

4\*4 矩阵按键使用说明

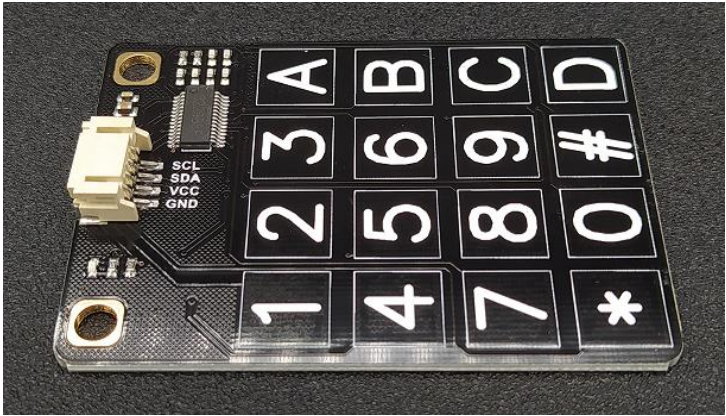
--2022/10/20

别墅套件根据实际用户的使用反馈不断更新升级模块与程序，以追求更好的教学效果和使用体验，并提供定制版本的教学套件。

	实物	协议	细节
V1.0	触摸按键	类 I2C	上电时需静置 5S, 等待内部芯片矫正
V2.0	实体物理 按键	标准 I2C	上电时直接使用，与常用 I2C 模块一样

V1.0 触摸版本（22 年 5 月前使用）

实物图片



- 1、

TTP229初始化 引脚SCL A2 引脚SDA A3

设置 TTP229 模块的引脚 初始引脚为 A2 ， A3(如果选择 A5， A4 引脚时，不可以在使用其他 IIC 模块)；
- 2、

TTP229设置密码 1 2 3

设置 TTP229 模块的密码，和下面的验证密码的模块一起用（不设置密码，初始密码为 123）；

3、

### TTP229 读按键值(字符串)

返回 TTP229 模块的按键值，返回 (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, A, B, C, D, \*, #)，返回值以字符串格式；



实物效果：按下按键 2 时，D13(板载灯)亮起；否则，D13 熄灭。

4、

### TTP229 密码判断？

密码判断模块，密码正确返回 1，错误返回 2，未调用或没有输入到 3 位时返回 0，一般结合 if 函数使用；**注意：**由于本模块还未有提示已经输入了几位的功能，或者用户在使用时，经常误触，建议使用本密码判断语句时，别墅先断电，再上电。



实物效果：密码输入正确时，D13(板载灯)亮 1S。

5、

### TTP229 按键 1 状态按下？

判断单个按键的状态 按下返回 1， 未按下返回 0。



实物效果：按下按键 5 时，D13(板载灯)亮起；否则，D13 熄灭。

## 6、日常使用注意事项：

如果 4\*4 矩阵触摸按键连接到 UNO 上的硬件 I2C 引脚时，无法与硬件 I2C 功能同时使用。上电后，需等待 5S，4\*4 矩阵触摸按键内部初始后，增加灵敏度，方可使用，能明显减少误触概率。

## 7、模块没反应？密码总是按错？

课程中的任何模块如果使用起来是有问题，先拔掉其他模块，减少干扰，单独测试本问题模块。



(D13 为主板上数字 13 引脚上的板载 LED 小灯)

## V2.0 实体物理版本（2022 年 5 月开始出货）

### 实物图片



模块属于 I2C 模块，直接连接到 I2C 接口（A4/A5）就能使用。

1、增加别墅套件的专用图形化库：集合了 V1.0 与 V2.0



2、

设置新密码用，一般放初始化这里；如果不重新设置，模块密码默认是 123。

3、

假如密码是 123 时，用户完整输入 123 时，密码判断正确；等待输入中，或者输入错误，都会判断错误。打开串口监视，就可以看到本次输入的按键值，可以用作窗口去查看当前输入的按键值；输入正确时，最后输出“OK”，输入错误时，最后输出“Wrong”。



