

点屏

确认spec是否支持

因为ssd20x clk的限制，点panel前先确认panel的clk是否支持，ssd20x clk的spec如下：

TTL输出CLK支持范围9Mhz-75Mhz，

计算方法： $htotal * vtotal * fps$

MIPI DSI输出CLK支持范围100Mbps/lane - 1.5Gbps/lane

计算方法： $H_Total * V_Total * FPS * BitsPerPixel / lane\ number$

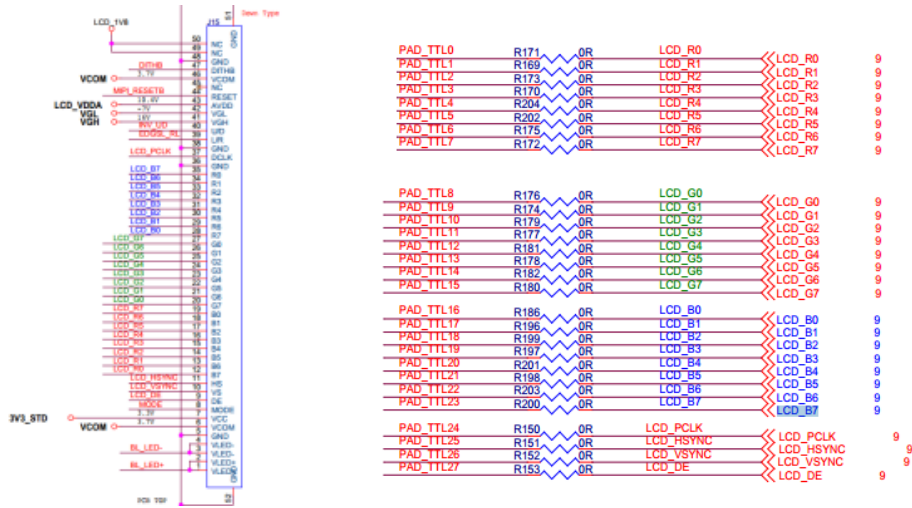
$BitsPerPixel = 24 (RGB888) / 18 (RGB666) / 16 (RGB565)$

