AN13530

LPC553x/LPC55S3x Low-power Modes and Wake-up Time

Rev. 2 — 4 September 2023 Application note

Document Information

Information	Content
Keywords	AN13530, LPC55S3x/LPC553x, low-power mode, wake-up time
Abstract	This application note introduces various low-power modes of the LPC553x/LPC55S3x series, the software APIs details to enter in the low-power mode, and a wake-up source used for each low-power mode.

1 Introduction

LPC553x/LPC55S3x is an Arm Cortex-M33-based microcontroller for embedded applications. The family offers a rich set of peripherals with a feature of low power consumption. For more details on the LPC553x/LPC55S3x MCU, see LPC5500 Cortex-M33.

This application note introduces various low-power modes of the LPC553x/LPC55S3x series, the software APIs details to enter in the low-power mode, and a wake-up source used for each low-power mode. This document also describes the hardware and software environment as well as a procedure to measure supply current and wake-up time for each low-power mode.

This application note explains:

- All low-power modes in LPC553x/LPC55S3x
- Entry and wake-up implementations for low-power modes
- Steps to measure current and wake-up time for each low-power mode

2 Low-power modes

LPC553x/LPC55S3x supports four reduced power modes: Sleep, Deep sleep, Power down, and Deep power down.

These modes can be activated by power library APIs from the SDK software package.

Power usage is controlled by settings in the register within the SYSCON block, regulator settings controlled via the power library APIs, and the operating mode of the CPU. The following modes are supported in order from maximum to minimum power consumption.

2.1 Active mode

The device is in the Active mode after any chip-level reset. The following reset values determine the default dynamic power consumption:

- Power Management Controller registers, `PDRUNCFG0/1` (For details, see "Chapter 36 Power Management Controller" in *LPC553x Reference Manual* (document LPC553xRM))
- SYSCON registers `AHBCLKCTRL0/1/2/3` and `AUTOCLKGATEOVERRIDE` (For details, see "Chapter 7 SYSCON" in *LPC553x Reference Manual* (document LPC553xRM)

In the Active mode, by default, the core logic supply source is the Low Drop-Out Regulator (`LDO_CORE`). The user can switch to the DC-DC converter to gain better dynamic power efficiency via a set of functions provided in the power library: `POWER_SetCorePowerSource(...)` and `POWER_GetCorePowerSource(...)`.

In the Active mode, it is possible to individually control all the SRAM instances power mode via the function `POWER_SRAMPowerModeControl(...)` provided in the power library.

In the Active mode, when the device is in operation, it is possible to reduce the power consumption of the various peripherals that are not used (temporarily or permanently) by:

- Switching off their functional clock; see SYSCON registers `AHBCLKCTRL0/1/2/3`
- Activating the "automatic clock gating" feature; see SYSCON register `AUTOCLKGATEOVERRIDE`

- Switching off the entire module when possible; see Power Management Controller registers PDRUNCFG0 and PDRUNCFG1

The Active mode consumes the highest power among all power modes. All low-power modes can be invoked from this power mode.

2.2 Sleep mode

Sleep mode saves some power by stopping the Cortex-M33 CPU execution without affecting peripherals or requiring significant wake-up time. The clock to the CPU is shut off. Peripherals and memories are active and operational. CPU is stopped and execution of instructions is suspended until either a reset or an interrupt occurs.

Peripheral functions, if selected to be clocked in the AHBCLKCTRL registers, continue operation during Sleep mode and can generate interrupts to cause the processor to resume execution. Sleep mode only eliminates dynamic power used by the processor itself, memory systems and related controllers, and internal buses. The processor state and registers, peripheral registers, and internal SRAM values are maintained.

2.3 Deep sleep mode

In the Deep sleep mode, the full device remains powered, but flash and ROM are shut down, with the cost of a longer wake-up time compared to the Sleep mode. The system clock to the CPU is disabled as in the Sleep mode. Analog blocks are powered down by default but can be selected to keep running through the power API if needed as wake-up sources. The main clock and all peripheral clocks are disabled.

The Deep sleep mode eliminates power used by the analog peripherals and all dynamic power used by the CPU, its memory systems and related controllers, and internal buses. The CPU state and registers, peripheral registers, and internal SRAM values are maintained, and the logic levels of the pins remain static. All SRAM instances that are not configured to enter in the SRAM Deep sleep low-power mode keep the state they are in before entering the Deep sleep. This means that if a SRAM was in the Active state before entering the Deep sleep, it stays in the Active state during the Deep sleep and therefore it consumes more power.

Selected peripherals, such as GPIO group interrupts (GINT), USB Full speed, Flexcomms (SPI, I2C, USART), Widowed-Watchdog Timer, RTC, standard Counter/Timers can be left running in the Deep sleep mode. The oscillators, such as FRO1M, FRO32K, XTAL32K but also FRO192M (which delivers the 12 MHz and 96 MHz clocks), can also be left running.

Some peripherals can have DMA service during the Deep sleep mode without waking up the entire device.

2.4 Power-down mode

The Power-down mode turns off nearly all on-chip power consumption by:

- Eliminating power used by almost all analog modules and
- Shutting down the DC-DC and LDO_CORE

This eliminates almost all digital peripherals power, with the cost of a longer wake-up time compared to the Deep sleep mode. FRO192M is disabled. The flash memory is also disabled. The clock to the CPU and peripherals is shut down and if not configured, the peripherals in the power domains PD_SYSTEM and PD_AO receive no internal clocks. All SRAM instances can be configured to maintain their internal states and all registers lose their internal states except those located in the power domains PD_SYSTEM and PD_AO. Any SRAM instance that is not configured to maintain its internal state loses it.

The Cortex-M33 CPU state and some critical peripherals, such as security controller, PRINCE, and analog controller are retained and the logic levels of the pins remain static. GPIO group interrupts (GINT0 and GINT1), selected serial peripherals in Flexcomm3 (SPI, I2C, USART), RTC, OS event timers, and analog comparator can be left running to wake up the device.

When a wake-up event occurs, the Cortex-M33 CPU code execution resumes. It is the responsibility of the customer application to reconfigure all modules in the power domain core PD_CORE (whose states have not been retained, for example all Flexcomm except Flexcomm3, SDMA, Power Quad, DMIC).

2.5 Deep Power-down mode

Deep Power-down mode shuts down virtually all on-chip power consumption, but requires a longer wake-up time (compared to the Power-down mode). For maximal power savings, the entire power domains PD_CORE (CPU, DSP, Flexcomms, SDMA) and PD_SYSTEM (Reset, Clocks, SYSCON, IOCON) are shut down. Only the Always-on power domain PD_AO (PMU, PMC, RTC, and OS Event Timer) stays powered. Clocks are shut off to the entire chip device except for the RTC and the OS Event Timer, if they are needed. On wake-up, the device reboots.

During the Deep Power-down mode, the contents of some SRAM can be retained (software configured via the POWER_EnterDeepPowerDown() low-power API) but registers (other than those in PMC, RTC, and OS Event Timer) are not retained. All functional pins are tri-stated in the Deep Power-down mode, except the 5 wake-up pins and the RESETN pin.

2.6 Low-power mode summary

The following table summarizes the power state of the different power domains according to the power modes.

Table 1. Power modes vs Power domains

Power mode	PD_CORE	PD_SYSTEM	PD_AON	PD_MEM_0	PD_MEM_1	PD_MEM_2
Active	ON	ON	ON	ON	ON	ON
Sleep	ON	ON	ON	ON	ON	ON
Deep sleep	ON	ON	ON	Configurable [1]	Configurable [2]	Configurable[3]
Power down	OFF	ON	ON	Configurable	Configurable	Configurable
Deep power down	OFF	OFF	ON	Configurable	Configurable	Configurable

[1] Depending on whether the RAM_01 Deep sleep power mode (data retention) is required by the user via the relevant Power API function, this power domain is ON or OFF.
[2] Depending on whether the Deep sleep power mode (data retention) is required for one SRAM instance in this power domain by the user via the relevant Power API function, this power domain is ON or OFF.
[3] Depending on whether the RAM_00 Deep sleep power mode (data retention) is required by the user via the relevant Power API function, this power domain is ON or OFF.

The following table summarizes the power state of the different peripherals according to the power modes.

Table 2. Peripherals reduced power modes

Peripherals		Device low-power modes			
Name	Description	Sleep mode	Deep sleep mode	Power-down mode	Deep Power-down mode
DCDC	Bulk DC-DC converter	Same state as in Active mode	OFF	OFF	OFF
BIAS	Analog references	ON	ON	Software configured	OFF
BODCORE	Core logic supply brownout detector	Same state as in Active mode	Software configured	OFF	OFF

Table 2. Peripherals reduced power modes*...continued*

Peripherals		Device low-power modes			
Name	**Description**	**Sleep mode**	**Deep sleep mode**	**Power-down mode**	**Deep Power-down mode**
BODVDDMAIN	VDD_MAIN supply brownout detector	Same state as in Active mode	Software configured	OFF	OFF
FRO1M	1 MHz free running oscillator	ON	Software configured	Software configured	Software configured
FRO192M	High-speed free running oscillator	ON	Software configured	OFF	OFF
FRO32K	32 kHz free running oscillator	Same state as in Active mode	Software configured	Software configured	Software configured
XTAL32K	32 kHz crystal oscillator	Same state as in Active mode	Software configured	Software configured	Software configured
XTALHF	High frequency crystal oscillator	Same state as in Active mode	Software configured	OFF	OFF
PLL0	Phase-locked loop module 0	Same state as in Active mode	Software configured	OFF	OFF
PLL1	Phase-locked loop module 1	Same state as in Active mode	Software configured	OFF	OFF
USBFSPHY	USB full speed physical interface	Same state as in Active mode	Software configured	OFF	OFF
COMP	Analog comparator	Same state as in Active mode	Software configured	Software configured	OFF
LDOEFUSEPROG	eFUSE programming low drop-out regulator	Same state as in Active mode	Software configured	OFF	OFF
LDOXTALHF	High frequency crystal oscillator Low drop-out regulator	Same state as in Active mode	Software configured	OFF	OFF
LDOFLASHNV	Non-volatile flash macro low Drop-out regulator	ON	OFF	OFF	OFF
PLL0_SSCG	PLL0 spread spectrum clock Generator	Same state as in Active mode	Software configured	OFF	OFF
HSCMP0	High-speed comparator 0	Same state as in Active mode	Software configured	OFF	OFF
HSCMP1	High-speed comparator 1	Same state as in Active mode	Software configured	OFF	OFF
HSCMP2	High-speed comparator 2	Same state as in Active mode	Software configured	OFF	OFF
OPAMP0	Operational amplifier 0	Same state as in Active mode	Software configured	OFF	OFF

Table 2. Peripherals reduced power modes*...continued*

Peripherals		Device low-power modes			
Name	**Description**	**Sleep mode**	**Deep sleep mode**	**Power-down mode**	**Deep Power-down mode**
OPAMP1	Operational amplifier 1	Same state as in Active mode	Software configured	OFF	OFF
OPAMP2	Operational amplifier 2	Same state as in Active mode	Software configured	OFF	OFF
VREF	ADCs/DACs analog reference module	Same state as in Active mode	Software configured	Software configured	OFF
CMPBIAS	High-speed comparators biasing	Same state as in Active mode	Software configured	OFF	OFF
HSCMP0_DAC	HSCMP0 (internal) DAC	Same state as in Active mode	Software configured	OFF	OFF
HSCMP1_DAC	HSCMP1 (internal) DAC	Same state as in Active mode	Software configured	OFF	OFF
HSCMP2_DAC	HSCMP2 (internal) DAC	Same state as in Active mode	Software configured	OFF	OFF
DAC0	Digital-to-analog converter 0	Same state as in Active mode	Software configured	OFF	OFF
DAC1	Digital-to-analog converter 1	Same state as in Active mode	Software configured	OFF	OFF
DAC2	Digital-to-analog converter 2	Same state as in Active mode	Software configured	OFF	OFF
STOP_DAC0	DAC0 Stop mode	Same state as in Active mode	Software configured	OFF	OFF
STOP_DAC1	DAC1 Stop mode	Same state as in Active mode	Software configured	OFF	OFF
STOP_DAC2	DAC2 Stop mode	Same state as in Active mode	Software configured	OFF	OFF

3 Entering Low-power Modes and Waking up

Power usage is controlled by settings in the register within the `SYSCON` block, regulator settings controlled by the Power APIs, and the operating mode of the CPU. This application note describes how to enter and wake up from various low-power modes.

3.1 Power control API

The power control APIs provide the functions to configure the system for expected performance requirements. The following table lists the power APIs used in the application.

