Freescale Semiconductor, Inc.
Application Note

Document Number: AN5088
Rev. 0, 3/2015

Power Management for Kinetis L Family

How to use Kinetis L family low power mode

1. Introduction

The Kinetis L microcontroller family provides an ultra-low power feature for the power sensitive market. Several low power modes are implemented in this MCU family to meet this requirement. This application note show users details of each power modes and provides user case examples in the SDK power manage demo. Tips are given for using each of the power modes.

The Kinetis software development kit (SDK) provides users with robust peripheral drivers, stacks, middleware, and example applications designed to simplify and accelerate application development on any Kinetis MCU. The Kinetis SDK is complimentary and includes full source code under a permissive open-source license for all hardware abstraction and peripheral driver software.

This application note focuses on the Power Management Controller (PMC), System Mode Controller (SMC), Multipurpose Clock Generator (MCG), and Low Leakage Wakeup Unit (LLWU).

Contents

2. Power modes

The ARM® Cortex-M0+ power modes are Run, Sleep, and Deep Sleep. In the Kinetis L family, this core uses wake from interrupt (WFI) instruction to invoke Sleep and Deep Sleep modes. The Kinetis L MCU's extended power modes and their relationship are presented in this table.

ARM CM0+ Power Modes	Freescale MCU Power Modes	Wakeup Module	If Reset?
RUN	RUN, VLPR	-	-
RUN	CPO	AWIC/NVIC	No
SLEEP	WAIT, VLPW	NVIC	No
DEEP SLEEP	STOP, VLPS	AWIC	No
DEEP SLEEP	PSTOP1	AWIC	No
DEEP SLEEP	PSTOP2	AWIC/NVIC	No
DEEP SLEEP	LLS	LLWU	No
DEEP SLEEP	VLLS0/1/3	LLWU	LLWU reset

Table 1. **Kinetis L family power modes**

NVIC means any interrupt source can wake up an MCU from WAIT/VLPW mode. AWIC means only the AWIC wake up source in the reference manual can wake up the MCU from STOP/VLPS mode. LLWU means only the LLWU wake up source in the reference manual can wake up the MCU from LLS/VLLSx modes. To wake up from VLLSx mode, go through a reset flow and call LLWU reset. For Compute Operation mode (CPO), ARM core is in run mode. Any asynchronous interrupt and ARM core synchronous interrupt can wake up the MCU to run mode.

For Kinetis L family devices, the NMI pin can wake up all power modes, while the reset pin resets MCU power mode into default RUN mode if the reset pin is not filtered by the bus clock.

2.1. Power mode details

Power mode implementations in different Kinetis L devices can vary. The user should look into the related reference manual for details. Each power mode is described below, with details about the mode and the basics of mode entry and exit. For more advanced information, including what would prevent entering low-power mode, consult the reference manual power mode transitions section. The measurement data, frequencies, and other limit data given below are guidelines only. Make sure to use the MCU data sheet for the official values.

2.1.1. Run mode (RUN)

- Selected after any reset (by default, FOPT = 0xFF).
- On-chip voltage regulator is on, full capability.
- Stack pointer (SP), Program Counter (PC), and link register are set.
- ARM Cortex-M0+ processor exits reset and reads the start SP and PC from flash address 0x0 and 0x4.

- Only DMA and Flash clock are enabled by default.
- Reduce power by clearing peripherals clock gating bits in SCGCx registers if not used.

In Run mode, expect typical IDDs from 4 to 6 mA in room temperature (check the MCU device datasheet for more information), depending on the clock frequency and MCU family. Some Kinetis L family devices support boot into VLPR mode. Check Chapter 6 of the reference manual for details.

2.1.2. Very low power run mode (VLPR)

- Selected after any reset (if MCU supports this feature, and FOPT has VLPR boot setting).
- On-chip voltage regulator is in a mode that supplies only enough power to run the MCU at a reduced frequency.
- Core frequency limited up to 4 MHz (Clock divider should set to proper value to meet this).
- Bus/Flash frequency limited up to 800 kHz–1 MHz (Bus clock sharing clock divider with flash).
- Reduce power by clearing clock gating bits in SCGCx registers.
- Flash programming and erasing is not allowed.
- Can enter VLPR in BLPI mode only with the internal IRC (4MHz or 8/2MHz) selected.
- Can enter VLPR in BLPE mode if the external clock is less than 16 MHz.

