

M480 Power Management

Application Note for 32-bit NuMicro® Family

Document Information

Abstract	Introduce power modes including configuration, wake-up method and sample code.
Apply to	NuMicro® M480 series

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

*Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design.
Nuvoton assumes no responsibility for errors or omissions.*

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

Table of Contents

1	INTRODUCTION	3
2	POWER MODES	4
2.1	Normal Mode	5
2.2	Idle Mode	6
2.3	Power-down Mode.....	8
2.3.1	Normal Power-down Mode (NPD)	10
2.3.2	Low Leakage Power-down Mode (LLPD).....	10
2.3.3	Fast Wake-up Power-down Mode (FWPD).....	11
2.3.4	Standby Power-down Mode 0 (SPD0)	11
2.3.5	Standby Power-down Mode 1 (SPD1)	12
2.3.6	Deep Power-down Mode (DPD)	13
3	SAMPLE CODE.....	15
3.1	Normal Mode	15
3.2	Idle Mode	16
3.3	Power-down Mode.....	17
3.3.1	Normal Power-down Mode (NPD)	19
3.3.2	Low Leakage Power-down Mode (LLPD).....	19
3.3.3	Fast Wake-up Power-down Mode (FWPD).....	20
3.3.4	Standby Power-down Mode 0 (SPD0)	21
3.3.5	Standby Power-down Mode 1 (SPD1)	22
3.3.6	Deep Power-down Mode (DPD)	24
4	CONCLUSION	25

1 Introduction

The NuMicro® M480 series microcontroller provides some power modes with different power consumption level and wake-up time. In the following chapters, Chapter 2 introduces different power modes. Chapter 3 demonstrates sample code of power mode configuration and wake-up method. The last Chapter 4 lists the power modes comparisons for user reference to design their low power and efficient application.

2 Power Modes

The NuMicro® M480 power modes include Normal mode, Idle mode and Power-down mode. The system starts up in Normal mode. In Idle mode, only CPU clock is disabled while other peripherals work normally. If system is waiting for an interrupt only, you can set system in Idle mode to save power and wake-up quickly. If system does not need to work for a long time, you can set system in Power-down mode. The M480 provides some different Power-down modes with different power consumption. But the lower power consumption mode needs more wake-up time and causes more data lost. This section introduces these power modes, configuration and wake-up method.

The M480 series includes the following power modes:

- Normal mode
System startup power mode supplies power to all system, peripherals and memories.
- Idle mode
Only CPU clock is disabled. Other peripherals keep working normally. The fastest power mode returns to Normal mode.
- Power-down mode
Besides CPU clock is disabled, more clock sources and peripherals are disabled to save power. Wake-up time is longer because after system wake-up, it needs to wait clock stable and initialization. Some power-down modes even disable memories power that causes data lost. The difference between Standby Power-down Mode 0 (SPD0) and Standby Power-down Mode 1 (SPD1) is whether the system SRAM Bank0 is powered off. Entering SPD0 mode will not turn off the power supply to system SRAM Bank0, so the data will be retained. SPD1 will shut off the power supply to system SRAM Bank0 and the data will be lost. There are six Power-down modes:
 - Normal Power-down Mode (NPD)
 - Low Leakage Power Power-down Mode (LLPD)
 - Fast Wake-up Power-down Mode (FWPD)
 - Standby Power-down Mode 0 (SPD0)
 - Standby Power-down Mode 1 (SPD1)
 - Deep Power-down Mode (DPD)

Figure 2-1 shows M480 power mode transition. The system is powered on in Normal mode. In Normal mode, you can select different voltage regulator output voltage level (Power Level 0 or Power Level 1). System can enter Idle mode, wait interrupt to wake up and return to Normal mode at any voltage level. If system enters Standby Power-down mode 0/1 (SPD0/1)

or Deep Power-down mode (DPD), it will reset and return to default voltage level Power Level 0.

