

Lesson #4 – Setting up the Camera

In this lesson, we're going to make the camera work.

If you look around online, you're going to find a lot of websites referring to a set of programs and commands like `raspicam` and `raspistill`. Those have been “deprecated” (retired), and things are a bit up in the air, because the RPi foundation seems to be going through a transition in how it wants to do things (ultimately, to enable better and more powerful cameras, which is a good thing). But what that means for us is that things are a bit complicated, and require a few additional steps just to get things working the way they should.

Luckily for you, I have gone through the trouble of trying to make sense of these changes and to figure out how to make older hardware work with newer software. It turns out that it's pretty simple and well documented, even if it's a bit hard to find.

These instructions apply to the current (as of this writing) RaspberryOS (the version is called Bookworm), and should only be necessary for older official cameras (the module 1 and module 2), and third party cameras, but I've included the information for all official cameras, in case problems arise. (This info can be found at https://www.raspberrypi.com/documentation/computers/camera_software.html).

If you don't know what version of RaspberryOS you are using, simply type `cat /etc/os-release` in a terminal.

The problem is that the software tries to recognize cameras that have been installed, but doesn't always recognize older ones. In Bookworm, it's no longer necessary to enable the camera using `raspi-config`, but if you're using Bullseye, Stretch, or an older version, you may need to open up `raspi-config` from the command line, or the Raspberry Pi Configuration program (under Preferences) to enable the camera.

Having gotten all that out of the way, the simple solution is to open up the boot configuration file in a terminal and add a few lines. Which lines you enter will depend on the version of camera you're using. (See the table on the second page if you're using something other than the Camera Module 2.0.)

-
0. Connect the camera to your Raspberry Pi using the white camera cable. Be careful with the connection. Using the cable connector closest to the USB ports, gently lift up the sides of the black portion of the port using your fingernails. If it is tight, it might help to lift one side, then lift the other. Once it's up a couple of millimeters, insert the cable so that the silver pads on the cable are facing the white portion of the connector on the board. (The blue side of the cable will be facing the black part of the connector port.) Once the cable is fully seated, gently push the black portion of the connector back down.
 1. Connect to your Raspberry Pi either through `ssh` or through VNC software. For the first step, either way is fine. But once the camera is working, we'll need VNC to test it.
 2. Once you're connected, open a terminal and use `nano` to open `/boot/firmware/config.txt`. (For previous versions, the correct file may be `/boot/congif.txt`.)
 3. Find the line that reads “# Automatically load overlays for detected cameras”. You will see a line underneath that reads `camera_auto_detect=1`.
 4. Change the value of that variable to 0.
 5. Finally, add a second line underneath the “camera_auto_detect” line that reads:
`dtoverlay=[overlay version]`

6. where [overlay version] is the value listed in the table below. So, for example, if you're using a v2 camera module, the line would read:

```
dtoverlay=imx219
```

7. After modifying the config.txt file, you need to reboot your system (`sudo reboot`).
8. Once you're back and connected through VNC, type in a terminal:

```
libcamera-hello
```

9. After that runs, try:

```
libcamera-vid
```

This should run a little program and give you a test window to verify that your camera is indeed working.

That's all there is to it.

Sample Code for Using the Camera

Now, what do we do with it? How about a time-lapse camera?

```
# import some libraries
import time
from picamera2 import Picamera2, Preview

# name some variables that refer to elements from the Picamera2 library
picamera = Picamera2()

config = picamera.create_preview_configuration()

# initialize the camera
picamera.configure(config)
picamera.start()

# start a loop
while True:
    DATE = time.strftime("%Y-%m-%d")
    TIME = time.strftime("%H-%M-%S")
    picamera.resolution = (1024, 768)
    picamera.capture_file('/home/pi/timelapse/img' + DATE + TIME + '.jpg')
    print("Picture taken")
    time.sleep(2)
```

How about making it work with the motion sensor?

For additional camera types/versions, using the following info:

Camera Module	In /boot/firmware/config.txt
V1 camera (OV5647)	dtoverlay=ov5647
V2 camera (IMX219)	dtoverlay=imx219
HQ camera (IMX477)	dtoverlay=imx477
GS camera (IMX296)	dtoverlay=imx296
Camera Module 3 (IMX708)	dtoverlay=imx708
IMX290 and IMX327	dtoverlay=imx290,clock-frequency=74250000 or dtoverlay=imx290,clock-frequency=37125000 (both modules share the imx290 kernel driver; please refer to instructions from the module vendor for the correct frequency)
IMX378	dtoverlay=imx378
OV9281	dtoverlay=ov9281
