

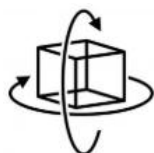
## AHRS IMU Analog Voltage Sensor | SINVT

*The Robust Acceleration, Angular velocity & Angle Detector*

---

*The SINVT is a IMU sensor device, detecting acceleration, angular velocity as well as angle. The robust housing and the small outline makes it perfectly suitable for industrial applications such as condition monitoring and predictive maintenance. Configuring the device enables the customer to address a broad variety of application by interpreting the sensor data by smart algorithms and Kalman filtering.*

### BUILT-IN SENSORS



Accelerometer



Gyroscope



## Tutorial Link

[Google Drive](#)

**Link to instructions DEMO:**

[WITMOTION Youtube Channel](#)

[SINVT Playlist](#)

If you have technical problems or cannot find the information that you need in the provided documents, please contact our support team. Our engineering team is committed to providing the required support necessary to ensure that you are successful with the operation of our AHRS sensors.

## Contact

[Technical Support Contact Info](#)

## Application

- AGV Truck
- Platform Stability
- Auto Safety System
- 3D Virtual Reality
- Industrial Control
- Robot
- Car Navigation
- UAV
- Truck-mounted Satellite Antenna Equipment

# Contents

Tutorial Link.....	- 2 -
Contact.....	- 2 -
Application.....	- 2 -
Contents.....	- 3 -
1 Overview.....	- 4 -
2 Features.....	- 5 -
3 Specification.....	- 6 -
3.1 Parameter.....	- 6 -
3.2 Size.....	- 7 -
3.3 Axial Direction.....	- 8 -
4 PIN Definition.....	- 9 -
5 Communication Protocol.....	- 10 -
5.1 Output Data Format.....	- 10 -
5.1.1 Time Output.....	- 10 -
5.1.2 Acceleration Output.....	- 11 -
5.1.3 Angular Velocity Output.....	- 11 -
5.1.4 Angle Output.....	- 12 -
5.1.5 Quaternion.....	- 13 -
5.2 Config Commands.....	- 14 -
5.2.1 Register Address.....	- 14 -
5.2.2 Save Configuration.....	- 15 -
5.2.3 Calibrate.....	- 15 -
5.2.4 Installation Direction.....	- 15 -
5.2.5 Sleep/ Wake up.....	- 15 -
5.2.6 Algorithm Transition.....	- 16 -
5.2.7 Gyroscope Automatic Calibration.....	- 16 -
5.2.8 Return Content.....	- 17 -
5.2.9 Return Rate.....	- 18 -
5.2.10 Baud Rate.....	- 18 -
5.2.11 Set X Axis Acceleration Bias.....	- 19 -
5.2.12 Set Y Axis Acceleration Bias.....	- 19 -
5.2.13 Set Z Axis Acceleration Bias.....	- 19 -
5.2.14 Set X Axis Angular Velocity Bias.....	- 20 -
5.2.15 Set Y Axis Angular Velocity Bias.....	- 20 -
5.2.16 Set Z Axis Angular Velocity Bias.....	- 20 -



# 1 Overview

SINVT's scientific name is AHRS IMU Analog sensor. A sensor measures 2-axis angle, 3-axis angular velocity and acceleration. Its strength lies in the algorithm which can calculate 2-axis angle accurately.

SINVT is employed where the highest measurement accuracy is required. SINVT offers several advantages over competing sensor:

- Heated for best data availability: new WITMOTION patented zero-bias automatic detection calibration algorithm outperforms traditional accelerometer sensor
- High precision (X Y Z axis) Acceleration + (X Y Z axis) Angular Velocity +(X Y axis) Roll Pitch Angle output
- The analog voltage output (0-5v) based on angle output of X Y axis (-90° , +90°)
- Low cost of ownership: remote diagnostics and lifetime technical support by WITMOTION service team
- Developed tutorial: providing manual, datasheet, Demo video, PC software, communication protocol
- WITMOTION sensors have been praised by thousands of engineers as a recommended attitude measurement solution



## 2 Features

- The default baud rate of this device is 9600 and could be changed.
- The interface of this product only leads to a serial port
- The module consists of a high precision gyroscope, accelerometer sensor. The product can solve the current real-time motion posture of the module quickly by using the high-performance microprocessor, advanced dynamic solutions and Kalman filter algorithm.
- The advanced digital filtering technology of this product can effectively reduce the measurement noise and improve the measurement accuracy.
- Maximum 100Hz data output rate. Output content can be arbitrarily selected, the output speed 0.2HZ~ 100HZ adjustable.

