



StarFive
赛昉科技

使用昉·星光使LED以PWM 频率闪烁

Python语言版本

应用说明

版本： 1.0

日期： 2022/07/29

Doc ID: VisionFive-ANCH-011-1.0

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，2018-2022。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（销售） support@starfivetech.com（支持）

前言

关于本指南和技术支持信息。

关于本手册

本应用说明提供使用昉·星光的GPIO引脚使LED以PWM频率闪烁的步骤。






修订历史

表 0-1 修订历史

版本	发布说明	修订
V1.0	2022/07/29	首次发布。

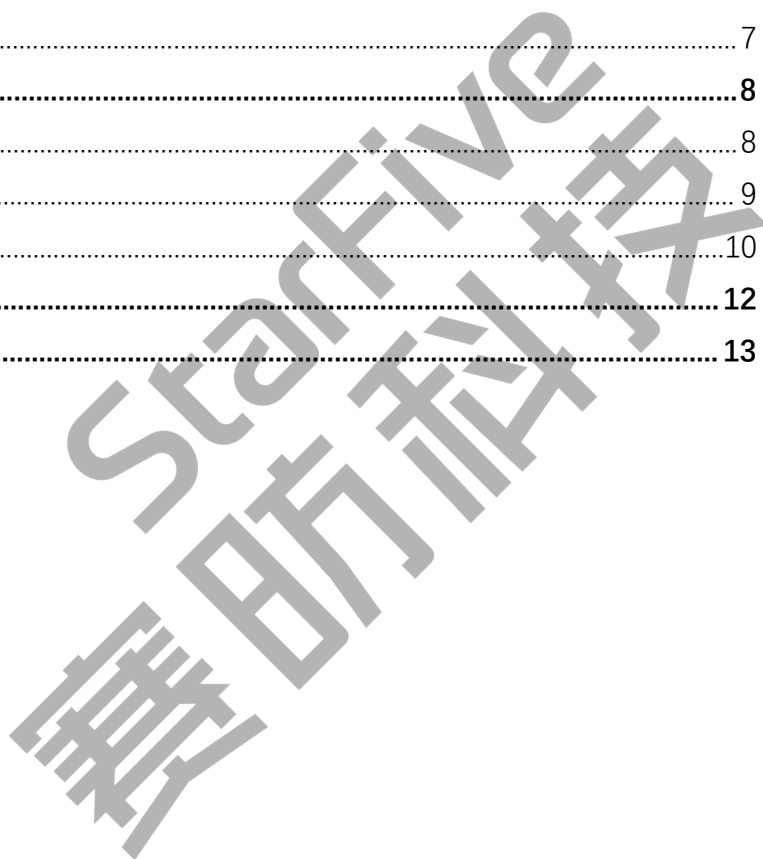
注释和注意事项

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某个重要的点。
-  **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。

目录

表格清单.....	5
插图清单.....	6
法律声明.....	ii
前言.....	iii
1. 介绍.....	7
1.1. 40-Pin Header定义.....	7
2. 准备.....	8
2.1. 准备硬件.....	8
2.1.1. 连接硬件.....	9
2.2. 准备软件.....	10
3. 执行演示代码.....	12
4. 演示源代码.....	13



