# Tutorial flask iot

### 1. install flask

in raspberry open terminal (alt+ctrl+t) and write script below into terminal

```
sudo pip install flask
```

### 2. install opencv

install opencv in raspberry, follow this tutorial from
[https://qengineering.eu/install-opencv-4.1-on-raspberry-pi-4.html](https://qengineering.eu/install-opencv-4.1-on-raspberry-pi-4.html)

### 3. config camera and I2C

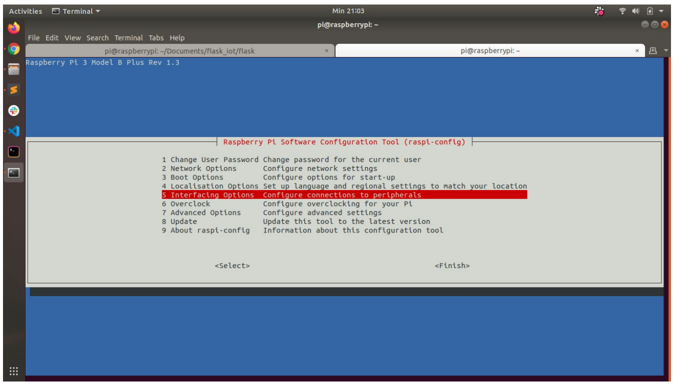
in raspberry open terminal (alt+ctrl+t) and write script below into terminal

```
sudo apt update
```

```
sudo apt full-upgrade
```

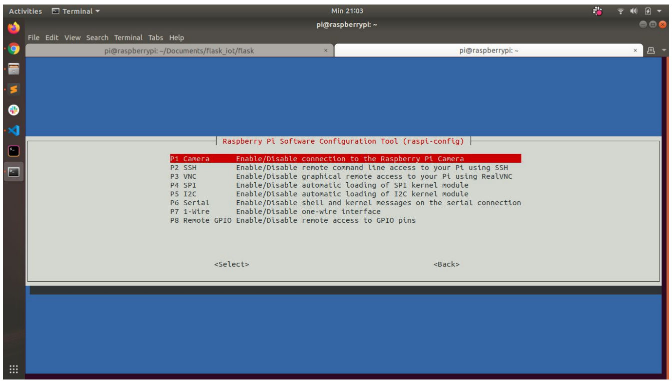
```
sudo raspi-config
```

Use the cursor keys to select and open *Interfacing Options*, and then select *Camera* and follow the prompt to enable the camera.



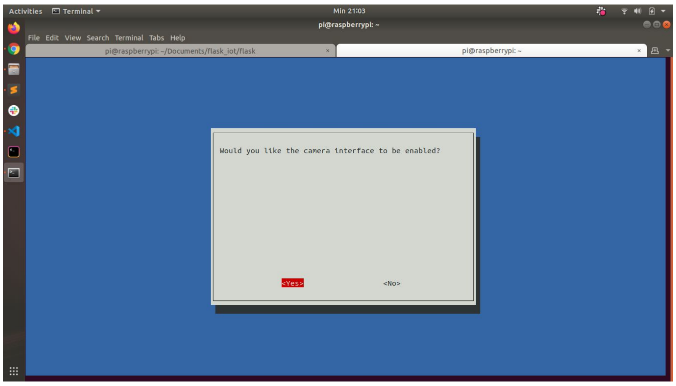
Activities
Terminal
Min 21:03
pi@raspberrypi: ~
File Edit View Search Terminal Tabs Help
pi@raspberrypi: ~/Documents/flask_iot/flask
pi@raspberrypi: ~

```
Raspberry Pi 3 Model B Plus Rev 1.3

         Raspberry Pi Software Configuration Tool (raspi-config)

 1 Change User Password Change password for the current user
 2 Network Options      Configure network settings
 3 Boot Options         Configure options for start-up
 4 Localisation Options Set up language and regional settings to match your location
 5 Interfacing Options  Configure connections to peripherals
 6 Overclock            Configure overclocking for your Pi
 7 Advanced Options     Configure advanced settings
 8 Update               Update this tool to the latest version
 9 About raspi-config   Information about this configuration tool

          <Select>                                   <Finish>
```



