



StarFive
赛昉科技

昉·星光 40-Pin GPIO Header用户手册

Version: V1.2

Date: 2022/05/05

Doc ID: VisionFive-UGCH-001-V1.2

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）

关于本手册

关于本手册和技术支持信息

简介

本文件旨在：






- 介绍40-Pin Header。
- 提供配置和调试GPIO、I2C、SPI、PWM和UART的说明。
- 提供使用40-Pin Header的外设示例。

修订历史

版本	已发布	修订
V1	2021/12/8	第一次正式发布。
V1.1	2021/12/27	<ul style="list-style-type: none">• 更新了“生成dtb文件 (第 12页)”部分中的命令，并完善了描述。• 在“GitHub仓库 (第 11页)”部分新增了一条注释。

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。

目录

表格清单.....	6
插图清单.....	7
法律声明.....	ii
关于本手册.....	iii
1. 概述.....	8
1.1. 40-Pin Header定义.....	8
2. GPIO引脚分布.....	9
3. 准备.....	10
3.1. 准备硬件.....	10
3.2. 准备软件.....	11
3.2.1. GitHub仓库.....	11
3.2.2. 将Fedora OS烧录到Micro SD卡.....	11
3.2.3. 生成dtb文件.....	12
3.2.4. 替换dtb文件.....	12
4. GPIO操作.....	15
4.1. 配置GPIO.....	15
5. I2C操作.....	17
5.1. 配置I2C GPIO.....	17
5.1.1. 硬件设置.....	17
5.1.2. 配置dts文件.....	19
5.2. 调试I2C.....	19
6. SPI操作.....	22
6.1. 配置SPI GPIO.....	22
6.1.1. 修改引脚.....	22
6.2. 调试SPI GPIO.....	23
6.2.1. 环回测试.....	23
6.2.2. 使用ADXL345模块测试SPI.....	25
7. PWM操作.....	28
7.1. 配置PWM GPIO.....	28
7.1.1. 修改引脚.....	28
7.1.2. PWM和引脚名映射.....	28
7.2. 调试PWM GPIO.....	28
8. UART操作.....	30

8.1. 配置UART GPIO.....	30
8.1.1. 修改dts文件.....	30
8.2. 调试UART GPIO.....	32
8.2.1. 硬件设置.....	33
8.2.2. 调试UART发送和接收功能.....	33
9. 外设示例.....	37
9.1. Sense Hat (B) 示例.....	37
9.1.1. 硬件设置.....	37
9.1.2. 示例.....	39
9.2. 2英寸LCD显示模块示例.....	40
9.2.1. 硬件设置.....	40



StarFive
赛昉科技

表格清单

表 2-1 GPIO引脚分布.....	9
表 3-1 硬件准备.....	10
表 3-2 GitHub仓库地址.....	11
表 3-3 dtb文件.....	12
表 7-1 PWM和引脚名映射.....	28
表 8-1 UART和DEV映射.....	32
表 9-1 将Sense Hat (B) 连接到40-Pin Header上.....	37
表 9-2 将2英寸LCD模块连接到40-Pin Header上.....	40



StarFive
赛昉科技

插图清单

图 1-1 40-Pin定义.....	8
图 3-1 识别到的目录.....	12
图 5-1 将Sense Hat (B)连接到扩展头.....	18
图 5-2 文件内容示例.....	19
图 5-3 输出示例.....	19
图 5-4 输出示例.....	20
图 5-6 输出示例.....	21
图 6-1 修改引脚.....	23
图 6-2 连接引脚18和16.....	24
图 6-3 返回示例.....	24
图 6-4 返回示例.....	25
图 6-5 将ADXL345模块连接到扩展头上.....	26
图 6-6 输出示例.....	27
图 7-1 示例文件内容.....	28
图 8-1 配置示例.....	30
图 8-2 配置示例.....	31
图 8-3 配置示例.....	32
图 8-4 将转换器连接到Header上.....	33
图 8-5 配置示例.....	33
图 8-6 配置示例.....	34
图 8-7 配置示例.....	35
图 8-8 输出示例.....	36
图 9-1 将Sense Hat (B) 连接到40-Pin Header上.....	38
图 9-2 将Sense Hat (B) 连接到40-Pin Header上.....	39
图 9-3 将2英寸LCD模块连接到40-Pin Header上.....	41
图 9-4 将2英寸LCD模块连接到40-Pin Header上.....	42
图 9-5 示例输出.....	43

1. 概述

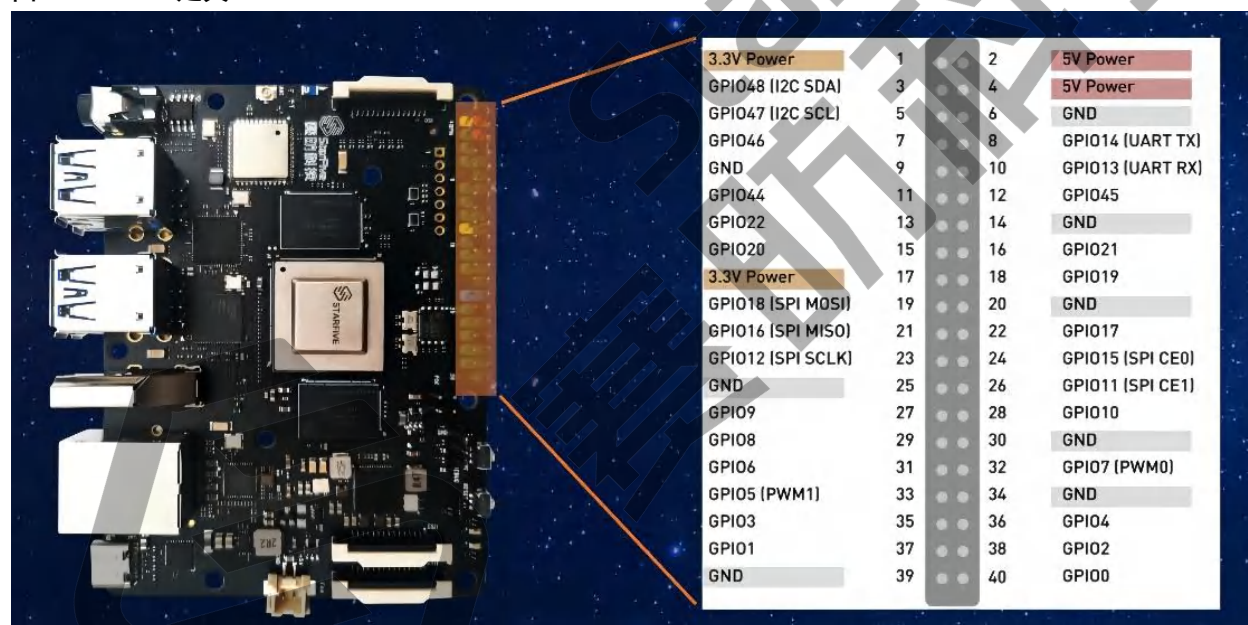
40-Pin Header使得星光板和昉·星光等赛昉科技单板计算机能够连接到各种外部组件。这是用户能够创建其项目的必要条件。本文件旨在：

- 介绍本章所述的40-Pin Header。
- 提供配置和调试GPIO、I2C、SPI和PWM的说明，如“[准备 \(第 10页\)](#)”、“[GPIO操作 \(第 15页\)](#)”、“[I2C操作 \(第 17页\)](#)”、“[SPI操作 \(第 22页\)](#)”、“[PWM操作 \(第 28页\)](#)”和“[UART操作 \(第 30页\)](#)”章节所述。
- 为如何使用40-Pin Header提供外设示例，如“[外设示例 \(第 37页\)](#)”一章所述。

