AN2100
APPLICATION NOTE

STR71x microcontroller power management

Introduction

This application note provides an overview of the STR71x power management features and gives some guidelines for using the low power modes to minimize the power consumption of the microcontroller.

Example software is provided with this application note for implementing and testing the various low power modes.

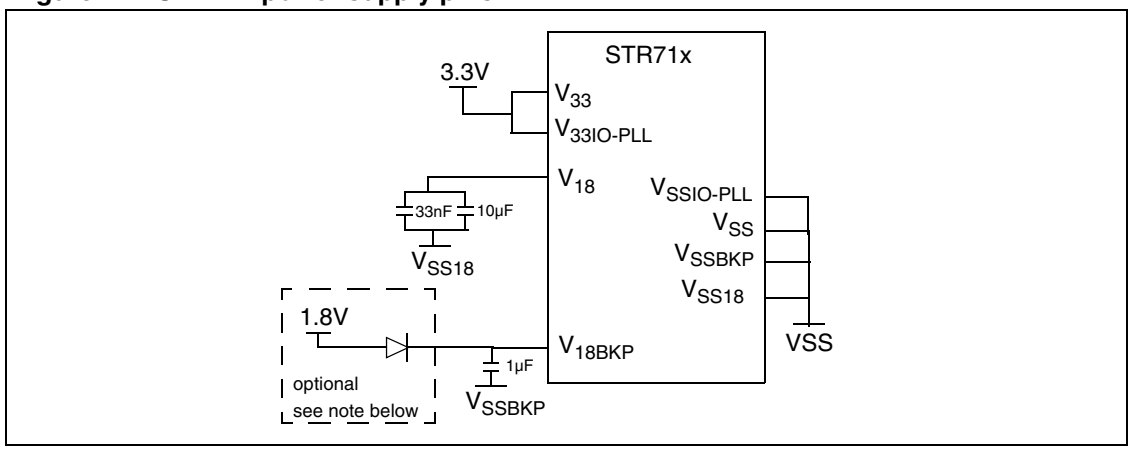
Contents

1 Power supply

1.1 Power supply pins

The following figure indicates the recommended configuration for the power supply pins:

Figure 1. STR71x power supply pins



- V_{33} pin: is the 3.3V main power supply pin. (V_{33} and $V_{33IO-PLL}$ are internally linked).
- V_{18} pin: must be connected to a capacitor of at least 10µF (Tantalum) + 33nF (ceramic) in order to guarantee the stability of the 1.8V supply to the core.
- V_{18BKP} pin: can be used to provide an external 1.8V supply to the backup block (RTC and Wakeup logic) when bypassing the internal voltage regulator during the STANDBY mode.

Caution: In STANDBY mode, when using an external 1.8V on V_{18BKP}, the V_{33} pin must remain connected to the 3.3V supply.

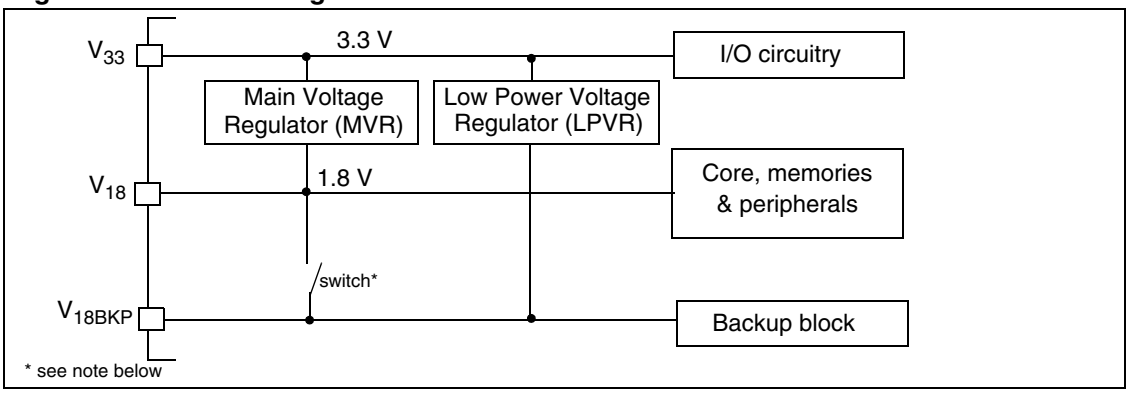
Note: 1 When not using an external 1.8V on V18BKP pin, a 1 µF capacitor must be connected between this pin and VSS_{BKP} to guarantee on-chip voltage stability.

