AN14187

Display settings for Win10 IoT Enterprise

Rev. 1 — 29 May 2024 Application note

Document information

Information	Content
Keywords	Windows IoT, Display, MIPI-DSI, LVDS, HDMI, EDID file, LCD, GPU
Abstract	This document describes how to change settings and configure custom displays on Windows 10 IoT BSP on i.MX SoC.

1 Introduction

This document provides a description of how to change settings and configure custom displays on Windows 10 IoT BSP on i.MX SoC.

Supported SoC:

- i.MX 8M Nano (i.MX 8MN)
- i.MX 8M Mini (i.MX 8MM)
- i.MX 8M Plus EVK (i.MX 8MP)
- i.MX 8M Quad (i.MX 8MQ)
- i.MX 8QuadXPlus (i.MX 8QXP)
- i.MX 93

2 Display support

This chapter gives detailed information about display support.

2.1 Display support in Windows

Table 1. Display support for BSP 1.5.0

Platform	Display option 1 [U-Boot, UEFI, Windows]	Display option 2 [U-Boot, UEFI, Windows]	Display option 3 [U-Boot, UEFI, Windows]	Multi-monitor [available, supported]
i.MX 8MN	1x MIPI-DSI [yes, yes, yes] [1]	-	-	1, 1
i.MX 8MM	1x MIPI-DSI [yes, yes, no] [2]	-	-	1, 1 (UEFI only)
i.MX 8MP	1x MIPI-DSI [yes, yes, yes][1]	1x LVDS [no, yes, yes][1]	1x HDMI [no, yes, yes][1]	3, 3 (HDMI+LVDS+MIPI)
i.MX 8MQ	1x MIPI-DSI [no, no, no][2]	1x HDMI [yes, no, yes][1]	-	2, 1 (HDMI)
i.MX 8QXP	2x MIPI-DSI [no, no, no][2]	2x LVDS [yes, yes, yes][1]	1x Parallel RGB [no, no, no][2]	3, 2 (LVDS0 + LVDS1)
i.MX 93	1x MIPI-DSI [yes, yes, no][2]	1x LVDS [no, yes, no][2]	1x Parallel RGB [no, no, no][2]	1, 1 (UEFI only)

[1] The Windows GPU driver is supported.
[2] The Windows GPU driver is not supported.

Note:

- *In the display option X column, the display is available on the SoC. In the brackets, the driver support in U-Boot, UEFI, and Windows is mentioned.*
- *In the multi-monitor column, the first number is the number of displays that can run at the same time on the SoC, the second number is the number of displays supported by the Windows driver that can run at the same time.*

2.2 Display support – maximum resolution

Table 2. Display support – maximum resolution

Platform	Display option 1 [max resolution, *supported*]	Display option 2 [max resolution, *supported*]	Display option 3 [max resolution, *supported*]
i.MX 8MN	MIPI-DSI [1920x1200@60]	-	-
i.MX 8MM	MIPI-DSI [1920x1200@60]	-	-
i.MX 8MP	MIPI-DSI [1]	LVDS [2]	HDMI [3840x2160@30, *1920x1080@60*]
i.MX 8MQ	MIPI-DSI [[3] , *N/A*]	HDMI [4096x2160p60, *1920x1080@60*]	-
i.MX 8QXP	MIPI-DSI [1920x1200@60, *N/A*]	LVDS [1920x1080p60]	Parallel RGB [1280x720@60, *N/A*]
i.MX 93	MIPI-DSI [1920x1200@60]	LVDS [1280x800@60]	Parallel RGB [1280x800@60, *N/A*]

[1] Supports up to 1920x1200@60 display per LCDIF if no more than 2 instances are used simultaneously, or 2x 1920x1080@60 + 1x 3840x2160@30 on HDMI if all 3 instances are used simultaneously.
[2] Single-channel 1280x800@60, Dual-channel – see option [1]
[3] From DCSS (HDMI not used), theoretically max 250MHz pclk (2560x1440@60), from LCDIF 1920x1080p60

Options in italics highlight Windows/UEFI driver limitations in comparison to max resolution.

3 Display selection

This chapter gives detailed information about display selection.

3.1 UEFI driver

The autodetection priorities are listed below. The first is the MIPI-DSI-to-HDMI converter detection. If it fails to initialize, the next option will be tried, and so on.

It is possible to change the order of priority in the file `iMX8LcdHwLib.c` for i.MX 8MP, i.MX 8MM, and i.MX 8MN. And in the file `iMX93DisplayHwLib.c` for i.MX 93. These files are at `\mu_platform_nxp\Silicon\ARM\NXP\iMX8Pkg\Library\iMX8LcdHwLib\`

Autodetection in the order of priority:

1. MIPI-DSI-to-HDMI converter (IMX-MIPI-HDMI, ADV7535)
2. LVDS-to-HDMI converter (IMX-LVDS-HDMI, IT6263)
3. Native HDMI (i.MX 8MP)
4. Display Interface defined by parameter `giMX8TokenSpaceGuid.PcdDisplayInterface` (in file `<Platform>.dsc`)

LVDS, MIPI-DSI displays typically fit to point 4 above.

For other dsc parameters, see the description at `\mu_platform_nxp\NXP\<board>`.

`PcdDisplayI2CBaseAddr` – I2C base address for MIPI or LVDS to HDMI converter.

`PcdDisplayReadEDID` – TRUE/FALSE – enable/disable reading EDID is available for the HDMI based interface (Native HDMI and converters IMX-MIPI-HDMI, IMX-LVDS-HDMI).

3.2 Windows driver

Below is an example of how to set up two display interfaces. For example, on the i.MX 8MP. This example sets `Display0Interface` to LVDS and `Display1Interface` to HDMI. It can be set for SoCs listed in the Display support in the Windows table.

The display is selected by the `Display0Interface` parameter for the first display, by the `Display1Interface` parameter for the second display, and so on. There are two options for setting up the interfaces. Use one of the following procedures for this modification:

1. Update `galcore.inf` and uninstall/re-install the GPU driver.

`Galcore.inf` update:

```
[GcWddmMP_AddReg] // Find appropriate platform (MP, MN, 8X)
; Enable support for multiple monitors
HKR,,EnableMultiMon,%REG_DWORD%,1 // Enable multiple monitors
; Display parameters for LVDS interface
HKR,,Display0Interface,%REG_DWORD%,%DISP_INTERFACE_LVDS0% //Select LVDS0 (first
 display)
HKR,,Display1Interface,%REG_DWORD%,%DISP_INTERFACE_HDMI% // Select HDMI (second
 display)
;Possible values for DisplayInterfaces,
DISP_INTERFACE_DISABLED = 0x0
DISP_INTERFACE_HDMI = 0x1
DISP_INTERFACE_MIPI_DSI0 = 0x2
DISP_INTERFACE_MIPI_DSI1 = 0x3
DISP_INTERFACE_LVDS0 = 0x4
DISP_INTERFACE_LVDS1 = 0x5
DISP_INTERFACE_LVDS_DUAL0 = 0x6
DISP_INTERFACE_PARALLEL_LCD = 0x7
```

2. Update the Registry database on the target and restart the GPU driver:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-
e325-11cebfc1-08002be10318}\0000
```

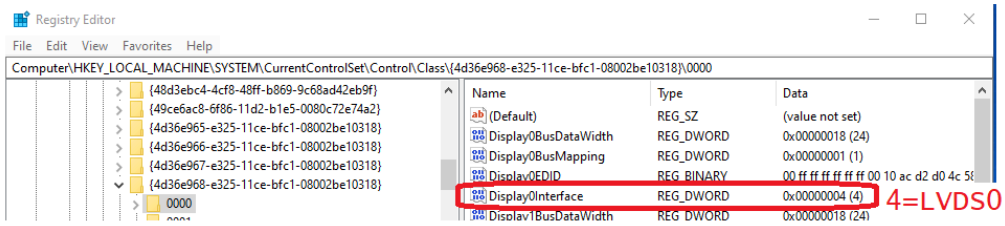

Figure 1. Registry update

The display interface is determined only by the Windows Registry:

1. LVDS interface:
   - If the LVDS-HDMI converter (IMX-LVDS-HDMI) is detected, initialize it.
   - Otherwise, initialize the custom LVDS display.
2. MIPI-DSI interface:
   - If the MIPI-HDMI converter (IMX-MIPI-HDMI) is detected, initialize it.
   - Otherwise, initialize the IMX-DSI-OLED1 NXP testing panel. See more details in the MIPI-DSI driver customization paragraph below.
3. Native HDMI interface

MIPI-DSI driver customization:

- The MIPI-DSI panel, IMX-DSI-OLED1 NXP testing panel. There is a template driver for this panel that must be customized. To customize it, follow the steps below:
  - Find the `panel-raydium-rm67191.c` driver at `\imx-Windows-iot\driver\display\dispdll\mipi_dsi\`.
  - Customize the following four functions of the driver according to the needs of a new panel being supported.
    For example, for the NXP IMX-DSI-OLED1 panel:

```
static const struct drm_panel_funcs
rad_panel_funcs = {
 .prepare = rad_panel_prepare,
 .unprepare = rad_panel_unprepare,
 .enable = rad_panel_enable,
 .disable = rad_panel_disable,
};
```

The calling sequence of the functions and their content is the following:
`rad_panel_probe` – allocate the memory for the driver object, set DSI formats, register panel callbacks
`rad_panel_prepare` – for example, panel reset deassert
`rad_panel_enable` – for example, the initialization sequence in the DSI low-power communication before the frame generator is started
`rad_panel_disable` – for example, disable sequence after the frame generator is stopped
`rad_panel_unprepare` – for example, panel reset assert
`rad_panel_remove` – free the driver object

4 Display resolution

This chapter gives detailed information about display resolution.

4.1 Display resolution terminology

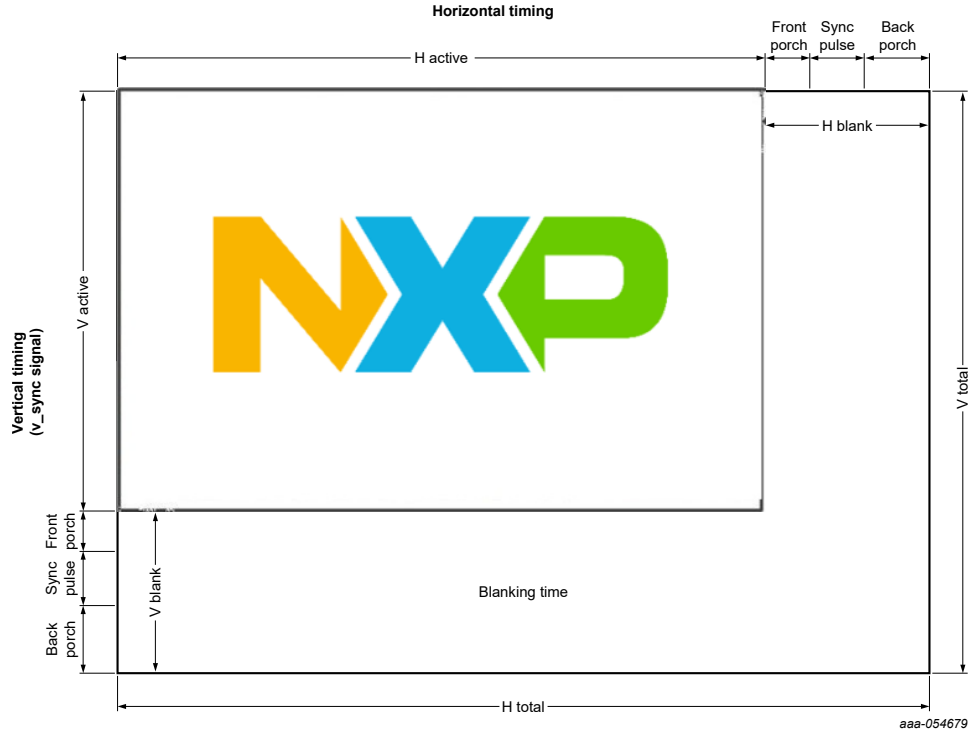

Figure 2. Display resolution terminology

- Front Porch = Sync Offset = FP
- Sync Pulse = Sync = Sync Len = Sync width = SW
- Back Porch = BP
- VRefresh = PixelClock / (VTotal + HTotal)
- HRefresh = PixelClock / HTotal

4.2 Changing display resolution - UEFI driver

The default resolution is set to 1920x1080. To change to a custom resolution, edit the files below:

For i.MX 8MN, i.MX 8MM, and i.MX 8MP edit the file `iMX8LcdHwLib.c` at `\mu_platform_nxp\Silicon\ARM\NXP\iMX8Pkg\Library\iMX8LcdHwLib\`.

For i.MX 93 edit the file `iMX93DisplayHwLib.c` at `\mu_platform_nxp\Silicon\ARM\NXP\iMX8Pkg\Library\iMX8LcdHwLib\`.

Changing display resolution in the UEFI driver:

```
/* Preferred timing mode. if PcdDisplayReadEDID == TRUE, it is overwritten with
 edid data */
IMX_DISPLAY_TIMING PreferredTiming;

/* Predefined modes - one selected is copied to PreferredTiming in
 LcdDisplayDetect */
/* 1080x1920@60Hz */
const IMX_DISPLAY_TIMING PreferredTiming_1080x1920_60 = {
 .PixelClock = 108000000,
  .HActive = 1080,
```

```
   .HBlank = 56,
   .VActive = 1920,
   .VBlank = 16,
   .HSync = 2,
   .VSync = 2,
   .HSyncOffset = 34,
   .VSyncOffset = 4,
   .HImageSize = 296,
   .VImageSize = 527,
   .HBorder = 0,
   .VBorder = 0,
   .EdidFlags = 0,
   .Flags = 0,
   .PixelRepetition = 0,
   .Bpp = 24,
   .PixelFormat = PIXEL_FORMAT_ARGB32,
};
//* Update the values highlighted in bold according to the display documentation
LcdDisplayDetect ( // Assign PreferredTiming in this function
  VOID
  )
….
/* Converter was not detected - select fixed default timimng */
  if (converter == transmitterUnknown) {
    if (displayInterface == imxMipiDsi) {
      videoModesCnt++;
      LcdInitPreferredTiming
(&PreferredTiming_1080x1920_60,&PreferredTiming);
 // For MIPI-DSI
      DEBUG((DEBUG_ERROR, "Mipi-dsi
 display interface. Default resolution used.
 %dx%d pclk=%d Hz\n",
           PreferredTiming.HActive,
 PreferredTiming.VActive,
 PreferredTiming.PixelClock));
      LcdDumpDisplayTiming(0,
 &PreferredTiming);
      return EFI_SUCCESS;
    } else if ((displayInterface == imxLvds0)
 || (displayInterface == imxLvds1) ||
 (displayInterface == imxLvds0dual)) {
      videoModesCnt++;
      LcdInitPreferredTiming
 (&PreferredTiming_1280x720_60,
 &PreferredTiming); // For LVDS
      DEBUG((DEBUG_ERROR, "LVDS%d
 display interface. Default resolution used.
 %dx%d pclk=%d Hz\n",
           displayInterface-2,
 PreferredTiming.HActive,
 PreferredTiming.VActive,
 PreferredTiming.PixelClock));
      LcdDumpDisplayTiming(0,
 &PreferredTiming);
      return EFI_SUCCESS;
    } else if (displayInterface ==
 imxNativeHdmi) {
```

`iMX8LcdHwLib.c` - only MIPI-DSI display.

For the MIPI-DSI display, there is one more step needed.

```
EFI_STATUS
LcdSetMode ( // Find this function
  IN  UINT32     ModeNumber
  )
 {
  IMX_DISPLAY_TIMING *Timing =
 &PreferredTiming;

  if (ModeNumber >= videoModesCnt) {
    return EFI_INVALID_PARAMETER;
  }

  if (displayInterface == imxMipiDsi) { // For MIPI-DSI only
  /*-------------------------
MIPI-------------------------------------------*/
    /* Mipi DSI set timing mode */

    CHECK_STATUS_RETURN_ERR(MipiDsiConfig(Timing, converter), "MIPI DSI
 config");
    if (converter == ADV7535) {
      /* ADV7535 set timing mode */

      CHECK_STATUS_RETURN_ERR(Adv7535SetMode(Timing), "ADV7535 config");
    } else {
      /* MIPI-DSI panel init must be called after MipiDsiConfig() */

      CHECK_STATUS_RETURN_ERR(Rm67191Init(
      displayInterface), "RM67191 config");
    } // Initialization for NXP IMX-DSI-OLED panel. Delete or put custom init in
 there, if needed.
```

4.3 Custom settings

Below is an example of determining the timing parameters from the documentation for the display that you want to configure.

For video modes supported by the hardware used, see Video Timings Calculator

4.3.1 Display EV121WXM-N12

Display EV121WXM-N12: LVDS panel 1280x800, accessory for i.MX 93 EVK industrial.

Table 3. An example for timing parameters from the documentation

Item			Symbol	Min	Typ	Max	Unit
LCD	Frame rate		-	58	60	62	Hz
	Pixels rate		-	66.3	72.4	78.9	MHz
Timing	Horizontal	Horizontal total time	tHP	1380	1440	1500	t_{CLK}
		Horizontal active time	tHadr	1280			t_{CLK}
		Horizontal back porch	tHBP	-	80	-	t_{CLK}

Table 3. An example for timing parameters from the documentation *...continued*

Item			Symbol	Min	Typ	Max	Unit
		Horizontal front porch	tHFP	-	48	-	t_{CLK}
	Vertical	Vertical total time	tvp	824	838	872	t_H
		Vertical active time	tVadr	800			t_H
		Vertical back porch	tVBP	-	14	-	t_H
		Vertical front porch	tVFP	-	9	-	t_H
Lane				-	1	-	Lane

List of parameters that can be read or computed from Table 3 :

- PixelClock = 72.4 MHz
- HTotal = 1440
- HActive = 1280
- HSyncOffset = 48
- HBackPorch = 80
- HBlank = HTotal - HActive = 160
- HSync = HBlank - HBackPorch - HFrontPorch = 32
- VTotal = 838
- VActive = 800
- VSyncOffset = 9
- VBackPorch = 14
- VBlank = VTotal - VActive = 38
- VSync = VBlank - VBackPorch - VFrontPorch = 15

4.3.2 Display Avnet AMA-121A01-DU2511-G010

Display Avnet AMA-121A01-DU2511-G010 LVDS panel 1280x800

Table 4. An example for timing parameters from the display documentation (2)

Parameter	Symbol	Min	Typ	Max	Unit
CLK frequency	$1/t_c$	67	71	75	MHz
Horizontal display area	thd	-	1280	-	tc
Horizontal period	th	1290	1440	-	tc
Vertical display area	tvd	-	800	-	th
Vertical period	tv	810	823	-	th
Frame rate	F	-	60	-	Hz
VDD=3.3V, GND=0V, Ta=25°C					

Not all parameters can be determined directly from Table 4

- PixelClock = 71 MHz
- HTotal = 1440
- HActive = 1280
- HFP = ??
- HBP = ??
- HBlank = HTotal - HActive = 160
- HSync = ??
- VTotal = 823
- VActive = 800
- VFP = ??
- VBP = ??
- VBlank = VTotal - VActive = 23
- VSync = ??

For determining missing parameters, use the VESA Coordinated Video Timings Standard (CVT).

Horizontal Sync Pulse duration and position

The Horizontal Sync Pulse duration is in all cases 32 pixel clocks in duration, with the position set so that the trailing edge of the Horizontal Sync Pulse is located in the center of the Horizontal Blanking period. This implies that the Horizontal Back Porch is fixed to 80 pixel clocks.

- HSYNC = 32
- HBP = 80
- HFP = HBlank – HBP – Hsync = 48

Vertical Sync Pulse duration and position

The Vertical Front Porch is in all cases fixed to three lines. The Vertical Back Porch must be the remainder of the Vertical Blanking Time.

- VSync = 6
- VFP = 3
- VBP = VBlank – VBP – VSync = 14

Table 5. Vertical Sync Duration

Vertical Sync Width	Aspect Ratio
3 or less	Not used by CVT, reserved for existing DMT and GFT
4	4:3
5	16:9
6	16:10

Alternatively, to determine missing parameters, one of the online calculators can be used, see Video Timings Calculator.

4.4 Windows driver

For HDMI, the EDID data with resolution and timing parameters settings are obtained from the display interface. For all others (MIPI-DSI and LVDS), it must be in the registers.

Resolution and timing parameters encoded in the EDID binary data are in the `Display0EDID` parameter for the first display, in the `Display1EDID` parameter for the second display, and so on. To use the EDID editor, see Section 4.5 for EDID settings.

Use one of the following procedures for this modification:

1. Update `galcore.inf` and uninstall/re-install the GPU driver.
   `Galcore.inf` update:

```
[GcWddmMN_AddReg] // Find appropriate platform (MP, MN, 8X)
…
;EDID - 128 bytes total length expected
;1280x720@60
HKR,,Display0EDID,%REG_BINARY%,\
0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x10,0xAC,0x7A,0xA0,0x53,0x4B,0x35,0x32,\
0x1E,0x1A,0x01,0x03,0x80,0x34,0x20,0x78,0xEA,0xEE,0x95,0xA3,0x54,0x4C,0x99,0x26,\
0x0F,0x50,0x54,0xA1,0x08,0x00,0x81,0x40,0x81,0x80,0xA9,0x40,0xB3,0x00,0xD1,0xC0,\
0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x1D,0x00,0x72,0x51,0xD0,0x1E,0x20,0x6E,0x28,\
0x55,0x00,0x40,0x44,0x21,0x00,0x00,0x1A,0x00,0x00,0x00,0xFF,0x00,0x59,0x50,0x50,\
0x59,0x30,0x36,0x37,0x56,0x32,0x35,0x4B,0x53,0x0A,0x00,0x00,0x00,0xFC,0x00,0x44,\
0x45,0x4C,0x4C,0x20,0x55,0x32,0x34,0x31,0x32,0x4D,0x0A,0x20,0x00,0x00,0x00,0xFD,\
0x00,0x32,0x3D,0x1E,0x53,0x11,0x00,0x0A,0x20,0x20,0x20,0x20,0x20,0x20,0x00,0x6D
```

2. Update the Registry database on the target and restart the GPU driver:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-
e325-11cebfc1-08002be10318}\0000
```

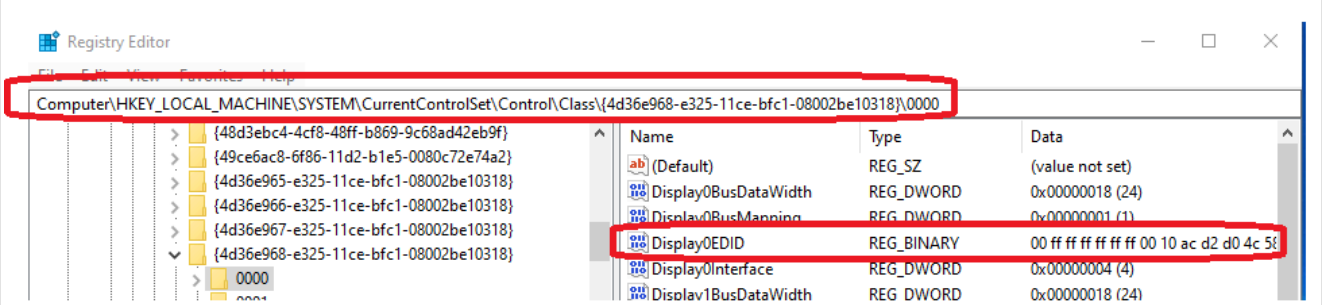

Figure 3. Registry update

4.5 EDID settings

To edit EDID binary data, it is possible to use a suitable editing program. In this example, we use the DELTACAST editor; however, any other EDID editor available online can be used for the same purpose.

