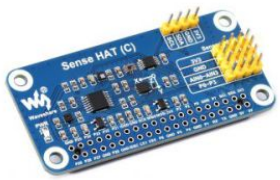
Overview

If you want to DIY the Raspberry Pi into a robot that can detect motion posture and orientation, or if you want the Raspberry Pi to collect data such as temperature, humidity, and atmospheric pressure in the environment, this HAT can just meet the above requirements.

Sense HAT (C)



I2C, AD

Features

- Standard Raspberry Pi 40PIN GPIO extension header, supports Raspberry Pi series boards
- Onboard QMI8658C+AK09918 (3-axis accelerometer, 3-axis gyroscope, and 3-axis magnetometer), detects movement, orientation, and magnetic
- Onboard SHTC3 digital temperature and humidity sensor, allows monitoring of the environment
- Onboard LPS22HB barometric pressure sensor, allows monitoring of the environment
- Onboard TCS34725 color sensor, identifies the color of a nearby object
- Onboard SGM58031, 4-ch 16-bit precision ADC, AD expansion to support more external sensors
- Breakout I2C control pins, for connecting other host boards like STM32 and Arduino
- Comes with development resources and manual (examples for Raspberry Pi/STM32/Arduino)

Parameters

Product Parameters	
Operating Voltage:	3.3V
Communication Interface:	I2C
Logic Voltage:	3.3V
Product Size:	65 x 30.5(mm)
Accelerometer Features:	Resolution: 16-bit
	Range (optional): ±2, ±4, ±8, ±16g
Gyrometer Features:	Resolution: 16-bit
	Range (optional): ±16, ±32, ±64, ±128, ±256, ±512, ±1024, ±2048°/sec
Magnetometer Features:	Resolution: 16-bit
	Range: ±4912μT
Barometer Characteristics:	Resolution: 24-bit pressure data, 16-bit temperature data;
	Measurement Accuracy (at room temperature): ±0.025hPa
	Measuring Range: 260 ~ 1260 hPa
	Measurement Rate: 1 Hz - 75 Hz
Temperature And Humidity Sensor Characteristics:	Measurement Accuracy (humidity): ±2% rH

	Measuring Range (humidity): 0% ~ 100% rH
	Measurement Accuracy (temperature): ±0.2°C
	Measuring Range (humidity): -30 ~ 100°C
Color Recognition Sensor	Resolution: 4 channel RGBC, 16 bits per channel
AD Conversion Chip	Resolution: 16 bits

Comparision

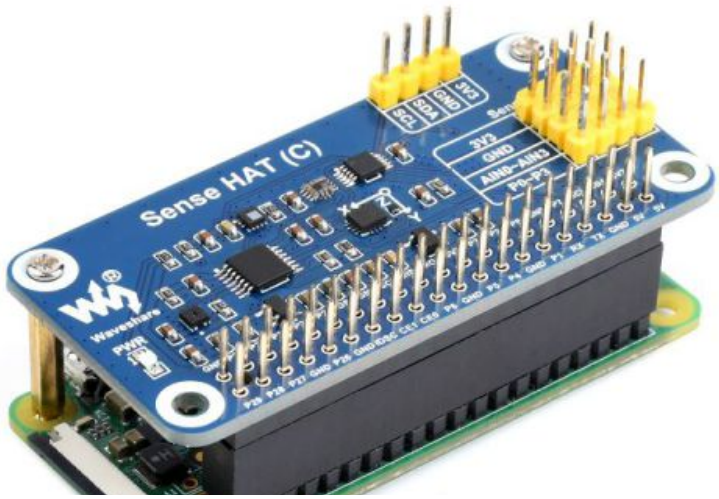
PK Item	Sense HAT (C)	Sense HAT (B)	Note
Gyroscope	Measurement range: ±16/32/64/128/256/512/1024/2048 dps Resolution: 16-bit	Measurement range: ±250/500/1000/2000 dps Resolution: 16 bits	C-type angular velocity range is more
Accelerometer	Measurement range: ±2/4/8/16 g Resolution: 16 bits	Measurement range: ±2/4/8/16 g Resolution: 16 bits	
Magnetometer	Measurement range: ±±49.12 gauss Resolution: 16-bit	Measurement range:±49 gauss Resolution: 16-bit	C-type magnetic measurement range is wider
Barometer	Measurement range: 260 ~ 1260 hPa Measurement accuracy (at room temperature): ±0.025hPa Measurement rate: 1 Hz - 75 Hz	Measuring range: 260 ~ 1260 hPa Measurement accuracy (room temperature): ±0.025hPa Measurement rate: 1 Hz - 75 Hz	
Temperature and Humidity Sensor	Measurement accuracy (humidity): ±2% rH Measurement range (humidity): 0% ~ 100% rH Measurement accuracy (temperature): ±0.2°C Measurement range (temperature): -30 ~ 100°C	Measurement accuracy (humidity): ±2% rH Measurement range (humidity): 0% ~ 100% rH Measurement accuracy (temperature): ±0.2°C Measurement range (temperature):-30 ~ 100°C	
Others	Color sensor High-precision 16-bit AD conversion chip	Color sensor High-precision 16-bit AD conversion chip	

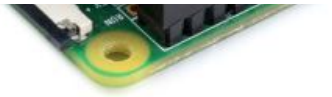
User Guides for Raspberry Pi

Hardware Connection

Hardware connection as shown:

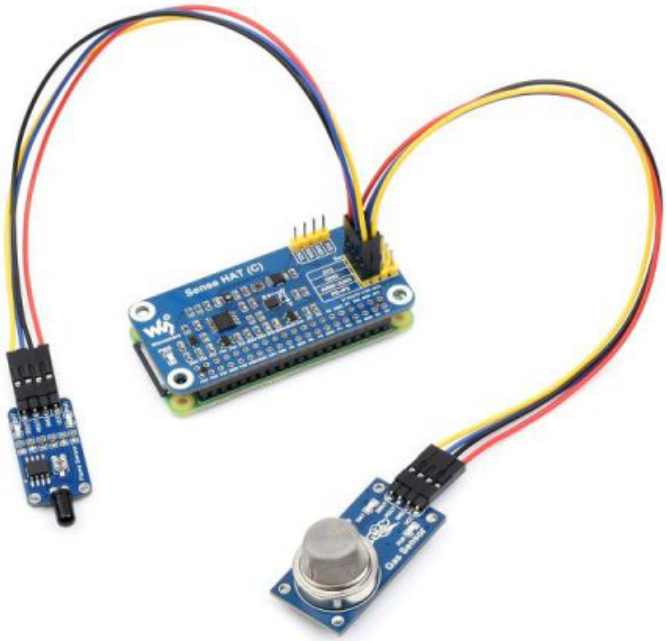
Connecting with Raspberry Pi Zero





Connect to Raspberry Pi 4 (required to add a 2x20PIN female header)

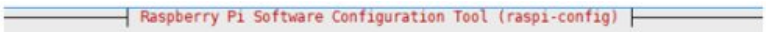
Connect to Raspberry Pi Zero

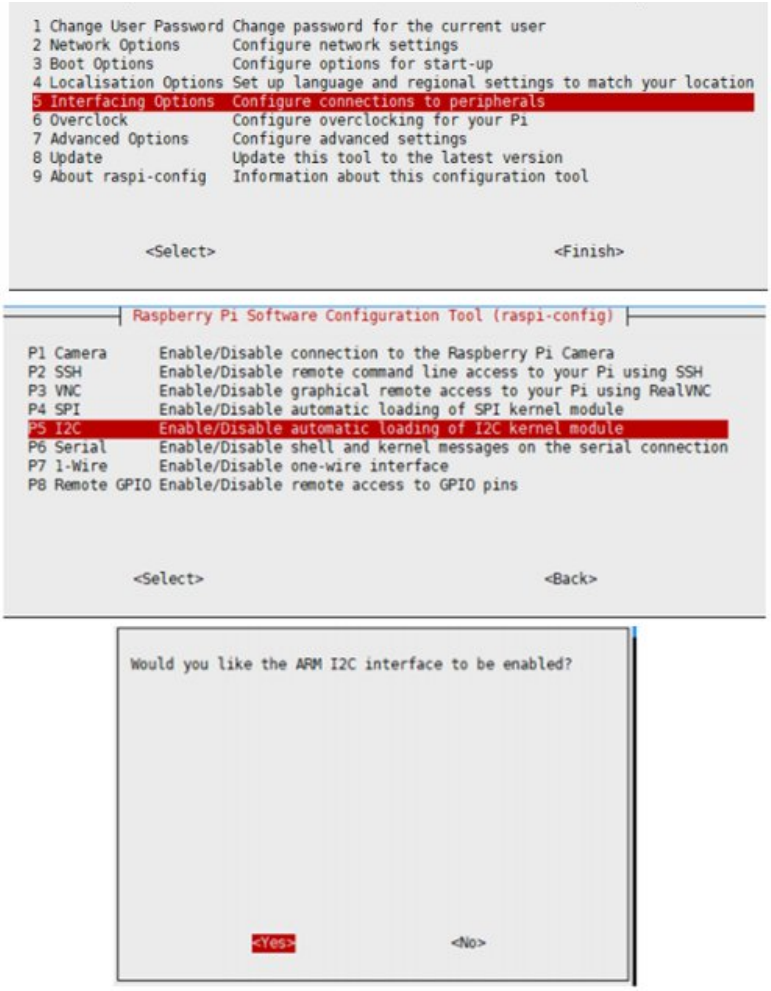


Open I2C Interface

- Open the Raspberry Pi terminal and enter the following command to enter the configuration interface.

```
sudo raspi-config
Choose Interfacing Options -> I2C ->yes start i2C kernel driver
```





and then reboot Raspberry Pi

```
sudo reboot
```

Install Function

BCM2835

```
#Open the Raspberry Pi terminal and run the following command
wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.71.tar.gz
tar zxvf bcm2835-1.71.tar.gz
cd bcm2835-1.71/
sudo ./configure && sudo make && sudo make check && sudo make install
# For more information, please refer to the official website: http://www.airspay
ce.com/mikem/bcm2835/
```

wiring Pi

```
 #Open Raspberry Pi terminal and execute
 sudo apt-get install wiringpi
 #For Raspberry Pi systems after May 2019 (earlier than before, it is not necess
ary to execute), an upgrade may be required
 wget https://project-downloads.drogon.net/wiringpi-latest.deb
 sudo dpkg -i wiringpi-latest.deb
 gpio -v
 # Run gpio -v and version 2.52 will appear. If it does not appear, the installa
tion is wrong.
```

```
#Bullseye Branch systems use the following commands:
git clone https://github.com/WiringPi/WiringPi
cd WiringPi
./build
gpio -v
# Run gpio -v and version 2.70 will appear. If it does not appear, the installa
tion is wrong.
```

Python

```
sudo apt-get update
#python2
sudo apt-get install python-pip
sudo pip install RPi.GPIO
sudo pip install spidev
sudo apt-get install python-smbus
#python3
sudo apt-get install python-pip3
sudo pip3 install RPi.GPIO
sudo pip3 install spidev
sudo apt-get install python3-smbus
```

Download Demo

```
wget https://files.waveshare.com/upload/0/04/Sense_HAT_C_Pi.zip
unzip Sense_HAT_C_Pi.zip -d Sense_HAT_C_Pi
cd Sense_HAT_C_Pi/RaspberryPi
```

I2C device address

```
ADS1015: AD conversion demos (STM32, BCM2835, WringPi and Python) Device addres
s: 0x48
```

```
QMI8658C: 6-axis sensor demos (STM32, BCM2835, WringPi and Python) Device addres
s: 0x6B
```

```
AK09918C: 3-axis sensor demos (STM32, BCM2835, WringPi and Python) Device addres
s: 0x0C
```

```
LPS22HB: Air pressure sensor demos (STM32, BCM2835, WringPi and Python) Device a
ddress: 0x5C
```

```
SHTC3: Temperature and humidity sensor demos (STM32, BCM2835, WringPi and Pytho
n) Device address: 0x70
```

```
TCS34087: Color recognition sensor demos (STM32, BCM2835, WringPi and Python) De
vice address: 0x29
```

Note: The I2C addresses of all the sensors of this module are different, and the user can use all the sensors on the board at the same time.

