

# MVINC-CO-X-XXX-YYYY-Z TRIAxIAL INCLINOMETER With CANopen®

## USER MANUAL

### AX060851 – Triaxial Inclinometer

#### All Models:

- AX060851-1PG11** – Triaxial Inclinometer, CANopen®, 1 PG11 Strain Relief  
(supersedes model MVINC-CO-1-180-PG11-1)
- AX060851-2PG11** – Triaxial Inclinometer, CANopen®, 2 PG11 Strain Reliefs  
(supersedes model MVINC-CO-1-180-PG11-2)
- AX060851-1MIL4** – Triaxial Inclinometer, CANopen®, 1 MIL 4 Connector  
(supersedes model MVINC-CO-1-180-MIL4-1)
- AX060851-2MIL4** – Triaxial Inclinometer, CANopen®, 2 MIL 4 Connectors  
(supersedes model MVINC-CO-1-180-MIL4-2)

## ACRONYMS

CAN	Controller Area Network
CANopen®	CANopen® is a registered community trademark of CAN in Automation e.V.
CAN-ID	CAN 11-bit Identifier
COB	Communication Object
EDS	Electronic Data Sheet
EMCY	Emergency
LSB	Least Significant Byte (or Bit)
LSS	Layer Settling Service
MSB	Most Significant Byte (or Bit)
NMT	Network Management
RO	Read Only Object
RPDO	Received Process Data Object
RW	Read/Write Object
SDO	Service Data Object
TPDO	Transmitted Process Data Object
WO	Write Only Object

## REFERENCES

- [DS-301] CiA DS-301 V4.1 – CANopen Application Layer and Communication Profile. CAN in Automation 2005
- [DS-305] CiA DS-305 V2.0 – Layer Setting Service (LSS) and Protocols. CAN in Automation 2006
- [DS-410] CiA DS-410 V1.2 – CANopen profile for Inclinator. (Used for Out-of-Range-Temperature EMCY purposes only)

These documents are available from the CAN in Automation e.V. website <http://www.can-cia.org/>.

# TABLE OF CONTENTS

<b>1. INTRODUCTION .....</b>	<b>6</b>
<b>2. INCLINOMETER DESCRIPTION .....</b>	<b>7</b>
<b>2.1. THEORY OF OPERATION.....</b>	<b>7</b>
2.1.1. Unit Coordinate System .....	7
2.1.2. Unit Reference Frames.....	7
2.1.3. Angle measurements .....	8
2.1.4. Tilt Angles .....	9
2.1.5. Euler Angles.....	10
2.1.6. Gimbal Lock.....	11
<b>2.2. HARDWARE BLOCK DIAGRAM .....</b>	<b>12</b>
<b>2.3. SOFTWARE ORGANIZATION.....</b>	<b>12</b>
<b>3. INCLINOMETER LOGICAL STRUCTURE .....</b>	<b>13</b>
<b>3.1. FUNCTION BLOCK SIGNALS .....</b>	<b>14</b>
3.1.1. Undefined Signal .....	14
3.1.2. Discrete Signal .....	14
3.1.3. Continuous Signal.....	14
3.1.4. Signal Type Conversion .....	15
3.1.5. Discrete to Continuous Conversion.....	15
3.1.6. Continuous to Discrete Conversion.....	15
3.1.7. Function Block Signal/Control Source List.....	15
<b>3.2. ACCELEROMETER.....</b>	<b>16</b>
<b>3.3. ANGLE MEASUREMENT .....</b>	<b>17</b>
<b>3.4. UNIT INSTALLATION .....</b>	<b>19</b>
<b>3.5. ANGLE CONFIGURATION .....</b>	<b>19</b>
<b>3.6. UNIT INSTALLATION: VERTICAL AND HORIZONTAL MOUNT .....</b>	<b>20</b>
<b>3.7. SENSOR CALIBRATION .....</b>	<b>20</b>
<b>3.8. BINARY FUNCTIONS.....</b>	<b>20</b>
<b>3.9. GLOBAL PARAMETERS .....</b>	<b>22</b>
<b>3.10. MISCELLANEOUS FUNCTION BLOCK.....</b>	<b>23</b>
3.10.1. RPDO Messages.....	24
3.10.2. Automatic Update of Objects.....	24
3.10.3. Startup.....	24
<b>4. CANOPEN ® OBJECT DICTIONARY .....</b>	<b>25</b>
<b>4.1. NODE ID AND BAUDRATE .....</b>	<b>25</b>
4.1.1. LSS Protocol to Update.....	25
<b>4.2. COMMUNICATION OBJECTS (DS-301) .....</b>	<b>29</b>
4.2.1. Object 1000h: Device Type .....	30
4.2.2. Object 1001h: Error Register.....	30
4.2.3. Object 1002h: Manufacturer Status Register.....	30
4.2.4. Object 1003h: Pre-Defined Error Field .....	30
4.2.5. Object 100Ch: Guard Time .....	32
4.2.6. Object 100Dh: Lifetime Factor .....	32
4.2.7. Object 1010h: Store Parameters.....	33
4.2.8. Object 1011h: Restore Parameters.....	34
4.2.9. Object 1016h: Consumer Heartbeat Time .....	35
4.2.10. Object 1017h: Producer Heartbeat Time .....	36
4.2.11. Object 1018h: Identity Object.....	36
4.2.12. Object 1020h: Verify Configuration .....	37
4.2.13. Object 1029h: Error Behaviour .....	38
4.2.14. RPDO Behaviour .....	39
4.2.15. TPDO Behaviour .....	41
<b>4.3. MANUFACTURER OBJECTS.....</b>	<b>44</b>
4.3.1. Object 2000h: CAN Slew Rate .....	45
4.3.2. Object 2002h: Sensor Input Filter Enabled .....	45

4.3.3.	Object 2003h: Sensor Input Filter Enabled .....	45
4.3.4.	Object 2010h: Accelerometer Data Decimal Digits .....	46
4.3.5.	Object 2011h: Accelerometer Data Inversion/Negation.....	46
4.3.6.	Object 2020h: Accelerometer Data Offset .....	47
4.3.7.	Object 2030h: Angular Data Decimal Digits .....	47
4.3.8.	Object 2031h: Angular Data Inversion/Negation .....	48
4.3.9.	Object 2032h: Angle Data Range .....	48
4.3.10.	Object 2040h: Angular Data Offset .....	49
4.3.11.	Object 2100h: Pitch and Roll Angle Type .....	49
4.3.12.	Object 2200h: Initial Pitch Angle .....	50
4.3.13.	Object 2210h: Initial Roll Angle .....	50
4.3.14.	Object 2220h: Mounting Installation .....	50
4.3.15.	Object 2300h: Auto-Null Command.....	51
4.3.16.	Object 2400h: Calibrated Pitch Angle Data .....	51
4.3.17.	Object 2401h: Calibrated Roll Angle Data .....	52
4.3.18.	Object 2500h: RPDO Received Messaged Process Value.....	52
4.3.19.	Object 2500h: RPDO Decimal Digits.....	52
4.3.20.	Object 2502h: RPDO Offset.....	53
4.3.21.	Object 3000h: Binary Function Output Field Value.....	53
4.3.22.	Object 300Ah: Binary Function Operation .....	54
4.3.23.	Object 300Bh: Binary Function Output Scaling .....	54
4.3.24.	Object 300Bh: Binary Function Output Scaling .....	55
4.3.25.	Object 300Ch: Binary Function Output Offset .....	55
4.3.26.	Object 30x0h: Binary Function x Signal 1&2 Source .....	56
4.3.27.	Object 30x1h: Binary Function x Signal 1&2 Is Default .....	56
4.3.28.	Object 30x2h: Binary Function x Signal 1&2 Default Value.....	57
4.3.29.	Object 30x3h: Binary Function x Signal 1&2 Unary Function.....	57
4.3.30.	Object 30x4h: Binary Function x Signal 1&2 Scaling .....	58
4.3.31.	Object 30x5h: Binary Function x Signal 1&2 Offset.....	58
4.3.32.	Object 4000h: Accelerometer Axis Data Field Value.....	59
4.3.33.	Object 4010h: Angular Measurement Latency .....	60
4.3.34.	Object 4020h: Angular Figure of Merit .....	60
4.3.35.	Object 4030h: Pitch Angle Data Field Value.....	60
4.3.36.	Object 4040h: Roll Angle Data Field Value.....	61
4.3.37.	Object 4050h: Gravity Angle Data Field Value .....	61
4.3.38.	Object 4060h: Sensor Temperature Data .....	62
4.3.39.	Object 4070h: Continuous Signal .....	62
4.3.40.	Object 4080h: Discrete Signal .....	62
4.3.41.	Object 5020h: Power Supply Field Value .....	63
4.3.42.	Object 5030h: Processor Temperature Field Value .....	63
4.3.43.	Object 5550h: Enable Automatic Updates .....	63
4.3.44.	Object 5555h: Start in Operational Mode.....	64
<b>5.</b>	<b>RS-232 COMMUNICATION .....</b>	<b>65</b>
<b>6.</b>	<b>TECHNICAL SPECIFICATIONS.....</b>	<b>67</b>
<b>6.1.</b>	<b>PERFORMANCE PARAMETERS.....</b>	<b>67</b>
6.1.1.	Static Parameters .....	67
6.1.2.	Dynamic Parameters .....	67
<b>6.2.</b>	<b>POWER SUPPLY INPUT .....</b>	<b>67</b>
<b>6.3.</b>	<b>CAN OUTPUT .....</b>	<b>67</b>
<b>6.4.</b>	<b>GENERAL SPECIFICATIONS .....</b>	<b>68</b>
6.4.1.	Compliances.....	68
6.4.2.	Certifications .....	69
<b>6.5.</b>	<b>CONNECTORS .....</b>	<b>69</b>
6.5.1.	MIL4 .....	69
6.5.2.	MIL6 .....	69
6.5.3.	Spring load PCB Connector.....	70
<b>6.6.</b>	<b>UNIT ORIENTATION.....</b>	<b>70</b>

**6.7. HOUSING ..... 71**  
6.7.1. MVINC ..... 71  
**6.8. INSTALLATION ..... 71**  
**7. VERSION HISTORY ..... 72**

## 1. INTRODUCTION

---

The following user manual describes: architecture, functionality, configuration parameters and object details for the Triaxial CANopen Inclinometer. It also contains technical specifications and installation instructions of the device.

The various function blocks supported by the Triaxial Inclinometer are outlined in the following sections. All objects are user configurable using standard commercially available tools that can interact with a CANopen® Object Dictionary via an *EDS* file.

## 2. INCLINOMETER DESCRIPTION

---

The inclinometer is designed to measure pitch and roll inclination angles in a full  $\pm 180$  Degree orientation range. The unit can also output: gravity angle and unit accelerations in three orthogonal directions.

The inclinometer transmits angular data over CAN bus using a standard CANopen protocol. The unit's original configuration can be changed using any third party CANopen tools.

The inclinometer can be configured through a set of configuration parameters to fit the user specific application requirements.

### 2.1. Theory of Operation

#### 2.1.1. Unit Coordinate System

The unit uses a standard right-handed Z-down Cartesian coordinate system, see:

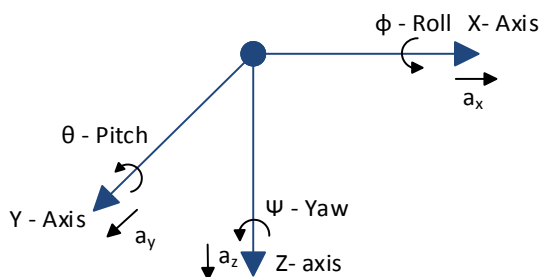


Figure 1. Inclinometer Coordinate System

The arrows in Figure 1 represent a direction of motion that produces a positive change of the parameter. For  $a_x$ ,  $a_y$ ,  $a_z$  accelerations, the positive acceleration direction is the same as the axis direction. For  $\theta$ ,  $\phi$ ,  $\psi$  rotation angles the positive direction is counterclockwise about the axis of rotation.

The Z-down coordinate system is described by in the SAE J670 standard for automotive applications. This system is similar to the NED (North-East-Down) coordinate system used in aerospace and navigation, but without reference to the cardinal directions.

#### 2.1.2. Unit Reference Frames

Several Z-down coordinate systems or frames are used to describe the inclinometer orientation. The (X,Y,Z) coordinate system attached to the unit forms a unit frame, see: Figure 2.

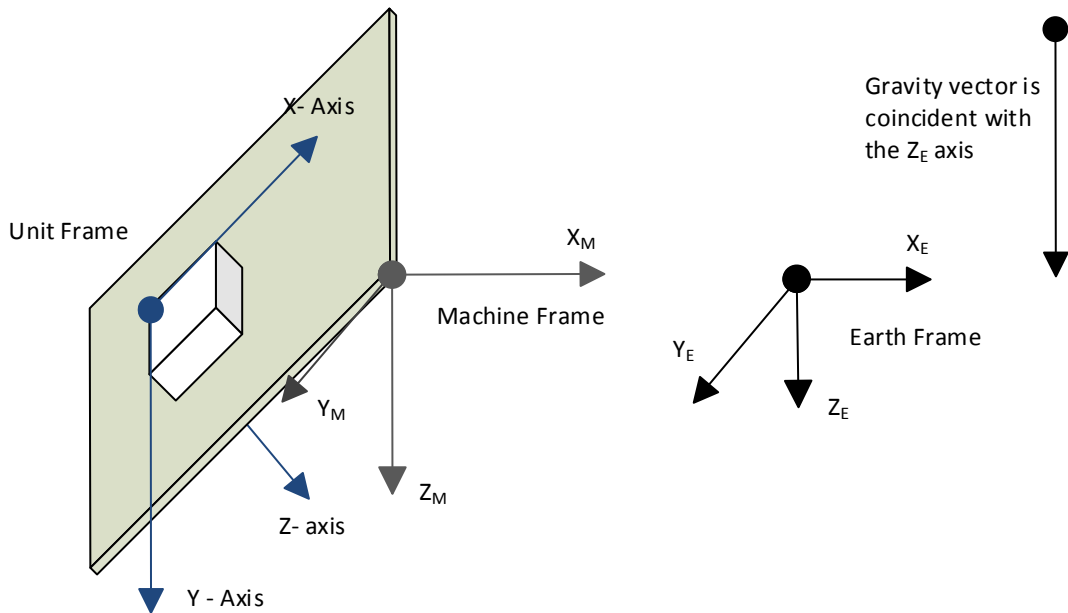


Figure 2. Inclinometer Reference Frames

The  $(X_M, Y_M, Z_M)$  coordinate system attached to the machine, where the sensor is installed, defines a machine frame. The Earth coordinate system  $(X_E, Y_E, Z_E)$ , aligned with the Earth gravity, defines the Earth absolute reference frame.

The unit calculates accelerations, angles and angular rates referred to the machine frame  $(X_M, Y_M, Z_M)$ . Conversion from the unit frame  $(X, Y, Z)$  to the machine frame  $(X_M, Y_M, Z_M)$  is performed internally using the unit initial installation angles.

The unit initial installation angles are set to zero by default, which means that the unit frame is coincident with the machine frame and all inclinometer parameters can be also referred to the unit frame  $(X, Y, Z)$ .

After the unit is installed on the machine at the customer site, the customer can set-up the unit initial installation angles through configuration parameters. After that, the unit will output angles and all other parameters referred only to the machine frame  $(X_M, Y_M, Z_M)$ .

To simplify further description of inclinometer operations, it will be assumed that the unit initial installation angles are zero and all inclinometer parameters are referred to the unit frame  $(X, Y, Z)$ .

### 2.1.3. Angle measurements

The inclination angles are measured by a three-axis MEMS accelerometer, which senses accelerations:  $a_x, a_y, a_z$  in three orthogonal directions.

The accelerations are then transformed into inclination angles based on the assumption that the acceleration caused by the gravity force  $\vec{g}$  is the only one that applies to the unit:

$$\vec{a} = -\vec{g} \tag{1}$$

The unit calculates:  $\theta$  – pitch,  $\phi$  – roll, and  $\rho$  – gravity angles. There is not enough information based only on the unit accelerations to calculate the  $\psi$  – yaw angle.



The pitch and roll angles can be calculated in two different ways: as tilt or Euler angles. The gravity angle is always a tilt angle.

The tilt angles are inclination angles that define the tilt of the unit relatively to the ground plain or the gravity vector. Pitch and roll tilt angles are the angles that a dual axis inclinometer (or two single axis inclinometers placed in orthogonal directions) will measure in the same position as the unit. They are measured relatively to the ground plane. The gravity angle is a tilt angle measured relatively to the gravity vector.

The Euler angles are coordinate system rotation angles performed in a specific order to rotate the unit from its original position, parallel to the Earth surface, to its current position.

The unit measures tilt angles by default. In comparison with the Euler angles, they are numerically stable in the whole angular range. They also cover a wider range of pitch angles up to  $\pm 180^\circ$ , whereas the Euler pitch angle is restricted only to the  $\pm 90^\circ$  range to avoid ambiguity in the angular rotations.

Despite all advantages, the tilt angles are not suitable to define the unit position in space, where the Euler angles are used. Even when the Euler angles are not used to calculate the pitch and roll angles, they are still used internally to compensate the unit initial installation angles.

#### 2.1.4. Tilt Angles

The tilt angles:  $\theta^t$  and  $\phi^t$ , are inclination angles of the unit relatively to the  $(X_E, Y_E)$  ground plane, parallel to the Earth surface in the Earth frame  $(X_E, Y_E, Z_E)$ , see: Figure 3.

