Amlogic

How to add an new resolution

Revision: 02

Release Date: 2020-01-08

Copyright

© 2018 Amlogic. All rights reserved. No part of this document may be reproduced. Transmitted, transcribed, or translated into any language in any form or by any means with the written permission of Amlogic.

Trademarks

Amlogic, and other Amlogic icons are trademarks of Amlogic companies. All other trademarks and registered trademarks are property of their respective companies.

Disclaimer

Amlogic may make improvements and/or changes in this document or in the product described in this document at any time.

This product is not intended for use in medical, life saving, or life sustaining applications.

Circuit diagrams and other information relating to products of Amlogic are included as a means or illustrating typical applications. Consequently, complete information sufficient for production design is not necessarily given. Amlogic makes no representations or warranties with respect to the accuracy or completeness of the contents presented in this document.

Contact Information

- Website: www.amlogic.com
- Pre-sales consultation: contact@amlogic.com
- Technical support: support@amlogic.com

Copyright © Amlogic. All rights reserved.

Revision History

Section	Change Description
copyright	
Page footer	
Revision History	2020.01.08 zhengrong.zhu v1

Copyright © Amlogic. All rights reserved.

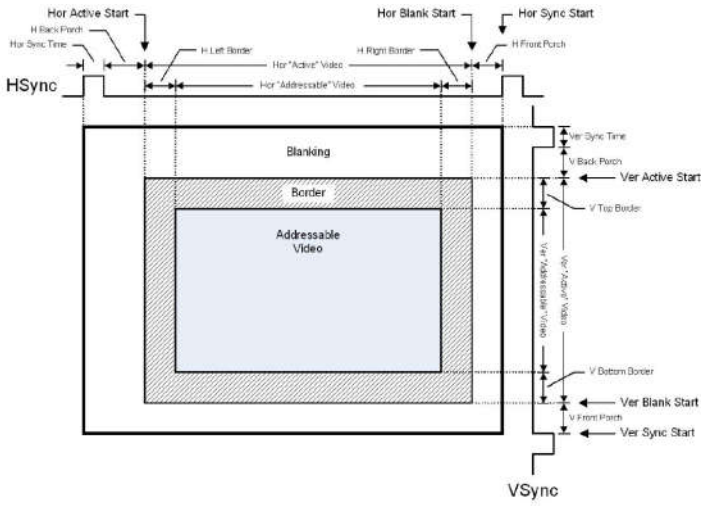
Contents

Copyright © Amlogic. All rights reserved.

1.添加自定义分辨率

一、核实分辨率参数

要添加自定义分辨率，首先需要核实客户提供的分辨率参数，包括像素时钟(Pixel Clock)，水平可见宽度(H Active)，水平总长(H Total)，水平前沿(H FPorch)，水平同步宽度(H Sync)，水平后沿(H BPorch)，垂直可见高度(V Active)，垂直总长(V Total)，垂直前沿(V FPorch)，垂直同步宽度(V Sync)，垂直后沿(V BPorch)。详细的参数描述如下图，其中 Border 为很久以前的概念，现在的分辨率 Border 全是为 0 的。



上述参数满足如下等式：

H Total = H Active + H FPorch + H Sync + H BPorch

V Total = V Active + V FPorch + V Sync + V BPorch

H Blank = H FPorch + H Sync + H BPorch = H Total - H Active

V Blank = V FPorch + V Sync + V BPorch = V Total - V Active

另外水平和垂直频率的计算方法是：

H Freq = Pixel Clock / H Total

V Freq = Pixel Clock / H Total / V Total = H Freq / V Total

Copyright © Amlogic. All rights reserved.

客户提供的参数必须满足上述条件，如不满足要和客户沟通好。

本文档中我们以客户要求的 3440x1440p60hz 为例，详细说明其配置方法。客户提供的详细参数如下：

H Active：3440，H Total：3600，H FPorch: 48，H Sync: 32，H BPorch: 80

V Active：1440，V Total：1481，V FPorch: 3，V Sync: 10，V BPorch: 28

Pixel Clock: 319750000Hz

二、在 Uboot 里面添加自定义分辨率按如下说明进行：

为了开机后能正常按照关机前的分辨率显示，必须在 Uboot 里面必须添加自定义分辨率相关的代码。

1. 添加 PLL 参数

芯片内部通过 PLL 产生像素时钟，因此需要计算 PLL 参数。参考《g12a_plls_application_note 2.8.1》，设置 PLL 产生像素时钟的方法如下。

$$\text{Target frequency} = 24MHz \bullet \frac{DPLL\_M + \frac{DIV\_FRAC}{2^{17}}}{DPLL\_N} \bullet \frac{1}{OD}$$

