**Ajustar resoluçao do monitor:**

1) Abrir o arquivo config.txt

$sudo nano /boot/config.txt

2) Descomentar a linha que diz #hdmi safe 1

**Configurar wifi LSA**

**Fonte:** [**https://www.raspberrypi.org/documentation/configuration/wireless/wireless-cli.md**](https://www.raspberrypi.org/documentation/configuration/wireless/wireless-cli.md)

1)Abrir o arquivo de configuração wpa-supplicant e botar o login e senha da rede no final do arquivo:

$sudo echo $'network={\n\tssid="lsa"\n\tpsk="isaacasimov\n}' >> /etc/wpa\_supplicant/wpa\_supplicant.conf

2) Plugar o modulo wifi e reiniciar a rasp.

**Se a conexão SSH não estiver funcionando (Connection refused):**

1) Reiniciar o ssh (na rasp)

$sudo service ssh restart

**Desativa o login serial da Raspberry (Para evitar o conflito da serial do Arduino)**

**Source: [http://spellfoundry.com/sleepy-pi/setting-arduino-ide-raspbian/#Reboot](http://spellfoundry.com/sleepy-pi/setting-arduino-ide-raspbian/" \l "Reboot)**

1. Desativando o serial login

$sudo systemctl mask serial-getty@ttyAMA0.service

~~Step2. desativando o boot info~~

Step3. Linkando a porta ama0 na aduino ide da rasp

**Acho que vamos ter que usar o passo "Setting up the Reset (DTR) pin" também, não fiz ele pq faltou tempo.**

**Segue o link.**

**Download do Player**

Obs: Baixar o player pelo seguinte link.

<https://github.com/lsa-pucrs/Player>

**Instalar Player**

**Fonte:** [**https://web.archive.org/web/20150206050236/http://fritz-hut.com/2014/01/20/installing-player-slash-stage-on-the-raspberry-pi/**](https://web.archive.org/web/20150206050236/http://fritz-hut.com/2014/01/20/installing-player-slash-stage-on-the-raspberry-pi/)

1) Instalar dependencias:

sudo apt-get install cmake

sudo apt-get install g++

sudo apt-get install swig

sudo apt-get install libboost-dev

sudo apt-get install libboost-all-dev

sudo apt-get install libltdl-dev

sudo apt-get install libjpeg-dev

sudo apt-get install libgstreamer0.10-dev

sudo apt-get install libstatgrab-dev

sudo apt-get install libusb-dev

sudo apt-get install libasound2-dev

sudo apt-get install libgsl0-dev

sudo apt-get install libxmu-dev

sudo apt-get install libxi-dev

sudo apt-get install libfltk1.1-dev

sudo apt-get install libopencv-contrib-dev

sudo apt-get install libcv-dev

sudo apt-get install freeglut3-dev

sudo apt-get install libopencv-dev

sudo apt-get install libpq-dev

\*pacotes retirados do tutorial do amory (não estão no tutorial indicado como source).

2) Baixar o fonte do player:

$wget http://downloads.sourceforge.net/project/playerstage/Player/3.0.2/player-3.0.2.tar.gz -O player.tar.gz

3) Baixar o patch “player\_3.0.2\_14.04.patch”(no link abaixo) e aplicar ele no player:

<https://github.com/lsa-pucrs/Player/raw/master/patch/install/player_3.0.2_14.04.patch>

1. mover o patch para dentro da pasta do player. “player-3.0.2”
2. aplicar o patch:

$patch -p1 --verbose < ../player\_3.0.2\_14.04.patch

4) Compilar e instalar o player

$cd ~/player-3.0.2

$mkdir build

$cd build

$cmake ../

$make

$sudo make install

**Em caso de Erro: Possiveis problemas e suas soluções:**

1. Problemas com falta de pacotes, pacotes quebrados ou parou de funcionar o apt-get install:
2. $sudo pat-get install --reinstall nomeDoPacoteIndicadoPeloTerminal

4) Botar as seguintes linhas no ponto .bashrc:

#Player/Stage  
export PATH=$PATH:"/usr/local/lib64"  
export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:"/usr/local/lib":"/usr/local/lib64"  
gmailexport PLAYERPATH="/usr/local/lib":"/usr/local/lib64"  
export STAGEPATH="/usr/local/lib":"/usr/local/lib64”

export CMAKE\_MODULE\_PATH=$CMAKE\_MODULE\_PATH:/usr/local/share/cmake/Modules

**Instalar OpenCV**

1. Instalar estes pacotes:

$sudo apt-get install build-essential

$sudo apt-get install cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev

$sudo apt-get install python-dev python-numpy libtbb2 libtbb-dev libjpeg-dev libpng-dev libtiff-dev libjasper-dev libdc1394-22-dev

2. Baixar ultima versão do OpenCV(3.1 foi a usada neste tutorial)

cd ~/

git clone <https://github.com/Itseez/opencv.git> opencv

3.Compilar o OpenCV:

$cd ~/opencv

$mkdir release

$cd release

$cmake -DWITH\_WEBP=OFF -D CMAKE\_BUILD\_TYPE=RELEASE -D CMAKE\_INSTALL\_PREFIX=/usr/local ..

