STM32H7A3/7B3 lines and STM32H7B0 Value line smart power management Expansion Package for STM32Cube

Introduction

This application note provides guidelines on STM32H7A3/7B3 lines and STM32H7B0 Value line smart power management Expansion Package for STM32Cube. STM32H7A3/7B3 lines and STM32H7B0 value line microcontrollers include Arm® Cortex®- M7 core devices with double-precision floating point unit running at up 280 MHz. The system embeds a dual-power domain architecture operating independently to optimize power efficiency.

This document is split into two parts:

The first part gives an overview of the system architecture, power domains with the SRD domain, STM32H7A3/7B3 lines and STM32H7B0 Value line devices power modes. It describes the low-power modes, the application, and configuration of the HSLV (high-speed low voltage) and the activation/deactivation process. The new DSTOP2 retention mode and its impact on reducing the power consumption of the system is detailed, along with information on inrush current.

The second part provides an example of temperature acquisition based on I^2C transmission, using the X-NUCLEO-IKS01A2 expansion board based on STM32H7A3ZI MCUs. The purpose of this example is to highlight the smart power management of STM32H7A3/7B3 lines and STM32H7B0 Value line devices:

- when using two power domains
- when minimizing power consumption while keeping some activities running when required (autonomous mode).

This document is provided with the X-CUBE-PWRMGT-H7 Expansion Package with one project and six different workspaces as listed below.

- STM32H7A3ZI_Mode1: temperature acquisition with CPU in CRun mode, CD in DRun mode and SRD in SRDRun mode
- STM32H7A3ZI_Mode2: temperature acquisition with CPU in CSleep mode, CD in DRun mode and SRD in SRDRun mode
- STM32H7A3ZI _Mode3: temperature acquisition with CPU in CStop mode, CD in DStop mode and SRD in SRDRun mode
- STM32H7A3ZI _Mode4: temperature acquisition with CPU in CStop mode, CD in DStop2 mode and SRD in SRDRun mode
- STM32H7A3ZI _Mode5: temperature acquisition with CPU in CStop mode, CD in DStop mode, and SRD switching between Run and Stop modes
- STM32H7A3ZI _Mode6: temperature acquisition with CPU in CStop mode, CD in DStop2 mode, and SRD switching between Run and Stop modes

This document applies to STM32H7A3/7B3 lines and STM32H7B0 Value line and is referred to as STM32H7A3/7Bx in the document. For further information on STM32H7A3/7B3 lines and STM32H7B0 Value line devices, refer to [R1] and [R2].

1 General information

This document applies to STM32H7A3/7B3 and STM32H7B0 Value line microcontrollers Arm® Cortex®-based devices.

Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

List of acronyms

Table 1. List of acronyms

Acronym	Description
eNVM	Embedded nonvolatile memories
LDO	Low-drop out voltage regulator
SMPS	Switch-mode power supply
HSLV	High-speed low voltage
WFI	Wait for interrupt
WFE	Wait for event

Reference documents

Table 2. Reference documents

Reference	Document title
[R1]	STM32H7A3/7B3 and STM32H7B0 Value line advanced Arm®-based 32-bit MCUs: reference manual (RM0455)(1)
[R2]	32-bit Arm® Cortex®-M7 280 MHz MCUs, up to 2-Mbyte Flash memory, 1.4-Mbyte RAM, 46 com. and analog interfaces, SMPS: datasheet (DS13195)(1)
[R3]	32-bit Arm® Cortex®-M7 280 MHz MCUs, up to 2-Mbyte Flash memory, 1.4-Mbyte RAM, 46 com. and analog interfaces, SMPS: datasheet (DS13195)(1) Refer to section 6.3 Operating conditions. For mode 1 and 2 refer to table 35 and table 39, for mode 3, 4, 5 and 6 refer to table 38 and table 40.

1. *Available at www.st.com.*

2 System architecture

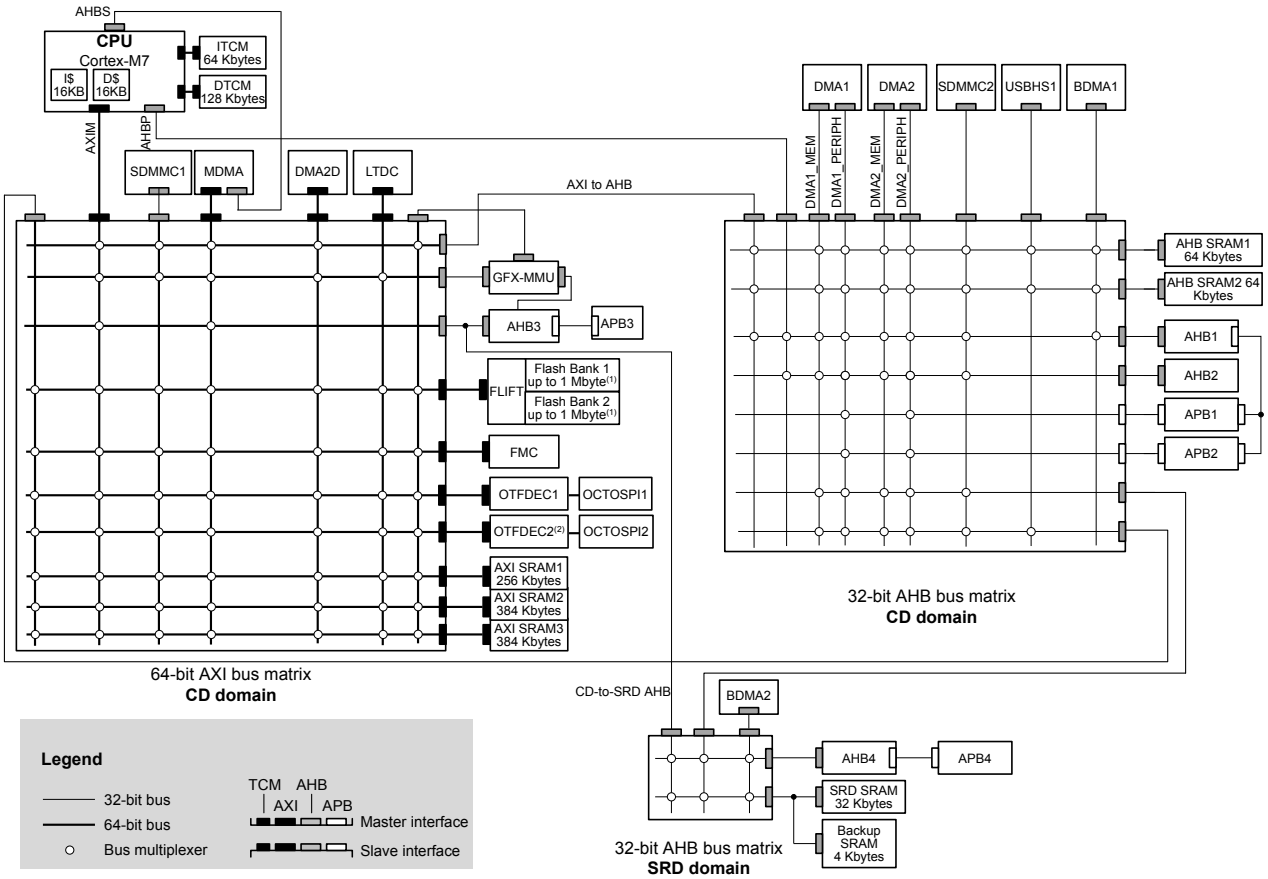
The advanced eNVM (embedded nonvolatile memories) technology and architecture used in STM32H7A/7Bx devices enable a frequency of the core of 280 MHz. The multiple power domains provide a smart solution to power-consumption challenges.

The system is partitioned as follows:

- One CPU subsystem Arm® Cortex®-M7 core, with associated peripheral clocks according to CPU activity.
- Two power domains:
  - CD domain: a high-bandwidth and high-performance domain with the Cortex®-M7 core and acceleration mechanisms. This domain encompasses high-bandwidth features and smart management thanks to the AXI and AHB bus matrices. It is also an I/O processing domain, that contains most peripherals and requires less bandwidth.
  - SRD domain: designed to manage the low-power mode feature. It embeds the system configuration block to keep the system state, and the GPIO status. This domain is designed to be autonomous. It embeds a 32-Kbyte RAM and a subset of peripherals to run basic functions. While the CD domain can be shut off to save power.