## 3 Specification

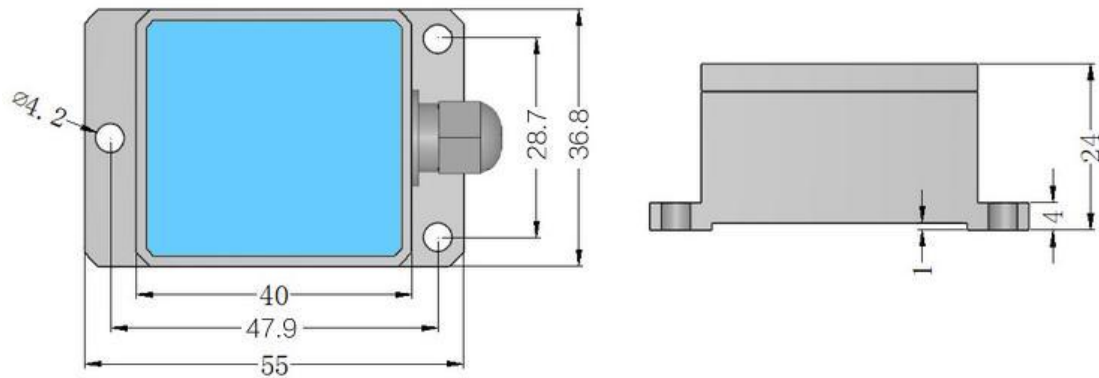
### 3.1 Parameter

Parameter	Specification
➤ Working Voltage	RS232: 12-36V
➤ Current	<40mA
➤ Size	55mm x 36.8mm X 24mm
➤ Data	Angle: X Y , 2-axis Acceleration: X Y Z, 3-axis Angular Velocity: X Y Z, 3-axis Time, Quaternion
➤ Output frequency	0.2Hz--100Hz(above 10Hz is recommended)
➤ Interface	Serial RS232 level, Baud rate--9600 (default, could not be changed )

#### Measurement Range & Accuracy

Sensor	Measurement Range	Accuracy/ Remark
➤ Accelerometer	X, Y, Z, 3-axis ±16g	Accuracy: 0.01g Resolution: 16bit Stability: 0.005g
➤ Gyroscope	X, Y, Z, 3-axis -±2000°/s	Resolution: 16bit Stability: 0.05°/s
➤ Angle/ Inclinator	X, Y, , 2-axis X, Y ±90° (Y-axis 90° is singular point)	Accuracy:X, Y-axis: 0.05°

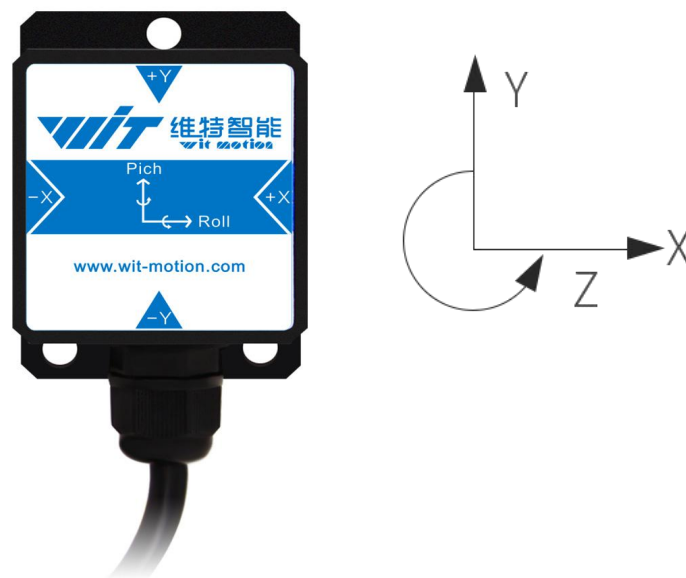
## 3.2 Size



Parameter	Specification	Tolerance	Comment
Length	55	$\pm 0.1$	Unit: millimeter.
Width	36.8	$\pm 0.1$	
Height	24	$\pm 0.1$	
Weight	100	$\pm 1$	Unit: gram

### 3.3 Axial Direction

The coordinate system used for attitude angle settlement is the northeast sky coordinate system. Place the module in the positive direction, as shown in the figure below, direction right is the X-axis, the direction forward is the Y-axis, and direction upward is the Z-axis. Euler angle represents the rotation order of the coordinate system when the attitude is defined as Z-Y-X, that is, first turn around the Z-axis, then turn around the Y-axis, and then turn around the X-axis.





