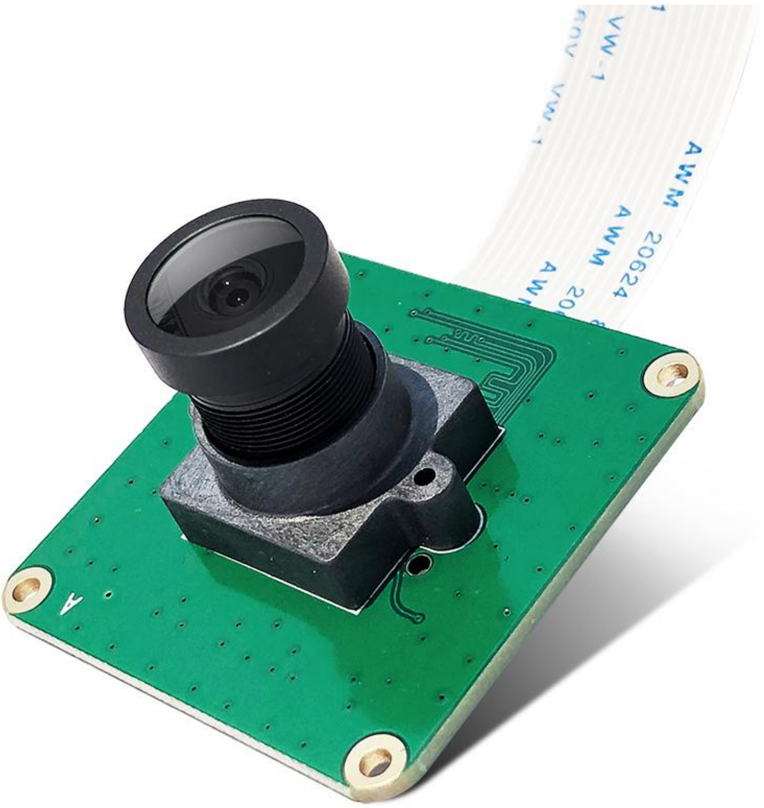
CAM-MIPI327RAW UserManual



1. General

CAM-MIPI327RAW is a low-cost fisheye Lens excellent low illumination and wide dynamic camera module, designed for whole series Raspberry(P4/Pi3B+/PI3A+/PI3/CM3/CM3+). Plug into the CSI-2 Pi camera interface directly.

On-board IMX327LQR-C is a diagonal 6.46 mm (Type 1/2.8) CMOS active pixel type solid-state image sensor with a square pixel array and 2.13 M effective pixels. This chip operates with analog 2.9 V, digital 1.2 V, and interface 1.8 V triple power supply, and has low power consumption. High sensitivity, low dark current and no smear are achieved through the adoption of R, G and B primary color mosaic filters. This chip features an electronic shutter with variable charge integration time.

2. Features

1. Raspberry Pi Camera CAM-MIPI327RAW is an Industrial Camera Module for Raspberry Pi 4,3B+ 3B, 3A+ CM3+, CM3, Pi zero.

2. Sensor Type: STARVIS IMX327LQR Color CMOS Active Pixel type Solid-state Image Sensor with Square Pixel Array 2.13 M Effective Pixels. Diagonal 6.46 mm (Type 1/2.8). Pixel count: 1920x1080. High to 60 fps.

3. This IMX327LQR module connects to the CSI connector on the Raspberry Pi directly. CSI-2 output: 2 lanes/RAW10 or RAW12. Support V4L2 and libcamera.

