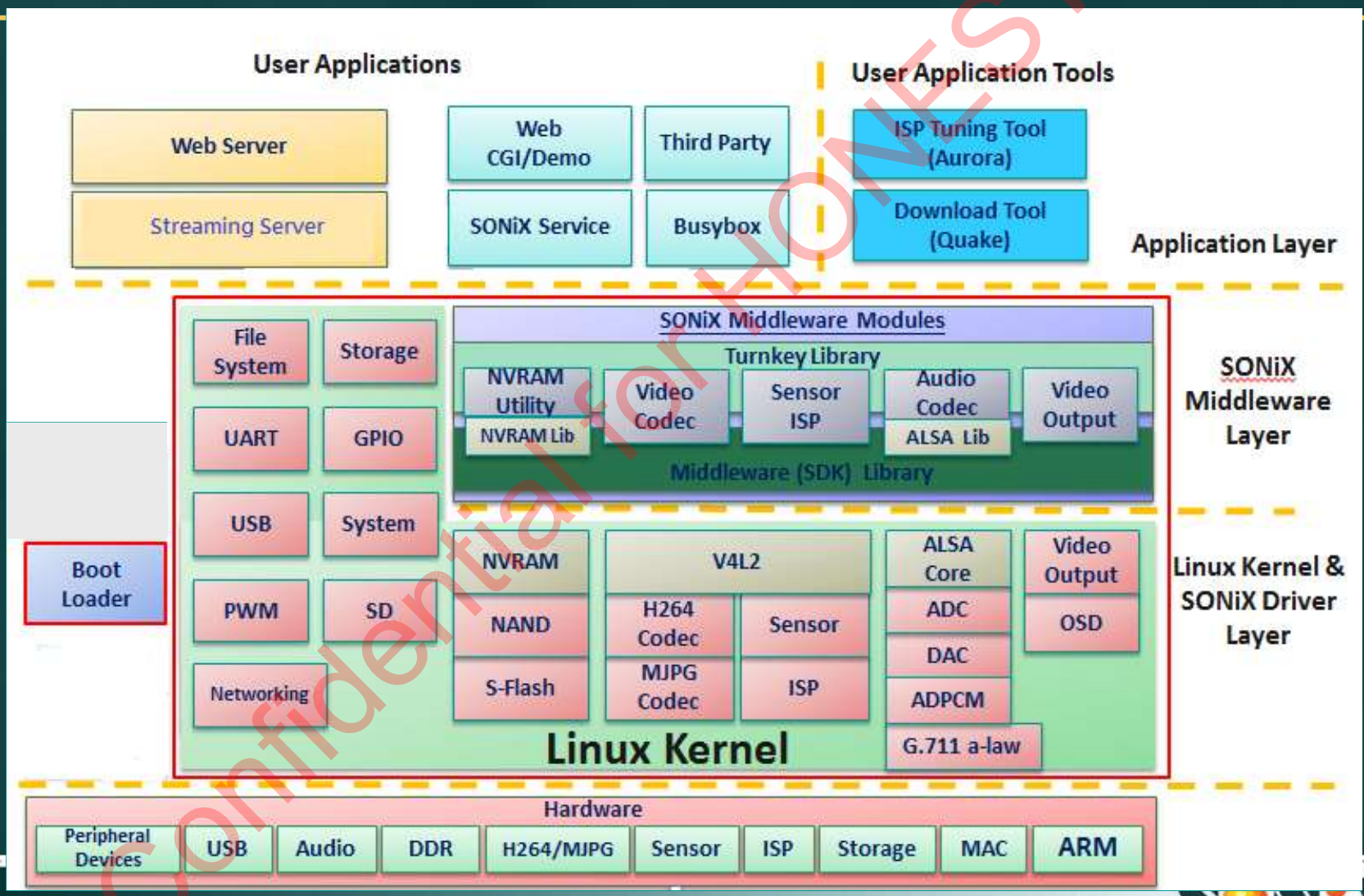
Linux V4L2 ALSA 架构简介and 外围IO 控制

SONIX TECHNOLOGY CO., LTD.

大纲

- SN98600 SDK 架构
  - Example code 目录说明
- Video Programming
  - Video encode 场景说明
  - Sonix Video Middleware
- Audio Programming
  - Alsa library
- Peripherals
  - GPIO
  - SPI
  - I2C
  - PWM
  - AES/DES/3DES/CRC
  - WDT / Timer / RTC
- Video output programming