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. GPIO引脚分布

下表介绍了GPIO引脚、映射图以及每个引脚的功能说明：

表 2-1 GPIO引脚分布

dts	sys	引脚名	序号	序号	引脚名	sys	dts
		3.3V电压	1	2	5V电压		
i2c1	i2c-1	GPIO48 (I2C SDA)	3	4	5V电压		
i2c1	i2c-1	GPIO47 (I2C SCL)	5	6	GND		
	494	GPIO46	7	8	GPIO14 (UART TX)	ttyS0	uart3
		GND	9	10	GPIO13 (UART RX)	ttyS0	uart3
	492	GPIO44	11	12	GPIO45	PWM2	
	470	GPIO22	13	14	GND		
	468	GPIO20	15	16	GPIO21	469	
		3.3V电压	17	18	GPIO19	467	
spi2	spidev0.0	GPIO18 (SPI MOSI)	19	20	GND		
spi2	spidev0.0	GPIO16 (SPI MISO)	21	22	GPIO17	465	
spi2	spidev0.0	GPIO12 (SPI SCLK)	23	24	GPIO15 (SPI CE0)	spidev0.0	spi2
		GND	25	26	GPIO11 (SPI CE1)	spidev0.0	spi2
	457	GPIO9	27	28	GPIO10	458	
	456	GPIO8	29	30	GND		
	454	GPIO6	31	32	GPIO7 (PWM0)	PWM0	
	PWM1	GPIO5 (PWM1)	33	34	GND		
	451	GPIO3	35	36	GPIO4	452	
	449	GPIO1	37	38	GPIO2	450	
		GND	39	40	GPIO0	448	

3. 准备

在配置和调试GPIO之前，需要确保完成以下准备：

3.1. 准备硬件

下表介绍了按照本指南配置、调试和测试40-Pin Header，需要准备的硬件项目：

表 3-1 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（Windows/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	这些项目用于将Fedora OS烧录到Micro SD上。
GPIO/PWM	O	一台示波器	示波器用于测量相应的引脚，并检查PWM周期和占空比。
I2C	O	<ul style="list-style-type: none">• Sense Hat (B)• 杜邦线	-
SPI	O	<ul style="list-style-type: none">• ADXL345模块• 杜邦线	-
UART	O	<ul style="list-style-type: none">• GNSS HAT• 杜邦线	基于MAX-7Q的GNSS HAT，支持GPS、GLONASS、QZSS、SBAS等定位系统。有关详细

表 3-1 硬件准备 (续)

类型	M/O	项目	注释
			规格, 请参考 MAX-7Q GNSS HAT 。
外设示例	O	<ul style="list-style-type: none"> • 2英寸LCD显示模块 • 杜邦线 	-

*M/O: M (必须) / O (可选)

3.2. 准备软件

在配置40-pin Header之前, 需将Fedora OS烧录到Micro SD卡中, 并且需要编译和替换dtb文件。按照如下步骤操作:

3.2.1. GitHub仓库

下表列出了GitHub仓库的地址:



注:

确保您已经切换到了对应分支。

表 3-2 GitHub仓库地址

类型	仓库	分支
Linux	Linux	visionfive-5.15.y
Linux仓库下的dts文件	<ul style="list-style-type: none"> • jh7100-common.dtsi • jh7100.dtsi 	-
Uboot	Uboot	JH7100_upstream
OpenSBI	OpenSBI	master
Fedora镜像 (Alpha版)	Fedora Image	-

3.2.2. 将Fedora OS烧录到Micro SD卡

本文介绍了两种烧录镜像的方式。一种应用于Mac或Linux, 另一种应用于Windows。按照[《昉·星光单板计算机快速参考手册》](#)将Fedora OS烧录到Micro SD卡”部分中的步骤操作。

3.2.3. 生成dtb文件

在Linux的根目录下执行以下命令，以编译生成设备树二进制文件：

```
make <Configuration_File> ARCH=riscv CROSS_COMPILE=riscv64-linux-gnu-
make CROSS_COMPILE=riscv64-linux-gnu- ARCH=riscv dtbs
```

i 提示：

<Configuration_File>可使用以下两个文件：`starfive_jh7100_fedora_defconfig`和`visionfive_defconfig`。

命令示例：

```
make starfive_jh7100_fedora_defconfig ARCH=riscv
CROSS_COMPILE=riscv64-linux-gnu-
make CROSS_COMPILE=riscv64-linux-gnu- ARCH=riscv dtbs
```

不同的单板计算机使用的dtb文件也各不相同。下表列出了单板计算机和dtb文件之间的对应关系：

表 3-3 dtb文件

单板计算机	文件
星光板	/linux/arch/riscv/boot/dts/starfive/jh7100-beaglev-starlight.dtb
昉·星光	/linux/arch/riscv/boot/dts/starfive/jh7100-starfive-visionfive-v1.dtb

3.2.4. 替换dtb文件

用于烧录镜像的SD卡能够自动识别出以下目录：

图 3-1 识别到的目录

```
/dev/sdb2 122M 4.5M 118M 4% /media/jianlong/DE31-0D9C
/dev/sdb3 458M 84M 360M 19% /media/jianlong/boot
/dev/sdb4 12G 7.0G 4.1G 64% /media/jianlong/
```

3.2.4.1. 方法1：直接替换dtb文件

在Linux的根目录下执行以下命令，以替换dtb文件：

```
sudo cp arch/riscv/boot/dts/starfive/<dtb_file>
<Mount_Directory>/__boot/dtbs/<Kernel_Version>/starfive
```

**注：**

- <dtb_file>为dtb文件的文件名。不同的单板计算机使用的dtb文件也各不相同。如需更多说明，请参阅“[dtb文件\(第12页\)](#)”。
- <Mount_Directory>为实际装载的路径。例如， /media/jianlong。
- <Kernel_Version>为Linux内核的版本号。例如， 5.14.0+。

命令示例：

```
sudo cp arch/riscv/boot/dts/starfive/jh7100-starfive-visionfive-v1.dtb
/media/jianlong/___boot/dtbs/5.14.0+/starfive
```

3.2.4.2. 方法2：添加启动项

执行以下操作，添加启动项，以替换dtb文件：

1. 在Linux根目录下，执行以下命令：

```
sudo cp arch/riscv/boot/dts/starfive/<dtb_file>
<Mount_Direcotry>/boot/dtbs/
```

**注：**

- <dtb_file>为dtb文件的文件名。不同的单板计算机使用的dtb文件也各不相同。如需更多说明，请参阅本文档中的“[dtb文件\(第12页\)](#)”表格。
- <Mount_Directory>为实际装载的路径。例如， /media/jianlong。

命令示例：

```
sudo cp
arch/riscv/boot/dts/starfive/jh7100-starfive-visionfive-v1.dtb /
media/jianlong/___boot/dtbs/
```

2. 进入SD卡的装载目录：

```
cd <Mount_Direcotry>/___boot
```

**注：**

<Mount_Directory>为实际装载的路径。例如， /media/jianlong。

3. 打开grub.cfg文件：

```
sudo gedit grub.cfg
```

4. 添加内容，保存并退出：

```
menuentry 'MY Fedora vmlinuz-5.14.0+' {
  linux /vmlinuz-5.14.0+ ro
  root=UUID=f852f7f6-aa4e-4404-8ea9-439568b767a1 rhgb console=tty0
  console=ttyS0,115200 earlycon rootwait stmmaceth=chain_mode:1
  selinux=0
  LANG=en_US.UTF-8
  devicetree /dtbs/<dtb_File>
  initrd /initramfs-<Kernel_Version>.img
}
```



注：

- <dtb_File>为单板计算机使用的dtb文件的名称。例如，jh7100-starfive-visionfive-v1.dtb。如需单板计算机和dtb文件之间的对应关系，参阅本文档中的“[dtb文件 \(第 12页\)](#)”表格。
- <Kernel_Version> 指Linux内核的版本号。例如，5.14.0+。
- MY Fedora vmlinuz-5.14.0：可配置菜单项名称。

