Freescale Semiconductor
Application Note

Document Number: AN4857
Rev. 2, 10/2014

Low-Power Modes on MKW2x Family

Juan Carlos Pacheco Serrano
MCU Wireless Applications Engineering
Guadalajara, Mexico

Contents

1 Introduction

Freescale offers solutions with energy efficient products that can help you achieve the power consumption needed for your application.

This application note will guide you to achieve the available power consumption in some low-power modes available on the MKW2x family.

This application note is based on the TWR-KW24D512 tower board, so some low-power modes will result in different power measurements depending on the hardware connections in place.

2 Low Power for MKW2x Overview

The MKW2x family incorporates a low-power 2.4 GHz radio frequency transceiver and a Kinetis ARM CortexM4 MCU into a single package. When entering into low-power mode, both the MCU and the transceiver must be set to a low power consumption state.

Lowest power in a system is more than just putting the radio and/or the MCU in a low-power mode. The relationship between the functions, the timing between them, and clock management must all be considered. The duty cycle between active operations is also very important as it can impact whether sleep operation or active operation will have the biggest impact over an extended time period.

3 MKW2x Power Modes

The power modes supported by the MKW2x family are divided in the MCU and the transceiver. To achieve the lowest consumption we need to set both modules into low power state.

MCU

- Run
- Wait
- Stop
- VLPR (Very Low Power Run)
- VLPW (Very Low Power Wait)
- VLPS (Very Low Power Stop)
- LLS (Low Leakage Stop)
- VLLS3 (Very Low Leakage Stop3)
- VLLS2 (Very Low Leakage Stop2)
- VLLS1 (Very Low Leakage Stop1)
- VLLS0 (Very Low Leakage Stop0)

Transceiver

- Run
- Idle
- Doze
- Hibernate
- Reset

You can find the details about each mode in the MKW2x Reference Manual.

Figure 1 shows the transitions available for each transceiver's power mode.

Figure 2 shows the transitions available for each MCU power mode.

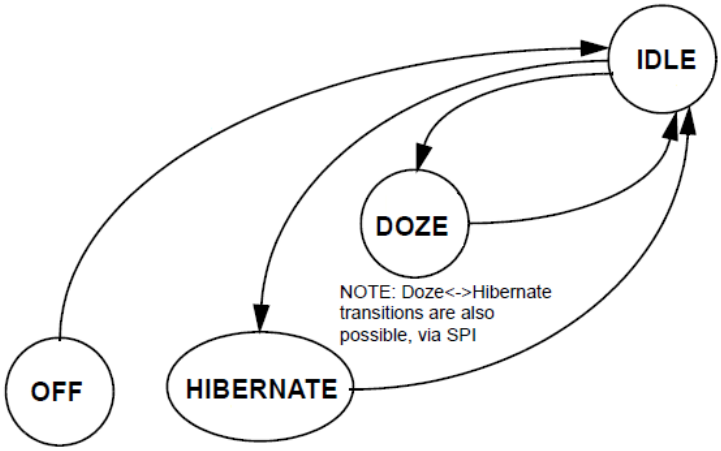

Figure 1. Transceiver's power modes state transitions

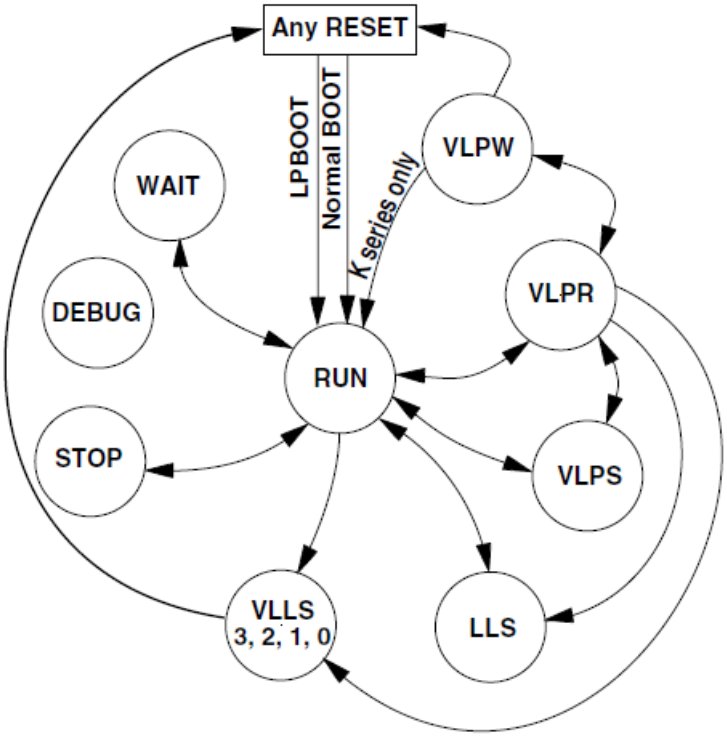

Figure 2. MCU power modes state transitions

4 Wake Up Sources

Each power mode disables different modules. Depending on what power mode was executed, different wake up sources will be available for usage. It's up to the programmer to decide which source is going to

be used depending on the wake up sources supported by the selected power mode. Refer to MKW2x Reference Manual to see the available wake up sources for each low power mode.

