Configurar servidor OctoPrint

Salva Gómez
SALVADOSPUNTOCERO

Contenido

Octoprint en Raspberry

Introducción

La finalidad del tutorial es que todo aquel que quiera hacer funcionar octoprint pueda hacerlo siguiendo los pasos que se detalla a continuación.

Decir que esto nace de recabar información y ajustarla a mis necesidades, todo lo he sacado de diversas fuentes, por tanto, no son mis méritos.

Para seguir el tutorial recomiendo una rpi3 ya que cuenta con más potencia que la rpi2, pero igualmente funcional. Tarjeta sd a poder ser de clase 10 y mínimo 8gb.

Descargar de sistema (raspbian)

Existe una imagen de octoprint lista para meter en la rpi y que sea funcional nada más arrancar el sistema. Es sencillo, pero me daba problemas para personalizar más cosas y añadirle funciones, por eso decidí poner raspbian, que, aunque lleva más tiempo y personalización no me dado ningún fallo.

- Para el que le interese la imagen de octoprint:

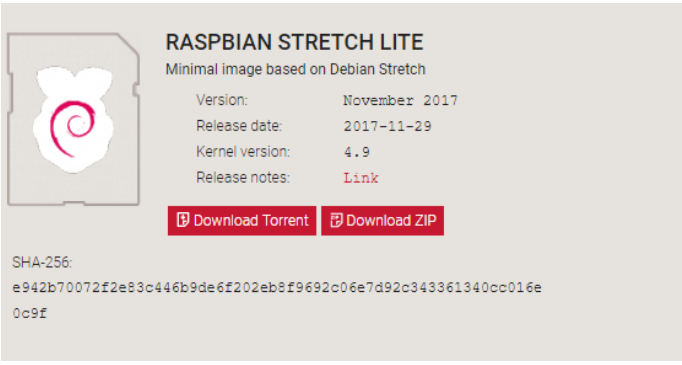
https://octoprint.org/download/

Es descargar la imagen y meterla en la tarjeta sd con cualquier programa de quemar imágenes, aquí mas adelante se va a explicar cómo se hace.

- Octoprint con raspbian:

Lo primero será descargar la imagen desde la página de raspbian, la última que haya y escoger la opción 'raspbian stretch lite'

https://www.raspberrypi.org/downloads/raspbian/

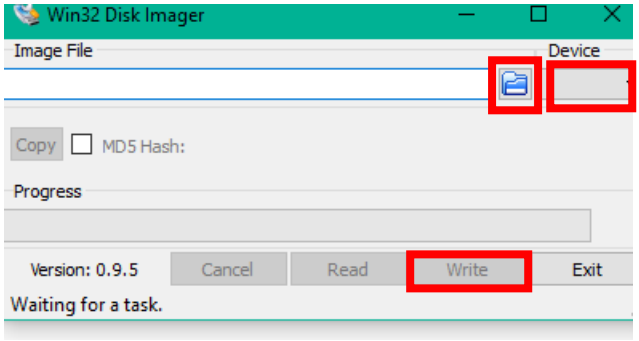


Instalación de imagen en tarjeta sd

Una vez descargada la imagen, hay que formatear la tarjeta sd en la que vamos a meter el sistema, yo utilizo 'sd formatter'. Una vez formateada se necesita un programa para escribir la imagen en la tarjeta sd.

Win32diskmanager: https://sourceforge.net/projects/win32diskimager/

Es muy sencillo la interfaz del programa y con un par de clicks se realiza la operación.



1. En el icono de la carpeta azul se selecciona la imagen de raspbian descargada anteriormente.
2. En 'device' se selecciona la tarjeta que se va a utilizar.
3. Dar a 'write' y empezara el proceso.

Iniciar sistema por primera vez

Terminado el paso anterior, hay que introducir la tarjeta en la rpi y encenderla, con esto el sistema empezara a cargarse, importante que esté conectada a internet mediante el cable RJ45 para poder configurar nuestro sistema remotamente.

Ahora tendremos que meternos con el navegador en nuestro router con la dirección (192.168.1.1) y ver qué dirección ip le ha asignado el router a la rpi.

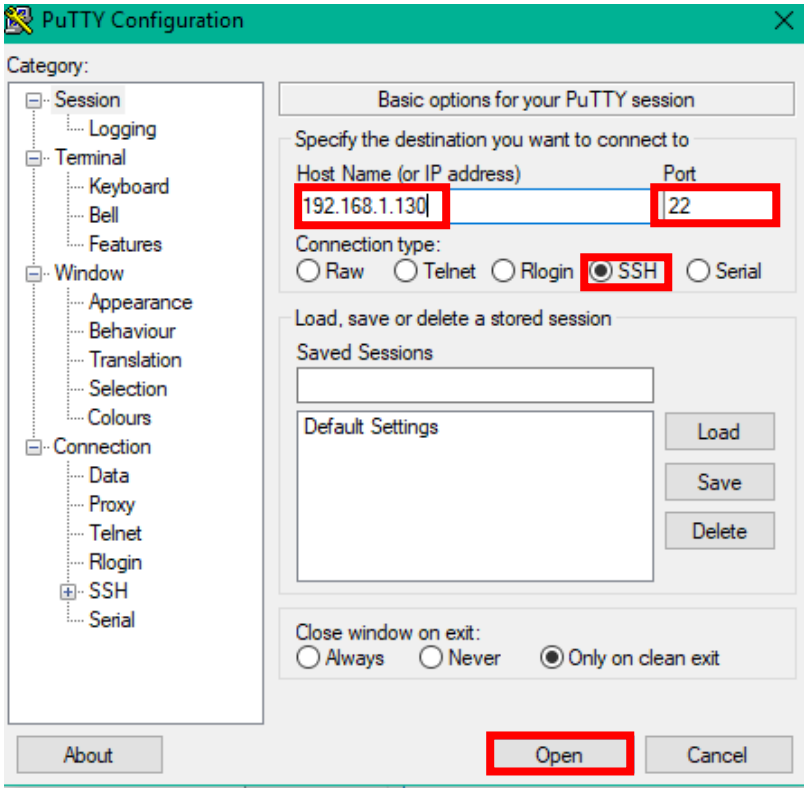
Si no entendéis bien como se hace buscar en internet como hacerlo, ya que hay muchos routers y será diferente para cada uno. En mi caso:

IP Address	Remaining Lease Time	Host Name	Port
192.168.1.130	153275	raspberrypi	LAN2
192.168.1.132	256247	DESKTOP-IB47	SSID5
192.168.1.138	5204		SSID5
192.168.1.137	229867	android-46dbf4	SSID5
192.168.1.133	140712	MI6-MiTel?fono	SSID5
192.168.1.131	211344	Susana-PC	SSID1

La ip asignada a mi rpi es esa y se utilizará para poder conectarnos a ella y poder configurarla.

Mediante el programa putty (www.putty.org) se hará la conexión por SSH.

Abrimos el programa y la interfaz es muy sencilla, solo hay que poner la ip que hemos mirado anteriormente en 'Ip address' y darle a intro.

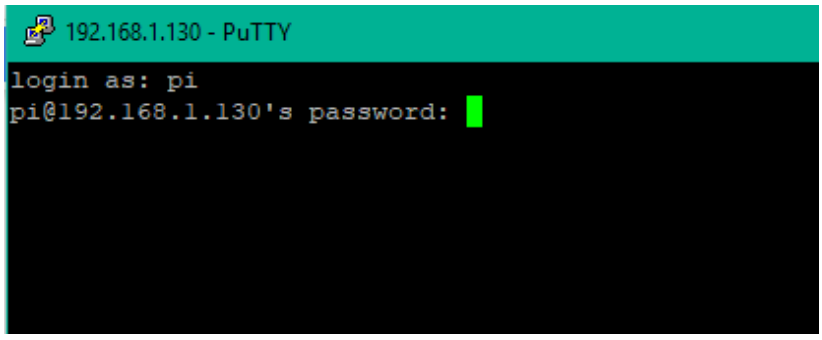


Una vez dado a open aparecerá un mensaje donde darás a aceptar y aparecerá una ventana en negro preguntando 'login as:' y 'password'.