5. 当系统启动到grub启动项时，选择上一步中设置的菜单项，例如MY Fedora vmlinuz-5.14.0+。



注：

可以根据dtb文件的实际数量，添加多个启动项。

4. GPIO操作

本节提供配置GPIO的命令：

4.1. 配置GPIO

执行以下操作配置GPIO：

1. 执行以下命令以配置GPIO0：

```
cd /sys/class/gpio  
echo 448 > export
```



命令中的448为sys编号，对应GPIO0。如需GPIO引脚名的更多详细说明，请参阅本文档中的“[GPIO引脚分布 \(第 9页\)](#)”部分。

2. 进入GPIO0目录：

```
cd gpio448
```

3. 将GPIO0的方向设为输入：

```
echo in > direction
```

4. 或者，将GPIO0的方向设为输出：

```
echo out > direction
```

5. 将GPIO0的电压设为高电平：

```
echo 1 > value
```



可以用示波器检测电压。

6. 将GPIO0的电压设为低电平：

```
echo 0 > value
```



可以用示波器检测电压。

7. 将**3.3V**电源引脚连接到GPIO0上，并检测GPIO0的电压：

```
cat value
```

8. 将**GND**引脚连接到GPIO0上，并检测GPIO0的电压：

```
cat value
```



5. I2C操作

本章介绍如何配置和调试I2C GPIO。

5.1. 配置I2C GPIO

系统支持4路I2C总线：i2c0、i2c1、i2c2和i2c3。

执行以下操作，以配置I2C：

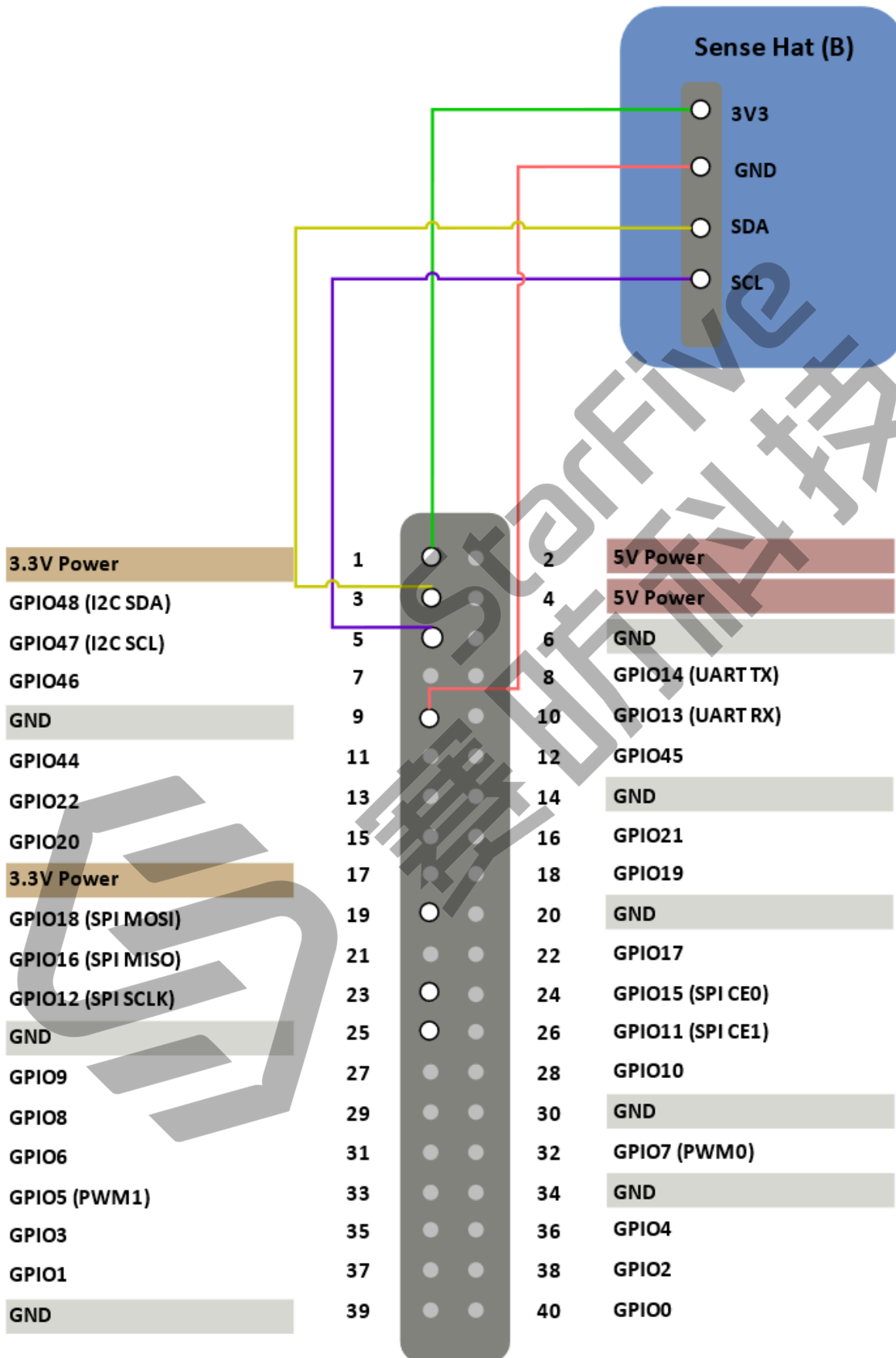
5.1.1. 硬件设置

将Sense Hat (B)连接到以下Header：



StarFive
赛昉科技

图 5-1 将Sense Hat (B)连接到扩展头



5.1.2. 配置dts文件

修改位于/linux/arch/riscv/boot/dts/starfive路径下的，jh7100-common.dtsi文件的文件内容。以下是默认设置。您可以根据需要配置未被占用的GPIO。

图 5-2 文件内容示例

```

184     i2c1_pins: i2c1-0 {
185         i2c-pins {
186             pinmux = <GPIOMUX(47) GPO_LOW,
187                 GPO_I2C1_PAD_SCK_OEN,
188                 GPI_I2C1_PAD_SCK_IN)>,
189                 <GPIOMUX(48) GPO_LOW,
190                 GPO_I2C1_PAD_SDA_OEN,
191                 GPI_I2C1_PAD_SDA_IN)>;
192             bias-pull-up;
193             input-enable;
194             input-schmitt-enable;
195         };
196     };

```



注：

I2C GPIO引脚编号是引脚名称中指示的编号。如需GPIO引脚名的更多详细说明，请参阅本文档中的“[GPIO引脚分布 \(第 9页\)](#)”部分。I2C GPIO的引脚名如下：

- GPIO48 (I2C SDA)
- GPIO47 (I2C SCL)

5.2. 调试I2C

执行以下步骤，以调试I2C：

1. 执行以下命令，以扫描总线：

```
i2cdetect -l
```

结果：

图 5-3 输出示例

```

# i2cdetect -l
i2c-1  i2c          Synopsys DesignWare I2C adapter      I2C adapter
i2c-0  i2c          Synopsys DesignWare I2C adapter      I2C adapter
# i2cdetect -l

```

2. 执行以下命令，以检测设备：

```
i2cdetect -y -r 1
```



注：

1为I2C总线序号。

结果：

图 5-4 输出示例

```
[root@fedora-starfive /]# i2cdetect -y -r 1
    ..... 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  29  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  48  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  5c  --  --  --
60:  --  --  --  --  --  --  --  --  68  --  --  --  --  --  --
70:  70  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

本图中，检测到的设备为：0x29、0x48、0x5c、0x68和0x70。

3. 执行以下命令，读取寄存器内容：

```
i2cget -f -y 1 0x5c 0x0f
```



注：

- 1: I2C总线序号。
- 0x5c: I2C设备地址。
- 0x0f: 从器件内存地址。

结果：