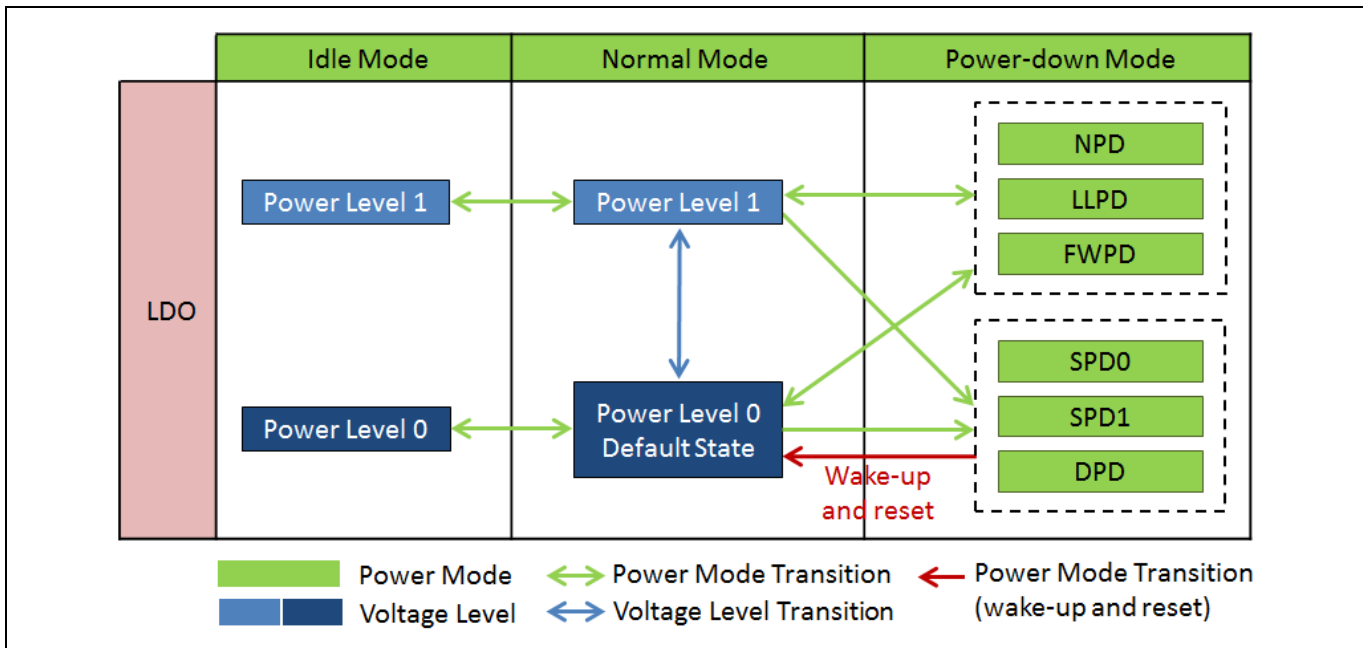


Figure 2-1 M480 Power Mode Transition

2.1 Normal Mode

The system starts up in Normal mode. All clock sources and peripherals can be enabled or disabled by user in register CLK_PWRCRL (System Power-down Control Register), CLK_AHBCLK (AHB Devices Clock Enable Control Register), CLK_APBCLK0 (APB Devices Clock Enable Control Register 0) and CLK_APBCLK1 (APB Devices Clock Enable Control Register 1). You can disable unused clock or peripheral to save power.

Voltage Level

The voltage level can be selected as Power Level 0 and Power Level 1 in Normal mode. The default system voltage level is Power Level 0. In M480, Power Level 0 indicates voltage level is 1.26V and CPU maximum operation frequency is 192MHz. Power level 1 indicates voltage level is 1.2V and CPU maximum operation frequency is 160MHz.

To set voltage level as Power Level 0, set Power Level Selection, PLSEL(SYS_PLCTL[1:0]) = 0.

```
/* Set power level to Power Level 0 */
SYS->PLCTL = (SYS->PLCTL & (~SYS_PLCTL_PLSEL_Msk)) | SYS_PLCTL_PLSEL_PL0;
```

Or you can call function SYS_SetPowerLevel() to set voltage level as Power Level 0.

```
/* Set power level to Power Level 0 */
SYS_SetPowerLevel(SYS_PLCTL_PLSEL_PL0);
```

To set voltage level as Power Level 1, set Power Level Selection, $PLSEL(SYS_PLCTL[1:0]) = 1$.

```
/* Set power level to Power Level 1 */
SYS->PLCTL = (SYS->PLCTL & (~SYS_PLCTL_PLSEL_Msk)) | SYS_PLCTL_PLSEL_PL1;
```

Or you can call function `SYS_SetPowerLevel()` to set voltage level as Power Level 1.

```
/* Set power level to Power Level 1 */
SYS_SetPowerLevel(SYS_PLCTL_PLSEL_PL1);
```

Figure 2-2 shows any voltage level (Power Level 0 or Power Level 1) can switch between each other in Normal mode.

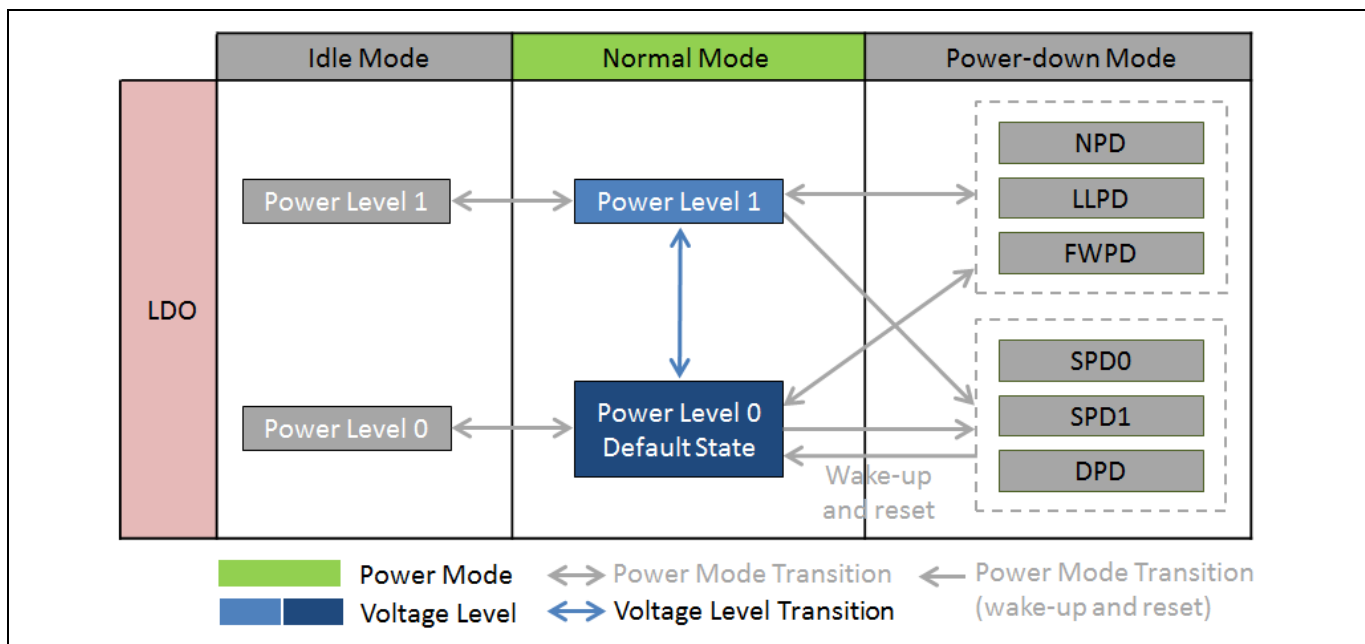


Figure 2-2 Voltage Level Transition in Normal Mode

2.2 Idle Mode

If system is waiting for an interrupt only, you can set system in Idle mode. In Idle mode, only CPU clock is disabled, other peripherals work normally. System waits interrupt to wake up, return to Normal mode and keeps working. All interrupts can wake-up system from Idle mode.

Figure 2-3 shows the transition between Normal mode and Idle mode can operate in any voltage level (Power Level 0 or Power Level 1).