SN98600 SDK 架构



SN986 Series SDK 范例程序目录

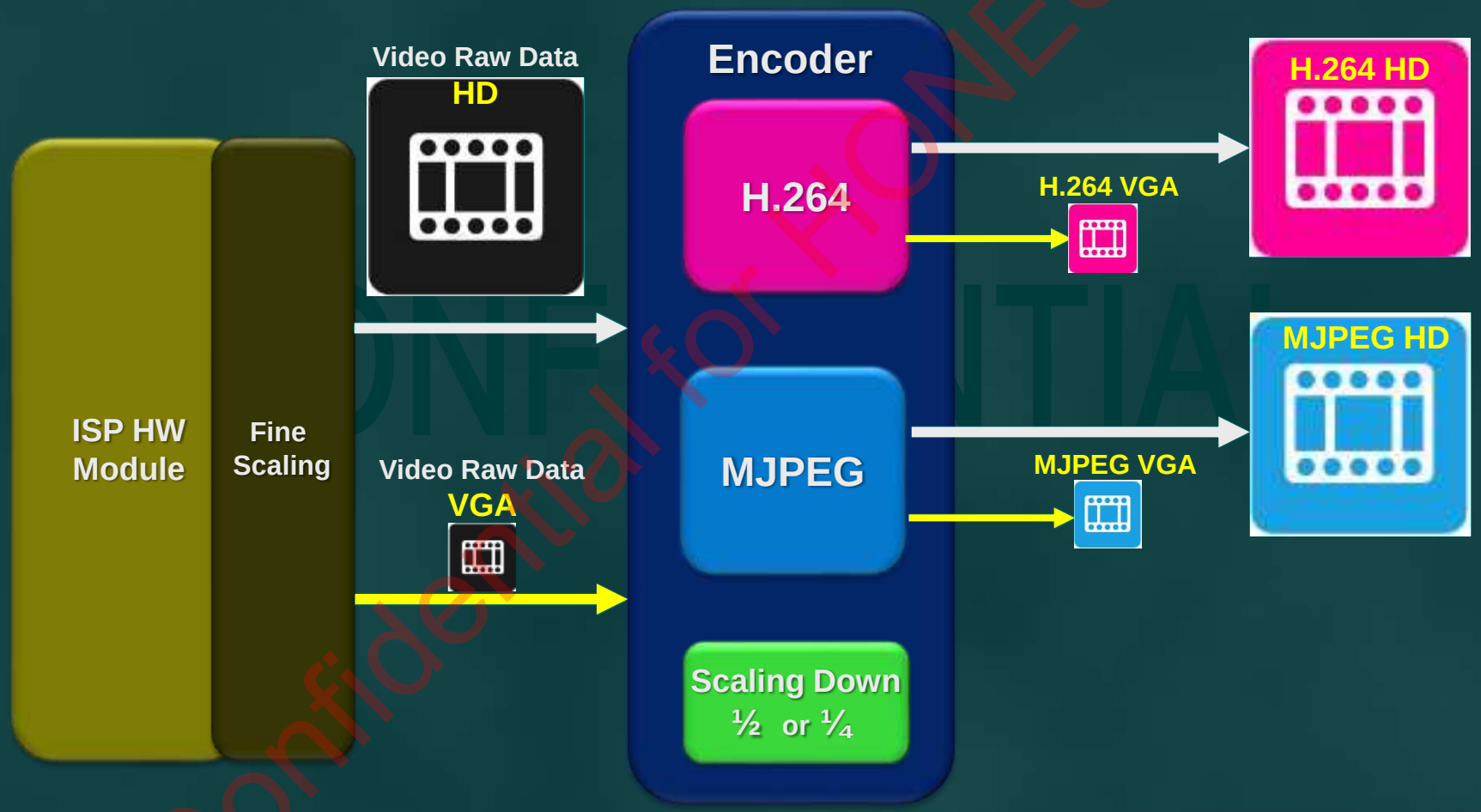
- App/example/src

Directory Name	Descriptions
Audio	Audio 录放音范例程序
Crypto	AES/DES/3DES/CRC16 范例程序
Gpio	Gpio 控制范例程序
I2c	I2c 控制范例程序
Pwm	Pwm 控制范例程序
Rtc	RTC 控制范例程序
Spi	Spi 控制范例程序
Timer	HW timer 范例程序
Video	Video 多路streaming 范例程序
Video_output	Video output范例程序
Watchdog	Watchdog 范例程序

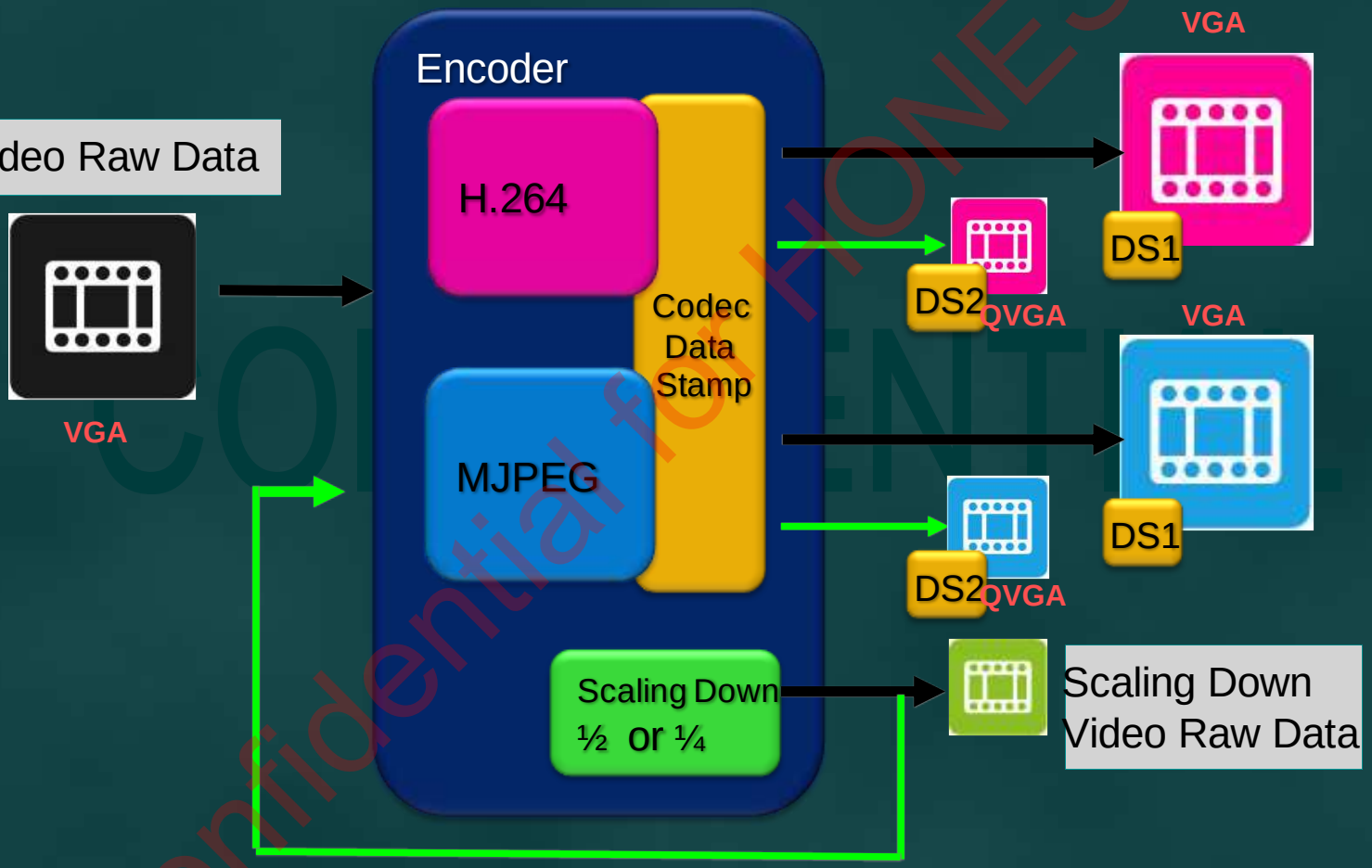
Video Programming

SONIX TECHNOLOGY CO., LTD.

