WitMotion Shenzhen Co., Ltd| Datasheet

AHRS IMU Sensor | WT61P

The Robust Acceleration, Angular velocity & Angle Detector

The WT61P is a IMU sensor device, detecting acceleration, angular velocity as well as angle. 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
WT61P 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

1 Overview

WT61P's scientific name is AHRS IMU sensor. A sensor measures 3-axis angle, angular velocity, acceleration. Its strength lies in the algorithm which can calculate three-axis angle accurately.

WT61P is employed where the highest measurement accuracy is required. WT61P 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 Roll Pitch Yaw (X Y Z axis) Acceleration + Angular Velocity + Angle output

• Low cost of ownership: remote diagnostics and lifetime technical support by WITMOTION service team

•Developed tutorial: providing manual, datasheet, Demo video, free software for Windows computer, APP for Android smartphones , and sample code for MCU integration including 51 serial, STM32, Arduino, Matlab, Raspberry Pi,communication protocol for project

• 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 and geomagnetic field 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 200Hz data output rate. Output content can be arbitrarily selected, the output speed 0.2HZ~ 200HZ adjustable.

3 Specification

3.1 Parameter

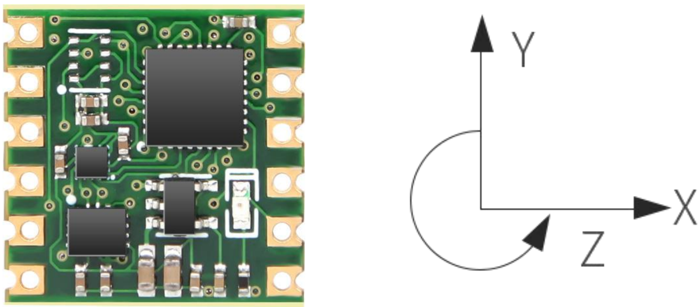
Parameter	Specification
➢ Working Voltage	3.3V-5V
➢ Current	<25mA
➢ Size	15mm x 15mm X 2mm
➢ Data	Angle: X Y Z, 3-axis Acceleration: X Y Z, 3-axis Angular Velocity: X Y Z, 3-axis Time, Quaternion
➢ Output frequency	0.2Hz--200Hz
➢ Interface	Serial TTL level
➢ Baud rate	4800,9600(default),19200,38400,57600, 115200,230400,460800,921600

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/ Inclinometer	X, Y, Z, 3-axis X, Z-axis: ±180° Y ±90° (Y-axis 90° is singular point)	Accuracy:X, Y-axis: 0.05° Z-axis: 1° (Angle of Z-axis will have accumulated error)

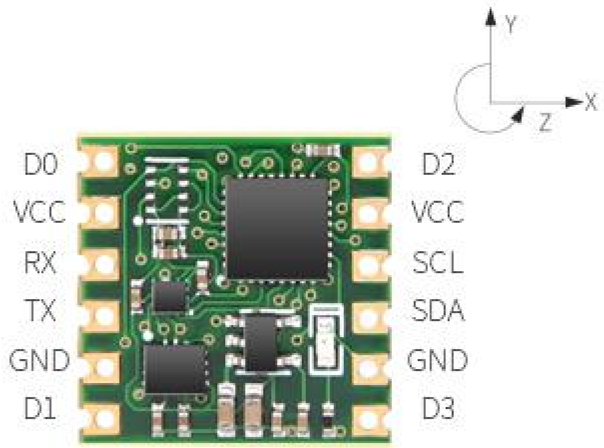
3.2 Size

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	Function
➢ VCC	3.3-5V input supply
➢ RX	Serial data input, TTL interface
➢ TX	Serial data output, TTL interface
➢ GND	Ground
➢ DO	Analog input, digital input and output, PWM
➢ D1	Analog input, digital input and output, PWM, connect GPS
➢ D2	Analog input, digital input and output, PWM
➢ D3	Analog input, digital input and output, PWM
➢ SDA	I2C signal line
➢ SCL	I2C clock line