## 4 PIN Definition



PIN	Color	Function
➤ VCC	RED	Input Supply RS232 power with 12-36V
➤ 232RX	GREEN	Serial data input RX Connected with 232TX
➤ 232TX	YELLOW	Serial data output TX Connected with 232RX
➤ GND	BLACK	Ground GND
➤ OUT X Voltage	White	X-axis analog voltage output
➤ OUT Y Voltage	Gray	Y-axis analog voltage output
➤ Relative Attitude	Purple	Set Angle to 0

## 5 Communication Protocol

Level: RS232 level

Baud rate: 4800, 9600 (default), 19200, 38400, 57600, 115200, 230400, 460800, 921600, stop bit and parity

### 5.1 Output Data Format

#### 5.1.1 Time Output

0x55	0x50	YY	MM	DD	hh	mm	ss	msL	msH	SUM
------	------	----	----	----	----	----	----	-----	-----	-----

YY: Year, 20YY Year

MM: Month

DD: Day

hh: hour

mm: minute

ss: Second

ms: Millisecond

Millisecond calculate formula:

$ms = ((msH \ll 8) | msL)$

$Sum = 0x55 + 0x51 + YY + MM + DD + hh + mm + ss + ms + TL$

## 5.1.2 Acceleration Output

0x55	0x51	AxL	AxH	AyL	AyH	AzL	AzH	TL	TH	SUM
------	------	-----	-----	-----	-----	-----	-----	----	----	-----

Calculate formula:

$$a_x = ((AxH < 8) | AxL) / 32768 * 16g \text{ (g is Gravity acceleration, } 9.8m/s^2)$$

$$a_y = ((AyH < 8) | AyL) / 32768 * 16g \text{ (g is Gravity acceleration, } 9.8m/s^2)$$

$$a_z = ((AzH < 8) | AzL) / 32768 * 16g \text{ (g is Gravity acceleration, } 9.8m/s^2)$$

Temperature calculated formular:

$$T = ((TH < 8) | TL) / 100 \text{ } ^\circ C$$

Checksum:

$$Sum = 0x55 + 0x51 + AxH + AxL + AyH + AyL + AzH + AzL + TH + TL$$

Note:

1. The data is sent in hexadecimal, not ASCII code.

Each data is transmitted in turn of low byte and high byte, and the two are combined into a signed short type data.

For example, X-axis acceleration data Ax, where AxL is low byte and AxH is high byte. The conversion method is as follows:

Assuming that Data is actual data, DataH is its high byte, and DataL is its low byte, then: Data = (short) (DataH < 8 | DataL).

It must be noted that DataH needs to be coerced into a signed short data and then shifted, and the data type of Data is also a signed short type, so that it can represent negative numbers.

## 5.1.3 Angular Velocity Output

0x55	0x52	wxL	wxH	wyL	wyH	wzL	wzH	TL	TH	SUM
------	------	-----	-----	-----	-----	-----	-----	----	----	-----

Calculated formular:

$$w_x = ((wxH < 8) | wxL) / 32768 * 2000(^{\circ}/s)$$

$$w_y = ((wyH < 8) | wyL) / 32768 * 2000(^{\circ}/s)$$

$$w_z = ((wzH < 8) | wzL) / 32768 * 2000(^{\circ}/s)$$

Temperature calculated formular:

$$T = ((TH < 8) | TL) / 100 \text{ } ^\circ C$$

Checksum:

$$Sum = 0x55 + 0x52 + wxH + wxL + wyH + wyL + wzH + wzL + TH + TL$$

## 5.1.4 Angle Output

0x55	0x53	RollL	RollH	PitchL	PitchH	YawL	YawH	VL	VH	SUM
------	------	-------	-------	--------	--------	------	------	----	----	-----

Calculated formular:

Roll(X axis) $Roll = ((RollH < 8) | RollL) / 32768 * 180(^{\circ})$

Pitch(Y axis) $Pitch = ((PitchH < 8) | PitchL) / 32768 * 180(^{\circ})$

Yaw(Z axis) $Yaw = ((YawH < 8) | YawL) / 32768 * 180(^{\circ})$

Version calculated formula:

Version $= (VH < 8) | VL$

Checksum:

Sum $= 0x55 + 0x53 + RollH + RollL + PitchH + PitchL + YawH + YawL + VH + VL$

Note:

1. The coordinate system used for attitude angle settlement is the northeast sky coordinate system. Place the module in the positive direction, as the figure shown in Chapter 3.3, direction right is the X-axis, the direction forward is the Y-axis, and direction upward is the Z-axis. Euler angle represents the rotation order of the coordinate system when the attitude is defined as Z-Y-X, that is, first turn around the Z-axis, then turn around the Y-axis, and then turn around the X-axis.
2. Although the range of the roll angle is  $\pm 180$  degrees, in fact, since the coordinate rotation sequence is Z-Y-X, when expressing the attitude, the range of the pitch angle (Y-axis) is only  $\pm 90$  degrees, and it will change to less than 90 after exceeding 90 degrees Degrees while making the X-axis angle greater than 180 degrees. For detailed principles, please Google Euler angle and posture-related information.
3. Since the three axes are coupled, they will show independent changes only at small angles, and the attitude angles will change at large angles. For example, when the Y-axis is close to 90 degrees, even if the attitude only rotates around the Y-axis, the angle of the axis will also change greatly, which is an inherent problem with Euler angles indicating attitude.