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

5. Comes with user manual, test demo and friendly technology support. We offer custom design service.

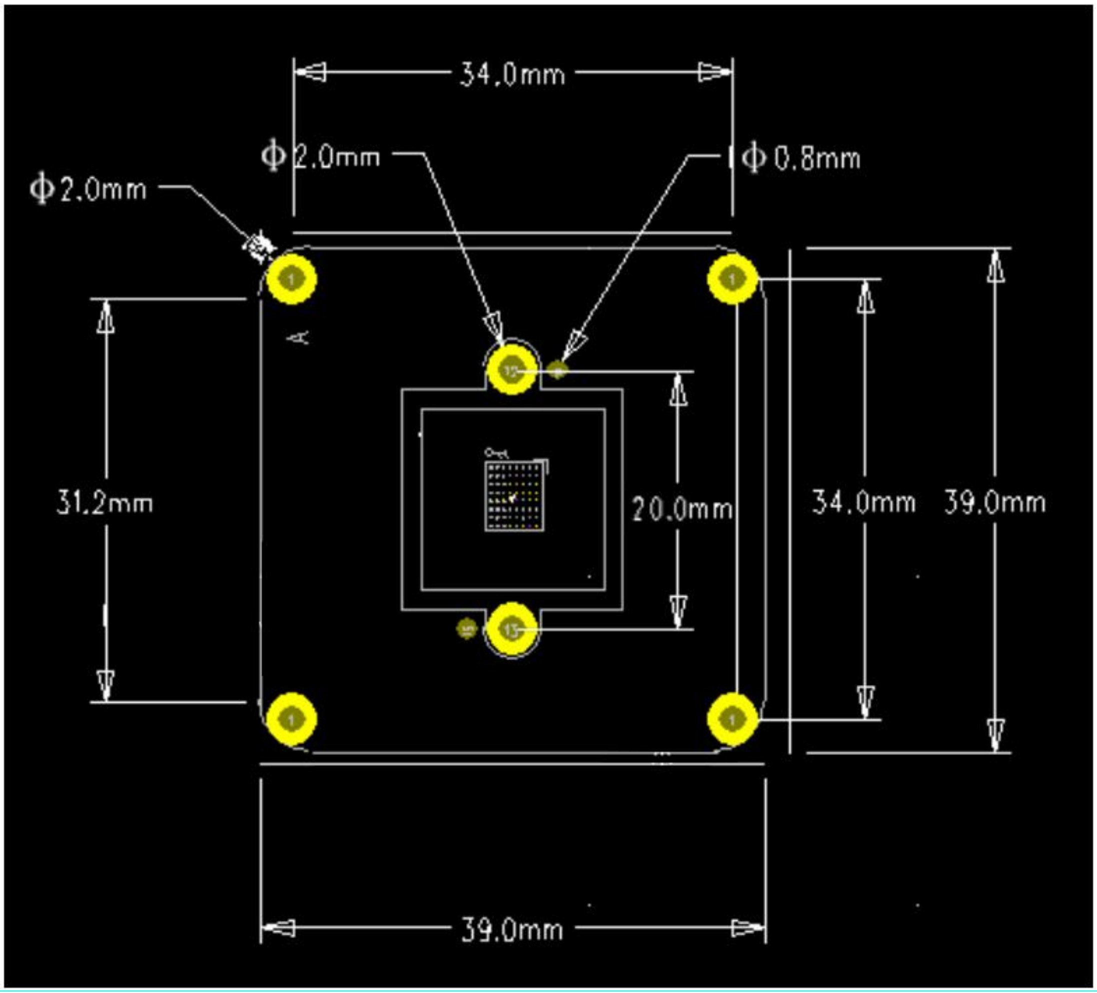
3. Hardware Description

3.1Overview

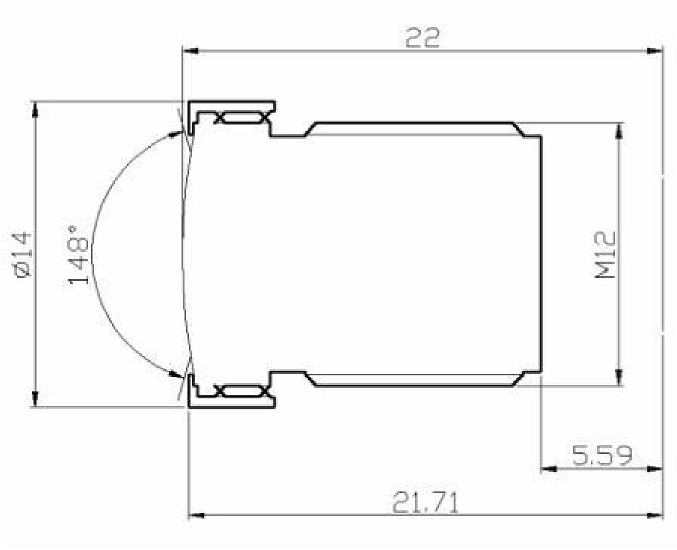
Features	
Size	39mm x 39mm
Weight	4g
Still Resolution	2.07M pixels
Video Modes	1920x1080 60fps, SRGGB10P
Linux integration	V4L2 Libcamera
Sensor	IMX327LQR-C
Sensor Resolution	1920 (H) × 1080 (V)
Sensor image area	5568 µm (H)*3132 µm (V)
Pixel size	2.9 µm (H) × 2.9 µm (V)
Optical size	1/2.8"
S/N ratio	55dB
Dynamic range	120db
Output interface	global shutter
Output formats	RAW10/RAW12
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

