

AN14791

Automatic Voltage Scaling using Process, Voltage, and Temperature Sensor (PVTs) on i.MX RT700

Rev. 3.0 — 10 November 2025

Application note

Document information

Information	Content
Keywords	AN14791, i.MX RT700, Dynamic Voltage Scaling (DVS)
Abstract	This document describes how the PVTs sensor works and how to use it in an application to extend battery life.



1 Introduction

The i.MX RT700 is an excellent, highly integrated MCU that is ideal for many low-power applications, such as wearables. The i.MX RT700 is architected for sustained low-power operation while supporting advanced peripherals and secure applications. It has 5 independent chip domains (Compute, Sense, Common, DSP, Media), 2 of which serve as the primary processing domains - the Compute and Sense Domains are powered by the digital supply rails VDD2 and VDD1 (respectively).

This MCU has many features to reduce and optimize power consumption to extend battery life. The RT700 also supports Dynamic Voltage Scaling (DVS) to optimize the active power with the operating frequencies. This document covers an integrated feature in the i.MX RT700. When used properly, it can potentially reduce the power further and lower the voltage below the minimum specifications in the data sheet. This feature monitors the Process, Voltage, and Temperature (PVT) variation of each unique device at runtime. The i.MX RT700 has 2 PVTS, 1 in the Compute domain PVTS0 and 1 in the Sense domain PVTS1. Using this PVTS sensor reduces the power on average by 20 % (on the VDD2 or VDD1 supply rails) for the digital logic.

Depending on the domain, the digital logic rail is called VDD2 for the Compute domain, to which the CPU0 and HIFI4 belong. For the Sense domain, to which the CPU1 and HIFI1 belong, the digital logic rail is called VDD1. The PVTS sensor reduces the voltage and power in active modes, including the i.MX RT700 sleep mode. The PVTS sensor does not improve the power consumption in the static low-power modes and it is not used in those modes.

This document describes how the PVTS sensor works and how to use it in an application to extend battery life.

- On the i.MX RT700, the PVTS0 sensor in the Compute domain can be used when the `compute_main_clk` and/or `dsp_clk` clocks are clocked at 325 MHz, 192 MHz, or 110 MHz.
- The PVTS1 sensor in the Sense domain can be used when the `sense_main_clk` and/or `sense_dsp_clk` clocks are clocked at 250 MHz, 160 MHz, or 100 MHz.

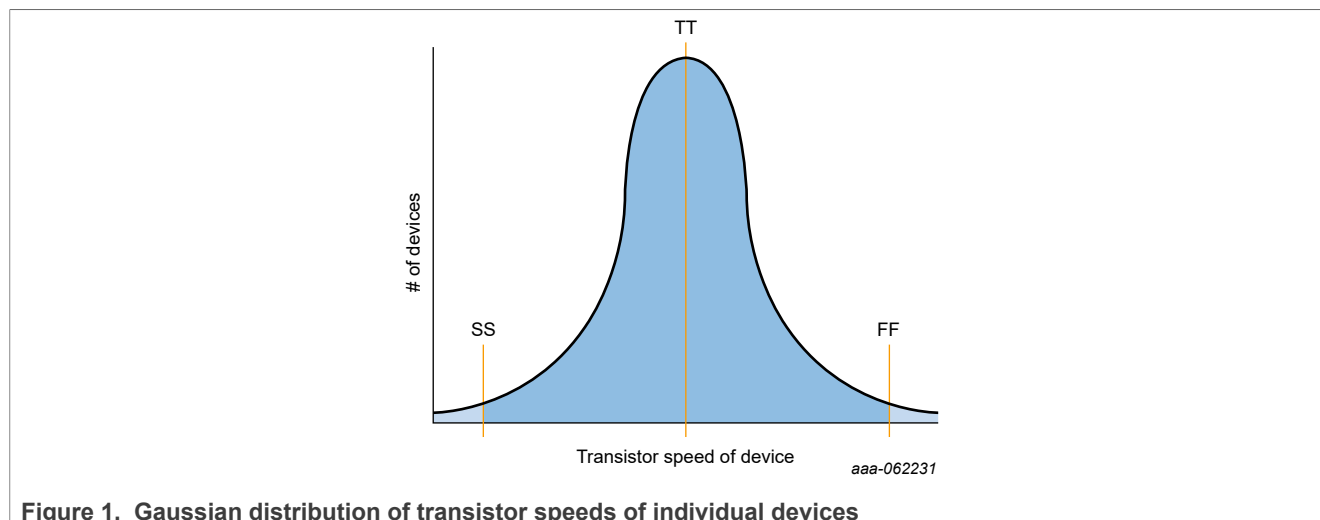
The i.MX RT700 SDK demonstrates how to use the PVTS on the Compute and Sense domains using the `mimxrt700evk_dvs_pvt_comp_only_primary` and `mimxrt700evk_dvs_pvt_comp_only_secondary` demos.

2 Process, Voltage, and Temperature (PVT) variation

The power consumption and performance of a semiconductor can change during its operation when the temperature or voltage vary. There are differences in comparing 1 specific device to another due to the variation in the manufacturing process. To optimize power consumption and performance of a device like i.MX RT700, it is important to understand how the silicon is affected by these variations.

2.1 Silicon process variation

Modern semiconductors are of such small geometries today that subtle differences in the manufacturing process can lead to noticeable differences in the key properties of each device. Even on the same wafer, each die can have different properties. One property that varies due to the process is the speed of the transistors. In this document, the variation in the speed of each device's transistors is 1-dimensional. The speed of both NFETs and PFETs on a specific device can be at the Slow-Slow (SS) limit, at the Fast-Fast (FF) limit, or anywhere in between. When we plot the distribution of this speed for every device manufactured, we have a Gaussian distribution; see [Figure 1](#). The median of these speeds is Typical-Typical (TT). The testing procedure has limits for what speeds are acceptable for production. The outliers that are slower than the SS limit or faster than the FF limit are discarded.



There are tradeoffs when comparing devices with different speeds depending on the process variation. The SS devices have a lower static leakage current, but the FF devices have faster performance when active. At the same clock frequency, the FF devices can be powered with a lower voltage than the SS devices and still achieve the same performance. Since the SS transistors are slower, they must be turned on harder using a higher supply voltage to keep up with the FF devices.

2.2 Temperature variation

Silicon performance and power consumption are also affected by temperature. i.MX RT700 has the operation range of ambient temperature from -30 °C to 85 °C, with the typical temperature of 30 °C. When comparing the performance of a specific device, if the temperature is reduced to -30 °C, the transistors are slower when compared to 30 °C. To achieve the same clocking frequency at 30 °C and -30 °C, the supply voltage must be increased to drive those transistors harder. When the device warms up back to 30 °C, the voltage can be reduced while achieving the same clock frequency.

2.3 Minimum voltage required

The i.MX RT700 can power VDD2, VDD1, and the digital power supply of the Compute and Sense domains using the internal LDOs or an external PMIC, such as NXP PCA9422, on the i.MX RT700 EVK board. This application note covers only the case when the VDD2 and VDD1 rails are powered by the external PMIC. However, it can be applied when the LDOs are used to power VDD2 and VDD1 as well.

To minimize power consumption in an application that must be clocked at a specific frequency, the supply voltage is set at the minimum voltage required for that frequency. Increasing the voltage above that minimum drains the battery. The device data sheet includes specifications for the minimum voltage. However, these limits are set to guarantee the operation of all devices with a wide range of PVT variations. Consider the example from the i.MXRT700 data sheet. [Table 1](#) is created from the "General Operating Conditions" table in the data sheet. See the data sheet for the latest specifications. For instance, on the Compute domain, when the CPU0 clock (compute_main_clk) is clocked at 192 MHz, the VDD2 supply rail requires a minimum voltage of 0.9 V to operate.