Panel相关pin脚的配置

首先跟硬件确认好是用那组pin脚(最好是使用下文中所列的pin，这些pin我们都有验证过)，然后在kernel的dts里面切到对应的mode，

TTL

以公版为例，用的是PAD_TTL0~PAD_TTL27这组：

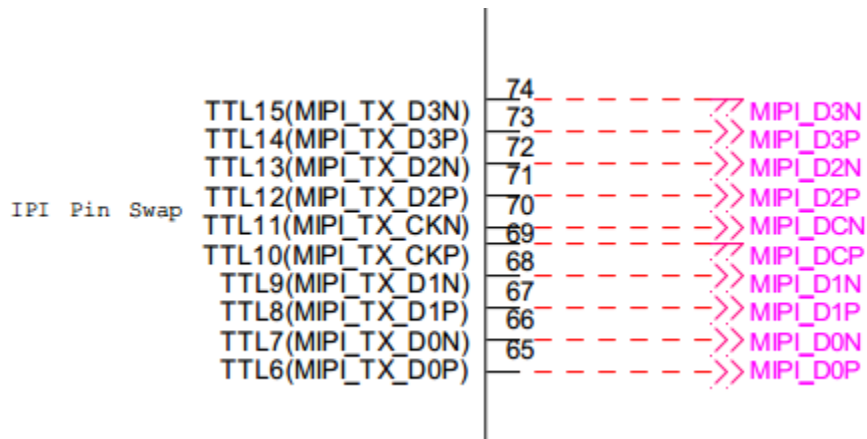


在infinity2m-ssc011a-s01a-padmux-display.dtsi中配置成PINMUX_FOR_TTL_MODE_1

```
<PAD_FUART_RTS> PINMUX_FOR_GPIO_MODE MDRV_FUSE_CPUFREQ_VID1>
<PAD_TTL0> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT0 >
<PAD_TTL1> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT1 >
<PAD_TTL2> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT2 >
<PAD_TTL3> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT3 >
<PAD_TTL4> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT4 >
<PAD_TTL5> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT5 >
<PAD_TTL6> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT6 >
<PAD_TTL7> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT7 >
<PAD_TTL8> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT8 >
<PAD_TTL9> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT9 >
<PAD_TTL10> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT10 >
<PAD_TTL11> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT11 >
<PAD_TTL12> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT12 >
<PAD_TTL13> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT13 >
<PAD_TTL14> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT14 >
<PAD_TTL15> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT15 >
<PAD_TTL16> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT16 >
<PAD_TTL17> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT17 >
<PAD_TTL18> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT18 >
<PAD_TTL19> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT19 >
<PAD_TTL20> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT20 >
<PAD_TTL21> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT21 >
<PAD_TTL22> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT22 >
<PAD_TTL23> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DOUT23 >
<PAD_TTL24> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_CLK >
<PAD_TTL25> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_HSYNC >
<PAD_TTL26> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_VSYNC >
<PAD_TTL27> PINMUX_FOR_TTL_MODE_1 MDRV_FUSE_TTL_DE >
<PAD_UART0_RX> PINMUX_FOR_UART0_MODE_1 MDRV_FUSE_UART0_RX >
<PAD_UART0_TX> PINMUX_FOR_UART0_MODE_1 MDRV_FUSE_UART0_TX >
```

MIPI

如下图所示，用的是PAD_TTL6~PAD_TTL15:



所以在padmux里面配置成PINMUX_FOR_TX_MIPI_MODE_1 mode:

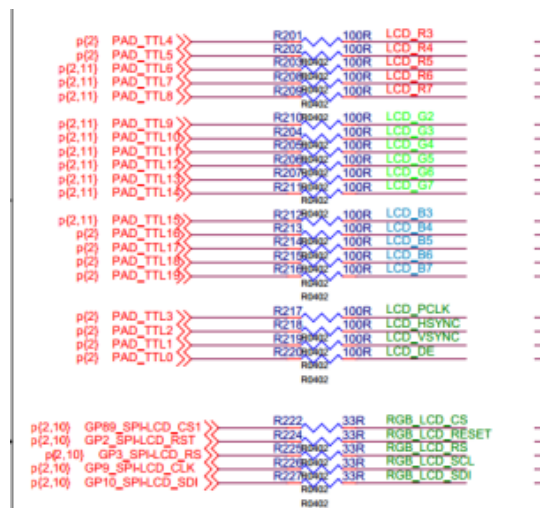
```

... //<PAD_GPIOT1 >,
... //<PAD_FUART_RX >,
... //<PAD_FUART_TX >, PINMUX_FOR_GPIO_MODE <<<< MDRV_PUSE_UTMI_POWER>,
... <PAD_FUART_CTS > PINMUX_FOR_GPIO_MODE <<<< MDRV_PUSE_CPUFREQ_VID0>,
... <PAD_FUART_RTS > PINMUX_FOR_GPIO_MODE <<<< MDRV_PUSE_CPUFREQ_VID1>,
... //<PAD_TTL0 > PINMUX_FOR_SDIO_MODE_1 <<<< MDRV_PUSE_SDIO_PWR >,
... <PAD_TTL6 > PINMUX_FOR_TX_MIPI_MODE_1 <<<< MDRV_PUSE_TX_MIPI_P_CH0 >,
... <PAD_TTL7 > PINMUX_FOR_TX_MIPI_MODE_1 <<<< MDRV_PUSE_TX_MIPI_N_CH0 >,
... <PAD_TTL8 > PINMUX_FOR_TX_MIPI_MODE_1 <<<< MDRV_PUSE_TX_MIPI_P_CH1 >,
... <PAD_TTL9 > PINMUX_FOR_TX_MIPI_MODE_1 <<<< MDRV_PUSE_TX_MIPI_N_CH1 >,
... <PAD_TTL10 > PINMUX_FOR_TX_MIPI_MODE_1 <<<< MDRV_PUSE_TX_MIPI_P_CH2 >,
... <PAD_TTL11 > PINMUX_FOR_TX_MIPI_MODE_1 <<<< MDRV_PUSE_TX_MIPI_N_CH2 >,
... <PAD_TTL12 > PINMUX_FOR_TX_MIPI_MODE_1 <<<< MDRV_PUSE_TX_MIPI_P_CH3 >,
... <PAD_TTL13 > PINMUX_FOR_TX_MIPI_MODE_1 <<<< MDRV_PUSE_TX_MIPI_N_CH3 >,
... <PAD_TTL14 > PINMUX_FOR_TX_MIPI_MODE_1 <<<< MDRV_PUSE_TX_MIPI_P_CH4 >,
... <PAD_TTL15 > PINMUX_FOR_TX_MIPI_MODE_1 <<<< MDRV_PUSE_TX_MIPI_N_CH4 >,
... <PAD_UART0_RX > PINMUX_FOR_UART0_MODE_1 <<<< MDRV_PUSE_UART0_RX >,
... <PAD_UART0_TX > PINMUX_FOR_UART0_MODE_1 <<<< MDRV_PUSE_UART0_TX >,
... //<PAD_UART1_RX >,
... //<PAD_UART1_TX >,
... //<PAD_SD_CLK > PINMUX_FOR_I2S_MODE_3 <<<< MDRV_PUSE_I2S_BCK >,
... <PAD_SD_CLK > PINMUX_FOR_I2S_MODE_3 <<<< MDRV_PUSE_I2S_CLK >,

