

StarFive
赛昉科技

昉·星光 2 40-Pin GPIO Header 用户手册

版本：1.3

日期：2023/03/31

Doc ID：VisionFive 2-UGCH-001

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责任及所有连带责任，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 GPIO Header的外设示例。

修订历史

表 0-1 修订历史

版本	发布说明	修订
1.0	2022/12/21	首次发布。
1.1	2022/12/27	更新了 GitHub代码仓 (第 13页) 中的DTS文件链接。
1.2	2023/03/16	更新步骤 <ul style="list-style-type: none">• 方法1：直接替换DTB文件 (第 15页)• 方法2：添加启动项 (第 16页)
1.3	2023/3/31	更新了 连接硬件 (第 35页) 的图片。

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。



StarFive 星五

目录

表格清单.....	7
插图清单.....	8
法律声明.....	ii
前言.....	iii
1. 概述.....	9
1.1. 40-Pin Header定义.....	9
2. GPIO Pin分布.....	10
3. 准备.....	12
3.1. 准备硬件.....	12
3.2. 准备软件.....	13
3.2.1. GitHub代码仓.....	13
3.2.2. 将Debian OS烧录到Micro-SD卡上.....	14
3.2.3. 生成DTB文件.....	14
3.2.4. 替换DTB文件.....	14
4. GPIO操作.....	19
4.1. 配置GPIO.....	19
5. I2C操作.....	21
5.1. 配置I2C GPIO.....	21
5.1.1. 连接硬件.....	21
5.1.2. 配置dts文件.....	22
5.2. 调试I2C GPIO.....	23
6. SPI操作.....	26
6.1. 配置SPI GPIO.....	26
6.1.1. 修改Pin.....	26
6.2. 调试SPI GPIO.....	27
6.2.1. 环回测试.....	27
6.2.2. 使用ADXL345模块测试SPI.....	28
7. PWM操作.....	31
7.1. 配置PWM GPIO.....	31
7.1.1. 修改Pin.....	31
7.1.2. PWM和pin Name映射.....	31
7.2. 调试PWM GPIO.....	32
8. UART操作.....	33

目录

8.1. 配置UART GPIO.....	33
8.1.1. 修改dts文件.....	33
8.2. 调试UART GPIO.....	35
8.2.1. 连接硬件.....	35
8.2.2. 调试UART发送和接收功能.....	35
9. 外设示例.....	39
9.1. Sense Hat (B) 示例.....	39
9.1.1. 连接硬件.....	39
9.1.2. Sense Hat (B)运行示例.....	40
9.2. 2.4英寸LCD显示模块示例.....	41
9.2.1. 连接硬件.....	42
9.2.2. 运行示例.....	44

表格清单

表 0-1 修订历史.....	iii
表 2-1 GPIO Pin分布.....	10
表 3-1 硬件准备.....	12
表 3-2 GitHub代码仓地址.....	13
表 5-1 将Sense Hat (B) 连接到40-Pin GPIO Header上.....	21
表 7-1 PWM和Pin Name映射.....	31
表 8-1 UART和DEV映射.....	35
表 9-1 将Sense Hat (B) 连接到40-Pin GPIO Header上.....	39
表 9-2 将2.4英寸LCD模块连接到40-Pin Header上.....	42

插图清单

图 1-1 40-Pin 定义.....	9
图 3-1 示例.....	16
图 5-1 将Sense Hat (B) 连接到40-Pin GPIO Header上.....	22
图 5-2 文件内容示例.....	23
图 5-3 示例输出.....	23
图 5-4 示例输出.....	24
图 5-5 示例输出.....	24
图 5-6 示例输出.....	25
图 6-1 修改Pin.....	26
图 6-2 连接Pin 19和Pin 21.....	27
图 6-3 示例输出.....	27
图 6-4 示例输出.....	28
图 6-5 将ADXL345模块连接到扩展头上.....	29
图 6-6 示例输出.....	30
图 7-1 文件内容示例.....	31
图 8-1 示例设置.....	33
图 8-2 示例设置.....	34
图 8-3 示例设置.....	34
图 8-4 将转换器连接到Header上.....	35
图 8-5 配置示例.....	36
图 8-7 示例输出.....	36
图 8-8 配置示例.....	37
图 8-10 示例命令和输出.....	37
图 8-11 示例输出.....	38
图 8-12 测试UART发送功能.....	38
图 8-13 测试UART接收功能.....	38
图 9-1 将Sense Hat (B) 连接到40-Pin GPIO Header上.....	40
图 9-2 将Sense Hat (B) 连接到40-Pin GPIO Header上.....	40
图 9-3 将2.4英寸LCD模块连接到40-Pin Header上.....	43
图 9-5 示例输出.....	44

1. 概述

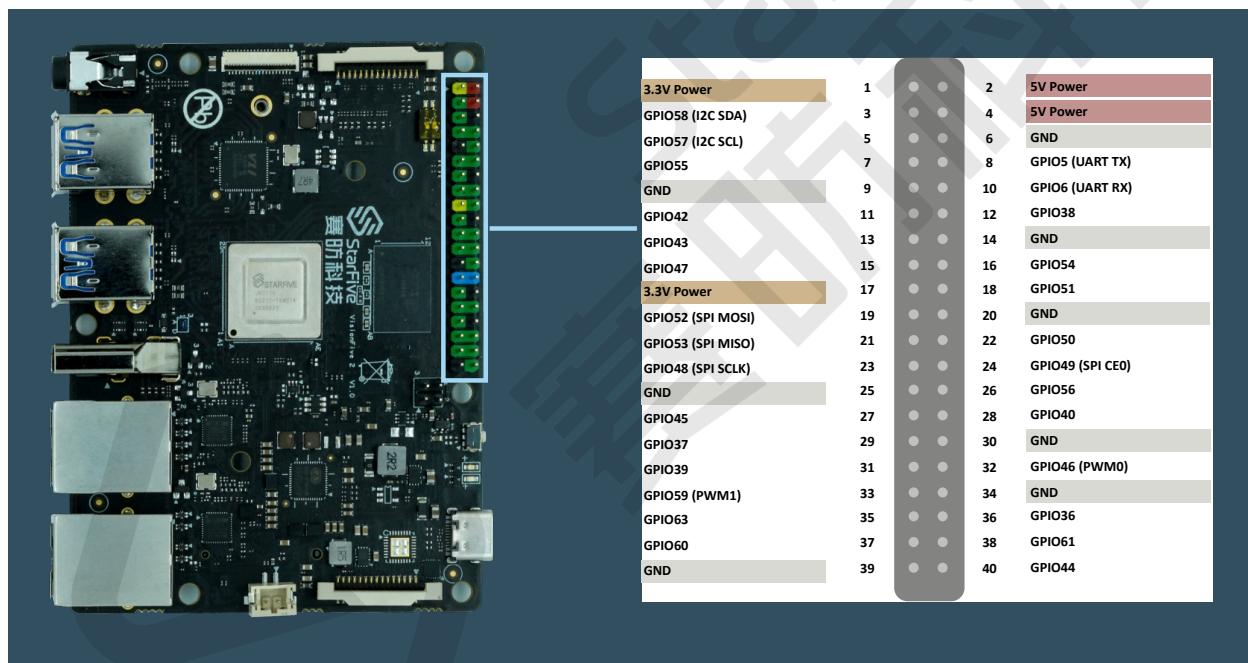
40-Pin Header使昉·星光 2单板计算机能够连接到各种外部组件，这是开发者能够创建其项目的必要条件。本文件旨在：

- 介绍本章所述的40-pin header。
- 提供配置和调试GPIO、I2C、SPI、PWM和UART的说明，如[GPIO操作\(第 19页\)](#)、[I2C操作\(第 21页\)](#)、[SPI操作\(第 26页\)](#)、[PWM操作\(第 31页\)](#)和[UART操作\(第 33页\)](#)章节所述。
- 为如何使用40-pin header提供外设示例，如“[外设示例\(第 39页\)](#)”一章所述。

1.1. 40-Pin Header定义

下图显示了40-Pin Header的位置。

图 1-1 40-Pin 定义



2. GPIO Pin分布

下表介绍了GPIO pin、映射图以及每个pin的功能说明。

表 2-1 GPIO Pin分布

Sys	dts	GPIO Num	Pin名	Pin Num	Pin Num	Pin名	GPIO Num	dts	Sys
		N/A	+3.3V	1	2	+5V	N/A		
i2c-0	i2c0	58	GPIO58 (I2C SDA)	3	4	+5V	N/A		
i2c-0	i2c0	57	GPIO57 (I2C SCL)	5	6	GND	N/A		
55		55	GPIO55	7	8	GPIO5 (UART TX)	5	uart0	ttyS0
		N/A	GND	9	10	GPIO6 (UART RX)	6	uart0	ttyS0
42		42	GPIO42	11	12	GPIO38	38		38
43		43	GPIO43	13	14	GND	N/A		
47		47	GPIO47	15	16	GPIO54	54		54
		N/A	+3.3V	17	18	GPIO51	51		51
spidev1.0spi0		52	GPIO52 (SPI MOSI)	19	20	GND	N/A		
spidev1.0spi0		53	GPIO53 (SPI MISO)	21	22	GPIO50	50		50
spidev1.0spi0		48	GPIO48 (SPI SCLK)	23	24	GPIO49 (SPI CE0)	49	spi0	spidev1.0
		N/A	GND	25	26	GPIO56	56		56
45		45	GPIO45	27	28	GPIO40	40		40
37		37	GPIO37	29	30	GND	N/A		

