CAM-MIPIOV9281 V2 UserManual

Menus

1. General Description

The CAM-MIPIOV9281 for Raspberry Pi is a high-speed global shutter camera designed to deliver ultra-fast, distortion-free imaging. Built around the OV9281 sensor from OmniVision, it provides a powerful combination of performance, flexibility, and reliability, making it an excellent choice for robotics, machine vision, AR/VR, scientific research, and AI-driven applications.

With full compatibility for Raspberry Pi OS and support through the native driver and tools (libcamera/rpicam), this module is easy to integrate into embedded vision projects. Its global shutter design eliminates rolling artifacts and motion blur, ensuring crisp image capture even in high-speed scenarios. It supports multiple resolutions (1280×800, 1280×700, 640×400) and delivers up to 309 frames per second, enabling real-time responsiveness and high-speed data analysis.

In addition, the module supports RAW8/RAW10 uncompressed output, providing maximum flexibility for image processing and computer vision algorithms. Thanks to its high sensitivity and low-noise performance, it also works reliably in low-light or near-infrared (NIR) environments.

By combining speed, accuracy, and ease of use, the OV9281 Camera Module empowers developers, researchers, and innovators to build advanced vision solutions, from real-time robotics and industrial inspection to immersive AR/VR experiences and scientific motion analysis.

2. Features

(1) CAM-MIPIOV9281 is an Industrial Camera Module for whole series Raspberry Pi.Support libcamera/rpicam tools.

(2) On-board OmniVision OV9281 Monochrome(Black&White) CMOS Sensor, 1M Pixel.With global shutter design eliminates motion distortion, making it especially suitable for high-speed imaging scenario.

(3) The built-in OV9281 driver on the Raspberry Pi Os supports RAW8 and RAW10 output formats, with resolutions of 1280x800, 1280x700, and 640x400. The maximum frame rate can reach up to 309 fps.

(4) Support for external trigger, LED and flash strobe mode and gain programmable. Interfaces with optical isolation.

(5) Comes a wide angle Lens. Fov(D)=148 degrees, Fov(H)=118 degrees. Focal distance is adjustable.