压缩工作场景



压缩工作场景 2



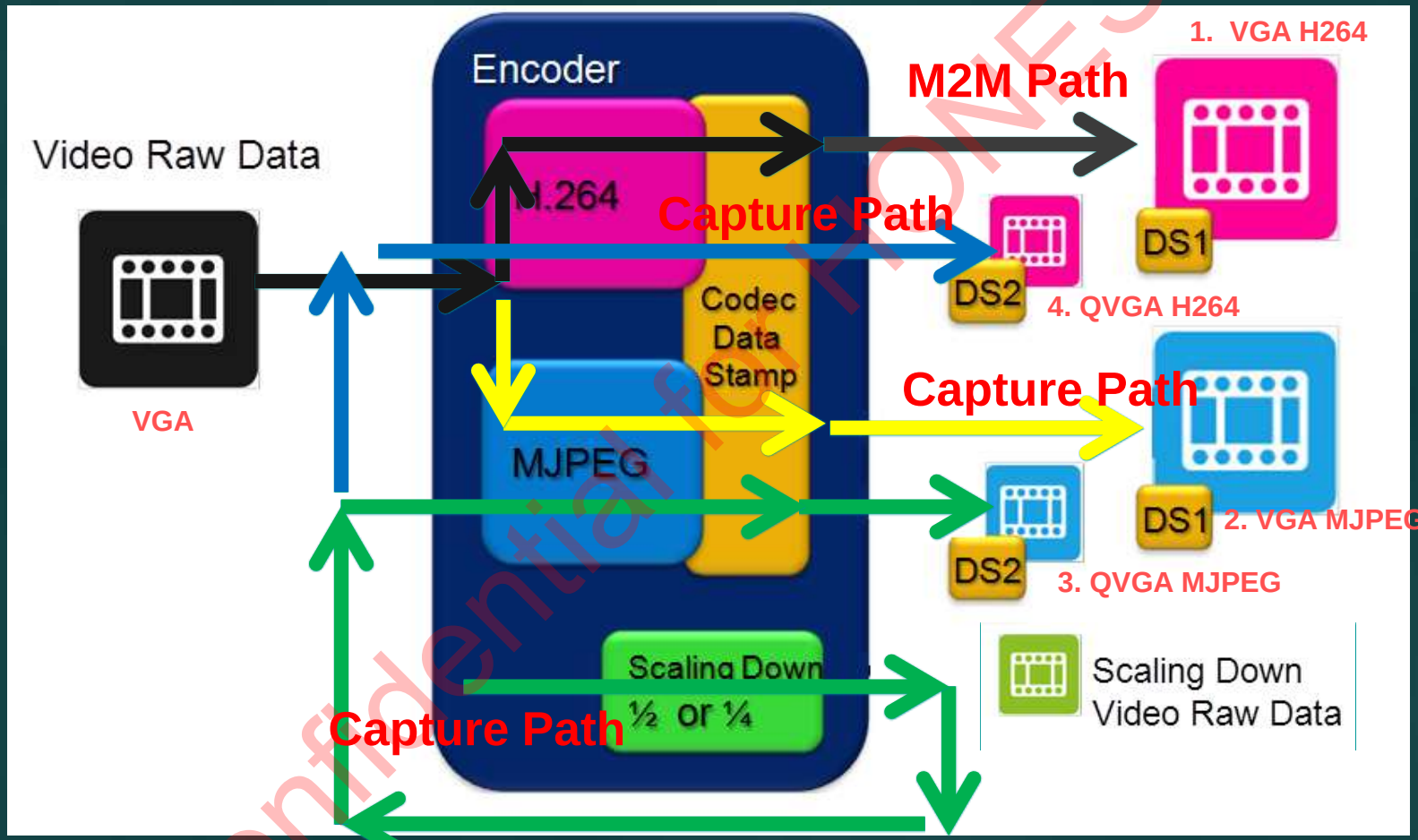
Video Encode

- ISP
  - Driver snx_isp.ko
  - 预设节点
    - /dev/video0
- Video encode
  - Driver snx_vb2.ko and snx_vc.ko
  - 预设节点
    - /dev/video1 and video2

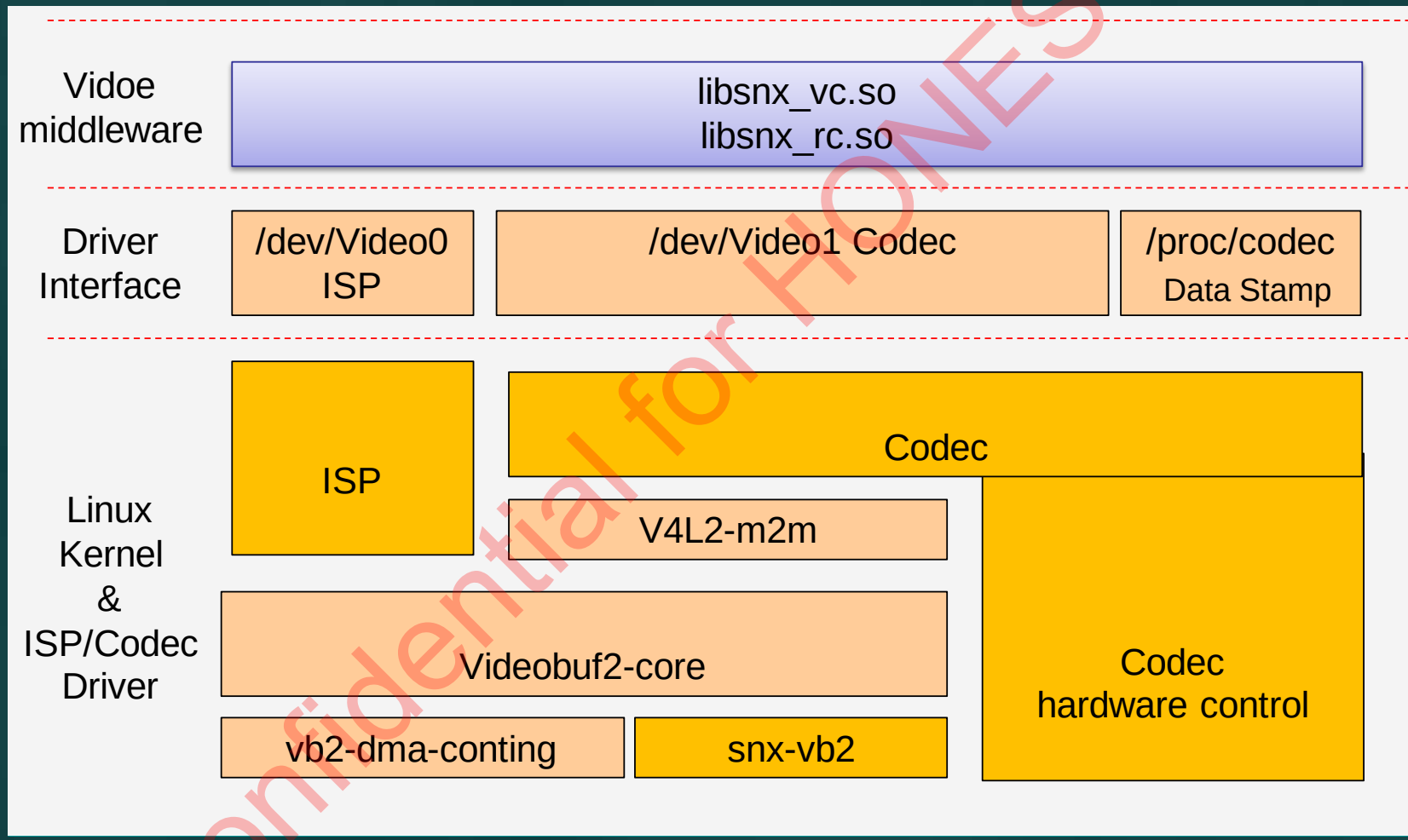
Video encode 流程描述

- 多码流 by time interleaving
- M2M Path
  - Encode source is from ISP capture
  - 可以独立存在
  - 最多同时2组m2m path存在 (isp0, isp1)
- Capture Path
  - Encode source is from existed path
  - 必须有对应的m2m path

M2M/Capture Path 说明



Video Codec 架构



Video Middleware

- 基于V4L2 Memory2Memory架构开发的中间层
  - 省去study V4L2 library 的时间
  - 简单了解开发的Sonix Video Middleware APIs
- 包含 Video Encode, Rate Control, Data Stamp
  - Video Encode 包括 H264, MJPEG, JPEG Encode APIs
  - Rate Control
    - Set QP for VBR (HW)
    - Sonix RC APIs for CBR (SW)
  - Data Stamp
    - Encode时添加图标或文字

