
keyes WiKi

keyes WiKi

2025 年 05 月 14 日

KE4045S Keyes DIY 电子积木自锁按键模块

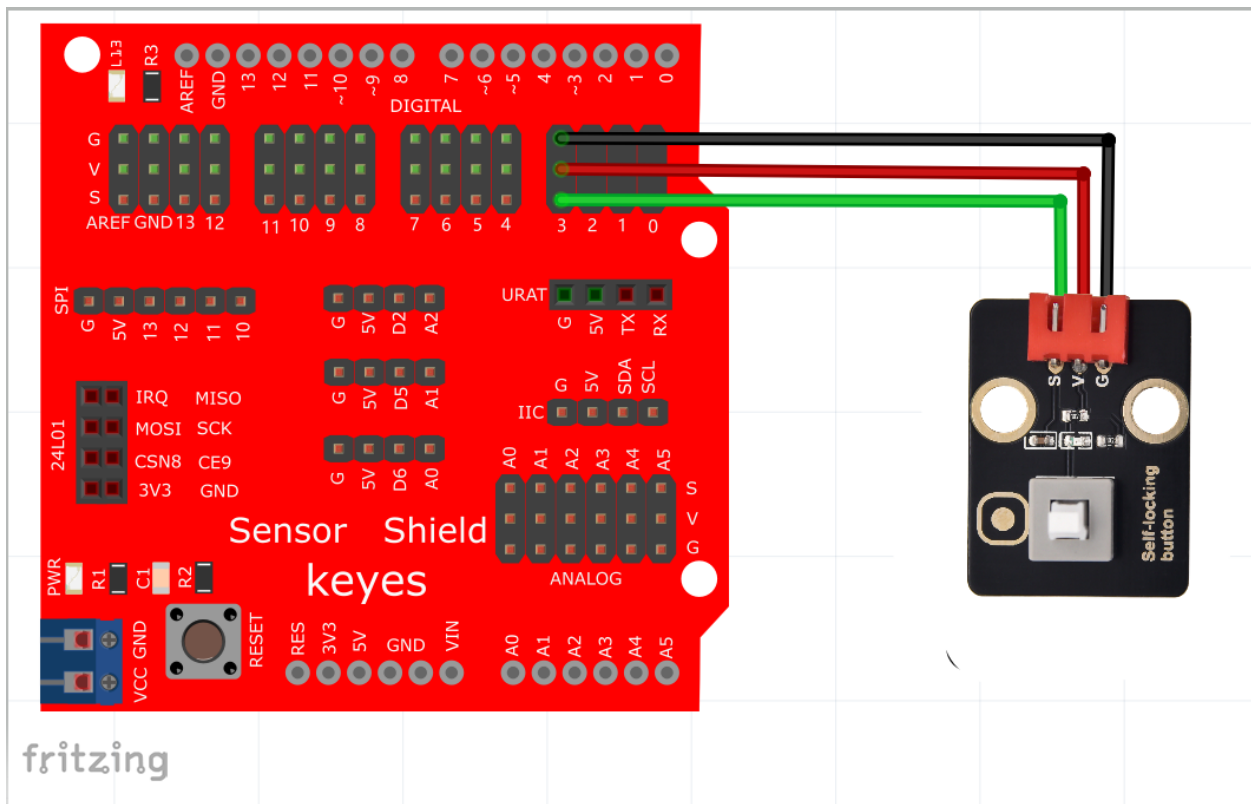
1	1. Arduino 简介	1
2	2. 接线图	3
3	3. 测试代码 (测试软件版本: Arduino 1.8.12)	5
4	4. 代码说明	7
5	5. 测试结果	9
6	KidsBlock	11
6.1	1. KidsBlock 简介	11
6.2	2. 接线图	12
6.3	3. 测试代码	14
6.4	4. 代码说明	15
6.5	5. 测试结果	15
7	Mixly	17
7.1	1. Mixly 简介	17
7.2	2. 接线图	18
7.3	3. 测试代码 (测试软件版本: Mixly 1.2.0)	19
7.4	4. 代码说明	19
7.5	5. 测试结果	20
8	Python	21
8.1	1. Python 简介	21
8.2	2. 接线图	21
8.3	3. 测试代码	22
8.4	4. 代码说明	22
8.5	5. 测试结果	22

