SC60 Display Driver Development Guide

Smart LTE Module Series

Rev. SC60_Display_Driver_Development_Guide_V1.1

Date: 2018-09-04

Status: Released

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.
7th Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local office. For more information, please visit:
http://www.quectel.com/support/sales.htm

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/technical.htm
Or email to: support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

COPYRIGHT

About the Document

History

Revision	Date	Author	Description
1.0	2018-01-12	Barnett WANG	Initial
1.1	2018-09-04	Arthur LIU	Added the description of WLED backlight current configuration (Chapter 4.2).

Contents

Table Index

1 Introduction

This document describes how to use the Display Serial Interface (DSI) to bring up the display panel on the MSM8953 Android platform of Quectel SC60 module.

This document only describes the important parts for display driver development.

NOTE

We will use SC60 Smart EVB G2's LCD ILI9881C as an example in this document.

2 XML Tags from GCDB for Panel Configuration

The Global Component Database (GCDB) parser provides configuration options for the display panel on MSM8953 chipset.

This chapter describes the display XML tags required to configure Android kernel and little kernel (LK) to bring up panel. To illustrate, it includes details such as XML tag input entries, descriptions, and possible values that can be used in the display parser module for the GCDB, version 1.0.

NOTE

This chapter only describes the improtant parameters. The parameters that we do not specify usually do not need to be modified.

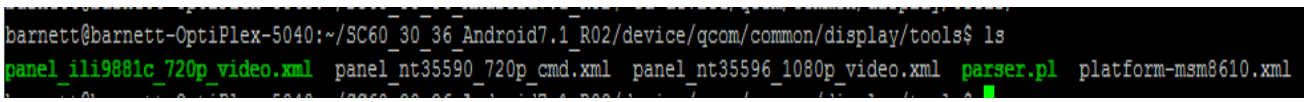
2.1. Display Driver Development Procedure

Take the following steps to develop display driver for any display panel and platform. The process is also documented at *device/qcom/common/display/tools/README.txt*.

1. Install the library by the command below.

```
#sudo apt-get install libxml-libxml-perl
#sudo apt-get install libxml-perl
```

2. Enter the tools directory *#cd device/qcom/common/display/tools* which will show information as below:



3. Copy a new XML to edit.

- Copy the *panel_nt35596_1080p_video.xml* to a new *<oem_panel_input_file>.xml*, if the LCD is in video mode.

- Copy the *panel_nt35590_720p_cmd.xml* to a new *<oem_panel_input_file>.xml*, if the LCD is in command mode.

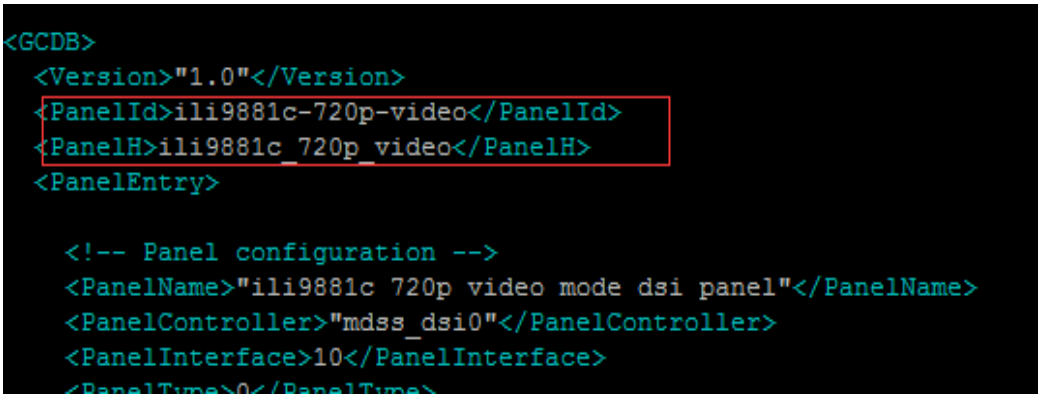
Taking ILI9881C as an example:

```
#cp panel_nt35596_1080p_video.xml panel_ili9881c_720p_video.xml
```

NOTE

Usually, LCD is in video mode and is determined by the LCD module type.

4. Edit the new XML to configure the display parameters.



The configuration details will be introduced in ***Chapter 2.2~2.8***.

5. Generate the device tree file (.dtsi) and header file (.h).

```
#perl parser.pl <oem_panel_input_file>.xml panel
```

Taking ILI9881C as an example:

```
#perl parser.pl panel_ili9881c_720p_video.xml panel
```

Then *dsi-panel-ili9881c-720p-video.dtsi* and *panel_ili9881c_720p_video.h* will be generated.

6. Copy .dtsi to *dts* folder and header file to bootloader folder.

```
#cp device/qcom/common/display/tools/dsi-panel-ili9881c-720p-video.dtsi
     kernel/arch/arm/boot/dts/qcom/
#cp device/qcom/common/display/tools/panel_ili9881c_720p_video.h
     bootable/bootloader/lk/dev/gcdb/display/include/
```

7. Configure the kernel and LK. Please refer to ***Chapter 3*** for details.

8. Build image by the command below.

```
source build/envsetup.sh
lunch msm8953_64-userdebug
make aboot
make bootimage
```

Then the user will get *emmc_appsboot.mbn* and *boot.img* in the directory *out/target/product/msm8953_64/*.

2.2. Tags for Panel Configuration

This chapter defines the specific XML tag entries for the panel.

File path: *device/qcom/common/display/tools/<oem_panel_input_file>.xml*

2.2.1. Panel Information

The parser expects the entries in the following table to be part of the GCDB XML tag.

Table 1: Panel Information

XML Tag/Entry	Description	Value
Version	Each version is bind to be a set of panel tags.	MAJOR_VERSION.MINOR_VERSION For example: 1.0 represents 1 as major version and 0 as minor version.
PanelId	Name used to generate the .dtsi file for Android.	Refer to the following XML input example.
PanelH	Name used to generate the header file for LK.	Refer to the following XML input example.

XML input example:

```
<GCDB>
  <Version>"1.0"</Version>
  <PanelId>ili9881c-720p-video</PanelId>
  <PanelH>ili9881c_720p_video</PanelH>
  <PanelEntry>
```

2.2.2. Panel Configuration Information

The parser expects all other entries to be part of the <GCDB><PanelEntry> tag. Please refer to the following XML input example for more information.

Table 2: Panel Configuration

XML Tag/Entry	Description	Value
PanelName	Panel name	Name/label value
PanelType	Panel is in video mode or command mode.	0 = VIDEO_MODE 1 = COMMAND_MODE
PanelOrientation (Only used in LK)	Panel rotation value. If this entry is not mentioned, LK will be configured with default 0 degree rotation.	1 = 0 degree 2 = 180 degree
PanelFrameRate	An integer to define the panel refresh rate per second.	For example: 60 for 60fps panel.

XML input example:

```
<!-- Panel configuration -->
<PanelName>"ili9881c 720p video mode dsi panel"</PanelName>
<PanelController>"mdss_dsi0"</PanelController>
<PanelInterface>10</PanelInterface>
<PanelType>0</PanelType>
<PanelDestination>"DISPLAY_1"</PanelDestination>
<PanelOrientation>0</PanelOrientation>
<PanelFrameRate>60</PanelFrameRate>
<PanelChannelId>0</PanelChannelId>
<DSIVirtualChannelId>0</DSIVirtualChannelId>
<PanelBroadcastMode>0</PanelBroadcastMode>
<!-- Optional Panel configuration -->
<!--BitClockFrequency>0</BitClockFrequency -->
<DSIStream>0</DSIStream>
<PanelCompatible>"qcom,mdss-dsi-panel"</PanelCompatible>
<InterleaveMode>0</InterleaveMode>
```

2.2.3. Panel Resolution

The following table defines different panel resolution-specific tags. All dimensions should be provided in pixel format.

