

# Camera module breakout board

---

The aim is to create a batch of 10 breakout PCBs for testing and connecting the camera module via a Raspberry Pi connector.

The board has a 34 pin socket for connecting a 201 camera module. This is connected and voltage shifted where needed to a Raspberry Pi Zero 22 pin connector and a RPi 15 pin connector for connecting to a RPi compatible dev board.

Additionally there should be a 30 pin IPEX connector for connecting to the UCM Compulab board.

Regular 1/100" header pins are broken out for I2C, I2S and other signal pins. It should be possible to connect with 1v8 and 3v3 signals. Add header pins for power as well.

Note: Perhaps it makes no sense to have a 22 pin connector since the Raspberry Pi standard is 15 pins on the camera module side. Consider dropping this connector.

## Verification of modules

The aim is to verify and validate the viability of the supplied camera modules. You will get two day time and two night time modules and assorted lenses. I will also send you a dev board.

- Test that the camera registers can be accessed with I2S
- Test that the laser range and ambient light can be accessed with I2S
- Test that camera sensor can be read using CSI2
- Test that microphone can be read using I2S

## Testing lenses

The supplied lenses are of various types, the aim is to evaluate the various lenses on the dimensions of,

- Image quality
- How close an object can be to the camera and remain mostly in focus
- How far away can a 5cm x 5cm toy be in a picture and remain recognizable
- How far off center can an object or person be and be in view

At the completion of the project you will return the supplied development board, camera modules, lenses and produced test boards.

## Testing Setups

The test board is used to test Camera Modules. Development boards that may be used to test are,

- [I-Pi SMARC IMX8M Plus](#)
- [Compulab SB-UCM-iMX8PLUS](#)
- [Raspberry Pi CM 4 IO Board B](#)
- [Tinker Edge R](#) for testing.

## Price / Milestones

Price \$1500 plus cost of board production

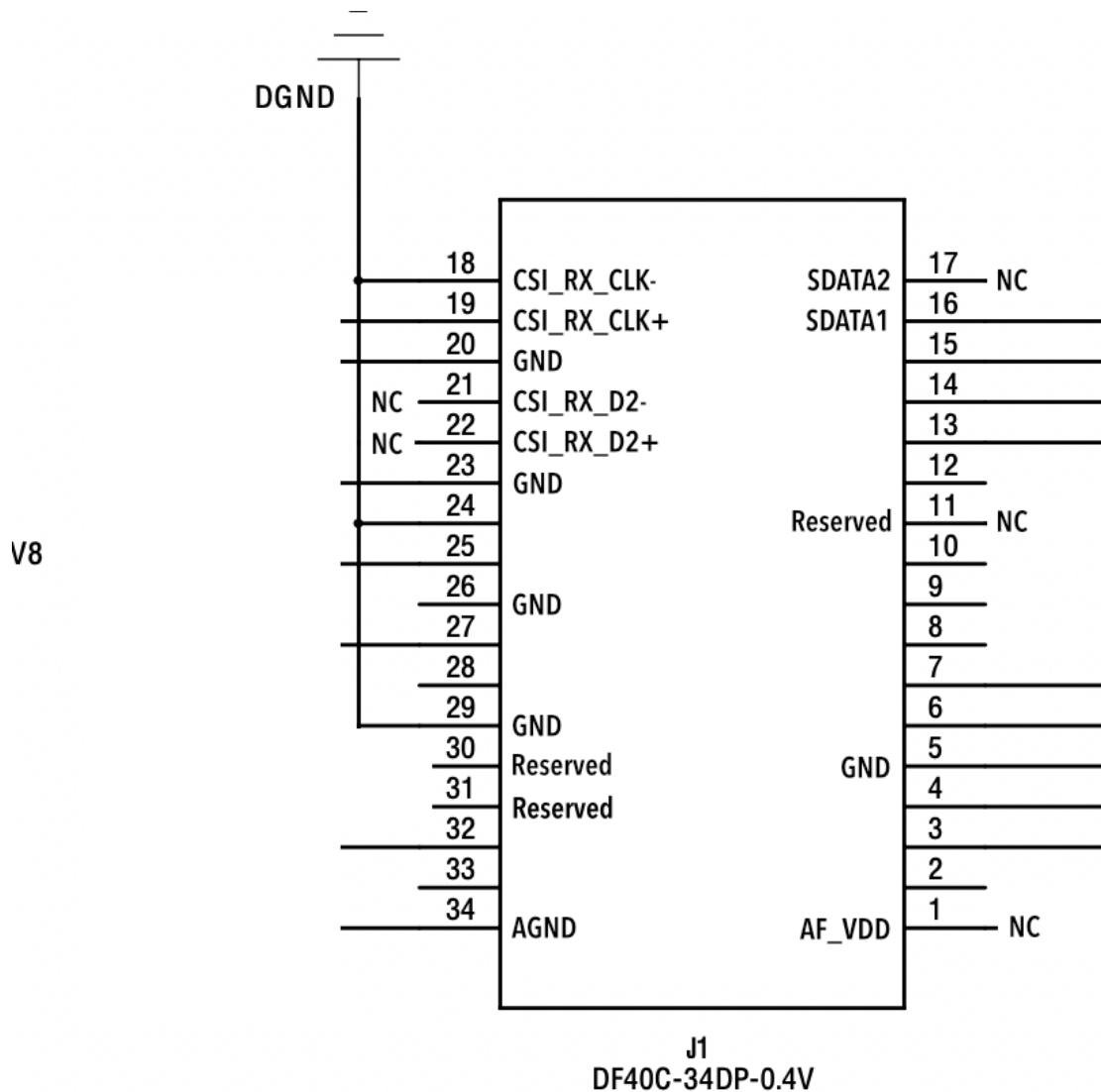
Milestones 300 Testing board design completed with design files 150 + cost, produce test boards 600  
verification of camera modules 150 testing lenses 300 boards, modules, lenses are returned