其中 $24\ MHz \bullet \frac{DPLL\_M + \frac{DIV\_FRAC}{2^{17}}}{DPLL\_N}$ 为 **DCO 的输出频率**，Target frequency 为像素时钟乘以 10，DPLL_M 为 8bit 无符号整数，DPLL_N 为 5bit 无符号整数，DIV_FRAC 为 19bit 无符号整数。

要确保 **DCO 的输出频率**在 3G 和 6G 之间，当 Target frequency 不足 3G 时可以调节 OD，使之满足条件。DPLL_M，DPLL_N，DIV_FRAC 和 OD 在 HHI_GP0_PLL_CNTL0 和 HHI_GP0_PLL_CNTL1 寄存器中。在实际的计算中，Target frequency 以 KHz 为单位。

Copyright © Amlogic. All rights reserved.

需要修改的文件、函数或变量如下表：

文件	函数或变量
arch/arm/cpu/armv8/g12a/hdmitx20/enc_clk_config.c	修改 set_hpll_clk_out
arch/arm/cpu/armv8/g12a/hdmitx20/enc_clk_config.c	修改 setting_enc_clk_val_24
arch/arm/cpu/armv8/g12b/hdmitx20/enc_clk_config.c	修改 set_hpll_clk_out
arch/arm/cpu/armv8/g12b/hdmitx20/enc_clk_config.c	修改 setting_enc_clk_val_24
arch/arm/cpu/armv8/tm2/hdmitx20/enc_clk_config.c	修改 set_hpll_clk_out
arch/arm/cpu/armv8/tm2/hdmitx20/enc_clk_config.c	修改 setting_enc_clk_val_24

set_hpll_clk_out 函数中添加配置 P_HHI_HDMI_PLL_CNTL0/1/2/3/4/5/6 的代码，相关寄存器说明可参考《G12A-HIU-Registers.docx》，至于非 DPLL_M，DPLL_N，DIV_FRAC，OD 的字段，不知道准确的理论根据，按照最接近的 Target frequency Copy 即可。set_hpll_clk_out 函数中并没有对 OD 进行配置，后面的代码会配置的。示例代码如下，注意 3197500 是 **DCO 的输出频率**，只有当 OD 等于 1 时，它才等于像素时钟。

```
case 3197500:
    hd_write_reg(P_HHI_HDMI_PLL_CNTL0, 0x3b000485);  0x85就是DPLL_M,
    hd_write_reg(P_HHI_HDMI_PLL_CNTL1, 0x00007555);  0x04>>2后就是
    hd_write_reg(P_HHI_HDMI_PLL_CNTL2, 0x00000000);
    hd_write_reg(P_HHI_HDMI_PLL_CNTL3, 0x0a691c00);  DPLL_N,
    hd_write_reg(P_HHI_HDMI_PLL_CNTL4, 0x33771290);  0x7555就是DIV_FRAC
    hd_write_reg(P_HHI_HDMI_PLL_CNTL5, 0x39270000);
    hd_write_reg(P_HHI_HDMI_PLL_CNTL6, 0x50540000);
    hd_set_reg_bits(P_HHI_HDMI_PLL_CNTL0, 0x0, 29, 1);
    WAIT_FOR_PLL_LOCKED(P_HHI_HDMI_PLL_CNTL0);
    pr_info("HPLL: 0x%lx\n", hd_read_reg(P_HHI_HDMI_PLL_CNTL0));
    break;
```


Copyright © Amlogic. All rights reserved.