Table 3: Panel Resolution Configuration

XML Tag/Entry	Description
PanelWidth	Panel width in pixel
PanelHeight	Panel height in pixel
HFrontPorch	Horizontal front porch value
HBackPorch	Horizontal back porch value
HPulseWidth	Horizontal pulse width
HSyncSkew	Horizontal sync skew value
VBackPorch	Vertical back porch value
VFrontPorch	Vertical front porch value
VPulseWidth	Vertical pulse width

XML input example:

```
<!-- Panel Resolution -->
<PanelWidth>720</PanelWidth>
<PanelHeight>1280</PanelHeight>
<HFrontPorch>52</HFrontPorch>
<HBackPorch>100</HBackPorch>
<HPulseWidth>36</HPulseWidth>
<HSyncSkew>0</HSyncSkew>
<VBackPorch>20</VBackPorch>
<VFrontPorch>8</VFrontPorch>
<VPulseWidth>4</VPulseWidth>
<HLeftBorder>0</HLeftBorder>
<HRightBorder>0</HRightBorder>
<VTopBorder>0</VTopBorder>
<VBottomBorder>0</VBottomBorder>
<!-- Optional Panel resolution configuration -->
<!--HActiveRes>0</HActiveRes>
<VActiveRes>100</VActiveRes>
<InvertDataPolarity>0</InvertDataPolarity>
<InvertVsyncPolarity>0</InvertVsyncPolarity>
<InvertHsyncPolarity>0</InvertHsyncPolarity -->
```

2.2.4. Panel Color Information

The panel color format, swap values, and border color are defined below.

Table 4: Panel Color Information

XML Tag/Entry	Description	Value
ColorFormat	Defines the bits per pixel.	24 = 888_RGB 18 = 666_RGB 16 = 565_RGB 12 = 444_RGB 8 = 332_RGB 3 = 111_RGB
ColorOrder	Defines pixel component color order between MSM and panel.	0 = DSI_RGB_SWAP_RGB 1 = DSI_RGB_SWAP_RBG 2 = DSI_RGB_SWAP_BGR 3 = DSI_RGB_SWAP_BRG 4 = DSI_RGB_SWAP_GRB 5 = DSI_RGB_SWAP_GBR

XML input example:

```
<!-- Panel Color Information -->
<ColorFormat>24</ColorFormat>
<ColorOrder>0</ColorOrder>
<UnderFlowColor>0xff</UnderFlowColor>
<BorderColor>0</BorderColor>
```

2.3. Panel Command Information

This chapter describes the on/off commands sent to the panel. Customers can send all valid DSI commands to the panel.

Table 5: Panel Command Information

XML Tag/Entry	Description	Value
OnCommand	Array of variable length that lists the initialization commands of the panel.	Refer to ***Chapter 2.3.1***

OffCommand	Array of variable length that lists the deinitialization commands of the panel.	
OnCommandState	Panel state when sending the on command.	0 = DSI_LP_MODE 1 = DSI_HP_MODE
OffCommandState	Panel state when sending the off command.	

Generally speaking, <OnCommandState> and <OffCommandState> only need to keep the default value.

```
<!-- Panel Command information -->
<OnCommand>"0x39, 0x01, 0x00, 0x00, 0x00, 0x00, 0x04, 0xFF, 0x98, 0x81, 0x03,
0x39, 0x01, 0x00, 0x00, 0x00, 0x00, 0x02, 0x01, 0x00,
0x39, 0x01, 0x00, 0x00, 0x00, 0x00, 0x02, 0x02, 0x00,
```

```
<OffCommand>"0x05, 0x01, 0x00, 0x00, 0x32, 0x00, 0x02, 0x28, 0x00,
            0x05, 0x01, 0x00, 0x00, 0x78, 0x00, 0x02, 0x10, 0x00"</OffCommand>
<OnCommandState>0</OnCommandState>
<OffCommandState>1</OffCommandState>
```

2.3.1. Command Format

Each command needs the input information shown in the table below. The current parser supports one complete command per line.

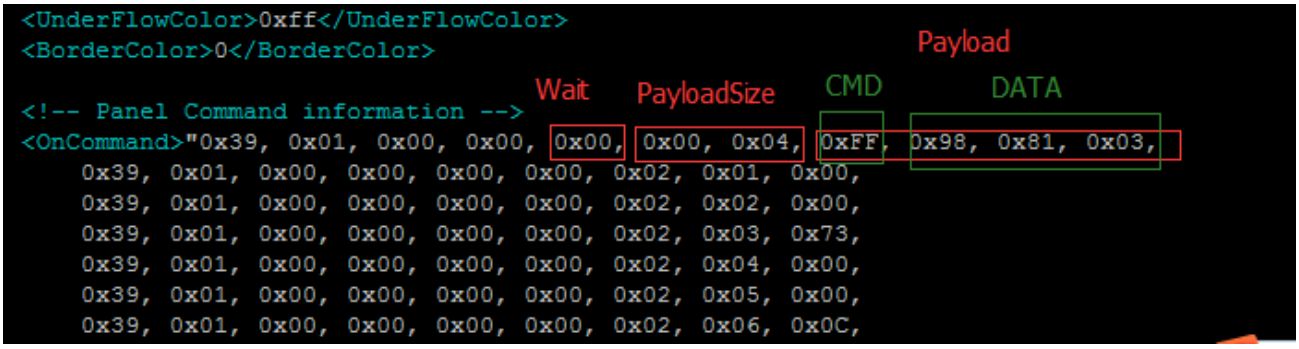
Table 6: Command Format

Subentry	Description	Byte Length
CommandType	Data type of command	1
Last	Specifies if this command packet is individual or not.	1
VC	Virtual channel used to send this command.	1
Ack	Needs acknowledgement from the panel.	1
Wait	Sleep in microseconds before sending next command.	1
PayloadSize	Payload size	2
Payload	Actual command	Based on PayloadSize

1. Usually we only need to modify the parameters “PayloadSize” and “Payload”.

Some LCD may require sleep several microseconds before sending the next command. The parameter "Wait" can be modified.

For example:



2. Parameter "Payload" is provided by the LCD module manufacturer.

We hope that the parameters provided by LCD module manufacturer are available in the following format:

```
REGISTER, CMD(Hex), Number(Hex), DATA(Hex), DATA(Hex), …
```

When parameters are provided in the above format, then the <OnCommand> parameters will be:

```
0x39, 0x01, 0x00, 0x00, 0x00 = REGISTER
PayloadSize = Number+1
Payload = CMD,DATA,DATA,…
```

For example, if the parameters provided by LCD module manufacturers are:

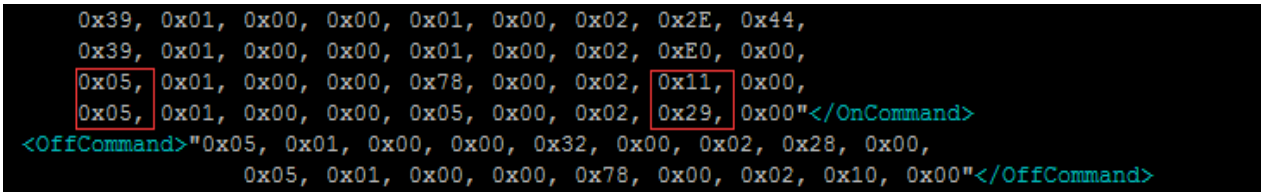
```
REGISTER,FF,3,98,81,03
REGISTER,01,01,00
```

Then the <OnCommand> parameters will be:

```
0x39, 0x01, 0x00, 0x00, 0x00, 0x00, 0x04, 0xFF, 0x98, 0x81, 0x03,
0x39, 0x01, 0x00, 0x00, 0x00, 0x00, 0x02, 0x01, 0x00,
```

3. Usually the last two commands of <OnCommand> are available as below, but some LCD chips do not need them.

```
0x05, 0x01, 0x00, 0x00, 0x78, 0x00, 0x02, 0x11, 0x00,
0x05, 0x01, 0x00, 0x00, 0x05, 0x00, 0x02, 0x29, 0x00,
```



0x11: sleep out command of LCD.
0x29: display on command of LCD.