2 Connecting an external 1.8V supply to the V18 pins is not supported.

1.2 Internal regulators

The following figure provides a schematic view of the power management block of the STR71x.

Figure 2. Power management block



*Note: * In normal operation, the switch connecting the V18 domain and the V18BKP domain is closed, it is opened only during STANDBY mode (refer to Section 3)*

1.2.1 Main Voltage Regulator (MVR)

In normal operation, the Main Voltage Regulator (MVR) provides the 1.8V supply.

The MVR can be switched off when entering a low power mode (refer to *Section 3*)

When the MVR is switched off, the Low Power Voltage Regulator (LPVR) can provide a power supply of approximately 1.6V .

Note: 1 The MVR has a static power consumption of 100 µA typ at 25°C.

2 When the MVR is switched off, The PLL (PLL1) is automatically disabled (PLL1 off) and the maximum allowed operating frequency is 1 MHz, this is due to the limitation imposed by the LPVR which is not able to generate sufficient current to operate in run mode.

1.2.2 Low Power Voltage Regulator (LPVR)

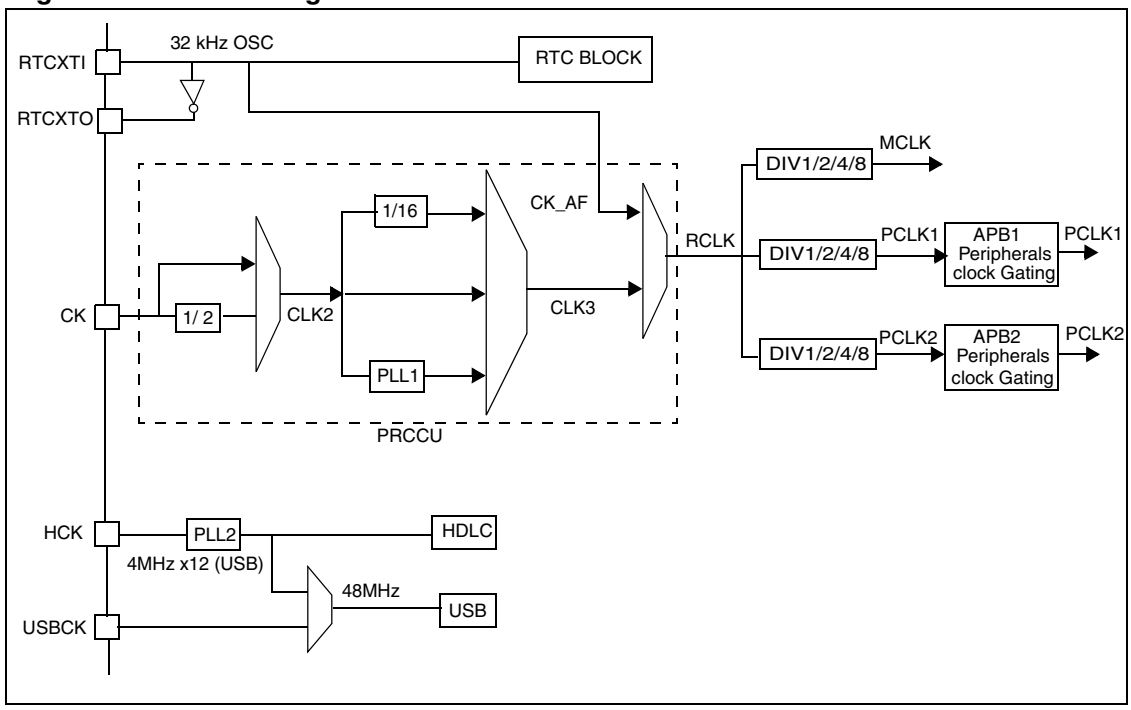
The Low Power Voltage Regulator (LPVR) is used when the MCU is in low-power mode and the main voltage regulator has been switched off. It has a different design from the main voltage regulator and generates a non-stabilized and non-thermally-compensated voltage of approximately 1.6V, its output current is not generally sufficient for the device to run in normal operation.

