level developments

# SOLAR-2 : Dual Axis Inclinometer, RS232 or RS485 Output

## Features

- Dual axis measurement, range from ±5 to ±45°
- High resolution and accuracy
- Low temperature drift, with optional temperature compensation to further improve temperature performance
- Single-drop RS232, RS485 or multi-drop RS485 interface with ModBus protocol
- Tough sealed anodised aluminium housing (IP67)
- CE certified and RoHS compliant
- Braided screen 4 core 3m PUR cable
- Low cost relative to performance
- Small size, 75 x 37.5 x 13.5mm and light weight


## Description

The SOLAR-2 inclinometers are range of high performance low cost dual axis tilt sensors for measurement of angle in both the pitch and roll axes. Through a flexible configuration and calibration program we can supply this device with any measurement range from ±5° to ±45°. It can also be supplied compensated for a specific operating temperature range. The housing is a small, low profile Aluminium housing, hermetically sealed to IP67. The cable is a shielded black PUR cable and is suitable for continuous outdoor use. They utilise a very high performance MEMS sensor which exhibits low long term drift compared with many competitive devices. It has an RS232 and RS485 interface option with our standard communication protocol as well as a version with RS485 multi drop ModBus communication protocol. They are CE and RoHS certified, and are manufactured, calibrated and tested in our UK factory to guarantee performance to the stated specification.

## General Specifications

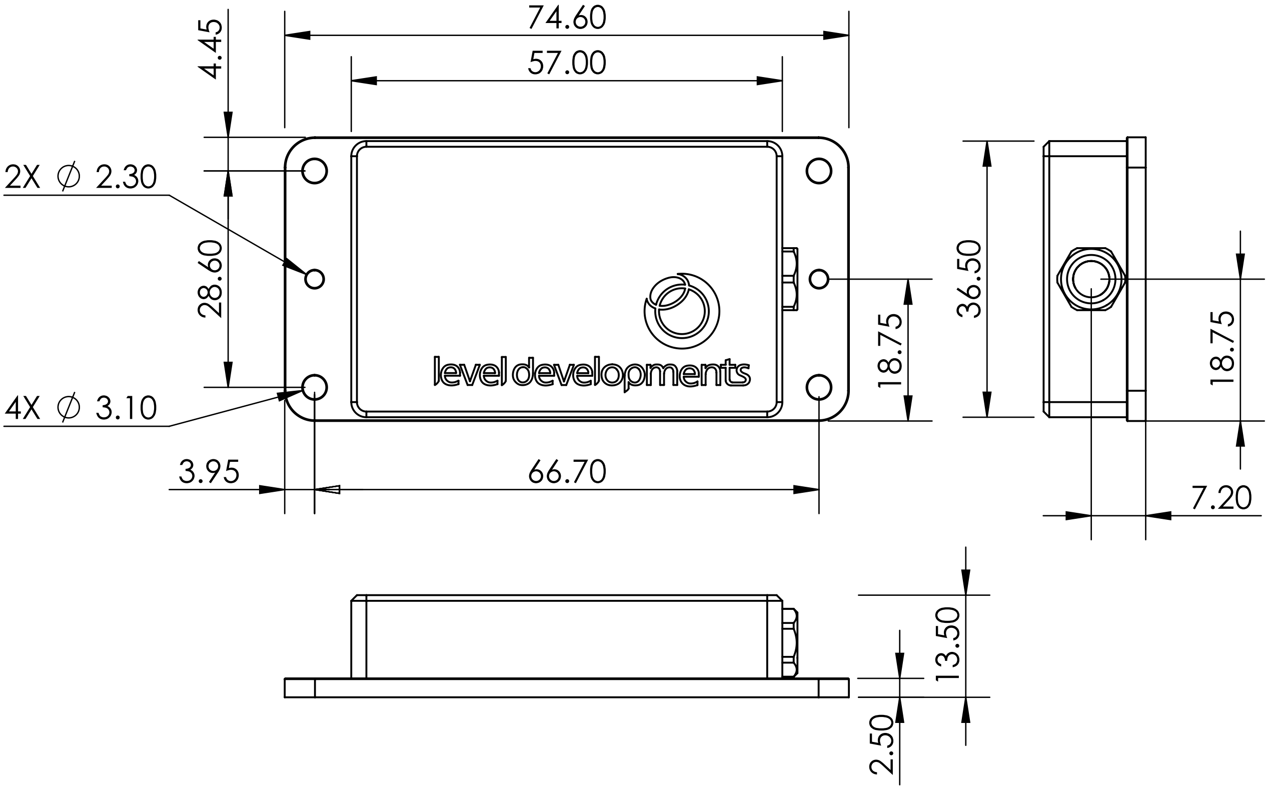
| Parameter | Value | Unit | Notes |
|---|---|---|---|
| **Supply Voltage** | 9-30 | V dc | Supply is filtered, suppressed and regulated internally, however we recommend the use of a low noise supply to prevent noise coupling to the sensor. |
| **Operating Current** | 30mA (@ 9V)<br>20mA (@ 12V)<br>10.5mA (@ 24V) | mA | Supply current depends on supply voltage. |
| **Operating Temperature** | -40 to 85 | °C | Maximum operating temperature range. Units can be calibrated between -20 and 70°C on request. |
| **RS232/485 Output Rate** | 38400 | bps | Bit rate is adjustable between 115.2k, 57.6k, 38.4k, 19.2k, 9.6k, 4.8k and 2.4k via the digital interface |
| **RS232 Data Format** | 38.4, 8,1,N | | 1 start bit, 8 data bits, 1 stop bit, no parity |
| **RS485 & ModBus Format** | 38.4, 8,1,N | | 1 start bit, 8 data bits, 1 stop bit, no parity |
| **Frequency Response** | 1 | Hz | This is the frequency at which the output is 3dB less than the input value, adjustable between 16Hz & 0.125Hz via the RS232/RS485/ModBus control commands |
| **Mechanical shock** | 5000 | G | Shock survival limit for internal sensor 5000G for 0.5ms |
| **Weight** | 45 | g | Not including cable |
| **Cable** | 3 | m | 4 Core braided screen cable with black PUR jacket |
| **Sealing** | IP67 | - | Seal rating applies to housing and cable gland. Gland is not designed for flexible cable installation, as this may compromise seal rating |

