AN14611

i.MX RT700 MIPI DSI Display Application with Two Low-Power Use Cases

Rev. 2.0 — 10 November 2025 Application note

Document information

Information	Content
Keywords	AN14611, i.MX RT700, MIPI DSI, Low Power
Abstract	This application note provides a display demo application that presents two different low-power modes, describes how to enter these two modes, and evaluates their power consumption.

1 Introduction

The i.MX RT700 is a microcontroller with a rich set of peripherals, embedded security, and very low power consumption. Featuring a Vector Graphics Processing Unit (VGPU), an LCD Interface (LCDIF), and a MIPI DSI Controller (MIPI_DSI_HOST), i.MX RT700 offers an embedded solution for smart watches or other wearable display devices.

Power consumption is one of the most important criteria for performance evaluations of display devices. i.MX RT700 offers a Power Management Controller (PMC) and a Sleep Controller (SLEEPCON) to manage the power throughout the chip. Also, the MIPI DSI Controller supports all MIPI D-PHY standard low-power modes.

This application note provides a display demo application that presents two different low-power modes, describes how to enter these two modes, and evaluates their power consumption.

2 System architecture

The system is composed of a main processor i.MX RT700, a MIPI DSI panel G1120B0MIPI, an external HyperRAM for frame buffer storage, and a PMIC PCA9422 for power management.

G1120B0MIPI is a display module with a 1.2-inch circular AMOLED display with a 392 x 392 resolution, an IC RM67162 driver, and an IC FT3267 touch panel. G1120B0MIPI has a 1-lane MIPI DSI interface and internal memory so that it supports the MIPI DSI command mode.

There are rich graphics and display components on i.MX RT700, together with external memory interfaces (XSPI). In this application note, the VGPU is used for vector drawing, LCDIF, MIPI_DSI_HOST, and MIPI D-PHY are coupled together, and the MIPI DSI graphic output streams to the G1120B0MIPI panel. The XSPI2 connects i.MX RT700 and the external HyperRAM.

i.MX RT700 has nine separate power domains and multiple distinct power rails. This application note mainly focuses on VDD2_COMP (contains CPU0), VDD2_MEDIA (contains graphic modules and XSPI2), and VDD1_SENSE (contains CPU1).

This demo uses a 220 V to 5 V adapter. Its 5 V output is connected to the G1120B0MIPI board and the PCA9422 on the i.MX RT700 EVK as their power supply. The G1120B0MIPI board has onboard LDOs supporting driver ICs and an OLED panel with different power requirements. PCA9422 supports the power supply of all power domains in i.MX RT700. It integrates three Buck converters, one ultralow quiescent Buck-Boost converter, and four LDO regulators. In this demo, SW1 (BUCK1) supplies the VDD2 power rail of i.MX RT700, SW3 (BUCK3) supplies VDD1, and SW4 (BUCK-BOOST) supplies VDD1V8.

Power consumption of the following three different ports is evaluated in the subsequent sections:

1. VDD2, where CPU0, display modules, and XSPI2 are located
2. VDD1V8_MIPI supplies the MIPI D-PHY IO power
3. PANEL_VCC5V0 supplies the MIPI panel board power

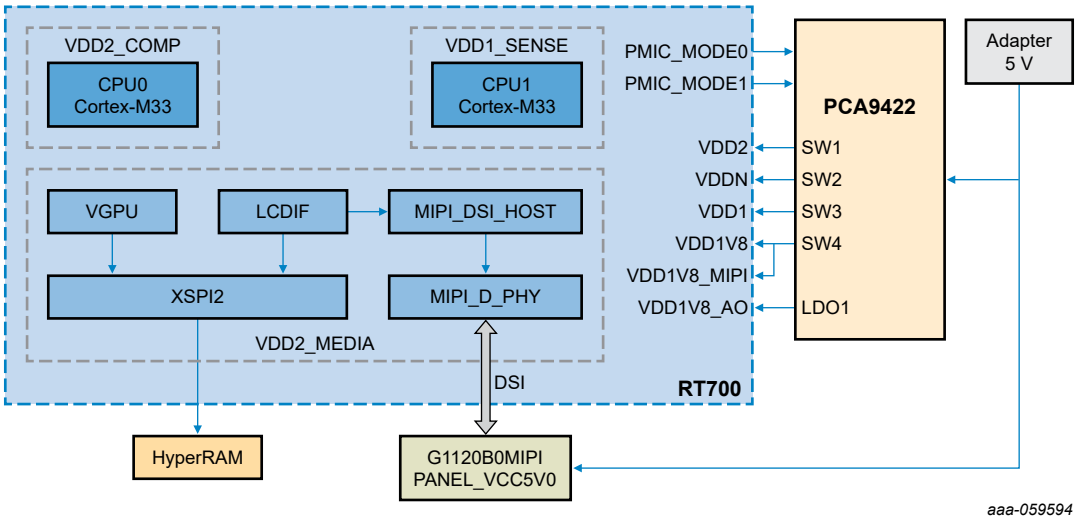

Figure 1. System architecture

3 Supported power modes

This section describes the supported power modes of i.MX RT700, PCA9422, MIPI D-PHY and panel, and the FreeRTOS system.

3.1 i.MX RT700 power modes

This section describes the i.MX RT700 power modes.

3.1.1 Power domains

A power domain is a collection of circuits that share the same voltage source and can be powered off even when their voltage source is still supplied. Table 1 lists the power domains within the chip.

Table 1. Power domain summary

Power domain	Modules
VDD2_COMP	CPU0, NPU, and Compute Domain modules
VDD2_DSP	HiFi4 DSP
VDD2_MEDIA	Media Domain modules on the VDD2 power rail
VDD2_COM	Common Domain modules on the VDD2 power rail
VDD1_SENSE	CPU1, Sense Domain (private region) modules, and Common Domain modules
VDDN_MEDIA	Media Domain modules on the VDDN power rail
VDDN_COM	Common Domain modules on the VDDN power rail
VDD1V8_AON	Always-on modules; includes RESETN, PMIC_MODE0, PMIC_MODE1, PMIC_IRQN, XTALIN32KHZ, and XTALOUT32KHZ pins
VDD1V8_PMC	PMC

3.1.2 Power supplies

The chip has separate power supplies for the digital logic: VDD1, VDD2, VDDN, VDD1V8, and VDD1V8_AO. In addition, independent power supplies are available for the I/O port pins.

