Guidelines for power management on STM32H5 MCUs

Introduction

All STM32H5 series feature a variety of power-saving mechanisms that can help to reduce significantly power consumption. Thus, it enables an optimal balance between performance processing and power efficiency.

This document dealing with the STM32H5 series describes the system architecture and configurations. It also provides guidelines related to the different power modes and the reduction of power consumption. The document outlines different ways of:

- Decreasing power consumption
- Optimizing battery life

The smart power management includes the cache configuration, the DMA access, and many other features.

1 General information

This document applies to the STM32H5 Arm®-based microcontrollers.

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

Table 1. Glossary

Acronym	Description
LDO	Linear voltage regulator
SRAM	Static RAM
SMPS	Switched mode power supply
ESR	Equivalent series resistance
SVOS	System Stop mode voltage scaling selection
RTC	Real-time clock
BOR	Brownout reset
IWDG	Independent watchdog

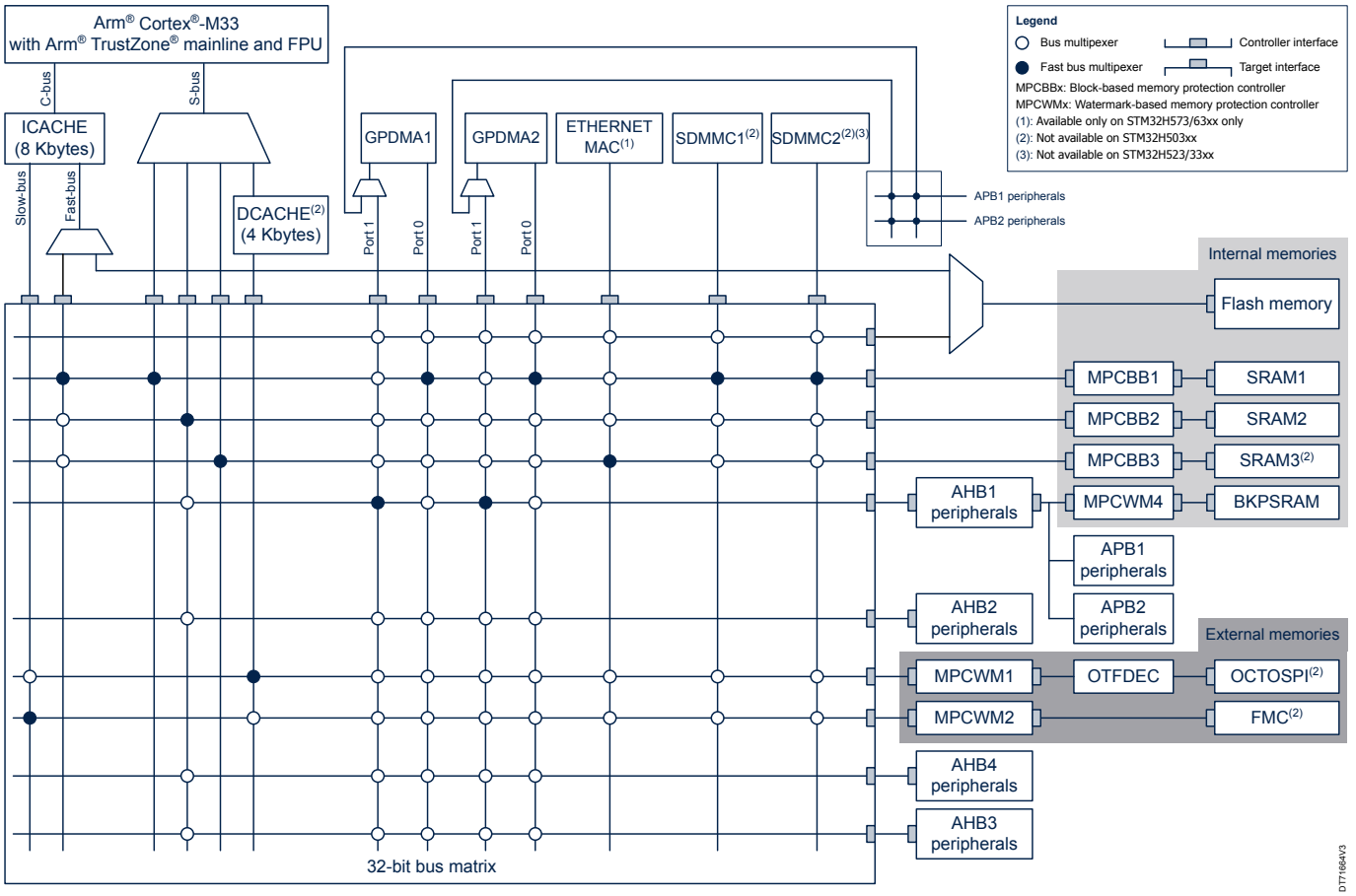
2 System architecture

The architecture relies on an Arm® Cortex®-M33 core optimized through an instruction cache. That instruction cache has a direct access to the embedded flash memory.

This architecture also features a 32-bit multilayer AHB bus matrix that interconnects up to 13 masters, and up to 10 slaves.

The bus matrix provides access from a master to a slave. It enables concurrent access and efficient operation even when several high-speed peripherals work simultaneously. The architecture is shown in Figure 1.

Figure 1. STM32H5 series system architecture



2.1 ICACHE

The Cortex®-M33 processor features an instruction cache on its C-AHB code bus to enhance performance when retrieving instructions and data from both internal and external memory sources. The ICACHE improves the fetching process, and thus enhances the overall performance of the processor.

2.2 DCACHE

The Cortex®-M33 processor also includes a data cache on its S-AHB system bus. The data cache improves the performance for data transfer to and from external memory. The DCACHE employs advanced features such as hit-under-miss and critical-word-first refill policy. It is in order to optimize data access performance when using external memory sources. The result is a faster and more efficient data transfer.