5 Low Power Considerations

A low power design should take into consideration the tradeoff between power consumption and performance, and use every possible feature provided by the device to accomplish the best results.

Tips and tricks to consider when designing for low power:

- Set the pins to a known state: Leave unused pines floating and configure them as disabled.
- Select and configure the desired clock mode. The higher the core frequency, the higher the power consumption.
- Higher operating temperature increases power consumption.
- Disable clock to all unused modules in both the transceiver and the MCU.
- Clock monitor output and debug being enabled will increase power consumption and must be avoided.

As previously stated, MKW2x devices consist on an MCU and a transceiver. Therefore you must send the transceiver to a low power state (i.e., Hibernate) before sending the MCU to sleep.

Both software and hardware design must be considered and analyzed to ensure the lowest power on each low power mode.

6 Low power measurements

The primary radio and MCU operating modes are combined so that overall power consumption can then be derived. Radio and MCU power consumption are shown in the following tables.

Table 1. Radio power modes and current consumption

Mode	Current Consumption
Reset	< 100 nA
Hibernate	< 1 uA
Doze	~ 500 uA
Idle	~ 700 uA
Receive	< 19.5 mA
Transmit	< 19 mA

Table 2. MCU power modes and consumption

Mode	Description	Typical Consumption	Max. Consumption	Unit	Notes
Run	Run mode current - All peripheral clocks disabled, code executing from flash				
	• @ 1.8 V	12.98	14.0	mA	1
	• @ 3.0 V	12.93	13.8	mA	
	Run mode current - All peripheral clocks enabled, code executing from flash				
	• @ 1.8 V	17.04	19.3	mA	
	• @ 3.0 V				2
	• @ 25°C	17.01	18.9	mA	
	• @125°C	19.80	21.3	mA	
Wait	Wait mode high frequency current at 3.0 V - all peripheral clocks disabled	7.95	9.5	mA	1
Stop	Stop mode current at 3.0 V				
	• @ -40 to 25°C	320	436	uA	-
	• @ 50°C	360	489		
	• @70°C	410	620		
	• @105°C	610	1100		
VLPS	Very-low-power stop mode current at 3.0 V				
	• @ -40 to 25°C	7.33	24.2	uA	-
	• @ 50°C	14	32		
	• @70°C	28	48		
	• @105°C	110	280		
LLS	Low-leakage stop mode current at 3.0 V				
	• @ -40 to 25°C	3.14	4.8	uA	-
	• @ 50°C	6.48	28.3		
	• @70°C	13.85	44.6		
	• @105°C	55.53	71.3		
VLLS3	Very-low-leakage stop mode 3 current at 3.0 V				
	• @ -40 to 25°C	2.19	3.4	uA	-
	• @ 50°C	4.35	4.35		
	• @70°C	8.92	24.6		
	• @105°C	35.33	45.3		
VLLS2	Very-low-leakage stop mode 2 current at 3.0 V				
	• @ -40 to 25°C	1.77	3.1	uA	-
	• @ 50°C	2.81	13.8		
	• @70°C	5.20	22.3		
	• @105°C	19.88	34.2		
VLLS1	Very-low-leakage stop mode 1 current at 3.0 V				
	• @ -40 to 25°C	1.03	1.8	uA	-
	• @ 50°C	1.92	7.5		
	• @70°C	4.03	15.9		
	• @105°C	17.43	28.7		
VLLS0	Very-low-leakage stop mode 0 current at 3.0 V				
	• @ -40 to 25°C	0.54	1.1	uA	-
	• @ 50°C	1.36	7.5		
	• @70°C	3.39	14.3		
	• @105°C	16.52	24.1		

Notes:

1. 50 MHz core and system clock 25 MHz bus clock and 25 MHz flash clock. MCG configured for FEI mode. All peripheral clocks disabled.
2. 50 MHz core and system clock 25 MHz bus clock and 25 MHz flash clock. MCG configured for FEI mode. All peripheral clocks enabled and peripherals are in active operation.

7 Recovery times from Low Power mode

The mode of operation is controlled by the MCU. The modem may be placed into reset if it is not in use while the MCU is doing another task or while the device is “sleeping”; i.e., the MCU is also powered down. Recovery time for both the modem and the MCU are important to system performance and the recovery times are independent of each other. Radio and MCU recovery times from low-power mode are shown in the following tables.