Table 3. Power API ROM calls

Function prototype	API description
`void POWER_EnterSleep(void);`	The `POWER_EnterSleep()` API puts the device in the Sleep low-power mode; only the activity of the Arm Cortex-M33 is stopped.
`void POWER_EnterDeepSleep(uint32_t exclude_from_pd[2], uint32_t sram_retention_ctrl, uint32_t wakeup_interrupts[4], uint32_t hardware_wake_ctrl);`	The `POWER_EnterDeepSleep()` API configures the Deep sleep low-power mode. It controls which peripherals are powered up and the SRAM instances low-power mode during Deep sleep.
`void POWER_EnterPowerDown(uint32_t exclude_from_pd[1], uint32_t sram_retention_ctrl, uint32_t wakeup_interrupts[2], uint32_t cpu_retention_addr);`	The `POWER_EnterPowerDown` API configures the power down low-power mode. It controls which peripherals are powered up and which SRAM instances are in retention state during power down.
`void POWER_EnterDeepPowerDown(uint32_t exclude_from_pd[1], uint32_t sram_retention_ctrl, uint32_t wakeup_interrupts[2], uint32_t wakeup_io_ctrl);`	The `POWER_EnterDeepPowerDown` API configures the Deep power down low-power mode. It controls which peripherals are powered up and which SRAM instances are in retention state during Deep power down.

3.1.1 POWER_EnterSleep

The `POWER_EnterSleep()` API puts the device in the Sleep low-power mode; only the activity of the Arm Cortex-M33 is stopped.

The device wakes up from the Sleep mode after:

- any interrupt enabled in the Cortex-M33 Nested Vector Interrupt Controller (NVIC) arrives at the processor
- any chip reset occurs (power-on reset, brownout detector reset, pin reset, and so on)

Table 4. POWER_EnterSleep

Routine	API description
Function Prototype	`void POWER_EnterSleep(void);`
Input Parameter	None
Result	None
Description	Configures and enters in Sleep low-power mode.

3.1.2 POWER_EnterDeepSleep

The `POWER_EnterDeepSleep()` API configures the Deep sleep low-power mode. It allows controlling which peripherals are powered up and the SRAM instances low-power mode during Deep sleep. It defines which peripheral interrupts can be a wake-up source during Deep sleep.

The `POWER_EnterDeepSleep` API switches the CPU and System Bus clock to 12 MHz. It is the responsibility of the user to reconfigure the CPU and System Bus clock after the function returns.

Table 5. POWER_EnterDeepSleep

Routine	API Description
Function Prototype	`void POWER_EnterDeepSleep(uint32_t exclude_from_pd[2], uint32_t sram_retention_ctrl, uint32_t wakeup_interrupts[4], uint32_t hardware_wake_ctrl);`
Input Parameters	

Table 5. POWER_EnterDeepSleep*...continued*

Routine	API Description
`exclude_from_pd[2]`	Defines which analog peripherals shall not be powered down. It is a 2 x 32 bit-vector, with each bit of the vector corresponding to one module. The definition of the vector is given in Table 6 and Table 7, and is aligned with the `power_pd_bit_t` type definition. For each bit field of the vector: • 0: Module is powered down during Deep sleep. • 1: Module is running during Deep sleep.
`sram_retention_ctrl`	Defines which SRAM instances shall be put in the SRAM Deep sleep low-power mode during Deep sleep. SRAM instances in the SRAM Deep sleep low-power mode do not lose their content, but they cannot be involved in any DMA transfer during Deep sleep. SRAM instances that are not required to be put in the SRAM Deep sleep low-power mode during Deep sleep keep the state they were in before calling the API, that is: • If the SRAM instance was in Active mode, it stays in the Active mode during Deep sleep and after wake-up from Deep sleep. Such SRAM instance can be involved in DMA transfer during Deep sleep. • If the SRAM instance was in Deep sleep mode, it stays in the SRAM Deep sleep low-power mode during Deep sleep and after wake-up from Deep sleep. Such SRAM instance cannot be involved in DMA transfer during Deep sleep. • If the SRAM instance was in Shutdown mode, it stays in the SRAM Shutdown low-power mode during Deep sleep and after wake-up from Deep sleep. Such SRAM instance cannot be involved in DMA transfer during Deep sleep. The `sram_retention_ctrl` parameter is a 32-bit vector, with each bit of the vector corresponding to one SRAM instance. The definition of the vector is given in Table 8 and is aligned with the `power_sram_bit_t` type definition. For each bit field of the vector: • 0: During Deep sleep, the SRAM instance keeps the state it had before entering Deep sleep. • 1: During Deep sleep, the SRAM instance is in the SRAM Deep sleep low-power mode (content preserved). ***Note:*** *When SDMA transfers are expected to occur during Deep sleep, make sure that the SRAM instances in which the SDMA descriptors and the Peripheral's data are stored are in Active mode during Deep sleep.*
wakeup_interrupts[4]	Defines which peripheral interrupts can be a wake-up source during Deep sleep. It is a 4 x 32-bit vector, with each bit inside the vector corresponding to one interrupt source. The definition of the vector is given in Table 9, Table 10, Table 11, Table 12 and is aligned with the low-power modes wake-up sources (`#define WAKEUP_*`) #defines definitions. For each bit field of the vector: • 0: The associated peripheral cannot be a wake-up source during Deep sleep. • 1: The associated peripheral can be a wake-up source during Deep sleep.
hardware_wake_ctrl	Provides the possibility for all Flexcomm, High-Speed SPI, all DAC and DMIC to have DMA service during Deep sleep without waking up an entire device. The detailed mapping of the parameter is given in Table 13 and is aligned with the definition of `#define LOWPOWER_HWWAKE_`.
Result	None
Description	Controls which peripherals are powered up and the SRAM instances low-power mode during Deep sleep. Defines which peripheral interrupts can be a wake-up source during Deep sleep.

Table 6. Parameter exclude_from_pd[0] definition (1st vector)

Bit	Symbol	Description	Value
1:0	-	Reserved	Must be set to 0
2	BODCORE	Core Logic supply brown-out detector	• 0: Powered down • 1: Running

Table 6. Parameter exclude_from_pd[0] definition (1st vector)*...continued*

Bit	Symbol	Description	Value
3	BODVDDMAIN	VDD_MAIN supply brown-out detector	• 0: Powered down • 1: Running
4	FRO1M	1 MHz free running oscillator	• 0: Powered down • 1: Running
5	FRO192M	High-speed free running oscillator	• 0: Powered down • 1: Running
6	FRO32K	32 kHz free running oscillator	• 0: Powered down • 1: Running
7	XTAL32K	32 kHz crystal oscillator	• 0: Powered down • 1: Running
8	XTALHF	High-frequency crystal oscillator	• 0: Powered down • 1: Running
9	PLL0	Phase-locked loop module 0	• 0: Powered down • 1: Running
10	PLL1	Phase-locked loop module 1	• 0: Powered down • 1: Running
11	USBFSPHY	USB full speed physical interface	• 0: Powered down • 1: Running
12	-	Reserved	Must be set to 0
13	COMP		• 0: Powered down • 1: Running
17:14	-	Reserved	Must be set to 0
18	LDOEFUSEPROG	eFUSE programming low drop-out regulator	• 0: Powered down • 1: Running
19	-	Reserved	Must be set to 0
20	LDOXTALHF	High-frequency crystal oscillator low drop-out regulator	• 0: Powered down • 1: Running
22:21	-	Reserved	Must be set to 0
23	PLL0_SSCG	PLL0 spread spectrum clock generator	• 0: Powered down • 1: Running
24	-	Reserved	Must be set to 0
25	HSCMP0	High-speed comparator 0	• 0: Powered down • 1: Running
26	HSCMP1	High-speed comparator 1	• 0: Powered down • 1: Running
27	HSCMP2	High-speed comparator 2	• 0: Powered down • 1: Running
28	OPAMP0	Operational amplifier 0	• 0: Powered down • 1: Running
29	OPAMP1	Operational amplifier 1	• 0: Powered down • 1: Running

Table 6. Parameter exclude_from_pd[0] definition (1st vector)*...continued*

Bit	Symbol	Description	Value
30	OPAMP2	Operational amplifier 2	• 0: Powered down • 1: Running
31	VREF	ADCs/DACs analog reference module	• 0: Powered down • 1: Running

Table 7. Parameter exclude_from_pd[1] definition (2nd vector)

Bit	Symbol	Description	Value
0	CMPBIAS	High-speed comparators biasing	• 0: Powered down • 1: Running
1	HSCMP0_DAC	HSCMP0 (internal) DAC	• 0: Powered down • 1: Running
2	HSCMP1_DAC	HSCMP1 (internal) DAC	• 0: Powered down • 1: Running
3	HSCMP2_DAC	HSCMP2 (internal) DAC	• 0: Powered down • 1: Running
4	DAC0	Digital-to-analog converter 0	• 0: Powered down • 1: Running
5	DAC1	Digital-to-analog converter 1	• 0: Powered down • 1: Running
6	DAC2	Digital-to-analog converter 2	• 0: Powered down • 1: Running
7	STOP_DAC0	DAC0 Stop mode	• 0: Stop mode enabled • 1: Stop mode disabled
8	STOP_DAC1	DAC1 Stop mode	• 0: Stop mode enabled • 1: Stop mode disabled
9	STOP_DAC2	DAC2 Stop mode	• 0: Stop mode enabled • 1: Stop mode disabled
31:10	-	Reserved	Must be set to 0

Table 8. Parameter sram_retention_ctrl

Bit	SRAM instance (SRAM size)	Value
0	RAM_X0 (16 kB)	• 0: Keeps state prior entering Deep sleep • 1: Deep sleep (content preserved)
1	RAM_00 (4 kB)	• 0: Keeps state prior entering Deep sleep • 1: Deep sleep (content preserved)
2	RAM_01 (4 kB)	• 0: Keeps state prior entering Deep sleep • 1: Deep sleep (content preserved)
3	RAM_02 (4 kB)	• 0: Keeps state prior entering Deep sleep • 1: Deep sleep (content preserved)
4	RAM_03 (4 kB)	• 0: Keeps state prior entering Deep sleep • 1: Deep sleep (content preserved)

Table 8. Parameter sram_retention_ctrl*...continued*

Bit	SRAM instance (SRAM size)	Value
5	RAM_10 (16 kB)	• 0: Keeps state prior entering Deep sleep • 1: Deep sleep (content preserved)
6	RAM_20 (32 kB)	• 0: Keeps state prior entering Deep sleep • 1: Deep sleep (content preserved)
7	RAM_30 (32 kB)	• 0: Keeps state prior entering Deep sleep • 1: Deep sleep (content preserved)
8	RAM_40 (4 kB)	• 0: Keeps state prior entering Deep sleep • 1: Deep sleep (content preserved)
9	RAM_41 (4 kB)	• 0: Keeps state prior entering Deep sleep • 1: Deep sleep (content preserved)
10	RAM_42 (4 kB)	• 0: Keeps state prior entering Deep sleep • 1: Deep sleep (content preserved)
11	RAM_43 (4 kB)	• 0: Keeps state prior entering Deep sleep • 1: Deep sleep (content preserved)
31:12	Reserved	Must be set to 0

Table 9. Parameter wakeup_interrupts[0] (1st vector)

Bit	Symbol	Description	Value
0	SYS	Watchdog, brownout Detectors (BOD VDDMAIN and BOD Core)	• 0: Interrupt disabled • 1: Interrupt enabled
1	SDMA0	SDMA0 controller	• 0: Interrupt disabled • 1: Interrupt enabled
2	GLOBALINT0	Group GPIO input interrupt 0 (GINT0)	• 0: Interrupt disabled • 1: Interrupt enabled
3	GLOBALINT1	Group GPIO input interrupt 1 (GINT1)	• 0: Interrupt disabled • 1: Interrupt enabled
4	GPIO_INT0_0	Pin interrupt 0 or pattern match engine slice 0 (PINT0)	• 0: Interrupt disabled • 1: Interrupt enabled
5	GPIO_INT0_1	Pin interrupt 1 or pattern match engine slice 1 (PINT1)	• 0: Interrupt disabled • 1: Interrupt enabled
6	GPIO_INT0_2	Pin interrupt 2 or pattern match engine slice 2 (PINT2)	• 0: Interrupt disabled • 1: Interrupt enabled
7	GPIO_INT0_3	Pin interrupt 3 or pattern match engine slice 3 (PINT3)	• 0: Interrupt disabled • 1: Interrupt enabled
8	UTICK	Micro-tick timer	• 0: Interrupt disabled • 1: Interrupt enabled
9	-	Reserved	Must be set to 0
10	CTIMER0	Standard counter/timer CTIMER0	• 0: Interrupt disabled • 1: Interrupt enabled
11	CTIMER1	Standard counter/timer CTIMER1	• 0: Interrupt disabled • 1: Interrupt enabled