The figure below shows the system architecture of STM32H7A/7Bx microcontrollers.

Figure 1. STM32H7A/7Bx system architecture



1. Bank 1 is limited to 128 Kbytes on STM32H7B0 devices. STM32H7A3xG and STM32H7A3xI/7B3xx devices feature two banks of 512 Kbytes and 1 Mbyte each, respectively.
2. OTFDEC1 and OTFDEC2 are available only on STM32H7B0 and STM32H7B3 devices.

3 System supply configurations

The V_{CORE} domain is supplied:

- either by the internal linear voltage regulator
- either by the embedded SMPS step down converter
- or directly by an external supply voltage (regulator bypass).

The SMPS step down converter can also be cascaded with the linear voltage regulator. For devices, which do not support the SMPS feature, only the LDO regulator is available.

V_{CORE} is the supply voltage for the digital circuit of the CPU domain and the smart run domain. The V_{CORE} domain is split as follows:

- CD domain containing the CPU (Cortex®-M7), flash memory, memories interface, graphical, analog, and digital peripherals.
- SRD domain containing the system control, I/O logic and low-power peripherals. The different supply configurations are controlled through the SMPSLEVEL, SMPSEXTHP, SMPSEN, LDOEN and BYPASS bits in PWR control register 3 (PWR_CR3). The choice of the V_{CORE} domain supply can be done only once, before configuring the system clock source.
- For more details about the supply configuration control, refer to the power supplies section in [R1].

4 Voltage regulator (LDO) supply

The LDO (low-drop out voltage regulator) supplies both power domains CD and SRD. CD can be set in low-power mode independently. The voltage regulator is always enabled after reset. For the system supply configuration where this regulator is not required, the end user must switch it off after startup of the system. For configurations where the V_{CORE} is supplied by the LDO, the default output level is set to 1.0 V (VOS3).

For more accurate values on the voltage-scaling output, refer to [R2].

5 SMPS step-down converter supply

The embedded SMPS (switch-mode power supply) step down converter has a higher efficiency than the embedded LDO regulator. Using the SMPS, improves the overall system power consumption for all power modes thanks to the additional external components (inductor and low ESR capacitor). For more information to compare power efficiencies refer to [R1].

The SMPS step down converter is always enabled after reset when its power supply is provided on the V_{DDSMPS} pin. The regulated output at startup is 1.2 V.

Note: For more accurate values on the voltage-scaling output, refer to [R2].

6 SMPS and LDO regulators mode

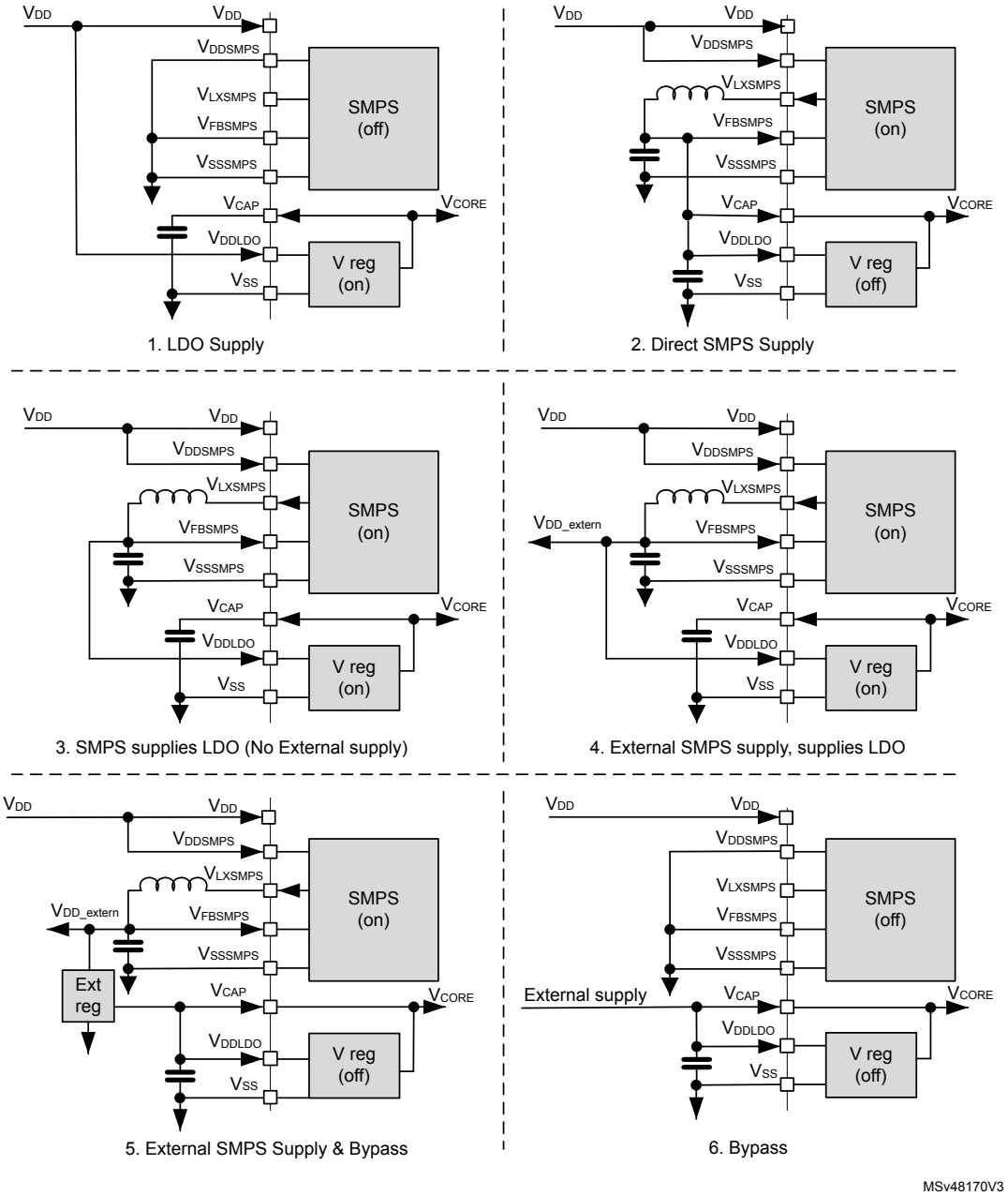
The figure below gives an overview of all possible use cases of the regulator:

- LDO power supply
- direct SMPS power supply
- LDO SMPS power supply
- LDO external SMPS power supply
- external SMPS power supply and bypass
- bypass

In STM32H7A3/7Bx microcontrollers, the software manages the regulator bypass through LDOEN, and BYPASS bits in the PWR_CR3 register. An external power supply delivers V_{CORE} through V_{CAP} pin. At system startup, the MCU starts under LDO. It then switches to the regulator-bypass mode to modify the LDOEN and BYPASS bits with the software.

The figure below illustrates the system supply configuration for STM32H7A/7Bx microcontrollers. For more information on external capacitor values refer to [R2].

Figure 2. System supply configuration



The STM32H7A3/7Bx microcontrollers embed an SMPS step-down converter and an LDO. There are several possible power-supply configurations to provide the digital power supply as follows:

- the internal linear voltage regulator
- the embedded SMPS step down converter
- or directly an external supply voltage in regulator bypass mode.

The SMPS step-down converter can also be cascaded with the linear voltage regulator.

Note: It is important to connect the device according to the schematics shown in Figure 2 to ensure a correct power-up of the device.

For devices not supporting the SMPS feature, only the LDO regulator is available. In this case, two configurations are possible and the V_{DDLDO} regulator supply is directly provided through V_{DD} without a dedicated package pin.

7 Power control modes

There are two power control modes available, the operating mode and the low-power mode.

7.1 Operating modes

The operating modes allow the control of the clock distribution on the different system blocks and to power them. The system operating mode is driven by the CPU subsystem and the system SmartRun autonomous wake-up. The CPU subsystem includes multiple domains depending on its peripheral allocation.

The operating modes available on the different system blocks are detailed in Table 3.

