

# PAJ-7620 用户手册

——手势识别模块

修订历史

日期	版本	更新内容
2018/5/2	1.0.0	

## 文档说明

本手册旨在帮助用户正确构建 PAJ-7620 模块的使用环境，引导用户快速使用该模块。

关于核心模块 PAJ-7620 的硬件参数请参考文档《PAJ7620U2\_Datasheet\_V0.8.pdf》

关于核心模块 PAJ-7620 模块原理图请参考文档《XC6206P332MR\_PDF\_C14255\_2013-06-25.pdf》

# 目录

PAJ-7620 用户手册.....	1
文档说明.....	2
目录 .....	3
1. PAJ-7620 模块说明 .....	4
1.1 PAJ-7620 简介 .....	4
1.2 产品特性参数.....	4
1.3 PAJ-7620 模块的引脚说明 .....	5
1.4 PAJ-7620 传感器工作原理 .....	5
1.5 手势操作说明.....	6
2. 使用单片机系统控制 PAJ-7620 模块 .....	7
2.1 通用控制说明.....	7
2.1.1 硬件资源.....	7
2.1.2 开发板引脚连接.....	8
2.2 野火 STM32 开发板控制说明.....	8
2.2.1 连接模块.....	8
2.2.2 程序简介 .....	10
3. 代码分析.....	12
3.1.1 实验描述及工程文件清单.....	12
3.1.2 运行流程.....	12
3.1.3 手势识别传感器的测试.....	13
4. 常见问题.....	16
5. 产品更新及售后支持.....	17

# 1. PAJ-7620 模块说明

## 1.1 PAJ-7620 简介

PAJ-7620 是野火设计的高性能、低功耗手势识别传感器模块。它采用了原相科技公司的 PAJ-7620U2 为核心的处理芯片，芯片内部集成了光学数组式传感器单元，可以快速准确的对输入信号进行感应和输出处理。它支持上、下、左、右、前、后、顺时针旋转、逆时针旋转和挥动的手势动作识别，以及支持物体接近检测等功能。操作方便其外观见图 1-1。



图 1-1 PAJ-7620 手势识别传感器模块

## 1.2 产品特性参数

PAJ-7620 模块产品特性参数见表 1-1。

表格 1-1 PAJ-7620 模块产品特性

特性	说明
IIC 接口	支持高达 400Khz 通信速率
手势识别功能可识别类型	上、下、左、右、前、后、顺时针旋转、逆时针旋转、挥动
接近检测功能	可检测物体体积大小和亮度
传感器类型	CMOS
电源输入电压范围	2.8V ~ 4.0V
最大输出电流（SCL, SDA, INT_N）	5 mA
工作温度	-20C ~ 70 C

### 1.3 PAJ-7620 模块的引脚说明



图 1-2 PAJ-7620 模块引脚图

表格 1-2 PAJ-7620 模块引脚说明

编号	名称	说明
1	VCC	电源正(4.6V-6.0V)
2	GND	地线
3	SDA	IIC 数据脚
4	SCL	IIC 时钟脚
5	INT	普通 I/O
6	NC	空

### 1.4 PAJ-7620 传感器工作原理

PAJ7620U2 是一款光学数组式传感器，内置光源和环境光抑制滤波器集成的 LED，镜头和手势感测器在一个小的立方体模组，能在黑暗或低光环境下工作。同时传感器内置手势识别，支持 9 个手势类型和输出的手势中断结果。并且内置接近检测功能，可用于感测物体接近或离开。

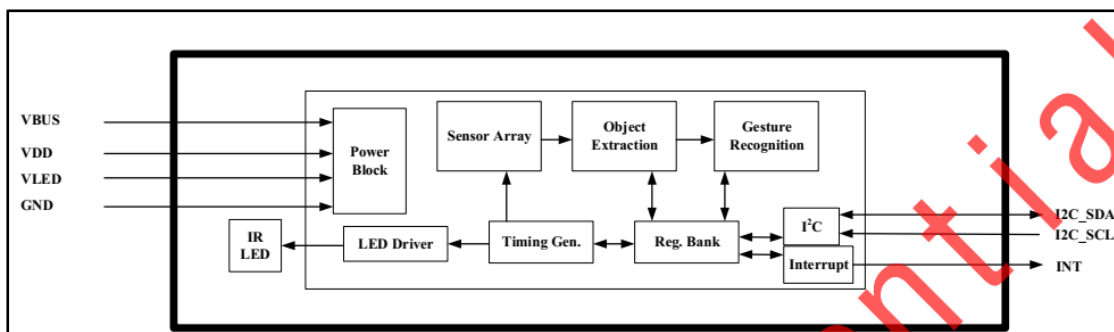


图 1-3 手势识别传感器模块工作原理图

传感器内部自带 LED 驱动器，传感器感应阵列、目标信息提取阵列和手势识别阵列。传感器工作时通过内部 LED 驱动器，驱动红外 LED 向外发射红外线信号，当传感器阵列在有效的距离中探测到物体时，目标信息提取阵列会对探测目标进行特征原始数据的获取，