CHAPTER 1

1. Arduino 简介

Arduino 是一种开源电子原型平台，旨在帮助用户进行互动项目的开发。它由硬件和软件两个部分组成，提供了丰富的开发环境和灵活的接口。Arduino 支持多种传感器和模块，用户可以通过简单的编程（基于 C/C++ 语言）构建各种项目。Arduino 的设计理念强调对初学者的友好性，提供了大量的教程和社区支持，使得学习电子原理和编程技术变得简单而有趣。

2. 接线图



3. 测试代码（测试软件版本：Arduino 1.8.12）

```
int ledPin = 13; // 定义数字口13
int inputPin = 3; // 定义数字口3

void setup() {
  pinMode(ledPin, OUTPUT); // 将ledPin设置为输出
  pinMode(inputPin, INPUT); // 将inputPin设置为输入
}

void loop() {
  int val = digitalRead(inputPin); // 读取数字口3的数值并赋值给 val
  if (val == LOW) { // 当val为低电平时，LED亮起
    digitalWrite(ledPin, HIGH); // LED亮起
  } else {
    digitalWrite(ledPin, LOW); // LED变暗
  }
}
```

4. 代码说明

1. `pinMode(button, INPUT);`
在这里我们定义按键管脚为数字口 2，设置为输入模式。通过 `pinMode()` 配置为 INPUT 的 Arduino (ATmega) 引脚处于高阻抗状态。配置为 INPUT 的引脚对采样电路的要求极小，便于读取传感器。
2. `Serial.begin(9600);`
初始化串口通信，并设置波特率为 9600。
3. `digitalRead(button);`
读取按键的数字电平，返回 HIGH 或 LOW，如果引脚未连接任何东西，则返回值可能会随机变化。
4. `if..else..` 语句：
当 `if` 后面的条件为真时，执行大括号里的代码；否则执行 `else` 后面 `{}` 里的代码。
5. 代码逻辑是传感器感应到按键按下时，信号端为低电平，D2 口为低电平，即 `val` 为 0，这时 LED 亮起；反之，当信号为高电平时，LED 熄灭。