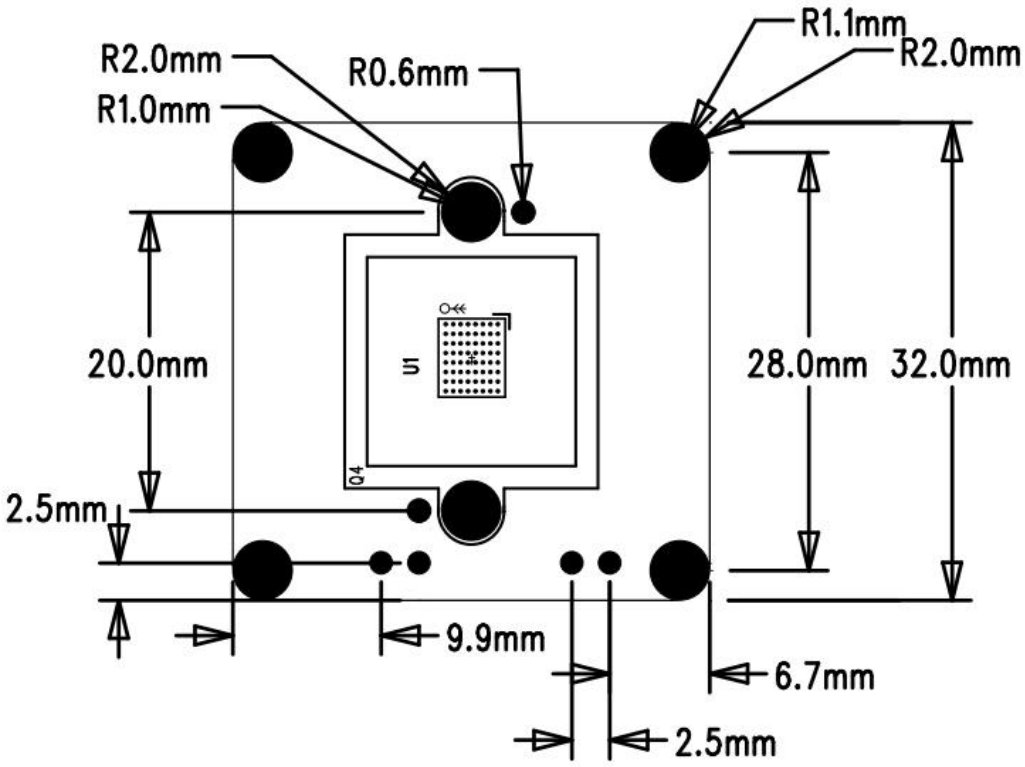
3. Hardware Description

3.1 Overview

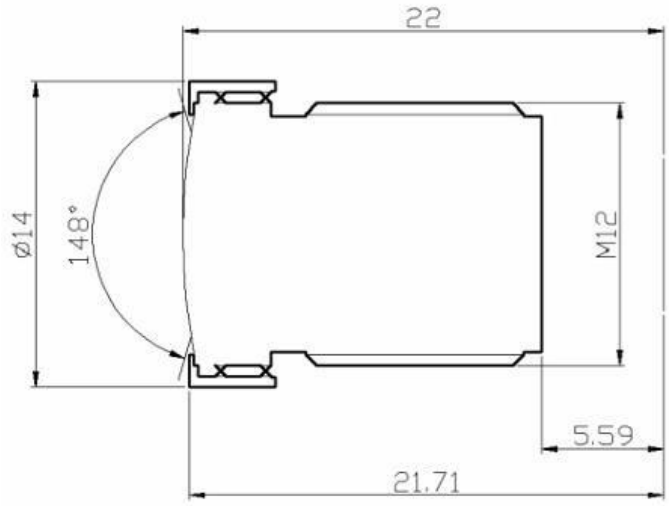
Sernor Board	
Size	32mm x 32mm
Weight	4g
Still Resolution	1 million pixels
Linux integration	Build-In driver on Raspberry Pi Os Support rpicam and libcamera
Sensor	Monochrome global shutter OV9281
Sensor Resolution	1280*800 pixels
Sensor image area	3896μm x 2453μm
Pixel size	3 μm x 3 μm
Optical size	1/4"
S/N ratio	38 dB
Dynamic range	68 dB
Output interface	2-lane MIPI Interface
Output formats	8/10-bit B&W RAW
Field of view	Fov(D) = 148 degrees , Fov(H) = 118 degrees
Focal Length	2.8 mm
Focal Distance	Adjustable
TV DISTORTION	<-17%
F(N) /Aperture	2.2

3.2 Size

3.2.1 PCB Size



3.2.2 Len Size



3.3 Connection Of The Hardware

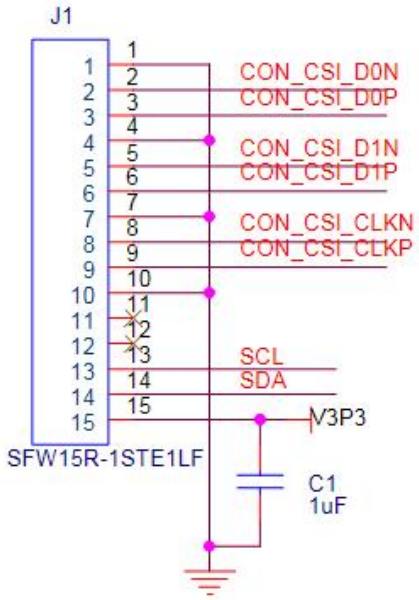
Blow link show you how to connect the cameras to the Raspberry Pi:

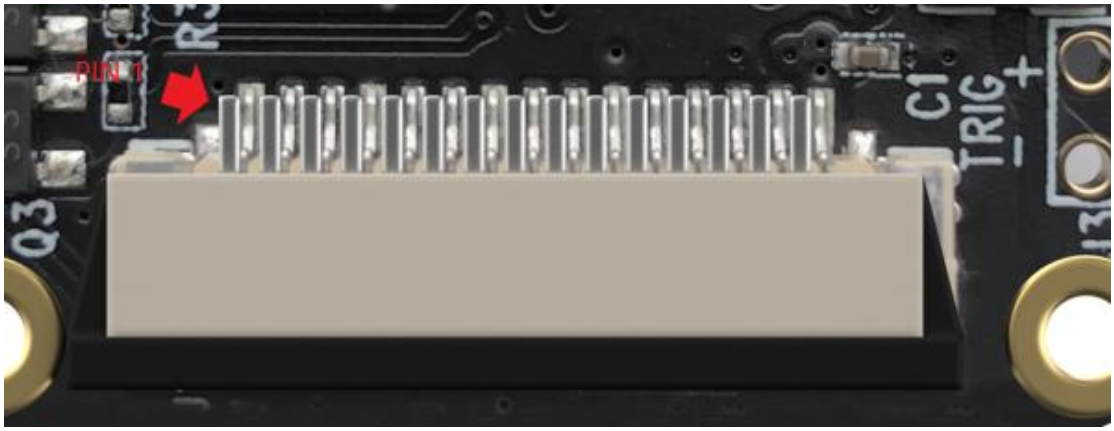
https://www.raspberrypi.com/documentation/accessories/camera.html#install-a-raspberry-pi-camera

3.4 Pins-Out

3.4.1 Signal/Power Connector J1

The J1 pin map is same Raspberry Pi camera.

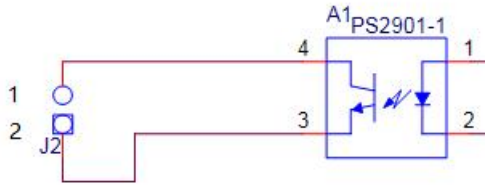




PIN	Symbol	Description
1	GND	Ground Pin
2	CON_CSI_D0N	Pixel Data Lane0 Negative
3	CON_CSI_D0P	Pixel Data Lane0 Positive
4	GND	Ground Pin
5	CON_CSI_D1N	Pixel Data Lane1 Negative
6	CON_CSI_D1P	Pixel Data Lane1Positive
7	GND	Ground Pin
8	CON_CSI_CLKN	Pixel Clock Output Form Sensor Negative
9	CON_CSI_CLKP	Pixel Clock Output Form Sensor Positive
10	GND	Ground Pin
11	None	None
12	None	None
13	SCL	CLK input, SIO_C of SCCB
14	SDA	DATA input,SIO_D of SCCB
15	3.3V Power	Power Supply

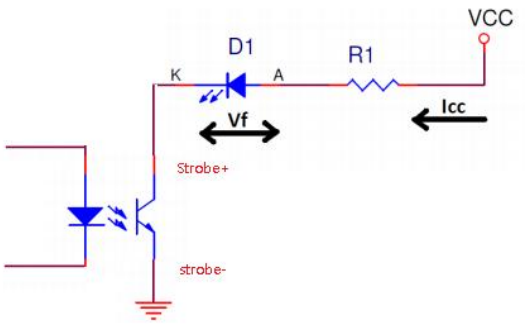
3.4.2 STROB Connector J2

(1)Pin Description



J2 PIN	Symbol
1	STROB+
2	STROB-

(2)Reference Circuit



On-board TLP281 optocoupler isolation, Notice the max collector current is 50mA.

Output Specifications

S. No	Parameter	Test Condition	Value			Unit
			Min	Typ	Max	
1	Driver Voltage (VCC)			12	24	V
2	Drive current (Icc)			10	50	mA
3	Collector Emitter Breakdown Voltage				80	V
4	Collector Emitter Saturation Voltage	Icc = 1 mA		0.1	0.2	V
5	Power Dissipation				150	mW

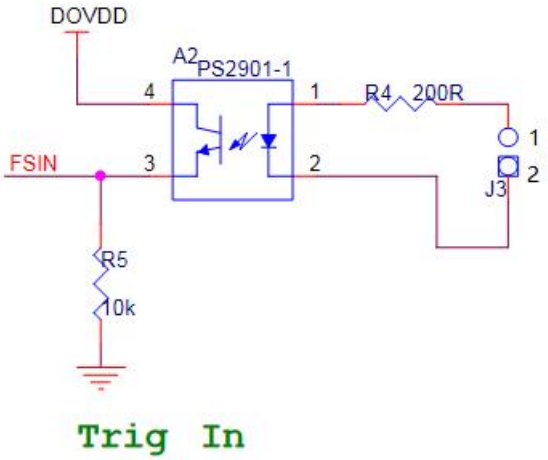
Collector-Emitter Saturation Voltage	$V_{CE(sat)}$	I_F = 10mA, I_C = 1mA		0.1	0.2	V