获取的数据被保存在寄存器中，同时手势识别阵列会对原始数据进行识别处理，最后将手势结果存到寄存器中，用户可根据 I2C 接口对原始数据和手势识别的结果进行读取。

## 1.5 手势操作说明

传感器可识别“上”、“下”、“左”、“右”、“前”、“后”、“挥动”“顺时针旋转”、“逆时针旋转”共 9 种手势。其中传感器的正方向放置方式如图 1-4。

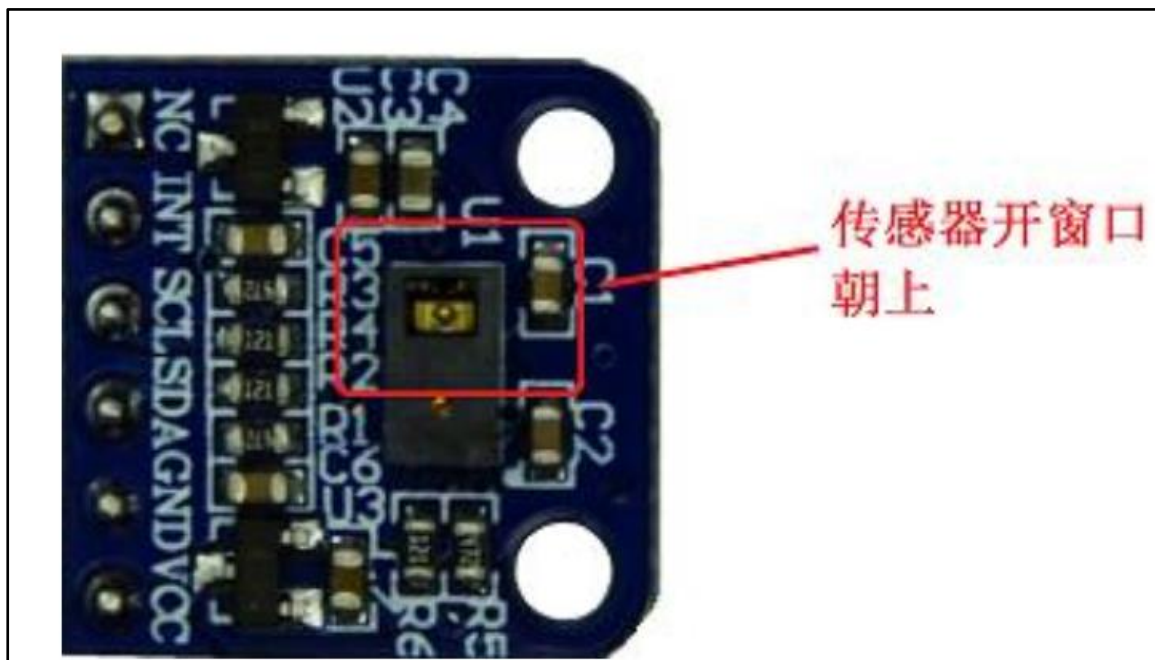


图 1-4 传感器正方向放置（面朝用户）

将手放置在传感器前方，分别作出如图 1-5 的手势，传感器可分别识别出“上”、“下”、“左”、“右”、“顺时针旋转”、“逆时针旋转”、“挥动”的识别结果。

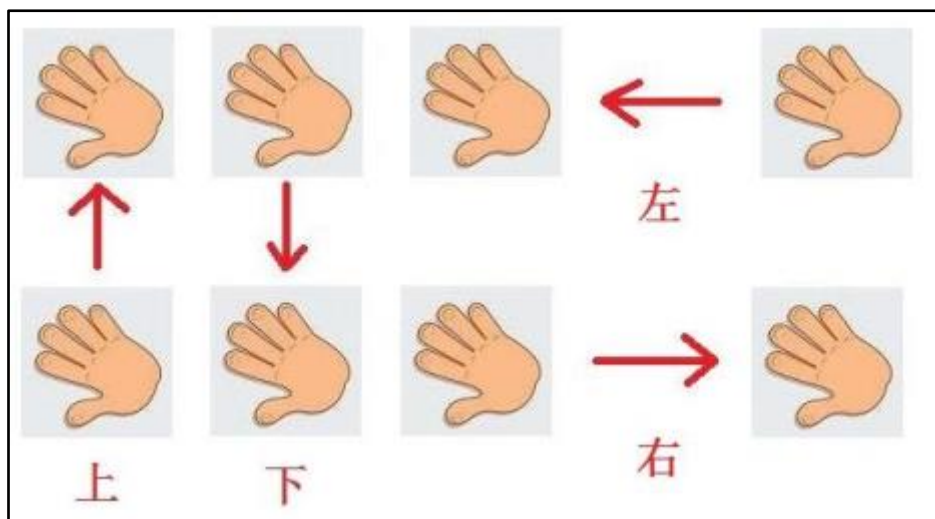


图 1-5 手势方向

另外，当在传感器的正前方靠近和远离传感器时如图 1-6，传感器可识别出“前”、“后”的识别结果。

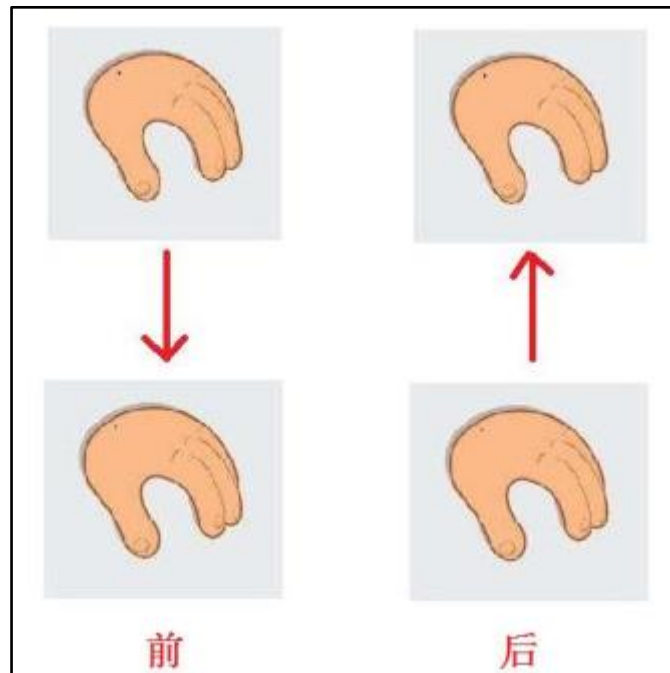


图 1-6 前后手势

