the hysteresis which is used for the comparison of the Alarm Input Value and Alarm Level.

It could be used to avoid Emergency Message when the Alarm Input Values are noisy. By default they are set to zero.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 750Bh | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Alarm 1 Hysteresis | Rw | Unsigned 16 |

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 751Bh | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Alarm 2 Hysteresis | Rw | Unsigned 16 |

**Example**: Read-Access, Node-ID = 2, Index 750Bh, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 0Bh | 75h | 01h | 00h | 00h | 00h | 00h |

The answer you will receive is:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 0Bh | 75h | 01h | F0h | 00h | 00h | 00h |

Byte 5 holds the value F0h that represents the Alarm 1 Hysteresis.

## Manufacturer Specific Profile Objects

In this section you will find the manufacturer specific profile indices for the transducer. These indices are implemented according to the DS-404 device profile.

| Index | Sub Index | Name | Type | Access | Default value | Pag |
|---|---|---|---|---|---|---|
| 2000h | 0 | Node-ID (Supp. Subindex) | Unsigned 8 | Ro | 1 | 35 |
| | 1 | Node-ID | Unsigned 8 | Rw | 127 | |
| 2001h | 0 | Baudrate (Supp. Subindex) | Unsigned 8 | Ro | 1 | 35 |
| | 1 | Baudrate | Unsigned 8 | Rw | 7 (500KB) | |
| 2131h | 0 | Full scale (Supp. Subindex) | Unsigned 8 | Ro | 1 | 36 |
| | 1 | Transducer full scale | Integer 16 | Ro | | |
| 2200h | 0 | Cal80 (Supp. Subindex) | Unsigned 8 | Ro | 1 | 36 |
| | 1 | Calibration 80%FS | Integer 16 | Ro | | |
| 2210h | 0 | Temperature (Supp. Subindex) | Unsigned 8 | Ro | 2 | 37 |
| | 1 | Temperature – T ambient | Integer 16 | Ro | | |
| | 2 | Temperature | Integer 16 | Ro | | |
| 2220h | 0 | Peak (Supp. Subindex) | Unsigned 8 | Ro | 2 | 37 |
| | 1 | Read the maximum peak | Unsigned 16 | Ro | | |
| | 2 | Reset the maximum peak | Unsigned 16 | Wo | | |
| 2230h | 0 | Op. time (Supp. Subindex) | Unsigned 8 | Ro | 2 | 38 |
| | 1 | Time counter | Integer 32 | Ro | | |
| | 2 | Number of reset | Integer 32 | Ro | | |

**Index 2000h Node-ID Setting**

By writing on Index 2000h the node-ID setting can be changed. The setting is active only after a reset-node command or a power up. (*) See settings.

When entering values that exceed the supported address-range an error-message will be sent.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 2000h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Node – ID | Rw | Unsigned 8 |

The message has the following format:

**Example**: Write-Access, Node-ID = 2, Index 2000h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 22h | 00h | 20h | 01h | 08h | 00h | 00h | 00h |

Byte 5 hold the new node-id number

The answer after successful storing you will receive is:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 60h | 00h | 20h | 01h | 00h | 00h | 00h | 00h |

After the reset-node the device will work with the node-ID = 8

**Index 2001h Baudrate Setting**

By writing on Index 2001h the buadrate setting can be changed. The setting is active only after a reset-node command or a power up. (*) See setting.

When entering values that exceed the supported range an error-message will be sent.

The object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 2001h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Baudrate | Rw | Unsigned 8 |

The supported baudrate are listed in the following table:

| Byte 5 | Baudrate |
|---|---|
| 1 | 10 Kbaud |
| 2 | 20 Kbaud |
| 3 | 50 Kbaud |
| 4 | 100 Kbaud |
| 5 | 125 Kbaud |
| 6 | 250 Kbaud |
| 7 | 500 Kbaud |
| 8 | 800 Kbaud |
| 9 | 1 Mbaud |

The message has the following format:

**Example**: Write-Access, Node-ID = 2, Index 2001h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 22h | 01h | 20h | 01h | 05h | 00h | 00h | 00h |

Byte 5 hold the new baudrate setting.

The answer after successful storing you will receive is:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 60h | 01h | 20h | 01h | 00h | 00h | 00h | 00h |

After the reset the device will work with a baudrate of 125 Kbaud.