Table 2. Power supply summary

Power supply	Description
VDD1	For VDD1_SENSE domain
VDD2	For VDD2_COMP, VDD2_DSP, VDD2_MEDIA, and VDD2_COM domains
VDDN	For high-speed logic in VDDN_MEDIA domain, analog interfaces, and VDDN_COM domain
VDD1V8	For 1.8-V logic
VDD1V8_AO	For always-on logic, such as power management

Both on-chip PMC and an optional external PMIC can supply power to the main power rails. In this application note, we use PMIC PCA9422 for the power supply (details are shown in Figure 2).

3.1.3 Power switches

Figure 2 shows the power switch architecture for each of the domains. The VDD1V8_AON and VDD1V8_PMC domains are not included because they are not switchable.

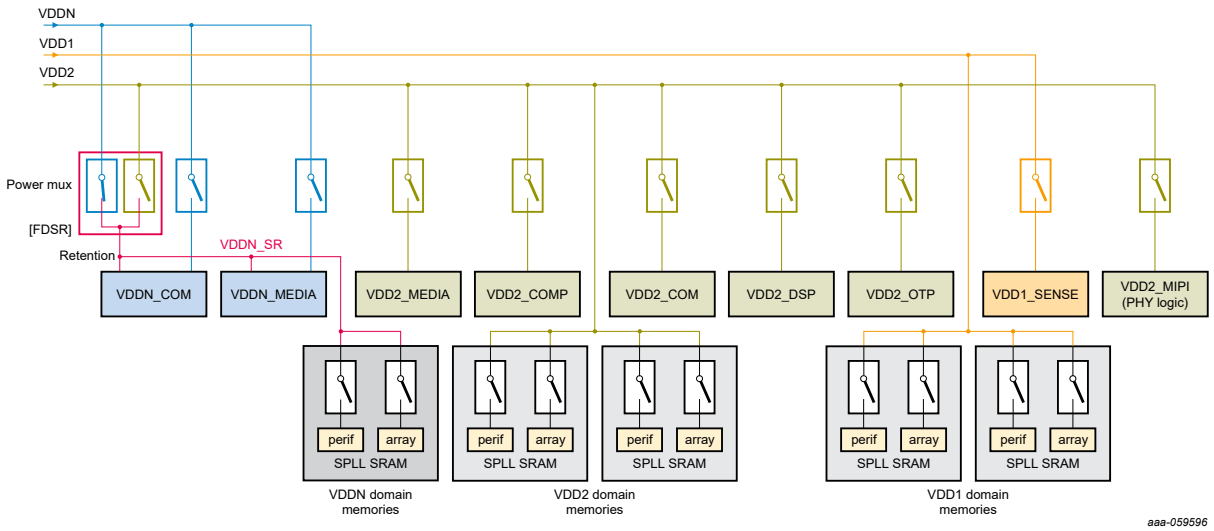

Figure 2. Power switch architecture

The VDDN domains (VDDN_COM and VDDN_MEDIA) have a power mutex to save the uSDHC/XSPI content. The power mutex switches from VDDN during the Active mode to VDD2 during the DSR mode. The power mutex supplies VDDN_SR for the retention logic in VDDN_MEDIA and VDDN_COM.

The power switch architecture establishes certain state dependencies and requirements between each of the power domains. For example, in the case of this application, when VDD2_MEDIA is off, VDDN_MEDIA must be off too.

3.1.4 Power modes overview

At any given time, power domains in i.MX RT700 can be active or placed in a lower-power state to meet the performance requirements of a system application. i.MX RT700 supports different power modes that moderate power consumption to match the performance requirements.

3.1.4.1 Active mode

The clocks to a CPU, memories, and modules are enabled. PMC.PDRUNCFGn, SLEEPCON.RUNCFG, and CLKCTL.PSCCTLn determine the power configurations.

3.1.4.2 Sleep mode

Power domains are powered, but the CPU clock is gated off via a WFI/WFE instruction. PMC.PDRUNCFGn, SLEEPCON.RUNCFG, and CLKCTL.PSCCTLn determine the power configurations. Execution of instructions is suspended until either a reset or an interrupt occurs.

3.1.4.3 Deep Sleep mode

Specific bus clocks, unused clock sources, ADC0, the main domain, and module clocks can be powered down. Unused power domains can be switched off. SLEEPCON.SLEEPCFG and PMC.PDSLEEPCFGn take effect.

3.1.4.4 Deep Sleep Retention (DSR)

Deep Sleep Retention (DSR) is a deeper low-power state than the Deep Sleep mode. The DSR mode allows significant portions of the chip to be powered down while maintaining some level of state retention, particularly the state of CPU0 and other system-critical functions.

Unlike the Deep Sleep mode, DSR does not provide the option to keep selected module clocks on for wake-up. For example, if the Compute, Media, and Common domains are placed in DSR, the modules supporting DSR would have their states retained and all modules within those domains would not operate.

3.1.4.5 Deep Power Down (DPD) mode

Only VDD1V8 and VDD1V8_AO are on. The RTC remains on. All functional pins are tristate. The SRAM content and module registers (other than those in the RTC) are not retained. On-chip regulators are off. PMC retains state. VDD1V8 monitors stay on. Voltage rail definitions are retained. TRIM is maintained, allowing for fewer voltage margins needed on wake-up.

3.1.4.6 Full Deep Power Down (FDPD) mode

Similar to the DPD mode, but only VDD1V8_AO is powered. An RTC event, an external interrupt, or an external reset can trigger a wake-up. The PMIC mode control pins remain powered to enable an optional external PMIC upon a wake-up event.

3.2 PCA9422 power modes

This section describes the PCA9422 power states, power modes, and the Dynamic Voltage Scaling (DVS) function to finely program its output voltages.

3.2.1 Power states and modes

PCA9422 supports different power states for different scenarios, like power off, storage, shipping, and running. In this application, PCA9422 always works in RUN states.

There are four kinds of power modes in a RUN state: ACTIVE, SLEEP, STANDBY, and DPSTANDBY modes, depending on STBY_MODE1 pin, SLEEP_MODE0 pin, STANDBY_CFG bit, and STANDBY_CTRL bit in the SYS_CFG1 register. In this application, only the ACTIVE and SLEEP modes are used.

