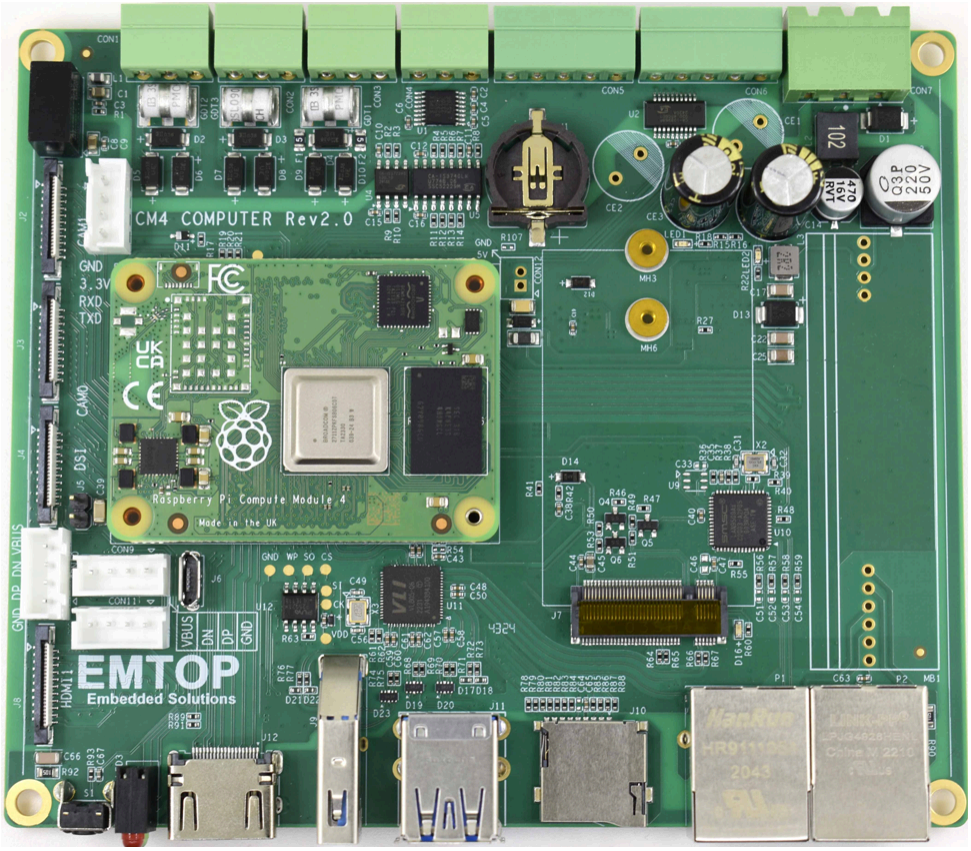
CM4-ET-IND
Linux User Manual



Version: 0.1
2025-04-30

www.emtop-tech.com	github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

Revision History

Version	Date	Description
0.1	2025-04-30	Initial Release

Contents

1. Product Overview

1.1 Introduction

1.2 Resource

2. Linux Operation System

2.1 Make A Bootable TF Card

1. Unxz image file ***CM4-ET-IND-SD-REVXX.img.xz*** with WinRAR, to generate .img file;
2. Write image file ***CM4-ET-IND-SD-REVXX.img*** into SD card with Win32DiskImager;
3. Install the SD card on ARM board;
4. Connect the UART debug port to PC. Open the serial port [115200, 8N1] with terminal tool such as PuTTY, SecureCRT etc.
5. Connect the power cable and power up the board [12V/2A];
6. After a while, about 1 ~ 2 seconds, the terminal window will show the booting message as below:

```
Debian GNU/Linux 12 raspberrypi ttyS0

My IP address is 192.168.3.44 fe80::9c42:d2bc:94b8:ced1

raspberrypi login:
```

Note

- Default Login Account: ***pi*** with password ***raspberry***
- The screen [HDMI or DSI] will show "Welcome to Raspberry Pi" dialog after the first boot. It will do some basic settings, such as language, keyboard and create user account.

2.2 Test and Demonstration

This section will run some tests on the peripheral devices.

