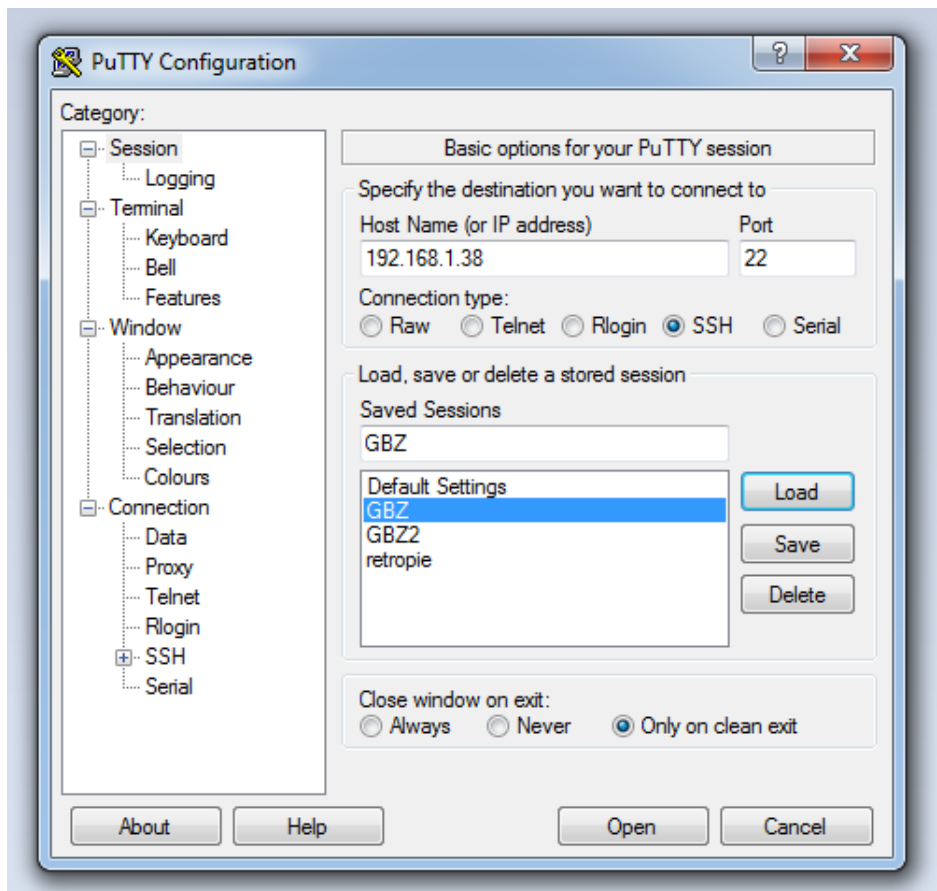


# *Programación Pi.*

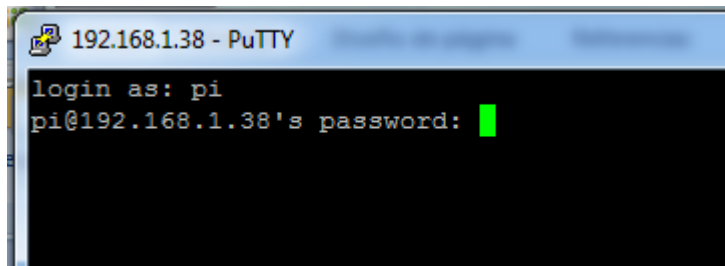
Para poder hacer funcionar correctamente tanto el audio por gpio como la botonera (mandos) en la Raspberry Pi hace falta instalar unos cuantos scripts que detallaremos en los siguientes apartados. Cabe mencionar que el sistema del que hablamos en todo momento durante este pequeño manual es RetroPie el cual está basado en raspbian. Mencionar también que si disponemos de una Raspberry Pi Zero W solo necesitaremos conectar mediante SSH de forma remota utilizando programas tipo PuTTY. En cambio, si utilizamos una Raspberry Pi Zero necesitaremos hacerlo con un dongle WiFi. En el caso que queramos ir al grano sin explicaciones pasad de todo lo siguiente e id al resumen de pasos a seguir que hay al final.

## *Conexión a través de ssh por PuTTY*

Para poder trabajar remotamente a través de ssh utilizaremos PuTTY, el cual tiene su versión Windows y Linux. Una vez descargado e instalado abriremos el programa e introduciremos la dirección IP de nuestra Raspberry en el apartado “Host Name” y le pondremos el puerto 22 en “Port”. Después pulsaremos en “Save” para que nos guarde la conexión y la próxima vez que vayamos a conectarnos no tengamos que teclear lo anterior de nuevo.