Table 3. Subsystem operating mode

CPU subsystem and domains	Mode	Description
CPU subsystem	CRun	CPU and CPU subsystem peripherals allocated via PERxEN bits in the RCC registers, are clocked.
	CSleep	The CPU clock is stalled, and the CPU subsystem allocated peripheral clock operates according to PERxLPEN bits in the RCC registers.
	CStop	CPU and CPU subsystem peripheral clocks are stalled. When the CPU subsystem is in CStop mode, the CPU domain is either in DStop or DStop2.
CPU domain	DRun	The domain bus matrix is clocked. The CPU subsystem operates in CRun or CSleep mode.
	DStop	The domain bus-matrix clock is stalled. The CPU subsystem operates in CStop mode and the RETDS_CD bit of PWR CPU control register (PWR_CPUCR) selects the DStop or DStop2 mode. The SRD domain peripherals are able to operate in Stop mode otherwise no peripherals in the CPU domain are operational.
	DStop2	
SRD domain	Run	The system clock and the SmartRun domain-bus matrix clock are running. • The CPU subsystem is in CRun or CSleep mode, or • A wake-up signal is active (system SmartRun autonomous mode). Run mode is entered after a POR reset and a wake-up from standby mode. The system supply configuration must be programmed in PWR control register 3 PWR_CR3. The system enters Run mode only when the ACTVOSRDY bit in PWR control status register 1 PWR_CSR1 is set to 1.
	Stop	The system clock and the SmartRun domain-bus matrix clock are stalled.(1) • The CPU domain is in DStop or DStop2 mode • The CPU subsystem is in CStop • All wake-up signals are inactive • The PDDS_SRD bit of PWR CPU control register (PWR_CPUCR) selects the Stop mode
	Standby	The system is powered down. • The CPU domain and CPU subsystem are powered off • The wake-up pins and RTC can wake up from Standby mode

1. *When the system oscillator HSI or CSI is used, HSIKERON and CSIKERON control the state, otherwise the system oscillator is off.*

7.2 Low-power modes

Several low-power modes are available to save power when the CPU does not execute code. This is, for example, when waiting for an external event. The end user must select the mode that provides the best compromise between low power-consumption, short start-up-time, and available wake-up sources for the application.

The features of the low-power modes for STM32H7A3/7Bx microcontrollers are:

- CSleep: CPU clock stopped
- CStop: CPU subsystem clock stopped
- DStop: domain bus-matrix clock stopped
- Stop: system clock stopped
- Standby: domain/system powered down

There are several ways to reduce power consumption on STM32H7A3/7Bx microcontrollers:

- Decreasing dynamic power consumption by slowing down the system clocks. This is done even in Run mode and by individually clock-gating peripherals not used.
- Saving power consumption when the CPU is idle, by selecting among the available low-power modes. This is made according to the requirements of the end-user application. This is the best compromise between short startup time, low-power consumption, and available wake-up sources.

The devices have several low-power modes:

- System run with CSleep:
  - CPU clock stopped
- Autonomous with CD domain in DStop:
  - CPU and CPU domain bus-matrix clocks stopped
- Autonomous with CD domain in DStop2:
  - CPU and CPU domain bus matrix clocks stopped, and CPU domain in retention mode
- System stop:
  - SRD domain clocks stopped and CD domain in DStop that is CPU and CPU domain bus matrix clocks stopped
- System stop:
  - SRD domain clocks stopped and CD domain in DStop2, that is CPU and CPU domain bus matrix clocks stopped, CPU domain in retention mode
- Standby:
  - system, CD, and SRD domains powered down

CSleep and CStop low-power modes are entered by the MCU when executing the WFI (wait for interrupt) or WFE (wait for event) instructions. These modes are also entered by the MCU when SLEEPONEXIT bit of the Cortex®-M7 core is set after returning from an interrupt service routine. The CPU domain enters low-power mode (DStop or DStop2) when the processor, its subsystem, and the peripherals allocated in the domain enter in low-power mode.

If part of the domain is not in low-power mode, the domain remains in the current mode. The system enters Stop mode or Standby mode when all EXTI wake-up sources are cleared and when the power domains are in DStop or DStop2 mode.

Table 4. System and domains low-power mode

System power mode	CD domain power mode	SRD domain power mode
Run	DRun/DStop/DStop2	DRun
Stop	DStop/DStop2	DStop
Standby	Standby	Standby
Autonomous	DStop/DStop2	SRDRun

8 DStop2

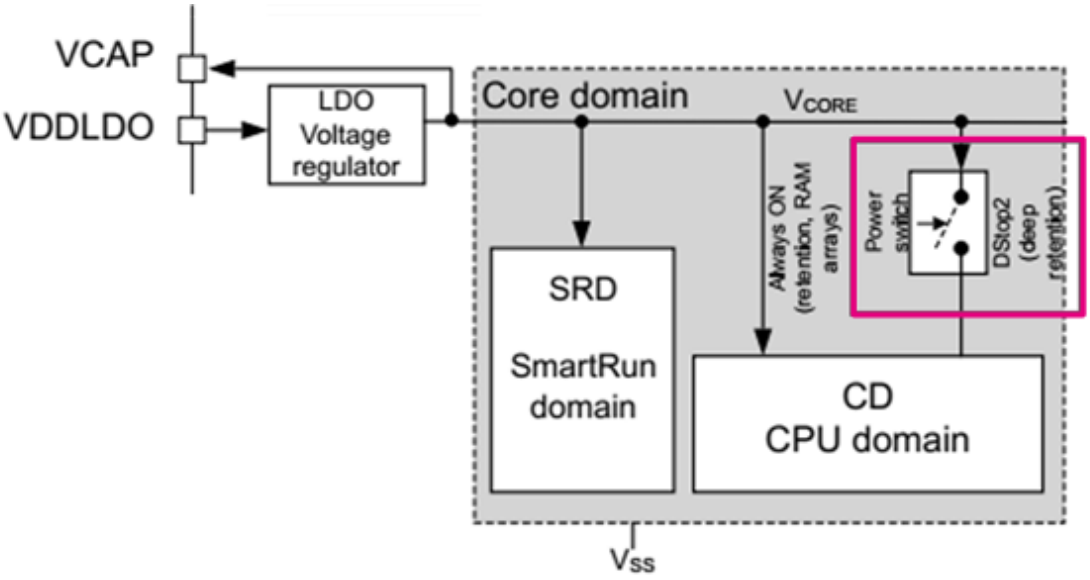
The DStop2 is a new mode in STM32H7A/7Bx microcontrollers, where part of the CPU domain-digital circuitry switches off, while the data retention is kept. This includes memory and register settings (see figure below). DStop2 mode significantly improves low-power consumption at a cost of a negligible wake-up time increase. The content of all memories is retained in DStop2.

The main differences between DStop and DStop2 are given below:

- In DStop mode, the entire logic is still supplied.
- In DStop2 mode, memories and registers are maintained, while asynchronous logic is switched off. This allows further leakage current reduction. When exiting DStop2, the CPU domain can resume normal execution.

The system state is retained in DStop and DStop2 as shown in the figure below. The CPU domain is in retention mode (DStop) or deep-retention mode (DStop2).

Figure 3. System supply



The CPU domain enters DStop or DStop2 depending on the configuration of RETDS_CD bit of PWR_CPUCR register.

Before entering DStop2, all the peripherals that belong to the CPU domain and have a kernel clock must be:

- either disabled by clearing the enable bit in the peripheral itself
- or reset by setting the corresponding bit in the associated AHB peripheral reset register (RCC_AHBxRSTR)
- or APB peripheral reset register (RCC_APBxRSTR).

In DStop2 mode, all the CD domain clocks are off, but data is retained. The power consumption in DStop2 mode can be further optimized by choosing to shut off some SRAMs with the consequence to lose their content. The PWR block controls the VCORE supply according to the system operating mode (CRun, CSleep, or CStop). The PWR block also controls the power switch (ePODs) to set the CD domain in retention mode (DStop2). The CPU domain can be set in a specific retention level. This is known as DStop2, whereby the logic is switched off and the register content is retained. The selection between DStop or DStop2 is made through the RETDS_CD bit of PWR CPU control register PWR_CPUCR. In DStop2 mode, the content of the memory blocks is maintained. Further power optimization can be obtained by switching off some memory blocks. This optimization implies the loss of the memory content. The end user can select which memory is discarded during Stop mode.