Table 3. PCA9422 mode transition in RUN state

STANDBY_CTRL bit	STANDBY_CFG bit	STBY_MODE1 Pin	SLEEP_MODE0 Pin	PCA9422 MODE
X	X	Low (0)	Low (0)	ACTIVE mode
X	X	Low (0)	High (1)	SLEEP mode
0	X	High (1)	Low (0)	STANDBY mode
0	X	High (1)	High (1)	DPSTANDBY mode
1	0	High (1)	X	STANDBY mode
1	1	High (1)	X	DPSTANDBY mode
Legend: X: Does not matter. ***Note:*** *In all the above cases, VSYS > V_{SYS_UVLO} should be satisfied.*				

3.2.1.1 ACTIVE mode

PCA9422 enters the ACTIVE mode when STBY_MODE1 pin = L and SLEEP_MODE0 pin = L. In this mode, the BUCKx (x=1, 2, and 3) output voltage is set by the BUCK1OUT_DVS0 register or the DVS_CTRL[2:0] pins, depending on the Bx_DVS_CTRL[1:0] settings. The LDOy (y = 2, 3, and 4) output voltage is set by the LDOy_OUT register and the BUCK-BOOST output voltage is set by the BB_VOUT[7:0] bits.

3.2.1.2 SLEEP mode

PCA9422 enters the SLEEP mode when STBY_MODE1 pin = L and SLEEP_MODE0 pin = H. In this mode, the BUCKx (x = 1, 2, and 3) output voltage is set by the BUCKxOUT_SLEEP register, the BUCK1OUT_DVS0 register, or the DVS_CTRL[2:0] pins, depending on the Bx_DVS_CTRL[1:0] settings. The LDOy (y = 2, 3, and 4) output voltage is set by the LDOy_OUT_SLEEP register and the BUCK-BOOST output voltage is set by the BB_VOUT_SLEEP[7:0] bits.

i.MX RT700 has two dedicated pins (PMIC_MODE0 and PMIC_MODE1) for the PCA9422 mode control. PMIC_MODE0 should be connected to the SLEEP_MODE0 pin of the PCA9422, and PMIC_MODE1 to STBY_MODE1. When i.MX RT700 enters the Deep Sleep mode using APIs in the SDK, the STBY_MODE1 pin is set to L and SLEEP_MODE0 to H. The PCA9422 enters the Sleep mode and lowers the output voltages to the i.MX RT700.

3.2.2 Dynamic Voltage Scaling (DVS)

All Buck converters (BUCK1 to BUCK3) support the DVS. Their outputs are managed by the DVS_CTRL[2:0] pins. Eight programmable output voltages (BUCKxOUT_DVSx) for each Buck can be selected during the RUN and SLEEP modes. For BUCK1 and BUCK3, the DVSx outputs are programmable from 0.40 V to 1.975 V in 6.25-mV steps. For BUCK2, that is 0.4 V to 3.4 V in 25-mV steps.

3.3 MIPI D-PHY power modes

This section describes the MIPI D-PHY power modes, especially the ULPS with very low power consumption.

3.3.1 Power-consumption levels, lane states, and line levels

Each lane within a MIPI PHY configuration has potentially three different power-consumption levels: High-Speed Transmission mode, Low-Power mode, and Ultra-Low Power State. In all these modes, the line can operate in two possible transmission schemes, which are defined by the line levels used: HS differential signaling (in the HS mode) and LP mode single-ended signaling (in the LP mode and ULPS). The typical HS common level is 200 mV and the typical differential swing is 200 mV. The typical LP signaling level is 1.2 V.

This results in two possible high-speed lane states and four possible low-power lane states. The high-speed lane states are Differential-0 and Differential-1. The low-power lane states are LP-00, LP-01, LP-10, and LP-11.

Table 4. Lane state descriptions

State code	Line voltage levels		High-speed	Low-power	
	Dp-line	Dn-line	Burst mode	Control mode	Escape mode
HS-0	HS Low	HS High	Differential-0	N/A[1]	N/A[1]
HS-1	HS High	HS Low	Differential-1	N/A[1]	N/A[1]
LP-00	LP Low	LP Low	N/A	Bridge	Space
LP-01	LP Low	LP High	N/A	HS-Rqst	Mark-0
LP-10	LP High	LP Low	N/A	LP-Rqst	Mark-1
LP-11	LP High	LP High	N/A	Stop	N/A[2]

[1] During high-speed transmission the low-power receivers observe LP-00 on the lines.
[2] If LP-11 occurs during the Escape mode the lane returns to the Stop state (Control mode LP-11).

3.3.2 Data lane ULPS

MIPI D-PHY has three operating modes: Control, High-Speed, and Escape. During normal operation in HS and LP modes, a data lane will be either in the Control or High-Speed modes. The Escape mode is a special mode of operation for data lanes using low-power states. With this mode, some additional functionality becomes available, including the ULPS.

During the data lane Ultra-Low Power State, the lines are in the Space state (LP-00), that is both data lines at LP Low.

3.3.3 Clock lane ULPS

Although a clock lane does not include the regular Escape mode, the clock lane shall support the Ultra-Low Power State. During this state, the lines are in the ULP State (LP-00).

3.4 MIPI panel sleep mode

The display modules that adhere to the MIPI DSI support the Sleep mode, which can be controlled by the MIPI DCS. In this mode, all unnecessary blocks inside the display module are disabled, except for interface communication. The frame memory maintains its contents. This is the lowest-power mode that the display module supports.

3.5 FreeRTOS Tickless mode

This application is FreeRTOS-based and FreeRTOS supports a low-power state called the Tickless mode that allows the microcontroller to periodically enter and exit low power consumption. The FreeRTOS Tickless idle mode stops the periodic tick interrupt during idle periods, which allows the MCU to remain in a deep power-

saving state until either an interrupt occurs, or it is time for the RTOS kernel to transition a task into the Ready state. It then makes a correcting adjustment to the RTOS tick count value when the tick interrupt is restarted.

The FreeRTOS Tickless mode can let the MCU enter the Low-Power mode safely when it is idle. So, in this application, any switching from the Active mode to the Low-Power modes (Sleep, Deep Sleep, and Deep Sleep Retention) is handled in the Tickless idle task vPortSuppressTicksAndSleep(), which is called when there is no VGPU and LCDIF task running.

4 Demo application and performance evaluation

This section describes the demo application and performance evaluation.