2.3 DCACHE and ICACHE power consumption impact

The inclusion of both an ICACHE and DCACHE in the Cortex®-M33 processor leads to a reduction in the number of memory accesses. This reduction results in a decrease of the power consumption. This is achieved by storing frequently accessed instructions and data in the cache. It enables faster retrieval and reduces the need to access the main memory. This process improves performance and helps to conserve power.

ICACHE and DCACHE contents are preserved in Sleep mode and Stop mode, and they are lost in Standby mode. The SRAM content is preserved in Sleep mode, and optionally preserved in Stop mode. Backup SRAM can be preserved in all low-power modes.

Note: The DCACHE is only available on some part numbers. Refer to the corresponding datasheet for more details.

3 System supply configuration

The V_{CORE} domain is supplied either by the internal linear voltage regulator LDO, or by the embedded SMPS step-down converter.

The devices can embed two types of regulators. LDO or SMPS. The objective is to provide the V_{CORE} supply for digital peripherals, SRAMs (except BKPSRAM), and embedded flash memory. The SMPS generates this voltage on V_{CAP} (two pins), with a total external capacitor of 10 µF (typical) and requires an external coil of 2.2 µH (typical).

The STM32H5 series support dynamic voltage scaling to optimize its power consumption in run mode. The voltage from the main regulator that supplies the logic (V_{CORE}) can be adjusted according to the maximum operating frequency of the system. Table 2 highlights the different voltage scalings with their supported frequency.

Table 2. Voltage scaling

Voltage scaling range	V_{CORE}	Maximum frequency
VOS0	1.35 V	250 MHz
VOS1	1.2 V	200 MHz
VOS2	1.1 V	150 MHz
VOS3	1.0 V	100 MHz

The LDO generates this voltage on V_{CAP} with a total of an external capacitor of 4.7 µF.

Both regulators can provide four different voltages (voltage scaling) and can operate in Stop modes. Depending on the package configuration (SMPS or LDO), the hardware selects the regulator. SMPS and LDO regulators are exclusively selected.

The embedded SMPS (switched-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).

The SMPS step-down convertor is always enabled after reset when its power supply is provided on the VDDSMPS pin.