## 2. 使用单片机系统控制 PAJ-7620 模块

### 2.1 通用控制说明

PAJ-7620 采用了 IIC 通信协议。本小节以野火 STM32 开发板为例子说明如何使用 STM32 与 PAJ-7620 模块手势识别实验电路连接。

#### 2.1.1 硬件资源

该试验所需要的硬件资源：

1. 计算机 1 台
2. 野火开发板 1 个
3. PAJ7620 模块 1 个
4. 野火下载器 1 个（带 USB 线）
5. USB 线



## 2.1.2 开发板引脚连接

单片机系统通过 IIC 引脚与 PAJ-7620 模块连接，与模块连接时，只要通过模块引出的排针连接好如下四根线即可，见错误!未找到引用源。

表格 2-1 单片机与 PAJ-7620 模块连接引脚表

单片机系统	PAJ-7620 模块
5V	VCC
GND	GND
IO 引脚	SDA
IO 引脚	SCL

## 2.2 野火 STM32 开发板控制说明

PAJ-7620 模块配套有适用于野火 STM32 开发板的源码，用户可以参考它来编写自己的应用。

### 2.2.1 连接模块

野火 F103 霸道、F103 指南者、MINI 及 F407 霸天虎以及 F429 挑战者板子配套的例程，都是通过 STM32 的 IIC 通道与 PAJ-7620 模块连接的。