Por defecto:

Login as: pi

Password: raspberry



Una vez metidas las credenciales, lo primero que hay que hacer al inciciar un sistema es actualizar, se hace con los siguientes comandos:

```
sudo apt-get update
sudo apt-get upgrade
```

Siempre que pegunte algo darle a 'y' y enter, para que siga actualizando, una vez termine el proceso hay que reiniciar con el comando y repetir el mismo proceso para conectarnos de nuevo mediante SSH.

```
sudo reboot
```

Asignar ip estática

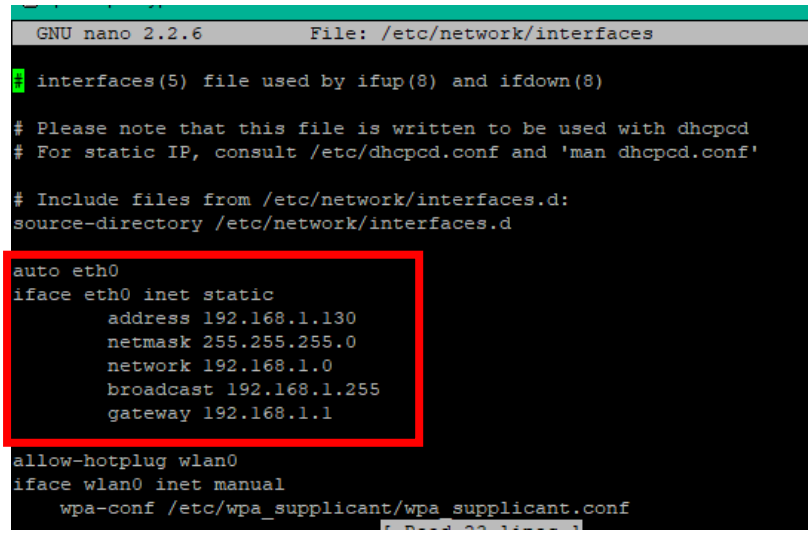
Una vez hemos vuelto a entrar por SSH a nuestra rpi vamos a hacer algunos pasos previos para luego el buen y correcto funcionamiento del servidor. Este paso es importante si tienes pensado acceder desde fuera de tu red local a la rpi o incluso para que a la hora de acceder por SSH siempre sea la misma dirección ip.

Lo que tenemos que hacer es poner el siguiente comando:

```
sudo nano /etc/network/interfaces
```

Ahora hay que editar este fichero, se abrirá con el editor de comandos nano y podremos cambiar y añadir líneas. En nuestro caso debemos localizar la línea que pone:

'iface eth0 inet dhcp'



Lo primero que hay que hacer es centrarse en el recuadro ese rojo que he señalado y en la línea auto borrar lo que haya y escribir ‘eth0’, dejarlo como en la imagen. Cambiamos la línea ‘iface eth0 inet dhcp’ por ‘iface eth0 inet static’.

Address: 192.168.1.130 (aquí pones la ip que quieras dejar como fija, en mi cado he dejado esa)

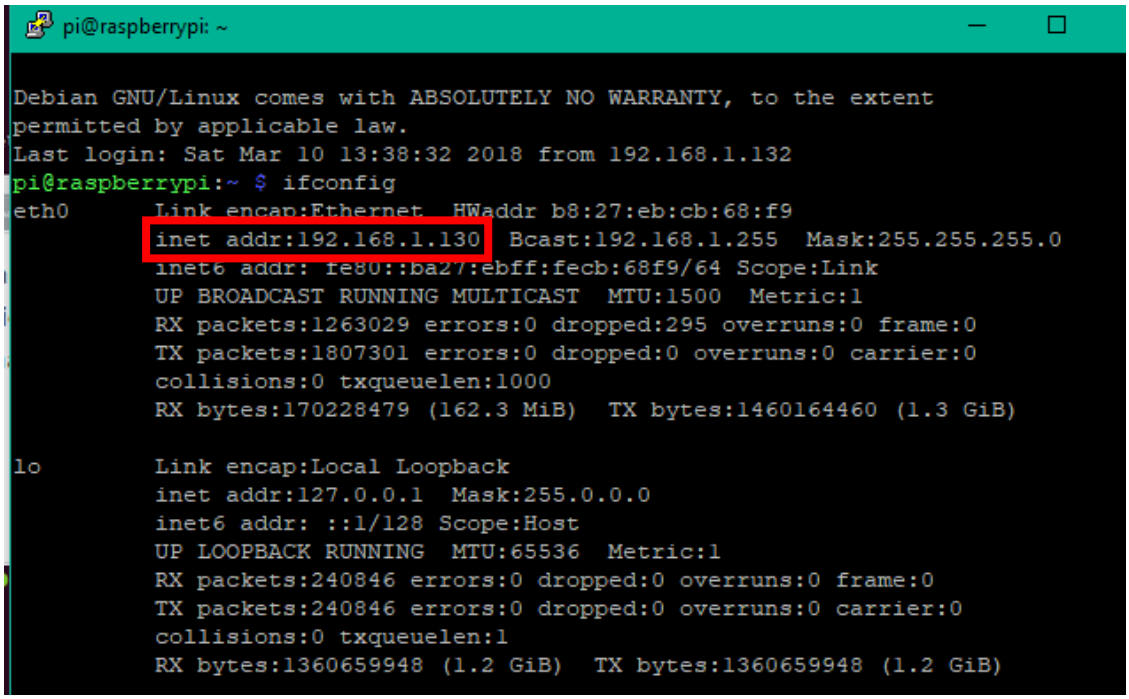
Las demás las dejamos tal cual en la foto y una vez hecho pulsamos CTRL+X y nos pedirá confirmación, pulsamos Y y enter y estará listo. Reiniciamos como antes, con el comando:

```
sudo reboot
```

Nos volvemos a conectar y tecleando el comando:

```
ifconfig
```

Nos debe aparecer la siguiente imagen, apareciendo la ip que le hemos asignado a nuestra rpi.



Cambiar contraseña de usuario

Este paso es importante para que solo podamos gestionar la rpi nosotros ya que al dejarla por defecto cualquiera con algo de conocimiento sabrá la password y por seguridad recomiendo no dejar la de por defecto.

Para este proceso haremos uso de un comando muy sencillo, que teclearemos en la consola una vez conectados por SSH.

```
passwd
```

Al dar enter nos pedirá que tecleemos la contraseña actual que es 'raspberry'

Una vez escrita nos pedirá dos veces que pongamos la nueva contraseña que vayáis a establecer y deberá aparecer al final el mensaje de cambio de contraseña efectuado.

Instalación octoprint

Ahora una vez situados en la consola de comandos, vamos a proceder a instalar octoprint, es ir metiendo líneas de código y dejar que se vayan procesando, es un proceso sencillo si se sigue al pie de la letra.

```
sudo apt-get install python-pip python-dev python-setuptools git libyaml-dev build-essential

sudo pip install virtualenv
```

Esperamos a que termine y ponemos lo siguiente:

```
git clone https://github.com/foosel/OctoPrint.git
```

Mediante la consola accedemos a la carpeta que acabamos de crear, con el comando:

```
cd OctoPrint
```

Una vez dentro de la carpeta procedemos con los siguientes comandos:

```
virtualenv venv
./venv/bin/pip install pip --upgrade
./venv/bin/python setup.py install
```

Cuando acabe, volvemos a nuestro directorio con el comando ‘cd’ y con esto nos situaremos fuera de la carpeta OctoPrint.

Por tanto, tecleamos:

```
cd
mkdir .octoprint
```

Con mkdir creamos una carpeta que se llamara ‘.octoprint’ y en la que se guardaran las configuraciones.

Acto seguido tecleamos como no hemos cambiado nuestro usuario, es ‘pi’ pues no tenemos que modificar nada, poner las líneas tal cual. Si alguno ha cambiado su nombre de usuario en donde pongamos ‘pi’ en las líneas de código el debería cambiarlas por el nombre de su usuario.

```
sudo usermod -a -G tty pi
sudo usermod -a -G dialout pi
```

Hecho esto podemos ver si vamos bien con la instalación ya que el siguiente comando permitirá ejecutar octoprint.

Por tanto, ejecutamos:

```
~/OctoPrint/venv/bin/octoprint
```

Saldrás unas líneas por consola y eso indica que el servidor se ha iniciado, para comprobarlo lo que haremos es ir al navegador y en la barra de direcciones poner la ip con un puerto predeterminado que este caso será el 5000.

Pues a comprobar, cada uno pondrá la ip que le asignó anteriormente yo pongo esa porque fue la que asigné.

http://192.168.1.130:5000

Aparecerá un asistente de octoprint de configuración inicial, el cual habrá que poner las medidas de impresora, nuestro nombre de usuario y contraseña para poder acceder a octoprint. Con eso configurado ya estamos listo para seguir, volvemos a la consola y tecleamos CTRL+C y se detendrá el servidor.

Una vez hecho hacemos:

```
sudo reboot
```

Instalación de webcam

Ahora debemos instalar los paquetes necesarios para que la webcam recoja los datos y los muestre por el servidor octoprint.

Yo tengo la cámara oficial de la rpi pero funcionan muchos modelos enchufados por usb es cuestión de probar y si no poner la oficial que ya os aseguro que funcionará.

En consola introducir los siguientes comandos para descargar y compilar MJPG-Streamer:

```
cd

sudo apt-get install subversion libjpeg62-turbo-dev imagemagick ffmpeg libv4l-dev cmake

git clone https://github.com/jacksonliam/mjpg-streamer.git

cd mjpg-streamer/mjpg-streamer-experimental

export LD_LIBRARY_PATH=.

make
```

Si todo va bien debemos poder iniciar el servidor de la cámara con el siguiente comando:

```
./mjpg_streamer -i "./input_raspicam.so -fps 5" -o
```

Ahora iremos al navegador y pondremos en la barra de direcciones la ruta y deberíamos poder ver la imagen que da la webcam. Como antes pongo mi dirección ip que asigne a mi raspberry.

```
http://192.168.1.130:8080/?action=stream
```

Una vez verificado que funciona, nos vamos a la consola y CTRL+C y detenemos el servicio. Nos vamos la carpeta '.octoprint' que creamos y abrimos el archivo config.yaml y lo editamos con sudo nano.

```
nano /home/pi/.octoprint/config.yaml
```

Una vez abierto el archivo pegamos el siguiente texto, como siempre con vuestra ip, yo pongo la mía.

```
webcam:
  stream: http://192.168.1.201:8080/?action=stream
  snapshot: http://127.0.0.1:8080/?action=snapshot
  ffmpeg: /usr/bin/avconv
```

Para guardar y salir CTRL+X y Y para aceptar el cambio.

