StarFive
赛昉科技

在昉·星光 2上运行RPi演示，使LED以PWM频率闪烁

Python语言版本
应用指南
版本：1.11
日期：2025/04/27
Doc ID：VisionFive 2-ANCH-014

法律声明

阅读本文件前的重要法律告知。

版权注释

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。上海赛昉半导体科技有限公司（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。

联系我们：

地址： 中国（上海）自由贸易试验区盛夏路61弄张润大厦2号电梯楼层5层（实际楼层4层）06室

网站： http://www.starfivetech.com

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

Contents

List of Tables

List of Figures

前言

关于本指南和技术支持信息

关于本手册

该应用说明如何使用昉·星光 2运行 RPi GPIO演示，使LED以PWM频率闪烁。

修订历史

Table 0-1 修订历史

版本	发布日期	修订历史
1.11	2025/04/27	更新了连接硬件 *(on page 9)*中的表格。
1.1	2023/06/08	更新了准备软件 *(on page 10)*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:

- **Tip:**
  建议如何在某个主题或步骤中应用信息。
- **Note:**
  解释某个特例或阐释一个重要的点。
- **Important:**
  指出与某个主题或步骤有关的重要信息。
- **CAUTION:**
  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。
- **Warning:**
  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

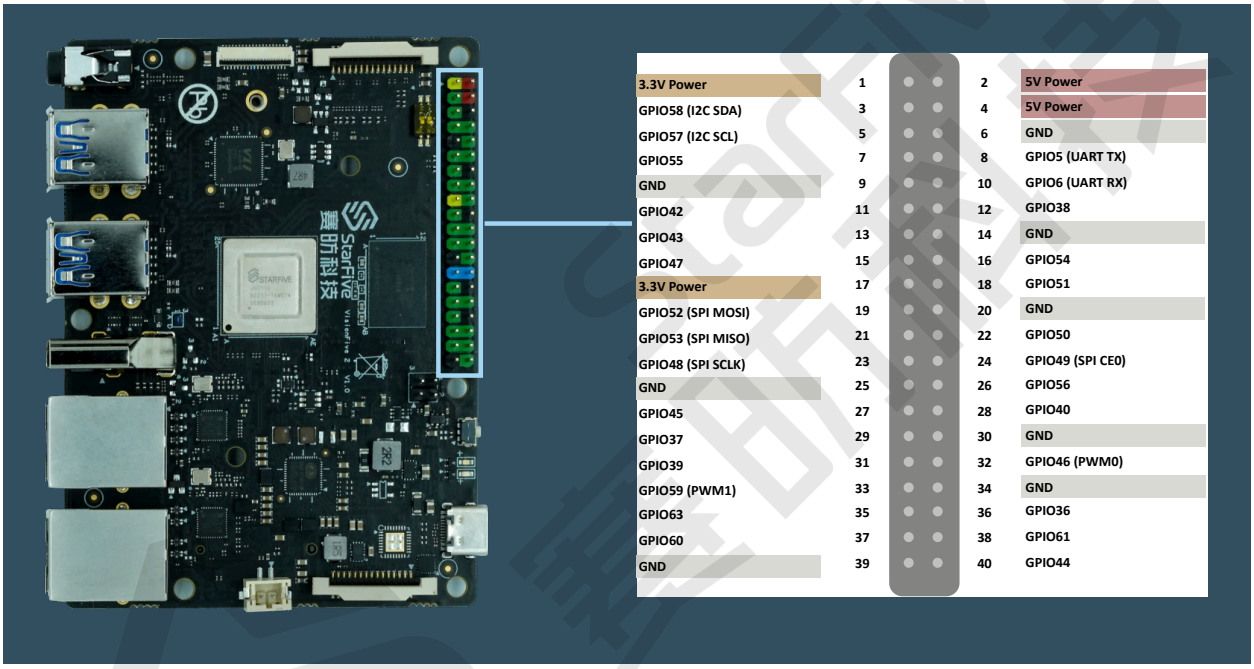
该应用说明如何使用昉·星光 2运行 RPi GPIO演示，使LED以PWM频率闪烁。

Tip:
`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:

Figure 1-1 40-Pin GPIO Header定义



Note:
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：

Table 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	• 容量不低于32 GB的Micro-SD卡 • Micro-SD卡读卡器 • 计算机（Windows/Mac OS/Linux） • USB转串口转换器（3.3 V I/O，带线） • 以太网电缆 • 电源适配器（5 V/ 3 A） • USB Type-C数据线	上述项目用于将Debian OS烧录到Micro-SD上。
GPIO演示(PWM)	M	• 一个LED灯 • 一个面包板 • 两根公母跳线 • 470Ω色环电阻器	• 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（称为“阴极”）连接到电源的负极，称为“接地”。 • 面包板：请参阅面包板介绍 *(on page 9)*。 • 电阻器：电阻器用于限制通过电路的电量；具体来说，它们限制的是允许流动的“电流”量。