表格 2-2 全系列开发板与 PAJ-7620 模块连接

全系列开发板	PAJ-7620 模块	引脚编号
VCC (5V)	VCC	1
GND	GND	2
PB11	SDA	3
PB10	SCL	4

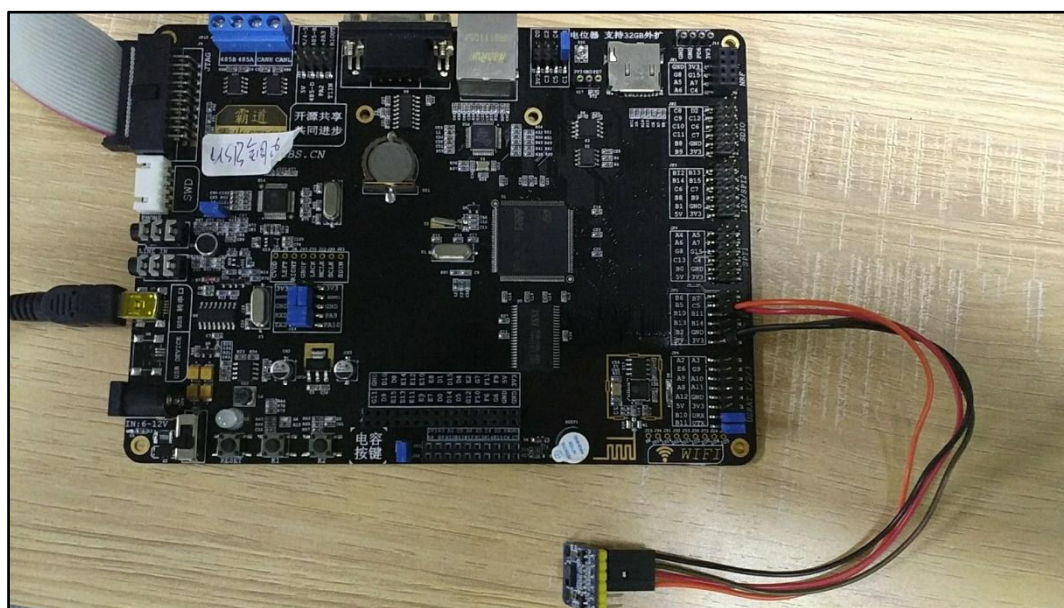


图 2-1 F103 霸道开发板接线图



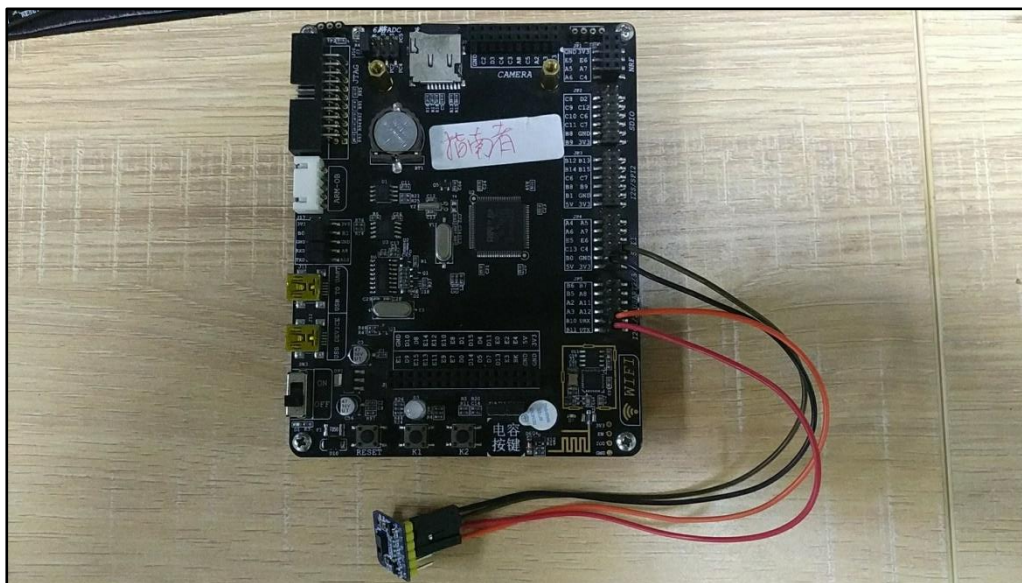


图 2-2 F103 指南者开发板接线图

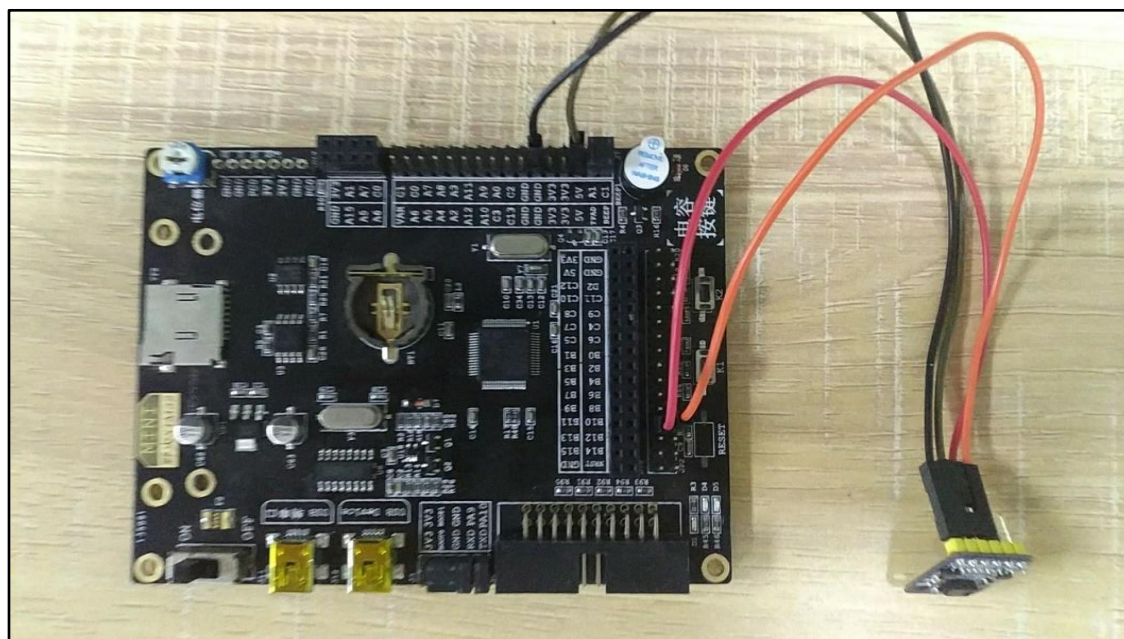


图 2-3 F103 MINI 开发板接线图

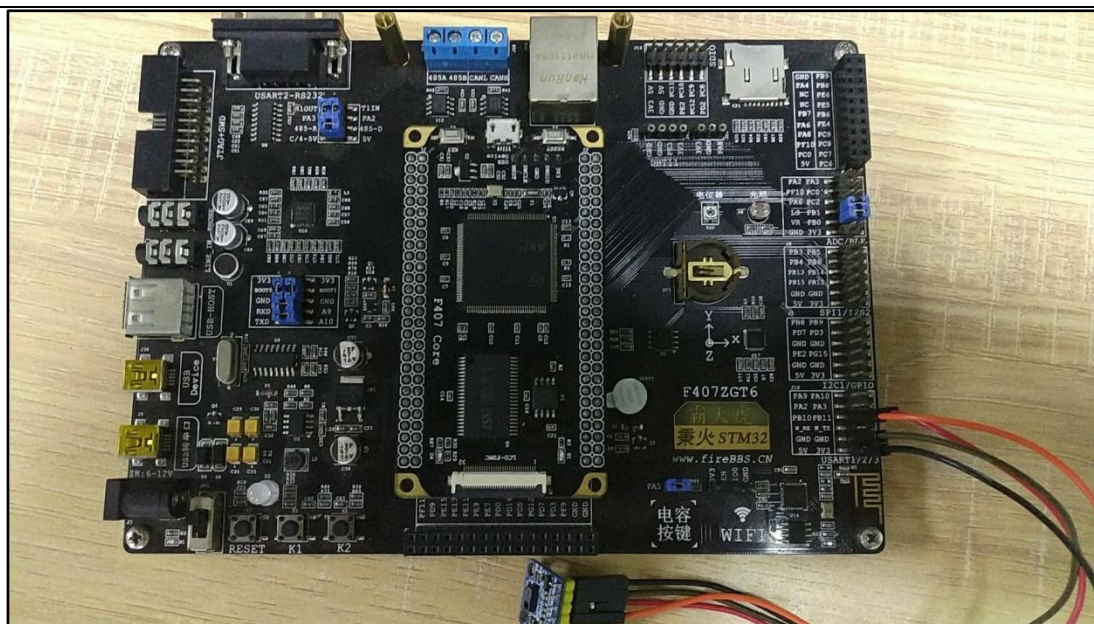