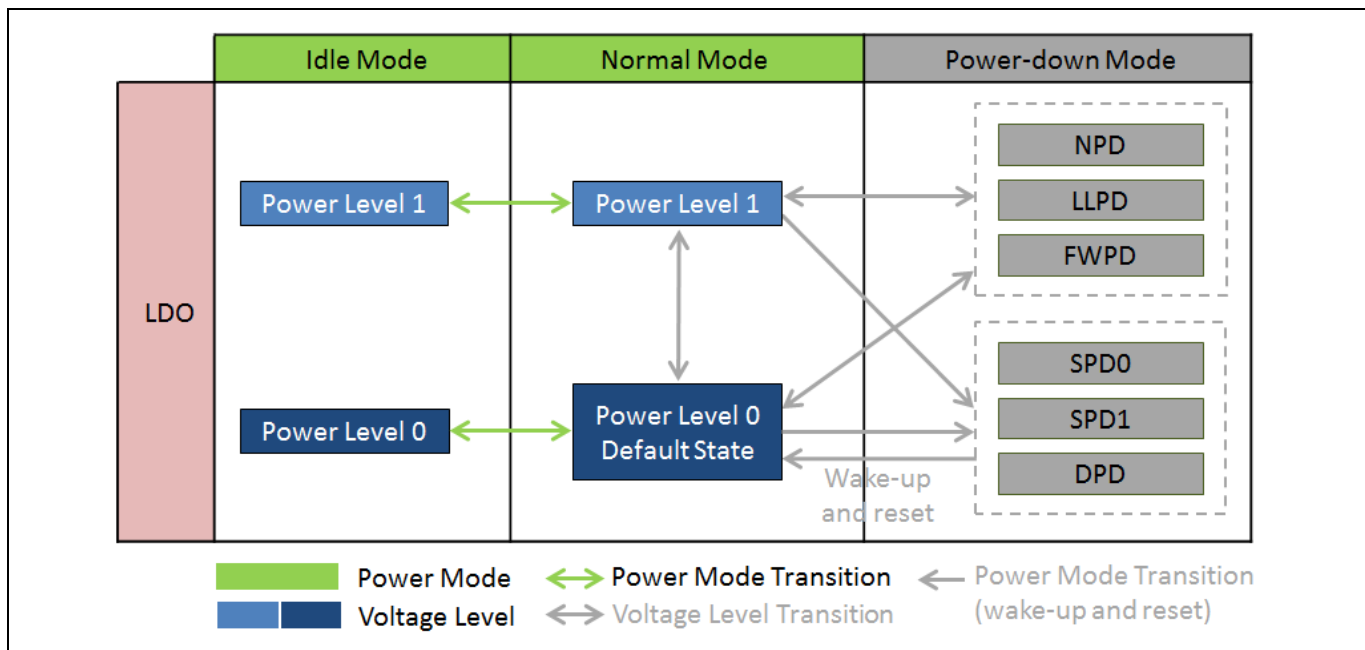


Figure 2-3 Normal Mode and Idle Mode Transition

1. Idle mode configuration:

Set Processor Low Power Mode Selection to select the system's low power mode as Sleep Mode, SLEEPDEEP (SCR[2]) = 0. Set the System Power-down Enable Bit to set the system to enter Idle mode after executing the WFI instruction, PDEN (CLK_PWRCTL [7]) = 0. Finally, when the WFI instruction is executed, the program will stop and system enters Idle mode.

Set Processor Low Power Mode Selection to select system low power mode is sleep mode, SLEEPDEEP (SCR[2]) = 0. Set System Power-down Enable Bit to set system enters Idle mode after executing WFI instruction. After executing WFI instruction, CPU will stop and system enters Idle mode.

```
/* Set idle mode */
void CLK_Idle(void)
{
    /* Set the processor uses sleep as its low power mode */
    SCB->SCR &= ~SCB_SCR_SLEEPDEEP_Msk;

    /* Set chip in Idle mode because of WFI command */
    CLK->PWRCTL &= ~CLK_PWRCTL_PDEN_Msk;

    /* Chip enter Idle mode after CPU run WFI instruction */
    __WFI();
}
```

You can also call function CLK_Idle() and system will enter Idle mode.

```
/* Set Idle mode */
```

```
CLK_Idle();
```

2. Idle mode wake-up source:

All interrupts can wake-up system from Idle mode.

2.3 Power-down Mode

You can set system into Power-down mode when system does not need to work for a long time. NuMicro® M480 provides some power modes with different power consumption level. The less power consumption mode needs more wake-up time and causes more data lost. Figure 2-4 shows power mode transition between Normal mode and Power-down mode. If system enters Standby Power-down mode 0/1 (SPD0/1) or Deep Power-down mode (DPD), it will reset and return to default voltage level Power Level 0