4.1 Demo workflow

This demo displays a rotating polygon on the screen. After displaying it for five seconds, the screen turns off and waits for the user to press the SW5 wake-up button. The screen then displays it for the next five seconds.

Two different low-power use cases for wearable devices (like smartwatches) are demonstrated in this demo. One is the active use case with the screen displaying a polygon at 60 FPS, and the other is the standby use case with a blank display and minimum power consumption.

4.1.1 Initialization before low power

This section describes the initialization needed before entering low-power modes, including clock, PMIC, CPU1, and MIPI DSI configurations.

4.1.1.1 Clock

After powering on, the clocks of all used modules are configured to the FRO and PLL clock sources for operating in the active use case.

4.1.1.2 PMIC

The DVSs of PCA9422 are set to the appropriate voltage levels. Three different voltage levels are used in this application:

- Active mode DVS0: VDD2 = 1.1 V, VDDN = 1.1 V, and VDD1 = 1.0 V. Used when CPU0 and CPU1 are both active, peripherals on both VDD2 and VDD1 power rails are used, or registers in VDD2 or VDD1 must be accessed. It is controlled by the DVS_CTRL[2:0] pins.
- Active mode DVS1: VDD2 = 1.1 V, VDDN = 1.1 V, and VDD1 = 0.63 V. Used when CPU0 is active and CPU1 is in the Deep-Sleep mode. Controlled by the DVS_CTRL[2:0] pins.
- Sleep mode: VDD2 = 0.63 V, VDDN = 1.0 V, and VDD1 = 0.63 V. Used when CPU0 and CPU1 are both in the Deep-Sleep mode. Controlled by the PMIC_MODESEL[1:0] pins.

4.1.1.3 CPU1

Because the CPU1 and Sense domain modules are not used in this application, it boots up, enters the Deep-Sleep mode, and remains in this mode during the operation. CPU1 powers down modules as much as possible, including those that can be controlled by both the Compute and Sense domains and those controlled only by the Sense domain. Then CPU1 gives the control of all available shared modules to CPU0. In this way, CPU0 has full control of these modules.

After CPU1 enters the deep sleep, the clock and power supply for the Sense domain can also be decreased and keep this low level for the whole operation process, except for special cases when the registers in the Sense domain must be accessed.

4.1.1.4 MIPI DSI

Display modules must be initialized before starting to display. To use the DSI ULPS mode in this application, the following special configurations must be set:

Table 5. MIPI DSI configurations

DSI configurations (dsi_config_t)		
Parameter	autoInsertEoTp	enableNonContinuousHsClk
Value	true	true

Table 6. MIPI DSI D-PHY configurations

DSI D-PHY configurations (dsi_dphy_config_t)	
Parameter	tWakeup_EscClk
Value	Customized to meet the minimum TWAKEUP time (1 ms) required by the *Specification for D-PHY*.

The SDK APIs can be used to set these configurations:

MIPI DSI configurations

```
DSI_Init(DEMO_MIPI_DSI, &dsiConfig);
DSI_InitDphy(DEMO_MIPI_DSI, &dphyConfig, 0);
```

4.1.2 Use case 1: active

When the VGPU is drawing a new frame or when the LCDIF is refreshing a new frame to the panel, CPU0 and all peripherals are active.

During the interval between two frames, CPU0 enters the Deep-Sleep mode, the VDD2_MEDIA domain remains powered on, D-PHY enters the ULPS mode, and other modules in the i.MX RT700 are running at as low power as possible. The panel keeps displaying the last frame that was sent to it.

Note: *Letting D-PHY enter ULPS means that it is still powered on, so that the VDD2_MEDIA domain cannot be in the DSR mode.*

This application uses the Tearing Effect (TE) signal to synchronize the i.MX RT700 and the panel. The DSI command mode panel has its own timing controller for the display refresh. When the panel is ready for a new frame, it sends a rising edge, and CPU0 starts the data streaming. When it finishes sending one frame, it enters this lower-power mode and wait for the next TE signal to wake it up for drawing and refreshing a new frame.

4.1.3 Use case 2: standby

When the device has not been operated for a relatively long time, it can enter the Standby mode to save power. In this use case, CPU0 enters the Deep-Sleep mode with MIPI D-PHY powered off and VDD2_MEDIA in the DSR. According to power switch dependencies, VDDN_MEDIA should be in the DSR too. The panel display is in the Sleep mode (blank). This decreases the power consumption as much as possible.

In this application, after displaying for five seconds, an RTC interrupt lets the system enter the Standby mode. Users can press a button to generate a pin interrupt to wake up CPU0, and then the system is active for another five seconds.

Note: *Because CPU1 is not used in this application, we keep it in the Deep-Sleep mode in both use cases.*

Table 7. Demo power modes

Use case	Status	CPU0	Media domain	D-PHY	Panel	Power consumption
1: active	Drawing or refreshing	Active	Power on	Active	Active	High
	Frame interval	Deep sleep	Power on	ULPS	Active	Low
2: standby	Standby	Deep sleep	DSR	Power down	Sleep	Very low

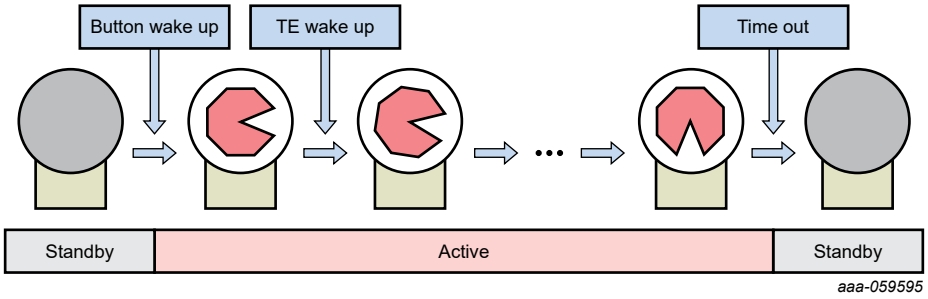

Figure 3. Demo workflow

4.2 Hardware settings

To measure the power consumption of VDD2, VDD1V8_MIPI, and PANEL_VCC5V0, some hardware settings must be done.