Note: In STANDBY mode the LPVR can be switched off while an external supply provides 1.8V to the chip through the V18BKP pin for use by RTC and Wake-Up block.

2 Clock management

The following figure provides an overview of the clock management block of the STR71x:

Figure 3. Clock management block



Legend:

MCLK = core and memory clock

PCLK1 = peripherals on APB1 clock

PCLK2 = peripherals on APB2 clock

As shown in the figure above, different clock configurations are provided by the STR71x MCU offering the means to optimize the power consumption in the device, the main features are:

- Separate clock selection for the core, the APB1 peripherals and APB2 peripherals.
- Peripherals clock gating (refer to *Section 2.1*).
- Up to three low power system clocks: the RTC 32 kHz clock, CLK2/16 and the PLL1 Free running mode clock (refer to *Section 2.2*).

2.1 Peripherals clock gating

It is possible to disable a peripheral clock by setting its corresponding bit in the APB Clock Disable Register (APBn_CKDIS). Refer to the APB Bridge Registers section in the STR71x reference manual.

The EMI and the USB Kernel clocks are stopped by resetting the corresponding bits in the RCCU_PER register.

Caution: After a device reset, to avoid extra power consumption, bits 0, 1 and 3 must be reset in the RCCU_PER register in order to disable the reserved clocks used only for factory test purposes.

2.2 PLL free running mode

The PLL is able to provide a low-precision clock, usable for slow program execution. The frequency range is from 125 kHz to 500 kHz.

The output frequency is selectable using the MX[1:0] bits and the FREE_RANGE bits according to the following table:

Table 1. PLL Free Running Mode Clocks

	Free Running Mode Frequency	
MX[1:0]	**FREE_RANGE = 0**	**FREE_RANGE = 1**
'01', '11'	~125KHz	~250KHz
'00', '10'	~250KHz	~500KHz

To Enable the PLL Free Running clock, you have to set bits DX[2:0] and set the FREEN bit in the RCCU_PLL1CR register.

Note: 1 The PLL2 could be disabled when not in use by setting bits DX[2:0] in the PCU_PLL2CR register.

3 STR71 low power modes

3.1 Low power mode characteristics

The STR71x low power modes are summarized in the following table:

Table 2. STR71x low power modes

Low Power mode	Description
SLOW	- PLL1 off - RCLK[3] = CLK2 or CLK2/16 or CK_AF (external 32 kHz)
Wait For Interrupt (WFI)	- Core clock stopped (MCLK off) - Wake-up by peripherals interrupts acknowledged by the interrupt controller (EIC) - MCU state retained (context restored after wakeup)
Low Power Wait For Interrupt (LPWFI)	- Core clock stopped (MCLK off) - Peripherals running at slow clock: RCLK = CLK2/16 or CK_AF (32kHz) - Wake-up by peripherals interrupts acknowledged by the EIC - MCU state retained
STOP	- Core and peripherals (on APB1, APB2) clocks stopped (RCLK off) - Wake-up by the configured external wake-up lines (XTI unit) - MCU state retained
STANDBY	- Core, memories and peripherals switched off[1] (except RTC and wakeup logic) - Main voltage regulator switched off - All I/Os are forced to high impedance except the Standby I/Os[2] - Wake-up by: the WAKEUP pin, the RTC alarm and the RESET pin. - MCU Reset after wake-up

Note: 1 Core, memories and peripherals not powered by 1.8V internal supply.