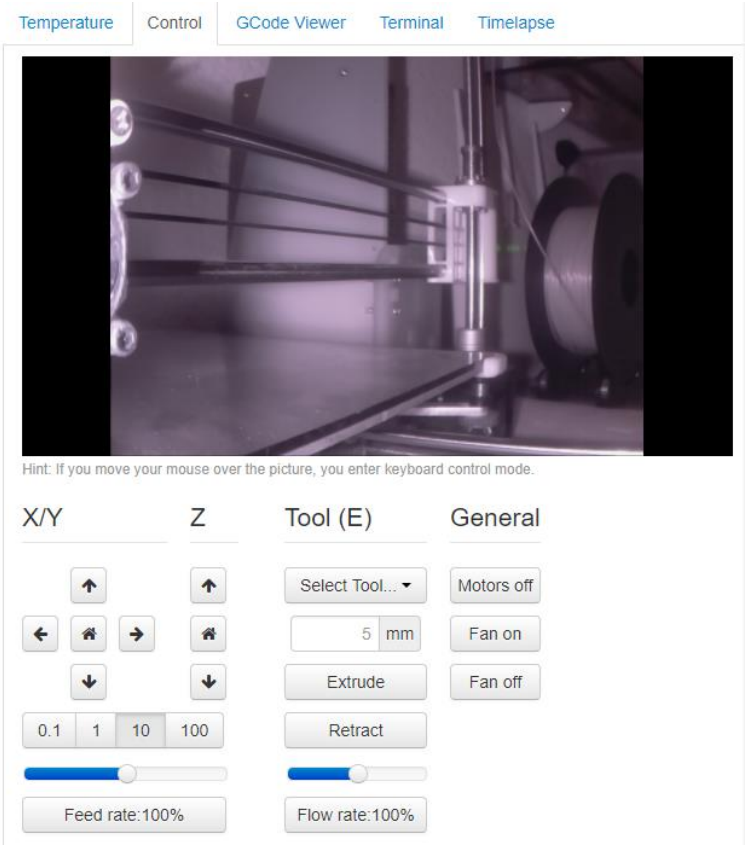
Ahora reiniciando como hemos hecho mas veces con ‘sudo reboot’ y volviendo a iniciar el servidor octoprint debiéramos de ver en la pestaña control la imagen de la cámara web.

Para comprobar esto debemos ejecutar los siguientes comandos:

```
sudo reboot

~/OctoPrint/venv/bin/octoprint
```

Si en la pestaña control visualizamos la imagen de la webcam seguimos.



Ahora lo que vamos a hacer es un script para poder iniciar o detener la transmisión de la cámara web. Para ello hay que crear una carpeta llamada scripts con un archivo llamado webcam que contendrá el script. Utilizamos los siguientes comandos:

```
cd
mkdir scripts
nano /home/pi/scripts/webcam
```

Y ahora dentro del editor de archivos copiamos el siguiente script:

```
#!/bin/bash
# Start / stop streamer daemon

case "$1" in
        start)
        /home/pi/scripts/webcamDaemon >/dev/null 2>&1 &
        echo "$0: started"
        ;;
        stop)
        pkill -x webcamDaemon
        pkill -x mjpg_streamer
        echo "$0: stopped"
        ;;
        *)
        echo "Usage: $0 {start|stop}" >&2
        ;;
esac
```

Guardamos con CTRL+X y Y para aceptar.

El siguiente fichero que haremos se hace exactamente igual:

```
nano /home/pi/scripts/webcamDaemon
```

```
#!/bin/bash

MJPGSTREAMER_HOME=/home/pi/mjpg-streamer/mjpg-streamer-experimental
MJPGSTREAMER_INPUT_USB="input_uvc.so"
MJPGSTREAMER_INPUT_RASPICAM="input_raspicam.so"

# init configuration
camera="auto"
camera_usb_options="-r 640x480 -f 10"
camera_raspi_options="-fps 10"

if [ -e "/boot/octopi.txt" ]; then
        source "/boot/octopi.txt"
fi

# runs MJPG Streamer, using the provided input plugin + configuration
function runMjpgStreamer {
        input=$1
        pushd $MJPGSTREAMER_HOME
        echo Running ./mjpg_streamer -o "output_http.so -w ./www" -i "$input"
        LD_LIBRARY_PATH=. ./mjpg_streamer -o "output_http.so -w ./www" -i "$input"
        popd
}

# starts up the RasPiCam
function startRaspi {
        logger "Starting Raspberry Pi camera"
        runMjpgStreamer "$MJPGSTREAMER_INPUT_RASPICAM $camera_raspi_options"
}

# starts up the USB webcam
function startUsb {
        logger "Starting USB webcam"
        runMjpgStreamer "$MJPGSTREAMER_INPUT_USB $camera_usb_options"
}

# we need this to prevent the later calls to vcgencmd from blocking
# I have no idea why, but that's how it is...
vcgencmd version

# echo configuration
echo camera: $camera
echo usb options: $camera_usb_options
echo raspi options: $camera_raspi_options

# keep mjpg streamer running if some camera is attached
while true; do
        if [ -e "/dev/video0" ] && { [ "$camera" = "auto" ] || [ "$camera" = "usb" ] ; }; then
        startUsb
        elif [ "`vcgencmd get_camera`" = "supported=1 detected=1" ] && { [ "$camera" = "auto" ] || [
"$camera" = "raspi" ] ; }; then
        startRaspi
        fi

        sleep 120
done
```

Guardamos con CTRL+X y Y para aceptar.

Ya creados los dos scripts, escribimos los siguientes comandos para hacer que sean ejecutables.

```
chmod +x /home/pi/scripts/webcam
chmod +x /home/pi/scripts/webcamDaemon
```

Y editamos el fichero config.yaml:

```
nano /home/pi/.octoprint/config.yaml
```

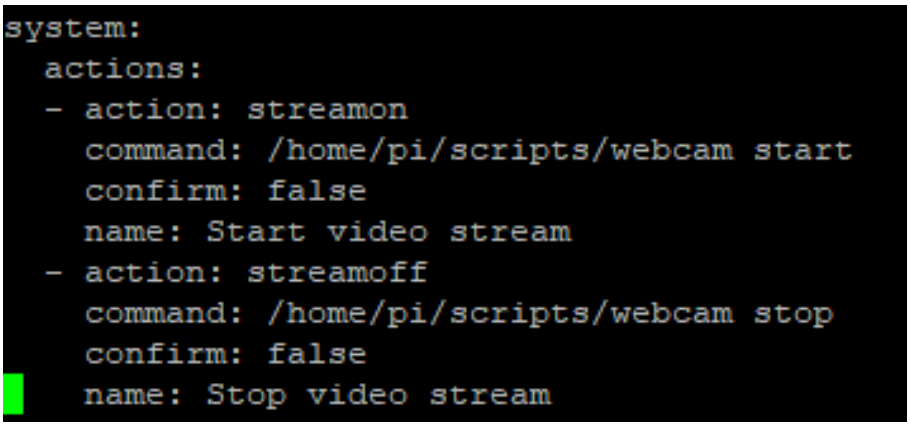
Y añadimos como antes lo siguiente:

```
system:
  actions:
   - action: streamon
     command: /home/pulpo/scripts/webcam start
     confirm: false
     name: Start video stream
   - action: streamoff
     command: /home/pulpo/scripts/webcam stop
     confirm: false
     name: Stop video stream
```

El archivo que estamos editando el 'config.yaml' es un poco especial y suele dar errores, hay que estar atento y cerciorarse de no introducir tabulaciones extra ni espacios innecesarios.

Si os da error el archivo yaml, la solución es mantener la estructura pero en vez con tabuladores, poned espacios.

Hay que mantener la estructura, si esta como en la foto con espacios no habrá problemas.



