


WAVESHARE Sense HAT (B) Onboard Multi Powerful Sensors



WAVESHARE Sense HAT (B) Onboard Multi Powerful Sensors User Guide

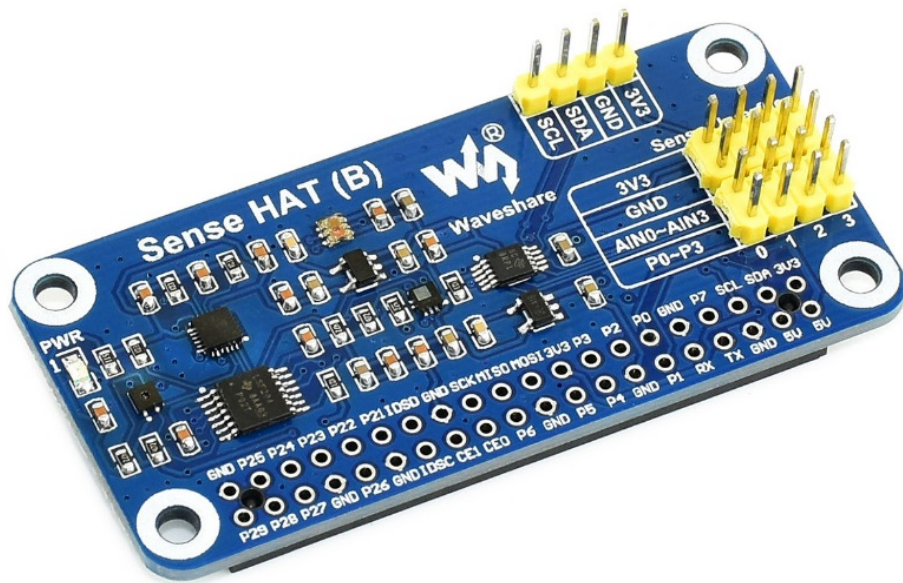
[Home](#) » [WAVESHARE](#) » WAVESHARE Sense HAT (B) Onboard Multi Powerful Sensors User Guide 

Contents

- 1 WAVESHARE Sense HAT (B) Onboard Multi Powerful Sensors
- 2 Product Usage Instructions
- 3 Introduction
- 4 Feature
- 5 Specification
- 6 Product PK
- 7 User Guides for Raspberry Pi
- 8 Install Library Functions
- 9 ICM20948 Examples—9-axis Sensor Demo
- 10 Parameter Calibration And Calculation
- 11 LPS22HBTR Example—Air Pressure Sensor Demo
- 12 SHTC3 Example—Temperature and Humidity Sensor Demo
- 13 TCS34725 Example—Color Recognition Sensor Demo
- 14 ADS1015 Example—AD Conversion
- 15 FAQ
- 16 Documents / Resources
 - 16.1 References



WAVESHARE Sense HAT (B) Onboard Multi Powerful Sensors



Specifications

- Working Voltage: 3.3V
- Interface: I2C
- Dimension: 65mm x 56.5mm
- Accelerometer: Built-in
- Gyroscope: Built-in
- Magnetometer: Built-in
- Barometer: Built-in
- Temperature & Humidity Sensor: Built-in
- Color Sensor ADC: Available

Product Usage Instructions

Hardware Connection

Connect the Sense HAT (B) to your Raspberry Pi using the appropriate pins as shown in the hardware connection diagram.

Open I2C Interface

To enable the I2C interface, access the Raspberry Pi terminal and execute the following commands:

```
sudo raspi-config Choose Interfacing Options -> I2C ->
Yes sudo reboot
```

Install Library Functions

Install necessary libraries for the Sense HAT (B) using the following commands:

- BCM2835 Library: `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`

- WiringPi Library: `sudo apt-get install`

`wiringpi`

- Python Libraries: `sudo apt-get install python-pip sudo`

`pip install RPi.GPIO spidev sudo apt-get install`

`python-smbus`

Download Demo

Download the demo files for Sense HAT (B) using the following commands:

```
sudo apt-get install p7zip-full wget
```

<https://www.waveshare.com/w/upload/6/6c/Sense-HAT-B-Demo.7z> 7z x

```
Sense-HAT-B-Demo.7z -O./Sense-HAT-B-Demo cd Sense-HAT-B-Demo
```

I2C Device Addresses

- ADS1015: Device address – 0x48
- ICM-20948: Device address – 0x68
- LPS22HB: Device address – 0x5C
- SHTC3: Device address – 0x70

Frequently Asked Questions (FAQ)

- **Q: Can I use Sense HAT (B) with other Raspberry Pi models?**

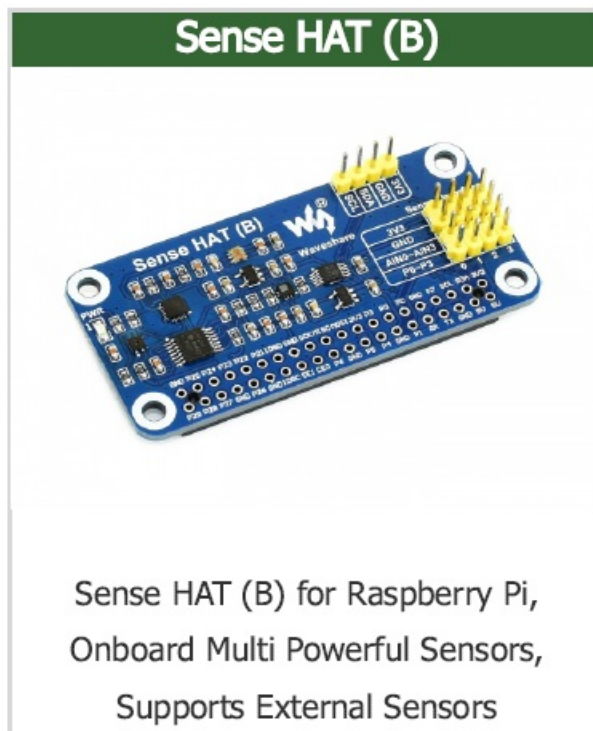
A: Yes, the Sense HAT (B) is compatible with various Raspberry Pi models but may require different hardware connections.