```

RMI1

如下图所示，用的是PAD_TTL0~PAD_TTL19:



所以在padmux里面配置成PINMUX_FOR_TTL_MODE_10:

```

<PAD_FUART_C13> ..... PINMUX_FOR_I2C1_MODE_3 ..... MDRV_PUSE_I2C1_SCL>,
<PAD_FUART_RTS> ..... PINMUX_FOR_I2C1_MODE_3 ..... MDRV_PUSE_I2C1_SDA>,
<PAD_TTL0> ..... PINMUX_FOR_TTL_MODE_10 ..... MDRV_PUSE_TTL_DE >,
<PAD_TTL1> ..... PINMUX_FOR_TTL_MODE_10 ..... MDRV_PUSE_TTL_VSYNC >,
<PAD_TTL2> ..... PINMUX_FOR_TTL_MODE_10 ..... MDRV_PUSE_TTL_HSYNC >,
<PAD_TTL3> ..... PINMUX_FOR_TTL_MODE_10 ..... MDRV_PUSE_TTL_CLK >,
<PAD_TTL4> ..... PINMUX_FOR_TTL_MODE_10 ..... MDRV_PUSE_TTL_DOUT00 >,
<PAD_TTL5> ..... PINMUX_FOR_TTL_MODE_10 ..... MDRV_PUSE_TTL_DOUT01 >,
<PAD_TTL6> ..... PINMUX_FOR_TTL_MODE_10 ..... MDRV_PUSE_TTL_DOUT02 >,
<PAD_TTL7> ..... PINMUX_FOR_TTL_MODE_10 ..... MDRV_PUSE_TTL_DOUT03 >,
<PAD_TTL8> ..... PINMUX_FOR_TTL_MODE_10 ..... MDRV_PUSE_TTL_DOUT04 >,
<PAD_TTL9> ..... PINMUX_FOR_TTL_MODE_10 ..... MDRV_PUSE_TTL_DOUT05 >,
<PAD_TTL10> ..... PINMUX_FOR_TTL_MODE_10 ..... MDRV_PUSE_TTL_DOUT06 >,
<PAD_TTL11> ..... PINMUX_FOR_TTL_MODE_10 ..... MDRV_PUSE_TTL_DOUT07 >,
<PAD_TTL12> ..... PINMUX_FOR_TTL_MODE_10 ..... MDRV_PUSE_TTL_DOUT08 >,
<PAD_TTL13> ..... PINMUX_FOR_TTL_MODE_10 ..... MDRV_PUSE_TTL_DOUT09 >,
<PAD_TTL14> ..... PINMUX_FOR_TTL_MODE_10 ..... MDRV_PUSE_TTL_DOUT10 >,
<PAD_TTL15> ..... PINMUX_FOR_TTL_MODE_10 ..... MDRV_PUSE_TTL_DOUT11 >,
<PAD_TTL16> ..... PINMUX_FOR_TTL_MODE_10 ..... MDRV_PUSE_TTL_DOUT12 >,
<PAD_TTL17> ..... PINMUX_FOR_TTL_MODE_10 ..... MDRV_PUSE_TTL_DOUT13 >,
<PAD_TTL18> ..... PINMUX_FOR_TTL_MODE_10 ..... MDRV_PUSE_TTL_DOUT14 >,
<PAD_TTL19> ..... PINMUX_FOR_TTL_MODE_10 ..... MDRV_PUSE_TTL_DOUT15 >,
<PAD_TTL20> ..... PINMUX_FOR_GPIO_MODE ..... MDRV_PUSE_TTL_DOUT20 >,
<PAD_TTL21> ..... PINMUX_FOR_ETH1_MODE_5 ..... MDRV_PUSE_NA >,
<PAD_TTL22> ..... PINMUX_FOR_ETH1_MODE_5 ..... MDRV_PUSE_NA >,
<PAD_TTL23> ..... PINMUX_FOR_ETH1_MODE_5 ..... MDRV_PUSE_NA >,
<PAD_TTL24> ..... PINMUX_FOR_ETH1_MODE_5 ..... MDRV_PUSE_NA >,