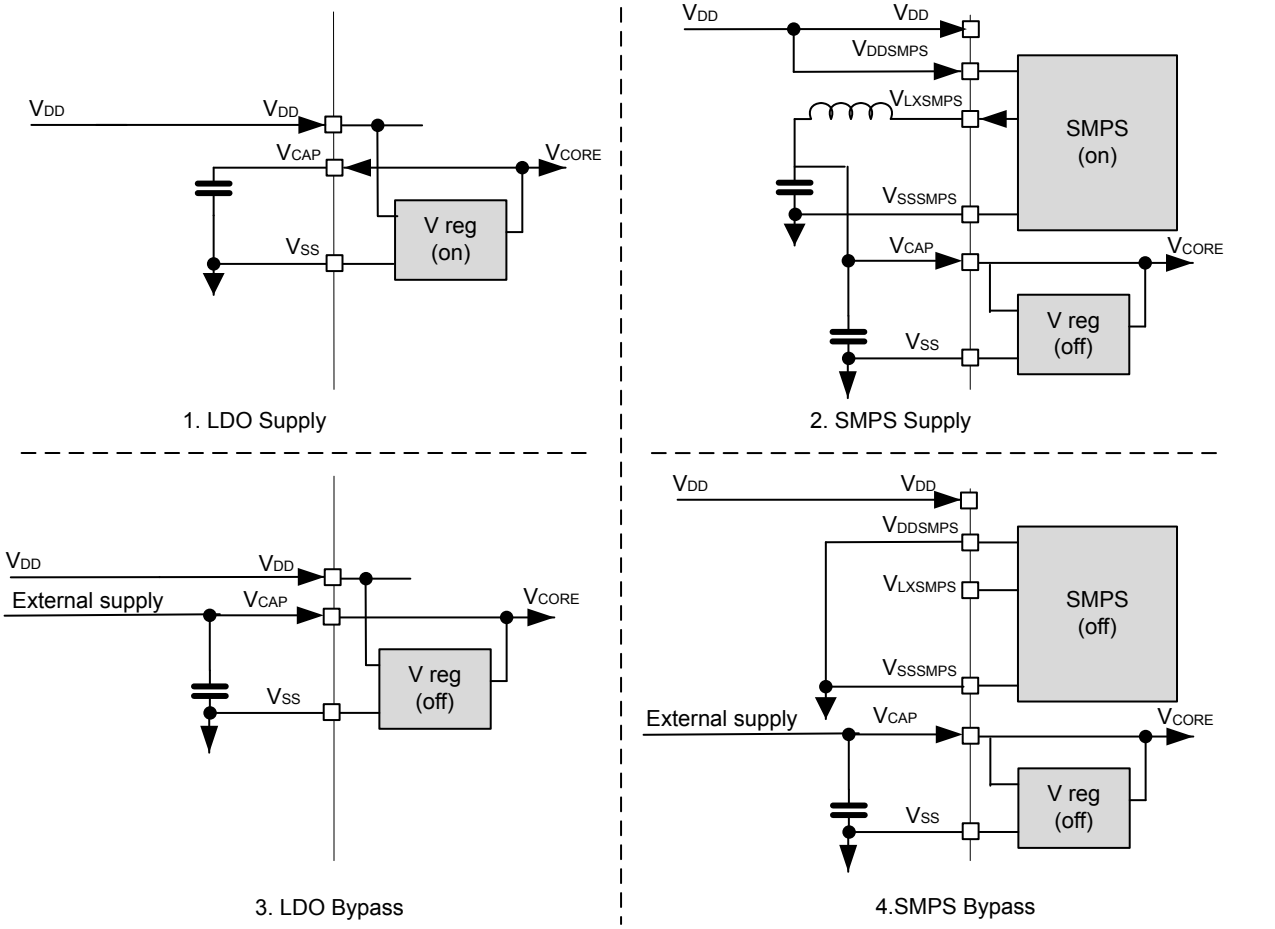
Figure 2 below gives an overview of all possible use cases of the regulator (V_{CORE} domain supply):

- LDO power supply
- SMPS power supply
- Bypass mode

Note: Both LDO and SMPS regulators are enabled by the hardware depending on the package configuration

The SMPS power supply pins are available only on a specific package with an SMPS step-down converter option. Refer to the corresponding STM32H5 datasheet for more details.

Figure 2. System supply configurations for packages



Note: The SMPS regulator is available on specific packages. Refer to the corresponding datasheet for more details.

3.1 Bypass mode and LDO constraints

When V_{CORE} is in bypass mode, it needs to settle at the default level of at least 1.1 V before V_{DD} reaches the POR threshold level. The LDO is enabled by default after power-up, so the external V_{CORE} voltage needs to stay above 1.1 V until the software disables the LDO.

