**Installation and Configuration**

This document shows the additional tools and configurations that the Raspberry needs for the execution of the BionicKitchen project. It is explained step by step the installation of the necessary tools and respective configurations, for easy understanding and optimum functionality.

**Content**

[**Content** 2](#_Toc501093960)

[**Installation of LAMP SERVER** 3](#_Toc501093961)

[**Installation Kivy on Raspberry Pi** 4](#_Toc501093962)

[**Configuration of user** 5](#_Toc501093963)

[**Automate turning off/on** 5](#_Toc501093964)

[**Run GUI at start-up** 7](#_Toc501093965)

[**Re-installation Raspbian on Raspberry Pi** 9](#_Toc501093966)

**\***NOTE: The steps below are for Raspbian Jessie OS. The following steps (in green letters) are execute in the OS terminal.

# **Installation of LAMP SERVER**

LAMP is the acronym used to describe an internet infrastructure system that uses the following tools.

|  |  |
| --- | --- |
| **L** | Linux (OS) |
| **A** | Apache (Web server) |
| **M** | MySQL or MariaDB (Database) |
| **P** | PHP, Python or Pearl (Programming languages) |

**Manual installation**

* Install updates:

sudo apt-get update && sudo apt-get upgrade

* Install Apache2:

sudo apt-get install apache2 apache2-utils

* Install MySQL server

sudo apt-get install mysql-server

* Install PHP 7.0:

sudo apt-get install php7.0 php7.0-mysql

* Install PHP-My-Admin:

sudo apt-get install phpmyadmin

5.1 In this installation, the terminal show you a message and you need the choose the follow:

apache2

5.2 Next step, you choose if you want to configure database for phpmyadmin with dbconfig-common and next you need choose a password:

yes

* Configure Apache to work with PHP-My-Admin:  
   sudo nano /etc/apache2/apache2.conf

6.1 And include this to end the file:

Include /etc/phpmyadmin/apache.conf

ctrl + c –-- to save

ctrl + x –-- to close

* 1. Restart Apache:

/etc/init.d/apache2 restart

* Access in phpmyadmin at root:

sudo –s

mysql -u root -p

mariadb: USE mysql;

mariadb: SELECT User, Host, plugin FROM mysql.user;

7.1 If root col plugin show you unix\_socket or anything you need the follow next:

mariadb: UPDATE user SET plugin=’’ WHERE User=’root’;

mariadb: FLUSH PRIVILEGES;

7.2 And you need a set new password for root:

mariadb: SET PASSWORD FOR ‘root’@’localhost’ = PASSWORD(‘<password>’);

# **Installation Kivy on Raspberry Pi**

**Manual installation**

1. Install the dependencies:

sudo apt-get update

sudo apt-get install libsdl2-dev libsdl2-image-dev libsdl2-mixer-dev libsdl2-ttf-dev \pkg-config libgl1-mesa-dev libgles2-mesa-dev \

python-setuptools libgstreamer1.0-dev git-core \

gstreamer1.0-plugins-{bad,base,good,ugly} \

gstreamer1.0-{omx,alsa} python-dev libmtdev-dev \

xclip

1. Install a new enough version of Cython:

sudo pip install -U Cython==0.25.2

1. Install Kivy globally on your system:

sudo pip install git+https://github.com/kivy/kivy.git@master

# **Configuration of user**

**Manual installation**

1. For create a user:

sudo adduser <*username*>

1. Need add the new user to list of root:

sudo visudo

2.1 add in this line User privilege specification:

<*username*> ALL=(ALL:ALL) ALL

ctrl + c –-- to save

ctrl + x –-- to close

1. To eliminate the old user you need change the user and in the terminal put the follow:

sudo -s

userdel –f <*username*>

3.1 if you cant delete it, in the screen showing the process PID and you make the follow:

pkill <*pid*> or pkill -f <*pid*>

1. Eliminate the home folder of old user:

sudo rm -rf /home/<*username*>

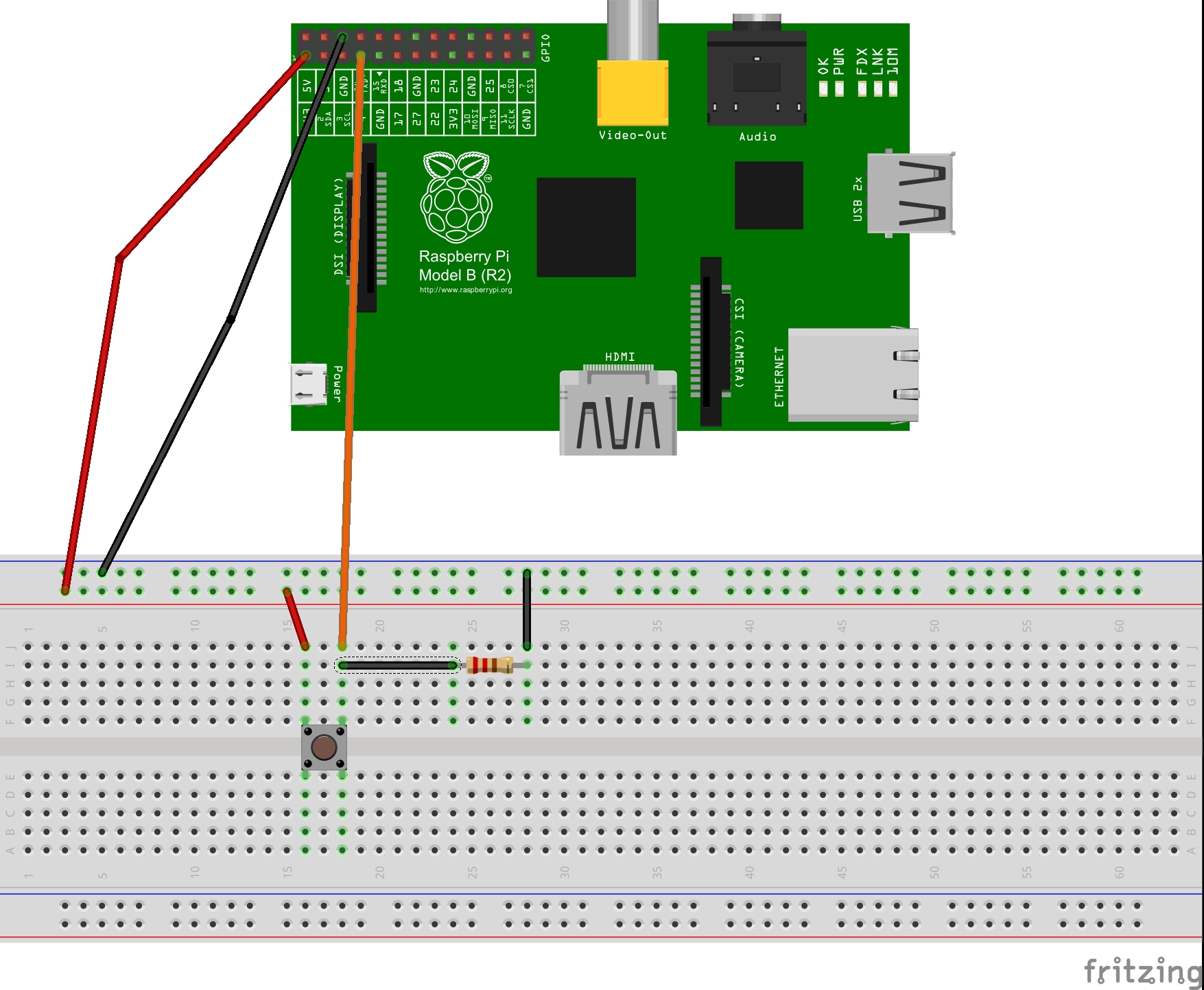
# **Automate turning off/on**

The present document, shows the code and the steps to follow to configure the automatic turning off and on of the Raspberry, by means of a push button.

**Connection**

You need:

* Push button
* Resistance 220/330 ohms
* The type of connection is going to be pull down

  
Illustration 1: Example of connection

NOTES:

* Check hardware documentation to see the requirements.
* Illustration 1, is only an example of the connection, the BionicKitchen project uses a perforated plate and the components are welded, instead than a protoboard like in the example.

