



StarFive
赛昉科技

使用昉·星光的IIC读取 SHTC3数据

Python语言版本

应用说明

版本： 1.0

日期： 2022/07/29

Doc ID: VisionFive-ANCH-010-1.0

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通过Python利用昉·星光的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	2022/07/29	首次发布。

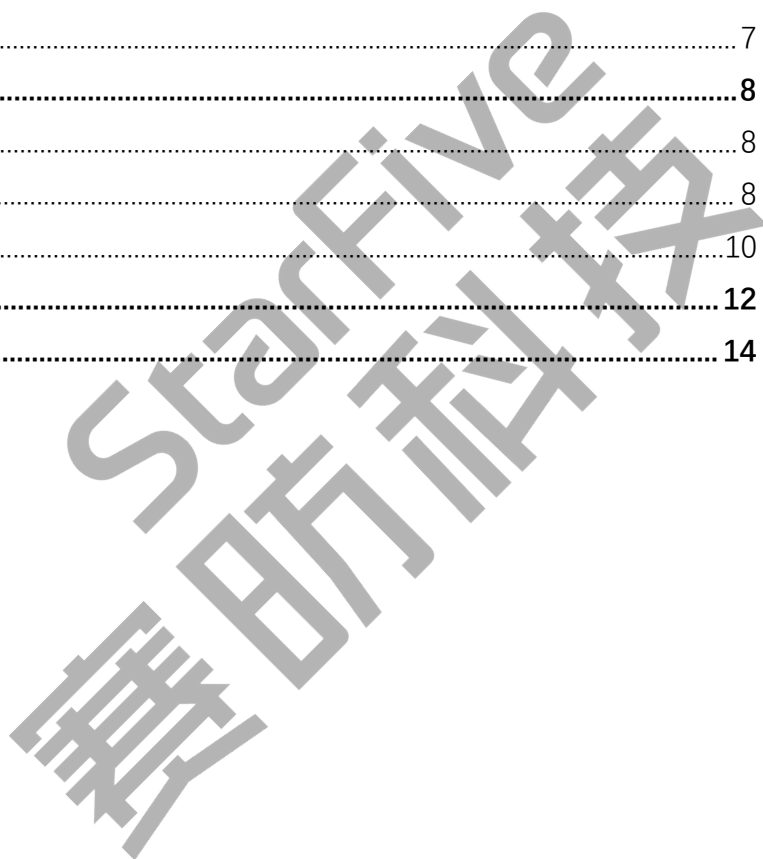
注释和注意事项

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.....	8
2.2. 准备软件.....	10
3. 执行演示代码.....	12
4. 演示源代码.....	14



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 1-1 40-Pin定义.....	7
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



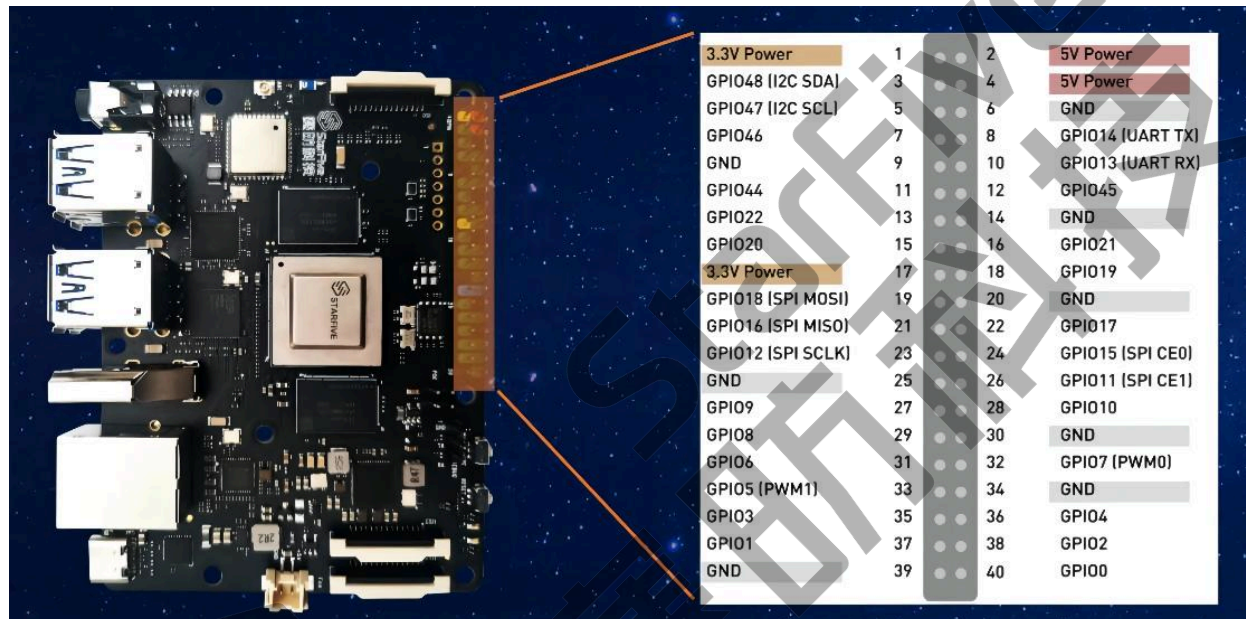
1. 介绍

本应用说明提供使用通过Python利用昉·星光的IIC总线，运行示例程序，以读取SHTC3数据的步骤。

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。
I2C演示	M	<ul style="list-style-type: none">• Sense Hat (B)• 杜邦线	-