```
# i2cget -f -y 1 0x5c 0x0f
0xb1
#
```

示例输出中对应的寄存器值为0xb1。

4. 执行以下命令，写寄存器数据：

```
i2cset -y 1 0x5c 0x11 0x10
```

 注：

- 1: I2C总线序号。
- 0x5c: I2C设备地址。
- 0x11: 从器件内存地址。
- 0x10: 需要写入寄存器的内容。

5. 执行以下命令，读取所有寄存器值：

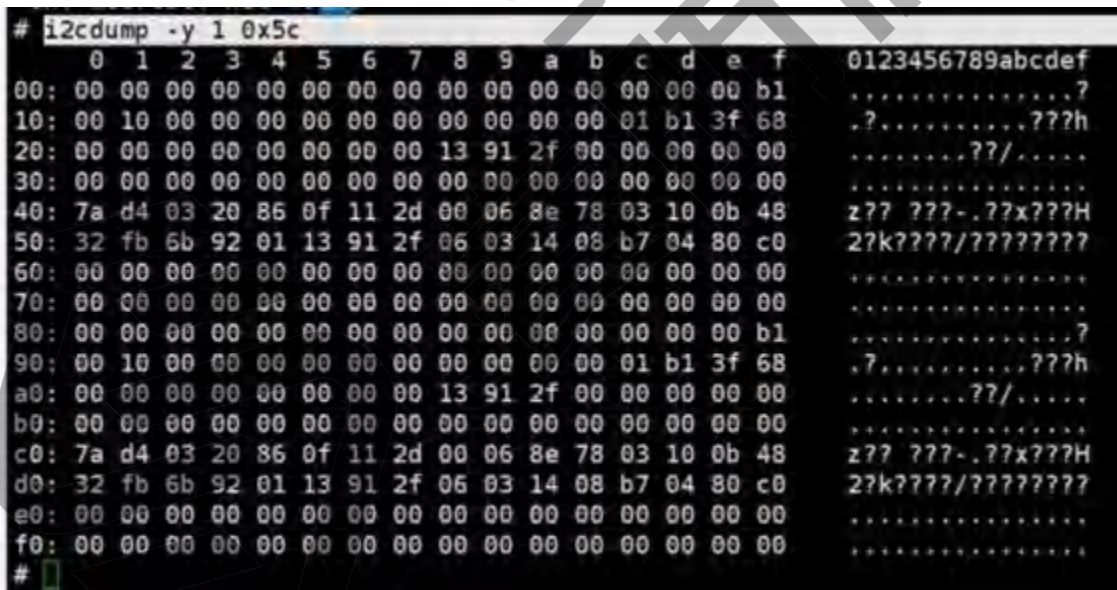
```
i2cdump -y 1 0x5c
```

 注：

- 1: I2C总线序号
- 0x5c: I2C设备地址

结果：

图 5-6 输出示例



```
# i2cdump -y 1 0x5c
# 0 1 2 3 4 5 6 7 8 9 a b c d e f 0123456789abcdef
00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 b1 .....?
10: 00 10 00 00 00 00 00 00 00 00 00 00 01 b1 3f 68 .?.....??h
20: 00 00 00 00 00 00 00 00 13 91 2f 00 00 00 00 00 .....??/....
30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
40: 7a d4 03 20 86 0f 11 2d 00 06 8e 78 03 10 0b 48 z?? ???-..??x??H
50: 32 fb 6b 92 01 13 91 2f 06 03 14 08 b7 04 80 c0 2?k????/????????
60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....?
90: 00 10 00 00 00 00 00 00 00 00 00 00 00 01 b1 3f 68 .?.....??h
a0: 00 00 00 00 00 00 00 00 13 91 2f 00 00 00 00 00 .....??/....
b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
c0: 7a d4 03 20 86 0f 11 2d 00 06 8e 78 03 10 0b 48 z?? ???-..??x??H
d0: 32 fb 6b 92 01 13 91 2f 06 03 14 08 b7 04 80 c0 2?k????/????????
e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
#
```

6. SPI操作

本章介绍如何配置和调试SPI GPIO。

6.1. 配置SPI GPIO

系统支持2路SPI总线：spi2和spi3。

配置文件，jh7100-common.dtsi，位于以下目录：`/linux/arch/riscv/boot/dts/starfive`。

6.1.1. 修改引脚

配置的SPI GPIO编号即“引脚名”中的编号。如需GPIO引脚名的更多详细说明，请参阅本文档中的“[GPIO引脚分布 \(第 9页\)](#)”部分。未被占用的引脚均可配置。以下是jh7100-common.dtsi文件中的默认设置：

图 6-1 修改引脚

```

284 spi2_pins: spi2-0 {
285     mosi-pin {
286         pinmux = <GPIOMUX(18, GPO_SPI2_PAD_TXD,
287             GPO_ENABLE, GPI_NONE)>;
288         bias-disable;
289         input-disable;
290         input-schmitt-disable;
291     };
292     miso-pin {
293         pinmux = <GPIOMUX(16, GPO_LOW, GPO_DISABLE,
294             GPI_SPI2_PAD_RXD)>;
295         bias-pull-up;
296         input-enable;
297         input-schmitt-enable;
298     };
299     sck-pin {
300         pinmux = <GPIOMUX(12, GPO_SPI2_PAD_SCK_OUT,
301             GPO_ENABLE, GPI_NONE)>;
302         bias-disable;
303         input-disable;
304         input-schmitt-disable;
305     };
306     ss-pins {
307         pinmux = <GPIOMUX(15, GPO_SPI2_PAD_SS_0_N,
308             GPO_ENABLE, GPI_NONE)>,
309             <GPIOMUX(11, GPO_SPI2_PAD_SS_1_N,
310             GPO_ENABLE, GPI_NONE)>;
311         bias-disable;
312         input-disable;
313         input-schmitt-disable;
314     };
315 };

```

6.2. 调试SPI GPIO

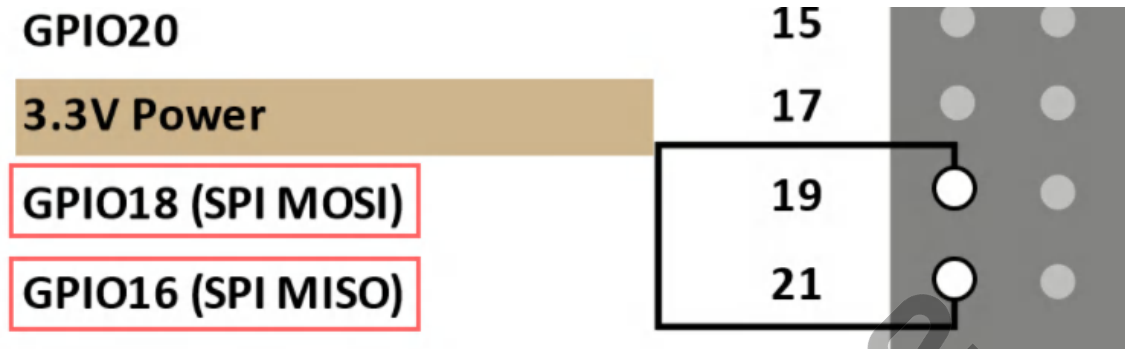
本节提供了环回测试的步骤，以及使用ADXL345模块测试SPI的步骤。

6.2.1. 环回测试

按照以下步骤，执行环回测试：

1. 接线：按照以下步骤连接引脚18和16：

图 6-2 连接引脚18和16



2. 进入以下路径，找到测试工具spidev_test.c：

```
cd /linux/tools/spi
```

3. 在测试工具目录下，执行以下命令：

```
make CROSS_COMPILE=riscv64-linux-gnu- ARCH=riscv
```

结果：

系统将在同一目录生成名为spidev_test的文件。

4. 将spidev_test文件上传到单板计算机上，例如星光板，并执行以下命令，更改执行权限：