Video Middleware APIs

Middleware API

libsnx_vc & libsnx_rc

- snx_open_device
- snx_isp_init
- snx_isp_start
- snx_isp_uninit
- snx_isp_stop
- snx_codec_init
- snx_codec_start
- snx_codec_read
- snx_codec_uninit
- snx_codec_stop
- snx_codec_reset
- snx_codec_set_qp
- snx_codec_get_qp
- snx_codec_rc_update
- snx_codec_rc_init

Driver interface

/dev/Video0 ISP

/dev/Video1 /dev/Video2 Codec

详细内容请参考:

SN986 Series Video Codec Programing Guide

Video Encode Middleware APIs

Function Name	Descriptions
snx_open_device(char *dev_name)	Open device name
snx_isp_init(struct snx_m2m * m2m)	Integrate V4L2 I/O control: VIDIOC_QUERYCAP, VIDIOC_S_CTRL and VIDIOC_S_FMT
snx_isp_start(struct snx_m2m * m2m)	Integrate V4L2 I/O control: VIDIOC_REQBUFS, VIDIOC_QUERBUF, VIDIOC_QBUF, VIDIOC_STREAMON
snx_isp_stop(struct snx_m2m * m2m)	Integrate V4L2 I/O control: VIDIOC_STREAMOFF
snx_isp_uninit(struct snx_m2m * m2m)	release memory

Video Encode Middleware APIs (2)

Function Name	Descriptions
snx_codec_init(struct snx_m2m * m2m)	Integrate V4L2 I/O control: VIDIOC_QUERYCAP, VIDIOC_S_CTRL and VIDIOC_S_FMT
snx_codec_start(struct snx_m2m * m2m)	Integrate V4L2 I/O control: VIDIOC_REQBUFS, VIDIOC_QUERBUF, VIDIOC_QBUF, VIDIOC_STREAMON
snx_codec_read(struct snx_m2m * m2m)	Integrate V4L2 I/O control: VIDIOC_DQBUF, VIDIOC_QBUF
snx_codec_reset(struct snx_m2m * m2m)	Integrate V4L2 I/O control: VIDIOC_DQBUF, include snx_isp_read() API
snx_codec_stop(struct snx_m2m * m2m)	Integrate V4L2 I/O control: VIDIOC_STREAMOFF
snx_codec_uninit(struct snx_m2m * m2m)	release memory

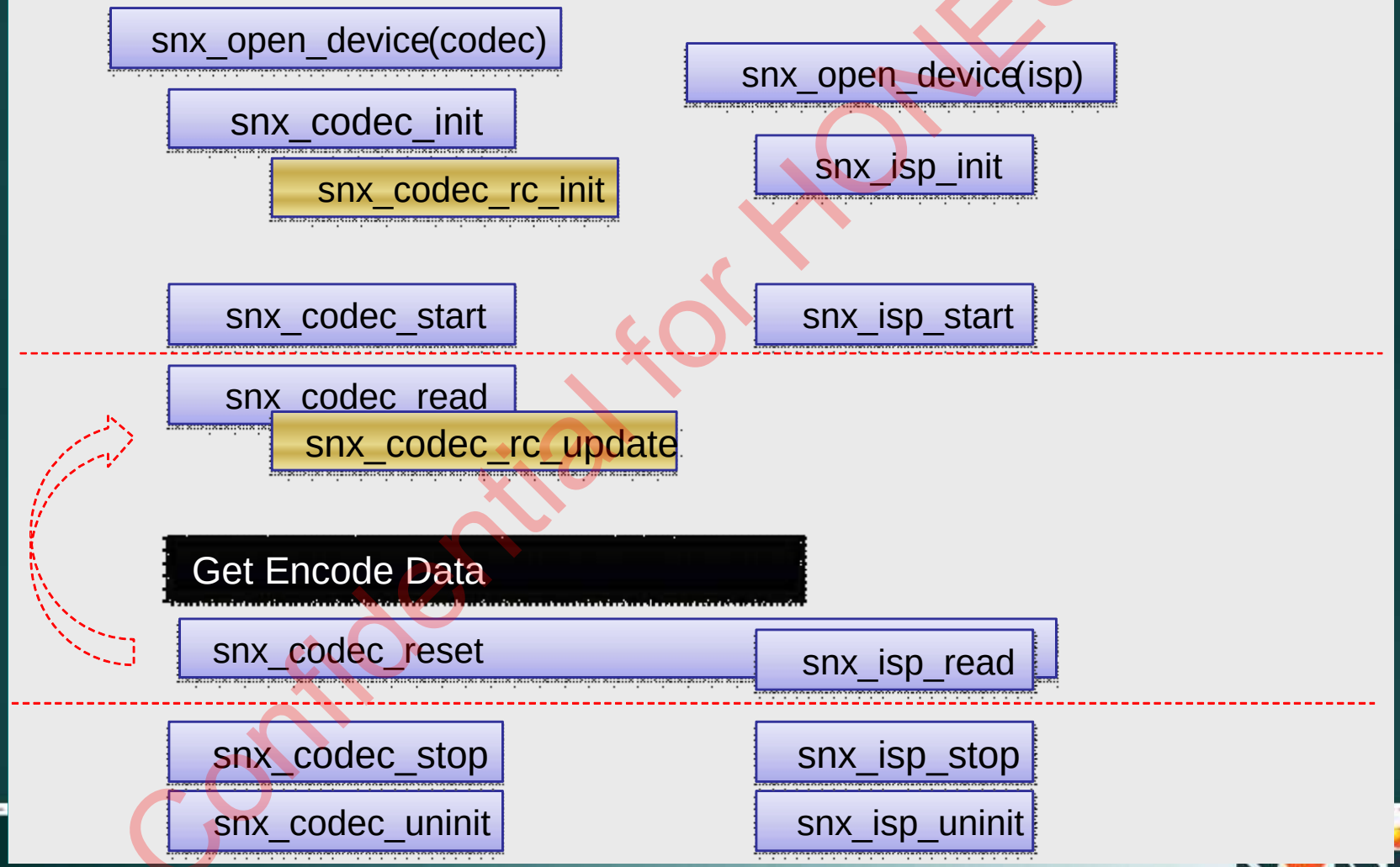
Video Rate Control Middleware APIs

Function Name	Descriptions
snx_codec_set/gp_qp	set / get QP
snx_codec_rc_init	Rate control function init function
snx_codec_rc_update	software algorithm calculator new QP value

Video Data Stamp Middlware APIs

Function Name	Descriptions
snx_cds_get_enable	Get codec data stamp enable/disable status
snx_cds_set_enable	Set codec data stamp enable/disable
snx_cds_set/get_scale	Set / get codec data stamp scale ratio
snx_cds_set/get_color_attr	Set / get codec data stamp color attrubilt
snx_cds_set/get_color	Set / get codec data stamp foreground/background color
snx_cds_set/get_position	Set / get codec data stamp position/dimension
snx_cds_set/get_datastamp	Set / get codec data stamp value (binary)
snx_cds_set_string	Set string to codec data stamp (ascii)
snx_cds_set_bmp	Set bitmap file to codec data stamp (bitmap)

