

---

## 角度模块，用于测量累计角度，瞬时角速度，瞬时加速度

---

### 主要特点

- 累计角度和瞬时角速度输出
- 3 轴瞬时加速度数据输出
- 累计低偏置漂移
- 高分辨率和高精度的数据输出
- 振动感知
- 优秀的角度因子线性度
- 上电后快速启动
- 串口总线数据输出
- 低功耗
- 符合 RoHS 标准

### 应用指南

- 路径识别
- 机器人自主导航
- 车辆导航
- 无人机导航
- 医疗导轨应用
- 平衡车平衡检测
- 柜门开关角度检测
- 工业机器人振动检测

## 产品概述

明碁传感 MMC127A 是用于测量角速度的数字陀螺仪和加速度计的集成模块。它可以在动态条件下测量航向角（也称为取向或偏航角）和加速度。MMC127A 内部包含一颗高精度陀螺仪、3 轴加速度计、内部电压调节器、信号处理电路、晶振电路和一颗微处理器运行专利滤波算法。MMC127A 使用自适应降阶卡尔曼滤波器降低传感器自身材料和制作工艺带来的误差（即偏置漂移，零度漂移，随机漂移等非线性误差），因此它具有非常精确稳定的角速度和累计航向角以及加速度值。启动时间小于 2 秒，初始累计角度为零，它不需要进一步的校准，通过串口可以直接取出上述三种数据。综上，MMC127A 是提供低成本高精度的消费类智能产品应用的最佳解决方案。

## 索引

1	硬件引脚定义.....	4
2	机械和电气规范.....	5
2.1	机械特性.....	5
2.2	电气特性.....	5
2.3	温度最大额定参数.....	5
2.4	设备启动信息.....	6
3	串口数据格式.....	7
3.1	串口传输参数设定.....	7
3.2	输出数据格式.....	7
4	应用指南.....	8
4.1	封装信息.....	8
4.2	硬件线路接线参考图.....	9
4.3	上电时序图.....	9
4.4	模块坐标系.....	10
4.5	输出数据解析（C 语言）.....	11
4.6	漂移校准步骤.....	12
4.7	旋转平面切换.....	12
4.8	振动档位阈值修改.....	13
5	修订历史.....	14

## 图

图 1	模块正面俯视图.....	4
图 2	机械结构和封装尺寸描述(单位:mm).....	8
图 3	外围 MCU 接线参考图.....	9
图 4	上电时序图.....	9
图 5	顶视图.....	10
图 6	校准步骤.....	12
图 7	档位选择.....	12

## 表

表 1	硬件引脚推荐连接方式.....	4
表 2	机械特性.....	5
表 3	电气特性.....	5
表 4	温度最大额定参数.....	5
表 5	串口传输协议参数.....	7
表 6	数据位描述和说明见下表.....	7
表 7	字节数据详细含义.....	7
表 8	文档修订清单.....	14

# 1 硬件引脚定义

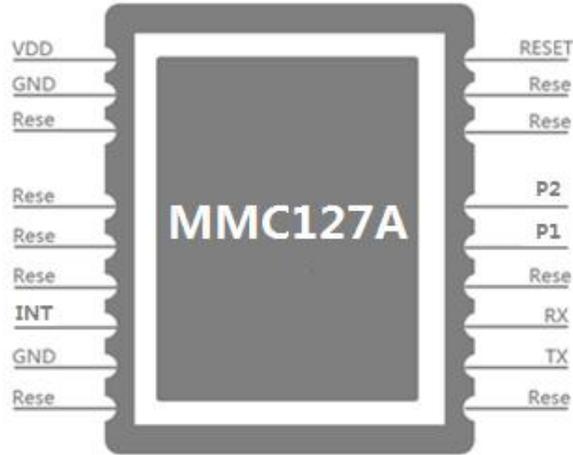


图 1 模块正面俯视图

表 1 硬件引脚推荐连接方式

引脚序列	名称	功能描述
1	VDD	电源
2	GND	电源地
3	保留	悬空
4	保留	悬空
5	保留	悬空
6	保留	悬空
7	INT	提示振动报警，默认为高阻态，当触发振动报警时，输出 5 秒低电平
8	GND	电源地
9	保留	悬空
10	保留	悬空
11	TX	串口发送数据
12	RX	串口接收数据
13	保留	悬空
14	P1	振动档位选择 1 脚，具体配置请参见 4.8
15	P2	振动档位选择 2 脚，具体配置请参见 4.8
16	保留	悬空
17	保留	悬空
18	RESET	重启管脚，低电平有效，持续拉低 20 毫秒重启