## Performance Specifications

| Parameter | SOLAR-05 | SOLAR-15 | SOLAR-30 | SOLAR-45 | Unit |
|---|---|---|---|---|---|
| **Measuring range** | ±5 | ±15 | ±30 | ±45 | ° |
| **Zero Bias Error** | ±0.005 | ±0.010 | ±0.015 | ±0.020 | ° |
| **Accuracy (@20°C)** | ±0.010 | ±0.020 | ±0.030 | ±0.040 | ° |
| **Temperature Errors (without compensation)**<br>Zero Drift<br>Sensitivity Drift | ±0.0015<br>±0.0030 | ±0.0015<br>±0.0030 | ±0.0015<br>±0.0030 | ±0.0015<br>±0.0030 | °/°C<br>%/°C |
| **Temperature Errors (with compensation)**<br>Zero Drift<br>Sensitivity Drift | ±0.0003<br>±0.0006 | ±0.0003<br>±0.0006 | ±0.0003<br>±0.0006 | ±0.0003<br>±0.0006 | °/°C<br>%/°C |
| **Accuracy -10 to 60°C (without compensation)** | ±0.070 | ±0.090 | ±0.120 | ±0.150 | ° |
| **Accuracy -10 to 60°C (with compensation)** | ±0.025 | ±0.030 | ±0.050 | ±0.065 | ° |
| **Long Term Stability** | ±0.007 | ±0.007 | ±0.007 | ±0.007 | ° |
| **Resolution (@1Hz BW)** | 0.001 | 0.001 | 0.001 | 0.001 | ° |