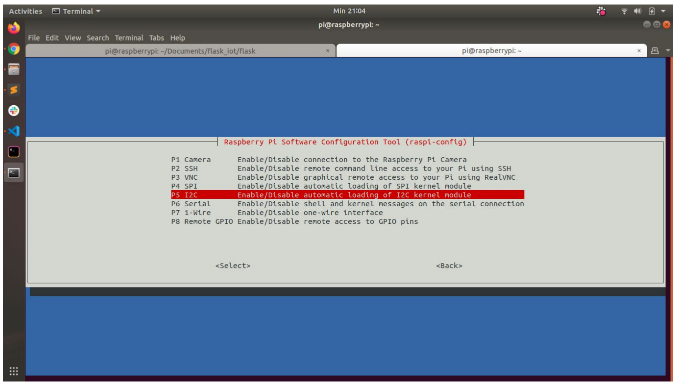
Activities
Terminal
Min 21:03
pi@raspberrypi: ~
File Edit View Search Terminal Tabs Help
pi@raspberrypi: ~/Documents/flask_iot/flask
pi@raspberrypi: ~

```
         Raspberry Pi Software Configuration Tool (raspi-config)

 P1 Camera      Enable/Disable connection to the Raspberry Pi Camera
 P2 SSH         Enable/Disable remote command line access to your Pi using SSH
 P3 VNC         Enable/Disable graphical remote access to your Pi using RealVNC
 P4 SPI         Enable/Disable automatic loading of SPI kernel module
 P5 I2C         Enable/Disable automatic loading of I2C kernel module
 P6 Serial      Enable/Disable shell and kernel messages on the serial connection
 P7 1-Wire      Enable/Disable one-wire interface
 P8 Remote GPIO Enable/Disable remote access to GPIO pins

          <Select>                                   <Back>
```

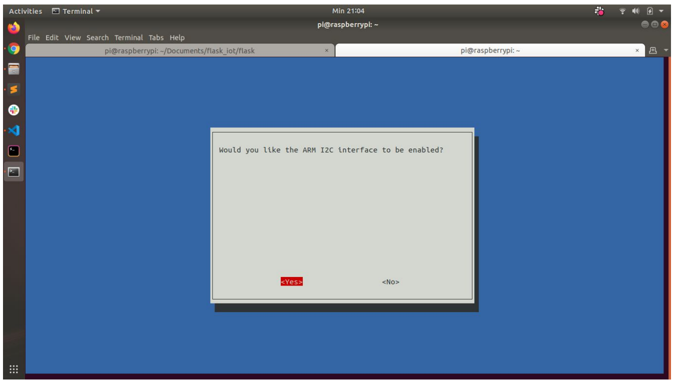

Activities
Terminal
Min 21:03
pi@raspberrypi: ~
File Edit View Search Terminal Tabs Help
pi@raspberrypi: ~/Documents/flask_iot/flask
pi@raspberrypi: ~

```
Would you like the camera interface to be enabled?

        <Yes>                    <No>
```

## Enable I2C


Activities
Terminal
Min 21:04
pi@raspberrypi: ~
File Edit View Search Terminal Tabs Help
pi@raspberrypi: ~/Documents/flask_iot/flask
pi@raspberrypi: ~

```
Raspberry Pi Software Configuration Tool (raspi-config)

P1 Camera      Enable/Disable connection to the Raspberry Pi Camera
P2 SSH         Enable/Disable remote command line access to your Pi using SSH
P3 VNC         Enable/Disable graphical remote access to your Pi using RealVNC
P4 SPI         Enable/Disable automatic loading of SPI kernel module
P5 I2C         Enable/Disable automatic loading of I2C kernel module
P6 Serial      Enable/Disable shell and kernel messages on the serial connection
P7 1-Wire      Enable/Disable one-wire interface
P8 Remote GPIO Enable/Disable remote access to GPIO pins

           <Select>                          <Back>
```


Activities
Terminal
Min 21:04
pi@raspberrypi: ~
File Edit View Search Terminal Tabs Help
pi@raspberrypi: ~/Documents/flask_iot/flask
pi@raspberrypi: ~

```
Would you like the ARM I2C interface to be enabled?

        <Yes>                    <No>
```

restart your raspberry pi, open again terminal and write script below into terminal.