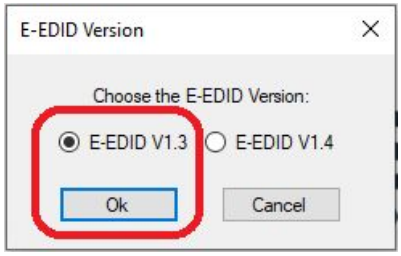

Figure 4. E-DID version selection

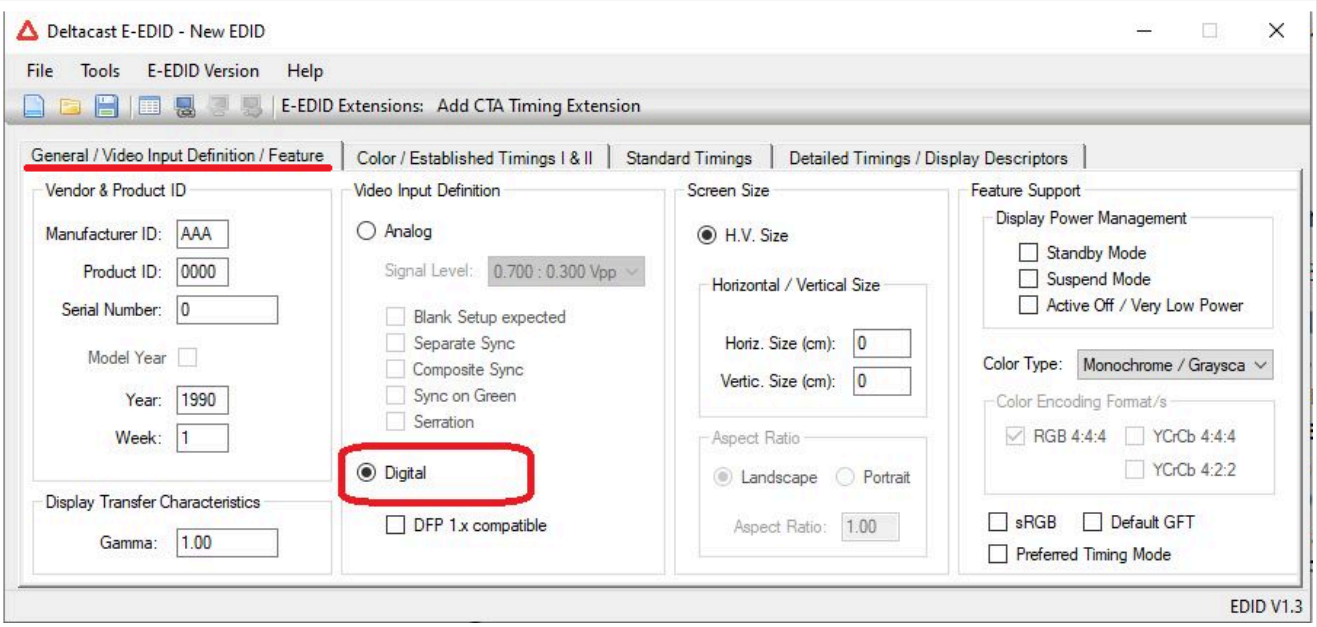

Figure 5. E-DID general settings

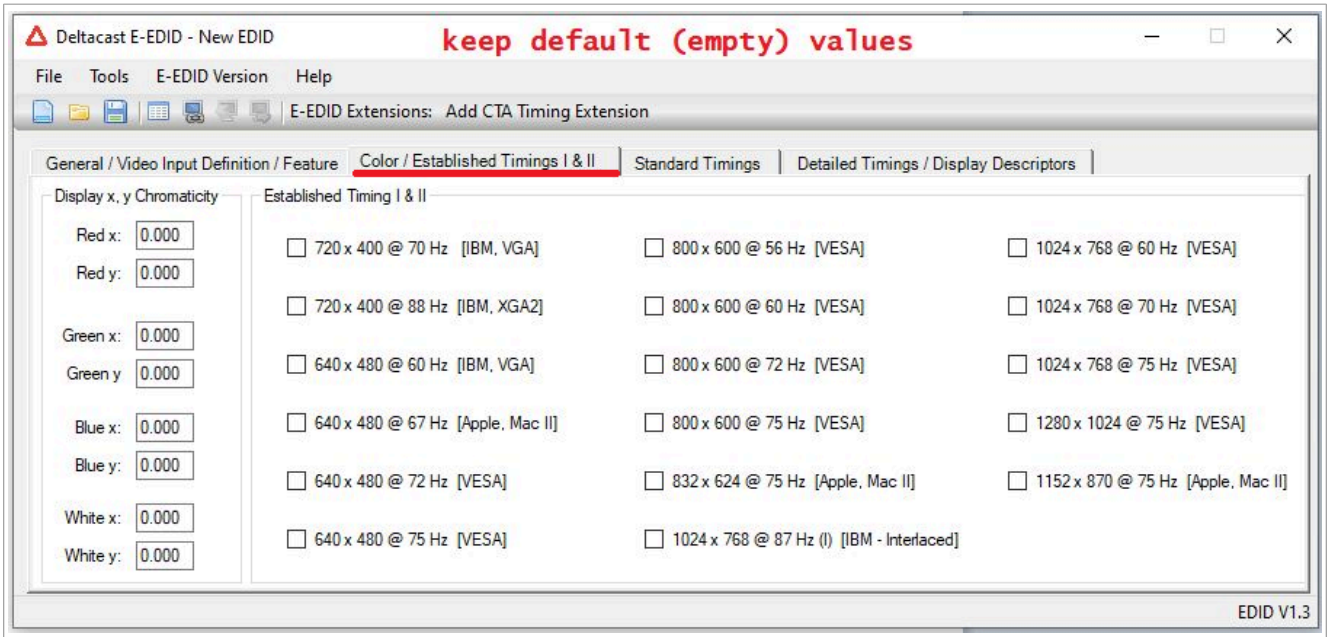

Figure 6. E-DID color and timing settings

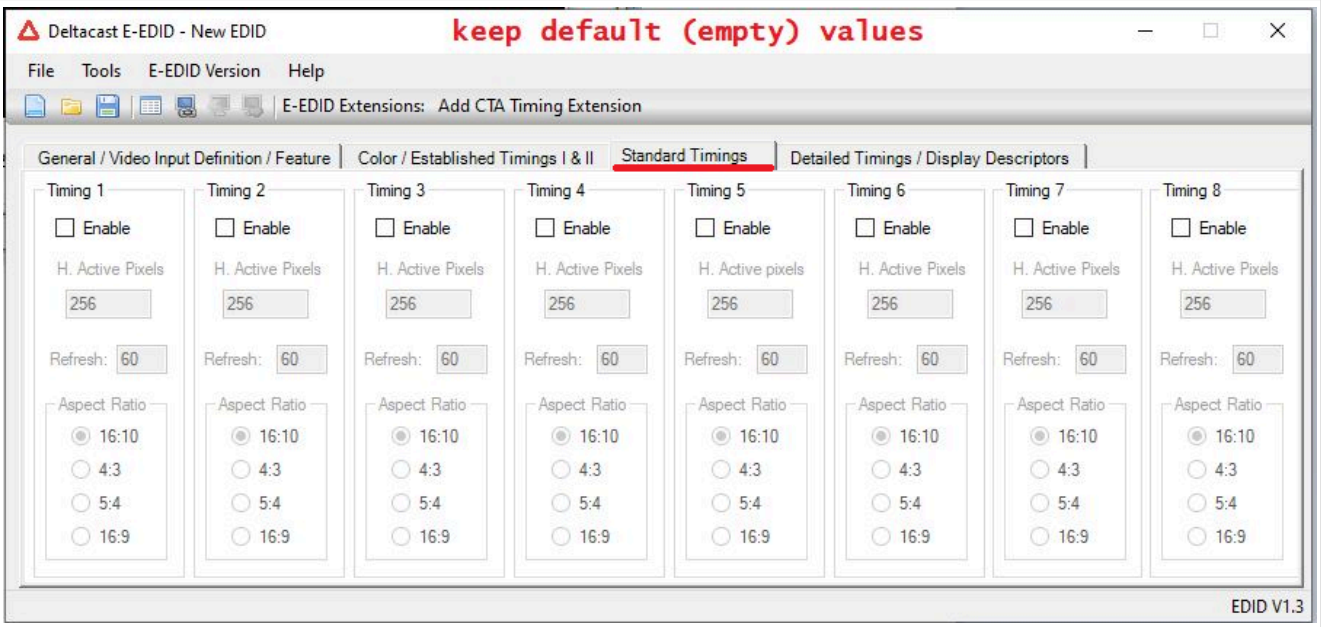

Figure 7. E-DID standard timing settings

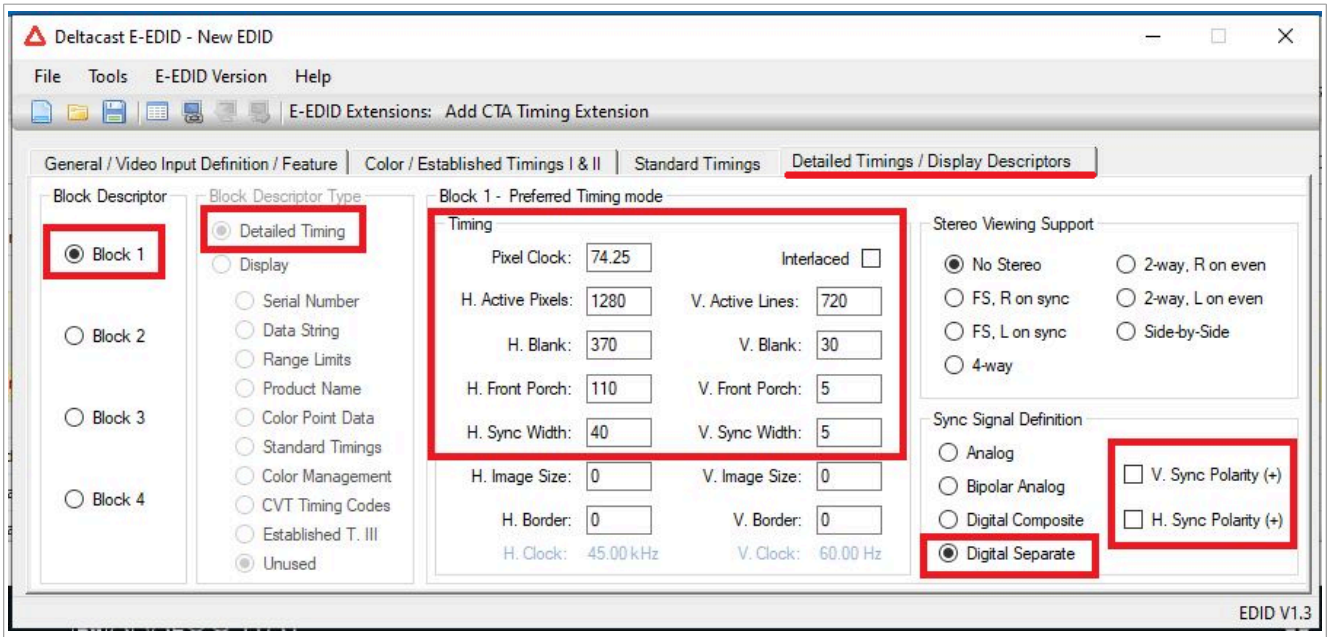

Figure 8. E-DID detailed timing settings

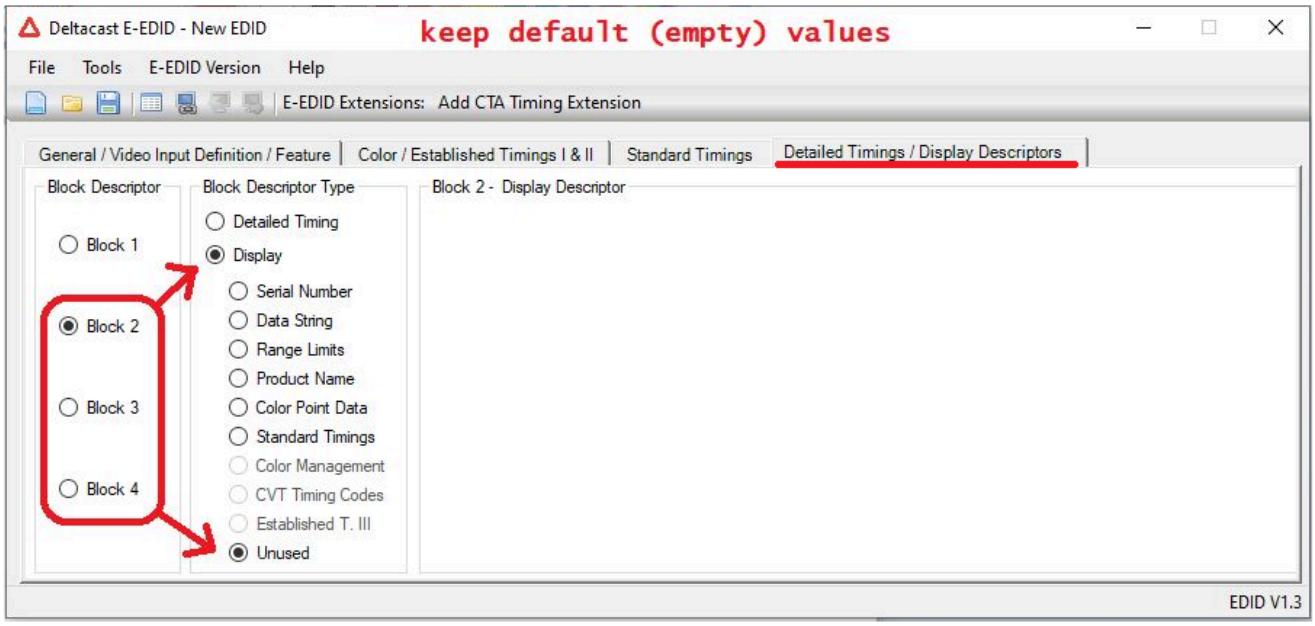

Figure 9. E-DID display block selection

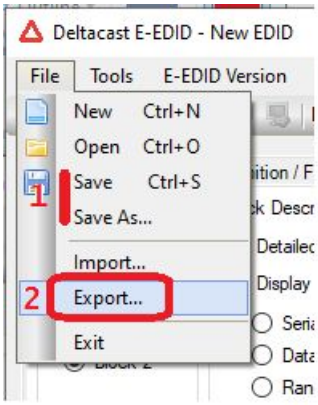

Figure 10. E-DID settings export

Once exported, open the `.dat` file and edit the string to look as in the `galcore.inf` file.

5 LVDS signal

This chapter gives detailed information about the LVDS signal and the LVDS display.

5.1 Workaround for 3 lanes from 4 lanes

Some displays support 3 lanes (18 bpp) instead of 4 lanes (24 bpp). For details, see the sections below.

