# USER MANUAL

## QG65D J1939 Dynamic Inclinometer (Type JA)

V1.0, Dec-21


**DIS Sensors bv**
Oostergracht 40
3763 LZ Soest | The Netherlands

**T** +31 (0)35 - 603 81 81
**M** info@dis-sensors.com
**W** www.dis-sensors.com

# Contents

# 1. Safety information

## 1.1. Intended use

The QG65D J1939 Dynamic Inclinometer is a MEMS-based inclination sensor with J1939 interface. This device provides precise and reliable inclination measurement in both static and dynamic conditions. The use of this device in a machine or system is permitted only under the following conditions:

- The user is trained and competent in the integration and use of inclination sensors in machinery.
- The user is familiar with the contents of both the datasheet and user manual.
- The user has a full understanding of CAN and J1939.
- The device is used within the specified environmental conditions.
- The device is properly configured for its intended use.
- The device is mounted correctly as described in the datasheet and user manual.
- The device data is expressly not interpreted as safety data, except when used redundantly in a control system that is designed and tested for cross-check functionality between the primary and redundant devices.

## 1.2. Incorrect use

- Use of non-standard CAN cables.
- Device mounted incorrectly.
- Zero adjustment outside the specified range.
- Incorrect or absent CAN bus termination.
- Device used outside the specifications.

## 1.3. Sensor defective

The sensor is considered “defective” when

- No CAN output can be detected.

In the event of uncertainty, contact the distributor or manufacturer. Any unauthorised modification or unapproved use will void both the warranty and any liability on the part of the manufacturer.

# 2. About this manual

## 2.1. Intended use

This manual applies solely to the QG65D J1939 dynamic inclinometer (Type JA) with the following model designations:

- QG65D-KIXv-360H-CANJ-C(F)M-UL —— 1 axis, ± 180°, vertical mounting, high accuracy
- QG65D-KDXYh-090H-CANJ-C(F)M-UL —— 2 axes, ± 90°, horizontal mounting, high accuracy
- QG65D-KDXYh-030H-CANJ-C(F)M-UL —— 2 axes, ± 30°, horizontal mounting, high accuracy
- QG65D-KIXv-360-CANJ-C(F)M-UL —— 1 axis, ± 180°, vertical mounting, std. accuracy
- QG65D-KDXYh-090-CANJ-C(F)M-UL —— 2 axes, ± 90°, horizontal mounting, std. accuracy
- QG65D-KDXYh-030-CANJ-C(F)M-UL —— 2 axes, ± 30°, horizontal mounting, std. accuracy

DIS product code can be identified here.

## 2.2. Symbols used in the text

| | |
|---|---|
| h | Subscript for hexadecimal values. |
| d | Subscript for decimal values. |
| ⚠ | Caution that indicates either potential damage to the sensor or explains how to avoid a problem. |
| 🛈 | Important information. |
| → | Cross-reference |

## 2.3. Copyright

## 2.4. Document revision control

| Version | Date(y-m-d) | Revision |
|---|---|---|
| V1.0 | 2021-12-03 | 1st release |
| | | |

Table 1 - Document revision

# 3. Quick reference

**General**

- 1-axis ± 180° and 2-axes ± 30°/± 90° dynamic inclinometer
- Output type: SAE J1939
- Internal sample rate MEMS: 1000Hz
- Dynamic inclination application limitations: max. acceleration 1.5 g, max. angle rate 500°/s
- Approved for automotive use, approval number: E4-10R-05-4662

**CAN, J1939**

- CAN interface: CAN2.0 A and B (complies with ISO11898-1&2)
- Application layer: SAE J1939 (29-bit CAN identifier)
- CAN bus bit rate: 250 kbit/s default, optional 500kbit/s
- Device address: $128_d$ ($80_h$) default
- PGN cycle time: 100ms default
- Output data: PGN 65280 (Inclination), PGN 65283 (Acceleration)
- Sensor configuration: PGN 61184 ($EF00_h$)
- Zero adjustment via object $300F_h$
- Negative values: two’s complement
- Byte-sequence on CAN bus: little-endian (least significant byte first)

# 4. Installation guide

## 4.1. Mechanical drawing

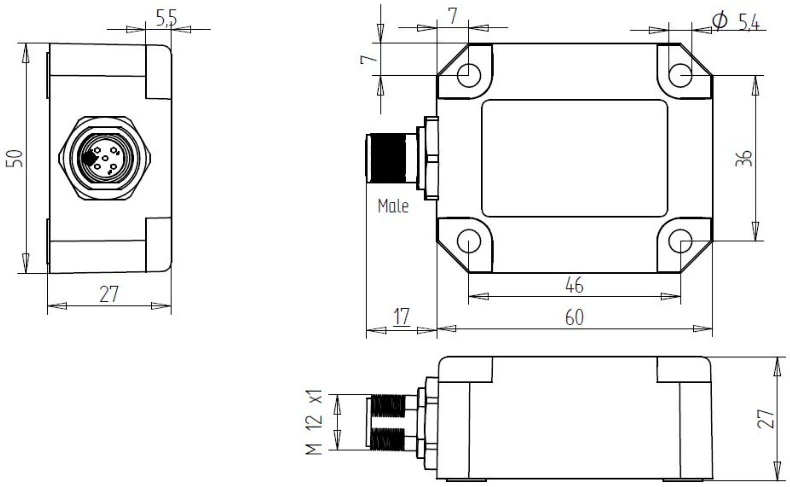



Figure 1 - Mechanical drawing of QG65D with a male M12.

## 4.2. Mounting

- Use the 4 screws provided (M5x25 mm zinc plated steel Pozidrive pan head screws with self-tapping PZ DIN7500CZ) to mount the device.
- If a software zero adjustment is impossible or undesirable, DIS provides an optional reference solution of replacing 2 of the provided screws with 2 optional 4mm Ø positioning pins, which can serve as mechanical reference.
- Tightening torque max. 2.5 Nm. A higher torque can cause damage to the housing or the internal electronics.
- Fix the sensor to a perfectly flat surface.
- A 1-axis inclinometer must be mounted vertically. The factory default zero position is with the male connector pointing down as shown in Figure 3. Sensor can be zero adjusted at any position within the full range.
- A 2-axis inclinometer must be mounted horizontally. The factory default zero position is shown in Figure 2. After installation, sensor can be zero adjusted to eliminate the mechanical offsets within a ±5° offset range.

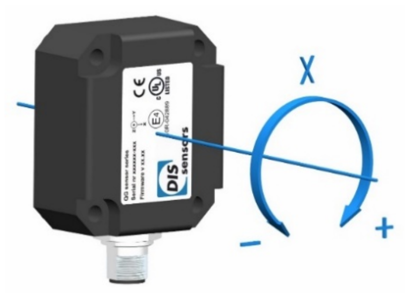



Figure 3 - Vertical mounting 1-axis inclination

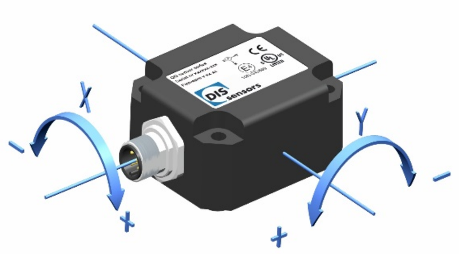



Figure 2 - Horizontal mounting 2-axes inclination

## 4.3. Connection

The sensor is equipped with either a single 5-pole M12 (A-coded) male connector according to IEC 61076-2-101, or with an additional 5-pole M12 (A-coded) female connector. This is specified in the suffix of the device model designation: suffix CM indicates single male connector; suffix CFM indicates both a female and a male connector.

The advantage of two connectors is that devices can be daisy-chained in the CAN network. The voltage supply is forwarded from the male to the female connector, providing both power and bus-connection to the next sensor with a single cable harness. The connection between male and female connector acts as an internal T-junction.

