

UVC UAC 使用指南

文档版本 V1.3

发布日期 2025-12-16

前言

概述

本文档用于指导调试 UVC 和 UAC。



说明

未经特殊说明，以 XM7606V13 指代 XM76 系列所有芯片。

以 XM7206V11A 指代 XM7206V1 系列所有芯片。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修订日期	版本	修订说明
2024-12-21	V1.0	DI 版本发布。
2025-03-10	V1.1	修订 1.3.1, ConfigUVC.sh 支持配置多种 YUV 格式。 修订 1.3.2, 优化 Formats 和 Frames 配置方式。
2025-06-04	V1.2	支持 XM7206V11A 平台 修订 1.3.1, ConfigUVC.sh 支持 USB3.0 特性配置和支持 USB BULK 传输控制。 增加 3.2, 平台差异对照表。
2025-12-16	V1.3	修订 1.2 章节, 增加 4.9 内核配置指导。 修订 2.2 章节, 增加 4.9 内核配置指导。 修订 1.3 章节, 优化 UVCConfig.sh, 增加 uvc_app 缓存池配置和 uvc 传输带宽配置指导。

目 录

1 UVC 使用指南	1
1.1 配置 USB 为 device 设备	1
1.2 使能 UVC 内核配置项	2
1.2.1 Linux-5.10.y 内核配置项	2
1.2.2 Linux-4.9.y 内核配置项	2
1.3 UVC 设备初始化	4
1.3.1 ConfigUVC.sh	4
1.3.2 uvc_app.conf	17
1.3.3 uvc_app 缓存池配置	18
1.3.4 UVC 传输带宽配置	18
1.4 编译 uvc_app	19
1.5 运行 uvc_app	20
2 UAC 使用指南	22
2.1 配置 USB 为 device 设备	22
2.2 使能 UAC 内核配置项	22
2.2.1 Linux-5.10.y 内核配置项	22
2.2.2 Linux-4.9.y 内核配置项	23
2.3 UAC 设备初始化	25
2.3.1 ConfigUVC.sh	25
2.4 编译 uvc_app (集成 uac 功能)	25
2.5 运行 uvc_app	26
3 附录	28
3.1 alsa_readme.txt	28
3.2 平台差异对照表	29

插图目录

图 1-1 配置 usb 为 device 设备	1
图 1-2 uvc_formats	17
图 1-3 sensor type	20
图 1-4 Potplayer 测试 uvc	21
图 2-1 uac 设备	26
图 2-2 potplayer 测试 uac	27
图 2-3 Audacity 测试 uac	27

1 UVC 使用指南

UVC，全称为：Universal Serial Bus Device Class Definition for Video Devices。是微软与另外几家设备厂商联合推出的为 USB 视频捕获设备定义的协议标准。目前已成为 USB org 标准之一。

XM7206V11A、XM7606V13 平台上使用 UVC 功能，需要 Kernel 支持，并搭配 UVC_APP 使用。Kernel 中需要配置 USB 为 device 设备，并使能 UVC 配置项。UVC_APP 完成 UVC 设备初始化，并使能 UVC。

UVC_APP 路径：sdk/sample/uvc_app

Kernel 路径：sdk/source/kernel/linux-5.10.y、sdk/source/kernel/linux-4.9.y

1.1 配置 USB 为 device 设备

修改 dts 文件：<linux-X.X.y>/lotus/machine/<CHIP_TYPE>/dts/ <CHIP_TYPE>.dtsi。

设置 dwc3 的 dr_mode = "peripheral"；

图1-1 配置 usb 为 device 设备

```
@@ -580,7 +580,7 @@
phy-names = "usb2-phy";
phy_type = "utmi_wide";
maximum-speed = "high-speed";
-   dr_mode = "host";
+   dr_mode = "peripheral";
eps_directions = <0x6a>;
snps,eps_new_init;
eps_map=<0x0 0x1 0x2 0x3 0x4 0x5 0x7>;

@@ -607,7 +607,7 @@
phy-names = "usb3-phy", "usb2-phy";
phy_type = "utmi_wide";
maximum-speed = "super-speed";
-   dr_mode = "host";
+   dr_mode = "peripheral";
eps_directions = <0x6a>;
snps,eps_new_init;
eps_map=<0x0 0x1 0x2 0x3 0x4 0x5 0x7>;
```

1.2 使能 UVC 内核配置项

1.2.1 Linux-5.10.y 内核配置项

```
Device Drivers --->
  <*> Multimedia support --->
    [*] Filter media drivers
      Media device types --->
        [*] Cameras and video grabbers
  [*] USB support --->
    <*> Support for Host-side USB
    <*> xHCI HCD (USB 3.0) support
    -*- Generic xHCI driver for a platform device
    <*> DesignWare USB3 DRD Core Support
    DWC3 Mode Selection (Dual Role mode) --->
    *** Platform Glue Driver Support ***
    <*> Generic OF Simple Glue Layer (NEW)
    <*> USB Gadget Support --->
      <*> USB Gadget functions configurable through configs
      [*] USB Webcam function

Lotus Support --->
  Lotus Device Drivers --->
    <*> Platforms Device DWC3
```

1.2.2 Linux-4.9.y 内核配置项

```
Device Drivers --->
  <*> Multimedia support --->
    [*] Cameras/video grabbers support

[*] USB support --->
  <*> Support for Host-side USB
  [*] Enable USB persist by default
```

<*> xHCI HCD (USB 3.0) support
-*- Generic xHCI driver for a platform device
<*> DesignWare USB3 DRD Core Support
DWC3 Mode Selection (Dual Role mode) --->
<*> Generic OF Simple Glue Layer
<*> USB Gadget Support --->
 <*> USB functions configurable through configs
 [*] USB Webcam function