- **Q: What programming languages can I use to interact with Sense HAT (B)?**

A: You can use Python to interface with the sensors and utilize the capabilities of Sense HAT (B).

Introduction

It is a sensor expansion board specially designed for Raspberry Pi. It has on-board gyroscope, accelerometer, magnetometer, barometer, temperature and humidity sensor, etc. I2C interface communication supports more external sensors. 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 sensor data such as temperature, humidity and atmospheric pressure in the surrounding environment.



Feature

- Standard Raspberry Pi 40PIN GPIO extension header, supports Raspberry Pi series boards.
- Onboard ICM20948 (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 ADS1015 ADC, 4-ch 12-bit precision, AD expansion to support more external sensors.
- Brings I2C control pins, for connecting other host boards like STM32.
- Comes with development resources and manual (examples for Raspberry Pi/STM32).

Specification

Sense HAT (B) Specification

Working voltage	3.3V
Interface	I2C
Dimension	65mm x 30.5mm
Accelerometer	Resolution: 16-bit
	Ranging: $\pm 2/\pm 4/\pm 8/\pm 16g$
Gyroscope	Resolution: 16-bits
	Ranging: $\pm 250/\pm 500/\pm 1000/\pm 2000^\circ/\text{sec}$
Magnetometer	Resolution: 16-bits
	Ranging: $\pm 4900\mu T$
Barometer	Resolution: 24-bits (Pressure), 16-bits (Temperature)
	Accuracy (ordinary temperature): $\pm 0.025hPa$
	Speed: 1 Hz - 75 Hz
Temperature & Humidity	Accuracy (humidity): $\pm 2\% rH$
	Ranging (humidity): $0\% \sim 100\% rH$
	Accuracy (temperature): $\pm 0.2^\circ C$
	Ranging (temperature): $-30 \sim 100^\circ C$
Color sensor	Resolution: 4-channels RGBC, 16-bits per channel
ADC	Resolution: 12-bits

Product PK

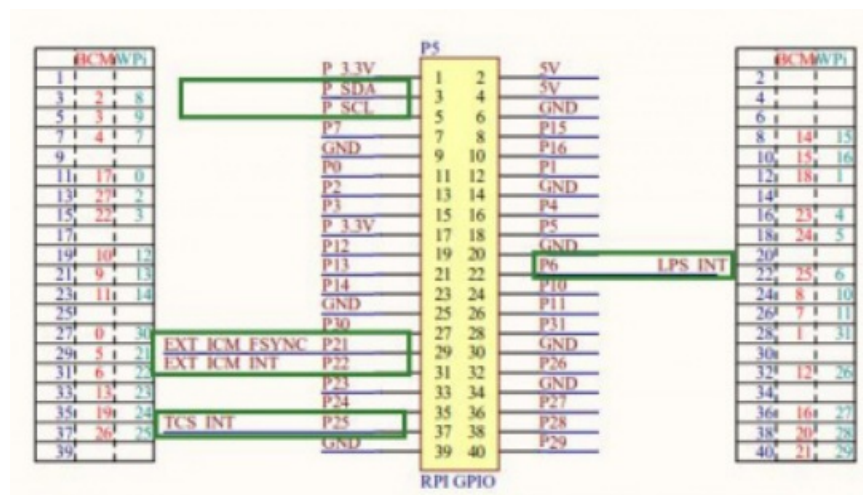
PK	Sense HAT (B)	Raspberry Pi Sense HAT	Note
Gyroscope	Measuring range: $\pm 250/500/1000/2000$ dps Resolution: 16 Bit	Measuring range: $\pm 245/500/2000$ dps Resolution: 16 Bit	More B-type angular velocity optional range
Accelerometer	Measuring range: $\pm 2/4/8/16$ g Resolution: 16 Bit	Measuring range: $\pm 2/4/8/16$ g Resolution: 16 Bit	
Magnetometer	Measuring range: ± 49 gauss Resolution: 16 Bit	Measuring range: $\pm 4/8/12/16$ gauss Resolution: 16 Bit	Wider B-type magnetic measurement range
Barometer	Measuring range: $260 \sim 1260$ hPa Measurement accuracy (Room Temperature): $\pm 0.025hPa$ Measurement rate: 1 Hz - 75 Hz	Measuring range: $260 \sim 1260$ hPa Measurement accuracy (Room Temperature): $\pm 0.1hPa$ Measurement rate: 1 Hz - 25 Hz	Type B barometric pressure measurements are more accurate and faster