On the Sense domain, when the HIF11 clock (sense_dsp_clk) is clocked at 100 MHz, the VDD1 supply rail requires a minimum voltage of 0.8 V to operate.

Automatic Voltage Scaling using Process, Voltage, and Temperature Sensor (PVTs) on i.MX RT700

Table 1. VDD1 and VDD2 operating conditions when using an external PMIC

Digital rail	Conditions	Minimum voltage (V)
VDD1	Active mode ; CPU1/HIFI1 max freq = 250 MHz	1.1
	Active mode ; CPU1/HIFI1 max freq = 205 MHz	1.0
	Active mode ; CPU1/HIFI1 max freq = 160 MHz	0.9
	Active mode ; CPU1/HIFI1 max freq = 100MHz	0.8
	Active mode ; CPU1/HIFI1 max freq = 45 MHz	0.7
	Deep Sleep mode	0.63
	Deep Sleep mode async	0.5
VDD2	Active mode ; CPU0/HIFI4 max freq = 325 MHz	1.1
	Active mode ; CPU0/HIFI4 max freq = 250 MHz	1.0
	Active mode ; CPU0/HIFI4 max freq = 192 MHz	0.9
	Active mode ; CPU0/HIFI4 max freq = 110MHz	0.8
	Active mode ; CPU0/HIFI4 max freq = 45 MHz	0.7
	Deep Sleep mode	0.63
	Deep Sleep mode async	0.5

The 0.9 V for VDD2 and 0.8 V for VDD1 in the previous example are requirements based on the worst-case conditions; an SS device running at a temperature of -30 °C. Since most devices manufactured are faster than SS, most devices can clock at 192 MHz for VDD2 or 100 MHz for VDD1 with a voltage less than 0.9 V and 0.8 V, respectively. Since most devices do not spend much time operating at -30 °C and are closer to room temperature, it enables operating at a voltage lower than 0.9 V and 0.8 V. It is rare that a device requires 0.9 V to operate at 192 MHz for the Compute domain or 0.8 V to operate at 100 MHz for the Sense domain.

To operate at the ideal minimum voltage for a given frequency, each unique device must determine that voltage at runtime, while accounting for temperature changes. The integrated PVTs sensor is the solution to this challenge.

3 Operation with PVTs sensors

This section describes the operation with PVTs sensors.

3.1 PVTs architecture

i.MX RT700 has 2 PVTs, 1 in each domain (Compute and Sense). Each PVTs instance has 2 delay lines able to monitor 2 clocks in the same power domain in parallel. Thus, in the Compute domain with PVTs0, 1 delay line monitors CPU0 and the other delay line monitors HIFI4. In the Sense domain with PVTs1, 1 delay line monitors CPU1 and the other delay line monitors HIFI1. The 2 PVTs instances are completely independent and can be used separately or in parallel.

The PVTs sensor is a peripheral in the i.MX RT700 that monitors the timing margin of a given clock and triggers an interrupt when this timing margin is not met, which indicates that VDD2 and/or VDD1 must be increased to meet the required timing. [Figure 3](#) shows a closed loop system using an external PMIC to adjust the voltages to VDD2 and/or VDD1. When conditions change and the PVTs sensors indicate that VDD2 and/or VDD1 must be increased, the application requests the PMIC to increase this voltage through I2C.

PVTs0 is accessible both by CPU0 and HIFI4, meaning each core can enable and configure its delay line and PVTs0 instance on itself. The same mechanism exists for PVTs1 with CPU1 and HIFI1.

However, only CPU0 is connected to the PVTS0 interrupt and only CPU1 is connected to the PVTS1 interrupt. For instance, if PVTS0 monitors only HIFI4 while CPU0 is in a low-power mode, CPU0 must be woken up to take care of the interrupt generated by the delay line monitoring HIFI4 and potentially increase VDD2 by a PMIC step.

As VDD2 and VDD1 are controlled by the same external PMIC using LP_FLEXCOMM as the I2C and if both PVTS0 and PVTS1 are enabled, we must implement a secure way to access the PMIC. We can implement a mutex to ensure that the PMIC using the I2C is accessed by 1 core at a time (CPU0 or CPU1).