```
chmod +x spidev_test
```

5. 确认SPI设备。

```
ls /dev/spidev*
```

示例：

图 6-3 返回示例

```
[root@fedora-starfive ~]# ls /dev/spidev*
/dev/spidev0.0
[root@fedora-starfive ~]#
```

6. 执行以下命令，以运行测试：

```
./spidev_test -D /dev/spidev0.0 -v -p string_to_send
```



注：

spidev0.0即上一步中获取的设备名。

结果：

图 6-4 返回示例

```
# ./spidev_test -D /dev/spidev1.0 -v -p string_to_send
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | 73 74 72 69 6E 67 5F 74 6F 5F 73 65 6E 64 | string_to_
sen
RX | 73 74 72 69 6E 67 5F 74 6F 5F 73 65 6E 64 | string_to_
send
```

如测试结果和本图中红色框内的部分显示相同，接收的数据和发送的完全一致，则表示测试成功。

6.2.2. 使用ADXL345模块测试SPI

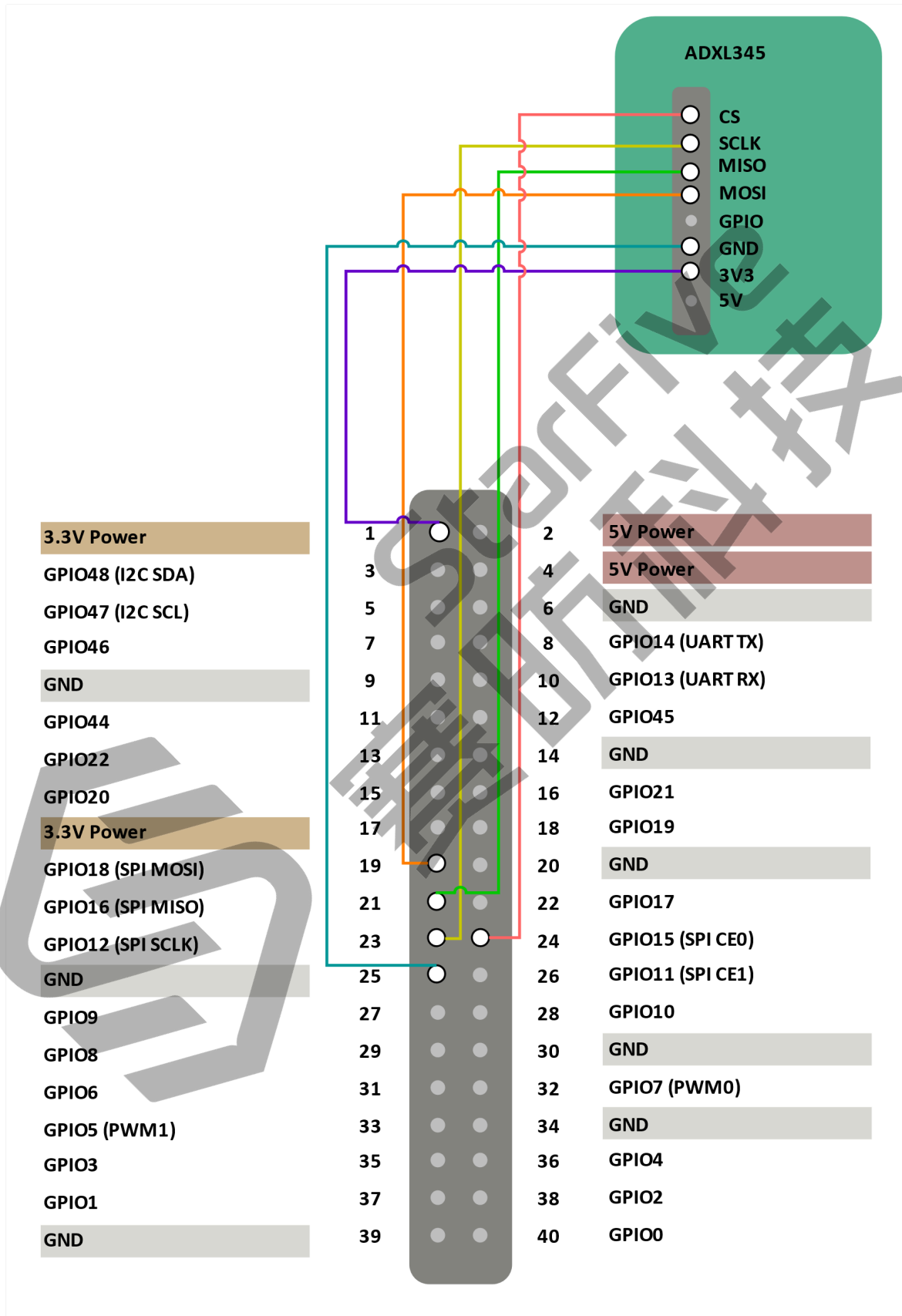
按照以下步骤，使用ADXL345模块测试SPI：



StarFive
赛昉科技

1. 按照以下步骤，将ADXL345模块连接到40-Pin Header上：

图 6-5 将ADXL345模块连接到扩展头上



2. 进入测试工具spidev_test.c所在的目录：

```
cd /linux/tools/spi
```

3. 在测试工具目录下，执行以下命令：

```
make CROSS_COMPILE=riscv64-linux-gnu- ARCH=riscv
```

结果：

系统将在同一目录生成名为spidev_test的文件。

4. 将spidev_test文件上传到单板计算机上，例如星光板，并执行以下命令，更改执行权限：

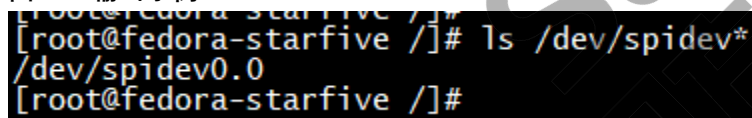
```
chmod +x spidev_test
```

5. 确认SPI设备。

```
ls /dev/spidev*
```

示例：

图 6-6 输出示例



```
[root@fedora-starfive /]# ls /dev/spidev*
/dev/spidev0.0
[root@fedora-starfive /]#
```

6. 执行以下命令，读取设备ID：

```
./spidev_test -H -O -D /dev/spidev0.0 -v -p \\x80\\x00
```

7. 执行以下命令，读取多个寄存器的值：

```
./spidev_test -H -O -D /dev/spidev0.0 -v -p
\\xec\\x00\\x00\\x00\\x00\\x00\\x00
```

8. 执行以下命令，以读取寄存器值：

```
./spidev_test -H -O -D /dev/spidev0.0 -v -p \\x9e\\x00
```

9. 执行以下命令，以写入寄存器值：

```
./spidev_test -H -O -D /dev/spidev0.0 -v -p \\x1e\\xaa
```

10. 执行以下命令，以读取验证说明：

```
./spidev_test -H -O -D /dev/spidev0.0 -v -p \\x9e\\x00
```

7. PWM操作

本章介绍如何配置和调试PWM GPIO:

7.1. 配置PWM GPIO

系统支持最多8路PWM。

配置文件jh7100-common.dtsi, 位于以下目录:

```
/linux/arch/riscv/boot/dts/starfive
```

7.1.1. 修改引脚

下图显示了用于修改引脚的示例文件的内容:

图 7-1 示例文件内容