## Connectors

- 1 \* RPi compatible [TE Connectivity 15pin 1mm FPC 1-84952-5](#)
- 1 \* [Molex 22PIN 0.5mm pitch 54548-2271](#)
- 1 \* [I-PEX 30PIN 0.4mm pitch 20525-030E-02](#)
- 1 \* [Hirose DF40C-34DS-0.4V](#) (Mouser)

### Pinouts: Hirose DF40 single eye connector 34 pins

**Just to be clear:** All CSI lanes are laid out on one side of the connector with GND between.



Pin 1 is indicated on the board by a dot.

Toward thin part with microphone and other sensors

Pin	Code	Type	Details	Voltage
1	AF_VDD	Power	Reserved for Autofocus	3.3V
2	AVDD_2V8	Power	Analog, Max 500mA	2.8V
3	DOVDD	Power	Power for I/O circuit, Max 500mA	1.8V
4	VCC_1V8	Power	1.8V ,MAX 200mA	1.8V
5	GND	Power	GND	
6	CAM_FSIN	I/O	Frame sync input	
7	CAM_STROBE	I/O	Frame sync output	
8	EXTCLK	Input	External Clock Input (MCLK)	
9	ATT_INT	Output	Interrupt Attached Sensor, Active L	1.8V?
10	ATT_XSHUT	Input	Attached Sensor XSHUTDOWN	1.8V
11	Reserved	AF/PWM	PWM Motor control (NC)	
12	I2C_SCL	I/O	I2C?_SCL(pullup resistor 2.2K)	1.8V
13	I2C_SDA	I/O	I2C?_SDA(pullup resistor 2.2K)	1.8V
14	BCLK / SCK	I2S	Bit clock line	1.8V
15	WS / LRCLK	I2S	Word clock line	1.8V
16	SDATA1	I2S	Input data 1	1.8V
17	SDATA2	I2S	Input data 2 (NC)	1.8V