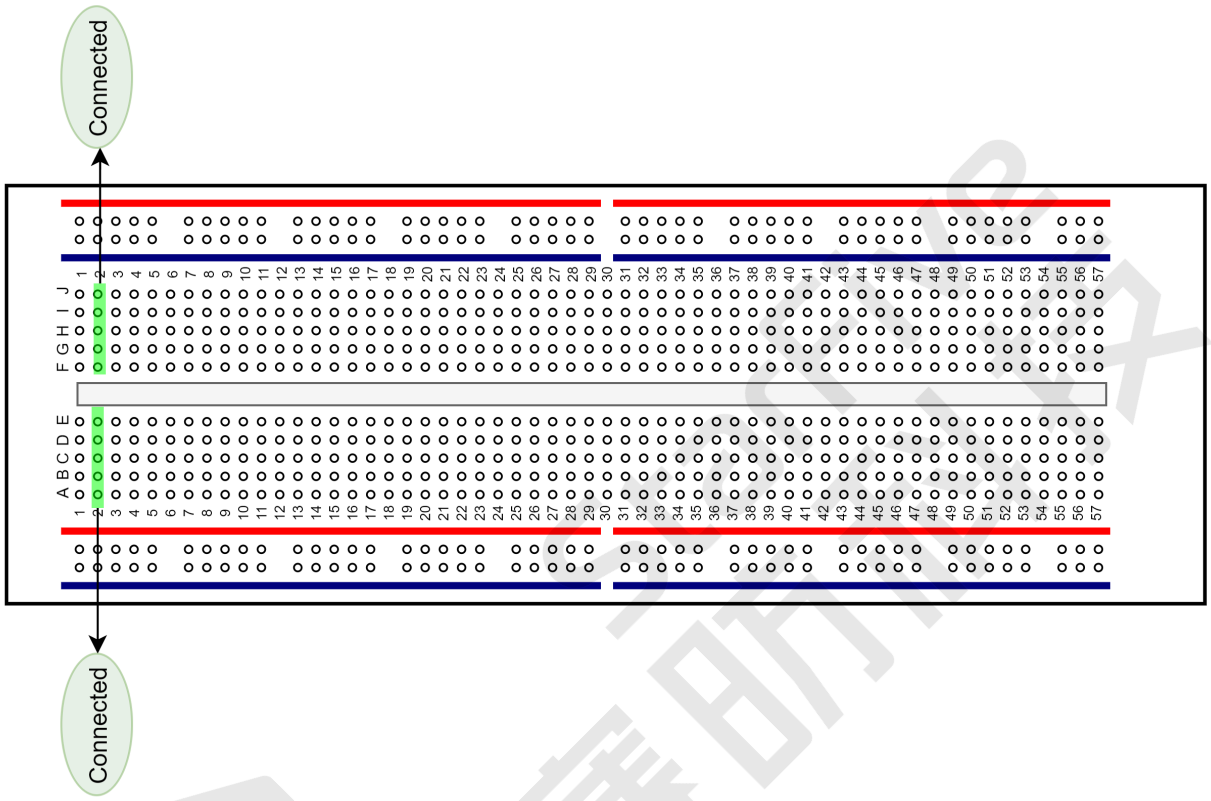
Note:
*: M：必须。O：可选

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）的行的小孔是连接的；每一行（1到57）是没有连接的。

Figure 2-1 面包板概述图



2.2.1. 连接硬件

以下表格和图片描述了如何将LED连接到40-Pin GPIO Header上：

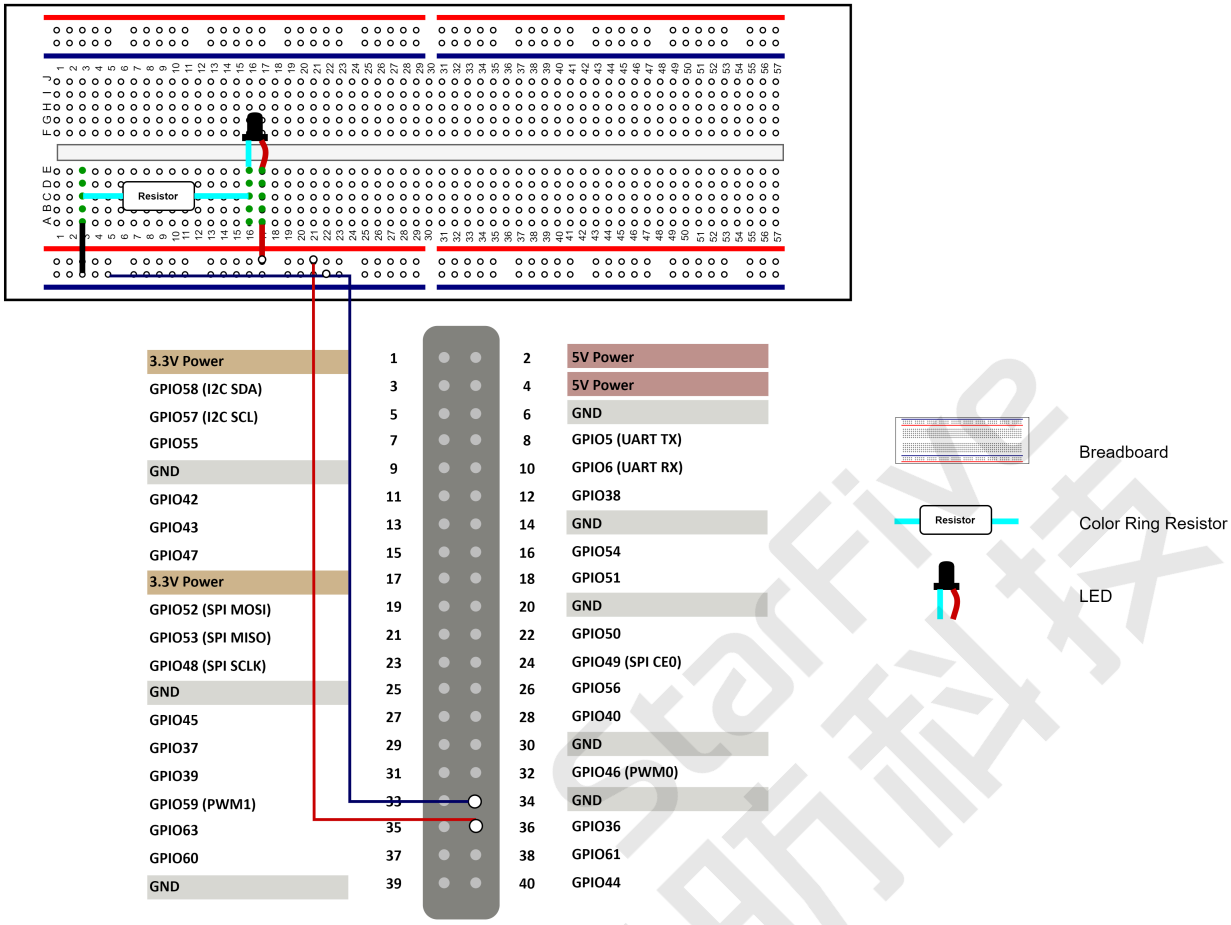
Table 2-2 将按键连接到40-Pin GPIO Header上

LED	40-Pin GPIO Header	
	Pin Number	Pin Name
正极	36	GPIO36
负极	34	GND

执行以下步骤，将LED连接到40-pin GPIO Header上：

1. 将昉·星光 2 GPIO36的pin线连接到面包板的红线。
2. 如下图所示连接电阻器。
3. 将LED的长脚连接到面包板的红线。
4. 将LED的短脚连接到面包板的蓝线。
5. 将昉·星光 2 GND的pin线连接到面包板的蓝线。

Figure 2-2 连接LED、电阻器和40-Pin GPIO Header



2.3. 准备软件

确认按照以下步骤进行操作:

Note:
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:

   Note:
   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. 在昉·星光 2上运行RPi演示，需要将RPi python演示中的`import RPi.GPIO as GPIO`替换为`import VisionFive.gpio as GPIO`。
2. 找到测试代码`RPi_demo_#PWM#_run_on_VisionFive.py`所在的目录：
   a. 执行以下命令安装依赖：

   ```
   pip install pillow
   ```

   b. 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
   ```

   Note:

   实际输出取决于应用的安装方式。

   c. 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
   ```

   d. 执行以下命令进入`sample-code`目录：

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

3. 在`sample-code`目录下，执行以下命令以运行演示代码：