So If the current required to drive the Flash LED is no more than 50mA
The value of series resistor: R1 = (VCC- Vf - VCE) / If
VCC: system Voltage
Vf: Forward voltage of Flash LED for current Icc
VCE: Collection Emitter voltage, typical:0.1V

If the current required to drive the flash exceeds 50mA, then it is required to drive it with the help of LED driver circuit, and LED driver circuit can be controlled by using the strobe output pin.

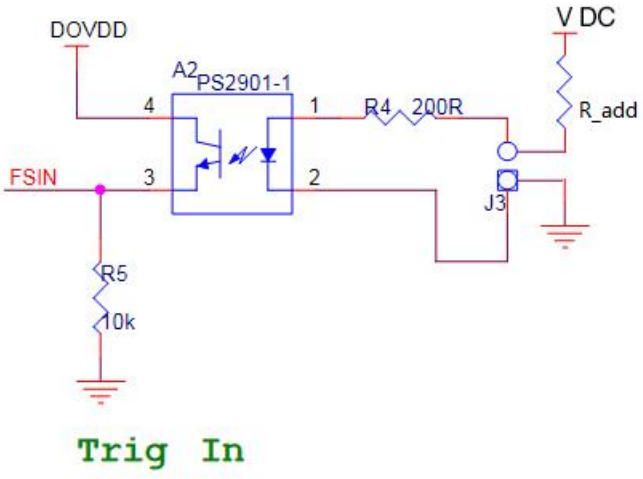
3.4.3 EXT TRIG Connector J3

(1)Pin Description



J3 PIN	Symbol	Description
1	TRIG+	3.3V-5.0V External Trigger Input
2	TRIG-	External GND

(2) Reference Circuit



For example, VCC = 12V, Vf = 1.25V

The calculations done here are based on 12VDC. Please do follow these calculations for other voltages like 24VDC.

Let's take the current through IR LED I_f = 20mA.
Voltage drop across the IR LED = 1.25V
The value of Resistor $R_1 = (V_{cc}-V_f)/I_f = (12 – 1.25)/0.02 = 537.5\ \Omega$
Wattage of resistor $R_1 > I_f^2 * R_1 = 0.02^2*537.5 = 0.215W$
Wattage of the resistor R_1 selected should be greater than 0.215W.

And there is a resistor on board(R4 = 200 Ω), So the R_add = R1 - R4 = 537.5 - 200 = 337.5 Ω

4. Using Raspberry Os Build-in Driver

4.1 Load Raspberry Pi image

Prepare a capacity of more than 8GB TF card(16Gb Class10 is better) and a card reader. Load the image file on to the SD card, using the instructions provided on the Raspberry Pi website for Linux, Mac or PC:
https://www.raspberrypi.com/documentation/computers/getting-started.html#installing-the-operating-system