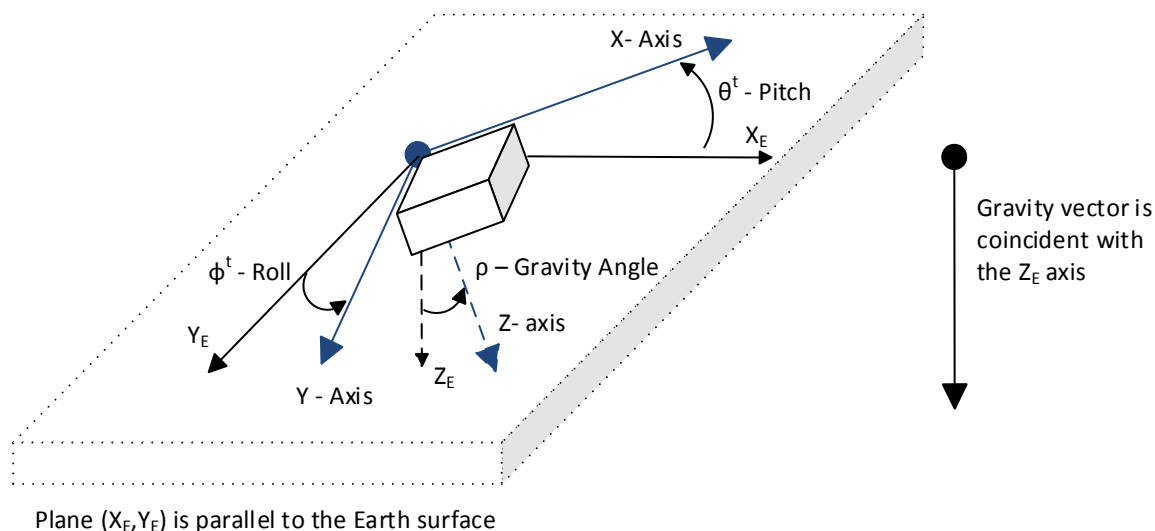


Figure 3. Tilt Angles

The gravity angle  $\rho$  is an inclination angle of the unit relatively to the gravity force vector or  $Z_E$  axis of the Earth frame.

The tilt angles are calculated using unit accelerations:  $a_x, a_y, a_z$  the following way:

$$\theta^t = \text{atan2}(-g_x, \text{sgn}(g_z) \cdot \sqrt{g_y^2 + g_z^2}), \quad \theta^t \in [-180^\circ; 180^\circ], \quad (2)$$

$$\phi^E = \text{atan2}(g_y, \text{sgn}(g_x) \cdot \sqrt{g_y^2 + g_z^2}), \quad \phi^E \in [-180^\circ; 180^\circ],$$

$$\rho = \text{atan2}(\sqrt{g_x^2 + g_y^2}, g_z), \quad \rho \in [0^\circ; 180^\circ],$$

where:  $\text{sgn}(x) = \begin{cases} -1, & x < 0 \\ 1, & x \geq 0 \end{cases}$

and:  $g_x = -a_x, g_y = -a_y, g_z = -a_z$ .

When the unit is parallel to the Earth surface, all tilt angles are zero:  $\theta^E = \phi^E = \rho = 0^\circ$ .

### 2.1.5. Euler Angles

The Euler angles:  $\theta^E$  and  $\phi^E$ , together with the  $\psi^E$ , are rotation angles around the  $X_E, Y_E^*$  and  $X$  axes performed in a standard (yaw, pitch, roll) rotation sequence used in aerospace and defined in SAE J670 standard for automotive applications, see: Figure 4.

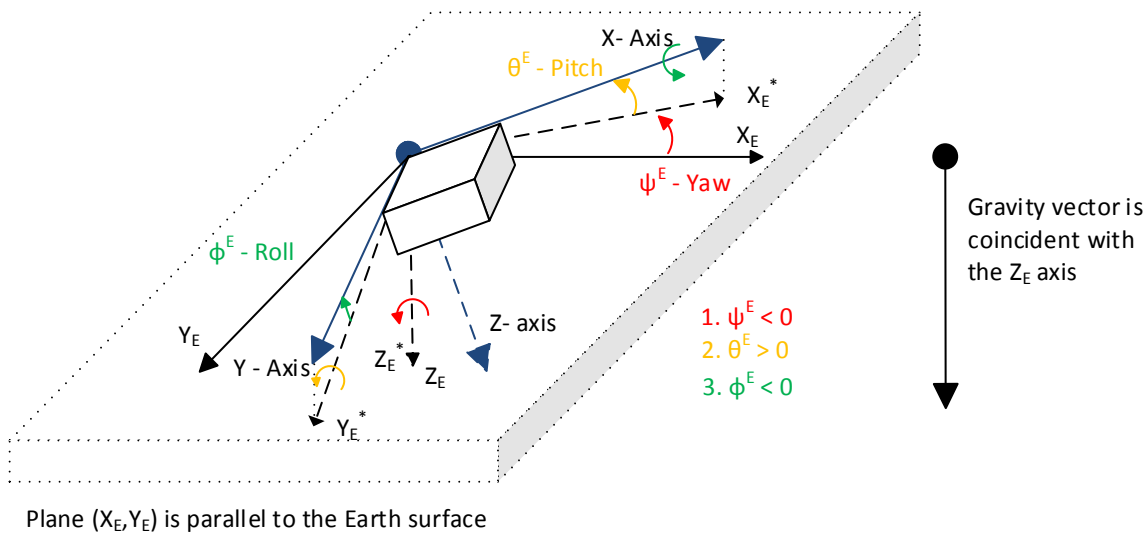


Figure 4. Euler Angles

The first rotation defines the  $\psi^E$  – yaw angle. It is performed about the  $Z_E$  axis of the Earth-fixed coordinate system  $(X_E, Y_E, Z_E)$  from the  $X_E$  axis to the  $X_E^*$  axis. An intermediate coordinate system  $(X_E^*, Y_E^*, Z_E^*)$  is a Z-down coordinate system whose  $X_E^*$  and  $Y_E^*$  axes are parallel to the ground plane  $(X_E, Y_E)$ , with the  $X_E^*$  axis aligned with the vertical projection of the X axis onto the ground plane. Since the yaw rotation  $\psi^E$  on Figure 4 is opposite to the positive rotation direction, shown by the red arrow about the  $Z_E$  axis, the resulted angle is negative.

The second rotation defines the  $\theta^E$  – pitch angle. It is performed about the  $Y_E^*$  axis of the intermediate coordinate system  $(X_E^*, Y_E^*, Z_E^*)$  from the  $X_E^*$  axis to the X axis. The pitch rotation  $\theta^E$  on Figure 4 is in the positive rotation direction, defined by the yellow arrow about the  $Y_E^*$  axis, and the resulted angle is therefore positive.

The final third rotation defines the  $\phi^E$  – roll angle, as a rotation about the X axis from the  $Y_E^*$  axis to the Y axis. The roll rotation  $\phi^E$  on Figure 4 is negative. It is performed in the direction opposite to the positive rotation direction shown by the green arrow about the X axis.

The set of the three: yaw, pitch and roll Euler angles fully represents the angular position of the inclinometer in space.

There is not enough information for the unit to calculate the yaw angle based only on the accelerometer data. It is assumed that the yaw angle is always zero:  $\psi^E = 0$  in the coordinate rotations.

The Euler angles are calculated using the following formulae:

$$\theta^E = \text{atan2}(-g_x, \sqrt{g_y^2 + g_z^2}), \quad \theta^E \in [-90^\circ; 90^\circ], \quad (3)$$

$$\phi^E = \text{atan2}(g_y, g_z), \quad \phi^E \in [-180^\circ; 180^\circ],$$

where:  $g_x = -a_x$ ,  $g_y = -a_y$ ,  $g_z = -a_z$ .

When the unit is parallel to the Earth surface, the Euler angles are zero:  $\theta^E = \psi^E = 0^\circ$ .

### 2.1.6. Gimbal Lock

The formula for the roll angle  $\phi^E$  is numerically unstable when both:  $g_y = g_z = 0$ . This condition, called a gimbal lock, happens when the unit is placed in the vertical position with the Z axis parallel to the gravity vector, see Figure 5. When this happens, the unit effectively loses one degree of freedom and the roll angle  $\phi^E$  becomes undefined and can take any random value.

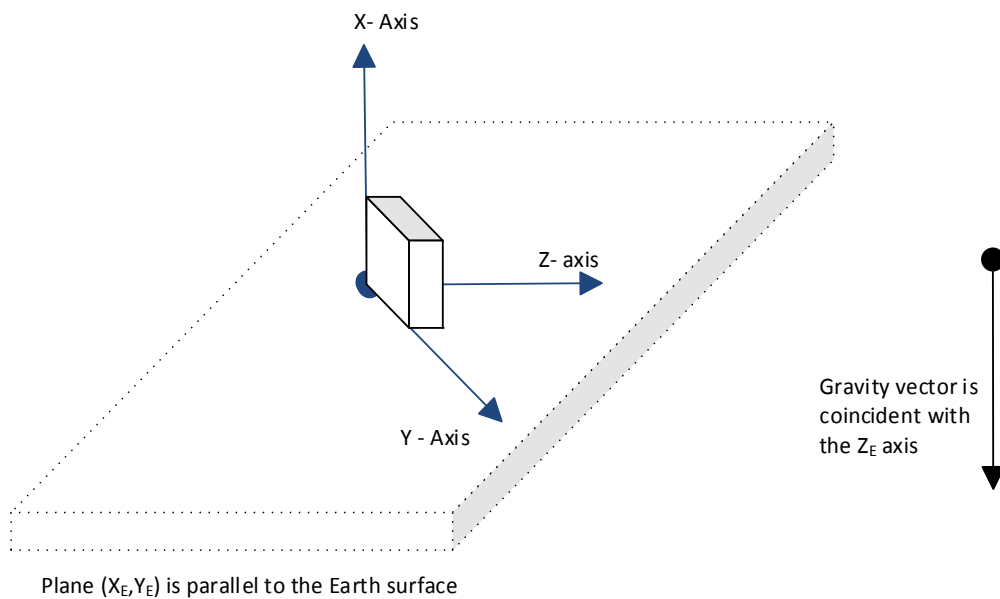


Figure 5. Gimbal Lock

For the obvious reason, the gimbal lock should be avoided in the unit initial installation. It should be also avoided in the inclinometer working range, when Euler angles are used to calculate pitch and roll angles.

## 2.2. Hardware Block Diagram

The unit contains a three-axis MEMS accelerometer, which senses acceleration in three orthogonal directions: X, Y and Z.

The outputs of MEMS accelerometer are processed by a 32-bit microcontroller to calculate the unit accelerations and inclination angles. The inclination angles are then output to CAN bus, see Figure 6.

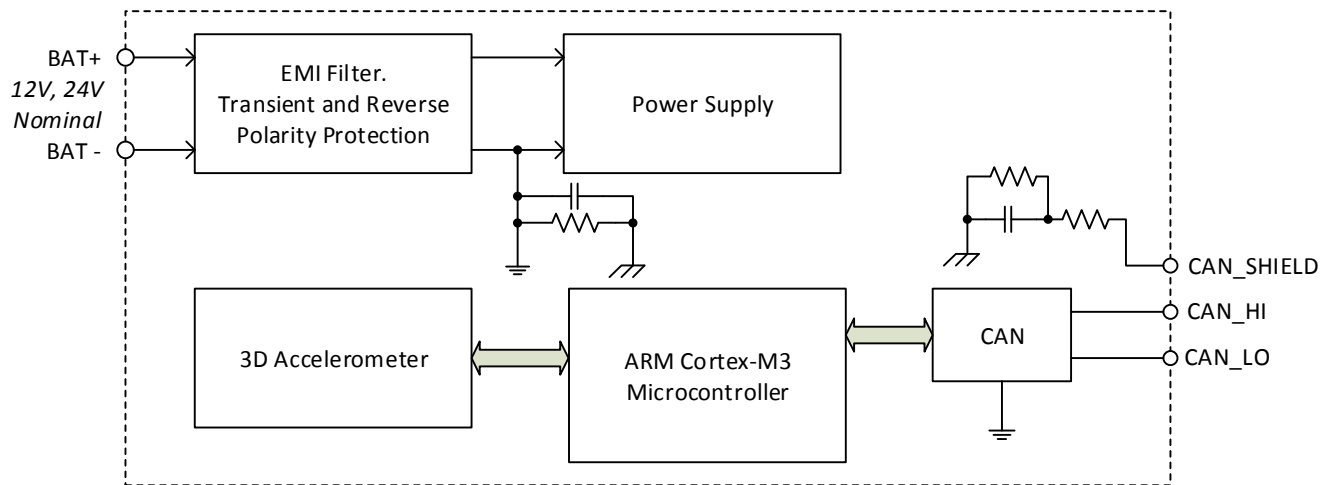


Figure 6. The Inclinometer Hardware Block Diagram

The inclinometer has a wide range of protections features including a transient and reverse polarity protection, see [Technical Specifications](#) section.

## 2.3. Software Organization

The Tri-axial J1939 CAN Inclinometer belongs to a family of Axiomatic smart controllers with configurable internal architecture. This architecture allows building a controlling algorithm based on a set of predefined internal configurable function blocks without the need of custom software.

The various function blocks supported by the Triaxial Inclinometer are outlined in the following sections. All objects are user configurable using standard commercially available tools that can interact with a CANopen® Object Dictionary via an *EDS* file.

### 3. INCLINOMETER LOGICAL STRUCTURE

The inclinometer is internally organized as a set of function blocks, which can be individually configured and arbitrarily connected together to achieve the required unit functionality, see: **Error! Reference source not found..**

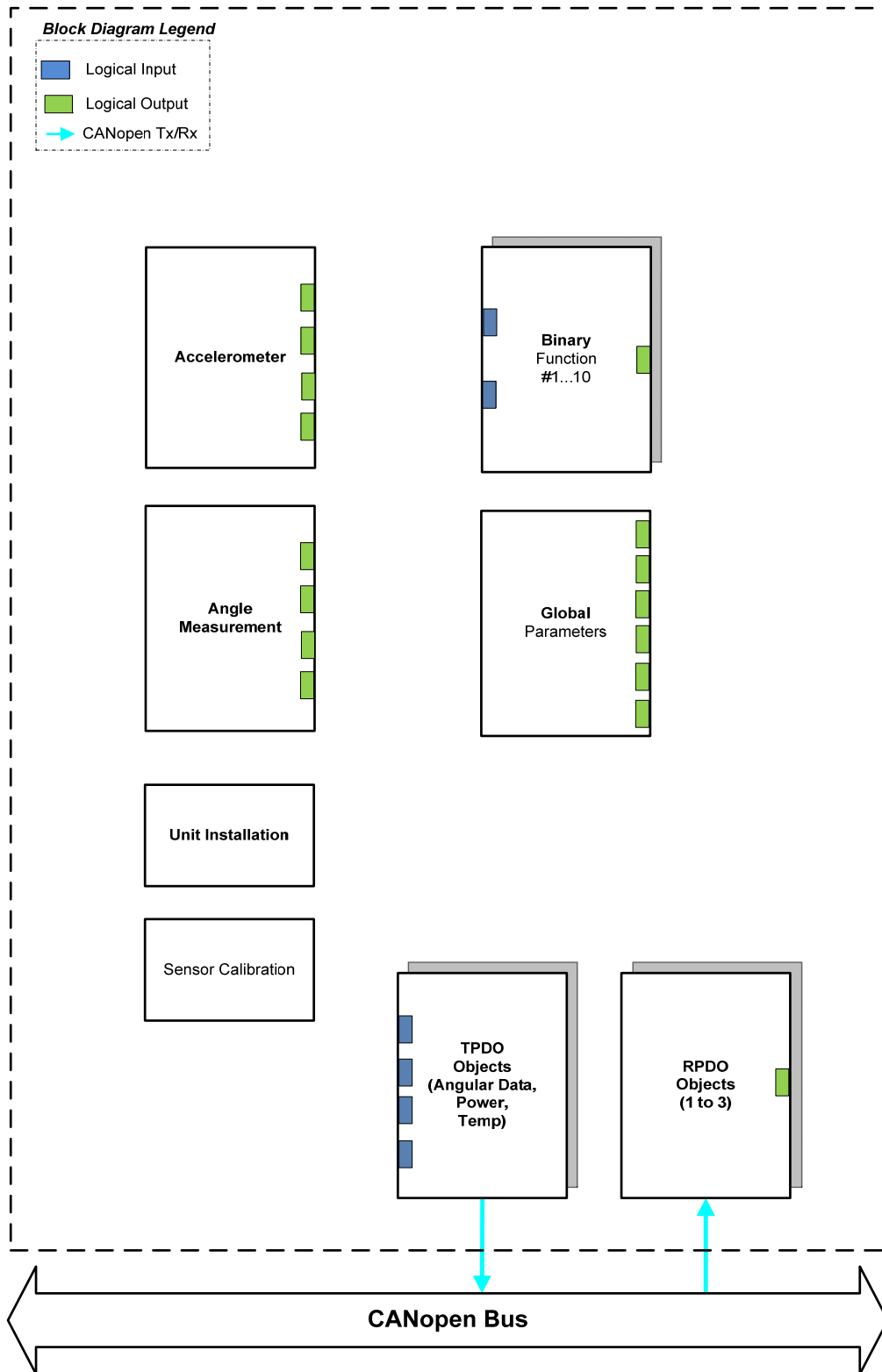


Figure 7. The Inclinometer Logical Block Diagram

Each function block is absolutely independent and has its own set of configuration objects. The objects can be viewed and changed through CANopen protocol.

