

Objetivos

- × Aprender a instalar Raspbian OS en Raspberry Pi
- × Familiarizarse con Linux y la ventana de comandos
- × Introducirse en el mundo de la electrónica y la programación con el lenguaje Python.

En resumen, es programar un sistema embebido a alto nivel con el lenguaje Python bajo un sistema operativo Linux Raspbian OS. 1.

Instalación de Raspbian OS





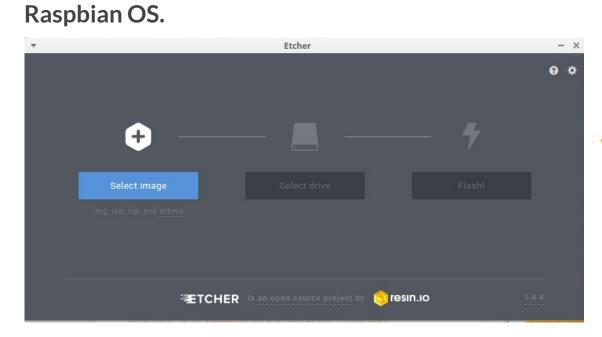


Raspbian OS
(.img o .zip)

MicroSD

Mínimo - 4GB Recomendable -8GB in which the

Paso 1. Seleccionamos el archivo de



Atención: en Linux es necesario dar permisos a la aplicación para ejecutarse.



Paso 2. Seleccionamos la tarjeta

MicroSD.



Atención: Etcher suele autodetectar la tarjeta SD o el usb, por tanto, tener cuidado si teneis mas de uno conectado para formatear el deseado.

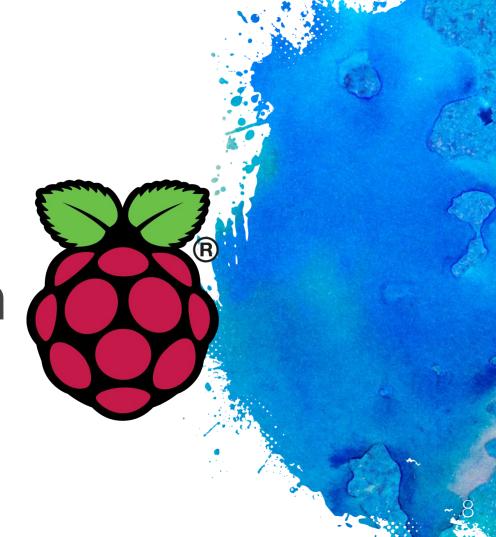


Que termine de instalar la imagen en la SD.

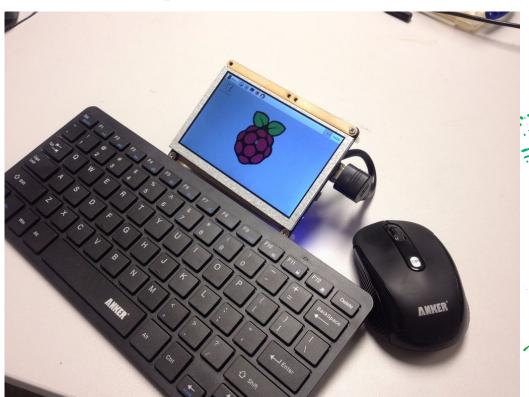


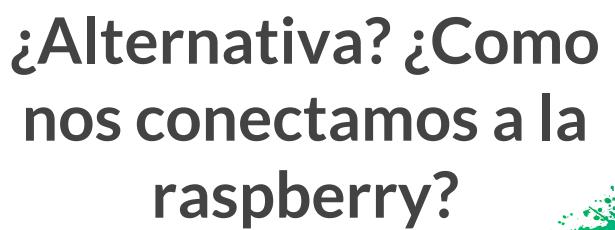


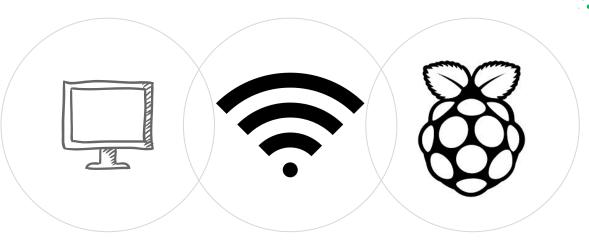
Conexión con Raspberry Pl



Llegados a este punto, normalmente disponemos de....











Activar la conexión wifi

Para conectarnos a la raspberry sin pantalla, debemos conectarnos a la conexión wifi.

Para ello, solo tienes que añadir un archivo que se llame 'wpa_supplicant.conf' en /boot.