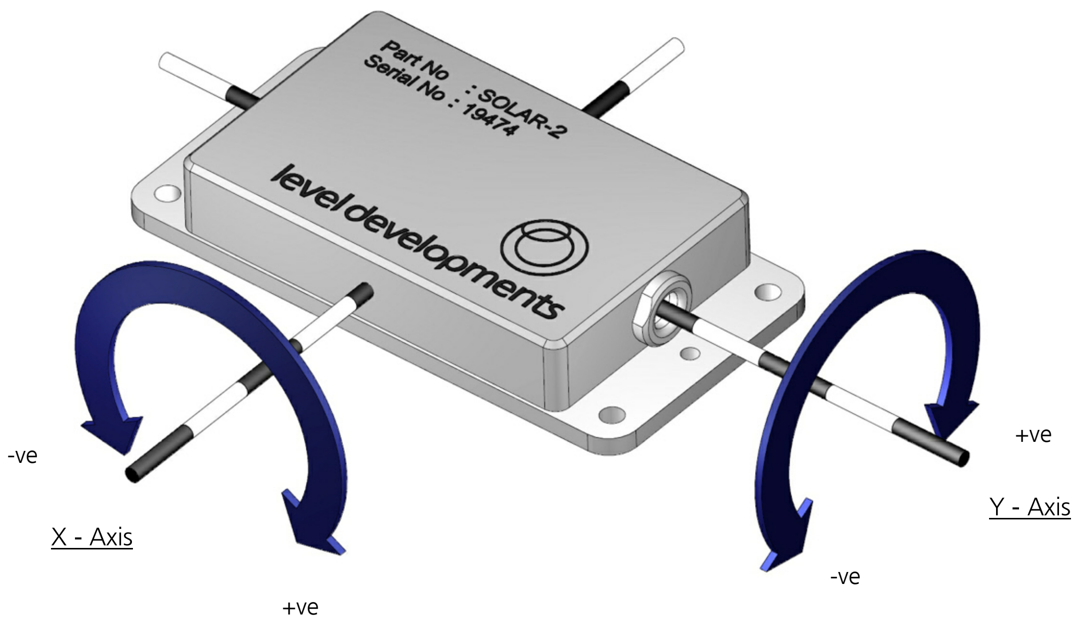
| Parameter | Notes |
|---|---|
| **Measuring range** | Defines the calibrated measurement range. Direction of measurement can be reversed and zero position can be reset anywhere in range. Settings are stored in non volatile memory so are remembered after power down. |
| **Zero Bias Error** | This is the **maximum** angle from the device when it is placed on a perfectly level surface. The zero bias error can be removed from measurement errors either by mechanical adjustment, or as a fixed offset value after installation, or by using the 'setzcur' command to zero the device (see page 8) |
| **Accuracy (@20°C)** | This is the **maximum** error between the measured and displayed value at any point in the measurement range when the device is at room temperature (20°C). This value includes cross axis errors. |
| **Temperature Errors** | These figures are for devices without additional temperature compensation. See part numbering options on page 7 for further details. |
| Zero Drift | If the device is mounted to a level surface in the zero position, this value is the **maximum** drift of the output angle per °C change in temperature. |
| Sensitivity Drift | When the temperature changes there is a change in sensitivity of the sensor's output. The error this causes in the measurement is calculated from the formula:<br>$\mathbf{E_{sd} = SD \times \Delta T \times \theta}$<br>Where:<br>$E_{sd}$ is the change in output (in degrees) due to sensitivity temperature change<br>**SD** is the sensitivity drift specification from the above table (0.003%)<br>**ΔT** is the change is temperature in °C<br>θ is the current angle of the inclinometer axis in question in degrees. |
| **Accuracy -10 to 60°C (without compensation)** | This is the **maximum** error between the measured and displayed value at any point in the measurement range at any temperature over the specified temperature range without individual temperature compensation. |
| **Accuracy -10 to 60°C (with compensation)** | This is the **maximum** error between the measured and displayed value at any point in the measurement range at any temperature over the calibrated temperature range with individual temperature compensation. |
| **Long Term Stability** | Stability depends on environment (temperature, shock, vibration and power supply). This figure is based on being powered continuously in an ideal environment. |
| **Resolution (@1Hz bandwidth)** | Resolution is the smallest measurable change in output. |

## Housing Drawing



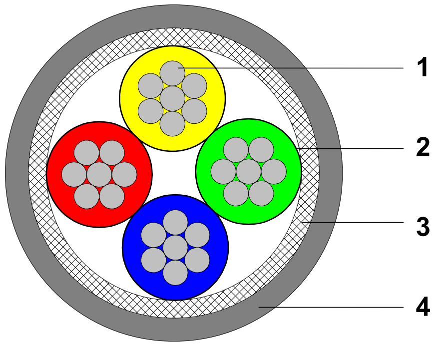

## Axis Direction and Mounting Orientation and Wiring Details

Mounted on a Horizontal Surface



## Cable Details

1. Core wires, tin plated copper, 18x0.1mm strands per conductor (26 AWG).
2. 4 conductors, colours red, blue, yellow and green. PVC core insulation.
3. Braided screen of tin copper wire with minimum 85% coverage.
4. Black PUR Solar jacket. Flame retardant, reduced smoke generation, zero halogen, excellent for use in water and oil, good for use in acids and fuels, radiation tolerance: 10E6 Gy, UV stable, suitable for continuous outdoor use.




| Wire Colour | RS232 Version | RS485 Version |
|---|---|---|
| Red | +ve Supply | +ve Supply |
| Blue | Ground | Ground |
| Yellow | RS232 Txd | RS485 - |
| Green | RS232 Rxd | RS485 + |

| Parameter | Value | Unit | Notes |
|---|---|---|---|
| **Approximate Weight** | 40 | g/m | |
| **Operating Temperature** | -20 to 70 | °C | |
| **Conductor Resistance** | 100 | Ω/Km | Maximum resistance |
| **Insulation Resistance** | 1500 | MΩ/Km | Minimum resistance |
| **Test Voltage** | 1 | KV DC | |
| **Voltage Rating** | 250 | V | |
| **Core Current Rating** | 0.5 | A | At 40°C air temperature |
| **Individual Core Diameter** | 1.1 | mm | |
| **Overall Diameter** | 4.5 | mm | |