The pin assignment is according to CiA 303-1 V1.8.0.

| Pin | Assignment |
|---|---|
| Pin 1 | Shield |
| Pin 2 | Vcc |
| Pin 3 | Gnd & CAN_GND |
| Pin 4 | CAN_H |
| Pin 5 | CAN_L |

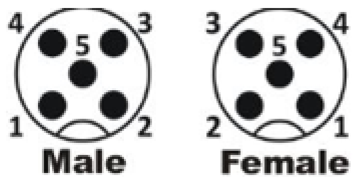



Figure 4 - Pin definition

- The maximum current for the internal T-junction is 2.5 A. This must be taken into account when using daisy-chained connections.
- The voltage supply must be dimensioned to prevent exceeding the specified voltage limits.
- The power consumption is about 50mA typical.
- This device must be connected to a class 2 power supply.
- Connect the sensor only to an approved CAN controller which must have a grounded shield. Alternative: connect the sensor housing to a grounded shield. All mentioned EMC standards that are met (see Declaration of Conformity) have been done with the housing connected to a grounded shield.

*Installation of wiring and the opening and closing of electrical connections must only be carried out in de-energized state! Short-circuits, voltage spikes, etc. can cause malfunctions and/or cause the installation to become uncontrolled, creating a serious risk of injury or damage to property.*

### 4.4. CAN termination

A CAN bus should be terminated properly to prevent signal reflections. The QG65D dynamic Inclinometer has an internal 120 Ohm termination resistor. By default, this is disabled.

A separate M12 termination resistor and T-connector are available as accessories.

**Figure 5 - M12 CAN terminator male 5p.**
**Order code: 10217**

**Figure 6 - M12 CAN terminator female 5p.**
**Order code: 10194**

**Figure 7 - M12 T-connector male 5p.**
**Order code: 11822**

The advantage of using a separate termination plug instead of the internal termination resistor is that an external termination plug is clearly visible. This helps to prevent an inadvertent double termination that could cause the entire CAN bus to malfunction.

## 5. Technical data

Please refer to the datasheet of your product.

# 6. Functional description

## 6.1. Dynamic inclination measurement principle

Conventional inclinometers measure inclination by measuring g-forces using an accelerometer. In static conditions, inclination measurement by this principle is accurate and reliable. However, the vibrations or movements present on almost all industrial machines generate undesirable g-forces. These dynamic g-forces accumulate and overlay the static g-forces, causing a ‘dirty’ inclination output that includes spurious or noise components.

In certain applications a low-pass filter can be used to stabilize the output. The downside of this filtering is that it significantly increases the reaction time (latency) of the inclinometer. In a control system this results in a limit on the speed of movements, and reduced machine efficiency.

A gyroscope measures the speed of rotation. It therefore measures inclination perfectly in dynamic situations, but in static situations it is useless, due to bias drift.

The QG65D dynamic inclinometer combines accelerometer MEMS and gyro MEMS. This device is unaffected by vibration or movement and can therefore measure the true angle at the sensor position in real time. Our smart algorithms (such as Kalman filtering) within the device can determine the balance between the accelerometer MEMS (best result in static situations) and the gyro MEMS (best result in dynamic situations).

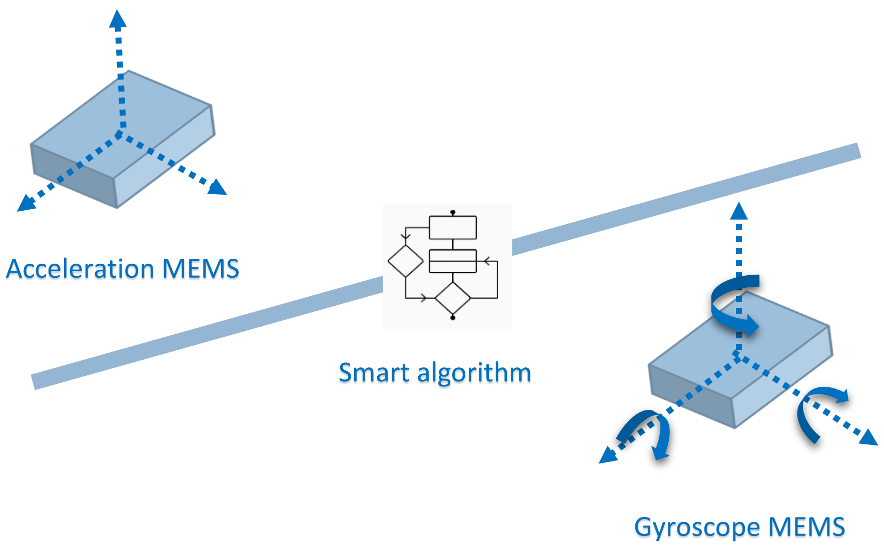



Figure 8 - Measurement principle of the dynamic inclinometer

## 6.2. Measuring axis and direction

### 6.2.1. Measuring axis and direction for Inclination

**A single-axis inclinometer (vertical mount)** measures the inclination in the vertical plane over the full range 0-360° (X-output). The default 0° position and the measuring direction are shown in Figure 9 - Measuring axis and direction of 1-axis inclination.

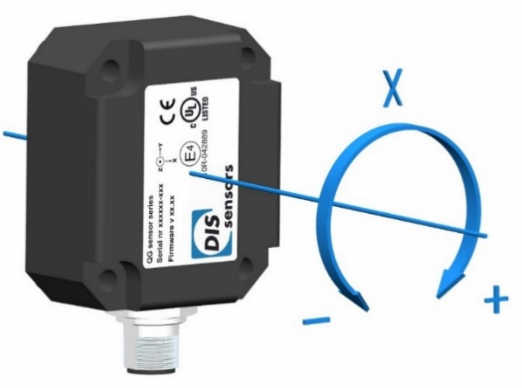



**Figure 9 - Measuring axis and direction of 1-axis inclination.**

**A dual-axis inclinometer (horizontal mount)** measures the inclination on both X and Y axis. Measuring range is up to 90°. Due to the measurement principle, only one axis may tilt more than 45°.

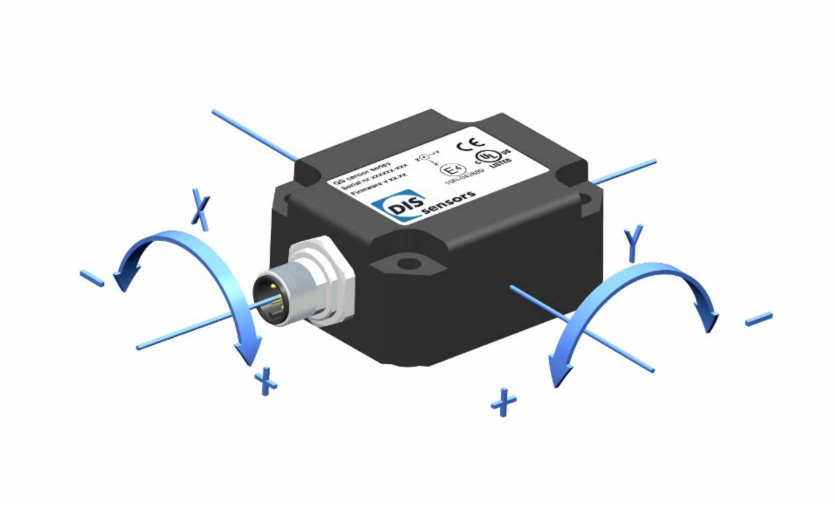



**Figure 10 – Measuring axis and direction of 2-axis inclination.**

⚠ *Cross-tilt error: when the cross axis (axis perpendicular to the measuring axis) tilts more than 45° during the measurement, the measuring axis will be disabled. The sensor will send a fixed output to indicate that the sensor is now at a cross-tilt error position (→Knowledgebase): 20000$_d$ (200°) for the measuring range ±180°, ±30°/ ±90°, and 40000$_d$ (400°) for the measuring range 0°-360°.*

### 6.2.2. Measuring axis and direction for acceleration

The device measures the acceleration values on all 3 axes. The measuring direction of three axes in the factory default setting is shown in Figure 11. Raw acceleration value range are up to ±1.5/±2.0 g with resolution of 0.001g.

