

# **ALINX FPGA Development Board**

VCU H264/H265 Codec



## Version Record

Version	Date	Release By	Description
Rev1.0.1	2021-04-18	Rachel Zhou	First Release

We promise that this tutorial is not a permanent, consistent document. We will continue to revise and optimize the tutorial based on the feedback of the forum and the actual development experience.

## Content

Version Record.....	2
Content.....	3
Part 1: Environment Introduction.....	4
Part 1.1: VCU Introduction.....	4
Part 2: Preparation for program operation.....	4
Part 2.1: Make sd Card.....	4
Part 2.2: Connect dp monitor.....	8
Part 2.3: Open the serial terminal to connect to the debug port of the board.....	8
Part 2.4: Log in to the system.....	8
Part 3: Video codec.....	8
Part 3.1: gst command decoding example.....	8
Part 3.2: gst command coding example.....	8

# Part 1: Environment Introduction

## Part 1.1: VCU Introduction

VCU, namely **Video Codec Unit**. This is a **H264/H265** video codec processing module. Maximum support **3840x2160@60fps** video encoding and decoding at the same time.

This module is generally integrated in the EV family of UltraScale+MPSOC. Located on the PL side, it needs to interact with the PS through the axi interface.

# Part 2: Preparation for program operation

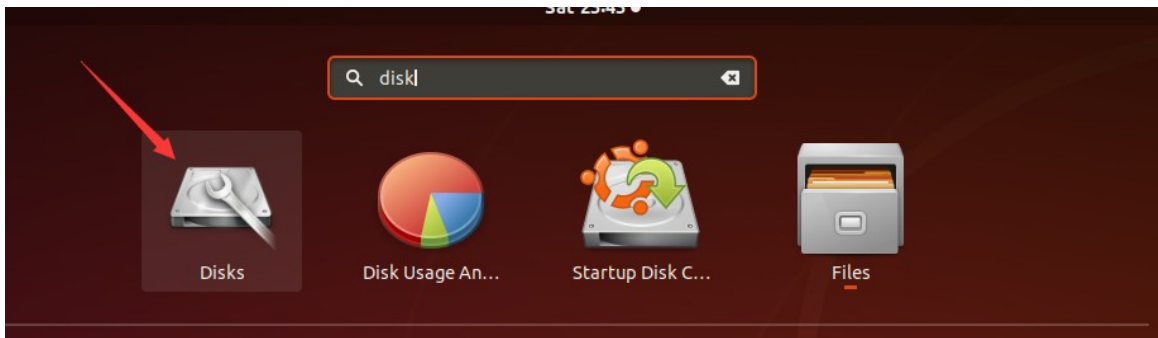
## Part 2.1: Make sd Card

1) Connect the sd card to the ubuntu virtual machine



2) Open Disk tool

In **ubuntu**, press the **win** key, in the input box that appears, enter **disk** and click the icon pointed by the arrow below

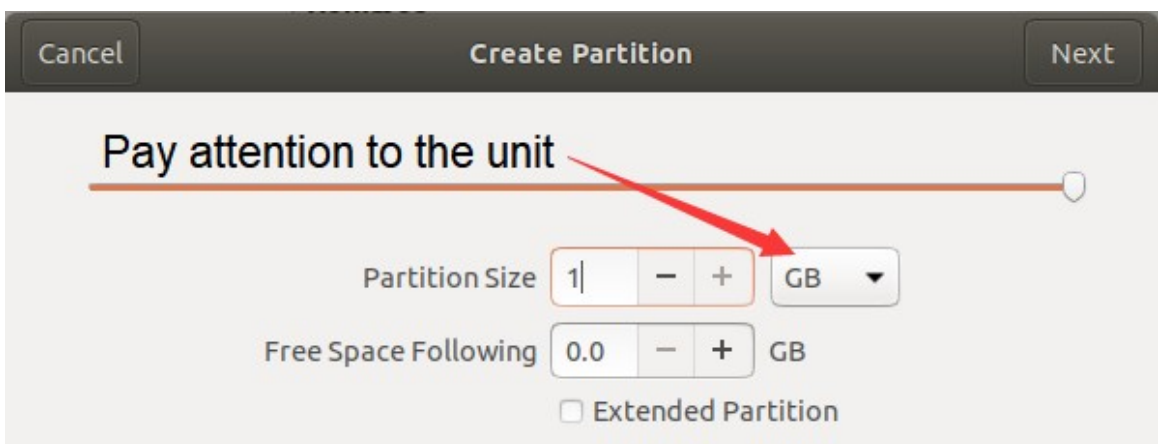


- 3) After selecting the following U disk, if there is already a partition, you can click the "-" button in the figure below to delete the original partition



- 4) Create partition

Click the "+" button in the picture to add a 1G partition



Click the **Next** button, in the following interface, set the partition name to **BOOT**, the default is **FAT** partition

Previous **Format Volume** Create

Volume Name   
For example: "Angela's Files" or "Backup".

Erase ☐ OFF  
Overwrites existing data, but takes longer.