**Index 2131 h Transducer Full Scale**

By a read-access on Index 2131h the transducer full scale can be requested.

This object is read-only and has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 2131h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Transducer full scale | Ro | Integer 16 |

**Example**: Read-Access, Node-ID = 1, Index 2131h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 31h | 21h | 01h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 31h | 21h | 01h | LBh | HBh | 00h | 00h |

In byte 5 and 6 the full scale given in Intel-Format (Low-Byte first, example 88B8h = 350 bar) Index 2131h is Read-Only.

By writing on this Index you will receive a SDO-Communication Error (see “SDO-Error Messages” ).

**Index 2200h  Calibration 80% FS**

By a read-access on Index 2200h the real 80% F.S. of the transducer can be requested.

This object has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 2200h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Transducer Cal | Ro | Integer 16 |

**Example**: Read-Access, Node-ID = 2, Index 2220h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 20h | 22h | 01h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 20h | 22h | 01h | LBh | HBh | 00h | 00h |

In byte 5 and 6 the full scale given in Intel-Format (Low-Byte first, example 6D60h = 280 bar --> 80% of 350bar)

**Object 2210 h  Circuit temperature**

By a read-access on Index 2210h the circuit temperature in °C can be requested.

This object is read-only and has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 2210h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Circuit temperature – T ambient | Ro | Integer 16 |
| | 2 | Circuit temperature | Ro | Integer 16 |

The sub-index 1 holds the difference between the circuit temperature and the ambient temperature (25°C).
The sub-index 2 holds the effective circuit temperature

**Example**: Read-Access, Node-ID = 1, Index 2210h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 10h | 22h | 01h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 10h | 22h | 01h | LBh | HBh | 00h | 00h |

In byte 5 and 6 the temperature in °C with one decimal digit is given in Intel-Format (Low-Byte first, example 0112h = 274dec= 27.4°C) Index 2210h is Read-Only.

By writing on this Index you will receive a SDO-Communication Error (see “SDO-Error Messages” ).

**Object 2220 h Maximum Peak**

By a read-access on Index 2220h sub-index 1 the maximum pressure peak measured can be requested. The maximum peak could be reset by writing on sub-index 2.

This object is read-only and has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 2220h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Maximum peak | Ro | Unsigned 16 |
| | 2 | Reset Maximum peak | Wo | Unsigned 16 |

**Example**: Read-Access, Node-ID = 1, Index 2220h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 20h | 22h | 01h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 20h | 22h | 01h | LBh | HBh | 00h | 00h |

In byte 5 and 6 the maximum pressure peak measure is given in Intel-Format (Low-Byte first, 16-Bit Signed Integer). (example 7148h = 29000dec= 290 bar)

**Example**: Write-Access, Node-ID = 2, Index 2220h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 22h | 20h | 22h | 02h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 60h | 20h | 22h | 02h | 00h | 00h | 00h | 00h |

After a successful writing the Maximum peak is reset.

**Index 2230h Operating Time**

By a read-access on Index 2230h the operating time can be requested.

This object is read-only and has the following structure:

| Index | Sub-Index | Parameter | Access | Type |
|---|---|---|---|---|
| 2230h | 0 | Number of Sub-Index supported | Ro | Unsigned 8 |
| | 1 | Operating time | Ro | Integer 32 |
| | 2 | Number of reset / power up sequences | Ro | Integer 32 |

The operating time is stored in sub-index 1 every 5 minutes. The sub-index 2 contains the number of power up of the transducer and the number of reset (a reset is done if the timer doesn't reach 5 minutes).

**Example**: Read-Access, Node-ID = 1, Index 2230h, Sub-Index = 01h

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1538 | 8 | 40h | 30h | 22h | 01h | 00h | 00h | 00h | 00h |

As answer you will receive from the transducer:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1410 | 8 | 42h | 30h | 22h | 01h | 78h | 00h | 00h | 00h |

In byte 5 ,6, 7 and 8 the operating time is given in Intel-Format
(Low-Byte first, example 00000078h = 120d = 120*5 = 600 minutes)

Index 2230h is Read-Only. By writing on this Index you will receive a SDO-Communication Error (see "SDO-Error Messages" ).

**GEFRAN spa**
via Sebina, 74 - 25050 Provaglio d'Iseo (BS) - ITALIA
Tel. +39 0309888.1 - Fax +39 0309839063
www.gefran.com