



DESCRIPTION

The GravityGyro™ Dynamic Inclinometer AHRS provides true clean three axis orientation (pitch, roll, yaw) even in the presence of high shock, magnetic interference, acceleration and vibration.

The system employs high stability temperature compensated ceramic packaged MEMS accelerometers and gyroscopes for excellent long-term performance and reliability.

The complexity of multi-axis inertial calibrations, quaternion angles, Kalman filters, and dynamic adaptive sensor fusion is handled by the sensor itself. The GravityGyro™ is the ultimate drop-in upgrade for legacy, inclinometers and magnetic AHRS systems.

KEY FEATURES

- **High accuracy while moving** - across full temp. range
- **Speed aiding** – input enhances accuracy
- **Heading aiding** – input enhances accuracy

APPLICATIONS

- Measuring angle on moving vehicles in harsh environments
- Excavators, mining vehicles, skid-steers, mobile cranes

HARDWARE OPTIONS

- Rugged, IP68/69K (aluminum or stainless steel)
- Internal CAN terminator (software or hardware)
- M12 or Deutsch (single or dual connectors)

OUTPUT OPTIONS

- CAN bus (J1939, CANopen, and custom)
- RS232, RS485, 0 – 10V, 4 – 20 mA

SPECIFICATIONS OVERVIEW

Parameter	Specification
Axes	Three Axis (pitch, roll, yaw)
Range	Elevation: +/-90° Roll: +/-180° Heading: +/-180°
Static tilt accuracy	0.1 °
Dynamic tilt accuracy	0.2 ° RMS error (aided) 0.5 ° RMS error (unaided)
Shock, acceleration and vibration use conditions	<ul style="list-style-type: none"> ▪ 1 gRMS random vibration 5 Hz to 500 Hz ▪ 1 g acceleration 1 second ▪ 20 g ½ sin 10 mSec ▪ 100 g ½ sin 0.1 mSec
Output rate	10 Hz to 250 Hz
Shock survival	1000 g ½ sin 0.1 mSec, 3x any axis
Temperature range	-40 ° to 85 ° C
Voltage	5 – 36 V
Current	22 mA typ. @ 13.6 VDC
Protection	IP68/69K
ESD, Emissions & Immunity	EN 13309 Construction Machinery ISO 14982 Agricultural and Forestry ISO 13766 (200 V/m) Earth Moving

DESIGNED FOR HEAVY VEHICLES

- Technology leader and market leader - SignalQuest produces more dynamic inclinometers than all competitors combined
- Primary tier 1 supplier to more than half of the world's leading heavily vehicles OEMs
- Specifically designed, tested, and qualified to meet the unique environmental operating requirements of commercial, construction, military, agricultural and mining vehicles.



TABLE OF CONTENTS

Description	1
Key Features	1
Applications.....	1
Hardware Options.....	1
Output Options	1
Specifications Overview	1
Designed for Heavy Vehicles	1
CANopen Implementation.....	2
Accessing Objects in the Dictionary.....	15
Device Setup and Commissioning.....	18
CANopen Network Management	23
CANopen Heartbeat.....	25
CANopen SYNC	25
TPDO1: Quaternion Output Message [Factory Default TPDO1 mapping]	26
TPDO2: Accel Message [Factory Default TPDO2 mapping].....	26
TPDO3: Gyro Message [Factory Default TPDO3 mapping]	27
TPDO4: Message [Factory Default TPDO4 mapping]	27
CANopen Further Reading	27
Speed Aiding	28
Heading Aiding.....	28
Yaw/Heading Measurements.....	29
Quaternion Outputs.....	29
Built-In Tests and Failures.....	31
Absolute Maximum Ratings	32
Electrical Characteristics	32
Output Characteristics	32
Performance Specifications	33
Environmental Protection	33
Connector Diagram.....	34
Pin Descriptions.....	34
Device Orientation.....	36
Orientation Configuration Chart	37
Packaging	38
Environment Protection And Testing	39
Limitations and Warnings.....	39
Revision Table SQ-GIX-2150	39

CANOPEN IMPLEMENTATION

This device supports a basic version of the CANopen CiA410 device profile for inclinometers. This datasheet presents a simplified view of CANopen, as necessary to describe the operation of this device.

In CANopen, all settings, controls, communications parameters, and measurements are called objects. Each object is assigned an index. One or more related objects can be grouped under a single index, where each element is assigned a sub-index. The collection of object indices and their meanings are referred to as the Object Dictionary.

This chart explains the implementation details and the Object Dictionary. A CANopen Electronic Data Sheet (EDS) is available that fully describes the device.

Persistent means that the object can be stored in non-volatile memory using object 0x1010 and restored from non-volatile memory using object 0x1011. Persistent objects are also restored from non-volatile memory when the device is power-cycled or reset. If you do not save them to non-volatile memory, the previously saved setting will take effect on the start-up.

Object Index	Sub Index	Feature/Name	Data Type	Access	Persistent (see pg. 3)	Factory Default	Details
		Bit Rate					Bit rate defaults to 250 Kbaud and can be changed via object 0x2100.
		Node ID					Node ID defaults to 0x11 and can be changed via object 0x2101.
		CAN Message Frame					Unit will recognize and send only Standard CAN Frames (11-bit CAN identifiers)
		Logical Devices					Only one logical device is implemented, its parameters have object ID's in the range 0x6000 to 0x67FF
		LSS					Layer Setting Services (LSS) are not supported on this device.
0x1000		Device Type	Unsigned32	Read Only		0x0002019A	CiA410 Inclinometer; Class C2 (2-axes, 16-bit angle measurements); Default PDO mapping; CANopen Safety Not Supported
0x1001		Error Register	Unsigned8	Read Only		0x00	Bit 0 (Generic Error) and Bit 5 (Device-profile specific error) both set in tandem for a device failure. More information on device failures is available in Object 0x4013. <i>NOTE: A Store Parameters "save" command (0x1010) may cause errors to be reported in registers 0x1001 and 0x4013. See page 31.</i>
0x1005		COB-ID SYNC	Unsigned32	Read/Write	Yes	0x00000080	COB-ID of SYNC messages.
0x1010	0x00	Store Parameters	Unsigned8	Read Only		0x01	Highest sub-index is 1.
	0x01	All parameters	Unsigned32	Read/Write		0x00000001	When read, always returns 0x00000001: All parameters marked 'persistent' are written to non-volatile memory on command. The command is executed by writing "save" (4-byte value, with the 's' in the Least-Significant byte position) to this object. <i>NOTE: The sensor's output will be disrupted for a few seconds during and after a "save" command.</i> <i>NOTE: A Store Parameters "save" command may cause errors to be reported in registers 0x1001 and 0x4013. See page 31.</i>

Object Index	Sub Index	Feature/Name	Data Type	Access	Persistent (see pg. 3)	Factory Default	Details
0x1011	0x00	Restore Default Parameters	Unsigned8	Read Only		0x01	Highest sub-index is 1.
	0x01	All parameters	Unsigned32	Read/Write		0x00000001	When read, always returns 0x00000001: All Factory default parameters are restored from non-volatile memory on command (with the exception of Baud Rate [0x2100], Node ID [0x2101], Enable Terminator [0x2102] and Orientation [0x3000]). The command is executed by writing "load" (4-byte value, with the 'l' in the Least-Significant byte position) to this object. The "default parameters" are the ones previously saved to non-volatile memory using command 0x1010.
0x1014		COB-ID EMCY message	Unsigned32	Read/Write	Yes	0x00000080 + Node ID	EMCY messages are generated using this COB-ID when errors or exceptional events occur within the device.
0x1017		Heartbeat Time	Unsigned16	Read/Write	Yes	0	Heartbeat time, in mSec. To activate the heartbeat, write a non-zero value to this location. To disable heartbeat, write a zero. This setting takes effect immediately, but if you want it to be remembered, it must be stored to non-volatile memory using object 0x1010.
0x1018	0x00	Identity	Unsigned8	Read Only		0x04	Highest sub-index is 4.
	0x01	Vendor ID	Unsigned32	Read Only		0x00005351	SQ – SignalQuest
	0x02	Product Code	Unsigned32	Read Only		2150	SQ-GIX-2150
	0x03	Revision	Unsigned32	Read Only			SQ revision numbers consist of a date code and a letter, e.g. 20160627_A. The low 5 bits of the Revision represent the letter (0 for A, 1 for B, 2 for C, etc.); and the rest of the word represents the numeric part of the serial number. 20160627_A: $20160627 * 32 + 0 = 645140064d = 0x26740E60$.
	0x04	Serial Number	Unsigned32	Read Only			Serial number of the device.
0x1019		Synchronous counter overflow value	Unsigned8	Read/Write	Yes	0x00	Indicates whether the SYNC message will contain a 1-byte counter and indicates the range of the counter. 0x00 = SYNC message has no data field. 0x02 to 0xF0 = SYNC message has a 1-byte counter field.
0x1029	0x00	Error Behavior	Unsigned8	Read Only		0x01	Highest sub-index is 1.
	0x01	Comm. Error	Unsigned8	Read/Write	Yes	0x00	Determines what effect a communication error will have on the NMT state. 0x00 = Changes state to Pre-Operational. 0x01 = No change in NMT state. 0x02 = Change NMT state to Stopped.
0x1200	0x00	Server SDO	Unsigned8	Read Only		0x02	Highest sub-index is 2.
	0x01	CAN-ID To Device	Unsigned32	Read Only		0x00000600 + Node ID	CAN-ID for SDO messages TO the device.
	0x02	CAN-ID From Device	Unsigned32	Read Only		0x00000580 + Node ID	CAN-ID for SDO messages FROM the device.

Object Index	Sub Index	Feature/Name	Data Type	Access	Persistent (see pg. 3)	Factory Default	Details
0x1400	0x00	RPDO1	Unsigned8	Read Only		0x05	Highest sub-index is 5
	0x01	COB-ID	Unsigned32	Read/Write	Yes	0x200 + Node ID	RPDO COB-ID Bit 31 set disables the RPDO
	0x02	Type	Unsigned8	Read/Write	Yes	0xFF	0x00 to 0xF0 means the RPDO is actuated upon receipt of the next SYNC. 0xFF means that the RPDO is transmitted asynchronously. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in Sub Index 0x01.
	0x03	Inhibit Time	Unsigned16	Read Only		0x0000	Not used in this device.
	0x04	Compatibility Entry	Unsigned8	Read Only		0x00	Not used.
	0x05	Event Timer	Unsigned16	Read/Write	Yes	1500	Sets maximum time in mS between transmissions of the RPDO. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in Sub Index 0x01
0x1401	0x00	RPDO2	Unsigned8	Read Only		0x05	Highest sub-index is 5
	0x01	COB-ID	Unsigned32	Read/Write	Yes	0x80000300 + Node ID	RPDO COB-ID Bit 31 set disables the RPDO
	0x02	Type	Unsigned8	Read/Write	Yes	0xFF	0x00 to 0xF0 means the RPDO is actuated upon receipt of the next SYNC. 0xFF means that the RPDO is transmitted asynchronously. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in Sub Index 0x01.
	0x03	Inhibit Time	Unsigned16	Read Only		0x0000	Not used in this device.
	0x04	Compatibility Entry	Unsigned8	Read Only		0x00	Not used.
	0x05	Event Timer	Unsigned16	Read/Write	Yes	500	Sets maximum time in mS between transmissions of the RPDO. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in Sub Index 0x01
0x1402	0x00	RPDO3	Unsigned8	Read Only		0x05	Highest sub-index is 5
	0x01	COB-ID	Unsigned32	Read/Write	Yes	0x80000400 + Node ID	RPDO COB-ID Bit 31 set disables the RPDO
	0x02	Type	Unsigned8	Read/Write	Yes	0xFF	0x00 to 0xF0 means the RPDO is actuated upon receipt of the next SYNC. 0xFF means that the RPDO is transmitted asynchronously. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in Sub Index 0x01.
	0x03	Inhibit Time	Unsigned16	Read Only		0x0000	Not used in this device.
	0x04	Compatibility Entry	Unsigned8	Read Only		0x00	Not used.
	0x05	Event Timer	Unsigned16	Read/Write	Yes	500	Sets maximum time in mS between transmissions of the RPDO. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in Sub Index 0x01