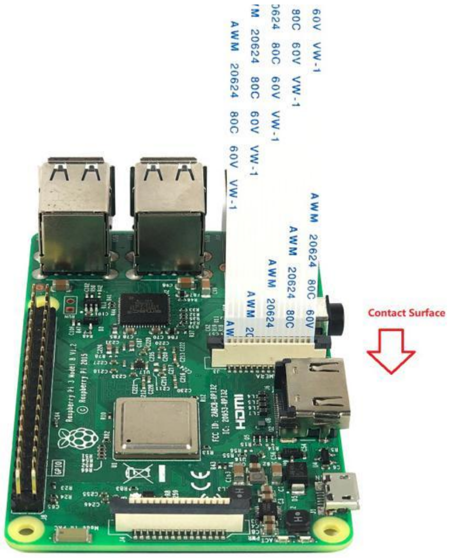
3.2.1 PCB Size

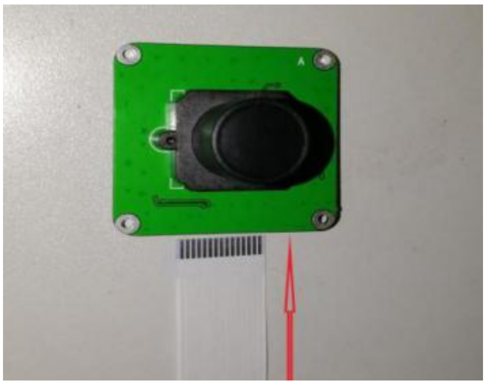


3.2.2 Len Size



3.3 Connection Of The Hardware

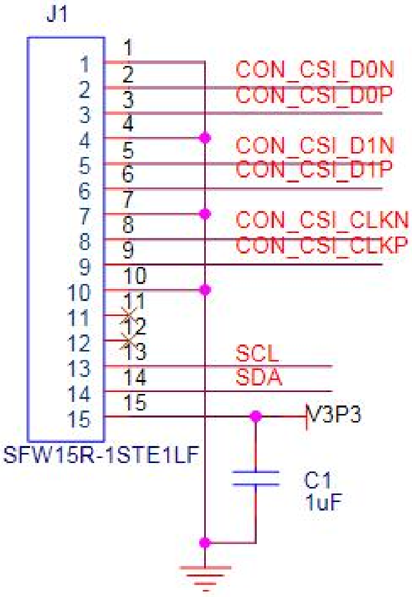


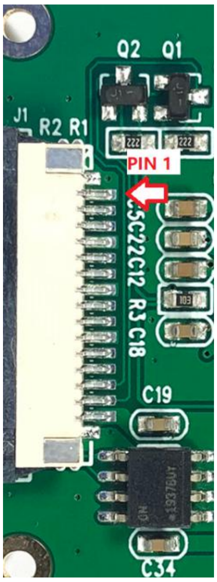


3.4 Pin-Out

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

4. Using Raspbian 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.org/documentation/installation/installing-images/README.md

Raspbian Image download:

https://www.raspberrypi.org/downloads/

4.2 Driver Sources Codes

The existing IMX290 driver and overlay also work fine with IMX327 and IMX426.

The open source driver on Raspbian:

https://github.com/raspberrypi/linux/blob/rpi-5.10.y/drivers/media/i2c/imx290.c

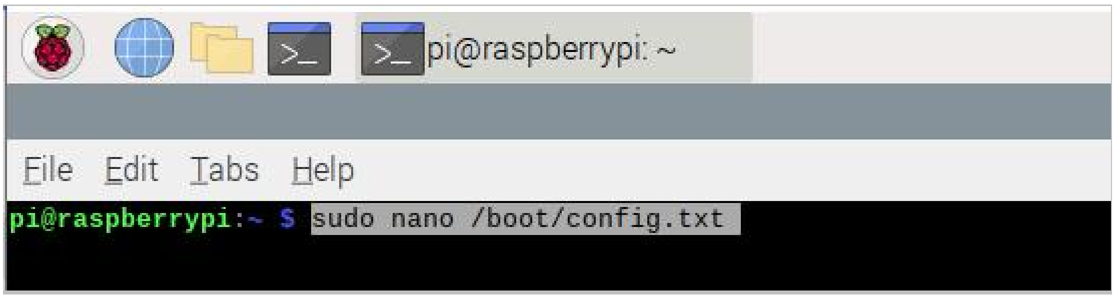
Reference codes：

https://github.com/torvalds/linux/blob/master/drivers/media/i2c/imx290.c

4.3 Dtoverlay on Raspberry Pi board（RPI4/3/Zero）

(1) Open the config.txt on Raspbian:

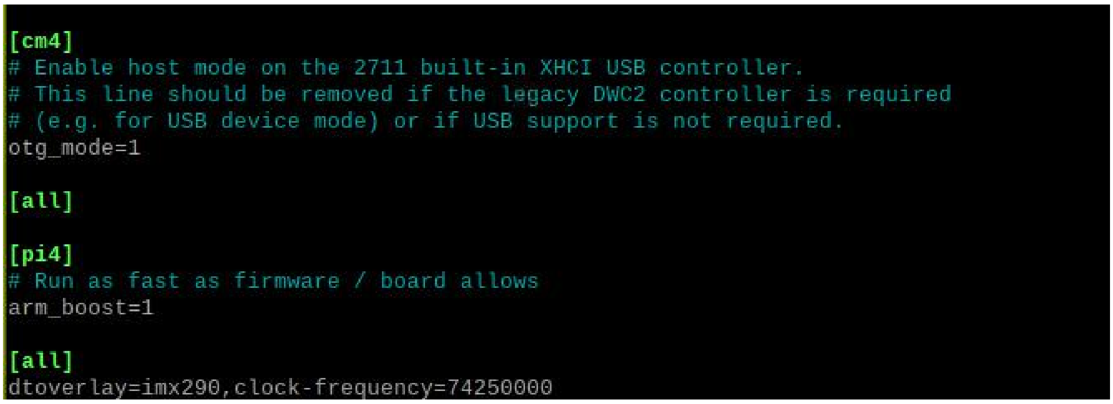
```
sudo nano /boot/config.txt
```



```
pi@raspberrypi:~ $ sudo nano /boot/config.txt
```