5 Communication Protocol

Level: TTL 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<<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(g is Gravity acceleration, 9.8m/s^2)

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

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

Temperature calculated formular：

T=((TH<<8)|TL)/100 ℃

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(°/s)

w_y=((wyH<<8)|wyL)/32768*2000(°/s)

w_z=((wzH<<8)|wzL)/32768*2000(°/s)

Temperature calculated formular：

T=((TH<<8)|TL) /100 ℃

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(°)
Pitch(Y axis)Pitch=((PitchH<<8)|PitchL)/32768*180(°)
Yaw(Z axis)Yaw=((YawH<<8)|YawL)/32768*180(°)
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 ± 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 ± 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 Data Output Port Status

0x55	0x55	D0L	D0H	D1L	D1H	D2L	D2H	D3L	D3H	SUM

Calculated formular:
D0 = (D0H<<8)| D0L
D1 = (D1H<<8)| D1L
D2 = (D2H<<8)| D2L
D3 = (D3H<<8)| D3L

Note:
Analog input port mode:
U=DxStatus/1024*U_{vcc}
Uvcc is the power supply voltage of the module, because the module has LDO, if the module power supply voltage is greater than 3.5V, Uvcc is 3.3V. If the module supply voltage is less than 3.5V, Uvcc equal to the supply voltage minus 0.2V
Digital input mode:
Voltage level is high, the data is 1,
Voltage level is low, the data is 0.
Digital output mode:
Output is high, the data is 1.
Output is low, the data is 0.
PWM output mode:
When the port is set to PWM output mode, port status data indicates high level width, the unit is "us".

5.1.6 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:
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
0x0e	D0MODE	D0 mode
0x0f	D1MODE	D1 mode
0x10	D2MODE	D2 mode
0x11	D3MODE	D3 mode
0x12	D0PWMH	D0PWM High-level width
0x13	D1PWMH	D1PWM High-level width
0x14	D2PWMH	D2PWM High-level width
0x15	D3PWMH	D3PWM High-level width
0x16	D0PWMT	D0PWM Period
0x17	D1PWMT	D1PWM Period
0x18	D2PWMT	D2PWM Period
0x19	D3PWMT	D3PWM Period
0x1a	IICADDR	IIC address
0x1b	LEDOFF	Turn off LED
0x1c	GPSBAUD	GPS baud rate
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
0x3f	Yaw	Z axis Angle
0x40	TEMP	Temperature
0x41	D0Status	D0Status
0x42	D1Status	D1Status
0x43	D2Status	D2Status
0x44	D3Status	D3Status
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.

Eg:

1. FF AA 69 88 B5(Unlock)
2. FF AA 22 01 00 (Sleep State)
3.FF AA 22 01 00 (Unsleep State)

5.2.6 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. 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

0x55 pack：Port status pack
- 0：Not output 0x55 pack
- 1：Output 0x55 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.

Eg(20Hz of Return Rate):
1. FF AA 69 88 B5(Unlock)
2. FF AA 03 07 00(20HZ)
3.FF AA 00 00 00(Save Config)

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	GXOFFSETL	GXOFFSETH

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.

5.2.17 Set port D0 mode

0xFF	0xAA	0x0e	D0MODE	0x00

D0MODE：

- 0x00：Analog Input（default）
- 0x01：Digital Input
- 0x02：Digital Output high
- 0x03：Digital Output low
- 0x04：PWM Output

5.2.18 Set port D1 mode

0xFF	0xAA	0x0f	D1MODE	0x00

D1MODE：

0x00：Analog Input（default）
0x01：Digital Input
0x02：Digital Output high
0x03：Digital Output low
0x04：PWM Output
0x05：Connect to TX of GPS

5.2.19 Set port D2 mode

0xFF	0xAA	0x0e	D2MODE	0x00

D2MODE：

0x00：Analog Input（default）
0x01：Digital Input
0x02：Digital Output high
0x03：Digital Output low
0x04：PWM Output

