GitHub <https://github.com/internetdelascosas/RaspberryPi-DHTxx>

git clone https://github.com/internetdelascosas/RaspberryPi-DHTxx.git

[GitHub - internetdelascosas/RaspberryPi-DHTxx: Programas ejemplo de como conectar un sensor de temperatura y humedad DHTxx (DHT11 y DHT22) a una Raspberry Pi](https://github.com/internetdelascosas/RaspberryPi-DHTxx)

config.py

dht\_consola.py

dht\_log.py

dht\_mysql.py

README.md

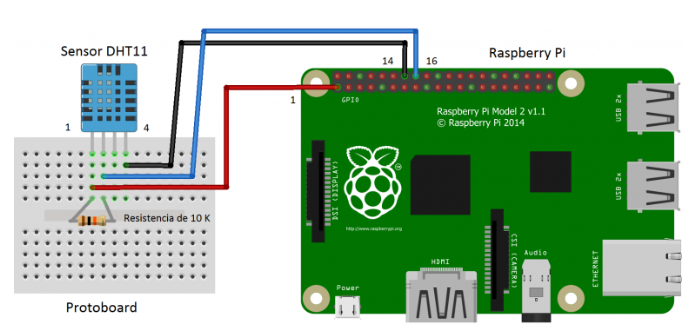
sql

sudo apt-get update

sudo apt-get install git-core

sudo apt-get install python3-dev \*\*default-libmysqlclient-dev\*\*

**DIAGRAMA**



|  |  |
| --- | --- |
| Sensor DHT11 | Raspberry Pi |
| Pin 1 | Pin 1 |
| Pin 2 | Pin 16 |
| Pin 3 (No se utiliza) |  |
| Pin 4 | Pin 14 |

**PRESENTAR TEMPERATURA Y HUMEDAD**

Creamos la carpeta proyectos

$ sudo mkdir /var/proyectos

#!/usr/bin/python3

import sys

import time

import adafruit\_dht

sensor = adafruit\_dht.DHT22(23) # GPIO 23

while True:

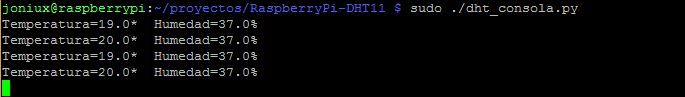
humedad = sensor.humidity

temp = sensor.temperature

print('Temperatura={0:0.1f} C Humedad={1:0.1f}%'.format(temp, humedad))

time.sleep(10)

sudo ./dht\_consola.py



Hasta presionar CTL-C

**GRABAR EN ARCHIVO LOG**

Crea directorio donde estará el archivo log

$ sudo mkdir /var/log

def WriteLog(texto):

path = "/var/log/"

log = open(path + datetime.datetime.now().strftime("%Y-%m-%d") + "\_dht.log","a")

line = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S") + " " + texto + "\n"

log.write(line)

log.close()

text = 'Temperatura={0:0.1f} C Humedad={1:0.1f}%'.format(temp, humedad))

WriteLog(text)

Ejecutar el programa dht\_log.py también como súper usuario con el comando sudo

**EJECUTAR EN BACKGROUND**

$ sudo ./dht\_log.py & [1] 13278

**Se ejecuta en background con numero de proceso 13278.**

Si ejecutas el comando ps, verás que existen dos procesos corriendo con el nombre dht\_log

$ ps aux | grep dht\_log

root     13350  1.0  0.3   6184  3088 pts/1    S    00:47   0:00 sudo ./dht\_log.py

root     13354 10.3  0.7  10752  7504 pts/1    S    00:47   0:00 /usr/bin/python ./dht\_log.py

joniux   13356  0.0  0.2   4276  1896 pts/1    S+   00:47   0:00 grep --color=auto dht

y si revisas el archivo log, veras que esta escribiendo el registro cada 10 segundos.