Accelerometer sensor is presented by *Accelerometer* function block. *Angle Measurement* function block controls measurements of the inclination angles. *Unit Installation* function block is used to compensate installation angles after the unit is mounted at a customer's site. *Sensor Calibration* is an auxiliary function block presenting sensor calibration parameters.

In case the sensor data needs to be processed before been output, the unit has ten *Binary Function* blocks to do simple data conversion operations.

The inclinometer also has *Global Parameters* function block containing four constant output signals and other auxiliary output signals.

### 3.1. Function Block Signals

The inclinometer function blocks can contain signal inputs and outputs to communicate with each other. Each signal input can be connected to any signal output using an appropriate configuration parameter. There is no limitation on the number of signal inputs connected to a signal output.

When a signal input is connected to a signal output, data from the signal output of one function block is available on the signal input of another function block.

The function block signal data can have the following signal types: *{Undefined, Discrete or Continuous}*.

#### 3.1.1. Undefined Signal

The *Undefined* signal type is used to present a no-signal condition in signal data or to specify that the signal input is not connected (not used).

#### 3.1.2. Discrete Signal

The *Discrete* signal type is used to present a discrete signal that has a finite number of states in signal data or to specify that the signal input or output is communicating this type of signals.

The discrete signals are stored in four byte unsigned integer variables that can present any state value in the 0...0xFFFFFFFF range.

#### 3.1.3. Continuous Signal

The *Continuous* signal type presents continuous signals, usually physical parameters, in signal data or as a signal input or output type.

The continuous signals are stored in floating point variables. They are not normalized and present data in the appropriate physical units. The user can do simple scaling of the continuous signal data by changing *Scale (Resolution)* and *Offset* configuration parameters in the appropriate function blocks.

### 3.1.4. Signal Type Conversion

*Discrete* and *Continuous* signals are automatically converted into each other when a signal input of one signal type is connected to a signal output of a different signal type.

### 3.1.5. Discrete to Continuous Conversion

A *Discrete* signal is converted into a positive *Continuous* signal of the same value.

### 3.1.6. Continuous to Discrete Conversion

A positive *Continuous* signal is converted into the same value *Discrete* signal. A fractional part of the *Continuous* signal is truncated. If the *Continuous* signal value is above the maximum *Discrete* signal value, the resulted *Discrete* signal value will saturate to the maximum *Discrete* signal value: 0xFFFFFFFF.

All negative *Continuous* signals are converted into zero value *Discrete* signals.

### 3.1.7. Function Block Signal/Control Source List

Table 1 lists the Function Block Signals/Control Sources that can be selected for any of the application data function blocks.

Table 1. Function Block Signal/Control Source List

Value	Meaning
0	Control Source Not Used (Ignored)
1	Accelerometer X-Axis Data
2	Accelerometer Y-Axis Data
3	Accelerometer Z-Axis Data
4	Accelerometer Sensor Temperature
5	Pitch Angular Data
6	Roll Angular Data
7	Gravity Angular Data
8	Angular Measurement Latency
9	Angular Figure of Merit
10...	Binary Function 1
...19	Binary Function 10
20...	CAN RPDO Message 1
...22	CAN RPDO Message 3
23	Constant Discrete Data
24	Constant Continuous Data
25	Constant Zero/FALSE Signal
26	Constant One/TRUE Signal
27	Power Supply Measured
28	Temperature Measured

In addition to a source, each control also has a number which corresponds to the sub-index of the function block in question. Table 2 outlines the ranges supported for the number objects, depending on the source that had been selected.

Table 2. Function Block Signal/Control Source Range with Respective CANopen Object

Control Source	Range	Object (Meaning)
Control Source Not Used/Undefined	0	Ignored
Accelerometer Data	1	4000h sub-index 1 (X-Axis Data FV)
	2	4000h sub-index 2 (Y-Axis Data FV)
	3	4000h sub-index 3 (Z-Axis Data FV)
Sensor Temperature Measured	N/A	4060h sub-index 0
Angular Measurement Data	N/A	4030h sub-index 0 (Pitch Angle Data FV)
	N/A	4040h sub-index 0 (Roll Angle Data FV)
	N/A	4050h sub-index 0 (Gravity Angle Data FV)
Angular Measurement Latency	N/A	4010h sub-index 0
Angular Figure of Merit	N/A	4020h sub-index 0
Binary Function Block	1	3000h sub-index 1 (Bin Function 1 Out FV)
	2	3000h sub-index 2 (Bin Function 2 Out FV)
	3	3000h sub-index 3 (Bin Function 3 Out FV)
	4	3000h sub-index 4 (Bin Function 4 Out FV)
	5	3000h sub-index 5 (Bin Function 5 Out FV)
	6	3000h sub-index 6 (Bin Function 6 Out FV)
	7	3000h sub-index 7 (Bin Function 7 Out FV)
	8	3000h sub-index 8 (Bin Function 8 Out FV)
	9	3000h sub-index 9 (Bin Function 9 Out FV)
	10	3000h sub-index 10 (Bin Function 10 Out FV)
CANopen Message (RPDO)	1	2500h sub-index 1 (Extra Received PV 1)
	2	2500h sub-index 2 (Extra Received PV 2)
	3	2500h sub-index 3 (Extra Received PV 3)
Global Parameter Function Block	N/A	4070h sub-index 0 (Continuous Signal)
	N/A	4080h sub-index 0 (Discrete Signal)
Power Supply Measured	N/A	5020h (Power Supply FV)
Processor Temperature Measured	N/A	5030h (Processor Temperature FV)

### 3.2. Accelerometer

The *Accelerometer* function block presents the 3D accelerometer sensor.

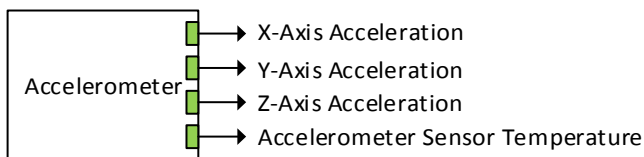


Figure 8. Accelerometer Function Block



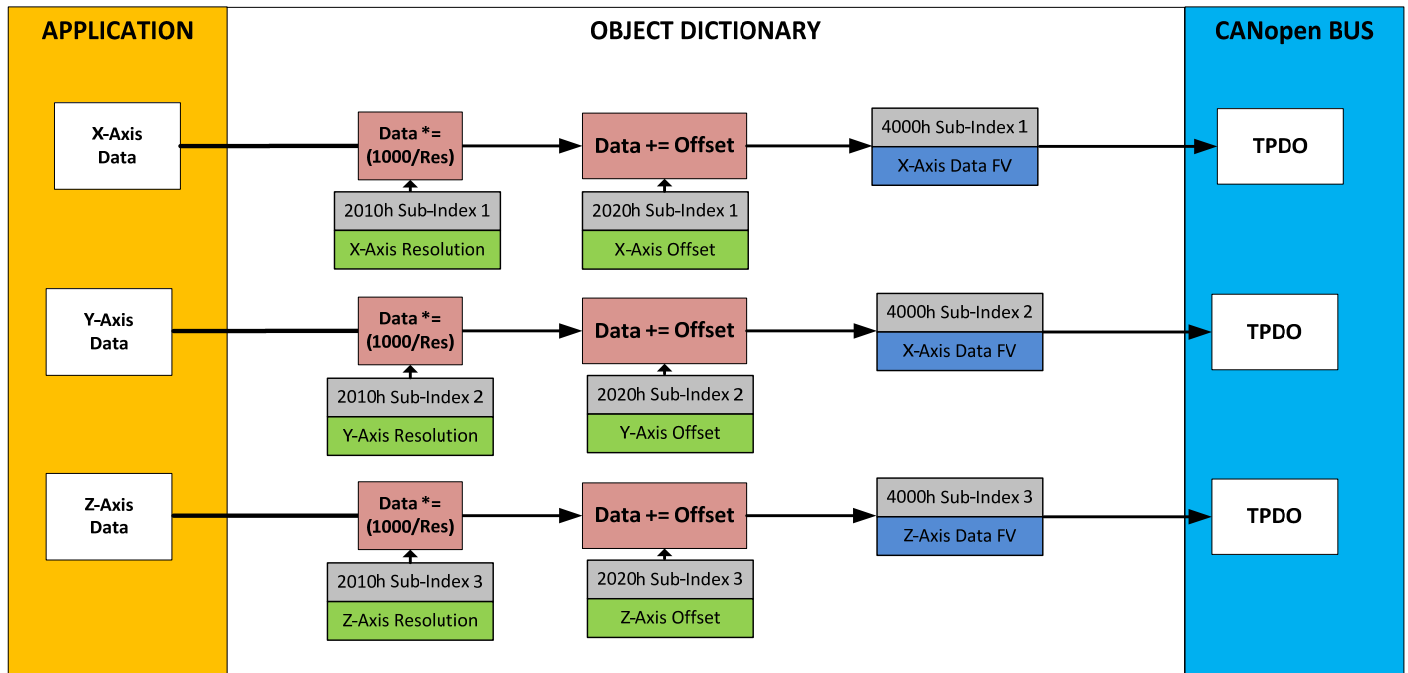


Figure 9. Accelerometer CANopen Object Directory Function Block Diagram

The *Accelerometer* function block has four continuous signal outputs. The unit accelerations: *X-Axis Acceleration*, *Y-Axis Acceleration*, *Z-Axis Acceleration* are presented in the gravity units [g] in the machine frame. Depending on mounting orientation of the controller, all *Axis Accelerations* can be inverted/negated through object 2031h subindex 1-3 for X-Axis, Y-Axis, Z-Axis, respectively.

The *Accelerometer Sensor Temperature* output presents the sensor temperature in [°C].

The *Accelerometer* function block configuration parameters are defined below.

Table 3. Accelerometer Function Block Configuration Parameters

Name	Default Value	Range	Units	Description
Input Filter	On	{Off, On}	–	Low-pass input filter.
Input Filter Cut-Off Frequency	5	[1...50]	Hz	Cut-Off Frequency when <i>Input Filter</i> is <i>On</i> .

### 3.3. Angle Measurement

The *Angle Measurement* function block calculates pitch, roll and gravity angles in the machine frame based on the accelerometer sensor output.

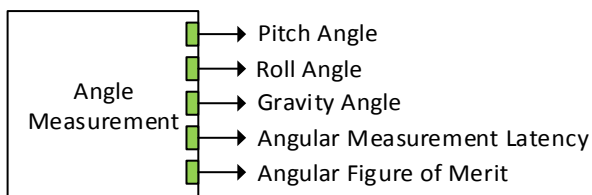


Figure 10. Angle Measurement Function Block

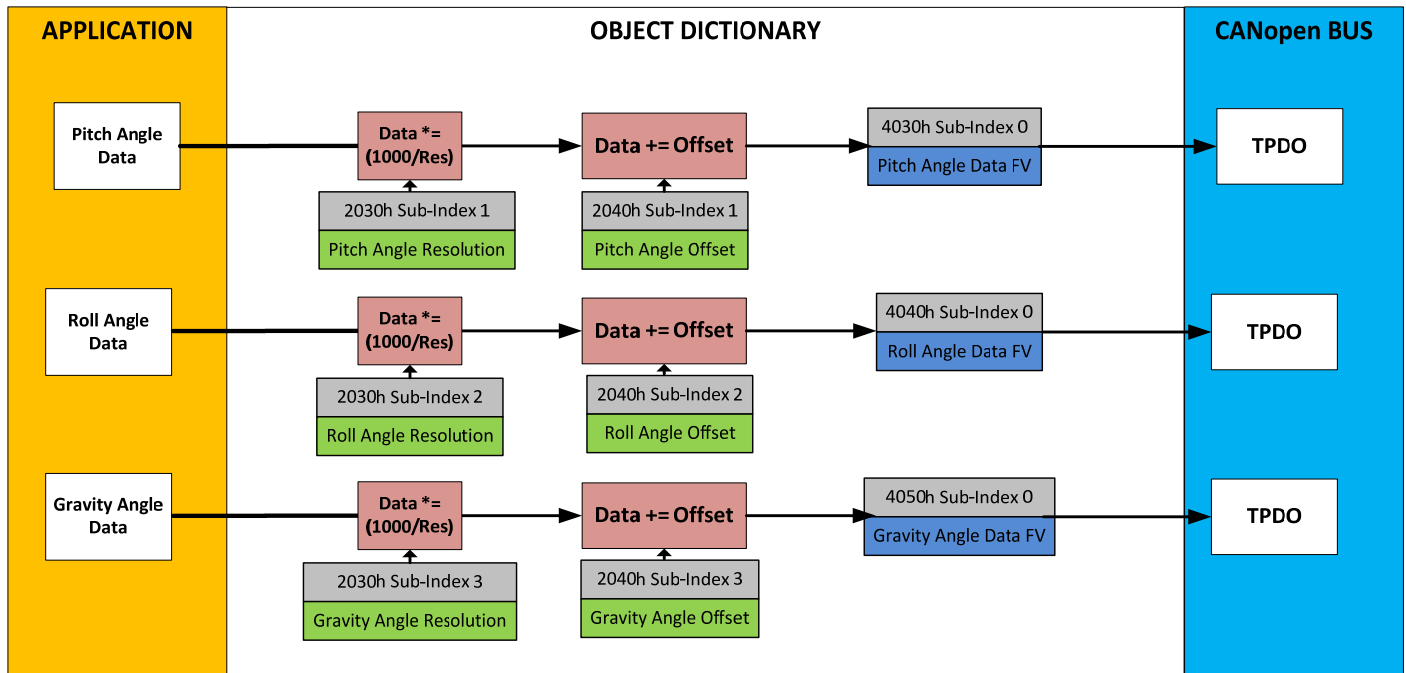


Figure 11. Angular Measurement Data CANopen Object Directory Function Block Diagram

The *Pitch Angle* continuous signal output defines the unit pitch angle in [deg]. It has a full  $\pm 180$  [deg] range by default, when the angle is calculated as a tilt angle. When the angle is calculated as an Euler angle, it has  $\pm 90$  [deg] range. Depending on mounting orientation of the controller, the *Pitch Angle* can be inverted/negated through object 2011h subindex 1.

The *Roll Angle* continuous signal output defines the roll angle in [deg]. It has a full  $\pm 180$  [deg] range independently whether tilt or Euler angle is calculated. Depending on mounting orientation of the controller, the *Roll Angle* can be inverted/negated through object 2011h subindex 2.

The *Gravity Angle* continuous signal output defines the inclination angle of the unit relatively to the gravity vector. It has 0...180 [deg] range. Depending on mounting orientation of the controller, the *Gravity Angle* can be inverted/negated through object 2011h subindex 3.

The *Angular Measurement Latency* continuous signal output defines the angular measurement latency in [ms].

The *Angular Figure of Merit* discrete signal output defines whether the angular output data can be trusted. It has the following set of states:

Table 4. Angular Figure of Merit

State	Description
0	Angular data is fully functional. Data is within the sensor specification.
1	Angular data is suspect due to environmental conditions. Set when the accelerometer sensor temperature is less than $-40^{\circ}\text{C}$ or greater than $+125^{\circ}\text{C}$ .
2	Error condition has been detected.

The *Angle Measurement* function block configuration parameters are presented below:

Table 5. Angle Measurement Function Block Configuration Parameters

Name	Default Value	Range	Units	Description
Pitch and Roll Angle Type	Tilt Angle	{Euler Angle, Tilt Angle}	–	Type the pitch and roll angle.

### 3.4. Unit Installation

The *Unit Installation* function block is used to compensate initial installation angles after the unit is mounted on a machine at a customer’s site.

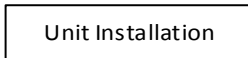


Figure 12. Unit Installation Function Block

The function block has no signal inputs and outputs. Its configuration parameters are presented below.

Table 6. Unit Installation Function Block Configuration Parameters

Name	Default Value	Range	Units	Description
Initial Pitch Angle	0	[-90...90]	Deg	Initial installation pitch angle.
Initial Roll Angle	0	[-180...180]	Deg	Initial installation roll angle.
Auto-Null Command	No <sup>1</sup>	{No, Yes}	–	Auto-Null Command. Set <i>Yes</i> to automatically update the <i>Initial Pitch Angle</i> and <i>Initial Roll Angle</i> .

<sup>1</sup> The Auto-Null Command is not a real configuration parameter. It always returns *No* value when being read.

The initial installation pitch and roll angles are Euler angles. They can be written manually or set up automatically when *Auto-Null Command* is set to *Yes*. These angles are used to transform the unit accelerations from the unit frame to the machine frame.

When the initial installation angles are set up automatically, the user issues the *Auto-Null Command* when the machine is in its original position and the machine frame is coincident with the Earth frame, see: [Unit Reference Frames](#).

The user should avoid the strictly vertical position of the sensor when issuing the *Auto-Null Command* due to the gimbal lock condition, when the *Initial Roll Angle* cannot be defined, see: [Gimbal Lock](#).