## Certification

The products are type approved to in accordance with the following directive(s):

EMC Directive 2004/108/EC

And it has been designed, manufactured and tested to the following specifications:

BS EN61326-1:2006 — Electrical equipment for measurement, control and laboratory use – EMC Requirements

BS EN55011:2007, Group 1 Class B

Certification is available on request.

## Part Numbering

SOLAR - 2 - XX - X - X - XX

Series Prefix

05 - ±5° Full Scale Measurement Range
15 - ±15° Full Scale Measurement Range
30 - ±30° Full Scale Measurement Range
45 - ±45° Full Scale Measurement Range

1 - No additional temperature compensation
2 - Temperature compensation over -10 to 60°C

RS232 - RS232 Interface with LD standard communication protocol
RS485 - RS485 Interface with LD standard communication protocol
RS485M - RS485 Interface with ModBus communication protocol

Customer Specific Options (Optional)

Example:

**SOLAR-2-15-2-RS485M**
SOLAR-2 Series dual axis inclinometer
±15° Full Scale Measurement Range
Temperature compensated over the range -10 to 60°C
RS485 Interface with ModBus communication protocol

## Level Developments Simplified Control Command Set

Data is transmitted and received over RS232 in full duplex mode and for RS485 versions in half duplex mode. The default configuration is with the baud rate set to 38.4kbps, with 8 data bits, 1 stop bit and no parity. All commands are lower case and 7 bytes long. The time between each character of the command must be less than 100ms otherwise the device will discard the command. The settings are all stored in non volatile memory.