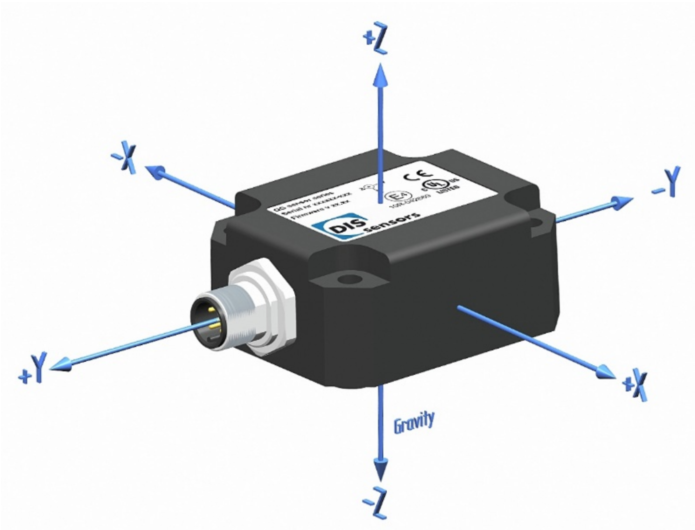



**Figure 11 – Measuring axis and direction of acceleration**

## 6.3. Signal processing

**MEMS Sampling**
- 6 DOF raw outputs
- Sampling rate 1000Hz

**Fusion Algorithm**
- Function mode
- Application profiles
- User adjustable parameters
- Calculation rate 100Hz
- Max. angular rate 500°/s
- Max. acceleration 1.5/2.0g

**Digital Filters**
- Output filter
- Moving average filter

### 6.3.1. Sampling rate

Each axis of the acceleration and gyroscope MEMS is sampled by the internal microcontroller at a rate of 1000 Hz. Those samples are then processed and fed into the sensor fusion algorithm.

### 6.3.2. Sensor fusion algorithm

The sensor fusion algorithm transforms the raw MEMS outputs into a reliable inclination value. Based on the selected function mode, application profile and user-adjustable parameters, the smart algorithm will combine the accelerometer and gyroscope outputs in a balanced way optimised for your application. The limitations for the sensor fusion algorithm are: acceleration max 1.5 / 2.0 g, angular rate max. 500°/s. The calculation rate, and therefore also the output rate, is 100 Hz.

## 6.4. Boot-up procedure

When the device is powered on, the device will be initialised with the last-stored parameters. When initialisation is complete, an Address Claim Message (ACM) containing the address used and NAME will be sent. (→7.5 NAME and ACM (PGN 60928))

| Bus | CAN-ID (h... | Type | Prio. | PGN | PGN (hex) | Len... | Symbol | Data | Source | Source (hex) |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 18EEFF80 | J1939 | 6 | 60928 | EE00h | 8 | ACM | NAMEOfControllerApplication = 9223533667373223231<br>IdentityNumber = 199999<br>ManufacturerCode = DIS Sensors B.V.<br>ECUInstance = 0<br>FunctionInstance = 0<br>Function = 147<br>VehicleSystem = 0<br>VehicleSystemInstance = 0<br>IndustryGroup = Global, applies to all<br>ArbitraryAddressCapable = CA is capable | 128 | 80h |

Figure 12 – example ACM at boot-up

The CAN-ID 18EEFF80 shows that the sensor has a default source address of $80_h$.

The data of the ACM is

```
Hex View:  Rx, Bus 1, 18EEFF80h (ACM), DLC 8, Length 8
0 | 3F 0D A3 89 00 93 00 80
```

Figure 13 - ACM data

This is the 64-bit device NAME for DIS sensors which contains the following fields:

| MSB | | | | 64 bits | | | | | LSB |
|---|---|---|---|---|---|---|---|---|---|
| 1 bit | 3 bit | 4 bit | 7 bit | 1 bit | 8 bit | 5 bit | 3 bit | 11 bit | 21 bit |
| Arbitrary Address Capable | Industry group | Vehicle system Instance | Vehicle System | Reserved | Function | Function Instance | ECU Instance | Manufacturer Code | Identify Number |
| *1* | *000* | *0000* | *0000000* | *0* | *10010011* | *00000* | *000* | *10001001101* | *000110000110100111111* |

Table 2 – NAME definition and data example

| Bit length | Definition | Data | Meaning |
|---|---|---|---|
| 21 bits | Identify Number | $000110000110100111111_b$ | The least 21 bits of the 32-bit serial number of the sensor |
| 11 bits | Manufacturer Code | $10001001101_b = 1101_d$ | DIS Sensors B.V. |
| 3 bits | ECU Instance | $000_b$ | |
| 5 bits | Function Instance | $00000_b$ | |
| 8 bits | Function | $10010011_b = 147_d$ | Inertial Sensor |
| 1 bit | Reserved | $0_b$ | |
| 7 bits | Vehicle System | $0000000_b$ | |
| 4 bits | Vehicle system Instance | $0000_b$ | |
| 3 bits | Industry group | $000_b$ | Global, applies to all industries |
| 1 bit | Arbitrary Address | $1_b$ | yes |

Table 3 - NAME explanation

## 6.5. Output format

The device supports 2 different parameter groups (Tx PGNs):

- Proprietary B Tx PGN1 65280 for Longitudinal, Lateral Inclination (Availability of lateral information depends on sensor type)
- Proprietary B Tx PGN2 65283 for X, Y, Z Acceleration

### 6.5.1. Inclination outputs PGN 65280 (FF00h).

| PGN | Name | Priority | SPN position(bit) | SPN Width(bit) | Resolution | Offset | Data Range |
|---|---|---|---|---|---|---|---|
| **$65280_d$ $FF00_h$** | X axis (Longitudinal) | 3 | 0 | 16 | 0.01°/bit | – | -90° - +90° |
| | Y axis (Lateral) | 3 | 16 | 16 | 0.01°/bit | – | -90° - +90° |

Table 4 – Transmit PGN1 inclination 2-axis

| PGN | Name | Priority | SPN position(bit) | SPN Width(bit) | Resolution | Offset | Data Range |
|---|---|---|---|---|---|---|---|
| **$65280_d$ $FF00_h$** | X axis (Longitudinal) | 3 | 0 | 16 | 0.01°/bit | – | -180° - +180° |

Table 5 - Transmit PGN1 inclination 1-axis

The angle can be calculated with formula: α = J1939 output / 100. (α = angle in degrees, factory resolution is 0.01°)

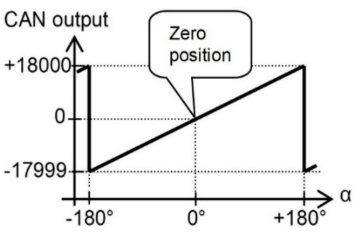



Figure 14 - Output 1-axis ±180° sensor

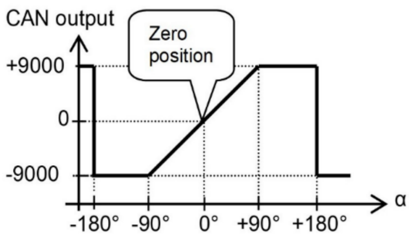



Figure 15 - Output 2-axis ±90° sensor

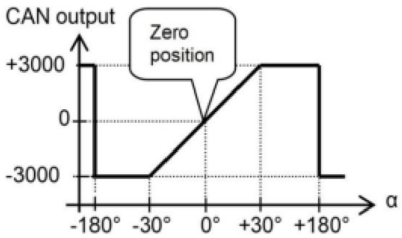



Figure 16 - Output 2-axis ±30° sensor

Example – Inclination output of a 1-axis sensor.

| Time (rel.) | Bus | ID (hex.) | Symbol | Rx/Tx | Type | Prio | PGN | SA | DA | Length | Data (hex.) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0,1217 | 1 | 0CFF0080 | Inclination Output | Rx | Data | 3 | 65280 | 128 | 255 | 8 | 0A 0A FF FF FF FF FF FF |
| 0,2217 | 1 | 0CFF0080 | Inclination Output | Rx | Data | 3 | 65280 | 128 | 255 | 8 | 30 0A FF FF FF FF FF FF |
| 0,3217 | 1 | 0CFF0080 | Inclination Output | Rx | Data | 3 | 65280 | 128 | 255 | 8 | 44 0A FF FF FF FF FF FF |

The first 2 bytes are the angle value, the rest 6 bytes are not used..

| Byte0 | Byte1 |
|---|---|
| $0A_h$ | $0A_h$ |
| Inclination output: $0A0A_h = 2570_d = 25.70°$ | |

Table 6 - Output 1-axis ±180° sensor explanation

Example – Inclination output of a 2-axis sensor

| Time (diff.) | Bus | ID (hex.) | Symbol | Rx/Tx | Type | Prio | PGN | SA | DA | Length | Data (hex.) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 1 | 00FF0080 | Inclination Output | Rx | Data | 0 | 65280 | 128 | 255 | 8 | 8E FC 4B 08 FF FF FF FF |
| 0,1000 | 1 | 00FF0080 | Inclination Output | Rx | Data | 0 | 65280 | 128 | 255 | 8 | F4 FB 7A 08 FF FF FF FF |

