AN13166

LPC55S1x Low Power and Wake-up Optimization

Rev. 0 — March 4, 2021 Application Note

Contents

1 Preamble....1
2 LPC55S1x overview....1
3 Power management....2
3.1 Power modes....2
3.2 Wake-up process....2
4 Power configurations....2
4.1 Active mode....3
4.2 Sleep mode....3
4.3 Deep sleep mode....3
4.4 Power down mode....3
4.5 Deep power down mode....4
4.6 Some tips for lower power....4
4.7 About shorter wake-up time....4
5 Example to achieve typical data on datasheet....4
5.1 Hardware environment....4
5.2 Hardware setup....5
5.3 Software environment setup....6
5.4 Reference software implementation....7
5.5 Running & Measure results....7

1 Preamble

In an embedded application, low power design is quite important, which means to achieve the same performance, the lower power consumption the better. Especially with the IoT's development, it is more and more universal to get lower power in product. NXP has provided many application notes to introduce low power and wake-up for different LPC series. The basic ideas on achieving lower power and shortering wake-up time are same. Basically, the fewer modules on and fewer system clock, the lower power there is. However, with more modules on and more system clock, the wake-up time becaomes shorter. But some features on power management and the specifications on power consumptions and wake-up time are different.

This application note describes how to get the lower power consumption and shorter wake-up time in different power modes by achieving the typical data of the specifications on power consumptions and wake-up time listed in *LPC55S1x/LPC551x Data Sheet* (document LPC55S1x_PDS).

2 LPC55S1x overview

The LPC55S1x/LPC551x is an Arm® Cortex®-M33 based microcontroller for embedded applications. These devices include CASPER Crypto engine, up to 256 KB on-chip flash, up to 96 KB of on-chip SRAM, PRINCE module for on-the-fly flash encryption/decryption, Code Watchdog, high-speed/full-speed USB host and device interface with crystal-less operation for full-speed, CAN FD, five general-purpose timers, one SCTimer/PWM, one RTC/alarm timer, one 24-bit Multi-Rate Timer (MRT), a Windowed Watchdog Timer (WWDT), nine flexible serial communication peripherals (which can be configured as a USART, SPI, high speed SPI, I2C, or I2S interface), Programmable Logic Unit (PLU), one 16-bit 2.0 Msamples/sec ADC capable of simultaneous conversions, comparator, and temperature sensor.

Features relevant to power include:

- Power-saving modes and wake-up:
  - Integrated PMU (Power Management Unit) to minimize power consumption.
  - Reduced power modes: Sleep, deep-sleep with RAM retention, power-down with RAM retention and CPU retention, and deep power-down with RAM retention.
  - Configurable wake-up options from peripherals interrupts.
  - The Micro-Tick Timer running from the watchdog oscillator, and the Real-Time Clock (RTC) running from the 32.768 kHz clock, can be used to wake-up the device from sleep and deep-sleep modes.
- Operating from internal DC-DC converter.
- Single power supply 1.8 to 3.6 V.
- Operating temperature range -40 °C to +105 °C.

3 Power management

LPC55S1x devices include a variety of power switches and clock switches to allow fine tuning power usage to match requirements at different performance levels and reduced power modes.

3.1 Power modes

There are five power modes on LPC55S1x, listed as below from highest to lowest power consumption:

1. **Active mode**: The part is in *Active mode* after a Power-On Reset (POR), hardware pin reset or software reset and when it is fully powered.
2. **Sleep mode**: *Sleep-mode* saves a significant amount of power by stopping CPU execution without affecting peripherals or requiring significant wake-up time. Sleep-mode 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, and the logic levels of the pins remain static.
3. **Deep-sleep mode**: In deep-sleep mode, the system clock to the CPU is disabled as in sleep-mode. The main clock and all peripheral clocks are disabled. 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 flash memory and ROM are put in shutdown mode, with the cost of a longer wake-up time compared to the sleep-mode. 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 logic levels of the pins remain static.
4. **Power-down mode**: Power-down mode turns off nearly all on-chip power consumption by shutting down the DCDC, with the cost of a longer wake-up time compared to deep-sleep mode. The clock to the CPU and peripherals is shut down as deep-sleep mode. Both FRO 192 MHz and FRO 1 MHz are disabled. All SRAM can be configured to maintain their internal state and all registers lose their internal states except those located in the always-on power domains. The internal state of the CPU0, ROM patch unit, AHB security controller and PRINCE are maintained and the logic levels of the pins remain static.
5. **Deep power down mode**: Deep-power down mode shuts down virtually all on-chip power consumption but requires a significantly longer wake-up time (compared to power down mode). For maximal power savings, the entire system (CPU and all peripherals) is shut down except for the PMU, the PMC, the RTC and the OS event timer. During deep-power down mode, the contents of the SRAM can be retained (software configured via the low power API) and 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 four wake-up pins and the RESET pin.