```

TTL panel

首先找一个已经配好的相同分辨率的ttl 屏参，然后根据panel spec填好以下参数：

u16HSyncWidth & u16HSyncBackPorch & u16VSyncWidth & u16VSyncBackPorch

对应panel spec的数据如下：

Item	Symbol	Values			Unit	Remark
		Min	Typ	Max		
Horizontal Display Area	thd	-	1024	-	DCLK	
DCLK frequency	fck	44.9	51.2	63	MHz	
One horizontal line	th	1200	1344	1400	DCLK	
HS pulse width	thpw	1	-	140	DCLK	
HS Blanking	thb	160	160	160	DCLK	
HS Front Porch	thfp	16	160	216	DCLK	

Item	Symbol	Values			Unit	Remark
		Min	Typ	Max		
Vertical display Area	tvd	-	600	-	TH	
VS period time	tv	624	635	750	TH	
VS pulse width	tvpw	1	-	20	TH	
VS Blanking	tvb	23	23	23	TH	
VS Front Porch	tvfp	1	12	127	TH	

u16HStart = u16HSyncWidth + u16HSyncBackPorch

u16VStart = u16VSyncWidth + u16VSyncBackPorch

u16Width & u16Height对应panel spec数据如下：

6.2 Timing Conditions

Item	Symbol	Values			Unit	Remark
		Min	Typ	Max		
Horizontal Display Area	thd	-	1024	-	DCLK	
DCLK frequency	fck	44.9	51.2	63	MHz	
One horizontal line	th	1200	1344	1400	DCLK	
HS pulse width	thpw	1	-	140	DCLK	
HS Blanking	thb	160	160	160	DCLK	
HS Front Porch	thfp	16	160	216	DCLK	

Item	Symbol	Values			Unit	Remark
		Min	Typ	Max		
Vertical display Area	tvd	-	600	-	TH	
VS period time	tv	624	635	750	TH	
VS pulse width	tvpw	1	-	20	TH	
VS Blanking	tvb	23	23	23	TH	
VS Front Porch	tvfp	1	12	127	TH	

$$u16HTotal = u16Width + u16HSyncWidth + u16HSyncBackPorch + HSyncFrontPorch$$

$$u16VTotal = u16Height + u16VSyncWidth + u16VSyncBackPorch + VSyncFrontPorch$$

对应panel spec数据如下：

6.2 Timing Conditions

Item	Symbol	Values			Unit	Remark
		Min	Typ	Max		
Horizontal Display Area	thd	-	1024	-	DCLK	
DCLK frequency	fck	44.9	51.2	63	MHz	
One horizontal line	th	1200	1344	1400	DCLK	
HS pulse width	thpw	1	-	140	DCLK	
HS Blanking	thb	160	160	160	DCLK	
HS Front Porch	thfp	16	160	216	DCLK	