Raspbian Image download:
https://www.raspberrypi.org/downloads/

4.2 Driver Sources Codes

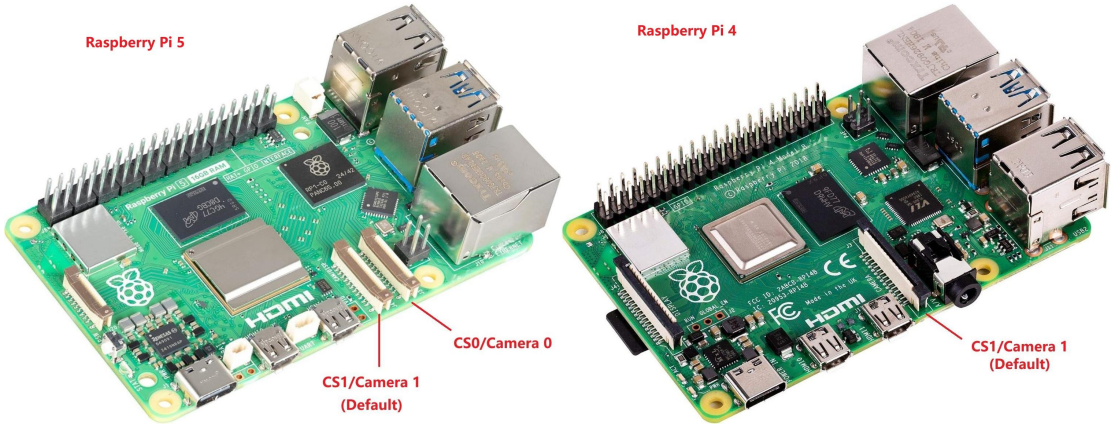
The open source driver on Raspbian:
https://github.com/raspberrypi/linux/blob/rpi-5.10.y/drivers/media/i2c/ov9281.c

4.3 Dtoverlay

(1) Open the config.txt on Raspbian:

New version:
```
sudo nano /boot/firmware/config.txt
```
Legacy version:
```
sudo nano /boot/config.txt
```

(2) Add the dtoverlay into the config.txt file,
The Pi 5 board exposes both the CS0 and CS1 interfaces, while the Pi 4 board only exposes the CS1 interface. However, the CM4 core board exposes both the CS0 and CS1 interfaces. Unless specifically indicated, the Raspberry Pi will generally default to the CSI1 interface.

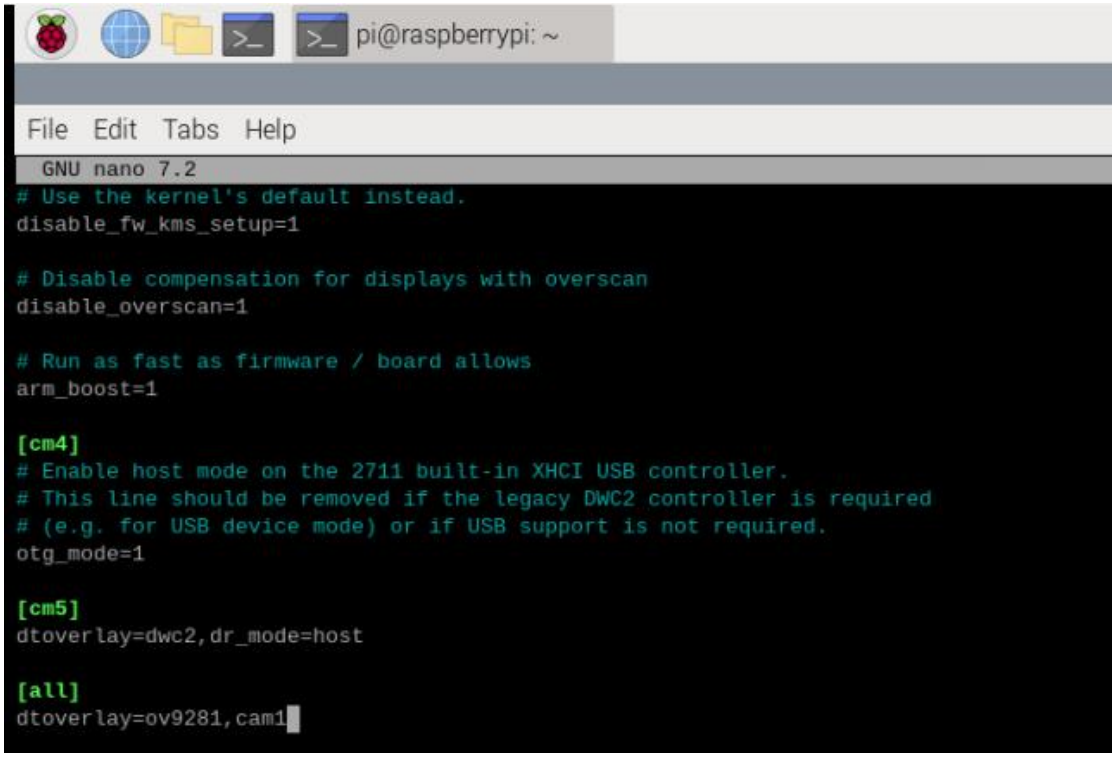


For CSI 0:

dtoverlay=ov9281,cam0

For CSI 1:

dtoverlay=ov9281,cam1



(3) And then press ‘ctrl+ x’ to exit nad press ‘y’ to save.