Table 9. Parameter wakeup_interrupts[0] (1st vector)*...continued*

Bit	Symbol	Description	Value
12	-	Reserved	Must be set to 0
13	CTIMER3	Standard counter/timer CTIMER3	• 0: Interrupt disabled • 1: Interrupt enabled
14	FLEXCOMM0	Flexcomm Interface 0 (USART, SPI, I2C)	• 0: Interrupt disabled • 1: Interrupt enabled
15	FLEXCOMM1	Flexcomm Interface 1 (USART, SPI, I2C)	• 0: Interrupt disabled • 1: Interrupt enabled
16	FLEXCOMM2	Flexcomm Interface 2 (USART, SPI, I2C)	• 0: Interrupt disabled • 1: Interrupt enabled
17	FLEXCOMM3	Flexcomm Interface 3 (USART, SPI, I2C)	• 0: Interrupt disabled • 1: Interrupt enabled
18	FLEXCOMM4	Flexcomm Interface 4 (USART, SPI, I2C)	• 0: Interrupt disabled • 1: Interrupt enabled
19	FLEXCOMM5	Flexcomm Interface 5 (USART, SPI, I2C)	• 0: Interrupt disabled • 1: Interrupt enabled
20	FLEXCOMM6	Flexcomm Interface 6 (USART, SPI, I2C)	• 0: Interrupt disabled • 1: Interrupt enabled
21	FLEXCOMM7	Flexcomm Interface 7 (USART, SPI, I2C)	• 0: Interrupt disabled • 1: Interrupt enabled
23:22	-	Reserved	Must be set to 0
24	ACMP	Analog comparator	• 0: Interrupt disabled • 1: Interrupt enabled
25	-	Reserved	Must be set to 0
26	HWVAD	Hardware voice activity detector	• 0: Interrupt disabled • 1: Interrupt enabled
27	USB0_NEEDCLK	USB full speed	• 0: Interrupt disabled • 1: Interrupt enabled
28	-	Reserved	Must be set to 0
29	RTC	Real-time clock (counter event and tamper event)	• 0: Interrupt disabled • 1: Interrupt enabled
31:30	-	Reserved	Must be set to 0

Table 10. Parameter wakeup_interrupts[1] (2nd vector)

Bit	Symbol	Description	Value
0	GPIO_INT0_4	Pin interrupt 4 or pattern match engine slice 4 (PINT4)	• 0: Interrupt disabled • 1: Interrupt enabled
1	GPIO_INT0_5	Pin interrupt 5 or pattern match engine slice 5 (PINT5)	• 0: Interrupt disabled • 1: Interrupt enabled
2	GPIO_INT0_6	Pin interrupt 6 or pattern match engine slice 6 (PINT6)	• 0: Interrupt disabled • 1: Interrupt enabled

Table 10. Parameter wakeup_interrupts[1] (2nd vector)*...continued*

Bit	Symbol	Description	Value
3	GPIO_INT0_7	Pin interrupt 7 or pattern match engine slice 7 (PINT7)	• 0: Interrupt disabled • 1: Interrupt enabled
4	CTIMER2	Standard counter/timer CTIMER2	• 0: Interrupt disabled • 1: Interrupt enabled
5	CTIMER4	Standard counter/timer CTIMER4	• 0: Interrupt disabled • 1: Interrupt enabled
6	OS_EVENT_TIMER	OS Event Timer	• 0: Interrupt disabled • 1: Interrupt enabled
12:7	-	Reserved	Must be set to 0
13	SPIFILTER	SPI Filter	• 0: Interrupt disabled • 1: Interrupt enabled
17:14	-	Reserved	Must be set to 0
18	SEC_GPIO_INT0_0	Secure Pin interrupt 0 or pattern match engine slice 0 (Secure PINT0)	• 0: Interrupt disabled • 1: Interrupt enabled
19	SEC_GPIO_INT0_1	Secure Pin interrupt 1 or pattern match engine slice 1 (Secure PINT1)	• 0: Interrupt disabled • 1: Interrupt enabled
25:20	-	Reserved	Must be set to 0
26	SDMA1	SDMA1 control	• 0: Interrupt disabled • 1: Interrupt enabled
27	LSPI_HS	High-Speed SPI	• 0: Interrupt disabled • 1: Interrupt enabled
29:28	-	Reserved	Must be set to 0
30	I3C	MIPI I3C	• 0: Interrupt disabled • 1: Interrupt enabled
31	-	Reserved	Must be set to 0

Table 11. Parameter wakeup_interrupts[2] (3rd vector)

Bit	Symbol	Description	Value
1:0	-	Reserved	Must be set to 0
2	-	Reserved	Must be set to 0
9:3	-	Reserved	Must be set to 0
10	DAC0	Digital-to-analog converter 0	• 0: Interrupt disabled • 1: Interrupt enabled
11	DAC1	Digital-to-analog converter 1	• 0: Interrupt disabled • 1: Interrupt enabled
12	DAC2	Digital-to-analog converter 2	• 0: Interrupt disabled • 1: Interrupt enabled
31:13	-	Reserved	Must be set to 0

Table 12. Parameter wakeup_interrupts[3] (4th vector)

Bit	Symbol	Description	Value
31:0	-	Reserved	Must be set to 0

Table 13. Parameter hardware_wake_ctrl

Bit	Symbol	Description	Value
0	-	Reserved	Must be set to 0
1	HWWAKE_ PERIPHERALS	Wake for Flexcomms. Any Flexcomm FIFO reaching the level specified by its own TXLVL causes peripheral clocking to wake up temporarily while the related status is asserted	• 0: Disabled • 1: Enabled
2	HWWAKE_DMIC	Wake for Digital Microphone (DMIC). When 1, the DMIC input FIFO reaching the level specified by TRIGLVL of either channel causes peripheral clocking to be enabled temporarily while the DMIC FIFO level is at or above TRIGLVL. This allows DMA to become active to move data out of the DMIC FIFO. Used with LOWPOWER_ HWWAKE_PERIPHERALS	• 0: Disabled • 1: Enabled
3	HWWAKE_SDMA0	Wake for SDMA0. SDMA0 being busy causes peripheral clocking to remain running until DMA completes. Used with HWWAKE_ PERIPHERALS	• 0: Disabled • 1: Enabled
4	-	Reserved	Must be set to 0
5	HWWAKE_SDMA1	Wake for SDMA1. SDMA1 being busy causes peripheral clocking to remain running until DMA completes. Used with HWWAKE_ PERIPHERALS	• 0: Disabled • 1: Enabled
6	HWWAKE_DAC	Wake for DAC0, DAC1, DAC2. Any DAC0/1/2 FIFO reaching the level specified by the configuration generates an asynchronous SDMA0 request, and SDMA0 wakes up the bus clock temporarily to transfer data to DAC0/1/2	• 0: Disabled • 1: Enabled
31:7	-	Reserved	Must be set to 0

3.1.3 POWER_EnterPowerDown

The `POWER_EnterPowerDown` API configures the power down low-power mode and allows controlling which peripherals are powered up and which SRAM instances are in retention state during power down.

When a wake-up event occurs, CPU resumes code execution after the call to the low-power API function. However, during power down, a couple of modules lose their states and therefore, they must be reconfigured by the user application.

It is the responsibility of the user to make sure that the SRAM instance(s) containing the application software stacks and variables are preserved during power down.

The `POWER_EnterPowerDown` API switches the CPU and System Bus clock to 12 MHz. It is the responsibility of the user to reconfigure the CPU and System Bus clock after the function returns.

During power down, a couple of modules lose their state. It is the responsibility of the user application to reconfigure them if the application requires them.

Table 14. POWER_EnterPowerDown

Routine	API Description
Function Prototype	`void POWER_EnterPowerDown(uint32_t exclude_from_pd[1], uint32_t sram_retention_ctrl, uint32_t wakeup_interrupts[2], uint32_t cpu_retention_addr);`
Input Parameters	
`exclude_from_pd[1]`	Defines which analog peripherals shall not be powered down. It is a 1 x 32-bit vector, with each bit of the vector corresponding to one module. The definition of the vector is given in Table 15, and is aligned with the `power_pd_bit_t` type definition. For each bit field of the vector: • 0: The module is powered down during power down. • 1: The module is running during power down.
sram_retention_ctrl	Defines which SRAM instances shall preserve their content during power down. The `sram_retention_ctrl` parameter is a 32-bit vector, with each bit of the vector corresponding to one SRAM instance. The definition of the vector is given in Table 16, and is aligned with the `power_sram_bit_t` type definition. For each bit field of the vector: • 0: During power down, the SRAM instance is in the SRAM Shut-down low-power mode (content not preserved). • 1: During power down, the SRAM instance is in the SRAM Deep sleep low-power mode (content preserved). ***Note:*** *The SRAM instance, which is used to save Cortex-M33, AHB secure controller and Flash PRINCE modules states, is automatically put in the Deep sleep low-power mode (content preserved) by Low power.*
wakeup_interrupts[2]	Defines which peripheral interrupts can be a wake-up source during power down. It is a 2 x 32-bit vector, with each bit inside the vector corresponding to one interrupt source. The definition of the vector is given in Table 17 and Table 18, and is aligned with the Low-power modes wake-up sources (`#define WAKEUP_*`) #defines definitions. For each bit field of the vector: • 0: The associated peripheral cannot be a wake-up source during power down • 1: The associated peripheral can be a wake-up source during power down ***Note:*** *In case Flexcomm3 UART is used as wake-up source, a 32 kHz clock source (either FRO32K or XTAL32K) must be enabled during power down. The unique baud rate supported is 9600 Baud.*
cpu_retention_addr	Defines the start address of the area inside SRAM region where the Cortex-M33, AHB secure controller, and Flash PRINCE modules state are saved during power down: • Must be word-aligned (address ending by 0x0, 0x4, 0x8, and 0xC) • Can be any value: – Between 0x20000000 and 0x200009FC (inside RAM_00) – Between 0x20001000 and 0x200019FC (inside RAM_01) – Between 0x20002000 and 0x200029FC (inside RAM_02) – Between 0x20003000 and 0x200039FC (inside RAM_03) If the parameter does not meet the boundaries mentioned above, it is set to 0x20000000 (start of RAM_00) by the low-power API. ***Note:*** *The states of the Cortex-M33, AHB security controller, and PRINCE modules are stored in SRAM from cpu_retention_addr to cpu_retention_addr + 1540 Bytes - 1. Therefore, any user data present in this area before calling the function is overwritten and definitely lost.*
Result	None
Description	Controls which peripherals are powered up and which SRAM instances are in retention state during power down. Defines which SRAM instances shall preserve their content during power down.

Table 15. Parameter exclude_from_pd[0] definition

Bit	Symbol	Description	Value
3:0	-	Reserved	Must be set to 0
4	FRO1M	1 MHz free running oscillator	• 0: Powered down • 1: Running
5	-	Reserved	Must be set to 0
6	FRO32K	32 kHz free running oscillator	• 0: Powered down • 1: Running
7	XTAL32K	32 kHz crystal oscillator	• 0: Powered down • 1: Running
12:8	-	Reserved	Must be set to 0
13	COMP	Analog comparator	• 0: Powered down • 1: Running
30:14	-	Reserved	Must be set to 0
31	VREF	ADCs/DACs analog reference module	• 0: Powered down • 1: Running

Table 16. Parameter sram_retention_ctrl

Bit	SRAM instance (SRAM size)	Value
0	RAM_X0 (16 kB)	• 0: Shut Down (content not preserved) • 1: Deep Sleep (content preserved)
1	RAM_00 (4 kB)	• 0: Shut Down (content not preserved) • 1: Deep Sleep (content preserved)
2	RAM_01 (4 kB)	• 0: Shut Down (content not preserved) • 1: Deep Sleep (content preserved)
3	RAM_02 (4 kB)	• 0: Shut Down (content not preserved) • 1: Deep Sleep (content preserved)
4	RAM_03 (4 kB)	• 0: Shut Down (content not preserved) • 1: Deep Sleep (content preserved)
5	RAM_10 (16 kB)	• 0: Shut Down (content not preserved) • 1: Deep Sleep (content preserved)
6	RAM_20 (32 kB)	• 0: Shut Down (content not preserved) • 1: Deep Sleep (content preserved)
7	RAM_30 (32 kB)	• 0: Shut Down (content not preserved) • 1: Deep Sleep (content preserved)
8	RAM_40 (4 kB)	• 0: Shut Down (content not preserved) • 1: Deep Sleep (content preserved)
9	RAM_41 (4 kB)	• 0: Shut Down (content not preserved) • 1: Deep Sleep (content preserved)
10	RAM_42 (4 kB)	• 0: Shut Down (content not preserved) • 1: Deep Sleep (content preserved)
11	RAM_43 (4 kB)	• 0: Shut Down (content not preserved) • 1: Deep Sleep (content preserved)

Table 16. Parameter sram_retention_ctrl*...continued*

Bit	SRAM instance (SRAM size)	Value
31:12	Reserved	Must be set to 0

Table 17. Parameter wakeup_interrupts[0] (1st vector)

Bit	Symbol	Description	Value
1:0	-	Reserved	Must be set to 0
2	GLOBALINT0	Group GPIO input interrupt 0 (GINT0)	• 0: Interrupt disabled • 1: Interrupt enabled
3	GLOBALINT1	Group GPIO input interrupt 1 (GINT1)	• 0: Interrupt disabled • 1: Interrupt enabled
16:4	-	Reserved	Must be set to 0
17	FLEXCOMM3	Any module inside Flexcomm3 (UART, SPI, I2C)	• 0: Interrupt disabled • 1: Interrupt enabled
23:18	-	Reserved	Must be set to 0
24	ACMP	Analog comparator	• 0: Interrupt disabled • 1: Interrupt enabled
28:25	-	Reserved	Must be set to 0
29	RTC	Real-time clock (counter event and tamper event)	• 0: Interrupt disabled • 1: Interrupt enabled
31:30	-	Reserved	Must be set to 0

Table 18. Parameter wakeup_interrupts[1] (2nd vector)

Bit	Symbol	Description	Value
5:0	-	Reserved	Must be set to 0
6	OSTIMER	OS event timer	• 0: Interrupt disabled • 1: Interrupt enabled
31:7	-	Reserved	Must be set to 0

3.1.4 POWER_EnterDeepPowerDown

The `POWER_EnterDeepPowerDown` API configures the Deep power-down low-power mode. It allows controlling which peripherals are powered up and which SRAM instances are in retention state during the Deep power-down.