Para dejar este apartado listo reiniciamos:

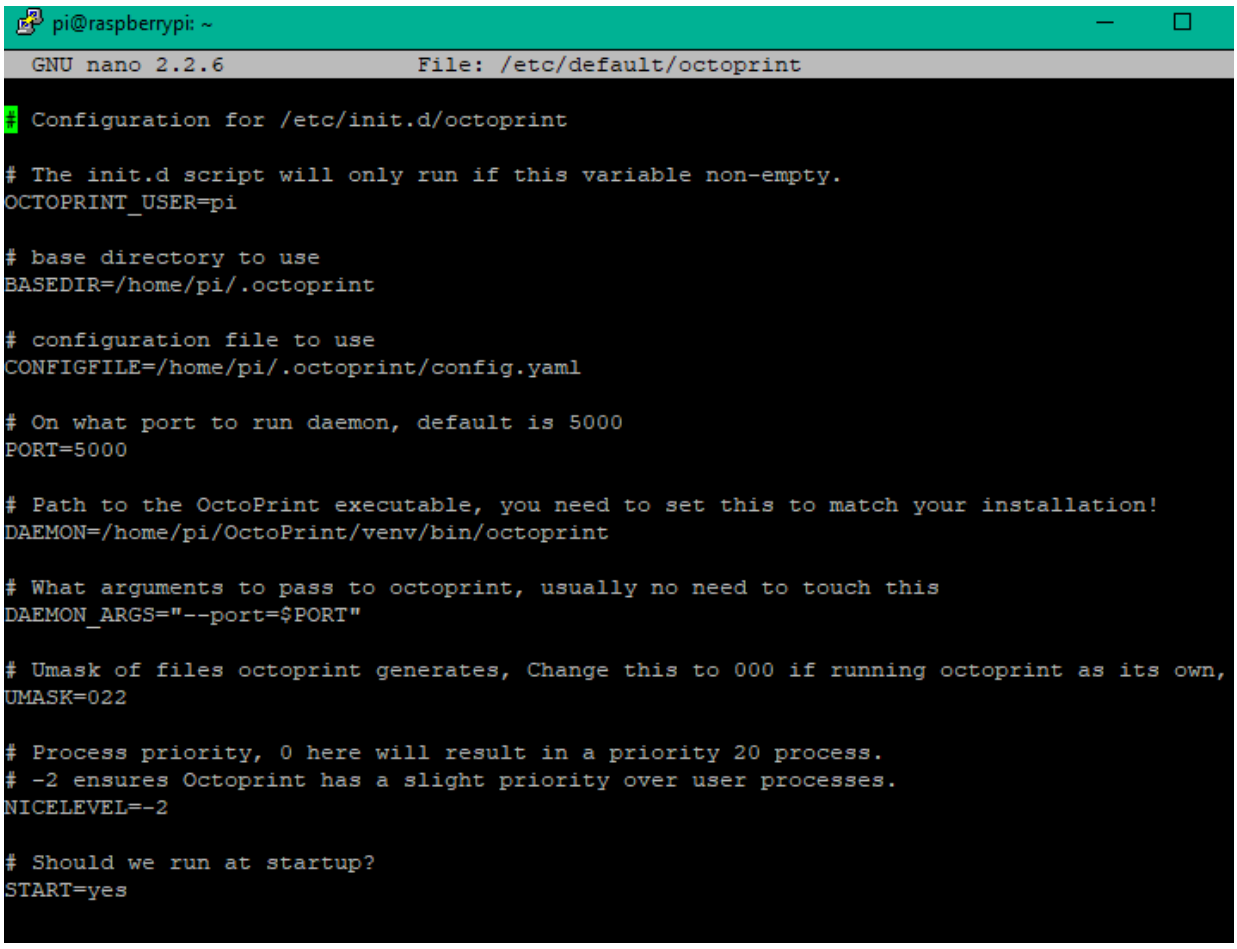
```
sudo reboot
```

Arranque automático de octoprint al iniciar rpi

Esto es necesario para automatizar el inicio de octoprint, ya que si no en cada reinicio deberíamos lanzar octoprint manualmente. Para hacer esto usamos los siguientes comandos:

```
sudo cp ~/OctoPrint/scripts/octoprint.init /etc/init.d/octoprint
sudo chmod +x /etc/init.d/octoprint
sudo cp ~/OctoPrint/scripts/octoprint.default /etc/default/octoprint
```

Abriremos el siguiente archivo una vez terminado los comandos anteriores y lo único que tendremos que hacer es dejarlo tal cual el de la foto siguiente quitando las almohadillas '#' que sean necesarias.



Guardamos como siempre con CTRL+X y ponemos el siguiente comando:

```
sudo update-rc.d octoprint defaults
```

```
sudo reboot
```

Una vez se reinicie deberíamos verificar yendo a octoprint con el navegador como hicimos la primera vez y viendo que la página se carga correctamente.

http://192.168.1.130:5000

Tarda un ratito en iniciar una vez le damos a reiniciar.

Instalación de haproxy

Este apartado no es mas que facilitar cierto tipo de tareas para el tema de los puertos en referencia al sistema y la cámara web, existen diversas ventajas de utilizar esto en vez de configurar octoprint en el puerto 80.

Lo primero que haremos será instalar haproxy:

```
sudo apt-get install haproxy
```

Una vez instalado lo único que hay que hacer es ir a un archivo y editarlo con la siguiente información.

```
sudo nano /etc/haproxy/haproxy.cfg
```

Borramos todo lo que haya dentro del archivo y copiamos lo siguiente:

```
global
        maxconn 4096
        user haproxy
        group haproxy
        daemon
        log 127.0.0.1 local0 debug

defaults
        log     global
        mode    http
        option  httplog
        option  dontlognull
        retries 3
        option redispatch
        option http-server-close
        option forwardfor
        maxconn 2000
        timeout connect 5s
        timeout client  15min
        timeout server  15min

frontend public
        bind *:80
        use_backend webcam if { path_beg /webcam/ }
        default_backend octoprint

backend octoprint
        reqrep ^([^\ :]*)\ /(.*)     \1\ /\2
        option forwardfor
        server octoprint1 127.0.0.1:5000

backend webcam
        reqrep ^([^\ :]*)\ /webcam/(.*)     \1\ /\2
        server webcam1  127.0.0.1:8080
```

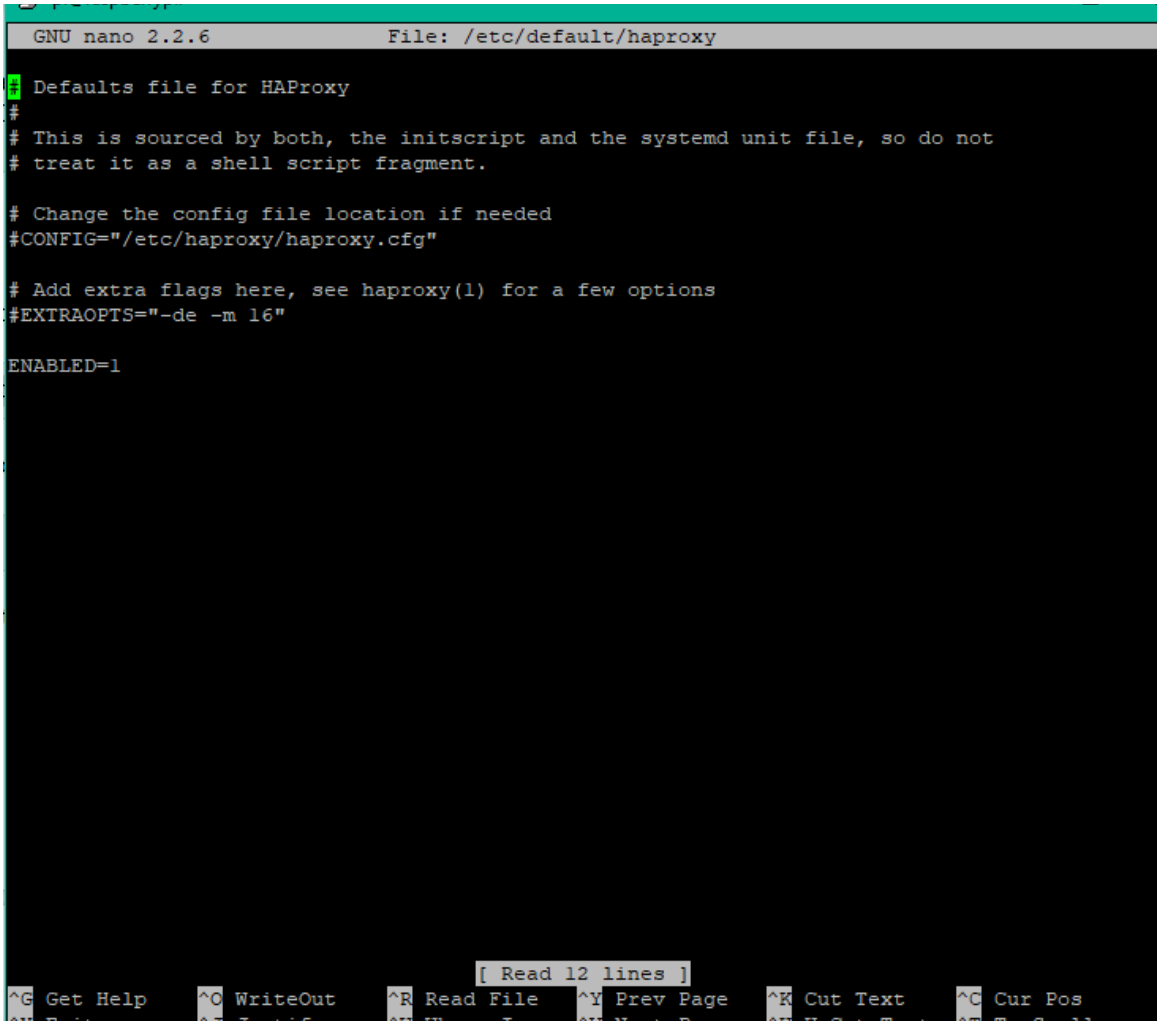
Guardamos y cerramos con CTRL+X y editamos otro fichero para activar haproxy:

```
sudo nano /etc/default/haproxy
```

Al final de sus líneas escribiremos, añadiendo lo siguiente:

```
ENABLED=1
```

Quedando como en la siguiente imagen:



Guardamos y cerramos con CTRL+X

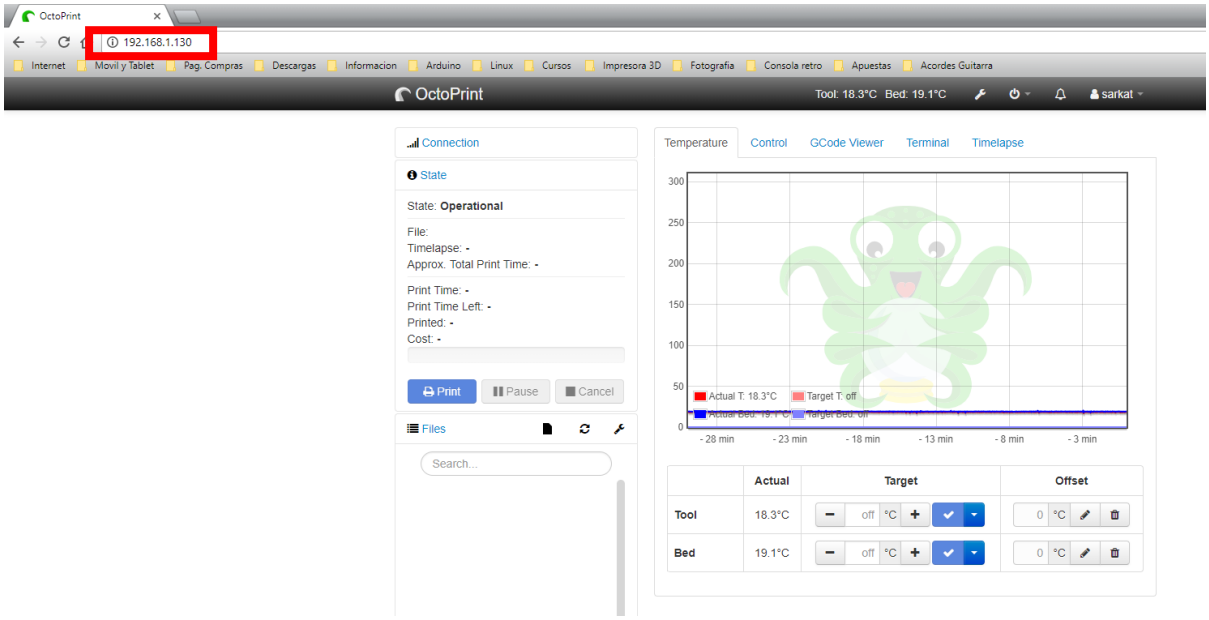
Arrancamos haproxy con la siguiente línea de comando, y una vez hecho reiniciamos el la rpi.

```
sudo service haproxy start
sudo reboot
```

Con esto deberíamos poder ir al navegador ahora y entrar a nuestro octoprint tecleando en la barra de direcciones tan solo nuestra ip.

http://192.168.1.130

Así es como yo ahora mismo puedo acceder a mi servidor.



Si va todo bien y estamos en el mismo punto vamos a hacer unas ultimas modificaciones en el archivo 'config.yaml'

```
nano ~/.octoprint/config.yaml
```

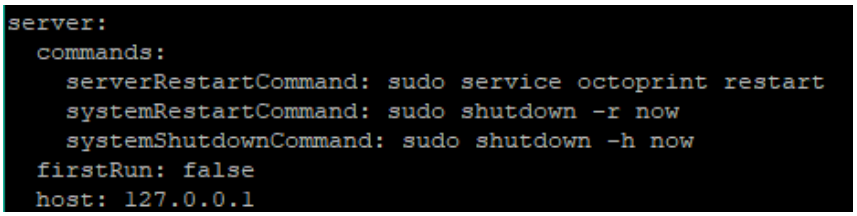
Y este archivo es delicado, vamos por partes, el mismo esta dividido en apartados, y tenemos que buscar el apartador webcam y tiene que estar como en la imagen, sol oes sustituir la línea perteneciente a stream, por lo que hay en la imagen:

```
webcam:
  ffmpeg: /usr/bin/avconv
  flipH: true
  flipV: true
  snapshot: http://127.0.0.1:8080/?action=snapshot
  stream: /webcam/?action=stream
```

Establezca la URL de webcam por " `/webcam/?action=stream`"

Y ahora igual que hay un apartado llamado webcam, se sabe porque son los que no tienen espacios en el editor como se aprecia en la foto de arriba.

Debemos buscar el apartado server, y si no esta crearlo, con la misma distribución que los demás apartados. El apartado va sin espacios y lo que se quiera añadir va en la línea de abajo, pero con espacios, sin tabulador quedando como en la foto siguiente:



Y hay que añadir en el apartado server, la línea siguiente:

```
host: 127.0.0.1
```

Guardamos y cerramos con CTRL+X

Una vez listo solo queda reiniciar con 'sudo reboot' y ya debería estar todo disponible con el puerto 80.

Todo esto se hace para que como en mi caso poder acceder fuera de mi red domestica a mi servidor y el puerto será 80.

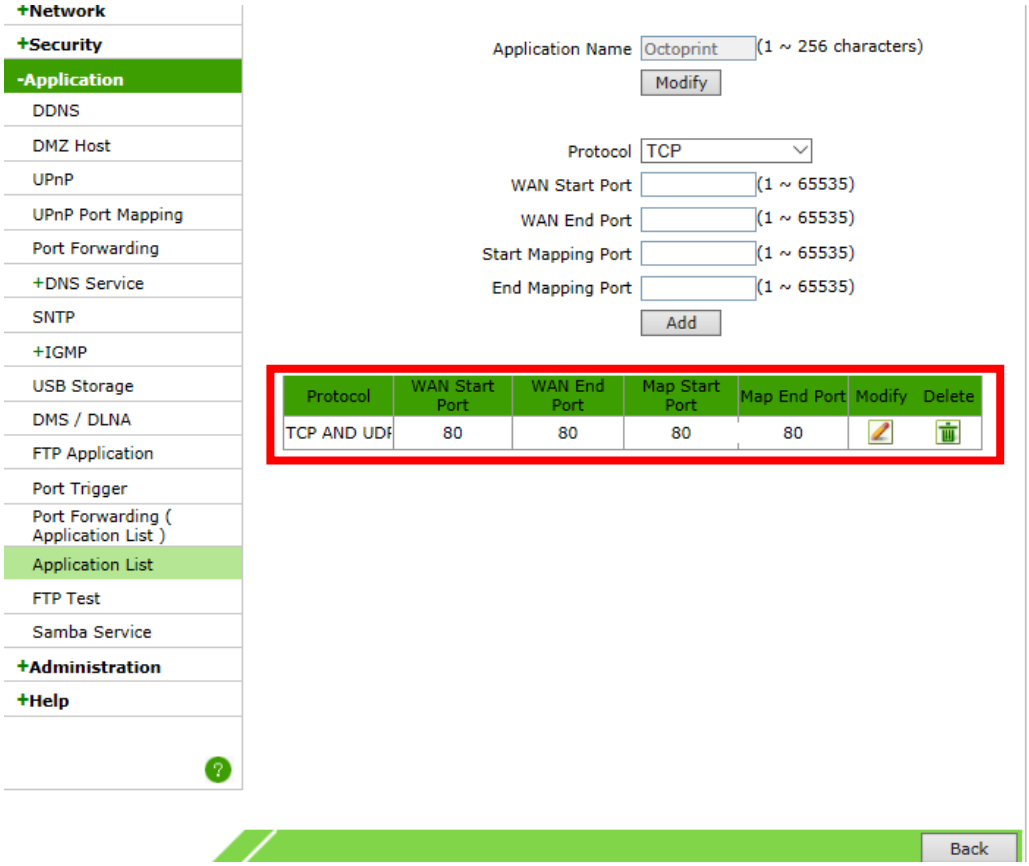
Configuración para poder acceder desde red externa

Simplemente este apartado es para acceder desde cualquier parte mediante internet a nuestro servidor y poder controlar la impresora remotamente desde cualquier punto con internet.

Se necesita conocer un poco el router propio, aquí se darán nociones genéricas, pero cada router es diferente y entonces este tutorial sería muy extenso y laborioso. Aun asi según las necesidades de cada uno, es tan fácil como buscar información de Google de como abrir los puertos de vuestro router personal. Suelen ser de una manera parecida en todos.