表 2-1 GPIO Pin分布 (续)

Sys	dts	GPIO Num	Pin名	Pin Num	Pin Num	Pin名	GPIO Num	dts	Sys
39		39	GPIO39	31	32	GPIO46	(PWM0)		pwm0
pwm1		59	GPIO59	(PWM1)	34	GND	N/A		
63		63	GPIO63	35	36	GPIO36	36		36
60		60	GPIO60	37	38	GPIO61	61		61
		N/A	GND	39	40	GPIO44	44		44



注:

- dts列为DTSL文件 (jh7110-visionfive-v2.dtsl) 中的节点名。如需查找相关节点，搜索节点名即可。
- Sys列为在/sys/class/gpio下导出GPIO pin时使用的pin号。

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*	文件	注释
通用	M	昉·星光 2 单板计算机	-
通用	M	<ul style="list-style-type: none">• 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	O	一台示波器	示波器用于测量GPIO电压。
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	-
PWM	O	一台示波器	示波器用于测量pin电压，并检查PWM周期和占空比。
SPI LCD	O	<ul style="list-style-type: none">• 2.4英寸LCD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。具有定位快速准确、漂

表 3-1 硬件准备 (续)

类型	M/ O*	文件	注释
			移小、功耗低、反欺骗和抗干扰能力强等特点。有关详细规格, 请参考 MAX-7Q GNSS HAT 。



注:

*: M: 必须。O: 可选

3.2. 准备软件

在配置40-pin Header之前, 需将Debian 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代码仓

GitHub代码仓地址汇总表如下:



注:

请确保您已切换到对应分支。

表 3-2 GitHub代码仓地址

类型	代码仓	分支
Linux	Linux	JH7110_VisionFive2-devel
Linux代码仓下的DTSI文件	<ul style="list-style-type: none"> • jh7110.dtsi • jh7110-visionfive-v2.dts • jh7110-visionfive-v2.dtsi 	-
Uboot	Uboot	JH7110_VisionFive2-devel
OpenSBI	OpenSBI	master
Debian	Debian	-

3.2.2. 将Debian OS烧录到Micro-SD卡上

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 2单板计算机快速参考手册》](#)中的“将OS烧录到Micro-SD卡上”章节，将Debian OS烧录到Micro-SD卡上。

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
```

 提示：

执行以下命令，安装crossbuild-essential-riscv64包：

```
sudo apt-get install crossbuild-essential-riscv64
```

 提示：

<Configuration_File>: 该命令

对starfive_jh7110_defconfig和starfive_visionfive2_defconfig均适用。

以下是命令示例：

```
make starfive_jh7110_defconfig ARCH=riscv CROSS_COMPILE=riscv64-linux-gnu-
make CROSS_COMPILE=riscv64-linux-gnu- ARCH=riscv dtbs
```

不同的开发板使用不同的dtb文件：

- jh7110-visionfive-v2.dtb：用于1.2A和1.3B版的开发板。
- jh7110-visionfive-v2-ac108.dtb：用于带有ac108编解码器的1.2A和1.3B版的开发板。
- jh7110-visionfive-v2-wm8960.dtb：用于带有wm8960编解码器的1.2A和1.3B版的开发板。

 提示：

您可查看开发板上的丝印获取版本信息。

3.2.4. 替换DTB文件

本节提供替换DTB文件的两种方法：

- [方法1：直接替换DTB文件 \(第 15页\)](#)
- [方法2：添加启动项 \(第 16页\)](#)

3.2.4.1. 方法1：直接替换DTB文件

前提条件

确保您已经执行了[生成DTB文件 \(第 14页\)](#)中的步骤。

在Linux的根目录下执行以下命令，以替换DTB文件：

1. 将micro-SD卡插入到带有Ubuntu系统的PC上，并执行以下命令检查SD卡分区：

```
sudo fdisk -l
```

示例输出：

```
Device Start End Sectors Size Type
/dev/sdc1 4096 8191 4096 2M unknown
/dev/sdc2 8192 16383 8192 4M unknown
/dev/sdc3 16384 221183 204800 100M EFI System
/dev/sdc4 221184 4503518 4282335 2G Linux filesystem
```

在此输出中，/dev/sdc3分区是SD卡分区。

2. 执行以下命令，在mnt文件路径下挂载SD卡分区：

```
sudo mount /dev/sdc3 /mnt
```

3. 在Linux根目录下执行以下命令：

```
sudo cp arch/riscv/boot/dts/starfive/<DTB_File> /mnt/dtbs/starfive
```

i 提示：

不同的开发板使用不同的dtb文件：

- jh7110-visionfive-v2.dtb：用于1.2A和1.3B版的开发板。
- jh7110-visionfive-v2-ac108.dtb：用于带有ac108编解码器的1.2A和1.3B版的开发版。
- jh7110-visionfive-v2-wm8960.dtb：用于带有wm8960编解码器的1.2A和1.3B版的开发板。

i 提示：

您可查看开发板上的丝印获取版本信息。

示例命令：

```
sudo cp
arch/riscv/boot/dts/starfive/jh7110-visionfive-v2.dtb /
mnt/dtbs/starfive
```

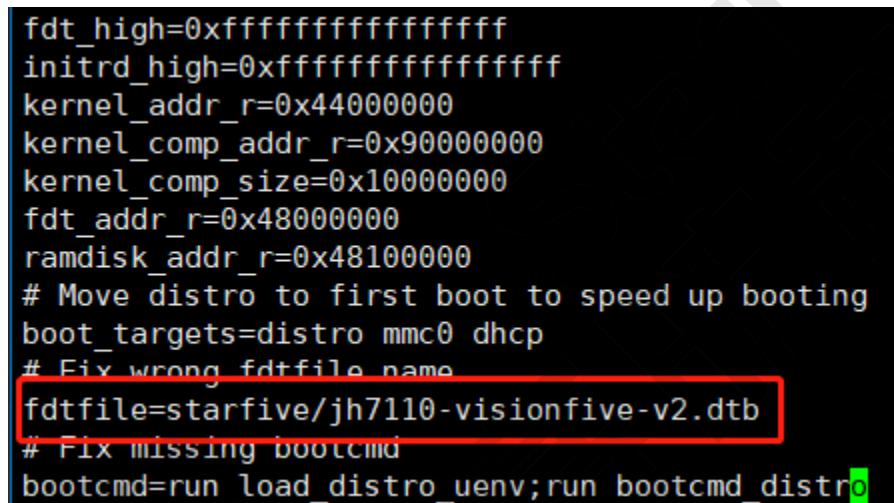
4. 执行以下命令，进入/mnt目录，修改配置文件（uEnv.txt）：

```
cd /mnt
sudo vim uEnv.txt
fdtfile=starfive/<DTB_File>
```

示例命令：

```
cd /mnt
sudo vim uEnv.txt
fdtfile=starfive/jh7110-visionfive-v2.dtb
```

图 3-1 示例



```
fdt_high=0xfffffffffffffff
initrd_high=0xfffffffffffffff
kernel_addr_r=0x44000000
kernel_comp_addr_r=0x90000000
kernel_comp_size=0x10000000
fdt_addr_r=0x48000000
ramdisk_addr_r=0x48100000
# Move distro to first boot to speed up booting
boot_targets=distro mmc0 dhcp
# Fix wrong fdtfile name
fdtfile=starfive/jh7110-visionfive-v2.dtb
# Fix missing bootcmd
bootcmd=run load_distro_uenv;run bootcmd_distro
```

5. 取消/mnt目录的挂载：

```
sudo umount /mnt
```

6. 从PC上取出micro-SD卡，插入昉·星光2，然后上电。



注：

您还可以通过以下步骤替换DTB文件：

- 使用USB驱动器或通过SCP直接将DTB文件复制到Debian系统下的/boot/dtbs/starfive目录。
- 修改配置文件uEnv.txt。请参见[步骤4\(第 16页\)](#)。

3.2.4.2. 方法2：添加启动项

前提条件

确保您已经执行了[生成DTB文件\(第14页\)](#)中的步骤。

在Linux的根目录下执行以下命令，以替换DTB文件：

- 将micro-SD卡插入到带有Ubuntu系统的PC上，并执行以下命令检查SD卡分区：

```
sudo fdisk -l
```

示例输出：

```
Device Start End Sectors Size Type
/dev/sdc1 4096 8191 4096 2M unknown
/dev/sdc2 8192 16383 8192 4M unknown
/dev/sdc3 16384 221183 204800 100M EFI System
/dev/sdc4 221184 4503518 4282335 2G Linux filesystem
```

在此输出中，/dev/sdc3分区是SD卡分区。

- 执行以下命令，在mnt文件路径下挂载SD卡分区：