2 Standby I/Os are: nSTDBY, nRSTIN, WAKEUP, RTCXTI and RTCXTO.

3 Refer to Figure 3 for the definition of the various clocks (RCLK, MCLK, CLK2, CLK2/16, CK_AF).

3.2 Guidelines for entering/exiting low power modes

3.2.1 SLOW mode

To enter SLOW mode, RCLK must be configured as CLK2, CLK2/16 or CK_AF.

When using these clock settings the PLL1 can be disabled by writing ‘111’ in the DX[2:0] bits in the RCCU_PLL1CR register.

Table 3. SLOW mode selection

RCLK	CSU_CKSEL[1)]	CK2_16 [2)]	CKAF_SEL [3)]
CLK2	0	1	0
CLK2/16	x	0	0
CK_AF [4)5)]	x	x	1

Note: 1 CSU_CKSEL is bit 0 in register RCCU_CFR.

2 CK2_16 is bit 3 in register RCCU_CFR

3 CKAF_SEL is bit 2 in register RCCU_CCR, as mentioned in the table above this bit selection overrides the other clock selection bits.

4 The PLL can be configured to be automatically disabled when selecting the CK_AF (32KHz) as system clock, this can be done when setting bit CKSTOP_EN in the RCCU_CFR register.

5 When selecting the CK_AF(32KHz) as system clock, you can reduce power consumption by stopping the external oscillator using a GPIO pin.

3.2.2 WAIT For Interrupt mode (WFI)

To enter WFI mode, you must write a '0' in the WFI bit of the RCCU_SMR register.

To wakeup from WFI mode an interrupt request must be acknowledged by the EIC.

3.2.3 Low Power WAIT For Interrupt mode (LPWFI)

To enter LPWFI mode you have to:

1. Select the clock to be used by peripherals during LPWFI: CLK2/16 or CK_AF (WFI_CKSEL bit of the RCCU_CCR register)
2. Select LPWFI mode by setting the LPOWFI bit in the RCCU_CCR register.
3. Write 0 in the WFI bit of the RCCU_SMR register to enter LPWFI mode

Like WFI mode, to wakeup from LPWFI mode an interrupt request must be acknowledged by the EIC.

To further reduce MCU power consumption in LPWFI mode you can:

- Stop the Main voltage regulator (MVR) by setting bit LPVRWFI in the PCU_PWRCR register.
- Put the Flash in power-down mode by setting bit PWD in the FLASH_CR0 register, in this case the vectors and the interrupt handlers must be mapped in RAM and a delay of at least 20µs must be inserted in the wakeup interrupt routine in order to ensure that the Flash module has fully restarted before it is accessed.
- When selecting the CK_AF(32KHz) as system clock, you can reduce power consumption by stopping the external oscillator during LPWFI using a GPIO pin.

Note: 1 After exit from LPWFI mode, the Flash and the main voltage regulator are re-enabled automatically if they were switched off during the low power mode.

2 After wakeup, the original RCLK clock configuration must be restored by software.

3.2.4 STOP mode

To enter STOP mode you have to:

1. Configure at least one external wake-up line to wake-up the MCU from STOP mode. (Refer to the XTI section in the STR71x reference manual)
2. Reset the STOP bit in register XTI_CTRL and the STOP_I bit in the RCCU_CFR register.
3. To enter STOP mode, write the sequence 1, 0, 1 to the STOP bit in the XTI_CTRL register.
4. In order to avoid executing any valid instructions after a STOP bit setting sequence and before entering STOP mode, it is mandatory to execute a few (at least 6) dummy instructions after the STOP bit setting sequence.
5. To be sure that STOP mode was really entered, immediately after the end of the STOP bit setting sequence (including the dummy instructions), poll the PRCCU STOP_I flag bit and the STOP bit (XTI_CTRL register). If the STOP bit setting sequence has been correctly executed, these bits must be STOP_I = 1 and STOP = 0. If it is not the case you must restart all the sequence from the beginning.

When exiting STOP mode, clear the pending XTI interrupt bits (XTI_PRH and XTI_PRL registers).