This is by means of xxSO bits in PWR control register 1 (PWR_CR1) as follows to keep the content of:

- SmartRun domain AHB memory in DStop2 mode, SRDRAMSO bit must be set
- USB and FDCAN memories in DStop2 mode, HSITFSO bit must be set
- GFXMMU and GFXMMU memories in DStop2 mode, GFXSO bit must be set.
- ITCM and ETM memories in DStop2 mode, ITCMSO bit must be set
- AHB SRAM2 in DStop2 mode, AHBRAM2SO bit must be set
- AHB SRAM1 in DStop2 mode, AHBRAM1SO bit must be set
- AXI SRAM3 in DStop2 mode, AXIRAM3SO bit must be set
- AXI SRAM2 in DStop2 mode, AXIRAM2SO bit must be set
- AXI SRAM1 in DStop2 mode, AXIRAM1SO bit must be set

Before entering DStop2 mode, the flash memory must be configured in low-power mode by setting the FLPS bit of PWR_CR1. This results in a power consumption improved with a slightly longer wake-up time.

Note: It is mandatory to set FLPS before entering DStop2 mode.

The core domain cannot be switched to DStop2 when the embedded flash memory is busy: BSY1/2, QW1/2, WBNE1/2 is set). These bits must be cleared in order to allow the microcontroller to switch the domain to DStop2 mode.

When entering DStop2 domain it generates a reset, the NRST_STOP bits of FLASH_OPTSR_CUR and FLASH_OPTSR_PRG registers are then set.

9 SRD autonomous mode

The autonomous mode permits basic operations with an inactive CPU domain in low-power modes, namely in SRD in Run mode.

The autonomous mode is entered when the SLEEPDEEP bit in the Cortex®-M system control register is set. The autonomous mode is exited by enabling an EXTI interrupt or event depending on how the low-power mode is entered. In autonomous mode, the system can wake up from Stop mode by enabling an EXTI wake-up, without waking up the CPU subsystem.

Refer to Table 5 for more details for entering and exiting the autonomous mode.

Table 5. Autonomous mode

Autonomous mode	Description
Mode entry	WFI or WFE when: • SLEEPDEEP = 1 (refer to Cortex®-M system control register.) • CPU NVIC interrupts and events cleared • ALL CPU EXTI wake-up sources are cleared
	On return from ISR when: • SLEEPDEEP = 1 and • SLEEPONEXIT = 1 (refer to Cortex®-M system control register.) • CPU NVIC interrupts and events cleared • ALL CPU EXTI wake-up sources are cleared • The RETDS_CD bit selects DStop or DStop2 • No CPU domain peripherals with kernel clocks or cadenced by LSI, LSE, HSI, or CSI
Mode exit	If WFI or return from ISR is used for entry: • EXTI interrupt enabled in NVIC • If WFE is used for entry and SEVONPEND = 0: • EXTI event • If WFE is used for entry and SEVONPEND = 1: • EXTI interrupt even when disabled in NVIC or EXTI event
Wake-up latency	EXTI and RCC wake-up synchronization

Peripherals located in the SmartRun domain are provided with the clocks thanks to the autonomous mode, even if the CPU is in CStop mode. When a peripheral has its autonomous bit activated, it receives its peripheral clocks according to the SmartRun domain state. When the CPU is in CStop mode and:

- If the SmartRun domain is in DRun mode, peripherals with autonomous mode activated receive their peripheral clocks.
- If the SmartRun domain is in DStop mode, no peripheral clocks are provided.

Autonomous mode does not prevent the SmartRun domain to enter in DStop or DStop2 mode. Autonomous mode allows the delivery of the peripheral clocks to peripherals located in the SRD domain, even if the CPU is in CStop mode. When a peripheral is enabled and has its autonomous bit activated (PERxAMEN bit in the RCC_SRDAMR register), this peripheral receives its clocks to the SRD domain state, if the CPU is in CStop mode.

In autonomous mode, the individual peripheral clocks can remain active by setting the corresponding PERxAMEN bit of the RCC_SRDAMR register:

- SRDSRAMAMEN bit, SRDSRAM bus clock is enabled when the SmartRun domain is in Run mode.
- BKPRAMAMEN bit, backup RAM clock enabling is controlled by the SmartRun domain state.
- DFSDM2AMEN bit, DFSDM2 peripheral clocks are enabled when the SmartRun domain is in Run mode. Kernel clock is enabled when the SmartRun domain is in Stop mode.
- DTSAMEN bit, digital temperature sensor DTS clocks are enabled when the SmartRun domain is in Run mode.
- RTCAMEN bit, RTC bus clocks are enabled when the SmartRun domain is in Run mode.

- VREFAMEN bit, VREF clocks are enabled when the SmartRun domain is in Run mode or Stop mode.
- COMP12AMEN bit, COMP1, and two peripheral clocks are enabled when the SmartRun domain is in Run mode.
- DAC2AMEN bit, DAC2 (containing one converter) peripheral clocks are enabled when the SmartRun domain is in Run mode.
- LPTIM3AMEN, LPTIM3 peripheral clocks are enabled when the SmartRun domain is in Run mode. Kernel clock is enabled when the SmartRun domain is in Stop mode.
- LPTIM2AMEN, LPTIM2 peripheral clocks are enabled when the SmartRun domain is in Run mode. Kernel clock is enabled when the SmartRun domain is in Stop mode.
- $I^2C4AMEN$ bit, I^2C4 peripheral clocks are enabled when the SmartRun domain is in Run mode. Kernel clock is enabled when the SmartRun domain is in Stop mode.
- SPI6AMEN bit, SPI6 peripheral clocks are enabled when the SmartRun domain is in Run mode. The kernel clock is enabled when the SmartRun domain is in Stop mode.
- LPUART1AMEN bit, LPUART1 peripheral clocks are enabled when the SmartRun domain is in Run mode. Kernel clock is enabled when the SmartRun domain is in Stop mode.
- GPIOAMEN bit, GPIO peripheral clocks are enabled when the SmartRun domain is in Run.
- BDMA2AMEN bit, BDMA2 and DMAMUX2 peripheral clocks are enabled when the SmartRun domain is in Run.

The SmartRun domain can be kept in DRun mode while the CPU is in CStop mode and the CPU domain is in DStop or DStop2 mode. This is done by setting RUN_SRD bit in PWR_CPUCR register. It is worth to note that the CPU can control:

- if the CPU domain is allowed entering DStop or DStop2 modes
- or the SmartRun domain is allowed entering in DStop when conditions are met, through bits RETDS_CD and PDDS_SRD of PWR control register (PWR_CPUCR)

10 HSLV

The high-speed low voltage option allows the output buffer optimization at low voltage with adjustable speed. When activated, HSLV mode enables a performance close to 3.3 V at 1.8 V at the same frequency. To maximize the performance, the I/O high-speed feature, HSLV, must be activated at low-device supply voltage.

For STM32H7A/7Bx microcontrollers, HSLV must be activated when V_{DD} is lower than 2.7 V. HSLV must not be enabled when V_{DD} is higher than 2.7 V, as this can damage the STM32 device.

In order to enable the HSLV mode, a specific bytes option must be enabled, and specific bits in the SYSCFG register must be set. These are listed below:

- Option bytes

  VDDIO_HSLV and VDDMMC_HSLV option bytes enable the configuration of pads below 2.7 V. This is respectively for V_{DDIO} and V_{DDMMC} power rails if set to one.

- SYSCFG register

  The speed of some I/Os can be increased at low voltage. This can be done by programming the corresponding HSLVx bits in the SYSCGF_CCCSR interfaces. This feature must be used only when the I/O power supply is below 2.7 V. The software writes the HSLVx bits to optimize the I/O speed when the product voltage is low.

  – HSLV3 indicates the high speed at low voltage for V_{DDMMC} I/Os. It is active only when VDDMMC_HSLV user option bit is set. It mainly controls the speed of SDIO on V_{DDMMC} power rail.

  – HSLV2, HSLV1, and HSLV0 indicate the high speed at low voltage for V_{DD} I/Os. They are active only when VDDIO_HSLV user option bit is set. They respectively control the speed of FMC, OCTOSPI, and SDMMC.

  – When testing a bus at 1.8 V (SDMMC data bus), special care must be taken before activating HSLV mode. All I/Os in the bus must support this mode. Otherwise, there is an unbalance in the timings.

11 Low-power application

This section provides information on smart power management of STM32H7A3/7B3 microcontrollers through an application example. It provides as well an example of SRD autonomous mode beneficial to power consumption.