5.1.1 24 bpp and 18 bpp

Figure 11 describes the difference between 24 bpp and 18 bpp for LVDS:

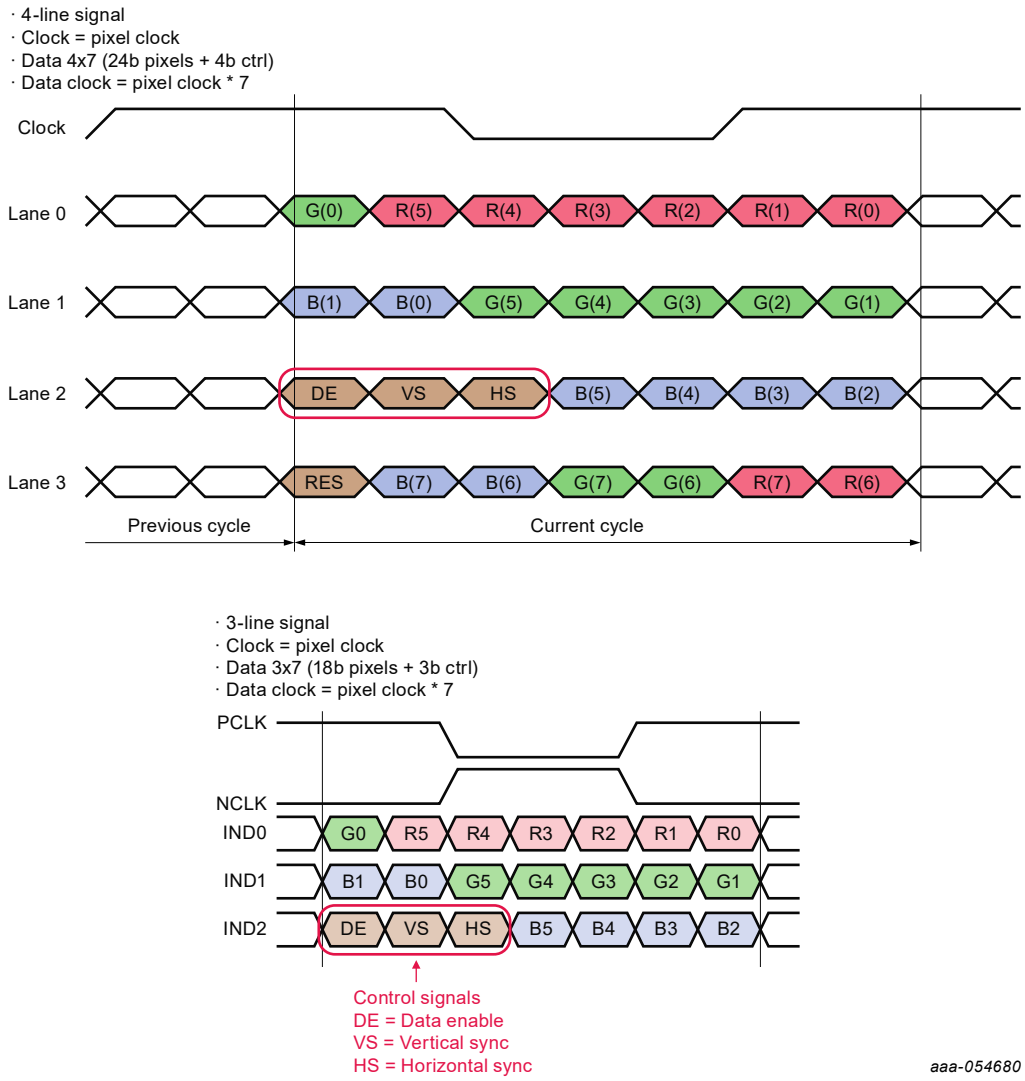

Figure 11. Difference between 24 bpp and 18 bpp for LVDS

5.1.1.1 VESA and JEIDA

As illustrated below, there are two different standards for 24 bpp mapping. As default, NXP EVK boards use the VESA standard. However, the settings could be switched to JEIDA.

Table 6. SPWG/PSWGNESA 24 bpp data mapping

Serializer input	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
data0	G0	R5	R4	R3	R2	R1	R0
data1	B1	B0	G5	G4	G3	G2	G1
data2	DE	VSYNC	HSYNC	B5	B4	B3	B2
data3	CTL	B7	B6	G7	G6	R7	R6

Table 7. JEIDA 24 bpp data mapping

Serializer input	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
data0	G2	R7	R6	R5	R4	R3	R2
data1	B3	B2	G7	G6	G5	G4	G3
data2	DE	VSYNC	HSYNC	B7	B6	B5	B4
data3	CTL	B1	B0	G1	G0	R1	R0

5.1.2 Mapping from 24 bpp to 18 bpp

i.MX 8QXP and i.MX 8MP have selection between 24 bpp and 18 bpp, otherwise use a workaround below:

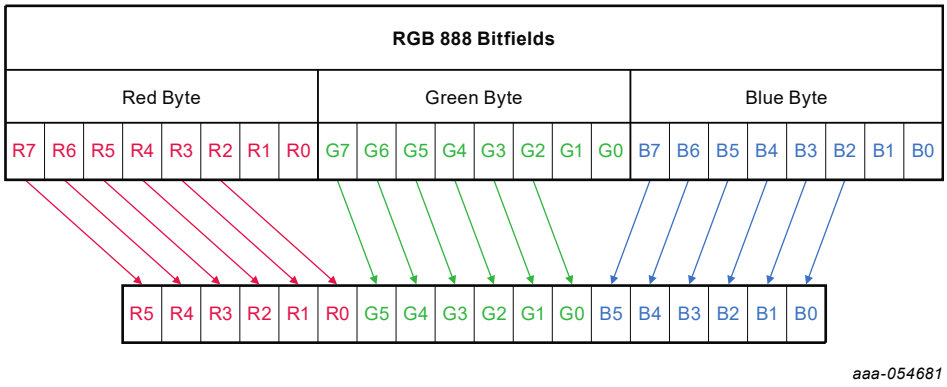

Figure 12. Bit mapping between 24 bpp signal and 18 bpp signal

Make the following changes in JEIDA 24 bpp data mapping and LSB will be discarded:

Table 8. SPWG/PSWG/VESA 24 bpp data mapping

Serializer input	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
data0	G0	R5	R4	R3	R2	R1	R0
data1	B1	B0	G5	G4	G3	G2	G1
data2	DE	VSYNC	HSYNC	B5	B4	B3	B2
data3	CTL	B7	B6	G7	G6	R7	R6

Table 9. JEIDA 24 bpp data mapping

Serializer input	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
data0	G2	R7	R6	R5	R4	R3	R2
data1	B3	B2	G7	G6	G5	G4	G3
data2	DE	VSYNC	HSYNC	B7	B6	B5	B4
data3	CTL	B1	B0	G1	G0	R1	R0

Note: *Ignoring the fourth line of the 24 bpp JEIDA signal produces a valid 3-line 18 bpp VESA signal. As shown in Figure 12, R7 signal goes to R5 signal, G7 signal goes to G5 signal and so on. This means that for NXP i.MX EVK boards, it is necessary to switch from VESA to JEIDA and reduce to three data channels.*

5.2 LVDS signal settings

In the case of changes related to LVDS, it is necessary to edit both the UEFI driver and the Windows driver.

5.2.1 UEFI driver

Platform-specific LVDS driver source code must be updated based on the EVK board used.

For the register description, see the SoC reference manual:

i.MX 8MP: the UEFI driver, the `ldb.c` file, the `LcdConfig()` function.

i.MX 93: the UEFI driver, the `iMX9xLvds.c` file, the `LcdEnable()` function.

The abovementioned files are at `\mu_platform_nxp\Silicon\ARM\NXP\iMX8Pkg\Library\iMX8LcdHwLib\`.

In BSP 1.5.0 i.MX 8QXP UEFI LVDS driver was already added.

i.MX 8QXP: the UEFI driver, the `ldb_imx8x.c` file, the `Imx8xLdbConfigure` function.

Below is an example of how to change to JEIDA format for the i.MX 93 platform:

```
EFI_STATUS LdbEnable(IN INTN Ldb, IN
CONST IMX_DISPLAY_TIMING *Timing)
{
  if (Ldb < 0 || Ldb >= LVDS_MAX_DEV) {
    return EFI_DEVICE_ERROR;
  }
  (VOID)Timing;
  /*
  * Leave default negative polarity, SPWG
 mapping,
  * set 24bit data width, LDB data always
 from source 0.
  */
  MmioWrite32(BasePtrs[Ldb] + LDB_CTRL,
 CH0_ENABLE | CH0_DATA_WIDTH | CH0_BIT_MAPPING
  return EFI_SUCCESS;
}
```

5.2.2 Windows driver

The LVDS signal properties are set by the `Display0BusDataWidth` parameter for the first display; the value is 18 or 24. The `Display0BusMapping` parameter for the first display; the value is 1 (VESA) or 2 (JEIDA). There are two options:

1. Update `galcore.inf` and uninstall/re-install the GPU driver.
   `Galcore.inf` update

```
[GcWddmMP_AddReg] // Find
appropriate platform (MP, MN, 8X)
…
;Following parameters relevant do LVDS interface
HKR,,Display0BusDataWidth,%REG_DWORD%,24
HKR,,Display0BusMapping,%REG_DWORD%,%DISP_BUS_MAPPING_SPWG%
```

2. Update the Registry database on the target and restart the GPU driver: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-e325-11cebfc1-08002be10318}\0000`

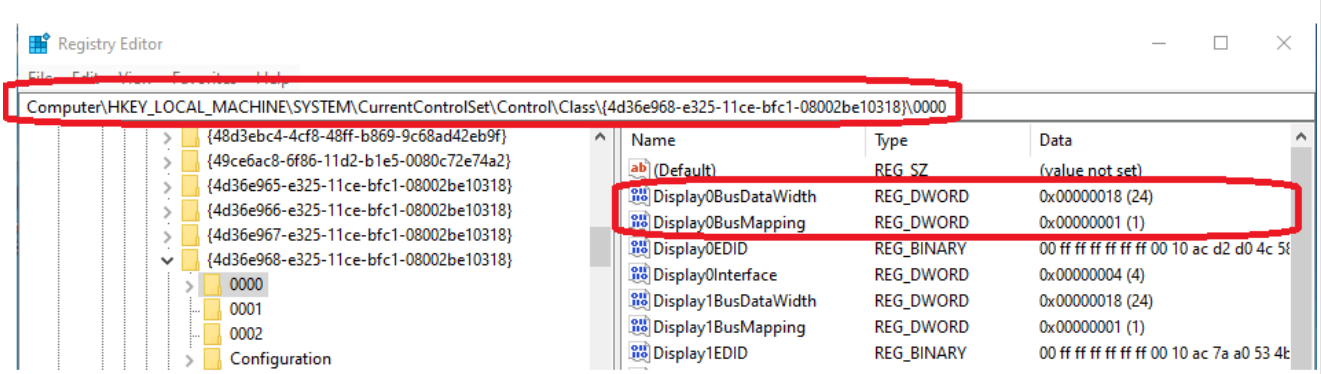

Figure 13. Registry update

5.3 Changing the backlight

Setting the backlight on the display is optional. Only the ON or OFF state is available. It is not possible to change the brightness intensity.