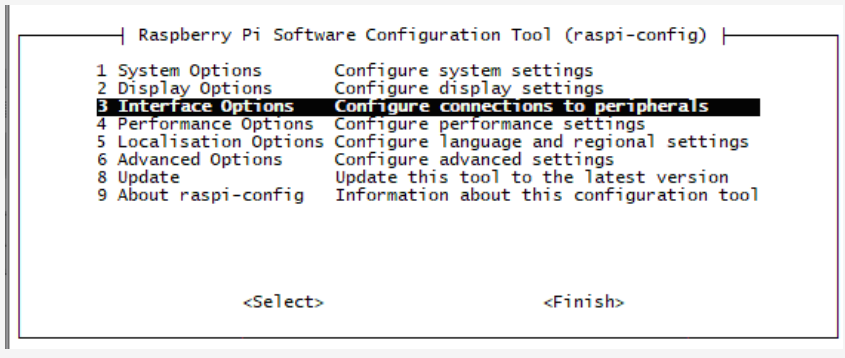
POWER: **DC 12V-3A**

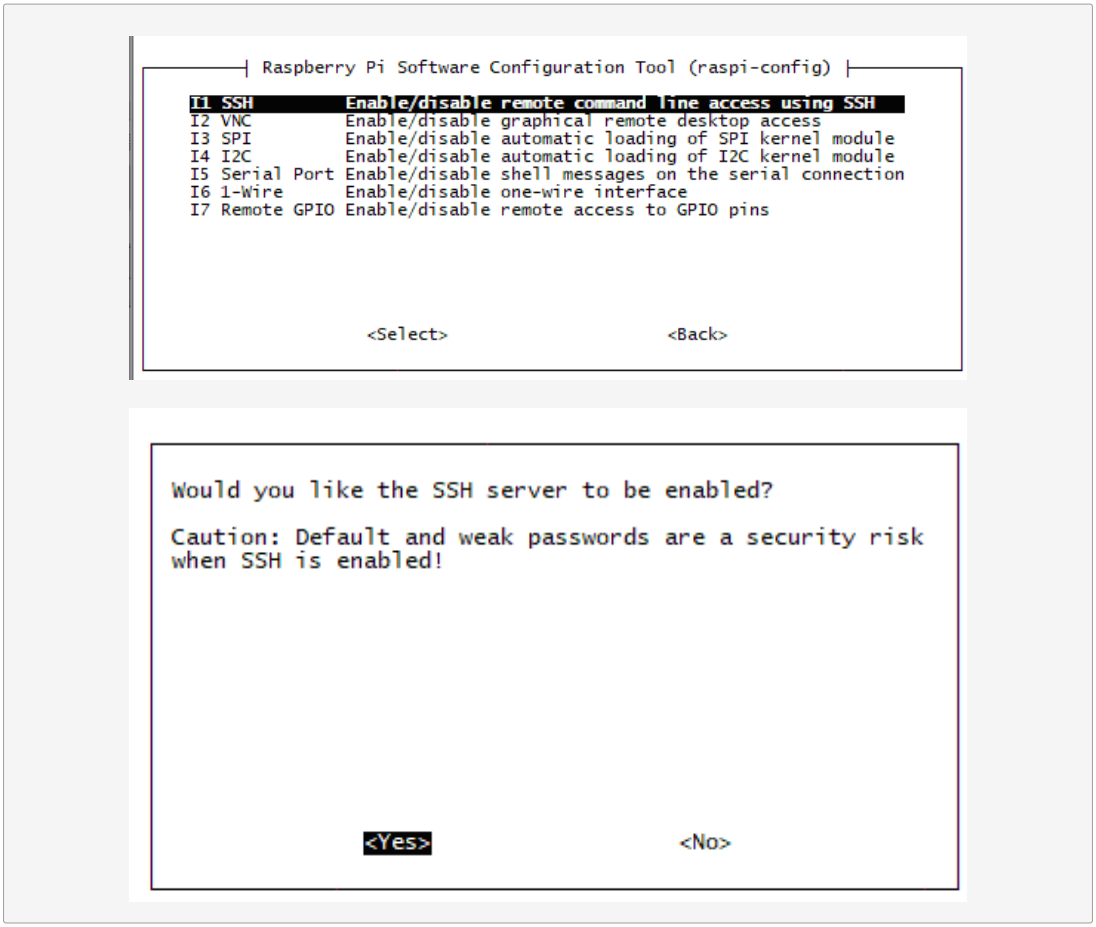
Debug Port: **UART0, 115200 1N8, CON8**

2.2.1 SSH LOGIN

Enable SSH server:

```
root@arm:~# raspi-config
```





2.2.2 RTC

There is a RTC chip PCF8563T on the base board.

```
root@arm:~# cat /sys/class/rtc/rtc0/name
rtc-pcf8563 22-0051
```

Let's set the current time to 2024-07-05 10:12:

```
root@arm:~# date -s "2024-02-07 10:12"; hwclock -f /dev/rtc0 -w
```

Reboot the board, and check the hardware RTC time with below command:

```
root@arm:~# hwclock -f /dev/rtc0
2024-02-05 10:12:07.365014+00:00
```

Note

- If the board is connected to Internet, it will automatically synchronize to standard NTP time, the command 'date -s' will not take effect.
- It the current time zone is not correct, please refer to the below chapter to set time zone.

2.2.3 TIMEZONE SETTING

Let's set Beijing Time as an example:

```
root@arm:~# echo "Asia/Shanghai" > /etc/timezone
```

```
root@arm:~# ln -sf /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
```

```
root@arm:~# sync
```