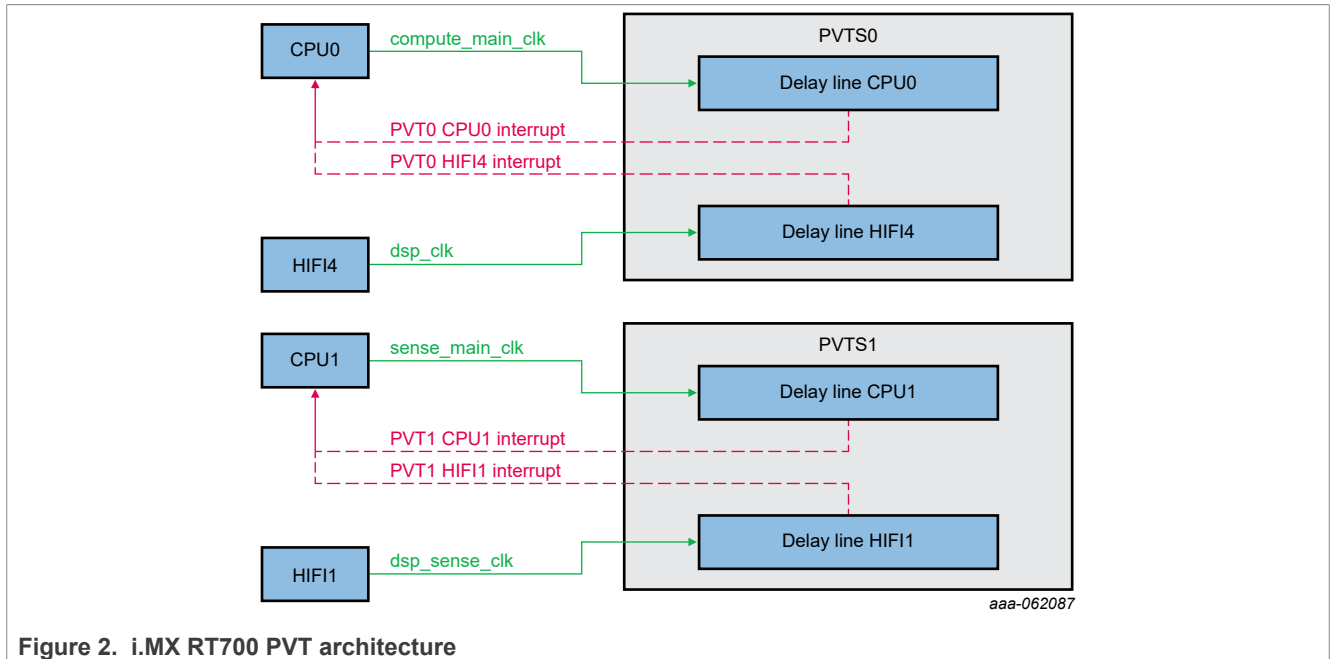


Figure 2. i.MX RT700 PVT architecture

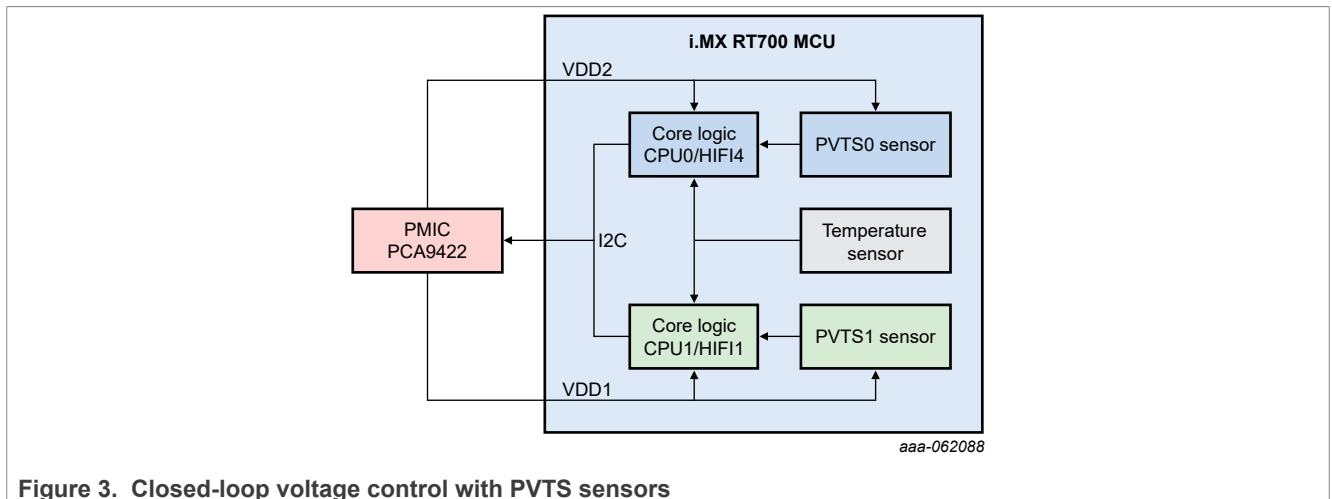


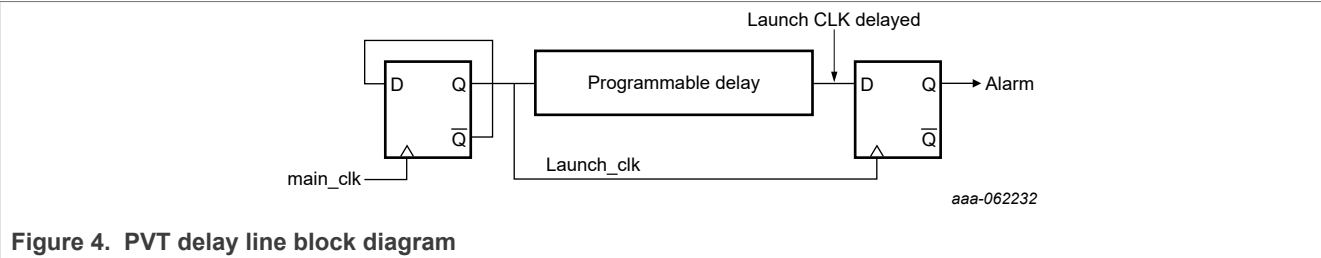
Figure 3. Closed-loop voltage control with PVTS sensors

3.2 Monitor timing margin

In this chapter, CPU0 is monitored by PVTS0 using the compute_main_clk clock. We use the same mechanism when a PVT instance monitors CPU1, HIFI1, or HIFI4.

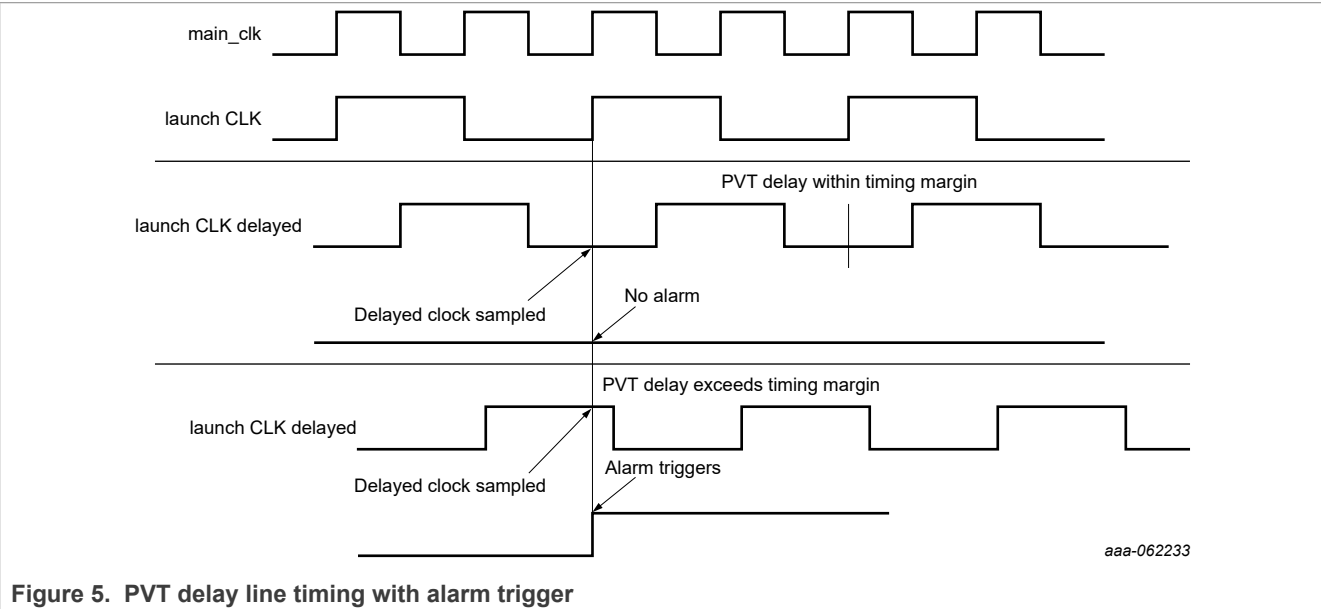
The PVTS sensor monitors the timing margin of the compute_main_clk clock using a delay line; see Figure 4. The compute_main_clk clock is divided by 2 and generates the Launch CLK. The Launch CLK is compared to a delayed version of itself. This delay is programmable and must be set to a value that correlates to the speed

of that unique device (as in SS/TT/FF). The timing comparison is done through a flip-flop that uses Launch CLK to clock the delayed version of itself. If the added delay is too long, the flip-flop output is high, indicating a timing issue and triggering an alarm.



For the PVTs0 sensor to properly monitor the compute_main_clk clock timing, we must program the delay line with a delay that matches the worst-case timing path of the logic clocked by the compute_main_clk clock. Once programmed with this delay, the PVTs0 sensor emulates this worst-case timing path in the VDD2 logic. This delay tracks the actual timing required by this unique device across temperature and voltage variations. As the voltage decreases or as the temperature gets colder, the delayed Launch CLK has a longer delay. When the alarm signal triggers, the application increases the VDD2 voltage, which increases the performance of the logic, which decreases the delay on the Launch CLK.

Figure 5 shows the timing of the PVTs0 delay line. The status of the delayed Launch CLK is sampled on every rising edge of the Launch CLK. When operating above the minimum voltage needed for that device and current temperature, the delay is short and the delayed Launch CLK is low at the rising edge of the Launch CLK, so the alarm output remains low. However, when the VDD2 voltage is too low for the current conditions, the delay is so long that the delayed Launch CLK is still high at the rising edge of the Launch CLK. The alarm output is high, which triggers an interrupt to indicate that we need a higher VDD2 voltage to reduce the timing delay.