The i.MX RT700 EVK board has the JP1 measuring socket for VDD2. JP1 is shorted on the board by default. To measure the current of VDD2, JP1 should be connected to an ammeter (we used the Nordic Semiconductor Power Profiler Kit II (PPK2)) in series. For VDD1V8_MIPI, there is no socket on the i.MX RT700 EVK board. R311 must be removed and then connected to PPK2. For PANEL_VCC5V0, J2 on the G1120B0MIPI panel board should be connected to PPK2.

4.3 Software implementation

This section explains how to write code to achieve low-power use cases described in this application note.

4.3.1 Boot clock configurations

First, FRO0 and FRO1 should be configured, ungated, and enabled. The auto trim configurations for FRO0 are shown in Table 8.

Table 8. FRO0 configurations

FRO0 configurations (clock_fro_config_t)							
Parameter	targetFreq	refDiv	trim1DelayUs	trim2DelayUs	range	enableInt	coarseTrimEn
Value	300000000U	1U	15U	15U	50U	0U	true

FRO0 pfd0 to pfd3 and FRO1 pfd1 to pfd3 must all be enabled and powered on. They are used as a clock source instead of the PLLs to save power when entering low-power modes. The SLEEPCON0 RUNCFG register should be configured that no gating and powering down is applied to FRO0 and FRO1.

The PLLs must be initialized. For this application note, the PLL configurations shown in Table 9 and Table 10 are used.

Table 9. Main PLL configurations

Main PLL configurations (clock_main_pll_config_t)				
Parameter	main_pll_src	numerator	denominator	main_pll_mult
Value	kCLOCK_MainPllFro1Div8	0U	1U	kCLOCK_MainPllMult22

Table 10. Audio PLL configurations

Audio PLL configurations (clock_audio_pll_config_t)					
Parameter	audio_pll_src	numerator	denominator	audio_pll_mult	enableVcoOut
Value	kCLOCK_MainPllFro1Div8	0U	65536U	kCLOCK_AudioPllMult22	true

After initializing FROs and PLLs, clocks can be configured according to Table 11 and Table 12. If XSPI0 or XSPI1 are not used, their clock can be selected to NONE to save power. These clock settings are used when CPU0 is in the Active mode.

Table 11. Generated clock source configurations for CPU0 in Active mode

Generated clock sources			
Clock controller	**Clock source**	**Clock divider**	**Generated clock source**
CLKCTL0	fro1_max	---	baseclk_cmpt
CLKCTL2	fro1_max	---	baseclk_comn
CLKCTL4	fro0_div3	---	baseclk_md2

Table 12. Clock root configurations for CPU0 in Active mode

Clock roots			
Clock controller	**Clock source**	**Clock divider**	**Clock root**
CLKCTL0	main_pll_pfd0	2	COMPUTE_MAIN_CLK
CLKCTL0	main_pll_pfd0	2	COMMON_RAM_CLK
CLKCTL0	main_pll_pfd1	2	XSPI0_FCLK
CLKCTL0	baseclk_comn	1	XSPI1_FCLK
CLKCTL2	main_pll_pfd3	2	COMMON_VDDN_CLK
CLKCTL3	baseclk_sense	2	SENSE_MAIN_CLK
CLKCTL3	fro1_max	1	SENSE_RAM_CLK
CLKCTL4	main_pll_pfd0	2	MEDIA_MAIN_CLK
CLKCTL4	main_pll_pfd0	2	MEDIA_VDDN_CLK
CLKCTL4	main_pll_pfd3	1	XSPI2_FCLK
CLKCTL4	audio_pll_pfd2	1	DPHY_BIT_CLK

Table 12. Clock root configurations for CPU0 in Active mode...*continued*

Clock roots			
Clock controller	**Clock source**	**Clock divider**	**Clock root**
CLKCTL4	audio_pll_pfd2	5	DPHY_RX_CLK
CLKCTL4	DPHY_RX_CLK	4	DPHY_TX_CLK

4.3.2 Power configurations

This section describes the power configurations.

4.3.2.1 Step 1: configure PCA9422 DVS control registers

In this application, the DVS_CTRL[2:0] pins in the ACTIVE mode and BUCKxOUT_SLEEP in the SLEEP mode control the output voltages of SW1 to SW3.

PCA9422 DVS configuration

```
pca9422RegConfig.buck[0].dvsCtrl =
  (uint8_t)kPCA9422_PinInActiveAndBxOUTSLEEPInSleep;
pca9422RegConfig.buck[1].dvsCtrl =
  (uint8_t)kPCA9422_PinInActiveAndBxOUTSLEEPInSleep;
pca9422RegConfig.buck[2].dvsCtrl =
  (uint8_t)kPCA9422_PinInActiveAndBxOUTSLEEPInSleep;
PCA9422_InitRegulator(&pca9422Handle, &pca9422RegConfig);
```

4.3.2.2 Step 2: configure voltage level for each mode

Three voltage settings are used for different purposes.

Table 13. PCA9422 power mode configurations

PCA9422	VDD1	VDD2	VDDN	Used when
Active mode DVS0	1.0 V	1.1 V	1.1 V	CPU0 and CPU1 are both active/registers in VDD2 and VDD1 must be accessed.
Active mode DVS1	0.63 V	1.1 V	1.1 V	CPU0 is active and CPU1 is in the Deep-Sleep mode.
Sleep mode	0.63 V	0.63 V	1.0 V	CPU0 and CPU1 are both in the Deep-Sleep mode.

The following API can be used to write these configurations to PCA9422:

PCA9422 power mode configuration

```
PCA9422_WritePowerModeConfigs(&pca9422Handle, pca9422CurrMode,
 pca9422CurrModeCfg);
```

4.3.2.3 Step 3: CPU1 boot and configuration

After booting up, CPU1 enters the Deep-Sleep mode with minimum power consumption.

First, enable RBB for all possible domains:

Enabling RBB of all domains

```
POWER_EnableSleepRBB(kPower_BodyBiasVdd1 | kPower_BodyBiasVdd2 |
 kPower_BodyBiasVddn | kPower_BodyBiasVdd1Sram | kPower_BodyBiasVdd2Sram);
```

Change to a lower frequency to safely decrease the VDD1 voltage when CPU1 enters the low-power mode with CPU0 active and sense the shared main clock:

CPU1 clock configuration before deep sleep

```
CLOCK_AttachClk(kLPOSC_to_SENSE_BASE);
```

Enter the deep sleep with all modules powered down (for exclude_from_pd, no field is excluded):

Entering deep sleep

```
POWER_EnterDeepSleep(exclude_from_pd);
```

Note: *For the detail code example of the CPU1 boot, see the SDK demo application power_mode_comp_only.*

4.3.3 Low-Power mode for use case 1

This section describes the Low-Power mode for use case 1.