**Shutdown script**

**1*.***  *#!/usr/bin/python2.7*

**2***. # -\*- coding: utf8 -\*-*

**3**.

**4**. import RPi.GPIO as GPIO

**5**. from subprocess import call

**6**.

**7**.

**8**. GPIO.setmode(GPIO.BCM)

**9**. GPIO.setup(3, GPIO.IN)

**10**. GPIO.wait\_for\_edge(3, GPIO.FALLING)

**11**.

**12**. call(['shutdown', '-h', 'now'], shell=False)

**NOTE:**

* Preferably name the code: **Shutdown.py**
* The code can be an any location of the Raspberry, preferably save in the BionicKitchen Resources folder.

**Manual configuration (****automation)**

1. **Import the Rpi.GPIO library (module) to Python.**

Introduce the next sentence in the OS terminal (shell), for installation of the library:

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

1. **Run the script** **for testing.**

Introduce the next sentence in the OS terminal (shell), for installation of the library:

*python2.7 script\_location/shutdown.py*

Press the button to check if it turns off, if so, turn on the Raspberry again by pressing the button. If all works, continue to the next step.

1. **Automate the script.**

Introduce the next sentence in the OS terminal (shell), for open a configuration file:

*crontab -e*

**3.1.** When the file is opened, add the following statement to the end of the file:

*@reboot sudo python script\_location/shutdown.py &*

**3.2.** Save and close the file:

ctrl + c------ **to save**

ctrl + x------ **to close**

1. **The Raspberry is turned off / on automatically when the button is pressed.**

# **Run GUI at start-up**

**Start program script:**

**1*.***  *#!/usr/bin/python2.7*

**2***. # -\*- coding: utf8 -\*-*

**3**.

**4**. from os import system

**5**.

**6**. system('python2.7 -B /home/pi/Desktop/BionicKitchen/Resources/gui.py')

**NOTE:**

* Preferably name the code: **StartProgram.py**
* The code can be an any location of the Raspberry, preferably save in the BionicKitchen Resources folder.

**Manual configuration (****automation)**

1. **Create script in /etc/init.d**

Introduce the next sentence in the OS terminal (shell), for open a configuration file:

*sudo nano /etc/init.d/automate\_startup*

**1.1.** When the file is opened, add the following statement to the end of the file:

*#! /bin/sh*

*# /etc/init.d/automate\_startup*

*### BEGIN INIT INFO*

*# Provides: automate\_startup*

*# Required-Start: $all*

*# Required-Stop: $remote\_fs $syslog*

*# Default-Start: 2 3 4 5*

*# Default-Stop: 0 1 6*

*# Short-Description: Script to start a program at boot*

*# Description: Script to start up the BionicKitchen main program (GUI) on automatically*

*### END INIT INFO*

*# Dependiendo de los parámetros que se le pasen al programa se usa una opción u otra*

**case** "$1" **in**

start**)**

**echo** "Arrancando GUI"

*# Aquí hay que poner el programa que quieras arrancar automáticamente*

**/**usr**/**bin**/**python **/**home**/**pi**/**Desktop/BionicKitchen/Resources/StartProgram.py

**;;**

stop**)**

**echo** "Deteniendo GUI"

**;;**

**\*)**

**echo** "Modo de uso: /etc/init.d/automate\_startup {start|stop}"

**exit** 1

**;;**

**esac**

**exit** 0

NOTES:

* # Required-Start: $ all - With this we indicate to the system that first load all the other modules.
* # Default-Start: 2 3 4 5 - Here we tell the system in which run levels we want our script to start up.
* Verify the location of the program on the Raspberry

1. **Make script executable**

Introduce the next sentence in the OS terminal (shell), for open a configuration file:

*sudo chmod 755 /etc/init.d/automate\_startup*

1. **Test starting the program**

Introduce the next sentence in the OS terminal (shell), for open a configuration file:

*sudo /etc/init.d/automate\_startup start*

1. **Test stopping the program**

Introduce the next sentence in the OS terminal (shell), for open a configuration file:

*sudo /etc/init.d/automate\_startup stop*

1. **Register script to be run at start-up**

Introduce the next sentence in the OS terminal (shell), for open a configuration file:

*sudo update-rc.d automate\_startup defaults*

# **Re-installation Raspbian on Raspberry Pi**

**Manual Re-installation**

1. Reboot the raspberry and after to show the image multicolor you need press shift and you need follow the instruction for reinstall the system.
2. Press shift when multicolor image is shown on the Raspberry boot.
3. Follow system instructions for reinstall OS