2.2.4 USB HOST

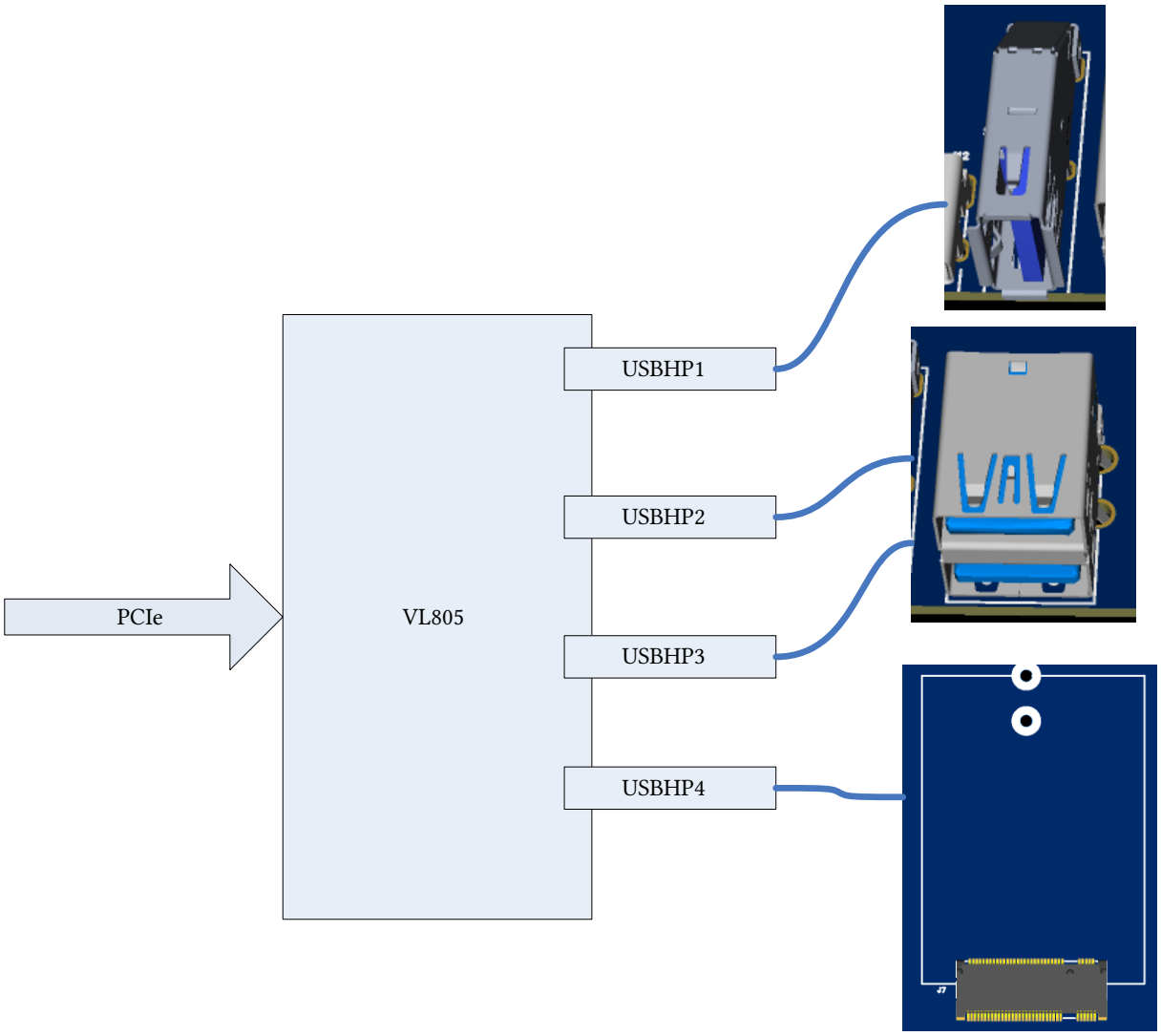


Figure 2-4: PCIe to USB3.0

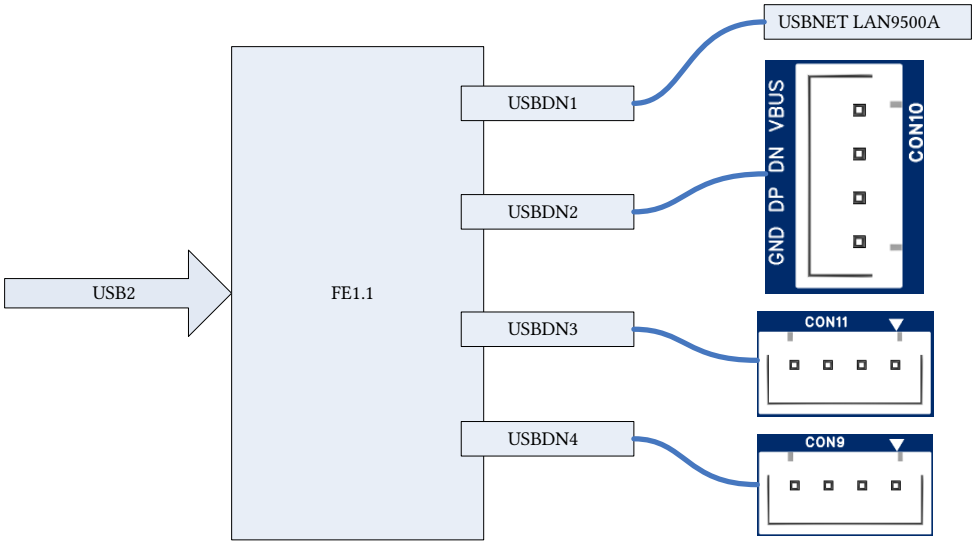


Figure 2-5: USB2.0 HUB

```
root@arm:~# lspci

00:00.0 PCI bridge: Broadcom Inc. and subsidiaries BCM2711 PCIe Bridge (rev 20)
01:00.0 USB controller: VIA Technologies, Inc. VL805/806 xHCI USB 3.0 Controller (rev 01)
```

```
root@arm:~# lsusb

Bus 003 Device 003: ID 0424:9e00 Microchip Technology, Inc. (formerly SMSC) LAN9500A/LAN9500Ai
Bus 003 Device 002: ID 1a40:0101 Terminus Technology Inc. Hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

2.2.5 NETWORK

There are two network chips on board. One of them is an USB net LAN9500A.

INTERFACE	DISCRIPTION	SPEED
eth0	Core Board	1Gbps
eth1	LAN9500A, USB Extend	100Mbps

```
root@arm:~# ifconfig eth0

eth0      Link encap:Ethernet  HWaddr 3a:f7:82:bc:fa:0a
          inet addr:192.168.1.81  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::38f7:82ff:febc:fa0a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:11 errors:0 dropped:4 overruns:0 frame:0
          TX packets:42 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1555 (1.5 KiB)  TX bytes:7192 (7.0 KiB)
```

DHCP feature is enabled as default; the board can request a valid IP address from DHCP server in local network. Also, you can try the below command to force to request IP address:

```
root@arm:~# ping -I eth0 www.baidu.com
```

```
PING www.a.shifen.com (14.215.177.38) from 192.168.1.81 eth0: 56(84) bytes of data.
64 bytes from www.baidu.com (183.232.231.174): icmp_seq=1 ttl=56 time=12.1 ms
64 bytes from www.baidu.com (183.232.231.174): icmp_seq=2 ttl=56 time=12.2 ms
64 bytes from www.baidu.com (183.232.231.174): icmp_seq=3 ttl=56 time=12.1 ms
64 bytes from www.baidu.com (183.232.231.174): icmp_seq=4 ttl=56 time=12.5 ms
^C
--- www.a.shifen.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 7.058/7.447/7.771/0.319 ms
```

2.2.6 HDMI

There a two HDMI interfaces on board: a standard HDMI type A slot J12 and a FPC slot J8.

2.2.7 MIPI-DSI

Devices already tested:

MODULE	DISCRIPTION
WAVESHARE 7 INCH DSI LCD	800*480, with capacitive touch panel
RASPBERRYPI DISPLAY	800*480, with capacitive touch panel

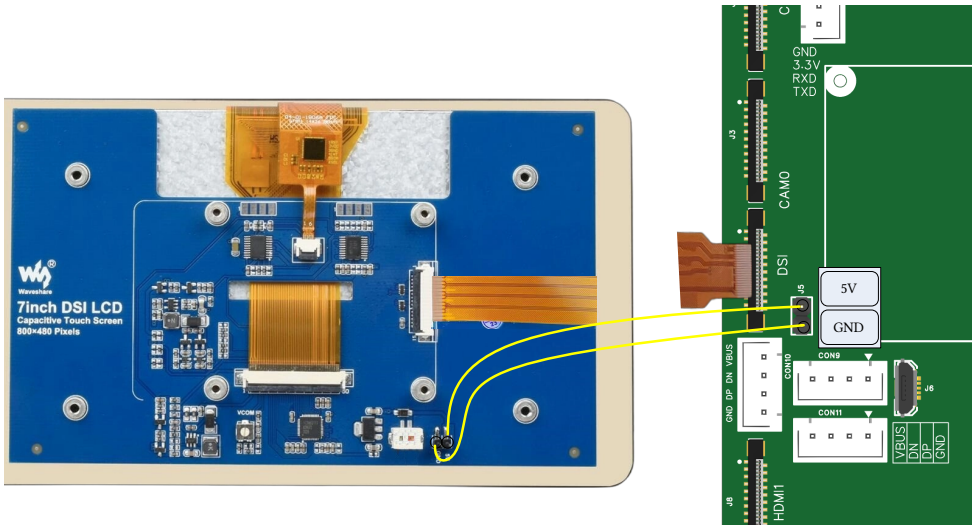


Figure 2-6: WaveShare 7 inch DSI LCD

2.2.8 TOUCH PANEL

The DSI display modules are already equipped touch panel.

root@arm:~# apt-get install -y evtest

root@arm:~# evtest

```
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      0-0038 generic ft5x06 (79)
/dev/input/event1:      vc4-hdmi-0
/dev/input/event2:      vc4-hdmi-0 HDMI Jack
```

```
/dev/input/event3:      vc4-hdmi-1
/dev/input/event4:      vc4-hdmi-1 HDMI Jack
Select the device event number [0-4]:
```

2.2.9 CAMERA

Devices already tested:

MODULE	DISCRIPTION
Raspberry PI Camera V2	IMX219 inside, 8 MegaPixels

There are two CSI channels on the base board: CAM0 and CAM1.

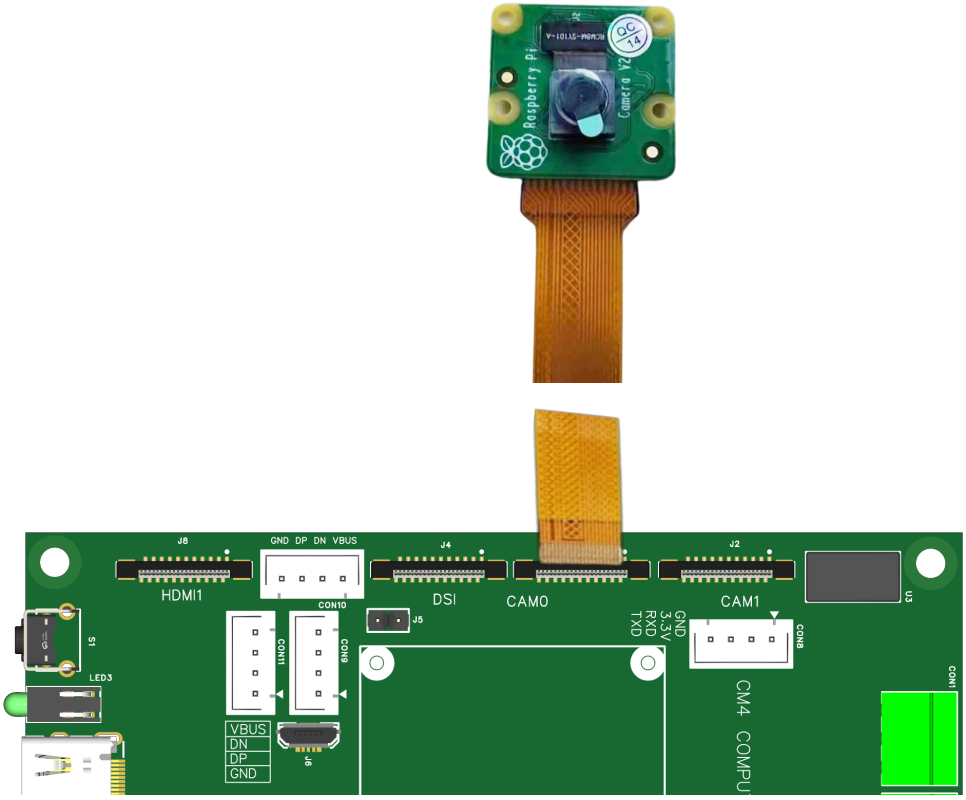


Figure 2-7: Raspberry PI Camera V2

When we connect two CSI cameras, the ***--list-cameras*** command will find 2 devices:

root@arm:~# rpicam-hello --list-cameras

