



StarFive  
赛昉科技

# 在昉·星光 2上进行指定时间 内的按键检测

Python语言版本

应用指南

版本： 1.1

日期： 2023/06/08

Doc ID: VisionFive 2-ANCH-011

# 法律声明

阅读本文件前的重要法律告知。

## 版权注释

版权 ©上海赛昉科技有限公司，2023。版权所有。

本文档中的说明均基于“视为正确”提供，可能包含部分错误。内容可能因产品开发而定期更新或修订。上海赛昉科技有限公司（以下简称“赛昉科技”）保留对本协议中的任何内容进行更改的权利，恕不另行通知。

赛昉科技明确否认任何形式的担保、解释和条件，无论是明示的还是默示的，包括但不限于适销性、特定用途适用性和非侵权的担保或条件。

赛昉科技无需承担因应用或使用任何产品或电路而产生的任何责任，并明确表示无需承担任何及所有连带责任，包括但不限于间接、偶然、特殊、惩戒性或由此造成的损害。

本文件中的所有材料受版权保护，为赛昉科技所有。不得以任何方式修改、编辑或断章取义本文件中的说明，本文件或其任何部分仅限用于内部使用或教育培训。使用文件中包含的说明，所产生的风险由您自行承担。赛昉科技授权复制本文件，前提是您保留原始材料中包含的所有版权声明和其他相关声明，并严格遵守此类条款。本版权许可不构成对产品或服务的许可。

## 联系我们：

地址：浦东新区盛夏路61弄张润大厦2号楼502，上海市，201203，中国

网站：<http://www.starfivetech.com>

邮箱：[sales@starfivetech.com](mailto:sales@starfivetech.com)（销售） [support@starfivetech.com](mailto:support@starfivetech.com)（支持）

---

# 目录

表格清单.....	4
插图清单.....	5
法律声明.....	ii
前言.....	vi
<b>1. 产品简介.....</b>	<b>7</b>
1.1. 40-Pin GPIO Header定义.....	7
<b>2. 准备.....</b>	<b>8</b>
2.1. 运行环境要求.....	8
2.2. 准备硬件.....	8
2.2.1. 连接硬件.....	9
2.3. 准备软件.....	11
<b>3. 执行演示代码.....</b>	<b>14</b>
<b>4. 演示源代码.....</b>	<b>16</b>
<b>5. 资源下载.....</b>	<b>18</b>
<b>6. 立即购买.....</b>	<b>19</b>

# 表格清单

表 0-1 修订历史.....	vi
表 2-1 硬件准备.....	8
表 2-2 将按键连接到40-Pin Header上.....	10



# 插图清单

图 1-1 40-Pin GPIO Header定义.....	7
图 2-1 面包板概述图.....	9
图 2-2 连接按键和40-Pin Header.....	10



# 前言

关于本指南和技术支持信息

## 关于本手册

该应用说明提供使用昉·星光 2的GPIO Pin，实现指定时间窗口内的上升沿或者下降沿检测。检测上升沿或下降沿意味着检测按钮的信号。

## 修订历史

表 0-1 修订历史

版本	发布说明	修订
1.1	2023/06/08	更新了 <a href="#">准备软件(第 11页)</a> 中安装VisionFive.gpio包的方法。
1.0	2023/05/31	首次发布。

## 注释和注意事项

本指南中可能会出现以下注释和注意事项：

-  **提示：**  
建议如何在某个主题或步骤中应用信息。
-  **注：**  
解释某个特例或阐释一个重要的点。
-  **重要：**  
指出与某个主题或步骤有关的重要信息。
-  **警告：**  
表明某个操作或步骤可能会导致数据丢失、安全问题或性能问题。
-  **警告：**  
表明某个操作或步骤可能导致物理伤害或硬件损坏。