In VLPR mode, expect typical IDDs from 0.16 to 0.5 mA in room temperature (check the MCU device datasheet for more information). Due to the different MCG implementation, the internal IRC max frequency is different in different MCUs. Some Kinetis L family devices also support boot into VLPR mode. Users should see the reference manual for details.

2.1.3. Wait mode (WAIT)

- NVIC remains sensitive to interrupts.
- Peripherals continue to be clocked.
- Reduce power by clearing peripherals clock gating bits in SCGCx registers if not used.
- On the interrupt, the ARM Cortex-M0+ core exits Sleep mode and resumes Run mode processing.

Expect IDDs to be about RUN IDD minus 2 in room temperature (check the MCU device datasheet for more information).

2.1.4. Very low power wait mode (VLPW)

- Can only enter VLPW from VLPR.
- On-chip voltage regulator stays in a mode that supplies only enough power to run the MCU in a reduced frequency.
- NVIC remains sensitive to interrupts.

- Peripherals continue to be clocked.
- Reduce power by clearing peripherals clock gating bits in SCGCx registers if not used.
- Can optionally keep the external reference clock enabled (16 MHz max).
- On the interrupt, the ARM Cortex-M0+ core exits Sleep mode and resumes VLPR mode processing.

Expect IDDs to be about VLPR IDD minus 0.08 to 0.1 mA in room temperature (check the MCU device datasheet for more information).

2.1.5. Normal stop mode (STOP)

- Asynchronous Wakeup Interrupt Controller (AWIC) is used to wake from interruptions.
- System and peripheral clocks are stopped.
- All SRAM content retained and I/O states held.
- Can enter Stop from any normal run mode.
- MCG module can be configured to leave reference clocks running.
- Can optionally keep the PLL enabled but the output will be gated off—C5[PLLSTEN].
- Can optionally keep the internal reference clock enabled—C1[IREFSTEN].
- Can optionally keep the external reference clock enabled—OSC_CR[EREFSTEN].
- Will exit Stop in the same MCG mode when Stop was entered, except if Stop is entered when in PEE mode and the MCG will be in PBE mode when Stop is exited.

Expect IDDs around 200uA in room temperature (check the MCU device datasheet for more information).

2.1.6. Partial stop mode 1 (PSTOP1)

- Asynchronous Wakeup Interrupt Controller (AWIC) is used to wake from interruptions.
- System and peripheral clocks are stopped.
- All SRAM content retained and I/O states held.
- Can enter PSTOP1 from any run mode.
- MCG and PMC module state is not changed.
- Will exit PSTOP1in the same MCG mode when PSTOP1 was entered.

Expect IDDs around 2mA in room temperature, depending on core speed and peripherals on.

2.1.7. Partial stop mode 2 (PSTOP2)

- AWIC/NVIC are used to wake from interruptions.

- System clock is stopped, peripherals continue to be clocked.
- All SRAM content retained and I/O states held.
- Can enter PSTOP2 from any run mode.
- MCG and PMC module state is not changed.
- Will exit PSTOP2 in the same MCG mode when PSTOP2 was entered.

Expect IDDs around 2mA in room temperature, depending on core speed and peripherals on.

2.1.8. Compute operation mode (CPO)

- AWIC/NVIC is used to wake from interruptions.
- MCU core and SRAM/Flash access are enabled, other peripherals clocks are stopped.
- Can enter CPO from any run mode.
- MCG and PMC module state is not changed.

Expect IDDs is nearly the same as run mode in room temperature.

2.1.9. Very low power stop mode (VLPS)

- NVIC is disabled.
- AWIC is used to wake from interruptions.
- System and peripheral clocks are stopped.
- All SRAM content retained and I/O states held.
- Can enter VLPS from any MCG mode.
- MCG module can be configured to leave reference clocks running.
- Can optionally keep the internal reference clock enabled—C1[IREFSTEN].
- Can optionally keep the external reference clock enabled—OSC_CR[EREFSTEN].
- Will exit VLPS in the same MCG mode when VLPS was entered, except if VLPS is entered when in PEE mode and the MCG will be in PBE mode when VLPS is exited.