### 3.5. Angle Configuration

The angle configuration object allows the inclinometer to provide different measuring ranges as it rotates about each axis. The angle configuration options are presented in the table below:

Table 7. Angle Configuration

Name	Range	Units
180 Degree Range	[-180...180]	Deg
360 Degree Range	[0...360]	Deg

Name	Range	Units
90 Degree Range	[0...90]	Deg

### 3.6. Unit Installation: Vertical and Horizontal Mount

The inclinometer, by default, is configured to be mounted horizontally, parallel to the surface. However, it can also be configured so that it can be mounted vertically. In the case the inclinometer is mounted vertically, the *Angle Configuration* may also be needed to be updated to provide the desired measurements

### 3.7. Sensor Calibration

The *Sensor Calibration* function block presents internal calibration read-only parameters. It does not have any signal inputs and outputs.

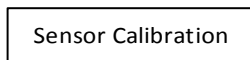


Figure 13. Sensor Calibration

The calibration parameters can be inspected in the field by a qualified technician. They are also written in a setpoint file together with other configuration parameters.

### 3.8. Binary Functions

There are ten *Binary Function* blocks available to the user for performing simple data conversions. Each *Binary Function* block has two continuous signal inputs and one continuous signal output. The continuous output is then scaled into an INTEGER16 data value.

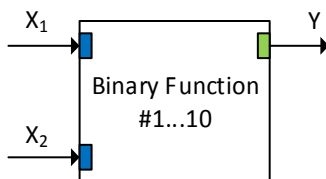


Figure 14. Binary Function Block

The *Binary Function* block performs the following data conversion:

$$Y = A \cdot F[a_1 \cdot f_1(X_1) + b_1; a_2 \cdot f_2(X_2) + b_2] + B, \quad n = 1, 2; \quad (4)$$

- where:
- $X_n$  – Input signal;
  - $f_n(X_n)$  – Unary function;
  - $a_n$  – Scale;
  - $b_n$  – Offset;
  - $F[x; y]$  – Binary Function;
  - $A$  – Output Scale;
  - $B$  – Output Offset.

The function block input signals can be undefined. The user can specify a default signal value that will be used when the signal is not defined. If the default signal value is not specified, the output signal of the function block will become undefined too.

The following unary functions can be used to process the input signals.

*Table 8. Unary Functions*

Function Name	Description	Comment
Undefined	$f(x) = x$	Signal is not processed
! Logical Not	$f(x) = !x$	x is converted into 4-byte unsigned integer before function is applied
~ Bitwise Not	$f(x) = \sim x$	x is converted into 4-byte unsigned integer before function is applied
abs(x) Absolute	$f(x) = x, \text{ if } x \geq 0$ $f(x) = -x, \text{ if } x < 0$	

The following binary functions are defined in the function block:

*Table 9. Binary Functions*

Function Name	Description	Comment
Undefined	$F[x;y] =$ Undefined	Output signal is undefined
+ Addition	$F[x;y] = x + y$	
- Subtraction	$F[x;y] = x - y$	
* Multiplication	$F[x;y] = x * y$	
/ Division	$F[x;y] = x / y$	Division by 0 gives 0
% Modulus	$F[x;y] = x \% y$	x and y are converted into 4-byte unsigned integers before function is applied
max(x,y) Maximum	$F[x;y] = x, \text{ if } x \geq y$ $F[x;y] = y, \text{ if } x < y$	
min(x,y) Minimum	$F[x;y] = x, \text{ if } x \leq y$ $F[x;y] = y, \text{ if } x > y$	
== Equal	$F[x;y] = 1, \text{ if } x = y$ $F[x;y] = 0, \text{ if } x \neq y$	
!= Not Equal	$F[x;y] = 1, \text{ if } x \neq y$ $F[x;y] = 0, \text{ if } x = y$	
> Great	$F[x;y] = 1, \text{ if } x > y$ $F[x;y] = 0, \text{ if } x \leq y$	
>= Great Equal	$F[x;y] = 1, \text{ if } x \geq y$ $F[x;y] = 0, \text{ if } x < y$	
< Less	$F[x;y] = 1, \text{ if } x < y$ $F[x;y] = 0, \text{ if } x \geq y$	
<= Less Equal	$F[x;y] = 1, \text{ if } x \leq y$ $F[x;y] = 0, \text{ if } x > y$	
Logical OR	$F[x;y] = x \vee y$	x and y are converted into 4-byte unsigned integers before function is applied
&& Logical AND	$F[x;y] = x \wedge y$	x and y are converted into 4-byte unsigned integers before function is applied
Bitwise OR	$F[x;y] = x   y$	x and y are converted into 4-byte unsigned integers before function is applied
& Bitwise AND	$F[x;y] = x \& y$	x and y are converted into 4-byte unsigned integers before function is applied

Function Name	Description	Comment
^ Bitwise XOR	$F[x;y] = x \wedge y$	x and y are converted into 4-byte unsigned integers before function is applied
<< Left Shift	$F[x;y] = x \ll y$	x and y are converted into 4-byte unsigned integers before function is applied
>> Right Shift	$F[x;y] = x \gg y$	x and y are converted into 4-byte unsigned integers before function is applied

The *Binary Function* has the following set of configuration parameters:

Table 10. Binary Function Block Configuration Parameters

Name	Default Value <sup>1</sup>	Range	Units	Description
Binary Function	^ Bitwise XOR	See Binary Function table	–	$F[x;y]$ – Binary function
Output Scale	1	Any value	–	A – Output Scale
Output Offset	0	Any value	–	B – Output Offset
Input #1 Signal Source	Pitch Angle Compensation	Any signal output of any function block or “Not Connected”	–	$X_1$ – Input Signal #1
Input #1 Signal Default	No	{No, Yes}	–	Defines whether the default signal value for $X_1$ is defined.
Input #1 Signal Default Value	0	Any value	–	$X_1$ default value, if <i>Input #1 Signal Default</i> is <i>Yes</i> .
Unary Function #1	Undefined	See Unary Function table	–	$f_1(x)$ – Unary function #1
Scale #1	1	Any value	–	$a_1$ – Scale #1
Offset #1	0	Any value	–	$b_1$ – Offset #1
Input #2 Signal Source	Not Connected	Any signal output of any function block or “Not Connected”	–	$X_2$ – Input Signal #2
Input #2 Signal Default	Yes	{No, Yes}	–	Defines whether the default signal value for $X_2$ is defined.
Input #2 Signal Default Value	1	Any value	–	$X_2$ default value, if <i>Input #2 Signal Default</i> is <i>Yes</i> .
Unary Function #2	Undefined	See Unary Function table	–	$f_2(x)$ – Unary function #2
Scale #2	1	Any value	–	$a_2$ – Scale #2
Offset #2	0	Any value	–	$b_2$ – Offset #2

<sup>1</sup>For Binary Function #1.

### 3.9. Global Parameters

The *Global Parameters* functional block gives the user access to a set of global constants, unit supply voltage and the microcontroller internal temperature.

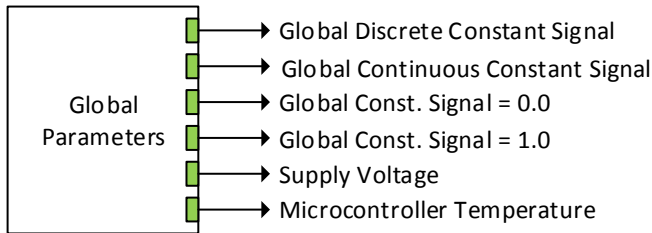


Figure 15. Global Parameters Function Block

The function block has one configurable *Global Discrete Constant Signal* output, one configurable *Global Continuous Constant Signal* output and two continuous pre-set constant signal outputs: *Global Const. Signal = 0.0* and *Global Const. Signal = 1.0*.

The function block also contains *Supply Voltage* continuous signal output presenting the inclinometer supply voltage in [V]. Please note, that this voltage is not the voltage on the inclinometer power supply connector pins. It is an internal voltage measured after the EMI filter, reverse polarity and transient protection circuit. It is always less than the actual power supply voltage by approximately 0.7...0.95 V.

The microcontroller internal temperature is presented on the *Microcontroller Temperature* continuous signal output in [°C].

The *Global Parameters* function block has the following configuration parameters.

Table 11. Global Parameters Function Block Configuration Parameters

Name	Default Value	Range	Units	Description
Global Continuous Constant Signal	0	Any value	–	Output signal value of the <i>Global Continuous Constant Signal</i> .
Global Discrete Constant Signal	0	[0... 4294967295 (0xFFFFFFFF)]	–	Output signal value of the <i>Global Discrete Constant Signal</i> .

No signal outputs from this function block are used in the default inclinometer configurations.

### 3.10. Miscellaneous Function Block

There are some other objects available which have not yet been discussed, or mentioned briefly in passing (i.e. constants.) These objects are not necessarily associated with one another, but are all discussed here.

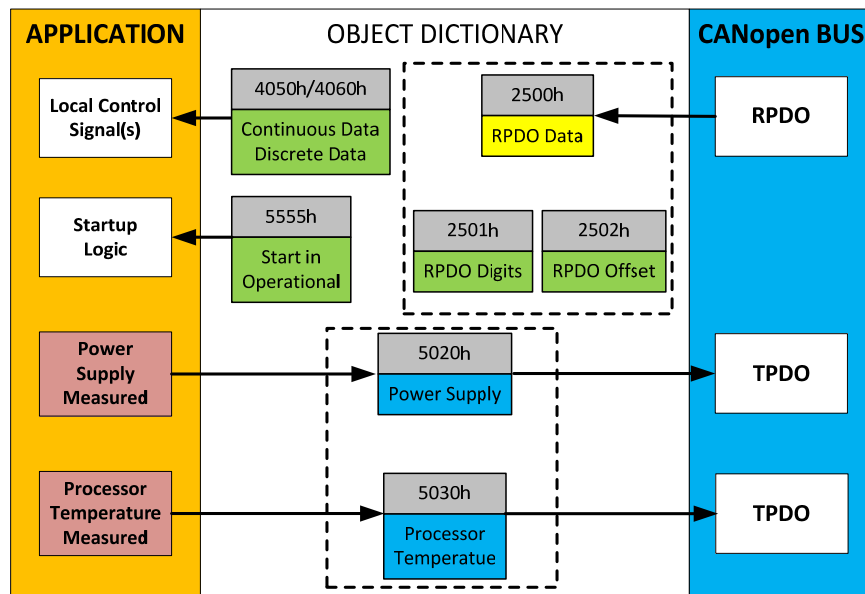


Figure 16. Miscellaneous Objects

### 3.10.1. RPDO Messages

Objects 2500h **RPDO Data Received PV**, 2501h **RPDO Data Received Resolution PV** and 2502h **RPDO Data Received Offset PV** have been mentioned in Section 3.1.7, Table 2. These objects allow for data received on a CANopen ® RPDO to be mapped independently to various function blocks as a control source. For example, a Binary Function is used to compare Pitch Angle to a target/limit angle from another source from the CAN bus.

### 3.10.2. Automatic Update of Objects

Object 5550h **Enable Automatic Updates** allows for the controller to automatically update the objects related to the output to defaults when it is changed. By default, this object is set to TRUE, in which case the objects are set to their default values depending on the type selected.

On the other hand, when this object is FALSE, the objects are not set to defaults and are left with the same values previous to changing the type. In this case, these are to be configured manually.

### 3.10.3. Startup

The last object 5555h **Start in Operational** is provided as a ‘cheat’ when the unit is not intended to work with a CANopen network (i.e. a stand-alone control), or is working on a network comprised solely as slaves so the OPERATION command will never be received from a master. By default this object is disabled (FALSE).

When using the Triaxial Inclinometer as a stand-alone controller where 5555h is set to TRUE, it is recommended to disable all TPDOs (set the Event Timer to zero) so that it does not run with a continuous CAN error when not connected to a bus.



## 4. CANOPEN ® OBJECT DICTIONARY

---

The CANopen object dictionary of the Triaxial Inclinometer Controller is based on Manufacturer-Specific objects for functionalities relating to Axiomatic's triaxial inclinometer. The object dictionary includes Communication Objects beyond the minimum requirements in the profile, as well as several manufacturer-specific objects for extended functionality.

### 4.1. NODE ID and BAUDRATE

By default, the Triaxial Inclinometer Controller ships factory programmed with a Node ID = 127 (0x7F) and with Baudrate = 125 kbps.

#### 4.1.1. LSS Protocol to Update

The only means by which the Node-ID and Baudrate can be changed is to use Layer Settling Services (LSS) and protocols as defined by CANopen ® standard DS-305.

Follow the steps below to configure either variable using LSS protocol. If required, please refer to the standard for more detailed information about how to use the protocol.

##### 4.1.1.1. Setting Node-ID

- Set the module state to LSS-configuration by **sending** the following message:

<i>Item</i>	<i>Value</i>
COB-ID	0x7E5
Length	2
Data 0	0x04 (cs=4 for switch state global)
Data 1	0x01 (switches to configuration state)

- Set the Node-ID by **sending** the following message:

<i>Item</i>	<i>Value</i>
COB-ID	0x7E5
Length	2
Data 0	0x11 (cs=17 for configure node-id)
Data 1	Node-ID (set new Node-ID as a hexadecimal number)

- The module will send the following response (any other response is a failure).

<i>Item</i>	<i>Value</i>
COB-ID	0x7E4
Length	3
Data 0	0x11 (cs=17 for configure node-id)
Data 1	0x00
Data 2	0x00

- Save the configuration by **sending** the following message:

<i>Item</i>	<i>Value</i>
-------------	--------------

COB-ID	0x7E5
Length	1
Data 0	0x17 (cs=23 for store configuration)

- The module will send the following response (any other response is a failure):

<i>Item</i>	<i>Value</i>
COB-ID	0x7E4
Length	3
Data 0	0x17 (cs=23 for store configuration)
Data 1	0x00
Data 2	0x00

- Set the module state to LSS-operation by **sending** the following message:  
(Note, the module will reset itself back to the pre-operational state)

<i>Item</i>	<i>Value</i>
COB-ID	0x7E5
Length	2
Data 0	0x04 (cs=4 for switch state global)
Data 1	0x00 (switches to waiting state)

#### 4.1.1.2. Setting Baud Rate

- Set the module state to LSS-configuration by **sending** the following message:

<i>Item</i>	<i>Value</i>
COB-ID	0x7E5
Length	2
Data 0	0x04 (cs=4 for switch state global)
Data 1	0x01 (switches to configuration state)

- Set the baudrate by **sending** the following message:

<i>Item</i>	<i>Value</i>
COB-ID	0x7E5
Length	3
Data 0	0x13 (cs=19 for configure bit timing parameters)
Data 1	0x00 (switches to waiting state)
Data 2	Index (select baudrate index per Table 21)

<i>Index</i>	<i>Bit Rate</i>
<b>0</b>	1 Mbit/s
<b>1</b>	800 kbit/s
<b>2</b>	500 kbit/s
<b>3</b>	250 kbit/s
<b>4</b>	125 kbit/s (default)
<b>5</b>	reserved (100 kbit/s)
<b>6</b>	50 kbit/s

<b>7</b>	20 kbit/s
<b>8</b>	10 kbit/s

**Table 21 – LSS Baudrate Indices**

- The module will send the following response (any other response is a failure):

<i>Item</i>	<i>Value</i>
COB-ID	0x7E4
Length	3
Data 0	0x13 (cs=19 for configure bit timing parameters)
Data 1	0x00
Data 2	0x00

- Activate bit timing parameters by **sending** the following message:

<i>Item</i>	<i>Value</i>
COB-ID	0x7E5
Length	3
Data 0	0x15 (cs=19 for activate bit timing parameters)
Data 1	<delay_lsb>
Data 2	<delay_msb>

The delay individually defines the duration of the two periods of time to wait until the bit timing parameters switch is done (first period) and before transmitting any CAN message with the new bit timing parameters after performing the switch (second period). The time unit of switch delay is 1 ms.

- Save the configuration by **sending** the following message (on the NEW baudrate):

<i>Item</i>	<i>Value</i>
COB-ID	0x7E5
Length	1
Data 0	0x17 (cs=23 for store configuration)

- The module will send the following response (any other response is a failure):

<i>Item</i>	<i>Value</i>
COB-ID	0x7E4
Length	3
Data 0	0x17 (cs=23 for store configuration)
Data 1	0x00
Data 2	0x00

- Set the module state to LSS-operation by **sending** the following message:  
(Note, the module will reset itself back to the pre-operational state)

<i>Item</i>	<i>Value</i>
COB-ID	0x7E5
Length	2
Data 0	0x04 (cs=4 for switch state global)

Data 1	0x00	(switches to waiting state)
--------	------	-----------------------------

The following screen capture (left) shows the CAN data was sent (7E5h) and received (7E4h) by the tool when the baudrate was changed to 250 kbps using the LSS protocol. The other image (right) shows what was printed on an example debug RS-232 menu while the operation took place.

Between CAN Frame 98 and 99, the baudrate on the CAN Scope tool was changed from 125 to 250 kbps.