4.3.3.1 Step 1: enter D-PHY ULPS

i.MX RT700 supports the MIPI D-PHY ULPS mode. The D-PHY ULPS mode can be controlled by the ULPS_ENABLE register. The ULPS_ACTIVE register shows the current status of the ULPS. The SDK provides APIs for entering and exiting ULPS.

MIPI D-PHY entering ULPS

```
DSI_SetUlpsStatus(DEMO_MIPI_DSI, (kDSI_DataLane0UlpsEnable |
 kDSI_ClockLaneUlpsEnable));
while(DEMO_MIPI_DSI->ULPS_ACTIVE != (kDSI_DataLane0UlpsEnable |
 kDSI_ClockLaneUlpsEnable))
{
}
```

4.3.3.2 Step 2: disable PLLs

To disable PLLs, SLEEPCON0 and SLEEPCON1 must be accessed. The VDD1 voltage and the SENSE_MAIN_CLK frequency must be increased before disabling PLLs:

Power and clock configurations before disabling PLLs

```
BOARD_SetPmicDVSPinStatus(kPCA9422_ActiveModeDVS0);
CLOCK_AttachClk(kFRO1_DIV1_to_SENSE_MAIN);
```

To decrease the power consumption during deep sleep, disable the main PLL and audio PLL before entering deep sleep, and clocks that use PLLs as their clock source should be moved to FROs, as shown in Table 14 and Table 15.

Table 14. Generated clock source configurations for CPU0 Deep-Sleep mode

Generated clock sources			
Clock controller	**Clock source**	**Clock divider**	**Generated clock source**
CLKCTL0	fro1_max	---	baseclk_cmpt
CLKCTL2	fro1_max	---	baseclk_comn
CLKCTL4	fro0_div3	---	baseclk_md2

Table 15. Clock root configurations for CPU0 in Deep-Sleep mode

Clock roots			
Clock controller	**Clock source**	**Clock divider**	**Clock root**
CLKCTL0	baseclk_cmpt	1	COMPUTE_MAIN_CLK
CLKCTL0	fro1_max	1	COMMON_RAM_CLK
CLKCTL0	baseclk_comn	1	XSPI0_FCLK
CLKCTL0	baseclk_comn	1	XSPI1_FCLK
CLKCTL2	baseclk_comn	1	COMMON_VDDN_CLK
CLKCTL3	fro1_max	2	SENSE_MAIN_CLK
CLKCTL3	fro1_max	1	SENSE_RAM_CLK
CLKCTL4	fro0_max	1	MEDIA_MAIN_CLK
CLKCTL4	fro0_max	1	MEDIA_VDDN_CLK
CLKCTL4	fro0_max	1	XSPI2_FCLK
CLKCTL4	baseclk_md2	1	DPHY_BIT_CLK
CLKCTL4	baseclk_md2	5	DPHY_RX_CLK
CLKCTL4	DPHY_RX_CLK	4	DPHY_TX_CLK

4.3.3.3 Step 3: enable RBB

Enable RBB for all possible domains to decrease power consumption (see enabling RBB of all domains in Section 4.3.4.3).

4.3.3.4 Step 4: enter deep sleep

The power status of i.MX RT700 during deep sleep is configured by registers PDSLEEPCFG0 to PDSLEEPCFG5. In this scenario, we use these configurations, which keep all RAM array, MIPI_DSI_HOST, VDD2_MEDIA, VDDN_MEDIA, and VDDN_COM powered on during deep sleep. PCA9422 enters the Sleep mode (VDD2 = 0.63 V, VDDN = 1.0 V, and VDD1 = 0.63 V) after calling POWER_EnterDeepSleep().

In this use case, most modules are powered down after entering the Deep-Sleep mode. The modules excluded from power down are shown in Table 16.

Table 16. Modules excluded from power down for CPU0 in use case 1

Register	Field
PDSLEEPCFG0	V2MIPI_PD
	VNCOM_DSR

Table 16. Modules excluded from power down for CPU0 in use case 1*...continued*

Register	Field
	V2NMED_DSR
PDSLEEPCFG2	$SRAM_{29-0}$
PDSLEEPCFG4	CPU0_SCACHE
	CPU0_CCACHE
	XSPI0[1]

Legend:
1. Only necessary when the XIP method is used.

For code example, see entering deep sleep in Section 4.3.2.3.

4.3.3.5 Step 5: wake up

The TE pin interrupt is enabled as a wake-up source to wake up CPU0 from deep sleep.

4.3.3.6 Step 6: restore PLLs

After waking up from deep sleep, the previous PLL and clock configurations (shown in Table 11 and Table 12) should be restored. The VDD1 voltage and the SENSE_MAIN_CLK frequency can be decreased to save power:

Power and clock configurations before restoring PLLs

```
CLOCK_AttachClk(kSENSE_BASE_to_SENSE_MAIN);
BOARD_SetPmicDVSPinStatus(kPCA9422_ActiveModeDVS1);
```

4.3.3.7 Step 7: exit ULPS

MIPI D-PHY exiting ULPS

```
DSI_SetUlpsStatus(DEMO_MIPI_DSI, 0);
while(DEMO_MIPI_DSI->ULPS_ACTIVE != 0)
{
}
```

4.3.4 Low-Power mode for use case 2

This section describes the Low-Power mode for use case 2.

4.3.4.1 Step 1: display entering Sleep mode

RM67162 supports the `enter_sleep_mode` DCS command (0x10) to enter the Sleep mode.

Display entering Sleep mode

```
dc_fb_dbi_handle_t *dcDbiHandle = (dc_fb_dbi_handle_t *)(g_dc.prvData);
DEMO_DBI_IFACE_WriteCmdData(&(dcDbiHandle->dbiIface), kMIPI_DBI_EnterSleepMode,
 NULL, 0);
```

4.3.4.2 Step 2: power off D-PHY

D-PHY should be powered off before VDD2_MEDIA enters the DSR.

Powering off MIPI D-PHY

```
DEMO_MIPI_DSI->PD_DPHY = 1U;
```

4.3.4.3 Step 3: disable PLLs

See Section 4.3.3.2.

4.3.4.4 Step 4: enable RBB

See Section 4.3.3.3.