Expect IDD around 2 uA in room temperature (check the MCU device datasheet for more information).

2.1.10. Low leakage stop models (LLS)

- NVIC is disabled.
- AWIC is disabled.
- LLWU configured to enable the desired wake-up source.
- System and peripheral clocks are stopped.
- All SRAM content retained and I/O states held.
- Most peripherals are in state retention mode.
- Can enter LLS from any MCG mode.
- Can optionally keep the external reference clock enabled in low range and low-power oscillator mode (32 kHz).
- MCG is static with no clocks active (IREFSTEN and PLLSTEN have no effect).
- Will exit LLS in the same MCG mode when LLS was entered, except if LLS is entered when in PEE mode and the MCG will be in PBE mode when LLS is exited.
- Upon a wake-up event, the WAKEUP bit in the SRS register is set.
- After executing the LLWU interrupt code, execution continues at the next instruction following the LLS mode entry.

Expect IDD around 2 uA in room temperature (check the MCU device datasheet for more information). This power mode is not implemented in all Kinetis MCU L devices. See the reference manual for details.

2.1.11. Very low leakage stop 3 mode (VLLS3)

- NVIC is disabled.
- AWIC is disabled.
- LLWU configured to enable the desired wake-up source.
- System and peripheral clocks are stopped.
- Most modules are disabled.
- All SRAM content retained and I/O states held.
- Can enter VLLS3 from any MCG mode.
- Can optionally keep the external reference clock enabled in low range and low-power oscillator mode (32 kHz).
- MCG is off with no clocks active (IREFSTEN and PLLSTEN have no effect).
- Upon a wake-up event, exit VLLS3 through the reset flow. MCG comes up in FEI mode.

- Upon a wake-up event, the WAKEUP bit in the SRS register is set and the MCU executes the code from the reset vector. Then the LLWU NVIC interrupt is executed when the LLWU interrupt vector is enabled.

Expect IDD around 1 uA in room temperature (check the MCU device datasheet for more information). This power mode is implemented in all latest Kinetis MCU L devices. See the reference manual for details.

2.1.12. Very low leakage stop 1 (VLLS1)

- NVIC is disabled.
- AWIC is disabled.
- LLWU configured to enable the desired wake-up source.
- System and peripheral clocks are stopped.
- Most modules are disabled.
- No SRAM content retained, Register File retained and I/O states held.
- Can enter VLLS1 from any MCG mode.
- Can optionally keep the external reference clock enabled in low range and low-power oscillator mode (32 kHz).
- MCG is off with no clocks active (IREFSTEN and PLLSTEN have no effect).
- Upon a wake-up event, exit VLLS1 through the reset flow. MCG comes up in FEI mode.
- Upon a wake-up event, the WAKEUP bit in the SRS register is set and the MCU executes the code from Reset vector. Then the LLWU NVIC interrupt is executed when the LLWU interrupt vector is enabled.

Expect IDD around 0.5 uA in room temperature (check the MCU device datasheet for more information). This power mode is implemented in all latest Kinetis MCU L device. See the reference manual for details.

2.1.13. Very low leakage stop 0 mode (VLLS0)

- NVIC is disabled.
- AWIC is disabled.
- LLWU configured to enable the desired wake-up source.
- System and peripheral clocks are stopped.
- Power on Protection (POR) optionally enabled.
- No SRAM content retained, Register File retained and I/O states held.
- Can enter VLLS0 from any MCG mode.

- Can optionally turn off the Low power Oscillator (1 KHz) clock.
- Can optionally keep the external reference clock enabled in low range and low-power oscillator mode (32 kHz).
- MCG is off with no clocks active (IREFSTEN and PLLSTEN have no effect).
- Upon a wake-up event, exit VLLS0 through the reset flow. MCG comes up in FEI mode.
- Upon a wake-up event, the WAKEUP bit in the SRS register is set and the MCU executes the code from reset vector. The LLWU NVIC interrupt is executed when the LLWU interrupt vector is enabled.

Expect IDD around 0.3 uA in room temperature (Check the MCU device datasheet for more information). This power mode is implemented in all Kinetis MCU L devices. See the reference manual for details.