5.2.20 Set port D3 mode

0xFF	0xAA	0x0e	D3MODE	0x00

D3MODE：

0x00：Analog Input（default）

0x01：Digital Input

0x02：Digital Output high

0x03：Digital Output low

0x04：PWM Output

5.2.21 Set the PWM width of Port D0

0xFF	0xAA	0x12	D0PWMHL	D0PWMHH

D0PWMHL：the PWM width of Port D0 low byte

D0PWMHH：the PWM width of Port D0 high byte

D0PWMH = (D0PWMHH<<8) | D0PWMHL

Note: The unit of PWM high-level width and period is us, such as high-level width is 1500us, just set D0PWMH 1500.

5.2.22 Set the PWM width of Port D1

0xFF	0xAA	0x13	D1PWMHL	D1PWMHH

D1PWMHL：the PWM width of Port D1 low byte
D1PWMHH：the PWM width of Port D1 high byte
D1PWMH = (D1PWMHH<<8) | D1PWMHL

Note: The unit of PWM high-level width and period is us, such as high-level width is 1500us, just set D0PWMH 1500.

5.2.23 Set the PWM width of Port D2

0xFF	0xAA	0x14	D2PWMHL	D2PWMHH

D2PWMHL：the PWM width of Port D2 low byte
D2PWMHH：the PWM width of Port D2 high byte
D2PWMH = (D2PWMHH<<8) | D2PWMHL

Note: The unit of PWM high-level width and period is us, such as high-level width is 1500us, just set D0PWMH 1500.

5.2.24 Set the PWM width of Port D3

0xFF	0xAA	0x15	D3PWMHL	D3PWMHH

D3PWMHL：the PWM width of Port D3 low byte
D3PWMHH：the PWM width of Port D3 low byte
D3PWMH = (D3PWMHH<<8) | D3PWMHL

Note: The unit of PWM high-level width and period is us, such as high-level width is 1500us, just set D0PWMH 1500.

5.2.24 Set period of Port D0

0xFF	0xAA	0x16	D0PWMTL	D0PWMTH

D0PWMTL：PWM period of Port D0 low byte
D0PWMTH：PWM period of Port D0 high byte
D0PWMT = (D0PWMTH<<8) | D0PWMTL

Note: The unit of PWM high-level width and period is us, such as high-level width is 1500us, just set D0PWMH 1500.Period is 20000us, just set D0PWMT 20000.

5.2.25 Set period of Port D1

0xFF	0xAA	0x17	D1PWMTL	D1PWMTH

D1PWMTL：PWM period of Port D1 low byte
D1PWMTH：PWM period of Port D1 high byte
D1PWMT = (D1PWMTH<<8) | D1PWMTL

Note: The unit of PWM high-level width and period is us, such as high-level width is 1500us, just set D0PWMH 1500.Period is 20000us, just set D0PWMT 20000.

5.2.26 Set period of Port D2

0xFF	0xAA	0x18	D2PWMTL	D2PWMTH

D2PWMTL：PWM period of Port D2 low byte
D2PWMTH：PWM period of Port D2 high byte
D2PWMT = (D2PWMTH<<8) | D2PWMTL

Note: The unit of PWM high-level width and period is us, such as high-level width is 1500us, just set D0PWMH 1500.Period is 20000us, just set D0PWMT 20000.

5.2.27 Set period of Port D3

0xFF	0xAA	0x19	D3PWMTL	D3PWMTH

D3PWMTL：PWM period of Port D3 low byte
D3PWMTH：PWM period of Port D3 high byte
D3PWMT = (D3PWMTH<<8) | D3PWMTL

Note: The unit of PWM high-level width and period is us, such as high-level width is 1500us, just set D0PWMH 1500.Period is 20000us, just set D0PWMT 20000.