Lotus Support --->

Lotus Device Drivers --->

<*> Platforms Device DWC3



说明

使能如下内核配置项:

- CONFIG_MEDIA_SUPPORT=y
- CONFIG_MEDIA_SUPPORT_FILTER=y
- CONFIG_MEDIA_CAMERA_SUPPORT=y
- CONFIG_VIDEO_DEV=y
- CONFIG_MEDIA_CONTROLLER=y
- CONFIG_VIDEO_V4L2=y
- CONFIG_VIDEOBUF2_CORE=y
- CONFIG_VIDEOBUF2_V4L2=y
- CONFIG_VIDEOBUF2_MEMOPS=y
- CONFIG_VIDEOBUF2_VMALLOC=y
- CONFIG_USB_SUPPORT=y
- CONFIG_USB_COMMON=y
- CONFIG_USB=y
- CONFIG_USB_XHCI_HCD=y
- CONFIG_USB_XHCI_PLATFORM=y
- CONFIG_USB_DWC3=y
- CONFIG_USB_DWC3_DUAL_ROLE=y
- CONFIG_USB_DWC3_OF_SIMPLE=y
- CONFIG_USB_GADGET=y
- CONFIG_USB_GADGET_VBUS_DRAW=2
- CONFIG_USB_GADGET_STORAGE_NUM_BUFFERS=2
- CONFIG_USB_LIBCOMPOSITE=y
- CONFIG_USB_F_FS=y
- CONFIG_USB_F_UVC=y
- CONFIG_USB_CONFIGFS=y
- CONFIG_USB_CONFIGFS_F_UVC=y
- CONFIG_USB_DWC3_LOTUS=y

1.3 UVC 设备初始化

XM7206V11A、XM7606V13 平台上, UVC 驱动默认支持 3 种流: YUV、MJPEG、H264。每种流所支持的分辨率通过 shell 脚本和 uvc_app.conf 动态配置。

脚本路径: sdk/sample/uvc_app/ConfigUVC.sh。

APP 中配置: sdk/sample/uvc_app/uvc_app.conf。

1.3.1 ConfigUVC.sh

```
#!/bin/sh
```

```
export LD_LIBRARY_PATH=$PWD:$LD_LIBRARY_PATH
```

UVC_ENABLE="true"

UVC1_ENABLE="false"

UAC_ENABLE="true"

UVC_ENABLE_YUYV="true"

UVC_ENABLE_NV21="true"

UVC_ENABLE_MJPEG="true"

UVC_ENABLE_H264="true"

UVC_ENABLE_H265="false"

UVC data endpoint transmission mode: iso or bulk

UVC_TRANSFER_MODE="isoc"

USB speed.

ss: super-speed; hs: high-speed; fs: full-speed.

USB_SPEED="hs"

export VID="0x3337"

export PID="0x4321"

export MANUFACTURER="XMedia"

export PRODUCT="HD USB Camera"

export SERIALNUMBER="20251215"

export YUYV="640x360@30 1280x720@13"

export NV21="640x360@30 1280x720@13"

export MJPEG="640x360@30 1280x720@30 1920x1080@30"

export H264="640x360@30 1280x720@30 1920x1080@30"

export H265="640x360@30 1280x720@30 1920x1080@30"

export YUYV_GUID="YUY2\x00\x00\x10\x00\x80\x00\x00\xaa\x00\x38\x9b\x71"

export YUYV_BITS="16"

export NV21_GUID="NV21\x00\x00\x10\x00\x80\x00\x00\xaa\x00\x38\x9b\x71"

export NV21_BITS="12"

```
export H264_GUID="H264\x00\x00\x10\x00\x80\x00\x00\xaa\x00\x38\x9b\x71"
export H265_GUID="H265\x00\x00\x10\x00\x80\x00\x00\xaa\x00\x38\x9b\x71"
```

```
export FUNCTION_UVC="uvc.usb0"
export FUNCTION_UVC1="uvc1.usb0"
export FUNCTION_UAC1="uac1.usb0"
```

#-USB Device descriptor.

```
function gen_dev_desc() {
    echo "0x01" > bDeviceProtocol
    echo "0x02" > bDeviceSubClass
    echo "0xEF" > bDeviceClass
    echo "0x200" > bcdUSB
    echo $VID > idVendor
    echo $PID > idProduct
    echo "0x100" > bcdDevice
    mkdir -p strings/0x409
    echo $MANUFACTURER > strings/0x409/manufacturer
    echo $PRODUCT > strings/0x409/product
    echo $SERIALNUMBER > strings/0x409/serialnumber
}
```

#-Payload Format Descriptors

```
function gen_payload_format_desc() {
    local PAYLOAD=$1
    local FORMAT=$2
    local GUID=$3
    local BITS=$4
    local RESOLUTIONS=$5
    local CR=1

    mkdir -p streaming/$PAYLOAD/$FORMAT/
```

```

if [ $GUID != "null" ];then
    echo -n -e $GUID > streaming/$PAYLOAD/$FORMAT/guidFormat
fi

if [ $BITS -ne 0 ];then
    echo -n -e $BITS > streaming/$PAYLOAD/$FORMAT/bBitsPerPixel
fi

if [ $FORMAT = "mjpeg" ];then
    # The compression ratio is calculated at 10:1.
    CR=10
elif [ $FORMAT = "h264" ];then
    # The compression ratio is calculated at 50:1.
    CR=50
elif [ $FORMAT = "h265" ];then
    # The compression ratio is calculated at 80:1.
    CR=100
fi

for str in $RESOLUTIONS; do
    # Width x Height
    local RES=$(echo "$str" | awk -F '@' '{print $1}')
    # Interval
    local I=$(echo "$str" | awk -F '@' '{print $2}')
    # Width
    local W=$(echo "$RES" | awk -F 'x' '{print $1}')
    # Height
    local H=$(echo "$RES" | awk -F 'x' '{print $2}')

    RES="{RES}p"

    # microsecond
    local TIME=10000000