M2M Path with Rate Control



Data Stamp

- 4096 color with transparent
- 支持前景色和背景色
- x1 x2 x4 scaling
- 16x16 pixels per char
- Time interleaving

Data stamp programming流程

Video Example Code

- Example/Video/
  - snx_m2m_one_stream
  - snx_m2m_two_stream
  - snx_m2m_capture_2stream
  - snx_m2m_capture_4stream
  - snx_m2m_one_stream_with_rc
  - snx_m2m_capture_2stream_with_rc
  - snx_vc_ds

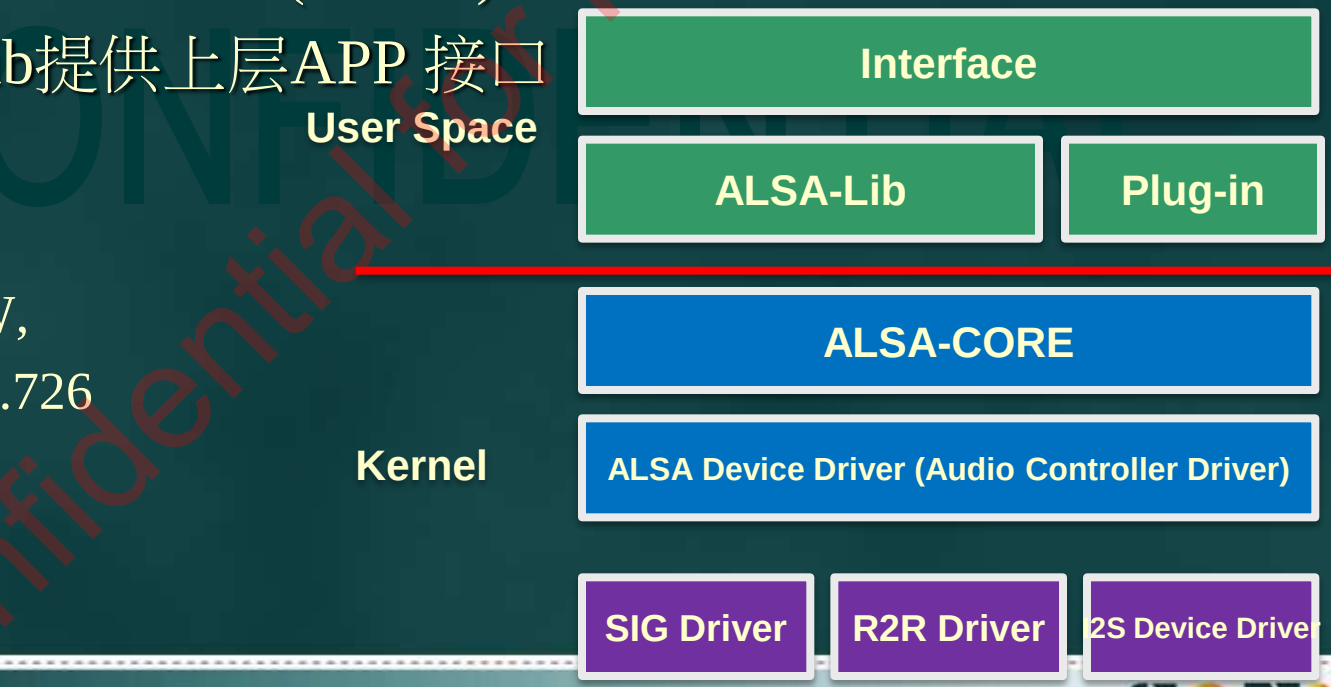
Audio Programming

Audio Codec Feature

- Driver snx_aud_core.ko snx_aud_sigma.ko snx_aud_r2r.ko
- Audio codec formats:
  - Hardware audio codec
    - MS-ADPCM
    - A-LAW
  - Software audio codec
    - A-LAW
    - Mu-LAW
    - G.722
    - G.726
- Audio Codec sample rate
  - 8/16/24/32/44.1/48K bps

Audio SDK 架构

- 使用ALSA 架构 (Advanced Linux Sound Architecture)
- Driver部分包括:
  - sigma-delta driver, R2R driver (internal sub-module)
  - AIC23, tw2866 driver (external)
- 使用alsa-lib提供上层APP 接口
- Plug-in
  - A-LAW,
  - Mu-LAW,
  - G.722, G.726



Audio Middleware

- 支持ALSA-LIB APIs

Functions	Descriptions
snd_pcm_open	Open a pcm device
snd_pcm_hw_params_alloca	Allocate a hardware parameters object.
snd_pcm_hw_params_any	Fill it in with default values
snd_pcm_hw_params_set_access	Set Access mode INTERLEAVED , NONINTERLEAVED , etc.
snd_pcm_hw_params_set_format	Set Format
snd_pcm_hw_params_set_channels	Set channel
snd_pcm_hw_params_set_rate	Set sample rate
snd_pcm_hw_params_set_period_size	Set Period Size (frame number)
snd_pcm_hw_params	Write the parameters to the driver
snd_pcm_readi	Read frames to the buffer
snd_pcm_writei	Write frames from the buffer
snd_pcm_drain	Change the pcm state to standby
snd_pcm_close	Close the pcm device

SONIX Audio Middleware APIs

Function Name	Descriptions
int snx_audio_mic_vol_get_items(int card_num, int *items)	Get the items of microphone volume adjustment.
int snx_audio_mic_vol_set(int card_num, int vol)	Set the volume of microphone.
int snx_audio_mic_vol_get(int card_num, int *vol)	Get the volume of microphone.
int snx_audio_mic_vol_set_mute(int card_num, int mute)	Set the microphone mute.
int snx_audio_mic_vol_get_mute(int card_num, int *mute)	Determine whether the microphone is mute.
int snx_audio_spk_vol_get_items(int card_num, int *items)	Get the items of speaker volume adjustment.
int snx_audio_spk_vol_set(int card_num, int vol)	Set the volume of speaker.
int snx_audio_spk_vol_get(int card_num, int *vol)	Get the volume of speaker.
int snx_audio_spk_vol_set_mute(int card_num, int mute)	Set the speaker mute.
int snx_audio_spk_vol_get_mute(int card_num, int *mute)	Determine whether the speaker is mute.