Each reduced power mode (sleep, deep sleep, power down and deep power down mode) is entered from the active mode. The power and clock to the peripherals are configurable. LPC55S1x provides some registers for it. For more convenience, power APIs are implemented in the power library in SDK software. Thus, control of device power consumption or entry to low power modes can be configured through simple calls to the power APIs. For the details, see Power configurations.

3.2 Wake-up process

The part always wakes up to the active mode. To wake up from the reduced power modes, configure the wake-up source. Each reduced power mode supports its own wake-up sources and needs to be configured accordingly.

4 Power configurations

The LPC55S1x supports a variety of power control features. In active mode, when the chip is running, power and clocks to selected peripherals can be optimized for power consumption by register configuration. In the four reduced power modes (sleep, deep sleep, power down and deep power down), the power consumption can be optimized and wake-up source can be selected by calling the power APIs with different parameters.

4.1 Active mode

As you know, the CPU, memories, and peripherals are clocked by the AHB/CPU clock.

As mentioned, the chip is in active mode after reset. The default power configuration is determined by the reset values of the PDRUNCFG0, AHBCLKCTRL0, AHBCLKCTRL1 and AHBCLKCTRL2 registers. For the details of the registers, see *LPC55S1x/LPC551x User Manual* (document UM11295). The power configuration can be changed during run time. If specific times are known when certain functions will not be needed, they can be turned OFF temporarily and turned back ON when they will be needed. The general way of power configuration is as below:

- The AHBCLKCTRL registers control which memories and peripherals are enabled.
- The power to various analog blocks (PLL, oscillators, and the BOD circuit) can be controlled individually through the PDRUNCFG0 register. If turned OFF, time will be needed before these blocks can be used again after being turned ON.
- The power library provides an easy way to optimize power consumption depending on CPU load and performance requirements.

NOTE

For more details, see **LPC55S1x/LPC551x Power Profiles/Power Control API** in *LPC55S1x/LPC551x User Manual* (document UM11295).

4.2 Sleep mode

In the sleep mode, compared to active mode, the system clock to the CPU is stopped and execution of instructions is suspended. So power consumption in sleep-mode is configured by the same settings as in active mode. In other words, by calling the power API `POWER_EnterSleep()`, the chip enters the sleep mode (CPU is automatically shut off by hardware) with inheriting the power configurations in active mode.

4.3 Deep sleep mode

The power consumption in deep sleep mode is determined primarily by which analog/digital wake-up sources remain enabled. Serial peripherals and pin interrupts configured to wake-up the part, contribute to the power consumption only to the extent that they are clocked by external sources.

Based on the requirements in application, call the power API `POWER_EnterDeepSleep()` with the parameter to control which the analog/digital peripherals are powered up, which SRAM instances are in retention state and which wake-up sources/events are enabled.

4.4 Power down mode

The power consumption in power down mode is determined primarily by the number of enabled SRAM instances (retention mode). Serial peripherals in Flexcomm3 and pin interrupts configured to wake-up contribute to the dynamic power consumption only to the extent that they are clocked by external sources.

Based on the requirements in application, call the power API `POWER_EnterPowerDown()` with the parameter to control which the analog/digital peripherals are powered up, which SRAM instances are in retention state and which wake-up sources/events are enabled.

NOTE

In power-down mode entered by calling `POWER_EnterPowerDown()`, the CPU state is retained which is implemented by shifting the CPU0 registers values inside SRAM instance `RAMX_2`, meaning that `RAMX_2` must be kept in retention, Along with CPU, the state of AHB security controller and PRINCE registers values will also be shifted in `RAMX_2`. Address range `[0x0400_2000 - 0x0400_25FF]` inside `RAMX_2` is used, which means that any data in this area prior to calling the low power API will be lost.

After a wake-up event occurs, CPU resumes code execution after the call to the low power API function. It is the responsibility of the customer application to re-configure the modules whose states have not been retained.

When CPU state is retained, all SRAM instances that contain CPU variables, stack and heap, must also be retained.

4.5 Deep power down mode

Deep power-down mode has the following configuration options (via the low power API):

- RAMs instances to be retained
- Wake-up pins
- 32 kHz clock source for RTC and OS Event Timer
- On wake-up, the part reboots

4.6 Some tips for lower power

The essential idea and way to save power is to shut off the unused resources on chip, e.g. various peripherals. Although most resources are shut off on reset, some unused ones are not easy to be found because a chip contains so many resources. There are some tips to help dig them out for lower power:

- Generally speaking, everything uses less power at **lower frequencies**, so running the CPU and other device features at a frequency sufficient for the application (plus some margin) will save power.
- If the **PLL** is not needed, it should be turned OFF to save power. Also, running the PLL at a lower CCO frequency saves power.
- Several peripherals use individual peripheral clocks with their own clock dividers. The peripheral clocks can be shut down through the corresponding **clock divider** registers if the base clock is still needed for another function. Disabling clock divider itself can save power since it has some power consumptions.
- Disabling the internal pull-ups and pull-downs on pins as possible.
- Enabling automatic clock gating for some peripherals can save power. But note it causes a delay for the next access.
- If using FRO 12 MHz, turning off FRO 96 MHz can save powers.
- The IOCON clock can be disabled to save power after pins are configured.
- Disable BOD VBAT reset if not needed.
- Shut off the analog and digital peripherals used in ROM code but unused in user code.