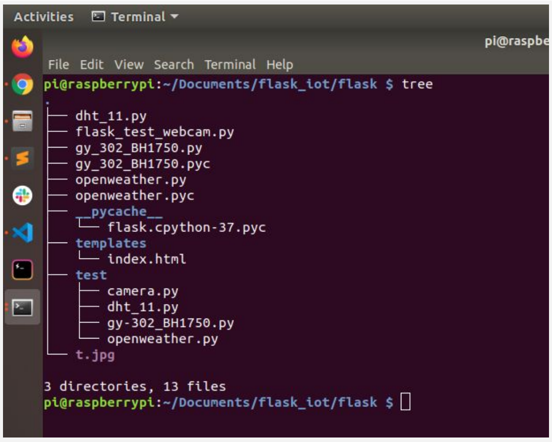
```
raspistill -v -o test.jpg
```

check in pictures if there are a picture from camera your camera was connected. if the camera not found, test camera using cheese

```
sudo apt-get install cheese
and write
cheese
```

### 4. structure file

the root folder includes file py and html. if you start the program, make sure you are in the right path.



```
pi@raspberrypi:~/Documents/flask_iot/flask $ tree
.
├── dht_11.py
├── flask_test_webcam.py
├── gy_302_BH1750.py
├── gy_302_BH1750.pyc
├── openweather.py
├── openweather.pyc
├── __pycache__
│   └── flask.cpython-37.pyc
├── templates
│   └── index.html
├── test
│   ├── camera.py
│   ├── dht_11.py
│   ├── gy-302_BH1750.py
│   └── openweather.py
└── t.jpg

3 directories, 13 files
pi@raspberrypi:~/Documents/flask_iot/flask $ 
```

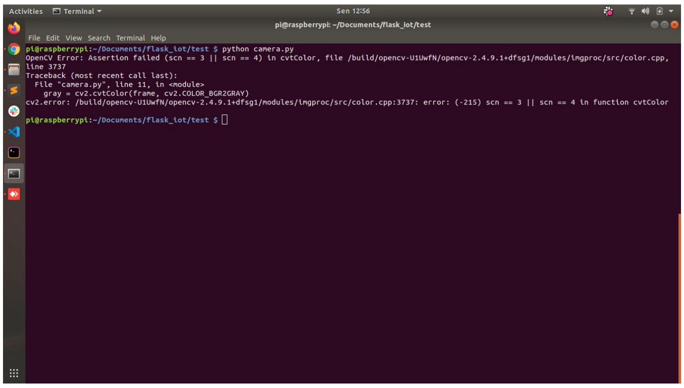
### 5. test camera

test camera using opencv open folder test and open terminal. make sure you are in **test** folder before you write script bellow into terminal

```
python camera.py
```

if success the window will showing video from your camera and press q to exit
if the window show error like picture below, try to change port from 0 or 1 or 2 or 3 in the file camera.py
example

cap = cv2.VideoCapture(0) #try change value ini VideoCapture


Activities
Terminal
Sen 12:56
pi@raspberrypi: ~/Documents/flask_iot/test
File Edit View Search Terminal Help

```
pi@raspberrypi:~/Documents/flask_iot/test $ python camera.py
OpenCV Error: Assertion failed (scn == 3 || scn == 4) in cvtColor, file /build/opencv-U1UwfN/opencv-2.4.9.1+dfsg1/modules/imgproc/src/color.cpp,
line 3737
Traceback (most recent call last):
  File "camera.py", line 11, in <module>
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
cv2.error: /build/opencv-U1UwfN/opencv-2.4.9.1+dfsg1/modules/imgproc/src/color.cpp:3737: error: (-215) scn == 3 || scn == 4 in function cvtColor

pi@raspberrypi:~/Documents/flask_iot/test $ 
```