图 2-4 F407 霸天虎开发板接线图

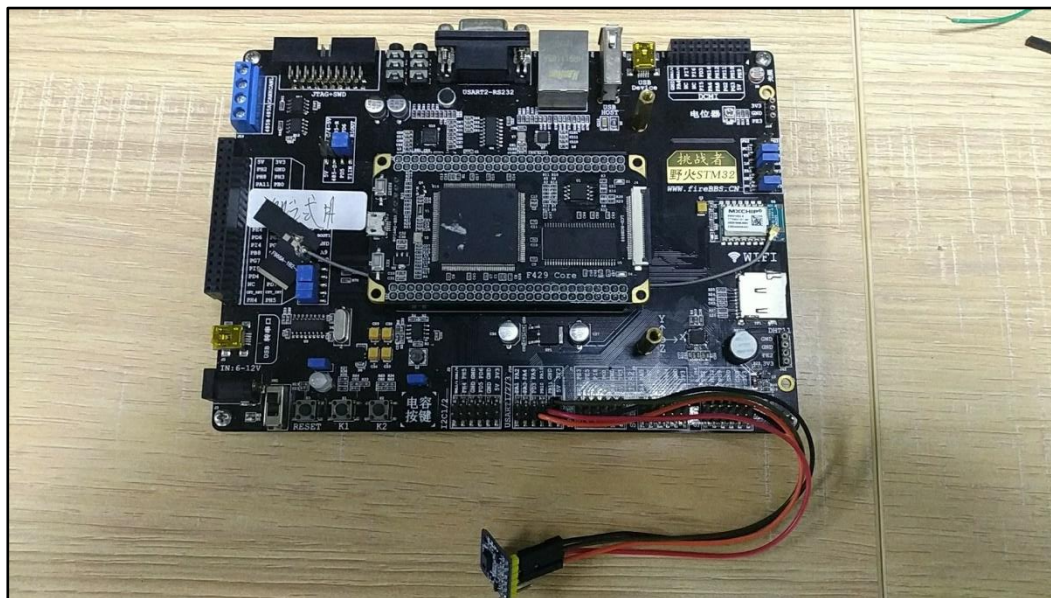


图 2-5 F429 挑战者开发板接线图

## 2.2.2 程序简介

下面以 F103 系列开发板的程序为例进行介绍，F4 的代码类似。

解压野火 PAJ-7620 资料后，在如下路径可以找到配套各个开发板的例程：PAJ-7620 手势识别传感器模块\2-开发板配套例程。

当手势识别传感器工作中识别到有效信息后，通过 IIC 通信方式向开发板传输信息，并将处理结果通过串口调试助手显示了处理结果。实现现象如图 2-6、图 2-7。



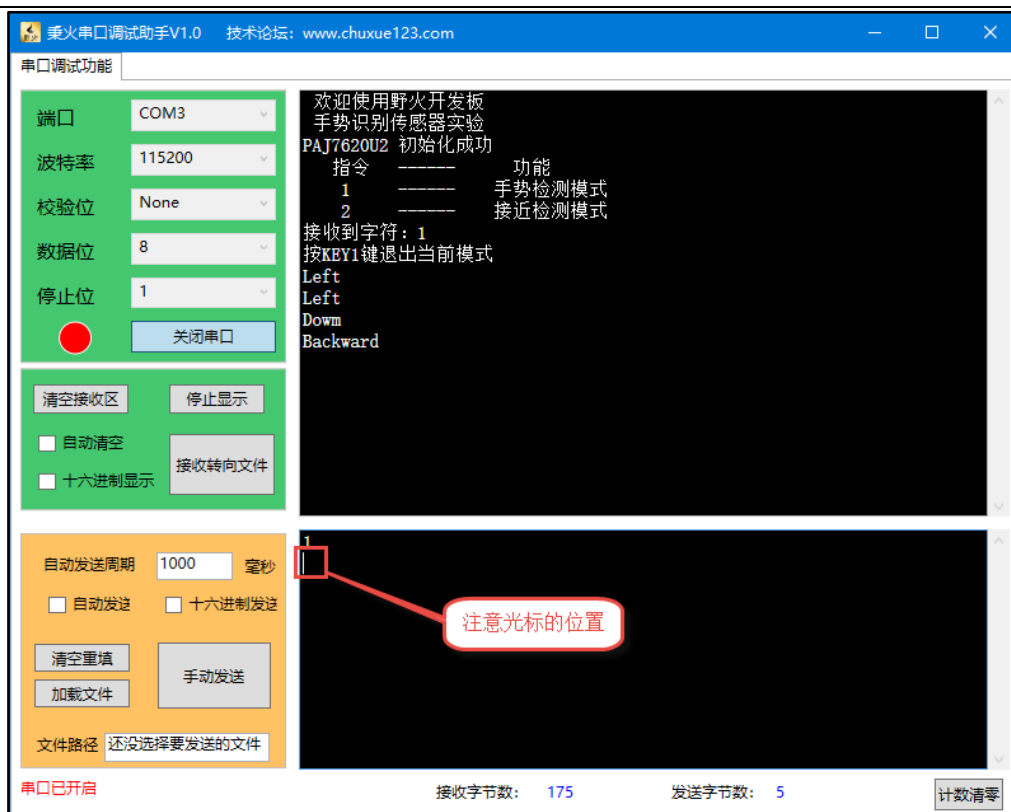


图 2-6 手势识别模式