Refer to Figure 2 for more information.

If V_{CORE} is supplied from an external source, the voltage level needs to be reflected in the VOS[1:0] bits in the PWR_VOSCR register, which affects performance.

Bypass mode needs to be used with caution. The software needs to select intermediate levels sequentially when increasing or decreasing performance. For example, when increasing from VOS3 to VOS0, VOS2 and VOS1 need to be passed through first. Then, the external voltage and system frequency can be changed accordingly.

4 Low-power modes

To preserve the power during periods of idle CPU activity, such as waiting for an external event, a variety of low-power modes are available.

For each application, the end-user must evaluate and select the mode that optimally balances energy efficiency, quick startup times, and the availability of wake-up triggers. Selecting the right low-power mode can help to reduce power consumption and extend the battery life.

There are several ways to reduce power consumption on STM32H5 microcontrollers:

- Decreasing dynamic power consumption by slowing down the system clocks. This is done even in run mode and by individually clock-gating the unused peripherals.
- To optimize power consumption and prolong battery life, the CPU uses various low-power modes when idle. These low-power modes are carefully selected according to the specific needs of the end-user application. This approach strikes a balance between quick startup times, energy efficiency, and the availability of wake-up triggers.

4.1 Static mode overview

The low-power modes belonging to the static mode for STM32H5 microcontrollers are:

- Sleep mode
  - CPU clock is stopped.
  - All peripherals can wake up the CPU (Interruption/event).
- Stop mode
  - The voltage regulator supplies the V_{CORE} domain to retain the content of registers and internal memories.
  - It enables optimizing the power consumption. The unused RAMs can be totally, or partially, shut-off.
  - Internal oscillator HSI64 or CSI can be kept active during Stop modes in order to reduce the wake-up time.
  - All clocks in a V_{CORE} domain are stopped. Only the LSE or LSI clock is still running.
  - The RTC can remain active.
  - System clock, when exiting from the Stop mode, can be either HSI up to 64 MHz, or CSI. It depends on the software configuration.
- Standby mode
  - It achieves the lowest power consumption with the BOR.
  - The internal regulator is switched off and the core domain is powered off.
  - All the oscillators are switched off.
  - RTC can remain active (with or without RTC).
  - The brownout reset (BOR) always remains active in Standby mode.
  - The state of the I/Os during Standby mode can be:
    1. I/O retention is disabled and I/Os are in High-Z state.
    2. I/O retention enabled and the state is preserved.
  - The SRAMs and register contents are lost, except for registers and backup SRAM that are in the backup domain and standby circuitry.
- V_{BAT} mode
  - The hardware sets it automatically when V_{DD} is OFF and V_{BAT} mode is supplied.
  - The V_{BAT} pin facilitates an external power supply for the device V_{BAT} domain through the use of an external battery or a super-capacitor. Thus, it provides a reliable and alternative power source.
  - The V_{BAT} pin supplies power and provides the RTC with a low-speed external clock source (LSE), antitampering detection (TAMP), backup registers, and 2-Kbyte backup SRAM. Additionally, eight dedicated antitampering detection pins are available when the device is operating in V_{BAT} mode. They further enhance the security measures.
  - The V_{BAT} function is automatically activated in the event of a loss of V_{DD} power, ensuring seamless operation. Furthermore, an integrated V_{BAT} battery charging circuit is present and can be activated when V_{DD} power is available, allowing for efficient charging and extended battery life.

Table 3. Low-power mode summary

Mode name	Wake-up source	Wake-up system clock	Effect on clock	Voltage regulators
Sleep (sleep-now or sleep-on-exit)	Any interrupt Wake-up event	No impact on the clock	CPU clock is off. There is no effect on other clocks or the analog clock source.	VOS3, VOS2, VOS1, or VOS0
Stop	• Any EXTI line that is configured in the EXTI registers. • Specific peripherals events(1)	• CSI when STOPWUCK = 1 in RCC_CFGR • HSI with the frequency before entering Stop mode, up to 64 MHz, when STOPWUCK = 0	• All clocks are off except LSI and LSE. • HSI or CSI can be enabled temporarily when requested by the software.	SVOS3, SVOS4, or SVOS5
Standby	Wake-up pin edge, RTC event, IWDG reset, external reset in nRST pin	HSI clock at 32 MHz	All clocks are off except LSI and LSE.	Off

1. *Peripherals are able to wake up the system from Stop mode. This is only possible when SVOS3 is selected before entering the Stop mode.*

Note: Peripherals are able to wake up the system from Stop mode. This is only possible when SVOS3 is selected before entering the Stop mode.