## 2 机械和电气规范

### 2.1 机械特性

VDD = 3.3 V, T = 25 °C 标准测试环境，除非另有标注

表 2 机械特性

参数	条件	最小	类型	最大	单位
角度 & 角速度	角速度范围	-250		250	° /s
	角度范围	-180		180	°
	噪声范围			0.3	° /s
	非线性比例因子		1%		
	带宽		50		Hz
	偏置漂移		30		° /hr
加速度	输入动态范围	-4		4	g
输出频率			50		Hz

### 2.2 电气特性

表 3 电气特性

参数	测试条件	最小	参考值	最大	单位
功耗	25°C		20		mA
输入电压	25°C	2.4	3.3	3.6	V
逻辑电平	25°C	-0.3	3.3	3.6	V

### 2.3 温度最大额定参数

环境温度高于下列表格中“绝对最大额定参数”可能会造成永久性损坏设备。这仅仅是存储或运输环境的温度等级，运行情况下温度也必须严格遵守下列表格中参数。长时间接触最大额定值条件可能影响设备的可靠性。

表 4 温度最大额定参数

参数	条件	最小	最大	单位
存储温度		-45	125	°C
工作温度		-20	85	°C



这是一个机械冲击敏感设备, 处理不当会造成传感器器件永久损害。



这是一个静电敏感设备, 处理不当会造成电子器件的永久损害。

## 2.4 设备启动信息

MMC127A 的启动时间小于 2 秒, 内部有温度补偿机制。MMC127A 必须在启动期间保持静止, 否则会引起数据的漂移。

## 3 串口数据格式

### 3.1 串口传输参数设定

表 5 串口传输协议参数

电平类型	波特率	数据位	奇偶校验	停止位	硬件流控制
TTL	115200	8Bit	None	1Bit	None

### 3.2 输出数据格式

MMC127A 提供角速度、角度和加速度输出。角度输出是相对的，角速度输出范围在 $\pm 250^\circ /s$ ，加速度输出范围在 $\pm 4g$ 。

#### 3.2.1 整数输出格式

表 6 数据位描述和说明见下表

数据头		数据											校验位	
		索引	角度		角速度		X 轴 加速度		Y 轴 加速度		Z 轴 加速度			保留
0xAA	0x00		MSB	LSB	MSB	LSB	MSB	LSB	MSB	LSB	MSB	LSB		
2byte		1byte	2byte		2byte		2byte		2byte		2byte		1byte	1byte

表 7 字节数据详细含义

输出数据	字节	描述
数据头	1-2	十六进制值: 0xAA00
索引	3	0x00~ 0xFF
角度	4-5	分辨率为百分之一度 范围为正负 180 度
角速度	6-7	分辨率百分之一度，单位为度/秒
X 轴加速度	8-9	分辨率 1 mg
Y 轴加速度	10-11	分辨率 1 mg
Z 轴加速度	12-13	分辨率 1 mg
保留位	14	
校验位	15	CRC = index + angle(LSB) + angle(MSB) + rate(LSB) + rate(MSB) + Xacc(LSB) + Xacc(MSB) + Yacc(LSB) + Yacc(MSB) + Zacc(LSB) + Zacc(MSB) + reserved

