AN13265

LPC55S6x 2x Low-power & Wake-up Optimization

Rev. 0 — 20 May, 2021 Application Note

Contents

1 Preamble

In an embedded application, low-power design is quite important and it is often required to achieve a possible lowest power consumption with meeting the performance. 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 about low power and wake-up for different LPC series. The basic ideas on achieving lower power and shorter wake-up time are same. Basically, the less modules on and less system clock, the lower power; however, the more modules on and more system clock, the shorter wake-up time. 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 LPC55S6x_2x datasheet.

2 LPC55S6x_2x overview

The LPC55S6x/LPC55S2x/LPC552x is an Arm® Cortex®-M33 based micro-controller for embedded applications. These devices include:

- Up to 320 kB of on-chip SRAM
- Up to 640 kB on-chip flash
- High-speed and full-speed USB host and device interface with crystal-less operation for full-speed
- One SD/MMC/SDIO interface
- Five general-purpose timers
- One SCTimer/PWM
- One RTC/alarm timer
- One 24-bit Multi-Rate Timer (MRT)
- One Windowed Watchdog Timer (WWDT)
- One high speed SPI (50 MHz)
- Eight flexible serial communication peripherals, each of which can be a USART, SPI, I2C, or I2S interface
- One 16-bit 1.0 M samples/sec ADC capable of simultaneous conversions
- One temperature sensor

Features relevant to power:

- Power-saving modes and wake-up:
  - Integrated Power Management Unit (PMU) to minimize power consumption.

  - Reduced power modes: Sleep, deep-sleep with RAM retention, power-down with RAM retention and CPU0 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 V to 3.6 V.
- Operating temperature ranging from -40 °C to +105 °C.

3 Power management

LPC55S6x/LPC55S2x/LPC552x devices include a variety of power switches and clock switches to fine tune power usage to meet the requirements for different performance levels and reduced power modes.

3.1 Power modes

There are five power modes on LPC55S6x/LPC55S2x/LPC552x, listed as below on power consumption from highest to lowest:

1. **Active mode**

   The part is in the active mode after a Power-On Reset (POR), hardware pin reset or software reset and when it is fully powered.

2. **Sleep mode**

   The 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 the 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 via registers. 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 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 sources. Each reduced power mode supports its own wake-up sources and needs to be configured accordingly.

4 Power configurations

The LPC55S6x/LPC55S2x/LPC552x 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 *LPC55S6x/LPC55S2x/LPC552x User manual* (document UM11126). The power configuration can be changed during run time. If specific times are known when certain functions are needed, they can be turned OFF temporarily and turned ON when 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 **LPC55S6x/LPC55S2x/LPC552x Power Profiles/Power Control API** in *LPC55S6x/LPC55S2x/LPC552x User manual* (document UM11126).

4.2 Sleep mode

As mentioned above, in the sleep-mode, compared to active mode, the system clock to the CPU is stopped and the 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

Power consumption in the 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 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

Power consumption in power-down mode is determined primarily by the number of SRAM instances which remain enabled (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 CPU0 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 CPU0, the state of AHB security controller and PRINCE registers values will also be shifted in RAMX_2. Address range [0x0400_6000 - 0x0400_65FF] 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, CPU0 will resume 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 CPU0 state is retained, all SRAM instances that contain CPU0 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):

- Analog peripherals, FRO32K & XTAL32K to be running.
- 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. Some unused ones are not easy to be found out from so much resources in a chip although most of these resources are shut off on reset. 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.
- Disable as many internal pull-ups and pull-downs on pins as possible.
- Enabling automatic clock gating for some peripherals can save power, but it causes a delay for the next access.
- If using FRO 12 MHz, turning off FRO 96 MHz can save power.
- 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 should be usually balanced with the performance depending on an application. And it also needs to be balanced with wake-up time. Lower-power consumption often means longer wake-up time. Inversely, shorter wake-up time means higher power consumption.

5 Example to achieve typical data on DS

This section introduces the example implemented based on LPC55S6x SDK to demonstrate how to achieve the typical data on power consumptions and wake-up time presented in *LPC55S6x Data Sheet* (document LPC55S6xDS) (in different power modes.