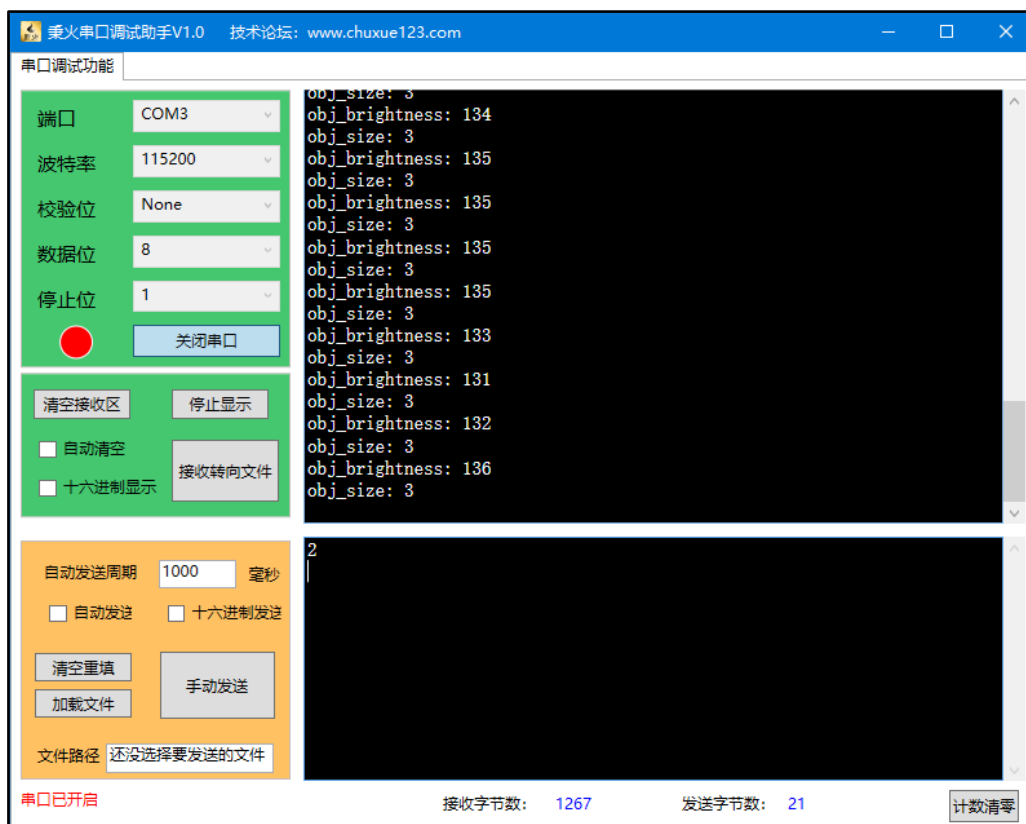


图 2-7 接近检测模式

## 3. 代码分析

在本小节中我们将分析如何使用手势传感器进行数据采集以及如何使用 KEY (EXTI)中断方式对工作模式进行切换的。

### 3.1.1 实验描述及工程文件清单

#### 1. 实验描述

手势传感器获取感应信息，将结果通过开发板 usart1 串口输给串口调试助手程序显示出来。

#### 2. 主要工程文件

表格 3-1 文件清单

工程名称	PAJ-7620
库文件	Libraries
用户编写的文件	USER/main USER/paj7620u2 USER/SysTick USER/usart USER/ Key

