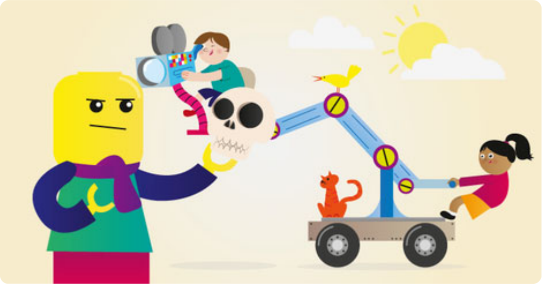
Projects

Push Button Stop Motion

Make your own stop motion animation rig with a push button using Python Picamera and GPIO



Step 1 What you will make

Make your own stop motion animation video using a Raspberry Pi, Python and a camera module to take pictures, controlled by a push button connected to the Pi's GPIO pins.

You can use LEGO to animate a tower being built, figures acting out a scene, or anything else you can think of

What you will learn

By creating a push button stop motion machine with your Raspberry Pi you will learn

- How to set up and use the Raspberry Pi camera module
- How to use the Python picamera library to capture photograph
- How to connect a button to the GPIO pins on a Raspberry Pi
- How to control the camera with a button using GPIO Zero
- How to generate a video from the command line using `avconv`

This resource covers elements from the following strands of the **Raspberry Pi Digital Making Curriculum** (https://www.raspberrypi.org/curriculum/):

- **Combine programming constructs to solve a problem** (https://www.raspberrypi.org/curriculum/programming/builder)
- **Combine inputs and/or outputs to create projects or solve a problem** (https://www.raspberrypi.org/curriculum/physical-computing/builder)

What you will need

Hardware

- Raspberry Pi camera module
- 1 x Full size breadboard
- 2 x Male-to-female jumper leads
- 1 x Tactile button

Software

You'll need to make sure you have the following packages installed.

- libav-tools

You'll need to be online to install packages.

First update and upgrade your system. Enter the following commands in to the termina

```
sudo apt-get update
sudo apt-get upgrade
```

Now install the packages you'll need:

```
sudo apt-get install libav-tools
```

- Test you have `libav-tools` installed by entering`avconv` at the command line. You should see some information about the `avconv` version.

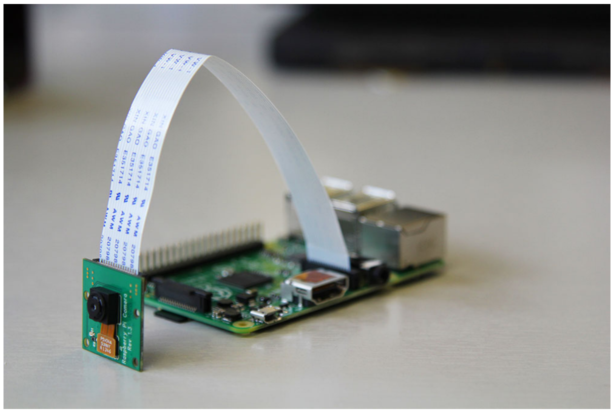
Additional information for educators

If you need to print this project, please use the**printer-friendly version** (https://projects.raspberrypi.org/en/projects/push-button-stop-motion/print).

You can find the finished code for this project**here** (http://rpf.io/p/en/push-button-stop-motion-get).

Step 2 Connect the camera

Before booting your Pi, you'll need to connect the camera



Locate the camera port next to the Ethernet port. Lift the tab on the to ☑

Place the strip in the connector, with the blue side facing the Ethernet port. While holding the strip in place, push down the t ☑

Turn the power on to boot the Pi. ☑

Step 3 Test the camera

Open a terminal window from the application menu. Enter the following comman ☑

```
raspistill -k
```

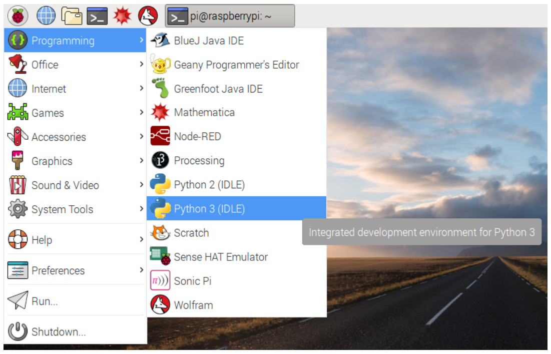
You should see a preview appear on the screen. It doesn't matter if the picture is upside-down; you can configure this later. Pres `Ctrl + C` to exit the preview.

Run the command `ls` to see the files in your home directory; you should see `image1.jpg` listed. ☑

Click the file manager icon in the taskbar and you should see some folders and files. Double-clic `image1.jpg` to preview it. ☑

Step 4 Take a picture with Python

Open **Python 3 (IDLE)** from the main menu:



Select `File > New Window` from the menu to open a Python file editor

Carefully enter the following code into the new window (case is important!

```
from picamera import PiCamera
from time import sleep

camera = PiCamera()

camera.start_preview()
sleep(3)
camera.capture('/home/pi/Desktop/image.jpg')
camera.stop_preview()
```

Select `File > Save` from the menu (or press`Ctrl + S`) and save as `animation.py`.