NOTE

Hereinafter, the introductions will be based on LPC55S6x. The measurement and data on LPC55S2x/LPC552x are same to LPC55S6x.

5.1 Hardware environment

- LPC55S69-EVK Rev. A2 with Rev. 1B of LPC55S69 part
- 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, the hardware can be set up on the LPC55S69-EVK (hereinafter it means Rev. A2) board which is described as below.

5.2.1 Measure current consumption

According to the definitions on DS, the current consumption on MCU is total current from VBAT_DCDC, VBAT_PMU, VDDA, and VDD supply domains. Referring to the design of LPC55S69-EVK schematic, some revisions are needed to be done on LPC55S69-EVK board for current measurement. The **P12** should be shorted and R92 should be removed. The total current can be measured by connecting digital multimeter on the opening jumper **P13** (see Figure 1).

5.2.2 Measure wake-up time

On LPC55S69 EVK board, there is a keypad labeled **USER/S3** connected to a GPIO pin for wake-up source of sleep and deep-sleep modes and a keypad labeled **WAKEUP/S2** connected to a WAKEUP pin for wake-up source of power down and deep power-down modes. See Figure 1.

When one of keypads is pushed down, there is a falling edge on the pin, and with the wake-up signal on the pin being asserted, the MCU will wake up and enter the interrupt handler where toggle another GPIO pin for strobe. This GPIO pin is set to a logic **0** before going into low power mode and set to a logic **1** after the MCU wakes up. In this way, the time between the assertion of the wake-up signal and rising edge of the strobe pin will be 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 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.

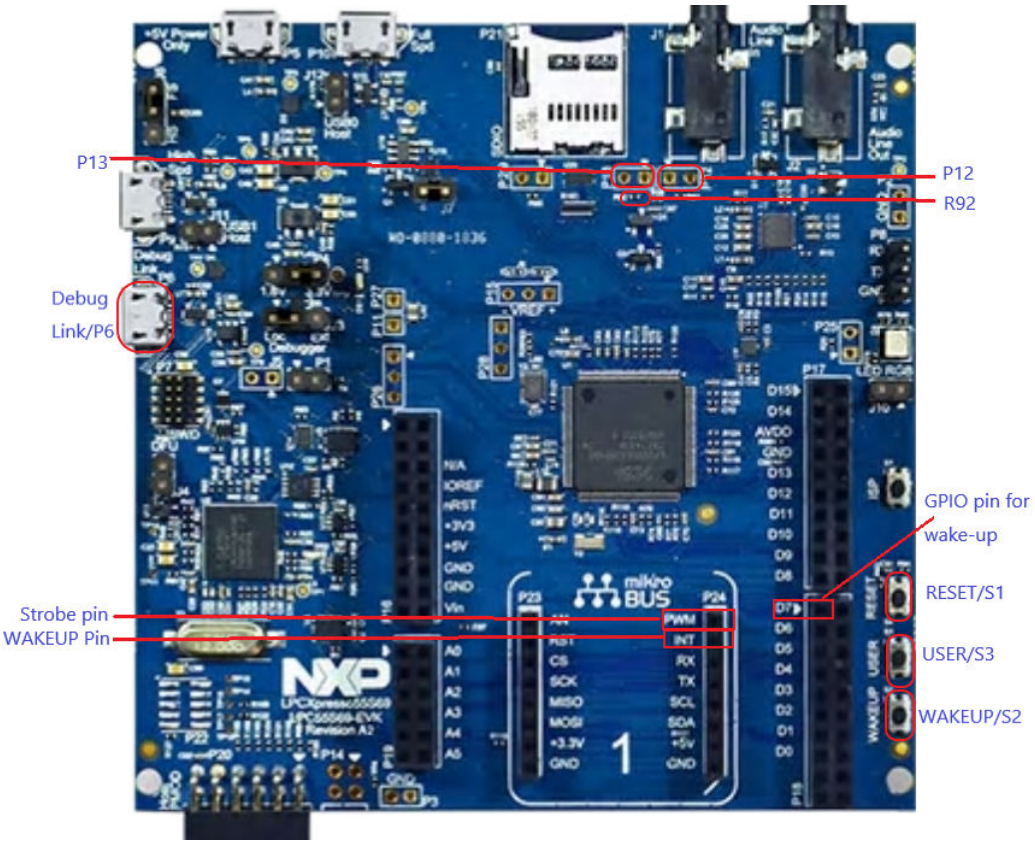

Figure 1. Hardware setup on LPC55S69-EVK Rev. A2 board

With the above setup, connect PC to USB port with debugger and VCOM labeled **Debug Link/P6** on the target board via a micro USB cable. Thus the hardware is ready for work.