```
----------------
0 : imx219 [3280x2464 10-bit RGGB] (/base/soc/i2c0mux/i2c@1/imx219@10)
    Modes: 'SRGGB10_CSI2P' : 640x480 [206.65 fps - (1000, 752)/1280x960 crop]
                             1640x1232 [41.85 fps - (0, 0)/3280x2464 crop]
                             1920x1080 [47.57 fps - (680, 692)/1920x1080 crop]
                             3280x2464 [21.19 fps - (0, 0)/3280x2464 crop]
           'SRGGB8' : 640x480 [206.65 fps - (1000, 752)/1280x960 crop]
                      1640x1232 [83.70 fps - (0, 0)/3280x2464 crop]
                      1920x1080 [47.57 fps - (680, 692)/1920x1080 crop]
                      3280x2464 [21.19 fps - (0, 0)/3280x2464 crop]

1 : imx219 [3280x2464 10-bit RGGB] (/base/soc/i2c0mux/i2c@0/imx219@10)
```

```
    Modes: 'SRGGB10_CSI2P' : 640x480 [206.65 fps - (1000, 752)/1280x960 crop]
                             1640x1232 [41.85 fps - (0, 0)/3280x2464 crop]
                             1920x1080 [47.57 fps - (680, 692)/1920x1080 crop]
                             3280x2464 [21.19 fps - (0, 0)/3280x2464 crop]
           'SRGGB8' : 640x480 [206.65 fps - (1000, 752)/1280x960 crop]
                      1640x1232 [83.70 fps - (0, 0)/3280x2464 crop]
                      1920x1080 [47.57 fps - (680, 692)/1920x1080 crop]
                      3280x2464 [21.19 fps - (0, 0)/3280x2464 crop]
```

Open a terminal window on Pi desktop:

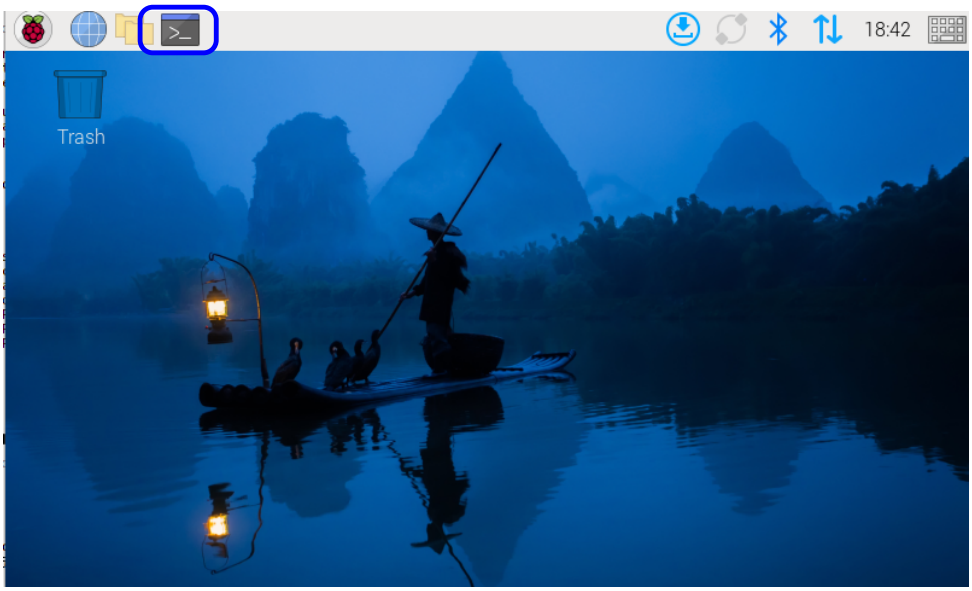


Figure 2-8: Open Terminal Window

Run below command to capture stream and display on screen:

```
root@arm:~# rpicam-hello --camera 0

[0:13:06.678417548] [2414]  INFO Camera camera_manager.cpp:325 libcamera
v0.3.2+27-7330f29b
[0:13:06.729028715] [2417]  WARN RPiSdn sdn.cpp:40 Using legacy SDN tuning - please
consider moving SDN inside rpi.denoise
[0:13:06.730857326] [2417]  WARN RPI vc4.cpp:393 Mismatch between Unicam and CamHelper for
embedded data usage!
[0:13:06.731488993] [2417]  INFO RPI vc4.cpp:447 Registered camera /base/soc/i2c0mux/
i2c@0/imx219@10 to Unicam device /dev/media4 and ISP device /dev/media0
[0:13:06.731531808] [2417]  INFO RPI pipeline_base.cpp:1126 Using configuration file '/
usr/share/libcamera/pipeline/rpi/vc4/rpi_apps.yaml'
[0:13:06.743154030] [2417]  WARN RPiSdn sdn.cpp:40 Using legacy SDN tuning - please
consider moving SDN inside rpi.denoise
[0:13:06.745071548] [2417]  WARN RPI vc4.cpp:393 Mismatch between Unicam and CamHelper for
embedded data usage!
[0:13:06.745700271] [2417]  INFO RPI vc4.cpp:447 Registered camera /base/soc/i2c0mux/
i2c@1/imx219@10 to Unicam device /dev/media5 and ISP device /dev/media2
[0:13:06.745743548] [2417]  INFO RPI pipeline_base.cpp:1126 Using configuration file '/
usr/share/libcamera/pipeline/rpi/vc4/rpi_apps.yaml'
Made DRM preview window
Mode selection for 1640:1232:12:P
    SRGGB10_CSI2P,640x480/0 - Score: 4504.81
    SRGGB10_CSI2P,1640x1232/0 - Score: 1000
    SRGGB10_CSI2P,1920x1080/0 - Score: 1541.48
    SRGGB10_CSI2P,3280x2464/0 - Score: 1718
```