**Net0 | CAN USB331 | 250 - CANscope**

File Can Help

Add/Delete ID Area

from to Add > 0x000 <> 0x7FF Net: 0 - CAN\_USB331

< Del Baud rate: 250

IDs decimal  29 >

Fram...	Absolute Time	RelTime	Id	Atr	L	d1	d2	d3	d4
95	11:42:45.248	6110	07E5		2	04	01		
96	11:42:54.468	9219	07E5		3	13	00	03	
97	11:42:54.468	0	07E4		3	13	00	00	
98	11:42:58.687	4218	07E5		3	15	88	13	
99	11:43:16.579	17891	07E5		1	17			
100	11:43:16.907	328	07E4		3	17	00	00	
101	11:43:23.017	6109	07E5		2	04	00		
102	11:43:23.017	0	0750		1	00			

Send ID: 29-Bit RTR Len: Data\$:

Fill:102(10.2%) Bus:ok STARTED

File Edit Setup Control Window Help

===== **Main Menu** =====

**Choose one of the following:**

**U: View Object Dictionary**

**D: Default Object Dictionary**

**T: Toggle RS-232 Stream On/Off**

**S: Show/Stop Diagnostics**

**L: Load New Software**

**M: Main Menu <this>**

->Node Id = 80

->Baudrate= 125 [kbps]

CO: PRE-OPERATIONAL

Activating new baud = 250 [kbps]

CO: STOP

Restarting CAN in 5000 [ms]

CO: PRE-OPERATIONAL

Storing ID

Storing Factory Parameters

Storing Baud

Storing Factory Parameters

Storing Communication Parameters

->Node Id = 80

->Baudrate= 250 [kbps]

CO: PRE-OPERATIONAL

## 4.2. COMMUNICATION OBJECTS (DS-301)

The communication objects supported by the Triaxial Inclinometer Controller are listed in the following table. A more detailed description of some of the objects is given in the following subchapters. Only those objects that have device-profile specific information are described. For more information on the other objects, refer to the generic CANopen protocol specification DS-301.

<b>Index (hex)</b>	<b>Object</b>	<b>Object Type</b>	<b>Data Type</b>	<b>Access</b>	<b>PDO Mapping</b>
1000	Device Type	VAR	UNSIGNED32	RO	No
1001	Error Register	VAR	UNSIGNED8	RO	No
1002	Manufacturer Status Register	VAR	UNSIGNED32	RO	No
1003	Pre-Defined Error Field	ARRAY	UNSIGNED32	RO	No
100C	Guard Time	VAR	UNSIGNED16	RW	No
100D	Life Time Factor	VAR	UNSIGNED8	RW	No
1010	Store Parameters	ARRAY	UNSIGNED32	RW	No
1011	Restore Default Parameters	ARRAY	UNSIGNED32	RW	No
1016	Consumer Heartbeat Time	ARRAY	UNSIGNED32	RW	No
1017	Producer Heartbeat Time	VAR	UNSIGNED16	RW	No
1018	Identity Object	RECORD		RO	No
1020	Verify Configuration	ARRAY	UNSIGNED32	RO	No
1029	Error Behaviour	ARRAY	UNSIGNED8	RW	No
1400	RPDO1 Communication Parameter	RECORD		RW	No
1401	RPDO2 Communication Parameter	RECORD		RW	No
1402	RPDO3 Communication Parameter	RECORD		RW	No
1403	RPDO4 Communication Parameter	RECORD		RW	No
1600	RPDO1 Mapping Parameter	RECORD		RO	No
1601	RPDO2 Mapping Parameter	RECORD		RO	No
1602	RPDO3 Mapping Parameter	RECORD		RO	No
1603	RPDO4 Mapping Parameter	RECORD		RO	No
1800	TPDO1 Communication Parameter	RECORD		RW	No
1801	TPDO2 Communication Parameter	RECORD		RW	No
1802	TPDO3 Communication Parameter	RECORD		RW	No
1803	TPDO4 Communication Parameter	RECORD		RW	No
1A00	TPDO1 Mapping Parameter	RECORD		RW	No
1A01	TPDO2 Mapping Parameter	RECORD		RW	No
1A02	TPDO3 Mapping Parameter	RECORD		RW	No
1A03	TPDO4 Mapping Parameter	RECORD		RW	No

#### 4.2.1. Object 1000h: Device Type

This object contains information about the device type as per a device profile it follows. The Triaxial Inclinometer does not follow a standardized device profile for being a triaxial inclinometer. The 32-bit parameter is divided into two 16-bit values, showing General and Additional information as shown below.

MSB	LSB
<b>Additional Information</b> = 0x0000	<b>General Information</b> = 0x0000 (non-standard)

The triaxial inclinometer is composed of Manufacturer-Specific objects to encompass the device's features and capabilities of a 3-Axis MEMS controller.

##### ***Object Description***

Index	1000h
Name	Device Type
Object Type	VAR
Data Type	UNSIGNED32

##### ***Entry Description***

Access	RO
PDO Mapping	No
Value Range	0x00000000
Default Value	0x00000000

#### 4.2.2. Object 1001h: Error Register

This object is an error register for the device. Any time there is an error detected by the TRIAXIAL INCLINOMETER Controller, the Generic Error Bit (bit 0) is set. Only if there is no errors in the module will this bit will be cleared. No other bits in this register are used by the TRIAXIAL INCLINOMETER Controller.

##### ***Object Description***

Index	1001h
Name	Error Register
Object Type	VAR
Data Type	UNSIGNED8

##### ***Entry Description***

Access	RO
PDO Mapping	No
Value Range	00h or 01h
Default Value	0

#### 4.2.3. Object 1002h: Manufacturer Status Register

This object is used for manufacturer debug purposes.

#### 4.2.4. Object 1003h: Pre-Defined Error Field

This object provides an error history by listing the errors in the order that they have occurred. An error is added to the top of the list when it occurs, and is immediately removed when the error condition has been cleared. The latest error is always at sub-index 1, with sub-index 0 containing the number of errors currently in the list. When the device is in an error-free state, the value of sub-index 0 is zero.

The error list may be cleared by writing a zero to sub-index 0, which will clear all errors from the list, regardless of whether or not they are still present. Clearing the list does NOT mean that the module will return to the error-free behaviour state if at least one error is still active.

The Triaxial Inclinometer Controller has a limitation of a maximum of 4 errors in the list. If the device registers more errors, the list will be truncated, and the oldest entries will be lost.

The error codes stored in the list are 32-bit unsigned numbers, consisting of two 16-bit fields. The lower 16-bit field is the EMCY error code, and the higher 16-bit field is a manufacturer-specific code. The manufacturer-specific code is divided into two 8-bit fields, with the higher byte indicating the error description, and the lower byte indicating the channel on which the error occurred.

MSB	LSB
<b>Error Description</b>	<b>Channel-ID</b>
	<b>EMCY Error Code</b>

If node-guarding is used (not recommended per the latest standard) and a lifeguard event occurs, the manufacturer-specific field will be set to 0x1000. On the other hand, if a heartbeat consumer fails to be received within the expected timeframe, the Error Description will be set to 0x80 and the Channel-ID (nn) will reflect the Node-ID of the consumer channel that was not producing. In this case, the manufacturer-specific field will therefore be 0x80nn. In both cases, the corresponding EMCY Error Code will be the Guard Error 0x8130.

When an analog output is not working as described in Section 1.3, then the Error Description will reflect what channel(s) is at fault using the following table. Also, if an RPDO is not received within the expected “Event Timer” period, an RPDO timeout will be flagged. Table 22 outlines the resulting Error Field Codes and their meanings.

Code	Meaning
0000h	Error Reset (No Errors)
4200h	Temperature Sensor Error
8110h	CAN Overrun
8130h	Lifeguard/Heartbeat Error
8140h	Recover from Bus Off Error
FF00h	Accelerometer Sensor Error

**Table 22 – Pre-Defined Error Field Codes**

**Object Description**

Index	1003h
Name	Pre-Defined Error Field
Object Type	ARRAY
Data Type	UNSIGNED32

**Entry Description**

Sub-Index	0h
Description	Number of entries
Access	RW
PDO Mapping	No

Value Range	0 to 15
Default Value	0

Sub-Index	1h to 15
Description	Standard error field
Access	RO
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	0

#### 4.2.5. Object 100Ch: Guard Time

The objects at index 100Ch and 100Dh shall indicate the configured guard time respective to the life time factor. The life time factor multiplied with the guard time gives the life time for the life guarding protocol described in DS-301. The Guard Time value shall be given in multiples of ms, and a value of 0000h shall disable the life guarding.

It should be noted that this object, and that of 100Dh are only supported for backwards compatibility. The standard recommends that newer networks do not use the life guarding protocol, but rather heartbeat monitoring instead. Both life guarding and heartbeats can NOT be active simultaneously.

##### ***Object Description***

Index	100Ch
Name	Guard Time
Object Type	VAR
Data Type	UNSIGNED16

##### ***Entry Description***

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	0 to 65535
Default Value	0

#### 4.2.6. Object 100Dh: Lifetime Factor

The life time factor multiplied with the guard time gives the life time for the life guarding protocol. A value of 00h shall disable life guarding.

##### ***Object Description***

Index	100Dh
Name	Life time factor
Object Type	VAR
Data Type	UNSIGNED8

##### ***Entry Description***

Sub-Index	0h
Access	RW



PDO Mapping	No
Value Range	0 to 255
Default Value	0

#### 4.2.7. Object 1010h: Store Parameters

This object supports the saving of parameters in non-volatile memory. In order to avoid storage of parameters by mistake, storage is only executed when a specific signature is written to the appropriate sub-index. The signature is “save”.

The signature is a 32-bit unsigned number, composed of the ASCII codes of the signature characters, according to the following table:

MSB		LSB	
e	v	a	s
65h	76h	61h	73h

On reception of the correct signature to an appropriate sub-index, the TRIAXIAL INCLINOMETER Controller will store the parameters in non-volatile memory, and then confirm the SDO transmission.

By read access, the object provides information about the module’s saving capabilities. For all sub-indexes, this value is 1h, indicating that the Triaxial Inclinometer Controller saves parameters on command. **This means that if power is removed before the Store object is written, changes to the Object Dictionary will NOT have been saved in the non-volatile memory, and will be lost on the next power cycle.**

#### ***Object Description***

Index	1010h
Name	Store Parameters
Object Type	ARRAY
Data Type	UNSIGNED32

#### ***Entry Description***

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	8
Default Value	8

Sub-Index	1h
Description	Save all parameters
Access	RW
PDO Mapping	No
Value Range	0x65766173 (write access) 1h (read access)
Default Value	1h

Sub-Index	2h
Description	Save communication parameters

Access	RW
PDO Mapping	No
Value Range	0x65766173 (write access) 1h (read access)
Default Value	1h

Sub-Index	3h
Description	Save manufacturer parameters
Access	RW
PDO Mapping	No
Value Range	0x65766173 (write access) 1h (read access)
Default Value	1h

#### 4.2.8. Object 1011h: Restore Parameters

This object supports the restoring of the default values for the object dictionary in non-volatile memory. In order to avoid restoring of parameters by mistake, the device restores the defaults only when a specific signature is written to the appropriate sub-index. The signature is “load”.

The signature is a 32-bit unsigned number, composed of the ASCII codes of the signature characters, according to the following table:

MSB		LSB	
d	a	o	l
64h	61h	6Fh	6Ch

On reception of the correct signature to an appropriate sub-index, the Triaxial Inclinometer Controller will restore the defaults in non-volatile memory, and then confirm the SDO transmission. **The default values are set valid only after the device is reset or power-cycled.** This means that the Triaxial Inclinometer Controller will NOT start using the default values right away, but rather continue to run from whatever values were in the Object Dictionary prior to the restore operation.

By read access, the object provides information about the module’s default parameter restoring capabilities. For all sub-indexes, this value is 1h, indicating that the Triaxial Inclinometer Controller restores defaults on command.

#### ***Object Description***

Index	1011h
Name	Restore Default Parameters
Object Type	ARRAY
Data Type	UNSIGNED32

#### ***Entry Description***

Sub-Index	0h
Description	Largest sub-index supported
Access	RO
PDO Mapping	No
Value Range	4
Default Value	4

Sub-Index	1h
Description	Restore all default parameters
Access	RW
PDO Mapping	No
Value Range	0x64616F6C (write access), 1h (read access)
Default Value	1h

Sub-Index	2h
Description	Restore default communication parameters
Access	RW
PDO Mapping	No
Value Range	0x64616F6C (write access), 1h (read access)
Default Value	1h

Sub-Index	3h
Description	Restore default manufacturer parameters
Access	RW
PDO Mapping	No
Value Range	0x64616F6C (write access), 1h (read access)
Default Value	1h

#### 4.2.9. Object 1016h: Consumer Heartbeat Time

The Triaxial Inclinometer Controller can be a consumer of heartbeat objects for a single module. This object defines the expected heartbeat cycle time for that module, and if set to zero, it is not used. When the object is non-zero, the time is a multiple of 1ms, and monitoring will start after the reception of the first heartbeat from the module. If the Triaxial Inclinometer Controller fails to receive a heartbeat from a node in the expected timeframe, it will indicate a communication error, and respond as per object 1029h.

Bits	31-24	23-16	15-0
Value	Reserved 00h	Node-ID	Heartbeat time
Encoded as		UNSIGNED8	UNSIGNED16

#### ***Object Description***

Index	1016h
Name	Consumer heartbeat time
Object Type	ARRAY
Data Type	UNSIGNED32

#### ***Entry Description***

Sub-Index	0h
Description	Number of entries
Access	RO
PDO Mapping	No
Value Range	1
Default Value	1

Sub-Index	1h
-----------	----

Description	Consumer heartbeat time
Access	RW
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	0

#### 4.2.10. Object 1017h: Producer Heartbeat Time

The Triaxial Inclinometer Controller could be configured to produce a cyclical heartbeat by writing a non-zero value to this object. The value will be given in multiples of 1ms, and a value of 0 shall disable the heartbeat.

##### ***Object Description***

Index	1017h
Name	Producer heartbeat time
Object Type	VAR
Data Type	UNSIGNED16

##### ***Entry Description***

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	10 to 65535
Default Value	0

#### 4.2.11. Object 1018h: Identity Object

The identity object indicates the data of the Triaxial Inclinometer Controller, including vendor id, device id, software and hardware version numbers, and the serial number.

In the Revision Number entry at sub-index 3, the format of the data is as shown below

MSB	LSB
Major revision number (object dictionary)	Hardware Revision      Software Version

##### ***Object Description***

Index	1018h
Name	Identity Object
Object Type	RECORD
Data Type	Identity Record

##### ***Entry Description***

Sub-Index	0h
Description	Number of entries
Access	RO
PDO Mapping	No
Value Range	4
Default Value	4

Sub-Index	1h
-----------	----

Description	Vendor ID
Access	RO
PDO Mapping	No
Value Range	0x00000055
Default Value	0x00000055 (Axiomatic)

Sub-Index	2h
Description	Product Code
Access	RO
PDO Mapping	No
Value Range	0xAA060851
Default Value	0xAA060851

Sub-Index	3h
Description	Revision Number
Access	RO
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	0x00010101

Sub-Index	4h
Description	Serial Number
Access	RO
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	No

#### 4.2.12. Object 1020h: Verify Configuration

This object can be read to see what date the software (version identified in object 1018h) was compiled. The date is represented as a hexadecimal value showing day/month/year as per the format below. The time value at sub-index 2 is a hexadecimal value showing the time in a 24 hour clock

MSB		LSB
Day (in 1-Byte Hex)	Month (in 1-Byte Hex)	Year (in 2-Byte Hex)
00	00	Time (in 2-Byte Hex)

For example, a value of 0x30042014 would indicate that the software was compiled on April 30<sup>th</sup>, 2014. A time value of 0x00001842 would indicate it was compiled at 6:42pm.

##### **Object Description**

Index	1020h
Name	Verify configuration
Object Type	ARRAY
Data Type	UNSIGNED32

##### **Entry Description**

Sub-Index	0h
Description	Number of entries

Access	RO
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1h
Description	Configuration date
Access	RO
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	No

Sub-Index	2h
Description	Configuration time
Access	RO
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	No

#### 4.2.13. Object 1029h: Error Behaviour

This object controls the state that the Triaxial Inclinometer Controller will be set into in case of an error of the type associated with the sub-index.

A network fault is flagged when an RPDO is not received within the expected time period defined in the “Event Timer” of the associated communication objects, (see Section 2.2.14 for more information) or if a lifeguard or heartbeat message is not received as expected.

For all sub-indexes, the following range and definitions hold true:

- 0 = Pre-Operational (node reverts to a pre-operational state when this fault is detected)
- 1 = No State Change (node remains in the same state it was in when the fault occurred)
- 2 = Stopped (node goes into stopped mode when the fault occurs)

#### ***Object Description***

Index	1029h
Name	Error Behaviour
Object Type	ARRAY
Data Type	UNSIGNED8

#### ***Entry Description***

Sub-Index	0h
Description	Number of entries
Access	RO
PDO Mapping	No
Value Range	5
Default Value	5

Sub-Index	1h
Description	Communication Fault
Access	RW

PDO Mapping	No
Value Range	See above
Default Value	0 (Pre-Operational)

Sub-Index	2h
Description	Sensor Error
Access	RW
PDO Mapping	No
Value Range	See above
Default Value	0 (Pre-Operational)

#### 4.2.14. RPDO Behaviour

Per the CANopen ® standard DS-301, the following procedure shall be used for re-mapping, and is the same for both RPDOs and TPDOs.