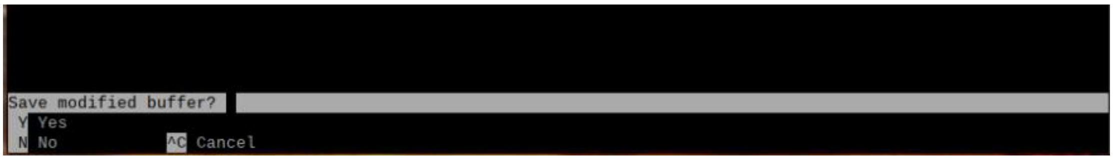
(2) Add the dtoverlay into the config.txt file,

dtoverlay=imx290,clock-frequency=74250000



```
[cm4]
# Enable host mode on the 2711 built-in XHCI USB controller.
# This line should be removed if the legacy DWC2 controller is required
# (e.g. for USB device mode) or if USB support is not required.
otg_mode=1

[all]

[pi4]
# Run as fast as firmware / board allows
arm_boost=1

[all]
dtoverlay=imx290,clock-frequency=74250000
```

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



```
Save modified buffer?
 Y Yes
 N No           ^C Cancel
```

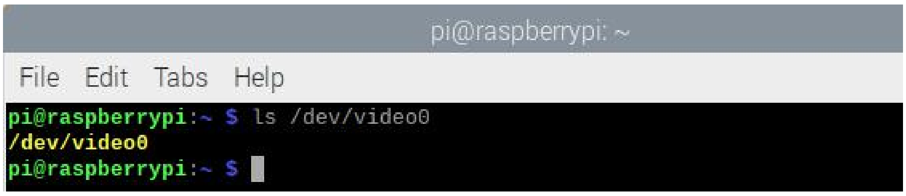
(4) Rebooted your Pi

sudo reboot

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

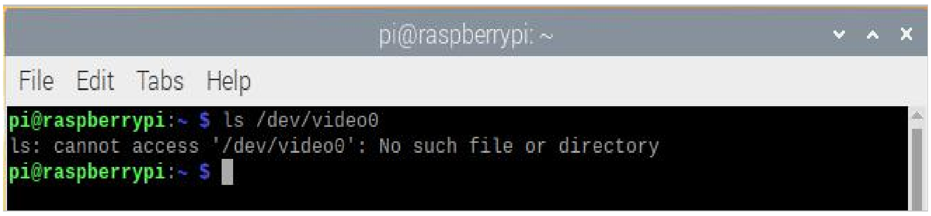
ls /dev/video0

Successful:



```
pi@raspberrypi:~ $ ls /dev/video0
/dev/video0
pi@raspberrypi:~ $
```

Unsuccessful:

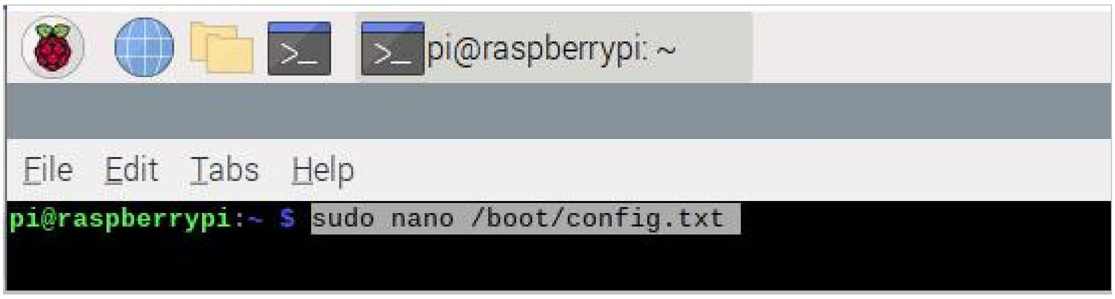


```
pi@raspberrypi:~ $ ls /dev/video0
ls: cannot access '/dev/video0': No such file or directory
pi@raspberrypi:~ $
```

4.4 Dtoverlay on CM4

(1) Open the config.txt on Raspbian:

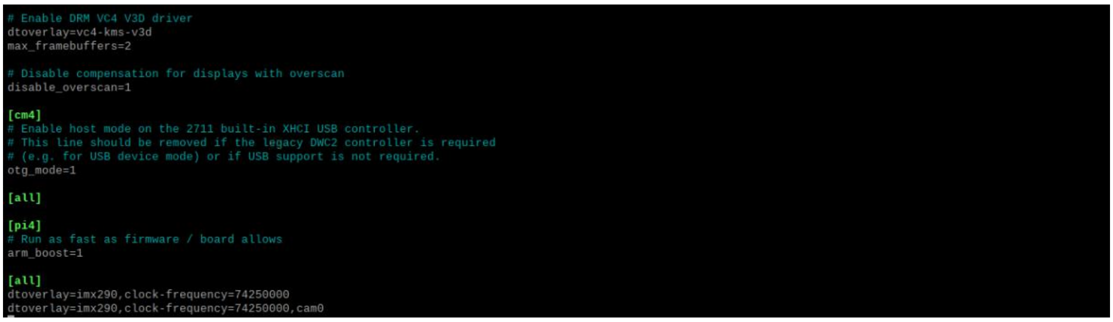
```
sudo nano /boot/config.txt
```



```
pi@raspberrypi:~ $ sudo nano /boot/config.txt
```

