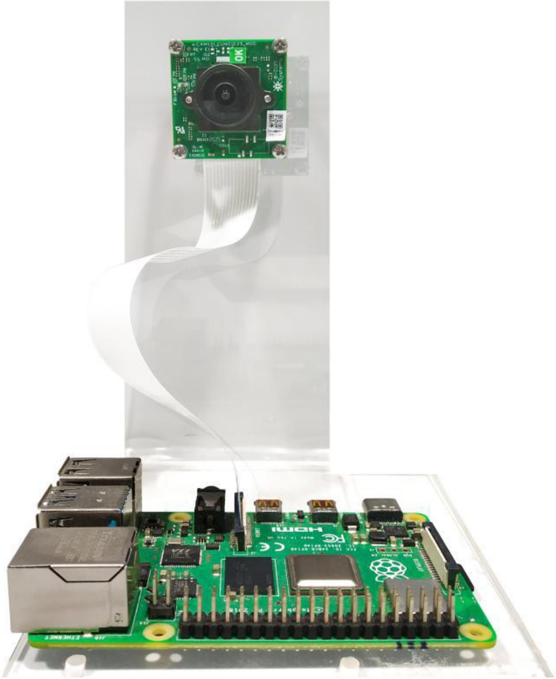
e-CAM20_CURB

Getting Started Manual



Version 1.4

e-con Systems

30/11/2021

e-con Systems

Your Product Development Partner

Disclaimer

The specifications of e-CAM20_CURB camera module and instructions on how to use this camera module with Raspberry Pi 4 are provided as reference only and e-con Systems reserves the right to edit/modify this document without any prior intimation of whatsoever.

Contents

Introduction to e-CAM20_CURB

e-CAM20_CURB is a 2.3 MP fixed focus, global shutter color camera with MIPI 2-lane camera module designed and developed by e-con Systems, a leading Embedded Product Design Services Company which specializes in the advanced camera solutions. It is a two-board solution which includes eCAM217_CUMI0234_MOD and ACC-XVRNX-MIPICAMERA-ADP boards. This camera module can be directly interfaced to the Raspberry Pi 4 using a FPC cable and has support on yocto custom build OS.

e-CAM20_CURB can stream uncompressed HD(1280x720) at 120 FPS , FHD (1920x1080) at 65 FPS and 2.3 MP (1920x1200) at 60 FPS.

This document describes how to interface and use the e-CAM20_CURB with Raspberry Pi 4.

Prerequisites

The prerequisites are as follows:

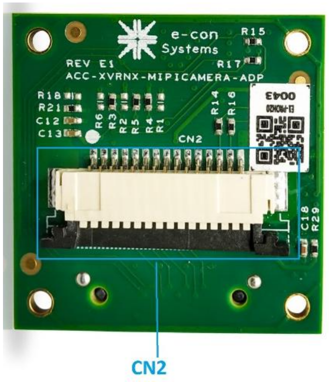
- Micro SD card with minimum 16GB capacity
- Raspberry Pi 4 with power supply

Parts Supplied

The following table lists the parts supplied with the kit.

Table 1: Parts Supplied

Parts Supplied	Images	Quantity
Camera Module Board with lens		1

Adapter Board		1
FPC Cable		1

Description

The e-CAM20_CURB board consists of 2.3 MP fixed focus global shutter camera module board and ACC-XVRNX-MIPICAMERA-ADP adapter board. The camera adapter board is based on AR0234 image sensor from ON Semiconductor® and an on-board Image Signal Processor (ISP). The ACC-XVRNX-MIPICAMERA-ADP adapter board connects camera module board and Raspberry Pi 4 using the FFC cable.

The top views of ACC-XVRNX-MIPICAMERA-ADP and e-CAM217_CUMI0234_MOD boards are shown in following figures.

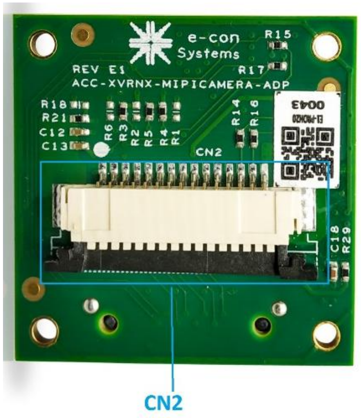


Figure 1: Top View of   ACC-XVRNX-MIPICAMERA-ADP

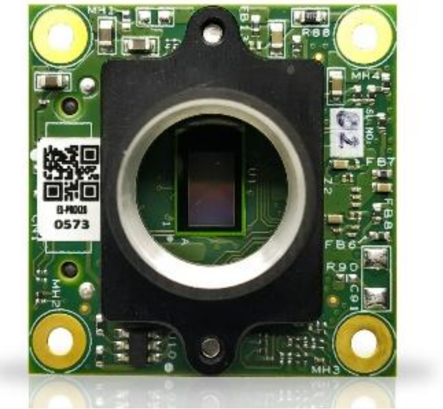


Figure 2: Top View of e-CAM217_CUMI0234_MOD Board

e-CAM20_CURB Board Handling Procedure

This section describes the handling procedure of e-CAM20_CURB board.