Towards image sensors

Pin	Code	Type	Details	Voltage
34	AGND	Power	Analog ground	
33	RESET	Input	Camera Reset, Active Low (RSTB)	
32	PWRDN	Input	Camera Power Down	
31	Reserved			
30	Reserved			
29	-		GND	
28	CSI_RX_D0P	Camera	MIPI_CSI_RX_D0+	1.8V
27	CSI_RX_D0N	Camera	MIPI_CSI_RX_D0-	1.8V
26	-		GND	

Pin	Code	Type	Details	Voltage
25	CSI_RX_D1P	Camera	MIPI_CSI_RX_D1+	1.8V
24	CSI_RX_D1N	Camera	MIPI_CSI_RX_D1-	1.8V
23	-		GND	
22	CSI_RX_D2P	Camera	MIPI_CSI_RX_D2+	1.8V
21	CSI_RX_D2N	Camera	MIPI_CSI_RX_D2-	1.8V
20	-		GND	
19	CSI_RX_CLKP	Camera	MIPI_CSI_RX_CLK+	1.8V
18	CSI_RX_CLKN	Camera	MIPI_CSI_RX_CLK-	1.8V

## NVIDIA FPC 30 pins

The connector is an [I-PEX type 20525-030E-02](#) with 0.4mm pitch & 30 pins. Data pins are 1.8V level.

Pin	Code	Details
1	CAM_3V3	3.3V Power Input
2	CAM_3V3	
3	CAM_1V8	1.8V Power Input
4	GND	
5	GND	
6	PWR DWN	PWRDN on 34pin
7	I2C SCL	
8	I2C SDA	
9	GND	
10	CSI D2-	
11	CSI D2+	
12	TRIGGER	
13	MCLK	EXTCLK on 34pin
14	Reserved	
15	CSI D1-	
16	CSI D1+	
17	GND	
18	GND	

Pin	Code	Details
19	CSI D0-	
20	CSI D0+	
21	RESET	RESET on 34pin
22	GND	
23	Reserved	
24	CSI CLK-	
25	CSI CLK+	
26	GND	
27	CSI D3-	
28	CSI D3+	
29	Flash	
30	Reserved	

#### Refs

- <https://www.leopardimaging.com/product/accessories/cables/faw-1233-03/>
- [https://www.mouser.com/datasheet/2/233/LI-TX1-CB-6CAM\\_datasheet-1395894.pdf](https://www.mouser.com/datasheet/2/233/LI-TX1-CB-6CAM_datasheet-1395894.pdf)
- [https://connecttech.com/ftp/pdf/ASG006\\_Spacely.pdf](https://connecttech.com/ftp/pdf/ASG006_Spacely.pdf)
- <https://www.i-pex.com/product/cabline-ca>