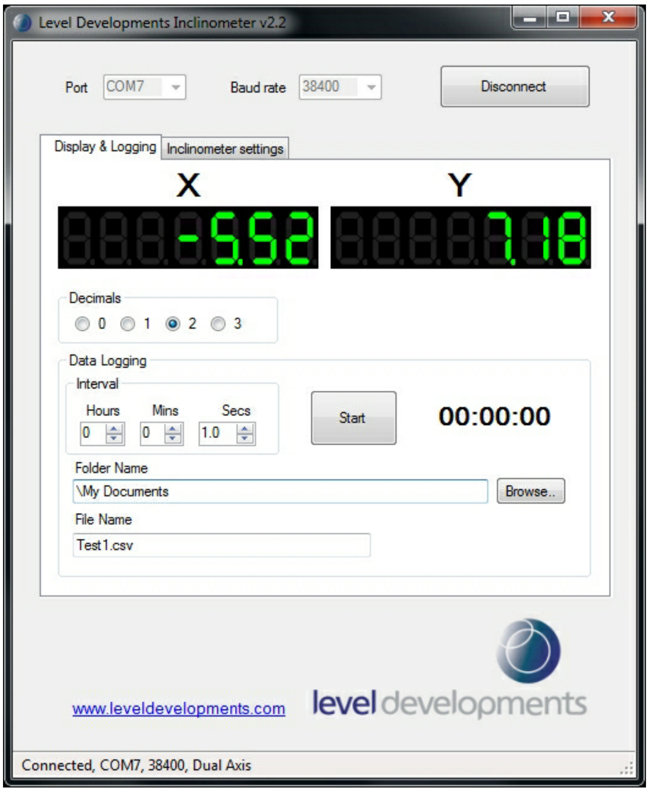
| Command | Description | Response Length | Response |
|---|---|---|---|
| `get---x` | Returns the X axis angle as either:<br>- An INT32 value equal to the angle x 1000<br>- A fixed length ASCII string terminated with a carriage return depending on the setting of commands 'setoasc' or 'setoint'<br>Shipping default is INT32. | 4 bytes<br>9 bytes | 0x XX XX XX XX<br>+025.430<CR> |
| `get---y` | Returns the Y axis angle as either:<br>- An INT32 value equal to the angle x 1000<br>- A fixed length ASCII string terminated with a carriage return depending on the setting of commands 'setoasc' or 'setoint'<br>Shipping default is INT32. | 4 bytes<br>9 bytes | 0x YY YY YY YY<br>+025.430<CR> |
| `get-x&y` | Returns the X and Y axis angle (X is transmitted first) as either:<br>- A pair of INT32 value equal to the angle x 1000<br>- A fixed length comma separated ASCII string terminated with <CR> depending on the setting of commands 'setoasc' or 'setoint'<br>Shipping default is INT32. | 8 bytes<br><br>18 bytes | 0x XX XX XX XX<br>YY YY YY YY<br>±xxx.xxx,±yyy.yyy<CR> |
| `gettemp` | Returns the temperature of the sensor as either:<br>- An INT16 value equal to the temperature x 100<br>- A fixed length ASCII string terminated with a carriage return depending on the setting of commands 'setoasc' or 'setoint'<br>Shipping default is INT32. | 2 bytes<br>6 bytes | 0x XX XX<br>±tt.t<CR> |
| `str9999` | Set continuous output transmission rate in milliseconds (25-9999ms)<br>- str0100 - 100ms (0.1s) between transmissions | 2 bytes | OK |
| `setcasc` | Sets the output to transmit the X and Y angle continuously in ASCII format at the rate defined by strXXXX. | 18 bytes | ±xxx.xxx,±yyy.yyy<CR> |
| `stpcasc` | Stops the continuous transmission of ASCII data | 2 bytes | OK |
| `get-flt` | Returns the value of the current filter time constant in ms as an INT16 | 2 bytes | 0x XX XX |
| `setdir1`<br>`setdir2`<br>`setdir3`<br>`setdir4` | Sets the X axis measurement direction to positive clockwise<br>Sets the X axis measurement direction to negative clockwise<br>Sets the Y axis measurement direction to positive clockwise<br>Sets the Y axis measurement direction to negative clockwise | 2 bytes | OK |
| `setzcur` | Tare function to set the current position to zero | 2 bytes | OK |
| `setzfac` | Cancels tare function and resets zero to factory setting | 2 bytes | OK |
| `setoasc` | Sets the output to ASCII format | 2 bytes | OK |
| `setoint` | Sets the output to Integer format | 2 bytes | OK |
| `setflt1`<br>`setflt2`<br>`setflt3`<br>`setflt4`<br>`setflt5`<br>`setflt6`<br>`setflt7`<br>`setflt8` | Sets the digital filter frequency response to 0.125Hz<br>Sets the digital filter frequency response to 0.25Hz<br>Sets the digital filter frequency response to 0.5Hz<br>Sets the digital filter frequency response to 1Hz<br>Sets the digital filter frequency response to 2Hz<br>Sets the digital filter frequency response to 4Hz<br>Sets the digital filter frequency response to 8Hz<br>Sets the digital filter frequency response to 16Hz | 2 bytes | OK |
| `set-br1`<br>`set-br2`<br>`set-br3`<br>`set-br4`<br>`set-br5`<br>`set-br6`<br>`set-br7` | Sets the BAUD rate to 2400bps<br>Sets the BAUD rate to 4800bps<br>Sets the BAUD rate to 9600bps<br>Sets the BAUD rate to 19200bps<br>Sets the BAUD rate to 38400bps<br>Sets the BAUD rate to 57600bps<br>Sets the BAUD rate to 115200bps | 2 bytes | OK |

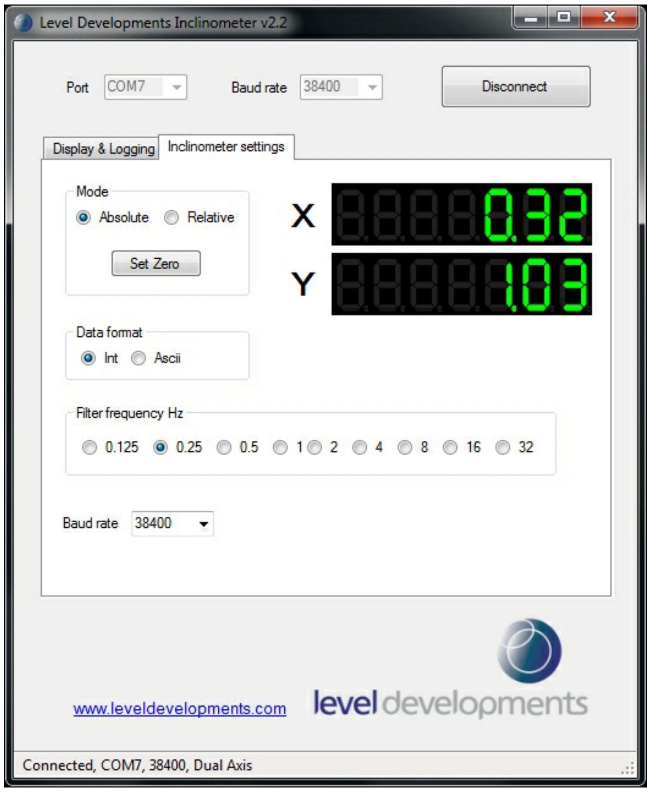
## Software

A free Windows based application for reading angle, logging and device configuration is available from our website. It requires Windows XP SP3, Windows 7 or Windows 8, and works with 32 and 64 bit systems. It also requires the .net framework V3.5 or higher, and will prompt you to download and install this from Microsoft if it is not already installed on your system. A COM port is also required, and can either be a built in COM port, or a USB to Serial COM port.