```
    SRGGB8,640x480/0 - Score: 5504.81
    SRGGB8,1640x1232/0 - Score: 2000
    SRGGB8,1920x1080/0 - Score: 2541.48
    SRGGB8,3280x2464/0 - Score: 2718
[0:13:06.748326567] [2414]  INFO Camera camera.cpp:1197 configuring streams: (0)
1640x1232-YUV420 (1) 1640x1232-SBGGR10_CSI2P
[0:13:06.748729308] [2417]  INFO RPI vc4.cpp:622 Sensor: /base/soc/i2c0mux/i2c@1/imx219@10
- Selected sensor format: 1640x1232-SBGGR10_1X10 - Selected unicam format: 1640x1232-pBAA
```

Pass the parameter ***--camera 1*** to capture image from the other camera.

2.2.10 UART

DEVICE NODE	HARDWARE	USAGE
/dev/ttyS0	UART0	DEBUG PORT
/dev/ttyAMA3	UART3	3-WIRE
/dev/ttyAMA4	UART4	RS485
/dev/ttyAMA5	UART5	RS485

Let us test the orinary UART port UART3. Connect its TXD and RXD pins, and run command:

```
root@arm:~# /test/app/com -d /dev/ttyAMA3

SEND: 1234567890
RECV: 1234567890
SEND: 1234567890
RECV: 1234567890
```

The default baud rate is 115200. If you want to assign another specific baud rate:

```
root@arm:~# /test/app/com -d /dev/ttyAMA3 -b 9600
```

Please refer to the source code ***com.tar.xz*** for all supported baud rates.

2.2.11 RS485

DEVICE NODE	HARDWARE	USAGE	REMARK
/dev/ttyAMA4	UART4	RS485	without RTS
/dev/ttyAMA5	UART5	RS485	without RTS

There are 2 RS485 buses on board, let us connect them together.

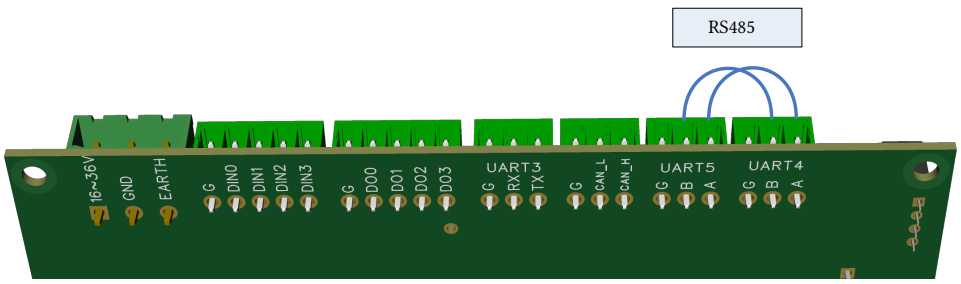


Figure 2-9: Connect RS485 Bus

```
root@arm:~# /test/app/com -d /dev/ttyAMA4 -s "Hello world" &
```

```
root@arm:~# /test/app/com -d /dev/ttyAMA5

SEND: 1234567890
SEND: Hello world
RECV: 1234567890
RECV: Hello world
SEND: 1234567890
SEND: Hello world
RECV: 1234567890
RECV: Hello world
```

2.2.12 CAN BUS

There is one CAN bus on board. Connect the CAN bus with 2 boards:

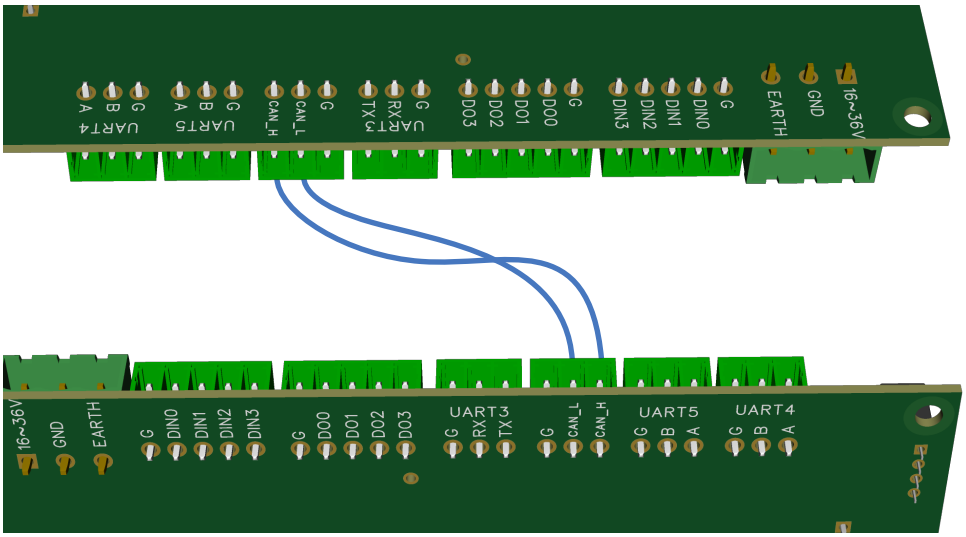


Figure 2-10: Connect CAN Bus

Install necessary package:

```
root@arm:~# apt-get install -y can-utils
```

```
root@arm:~# ifconfig can0

can0: flags=128<NOARP>  mtu 16
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 10  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Configure parameters [both sending and receiving side]:

```
root@arm:~# ifconfig can0 down
```

```
root@arm:~# ip link set can0 type can bitrate 125000
```

```
root@arm:~# ip link set can0 type can restart-ms 100
```

```
root@arm:~# ifconfig can0 up
```

Start to listen on one board:

```
root@arm:~# candump can0 &
```

Send package on the other board:

```
root@arm:~# cansend can0 "5A1#1122334455667788"
```

Then the receiving side will report message as below:

```
can0  5A1   [8] 11 22 33 44 55 66 77 88
```

For more information, please refer to project can-utils.