Table 4. Stop mode voltage scaling

Voltage scaling range	V_{CORE}
SVOS3	1 V
SVOS4	0.9 V
SVOS5	0.74 V

4.2 Dynamic mode overview

The voltage supplied to the V_{CORE} domain is the highest level. It provides optimal power to the system. Adjusting the regulator output voltage through the software enables matching the specific voltage requirements. The voltage scaling (VOS) feature enables a fine-tuning of the power consumption, particularly when the system is operating at frequencies below its maximum capabilities. This enables efficient power management and optimized performance.

The use of HSI and CSI as source clocks has distinct differences, with HSI consuming more energy than CSI. When using low-power modes, making CSI as the source clock is a more reliable and efficient option.

5 Power consumption and performance enhancement

5.1 Tips for saving power consumption

All STM32 series feature a variety of power-saving mechanisms. These help to reduce significantly current consumption, enabling an optimal balance to be struck between performance processing and power efficiency.

Cache configuration

The Cortex®-M33 processor features respectively instruction and data caches on the C-AHB code bus and S-AHB system bus. Thus, it enhances the performance when retrieving instructions and data from internal and external memories. Moreover, it also reduces power consumption.

System clock configuration and management

The system overclocking should be avoided by slowing the system clock when the maximum rate is not needed.

Voltage regulator power tricks

- Dynamic voltage and frequency scaling
  The voltage supplied to the processor can be lowered when the power frequency is sufficient. With such power management, the power drawn from the battery is reduced. The reduction is done by monitoring the processor input voltage to meet the system performance requirements. That consists in scaling the STM32H5 regulator output voltage that supplies the V_{CORE} domain (core, memories and digital peripherals) when the clock frequency is lowered according to the processing needs.
  The STM32H5 series offers four voltage scales. Refer to Table 2.
- I/Os configuration
  To minimize unnecessary, I/O current, unused pins should be configured as analog inputs.
- Use of the DMA access
  It is useful to improve the performance. It enables also the device to consume less average current over the life of the application.
- Code optimization:
  – Uses compiler optimization.
  – Simplifies the program flow for common cases.

Low-power mode

- Disable the clock of the unused peripherals. This helps us to reduce the power consumption.
- Shut-off all the unused memories partial retention of SRAM2 content. It enables saving data content and reduces power consumption
- Put the flash memory in power down. Before entering the Stop mode, configure the flash memory in low-power mode by setting the FLPS bit of PWR_PMCR. This results in an improved power consumption.
- SVOS (system Stop mode voltage scaling) Stop mode power consumption can be reduced using SVOS4, and even further with SVOS5

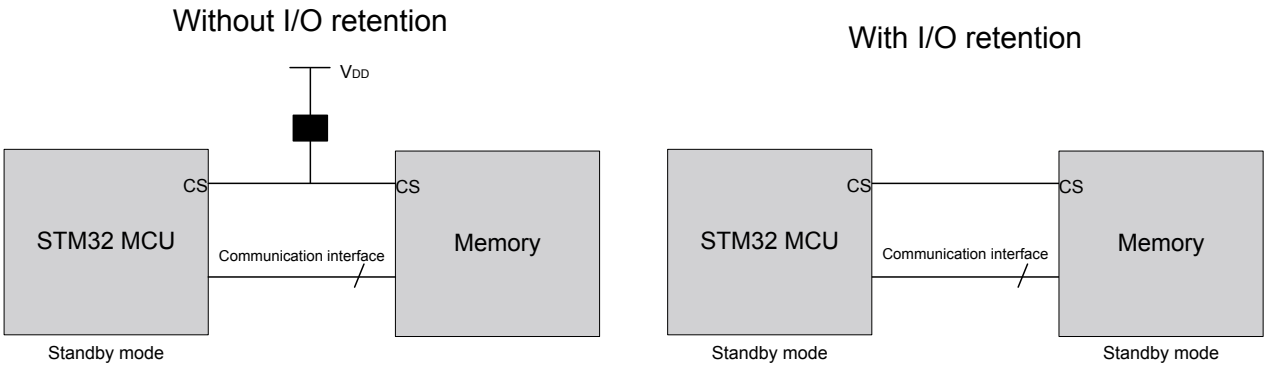
Note: When the system enters the Stop mode with SVOS5 enabled, the flash memory is automatically forced into low-power mode.