Object Index	Sub Index	Feature/Name	Data Type	Access	Persistent (see pg. 3)	Factory Default	Details
0x1403	0x00	RPDO4	Unsigned8	Read Only		0x05	Highest sub-index is 5
	0x01	COB-ID	Unsigned32	Read/Write	Yes	0x80000500 + Node ID	RPDO COB-ID Bit 31 set disables the RPDO
	0x02	Type	Unsigned8	Read/Write	Yes	0xFF	0x00 to 0xF0 means the RPDO is actuated upon receipt of the next SYNC. 0xFF means that the RPDO is transmitted asynchronously. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in Sub Index 0x01.
	0x03	Inhibit Time	Unsigned16	Read Only		0x0000	Not used in this device.
	0x04	Compatibility Entry	Unsigned8	Read Only		0x00	Not used.
	0x05	Event Timer	Unsigned16	Read/Write	Yes	500	Sets maximum time in mS between transmissions of the RPDO. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in Sub Index 0x01
0x1600	0x00	RPDO1 Mapping	Unsigned8	Read/Write	Yes	0x03	Number of parameters (0x00 to 0x08). Writing this to 0x00 disables this RPDO. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in 0x1400 Sub Index 0x01.
	0x01	RPDO Parameter 1	Unsigned32	Read/Write	Yes	0x31100110	Heading (Object 0x3110 sub 01, 16 bits in length) Heading Std. Dev. (Object 0x3110 sub 03, 16 bits) Heading T.O.V. (Object 0x3110 sub 02, 16 bits) Unused 32-bit integer All other parameters unused. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in 0x1400 Sub Index 0x01, and writing 0x00 to Sub Index 0x00 of this Object. NOTE: This device only supports byte level mapping. All bit lengths specified in the mappings must be a multiple of 8 bits.
	0x02	RPDO Parameter 2				0x31100310	
	0x03	RPDO Parameter 3				0x31100210	
	0x04	RPDO Parameter 4				0x00000000	
	0x05	RPDO Parameter 5				0x00000000	
	0x06	RPDO Parameter 6				0x00000000	
	0x07	RPDO Parameter 7				0x00000000	
	0x08	RPDO Parameter 8				0x00000000	
0x1601	0x00	RPDO2 Mapping				Unsigned8	
	0x01	RPDO Parameter 1	Unsigned32	Read/Write	Yes	0x00000000	All parameters unused. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in 0x1401 Sub Index 0x01, and writing 0x00 to Sub Index 0x00 of this Object. NOTE: This device only supports byte level mapping. All bit lengths specified in the mappings must be a multiple of 8 bits.
	0x02	RPDO Parameter 2				0x00000000	
	0x03	RPDO Parameter 3				0x00000000	
	0x04	RPDO Parameter 4				0x00000000	
	0x05	RPDO Parameter 5				0x00000000	
	0x06	RPDO Parameter 6				0x00000000	
	0x07	RPDO Parameter 7				0x00000000	
	0x08	RPDO Parameter 8				0x00000000	
0x1602	0x00	RPDO3 Mapping				Unsigned8	
	0x01	RPDO Parameter 1	Unsigned32	Read/Write	Yes	0x00000000	All parameters unused. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in 0x1402 Sub Index 0x01, and writing 0x00 to Sub Index 0x00 of this Object. NOTE: This device only supports byte level mapping. All bit lengths specified in the mappings must be a multiple of 8 bits.
	0x02	RPDO Parameter 2				0x00000000	
	0x03	RPDO Parameter 3				0x00000000	
	0x04	RPDO Parameter 4				0x00000000	
	0x05	RPDO Parameter 5				0x00000000	
	0x06	RPDO Parameter 6				0x00000000	
	0x07	RPDO Parameter 7				0x00000000	
	0x08	RPDO Parameter 8				0x00000000	

Object Index	Sub Index	Feature/Name	Data Type	Access	Persistent (see pg. 3)	Factory Default	Details
0x1603	0x00	RPDO4 Mapping	Unsigned8	Read/Write	Yes	0x00	Number of parameters (0x00 to 0x08). Writing this to 0x00 disables this RPDO. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in 0x1403 Sub Index 0x01.
	0x01	RPDO Parameter 1	Unsigned32	Read/Write	Yes	0x00000000	All parameters unused. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in 0x1403 Sub Index 0x01, and writing 0x00 to Sub Index 0x00 of this Object. NOTE: This device only supports byte level mapping. All bit lengths specified in the mappings must be a multiple of 8 bits.
	0x02	RPDO Parameter 2					
	0x03	RPDO Parameter 3					
	0x04	RPDO Parameter 4					
	0x05	RPDO Parameter 5					
	0x06	RPDO Parameter 6					
	0x07	RPDO Parameter 7					
	0x08	RPDO Parameter 8					
0x1800	0x00	TPDO1	Unsigned8	Read Only		0x06	Highest sub-index is 6.
	0x01	COB-ID	Unsigned32	Read/Write	Yes	0x4000180 + Node ID	TPDO COB-ID. Bit 30 set means No RTR allowed on this PDO. Bit 31 set disables this PDO.
	0x02	Transmission type	Unsigned8	Read/Write	Yes	0xFF	0x01 to 0xF0 means the TPDO is transmitted after every SYNC (0x01), every other SYNC (0x02), etc. up to every 240 th SYNC (0xF0). 0xFF means that the TPDO is transmitted asynchronously at a rate determined by the Event Timer. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in Sub Index 0x01.
	0x03	Inhibit Time	Unsigned16	Read/Write	Yes	0x0000	Not used on this device. NOTE: Writing to this Sub Index requires that bit 0x80000000 is set in Sub Index 0x01.
	0x04	Compatibility Entry	Unsigned8	Read/Write		0x00	Not used. NOTE: Writing to this Sub Index requires that bit 0x80000000 is set in Sub Index 0x01.
	0x05	Event Timer	Unsigned16	Read/Write	Yes	100	Effective only if Transmission type is 0xFF. If 0, stops transmission of the TPDO. Otherwise, it is the output data rate for sending the TPDO, in mS. NOTE: Writing to this Sub Index requires that bit 0x80000000 is set in Sub Index 0x01.
	0x06	SYNC start value	Unsigned8	Read/Write	Yes	0x00	Indicates the value of the SYNC message counter on which the FIRST TPDO will be transmitted. Ignored if 0x00 or if object 0x1019 is 0x00. NOTE: Writing to this Sub Index requires that bit 0x80000000 is set in Sub Index 0x01.

Object Index	Sub Index	Feature/Name	Data Type	Access	Persistent (see pg. 3)	Factory Default	Details
0x1801	0x00	TPDO2	Unsigned8	Read Only		0x06	Highest sub-index is 6.
	0x01	COB-ID	Unsigned32	Read/Write	Yes	0x40000280 + Node ID	TPDO COB-ID. Bit 30 set means No RTR allowed on this PDO. Bit 31 set disables this PDO.
	0x02	Transmission type	Unsigned8	Read/Write	Yes	0xFF	0x01 to 0xF0 means the TPDO is transmitted after every SYNC (0x01), every other SYNC (0x02), etc. up to every 240 th SYNC (0xF0). 0xFF means that the TPDO is transmitted asynchronously at a rate determined by the Event Timer. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in Sub Index 0x01.
	0x03	Inhibit Time	Unsigned16	Read/Write	Yes	0x0000	Not used on this device. NOTE: Writing to this Sub Index requires that bit 0x80000000 is set in Sub Index 0x01.
	0x04	Compatibility Entry	Unsigned8	Read/Write		0x00	Not used. NOTE: Writing to this Sub Index requires that bit 0x80000000 is set in Sub Index 0x01.
	0x05	Event Timer	Unsigned16	Read/Write	Yes	100	Effective only if Transmission type is 0xFF. If 0, stops transmission of the TPDO. Otherwise, it is the output data rate for sending the TPDO, in mS. NOTE: Writing to this Sub Index requires that bit 0x80000000 is set in Sub Index 0x01.
	0x06	SYNC start value	Unsigned8	Read/Write	Yes	0x00	Indicates the value of the SYNC message counter on which the FIRST TPDO will be transmitted. Ignored if 0x00 or if object 0x1019 is 0x00. NOTE: Writing to this Sub Index requires that bit 0x80000000 is set in Sub Index 0x01.
0x1802	0x00	TPDO3	Unsigned8	Read Only		0x06	Highest sub-index is 6.
	0x01	COB-ID	Unsigned32	Read/Write	Yes	0x40000380 + Node ID	TPDO COB-ID. Bit 30 set means No RTR allowed on this PDO. Bit 31 set disables this PDO.
	0x02	Transmission type	Unsigned8	Read/Write	Yes	0xFF	0x01 to 0xF0 means the TPDO is transmitted after every SYNC (0x01), every other SYNC (0x02), etc. up to every 240 th SYNC (0xF0). 0xFF means that the TPDO is transmitted asynchronously at a rate determined by the Event Timer. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in Sub Index 0x01.
	0x03	Inhibit Time	Unsigned16	Read/Write	Yes	0x0000	Not used on this device. NOTE: Writing to this Sub Index requires that bit 0x80000000 is set in Sub Index 0x01.
	0x04	Compatibility Entry	Unsigned8	Read/Write		0x00	Not used. NOTE: Writing to this Sub Index requires that bit 0x80000000 is set in Sub Index 0x01.
	0x05	Event Timer	Unsigned16	Read/Write	Yes	100	Effective only if Transmission type is 0xFF. If 0, stops transmission of the TPDO. Otherwise, it is the output data rate for sending the TPDO, in mS. NOTE: Writing to this Sub Index requires that bit 0x80000000 is set in Sub Index 0x01.
	0x06	SYNC start value	Unsigned8	Read/Write	Yes	0x00	Indicates the value of the SYNC message counter on which the FIRST TPDO will be transmitted. Ignored if 0x00 or if object 0x1019 is 0x00. NOTE: Writing to this Sub Index requires that bit 0x80000000 is set in Sub Index 0x01.

Object Index	Sub Index	Feature/Name	Data Type	Access	Persistent (see pg. 3)	Factory Default	Details
0x1803	0x00	TPDO4	Unsigned8	Read Only		0x06	Highest sub-index is 6.
	0x01	COB-ID	Unsigned32	Read/Write	Yes	0xC0000480 + Node ID	TPDO COB-ID. Bit 30 set means No RTR allowed on this PDO. Bit 31 set disables this PDO.
	0x02	Transmission type	Unsigned8	Read/Write	Yes	0xFF	0x01 to 0xF0 means the TPDO is transmitted after every SYNC (0x01), every other SYNC (0x02), etc. up to every 240 th SYNC (0xF0). 0xFF means that the TPDO is transmitted asynchronously at a rate determined by the Event Timer. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in Sub Index 0x01.
	0x03	Inhibit Time	Unsigned16	Read/Write	Yes	0x0000	Not used on this device. NOTE: Writing to this Sub Index requires that bit 0x80000000 is set in Sub Index 0x01.
	0x04	Compatibility Entry	Unsigned8	Read/Write		0x00	Not used. NOTE: Writing to this Sub Index requires that bit 0x80000000 is set in Sub Index 0x01.
	0x05	Event Timer	Unsigned16	Read/Write	Yes	0	Effective only if Transmission type is 0xFF. If 0, stops transmission of the TPDO. Otherwise, it is the output data rate for sending the TPDO, in mS. NOTE: Writing to this Sub Index requires that bit 0x80000000 is set in Sub Index 0x01.
	0x06	SYNC start value	Unsigned8	Read/Write	Yes	0x00	Indicates the value of the SYNC message counter on which the FIRST TPDO will be transmitted. Ignored if 0x00 or if object 0x1019 is 0x00. NOTE: Writing to this Sub Index requires that bit 0x80000000 is set in Sub Index 0x01.
0x1A00	0x00	TPDO1 Mapping	Unsigned8	Read/Write	Yes	0x04	Number of parameters (0x00 to 0x08). Writing this to 0x00 disables this TPDO. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in 0x1800 Sub Index 0x01.
	0x01	TPDO Parameter 1 TPDO Parameter 2 TPDO Parameter 3 TPDO Parameter 4 TPDO Parameter 5 TPDO Parameter 6 TPDO Parameter 7 TPDO Parameter 8	Unsigned32	Read/Write	Yes	0x60100010	Elevation (Object 0x6010, 16 bits in length) Roll (Object 0x6020, 16 bits in length) Yaw (Object 0x4005, 16 bits in length) *Temperature (Object 0x6511, 8 bits in length) All other parameters unused. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in 0x1800 Sub Index 0x01, and writing 0x00 to Sub Index 0x00 of this Object. NOTE: This device only supports byte level mapping. All bit lengths specified in the mappings must be a multiple of 8 bits.
	0x60200010						
	0x40050010						
	0x65110008						
	0x00000000						
	0x00000000						
	0x00000000						
	0x00000000						