3.3 Reducing supply voltage at runtime

In this section, PVTs0 monitors CPU0 at 192 MHz. However, the mechanism remains the same for the other cores (CPU1, HIFI1, HIFI4) at different supported frequencies.

When using the PVTs0 sensor and reducing VDD2, the PVTs0 alarm indicates when a higher voltage is required. However, PVTs0 does not give any indication of when VDD2 can be reduced further. Instead, the

Automatic Voltage Scaling using Process, Voltage, and Temperature Sensor (PVTS) on i.MX RT700

application tests a lower voltage periodically. If that reduced voltage is too low, PVTS0 immediately triggers the alarm and the application increases the voltage back to the original level. If no alarm is triggered, VDD2 can remain at that reduced voltage.

[Figure 6](#) shows an example of VDD2 dynamically changing as the temperature changes. This chart is an example of clocking at 192 MHz. The minimum VDD2 specification for CPU0 at 192 MHz in the data sheet is 900 mV. This graph starts where the VDD2 in blue has already been reduced to 850 mV using PVTS0 and operating at 30 °C. We use the PMIC PCA9422, which has voltage steps of 6.25 mV. When the temperature drops, the PVTS0 delay line monitoring CPU0 gets slower. At some point, the PVTS0 alarm triggers. The application requests the PMIC to increase the voltage by 1 step to 856.25 mV. The temperature continues to drop, so another PVTS0 alarm triggers, and VDD2 is increased to 862.5 mV. The application is continually testing if a reduced voltage is acceptable. Reducing the voltage by 1 step immediately triggers the PVTS0 interrupt and the voltage is increased 1 step back to the previous level. As the temperature rises, at some point the reduced voltage test does not trigger an alarm and PVTS0 allows the VDD2 to be reduced by 1 step to 856.25 mV. As the temperature continues to rise, this example eventually drops down to 800 mV. As the temperature cools back to 30 °C, PVTS0 triggers its alarm. When the voltage must be increased, the VDD2 returns to 850 mV.

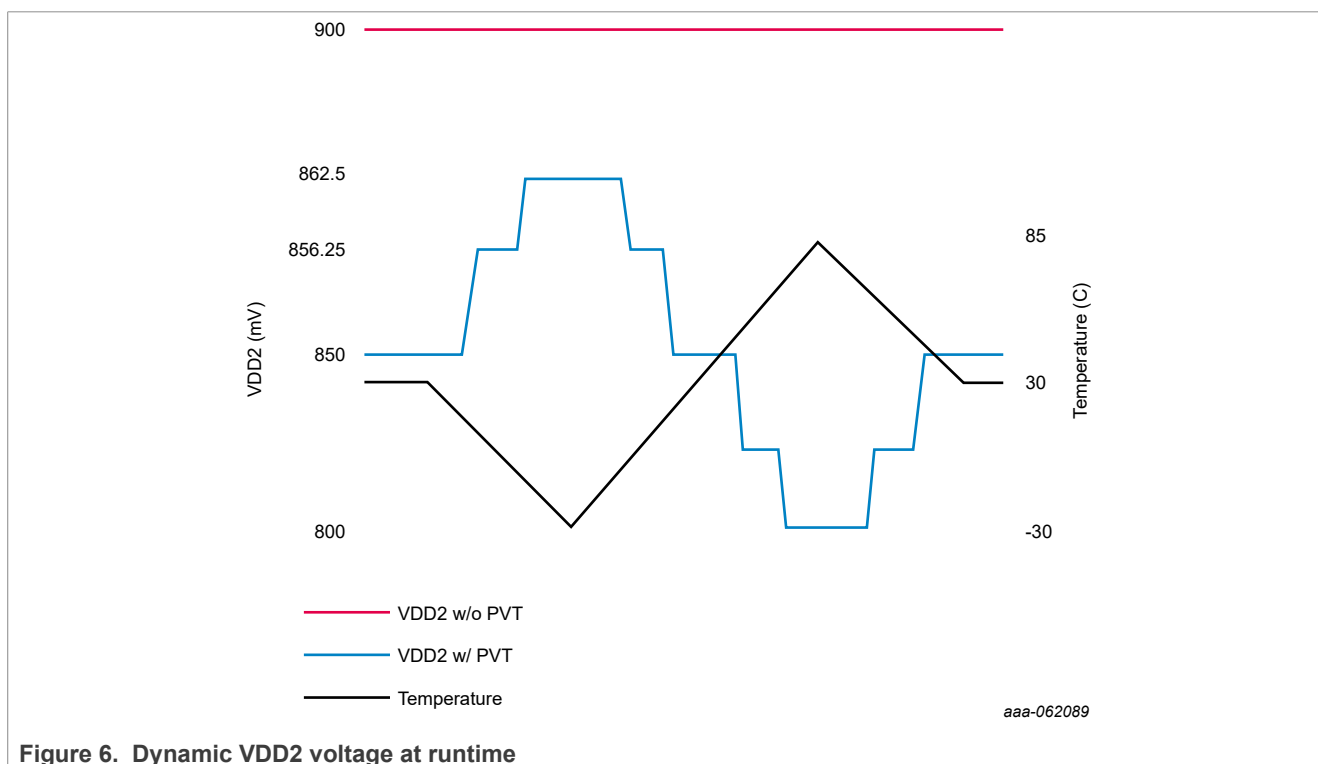


Figure 6. Dynamic VDD2 voltage at runtime

When the application reduces VDD2 using the PVTS sensor, it can test for a lower voltage. This method reduces the voltage by 1 step, waits for the voltage to settle, and then repeats. See [Figure 7](#) for an example of this reduction method. When PVTS0 detects that the voltage is too low, it interrupts the application, which raises the voltage by 1 step.

[Section 2.3](#) discusses the minimum voltage requirements for the PVTS sensor. If PVTS0 has already reduced the voltage to the minimum allowed by the application, the application can stop testing for a lower VDD2 until after VDD2 is increased again.

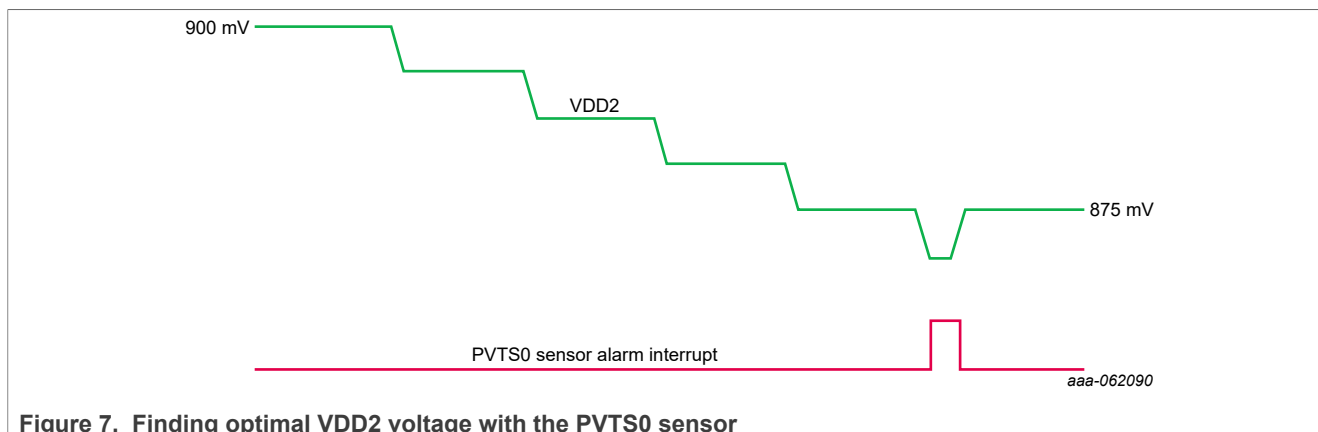


Figure 7. Finding optimal VDD2 voltage with the PVTs0 sensor

After initializing the PVTs sensor, the goal is to quickly find the minimum acceptable voltage in the current conditions. With each reduction in voltage, the application must give the PMIC and supply voltage rail enough time to settle at the new voltage. NXP's testing with the PCA9422 PMIC found that when the PMIC changes the voltage by 1 step, the supply rail settles very quickly. The application needed more time to control the PMIC using I2C and send the new register settings. The time measured for this PMIC communication was about 2 ms from calling the API to update the PMIC to when the PMIC changed the VDD2 voltage. With some added headroom, the SDK example uses a 5 ms delay for the PMIC during the initial voltage reduction. Note that this delay can be affected by the source clock of the I2C used to communicate with the external PMIC.