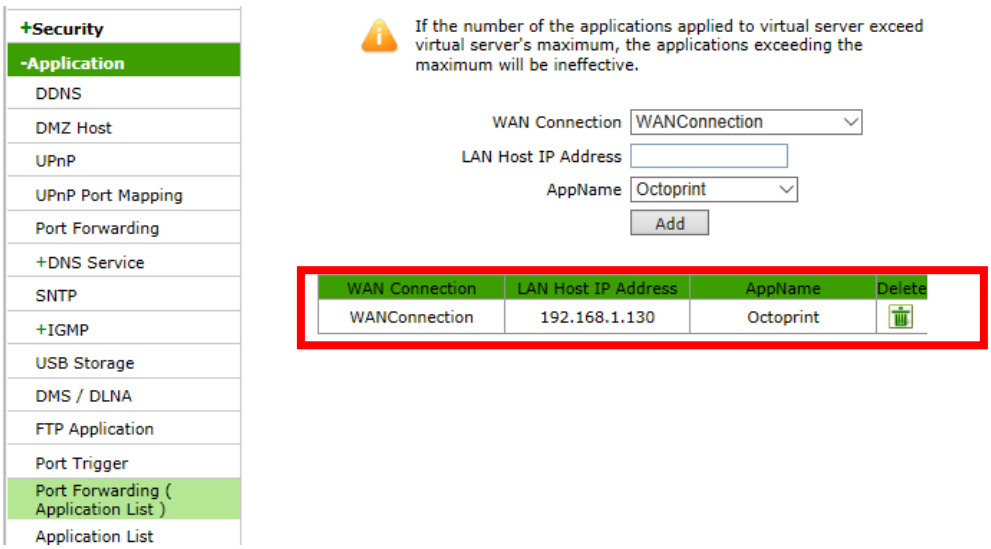
- Lo primero que hay que hacer es abrir el puerto que asociaremos con nuestra ip.

En este apartado como he comentado para cada router será diferente pero parecido, se ha abierto el puerto 80 porque es el que reconocerá la rpi ya que tiene el 80 por el haproxy antes instalado.



Es importante poner un nombre coherente ya que luego se utiliza ese nombre, tan solo es poner 80 en todos los casilleros vacíos y seleccionar TCP and UDP.

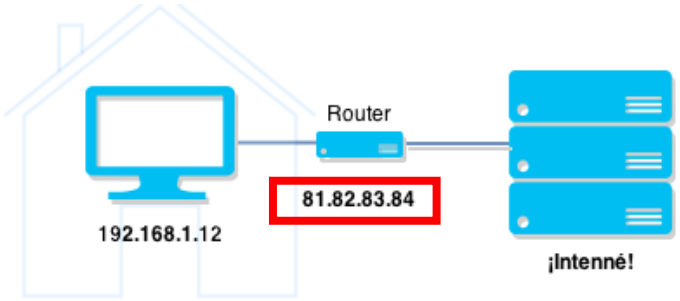
- El siguiente paso es asociar ese puerto a la dirección ip de la rpi, para que cuando llegue una petición externa a nuestra red, nuestro router solo escuchara si la petición viene por el puerto 80 y si es así directamente enlazara con la ip que se tenga asociada la rpi.



Lo único es añadir la ip que tiene tu rpi y en 'Appname' seleccionar el nombre del puerto abierto en el paso anterior.

- Hacer host en no-ip

El problema en estos casos es que el router cambia si ip publica cada tiempo por seguridad, y para contactar con nuestro router desde fuera de nuestra red se necesita saber la ip publica, aunque se supiese no serviría de mucho ya que cuando cambie de manera aleatoria ya no se tendría acceso al router por no conocer la dirección.



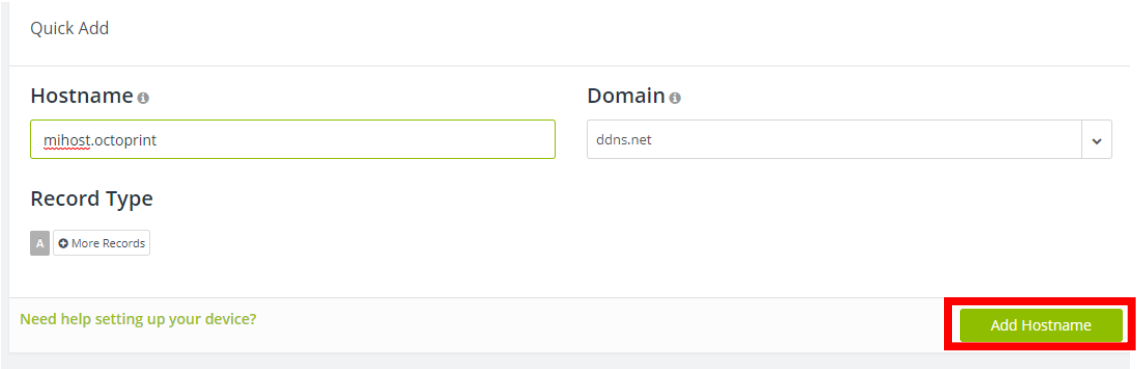
Esta imagen representa lo explicado anteriormente, como se ve el router es el intermediario entre tu red local e internet. El valor marcado es la dirección actual del router que es el que varía con el tiempo.

Lo que vamos a conseguir es asociar un nombre (host) que siempre será fijo y estará asociado a la dirección del router que será variable. Con esto se consigue que mientras pongas el host da igual que varíe la dirección ya que cada cambio se asociará al host.

1. Crear cuenta en

   https://www.noip.com

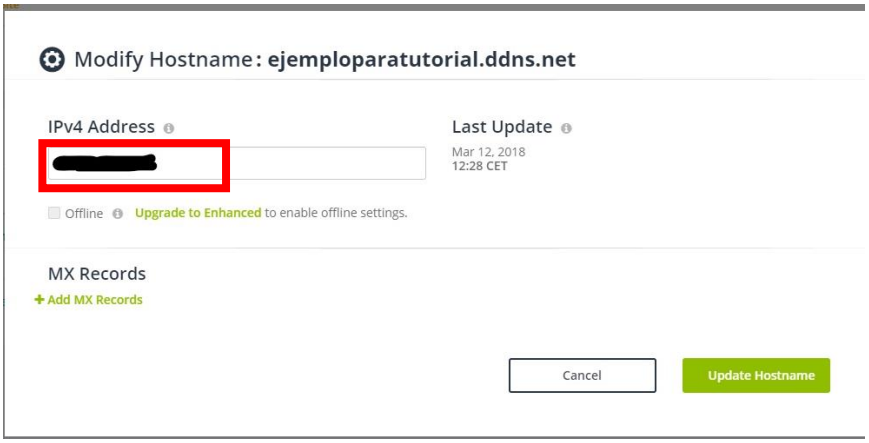
2. Una vez con acceso, se crea un host, con el nombre que quieras, si esta en uso te avisará.



Por ejemplo, este podría ser válido y se crea el hostname. Una vez creado te vas al apartado a tu host creados, y en el apartado modificar tenemos que poner nuestra ip pública del router, la que sea actualmente.

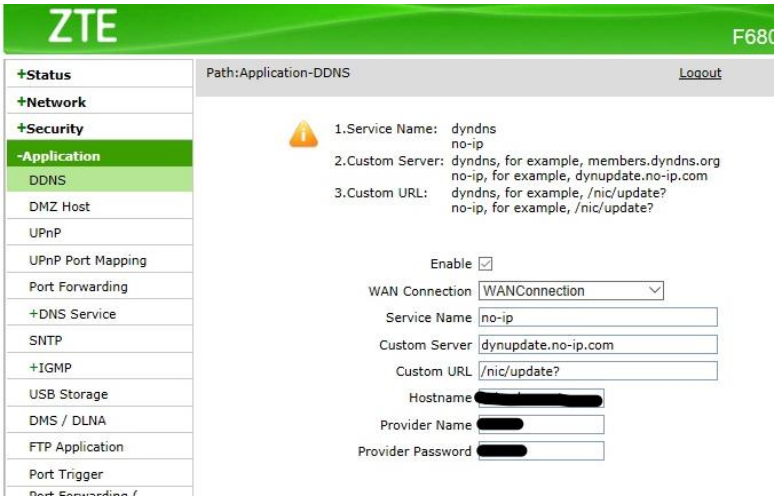
3. Obtener la ip publica de nuestro router, es tan simple como mirarlo en la configuración de tu router en información en apartado ip. Se accede a la configuración del router mediante la dirección (192.168.1.1). También se puede saber tan fácil como escribir en Google conocer mi ip' y habrá muchas paginas que te dicen la ip publica de tu router.

Ese numero es que se tendrá que asociar a ese host que acabamos de crear. En el recuadro se introduce nuestra ip y se guardará la configuración.



Por defecto gratuitamente podrás crear hasta 3 host y el único contra es que hay que actualizar el host manual cada mes, no es mas que meterte en tu host creados y darle a update hostname. Avisan con un correo 7 días antes de que expira, cada mes habrá que meterse y actualizar con un clic y listo.

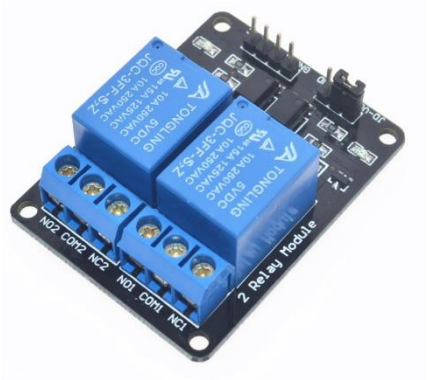
4. Lo único que queda es en configuración de router en el apartado ddns, poner el host creado con la cuenta creada.