Object Index	Sub Index	Feature/Name	Data Type	Access	Persistent (see pg. 3)	Factory Default	Details
0x1A01	0x00	TPDO2 Mapping	Unsigned8	Read/Write	Yes	0x03	Number of parameters (0x00 to 0x08). Writing this to 0x00 disables this TPDO. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in 0x1801 Sub Index 0x01.
	0x01	TPDO Parameter 1	Unsigned32	Read/Write	Yes	0x40000110	Acceleration X (Object 0x4000 sub 0x01), 16 bits Acceleration Y (Object 0x4000 sub 0x02), 16 bits Acceleration Z (Object 0x4000 sub 0x03), 16 bits All other parameters unused. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in 0x1801 Sub Index 0x01, and writing 0x00 to Sub Index 0x00 of this Object. NOTE: This device only supports byte level mapping. All bit lengths specified in the mappings must be a multiple of 8 bits.
	0x02	TPDO Parameter 2				0x40000210	
	0x03	TPDO Parameter 3				0x40000310	
	0x04	TPDO Parameter 4				0x00000000	
	0x05	TPDO Parameter 5				0x00000000	
	0x06	TPDO Parameter 6				0x00000000	
	0x07	TPDO Parameter 7				0x00000000	
	0x08	TPDO Parameter 8				0x00000000	
0x1A02	0x00	TPDO3 Mapping	Unsigned8	Read/Write	Yes	0x03	Number of parameters (0x00 to 0x08). Writing this to 0x00 disables this TPDO. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in 0x1802 Sub Index 0x01.
	0x01	TPDO Parameter 1	Unsigned32	Read/Write	Yes	0x40010110	Gyro X (Object 0x4001 sub 0x01), 16 bits Gyro Y (Object 0x4001 sub 0x02), 16 bits Gyro Z (Object 0x4001 sub 0x03), 16 bits All other parameters unused. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in 0x1802 Sub Index 0x01, and writing 0x00 to Sub Index 0x00 of this Object. NOTE: This device only supports byte level mapping. All bit lengths specified in the mappings must be a multiple of 8 bits.
	0x02	TPDO Parameter 2				0x40010210	
	0x03	TPDO Parameter 3				0x40010310	
	0x04	TPDO Parameter 4				0x00000000	
	0x05	TPDO Parameter 5				0x00000000	
	0x06	TPDO Parameter 6				0x00000000	
	0x07	TPDO Parameter 7				0x00000000	
	0x08	TPDO Parameter 8				0x00000000	
0x1A03	0x00	TPDO4 Mapping	Unsigned8	Read/Write	Yes	0x00	Number of parameters (0x00 to 0x08). Writing this to 0x00 disables this TPDO. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in 0x1803 Sub Index 0x01.
	0x01	TPDO Parameter 1	Unsigned32	Read/Write	Yes	0x00000000	All parameters unused, TPDO is disabled. NOTE: Writing to this Sub Index requires that bit 31 (0x80000000) is set in 0x1803 Sub Index 0x01, and writing 0x00 to Sub Index 0x00 of this Object. NOTE: This device only supports byte level mapping. All bit lengths specified in the mappings must be a multiple of 8 bits.
	0x02	TPDO Parameter 2				0x00000000	
	0x03	TPDO Parameter 3				0x00000000	
	0x04	TPDO Parameter 4				0x00000000	
	0x05	TPDO Parameter 5				0x00000000	
	0x06	TPDO Parameter 6				0x00000000	
	0x07	TPDO Parameter 7				0x00000000	
	0x08	TPDO Parameter 8				0x00000000	
0x1F80		NMT Startup	Unsigned32	Read/Write	Yes	0x00000008	Device is an NMT slave. If this value is set to 0x00000000, then the device will await an NMT command to enter operational state. If this value is set to 0x00000008, then the device will enter operational state autonomously. Has no effect until next time the sensor is started.
0x2100		Baud Rate	Unsigned8	Read/Write	Yes Not restored by Object 0x1011.	0x03	Baud Rate, encoded as one of the following: 0x00 for 1,00K; 0x02 for 500K; 0x03 for 250K; 0x04 for 125K. This setting does not take effect immediately, and must be stored to non-volatile memory using object 0x1010. The new rate takes effect after next power cycle.
0x2101		Node ID	Unsigned8	Read/Write	Yes Not restored by Object 0x1011.	0x11	Any Node ID from 0x01 to 0x7F is acceptable. . This setting does not take effect immediately, and must be stored to non-volatile memory using object 0x1010. The new Node ID takes effect after next power cycle.

Object Index	Sub Index	Feature/Name	Data Type	Access	Persistent (see pg. 3)	Factory Default	Details
0x2102		Enable Terminator	Unsigned8	Read/Write	Yes Not restored by Object 0x1011.	0x00	Set to 0x00 to disable internal CAN terminator, or 0x01 to enable it. This setting does not take effect immediately, and must be stored to non-volatile memory using object 0x1010. The new terminator setting takes effect after next power cycle.
0x2110		Reset Device	Unsigned32	Read/Write		0x00000001	When read, always returns 0x00000001. Performs a hard reset of the device, which is equivalent to cycling power, allowing some settings, such as Orientation to be changed without needing to physically cycle power on the device. This command exists in sensors with firmware version 20171117 or older. To issue the command write "rese" (4-byte value, with the 'r' in the Least-Significant byte position) to this object.
0x3000		Orientation	Unsigned8	Read/Write	Yes Not restored by Object 0x1011.	20	Selects one of 24 standard orientations that determine which axes and what directions are used for pitch and roll measurements. The orientations are numbered 0x00 through 0x17 (0 through 23 decimal). This setting does not take effect immediately, and must be stored to non-volatile memory using object 0x1010. The new orientation setting takes effect after next power cycle.
0x3010		Speed Resolution	Unsigned32	Read/Write	Yes	100	Resolution of Speed Inputs in 1×10^{-9} meter/sec increments.
0x3101	0x00	Speed Input 32	Unsigned8	Read Only		0x04	Highest sub-index is 4.
	0x01	Speed Input 32 1	Signed32	Read/Write		0	When written, which is normally done through an RPDO, sets the vehicle velocity for Speed Aiding, which will improve angle accuracy. Not all speed inputs need to be used. Those that are actively updated will be averaged to determine vehicle velocity. Resolution of this input is set by object 0x3010. A value of 0x80000000 (-2,147,483,648) represents an invalid speed measurement.
	0x02	Speed Input 32 2				0	
	0x03	Speed Input 32 3				0	
	0x04	Speed Input 32 4				0	
0x3102	0x00	Speed Input 16	Unsigned8	Read Only		0x04	Highest sub-index is 4.
	0x01	Speed Input 16 1	Signed16	Read/Write		0	When written, which is normally done through an RPDO, sets the vehicle velocity for Speed Aiding, which will improve angle accuracy. Not all speed inputs need to be used. Those that are actively updated will be averaged to determine vehicle velocity. Resolution of this input is set by object 0x3010. A value of 0x8000 (-32,768) represents an invalid speed measurement.
	0x02	Speed Input 16 2				0	
	0x03	Speed Input 16 3				0	
	0x04	Speed Input 16 4				0	

Object Index	Sub Index	Feature/Name	Data Type	Access	Persistent (see pg. 3)	Factory Default	Details
0x3110	0x00	Heading	Unsigned8	Read Only		0x03	Highest sub-index is 3.
	0x01	Heading Input	Unsigned16	Read/Write	No	0	When written, which is normally done through an RPDO, sets the vehicle heading for Heading Aiding, which will improve dynamic angle accuracy. Heading is in angle in 0.01° increments, with 0° representing North. A value of 0xFFFF (65535) represents an invalid heading.
	0x02	Heading Time-of-Validity	Unsigned16	Read Only	No	0	When written, which is normally done through an RPDO, sets the time of validity for the current Heading. The value should be a time in milliseconds, representing the time BEFORE the RPDO message was queued for transmission that the heading measurement was made.
	0x03	Heading Standard Deviation	Unsigned16	Read/Write	No	0	When written, which is normally done through an RPDO, sets the Heading Standard Deviation, in units of 0.01°. Values less than 0.05° will be treated as 0.05°. A value of 0xFFFF (65535) represents an invalid heading deviation.
0x3210		Motion Hold	Unsigned8	Read/Write	No	0	Turn Motion Hold ON (1) or OFF (0). If used, motion hold must be turned on when it is certain that the sensor is motionless. It must be turned off BEFORE the sensor starts moving. Once this feature is used after power-on, the sensor depends upon continued Motion Hold commands to perform internal dynamic calibration.
0x3211		Yaw Reset	Unsigned8	Read/Write	No	0	When a 1 is written to this object, a Yaw Reset command is issued, and the Yaw measurement will return to 0 degrees. The value of this will revert to 0 when that operation is complete.
0x3212		Gyro Bias Trim	Unsigned8	Read/Write	No	0 or 1	When read, this object indicates whether the gyro bias has been trimmed since the last power-on or reset. If this value is a 0, then the gyro bias has not been trimmed since power-on. In that case, you should make sure the sensor is not moving or vibrating and write a 1 to this object, which will issue a Gyro Bias Trim command.

Object Index	Sub Index	Feature/Name	Data Type	Access	Persistent (see pg. 3)	Factory Default	Details
0x3402	0x00	ADC Firmware Calibration	Unsigned8	Const		0x03	Highest sub-index is 3
	0x01	Calibrated Firmware	Unsigned32	Read/Write	Yes	2150	Product code for the firmware that was in the sensor when it was calibrated. Will be 2150 for SQ-GIX-2150 firmware. DO NOT WRITE TO THIS OBJECT UNLESS INSTRUCTED TO DO SO BY THE FACTORY.
	0x02	Calibrated Version	Unsigned32	Read/Write	Yes		Firmware revision code for the firmware version that was in the sensor when it was calibrated. Encoding is the same as object 0x1018 sub 0x03. DO NOT WRITE TO THIS OBJECT UNLESS INSTRUCTED TO DO SO BY THE FACTORY.
	0x03	Calibration Status	Unsigned8	Read Only			The current status of the ADC calibration. The following values are used: 0 – Not Determined yet 1 – Original calibrated firmware not known 2 – Original calibrated firmware not recognized 3 – Original calibrated firmware good 4 – Factory test mode (error)
0x4000	0x00	Acceleration	Unsigned8	Const		0x03	Highest sub-index is 3.
	0x01	Acceleration X	Signed16	Read Only			Provides X-Axis Acceleration measurement as a 16-bit signed integer in mg.
	0x02	Acceleration Y	Signed16	Read Only			Provides Y-Axis Acceleration measurement as a 16-bit signed integer in mg.
	0x03	Acceleration Z	Signed16	Read Only			Provides Z-Axis Acceleration measurement as a 16-bit signed integer in mg.
0x4001	0x00	Gyro	Unsigned8	Const		0x03	Highest sub-index is 3.
	0x01	Gyro Roll (X)	Signed16	Read Only			Provides Roll Gyro measurement as a 16-bit signed integer in 0.1°/sec steps.
	0x02	Gyro Pitch (Y)	Signed16	Read Only			Provides Pitch Gyro measurement as a 16-bit signed integer in 0.1°/sec steps.
	0x03	Gyro Yaw (Z)	Signed16	Read Only			Provides Yaw Gyro measurement as a 16-bit signed integer in 0.1°/sec steps.
0x4002		Elevation Rate	Signed16	Read Only			Provides rate of elevation as a 16-bit signed integer in 0.1°/sec steps. This measurement is similar to Gyro Pitch, but with bias correction.
0x4003		Roll Rate	Signed16	Read Only			Provides rate of roll as a 16-bit signed integer in 0.1°/sec steps. This measurement is similar to Gyro Roll, but with bias correction.
0x4005		Yaw/Heading (16-bit)	Unsigned16	Read Only			Provides heading or yaw as a 16-bit signed integer in steps determined by Resolution (0x6000). If integer value exceeds 65535, the value will be reported as 65535.
0x4006		Yaw/Heading (32-bit)	Unsigned32	Read Only			Provides heading or yaw as a 32-bit signed integer in steps determined by Resolution (0x6000).
0x4007	0x00	Quaternion Output	Unsigned8	Const		0x04	Highest sub-index is 4.
	0x01	Coefficient q1	Signed16	Read Only			Each coefficient is a value in the range -1.0 to +1.0, represented here as 16-bit signed integers, scaled such that the value 10,000 represents 1. The quaternions in this sensor use a non-standard convention, please see section QUATERNION MEASUREMENTS .
	0x02	Coefficient q2					
	0x03	Coefficient q3					
	0x04	Coefficient q4					
0x4008		Measurement Status	Unsigned8	Read Only			A set of bit flags that indicates the status of measurement-related features: bit 0: DKC valid speed measurements received bit 1: Valid heading aiding received bit 2-7: reserved