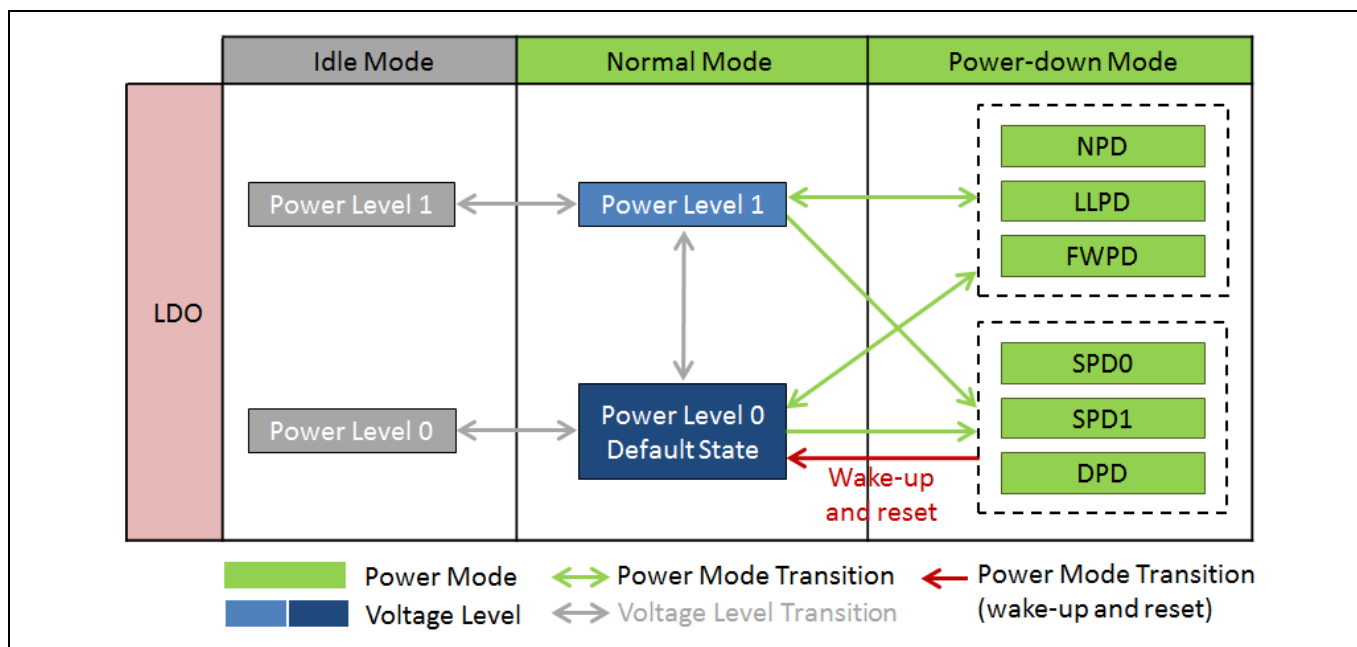


Figure 2-4 Normal Mode and Power-down Mode Transition

Power-down Mode Configuration

Set Power-down Mode Selection, PDMSEL (CLK_PMUCTL[2:0]) to select which Power-down mode. Set Processor Low Power Mode Selection to select system low power mode is Deep sleep mode, SLEEPDEEP (SCR[2]) = 1. Set System Enter Power-down Enable Bit to set system enters Power-down mode after executing WFI instruction. After executing WFI instruction CPU will stop and system enters Power-down mode. Table 2-1 lists different Power-down mode configuration.

Power-down Mode	PDMSEL	SLEEPDEEP	PDEN	Run WFI Instruction
-----------------	--------	-----------	------	---------------------

	(CLK_PMUCTL[2:0])	(SCR[2])	(CLK_PWRCTL [7])	
NPD	0	1	1	Yes
LLPD	1			
FWRPD	2			
SPD0	4			
SPD1	5			
DPD	6			

Table 2-1 Different Power-down Mode Configuration

In the sample code, set Power-down Mode Selection, PDMSEL (CLK_PMUCTL[2:0]), to select a Power-down mode.

```
/* Select Power-down mode */
void CLK_SetPowerDownMode(uint32_t u32PDMMode)
{
    CLK->PMUCTL = (CLK->PMUCTL & (~CLK_PMUCTL_PDMSEL_Msk)) | (u32PDMMode);
}
```

Set SLEEPDEEP (SCR[2]) = 1 to set system low power mode is deep sleep mode. Set PDEN (CLK_PWRCTL [7]) = 1, System Enter Power-down Enable Bit. After executing WFI instruction, CPU will stop and enter Power-down mode.

```
/* Set Power-down mode */
void CLK_PowerDown(void)
{
    /* Set the processor uses deep sleep as its low power mode */
    SCB->SCR |= SCB_SCR_SLEEPDEEP_Msk;

    /* Set system Power-down enabled */
    CLK->PWRCTL |= CLK_PWRCTL_PDEN_Msk;

    /* Chip enter Power-down mode after CPU run WFI instruction */
    __WFI();
}
```

2.3.1 Normal Power-down Mode (NPD)

In Normal Power-down mode (NPD), LDO enters low power mode. CPU clock is stopped. All clock source will be disabled except LXT and LIRC. LXT and LIRC can be controlled by CLK_PWRCTL register. If they are enabled and peripheral clock source are selected as LXT or LIRC, the peripheral can keep working in Power-down mode. System waits wake-up source occurred and returns to Normal mode. After wake-up, system waits for LDO recovery, and clock source is enabled again and stable. The Normal Power-down mode (NPD) wake-up sources include EINT, GPIO, UART, USBD, USBH, OTG, CAN, BOD, WDT, SDH, Timer, I²C, USCI, RTC and ACMP. They need to be configured interrupt in the peripheral registers.

1. Normal Power-down Mode (NPD) configuration:

Call function CLK_SetPowerDownMode() to set Normal Power-down mode (NPD) and call function CLK_PowerDown() then system will enter Normal Power-down mode (NPD).

```
/* Set Power-down mode (NPD) */
CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_PD);

/* Enter Power-down mode */
CLK_PowerDown();
```

2. Normal Power-down Mode (NPD) wake-up source:

Normal Power-down mode (NPD) wake-up sources are EINT, GPIO, UART, USBD, USBH, OTG, CAN, BOD, WDT, SDH, Timer, I²C, USCI, RTC and ACMP.

2.3.2 Low Leakage Power-down Mode (LLPD)

In Low Leakage Power-down mode (LLPD), LDO voltage drops down from current working voltage to 0.9V to save power. After system wake-up from Low Leakage Power-down mode (LLPD), besides waiting for LDO recovery and clock source stable, system also needs to wait for LDO voltage rising to the original working voltage (Power Level 0 or Power Level 1). The Low Leakage Power-down mode (LLPD) wake-up sources are the same as Normal Power-down mode (NPD).

1. Low Leakage Power-down mode (LLPD) configuration:

Call function CLK_SetPowerDownMode() to set Low Leakage Power-down mode (LLPD) and call function CLK_PowerDown() then system will enter Low Leakage Power-down mode (LLPD).

```
/* Set Low Leakage Power-down mode (LLPD) */
CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_LLPD);
```

```
/* Enter Power-down mode */  
CLK_PowerDown();
```

2. Low Leakage Power-down mode (LLPD) wake-up source:

Low Leakage Power-down mode (LLPD) wake-up sources are EINT, GPIO, UART, USB, USBH, OTG, CAN, BOD, WDT, SDH, Timer, I²C, USCI, RTC and ACMP.