Item	Symbol	Values			Unit	Remark
		Min	Typ	Max		
Vertical display Area	tvd	-	600	-	TH	
VS period time	tv	624	635	750	TH	
VS pulse width	tvpw	1	-	20	TH	
VS Blanking	tvb	23	23	23	TH	
VS Front Porch	tvfp	1	12	127	TH	

$$u16DCLK = u16HTotal * u16VTotal * frequency(60)/1000*1000$$

如下三个参数分别对应R G B 三个channel的数据是否要做swap

u8SwapOdd_RG & u8SwapEven_RG & u8SwapOdd_GB

对应channel选择的情况如下：

0 default(不做swap)

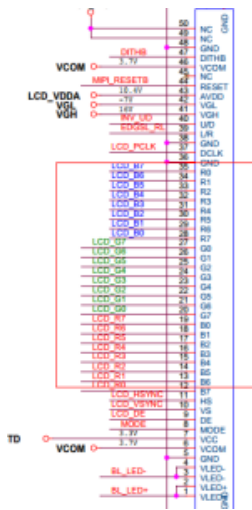
1 select R

2 select G

3 select B

u8SwapEven_GB 表示rgb数据线的高低位是否需要做swap

eg: 如demo板原理图如下，R和B以及高低位反了，所以屏参的配置为3, 2, 1, 1



对应debug 寄存器如下:

1129

11	[bit0-bit12] h total
12	[bit0-bit12] v total
13	[bit0-bit12] hsync start
14	[bit0-bit12] hsync end
15	[bit0-bit12] vsync start
16	[bit0-bit12] vsync end
17	[bit0-bit12] H frame de start
18	[bit0-bit12] H frame de end
19	[bit0-bit12] V frame de start
1a	[bit0-bit12] V frame de end

$$\text{total} - (\text{de end} - \text{de start}) = \text{bp} + \text{fp} + \text{sync width}$$

		[bit0-bit5] rgb swap [bit0-bit1] swap for B channel [bit2-bit3] swap for G channel [bit4-bit5] swap for R channel [bit6-bit7] rgb mode 00:rgb 888 01:rgb 666 10:rgb 565-1 11:rgb 565-2 [bit8] MSB/LSB swap
	7e	

mipi panel

mipi panel跟ttl panel屏参部分相差不大, 主要是需要额外配置mipi data line相关数据, 具体如下:

下面对应的是mipi的四对data line & clk line是否需要swap, 可以根据公版的线序和屏参配置进行设定

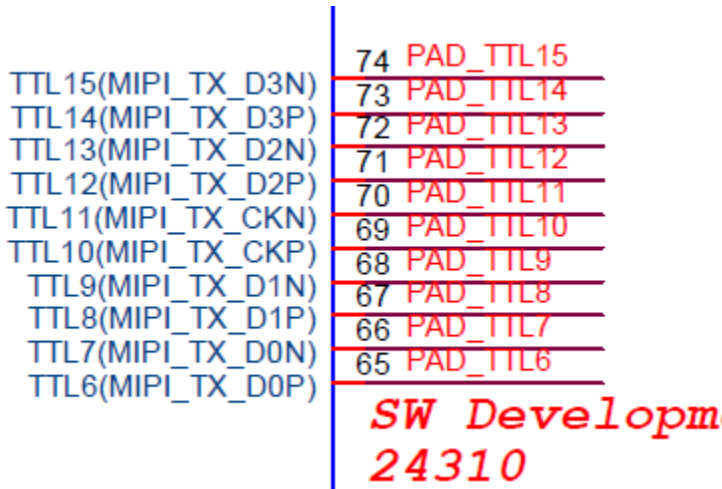
MI_PANEL_ChannelSwapType_e 这个结构体成员分别代表下面几组pin:

eCh0 → TTL11 - 10
 eCh1 → TTL15 - 14
 eCh2 → TTL13 - 12
 eCh3 → TTL9 - 8
 eCh4 → TTL7 - 6

