



STR71x microcontroller power management

Introduction

This application note is intended for system designers who require a hardware implementation overview of the low power modes of the STR71x product family. It shows how to use the STR71x product family and describes the minimum hardware resources required to develop an STR71x application and to take power consumption measurements

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

Contents

- 1 Power supply 3**
 - 1.1 Power supply pins 3
 - 1.2 Internal regulators 4
 - 1.2.1 Main voltage regulator (MVR) 4
 - 1.2.2 Low power voltage regulator (LPVR) 5

- 2 Clock management 6**
 - 2.1 Peripheral clock gating 7
 - 2.2 PLL free running mode 8

- 3 Flash low power modes 9**
 - 3.1 Power-down mode 9
 - 3.2 Flash low-power (low-speed) 9

- 4 STR71x low power modes 10**
 - 4.1 Low power mode characteristics 10
 - 4.2 Guidelines for entering/exiting low power modes 12
 - 4.2.1 SLOW mode 12
 - 4.2.2 WAIT For Interrupt mode (WFI) 13
 - 4.2.3 Low Power WAIT For Interrupt mode (LPWFI) 13
 - 4.2.4 STOP mode 16
 - 4.2.5 STANDBY mode 17

- 5 Example application 18**
 - 5.1 Example hardware 18
 - 5.2 Firmware Example 19
 - 5.2.1 Low power mode examples 20

- 6 Revision history 22**

1 Power supply

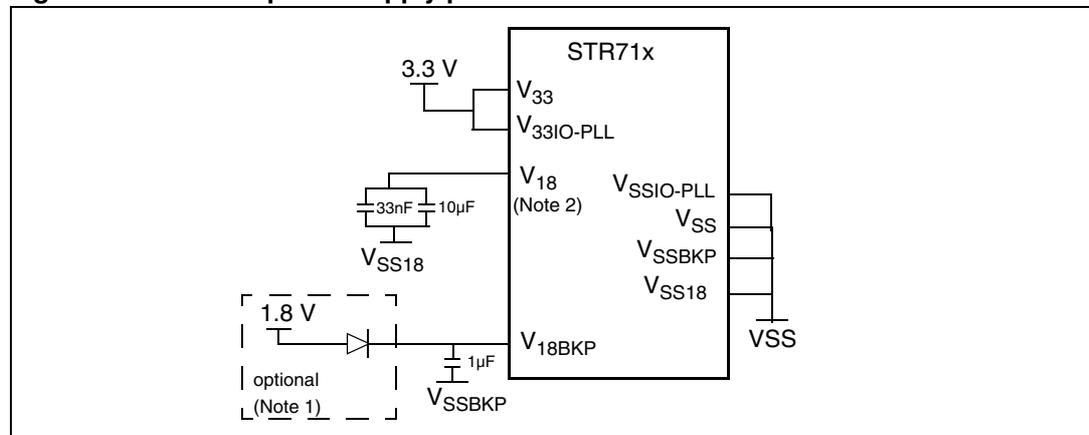
1.1 Power supply pins

Figure 1 shows the recommended configuration for the power supply pins:

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

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

Figure 1. STR71x power supply pins

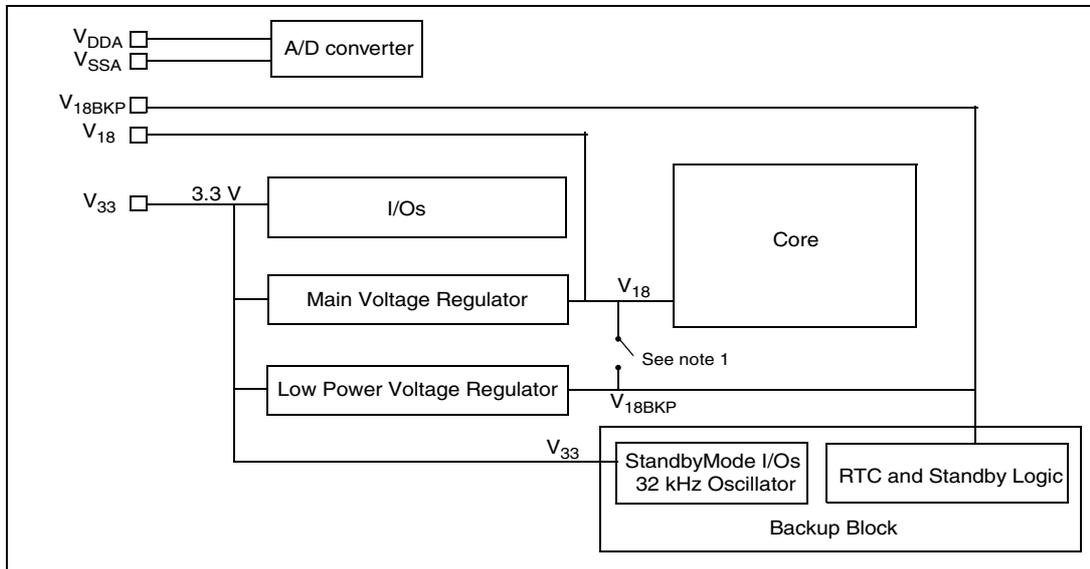


1. When not using an external 1.8 V on the V_{18BKP} pin, a $1\mu\text{F}$ capacitor must be connected between this pin and V_{SSBKP} to guarantee on-chip voltage stability.
2. Connecting an external 1.8 V supply to the V_{18} pins is not supported.

1.2 Internal regulators

Figure 2 provides a schematic view of the power management block of the STR71x.

Figure 2. Power management block



1. 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 4)

1.2.1 Main voltage regulator (MVR)

The Main Voltage Regulator (MVR) is able to generate sufficient current for the device to operate in any mode through ballast P-channel transistors located inside the I/O ring. It includes a bandgap reference for thermal compensation and it has a static power consumption of 100 μ A (typical).

- Note:**
- 1 The MVR is automatically switched off in Standby mode.
 - 2 The MVR can be configured (using the LPVRWFI bit in the PCU_PWRCCR register) to automatically switch off when the device enters Stop mode or LPWFI mode, leaving the Low Power VR as the only power supply.
 - 3 The MVR can be switched off using the VRBYP bit in the PCU_PWRCCR register, In this configuration the device is only powered by the Low Power VR, the maximum allowed operation frequency is 1 MHz and the PLL is disabled.
 - 4 When the VROK bit in the PCU_PWRCCR register is set by hardware this indicates that the main regulator output voltage is stabilized at the specification value.

1.2.2 Low power voltage regulator (LPVR)

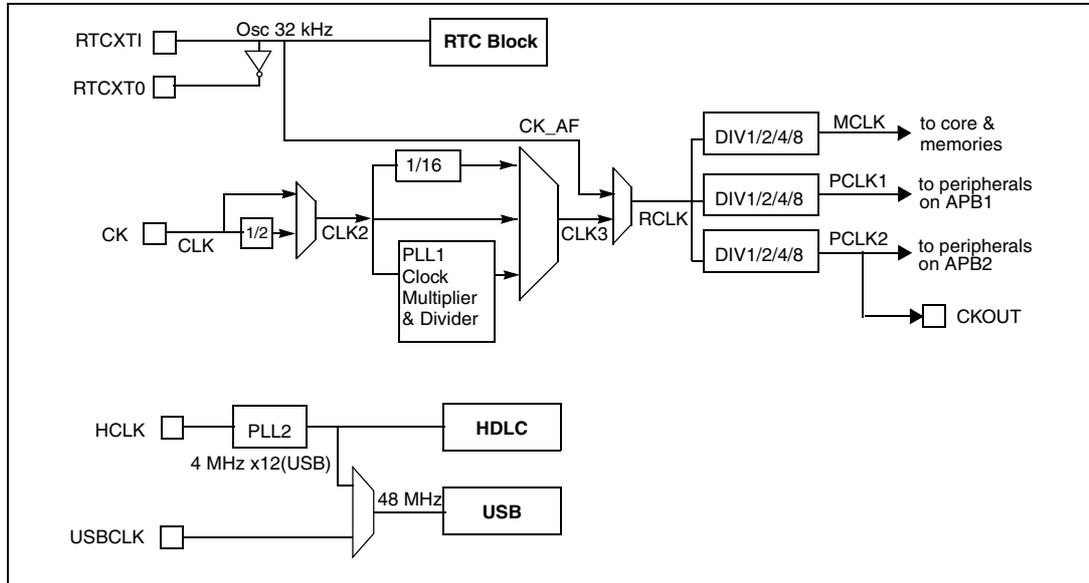
The Low Power Voltage Regulator (LPVR) is used when the MCU is in low-power mode and the main voltage regulator has been switched off. It has a different design from the main voltage regulator and generates a non-stabilized and non-thermally-compensated voltage of approximately 1.6V, its output current is not generally sufficient for the device to run in normal operation. Because of this limitation, the PLL is automatically disabled when the Main VR is switched off and the maximum allowed operating frequency is 1 MHz.