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
network={
    ssid="Nombre de la wifi"
    psk="La contraseña de la wifi"
    key_mgmt=WPA-PSK
}
```



Activar IP Estática

Para facilitar la conexión remota a la Raspberry procederemos a darle una IP estática conocida para evitar que el DHCP nos asigne una IP y tengamos que buscarla.

Para ello, solo tienes que añadir un archivo que se llame 'cmdline.txt' en /boot.

ip=192.168.1.99



Conexión Remota. SSH - Secure Shell

SSH es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente. Además, encripta la sesión de conexión, haciendo imposible que alguien pueda obtener contraseñas no encriptadas.







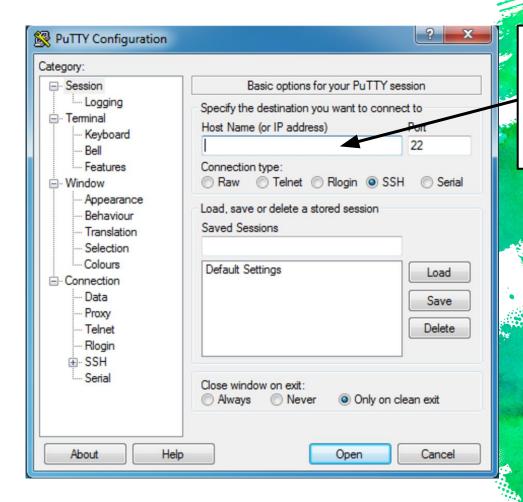
Activar la conexión por SSH. ¿Qué datos son necesarios para conectarnos?

Para conectarnos a la raspberry sin pantalla, debemos habilitar la conexión remota. Está desactivada por defecto por seguridad.

Para ello, solo tienes que añadir un archivo vacío sin extensión que se llame 'ssh' en /boot.

Las redes de ordenadores viene identificas por una ip. Está es necesaria para conectarnos a la raspberry pi. Además del usuario y la contraseña del ordenador remoto. Por defecto, en Raspberry son pi y raspberry, usuario y contraseña respectivamente.





Sino aparece la siguiente pantalla, seleccionamos la 'Category' - 'Session'.

Aquí introducimos la IP de la raspberry. Después pulsamos el botón Open y habremos realizado la conexión.

WINDOWS

Para windows, en la mayoría de versiones no incluye un cliente para conectarse de forma remota por ssh, por tanto, debemos descargar un programa.

P.e. -> Putty



En Mac, podemos encontrar el terminal en el Finder -> Aplicaciones -> Utilidades -> Terminal.

MacOS

MacOs incorpora un cliente ssh via terminal con el cual podemos acceder de forma remota a la raspberry.



chris — -bash — 101×24

Last login: Sun Jun 11 14:41:51 on ttys000 Chriss-MacBook-Air:~ chris\$ ssh chris@ssh.example.com

Aqui introducimos:

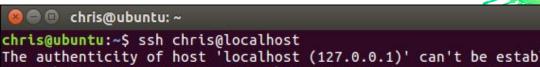
ssh user@ssh.server.com

Este comando sirve para conectarnos a un servidor ssh, en este caso, la raspberry pi. En el servidor tenemos un usuario que corresponde a user y disponemos de una IP o dirección a la que conectarse al servidor. Como resultado obtendremos, donde la IP suele ser la IP de una red local como 192.168.1.X:

ssh pi@IP

MacOS

MacOs incorpora un cliente ssh via terminal con el cual podemos acceder de forma remota a la raspberry.



The authenticity of host 'localhost (127.0.0.1)' can't be established. ECDSA key fingerprint is SHA256:k9RDuVinuWTILvr+10g2ADSwY7qVnEwTS3BeVGlW Are you sure you want to continue connecting (yes/no)?



chris@ubuntu:~\$ ssh chris@localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:k9RDuVinuWTILvr+10g2ADSwY7qVnEwTS3BeVGlW
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known host
chris@localhost's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-54-generic x86 64)

Al igual que en la diapositiva anterior tenemos, el usuario, en este ejemplo 'chris' @ IP, donde en este caso, localhost hace referencia a la misma computadora empleada, en nuestro caso sera la IP de la raspberry pi.

En Linux, podemos encontrar el terminal en el buscador de aplicaciones. Normalmente está ubicado en aplicaciones de menu.

Linux

Linux incorpora un cliente ssh via terminal con el cual podemos acceder de forma remota a la raspberry.

Terminal Linux