Of course, there are other ways to save power on an application beside the above tips.

4.7 About shorter wake-up time

The lower power consumption is balanced with the performance depending on an application. It also needs to be balanced with wake-up time. Lower power consumption often means longer wake-up time and shorter wake-up time means higher power consumption.

5 Example to achieve typical data on datasheet

This section introduces the example implemented based on LPC55S1x SDK and demonstrates how to achieve the typical data on power consumptions and wake-up time, as presented in *LPC55S1x/LPC551x Data Sheet* (document LPC55S1x_PDS), in different power modes.

5.1 Hardware environment

- LPC55S16-EVK REV A
- Personal computer
- USB cable

- Digital multimeter for current measure
- Oscillograph for wake-up time measure

5.2 Hardware setup

To measure the current consumption and wake-up time, set the hardware on the LPC55S16-EVK board as below.

5.2.1 Measure current consumption

According to the definitions in the datasheet, the current consumption on MCU is total current from VBAT_DCDC, VBAT_PMU, VDDA, and VDD supply domains. On the LPC55S16 EVK board, the supplies are separated for measure. As shown in Figure 1, **JP22** is used for VBAT_DCDC and VBAT_PMU, **JP20** for VDD and **JP21** for VDDA. To measure the individual current, connect the digital multimeter to one open jumper. Then, add results to get the total current value.

5.2.2 Measure wake-up time

As shown in Figure 1, on the LPC55S16 EVK board:

- one keypad labeled **USR/SW3** is connected to a GPIO pin for wake-up source of sleep and deep sleep modes.
- one keypad labeled **WAKEUP/SW1** is connected to a WAKEUP pin for wake-up source of power down and deep power down modes.

When either keypad is pushed down, there is a falling edge on the pin. With the wake-up signal on the pin being asserted, the MCU wakes up and enters the interrupt handler where toggling another GPIO pin for strobe. This GPIO pin is set to a logic **0** before entering the low power mode and set to a logic **1** after the MCU wakes up. In this way, the duration between the assertion of the wake-up signal and rising edge of the strobe pin is the approximate wake-up time taken by MCU. The time can be measured easily after the edge waveforms on both pins are captured by oscillograph. See Figure 1 for the pins connected to oscillograph:

- Measure **GPIO pin for wake-up** and **strobe pin** for wake-up time on sleep and deep sleep mode.
- Measure **WAKUP pin** and **strobe pin** for wake-up time on power down and deep power down mode.