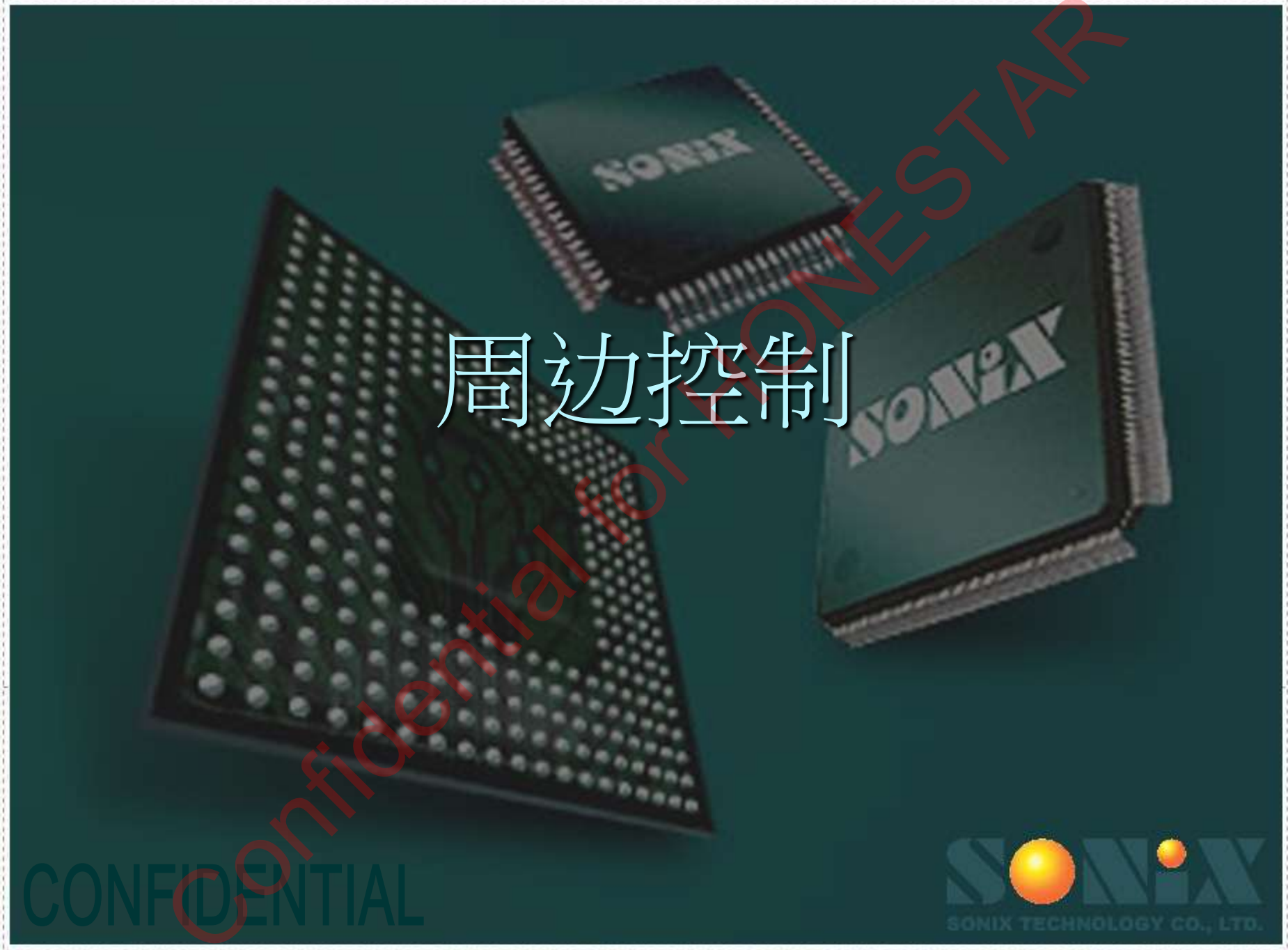
Audio Codec Plug-in

- **ALSA lib 支持物理和虚拟的设备创建**
  - **/usr/share/alsa/alsa.conf**
- 目前支持的设备

Device Name	Descriptions
hw:0	Hardware audio codec (PCM, A-LAW)
snx_audio_pcm	Hardware audio codec (PCM)
snx_audio_alaw	Software A-LAW codec
snx_audio_mulaw	Software Mu-LAW codec
snx_audio_g722	Software G.722 codec
snx_audio_g726	Software G.726 codec

Audio Test Tools and Example

- Aplay and Arecord
  - ALSA official utilities
- Example/audio
  - snx_audio_record
  - snx_audio_playback
  - snx_audio_vol_ctl



周边控制

SONIX TECHNOLOGY CO., LTD.

GPIO

- 4 个通用GPIO for SN98600 / 98601
- 6个通用GPIO for SN98610
- snx_gpio.ko
- 通用GPIO 支持
  - GPIO 方向设置
  - GPIO input/output value set /get
  - GPIO interrupt request
  - 可透过/sys/class/gpio读写操作
- Pin muxing GPIO
  - MS-1 (11 pins)
  - PWM (2 pins)
  - SPI (4 pins)

GPIO 设备操作

Function Name	Description
/sys/class/gpio/export	export control of a GPIO to user space
/sys/class/gpio/unexport	unexport control of a GPIO to user space
/sys/class/gpio/gpioN/direction	Set/get gpio pin direction. Read/write as either “in” or “out”
/sys/class/gpio/gpioN/value	Set/get gpio value. Read/write as either 0 (low) or 1 (high)
/sys/class/gpio/gpioN/edge	set interrupt edge none/rasing/falling/both

GPIO Middleware

- 整合GPIO 设备操作
  - 提供APIs
- 提供PWM, SPI, MS-1等 pin-muxing GPIO APIs

GPIMO Middleware APIs

General-purpose gpio function	description
snx_gpio_open	Open gpio device (export to user space)
snx_gpio_close	Close gpio device (unexport)
snx_gpio_write	Set gpio mode and value
snx_gpio_read	Read gpio mode and value
snx_gpio_set_interrupt	Set gpio interrupt trigger method
snx_gpio_poll	GPIO polling
ms1 gpio function	**description**
snx_ms1_gpio_open	Open ms-1 gpio device
snx_ms1_gpio_close	Close ms-1 gpio device
snx_ms1_gpio_write	Set gpio mode and value
snx_ms1_gpio_read	Read gpio mode and value

GPIMO Middleware APIs

spi gpio function	description
snx_spi_gpio_open	Open spi gpio device
snx_spi_gpio_close	Close spi gpio device
snx_spi_gpio_write	Set gpio mode and value
snx_spi_gpio_read	Read gpio mode and value
pwmgpio function	**description**
snx_pwm_gpio_open	Open pwm gpio device
snx_pwm_gpio_close	Close pwm gpio device
snx_pwm_gpio_write	Set gpio mode and value
snx_pwm_gpio_read	Read gpio mode and value

GPIO Middleware Library

- 支持各GPIO以及pin muxing gpio mode的定义设定
  - 不支持pin mux动态调整，先在板级设定好
  - /board-info/gpio/sn986xx/gpio.txt

```
 8  */
 9  GPIO Initial Setting
10  ---
11  PIN                         PINMUX  MODE            OUTPUTVALUE         Interrupt
12  ===
13  GPIO_00 ------------------ ENABLE  OUTPUT  ------ 1 ------------------- EDGE_SINGLE_RISE
14  GPIO_01 ------------------ ENABLE  OUTPUT  ------ 1 ------------------- DISABLE
15  GPIO_02 ------------------ ENABLE  OUTPUT  ------ 1 ------------------- EDGE_BOTH
16  GPIO_03 ------------------ ENABLE  OUTPUT  ------ 1 ------------------- DISABLE
17  GPIO_04 ------------------ ENABLE  OUTPUT  ------ 1 ------------------- EDGE_SINGLE_FALL
18  GPIO_05 ------------------ ENABLE  OUTPUT  ------ 1 ------------------- EDGE_BOTH
```