表格清单

表 0-1 修订历史.....	iii
表 2-1 硬件准备.....	8
表 2-2 将LED连接到40-Pin Header上.....	9



插图清单

图 1-1 40-Pin定义.....	7
图 2-1 面包板概述图.....	9
图 2-2 将LED连接到40-Pin Header上.....	10



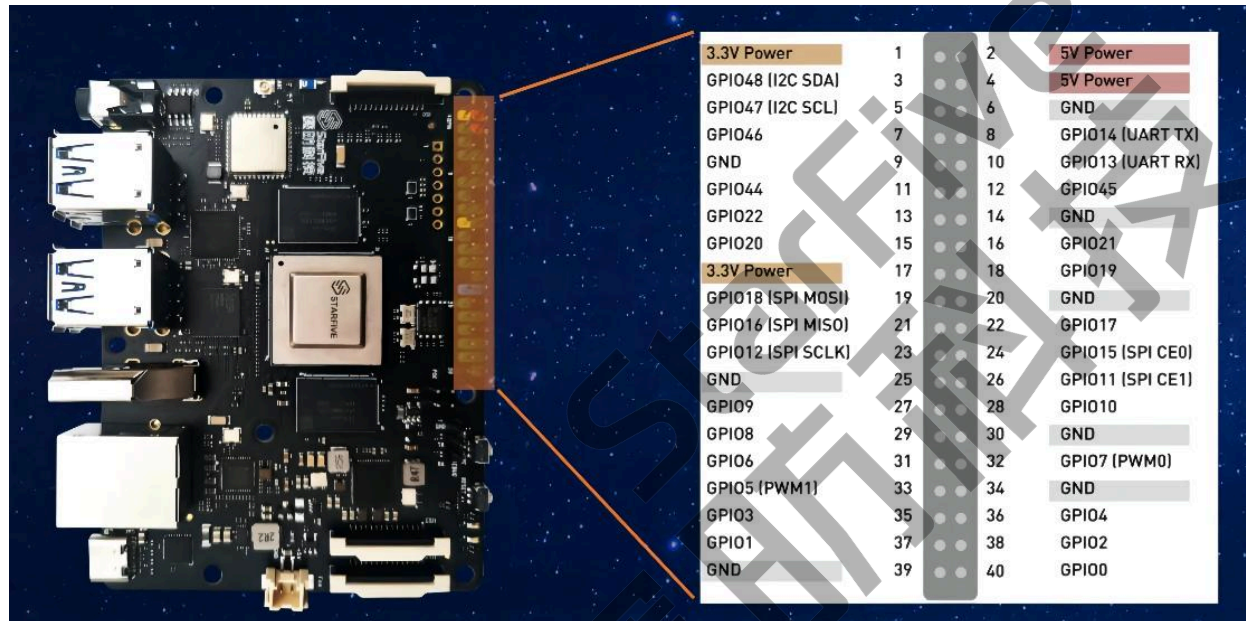
1. 介绍

本应用说明提供使用昉·星光的GPIO引脚使LED以PWM频率闪烁的步骤。

1.1. 40-Pin Header定义

下图以昉·星光开发板为例说明40-Pin Header的位置：

图 1-1 40-Pin定义



2. 准备

在执行演示代码前，请确保您已准备好以下事项：

2.1. 准备硬件

在执行演示程序前，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	赛昉科技 单板计算机	可使用以下单板计算机： <ul style="list-style-type: none">• 星光板• 昉·星光
通用	M	<ul style="list-style-type: none">• 16GB（或更大）的Micro SD卡• Micro SD卡读卡器• 计算机（PC/Mac/Linux）• USB转串口转换器（3.3 V I/O）• 网线• 电源适配器• USB Type-C数据线	上述项目用于将Fedora OS烧录到Micro SD上。
GPIO 演示 (LED)	M	<ul style="list-style-type: none">• 一个LED灯• 一块面包板• 两根公对母跳线• 470 Ω 色环电阻器	<ul style="list-style-type: none">• 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 (第 9页)• 电阻器：电阻器用于限制通过电路的电量；具体来说，限制的是电流的“流动”量。