```
sudo mount /dev/sdc3 /mnt
```

- 在Linux根目录下执行以下命令：

```
sudo cp arch/riscv/boot/dts/starfive/<DTB_File> /mnt/dtbs/starfive
sudo umount /mnt
```

不同的开发板使用不同的dtb文件：

- jh7110-visionfive-v2.dtb：用于1.2A和1.3B版的开发板。
- jh7110-visionfive-v2-ac108.dtb：用于带有ac108编解码器的1.2A和1.3B版的开发板。
- jh7110-visionfive-v2-wm8960.dtb：用于带有wm8960编解码器的1.2A和1.3B版的开发板。



提示：

您可查看开发板上的丝印获取版本信息。

示例命令：

```
sudo cp
arch/riscv/boot/dts/starfive/jh7110-visionfive-v2.dtb /
mnt/dtbs/starfive
```

- 执行以下命令，打开extlinux.conf：

```
sudo vim /mnt/extlinux/extlinux.conf
```

- 添加了以下各行，然后保存并退出：

```
label l1
menu label Debian GNU/Linux bookworm/sid 5.15.0-starfive (customized)
linux /vmlinuz-5.15.0-starfive
initrd /initrd.img-5.15.0-starfive
fdt /dtbs/starfive/<DTB_File>
append root=/dev/mmcblk1p4 rw console=tty0 console=ttyS0,115200
earlycon rootwait
stmmaceth=chain_mode:1 selinux=0
label l1r
menu label Debian GNU/Linux bookworm/sid 5.15.0-starfive
(customized)(rescue target)
linux /vmlinuz-5.15.0-starfive
initrd /initrd.img-5.15.0-starfive
fdt /dtbs/starfive/<DTB_File>
append root=/dev/mmcblk1p4 rw console=tty0 console=ttyS0,115200
earlycon rootwait
stmmaceth=chain_mode:1 selinux=0 single
```

6. 从PC上取出micro-SD卡，插入昉·星光2，然后上电。
7. 您将在U-Boot启动菜单中看到自定义菜单项，例如**Debian GNU/Linux bookworm/sid 5.15.0-starfive (customized)**，选择此项。

4. GPIO操作

本节提供配置GPIO的命令：

4.1. 配置GPIO

1. 执行以下操作配置GPIO：

执行以下命令以配置GPIO44：

```
cd /sys/class/gpio  
echo 44 > export
```

2. 进入GPIO44目录：

```
cd gpio44
```



注：

命令中的44为Sys编号，对应GPIO44。如需获取GPIO Pin Name的更多详细说明，请参阅本文档中的“[GPIO Pin分布\(第 10页\)](#)”部分。

3. 将GPIO44的方向设为输入：

```
echo in > direction
```

4. 或者，将GPIO44的方向设为输出：

```
echo out > direction
```

5. 将GPIO44的电压设为高电平：

```
echo 1 > value
```



提示：

可以用示波器检测电压。

6. 将GPIO44的电压设为低电平：

```
echo 0 > value
```



提示：

可以用示波器检测电压。

7. 将3.3V电源的pin连接到GPIO44上，并检测GPIO44的电压：

```
cat value
```

8. 将GND pin连接到GPIO44上，并检测GPIO44的电压：

```
cat value
```



StarFive 赛昉科技

5. I2C操作

本章介绍如何配置和调试I2C GPIO。

5.1. 配置I2C GPIO

执行以下操作，以配置I2C：

- [连接硬件 \(第 21页\)](#)
- [配置dts文件 \(第 22页\)](#)

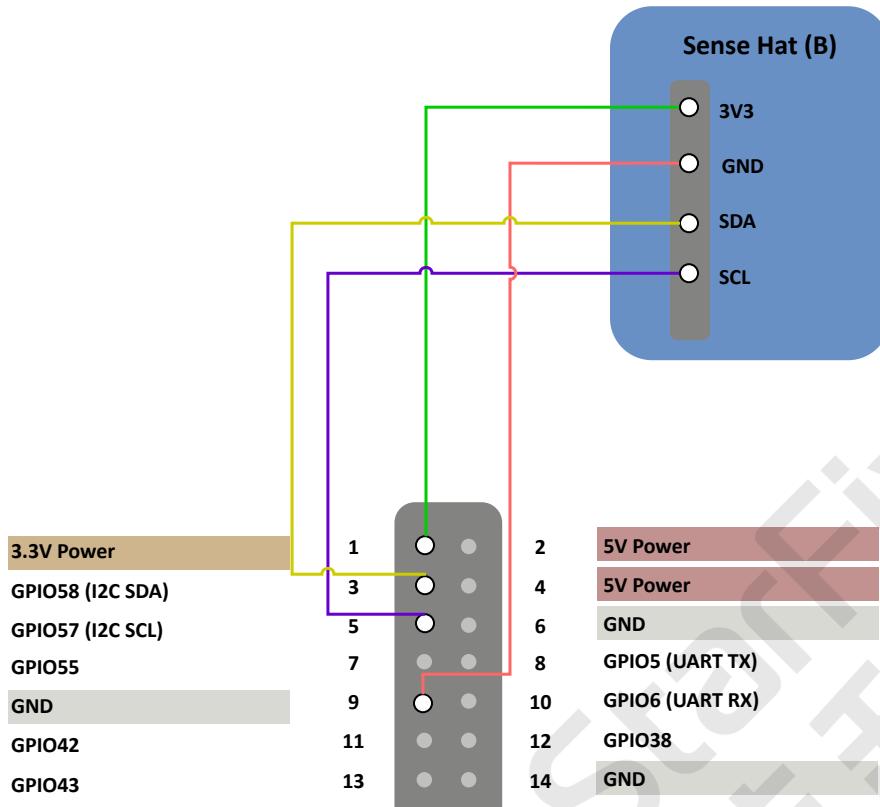
5.1.1. 连接硬件

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

表 5-1 将Sense Hat (B) 连接到40-Pin GPIO Header上

Sense HAT (B)	40-Pin GPIO Header	
	Pin Number	Pin Name
3V3	1	3.3V 电压
GND	9	GND
SDA	3	GPIO58 (I2C SDA)
SCL	5	GPIO57 (I2C SCL)

图 5-1 将Sense Hat (B) 连接到40-Pin GPIO Header上



5.1.2. 配置dts文件

系统支持7路I2C总线：i2c0到i2c6。

DTSI文件jh7110-visionfive-v2.dtsi位于/linux/arch/riscv/boot/dts/starfive路径下。

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

```

81     i2c0_pins: i2c0-pins {
82         i2c0-pins-scl {
83             sf,pins = <PAD_GPIO57>;
84             sf,pinmux = <PAD_GPIO57_FUNC_SEL_0>;
85             sf,pin-ioconfig = <IO(GPIO_IE(1) | (GPIO_PU(1)))>;
86             sf,pin-gpio-dout = <GPO_LOW>;
87             sf,pin-gpio-doen = <OEN_I2C0_IC_CLK_OE>;
88             sf,pin-gpio-din = <GPI_I2C0_IC_CLK_IN_A>;
89         };
90
91         i2c0-pins-sda {
92             sf,pins = <PAD_GPIO58>;
93             sf,pinmux = <PAD_GPIO58_FUNC_SEL_0>;
94             sf,pin-ioconfig = <IO(GPIO_IE(1) | (GPIO_PU(1)))>;
95             sf,pin-gpio-dout = <GPO_LOW>;
96             sf,pin-gpio-doen = <OEN_I2C0_IC_DATA_OE>;
97             sf,pin-gpio-din = <GPI_I2C0_IC_DATA_IN_A>;
98         };
99     };
100

```



注:

I2C GPIO pin number是pin name中指示的编号。如需获取GPIO pin name的更多详细说明, 请参阅本文档中的“[GPIO Pin分布 \(第 10页\)](#)”部分。I2C GPIO的pin name如下:

- GPIO58 (I2C SDA)
- GPIO57 (I2C SCL)

5.2. 调试I2C GPIO

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 示例输出

```

root@starfive:~# i2cdetect -l
i2c-0  i2c          Synopsys DesignWare I2C adapter      I2C adapter
i2c-2  i2c          Synopsys DesignWare I2C adapter      I2C adapter
i2c-5  i2c          Synopsys DesignWare I2C adapter      I2C adapter
i2c-6  i2c          Synopsys DesignWare I2C adapter      I2C adapter
i2c-7  i2c          Inno HDMI                         I2C adapter

```

2. 执行以下命令, 以检测设备:

```
i2cdetect -y -r 0
```

i 提示：

0为I2C总线序号。

结果：

图 5-4 示例输出

```
root@starfive:~# i2cdetect -y -r 0
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          - - - - - - - - - - - - - - - - - - - - - -
10:          - - - - - - - - - - - - - - - - - - - - - -
20:          - - - - - - - - - - - - - - - - - - - - - -
30:          - - - - - - - - - - - - - - - - - - - - - -
40:          - - - - - - - - - - - - - - - - - - - - - -
50:          - - - - - - - - - - - - - - - - - - - - - -
60:          - - - - - - - - - - - - - - - - - - - - - -
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 0 0x5c 0x0f
```

i 提示：

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

结果：

图 5-5 示例输出