$make

$sudo make install

4. Testar OpenCV:

TODO - botar arquivos em algum repositório

$cd ~/opencv

$mkdir test

$cd test

$TODO Baixar os aquivos indicados nesta pasta

$ccmake .

$make

$wget http://imagens.kboing.com.br/papeldeparede/14262gatinhos2.jpg

$mv 14262gatinhos2.jpg image.jpg

$./DisplayImage imagem.jpg

Obs: É para abrir a imagem em uma nova janela. Se estiver executando ssh use o parametro -X. Ex: ssh -X 192.1…

**Instalar a Raspcam**

**Obs:** A ordem importa! O correto é instalar primeiro o OpenCV e depois a Raspcam

1)Baixar ultima versão da Raspcam no site [**https://sourceforge.net/projects/raspicam/files/**](https://sourceforge.net/projects/raspicam/files/)

2) Descompactar e compilar

$tar xvzf raspicamXX.tgz

$cd raspicamXX

$mkdir build

$cd build

$cmake ..

$make

$sudo make install

3) Para testar a Raspcam

$mkdir build

$cd build

$cmake ..

$make

$./simpletest\_raspicam\_cv

Obs: Se quiser visualizar o video da Raspcam na tela (sem ser por ssh) Podeusar o código abaixo para testa.

$raspivid -o - -t 9999999 -w 1280 -h 1024 -b 500000 -fps 20 -vf

**Atenção:** **A ordem importa! O correto é instalar primeiro o OpenCV e depois a Raspcam**

**Instalar o TIMIDITY para poder usar os canais virtuais de MIDI (Virtual MIDI Port) e usar notas musicais no autofalante**

$sudo apt-get install timidity

**Configurando o desligamento da raspberry (Shutdown) por interrupção do pino GPIO4**

Referencia(No link tem outros uso para a GPIO Zero também)**:** [**http://bennuttall.com/gpio-zero-developing-a-new-friendly-python-api-for-physical-computing/**](http://bennuttall.com/gpio-zero-developing-a-new-friendly-python-api-for-physical-computing/)

1)Instalar libs para controle da gpio

$sudo apt-get install python3-gpiozero python-gpiozero

2) Criar pasta para o script:

$mkdir ~/scripts

$nano /home/pi/scripts/softshutdown.py

1. Colar e salvar o seguinte script:

##############################################################

import os

from gpiozero import Button

from signal import pause

button = Button(4,pull\_up=False)

def pressed(button):

print("Pin %s pressed. The system is going to try shutdown now!" % button.pin)

os.system("shutdown now -h")

button.when\_pressed = pressed

pause()

##############################################################

3)Definir execução do script após o boot:

$sudo nano /etc/rc.local

1. Adicionar a seguinte linha antes da linha “exit 0”:

**sudo python /home/pi/scripts/softshutdown.py &**

b) Salvar o arquivo (Ctrl+X) e depois (Ctlr+Y)

4) Reiniciar a rasp e apartir de agora cada vez que o pino ficar em nivel alto a raspberry ira desligar.

**Configurando o festival para falar o IP ao ligar**

1) Instalar o festival (se ja não estiver instalado)

$sudo apt-get install festival

2)Criar pasta para o script

$nano /home/pi/scripts/speechPI.py

1. Colar e salvar o seguinte script:

##############################################

#!/usr/bin/python2.7

import subprocess

ipNumber = subprocess.check\_output("ip route get 8.8.8.8 | awk '{print $NF; exit}'", shell=True)

ipNumber = ipNumber.split('\n')[0].split('.')

msg = "My final ip is " + ipNumber[3]

subprocess.call('echo '+msg+'|festival --tts', shell=True)

subprocess.call('echo '+', repeating... is '+ ipNumber[3]+'|festival --tts', shell=True)

##############################################

3)Definir execução do script após o boot:

$sudo nano /etc/rc.local

1. Adicionar a seguinte linha antes da linha “exit 0”:

**$ sudo python /home/pi/scripts/speechIP.py &**

b) Salvar o arquivo (Ctrl+X) e depois (Ctlr+Y)

4) Reiniciar a rasp e apartir de agora cada vez ligar o final do ip sera ditado.

