1. Guia instalacion raspeberry radio

* Instalar programas:

**sudo apt-get install mpc mpd**

* Probar sonido

**mpc clear** -> limpia lista

**mpc playlist** -> muestra lista

**mpc add http://mp3.somafm.com:80**  -> añade url a la lista

**mpc play** -> reproduce

**mpc stop** -> para

**mpc save Chillout** -> guarda lista

**mpc load News** -> carga lista

Path playlists para edición a mano:

**/var/lib/mpd/playlists**

Las listas de reproduccion que se tengan al ejecutar **"mpc ls"**

deben coincidir con los que haya en el codigo python:

* 1. ALGUNAS EMISORAS:
     1. Spoken ingles

<http://nprdmp.ic.llnwd.net/stream/nprdmp_live01_mp3>

<http://bbcwssc.ic.llnwd.net/stream/bbcwssc_mp1_ws-eieuk>

* + 1. Chill

[http://mp3.somafm.com:80](http://mp3.somafm.com:80/)

<http://pub4.di.fm:80/di_chillout>

<http://pub2.di.fm:80/di_vocalchillout>

<http://pub2.di.fm:80/di_lounge>

[http://sfstream1.somafm.com:2800](http://sfstream1.somafm.com:2800/)

<http://pub4.di.fm:80/di_ambient>

<http://pub2.di.fm:80/di_chilloutdreams>

<http://pub2.di.fm:80/di_psychill>

* + 1. Maquineta

<http://pub6.di.fm:80/di_vocaltrance>

<http://pub2.di.fm:80/di_trance>

<http://pub3.di.fm:80/di_eurodance_aacplus>

<http://pub2.di.fm:80/di_classictrance>

<http://pub3.di.fm:80/di_epictrance>

<http://pub2.di.fm:80/di_classicvocaltrance>

<http://pub2.di.fm:80/di_classiceurodance>

<http://pub1.di.fm:80/di_classiceurodisco_aacplus>

* + 1. DI

http://pub5.di.fm:80/di\_breaks

http://pub3.di.fm:80/di\_chiptunes\_aacplus

http://pub2.di.fm:80/di\_oldschoolelectronica

http://pub3.di.fm:80/di\_clubdubstep

http://pub1.di.fm:80/di\_clubsounds

http://pub2.di.fm:80/di\_cosmicdowntempo

http://pub2.di.fm:80/di\_darkdnb

http://pub3.di.fm:80/di\_deephouse

http://pub2.di.fm:80/di\_deepnudisco

http://pub1.di.fm:80/di\_deeptech\_aacplus

http://pub2.di.fm:80/di\_discohouse

http://pub3.di.fm:80/di\_djmixes

http://pub3.di.fm:80/di\_drumandbass

http://pub2.di.fm:80/di\_dubstep

http://pub2.di.fm:80/di\_electrohouse

http://pub2.di.fm:80/di\_funkyhouse

http://pub2.di.fm:80/di\_futuresynthpop

http://pub1.di.fm:80/di\_glitchhop

http://pub2.di.fm:80/di\_goapsy

http://pub1.di.fm:80/di\_handsup

http://pub2.di.fm:80/di\_hardcore

http://pub2.di.fm:80/di\_harddance

http://pub2.di.fm:80/di\_hardstyle

http://pub2.di.fm:80/di\_house

http://pub2.di.fm:80/di\_latinhouse

http://pub2.di.fm:80/di\_liquiddnb

http://pub1.di.fm:80/di\_liquiddubstep

http://pub2.di.fm:80/di\_minimal

http://pub2.di.fm:80/di\_oldschoolacid

http://pub2.di.fm:80/di\_progressive

http://pub1.di.fm:80/di\_progressivepsy

http://pub3.di.fm:80/di\_soulfulhouse\_aacplus

http://pub2.di.fm:80/di\_spacemusic

http://pub2.di.fm:80/di\_techhouse

http://pub3.di.fm:80/di\_techno\_aacplus

http://pub2.di.fm:80/di\_tribalhouse

http://pub1.di.fm:80/di\_ukgarage\_aacplus

1. CODIGO SCRIPT PYTHON

|  |
| --- |
| import os  import time  import RPi.GPIO as GPIO  GPIO.setmode(GPIO.BCM)  pI = 0  playLists = [ "News" ,"Chillout" ,"Trance","DI\_resto" ]  playing = 0  haltCount = 0  def prevt ( ):  print "prevt"  os.system("mpc prev")  global haltCount  haltCount = 0  def nextt ( ):  print "next"  os.system("mpc next")  global haltCount  haltCount = 0  def plist ( ):  print "plist"  global pI  pI = pI -1;  if ( pI < 0 ) :  pI = len(playLists)-1  os.system("mpc clear")  os.system("mpc load " + playLists[pI] )  os.system("mpc play")  global playing  playing = 1  global haltCount  haltCount = 0  def nlist ( ):  print "nlist"  global pI  pI = pI + 1  if ( pI >= len(playLists) ) :  pI = 0  os.system("mpc clear")  os.system("mpc load " + playLists[pI] )  os.system("mpc play")  global playing  playing = 1  global haltCount  haltCount = 0  def play\_stop ( ):  global playing  if ( not playing ):  os.system("mpc play")  playing = 1  print "play"  else:  os.system("mpc stop")  playing = 0  print "stop"  global haltCount  haltCount = 0  def down ( ):  print "down"  os.system("mpc stop")  global haltCount  haltCount = haltCount + 1  if ( haltCount == 7 ):  os.system("halt")    outButton = { 22 : ('prevt',prevt),  4 : ('nextt',nextt),  24 : ('plist',plist),  23 : ('nlist',nlist),  17 : ('play\_stop',play\_stop),  18 : ('down',down ) }    inputPins= outButton.keys()  inp = [ 0 ,0 ,0 ,0 ,0 ,0 ]  prev\_input =[ 0 ,0 ,0 ,0 ,0 ,0 ]  for i in range(0,len(inputPins)):  GPIO.setup(inputPins[i],GPIO.IN)  print ("radior")  while True:  for i in range(0,len(inputPins)):  #take a reading  inp[i] = GPIO.input(inputPins[i])  #if the last reading was low and this one high, print  if ( (not prev\_input[i]) and ( inp[i] ) ):  print(inputPins[i])  (outButton[inputPins[i]][1])()  prev\_input[i] = inp[i]  time.sleep(0.10) |

1. DIAGRAMA CONEXION