```
root@starfive:~# i2cget -f -y 0 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 0 0x5c 0x11 0x10
```

i 提示：

- 0：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 0 0x5c
```



提示：

- 0：I2C总线序号。
- 0x5c：I2C设备地址。

结果：

图 5-6 示例输出

```
root@starfive:~# i2cdump -y 0 0x5c
No size specified (using byte-data access)
  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 00 b1 .....??
10: 00 10 00 00 00 00 00 00 00 00 00 00 00 01 b1 68 6e .?.....??hn
20: 00 00 00 00 00 00 00 02 71 8f 2f 00 00 00 00 00 .....?q?/.....
30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....?
40: 72 e4 03 20 8a 0f 05 49 00 06 27 7b 8b 11 0b 44 r?? ???I.?'{???D
50: 42 fd 7b 0e 00 71 8f 2f 06 03 15 0a b9 04 80 c0 B?{?.q?/?????????
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 b1 .....??
90: 00 10 00 00 00 00 00 00 00 00 00 00 00 01 b1 68 6e .?.....??hn
a0: 00 00 00 00 00 00 00 71 8f 2f 00 00 00 00 00 00 .....?q?/.....
b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....?
c0: 72 e4 03 20 8a 0f 05 49 00 06 27 7b 8b 11 0b 44 r?? ???I.?'{???D
d0: 42 fd 7b 0e 00 71 8f 2f 06 03 15 0a b9 04 80 c0 B?{?.q?/?????????
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

DTSI文件jh7110-visionfive-v2.dtsi位于/linux/arch/riscv/boot/dts/starfive路径下。

系统支持7路SPI总线：spi0到spi6。

6.1.1. 修改Pin

配置的SPI GPIO Name即“pin name”中的编号。如需获取GPIO pin name的更多详细说明，请参阅本文档中的[“GPIO Pin分布\(第 10页\)”](#)部分。未被占用的pin均可配置。以下是jh7110-visionfive-v2.dtsi文件中的默认设置：

图 6-1 修改Pin

```
187      ssp0_pins: ssp0-pins {
188          ssp0-pins_tx {
189              sf,pins = <PAD_GPIO52>;
190              sf,pinmux = <PAD_GPIO52_FUNC_SEL_0>;
191              sf,pin-ioconfig = <IO(GPIO_IE(1))>;
192              sf,pin-gpio-dout = <GPO_SPI0_SSPTXD>;
193              sf,pin-gpio-doen = <OEN_LOW>;
194          };
195
196          ssp0-pins_rx {
197              sf,pins = <PAD_GPIO53>;
198              sf,pinmux = <PAD_GPIO53_FUNC_SEL_0>;
199              sf,pin-ioconfig = <IO(GPIO_IE(1))>;
200              sf,pin-gpio-doen = <OEN_HIGH>;
201              sf,pin-gpio-din = <GPI_SPI0_SSPrXD>;
202          };
203
204          ssp0-pins_clk {
205              sf,pins = <PAD_GPIO48>;
206              sf,pinmux = <PAD_GPIO48_FUNC_SEL_0>;
207              sf,pin-ioconfig = <IO(GPIO_IE(1))>;
208              sf,pin-gpio-dout = <GPO_SPI0_SSPCLKOUT>;
209              sf,pin-gpio-doen = <OEN_LOW>;
210          };
211
212          ssp0-pins_cs {
213              sf,pins = <PAD_GPIO49>;
214              sf,pinmux = <PAD_GPIO49_FUNC_SEL_0>;
215              sf,pin-ioconfig = <IO(GPIO_IE(1))>;
216              sf,pin-gpio-dout = <GPO_SPI0_SSPFSSOUT>;
217              sf,pin-gpio-doen = <OEN_LOW>;
218          };
219      };
```

6.2. 调试SPI GPIO

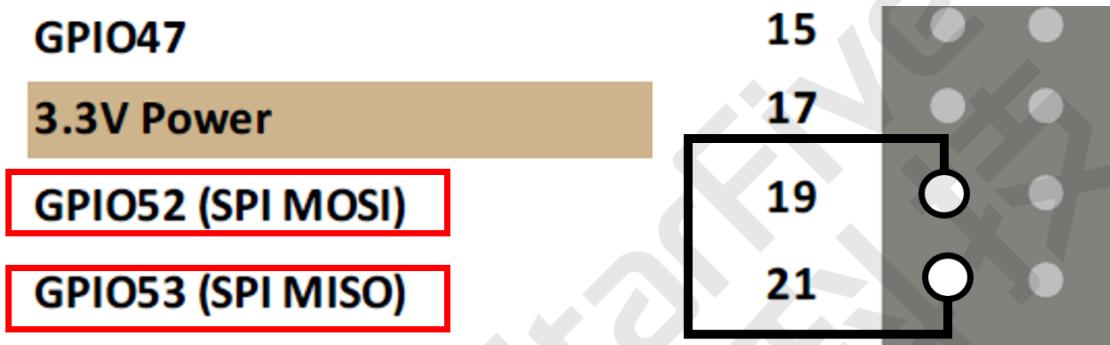
本节提供了环回测试的步骤，以及使用ADXL345模块测试SPI的步骤。

6.2.1. 环回测试

按照以下步骤，执行环回测试：

- 接线：按照下图连接Pin 19和Pin 21。

图 6-2 连接Pin 19和Pin 21



- 进入测试工具spidev_test.c所在的目录：

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

- 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。

- 将spidev_test文件上传到昉·星光 2上，并执行以下命令，更改执行权限：

```
chmod +x spidev_test
```

- 确认SPI设备。

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

结果：

图 6-3 示例输出

```
root@starfive:~# ls /dev/spi*
/dev/spidev1.0
root@starfive:~#
```

输出中的spidev1.0为设备名。

- 执行以下命令，以运行测试：

```
./spidev_test -D /dev/spidev1.0 -v -p string_to_send
```



提示:

`spidev1.0`即上一步中获取的设备名。

结果：

图 6-4 示例输出

```
root@starfive:~/spi# ./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_send|
RX | 73 74 72 69 67 5F 74 6F 5F 73 65 6E 64 ----- |string_to_send|
root@starfive:~/spi#
```

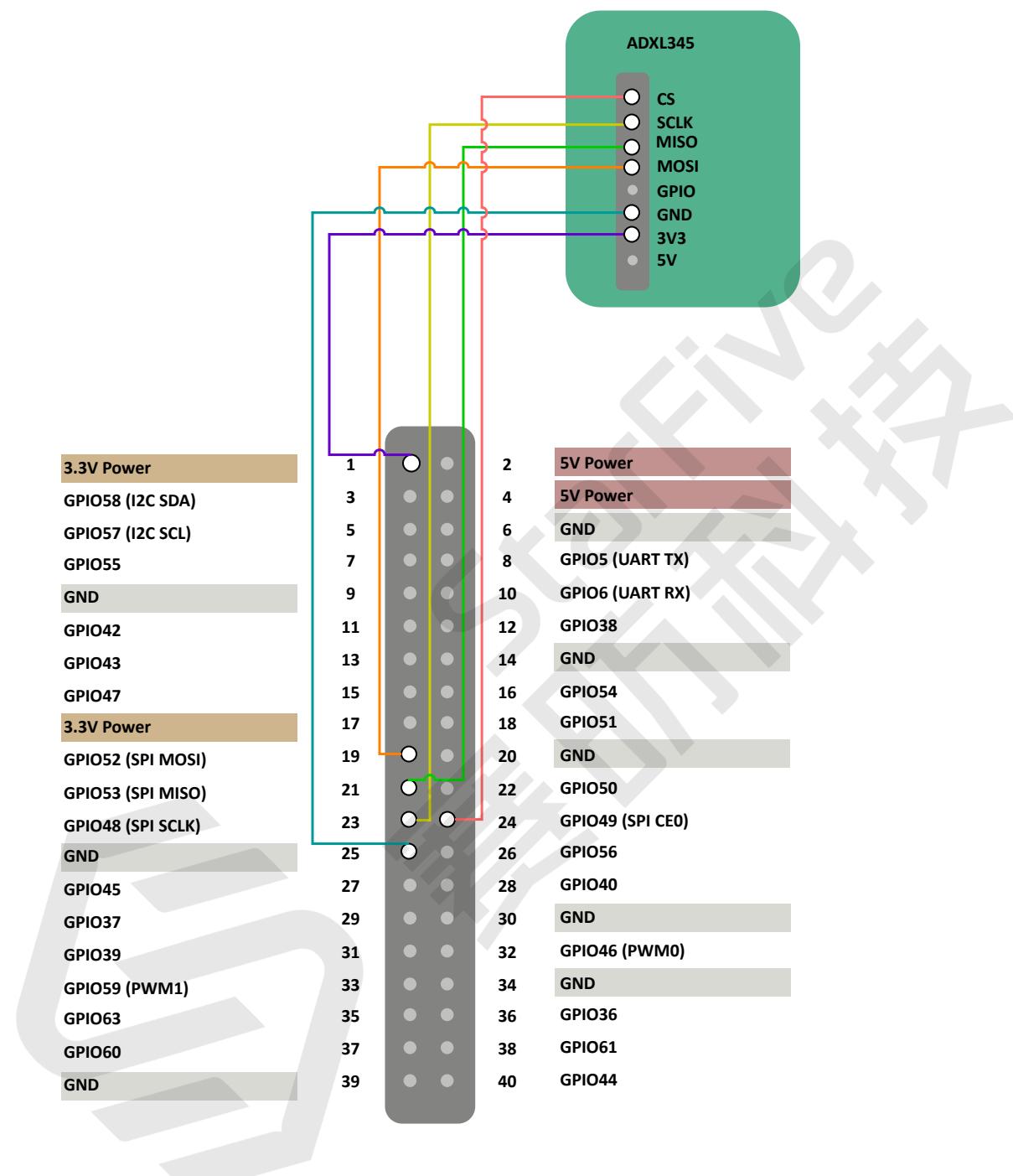
在输出中，**TX**行和**RX**行的测试结果显示相同，接收的数据和发送的完全一致，表示测试成功。

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：

- 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



- 进入测试工具spidev_test.c所在的目录：

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

- 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文件上传到昉·星光 2上，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*
```