```

```

# Interval

local INTERVAL=$((TIME / I))

# Video Frame Buffer Size

local FRAMESIZE=$((W * H))

if [ $BITS -eq 16 ]; then
    FRAMESIZE=$((FRAMESIZE * 2))
else
    # The data before encoding is YUV420
    FRAMESIZE=$((FRAMESIZE * 3))
    FRAMESIZE=$((FRAMESIZE / 2))
fi

FRAMESIZE=$((FRAMESIZE / CR))

# Max Bits Rate

local MAXBITRATE=$((FRAMESIZE * 8))
MAXBITRATE=$((MAXBITRATE * I))

# Min Bits Rate

local MINBITRATE=$MAXBITRATE

mkdir -p streaming/$PAYLOAD/$FORMAT/$RES/
echo "${W}" > streaming/$PAYLOAD/$FORMAT/$RES/wWidth
echo "${H}" > streaming/$PAYLOAD/$FORMAT/$RES/wHeight
echo "${INTERVAL}" > streaming/$PAYLOAD/$FORMAT/$RES/dwFrameInterval
echo "${INTERVAL}" > streaming/$PAYLOAD/$FORMAT/$RES/dwDefaultFrameInterval
echo "${MAXBITRATE}" > streaming/$PAYLOAD/$FORMAT/$RES/dwMaxBitRate
echo "${MINBITRATE}" > streaming/$PAYLOAD/$FORMAT/$RES/dwMinBitRate

if [ $PAYLOAD != "framebased" ];then
    echo "${FRAMESIZE}" >
streaming/$PAYLOAD/$FORMAT/$RES/dwMaxVideoFrameBufferSize
fi

#echo "$FORMAT $W x $H @ $INTERVAL $FRAMESIZE $MAXBITRATE"

```

```

done
}

#-For YUV Payload Format Descriptor
function gen_uncompressed_desc() {
    local PAYLOAD="uncompressed"
    local FORMAT=$1
    local GUID=$2
    local BITS=$3
    local RESOLUTIONS=$4

    gen_payload_format_desc ${PAYLOAD} ${FORMAT} ${GUID} ${BITS} "${RESOLUTIONS}"

    ln -s streaming/$PAYLOAD/$FORMAT/ streaming/header/h/
}

#-For FRAMEBASE Payload Format Descriptor
function gen_framebase_desc() {
    local PAYLOAD="framebased"
    local FORMAT=$1
    local GUID=$2
    local BITS=0
    local RESOLUTIONS=$3

    gen_payload_format_desc ${PAYLOAD} ${FORMAT} ${GUID} ${BITS} "${RESOLUTIONS}"

    ln -s streaming/$PAYLOAD/$FORMAT/ streaming/header/h/
}

#-For MJPEG Payload Format Descriptor
function __gen_mjpeg_desc() {
    local PAYLOAD="mjpeg"
    local FORMAT="mjpeg"

```

```

local GUID="null"

local BITS=0

local RESOLUTIONS=$1

gen_payload_format_desc ${PAYLOAD} ${FORMAT} ${GUID} ${BITS} "${RESOLUTIONS}"

In -s streaming/$PAYLOAD/$FORMAT/ streaming/header/h/
}

#-YUYV Format
function gen_yuyv_desc() {
    gen_uncompressed_desc "yuyv" "${YUYV_GUID}" "${YUYV_BITS}" "${YUYV}"
}

#-NV21 Format
function gen_nv21_desc() {
    gen_uncompressed_desc "nv21" "${NV21_GUID}" "${NV21_BITS}" "${NV21}"
}

#-H264 Format
function gen_h264_desc() {
    gen_framebase_desc "h264" "${H264_GUID}" "${H264}"
}

#-H265 Format
function gen_h265_desc() {
    gen_framebase_desc "h265" "${H265_GUID}" "${H265}"
}

#-MJPEG Format
function gen_mjpeg_desc() {
    __gen_mjpeg_desc "${MJPEG}"
}

```

```

#-UVC common

function __gen_uvc_desc() {
    local FUNNAME=$1

    mkdir -p functions/$FUNNAME
    cd functions/$FUNNAME

    # streaming_maxpacket: 1 ~ 3072
    # high-speed and super-speed: (1 ~ 3072); full-speed: (1 ~ 1023)
    if [ $UVC_TRANSFER_MODE = "isoc" ];then
        # USB ISOC transfer mode
        echo 0 > streaming_bulk
        echo 1 > streaming_interval

        if [ $USB_SPEED = "fs" ];then
            # USB full-speed
            echo 0 > streaming_maxburst
            echo 512 > streaming_maxpacket
        elif [ $USB_SPEED = "hs" ];then
            # USB high-speed
            echo 0 > streaming_maxburst
            echo 3072 > streaming_maxpacket
        elif [ $USB_SPEED = "ss" ];then
            # USB super-speed
            echo 9 > streaming_maxburst
            echo 3072 > streaming_maxpacket
        fi
    else
        # USB BULK transfer mode
        echo 1 > streaming_bulk
        echo 0 > streaming_interval
    fi
}