Temperature and humidity sensor	Measurement accuracy (Humidity) : $\pm 2\%$ rH Measuring range (Humidity) : 80% rH 0% ~ 100% rH Measurement accuracy (Temperature) : $\pm 0.2^{\circ}\text{C}$ Measuring range (Temperature) : $-30 \sim 100^{\circ}\text{C}$	Measurement accuracy: $\pm 4.5\%$ rH Measuring range : 20% ~ Measurement accuracy (Temperature) : $\pm 0.5^{\circ}\text{C}$ Measuring range (Temperature) : $15 \sim 40^{\circ}\text{C}$	Type B temperature and humidity measurement has higher accuracy and wider range
Others	color recognition sensor High-precision 12-bit AD conversion chip	8x8 RGB LED matrix Five-way rocker	Type B onboard color sensor, support AD access to more sensors

User Guides for Raspberry Pi

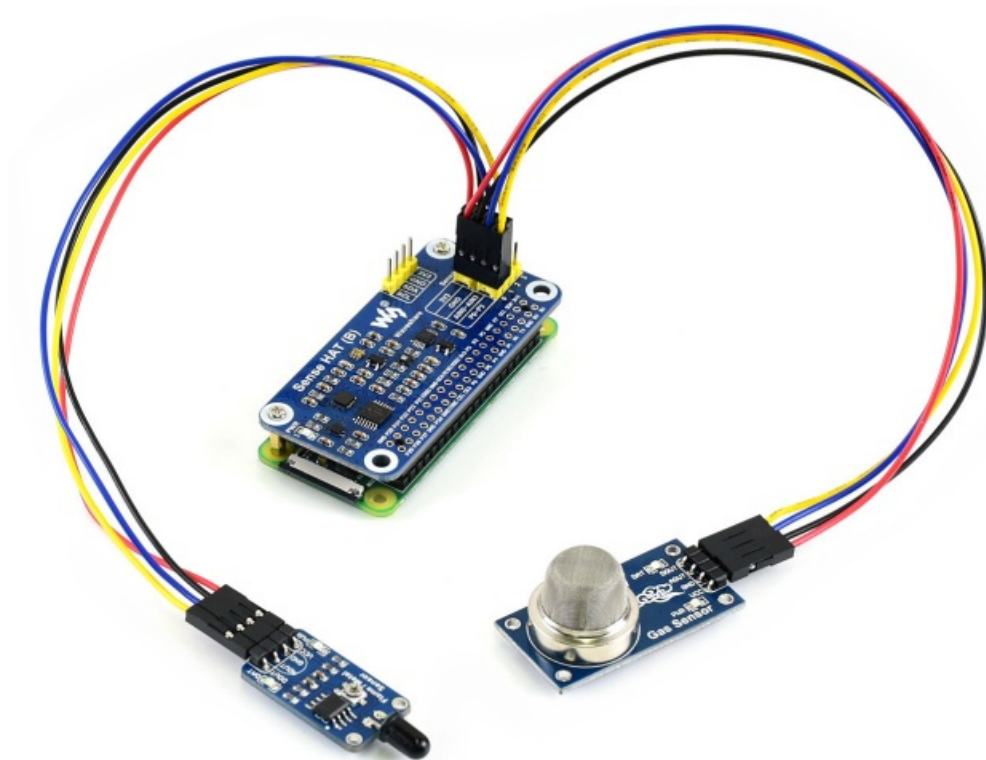
- **Hardware connection**

The pins used to connect to the Raspberry Pi are shown on the right:



- **Hardware connection diagram:**

Connect to Raspberry Pi Zero (pin header is not required)



- Connect to Raspberry Pi 4 (requires a pin header)



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 Library Functions

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.airspayce.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 necessary t

- o 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 installation 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 installation 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

- sudo apt-get install p7zip-full
- wget <https://www.waveshare.com/w/upload/6/6c/Sense-HAT-B-Demo.7z>
- 7z x Sense-HAT-B-Demo.7z -O./Sense-HAT-B-Demo

I2C device address

- ADS1015: AD conversion demo (STM32, BCM2835, WringPi and Python four demos) Device address: 0x48
- ICM-20948: 9-axis sensor demo (STM32, BCM2835, WringPi and Python four demos) Device address: 0x68
- LPS22HB: Air pressure sensor demo (STM32, BCM2835, WringPi, and Python four demos) Device address: 0x5C
- SHTC3: Temperature and humidity sensor demo (STM32, BCM2835, WringPi, and Python four demos) Device address: 0x70

- TCS34725: Color recognition sensor demo (STM32, BCM2835, WringPi and Python four demo s) Device 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.

ICM20948 Examples—9-axis Sensor Demo

bcm2835

- Open the terminal of Linux, compile codes, and run the example by command:
 - `cd ICM-20948/Raspberry\ Pi/bcm2835/`
 - `sudo make clean`
 - `sudo make`
 - `sudo ./ICM20948_DEMO`

```
/-----/
Roll: 18.22    Pitch: -10.73    Yaw: -162.27
Acceleration: X: 64    Y: 1935    Z: 3248
Gyroscope: X: -2    Y: -1    Z: -2
Magnetic: X: 3    Y: -55    Z: -22
/-----/
```

- Press Ctrl+C to end the demo.

WiringPi

- Open the terminal of Linux, run the example by command:
 - `cd ICM-20948/Raspberry\ Pi/wiringPi/`
 - `sudo make clean`
 - `sudo make`
 - `sudo ./ICM20948_DEMO`
- **Expected result:**

```
/-----/
Roll: 18.22      Pitch: -10.73      Yaw: -162.27
Acceleration: X: 64      Y: 1935      Z: 3248
Gyroscope: X: -2      Y: -1      Z: -2
Magnetic: X: 3      Y: -55      Z: -22
/-----/
```

- Press Ctrl+C to end the demo.