Object Index	Sub Index	Feature/Name	Data Type	Access	Persistent (see pg. 3)	Factory Default	Details
0x4013		Fault Register	Unsigned32	Read Only			<p>A set of bit flags indicating various failures that may occur in the sensor:</p> <ul style="list-style-type: none"> Bit 0: Accelerometer Value Out-of-range. Bit 1: Gyro Value Out-of-range. Bit 2: Accelerometer Stuck Bit 3: Gyro Stuck Bit 4: Self-Test Failure Bit 5: Accelerometer Failed Self-Test* Bit 6: Gyro Failed Self-Test* Bit 7: Digital Gyro Communications Failure Bit 8: Input Voltage Out-of-range. Bit 9: Temperature Out-of-range Bit 10: Timing Error Bit 11: Flash Memory CRC Failure Bit 12: Firmware ADC calibration problem <p>*Gyro or accelerometer self-test failures may inadvertently occur on power-up if the sensor is moving or subject to vibration.</p> <p><i>NOTE: A Store Parameters "save" command (0x1010) may cause errors to be reported in registers 0x1001 and 0x4013. See page 31.</i></p>
0x6000		Resolution	Unsigned16	Read/Write	Yes	10	<p>Resolution of angle measurements in milli-degrees. Can be set for 1000 (1°); 100 (0.1°); or 10 (0.01°).</p> <p>Note that accuracy of the sensor is specified in the Performance Characteristics, and does not change with this setting. SignalQuest does not promise any performance or stability below 0.1°.</p>
0x6010		Elevation	Signed16	Read Only			<p>Provides Longitudinal Slope (Elevation) as a 16-bit signed integer in steps determined by Resolution (0x6000). If integer value exceeds the range ± 32767, the value will be reported as ± 32767.</p>
0x6011		Elevation Parameter	Unsigned8	Const		0x00	<p>Elevation measurement may not be scaled or inverted.</p>
0x6020		Roll	Signed16	Read Only			<p>Provides Lateral Slope (Roll) as a 16-bit signed integer, in steps determined by the Resolution (0x6000). If integer value exceeds the range ± 32767, the value will be reported as ± 32767.</p>
0x6021		Roll Parameter	Unsigned8	Const		0x00	<p>Roll measurement may not be scaled or inverted.</p>
0x6110		Elevation (32-bit) [slope_long32]	Signed32	Read Only			<p>Provides Longitudinal Slope (Elevation) as a 32-bit signed integer in steps determined by Resolution (0x6000).</p>
0x6120		Roll 32-bit [slope_lateral32]	Signed32	Read Only			<p>Provides Lateral Slope (Roll) as a 32-bit signed integer, in steps determined by the Resolution (0x6000).</p>
0x6511		Device Temperature	Signed8	Read Only			<p>Provides measurement of temperature in 1°C steps, as a signed 8-bit integer.</p>

ACCESSING OBJECTS IN THE DICTIONARY

The Object Dictionary within the device contains values for sensor readings, device setting, and configuration data. These Objects can be read and written using SDO (Service Data Object) messages.

Each CAN message transmitted on the bus contains a CAN-ID that indicates either the source of the message, the destination, or both. Each CAN message transmitted on the bus contains a data field that can hold up to 8 bytes. In CANopen, each device on the bus is assigned a unique NODE ID, which is a value between 1 (0x01) and 127 (0x7F) and generally is the lowest 7 bits of the CAN ID [This device only supports Standard CAN 11 bit identifiers and does not support Extended CAN messaging with 29 bit identifiers.]

Service Data Objects (SDO's) are sent by a client to either request the contents of an object from a device's (or server's) dictionary, or to set the value of a writable object in the device's dictionary. SDO's are sent by the device (or the server) to either send the specific data requested by a client, or to acknowledge and report success or failure of an attempt to write an object by the client. SDO's, and their purpose, are identified by their CAN-ID.

The CAN-ID for sending SDO's to a device is 0x600 plus the NODE ID of the device. The CAN-ID used by the device to send its reply is 0x580 plus the NODE ID of the device. These CAN-ID's are, themselves, identified in the object dictionary at index 0x1200.

READING AN OBJECT

To READ an object in the dictionary, compose a CAN message containing the request, as in this example, which is to read the device revision (object index 0x1018, sub-index 0x03) from a device on NODE ID 0x11:

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
CAN-ID	Function Code	Obj. Index Low Byte	Obj. Index High Byte	Sub-Index	(Ignored)	(Ignored)	(Ignored)	(Ignored)
0x611:	0x40	0x18	0x10	0x03	0x00	0x00	0x00	0x00

In reply, the device should send the following SDO (in this example, the revision code returned is 0x00020001):

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
CAN-ID	Func. Code/Length	Obj. Index Low Byte	Obj. Index High Byte	Sub-Index	Data Low Byte	Data High Byte
0x591:	0x43	0x18	0x10	0x03	0x01	0x00	0x02	0x00

In the reply SDO, the function code also indicates the length of the reply:
 0x43 means the length of the reply is 4 bytes (all 4 data bytes in the message, bytes 5 through 8, are valid);
 0x4B means the length of the reply is 2 bytes (bytes 7 and 8 are invalid); and
 0x4F means the length of the reply is 1 byte (bytes 6 through 8 are invalid).

Note that when the reply contains less than 4 bytes, the remaining bytes in the message may contain any data, but should be ignored.

WRITING AN OBJECT

To WRITE an object in the dictionary, compose a CAN message containing the request, as in this example, which is to write the Event Timer to 20mS (0x0014) in a device on NODE ID 0x11:

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
CAN-ID	Func. Code/ Length	Obj. Index Low Byte	Obj. Index High Byte	Sub-Index	Data Low Byte	Data High Byte	(Ignored)	(Ignored)
0x611:	0x2B	0x00	0x18	0x05	0x14	0x00	0x00	0x00

In this SDO, the function code also indicates the length of data being written:
 0x23 means the length of data is 4 bytes (all 4 data bytes in the message, bytes 5 through 8, are valid);
 0x2B means the length of data is 2 bytes (bytes 7 and 8 are invalid); and
 0x2F means the length of data is 1 byte (byte 6 through 8 are invalid).

In reply, the device should send the following SDO, confirming that the command was valid:

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
CAN-ID	Function Code	Obj. Index Low Byte	Obj. Index High Byte	Sub-Index	(Reserved)	(Reserved)	(Reserved)	(Reserved)
0x591:	0x60	0x00	0x18	0x05	0x00	0x00	0x00	0x00

SDO Errors (ABORT REQUEST)

In any situation where something is wrong with the request, instead of an SDO reply, as shown above, an ABORT REQUEST SDO reply is generated instead. In this case, the client can assume that the request was not obeyed. The ABORT REQUEST SDO reply message will contain an abort code, which will have one of the following values:

Abort Code	Description
0x05030000	Toggle bit not alternated.
0x05040000	SDO protocol timed out.
0x05040001	Client/server command specifier not valid or unknown. [Function code not recognized.]
0x05040002	Invalid block size (block mode only).
0x05040003	Invalid sequence number (block mode only).
0x05040004	CRC error (block mode only).
0x05040005	Out of memory.
0x06010000	Unsupported access to an object. [e.g. Trying to write a communication object when in Operational state.]
0x06010001	Attempt to read a write only object.
0x06010002	Attempt to write a read-only or const object.
0x06020000	Object does not exist in the object dictionary.
0x06040041	Object cannot be mapped to the PDO [Object cannot be mapped to a PDO.]
0x06040042	The number and length of the objects to be mapped would exceed PDO length. [Attempt to map more objects than will fit in a PDO.]
0x06040043	General parameter incompatibility reason.
0x06040047	General internal incompatibility in the device.
0x06060000	Access failed due to a hardware error. [Non-volatile memory write failure.]
0x06070010	Data type does not match, length of service parameter does not match

Abort Code	Description
0x06070012	Data type does not match, length of service parameter too high [Attempt to write data LARGER than dictionary object.]
0x06070013	Data type does not match, length of service parameter too low [Attempt to write data SMALLER than dictionary object.]
0x06090011	Sub-index does not exist.
0x06090030	Invalid value for parameter (download only). [Attempt to write an invalid value to an object.]
0x06090031	Value of parameter written too high (download only).
0x06090032	Value of parameter written too low (download only).
0x06090036	Maximum value is less than minimum value.
0x060A0023	Resource not available: SDO connection
0x08000000	General error
0x08000020	Data cannot be transferred or stored to the application. [Attempt to write to object 0x1010 without 'save'.]
0x08000021	Data cannot be transferred or stored to the application because of local control.
0x08000022	Data cannot be transferred or stored to the application because of the present device state.
0x08000023	Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of a file error).
0x08000024	No data available

For example, here is a request to read object 0x1018, sub-index 0x0A, which doesn't exist in this device's dictionary:

CAN-ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Function Code	Obj. Index Low Byte	Obj. Index High Byte	Sub-Index	(Ignored)	(Ignored)	(Ignored)	(Ignored)
0x611:	0x40	0x18	0x10	0x0A	0x00	0x00	0x00	0x00

The following SDO would be generated in response, containing abort code 0x06090011:

CAN-ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Function Code	Obj. Index Low Byte	Obj. Index High Byte	Sub-Index	Abort Code Low Byte	Abort Code High Byte
0x591:	0x80	0x18	0x10	0x0A	0x11	0x00	0x09	0x06

NOTE: Some Objects can only be written when the device is in a certain state, for instance, Sub Index fields of Objects 0x1800, 0x1801, 0x1802 and 0x1803 can only be written if they are marked 'invalid' in their respective COB ID fields by setting the 0x80000000 bit.

For example, set Object Index 0x1800 Subindex 0x01 to 0xC0000191, its default value is 0x40000191 before attempting to modify Sub-Indices 2 through 6.

Similarly, Sub Index fields of Objects 0x1A00, 0x1A01, 0x1A02 and 0x1A03 can only be written if the associated COB ID is set invalid (bit 0x80000000 is set as previously described) and also have the number of Mapping Parameters set to 0x00.

For example to change any of the mapping parameters in Object 0x1A00, first set Object Index 0x1800 Subindex 0x01 to 0xC0000191, then set Object Index 0x1A00 Subindex 0x00 to 0x00.

DEVICE SETUP AND COMMISSIONING

CANopen devices require some set-up, prior to being installed on a network. It is possible that the default settings are appropriate for your network, but you should at least check them to be certain.

COMMUNICATIONS

BIT RATE: All devices on a CAN network should communicate at the same bit rate. If the bit rate needs to be changed, consult object index 0x2100.

NODE ID: All devices on a CANopen network should have unique NODE ID's. If you need to change the NODE ID of the inclinometer, so that it does not conflict with another device on your network, consult object index 0x2101.

TERMINATOR: A CAN network should be terminated in a 120Ω resistance at the ends of the network furthest from one another. Thus, only two such terminators should exist on the network. Ideally, the terminations should be part of the network wiring. Although some devices, such as this one, contain internal terminations that may be electronically enabled or disabled, such terminations are not ideal, because if the device loses power or fails, the terminator will be off, and the network will be improperly terminated. Also, if you need to remove the device for service, your network will no longer be properly terminated. If you need to use the internal terminator, consult object index 0x2102.

NETWORK MANAGEMENT

Many CANopen networks have a master that expects to control when devices start up, and when they may start transmitting data on the bus. These services fall under the category of Network Management, and consist of the following settings:

DEVICE START-UP: When a device starts up, you can either have it wait for a command from a network master to start operating, or you can have it begin operating immediately. To control this, refer to object index 0x1F80.

HEARTBEAT: A device can be configured to periodically transmit a message to ensure the network that it is still alive and operating. To do so, object index 0x1017 must be set to a non-zero value, equal to the period at which the device must generate heartbeat messages. [Note: this device does not support Guarding.]

ORIENTATION

Depending on how your inclinometer is to be mounted, and which angles you will be measuring, you may need to change the device Orientation. See the section "Device Orientation" [page 18], describing orientation, and refer to object index 0x3000.

TRANSMIT PDO (TPDO) MAPPING AND CONTROL

You may obtain measurements from the inclinometer, simply by polling the objects in the dictionary containing the measurements of interest to you. Generally, these are 0x6010 (Elevation) and 0x6020 (Roll). However, this method uses significantly more bandwidth, as every measurement then requires two SDO's on the bus.

CANopen allows the mapping of measurements of interest into TPDO's, controls whether those TPDO's are transmitted synchronously or asynchronously, the rate at which they are transmitted. TPDO's remove the burden of polling each device on the bus for individual parameters, allowing for more efficient use of bandwidth.

