AN13267

Adding DVFS Feature Support in i.MX RT1170

Rev. 0 — 28 May 2021

Application Note

Contents

1 Introduction

The i.MX RT1170 is a new i.MX RT family of processors that provides high CPU performance and real-time response. The i.MX RT1170 integrates advanced power management module with DCDC and LDOs that reduce complexity of external power supply and simplify power sequencing.

Dynamic voltage and frequency scaling (DVFS) is a power management technique that allows dynamically reducing power consumption of a CPU by dynamically scaling down supply voltage and CPU frequency. Because the internal DCDC of the i.MX RT1170 cannot cover the needed maximum current requirement at the junction temperature of 125 °C, the DVFS technique can be used to reduce current drain for compatibility with the internal DCDC. Lowering the processor frequency dynamically can help reduce the chip input current demand and ensure that the chip can continue to work at the junction temperature of 125 °C.

Adding DVFS feature support requires configuring the following modules of the i.MX RT1170:

- Clock Control Module (CCM)
- DCDC Converter (DCDC)
- Temperature Sensor (TEMP SENSE)
- General Purpose Timer (GPT) — optional

This document describes how to support DVFS feature in the i.MX RT1170 processor with internal DCDC for automotive applications. It also explains DVFS process workflow.

A demo corresponding to this document can be accessed from NXP Community.

2 i.MX RT1170 operating ranges

The figure below shows a portion of the "Operating ranges" table from *i.MX RT1170 Crossover Processors Data Sheet for Automotive Products* (IMXRT1170AEC) Rev. 1. It displays the i.MX RT1170 operating ranges in Run mode with DCDC enabled.