The first 2 bytes are X inclination, and the second 2 bytes are Y inclination. The rest 4 bytes are not used.

| Byte0 | Byte1 | Byte2 | Byte3 |
|---|---|---|---|
| **X inclination** | | **Y inclination** | |
| $8E_h$ | $FC_h$ | $4B_h$ | $08_h$ |
| $FC8E_h = -882_d = -8.82°$ | | $084B_h = 2123_d = 21.23°$ | |

Table 7 – Output 2-axis 90° sensor explanation

### 6.5.2. Acceleration outputs PGN 65283 (FF03h).

| PGN | Name | Priority | SPN position(bit) | SPN Width(bit) | Resolution | Offset | Data Range |
|---|---|---|---|---|---|---|---|
| **$65283_d$ $FF03_h$** | X axis (Longitudinal) | 3 | 0 | 16 | 0.001g/bit | – | H series: -1.5g to +1.5g<br>Std. series: -2.0g to +2.0g |
| | Y axis (Lateral) | | 16 | 16 | 0.001g/bit | – | H series: -1.5g to +1.5g<br>Std. series: -2.0g to +2.0g |
| | Z axis (Vertical) | | 32 | 16 | 0.001g/bit | – | H series: -1.5g to +1.5g<br>Std. series: -2.0g to +2.0g |

Table 8 - Transmit PGN2 acceleration

Acceleration can be calculated with formula: a = J1939 output / 1000. (a = acceleration in g, factory resolution is 0.001g)

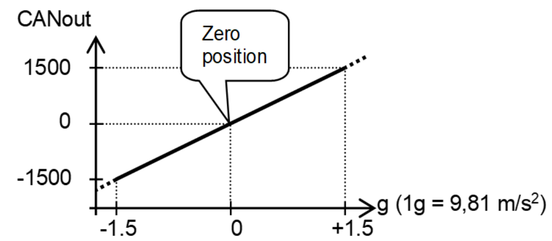



Figure 17 - Output acceleration high accuracy series

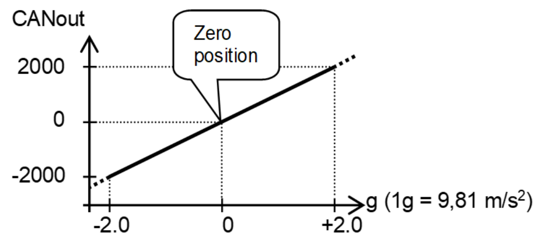



Figure 18 - Output acceleration std. accuracy series

For example:

| Time (diff.) | Bus | ID (hex.) | Symbol | Rx/Tx | Type | Prio | PGN | SA | DA | Length | Data (hex.) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 1 | 00FF0380 | Acceleration Output | Rx | Data | 0 | 65283 | 128 | 255 | 8 | EA FF 0A 00 18 FC FF FF |
| 0,1000 | 1 | 00FF0380 | Acceleration Output | Rx | Data | 0 | 65283 | 128 | 255 | 8 | EA FF 0A 00 18 FC FF FF |

Figure 19 - Acceleration output trace

The 6 bytes data are defined as below.

| Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 |
|---|---|---|---|---|---|
| **X acceleration** | | **Y acceleration** | | **Z acceleration** | |
| $EA_h$ | $FF_h$ | $0A_h$ | $00_h$ | $18_h$ | $FC_h$ |
| $FFEA_h = -22_d = -0.022g$ | | $000A_h = 10_d = 0.010g$ | | $FC18_h = -1000_d = -1.000g$ | |

## 6.6. Sensor configuration (PGN 61184)

For reading and writing the sensor configuration, Prop A PGN 61184 (EF00$_h$) (->PGN (Parameter Group Number)) is used.

Example – Read serial number.

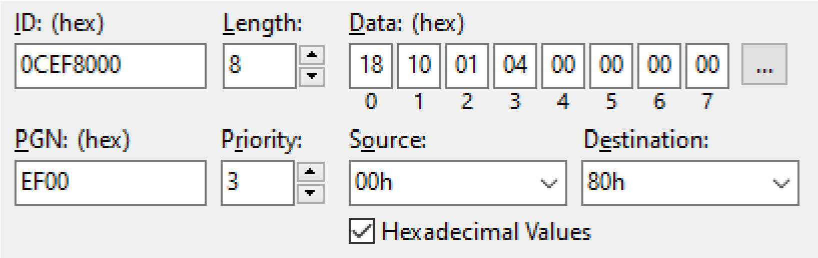



Figure 20 - CMD read serial number

The 8 bytes data are defined as below:

| D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
|---|---|---|---|---|---|---|---|
| Object Index | | CMD | Sub index /Status | Data0 | Data1 | Data2 | Data3 |

<table>
<tr><td>Object index</td><td>Index of the PGN object. (→ 8 Object overview)</td></tr>
<tr><td>CMD</td><td>01<sub></sub>$_h$ = Command for read data<br>02$_h$ = Command for write data</td></tr>
<tr><td>Sub Index:</td><td>message to the sensor: Sub index of the PGN object. (→ 8 Object overview)<br>message from the sensor: Status reply
<table>
<tr><th>Message</th><th>Description</th></tr>
<tr><td>00$_h$</td><td>Ok processing successfully</td></tr>
<tr><td>F0$_h$</td><td>Invalid index</td></tr>
<tr><td>F1$_h$</td><td>Invalid parameter, parameter out of range</td></tr>
<tr><td>F2$_h$</td><td>EEPROM error</td></tr>
</table></td></tr>
<tr><td>Data0 to Data3</td><td>0 to 4 data bytes (Number of valid bytes depends on the object)</td></tr>
</table>

Table 9 - PGN 61184 data definition

More configuration examples can be found in 9 Configuration examples.

## 6.7. Load factory default settings

Factory reset or load factory default settings allows users to restore the device to its original manufacturer settings. The customer settings which were saved to the EEPROM will be overwritten by the factory default settings.

With object $1011_h$, the factory default settings below can be restored. Customised products may have different factory default values, which may deviate from this manual. See the datasheet of your customized products for these customized factory default values.

**DIS factory default**

| Parameters | Object | DIS standard default value |
|---|---|---|
| **Device Address** | $3101_h$ | $80_h$ |
| **Baud Rate** | $3001_h$ | $03_h$ - 250 kBit/s |
| **PGN1 (65280)** | $3102_h$ | Cycle time = $100_d$ = Enabled |
| **PGN2 (65283)** | $3103_h$ | Cycle time = $0_d$ = Disabled |
| **Application profile** | 3001 | 01 = Profile 1 Dynamic (fast mode) |

Table 10 – DIS standard factory default communication parameters

Procedures:


- Write “load” to sub-index 01h = All parameters will be restored.
- Write “load” in ASCII or “64616F6Ch” have the same effect.

Example:

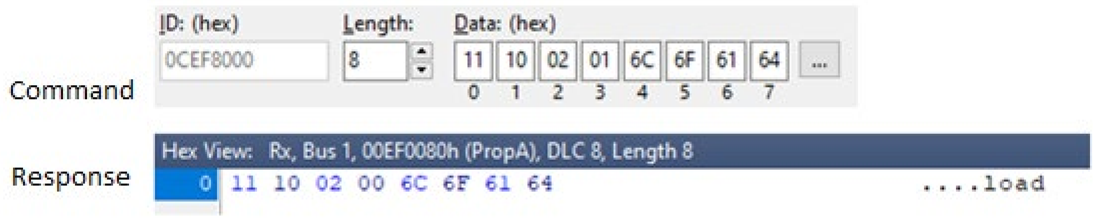



Figure 21 - Restore default settings

⚠

## 6.8. Zero adjustment

Zero adjustment allows users to compensate for mechanical offsets of a horizontally mounted 2-axis device or set a customised 0° position of a vertically mounted 1-axis sensor. The measured inclination value at the 0° position will be stored in the device as a permanent offset, which is used to calculate the actual inclination output of the device. The zero offset value can be read from Object 6013$_h$, 6023$_h$.

Zero adjustment can be done in two ways and can be repeated at any time.