```

```

if [ $USB_SPEED = "fs" ];then
    # USB full-speed
    echo 0 > streaming_maxburst
    echo 64 > streaming_maxpacket
elif [ $USB_SPEED = "hs" ];then
    # USB high-speed
    echo 0 > streaming_maxburst
    echo 512 > streaming_maxpacket
elif [ $USB_SPEED = "ss" ];then
    # USB super-speed
    echo 9 > streaming_maxburst
    echo 1024 > streaming_maxpacket
fi
fi

mkdir -p control/header/h/
mkdir -p streaming/header/h/

echo "0x0110" > control/header/h/bcdUVC
echo "96000000" > control/header/h/dwClockFrequency

ln -s control/header/h/ control/class/fs/
ln -s control/header/h/ control/class/ss/

#YUYV
if [ $UVC_ENABLE_YUYV = "true" ];then
    gen_yuyv_desc
fi

#NV21
if [ $UVC_ENABLE_NV21 = "true" ];then
    gen_nv21_desc
fi

```

```

#MJPEG
if [ $UVC_ENABLE_MJPEG = "true" ];then
    gen_mjpeg_desc
fi

#H264
if [ $UVC_ENABLE_H264 = "true" ];then
    gen_h264_desc
fi

#H265
if [ $UVC_ENABLE_H265 = "true" ];then
    gen_h265_desc
fi

ln -s streaming/header/h/ streaming/class/fs/
ln -s streaming/header/h/ streaming/class/hs/
ln -s streaming/header/h/ streaming/class/ss/

cd ../../
}

#-UAC common
function __gen_uac_desc() {
    local FUNNAME=$1

    mkdir -p functions/$FUNNAME
    echo "0x01" > functions/$FUNNAME/c_chmask
    echo "16000" > functions/$FUNNAME/c_srate

    echo "0x01" > functions/$FUNNAME/p_chmask
    echo "16000" > functions/$FUNNAME/p_srate
}

```

```

}

#-UVC
function gen_uvc_desc() {
    __gen_uvc_desc "${FUNCTION_UVC}"
}

```

```

#-UVC1
function gen_uvc1_desc() {
    __gen_uvc_desc "${FUNCTION_UVC1}"
}

```

```

#-UAC1
function gen_uac1_desc() {
    __gen_uac_desc "${FUNCTION_UAC1}"
}

```

```

#####
#####

```

```

# Main
APP_PATH=$PWD

```

```

# Step 1
mount -t configfs none /sys/kernel/config/
cd /sys/kernel/config/usb_gadget/
mkdir -p camera
cd camera

```

```

# Step 2
gen_dev_desc

```

```

# Step 3
if [ $UVC_ENABLE = true ];then

```

```

        gen_uvc_desc
    fi

    if [ $UVC1_ENABLE = true ];then
        gen_uvc1_desc
    fi

    if [ $UAC_ENABLE = true ];then
        gen_uac1_desc
    fi

    # Step 4
    #-Create and setup configuration
    mkdir -p configs/c.1/
    echo "500" > configs/c.1/MaxPower
    echo "0x80" > configs/c.1/bmAttributes
    mkdir -p configs/c.1/strings/0x409/
    echo "Config 1" > configs/c.1/strings/0x409/configuration

    # Step 5
    if [ $UVC_ENABLE = true ];then
        ln -s functions/${FUNCTION_UVC}/ configs/c.1/
    fi

    if [ $UVC1_ENABLE = true ];then
        ln -s functions/${FUNCTION_UVC1}/ configs/c.1/
    fi

    if [ $UAC_ENABLE = true ];then
        ln -s functions/${FUNCTION_UAC1}/ configs/c.1/
    fi

    # Step 6

```

```
dwc=$(ls /sys/class/udc)
```

```
echo "${dwc}" > UDC
```

Step 7

```
#cd $APP_PATH
```

```
#./uvc_app
```

 说明

➤ dwFrameInterval

帧间间隔；计算公式： $1000 / \text{FPS} \times 10000$ 。

例如帧率为 30FPS 时： $\text{dwFrameInterval} = 1000 / 30 \times 10000 = 333333$ 。

➤ dwDefaultFrameInterval

默认帧间间隔；计算方法同 dwFrameInterval

➤ dwMaxVideoFrameBufferSize

存放视频帧的 buffer 的最大大小；计算公式： $\text{width} \times \text{height} \times 2$ 。

例如分辨率为 1080p 时： $\text{dwMaxVideoFrameBufferSize} = 1920 \times 1080 \times 2 = 4147200$ 。

➤ dwMaxBitRate

最大 bit 率；计算公式： $\text{width} \times \text{height} \times \text{bpp} \times \text{MaxFps}$

例如分辨率为 360p，帧率为 30fps 的 NV12 时：

$\text{dwMaxBitRate} = 640 \times 360 \times 12 \times 30 = 82944000$ 。

bpp (bits per pixel)，像素深度，YUV422 = 16，YUV420 = 12。

➤ dwMinBitRate

最小 bit 率；计算公式： $\text{width} \times \text{height} \times \text{bpp} \times \text{MinFps}$

➤ 对于 MJPEG、H264 等压缩格式，BitRate 根据实际码率来设置。一般 jpeg 的压缩率 (CR) 为 0.2 ~ 0.5，H264 的压缩率 (CR) 为 0.01。

因此 $\text{BitRate} = \text{width} \times \text{height} \times \text{bpp} \times \text{fps} \times \text{CR}$ 。