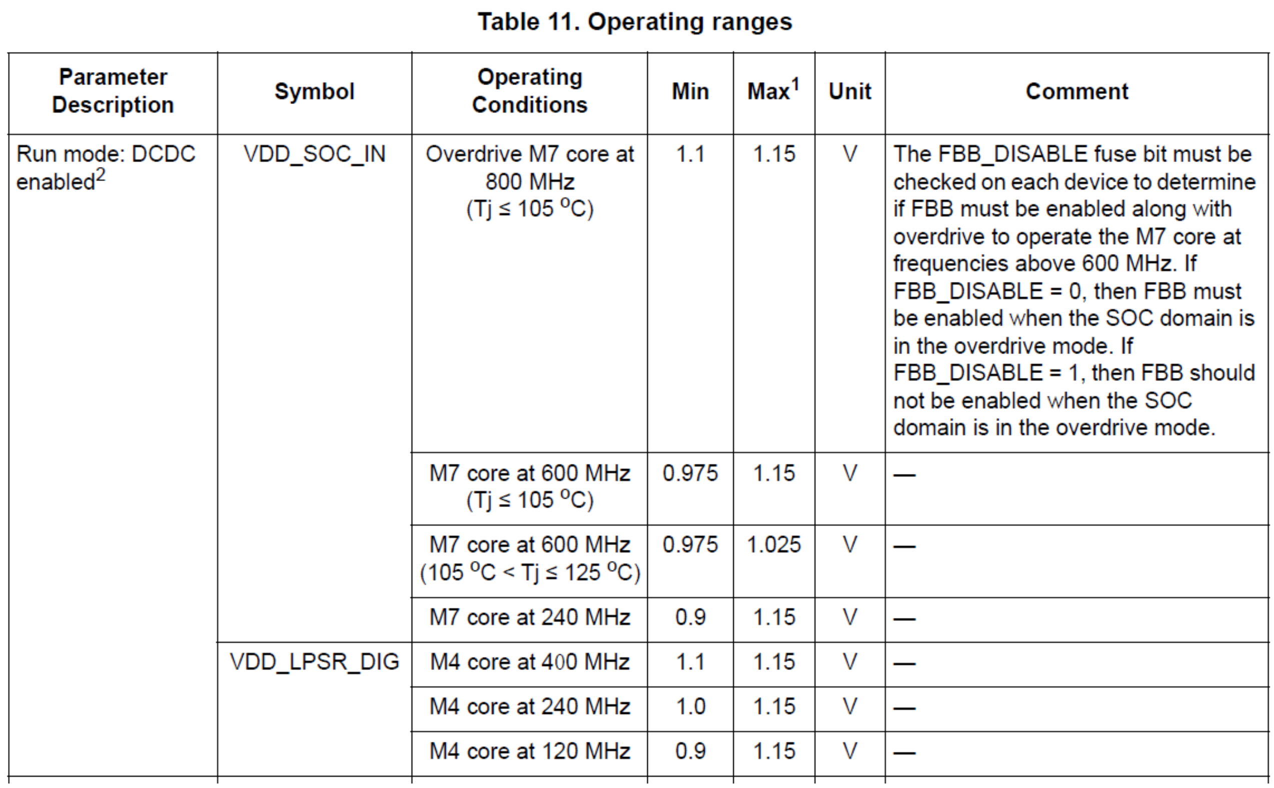


Table 11. Operating ranges

Parameter Description	Symbol	Operating Conditions	Min	Max[1]	Unit	Comment
Run mode: DCDC enabled[2]	VDD_SOC_IN	Overdrive M7 core at 800 MHz (Tj ≤ 105 °C)	1.1	1.15	V	The FBB_DISABLE fuse bit must be checked on each device to determine if FBB must be enabled along with overdrive to operate the M7 core at frequencies above 600 MHz. If FBB_DISABLE = 0, then FBB must be enabled when the SOC domain is in the overdrive mode. If FBB_DISABLE = 1, then FBB should not be enabled when the SOC domain is in the overdrive mode.
		M7 core at 600 MHz (Tj ≤ 105 °C)	0.975	1.15	V	—
		M7 core at 600 MHz (105 °C < Tj ≤ 125 °C)	0.975	1.025	V	—
		M7 core at 240 MHz	0.9	1.15	V	—
	VDD_LPSR_DIG	M4 core at 400 MHz	1.1	1.15	V	—
		M4 core at 240 MHz	1.0	1.15	V	—
		M4 core at 120 MHz	0.9	1.15	V	—

Figure 1. Operating ranges

As you can see in the figure above, voltage ranges are shown for different operating frequencies and temperature ranges at the core supply input voltage pin (VDD_SOC_IN) for the Cortex®-M7 core. The internal DCDC can only provide the VDD_SOC_IN maximum supply current at 105 °C if the Cortex-M7 core frequency is 800 MHz. When the i.MX RT1170 junction temperature is below 105 °C, the maximum current requirement for VDD_SOC_IN is 850 mA.

As temperature rises, the current required by VDD_SOC_IN increases. When the i.MX RT1170 junction temperature reaches 125 °C, the maximum current required reaches 1200 mA. However, the internal DCDC can only cover a current supply of 850 mA.

Therefore, if the junction temperature is higher than 105 °C and it is acceptable to lower the Cortex-M7 core frequency down to 600 MHz, the DVFS solution described in this document can be applied. This allows the i.MX RT1170 part to be functional without requiring external power components. When the temperature drops, the Cortex-M7 core clock can be switched back to 800 MHz to get higher performance.

While implementing DVFS, the state of the Cortex®-M4 core is not considered because it is powered by VDD_LPSR_DIG, and not from the internal DCDC.

3 Temperature alarms

The i.MX RT1170 has a temperature sensor, which supports the following features:

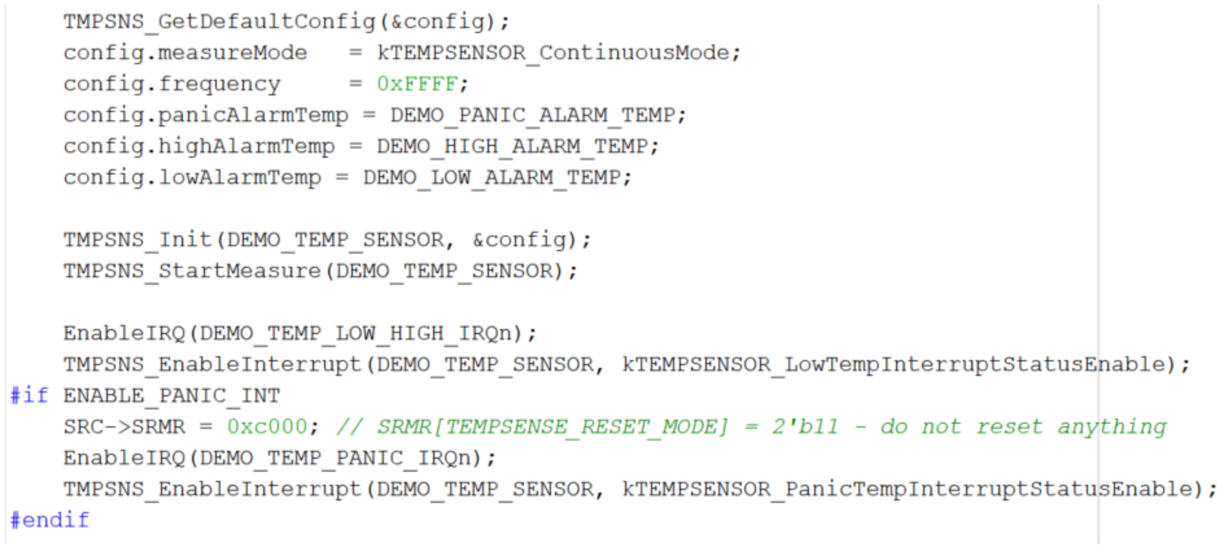
- Supports temperature alarm interrupts to indicate if temperature is beyond the programmed thresholds
- Supports three-tier alarms: Low, High, and Panic
- Supports Single Measurement and Continuous Measurement modes

This temperature sensor module uses a 24 MHz reference clock from the 24 MHz crystal oscillator. Because the field FREQ in the temperature sensor control register CTRL1 is 16 bits, the maximum detection interval of temperature sensor is about:

$$\frac{2^{16}}{2^{20} \times 24} s = 2.6\, ms$$

The temperature sensor module features alarm functions that can raise independent interrupt signals if the temperature increases above one or both high-temperature thresholds or decreases below the low-temperature threshold. These temperature thresholds are programmable and designated as low, high, and panic temperature. The panic threshold is a special programmable threshold in the sense that if the temperature increases above this value and the temperature-panic-reset is enabled in the System Reset Controller, the hardware will assume that software no longer has control over the thermal situation and will initiate a reset of the chip. The temperature-panic-reset can also be disabled. If so, panic interrupt is triggered when the temperature increases above the panic threshold. Note that panic interrupt and temperature-panic-reset cannot work at the same time.

DVFS mainly uses high and low temperature alarms. That is, when the temperature is higher than the high-temperature threshold value or lower than the low-temperature threshold value, the high-temperature or low-temperature interrupt is triggered. The interrupt trigger conditions in these two cases are different but they share the same interrupt number. Therefore, they enter the same interrupt processing function. Which interrupt source triggered the interrupt must be determined through the flag bit of STATUS0 register of temperature sensor. However, due to the delay in getting voltage and frequency stable, it is possible to encounter a situation where both high and low temperature flags are set. Therefore, the interrupt source cannot be determined only by the flag bit of STATUS0 register, but together with the current operating frequency. Part of temperature sensor configuration code is shown in the figure below.



```
    TMPSNS_GetDefaultConfig(&config);
    config.measureMode    = kTEMPSENSOR_ContinuousMode;
    config.frequency      = 0xFFFF;
    config.panicAlarmTemp = DEMO_PANIC_ALARM_TEMP;
    config.highAlarmTemp = DEMO_HIGH_ALARM_TEMP;
    config.lowAlarmTemp = DEMO_LOW_ALARM_TEMP;

    TMPSNS_Init(DEMO_TEMP_SENSOR, &config);
    TMPSNS_StartMeasure(DEMO_TEMP_SENSOR);

    EnableIRQ(DEMO_TEMP_LOW_HIGH_IRQn);
    TMPSNS_EnableInterrupt(DEMO_TEMP_SENSOR, kTEMPSENSOR_LowTempInterruptStatusEnable);
#if ENABLE_PANIC_INT
    SRC->SRMR = 0xc000; // SRMR[TEMPSENSE_RESET_MODE] = 2'b11 - do not reset anything
    EnableIRQ(DEMO_TEMP_PANIC_IRQn);
    TMPSNS_EnableInterrupt(DEMO_TEMP_SENSOR, kTEMPSENSOR_PanicTempInterruptStatusEnable);
#endif
```