CHAPTER 5

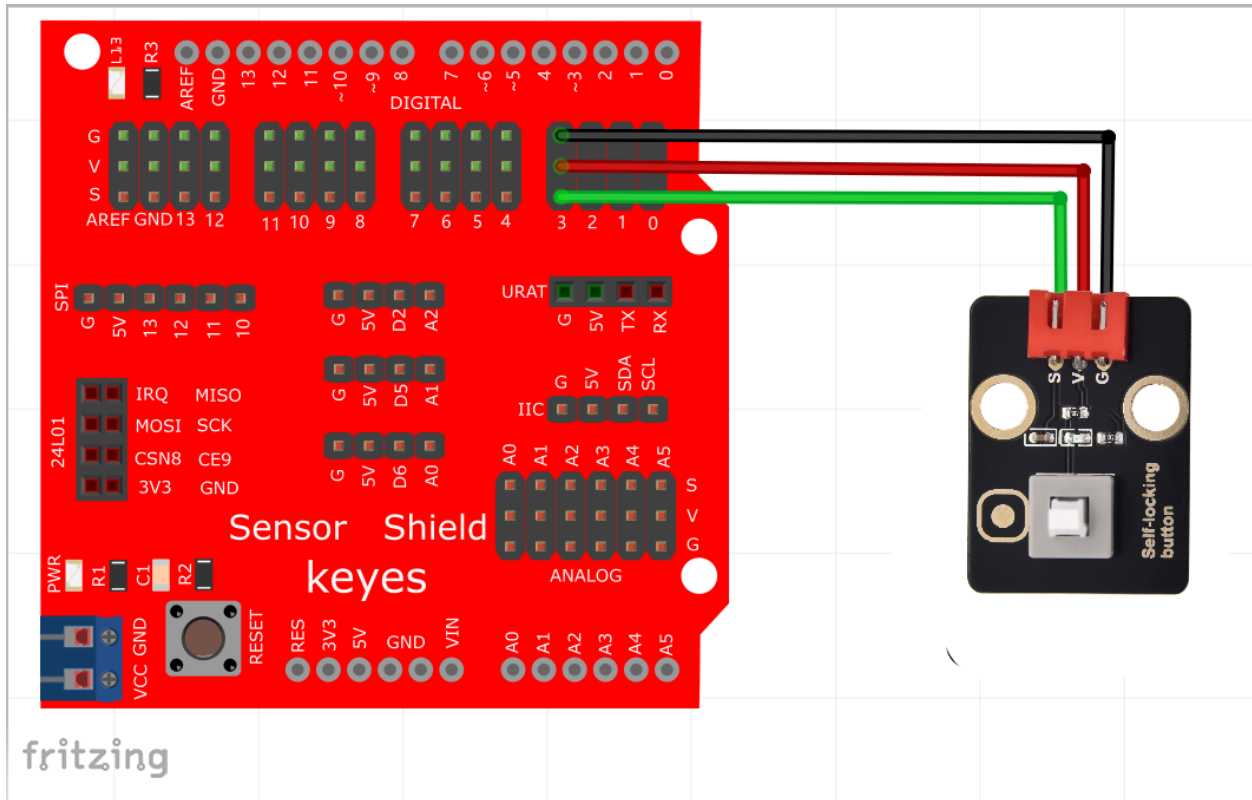
5. 测试结果

按照上图接好线，烧录好代码；上电后，按下按键后，板载 LED 亮起，放开按键后，LED 熄灭。

6.1 1. KidsBlock 简介

KidsBlock 是一款面向儿童和初学者的图形化编程学习平台，旨在通过简单易用的方式引导用户了解编程和电子原理。它结合了图形化编程界面和实际硬件，用户可以通过拖放模块创建代码，从而轻松控制各种电子设备和传感器。KidsBlock 支持多种编程语言，包括 Scratch 和 Arduino C，提供丰富的示例和教程，使得学习编程变得有趣且容易。该平台特别适合在教育环境中使用，帮助学生培养逻辑思维和创造力。

6.2 2. 接线图



6.3 3. 测试代码

The image shows a sequence of code blocks for testing an Arduino button. The blocks are as follows:

- 当Arduino启动** (When Arduino starts)
- 声明** (Declare): 全局 (Global), 数字变量 类型 (Integer), 名称 (button), 赋值为 (0)
- 串口启动波特率** (Serial port baud rate): 9600
- 设置引脚** (Set pin): 3, 模式 (Input)
- 重复执行** (Repeat):
 - 设置** (Set): button 变量为 (读取数字引脚 3)
 - 如果** (If): 变量 button = 0 那么
 - 串口打印** (Serial print): 不换行
 - 串口打印** (Serial print): Press the button 换行
 - 等待** (Wait): 0.1 秒
 - 否则** (Else):
 - 串口打印** (Serial print): 不换行
 - 串口打印** (Serial print): Loosen the button 换行
 - 等待** (Wait): 0.1 秒

6.4 4. 代码说明

1. 在变量类型单元中找到，将 item 变量名改为“button”，这个修改是为了更好地理解与记忆。设置 button 初始变量为整数，并赋值为 0。
2. 初始化设置波特率为 9600，表示串口通信的传输速度为 9600 位每秒。
3. 在引脚单元中找到，使该管脚设置为 3，并设置为输入模式，以读取按键的值。
4. 在控制单元中找到条件控制语句，如果…否则…，填入 button 和“0”以判断按键是否被按下。如果按键被按下，则输出相应的信息。
5. 整个代码逻辑是当传感器感应到按键按下时，button 为 0，串口监视器显示“Press the button”；松开按键时，button 为 1，显示“Loosen the button”。