1. **Via Object 300F$_h$ Zero adjustment.**
   To perform a zero adjustment, write the corresponding value to object 300F$_h$ subindex 00$_h$. The zero adjustment can be done for each axis separately or for all axes at the same time. Status information of the result is available from object 300F$_h$.

| Value | | Response |
|---|---|---|
| **01$_h$** | Start zero adjustment X-axis - inclination | 00$_h$ = Successful; FF$_h$ = Failed |
| **02$_h$** | Start zero adjustment Y axis - inclination | 00$_h$ = Successful; FE$_h$ = Failed |
| **03$_h$** | Start zero adjustment X and Y axis - inclination | 00$_h$ = Successful; FD$_h$ = Failed |

Table 11 - Definition object 300F$_h$

For example:
Write 03$_h$ to zero both X and Y axis.

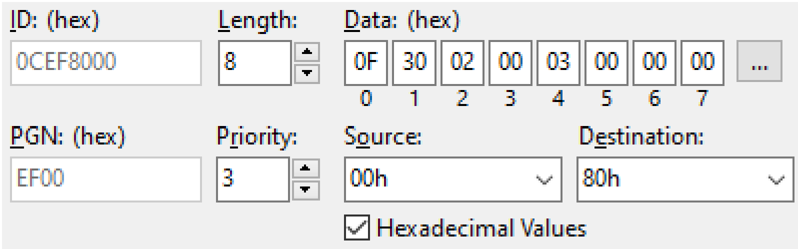



Figure 22 - commands to zero both X and Y axis

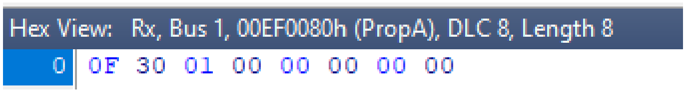



Figure 23 - response from the sensor indicating that zero is successful (data=00)

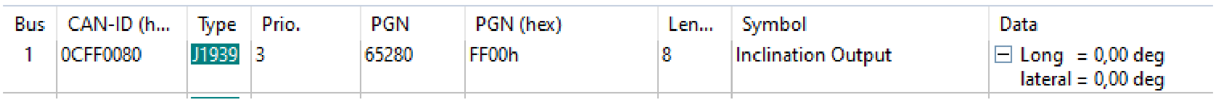



Figure 24 - sensor live output

If the sensor is zeroed at an illegal position, the response (byte4) will be FF/FE/FD$_h$

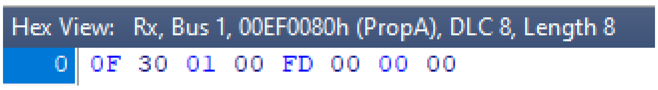



Figure 25 - response shows that zero at X and Y axis both failed.

The offset value will be stored permanently in the device without "save" to object 1010$_h$.

**2. Via Object 6012h, 6022h Pre-set inclination value.**

To set the output of X/Y axis to 0°, write $0000_h$ to object $6012_h$ / $6022_h$ at your desired 0° position. The offset will be stored in Object $6012_h$, $6022_h$ . Those parameters must be saved to the device.

| Index | Sub-index | Definition | Value Example |
|---|---|---|---|
| $6012_h$ | $00_h$ | Pre-set inclination X-axis | $0000_d$ - 0° |
| $6022_h$ | $00_h$ | Pre-set inclination Y axis | $0100_d$ - 1° |

Table 12 - Object $6012/6022_h$ definition

Example - Pre-set the X axis to 0°

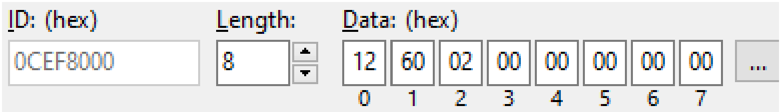



Figure 26 - commands to pre-set X axis to 0°

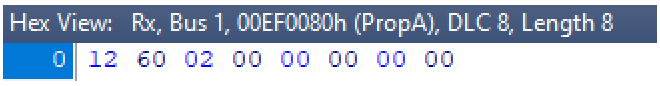



Figure 27 - response from the sensor indicating pre-set is successful (Byte3=$00_h$)

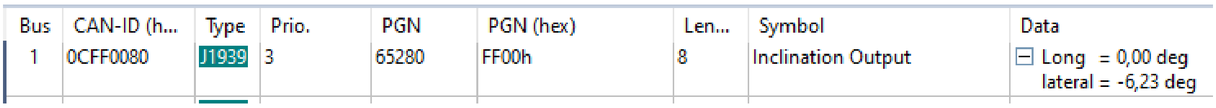



Figure 28 - sensor live output shows that The X axis is zeroed

When an out-of-range pre-set value is sent, the sensor sends "F1" as response indicating "Invalid parameter, parameter out of range" (->6.6 Sensor configuration)

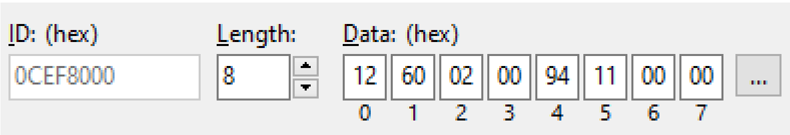



Figure 29 - pre-set X angle to 1194h/45°

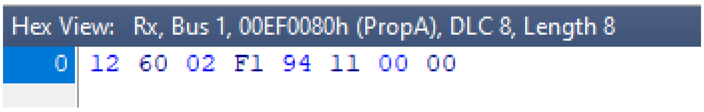



Figure 30 - response from sensor indicating that pre-set is failed. (Byte3=$F1_h$)

*1-axis ±180° sensors can be zero adjusted over the full range, the position of the male connector is at customer discretion. 2-axis ±30°/±90° sensors can only be zero adjusted in a horizontal position within an offset limit of ±5°.*

*After you load factory default settings, the sensor will be restored with the factory zero offset, which means the pre-set inclination value will be restored to $00_h$, object $6013_h$ and $6023_h$ will be rewritten to a factory default value and you need to zero your sensor again.*

## 6.9. Firmware management

The factory-installed firmware version is stated on the device label.


Figure 31 - Firmware version

It can also be requested with object $100A_h$ sub-index $00_h$ Manufacturer software version. The message format is “Vx.x.x” in ASCII, for example: V1.0.0.

⚠

The firmware release notes are available at www.dis-sensors.com under “downloads/user manuals”.

# 7. SAE J1939 specification

The QG65D J1939 series of devices communicates with the SAE J1939 interface. SAE J1939 uses the CAN protocol to transmit a message on the network. Every message includes an identifier which defines the message priority, who sent it, and what data is contained within it.

More knowledge of J1939 can be found in our **Knowledgebase**.

## 7.1. PDU (Protocol Data Unit)

The Protocol Data Unit provides a framework for organizing the key information that is sent with every CAN data frame.

| PDU format | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **SOF** | **Identifier** | **SRR** | **IDE** | **Identifier** | **RTR** | **r1** | **r0** | **DLC** | **Data** | **CRC** | **ACK** | **EOF + IFS** |
| 1 bit | 11 bits | 1 bit | 1 bit | 18 bits | 1 bit | 2 bits | | 4 bits | 0 - 8*8 bits | 16 bits | 2 bits | 10 bits |

Figure 32 - J1939 PDU format

## 7.2. CAN-ID (CAN Identifier)

| PDU format | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **SOF** | **Identifier** | **SRR** | **IDE** | **Identifier** | **RTR** | **r1** | **r0** | **DLC** | **Data** | **CRC** | **ACK** | **EOF + IFS** |
| 1 bit | 11 bits | 1 bit | 1 bit | 18 bits | 1 bit | 2 bits | | 4 bits | 0 - 8*8 bits | 16 bits | 2 bits | 10 bits |

**29-bit CAN-ID**

| 28 | | | 25 | | | | | | | | | | | | | | | | | 8 | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| priority | | | Parameter Group Number (PGN) | | | | | | | | | | | | | | | | | | Source Address (SA) | | | | | | | |

Table 13 - CAN-ID definition

| | |
|---|---|
| **Priority (3 bits)** | Message priority during the arbitration process.<br>0= high and 7=low. Not used by receiver. |
| **PGN (18 bits)** | The PGN uniquely identifies the parameter group (PG) that is being transmitted in the message. |
| **SA (8 bits)** | This field defines the specific address to which the message is being sent. All other destinations should ignore this message. In the case of the global destination address (255), all devices are required to listen and respond as message recipients |

Table 14 - CAN-ID explanation

0CFF0080h

| 28 | | | 25 | | | | | | | | | | | | | | | | | 8 | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| priority = 3 | | | Parameter Group Number (PGN) = 0FF00h | | | | | | | | | | | | | | | | | | Source Address (SA) = 80h | | | | | | | |

Figure 33 - Example CAN-ID of inclination output

*11-bit identifier devices can be used within the same network without conflict, but it's not defined in J1939 standards.*

## 7.3. PGN (Parameter Group Number)

The J1939 PGN comprises an 18-bit subset of the 29-bit extended CAN ID. The PGN uniquely identifies the parameter group (PG) that is being transmitted in the message. Every PG (grouping of specific parameters) has its own specific definition comprising the assignment of each parameter within an 8-byte data field (size in bytes, location of the LSB), and the transmission rate and priority of the message.