Figure 2. Part of temperature sensor configuration code

4 Implementing DVFS feature

Implementing DVFS feature for the i.MX RT1170 involves dynamically configuring Cortex-M7 core frequency and input voltage. The table below shows the target frequency and target input voltage for the Cortex-M7 core at different junction temperatures. The target input voltage is calculated considering that the internal DCDC has a ripple within 25 mV. For this implementation, the target bus frequency is kept at 240 MHz.

Table 1. Target frequency and target input voltage

Junction temperature	M7 core target frequency	M7 core target input voltage (VDD_SOC_IN)
Tj ≤ 105 °C	800 MHz	1.125 V
Tj ＞ 105 °C	600 MHz	1.0 V

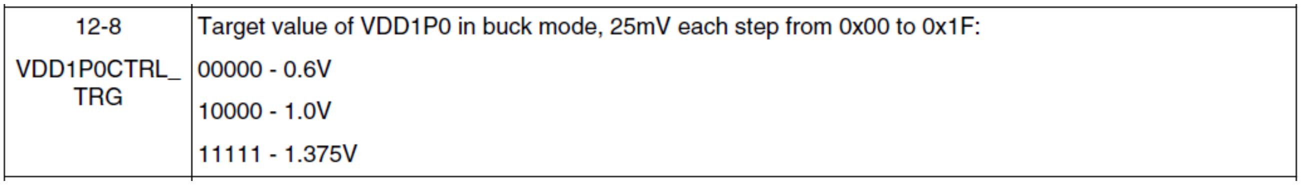
Achieving target frequency or target input voltage mentioned in the table above requires modification at the source that supplies the clock or voltage. The following subsections explain how to make these modifications.

4.1 Configure voltage

The DCDC module is used for generating power supply for core logic. It provides the following two outputs:

- DCDC_DIG (voltage: VDD1P0)
- DCDC_ANA (voltage: VDD1P8)

VDD1P0 supplies power to Cortex-M7 core. In Run mode, VDD1P0 output can be set by setting the VDD1P0CTRL_TRG bit field of DCDC Control Register 1 (CTRL1). The figure below shows VDD1P0 configuration in Run mode.



12-8 VDD1P0CTRL_ TRG	Target value of VDD1P0 in buck mode, 25mV each step from 0x00 to 0x1F: 00000 - 0.6V 10000 - 1.0V 11111 - 1.375V

Figure 3. VDD1P0 configuration in Run mode

The CTRL1 register bit field VDD1P0CTRL_TRG can be set using the `DCDC_SetVDD1P0BuckModeTargetVoltage()` function. The Cortex-M7 core target input voltages shown in Table 1 can be achieved by calling the `DCDC_SetVDD1P0BuckModeTargetVoltage()` function, as follows:

```
DCDC_SetVDD1P0BuckModeTargetVoltage(DCDC, kDCDC_1P0BuckTarget1P125V);
```

```
DCDC_SetVDD1P0BuckModeTargetVoltage(DCDC, kDCDC_1P0BuckTarget1P0V);
```

The `DCDC_SetVDD1P0BuckModeTargetVoltage()` function sets the output of VDD1P0 by setting the CTRL1 register bit field VDD1P0CTRL_TRG, and returns after the output is stable.

4.2 Configure frequency

The Cortex-M7 core uses ARM_PLL clock as the source clock. This clock is generated by a PLL, ARM_PLL, which uses a 24 MHz clock as reference clock. The figure below depicts ARM_PLL clock generation.

