

# Introducción a Raspberry Pi

Alma González

## SEGUNDA CLASE

Puede conectarse a la Raspberry Pi a través de Internet desde otra computadora o un dispositivo móvil. Hay varias maneras de hacer esto. [remot3.it](http://remot3.it), de Weaved, Inc. es un software que instalas en tu Raspberry Pi para acceder a una sola Pi, o administrar un gran número de microcomputadoras, desde cualquier lugar vía Internet. Pueden utilizar remot3.it para acceder a cualquier puerto TCP en su Pi a través de la Web incluyendo SSH, VNC, HTTP (S), RDP y servicios TCP personalizados. El instalador de remot3.it te pedirá crear una cuenta de usuario introduciendo una dirección de correo electrónico y una contraseña en [remot3.it](http://remot3.it)

Desde la línea de comandos de tu Pi, actualiza tus listas de paquetes Raspbian:

```
sudo apt-get update
```

Instala el paquete remot3.it:

```
sudo apt-get install weavedconnectd
```

A continuación, ejecuta el instalador remot3.it con este comando:

```
sudo weavedinstaller
```

La siguiente pantalla que debe verse es la siguiente, si eso no pasa, ve más adelante, a la siguiente parte, de actualización.

```
pi@raspberrypi:~$ sudo weavedinstaller

remot3.it connection installer Version: v1.3-07_Pi lib_v1.3-07_Pi
Modified: August 22, 2016 (library) August 24, 2016

Checking your network for compatibility...

Your network is compatible with remot3.it services.

***** Sign In Menu *****

1) Sign in to your existing remot3.it account
2) Request a code for a new remot3.it account
3) Enter a verification code received in e-mail
4) Exit

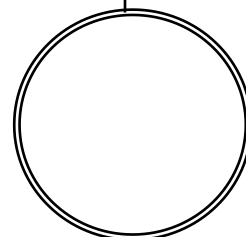
*****

Use your Weaved account with remot3.it!
*****
```

En este menú debes iniciar de sesión en la ventana del terminal. Ingresa a la opción de menú "1" para iniciar sesión en tu cuenta remot3.it. Si aún no tienes una cuenta remot3.it puede crear una introduciendo el número "2" en el menú.

```
Please select from the above options (1-4):
1
Please enter your remot3.it Username (e-mail address):
jim@remot3.it ← Enter your remot3.it (or Weaved) account email

Please enter your remot3.it password:
..... ← Enter your remot3.it (or Weaved) account password
```



Después de iniciar sesión, pedirá que introduzcas un nombre de dispositivo para la RBPi (por ejemplo, mi\_Pi\_001) como se muestra a continuación.

```
Enter a name for your device (e.g. my_Pi_001).
The Device Name identifies your device in the remot3.it portal.
Your services will be grouped under the Device Name.

Only letters, numbers, underscore, space and dash are allowed.
my_Pi_001      enter a device name
.
Registering my_Pi_001.....

Updating /etc/weaved/services/Weavedrmt365535.conf
.
===== Installed remot3.it Services =====
Protocol      Port      Service      remot3.it Service Name
-----
Device Name: my_Pi_001
=====
```

Después, en el siguiente menú selecciona el número 1 para Adjuntar / reinstalar remot3.it a un Servicio.

```
***** Main Menu *****

1) Attach/reinstall remot3.it to a Service
2) Remove remot3.it attachment from a Service
3) Remove all remot3.it attachments, then exit
4) Exit

*****

Please select from the above options (1-4):
1      enter "1" to attach remot3.it to your services on your Pi like ssh, vnc, etc
```

Aparecerá el menú de selección de protocolos.

```
***** Protocol Selection Menu *****

1) SSH on port 22
2) Web (HTTP) on port 80
3) VNC on port 5901
4) Custom (TCP)
5) Return to previous menu

*****

You can change the port value during install

*****

Please select from the above options (1-5):
```

Selecciona el elemento de menú deseado para habilitar los servicios TCP de Pi (SSH, por ejemplo) para acceso remoto. Recomendamos instalar remot3.it para SSH en el puerto 22 para comenzar. (Elemento de menú "1"). Cuando haya terminado de añadir servicios a remot3.it ingrese "5" (Regrese al menú anterior) y salga de los menús del instalador.

Después de ello puedes conéctese a la RB Pi usando remot3.it. En <https://www.remot3.it> ingresas con tu cuenta (la misma que logeaste en la consola de la RB).