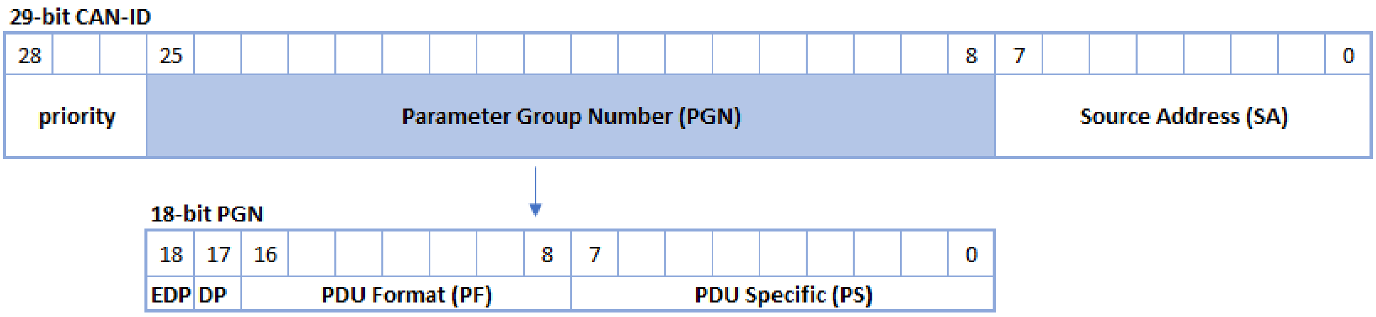



Figure 34 - PGN format

<table>
<tr><td>EDP (Extended Data Page)</td><td>Select a page for the PGNs.</td></tr>
<tr><td>DP (Data Page)</td><td>
<table>
<tr><th>EDP</th><th>DP</th><th>Definition</th></tr>
<tr><td>0</td><td>0</td><td>J1939 Page 0 PGNs</td></tr>
<tr><td>0</td><td>1</td><td>J1939 Page 1 PGNs</td></tr>
<tr><td>0</td><td>0</td><td>Reserved</td></tr>
<tr><td>0</td><td>1</td><td>Diagnostic message</td></tr>
</table>
</td></tr>
<tr><td>PDU format (PF):</td><td>Defines whether a PGN has a PDU1 or PDU2 format.<br>Determines the PGN assigned to the Data field.
<table>
<tr><th></th><th>Format 1</th><th>Format 2</th></tr>
<tr><td>PF value</td><td>0 – 239d/EF$_h$</td><td>240$_d$/F0$_h$ – 255$_d$/FF$_h$</td></tr>
<tr><td>PGNs</td><td>2*240 = 480</td><td>2*16*256 = 8192</td></tr>
<tr><td>Usage</td><td>point-to-point messages<br>or global message</td><td>Global message only<br>Broadcast message etc.</td></tr>
</table>
</td></tr>
<tr><td>PS – PDU Specific</td><td>This field depends on the value of PDU Format and contains either a destination address or a group extension, depending on PDU Format.</td></tr>
</table>

| 18 | 17 | 16 | | | | | 8 | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EDP | DP | PDU Format 1(PF) < 240 | | | | | | Destination Address (DA) | | | | | | | |

| 18 | 17 | 16 | | | | | 8 | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EDP | DP | PDU Format 2 (PF) >= 240 | | | | | | Group Extension | | | | | | | |

**DA:** defines the specific address to which the message is being sent.

**GE:** provides extra 4096 PGNs per data page.

Table 15 - PGN definition

PGNs used by DIS J1939 sensors are explained as below:

| PGN | PG Name | J1939 Definition | DIS |
|---|---|---|---|
| **$61184_d$ ($00EF00_h$)** | PropA | This PG uses the destination-specific PDU1 Format allowing manufacturers to direct their proprietary communications to a specific destination node. How the Data field of this message is used is up to each manufacturer. | Sensor configuration is sent with PGN 61184 |
| **$65280_d$ ($00FF00_h$)** | PropB | This PG uses the PDU2 Format message allowing manufacturers to define the PS (GE) field content as they desire. | Inclination output is sent by PGN 65280. |
| **$65283_d$ ($00FF03_h$)** | PropB | This PG uses the PDU2 Format message allowing manufacturers to define the PS (GE) field content as they desire. | Acceleration output is sent by PGN 65283. |
| **$60928_d$ ($00EE00_h$)** | AC | The Address Claimed PGN may be used in two ways: to claim an address, and to announce that a CA was unable to claim an address. | ACM |
| **$59904_d$** | Request Message | The request message (PGN 59904) for the Address Claimed message (PGN 60928) is used by any CA to request the NAMEs and addresses of CAs in ECUs attached to the network. | RQST |

Table 16 - PGN examples

## 7.4. SPN (Suspect Parameter Number)

SPN is a 19-bit identifier for the CAN signals (parameters) contained in the data. SPNs are grouped by PGNs and can be described in terms of their bit start position, bit length, scale, offset and unit - information required to extract and scale the SPN data to physical values.

The assignment of SPNs can be found in SAE J1939-21 documents. DIS sensors B.V. is using PGN 2551 for definition of inclination and acceleration outputs.

| | | | | |
|---|---|---|---|---|
| | PGN 65280 to 65535 Proprietary B | Manufacturer Defined Usage (PropB_PDU2) | 2551 | 64 to 14280 |

Example – SPN 2551 by PGN 65280 Inclination output defined by DIS Sensors B.V.

| PGN | Name | Priority | Position in PG | Bit Size | Resolution (scale) | Offset | Data Range |
|---|---|---|---|---|---|---|---|
| **65280$_d$ FF00$_h$** | X axis (Longitudinal) | 3 | 0 | 16 | 0.01°/bit | | -90° - +90° |
| | Y axis (Lateral) | | 16 | 16 | 0.01°/bit | | -90° - +90° |

## 7.5. NAME and ACM (PGN 60928)

Every device shall have one address and an associated **NAME** to communicate on the SAE J1939 network.

SAE J1939 defines a 64-bit NAME to identify the primary function of a controller application. The NAME consists of fields shown in table below.

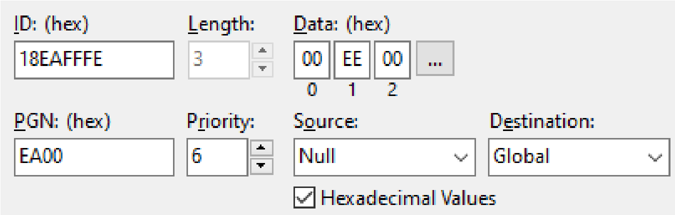
| MSB | | | | 64 bits | | | | | LSB |
|---|---|---|---|---|---|---|---|---|---|
| 1 bit | 3 bit | 4 bit | 7 bit | 1 bit | 8 bit | 5 bit | 3 bit | 11 bit | 21 bit |
| Arbitrary Address Capable | Industry group | Vehicle system Instance | Vehicle System | Reserved | Function | Function Instance | ECU Instance | Manufacturer Code | Identify Number |

Table 17 - NAME format

**Address Claim**, defined in SAE J1939-81, is the process for claiming a network address and associating the NAME, or vehicle function, to that address.

Upon power-up or whenever requested, a device must send an ACM to claim an address. Other devices must compare the newly claimed address with their own to avoid conflicts. When multiple devices claim the same address, the device with lower NAME value uses the address, the others must claim a different address or stop communicating on the network.

A device can send request for the ACM to determine whether addresses claimed by other devices. PGN 59904 (RQST) for ACM (PGN 60928) is described as below:




All requested devices send their ACMs so that you know which devices are currently on the network.

