2022 - 09 - 10

Connect I2C temperature, humidity and pressure sensor BME280 to NanoPi NEO

In the previous article, I introduced **NanoPi NEO, a small Linux machine, and how to set it up.**

This time, we will connect the **BME280 sensor, which can measure temperature, humidity, and atmospheric pressure, to the NanoPi NEO.**

I previously connected this sensor to my Raspberry Pi . It can be used in the same way with NanoPi NEO.

BME280

BME280 is a humidity/barometric pressure sensor produced by Bosch in Germany. It also has a built-in temperature sensor to accurately measure humidity. It's not necessarily a temperature sensor to measure the teacher, so it may be slightly off, but it's probably about the same.

- Official web page
- Data sheet (PDF)

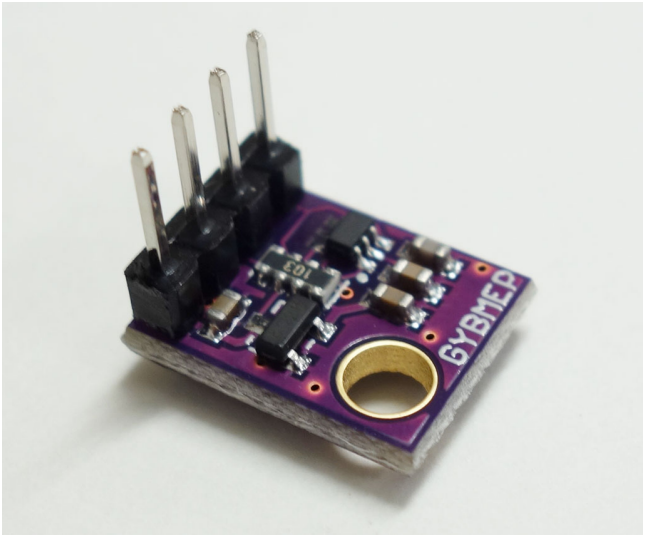
By communicating with I2C or SPI, data on **temperature, humidity, and atmospheric pressure can be obtained.**

Sensors are subject to individual differentes, but they are adjusted at the factory to absorb individual differentces before feeling shipping, making it convenient to obtain data with fewer errors.

Let's connect it to Raspberry Pi

Preparing the BME280 module

The BME280 module I bought was of the type that require you to seller the pin header yourself, so I tried soldering it.



If you don't have a soldering iron, you can also hook **the** terminals together using this **IC test clip cable .** It is convenient because it can be clipped to both pin headers and through-hole holes. However, please be careful not to short it.