Verás la Pi en la pestaña "Administrar dispositivos". Haz clic en "Nombre del dispositivo" para conectarte

Si no aparece la primer pantalla y más importante si después de ingresar el resto de los pasos la RB no aparece en el Administrador de dispositivos es que algo salió mal. Verifica la revisión del paquete `weavedconnectd` debian existente en su Pi. Antes de hacer cualquier otra cosa, asegúrese de que su paquete actual sea la versión 1.3-02. Lo que sucedió en la clase fue que el instalador de Raspbian no descargó la última versión, la necesaria para hacer la conexión, entonces eso debe hacerse de manera manual.

- Abre una ventana de consola de comandos en la Pi utilizando
- Descarga e instala el paquete `weavedconnectd` que incluye soporte para `remot3.it` ingresando el siguiente comando.

```
sudo wget https://github.com/weaved/installer/raw/master/Raspbian%20deb/1.3-07/weavedconnectd_1.3-07v_ar_mhf.deb
```

Instale el paquete deb nuevo utilizando el siguiente comando. Asegurate de copiar/pegar el comando exactamente como se indica a continuación. Este comando asegura que el nuevo paquete deb se instale por completo. El comando activará el reemplazo del software Weaved actual en su Pi con una nueva versión que sea compatible con `remot3.it`.

```
sudo nohup dpkg -i weavedconnectd_1.3-07v_ar_mhf.deb 2>&1 &
```

## Editores de Texto

En Linux, hay una selección de editores de texto.

Algunos son fáciles de usar pero tienen funcionalidad limitada; Otros requieren bastante práctica para usarlos y tomar un cierto tiempo para dominar, pero ofrecen una funcionalidad increíble.

### Nano

El uso de una Pi requiere algunos conocimientos básicos de un editor de texto basado en terminal como **Nano** para la programación o el cambio de archivos del sistema. Básicamente, sólo los pocos comandos son necesarios.

Si necesito editar archivos de texto directamente en la RB Pi el editor de texto de elección es nano. Hay otros editores de texto disponibles. Como una utilidad basada en la línea de comandos puede parecer extraño para los usuarios que están más familiarizados con una interfaz gráfica, pero es fácil aprender los conceptos básicos.

Para iniciar nano simplemente puede escribir `nano` en la ventana de comandos. Esto se iniciará con un "nuevo búfer". En otras palabras, un archivo de texto vacío que no tiene nombre.

Para iniciar un archivo de texto existente escriba nano seguido del nombre de archivo:

```
nano archivo.py
```

Si su archivo no está en el directorio actual, deberá especificar la ruta de acceso completa:

```
nano /home/pi/Desktop/archivo.py
```

Si el archivo es un archivo de sistema, puede que tenga que utilizar permisos elevados:

```
sudo nano -w /etc/fstab
```

Nota: Utiliza siempre el modificador -w al abrir los archivos del sistema. Deshabilita las líneas largas y asegura que el archivo no se modifica de manera que pueda afectar su sistema.

Nano se controla con atajos de teclado. Los más importantes son

<code>CTRL + G</code>	Visualizar el texto de ayuda
<code>CTRL + X</code>	Cerrar el búfer de archivo actual / Salir de nano
<code>CTRL + O</code>	Escribir el archivo actual en el disco
<code>CTRL + R</code>	Insertar otro archivo en el actual
<code>CTRL + W</code>	Buscar una cadena o una expresión regular
<code>CTRL + \</code>	Reemplazar una cadena o una expresión regular
<code>CTRL + K</code>	Cortar la línea actual y almacenarla en el cutbuffer
<code>CTRL + U</code>	Pegar desde el cutbuffer en la línea actual
<code>CTRL + C</code>	Muestra la posición del cursor
<code>CTRL + _</code>	Ir al número de línea y columna
<code>CTRL + T</code>	Invoke el corrector ortográfico, si está disponible

## IDLE

IDLE es un Python REPL "**Read-Eval-Print-Loop**", también conocido como alto nivel interactivo o consola de lenguaje, toma las entradas individuales del usuario, las evalúa y devuelve el resultado al usuario; un programa escrito en un entorno REPL es ejecutado parte por parte. Junto con un IDE Integrated Development Environment, por lo que puede escribir y editar código Python en una ventana y ejecutar desde allí.

IDLE tiene ventanas independientes y resaltado de sintaxis. Puede usar atajos de teclado como Ctrl + S para guardar un archivo, o Alt + P (comando anterior) y Alt + N (siguiente comando) en la REPL. Ten en cuenta que IDLE utiliza Python 2 e IDLE 3 utiliza Python 3.