5.3 Software environment setup

- IAR Embedded Workbench IDE 8.50.6
- Serial terminal program, i.e. PuTTY, with the serial line setup: 115200+8+1+N
- A reference design software package *LPC55S6X_power_management_optimization* based on LPC55S69 SDK V2.8.2 is attached with this application note.

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

- Copy the entire folder, **power_manager_optimization_lpc**, to the path *\SDK_2.8.2_LPCXpresso55S69\boards\lpcxpresso55s69\demo_apps* of the SDK.
- Copy the files *system_LPC55S69_cm33_core0_wkup_measure.c* in the folder, **device** to the path *\SDK_2.8.2_LPCXpresso55S69\devices\LPC55S69* of the SDK.
- Copy the file *fsl_gint_wkup_measure* in the **driver** folder to the path *\SDK_2.8.2_LPCXpresso55S69\devices\LPC55S69\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 DS as possible and is very simple without any function but using flexcomm0 USART for user interface – 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. The main methods to optimize the power consumption are:

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 96Mhz 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, 2, 3, 4, 5 can be seen in the *main.c* file. And the codes 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 (see *LPC55S69_cm33_core0_sram.icf*) so that the SRAM can be shut off for saving power. However, to follow the measure conditions (all SRAM on) in datasheet, they are not shut off.

The method to shorter the wake-up time in the software is:

1. Place the wake-up ISR codes for power down `GINT0_DriverIRQHandler()` in SRAM instead of Flash.

5.5 Running & Measure results

To set up the hardware and software environments, follow the steps as below:

1. Connect digital multimeter on JP22 to measure the main current consumption on VBAT.
2. Open the **power_manager_lpc.eww** projects under *\SDK_2.8.2_LPCXpresso55S69\boards\lpcxpresso55s69\demo_apps\power_manager_optimization_lpc\cm33_core0\iar* using IAR.
3. 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 the LPC55S69 EVK board, the power supplied to MCU is 3.3 V which is a bit different from the DS (3.0 V). It should cause the measure result a bit less than the data on DS.

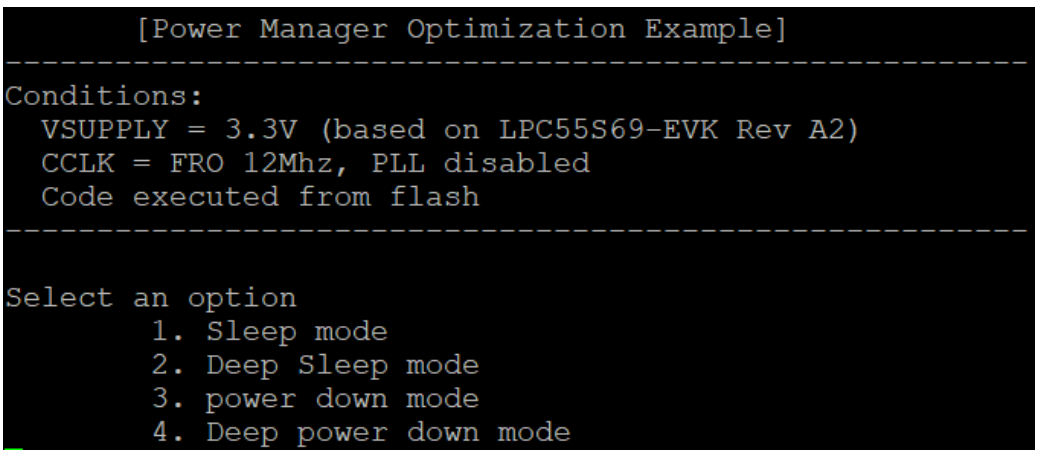

Figure 2. Active mode on POR/reset

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.