## RPI FPC 22 pins

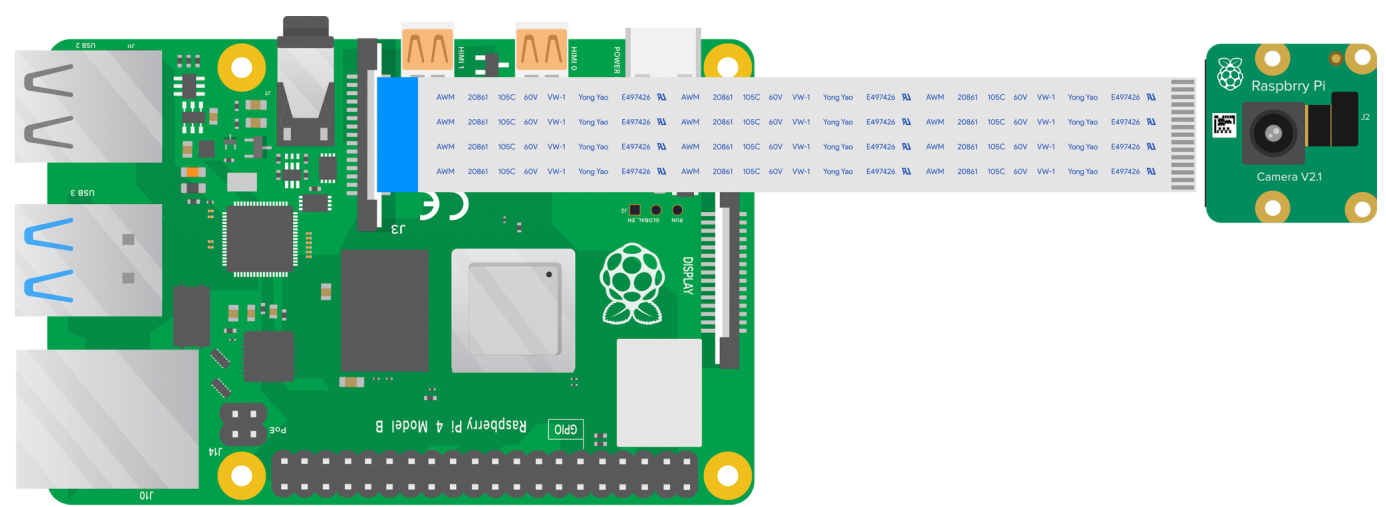
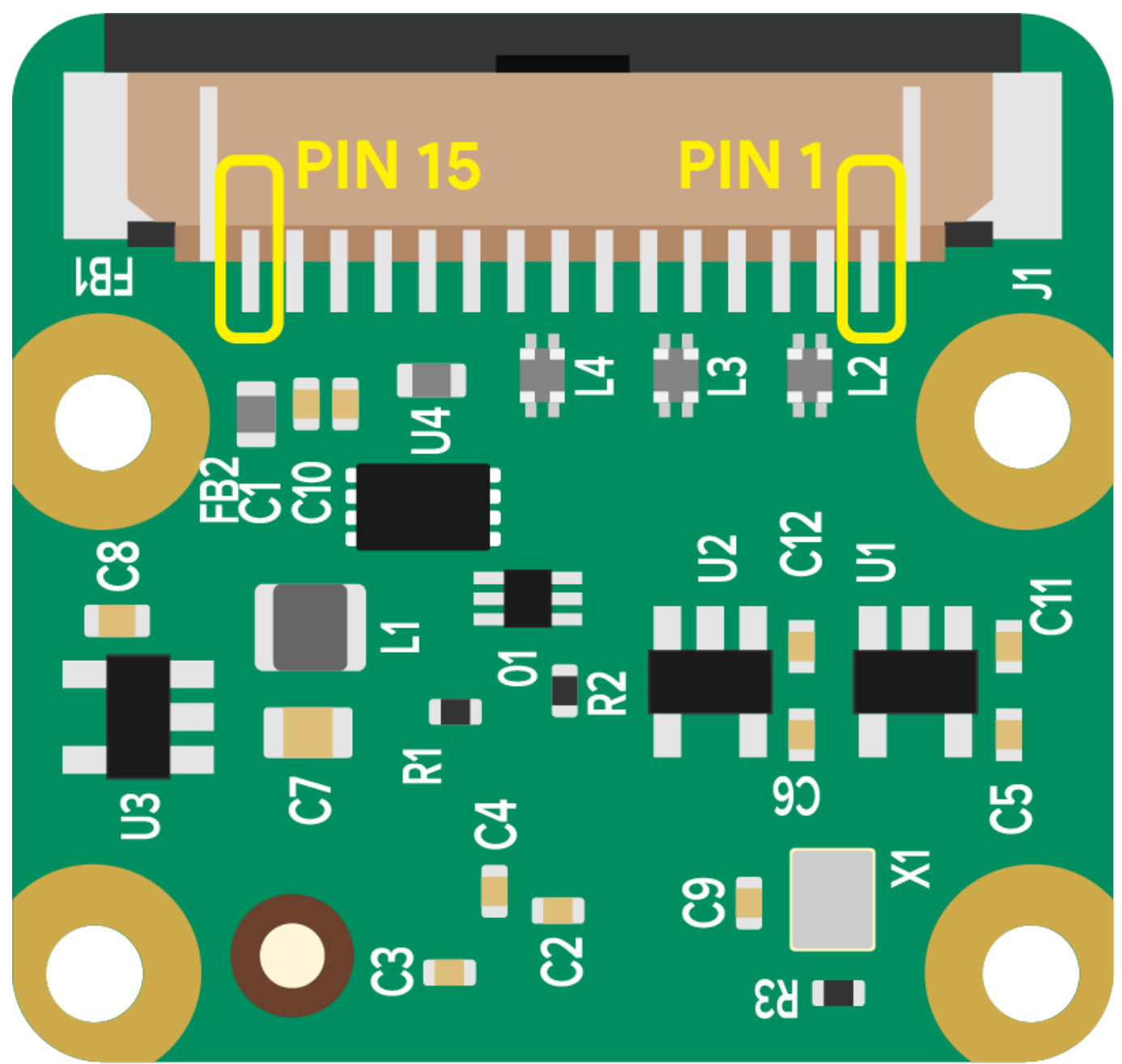
Pin	Code	Type	Details	Voltage
1	GND	Power	Ground	
2	CAM_D0_N	Data	MIPI Data Lane 0 Negative	
3	CAM_D0_P	Data	MIPI Data Lane 0 Positive	
4	GND	Power	Ground	
5	CAM_D1_N	Data	MIPI Data Lane 1 Negative	
6	CAM_D1_P	Data	MIPI Data Lane 1 Positive	
7	GND	Power	Ground	
8	CAM_CK_N	Data	MIPI Clock Lane Negative	
9	CAM_CK_P	Data	MIPI Clock Lane Positive	
10	GND	Power	Ground	
11	CAM_D2_N	Data	MIPI Data Lane 2 Negative	