## 4 应用指南

### 4.1 封装信息

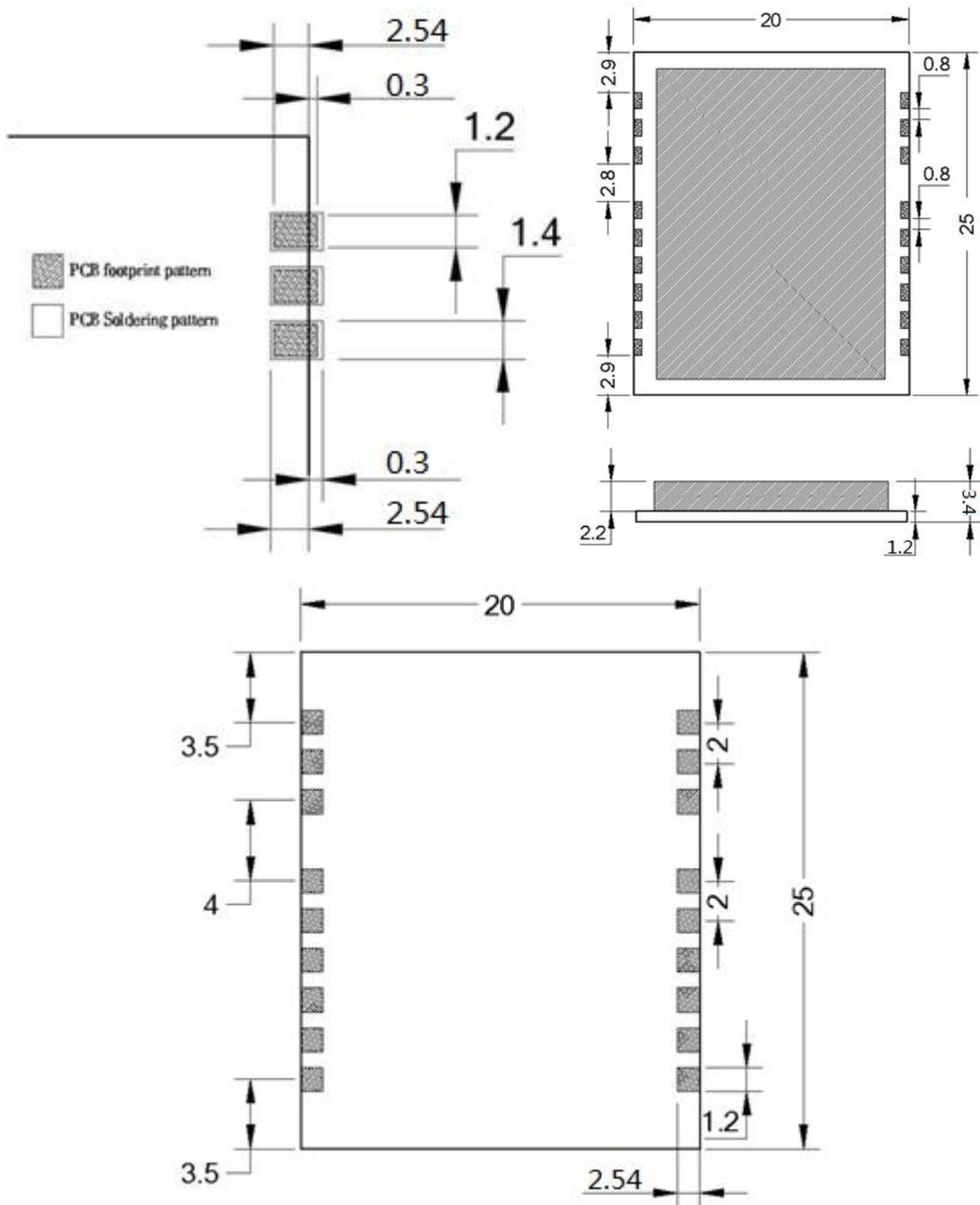


图 2 机械结构和封装尺寸描述(单位:mm)