For applications using the PVTs sensor, the initial voltage reduction ends when the PVTs sensor alarm triggers. The application then increases the voltage by 1 step, and finds the minimum acceptable voltage in the present conditions. The example in [Figure 7](#) assumes the clocking at 192 MHz and starts at the VDD2 safe voltage of 900 mV. The voltage decrements by 6.25 mV per step and waits for the VDD2 voltage to settle. When VDD2 is reduced to 868.75 mV, the PVTs sensor interrupt immediately triggers. The application increases VDD2 by 1 step and finds the optimal voltage of 875 mV.

The next stage for the application is to check if the temperature has changed enough that a lower voltage is acceptable. To check this, 1 method is to periodically reduce the voltage. Based on the analysis of the thermal mass of the system and testing applications with different delay lengths, NXP found that testing for a lower voltage every 10 s is a good option for most applications. This application example uses a task timer to test every 10 s. Other applications can change this test period to optimize. Shorter periods mean that the app spends more time adjusting the PMIC and testing the voltage. The downside of longer periods is that the app may execute at a higher power consumption longer than necessary.

4 Ideal PVT delay settings

As described in [Section 3.2](#), the key to tune the PVTs sensor is to program a delay line with a value that correlates with the process variation (speed) of each unique device. NXP has tested and characterized many devices that range from the SS limit to FF. This testing was done to find the ideal PVT delay value to use on each device. NXP tested it with the firmware written to exercise many of the critical timing paths in the SoC and tested at various temperatures to find the minimum VDD1 and VDD2 voltages required under all these test conditions. Then, for all these devices and conditions, the PVT delay values were found to correlate with the minimum voltage of each device. For the same minimum VDD2 in the Compute domain, the PVT delay value can be different between CPU0 and HIFI4. The same logic exists for the Sense domain VDD1 with CPU1 and HIFI1. This feature has found reliable PVT delay values that optimize the VDD2 and VDD1 voltages but have enough margin to avoid faults at runtime. This margin allows the application to use the PVTs alarm as an early indication that the voltage is getting too low before a failure occurs. The application then has time to increase the voltage before conditions change further, but the application must do so immediately using the PVTs sensor interrupt.

Automatic Voltage Scaling using Process, Voltage, and Temperature Sensor (PVTS) on i.MX RT700

To better enable using the PVTS sensor, NXP started programming the ideal PVT delay value in the OTP fuses. NXP recommends using the value stored in these fuses, as they were tested and confirmed with each device during the manufacturing test.

4.1 PVT delays stored in OTP fuses

NXP programs each i.MX RT700 device during manufacturing with the ideal PVT delay values to use. NXP recommends reading this value using the PVTS driver and `PVTS_ReadDelayFromOTP()` API. Then program the PVTS sensor with this delay value using `PVTS_SetDelay()`.

Table 2. PVT fuses for i.MX RT700

Cores	Frequency	OTP address	Data
CPU0	325 MHz	483	Bits 0-7
HIFI4			Bits 8-15
CPU0	192 MHz	485	Bits 0-7
HIFI4			Bits 8-15
CPU0	110 MHz	487	Bits 0-7
HIFI4			Bits 8-15
CPU1	250 MHz	482	Bits 0-7
HIFI1			Bits 8-15
CPU1	160 MHz	484	Bits 0-7
HIFI1			Bits 8-15
CPU1	100 MHz	486	Bits 0-7
HIFI1			Bits 8-15

- The OTP fuse word location that stores the PVT delay value depends on the clock frequency and the core; see [Table 2](#). The valid range of this value is from 2 to 62.
- `PVTS_ReadDelayFromOTP()` reports if this value in OTP is not in the valid range.
- Be aware of the requirements before reading the fuses in OTP. The data sheet specifies the minimum VDD2 voltage and frequency for reading OTP, and the application may need to increase VDD2 first. See the data sheet for the latest requirements.
- The OTP driver must be initialized before reading the fuses. `PVTS_ReadDelayFromOTP()` initializes the driver for the application if the `otp_initialized` argument is false.
- The MCUXpresso SDK includes an OTP driver example that demonstrates initializing the OTP and reading the fuses.

On i.MX RT700, the Sense domain cannot access the OTP fuses' location where the PVT delays are stored. Thus, when the PVTS1 is enabled on CPU1 and/or HIFI1, CPU0 must retrieve the PVT delays for the Sense domain in the OTP locations 482, 484, and 486 and send it to CPU1. Once received, CPU1 can configure and start PVTS1. A possible implementation is to use the messaging unit to transfer data, in that case the PVT delays, from CPU0 to CPU1.

4.2 Optimizing PVT delays with temperature

The programmable delay in the PVTS sensor does not perfectly correlate with the temperature with the ideal VDD2 or VDD1. The PVT delay value is a safe value that works for the whole temperature range, including the worst-case conditions at -30 °C. However, the power can be further optimized at warmer temperatures. When the temperature is at or above 30 °C, we can potentially reduce the PVT delay value; see [Table 3](#). Doing so

Automatic Voltage Scaling using Process, Voltage, and Temperature Sensor (PVTS) on i.MX RT700

potentially allows reducing VDD2 or VDD1 further. Updating the PVT delay at runtime requires the application to monitor the temperature. To monitor the temperature, use the built-in temperature sensor with the ADC.

Table 3. PVT delay values depending on the temperature range

Cores	Frequency	-30 °C to 30 °C	30 °C to 85 °C
CPU0/HIFI4	325 MHz	PVT delay = OTP value	PVT delay = OTP value - 1
CPU0/HIFI4	192 MHz		PVT delay = OTP value - 1
CPU0/HIFI4	110 MHz		PVT delay = OTP value - 1
CPU1/HIFI1	250 MHz		PVT delay = OTP value
CPU1/HIFI1	160 MHz		PVT delay = OTP value
CPU1/HIFI1	100 MHz		PVT delay = OTP value - 1

The application can then use the following sequence of API calls to update the PVT delay value based on temperature:

1. PVTS_Stop()
2. PVTS_SetDelay(delay)
3. PVTS_ClearAlertCount()
4. PVTS_Start()

Updating the PVTS delay with this additional optimization requires that the application monitors the temperature. Some applications may find that monitoring the temperature and changing the PVTS delay is not worth doing. Those applications can use the fixed PVTS delay value with no optimization.

5 Using the PVTS sensor in application

This section describes the use of the PVTS sensor in an application.

5.1 Requirements for using the PVTS sensor

Proper usage of the PVTS sensor can allow applications to operate at VDD2 and/or VDD1 levels below the minimum data sheet specifications. There are various requirements to carefully consider and follow for an application to run safely.