Pin	Code	Type	Details	Voltage
12	CAM_D2_P	Data	MIPI Data Lane 2 Positive	
13	GND	Power	Ground	
14	CAM_D3_N	Data	MIPI Data Lane 3 Negative	
15	CAM_D3_P	Data	MIPI Data Lane 3 Positive	
16	GND	Power	Ground	
17	CAM_IO0	Power	Power Enable	
18	CAM_IO1	LED	LED Indicator	
19	GND	Power	Ground	
20	SCL	I2C	I2C SCL	
21	SDA	I2C	SCCB serial Interface data IO	
22	VCC	Power	3.3V Power Supply	

## RPI FPC 15 pins

Pin	Code	Type	Details	Voltage
1	GND	Power	Ground	
2	CAM_D0_N	Data	MIPI Data Lane 0 Negative	
3	CAM_D0_P	Data	MIPI Data Lane 0 Positive	
4	GND	Power	Ground	
5	CAM_D1_N	Data	MIPI Data Lane 1 Negative	
6	CAM_D1_P	Data	MIPI Data Lane 1 Positive	
7	GND	Power	Ground	
8	CAM_CK_N	Data	MIPI Clock Lane Negative	
9	CAM_CK_P	Data	MIPI Clock Lane Positive	
10	GND	Power	Ground	
11	CAM_IO0	Power	Power Enable	
12	CAM_IO1	LED	LED Indicator	
13	SCL	I2C	I2C SCL	
14	SDA	I2C	SCCB serial Interface data IO	
15	CAM_3V3	Power	3.3V Power Supply	

[ArduCam description for connector](#). ArduCam uses FCI connector [SFW15R-2STE1LF](#).



Connecting Raspberry Pi to camera module

Using a supported camera module together with a Raspberry Pi is the most straightforward way of adding visual input to a project. The only thing needed is a supported camera module such as the official Raspberry Pi camera module. However, many alternative modules exist: for example, this breakout board with a night-vision camera and infrared LEDs.

Regardless of the chosen camera, the setup process always works similarly, as long as the camera is supported by the Raspberry Pi. First, plug the ribbon connector of the camera module into the connector on the Raspberry Pi. The white connector closer to the USB and Ethernet ports is the one for the camera. The other port, located on the other side of the single-board computer, is meant for connecting a display.