实物效果：密码输入正确时，D13(板载灯)长亮。（ESP32 版本库会有不同效果，密码正确后，会立刻密码时效，相当于 D13 亮一下，就立刻熄灭）



4、



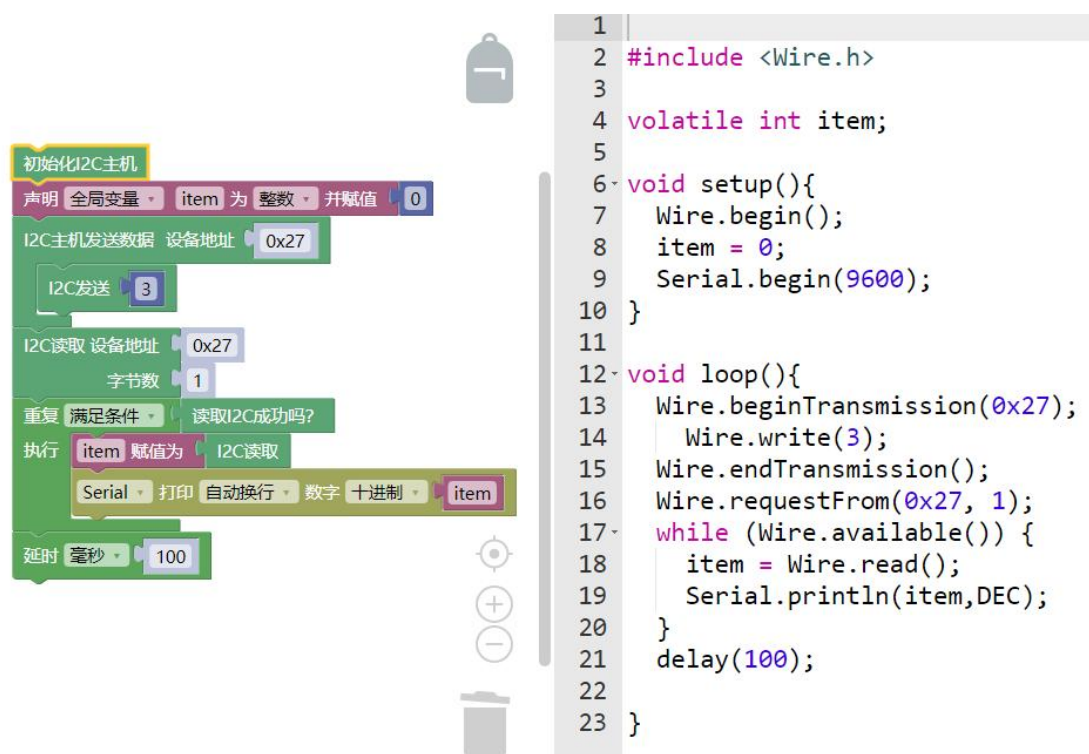
等待读一次按键值，例如按下 2，则返回一个数字 2，可理解为按键开关。返回值是数字：按下 1，就返回数字 1。A 对应 10，B 对应 11，C 对应 12，D 对应 13，\* 对应 14，# 对应 15，没按下，返回 16。