2.4. Video Mode Panel

The following table describes the video mode panel configuration entries. Some configurations are also used for command mode panel.

Table 7: Video Mode Panel

XML Tag/Entry	Description	Value
HSyncPulse	Horizontal sync pulses. It determines whether the hardware should send horizontal sync pulses during the vertical blanking period.	0 = Sync pulse disable 1 = Sync pulse enable
TrafficMode	Panel traffic mode type	0 = non burst with sync pulses 1 = non burst with sync start event 2 = burst mode

Usually these parameters do not need to be modified. The user may need to note the parameter <TrafficMode> which may vary with the LCD chip.

```
<!-- Video mode panel information -->
<HSyncPulse>1</HSyncPulse>
<HFPPowerMode>0</HFPPowerMode>
<HBPPowerMode>0</HBPPowerMode>
<HSAPowerMode>0</HSAPowerMode>
<BLLPEOFPowerMode>1</BLLPEOFPowerMode>
<BLLPPowerMode>1</BLLPPowerMode>
<TrafficMode>2</TrafficMode>
<DMADelayAfterVsync>0</DMADelayAfterVsync>
<BLLPEOFPower>0x9</BLLPEOFPower>
```

2.5. Lane Configuration

The table below describes the DSI panel's lane configuration entries.

Table 8: Lane Configuration

XML Tag/Entry	Description	Value
DSILanes	Number of lanes used for communication with DSI.	2 for 2 lane panel 4 for 4 lane panel
DSILaneMap	It defines how the data lanes are mapped to the panel.	0 = DLANE_SWAP_0123 1 = DLANE_SWAP_3012 2 = DLANE_SWAP_2301 3 = DLANE_SWAP_1230 4 = DLANE_SWAP_0321 5 = DLANE_SWAP_1032 6 = DLANE_SWAP_2103 7 = DLANE_SWAP_3210
Lane0State	Lane 0 state	1 = ENABLE 0 = DISABLE
Lane1State	Lane 1 state	
Lane2State	Lane 2 state	
Lane3State	Lane 3 state	

For example:

```
<!-- Lane Configuration -->
<DSILanes>4</DSILanes>
<DSILaneMap>0</DSILaneMap>
<Lane0State>1</Lane0State>
<Lane1State>1</Lane1State>
<Lane2State>1</Lane2State>
<Lane3State>1</Lane3State>
```

2.6. Panel Physical Data

The table below describes all panel's physical configuration register entries.

Table 9: Panel Physical Data

XML Tag/Entry	Description	Value
PanelTimings	An array of length 12 that specifies the physical timing settings for the panel	
TClkPost	DSI timing control clock post value	
TClkPre	DSI timing control clock pre value	

For example:

```
<!-- Panel Timing -->
<PanelTimings>"0xf9, 0x3d, 0x34, 0x00, 0x58, 0x4d, 0x36, 0x3f, 0x53, 0x03, 0x04, 0x00"</PanelTimings>
<DSIMDPTrigger>0</DSIMDPTrigger>
<DSIDMATrigger>4</DSIDMATrigger>
<TClkPost>0x1e</TClkPost>
<TClkPre>0x38</TClkPre>
```

2.6.1. Generate DSI Timing

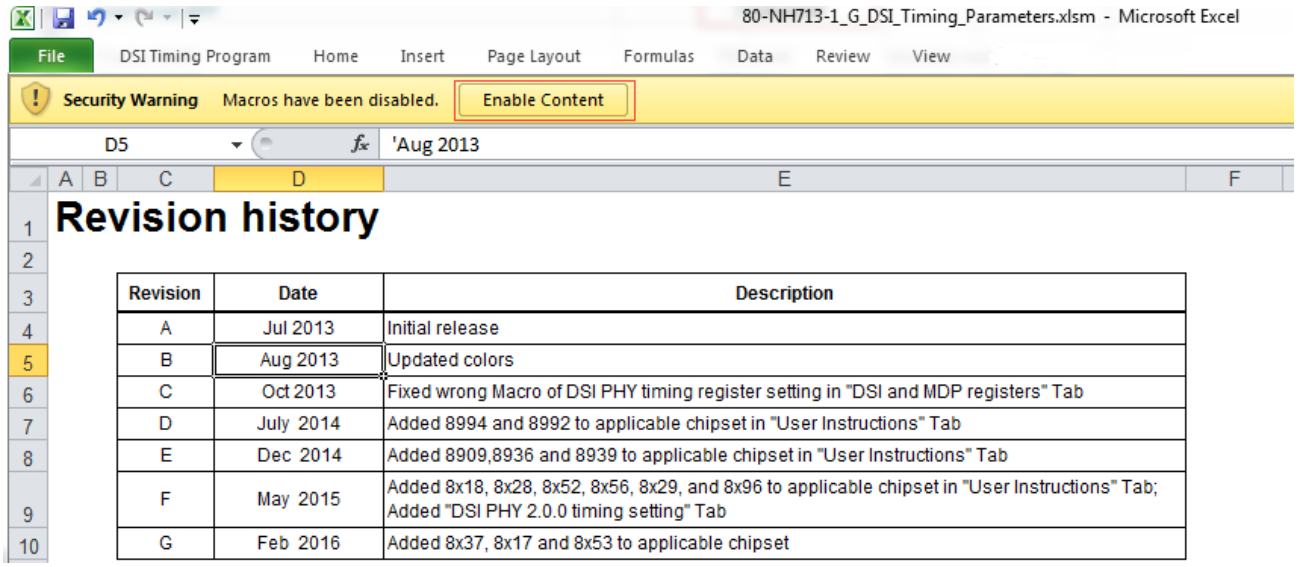
The panel requires the PHY value setup for "bitclk" in the DSI PHY register. The *80-NH713-1_G_DSI_Timing_Parameters.xlsm* in *80-NH713-1_DSI.zip* (the zip file is available in *tools* directory) can be used to calculate the timing values automatically.

NOTE

Please use Microsoft Excel (Microsoft Excel 2016 is recommended) to open this file. Other tools such as WPS are not recommended.