IMU(QMI8658C+AK09918C)

bcm2835

Enter the terminal of the Linux and execute the following command:

```
cd /home/pi/Sense_HAT_C_Pi/RaspberryPi/IMU/bcm2835/
#pi for account
sudo make clean
sudo make
sudo ./main
```

Expected result:

```
/------------------------------------------------------------/
Roll: 6.76     Pitch: 3.16     Yaw: 9.92
Acceleration: X: -937     Y: 2038     Z: 16088
Gyroscope: X: -56     Y: 44     Z: -18
Magnetic: X: -160     Y: 6     Z: -205
QMITemp:31.60 C
```

Press Ctrl+C to end the program.

WiringPi Program

Enter the terminal of the Linux and execute the following command:

```
cd /home/pi/Sense_HAT_C_Pi/RaspberryPi/IMU/wiringPi/
#pi for account
sudo make clean
sudo make
sudo ./main
```

Expected result:

```
/------------------------------------------------------------/
Roll: 6.76     Pitch: 3.16     Yaw: 9.92
Acceleration: X: -937     Y: 2038     Z: 16088
Gyroscope: X: -56     Y: 44     Z: -18
Magnetic: X: -160     Y: 6     Z: -205
QMITemp:31.60 C
```

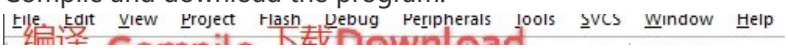
Press Ctrl+C to end the program.

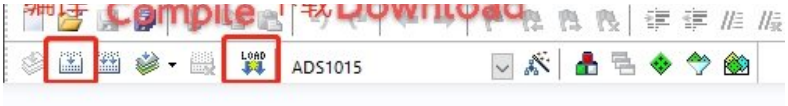
STM32

This demo is based on XNUCLEO-F103RB and exports the data by serial port 2. The wiring as shown below:

Sense HAT (C)	STM32
3V3	+3.3V
GND	GND
SDA	PB9
SCL	PB8

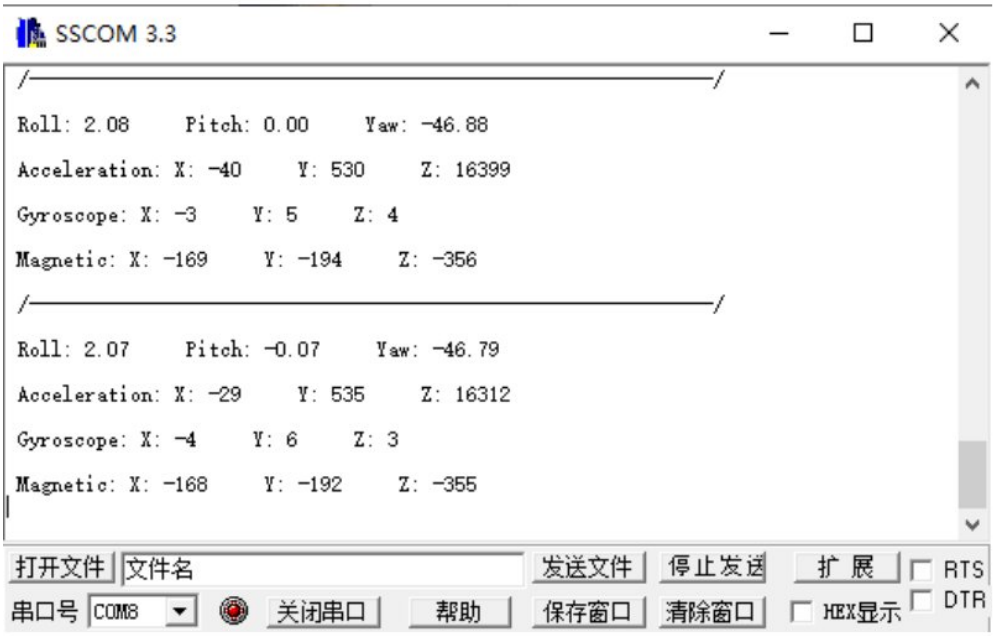
Compile and download the program:





Open the serial port assistant and set the baud rate to 115200.

Expected result:



Parameter Calibration And Calculation

Calculate Acceleration

The unit of acceleration measured by the program is LSB (least significant bit), and the unit is often converted to gravitational acceleration (g) in actual use. The default range is 16384 LSB/g (±2g), so the actual acceleration measured is:

a=Acceleration/16384 ,$Unit$:g

Please refer to

QMI8658C_datasheet P17

Gyroscope Angular Velocity

The unit of angular velocity measured by the program is LSB (least significant bit). In practice, the unit is often converted to angular velocity (°/sec). The default range is 64 LSB/(°/s) (±500°/s), so the actual angular velocity measured is:

ω=Gyroscope/64 ,Unit:°/s

Please refer to

QMI8658C_datasheet P18

LPS22HBTR

Note: The temperature detection of the air pressure sensor is only used for compensation. For accurate temperature detection, please observe the value of the SHTC3 temperature and humidity sensor.