注：

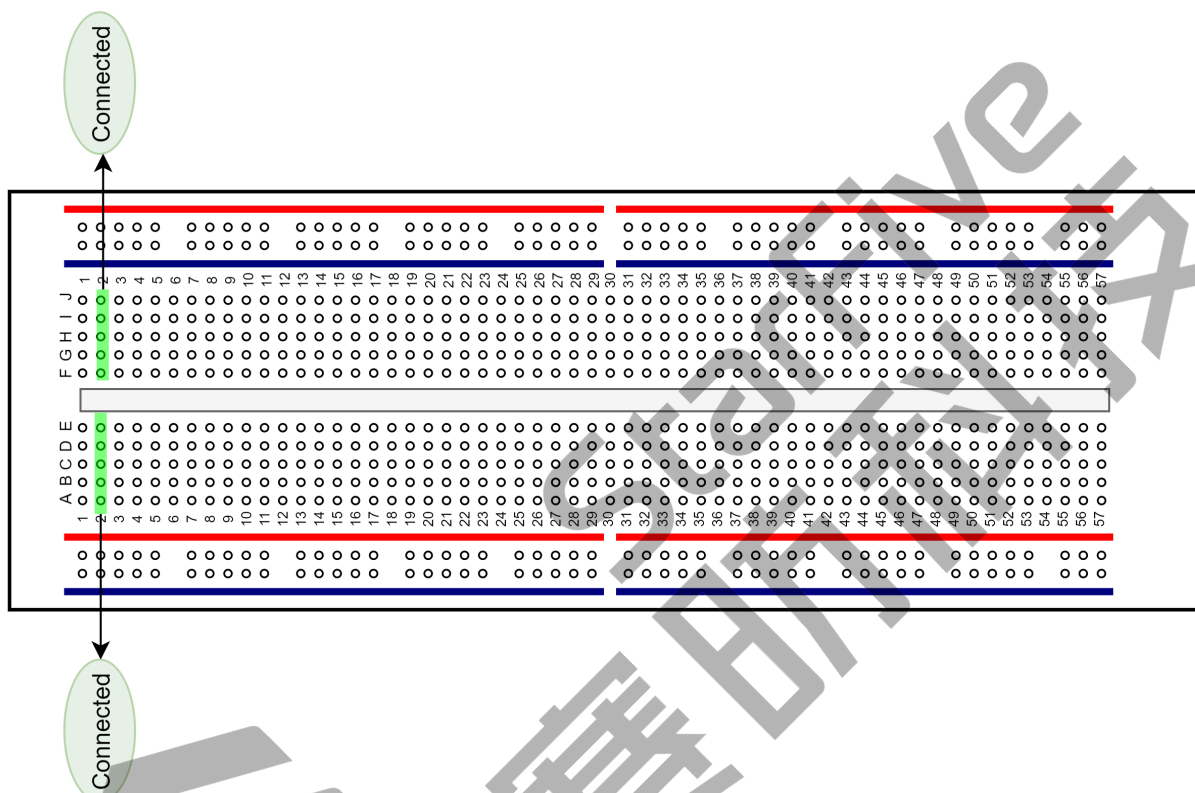
*: M: 必须。O: 可选

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）的行的的小孔是连接的；每一行（1到57）是没有连接的。