Press `F5` to run your program.

You should see `image.jpg` saved on your Desktop. Double-click the icon to open the image

If the picture is upside-down you can either reposition your camera using a mount, or leave it as it is and tell Python to flip th image. To do this, add the following lines: ☑

```
camera.rotation = 180
```

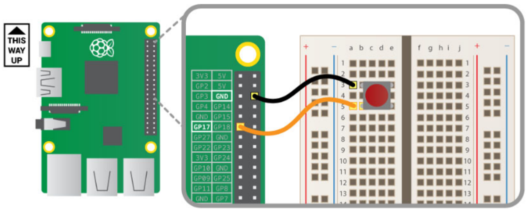
after `camera = PiCamera()`, so it becomes:

```
from picamera import PiCamera
from time import sleep

camera = PiCamera()

camera.rotation = 180
camera.start_preview()
sleep(3)
camera.capture('/home/pi/Desktop/image.jpg')
camera.stop_preview()
```

Run the file again and it will overwrite `image.jpg` with a new image in the correct orientation. Remember to keep these lines in your code while you alter it in the next few steps ☑

Step 5 Connect a hardware button

Using your breadboard and jumper leads, connect the Pi to the button as shown in the diagram belov ☑



Import `Button` from the `gpiozero` module at the top of the code, create up a `Button` connected to pin 17, and change the `sleep` line to use `button.wait_for_press` like so: ☑

```
from picamera import PiCamera
from time import sleep
from gpiozero import Button

button = Button(17)
camera = PiCamera()

camera.start_preview()
button.wait_for_press()
camera.capture('/home/pi/image.jpg')
camera.stop_preview()
```

Save and run your program. ☑

Once the preview has started, press the button connected to your Pi to capture an imag ☑

Return to the file manager window and you should see your `image.jpg`. Again, double-click to view. ☑

Step 6 Take a selfie

If you want to take a photograph of yourself with the camera board, you are going to have to add in a delay to enable you to get in position. You can do this by modifying your program

Add a line to your code to tell the program to sleep briefly before capturing an image, as below ☑

```
camera.start_preview()
button.wait_for_press()
sleep(3)
camera.capture('/home/pi/Desktop/image.jpg')
camera.stop_preview()
```

Save and run your program.

Press the button and try to take a selfie. Be sure to keep the camera still! Ideally, it should be mounted in position ☑

Again, feel free to check the image in the file manager. You can run the program again to take another selfie ☑

Step 7 Stop motion animation

Now that you have successfully taken individual photographs with your camera, it's time to try combining a series of still images to make stop motion animation.

IMPORTANT You must create a new folder to store your stills. In the terminal window, ente`mkdir animation`. ☑

Modify your code to add a loop to keep taking pictures every time the button is presse ☑

```
camera.start_preview()
frame = 1
while True:
    try:
        button.wait_for_press()
        camera.capture('/home/pi/animation/frame%03d.jpg' % frame)
        frame += 1
    except KeyboardInterrupt:
        camera.stop_preview()
        break
```

Because `while True` goes on forever, you have to be able to make it exit gracefully. Using`try` and `except` means it can deal with an exceptional circumstance - if you force it to stop wi`Ctrl + C` it will close the camera preview and exit the loo

`frame%03d` means the file will be saved as the name "frame" followed by a 3-digit number with leading zeroes - 001, 002, 00 etc. This allows them to be easily sorted into the correct order for the vide

Now set up your animation subject (e.g. LEGO), ready to start the stop motion animation ☑

Press the button to capture the first frame, then rearrange the animation subject and press the button again to capture ea subsequent frame. ☑

Once all the frames have been captured, press `Ctrl + C` to terminate the program. ☑

Open the `animation` folder in the file manager to see your stills collection ☑

Step 8 Generate the video

To generate the video, begin by returning to the terminal windov ☑

Run the video rendering command: ☑

```
avconv -r 10 -i animation/frame%03d.jpg -qscale 2 animation.h264
```

Note you're using `%03d` again - this is a common format which both Python an`avconv` understand, and means the photos will be passed in to the video in order.

Play your video using `omxplayer`. ☑

```
omxplayer animation.h264
```

You can adjust the frame rate by editing the rendering command. Try changin`-r 10` (10 frames per second) to another number.

You can also change the filename of the rendered video to stop it from overwriting your first attempt. To do this, chang `animation.h264` to something else.

Step 9 What next?

- Why not share your video? Try uploading it to YouTube!
- Now you know how to wire up a button to take a picture with the camera module, what else could you use this fc
- Could you do something similar for a time-lapse video?
- What could you use instead of a button? A motion sensor?
- Instead of making a video, what else could you do with photos taken with the camera module? You could post them to Twitter, another social media site.