24*1000*(133+30037/(2^17))=3197499，与需要的 3197500 相差无几。

在 setting_enc_clk_val_24 结构体数组中添加一个元素，该结构体的 hpll_clk_out 变量为 $24MHz \bullet \frac{DPLL\_M + \frac{DIV\_FRAC}{2^{17}}}{DPLL\_N}$，od1/2/3 的乘积为 OD 值。示例代码如下：

```
{
    {
        HDMIV_3440x1440p60hz, GROUP_END
    },
    1, VIU_ENCP, 3197500, 1, 1, 1, CLK_UTIL_VID_PLL_DIV_5, 2, 1, 1, -1
},
```

2. 添加支持模式

需要修改的文件、函数或变量如下表：

文件	函数或变量
arch/arm/cpu/armv8/g12a/hdmitx20/hdmitx_set.c	修改 gxbb_modes
arch/arm/cpu/armv8/g12b/hdmitx20/hdmitx_set.c	修改 gxbb_modes
arch/arm/cpu/armv8/tm2/hdmitx20/hdmitx_set.c	修改 gxbb_modes

在 gxbb_modes 结构体数组中添加 VIC 宏定义和模式名称字符串。示例代码如下：

```
{HDMIV_3440x1440p60hz, "3440x1440p60hz", 0},
```


Copyright © Amlogic. All rights reserved.

3. 添加 timing 参数

需要修改的文件、函数或变量如下表：

文件	函数或变量
arch/arm/cpu/armv8/g12a/hdmitx20/hdmitx_tvenc.c	新增 tvregs_vesa_3440x1440p60hz
arch/arm/cpu/armv8/g12a/hdmitx20/hdmitx_tvenc.c	修改 tvregsTab
arch/arm/cpu/armv8/g12b/hdmitx20/hdmitx_tvenc.c	新增 tvregs_vesa_3440x1440p60hz
arch/arm/cpu/armv8/g12b/hdmitx20/hdmitx_tvenc.c	修改 tvregsTab
arch/arm/cpu/armv8/tm2/hdmitx20/hdmitx_tvenc.c	新增 tvregs_vesa_3440x1440p60hz
arch/arm/cpu/armv8/tm2/hdmitx20/hdmitx_tvenc.c	修改 tvregsTab

新增类型为 static const struct reg_t 结构体的全局变量，同时在 tvregsTab 数组中添加刚定义的变量。

static const struct reg_t 结构体全局变量的值可以通过_ENC_g9tv_4k1k_4k05k.xls 计算，使用说明如下图。 

Copyright © Amlogic. All rights reserved.

					ok		debug				
ENCP_VIDEO_MODE	0x1b8d	0x4040	红色方框内的最右列就是计算出的数据						Hacitve	3440	3440
ENCP_VIDEO_MODE_ADV	0x1b8e	0x18							Vactive	1440	1440
ENCP_VIDEO_MAX_PXCNT	0x1b97	0xE0F	3599	**2200** - 1	3599	E0F	3599	0	Htotal	3600	3599
ENCP_VIDEO_MAX_LNCNT	0x1bae	0x5C8	1480	**1125** – 1	1480	5C8	1480	0	Hblank	160	
ENCP_VIDEO_HAVON_BEGIN	0x1ba4	0x70	112	**148**	192	C0	192	E0F	Vtotal	1481	1480
ENCP_VIDEO_HAVON_END	0x1ba3	0xDDF	3551	2067 – 148 = **1920** – 1	2111	83F	2111	5C8	Vblank	41	
ENCP_VIDEO_VAVON_BLINE	0x1ba6	0x26	38	**41**	41	29	41	C0	Hfront	48	48
ENCP_VIDEO_VAVON_ELINE	0x1baf	0x5C5	1477	1120 – 41 = **1080** – 1	1120	460	1120	83F	Hsync	32	32
ENCP_VIDEO_HSO_BEGIN	0x1ba7	0x0	0		0	0	0	29	Hback	80	80
ENCP_VIDEO_HSO_END	0x1ba8	0x20	32		44	2C	44	460	Vfront	3	3
ENCP_VIDEO_VSO_BEGIN	0x1ba9	0x1E	30		0	0	0		Vsync	10	10
ENCP_VIDEO_VSO_END	0x1baa	0x32	50		2199	897	2199	0	Vback	28	28
ENCP_VIDEO_VSO_BLINE	0x1bab	0x0	0	**0**	0	0	0	2C			
ENCP_VIDEO_VSO_ELINE	0x1bac	0xA	10	**5**	5	5	5	0			
ENCP_DVI_HSO_BEGIN	0x1c30	0x2	2	2200 – 2158 + 2 = **44**	2	2	2	897	根据客户的参数		
ENCP_DVI_HSO_END	0x1c31	0x22	34		46	2E	46	0	填写上面绿色方		
ENCP_DVI_VSO_BLINE_EVN	0x1c32	0x0	0		0	0	0	5			
ENCP_DVI_VSO_BLINE_ODD	0x1c33	0x0	0	1125 – 1124 + 4 = **5**					框内的数值		
ENCP_DVI_VSO_ELINE_EVN	0x1c34	0xA	10		5	5	5	2			
ENCP_DVI_VSO_ELINE_ODD	0x1c35	0x0	0					2E			
ENCP_DVI_VSO_BEGIN_EVN	0x1c36	0x2	2		2	2	2	0			
ENCP_DVI_VSO_BEGIN_ODD	0x1c37	0x0	0								
ENCP_DVI_VSO_END_EVN	0x1c38	0x2	2		2	2	2	5			
ENCP_DVI_VSO_END_ODD	0x1c39	0x0	0								
ENCP_DE_H_BEGIN	0x1c3a	0x72	114	2070 – 150 = **1920**	194	C2	194	2			
ENCP_DE_H_END	0x1c3b	0xDE2	3554		2114	842	2114				
ENCP_DE_V_BEGIN_EVEN	0x1c3c	0x26	38	1121 – 41 = **1080**	41	29	41	2			
ENCP_DE_V_END_EVEN	0x1c3d	0x5C6	1478		1121	461	1121				
ENCP_DE_V_BEGIN_ODD	0x1c3e	0x0	0					C2			
ENCP_DE_V_END_ODD	0x1c3f	0x0	0					842			
VPU_HDMI_SETTING	0x271b							461			
VPP_POSTBLEND_H_SIZE	0x1d21	0xD70	3439								