Python

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

- `cd ICM-20948/Raspberry\ Pi/python/`
- `sudo python3 ICM20948.py`

Expected result:

```
/-----/
Roll: 18.22      Pitch: -10.73      Yaw: -162.27
Acceleration: X: 64      Y: 1935      Z: 3248
Gyroscope: X: -2      Y: -1      Z: -2
Magnetic: X: 3      Y: -55      Z: -22
/-----/
```

Press Ctrl+C to end the demo.

VisionFive2 Demo

- Enter the Linux terminal and run the following commands in the terminal:
- `cd ICM-20948/VisionFive2`
- `python3 ICM20948.py`
- Press Ctrl+C to end the demo.

STM32 Example

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

The connection is as follows:

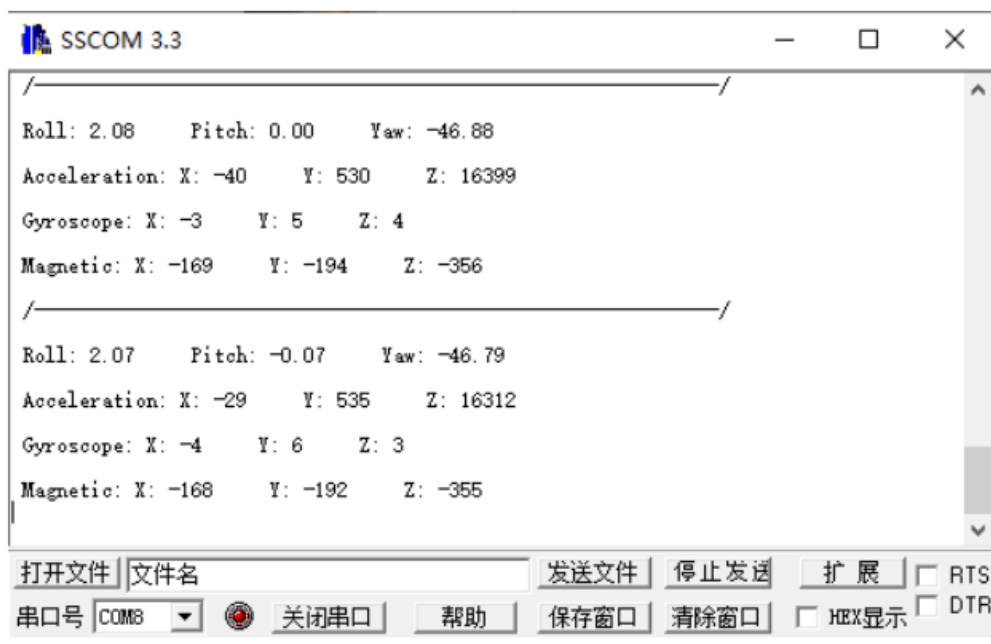
Sense HAT (B)	STM32
VCC (Note: jumper cap should jump to the corresponding position)	+5V/+3.3V
GND	GND
SDA	PB9
SCL	PB8

Compile and download the demo:



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 demo is LSB (least significant bit), and the unit is often converted to gravitational acceleration (g) in actual use. The sample demo of the module sets AFS_SEL=0 by default, and the corresponding range is 16384 LSB/g ($\pm 2g$), so the actual acceleration measured is:

$a = \text{Acceleration} / 16384$, Unit: g

Please refer to ICM-20948-v1.3.pdf P12.

Calculate 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 ($^{\circ}/\text{sec}$). The sample program of the module sets FS_SEL=2 by default, and the corresponding range is 32.8 LSB/ $^{\circ}/\text{s}$ ($\pm 1000^{\circ}/\text{s}$), so the actual angular velocity measured is: $w = \text{Gyroscope} / 32.8$, Unit: %

Please refer to ICM-20948-v1.3.pdf P11.