```
358     pwm_pins: pwm_pins-0 {
359         ptc-pins {
360             pinmux = <GPIOMUX(7, GPO_PWM_PAD_OUT_BIT0, GPO_PWM_PAD_OE_N_BIT0, GPI_NONE)>,
361                    <GPIOMUX(5, GPO_PWM_PAD_OUT_BIT1, GPO_PWM_PAD_OE_N_BIT1, GPI_NONE)>,
362                    <GPIOMUX(45, GPO_PWM_PAD_OUT_BIT2, GPO_PWM_PAD_OE_N_BIT2, GPI_NONE)>;
363             bias-disable;
364             drive-strength = <35>;
365             input-disable;
366             input-schmitt-disable;
367             slew-rate = <0>;
368         };
369     };
370 };
```

配置的PWM GPIO编号即“引脚名”中的编号。如需GPIO引脚名的更多详细说明, 请参阅本文档中的[“GPIO引脚分布 \(第 9页\)”](#)部分。

7.1.2. PWM和引脚名映射

下表介绍了PWM和引脚名称的映射关系:

表 7-1 PWM和引脚名映射

PWM	GPIO (引脚名)
PWM0	GPIO7
PWM1	GPIO5
PWM2	GPIO45

7.2. 调试PWM GPIO

本章介绍如何调试PWM GPIO:

1. 执行以下命令，以配置PWM通道：

```
cd /sys/class/pwm/pwmchip0  
echo 0 > export
```

2. 执行以下命令，以配置PWM周期（单位：ns）：

```
cd pwm0  
echo 5000000 > period
```

3. 执行以下命令，以配置高电平时间（单位：ns）：

```
echo 1000000 > duty_cycle
```

4. 使用示波器测量对应引脚，并检查PWM周期和占空比。



注：

占空比为高电平在一个周期之内所占的时间比率。

8. UART操作

本章介绍如何配置和调试UART GPIO:

8.1. 配置UART GPIO

系统支持最多4路UART:

- UART3用于调试。
- UART0用于蓝牙。
- 可使用UART1和UART2。

配置文件jh7100-common.dtsi, 位于以下目录:

```
/linux/arch/riscv/boot/dts/starfive
```

配置的UART GPIO编号即“引脚名”中的编号。如需GPIO引脚名的更多详细说明, 请参阅本文档中的[“GPIO引脚分布” \(第 9页\)](#)部分。

8.1.1. 修改dts文件

按照以下步骤, 修改dts文件:

1. 在aliases节点上添加uart1或uart2的alias。以下是一个示例:

图 8-1 配置示例

```
/{
    aliases {
        mshc0 = &sdio0;
        mshc1 = &sdio1;
        serial0 = &uart3;
        serial1 = &uart0;
        serial2 = &uart1;
        serial3 = &uart2;
    };
    chosen {
        stdout-path = "serial0:115200n8";
    };
    cpus {
        timebase-frequency = <6250000>;
    };
};
```

2. 在dts上添加uart1或uart2节点。以下是一个示例:

图 8-2 配置示例

```
&uart3 {  
    pinctrl-names = "default";  
    pinctrl-0 = <&uart3_pins>;  
    status = "okay";  
};  
  
&uart1 {  
    pinctrl-names = "default";  
    pinctrl-0 = <&uart1_pins>;  
    status = "okay";  
};  
  
&uart2 {  
    pinctrl-names = "default";  
    pinctrl-0 = <&uart2_pins>;  
    status = "okay";  
};
```

3. 在&gpio节点上添加uart1_pins或uart2_pins节点。

配置的UART GPIO编号即“引脚名”中的编号，未被占用的引脚均可配置。如需GPIO引脚名的更多详细说明，请参阅本文档中的“[GPIO引脚分布 \(第 9页\)](#)”部分。

图 8-3 配置示例

```

};
};
uart1_pins: uart1-0 {
    rx-pins {
        pinmux = <GPIOMUX(1, GPO_LOW, GPO_DISABLE,
            GPI_UART1_PAD_SIN)>;
        bias-pull-up;
        drive-strength = <14>;
        input-enable;
        input-schmitt-enable;
    };
    tx-pins {
        pinmux = <GPIOMUX(3, GPO_UART1_PAD_SOUT,
            GPO_ENABLE, GPI_NONE)>;
        bias-disable;
        drive-strength = <35>;
        input-disable;
        input-schmitt-disable;
    };
};
uart2_pins: uart2-0 {
    rx-pins {
        pinmux = <GPIOMUX(0, GPO_LOW, GPO_DISABLE,
            GPI_UART2_PAD_SIN)>;
        bias-pull-up;
        drive-strength = <14>;
        input-enable;
        input-schmitt-enable;
    };
    tx-pins {
        pinmux = <GPIOMUX(2, GPO_UART2_PAD_SOUT,
            GPO_ENABLE, GPI_NONE)>;
        bias-disable;
        drive-strength = <35>;
        input-disable;
        input-schmitt-disable;
    };
};
};
&i2c0 {
    clock-frequency = <100000>;
};

```

8.1.1.1. UART和DEV映射

下表描述了UART和设备文件（DEV）的映射关系：

表 8-1 UART和DEV映射

UART	DEV
UART1	/dev/ttyS2
UART2	/dev/ttyS3

8.2. 调试UART GPIO

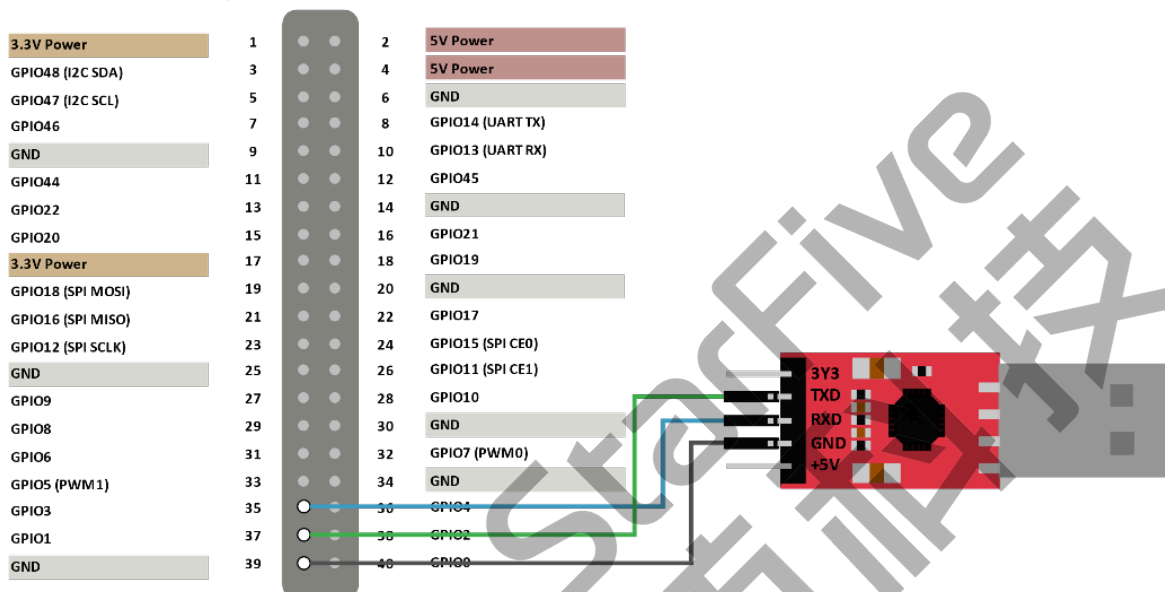
8.2.1. 硬件设置

按照如下步骤设置硬件：

步骤：

1. 将USB转串口转换器的跳线连接到昉·星光的40-Pin GPIO Header上，如下所示。

图 8-4 将转换器连接到Header上



2. 将USB转串口转换器的另一端连接到设备（Windows/Mac/Linux）。

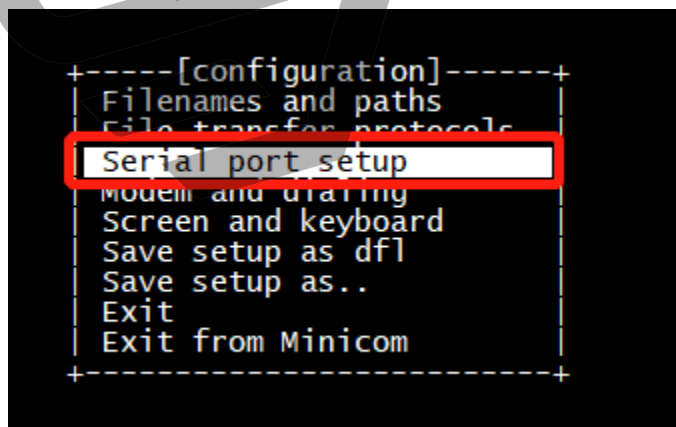
8.2.2. 调试UART发送和接收功能

1. 配置昉·星光的minicom