1.3.2 uvc_app.conf

图1-2 uvc_formats

```
UVC Camera Application configuration file
;it need to be located at /etc/ directory

[stream]
; width-height-fps,width-height-fps
fmt1 = yuyv,640-360-30,1280-720-13;
fmt2 = nv21,640-360-30,1280-720-13,1920-1080-30;
fmt3 = mjpeg,640-360-30,1280-720-30,1920-1080-30;
fmt4 = h264,640-360-30,1280-720-30,1920-1080-30;
fmt5 = h265,640-360-30,1280-720-30,1920-1080-30;

[uvc]
; set max PayloadImage size
imagesize = 0x200000
bulksize = 0x200000

; set cache node number
cache_count = 3

; 0x1: IO_METHOD_MMAP, 0x2: IO_METHOD_USERPTR
iomethod = 2

; cat /sys/kernel/config/usb_gadget/camera/functions/uvc.usb0/streaming_maxpacket
maxpacket = 3072
~
```

说明

uvc_app.conf 中 Format 的顺序，以及每种 Format 中 Frame 的顺序，需要和 ConfigUVC.sh 中配置流时的顺序一致。例如 1.3.1 节的脚本中配置 Format 的顺序如下，yuyv, nv21, mjpeg, h264, h265:

```
265 ln -s control/header/h/ control/class/fs/
266 ln -s control/header/h/ control/class/ss/
267
268 #YUYV
269 if [ $UVC_ENABLE_YUYV = "true" ];then
270     gen_yuyv_desc
271 fi
272
273 #NV21
274 if [ $UVC_ENABLE_NV21 = "true" ];then
275     gen_nv21_desc
276 fi
277
278 #MJPEG
279 if [ $UVC_ENABLE_MJPEG = "true" ];then
280     gen_mjpeg_desc
281 fi
282
283 #H264
284 if [ $UVC_ENABLE_H264 = "true" ];then
285     gen_h264_desc
286 fi
287
288 #H265
289 if [ $UVC_ENABLE_H265 = "true" ];then
290     gen_h265_desc
291 fi
292
293 ln -s streaming/header/h/ streaming/class/fs/
294 ln -s streaming/header/h/ streaming/class/hs/
295 ln -s streaming/header/h/ streaming/class/ss/
296
```

例如：配置 MJPEG 的 Frame 顺序如下：360p, 720p, 1080p。

```
export YUYV="640x360@30 1280x720@13"
export NV21="640x360@30 1280x720@13"
export MJPEG="640x360@30 1280x720@30 1920x1080@30"
export H264="640x360@30 1280x720@30 1920x1080@30"
export H265="640x360@30 1280x720@30 1920x1080@30"
```

1.3.3 uvc_app 缓存池配置

```
UVC Camera Application configuration file
;it need to be located at /etc/ directory

[stream]
; width-height-fps,width-height-fps
fmt1 = yuyv,640-360-30,1280-720-13;
fmt2 = nv21,640-360-30,1280-720-13,1920-1080-30;
fmt3 = mjpeg,640-360-30,1280-720-30,1920-1080-30;
fmt4 = h264,640-360-30,1280-720-30,1920-1080-30;
fmt5 = h265,640-360-30,1280-720-30,1920-1080-30;

[uvc]
; set max PayloadImage size
imagesize = 0x200000
bulksize = 0x200000

; set cache node number
cache_count = 3

; 0x1: IO_METHOD_MMAP, 0x2: IO_METHOD_USERPTR
iomethod = 2

; cat /sys/kernel/config/usb_gadget/camera/functions/uvic.usb0/streaming_maxpacket
maxpacket = 3072
```

缓存池的内存空间大小为：imagesize * cache_count。

注意： uvc_app/video/hal/uvic_hal.c 中： $1 < \text{WAITED_NODE_SIZE} < \text{cache_count}$ 。

bulksize 仅在使用 bulk 传输模式时生效，bulksize 应等于 imagesize。

1.3.4 UVC 传输带宽配置

```
UVC Camera Application configuration file
;it need to be located at /etc/ directory

[stream]
; width-height-fps,width-height-fps
fmt1 = yuyv,640-360-30,1280-720-13;
fmt2 = nv21,640-360-30,1280-720-13,1920-1080-30;
fmt3 = mjpeg,640-360-30,1280-720-30,1920-1080-30;
fmt4 = h264,640-360-30,1280-720-30,1920-1080-30;
fmt5 = h265,640-360-30,1280-720-30,1920-1080-30;

[uvic]
; set max PayloadImage size
imagesize = 0x200000
bulksize = 0x200000

; set cache node number
cache_count = 3

; 0x1: IO_METHOD_MMAP, 0x2: IO_METHOD_USERPTR
iomethod = 2

; cat /sys/kernel/config/usb_gadget/camera/functions/uvic.usb0/streaming_maxpacket
maxpacket = 3072

23 # UVC data endpoint transmission mode: iso or bulk
24 UVC_TRANSFER_MODE="isoc"
25
26 # USB speed.
27 # ss: super-speed; hs: high-speed; fs: full-speed.
28 USB_SPEED="hs"
29
```

```

219 # streaming_maxpacket: 1 ~ 3072
220 # high-speed and super-speed: (1 ~ 3072); full-speed: (1 ~ 1023)
221 if [ $UVC_TRANSFER_MODE = "isoc" ];then
222     # USB ISOC transfer mode
223     echo 0 > streaming_bulk
224     echo 1 > streaming_interval
225
226     if [ $USB_SPEED = "fs" ];then
227         # USB full-speed
228         echo 0 > streaming_maxburst
229         echo 512 > streaming_maxpacket
230     elif [ $USB_SPEED = "hs" ];then
231         # USB high-speed
232         echo 0 > streaming_maxburst
233         echo 3072 > streaming_maxpacket
234     elif [ $USB_SPEED = "ss" ];then
235         # USB super-speed
236         echo 9 > streaming_maxburst
237         echo 3072 > streaming_maxpacket
238     fi
239 else
240     # USB BULK transfer mode
241     echo 1 > streaming_bulk
242     echo 0 > streaming_interval
243
244     if [ $USB_SPEED = "fs" ];then
245         # USB full-speed
246         echo 0 > streaming_maxburst
247         echo 64 > streaming_maxpacket
248     elif [ $USB_SPEED = "hs" ];then
249         # USB high-speed
250         echo 0 > streaming_maxburst
251         echo 512 > streaming_maxpacket
252     elif [ $USB_SPEED = "ss" ];then
253         # USB super-speed
254         echo 9 > streaming_maxburst
255         echo 1024 > streaming_maxpacket
256     fi
257 fi
258