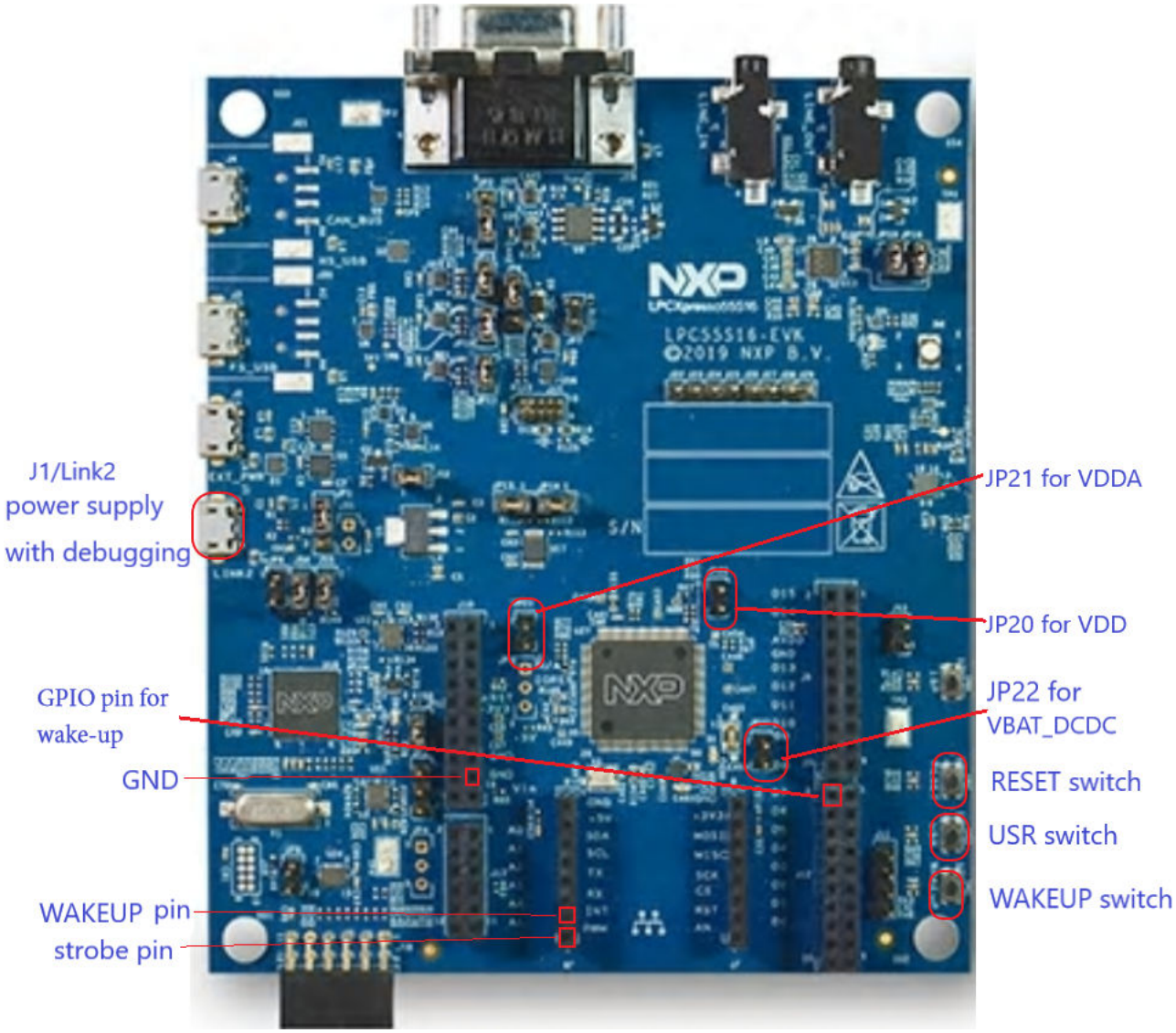

Figure 1. Hardware setup on LPC55S16 EVK Board

With the above setup completed, connect PC to USB port with debugger and VCOM labeled **J1/LINK2** on the target board via a micro USB cable. Now, the hardware is ready for work.

5.3 Software environment setup

- IAR Embedded Workbench IDE 8.32.4
- Serial terminal program, i.e. PuTTY, with the serial line setup: 115200+8+1+N
- *LPC55S1X_power_management_optimization* (document AN13166SW) based on LPC55S1x SDK V2.8.2

After unzipping the software package, copy and paste it to the SDK package for work:

- Copy the entire **power_manager_optimization_lpc** folder to *\SDK_2.8.2_LPCXpresso55S16\boards\lpcxpresso55s16\demo_apps* of the SDK.
- Copy the *system_LPC55S16_wkup_measure.c* files in the **device** folder to *\SDK_2.8.2_LPCXpresso55S16\devices\LPC55S16* of the SDK.

- Copy the *fsl_gint_wkup_measure* file in the **driver** folder to *\SDK_2.8.2_LPCXpresso55S16\devices\LPC55S16\drivers* of the SDK.

5.4 Reference software implementation

For better showing the optimization of the power consumption and wake-up time, the reference software follows the test conditions on the power consumptions and wake-up time in the datasheet as possible. It is simple to use `flexcomm0` USART for user interface without any function, communicating with serial terminal program on PC through which users can test conveniently the data in different power modes. This adds a few power consumption which can be ignored.

The software uses FRO 12 MHz for the system clock source with PLL disabled. To optimize the power consumption, follow the steps as below:

1. Disable BOD VBAT Reset.
2. Disable IOCON block after IO configured.
3. Shut off the analog and digital peripherals used in ROM code but unused in user code.
4. Shut off FRO 96 MHz clock since using FRO 12 MHz.
5. Enable all automatic clock gating bits.
6. Disable the internal pull-ups and pull-downs for the used pins, e.g. pins for wake-up. This contributes to the power savings on VDD.

NOTE

- The codes for 1 to 5 can be seen in the *main.c* file and for 6 can be seen in the *pin_mux.c* file.
- In the reference software project, SRAM0-2 is replaced with SRAMX for R/W region in scatter file, as described in *LPC55S16_ramx.icf*. SRAM can be shut off for saving power. However, to follow the measure conditions (all SRAM on) in the datasheet, they are not shut off.

To shorter the wake-up time in the software, place the codes of wake-up ISR `GINT0_DriverIRQHandler()` for power down mode in SRAM instead of Flash.

5.5 Running & Measure results

To set up the hardware and software environments, first connect digital multimeter on JP22 to measure the main current consumption on VBAT. Open the **power_manager_lpc.eww** project under *\SDK_2.8.2_LPCXpresso55S16\boards\lpcxpresso55s16\demo_apps\power_manager_optimization_lpc\iar* using IAR to build projects.

After power cycle or reset, the example runs and prints the basic information of the example on the window of the terminal program on PC, as shown in Figure 2. The basic test conditions are presented. On LPC55S16 EVK board, the power supplied to MCU is 3.3 V which is a bit different from the datasheet (3.0 V). It should cause the measure result a bit less than the data on the datasheet.