2.3.3 Fast Wake-up Power-down Mode (FWPD)

In Fast Wake-up Power-down mode (FWPD), LDO is not in lower power mode so system does not need to wait LDO recovery after wake-up. The wake-up time is faster than Normal Power-down mode (NPD) but it consumes more power, too. Fast Wake-up Power-down mode (FWPD) wake-up sources are the same as Normal Power-down mode (NPD).

1. Fast Wake-up Power-down Mode (FWPD) Configuration:

Call function CLK_SetPowerDownMode() to set Fast Wake-up Power-down mode (FWPD) and call function CLK_PowerDown() then system will enter Fast Wake-up Power-down mode (FWPD).

```
/* Set Fast Wake-up Power-down mode (FWPD) */  
CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_FWPD);  
  
/* Enter Power-down mode */  
CLK_PowerDown();
```

2. Fast Wake-up Power-down Mode (FWPD) wake-up source:

Fast Wake-up Power-down mode (FWPD) wake-up sources are EINT, GPIO, UART, USB, USBH, OTG, CAN, BOD, WDT, SDH, Timer, I²C, USCI, RTC and ACMP.

2.3.4 Standby Power-down Mode 0 (SPD0)

In Standby Power-down mode 0 (SPD0), all power supply is disabled except SPD0 control logic to control SPD Power-down mode wake-up and system SRAM bank0 to retain data. After wake-up from Standby Power-down mode 0 (SPD0), system resets and executes code from the beginning again. All peripheral configurations return to default value. All SRAM data will be lost except system SRAM bank0 data can be retained.

After wake-up from Standby Power-down mode 0 (SPD0), GPIO will keep their states before entering Standby Power-down mode 0 (SPD0). They cannot be controlled by GPIO register or peripherals after wake-up. To control GPIO, you have to write 1 to register CLK_IOPDCTL (GPIO Standby Power-down Control Register) to release this GPIO hold state function. If the

GPIO hold state function is not released, GPIO cannot be controlled by any GPIO or peripheral registers, for example: UART cannot print message and ICE cannot download code. Remember to release GPIO hold state function after system wake-up from SPD Standby Power-down mode 0 (SPD0) by writing 1 to CLK_IOPDCTL register. Then you can control GPIO normally through GPIO or peripherals. Standby Power-down mode 0 (SPD0) wake-up sources includes GPIO (PA, PB, PC and PD), BOD, LVR, ACMP, Wake-up Timer and RTC.

Set register CLK_IOPDCTL (GPIO Standby Power-down Control Register) to release GPIO hold state function after wake-up from Standby Power-down mode 0 (SPD0).

```
/* Release I/O hold status after wake-up from Standby Power-down Mode 0 (SPD0) */
CLK->IOPDCTL = CLK_IOPDCTL_IOHR_Msk;
```

1. Standby Power-down Mode 0 (SPD0) configuration:

Call function CLK_SetPowerDownMode() to set Standby Power-down mode 0 (SPD0) and call function CLK_PowerDown() then system will enter Standby Power-down mode 0 (SPD0).

```
/* Set Standby Power-down Mode (SPD0) */
CLK_SetPowerDownMode(CLK_PMUCTL_PDMSSEL_SPD0);

/* Enter to Power-down Mode */
CLK_PowerDown();
```

2. Standby Power-down Mode 0 (SPD0) wake-up source:

The Standby Power-down mode 0 (SPD0) wake-up sources includes GPIO (PA, PB, PC and PD), BOD, LVR, ACMP, Wake-up Timer and RTC.

3. Standby Power-down Mode 0 (SPD0) SRAM Bank0 data retention:

System SRAM Bank0 data will be retained. Peripheral configuration and peripheral SRAM data will be lost.

2.3.5 Standby Power-down Mode 1 (SPD1)

In Standby Power-down mode 1 (SPD1), all power supply is disabled except SPD1 control logic to control SPD Power-down mode wake-up. After wake-up from Standby Power-down mode 1 (SPD1), system resets and executes code from the beginning again. All peripheral configurations return to default value. System SRAM Bank0 data, peripheral configuration and peripheral SRAM data will all be lost.

After wake-up from Standby Power-down mode 1 (SPD1), GPIO will keep their states before entering Standby Power-down mode 1 (SPD1). They cannot be controlled by GPIO register or peripherals after wake-up. To control GPIO, you have to write 1 to register CLK_IOPDCTL

(GPIO Standby Power-down Control Register) to release this GPIO hold state function. If the GPIO hold state function is not released, GPIO cannot be controlled by any GPIO or peripheral registers, including: UART cannot print message and ICE cannot download code. Remember to release GPIO hold state function after system wake-up from SPD Standby Power-down mode 1 (SPD1) by writing 1 to CLK_IOPDCTL register. Then you can control GPIO normally through GPIO or peripherals. Standby Power-down mode 1 (SPD1) wake-up sources includes GPIO (PA, PB, PC and PD), BOD, LVR, ACMP, Wake-up Timer and RTC.

Set register CLK_IOPDCTL (GPIO Standby Power-down Control Register) to release GPIO hold state function after wake-up from Standby Power-down mode 1 (SPD1).

```
/* Release I/O hold status after wake-up from Standby Power-down Mode 1 (SPD1) */
CLK->IOPDCTL = CLK_IOPDCTL_IOHR_Msk;
```

1. Standby Power-down Mode 1 (SPD1) configuration:

Call function CLK_SetPowerDownMode() to set Standby Power-down mode 0 (SPD1) and call function CLK_PowerDown() then system will enter Standby Power-down mode 0 (SPD1).

```
/* Set Standby Power-down Mode (SPD1) */
CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_SPD1);

/* Enter to Power-down Mode */
CLK_PowerDown();
```

2. Standby Power-down Mode 1 (SPD1) wake-up source:

The Standby Power-down mode 0 (SPD1) wake-up sources includes GPIO (PA, PB, PC and PD), BOD, LVR, ACMP, Wake-up Timer and RTC.

3. Standby Power-down Mode (SPD1) SRAM Bank0 data retention:

System SRAM Bank0 data, peripheral configuration and peripheral SRAM data will all be lost.