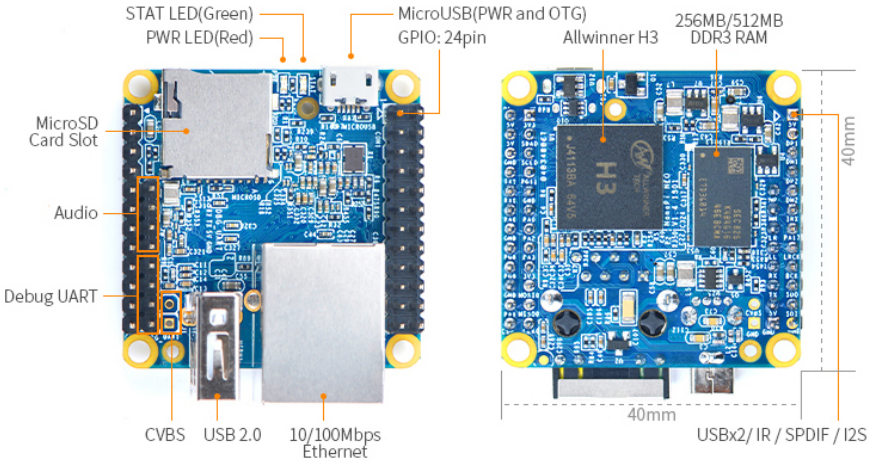
Teishin IC test lead small TLA101

Teishin Electric

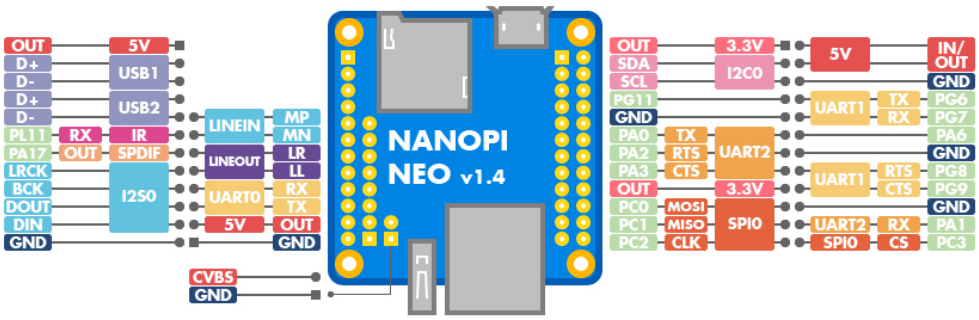
Amazon

Connecting to NanoPi NEO

– Layout & Interface



NanoPi NEO v1.4 pinout diagram



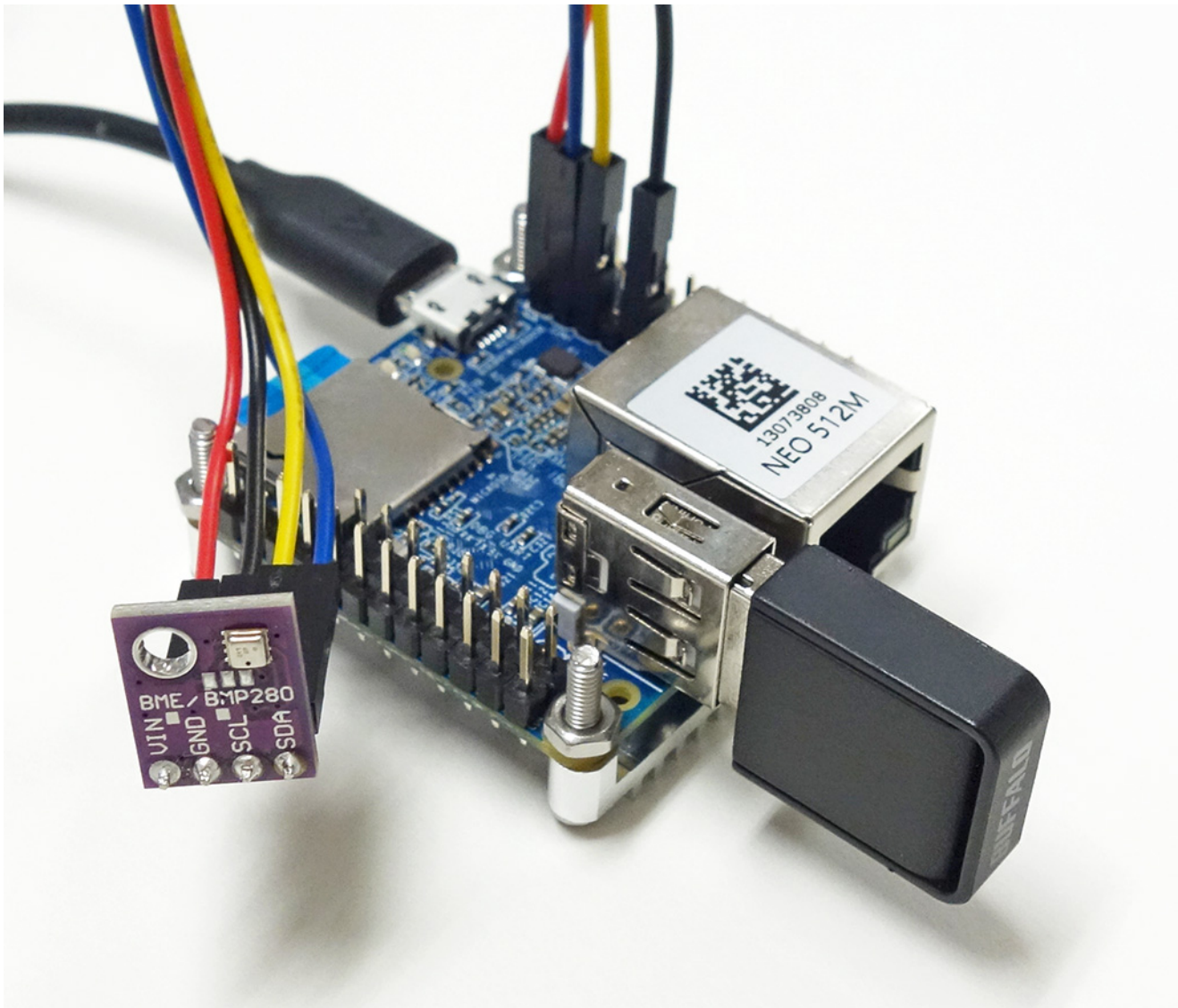
Source: friendlyelec.com

The functions of the Raspberry Pi pin headers are assigned as shown in the diagram above. This module communicates with **I2C** , So connect it to the following pins:

NanoPi pin header	BME280 module	splanation
OUT (3.3V)	VIN	Power supply (3.3V)