## PYTHON

Python es un lenguaje de programación cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Fue creado a finales de los ochenta por Guido van Rossum en el Centro para las Matemáticas y la Informática (CWI, Centrum Wiskunde & Informatica), en los Países Bajos

La sintaxis de Python es muy limpia, con énfasis en la legibilidad y utiliza palabras clave estándar en inglés. Comience abriendo IDLE desde el escritorio.

Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).

Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programable.

### Filosofía de Python

### Basada en Unix

- Bello es mejor que feo.
- Explícito es mejor que implícito.
- Simple es mejor que complejo.
- Complejo es mejor que complicado.
- Plano es mejor que anidado.
- Disperso es mejor que denso.
- La legibilidad cuenta.
- Los casos especiales no son tan especiales como para quebrantar las reglas.
- Lo práctico gana a lo puro.
- Los errores nunca deberían dejarse pasar silenciosamente.
- A menos que hayan sido silenciados explícitamente.
- Frente a la ambigüedad, rechaza la tentación de adivinar.
- Debería haber una -y preferiblemente sólo una- manera obvia de hacerlo.
- Aunque esa manera puede no ser obvia al principio a menos que usted sea holandés

## Idéntar

ÇÉÑ=Ñ~À çêá~ãEñF W

====~ëëÉëï ñ Z=M~à Çñ=B=N=ZZ=M=?ñ=ÇÉÄÉ=ëÉê ã=Éáí Éêç~ã~óçê=ç=á Ö ~ä~=MK?

====áÑ=ñ=ZZ= MW

=====êËì êâ=N

====ÉäëÉ W

=====êÊì êâ=ñ=G= Ñ~Á çêá~äEñ=J=NF

Debido al significado sintáctico de la indentación, cada instrucción debe estar contenida en una sola línea. No obstante, si por legibilidad se quiere dividir la instrucción en varias líneas, añadiendo una barra invertida \ al final de una línea, se indica que la instrucción continúa en la siguiente.

ääëí ~Zx D ~äçê=NDI D ~äçê=ODI D ~äçê=PDz

==Ä~ÇÉâ~Z Dbëí ç=Éë=ì â~=Ä~ÇÉâ~=Ä~ëí ~ái É=ä~ê Ö~D

ääëí ~Zx D ~äçê=NDI D ~äçê=OD=y

=====I D ~äçê=PDz

==Ä~ÇÉâ~Z Dbëí ç=Éë=ì â~=Ä~ÇÉâ~=D=y

=====DÄ~ëí ~ái É=ä~ê Ö~D

## Comentarios

Los comentarios se pueden poner de dos formas. La primera y más apropiada para comentarios largos es utilizando la notación ''' comentario ''', tres apóstrofes de apertura y tres de cierre. La segunda notación utiliza el símbolo #, y se extienden hasta el final de la línea.

El intérprete no tiene en cuenta los comentarios, lo cual es útil si deseamos poner información adicional en el código. Por ejemplo, una explicación sobre el comportamiento de una sección del programa.

DDD

`çãÉâí ~êáç=ã! ë=ä~ê Öç=Éâ=ì â~=ä-äÉ~=Éâ= mól Üçâ

DDD

éêää ?eçä~ã! âÇç?= @q~ãÄá Ýâ=Éë=éçëäÄäÉ=³ ~Çäé÷â=ÄçãÉâí ~éáç~ä=Ñââ~ä=ÇÉ÷â~ä-äÉ=ÇÉ= Ä ¼Çá Öç

## Variables

Las variables se definen de forma dinámica, lo que significa que no se tiene que especificar cuál es su tipo de antemano y puede tomar distintos valores en otro momento, incluso de un tipo diferente al que tenía previamente. Se usa el símbolo = para asignar valores.

ñ=Z=N

ñ=Z=?í Éñí ç?= @bëí ç=Éë=éçëäÄäÉ=éçëèí É=äçë÷ áéçë=ëçâ~éá Öâ~Ççë=Çáâ ääÄ~äÉâí É

## USO BÁSICO DE PYTHON

Algunos lenguajes usan corchetes { y } para marcar líneas de código que pertenecen juntas, y dejar que el escritor haga sangría para que estas líneas aparezcan anidadas visualmente. Sin embargo, Python no utiliza llaves, sino que requiere indentación para anidar. Por ejemplo un for en Python:

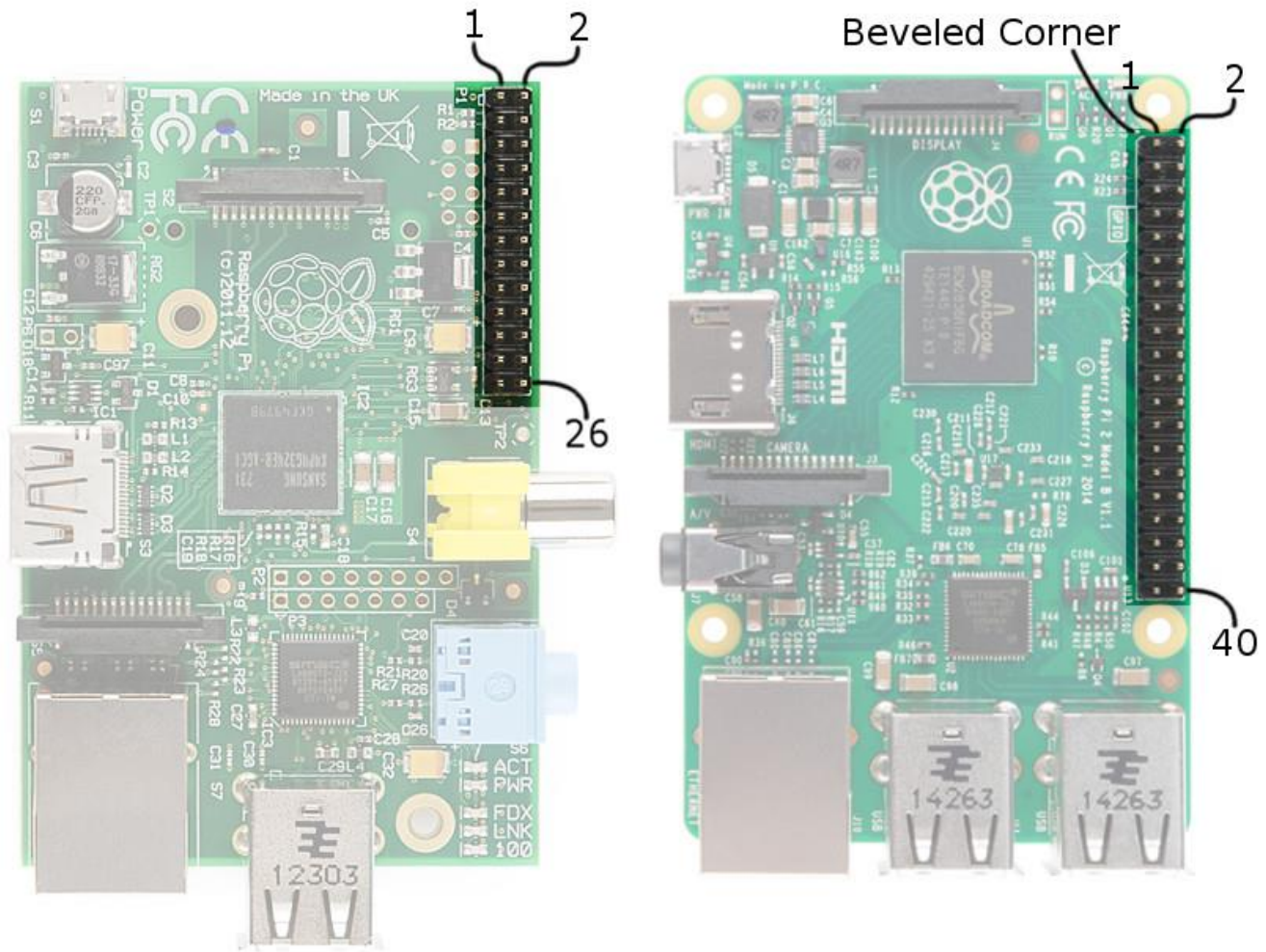
```
for i in range(10):
    print("Hello")
```

Algunos tipos de datos son iterables, lo que significa que puede realizar bucle sobre los valores que contienen. Por ejemplo una lista:



## GPIO

La Raspberry Pi ofrece sus GPIO en pines macho estándar en la placa. Con los años se han expandido de 26 pines a 40 pines manteniendo el pinout original.























La programación del hardware de la RB Pi referencia los pines con un único número único.

Existen (al menos) dos esquemas de numeración en los pines de la RB, cuando se hace referencia a números de pin Pi:

- (1) números específicos de pines de Broadcom
- (2) Pines físicos

Por lo general, puede utilizar cualquiera de los sistemas numéricos, pero muchos programas requieren que declare qué esquema está utilizando al principio de su programa.