6.5 5. 测试结果

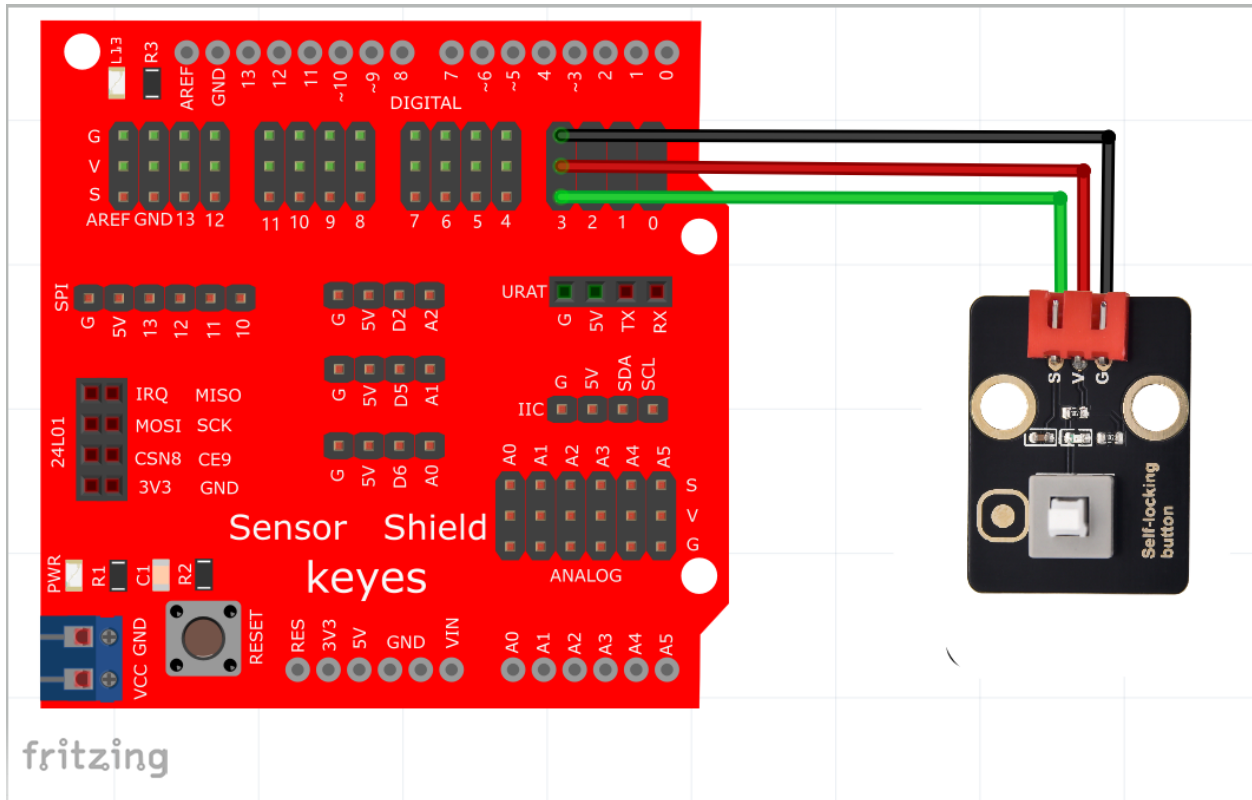
成功上传测试代码后，利用 USB 进行供电，串口监视器显示对应数据和字符。当按下按键时，button 为 0，显示“Press the button”；松开按键时，button 为 1，显示“Loosen the button”，如下图。