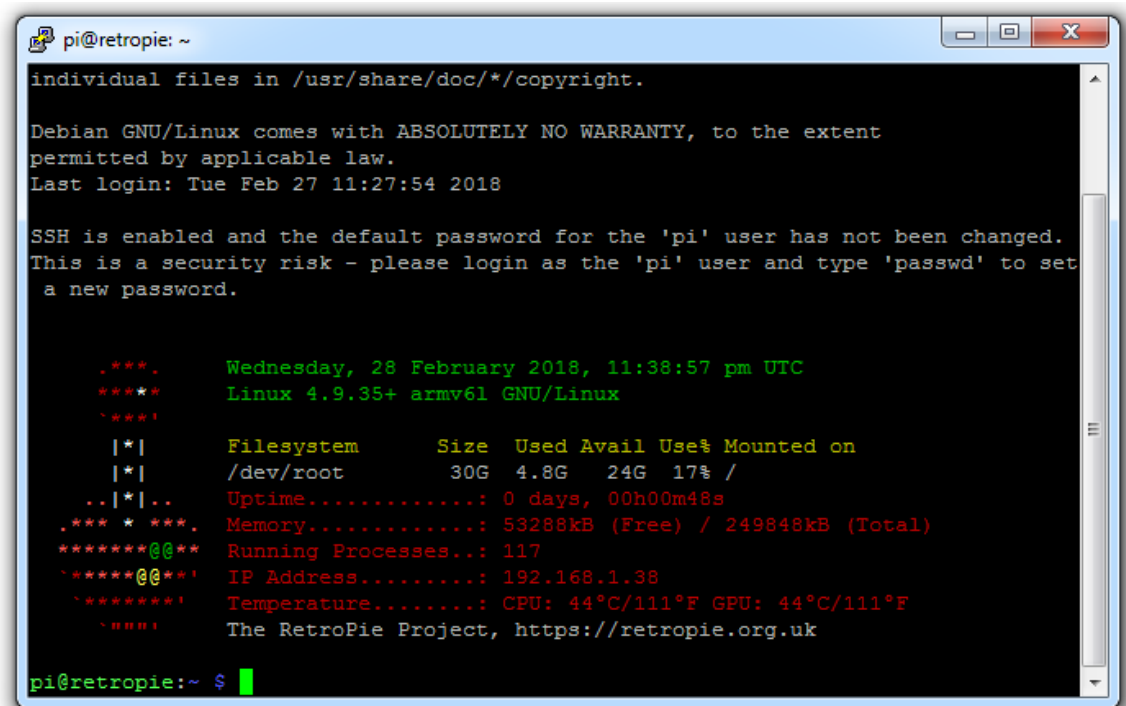


Una vez tengamos hecho lo anterior pulsaremos en “Open” y se nos intentara abrir la conexión. Si todo está configurado de manera correcta nos pedirá un usuario y una clave para poder conectarnos.



El usuario y clave por defecto son:  
login as: pi  
password: raspberry (no saldrá nada al teclearlo)

Hecho lo anterior deberíamos llegar a una pantalla similar a esta:



Esta pantalla la denominaremos de ahora en adelante consola terminal o consola de comandos. En este punto ya tendremos realizada la conexión necesaria para realizar los cambios en el audio e introducir los diferentes scripts. **IMPORTANTE: para pegar texto en esta consola lo seleccionaremos como siempre y lo pegaremos pulsando los botones derecho e izquierdo del ratón a la vez encima de la consola si estamos desde Linux (Ubuntu) o si estamos en Windows exactamente de la misma manera pero solamente con el botón derecho del ratón.**

## ***Descarga de archivos necesarios para el funcionamiento***

Para el correcto funcionamiento deberemos descargar una serie de archivos para después colocarlos en el sitio que le corresponde y así funcione todo de manera correcta y rápida. Dicha descarga la realizaremos conectándonos a través de PuTTY tecleando en la consola de comandos lo siguiente:

```
git clone https://github.com/julenvitoria/miguelito
```

Una vez pulsemos Intro veremos que la descarga comienza.

## ***Modificar el teclado para poder escribir de forma correcta en consola***

Esto es algo opcional pero muy recomendable ya que el sistema tendrá como teclado predeterminado el teclado inglés con una distribución de teclas (sobre todo para símbolos) diferente al español. Para cambiar la distribución del teclado deberemos teclearnos:

```
sudo nano /etc/default/keyboard
```

Una vez dentro nos desplazaremos hasta la línea `XKBLAYOUT="gb"` y la modificaremos quedando de la siguiente manera:

```
XKBLAYOUT="es"
```

Para guardar presionaremos **Ctrl+O** confirmando con **Intro** y por último saldremos con **Ctrl+X**.

Una vez hayamos hecho lo anterior tendremos el teclado en español y podremos escribir cualquier símbolo de la manera que estamos acostumbrados.

## ***Audio por pwm (a través de GPIO)***

En la consola de comandos editaremos el archivo `/boot/config.txt` escribiendo lo siguiente:

```
sudo nano /boot/config.txt
```

Una vez hayamos entrado a editar el archivo añadiremos lo siguiente al final del archivo:

```
#Audio por pwm
dtoverlay=pwm-2chan,pin=18,func=2,pin2=13,func2=4
```

Para guardar presionaremos **Ctrl+O** confirmando con **Intro** y por último saldremos con **Ctrl+X**.

**Opcional (en muchos casos debería funcionar sin hacer esto):**

Para forzar la salida de audio como jack de 3,5mm podemos hacerlo de diversas maneras:

- 1-Desde el menú raspi-config en la sección RetroPie del menú principal.
- 2-Desde el menú audio dentro de la sección RetroPie del menú principal.
- 3-Manualmente escribiendo en el terminal:

```
sudo nano /etc/modules
```

y añadiendo la línea:

```
sudo amixer cset numid=3 1
```

## **Instalar y configurar script retrogame para control de los botones**

Retrogame básicamente es el programa que utilizaremos para que nuestra raspberry reconozca las pulsaciones de los botones a través de los puertos GPIO. Para poder utilizarlo deberemos descargar, instalar y configurar el programa con los siguientes comandos desde terminal a través de PuTTY:

```
curl -O https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Installer-Scripts/master/retrogame.sh
```

y después:

```
sudo bash retrogame.sh
```

Una vez hecho lo anterior nos aparecerá la siguiente pantalla:

```
This script downloads and installs
retrogame, a GPIO-to-keypress utility
for adding buttons and joysticks, plus
one of several configuration files.
Run time <1 minute. Reboot recommended.

Select configuration:
1. PiGRRL 2 controls
2. Two buttons + joystick
3. Six buttons + joystick
4. Adafruit Arcade Bonnet
5. Quit without installing

SELECT 1-5: 2
```

Elegiremos la opción “Six buttons + joystick” . Después de descargarse e instalarse el ejecutable retrogame y el archivo de configuración (/boot/retrogame.cfg), realiza modificaciones en /etc/rc.local para hacer que arranque al iniciar la Raspberry. Una vez finalizadas las operaciones nos pedirá reiniciar para que se apliquen los cambios.