## 4.2 硬件线路接线参考图

如果模块和 MCU 之间的逻辑电压超出图 3 中给出的值，那么需要在电路中加入一个电平转换芯片。

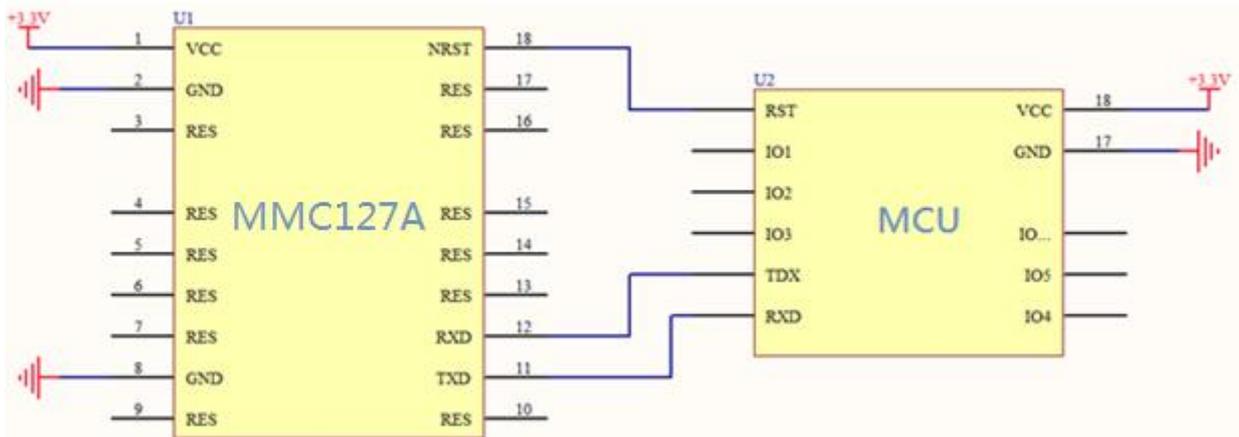


图 3 外围 MCU 接线参考图

## 4.3 上电时序图

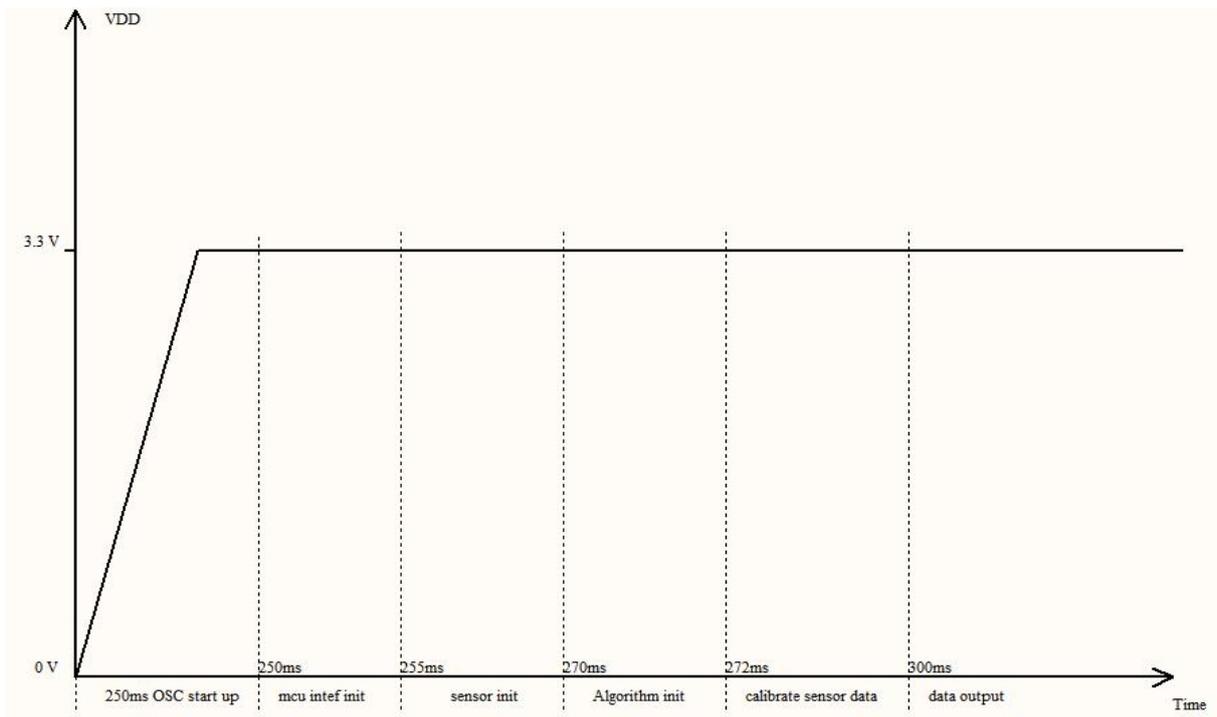


图 4 上电时序图

## 4.4 模块坐标系

示例：如果传感器在重力场中静止或均匀运动，则根据下图，输出信号为：

- AX ACC 通道输出  $\pm 0g$  ，并且GZ通道输出  $\pm 0^\circ / \text{sec}$
- AY ACC 通道输出  $\pm 0g$
- AZ ACC 通道输出  $\pm 0g$