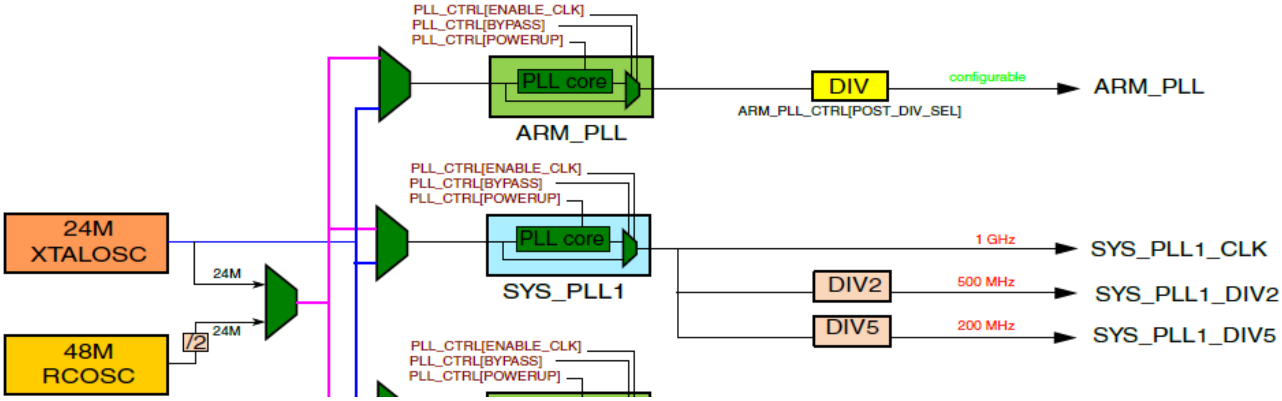


Figure 4. ARM_PLL clock generation

ARM_PLL clock frequency (Fout) is calculated as follows:

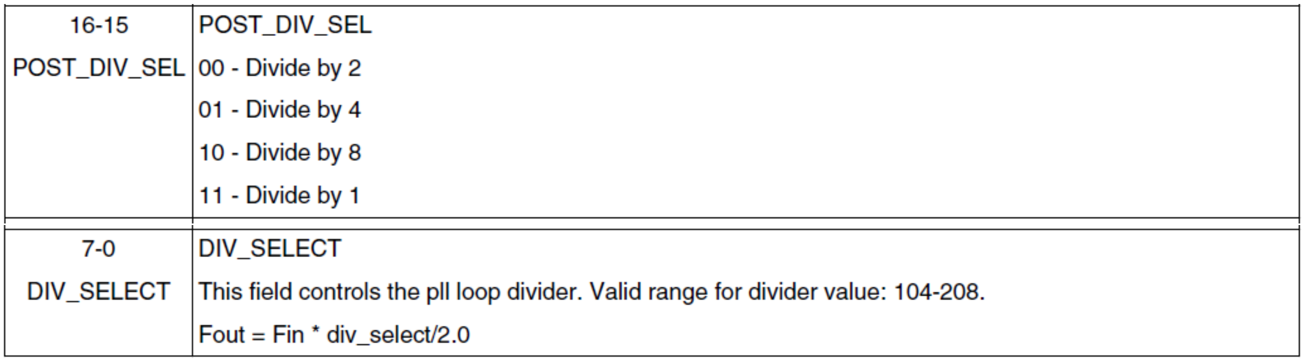
Fout = (Fin * DIV_SELECT / 2.0) / POST_DIV_SEL

where:

- Fin is the reference clock frequency

- DIV_SELECT and POST_DIV_SEL are bit fields of ARM_PLL_CTRL_REGISTER (ARM_PLL_CTRL)

The figure below shows DIV_SELECT and POST_DIV_SEL configuration.