Después de reiniciar el sistema volveremos a conectarnos a través de SSH (PuTTY) y copiaremos el archivo retrogame.cfg que tenemos en la carpeta miguelito a /boot:

**sudo cp -R /home/pi/miguelito/miguelito/retrogame.cfg /boot**

Esta configuración es un tanto peculiar en cuanto a las teclas elegidas se refiere porque así podremos movernos por los menús de configuración (como por ejemplo raspi-config) sin necesidad de conectar un teclado. De esta manera podremos, por ejemplo, instalar un emulador de manera autónoma sin teclado. También cabe destacar que esta configuración tiene, además de la cruceta y los botones start y select, diez botones configurados, los cuales se pueden omitir comentando la línea escribiendo el símbolo # al principio de la línea de la configuración que no deseemos o directamente borrando esa línea. Otra opción es dejarlo tal cual, no ocurrirá nada si no hay botones.

Si por otro lado queremos darle otra configuración de teclas podremos consultar el nombre de las teclas en el archivo /usr/include/linux/input.x, cambiando el nombre de la tecla y siempre respetando la estructura del texto (espacios, etc...). No cambiar los GPIO ya que ya están asignados en la placa físicamente, solo el nombre de la tecla. Es **MUY** importante mantener los espacios entre la letra y el puerto GPIO, es decir, mantener la estructura original

Una vez pegado el texto para guardar el fichero pulsáis **CTRL+O** y confirmar con la tecla **Intro**, seguidamente **CTRL+X** para salir.. No hará falta reiniciar para que la configuración surta efecto, por lo que podremos realizar la comprobación de si funciona directamente en el menú de RetroPie.

## **Instalar temas para pantalla 3,5”**

Tenemos tres opciones para elegir:

- tema gbz35: unos 3 megas
- tema snes-mini: unos 18 megas
- tema carbon modificado: unos 7 megas
- tema pixel modificado: unos 2 megas
- tema pixel modificado: unos 1,5 megas

Estos temas se pueden instalar y convivir todos juntos o instalar solo uno para ahorrar espacio. Para instalar los temas abrir la consola de terminal y teclear lo siguiente:

Para tema gbz35

**sudo mv /home/pi/miguelito/miguelito/gbz35 /etc/emulationstation/themes/**

Para tema snes-mini

**sudo mv /home/pi/miguelito/miguelito/snes-mini /etc/emulationstation/themes/**

Para tema pixel

**sudo mv /home/pi/miguelito/miguelito/pixel /etc/emulationstation/themes/**

Para tema pixel-tft

**sudo mv /home/pi/miguelito/miguelito/pixel /etc/emulationstation/themes/**

Para tema carbon

Al ser el tema que viene por defecto tendremos inicialmente que borrarlo y después pasar el tema modificado. Para borrarlo primeramente teclear:

**sudo rm -R /etc/emulationstation/themes/carbon**

Para pasarlo teclear:

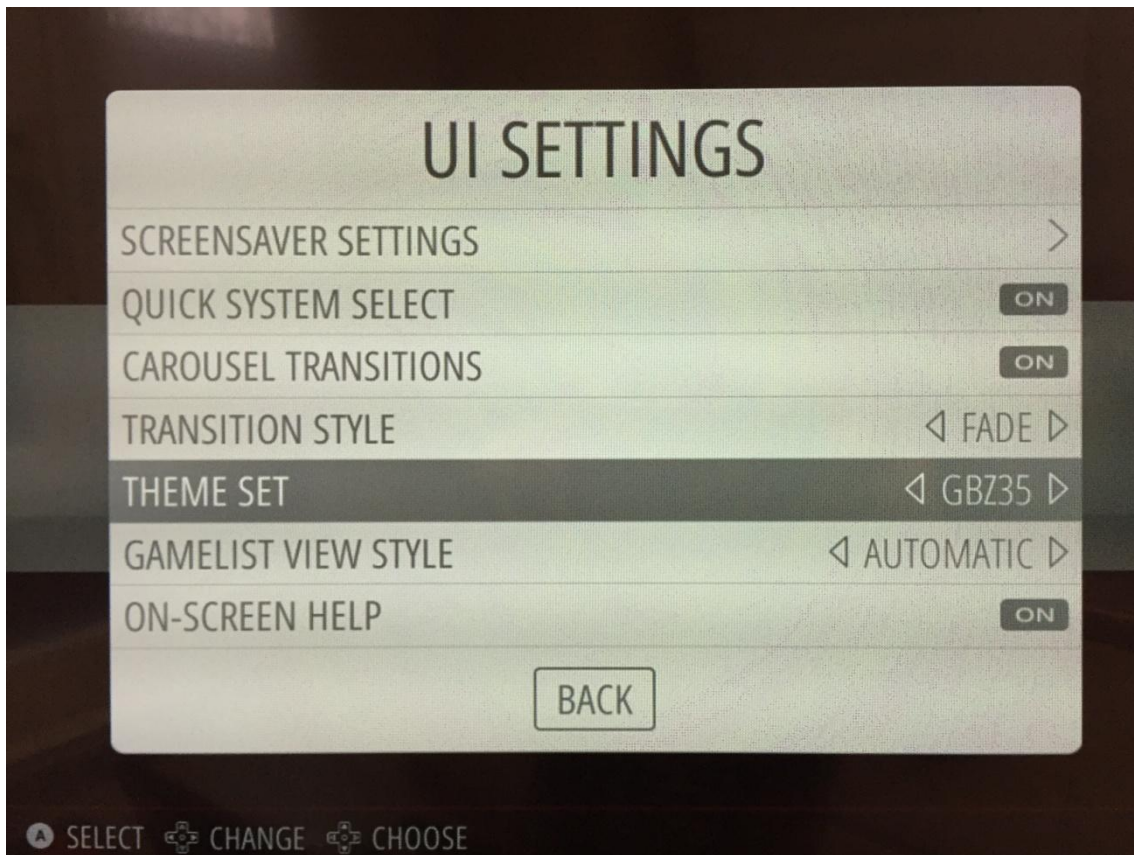
**sudo mv /home/pi/miguelito/miguelito/carbon /etc/emulationstation/themes/**

Con esto lo que haremos será mover los temas que tenemos en la carpeta miguelito a la carpeta de temas de emulationstation. Si no deseamos tener alguno de los temas se puede borrar a través de WinSCP, Filezilla o a través de terminal con PuTTY con el comando:

**sudo rm -R /home/pi/miguelito/*nombredeltema***

siendo *nombredeltema* el tema en cuestión que queremos borrar.