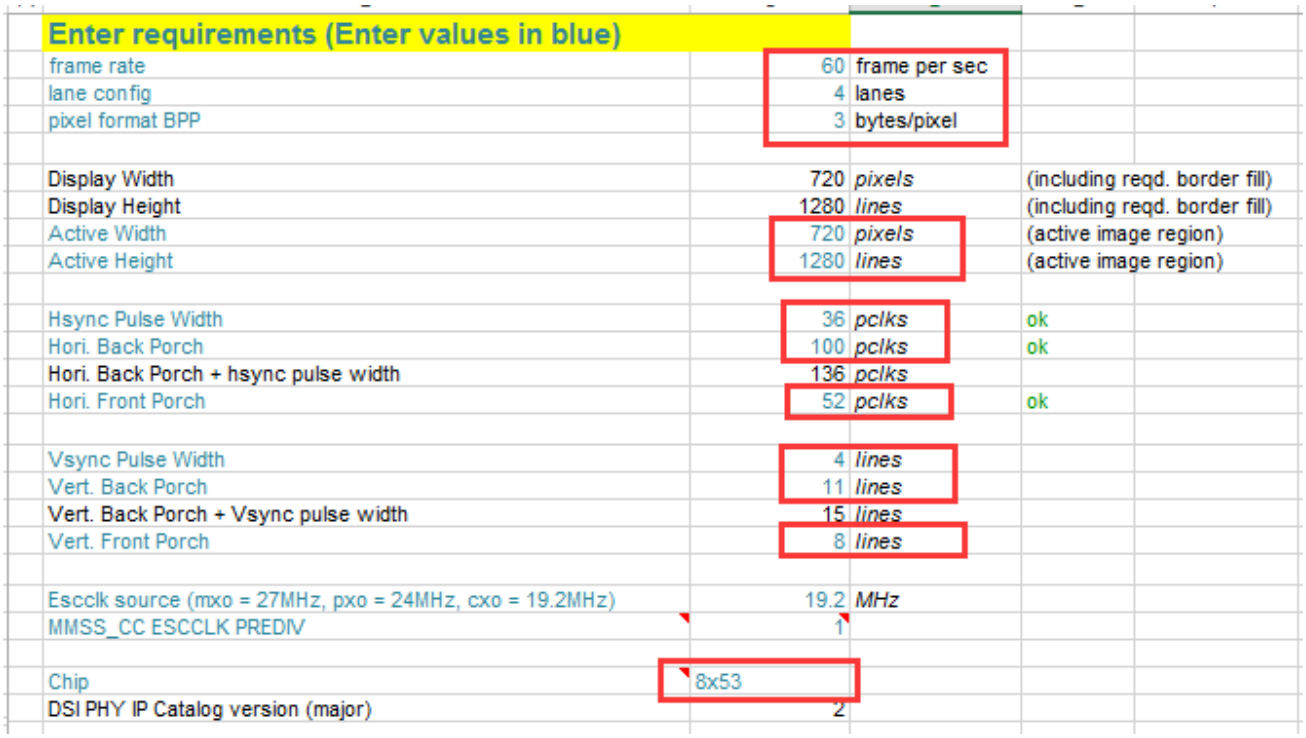
To auto-calculate these values, follow the steps below.

1. Click “Enable Content”.



Revision	Date	Description
A	Jul 2013	Initial release
B	Aug 2013	Updated colors
C	Oct 2013	Fixed wrong Macro of DSI PHY timing register setting in "DSI and MDP registers" Tab
D	July 2014	Added 8994 and 8992 to applicable chipset in "User Instructions" Tab
E	Dec 2014	Added 8909,8936 and 8939 to applicable chipset in "User Instructions" Tab
F	May 2015	Added 8x18, 8x28, 8x52, 8x56, 8x29, and 8x96 to applicable chipset in "User Instructions" Tab; Added "DSI PHY 2.0.0 timing setting" Tab
G	Feb 2016	Added 8x37, 8x17 and 8x53 to applicable chipset

2. Open the **DSI and MDP Registers** sheet. Input frame rate/panel resolution/porch/lane numbers/chip value to the area framed in red line in the example as below.



If the spreadsheet says to change the porches to be multiple of 4 or make it even, please change it accordingly and also update the panel porch node. Customers can refer to the user instructions in the Excel sheet.

3. Open the **DSI PHY Timing Setting** sheet to see the calculated DSI-related clock rate in the blue fields. Press **CTRL+J**, a value of **INVALID** will appear in the **Check for T_CLK_ZERO** field. Then press

CTRL+K to recalculate **T_CLK_ZERO** to **VALID**.

1	**1. PHY Timing parameters calculated from bitclk calculated in "dsi and mdp registers" and escclk source set in "dsi and mdp**								
2	**(User may overwrite the values in blue)**								
3	**Full Rate Bitclk**	1141.00	Mbps						
4	**escclk**	19.2	MHz				Check for T_CLK_ZERO	INVALID	
5	**UI**	0.876424189	ns						
6	**Tlpx**	52.08333333	ns						
7	**Treot**	20	ns						
8		MIPI PHY v1.1 requirement		Recommended register settings (dec)			program value	theoretical	
9		min (ns)	max (ns)	min	max		(hardwired to PHY inputs)	value (ns)	
10	T_CLK_PREPARE	38	95	42	107	48	48	43.82120947	
11	T_CLK_ZERO	256.1787905		291	511	320	INVALID	#VALUE!	
12	T_CLK_TRAIL	60	95.51709027	67	107	70	70	63.10254163	
13	T_HS_PREPARE	43.50569676	90.25854514	48	101	54	54	50.83260298	
14	T_HS_ZERO	102.9316389		116	255	130	130	115.687993	
15	T_HS_TRAIL	63.50569676	95.51709027	71	107	74	74	66.60823839	
16	T_HS_RQST					59	59	52.58545136	
17	T_HS_EXIT	100		113	255	128	128	113.9351446	
18	T_TA_GO	208.3333333	208.3333333			208.3333333	3	208.3333333	
19	T_TA_SURE	52.08333333	104.1666667			52.08333333	0	104.1666667	
20	T_TA_GET	260.4166667	260.4166667			260.4166667	4	260.4166667	
21	TEOT of data lane		115.5170903					76.60823839	
22	TEOT of clock lane		115.5170903					73.10254163	
23	T_CLK_POST	105.5740578		-6	63	1	1	148.9921122	
24	T_CLK_PRE	7.011393514		#VALUE!	63	#VALUE!	#VALUE!	#VALUE!	
25	overhead in data transmission							#VALUE!	

1	**1. PHY Timing parameters calculated from bitclk calculated in "dsi and mdp registers" and escclk source set in "dsi and mdp registers"**								
2	**(User may overwrite the values in blue)**								
3	**Full Rate Bitclk**	849.00	Mbps						
4	**escclk**	19.2	MHz				Check for T_CLK_ZERO	VALID	
5	**UI**	1.177856302	ns						
6	**Tlpx**	52.08333333	ns						
7	**Treot**	20	ns						
8		MIPI PHY v1.1 requirement		Recommended register settings (dec)			program value	actual value (ns	
9		min (ns)	max (ns)	min	max		(hardwired to PHY inputs)		
10	T_CLK_PREPARE	38	95	31	79	53	53	63.60424028	
11	T_CLK_ZERO	236.3957597		199	255	209	209	247.3498233	
12	T_CLK_TRAIL	60	99.13427562	49	83	51	51	61.24852768	
13	T_HS_PREPARE	44.71142521	92.06713781	36	77	55	55	65.95995289	
14	T_HS_ZERO	90.81861013		76	255	83	83	98.93992933	
15	T_HS_TRAIL	64.71142521	99.13427562	53	83	56	56	68.31566549	
16	T_HS_RQST					42	42	51.82567727	
17	T_HS_EXIT	100		83	255	92	92	110.7184923	
18	T_TA_GO	208.3333333	208.3333333			208.3333333	3	208.3333333	
19	T_TA_SURE	52.08333333	104.1666667			52.08333333	0	104.1666667	
20	T_TA_GET	260.4166667	260.4166667			260.4166667	4	260.4166667	
21	TEOT of data lane		119.1342756					78.31566549	
22	TEOT of clock lane		119.1342756					71.24852768	
23	T_CLK_POST	121.2485277		-3	63	4	4	186.1012956	
24	T_CLK_PRE	9.422850412		39	63	42	42	42.14517079	
25	overhead in data transmission							1057.714959	
26									

clock related information

T_CLK_ZERO:
This is a SW workaround for Aragorn V1:
If G4 cell states "VALID", then use all programming values as listed.