图 6-6 示例输出

```
root@starfive:~# ls /dev/spi*
/dev/spidev1.0
root@starfive:~#
```

输出中的spidev1.0为设备名。

6. 执行以下命令，读取设备ID：

```
./spidev_test -H -O -D /dev/spidev1.0 -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/spidev1.0 -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/spidev1.0 -v -p \\x9e\\x00
```

9. 执行以下命令，以写入寄存器值：

```
./spidev_test -H -O -D /dev/spidev1.0 -v -p \\x1e\\xaa
```

10. 执行以下命令，以读取验证说明：

```
./spidev_test -H -O -D /dev/spidev1.0 -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

DTSI文件jh7110-visionfive-v2.dtsi位于/linux/arch/riscv/boot/dts/starfive路径下。

系统最多支持8路PWM。

7.1.1. 修改Pin

下图为修改pin的文件内容示例：

图 7-1 文件内容示例

```
169      pwm_pins: pwm-pins {
170          pwm_ch0-pins {
171              sf,pins = <PAD_GPIO46>;
172              sf,pinmux = <PAD_GPIO46_FUNC_SEL_0>;
173              sf,pin-ioconfig = <IO(GPIO_IE(1))>;
174              sf,pin-gpio-dout = <GPO_PTC0_PWM_0>;
175              sf,pin-gpio-doen = <OEN_PTC0_PWM_0_OE_N>;
176          };
177
178          pwm_ch1-pins {
179              sf,pins = <PAD_GPIO59>;
180              sf,pinmux = <PAD_GPIO59_FUNC_SEL_0>;
181              sf,pin-ioconfig = <IO(GPIO_IE(1))>;
182              sf,pin-gpio-dout = <GPO_PTC0_PWM_1>;
183              sf,pin-gpio-doen = <OEN_PTC0_PWM_1_OE_N>;
184          };
185      };
186  
```

配置的PWM GPIO Number即“Pin Name”中的编号。如需获取GPIO pin name的更多详细说明，请参阅本文档中的[“GPIO Pin分布 \(第 10页\)”](#)部分。

7.1.2. PWM和pin Name映射

下表介绍了PWM和Pin Name的映射关系：

表 7-1 PWM和Pin Name映射

PWM	GPIO (Pin Name)
PWM0	GPIO46
PWM1	GPIO59

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PWM占空比：

```
echo 1000000 > duty_cycle
```

4. 使用示波器测量对应pin，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URAT GPIO:

8.1. 配置UART GPIO

DTSI文件jh7110-visionfive-v2.dtsi位于/linux/arch/riscv/boot/dts/starfive路径下。

系统最多支持6路URAT。

配置的URAT GPIO Number即“pin name”中的编号。如需获取GPIO pin name的更多详细说明，请参阅本文档中的[“GPIO Pin分布 \(第 10页\)”](#)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 示例设置

```
aliases {  
    spi0 = &qspi;  
    gpio0 = &gpio;  
    ethernet0 = &gmac0;  
    ethernet1 = &gmac1;  
    mmc0 = &sdio0;  
    mmc1 = &sdio1;  
    serial0 = &uart0;  
    serial1 = &uart1;  
    serial2 = &uart2;  
    serial3 = &uart3;
```

2. 在dts文件中添加“uart1”或“uart2”节点。以下是一个示例：

图 8-2 示例设置

```

&uart0 {
    pinctrl-names = "default";
    pinctrl-0 = <&uart0_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。



注：

配置的URAT GPIO Number即“Pin Name”中的编号。未被占用的pin均可配置。
如需获取GPIO pin name的更多详细说明，请参阅本文档中的[“GPIO Pin分布\(第10页\)”](#)部分。

图 8-3 示例设置

```

&gpio {
    uart0_pins: uart0-pins {
        uart0-pins-tx {
            starfive,pins = <PAD_GPIO5>;
            starfive,pin-ioconfig = <IO(GPIO_IE(1) | GPIO_DS(3))>;
            starfive,pin-gpio-dout = <GPO_UART0_SOUT>;
            starfive,pin-gpio-doen = <OEN_LOW>;
        };

        uart0-pins-rx {
            starfive,pins = <PAD_GPIO6>;
            starfive,pinmux = <PAD_GPIO6_FUNC_SEL_0>;
            starfive,pin-ioconfig = <IO(GPIO_IE(1) | GPIO_PU(1))>;
            starfive,pin-gpio-doen = <OEN_HIGH>;
            starfive,pin-gpio-din = <GPI_UART0_SIN>;
        };
    };

    uart1_pins: uart1-pins {
        uart1-pins-tx {
            starfive,pins = <PAD_GPIO63>;
            starfive,pinmux = <PAD_GPIO63_FUNC_SEL_0>;
            starfive,pin-ioconfig = <IO(GPIO_IE(1) | GPIO_DS(3))>;
            starfive,pin-gpio-dout = <GPO_UART1_SOUT>;
            starfive,pin-gpio-doen = <OEN_LOW>;
        };
    };
}