11.1 Low-power application example

This example is based on I^2C transmission using the ST shield (X-NUCLEO-IKS01A2). The purpose of this example is to highlight the smart power management of STM32H7A3/7Bx microcontrollers when using two power domains.

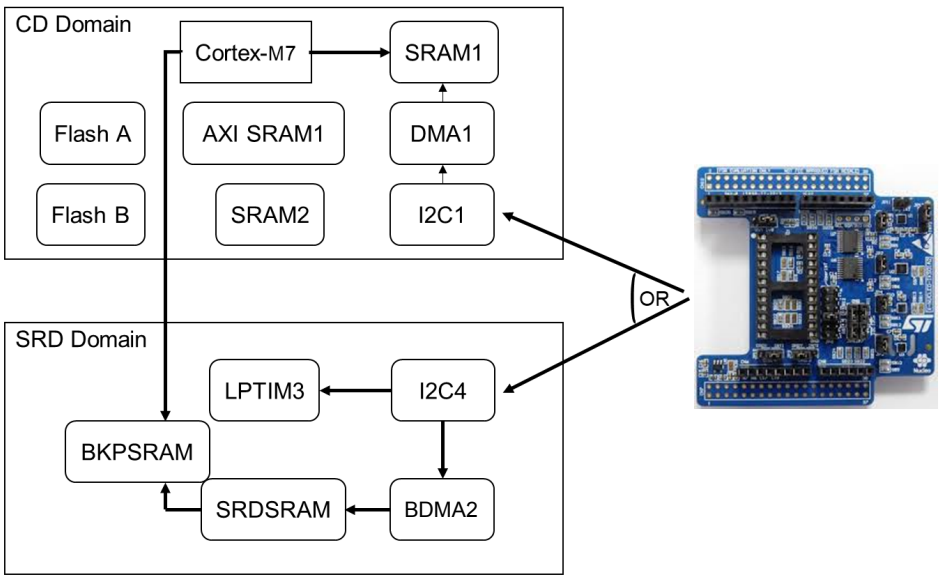
The demo code is based on six modes:

- Mode 1: this mode is equivalent to legacy products that keep all domains active to transfer data. This is performed from the ST shield through I^2C1 with the CPU in Run mode.
- Mode 2: this mode is equivalent to legacy products that keep all domains active to transfer data. This is performed from the ST shield through I^2C1 with the CPU in Sleep mode.
- Mode 3: in this mode, the power efficiency enhances keeping SRD in Run mode and CD in DStop. The data transfer is switched to I^2C4 instead of I^2C1. This highlights the architecture flexibility to move between peripheral instances. Namely, from CD to SRD without changing external connections.
- Mode 4: as mode 3 while the CD domain is in DStop2 for more power-energy saving. This mode shows the SRD autonomous Run mode.
- Mode 5: as mode 3 while the SRD domain switches between Run mode and Stop mode during transfer. This mode optimizes power consumption.
- Mode 6: as mode 4 while the SRD domain switches between Run mode and Stop mode during transfer. This mode optimizes power consumption

  The demo code uses the Nucleo board NUCLEO-H7A3ZI-Q and the X-NUCLEO-IKS01A2/A3 expansion board to get temperature values from the HTS221 sensor via the I^2C bus.

  The figure below shows the connections between peripherals and memories.

Figure 4. Peripherals and memories connections



Mode 1

1. Temperature acquisition from shield to SRAM1 through I²C1 in the CD domain.
2. Saving data from SRAM1 to BKPSRAM.

Table 6. Mode 1: operating mode and domains

CD domain	SRD domain
DRun(CRUN)	SRDRun

Mode 2

1. Temperature acquisition from shield to SRAM1 through I²C1 in the CD domain. The CPU is in Sleep mode.
2. Saving data from SRAM1 to BKPSRAM after waking up the Cortex®-M7 core.

Table 7. Mode 2: operating mode and domains

CD domain	SRD domain
DRun(CSleep)	SRDRun

Mode 3

1. Temperature acquisition from shield to SRDSRAM through I²C4 in the SRD domain in autonomous mode. The CD domain is in DStop mode.
2. Saving data from SRDSRAM to BKPSRAM after waking up the Cortex®-M7 core.

Table 8. Mode 3: operating mode and domains

CD domain	SRD domain
DStop(CStop)	SRDRun

Mode 4

1. Temperature acquisition from shield to SRDSRAM through I²C4 in the SRD domain in autonomous mode. The CD domain in DStop2 (retention) mode.
2. Saving data from SRDSRAM to BKPSRAM after waking up the Cortex®-M7 core.

Table 9. Mode 4: operating mode and domains

CD domain	SRD domain
DStop2(CStop)	SRDRun

Mode 5

1. Temperature acquisition from shield to SRDSRAM through I²C4 in SRD domain. In autonomous mode (SRDRun) with the CD domain in DStop mode.
2. System state in Stop mode (CD domain in DStop mode and SRD domain in SRDStop mode). Toggling of five times between state 1 and state 2.
3. Saving data from SRDSRAM to BKPSRAM after waking up the Cortex®-M7 core.

Table 10. Mode 5: operating mode and domains

CD domain	SRD domain
DStop(CStop)	SRDRun/SRDStop

Mode 6

1. Temperature acquisition from shield to SRDSRAM through I²C4 in SRD domain. In autonomous mode (SRDRun) with the CD domain in DStop2 mode (retention).
2. System state in Stop mode (CD domain in DStop2 mode and SRD domains in SRDStop mode). Toggling of five times between state 1 and state 2.
3. Saving data from SRDSRAM to BKPSRAM after waking up the Cortex®-M7 core.

Table 11. Mode 6: operating mode and domains

CD domain	SRD domain
DStop2(CStop)	SRDRun/SRDStop

11.2 Mode 6 - SRD autonomous mode

This section details the SRD autonomous mode with mode 6 example in Section 11.1 , and its power consumption benefits.

11.2.1 Memory retention

The SRD domain features a 32 kbytes of SRAM (SRDSRAM) to retain data while the CD is in DStop2 mode. This feature can be used in several use-cases:

- To retain the application code in order to recover properly from DStop2 mode.
- To retain the data from or to a sensor when the CPU enters CStop. CD domain is in DStop2 between two consecutive operations.

In Figure 5, both SRDSRAM, and BKPSRAM are used, respectively to save the transmitted data from the temperature sensor when the system is in autonomous mode. The final data is saved after waking up the CPU from CStop mode.

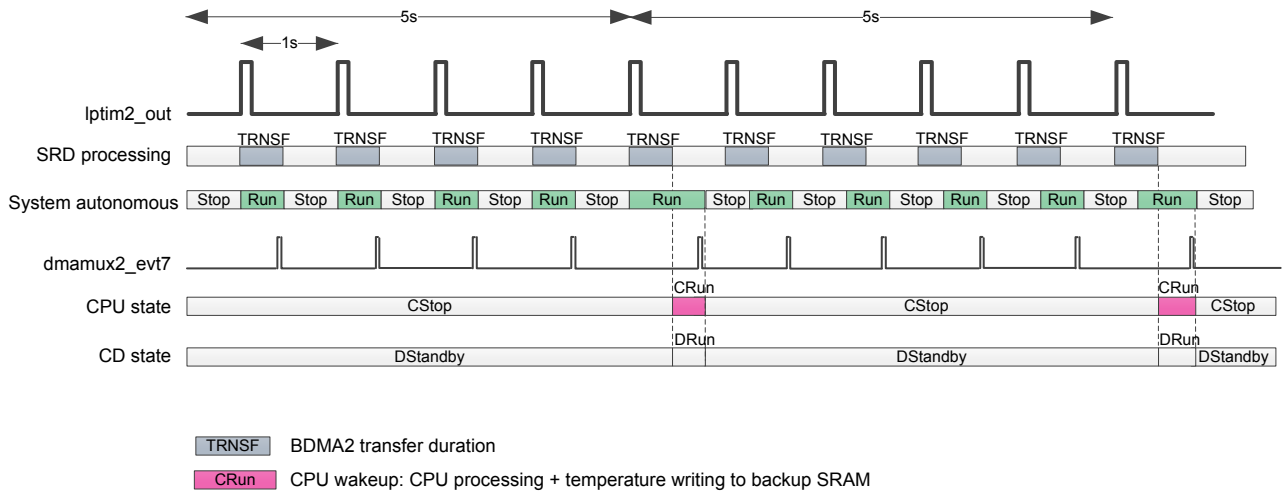
11.2.2 Example description

Figure 5 shows the proposed implementation of SRDSRAM to I²C4 using BDMA2. The timing diagram describes the different steps. The LPTIM3 gives intervals each second. The SRD domain wakes up from Stop mode and the SRD domain executes the following tasks in autonomous mode:

1. Transfer data from SRDSRAM to I²C 4 using BDMA2.
2. When the I²C4 interface indicates that the last 2 bytes have been transferred from I²C4 to SRDSRAM, the SRD domain switches to Stop mode.