5.2.28 Set IIC Address

0xFF	0xAA	0x1a	IICADDR	0x00

IICADDR：

IIC address of the module, default is 0x50. IIC address using 7bit address, can not exceed the maximum 0x7f. After the setup is complete , need to click save, and re-power the module to take effect.

5.2.29 LED

0xFF	0xAA	0x1b	LEDOFF	0x00

LEDOFF：
0x01：Turn off LED
0x00：Turn on LED

6 IIC Protocol

WT61P module can be fully accessed through IIC, the maximum IIC communication speed support 400khz, slave module address is 7bit, default address is 0x50, you can change the command through the serial port or the methods of IIC writing address ways. Many WT61P modules can be connect to IIC bus at the same time, The precondition is that the module has the different IIC address.

IIC protocol module using the register address accessible way. The length of each address are 16bits, two bytes. The register address is defined in the following table:

RegAddr	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
0x1a	IICADDR	IIC address
0x1b	LEDOFF	Turn off LED
0x1c	GPSBAUD	GPS baud rate
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

0x3f	Yaw	Z axis Angle
0x40	TEMP	Temperature

6.1 IIC Write the Module

When IIC write the module, the format is as below:

IICAddr<<1	RegAddr	Data1L	Data1H	Data2L	Data2H	……

First IIC host sends a Start signal to WT61P module, then write IICAddr to register address and then write RegAddr, write the Data1L Data1H Data2L Data2H …. Sequentially, when the last data has been written, the host sends a stop signal to the module to release the IIC bus.

When finish writing the data, the register will be updated and module will execute the order.At the same time,the address of the module will add 1 automatically.The address Pointer will point to next address. So it can be written Continuously

For example:
Set D0 as Digital output high
RegAddr :0x0e DataL:0x02 DataH:0x00
Logic Analyzer captures waveforms as shown below:

Register set up by the module approach is consistent with the serial protocol, please refer 6.1

6.2 IIC Read the Module

IIC read the module, the format is as follow

IICAddr<<1	RegAddr	(IICAddr<<1)|1	Data1L	Data1H	Data2L	Data2H	......

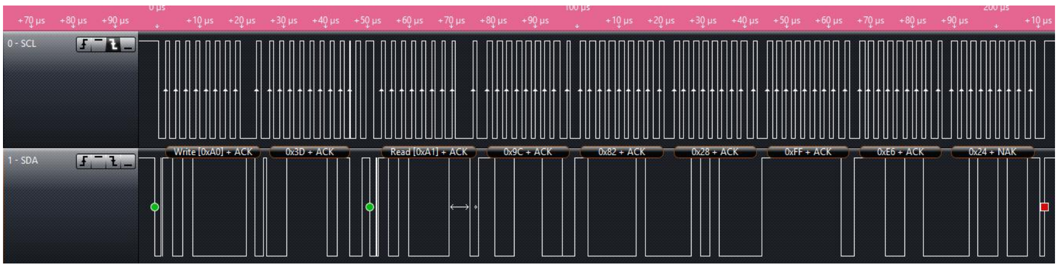
First IIC host sends a Start signal to WT61P module,then write IICAddr to register address, then IIC host sends a read signal(IICAddr<<1)|1) to WT61P module, if the IIC address is 0x50(default),then the host send 0xa0

Thereafter the module will export the data follow the rule: low byte first, high byte Sequentially. The host will make SDA bus low after receiving each byte, and sends a response signal to the module .After the specified number of data has been received completely, the host stop sending response signal back to the module, then the module will stop export data.The host send a stop signal to end this operation.

For example:

Read the Angle of the module,

RedAddr: 0x3d,read 6 bytes continuously, the logic analyzer captures waveforms as shown below:



Start reading out data from 0x3d,the data is 0x9C, 0x82,0x28,0xFF, 0xE6,0x24. That means X-axis angle is 0x829C, Y-axis angle is 0xFF28, Z-axis angle is 0x24E6. According to section 5.1.4 , X axis angle is -176.33 °, Y-axis angle is -1.19 °, Z-axis angle is 51.89 °