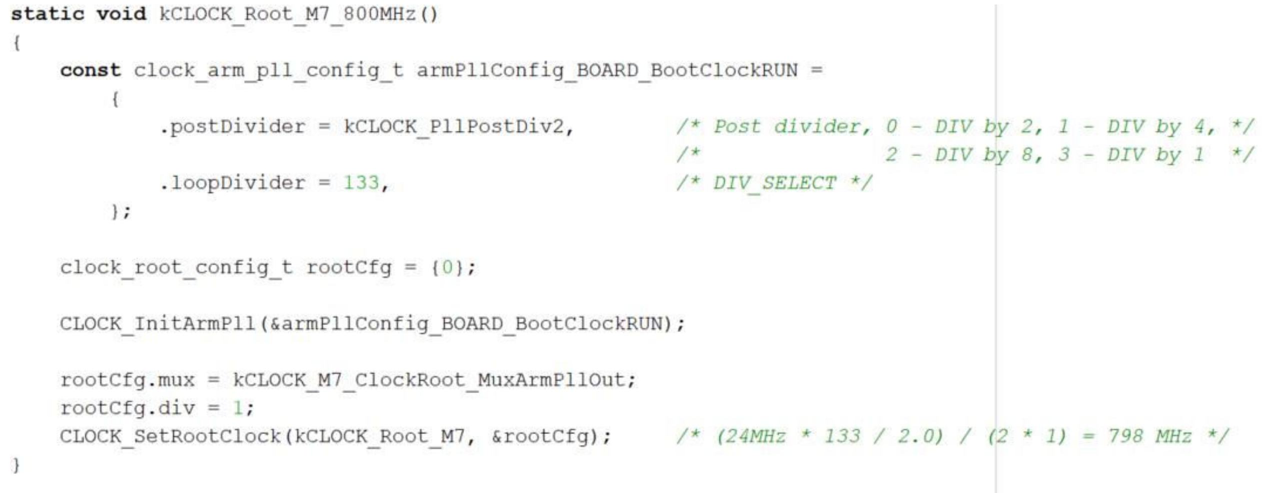


16-15 POST_DIV_SEL	POST_DIV_SEL 00 - Divide by 2 01 - Divide by 4 10 - Divide by 8 11 - Divide by 1
7-0 DIV_SELECT	DIV_SELECT This field controls the pll loop divider. Valid range for divider value: 104-208. Fout = Fin * div_select/2.0

Figure 5. DIV_SELECT and POST_DIV_SEL configuration

The Cortex-M7 core target frequencies shown in Table 1 can be achieved by setting the DIV_SELECT and POST_DIV_SEL bit fields of ARM_PLL_CTRL register.

For example, if the expected output frequency is 800 MHz, then the ARM_PLL_CTRL register bit fields and Cortex-M7 core frequency can be set using the configuration code given below.



```
static void kCLOCK_Root_M7_800MHz()
{
    const clock_arm_pll_config_t armPllConfig_BOARD_BootClockRUN =
        {
            .postDivider = kCLOCK_PllPostDiv2,        /* Post divider, 0 - DIV by 2, 1 - DIV by 4, */
                                                      /*               2 - DIV by 8, 3 - DIV by 1  */
            .loopDivider = 133,                       /* DIV_SELECT */
        };

    clock_root_config_t rootCfg = {0};

    CLOCK_InitArmPll(&armPllConfig_BOARD_BootClockRUN);

    rootCfg.mux = kCLOCK_M7_ClockRoot_MuxArmPllOut;
    rootCfg.div = 1;
    CLOCK_SetRootClock(kCLOCK_Root_M7, &rootCfg);     /* (24MHz * 133 / 2.0) / (2 * 1) = 798 MHz */
}
```

Figure 6. Core clock configuration code

Based on the configuration shown in the figure above:

Output frequency = (24 MHz * DIV_SELECT / 2.0) / (POST_DIV_SEL * rootCfg.div)

= (24 MHz * 133 / 2.0) / (2 * 1) = 798 MHz

NOTE

The uSDHC module of the i.MX RT1170 may be affected by the adjustment of ARM_PLL. Ensure that the clock source selected by uSDHC module is not ARM_PLL or switch uSDHC clock source to another stable clock while adjusting ARM_PLL.

5 DVFS process workflow

The figure below shows the example temperature diagram for the i.MX RT1170. It indicates when a high or low temperature interrupt is triggered by the temperature sensor. "High Temp" represents the high temperature threshold and "Low Temp" represents the low temperature threshold.

