



StarFive
赛昉科技

使用昉·星光 2的IIC读取SHTC3数据

Python语言版本

应用指南

版本： 1.11

日期： 2025/04/27

Doc ID: VisionFive 2-ANCH-004

法律声明

阅读本文件前的重要法律告知。

版权注释

版权 ©上海赛昉半导体科技有限公司，2025。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。上海赛昉半导体科技有限公司（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：中国（上海）自由贸易试验区盛夏路61弄张润大厦2号电梯楼层5层（实际楼层4层）06室

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 2的IIC总线，运行示例程序，以读取SHTC3数据的步骤。






修订历史

Table 0-1 修订历史

版本	发布说明	修订
1.11	2025/04/27	<ul style="list-style-type: none">修正执行演示代码 (on page 12)中的笔误。
1.1	2023/06/08	<ul style="list-style-type: none">在40-Pin GPIO Header定义 (on page 7)增加注释。在准备软件 (on page 9)中更新安装方式。新增资源下载 (on page 15)和立即购买 (on page 16)章节。
1.0	2022/12/15	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：

-  **Tip:**
建议如何在某个主题或步骤中应用信息。
-  **Note:**
解释某个特例或阐释一个重要的点。
-  **Important:**
指出与某个主题或步骤有关的重要信息。
-  **CAUTION:**
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。
-  **Warning:**
表明某个操作或步骤可能导致物理伤害或硬件损坏。

Contents

List of Tables.....	5
List of Figures.....	6
法律声明.....	2
前言.....	3
1. 产品简介.....	7
1.1. 40-Pin GPIO Header定义.....	7
2. 准备.....	8
2.1. 运行环境要求.....	8
2.2. 准备硬件.....	8
2.2.1. 连接硬件.....	8
2.3. 准备软件.....	9
3. 执行演示代码.....	12
4. 演示源代码.....	13
5. 资源下载.....	15
6. 立即购买.....	16

List of Tables

Table 0-1 修订历史.....3

Table 2-1 硬件准备.....8

Table 2-2 将Sense Hat (B) 连接到40-Pin GPIO Header上.....8

List of Figures

Figure 1-1 40-Pin GPIO Header定义.....7
Figure 2-1 将Sense Hat (B) 连接到40-Pin GPIO Header上.....9