This inclinometer provides up to 4 TPDO's called TPDO1 through TPDO4.

The transmission of TPDO's is controlled through objects 0x1800 through 0x1803. For each TPDO, you can configure the message ID (COB-ID) used to transmit, whether the TPDO is transmitted synchronously or not, and how frequently it is transmitted.

The specific data in each TPDO message is controlled through parameter mapping, objects 0x1A00 through 0x1A03. These objects allow you to specify other object dictionary ID's that are transmitted, and how many bits in each message are used for each parameter.

As shipped, the TPDO's are configured as follows:

TPDO1 is pre-mapped to transmit Elevation and Roll. [See 0x1A00]

TPDO2 is pre-mapped to transmit 3 axes of Acceleration. [See 0x1A01]

TPDO3 is pre-mapped to transmit 3 axes of Gyro. [See 0x1A02]

TPDO4 is not pre-mapped, and disabled (it's COB-ID has bit 31 set). [See 0x1803 and 0x1A03]

To change the settings or the parameter mapping of any TPDO, you must first disable the TPDO, by setting bit 31 in the COB-ID object (one of 0x1800 through 0x1803, sub-index 0x01). If you intend to change the parameter mapping, you must also set the number of parameters in the mapping object (0x1A00 through 0x1A03, sub-index 0) to 0. Once this is done, you are free to change the parameters and transmission settings of the TPDO at will.

For example, let's say you wish to configure TPDO1 to send the 32-bit versions of Elevation and Roll, and you wish to transmit these values every 50mS:

1. Optionally, set the NMT state for Pre-Operational, by sending 0: 0x81 0x11 (assuming NODE ID of this sensor is 0x11). 0: represents the CAN address 0, which is used for ALL NMT commands. This is not necessary, but it decreases traffic on the bus while you execute this procedure, which may make it easier to observe each command and its reply.
2. Disable TPDO1 by setting bit 31 in the COB-ID parameter, 0x1800 sub-index 0x01. Best practice would be to read the existing COB-ID, and then write it back with bit 31 set:

TX 0x611: 0x40 0x00 0x18 0x01 0x00 0x00 0x00 0x00 (read object 0x1800 sub 0x01)

RX 0x591: 0x43 0x00 0x18 0x01 0x80 0x01 0x00 0x40 (COB-ID is 0x40000180)

TX 0x611: 0x23 0x00 0x18 0x01 0x80 0x01 0x00 0xC0 (write COB-ID 0xC0000180)

RX 0x591: 0x60 0x00 0x18 0x01 0x00 0x00 0x00 0x00 (acknowledge)

3. Disable TPDO1 mapping by setting the number of parameters, 0x1A00 sub-index 0x00 to 0.

TX 0x611: 0x2F 0x00 0x1A 0x00 0x00 0x00 0x00 0x00 (write object 0x1A00 sub 0x00 to 0)

RX 0x591: 0x60 0x00 0x1A 0x00 0x00 0x00 0x00 0x00 (acknowledge)

4. Set the first parameter for the 32-bit version of Elevation, which is object 0x6110. The first parameter transmitted is defined by object 0x1A00, sub-index 0x01. It is a 32-bit value, where the high 16-bits represent the object ID, the high 8 bits of the low word represent the sub-index, and the low 8 bits represent the number of bits in the message.

For this mapping, that number is 0x61100020 = object 0x6110, sub-index 0x00, 32-bits (0x20).

TX 0x611: 0x23 0x00 0x1A 0x01 0x20 0x00 0x10 0x61 (write 0x61100020 to 0x1A00 sub-index 0x01)

RX 0x591: 0x60 0x00 0x1A 0x01 0x00 0x00 0x00 0x00 (acknowledge)

5. Set the second parameter for the 32-bit version of Roll, which is object 0x6120.

TX 0x611: 0x23 0x00 0x1A 0x02 0x20 0x00 0x20 0x61 (write 0x61200020 to 0x1A00 sub-index 0x02)

RX 0x591: 0x60 0x00 0x1A 0x02 0x00 0x00 0x00 0x00 (acknowledge)

6. Set the number of parameters in this TPDO to 2 (Elevation32 and Roll32). This re-enabled the TPDO mapping.

TX 0x611: 0x2F 0x00 0x1A 0x00 0x02 0x00 0x00 0x00 (write object 0x1A00 sub 0x00 to 2)
RX 0x591: 0x60 0x00 0x1A 0x00 0x00 0x00 0x00 0x00 (acknowledge)

7. Set the TPDO for non-synchronous operation, by writing 0xFF to 0x1800 sub-index 0x02 (Transmission type).

TX 0x611: 0x2F 0x00 0x18 0x02 0xFF 0x00 0x00 0x00 (write 0xFF to 0x1800 sub-index 0x02)
RX 0x591: 0x60 0x00 0x18 0x02 0x00 0x00 0x00 0x00 (acknowledge)

8. Set the message rate for 50mS, by writing 50 (0x0032) to 0x1800 sub-index 0x05 (Event timer).

TX 0x611: 0x2B 0x00 0x18 0x05 0x32 0x00 0x00 0x00 (write 0x0032 to 0x1800 sub-index 0x05)
RX 0x591: 0x60 0x00 0x18 0x05 0x00 0x00 0x00 0x00 (acknowledge)

9. The TPDO is now configured. Re-enable transmission of the TPDO by writing the desired COB-ID to object 0x1800 sub-index 0x01.

TX 0x611: 0x23 0x00 0x18 0x01 0x80 0x01 0x00 0x40 (write 0x40000180 to COB-ID)
RX 0x591: 0x60 0x00 0x18 0x01 0x00 0x00 0x00 0x00 (acknowledge)

10. If you set the state for Pre-Operational in step 1, you may now set the state for Operational again, by sending the NMT command 0: 0x01 0x11 (assuming NODE ID of this sensor is 0x11).

Synchronous TPDO's require the existence of a SYNC producer on the network. The SYNC producer transmits SYNC messages at a fixed rate, that all the devices on the network use to time the transmission of their TPDO's. Normally, SYNC messages are transmitted on CAN ID 0x080, as indicated by object index 0x1005.

SYNC message may optionally contain a one-byte counter, which is incremented upon the transmission of each message, and wraps around after reaching some value (see object index 0x1019).

When configuring a synchronous TPDO, you may specify how many SYNC messages occur between each transmission of the TPDO, 0x01 indicates that the TPDO should be transmitted after every SYNC, 0x02 indicates after every other SYNC, etc. You can also specify the SYNC counter value on which the FIRST TPDO will be transmitted (object index 0x1800 through 0x1803, sub-index 0x06).

To control whether a TPDO is sent synchronously or asynchronously, refer to object index 0x1800 through 0x1803, sub-index 0x02.

Notes:

1. The COB ID must always have the RTR bit 30 (0x40000000) set, since RTR operation is not supported on this device.
2. Transmission Type may be 0x01 through 0xF0, which sets synchronous operation; or 0xFE or 0xFF which sets asynchronous operation.
 - a. Type 0xFE "event-driven (manufacturer-specific)" behaves identically to Type 0xFF on this device.
 - b. The RTR types 0xFC and 0xFD are not supported on this device.
 - c. Types 0xF1 through 0xFB are reserved.
 - d. The Synchronous types 0x01 through 0xF0 are supported.
3. Inhibit Time is used to limit the spacing between messages. This is supported and functional, but since the spacing between messages on this device is regular, it is not of much utility.
4. On earlier versions of this sensor, it was not necessary to set the invalid bit (0x80000000) in sub-index 0x01 before writing to the other sub-indices. This behavior was not compliant with CANopen standards, and thus was changed.

RECEIVE PDO (RPDO) MAPPING AND CONTROL

Some features in SQ-GIX sensors, such as Speed Aiding and Heading Aiding requires the sensor to gather information from the CAN bus of the vehicle on which it is mounted, from other CAN devices that are periodically transmitting the necessary information. For example, Speed Aiding requires that the sensor receive measurements of the vehicle speed, from wheel speed sensors or encoders.

The RPDO mechanism in CANopen allows a sensor to be configured to recognize and receive these periodic messages, automatically writing the values to the appropriate objects in the dictionary, as these messages are received.

Your sensor is configured at the factory to use those features you have requested.

This sensor provides 4 RPDO's, called RPDO1 through RPDO4.

The RPDO's are primarily controlled objects 0x1400 through 0x1403. For each RPDO, you can configure the message ID (COB-ID) which the sensor will use to identify the message when it is transmitted, whether that message is transmitted synchronously or not, and how frequently it is transmitted.

The specific data in each RPDO message is controlled through parameter mapping, objects 0x1600 through 0x1603. These objects allow you to specify which object dictionary ID's data should be written to when received, and the location and length of each parameter in the message.

As shipped, the RPDO's are configured as follows:

RPDO1 is pre-mapped to receive Heading and Heading Standard Deviation. [See 0x1600]

RPDO2 through 4 are disabled. [See 0x1601 through 0x1603].

To change the settings or the parameter mapping of any RPDO, you must first disable the RPDO, by setting bit 31 in the COB-ID object (one of 0x1400 through 0x1403, sub-index 0x01). If you intend to change the parameter mapping, you must also set the number of parameters in the mapping object (0x1600 through 0x1603, sub-index 0) to 0. Once this is done, you are free to change the parameters and settings of the RPDO at will.

For example, let's say you wish to configure RPDO2 to receive Heading Standard Deviation from a message on address 0x123, and you expect those messages to be transmitted no more than 200mS apart:

1. Optionally, set the NMT state for Pre-Operational, by sending 0: 0x81 0x11 (assuming NODE ID of this sensor is 0x11). 0: represents the CAN address 0, which is used for ALL NMT commands. This is not necessary, but it decreases traffic on the bus while you execute this procedure, which may make it easier to observe each command and its reply.
2. Disable RPDO1 by setting bit 31 in the COB-ID parameter, 0x1401 sub-index 0x01. Best practice would be to read the existing COB-ID, and then write it back with bit 31 set:

```
TX 0x611: 0x40 0x01 0x14 0x01 0x00 0x00 0x00 0x00 (read object 0x1401 sub 0x01)
RX 0x591: 0x43 0x01 0x14 0x01 0x00 0x03 0x00 0x00 (current COB-ID is 0x00000300)
TX 0x611: 0x23 0x01 0x14 0x01 0x00 0x03 0x00 0x80 (write COB-ID 0x80000300)
RX 0x591: 0x60 0x01 0x14 0x01 0x00 0x00 0x00 0x00 (acknowledge)
```

3. Disable RPDO1 mapping by setting the number of parameters, 0x1601 sub-index 0x00 to 0.

```
TX 0x611: 0x2F 0x01 0x16 0x00 0x00 0x00 0x00 0x00 (write object 0x1601 sub 0x00 to 0)
RX 0x591: 0x60 0x01 0x16 0x00 0x00 0x00 0x00 0x00 (acknowledge)
```

4. Set the first parameter for the 16-bit value Heading Standard Deviation, which is object 0x3110 sub-index 0x03. The first parameter is defined by object 0x1601, sub-index 0x01. It is a 32-bit value, where the high 16-bits

represent the object ID, the high 8 bits of the low word represent the sub-index, and the low 8 bits represent the number of bits in the message.

For this mapping, that number is $0x31100310 = \text{object } 0x3110, \text{ sub-index } 0x03, 16\text{-bits } (0x10)$.

TX $0x611: 0x23 0x01 0x16 0x01 0x10 0x03 0x10 0x31$ (write $0x31100310$ to $0x1601$ sub-index $0x01$)
RX $0x591: 0x60 0x01 0x16 0x01 0x00 0x00 0x00 0x00$ (acknowledge)

5. Set the number of parameters in this RPDO to 1. This re-enables the RPDO mapping.

TX $0x611: 0x2F 0x01 0x16 0x00 0x01 0x00 0x00 0x00$ (write object $0x1601$ sub $0x00$ to 1)
RX $0x591: 0x60 0x01 0x16 0x00 0x00 0x00 0x00 0x00$ (acknowledge)

6. Set the RPDO for non-synchronous operation, by writing $0xFF$ to $0x1401$ sub-index $0x02$ (Type).

TX $0x611: 0x2F 0x01 0x14 0x02 0xFF 0x00 0x00 0x00$ (write $0xFF$ to $0x1401$ sub-index $0x02$)
RX $0x591: 0x60 0x01 0x14 0x02 0x00 0x00 0x00 0x00$ (acknowledge)

7. Set the message rate for 200mS, by writing 200 ($0x00C8$) to $0x1401$ sub-index $0x05$ (Event timer).

TX $0x611: 0x2B 0x01 0x14 0x05 0xC8 0x00 0x00 0x00$ (write $0x00C8$ to $0x1401$ sub-index $0x05$)
RX $0x591: 0x60 0x01 0x14 0x05 0x00 0x00 0x00 0x00$ (acknowledge)

8. Set the COB-ID for $0x123$, which also re-enables this RPDO. Do this by writing $0x00000123$ to $0x1401$ sub-index $0x01$.