- Destroy the PDO by setting bit **exists** (most significant bit) of sub-index 01h of the according PDO communication parameter to 1b
- Disable mapping by setting sub-index 00h of the corresponding mapping object to 0
- Modify the mapping by changing the values of the corresponding sub-indices
- Enable mapping by setting sub-index 00h to the number of mapped objects
- Create the PDO by setting bit **exists** (most significant bit) of sub-index 01h of the according PDO communication parameter to 0b

The Triaxial Inclinometer Controller can support up to four RPDO messages. All RPDOs on the Triaxial Inclinometer Controller use the similar default communication parameters, with the PDO IDs set according to the pre-defined connection set described in DS-301. Most RPDOs do not exist, there is no RTR allowed, they use 11-bit CAN-IDs (base frame valid) and they are all event-driven. While all six have valid default mappings defined (see below) only RPDO1 is enabled by default (i.e. RPDO exists).

##### ***RPDO1 Mapping at Object 1600h: Default ID 0x200 + Node ID***

<b><i>Sub-Index</i></b>	<b><i>Value</i></b>	<b><i>Object</i></b>
0	3	Number of mapped application objects in PDO
1	0x25000110	Received Message 1 PV
2	0x25000210	Received Message 2 PV
3	0x25000310	Received Message 3 PV
4	0	Not used by default

##### ***RPDO2 Mapping at Object 1601h: Default ID 0x300 + Node ID***

<b><i>Sub-Index</i></b>	<b><i>Value</i></b>	<b><i>Object</i></b>
0	0	Number of mapped application objects in PDO
1	0	Not used by default
2	0	Not used by default
3	0	Not used by default
4	0	Not used by default

##### ***RPDO3 Mapping at Object 1602h: Default ID 0x400 + Node ID***

<b><i>Sub-Index</i></b>	<b><i>Value</i></b>	<b><i>Object</i></b>
0	0	Number of mapped application objects in PDO
1	0	Not used by default

2	0	Not used by default
3	0	Not used by default
4	0	Not used by default

**RPDO4 Mapping at Object 1603h: Default ID 0x500 + Node ID**

Sub-Index	Value	Object
0	0	Number of mapped application objects in PDO
1	0	Not used by default
2	0	Not used by default
3	0	Not used by default
4	0	Not used by default

None of them have the timeout feature enabled, i.e. the “Event Timer” on sub-index 5 is set to zero. When this is changed to a non-zero value, if the RPDO has not been received from another node within the time period defined (while in Operational mode), a network fault is activated, and the controller will go to the operational state define in Object 1029h sub-index 4.

**Object Description**

Index	1400h to 1403h
Name	RPDO communication parameter
Object Type	RECORD
Data Type	PDO Communication Record

**Entry Description**

Sub-Index	0h
Description	Number of entries
Access	RO
PDO Mapping	No
Value Range	5
Default Value	5

Sub-Index	1h
Description	COB-ID used by RPDO
Access	RW
PDO Mapping	No
Value Range	See value definition in DS-301

X	RPDOx ID
1	0200h
2	0300h
3	0400h
4	0500h

Node-ID = Node-ID of the module. The RPDO COB-IDs are automatically updated if the Node-ID is changed by LSS protocol.

80000000h in the COB-ID indicates that the PDO does not exist (destroyed)

04000000h in the COB-ID indicates that there is no RTR allowed on the PDO

Sub-Index	2h
Description	Transmission type
Access	RO
PDO Mapping	No
Value Range	See value definition in DS-301
Default Value	255 (FFh) = Event Driven



Sub-Index	3h
Description	Inhibit Time
Access	RW
PDO Mapping	No
Value Range	See value definition in DS-301
Default Value	0

Sub-Index	4h
Description	Compatibility entry
Access	RW
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	0

Sub-Index	5
Description	Event-timer
Access	RW
PDO Mapping	No
Value Range	See value definition in DS-301
Default Value	0

*Recall: A non-zero event timer for an RPDO means that it will result in a network fault being flagged if it has not been received within this timeframe while in Operational mode.*

#### 4.2.15. TPDO Behaviour

The Triaxial Inclinometer Controller can support up to four TPDO messages. All TPDOs on the Triaxial Inclinometer Controller use the similar default communication parameters, with the PDO IDs set according to the pre-defined connection set described in DS-301. Most TPDOs do not exist, there is no RTR allowed, they use 11-bit CAN-IDs (base frame valid) and they are all time-driven. While all have valid default mappings defined (see below) only TPDO1 to TPDO3 are enabled by default (i.e. TPDO exists).

##### ***TPDO1 Mapping at Object 1A00h: Default ID 0x180 + Node ID***

<b><i>Sub-Index</i></b>	<b><i>Value</i></b>	<b><i>Object</i></b>
0	4	Number of mapped application objects in PDO
1	0x40400010	Angular Data Roll Angle Field Value
2	0x40300010	Angular Data Pitch Angle Field Value
3	0x40500010	Angular Data Gravity Angle Field Value
4	0x40200008	Angular Data Figure of Merit

##### ***TPDO2 Mapping at Object 1A01h: Default ID 0x280 + Node ID***

<b><i>Sub-Index</i></b>	<b><i>Value</i></b>	<b><i>Object</i></b>
0	4	Number of mapped application objects in PDO
1	0x40000110	Accelerometer X-Axis Data Field Value
2	0x40000210	Accelerometer Y-Axis Data Field Value
3	0x40000310	Accelerometer Z-Axis Data Field Value
4	0x40600010	Accelerometer Sensor Temperature

##### ***TPDO3 Mapping at Object 1A02h: Default ID 0x380 + Node ID***

<b>Sub-Index</b>	<b>Value</b>	<b>Object</b>
0	2	Number of mapped application objects in PDO
1	0x50200020	Power Supply Field Value (measured)
2	0x50300020	Processor Temperature Field Value (measured)
3	0	Not used by default
4	0	Not used by default

**TPDO4 Mapping at Object 1A03h: Default ID 0x480 + Node ID**

<b>Sub-Index</b>	<b>Value</b>	<b>Object</b>
0	0	Number of mapped application objects in PDO
1	0	Not used by default
2	0	Not used by default
3	0	Not used by default
4	0	Not used by default

Since only TPDO1 to TPDO2 have a non-zero value transmission rate (i.e. Event Timer in sub-index 5 of communication object), only these TPDOs will be automatically broadcasted when the unit goes into OPERATIONAL mode.

**Object Description**

Index	1800h to 1803h
Name	TPDO communication parameter
Object Type	RECORD
Data Type	PDO Communication Record

**Entry Description**

Sub-Index	0h
Description	Number of entries
Access	RO
PDO Mapping	No
Value Range	5
Default Value	5

Sub-Index	1h
Description	COB-ID used by TPDO
Access	RW
PDO Mapping	No
Value Range	See value definition in DS-301
Default Value	40000000h + TPDOx + Node-ID C0000000h + TPDOy + Node-ID

<b>X</b>	<b>TPDOx ID</b>
1	0180h
2	0280h
3	0380h
4	0480h

Node-ID = Node-ID of the module. The TPDO COB-IDs are automatically updated if the Node-ID is changed by LSS protocol.

80000000h in the COB-ID indicates that the PDO does not exist (destroyed)

04000000h in the COB-ID indicates that there is no RTR allowed on the PDO

Sub-Index	2h
Description	Transmission type
Access	RO
PDO Mapping	No
Value Range	See value definition in DS-301

Default Value	254 (FEh) = Event Driven
---------------	--------------------------

Sub-Index	3h
Description	Inhibit Time
Access	RW
PDO Mapping	No
Value Range	See value definition in DS-301
Default Value	0

Sub-Index	4h
Description	Compatibility entry
Access	RW
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	0

Sub-Index	5
Description	Event-timer
Access	RW
PDO Mapping	No
Value Range	See value definition in DS-301
Default Value	50ms on TPDO1; 0ms on TPDO2... TPDO4

### 4.3. MANUFACTURER OBJECTS

<b>Index (hex)</b>	<b>Object</b>	<b>Object Type</b>	<b>Data Type</b>	<b>Access</b>	<b>PDO Mapping</b>
2000	CAN Slew Rate	VAR	UNSIGNED8	RW	No
2002	Sensor Input Filter Enabled	VAR	UNSIGNED8	RW	No
2003	Input Filter Cut-Off Frequency	VAR	UNSIGNED8	RW	No
2010	Accelerometer Data Decimal Digits	ARRAY	UNSIGNED16	RW	No
2011	Accelerometer Data Inversion/Negation	ARRAY	UNSIGNED8	RW	No
2020	Accelerometer Data Offset	ARRAY	FLOAT32	RW	No
2030	Angular Data Decimal Digits	ARRAY	UNSIGNED16	RW	No
2031	Angular Data Inversion/Negation	ARRAY	UNSIGNED8	RW	No
2032	Angle Data Range	ARRAY	UNSIGNED8	RW	No
2040	Angular Data Offset	ARRAY	FLOAT32	RW	No
2100	Pitch and Roll Angle Type	VAR	UNSIGNED8	RW	No
2200	Initial Pitch Angle	VAR	FLOAT32	RW	No
2210	Initial Roll Angle	VAR	FLOAT32	RW	No
2220	Mounting Installation	VAR	UNSIGNED8	RW	No
2300	Auto-Null Command	VAR	UNSIGNED8	RW	No
2400	Calibrated Pitch Angle	VAR	FLOAT32	RO	No
2401	Calibrated Roll Angle	VAR	FLOAT32	RO	No
2500	RPDO Received Message Process Value	ARRAY	INTEGER16	RW	Yes
2501	RPDO Decimal Digits	ARRAY	UNSIGNED16	RW	No
2502	RPDO Offset	ARRAY	FLOAT32	RW	No
3000	Binary Function Output Field Value	ARRAY	INTEGER16	RW	Yes
300A	Binary Function Operation	ARRAY	UNSIGNED8	RW	No
300B	Binary Function Output Scaling	ARRAY	FLOAT32	RW	No
300C	Binary Function Output Offset	ARRAY	FLOAT32	RW	No
30x0	Binary Function x Signal 1&2 Source	ARRAY	UNSIGNED8	RW	No
30x1	Binary Function x Signal 1&2 Is Default	ARRAY	UNSIGNED8	RW	No
30x2	Binary Function x Signal 1&2 Default Value	ARRAY	FLOAT32	RW	No
30x3	Binary Function x Signal 1&2 Unary Function	ARRAY	UNSIGNED8	RW	No
30x4	Binary Function x Signal 1&2 Scale	ARRAY	FLOAT32	RW	No
30x5	Binary Function x Signal 1&2 Offset	ARRAY	FLOAT32	RW	No
4000	Accelerometer Axis Data Field Value	ARRAY	INTEGER16	RO	Yes
4010	Angular Measurement Latency	VAR	UNSIGNED16	RO	Yes
4020	Angular Figure of Merit	VAR	UNSIGNED8	RO	Yes
4030	Pitch Angle Data Field Value	VAR	INTEGER16	RO	Yes
4040	Roll Angle Data Field Value	VAR	INTEGER16	RO	Yes
4050	Gravity Angle Data Field Value	VAR	INTEGER16	RO	Yes
4060	Sensor Temperature Data	VAR	INTEGER16	RO	Yes
4070	Continuous Signal	VAR	FLOAT32	RW	Yes
4080	Discrete Signal	VAR	UNSIGNED32	RW	Yes
5020	Power Supply Field Value	VAR	FLOAT32	RO	Yes
5020	Processor Temperature Field Value	VAR	FLOAT32	RO	Yes
5555	Start in Operational Mode	VAR	BOOLEAN	RW	No

Where x represents Binary Function 1...10

#### 4.3.1. Object 2000h: CAN Slew Rate

This object is available for configuring the slew rate of the CAN transceiver. The controller automatically updates its CAN slew rate depending on the baud rate configured, however, this object is configurable and can be updated at any time.

##### ***Object Description***

Index	2000h
Name	CAN Slew Rate
Object Type	VARIABLE
Data Type	UNSIGNED8

##### ***Entry Description***

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	0(Slow) or 1(Fast) Slow = 4V/ $\mu$ s Fast = 19V/ $\mu$ s
Default Value	0 (Slow)

#### 4.3.2. Object 2002h: Sensor Input Filter Enabled

This object enables or disables the software filtering of the accelerometer data.

##### ***Object Description***

Index	2002h
Name	Sensor Input Filter
Object Type	VARIABLE
Data Type	UNSIGNED8

##### ***Entry Description***

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	0(Disabled) or 1(Enabled)
Default Value	1 (Enabled)

#### 4.3.3. Object 2003h: Sensor Input Filter Enabled

This object is engaged depending on the settings of object 2002h. It determines the cut-off frequency of the data filtering of accelerometer data.

##### ***Object Description***

Index	2003h
Name	Input Filter Cut-Off Frequency
Object Type	VARIABLE
Data Type	UNSIGNED8

##### ***Entry Description***

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	0 to 50 [Hz]
Default Value	1 [Hz]

#### 4.3.4. Object 2010h: Accelerometer Data Decimal Digits

This object allows for the accelerometer data to have greater resolution for each of the three axis.

##### ***Object Description***

Index	2010h
Name	Accelerometer Data Decimal Digits
Object Type	ARRAY
Data Type	UNSIGNED16

##### ***Entry Description***

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

##### ***Entry Description***

Sub-Index	1h...3h
Access	RW
PDO Mapping	No
Value Range	10 to 1000 (Res = 1000/ <i>Digits</i> )
Default Value	1000 (Res = 1 (1000/1000))

#### 4.3.5. Object 2011h: Accelerometer Data Inversion/Negation

This object allows for the accelerometer data to be inverted in sign (negation).

##### ***Object Description***

Index	2011h
Name	Accelerometer Data Inversion
Object Type	ARRAY
Data Type	UNSIGNED8

##### ***Entry Description***

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

##### ***Entry Description***

Sub-Index	1h...3h
Access	RW

PDO Mapping	No
Value Range	0 – 1
Default Value	0 (False, no inversion)

#### 4.3.6. Object 2020h: Accelerometer Data Offset

This object allows to have add and offset to the data being transmitted by the Triaxial Inclinometer in the accelerometer data in the three axis.

##### ***Object Description***

Index	2020h
Name	Accelerometer Data Offset
Object Type	ARRAY
Data Type	FLOAT32

##### ***Entry Description***

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

##### ***Entry Description***

Sub-Index	1h...3h
Access	RW
PDO Mapping	No
Value Range	-0xFFFFFFFF to 0xFFFFFFFF
Default Value	0

#### 4.3.7. Object 2030h: Angular Data Decimal Digits

This object allows for the angular measurement data to have greater resolution for each of the three angular vectors.

##### ***Object Description***

Index	2030h
Name	Angular Data Decimal Digits
Object Type	ARRAY
Data Type	UNSIGNED16

##### ***Entry Description***

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

##### ***Entry Description***

Sub-Index	1h...3h
-----------	---------

Access	RW
PDO Mapping	No
Value Range	10 to 1000 (Res = 1000/ <i>Digits</i> )
Default Value	100(Res = 10 (1000/ <i>100</i> ))

#### 4.3.8. Object 2031h: Angular Data Inversion/Negation

This object allows for the angular measurement data to be inverted in sign (negation).

##### ***Object Description***

Index	2031h
Name	Angular Data Inversion
Object Type	ARRAY
Data Type	UNSIGNED8

##### ***Entry Description***

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

##### ***Entry Description***

Sub-Index	1h...3h
Access	RW
PDO Mapping	No
Value Range	0 – 1
Default Value	0 (False, no inversion)

#### 4.3.9. Object 2032h: Angle Data Range

This object allows the angle measurement of the inclinometer to be configured in order to read the desired values.

##### ***Object Description***

Index	2032h
Name	Angle Data Range
Object Type	ARRAY
Data Type	UNSIGNED8

##### ***Entry Description***

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

##### ***Entry Description***

Sub-Index	1h...3h
Access	RW



PDO Mapping	No
Value Range	0 – 2 {[-180...180], [0...360], [0...+/-90]}
Default Value	0 [-180...180]

#### 4.3.10. Object 2040h: Angular Data Offset

This object allows to have add and offset to the data being transmitted by the Triaxial Inclinometer in the angular data measurements.

##### **Object Description**

Index	2040h
Name	Angular Data Offset
Object Type	ARRAY
Data Type	FLOAT32

##### **Entry Description**

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

##### **Entry Description**

Sub-Index	1h...3h
Access	RW
PDO Mapping	No
Value Range	-0xFFFFFFFF to 0xFFFFFFFF
Default Value	0

#### 4.3.11. Object 2100h: Pitch and Roll Angle Type

This object changes the angle tilt type for the Pitch and Roll angle to an Euler angle or a Tilt Angle.

##### **Object Description**

Index	2100h
Name	Pitch and Roll Angle Type
Object Type	VARIABLE
Data Type	UNSIGNED8

##### **Entry Description**

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	0 (Euler Angle) or 1 (Tilt Angle)
Default Value	1 (Tilt Angle)

#### 4.3.12. Object 2200h: Initial Pitch Angle

This object is useful for installation purposes of the controller when mounted to a machine/system. The *Initial Pitch Angle* and the *Initial Roll Angle* (Object 2210h) are Euler angles. They can be setup manually or automatically when the *Auto-Null Command* (Object 2300h) is set to Yes.

##### **Object Description**

Index	2200h
Name	Initial Pitch Angle
Object Type	VARIABLE
Data Type	FLOAT32

##### **Entry Description**

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	-90 to 90 [Deg]
Default Value	0

#### 4.3.13. Object 2210h: Initial Roll Angle

This object is useful for installation purposes of the controller when mounted to a machine/system. The *Initial Roll Angle* and the *Initial Pitch Angle* (Object 2200h) are Euler angles. They can be setup manually or automatically when the *Auto-Null Command* (Object 2300h) is set to Yes.