### 3.1.2 运行流程

#### 1. 主函数与初始化配置

以下按照程序的执行流程，从 main 文件开始分析，见代码清单 3-1。

代码清单 3-1 main 函数（main.c 文件）

```
1  /**
2   * @brief 主函数
3   * @param 无
4   * @retval 无
5   */
6  int main(void)
7  {
8      /*初始化 USART 配置模式为 115200 8-N-1，中断接收*/
9      USART_Config();
10     EXTI_Key_Config();
11
12     printf(" 欢迎使用野火开发板\r\n");
13     printf(" 手势识别传感器实验\r\n");
14
15     while (!paj7620u2_init()) { //PAJ7620U2 传感器初始化
16         printf("PAJ7620U2 初始化失败!!!\r\n");
17         delay_ms(600);
18     }
19     printf("PAJ7620U2 初始化成功\r\n");
20     while (1) {
21         paj7620u2_sensor_test(); //PAJ7620U2 传感器测试
```

```
22     }
23 }
```

在 main 函数中，首先对 USART 串口与 KEY 按键进行初始化配置。然后对传感器 paj7620u2 进行初始化配置。当对传感器初始化成功后则退出 while 循环，进入第 21 行 paj7620u2 传感器功能测试阶段。

代码清单 3-2 PAJ7620U2 传感器测试（PAJ7620U2.c 文件）

```
1 /**
2  * @brief PAJ7620U2 传感器测试
3  * @param 无
4  * @retval 无
5  */
6 void paj7620u2_sensor_test(void)
7 {
8     uint32_t  ch;
9
10    Show_Mode_Message(); //显示普通测量模式 UI
11    while (1) {
12        scanf("%d", &ch);
13        printf("接收到字符: %d\r\n", ch);
14        switch (ch) {
15            case 1:
16                Gesrure_test();
17                break; //手势检测模式
18            case 2:
19                Ps_test();
20                break; //接近检测模式
21            default:
22                printf("请输入合法指令! \r\n");
23                break;
24        }
25        Show_Mode_Message();
26    }
27 }
28 }
```

在 test 函数中，第 12 行使用 scanf 函数，使开发板获取来自串口调试助手发出的控制指令。当开发板接收到的指令字符为“1”时，开发板控制传感器进入手势识别工作模式；当接收到的指令字符为“2”时，开发板控制传感器进入接近检测工作模式；当接收的字符既不是“1”也不是“2”时，则打印提示用户输入合法指令。

### 3.1.3 手势识别传感器的测试

手势识别传感器可以工作在手势识别与接近检测 2 种模式状态下。为了进行工作模式的切换，我们采用了 KEY 按钮外部中端的方式。

#### 1. KEY 外部中断

当我们想退出当前的工作模式时，可通过按住 key1 按钮，激发产生 EXTI 中断响应。中断代码如下。

代码清单 3-3 KEY1 中断函数（stm32f10x\_it.c 文件）

```
1 /**
2  * @brief KEY1 中断函数
3  * @param 无
4  * @retval 无
```

```

5  */
6 void KEY1_IRQHandler(void)
7 {
8     //确保是否产生了 EXTI_Line 中断
9     if (EXTI_GetITStatus(KEY1_INT_EXTI_LINE) != RESET) {
10         flag=1;
11         EXTI_ClearITPendingBit(KEY1_INT_EXTI_LINE);
12     }