  - 透过gpio_init去套用设定

- 详细APIs 使用方式请参考
  - middleware/gpio/src/gpio-init/
  - middleware/gpio/src/gpio-led
  - app/example/src/gpio

I2C

- SN98600/98601 支持 1个I2C
- SN98610支持2个I2C
- 支持标准Linux I2C接口
- The I2C features include:
  - 支持100Khz and 400KHz
  - 支持7-bit, 10-bit, and general call addressing modes
  - Linux I2C sub-system

I2C Programing

- Kernel Space API

i2c_transfer	execute a single or combined I2C message

- User Space (IOCTL)

I2C_RDWR	ioctl cmd	Combined R/W transfer

/dev/i2c-0

- Example 请参考
  - app/example/src/i2c

SPI

- 2 SPI instance for SN98610
- 4-wire serial bus
- Feature
  - 支持SPI master / SPI slave / GPIO mode
  - 支持read/write/write_read operation
  - Programmable bits per word
  - Programmable max speed
  - Programmable serial bit data sequence (MSB or LSB first)
  - 支持 标准Linux SPI dev 接口
  - /dev/spidev0.0
  - /dev/spidev1.0 (SN98610)

SPI Programming

- Kernel space
  - spi_message_init
  - spi_message_add_tail
  - spi_sync

SPI Programming

User Space

Function Name	Category	Description
SPI_IOC_RD_MODE	ioctl	Get the current SPI mode
SPI_IOC_WR_MODE	ioctl	Set SPI mode
SPI_IOC_RD_LSB_FIRST	ioctl	Get the transmission setting (LSB first or not)
SPI_IOC_WR_LSB_FIRST	ioctl	Set LSB first or not in transmission
SPI_IOC_RD_BITS_PER_WORD	ioctl	Get how many bits per word for this device
SPI_IOC_WR_BITS_PER_WORD	ioctl	Set how many bits per word for this device
SPI_IOC_RD_MAX_SPEED_HZ	ioctl	Get the max speed of the spi device
SPI_IOC_WR_MAX_SPEED_HZ	ioctl	Set the max speed of the spi device
SPI_IOC_MESSAGE(num)	ioctl	SPI operation (num: numbers of the transmissions)
read	io	Read operation to SPI device
write	io	Write operation to SPI device

SPI Programming

- Example 请参考
  - app/example/src/spi

PWM

- 2 PWM pins output
- Driver snx_pwm.ko
- 预设节点
  - /dev/pwm
- 可调整的period , duty 时间

PWM Programming

Function Name	Category	Description
SONIX_PWM_REQUEST	ioctl cmd	equest a usable pwm device for pwm fd, the arg must be pwm_ID_1 or pwm_ID_2. these pwm_ID is defined in the struct of SNX_PWM_ID
SONIX_PWM_FREE	ioctl cmd	free a usable pwm device
SONIX_PWM_ENABLE	ioctl cmd	Enable pwm device
SONIX_PWM_DISABLE	ioctl cmd	disable pwm device
SONIX_PWM_READ	ioctl cmd	use pwm read mode and read the value
SONIX_PWM_INVERSE	ioctl cmd	inverse the square wave of pwm output
SONIX_PWM_CONFIG	ioctl cmd	config pwm device duty and period time, the arg is struct of pwm_config_param, (defined in snx_pwm.h)

PWM Programming

- Example 请参考
  - app/example/src/pwm
  - middleware/gpio/src/pwm-period

AES/DES/3DES/CRC

- 共享的IP
- 功能不能同时使用
- CRC16
  - 计算方程 $X^{16}+X^{15}+X^{2}+1$
  - /dev/crypto

- AES/DES/3DES
  - Cipher key only with lengths of 128 bits (AES-128)
  - Supports ECB mode
  - Supports DMA function
  - /dev/crypto

CRC16 Programming

- Kernel Space
  - snx_crypto_buffer_alloc
  - snx_crc16_calculate
- User Space

Function Name	Category	Description
SNX_INIT_BUF	ioctl cmd	Initial the buffer for CRC calcuation
SNX_CRC_CALCULATE	ioctl cmd	Enable the calculate

AES/DES/3DES Programming

- Kernel Space
  - snx_crypto_buffer_alloc
  - snx_crypto
- User Space

Function Name	Category	Description
SNX_INIT_BUF	Ioctl cmd	Initial buffer
SNX_CRYPTO_CRYPT	Ioctl cmd	Enable the encrypt/decrypt, use struct snx_cipher_info (define in snx_cipher.h)

AES/DES/3DES/CRC Programming

- Example 请参考
  - app/example/src/crypto

WDT/Timer/RTC

- WDT
  - /dev/watchdog
  - 请参考
    - app/example/src/watchdog/snx_watchdog_test.c
- Timer
  - 请参考
    - app/example/src/timer
- RTC
  - /dev/rtc
  - snx_rtc.ko
  - 请参考
    - app/example/rtc/ snx_rtc_test.c

Reference

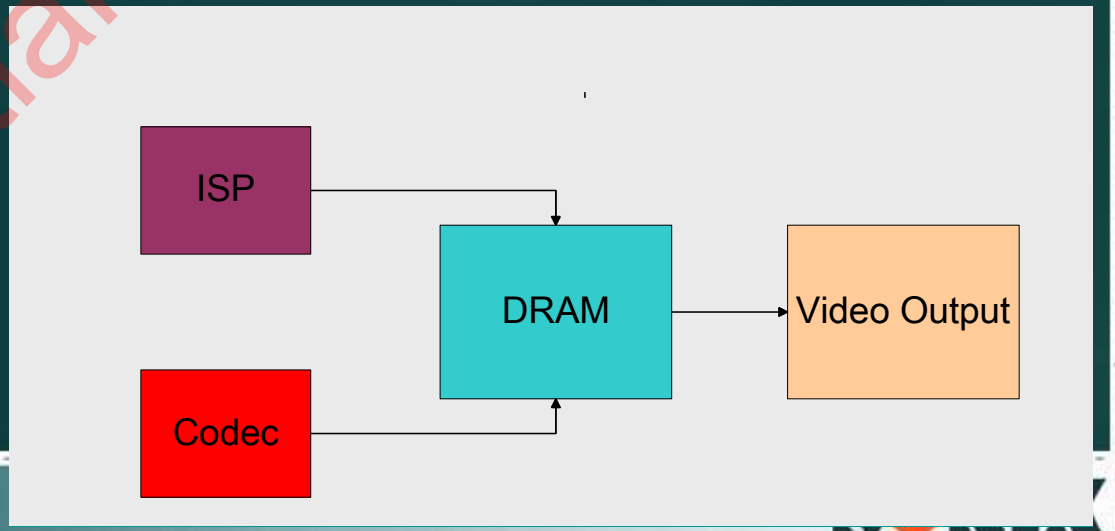
- Documents
  - SN986 Series Audio Codec Programing Guide
  - SN986 Series Video Codec Programing Guide
  - SN986 Series Video Output Programing Guide
  - SN986 Series SDK Programming Guide
- Example code
  - App/example