2.2. Power mode transition

This section describes power mode transition for the Kinetis L family. To enter VLPx/LLS/VLLSx mode, users must enable mode entry by setting related bits in SMC_PMPROT. This register is write-once, and reset by any non-VLLSx mode recovery reset. Users must first enable all power modes entry in this register before trying to enter these low power modes. Some of the latest Kinetis MCUs allow VLPR boot mode, while setting flash configuration filed 0x40D bit 4 to 0 (0bxxx0xxxx). This diagram displays this in the dashed line. Note that any reset means all reset types of this MCU, so when running VLLSx mode recovery it is also possible to run into VLPR mode if enabled.

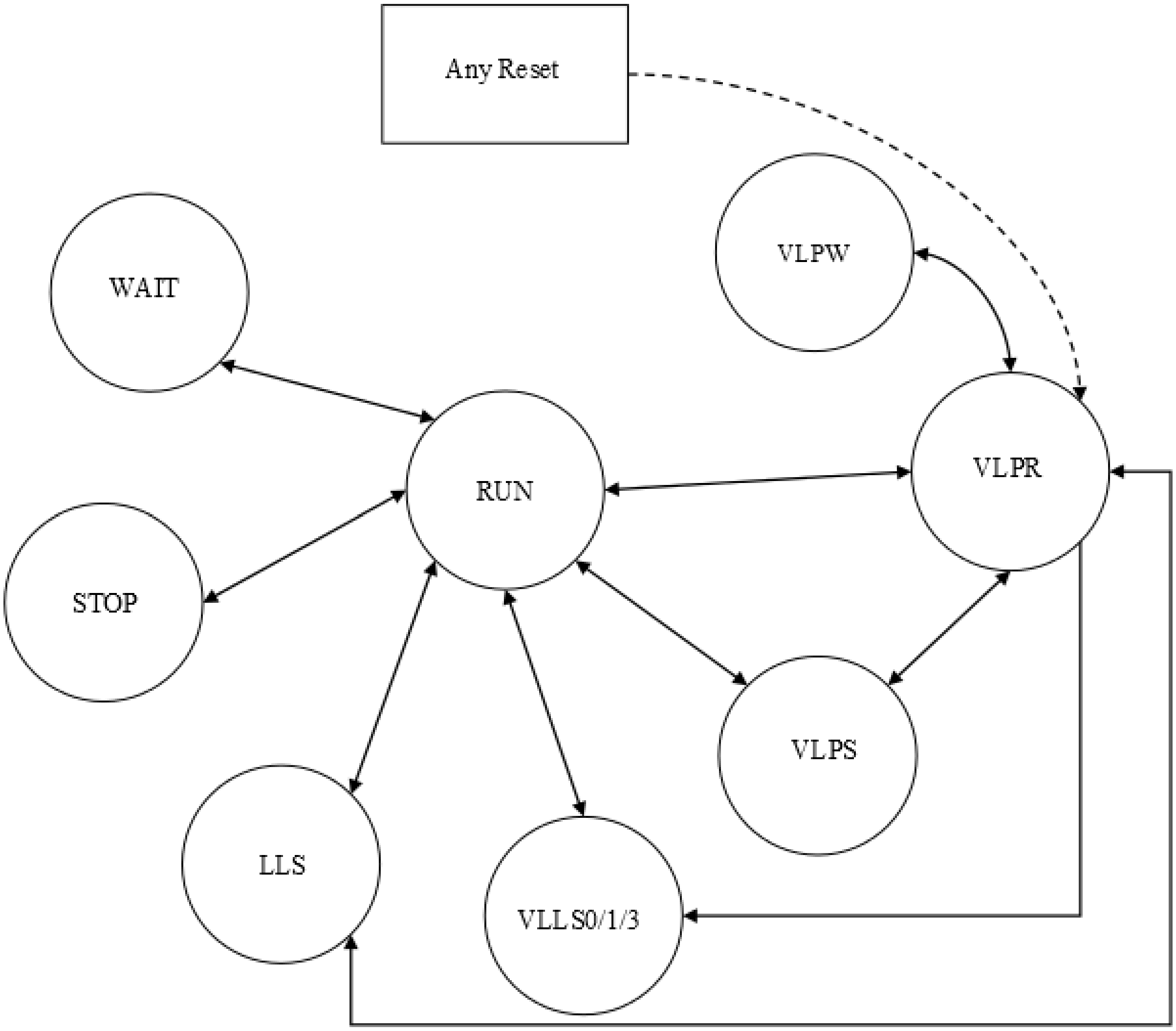

Figure 1. Power mode transition

3. MCG and MCG_Lite

There are two types of MCG used in Kinetis L family. One is FLL/PLL based MCG, while the other is IRC based MCG, calling MCG_Lite. See the differences of MCG and MCG_Lite in this table.