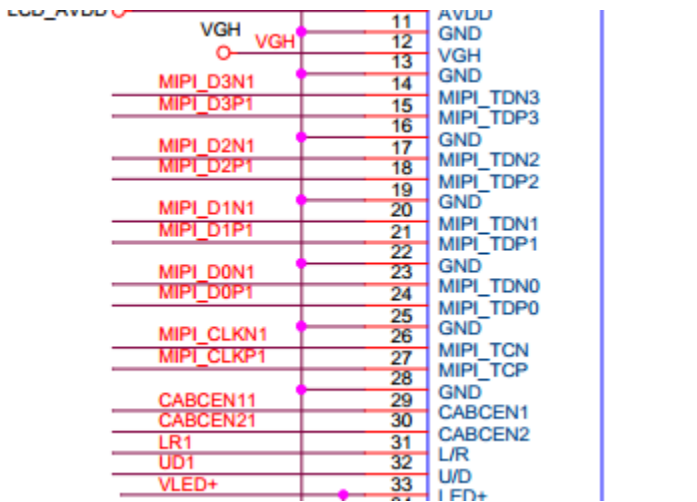
对应取值的定义为:

D0 - 0
 D1 - 1
 D2 - 3
 D3 - 4
 CLK - 2

所以按照公版的原图图对应关系如下:



eg: demo板原理图和配置如下



配置应该为:

(MI_PANEL_ChannelSwapType_e) 2,	- clk (TTL11~10)
(MI_PANEL_ChannelSwapType_e) 4,	- D3 (TTL15~14)
(MI_PANEL_ChannelSwapType_e) 3,	- D1 (TTL13~12)
(MI_PANEL_ChannelSwapType_e) 1,	- D2 (TTL9~8)
(MI_PANEL_ChannelSwapType_e) 0,	- D0 (TTL7~6)

MI_PANEL_MipiDsiConfig_t需要重点关注的参数如下:

u16Hactive u16Hpw u16Hbp u16Hfp u16Vactive u16Vpw u16Vbp u16Vfp

这几个参数对应的是panel spec中的：

u16Width u16HSyncWidth u16HSyncBackPorch HSyncFrontPorch

u16Height u16VSyncWidth u16VSyncBackPorch VSyncFrontPorch

这几个参数需要保持和MI_PANEL_ParamConfig_t一致

enLaneNum 对应mipi 几line mode根据实际情况配置

pu8CmdBuf u32CmdBufSize 对应mipi的初始化cmd, 对应格式如下:

屏厂提供的mipi初始化命令如下:

```
regw(0x01) ;//Reset
Delay 30ms
regw(0x80, 0x58);
regw(0x81, 0x47);

regw(0xB0, 0x00, 0x11, 0x18, 0x0E, 0x11, 0x06, 0x07, 0x08, 0x07, 0x22, 0x04, 0x12, 0x0F, 0xAA, 0x31, 0x18);
```

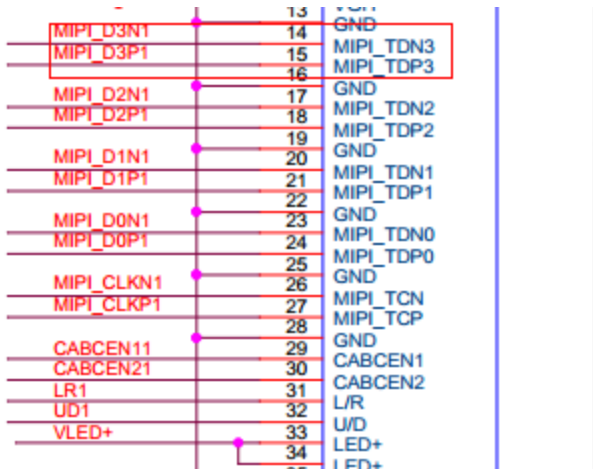
填入屏参的数据格式如下:

```
0x01, 0, 0x0, //0x01: 寄存器, 0: 参数个数, 0x0: 写寄存器的值
FLAG_DELAY, FLAG_DELAY, 30,
0x80, 1, 0x58,
0x81, 1, 0x47,

0xB0, 16, 0x00, 0x11, 0x18, 0x0E, 0x11, 0x06, 0x07, 0x08, 0x07, 0x22, 0x04, 0x12, 0x0F, 0xAA, 0x31, 0x18,
```