The minimum VDD2 and VDD1 data sheet voltages are not necessarily limited by the compute_main_clk, dsp_clk, sense_main_clk, and sense_dsp_clk frequencies alone. There are more data sheet restrictions on VDD2 and VDD1 for specific peripherals, features, or use cases of the MCU that are in the data sheet.

The PVTS sensors were designed to monitor the timing margin in the VDD2 and VDD1 domains. However, do not use it when other VDD2 or VDD1 data sheet restrictions are in place.

The following items and discussions present more limitations to consider. Always see the data sheet for the full details, latest specifications, and any other requirements.

- If there are multiple VDD2 and/or VDD1 minimum requirements that apply, use the highest minimum voltage, regardless of what the PVTS sensor reports.
- Clock sources can have the minimum VDD2 and VDD1 voltages requirements based on the trim frequency and dividers used.
- Reading the OTP fuses requires a high VDD2 voltage. It is critical when reading the PVT delay value stored in the OTP.
- TRNG and ELS cannot be used with PVT. The data sheet minimum voltage for VDD2 must be met when using the TRNG and ELS.

Automatic Voltage Scaling using Process, Voltage, and Temperature Sensor (PVTS) on i.MX RT700

Clock frequencies are very important to consider when using the PVTS sensor. The characterization and testing (described in [Section 4](#)) were performed with the `compute_main_clk` CPU0 clock and the `dsp_clk` HIFI4 clock at 325 MHz, 192 MHz, and 110 MHz. For the `sense_main_clk` CPU1 clock and the `sense_dsp_clk` HIFI1 clock, the characterization and testing were performed at 250 MHz, 160 MHz, and 100 MHz. Other frequencies are currently not supported with the PVTS sensor.

The application can also place the maximum VDD2 and/or VDD1 voltage limits used with the PVTS sensor. Consider a use case like the following example application. The `compute_main_clk` clock is at 192 MHz, clocked from the FRO0. The application does not use other features that require a higher VDD2 voltage. The data sheet's "General Operating conditions" minimum specification of 0.9 V applies for 192 MHz. It is the starting voltage for the application before it enables PVTS0.

Once enabled, the application tries to reduce the voltage. While using PVTS0, if the voltage increases to 0.9 V and PVTS0 triggers another alarm, the voltage does not have to be increased. The application can keep VDD2 at 0.9 V because the data sheet confirms this voltage works for all devices and across the full temperature range.

The VDD2 and VDD1 voltages' requirements in the data sheet and in this document are for the voltages at the VDD2 and VDD1 pins of the MCU. If you set the PMIC to the same minimum VDD2 and VDD1 limits, there is potential for the voltage at the VDD2 and VDD1 pins to be under the requirement, which could lead to issues. For example, if the minimum requirement for VDD2 is 0.8 V and the PMIC is set to supply 0.8 V, the voltage at the VDD2 pins could be less than 0.8 V. When maximizing the battery life and optimizing VDD2 with PVTS0, the application is accepting more risk by reducing the margin in the system. This can lead to the voltage margin for VDD2 operation down in the millivolt level. Therefore, consider the power supply system and the PCB design when driving VDD2 and VDD1.

The PMIC has a voltage accuracy tolerance for the output. The PMIC output voltage varies with temperature. Typically, a switching regulator is driving VDD2 and VDD1, so that also introduces ripple on the supply rail. The VDD2/VDD1 voltages can also be lowered in the system before the VDD2/VDD1 pins by an IR drop or due to transients. Consider these and other impacts to the VDD2/VDD1 voltages during the hardware design and for the PMIC voltage setting at runtime to ensure that the VDD2/VDD1 voltages' requirements are met.

Another requirement is the settling time for the VDD2/VDD1 power rails. Anytime the application uses the PVTS sensor and reduces the VDD2/VDD1 voltage, it must wait long enough for the PMIC to change the voltage and the supply rail to settle to the new voltage. The exact delay depends on the board design, such as the PMIC/regulator driving VDD2/VDD1 and the capacitance on this supply rail. If the application drops the voltage too quickly before it has settled, the PVTS sensor cannot detect a timing issue before the voltage drops again and a fault occurs.

The final requirement is when both PVTS0 and PVTS1 are used to monitor the clock in the Compute and Sense domains. As these domains are powered by VDD2 and VDD1 (respectively) coming from the same external PMIC, if an interrupt is raised by both PVTs at the same time, then the application must ensure that only 1 access to the PMIC is done at a time. A mutex or semaphore can be used to ensure exclusive access between CPU0 and CPU1 to manage PMIC accesses via I2C.

5.2 PVTS sensor considerations with low-power modes

Using the low-power modes with the PVTS sensor can complicate the application. In this section, low-power modes are related to CPU0 and/or CPU1 in static modes.

For different power mode options, the Sleep mode does not have an impact when using the PVTS sensor. In the Sleep mode, the clock to the core is gated, but the `compute_main_clk` or `sense_main_clk` clocks remain clocked and the PVTS sensor operates in the Sleep mode and can wake the MCU with its interrupt. Waking from ultra-low power modes, such as DPD (Deep Power Down) and FDPD (Full Deep Power Down), wakes through the reset process, which means that the application must start at safe VDD2/VDD1 voltages and restart the PVTS sensors.

Automatic Voltage Scaling using Process, Voltage, and Temperature Sensor (PVTS) on i.MX RT700

The power mode considered here with the PVTS sensor is the Deep Sleep mode. Consider a use case similar to this example application, clocking the `compute_main_clk` clock at 192 MHz. The MCU is in the Active mode using the PVTS sensor. VDD2 originally started at 0.9 V, and started dropping with PVTS0. The temperature is high and VDD2 drops to a lower voltage. Now, the application enters the Deep Sleep power mode, which also disables clocks. The logic is in a static state, so PVTS0 is no longer active. The MCU is in this state for a certain amount of time, and during this state the temperature drops. The worst-case example is when the temperature drops to the minimum of -30 °C. Now, the device wakes up and at these conditions, the device needs VDD2 of up to 0.9 V before it can clock properly at 192 MHz. If the application wakes up at the current VDD2 level and attempts to clock at 192 MHz, the logic is not going to operate correctly and a fault occurs. PVTS0 triggers the alarm immediately after the clock reaches a high frequency with a VDD2 voltage that is too low. However, as the logic is not operating correctly, the firmware cannot execute properly and the application cannot request higher voltage from the PMIC.

The PVTS sensor can still be used with low-power modes, but situations like the above example must be avoided. The safest option is to wake back to a safe VDD2 voltage. Revisit the example above, except this time when the MCU wakes, the hardware automatically raises the VDD2 voltage to 0.9 V. This can be done before the application resumes execution and before the FRO ungates the clock at 192 MHz. The application can execute at 192 MHz, reenables PVTS0, and reduce the voltage to an optimized level.

The same mechanism applies to CPU1 in the Sense domain monitored by PVTS1 and managing the VDD1 power rail.

The PMIC_MODE pins on the MCU enable the above method to wake up at a preconfigured safe voltage. There are 2 PMIC_MODE pins, which allow the MCU to request up to 4 different power configurations in hardware. The PMIC_MODE 0b00 is a special mode because the MCU switches to this mode after a POR or when waking from the DPD or FDPD modes. The other 3 PMIC_MODE options are arbitrary and can be used by the application as desired. The PMIC can be preconfigured to respond to these mode pins with the desired power configuration. When waking from the Deep Sleep mode, the PMIC_MODE pins revert to the state in the PDRUNCFG0[PMIC_MODE] bits.

