StarFive 赛昉科技

Using VisionFive 2 IIC to Read SHTC3 Data

with Python

Application Note

Version: 1.2

Date: 2025/08/06

Doc ID: VisionFive2-ANEN-004

Legal Statements

Important legal notice before reading this documentation.

PROPRIETARY NOTICE

Copyright©Shanghai StarFive Semiconductor Co., Ltd., 2025. All rights reserved.

Information in this document is provided "as is," with all faults. Contents may be periodically updated or revised due to the product development. Shanghai StarFive Semiconductor Co., Ltd. (hereinafter "StarFive") reserves the right to make changes without further notice to any products herein.

StarFive expressly disclaims all warranties, representations, and conditions of any kind, whether express or implied, including, but not limited to, the implied warranties or conditions of merchantability, fitness for a particular purpose and non-infringement.

StarFive does not assume any liability rising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation indirect, incidental, special, exemplary, or consequential damages.

All material appearing in this document is protected by copyright and is the property of StarFive. You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. StarFive authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services.

Contact Us

Address: Room 506, Building 2, No. 61 Shengxia Rd., China (Shanghai) Pilot Free Trade Zone, Shanghai, 201203, China

Website: http://www.starfivetech.com

Email: sales@starfivetech.com(sales) , support@starfivetech.com(support)

Preface

About this guide and technical support information.

About this document

This application note provides steps to use VisionFive 2's IIC to read SHTC3 data through an example program with Python.

Revision History

Table 0-1 Revision History

Version	Released	Revision
1.2	2025/08/06	Updated the Linux and OS version in Environment Requirements *(on page 8)*. Updated the steps in Preparing Software *(on page 9)*. Updated the steps in Running Demo Code *(on page 12)*.
1.1	2023/06/08	• Added a note in 40-Pin GPIO Header Definition *(on page 7)*. • Updated the method for installing `VisionFive.gpio` package in Preparing Software *(on page 9)*. • Added Resources *(on page 16)* and Buy Now *(on page 17)* chapters.
1.0	2022/11/30	The first official release.

Notes and notices

The following notes and notices might appear in this guide:

- **Tip:**
  Suggests how to apply the information in a topic or step.
- **Note:**
  Explains a special case or expands on an important point.
- **Important:**
  Points out critical information concerning a topic or step.
- **CAUTION:**
  Indicates that an action or step can cause loss of data, security problems, or performance issues.
- **Warning:**
  Indicates that an action or step can result in physical harm or cause damage to hardware.

Contents

© 2018-2025 StarFive Technology
All rights reserved

List of Tables

StarFive 赛昉科技

© 2018-2025 StarFive Technology
All rights reserved

List of Figures

© 2018-2025 StarFive Technology
All rights reserved

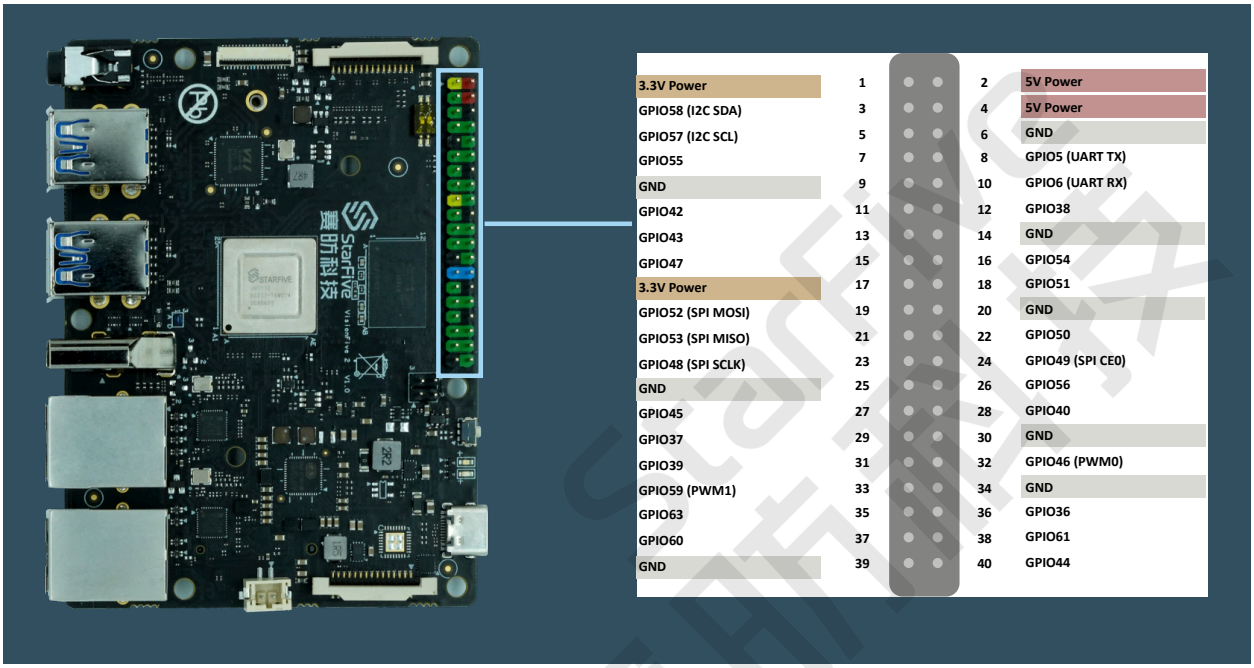
1. Introduction

This application note provides steps to use VisionFive 2's IIC to read SHTC3 data through an example program with Python.