u8Po1Ch0 u8Po1Ch1 u8Po1Ch2 u8Po1Ch3 u8Po1Ch4 对应mipi四组data line一组clk line是否需要反转

eg: 如下原理图就不需要反转



如果出现clk&data都正常，但是panel不亮可以微调BK1033_44[15:7]，默认是9，可以左右6 7 8 9 a b c d e f 微调，

这个值是clk和data之间的时间差，对应mipi屏参的最后一个:

```

MI_PANEL_MipiDsiConfig_t stMipiDsiConfig =
{
    //HsTrail HsPrpr HsZero ClkHsPrpr ClkHsExit ClkTrail ClkZero ClkHsPost DaHsExit Cont
    0x05, 0x03, 0x05, 0x0A, 0x0E, 0x03, 0x0B, 0x0A, 0x05, 0x00,
    //Lpx TaGet TaSure TaGo
    0x10, 0x1a, 0x16, 0x32,

    //Hac, Hpw, Hbp, Hfp, Vac, Vpw, Vbp, Vfp, Bllp, Fps
    800, 20, 20, 20, 1280, 4, 10, 30, 0, 60,

    E_MI_PNL_MIPI_DSI_LANE_4, // MIPnlMipiDsiLaneMode_e enLaneNum;
    E_MI_PNL_MIPI_DSI_RGB888, // MIPnlMipiDsiFormat_e enFormat;
    E_MI_PNL_MIPI_DSI_SYNC_PULSE, // MIPnlMipiDsiCtrlMode_e enCtrl;

    JD9365_CMD,
    sizeof(JD9365_CMD),
    1, 0x01AF, 0x01B9, 0x80D2, 9,
};

```

```

MI_PANEL_MipiDsiConfig_t stMipiDsiConfig =
{
    //HsTrail HsPrpr HsZero ClkHsPrpr ClkHsExit ClkTrail ClkZero ClkHsPost DaHsExit ContDet
    0x7, 0x05, 0xC, 0x05, 0x0C, 0x07, 0x1D, 0x0E, 0x0C, 0x00,
    //Lpx TaGet TaSure TaGo
    0x10, 0x1A, 0x18, 0x32,

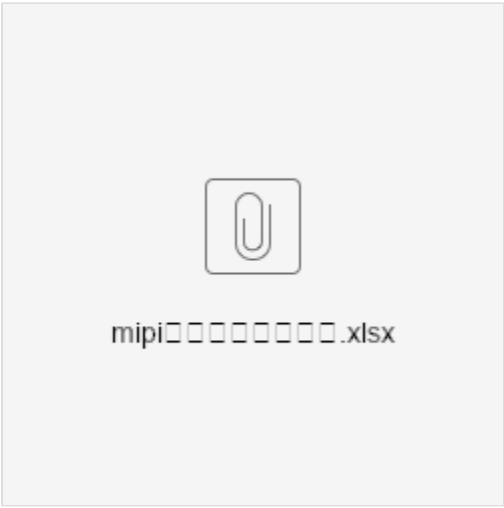
    //Hac, Hpw, Hbp, Hfp, Vac, Vpw, Vbp, Vfp, Bllp, Fps
    1920, 44, 148, 88, 1080, 5, 36, 4, 0, 60,

    .....