The `POWER_EnterDeepPowerDown` API switches the CPU and System Bus clock to 12 MHz. It is the responsibility of the user to reconfigure the CPU and System Bus clock after the function has returned (this happens when the Deep power-down state is not entered because an RTC or OS Event Timer interrupt is pending when the API is called. Otherwise, this API never returns).

Table 19. POWER_EnterDeepPowerDown

Routine	API Description
Function Prototype	`void POWER_EnterDeepPowerDown (uint32_t exclude_from_pd[1], uint32_t sram_retention_ctrl, uint32_t wakeup_interrupts[2], uint32_t wakeup_io_ctrl);`

Table 19. POWER_EnterDeepPowerDown*...continued*

Routine	API Description
	Input Parameters
`exclude_from_pd[1]`	Defines which analog peripherals shall not be powered down. It is a 1 x 32-bit vector, with each bit of the vector corresponding to one module. The definition of the vector is given in Table 20, and is aligned with the `power_pd_bit_t` type definition. For each bit field of the vector: • 0: The module is powered down during Deep power-down. • 1: The module is running during Deep power-down.
sram_retention_ctrl	Defines which SRAM instances shall preserve their content during Deep power-down. The `sram_retention_ctrl` parameter is a 32-bit vector, with each bit of the vector corresponding to one SRAM instance. The definition of the vector is given in Table 21, and is aligned with the `power_sram_bit_t` type definition. For each bit field of the vector: • 0: During Deep power-down, the SRAM instance is in the SRAM Shut Down low-power mode (content not preserved). • 1: During Deep power-down, the SRAM instance is in the SRAM Deep deep low-power mode (content preserved). ***Note:*** *During SoC boot time, if the FlexSPI or 1-bit SPI is used to load an executable code image from an external flash to internal SRAM, then only the following SRAM instances can be in the SRAM Deep sleep low-power mode (content preserved) during Deep power-down:* `RAM_00` *and* `RAM_01`*. Otherwise, the following SRAM instances can be in the SRAM Deep Sleep low-power mode (content preserved) during Deep power-down:* `RAM_00`*,* `RAM_01`*,* `RAM_10`*,* `RAM_20`*,* `RAM_30`*,* `RAM_40`*,* `RAM_41`*,* `RAM_42`*, and* `RAM_43`*. If VDDMAIN power is removed during Deep power-down mode, only* `RAM_00` *SRAM instance can be in the SRAM Deep Sleep low-power mode (content preserved) during the Deep power-down mode.*
wakeup_interrupts[2]	Defines which peripheral interrupts can be executed if they are pending before entering Deep power-down. It is a 2 x 32-bit vector, with each bit inside the vector corresponding to one interrupt source. The definition of the vectors is given in Table 22 and Table 23, and is aligned with the low-power modes wake-up sources (`#define WAKEUP_*`) #defines definitions. For each bit field of the vector: • 0: The peripheral interrupt handler is not executed, if the interrupt is pending before entering Deep power-down. • 1: The peripheral interrupt handler is executed, if the interrupt is pending before entering Deep power-down. Unlike Deep sleep and Power-down modes, in the Deep power-down mode, the `wakeup_interrupts` parameter has no influence on the ability of the peripheral (RTC or OS Event Timer) to wake up the device from the Deep power-down mode. That is, even if a peripheral interrupt is not enabled via the `wakeup_interrupts` parameter (`wakeup_interrupts[0] = 0` and `wakeup_interrupts[1] = 0`), the device wakes up from Deep power-down when the peripheral's wake-up event occurs (RTC countdown or tamper events, OS Event Timer countdown). ***Note:*** *After wake-up from Deep power-down, the RTC interrupt (if the RTC module is enabled and was the wake-up source) is always pending on the Cortex-M33 Nested Vector Interrupt Controller, but it is executed only if the user enables it directly in the Cortex-M33 Nested Vector Interrupt Controller (after the wake up).* *After wake up from Deep power down, the OS Event Timer interrupt is never pending on to the Cortex-M33 Nested Vector Interrupt Controller even if it is enabled via this* `wakeup_interrupts` *prior entering Deep power-down mode.*
wakeup_io_ctrl	Configures the five wake-up pins that can wake up the device from the Deep power-down mode. The definition of the parameter is given in Table 24, and it is aligned with the wake-up I/O sources (`#define LOWPOWER_WAKEUPIO*`) #define definitions. ***Note:*** *During the Deep power-down mode, the five wake-up I/Os are all configured as Input. The user can choose to enable or disable the internal pull-up and pull-down for each wake-*

Table 19. POWER_EnterDeepPowerDown*...continued*

Routine	API Description
	up I/O individually. After wake up from Deep power-down, the five wake-up I/Os retrieve their default configuration (Input and output disabled, both internal pull-up and pull-down disabled) when the `POWER_PowerInit` API is called.
Result	None
Description	Configures the Deep power-down low-power mode. It allows controlling which peripherals are powered up and which SRAM instances are in retention state during Deep power-down. Also, it defines which peripheral interrupts can be executed if they are pending before entering Deep power-down.

Table 20. Parameter exclude_from_pd[0] definition

Bit	Symbol	Description	Value
3:0	-	Reserved	Must be set to 0
4	FRO1M	1 MHz Free Running Oscillator	• 0: Powered down • 1: Running
5	-	Reserved	Must be set to 0
6	FRO32K	32 kHz Free Running Oscillator	• 0: Powered down • 1: Running
7	XTAL32K	32 kHz Crystal Oscillator	• 0: Powered down • 1: Running
31:8	-	Reserved	Must be set to 0

Table 21. Parameter sram_retention_ctrl

Bit	SRAM instance (SRAM size)	Value
0	Reserved	Must be set to 0
1	RAM_00 (4 kB)	• 0: Shut Down (content not preserved) • 1: Deep sleep (content preserved)
2	RAM_01 (4 kB)	• 0: Shut Down (content not preserved) • 1: Deep sleep (content preserved)
4:3	Reserved	Must be set to 0
5	RAM_10 (16 kB)[1][1]	• 0: Shut Down (content not preserved) • 1: Deep sleep (content preserved)
6	RAM_20 (32 kB) [2][1][2]	• 0: Shut Down (content not preserved) • 1: Deep sleep (content preserved)
7	RAM_30 (32 kB)[1][2]	• 0: Shut Down (content not preserved) • 1: Deep sleep (content preserved)
8	RAM_40 (4 kB)[1][2]	• 0: Shut Down (content not preserved) • 1: Deep sleep (content preserved)
9	RAM_41 (4 kB)[1][2]	• 0: Shut Down (content not preserved) • 1: Deep sleep (content preserved)
10	RAM_42 (4 kB)[1][2]	• 0: Shut Down (content not preserved) • 1: Deep sleep (content preserved)

Table 21. Parameter sram_retention_ctrl*...continued*

Bit	SRAM instance (SRAM size)	Value
11	RAM_43 (4 kB)[1][2]	• 0: Shut Down (content not preserved) • 1: Deep sleep (content preserved)
31:12	Reserved	Must be set to 0

[1] If VDDMAIN power is removed during Deep power-down mode, this SRAM instance cannot be in SRAM Deep sleep low-power mode (content preserved) during Deep power-down mode.

[2] During SoC Boot time, If the FlexSPI or a 1-bit SPI are used to load an executable code image from an external flash to internal SRAM, then this SRAM instance cannot be in SRAM Deep sleep low-power mode (content preserved) during Deep power-down.

Table 22. Parameter wakeup_interrupts[0] (1st vector)

Bit	Symbol	Description	Value
28:0	-	Reserved	Must be set to 0
29	RTC	Real Time Clock (counter event and tamper event)	• 0: Interrupt disabled • 1: Interrupt enabled
31:30	-	Reserved	Must be set to 0

Table 23. Parameter wakeup_interrupts[1] (2nd vector)

Bit	Symbol	Description	Value
5:0	-	Reserved	Must be set to 0
6	OSTIMER	OS Event Timer	• 0: Interrupt disabled • 1: Interrupt enabled
31:7	-	Reserved	Must be set to 0

Table 24. Parameter wakeup_io_ctrl

Bit	Symbol	Description	Value
1:0	MODEWAKEUP0	Select wake-up pin 0 operating mode during Deep power-down.	• 0: Disabled • 1: Rising edge • 2: Falling edge • 3: Rising and falling edges
3:2	MODEWAKEUP1	Select wake-up pin 1 operating mode during Deep power-down.	• 0: Disabled • 1: Rising edge • 2: Falling edge • 3: Rising and falling edges
5:4	MODEWAKEUP2	Select wake-up pin 2 operating mode during Deep power-down.	• 0: Disabled • 1: Rising edge • 2: Falling edge • 3: Rising and falling edges
7:6	MODEWAKEUP3	Select wake-up pin 3 operating mode during Deep power-down.	• 0: Disabled • 1: Rising edge • 2: Falling edge • 3: Rising and falling edges
9:8	MODEWAKEUP4	Select wake-up pin 4 operating mode during Deep power-down.	• 0: Disabled • 1: Rising edge

Table 24. Parameter wakeup_io_ctrl*...continued*

Bit	Symbol	Description	Value
			• 2: Falling edge • 3: Rising and falling edges
19:10	-	Reserved	Must be set to 0
20	DISABLEPULLUPDOWN WAKEUP0	Disable both pull-up and pull-down for wake-up pin 0 during Deep power-down. When the wake-up pin 0 is enabled (MODEWAKEUP0 is not 0) and pull-up or pull-down is enabled (DISABLEPULLUPDOWNWAKEUP0=0), then the state of the pull-up and pull-down is determined by the wake-up pin mode (and not by the field PULLUPDOWNWAKEUP0) as described in Table 25.	• 0: Enable pull-up or pull-down (see PULLUPDOWNWAKEUP 0) • 1: Disable both pull-up and pull-down
21	DISABLEPULLUPDOWN WAKEUP1	Disable both pull-up and pull-down for wake-up pin 1 during Deep power-down. When wake-up pin 1 is enabled (MODEWAKEUP1 is not 0) and pull-up or pull-down is enabled (DISABLEPULLUPDOWNWAKEUP1=0), then the state of the pull-up and pull-down is determined by the wake-up pin mode (and not by the field PULLUPDOWNWAKEUP1) as described in Table 25.	• 0: Enable pull-up or pull-down (see PULLUPDOWNWAKEUP 0) • 1: Disable both pull-up and pull-down
22	DISABLEPULLUPDOWN WAKEUP2	Disable both pull-up and pull-down for the wake-up pin 2 during Deep power-down. When wake-up pin 2 is enabled (MODEWAKEUP2 is not 0) and pull-up or pull-down is enabled (DISABLEPULLUPDOWNWAKEUP2=0), then the state of the pull-up and pull-down is determined by the wake-up pin mode (and not by the field PULLUPDOWNWAKEUP2) as described in Table 25.	• 0: Enable pull-up or pull-down (see PULLUPDOWNWAKEUP 0) • 1: Disable both pull-up and pull-down
23	DISABLEPULLUPDOWN WAKEUP3	Disable both pull-up and pull-down for the wake-up pin 3 during Deep power-down. When wake-up pin 3 is enabled (MODEWAKEUP3 is not 0) and pull-up or pull-down is enabled (DISABLEPULLUPDOWNWAKEUP3=0), then the state of the pull-up and pull-down is determined by the wake-up pin mode (and not by the field PULLUPDOWNWAKEUP3) as described in Table 25.	• 0: Enable pull-up or pull-down (see PULLUPDOWNWAKEUP 0) • 1: Disable both pull-up and pull-down
24	DISABLEPULLUPDOWN WAKEUP4	Disable both pull-up and pull-down for wake-up pin 4 during Deep power-down. When wake-up pin 4: • is enabled (MODEWAKEUP4 is not 0) AND • pull-up or pull-down is enabled (DISABLEPULLUPDOWNWAKEUP4=0) then the state of the pull-up and pull-down is determined by the wake-up pin mode (and not by the field PULLUPDOWNWAKEUP4) as described in Table 25.	• 0: Enable pull-up or pull-down (see PULLUPDOWNWAKEUP 0) • 1: Disable both pull-up and pull-down
25	PULLUPDOWNWAKEUP0	Select pull-up or pull-down for wake-up pin 0 during Deep power-down. This field has effect only when wake-up pin 0 is disabled	• 0: Pull-down • 1: Pull-up

