



StarFive  
赛昉科技

# 使用昉·星光单板计算机的 IIC读取SHTC3数据

应用说明

版本： 1.1

日期： 2022/01/12

Doc ID: VisionFive-ANCH-002-1.1

版本： 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](mailto:sales@starfivetech.com)（销售） [support@starfivetech.com](mailto: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通过利用昉·星光单板计算机的IIC总线，运行示例程序，以读取SHTC3数据的步骤。






## 修订历史

表 0-1 修订历史

版本	发布说明	修订
V1.0	2021/12/15	首次发布。
V1.1	2022/01/12	<ul style="list-style-type: none"><li>• 新增了&lt;User_Name&gt;字段说明。</li><li>• 将演示文件名更新为test-shtc3.c。</li></ul>

## 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
<b>1. 介绍.....</b>	<b>7</b>
<b>2. 准备.....</b>	<b>8</b>
2.1. 准备硬件.....	8
2.1.1. 连接硬件.....	8
2.2. 准备软件.....	10
<b>3. 执行演示代码.....</b>	<b>11</b>



Starfive  
赛昉科技

# 表格清单

表 0-1 修订历史.....	iii
表 2-1 硬件准备.....	8
表 2-2 将Sense Hat (B)连接到40-Pin Header上.....	9



# 插图清单

图 2-1 将Sense Hat (B)连接到40-Pin Header上.....	9
图 2-2 将Sense Hat (B)连接到40-Pin Header上.....	10
图 3-1 示例输出.....	14



---

# 1. 介绍

本应用说明提供使用通过利用昉·星光单板计算机的IIC总线，运行示例程序，以读取SHTC3数据的步骤。



## 2. 准备

在执行演示代码之前，请确保您已准备好以下设备：

### 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"><li>• 星光板</li><li>• 昉·星光单板计算机</li></ul>
通用	M	<ul style="list-style-type: none"><li>• 容量不低于32 GB的Micro-SD卡</li><li>• Micro-SD卡读卡器</li><li>• 计算机 (PC/Mac/Linux)</li><li>• USB转串口转换器 (3.3 V I/O)</li><li>• 网线</li><li>• 电源适配器 (5 V/ 3 A)</li><li>• USB Type-C数据线</li></ul>	上述项目用于将Fedora OS烧录到Micro-SD上。
I2C演示	M	<ul style="list-style-type: none"><li>• Sense Hat (B)</li><li>• 杜邦线</li></ul>	-



注：

\*: M: 必须。O: 可选

#### 2.1.1. 连接硬件

以下图表描述了如何将Sense HAT连接到40-Pin Header上：



表 2-2 将Sense Hat (B)连接到40-Pin Header上

Sense HAT (B)	40-Pin GPIO Header	
	引脚序号	引脚名
3V3	1	3.3V Power
GND	9	GND
SDA	3	GPIO48 (I2C SDA)
SCL	5	GPIO47 (I2C SCL)