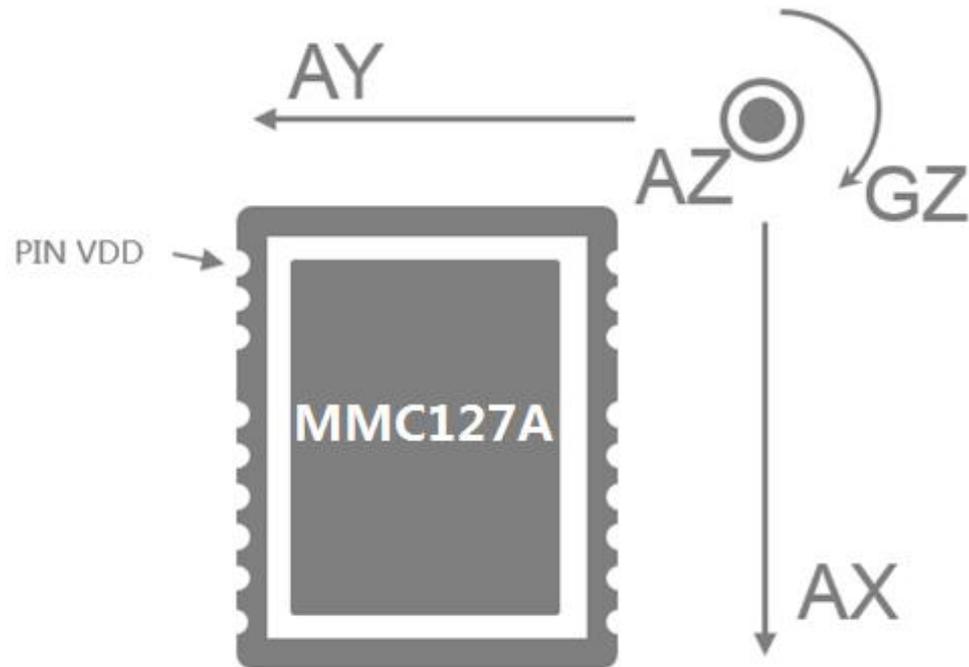


图 5 顶视图

## 4.5 输出数据解析（C 语言）

The following C program shows how to parse MMC127A output data packet.

```
//This program assumes that the complete data package has been conveniently stored in the
//data_string array variable that is passed as an argument. After parsing the data packet,
//this function stores the results in the global variables gRate, gAngle, gX_acc, gY_acc, gZ_acc.
//If successful the function returns true otherwise false

extern float32_t gAngle;
extern float32_t gRate;
extern float32_t gX_acc;
extern float32_t gY_acc;
extern float32_t gZ_acc;

bool parse_data(uint8_t *data_string) {
    uint8_t index;
    int16_t angle;
    int16_t rates;
    int16_t x_acc;
    int16_t y_acc;
    int16_t z_acc;
    int16_t pitch;
    int16_t roll;
    uint8_t check_sum;
    //Verify packet heading information
    if (data_string[0] != 0xAA || data_string[1] != 0x00) {
        printf("Data heading error");
        return false;
    }
    //Assemble data
    index = data_string[2];
    angle = (data_string[3] & 0xFF) | ((data_string[4] << 8) & 0xFF00);