## 5.1.5 Quaternion

0x55	0x59	Q0L	Q0H	Q1L	Q1H	Q2L	Q2H	Q3L	Q3H	SUM
------	------	-----	-----	-----	-----	-----	-----	-----	-----	-----

Calculated formular:

$$Q0 = ((Q0H < 8) | Q0L) / 32768$$

$$Q1 = ((Q1H < 8) | Q1L) / 32768$$

$$Q2 = ((Q2H < 8) | Q2L) / 32768$$

$$Q3 = ((Q3H < 8) | Q3L) / 32768$$

Checksum:

$$\text{Sum} = 0x55 + 0x59 + Q0L + Q0H + Q1L + Q1H + Q2L + Q2H + Q3L + Q3H$$

## 5.2 Config Commands

Reminder:

1. Data format

0xFF	0xAA	Address	DataL	DataH
------	------	---------	-------	-------

### 5.2.1 Register Address

Address	Symbol	Meaning
0x00	SAVE	Save
0x01	CALSW	Calibration
0x02	RSW	Return data content
0x03	RATE	Return data Speed
0x04	BAUD	Baud rate
0x05	AXOFFSET	X axis Acceleration bias
0x06	AYOFFSET	Y axis Acceleration bias
0x07	AZOFFSET	Z axis Acceleration bias
0x08	GXOFFSET	X axis angular velocity bias
0x09	GYOFFSET	Y axis angular velocity bias
0x0a	GZOFFSET	Z axis angular velocity bias
0x30	MMYY	Month , Year
0x31	HHDD	Hour , Day
0x32	SSMM	Second , Minute
0x33	MS	Millisecond
0x34	AX	X axis Acceleration
0x35	AY	Y axis Acceleration
0x36	AZ	Z axis Acceleration
0x37	GX	X axis angular velocity
0x38	GY	Y axis angular velocity
0x39	GZ	Z axis angular velocity
0x3d	Roll	X axis Angle
0x3e	Pitch	Y axis Angle
0x40	TEMP	Temperature
0x51	Q0	Quaternion Q0
0x52	Q1	Quaternion Q1
0x53	Q2	Quaternion Q2
0x54	Q3	Quaternion Q3

## 5.2.2 Save Configuration

0xFF	0xAA	0x00	SAVE	0x00
------	------	------	------	------

SAVE: Save

0: Save current configuration

1: set to default setting

## 5.2.3 Calibrate

0xFF	0xAA	0x01	CALSW	0x00
------	------	------	-------	------

CALSW: Set calibration mode

0: Exit calibration mode

1: Enter Gyroscope and Accelerometer calibration mode

## 5.2.4 Installation Direction

0xFF	0xAA	0x23	DIRECTION	0x00
------	------	------	-----------	------

DIRECTION: set installation direction

0: set to horizontal installation

1: set to vertical installation

## 5.2.5 Sleep/ Wake up

0xFF	0xAA	0x22	0x01	0x00
------	------	------	------	------

After sending the command, the module enters the sleep (standby) state, and once again, the module enters the working state from the standby state.

## 5.2.6 Algorithm Transition

0xFF	0xAA	0x24	ALG	0x00
------	------	------	-----	------

ALG: 6-axis algorithm transition  
1: switch to 6-axis algorithm

## 5.2.7 Gyroscope Automatic Calibration

0xFF	0xAA	0x63	GYRO	0x00
------	------	------	------	------

GYRO: gyroscope automatic calibration  
0: set to gyroscope automatic calibration  
1: removed to gyroscope automatic calibration



## 5.2.8 Return Content

0xFF	0xAA	0x02	RSWL	RSWH
------	------	------	------	------

RSWL byte definition

byte	7	6	5	4	3	2	1	0
Name	0x57 pack	0x56 pack	0x55 pack	0x54 pack	0x53 pack	0x52 pack	0x51 pack	0x50 pack
default	0	0	0	1	1	1	1	0

RSWH byte definition

byte	7	6	5	4	3	2	1	0
Name	X	X	X	X	X	0x5A pack	0x59 pack	0x58 pack
default	0	0	0	0	0	0	0	0

X is an undefined value.