(4) Rebooted your Pi

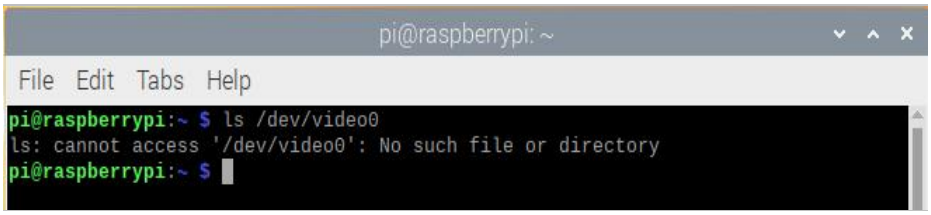
```
sudo reboot
```

(5) Use below command to check the camera is ready.

```
ls /dev/video*
```

Successful:

Unsuccessful:



4.4 Rpicam-apps / Libcamera

libcamera is an open source Linux community project. More information is available at the libcamera website:
https://libcamera.org/

The libcamera source code can be found and checked out from the official libcamera repository.
https://git.linuxtv.org/libcamera.git/

Raspberry Pi OS Bookworm renamed the camera capture applications from libcamera-* to rpicam-*. Symbolic links allow users to use the old names for now. Adopt the new application names as soon as possible. Raspberry Pi OS versions prior to Bookworm still use the libcamera-* name.
https://www.raspberrypi.com/documentation/computers/camera_software.html#rpicam-apps
https://github.com/raspberrypi/rpicam-apps

Here are some simple commands, for more details, please refer to the official Raspberry Pi documentation

3.4.1 Camera Information.

Enter the following command to query camera information, and you will be able to see the data formats and frame rates supported by OV9281 on the Raspberry Pi. R8 and R10 represent the RAW8 and RAW10 original data formats, respectively.

rpicam-hello --list-cameras

```
pi@raspberrypi:~ $ rpicam-hello --list-cameras
Available cameras
-----------------
0 : ov9281 [1280x800 10-bit MONO] (/base/axi/pcie@1000120000/rp1/i2c@88000/ov9281@60)
    Modes: 'R8' : 640x400 [309.79 fps - (0, 0)/1280x800 crop]
                  1280x720 [171.79 fps - (0, 0)/1280x720 crop]
                  1280x800 [143.66 fps - (0, 0)/1280x800 crop]
           'R10_CSI2P' : 640x400 [247.83 fps - (0, 0)/1280x800 crop]
                         1280x720 [137.42 fps - (0, 0)/1280x720 crop]
                         1280x800 [114.93 fps - (0, 0)/1280x800 crop]

1 : ov9281 [1280x800 10-bit MONO] (/base/axi/pcie@1000120000/rp1/i2c@80000/ov9281@60)
    Modes: 'R8' : 640x400 [309.79 fps - (0, 0)/1280x800 crop]
                  1280x720 [171.79 fps - (0, 0)/1280x720 crop]
                  1280x800 [143.66 fps - (0, 0)/1280x800 crop]
           'R10_CSI2P' : 640x400 [247.83 fps - (0, 0)/1280x800 crop]
                         1280x720 [137.42 fps - (0, 0)/1280x720 crop]
                         1280x800 [114.93 fps - (0, 0)/1280x800 crop]
```

3.4.2 Camera Preview

The following command can be used to directly open the camera preview. Two cameras can be opened simultaneously.

For CSI 0:

rpicam-hello -t 0 --camera 0

For CSI 1:

rpicam-hello -t 0 --camera 1

3.4.3 Camera Modes

You can use the result of --list-cameras, and then specify the camera operating mode with the --mode option in a colon-separated format:

<width>:<height>:<bit-depth>:<packing>

If the values you provide do not match exactly, the system will select the closest available option for the sensor. You can use either packed (P) or unpacked (U) packing formats. This affects the format of stored video and still images, but it does not affect the format of frames passed to the preview window. The default bit-depth is 12, and the default packing is P (packed).