Aquí hay una tabla que muestra los 26 pines en la cabecera P1, incluyendo cualquier función especial que puedan tener, y sus números duales:

Raspberry Pi2 GPIO Header				
Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I <sup>2</sup> C)		DC Power 5v	04
05	GPIO03 (SCL1 , I <sup>2</sup> 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
<hr/>				
27	ID_SD (I <sup>2</sup> C ID EEPROM)		(I <sup>2</sup> C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Early Models

Late Models

Rev. 1  
26/01/2014

<http://www.element14.com>

Aquí hay una tabla que muestra los 26 pines de los primeros modelos de la Raspberry Pi, incluyendo cualquier función especial que puedan tener, y sus números duales: variables se definen de forma dinámica, lo que significa que no se tiene que especificar cuál es su tipo de antemano y puede tomar distintos valores en otro momento, incluso de un tipo diferente al que tenía previamente. Se usa el símbolo = para asignar valores. La Raspberry Pi no sólo le da acceso a los pines de I/O bidireccionales, sino también a la Serie (UART), I2C, SPI, e incluso algunos PWM ("salida analógica").

## PWM



La función **PWM como abreviatura de la modulación por ancho de pulsos**, algo que se ha convertido en una práctica habitual de los interruptores de potencia modernos, controlando la energía de inercia. Esta acción tiene en cuenta la modificación del proceso de trabajo de una señal de tipo periódico. Puede tener varios objetivos, como tener el control de la energía que se proporciona a una carga o llevar a cabo la transmisión de datos.

Las señales PWM son comúnmente usadas para el control de velocidad de motores DC (si decrementas el ciclo de trabajo sobre la señal de control del circuito de potencia que actúa sobre el motor el motor se mueve más lentamente), ajustar la intensidad de brillo de un LED, etc.

PW (Pulse Width) o ancho de pulso, representa al ancho (en tiempo) del pulso.

length/period (periodo), o ciclo, es el tiempo total que dura la señal.

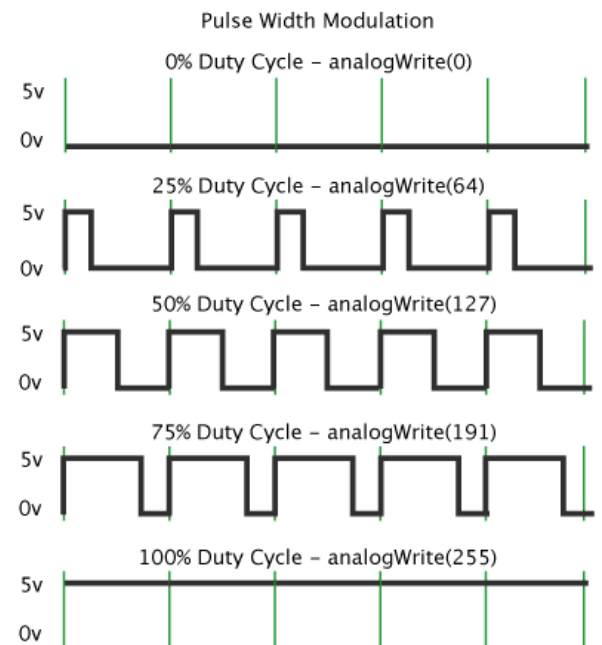
La frecuencia se define como la cantidad de pulsos (estado on/off) por segundo y su expresión matemática es la inversa del periodo, como muestra la siguiente ecuación.  $frecuencia = \frac{1}{periodo}$

El periodo se mide en segundos, de este modo la unidad en la cual se mide la frecuencia (hertz) es la inversa a la unidad de tiempo (segundos).

Existe otro parámetro asociado o que define a la señal PWM, denominado "Duty cycle", Ciclo de Trabajo, el cual determina el porcentaje de tiempo que el pulso (o voltaje aplicado) está en estado activo (on) durante un ciclo.

Por ejemplo, si una señal tiene un periodo de 10 ms y sus pulsos son de ancho (PW) 2ms, dicha señal tiene un ciclo de trabajo (duty cycle) de 20% (20% on y 80% off). El siguiente gráfico muestra tres señales PWM con diferentes "duty cycles".

La señal PWM se utiliza como técnica para controlar circuitos analógicos. El periodo y el ciclo de trabajo (duty cycle) del tren de pulsos puede determinar la tensión entregada a dicho circuito. Si, por ejemplo, tenemos un voltaje de 5v y lo modulamos con un duty cycle del 10%, obtenemos 0.5V de señal analógica de salida.



## UART

UART son las siglas de Universal Asynchronous Receiver Transmitter.