图 2-1 面包板概述图



2.1.1. 连接硬件

以下图表描述了如何将LED连接到40-Pin Header上：

表 2-2 将LED连接到40-Pin Header上

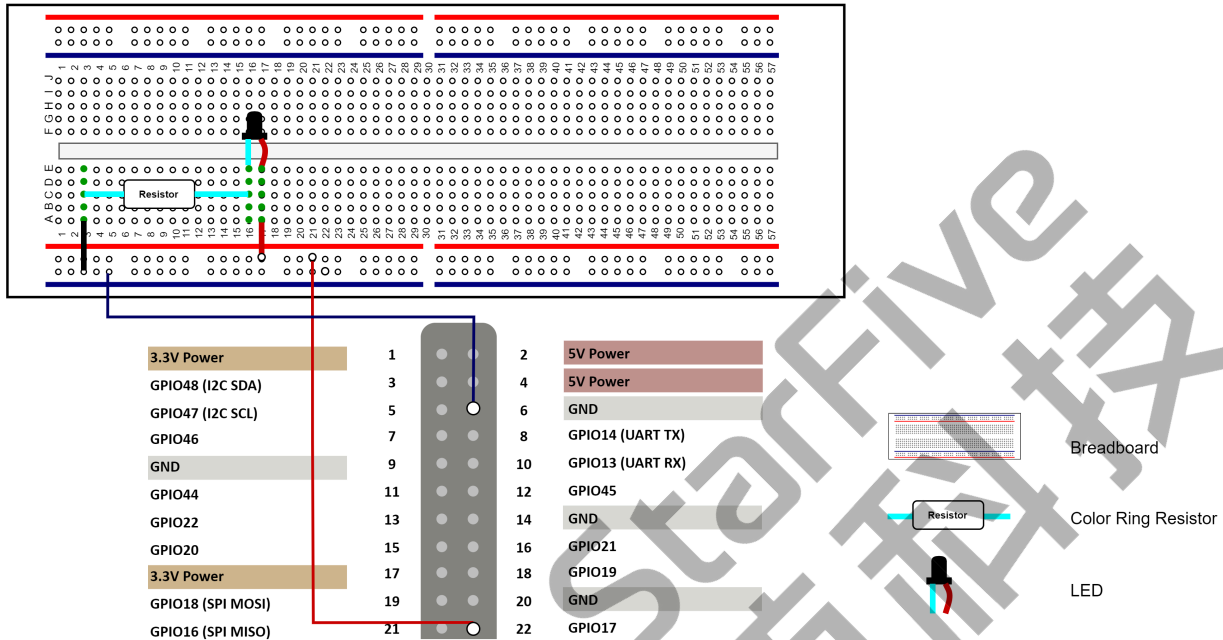
LED	40-Pin GPIO Header	
	引脚序号	引脚名
正极	22	GPIO17
负极	6	GND

执行以下步骤，将LED连接到40-pin GPIO Header上：

1. 将昉·星光上GPIO17的引脚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·星光上GND的引脚线连接到面包板的蓝线。

图 2-2 将LED连接到40-Pin Header上



2.2. 准备软件

确认按照以下步骤进行操作：

1. 按照《昉·星光单板计算机快速入门指南》中的“将Fedora烧录到Micro SD上”章节，将Fedora OS烧录到Micro SD卡上。
2. 登录Fedora并确保昉·星光已联网。有关详细说明，请参阅《昉·星光单板计算机快速入门指南》中“通过以太网使用SSH登录”或“使用USB转串口转换器连接并登录”章节。
3. 在昉·星光Fedora上执行pip命令，以安装VisionFive.gpio包：

```
sudo pip install VisionFive.gpio
```

或者，您也可以执行以下命令：

```
sudo pip3 install VisionFive.gpio
```

4. (可选) 如果您将源代码复制到昉·星光Fedora的本地目录下，请在源代码目录下执行以下命令：



提示：

点击以下链接可下载源代码：[VisionFive.gpio](#)。

```
sudo yum install python-devel python3-devel  
sudo python setup.py install
```

或者，您也可以执行以下命令：

```
sudo python3 setup.py install
```



3. 执行演示代码

在昉·星光上执行以下步骤，运行演示代码：

1. 找到测试代码`pwm_led.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 pwm_led.py
```

或者，您也可以执行以下命令：

```
sudo python3 pwm_led.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。

pwm_led.py:

```
'''
Please make sure the LED is connected to the correct pins.
The following table describes how to connect LED to the 40-pin header.
-----
      LED      Pin Number  Pin Name
      Positive      22      GPIO17
      Negative      6       GND
-----
'''

import time
import VisionFive.gpio as GPIO

led_pin = 17

#Configure the direction of led_pin as out.
GPIO.setup(led_pin, GPIO.OUT)
#Configure the voltage level of led_pin as high.
GPIO.output(led_pin, GPIO.HIGH)

#Configure the frequency as 10.
p = GPIO.PWM(led_pin, 10)
#Initialize the duty ratio as 0.
p.start(0)

try:
    #Change the LED blink frequency.
    while True:
        for dc in range(0, 101, 5):
            #Change the duty ratio from 0 to 100. Step size: 5
            p.ChangeDutyRatio(dc)
            time.sleep(1)
        for dc in range(100, -1, -5):
            #Change the duty ratio from 100 to 0. Step size: -5
            p.ChangeDutyRatio(dc)
            time.sleep(1)
except KeyboardInterrupt:
    pass

p.stop()
GPIO.cleanup(led_pin)
```