To further reduce power consumption during STOP mode, it is possible to:

- Put the Flash in power-down mode by writing '1' in bit PWD in the FLASH_CR0 register.
- Disable the main voltage regulator by writing '1' in bit LPVRWFI in the PCU_PWRCR register.
- Stop the external oscillator using a GPIO pin

Note: After exit from STOP mode, the Flash and the main voltage regulator are re-enabled automatically if they were switched off during the STOP mode

Caution: The external oscillator must be reenabled in order to exit from STOP mode (see Example Application in *Section 4*).

3.2.5 STANDBY mode

The STANDBY sequence is initiated either:

- by setting the PWRDWN bit in the PCU_PWRCR register (software STANDBY entry)
- or by externally forcing the nSTDBY pin to '0' (hardware STANDBY entry).

Caution: You have to manage the nSTDBY, WAKEUP and RESET pin states in order to enter or exit STANDBY mode (refer to the STR71x reference manual, STANDBY section).

Note: To further reduce power consumption in STANDBY mode it is possible to:

- Bypass the Low Power Voltage regulator. In this case V_{18BKP} pin must be connected to an external 1.8V through a diode.

- Stop the LVD by setting bit LVD BYP in the PCU_PWRCR register

- Stop the 32 kHz oscillator by setting bit OSC BYP in the PCU_PWRCR

To reduce the board power consumption, the nSTDBY pin which is forced to low level during STANDBY mode, can be used to stop other on-board components (oscillators, power supplies...).

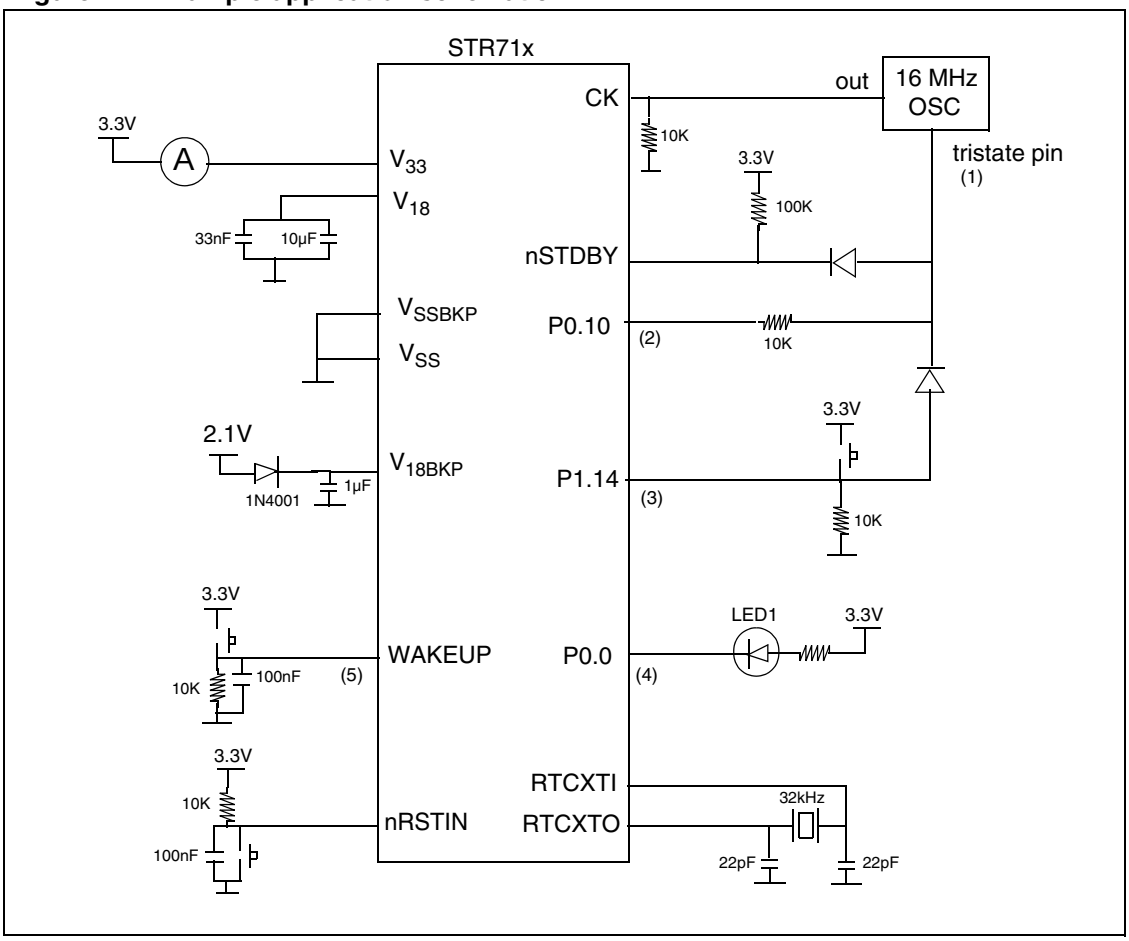
Caution: Because it powers the I/Os, V33 must not be switched off in STANDBY mode to allow the nSTDBY, nRSTIN and WAKEUP pins to remain functional.