Feature	MCG	MCG_Lite
High frequency source	FLL or PLL-Configurable	HIRC- Fixed 48MHz
Low frequency source	4MHz and 32KHz IRC	8/2 MHz IRC
Auto trim	Yes (4MHz and 32KHz IRC)	No (Trimmed by factory)
External clock monitor	Yes	No
VLPR clock source	4MHz IRC/Ext OSC	8/2 MHz IRC/Ext OSC
STOP/LLS recovery MCG mode	No change, except for PEE(back to PBE mode)	No change, back to the same MCG mode
Default MCG mode after Reset	FEI mode	8MHz IRC

Table 2. **MCG and MCG_Lite**

Users should see the reference manual, where MCG is used in a certain MCU device. The MCG implementation may be different among all devices.

4. Quick start guide

This quick start guide is based on the SDK 1.1 GA power management demo (power_manager_demo). The user needs to initialize and register all power parameter structures and wakeup the source before entering a certain power mode. RTC is used as wakeup source for this demo. This is example code.

Example 1. Initialize parameters and wake up source

```
// Power mode protection initialization
// Must make sure all request power mode entry are allowed
#ifdef SYSTEM_SMC_PMPROT_VALUE
  SMC->PMPROT = SYSTEM_SMC_PMPROT_VALUE;
#endif

//define a power manager user config list for all power mode will use
    power_manager_user_config_t vlpwConfig;
#if FSL_FEATURE_SMC_HAS_STOP_SUBMODE0 & BOARD_SW_HAS_LLWU_PIN
    power_manager_user_config_t vlls0Config;
#endif
    power_manager_user_config_t vlls1Config;
#if FSL_FEATURE_SMC_HAS_STOP_SUBMODE2
    power_manager_user_config_t vlls2Config;
#endif
    power_manager_user_config_t vlls3Config;
#if FSL_FEATURE_SMC_HAS_LOW_LEAKAGE_STOP_MODE
    power_manager_user_config_t llsConfig;
#endif
    power_manager_user_config_t vlpsConfig;
    power_manager_user_config_t waitConfig;
    power_manager_user_config_t stopConfig;
    power_manager_user_config_t runConfig;
#if FSL_FEATURE_SMC_HAS_HIGH_SPEED_RUN_MODE
    power_manager_user_config_t hsrunConfig;
#endif
// Example of constant configuration
// It may save the space in RAM
const power_manager_user_config_t vlprConfig = {
        .mode = kPowerManagerVlpr,
#if FSL_FEATURE_SMC_HAS_LPWUI
        .lowPowerWakeUpOnInterruptOption = true,
        .lowPowerWakeUpOnInterruptValue = kSmcLpwuiEnabled,
#endif
        .sleepOnExitValue = false,
        .sleepOnExitOption = false
};
// Initializes array of pointers to power manager configurations
power_manager_user_config_t const *powerConfigs[] =
{
   &vlls0Config,
   &llsConfig,
&stopConfig,
   &runConfig,
   &vlprConfig
};

// User callback data
user_callback_data_t callbackData0;
// Initializes callback configuration structure for power manager
const power_manager_callback_user_config_t callbackCfg0 = { callback0,
        kPowerManagerCallbackBeforeAfter,
        (power_manager_callback_data_t*) &callbackData0 };

// Initializes array of pointers to power manager callbacks
power_manager_callback_user_config_t const * callbacks[] =
{ &callbackCfg0 };

// Initializes hardware
hardware_init();
// Initializes OS abstraction layer which uses LPTMR HAL layer
OSA_Init();

// Make the current Clock Manager mode configuration 1 (default configuration)
/* Set clock configurations to clock manager. */
CLOCK_SYS_Init(g_defaultClockConfigurations, CLOCK_CONFIG_NUM,
                  &clockCallbackTable, ARRAY_SIZE(clockCallbackTable));

CLOCK_SYS_UpdateConfiguration(cmConfigMode, kClockManagerPolicyForcible);

// Using RTC as interrupt source for this demo
#ifndef FRDM_K22F120M
    // Configure RTC pins
    configure_rtc_pins(BOARD_RTC_FUNC_INSTANCE);
#endif
    // select the 1Hz for RTC_CLKOUT
    CLOCK_SYS_SetRtcOutSrc(kClockRtcoutSrc1Hz);

RTC_DRV_Init(0);

// Set a start date time and start RTC
date.year = 2014U;
date.month = 4U;
date.day = 30U;
date.hour = 14U;
date.minute = 0U;
date.second = 0U;
RTC_DRV_SetDatetime(0, &date);

// Initializes debug UART console
```

```
dbg_uart_init();

// Initializes GPIO driver for LEDs and buttons
GPIO_DRV_Init(switchPins, ledPins);

memset(&callbackData0, 0, sizeof(user_callback_data_t));

// initializes configuration structures

vlpwConfig = vlprConfig;
vlpwConfig.mode = kPowerManagerVlpw;

// VLLS0 mode is supported only by some SOCs.
vlls0Config = vlprConfig;
vlls0Config.mode = kPowerManagerVlls0;

// classic LLS mode retains all ram content too
llsConfig = vlprConfig;
llsConfig.mode = kPowerManagerLls;

stopConfig = vlprConfig;
stopConfig.mode = kPowerManagerStop;

runConfig.mode = kPowerManagerRun;

// initialize power manager driver
POWER_SYS_Init(&powerConfigs,
sizeof(powerConfigs)/sizeof(power_manager_user_config_t *),
    &callbacks,
    sizeof(callbacks)/sizeof(power_manager_callback_user_config_t *));

// Enables LLWU interrupt
INT_SYS_EnableIRQ(LLW_IRQn);
```