2.2.13 SPIFLASH

A SPIFlash XT25F64BSOIGT is equipped on baseboard.

```
root@arm:~# dmesg | grep -i spi-nor
[    9.019647] spi-nor spi1.1: spi-nor-generic (8192 Kbytes)
```

```
root@arm:~# cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00800000 00001000 "spi1.1"
```

```
root@arm:~# apt-get install -y mtd-utils
```

```
root@arm:~# modprobe mtdblock
```

Erase and format:

```
root@arm:~# flash_erase /dev/mtd0 0 0

Erasing 8192 Kibyte @ 0 -- 100 % complete
```

```
root@arm:~# mount -t jffs2 /dev/mtdblock0 /mnt
```

Write and read under directory ***/mnt***, the content will be kept in the SPIFlash memory.

```
root@arm:~# umount /mnt
```

Next boot, mount the flash and you can see the contents written before.

2.2.14 DI/DO

DI/DO

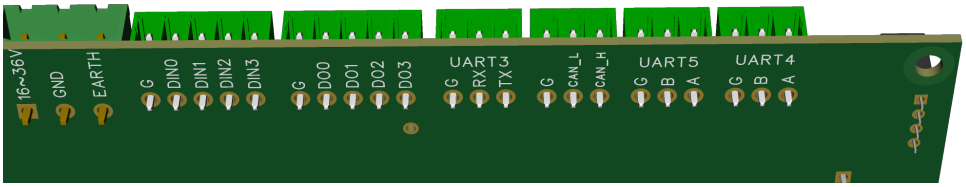


Figure 2-11: DI/DO

DI/DO	GPIO	IO_BANK	IO_OFFSET
DIN0	GPIO2	0	2
DIN1	GPIO3	0	3
DIN2	GPIO6	0	6
DIN3	GPIO7	0	7
DOUT0	GPIO22	0	22
DOUT1	GPIO23	0	23
DOUT2	GPIO24	0	24
DOUT3	GPIO25	0	25

Set DOUT0 output low:

```
root@arm:~# gpioset 0 22=0
```

Set DOUT0 output heigh:

```
root@arm:~# gpioset 0 22=1
```

Note

- **gpioset <io_bank> <io_offset>=<output value>**

Read DIN0 input:

```
root@arm:~# gpioget 0 2

1 or 0
```

Note

- **gpioget <io_bank> <io_offset>**

Or monitor the IO status changing event:

```
root@arm:~# gpiomon 0 2

event:  RISING EDGE offset: 5 timestamp: [    1151.814356387]
event:  FALLING EDGE offset: 5 timestamp: [    1151.815449803]
event:  RISING EDGE offset: 5 timestamp: [    1152.091556803]
```

2.2.15 PCIe GSM

Devices already tested:

MODULE	DISCRIPTION
QUECTEL EM05-CE	4G module
QUECTEL RM500Q-GL	5G module

Figure 2-12: 4G/5G Module

Install QUECTEL GSM module, SIM card and antenna.

Enable GSM power supply:

```
root@arm:~# gpioset 0 17=1

[Wait about 10 seconds...]
[  696.459095] option 3-1.1:1.0: GSM modem (1-port) converter detected
```

```
[  696.465847] usb 3-1.1: GSM modem (1-port) converter now attached to ttyUSB0
[  696.473511] option 3-1.1:1.1: GSM modem (1-port) converter detected
[  696.480292] usb 3-1.1: GSM modem (1-port) converter now attached to ttyUSB1
[  696.487876] option 3-1.1:1.2: GSM modem (1-port) converter detected
[  696.494574] usb 3-1.1: GSM modem (1-port) converter now attached to ttyUSB2
[  696.502194] option 3-1.1:1.3: GSM modem (1-port) converter detected
[  696.508949] usb 3-1.1: GSM modem (1-port) converter now attached to ttyUSB3
```

Terminate ***pppd*** program which may be running background:

root@arm:~# killall -q pppd && sleep 3

root@arm:~# pppd call quectel-ppp &

```
......
Script /usr/local/sbin/chat -E -s -v -f /etc/ppp/peers/quectel-chat-connect finished (pid
891), status = 0x0
Serial connection established.
using channel 6
Using interface ppp0
Connect: ppp0 <--> /dev/ttyGSM03
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0x99ca38bd> <pcomp> <accomp>]
rcvd [LCP ConfReq id=0xa <asyncmap 0x0> <auth chap MD5> <magic 0x8fb21dd6> <pcomp> <accomp>]
sent [LCP ConfAck id=0xa <asyncmap 0x0> <auth chap MD5> <magic 0x8fb21dd6> <pcomp> <accomp>]
rcvd [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0x99ca38bd> <pcomp> <accomp>]
sent [LCP EchoReq id=0x0 magic=0x99ca38bd]
rcvd [LCP DiscReq id=0xb magic=0x8fb21dd6]
rcvd [CHAP Challenge id=0x1 <ede1a1633678b8a18ed16d5f1891b8cf>, name = "UMTS_CHAP_SRVR"]
sent [CHAP Response id=0x1 <68c3d55a12080e299e8b3751431746cf>, name = "$LTE_USERNAME"]
rcvd [LCP EchoRep id=0x0 magic=0x8fb21dd6 99 ca 38 bd]
rcvd [CHAP Success id=0x1 ""]
CHAP authentication succeeded
CHAP authentication succeeded
sent [IPCP ConfReq id=0x1 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
sent [IPV6CP ConfReq id=0x1 <addr fe80::a062:33a3:7882:408f>]
rcvd [IPCP ConfReq id=0x8]
sent [IPCP ConfNak id=0x8 <addr 0.0.0.0>]
rcvd  [IPCP  ConfNak  id=0x1  <addr  10.33.200.184>  <ms-dns1  202.96.128.86>  <ms-dns2
202.96.134.133>]
sent  [IPCP  ConfReq  id=0x2  <addr  10.33.200.184>  <ms-dns1  202.96.128.86>  <ms-dns2
202.96.134.133>]
rcvd [IPCP ConfReq id=0x9]
sent [IPCP ConfAck id=0x9]
rcvd  [IPCP  ConfAck  id=0x2  <addr  10.33.200.184>  <ms-dns1  202.96.128.86>  <ms-dns2
202.96.134.133>]
Could not determine remote IP address: defaulting to 10.64.64.64
local  IP address 10.33.200.184
remote IP address 10.64.64.64
primary   DNS address 202.96.128.86
secondary DNS address 202.96.134.133
Script /etc/ppp/ip-up started (pid 900)
Script /etc/ppp/ip-up finished (pid 900), status = 0x0
```