After the transfer of the first or last 10 bytes, BDMA2_channel7 (BDMA2_ch7) sends an interrupt to wake up the CPU from CStop mode. When the CPU is in CRun mode, it prepares data located into the SRDSRAM for process and transfer to the backup SRAM.

Figure 5. SRDSRAM timing diagram to I²C 4 transfer with BDMA2



Note: To toggle the SRD domain between Stop mode and Run mode, the wake-up event (lptim3_out each second) must trigger Run mode. The SRD domain can clear this event with the dmamux2_evt7 signal and switch back the SRD domain to Stop mode.

RCC programming

In this example, the CPU subsystem also includes the peripherals of the SRD domain that are used for data transfer. Namely, LPTIM3, BDMA2, DMAMUX2, SRDSRAM, I2C4, backup SRAM.

These peripherals must be programmed in autonomous mode to operate even when the CPU is in CStop mode.

When STM32Cube_FW_H7 is used, the following functions defined in the `stm32h7xx_hal_rcc.h` file permit the programming of peripherals in autonomous mode:

```
__HAL_RCC_LPTIM3_CLKAM_ENABLE ();
__HAL_RCC_BDMA_CLKAM_ENABLE ();
__HAL_RCC_SRDSRAM_CLKAM_ENABLE
__HAL_RCC_I2C4_CLKAM_ENABLE ();
__HAL_RCC_BKPRAM_CLKAM_ENABLE ();
```

PWR programming

In this mode, the PWR block must be programmed:

- to prevent the system SRD domain enters in Standby mode, when the data transfer completes
- to allow CD domain to enter DStop2 mode
- to define the working voltage according to the system modes (SVOS3).

EXTI programming

The lptim3_out signal is used to wake up the SRD domain from Stop mode:

- When LPTIM3 time intervals have elapsed at each second, the lptim3_out signal is connected to the EXTI input event, line number 51.
- When selecting BDMA2_ch7 as pendclear source of SRD domain, the SRD domain switches back to Stop mode.
- When using STM32Cube_FW_H7 function:
  `__HAL_EXTI_SRD_EventInputConfig(EXTI_LINE51,ENABLE,BDMA_CH7_CLEAR);` defined in `stm32h7xx_hal.c` it selects lptim3_out signal as wake-up source and dmamux2_evt7 as pendclear source:

BDMA2 and DMAMUX2 programming

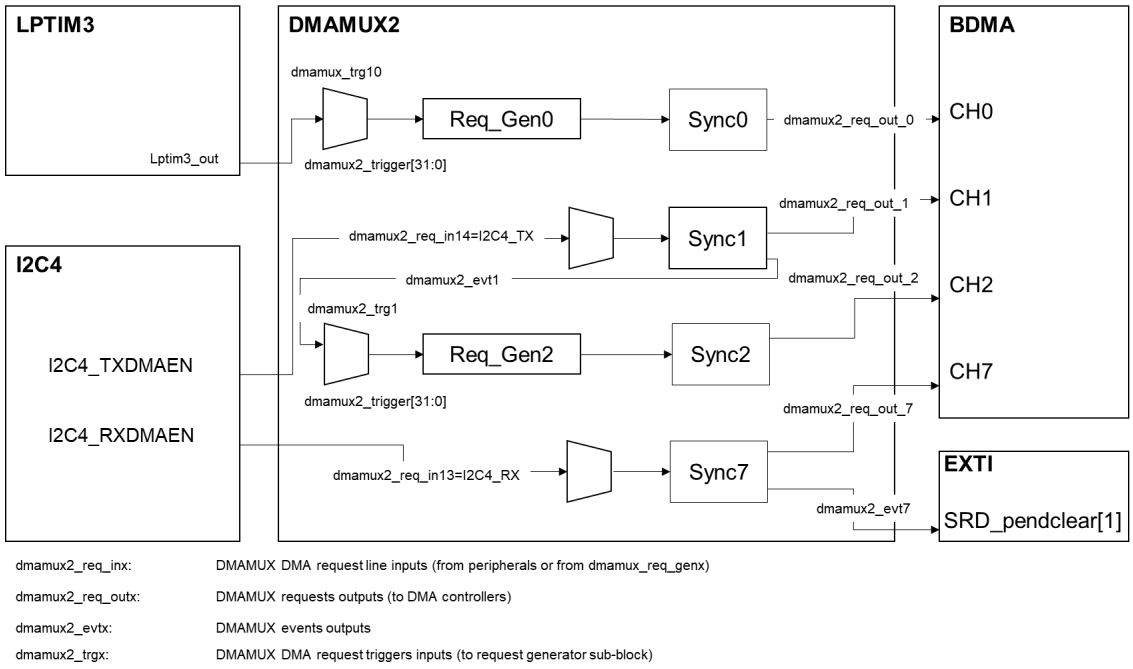
To execute the data transfers between BKPSRAM and I²C4 they must be four BDMA channels.

- BDMA_channel0 (BDMA_ch0) transfers data from BKPSRAM to the I2C4_CR2 register, to configure the I²C4 interface to request write transfer to HTS221sensor.
- BDMA_ch0 uses Req_Gen0 to generate BDMA requests. The rising edge of the lptim3_out signal. triggers the generation of BDMA requests. LPTIM3 generates the signal each second.
- BDMA_channel1 (BDMA_ch1) transfers data from BKPSRAM to I2C4_TXDR register to send a specified address to the HTS221 sensor. BDMA_ch1 request is generated when I2C4 TX-FIFO enables TXDMAEN bit in I2C4_CR1 register. The DMAMUX SYNC1 block is programmed to generate a pulse on its dmamux2_evt1 output when the BDMA request is complete.
- BDMA_channel2 (BDMA_ch2) transfers data from BKPSRAM to I2C4_CR2 register, to configure the I²C interface. This allows to request read transfer from the HTS221 sensor.
- BDMA_ch2 uses Req_Gen2 to generate BDMA requests. The falling edge of dmamux2_evt1 triggers the generation of BDMA requests.
- BDMA_channel7 (BDMA_ch7) transfers the received data from I2C4_RXDR register. Data from the HTS221 sensor to BKPSRAM. BDMA_ch7 transmission is enabled by setting RXDMAEN bit. BDMA_ch7 is configured to generate interrupts to wake up the CPU from CStop mode. The interrupts are generated each time after a half-transfer completes. The transfer completes at 10 bytes of data or after full transfer at 20 bytes of data.

The DMAMUX2 SYNC7 block is programmed to generate a pulse on its dmamux2_evt7 output when the BDMA2 request is complete. The dmamux2_evt7 signal is used by the EXTI to switch back the SRD domain to Stop mode.

The figure below shows the active signal paths via DMAMUX2.

Figure 6. BDMA2 and DMAMUX2 interconnection



LPTIM3 programming

LPTIM3 operates when the SRD domain enters Stop mode and uses ck_lsi clock. LPTIM3 wakes up the SRD domain from Stop mode each second and to start `BDMA2_ch0 transfer through Req_Gen0`.

I²C programming

I²C4 is configured to use `ck_hsi` as the kernel clock to generate BDMA2 request when TX-FIFO/RX-FIFO is empty or full. A dedicated function called`I2C4_ENABLE_DMA_REQUEST()` is defined in the common.h file to enable BDMA2 requests generation by the I²C4.

11.2.3 Mode 6 example - general description

The steps below details Figure 5:

Step 1. The lptim3_out signal wakes up the SRD domain from Stop mode each one second.

Step 2. When the SRD domain wakes up from Stop mode, the rising edge of lptim3_out signal triggers BDMA2_ch0 to begin data transfer (TRNSF1).

Step 3. As soon as I2C4 TX-FIFO is empty, BDMA_ch1 begins the data transfer from BKPSRAM to I2C4_TXDR register (TRNSF2).