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

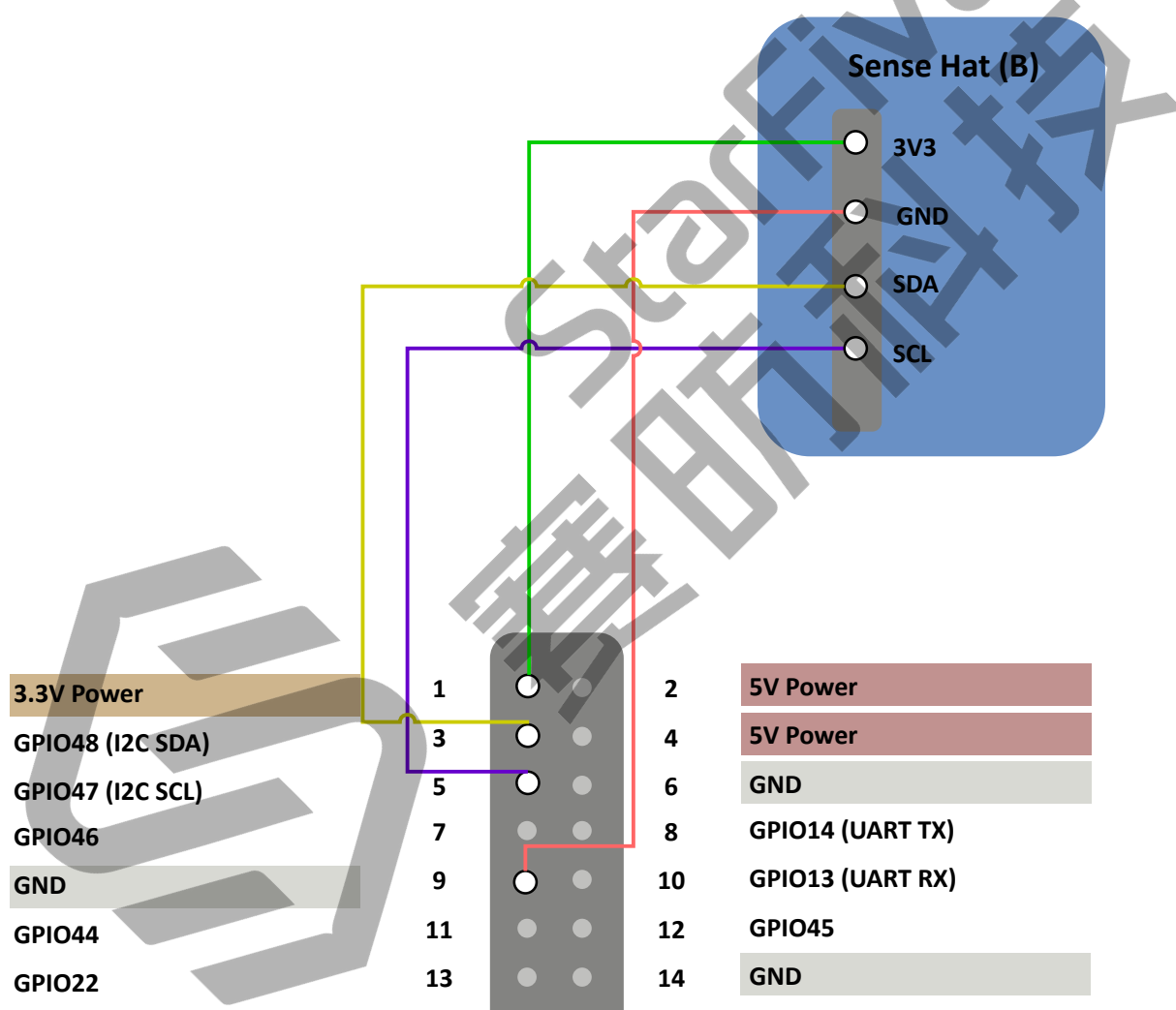


图 2-2 将Sense Hat (B)连接到40-Pin Header上



2.2. 准备软件

确认按照以下步骤进行操作：

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：

i 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·星光的Fedora上执行以下步骤，运行演示代码：

1. 找到测试代码I2C_Sense_Hat.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 I2C_Sense_Hat.py
```

或者，您也可以执行以下命令：

```
sudo python3 I2C_Sense_Hat.py
```

结果：

终端上输出了温湿度数据：

```
[riscv@fedora-starfive sample-code]$ sudo python3 led.py
Enter delay(seconds): /dev/i2c-1
Temperature = 27.85°C , Humidity = 56.59 %