$ tail /var/log/2017-05-19\_dht.log

2017-05-19 17:57:14 DHT Sensor - Temperatura: 21.0

2017-05-19 17:57:14 DHT Sensor - Humedad:  38.0

2017-05-19 17:57:24 DHT Sensor - Temperatura: 21.0

2017-05-19 17:57:24 DHT Sensor - Humedad:  38.0

2017-05-19 17:57:34 DHT Sensor - Temperatura: 21.0

2017-05-19 17:57:34 DHT Sensor - Humedad:  38.0

2017-05-19 17:57:44 DHT Sensor - Temperatura: 21.0

2017-05-19 17:57:44 DHT Sensor - Humedad:  38.0

Si deseas terminar la ejecución el programa debes «matar» los dos procesos que están corriendo con kill y el numero de proceso, información que se muestra en la segunda columna al ejecutar el comando ps.

En este ejemplo para matar los dos procesos se debe ejecutar

$ kill 13350 13354

**SI DESEA GUARDAR DATOS EN MYSQL EN RASPBERRY PI**

Para crear la base de datos  primero debes instalar el servidor MySQL en la Raspberry Pi crear la base de datos y luego crear la tabla con el script disponible en la carpeta sql.

$ cat ./sql/create\_data.sql | mysql -u [tu\_usuario] -p [tu\_base\_de\_datos]

Enter password:

La ejecución del programa es muy similar al programa anterior.

CREATE TABLE `datos` (

`id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

`fecha` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',

`temperatura` decimal(10,2),

`humedad` decimal(10,2)

);

def write\_mysql(temperatura, humedad):

# db - mysql.connector.connet(

db = MySQLdb.connect(

host="localhost",

user="usuario",

passwd="clave",

db=”MYSQL\_BD”)

cursor = db.cursor()

sql = "INSERT INTO datos (fecha, temperatura, humedad) values (now(),'" + str(temperatura) + "','" + str(humedad) + "');"

try:

cursor.execute(sql)

db.commit()

except:

db.rollback()

cursor.close

db.close

$ sudo ./dht\_mysql.py & [1] 13278

**SI DESEA LLAMAR A FUNCION MAIN DESDE INICIO**

if \_\_name\_\_ == "\_\_main\_\_":

main()

Cómo conectar un Sensor de temperatura DS18B20 a Raspberry Pi

En este pequeño tutorial veremos cómo conectar un sensor de temperatura DS18B20 a una tarjeta Raspberry Pi, también encontrarás un código de programación para tus primeras pruebas.

El sensor DS18B20 es un componente**digital** de **entrada**, es decir, su funcionamiento consiste en entregar la información correspondiente a la temperatura del ambiente por medio de un protocolo de comunicación específico implementado en su pin de señal, la tarjeta Raspberry Pi a su vez recibe la señal y la lee con uno de sus GPIO.

Raspberry Pi – Sensor de temperatura DS18B20 – Resistor de 4.7 kohms

1.- Instalar paquetes de Python3

En este código hacemos uso del paquete de Python3 «w1thermsensor». Para instalarlo ejecuta el siguiente comando en la terminal:

pi@raspberrypi:~$ sudo pip3 install w1thermsensor

2.- Habilitar comunicación One-Wire en la Raspberry Pi

Para que podamos ejecutar un programa que intercambie información por One-Wire en nuestra Raspberry Pi, debemos habilitar el hardware correspondiente en dicha tarjeta. Esto lo podemos hacer de manera sencilla con ayuda de la herramienta de configuración *raspi-config*, si no has utilizado esta herramienta antes [da click aquí para ver el paso a paso](https://blog.330ohms.com/2020/06/17/como-habilitar-la-comunicacion-1-wire-en-raspberry-pi/). Una vez que termines regresa para ejecutar el código.

Código de Python3

import time

from w1thermsensor import W1ThermSensor

sensor = W1ThermSensor()

while True:

temperature = sensor.get\_temperature()

print("The temperature is %s celsius" % temperature)

time.sleep(1)