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

2 Clock management

The following figure gives the STR71x clock distribution scheme:

Figure 3. Clock distribution scheme



The source clock CLK is derived from an external oscillator, through the CK pin. This clock may be turned off during Low power modes. (refer to [Section 4](#)).

The system PLL (PLL1) is used to multiply the input internally, to generate the appropriate operating frequency. RCLK is the output of the system PLL or if enabled the CK_AF alternate source (32 kHz RTC clock).

Several clock domains exist in the device:

- Main Clock MCLK, including CPU, internal memories, External Memory Interface, PRCCU Registers (except RCCU registers)
- PCLK1, including APB1 peripherals (serial communication peripherals)
- PCLK2, including APB2 peripherals (system peripherals)

Each domain may use different frequencies independently by programming the various clock dividers. Having a divider dedicated to the CPU subsystem allows software to dynamically change CPU operating frequency, tailoring computing speed and power consumption to application needs, while maintaining a stable operation of all the peripherals.

On-chip peripherals, mapped in the APB memory space, make use of the RCLK output divided, independently from MCLK, by 2, 4 or 8. Wait states are automatically added by the bus bridge when accessing their registers.

Note that clock may be enabled/disabled independently to each peripheral. Each peripheral may also be reset under software control (Refer to [Section 2.1](#))

It is forbidden to access peripheral registers if the CPU clock MCLK is slower than the related peripheral clocks PCLK1 and PCLK2.

If the MCLK clock divider is set to a prescaling value other than 1 (i.e. if RCLK frequency differs from MCLK), it is not possible to access the RCCU registers since they are always clocked by RCLK.

For non-intensive operations, PLL1 may be disabled; in addition CLK2 may be divided by a factor of 16, to allow low-power operation while maintaining fast interrupt response.

System blocks (ARM7TDMI[®], PRCCU, on-chip memories and bridges) are driven by MCLK (Bus clock) and cannot be disabled by software in order to guarantee basic functionality.

A 32 kHz oscillator is present to maintain a real-time-clock (RTC), with programmable WAKEUP alarm. It may be deactivated if not required; when active it is not influenced by any low-power-mode switch.

The HDLC peripheral can optionally receive its reference clock from an external pin, and may dynamically change its frequency independently from CPU operation. An internal PLL (PLL2) allows the use of a low-frequency external signal, thus reducing power consumption and generated noise.

The USB Interface needs a precise 48 MHz clock reference. This may be generated either externally through the USBCLK pin, or by the internal PLL2, multiplying an external reference at lower speed, if PLL2 is not used by the HDLC interface.

If the USB interface is not used, set bits 0, 1 and 2 in the PCU_PLL2CR register to switch off PLL2 (and reduce power consumption).

To reduce power consumption, bits 0, 1 and 3 in the RCCU_PER register must be reset by the application software in the initialization phase. These bits are enabled by hardware at reset for factory test purposes only.

2.1 Peripheral 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

MX[1:0]	Free running mode frequency	
	FREE_RANGE = 0	FREE_RANGE = 1
'01', '11'	~125 kHz	~250 kHz
'00', '10'	~250 kHz	~500 kHz

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

3 Flash low power modes

3.1 Power-down mode

When the microcontroller is put in STOP or LPWFI mode, the Flash module can be put into two different low power modes.

- Normal Mode (default), Flash Stand-by mode is selected (immediate read from Flash is possible, but some residual power consumption is present).
- The Flash module enters Power-Down mode if the PWD bit is set in the FLASH_CR0 register. In this mode, a recovery time of 20 μ s needed before reading, but power consumption is lower.

3.2 Flash low-power (low-speed)

By default the Flash works in FAST mode, using BURST mode for sequential accesses, to allow zero wait state operation up to the maximum device frequency, and generating a WAIT cycle on non-sequential memory accesses.

It is possible to put the Flash into Low-Power mode, by setting the FLASH LP bit in the PCU_PWRCCR register. This disables BURST mode and WAIT states are never generated. LP mode may be used if the operating frequency (MCLK) is lower than 33 MHz.