TX $0x611: 0x23 0x01 0x14 0x01 0x23 0x01 0x00 0x00$ (write $0x00000123$ to COB-ID)
RX $0x591: 0x60 0x01 0x14 0x01 0x00 0x00 0x00 0x00$ (acknowledge)

9. If you set the state for Pre-Operational in step 1, you may now set the state for Operational again, by sending the NMT command 0: $0x01 0x11$ (assuming NODE ID of this sensor is $0x11$).

DKC SPEED INPUT RESOLUTION

If you are using Speed Aiding, you must provide the scale factor that converts the integers being transmitted by your speed sensors into a velocity in meters/second. To do this, you must set object $0x3010$ for the resolution of your sensor. This object is a 32-bit integer that represents the resolution in steps of 1×10^{-9} meters/sec.

For example, if your wheel sensors transmit speed in integer increments of 0.1mm/sec:

$$0.1\text{mm/sec} / 1 \times 10^{-9} \text{ meters/sec} = 1 \times 10^{-4} \text{ meters/sec} / 1 \times 10^{-9} \text{ meters/sec} = 1 \times 10^5 = 100,000$$

You would write 100,000 to object $0x3010$.

SAVING SETTINGS TO NON-VOLATILE MEMORY

After changing any of the settings marked ‘persistent’ to conform to your network and requirements, the settings can be saved to non-volatile memory. Otherwise, the next time the device is power-cycled, the settings will all revert to their original state.

[NOTE: Changes to Baud Rate [0x2100], Node ID [0x2101], Enable Terminator [0x2102] and Orientation [0x3000] will not take effect until the next time the device is power-cycled.]

To save all settings marked ‘persistent’ to non-volatile memory, use the following SDO (in this example the NODE ID is 0x11):

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
CAN-ID	Func. Code/ Length	Obj. Index Low Byte	Obj. Index High Byte	Sub-Index	‘s’	‘a’	‘v’	‘e’
0x611:	0x23	0x10	0x10	0x01	0x73	0x61	0x76	0x65

RESTORING SAVED SETTINGS

All settings can be returned to their Saved values by issuing a command to Object 0x1011.

[NOTE: Changes to Baud Rate [0x2100], Node ID [0x2101], Enable Terminator [0x2102] and Orientation [0x3000] will not take effect until the next time the device is power-cycled.]

To restore all settings marked ‘persistent’, use the following SDO (in this example the NODE ID is 0x11):

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
CAN-ID	Func. Code/ Length	Obj. Index Low Byte	Obj. Index High Byte	Sub-Index	‘l’	‘o’	‘a’	‘d’
0x611:	0x23	0x11	0x10	0x01	0x6C	0x6F	0x61	0x64

CANOPEN NETWORK MANAGEMENT

CANopen defines four states of a network device: Initialization, Pre-Operational, Operational, and Stopped.

Initialization

After a power-up or reset, the device is transiently in the Initialization state. In this state, the device is preparing for operation by reading the object dictionary out of non-volatile memory, and preparing for operation. If object index 0x1F80 is set to 0x00000008, then the device will transition directly to Operational state. If it is set to 0x00000000, it enters the state Pre-Operational.

Pre-Operational

A device whose object index 0x1F80 is set to 0x00000000 will enter Pre-Operational state, and remain there until it receives a network management command to start.

Upon entering the Pre-Operational state for the first time, the device will transmit a boot-up message (this example assumes NODE ID 0x11):

	Byte 1
CAN-ID	State
0x711:	0x00

Only the first message upon entering pre-operational state will contain the State code 0x00.

Operational

A device whose object index 0x1F80 is set to 0x00000008, or one who has received the network start remote node command will be in the Operational state.

In Operational state, writes to objects that can affect communications are prevented.

The device may be placed in the Stopped state, through a network management stop command. In the Stopped state, TPDO's are not sent, but Heartbeats (if enabled) are sent.

Network Start Remote Node Command

The command to start the node (NODE ID 0x11 in this example) is as follows:

	Byte 1	Byte 2
CAN-ID	NMT Command	NODE ID
0x000:	0x01	0x11

Upon receiving this command, the node transitions from state Stopped or Pre-Operational to Operational.

Network Stop Remote Node Command

The command to STOP the node (NODE ID 0x11 in this example) is as follows:

	Byte 1	Byte 2
CAN-ID	NMT Command	NODE ID
0x000:	0x02	0x11

Upon receiving this command, the node transitions from state Operational or Pre-Operational to Stopped.

Network Enter Pre-Operational Command

The command to place the node into Pre-Operational state (NODE ID 0x11 in this example) is as follows:

	Byte 1	Byte 2
CAN-ID	NMT Command	NODE ID
0x000:	0x80	0x11

Upon receiving this command, the node transition from state Stopped or Operational to Pre-Operational.

Network Reset Node Command

The command to reset the Node (same as Power-Up) is as follows:

	Byte 1	Byte 2
CAN-ID	NMT Command	NODE ID
0x000:	0x81	0x11

Upon receiving this command, the device resets, and transitions through Initialization.

Network Reset Communications Command

The command to reset communications is as follows:

	Byte 1	Byte 2
CAN-ID	NMT Command	NODE ID
0x000:	0x82	0x11

Upon receiving this command, the device resets all persistent communications parameters to those last saved in non-volatile memory, and all others to default values, and re-enters the Pre-Operational or Operational state, as indicated by object 0x1F80.

CANOPEN HEARTBEAT

If object index 0x1017 is set to a non-zero value, then heartbeat messages will be transmitted by the device when it is in states Pre-Operational, Operational, or Stopped. The object 0x1017 indicates the time between heartbeat messages in mSec.

A heartbeat message, for a device on NODE ID 0x11, and in the Operational State, looks like this:

	Byte 1
CAN-ID	State
0x711:	0x05

The State can be one of the following:

- 0x00: Boot-up (the message transmitted the first time the device enters Pre-Operational state).
- 0x04: Stopped
- 0x05: Operational
- 0x7F: Pre-Operational

CANOPEN SYNC

TPDO's may be transmitted synchronously. When set for synchronous operation, a SYNC PRODUCER must be present on the bus, and it must generate periodic SYNC messages.

SYNC messages may contain a one byte counter that starts counting at 1, is incremented for each SYNC message transmitted, and wraps back around to 1 after reaching a maximum value.

A SYNC message looks like this. In this example, the SYNC ID is set for 0x080, and the counter is at 1.

	Byte 1
CAN-ID	Counter
0x080:	0x01

If set for synchronous operation, a TPDO message is transmitted upon reception of every SYNC message, every other SYNC message, every third, up to every 240th SYNC message, as indicated by sub-index 0x02 in the TPDO settings object (index 0x1800 through 0x1802).

TPDO1: QUATERNION OUTPUT MESSAGE [FACTORY DEFAULT TPDO1 MAPPING]

TPDO1 is set by default to transmit the quaternion representation of its pose.

The Object Dictionary describes the TPDO message, the CAN ID used, whether the measurement is transmitted synchronously, how often the message is sent, which objects are included in the transmission, etc.

Object 0x1800 describes the TPDO1 message. It is transmitted on CAN-ID 0x180 + NODE ID. Its Parameter mapping is described by Object 0x1A00.

A TPDO1 message appears as follows. In this example, NODE ID is 0x11, and the quaternion representation of pose is [0.5, 0.70219, 0.485489, -0.146]. In integer format, scaled by 10,000, this is [5000 (0x1388), 7022 (0x1B6E), 4855 (0x12F7), -1460 (0xFA4C two's complement signed representation)].

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
CAN-ID	q1 Low Byte	q1 High Byte	q2 Low Byte	q2 High Byte	q3 Low Byte	q3 High Byte	q4 Low Byte	q4 High Byte
0x191:	0x88	0x13	0x6E	0x1B	0xF7	0x12	0x4C	0xFA

TPDO2: ACCEL MESSAGE [FACTORY DEFAULT TPDO2 MAPPING]

By default, this message provides the Accelerometer readings from the 3 axes [X, Y and Z]. Note that the interpretation of axis Orientation is determined by the Orientation Object 0x3000.

The Object Dictionary describes the TPDO message, the CAN ID used, whether the measurement is transmitted synchronously, how often the message is sent, which objects are included in the transmission, etc.

Object 0x1801 describes the TPDO2 message. It is transmitted on CAN-ID 0x280 + NODE ID. Its Parameter mapping is described by Object 0x1A01.

A TPDO2 message appears as follows: (In this example, NODE ID is 0x11, Accel. X is 30 mG [30 decimal, or 0x001E], Accel. Y is 23 mG [23 decimal or 0x0017] and Accel. Z is -999 mG [-999 decimal, or 0xFC18, twos- complement signed representation]).

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
CAN-ID	Accel. X Low Byte	Accel. X High Byte	Accel. Y Low Byte	Accel. Y High Byte	Accel. Y Low Byte	Accel. Y High Byte
0x291:	0x1E	0x00	0x17	0x00	0x18	0xFC

TPDO3: GYRO MESSAGE [FACTORY DEFAULT TPDO3 MAPPING]

By default, this message provides the Gyro readings from the 3 axes [X, Y and Z]. Note that the interpretation of axis Orientation is determined by the Orientation Object 0x3000.

The Object Dictionary describes the TPDO message, the CAN ID used, whether the measurement is transmitted synchronously, how often the message is sent, which objects are included in the transmission, etc.

Object 0x1802 describes the TPDO3 message. It is transmitted on CAN-ID 0x380 + NODE ID. Its Parameter mapping is described by Object 0x1A02.

A TPDO3 message appears as follows: (In this example, NODE ID is 0x11, Gyro X is 3.0 deg/sec [30 decimal, or 0x001E], Gyro Y is 2.3 deg/sec [23 decimal, or 0x0017] and Gyro Z is -1.0 deg/sec [-10 decimal, or 0xFFF6, twos- complement signed representation]).

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
CAN-ID	Gyro X Low Byte	Gyro X High Byte	Gyro Y Low Byte	Gyro Y High Byte	Gyro Y Low Byte	Gyro Y High Byte
0x391:	0x1E	0x00	0x17	0x00	0xF6	0xFF

TPDO4: MESSAGE [FACTORY DEFAULT TPDO4 MAPPING]

This Factory Default is to disable this message.

The Object Dictionary describes the TPDO message, the CAN ID used, whether the measurement is transmitted synchronously, how often the message is sent, which objects are included in the transmission, etc.

Object 0x1803 describes the TPDO4 message. It is transmitted on CAN-ID 0x480 + NODE ID. Its Parameter mapping is described by Object 0x1A03.

CANOPEN FURTHER READING

For more information, please refer to:
 CiA301, “CANopen Application Layer and Communication Profile”
 CiA302, “CANopen Additional Application Layer Functions”
 CiA410, “CANopen Device Profile for Inclinometer”

SPEED AIDING

For ground-based driving vehicle applications, the sensor's accuracy can be further enhanced with velocity aiding. To do so, the user simply sends a forward speed measurement to the sensor at a rate of 100 Hz (minimum of 20 Hz), as a TPDO onto the CAN bus, using an RPDO in this sensor to map that value to one of the speed objects 0x3101 (32-bit) or 0x3102 (16-bit).

When the sensor starts detecting writes to this object, it will activate the Speed Aiding, and start using the velocity updates to correct the angle measurements.

The resolution of the speed integer can be configured using object 0x3010.

The sensor's Internal X axis (see the section on Orientation) must be aligned with the forward direction of the vehicle. And the sensor's Internal Z axis must be aligned toward the bottom of the vehicle.

The value written to this object must represent a speed of less than or equal to 30 meters/second (about 67 miles/hour, or 108 kilometers per hour). If no valid speed is received for a period of 500ms, the Speed Aiding will be turned off.

HEADING AIDING

Heading Aiding can be used to improve the dynamic accuracy of angle measurements. For the best performance, use the time synchronization feature. This will align the time of validity of the heading aiding message with the sensor's reference clock. If this is not possible, minimize delay and jitter in the aiding message to the SQ-GIX unit.

Two use cases:

For yaw rates of 6 °/s, keep delay / jitter below 250 ms.

For yaw rates of 90 °/s, keep delay / jitter below 16.8 ms.

Exceeding these limits will result in degraded or erratic behavior.