BCM2835

Enter the Linux terminal and run the following command:

```
cd /home/pi/Sense_HAT_C_Pi/RaspberryPi/LPS22HBTR/bcm2835/
#pi is the account
sudo make clean
sudo make
sudo ./main
```

Expected result:

```
Pressure Sensor Test Program ...

Pressure Sensor OK
Pressure = 997.15 hPa , Temperature =  31.86 °C
Pressure = 997.15 hPa , Temperature =  31.86 °C
Pressure = 997.17 hPa , Temperature =  31.89 °C
Pressure = 997.17 hPa , Temperature =  31.89 °C
Pressure = 997.11 hPa , Temperature =  31.90 °C
Pressure = 997.11 hPa , Temperature =  31.90 °C
Pressure = 997.13 hPa , Temperature =  31.92 °C
Pressure = 997.13 hPa , Temperature =  31.92 °C
Pressure = 997.10 hPa , Temperature =  31.94 °C
Pressure = 997.10 hPa , Temperature =  31.94 °C
Pressure = 997.11 hPa , Temperature =  31.94 °C
```

Press Ctrl+C to end the terminal.

WiringPi Program

Enter the Linux terminal, and run the following command:

```
cd /home/pi/Sense_HAT_C_Pi/RaspberryPi/LPS22HBTR/wiringPi/
#pi is the account
sudo make clean
sudo make
sudo ./main
```

Expected result:

```
Pressure Sensor Test Program ...

Pressure Sensor OK
Pressure = 997.15 hPa , Temperature =  31.86 °C
Pressure = 997.15 hPa , Temperature =  31.86 °C
Pressure = 997.17 hPa , Temperature =  31.89 °C
Pressure = 997.17 hPa , Temperature =  31.89 °C
Pressure = 997.11 hPa , Temperature =  31.90 °C
Pressure = 997.11 hPa , Temperature =  31.90 °C
Pressure = 997.13 hPa , Temperature =  31.92 °C
Pressure = 997.13 hPa , Temperature =  31.92 °C
Pressure = 997.10 hPa , Temperature =  31.94 °C
Pressure = 997.10 hPa , Temperature =  31.94 °C
Pressure = 997.11 hPa , Temperature =  31.94 °C
```

Press Ctrl+C to end the terminal.

Python Program

Enter the Linux terminal and run the following command:

```
cd /home/pi/Sense_HAT_C_Pi/RaspberryPi/LPS22HBTR/python/
#pi is the account
sudo python3 LPS22HB.py
```

Expected result:

```
Pressure Sensor Test Program ...
Pressure = 1023.31 hPa , Temperature =  37.88 °C
Pressure = 1023.31 hPa , Temperature =  37.81 °C
Pressure = 1023.26 hPa , Temperature =  37.82 °C
Pressure = 1023.30 hPa , Temperature =  37.84 °C
Pressure = 1023.30 hPa , Temperature =  37.84 °C
Pressure = 1023.24 hPa , Temperature =  37.86 °C
Pressure = 1023.30 hPa , Temperature =  37.87 °C
Pressure = 1023.32 hPa , Temperature =  37.88 °C
Pressure = 1023.25 hPa , Temperature =  37.88 °C
Pressure = 1023.28 hPa , Temperature =  37.89 °C
```

Press Ctrl+C to end the program.

STM32 Program

This demo is based on XNUCLEO-F103RB and outputs the data by serial port 2.

The wiring is shown below:

Sense HAT (C)	STM32
3V3	+3.3V
GND	GND
SDA	PB9
SCL	PB8

Compile and download the program:

Open SSCOM and set the baud rate as 115200.

Expected result:

SHTC3 Example

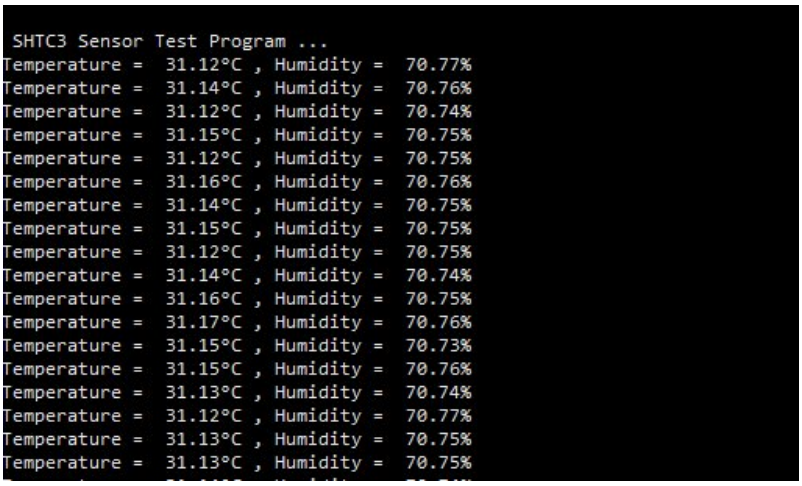
Note: The heating of the Raspberry Pi will affect the actual temperature measurement. The board has an I2C interface. If you need an accurate ambient temperature, you can separate the Raspberry Pi from the module and connect it through a wire to detect it.

bcm2835

Enter the Linux terminal and run the following command:

```
cd /home/pi/Sense_HAT_C_Pi/RaspberryPi/SHTC3/bcm2835/
#pi is the account
sudo make clean
sudo make
sudo ./main
```

Expected result:



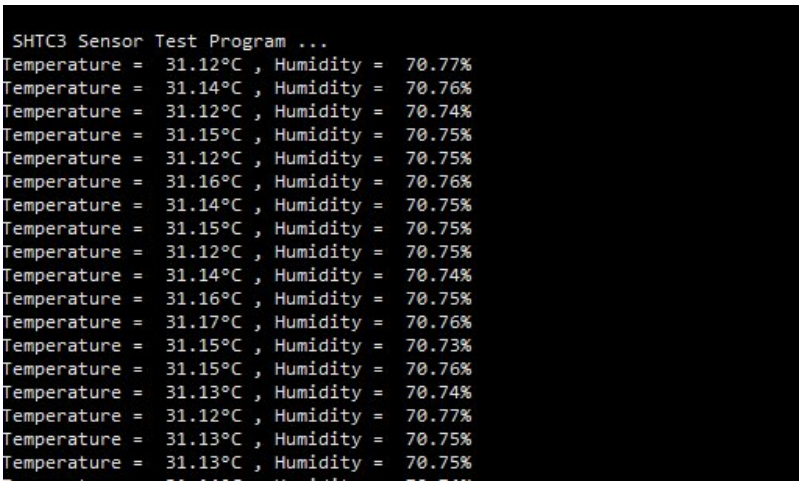
Press Ctrl+C to end the program.