Table 3. Radio mode definitions and transition times

Mode	Definition	Transition Time to Idle
Off	All modem functions Off, Leakage only. RST asserted. Digital outputs are tristated including IRQ	500 µs
Hibernate	Crystal Reference Oscillator Off. Modem responds to SPI activity.	250 µs
Doze	Crystal reference oscillator ON but CLK_OUT output available only if selected. Digital regulator in Low Power mode.	<1 µs[1]
Reset	Crystal reference oscillator ON, enable CLK_OUT output at 4MHz and 32.787kHz.	376 µs
Receive	Crystal reference oscillator ON. Receiver ON.	144 µs
Transmit	Crystal reference oscillator ON. Transmitter ON.	144 µs

1 At 9 MHz, this transition time will be less than two SPI writes (1.8 µs), which is greater than this transition.

Table 4. MCU recovery time from low power modes

Mode	Time
Wait -> Run	Interrupt Latency
Stop -> Run	2 us + Interrupt Latency
VLPS -> Run	Interrupt Latency
LLS -> Run	2 us + Interrupt Latency
VLLSx -> Run	BOOT + 53 us to 115 us

– The interrupt latency of M4 core based Kinetis devices is 12 cycles.

8 SMAC Low Power demo

MKW2x SMAC Wireless UART demo incorporates low power capabilities to demonstrate the instructions and considerations needed to put the device in a low power state.

The objective of this demo is to get familiar with the steps required to enter and exit from some low-power modes in the MKW2x family of devices.

This demo shows three different low power modes and three different wake up sources. Low-power mode and wake up source are defined before compilation.

The following is required:

- IAR Embedded Workbench ARM v6.50.1 or later
- TWR-KW24D512 tower board
- P&E OpenSDA firmware on the board or IAR JLink for flashing
- Serial communication software such as Hyperterminal from Windows
- Demo project: MKW2x Wireless UART (under SMAC folder, included with BeeKit Wireless Connectivity Toolkit)

8.1 Steps to load the demo

Follow instructions in Chapter 1, Introducing the SMAC from *MKW2x Simple MAC Demo Applications User's Guide (*MKW2xSMACDAUG.pdf) targeting the Wireless UART demo.

8.2 Low-power modes

The Wireless UART demo allows you to exercise the radio's functionality but also has a low power option to demonstrate the steps to follow to get the device in low-power state.

Low-power modes implemented in the demo: VLPS, LLS and VLLS2. To specify the low-power mode to be used, modify the following macro at ApplicationConf.h:

```
#define gDefaultLowPowerMode_c  gWUAppVLLS2Mode_c
```

8.3 Wake up sources

Wake up sources implemented in the demo: GPIO, Low Power Timer and RTC. To specify the wake up source to be used, modify the following macro at ApplicationConf.h:

```
#define gDefaultWakeupSource_c  gWUAppGPIO_c
```

8.4 Run Low Power mode

These steps need to be implemented in order to enable low power.

8.4.1 Set Power Mode Protection register

The PMPROT register provides protection for entry into any low-power run or stop mode. This register can only be written once after a system reset.

Depending on the low power mode that will be selected, this register must be set:

VLPS mode:

```
SMC_PMPROT |= SMC_PMPROT_AVLP_MASK;
```

LLS mode:

```
SMC_PMPROT |= SMC_PMPROT_ALLS_MASK;
```

VLLSx mode:

```
SMC_PMPROT |= SMC_PMPROT_AVLLS_MASK;
```

Refer to MKW2x Reference Manual for more details about PMPROT register.

8.4.2 Set Wake up Source

The LLWU_ME register contains the bits to enable the internal module to be used as a wake up source.

The LLWU_PE1 contains the field to enable and select the edge detect type for the external wake up input pins.

Depending on the wake up source, the corresponding calls must be set:

GPIO wake up:

```
PWRLib_LLWU_WakeupPinEnable( (PWRLib_LLWU_WakeupPin_t)
gLLWU_WakeUp_PIN_Number_c, gPWRLib_LLWU_WakeupPin_AnyEdge_c);
```

LPTMR wake up:

```
NVIC_EnableIRQ(gLPTMR_IRQ_Number_c);

PWRLib_LLWU_WakeupModuleEnable(gPWRLib_LLWU_WakeupModule_LPTMR_c);
```

RTC wake up:

```
PWRLib_RTC_Init();

NVIC_EnableIRQ(gRTC_IRQ_Number_c);

PWRLib_LLWU_WakeupModuleEnable(gPWRLib_LLWU_WakeupModule_RTC_Alarm_c);
```

After the LLWU module is configured, enable the corresponding IRQ.

```
NVIC_EnableIRQ(gLLWU_IRQ_Number_c);
```

Refer to MKW2x Reference Manual for more details about the LLWU module and other available wake up sources.

8.4.3 Backup configuration

Depending on the selected low power mode, there's a possibility to backup your current settings or information in RAM or NVM in order to restore your information or register settings upon wake up.