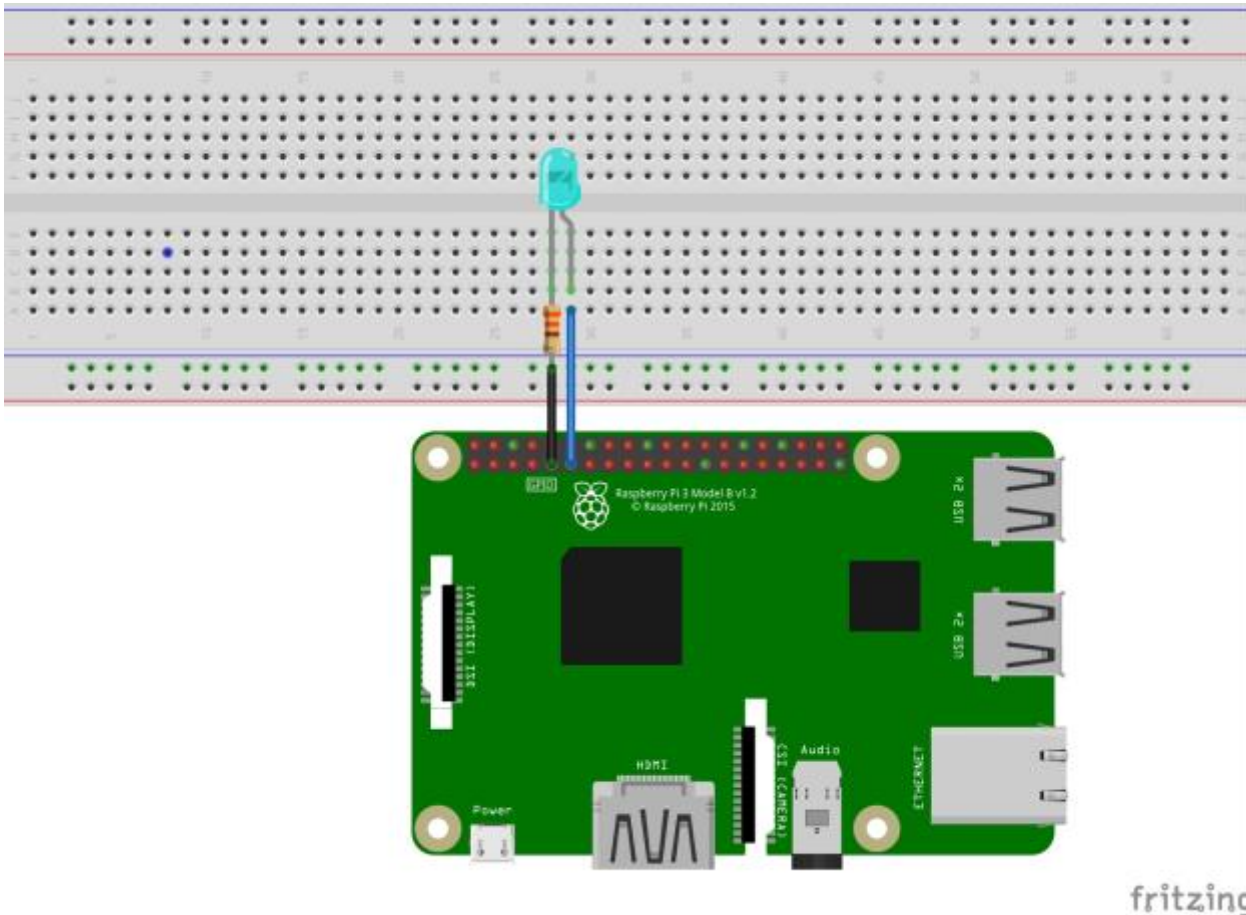
La comunicación serial permite a la Raspberry, a microcontroladores como Arduino comunicarse con una computadora o con otros dispositivos.

En términos de dispositivos en Linux, por defecto, / dev / ttyS0 indica Mini a la UART, y / dev / ttyAMA0 Se refiere a la PL011. El UART primario se Que asignado a la consola de Linux, que depende del modelo de Frambuesa Pi como se describió anteriormente, y se puede acceder a través de / dev / serial0.

Ambos son 3.3V dispositivos, lo que significa un cuidado extra debe tener cuidado al conectar a un puerto RS232 o de otro tipo Que sistema utiliza diferentes niveles de tensión. Un adaptador debe ser utilizado para convertir los niveles de tensión entre los dos protocolos. Como alternativa, los adaptadores USB 3.3V UART se pueden comprar a precios muy bajos.

El puerto serie ocupa los pines GPIO14 (Tx) y GPIO15 (Rx). Por defecto Raspbian utiliza estos pines para una consola serie. El SoC de Broadcom soporta dos UARTs pero ambas se configuran en los mismos pines de la cabecera J8. UART0 es un periférico estándar de ARM (PL011) e implementa todas las capacidades esperables en una UART de alto rendimiento.