The procedure to assemble the board setup are as follows:

1. Interface the e-CAM217_CUMI0234_MOD module board, ACC-XVRNX-MIPICAMERA-ADP adapter board and Raspberry Pi 4 using the provided FPC cable.

   The provided FPC cable is shown in following figure.

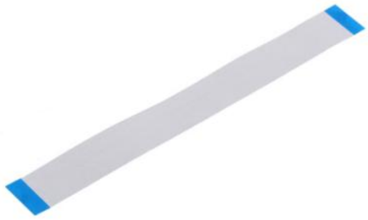


Figure 3: FFC Cable

The insertion and locking position of FFC cable on the ACC-XVRNX-MIPICAMERA-ADP and Raspberry Pi 4 boards are shown in following figures.

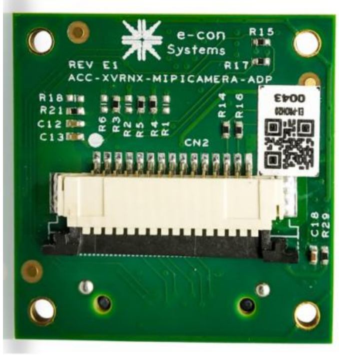


Figure 4: Adapter Board (Lock open)

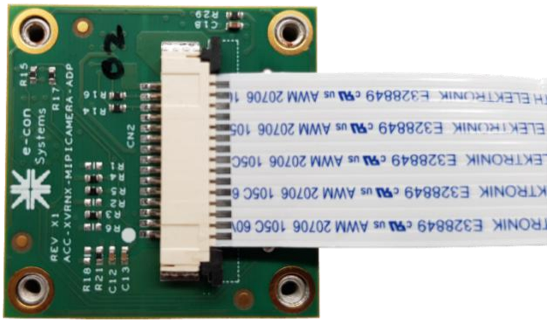


Figure 5: FFC Cable Connector Locked Position

2. Insert the MIPI CSI connectors (CN1, CN2) of the e-CAM217_CUMI0234_MOD module board to MIPI CSI connectors (CN1,CN2) of ACC-XVRNX-MIPICAMERA-ADP adapter board as shown in following figure.

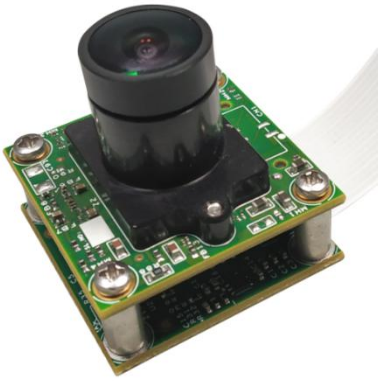


Figure 6: Module and Adapter Board Connection

The connector location of Raspberry Pi 4 is shown in the following figure.

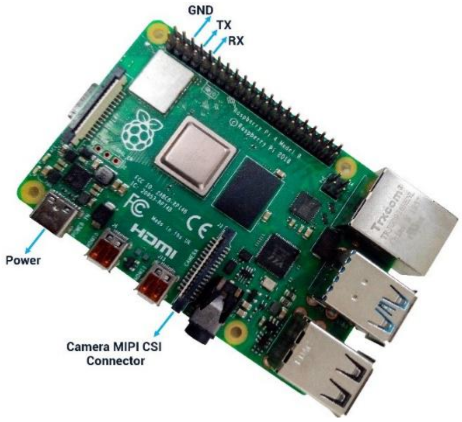


Figure 7: MIPI Camera Connector in Raspberry Pi 4

Once the camera is connected, you must remove the protective cover from the camera lens (if present).

Make sure you connected FPC cable properly as shown in the following figure. The blue side (Non-conductor side) should face ethernet port and conductor side should face HDMI port.

