



CONTROL DEL PUERTO GPIO DE LA RASPBERRY

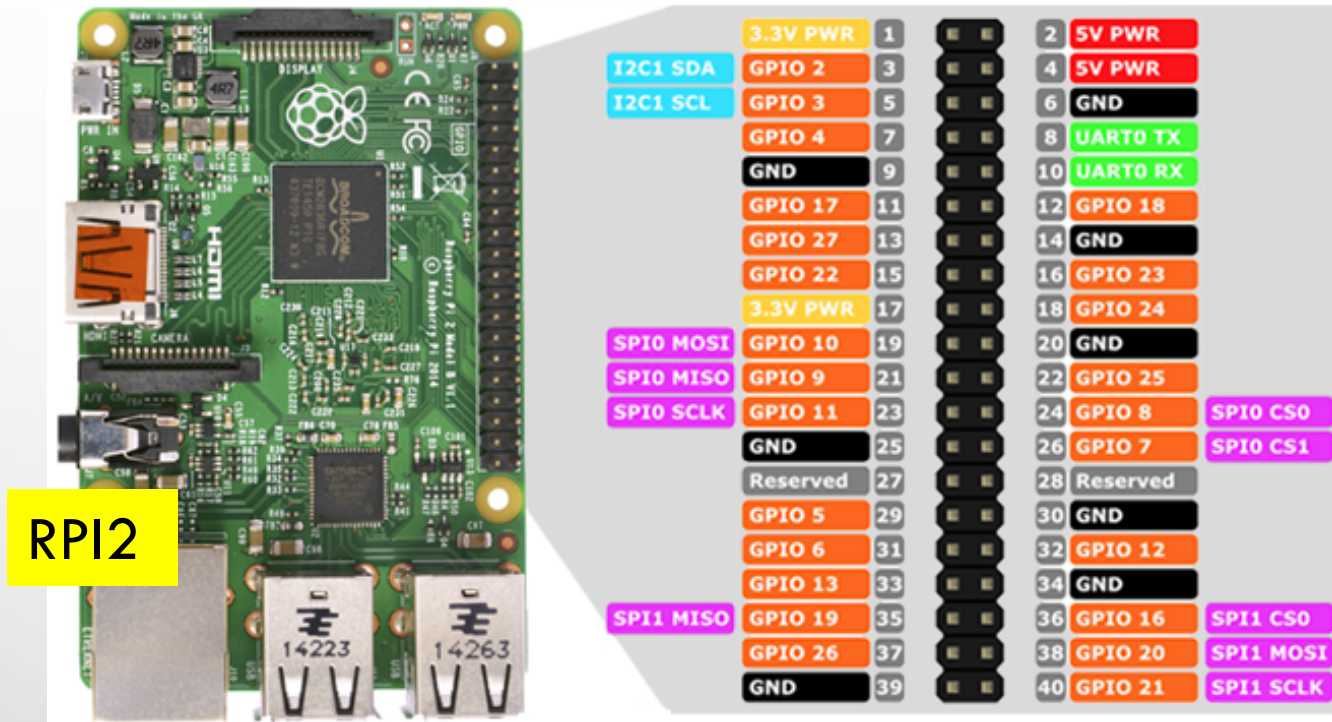
PYTHON PARA EL CONTROL DEL GPIO

presentado por:

Francisco Javier Ferrández Pastor

SENSOR DHT CONECTADO A RPI

INTRODUCCIÓN



CONTROL DEL PUERTO GPIO DE LA RASPBERRY





















INTRODUCCIÓN

De los pines GPIO disponibles, hay una serie de pines con capacidad de PWM (para control de potencia en pin12 GPIO18).

Sin embargo no se dispone de ningún convertidor de analógico a digital. Esto quiere decir que para medir valores de sensores analógicos necesitaremos utilizar un convertidor (conversor A/D o Arduino).