4.3.4.5 Step 5: enter deep sleep

Comparing to use case 1, the MIPI PHY is powered down and the VDDN_COM, VDD2_MEDIA, and VDDN_MEDIA domains are in the DSR mode in this use case. After entering deep sleep, VDD2 = 0.63 V, VDDN = 1.0 V, and VDD1 = 0.63 V. The following modules are excluded from powering down:

Table 17. Modules excluded from power down for CPU0 in use case 2

Register	Field
PDSLEEPCFG2	$SRAM_{29-0}$
PDSLEEPCFG4	CPU0_SCACHE
	CPU0_CCACHE
	XSPI0[1]

Legend:
1. Only necessary when the XIP method is used.

For code example, see entering deep sleep in Section 4.3.2.3.

4.3.4.6 Step 6: wake up

The SW5 button is used as the wake-up source.

4.3.4.7 Step 7: GPU reinitialization

VGPU must be reset to its previous state after CPU0 wakes up and VDD2_MEDIA exits the DSR mode. The vg_lite_powerup() function, based on Vivante VGLite Graphics APIs, is called to reset the VGPU.

Reinitialization GPU

```
vg_lite_error_t vg_lite_powerup(void)
{
    vg_lite_error_t error;
    uint32_t i;
    VG_LITE_RETURN_ERROR(vg_lite_kernel(VG_LITE_RESET, NULL));
    VG_LITE_RETURN_ERROR(program_tessellation(&s_context));
    for (i = 0; i < 8; i++)
{
        VG_LITE_RETURN_ERROR(push_state(&s_context, 0x0A90 + i, 0));
```

```
    }
    return VG_LITE_SUCCESS;
}
```

4.3.4.8 Step 8: restore PLLs

See Section 4.3.3.6.

4.3.4.9 Step 9: power on D-PHY

Powering on MIPI D-PHY

```
DEMO_MIPI_DSI->PD_DPHY = 0U;
```

4.3.4.10 Step 10: display exit Sleep mode

Use the `exit_sleep_mode` DCS command (0x11) to exit the Sleep mode.

Display exiting Sleep mode

```
dc_fb_dbi_handle_t *dcDbiHandle = (dc_fb_dbi_handle_t *)(g_dc.prvData);
DEMO_DBI_IFACE_WriteCmdData(&(dcDbiHandle->dbiIface), kMIPI_DBI_ExitSleepMode,
 NULL, 0);
```

4.4 Performance evaluation

This section describes the performance evaluation.

4.4.1 Demo operation timing analysis

Figure 4 was captured using Saleae Logic Analyzer, demonstrating how the VGPU and LCDIF refresh the panel and the entering and exiting timing of two different low-power modes.

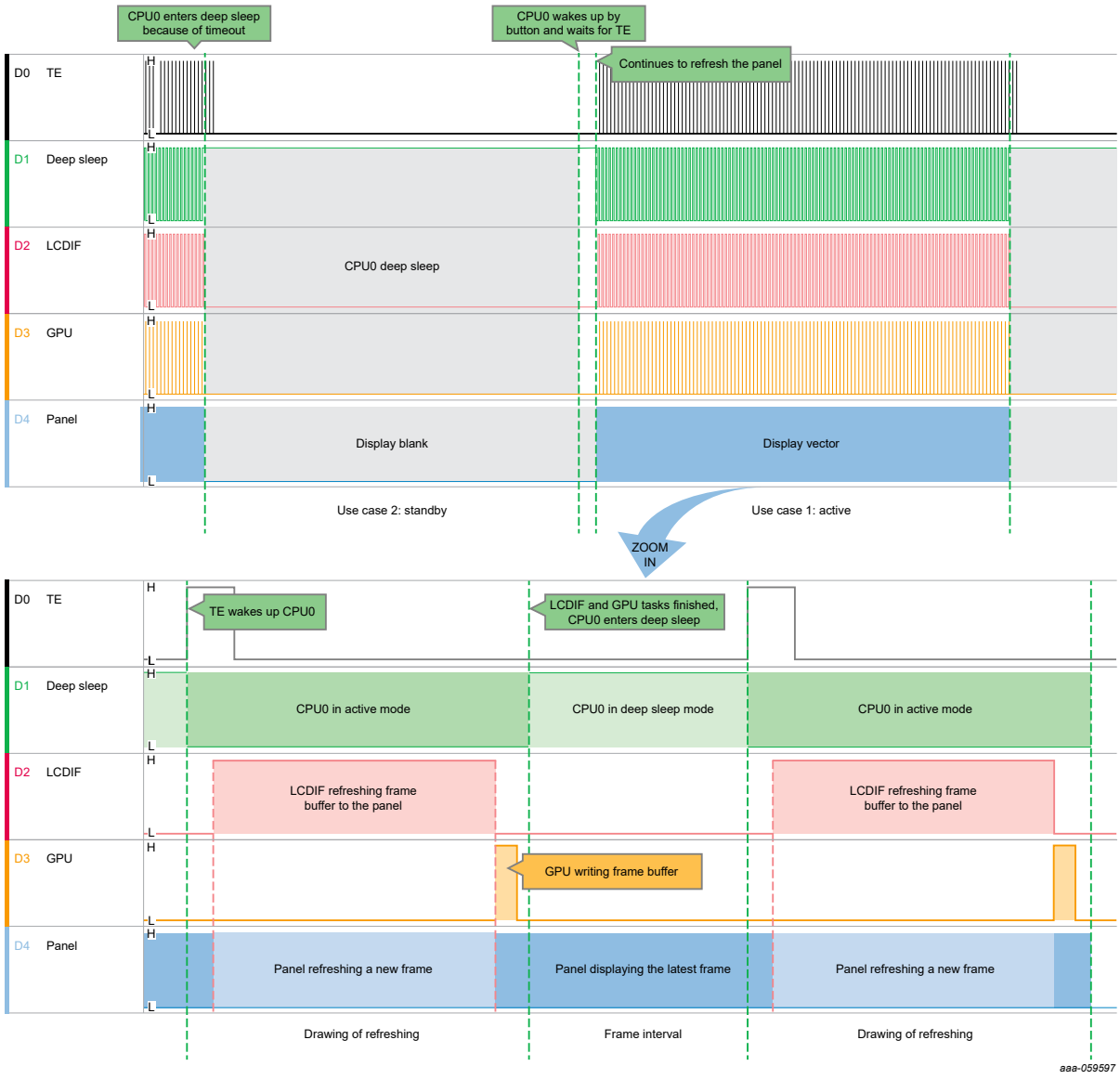

Figure 4. Demo operation timing analysis

4.4.2 Demo power consumption

The currents of VDD2, VDD1V8_MIPI, and PANEL_VCC5V0 were measured using PPK2. Table 18 shows the power consumption for this demo.