Para seleccionar el tema y poder disfrutar de él en nuestra pantalla de 3,5” deberemos de pulsar la tecla start para que se nos despliegue un menú. Nos desplazaremos hasta la opción “UI-Settings” y entraremos en ella pulsando A.



Nuevamente nos desplazaremos hasta la opción “Theme Set” y cambiaremos el tema pulsando la tecla derecha o izquierda hasta encontrar “gbz35”, “snes-mini” o “showcase”. Al volver al menú principal ya tendremos el tema funcionando con los textos adaptados a nuestra pantalla.

Si deseamos liberar algo de espacio en la microsd podemos borrar el tema que no vayamos a usar de la carpeta miguelito con los siguientes comandos:

Para tema gbz35

**sudo rm -R /home/pi/miguelito/miguelito/gbz35**

Para tema snes-mini

**sudo rm -R /home/pi/miguelito/miguelito/snes-mini**

Para tema showcase

**sudo rm -R /home/pi/miguelito/miguelito/showcase**

## *Instalacion splashscreen gameboy*

Abrir la consola de terminal y teclear lo siguiente para mover el video de la intro a la carpeta de los splashscreens:

**sudo mv /home/pi/miguelito/miguelito/GBZ-Miguelito.mp4  
/home/pi/RetroPie/splashscreens**



Copiar el archivo siguiente con el comando:

```
sudo mv /home/pi/miguelito/miguelito/splashscreen.list /etc
```

En el siguiente arranque ya tendremos nuestro splashscreen reproduciéndose.

## *Instalación y configuración script de música de fondo en el menú de RetroPie*

Lo primero de todo comentar que este script funciona a través del modulo pygame, el cual deberemos instalar ya que no viene por defecto con RetroPie. Lo instalaremos como de costumbre a través de SSH (PuTTY) con los siguientes comandos:

```
sudo apt-get update
```

Y después:

```
sudo apt-get -y install python-pygame
```

Una vez instalado el módulo deberemos crear una carpeta llamada music dentro de la carpeta de roms y otra llamada script dentro de music. Lo haremos con los siguientes comandos:

```
mkdir /home/pi/RetroPie/roms/music
```

y después

```
mkdir /home/pi/RetroPie/roms/music/script
```

Una vez hayamos creado las carpetas copiaremos el script de música que tenemos en la carpeta miguelito a la carpeta script con el comando:

```
cp /home/pi/miguelito/miguelito/music.py /home/pi/RetroPie/roms/music/script
```

Hasta aquí ya tenemos pasado el script de música (music.py) a su carpeta correspondiente (/home/pi/RetroPie/roms/music/script). Ahora debemos hacer que el sistema lo ejecute al arranque para que tengamos música de fondo en nuestro menú. Esto lo haremos conectándonos de la manera habitual con PuTTY editando el archivo rc.local con el siguiente comando:

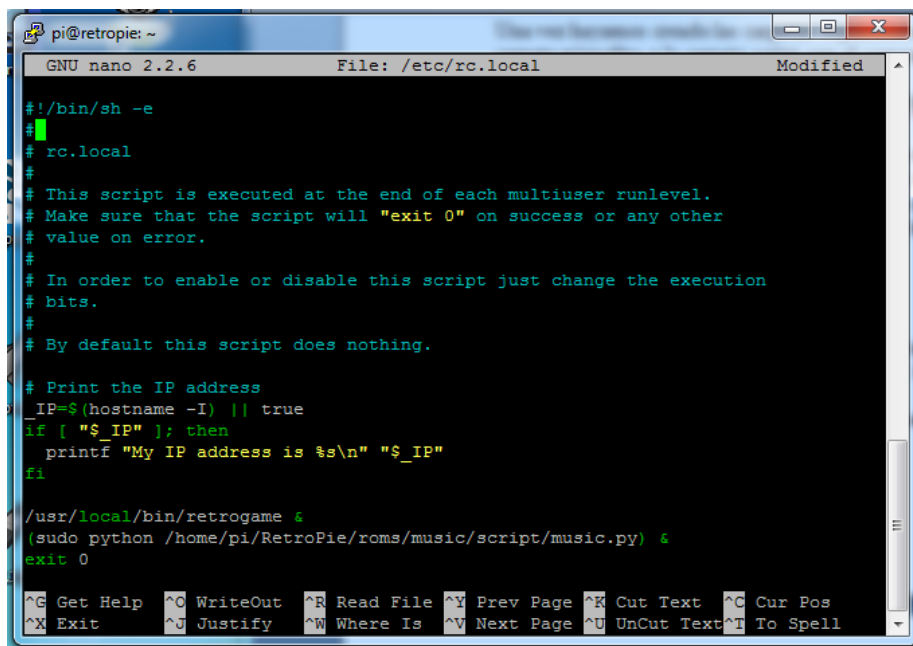


**sudo nano /etc/rc.local**

Una vez haya abierto el editor veremos que al final del todo hay una línea que pone "exit 0". Justo una línea por encima incluiremos lo siguiente:

**(sudo python /home/pi/RetroPie/roms/music/script/music.py) &**

Debería quedarnos algo así:



```
GNU nano 2.2.6 File: /etc/rc.local Modified
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
#
# Print the IP address
IP=$(hostname -I) || true
if [ "$IP" ]; then
  printf "My IP address is %s\n" "$IP"
fi
/usr/local/bin/retrogame &
(sudo python /home/pi/RetroPie/roms/music/script/music.py) &
exit 0

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Es importante mantener todo el código, con los paréntesis y todo. Una vez hecha la modificación para guardar el fichero pulsar **CTRL+O** y confirmar con la tecla **Intro**, seguidamente **CTRL+X** para salir. Al siguiente arranque conseguiremos que se ejecute el script y reproduzca las canciones que hemos pasado a la carpeta music. Si queremos reiniciar la Raspberry para probar si se ejecuta bien teclearemos:

**sudo reboot**

## *Programación Raspberry apagado pulsador y batería baja*

Para hacer que el script de apagados funcione primero debemos asegurarnos que el módulo python-rpi.gpio está instalado. Para ello teclearemos desde PuTTY los siguientes comandos:

**sudo apt-get update**

seguido de:

**sudo apt-get -y install python-rpi.gpio**

Una vez instalado el módulo podemos comprobar su funcionamiento introduciendo el siguiente comando:

**sudo python /home/pi/miguelito/miguelito/shutdown.py**

Una vez introducido veremos a través de PuTTY que el script se ejecuta y se queda esperando a la “escucha” de los dos GPIO especificados. Para parar el proceso pulsaremos **CTRL+C**.

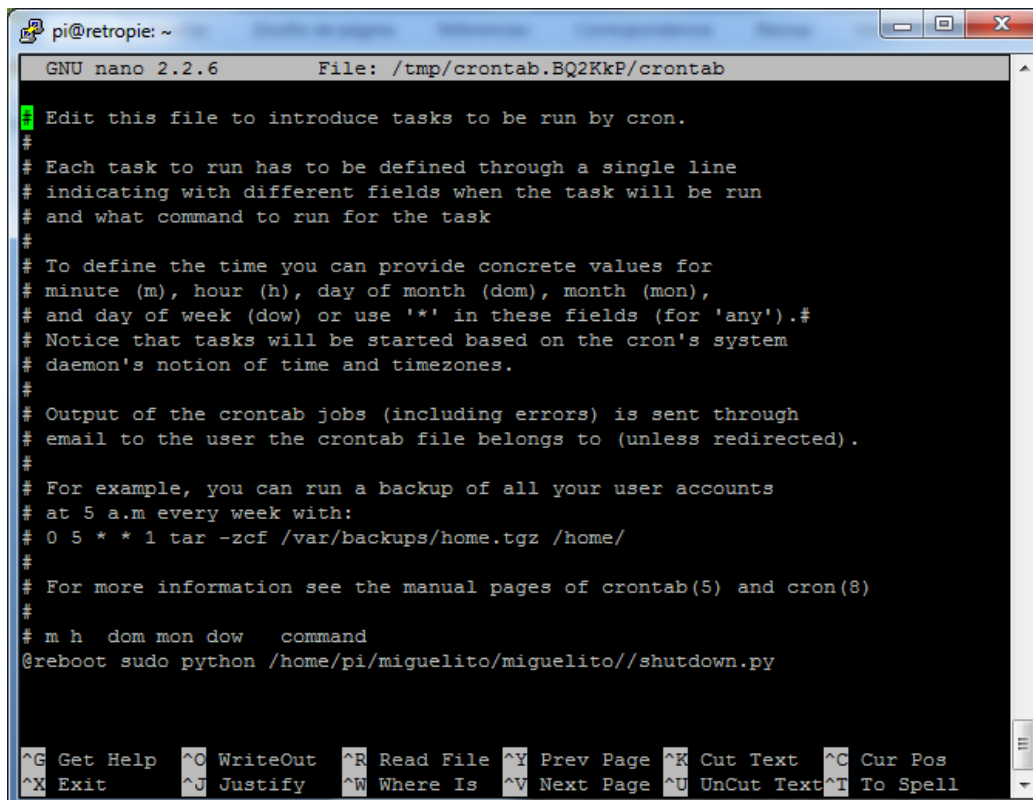
Para que se ejecute durante el arranque de manera automática debemos modificar crontab con el siguiente comando:

**crontab -e**

Nos dará varias opciones a elegir (son editores posibles con los que editar crontab). Elegiremos el 2 (nano) y bajaremos hasta el final del texto que aparece y escribiremos:

**@reboot sudo python /home/pi/miguelito/miguelito//shutdown.py**

Debería quedar algo como lo siguiente:



```
pi@retropie: ~
GNU nano 2.2.6 File: /tmp/crontab.BQ2KkP/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow command
@reboot sudo python /home/pi/miguelito/miguelito//shutdown.py

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

De nuevo, para guardar el fichero pulsar **CTRL+O** y confirmar con la tecla **Intro**, seguidamente **CTRL+X** para salir. Al siguiente arranque conseguiremos que se ejecute el script con lo que conseguiremos que se apague con botón a través del GPIO 5 y que se apague por batería baja a través del GPIO 27.

## *Otras configuraciones (opcionales)*

### *Ajustes de pantalla*

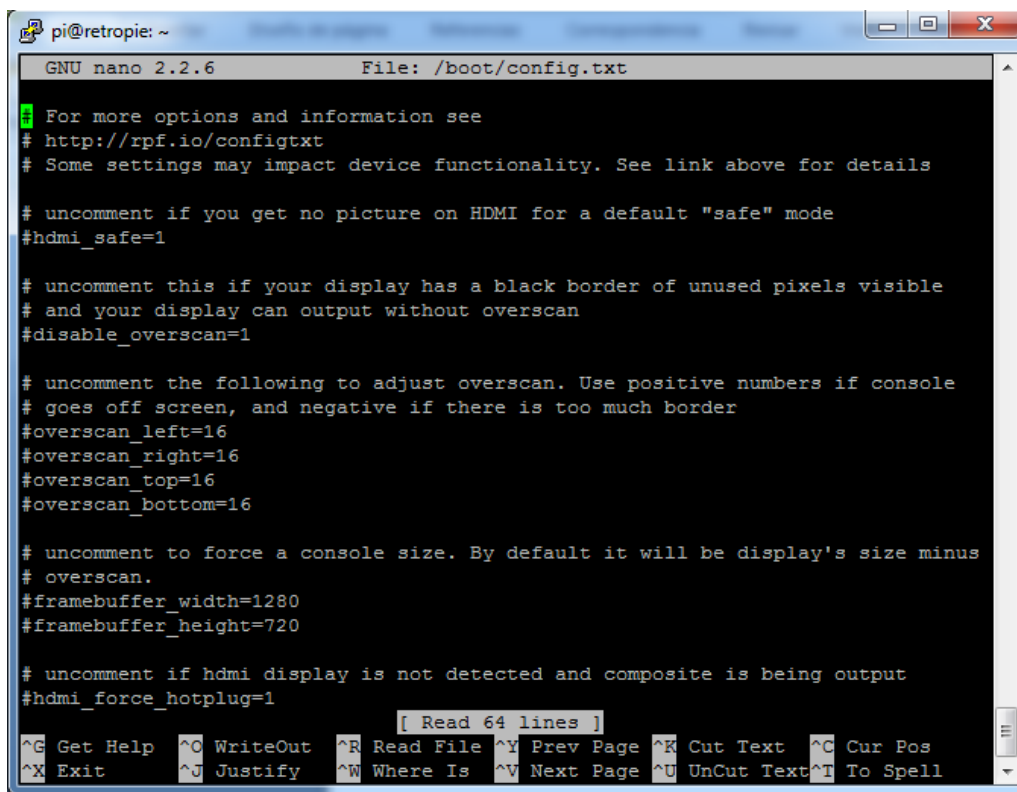
Nota: esta sección es única y exclusivamente si no vemos bien la pantalla de nuestra gameboy por ajuste del marco o algo parecido.