Table 24. Parameter wakeup_io_ctrl*...continued*

Bit	Symbol	Description	Value
		(MODEWAKEUP0=0) and when wake-up pin 0 pull-up/pull-down are enabled (DISABLEPULLUPDOWNWAKEUP0=0)	
26	PULLUPDOWNWAKEUP1	Select pull-up or pull-down for wake-up pin 1 during Deep power-down. This field has effect only when wake-up pin 1 is disabled (MODEWAKEUP1=0) and when wake-up pin 1 pull-up/pull-down are enabled (DISABLEPULLUPDOWNWAKEUP1=0)	• 0: Pull-down • 1: Pull-up
27	PULLUPDOWNWAKEUP2	Select pull-up or pull-down for the wake-up pin 2 during Deep power-down. This field has effect only when the wake-up pin 2 is disabled (MODEWAKEUP2=0) and when wake-up pin 2 pull-up/pull-down are enabled (DISABLEPULLUPDOWNWAKEUP2=0)	• 0: Pull-down • 1: Pull-up
28	PULLUPDOWNWAKEUP3	Select pull-up or pull-down for the wake-up pin 3 during Deep power-down. This field has effect only when the wake-up pin 3 is disabled (MODEWAKEUP3=0) and when the wake-up pin 3 pull-up/pull-down are enabled (DISABLEPULLUPDOWNWAKEUP3=0)	• 0: Pull-down • 1: Pull-up
29	PULLUPDOWNWAKEUP4	Select pull-up or pull-down for the wake-up pin 4 during Deep power-down. This field has effect only when the wake-up pin 4 is disabled (MODEWAKEUP4=0) and when the wake-up pin 4 pull-up/pull-down are enabled (DISABLEPULLUPDOWNWAKEUP4=0)	• 0: Pull-down • 1: Pull-up
31:30	-	Reserved	Must be set to 0

Table 25. Pull-up/pull-down configuration

User Configuration			Resulting Pull-up/pull-down State		Comments
DISABLEPULLUPDOWNWAKEUPx	MODEWAKEUPx	PULLUPDOWNWAKEUPx	Pull-down state	Pull-up state	-
0	0 (Disabled)	0 (pull-down)	Enabled	Disabled	The wake-up pin is disabled (MODEWAKEUP = 0) and both the pull-up/pull-down are not disabled (DISABLEPULLUPDOWNWAKEUP=0). In this case, the pull-up and pull-down states depend on the value of PULLUPDOWNWAKEUPx.
0	0 (Disabled)	1 (pull-up)	Disabled	Enabled	
0	1 (Rising edge)	-	Enabled	Disabled	The wake-up pin is enabled (MODEWAKEUP != 0) and both the pull-up/pull-down are not disabled (DISABLEPULLUPDOWNWAKEUP=0). In this case, the pull-up and pull-down states depend only on the value of MODEWAKEUPx
0	2 (Falling edge)	-	Disabled	Enabled	
0	3 (Rising and falling edges)	-	Enabled	Disabled	
1	-	-	Disabled	Disabled	Both the pull-up and the pull-down are disabled, whatever the values of MODEWAKEUP and PULLUPDOWNWAKEUP. Use this configuration when an external pullup or pull-down resistor is connected on the wake-up pin.

3.2 Sleep mode

Sleep mode saves some power by stopping Cortex-M33 CPU execution without affecting peripherals or requiring significant wake-up time. The clock to the CPU is shut off. Peripherals and memories are active and

operational. The CPU is stopped and execution of instructions is suspended until either a reset or an interrupt occurs.

Peripheral functions, if selected to be clocked in the AHBCLKCTRL registers, continue operation during Sleep mode and can generate interrupts to cause the processor to resume execution. Sleep mode only eliminates dynamic power used by the processor itself, memory systems and related controllers, and internal buses. The processor state and registers, peripheral registers, and internal SRAM values are maintained.

3.2.1 Entering Sleep mode

The `POWER_EnterSleep()` API is used to enter the Sleep mode.

```
case kPmu_Sleep: /* Enter sleep mode. */
    POWER_EnterSleep();
    break;
```

3.2.2 Waking up from Sleep mode

In this application, a group interrupt generated by the SW3 (USR) switch is used to wake up the device from the Sleep mode. Pin 17 from the Port 0 (`P0_17`) is used as the source of a group interrupt and this pin is pulled low by the pressing the SW3 (USR) switch.

3.3 Deep sleep mode

In Deep sleep mode, the full device remained powered, but flash and ROM are shut down, with the cost of a longer wake-up time compared to the Sleep mode. The system clock to the CPU is disabled as in Sleep mode. Analog blocks are powered down by default but can be selected to keep running through the power API if needed as wake-up sources. The main clock and all peripheral clocks are disabled.

Deep sleep mode eliminates power used by analog peripherals and all dynamic power used by the CPU, its memory systems and related controllers, and internal buses. The CPU state and registers, peripheral registers, and internal SRAM values are maintained, and the GPIO logic levels of the pins remain static. All SRAM instances that are not configured to enter in SRAM Deep sleep low-power mode keeps the state they had before entering Deep sleep. This means that if a SRAM was in Active state before entering in Deep sleep, it stays in Active state during Deep sleep and therefore it consumes more power.

Selected peripherals such as GPIO group interrupts (GINT), USB Full Speed, Flexcomms (SPI, I2C, USART), Windowed-Watchdog Timer, RTC, standard Counter/Timers, Micro-tick, RTC alarm, a watchdog timer interrupt/reset, BOD interrupt/reset and comparator can be left running in Deep sleep mode. The oscillators like the FRO1M, FRO32K, XTAL32K but also the FRO192M (which delivers the 12 MHz and 96 MHz clocks), can also be left running.

Some peripherals can have DMA service during Deep sleep mode without waking up the entire device.

3.3.1 Entering Deep sleep mode

The `POWER_EnterDeepSleep()` API is used to enter the Deep sleep mode.

In this application, `RAM_X0` (16 kB), `RAM_00` (4 kB), `RAM_01` (4 kB), `RAM_02` (4 kB), `RAM_03` (4 kB), `RAM_10` (32 kB), `RAM_20` (32 kB), `RAM_30` (16 kB), `RAM_40` (4 kB), `RAM_41` (4 kB), `RAM_42` (4 kB), `RAM_43` (4 kB) are in SRAM Deep Sleep low-power mode during Deep sleep.

```
case kPmu_Deep_Sleep: /* Enter deep sleep mode. */
    POWER_EnterDeepSleep(APP_EXCLUDE_FROM_DEEPSLEEP, 0x0, APP_WAKEUP_FROM_DEEPSLEEP, 0x0);
    break;
```

3.3.2 Waking up from Deep sleep mode

In this application, a pin interrupt generated by the SW1 (WK_TMP) switch is used to wake up the device from the Deep sleep mode.

Pin 18 from Port 1 (P1_18) is used as the source of a pin interrupt and this pin is pulled low by pressing the SW1 (WK_TMP) switch. The same interrupt is enabled in NVIC using the EnableDeepSleepIRQ() API.

3.4 Power-down mode

Power-down mode turns off nearly all on-chip power consumption by eliminating power used by all analog modules and by shutting down the DCDC and the LDO_CORE. This eliminates all digital peripherals power, with the cost of a longer wake-up time compared to Deep sleep mode. FRO192M is disabled. The flash memory is also disabled. The clock to the CPU and peripherals is shut down, and if not configured, the peripherals in the power domains PD_SYSTEM and PD_AO receive no internal clocks. All SRAM can be configured to maintain their internal state, and all registers lose their internal states except those located in the power domains PD_SYSTEM and PD_AO. Any SRAM instance not configured to maintain its internal state loses it. The Cortex-M33 CPU state and some critical peripherals like the analog controller are retained. The GPIO logic level does not remain static in Power-down mode. All GPIO pin states are logic '0' in Power-down mode. All IOCON registers and peripheral registers related ONLY to Flexcomm3 (SPI, I2C, I2S, USART), GINTO0, RTC, OS event timer, and analog comparator maintain state in Power-down mode. GPIO group interrupts (GINT0 and GINT1), selected serial peripherals in Flexcomm3 (SPI, I2C, USART), RTC, OS event timers, and analog comparator can be left running to wake up the device.

When a wake-up event occurs, the Cortex-M33 CPU code execution resumes from where it has stopped. It is the responsibility of the customer application to re-configure all modules in the power domain core PD_CORE (whose states have not been retained, for example, all Flexcomm -except Flexcomm3-, SDMA, Power Quad, DMIC, and so on.).

3.4.1 Entering Power-down mode

The POWER_EnterPowerDown() API is used to enter the Power-down mode. In this application, by default, RAM_00 (4 kB), RAM_43 (4 kB) preserve their content during power down.

By changing #define PD_WAKE_UP_SRAM24K from 0 to 1, RAM_X0 (16 kB), RAM_00 (4 kB), RAM_43 (4 kB) preserve their content during power down.

By changing #define PD_WAKE_UP_SRAM128K from 0 to 1, RAM_X0 (16 kB), RAM_00 (4 kB), RAM_01 (4 kB), RAM_02 (4 kB), RAM_03 (4 kB), RAM_10 (32 kB), RAM_20 (32 kB), RAM_30 (16 kB), RAM_40 (4 kB), RAM_41 (4 kB), RAM_42 (4 kB), RAM_43 (4 kB) preserve their content during power down.

3.4.2 Waking up from Power-down mode

In this application, a group interrupt or a reset is used to wake up the device from the Power-down mode.

- Pin 17 from the Port 0 (P0_17) is used as the source of a group interrupt and this pin is pulled low by pressing the SW3 (USR) switch.
- A reset from the RESETN pin is generated by pressing the SW2 (RESET) switch.

3.5 Deep power-down mode

Deep power-down mode shuts down virtually all on-chip power consumption but requires a longer wake-up time (compared to Power-down mode). For maximal power savings, the entire power domains PD_CORE (CPU, DSP, Flexcomms, SDMA, and so on) and PD_SYSTEM (Reset, Clocks, SYSCON, IOCON, and so on) are shut down. Only the Always-on power domain PD_AO (PMU, PMC, the RTC, and the OS event timer) stays powered. Clocks are shut off to the entire chip device except for the RTC and the OS event timer if needed.

On wake-up, the device reboots. During Deep power-down mode, the contents of some SRAM can be retained (software configured via the POWER_EnterDeepPowerDown() low-power API), but registers (other than those in the PMC, the RTC, and OS event timer) are not retained. All functional pins are tri-stated in Deep power-down mode, except the 5 wake-up and RESET pins.

3.5.1 Entering Deep power-down mode

The `POWER_EnterDeepPowerDown()` API is used to enter the Deep power-down mode.

```
            case kPmu_Deep_PowerDown: /* Enter deep power down mode. */
#if DPD_WAKE_UP_SRAM4K
#if DPD_WAKE_UP_PIN
                POWER_EnterDeepPowerDown(APP_EXCLUDE_FROM_DEEPPOWERDOWN, kPOWER_SRAM_RAM_00,
                                         APP_WAKEUP_FROM_DEEPPOWERDOWN, wakeup_io_ctrl);
#else
                POWER_EnterDeepPowerDown(APP_EXCLUDE_FROM_DEEPPOWERDOWN, kPOWER_SRAM_RAM_00,
                                         APP_WAKEUP_FROM_DEEPPOWERDOWN, 0);
#endif
#else
#if DPD_WAKE_UP_PIN
                POWER_EnterDeepPowerDown(APP_EXCLUDE_FROM_DEEPPOWERDOWN, 0,
                                         APP_WAKEUP_FROM_DEEPPOWERDOWN, wakeup_io_ctrl);
#else
                POWER_EnterDeepPowerDown(APP_EXCLUDE_FROM_DEEPPOWERDOWN, 0,
                                         APP_WAKEUP_FROM_DEEPPOWERDOWN, 0);
#endif
#endif
                break;
```

3.5.2 Waking up from Deep power-down mode

Wake up from the Deep power-down mode can be done via the wake-up pin, or the `RESETN` pin. Pin 18 from Port 1 (`P1_18`) is used as the source of the wake-up pin and this pin is pulled low by pressing the SW1 (`WK_TMP`) switch. A reset from the `RESETN` pin by pressing SW2 (RESET) switch is used to wake up the device from the Deep power-down mode.

4 Low-power mode Demo

This section explains how LPC553x/LPC55S3x can be put into and waken up from the low-power modes.

- To wake up the device from the Sleep mode, configure the GPIO pin (`P0_17`) (SW3 (USR)) as the source of a group interrupt.
- To wake up the device from the Deep sleep mode, configure the GPIO pin (`P1_18`) (SW1 (WK_TMP)) as the source of a pin interrupt.
- To wake up the device from the Power-down mode, configure the GPIO pin (`P0_17`) (SW3 (USR)) as the source of a group interrupt.
- To wake up the device from the Power-down or Deep power-down mode, assert reset from the `RESETN` pin by pressing the SW2 (RESET) switch.
- To wake up the device from the Deep power-down mode, configure the GPIO pin (`P1_18`) (SW1 (WK_TMP)) as the source of a wake-up pin.