Temperature = 27.83°C , Humidity = 56.60 %

Temperature = 27.85°C , Humidity = 56.61 %

Temperature = 27.86°C , Humidity = 56.60 %

Temperature = 27.86°C , Humidity = 56.60 %
```

Temperature = 27.80°C , Humidity = 56.60 %

Temperature = 27.87°C , Humidity = 56.60 %



4. 演示源代码

本演示中的资源代码仅作为参考。

I2C_Sense_Hat.py:

```
#!/usr/bin/python
'''
Please make sure the sense HAT(B) is connected to the correct pins.
The following table describes how to connect the Sense HAT(B) to the 40-pin
header.
Sense HAT (B)-----
__Sense HAT (B)__Pin Number__Pin Name
    3V3             1           3.3 V Power
    GND             9           GND
    SDA             3           I2C SDA
    SCL             5           I2C SCL
-----
'''

import sys
import struct
import fcntl
import os
import math
import time
import VisionFive.i2c as I2C

SHTC3_I2C_ADDRESS = 0x70
I2C_SLAVE = 0x0703
I2C_DEVICE = "/dev/i2c-1"

##Commands
cmd_dict = {
    "SHTC3_WakeUp": 0x3517,
    "SHTC3_Sleep": 0xB098,
    "SHTC3_NM_CE_ReadTH": 0x7CA2,
    "SHTC3_NM_CE_ReadRH": 0x5C24,
    "SHTC3_NM_CD_ReadTH": 0x7866,
    "SHTC3_NM_CD_ReadRH": 0x58E0,
    "SHTC3_LM_CE_ReadTH": 0x6458,
    "SHTC3_LM_CE_ReadRH": 0x44DE,
    "SHTC3_LM_CD_ReadTH": 0x609C,
    "SHTC3_LM_CD_ReadRH": 0x401A,
    "SHTC3_Software_RES": 0x401A,
    "SHTC3_ID": 0xEFC8,
    "CRC_POLYNOMIAL": 0x131,
}
```

```

def SHTC3_CheckCrc(data, len, checksum):
    crc = 0xff
    for byteCtr in range(0, len):
        crc ^= data[byteCtr]
        for bit in range(8, 0, -1):
            if(crc & 0x80):
                crc = (crc << 1) ^ cmd_dict["CRC_POLYNOMIAL"]
            else:
                crc = crc << 1
    if (crc != checksum):
        return 1
    else:
        return 0

def SHTC3_WriteCommand(cmd):
    buf0 = (cmd >> 8) & 0xff
    buf1 = cmd & 0xff
    buf = [buf0, buf1]
    I2C.write(buf)

def SHTC3_WAKEUP():
    SHTC3_WriteCommand(cmd_dict["SHTC3_WakeUp"])
    time.sleep(0.03)

def SHTC3_SLEEP():
    SHTC3_WriteCommand(cmd_dict["SHTC3_Sleep"])

def SHTC_SOFT_RESET():
    SHTC3_WriteCommand(cmd_dict["SHTC3_Software_RES"])
    time.sleep(0.03)

def getdata():
    time.sleep(0.02)
    buf_list = I2C.read(3)
    checksum = buf_list[2]
    DATA = 0
    if (not SHTC3_CheckCrc(buf_list, 2, checksum)):
        DATA = (buf_list[0] << 8 | buf_list[1])
    return DATA

def SHTC3_Read_DATA():
    SHTC3_WriteCommand(cmd_dict["SHTC3_NM_CD_ReadTH"])
    TH_DATA = getdata()
    SHTC3_WriteCommand(cmd_dict["SHTC3_NM_CD_ReadRH"])
    RH_DATA = getdata()
    TH_DATA = 175 * TH_DATA / 65536.0 - 45.0    #Calculate the temperature
value.
    RH_DATA = 100 * RH_DATA / 65536.0        #Calculate the humidity value.

```

```
DATA = [TH_DATA, RH_DATA]
return DATA

def getTem():
    SHTC3_WriteCommand(cmd_dict["SHTC3_NM_CD_ReadTH"])
    TH_DATA = getdata()
    TH_DATA = 175 * TH_DATA / 65536.0 - 45.0    #Calculate the temperature
value.
    return TH_DATA

def getHum():
    SHTC3_WriteCommand(cmd_dict["SHTC3_NM_CD_ReadRH"])
    RH_DATA = getdata()
    RH_DATA = 100 * RH_DATA / 65536.0        #Calculate the humidity value.
    return RH_DATA

def main():
    #Open the Sense HAT by I2C.
    ret = I2C.open(I2C_DEVICE, SHTC3_I2C_ADDRESS)
    if (ret < 0):
        return 0

    SHTC_SOFT_RESET()
    i = 0
    while i < 7:
        Temp = getTem()
        Hum = getHum()
        SHTC3_SLEEP()
        SHTC3_WAKEUP()
        print("Temperature = {:.2f}°C , Humidity = {:.2f} %\n".format(Temp,
Hum))
        i = i + 1

    I2C.close()
    return 0

if __name__ == "__main__":
    sys.exit(main())
```