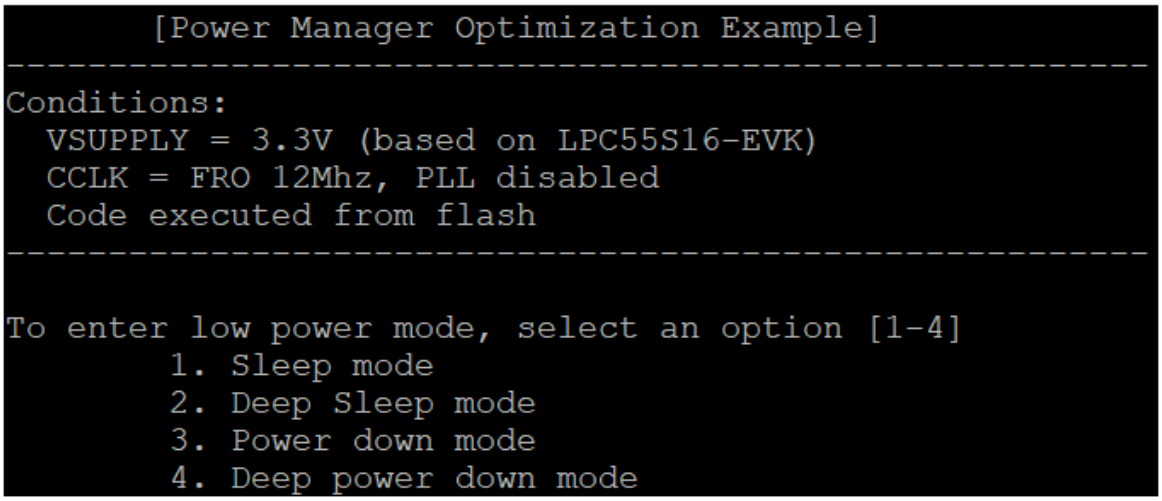

Figure 2. Active mode on POR/reset

At this point, the chip is in active mode, and the current value on power domain of VBAT_DCDC & VBAT_PMU can be observed on the digital multimeter.

Per the hints on the console, the chip may enter sleep mode, deep sleep mode, power down mode or deep power down mode with typing **1**, **2**, **3** or **4** accordingly on the keypad of PC when the current value on VBAT_DCDC & VBAT_PMU for the corresponding low power mode can be observed on the digital multimeter.

Then with the hint, press USR/WAKEUP switch on the target board to wake up the chip returning to active mode when the wake-up time can be measured with the edge signal waveforms on both USR/WAKEUP pin and strobe pin being captured on the oscillograph.

Moreover, by connecting the digital multimeter to JP20 and JP21, repeating the above steps can measure the current consumption values on VDD and VDDA in the five power modes. In this example, both are measured as **0** in all power modes.

NOTE

Actually, VDD is in the nA range which can be ignored.

So the total current consumption is equal to the one on VBAT_DCDC & VBAT_PMU.

As the data of wake-up time in sleep mode is provided at FRO 96 MHz in the datasheet, the example provides a macro definition to switch the system clock from FRO12 MHz to FRO96 MHz before entering sleep mode for this test. It is defined in the *power_manager_optimization_lpc.c* file as below:

```
#define DEMO_WAKEUP_AT_FRO96M (0)
```

When setting it to **1**, the system clock will be changed to 96 MHz for measuring the wake-up time from sleep mode. Or, it runs at 12 MHz.

NOTE

This definition is only applied to sleep mode.

Figure 3 to Figure 7 show the measure of wake-up time from sleep (12 MHz & 96 MHz), deep sleep, power down and deep power down modes for reference. The time is the difference between Cursor a and Cursor b. The value is highlighted with the red circle.