4.1 Software setup

Download and extract the file (`SDK_2_14_0_LPCXpresso55S36.zip`) associated with this Application Note. You can use following IDEs to verify the demo application available at `SDK_2_14_0_LPCXpresso55S36\boards\lpcxpresso55s36\demo_apps\power_mode_switch_lpc`

- Keil IDE version 5.38 (+ NXP.LPC55S36_DFP.17.0.0.pack)
- IAR IDE 9.40.1
- MCUXpresso IDE 11.8.0 (available for download at https://www.nxp.com/design/software/development-software/mcuxpresso-software-and-tools-/mcuxpresso-integrated-development-environment-ide:MCUXpresso-IDE)

For more details on building and downloading the application, refer *Getting Started with MCUXpresso SDK for LPCXpresso55S36.pdf* located at `\SDK_2_14_0_LPCXpresso55S36`.

4.2 Hardware setup

The hardware requirements for the LPC553x/LPC55S3x low-power consumption measurement are as follows:

- LPC553x/LPC55S3x evaluation board
- Digital multimeter for current measurement
- Oscilloscope for wake-up time measurement

For the LPC553x/LPC55S3x low-power consumption measurement, install jumper top to JP33 (R432 as DNP), JP74, and JP75.

- To measure `IMCU_MAIN`, remove jumper top from JP28 (R436 as DNP) and connect current meter to JP28 (R436 as DNP).
- To measure `IMCU_VBAT`, remove jumper top from JP22 (R435 as DNP) and connect current meter to JP22 (R435 as DNP).
- To measure `IMCU_VDDA`, remove jumper top from JP21 (R434 as DNP) and connect current meter to JP21 (R434 as DNP).
- To measure `IMCU_VDD`, remove jumper top from JP20 (R433 as DNP) and connect current meter to JP20 (R433 as DNP).

4.3 Measuring power in low-power mode

To measure power in the low-power mode:

1. When using UART with FTDI:
   a. Install jumper on JP26.
   b. Remove jumper from JP12.
   c. Connect FTDI cable to JP3, as shown in Figure 2.
   d. Connect current meter to JP28 (`IMCU_MAIN`), JP22 (`IMCU_VBAT`), JP21 (`IMCU_VDDA`), JP20 (`IMCU_VDD`) as shown in Figure 1.
   e. Connect power from J2 to PC through USB cable, as shown in Figure 2.
2. When using USB:
   a. Remove jumper from JP26.
   b. Install jumper on JP12.
   c. Use the LPC-Link II UCom port.
   d. Connect current meter to JP28 (`IMCU_MAIN`), JP22 (`IMCU_VBAT`), JP21 (`IMCU_VDDA`), JP20 (`IMCU_VDD`) as shown in Figure 1.
   e. Connect power and MCU-Link onboard debugger from J1 to PC through USB cable, as shown in Figure 2.

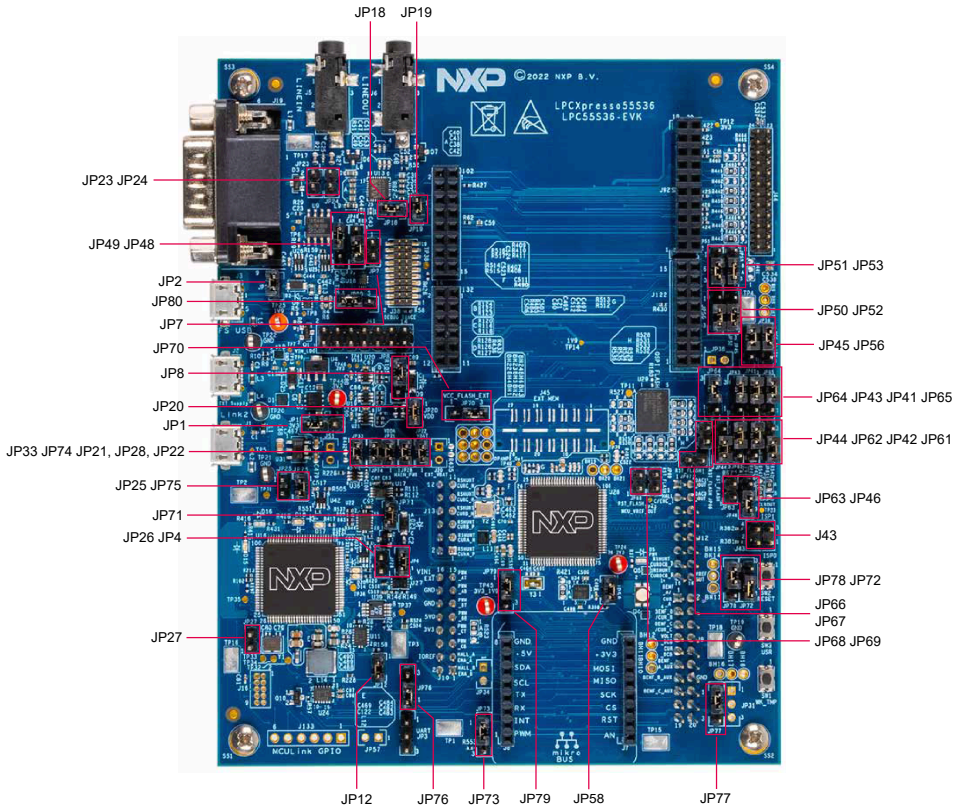

Figure 1. Jumpers

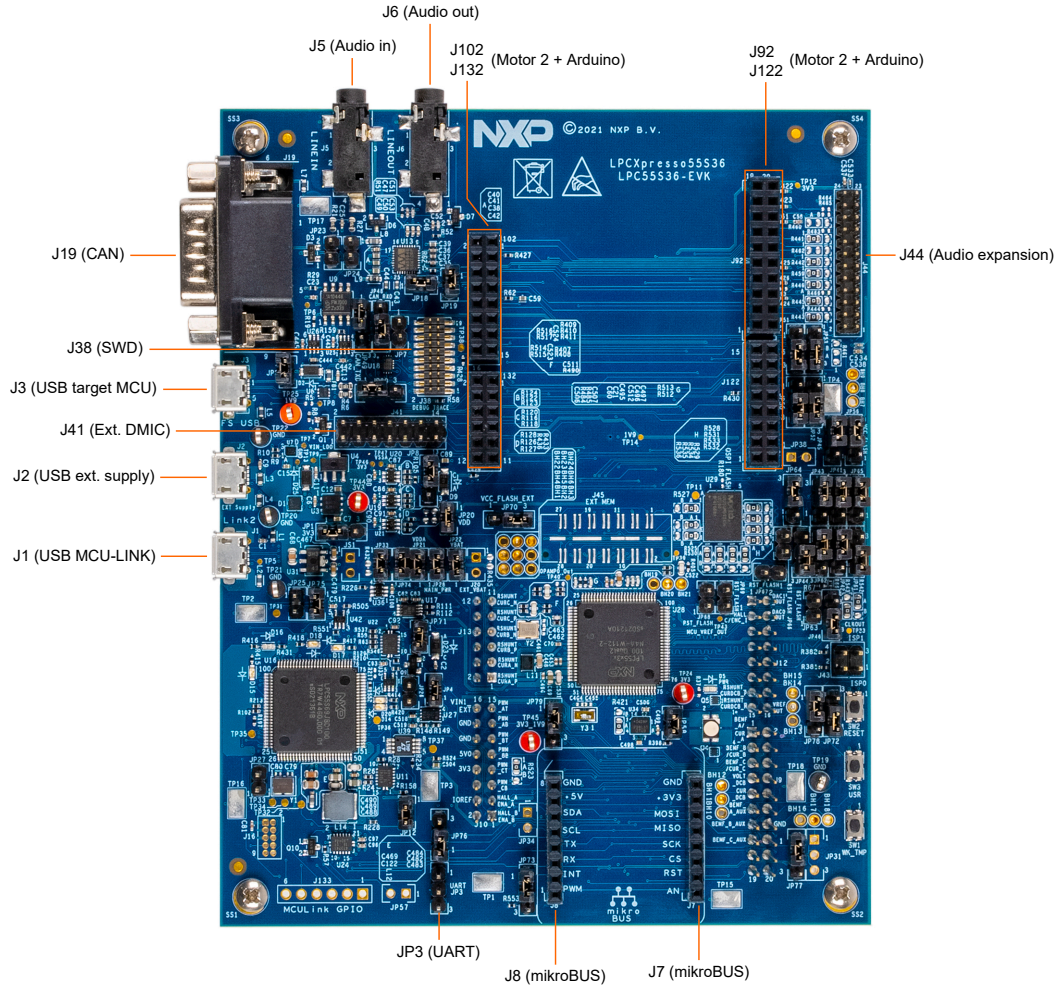

Figure 2. Connectors

3. Launch a serial terminal application, such as TeraTerm.
4. Select the LPC553x/LPC55S3x serial port and configure the following settings for the serial console:
   - 115200 baud rate
   - No parity
   - 8 data bits
   - 1 stop bits
5. Reset the board using the SW2 (RESET) switch. The following output appears at the serial terminal.

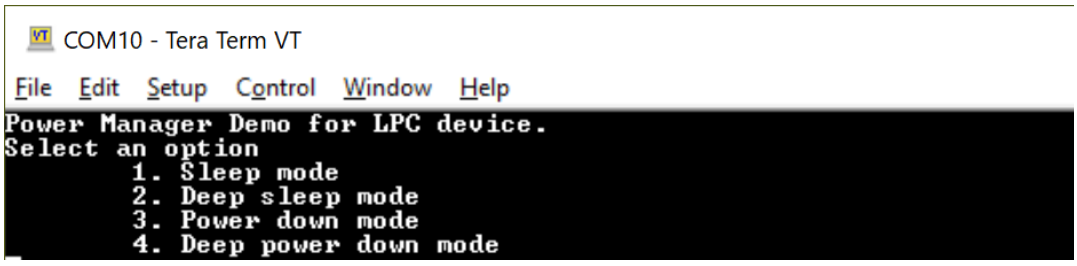

Figure 3. Console window

6. Take current measurement from the current meter.
   Note: *For details about waking up the device from various low-power modes, see Section 4.*
7. Repeat steps 5 - 7 to take next measurements after the device wakes up from the low-power mode.

4.4 Measuring wake-up time

The wake-up time for the Sleep mode can be measured between the I/O line (P0_17) pulled low (press SW3 (USR) switch to wake up LPC553x/LPC55S3x) and the I/O line (P0_22) going high, as this I/O line is pulled high at the second instruction inside the group (SW3) interrupt service routine.

The wake-up time for the Deep sleep mode can be measured between the I/O line (P1_18) pulled low (on SW1 (WK_TMP) switch press to wake up the LPC553x/LPC55S3x) and the I/O line (P0_22) going high, as this I/O line is pulled high at the first instruction inside the pin (SW1) interrupt service routine.

The two wake-up sources for the LPC553x/LPC55S3x Power-down mode are as follows:

- **Group interrupt:** The wake-up time can be measured between I/O line (P0_17) being pulled low (press SW3 (USR) switch to wake up LPC553x/LPC55S3x) and I/O line (P0_22) going high, as this I/O line is pulled high at the second instruction inside the group (SW3) interrupt service routine.
- **RESETN pin:** LPC553x/LPC55S3x can wake up by providing low pulse to the RESETN pin, by pressing the SW2 (RESET) switch.
  The wake-up time can be measured between the RESETN pin going high (when we press and release SW2 (RESET) switch to wake up LPC553x/LPC55S3x) and I/O line (P0_22) going high, as this I/O line is pulled high at the 17th instruction inside the reset handler.

The two wake-up sources for the LPC553x/LPC55S3x deep Power down mode are as follows:

- **Wake-up pin:** The wake-up time can be measured between the I/O line (P1_18) pulled low (press SW1 (WK_TMP) switch to wake up LPC553x/LPC55S3x) and I/O line (P0_22) going high, as this I/O line is pulled high at the 17th instruction inside the reset handler.
- **RESETN pin:** LPC553x/LPC55S3x can wake up by providing low pulse to the RESETN pin, by pressing the SW2 (RESET) switch.

The wake-up time can be measured between the RESETN pin going high (when we press and release the SW2 (RESET) switch to wake up LPC553x/LPC55S3x) and the I/O line (P0_22) going high, as this I/O line is pulled high at the 17th instruction inside the reset handler.

4.4.1 Measuring wake-up time from Sleep mode and Deep sleep mode

To measure the wake-up time from the Sleep or Deep sleep mode:

1. Wake up LPC553x/LPC55S3x from the Sleep or Deep sleep mode:
   - To wake up from the Sleep mode, connect P0_17 (configured as the source of a group interrupt) to an oscilloscope probe as trigger input. P0_17 can be accessed from JP73.2.
   - To wake up from the Deep sleep mode, connect P1_18 (configured as the source of a pin interrupt) to an oscilloscope probe as trigger input. P1_18 can be accessed from R43, R46, C32, SW1 (WK_TMP).
2. Connect P0_22 to another oscilloscope probe. P0_22 can be accessed from J12.8.
3. Connect both oscilloscope probes with GND on TP19. Once the connections are done, power up the board by connecting the USB cable between J1 (if using USB with MCU-Link) or J2 (if using UART with FTDI) and PC.
4. Select the Sleep or Deep sleep mode as the low-power mode by entering 1 or 2 from the keyboard.
5. Wake up LPC553x/LPC55S3x from the Sleep or Deep sleep mode
   - To wake up from the Sleep mode, press the SW3 (USR) switch.
   - To wake up from the Deep sleep mode, press the SW1 (WK_TMP) switch.
6. Measure wake-up time:
   - For the Sleep mode, measure the time passed on the oscilloscope between the P0_17 wake-up pin going low to the P0_22 pin going high as shown in the following figure.

