



StarFive
赛昉科技

测试昉·星光 GPIO 应用说明

版本：V1.1

日期：2021-12-27

版权注释

版权 © 上海赛昉科技有限公司 2018-2022。版权所有。

本文档中的信息均基于“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。

上海赛昉科技有限公司

地址：浦东新区盛夏路 61 弄张润大厦 2 号楼 502，上海，201203，中国

网站：www.starfivetech.com

邮箱：sales@starfivetech.com（销售）

support@starfivetech.com（支持）

关于本手册

介绍

本手册介绍了测试昉·星光 GPIO 的两种方法：

- 使用命令行测试。
- 使用演示代码测试。

修订历史

版本	已发布	修订
V1.0	2021/12/15	初始版本。
V1.1	2021/12/27	在“运行演示代码”部分： <ul style="list-style-type: none">• 新增了应用程序目录的说明。• 新增了 rsync 命令的说明。• 新增<User_Name> 字段说明。• 修正了一个拼写错误。

关于本手册	ii
1 准备	4
1.1 准备硬件	4
1.2 准备软件	4
2 使用命令行测试 GPIO	5
3 运行演示代码	6
3.1 编译源代码	6
4 使用演示代码测试 GPIO	7

StarFive

1 准备

在执行演示程序之前，务必确认已准备好以下项目：

1.1 准备硬件

在执行演示程序之前，请务必准备以下硬件：

表 1-1 硬件准备

类型	M/O	项目	注释
通用	M	一台单板计算机	可使用以下开发板： <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（5 V/3 A）• USB Type-C 转接线	上述项目用于将 Fedora 操作系统烧录至 Micro SD 卡中。
GPIO	M	一台示波器	示波器用于测量引脚电压，并检查 PWM 周期和占空比。

*M/O: M（必须）/ O（可选）

1.2 准备软件

- 软件环境
 - 电脑：Ubuntu 20.04
 - RISC-V 平台：Linux 5.16.0
- 将 Fedora OS 烧录到 Micro SD 卡中，并按照《赛昉科技 40-Pin GPIO Header 用户手册》中的“准备软件”部分编译和替换 dtb 文件。

2 使用命令行测试 GPIO

按照《赛昉科技 40-Pin GPIO Header 用户指南》中“配置 GPIO”部分测试 GPIO0。

StarFive

3 运行演示代码

执行以下操作，以运行演示代码：

3.1 编译源代码

1. 点击 [test-gpio.c](#) 下载测试源代码文件，保存到 Ubuntu 下的目标目录。例如，应用程序目录。
2. (可选) 安装编译工具。以下是安装示例：

```
sudo apt-get install gcc-riscv64-linux-gnu
```

说明：

如已安装该工具，则可跳过此步。

安装完成后，运行以下代码，以确认版本是否已更新：`linus@star-five$ riscv64-linux-gnu-gcc -v`。以下为输出示例：

```
Thread model: posix
gcc version 9.3.0 (Ubuntu 9.3.0-17ubuntu1~20.04)
```

图 3-1 输出示例

3. 执行以下命令，以编译源代码：

```
riscv64-linux-gnu-gcc -o test-gpio test-gpio.c
```

结果：

系统在当前目录生成了名为 `test-gpio` 的可执行文件。

4. 执行以下命令，以测试编译是否成功：

```
file test-gpio
```

结果：

在输出结果中出现 `UCB RISC-V`，则表示编译成功：

```
Riscv@starfive:~/work/app$ file test-gpio
test-gpio: ELF 64-bit LSB executable, UCB RISC-V, version 1 (SYSV), dynamically
linked, interpreter /lib/ld-linux-riscv64-lp64d.so.1, for GNU/Linux 4.15.0,
BuildID[sha1]=476d5a99c84f995d03227a18285222ac25e2cd0d, not stripped
c-v2x@starfive:~/work/app$
```

4 使用演示代码测试 GPIO

上电启动昉·星光，检查 GPIO22 的电压变化。

1. 在 Ubuntu 中执行以下命令，将可执行文件 test-gpio 上传到所需的单板计算机目录，例如 test 命令：

```
rsync ./test-gpio <User_Name>@<Board_IP_Address>:/home/riscv/test
```

说明：

- <User_Name>：开发板的用户名。例如：riscv。
- <Board_IP_Address>：开发板的 IP 地址。例如：192.168.92.133。

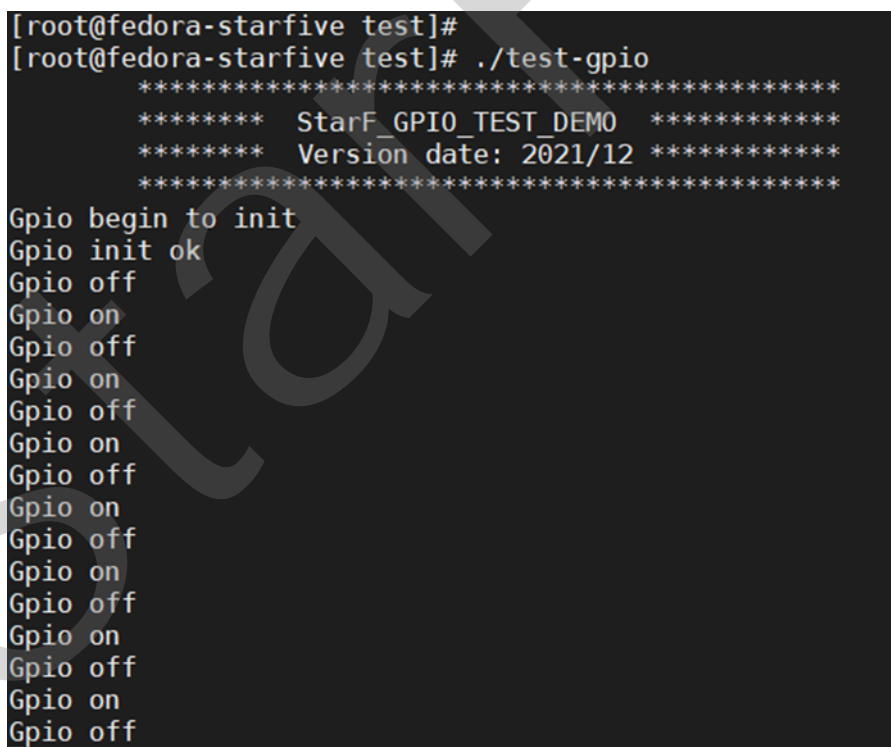
示例：

```
rsync ./test-gpio riscv@192.168.92.133:/home/riscv/test
```

2. 执行以下命令在昉·星光上运行演示代码：

```
./test-gpio
```

以下是输出示例：



```
[root@fedora-starfive test]#  
[root@fedora-starfive test]# ./test-gpio  
*****  
***** StarF_GPIO_TEST_DEMO *****  
***** Version date: 2021/12 *****  
*****  
Gpio begin to init  
Gpio init ok  
Gpio off  
Gpio on  
Gpio off  
Gpio on  
Gpio off  
Gpio on  
Gpio off  
Gpio on  
Gpio off  
Gpio on  
Gpio off  
Gpio on  
Gpio off  
Gpio on  
Gpio off  
Gpio on  
Gpio off
```

图 4-1 输出示例

说明：

- Gpio on：高电压
- Gpio off：低电压