LPS22HBTR Example—Air Pressure Sensor Demo

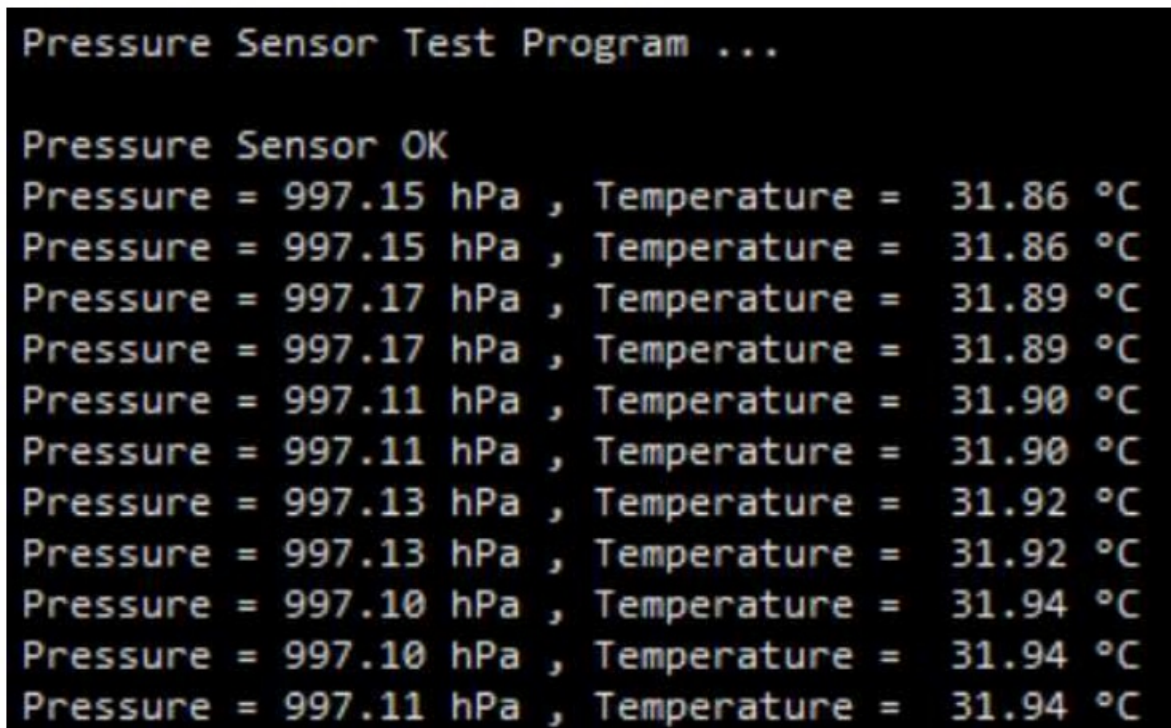
Note: The temperature detection of the air pressure sensor is only used for compensation. For accurate temperature detection, please observe the value of the SHTC3 temperature and humidity sensor.

bcm2835 Demo

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

- `cd LPS22HBTR/Raspberry\ Pi/bcm2835/`
- `sudo make clean`
- `sudo make`
- `sudo ./LPS22HB`

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 demo.

Wiring Pi demo

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

- `cd LPS22HBTR/Raspberry\ Pi/wiringPi/`
- `sudo make clean`
- `sudo make`
- `sudo ./LPS22HB`

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 demo.

python

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

- cd LPS22HBTR/Raspberry\ Pi/python/
- sudo python3 LPS22HB.py

Expected result:

```

pi@raspberrypi:~/Sense-HAT-B-Demo/LPS22HBTR/Raspberry Pi/python $ sudo python3 LPS22HB.py
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 demo.

VisionFive2 Demo

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

- cd LPS22HB/VisionFive2
- python3 LPS22HB.py
- Press Ctrl+C to end the terminal.

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
VCC (Note: jumper cap should jump to the corresponding position)	+5V/+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:



SHTC3 Example—Temperature and Humidity Sensor Demo

Note: The heating of the Raspberry Pi will affect the actual measured temperature. The board has an I2C interface. If you need an accurate ambient temperature, you can separate the Raspberry Pi from the module for a distance and connect it with a cable to test it.

BCM2835 demo

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

- `cd SHTC3/Raspberry\ Pi/bcm2835/`
- `sudo make clean`
- `sudo make`
- `sudo ./SHTC3`

Expected result:

```
SHTC3 Sensor Test Program ...
Temperature = 31.12°C , Humidity = 70.77%
Temperature = 31.14°C , Humidity = 70.76%
Temperature = 31.12°C , Humidity = 70.74%
Temperature = 31.15°C , Humidity = 70.75%
Temperature = 31.12°C , Humidity = 70.75%
Temperature = 31.16°C , Humidity = 70.76%
Temperature = 31.14°C , Humidity = 70.75%
Temperature = 31.15°C , Humidity = 70.75%
Temperature = 31.12°C , Humidity = 70.75%
Temperature = 31.14°C , Humidity = 70.74%
Temperature = 31.16°C , Humidity = 70.75%
Temperature = 31.17°C , Humidity = 70.76%
Temperature = 31.15°C , Humidity = 70.73%
Temperature = 31.15°C , Humidity = 70.76%
Temperature = 31.13°C , Humidity = 70.74%
Temperature = 31.12°C , Humidity = 70.77%
Temperature = 31.13°C , Humidity = 70.75%
Temperature = 31.13°C , Humidity = 70.75%
```

Press Ctrl+C to end the demo.

wiring Pi

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

- `cd SHTC3/Raspberry\ Pi/wiringPi/`
- `sudo make clean`
- `sudo make`
- `sudo ./SHTC3`

Expected result:

```

SHTC3 Sensor Test Program ...
Temperature = 31.12°C , Humidity = 70.77%
Temperature = 31.14°C , Humidity = 70.76%
Temperature = 31.12°C , Humidity = 70.74%
Temperature = 31.15°C , Humidity = 70.75%
Temperature = 31.12°C , Humidity = 70.75%
Temperature = 31.16°C , Humidity = 70.76%
Temperature = 31.14°C , Humidity = 70.75%
Temperature = 31.15°C , Humidity = 70.75%
Temperature = 31.12°C , Humidity = 70.75%
Temperature = 31.14°C , Humidity = 70.74%
Temperature = 31.16°C , Humidity = 70.75%
Temperature = 31.17°C , Humidity = 70.76%
Temperature = 31.15°C , Humidity = 70.73%
Temperature = 31.15°C , Humidity = 70.76%
Temperature = 31.13°C , Humidity = 70.74%
Temperature = 31.12°C , Humidity = 70.77%
Temperature = 31.13°C , Humidity = 70.75%
Temperature = 31.13°C , Humidity = 70.75%

```

Press Ctrl+C to end the demo.

python

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

- cd SHTC3/Raspberry\ Pi/python/
- 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 demo.

VisionFive Demo

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

- cd SHTC3/VisionFive2
- python3 SHTC3.py
- Press Ctrl+C to end the demo.

STM32

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

- The connection is as follows:

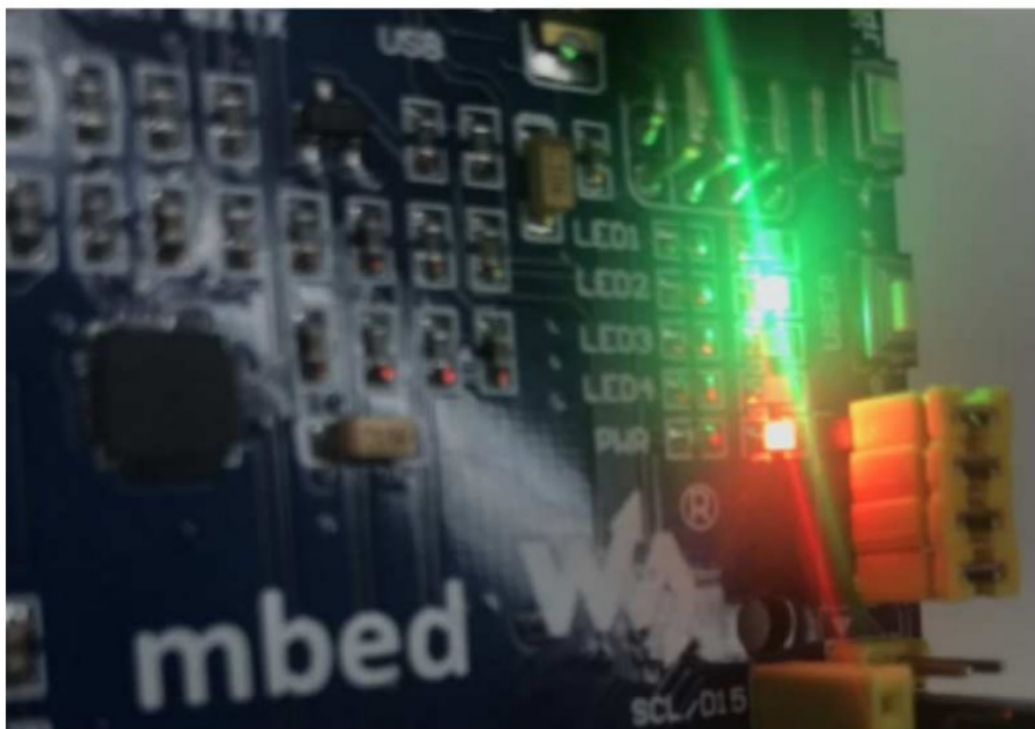
Sense HAT (B)	STM32
VCC(note:Jumper cap to jump to the corresponding position)	+5V/+3.3V
GND	GND
SDA	PB9
SCL	PB8

- Compile and download the program:



- Expected result:

The sensor is ok with no errors, LED2 is on:



- When the air humidity is lower than 80%, LED3 does not light up. LED3 lights up when the air humidity is greater than or equal to 80%.



TCS34725 Example—Color Recognition Sensor Demo

bcm2835

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

- `cd TCS34725/Raspberry\ Pi/bcm2835/`
- `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=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 demo.

wiring Pi

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

- `cd TCS34725/Raspberry\ Pi/wiringPi/`
- `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=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 demo.

Python

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

- cd TCS34725/RaspberryPi/python3/
- sudo python3 main.py

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 demo.

VisionFive2 Demo

Enter the Linux terminal, and execute the following commands in the terminal:

- **Install pygame library:**
 - apt-get install pygame
- **Run the demo:**
 - cd TCS34725/VisionFive2
 - python3 main.py
 - Press Ctrl+C to end the demo.

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
VCC (Note: jumper cap should jump to the corresponding position)	+5V/+3.3V
GND	GND
SDA	PB9
SCL	PB8

Compile and download the demo:



Open the SSCOM 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://www.waveshare.com/w/upload/0/05/Hexacolor3.7z>



ADS1015 Example—AD Conversion

bcm2835

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

- `cd ADS1015/Raspberry\ Pi/bcm2835/`
- `sudo make clean`
- `sudo make`
- `sudo ./AD`

Expected result:

```
AIN0 = 1186(2372mv) ,AIN1 = 0(0mv) ,AIN2 = 261(522mv) AIN3 = 276(552mv)
AIN0 = 1462(2924mv) ,AIN1 = 0(0mv) ,AIN2 = 263(526mv) AIN3 = 280(560mv)
AIN0 = 1462(2924mv) ,AIN1 = 0(0mv) ,AIN2 = 266(532mv) AIN3 = 262(524mv)
AIN0 = 1655(3310mv) ,AIN1 = 1(2mv) ,AIN2 = 264(528mv) AIN3 = 261(522mv)
AIN0 = 1654(3308mv) ,AIN1 = 0(0mv) ,AIN2 = 260(520mv) AIN3 = 266(532mv)
AIN0 = 1654(3308mv) ,AIN1 = 0(0mv) ,AIN2 = 259(518mv) AIN3 = 278(556mv)
AIN0 = 1655(3310mv) ,AIN1 = 0(0mv) ,AIN2 = 260(520mv) AIN3 = 276(552mv)
AIN0 = 1655(3310mv) ,AIN1 = 0(0mv) ,AIN2 = 264(528mv) AIN3 = 260(520mv)
AIN0 = 1655(3310mv) ,AIN1 = 0(0mv) ,AIN2 = 267(534mv) AIN3 = 261(522mv)
AIN0 = 1654(3308mv) ,AIN1 = 0(0mv) ,AIN2 = 262(524mv) AIN3 = 272(544mv)
AIN0 = 1654(3308mv) ,AIN1 = 0(0mv) ,AIN2 = 260(520mv) AIN3 = 282(564mv)
```


Press Ctrl+C to end the demo.

Wiring Pi

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

- `cd ADS1015/Raspberry\ Pi/wiringPi/`
- `sudo make clean`
- `sudo make`
- `sudo ./AD`

Expected result:

```
AIN0 = 1186(2372mv) ,AIN1 = 0(0mv) ,AIN2 = 261(522mv) AIN3 = 276(552mv)
AIN0 = 1462(2924mv) ,AIN1 = 0(0mv) ,AIN2 = 263(526mv) AIN3 = 280(560mv)
AIN0 = 1462(2924mv) ,AIN1 = 0(0mv) ,AIN2 = 266(532mv) AIN3 = 262(524mv)
AIN0 = 1655(3310mv) ,AIN1 = 1(2mv) ,AIN2 = 264(528mv) AIN3 = 261(522mv)
AIN0 = 1654(3308mv) ,AIN1 = 0(0mv) ,AIN2 = 260(520mv) AIN3 = 266(532mv)
AIN0 = 1654(3308mv) ,AIN1 = 0(0mv) ,AIN2 = 259(518mv) AIN3 = 278(556mv)
AIN0 = 1655(3310mv) ,AIN1 = 0(0mv) ,AIN2 = 260(520mv) AIN3 = 276(552mv)
AIN0 = 1655(3310mv) ,AIN1 = 0(0mv) ,AIN2 = 264(528mv) AIN3 = 260(520mv)
AIN0 = 1655(3310mv) ,AIN1 = 0(0mv) ,AIN2 = 267(534mv) AIN3 = 261(522mv)
AIN0 = 1654(3308mv) ,AIN1 = 0(0mv) ,AIN2 = 262(524mv) AIN3 = 272(544mv)
AIN0 = 1654(3308mv) ,AIN1 = 0(0mv) ,AIN2 = 260(520mv) AIN3 = 282(564mv)
```

Press Ctrl+C to end the demo.

Python

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

- `cd ADS1015/Raspberry\ Pi/python/`
- `sudo python3 AD.py`

Expected result:


```

AIN0 = 1186(2372mv) ,AIN1 = 0(0mv) ,AIN2 = 261(522mv) AIN3 = 276(552mv)
AIN0 = 1462(2924mv) ,AIN1 = 0(0mv) ,AIN2 = 263(526mv) AIN3 = 280(560mv)
AIN0 = 1462(2924mv) ,AIN1 = 0(0mv) ,AIN2 = 266(532mv) AIN3 = 262(524mv)
AIN0 = 1655(3310mv) ,AIN1 = 1(2mv) ,AIN2 = 264(528mv) AIN3 = 261(522mv)
AIN0 = 1654(3308mv) ,AIN1 = 0(0mv) ,AIN2 = 260(520mv) AIN3 = 266(532mv)
AIN0 = 1654(3308mv) ,AIN1 = 0(0mv) ,AIN2 = 259(518mv) AIN3 = 278(556mv)
AIN0 = 1655(3310mv) ,AIN1 = 0(0mv) ,AIN2 = 260(520mv) AIN3 = 276(552mv)
AIN0 = 1655(3310mv) ,AIN1 = 0(0mv) ,AIN2 = 264(528mv) AIN3 = 260(520mv)
AIN0 = 1655(3310mv) ,AIN1 = 0(0mv) ,AIN2 = 267(534mv) AIN3 = 261(522mv)
AIN0 = 1654(3308mv) ,AIN1 = 0(0mv) ,AIN2 = 262(524mv) AIN3 = 272(544mv)
AIN0 = 1654(3308mv) ,AIN1 = 0(0mv) ,AIN2 = 260(520mv) AIN3 = 282(564mv)

```

Press Ctrl+C to end the demo.

VisionFive2 Demo

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

- cd ADS1015/VisionFive2
- python3 AD.py
- Press Ctrl+C to end the demo.

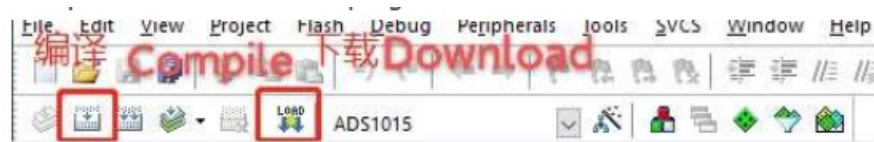
STM32

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
VCC(note:Jumper cap to jump to the corresponding position)	+5V/+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.



Resources

- **Document**

Schematic

- **Demo code**

Code

- **Datasheet**

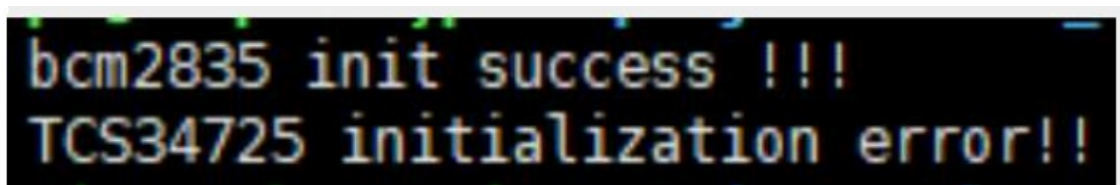
- ADS1015
- ICM-20948
- LPS22HBTR
- LSF0204
- SHTC3
- TCS34725

FAQ

- **Question:**

Why the Raspberry Pi demo failed to initialize? (The following takes TC34725 color recognition sensor as an example)

Answer:



- For the BCM2835 and wiringPi demo such prompts appear,