4 STR71x low power modes

4.1 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 ⁽¹⁾
Wait For Interrupt (WFI)	<ul style="list-style-type: none"> – Core clock is stopped (MCLK off). – Wake-up by peripheral interrupts acknowledged by the interrupt controller (EIC). – MCU state retained (context restored after wakeup).
Low Power Wait For Interrupt 1 (LPWFI1)	<ul style="list-style-type: none"> – Core clock stopped (MCLK off). – Peripherals running at slow clock: RCLK = CLK2/16. – PLL1 switched off automatically. – FLASH can be set to power down mode automatically when LPWFI is entered. – MCU state retained. – Wake-up by peripheral interrupts acknowledged by the EIC, and CLK2 is set as system clock automatically.
Low Power Wait For Interrupt 2 (LPWFI2)	<ul style="list-style-type: none"> – Core clock stopped (MCLK off). – Peripherals running at slow clock speed: RCLK = CK_AF. – PLL1 can be switched off automatically if CK_AF has been selected. – FLASH can be set to power down mode automatically when LPWFI is entered. – MVREG can be set to be switched off automatically when LPWFI is entered. – MCU state retained. – Wake-up by peripheral interrupts acknowledged by the EIC and CK_AF remains the system clock (RCLK)
SLOW1	<ul style="list-style-type: none"> – CLK2 is set as system clock. – PLL1 can be switched off (DX[2:0]) – FLASH can be set to LP mode.
SLOW2	<ul style="list-style-type: none"> – CLK2 16 is set as system clock. – PLL1 can be switched off (DX[2:0]) – FLASH can be set to LP mode. – MVREG can be switched off.
SLOW3	<ul style="list-style-type: none"> – CK_AF is set as system clock. – PLL1 can be switched off automatically if CK_AF has been selected. – FLASH can be set to LP mode. – MVREG can be switched off.

Table 2. STR71x low power modes (continued)

Low Power mode	Description ⁽¹⁾
SLOW4	<ul style="list-style-type: none"> – PLL1 configured to free running mode. – PLL1 output is set as system clock. – FLASH can be set to LP mode. – MVREG can be switched off.
STOP	<ul style="list-style-type: none"> – Core and peripherals (on APB1, APB2) clocks stopped (RCLK off). – PLL1 switched off automatically. – FLASH can be set to power down mode automatically when STOP is entered. – MVREG can be set to be switched off automatically when STOP is entered. – Wake-up by the configured external wake-up lines (XTI unit). – MCU state retained.
STANDBY	<ul style="list-style-type: none"> – Core, memories and peripherals switched off⁽²⁾ (except RTC and wakeup logic). – Main voltage regulator switched off. – All I/Os are forced to high impedance (except the Standby I/Os)⁽³⁾. – Wake-up by: the WAKEUP pin, the RTC alarm and the RESET pin. – MCU Reset after wake-up.

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

2. Core, memories and peripherals not powered by 1.8V internal supply.

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

4.2 Guidelines for entering/exiting low power modes

4.2.1 SLOW mode

To enter SLOW mode, RCLK must be configured as CLK2, CLK2/16, CK_AF (32 kHz clock) or PLL1 output configured in free running mode.

To reduce power consumption, you can turn off the PLL1 by setting bits DX[2:0] 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⁾	x	x	1

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 (32 kHz) as the system clock, this can be done when setting bit CKSTOP_EN in the RCCU_CFR register.

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

4.2.2 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.

4.2.3 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_PWRCCR register.
- Put the FLASH in power-down mode by setting bit PWD in the FLASH_CR0 register (refer to the STR7 Flash programming reference manual)
- When selecting the CK_AF(32 kHz) 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 from LPWFI mode, if the clock selected during LPWFI mode is CK2_16, the system clock (RCLK) switches automatically to CK2.
After wakeup from LPWFI mode, if the clock selected during LPWFI mode is CK_AF, this clock remains the system clock (RCLK).
 - 3 Refer to [Figure 4](#) and [Figure 5](#) for examples.

