StarFive 赛昉科技

Using VisionFive IIC to Read SHTC3 Data

Application Note

Version: 1.1

Date: 2022/01/12

Doc ID: VisionFive-ANEN-002-1.1

Legal Statements

Important legal notice before reading our documentation.

PROPRIETARY NOTICE

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 Technology 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.

Contact Us

Address: Room 502, 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's IIC to read SHTC3 data through an example program.

Revision History

Table 0-1 Revision History

Version	Released	Revision
V1.0	2021-12-15	Preliminary release.
V1.1	2022-01-12	• Added description for `<User_Name>`. • Updated the demo file name as `test-shtc3.c`.

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

List of Tables

List of Figures

1. Introduction

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

2. Preparation

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

2.1. Preparing Hardware

Prepare the following hardware items before running the demo code:

Table 2-1 Hardware Preparation

Type	M/O*	Item	Notes
General	M	StarFive single board computer	The following boards are applicable: • StarLight • VisionFive
General	M	• 16 GB (or more) micro-SD card • micro-SD card reader • Computer (Windows/MAC/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 Fedora OS into a micro-SD card.
I2C Demo	M	• Sense Hat (B) • Dupont Line	-

Note:
*: M: Mandatory, O: Optional

2.1.1. 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
SDA	3	GPIO48 (I2C SDA)
SCL	5	GPIO47 (I2C SCL)

Figure 2-1 Connect Sense Hat (B) to the 40-Pin Header

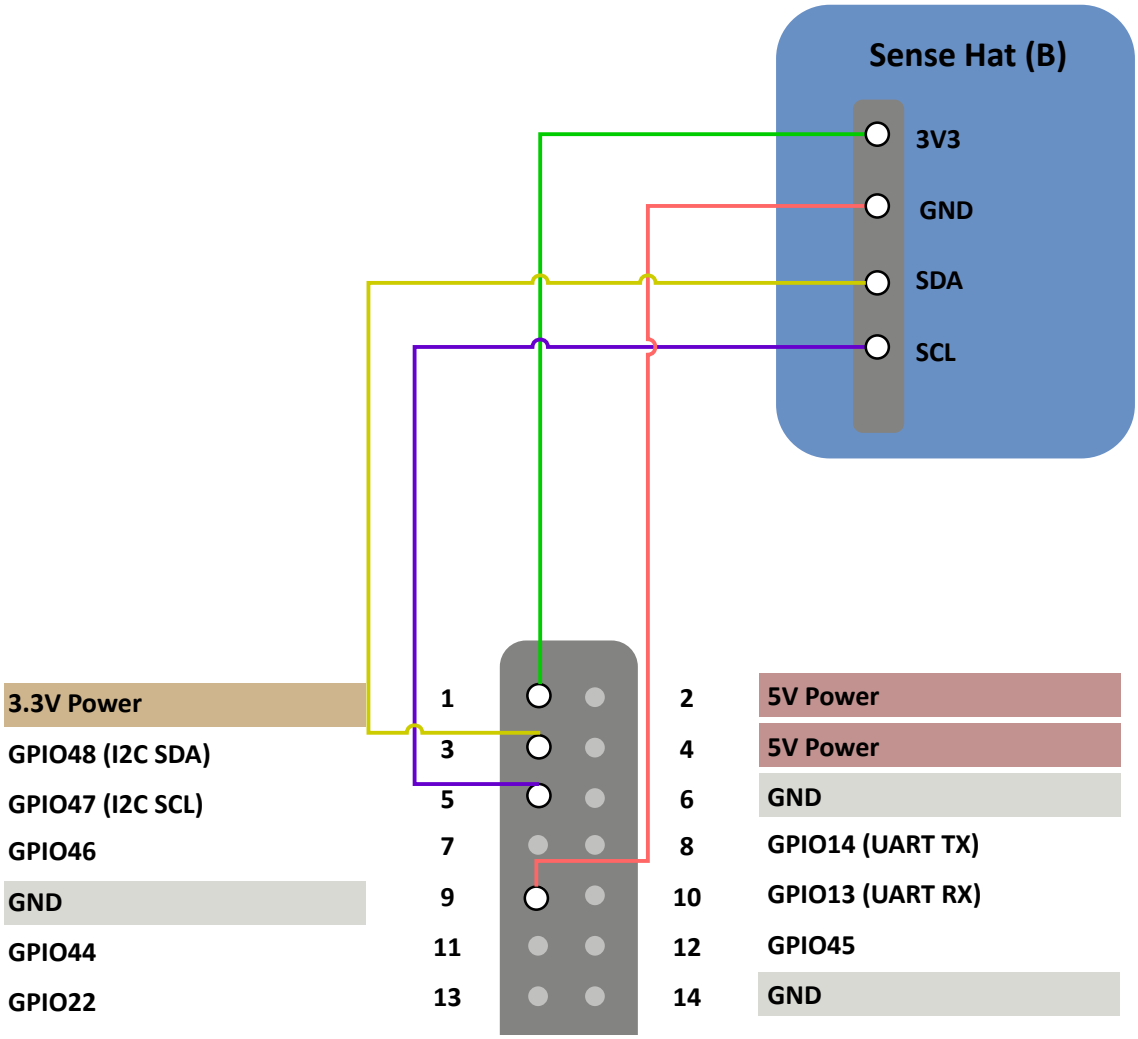
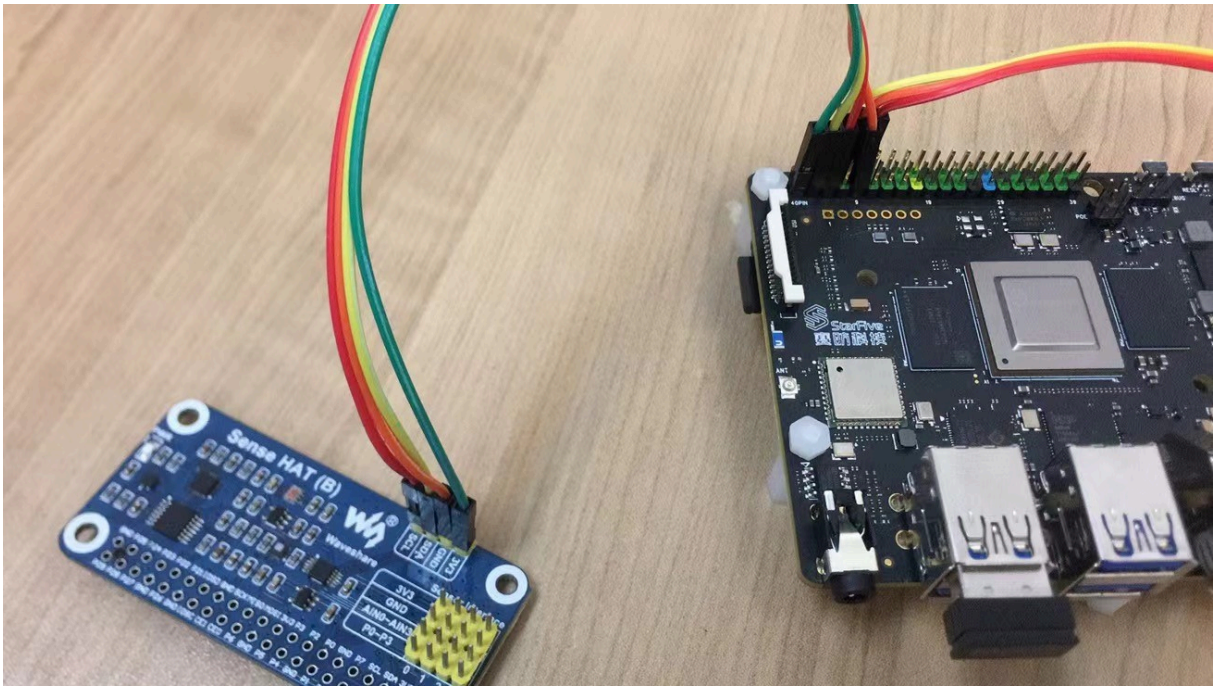