4.1. VPLR mode entry and exit example

4.1.1. MCG/MCG_Lite mode change for VLPR/VLPW

For VLPR/VLPW mode entry, the user needs to switch MCG mode into BLPI/BLPE mode, or switch MCG_Lite into 8/2MHz IRC or EXT mode. SDK clock driver CLOCK_SYS_UpdateConfiguration has provided such functionality. The user example code and API is following:

Example 2. MCG mode change example

```
//If apps default CM config mode is not VLPR, but needs to enter VLPR, and real CM config
 //is not VLPR, then we need to update it to VLPR mode here. Otherwise pass through
if ((cmConfigMode != CLOCK_CONFIG_INDEX_FOR_VLPR) &&
(CLOCK_SYS_GetCurrentConfiguration() != CLOCK_CONFIG_INDEX_FOR_VLPR))
{
     CLOCK_SYS_UpdateConfiguration(CLOCK_CONFIG_INDEX_FOR_VLPR, kClockManagerPolicyForcible);
}
```

See the API user manual for more details of how to use these APIs.

4.1.2. VLPR mode entry

SDK driver POWER_SYS_SetMode provide user how to enter certain power mode. Example code is similar to the following:

Example 3. VLPR mode entry example

```
  mode = kDemoVlpr;
//If apps default CM config mode is not VLPR, but needs to enter VLPR, and real CM config
is not VLPR, then we need to update it to VLPR mode here. Otherwise pass through.
 if ((cmConfigMode != CLOCK_CONFIG_INDEX_FOR_VLPR) &&
(CLOCK_SYS_GetCurrentConfiguration() != CLOCK_CONFIG_INDEX_FOR_VLPR))
{
 CLOCK_SYS_UpdateConfiguration(CLOCK_CONFIG_INDEX_FOR_VLPR, kClockManagerPolicyForcible);
}
ret = POWER_SYS_SetMode(mode, kPowerManagerPolicyAgreement);
```

See the API user manual for more details of how to use these APIs.

4.1.3. VLPR mode exit

Kinetis L family device does not support interrupt back to normal run mode, so the user needs to use the power management function to go back to normal run mode.