Set the working mode of camera 1 to 640x480, RAW8, and 300 frame rate using the following command

rpicam-still --viewfinder-mode 640:400:8 --framerate 300 -t 0 --camera 1

Set the working mode of camera 1 to 640x480, RAW8, and 300 frame rate using the following command

```
rpicam-still --viewfinder-mode 1280:800:10 --framerate 114 -t 0 --camera 1
```

4.5 Rpicam-apps / Libcamera Trigger Mode

3.4.1 Download Tools

When using trigger mode, you need to download the trigger toolkit from our GitHub. You will see a folder named **OV9281_libcamera**.

Download link:

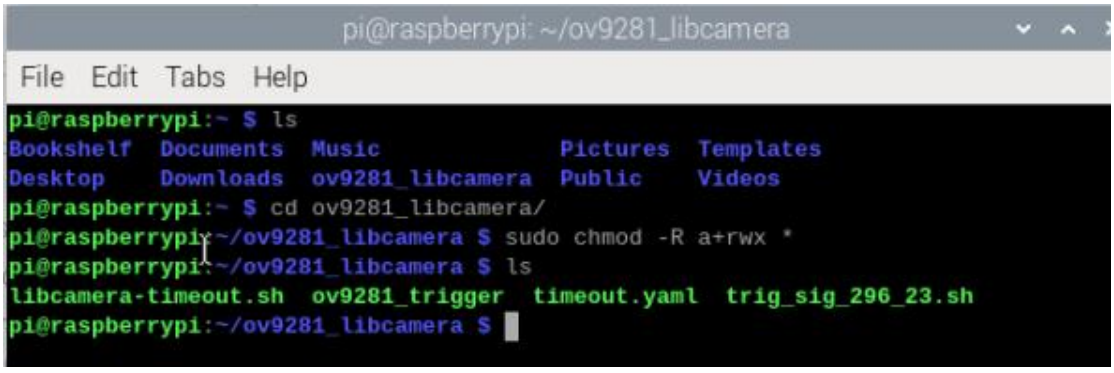
https://github.com/INNO-MAKER/CAM-OV9281RAW-V2

Alternatively, you can enter the following commands in the terminal:

```
git clone https://github.com/INNO-MAKER/CAM-OV9281RAW-V2.git
```

Then, enter the folder directory and change the folder's permissions.

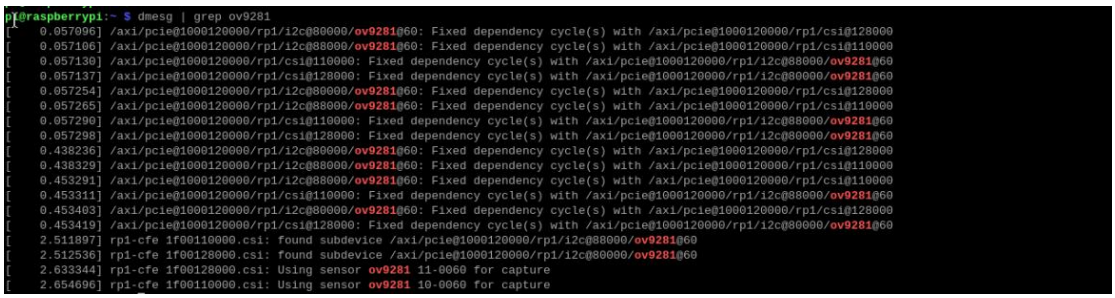
```
cd ov9281_libcamera
sudo chmod -R a+rwx *
```



3.4.2 Check The I2C Bus Addresses

To check the I2C bus addresses, use the following command. Since two OV9821 cameras are connected, you should see addresses 10 and 11. Typically, address 11 corresponds to camera 1, and address 10 corresponds to camera 0. Generally, these are fixed addresses, but occasionally, after a Raspberry Pi system upgrade, this numbering may change

dmesg | grep ov9281



3.4.3 Time out Setting

If the cameras are not all started within 1 second, the rpicam applications can time out. To prevent this, you must edit a configuration file on any Raspberry Pi(s) with sink cameras. You can follow the official documentation to perform the operation and edit the copy.