```
Traceback (most recent call last):
  File "main.py", line 28, in <module>
    GPIO.cleanup()
NameError: name 'GPIO' is not defined
```

- If Python demo occurs the above problem, this is a device data I2C data transfer error. Most of them are hardware connection errors, please check whether the hardware connection is correct, check whether there is any problem with the hardware connection, run `i2cdetect -y 1` If the IIC address is displayed, it means that the hardware connection is no problem. If the hardware connection is correct, it may be caused by incorrect use of Raspberry Pi control (see below for details), just restart the Raspberry Pi.

```
pi@raspberrypi:~ $ i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
10:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
20:  -- -- -- -- -- -- -- 29 -- -- -- -- -- --
30:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
40:  -- -- -- -- -- -- 48 -- -- -- -- -- -- --
50:  -- -- -- -- -- -- -- -- -- 5c -- -- -- --
60:  -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: 70 -- -- -- -- -- -- -- -- -- -- -- -- --
```

- **Question:** What will happen if using the Raspberry Pi controls incorrectly?

Answer: If running the wiringPi demo is normal, then running python or BCM2835 may cause the screen to fail to refresh normally, because the bcm2835 library is a library function of the Raspberry Pi cpu chip, and the bottom layer is to directly operate the registers, while the bottom layer of the wiringPi library and python are through reading and writing device file of the linux system to operate the device, which may cause the GPIO port to be abnormal. In this case, restarting the Raspberry Pi can solve the problem perfectly.

- **Question:** What should I do if the serial port output of the STM32 demo has no data or the 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.

WAVESHARE Sense HAT (B) Onboard Multi Powerful Sensors [pdf] User Guide
Sense HAT B Onboard Multi Powerful Sensors, Sense HAT B, Onboard Multi Powerful Sensors,
Multi Powerful Sensors, Powerful Sensors

Sense HAT B Onboard Multi Powerful Sensors, Sense HAT B, Onboard Multi Powerful Sensors, Multi Powerful Sensors, Powerful Sensors

Sense HAT B Onboard Multi Powerful Sensors, Sense HAT B, Onboard Multi Powerful Sensors, Multi Powerful Sensors, Powerful Sensors



References

- airspayce.com/mikem/bcm2835/bcm2835-1.71.tar.gz
- [GitHub - WiringPi/WiringPi: The arguably fastest GPIO Library for the Raspberry Pi](https://github.com/WiringPi/WiringPi)
- project-downloads.drogon.net/wiringpi-latest.deb
- [RGB](#)
- [Log in - Waveshare Wiki](#)
- [File:A.png - Waveshare Wiki](#)
- [File:ADS.png - Waveshare Wiki](#)
- [File:ADS2.png - Waveshare Wiki](#)
- [File:ADS3.png - Waveshare Wiki](#)
- [File:An1.png - Waveshare Wiki](#)
- [File:An2.png - Waveshare Wiki](#)
- [File:An3.png - Waveshare Wiki](#)
- [File:An4.png - Waveshare Wiki](#)
- [File:An5.png - Waveshare Wiki](#)
- [File:Com.png - Waveshare Wiki](#)
- [File:Hexa.png - Waveshare Wiki](#)
- [File:lcm2048.png - Waveshare Wiki](#)
- [File:Led2.png - Waveshare Wiki](#)
- [File:Led3.png - Waveshare Wiki](#)
- [File:Load.png - Waveshare Wiki](#)
- [File:Lps.png - Waveshare Wiki](#)
- [File:Lps2.png - Waveshare Wiki](#)
- [File:Lps3.png - Waveshare Wiki](#)
- [File:Result2.png - Waveshare Wiki](#)
- [File:Result3.png - Waveshare Wiki](#)
- [File:Rgb.png - Waveshare Wiki](#)
- [File:Rgb33.png - Waveshare Wiki](#)
- [File:Senhat.png - Waveshare Wiki](#)
- [File:Sense hat c RPI.jpg - Waveshare Wiki](#)
- [File:Sense-HAT-B-1.jpg - Waveshare Wiki](#)
- [File:Sense-HAT-B-7.jpg - Waveshare Wiki](#)

-  [File:Sense-HAT-B-9.jpg - WaveShare Wiki](#)
-  [File:Sensor00.png - WaveShare Wiki](#)
-  [File:Sensor001.png - WaveShare Wiki](#)
-  [File:Shtc2.png - WaveShare Wiki](#)
- [User Manual](#)

[Manuals+](#), [Privacy Policy](#)

This website is an independent publication and is neither affiliated with nor endorsed by any of the trademark owners. The "Bluetooth®" word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. The "Wi-Fi®" word mark and logos are registered trademarks owned by the Wi-Fi Alliance. Any use of these marks on this website does not imply any affiliation with or endorsement.