```

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/ttyS1
UART2	/dev/ttyS2

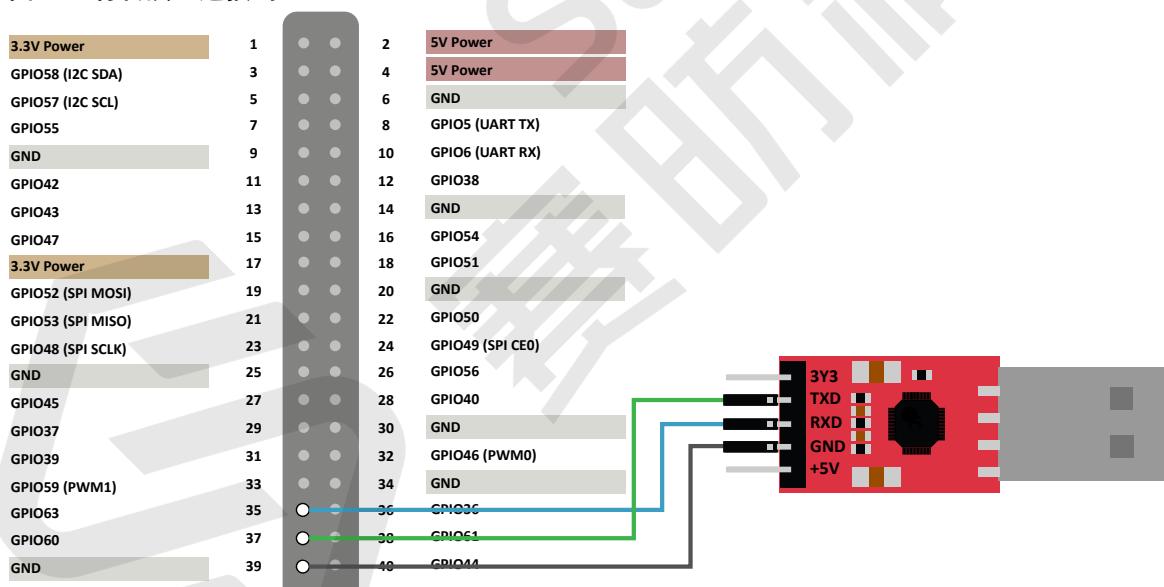
8.2. 调试UART GPIO

8.2.1. 连接硬件

按照如下步骤设置硬件：

- 将USB转串口转换器的跳线连接到昉·星光2的40-Pin GPIO Header上，如下图：

图 8-4 将转换器连接到Header上



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

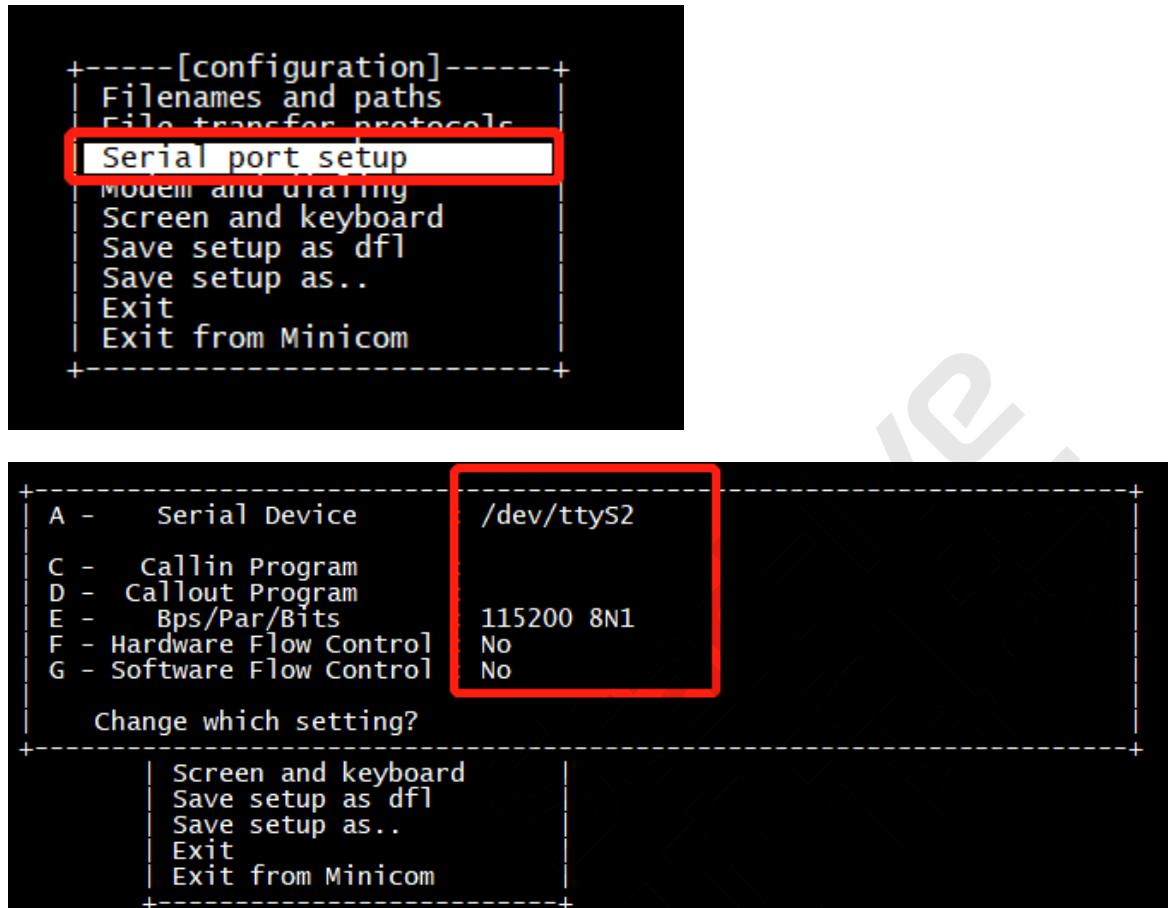
8.2.2. 调试UART发送和接收功能

- 配置昉·星光2的minicom：

```
sudo minicom -s
```

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

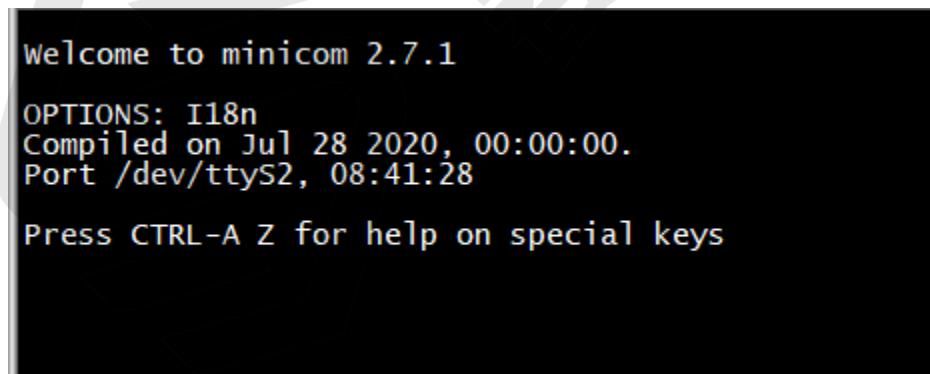
图 8-5 配置示例



3. 在PC上，输入以下命令，启动昉·星光2的minicom：

```
minicom -o -D /dev/ttys1
```

图 8-7 示例输出

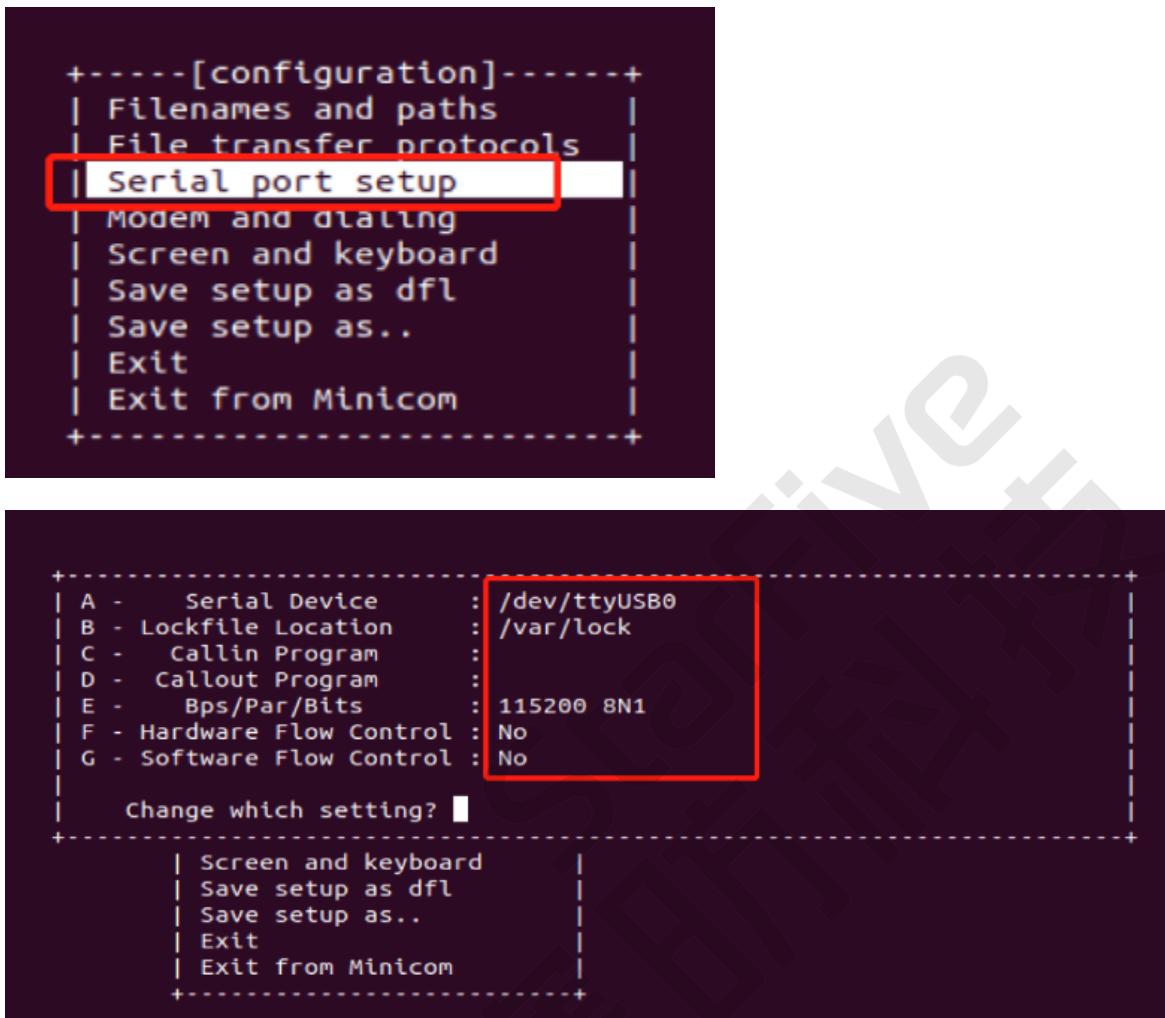


4. 输入以下命令，配置Ubuntu minicom：

```
sudo minicom -s
```

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

图 8-8 配置示例



注：

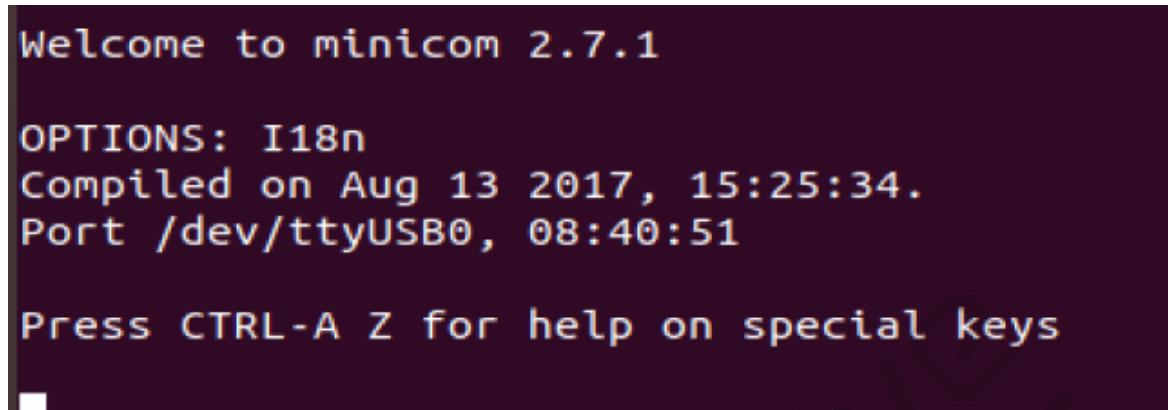
在Ubuntu上，可以使用`dmesg | grep tty`命令检测串口设备。