##### **Object Description**

Index	2210h
Name	Initial Roll Angle
Object Type	VARIABLE
Data Type	FLOAT32

##### **Entry Description**

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	-180 to 180 [Deg]
Default Value	0

#### 4.3.14. Object 2220h: Mounting Installation

This object allows the inclinometer to be mounted horizontally or vertically to the surface and provide the correct angle measurements. When the unit is configured to be mounted vertically, it may be necessary to update object 2032h to provide the correct angle measurement range.

##### **Object Description**

Index	2220h
Name	Mounting Installation
Object Type	VARIABLE
Data Type	UNSIGNED8

### ***Entry Description***

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	0 to 1
Default Value	0 (Horizontal Mount)

#### **4.3.15. Object 2300h: Auto-Null Command**

When this object is set to *True*, the *Initial Pitch Angle* and *Initial Roll Angle* are updated/set with the angles measured at the time the *Auto-Null Command* was used. When the *Auto-Null Command* is set to *True*, the controller will perform the required action but the data in the object itself will be reset to 0 (*False*).

### ***Object Description***

Index	2300h
Name	Auto-Null Command
Object Type	VARIABLE
Data Type	UNSIGNED8

### ***Entry Description***

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	0 (False) or 1 (True)
Default Value	0 (False)

#### **4.3.16. Object 2400h: Calibrated Pitch Angle Data**

This Read-Only object represents the calibration parameter of the Pitch Angle which can be inspected in the field by a qualified technician.

### ***Object Description***

Index	2400h
Name	Calibrated Pitch Angle Data
Object Type	VARIABLE
Data Type	FLOAT32

### ***Entry Description***

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	-90 to 90 [Deg]
Default Value	0

#### 4.3.17. Object 2401h: Calibrated Roll Angle Data

This Read-Only object represents the calibration parameter of the Roll Angle which can be inspected in the field by a qualified technician.

##### ***Object Description***

Index	2401h
Name	Calibrated Roll Angle Data
Object Type	VARIABLE
Data Type	FLOAT32

##### ***Entry Description***

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	-180 to 180 [Deg]
Default Value	0

#### 4.3.18. Object 2500h: RPDO Received Messaged Process Value

This object is used as RPDO process values that can be used as control sources to other function blocks within the controller's software architecture.

##### ***Object Description***

Index	2500h
Name	RPDO Received Message Process Value
Object Type	ARRAY
Data Type	INTEGER16

##### ***Entry Description***

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

##### ***Entry Description***

Sub-Index	1h...3h
Access	RW
PDO Mapping	Yes
Value Range	-32768 to 32767
Default Value	0

#### 4.3.19. Object 2500h: RPDO Decimal Digits

This object is used to scale the process value of RPDO process values.

##### ***Object Description***

Index	2500h
-------	-------

Name	RPDO Decimal Digits
Object Type	ARRAY
Data Type	UNSIGNED16

**Entry Description**

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

**Entry Description**

Sub-Index	1h...3h
Access	RW
PDO Mapping	No
Value Range	10 to 1000 (Res = 1000/Dec Digits)
Default Value	1000 (Res = 1 (1000/1000))

**4.3.20. Object 2502h: RPDO Offset**

This object is used to offset the process value of RPDO process values before it has been scaled by object 2501h.

**Object Description**

Index	2502h
Name	RPDO Offset
Object Type	ARRAY
Data Type	FLOAT32

**Entry Description**

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

**Entry Description**

Sub-Index	1h...3h
Access	RW
PDO Mapping	No
Value Range	-0xFFFFFFFF to 0xFFFFFFFF
Default Value	0

**4.3.21. Object 3000h: Binary Function Output Field Value**

This object stores the outputs of the Binary Functions (1...10) which can be connected to another functional block or mapped to a TPDO for data transmission.

**Object Description**

Index	3000h
Name	Binary Function Output Field Value
Object Type	ARRAY
Data Type	INTEGER16

**Entry Description**

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	10
Default Value	10

**Entry Description**

Sub-Index	1h...Ah
Access	RO
PDO Mapping	Yes
Value Range	-32768 to 32767
Default Value	0

**4.3.22. Object 300Ah: Binary Function Operation**

This object is used to select the Binary Function operator. Refer to Section 3.6 for more details.

**Object Description**

Index	300Ah
Name	Binary Function Operation
Object Type	ARRAY
Data Type	UNSIGNED8

**Entry Description**

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	10
Default Value	10

**Entry Description**

Sub-Index	1h...Ah
Access	RW
PDO Mapping	No
Value Range	Refer to Table 8
Default Value	0 ( <i>Undefined</i> )

**4.3.23. Object 300Bh: Binary Function Output Scaling**

This object is used to select the Binary Function scaling factor prior to the field value object 3000h. Refer to Section 3.6 for more details.

**Object Description**

Index	300Bh
-------	-------

Name	Binary Function Output Scaling
Object Type	ARRAY
Data Type	FLOAT32

***Entry Description***

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	10
Default Value	10

***Entry Description***

Sub-Index	1h...Ah
Access	RW
PDO Mapping	No
Value Range	-0xFFFFFFFF to 0xFFFFFFFF
Default Value	0

**4.3.24. Object 300Bh: Binary Function Output Scaling**

This object is used to select the Binary Function scaling factor prior to the field value object 3000h. Refer to Section 3.6 for more details.

***Object Description***

Index	300Bh
Name	Binary Function Output Scaling
Object Type	ARRAY
Data Type	FLOAT32

***Entry Description***

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	10
Default Value	10

***Entry Description***

Sub-Index	1h...Ah
Access	RW
PDO Mapping	No
Value Range	-0xFFFFFFFF to 0xFFFFFFFF
Default Value	1

**4.3.25. Object 300Ch: Binary Function Output Offset**

This object is used to add a Binary Function offset prior to the field value object 3000h. Refer to Section 3.6 for more details.

***Object Description***

Index	300Ch
-------	-------

Name	Binary Function Output Offset
Object Type	ARRAY
Data Type	FLOAT32

***Entry Description***

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	10
Default Value	10

***Entry Description***

Sub-Index	1h...Ah
Access	RW
PDO Mapping	No
Value Range	-0xFFFFFFFF to 0xFFFFFFFF
Default Value	0

**4.3.26. Object 30x0h: Binary Function x Signal 1&2 Source**

This object is used to select the control source for each of the two signals in a Binary Function offset prior to the field value object 3000h. Refer to Section 3.6 for more details.

***Object Description***

Index	30x0h, x [1...10]
Name	Binary Function x Signal 1&2 Source
Object Type	ARRAY
Data Type	UNSIGNED8

***Entry Description***

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	2
Default Value	2

***Entry Description***

Sub-Index	1h...2h
Access	RW
PDO Mapping	No
Value Range	Refer to Section 3.1.7
Default Value	0 (Control Source Not Used/Undefined)

**4.3.27. Object 30x1h: Binary Function x Signal 1&2 Is Default**

This object is used to determine if the two input signals have a default value enabled to be used instead of a control source from list in Section 3.1.7

***Object Description***



Index	30x1h, x [1...10]
Name	Binary Function x Signal 1&2 Is Default
Object Type	ARRAY
Data Type	UNSIGNED8

**Entry Description**

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	2
Default Value	2

**Entry Description**

Sub-Index	1h...2h
Access	RW
PDO Mapping	No
Value Range	Refer to Section 3.1.7
Default Value	0 (False)

**4.3.28. Object 30x2h: Binary Function x Signal 1&2 Default Value**

This object is used to enter the default value instead of a control source value from list in Section 3.1.7

**Object Description**

Index	30x2h, x [1...10]
Name	Binary Function x Signal 1&2 Default Value
Object Type	ARRAY
Data Type	FLOAT32

**Entry Description**

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	2
Default Value	2

**Entry Description**

Sub-Index	1h...2h
Access	RW
PDO Mapping	No
Value Range	-0xFFFFFFFF to 0xFFFFFFFF
Default Value	0

**4.3.29. Object 30x3h: Binary Function x Signal 1&2 Unary Function**

For each of the Binary Function input signals, a *Unary Function* can be applied. This object allows for the selection of the *Unary Functions* listed in Section 3.1.7

**Object Description**

Index	30x3h, x [1...10]
Name	Binary Function x Signal 1&2 Unary Function
Object Type	ARRAY
Data Type	UNSIGNED8

**Entry Description**

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	2
Default Value	2

**Entry Description**

Sub-Index	1h...2h
Access	RW
PDO Mapping	No
Value Range	See Table 7
Default Value	0 ( <i>Undefined</i> )

**4.3.30. Object 30x4h: Binary Function x Signal 1&2 Scaling**

This object is used to scale the individual input signals prior to be processed in object 300Ah. Refer to Section 3.1.7 for more details.

**Object Description**

Index	30x4h, x [1...10]
Name	Binary Function x Signal 1&2 Scaling
Object Type	ARRAY
Data Type	FLOAT32

**Entry Description**

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	2
Default Value	2

**Entry Description**

Sub-Index	1h...2h
Access	RW
PDO Mapping	No
Value Range	-0xFFFFFFFF to 0xFFFFFFFF
Default Value	0

**4.3.31. Object 30x5h: Binary Function x Signal 1&2 Offset**

This object is used to add an offset value to the individual input signals prior to be processed in object 300Ah. Refer to Section 3.1.7 for more details.

**Object Description**

Index	30x5h, x [1...10]
Name	Binary Function x Signal 1&2 Offset
Object Type	ARRAY
Data Type	FLOAT32

**Entry Description**

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	2
Default Value	2

**Entry Description**

Sub-Index	1h...2h
Access	RW
PDO Mapping	No
Value Range	-0xFFFFFFFF to 0xFFFFFFFF
Default Value	0

**4.3.32. Object 4000h: Accelerometer Axis Data Field Value**

This object holds the field value of the accelerometer data for the X-, Y- and Z-axis after having applied the decimal digits and offset from object 2000h and object 2020h, respectively, along with the sign applied by object 2011h. This object is mapped to TPDO2 by default.

**Object Description**

Index	4000h
Name	Accelerometer Axis Data FV
Object Type	ARRAY
Data Type	INTEGER16

**Entry Description**

Sub-Index	0h
Access	RO
PDO Mapping	No
Value Range	3
Default Value	3

**Entry Description**

Sub-Index	1h...3h (X-Axis, Y-Axis, Z-Axis)
Access	RO
PDO Mapping	Yes
Value Range	-32768 to 32767
Default Value	0

#### 4.3.33. Object 4010h: Angular Measurement Latency

This object holds the angular measurement latency. This object can be mapped to a TPDO or used as a control source/function block input signal to other functional blocks.

##### ***Object Description***

Index	4010h
Name	Angular Measurement Latency
Object Type	VARIABLE
Data Type	UNSIGNED16

##### ***Entry Description***

Sub-Index	0h
Access	RO
PDO Mapping	Yes
Value Range	0 to 0xFFFFF
Default Value	0

#### 4.3.34. Object 4020h: Angular Figure of Merit

This object holds the angular figure of merit value. This value provides information regarding the status of the sensor in terms of data validity and error conditions. This object can be mapped to a TPDO or used as a control source/function block input signal to other functional blocks. Refer to Section 3.3 for more details.

##### ***Object Description***

Index	4020h
Name	Angular Figure of Merit
Object Type	VARIABLE
Data Type	UNSIGNED8

##### ***Entry Description***

Sub-Index	0h
Access	RO
PDO Mapping	Yes
Value Range	Table 6
Default Value	0

#### 4.3.35. Object 4030h: Pitch Angle Data Field Value

This object holds the *Pitch Angle Data FV* after having applied the decimal digits and offset from object 2030h sub-index 1 and object 2040h sub-index 1, respectively, along with the sign applied by object 2031h sub-index 1. This object can be mapped to a TPDO or used as a control source/function block input signal to other functional blocks. Refer to Section 3.3 for more details.

##### ***Object Description***

Index	4030h
Name	Pitch Angle Data FV
Object Type	VARIABLE
Data Type	INTEGER16

### ***Entry Description***

Sub-Index	0h
Access	RO
PDO Mapping	Yes
Value Range	-90 to 90 [Deg]
Default Value	0

#### **4.3.36. Object 4040h: Roll Angle Data Field Value**

This object holds the *Roll Angle Data* FV after having applied the decimal digits and offset from object 2030h sub-index 2 and object 2040h sub-index 2, respectively, along with the sign applied by object 2031h sub-index 2. This object can be mapped to a TPDO or used as a control source/function block input signal to other functional blocks. Refer to Section 3.3 for more details.

### ***Object Description***

Index	4040h
Name	Roll Angle Data FV
Object Type	VARIABLE
Data Type	INTEGER16

### ***Entry Description***

Sub-Index	0h
Access	RO
PDO Mapping	Yes
Value Range	-180 to 180 [Deg]
Default Value	0

#### **4.3.37. Object 4050h: Gravity Angle Data Field Value**

This object holds the *Gravity Angle Data* FV after having applied the decimal digits and offset from object 2030h sub-index 3 and object 2040h sub-index 3, respectively, along with the sign applied by object 2031h sub-index 3. This object can be mapped to a TPDO or used as a control source/function block input signal to other functional blocks. Refer to Section 3.3 for more details.

### ***Object Description***

Index	4050h
Name	Gravity Angle Data FV
Object Type	VARIABLE
Data Type	INTEGER16

### ***Entry Description***

Sub-Index	0h
Access	RO
PDO Mapping	Yes
Value Range	0 to 180 [Deg]
Default Value	0

#### 4.3.38. Object 4060h: Sensor Temperature Data

This object holds the sensor temperature measured. This differs from the microcontroller temperature measured. It is stored as a 16-bit signed value.

##### **Object Description**

Index	4060h
Name	Sensor Temperature Data
Object Type	VARIABLE
Data Type	INTEGER16

##### **Entry Description**

Sub-Index	0h
Access	RO
PDO Mapping	Yes
Value Range	0 to 180 [Deg]
Default Value	0

#### 4.3.39. Object 4070h: Continuous Signal

This object is part of the *Global Parameter* function block. It is FLOAT32 value data that can be used as input signal/control source to another function block as a *Continuous* signal data type. It can also be mapped as a TPDO message parameter.

##### **Object Description**

Index	4070h
Name	Continuous Signal
Object Type	VARIABLE
Data Type	FLOAT32

##### **Entry Description**

Sub-Index	0h
Access	RW
PDO Mapping	Yes
Value Range	-0xFFFFFFFF to 0xFFFFFFFF
Default Value	0

#### 4.3.40. Object 4080h: Discrete Signal

This object is part of the *Global Parameter* function block. It is used a UNSIGNED32 value data that can be used as input signal/control source to another function block *Discrete* signal data type. It can also be mapped as a TPDO message parameter.

##### **Object Description**

Index	4080h
Name	Discrete Signal
Object Type	VARIABLE
Data Type	UNSIGNED32

##### **Entry Description**

Sub-Index	0h
Access	RW
PDO Mapping	Yes
Value Range	-0xFFFFFFFF to 0xFFFFFFFF
Default Value	0

#### 4.3.41. Object 5020h: Power Supply Field Value

This read-only object is available for diagnostic feedback purposes. It reflects the measured voltage powering the controller. The physical unit for this object is volts.

##### ***Object Description***

Index	5020h
Name	Power Supply Field Value
Object Type	VARIABLE
Data Type	FLOAT32

##### ***Entry Description***

Sub-Index	0h
Access	RO
PDO Mapping	Yes
Value Range	0 to 70 [V]
Default Value	No

#### 4.3.42. Object 5030h: Processor Temperature Field Value

This read-only object is available for diagnostic feedback purposes. It reflects the measured processor temperature of the controller. The physical unit for this object is celsius.

##### ***Object Description***

Index	5030h
Name	Processor Temperature Field Value
Object Type	VARIABLE
Data Type	FLOAT32

##### ***Entry Description***

Sub-Index	0h
Access	RO
PDO Mapping	Yes
Value Range	-50 to 150 [DegC]
Default Value	No

#### 4.3.43. Object 5550h: Enable Automatic Updates

This object allows the controller to update objects to defaults automatically when an output type is changed. Be default this object is TRUE.

##### ***Object Description***

Index	5550h
Name	Enable Auto Updates

Object Type	VARIABLE
Data Type	BOOLEAN

***Entry Description***

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	0 (FALSE) or 1 (TRUE)
Default Value	1 [TRUE]

**4.3.44. Object 5555h: Start in Operational Mode**

This object allows the unit to start in Operational mode without requiring the presence of a CANopen ® Master on the network. It is intended to be used only when running the Triaxial Inclinometer controller as a stand-alone module. This should always be set FALSE whenever it is connected to a standard master/slave network.

***Object Description***

Index	5555h
Name	Start in Operational Mode
Object Type	VARIABLE
Data Type	BOOLEAN

***Entry Description***

Sub-Index	0h
Access	RW
PDO Mapping	No
Value Range	0 (FALSE) or 1 (TRUE)
Default Value	0 [FALSE]



## 5. RS-232 Communication

RS-232 interface is provided to increase backward compatibility with MVINC-CO product family. Measurement data is available in continuous print, which can set up through RS-232 interface. Available RS-232 commands are listed in table below.