ADVERTENCIA

Cuando se utilizan los pines de *GPIO* hay que poner mucho cuidado para no dañar la propia Raspberry Pi. Es muy importante comprobar los niveles de tensión y la corriente solicitada. Los pines de GPIO pueden generar y consumir tensiones compatibles con los circuitos de 3.3V. No conectar nunca componentes de 5V a los IO, o se puede quemar la electrónica.

BOARD	GPIO		GPIO	BOARD
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)		DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)		(I ² C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37			GPIO20	38
39			GPIO21	40

RPI3

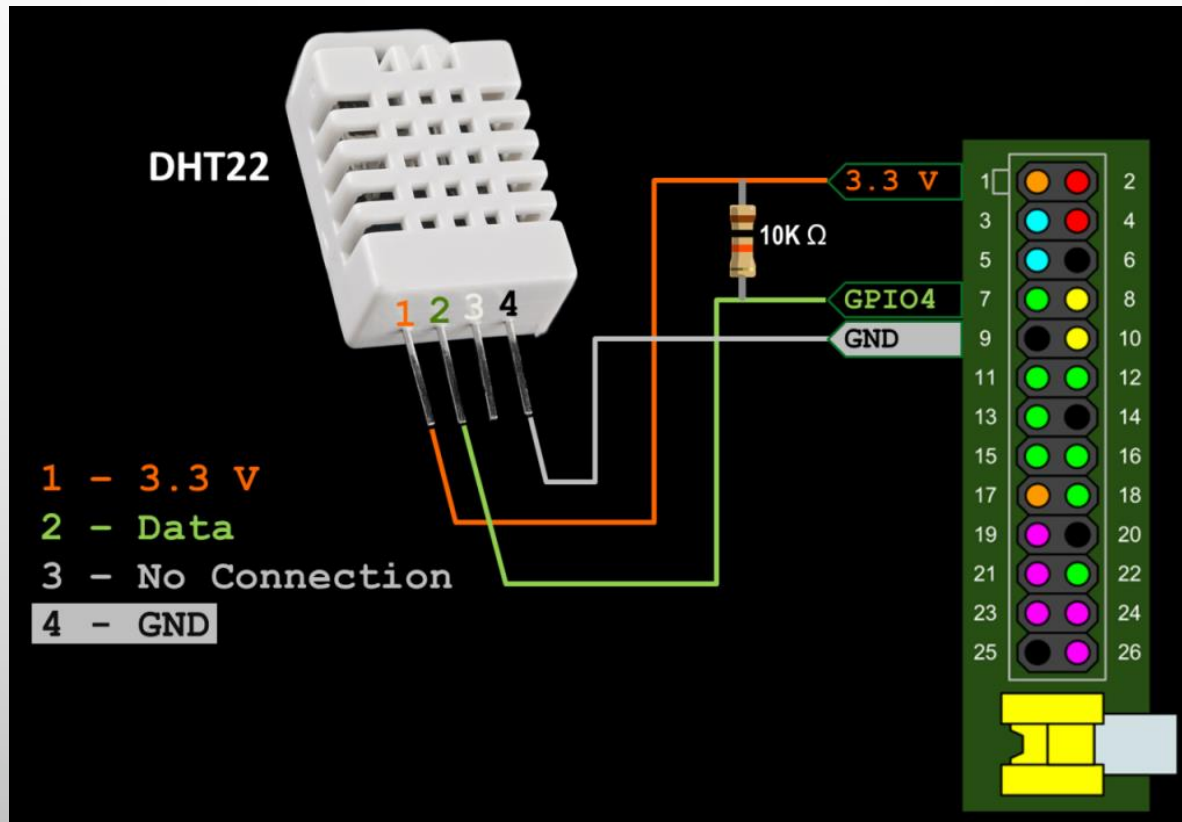
SENSOR DHT CONECTADO A RPI

PROGRAMACIÓN EN PYTHON

EJEMPLO 2:

- Sensor de Temperatura y Humedad (DHT22) conectado al GPIO

Programa en Python que captura la T y la H utilizando librería de ADAFRUIT y sensor DHT22 conectado al GPIO4

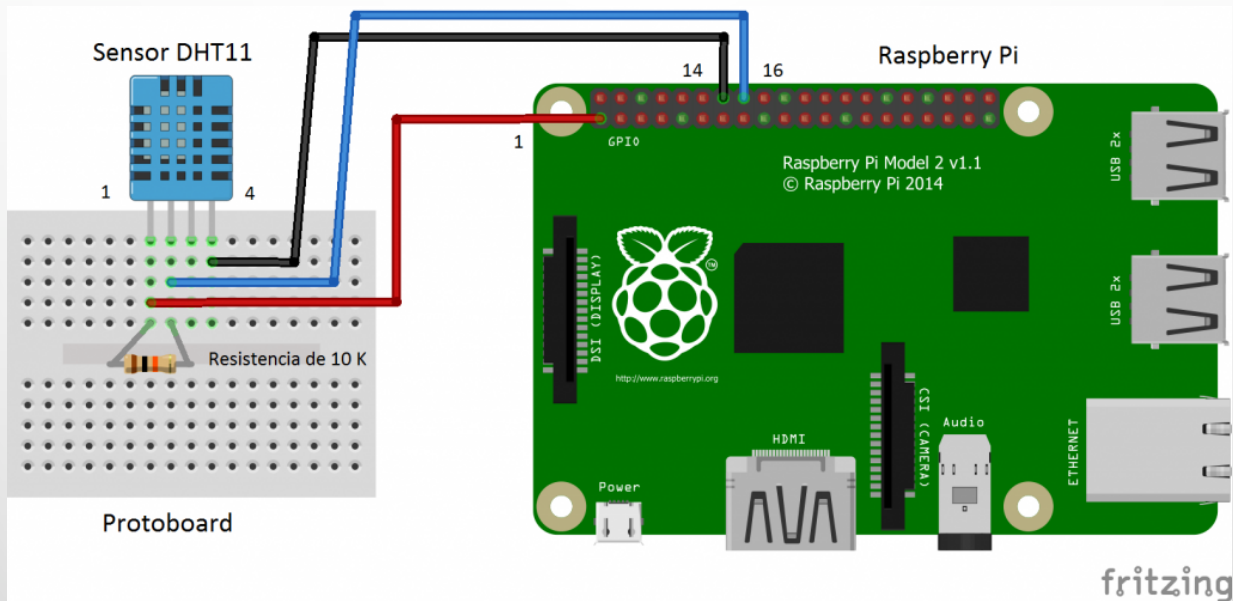


SENSOR DHT CONECTADO A RPI

PROGRAMACIÓN EN PYTHON

- Sensor de Temperatura y Humedad (DHT11 o 22) conectado al GPIO

<https://www.internetdelascosas.cl/2017/05/19/raspberry-pi-conectando-un-sensor-de-temperatura-y-humedad-dht11/>



SENSOR DHT CONECTADO A RPI

PROGRAMACIÓN EN PYTHON

Procedimiento

Lo primero que debemos hacer es instalar git en nuestra Raspberry Pi en el caso de que no lo tengamos instalado. Siempre es recomendable hacer un apt-get update antes de comenzar.

```
>> sudo apt-get update
```

Luego instalar git

```
>> sudo apt-get install git-core
```

Con git instalado procedemos a crearnos una carpeta para almacenar las librerías que instalaremos para este u otros proyectos

```
>> mkdir librerias
```

Luego clonamos el repositorio git de la librería de Adafruit que nos permitirá obtener las lecturas

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

Para instalar esta librería en nuestro sistema debemos hacer lo siguiente. Instalar software que nos permite adicionar librerías Python a nuestra Raspberry Pi

```
>> sudo apt-get install build-essential python-dev
```

Entrar a la carpeta que se creo al clonar con git el repositorio

```
>> cd Adafruit_Python_DHT
```

Instalar la librería DHT de Adafruit

```
>> sudo python setup.py install
```

Con esto la librería debería estar instalada y disponible para ser llamada desde cualquier programa en Python.