    rates = (data_string[5] & 0xFF) | ((data_string[6] << 8) & 0xFF00);
    x_acc = (data_string[7] & 0xFF) | ((data_string[8] << 8) & 0xFF00);
    y_acc = (data_string[9] & 0xFF) | ((data_string[10] << 8) & 0xFF00);
    z_acc = (data_string[11] & 0xFF) | ((data_string[12] << 8) & 0xFF00);
    reserved = data_string[13];
    //Verify checksum
    check_sum = data_string[2] + data_string[3] + data_string[4] + data_string[5]
        + data_string[6] + data_string[7] + data_string[8] + data_string[9]
        + data_string[10] + data_string[11] + data_string[12] + data_string[13];

    if((check_sum != data_string[14]) {
        printf("Checksum mismatch error");
        return false;
    }

    //Scale and store data
    gRate = rates / 100.0;
    gAngle = angle / 100.0;
    gX_acc = x_acc;
    gY_acc = y_acc;
    gZ_acc = z_acc;
    return true;
}
```

## 4.6 漂移校准步骤

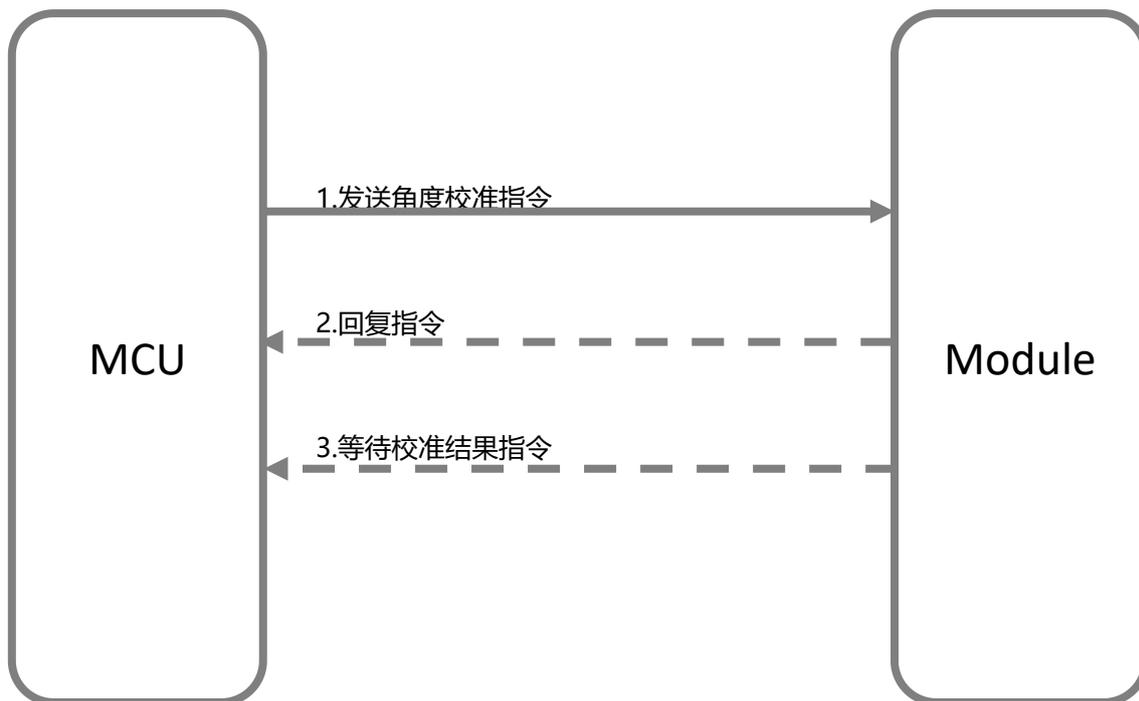


图 6 校准步骤

1. 发送角度校准命令：0xFF 0xFE 0x00 0x09
2. 等待返回命令：0xFF 0xFE 0x03 0x09
3. 等待并获得结果响应命令：
  - 如果校准成功则返回：0xFF 0xFE 0x01 0x09
  - 如果尝试失败则返回：0xFF 0xFE 0x02 0x09
 整个校准过程需要 3-15 秒。