Figure 4. Example of LPWFI mode using CK_AF

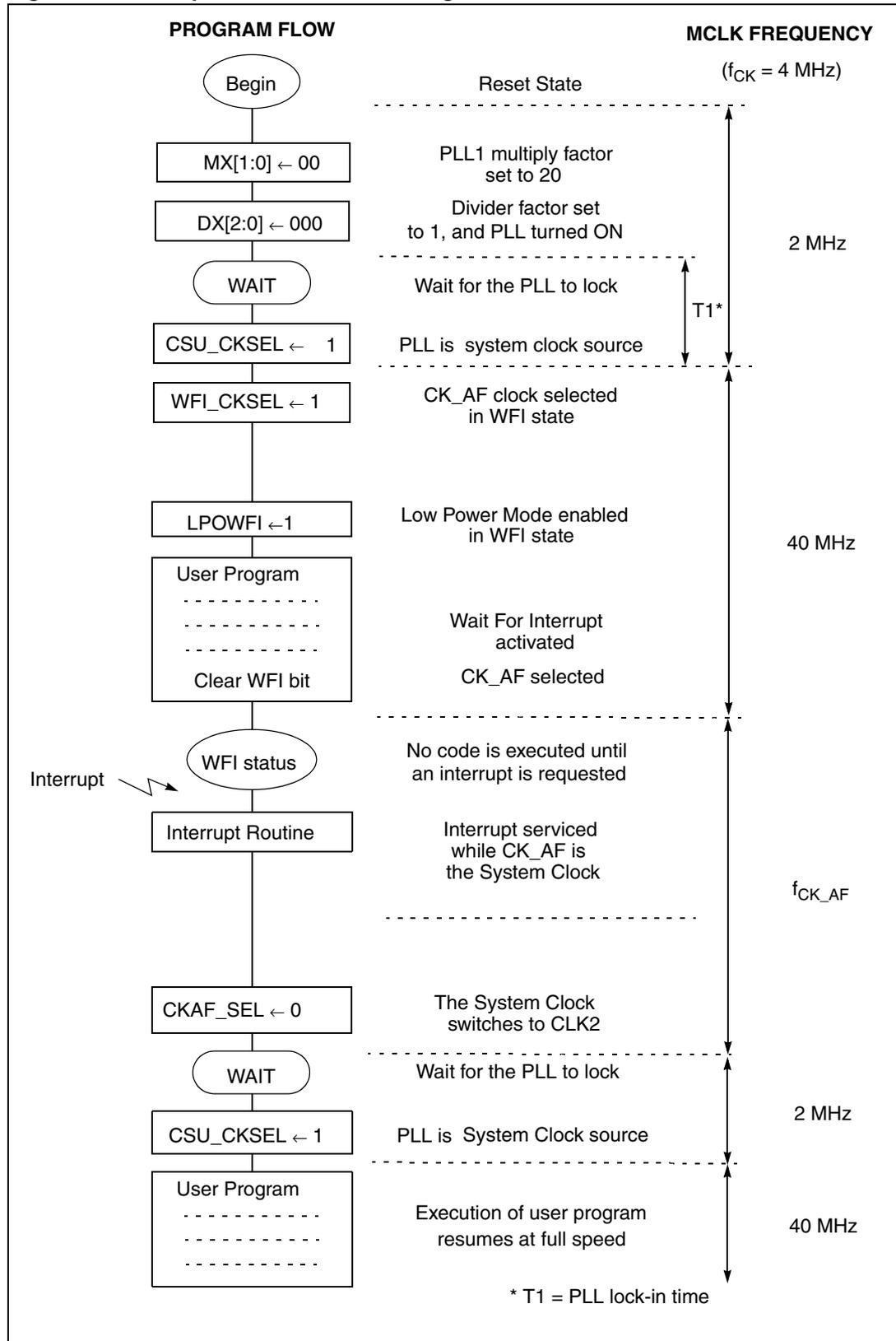
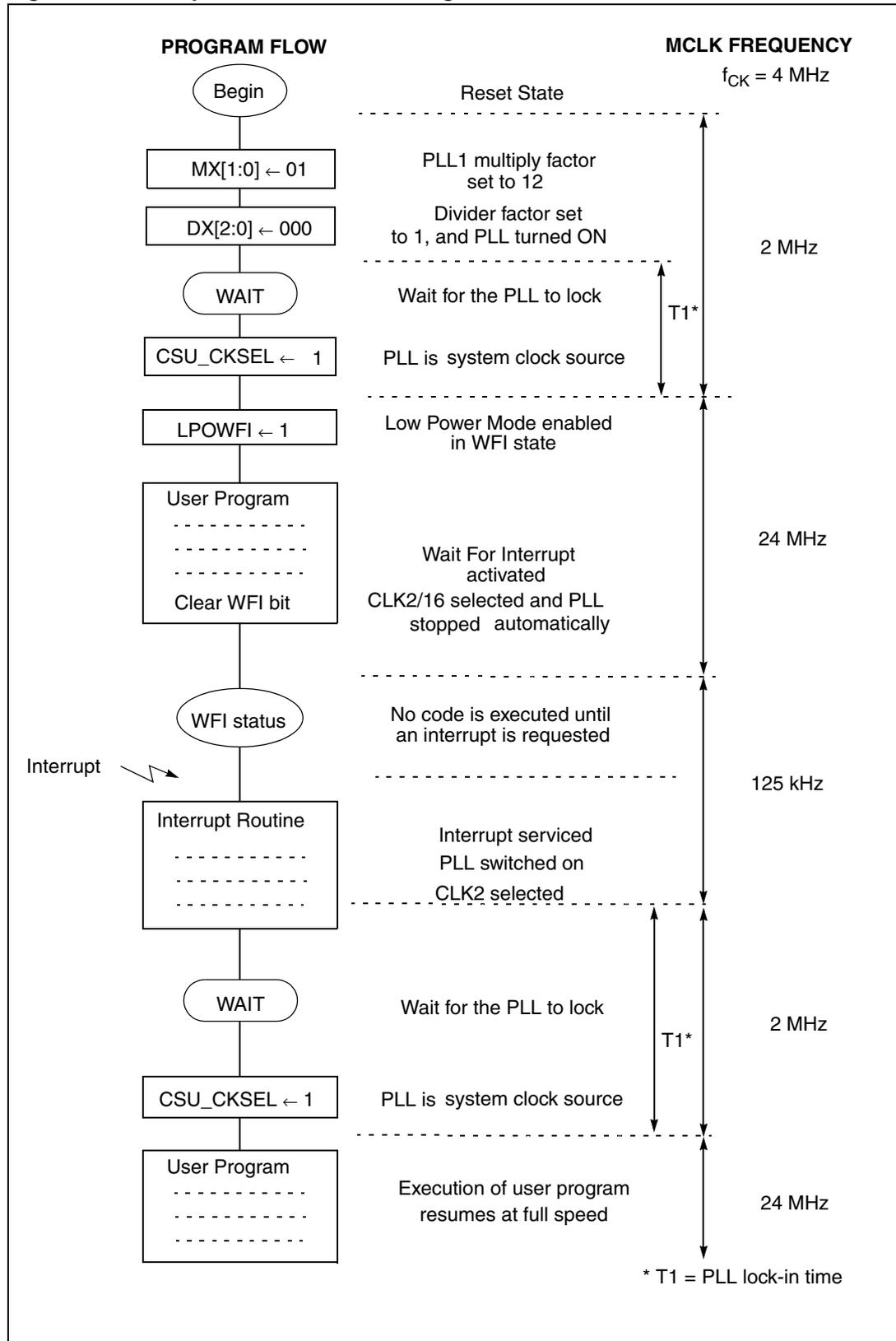