Type ☐ Internal disk for use with Linux systems only (Ext4)  
☐ Password protect volume (LUKS)  
☐ For use with Windows (NTFS)  
☒ For use with all systems and devices (FAT)  
☐ Other

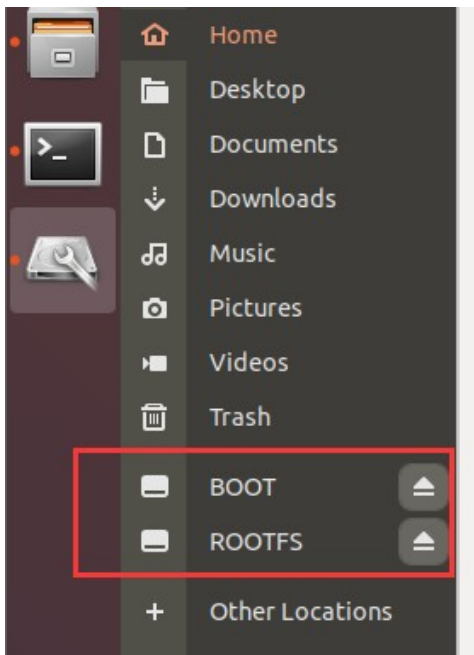
Click Create and wait for the initialization to complete

5) Create the second root file system partition

Similar to the above creation process, create a partition named ROOTFS with the format of Ext4 from the remaining U disk space

6) Mount the newly created file system

After the above partition is created, open a folder and on the sidebar, you can see the newly created partition



Click the two drive letters with the mouse, and the system will automatically mount the two partitions

7) Copy BOOT file

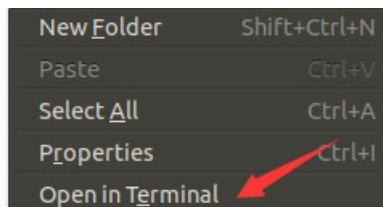
Copy the contents of the [run\\_file/BOOT](#) folder into the [BOOT](#) directory of the [SD](#) card

8) Unzip the file system to ROOTFS

Click on [ROOTFS](#) to enter the [ROOTFS](#) folder

Copy [run\\_file/rootfs.tar.gz](#) to this directory

Right-click and select the following items in the pop-up menu to open a terminal

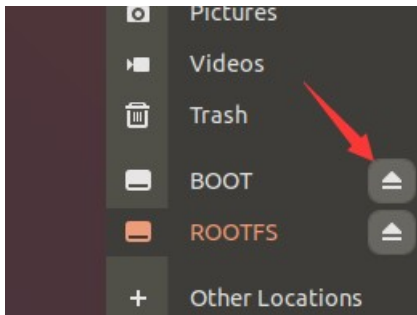


Enter the following command in the terminal to decompress the file

```
tar -zxvf rootfs.tar.gz
```

9) Wait for the SD card writing to complete

After the above command is executed, close the terminal. Click the button as shown in the figure below, when these two drive letters disappear, you can dial out the U disk



## **Part 2.2: Connect dp monitor**

## **Part 2.3: Open the serial terminal to connect to the debug port of the board**

## **Part 2.4: Log in to the system**

# **Part 3: Video codec**

## **Part 3.1: gst command decoding example**

Here will call the decoding unit of vcu, play the video file `h264_720.mp4`, and display it to the `dp` display.

- 1) Enter the directory where the video file is located
- 2) Enter the following command

```
gst-launch-1.0 uridecodebin
uri=File:///media/sd-mmcblk1p1/h264_720P.mp4 ! queue max-size-
bytes=0 ! videoconvert ! kmssink bus-id=fd4a0000.zynqmp-display
fullscreen-overlay=1
```

At this time, you can see the video screen displayed on the dp monitor. If the monitor supports it, you can replace the video with another resolution video



### Part 3.2: gst command coding example

The codec unit will be used at the same time here. First, capture the usb camera data, encode it into an H264 stream, and send it to 127.0.0.1 (local) via UDP, and the port is 5000. Then through a set of commands, the network stream is received and decoded, and finally displayed on the DP display.

It should be noted that whether the format of the usb camera supports the yuyv 4:2:2 format, if not, it needs to be adjusted to the supported parameters accordingly.

#### 1) Enter decode display command

```
gst-launch-1.0 udpsrc port=5000 buffer-size=10000000
caps="application/x-rtp, media=video, clock-rate=90000,
payload=96, encoding-name=H264" ! rtph264depay ! h264parse !
video/x-h264, alignment=nal ! omxh264dec low-latency=1 ! kmssink
bus-id=fd4a0000.zynqmp-display fullscreen-overlay=1 &
```

#### 2) Enter the capture camera and encode commands

```
gst-launch-1.0 -v v4l2src device=/dev/video0 ! video/x-
raw,format=YUY2,width=1280,height=720 ! videoconvert !
omxh264enc num-slices=16 gop-length=1000 periodicity-idr=1000
control-rate=low-latency prefetch-buffer=true target-bitrate=20000
gop-mode=low-delay-p qp-mode=auto ! video/x-h264,
alignment=nal ! rtph264pay ! udpsink buffer-size=10000000
host=127.0.0.1 port=5000 max-latency=-1 qos-dscp=60 async=false
max-bitrate=120000000 -v
```