   ```
   sudo python RPi_demo_#PWM#_run_on_VisionFive.py
   ```

   或者，您也可以执行以下命令：

   ```
   sudo python3 RPi_demo_#PWM#_run_on_VisionFive.py
   ```

   结果：

   LED闪烁，闪烁频率将根据PWM频率的变化而变化。

4. 演示源代码

本演示中的资源代码仅作为参考。

`RPi_demo_#PWM#_run_on_VisionFive.py`:

```
#import RPi.GPIO as GPIO
#*****************************************************
#Note: above command must be replaced with command below
import VisionFive.gpio as GPIO
#*****************************************************
from time import sleep

ledpin = 36    # PWM pin connected to LED
GPIO.setwarnings(False)      # disable warnings
GPIO.setmode(GPIO.BOARD)     # set pin numbering system
GPIO.setup(ledpin,GPIO.OUT)
pi_pwm = GPIO.PWM(ledpin,1000)  # create PWM instance with frequency
pi_pwm.start(0)          # start PWM of required duty cycle
while True:
    for duty in range(0,101,1):
        pi_pwm.ChangeDutyCycle(duty)          # provide duty cycle in the range 0-100
        sleep(0.01)
    sleep(0.5)

    for duty in range(100,-1,-1):
        pi_pwm.ChangeDutyCycle(duty)
        sleep(0.01)
    sleep(0.5)
```

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

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