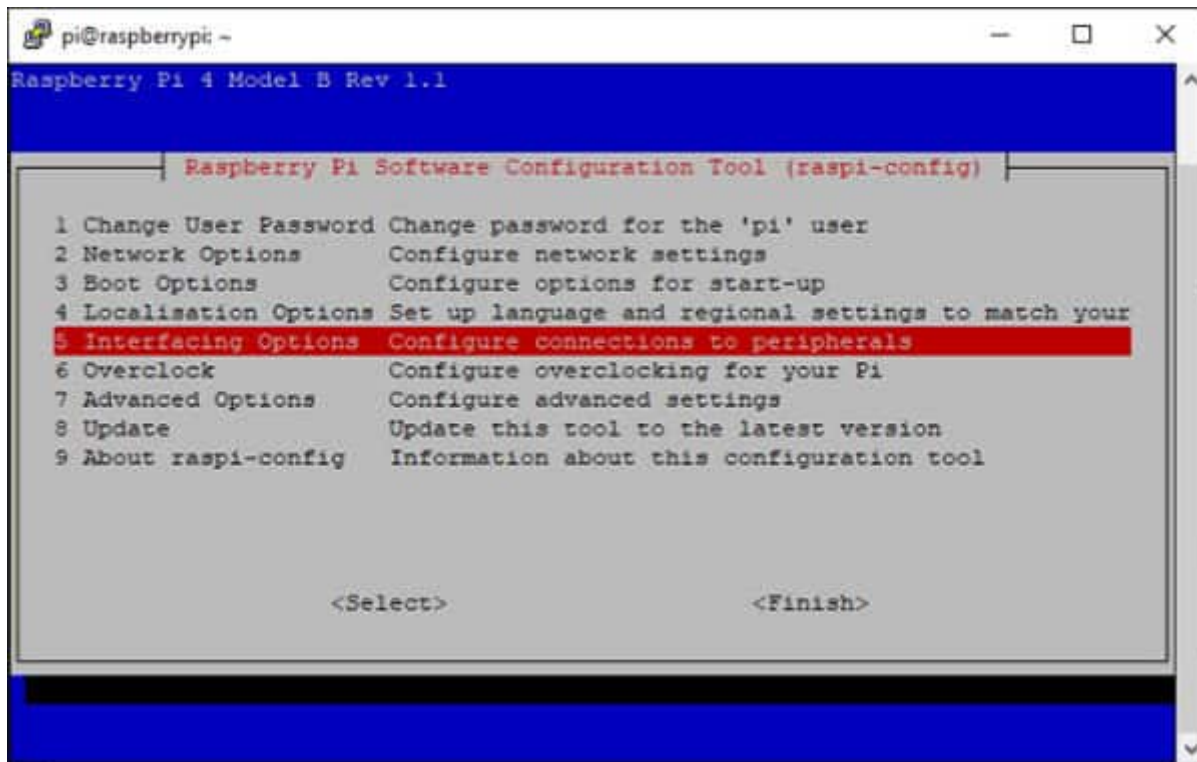


Connect the camera module while the Raspberry Pi is still powered off. Note that the shiny contacts on the ribbon cable should face away from the USB ports. Then, turn the Raspberry Pi on and launch the Raspberry Pi Software Configuration tool by typing the following command:

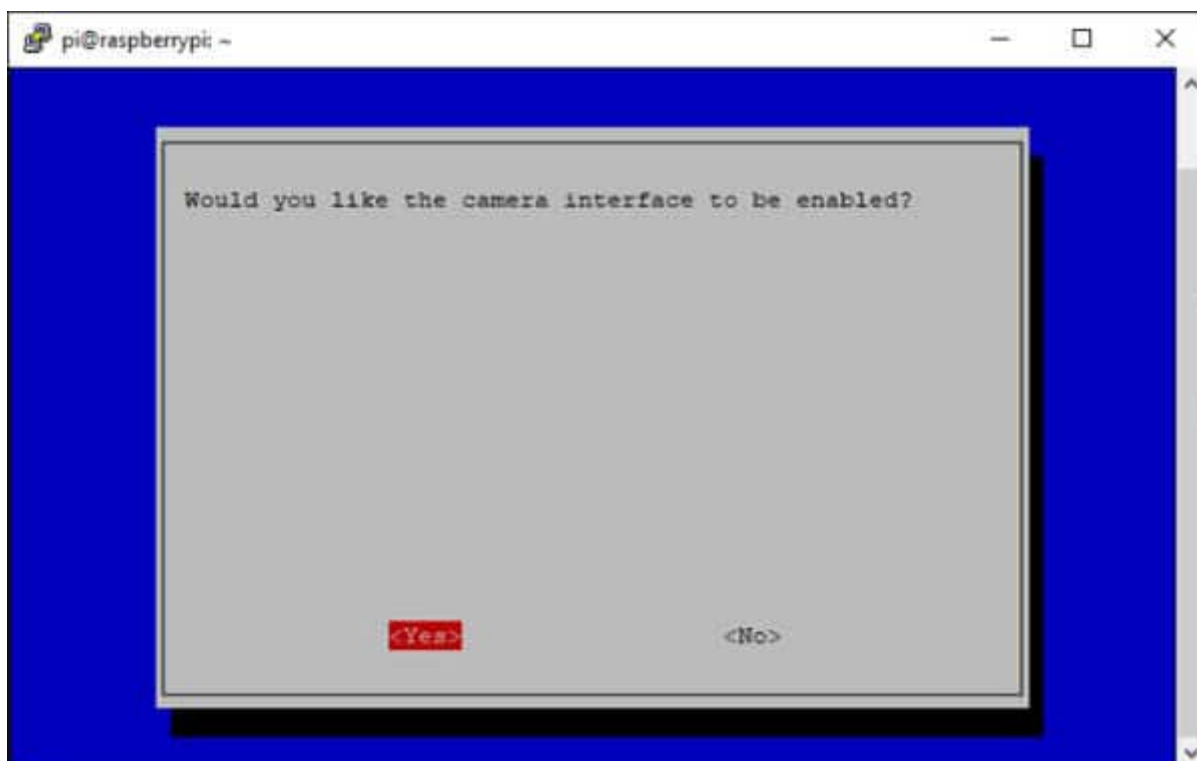
```
sudo raspi-config
```

In the graphical user interface, choose the fifth option in the list:





Then, select the 'Camera' option from the list and enable it:



When done, use the tab key in the main menu to select the 'Finish' option and then hit the enter key. After a reboot, the Raspberry Pi will be ready to interface with a connected camera module.

Once the computer is finished rebooting, you can check the status of the camera using the following command:

```
vcgencmd get_camera
```

If the camera is connected correctly and the Raspberry Pi is configured to use the camera interface, the `vcgencmd` command should create an output similar to the following image:

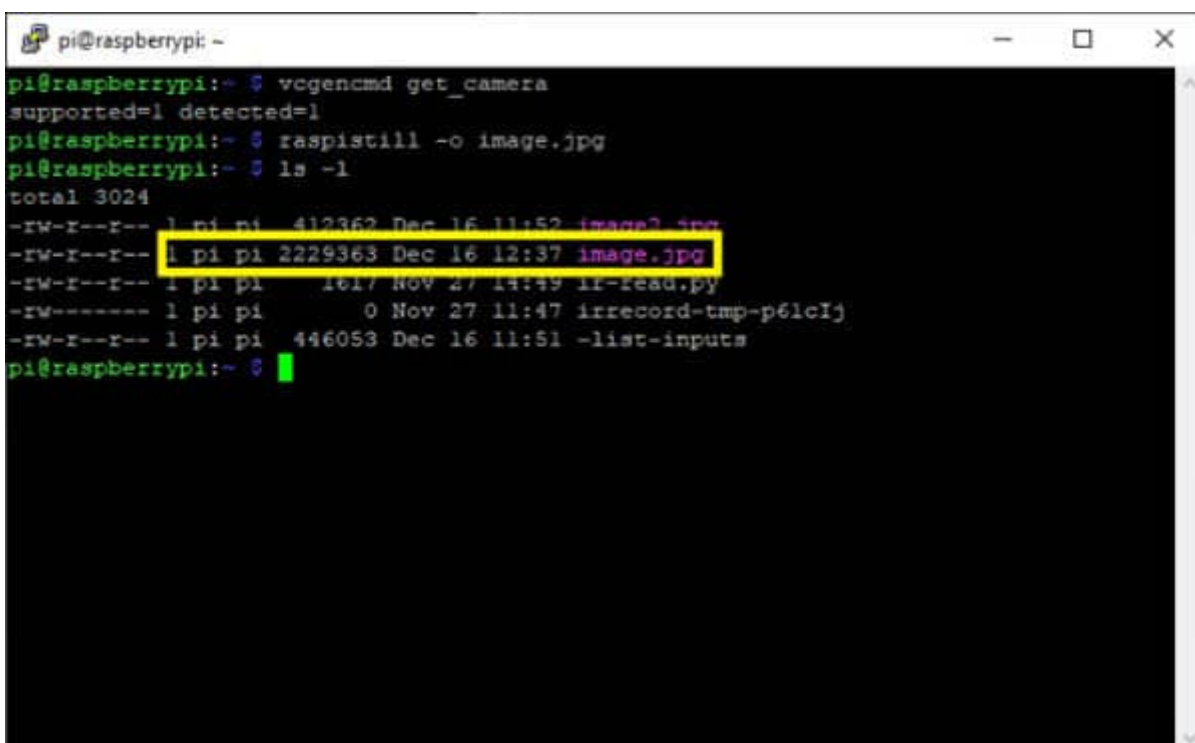


```
pi@raspberrypi:~$ vcgencmd get_camera
supported=1 detected=1
pi@raspberrypi:~$
```