实物效果：当按下按键 2 时，D13(板载灯)长亮；松开按键 2 后，D13(板载灯)熄灭。

## 二、Mixly、arduino IDE 测试程序

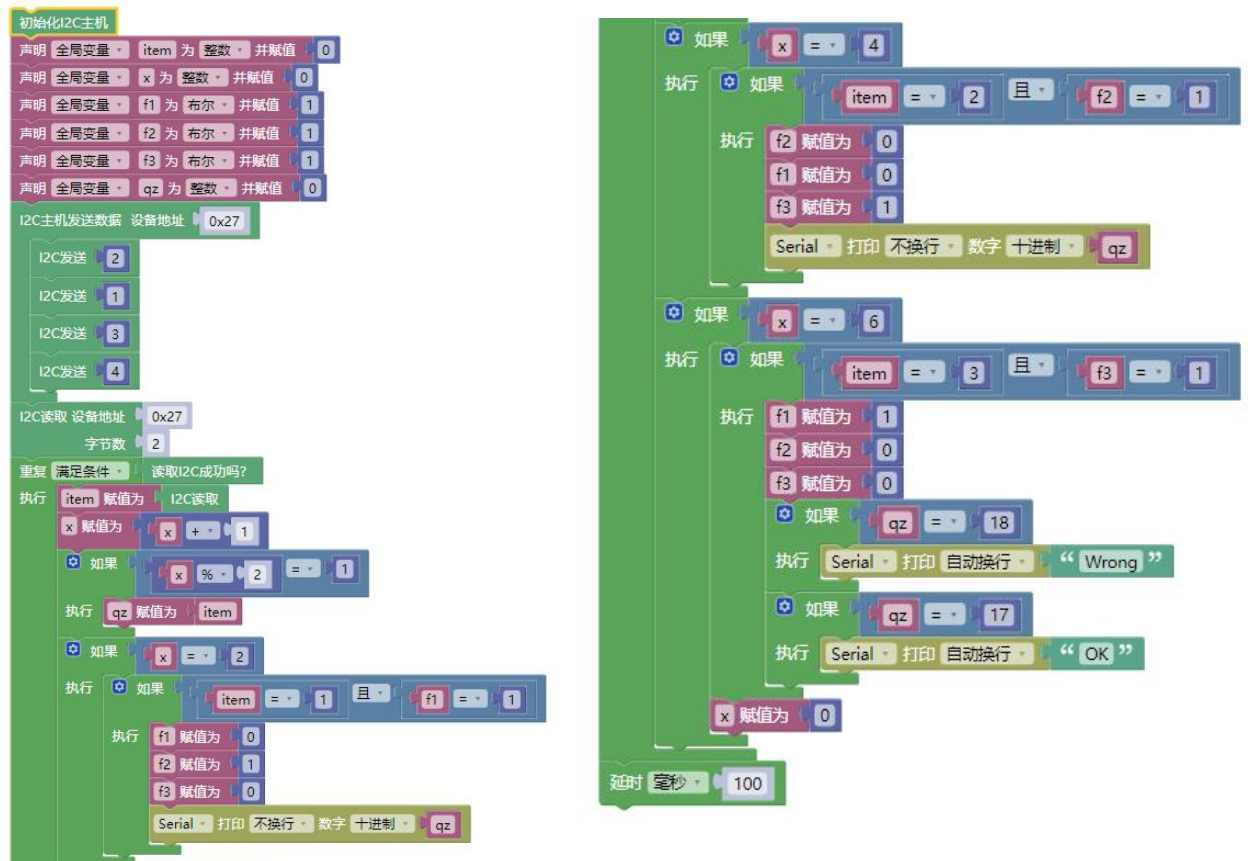
### 2.1、Mixly、arduino IDE 读单个键值



4x4 矩阵按键的 I2C 地址是 0x27，读单个键值的命令为 3，模块一致上传键值（包含按下与未按下）。

## 2.2、Mixly、arduino IDE 读密码

### 【Mixly 程序】



### 【Arduino IDE 程序】

```

1 #include <Wire.h>
2
3 volatile int item,x,qz;
4 volatile boolean f1,f2,f3;
5
6 void setup(){
7     Wire.begin();
8     item = 0;
9     x = 0;
10    f1 = 1;
11    f2 = 1;
12    f3 = 1;
13    qz = 0;
14    Serial.begin(9600);
15 }
16
17 void loop(){
18     Wire.beginTransaction(0x27);
19     Wire.write(2);
20     Wire.write(1);
21     Wire.write(3);
22     Wire.write(4);
23     Wire.endTransmission();
24     Wire.requestFrom(0x27, 2);
25     while (Wire.available()) {
26         item = Wire.read();
27         x = x + 1;
28         if ((long) (x) % (long) (2) == 1) {
29             qz = item;
30         }

```

```

31     if (x == 2) { //处理、并打印第一个输入的按键
32         if (item == 1 && f1 == 1) {
33             f1 = 0;
34             f2 = 1;
35             f3 = 0;
36             Serial.print(qz,DEC);
37         }
38     }
39     if (x == 4) { //处理、并打印第一个输入的按键
40         if (item == 2 && f2 == 1) {
41             f2 = 0;
42             f1 = 0;
43             f3 = 1;
44             Serial.print(qz,DEC);
45         }
46     }
47     if (x == 6) { //处理、并打印第一个输入的按键
48         if (item == 3 && f3 == 1) {
49             f1 = 1;
50             f2 = 0;
51             f3 = 0;
52             if (qz == 18) {
53                 Serial.println("Wrong");
54             }
55             if (qz == 17) {
56                 Serial.println("OK");
57             }
58         }
59         x = 0;
60     }
61 }
62 delay(100);
63 }

```