程序中的示例代码如下，其中 P_ENCP_VIDEO_EN，P_ENCI_VIDEO_EN，MREG_END_MARKER 的数据来源不清楚，Copy 其它逐行扫描的分辨率都是 0。

```
static const struct reg_t tvregs_vesa_3440x1440p60hz[] = {
    {P_ENCP_VIDEO_EN, 0,},
    {P_ENCI_VIDEO_EN, 0,},

    {P_ENCP_VIDEO_MODE, 0x4040,},
    {P_ENCP_VIDEO_MODE_ADV, 0x18,},
    {P_ENCP_VIDEO_MAX_PXCNT, 0xE0F,},
    {P_ENCP_VIDEO_MAX_LNCNT, 0x5C8,},
    {P_ENCP_VIDEO_HAVON_BEGIN, 0x70,},
    {P_ENCP_VIDEO_HAVON_END, 0xDDF,},
    {P_ENCP_VIDEO_VAVON_BLINE, 0x26,},
    {P_ENCP_VIDEO_VAVON_ELINE, 0x5C5,},
    {P_ENCP_VIDEO_HSO_BEGIN, 0x0,},
    {P_ENCP_VIDEO_HSO_END, 0x20,},
    {P_ENCP_VIDEO_VSO_BEGIN, 0x1E,},
    {P_ENCP_VIDEO_VSO_END, 0x32,},
    {P_ENCP_VIDEO_VSO_BLINE, 0x0,},
    {P_ENCP_VIDEO_VSO_ELINE, 0xA,},

    {P_ENCI_VIDEO_EN, 0},
    {MREG_END_MARKER, 0}
};



  {HDMIV_3440x1440p60hz, tvregs_vesa_3440x1440p60hz},
```

4. 添加 mode 的详细信息

需要修改的文件、函数或变量如下表：

文件	函数或变量
common/hdmi_parameters.c	新增 fmt_para_vesa_3440x1440p60_43x18
common/hdmi_parameters.c	修改 all_fmt_paras

在 common/hdmi_parameters.c 文件中添加类型为 static struct hdmi_format_para 结构体的全局变量，同时在 all_fmt_paras 数组中添加刚定义的变量。示例代码如下：