## 4.7 旋转平面切换

由于不同安装位置的需求需要对模块的旋转平面做适度的调整，可以通过以下指令切换 X,Y,Z 三个轴为转轴。指令设置后配置会保存，下次启动配置依然生效。

1. X 轴：发送 0xFF 0xFE 0x00 0x11 回复 0xFF 0xFE 0x03 0x11
2. Y 轴：发送 0xFF 0xFE 0x01 0x11 回复 0xFF 0xFE 0x03 0x11
3. Z 轴：发送 0xFF 0xFE 0x02 0x11 回复 0xFF 0xFE 0x03 0x11

## 4.8 振动档位硬件配置

拨码位置	CODE	
	对应管脚P1	对应管脚P2
0	1	1
1	0	1
2	1	0
3	0	0
0: 低电平输出		
1: 高电平输出		

图 7 档位选择

## 4.9 振动档位阈值修改

档位 1:

查询:

以 16 进制发送: 0xFF 0xFE 0xB1 0x00 返回: 0xFF 0xFE 0xB0 0x01 第四位 0x01 则为设置档位值实际阈值为: 档位值\*16mg。

设置:

以 16 进制发送: 0xFF 0xFE 0xB1 0xff, 其中第四位 0xff 为配置的值, 范围为 0x01-0xff, 对应的阈值为设定值乘 16mg, 默认第一档阈值为 400mg。

例: 配置 0x01 则第一档设置值为 16mg

发送: 0xFF 0xFE 0xB1 0x01

回复: 0xFF 0xFE 0xB0 0x01

档位 2:

查询:

以 16 进制发送: 0xFF 0xFE 0xB2 0x00 返回: 0xFF 0xFE 0xB0 0x01 第四位 0x01 则为设置档位值实际阈值为: 档位值\*16mg。

设置:

以 16 进制发送: 0xFF 0xFE 0xB2 0xff, 其中第四位 0xff 为配置的值范围为 0x01-0xff, 对应的阈值为设定值乘 16mg, 默认第二档阈值为 1000mg。

例: 配置 0x02 则第二档设置值为 32mg

发送: 0xFF 0xFE 0xB2 0x02

回复: 0xFF 0xFE 0xB0 0x02

档位 3:

查询:

以 16 进制发送: 0xFF 0xFE 0xB3 0x00 返回: 0xFF 0xFE 0xB0 0x01 第四位 0x01 则为设置档位值实际阈值为: 档位值\*16mg。

设置:

以 16 进制发送: 0xFF 0xFE 0xB3 0xff, 其中第四位 0xff 为配置的值范围为 0x01-0xff, 对应的阈值为设定值乘 16mg, 默认第三档阈值为 1250mg。

例：配置 0x03 则第三档设置值为 48mg

发送：0xFF 0xFE 0xB3 0x03

回复：0xFF 0xFE 0xB0 0x03

档位 4:

查询:

以 16 进制发送：0xFF 0xFE 0xB4 0x00 返回：0xFF 0xFE 0xB0 0x01 第四位 0x01 则为设置档位值实际阈值为：档位值\*16mg。

设置:

以 16 进制发送：0xFF 0xFE 0xB4 0xff，其中第四位 0xff 为配置的值范围为 0x01-0xff，对应的阈值为设定值乘 16mg，默认第四档阈值为 2000mg。

例：配置 0x04 则第四档设置值为 64mg

发送：0xFF 0xFE 0xB4 0x04

回复：0xFF 0xFE 0xB0 0x04

阈值可设定范围：1~255，单位：16mg，最大值：4080mg。

## 5 修订历史

表 8 文档修订清单

日期	版本	变更
22-06-2020	1.0	Initial release