Step 4. The end of this transfer triggers BDMA2_ch2 transfer (through dmamux2_ch1_evt) from BKPSRAM to I2C_CR2 register (TRNSF3).

Step 5. As soon as I2C4 RX-FIFO is full, BDMA2_ch7 transfers 2 bytes of data from I2C4_RXDR register to BKPSRAM (TRNSF4).

Step 6. The end of this transfer triggers a dmamux2_evt7 signal which is used to clear the SRD_PendClear bit.

Step 7. The whole system returns to Stop mode.

Step 8. After five seconds and when the expected amount of data has been transmitted (NDT bits of BDMA_CNDTR0 set to 10 indicating half-transfer is complete), the BDMA_ch7 generates an interrupt to wake up the CPU from CStop mode.

Step 9. The CPU processes the data located in BKPSRAM for transfer, copy it to BackupSRAM (CPU_TRSF). The data stored in BackupSRAM is retained while the system is in Stop mode.

Step 10. The CPU returns to CStop mode and the whole system return to Stop mode. LPTIM3 continues to wake up the SRD domain after each one second, and BDMA2 channels continue to do all transfers as mentioned above. When the expected amount of data has been transmitted (NDT bits of BDMA_CNDTR0 set to 20 indicating that the transfer is complete), the BDMA2_ch7 generates an interrupt to wake up the CPU from CStop mode. The CPU processes the data located in SRDSRAM for transfer and copies it to BackupSRAM. (CPU_TRSF).

11.3 How to use the application

The hardware requirements and set-up of the application are listed below.

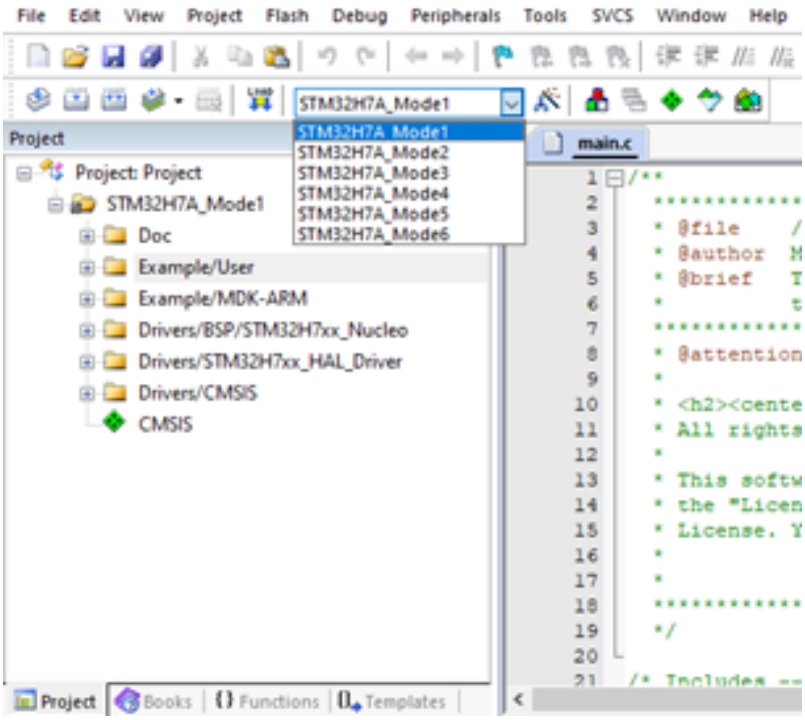
- NUCLEO-H7A3ZI-Q board
- X-NUCLEO-IKS01A2 expansion board to measure temperature values from HTS221 sensor via I2C bus
- The X-NUCLEO-IKS01A2 expansion board must be plugged on the matching pins of the STM32
- Nucleo boards connector:
  - Connect the board to a computer through ST-LINK with a mini USB for programming and debugging
- A PowerShield to measure the current consumption

11.3.1 Application example

Follow these steps to perform correctly the example.

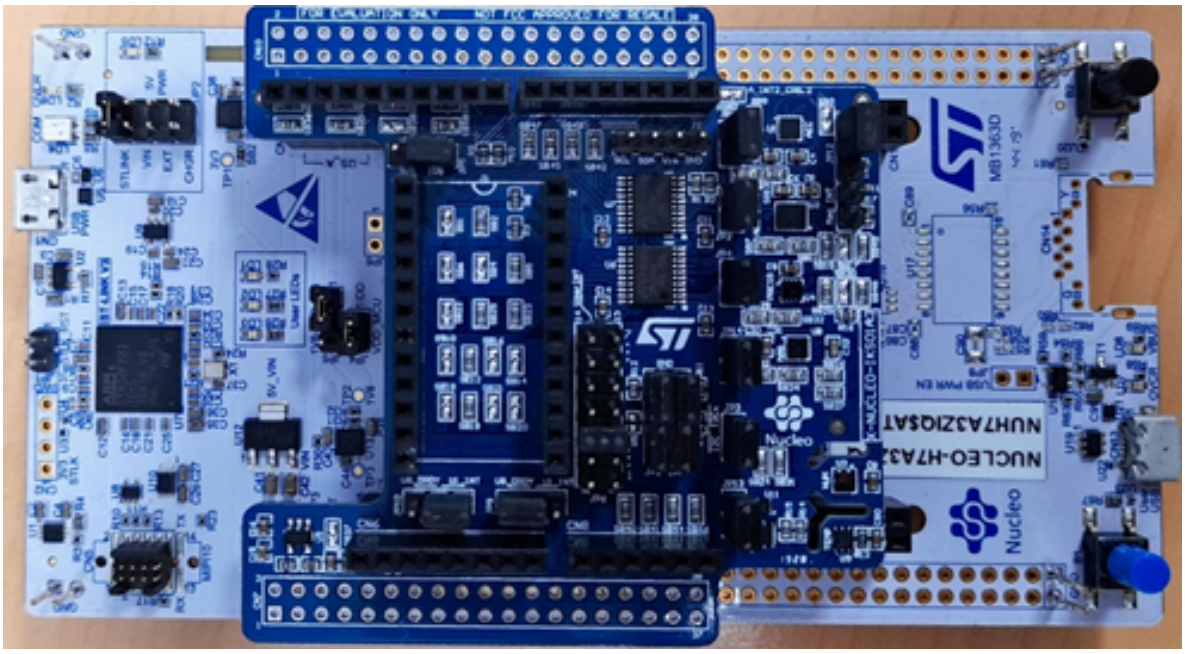
1. Connect the NUCLEO-H7A3ZI-Q board to a computer through ST-LINK with a mini-USB for programming and debugging.
2. X-CUBE-PWRMGT-H7.
3. Compile the project with the selected mode, and download it to the MCU.

Figure 7. How to select the different modes



4. Plug in the shield X-NUCLEO-IKS01A2/A3 to the matching pins of the STM32H7A3ZI-Q board.

Figure 8. X-NUCLEO-IKS01A2/A3 connection



5. Remove the jumper 4 (JP4) and connect instead an ampere meter to measure the current consumption. It is also possible to use the PowerShield X-NUCLEO-LPM01A (see Figure 9).

   If the PowerShield is used:

   a. Remove the IDD jumper JP4 in the NUCLEO-H7A3ZI-Q board.

   b. Connect the PowerShield on the X-NUCLEO-IKS01A2/A3.

   c. Reset the PowerShield X-NUCLEO-LPM01A.

      For more details, follow the instructions on how to use the Cube monitor. See *STM32CubeMonitor-Power software tool for power and ultra-low-power measurements* (UM2202). It is also possible to use only the LCD screen on the board.