```

HsTrail HsPrpr HsZero ClkHsPrpr ClkHsExit ClkTrail ClkZero ClkHsPost DaHsExit的计算可以参考如下表格，之需要填写H_Total, V_Total, FPS, BitsPerPixel, lane number就可以自动算来相关数据，手动取整下就可以了。

需要的公式和概念:



导入屏参

将填好的ttl&mipi 屏参替换到对应点屏demo(H264Player & JpegPlayer)，然后在板子跑点屏demo即可点亮panel

SPI panel

- kernel需要enable如下两个config打开spi 驱动:
 CONFIG_SPI_SPIDEV=y
 CONFIG_MS_SPI_INFINITY=y

2. dts配置好spi相关pin, kernel/arch/arm/boot/dts/infinity2m.dtsi

```
.....spi: spi {
.....compatible = "sstar_spi";
.....io_phy_addr = <0x1f000000>;
.....banks = <0x1110>,<0x1111>,<0x1038>,<0x101E>;
.....clocks = <clk_mspi0>;
.....interrupts = <GIC_SPI INT_IRQ_MSPI_0 IRQ_TYPE_LEVEL_HIGH>;
...../*
.....* padmux:
.....* mode=1: PAD_PM_SD_CDZ, PAD_SD_D1, PAD_SD_D0, PAD_SD_CLK, PAD_SD_CMD
.....* mode=2: PAD_TTL16, PAD_TTL17, PAD_TTL18, PAD_TTL19
.....* mode=3: PAD_GPIO4, PAD_GPIO5, PAD_GPIO6, PAD_GPIO7
.....* mode=4: PAD_FUART_RX, PAD_FUART_TX, PAD_FUART_CTS, PAD_FUART_RTS
.....* mode=5: PAD_GPIO8, PAD_GPIO9, PAD_GPIO10, PAD_GPIO11
.....* mode=6: PAD_GPIO0, PAD_GPIO1, PAD_GPIO2, PAD_GPIO3
.....*/
.....spi0_mode = <5>;
.....status = "ok";
.....};
```

需要注意spi用到的这几根pin不能在padmux里面配置成其他用途

3. 修改makefile, 编译spi 点屏demo

a. sdk/verify/feature/disp/config.mk

```
DEPS:=disp/common

ifeq ($(USE_CASE), fb)
    SRCS:= fb_ut.c
    DEPS+=disp/ugui
else
    ifeq ($(USE_CASE), disp)
        SRCS:= spi_panel.c
    else
        ifeq ($(USE_CASE), vdec_disp)
            SRCS:= vdec_ut.c
        endif
    endif
endif
```

b. project/release/customer_tailor/nvr_i2m_display_glibc_tailor.mk

```
15 interface_venc:=enable
16 interface_wlan:=enable
17
18 misc_fbdev:=enable
19 #verify_zk_full:=enable
20 verify_disp:=enable
21 #mhal
22 #mhal_aio:=disable
23 mhal_csi:=disable
24 #mhal_disp:=disable
25 #mhal_divp:=disable
```

4. 编译spi点屏demo

a. cd sdk/verify/feature

b. make clean;make disp

编译生成的demo在sdk/verify/feature/disp/prog

5. 将prog 和 240_320_rgb565_20.bin放到同一目录下执行./prog即可点亮panel

```
/mnt/P2/spipanel # ./prog
SPI - Open Succeed. Start Init SPI...
spi mode: 0
bits per word: 8
max speed: 60000 KHz (60 MHz)
init panel start
user export gpio:47
user export gpio:48
user export gpio:6
user export gpio:7
gpio:47 direction set ok
gpio:48 direction set ok
gpio:7 direction set ok
gpio:6 direction set ok
input 'q' exit
```

需要注意的是目前demo的pin的配置如下:

CS GPIO47
CS1 GPIO48
RST GPIO6
RS GPIO7

SPI相关pin:
CZ GPIO8
CLK GPIO9
DI GPIO10
DO GPIO11

如果跟demo的pin脚有差异需要在sdk\verify\feature\disp\spi_panel.c中如下地方更改pin的配置

```
: #include "gpio_operation.h"
:
: #define CS_PANEL_0 ... (47)
: #define CS_PANEL_1 ... (48)
:
: #define RS_GPIO_PIN ... (7)
: #define RST_GPIO_PIN ... (6)
:
: static void WaitTime(long ms)
: {
:     int i;
```

FAQ

- 关闭panel clk

```
/config/riu_w 0x1038 53 101
/config/riu_w 0x1038 63 1
```

- 已经点过的panel list



- 展频计算表格