0x50 pack: time pack

0: Not output 0X50 pack

1: Output 0X50 pack

0x51 pack: Acceleration pack

0: Not output 0x51 pack

1: Output 0x51 pack

0x52 pack: Angular velocity pack

0: Not output 0x52 packet

1: Output 0x52 pack

0x53 pack: Angle Pack

0: Not output 0x53 pack

1: Output 0x53 pack

0x59 pack: Quaternion Pack

0: Not output 0x59 pack

1: Output 0x59 pack

## 5.2.9 Return Rate

0xFF	0xAA	0x03	RATE	0x00
------	------	------	------	------

RATE: return rate

- 0x01: 0.2Hz
- 0x02: 0.5Hz
- 0x03: 1Hz
- 0x04: 2Hz
- 0x05: 5Hz
- 0x06: 10Hz(default)
- 0x07: 20Hz
- 0x08: 50Hz
- 0x09: 100Hz
- 0x0a: 125Hz
- 0x0b: 200Hz
- 0x0c: Single
- 0x0d: Not output

After the setup is complete, need to click save, and re-power the module to take effect.

## 5.2.10 Baud Rate

0xFF	0xAA	0x04	BAUD	0x00
------	------	------	------	------

BAUD:

- 0x01: 4800
- 0x02: 9600(default)
- 0x03: 19200
- 0x04: 38400
- 0x05: 57600
- 0x06: 115200
- 0x07: 230400
- 0x08: 460800
- 0x09: 921600

### 5.2.11 Set X Axis Acceleration Bias

0xFF	0xAA	0x05	AXOFFSETL	AXOFFSETH
------	------	------	-----------	-----------

AXOFFSETL: X axis Acceleration bias low byte

AXOFFSETH: X axis Acceleration bias high byte

AXOFFSET= (AXOFFSETH <<8) | AXOFFSETL

Note: After setting the acceleration bias, the output value of the acceleration is the sensor measured value minus the bias value.

### 5.2.12 Set Y Axis Acceleration Bias

0xFF	0xAA	0x06	AYOFFSETL	AYOFFSETH
------	------	------	-----------	-----------

AYOFFSETL: Y axis Acceleration bias low byte

AYOFFSETH: Y axis Acceleration bias high byte

AYOFFSET= (AYOFFSETH <<8) | AYOFFSETL

Note: After setting the acceleration bias, the output value of the acceleration is the sensor measured value minus the bias value.

### 5.2.13 Set Z Axis Acceleration Bias

0xFF	0xAA	0x07	AZOFFSETL	AZOFFSETH
------	------	------	-----------	-----------

AZOFFSETL: Z axis Acceleration bias low byte

AZOFFSETH: Z axis Acceleration bias high byte

AZOFFSET= (AZOFFSETH <<8) | AZOFFSETL

Note: After setting the acceleration bias, the output value of the acceleration is the sensor measured value minus the bias value.

## 5.2.14 Set X Axis Angular Velocity Bias

0xFF	0xAA	0x08	GXOFFSETL	GXOFFSETH
------	------	------	-----------	-----------

GXOFFSETL: Set X axis Angular velocity bias low byte

GXOFFSETH: Set Y axis Angular velocity bias high byte

GXOFFSET= (GXOFFSETH <<8) | GXOFFSETL

Note: After setting the angular velocity zero deviation, the output value of the angular velocity is the sensor measurement value minus the zero deviation value.

## 5.2.15 Set Y Axis Angular Velocity Bias

0xFF	0xAA	0x09	GYOFFSETL	GYOFFSETH
------	------	------	-----------	-----------

GYOFFSETL: Set X axis Angular velocity bias low byte

GYOFFSETH: Set X axis Angular velocity bias high byte

GYOFFSET= (GYOFFSETH <<8) | GYOFFSETL

Note: After setting the angular velocity zero deviation, the output value of the angular velocity is the sensor measurement value minus the zero deviation value.

## 5.2.16 Set Z Axis Angular Velocity Bias

0xFF	0xAA	0x0A	GZOFFSETL	GZOFFSETH
------	------	------	-----------	-----------

GZOFFSETL: Set Z axis Angular velocity bias low byte

GZOFFSETH: Set Z axis Angular velocity bias low byte

GZOFFSET= (GZOFFSETH <<8) | GZOFFSETL

Note: After setting the angular velocity zero deviation, the output value of the angular velocity is the sensor measurement value minus the zero deviation value.