```
pi@raspberrypi:/ $ cd /home/pi
pi@raspberrypi: ~ $ pwd
/home/pi
pi@raspberrypi:~ $ ls -1
total 36
drwxr-xr-x 2 pi pi 4096 Jul 5 12:01 Desktop
drwxr-xr-x 6 pi pi 4096 Sep 13 16:38 Documents
drwxr-xr-x 2 pi pi 4096 Jul 5 12:01 Downloads
drwxr-xr-x 2 pi pi 4096 Jul 5 12:01 Music
drwxr-xr-x 2 pi pi 4096 Jul 5 12:01 Pictures
drwxr-xr-x 2 pi pi 4096 Jul 5 12:01 Public
drwxr-xr-x 2 pi pi 4096 Jul 5 11:29 python_games
drwxr-xr-x 2 pi pi 4096 Jul 5 12:01 Templates
drwxr-xr-x 2 pi pi 4096 Jul 5 12:01 Videos
pi@raspberrypi:~ $
```

Comandos Básicos de Linux

¿Que es un comando?

Un comando en su mayor parte más pequeños programas incorporados en el sistema operativo.

Comandos:

- cat - mv

- top

passwd

- Is

rm

find

reboot

- cd

mkdir

. clear

pwd

SU

- halt

- cr

kill

exit

man

- shutdown

21

Editor nano

Crear o abrir un archivo:

nano [nombre del archivo+extension]

Cerrar y guardar archivos:

Ctrl + o # Guardar

Ctrl + x # Cerrar

Enlace:

https://www.luisllamas.es/raspberry-pi-nano/



4.
Python

Hello-World!

La siguiente línea se copia en un fichero con la extensión '.py':

Este programa imprime Hello, world! - con # generamos un comentario.

print('Hello, world!')



¿Que es el PIP?

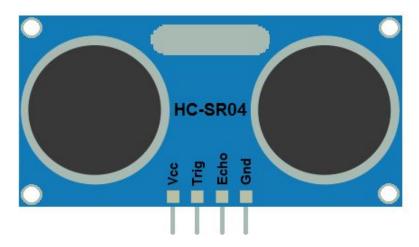
pip es un sistema de gestión de paquetes utilizado para instalar y administrar paquetes de software escritos en Python.



Instalación del PIP y librerías python

sudo apt-get update sudo apt-get install python3-pip sudo pip3 install RPi.GPIO





5.

Sensores





PINOUT RPY 3B+

Pin#	NAME		NAME	Pin
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1, I2C)	00	DC Power 5v	04
05	GPIO03 (SCL1, I ² C)	00	Ground	06
07	GPIO04 (GPIO_GCLK)	00	(TXD0) GPIO14	08
09	Ground	00	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	00	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	00	Ground	14
15	GPIO22 (GPIO_GEN3)	00	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	00	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	00	Ground	20
21	GPIO09 (SPI_MISO)	00	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	00	(SPI_CE0_N) GPIO08	24
25	Ground	00	(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)	00	(I ² C ID EEPROM) ID_SC	28
29	GPIO05	00	Ground	30
31	GPIO06	00	GPIO12	32
33	GPIO13	00	Ground	34
35	GPIO19	00	GPIO16	36
37	GPIO26	00	GPIO20	38
39	Ground	00	GPIO21	40
	Raspberry P	i 3 B+	PoE Header	
01	TR01	00	TR00	02
03	TR03	00	TR02	04



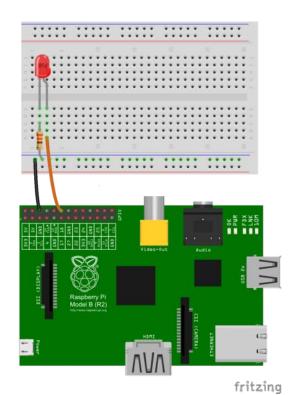
EJEMPLO 1 - ENCENDER UN LED

COMPONENTES:

- Un LED
- Una resistencia de 330 Ohm
- Varios cables
- Una raspberry pi
- Y espiritu maker!



ESQUEMA ELECTRICO





#!/usr/bin/env python3
import time
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)
GPIO.setup(3, GPIO.OUT)

while True:

GPIO.output(3, GPIO.HIGH)

time.sleep(0.5)

GPIO.output(3, GPIO.LOW)

time.sleep(0.5)



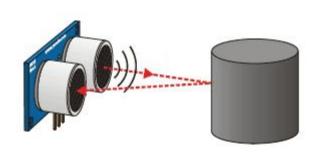
EJEMPLO 2 - MEDIR DISTANCIA CON UN ULTRASONIDOS

COMPONENTES:

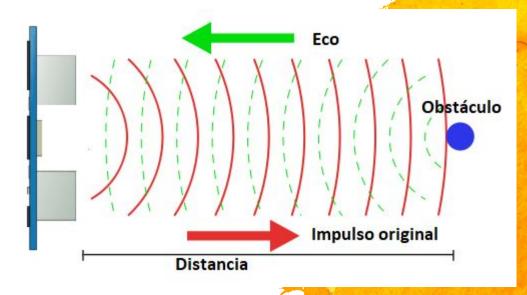
- Un ultrasonidos HC-SR04
- Una resistencia de 330 Ohm
- Una resistencia de 470 Ohm
- Varios cables
- Una raspberry pi
- Y espiritu maker!



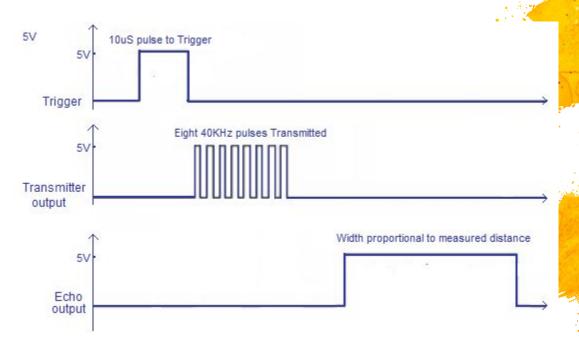
Principio físico HC-SR04



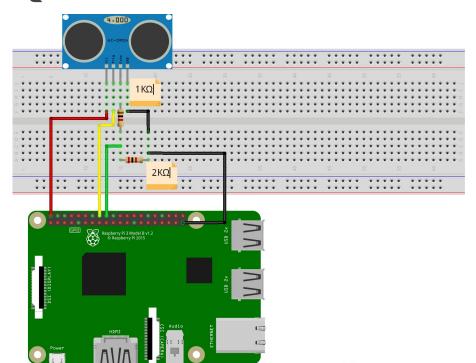
Tiempo = 2 * (Distancia / Velocidad) Distancia = Tiempo · Velocidad / 2



Cómo funciona el HC-SR04



ESQUEMA ELECTRICO





#!/usr/bin/env python3
Libreries
import RPi.GPIO as GPIO
import time

GPIO Mode (BOARD / BCM)
GPIO.setmode(GPIO.BCM)

GPIO Pins

GPIO_TRIGGER = 2

GPIO_ECHO = 3

GPIO direction (IN / OUT) GPIO.setup(GPIO_TRIGGER, GPIO.OUT) GPIO.setup(GPIO_ECHO, GPIO.IN)

Function

def distance():

set Trigger to HIGH

set Trigger after 0.01ms to LOW time.sleep(0.00001)

GPIO.output(GPIO_TRIGGER, False)

GPIO.output(GPIO_TRIGGER, True)

StartTime = time.time()
StopTime = time.time()

save StartTime while GPIO.input(GPIO ECHO) == 0: StartTime = time.time() # save time of arrival while GPIO.input(GPIO ECHO) == StopTime = time.time() # time difference between start and arrival TimeElapsed = StopTime - StartTime # multiply with the sonic speed (34300 cm/s) # and divide by 2, because there and back distance = (TimeElapsed * 34300) / 2 return distance