Figure 9. PowerShield connection

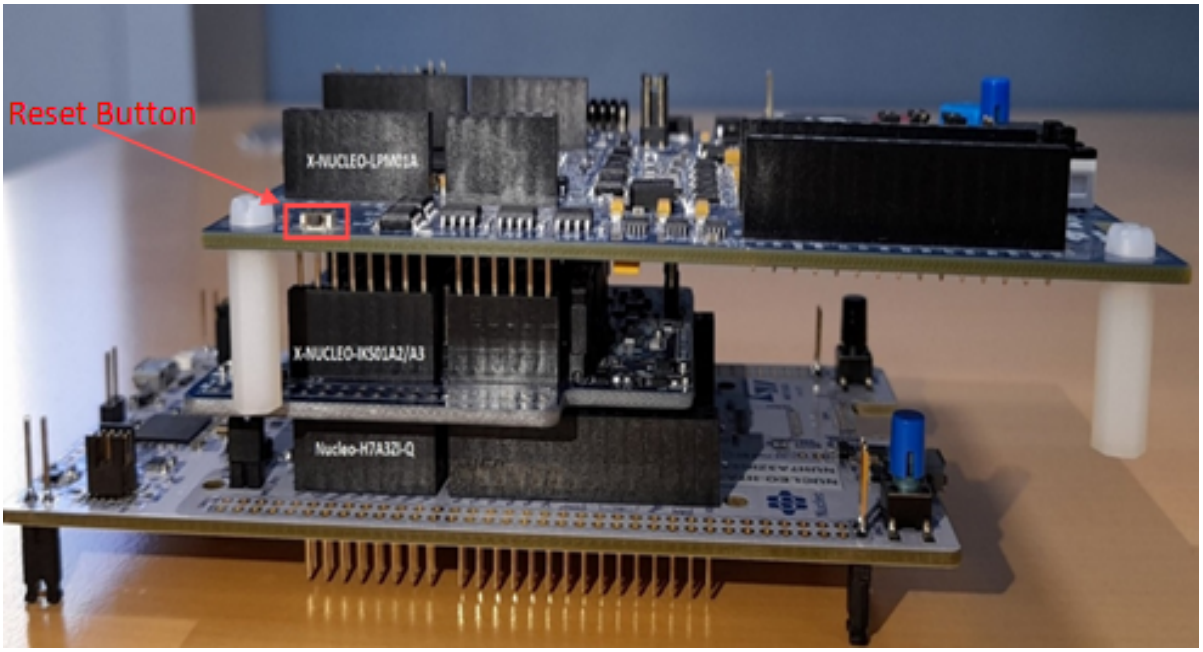
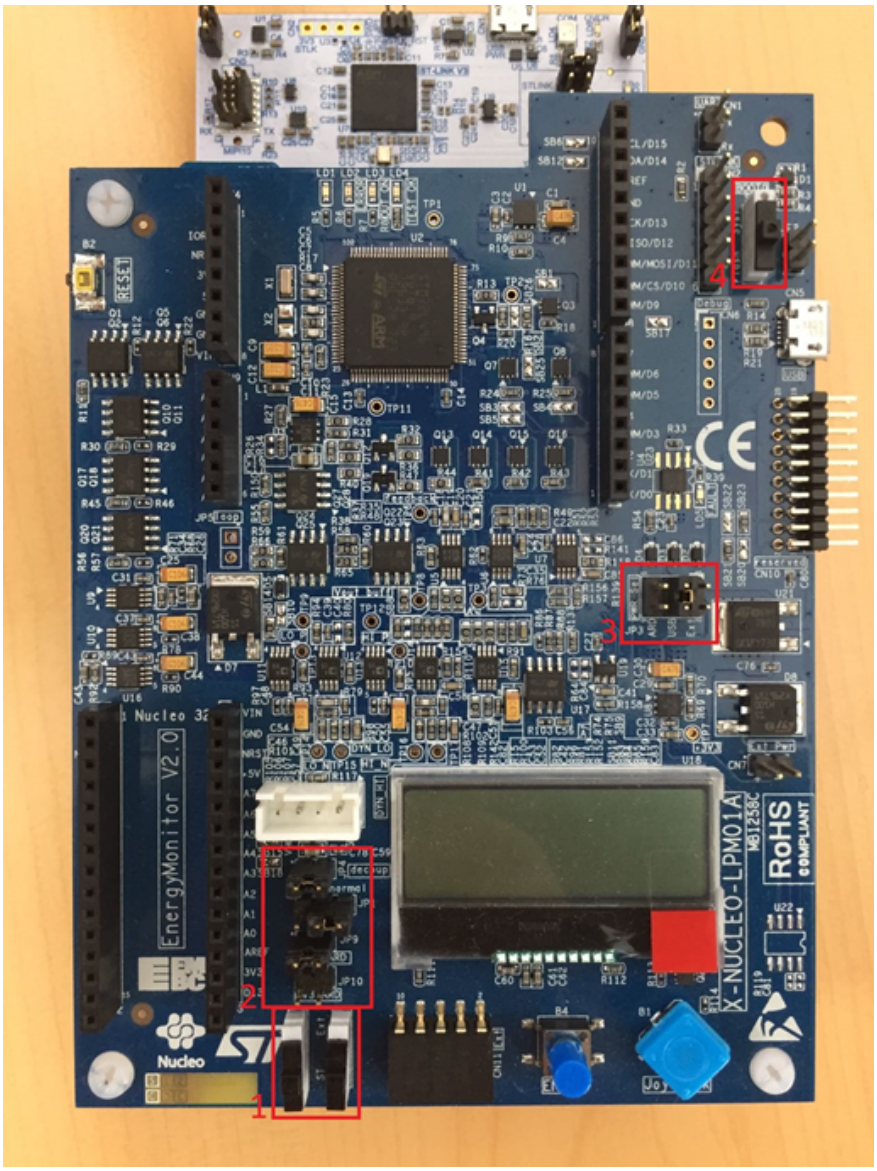


Figure 10 illustrates the PowerShield connection as follows:

– S1 and S2 must be in ST position
– Jumper position:
  ◦ JP4 = on
  ◦ JP1: for measurements, the jumper must be always in the normal position
  ◦ JP9 = on
  ◦ JP10 = off
– JP3: the jumper must be inserted in the USBposition
– S3 in flash memory position

Figure 10. **PowerShield top view**



To setup Tera Term, see Figure 11:

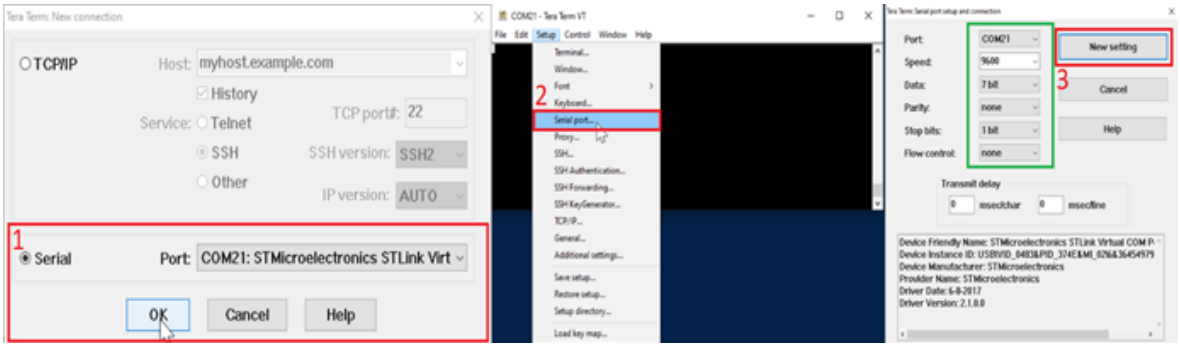
6. Open the terminal to see the output messages (Tera Term for example) with the setup
   a. Open the terminal and select the serial protocol.
   b. Select **Setup** then **Serial port.**
      i. The last step is to set up the communication port configuration. This is illustrated with the green frame.
   c. Follow the steps as illustrated with the red frames for the correct functioning of the terminal.

7. Press the reset button of the NUCLEO-H7A3ZI-Q board, the program starts the execution.

Note: Select and recompile the project for the other modes.

Note: Connect the PowerShield X-NUCLEO-LPM01A to a computer through the mini-USB cable to power it up.

Figure 11. Tera Term setup



11.3.2 Results comparison

To compare the obtained results, refer to [R3]. When the system is in Stop mode, all the peripherals bus interfaces clocks are off. However, peripherals in the SmartRun domain having a kernel clock request can stay active by programming their PERxAMEN bit (RCC_SRDAMEN).

In the provided example, it is the case for the LPTIM3. Thus, additionally to the power consumption of each mode, the LPTIM3 power consumption must be considered. Refer to LPTIM3 kernel in table Peripheral current consumption in Run mode from [R3].

Revision history

Table 12. **Document revision history**

Date	Version	Changes
27-May-2022	1	Initial release.

Contents

List of tables

List of figures