Because of the data of wake-up time in sleep mode is provided at FRO 96 MHz on 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.

Below figures show the measure of wake-up time from sleep at FRO 12 MHz (see Figure 3) and FRO 96 MHz (see Figure 4), deep-sleep (see Figure 5), power-down (see Figure 6) and deep power-down (see Figure 7) modes for user's reference. The time is the difference between Cursor a and Cursor b. The value is highlighted with the red circle.

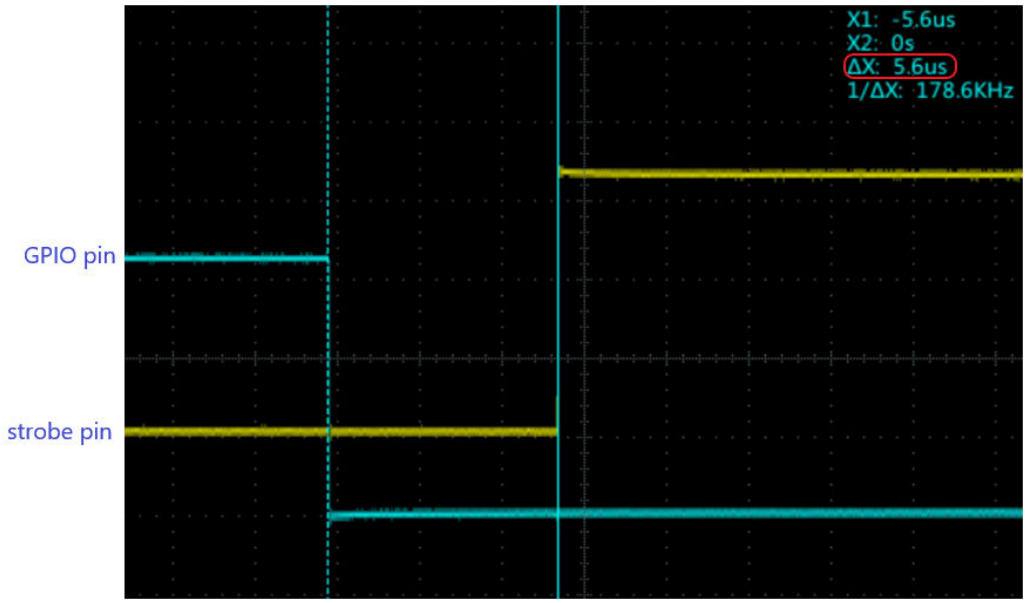

Figure 3. Wake-up time from sleep mode at FRO 12 MHz

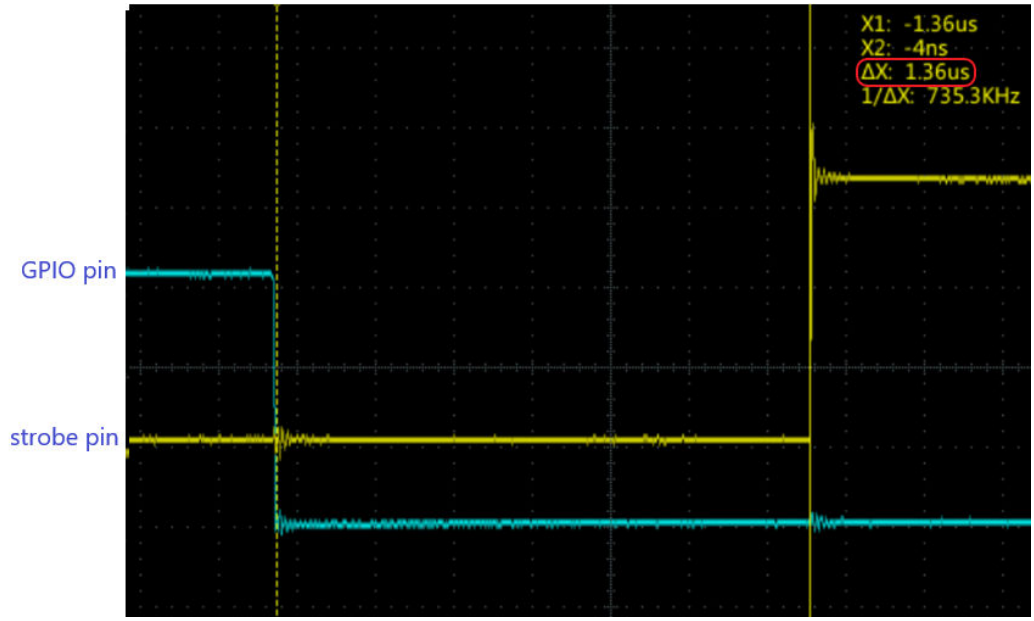