WiringPi

Enter the Linux terminal and execute the following commands:

```
cd /home/pi/Sense_HAT_C_Pi/RaspberryPi/SHTC3/wiringPi/
#pi is the account
sudo make clean
sudo make
sudo ./main
```

Expected result:



Press Ctrl+C to end the program.

Python Program

Enter the Linux terminal and run the following commands.

```
cd /home/pi/Sense_HAT_C_Pi/RaspberryPi/SHTC3/python/
#pi is the account
sudo python3 SHTC3.py
```

Expected result:

```
pi@raspberrypi:~/Sense-HAT-B-Demo/SHTC3/Raspberry Pi/python $ sudo python3 SHTC3.py
Temperature =  30.42°C , Humidity =  43.06%
Temperature =  35.30°C , Humidity =  43.45%
Temperature =  35.34°C , Humidity =  43.79%
Temperature =  35.34°C , Humidity =  44.13%
Temperature =  35.35°C , Humidity =  44.49%
Temperature =  35.36°C , Humidity =  44.82%
Temperature =  35.36°C , Humidity =  45.14%
Temperature =  35.34°C , Humidity =  45.46%
Temperature =  35.34°C , Humidity =  45.80%
Temperature =  35.37°C , Humidity =  46.10%
Temperature =  35.36°C , Humidity =  46.42%
Temperature =  35.33°C , Humidity =  46.73%
Temperature =  35.41°C , Humidity =  47.03%
```

Press Ctrl + C to end the program.

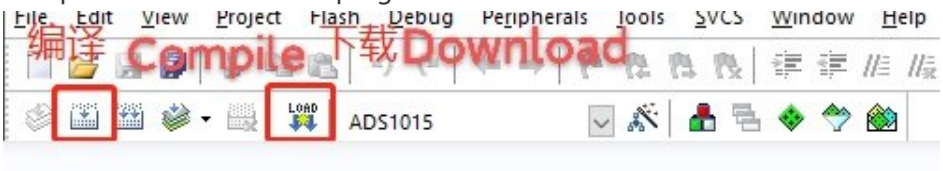
STM32 Program

This demo is based on XNUCLEO-F103RB and outputs the data by serial port 2.

The wiring is shown below:

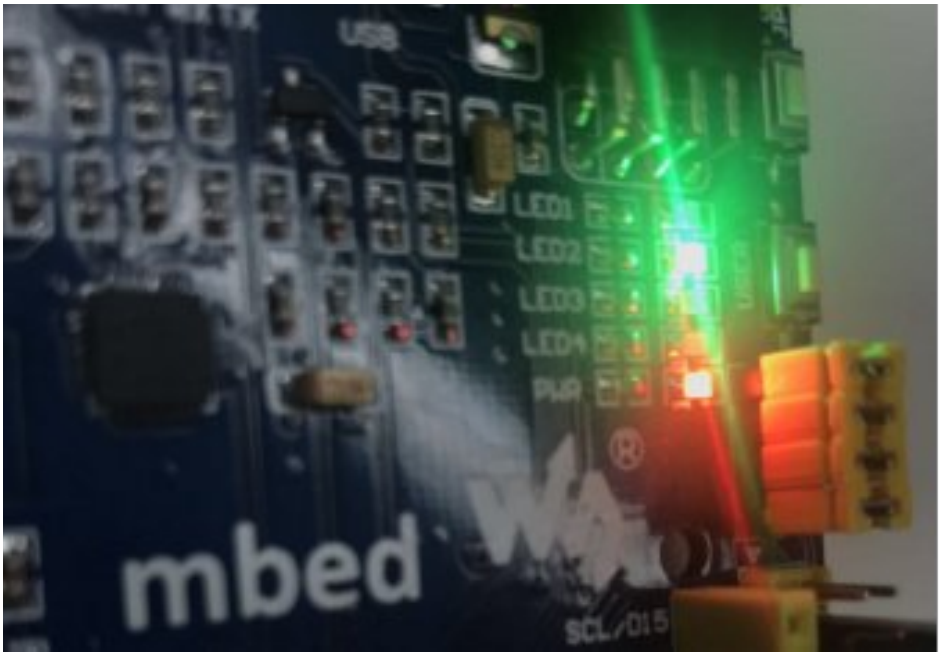
Sense HAT (C)	STM32
3V3	+3.3V
GND	GND
SDA	PB9
SCL	PB8

Compile and download the program:

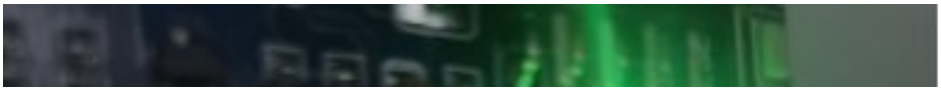


Expected result:

No error in the sensor, LED 2 is on:



When the humidity is less than 80%, LED 3 will not be on. When the humidity is greater than or equal to 80%, LED 3 will be on.