SENSOR DHT CONECTADO A RPI

PROGRAMACIÓN EN PYTHON

Ejecutamos el comando cd para volver al directorio inicial

```
>>cd
```

Creamos la carpeta proyectos, nos cambiamos a esa carpeta y luego clonamos el repositorio para este proyecto

```
>>mkdir proyectos
```

```
>>cd proyectos
```

```
>>git clone https://github.com/internetdelascosas/RaspberryPi-DHT11.git
```

```
>>cd RaspberryPi-DHT11
```

AL hacer un ls podemos ver que dentro de la carpeta hay tres archivos

dht_consola.py dht_log.py README.md

README.md es un archivo de texto que contiene información sobre el proyecto, este archivo es el que permite crear la página web que se visualiza al entrar a la pagina del proyecto

en GitHub <https://github.com/internetdelascosas/RaspberryPi-DHT11>

dht_consola.py es el programa escrito en Python que ejecutaremos primero usando el comando

```
>>sudo ./dht_consola.py
```

Si el sensor esta funcionando y esta bien conectado deberíamos obtener una salida como la siguiente

```
joniux@raspberrypi:~/proyectos/RaspberryPi-DHT11 $ sudo ./dht_consola.py
Temperatura=19.0* Humedad=37.0%
Temperatura=20.0* Humedad=37.0%
Temperatura=19.0* Humedad=37.0%
Temperatura=20.0* Humedad=37.0%
```

SENSOR DHT CONECTADO A RPI

PROGRAMACIÓN EN PYTHON

El programa se ejecutara en un ciclo infinito por lo que para terminar su ejecución debes presionar Control+C.

dht_log.py es un programa un poco mas avanzado, básicamente hace exactamente lo mismo que el programa *dht_console.py* pero en vez de mostrar las variables en la consola, las escribe a un archivo de log en `/var/log/iot/`, al igual que el programa anterior, puedes ver el [código fuente](#) en nuestro repositorio GitHub.

El objetivo de este programa es que se ejecute en modo background, de tal forma que podamos dejar la Raspberry Pi encendida monitoreando constantemente las dos variables y registrandolas en el archivo log para un procesamiento posterior.

Para ejecutarlo primero debes crear la carpeta `/var/log/iot` que sera el lugar donde se escribirán los archivos log, este comando debe ser ejecutado como súper usuario con el comando `sudo`

```
ioniux@raspberrypi:~/proyectos/RaspberryPi-DHT11 $ sudo mkdir /var/log/iot
```

Y luego ejecutar el programa *dht_log.py* también como súper usuario con el comando `sudo`

```
ioniux@raspberrypi:~/proyectos/RaspberryPi-DHT11 $ sudo ./dht_log.py &
```

Respuesta: [1] 13278

El carácter `&` al final de la línea es muy importante porque este permite dejar la tarea ejecutándose en background, informando del numero de proceso, en este caso 13278 y devolviendo el control a línea de comandos para ingresar nuevos comandos.

SENSOR DHT CONECTADO A RPI

PROGRAMACIÓN EN PYTHON

Si ejecutas el comando ps, verás que existen dos procesos corriendo con el nombre dht_log

```
jonix@raspberrypi:~/proyectos/RaspberryPi-DHT11 $ 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
jonix   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.

```
jonix@raspberrypi:~ $ tail /var/log/iot/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: 38.0Si deseas terminar la ejecución el programa debes

“matar” los dos procesos que están corriendo, para matar un proceso se usa el comando 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

```
jonix@raspberrypi:~/proyectos/RaspberryPi-DHT11 $ kill 13350 13354
```