Any CA that is unable to successfully claim an address shall respond with a Cannot Claim Address message. It has the same PGN as ACM but has a source address of 254d (the NULL address).

# 8. Object overview

Some CAN open objects are kept for J1939 sensors for the configuration purpose. Use PGN 61184 to read/write those objects. (→ 6.6 Sensor configuration (PGN 61184))

| Index | Subindex | Data Type | Access | Object Name and description | Factory Default | Detail |
|---|---|---|---|---|---|---|
| **$1000_h$** | **$00_h$** | U32 | r | **Device type** | $0101019A_h$ – 1-axis inclination<br>$0102019A_h$ – 2-axis inclination | |
| **$1008_h$** | **$00_h$** | VSTR | const | **Manufacturer device name** | | |
| **$1009_h$** | **$00_h$** | VSTR | const | **Manufacturer hardware version** | Depends on sensor. E.g. “V8.00” | |
| **$100A_h$** | **$00_h$** | VSTR | const | **Manufacturer software version** | Depends on sensor. E.g. “V1.0.0” | |
| **$1010_h$** | **$01_h$** | | | **Save all parameters**<br>Write “save” or “$65766173_h$” | | 9.4 |
| **$1011_h$** | **$01_h$** | | | **Restore all parameters**<br>Write “load” or “$64616F6C_h$” | | |
| **$1018_h$** | | | | **Identity object** | | |
| | **$00_h$** | U8 | const | Highest subindex supported | $04_h$ | |
| | **$01_h$** | U32 | r | Vendor ID | $000001BD_h$ | |
| | **$02_h$** | U32 | r | Product code | $08000001_h$ – 1-axis inclination 360°<br>$08000002_h$ – 2-axes inclination ±90°<br>$08000003_h$ – 2-axes inclination ±30° | |
| | **$03_h$** | U32 | r | Revision number | $00010000_h$ – V1.0.0 | |
| | **$04_h$** | U32 | r | Serial number | | |
| **$3001_h$** | **$00_h$** | U8 | rw | **Baud Rate**<br>$02_h$ = 500 kBit/s<br>$03_h$ = 250 kBit/s | $03_h$ | |
| **$3003_h$** | **$00_h$** | U8 | rw | **CAN bus termination resistor 120Ω**<br>$00_h$ = Disable<br>$01_h$ = Enable | $00_h$ | |
| **$300F_h$** | **$00_h$** | U8 | rw | **Zero adjustment Inclination**<br>$01_h$ – Zero X-axis – $00_h$=successful, $FF_h$=failed<br>$02_h$ – Zero X-axis – $00_h$=successful, FE=failed<br>$03_h$ – Zero X and Y axis – $00_h$=successful, $FD_h$=failed<br><br>**Zero adjustment Acceleration**<br>$01_h$ – Zero X-axis – $00_h$=successful, $FF_h$=failed<br>$02_h$ – Zero Y-axis – $00_h$=successful, FE=failed<br>$03_h$ – Zero Z-axis – $00_h$=successful, $FD_h$=failed<br>$07_h$ – Zero all axis – $00_h$=successful, $F9_h$=failed | | |
| **$3101_h$** | **$00_h$** | U8 | rw | **J1939 Address**<br>(Range: $0_d$ – $254_d$) | $80_h$ | |

| Index | Subindex | Data Type | Access | Object Name and description | Factory Default | Detail |
|---|---|---|---|---|---|---|
| **$3102_h$** | | | | **PGN 65280 Angle output** | | |
| | **$01_h$** | U32 | r | Priority (range: 0-7) | $03_h$ | |
| | **$02_h$** | U32 | r | Cycle time [ms]<br>0 = deactivated | $0064_h$ = 100ms | |
| **$3103_h$** | | | | **PGN 65283 Acceleration output** | | |
| | **$01_h$** | U32 | r | Priority (range: 0-7) | $03_h$ | |
| | **$02_h$** | U32 | r | Cycle time [ms]<br>0 – deactivated | $0064_h$ = 100ms | |
| **$6000_h$** | **$00_h$** | U16 | r | **Resolution**<br>$1_d$=0.001, $10_d$=0.01, $100_d$=0.1, $1000_d$=1.0 | $10_d$ | |
| **$6012_h$** | 00h | S16 | rw | **Pre-set Inclination X-axis** | $0000_h$ | |
| **$6013_h$** | **$00_h$** | S16 | rw | **Offset Inclination X-axis** | $0000_h$ | |
| **$6022_h$** | **$00_h$** | S16 | rw | **Pre-set Inclination Y-axis** | $0000_h$ | |
| **$6023_h$** | **$00_h$** | S16 | rw | **Offset Inclination Y-axis** | $0000_h$ | |

⚠ *To permanently save communication parameters in the EEPROM of the device, use only CAN Object 1010h, otherwise the changes will be lost after a power cycle. All indices and/or subindices not described in the table are reserved exclusively for factory use.*

Data types used in the application layers are explained in the following table.

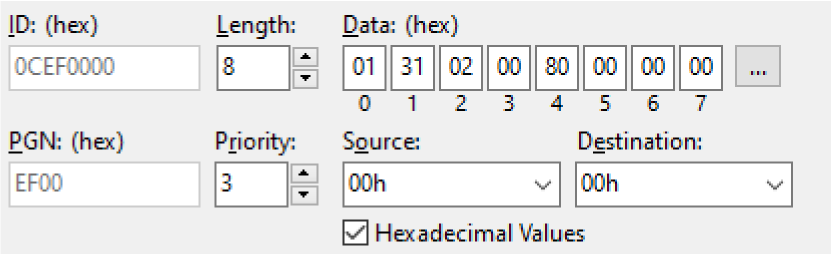
| Data type | Definitions |
|---|---|
| **U8** | Unsigned 8-bit number (0 – $255_d$) |
| **U16** | Unsigned 16-bit number (0 – $65535_d$) |
| **U32** | Unsigned 32-bit number (0 – $4294967295_d$) |
| **S8/Integer 8** | Signed integer 8-bit number ($-128_d$ – $+127_d$) |
| **S16/Integer 16** | Signed integer 16-bit number ($-32768_d$ – $+32767_d$) |
| **S32/Integer 32** | Signed integer 16-bit number ($-2147483648_d$ – $+2147483647_d$) |
| **VSTR** | Visible String |

Table 18 - Data type

# 9. Configuration examples

## 9.1. Change device address

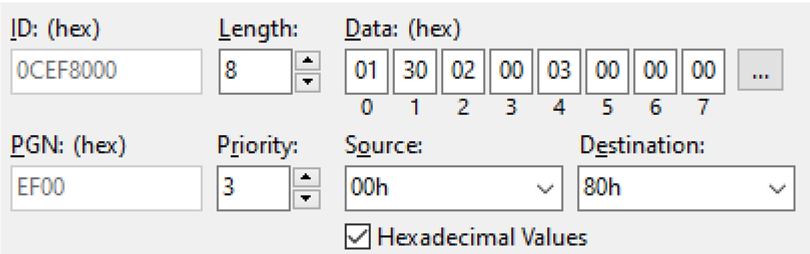
Use object $3101_h$ sub-index $00_h$ to change the device address.




| | |
|---|---|
| Byte 0-1 | Object index = $3101_h$ |
| Byte 2 | CMD = $02_h$, write to the sensor. |
| Byte 3 | Object sub index = $00_h$ |
| Byte 4 | New Device address = $80_h$ |
| Byte 5-7 | 00 = Not used |

## 9.2. Change baud rate

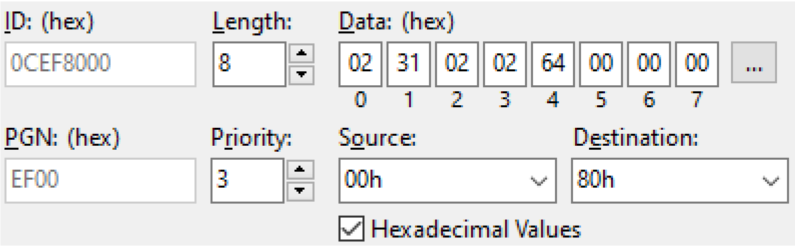
Use object $3001_h$ sub-index $00_h$ to change the baud rate.