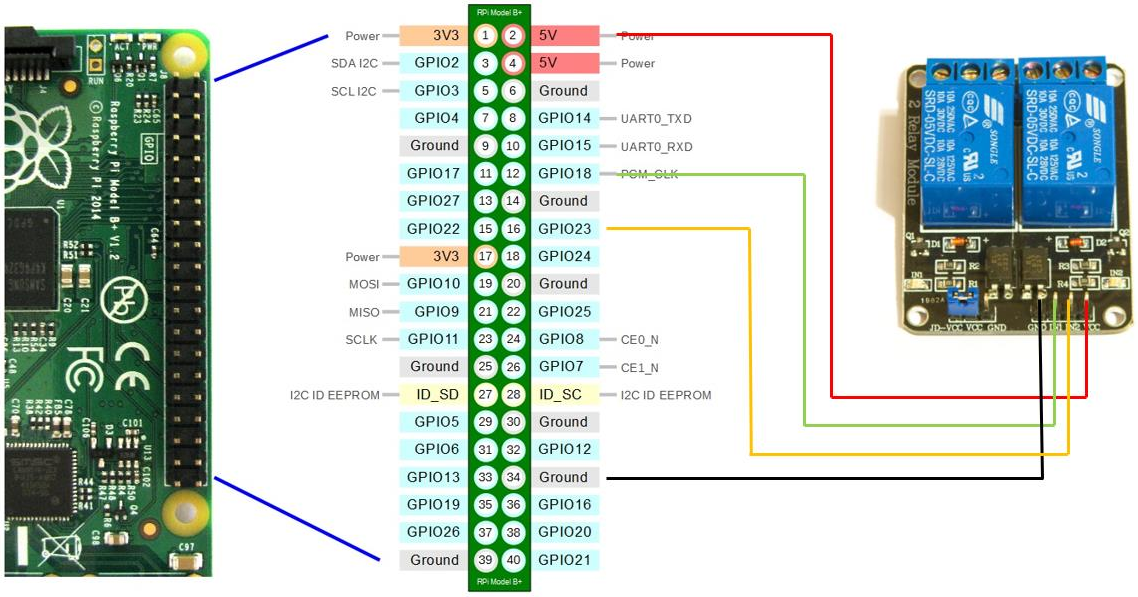
Con todo lo realizado anteriormente, podemos comprobar si podemos acceder desde fuera de nuestra red a octoprint. Solo hay que tener en cuenta de actualiza el host cada mes.

Poner relé para encender o apagar la impresora desde octoprint

Lo primero que habría que hacer es adquirir un módulo de relés, en el tutorial se usa un modulo de dos relés, uno para encender y apagar impresora y el otro para encender o apagar una tira de led.



Las conexiones se efectuarán de la siguiente forma:



Los pines escogidos son el GPIO 18 y el GPIO 23.

Se comenzará por instalar algunas librerías en la rpi para tener control de los pines. Nos conectaremos SSH a la rpi y teclearemos los siguientes comandos:

```
sudo apt-get install git-core
sudo apt-get update
sudo apt-get upgrade
git clone git://git.drogon.net/wiringPi
cd wiringPi
git pull origin
./build
```

Ahora configuraremos estos pines como salidas con los comandos:

```
gpio -g mode 23 out
gpio -g mode 18 out
```

Ahora ya deberíamos poder controlar el relé usando comandos por consola para ello, estando todo conectado como se ha explicado anteriormente, tiramos el siguiente comando:

```
gpio -g write 23 1
gpio -g write 18 1
```

Al ejecutar el comando el relé deberá cambiar de estado y con el siguiente comando se ponen a cero:

```
gpio -g write 23 0
gpio -g write 18 0
```

Comprobado que podemos manejar los relés, vamos a asociar estas acciones con octoprint.

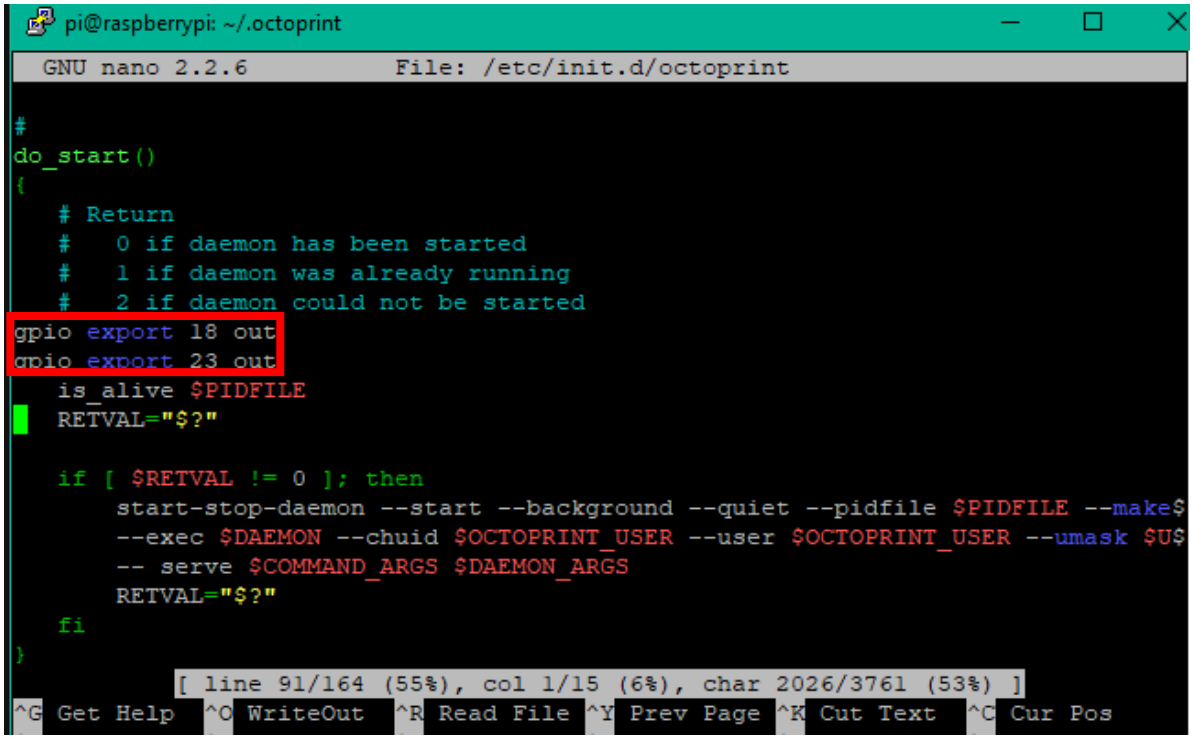
Ahora iremos al archivo siguiente mediante el siguiente comando:

```
sudo nano /etc/init.d/octoprint
```

Buscamos la línea que ponga (RETVAL=”$?”) y tenemos que añadir lo siguiente antes de que se ejecute esa línea.

```
gpio export 18 out
gpio export 23 out
```

Quedando de la siguiente forma:



Una vez listo, como siempre CTRL+X y aceptar

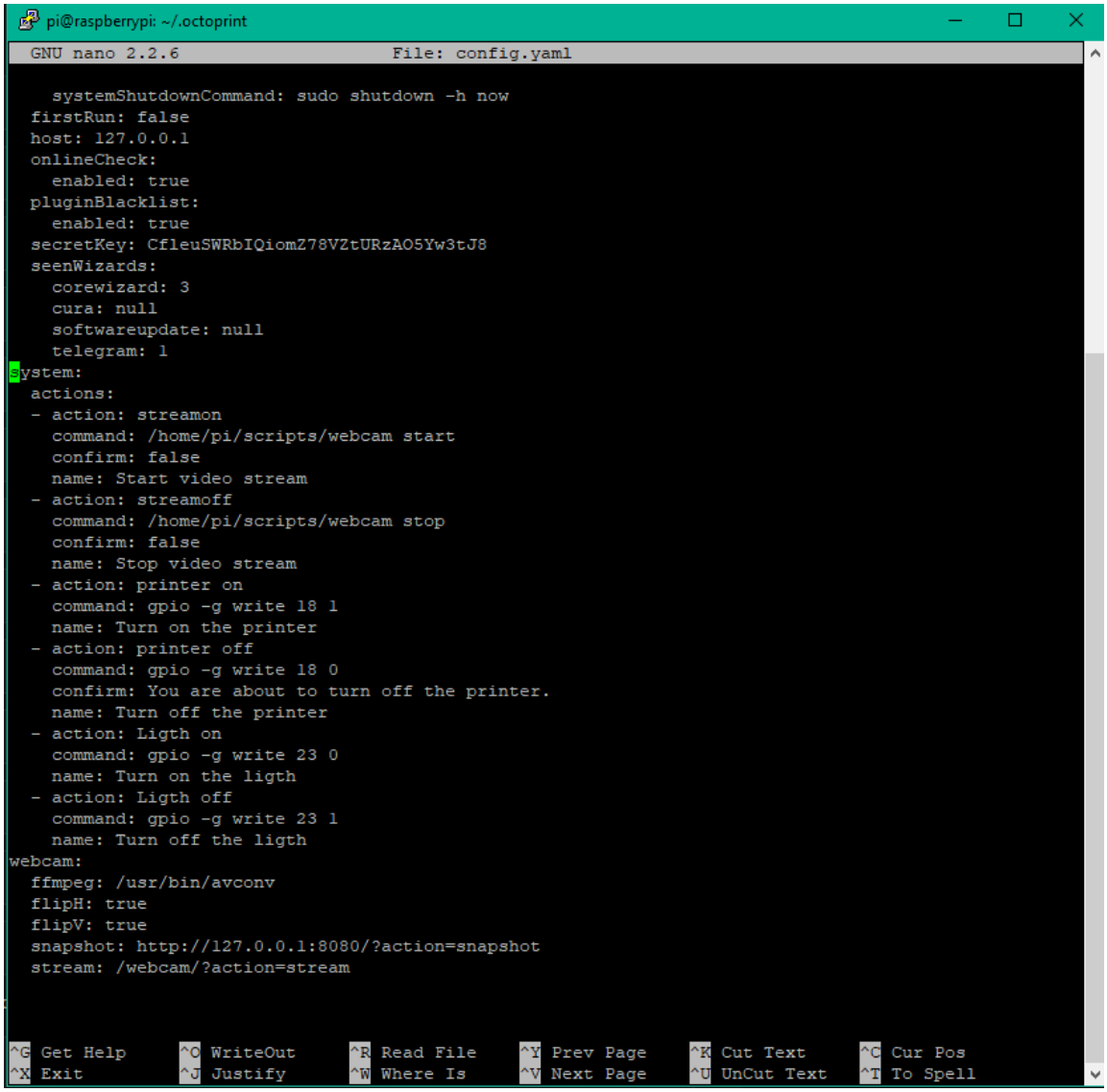
Llegados a este punto tenemos que editar un archivo ya conocido, el config.yaml y en el apartado ‘system’ se añadirá lo siguiente:

```
cd
cd /home/pi/.octoprint
sudo nano config.yaml
```