To use Heading Aiding, the user transmits Heading, Heading Standard Deviation, and Heading Time-of-Validity periodically in a TPDO onto the CAN bus, using an RPDO in this sensor to map that message to the corresponding objects: 0x3110 sub-index 0x01 for Heading; 0x3110 sub-index 0x03 for Heading Std. Deviation; and 0x3110 sub-index 0x02 for Heading Time-of-Validity.

Heading and Standard Deviation are both represented as a 16-bit unsigned integer, in steps of 0.01°. A heading of 0.00° is North. Heading can range from 0.00° to 360.00°. The Standard Deviation should represent the accuracy of the provided heading measurement.

Heading Time-of-Validity is a 16-bit unsigned integer, representing time in milli-seconds. It should be a measure of when the heading measurement was made relative to the time the message was queued for transmission. If the heading measurement was made 120 milli-seconds prior to queueing the TPDO, then the value should be 120.

When the sensor detects an RPDO mapped to the three heading values, and all the values are valid, it will activate Heading Aiding, and start using the heading updates to stabilize the angle measurements. If no valid heading measurements are received for 1500 milli-seconds, heading aiding will time out and will no longer be used. The Yaw/Heading angle of the sensor is available in objects 0x4005 and 0x4006. The yaw angle is also encoded by the quaternion object 0x4007.

YAW/HEADING MEASUREMENTS

Yaw outputs are atypical for inclinometer-class sensors. Yaw is defined as right-handed rotation of a vector about gravity, where the vector is a projection of the Internal X axis onto a plane that is horizontal. Because the Yaw measurement is in a plane that is horizontal (perpendicular to gravity), it cannot be measured by an accelerometer detecting gravity. Instead, the sensor maintains its heading solution by integrating a 3-axis gyroscope. This integration process is subject to drift and will accumulate unbounded error in the absence of external aiding. The rate at which the system accumulates heading error is driven by complex processes that are affected by a) time and b) environmental factors like temperature, temperature rate, acceleration and shock events, and vibration at the sensor mounting location.

Because the GravityGyro™ cannot directly measure heading, its unaided Yaw angle output is relative to the initial heading of the sensor at power-up, but the reference can be changed using Heading Aiding.

When the sensor is power-cycled or reset, the direction of the +X internal axis is assumed to be coincident with North (0° Yaw). The first data packet with a Heading Aiding measurement provided to the GravityGyro™ after a power-cycle, reset, or outage of heading updates will slave the heading output, causing the sensor's Yaw output to immediately “snap” to the commanded value. Subsequent heading messages are processed by the data-fusion algorithm and used to improve the dynamic accuracy of the tilt angle estimates.

The following features in this sensor help to improve the accuracy of the Yaw Measurement.

Yaw Reset (Object 0x3211)

The Yaw Reset command may be issued that will reset the yaw measurement (and the integrator) to 0. This happens automatically when the sensor is powered up or reset. This command should be used any other time you wish to establish a reference point for your yaw measurement.

Motion Hold (Object 0x3210)

A Motion Hold command allows your system to turn off updates to the Yaw measurement, any time the system has knowledge that the sensor is not moving. While Motion Hold is ON, the Yaw measurement will not be affected by Angular Random Walk. But it won't be updated in response to real motion either, so it is important that you turn off Motion Hold before the sensor starts to move, otherwise significant errors in the Yaw Measurement will occur.

Additionally, the sensor uses the command “Motion Hold On” to improve the dynamic accuracy of the sensor. Issuing a Motion Hold On command causes the sensor to refine its gyroscope calibration while the platform is known to be stationary. Thus, a Motion Hold can be intentionally performed before a vehicle maneuver to improve the dynamic accuracy of the sensor.

Heading Aiding (Object 0x3110, updated via RPDO)

If you have an independent measurement of the vehicle heading, you may also choose to use the Heading Aiding command.

External measurements of heading are processed by the onboard data-fusion algorithm to align the sensor's Yaw angle output with the reference input. When aided, the Yaw angle solution provided by the sensor is a blended solution that combines the high-rate gyroscope signal with the low-rate Heading input. The gyroscope effectively smooths/interpolates the Heading reference providing a high-bandwidth, high-data-rate Yaw angle output with accuracy and stability bounded by that of the reference input.

See the section on Heading Aiding for more information.

QUATERNION OUTPUTS

This sensor may transmit its pose using either Euler or quaternion outputs. The Euler angle representation of sensor pose (Elevation or Pitch, Roll and Yaw) contains a singularity when the sensor is in gimbal lock (Elevation at +90° or -90°),

because Roll cannot be represented in that pose. A quaternion is a representation of the pose of the sensor that does not suffer from a singularity in gimbal lock.

In the Euler representation, if you start with a sensor oriented with the measurement axis pointing “north” and with the down-axis pointing directly toward the earth, the reference for the Local (L) frame, then you would perform three different rotations, Yaw (ψ), Pitch (θ), and Roll (φ) to reach its current pose, the Sensor (S) frame.

A quaternion representation is derived from both a 3-dimensional unit vector, referred to as \mathbf{e} , and an angle, referred to as ϑ . Mathematically, there exists an \mathbf{e} and ϑ such that if you start with a sensor that is oriented with the reference for the Local frame (as described above) and you rotate the sensor by angle ϑ about the unit vector \mathbf{e} , you will bring the sensor into the same pose as the three Euler rotations (the Sensor frame).

A quaternion is represented by four coefficients, q_1 through q_4 . Each coefficient can range in value from -1.0 to +1.0. Three of the values represent a vector, and one represents a scalar. Different conventions exist to determine how the four coefficients are assigned. The SignalQuest convention is to place the vector component in elements q_1 through q_3 , and the scalar component in q_4 .

To derive the quaternion, the following math is used:

$$\mathbf{e} = \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix} = \text{The three-dimensional unit vector (in the Local frame) about which we will rotate.}$$

$$\vartheta = \text{The angle we will rotate about the vector } \mathbf{e} \text{ to attain the current sensor pose.}$$

$$\mathbf{q}_L^S = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \text{Quaternion that rotates from the Local (L) frame to the Sensor (S) frame.}$$

$$\mathbf{q}_L^S = \begin{bmatrix} \mathbf{e} \sin \frac{\vartheta}{2} \\ \cos \frac{\vartheta}{2} \end{bmatrix}$$

The rotation matrix that translates the Local frame to the sensor frame can be computed from the coefficients of the quaternion as follows:

$$\mathbf{C}_L^S = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_2q_1 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_3q_1 + q_2q_4) & 2(q_3q_2 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix}$$

BUILT-IN TESTS AND FAILURES

The sensor can detect some failures in its internal components. A self-test of accelerometers and gyros is run upon power-up or reset of the sensor, and the sensor continues to monitor for certain conditions while it is operating.

Failures are reported through the CANopen Error Register (0x1001) and through the Fault Register (0x4013). The CANopen Error Register reports a summary – two flags are set in this register if any flags are set in the Fault Register. The Fault Register provides details as to which type of failure was detected.

Some conditions that may occur during normal operation may cause failures to be reported that are not real:

- Movement or vibration of the sensor during start-up (while the internal self-test is running).
- Issuing a CANopen “save” command (writing “save” to Object ID 0x1010, sub-index 0x01).
- Power-supply brownouts, or overshoots beyond the recommended supply voltage range.

Usually, spurious failure reports caused by one of these conditions will be transient, and the fault flags will automatically clear. In some cases, however, the flags may remain set afterwards. In this case, you may clear them by issuing a reset command to the sensor.

If a failure is reported, and you have reason to believe that one of these conditions has existed, you should ignore the failure. However, if the failure is reported consistently, this may indicate a real failure.

ABSOLUTE MAXIMUM RATINGS

PARAMETER	NOTES	MIN		MAX	UNITS
Supply voltage		-36		36	V _{dc}
Voltage on CANH, CANL with respect to GND		-24		24	V _{dc}
CANH-CANL differential		-24		24	V _{dc}

ELECTRICAL CHARACTERISTICS

PARAMETER	NOTES	MIN	TYP	MAX	UNITS
Supply voltage (V+)		5		30	V _{dc}
Supply current	At 12V across full temperature range	20	22	27	mA
CAN bus rate			250		kbps
Power pass through (from A to B connector)				0.5	A
Inrush current				2 4	A, 400 us max A, 200 us
Hold up time	12 V supply disconnected 24 V supply disconnected	6 12			ms
CAN Terminator	Software enabled		120		Ω

OUTPUT CHARACTERISTICS

PARAMETER	SPECIFICATION	UNITS	NOTES
Axes	3		Full 6DOF IMU internal
Roll angle range	±180	°	Roll valid only if: -75° ≤ Elevation ≤ +75°
Elevation angle range	±90	°	
Heading angle range	0 - 360	°	0° = North. Yaw valid only if Elevation is between -75 and +75°. Yaw reference valid only if sensor is receiving Heading updates.
Angular resolution	0.1	°	
Dynamic accuracy limit	300	° / sec	Beyond range accuracy may be degraded
Dynamic compensation limit	900	° / sec	Beyond range dynamic errors may become large
Operating vibration limit	5	g peak	Beyond range for sustained periods may cause large errors
Static vibration limit	0.003	g RMS	5 to 500 Hz random flat spectrum. Within range, static accuracy and settling time are met.
Angle response time (lag)	0.1	s	Delay between true angle and reported angle (max)
Sampling Synchronization	4	ms	Typical 1 to 2 ms synch to master
Output Data Rate	10	Hz	Configurable, default listed
Settling time	1	s	Time to recover from dynamic accuracy to static accuracy

PERFORMANCE SPECIFICATIONS

PARAMETER	SPECIFICATION*	UNITS	NOTES
Noise (Elevation/Roll)	0.1	°RMS	
Case null alignment	0.1	°RMS	Measured with respect to the two case reference surfaces on a surface plate
Static accuracy (Elevation/Roll)	0.1	°RMS	RMS error value of 12 equally spaced orientations around the unit sphere
Dynamic accuracy (Elevation/Roll)	0.100 °/g + 3% of Δ angle		Measured as RMS error compared against an encoder. Δ angle is the difference in angle from the beginning to the end of the test where the test begins in a static condition
Zero-point temp drift	0.005	° / °C	
Long term drift	0.5	°	
Shock, acceleration and vibration error	0.5	°RMS	Independent linear conditions: <ul style="list-style-type: none"> ▪ 1 gRMS random vibration, flat spectrum 5 Hz to 500 Hz ▪ 1 g acceleration, 1 second step ▪ 20 g ½ sin 10 mSec ▪ 100 g ½ sin 0.1 mSec

* 1 sigma, 25 °C unless otherwise noted.

ENVIRONMENTAL PROTECTION

PARAMETER	SPECIFICATION	UNITS	NOTES
Altitude	12,000	Feet	
Ingress protection	IP68/69K		
Shock survival	1000	g	½ sin 0.1 mSec, 3x any axis
EMC	<ul style="list-style-type: none"> ▪ EN 13309 ▪ ISO 14982 ▪ ISO 13766 (200 V/m) 		
Housing	Anodized Aluminum 6061		

CONNECTOR DIAGRAM

M12, Male/Female, 5-pin Connector, (A-coded)



PIN DESCRIPTIONS

PIN	NAME	DESCRIPTION
1	CAN Shield	Shield (Case)
2	CAN V+	Supply Voltage
3	CAN GND	Ground (Power Return)
4	CAN H	CAN Bus High
5	CAN L	CAN Bus Low

Note: This is an Industry Standard M12 CAN Bus Pinout and is compliant with industry standard DeviceNet™ or CANopen network cabling standards and products.

CAN TERMINATION:

A properly terminated CAN bus network has a pair of 120 Ohm terminators at the two points furthest from each other on the CAN network.

In typical CAN networks either none of the devices have internal CAN Termination enabled (external dedicated CAN terminators are used) or at most 2 device(s) have internal CAN terminator(s) enabled.

Typical CAN network installations have separate terminators designed into the network to simplify debugging and improve reliability since the removal or failure of a device containing an internal CAN terminator will usually cause the network to fail to function properly and be difficult to debug until CAN terminators are installed (to replace the one lost in the failed device).

If this device contains an internal CAN terminator, this device should be installed at the farthest end of the CAN Bus and replaces one of the two network terminators. It should not be used in the middle of the network or on a 'Drop', unless the 'Drop' is long enough to put the sensor at the furthest distance from the other network Terminator, (in which case it is not a 'Drop').

CURRENT LIMIT AND FUSING

The use of a current limited supply for positive and negative rails and 0.5 A fuses in series with communication lines is recommended to protect against damaging or dangerous high current conditions caused by cable mis-wiring, ground loops or welding in the field.