1.1. 40-Pin GPIO Header Definition

The following figure shows the location of the 40-pin header on VisionFive 2.

Figure 1-1 40-Pin GPIO Header Definition



Note:
The multiplexed pin has been initialized and cannot be used as a general GPIO.

© 2018-2025 StarFive Technology
All rights reserved

2. Preparation

Before executing the demo program, make sure you prepare the following:

2.1. Environment Requirements

The environment requirements are as follows:

- Linux Kernel: Linux 6.6
- OS: Debian 13
- SBC: VisionFive 2
- SoC: JH-7110

2.2. Preparing Hardware

Prepare the following hardware items before running the demo code:

Table 2-1 Hardware Preparation

Type	M/O*	Item	Notes
General	M	VisionFive 2 Board	-
General	M	• 32 GB (or more) micro-SD card • Micro-SD card reader • Computer (Windows/Mac OS/Linux) • USB to serial converter (3.3 V I/O) • Ethernet cable • Power adapter (5 V / 3 A) • USB Type-C Cable	These items are used for flashing Debian OS into a Micro-SD card.
I2C Demo	M	• Sense Hat (B) • Dupont Line	-

Note:
*: M: Mandatory, O: Optional

2.2.1. Hardware Setup

The following table and figure describe how to connect Sense HAT to the 40-pin header:

Table 2-2 Connect Sense Hat (B) to the 40-Pin Header

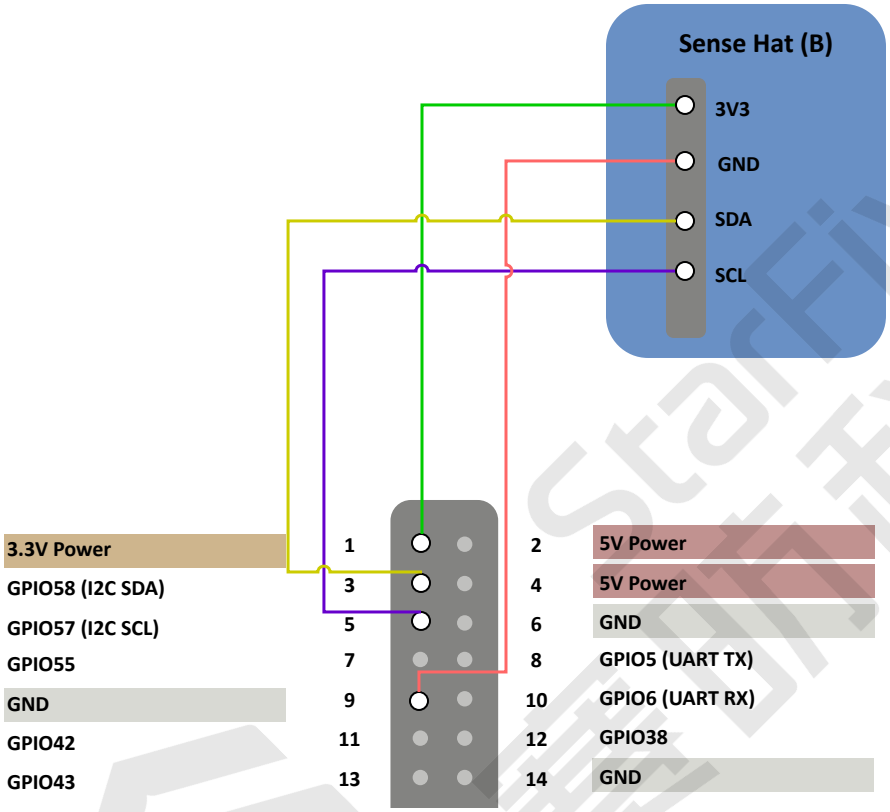
Sense HAT (B)	40-Pin GPIO Header	
	Pin Number	Pin Name
3V3	1	3.3V Power
GND	9	GND

© 2018-2025 StarFive Technology
All rights reserved

Table 2-2 Connect Sense Hat (B) to the 40-Pin Header (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 Connect Sense Hat (B) to the 40-Pin GPIO Header



2.3. Preparing Software

Make sure the following procedures are performed:

Note:
The python project, `VisionFive.gpio`, is applicable for VisionFive, VisionFive 2 and JH-7110 EVB.

1. Flash Debian OS into a Micro-SD card as described in the *Flashing Fedora OS to a Micro-SD Card* section in the *VisionFive 2 Single Board Computer Quick Start Guide*.
2. Log into the Debian and make sure VisionFive 2 is connected to the Internet. For detailed instructions, refer to the Using SSH over Ethernet or *Using a USB to Serial Converter* section in the *VisionFive 2 Single Board Computer Quick Start Guide*.
3. Extend the partition on Debian as described in *Extend Partition* in the *VisionFive 2 Single Board Computer Quick Start Guide*.
4. Execute the following command to install and create a Python3 Virtual Environment on Debian:

```
sudo apt install python3-venv
python3 -m venv myvenv
```


© 2018-2025 StarFive Technology
All rights reserved

Note:

You may rename “myvenv” according to your preference.