```
sudo minicom -s
```

2. 选择Serial port setup，并按照如下方式，配置minicom：

图 8-5 配置示例



```
+-----+
| A -   Serial Device       /dev/ttyS2
| C -   Callin Program
| D -   Callout Program
| E -   Bps/Par/Bits        115200 8N1
| F -   Hardware Flow Control No
| G -   Software Flow Control No
|
| Change which setting?
|
| Screen and keyboard
| Save setup as dfl
| Save setup as..
| Exit
| Exit from Minicom
+-----+
```

3. 在PC上，输入以下命令，启动昉·星光：

```
minicom -o -D /dev/ttyS2
```

结果：

图 8-6 配置示例

```
Welcome to minicom 2.7.1
OPTIONS: I18n
Compiled on Jul 28 2020, 00:00:00.
Port /dev/ttyS2, 08:41:28

Press CTRL-A Z for help on special keys
```

4. 输入以下命令，配置Ubuntu minicom：

```
sudo minicom -s
```

1. 选择Serial port setup, 并按照如下方式, 配置minicom:

图 8-7 配置示例

```

+-----[configuration]-----+
| Filenames and paths          |
| File transfer protocols      |
| Serial port setup          |
| Modem and dialing           |
| Screen and keyboard         |
| Save setup as dfl           |
| Save setup as..             |
| Exit                         |
| Exit from Minicom           |
+-----+

A - Serial Device      : /dev/ttyUSB0
B - Lockfile Location  : /var/lock
C - Callin Program    :
D - Callout Program   :
E - Bps/Par/Bits      : 115200 8N1
F - Hardware Flow Control : No
G - Software Flow Control : No

Change which setting? █

| Screen and keyboard         |
| Save setup as dfl           |
| Save setup as..             |
| Exit                         |
| Exit from Minicom           |
+-----+

```



注:

在Ubuntu上, 可以使用 `dmesg | grep tty` 命令检测串口设备

```

jlanlong@ubuntu:~$ dmesg | grep tty
[ 0.004000] console [tty0] enabled
[ 1.846654] 00:05: ttyS0 at I/O 0x3f8 (irq = 4, base_baud = 115200) is a 16550A
[30998.431828] usb 3-2: FTDI USB Serial Device converter now attached to ttyUSB0

```

启动Ubuntu minicom，您可以观察到如下输出：

图 8-8 输出示例

```

Welcome to minicom 2.7.1

OPTIONS: I18n
Compiled on Aug 13 2017, 15:25:34.
Port /dev/ttyUSB0, 08:40:51

Press CTRL-A Z for help on special keys

```

测试UART发送功能：

要测试UART发送功能，可以在昉·星光的minicom上输入字符，例如Hello Ubuntu。然后，您可以看到，该字符显示在Ubuntu minicom上，如下所示：

<pre> Welcome to minicom 2.7.1 OPTIONS: I18n Compiled on Aug 13 2017, 15:25:34. Port /dev/ttyUSB0, 09:01:15 Press CTRL-A Z for help on special keys hello ubuntu </pre>	<pre> Report bugs to <minicom-devel@lists.aliases.debian.org>. [root@fedora-starfive /]# minicom -o -D /dev/ttyS2 F Welcome to minicom 2.7.1 OPTIONS: I18n Compiled on Jul 28 2020, 00:00:00. Port /dev/ttyS2 Press CTRL-A Z for help on special keys </pre>
--	--

- 左图：Ubuntu minicom接口
- 右图：昉·星光minicom interface

测试UART接收功能：

输入字符即可测试UART信号接收，例如在Ubuntu minicom上输入Hello VisionFive。然后，您将观察在昉·星光的minicom上输出的字符：

<pre> Welcome to minicom 2.7.1 OPTIONS: I18n Compiled on Aug 13 2017, 15:25:34. Port /dev/ttyUSB0, 09:01:15 Press CTRL-A Z for help on special keys </pre>	<pre> Report bugs to <minicom-devel@lists.aliases.debian.org>. [root@fedora-starfive /]# minicom -o -D /dev/ttyS2 F Welcome to minicom 2.7.1 OPTIONS: I18n Compiled on Jul 28 2020, 00:00:00. Port /dev/ttyS2 Press CTRL-A Z for help on special keys hello visionfive </pre>
---	--

- 左图：Ubuntu minicom接口
- 右图：昉·星光minicom接口

9. 外设示例

在本演示中使用了Sense Hat (B)。更多详细规格，请参考：

[https://www.waveshare.com/wiki/Sense_HAT_\(B\)](https://www.waveshare.com/wiki/Sense_HAT_(B))。



注：

系统不支持BCM2835、Python和wiringPi的官方库，这里，我们将采用系统调用。示例需要按照实际情况修改。

9.1. Sense Hat (B) 示例

9.1.1. 硬件设置

下表和下图描述了如何将Sense HAT连接到40-Pin Header上：

表 9-1 将Sense Hat (B) 连接到40-Pin Header上

Sense HAT (B)	引脚序号
3V3	1
GND	9
SDA	3
SCL	5

图 9-1 将Sense Hat (B) 连接到40-Pin Header上

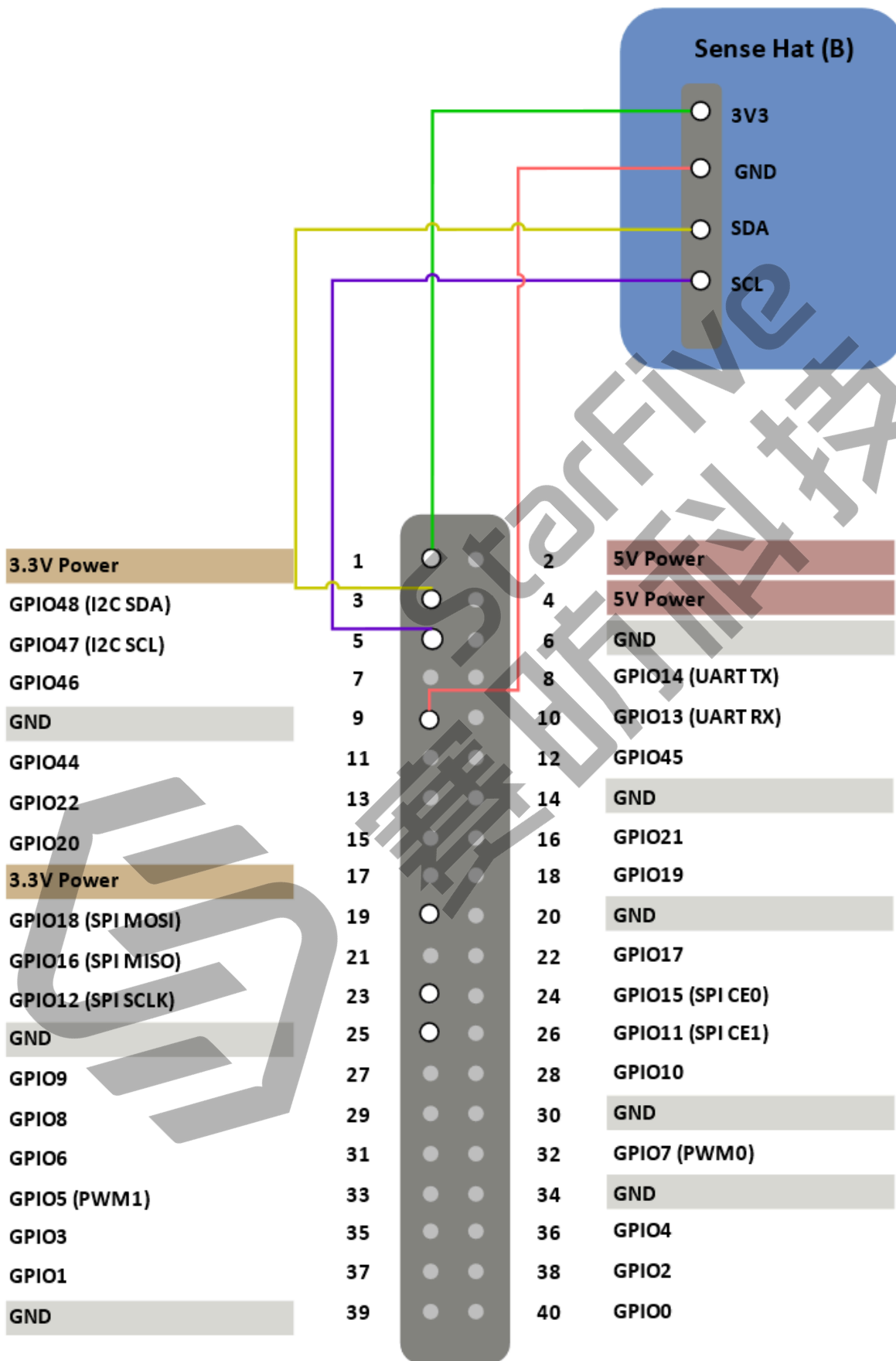


图 9-2 将Sense Hat (B) 连接到40-Pin Header上



9.1.2. 示例

以SHTC3传感器为例：

1. 从以下网址下载源代码：[SHTC3_dev.c](https://github.com/starfivetech/SHTC3_dev.c)
2. (可选) 安装编译工具。以下是安装示例：

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



注：

如已安装该工具，则可跳过此步。

3. 执行以下命令进行编译：