2.3.6 Deep Power-down Mode (DPD)

In Deep Power-down mode (DPD), CPU will be stopped. All power supply will be disabled expect DPD control logic to control DPD Power-down mode wake-up source. After wake-up from Deep Power-down mode (DPD), system resets and executes code from the beginning again, all peripheral configuration and SRAM data will be lost. The Deep Power-down mode (DPD) wake-up sources include Wake-up Timer and Wake-up Pin(PC.0). Deep Power-down mode (DPD) disables more logic to save power and needs more wake-up time to re-start system.

1. Deep Power-down Mode (DPD) configuration:

Call function CLK_SetPowerDownMode() to set Deep Power-down Mode (DPD) and call function CLK_PowerDown() then system will enter Deep Power-down Mode (DPD).

```
/* Set Deep Power-down mode (DPD) */  
CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_DPD);  
  
/* Enter Power-down mode */  
CLK_PowerDown();
```

2. Deep Power-down Mode (DPD) wake-up source:

Deep Power-down mode (DPD) wake-up sources include Wake-up Timer and Wake-up Pin(PC.0).

3 Sample Code

This section shows configuration and wake-up method of different power modes.

- Normal Mode (3.1)
Set different voltage level.
- Idle Mode (3.2)
System enters Idle Mode and wakes up by WDT interrupt.
- Power-down Mode (3.3)
System enters different Power-down Mode and wakes up by RTC 1 second tick interrupt except Deep Power-down Mode (DPD). System enters Deep Power-down Mode (DPD), it wakes up by Wake-up Timer time-out interrupt.
 1. Normal Power-down Mode (NPD) (3.3.1)
 2. Low Leakage Power-down Mode (LLPD) (3.3.2)
 3. Fast Wake-up Power-down Mode (FWPD) (3.3.3)
 4. Standby Power-down Mode 0 (SPD0) (3.3.4)
 5. Standby Power-down Mode 1 (SPD1) (3.3.5)
 6. Deep Power-down Mode (DPD) (3.3.6)

3.1 Normal Mode

The default system voltage level is Power Level 0 (1.26V) and the maximum CPU operation frequency is 192MHz. To save power, you can set Power Level 1 (1.2V) and CPU maximum operation frequency is lower 160MHz. To set voltage level from higher voltage to lower voltage, for example, switch from 1.26V to 1.2V, you need to make sure that the CPU operation frequency is lower or equal to 160MHz, and then switch to 1.2V.

```
void main(void)
{
    /* Unlock protected registers */
    SYS_UnlockReg();

    /* Set power level to 1.26V */
    SYS_SetPowerLevel(SYS_PLCTL_PLSEL_PL0);

    /* Set core clock as 192MHz from PLL */
    CLK_SetCoreClock(192000000);

    /* Set core clock as 160MHz from PLL */
    CLK_SetCoreClock(160000000);
}
```

```

/* Set power level to 1.2V */
SYS_SetPowerLevel(SYS_PLCTL_PLSEL_PL1);

while(1);
}

```

3.2 Idle Mode

If CPU is waiting for an interrupt only, you can set system in Idle mode. It is the fastest power mode can return to Normal mode to keep working. The following sample code shows how to wake up system from Idle mode by WDT time-out interrupt. The WDT time-out interrupt configuration is: enable WDT clock, select WDT clock source as LIRC, enable WDT interrupt and set time-out interval is 16384 LIRC clocks.

```

/* WDT time-out wake-up source setting */
void WDT_Init(void)
{
    /* Enable WDT module clock */
    CLK_EnableModuleClock(WDT_MODULE);

    /* Select WDT module clock source as LIRC */
    CLK_SetModuleClock(WDT_MODULE, CLK_CLKSEL1_WDTSEL_LIRC, 0);

    /* Enable WDT NVIC */
    NVIC_EnableIRQ(WDT_IRQn);

    /* Configure WDT settings and start WDT counting */
    WDT_Open(WDT_TIMEOUT_2POW14, NULL, FALSE, TRUE);

    /* Enable WDT interrupt function */
    WDT_EnableInt();
}

```

The WDT time-out interrupt occurs every 16384 LIRC clocks. Clear WDT time-out interrupt flag in WDT interrupt handler. The WDT time-out interrupt will wake up system if system is in Idle mode.

```

volatile uint8_t g_u8IsINTEvent = 0;

/* WDT IRQ Handler */
void WDT_IRQHandler(void)
{
    /* Check if WDT time-out interrupt occurred */
}

```



```

if(WDT_GET_TIMEOUT_INT_FLAG())
{
    /* Clear WDT time-out interrupt flag */
    WDT_CLEAR_TIMEOUT_INT_FLAG();
}

g_u8IsINTEvent = 1;
}

```

After WDT configuration, call CLK_Idle() to enter Idle mode and wait system wake-up by WDT time-out interrupt.

```

void main(void)
{
    /* Unlock protected registers */
    SYS_UnlockReg();

    /* WDT time-out wake-up source setting */
    WDT_Init();

    /* Enter to Idle Mode */
    CLK_Idle();

    /* Check if WDT time-out interrupt and wake-up occurred or not */
    while(g_u8IsINTEvent == 0);

    /* End of sample code */
    while(1);
}

```

3.3 Power-down Mode

The following shows wake-up from different Power-down modes by RTC tick interrupt. To configure RTC tick before entering Power-down mode: enable RTC clock, set RTC clock source as LXT and set RTC tick interval as 1 second.

```

/* RTC tick wake-up source setting */
void RTC_Init(void)
{
    /* Set PF multi-function pins for X32_OUT(PF.4) and X32_IN(PF.5) */
    SYS->GPF_MFPL = (SYS->GPF_MFPL & (~SYS_GPF_MFPL_PF4MFP_Msk)) | X32_OUT_PF4;
    SYS->GPF_MFPL = (SYS->GPF_MFPL & (~SYS_GPF_MFPL_PF5MFP_Msk)) | X32_IN_PF5;
}