The basic features are shown below:

- Automatic or manual configuration of COM port parameters
- Compatible with single or dual axis sensors
- Adjustable number of decimal places on displays
- Logging of data at specified intervals into CSV file
- Setting device to absolute or relative measurement mode
- Switching the data transfer protocol between Integer and ASCII
- Changing the frequency response of the sensor
- Changing the Baud rate of the sensor




We can also offer custom software development services, please contact us for further information.

**This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.**

## ModBus Control Command Set

Data is transmitted and received over RS485 in half duplex mode using the ModBus RTU protocol. The following section provides some basic information about the serial communication between the host PC or PLC and the SOLAR-2. The full ModBus specification can be obtained from http://www.modbus.org. ModBus is a command/response protocol over a serial bus.

The default ModBus serial parameters are: 38400 baud, 1 start bit, 8 data bits, no parity and 1 stop bit. The 8 data bits are sent LSB first. The baud rate can be changed to 115200, 57600, 38400, 19200, 9600, 4800 or 2400 by sending the appropriate command.

The byte order for all 16-bit values is Big Endian (most significant byte first).

Read and write access to the SOLAR-2 is done using ModBus Function Code 3 (read holding registers) and ModBus Function Code 6 (write single register) commands. These two function codes provide the basic functionality needed by most users of the SOLAR-2. A user defined ModBus function code 110 is provided for less commonly used, off-line functions such as setting serial port parameters and changing the device address.

ModBus device address must be in the range 1 to 247. All devices are shipped with a default address of 100 (decimal). Address 0 is the ModBus broadcast address. With this address all devices will perform the action of the function code. The maximum number of these devices that can be connected on a single network is 128.

All ModBus commands and responses have a 16-bit CRC for error detection. ModBus RTU data is in binary format rather than ASCII, so it cannot be viewed properly on a text terminal.

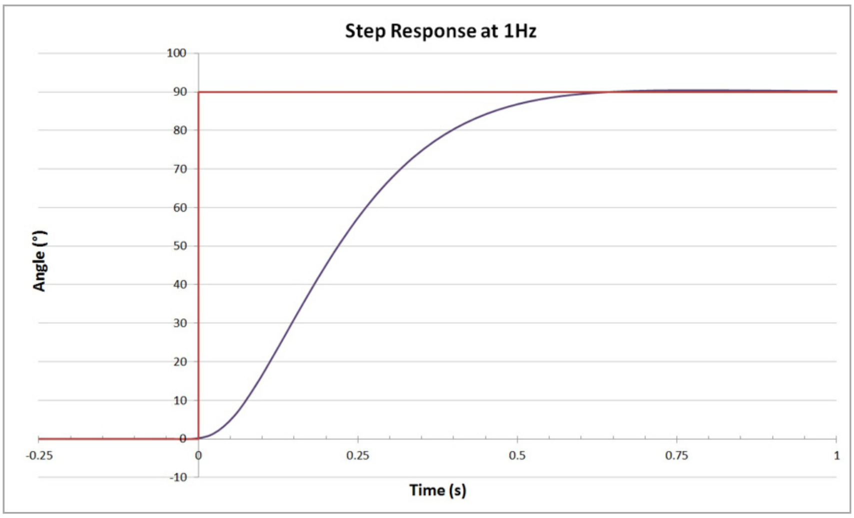
Below is a list of the register locations for reading and writing:

## ModBus Registers

| Parameter | Address | ModBus Register Address | Description | Read/Write |
|---|---|---|---|---|
| X Axis Angle | 0x00 | 40,001 | Address 0x00 returns the upper 16 bits of the sensor X axis angle. This combines with address 0x01 to form a 32 bit signed integer value equal to the measured angle x 1000. | Read Only |
| | 0x01 | 40,002 | | |
| Y Axis Angle | 0x02 | 40,003 | Address 0x02 returns the upper 16 bits of the sensor Y axis angle. This combines with address 0x03 to form a 32 bit signed integer value equal to the measured angle x 1000. | Read Only |
| | 0x03 | 40,004 | | |
| Sensor Temperature | 0x06 | 40,007 | Returns a 16 bit signed integer value equal to the temperature of the sensor in degrees Celsius x 100 | Read Only |
| Sensor Filter Index | 0x09 | 40,010 | Returns a 16 bit integer value between 1 and 7 which relates to a table of filter responses from 0.125 to 16Hz | Read / Write |
| Tare Function | 0x14 | 40,021 | When set to '1' the device is zeroed at the current position (relative mode). When set to '0' the device is returned to absolute measurement mode (tare cancelled) | Read / Write |