图 8-10 示例命令和输出

```
xiangyao@xiangyao-VirtualBox:~$ dmesg | grep tty
[    0.158110] printk: console [tty0] enabled
[    5.322731] ttys2: LSR safety check engaged!
[    5.323281] ttys2: LSR safety check engaged!
[91855.795788] usb 1-2: pl2303 converter now attached to ttys2
[92154.097583] ttys2: LSR safety check engaged!
```

6. 启动Ubuntu minicom，您可以看到如下输出：

图 8-11 示例输出



```
Welcome to minicom 2.7.1

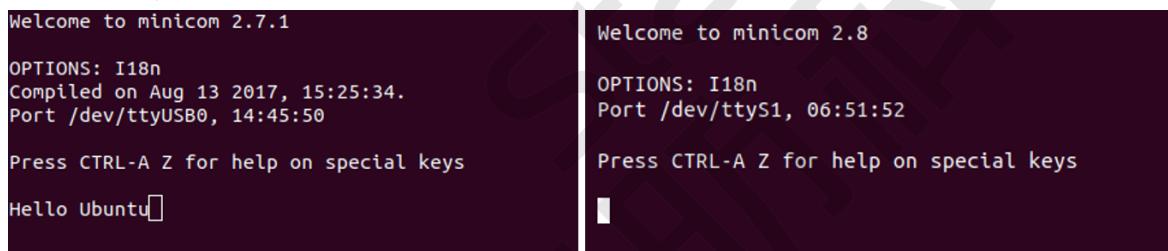
OPTIONS: I18n
Compiled on Aug 13 2017, 15:25:34.
Port /dev/ttyUSB0, 08:40:51

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

测试UART发送功能：

7. 要测试UART发送功能，可以在昉·星光 2的minicom上输入字符，例如Hello Ubuntu。
然后，您将看到该字符显示在Ubuntu minicom上，如下所示：

图 8-12 测试UART发送功能



```
Welcome to minicom 2.7.1
OPTIONS: I18n
Compiled on Aug 13 2017, 15:25:34.
Port /dev/ttyUSB0, 14:45:50

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

```
Welcome to minicom 2.8
OPTIONS: I18n
Port /dev/ttys1, 06:51:52

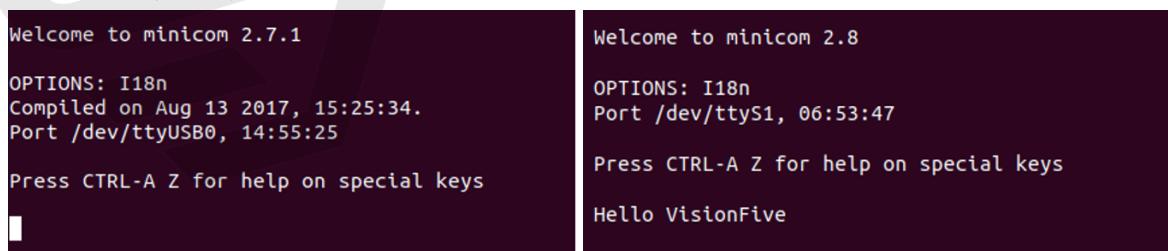
Press CTRL-A Z for help on special keys
```

- 左图：Ubuntu minicom界面
- 右图：昉·星光 2 minicom界面

测试UART接收功能：

8. 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，您将看到在昉·星光 2的minicom上输出的字符：

图 8-13 测试UART接收功能



```
Welcome to minicom 2.7.1
OPTIONS: I18n
Compiled on Aug 13 2017, 15:25:34.
Port /dev/ttyUSB0, 14:55:25

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

```
Welcome to minicom 2.8
OPTIONS: I18n
Port /dev/ttys1, 06:53:47

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

- 左图：Ubuntu minicom界面
- 右图：昉·星光 2 minicom界面

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. 连接硬件

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

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

Sense HAT (B)	Pin Number
3V3	1
GND	9
SDA	3
SCL	5

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

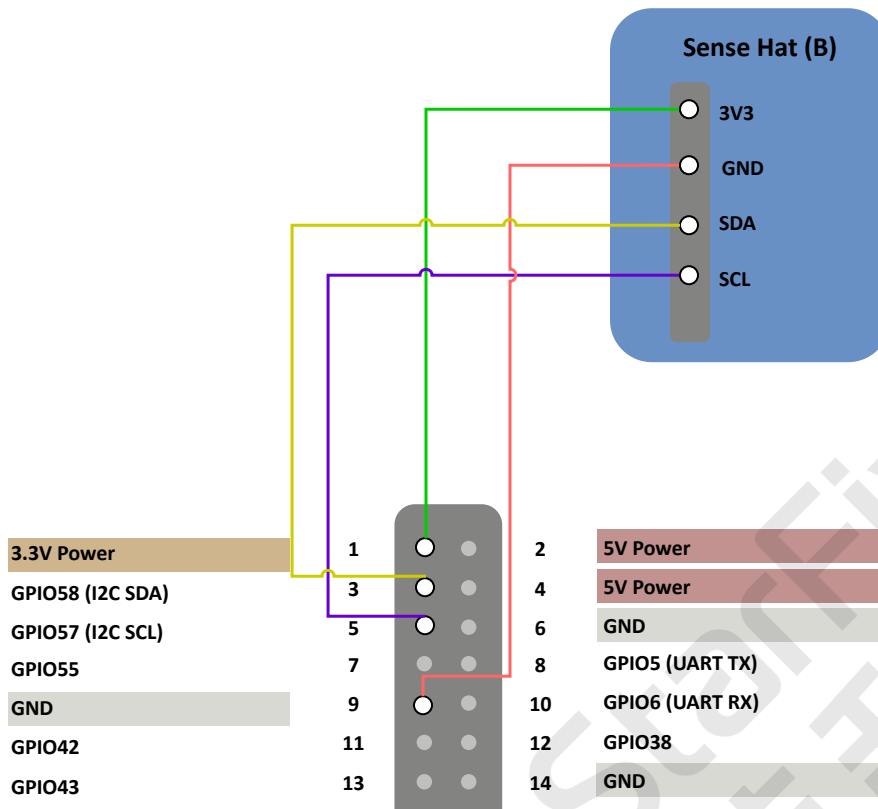
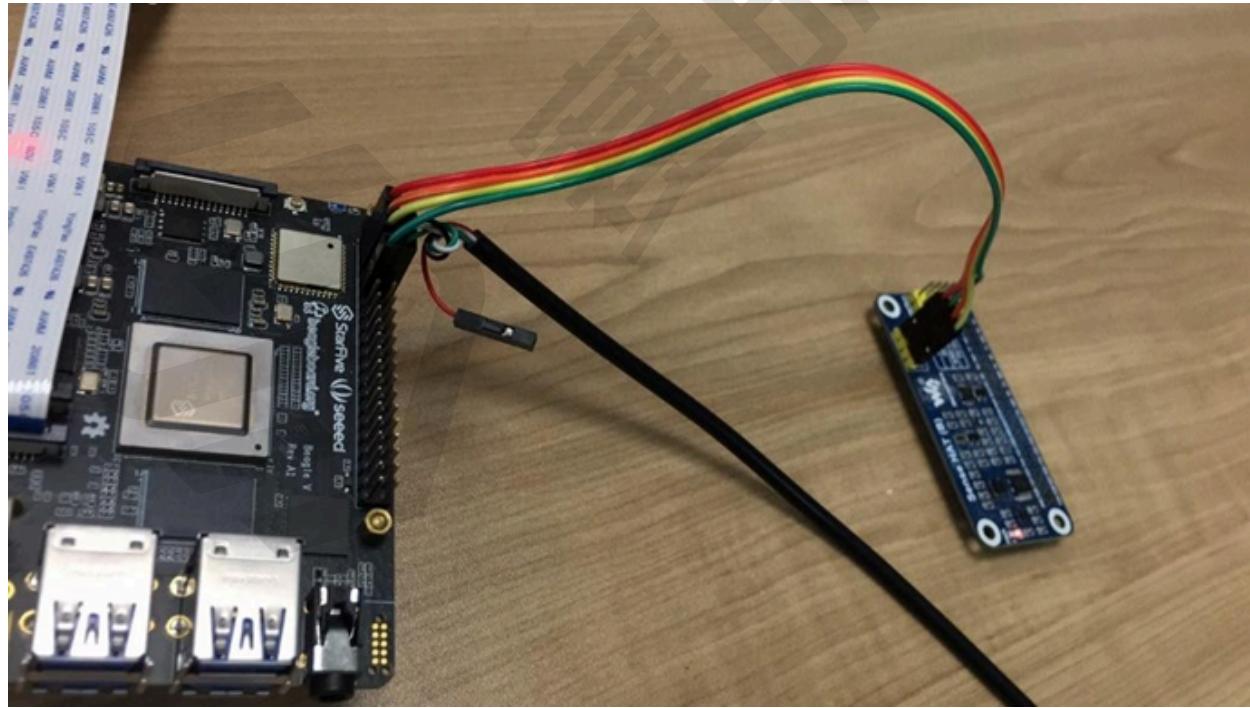