```

在 IRQHandler 中断处理函数中，（第 9 行）当接受到中断信号时，变量 flag 赋值为 1。

## 2. 手势识别工作模式

Gesrure\_test()函数，具体代码如下：

代码清单 3-4 Gesrure\_test 函数（paj7620u2.c 文件）

```

1  /**
2   * @brief  手势识别测试
3   * @param  无
4   * @retval 无
5   */
6 void Gesrure_test(void)
7 {
8     uint8_t status,i;
9     uint8_t data[2]= {0x00};
10    uint16_t gesture_data;
11
12    paj7620u2_selectBank(BANK0); //进入 BANK0
13    for (i=0; i<GESTURE_SIZE; i++) {
14        GS_Write_Byte(gesture_array[i][0],gesture_array[i][1]); //手势识别模式初始化
15    }
16    paj7620u2_selectBank(BANK0); //切换回 BANK0
17    printf("按 KEY1 键退出当前模式\r\n");
18
19    while (1) {
20        if (flag) {
21            flag=0;
22            GS_Write_Byte(PAJ_SET_INT_FLAG1,0X00); //关闭手势识别中断输出
23            GS_Write_Byte(PAJ_SET_INT_FLAG2,0X00);
24            break;
25        }
26        status = GS_Read_nByte(PAJ_GET_INT_FLAG1,2,&data[0]); //读取手势状态
27        if (!status) {
28            gesture_data =(uint16_t) data[1]<<8 | data[0];
29            if (gesture_data) {
30                switch (gesture_data) {
31                    case GES_UP:
32                        printf("Up\r\n");
33                        break; //向上
34                    case GES_DOWM:
35                        printf("Down\r\n");
36                        break; //向下
37                    case GES_LEFT:
38                        printf("Left\r\n");
39                        break; //向左
40                    case GES_RIGHT:
41                        printf("Right\r\n");
42                        break; //向右
43                    case GES_FORWARD:
44                        printf("Forward\r\n");

```

```

45         break; //向前
46     case GES_BACKWARD:
47         printf("Backward\r\n");
48         break; //向后
49     case GES_CLOCKWISE:
50         printf("Clockwise\r\n");
51         break; //顺时针
52     case GES_COUNT_CLOCKWISE:
53         printf("AntiClockwise\r\n");
54         break; //逆时针
55     case GES_WAVE:
56         printf("Wave\r\n");
57         break; //挥动
58     default:
59         break;
60     }
61 }
62 }
63 }
64 }

```

在 `Gesrure_test()` 函数中，首先对 PAJ7620U2 手势识别检测进行初始化，（第 14 行代码）配置其 `gesture_array` 数组的参数值。该数组有使能 9 个手势识别的中断标志输出的配置，初始化完毕后，在 `while` 循环中一直读取手势中断标志寄存器（第 26 行代码），当识别到相应的手势，手势标志会置 1，读取寄存器标志会自动清 0。根据手势的标志手势结果将被打印到串口上。当按下 `KEY` 按键，在中断处理函数中标志变量 `flag` 被赋值 1（第 20 行代码），此时关闭 9 个手势识别的检测输出，退出当前的手势识别测试模式。

### 3. 接近检测测试

`Ps_test()` 函数，具体代码如下：

代码清单 3-5 `Ps_test` 函数（`paj7620u2.c` 文件）

```

1  /**
2   * @brief 接近检测测试
3   * @param 无
4   * @retval 无
5   */
6  void Ps_test(void)
7  {
8      uint8_t i;
9      uint8_t data[2] = {0x00};
10     uint8_t obj_brightness=0;
11     uint16_t obj_size=0;
12
13     paj7620u2_selectBank(BANK0); //进入 BANK0
14     for (i=0; i<PROXIM_SIZE; i++) {
15         GS_Write_Byte(proximity_array[i][0], proximity_array[i][1]);
16         //接近检测模式初始化
17     }
18     paj7620u2_selectBank(BANK0); //切换回 BANK0
19     printf("按 KEY1 键退出当前模式\r\n");
20
21     while (1) {
22         if (flag) {
23             flag=0;
24             break;
25         }
26     }
27 }

```



```
26     }
27     obj_brightness =
        GS_Read_Byte(PAJ_GET_OBJECT_BRIGHTNESS); //读取物体亮度
28     data[0] = GS_Read_Byte(PAJ_GET_OBJECT_SIZE_1); //读取物体大小
29     data[1] = GS_Read_Byte(PAJ_GET_OBJECT_SIZE_2);
30     obj_size = ((uint16_t)data[1] & 0x0f) << 8 | data[0];
31     printf("obj_brightness: %d\r\n", obj_brightness);
32     printf("obj_size: %d\r\n", obj_size);
33
34     delay_ms(500);
35 }
36 }
```

`Ps_test()` 函数实现接近检测，获取接近物体的体积大小和亮度的数据。首先对 PAJ7620U2 接近检测进行初始化，配置其 `proximity_array` 的数组参数值（第 16 行代码），初始化完成后，在 `while(1)` 循环中，读取获取体积大小和亮度的寄存器，并将读取结果打印到串口上。当按下 `KEY` 按键，在中断处理函数中标志变量 `flag` 被赋值 1（第 23 行代码），退出当前的接近检测测试模式。

## 4. 常见问题

1. Q: 为什么在野火串口调试助手有时输入指令（如 1）后发送不出去？

答：因为输入数据时调用 `scanf` 库函数，由于 `scanf` 函数功能特性，往往需要先输入指令后，在输入窗口打回车键，才能将指令发送出去。

2. Q: 为什么我接到 3.3V 电压源，传感器模块工作不正常？

答：传感器的驱动电压较高，正常的工作电压为 5V，3.3V 无法正常驱动传感器模块。

## 5. 产品更新及售后支持

野火公司官网: <http://www.embedfire.com>

野火电子论坛: <http://www.firebbs.cn>

野火淘宝店铺: <http://fire-stm32.taobao.com>

在学习或使用野火产品时遇到问题可在论坛发帖子与我们交流。