# 1. 产品简介

该应用说明提供使用昉·星光 2的GPIO Pin，实现指定时间窗口内的上升沿或者下降沿检测。检测上升沿或下降沿意味着检测按钮的信号。

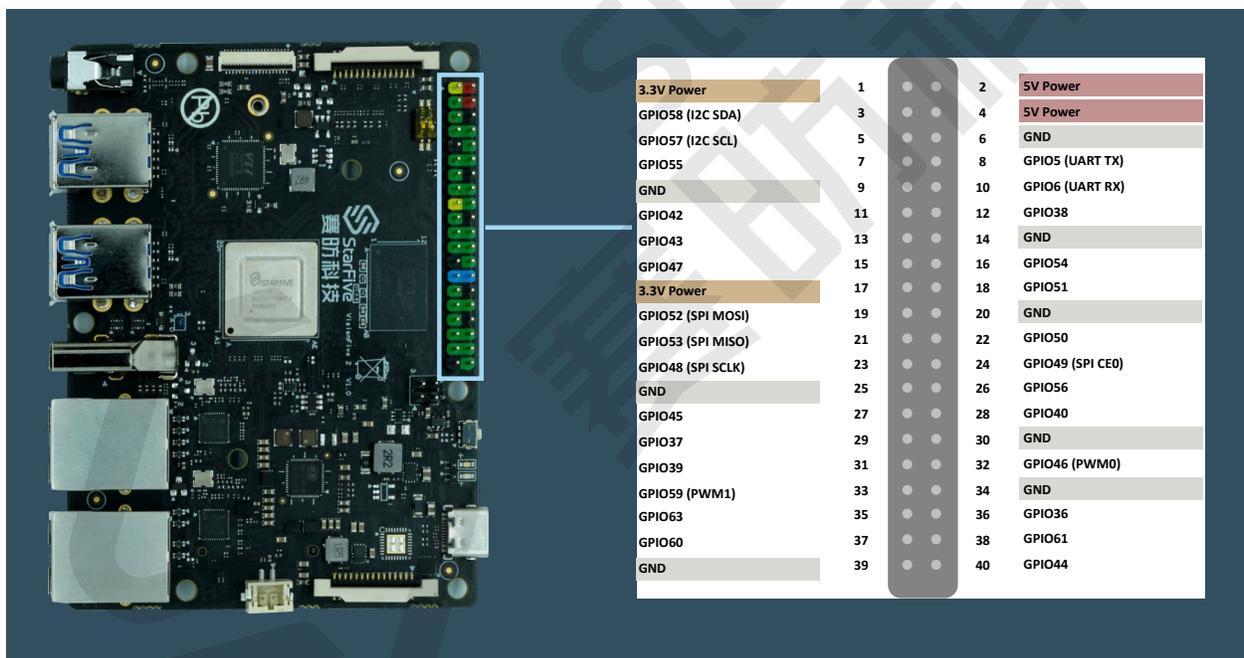
## 提示：

VisonFive.GPIO兼容Rpi.GPIO命令，即Rpi.GPIO python演示可以在昉·星光 2上运行。此外，API `add_event_detect()`的回调函数较Rpi.GPIO做了优化，回调函数增加一个参数`edge_type`。因此Rpi.GPIO涉及到回调函数的python演示需要手动修改，增加一个`edge_type`即可。

## 1.1. 40-Pin GPIO Header定义

下图显示了40-pin GPIO Header的位置：

图 1-1 40-Pin GPIO Header定义



## 注：

功能复用pin脚已初始化，不可作为通用GPIO使用。

## 2. 准备

在执行演示程序之前，务必确认已准备好以下项目：

### 2.1. 运行环境要求

该演示运行环境要求如下：

- Linux内核版本：Linux 5.15
- 操作系统：Debian 12
- 硬件版本：昉·星光 2
- SoC：昉·惊鸿7110

### 2.2. 准备硬件

在执行演示程序之前，请务必准备以下硬件：

表 2-1 硬件准备

类型	M/O*	项目	注释
通用	M	昉·星光 2 单板计算机	-
通用	M	<ul style="list-style-type: none"><li>• 容量不低于32 GB的Micro-SD卡</li><li>• Micro-SD卡读卡器</li><li>• 计算机 (Windows/Mac OS/Linux)</li><li>• USB转串口转换器 (3.3 V I/O, 带线)</li><li>• 以太网电缆</li><li>• 电源适配器 (5 V/ 3 A)</li><li>• USB Type-C数据线</li></ul>	上述项目用于将Debian OS烧录到Micro-SD上。
按键检测演示 (指定时间)	M	<ul style="list-style-type: none"><li>• 一个LED灯</li><li>• 一个面包板</li><li>• 两根公母跳线</li><li>• 一个4-pin按键</li></ul>	<ul style="list-style-type: none"><li>• LED代表发光二极管，当电流通过时发光。长腿（称为“阳极”）始终连接到电路的正极电源。短腿（称</li></ul>

表 2-1 硬件准备 (续)

类型	M/ O*	项目	注释
			<p>为“阴极”) 连接到电源的负极, 称为“接地”。</p> <ul style="list-style-type: none"> <li>面包板: 请参阅下文介绍。</li> </ul>



注:

\*: M: 必须。O: 可选

## 面包板介绍

面包板是一种连接各种电子元件的方法, 免去了焊接的工序。面包板通常用于制作PCB前的电路测试。如下图所示, 面包板的顶部和底部各有两行线, 通常用于主电源连接, 蓝线用作负极, 红线用作正极。此外红线和蓝线被分割为两段, 每段所覆盖的小孔已经连通。