4 Example application

4.1 Example hardware

Figure 4 shows an example schematic for testing the STR71x power management features.

Figure 4. Example application schematic



Note: 1 The nSTDBY pin is connected to the tristate pin of the oscillator in order to disable the oscillator when the MCU is in STANDBY mode, in fact during this mode the nSTDBY pin is forced to low level.

2 P0.10 can be used to stop the oscillator before entering LPWFI and STOP modes, in this case the system clock must be previously switched to the 32 KHz RTC clock or the PLL free running mode clock.

3 P1.14 is configured as an external interrupt pin, it is used to wake up the MCU from LPWFI or STOP mode. This pin is linked through a diode to the oscillator because it is also used to re-enable the oscillator during wakeup from STOP mode.

4 P0.0 is used to indicate that the core is running (GPIO toggling) during RUN/SLOW modes.

5 The WAKEUP pin can be used to wake up the MCU from STANDBY mode.

4.2 Example software

A program is provided with this application note for testing the different low power modes, it includes the following source files:

Table 4. Software files

File	Description
LPmode.c	Routines for entering the various low power modes
main.c	Test software for testing different low power modes
71x_it.c	Interrupt service routines

The following routines are provided in the LPmode.c file:

Table 5. Low power modes routines

routine	Description
void SLOW (SlowClock_Typedef clock)	Enters MCU in SLOW mode
void WFI (void)	Enters MCU in WFI mode
void LPWFI (SlowClock_Typedef clock, functionalstate VRstate)	Enters MCU in LPWFI mode
void STOP (functionalstate Flashstate, functionalstate VRstate)	Enters MCU in STOP mode
void STANDBY (functionalstate VRstate)	Enters MCU in STANDBY mode

4.2.1 MAIN.C test routines

The following tests are implemented in the main.c file:

Table 6. Test routines

Test	Test conditions/ Options during the selected low power mode
LPWFI_test	- RCLK= CK_AF (32 kHz) - Main Voltage Regulator (MVR) stopped (software selectable) - Flash in power-down - 16 MHz oscillator stopped using pin P0.10 - wakeup by the RTC alarm interrupt after a 10s delay or pin P1.14 configured as external rising edge interrupt
STOP_test	- Flash in power- down (software selectable) - Main Voltage Regulator (MVR) stopped (software selectable) - 16 MHz oscillator stopped using pin P0.10 - Wakeup by pin P1.14 configured as rising edge wakeup
STANDBY_test	- Low Power Voltage Regulator (LPVR) stopped (external 1.8v on V_{18BKP}) (software selectable) - Wakeup by the RTC alarm (after 10 s), the WAKEUP pin or RESET pin.

Note: 1 To select a test you have to uncomment the corresponding define code '#define Test_xxx' and rebuild the main.c file.

2 For power consumption measurement results, please refer to the STR71x datasheet.

5 Revision history

Date	Revision	Changes
01-Dec-2005	1	Initial release.