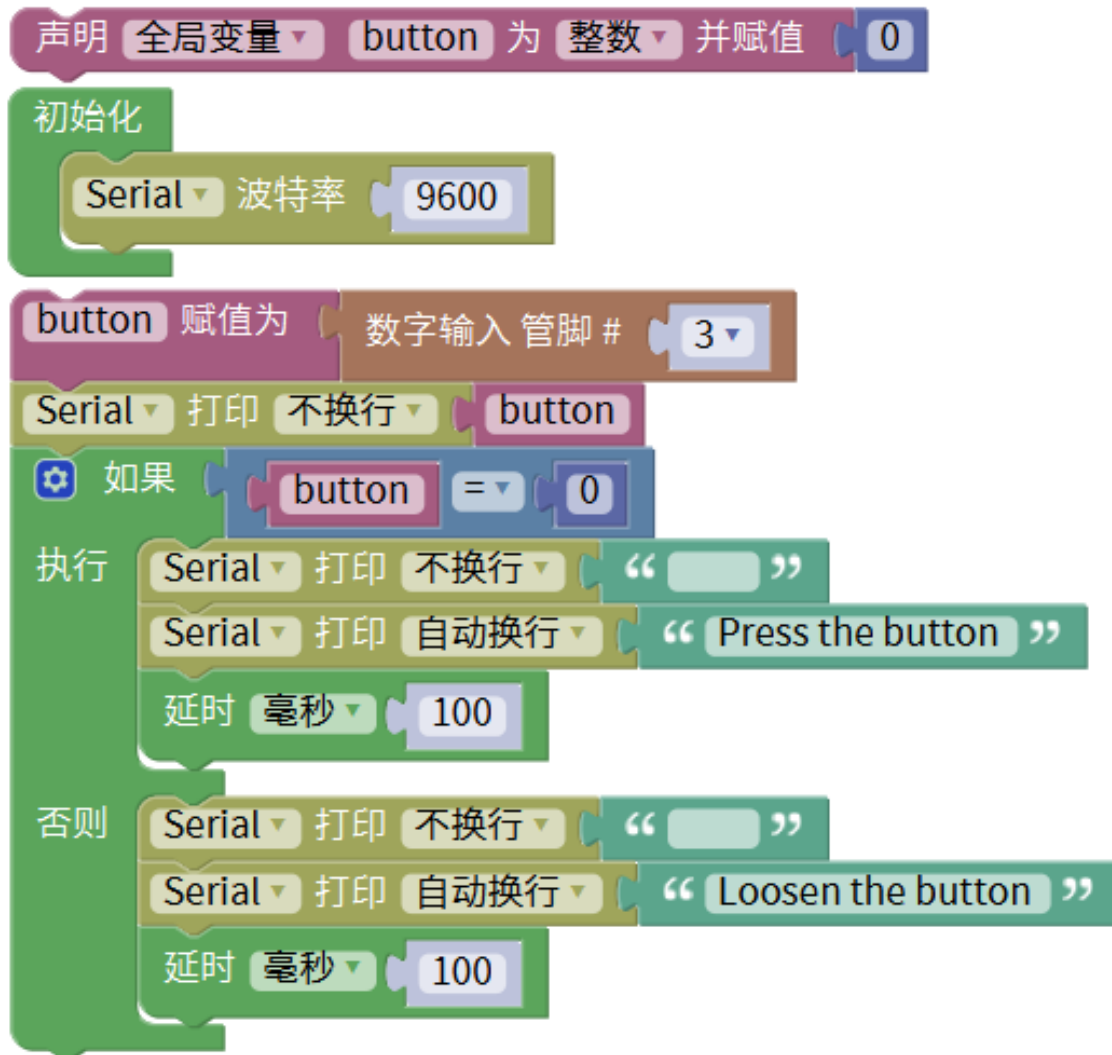
7.1 1. Mixly 简介

Mixly 是一款可视化编程软件，为用户提供了简单直观的方式来学习和使用编程。它适用于 Arduino、Micro:bit 等硬件平台，使得初学者能够通过拖放模块来构建程序，而无需深入了解编程语言的具体语法。此软件包含了丰富的内置库，支持多种控件和功能，帮助用户轻松实现各种项目，如传感器控制、灯光闪烁和用户交互等。Mixly 的设计使得教育工作者和学生在课堂上能够更有效地参与到编程和电子创作中。