Example 4. VLPR mode exit example

```
mode = kDemoRun;
ret = POWER_SYS_SetMode(mode, kPowerManagerPolicyAgreement);
// update Clock Mode if user want to go back to normal speed mode
updateClockManagerToRunMode(cmConfigMode);
```

4.2. LLS mode entry and exit example

Set the mode value to kDemoLls. Call POWER_SYS_SetMode to enter LLS mode while RTC is used as LLWU wake up source, which is already configured.

Example 5. LLS mode entry exit example

```
mode = kDemoLls;
ret = POWER_SYS_SetMode(mode, kPowerManagerPolicyAgreement);
// update Clock Mode if user want to go back to normal speed mode
updateClockManagerToRunMode(cmConfigMode);
```

4.3. VLLS0 mode entry and exit example

Set the mode value to kDemoVlls0. Call POWER_SYS_SetMode to enter VLLS0 mode while RTC is used as LLWU wake up source, which is already configured.

Example 6. VLLS0 mode entry exit example

```
mode = kDemoVlls0;
ret = POWER_SYS_SetMode(mode, kPowerManagerPolicyAgreement);
```

5. Lower power mode usage hints

The user should check the following items in the Kinetis L family device. Some of these items can be applied to other Kinetis family MCUs.

1. Use the NMI pin as a switch or wakeup source for VLLSx mode. When using this pin, VLLSx recovery will go through a reset flow. After the MCU is out of this reset, if the NMI pin is still asserted, it will enter NMI interrupt and may access other peripherals registers. Meanwhile, all other peripherals are disabled. This results in a hard fault interrupt, and may cause a WDOG reset. In this case, the user will need to have a proper NMI interrupt routine to handle this issue. For example, when recovering from VLLSx mode, the NMI interrupt routine first disables WDOG, then makes sure the peripheral clock is enabled before accessing the registers.
2. If the NMI pin is used for another function, to have PTA4 to wake up the MCU for example, disable the NMI function by clearing flash configuration filed 0x40D bit 2. Make sure this pin will not drive to 0 the first time programming the Kinetis L MCU. Otherwise, the user will fail to program the flash.
3. Kinetis L MCUs are CMOS circuits. The user must make sure any used GPIO input static voltage is around 0 or VDD. Otherwise, it will cause current leakage, especially when the input is around VDD/2.
4. Keep all unused/unbound pins in a known value, output low or high to get a best low power mode performance if they are not disabled by default. If any pin is floating or has an input voltage other than 0 or VDD, a large current leakage will result.
5. Keep all unnecessary peripherals or clock sources disabled.

6. References

The references listed below have additional information regarding power management for Kinetis family. You can find a particular reference manual, data sheet, or errata report by choosing a device on the Kinetis (www.freescale.com/Kinetis) pages and selecting a specific family for more information. To find latest SDK installer, visit www.freescale.com/kdsdk. To find the application notes below, search for the document number.

- **MCU Reference Manuals:** The reference manuals contain MCU-specific implementation details in the Chip Configuration chapters, and include a detailed description of the Resets and Power Management Features of each MCU.
- **MCU Data Sheet Specifications:** The data sheet includes all of the MCU specifications, including clock rates, low power mode power consumption expectations, and so on.
- **Errata for MCUs:** Device errata identify what functionality and/or specification is not being met due to a problem with the MCU. Most of the issues have workarounds.
- ***Freescale MQX™ Low-Power Management* (AN4447):** Freescale offers MQX, the full-featured and complimentary Real-Time Operating System (RTOS). Starting with version 3.8, MQX integrates a Low-Power Management (LPM) driver to take advantage of the low-power operating modes in MQX applications.

- ***Using Low Power modes on the Kinetis Family* (AN4470):** Includes low-power mode entry demonstration code that can be used across the Kinetis and ColdFire+ devices with mode controller 2.
- *Power Management for Kinetis MCUs* (AN4503): Includes Kinetis MCU power mode architecture introduction, power mode measurement, and power management technology.

7. Revision History

Table 3. **Revision history**

Revision Number	Date	Substantive changes
0	02/2015	Initial Release

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale 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.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale 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 Freescale 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. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Document Number: AN5088
Rev. 0
3/2015