Video Output Programming

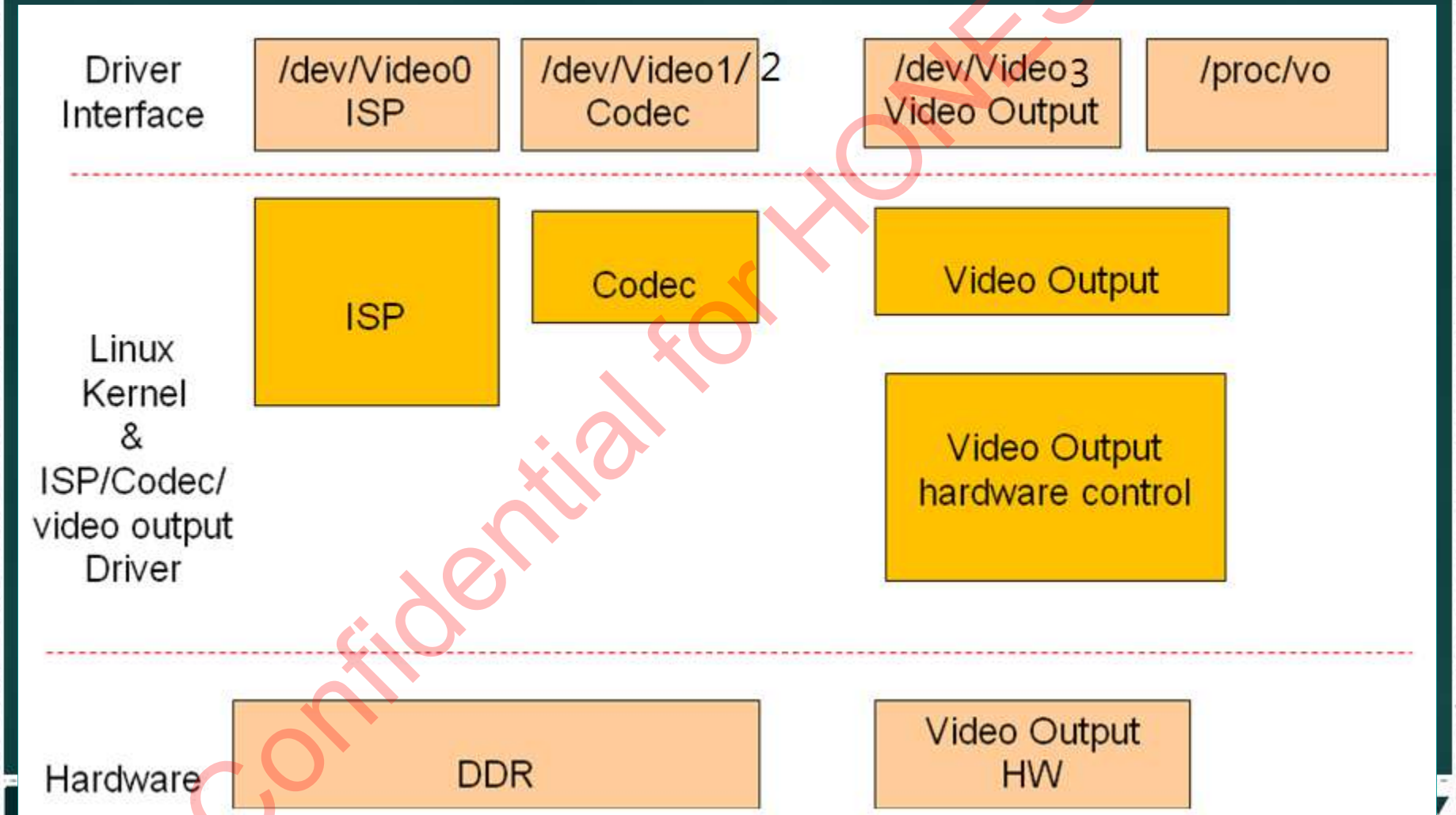
SONIX TECHNOLOGY CO., LTD.

Video Output

- SN98600 / SN98610 support TV output
- SN98610 support digital video output
- 支持ISP / Code source image
- 支持OSD, crop, scaling 功能
- snx_vo.ko



Video Output 架构

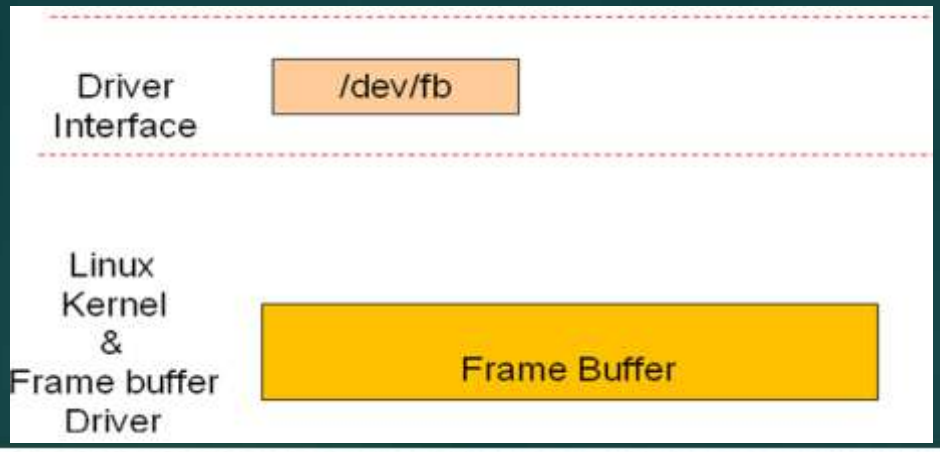


Video Output Programming

Name	Descriptions
VIDIOC_QUERYCAP	Query device capabilities
VIDIOC_S_OUTPUT	select the current video output
VIDIOC_CROPCAP	Information about the video cropping and scaling abilities
VIDIOC_S_CROP	set the current cropping rectangle
VIDIOC_S_FMT	set the data format
VIDIOC_REQBUFS	Initiate Memory Mapping or User Pointer I/O
VIDIOC_QUERYBUF	Query the status of a buffer
VIDIOC_QBUF	Exchange a buffer with the driver
VIDIOC_DQBUF	Exchange a buffer with the driver
VIDIOC_STREAMON	Start streaming I/O
VIDIOC_STREAMOFF	Stop streaming I/O

OSD

- 支持255 color and 1 transparent color
- 使用frame buffer device
- dev/fb0



OSD programming

Name	Descriptions
FBIOGET_VSCREENINFO	gets the variable screen information from the framebuffer, things like resolution, margins, color properties.
FBIOGET_FSCREENINFO	gets the fixed information about a framebuffer, such as the start and length of framebuffer memory, line length

- 透过ioctl 取得framebuffer info 然后使用mmap将更新的内容写入
- 详细请参考
  - SN986 Series Video Output Programing Guide

Video Output Programming

- Example :
  - App/example/video_output