图 2-1 将Sense Hat (B)连接到40-Pin Header上

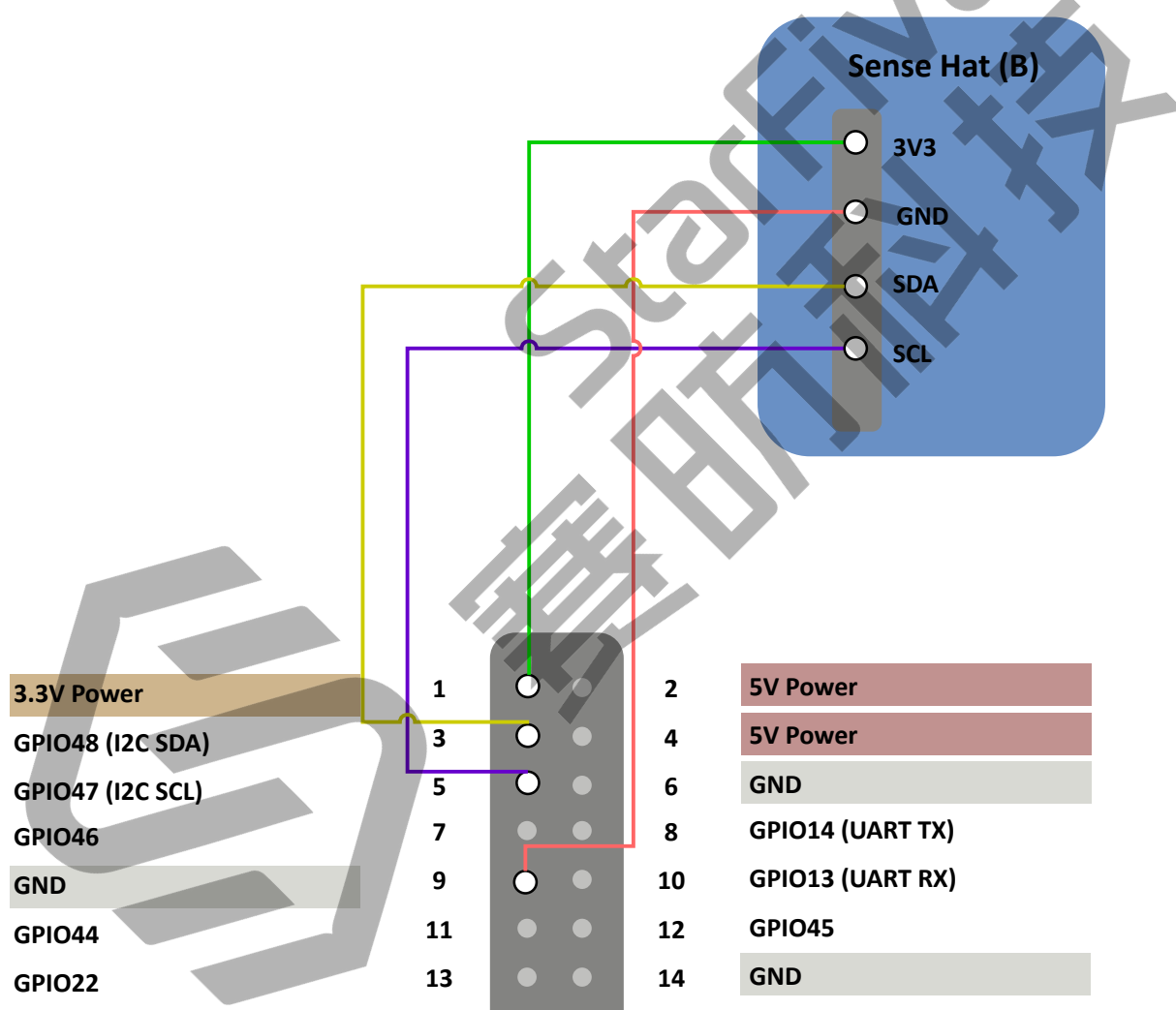
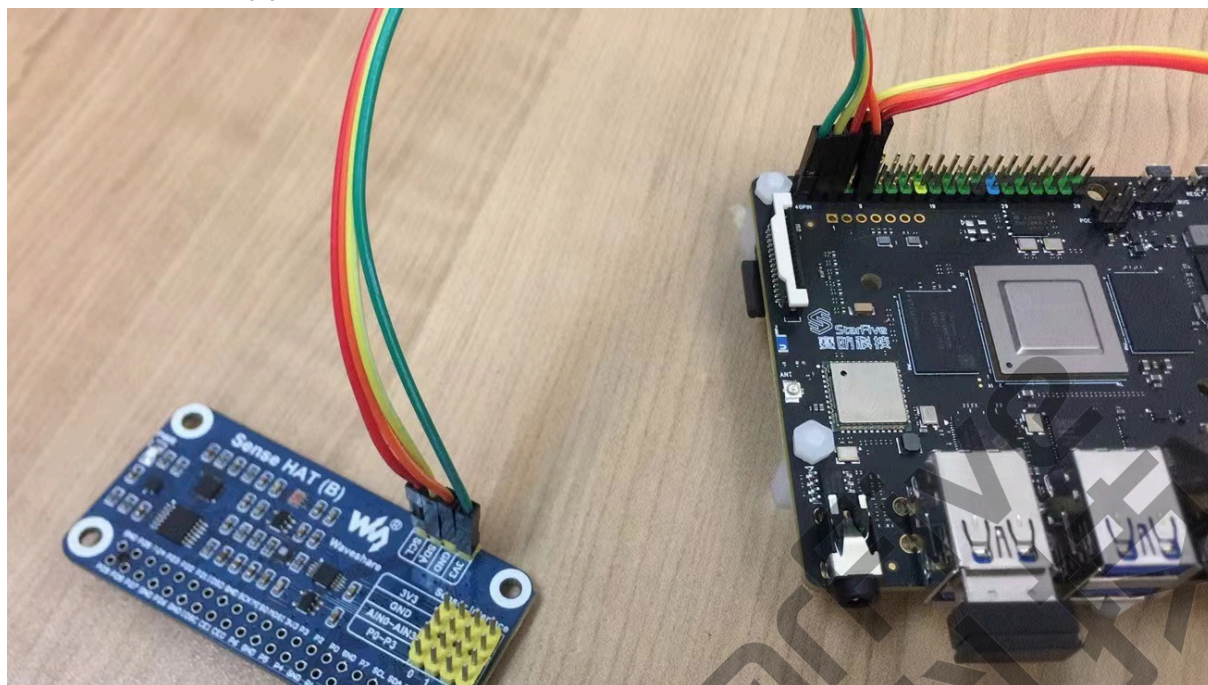