Copyright © Amlogic. All rights reserved.

```
static struct hdmi_format_para fmt_para_vesa_3440x1440p60_43x18 = {
    .vic = HDMIV_3440x1440p60hz,
    .name = "3440x1440p60hz",
    .sname = "3440x1440p60hz",
    .pixel_repetition_factor = 0,
    .progress_mode = 1,
    .scrambler_en = 0,
    .tmds_clk_div40 = 0,
    .tmds_clk = 319750,
    .timing = {
        .pixel_freq = 319750,
        .h_freq = 88819,
        .v_freq = 59973,
        .vsync_polarity = 1, /* +VSync */
        .hsync_polarity = 1, /* +HSync */
        .h_active = 3440,
        .h_total = 3600,
        .h_blank = 160,
        .h_front = 48,
        .h_sync = 32,
        .h_back = 80,
        .v_active = 1440,
        .v_total = 1481,
        .v_blank = 41,
        .v_front = 3,
        .v_sync = 10,
        .v_back = 28,
        .v_sync_ln = 1,
    },
};
```

```
    &fmt_para_vesa_3440x1440p60_43x18,
```

5. 添加 VIC 编号

需要修改的文件、函数或变量如下表：

文件	函数或变量
include/amlogic/hdmi.h	修改 enum hdmi_vic

在 enum hdmi_vic 枚举中添加一个成员。示例代码如下：

```
    HDMIV_3440x1440p60hz,
```

三、在 Kernel 里面添加自定义分辨率按如下说明进行：

1. 添加 PLL 参数

需要修改的文件、函数或变量如下表：

文件	函数或变量
drivers/amlogic/media/vout/hdmitx/hdmi_tx_20/hw/hw_g12a.c	修改 set_g12a_hpll_clk_out
drivers/amlogic/media/vout/hdmitx/hdmi_tx_20/hw/hw_clk.c	修改 setting_enc_clk_val_24

计算 PLL 参数的方法与 uboot 里面相同，在 set_g12a_hpll_clk_out 函数中做与 uboot 相同的添加即可。示例代码如下：

Copyright © Amlogic. All rights reserved.

```
case 3197500:
    hd_write_reg(P_HHI_HDMI_PLL_CNTL0, 0x3b000485);
    hd_write_reg(P_HHI_HDMI_PLL_CNTL1, 0x00007555);
    hd_write_reg(P_HHI_HDMI_PLL_CNTL2, 0x00000000);
    hd_write_reg(P_HHI_HDMI_PLL_CNTL3, 0x0a691c00);
    hd_write_reg(P_HHI_HDMI_PLL_CNTL4, 0x33771290);
    hd_write_reg(P_HHI_HDMI_PLL_CNTL5, 0x39270000);
    hd_write_reg(P_HHI_HDMI_PLL_CNTL6, 0x50540000);
    hd_set_reg_bits(P_HHI_HDMI_PLL_CNTL0, 0x0, 29, 1);
    WAIT_FOR_PLL_LOCKED(P_HHI_HDMI_PLL_CNTL0);
    pr_info("HPLL: 0x%lx\n", hd_read_reg(P_HHI_HDMI_PLL_CNTL0));
    break;
```

0x85就是DPLL_M,
0x04>>2后就是
DPLL_N,
0x7555就是DIV_FRAC

在 setting_enc_clk_val_24 结构体数组中添加一个元素，与 uboot 中的添加相同。示例代码如下：

```
{{HDMIV_3440x1440p60hz,
  HDMI_VIC_END},
    3197500, 1, 1, 1, VID_PLL_DIV_5, 2, 1, 1, -1},
```

2. 添加色域，色深，色彩空间等信息

需要修改的文件、函数或变量如下表：

文件	函数或变量
drivers/amlogic/media/vout/hdmitx/hdmi_tx_20/hdmi_tx_video.c	修改 hdmi_tx_video_params

在文件的 hdmi_tx_video_params 结构体数组中添加一个元素。示例代码如下：

```
{
    .VIC        = HDMIV_3440x1440p60hz,
    .color_prefer    = COLORSPACE_RGB444,
    .color_depth     = COLORDEPTH_24B,
    .bar_info    = B_UNVALID,
    .repeat_time     = NO_REPEAT,
    .aspect_ratio    = ASPECT_RATIO_SAME_AS_SOURCE,
    .cc      = CC_NO_DATA,
    .ss      = SS_NO_DATA,
    .sc      = SC_NO_UINFORM,
},
```

3. 添加 timing 参数

需要修改的文件、函数或变量如下表：

文件	函数或变量
drivers/amlogic/media/vout/hdmitx/hdmi_tx_20/hw/enc_cfg_hw.c	新增 tvregs_vesa_3440x1440p60hz
drivers/amlogic/media/vout/hdmitx/hdmi_tx_20/hw/enc_cfg_hw.c	修改 tvregsTab