If G4 states INVALID, press **CTRL + K** to auto-calculate new Tclk_zero program value.

Press **CTRL + J** to return to default equation.

The spreadsheet will show the PHY value setup for “bitclk” which the panel requires in the DSI PHY register.

20	T_TA_GET	260.4166667	260.4166667			260.4166667	4	260.4166667
21	TEOT of data lane		133.0373832					84.76635514
22	TEOT of clock lane		133.0373832					75.42056075
23	T_CLK_POST	181.4953271		-3	63	4	4	299.0654206
24	T_CLK_PRE	18.69158879		21	63	26	26	116.1409657
25	overhead in data transmission							1383.17757
26								
27	**2. DSI PHY registers**							
28	PHY Registers	value in hex						
29	DSIPHY_TIMING_CTRL_0	7A						
30	DSIPHY_TIMING_CTRL_1	1A						
31	DSIPHY_TIMING_CTRL_2	12						
32	DSIPHY_TIMING_CTRL_3	0						
33	DSIPHY_TIMING_CTRL_4	3E						
34	DSIPHY_TIMING_CTRL_5	42						
35	DSIPHY_TIMING_CTRL_6	16						
36	DSIPHY_TIMING_CTRL_7	1E						
37	DSIPHY_TIMING_CTRL_8	14						
38	DSIPHY_TIMING_CTRL_9	3						
39	DSIPHY_TIMING_CTRL_10	4						
40								
41	**3. DSI Registers (address)**							
42	DSI_CLKOUT_TIMING_CTRL	41A						
43	DSI_TEST_PATTERN_GEN_VIDEO_ENABLE	0						

4. Update the panel XML file <PanelTimings> using the values listed above that obtained from the Excel worksheet.

The following is an example. Values are copied from above.

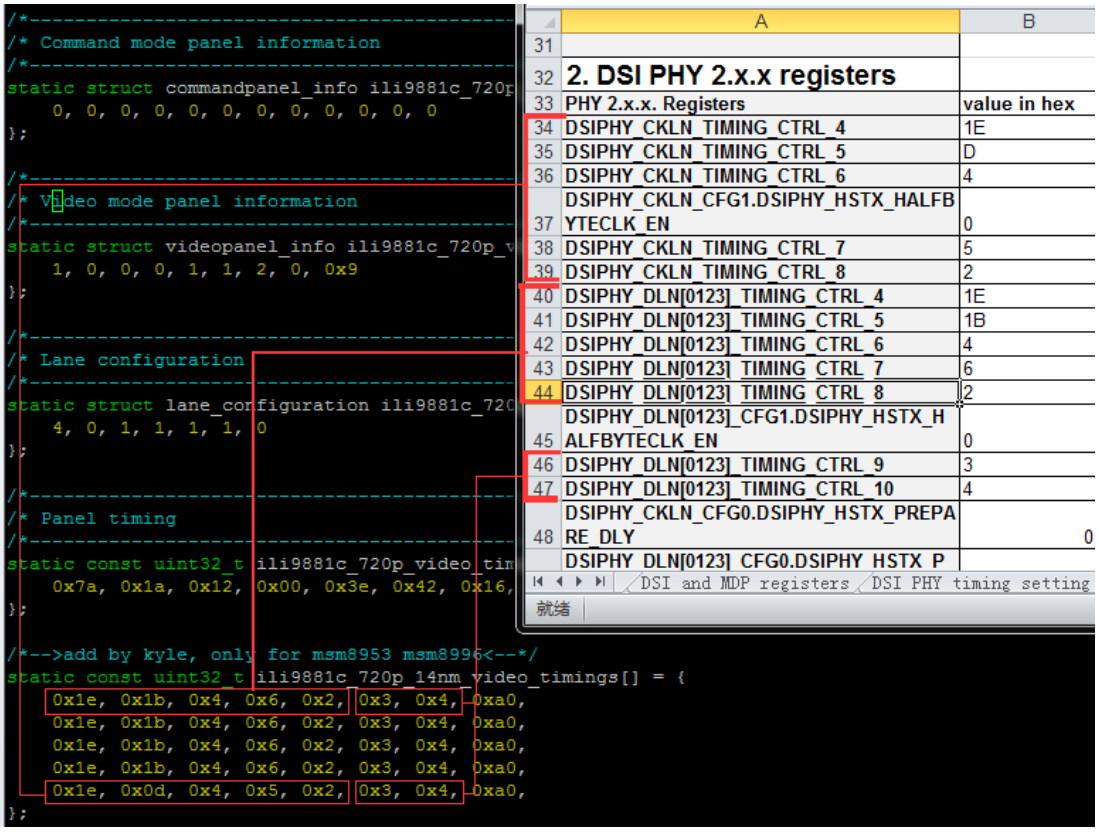
```
<PanelTimings>"0x7a, 0x1a, 0x12, 0x00, 0x3e, 0x42, 0x16, 0x1e, 0x14, 0x03, 0x04, 0x00"</PanelTimings>
```

5. Copy the program value for **T_CLK_POST** and **T_CLK_PRE** fields obtained from the Excel worksheet into the panel XML. The values in the Excel worksheet are in decimal. Make sure that they are converted to HEX before updating these two elements.

```
<TClkPost>0x4</TClkPost>
<TClkPre>0x1a</TClkPre>
```

6. For SC60, DSI PHY 2.0.0 timing config needs to be added, and it is in *panel_ili9881c_720p_video.h*. For example:

```
static const uint32_t ili9881c_720p_14nm_video_timings[] = {
    0x1e, 0x1b, 0x4, 0x6, 0x2, 0x3, 0x4, 0xa0,
    0x1e, 0x1b, 0x4, 0x6, 0x2, 0x3, 0x4, 0xa0,
    0x1e, 0x1b, 0x4, 0x6, 0x2, 0x3, 0x4, 0xa0,
    0x1e, 0x1b, 0x4, 0x6, 0x2, 0x3, 0x4, 0xa0,
    0x1e, 0x0d, 0x4, 0x5, 0x2, 0x3, 0x4, 0xa0,
};
```



2.7. Command Mode Panel

All TE-related entries shown in the table below are only used in Android kernel.

These parameters are only used in command mode panel and they can be ignored if the LCD is a video mode panel.

Table 10: TE-related Entries

Entry	Description	Value
TECheckEnable	It enables/disables the TE to check hardware block within MDP.	0 = disabled 1 = enabled
TEPinSelect	TE is enabled to check hardware block through external GPIO pin or an embedded TE signal.	0 = uses embedded DCS command 1 = through GPIO pin
TEUsingTEPin	TE is enabled to check hardware block through software Vsync or hardware Vsyc.	0 = Software Vsync 1 = Hardware Vsync
TEvSyncRdPtrIrqLine	This integer configures the scan line number that the DSI pixel transfer will start on. This should be left at the default value of 0. Adjusting this number to a higher value will result in delay in the start of the pixel transfer.	
TEvSyncContinueLines	This integer represents the difference in the number of lines between estimated read pointer and write pointer to allow the updating of all the lines except the first line of the frame. This threshold is maintained only when Tear Check block is enabled.	
TEDCSCommand	This entry inserts the DCS command.	

For example:

```
<!--- command mode panel -->
<TECheckEnable>1</TECheckEnable>
<TEPinSelect>1</TEPinSelect>
<TEUsingTEPin>1</TEUsingTEPin>
<TEvSyncRdPtrIrqLine>0x2c</TEvSyncRdPtrIrqLine>
<TEvSyncContinueLines>0x3c</TEvSyncContinueLines>
<TEDCSCommand>1</TEDCSCommand>
```

2.8. Backlight Configuration

This chapter describes the various backlight configuration entries.