Códigos de clase



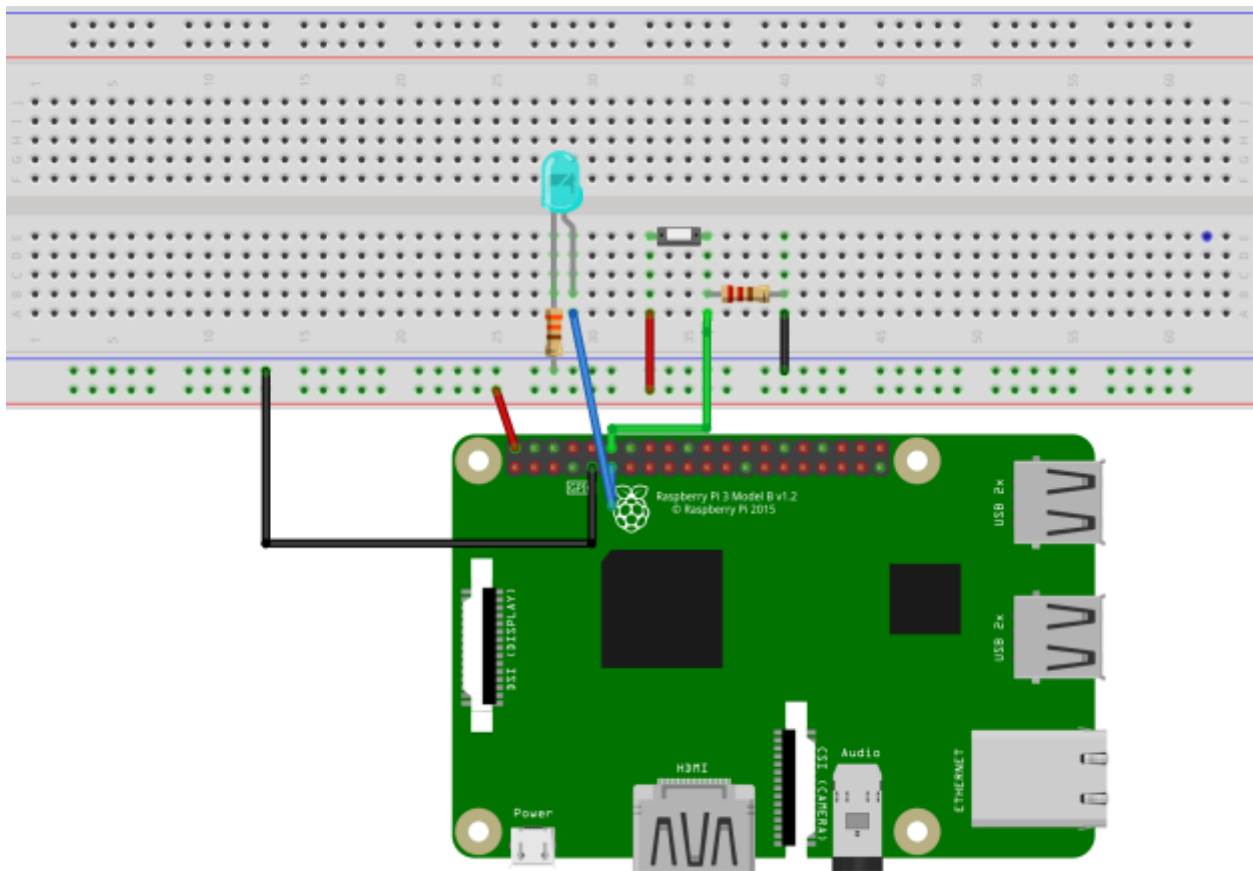
```
import RPi.GPIO as GPIO
import time

LED = 18 # Broadcom pin 18 (P1 pin 12)

# Pin Setup:
GPIO.setmode(GPIO.BCM) # Broadcom pin-numbering scheme
GPIO.setup(LED, GPIO.OUT) # LED pin set as output

# Initial state for LEDs:
GPIO.output(LED, GPIO.LOW)
try:
    while 1:

        GPIO.output(LED, GPIO.LOW)
        time.sleep(0.075)
        GPIO.output(LED, GPIO.HIGH)
        time.sleep(0.075)
except KeyboardInterrupt: # If CTRL+C is pressed, exit cleanly:
    GPIO.cleanup() # cleanup all GPIO
```