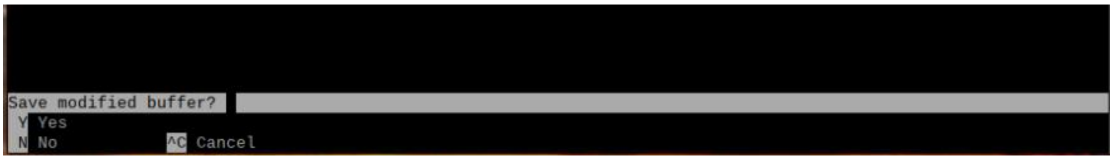
(2) Add the dtoverlay into the config.txt file, CAM1 is the default camera same as Raspberry Pi.

```
dtoverlay=imx290,clock-frequency=74250000
dtoverlay=imx290,clock-frequency=74250000,cam0
```



```
# Enable DRM VC4 V3D driver
dtoverlay=vc4-kms-v3d
max_framebuffers=2

# Disable compensation for displays with overscan
disable_overscan=1

[cm4]
# Enable host mode on the 2711 built-in XHCI USB controller.
# This line should be removed if the legacy DWC2 controller is required
# (e.g. for USB device mode) or if USB support is not required.
otg_mode=1

[all]

[pi4]
# Run as fast as firmware / board allows
arm_boost=1

[all]
dtoverlay=imx290,clock-frequency=74250000
dtoverlay=imx290,clock-frequency=74250000,cam0
```

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



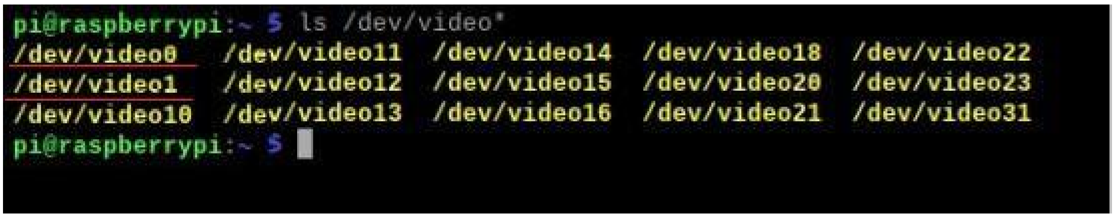
```
Save modified buffer?
 Y Yes
 N No           ^C Cancel
```

(7) Rebooted your Pi

```
sudo reboot
```

(8) Use below command to check the camera video0 and video1 are available.

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



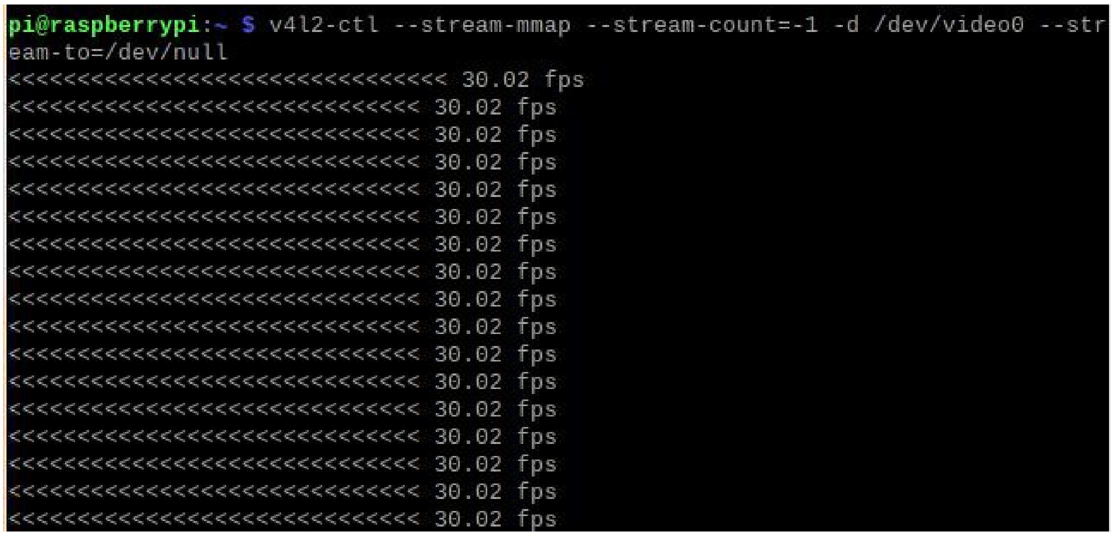
```
pi@raspberrypi:~ $ ls /dev/video*
/dev/video0   /dev/video11  /dev/video14  /dev/video18  /dev/video22
/dev/video1   /dev/video12  /dev/video15  /dev/video20  /dev/video23
/dev/video10  /dev/video13  /dev/video16  /dev/video21  /dev/video31
pi@raspberrypi:~ $ 
```

4.5 Frame Rate(fps) Test

Use below command, you can see frames-per-second information of your camera.