Table 11: Backlight Configuration

Entry	Description	Value
BLMinLevel	Minimum value of backlight.	
BLMaxLevel	Maximum value of backlight.	
BLPMICControlType	PMIC controller for current backlight.	0 = PWM GPIO 1 = WLED 2 = DCS COMMANDS (for OLED panel backlight controller) 3 = LPG

For example:

```
<!-- Backlight -->
<BLInterfaceType>1</BLInterfaceType>
<BLMinLevel>1</BLMinLevel>
<BLMaxLevel>4095</BLMaxLevel>
<BLStep>100</BLStep>
<BLPMICModel>"PMIC 8941"</BLPMICModel>
<BLPMICControlType>0</BLPMICControlType>
```

3 Configure Kernel and LK

NOTE

Panel parameters are configured in both LK and kernel. After startup, configuration parameters in LK will be used to drive the panel before pressing the power key. After system sleep and then wakeup, configuration parameters in kernel will be used to drive the panel.

3.1. Configure Kernel

After the file *dsi-panel-ili9881c-720p-video.dtsi* is generated, copy it to the following directory: *kernel/msm-3.18/arch/arm/boot/dts/qcom/*. And then configure the kernel to add a new panel.

1. Include .dtsi files.

Path: *kernel/msm-3.18/arch/arm/boot/dts/qcom/msm8953-mdss-panels.dtsi*

```
#include "dsi-panel-r69006-1080p-video.dtsi"
#include "dsi-panel-r69006-1080p-cmd.dtsi"
#include "dsi-panel-truly-wuxga-video.dtsi"
#include "dsi-panel-ili9881c-720p-video.dtsi"
#include "dsi-panel-ili9881c-720p-dsi1-video.dtsi"
#include "dsi-panel-lt8912-720p-video.dtsi"
```

2. Modify .dtsi files.

Add DSI PHY 2.0.0 timing config to the following directory: *kernel/msm-3.18/arch/arm/boot/dts/qcom/msm8953-mdss-panels.dtsi*.

```
&dsi_ili9881c_720p_video {
    qcom,mdss-dsi-panel-timings-phy-v2 = [
        1e 1b 04 06 02 03 04 a0     /*Data 0*/
        1e 1b 04 06 02 03 04 a0     /*Data 1*/
        1e 1b 04 06 02 03 04 a0     /*Data 2*/
        1e 1b 04 06 02 03 04 a0     /*Data 3*/
        1e 0d 04 05 02 03 04 a0];   /*CLK lane*/
};
```

The parameters of timing “mdss-dsi-panel-timings-phy-v2” are the same as that of the timimg configured

in LK “ili9881c_720p_14nm_video_timings”.

The following part should be modified also:

Path: *kernel/msm-3.18/arch/arm/boot/dts/qcom/msm8953-mtp.dtsi*

```
&mdss_dsi0 {
    qcom,dsi-pref-prim-pan = <&dsi_ili9881c_720p_video>;
    pinctrl-names = "mdss_default", "mdss_sleep";
    pinctrl-0 = <&mdss_dsi_active &mdss_te_active>;
    pinctrl-1 = <&mdss_dsi_suspend &mdss_te_suspend>;

    qcom,platform-te-gpio = <&tlmm 24 0>;
    qcom,platform-reset-gpio = <&tlmm 61 0>;
//  qcom,platform-bklight-en-gpio = <&tlmm 59 0>;  //not used
};
```

3. Configure the Backlight.

Path: *kernel/msm-3.18/arch/arm/boot/dts/qcom/msm8953-mtp.dtsi*

```
/*kyle: backlight control by wled*/
&dsi_ili9881c_720p_video {
    qcom,mdss-dsi-bl-pmic-control-type = "bl_ctrl_wled";
    qcom,panel-supply-entries = <&dsi_panel_pwr_supply>;
    //qcom,mdss-dsi-pan-enable-dynamic-fps;
    //qcom,mdss-dsi-pan-fps-update = "dfps_immediate_porch_mode_vfp";
    //qcom,cont-splash-enabled;
};
```

- Configure backlight type

qcom,mdss-dsi-bl-pmic-control-type: a string that specifies the implementation of backlight control for this panel.

"bl_ctrl_pwm" = Backlight controlled by PWM GPIO
"bl_ctrl_wled" = Backlight controlled by WLED
"bl_ctrl_dcs" = Backlight controlled by DCS commands(for OLED panel backlight controller)
Others: Unknown backlight control. (default)

- For the WLED backlight control, select the “control-type” shown as below.

Path: *kernel/msm-3.18/arch/arm/boot/dts/qcom/msm8953-mtp.dtsi*

```
&dsi_ili9881c_720p_video {
    qcom,mdss-dsi-bl-pmic-control-type = "bl_ctrl_wled";
    qcom,panel-supply-entries = <&dsi_panel_pwr_supply>;
};
```

- For the PWM backlight control, select the “control-type”, “pwm-frequency” and “pwm-gpio” shown as below.

```
&dsi_ili9881c_720p_video {
        qcom,mdss-dsi-bl-pmic-control-type = "bl_ctrl_pwm";
        qcom,mdss-dsi-bl-pmic-pwm-frequency = <100>;
        qcom,mdss-dsi-bl-pmic-bank-select = <0>;
        qcom,mdss-dsi-pwm-gpio = <&pm8953_mpps 4 0>;
        qcom,panel-supply-entries = <&dsi_panel_pwr_supply>;
};
```

If PWM backlight control is used on DSI0, please disable the DSI1 bridge. The configuration is not applicable to SC60 No PMI versions (firmware versions in the format of SC60XXTBXXXXXX or SC60XXYBXXXXXX).

```
    qcom,dsi1 bridge {
      //compatible = "qcom,dsi1 bridge";
        instance_id = <0>;
        mdss-dsi-bl-pmic-pwm-frequency = <100>;
        mdss-dsi-bl-pmic-bank-select = <0>;
        mdss-dsi-bl-max-level = <4095>;
    };
};
```

- Control the backlight by sending the DCS commands at any point.

Path: *kernel/arch/arm/boot/dts/qcom/dsi-panel-nt35521-720p-video.dtsi*

```
qcom,mdss-dsi-bl-min-level = <1>;
qcom,mdss-dsi-bl-max-level = <255>;
qcom,mdss-dsi-bl-pmic-control-type = "bl_ctrl_dcs";
```

Use the following code to send the backlight command.

Path: *kernel/msm-3.18/ drivers/video/msm/mdss/mdss_dsi_panel.c*

```
static char led_pwm1[2] = {0x51, 0x0};    /* DTYPE_DCS_WRITE1 */
static struct dsi_cmd_desc backlight_cmd = {
    {DTYPE_DCS_WRITE1, 1, 0, 0, 1, sizeof(led_pwm1)},
    led_pwm1
};
```

3.2. Configure LK

After the file *panel_ili9881c_720p_video.h* is generated, please copy it to the directory *bootable/bootloader/lk/dev/gcdb/display/include/*. Customers need to configure the LK to add a new panel.