There are 4 PMIC_MODE instances in the i.MX RT700:

- 2 in the Compute domain, part of PMC0:
 - In the PMC0->PDRUNCFG0 register to control the PMIC_MODE configuration when CPU0 is active.
 - In the PMC0->PDSLEEPCFG0 register to control the PMIC_MODE configuration when CPU0 is in a low-power mode.
- 2 in the Sense domain, part of PMC1:
 - In the PMC1->PDRUNCFG0 register to control the PMIC_MODE configuration when CPU1 is active.
 - In the PMC1->PDSLEEPCFG0 register to control the PMIC_MODE configuration when CPU1 is in a low-power mode.

The final PMIC_MODE configuration pins' value is the aggregated result of the PMC0->PDxxCFG0 and PMC1->PDxxCFG0 registers, depending on the state of CPU0 and CPU1. For more details, see the PMC chapter of the reference manual.

For each PMIC_MODE configuration, different programmable output voltages are available. Using the i.MX RT700 EVK with PCA9422, SW1 is connected to VDD2 and SW3 is connected to VDD1. Thus, the proper voltage must be programmed in the correct output voltage setting.

The downside of the safe voltage method is that the application must first raise VDD2/VDD1 to a safe voltage level before entering the Deep Sleep mode. This process is not the most energy-efficient. Some applications can wake and resume execution at the same reduced voltage, but it depends on the use case of each application. When the MCU is in the Deep Sleep mode, temperature change is the main condition that can cause an issue with the VDD2/VDD1 voltages during a wakeup. Applications where ambient temperature does not change much can wake to the previous voltage found with the PVTS sensor. Some applications can have large temperature changes but wake up fast enough relative to the slower temperature change. In those cases,

the temperature cannot change much during the short Deep Sleep duration. If an application chooses to wake with a reduced voltage, these limitations and conditions must be well understood and considered.

The application can measure the temperature with the integrated temperature sensor and compare it with a stored temperature from before sleeping. A better option is to use a hardware timer and measure its duration in the Deep Sleep mode. Either way, if the temperature change or sleep duration are large enough, the application increases VDD2/VDD1 to a safe voltage first, before switching to the fast clock frequency. If the application determines that the change is minimal compared to conditions before entering the Deep Sleep mode, it can resume at the reduced voltage found by the PVTS sensor before sleeping.

5.3 Potential implementation

i.MX RT700 is a complex and flexible device. Domain states can be controlled independently and PVTS instances also operate independently. For instance, the Sense domain can be in the Deep Sleep mode while the CPU0 is in the Sleep mode and HIFI4 is running, monitored by PVTS0. As another example, both the Compute and Sense domains can be ON with CPU0 and CPU1 active, monitored by PVTS0 and PVTS1 (respectively).

As a general guideline, here are some elements to consider while developing an application:

- OTPs' fuses are accessible only by CPU0 and/or HIFI4. If CPU1 or HIFI1 PVTS1 in the Sense domain are enabled, one can use the messaging unit to transfer the PVT delays' values between domains.
- PVTS configuration and start can be done by all cores in their respective domain; i.e. CPU0 and HIFI4 can configure PVTS0, and CPU1 and HIFI1 can configure PVTS1. As the same PVTS is used to monitor CPU0 and HIFI4, PVTS0 and related delay lines can be programmed by CPU0, HIFI4, or both, depending on the application. The same concept applies to CPU1 and HIFI1 with PVTS1.
- PVTS interrupts are connected only to the CPU cores. Both interrupts related to the 2 delay lines of the same PVTS must be handled by the CPU core.
- When the 2 PVTS0 and PVTS1 are used in parallel, you must implement a mutex or a semaphore to secure the access to the PMIC via I2C, while modifying the digital rail.
- When only the HIFI is enabled and monitored in a domain, for instance HIFI4 monitored by PVTS0 in the Compute domain, the related CPU, in this case CPU0, must be in the sleep mode to allow the PVT to run properly.

5.4 Configuring Low-Voltage Detectors (LVDs)

The i.MX RT700 integrates Low-Voltage Detectors (LVDs) that monitor the VDD2 and VDD1 supply rails separately. For VDD2, the low-voltage detection threshold is programmable using `PMCx->PD*CFG0[LDO2_VSEL]` and `PMCx->LVDVDD2CTRL`. For VDD1, the low-voltage detection threshold is programmable using `PMCx->PD*CFG0[LDO1_VSEL]` and `PMCx->LVDVDD1CTRL`.

The `LVDVDDxCTL` register sets the LVD falling trip voltage for 4 levels, with the falling trip = $0.5 + 10 \text{ mV} \times \text{value of the field}$.

`LDOx_VSEL` selects one of the 4 LVD trip points in `LVDVDDx_CTRL`.

These registers are present in PMC0 and PMC1 in the Compute and Sense domains, meaning that the result is the aggregated value of these registers.

There are some important considerations when using LVD with DFVS and with the PVTS sensor. LVD provides the following 2 primary functions:

- LVD can trigger when VDD2/VDD1 falls below the threshold. It can interrupt the application as an early warning, or it can reset the MCU.
- On initial powerup and on wakeup from the Deep Sleep mode, LVD determines when VDD2/VDD1 is high enough to allow the MCU to operate.

Automatic Voltage Scaling using Process, Voltage, and Temperature Sensor (PVTs) on i.MX RT700

The LVDs' minimum falling threshold is 500 mV and the application can increment in 10-mV steps using LVDVDDxCTL with a maximum of 1.13 V. The rising threshold is 30 mV above the falling threshold for hysteresis. However, there are tolerances for the threshold across the PVT variation. It means that each LVDVDDxCTL setting has ranges for each threshold, as shown in [Table 4](#). This table shows the LVD threshold ranges for level [LVL3], but it is the same for all 4 levels ([LVL0], [LVL1], [LVL2], [LVL3]) in LVDVDDxCTL.

Table 4. i.MXRT 700 LVDs threshold ranges

LVDVDDxCTL	Falling voltage (mV)		Rising voltage (mV)	
	Min	Max	Min	Max
0	483	517	512	548
1	493	527	522	558
2	503	537	532	568
3	512	548	542	578
4	522	558	552	588
5	532	568	561	599
6	542	578	571	609
7	552	588	581	619
8	561	599	591	629
9	571	609	601	639
10	581	619	610	650
11	591	629	620	660
12	601	639	630	670
13	610	650	640	680
14	620	660	650	690
15	630	670	659	701
16	640	680	669	711
17	650	690	679	721
18	659	701	689	731
19	669	711	699	741
20	679	721	708	752
21	689	731	718	762
22	699	741	728	772
23	708	752	738	782
24	718	762	748	792
25	728	772	757	803
26	738	782	767	813
27	748	792	777	823
28	757	803	787	833
29	767	813	797	843
30	777	823	806	854

Automatic Voltage Scaling using Process, Voltage, and Temperature Sensor (PVTs) on i.MX RT700

Table 4. i.MXRT 700 LVDs threshold ranges...continued

LVDVDDxCTL	Falling voltage (mV)		Rising voltage (mV)	
	Min	Max	Min	Max
31	787	833	916	864
32	797	843	826	874
33	806	854	836	884
34	816	864	846	894
35	826	874	855	905
36	836	884	865	915
37	846	894	875	925
38	855	905	885	935
39	865	915	895	945
40	875	925	904	956
41	885	935	914	966
42	895	945	924	976
43	904	956	934	986
44	914	966	944	996
45	924	976	953	1007
46	934	986	963	1017
47	944	996	973	1027
48	953	1007	983	1037
49	963	1017	993	1047
50	973	1027	1002	1058
51	983	1037	1012	1068
52	993	1047	1022	1078
53	1002	1058	1032	1088
54	1012	1068	1042	1098
55	1022	1078	1051	1109
56	1032	1088	1061	1119
57	1042	1098	1071	1129
58	1051	1109	1081	1139
59	1061	1119	1091	1149
60	1071	1129	1100	1160
61	1081	1139	1110	1170
62	1091	1149	1120	1180
63	1100	1160	1130	1190

The LVD's interrupt and reset are optional. In the Active mode, the application can disable these features and ignore the LVD. However, if the application uses the LVD for its monitoring feature while dynamically changing

the VDD2/VDD1 voltages, we must configure the LVD's threshold. One option is to consider the minimum VDD2/VDD1 voltages allowed by the application for all devices; 800 mV is a good example. The application can set LVDVDDxCTL to the falling threshold range that is below the minimum voltage at the pins of the rail. Another option for applications is to adjust the LVD's threshold when changing VDD2/VDD1, keeping them more tightly aligned.

