Bluetooth 5.0 communication protocol

Contact

Module to PC

The module uploads the data of Flag=0x61 (acceleration angular velocity angle) by default.

Flag=0x71 (magnetic field) needs to send the command to read the corresponding register to return.

Bluetooth upload data format: Bluetooth can upload data up to 20Byte each time.

acceleration angular velocity angle data packet (default)

packet header 1Byte	flag bit 1Byte	axL	axH	......	YawL	YawH
0x55	Flag	0xNN	0xNN	......	0xNN	0xNN

Note: 0xNN is the specific value received, the order of data return is acceleration X Y Z, angular velocity X Y Z, angle XYZ,

The low byte comes first, and the high byte follows.

Flag = 0x61 Data content 18Byte is acceleration, angular velocity, angle

0x55	packet header
0x61	flag bit
axL	X-axis acceleration low 8 bits
axH	X-axis acceleration high 8 bits
ayL	Y-axis acceleration low 8 bits
ayH	Y-axis acceleration high 8 bits
azL	Z-axis acceleration low 8 bits
azH	Z-axis acceleration high 8 bits
wxL	X-axis angular velocity low 8 bits
wxH	X-axis angular velocity high 8 bits
wyL	Y-axis angular velocity low 8 bits
wyH	Y-axis angular velocity high 8 bits
wzL	Z-axis angular velocity low 8 bits
wzH	Z-axis angular velocity high 8 bits

RollL	X-axis angle velocity low 8 bits
RollH	X-axis angle velocity high 8 bits
PitchL	Y-axis angle velocity low 8 bits
PitchH	Y-axis angle velocity high 8 bits
YawL	Z-axis angle velocity low 8 bits
YawH	Z-axis angle velocity high 8 bits

Acceleration calculation method: unit g

ax=((axH<<8)|axL)/32768*16g(g is the acceleration of gravity, it is desirable 9.8m/s2)

ay=((ayH<<8)|ayL)/32768*16g(g is the acceleration of gravity, it is desirable 9.8m/s2)

az=((azH<<8)|azL)/32768*16g(g is the acceleration of gravity, it is desirable 9.8m/s2)

Calculation method of angular velocity: unit °/s

wx=((wxH<<8)|wxL)/32768*2000(°/s)

wy=((wyH<<8)|wyL)/32768*2000(°/s)

wz=((wzH<<8)|wzL)/32768*2000(°/s)

Angle Calculation Method: Unit °

roll angle (x-axis)Roll=((RollH<<8)|RollL)/32768*180(°)

pitch angle (y-axis) Pitch=((PitchH<<8)|PitchL)/32768*180(°)

Yaw angle (z axis) Yaw=((YawH<<8)|YawL)/32768*180(°)

Note:

1. The coordinate system used in the settlement of the attitude angle is the northeast sky coordinate system, and the module is placed in the positive direction, such as "4 pin description"

Left is shown for the X axis, forward for the Y axis, and up for the Z axis. The rotation order of the coordinate system when the Euler angle represents the attitude

It is defined as Z-Y-X, that is, first rotate around the Z axis, then around the Y axis, and then around the X axis.

2. Although the range of the roll angle is ±180 degrees, in fact, because the coordinate rotation sequence is Z-Y-X, it represents the attitude

When , the range of the pitch angle (Y axis) is only ±90 degrees, and after exceeding 90 degrees, it will be changed to less than 90 degrees, and at the same time

Let the angle of the X axis be greater than 180 degrees. For the detailed principle, please Baidu the relevant information about Euler angle and attitude representation.

3. Since the three axes are coupled, they will show independent changes only at small angles, and the posture at large angles

The angle will be coupled and changed, 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 X axis

It will also change greatly, which is the inherent characteristic of Euler angles to express attitude.

illustrate:

1. Data is sent in hexadecimal, not ASCII.

2. Each data is transmitted sequentially in 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 the low byte, and AxH is the high byte. The conversion method is as follows:

Assuming that Data is the actual data, DataH is its high byte, and DataL is its low byte, then: Data=((short)DataH<<8)|DataL. It must be noted here that DataH needs to be cast to a signed short first

The data of type will be shifted later, and the data type of Data is also a signed short type, so that negative numbers can be represented.

A packet with a single return register

A single return data packet needs to send a read register instruction first, and the instruction format is as follows:

FF AA 27 XX 00

--XX refers to the corresponding register number, the number of the register is for reference, and the example of sending the command is as follows：

function	command
read magnetic field	FF AA 27 3A 00
read four elements	FF AA 27 51 00
read temperature	FF AA 27 40 00
read power	FF AA 27 64 00

After sending this command, the module will return a data packet beginning with 0x55 0x71, which contains the corresponding start register address data, the start of the start register address and the following 7 register data (8 registers are fixedly uploaded), and the returned data format is as follows:

Start register (2Byte) + register data (16Byte, 8 registers)

packet header	sign	Start register address low	Start register address high	Start (1st) register data low bit	Open (1st) register number high bit	......	The low bit of the 8th register data	The 8th register data high bit
0x55	0x71	RegL	RegH	0xNN	0xNN	......	0xNN	0xNN

Note: 0xNN is the specific value received, the low byte comes first, and the high byte follows.