TCS34087 Demo

bcm2835

Enter the Linux terminal and run the following commands in the terminal:

```
cd /home/pi/Sense_HAT_C_Pi/RaspberryPi/TCS34087/bcm2835/
#pi is the account
sudo make clean
sudo make
sudo ./main
```

Expected result:

```
TCS34725 initialization success!!
 RGB888 :R=0   G=0  B=0   RGB888=0X0  RGB565=0X0  Lux_Interrupt = 0
 RGB888 :R=0   G=0  B=0   RGB888=0X0  RGB565=0X0  Lux_Interrupt = 0
 RGB888 :R=241   G=248  B=200   RGB888=0XF1F8C8  RGB565=0XF7D9  Lux_Interrupt = 0
 RGB888 :R=241   G=248  B=200   RGB888=0XF1F8C8  RGB565=0XF7D9  Lux_Interrupt = 0
 RGB888 :R=241   G=248  B=200   RGB888=0XF1F8C8  RGB565=0XF7D9  Lux_Interrupt = 0
 RGB888 :R=241   G=248  B=200   RGB888=0XF1F8C8  RGB565=0XF7D9  Lux_Interrupt = 0
 RGB888 :R=241   G=248  B=200   RGB888=0XF1F8C8  RGB565=0XF7D9  Lux_Interrupt = 0
 RGB888 :R=241   G=248  B=200   RGB888=0XF1F8C8  RGB565=0XF7D9  Lux_Interrupt = 0
 RGB888 :R=241   G=248  B=201   RGB888=0XF1F8C9  RGB565=0XF7D9  Lux_Interrupt = 0
 RGB888 :R=241   G=249  B=201   RGB888=0XF1F9C9  RGB565=0XF7D9  Lux_Interrupt = 0
 RGB888 :R=241   G=249  B=201   RGB888=0XF1F9C9  RGB565=0XF7D9  Lux_Interrupt = 0
 RGB888 :R=241   G=248  B=201   RGB888=0XF1F8C9  RGB565=0XF7D9  Lux_Interrupt = 0
 RGB888 :R=241   G=248  B=201   RGB888=0XF1F8C9  RGB565=0XF7D9  Lux_Interrupt = 0
 RGB888 :R=241   G=248  B=200   RGB888=0XF1F8C8  RGB565=0XF7D9  Lux_Interrupt = 0
 RGB888 :R=241   G=248  B=201   RGB888=0XF1F8C9  RGB565=0XF7D9  Lux_Interrupt = 0
```

Press Ctrl+C to end the program.

wiring Pi

Open the terminal of Linux, compile codes, and run the example by command:

```
cd /home/pi/Sense_HAT_C_Pi/RaspberryPi/TCS34087/wiringPi/
#pi is the account
sudo make clean
sudo make
sudo ./main
```

Expected result:

```
TCS34725 initialization success!!
```

```
RGB888 :R=0   G=0  B=0   RGB888=0X0  RGB565=0X0  Lux_Interrupt = 0
RGB888 :R=0   G=0  B=0   RGB888=0X0  RGB565=0X0  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=200   RGB888=0XF1F8C8  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=200   RGB888=0XF1F8C8  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=200   RGB888=0XF1F8C8  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=200   RGB888=0XF1F8C8  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=200   RGB888=0XF1F8C8  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=200   RGB888=0XF1F8C8  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=201   RGB888=0XF1F8C9  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=249  B=201   RGB888=0XF1F9C9  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=249  B=201   RGB888=0XF1F9C9  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=201   RGB888=0XF1F8C9  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=201   RGB888=0XF1F8C9  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=200   RGB888=0XF1F8C8  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=201   RGB888=0XF1F8C9  RGB565=0XF7D9  Lux_Interrupt = 0
```

Press Ctrl+C to end the program.

Python

Open the terminal of Linux, Compile codes and run the example by command:

```
cd /home/pi/Sense_HAT_C_Pi/RaspberryPi/TCS34087/python/
#pi is the account
sudo python main.py
```

Expected result:

```
CS34725 initialization success!!
RGB888 :R=0   G=0  B=0   RGB888=0X0  RGB565=0X0  Lux_Interrupt = 0
RGB888 :R=0   G=0  B=0   RGB888=0X0  RGB565=0X0  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=200   RGB888=0XF1F8C8  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=200   RGB888=0XF1F8C8  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=200   RGB888=0XF1F8C8  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=200   RGB888=0XF1F8C8  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=200   RGB888=0XF1F8C8  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=200   RGB888=0XF1F8C8  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=201   RGB888=0XF1F8C9  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=249  B=201   RGB888=0XF1F9C9  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=249  B=201   RGB888=0XF1F9C9  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=201   RGB888=0XF1F8C9  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=201   RGB888=0XF1F8C9  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=200   RGB888=0XF1F8C8  RGB565=0XF7D9  Lux_Interrupt = 0
RGB888 :R=241   G=248  B=201   RGB888=0XF1F8C9  RGB565=0XF7D9  Lux_Interrupt = 0
```

Press Ctrl+C to end the program.

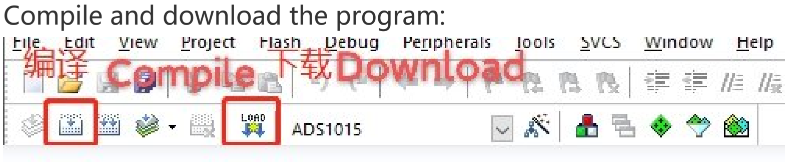
STM32

This demo is based on the XNUCLEO-F103RB development board and outputs data through serial port 2.

The connection is as follows:

Sense HAT (B)	STM32
3V3	+3.3V
GND	GND
SDA	PB9
SCL	PB8

Compile and download the program:



Open the serial port assistant and set the baud rate to 115200.