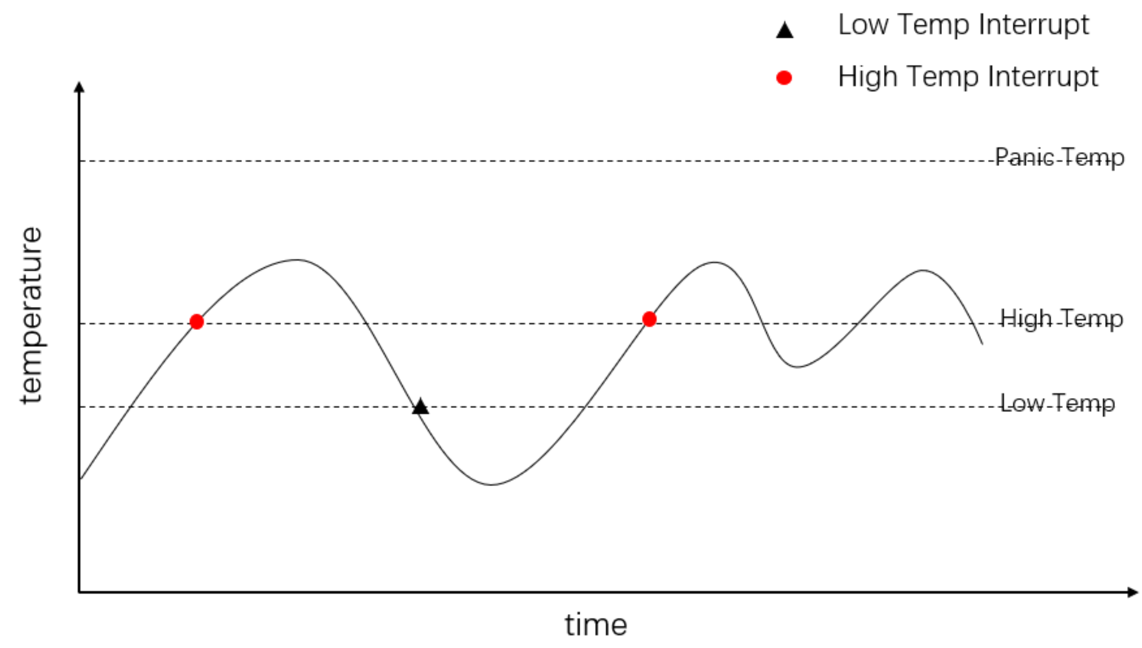


Figure 7. Temperature diagram

The high and low temperature interrupts have a short trigger interval; therefore, the interrupts need to be disabled at the right time to avoid repeated interrupts. In addition, high temperature threshold and low temperature threshold should be set with proper gap so that the interrupt does not occur when the temperature is jittering around the programmed threshold.

When the temperature rises above high temperature threshold, the high temperature interrupt flow is executed. When the temperature falls below low temperature threshold, the low temperature interrupt flow is executed. The high temperature interrupt and low temperature interrupt are triggered in a staggered manner. In other words; after the high temperature interrupt is triggered, only the low temperature interrupt can trigger; and after the low temperature interrupt is triggered, only the high temperature interrupt can trigger.

The figure below shows the temperature interrupt handling flow for the i.MX RT1170, with two flowcharts connected with a bidirectional arrow.