```

注意： uvc_app.conf 中 maxpacket 的取值计算公式为如下：

$$\text{maxpacket} = \text{streaming_maxpacket} * (\text{streaming_maxburst} + 1).$$

ISOC 模式时，UVC 的传输带宽计算公式如下：

$$\text{streaming_maxpacket} * (\text{streaming_maxburst} + 1) * (8 / 2^{(\text{streaming_interval}-1)}) * 1000$$

(bytes / s)

1.4 编译 uvc_app

步骤 1 参考《SDK 开发环境用户指南》完成 SDK 编译。

步骤 2 配置 SENSOR 类型。

vi sample/sample_base.mk

图1-3 sensor type

```
SENSOR0_TYPE := GALAXYCORE_GC2053_MIPI_2M_30FPS_10BIT
SENSOR1_TYPE := OMNIVISION_OS08A20_MIPI_8M_30FPS_10BIT
SENSOR2_TYPE := OMNIVISION_OS08A20_MIPI_8M_30FPS_10BIT
SENSOR3_TYPE := OMNIVISION_OS08A20_MIPI_8M_30FPS_10BIT
SENSOR4_TYPE := OMNIVISION_OS08A20_MIPI_8M_30FPS_10BIT

SAMPLE_CFLAGS += -DSENSOR0_TYPE=$(SENSOR0_TYPE)
SAMPLE_CFLAGS += -DSENSOR1_TYPE=$(SENSOR1_TYPE)
SAMPLE_CFLAGS += -DSENSOR2_TYPE=$(SENSOR2_TYPE)
SAMPLE_CFLAGS += -DSENSOR3_TYPE=$(SENSOR3_TYPE)
SAMPLE_CFLAGS += -DSENSOR4_TYPE=$(SENSOR4_TYPE)
```

SENSOR TYPE 请参考 sample_comm_sensor_type, 位于:

sdk/sample/common/sample_comm.h

步骤 3 编译 uvc_app

cd sample/ ; make -j

----结束

1.5 运行 uvc_app

步骤 1 参考《SDK 开发环境用户指南》完成镜像烧写和驱动加载。

步骤 2 将 uvc_app 目录下的 ConfigUVC.sh、uvc_app、uvc_app.conf 拷贝到板端。

步骤 3 运行 ConfigUVC.sh。

步骤 4 运行 uvc_app。

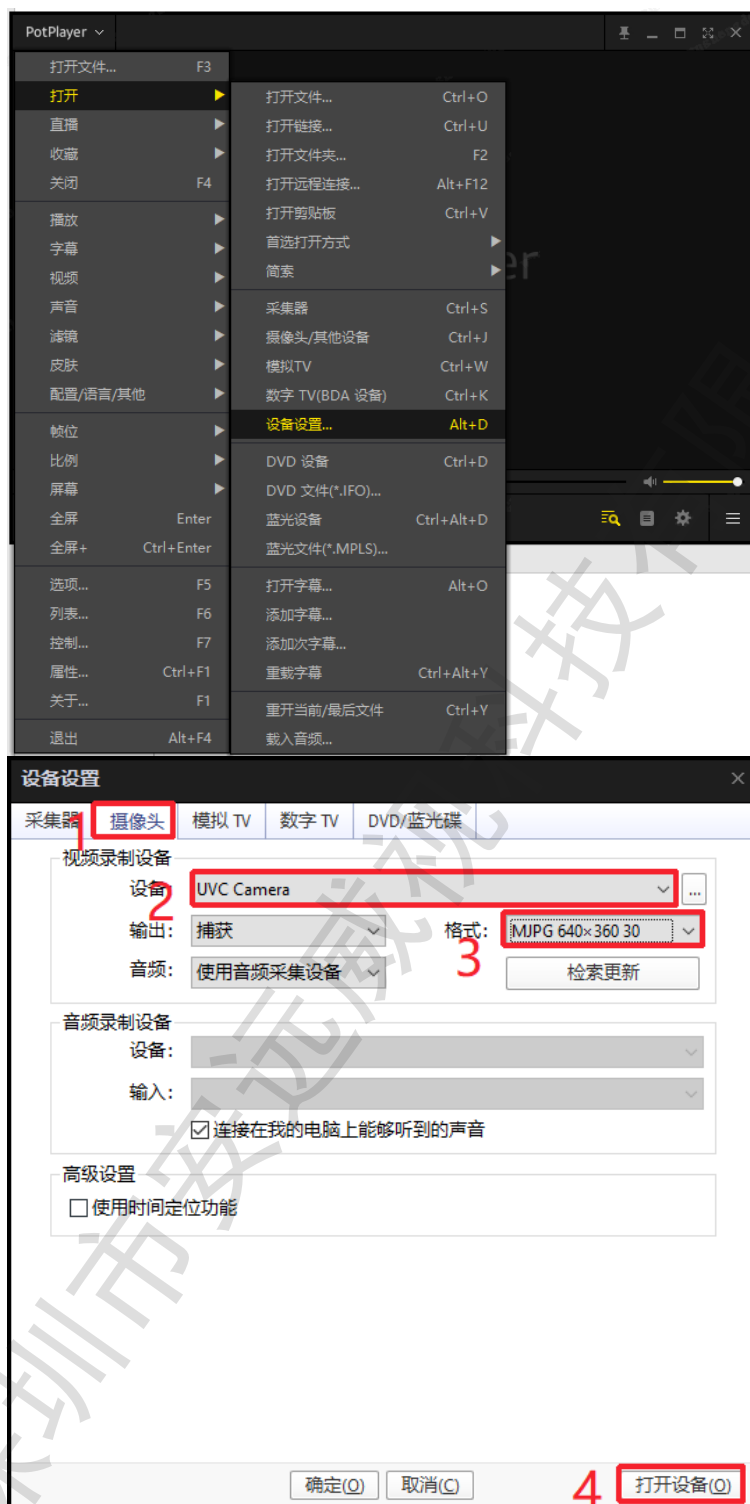
步骤 5 使用 usb 公对公线连接板端和 PC。

成功后 PC 端可看到 UVC Camera:

我的电脑->属性->设备管理->照相机->UVC Camera

步骤 6 PC 端运行 Potplayer 打开 UVC Camera。

图1-4 Potplayer 测试 uvc



----结束

2 UAC 使用指南

UAC, 全称为: Universal Serial Bus Device Class Definition for Audio Devices。是 USB 开发者论坛推出的为 USB 音频设备定义的协议标准。目前已成为 USB org 标准之一。

XM7206V11A、XM7606V13 平台上使用 UAC 功能, 需要 Kernel 支持, 并搭配 UVC_APP 使用。Kernel 中需要配置 USB 为 device 设备, 并使能 UAC 配置项。UVC_APP 完成 UAC 设备初始化, 并使能 UAC。

UVC_APP 路径: sdk/sample/uvc_app

Kernel 路径: sdk/source/kernel/linux-5.10.y、sdk/source/kernel/linux-4.9.y

2.1 配置 USB 为 device 设备

参考 1.1 节。

2.2 使能 UAC 内核配置项

2.2.1 Linux-5.10.y 内核配置项

```
Device Drivers  --->
  <*> Sound card support  --->
    <*>  Advanced Linux Sound Architecture  --->
      [*]  PCM timer interface
      [*]  Support old ALSA API
      [*]  Sound Proc FS Support
      [*]  Verbose procfs contents
      [*]  Generic sound devices  --->
      [*]  USB sound devices  --->
```

```

        <*>    USB Audio/MIDI driver
[*] USB support  --->
    <*>    Support for Host-side USB
    <*>    xHCI HCD (USB 3.0) support
    -*     Generic xHCI driver for a platform device
    <*>    DesignWare USB3 DRD Core Support
    DWC3 Mode Selection (Dual Role mode)  --->
    *** Platform Glue Driver Support ***
    <*>    Generic OF Simple Glue Layer (NEW)
    <*>    USB Gadget Support  --->
        <*>    USB Gadget functions configurable through configs
        [*]    Audio Class 1.0
        [*]    Audio Class 2.0

Lotus Support  --->
    Lotus Device Drivers  --->
        <*> Platforms Device DWC3

```

2.2.2 Linux-4.9.y 内核配置项

```

Device Drivers  --->
    <*> Sound card support  --->
        <*> Advanced Linux Sound Architecture  --->
            [*] PCM timer interface (NEW)
            [*] Support old ALSA API (NEW)
            [*] Sound Proc FS Support (NEW)
            [*] Verbose procfs contents (NEW)
            [*] Generic sound devices (NEW)  --->
            [*] USB sound devices (NEW)  --->
                <*>    USB Audio/MIDI driver

[*] USB support  --->
    <*>    Support for Host-side USB
    [*]    Enable USB persist by default

```

<*> xHCI HCD (USB 3.0) support
 -* Generic xHCI driver for a platform device
 <*> DesignWare USB3 DRD Core Support
 DWC3 Mode Selection (Dual Role mode) --->
 <*> Generic OF Simple Glue Layer
 <*> USB Gadget Support --->
 <*> USB functions configurable through configs
 [*] Audio Class 1.0
 [*] Audio Class 2.0

Lotus Support --->

Lotus Device Drivers --->

<*> Platforms Device DWC3

 说明

uac 需开启如下内核配置项:

- CONFIG_SOUND=y
- CONFIG_SND=y
- CONFIG_SND_PCM=y
- CONFIG_SND_HDA_PREALLOC_SIZE=64
- CONFIG_USB_SUPPORT=y
- CONFIG_USB_COMMON=y
- CONFIG_USB=y
- CONFIG_USB_XHCI_HCD=y
- CONFIG_USB_XHCI_PLATFORM=y
- CONFIG_USB_DWC3=y
- CONFIG_USB_DWC3_DUAL_ROLE=y
- CONFIG_USB_DWC3_OF_SIMPLE=y
- CONFIG_USB_GADGET=y
- CONFIG_USB_GADGET_VBUS_DRAW=2
- CONFIG_USB_GADGET_STORAGE_NUM_BUFFERS=2
- CONFIG_USB_LIBCOMPOSITE=y
- CONFIG_USB_U_AUDIO=y

- CONFIG_USB_F_FS=y
- CONFIG_USB_F_UAC1=y
- CONFIG_USB_CONFIGFS=y
- CONFIG_USB_CONFIGFS_F_UAC1=y
- CONFIG_USB_CONFIGFS_F_UAC2=y
- CONFIG_USB_DWC3_LOTUS=y