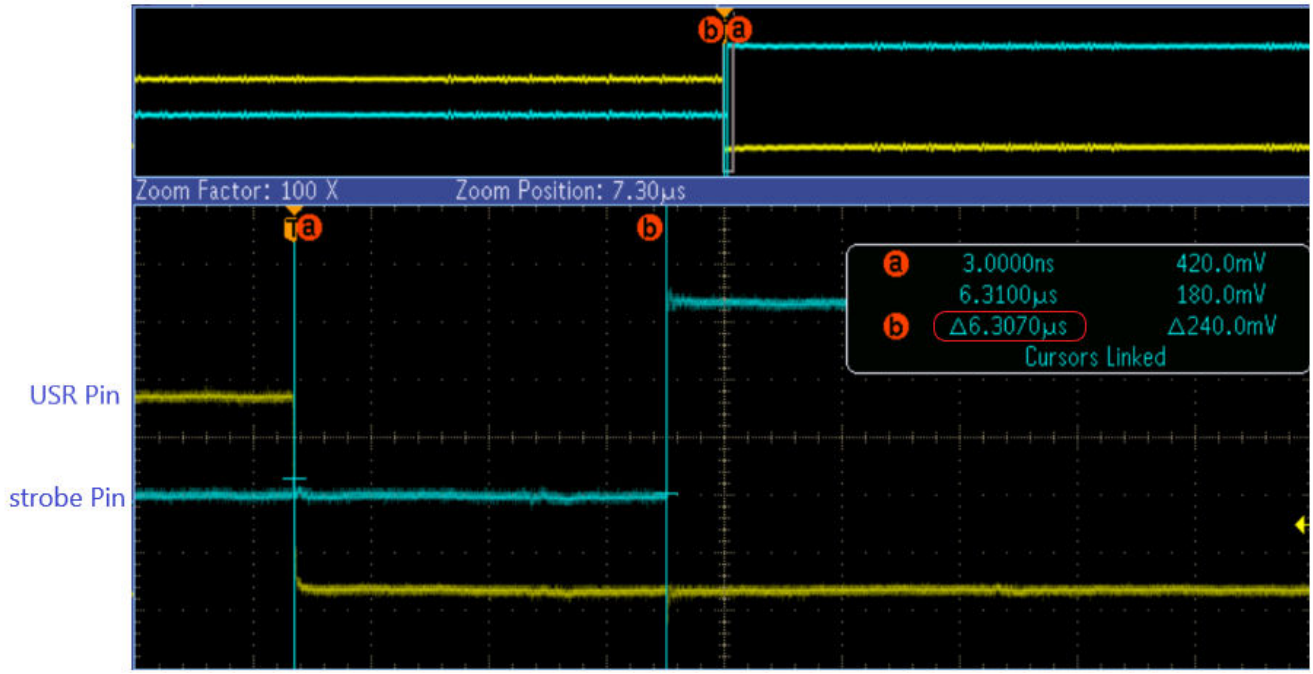

Figure 3. Wake-up time from sleep mode at FRO 12 MHz

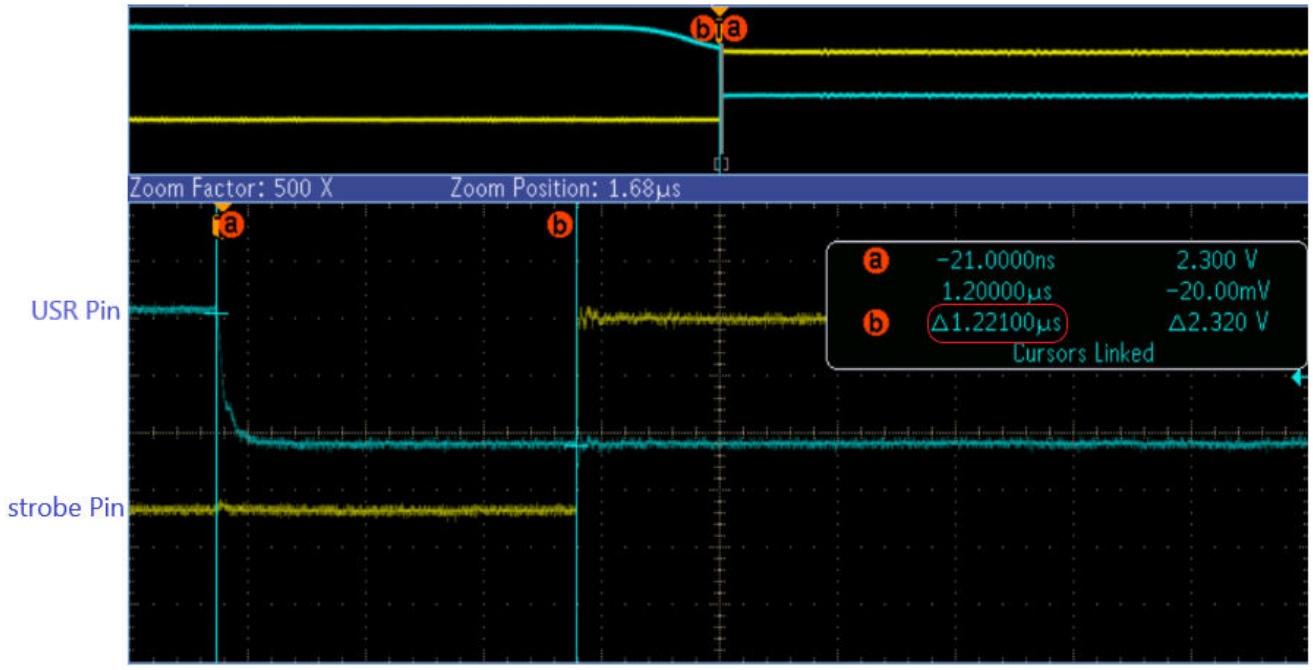

Figure 4. Wake-up time from sleep mode at FRO 96 MHz

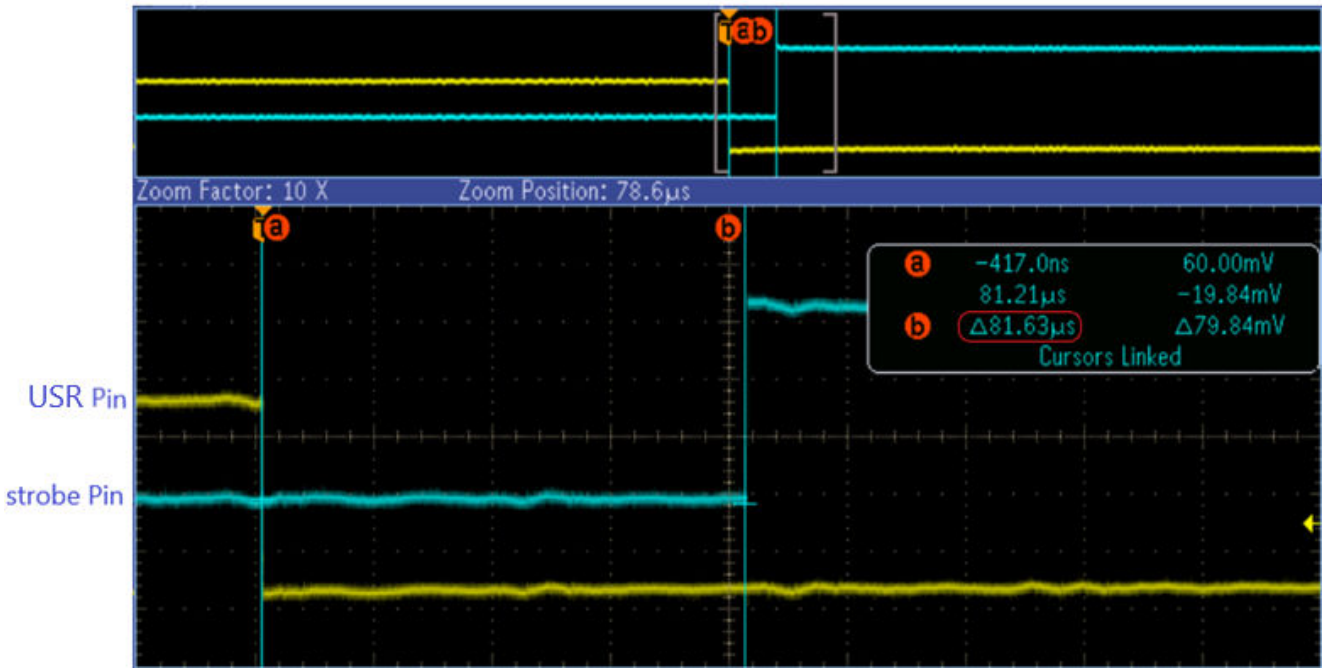

Figure 5. Wake-up time from deep sleep mode

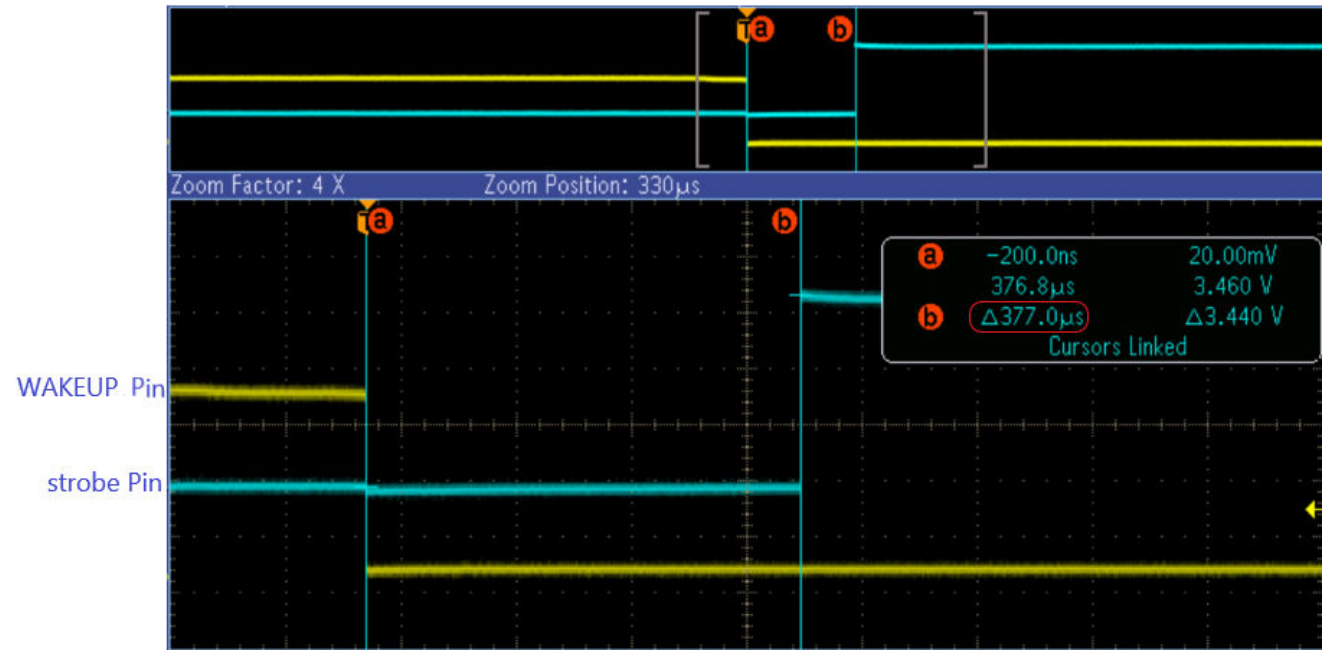

Figure 6. Wake-up time from power down sleep mode at FRO 12 MHz

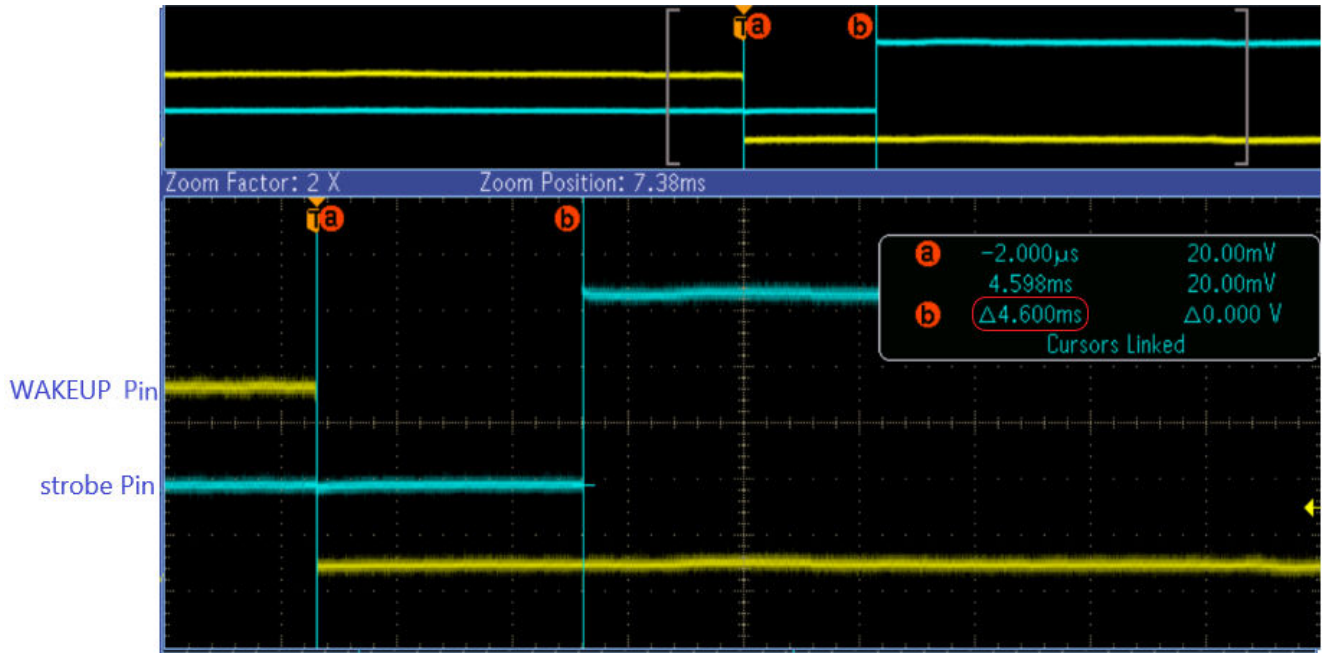

Figure 7. Wake-up time from deep power down mode

5.5.1 Summary of measure results

Table 1 summarizes the measure results of the power consumptions and wake-up time in all power modes with this example. The typical data, given by *LPC55S1x/LPC551x Data Sheet* (document LPC55S1x_PDS), can be achieved with the equivalent conditions.

Table 1. Typical power consumptions & wake-up time(Temp = 25ºC, Power supply = 3.3 V)

Power mode	Conditions	Idd current	Wake-up time
Active mode	FRO 12 MHz; PLL disabled; code executed from Flash; all SRAMs on	0.76 mA	—
Sleep mode	FRO 12 MHz; PLL disabled	0.58 mA	6.3 µS
	FRO 96 MHz; PLL disabled; No PRIMASK backup and restore	1.85 mA	1.2 µS
Deep-sleep mode	All SRAM on	63.3 µA	81.6 µS
Power-down mode	SRAMX_0 & SRAMX_2 on	2.5 µA	377µS
Deep Power-down mode	SRAMX_2 (4 KB) on; RTC oscillator disabled	0.4 µA	4.6 mS

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

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

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: March 4, 2021

Document identifier: AN13166

arm