图 2-2 将Sense Hat (B)连接到40-Pin Header上



## 2.2. 准备软件

确认按照以下步骤进行操作：

- 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。
- 按照《赛昉科技40-Pin GPIO Header用户指南》中“配置dts文件”部分配置dts文件。

## 3. 执行演示代码

执行以下操作，以运行演示代码：

1. 将以下源代码保存为`test-shtc3.c`，并复制到Ubuntu下所需的路径：

```
#include <stdio.h>
#include <math.h>
#include <unistd.h>
#include <stdlib.h>

#include <stdint.h>
#include <getopt.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <linux/types.h>
#include <linux/i2c.h>
#include <linux/i2c-dev.h>

//i2c address
#define SHTC3_I2C_ADDRESS      0x70
#define I2C_DEVICE             "/dev/i2c-1"

//Commands
#define SHTC3_WakeUp           0x3517
#define SHTC3_Sleep            0xB098
#define SHTC3_NM_CE_ReadTH    0x7CA2
#define SHTC3_NM_CE_ReadRH    0x5C24
#define SHTC3_NM_CD_ReadTH    0x7866
#define SHTC3_NM_CD_ReadRH    0x58E0
#define SHTC3_LM_CE_ReadTH    0x6458
#define SHTC3_LM_CE_ReadRH    0x44DE
#define SHTC3_LM_CD_ReadTH    0x609C
#define SHTC3_LM_CD_ReadRH    0x401A
#define SHTC3_Software_RES     0x401A
#define SHTC3_ID               0xEFC8

#define CRC_POLYNOMIAL         0x131 //  $P(x) = x^8 + x^5 + x^4 + 1$ 
    = 100110001

float TH_Value,RH_Value;
char checksum;
int fd;
char SDA = 8;
char SCL = 9;
char SHTC3_CheckCrc(char data[],unsigned char len,unsigned char
checksum)
{
```

```

unsigned char bit;          // bit mask
unsigned char crc = 0xFF;  // calculated checksum
unsigned char byteCtr;     // byte counter
// calculates 8-Bit checksum with given polynomial
for(byteCtr = 0; byteCtr < len; byteCtr++) {
    crc ^= (data[byteCtr]);
    for(bit = 8; bit > 0; --bit) {
        if(crc & 0x80) {
            crc = (crc << 1) ^ CRC_POLYNOMIAL;
        } else {
            crc = (crc << 1);
        }
    }
}

// verify checksum
if(crc != checksum) {
    return 1;                //Error
} else {
    return 0;                //No error
}
}

void SHTC3_WriteCommand(unsigned short cmd)
{
    char buf[] = { (cmd>>8) ,cmd};
    write(fd,buf,2);
}

void SHTC3_WAKEUP()
{
    SHTC3_WriteCommand(SHTC3_WakeUp);           // Write wake_up
    command
    usleep(300);                                //Delay 300us
}

void SHTC3_SLEEP()
{
    // bcm2835_i2c_begin();
    SHTC3_WriteCommand(SHTC3_Sleep);           // Write
    sleep command
}

void SHTC3_SOFT_RESET()
{
    SHTC3_WriteCommand(SHTC3_Software_RES);    // Write
    reset command
    usleep(300);                                //Delay 300us
}
}