```

```

/* Enable LXT clock and wait for LXT clock ready*/
CLK_EnableXtalRC(CLK_PWRCTL_LXTEN_Msk);
CLK_WaitClockReady(CLK_STATUS_LXTSTB_Msk);

/* Enable RTC peripheral clock */
CLK_EnableModuleClock(RTC_MODULE);

/* Wait RTC access enable */
RTC_WaitAccessEnable();

/* Clear RTC tick interrupt flag */
RTC_CLEAR_TICK_INT_FLAG(RTC);

/* Enable RTC NVIC */
NVIC_EnableIRQ(RTC_IRQn);

/* Enable RTC tick interrupt and wake-up function will be enabled also */
RTC_EnableInt(RTC_INTEN_TICKIEN_Msk);
RTC_SetTickPeriod(RTC_TICK_1_SEC);
}

```

After RTC tick interrupt configuration, system enters RTC tick interrupt every 1 second. Clear RTC tick interrupt in RTC interrupt handler. RTC tick interrupt will wake system up if system is in Power-down mode.

```

volatile uint32_t g_u32RTCTickINT = 0;

/* RTC IRQ Handler */
void RTC_IRQHandler(void)
{
    /* Check if RTC tick interrupt occurred */
    if(RTC_GET_TICK_INT_FLAG(RTC) == 1)
    {
        /* Clear RTC tick interrupt flag */
        RTC_CLEAR_TICK_INT_FLAG(RTC);
    }

    g_u32RTCTickINT++;
}

```

3.3.1 Normal Power-down Mode (NPD)

After RTC tick configuration, call function CLK_SetPowerDownMode() to set Normal Power-down mode (NPD) and call CLK_PowerDown() then system will enter Normal Power-down mode (NPD). System wakes up from Normal Power-down mode (NPD) by RTC tick interrupt and returns to Normal mode.

```
void main(void)
{
    /* Unlock protected registers */
    SYS_UnlockReg();

    /* RTC tick wake-up source setting */
    RTC_Init();

    /* Set Normal Power-down Mode (NPD) */
    CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_PD);

    /* Enter to Power-down Mode */
    CLK_PowerDown();

    /* Wait RTC tick interrupt and wake-up */
    while( g_u32RTCTickINT == 0);

    /* End of sample code */
    while(1);
}
```

3.3.2 Low Leakage Power-down Mode (LLPD)

After RTC tick configuration, call function CLK_SetPowerDownMode() to set Low Leakage Power-down mode (LLPD) and call CLK_PowerDown() then system will enter Low Leakage Power-down mode (LLPD). System wakes up from Low Leakage Power-down mode (LLPD) by RTC tick interrupt and returns to Normal mode.

```
void main(void)
{
    /* Unlock protected registers */
    SYS_UnlockReg();

    /* RTC tick wake-up source setting */
    RTC_Init();
```

```

/* Set Low Leakage Power-down Mode (LLPD) */
CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_LLPD);

/* Enter to Power-down Mode */
CLK_PowerDown();

/* Wait RTC tick interrupt and wake-up */
while( g_u32RTCTickINT == 0);

/* End of sample code */
while(1);
}

```

3.3.3 Fast Wake-up Power-down Mode (FWPD)

After RTC tick configuration, call function CLK_SetPowerDownMode() to set Fast Wake-up Power-down mode (FWPD) and call CLK_PowerDown() then system will enter Fast Wake-up Power-down mode (FWPD). System wakes up from Fast Wake-up Power-down mode (FWPD) by RTC tick interrupt and returns to Normal mode.

```

void main(void)
{
    /* Unlock protected registers */
    SYS_UnlockReg();

    /* RTC tick wake-up source setting */
    RTC_Init();

    /* Set Fast Wake-up Power-down Mode (FWPD) */
    CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_FWPD);

    /* Enter to Power-down Mode */
    CLK_PowerDown();

    /* Wait RTC tick interrupt and wake-up */
    while( g_u32RTCTickINT == 0);

    /* End of sample code */
    while(1);
}

```

3.3.4 Standby Power-down Mode 0(SPD0)

After RTC tick configuration, enable RTC wake-up at Standby Power-down mode. And write data 0x125ab234 to SRAM 0x20007FFC then call function CLK_SetPowerDownMode() to set Standby Power-down Mode 0(SPD0) and call CLK_PowerDown() then system will enter to Standby Power-down Mode 0(SPD0). System wakes up from Standby Power-down Mode 0(SPD0) by RTC tick interrupt, resets and returns to Normal Mode. After system wakes up, read data at SRAM 0x20007FFC to confirm SRAM Bank0 data will be retained. You have to write 1 to register CLK_IOPDCTL (GPIO Standby Power-down Control Register) to release GPIO hold state function and can control GPIO normally.

If this example declares variables to confirm the data retention state of SPD0, there are two things to notice.

1. The storage location of the variables must be in SRAM Bank0, because the variables may be configured to an unknown location by Compiler, and the variable data cannot be retained. The user can specify the variable placement position through the Scatter File/Linker Script.
2. Since the Standby Power-down Mode 0 (SPD0) returns to the Normal Mode, the system will reset. At this time, ARM's compiler platform (e.g. Keil) executive will initialize all variables to 0. The variable data stored in SRAM BANK0 cannot be retained. KEIL provides Nolnit options that users can set to avoid this problem.

```
#define SIGNATURE      0x125ab234
#define FLAG_ADDR      0x20007FFC

void main(void)
{
    /* if wake-up from Standby Power-down Mode 0/1(SPD0/1)*/
    If(CLK_GetPMUWKSrc())
    {
        /* Release I/O hold status after wake-up from Standby Power-down Mode (SPD) */
        CLK->IOPDCTL = 1;

        /* Clear Power Manager Status register */
        CLK->PMUSTS = CLK_PMUSTS_CLRWK_Msk;
    }

    /* Unlock protected registers */
    SYS_UnlockReg();

    /* RTC tick wake-up source setting */
    RTC_Init();
}
```

```

/* Check SRAM data is retained or not when system wake up from SPD mode */
if(M32(FLAG_ADDR) == SIGNATURE) {
    printf("System waken-up from SPD0 mode done!\n");
    M32(FLAG_ADDR) = 0;
    printf("Press any key to continue ...\n");
    getchar();
} else
    printf("System waken-up from SPD1 mode done!\n");

/* SRAM retention test */
M32(FLAG_ADDR) = SIGNATURE;

/* Set Standby Power-down Mode 0 (SPD0) */
CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_SPD0);

/* Enable RTC wake-up */
CLK_ENABLE_RTCWK();

/* Enter to Power-down Mode */
CLK_PowerDown();

/* Wait RTC tick interrupt and wake-up */
while( g_u32RTCTickINT == 0);

/* End of sample code */
while(1);
}