5. Execute the `pip` command on VisionFive 2 Debian to install the `VisionFive.gpio` package:

Note:

Due to the fact that `pypi.org` official website does not yet support uploading `whl` installation packages for the RISC-V platform, so it cannot directly execute `python3 -m pip install VisionFive.gpio` command to install online.

Please follow the steps below to install the `VisionFive.gpio` package.

a. Execute the following command to install dependent package within the newly created virtual environment:

```
sudo apt install libxml2-dev libxslt-dev
source ./myvenv/bin/activate
python3 -m pip install requests wget bs4
```

b. Execute the following command to run the installation script `Install_VisionFive_gpio.py`:

```
python3 Install_VisionFive_gpio.py
```

The installation script codes are as follows:

```
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):
    version_list = []
    req = requests.get(url=link_addr)
    req.encoding = "utf-8"
    html = req.text
    soup = BeautifulSoup(req.text, features="html.parser")
    search_data = soup.find_all(class_type, class_=key_str)
    for i in range(0, len(search_data)):
        search_data[i] = search_data[i].find("a").get("href)
        version_list.append(search_data[i].split("cp")[-1].split("-")[0])

    python_version = sys.version
    python_version = python_version.split(".")[0] + python_version.split(".")[1]

    for i in range(0, len(search_data)):
        if python_version == version_list[i]:
            return search_data[i]

    return search_data[0]

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]
```


© 2018-2025 StarFive Technology
All rights reserved

```
    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())
```

c. (Optional) Exit the Python3 Virtual Environment.

```
deactivate
```


© 2018-2025 StarFive Technology
All rights reserved

3. Running Demo Code

To run the demo code, perform the following on VisionFive 2 Debian:

1. Locate to the directory where the test code, `I2C_Sense_Hat.py`, exists:

   a. Source into the Python3 Virtual Environment:

   ```
   source ./myvenv/bin/activate
   ```

   b. Execute the following command to install dependency:

   ```
   python3 -m pip install pillow
   ```

   c. Execute the following command to get the directory where `VisionFive.gpio` exists:

   ```
   python3 -m pip show VisionFive.gpio
   ```

   Result:

   ```
   Location: /home/user/myvenv/lib/python3.11/site-packages
   ```

   Note:

   The actual output depends on how the application is installed.

   d. Execute the following to enter the directory, for example, `/home/user/myvenv/lib/python3.11/site-packages` as indicated in the previous step output:

   ```
   cd /home/user/myvenv/lib/python3.11/site-packages
   ```

   e. Execute the following command to enter the `sample-code` directory:

   ```
   cd ./VisionFive/sample-code/
   ```

2. Under the `sample-code` directory, execute the following command to run the demo code:

   ```
   sudo python I2C_Sense_Hat.py
   ```

   Alternatively, you can execute the following command:

   ```
   sudo python3 I2C_Sense_Hat.py
   ```

   Result:

   The temperature and the humidity data are displayed on the terminal:

   ```
   [riscv@fedora-starfive sample-code]$ sudo python3 I2C_Sense_Hat.py
   i2c_dev: /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 %
   ```

3. (Optional) Exit the Python3 Virtual Environment.

   ```
   deactivate
   ```


© 2018-2025 StarFive Technology
All rights reserved

4. Demo Source Code

The Python source code of this demo is provided for reference purposes only.

`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)___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
import VisionFive.boardtype as board_t

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

##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)
```


© 2018-2025 StarFive Technology
All rights reserved

```
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():
    #Determining cpu Type: 1 means visionfive1; 2 means visionfive 2
    vf_t = board_t.boardtype()
    if vf_t == 1:
        I2C_DEVICE = "/dev/i2c-1"
    elif vf_t == 2:
        I2C_DEVICE = "/dev/i2c-0"
    else:
        print('This module can only be run on a VisionFive board!')
        return 0

    #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
```


© 2018-2025 StarFive Technology
All rights reserved

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


© 2018-2025 StarFive Technology
All rights reserved

5. Resources

Click on this tab to find all SBC relevant resources.

StarFive provides the following resources to guide you through an extraordinary experience on using the VisionFive 2 SBC.

- RVspace Wiki
- Application Center
- Documentation Center
- Technical Forum
- VisionFive 2 GitHub Repository
- VisionFive 2 Debian OS Download
- Code download
- View All PDF Documents

© 2018-2025 StarFive Technology
All rights reserved

6. Buy Now

Click on this tab to find all the online shops and compatible accessories.

Buy SBC

Use the following page to find your nearest sales channel or the global channels for purchasing a VisionFive 2 Single Board Computer (SBC).

- Buy VisionFive 2

Buy Parts

Use the following page to find the parts that are tested as compatible with VisionFive 2.

- Buy Accessory

© 2018-2025 StarFive Technology
All rights reserved