



A PURE PYTHON INTERFACE FOR THE CAMERA MODULE: MEET PICAMERA!

62 Comments

If you have a Raspberry Pi camera module, you've probably used `rastill` and `raspivid`, which are command line tools for using the camera. [Dave Jones](#), a Database Admin, software developer and SQL know-it-all based in Manchester has been working on an equivalent, feature complete implementation of these in Python. This means you can access the camera module directly from a Python script, without using `os.system` or executing a subprocess.



At [Oggcamp](#) I set up a big TV with a Pi and camera taped on top. I SSH'd in to the Pi from my laptop and entered commands in to iPython using `picamera`.

Speaking as an avid Pythonist, `picamera`'s implementation is *beautiful* and it really is a wonderful library to use. It works really well for demonstrations using the Pi camera, and for real world applications. Part of the appeal of the Raspberry Pi is that you can work on embedded electronics projects

without needing to know low-level languages or have to program a microprocessor – instead you have the choice of a range of accessible high-level languages such as Python – and this is an extension of that kind of abstraction, which open up a world of possibilities to a wider diversity of makers.

Example usage:

```
import picamera
from time import sleep

camera = picamera.PiCamera()
camera.capture('image.jpg')

camera.start_preview()
camera.vflip = True
camera.hflip = True
camera.brightness = 60

camera.start_recording('video.h264')
sleep(5)
camera.stop_recording()
```

Also you can do things like this:

```
for i in range(100):
    camera.brightness = i
    sleep(0.1)
```

and watch the preview flow through the brightness levels.

The library has many configurations – you can change the brightness, contrast, saturation, image effects, exposure modes and such, as well as optionally show a live preview of the camera's view. You can capture single images and sequences of images as well as video streams.

Here's a presentation of picamera Dave gave at [Manchester Raspberry Jam XVI](#) – where he demonstrates the basic usage of the module by typing commands in to a Python prompt on his laptop, with a monitor displaying the camera output (unfortunately out of shot in the video):





Dave's wife Holly works in the Palaeontology department at Manchester University (Interesting fact: when Dave and Holly got married, they picked a new surname to take (Dave was previously a Hughes) – so Holly could become [Dr. Jones](#), and he would become [Davy Jones](#)) where they regularly capture images from microscopes. Rather than mount a huge camera on top of a microscope, Dave suggested attaching the Pi's camera module to the lens.



An early version of the crudely mounted Picroscope

He ended up writing a web app to support its use. This allows the user to control camera configuration settings through a web page, and capture photographs at the click of a button, as well as archive pictures taken in to a database along with metadata and extra information entered in a web form.



The web app showing pictures taken (smaller field of view due to temporary mounting setup)

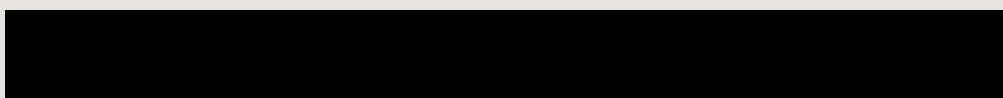


Single view of image taken through the web app, with metadata fields



Picroscopy configuration settings

Dave gave another presentation at the Manchester Jam, this time demonstrating a simple version of a Python web app for such an application (please excuse the numerous technical hiccups):





As Dave says in the video:

This is where having a library is particularly useful. If we were doing this with, say, raspistill and raspivid, when you want to alter the brightness of your preview, you'd have to shut down the program, regenerate the command line, restart the program. Here we're just saying "set a property". This is why a library is better for an interactive application like this. There's nothing wrong with raspistill and raspivid as far as they go, but they're not ideal for building this sort of application. If you have interactivity, you want a library.

See more videos from the Manchester Jam: [Jam #16 Videos](#).

`picamera` has been available in pypi since October ([v0.5](#)), so it's already in the wild – but now it's hit the v1.0 milestone Dave considered feature complete, it's packaged in the Raspbian archives so you install with `apt-get` (remember to run `apt-get update` first):

```
apt-get install python-picamera
```

or

```
apt-get install python3-picamera
```

depending on your taste.

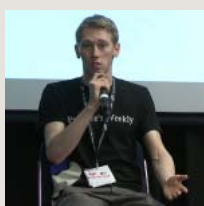
Dave also mentioned to me that this is by far the most popular project he's ever published – and he's been impressed by how brilliant the feedback has been from the Pi community. He's had a great response on the forums,

detailed bug reports provided by users and plenty of help from [James Hughes](#), the author of raspistill, and from [Alex Bradbury](#) in getting it packaged for Raspbian.

See the source for [picamera](#) and the [picroscopy web app](#) on github. Thanks to Dave for his hard work in building this library. Enjoy!

[python](#) [picamera](#) [dave jones](#) [camera module](#)

62 COMMENTS +



Posted by [Ben Nuttall](#)
*Development & Outreach at
the Raspberry Pi Foundation*
Likes Adventure Time
13th Jan 2014 at 11:04 am

[A PURE PYTHON INTERFACE FOR THE
CAMERA MODULE: MEET PICAMERA!](#)

[« BUY AN 8GB NOOBS SD CARD FOR £5](#)

[BLOG FEED](#)

[TEST TIM'S NEW SCRATCH ALPHA »](#)

[RSS FEED](#)

[VIEW THE ARCHIVE](#)



TAKE A BYTE!

- [Products](#)
- [Buy a Pi](#)
- [Buy Swag](#)
- [Buy Codecs](#)
- [Raspberry Jam](#)
- [The MagPi](#)
- [Documentation](#)
- [Troubleshooting](#)

ABOUT US

[About us](#)

[What is a Raspberry Pi?](#)

[FAQs](#)

[Cookies](#)

[Picademy](#)

[Creative Commons](#)

[Trademark rules](#)

[Contact us](#)

RASPBERRY PI FOUNDATION

UK REGISTERED CHARITY 1129409