## Frequency Response Filter Indexes

The frequency response of the sensor can be changed to any of the response times shown in the table. The filter is a 2nd order Besel low pass filter implemented in a IIR algorithm.




| Filter Index | Freq. Response (Hz) | Damping Time (ms) |
|---|---|---|
| 1 | 0.125 | 8000 |
| 2 | 0.25 | 4000 |
| 3 | 0.5 | 2000 |
| 4 | 1 | 1000 |
| 5 | 2 | 500 |
| 6 | 4 | 250 |
| 7 | 8 | 125 |
| 8 | 16 | 62.5 |

## Reading a Holding Register

The data from the device is stored in holding registers as detailed on page 4. Function code 0x03 is used to read these registers. Below is the command and response message format, including the error response in the even there is an error.

| | Byte Data | No Of Bytes | Description |
|---|---|---|---|
| Command | 0x64 | 1 | Slave address 100 |
| | 0x03 | 1 | Function code for read register |
| | 0x0000 | 2 | Starting register (0x0000 is X axis angle) |
| | 0x0002 | 2 | Number of registers to read |
| | 0xCDFE | 2 | CRC-16 of all bytes |

| | | | |
|---|---|---|---|
| Response | 0x64 | 1 | Slave address 100 |
| | 0x03 | 1 | Function code for read register |
| | 0x04 | 1 | Byte count (2 x number of regsiters) |
| | 0x0000 | 2 | First and second register data : 0x0000A69C = 42652 (decimal) |
| | 0xA69C | 2 | |
| | 0xB4FC | 2 | CRC-16 of all bytes |

| | | | |
|---|---|---|---|
| Error Response | 0x64 | 1 | Slave address 100 |
| | 0x83 | 1 | ModBus error function code |
| | 0x01 | 1 | Exception Code (0x01 invalid function code, 0x02 invalid register address) |
| | 0x90EF | 2 | CRC-16 of all bytes |

## Writing to a Holding Register

Data can be written to some registers, such as the registers that store the filter indexes for each axis frequency response. Function code 0x06 is used to write these registers as detailed below.

| | Byte Data | No Of Bytes | Description |
|---|---|---|---|
| Command | 0x64 | 1 | Slave address 100 |
| | 0x06 | 1 | Function code for write register |
| | 0x0009 | 2 | Register to write (0x0009 is axis filter) |
| | 0x0003 | 2 | Data to write (16 bit). 0x0003 = 0.5Hz |
| | 0x103C | 2 | CRC-16 of all bytes |

| | | | |
|---|---|---|---|
| Response (same as command) | 0x64 | 1 | Slave address 100 |
| | 0x06 | 1 | Function code for write register |
| | 0x0009 | 2 | Register to write (0x0009 is axis filter) |
| | 0x0003 | 2 | Data to write (16 bit). 0x0003 = 0.5Hz |
| | 0x103C | 2 | CRC-16 of all bytes |

| | | | |
|---|---|---|---|
| Error Response | 0x64 | 1 | Slave address 100 |
| | 0x83 | 1 | ModBus error function code |
| | 0x01 | 1 | Exception Code (0x01 invalid function code, 0x02 invalid register address, 0x03 parameter out of range) |
| | 0x90EF | 2 | CRC-16 of all bytes |

## Changing the BAUD Rate

The BAUD rate of the device can be changed using the special function code 0x6E and special command code 0x8F.

| | Byte Data | No Of Bytes | Description |
|---|---|---|---|
| Command | 0x64 | 1 | Slave address 100 |
| | 0x6E | 1 | Function code - 0x6E |
| | 0x8F | 1 | LD command - 0x8F = set baud |
| | 0x03 | 1 | 1 = 2400 |
| | | | 2 = 4800 |
| | | | 3 = 9600 |
| | | | 4 = 19200 |
| | | | 5 = 38400 |
| | | | 6 = 57600 |
| | | | 7 = 115200 |
| | 0x5AF8 | 2 | CRC-16 of all bytes |