Figure 2-2 Connect Sense Hat (B) to the 40-Pin Header



2.2. Preparing Software

Make sure the following procedures are performed:

- Flash Fedora OS into a Micro-SD card and compile and replace dtb files as described in the *Preparing Software* section in *StarFive 40-Pin GPIO Header User Guide*.
- Configure the dts file as described in the *Configuring dts File* section in the *StarFive 40-Pin GPIO Header User Guide*.

3. Running Demo Code

To run the demo code, perform the following:

1. Copy the following source code as `test-shtc3.c` to your desired path under Ubuntu:

```
#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 SHTC_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. (Optional) Install the tool to compile. The following is an example to install:

```
sudo apt-get install gcc-riscv64-linux-gnu
```

Note:

- This step can be skipped if the tool has been installed.
- After successful installation, check the version by running: `linus@starfive$ riscv64-linux-gnu-gcc -v`. The following is the example output:

Figure 3-1 Example Output

3. Execute the following to compile:

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

Result:

- The output file is `test-shtc3` in the same directory.
- `UCB RISC-V` in the following output indicates that executable codes to run on RISC-V platform are successfully generated:

```
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. Execute the following in the Ubuntu environment to upload the executable code from the `test-shtc3` file to the board through the Ethernet:

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

Information:

- `<User_Name>`: Your user name of the board. For example, riscv.
- `<Board_IP_Address>`: The board IP address. For example, 192.168.92.133.

Example:

```
rsync ./test-shtc3 riscv@192.168.92.133:/home/riscv
```

5. Execute the following command on VisionFive to run the codes:
   `./test-shtc3`Copy

   Now we have successfully read the temperature and humidity data.

   Result:

   The following output indicates the execution is successful:

```
[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%
```