Figure 5. Example of LPWFI mode using CLK2/16



4.2.4 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 RCCU 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).

The MCU resumes program execution after a delay of 2048 CLK clock periods after the Stop mode wakeup event.

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_PWRCCR 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 re-enabled in order to exit from STOP mode (see Example Application in [Section 5](#)).

4.2.5 STANDBY mode

The STANDBY sequence is initiated either:

- By setting the PWRDWN bit in the PCU_PWRCCR 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_PWRCCR register

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

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...).

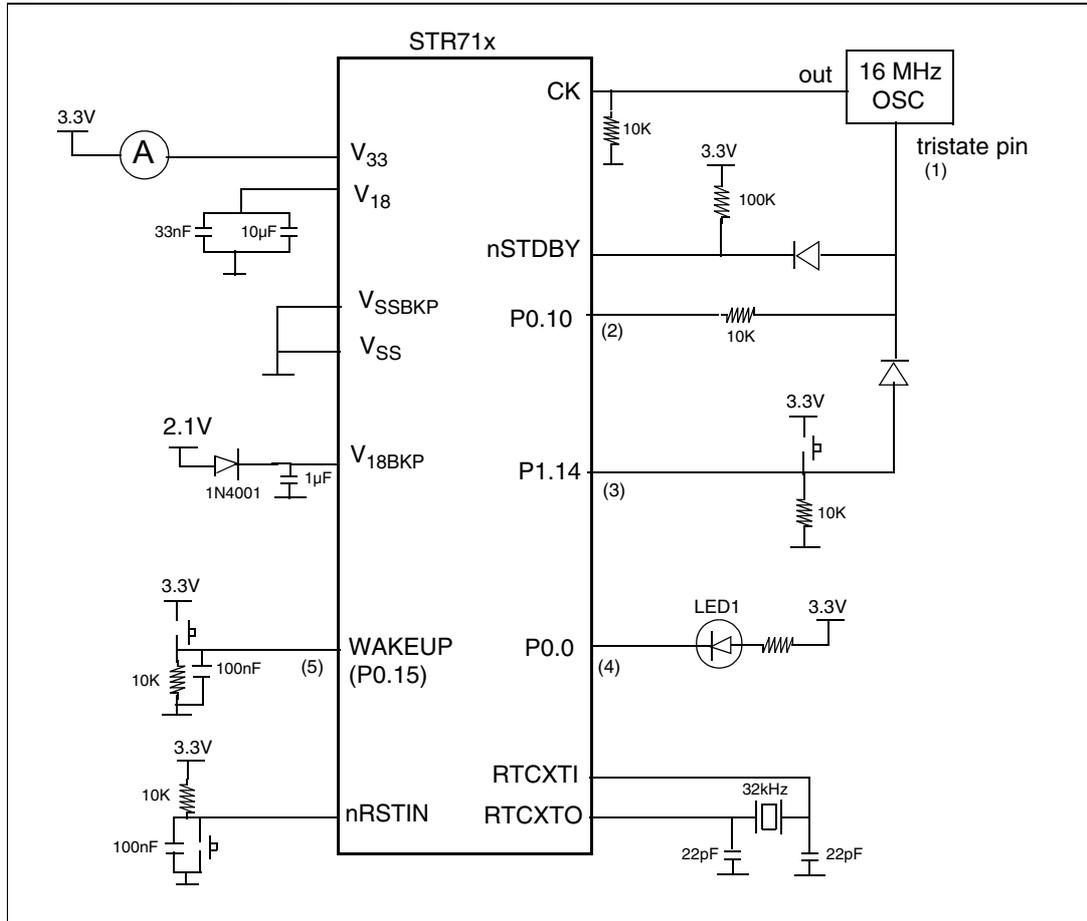
Caution: Because it powers the I/Os, V33 must not be switched off in STANDBY mode to allow the nSTDBY, nRSTIN and WAKEUP pins to remain functional.

5 Example application

5.1 Example hardware

Figure 6 shows an example schematic for testing the STR71x power management features.

Figure 6. Example application schematic



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 STOP or LPWFI 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 STOP or LPWFI mode. This pin is linked through a diode to the oscillator because it is also used to re-enable the oscillator during wake-up from STOP mode.
4. P0.0 is used to indicate that the core is running (GPIO toggling) during RUN/SLOW mode and when system wake-up from low power mode.
5. The WAKEUP pin can be used to wake up the MCU from STANDBY mode, WFI mode.

5.2 Firmware Example

There are 5 folders in the ZIP file provided with this application note giving an example of low power mode configuration (SLOW, WFI, LPWFI, STOP and STANDBY):

Directory content:

- *71x_conf.h*: Library Configuration file
- *71x_it.c*: Interrupt handlers
- *71x_it.h*: Interrupt handlers header file
- *main.c*: Main program

The **project** subfolder contains projects that compile the example files with the various standard toolchains:

- **EWARM**: contains the project for the EWARM toolchain
- **RVMDK**: contains the project for the RVMDK toolchain
- **RVDK**: contains the project for the RVDK toolchain
- **RIDE**: contains the project for the RIDE toolchain

The following routines are used to configure and enter the low power modes:

Table 4. Low power mode routines

Routine	Description
PCU_LPModesConfig	This routine is used to configure the STR71x low power modes: <ul style="list-style-type: none"> – PLL1 state when CK_AF is selected – MVREG and FLASH states in LPWFI and STOP modes – LP mode during WFI – LPWFI clock
RCCU_RCLKSourceConfig	This routine is used for clock selection in Slow mode.
PCU_WFI	Causes the MCU to enter WFI / LPWFI mode.
PCU_STOP	Causes the MCU to enter STOP mode.
PCU_STANDBY	Causes the MCU to enter STANDBY mode.