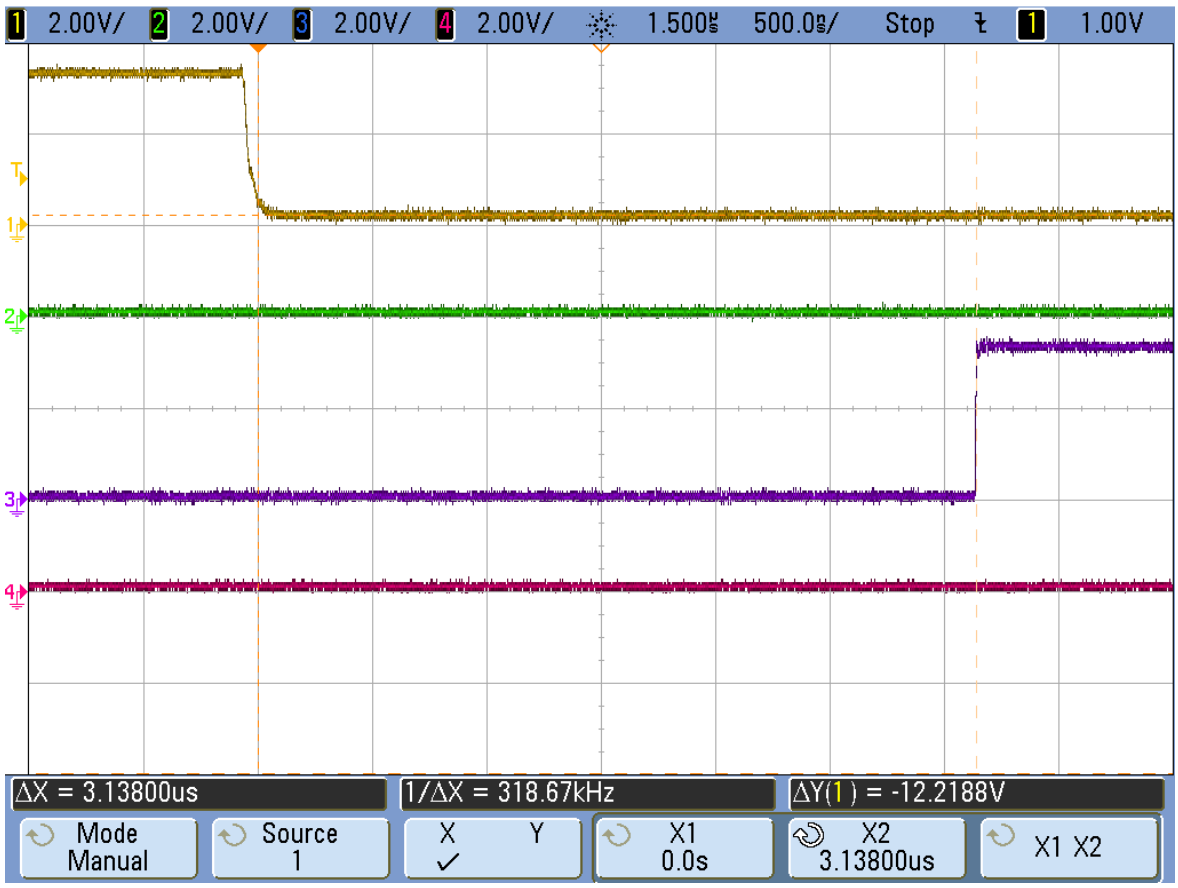

Figure 4. Wake-up time measurements for Sleep mode

- For the Deep sleep mode, measure the time passed on the oscilloscope between the `P1_18` wake-up pin going low to the `P0_22` pin going high as shown in the following figure.

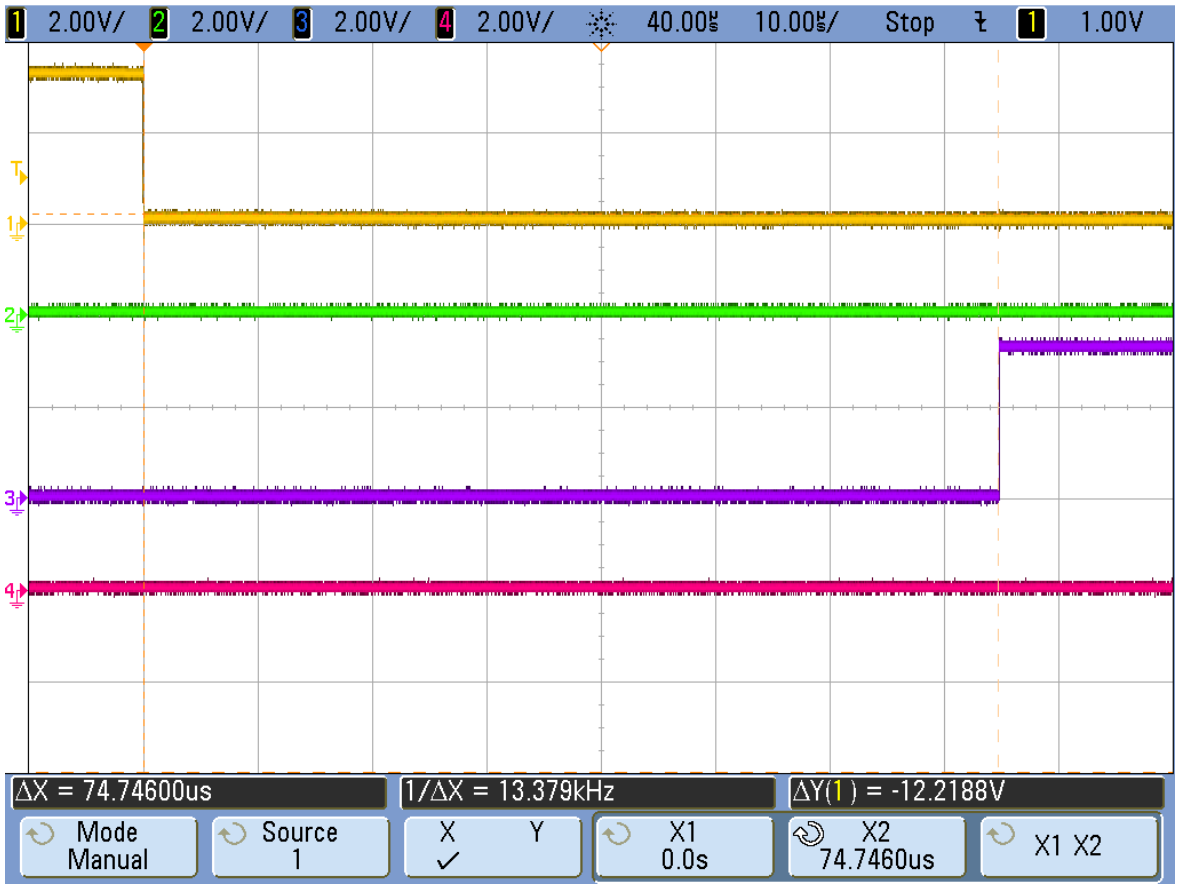

Figure 5. Wake-up time measurements for Deep sleep mode

4.4.2 Measuring wake-up time from Power-down mode and Deep power-down mode

To measure the wake-up time from the Power down or Deep power-down mode:

1. Wake up LPC553x/LPC55S3x from the Power-down or Deep power-down mode:
   - To wake up from the Power-down mode, connect `P0_17` (configured as the source of a group interrupt) to an oscilloscope probe as trigger input. `P0_17` can be accessed from JP73.2. Or, connect the `RESETN` pin to an oscilloscope probe as trigger input. The `RESETN` pin can be accessed from J102.6.
   - To wake up from the Deep power-down mode, connect `P1_18` (configured as the source of a wake up pin) to an oscilloscope probe as trigger input. `P1_18` can be accessed from R43, R46, C32, SW1 (`WK_TMP`). Or, connect the `RESETN` pin to an oscilloscope probe as trigger input. The `RESETN` pin can be accessed from J102.6.
2. Connect `P0_22` to another oscilloscope probe. `P0_22` can be accessed from J12.8.
3. Connect both oscilloscope probes with GND on TP19. Once the connections are done, power up the board by connecting the USB cable between J1 (if using USB with MCU-Link) or J2 (if using UART with FTDI) and PC.
4. Select the Power-down or Deep power-down mode as the low-power mode by entering 3 or 4 from the keyboard.
5. After entering 3 or 4, press any other key on the keyboard to confirm if LPC553x/LPC55S3x is in the Power-down or Deep power-down mode.
6. Wake up LPC553x/LPC55S3x from the Power-down or Deep power-down mode:
   - To wake up from the Power-down mode, use group interrupt (on SW3 (USR) switch press).

- To wake up from the Deep power-down mode, use the wake-up pin (on SW1 (WK_TMP) switch press).

7. Measure wake-up time:
   - For wake-up from the Power-down mode using group interrupt (on SW3 (USR) switch press), measure time passed on the oscilloscope between the `P0_17` wake-up pin going low to the `P0_22` pin going high as shown in the following figure.

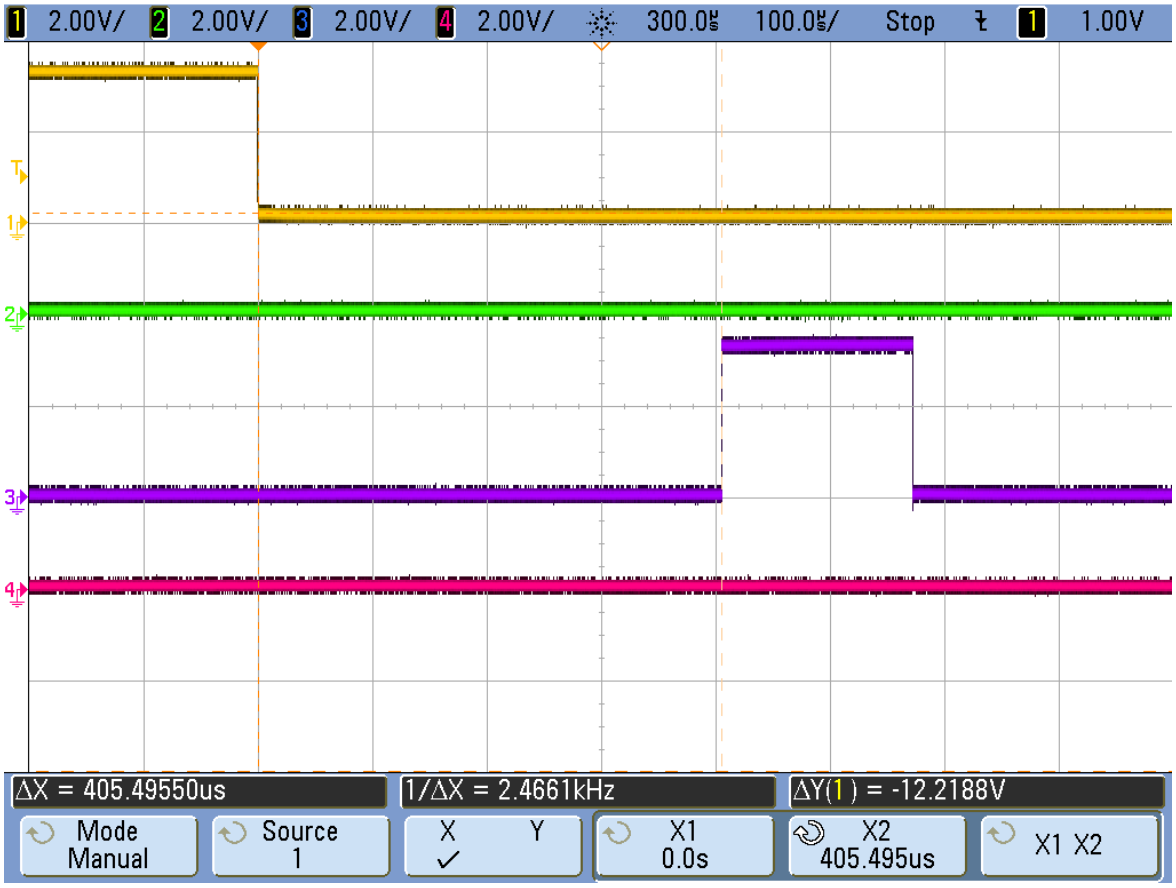

Figure 6. Wake-up time measurement for Power-down Mode using group interrupt

- For wake-up from the Deep power-down mode using the wake-up pin (on SW1 (WK_TMP) switch press), measure time passed on the oscilloscope between the `P0_18` wake-up pin going low to the `P0_22` pin going high as shown in the following figure.

