AN13172

Using the PMIC on RT500

Rev. 0 — 09 June 2021 Application Note

Contents

1 Introduction

This document describes the usage of a Power Management IC (PMIC) in the i.MX-RT500 crossover processor family. The PMIC used in the MIMXRT595-EVK is a PCA9420 which is targeted for low-power microcontroller applications. The main features of this PMIC are: 1 MHz I^2C-bus slave interface, linear battery charger for li-ion batteries, two step-down DC-DC buck converters, and two LDOs with programmable output voltages.

Using a PMIC adds flexibility to the RT500 to configure the power supply rails according to the needs of the application. The RT500 can operate in different power modes such as active, sleep, deep-sleep, and deep power-down. Each power mode may require different power settings (for example, enabling/disabling each rail and changing the output voltage of each rail) to achieve a better performance and efficiency.

The RT500 can also operate without an external PMIC by using an internal LDO to supply power to the core logic. However, this configuration has the disadvantage of not being very power efficient due to the nature of the LDO.

2 PMIC summary and features

The following describes the main features of the PCA9420 PMIC:

- Linear battery charger for charging single cell li-ion battery
- Two step-down DC-DC converters:
  - SW1: core buck converter, 0.5 - 1.5 V output, 25 mV/step, and a fixed 1.8 V, up to 250 mA
  - SW2: system buck converter, 1.5 - 2.1/2.7 - 3.3 V output, 25 mV/step, up to 500 mA
- Two LDOs
  - LDO1: always-on LDO, 1.70 - 1.90 V output, 25 mV/step, up to 1 mA
  - LDO2: system LDO, 1.5 - 2.1/2.7 - 3.3 V output, 25 mV/step, up to 250 mA
- 1 MHz I^2C-bus slave interface

This PMIC has four modes where each one can be configured with different settings. Each mode can be configured to enable/disable the four output voltage rails and their voltage level. The RT500 can then simply switch between these modes by using the external pins (MODESEL0/1) or via I^2C.

2.1 PMIC default voltage

Upon initial power-up of the PMIC, Mode 0 is selected. Table 1 describes the default output voltages for the four different modes. These values are defined by the `MODECFG_x` registers from the PMIC and can then be reconfigured via the I^2C.

Table 1. PMIC default voltages

PMIC default output voltages	Mode 0 (default)	Mode 1	Mode 2	Mode 3
SW1_OUT	1.0 V	1.2 V	1.2 V	1.2 V
SW2_OUT	1.8 V	1.8 V	1.8 V	1.8 V
LDO1_OUT	1.8 V	1.8 V	1.8 V	1.8 V
LDO2_OUT	3.3 V	1.8 V	1.8 V	1.8 V

NOTE

The user should be careful that the configured voltages in the PMIC are within the voltage specification for the RT500 power supplies. Refer to *i.MX RT500 Low-Power Crossover Processor* (document IMXRT500EC) for the voltage specification.

3 RT500 power domains

Table 2 describes the different power rails used by the RT500.

Table 2. RT500 power domain

Power rail	Description
VDDCORE	Power supply for core logic. May be supplied from the internal LDO or from an external PMIC.
VDDIO_0/1/2/4	Supply voltage for GPIO pins and PMIC I^2C pins. These supply pins can only be powered from 1.71 V to 1.89 V
VDDIO_3	Supply voltage for GPIO pins. This supply pin can be powered between 1.71 V to 3.6 V.
VDD_AO1V8	1.8 V supply for Always-on features.
VDD1V8	1.8 V supply voltage for on-chip analog functions other than the ADC and comparator including power to the internal LDO.
VDD1V8_1	1.8 V supply voltage for on-chip digital logic.
VDDA_ADC1V8	1.8 V analog supply voltage for ADC and comparator.
VDDA_BIAS	Bias for ADC and comparator. Must be equal to max input voltage.

For information on the power sequence needed by the RT500, refer to the *Power Sequence* section in *i.MX RT500 Low-Power Crossover Processor* (document IMXRT500EC).

4 PMIC connections in MIMXRT595-EVK board

Figure 1 shows a high-level diagram of the connections between the PMIC and the RT500. For more details on the EVK connections, refer to the MIMXRT595-EVK schematic files available for download on https://www.nxp.com.