### 6. install library sensor dht-11

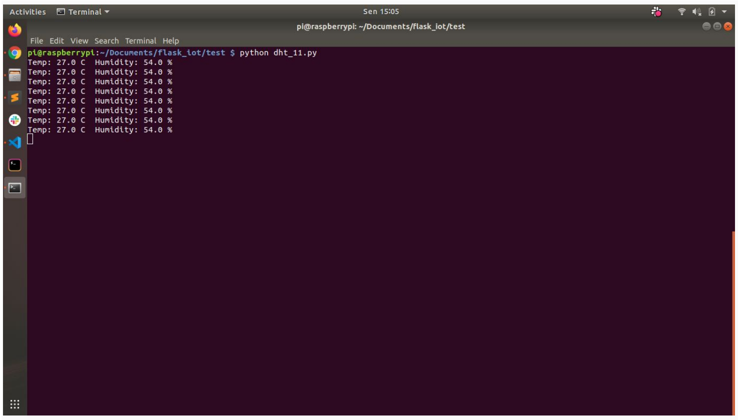
install library. open terminal and write script below

```
sudo apt-get install git-core
git clone https://github.com/adafruit/Adafruit_Python_DHT.git
cd Adafruit_Python_DHT
sudo apt-get install build-essential python-dev
sudo python setup.py install
```

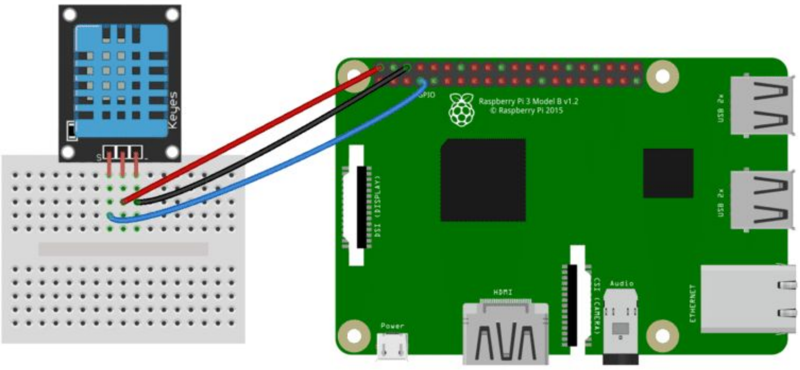
### 7. test sensor dht-11

to test the sensor, make sure you are in **test** folder before you write script below into terminal

```
python dht_11.py
```