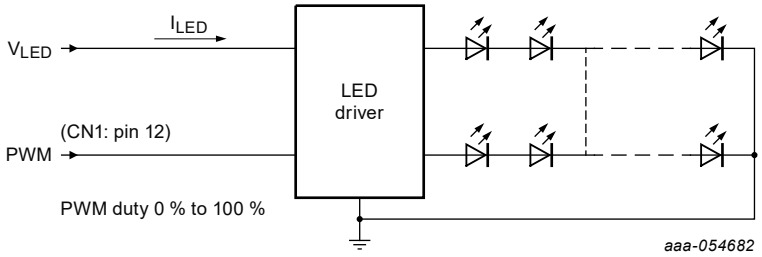

Figure 14. Example of backlight block diagram

Table 10 is an example for EV121WXM-N12: LVDS panel 1280x800, accessory for the i.MX 93 EVK industrial.

Table 10. Pin routing for EV121WXM-N12 LVDS panel

Pin number	Symbol	Description	Remarks
1	PWM	Luminance control	
2	BRTC	Backlight ON/OFF control	High or Open: Backlight ON Low: Backlight OFF
3	GND	Ground	
4	GND	Ground	
5	VDD	Power supply	
6	VDD	Power supply	

Table 10 suggests that the BRTC pin must be High and the PWM pin determines the brightness intensity. For full brightness, both pins must be High. A suitable place for the pin setting is the `BoardInit.c` driver in firmware. There is an example of how to make pin routing and setting there. For example, for the i.MX 8MP platform, the BoardInit file is located in `/mu_platform_nxp/NXP/MX8M_PLUS_EVK/Library/iMX8BoardLib/iMX8BoardInit.c`.

6 MIPI-DSI settings

In the case of changes related to MIPI-DSI, it is necessary to edit both the UEFI driver and the Windows driver.

6.1 UEFI driver

By default, 4 MIPI-DSI lanes are set.

The UEFI driver is available for i.MX 8MN, i.MX 8MM, i.MX 8MP: file `MipiDsi.c`, the `MipiDsiConfig` function, at

```
\mu_platform_nxp\Silicon\ARM\NXP\iMX8Pkg\Library\iMX8LcdHwLib\
```

MIPI-DSI lane interface:

```
EFI_STATUS
MipiDsiConfig (
  IMX_DISPLAY_TIMING* Timing,
  imxConverter MipiDsiConverter
  )
…
  MipiDsiPktRegisterCallback(&MipiDsiPktSend);
  /* Dafault 4 MIPI DSI lanes */
  lanes = 4U; // 4-lane DSI interface
  MipiDsiDisplayClockConfig(Timing)
```

6.2 Windows driver

Number of lanes (data wires) of MIPI-DSI connection.

The `Display0NumLanes` parameter is used for the first display, the `Display1NumLanes` is used for the second display, and so on. There are two options:

1. Update `galcore.inf` and uninstall/re-install the GPU driver.
   MIPI-DSI lane settings:

   ```
   [GcWddmMN_AddReg] // Find appropriate platform (MP, MN, 8X)
   …
   ;Following parameters relevant do MIPI-DSI interface
   HKR,,Display0NumLanes,%REG_DWORD%,4 // 4-lane DSI interface
   HKR,,Display0ChannelId,%REG_DWORD%,0
   ```

2. Update the Registry database on the target and restart the GPU driver:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-
e325-11cebfc1-08002be10318}\0000
```

7 Virtual mode

Physical resolution and screen position of the display stays the same.

The compositor creates virtual mode as rotation and/or resolution.

1. Rotation of the display (available for all SoC mentioned in Display support section).

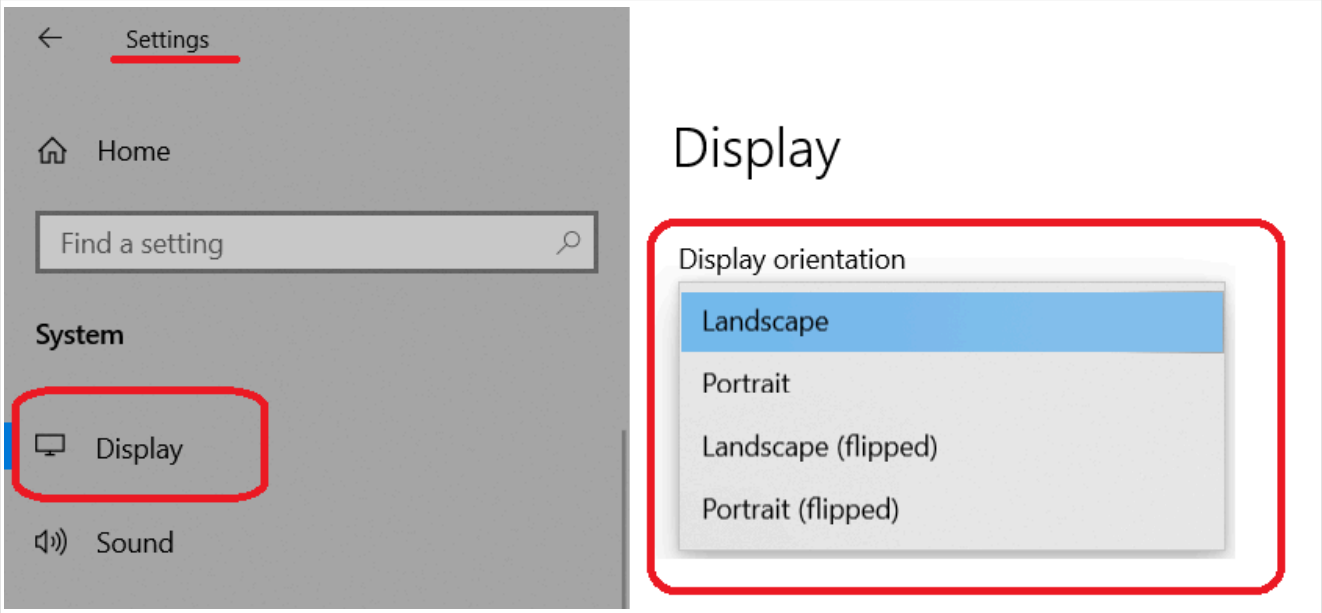

Figure 15. Display orientation setting

2. Changing screen resolution is available only for the native HDMI display interface for i.MX8MP.

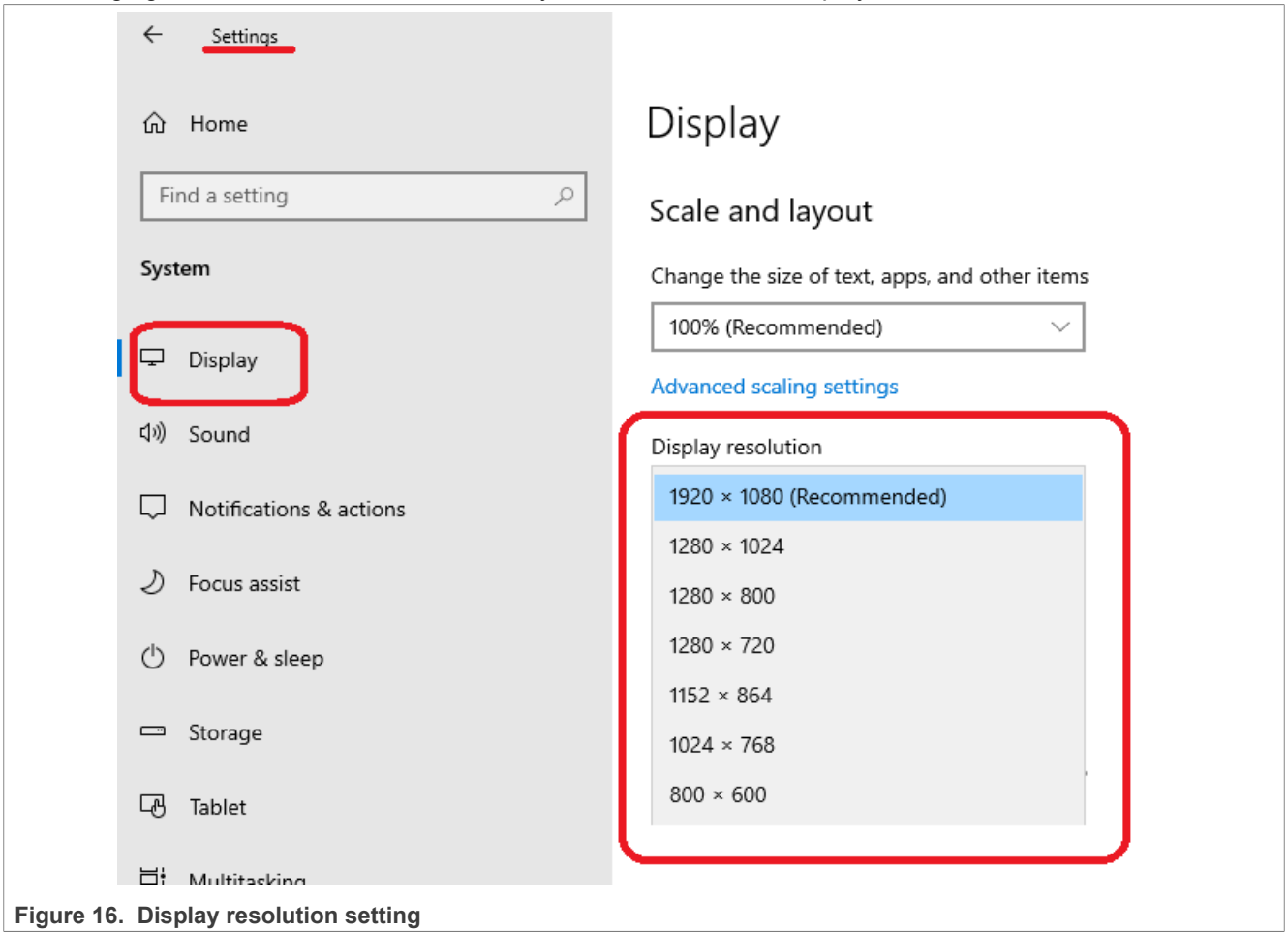

Figure 16. Display resolution setting

8 Workaround for LVDS1 on i.MX 8MP

For changing interface to LVDS1, follow the procedure below:

1. Boot Windows 10 with HDMI display (default) to see the screen. Switch to LVDS1 display in the Registry database: `Display0Interface = 0x5 (HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-e325-11cebfc1-08002be10318}\0000)`
2. Shut down Windows, power off the board. Install display and LVDS-HDMI converter to LVDS1 connector.
3. Update the firmware according to the following steps.
   a. `/mu_platform_nxp/NXP/MX8M_PLUS_EVK/MX8M_PLUS_EVK.dsc`. Update parameters:
      - `giMX8TokenSpaceGuid.PcdDisplayInterface|3`
      - `giMX8TokenSpaceGuid.PcdDisplayI2CBaseAddr|0x30A40000`

   b. `/mu_platform_nxp/NXP/MX8M_PLUS_EVK/AcpiTables/Dsdt-Gfx.asl`. Change the I2C address for the LVDS-HDMI converter:
      - `I2CSerialBus(0x4C, ControllerInitiated, 400000, AddressingMode7Bit, \\_SB.I2C3)`
      - `I2CSerialBus(0x33, ControllerInitiated, 400000, AddressingMode7Bit, \\_SB.I2C3)`
4. Compile the firmware, load it to the SD card or eMMC.
5. Boot Windows with the LVDS1 display.

9 IMX-DLVDS-LCD display on i.MX 8MP

The default interface is HDMI. To change to dual LVDS, follow the procedure below:

For dual-LVDS panel 1920x1200@60

- **Windows:** `.reg` scripts are available in Section 12.
  1. Boot Windows with the default HDMI display.
  2. Change display to dual-LVDS: `Display0Interface` = 0x6 (See example 11 in Section 12).
  3. Change display resolution to 1920x1200@60. `Display0EDID` parameter (See example 8 in Section 12).
  4. Shut down Windows, power off the board.
  5. Replace the HDMI display, connect IMX-DLVDS-LCD with two mini-SAS cables chan0=J8=LVDS1, chan1=J9=LVDS0
  6. Continue with the firmware update.
- **UEFI firmware:**
  1. Select the dual-LVDS interface in `MX8M_PLUS_EVK.dsc: giMX8TokenSpaceGuid.PcdDisplayInterface|4`
  2. Increase FB size in MX8M_PLUS_EVK.dsc, ~9MB (4*1920*1200): `gArmPlatformTokenSpaceGuid.PcdArmLcdDdrFrameBufferSize | 0x008CA000`
  3. Move subsequent memory areas in MX8M_PLUS_EVK.dsc accordingly:
     `giMXPlatformTokenSpaceGuid.PcdGlobalDataBaseAddress | 0x408CA000`
     `gOpteeClientPkgTokenSpaceGuid.PcdTpm2AcpiBufferBase | 0x408CB000`
- Set the 1920x1200@60 resolution in the `iMX8LcdHwLib.c, the LcdDisplayDetect` function.
  Dual LVDS settings for i.MX 8MP:

```
EFI_STATUS
LcdDisplayDetect (
```

```
  VOID
  )
…
/* Converter was not detected - select fixed
default timimng */
  if (converter == transmitterUnknown) {
    if (displayInterface == imxMipiDsi) {
      videoModesCnt++;
      LcdInitPreferredTiming
 (&PreferredTiming_1080x1920_60,
 &PreferredTiming);
      DEBUG((DEBUG_ERROR, "Mipi-dsi
 display interface. Default resolution used.
 %dx%d pclk=%d Hz\n",
            PreferredTiming.HActive,
 PreferredTiming.VActive,
 PreferredTiming.PixelClock));
      LcdDumpDisplayTiming(0,
 &PreferredTiming);
      return EFI_SUCCESS;
    } else if ((displayInterface == imxLvds0)
 || (displayInterface == imxLvds1) ||
 (displayInterface == imxLvds0dual)) {
      videoModesCnt++;
      LcdInitPreferredTiming
 (&PreferredTiming_1280x720_60,
 &PreferredTiming); // Set default 1920x1200@60
      DEBUG((DEBUG_ERROR, "LVDS%d
 display interface. Default resolution used.
 %dx%d pclk=%d Hz\n",
            displayInterface-2,
 PreferredTiming.HActive,
 PreferredTiming.VActive,
 PreferredTiming.PixelClock));
      LcdDumpDisplayTiming(0,
 &PreferredTiming);
      return EFI_SUCCESS;
    } else if (displayInterface ==
 imxNativeHdmi) {
      videoModesCnt++;
      LcdInitPreferredTiming
 (&PreferredTiming_1920x1080_60,
 &PreferredTiming);
      DEBUG((DEBUG_ERROR, "HDMI display
 interface. Fixed default resolution used. %
 dx%d pclk=%d Hz\n",
            PreferredTiming.HActive,
      PreferredTiming.VActive,
      PreferredTiming.PixelClock));
      LcdDumpDisplayTiming(0,
 &PreferredTiming);
      return EFI_SUCCESS;
```

- Compile the UEFI firmware, load to an SD card or eMMC, and boot Windows.
- Win->Settings->System->Display->Scale and layout: Change the size of text, apps, and other items to 200%.

10 Workaround for Multiple displays (3 displays) on i.MX 8MP

To enable 3 displays on the 8MP EVK board follow the procedure below:

1. Use the default `Galcore.inf` file (without the edits).
2. Install the OS preferably with one display connected (default HDMI).
3. Connect the second and third displays to the board.
4. Run `LVDS0_HDMI_MIPI0_multimon.reg` to enable multimonitor mode, restart the board. (See Example 15).
5. All 3 displays should be active as follows:
   Display 1 and Display 3 are duplicated, Display 2 is extended.
   1 Display - LVDS0
   2 Display - MIPI DSI
   3 Display - HDMI
6. To revert the registry settings, run `HDMI_multimon.reg`, then restart the board. (See Example 16).
7. Only one display (an HDMI display) works.

Note: *To set 3 displays independently, follow the instructions below:*

1. *Open regedit (Registry Editor) as administrator and navigate to:* `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-e325-11ce-bfc1-08002be10318}\0000`
2. *Change* ***GdiAccLevel*** *to* ***0***
3. *Reboot the board. Now, 3 displays can be displayed independently (extended).*

11 Setting display resolution with an EDID file

This example describes changing the resolution from 1920x1080 to 1080x1920 and back.

1. Connect the IMX-MIPI-HDMI converter to, for example, i.MX8MN-EVK.

2. Power on the board and boot Windows.

3. Use the `.reg` files with the appropriate resolution from Section 12 and copy them to the target.

4. Run the `1080x1920EDID.reg` script. It sets a 1080x1920 resolution to the registry.

5. Shut down Windows, power off the board.

6. Disconnect the IMX-MIPI-HDMI converter from the EVK and connect the mx8-dsi-oled1 panel instead.

7. Power on the board and boot Windows.

8. [Optional] If you want to go back to the 1920x1080 resolution to be used with the IMX-MIPI-HDMI converter, run the `1920x1080EDID.reg` script.

9. Shut-down Windows, power off the board.

12 EDID .reg scripts examples

The following script examples could be used to simplify the Windows Registry editing of specific display features. The EDID .reg scripts below can be used for MIPI and LVDS displays on i.MX 8MN, i.MX 8QXP, i.MX 8MP to set custom resolutions. Create an `.reg` file and copy the appropriate text below for custom editing.

- Example 1.`1024x768EDID.reg` file

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-e325-11ce-
bfc1-08002be10318}\0000]
"Display0EDID"=hex:\
 00, FF, FF, FF, FF, FF, FF, 00, 10, AC, 7A, A0, 53, 4B, 35, 32,\
 1E, 1A, 01, 03, 80, 34, 20, 78, EA, EE, 95, A3, 54, 4C, 99, 26,\
 0F, 50, 54, A1, 08, 00, 81, 40, 81, 80, A9, 40, B3, 00, D1, C0,\
```

```
 01, 01, 01, 01, 01, 01, 64, 19, 00, 40, 41, 00, 26, 30, 18, 88,\
 36, 00, 40, 44, 21, 00, 00, 1A, 00, 00, 00, FF, 00, 59, 50, 50,\
 59, 30, 36, 37, 56, 32, 35, 4B, 53, 0A, 00, 00, 00, FC, 00, 44,\
 45, 4C, 4C, 20, 55, 32, 34, 31, 32, 4D, 0A, 20, 00, 00, 00, FD,\
 00, 32, 3D, 1E, 53, 11, 00, 0A, 20, 20, 20, 20, 20, 20, 00, 1D
```

- Example 2. `1080x1920EDID.reg` file

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-e325-11ce-
bfc1-08002be10318}\0000]
"Display0EDID"=hex:\
 00, FF, FF, FF, FF, FF, FF, 00, 10, AC, 7A, A0, 53, 4B, 35, 32,\
 1E, 1A, 01, 03, 80, 20, 34, 78, EA, EE, 95, A3, 54, 4C, 99, 26,\
 0F, 50, 54, 00, 00, 00, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01,\
 01, 01, 01, 01, 01, 01, 30, 2A, 38, 38, 40, 80, 10, 70, 14, 02,\
 A2, 00, 44, 26, 12, 00, 00, 18, 00, 00, 00, FF, 00, 59, 50, 50,\
 59, 30, 36, 37, 56, 32, 35, 4B, 53, 0A, 00, 00, 00, FC, 00, 44,\
 45, 4C, 4C, 20, 55, 32, 34, 31, 32, 4D, 0A, 20, 00, 00, 00, FD,\
 00, 28, 3D, 1E, 78, 11, 00, 0A, 20, 20, 20, 20, 20, 20, 00, 23
```

- Example 3. `1280x1024EDID.reg` file

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-e325-11ce-
bfc1-08002be10318}\0000]
"Display0EDID"=hex:\
 00, FF, FF, FF, FF, FF, FF, 00, 10, AC, 7A, A0, 53, 4B, 35, 32,\
 1E, 1A, 01, 03, 80, 34, 20, 78, EA, EE, 95, A3, 54, 4C, 99, 26,\
 0F, 50, 54, A1, 08, 00, 81, 40, 81, 80, A9, 40, B3, 00, D1, C0,\
 01, 01, 01, 01, 01, 01, 30, 2A, 00, 98, 51, 00, 2A, 40, 30, 70,\
 13, 00, 40, 44, 21, 00, 00, 1A, 00, 00, 00, FF, 00, 59, 50, 50,\
 59, 30, 36, 37, 56, 32, 35, 4B, 53, 0A, 00, 00, 00, FC, 00, 44,\
 45, 4C, 4C, 20, 55, 32, 34, 31, 32, 4D, 0A, 20, 00, 00, 00, FD,\
 00, 32, 3D, 1E, 53, 11, 00, 0A, 20, 20, 20, 20, 20, 20, 00, E7
```