图 9-2 将Sense Hat (B) 连接到40-Pin GPIO Header上



9.1.2. Sense Hat (B)运行示例

以SHTC3传感器为例：

1. 从以下网址下载源代码: [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:

如输出以下结果，则表示执行成功:

```
root@starfive:~# ./shtc3

SHTC3 Sensor Test Program ...
Fopen : /dev/i2c-0
Temperature = 52.20°C , Humidity = 55.32%
Temperature = 23.81°C , Humidity = 55.29%
Temperature = 23.79°C , Humidity = 55.30%
Temperature = 23.82°C , Humidity = 55.29%
Temperature = 23.81°C , Humidity = 55.29%
Temperature = 23.82°C , Humidity = 55.30%
```

9.2. 2.4英寸LCD显示模块示例

本例中使用了2.4英寸LCD显示模块。更多详细规格，请参考：https://www.waveshare.com/wiki/2.4inch_LCD_Module。



注：

官方示例需要根据实际情况进行修改。

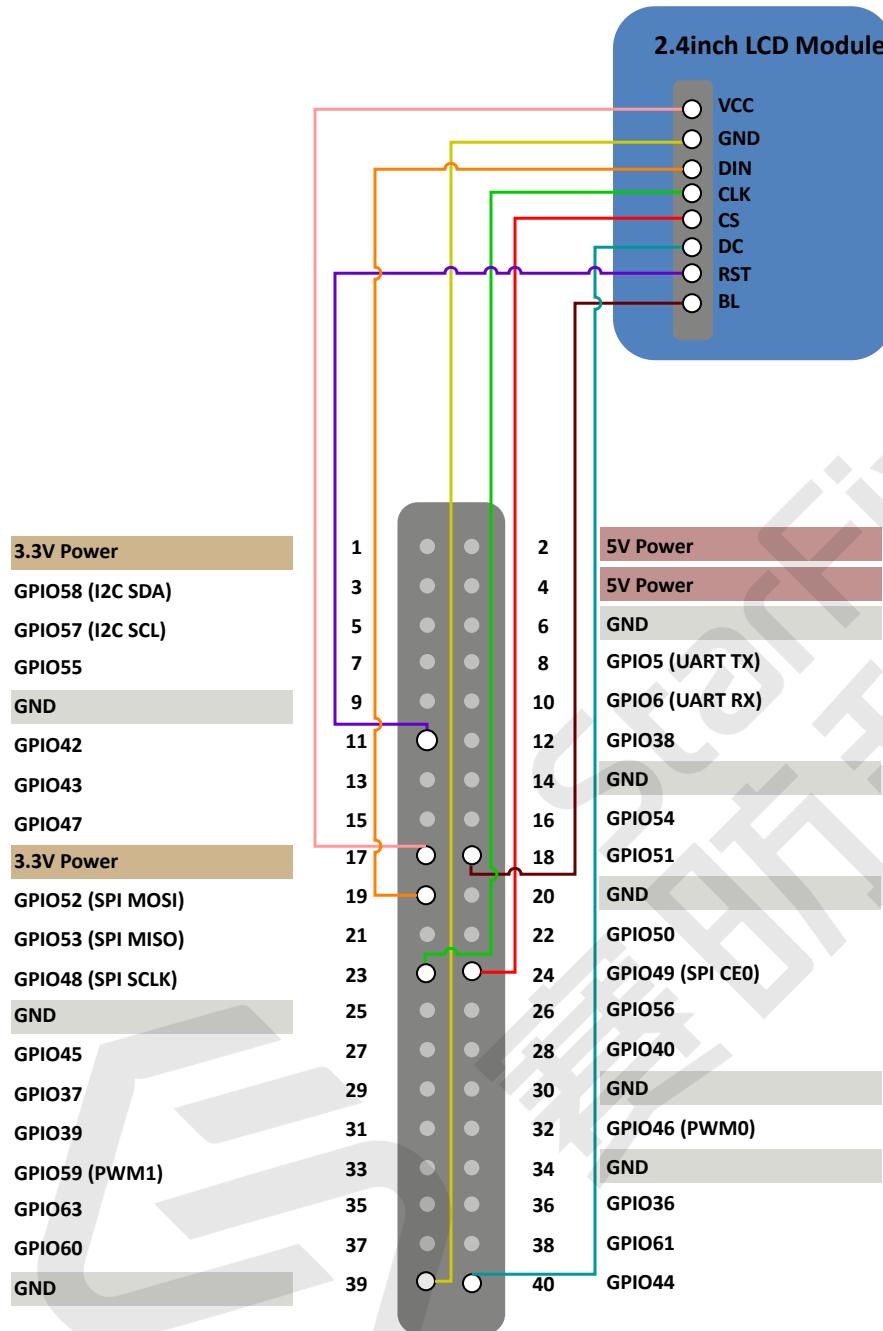
9.2.1. 连接硬件

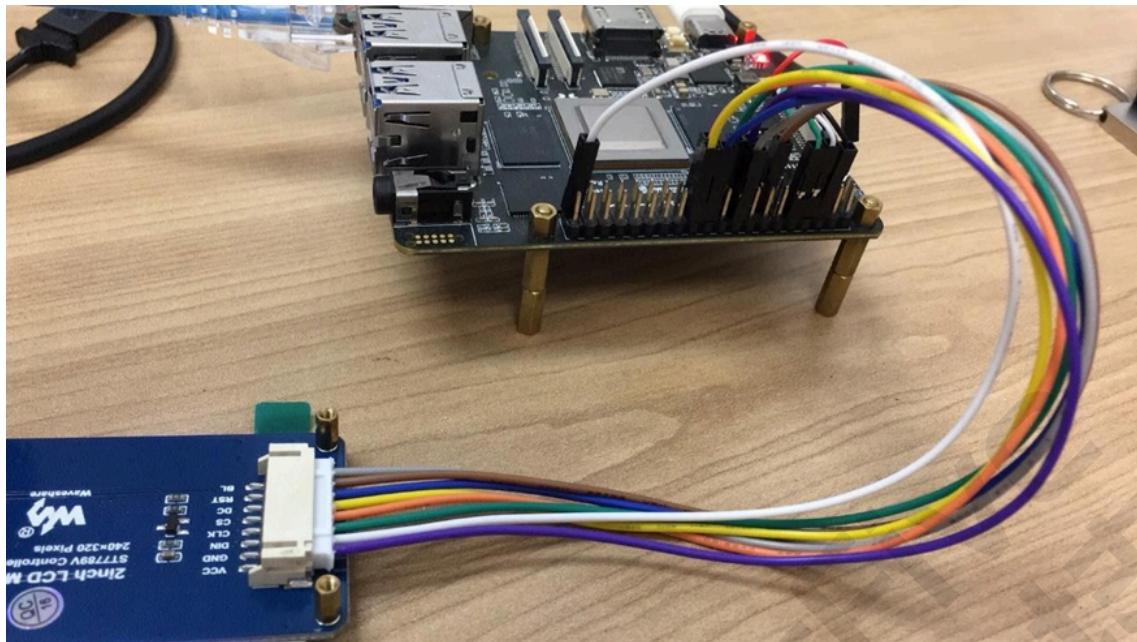
以下表格和图片描述了如何将2.4英寸LCD模块连接到40-Pin Header上：

表 9-2 将2.4英寸LCD模块连接到40-Pin Header上

2.4英寸LCD模块	Pin Number
VCC	17
GND	39
DIN	19
CLK	23
CS	24
DC	40
RST	11
BL	18

图 9-3 将2.4英寸LCD模块连接到40-Pin Header上





9.2.2. 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：[LCD Demo](#)。
2. 执行以下命令将代码复制到单板计算机上。例如，visionfive2。

```
tar -xvf LCD_Demo.tar.gz  
cd visionfive2/  
.main 2.4
```

结果：

依次显示以下两张图片。左图为官方示例图，右图为昉·星光 2。

图 9-5 示例输出