Table 18. Current measurement

Port	Use case	Status	Voltage	Current	Power
VDD2	1	Drawing or refreshing	1.1 V	91.34 mA	100.47 mW
		Frame interval	0.63 V	154.92 µA	97.60 µW
	2	Standby	0.63 V	52.59 µA	33.13 µW
VDD1V8_MIPI	1	Drawing or refreshing	1.8 V	771.41 µA	1.39 mW
		Frame interval	1.8 V	603.58 µA	1.09 mW

Table 18. Current measurement*...continued*

Port	Use case	Status	Voltage	Current	Power
	2	Standby	1.8 V	0.20 µA	0.36 µW
PANEL_VCC5V0	1	Drawing or refreshing	5.0 V	53.81 mA	269.05 mW
		Frame interval	5.0 V	47.89 mA	239.45 mW
	2	Standby	5.0 V	2.75 mA	13.75 mW

As the result:

- CPU0 entering the Deep-Sleep mode can significantly reduce the power consumption of the i.MX RT700 VDD2 (from 100.47 mW to 97.60 µW), and VDD2_MEDIA entering the DSR can further decrease this number to 33.13 µW.
- For the MIPI D-PHY, entering ULPS can reduce its power consumption (from 1.39 mW to 1.09 mW), and powering it down can lower this value to nearly zero (0.36 µW).
- The MIPI DSI command mode panel consumes slightly less power when it keeps displaying the last frame (269.05 mW) than when refreshing a new frame (239.45 mW), and turning it off can significantly reduce its power consumption to 13.75 mW.

4.4.3 Power consumption estimation for a smart watch application

For a smartwatch application using i.MX RT700 and a panel similar to G1120B0MIPI, we can have the following assumptions:

- It works at 30 FPS, the average VGPU drawing time is 15 ms, and the average LCDIF refreshing time is 8 ms.

Note: *These data are estimates based on the data measured from this demo.*

- It works under the active scenario (screen display on) one hour per day and in standby (screen display off) for the other 23 hours.
- It has a 1.2-Wh battery.

The average power of the i.MX RT700 display function in active scenario is:

$$(100.47\,mW + 1.39\,mW) \times \frac{(15\,ms + 8\,ms)}{33\,ms} + (97.60\,\mu W + 1.09\,mW) \times \left[1 - \frac{(15\,ms + 8\,ms)}{33\,ms}\right] = 71.35\,mW \quad (1)$$

The average power of the i.MX RT700 in the standby scenario is:

$$33.13\,\mu W + 0.36\,\mu W = 33.49\,\mu W \quad (2)$$

The average power of i.MX RT700 in one day is:

$$71.35\,mW \times \frac{1\,h}{24\,h} + 33.49\,\mu W \times \left(1 - \frac{1\,h}{24\,h}\right) = 3.01\,mW \quad (3)$$

For the panel, the average power in the active scenario is:

$$269.05\,mW \times \frac{(15\,ms + 8\,ms)}{33\,ms} + 239.45\,mW \times \left[1 - \frac{(15\,ms + 8\,ms)}{33\,ms}\right] = 260.08\,mW \quad (4)$$

While the standby average power is 13.75 mW, the average power of the panel in one day is:

$$260.08\,mW \times \frac{1\,h}{24\,h} + 13.75\,mW \times \left(1 - \frac{1\,h}{24\,h}\right) = 24.01\,mW \quad (5)$$

When considering the display as the only power-consuming function of this smart watch, then it can display continuously for:

$$1.2\,Wh \div (71.35\,mW + 260.08\,mW) = 3.6\,h \quad (6)$$

It can be in standby at the minimum power consumption for:

$$1.2\ Wh \div (33.49\ \mu W + 13.75\ mW) = 87\ h \quad (7)$$

For a normal daily use of one hour of display per day, the battery life is:

$$1.2\ Wh \div (3.01\ \mu W + 24.01\ mW) \div 24\ h = 1.9\ day \quad (8)$$

5 Acronyms and abbreviations

Table 19. Acronyms and abbreviations

Term	Meaning
ADC	Analog-to-Digital Converter
AMOLED	Active-Matrix Organic Light-Emitting Diode
API	Application Programming Interface
CPU	Central Processing Unit
DC	Direct Current
DCS	Display Command Set
DPD	Deep Power Down
DSI	Display Serial Interface
DSR	Deep Sleep Retention
DVS	Dynamic Voltage Scaling
EVK	Evaluation Kit
FDPD	Full Deep Power Down
FPS	Frames Per Second
FRO	Free-Running Oscillator
HS	High-Speed
IC	Integrated Circuit
LCD	Liquid Crystal Display
LCDIF	LCD Interface
LDO	Low Dropout Regulator
OLED	Organic Light-Emitting Diode
LP	Low Power
MCU	Microcontroller Unit
MIPI	Mobile Industry Processor Interface
PHY	Physical Layer
PLL	Phase-Locked Loop
PMC	Power Management Controller
PMIC	Power Management Integrated Circuit
PPK2	Power Profiler Kit II
HyperRAM	Pseudo-Static Random Access Memory
RAM	Random Access Memory

Table 19. Acronyms and abbreviations...*continued*

Term	Meaning
RTC	Real-Time Clock
RTOS	Real-Time Operating System
SDK	Software Development Kit
SLEEPCON	Sleep Controller
TE	Tearing Effect
ULPS	Ultra-Low Power State
VGPU	Vector Graphics Processing Unit
WFE	Wait for Event
WFI	Wait for Interrupt
XIP	Execute in Place
XSPI	External Serial Peripheral Interface

6 Note about the source code in the document

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

Copyright 2025 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.

7 Revision history

Table 20 summarizes the revisions done to this document.

Table 20. Revision history

Document ID	Release date	Description
AN14611 v.2.0	10 November 2025	Initial public release
AN14611 v.1.0	11 March 2025	Initial internal release

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.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

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.

Contents