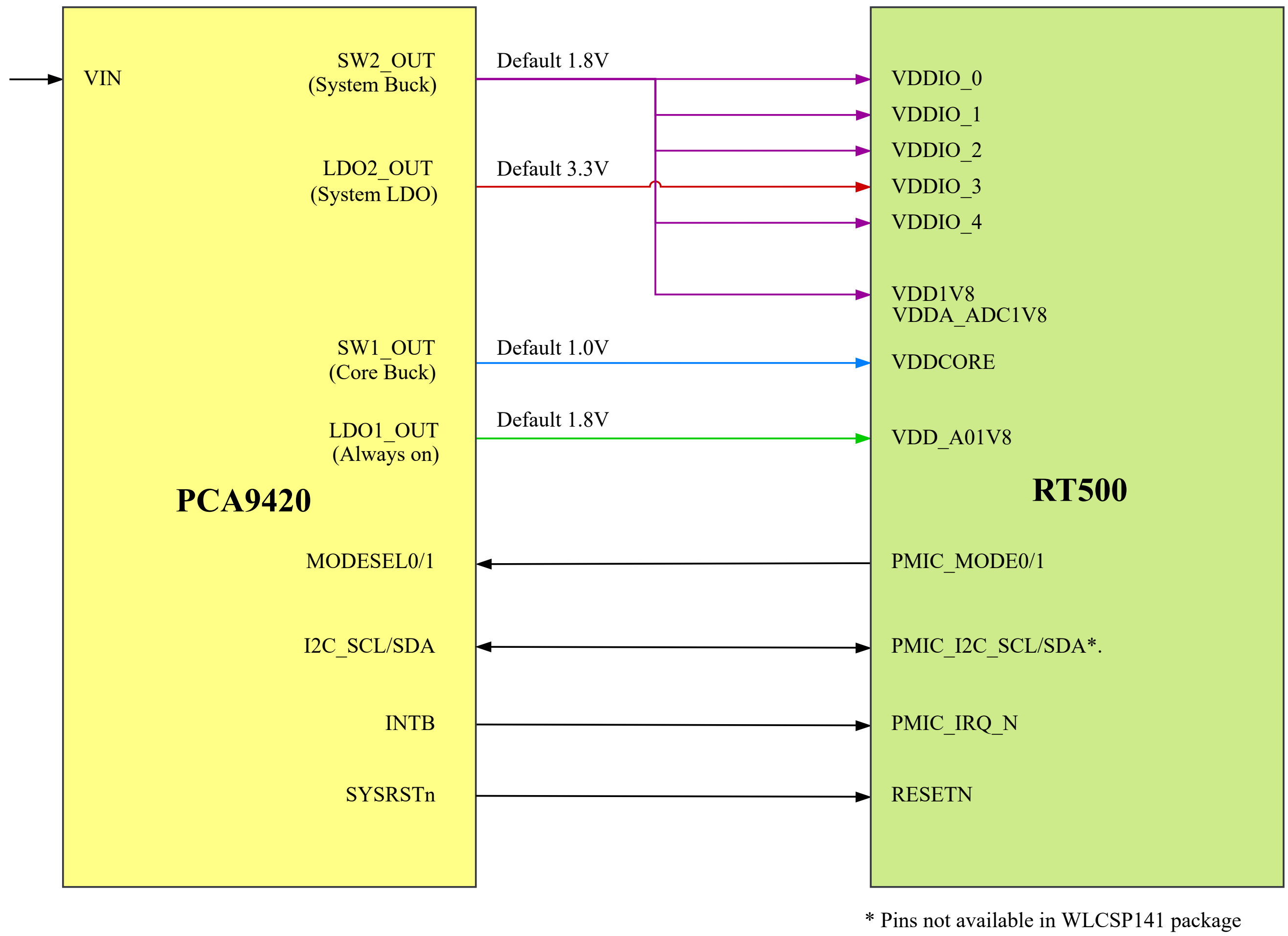

Figure 1. MIMXRT595-EVK PMIC connections

The RT500 has five dedicated pins for communication with an external PMIC, these include an I^2C interface, two output mode pins and an interrupt input pin. The I^2C interface can be used to configure all the different settings that the PMIC offers, it can also be used to change the current mode used by the PMIC, otherwise the dedicated `PMIC_MODE0/1` pins can be used for switching between the modes. The PMIC can also generate interrupts to inform the RT500 about events, such as, voltage threshold warnings, thermal warning, watchdog triggers, power good/bad indicators among other features.

NOTE

For the WLCSP141 package of the RT500, there are no dedicated I^2C pins for the PMIC. However, the user can use another Flexcomm I^2C interface to support PMIC communication over I^2C. Refer to PMIC usage for WLCSP package.

5 MCUXpresso SDK support for the PMIC

The MCUXpresso SDK for the RT500 includes support for the PMIC. The following sections describe how the PMIC usage is leveraged in the SDK.