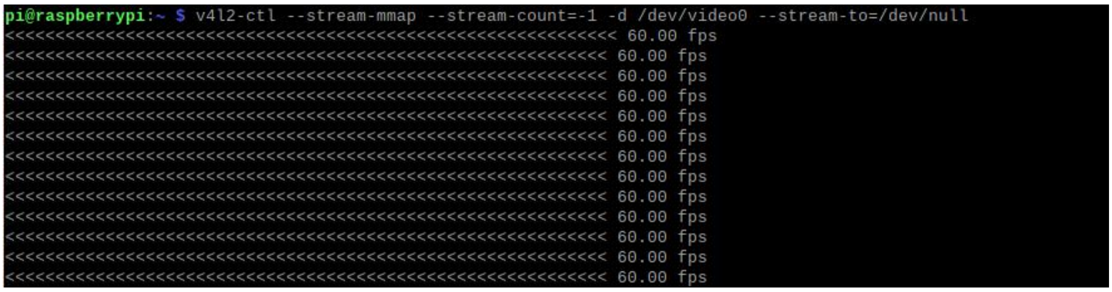
```
v4l2-ctl --stream-mmap --stream-count=-1 -d /dev/video0 --stream-to=/dev/null
```

30 fps:



```
pi@raspberrypi:~ $ v4l2-ctl --stream-mmap --stream-count=-1 -d /dev/video0 --str
eam-to=/dev/null
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 30.02 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 30.02 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 30.02 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 30.02 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 30.02 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 30.02 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 30.02 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 30.02 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 30.02 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 30.02 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 30.02 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 30.02 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 30.02 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 30.02 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 30.02 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 30.02 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 30.02 fps
```

60 fps:



```
pi@raspberrypi:~ $ v4l2-ctl --stream-mmap --stream-count=-1 -d /dev/video0 --stream-to=/dev/null
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 60.00 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 60.00 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 60.00 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 60.00 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 60.00 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 60.00 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 60.00 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 60.00 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 60.00 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 60.00 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 60.00 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 60.00 fps
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< 60.00 fps
```

5. Libcamera On Raspbian

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/

When running a Raspberry Pi OS based on Bullseye, the 5 basic libcamera-apps are already installed. In this case, official Raspberry Pi cameras will also be detected and enabled automatically. Below we only take 'libcamera-hello' for example. For more information, please refer to below link：

https://www.raspberrypi.com/documentation/accessories/camera.html#binary-packages

In addition, Below test base on Raspberry Pi 4, If you want libcamera apps on Bullseye running on Pi 0 - 3, please refer to below link:

https://forums.raspberrypi.com/viewtopic.php?t=323547

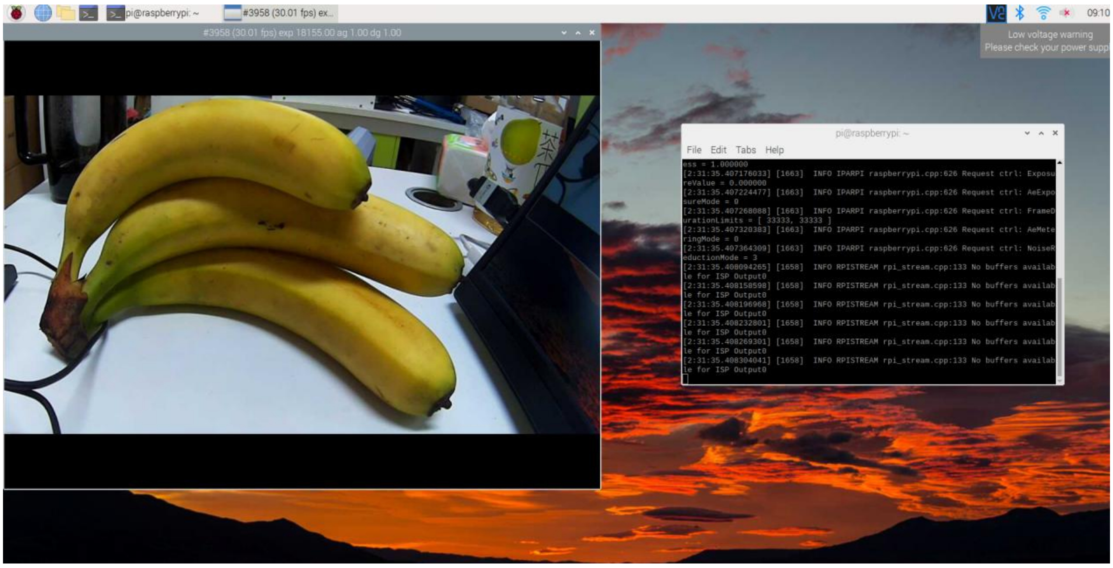
5.1 Libcamera-hello

libcamera-hello is the equivalent of a "hello world" application for the camera. It starts the camera, displays a preview window.

5.1.1 Preview

Use below command to start the preview.