> **Libcamera configuration**
>
> If the cameras are not all started within 1 second, the `rpicam` applications can time out. To prevent this, you must edit a configuration file on any Raspberry Pi(s) with sink cameras.
>
> On Raspberry Pi 5 or CM5:
>
> ```
> $ cp /usr/share/libcamera/pipeline/rpi/pisp/example.yaml timeout.yaml
> ```
>
> On other Raspberry Pi models:
>
> ```
> $ cp /usr/share/libcamera/pipeline/rpi/vc4/rpi_apps.yaml timeout.yaml
> ```
>
> Now edit the copy. In both cases, delete the # (comment) from the "`camera_timeout_value_ms`": line, and change the number to 60000 (60 seconds).

Alternatively, you can directly use the file provided in the ov9281_libcamera folder.

sources ./libcamera-timeout.sh

Then running libcamera tool to start preview. Please note that the command to start the preview must be run in the same terminal window as the script that sets the timeout. Do not open a new terminal window to run it.

For example :

libcamera-hello -t 0 --camera 0 Or

libcamera-hello -t 0 --camera 1

3.4.4 Start/Stop Trigger Mode

The usage of the script to start and stop the trigger mode is as follows:

```
./ov9281_trigger    [i2c bus] [on/off]
```

Open a new terminal window, and based on the previously obtained I2C bus numbers, you can use the following commands to start/stop the trigger for CAMERA0 and CAMERA1. After the trigger is successfully started, you will see the preview window remain still

camera 1 trigger on:

```
./ov9281_trigger 11 1
```

camera 1 trigger off:

```
./ov9281_trigger 11 0
```

camera 0 trigger on:

```
./ov9281_trigger 10 1
```

camera 0 trigger off:

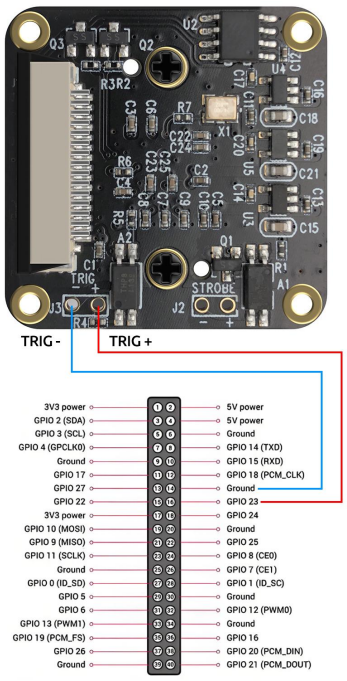
```
./ov9281_trigger 10 0
```

```
pi@raspberrypi:~/ov9281_libcamera $ ./ov9281_trigger 11 1
InnoMaker OV9281 camera Trigger mode setting tools
Setting Trigger mode on
pi@raspberrypi:~/ov9281_libcamera $ ./ov9281_trigger 11 0
InnoMaker OV9281 camera Trigger mode setting tools
Setting Trigger mode off
```

3.4.5 Generate a Trigger Signal

At this point, a trigger signal needs to be provided to the OV9281 module. We have provided a test script that generates a rising edge every 2 seconds on pin 23 of the 40-pin header on the Raspberry Pi to provide the trigger signal.

```
./trig_sig_pin_23.sh
```



6. Versions Description

Version	Description	Date	E-mail
V1.0		2022.01.02	support@inno-maker.com calvin@inno-maker.com
V1.1	Update Pictures	2022.04.03	support@inno-maker.com calvin@inno-maker.com
V1.2	Update PCB size	2022.05.06	support@inno-maker.com calvin@inno-maker.com
V1.3	Change all the content in the manual to support the Raspberry Pi's built-in drivers.	2025.07.31	support@inno-maker.com calvin@inno-maker.com

If you have any suggestions, ideas, codes and tools please feel free to email to me. I will update the user manual and record your name and E-mail in list. Look forward to your letter and kindly share.