NanoPi pin header	BME280 module	splanation
SDA (I2C0)	S.D.A.	I2C data
SCL (I2C0)	SCL	I2C clock
GND	GND	Ground (0V reference voltage)

After connecting them with jump wires, it looks like this:



Access BME280 from Python

Once you have successfully connected, let's try accessing it from **Python** on NanoPi NEO .

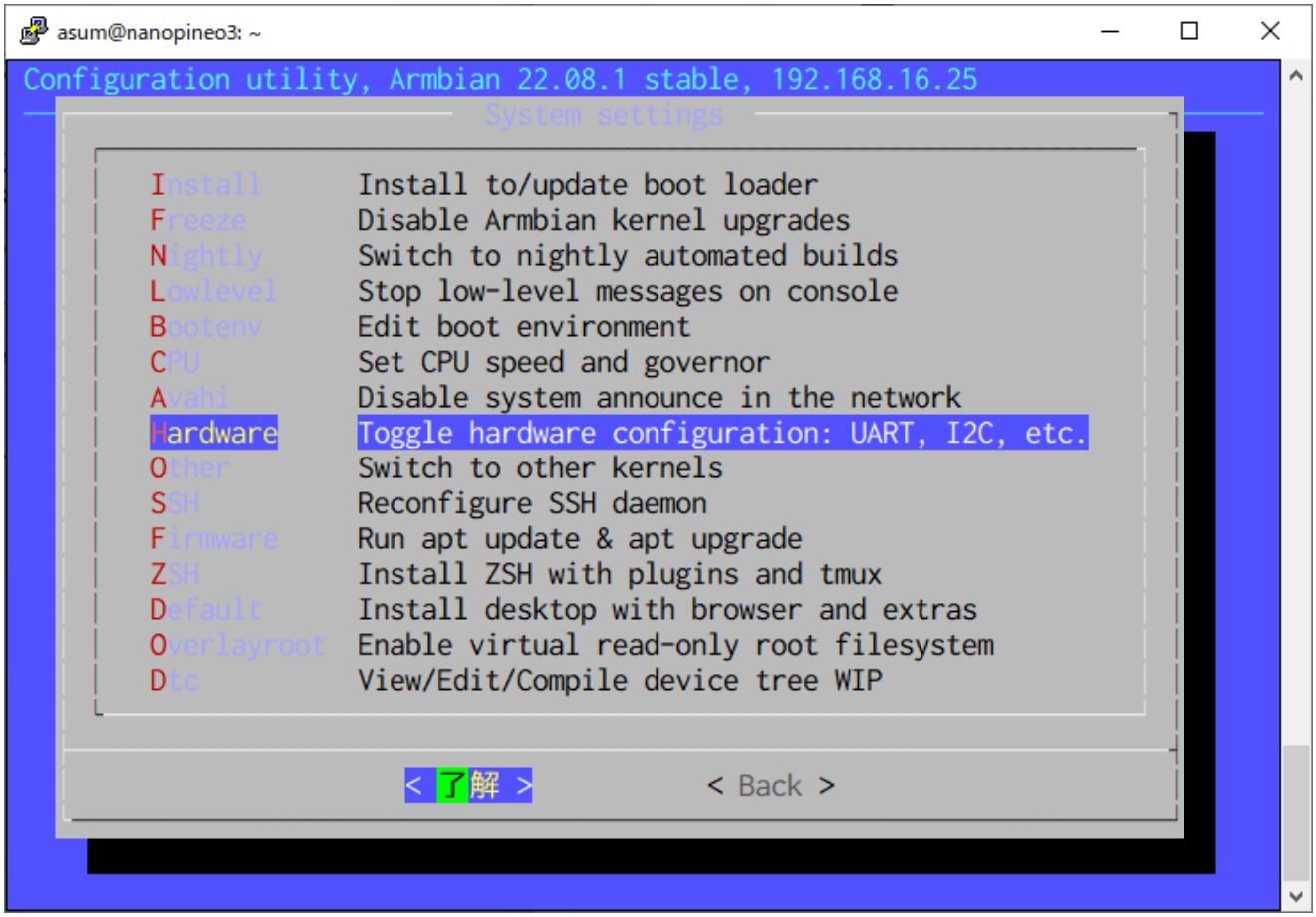
From here on, we will show an example using **Armbian 22.08 Jammy as the OS for NanoPi NEO.** Please note that the operation may be different for other Oses.