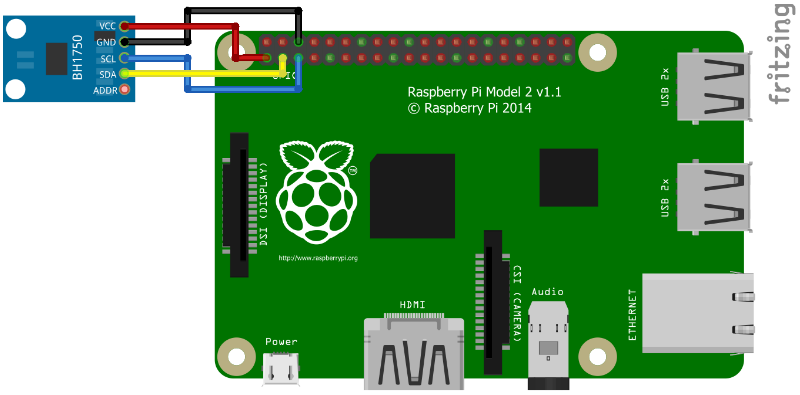


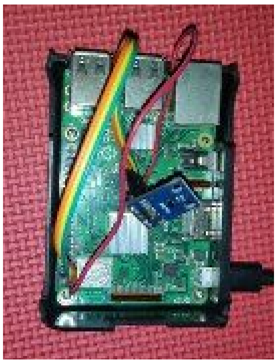
```
pi@raspberrypi:~/Documents/flask_iot/test $ python dht_11.py
Temp: 27.0 C  Humidity: 54.0 %
Temp: 27.0 C  Humidity: 54.0 %
Temp: 27.0 C  Humidity: 54.0 %
Temp: 27.0 C  Humidity: 54.0 %
Temp: 27.0 C  Humidity: 54.0 %
Temp: 27.0 C  Humidity: 54.0 %
Temp: 27.0 C  Humidity: 54.0 %
Temp: 27.0 C  Humidity: 54.0 %
```



## 8. sensor gy-302

follow the configuration of sensor like picture bellow and make sure the cable has been installed properly

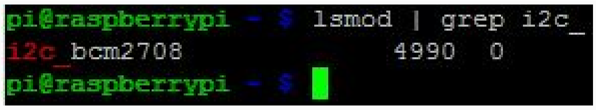




### 9. check your I2C connection

write script bellow into your terminal

```
lsmod | grep i2c_
```

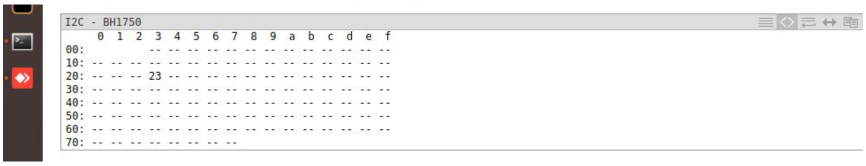


```
pi@raspberrypi ~ $ lsmod | grep i2c_
i2c_bcm2708            4990  0
pi@raspberrypi ~ $
```

then write script below into your terminal

```
sudo apt-get install i2c-tools
sudo i2cdetect -y 1
```

if you see the terminal show like picture bellow, your I2C has been connected



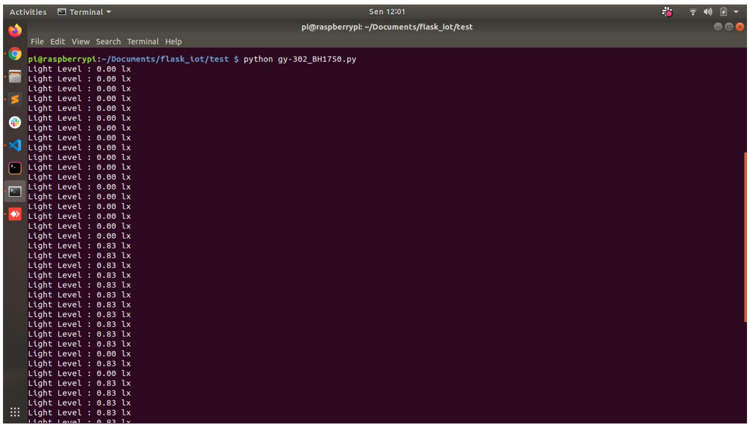
```
I2C - BH1750
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- 23 -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

### 10. test sensor gy-302

to test sensor,  make sure you are in **test** folder before you write script below into terminal.