8.4.4 Stop active tasks and activities

Stop all digital I/O including GPIO, serial interfaces, comparators driving outputs, timers, and so on.

8.4.5 Radio to Low Power

Configure MCG to stop using radio's output clock in case it's using it as reference:

```
/* configure MCG in FLL Engaged Internal (FEI) mode */
    MCG_Pee2Fei();
```

Disable radio's output clock.

```
/* disable transceiver CLK_OUT. */
    MC1324xDrv_Set_CLK_OUT_Freq(gCLK_OUT_FREQ_DISABLE);
```

Put radio in Hibernate state.

```
/* configure Radio in hibernate mode */
    PWRLib_Radio_Enter_Hibernate();
```

At this point, the radio will be in low power state. Depending on the wake up source, you may have to start the LPTMR or the RTC.

8.4.6 MCU to Low Power

The radio should already be in low power state. In case you are using a low power timer or an RTC, they have to be configured before activating the low power mode.

LPTMR:

```
PWRLib_LPTMR_ClockStart(cPWR_LPTMRTickTime, LPTMR_Duration);
```

RTC:

```
PWRLib_RTC_ClockStart(RTC_Duration);
```

Once a wake up source is in place and ready, you can proceed with the low power sequence for the MCU:

VLPS mode:

```
PWRLib_MCU_Enter_VLPS();
```

LLS mode:

```
PWRLib_MCU_Enter_LLS();
```

VLLSx mode:

```
PWRLib_MCU_Enter_VLLS();
```

This instruction will put the device in low power state, no further instructions will be executed until the device is awaken.

Refer to MKW2x Reference Manual for more details about these procedures and other available low power modes.

8.4.7 Return from Low Power

The normal exit procedure is to enable and detect an event that initiates a mode change.

Once the MCU detects a wake up trigger, depending on the low power, it will or will not execute the next instruction.

VLPS and LLS exit to RUN mode, the instruction followed by the low power function will be executed. If the settings and configurations were stored, they should be restored on wake up. Radio has to return from low-power state (hibernate) and the proper clock settings must take place.

If the wake up source was a timer, it has to be stopped upon wake up.

LPTMR:

```
PWRLib_LPTMR_ClockStop();
```

RTC:

```
PWRLib_RTC_ClockStop();
```

In the demo, the MCU uses Radio's output clock signal as reference, so the corresponding transitions must take place:

```
/* wake up and configure Radio in autodoze mode */
 PWRLib_Radio_Enter_AutoDoze();
/* Restore Radio's clock for input*/
 MC1324xDrv_Set_CLK_OUT_Freq(gCLK_OUT_FREQ_4_MHz);
/* PEE @ 48MHz: Restore radio's clock as reference*/
 gMCG_coreClkMHz = MCG_PLLInit();
```

VLLSx exits to RUN mode through a reset flow, all MCU and Radio initialization must take place again.

Refer to application note **AN4503.pdf - Power Mode Exit Transitions** for more information about the wake up transitions depending on what low power mode you are returning from.

8.4.8 Restore configuration

Once the MCU is in RUN mode and the radio is in idle/autoDoze, you can proceed with the restore of the settings that were previously backed up.

8.5 Demo low power measurements

Low power consumption is platform design dependent. The measurements shown in this table are obtained on a TWR-KW24D512.

Table 5. Wireless UART demo - MCU Low Power Measurements

Low Power mode / Wake up Source	VLPS	LLS	VLLS2
GPIO	19.7 uA	17.8 uA	16.6 uA
LPTMR	19.7 uA	17.8 uA	16.6 uA
RTC	20.3 uA	18.6 uA	17.3 uA

Refer to *MKW2x Simple MAC Demo Applications User's Guide*, 'Section 3.7 - Entering Low Power module' to follow specific instructions on how to get and measure the Low Power modes explained in this Application Note.

9 References

The references listed below have additional information regarding power management for Kinetis devices. You can find documentation online (www.freescale.com\Kinetis).

- **MKW2xSMACDAUG.pdf**, *MKW2x Simple MAC Demo Applications User's Guide* - Includes step by step instructions on how to load SMAC projects into a MKW2x device.
- **MKW2xDxxxRM.pdf,** *MKW2x Reference Manual* -The reference manual contains MCU-specific implementation details in the Chip Configuration chapters. Refer to Resets, Power Management and System Considerations.
- **AN4503.pdf,** *Power Management for Kinetis and ColdFire+ MCUs.* Describes details to take into consideration when using low power in Kinetis and Coldfire MCUs.
- **AN4470.pdf,** *Using Low Power modes on Kinetis Family.* Includes low-power mode entry demo code (MCU-only) used across the Kinetis and Coldfire+ devices that can be used as reference.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number: AN4857
Rev. 2
10/2014