- Example 4. `1280x720EDID.reg` file

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-e325-11ce-
bfc1-08002be10318}\0000]
"Display0EDID"=hex:\
 00, FF, FF, FF, FF, FF, FF, 00, 10, AC, 7A, A0, 53, 4B, 35, 32,\
 1E, 1A, 01, 03, 80, 34, 20, 78, EA, EE, 95, A3, 54, 4C, 99, 26,\
 0F, 50, 54, A1, 08, 00, 81, 40, 81, 80, A9, 40, B3, 00, D1, C0,\
 01, 01, 01, 01, 01, 01, 01, 1D, 00, 72, 51, D0, 1E, 20, 6E, 28,\
 55, 00, 40, 44, 21, 00, 00, 1A, 00, 00, 00, FF, 00, 59, 50, 50,\
 59, 30, 36, 37, 56, 32, 35, 4B, 53, 0A, 00, 00, 00, FC, 00, 44,\
 45, 4C, 4C, 20, 55, 32, 34, 31, 32, 4D, 0A, 20, 00, 00, 00, FD,\
 00, 32, 3D, 1E, 53, 11, 00, 0A, 20, 20, 20, 20, 20, 20, 00, 6D
```

- Example 5. `1280x800EDID.reg` file

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-e325-11ce-
bfc1-08002be10318}\0000]
"Display0EDID"=hex:\
  00, ff, ff, ff, ff, ff, ff, 00, 0e, 14, 0a, 14, 00, 00, 00, 00, \
  26, 11, 01, 03, 80, 21, 15, 78, 0a, 09, 2d, 9d, 56, 4f, 90, 27, \
  21, 50, 54, 00, 00, 00, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01, \
  01, 01, 01, 01, 01, 01, ea, 1a, 00, 80, 50, 20, 10, 30, 15, 20, \
  13, 00, 4b, cf, 10, 00, 00, 19, 00, 00, 00, 0f, 00, 00, 00, 00, \
```

```
  00, 00, 00, 00, 00, 20, 6e, 05, 0f, 00, 00, 00, 00, fe, 00, 46, \
  44, 31, 36, 33, 30, 31, 35, 34, 57, 42, 34, 20, 00, 00, 00, fe, \
  00, 2d, 40, 50, 59, 7d, a9, c8, ff, 01, 01, 20, 20, 20, 00, a3
```

- Example 6. `1366x768EDID.reg` file

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-e325-11ce-
bfc1-08002be10318}\0000]
"Display0EDID"=hex:\
 00, FF, FF, FF, FF, FF, FF, 00, 10, AC, 7A, A0, 53, 4B, 35, 32,\
 1E, 1A, 01, 03, 80, 34, 20, 78, EA, EE, 95, A3, 54, 4C, 99, 26,\
 0F, 50, 54, 00, 00, 00, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01,\
 01, 01, 01, 01, 01, 01, 66, 21, 56, AA, 51, 00, 1E, 30, 46, 8F,\
 33, 00, 40, 44, 21, 00, 00, 1A, 00, 00, 00, FF, 00, 59, 50, 50,\
 59, 30, 36, 37, 56, 32, 35, 4B, 53, 0A, 00, 00, 00, FC, 00, 44,\
 45, 4C, 4C, 20, 55, 32, 34, 31, 32, 4D, 0A, 20, 00, 00, 00, FD,\
 00, 32, 3D, 1E, 53, 11, 00, 0A, 20, 20, 20, 20, 20, 20, 00, A7
```

- Example 7. `1920x1080EDID.reg` file

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-e325-11ce-
bfc1-08002be10318}\0000]
"Display0EDID"=hex:\
 00, FF, FF, FF, FF, FF, FF, 00, 10, AC, D2, D0, 4C, 58, 37, 30,\
 0B, 1C, 01, 03, 80, 35, 1E, 78, EE, 21, 95, A9, 54, 4E, 9C, 26,\
 0F, 50, 54, A5, 4B, 00, 71, 4F, 81, 80, A9, C0, D1, C0, 01, 01,\
 01, 01, 01, 01, 01, 01, 02, 3A, 80, 18, 71, 38, 2D, 40, 58, 2C,\
 45, 00, 0F, 28, 21, 00, 00, 1E, 00, 00, 00, FF, 00, 42, 47, 4D,\
 50, 44, 4D, 32, 0A, 20, 20, 20, 20, 20, 00, 00, 00, FC, 00, 44,\
 45, 4C, 4C, 20, 53, 32, 34, 31, 39, 48, 0A, 20, 00, 00, 00, FD,\
 00, 38, 4C, 1E, 53, 11, 00, 0A, 20, 20, 20, 20, 20, 20, 00, B6
```

- Example 8. `1920x1200_156_68EDID.reg` file

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-e325-11ce-
bfc1-08002be10318}\0000]
"Display0EDID"=hex:\
00, FF, FF, FF, FF, FF, FF, 00, 04, 21, 00, 00, 00, 00, 00, 00,\
01, 00, 01, 03, 80, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,\
00, 00, 00, 00, 00, 00, 01, 00, 01, 00, 01, 00, 01, 00, 01, 00,\
01, 00, 01, 00, 01, 00, 34, 3D, 80, E6, 70, B0, 0F, 40, 5A, 28,\
55, 00, 00, 00, 00, 00, 00, 18, 00, 00, 00, 10, 00, 00, 00, 00,\
00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 10, 00, 00,\
00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,\
00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, FF
```

- Example 9. `800x480EDID.reg` file

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-e325-11ce-
bfc1-08002be10318}\0000]
"Display0EDID"=hex:\
 00, FF, FF, FF, FF, FF, FF, 00, 36, 09, 01, 70, 01, 01, 01, 01,\
 22, 15, 01, 03, 80, 29, 1A, 78, EE, E5, B5, A3, 55, 49, 99, 27,\
 13, 50, 54, AC, 00, 00, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01,\
 01, 01, 01, 01, 01, 01, 8D, 0B, 20, C0, 30, E0, 11, 10, 10, 50,\
 13, 00, FF, FF, 00, 00, 00, 1C, 00, 00, 00, FC, 00, 4D, 50, 49,\
 37, 30, 30, 31, 0A, 20, 20, 20, 20, 20, 00, 00, 00, FD, 00, 32,\
 4C, 1C, 51, 0E, 00, 0A, 20, 20, 20, 20, 20, 20, 00, 00, 00, FF,\
```

```
 00, 42, 33, 34, 33, 32, 38, 34, 35, 0A, 20, 20, 20, 20, 00, 5E
```

- Example 10. `800x600EDID.reg` file

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-e325-11ce-
bfc1-08002be10318}\0000]
"Display0EDID"=hex:\
 00, FF, FF, FF, FF, FF, FF, 00, 10, AC, 7A, A0, 53, 4B, 35, 32,\
 1E, 1A, 01, 03, 80, 34, 20, 78, EA, EE, 95, A3, 54, 4C, 99, 26,\
 0F, 50, 54, A1, 08, 00, 81, 40, 81, 80, A9, 40, B3, 00, D1, C0,\
 01, 01, 01, 01, 01, 01, A0, 0F, 20, 00, 31, 58, 1C, 20, 28, 80,\
 14, 00, 40, 44, 21, 00, 00, 1A, 00, 00, 00, FF, 00, 59, 50, 50,\
 59, 30, 36, 37, 56, 32, 35, 4B, 53, 0A, 00, 00, 00, FC, 00, 44,\
 45, 4C, 4C, 20, 55, 32, 34, 31, 32, 4D, 0A, 20, 00, 00, 00, FD,\
 00, 32, 3D, 1E, 53, 11, 00, 0A, 20, 20, 20, 20, 20, 20, 00, F7
```

- Example 11. `DLVDS.reg` file

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-e325-11ce-
bfc1-08002be10318}\0000]
"Display0Interface"=dword:00000006
```

- Example 12. `HDMI.reg` file

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-e325-11ce-
bfc1-08002be10318}\0000]
"Display0Interface"=dword:00000001
```

- Example 13. `LVDS0.reg` file

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-e325-11ce-
bfc1-08002be10318}\0000]
"Display0Interface"=dword:00000004
```

- Example 14. `LVDS1.reg` file

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-e325-11ce-
bfc1-08002be10318}\0000]
"Display0Interface"=dword:00000005
```

- Example 15. `LVDS0_HDMI_MIPI0_multimon.reg` file

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-e325-11ce-
bfc1-08002be10318}\0000]
"EnableMultiMon"=dword:00000001
"Display0Interface"=dword:00000004
"Display0BusDataWidth"=dword:00000018
"Display0BusMapping"=dword:00000001
"Display2Interface"=dword:00000002
"Display2NumLanes"=dword:0000004
"Display2ChannelId"=dword:00000000
"Display1Interface"=dword:00000001
```

- Example 16. `HDMI_multimon.reg` file

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-e325-11ce-
bfc1-08002be10318}\0000]
```

```
"EnableMultiMon"=dword:00000001
"Display0Interface"=-
"Display1Interface"=dword:00000001
"Display2Interface"=-
```

13 Note about the source code in the document

Example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2024 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials must be provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

14 Revision history

Table 11. Revision history

Document ID	Release date	Description
AN14187 v.1.0	29 May 2024	Initial version

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

i.MX — is a trademark of NXP B.V.

Microsoft, Azure, and ThreadX — are trademarks of the Microsoft group of companies.

Contents