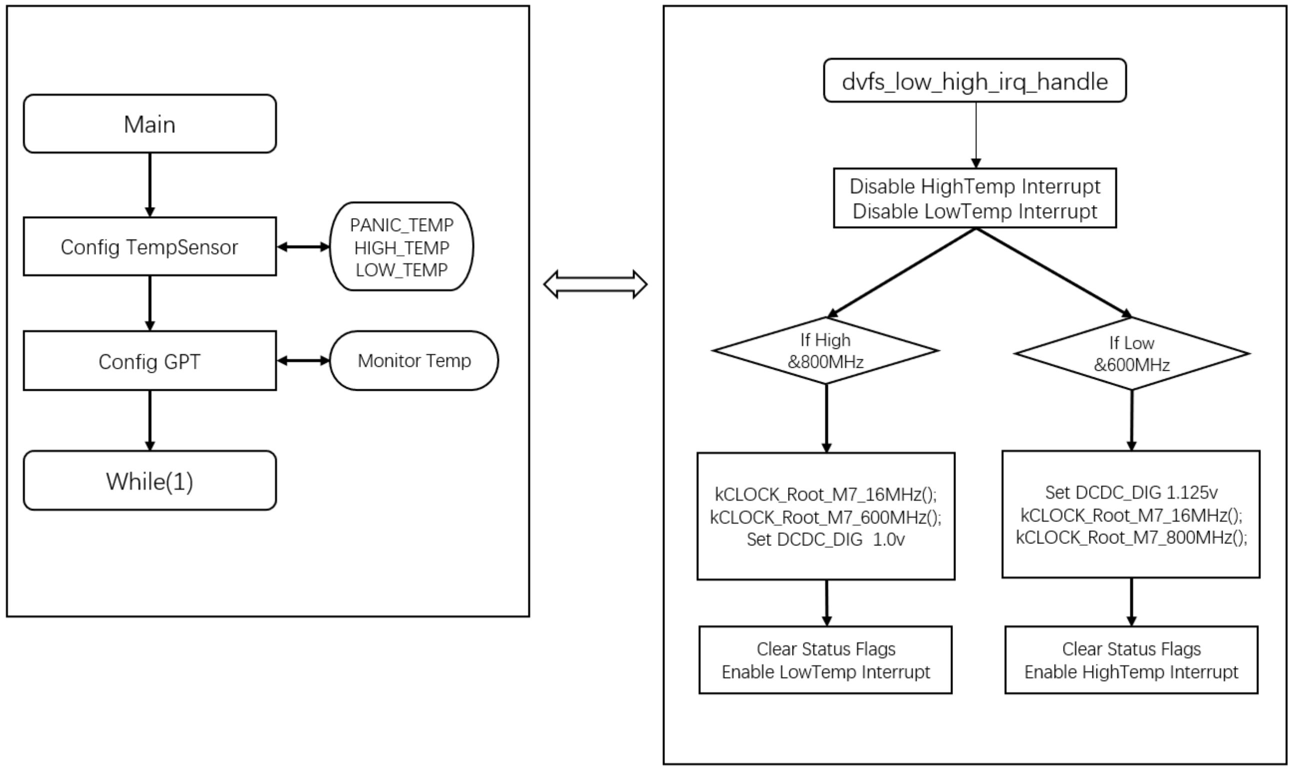


Figure 8. Temperature interrupt handling flow

The left flowchart represents the basic module configuration that must be completed when DVFS is implemented. This includes setting high temperature threshold and low temperature threshold. Configuring GPT to monitor temperature is optional.

The right flowchart represents the interrupt processing flow of DVFS. The DVFS flow is explained below with some guidelines:

- The high temperature interrupt and low temperature interrupt of the i.MX RT1170 temperature sensor share the same interrupt number. When a temperature interrupt is triggered, the two interrupts are disabled first, and the subsequent operations need to be performed according to the actual interrupt source. To determine the actual interrupt source, check the flag bit of the STATUS0 register of temperature sensor and the current operating frequency.
- When setting the high temperature and low temperature interrupt thresholds, the error of the temperature sensor must be considered. It is recommended to keep the high temperature threshold value slightly lower than the theoretical value. For example, panic/high/low temperature threshold can be set to 125/100/95 and the values can be adjusted according to the board thermal design.
- During PLL reconfiguration, switch the relevant clock sources to another PLL until the PLL output is stable
- When the core clock is switched from 800 MHz to 600 MHz, the frequency is reduced first and then the voltage is lowered. When the core clock is switched from 600 MHz to 800 MHz, the voltage is increased first and then the frequency is increased.
- Only the low temperature interrupt is enabled after the high temperature interrupt processing exits, and only the high temperature interrupt is enabled after the low temperature interrupt processing exits

6 References

For more information on the i.MX RT1170, see the following documents:

- *i.MX RT1170 Processor Reference Manual* (IMXRT1170RM)
- *i.MX RT1170 Crossover Processors Data Sheet for Automotive Products* (IMXRT1170AEC)

7 Revision history

The table below summarizes the revisions to this document.

Table 2. Revision history

Revision	Date	Topic cross-reference	Description
Rev. 0	28 May 2021		Initial public release

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Limited warranty and liability — Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

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.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. 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.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 28 May 2021
Document identifier: AN13267

arm