

## Raspbian OS Terminal

Command	Result
cat [name]	Show the contents of the file [name]
cd ..	Change to parent directory
cd [path]	Move to the directory at [path]
cd /	Change to root directory
cd ~	Change to your home directory - usually "/home/<username>"
chmod [who][+,-,=][permissions] [name]	Change the permissions for a file
chmod 777 [name]	Allow all users to read, write and execute the file [name]
chmod u+x [name]	Allow the user to execute [name]
cp -r [from] [to]	Copy all files and subdirectories from source [from] to destination [to]
cp [from] [to]	Copy a file from source [from] to destination [to]
find	Search for files and their contents
grep 'string' [name]	Search inside one or more files for occurrences of 'string'
head [name]	Return all occurrences of 'string' within file [name]
ls	List the contents of the current directory
ls -a	List all files including hidden files
ls -l	List the contents of the current directory with more file information
ls [path]	List the contents of the directory found at [path]
man [command]	Open the manual/help page for [command]
man man	Open the manual/help page for the 'man' command (helpception)
mkdir [name]	Create a directory called [name] in the current working directory
mv -r [from] [to]	Move all files and directories from source [from] to destination [to]
mv [from] [to]	Move a file from source [from] to destination [to]
pwd	Show the name of the current working directory
python/python3 --version	Shows you what version of Python you currently have installed
rm -r *	Remove all files and directories from the current working directory
rm [name]	Remove the specified file
rm *	Remove all files from the current working directory
rmdir [name]	Remove the empty directory [name] from the current working directory Superuser do. Execute [command] with elevated privileges (Allows you to do things you usually wouldn't have access to)
sudo [command]	
sudo apt-get install [package]	Install a package
sudo apt-get update	Update the list of packages
sudo apt-get upgrade	Upgrade the installed packages - must be run after sudo apt-get update
sudo chown pi:root [name]	Change the owner of the file [name] to user 'pi' and set the group to 'root'
sudo raspi-config	Launch the Raspberry Pi configuration menu
sudo reboot	Safely restart your Pi
sudo shutdown -h now	Safely shutdown your Pi immediately
sudo su	Places you in the root directory with root user access - be careful with this!
tail [name]	Show the end of file [name]
tar -cvzf [name] [path]	Create compressed file [name] from the contents of [path]
tar -xvzf [name]	Extract the contents of the compressed file [name]
wget [uri]	Download the file found at [uri] on the internet

## RPi.GPIO Library

Command	Result
import RPi.GPIO as GPIO	Import the RPi.GPIO module into the python sketch
GPIO.setmode(GPIO.BCM)	Use Broadcom pin numbers (GPIO 14, GPIO 15 etc)
GPIO.setmode(GPIO.BOARD)	Use board pin numbers (4,5, 8 etc)
GPIO.getmode()	Returns current pin numbering mode (BCM, BOARD, or None)
GPIO.setup([pin number], GPIO.IN)	Set up the pin at [pin number] to be an input
GPIO.setup([pin number], GPIO.IN, pull_up_down=GPIO.PUD_DOWN)	Set up the pin at [pin number] to be an input with internal pull down resistance
GPIO.setup([pin number], GPIO.IN, pull_up_down=GPIO.PUD_UP)	Set up the pin at [pin number] to be an input with internal pull up resistance
GPIO.setup([pin number], GPIO.OUT)	Set up the pin at [pin number] to be an output
GPIO.setup([pin number], GPIO.OUT, initial=1)	Set up the pin at [pin number] to be an output with the initial value '1'
GPIO.output([pin number], 1)	Set [pin number]'s value to 1. Note that 1, GPIO.HIGH and True are the same thing
GPIO.output([pin number], 0)	Set [pin number]'s value to 0. Note that 0, GPIO.LOW and False are the same thing

<code>i = GPIO.input([pin number])</code>	Set the variable <code>i</code> to the value of <code>[pin number]</code>
<code>if GPIO.input([pin number]):</code>	Use the value of <code>[pin number]</code> as a boolean in code
<code>GPIO.cleanup()</code>	Reset all GPIO pins (good practice to call before leaving any program)
<code>GPIO.VERSION</code>	Returns current RPi.GPIO version

## GPIO Zero Library

Command	Result
<b>LEDs</b>	
<code>from gpiozero import LED</code>	Import the LED section of the gpiozero library
<code>led = LED(17)</code>	Assign the 'led' variable to an LED on pin GPIO 17
<code>led.on()</code>	Turn on the LED stored in the 'led' variable
<code>led.off()</code>	Turn off the LED stored in the 'led' variable
<code>led.toggle()</code>	Toggle the LED stored in the 'led' variable (if it's off, turn it on and vice versa)
<b>Motors</b>	
<code>from gpiozero import Motor</code>	Import the Motor section of the gpiozero library
<code>motor = Motor(17, 18)</code>	Assign the variable 'motor' to a Motor object containing the forward and backward drive pin numbers
<code>motor.forward()</code>	Activate the forward pin of the variable 'motor'
<code>motor.backward()</code>	Activate the backward pin of the variable 'motor'
<code>motor.reverse()</code>	Reverse the current motor direction
<code>motor.stop()</code>	Stop the motor
<b>Buzzer</b>	
<code>from gpiozero import Buzzer</code>	Import the Buzzer section of the gpiozero library
<code>bz = Buzzer(3)</code>	Assign the variable <code>bz</code> to a Buzzer on pin GPIO3
<code>bz.on()</code>	Turn the buzzer on
<code>bz.off()</code>	Turn the buzzer off
<code>bz.toggle()</code>	Toggle the buzzer's state (if it's off, turn it on and vice versa)
<b>Servo</b>	
<code>from gpiozero import Servo</code>	Import the Servo section of the gpiozero library
<code>servo = Servo(17)</code>	Assign the 'servo' variable to a Servo on GPIO 17
<code>servo.min()</code>	Move the servo to its minimum value
<code>servo.mid()</code>	Move the servo to its middle value
<code>servo.max()</code>	Move the servo to its maximum value
<code>servo.value = 0.5</code>	Move the servo to a set numerical point (min = -1, max = 1)

## Raspi Camera Image

Command	Result
<code>raspistill</code>	Command to take a still image with attached camera, modify with arguments below
<code>--width, -w</code>	Set image width <size>
<code>--height, -h</code>	Set image height <size>
<code>--quality, -q</code>	Set JPEG quality <0 to 100> (75 is most common)
<code>--raw, -r</code>	Inserts raw Bayer data from the camera into the JPEG metadata
<code>--output, -o</code>	Output filename <filename> (required for saving)
<code>--latest, -l</code>	Add latest frame to filename <filename>
<code>--verbose, -v</code>	Verbose debugging information during run
<code>--timeout, -t</code>	Set a time to wait before capturing an image.
<code>--encoding, -e</code>	Encoding to use for output file - jpg, gif, bmp, or png

## Raspi Camera Video

Command	Result
<code>raspivid</code>	Command to take a video using attached camera, modify with arguments below
<code>--width, -w</code>	Set image width <size> (between 64px - 1920px)
<code>--height, -h</code>	Set image height <size> (between 64px - 1080px)
<code>--bitrate, -b</code>	Set bitrate in bits per second (i.e 15 Mbits/s = 15000000)
<code>--output, -o</code>	Output filename <filename> (required for saving)
<code>--verbose, -v</code>	Verbose debugging information during run
<code>--timeout, -t</code>	Set a time to wait before capturing video
<code>--framerate, -fps</code>	Specify the frames per second for recording