面包板的每一列 (A到E, F到J) 的行的孔是连接的; 每一行 (1到57) 是没有连接的。

图 2-1 面包板概述图



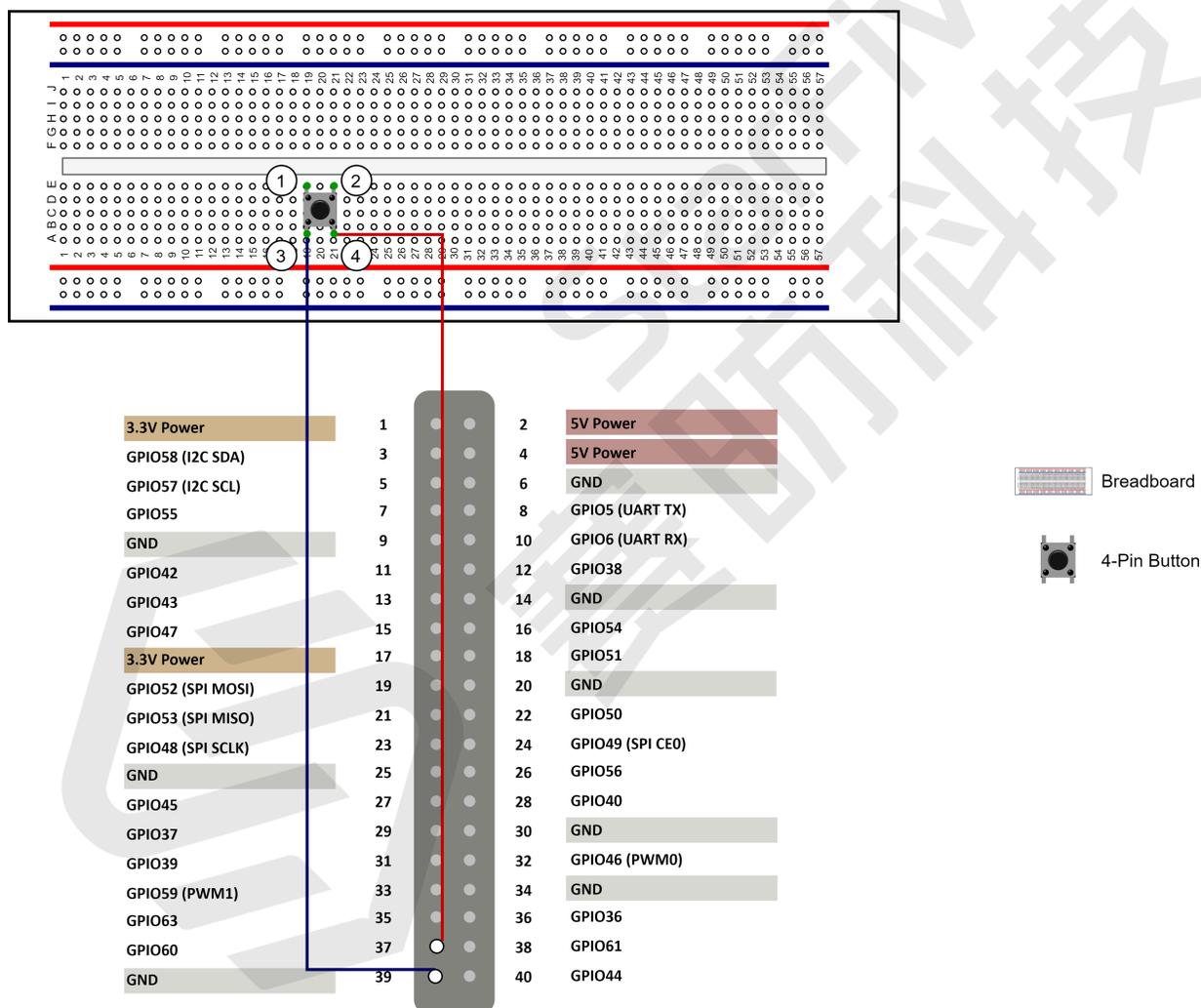
### 2.2.1. 连接硬件

将按键的Pin ①和Pin ②分别连接到面包板上第19行和第21行的小孔上，然后将按键的Pin ③连接到昉·星光 2的Pin 39，Pin ④连到昉·星光 2的Pin 37。以下表格和图片描述了如何将按键连接到40-Pin GPIO Header上：