| | |
|---|---|
| Byte 0-1 | Object index = $3001_h$ |
| Byte 2 | CMD = $02_h$, write to the sensor. |
| Byte 3 | Object sub index = $00_h$ |
| Byte 4 | $03_h$ = 250 kbit/s<br>$02_h$ = 500 kbit/s |
| Byte 5-7 | 00 = Not used |

## 9.3. Enable/Disable PGNs

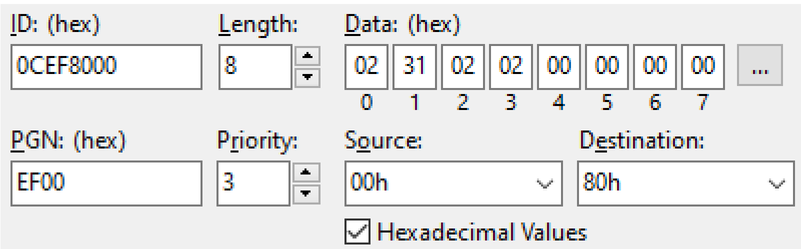
Use PGN cycle time to enable or disable PGNs. Object $3102_h$ is used for PGN 65280 and object $3103_h$ is used for PGN 65283. Below is an example for PGN 65280.




| | |
|---|---|
| Byte 0-1 | Object index = $3102_h$ (object used for PGN 65280) |
| Byte 2 | CMD = $02_h$, write to the sensor. |
| Byte 3 | Object sub index = $02_h$ |
| Byte 4 | PGN cycle time = $64_h$ = 100ms<br>Setting PGN cycle time to a non-zero value will enable PGN |
| Byte 5-7 | 00 = Not used |

Table 19 - Example enable PGN 65280

Disable PGN

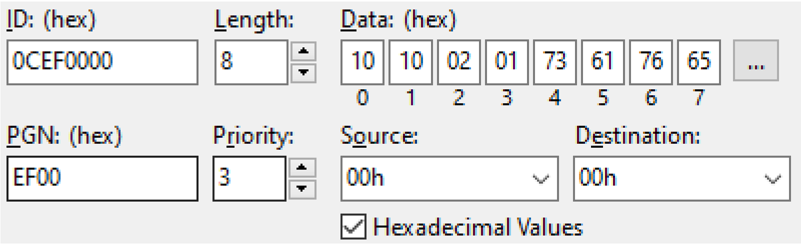



| | |
|---|---|
| Byte 0-1 | Object index = $3102_h$ |
| Byte 2 | CMD = $02_h$, write to the sensor. |
| Byte 3 | Object sub index = $02_h$ |
| Byte 4 | PGN cycle time = $00_h$<br>Setting PGN cycle time to zero will disable PGN |
| Byte 5-7 | 00 = Not used |

Table 20 - Example disable PGN 65280

## 9.4. Save changes to the sensor

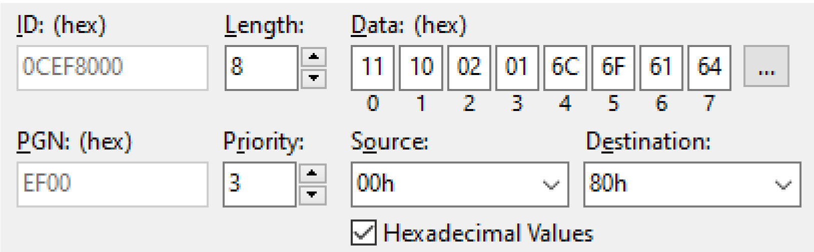
Use object $1010_h$ sub-index $01_h$




| | |
|---|---|
| Byte 0-1 | Object index = $1010_h$ |
| Byte 2 | CMD = $02_h$, write to the sensor. |
| Byte 3 | Object sub index = $01_h$ |
| Byte 4-7 | Write $65766173_h$ (save) |

## 9.5. Restore factory default

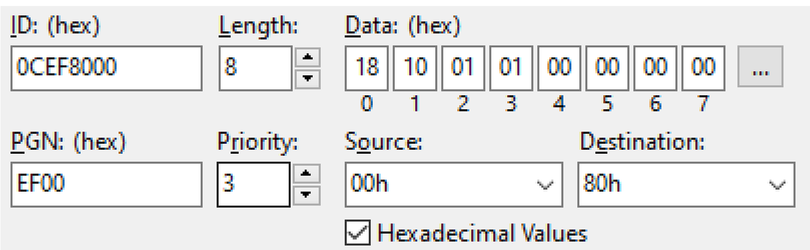
Use object $1011_h$ sub-index $01_h$




| | |
|---|---|
| Byte 0-1 | Object index = $1011_h$ |
| Byte 2 | CMD = $02_h$, write to the sensor. |
| Byte 3 | Object sub index = $01_h$ |
| Byte 4-7 | Write $64616F6C_h$ (load) |

## 9.6. Read vendor ID

Vendor ID can be read by object $1018_h$ sub-index $01_h$



| | |
|---|---|
| Byte 0-1 | Object index = $1018_h$ |
| Byte 2 | CMD = $01_h$, read sensor |
| Byte 3 | Object sub index = $01_h$ |
| Byte 4-7 | $00_h$ = Not used |

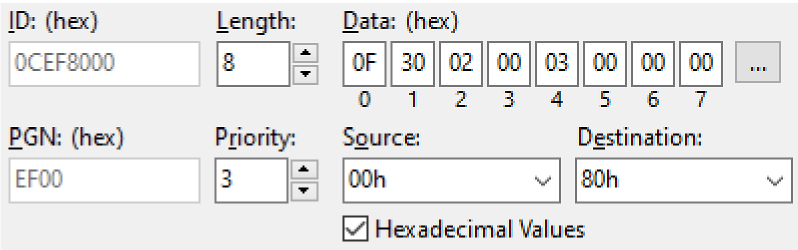
The status reply from the sensor is

```
Hex View:  Rx, Bus 1, 00EF0080h (PropA), DLC 8, Length 8
0 | 18 10 01 00 BD 01 00 00
```

| | |
|---|---|
| Byte 0-1 | Object index = $1018_h$ |
| Byte 2 | CMD = $01_h$, read sensor |
| Byte 3 | Status reply = $00_h$ = Read successful |
| Byte 4-7 | Data = $000001BD_h$ = DIS Sensors B.V. |

## 9.7. Zero adjustment

Use object $300F_h$ sub-index $00_h$ to zero the sensor. Below is an example of zeroing X and Y axis for a 2-axis sensor.




| | |
|---|---|
| Byte 0-1 | Object index = $300F_h$ |
| Byte 2 | CMD = $02_h$, write to the sensor |
| Byte 3 | Sub-index = $00_h$ |
| Byte 4-7 | Data = $03_h$ = Inclination Zero adjustment for X and Y axis.<br>**Zero adjustment Inclination**<br>$01_h$ Zero X-axis $00_h$=successful<br>$FF_h$=failed<br>$02_h$ Zero Y-axis $00_h$=successful<br>FE=failed<br>$03_h$ Zero X and Y axis $00_h$=successful<br>$FD_h$=failed<br>**Zero adjustment Acceleration**<br>$01_h$ Zero X-axis $00_h$=successful<br>$FF_h$=failed<br>$02_h$ Zero Y-axis $00_h$=successful<br>FE=failed<br>$03_h$ Zero Z-axis $00_h$=successful<br>$FD_h$=failed<br>$07_h$ Zero all axis $00_h$=successful<br>$F9_h$=failed |

# 10. Abbreviations and definitions

| | |
|---|---|
| **ACM** | Address Claimed Message |
| **CAN** | Controller Area Network |
| **CAN-ID** | CAN Identifier |
| **PGN** | Parameter Group Number |
| **PDU** | Protocol Data Unit |
| **SPN** | Suspect Parameter Number |
| **MEMS** | Microelectromechanical system |
| **DOF** | Degrees of freedom |
| **PropA** | Proprietary A, PGN 61184 |
| **PropB** | Proprietary B |

# 11. Normative references

| | |
|---|---|
| **ISO11898-1** | Controller area network (CAN) —Data link layer and physical signalling |
| **ISO11898-2** | Controller area network (CAN) —High-speed medium access unit |
| **SAE J1939** | SAE's family of standards relating to the Controller Area Network (CAN) for heavy-duty vehicles. |
| **IEC 61076-2-101** | Connectors for electronic equipment - Part 2-101: Circular connectors – Detail specification for circular connectors M8 with screw- or snap-locking, M12 with screw locking for low voltage applications |