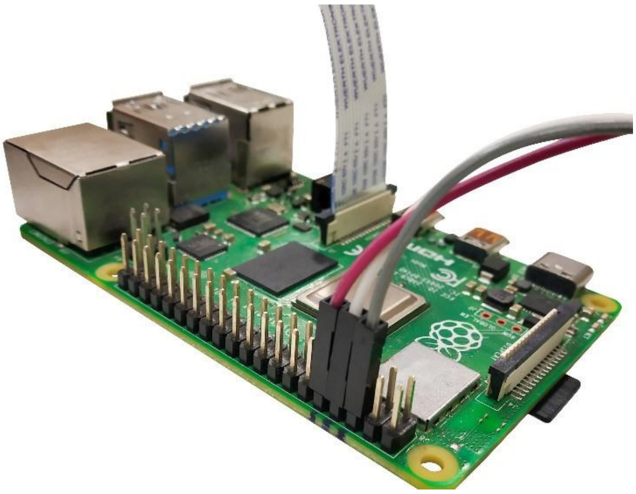


Figure 8: Camera Connection with Development Kit

Software Quick Setup

This section provides the steps to quickly prepare a bootable SD card to boot the Raspberry Pi 4 with e-CAM20_CURB. After successful flashing, the SD card will be bootable. The data in the SD card which is used for flashing needs to be backed up as a preliminary step.

The steps to prepare a bootable SD card with Yocto Image:

1. Connect the SD card of minimum 16GB size to be flashed to the Host PC.
2. Download release packages from the below FTP path

```
Hostname:  ftp://ftp.e-consystems.com
Username: e-CAM20-CURB@ftp.e-consystems.com
Password: Mae20rb%^
```

3. Run the following command to flash the prebuild SD card image in the release package.

```
unzip <file name*>.zip

YOCTO:
sudo dd if=e-con_yocto_e-
CAM20_CURB_Release_<VER>.rpi-sdimg of=/dev/sdx
bs=1M status=progress

RASPIAN:

sudo dd if=e-con_Raspian_e-
CAM20_CURB_Release_<VER>.rpi-sdimg  of=/dev/sdx
bs=1M status=progress
```

Note: /dev/sdX must be replaced with the appropriate device node such as **/dev/sdb** or **/dev/sdc** of the SD card. This can be obtained by running dmesg command on your host Linux PC after the SD card reader is inserted. If you introduce wrong device node might result in data loss in the corresponding device. So, this must be done with caution.

Steps to boot the board with flashed SD image:

1. Remove and connect the SD card to the Raspberry Pi 4 board, after the completion of flashing procedure.
2. Connect the power supply of 5V 3A to the Raspberry Pi 4 development board.
3. Connect micro HDMI to HDMI cable between display and the Raspberry Pi development kit at HDMI0 connector.
4. Run the following command to setup the serial port in the host PC using any serial port software.

For example, to setup picocom.

```
sudo picocom /dev/ttyUSB0 -b 115200
```

5. Power ON the board.
6. After booting, enter the login credentials in the Raspberry Pi development kit as shown in following table.

Yocto OS login credentials:

Fields	Inputs
Username	root
Password	root

7. Run the following command to check the presence of camera video nodes.

```
ls /dev/video0
```

The presence of video0 indicates driver probed successfully.

8. Run the following command to change the settings between Asynchronous mode and synchronous mode.

List controls:

```
v4l2-ctl -d 0 --list-ctrls
```

Asynchronous mode:

```
v4l2-ctl -d 0 -c frame_sync=0
```

synchronous mode:

```
v4l2-ctl -d 0 -c frame_sync=1
```

9. Run the following command to change the exposure value

Change it manual mode:

```
v4l2-ctl -d 0 -c exposure_auto=1
```

Change the exposure value:

```
v4l2-ctl -d 0 -c exposure_time_absolute=xx
```

Table 1: Values of e-CAM20_CURB Controls

Controls	Minimum Value	Maximum Value	Default Value	Manual Control	Auto Control
Gain	1	240	1	YES	NO
Exposure time absolute	1 (100 µs)	1000 (100ms)	312 (31.2 ms)	YES	NO
Frame_sync	0	1	0	YES	NO

Note:

Exposure time should be less than trigger pulse time otherwise current frame will be dropped or corrupted.

For example:

for 30 FPS , each trigger pulse time period will be 33.3ms. In this case, exposure time should not exceed 330 value.

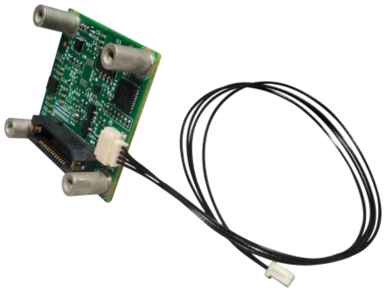


Figure 9: Trigger Pin connector

Note:

Trigger is pin no 3 of CN8. please refer to datasheet for more details

9. Run the following command to start camera streaming.

```
gst-launch-1.0 v4l2src device=/dev/video0  ! video/x-raw ,width=1920,
height=1080 ! videoconvert ! fpsdisplaysink video-
sink=autovideosink text-overlay=false sync=false -v
```

Note: The above GStreamer pipeline execution should be executed from terminal in Raspberry pi GUI, not from UART terminal (picocom or minicom) of Host PC.