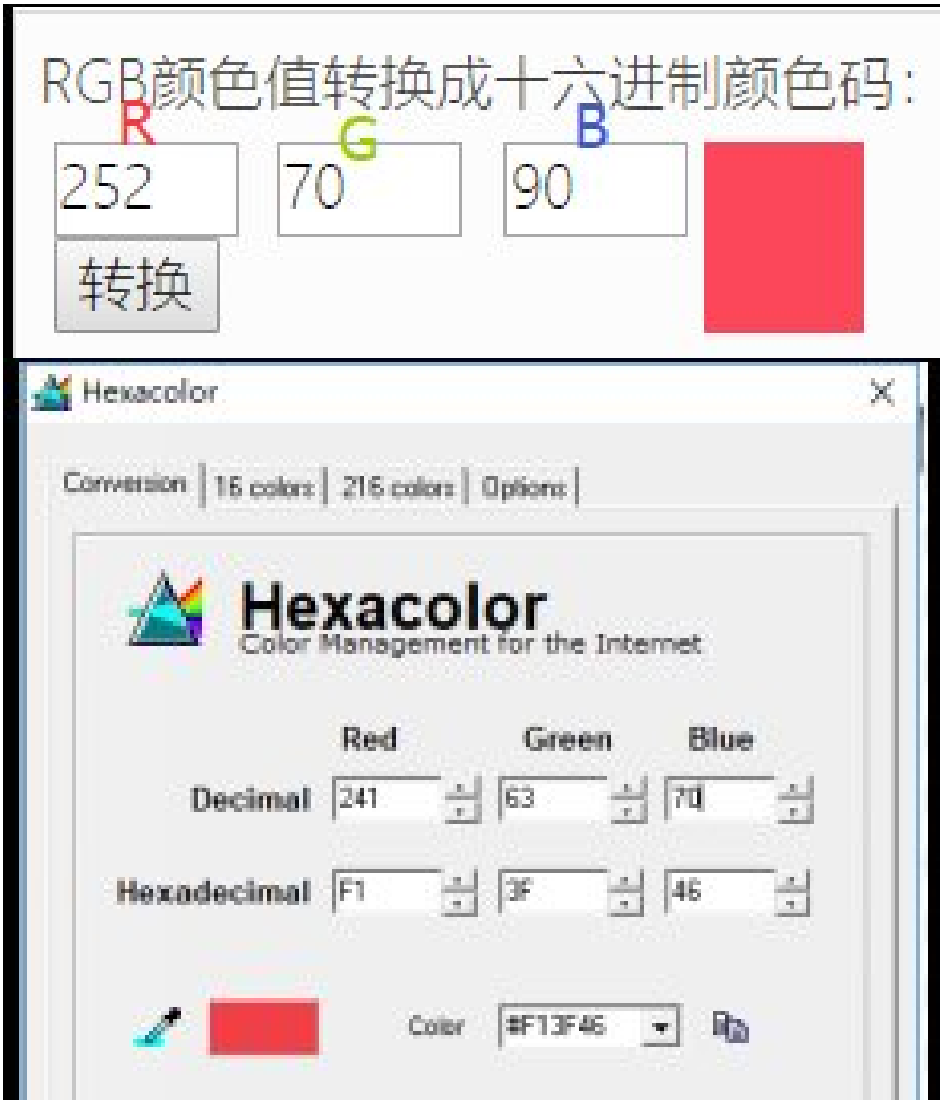
Expected outcome:

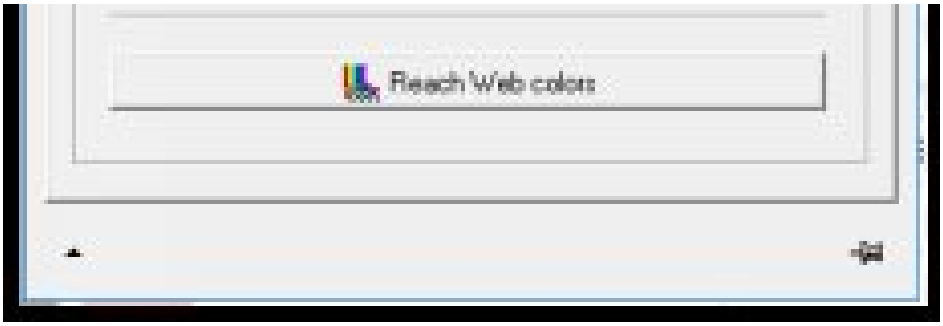
How to convert this data to color? The following introduces a tool, copy it to the browser and open it

```
https://www.sioe.cn/yingyong/yanse-rgb-16/
```

Or download:

```
https://files.waveshare.com/upload/0/05/Hexacolor3.7z
```





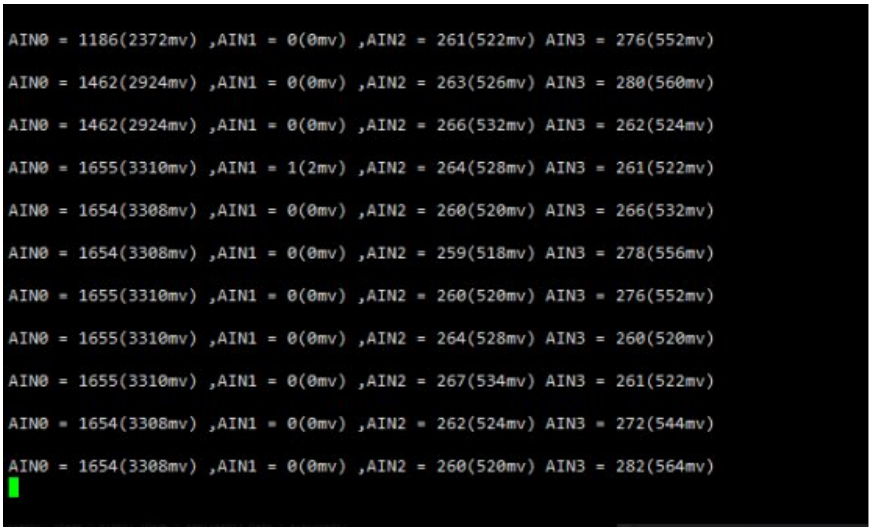
SGM58031

bcm2835

Enter the Linux terminal and run the following commands:

```
cd /home/pi/Sense_HAT_C_Pi/RaspberryPi/SGM58031/bcm2835/
#pi is the account
sudo make clean
sudo make
sudo ./main
```

Expected result:



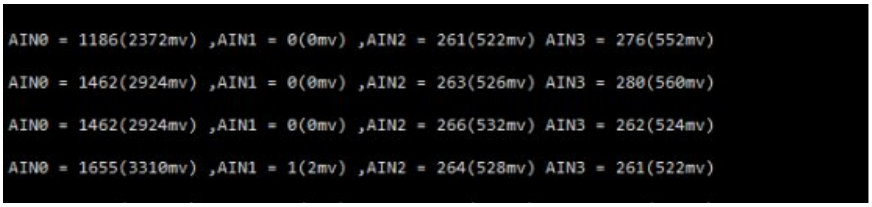
Press Ctrl+C to end the program.