5.1 PMIC I^2C driver

The MCUXpresso SDK for the RT500 includes a driver (`fsl_pca9420.c/h`) for interfacing the PCA9420 PMIC over I^2C. There is a specific example located in `<SDK>\boards\<board_name>\driver_examples\pca9420\` that demonstrates how to use this driver.

This PMIC driver example displays a menu-like interface in the serial console and allows the user to do the following:

1. Dumping mode settings
2. Switch mode
3. Dump PCA9420 register content
4. Feed watchdog

5.2 PMIC usage for WLCSP package

The WLCSP141 package of the RT500 family does not have dedicated I^2C pins for the PMIC. However, the user can select another Flexcomm I^2C instance to support PMIC communication. This is easily configured in the PMIC I^2C driver of the SDK as it is not tied to any specific Flexcomm I^2C instance. An example is as shown below, where the `BOARD_PMIC_I2C_<XXX>` implementation is board dependent and can be found in the `board.c` file under each example application.

```
BOARD_PMIC_I2C_Init();
PCA9420_GetDefaultConfig(&pca9420Config);
pca9420Config.I2C_SendFunc    = BOARD_PMIC_I2C_Send;
pca9420Config.I2C_ReceiveFunc = BOARD_PMIC_I2C_Receive;
PCA9420_Init(&pca9420Handle, &pca9420Config);
```

The user should also configure the correct pin function for the selected Flexcomm I^2C instance in the *pin_mux.c* file. For this, the MCUXpresso Config Tools can be used to easily configure the pins to the Flexcomm function. For more information on this tool, refer to *MCUXpresso Config Tools User's Guide (IDE)* (document MCUXIDECTUG).

5.3 SDK examples using PMIC

The SDK for the RT500 includes several example applications that configure the PMIC in a certain way.

Here are some examples of how the PMIC is used:

- DSP examples change the `SW1_OUT` (VDDCORE) voltage to run the Cortex-M33 and DSP at the maximum frequencies.
- Examples using different power modes such as the `power_manager` demo and some of the USB examples.

Another example where the PMIC can be useful is with the VDDIO_3 supply, which can support either 1.8 V or 3.3 V. The PMIC can switch between these voltages and add more flexibility for connecting the GPIO pins to the components with different voltage requirements.

5.4 Low-power modes using the PMIC

One of the advantages of using a PMIC with the RT500 is when entering different power modes. This way, for each power mode, you can have different output voltages or completely turn off a voltage rail that is not needed.

The RT500 SDK provides the `power_manager` example located under `<SDK>\boards\<board_name>\demo_apps\` which demonstrates the use of the different power modes available in the RT500 but also leverages the PMIC by configuring the power rails to different settings. Table 3 shows the voltages that each power mode configures for this demo.

To change between the different PMIC modes, the RT500 uses the `PMIC_MODE0/1` pins. The power library API from the SDK handles these pins and they are set when entering the corresponding power mode.

Table 3. Voltage vs power mode in power_manager example

PMIC output (Power rail) vs RT500 power mode	Active mode	Sleep mode	Deep Sleep mode	Deep power-down mode	Full deep-power down mode
`PMIC_MODE0/1`	0b00	0b00	0b01	0b10	0b11
`SW1_OUT`	1.0 V	1.0 V	0.6 V	0 V	0 V

Table continues on the next page...

Table 3. Voltage vs power mode in power_manager example (continued)

PMIC output (Power rail) vs RT500 power mode	Active mode	Sleep mode	Deep Sleep mode	Deep power-down mode	Full deep-power down mode
(VDDCORE)					
SW2_OUT (VDDIO_0/1/2/4, VDD1V8, VDDA_ADC1V8)	1.8 V	1.8 V	1.8 V	1.8 V	0 V
LDO1_OUT (VDD_AO1V8)	1.8 V	1.8 V	1.8 V	1.8 V	1.8 V
LDO2_OUT (VDDIO_3)	3.3 V	3.3 V	3.3 V	3.3 V	0 V

6 Conclusion

Using a PMIC on the RT500 provides great flexibility and allows applications to be more power efficient. The user should evaluate if using an external PMIC brings advantages to their application.

7 References

- *i.MX RT500 Low-Power Crossover MCU Reference Manual* (document IMXRT500RM)
- *i.MX RT500 Low-Power Crossover Processor* (document IMXRT500EC)
- *PCA9420 Product data sheet* (document PCA9420)
- MCUXpresso SDK for RT500

8 Revision history

Table 4 summarizes the changes made to this document since the initial release.

Table 4. Revision history

Revision number	Date	Substantive changes
0	09 June 2021	Initial release

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

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

Date of release: 09 June 2021
Document identifier: AN13172