10 . Run the following command to start camera recording

```
gst-launch-1.0 --gst-debug-level=3 -v v4l2src device=/dev/video0 !
capsfilter caps="video/x-raw, width=1920,height=1080,framerate=30/1" !
queue ! v4l2convert ! videorate ! queue ! v4l2h264enc  ! queue ! avimux !
filesink location = 1080p_recording.h264
```

Note:

H264 encoder of RaspberryPi is limited to 1080P@30 FPS.

Evaluation using Gst-capture Application

This section provides information of gst-capture application and procedure to use the application.

The gst-capture application is located in **/usr/bin**.

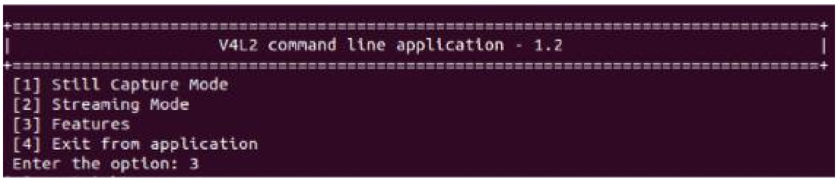
To evaluate using gst-capture application, follow these steps:

1. Run the following command to perform the application.

```
pi@raspberrypi:~$ gst-capture
```

3. Select the correct video node **/dev/video0** from options displayed in gst-capture application.

You can view the video streaming after choosing the video node. By default, the camera streams in the resolution of 1920 x 1080 and the main menu with four options appears in terminal as shown below.



```
+==================================================================================+
|                         V4L2 command line application - 1.2                     |
+==================================================================================+
 [1] Still Capture Mode
 [2] Streaming Mode
 [3] Features
 [4] Exit from application
 Enter the option: 3
```

The following table lists the main menu with the options.

Table 4: Main Menu Options

Option	Modes
1	Still Capture Mode
2	Streaming Mode
3	Features
4	Exit from application

Note: Main menu provides an option to perform the following process:

- **Still Capture Mode** - It is used to capture the still images with required available resolutions in .bmp and uyvy (Raw) format by entering the.
- **Streaming Mode** - It is used to start and stop streaming camera with required available resolutions.

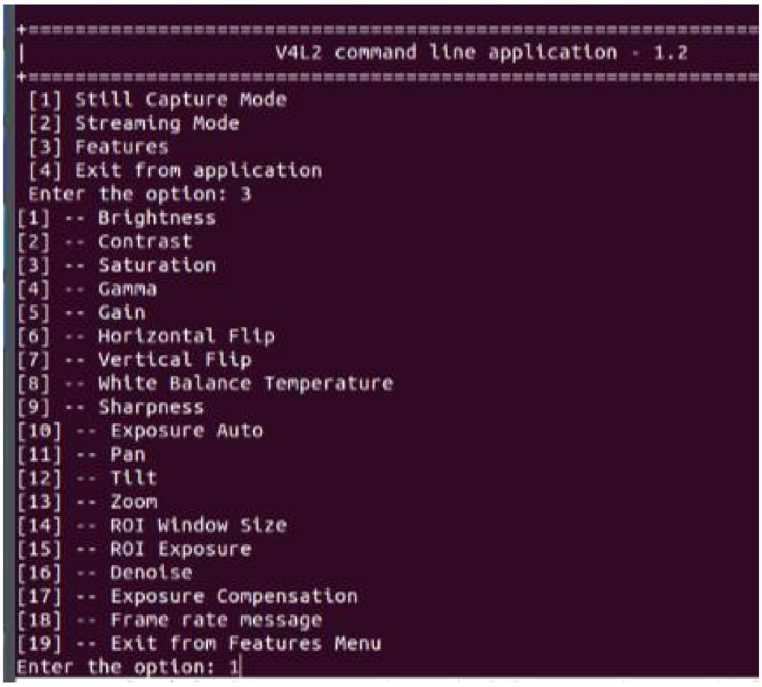
- **Features** - It contains camera controls such as brightness , contrast and so on, to be modified and evaluated during the stream.

  The following table lists the main menu with the options.