## Changing the Device Address

The Address of the device can be changed using the special function code 0x6E and special command code 0x91. The device will reply with the original address in the response, and will change internally after the response has been sent.

| | Byte Data | No Of Bytes | Description |
|---|---|---|---|
| Command | 0x64 | 1 | Slave address 100 |
| | 0x6E | 1 | Function code - 0x6e |
| | 0x91 | 1 | LD command - 0x91 = change address |
| | 0x01 | 1 | New Address = 1 |
| | 0xD299 | 2 | CRC-16 of all bytes |
| Response | 0x64 | 1 | Slave address 100 |
| | 0x6E | 1 | Function code - 0x6e |
| | 0x91 | 1 | LD command - 0x91 = change address |
| | 0x00 | 1 | 0 = success 1 = failed |
| | 0x1359 | 2 | CRC-16 of all bytes |

## Examples of Reading Angle

**Example 1: Read the angle from the sensor X axis with address 100 (0x64):**

Command

```
address (0x64 = 100 decimal)
|  function code
|  |  starting reg. to read (0x0000)
|  |  |     number of reg. to read (0x0002)
|  |  |     |
|  |  |     |     CRC-16
|  |  |     |     |
64 03 00 00 00 02 cd fe
```

Response (positive angle)

```
address (0x64 = 100 decimal)
|  function code
|  |  byte count
|  |  |  angle (0x0000a69c = 42652 decimal (42.652 degrees)
|  |  |  |           CRC-16
|  |  |  |           |
64 03 04 00 00 a6 9c b4 fc
```

Response (negative angle)

```
address (0x64 = 100 decimal)
|  function code
|  |  byte count
|  |  |  angle (0xfffda7d7 = -153641 decimal (-153.641 degrees)
|  |  |  |           CRC-16
|  |  |  |           |
64 03 04 ff fd a7 d7 54 bf
```

**Example 2: Read the angle from the Y axis with address 100 (0x64):**

Command

```
address (0x64 = 100 decimal)
|  function code
|  |  starting reg. to read (0x0002)
|  |  |     number of reg. to read (0x0002)
|  |  |     |
|  |  |     |     CRC-16
|  |  |     |     |
64 03 00 02 00 02 6c 3e
```

Response (positive angle)

```
address (0x64 = 100 decimal)
|  function code
|  |  byte count
|  |  |  angle (0x00005ba3 = 23459 decimal (23.459 degrees)
|  |  |  |           CRC-16
|  |  |  |           |
64 03 04 00 00 5b a3 b4 7c
```

Response (negative angle)

```
address (0x64 = 100 decimal)
|  function code
|  |  byte count
|  |  |  angle (0xffffa54d = -23219 decimal (-23.219 degrees)
|  |  |  |           CRC-16
|  |  |  |           |
64 03 04 ff ff a5 4d 74 74
```

**Example 3: Change the frequency response to 0.5Hz:**

Command

```
address (0x64 = 100 decimal)
|  function code
|  |  register to write to (0x0009)
|  |  |     data to write (0x0003 = 0.5Hz)
|  |  |     |
|  |  |     |     CRC-16
|  |  |     |     |
64 06 00 09 00 03 10 3c
```

Response

```
address (0x64 = 100 decimal)
|  function code
|  |  register written to (0x0009)
|  |  |     data written (0x0003 = 0.5Hz)
|  |  |     |
|  |  |     |     CRC-16
|  |  |     |     |
64 06 00 09 00 03 10 3c
```

Example 4: Setting the tare function (current position to zero):

**Command**

```
address (0x64 = 100 decimal)
|  function code
|  |  register to write to (0x0014)
|  |  |     data to write (0x0001 = set tare on)
|  |  |     |
|  |  |     |     CRC-16
|  |  |     |     |
64 06 00 14 00 01 01 fb
```

**Response**

```
address (0x64 = 100 decimal)
|  function code
|  |  register written to (0x0014)
|  |  |     data written (0x0001 = set tare on)
|  |  |     |
|  |  |     |     CRC-16
|  |  |     |     |
64 06 00 14 00 01 01 fb
```

Example 5: Change the device address from 100 to 1:

**Command**

```
address (0x64 = 100 decimal)
|  special function code
|  |  LD command for change address
|  |  |  new address (0x01)
|  |  |  |
|  |  |  |  CRC-16
|  |  |  |  |
64 6e 91 01 d2 99
```

**Response**

```
address (0x64 = 100 decimal)
|  special function code
|  |  LD command for change address
|  |  |  Success/Fail (0x00 = success)
|  |  |  |
|  |  |  |  CRC-16
|  |  |  |  |
64 6e 91 00 13 59
```