```
if __name__ == '__main__':
 try:
   while True:
     dist = distance()
     print ("Measured Distance = %.1f cm" % dist)
     time.sleep(1)
   # Reset by pressing CTRL + C
 except KeyboardInterrupt:
   print("Measurement stopped by User")
   GPIO.cleanup()
```



PROPUESTA-MEDIR DISTANCIA CON **UN ULTRASONIDOS Y ENCENDER UN LED SI LA** DISTANCIA ES MENOR A 15 CM.



Extra: Systemd

- Permite correr un programa o script en segundo plano
- × Gestiona el inicio automático programa

Info más detallada en:

https://wiki.archlinux.org/index.php/Systemd_(Español)

Crear servicio

- 1. Crear el archivo del demonio o servicio: nano /lib/systemd/system/blink.service
- 2. Actualizar el systemd: sudo systemctl daemon-reload
- 3. Iniciar el servicio: sudo systemctl enable blink sudo systemctl start blink
- 4. Para ver el estado del servicio o demonio: sudo systemctl status blink



/lib/systemd/system/blink.service

[Unit]

Description=blink.py - LED Blink service that runs in the background Documentation=https://example.com

[Service]

Environment=INTERVAL=1

Type=simple

User=pi

ExecStart=/usr/bin/python3 /home/pi/blink.py

Restart=on-failure

[Install]

WantedBy=multi-user.target

/lib/systemd/system/blink.service



¿Alguna pregunta?

Puedes encontrarme en:

@rafa.carbonell - Instagram

: racarla96@gmail.con

makersupv@gmail.com

github.com/makers-upv/taller-raspberry