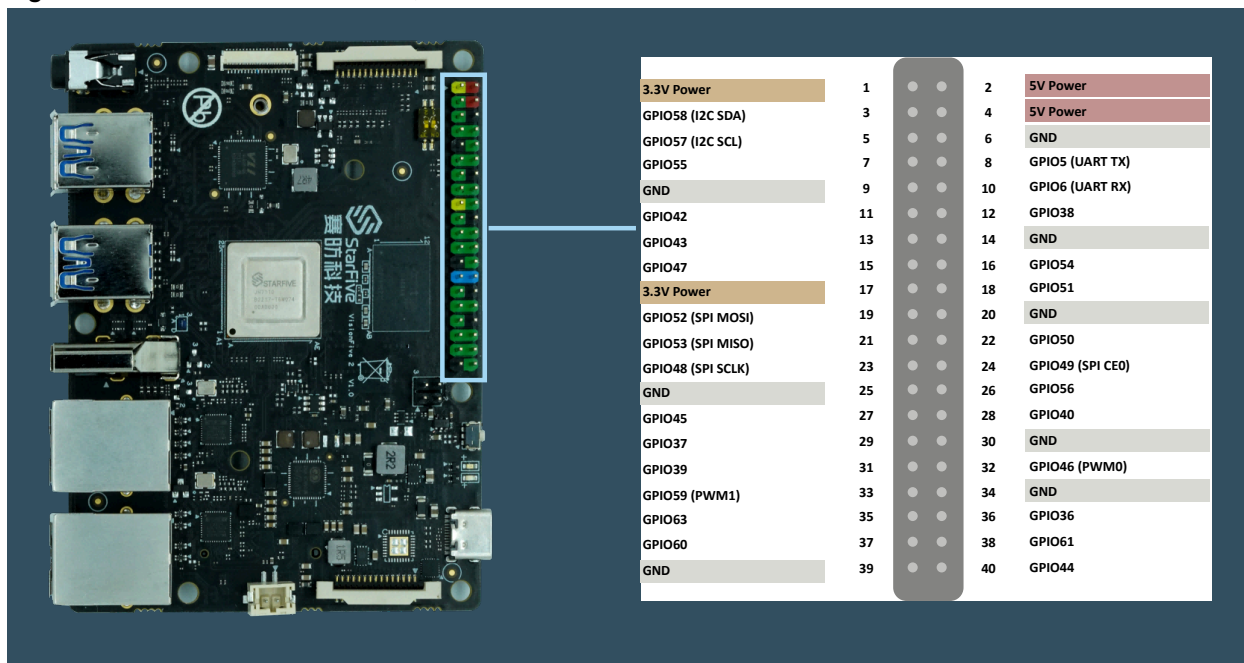
1. 产品简介

本应用说明提供使用通过Python利用昉·星光 2的IIC总线，运行示例程序，以读取SHTC3数据的步骤。

1.1. 40-Pin GPIO Header定义

下图显示了40-pin GPIO Header的位置：

Figure 1-1 40-Pin GPIO Header定义



Note:

功能复用pin脚已初始化，不可作为通用GPIO使用。

2. 准备

在执行演示程序之前，务必确认已准备好以下项目：

2.1. 运行环境要求

该演示运行环境要求如下：

- Linux内核版本：Linux 5.15
- 操作系统：Debian 12
- 硬件版本：昉·星光 2
- SoC：昉·惊鸿-7110

2.2. 准备硬件

在执行演示程序之前，请务必准备以下硬件：

Table 2-1 硬件准备

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



Note:

*: M: 必须。O: 可选

2.2.1. 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：

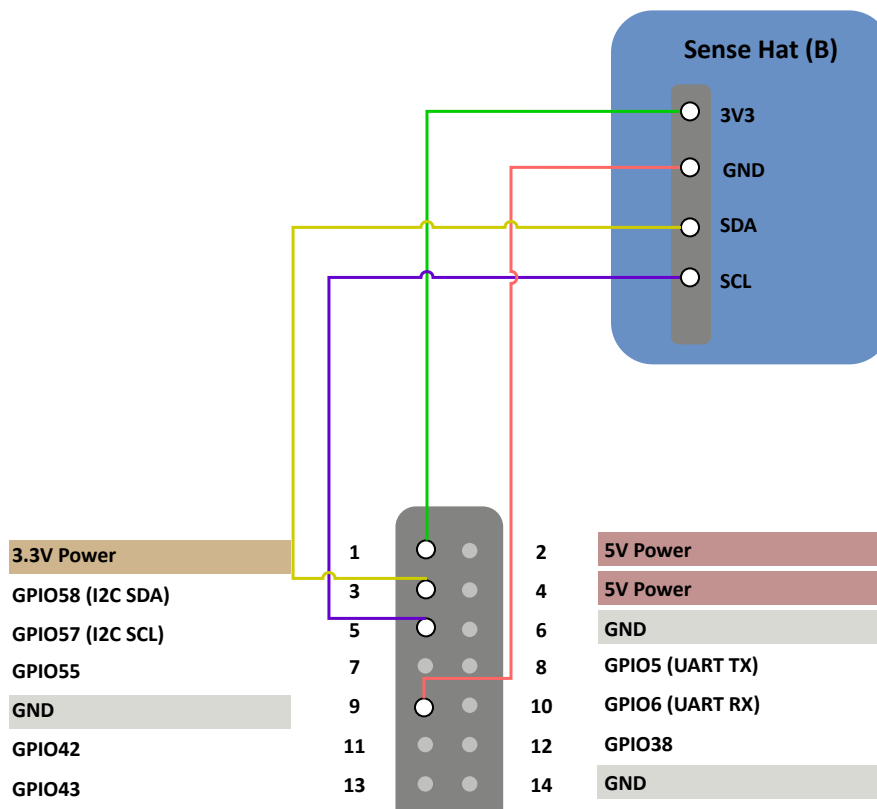
Table 2-2 将Sense Hat (B) 连接到40-Pin GPIO Header上

Sense HAT (B)	40-Pin GPIO Header	
	Pin Number	Pin Name
3V3	1	3.3V 电压
GND	9	GND

Table 2-2 将Sense Hat (B) 连接到40-Pin GPIO Header上 (continued)

Sense HAT (B)	40-Pin GPIO Header	
	Pin Number	Pin Name
SDA	3	GPIO58 (I2C SDA)
SCL	5	GPIO57 (I2C SCL)

Figure 2-1 将Sense Hat (B) 连接到40-Pin GPIO Header上



2.3. 准备软件

确认按照以下步骤进行操作：



Note:

该Python应用VisionFive.gpio适用于昉·星光单板计算机、昉·星光 2和昉·惊鸿-7110 EVB。

- 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”章节，将Debian OS烧录到Micro-SD卡上。
- 登录Debian并确保昉·星光 2已联网。有关详细说明，请参阅[《昉·星光 2单板计算机快速参考手册》](#)中“通过以太网使用SSH”或“使用USB转串口转换器”章节。
- 在Debian上扩展分区，请参见[《昉·星光 2单板计算机快速参考手册》](#)中“扩展分区”章节。
- 执行以下命令，在Debian系统上安装PIP：

```
apt-get install python3-pip
```

- 在昉·星光 2 Debian上执行pip命令，以安装VisionFive.gpio包：

**Note:**

由于pypi.org官网尚不支持上传RISC-V平台的whl安装包，不能直接使用`pip install VisionFive.gpio`命令在线安装，因此请按照以下步骤安装VisionFive.gpio包。

- a. 执行以下命令，安装依赖包：

```
apt install libxml2-dev libxslt-dev
python3 -m pip install requests wget bs4
```

- b. 执行以下命令，运行安装脚本Install_VisionFive_gpio.py：

```
python3 Install_VisionFive_gpio.py
```

安装脚本代码如下：

```
import requests
import wget
import sys
import os
from bs4 import BeautifulSoup

def parse_data(link_addr, class_type, key_str):
    req = requests.get(url=link_addr)
    req.encoding = "utf-8"
    html=req.text
    soup = BeautifulSoup(req.text,features="html.parser")
    package_version = soup.find(class_type,class_=key_str)
    dd = package_version.text.strip()
    data = dd.split()
    return data

def parse_link(link_addr, class_type, key_str):
    req = requests.get(url=link_addr)
    req.encoding = "utf-8"
    html=req.text
    soup = BeautifulSoup(req.text,features="html.parser")
    search_data = soup.find(class_type,class_=key_str)
    search_data_2 = search_data.find("a")
    dl_link_get = search_data_2.get("href")
    return dl_link_get

def get_dl_addr_page():
    link_address = "https://pypi.org/project/VisionFive.gpio/#history"
    key_str = "release__version"
    class_key = "p"
    data_get = parse_data(link_address, class_key, key_str)
    latest_version = data_get[0]

    dl_addr_page
    = "https://pypi.org/project/VisionFive.gpio/{}/#files".format(latest_version)

    return dl_addr_page

def get_dl_addr_of_latest_version(link_addr):
    key_str = "card file__card"
    class_key = "div"
    addr_get = parse_link(link_addr, class_key, key_str)

    return addr_get

def main():
    dl_addr_p = get_dl_addr_page()
    whl_dl_addr = get_dl_addr_of_latest_version(dl_addr_p)

    whl_name = whl_dl_addr.split("/")[-1]
    whl_name_suffix = os.path.splitext(whl_name)[-1]
    whl_name_prefix = os.path.splitext(whl_name)[0]
    whl_name_prefix_no_platform = whl_name_prefix[0: len(whl_name_prefix) - 3]
    new_platform = "linux_riscv64"
```



```
rename_whl_name = "{}{}{}".format(whl_name_prefix_no_platform, new_platform,
whl_name_suffix)

wget.download(whl_dl_addr, out=rename_whl_name)

os.system("pip install " + rename_whl_name)
os.system("rm -rf " + rename_whl_name)

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

3. 执行演示代码

执行以下操作，以在昉·星光 2的Debian系统上运行演示代码：

1. 找到测试代码I2C_Sense_Hat.py所在的目录：

- a. 执行以下命令安装依赖：

```
pip install pillow
```

- b. 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
```



Note:

实际输出取决于应用的安装方式。

- c. 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
```

- d. 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 I2C_Sense_Hat.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 SHTC3_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

    SHTC3_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())

```

5. 资源下载

点击本栏找到所有的代码下载资源。

本页包括所有赛昉科技提供的代码下载资源。

- [RVspace Wiki](#)
- [应用中心](#)
- [文档中心](#)
- [技术论坛](#)
- [昉·星光 2 GitHub代码仓](#)
- [昉·星光 2 Debian操作系统下载](#)
- [代码下载 \(赛昉科技官方GitHub页面\)](#)
- [所有开源技术文档](#)

6. 立即购买

点击本栏获取在线购买链接和配件购买链接。

购买单板计算机

点击以下页面，您可以找到所在地区的经销商，或覆盖全球的销售渠道，以购买昉·星光 2 单板计算机。

- [购买昉·星光 2 开发板](#)

购买配件

点击以下页面，您可以找到所有昉·星光 2 单板计算机已验证适配的配件及其购买链接。

- [购买配件](#)