1. Add several parameters to *panel_ili9881c_720p_video.h* to avoid compiling error.

```
static struct panel_config ili9881c_720p_video_panel_data = {
    "qcom,mdss_dsi_ili9881c_720p_video", "dsi:0:", "qcom,mdss-dsi-panel",
    10, 0, "DISPLAY_1", 0, 0, 60, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, NULL
};
```

```
/*---------------------------------------------------------------------------*/
/* Lane configuration                                                        */
/*---------------------------------------------------------------------------*/
static struct lane_configuration ili9881c_720p_video_lane_config = {
    4, 0, 1, 1, 1, 1, 0
};
```

2. Add the DSI PHY 2.0.0 timing config mentioned in step 6 in ***Chapter 2.6.1*** to *panel_ili9881c_720p_video.h*.

3. Add a new panel.

Path: *bootable/bootloader/lk/target/msm8953/oem_panel.c*

```
#include "panel_display.h"

#include "include/panel_ili9881c_720p_video.h"
#include "include/panel_otm9605a_qhd_video.h"
#include "include/panel_hx8394d_720p_video.h"
#include "include/panel_sharp_qhd_video.h"
#include "include/panel_truly_wvga_cmd.h"
#include "include/panel_hx8379a_fwvga_skua_video.h"
#include "include/panel_ili9806e_fwvga_video.h"
#include "include/panel_hx8394d_qhd_video.h"
#include "include/panel_hx8379c_fwvga_video.h"

#define DISPLAY_MAX_PANEL_DETECTION 2
#define ILI9806E_FWVGA_VIDEO_PANEL_POST_INIT_DELAY 68

enum {
    QRD_SKUA = 0x00,
    QRD_SKUC = 0x08,
    QRD_SKUE = 0x09,
};

/*---------------------------------------------------------------------------*/
/* static panel selection variable                                           */
/*---------------------------------------------------------------------------*/
static uint32_t auto_pan_loop = 0;

enum {
    ILI9881C_720P_VIDEO_PANEL,
    OTM9605A_QHD_VIDEO_PANEL,
    HX8394D_720P_VIDEO_PANEL,
```

```
static struct panel_list supp_panels[] = {
    {"ili9881c_720p_video", ILI9881C_720P_VIDEO_PANEL},
    {"otm9605a_qhd_video_panel",OTM9605A_QHD_VIDEO_PANEL},
    {"hx8394d_720p_video", HX8394D_720P_VIDEO_PANEL},
    {"sharp_qhd_video", SHARP_QHD_VIDEO_PANEL},
    {"truly_wvga_cmd", TRULY_WVGA_CMD_PANEL},
    {"hx8379a_fwvga_skua_video", HX8379A_FWVGA_SKUA_VIDEO_PANEL},
    {"ili9806e_fwvga_video",ILI9806E_FWVGA_VIDEO_PANEL},
    {"hx8394d_qhd_video", HX8394D_QHD_VIDEO_PANEL},
    {"hx8379c_fwvga_video",HX8379C_FWVGA_VIDEO_PANEL},
};
```

4. Add new panel selection branch in function “init_panel_data”.

```
static int init_panel_data(struct panel_struct *panelstruct,
            struct msm_panel_info *pinfo,
            struct mdss_dsi_phy_ctrl *phy_db)
{
    int pan_type = PANEL_TYPE_DSI;

    switch (panel_id) {
    case ILI9881C_720P_VIDEO_PANEL:
        panelstruct->paneldata    = &ili9881c_720p_video_panel_data;
        panelstruct->paneldata->panel_with_enable_gpio = 0;
        panelstruct->panelres     = &ili9881c_720p_video_panel_res;
        panelstruct->color        = &ili9881c_720p_video_color;
        panelstruct->videopanel   = &ili9881c_720p_video_video_panel;
        panelstruct->commandpanel = &ili9881c_720p_video_command_panel;
        panelstruct->state        = &ili9881c_720p_video_state;
        panelstruct->laneconfig   = &ili9881c_720p_video_lane_config;
        panelstruct->paneltiminginfo
            = &ili9881c_720p_video_timing_info;
        panelstruct->panelresetseq
                    = &ili9881c_720p_video_reset_seq;
        panelstruct->backlightinfo = &ili9881c_720p_video_backlight;
        pinfo->mipi.panel_on_cmds
            = ili9881c_720p_video_on_command;
        pinfo->mipi.num_of_panel_on_cmds
            = ILI9881C_720P_VIDEO_ON_COMMAND;
        pinfo->mipi.panel_off_cmds
            = ili9881c_720p_video_off_command;
        pinfo->mipi.num_of_panel_off_cmds
            = ILI9881C_720P_VIDEO_OFF_COMMAND;
        memcpy(phy_db->timing,
            ili9881c_720p_14nm_video_timings,
            MAX_TIMING_CONFIG * sizeof(uint32_t));
        //pinfo->dfps.panel_dfps = ili9881c_720p_video_dfps;
        pinfo->mipi.signature    = ILI9881C_720P_VIDEO_SIGNATURE;
        break;
```

5. Choose to use the new panel in function “oem_panel_select”. The “panel_id” will be passed to kernel.

```
    switch (hw_id) {
    case HW_PLATFORM_SURF:
    case HW_PLATFORM_MTP:
    case HW_PLATFORM_RCM:
        panel_id = ILI9881C_720P_VIDEO_PANEL;
        break;
    case HW_PLATFORM_QRD:
        switch (platform_subtype) {
            case QRD_SKUA:
                panel_id = HX8379A_FWVGA_SKUA_VIDEO_PANEL;
                break;
```

4 Configure Dual DSI

SC60 supports dual display. The chapters above describe the first display configuration. And the way to configure the *<oem_panel_input_file>.xml* of the second display is the same as that of the first display when the two displays are the same. But if the second display is different from the first one, the *<oem_panel_input_file>.xml* needs to be reconfigured and the steps are as below.

4.1. Configure Second Display

1. Add the second display in LK.

Modify the file *bootable/bootloader/lk/dev/gcdb/display/include/panel_ili9881c_720p_video.h*. Set the name of the second display slave_panel_node_id as *qcom,mdss_dsi_ili9881c_720p_dsi1_video* as shown below:

```
static struct panel_config ili9881c_720p_video_panel_data = {
    "qcom,mdss_dsi_ili9881c_720p_video",    /* panel_node_id */
    "dsi:0:",   /* panel_controller */
    "qcom,mdss-dsi-panel",  /* panel_compatible */
    10, /* panel_interface */
    0,  /* panel_type */
    "DISPLAY_1", /* panel_destination */
    0,  /* panel_orientation */
    0,  /* panel_clockrate */
    60, /* panel_framerate */
    0,  /* panel_channelid */
    0,  /* dsi_virtualchannel_id */
    0,  /* panel_broadcast_mode */
    0,  /* panel_lp11_init */
    0,  /* panel_init_delay */
    0,  /* dsi_stream */
    0,  /*  interleave_mode */
    0,  /* panel_bitclock_freq */
    0,  /* panel_operating_mode */
    0,  /* panel_with_enable_gpio */
    0,  /* mode_gpio_state */
    "qcom,mdss_dsi_ili9881c_720p_dsi1_video"    /* slave_panel_node_id */
};
```

After this, make aboot and burn the *emmc_appsboot.mbn*. Then reboot the module, and the second display name can be seen by the command below.

```
cat /proc/cmdline
```

```
msm8937_64:/ # cat /proc/cmdline
sched_enable_hmp=1 console=ttyHSL0,115200,n8 androidboot.console=ttyHSL0 androidboot.hardware=q
com msm_rtb.filter=0x237 ehci-hcd.park=3 lpm_levels.sleep_disabled=1 androidboot.bootdevice=782
4900.sdhci earlycon=msm_hsl_uart,0x78B0000 buildvariant=userdebug androidboot.emmc=true android
boot.verifiedbootstate=orange androidboot.veritymode=enforcing androidboot.keymaster=1 androidb
oot.serialno=c8b4513 androidboot.baseband=msm mdss_mdp.panel=1:dsi:0:qcom,mdss_dsi_ili9881c_720
p_video:1:qcom,mdss_dsi_ili9881c_720p_dsi1_video:cfg:dual_dsi
msm8937_64:/ #
```