表 2-2 将按键连接到40-Pin Header上

按键	40-Pin GPIO Header	
	Pin号	Pin名
Pin ④	37	GPIO60
Pin ③	39	GND

图 2-2 连接按键和40-Pin Header



**提示：**

按键内部①、③引脚连通，②、④引脚连通。

## 2.3. 准备软件

确认按照以下步骤进行操作：



**注：**

该Python应用VisionFive.gpio适用于昉·星光单板计算机、昉·星光 2和昉·惊鸿7110 EVB。

1. 按照[《昉·星光 2单板计算机快速参考手册》](#)中的“将OS烧录到Micro-SD”章节，将Debian OS烧录到Micro-SD卡上。
2. 登录Debian并确保昉·星光 2已联网。有关详细说明，请参阅[《昉·星光 2单板计算机快速参考手册》](#)中“通过以太网使用SSH”或“使用USB转串口转换器”章节。
3. 在Debian上扩展分区，请参见[《昉·星光 2单板计算机快速参考手册》](#)中“扩展分区”章节。
4. 执行以下命令，在Debian系统上安装PIP：

```
apt-get install python3-pip
```

5. 在昉·星光 2 Debian上执行pip命令，以安装VisionFive.gpio包：



**注：**

由于pypi.org官网尚不支持上传RISC-V平台的whl安装包，不能直接使用pip install visionfive.gpio命令在线安装，因此请按照以下步骤安装VisionFive.gpio包。

- a. 执行以下命令，安装依赖包：

```
apt install libxml2-dev libxslt-dev
python3 -m pip install requests wget bs4
```

- b. 执行以下命令，运行安装脚本Install\_VisionFive\_gpio.py：

```
python3 Install_VisionFive_gpio.py
```

安装脚本代码如下：

```
import requests
import wget
import sys
import os
from bs4 import BeautifulSoup

def parse_data(link_addr, class_type, key_str):
    req = requests.get(url=link_addr)
    req.encoding = "utf-8"
    html=req.text
```



```

soup = BeautifulSoup(req.text, features="html.parser")
package_version = soup.find(class_type, class_=key_str)
dd = package_version.text.strip()
data = dd.split()
return data

def parse_link(link_addr, class_type, key_str):
    req = requests.get(url=link_addr)
    req.encoding = "utf-8"
    html=req.text
    soup = BeautifulSoup(req.text, features="html.parser")
    search_data = soup.find(class_type, class_=key_str)
    search_data_2 = search_data.find("a")
    dl_link_get = search_data_2.get("href")
    return dl_link_get

def get_dl_addr_page():
    link_address
    = "https://pypi.org/project/VisionFive.gpio/#history"
    key_str = "release__version"
    class_key = "p"
    data_get = parse_data(link_address, class_key, key_str)
    latest_version = data_get[0]

    dl_addr_page
    = "https://pypi.org/project/VisionFive.gpio/{}/#files".format(latest_version)

    return dl_addr_page

def get_dl_addr_of_latest_version(link_addr):
    key_str = "card file__card"
    class_key = "div"
    addr_get = parse_link(link_addr, class_key, key_str)

    return addr_get

def main():
    dl_addr_p = get_dl_addr_page()
    whl_dl_addr = get_dl_addr_of_latest_version(dl_addr_p)

    whl_name = whl_dl_addr.split("/")[-1]
    whl_name_suffix = os.path.splitext(whl_name)[-1]
    whl_name_prefix = os.path.splitext(whl_name)[0]
    whl_name_prefix_no_platform = whl_name_prefix[0:
len(whl_name_prefix) - 3]
    new_platform = "linux_riscv64"

```



```
rename_whl_name
= "{}{}{}".format(whl_name_prefix_no_platform,
new_platform, whl_name_suffix)

wget.download(whl_dl_addr, out=rename_whl_name)

os.system("pip install " + rename_whl_name)
os.system("rm -rf " + rename_whl_name)

if __name__ == '__main__':
    sys.exit(main())
```

## 3. 执行演示代码

执行以下操作，以在昉·星光 2的Debian系统上运行演示代码：

1. 找到测试代码`edge_detection_with_waiting_time.py`所在的目录：

a. 执行以下命令以获取`VisionFive.gpio`所在的目录：

```
pip show VisionFive.gpio
```

示例结果：

```
Location: /usr/local/lib64/python3.9/site-packages
```



注：

实际输出取决于应用的安装方式。

b. 如前一步输出中所示，执行以下操作进入目录`/usr/local/lib64/python3.9/site-packages`：

```
cd /usr/local/lib64/python3.9/site-packages
```

c. 执行以下命令进入`sample-code`目录：

```
cd ./VisionFive/sample-code/
```

2. 在`sample-code`目录下，执行以下命令以运行演示代码：

```
sudo python edge_detection_with_waiting_time.py
```

