**Application IoT Raspberry avec Linux Embarqué**

La figure suivante présente le fonctionnement des GPIO de la carte Raspberry PI3.

# Résultat de recherche d'images pour "raspberry pi 3 gpio map"

# Préparation à l’environnement de programmation

**Raspberry Pi 3:**

## 1- module gpio python

sudo apt-get update

sudo apt-get install python-dev

sudo apt-get install python-rpi.gpio

## 2-module dht11

sudo apt-get install git-core

git clone https://github.com/adafruit/Adafruit\_Python\_DHT.git

cd Adafruit\_Python\_DHT

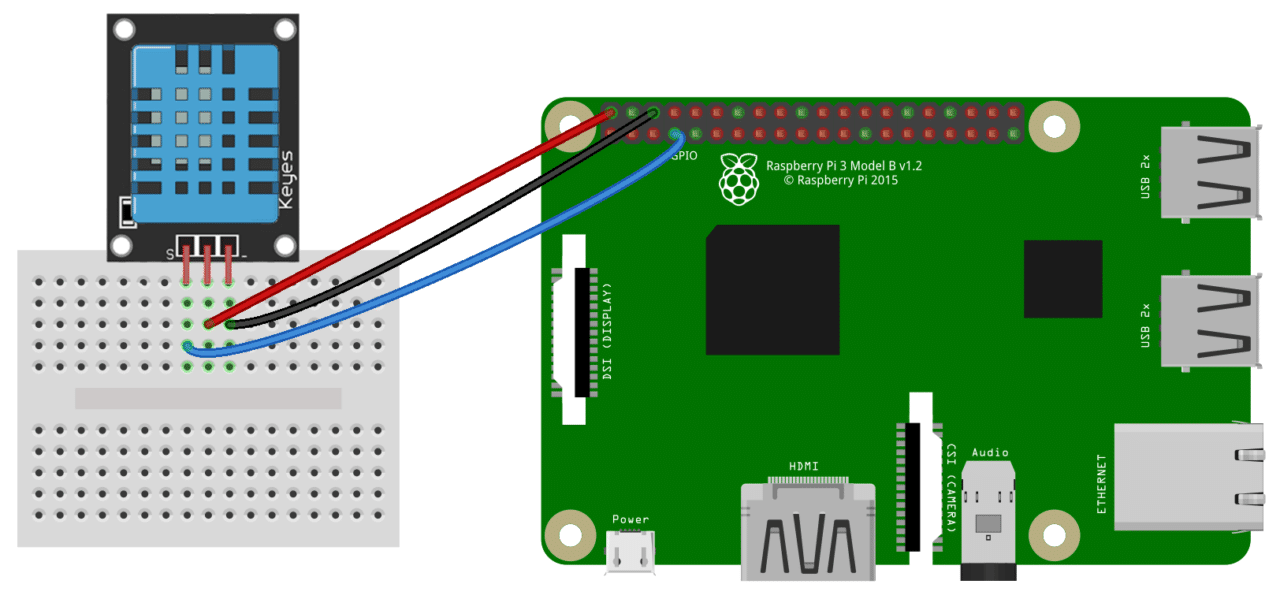
sudo apt-get install build-essential python-dev

sudo python setup.py install



Connecter DHT11 au GPIO27 selon :

|  |  |
| --- | --- |
| Module DHT11 | Carte Raspberry |
| VCC | PIN-2 (5V) |
| Gnd | PIN-6 (GND) |
| Data | PIN-13 (GPIO-27) |



## Code de lecture du capteur DHT11

import Adafruit\_DHT

while True:

humidity, temperature = Adafruit\_DHT.read\_retry(11, 27) # GPIO27 (BCM notation)

print ("Humidity = {} %; Temperature = {} C".format(humidity, temperature))

## Création et lancement d’un programme python

Utiliser l’éditeur « nano »

* nano fichier.py

Pour exécuter un script python :

* python fichier.py

**Raspberry Pi 4 utilisant la biblio pigpio-dht**

pip3 install pigpio-dht

sudo pigpiod #Start daemon

Ecrire votre programme :

from pigpio\_dht import DHT11, DHT22

import math

import time

gpio = 17 # BCM Numbering

sensor = DHT11(gpio)

#sensor = DHT22(gpio)

while True:

result = sensor.read()

print("Temperature :{}".format(result['temp\_c']))

print("Humidity :{}".format(result[‘humidity’]))

time.sleep(3)

Avant d’exécuter votre code, tapper :

pigs pud 4 u

Exécuter le code :

python3 code.py