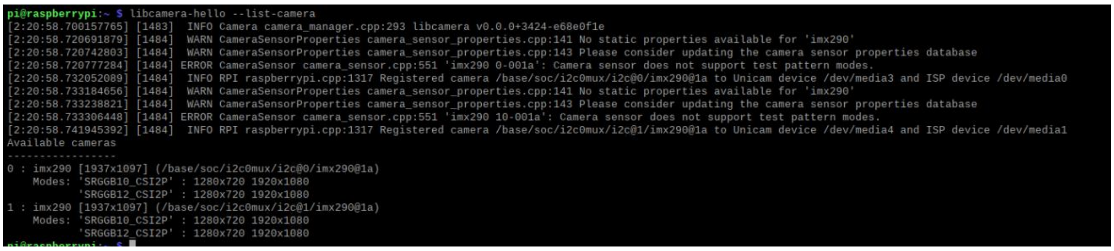
```
libcamera-hello -t 0
```



```
ess = 1.000000
[2:31:35.407176033] [1663]  INFO IPARPI raspberrypi.cpp:626 Request ctrl: Exposu
reValue = 0.000000
[2:31:35.407224477] [1663]  INFO IPARPI raspberrypi.cpp:626 Request ctrl: AeExpo
sureMode = 0
[2:31:35.407268088] [1663]  INFO IPARPI raspberrypi.cpp:626 Request ctrl: FrameD
urationLimits = [ 33333, 33333 ]
[2:31:35.407320383] [1663]  INFO IPARPI raspberrypi.cpp:626 Request ctrl: AeMete
ringMode = 0
[2:31:35.407364309] [1663]  INFO IPARPI raspberrypi.cpp:626 Request ctrl: NoiseR
eductionMode = 3
[2:31:35.408094265] [1658]  INFO RPISTREAM rpi_stream.cpp:133 No buffers availab
le for ISP Output0
[2:31:35.408158598] [1658]  INFO RPISTREAM rpi_stream.cpp:133 No buffers availab
le for ISP Output0
[2:31:35.408196968] [1658]  INFO RPISTREAM rpi_stream.cpp:133 No buffers availab
le for ISP Output0
[2:31:35.408232801] [1658]  INFO RPISTREAM rpi_stream.cpp:133 No buffers availab
le for ISP Output0
[2:31:35.408269301] [1658]  INFO RPISTREAM rpi_stream.cpp:133 No buffers availab
le for ISP Output0
[2:31:35.408304041] [1658]  INFO RPISTREAM rpi_stream.cpp:133 No buffers availab
le for ISP Output0
```

If you are using dual camera on the CM4 module ,please use below command, you could see two camera are available in you system.

```
libcamera-hello --list-camera
```



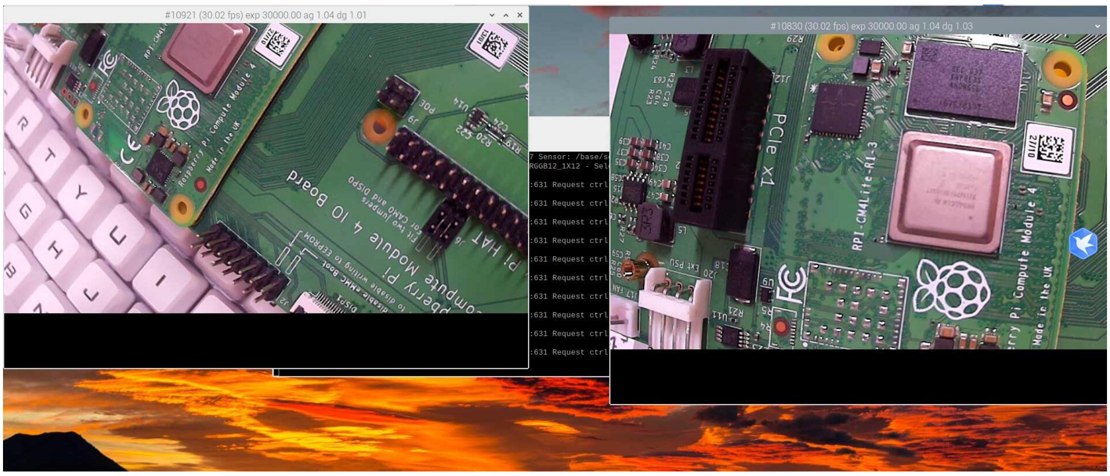
```
pi@raspberrypi:~ $ libcamera-hello --list-camera
[2:20:58.700157765] [1483]  INFO Camera camera_manager.cpp:293 libcamera v0.0.0+3424-e68e0f1e
[2:20:58.720691879] [1484]  WARN CameraSensorProperties camera_sensor_properties.cpp:141 No static properties available for 'imx290'
[2:20:58.720742803] [1484]  WARN CameraSensorProperties camera_sensor_properties.cpp:143 Please consider updating the camera sensor properties database
[2:20:58.720777284] [1484] ERROR CameraSensor camera_sensor.cpp:551 'imx290 0-001a': Camera sensor does not support test pattern modes.
[2:20:58.732052089] [1484]  INFO RPI raspberrypi.cpp:1317 Registered camera /base/soc/i2c0mux/i2c@0/imx290@1a to Unicam device /dev/media3 and ISP device /dev/media0
[2:20:58.733184656] [1484]  WARN CameraSensorProperties camera_sensor_properties.cpp:141 No static properties available for 'imx290'
[2:20:58.733238821] [1484]  WARN CameraSensorProperties camera_sensor_properties.cpp:143 Please consider updating the camera sensor properties database
[2:20:58.733306448] [1484] ERROR CameraSensor camera_sensor.cpp:551 'imx290 10-001a': Camera sensor does not support test pattern modes.
[2:20:58.741945392] [1484]  INFO RPI raspberrypi.cpp:1317 Registered camera /base/soc/i2c0mux/i2c@1/imx290@1a to Unicam device /dev/media4 and ISP device /dev/media1
Available cameras
-----------------
0 : imx290 [1937x1097] (/base/soc/i2c0mux/i2c@0/imx290@1a)
    Modes: 'SRGGB10_CSI2P' : 1280x720 1920x1080
           'SRGGB12_CSI2P' : 1280x720 1920x1080
1 : imx290 [1937x1097] (/base/soc/i2c0mux/i2c@1/imx290@1a)
    Modes: 'SRGGB10_CSI2P' : 1280x720 1920x1080
           'SRGGB12_CSI2P' : 1280x720 1920x1080
pi@raspberrypi:~ $
```