fritzing

```
import RPi.GPIO as GPIO
import time

pwmPin = 18 # Broadcom pin 18 (P1 pin 12)
butPin = 17 # Broadcom pin 17 (P1 pin 11)
dc = 5 # duty cycle (0-100) for PWM pin

# Pin Setup:
GPIO.setmode(GPIO.BCM) # Broadcom pin-numbering scheme
GPIO.setup(pwmPin, GPIO.OUT) # PWM pin set as output
GPIO.setup(butPin, GPIO.IN, pull_up_down=GPIO.PUD_DOWN) # Button pin set as input w/ pull-down

try:
    while 1:
        if GPIO.input(butPin): # button is released
            GPIO.output(ledPin, GPIO.HIGH)
            time.sleep(0.075)
        else:
            GPIO.output(ledPin, GPIO.LOW)
            time.sleep(0.075)
except KeyboardInterrupt: # If CTRL+C is pressed, exit cleanly:
    pwm.stop() # stop PWM
    GPIO.cleanup() # cleanup all GPIO
```

Con el mismo circuito

```
import RPi.GPIO as GPIO
import time

# Pin Definitons:
pwmPin = 18 # Broadcom pin 18 (P1 pin 12)
butPin = 17 # Broadcom pin 17 (P1 pin 11)

dc = 5 # duty cycle (0-100) for PWM pin

# Pin Setup:
GPIO.setmode(GPIO.BCM) # Broadcom pin-numbering scheme
GPIO.setup(pwmPin, GPIO.OUT) # PWM pin set as output
pwm = GPIO.PWM(pwmPin, 50) # Initialize PWM on pwmPin 100Hz frequency
GPIO.setup(butPin, GPIO.IN, pull_up_down=GPIO.PUD_DOWN) # Button pin set as input w/ pull-down

# Initial state for LEDs:
GPIO.output(ledPin, GPIO.LOW)
pwm.start(dc)

try:
    while 1:
        if GPIO.input(butPin): # button is released
            dc+=10
            time.sleep(0.075)
except KeyboardInterrupt: # If CTRL+C is pressed, exit cleanly:
    pwm.stop() # stop PWM
    GPIO.cleanup() # cleanup all GPIO
```

Estos deben de ser guardados como nuevos archivos de Python, fuera del Shell.

Puedes ejecutar los programas con F5

Si te aparece un error por permisos de administrador deberás hacerlo mediante la consola de comandos

Con cd te mueves a la carpeta que contiene el Archivo y escribes

```
sudo python blink.py
```

 donde blink.py es el nombre de tu archivo

Hay muchas maneras de conectar la Pi y Arduino, tales como el uso de los pines GPIO, Serial y utilizando I2C . Pero esto podría ser una de la manera más fácil para iniciar la charla, porque Que hardware necesario es mínimo: todo lo que necesita es un cable micro cable USB viene con el Arduino.

Vamos a enviar 'Hola' desde el Arduino a la Pi. Aquí está el código de Arduino.

```
int i=0;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  i++;
  Serial.print("Hello Pi ");
  Serial.println(i);
  delay(500);
}
```

En Python

```
serie de importaciones
ser = serial.Serial = ('/dev/ttyACM0', 9600)
```

El primer argumento - / dev / ttyACM0 es el nombre utilizado para la interfaz USB. Para averiguar el nombre del puerto, tenemos que ejecutar este comando en el terminal sin Arduino enchufado:

```
ls / dev / tty *
```

Ahora conecta el Arduino y ejecuta el comando. Si aparece el nombre nuevo, entonces este es el nombre de su puerto. También puedes verlo desde la IDE de arduino en la Pi, esto desde el menú superior Herramientas>Puerto> / dev /tty\*\*\*\*\*

El segundo argumento - la velocidad de transmisión es de 9600

```
While 1:
    ser.readline ()
```

Se necesitan dos pulsa enter dos veces después de escribir la segunda línea. Los mensajes 'Hola' ahora deben comenzar a aparecer. Puede pulsar Ctrl + C para detener (interrupción) del programa Python.

*Imágenes e información de esta guía son parte de la documentación oficial de Arduino Playground la cual se encuentra en la página <http://playground.arduino.cc>*

*Imágenes e información de esta guía son parte de la documentación oficial de Sparkfun Learn la cual se encuentra en la página <https://learn.sparkfun.com>*

*Esta obra está licenciada bajo la Licencia Creative Commons Atribución-CompartirIgual 4.0 Internacional. Para ver una copia de esta licencia, visita <http://creativecommons.org/licenses/by-sa/4.0/>.*

Alma González

[almagonzalez@engineer.com](mailto:almagonzalez@engineer.com)

[almagonzalezengineer.com](http://almagonzalezengineer.com)