Hay casos en que se producen problemas como que alguna parte de la pantalla quede oculta, que la imagen aparezca muy pequeña, y en el peor de los casos, que no se vea la imagen. Un problema habitual en la gameboy es que al recortar el marco y después colocar el cristal queda parte de la imagen debajo del marco. Podemos modificar los parámetros de pantalla para solventar estos problemas.

Para cambiar los parámetros de la pantalla Raspberry Pi escribiremos en la línea de comandos:

**sudo nano /boot/config.txt**

Una vez haya arrancado el editor veremos que aparece una pantalla similar a la siguiente:



```
GNU nano 2.2.6 File: /boot/config.txt

# For more options and information see
# http://rpf.io/configtxt
# Some settings may impact device functionality. See link above for details

# uncomment if you get no picture on HDMI for a default "safe" mode
#hdmi_safe=1

# uncomment this if your display has a black border of unused pixels visible
# and your display can output without overscan
#disable_overscan=1

# uncomment the following to adjust overscan. Use positive numbers if console
# goes off screen, and negative if there is too much border
#overscan_left=16
#overscan_right=16
#overscan_top=16
#overscan_bottom=16

# uncomment to force a console size. By default it will be display's size minus
# overscan.
#framebuffer_width=1280
#framebuffer_height=720

# uncomment if hdmi display is not detected and composite is being output
#hdmi_force_hotplug=1

[ Read 64 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

En el archivo veremos que aparecen múltiples líneas precedidas del símbolo #, este símbolo indica que la línea se ignora y por lo tanto no se tiene en cuenta su valor (al arrancar Raspberry Pi no se lee esta línea), si elimináis este símbolo de una línea Raspberry Pi al arrancar la próxima vez tendrá en cuenta este parámetro y usará el valor introducido en lugar de la configuración estándar. Seguidamente se muestra el significado de las líneas que tienen que ver con nuestro proyecto:

- **overscan\_left**: mueve la imagen hacia adentro desde el lado izquierdo, el número de píxeles indicado.
- **overscan\_right**: mueve la imagen hacia adentro desde el lado derecho, el número de píxeles indicado.
- **overscan\_top** : mueve la imagen hacia adentro desde el lado superior, el número de píxeles indicado.
- **overscan\_bottom**: mueve la imagen hacia adentro desde el lado inferior, el número de píxeles indicado.

- **display\_rotate:** rota la imagen en la pantalla para aquellos casos en que haya que darle la vuelta a la imagen de la pantalla (dependiendo de qué pantalla instalemos hay que girar la imagen). Los valores para este parámetro son 0 configuración por defecto, 1 configuración a 90°, 2 configuración a 180°, 3 configuración a 270°, 0x10000 para voltear la imagen horizontalmente y 0x20000 para voltear la imagen verticalmente. Lo más normal es que si tenéis que darle la vuelta sean 180° por lo que el parámetro quedará del modo `display_rotate=2`

Una vez hechas las modificaciones para guardar el fichero pulsáis **CTRL+O** y confirmar con la tecla **Intro**, seguidamente **CTRL+X** para salir. Para aplicar los cambios hay que reiniciar la Raspberry con el comando:

**sudo reboot**

## *Personalizacion script música de fondo*

El script de música de fondo se puede personalizar de diversas maneras. Estas son las opciones más significativas para la personalización, las cuales están justo al principio del script en la sección **CONFIG SECTION**:

- **startdelay:** son los segundos que tarda en empezar a reproducir la primera canción durante el arranque.
- **maxvolume:** es el volumen máximo al que sonarán las canciones expresado en porcentaje
- **volumefadespeed:** es lo que tarda la canción en dejar de sonar progresivamente el volumen hasta cero. Se utiliza para que la parada de la canción no sea tan brusca.
- **restart:** Si está como True hará que el script la música progresivamente y detenga la canción en lugar de pausarla. False hará lo contrario.
- **startsong:** si queremos que arranque siempre con una canción concreta pondremos el nombre exacto y su extensión entre las comillas. Por ejemplo, si tenemos una canción llamada inicio.mp3 quedará como "inicio.mp3" .

Por otro lado hay una serie de líneas más abajo que empiezan por la palabra **print**. Éstas líneas son las que hacen que salgan una serie de mensajes en la pantalla, como por ejemplo al reproducir una canción o al detectar un emulador. Esto se puede evitar si se desea comentando dicha línea poniendo **#** justo antes de **print**. Así, por ejemplo, si no deseamos que salga el mensaje de emulador detectado al iniciar un juego quedaría de la siguiente manera:

**#print "Emulator found! " + procname[:-1] + " Muting the music..."**

## *Configuración wifi manualmente*

Si deseamos configurar la conexión wifi de la raspberry pero no tenemos opción de conectar un teclado por usb es posible hacerlo con un sistema operativo Linux. Bastará con entrar vía terminal de linux a la carpeta /etc/wpa\_supplicant/ en la tarjeta microSD y editar el archivo de configuración con el siguiente comando:

```
sudo nano wpa_supplicant.conf
```

Al abrir el editor veremos algo así:

```
network={
    ssid="ssid"
    psk="password"
}
```

Donde ssid es el nombre de tu red y password es tu clave de red. Como siempre para guardar el archivo **CTRL+O**, confirmando con **Intro** y después salir con **CTRL+X**.

Si no disponemos de un Linux a mano podemos hacer esta operación bajo Windows. Para crear estos archivos de configuración debemos insertar la microSD para que Windows reconozca las particiones (una de ellas sin formato porque es Linux). Solo se debe cancelar (nunca formatear), pero una de estas la reconocerá como boot y la montará como un disco. Simplemente editaremos un wpa\_supplicant.conf con el bloc de notas y teclearemos lo mismo que antes. Al guardarlo lo llamaremos wpa\_supplicant.conf. Una vez guardado lo pasaremos a la partición /boot de la microSD.

## *Activar ssh manualmente*

Si lo que deseamos es activar el ssh manualmente simplemente crearemos un archivo llamado ssh sin nada dentro y lo pondremos dentro de la partición /boot. Esto podremos hacerlo con un sistema Linux o Windows indiferentemente. Con esto, en el siguiente arranque activaremos la comunicación por ssh.

# *Resumen de comandos a realizar para la programación*

Para realizar los siguientes pasos es necesario conectar a través de un terminal ssh como por ejemplo PuTTY:

## *Descarga del contenido:*

**git clone <https://github.com/julenvitoria/miguelito>**

## *Modificación del teclado al idioma español:*

**sudo nano /etc/default/keyboard**

y sustituir XKBLAYOUT="gb" por XKBLAYOUT="es"

Guardar con **CTRL-O**, confirmar con **Intro** y salir con **CTRL-X**.

## *Audio por pwm:*

**sudo nano /boot/config.txt**

al final del archivo escribir:

**#Audio por pwm**

**dtoverlay=pwm-2chan,pin=18,func=2,pin2=13,func2=4**

Guardar con **CTRL-O**, confirmar con **Intro** y salir con **CTRL-X**.

Opcional: para poner la salida de 3.5mm como la predeterminada seleccionarlo desde sección raspi-config en menú retropie, sección audio en menú retropié o manualmente:

**sudo nano /etc/modules**

y añadir la línea:

**sudo amixer cset numid=3 1**

Guardar con **CTRL-O**, confirmar con **Intro** y salir con **CTRL-X**.

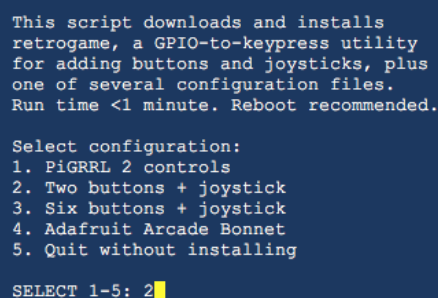
## *Script retrogame para el control de botones y cruceta:*

**curl -O <https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Installer-Scripts/master/retrogame.sh>**

y después:

**sudo bash retrogame.sh**

Una vez hecho lo anterior nos aparecerá la siguiente pantalla:



```
This script downloads and installs
retrogame, a GPIO-to-keypress utility
for adding buttons and joysticks, plus
one of several configuration files.
Run time <1 minute. Reboot recommended.

Select configuration:
1. PiGRRL 2 controls
2. Two buttons + joystick
3. Six buttons + joystick
4. Adafruit Arcade Bonnet
5. Quit without installing

SELECT 1-5: 2
```



Seleccionar “Six buttons + joystick”. Después pedirá reiniciar. Confirmar y reiniciar.  
Una vez arrancada de nuevo:

```
sudo cp -R /home/pi/miguelito/miguelito/retrogame.cfg /boot
```

### **Instalar el tema para la pantalla de 3,5”:**

Para tema gbz35

```
sudo mv /home/pi/miguelito/miguelito/gbz35 /etc/emulationstation/themes/
```

Para tema snes-mini

```
sudo mv /home/pi/miguelito/miguelito/snes-mini /etc/emulationstation/themes/
```

Para tema showcase

```
sudo mv /home/pi/miguelito/miguelito/showcase /etc/emulationstation/themes/
```

Después seleccionar el tema en emulationstation pulsar start ir a “UI-Settings”, “Theme Set” y seleccionar “gbz35”, “snes-mini” o “showcase”. Al volver al menú principal se cambiará.

Si se desea liberar espacio borrando el tema que no vayamos a usar teclear:

Para tema gbz35

```
sudo rm -R /home/pi/miguelito/miguelito/gbz35
```

Para tema snes-mini

```
sudo rm -R /home/pi/miguelito/miguelito/snes-mini
```

Para tema showcase

```
sudo rm -R /home/pi/miguelito/miguelito/showcase
```

### **Splashscreen gameboy**

Mover el video de la intro:

```
sudo mv /home/pi/miguelito/miguelito/GBZ-Miguelito.mp4  
/home/pi/RetroPie/splashscreens
```

Copiar el archivo siguiente con el comando:

```
sudo mv /home/pi/miguelito/miguelito/splashscreen.list /etc
```

En el siguiente arranque ya tendremos nuestro splashscreen reproduciéndose.

### **Script de música de fondo:**

Actualizar paquetes:

```
sudo apt-get update
```

y después instalar el módulo:

```
sudo apt-get -y install python-pygame
```

Crear carpetas con:

```
mkdir /home/pi/RetroPie/roms/music
```

y

```
mkdir /home/pi/RetroPie/roms/music/script
```

Copiar el script a su directorio con:

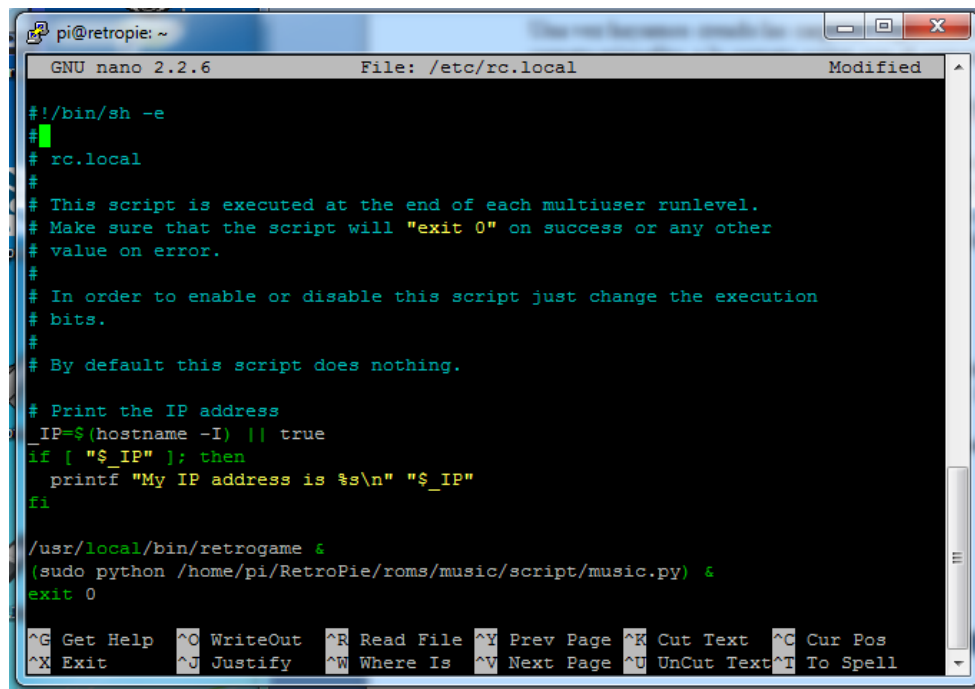
```
cp /home/pi/miguelito/miguelito/music.py /home/pi/RetroPie/roms/music/script
```

Programar el script al arranque de la zero con:

**sudo nano /etc/rc.local**

Al final del archivo pone **exit 0**. Justo una línea por encima escribir:

**(sudo python /home/pi/RetroPie/roms/music/script/music.py) &**



```
GNU nano 2.2.6 File: /etc/rc.local Modified
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
#
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi

/usr/local/bin/retrogame &
(sudo python /home/pi/RetroPie/roms/music/script/music.py) &
exit 0

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Guardar con **CTRL-O**, confirmar con **Intro** y salir con **CTRL-X**.

Pasar las canciones que deseemos (en formato mp3 u ogg) a la carpeta music que se encuentra dentro de roms. Se puede hacer con el explorador de Windows, WinSCP, Filezilla o cualquier programa para transferir archivos por ftp.

Reiniciar con

**sudo reboot**

Al siguiente arranque deberían sonar las canciones que hayamos pasado a music.

### **Apagado por pulsador y batería baja:**

Actualizar paquetes:

**sudo apt-get update**

Instalar el módulo:

**sudo apt-get -y install python-rpi.gpio**

Comprobar su funcionamiento con:

**sudo python /home/pi/miguelito/miguelito/shutdown.py**

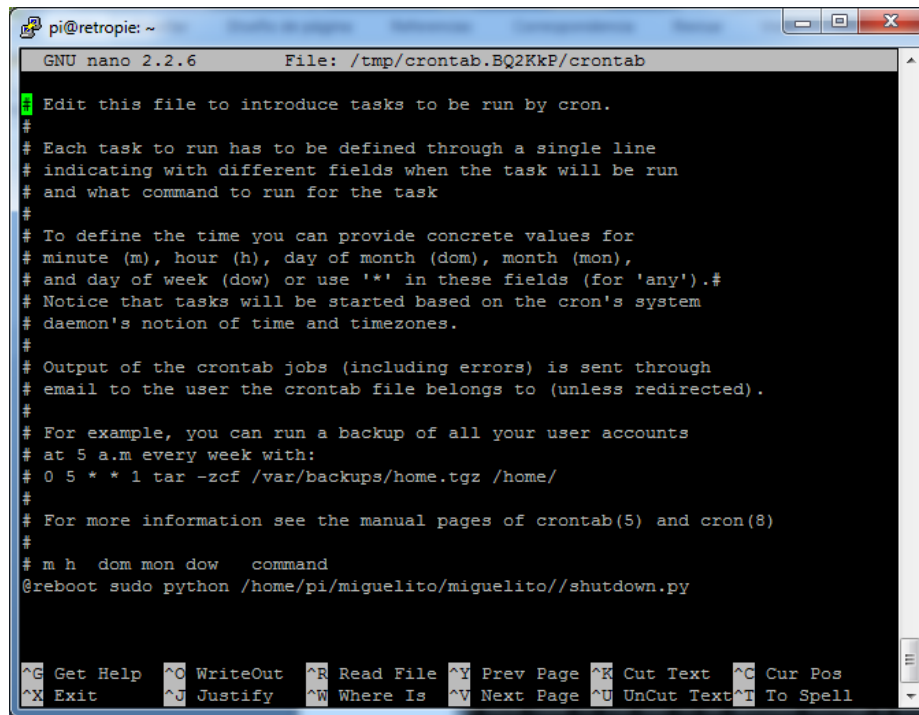
Con esto veremos que el script se ejecuta. Para pararlo pulsar **CTRL-C**.

Programar en el arranque con:

**crontab -e**

Al final del todo escribir:

**@reboot sudo python /home/pi/miguelito/miguelito//shutdown.py**



```
pi@retropie: ~  
GNU nano 2.2.6      File: /tmp/crontab.BQ2KkP/crontab  
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow   command  
@reboot sudo python /home/pi/miguelito/miguelito//shutdown.py  
  
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos  
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

Guardar con **CTRL-O**, confirmar con **Intro** y salir con **CTRL-X**.

Reiniciar con

**sudo reboot**

y comprobar que se ha ejecutado al arranque pulsando el botón de apagado.