2.3 UAC 设备初始化

XM7206V11A、XM7606V13 平台上，执行 ConfigUVC.sh 完成 UAC 设备初始化，UVC_APP 使能 UAC。

脚本路径：sdk/sample/uvic_app/ConfigUVC.sh。

2.3.1 ConfigUVC.sh

参考 1.3.1 节，设置 `UAC_ENABLE="true"`

```
#-UAC common
function __gen_uac_desc() {
    local FUNNAME=$1

    mkdir -p functions/$FUNNAME
    echo "0x01" > functions/$FUNNAME/c_chmask
    echo "16000" > functions/$FUNNAME/c_srate

    echo "0x01" > functions/$FUNNAME/p_chmask
    echo "16000" > functions/$FUNNAME/p_srate
}
```

2.4 编译 uvc_app (集成 uac 功能)

步骤 1 参考 sample/uvic_app/alsa_readme.txt，使用交叉编译工具链编译 alsa 库。

步骤 2 配置 sample/uvic_app/Makefile 中 “UAC_ENABLE := true”。

步骤 3 参考 1.4 节编译 uvc_app。



说明

UAC 使用了三方开源库 alsa-lib-1.1.7，用户需自行下载软件包，并放到 sdk/sample/uvic_app/audio/alsa/opensource 目录下，

sdk/sample/uvc_app/audio/alsa/Makefile 会编译该软件包，编译成功后，编译服用器上会生成 /home/\$(USER)/audio/alsa 目录，用户需将该目录下的所有文件拷贝到板端 /home/\$(USER)/audio/alsa 目录下！

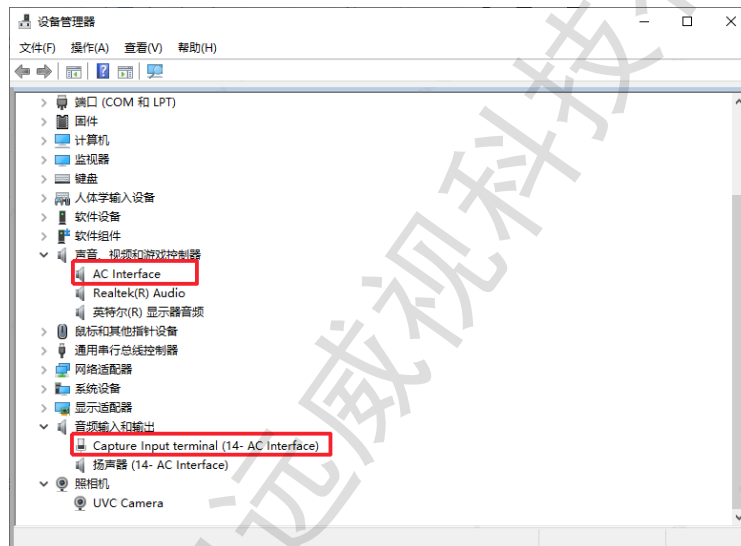
----结束

2.5 运行 uvc_app

步骤 1 参考 1.5 节配置 uvc_app，并运行： `./uvc_app -uac`

步骤 2 运行成功后 PC 上可看到如下设备：

图2-1 uac 设备



步骤 3 uac 测试。

3.1 alsa_readme.txt

1.download alsa-lib source code

Note: The alsa-lib source package is not released by default, only dynamic library files are released. The alsa-lib source package needs to be downloaded from the open source community.

Download the source code of alsa-lib v1.1.7 from the alsa-project open source community:

- 1) Go to the website: www.alsa-project.org
- 2) Select the <http://www.alsa-project.org/main/index.php/Download> option of the HTTP protocol resource to enter the subpage
- 3) Download `alsa-lib-1.1.7.tar.bz2`

2.compile alsa-lib

1) Store the downloaded `alsa-lib-1.1.7.tar.bz2` in the `sdk/sample/uvic_app/audio/alsa/opensource` directory.

2) In the linux server, enter the `sdk/sample/uvic_app/audio/alsa/opensource` directory and execute the following command:

```
tar -xjvf alsa-lib-1.1.7.tar.bz2
cd ./alsa-lib-1.1.7/
mkdir -p /home/install/alsa-lib-1.1.7/
./configure CC=arm-gcc12.2.0-linux-gcc STRIP=arm-gcc12.2.0-linux-strip --
host=arm-gcc12.2.0-linux --prefix=/home/install/alsa-lib-1.1.7/ --enable-static=no --
enable-shared=yes --with-configdir=/home/audio/alsa/ --disable-python
make
make install
cp -r /home/install/alsa-lib-1.1.7/lib/ ../..
cp -r /home/install/alsa-lib-1.1.7/include/ ../..
```

Note: The above command is used to generate the dynamic link library of alsa-lib. `--host` is used to specify the cross compiler. `--prefix` is used to specify the installation path of the compiled file. `--enable-static` is used to specify static library support. `--`

enable-shared is used to specify dynamic library support. --with-configdir is used to specify the installation path of the config file (also the path of the config file on the board).

3.install to the embedded platform

On the target board, the following files need to be copied to the corresponding location:

1) Copy the libasound.so.2 file in the /home/install/alsa-lib-1.1.7/lib/ directory on the server to the /lib/ directory on the board.

2) Copy all the files in the /home/audio/alsa/ directory on the server to the /home/audio/alsa/ directory of the board (create it if it does not exist).

3.2 平台差异对照表

	UVC	UAC	UVC Bulk Transfer	UVC ISO Transfer	USB 3.0	USB 2.0
XM7606V13	Y	Y	Y	Y	Y	Y
XM7206V11A	Y	Y	Y	Y	N	Y