2. Add new .dtsi file to device tree directory.

Path: *kernel/msm-3.18/arch/arm/boot/dts/qcom/dsi-panel-ili9881c-720p-dsi1-video.dtsi*

Customers can copy *dsi-panel-ili9881c-720p-video.dtsi* to *dsi-panel-ili9881c-720p-dsi1-video.dtsi*.

```
&mdss_mdp {
    dsi_ili9881c_720p_dsi1_video: qcom,mdss_dsi_ili9881c_720p_dsi1_video {
        qcom,mdss-dsi-panel-name = "ili9881c 720p dsi1 video mode dsi panel";
        //qcom,mdss-dsi-panel-controller = <&mdss_dsi0>;
        qcom,mdss-dsi-panel-type = "dsi_video_mode";
        //qcom,mdss-dsi-panel-destination = "display_1";
        qcom,mdss-dsi-panel-framerate = <60>;
        qcom,mdss-dsi-virtual-channel-id = <0>;
```

3. Add include file to msm8953-mdss-panels.dtsi.

Path: *kernel/msm-3.18/arch/arm/boot/dts/qcom/msm8953-mdss-panels.dtsi*

```
#include "dsi-panel-ili9881c-720p-video.dtsi"
#include "dsi-panel-ili9881c-720p-dsi1-video.dtsi"
#include "dsi-panel-lt8912-720p-video.dtsi"
```

Add DSI PHY 2.0.0 timing config.

```
&dsi_ili9881c_720p_dsi1_video {
    qcom,mdss-dsi-panel-timings-phy-v2 = [
        1e 1b 04 06 02 03 04 a0     /*Data 0*/
        1e 1b 04 06 02 03 04 a0     /*Data 1*/
        1e 1b 04 06 02 03 04 a0     /*Data 2*/
        1e 1b 04 06 02 03 04 a0     /*Data 3*/
        1e 0d 04 05 02 03 04 a0];   /*CLK lane*/
};
```

4. Change single display to dual DSI.

Path: *kernel/msm-3.18/arch/arm/boot/dts/qcom/msm8953-mtp.dtsi*

```
&mdss_dsi {
    //hw-config = "single_dsi";
    hw-config = "dual_dsi";
};
```

Parts to be modified when changing single display to dual display:

```
&mdss_dsi1 {
    status = "ok";
    qcom,dsi-pref-prim-pan = <&dsi_ili9881c_720p_dsi1_video>;
    pinctrl-names = "mdss_default", "mdss_sleep";
    pinctrl-0 = <&mdss_dsi1_active &mdss_te1_active>;
    pinctrl-1 = <&mdss_dsi1_suspend &mdss_te1_suspend>;

    qcom,bridge-index = <0>;
    qcom,pluggable;
    qcom,platform-te-gpio = <&tlmm 25 0>;
    qcom,platform-reset-gpio = <&tlmm 87 0>;
//  qcom,platform-bklight-en-gpio = <&tlmm 59 0>;    //not used
};
```

NOTE

Parameters “qcom,bridge-index = <0>;” and “qcom,pluggable;” cannot be modified.

Add node “dsi_ili9881c_720c_dsi1_video”.

```
&dsi_ili9881c_720p_dsi1_video {
    qcom,panel-supply-entries = <&dsi_panel_pwr_supply>;
    //qcom,mdss-dsi-pan-enable-dynamic-fps;
    //qcom,mdss-dsi-pan-fps-update = "dfps_immediate_porch_mode_vfp";
    //qcom,cont-splash-enabled;

    qcom,dba-panel;
    qcom,bridge-name = "dsi1-bridge";
};
```

5. Replace the display panel resolution and the porch values in header file.

Path: *kernel/msm-3.18/include/uapi/video/msm_hdmi_modes.h*

```
#define DEFAULT_VFRMT_TIMING                    \
    {DEFAULT_VFRMT, 720/*active_h*/, 52/*front_porch_h*/, 36/*pulse_width_h*/, 100/*back_porch_h*/, false,
     1280/*active_v*/, 8/*front_porch_v*/, 4/*pulse_width_v*/, 20/*back_porch_v*/, false, \
     71477/*pixel_freq/1000*/, 60000/*refresh_rate/1000*/, false, true, HDMI_RES_AR_16_9, 0}
#define HDMI_VFRMT_640x480p60_4_3_TIMING        \
    {HDMI_VFRMT_640x480p60_4_3, 640, 16, 96, 48, true,      \
     480, 10, 2, 33, true, 25200, 60000, false, true, HDMI_RES_AR_4_3, 0}
```

4.2. LCD Backlight

- SC60 supports to share LCD_BL_A with any of LCD_BL_K1, LCD_BL_K2, LCD_BL_K3 and LCD_BL_K4, which are described in the file *qcom,led-strings-list=[00 01 02 03]* under the directory of *arch/arm/boot/dts/qcom/msm-pmi8950.dtsi*.

```
qcom,fs-curr-ua = <20000>;
qcom,led-strings-list = [00 01 02 03];
```

Table 12: Pin Description of LCD Backlight

Pin No.	Pin Name	Description
21	LCD_BL_A	Current output for LCD backlight
22	LCD_BL_K1	Current sink for LCD backlight
23	LCD_BL_K2	Current sink for LCD backlight
24	LCD_BL_K3	Current sink for LCD backlight
25	LCD_BL_K4	Current sink for LCD backlight

- Also LCD_BL_A can be shared, LCD_BL_K1/LCD_BL_K2 provide one string of backlight control and LCD_BL_K3/LCD_BL_K4 provide another one. But the codes only support four strings of backlight control at the same time, thus the two strings of backlight control provided by LCD_BL_K1/LCD_BL_K2 and LCD_BL_K3/LCD_BL_K4 separately are not supported.

- The WLED current can be configured in the .dtsi file, and the max value is 30000.

Path: *kernel/msm-3.18/arch/arm/boot/dts/qcom/msm-pmi8950.dtsi*

```
wled: qcom,leds@d800 {
    compatible = "qcom,qpnp-wled";
    reg = <0xd800 0x100>,
        <0xd900 0x100>,
        <0xdc00 0x100>,
        <0xde00 0x100>;
    reg-names = "qpnp-wled-ctrl-base",
            "qpnp-wled-sink-base",
            "qpnp-wled-ibb-base",
            "qpnp-wled-lab-base";
    interrupts = <0x3 0xd8 0x2>;
    interrupt-names = "sc-irq";
    status = "okay";
    linux,name = "wled";
    linux,default-trigger = "bkl-trigger";
    qcom,fdbk-output = "auto";
    qcom,vref-mv = <350>;
    qcom,switch-freq-khz = <800>;
    qcom,ovp-mv = <29500>;
    qcom,ilim-ma = <980>;
    qcom,boost-duty-ns = <26>;
    qcom,mod-freq-khz = <9600>;
    qcom,dim-mode = "hybrid";
    qcom,dim-method = "linear";
    qcom,hyb-thres = <625>;
    qcom,sync-dly-us = <800>;
    qcom,fs-curr-ua = <20000>;
    qcom,led-strings-list = [00 01];
    qcom,en-ext-pfet-sc-pro;
    qcom,cons-sync-write-delay-us = <1000>;
};
```

5 Appendix A Reference

Table 13: Terms and Abbreviations

Abbreviation	Description
DCS	Display Command Set
DSI	Display Serial Interface
GCDB	Global Component Database
GPIO	General Purpose Input Output
LCD	Liquid Crystal Display
LK	Little Kernel
LPG	Light Pulse Generator
MDP	Mobile Display Processor
PMIC	Power Management IC
PWM	Pulse Width Modulation
TE	Tearing Effect
WLED	White Light Emitting Diode