You can use the `raspistill` command to test whether the camera module works as intended:

```
raspistill -o image.jpg
```

Note that running this command might take a couple of seconds before it's done. It should create an image file like this:



```
pi@raspberrypi:~$ vcgencmd get_camera
supported=1 detected=1
pi@raspberrypi:~$ raspistill -o image.jpg
pi@raspberrypi:~$ ls -l
total 3024
-rw-r--r-- 1 pi pi 412362 Dec 16 11:52 image2.jpg
-rw-r--r-- 1 pi pi 2229363 Dec 16 12:37 image.jpg
-rw-r--r-- 1 pi pi 1617 Nov 27 14:49 ll-read.py
-rw----- 1 pi pi 0 Nov 27 11:47 irrecord-tmp-p6lcIj
-rw-r--r-- 1 pi pi 446053 Dec 16 11:51 -list-inputs
pi@raspberrypi:~$
```

## Utilize a common USB webcam

If buying a new camera module is not feasible for your project, or if a higher resolution, longer cable, or other features are required, it's possible to connect common USB cameras to the Raspberry Pi and use them instead of a camera module. For that to work, shut down the Raspberry Pi and connect the USB webcam. Then turn the Raspberry Pi back on. The Pi should detect most USB cameras and create a new device.

To test whether the camera works with the Raspberry Pi as intended, you can install a small program named 'fswebcam' and create a still image as before:

```
sudo apt-get install fswebcam  
fswebcam image.jpg
```

Doing so creates a still image in the current folder. Note that it doesn't pose a problem to leave the Raspberry Pi's camera interface enabled when a webcam gets connected via USB. It is also possible to connect multiple USB cameras to the Raspberry Pi and build a home surveillance hub, for example.

Use the following command to take a picture with a specific webcam:

```
fswebcam --device /dev/video0 image.jpg
```

This will create a still image with the camera zero and store the result in a file named 'image.jpg'.

## Summary

As demonstrated, configuring a Raspberry Pi to interface a connected camera is a straightforward process. Essentially, two common methods exist, and both of them work without much configuration. The first option is to use a Raspberry Pi-compatible camera module. The benefits of this method are the great support, the large variety of different modules, and that most camera-based software for the Raspberry Pi supports the internal camera interface right out of the box. The relatively short cable length of such camera modules can be a disadvantage. USB cameras, on the other hand, usually have a longer cable and reasonable image quality. Support, however, cannot be guaranteed. On a positive note, using this method allows you to connect multiple USB cameras to a single Raspberry Pi, which is especially useful in surveillance applications.