或者，您也可以执行以下命令：

```
sudo python3 edge_detection_with_waiting_time.py
```

结果：

• 终端显示如下：

```
# python3 edge_detection_with_waiting_time.py
*-----Case 1-----
*
Note: don't press the key on pin 37 once within 5 seconds !!!
```

• 等待5秒后，终端显示如下：

```
Edge hasn't been detected within 5 seconds while setting 5
seconds to timeout.
The return value of GPIO.event_detected(37) within 5 seconds:
False
```

```
*-----Case 2-----  
-----*  
Please press the key on pin 37 once at any time !!!
```

- 最后根据提示按下按键，终端显示如下：

```
Edge has detected while setting -1 to timeout,  
timeout -1 means waiting until edge is detected.  
  
The return value (True) of GPIO.event_detected(37) should be True.  
The return value (False) of GPIO.event_detected(37) should be  
False, because of the secondly reading.
```

## 4. 演示源代码

本演示中的资源代码仅作为参考。

edge\_detection\_with\_waiting\_time.py:

```
'''
Please make sure the button is connected to the correct pins.
The following table describes how to connect the button to the 40-pin
header.
-----
| button | Pin Number | Pin Name |
| one end | 37 | GPIO60 |
| The other end | 39 | GND |
-----
'''

import VisionFive.gpio as GPIO
import sys
import time

key_pin = 37

def main():
    # Configure the direction of key_pin as input.
    GPIO.setup(key_pin, GPIO.IN)

    print("*-----Case
1-----*")
    print("Note: don't press the key on pin {} once within 5
seconds !!!".format(key_pin))
    print()
    # edge falling can be detected, also set bouncetime(unit: millisecond)
to avoid jitter.
    # timeout(unit: millisecond), it means if edge will be detected within
timeout time.
    # timeout -1 means waiting until edge is detected.
    edge_detected = GPIO.wait_for_edge(key_pin, GPIO.FALLING, bouncetime=2,
timeout=5000)

    if edge_detected == key_pin:
        print("Edge has detected within 5 seconds while setting 5 seconds to
timeout.")
    else:
        print("Edge hasn't been detected within 5 seconds while setting 5
seconds to timeout.")
```

```

# query if edge event happens.
edge_detected_flag = GPIO.event_detected(key_pin)

print("The return value of GPIO.event_detected({}) within 5 seconds:
{}".format(key_pin, edge_detected_flag))

print()
print("*-----Case
2-----*")

print("Please press the key on pin {} once at any
time !!!".format(key_pin))

# edge rising can be detected, also set bouncetime(unit: millisecond) to
avoid jitter.
# the default timeout is -1, meaning that waiting until edge is
detected.
edge_detected = GPIO.wait_for_edge(key_pin, GPIO.RISING, bouncetime=2)

if edge_detected == key_pin:
    print("Edge has detected while setting -1 to timeout,")
    print("timeout -1 means waiting until edge is detected.")
else:
    print("Edge hasn't been detected while setting -1 to timeout,")
    print("timeout -1 means waiting until edge is detected.")

# query if edge event happens.
edge_detected_flag = GPIO.event_detected(key_pin)

print()
print("The return value ({} of GPIO.event_detected({}) should be
True.".format(edge_detected_flag, key_pin))

# query if edge event happens.
edge_detected_flag = GPIO.event_detected(key_pin)

print("The return value ({} of GPIO.event_detected({}) should be False,
because of the secondly reading.".format(edge_detected_flag, key_pin))

print()

if __name__ == "__main__":
    sys.exit(main())

```

---

## 5. 资源下载

点击本栏找到所有的代码下载资源。

本页包括所有赛昉科技提供的代码下载资源。

- [RVspace Wiki](#)
- [应用中心](#)
- [文档中心](#)
- [技术论坛](#)
- [昉·星光 2 GitHub代码仓](#)
- [昉·星光 2 Debian操作系统下载](#)
- [代码下载 \(赛昉科技官方GitHub页面\)](#)
- [所有开源技术文档](#)



StarFive  
赛昉科技

---

## 6. 立即购买

点击本栏获取在线购买链接和配件购买链接。

### 购买单板计算机

点击以下页面，您可以找到所在地区的经销商，或覆盖全球的销售渠道，以购买昉·星光 2 单板计算机。

- [购买昉·星光2开发板](#)

### 购买配件

点击以下页面，您可以找到所有昉·星光 2 单板计算机已验证适配的配件及其购买链接。

- [购买配件](#)