Magnetic field output

0x55	0x71	0x3A	0x00	HxL	HxH	HyL	HyH	HzL	HzH.....

Calculation method: unit mG

Magnetic field (x-axis) Hx=((HxH<<8)| HxL)

Magnetic field (y axis) Hy=((HyH <<8)| HyL)

Magnetic field (z axis) Hz =((HzH<<8)| HzL)

Example: Send the command to read the magnetic field on the APP: FF AA 27 3A 00 (refer to 7.2.8 Read Register Value)

The module returns data to APP: 55 71 3A 00 68 01 69 00 7A 00 00 00 00 00 00 00 00 00 00 00, a total of 20 bytes.

For the 5th to 10th bytes, calculate as above, the magnetic field x=360, y=105, z=122.

Quaternion output

0x55	0x71	0x51	0x00	QxL	QxH	QyL	QyH	QzL	QzH.....

Calculation method:

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

temperature output

0x55	0x71	0x40	0x00	TL	TH	......

Temperature calculation formula:

T=((TH<<8)|TL) /100 ℃

Host computer to module Send command:

read register value

FF AA 27 XX 00	read register value

--XX refers to the corresponding register.

example:

Read magnetic field: FF AA 27 3A 00

Read four elements: FF AA 27 51 00

Read temperature: FF AA 27 40 00

After sending this command, the module will return a data packet beginning with 0x55 0x71, which contains the corresponding start register address data, the beginning of the start register address and the following 7 register data (8 registers are fixedly uploaded), and the return data format refers to

Acceleration Calibration and Magnetic Field Calibration

FF AA 01 01 00	Acceleration Calibration
FF AA 01 05 00	Acceleration Calibration L
FF AA 01 06 00	Acceleration Calibration R
FF AA 01 07 00	Magnetic Field Calibration
FF AA 01 00 00	Complete Magnetic Field Calibration

save configuration

FF AA 00 SAVE 00	save configuration

SAVE: set

0: save the current configuration

1: Restore the default configuration and save

set return rate

FF AA 03 RATE 00	set return rate

RATE：return rate

0x01：0.1Hz

0x02：0.5Hz

0x03：1Hz

0x04：2Hz

0x05：5Hz

0x06：10Hz（default）

0x07：20Hz

0x08：50Hz

0x09：100Hz

0x0A：200Hz

Read power

FF AA 27 64 00	Read the power of the module

return data：55 71 64 00 48 03 00 00 AA 00 00 00 00 00 00 00 00 00 00 00

Among them, the two data of 48 03 represent the power, which is 0348 when converted into a hexadecimal number, and 840 when converted into a decimal number, which means that the corresponding power is 100%.

The power relationship corresponding to the decimal number is as follows:

Greater than 830 is 100%

750～830 is 75%

715～750 is 50%

675～715 is 25%

Below 675 is 0%

Register Address Table

address	symbol	meaning
0x00	SAVE	save current configuration
0x01	CALSW	calibration
0x02	SAVE	
0x03	RATE	return data rate
0x04	BAUD	Serial baud rate
0x05	AXOFFSET	X-axis acceleration zero bias
0x06	AYOFFSET	Y-axis acceleration zero bias
0x07	AZOFFSET	Z-axis acceleration zero bias
0x08	GXOFFSET	X-axis angular velocity zero bias
0x09	GYOFFSET	Y-axis angular velocity zero bias
0x0a	GZOFFSET	Z axis angular velocity zero bias
0x0b	HXOFFSET	X-axis magnetic field zero bias
0x0c	HYOFFSET	Y-axis magnetic field zero bias
0x0d	HZOFFSET	Z-axis magnetic field zero bias
0x0e	D0MODE	D0 model
0x0f	D1MODE	D1 model
0x10	D2MODE	D2 model

0x11	D3MODE	D3 model
0x12	SAVE	
0x13	SAVE	
0x14	SAVE	
0x15	SAVE	
0x16	SAVE	
0x17	SAVE	
0x18	SAVE	
0x19	SAVE	
0x1a	SAVE	
0x1b	SAVE	
……	……	……
0x30	YYMM	Year,month
0x31	DDHH	day, time
0x32	MMSS	minutes, seconds
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
0x3a	HX	X-axis magnetic field
0x3b	HY	Y-axis magnetic field

0x3c	HZ	Z-axis magnetic field
0x3d	Roll	X-axis angle
0x3e	Pitch	Y-axis angle
0x3f	Yaw	Z-axis angle
0x40	TEMP	module temperature
0x49	SAVE	
0x4a	SAVE	
0x4b	SAVE	
0x4c	SAVE	
0x4d	SAVE	
0x4e	SAVE	
0x4f	SAVE	
0x50	SAVE	
0x51	Q0	Four elements Q0
0x52	Q1	Four elements Q1
0x53	Q2	Four elements Q2
0x54	Q3	Four elements Q3