Command	Value range	Default value	Available through CAN interface	Description	Notes	Saving
#SNvvv!	1-127	127	LSS protocol	Sets the node ID.	See 4.1.1.1	Saved automatically
#SBvvv!	10, 20, 50, 100, 125, 250, 500, 800, 1000	125 kbit/s	LSS protocol	Sets the baudrate	See 4.1.1.2	Saved automatically
#STv!	f,t	f (False, continuous data print off)	No	Turns the data print on (t) of off (f)		#SSt!
#STivvv!	10 .. 10000	200ms	No	Sets time interval for continuous data print		#SSt!
#STbv!	19200, 38400, 57600, 115200	115200	No	Sets the baudrate for RS-232 interface		#SSt!
#CDv!	f,t	t (True, CAN status prints on)	No	Turns CAN status prints on (t) and off (f)		#SSt!
#CRa!	-	-	1011h, Restore Parameters (restore all)	Restores object table to factory defaults	See 4.2.8	-
#CSa!	-	-	1010h, Store Parameters (save all)	Stores object table to non-volatile memory	See 4.2.7	-
#SRt!	-	-	-	Restores the RS-232 settings to factory defaults	Continuous data print, Data print interval, RS-232 baudrate, CAN status	-

					print	
#SSt!	-	-	-	Stores the RS-232 settings to non-volatile memory	Continuous data print, Data print interval, RS-232 baudrate, CAN status print	-
#RO!	-	-	-	Prints the object table		-
#SERvvv!	v: 10..1000	1000	2030h, Angular Data Decimal Digits (all three angles)	Sets angle field value resolution	See 4.3.7	#CSa!
#SEIav!	v: f, t a: x, y, x	f (False, no inversion)	2031h, Angular Data Inversion/Negation	Sets inversion of the angular data on/off	See 4.3.8	#CSa!
#SEPavv!	v: -90..90(x) -180..180(y) a: x, y	0	2200h, Initial Pitch Angle, 2210h, Initial Roll Angle	Sets initial Angle of given axis (x = pitch, y = roll)	See 4.3.12, 4.3.13	#CSa!
#SZ!	-	-	2300h, Auto-Null Command	The Initial Pitch Angle and Initial Roll Angle are updated/set with the current measured angles.	See 4.3.15	
#SECOa!	a: x, y		2200h, Initial Pitch Angle, 2210h, Initial Roll Angle	Zeros initial Angle of given axis (x = pitch, y = roll)	See 4.3.12, 4.3.13	#CSa!
#SEDOavvv!	v: a: x, y, z	0	2040h, Angular Data Offset		See 4.3.10	#CSa!

## 6. Technical Specifications

### 6.1. Performance Parameters

#### 6.1.1. Static Parameters

Table 12. Static Parameters

Parameter	Value	Remarks
Measurement Range	$\pm 180^\circ$ – Pitch & Roll $0 \dots 180^\circ$ – Gravity	
Resolution	$0.35^\circ$	Maximum Effective Resolution ( $3.46 \cdot \text{NoiseRMS}$ ) at cut-off frequency, $F_c=5\text{Hz}$ .
Initial Accuracy	$\pm 3^\circ$	Maximum, at $25^\circ\text{C}$ .
Temperature Drift	$\pm 3.5^\circ$	Maximum, in the full temperature range: $-40 \dots 85^\circ\text{C}$ .
Nonlinearity	$\pm 0.7\%$	Maximum, at $25^\circ\text{C}$ .
Cross-Axis Sensitivity	$\pm 3.5\%$	Maximum, at $25^\circ\text{C}$ .

#### 6.1.2. Dynamic Parameters

Table 13. Dynamic Parameters

Parameter	Value	Remarks
Cut-off frequency, $F_c$	$1 \dots 50$ Hz, 5 Hz default	User selectable.
Settling time	$\leq 0.2\text{s}$	Typical at default $F_c$ . From 0 to 95% of the static output value.

### 6.2. Power Supply Input

Table 14. Power Supply

Parameter	Value	Remarks
Supply Voltage	$9 \dots 36$ VDC	12V, 24V – nominal.
Supply Current <sup>1</sup>	20 mA 30 mA	Maximum at 24V. Maximum at 12V.
Protection	Reverse polarity, Transients	

<sup>1</sup> CAN bus is connected.

### 6.3. CAN Output

Table 15. CAN Parameters

Parameter	Value	Remarks
Number of ports	1 CAN Port	To output data and change the internal configuration of the inclinometer
Communication standards	CiA CANopen	Full support for a CANopen Node ID.
	ISO 11898	120Ohm terminated twisted pair, baud rate up to 1MBit/s. Termination resistor available in spring load connector models.

Parameter	Value	Remarks
	Bosch CAN protocol specification 2.0, Part A, B.	For the internal CAN controller.
Protection	Short circuit to ground	
	Connection to the power supply	Only for 12V systems, 24V max.

## 6.4. General Specifications

Table 16. General Specifications

Parameter	Value	Remarks
Sensor Type	MEMS	
Internal Logic	User Configurable	EDS File
Operating Temperature	-40...+85 °C	Industrial temperature range.
Environmental Protection	IP67	IEC 60529
Size	See dimensional drawing in Section 6.7.	
Weight	0.75 lb. (0.34 kg)	

### 6.4.1. Compliances

Table 17. Enclosure Protection

Standard	Description	Conditions
IEC 60529	Degrees of protection provided by enclosures (IP Code).	IP67. Mating connectors compliant with IEC 61076-2-101:2012 should be installed.

Table 18. Vibration and Shock

Standard	Description
MIL-STD-202G	<p>The sinusoidal component of the vibration testing was conducted following MIL-STD-202G, method 204D, test condition C (10g peak). A resonant frequency analysis was performed, no resonances were noted.</p> <p>Sweep Characteristic: 10Hz to 2000Hz to 10Hz  Sweep Period: 20 Minutes  Test Duration: 8hrs/axis  Test Intensity: 10g Peak</p> <p>The random component of the vibration testing was conducted meeting or exceeding the requirements of MIL-STD-202G, method 214A, test condition I/B (7.68 Grms):  Frequency Range: 5Hz to 2000Hz  Test Duration: 8hrs/axis  Overall Grms: 7.68 Grms</p> <p>The shock component of the vibration testing is based on MIL-STD-202G, method 213B, test condition A. The pulse duration was shortened to 9 ms from the standard 11 ms due to our vibration system limits. Eight pulses per axis were performed instead of six. The test was conducted as follows:  Pulse Type: half sine</p>

Pulse Duration:	9 ms
Peak Value:	50 g
Pulses per axis:	8

Table 19. Electromagnetic Compatibility (EMC). Emission

Standard	Description	Conditions
EN 61000-6-4	Electromagnetic compatibility (EMC). Generic standards – Emission standard for industrial environments.	Pending

Table 20. Electromagnetic Compatibility (EMC). Immunity

Standard	Description	Conditions
EN 61000-6-2	Electromagnetic compatibility (EMC). Generic standards – Immunity for industrial environments.	Pending

## 6.4.2. Certifications

Table 21. Certifications

Name	Remarks
FCC Part 15 Class A	Radiated emission.
CE Marking	EMC Directive, RoHS Directive.

## 6.5. Connectors

The MVINC module can be connected to CAN bus with MIL connector or spring load connector terminals. There is only one CAN port supported by the unit, so in case of two connectors, both CAN connectors are physically connected to facilitate cable routing in the user system.

In addition to CAN bus connection MVINC module provides a RS-232 interface through the spring load connector block. In case of two MIL6 connectors RS-232 is also available through one MIL6 connector. Connector pinouts are described in following sub chapters.

### 6.5.1. MIL4

ITT Cannon CA3102E14S-14-2P-F80, 4pins, male IP67, cable clamp provided.

Pin	Signal
A	BATT+
B	BATT-
C	CAN hi
D	CAN lo

Table 22. MIL4 Connector pinout

### 6.5.2. MIL6

ITT Cannon KPSE02E10-6P, 6pins. In case of two MIL6 connector, connector J1 contains RS-232 signals. Note that if RS-232 is not used connected cable wires should be pulled low or high to prevent any spurious noise being interpreted as RS-232 commands.

PIN		Signal
J1	J2	
A	A	BATT+
B	B	BATT- / RS-232 GND
C	C	CAN Hi
D	D	CAN Lo
E	-	RS-232 TXD
F	-	RS-232 RXD

Table 23. MIL6 Connector pinout

### 6.5.3. Spring load PCB Connector

The PCB connector consists of ZFKKDS-5.08 spring load terminal blocks, which are manufactured by Phoenix Contacts. Maximum allowed wire size is 1.5mm<sup>2</sup> (AWG 16). It is recommended to use minimum wire size 0.5mm<sup>2</sup> (AWG 20).

The table below illustrates the layout of the PCB mounted spring load connector as seen from above the PCB. On board 120Ω CAN bus termination resistor can be deployed by connecting X1.4 (RES) and X2.4 (CAN Lo) together.

	1	2	3	4	5
X1	BATT+	CAN Hi	CAN Shield	RES	RS-232 TXD
X2	BATT-	CAN Lo	RS-232 GND	CAN Lo	RS-232 RXD

Table 24. Spring load block pinout

### 6.6. Unit Orientation

The unit coordinates, together with the Pitch and Roll directions are shown on the inclinometer label, see: Figure 17.

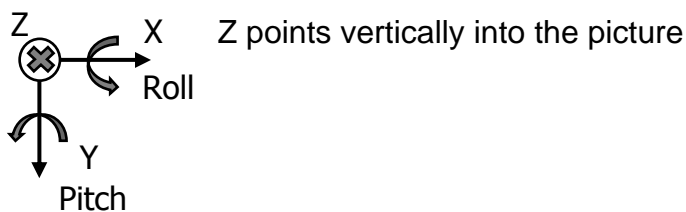
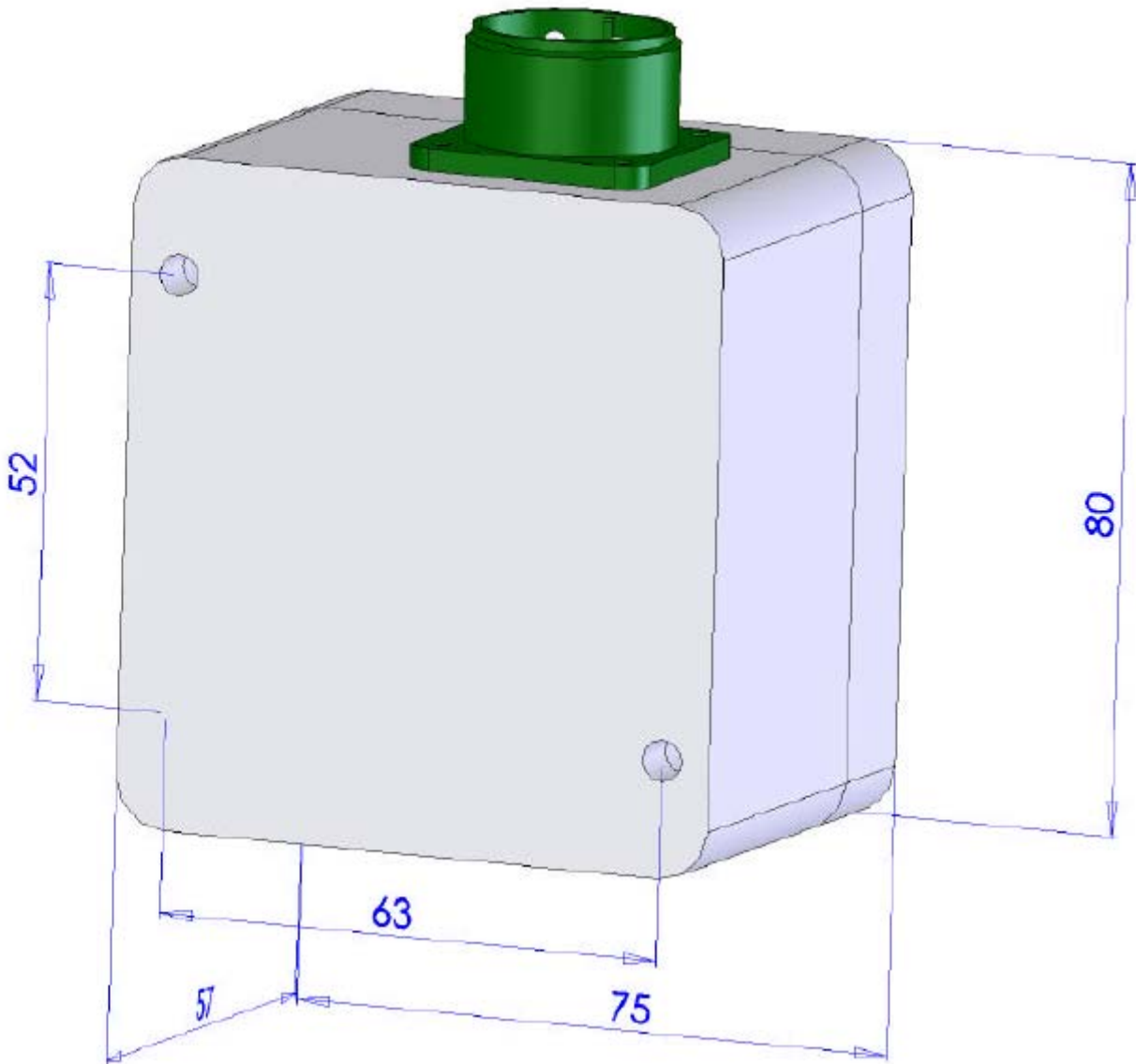


Figure 17. Unit Orientation

## 6.7. Housing

### 6.7.1. MVINC

Aluminium housing dimensions 75 x 79 x 58mm (W x L x H), mounting brackets and connectors excluded. Housing is available with different set of connectors: one or two MIL4 connectors, one or two MIL6 connectors, one or two PG-11 strain reliefs.



## 6.8. Installation

The CAN wiring is considered intrinsically safe. All field wiring should be suitable for the operating temperature range of the module. CAN wiring may be shielded using a shielded twisted conductor pair and the shield must be connected to the CAN\_SHIELD pin.

## 7. VERSION HISTORY

---

<b>Version</b>	<b>Date</b>	<b>Author</b>	<b>Modifications</b>
1.0	November 9 <sup>th</sup> , 2018	Ilona Korpelainen	Initial Draft
3.00	December 12 <sup>th</sup> , 2018	Ilona Korpelainen	Object default values updated to match firmware.
3.00A	December 21 <sup>th</sup> , 2018	Ilona Korpelainen	Changed mfg.pn to sales pn for clarity. Added note about CAN termination resistor.
-	May 16, 2019	Amanda Wilkins	Marketing Review, added model numbers. Discontinued MIL6 option.





## OUR PRODUCTS

Actuator Controls  
Battery Chargers  
CAN bus Controls, Gateways  
CAN/Wifi, CAN/Bluetooth  
CAN/Ethernet  
Current Converters  
DC/DC Power Converters  
DC Voltage/Current Signal Converters  
Engine Temperature Scanners  
Ethernet/CAN Converters  
Fan Drive Controllers  
Hydraulic Valve Controllers  
I/O Controls  
LVDT Simulators  
Machine Controls  
Motor Controls  
PID Controls  
Position Sensors, Angle Measurement Inclinometers, Gyroscopes  
Power Supplies  
PWM Signal Converters/Isolators  
Resolver Signal Conditioners  
Service Tools  
Signal Conditioners  
Strain Gauge CAN Controls  
Surge Suppressors

## OUR COMPANY

Axiomatic provides electronic machine controls, components, and systems to the off-highway, commercial vehicle, electric vehicle, power generator set, material handling, renewable energy and industrial OEM markets.

We provide efficient, innovative solutions that focus on adding value for our customers.

We emphasize service and partnership with our customers, suppliers, and employees to build long term relationships and mutual trust.

## QUALITY DESIGN AND MANUFACTURING

Axiomatic is an ISO 9001:2015 registered facility.

## SERVICE

All products to be returned to Axiomatic require a Return Materials Authorization Number (RMA#).

Please provide the following information when requesting an RMA number:

- Serial number, part number
- Axiomatic invoice number and date
- Hours of operation, description of problem
- Wiring set up diagram, application
- Other comments as needed

When preparing the return shipping paperwork, please note the following. The commercial invoice for customs (and packing slip) should state the harmonized international HS (tariff code), valuation and return goods terminology, as shown in italics below. The value of the units on the commercial invoice should be identical to their purchase price.

*Goods Made In Canada (or Finland)  
Returned Goods for Warranty Evaluation, HS: 9813.00  
Valuation Identical Goods  
Axiomatic RMA#*

## WARRANTY, APPLICATION APPROVALS/LIMITATIONS

Axiomatic Technologies Corporation reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. Users should satisfy themselves that the product is suitable for use in the intended application. All our products carry a limited warranty against defects in material and workmanship. Please refer to our Warranty, Application Approvals/Limitations and Return Materials Process as described on [www.axiomatic.com/service.html](http://www.axiomatic.com/service.html).

## CONTACTS

**Axiomatic Technologies Corporation**  
5915 Wallace Street  
Mississauga, ON  
CANADA L4Z 1Z8  
TEL: +1 905 602 9270  
FAX: +1 905 602 9279  
[www.axiomatic.com](http://www.axiomatic.com)

**Axiomatic Technologies Oy**  
Höytämöntie 6  
33880 Lempäälä  
FINLAND  
TEL: +358 103 375 750  
FAX: +358 3 3595 660  
[www.axiomatic.fi](http://www.axiomatic.fi)