Enabling I2C

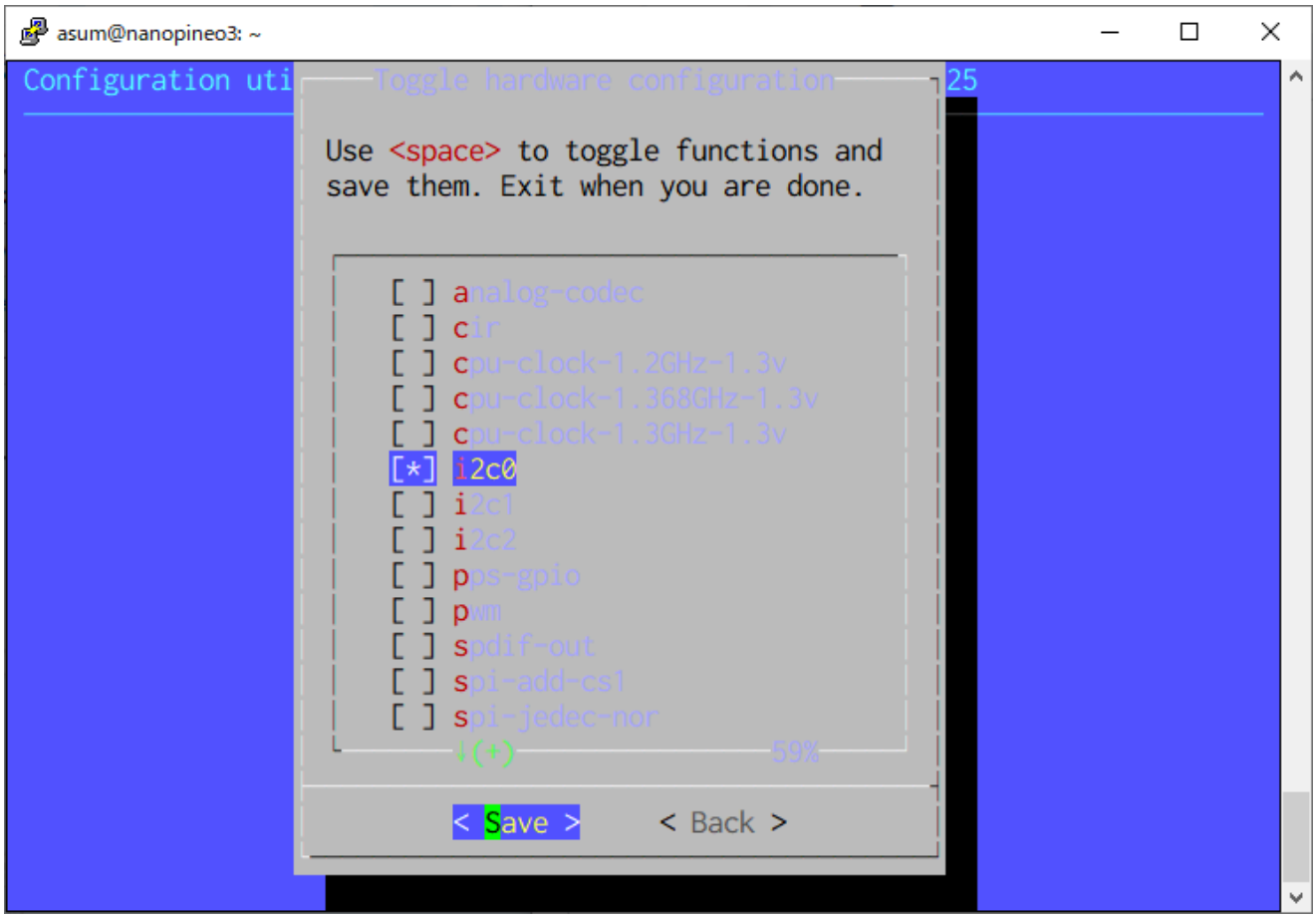
NanoPi NEO's Armbian has **I2C disabled by default.** To enable it, `armbian-config` it is convenient to use.

```
$ sudo armbian-config
```