Table 10: Main Menu Options

Option	Modes
1	Still Capture Mode
2	Streaming Mode
3	Features
4	Exit from application

When you enter the option 3. The Feature Menu appears as shown below.



```
+==============================================================
|                        V4L2 command line application - 1.2
+==============================================================
 [1] Still Capture Mode
 [2] Streaming Mode
 [3] Features
 [4] Exit from application
 Enter the option: 3
[1] -- Brightness
[2] -- Contrast
[3] -- Saturation
[4] -- Gamma
[5] -- Gain
[6] -- Horizontal Flip
[7] -- Vertical Flip
[8] -- White Balance Temperature
[9] -- Sharpness
[10] -- Exposure Auto
[11] -- Pan
[12] -- Tilt
[13] -- Zoom
[14] -- ROI Window Size
[15] -- ROI Exposure
[16] -- Denoise
[17] -- Exposure Compensation
[18] -- Frame rate message
[19] -- Exit from Features Menu
Enter the option: 1
```

On selection of any of the above listed features, you can view the current values and option to select the level of the features or to exit from the menu.

- **Exiting from application** - It is used to close the application.

Reference Documents

This section describes the software and hardware documents of e-CAM20_CURB.

Software Documents

The software documents and its description are shown in below table.

S.NO	What I need	Documents to Refer
1	Build yocto setup	e-CAM20_CURB_Developer_Guide_<VER>.pdf
2	Use GStreamer to control the e-CAM20_CURB camera on the Raspberry Pi 4.	e-CAM20_CURB_GStreamer_Usage_Guide_<VER>.pdf

Table 2: Description of Software Documents

Hardware Documents

The hardware documents and its description are shown in below table.

S.NO	Documents Name	Description
1	e-CAM20_CURB_Datasheet_<ver>.pdf	Describes the features, connector pin-out details and mechanical dimensions of e-CAM20_CURB.
2	e-CAM20_CURB_Lens_Datasheet_<ver>.pdf	Describes the optical specification of lenses used in e-CAM20_CURB.

Table 3: Description of Hardware Documents

Troubleshooting

In this section, you can view the commonly occurring issue and their troubleshooting step.

What can I do if frame rate is fluctuating in gst-capture application?

This issue occurs when the camera is in auto exposure mode as the camera changes exposure based on lighting conditions. If a fixed fps is required, change the camera to manual exposure mode.

Why video tearing observed while moving camera very fast?

HDMI signal strength and modes should be adjusted in config.txt properly according to your HDMI display interfaced. It is a known issue with Raspberry Pi which were discussed in the Pi forums. For confirmation, please try running video tearing test videos in your board.

FAQ

1. **Do e-con Systems have any plan to support longer length cable?**

   e-con Systems provide a very flexible and robust 15 cm coaxial cable along with this kit. For customization, please write to camerasolutions@e-consystems.com with your requirement.

2. **Why camera is not detected?**

   Please cross check whether the FPC cable is connected properly. Please connect blue color sides of the cable shown in the
   Figure 8: Camera Connection with Development Kit.

What's Next?

After understanding the specifications of camera daughter board and instructions on how to use this daughter board with Raspberry Pi board, you can refer to the following documents to understand more about e-CAM20_CURB.

- *e-CAM20_CURB Developer Guide*

Glossary

CMOS: Complementary Metal Oxide Semiconductor.

CSI: Camera Serial Interface.

FHD: Full HD (Industry name for 1920 x 1080P resolution).

HD: High Definition (Industry name for 1280 x 720 resolution).

HDMI: High-definition Multimedia Interface.

ISP: Image Signal Processor.

MIPI: Mobile Industry Processor Interface.

UART: Universal Asynchronous Receiver/Transmitter.

USB: Universal Serial Bus.

Support

Contact Us

If you need any support on e-CAM20_CURB product, please contact us using the Live Chat option available on our website - https://www.e-consystems.com/

Creating a Ticket

If you need to create a ticket for any type of issue, please visit the ticketing page on our website - https://www.e-consystems.com/create-ticket.asp

RMA

To know about our Return Material Authorization (RMA) policy, please visit the RMA Policy page on our website - https://www.e-consystems.com/RMA-Policy.asp

General Product Warranty Terms

To know about our General Product Warranty Terms, please visit the General Warranty Terms page on our website - https://www.e-consystems.com/warranty.asp

Revision History

Rev	Date	Description	Author
1.0	15-APR-2021	Initial draft	Product marketing team
1.1	16-MAY-2021	Debug prints removed from driver	Product marketing team
1.2	19-June-2021	Frame sync support added	Product marketing team
1.3	10-June-2021	Trigger pin details modified	Product marketing team
1.4	30-Nov-2021	New adapter with CN8 details added	Product marketing team