Ya estando en el archivo, ahora añadimos en system:

```
- action: printer on
    command: gpio -g write 18 1
    name: Encender impresora
- action: printer off
    command: gpio -g write 18 0
    name: Apagar impresora
- action: Light on
    command: gpio -g write 23 1
    name: Encender led
- action: Light off
    command: gpio -g write 23 0
    name: Apagar luz
```

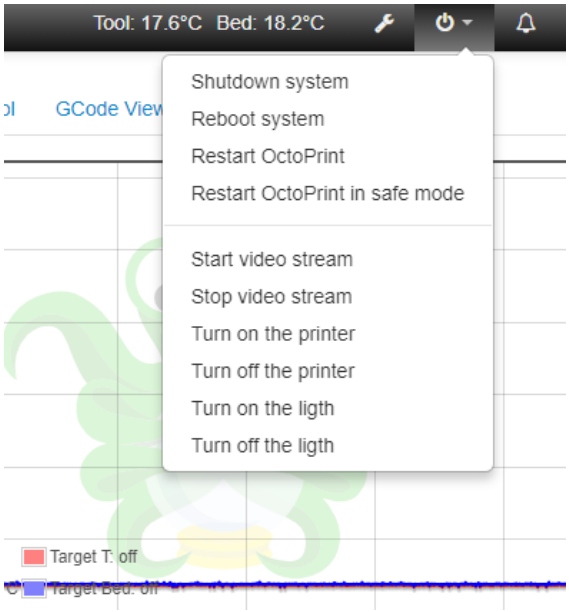
Quedando como en la siguiente imagen, recordad que este archivo no puede contener, espacios ni tabulaciones fuera de lugar, mantener la estructura. Y si hay fallo revisad bien el tema de espacios.



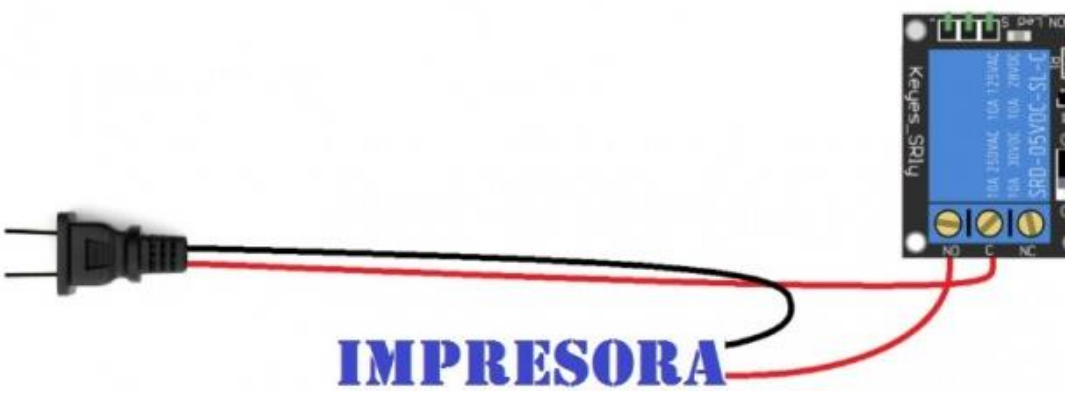
Como siempre CTRL+X y aceptar, una vez listo toca reiniciar la rpi:

```
sudo reboot
```

Si todo ha ido bien al reiniciar, en la interfaz octoprint, nos deberían aparecer todas las opciones que acabamos de añadir:



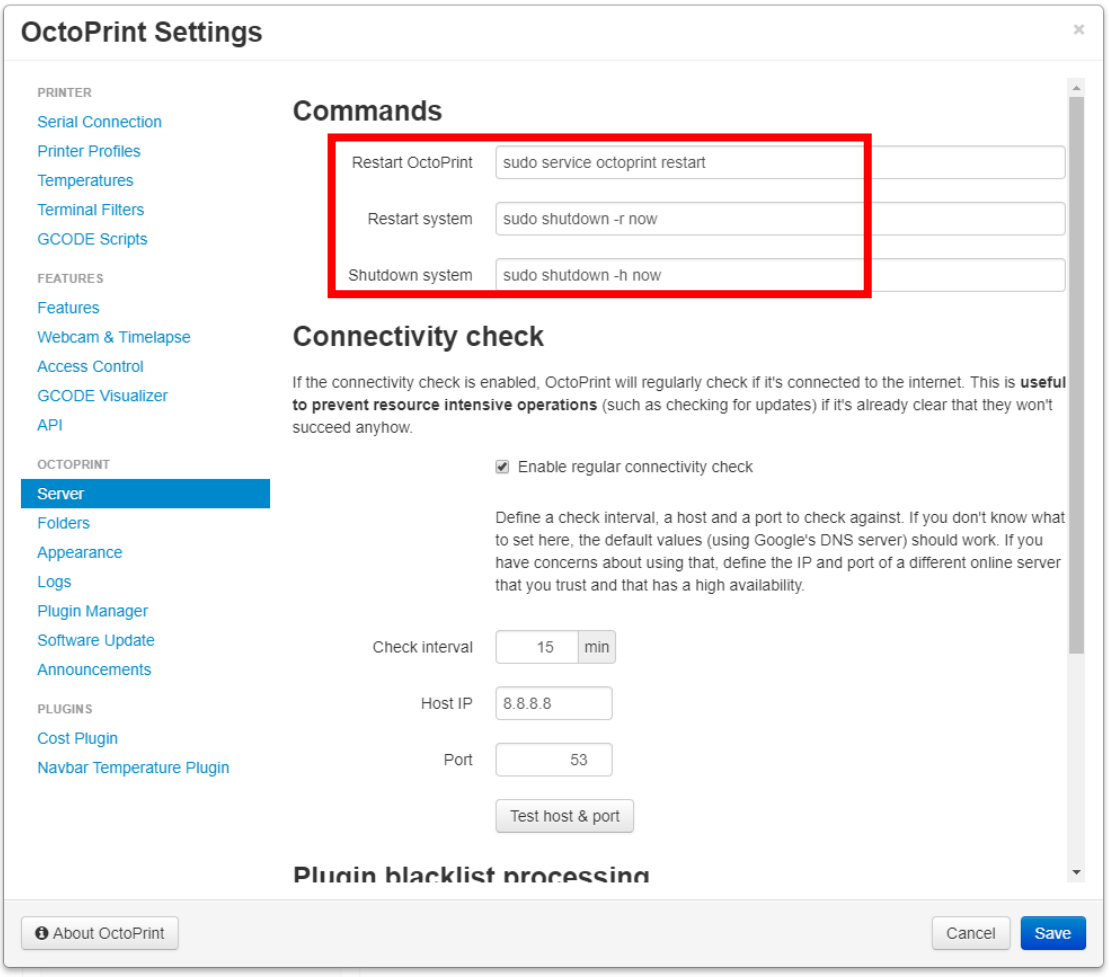
Ya podemos controlar los relés mediante la interfaz de octoprint. Lo que quedaría es la conexión con los relés no seria mas que cortar el cable de masa de el enchufe de la impresora y conectarlo a un relé para que abra o cierre el circuito según nuestra demanda. Con el led es exactamente igual.



Añadir apagado y reinicio de rpi a octoprint

Es simplemente para poder reiniciar o apagar la rpi desde la interfaz octoprint para no tener que abrir consola y poder hacerlo todo desde el mismo navegador.

- Nos vamos a ajustes de octoprint al apartado servers:



Añadimos los siguientes comandos tal como en la foto.

Actualizar octoprint cuando haya nueva versión

La actualización no se hará automática habrá que entrar por SSH y una vez estando en consola introducir los siguientes comandos:

```
cd ~/OctoPrint/
git pull
./venv/bin/python setup.py clean
./venv/bin/python setup.py install
```

Y reiniciamos con:

```
sudo reboot
```