```

```

void SHTC3_Read_DATA()
{
    unsigned short TH_DATA,RH_DATA;
    char buf[3];
    SHTC3_WriteCommand(SHTC3_NM_CD_ReadTH);           //Read
    temperature first, clock stretching disabled (polling)
    usleep(20000);
    read(fd, buf, 3);

    checksum=buf[2];
    if(!SHTC3_CheckCrc(buf,2,checksum))
        TH_DATA=(buf[0]<<8|buf[1]);

    SHTC3_WriteCommand(SHTC3_NM_CD_ReadRH);           //Read
    temperature first, clock stretching disabled (polling)
    usleep(20000);
    read(fd, buf, 3);

    checksum=buf[2];
    if(!SHTC3_CheckCrc(buf,2,checksum))
        RH_DATA=(buf[0]<<8|buf[1]);

    TH_Value=175 * (float)TH_DATA / 65536.0f - 45.0f;   //Calculate
    temperature value
    RH_Value=100 * (float)RH_DATA / 65536.0f;           //Calculate
    humidity value
}

int main()
{
    printf("\n SHTC3 Sensor Test Program ...\n");
    fd = open(I2C_DEVICE, O_RDWR);
    if(fd < 0)
    {
        printf("Fail to open i2c device\r\n");
        return -1;
    }
    else
    {
        printf("Fopen : %s\r\n", I2C_DEVICE);
    }

    if(ioctl(fd, I2C_SLAVE, SHTC3_I2C_ADDRESS) < 0)
    {
        printf("I2C: Failed to connect to the device\n");
        return -1;
    }

    SHTC_SOFT_RESET();
}

```

```

while (1)
{
    SHTC3_Read_DATA();
    SHTC3_SLEEP();
    SHTC3_WAKEUP();

    printf("Temperature = %6.2f°C , Humidity = %6.2f%% \r\n",
    TH_Value, RH_Value);
}
return 0;
}

```

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。
- 安装完成后, 运行以下命令检查版本是否已更新: `linus@star-five$ riscv64-linux-gnu-gcc -v`。以下是示例输出:

图 3-1 示例输出

```

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

```

3. 执行以下命令进行编译:

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

结果:

- 系统将在同一目录生成名为 `test-shtc3` 的文件。
- 如在输出中显示 `UCB RISC-V`, 则表示在 RISC-V 平台上运行的可执行代码已成功生成:

```

Riscv@starfive:~/work/app/iic$ file test-shtc3
test-shtc3: 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]=560aeb713ece667ab5f3a5f0dcbd75a149216e6f,
not strippe

```

4. 在 Ubuntu 环境中执行以下操作, 通过以太网, 将 `test-shtc3` 文件中的可执行代码上传到单板计算机上:

```
rsync ./test-shtc3 <User_Name>@<Board_IP_Address>:/home/riscv
```

**信息:**

- <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-shtc3 riscv@192.168.92.133:/home/riscv
```

5. 执行以下命令，以在昉·星光单板计算机上运行代码:

```
./test-shtc3Copy
```

现在，我们已经成功读取了温度和湿度数据。

**结果:**

如输出以下结果，则表示执行成功:

```
[root@fedora-starfive test]# ./test-shtc3

SHTC3 Sensor Test Program ...
Fopen : /dev/i2c-1
Temperature = -7.30°C , Humidity = 21.28%
Temperature = 25.74°C , Humidity = 21.26%
Temperature = 25.72°C , Humidity = 21.24%
Temperature = 25.74°C , Humidity = 21.24%
Temperature = 25.75°C , Humidity = 21.21%
Temperature = 25.75°C , Humidity = 21.21%
Temperature = 25.77°C , Humidity = 21.19%
Temperature = 25.76°C , Humidity = 21.18%
Temperature = 25.76°C , Humidity = 21.15%
Temperature = 25.77°C , Humidity = 21.15%
Temperature = 25.74°C , Humidity = 21.12%
Temperature = 25.76°C , Humidity = 21.12%
Temperature = 25.75°C , Humidity = 21.09%
Temperature = 25.78°C , Humidity = 21.09%
Temperature = 25.75°C , Humidity = 21.08%
Temperature = 25.81°C , Humidity = 21.09%
Temperature = 25.76°C , Humidity = 21.05%
Temperature = 25.77°C , Humidity = 21.03%
Temperature = 25.76°C , Humidity = 21.03%
Temperature = 25.78°C , Humidity = 21.02%
Temperature = 25.77°C , Humidity = 21.01%
Temperature = 25.75°C , Humidity = 20.98%
Temperature = 25.78°C , Humidity = 20.99%
Temperature = 25.75°C , Humidity = 20.96%
Temperature = 25.78°C , Humidity = 20.95%
```