If using the Deep Sleep mode and reducing VDD2/VDD1, the LVDs have further considerations. If the application lowers VDD2/VDD1 below the LVD falling threshold, then the LVD reset and interrupt must be disabled. The LVD rising threshold must also be set above the minimum VDD2/VDD1 voltage needed for the wakeup. This LVD value must be set before entering the Deep Sleep mode. The example of waking to a safe voltage, as discussed in [Section 5.2](#), considers a minimum of 900 mV for VDD2. To guarantee that the voltage at the VDD2 pins is at least 900 mV before the wake-up process starts, we can use LVDVDD2CTL = 40. However, the higher limit with option 40 is 956 mV. It requires the PMIC to drive VDD2 to at least 956 mV to ensure that the MCU wakes. After waking, the application can reduce LVDVDD2CTL and lower VDD2.

If the application does not use the LVDs, we can disable them completely. Use PMC0->CTRL[LVD2RE] to disable the LVD for VDD2. Use PMC0->CTRL[LVD1RE] to disable the LVD for VDD1. The CTRL register can only be written by PMC0 in the Compute domain. The CTRL register in PMC1 in the Sense domain is read-only.

5.5 Troubleshooting

Using the PVTs sensor can greatly benefit the power consumption. However, it can create some challenges when diagnosing issues in the application. There are several requirements and limitations to consider when reducing VDD2/VDD1 with the PVTs sensor. When issues related to the PVTs sensor occur, it is usually because the VDD2/VDD1 voltages are too low for the current conditions and clock frequencies.

Common symptoms when the logic cannot keep up with the clock frequency include the following:

- Application locked up or unresponsive
- Erratic and unpredictable faults that are difficult to replicate consistently
- Problems only occurring on a small percentage of devices because they are SS or FF devices
- Problems only occurring at certain temperatures
- Issues that happen while waking from the Deep Sleep mode

If issues like these occur and the PVTs sensor is used in the application, the first step is to disable the PVTs sensor. Keep VDD2/VDD1 at a fixed safe voltage. During debugging, we can use the maximum of 1.155 V. If the issue goes away, then it is likely related to VDD2/VDD1 going too low at some point. From there, we can debug the application to track down the point where the voltage rail is too low.

If using the PVTs sensor with low-power modes, try testing it by disabling the entry into the low-power modes to see if the issue goes away. If so, the issue is likely related to the entry/exit of a low-power mode. If the application is not waking to a safe voltage level, try testing that method to rule out VDD2/VDD1 during wakeup.

The LVDs can also be related to issues with using DVFS or the Deep Sleep mode. If unexpected resets are occurring, LVDs can trigger them. While debugging, try disabling the LVD's reset to see if it impacts the issue. Potentially, the LVDs' falling thresholds are set too high relative to the rail supplied and it occasionally triggers unwanted resets. If the LVDs are triggering, perhaps the issue is that VDD2/VDD1 is dropping lower than expected.

If issues are related to using the Deep Sleep mode, the LVDs can also be involved. For debugging, try disabling the LVDs' reset and interrupt before entering the Deep Sleep mode. The issue can also be related to VDD2/VDD1 and LVDs during wakeup. To rule out issues, increase LVDVDDxCTL to a high setting that ensures that VDD2/VDD1 are at a high and safe voltage before the MCU starts the wake-up process. Then, ensure that the PMIC is increasing VDD2/VDD1 above the upper limit of the LVDs rising threshold to trigger a wakeup.

Another method is using some output pins to capture the state of the MCU during the application. The CLKOUT_VDD2 and CLKOUT_VDD1 signals can be connected to a pin, and the compute_main_clk clock (or

a divided down main_clk clock or other clocks) and/or the sense_main_clk clock can be measured. Also, we can use a GPIO to track the entry/exit of low-power modes. The CLKOUT, wake GPIO, PMIC_MODE pins, and VDD2/VDD1 voltages can be monitored using test tools. It can identify when VDD2/VDD1 have issues during a wakeup or with changing clock frequencies. When measuring the VDD2/VDD1 voltages, try to measure as close to the MCU's VDD2/VDD1 pins as possible; for example, on the bypass capacitors. Measuring the output at the PMIC cannot expose how the VDD2/VDD1 pins are supplied.

6 Conclusion

The PVTs sensor integrated in the i.MX RT700 is a great option to increase the battery life in products. The operating requirements in this document are important for applications to avoid issues and maximize benefits. NXP has tested a large volume of i.MX RT700 devices using the PVTs sensor, following these guidelines and requirements. The power consumption on the VDD2 and/or VDD1 supply rails decreased on average 20 %, depending on the use case compared to devices running at the voltage required by the data sheet. The battery life benefits can be substantial. For applications where battery life matters the most, explore the PVTs sensor and see how it can help in your application.

7 Acronyms and abbreviations

Table 5. Acronyms and abbreviations

Acronym	Meaning
PVT	Process, Voltage, Temperature
MCU	Microcontroller
AVS	Automatic Voltage Scaling
DSP	Digital Signal Processor
PMIC	Power Management IC
SDK	Software Development Kit
I2C	Inter-Integrated Circuit
ADC	Analog-to-Digital Converter
OTP	One-Time Programmable
API	Application Programming Interface
FRO	Free-Running Oscillator
IR drop	I*R drop, voltage drop that occurs across a resistance
POR	Power-On Reset
DPD	Deep Power Down
FDPD	Full Deep Power Down
LVD	Low-Voltage Detect
DVFS	Dynamic Voltage Frequency Scaling

8 Revision history

Table 6 summarizes revisions to this document.

Table 6. Revision history

Document ID	Release date	Description
AN14791 v.3.0	10 November 2025	Initial public release
AN14791 v.2.0	22 September 2025	Updated frequency points
AN14791 v.1.0	19 August 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

1	Introduction	2
2	Process, Voltage, and Temperature (PVT) variation	2
2.1	Silicon process variation	2
2.2	Temperature variation	3
2.3	Minimum voltage required	3
3	Operation with PVTs sensors	4
3.1	PVTs architecture	4
3.2	Monitor timing margin	5
3.3	Reducing supply voltage at runtime	6
4	Ideal PVT delay settings	8
4.1	PVT delays stored in OTP fuses	9
4.2	Optimizing PVT delays with temperature	9
5	Using the PVTs sensor in application	10
5.1	Requirements for using the PVTs sensor	10
5.2	PVTs sensor considerations with low-power modes	11
5.3	Potential implementation	13
5.4	Configuring Low-Voltage Detectors (LVDs)	13
5.5	Troubleshooting	16
6	Conclusion	17
7	Acronyms and abbreviations	17
8	Revision history	17
	Legal information	19

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.