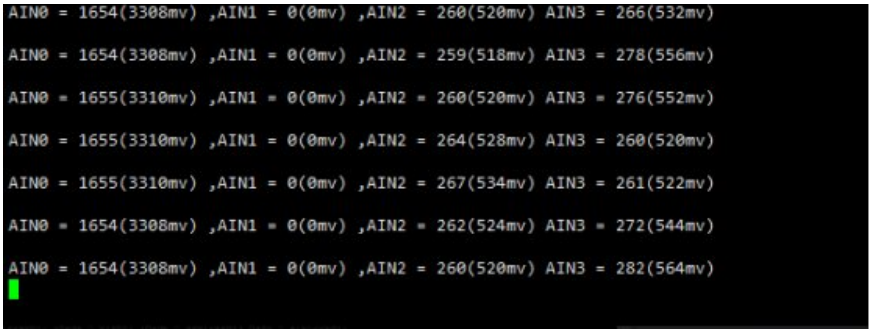
WiringPi

Enter the Linux terminal and run the following commands:

```
cd /home/pi/Sense_HAT_C_Pi/RaspberryPi/SGM58031/wiringPi/
#pi is the account
sudo make clean
sudo make
sudo ./main
```

Expected result:





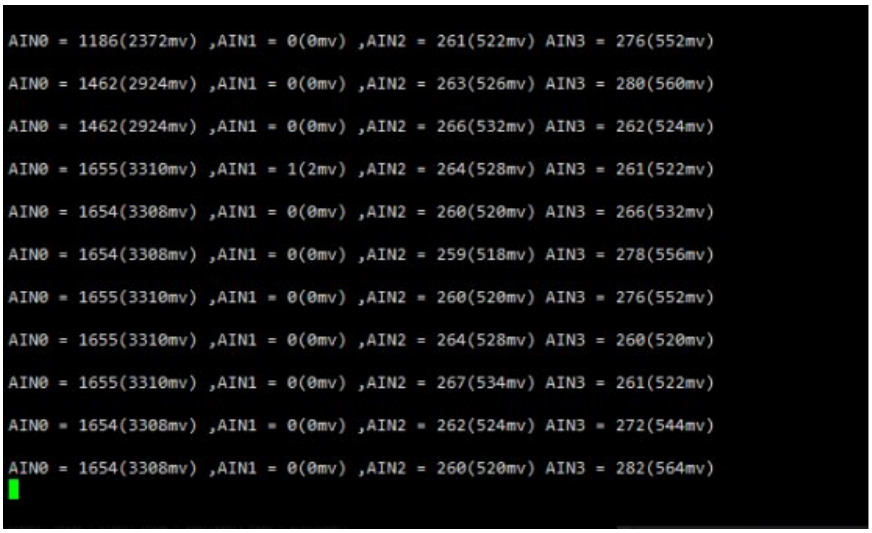
Press Ctrl+C to end the program.

python

Enter the Linux terminal and run the following commands:

```
cd /home/pi/Sense_HAT_C_Pi/RaspberryPi/SGM58031/python/
#pi is the account
sudo python AD.py
```

Expected result:



Press Ctrl+C to end the program.

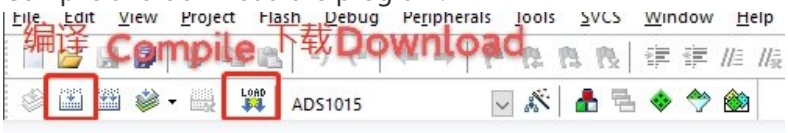
STM32

This demo is based on the XNUCLEO-F103RB development board and outputs data through serial port 2.

The connection is as follows:

Sense HAT (B)	STM32
3V3	+3.3V
GND	GND
SDA	PB9
SCL	PB8

Compile and download the program:



Open the serial port assistant and set the baud rate to 115200.

Expected outcome:

Resource

Document

- Sense_HAT_(C)_Schematic

Program

- Sense_HAT_C_Demo.7z

Related Resources

- Lps22hb Datasheet
- SHTC3 Datasheet.pdf
- TCS3408_datasheet
- QMI8658C datasheet
- AKM-AK09918C.pdf
- Lsf0204d.pdf
- SGM58031.pdf

FAQ

Question: Incorrect use of Raspberry Pi controls may cause?

Answer:
If you run python or bcm2835 and then the screen fails to refresh after running the wiringPi demo. It is because bcm2835 is the library of the RPI CPU chip, which directly operates the register, wiringPi, and python control the device by reading and writing the device files of the Linux system. So it may cause the GPIO header to work abnormally. All you need to do is to reboot the Raspberry Pi.

Question:STM32 demo serial port output has no data or data output is garbled?

Answer:
Confirm whether the baud rate is set to 115200. For the STM32 routine, please confirm that the computer is correctly connected to the development board USART2 (PA2, PA3), PA2 is TXD, and the correct COM port is selected. Control Panel -> Hardware -> Device Manager.

Support

Technical Support

If you need technical support or have any feedback/review, please click the **Submit Now** button to submit a ticket, Our support team will check and reply to you within 1 to 2 working days. Please be patient as we make every effort to help you to resolve the issue.
Working Time: 9 AM - 6 AM GMT+8 (Monday to Friday)

Submit Now