```
python gy-302_BH1750.py
```

if you see the terminal like this



Activities
Terminal
Sen 12:01
pi@raspberrypi: ~/Documents/flask_iot/test
File Edit View Search Terminal Help

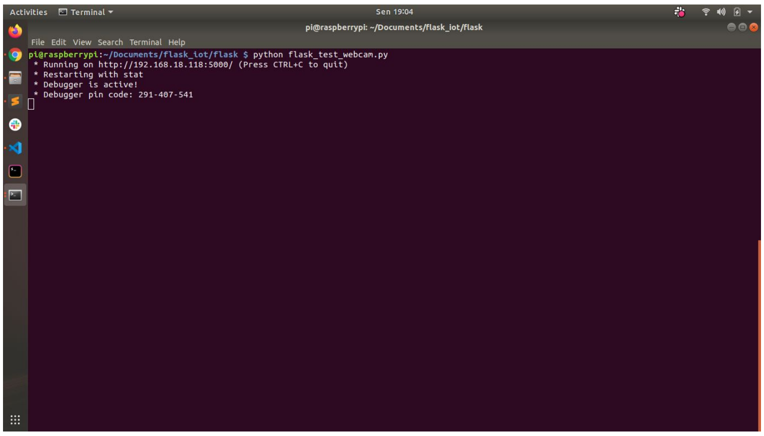
```
pi@raspberrypi:~/Documents/flask_iot/test $ python gy-302_BH1750.py
Light Level : 0.00 lx
Light Level : 0.00 lx
Light Level : 0.00 lx
Light Level : 0.00 lx
Light Level : 0.00 lx
Light Level : 0.00 lx
Light Level : 0.00 lx
Light Level : 0.00 lx
Light Level : 0.00 lx
Light Level : 0.00 lx
Light Level : 0.00 lx
Light Level : 0.00 lx
Light Level : 0.00 lx
Light Level : 0.00 lx
Light Level : 0.00 lx
Light Level : 0.00 lx
Light Level : 0.00 lx
Light Level : 0.00 lx
Light Level : 0.83 lx
Light Level : 0.83 lx
Light Level : 0.83 lx
Light Level : 0.83 lx
Light Level : 0.83 lx
Light Level : 0.83 lx
Light Level : 0.83 lx
Light Level : 0.83 lx
Light Level : 0.83 lx
Light Level : 0.83 lx
Light Level : 0.83 lx
Light Level : 0.00 lx
Light Level : 0.83 lx
Light Level : 0.00 lx
Light Level : 0.83 lx
Light Level : 0.83 lx
Light Level : 0.83 lx
Light Level : 0.83 lx
Light Level : 0.83 lx
```

sensor has been worked

## 11. flask web dashboard

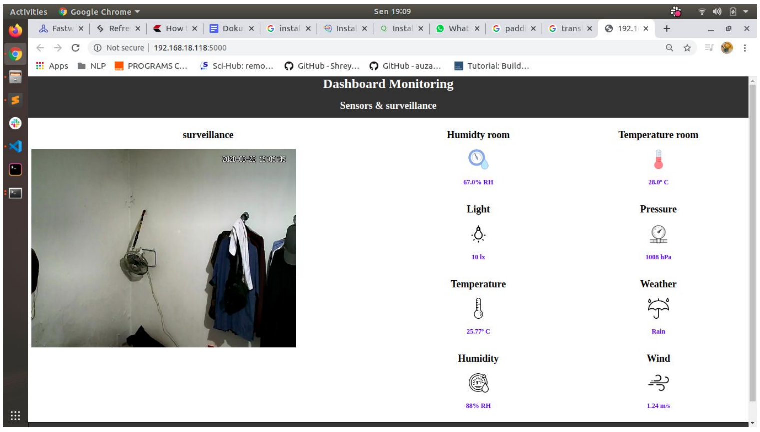
to start flask web service framework, make sure you are in the right path. the main program is in the flask folder by name **flask_test_webcam.py.** to start the program, open terminal and write script below into terminal

```
python flask_test_webcam.py
```



```
pi@raspberrypi:~/Documents/flask_iot/flask $ python flask_test_webcam.py
 * Running on http://192.168.18.118:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger pin code: 291-407-541
```

open link https into your browser. if no error, the browser will show dashboard like this



Dashboard Monitoring

Sensors & surveillance

surveillance

Humidty room: 67.0% RH

Temperature room: 28.0° C

Light: 10 lx

Pressure: 1008 hPa

Temperature: 25.77° C

Weather: Rain

Humidity: 88% RH

Wind: 1.24 m/s

your dashboard monitoring ready to use