SENSOR WIRE HARNESS SHIELDING & WIRING REQUIREMENTS

SignalQuest has tested and recommends one of the following three options for use with the sensors. The three options are listed in order of recommendation.

1. Two Shielded Twisted pairs with Drain wire, 5 wires total (1 pair for power, 1 pair for CANH/CANL, 1 Shield/Drain wire)
 - a. DeviceNet, CANopen compatible wiring, or SAE J1939/11 wiring with the addition of a shielded twisted pair for power, with shield connected to case ground via grounding lug on case.
2. Two Unshielded Twisted pairs, 4 wires total, (1 pair for power, 1 pair for CANH/CANL)
 - a. SAE J1939/15 compliant wiring with the addition of a twisted pair for power.
3. Four Wire Spiral Wound, 22 AWG, 1 to 3 twists per foot. This option is not supported by the J1939 specifications but is widely used by industry.

Shielded twisted pair (Option 1 above) provides the best performance. Options 2 and 3 are sufficient for certain cost conscious applications where shielded wiring is not desirable.

In order to meet certain high field immunity requirements beyond the specifications listed above, the following installation requirements must be observed:

1. The SHIELD signal (if used) must have a direct connection to GND at some point in the wiring harness and this point should be as close as possible to the power supply (or battery) negative terminal.
2. Shielded cabling for all communications/power signals between the sensor and the CAN bus controller is required.
3. It is also required to strap the sensor's enclosure to a chassis ground via the 0.25" deep 6-32 screw hole available on the surface of the sensor. Use of a screw, washer, lock washer and braided copper mesh to attach the enclosure to a bare metal tapped hole in the vehicle's metal floor is recommended. Conductive epoxy may be useful to prevent the screws from corroding or coming loose over time.
4. Many customers have met EMI/RFI requirements without Shielded cable (2 above) or grounding at the sensor (3 above). Likewise requirements have been met using Wiring Option 2 (2 unshielded twisted pairs per SAE J1939/15). The customer is responsible for testing to prove this.

DEVICE ORIENTATION

TERMINOLOGY

Label X means a vector parallel to the “X” arrow printed on the housing label.

Label Y means a vector parallel to the “Y” arrow printed on the housing label.

Label Z means a vector passing through the housing, from the label side, pointing toward the potted side opposite the label. (The Z axis is indicated by the ⊗ symbol on the label).

Internal X means a vector parallel to the internal X axis, whose relationship to a label axis is set by the Orientation.

Internal Y means a vector parallel to the internal Y axis, whose relationship to a label axis is set by the Orientation.

Internal Z means a vector parallel to the internal Z axis, whose relationship to a label axis is set by the Orientation.

Gravity means a vector pointing from the device toward the center of the earth.

Horizontal means a vector pointing at a right angle to gravity.

Elevation is the angle between the Internal X axis and a plane that is horizontal (perpendicular to gravity). Elevation is 0° when the Internal X axis is in a plane that is horizontal. Elevation increases when the Internal +X axis is tilted up, away from the surface of the earth.

Roll is right-handed rotation about the internal X axis. Roll is 0° when the Internal Y axis is horizontal, and the Internal Z axis is within 90° of the gravity vector, toward the surface of the earth. Right-handed rotation means that Roll increases when the sensor is rotated clockwise about the Internal X axis, with the internal X axis pointing away from the viewer.

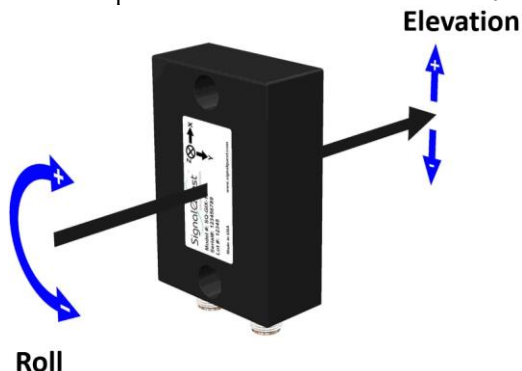
Yaw is right-handed rotation of a vector about gravity, where the vector is a projection of the Internal X axis onto a plane that is horizontal. Yaw ranges from 0° to 360°. The 0° reference for Yaw is arbitrary and is initially established as the position where the sensor is powered-on. Without any other aiding, Yaw may drift due to random walk, even when the sensor is still. In sensors with Heading Aiding, the 0° reference is updated based on received Heading information.

Home Position is the angle for Elevation or Roll that is at the center of the measurement range (0°).

The complete sensor Home Position is when both Elevation and Roll are in the Home Position, where the Internal X and Y axes are both horizontal, and the Internal +Z axis is parallel to gravity, pointing toward the surface of the earth.

Gimbal Lock occurs when Elevation is 90° away (in either direction) from the Elevation Home Position (also meaning that the Internal X Axis is parallel to gravity, pointing up or down). When the device is near the Gimbal Lock position, Roll and Yaw cannot be measured. The Roll and Yaw measurements may be stable, or it may drift about at random, but they cannot be relied upon in Gimbal Lock. The specified range for Elevation for valid Roll measurements (-75° to +75°) avoids the gimbal lock position.

Orientation is a setting that maps each of the three internal axes to one of the labeled axes, each in a positive or negative direction. There are 24 orientation settings, numbered 0 to 23 (0x00 to 0x17) as documented in the chart on page 37. This device ships from the factor in orientation #20.



Default: Orientation #20

With the connector side down:

Elevation = 0°, Range [-90° to +90°]

Roll = 0°, Range [-180° to +180°]

Roll valid when Elevation is -75° to +75°.

When using Speed Aiding, the Internal X axis must be aligned to the forward direction of the vehicle.

ORIENTATION CONFIGURATION CHART

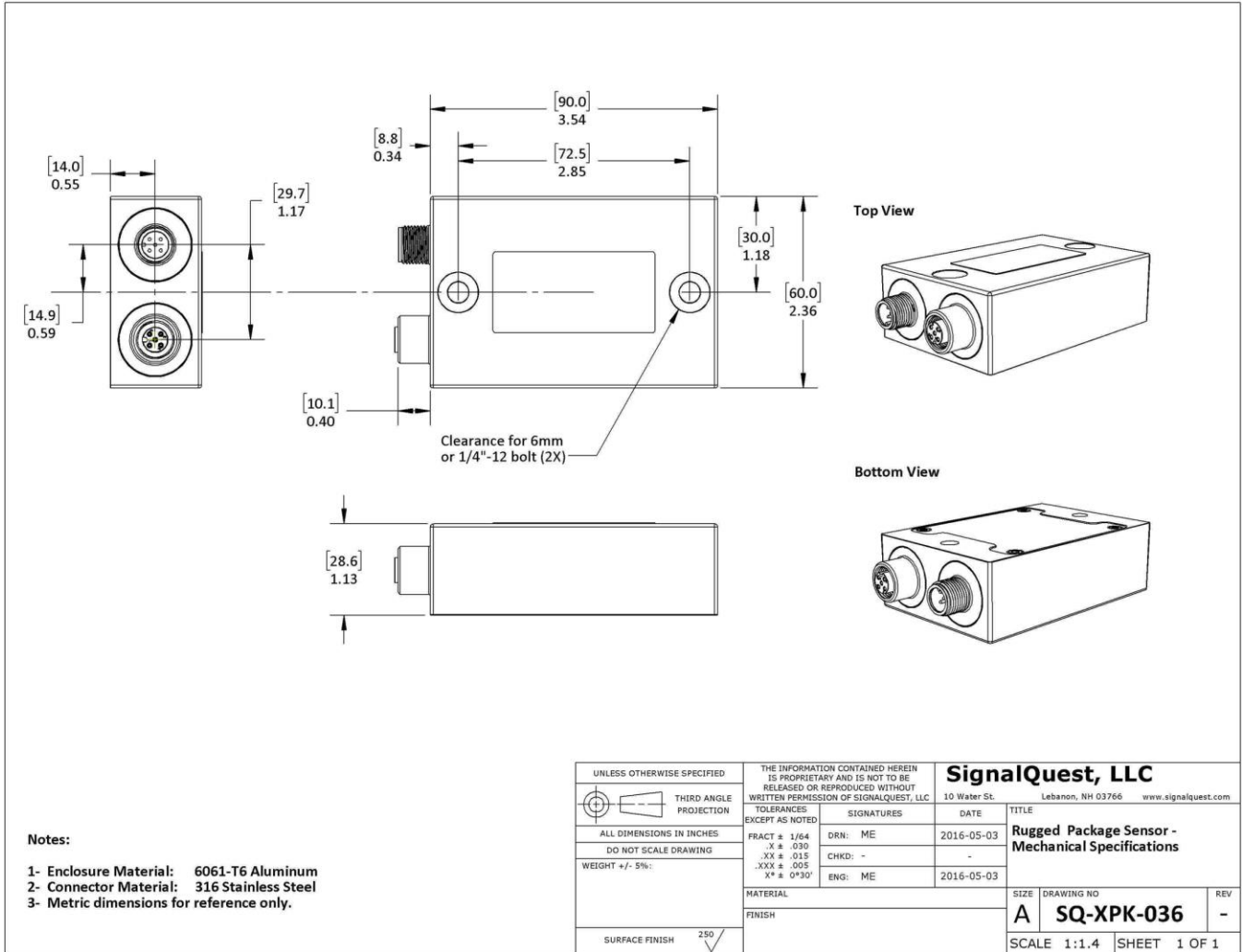
To configure the orientation of the device:

- Choose the Labeled Roll Axis which Roll will rotate about (+X, -X, +Y, -Y, +Z, -Z)
- Choose the Labeled Down Axis which will point down in the given orientation (+X, -X, +Y, -Y, +Z, -Z).
- The Home Position for Elevation and Roll is when the Labeled Roll Axis is horizontal and Labeled Down Axis is pointing towards the Earth.
- Select the corresponding Orientation Number
- Set the Orientation object (0x3000) for the selected orientation number, save it to non-volatile memory, and reboot.

Labeled Down Axis	Labeled Roll Axis	Orientation Number (Decimal)	Orientation Number (Hex)
+Z	+X	0	00
+Z	+Y	1	01
+Z	-X	2	02
+Z	-Y	3	03
-Z	+X	4	04
-Z	-Y	5	05
-Z	-X	6	06
-Z	+Y	7	07
+Y	+X	8	08
+Y	-Z	9	09
+Y	-X	10	0A
+Y	+Z	11	0B
-Y	+X	12	0C
-Y	+Z	13	0D
-Y	-X	14	0E
-Y	-Z	15	0F
+X	-Z	16	10
+X	+Y	17	11
+X	+Z	18	12
+X	-Y	19	13
-X	+Z	20	14
-X	+Y	21	15
-X	-Z	22	16
-X	-Y	23	17

Labeled Roll Axis	Labeled Down Axis	Orientation Number (Decimal)	Orientation Number (Hex)
+X	+Z	0	00
+X	-Z	4	04
+X	+Y	8	08
+X	-Y	12	0C
+Y	+Z	1	01
+Y	-Z	7	07
+Y	+X	17	11
+Y	-X	21	15
+Z	+Y	11	0B
+Z	-Y	13	0D
+Z	+X	18	12
+Z	-X	20	14
-X	+Z	2	02
-X	-Z	6	06
-X	+Y	10	0A
-X	-Y	14	0E
-Y	+Z	3	03
-Y	-Z	5	05
-Y	+X	19	13
-Y	-X	23	17
-Z	+Y	9	09
-Z	-Y	15	0F
-Z	+X	16	10
-Z	-X	22	16

PACKAGING



ENVIRONMENT PROTECTION AND TESTING

Reports available upon request.

LIMITATIONS AND WARNINGS

TESTING

The performance of each system is verified through build-time testing. Each system is tested before and after factory calibration to ensure reliable performance.

SYSTEM INTEGRATION TESTING

Thorough testing should be carried out prior to product release to ensure system integration has not introduced unforeseen problems. The system integrator assumes the ultimate responsibility for the safety of the target application.

NOTICE

Information furnished by SignalQuest, Inc is believed to be accurate and reliable. However, this document may contain ERRORS and OMISSIONS. Accordingly, the design engineer should use this document as a reference rather than a strict design guideline and should perform thorough testing of any product that incorporates this or any other SignalQuest product. No responsibility is assumed by SignalQuest, LLC for the use of this information, or for any infringements of patents or other rights of third parties that may result from its use. Specifications are subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of SignalQuest, LLC. Trademarks and registered trademarks are the property of their respective companies.

REVISION TABLE SQ-GIX-2150

Rev. #	Rev. Date	Revised By:	Test Report	Description	Firmware Version
A	3/26/2020	PBT		Initial release	20200313_C