**Copiando o SD da Rasp**

1) Botar o SD no PC e descobrir nome das partições

a) $df -h

2) Desmontar Partições

a) $sudo umount /dev/sdd\*

3) Copiar imagem (das duas partições da rasp, a BOOT e a de dados) para o pc

Obs: Copiar em partições é a solução que achamos para o comando dd não criar uma unica imagem de 8gb (todo tamanho do cartão)

$sudo dd bs=4M if=/dev/sdd1 of=/home/lsa/Desktop/raspBootXX.img

$sudo dd bs=4M if=/dev/sdd2 of=/home/lsa/Desktop/raspDataXX.img

**Gravando a imagem para um cartão SD**

1) Criar 2 partições uma **fat16 de 10MB (primaria)** para o boot e outra **ext4 4GB (lógica) para os dados**

2) Desmontar Partições

a) $sudo umount /dev/sdd\*

3)Passar imagem para SD.

Obs: Este exemplo esta com a partição sdd1=Boot e sdd2=Main

Obs2: Para verificar as partições e o tamanho usar $**df -h**

$sudo dd bs=4M if=~/raspDonnieBoot-XX.img of=/dev/sdd1

$sudo dd bs=4M if=~/raspDonnieMain-XX.img of=/dev/sdd2

4) Expandir a partição de dados no Gparted

**Apendices**

======================================================================

**Visualizando imagens por commando**

Fonte: <http://www.linuxcircle.com/2015/05/14/view-image-on-your-pi-remotely-over-ssh/>

~~1) Instalar gpicview~~

~~$sudo apt-get install gpicview~~

~~2)Abrir imagem:~~

~~$gpicview imagem.jpg~~

**Raspcam por ssh**

<https://www.raspberrypi.org/forums/viewtopic.php?t=67571>

<http://raspi.tv/2013/how-to-stream-video-from-your-raspicam-to-your-nexus-7-tablet-using-vlc>

1) Precisa do vlc (para o comando cvlc)

$sudo apt-get install vlc

2) Executar processo da camera

$raspivid -o - -t 99999 -w 640 -h 360 -fps 5 -vf|cvlc -vvv stream:///dev/stdin --sout '#standard{access=http,mux=ts,dst=:8090}' :demux=h264

Obs: Para conectar entrar no endereço (pelo streaming do VLC no PC) http://192.168.0.XX:8090

**OpenCV com Pip picampera**

<http://www.pyimagesearch.com/2015/02/23/install-opencv-and-python-on-your-raspberry-pi-2-and-b/>

<http://www.pyimagesearch.com/2015/03/30/accessing-the-raspberry-pi-camera-with-opencv-and-python/>

**Gravando audio na Rasp**

Fonte: <http://mutsuda.com/2012/09/07/raspberry-pi-into-an-audio-spying-device/>

1) Conectar microfone USB na rasp e executar o seguinte comando

$sudo modprobe snd\_bcm2835

2) Gravar Audio

$arecord -D plughw:1,0 test.wav

3) Reproduzir

$aplay test.wav

Obs: Se o volume ficar baixo é só aumentar a captação no alsa:

$alsamixer

E salvar:

$sudo alsactl store

**Conexão entre arduino e raspberry**

<http://blog.oscarliang.net/raspberry-pi-and-arduino-connected-serial-gpio/>

**Raspberry ros**

<http://www.instructables.com/id/Raspberry-Pi-and-ROS-Robotic-Operating-System/?lang=pt&ALLSTEPS>

**Raspberry tutorial instalaçao**

<http://www.raspberrypi.org/documentation/installation/installing-images/linux.md>

**Python executando comandos do linux**

<http://www.cyberciti.biz/faq/python-execute-unix-linux-command-examples/>

**Instalando o Player na Raspberry (Raspbian):**

Fonte: <https://web.archive.org/web/20150206050236/http://fritz-hut.com/2014/01/20/installing-player-slash-stage-on-the-raspberry-pi/>

**Shell Script para fazer o backup do SD em um Pendrive na própria Rasp**

<https://github.com/billw2/rpi-clone>

**Post de discussão sobre aplicações leves para Rasp**

<https://www.raspberrypi.org/forums/viewtopic.php?f=53&t=18946>

**Conectando via ssh (para Raspbian):**

- achar o nome “raspberry” na lista dhcp

$ssh pi@>IP<

senha: raspberry

**Testando saida do auto falante**

$speaker-test -t sine -f 1000 -c 2

**Selecionar a saida para o jack p2**

sudo amixer cset numid=3 1 # headphones

~~sudo amixer cset numid=3 0 # Auto~~

~~sudo amixer cset numid=3 2 # HDMI~~

**Controle do volume**

sudo alsamixer

**Tocando um mp3**

omxplayer mp3name.mp3