to start the settings screen.



Select **Hardware .**



Mark the interfaces you want to enable with an asterisk. Press the space key to to oggle ON/OFF.

Now, let's enable **I2C0 .**

`armbian-config` Once you exit, you will be prompted to restart, so do so .

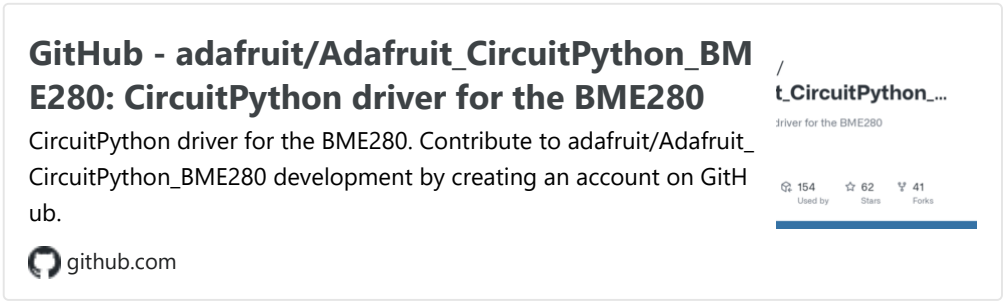
After rebooting, I2C0 is now available. As a test, `i2cdetect` let's run a device scan.

```
$ sudo i2cdetect -y 0
     0 1 2 3 4 5 6 7 8 9 abcdef
00: -- -- -- -- -- -- --
Ten: -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- 76 --
```

Device found at I2C address 0x76. The I2C address of this BME280 module is 0x76, so it is recognized correctly!

Installing Python libraries

Adafruit Industries, an American open source hardware company, has released a Python library for BME280, so let's use it 😊



github.com

It is also registered on PyPI, so you can easily install it using **pip .** First of all `pip` , Install it. You will also `libgpiod` need it, so install it as well.

```
$ sudo apt install python3-pip python3-libgpiod
```

Next, install the library mentioned earlier.

```
$ sudo pip3 install adafruit-circuitpython-bme280
```

The preparation is now complete.

sample program

Once you are ready, create a program like the one below with an appropriate name.

```
$ nano bme280_test.py
```

The content will look like this:

```
import time
 import board
 from adafruit_bme280 import basic as adafruit_bme280

i2c = board.I2C ()
```

```
bme280 = adafruit_bme280.Adafruit_BME280_I2C (i2c, address= 0x76 )

while  True :
     print ( " \n Temperature: % 0.1f C" % Bme280.temperature)
     print ( "Humidity: % 0.1f %%" % Bme280.relative_humidity)
     print ( "Pressure:% 0.1f hPa" % Bme280.pressure )
    time.sleep( 5 )
```

By doing this, you can read the sensor value.

```
$ sudo python3 bme280_simpletest.py

Temperature: 29.9C
Humidity: 49.2%
Pressure: 1012.3hPa

(...continues to display every 1 second)
```

congratulations! You can easily read the sensor value. It's easy if you use a convenient library.

I want to run it as a general user

As you can see from the previous commands, all are executed with **sudo privileges .** Actually, by default in Armbian, sudo privileges are required to access I2C and GPIO.

However, there are times when **you want to run it as a general user .** In such cases, you will need to change the permissions as follows:

I2C permission change

The device file for I2C `/dev/i2c-0` is . This permission looks like this:

```
$ ls -l /dev/i2c*
crw-rw---- 1 root i2c 89, 0 Sep 9 17:32 /dev/i2c-0
```

It seems that root and users in the i2c group can access I2C.