Enable camera 0

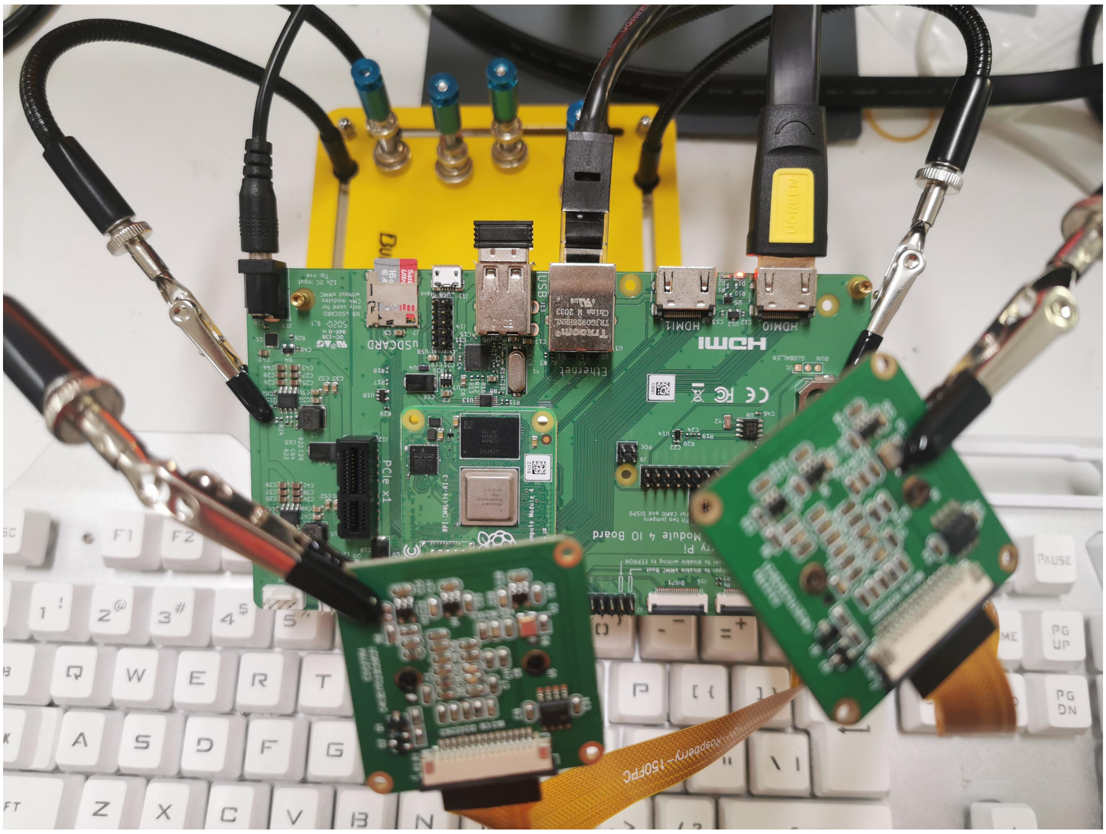
```
libcamera-hello -t 0 --camera 0
```

Enable camera 1

```
libcamera-hello -t 0 --camera 1
```


#10921 (30.02 fps) exp 30000.00 ag 1.04 dg 1.01
#10830 (30.02 fps) exp 30000.00 ag 1.04 dg 1.03

```
7 Sensor: /base/s
RGGB12_1X12 - Sel
:631 Request ctrl
:631 Request ctrl
:631 Request ctrl
:631 Request ctrl
:631 Request ctrl
:631 Request ctrl
:631 Request ctrl
:631 Request ctrl
:631 Request ctrl
:631 Request ctrl
```



5.1.2 Set the Frame Rate

The IMX327 modue can support up 60fps.

https://github.com/raspberrypi/linux/pull/4398/commits/192569ee019ad4bd642dc7012e176d1aa87ccd2c

https://github.com/raspberrypi/linux/pull/4398

```
libcamera-hello --framerate 60
```

5.1.2 Set the Bit-Depth

```
libcamera-hello    -t 10000 --mode 1920:1080:12:U
```

Above command set IMX327 work on 12bit , 1920(W) * 1080(H) unpacked mode. For more detail

--mode arg :Camera mode as W:H:bit-depth:packing, where packing is P (packed) or U(unpacked)

6. User Manual Version Descriptions

Version	Description	Date	E-mail
V1.0		2021.11.25	support@inno-maker.com sales@inno-maker.com
V1.1	Add camera runing on the CM4	2022.07.22	support@inno-maker.com sales@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.