5.2.1 Low power mode examples

The following table summarizes the low power mode examples provided:

Table 5. Low power mode configuration

Low power mode	Configuration/Options during the selected low power mode
SLOW mode	<p>FLASH in LP mode, MVR disabled, PLL2 disabled.</p> <ol style="list-style-type: none"> 1. Three examples of SLOW mode configuration can be selected: <ol style="list-style-type: none"> a. SlowMode_CLK2_16: System clock configured to CLK2_16 (by uncommenting the line #define SlowMode_CLK2_16 in the 71x_conf.h file). b. SlowMode_CK_AF (by uncommenting the line #define SlowMode_CK_AF in the 71x_conf.h file). c. SlowMode_PLL1FreeRunning (by uncommenting the line #define SlowMode_PLL1FreeRunning in the 71x_conf.h file2). 2. Disable PLL1. 3. Set the Flash to LP mode. 4. Disable MVR. 5. Toggle GPIO pin P0.0.
WFI Mode	<ol style="list-style-type: none"> 1. Set RCLK= MCLK=PCLK1=PCLK2= CLK2_16=500 kHz, PLL1 and PLL2 disabled. 2. Configure XTI to generate interrupts on P0.15. 3. Configure the EIC to acknowledge XTI interrupt. 4. Toggle the GPIO pin P0.0 5. Enter WFI mode. 6. Go back to step 4. 7. Clear the XTI pending bit in the XTI ISR.
LPWFI mode	<ol style="list-style-type: none"> 1. Set RCLK= MCLK=PCLK1=PCLK2= CLK2_16=500 kHz, PLL1 and PLL2 disabled. 2. Configure XTI to generate interrupts on P0.15. 3. Configure the EIC to acknowledge XTI interrupt. 4. Put the Flash in power down mode and disable MVREG in LP_WFI mode. 5. Toggle the GPIO pin P0.0. 6. Enter LP_WFI mode with RCLK = CLK2_16 or with RCLK=CK_AF by uncommenting the "#define __LPWFI_CK2_16" or "#define __LPWFI_CK_AF" lines in the 71x_conf.h file respectively. 7. Go back to step 5. 8. Clear the XTI pending bit in the XTI ISR <p>Note: It is possible to use the RTC to wake-up the system from LPWFI mode if RCLK=CK2_16 in LPWFI mode. This is selected by uncommenting the "#define Wakeup_RTC" line in the 71x_conf.h file. The RTC clock must be <i>at least 4 times slower than PCLK2 clock</i> even in low power mode. Refer to section 5.3.4 in the STR71x reference manual.</p>

Table 5. Low power mode configuration (continued)

Low power mode	Configuration/Options during the selected low power mode
STOP mode	Flash in PWD mode, MVR disabled and PLL2 disabled. 1. Configure XTI to generate interrupts on P0.15. 2. Configure the EIC to acknowledge XTI interrupt. 3. Configure MVREG/FLASH to be disabled/PWD during STOP mode. 4. Set the RCLK to the CK_AF clock source. 5. Toggle the GPIO pin P0.0. 6. Stop the external oscillator using a GPIO (P0.10). 7. Enter STOP mode. 8. Enable the external oscillator. 9. Go back to step (4). 10. Clear the XTI pending bit in the XTI_ISR.
STANDBY mode	1. Set the PLL1 to 24 MHz, RCLK = PLL1 output. 2. Configure the RTC. 3. Toggle a GPIO pin. 4. Configure RTC alarm after RTC_ALARM_DELAY. 5. Enter STANDBY mode.

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

6 Revision history

Table 6. Document revision history

Date	Revision	Changes
01-Dec-2005	1	Initial release
03-Dec-2007	2	Reference to additional clock added, Section 2 on page 6 New section added, Section 3: Flash low power modes on page 9 MVR description modified, Section 1.2.1 on page 4 Table 2: STR71x low power modes on page 10 updated. Section 4.2.3: Low Power WAIT For Interrupt mode (LPWFI) on page 13 text updated In Section 5.2 on page 19: Table 5 updated.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER'S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2007 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com