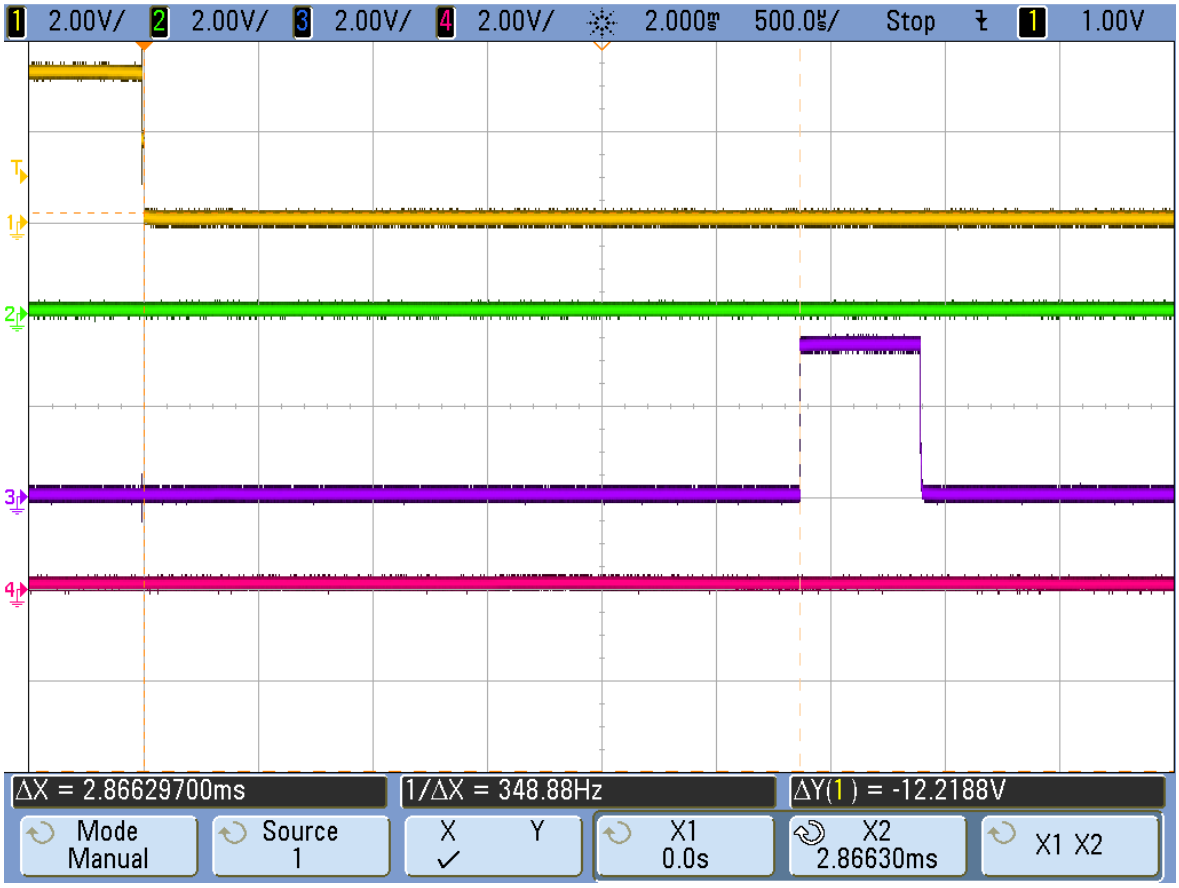

Figure 7. Wake-up time measurement for Deep power-down mode using wake-up pin

8. Measure wake-up time for wake-up from the Power-down or Deep power-down mode using the `RESETN` pin (on SW2 (RESET) switch press).
   - For wake up from the Power-down mode, measure time passed on the oscilloscope between the `RESETN` pin going high to the `P0_22` pin going high as shown in Figure 7.

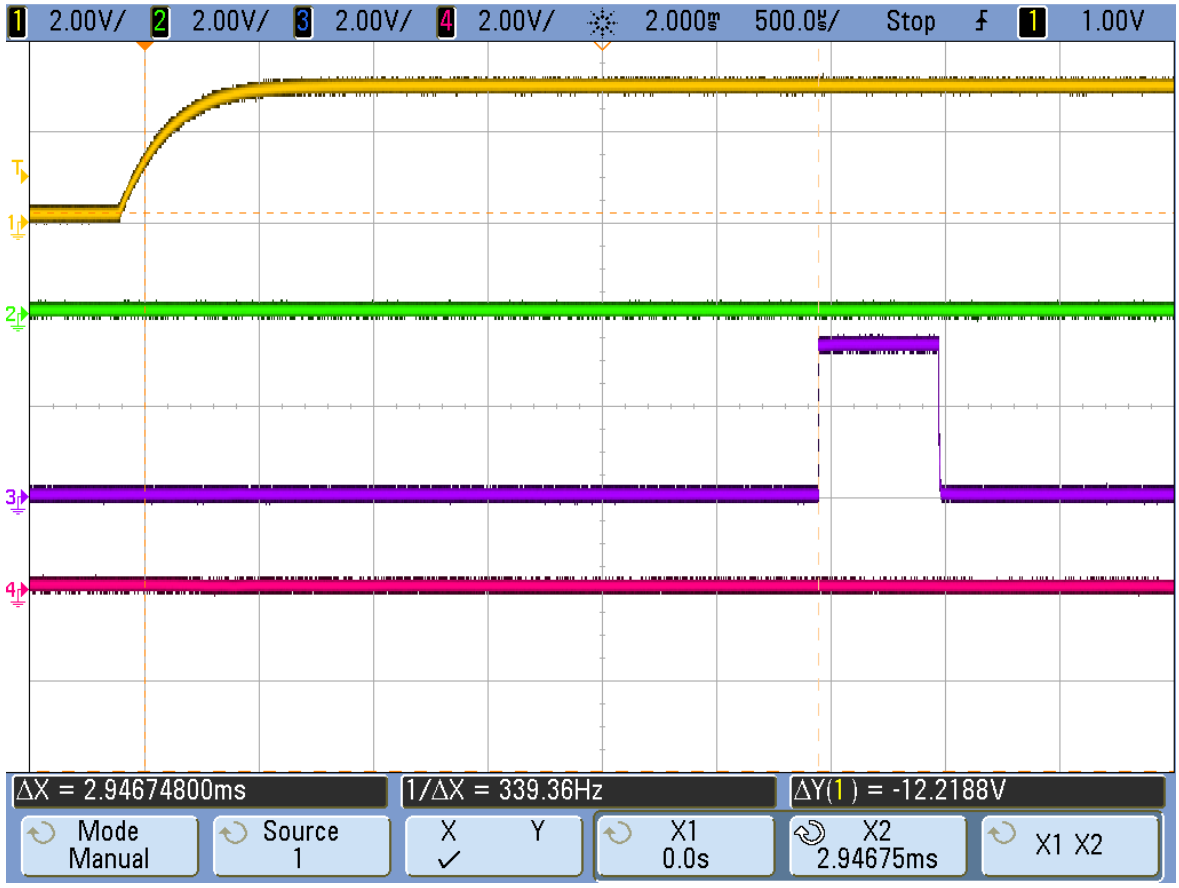

Figure 8. Wake-up time measurements for Power-down mode using RESETN pin

- For wake-up from the Deep power-down mode, measure time passed on the oscilloscope between the `RESETN` pin going high to the `P0_22` pin going high as shown in the following figure.

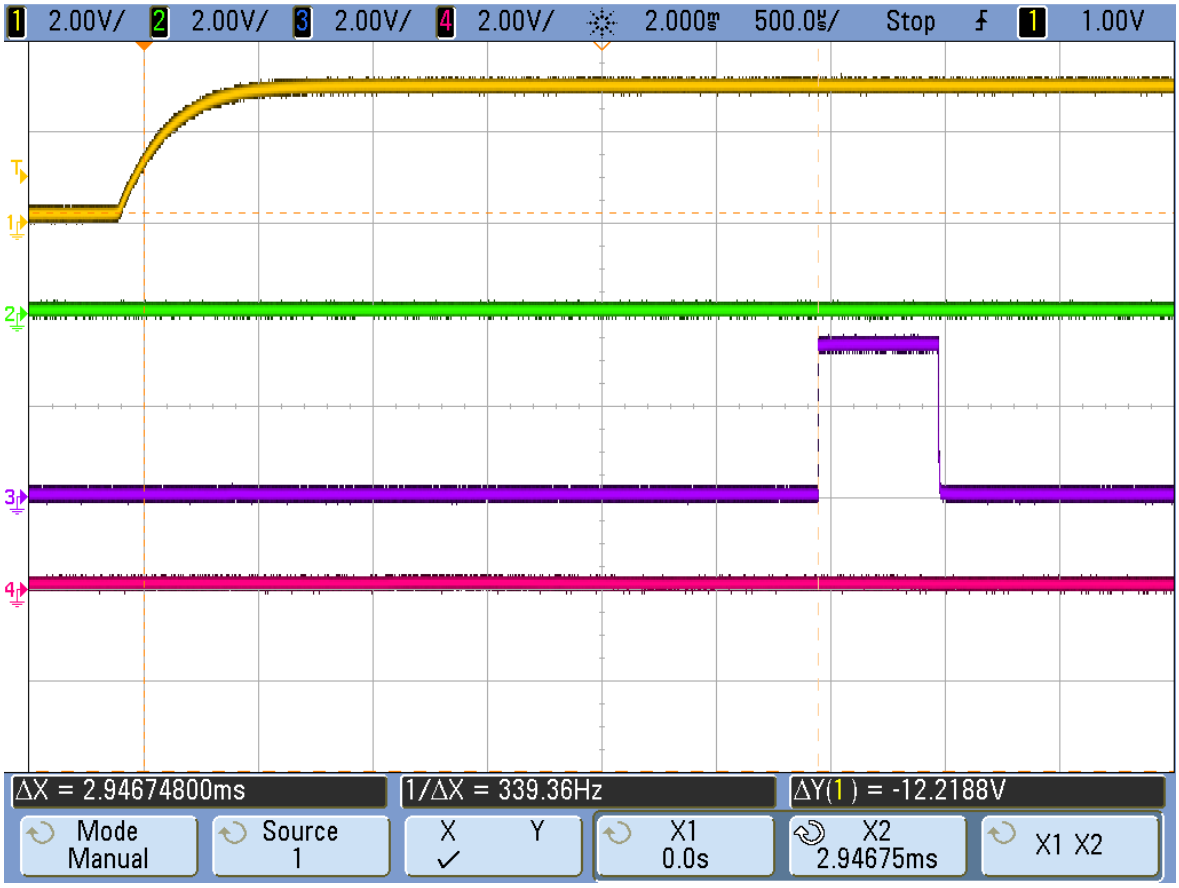

Figure 9. Wake-up time measurements for Deep power-down mode using RESETN pin

4.5 Typical (IAR IDE) low-power mode current and wake-up time

Table 26. Typical (IAR IDE) low-power mode current and wake-up time

Low-power mode	Measurement parameter	Observation	
		Current value	Wake-up time
Sleep	VDD_MAIN = 3.3 V, 12 MHz	2.448 mA	3.14 µs
Deep sleep	VDD_MAIN = 3.3 V, all RAM retained (128 kB)	97.9 µA	74.75 µs
Power down	VDD_MAIN = 3.3 V, RAM_00/RAM_01 retained (8 kB)	7.7 µA	405.5 µs (group interrupt) 2.947 ms (RESETN pin)
	VDD_MAIN = 3.3 V, RAM_X0/RAM_00/ RAM_01 retained (24 kB)	8.1 µA 12.2 µA	405.3 µs (group interrupt) 2.947 ms (RESETN pin)
	VDD_MAIN = 3.3 V, all RAM retained (128 kB)		404.6 us (group interrupt) 2.947 ms (RESETN pin)
Deep power down	VDD_MAIN = 3.3 V, RTC disabled, RAM_00 not retained (4 kB)	1.6 µA 1.8 µA	2.866 ms (wake-up pin) 2.947 ms (RESETN pin)
	VDD_MAIN = 3.3 V, RTC disabled, RAM_00 retained (4 kB)		2.866 ms (wake-up pin) 2.947 ms (RESETN pin)

5 Conclusion

LPC553x/LPC55S3x provide great flexibility and multiple options for the user to achieve low-power consumption with the required wake-up configuration. This flexibility allows a trade-off between power consumption and wake-up speed based on the application requirements of the user.

6 Revision history

Table 27 summarizes revisions to this document.

Table 27. Revision history

Revision number	Release date	Description
2	04 September 2023	• Updated software package • Updated Section 4.1
1	11 May 2022	Replaced Low-power Modes and Wake-up Time with LPC553x/ LPC55S3x Low-power Modes and Wake-up Time
0	11 February 2022	Initial public release

7 Legal information

7.1 Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

7.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

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.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. 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.

NXP B.V. - NXP B.V. is not an operating company and it does not distribute or sell products.

7.3 Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Contents

For more information, please visit: http://www.nxp.com

Date of release: 4 September 2023
Document identifier: AN13530