Note

- If ***pppd*** command reports error, please try to run it again.

Configure default gateway:

```
root@arm:~# route del default; route add default ppp0
```

Configure ***resolv.conf***:

```
root@arm:~# cat /etc/ppp/resolv.conf > /etc/resolv.conf
```

Note

- The ***resolv.conf*** is very important. If it's not correct, the ping command will report error like this: Temporary failure in name resolution.

Connection test:

```
root@arm:~# ping -I ppp0 www.baidu.com

PING www.a.shifen.com (14.215.177.38) from 10.32.232.200 ppp0: 56(84) bytes of data.
64 bytes from 14.215.177.38: icmp_seq=1 ttl=54 time=37.0 ms
64 bytes from 14.215.177.38: icmp_seq=2 ttl=54 time=43.5 ms
64 bytes from 14.215.177.38: icmp_seq=3 ttl=54 time=51.8 ms
64 bytes from 14.215.177.38: icmp_seq=4 ttl=54 time=41.4 ms
^C64 bytes from 14.215.177.38: icmp_seq=5 ttl=54 time=33.4 ms

--- www.a.shifen.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 20329ms
rtt min/avg/max/mdev = 33.408/41.456/51.856/6.272 ms
```

2.2.16 WIFI [CORE BOARD]

Scan remote WiFi access points:

```
root@arm:~# nmcli dev wifi

IN-USE  BSSID              SSID               MODE   CHAN  RATE        SIGNAL  >
        F4:91:1E:2C:04:D0  1e2c04d0           Infra  1     65 Mbit/s   54
        DC:73:85:76:53:6C  EMTOP       Infra  11    130 Mbit/s  46
        ......
```

Connect:

```
root@arm:~# nmcli dev wifi connect EMTOP password 12345678

Device 'wlan0' successfully activated with '58b5b546-62cd-4d66-b23b-d84b0dfec5f6'.
```

Check:

```
root@arm:~# ifconfig wlan0
```

```
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.202.99  netmask 255.255.255.0  broadcast 192.168.202.255
        inet6 fe80::fdef:b83a:7345:910d  prefixlen 64  scopeid 0x20<link>
        ether e4:5f:01:cc:40:13  txqueuelen 1000  (Ethernet)
        RX packets 149  bytes 10378 (10.1 KiB)
        RX errors 0  dropped 115  overruns 0  frame 0
        TX packets 134  bytes 21943 (21.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Transmission test:

root@arm:~# ping -I wlan0 www.baidu.com

```
PING www.baidu.com(2409:8c54:870:310:0:ff:b0ed:40ac) from
2409:895a:94b3:457f:7d0d:a2a8:31db:8548 wlan0: 56 data bytes
64 bytes from 2409:8c54:870:310:0:ff:b0ed:40ac: icmp_seq=1 ttl=51 time=42.0 ms
64 bytes from 2409:8c54:870:310:0:ff:b0ed:40ac: icmp_seq=2 ttl=51 time=28.8 ms
......
```

2.2.17 BLUETOOTH [CORE BOARD]

root@arm:~# bluetoothctl

```
Agent registered
[bluetooth]# power on
Changing power on succeeded
[bluetooth]# scan on
Discovery started
[CHG] Controller D0:C5:D3:F9:60:06 Discovering: yes
[NEW] Device 78:C5:28:67:88:03 78-C5-28-67-88-03
[NEW] Device 7B:A2:1E:1D:15:60 7B-A2-1E-1D-15-60
.. ..
[bluetooth]# scan off
```

Please search ***bluetoothctl*** usage on web for more information.

Note

- It's recommended to operate bluetooth with desktop GUI.

3. Modify Official Image

Base Image: ***2024-11-19-raspios-bookworm-arm64.img.xz***.

- Append the below contents marked blue to ***/boot/firmware/config.txt***.

/boot/firmware/config.txt

```
......
[all]
enable_uart=1
dtoverlay=emtop-cm4-et-ind
dtoverlay=vc4-kms-dsi-7inch
dtoverlay=imx219,cam0
dtoverlay=imx219,cam1
```

- Add board support overlay dtbo:

/boot/firmware/overlays/emtop-cm4-et-ind.dtbo

- Set the login password for account 'pi' to 'raspberry'. If not modify ***/etc/shadow***, user should connect HDMI and USB keyboard, create account during the first boot.

/etc/shadow

```
pi:$y$j9T$btMnr.S.VCNCEzqBP5VX5/
$0YK9j12kssKmcPFVjFsSXhqbOmoeD7z.nTYU3AFJht3:20046:0:99999:7:::
```

- Add udev rule for USB GSM module:

/etc/udev/rules.d/90-gsm-modem.rules

- Add ppp scripts for GSM module:

/etc/ppp/peers

- Files provided by EMTOP Tech.:

/test