7.2 2. 接线图



7.3 3. 测试代码（测试软件版本：Mixly 1.2.0）

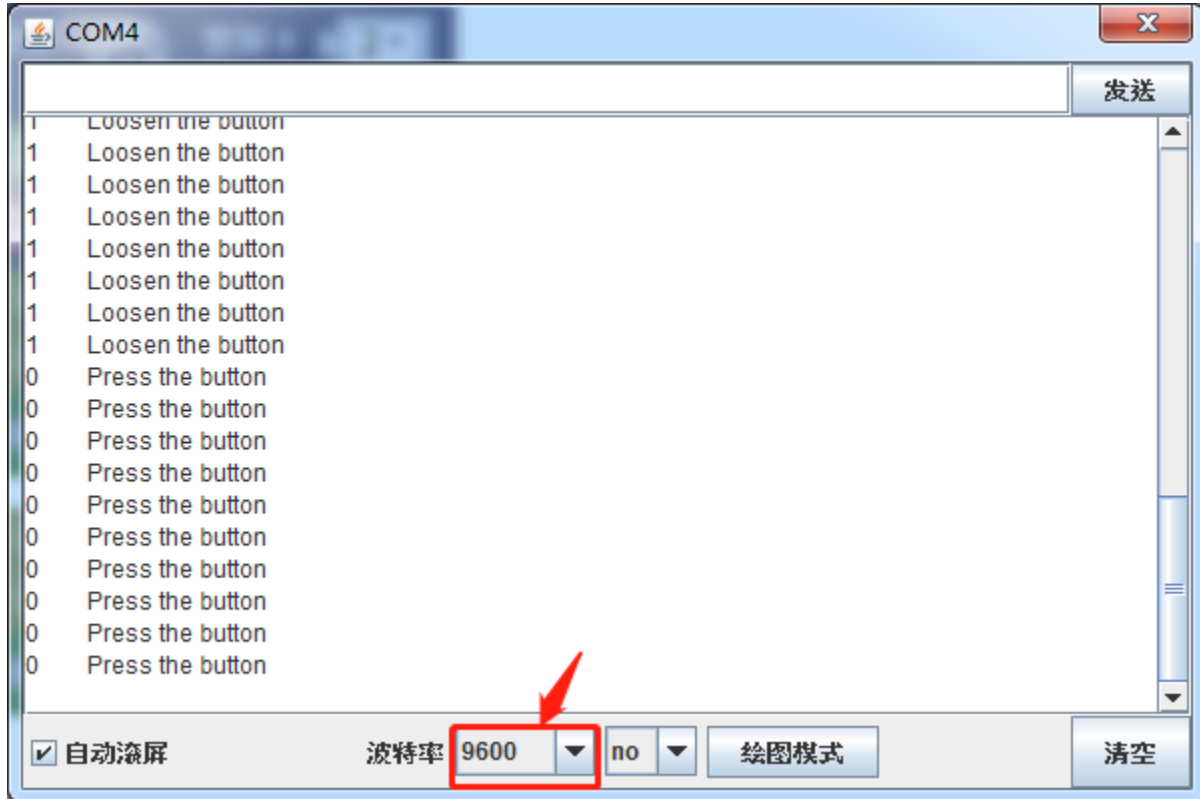


7.4 4. 代码说明

1. 在单元中找到，将 item 变量名改为“button”。这里修改变量名是为了便于理解与记忆，设置 button 初始变量为整数，并赋值为 0。
2. 初始化设置波特率为 9600，波特率是串口通信传输的速度，表示 9600 位每秒。
3. 在输入/输出模块中找到并拖出数字输入管脚，并将管脚设置为 3，读取到 D3 管脚的按键值赋值给变量 button。
4. 找到并拉出串口模块下的打印代码块，将自动换行改为不换行，并打印变量 button。
5. 在单元内设置“如果…否则…”逻辑单元，实验中只需添加一个否则单元。代码逻辑是当传感器感应到按键按下时，button 为 0，串口监视器显示相关字符；按键松开时，button 为 1，显示其他字符。

7.5 5. 测试结果

上传测试代码成功后，利用 USB 供电并打开串口监视器，设置波特率为 9600。实验中，当传感器按下按键时，button 为 0，串口监视器显示 “Press the button”；按键松开时，button 为 1，串口监视器显示 “Loosen the button”，如下图。



8.3 3. 测试代码

```
import machine
import utime

led_external = machine.Pin(25, machine.Pin.OUT) # 设置GPIO25为输出
button = machine.Pin(14, machine.Pin.IN, machine.Pin.PULL_UP) # 设置GPIO14为按键输入

# 延时函数
while True:
    if button.value() == 0: # 判断按键是否按下
        led_external.value(1) # 点亮板载LED灯
        utime.sleep(0.2) # 延时0.2S
        led_external.value(0) # 熄灭板载LED灯
```

8.4 4. 代码说明

1. `import machine` 和 `import utime`: 导入用于硬件控制及时间控制的库。
2. `led_external = machine.Pin(25, machine.Pin.OUT)`: 将 GPIO25 设置为输出模式, 控制外部 LED 的状态。
3. `button = machine.Pin(14, machine.Pin.IN, machine.Pin.PULL_UP)`: 将 GPIO14 设置为输入模式, 并配置为上拉电阻, 确保无按键按下时, 电平为高。
4. `while True`: 启动一个无限循环, 实时监测按键状态。
5. `if button.value() == 0`: 检查按键是否被按下, 当按键按下时, 执行点亮 LED 的操作。
6. `led_external.value(1)` 和 `led_external.value(0)`: 分别用于点亮和熄灭 LED 灯, 以实现指示效果。

8.5 5. 测试结果

接线图如上所示, 运行程序后, 当按下按键时, PICO 板载的 LED 灯亮起, 松开按键时, LED 灯熄灭。该实验展示了如何使用按钮控制 LED 的基本原理, 适用于多种电子应用场景。