So let's join **ourselves to the i2c group .** For example, if your username is mikan, do the following:

```
$ sudo usermod -aG i2c mikan
```

Now you can access I2C without sudo.

GPIO permission change

In the case of NanoPi NEO, it seems that to access I2C you also need to have GPIO access.

The device file for GPIO `/dev/gpiochip*` is . Let's take a look.

```
$ ls -l /dev/gpio*
crw------- 1 root root 254, 0 Sep 9 17:43 /dev/gpiochip0
crw------- 1 root root 254, 1 Sep 9 17:43 /dev/gpiochip1
```

It has strict permission settings so that **only root can modify it .** Unlike I2C, it can do a lot of things, so if it's tampered with, the system will break, so perhaps they're making it stricter for security reasons.

For example, if you want to make it usable for general users with some risk, you can customize **the udev rules .**

`/etc/udev/rules.d/99-gpiod.rules` Let's create a file with an editor .

```
$ sudo nano /etc/udev/rules.d/99-gpiod.rules
```

Write only the following line and save the contents.

```
SUBSYSTEM=="gpio", KERNEL=="gpiochip[01]", GROUP="gpiod", MODE="0666"
```

Now the permissions will be 666, which means that `/dev/gpiochip0` even `1` regular users can access it.

After rebooting, you will see that the permissions have changed.

```
$ ls -l /dev/gpio*
crw-rw-rw- 1 root root 254, 0 Sep 9 17:32 /dev/gpiochip0
crw-rw-rw- 1 root root 254, 1 Sep 9 17:32 /dev/gpiochip1
```

General users can now access GPIO.

Let's move it

Once you get to this point, you can run the sample program from earlier without sudo.

```
$ python3 bme280_simpletest.py
```

You were able to get the sensor values without any errors.

In this way, NanoPi NEO can also access various sensors etc. using I2C. It's a compact Linux board, so it can be used for many purposes.

Please take advantage of it!

みかん (id:kimura_khs) 1 year ago

Related article

2022-09-08

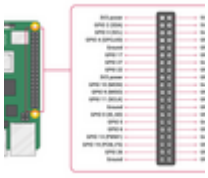
Linux machine smaller than Raspberry Pi ~ Nano Pi NEO

In an article a while back, I talked about not being able to get a Raspberry Pi...

2021-08-09

Connect a $6 volatile organic compound (VOC) sensor SGP30 to your Raspberry Pi

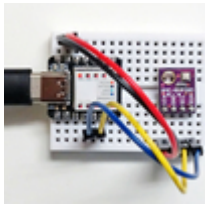
You can find cheap products on AliExpress, an online shopping mall where Chinese shops sell to people outside of China...



2021-06-22

I2C bitbang with Raspberry Pi!! ~ Use I2C with your favorite pins

As used in past articles, Raspberry Pi has an I2C function...



2021-06-04

Read the value of temperature, humidity and barometric pressure sensor BME280 with Zephyr OS

Continuing from last time, I installed Zephyr OS on Seeeduino XIAO and played with it...

2021-04-16

Let's take a look at the contents of the Zephyr OS sample project.

Note: This article is based on Zephyr OS v2.7 In the previous article, Zephyr O…

write a comment

PREV

Put a clock on your Raspberry Pi - $1.9...

NEXT

Linux machine smaller than Raspberry Pi ~ Nano…

profile

I am an engineer involved in the development of IoT services in general, with a focus on firmware development for wireless modules.

be a reader

About this blog

Featured articles

Turn Raspberry Pi into a WiFi access point for standalone operation 2 users

Play a speaker with Raspberry Pi ~ Play from Python 4 users

Bluetooth Low Energy ~ Learn how GATT works 1 user

Raspberry Pi WiFi access point + client simultaneous use 4 users

Raspberry Pi WiFi access point & router 3 users

lateest article

Raspberry Pi WiFi access point & router 3 users

Raspberry Pi WiFi access point + client simultaneous use 4 users

Turn Raspberry Pi into a WiFi access point for standalone operation 2 users

Put a clock on your Raspberry Pi -use the $ 1.9 RTC DS3231

Connect I2C template, humor and pressure sensor BME280 to NanoPi NEO

See more

Recommended products from Mikan

Sponsoring