添加类型为 static const struct reg_s 结构体的全局变量，同时在 tvregsTab 数组中添加刚定义的变量。相关的数值与 uboot 里面相同。示例代码如下：

Copyright © Amlogic. All rights reserved.

```
static const struct reg_s tvregs_vesa_3440x1440p60hz[] = {
    {P_ENCP_VIDEO_EN, 0,},
    {P_ENCI_VIDEO_EN, 0,},
    {P_VENC_VDAC_SETTING, 0xff,},

    {P_ENCP_VIDEO_MODE, 0x4040,},
    {P_ENCP_VIDEO_MODE_ADV, 0x18,},
    {P_ENCP_VIDEO_MAX_PXCNT, 0xE0F,},
    {P_ENCP_VIDEO_MAX_LNCNT, 0x5C5,},
    {P_ENCP_VIDEO_HAVON_BEGIN, 0x70,},
    {P_ENCP_VIDEO_HAVON_END, 0xDDF,},
    {P_ENCP_VIDEO_VAVON_BLINE, 0x26,},
    {P_ENCP_VIDEO_VAVON_ELINE, 0x5C5,},
    {P_ENCP_VIDEO_HSO_BEGIN, 0x0,},
    {P_ENCP_VIDEO_HSO_END, 0x20,},
    {P_ENCP_VIDEO_VSO_BEGIN, 0x1E,},
    {P_ENCP_VIDEO_VSO_END, 0x32,},
    {P_ENCP_VIDEO_VSO_BLINE, 0x0,},
    {P_ENCP_VIDEO_VSO_ELINE, 0xA,},

    {P_VPU_VIU_VENC_MUX_CTRL, 0xA},
    {P_ENCI_VIDEO_EN, 0},
    {MREG_END_MARKER, 0}
};


{HDMIV_3440x1440p60hz, tvregs_vesa_3440x1440p60hz},
```

4. 添加 mode 的详细信息

需要修改的文件、函数或变量如下表：

文件	函数或变量
drivers/amlogic/media/vout/hdmitx/hdmi_common/hdmi_parameters.c	新增 fmt_para_vesa_3440x1440p60_43x18
drivers/amlogic/media/vout/hdmitx/hdmi_common/hdmi_parameters.c	修改 all_fmt_paras

添加类型为 static struct hdmi_format_para 结构体的全局变量，同时在 all_fmt_paras 数组中添加刚定义的变量。示例代码如下：

Copyright © Amlogic. All rights reserved.

How to add an new resolution

```
static struct hdmi_format_para fmt_para_vesa_3440x1440p60_43x18 = {
    .vic = HDMIV_3440x1440p60hz,
    .name = "3440x1440p60hz",
    .pixel_repetition_factor = 0,
    .progress_mode = 1,
    .scrambler_en = 0,
    .tmds_clk_div40 = 0,
    .tmds_clk = 319750,
    .timing = {
        .pixel_freq = 319750,
        .h_freq = 88819,
        .v_freq = 59973,
        .vsync_polarity = 1, /* +VSync */
        .hsync_polarity = 1, /* +HSync */
        .h_active = 3440,
        .h_total = 3600,
        .h_blank = 160,
        .h_front = 48,
        .h_sync = 32,
        .h_back = 80,
        .v_active = 1440,
        .v_total = 1481,
        .v_blank = 41,
        .v_front = 3,
        .v_sync = 10,
        .v_back = 28,
        .v_sync_ln = 1,
    },
    .hdmitx_vinfo = {
        .name               = "3440x1440p60hz",
        .mode               = VMODE_HDMI,
        .width              = 3440,
        .height             = 1440,
        .field_height       = 1440,
        .aspect_ratio_num   = 43,
        .aspect_ratio_den   = 18,
        .sync_duration_num  = 60,
        .sync_duration_den  = 1,
        .video_clk          = 319750000,
        .htotal             = 3600,
        .vtotal             = 1481,
        .fr_adj_type        = VOUT_FR_ADJ_HDMI,
        .viu_color_fmt      = COLOR_FMT_YUV444,
        .viu_mux            = VIU_MUX_ENCP,
    },
};
```


Copyright © Amlogic. All rights reserved.