```

3.3.5 Standby Power-down Mode 1 (SPD1)

After RTC tick configuration, enable RTC wake-up at Standby Power-down mode. And write data 0x125ab234 to SRAM 0x20007FFC then call CLK_PowerDown() then system will enter to Standby Power-down Mode 1 (SPD1). System wakes up from Standby Power-down Mode 1 (SPD1) by RTC tick interrupt and, resets and returns to Normal Mode. At this time, there is no data in the SRAM 0x20007FFC. System SRAM Bank0 data has been lost after the system wakes up from Standby Power-down Mode 1 (SPD1). After system wakes up, you have to write 1 to register CLK_IOPDCTL (GPIO Standby Power-down Control Register) to release GPIO hold state function and can control GPIO normally.

```

#define SIGNATURE      0x125ab234
#define FLAG_ADDR      0x20007FFC

```

```

void main(void)
{
    /* if wake-up from Standby Power-down Mode 0/1(SPD0/1) */
    If(CLK_GetPMUWKSrc())
    {
        /* Release I/O hold status after wake-up from Standby Power-down Mode (SPD) */
        CLK->IOPDCTL = 1;

        /* Clear Power Manager Status register */
        CLK->PMUSTS = CLK_PMUSTS_CLRWK_Msk;
    }

    /* Unlock protected registers */
    SYS_UnlockReg();

    /* RTC tick wake-up source setting */
    RTC_Init();

    /* Check SRAM data is retained or not when system wake up from SPD mode */
    if(M32(FLAG_ADDR) == SIGNATURE) {
        printf("System waken-up from SPD0 mode done!\n");
        M32(FLAG_ADDR) = 0;
        printf("Press any key to continue ...\n");
        getchar();
    } else
        printf("System waken-up from SPD1 mode done!\n");

    /* SRAM retention test */
    M32(FLAG_ADDR) = SIGNATURE;

    /* Set Standby Power-down Mode 0(SPD1) */
    CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_SPD1);

    /* Enable RTC wake-up */
    CLK_ENABLE_RTCWK();

    /* Enter to Power-down Mode */
    CLK_PowerDown();

    /* Wait RTC tick interrupt and wake-up */

```

```

while( g_u32RTCTickINT == 0);

/* End of sample code */
while(1);
}

```

3.3.6 Deep Power-down Mode (DPD)

First, enable LIRC clock for Wake-up Timer. To configure Wake-up Timer's time-out interval is 16384 LIRC clocks and enabled Wake-up Timer at Deep Power-down Mode (DPD). And call function CLK_SetPowerDownMode() to set Deep Power-down Mode (DPD) and call CLK_PowerDown() then system will enter to Deep Power-down Mode (DPD). System wakes up from Deep Power-down Mode (DPD) by Wake-up Timer time-out interrupt. System resets and returns to Normal Mode.

```

void main(void)
{
    /* Unlock protected registers */
    SYS_UnlockReg();

    /* Set Wake-up Timer Time-out Interval */
    CLK_SET_WKTMR_INTERVAL(CLK_PMUCTL_WKTMRIS_16384);

    /* Set Deep Power-down Mode (DPD) */
    CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_DPD);

    /* Enable Wake-up Timer */
    CLK_ENABLE_WKTMR();

    /* Enter to Power-down Mode */
    CLK_PowerDown();

    /* End of sample code */
    while(1);
}

```


4 Conclusion

The NuMicro® M480 series has different power modes for users to design their application both high efficiency and low power. The following lists the comparison of these power modes including entry configuration, wake-up sources, and data retention states.

- Normal Mode
- Idle Mode
- Power-down Mode
 - Normal Power-down Mode (NPD)
 - Low Leakage Power-down Mode (LLPD)
 - Fast Wake-up Power-down Mode (FWPD)
 - Standby Power-down Mode 0 (SPD0)
 - Standby Power-down Mode 1 (SPD1)
 - Deep Power-down Mode (DPD)

For details of power consumption and wake-up time, please refer to the M480 Datasheet.

Table 4-1 shows the power mode configuration comparison.

Power Mode	PDMSEL (CLK_PMUCTL[2:0])	SLEEPDEEP (SCR[2])	PDEN (CLK_PWRCTL [7])	Run WFI Instruction
Normal Mode	N/A	N/A	N/A	N/A
Idle Mode	N/A	0	0	Yes
NPD	0	1	1	Yes
LLPD	1			
FWPD	2			
SPD0	4			
SPD1	5			
DPD	6			

Table 4-1 Different Power Mode Configuration Comparison

Table 4-2 shows the power mode wake-up source comparison.

Power Mode	Wake-up source
Normal Mode	System default states no wake-up source.
Idle Mode	Any interrupts.
Normal Power-down Mode (NPD)	Wake-up source: RTC, WDT, I ² C, Timer, UART, BOD, GPIO, EINT, USBH, USBD and ACMP.
Low Leakage Power-down Mode (LLPD)	
Fast Wake-up Power-down Mode (FWPD)	
Standby Power-down Mode 0 (SPD0)	SPD wake-up source: GPIO (PA, PB, PC and PD), BOD, LVR, ACMP, Wake-up Timer and RTC.
Standby Power-down Mode 1 (SPD1)	
Deep Power-down Mode (DPD)	DPD wake-up source: Wake-up Timer and Wake-up Pin(PC.0) °

Table 4-2 Different Power Mode Wake-up Source Comparison

Table 4-3 shows the power mode data retention comparison.

Power Mode	Peripheral configuration	Peripheral SRAM	System SRAM Bank0
Normal Mode	Retain	Retain	Retain
Idle Mode			
NPD			
LLPD			
FWPD			
SPD0	Lose	Lose	Retain

SPD1	Lose	Lost	Lose
DPD	Lose	Lose	Lose

Table 4-3 Power Mode Data Retention Comparison

Revision History

Date	Revision	Description
2018.11.28	1.00	1. Initially issued.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

*Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.*