Figure 4. Wake-up time from sleep mode at FRO 96 MHz

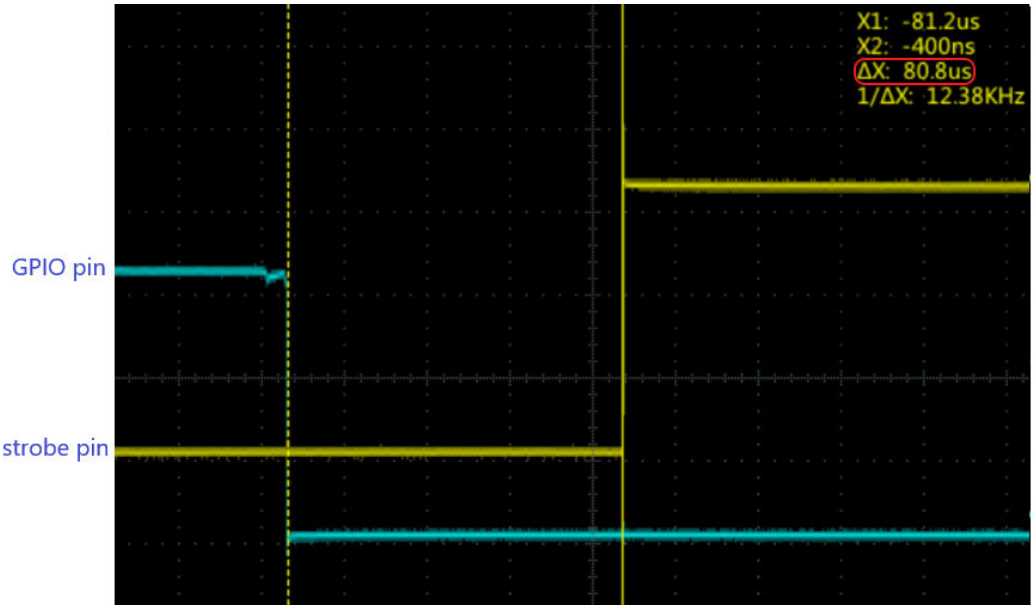

Figure 5. Wake-up time from deep-sleep mode

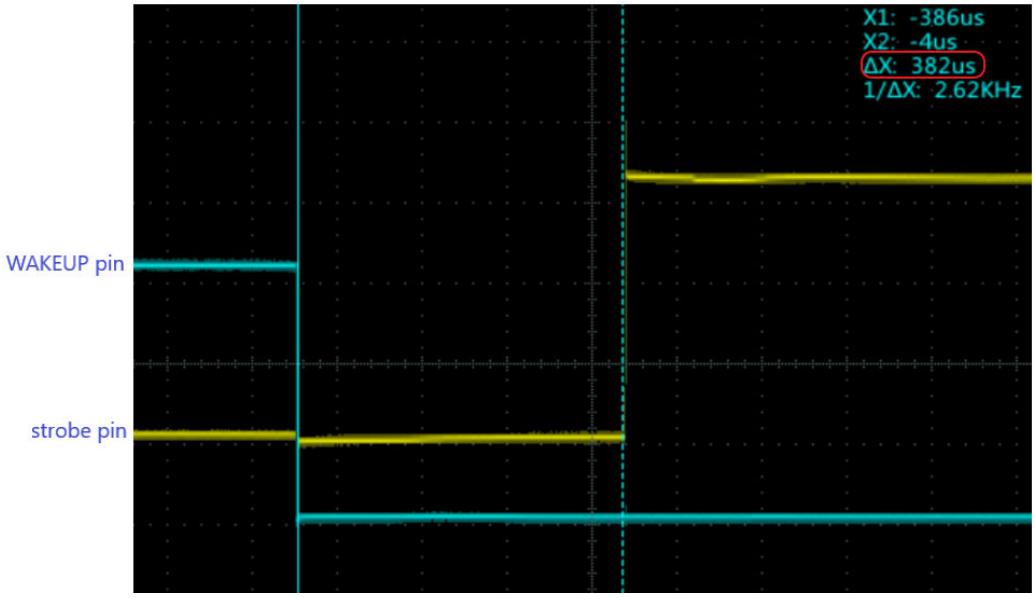

Figure 6. Wake-up time from power-down mode

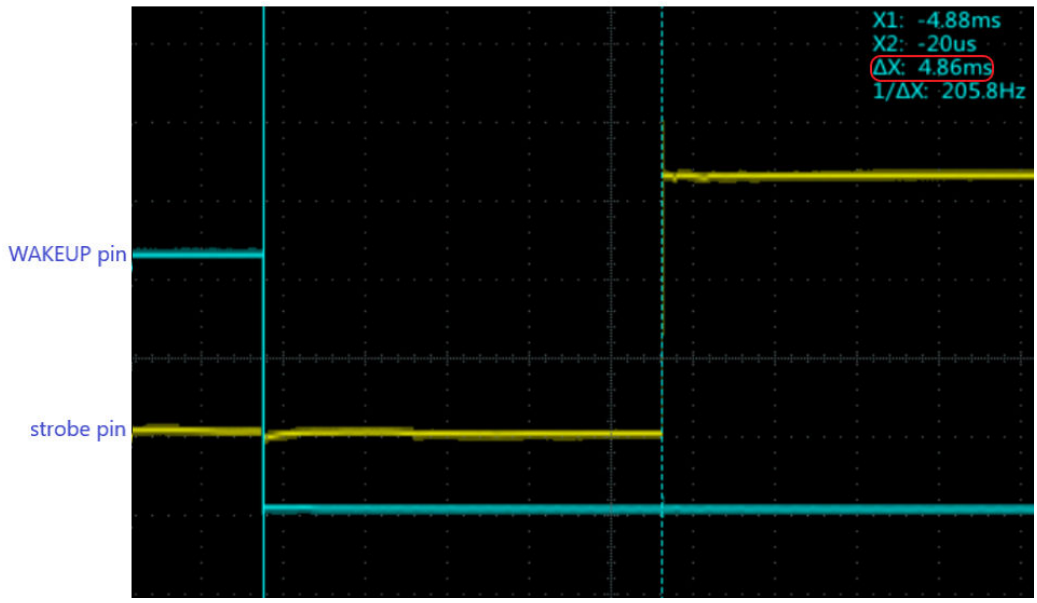

Figure 7. Wake-up time from deep power-down mode

5.5.1 Summary of the measure results

Table 1 summarizes the measure results of the power consumptions and wake-up time on all power modes with this example. It shows they can achieve the typical data given by LPC55S6x datasheet with the equivalent conditions.

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

Power mode	Conditions	Idd current	Wake-up time
Active mode	FRO12MHz; PLL disabled; code executed from Flash; all SRAMs on	0.94 mA	—
Sleep mode	FRO12MHz; PLL disabled	0.73 mA	5.6 µS
	FRO96MHz; PLL disabled; No PRIMASK backup and restore	2.39 mA	1.36 µS
Deep-sleep Mode	all SRAM on	101.9 µA	80.8 µS
Power-down Mode	SRAMX_2 & SRAMX_3 on	4.3 µA	382 µS
Deep Power-down Mode	SRAMX_2(4KB) on; RTC oscillator disabled	0.6 µA	4.86 mS

6 Revision history

Rev.	Date	Substantive changes
0	20 May, 2021	Initial release

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: 20 May, 2021

Document identifier: AN13265

arm