```
riscv64-linux-gnu-gcc SHTC3_dev.c -o shtc3
```

结果：

系统将在同一目录生成名为shtc3的文件。

4. 复制shtc3文件中的可执行代码，保存到单板计算机上，并执行以下命令，更改执行权限：

```
chmod +x shtc3
```

5. 执行以下命令，运行示例：

```
./shtc3
```

结果:

如显示以下结果，则表示执行成功：

```
[root@fedora-starfive riscv]# ./shtc3
SHTC3 Sensor Test Program ...
Fopen : /dev/i2c-1
Temperature = 75.61°C , Humidity = 68.55
Temperature = 27.40°C , Humidity = 68.54
Temperature = 27.40°C , Humidity = 68.55
Temperature = 27.40°C , Humidity = 68.54
Temperature = 27.39°C , Humidity = 68.54
```

9.2. 2英寸LCD显示模块示例

本示例中使用了2英寸LCD显示模块。如需详细规格，参考以下说明：

https://www.waveshare.com/wiki/2inch_LCD_Module



注:

官方示例需要根据实际情况进行修改。

9.2.1. 硬件设置

下表和下图描述了如何将2英寸LCD模块连接到40-Pin Header上：

表 9-2 将2英寸LCD模块连接到40-Pin Header上

2英寸LCD模块	引脚序号
VCC	17
GND	39
DIN	19
CLK	23
CS	28
DC	22
RST	13
BL	18

图 9-3 将2英寸LCD模块连接到40-Pin Header上

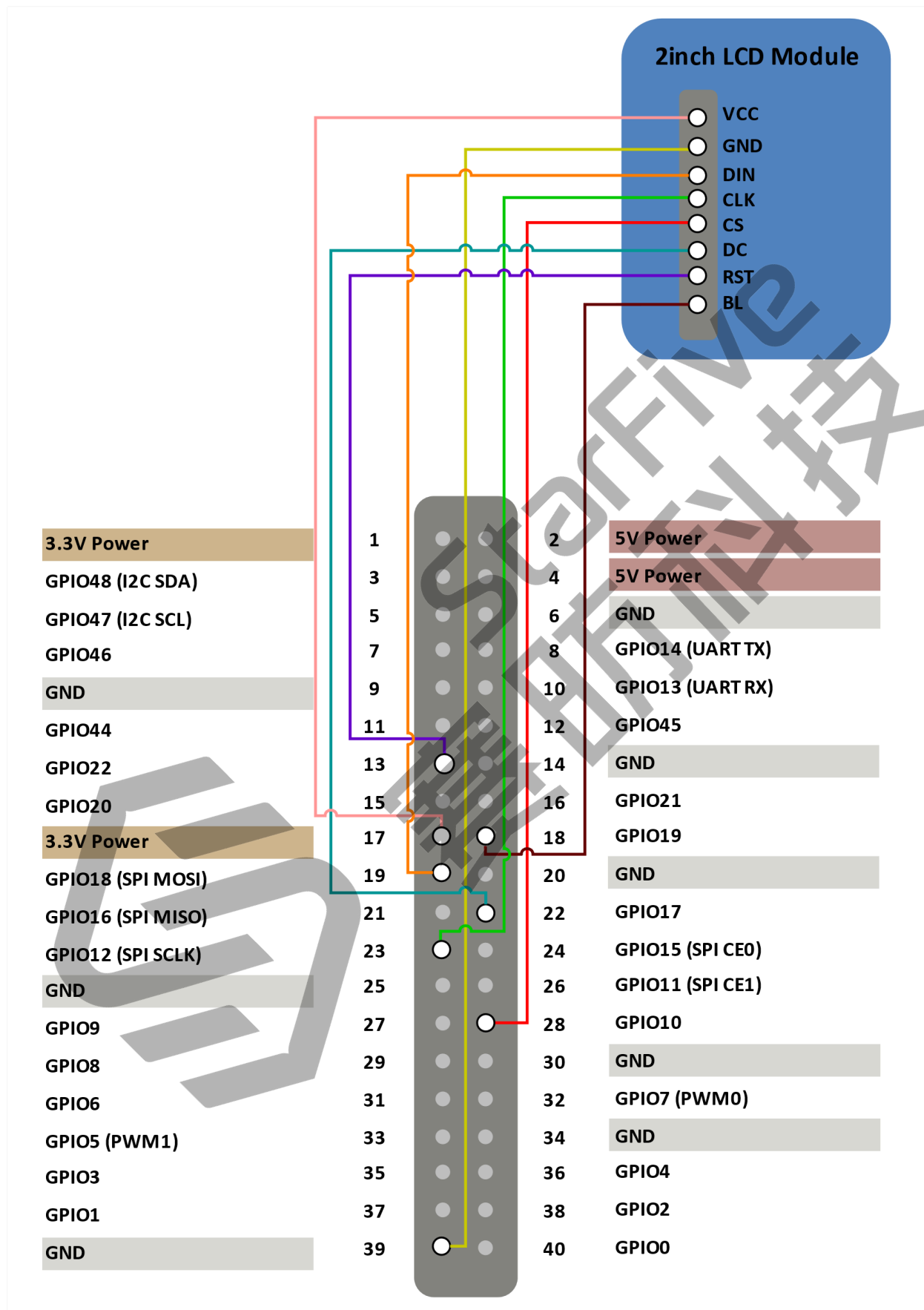


图 9-4 将2英寸LCD模块连接到40-Pin Header上



9.2.1.1. 运行示例

按照以下步骤，运行示例：

1. 从以下网址下载源代码：visionfive.tar.gz。
2. 执行以下命令将代码复制到单板计算机上。例如，在昉·星光上，执行以下命令。

```
tar -xvf visionfive.tar.gz
cd visionfive/
./main 2
```

结果：

您会观察到LCD模块上轮流显示两张图，一张为开发板的照片，另一张为官方示例图：

图 9-5 示例输出