I/O retention

In Standby mode, the internal regulator is turned off. In this mode, almost all power consumptions related to peripherals and internal memories are stopped. The application might also need to control external devices interfaced via the GPIOs. The goal is to avoid communication protocol violation and extra consumption due to floating control signals.

if the application needs to maintain the GPIO level for the control of an external device, the I/O retention feature must be enabled.

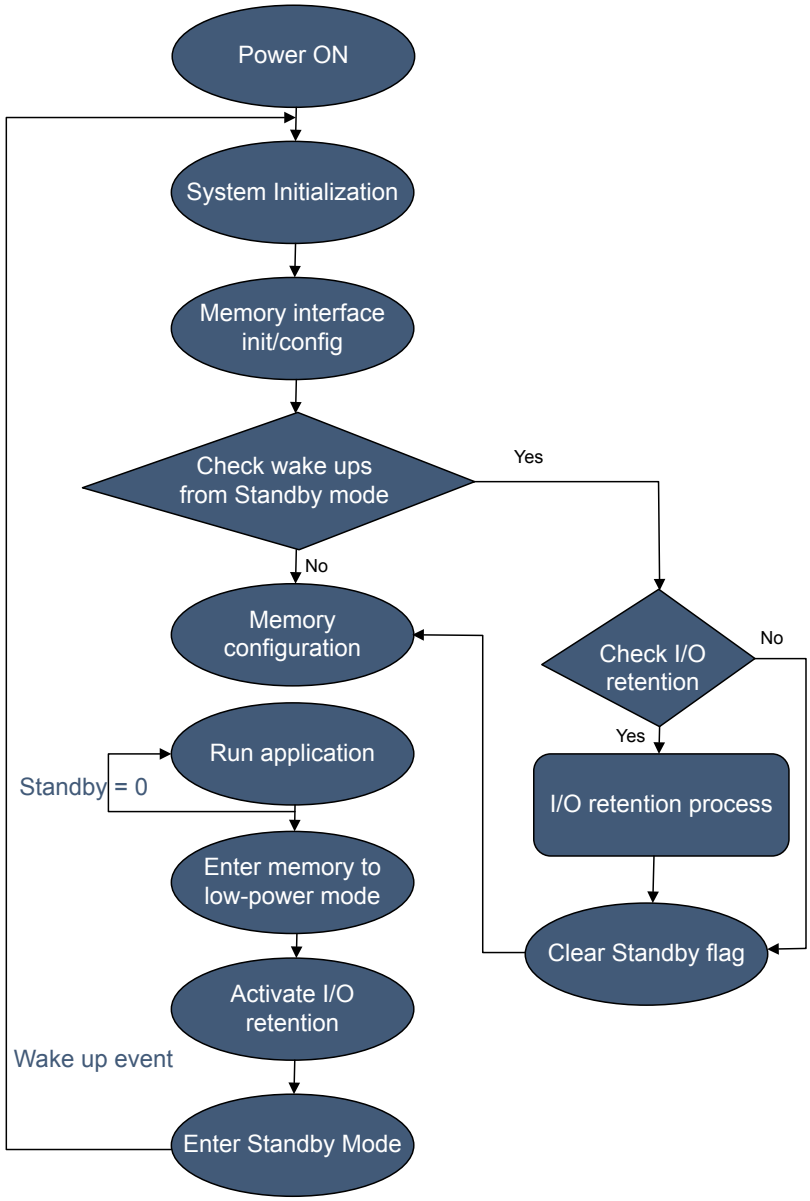
Figure 3. I/O retention



DT71656V1

- Without I/O retention:
  Memory interface is no more controlled and might use more power. In this case, an external component is needed to fix the interface level.
- With I/O retention:
  See below a flow chart that explains how the I/O retention works.

Figure 4. Wake up from Standby mode with I/O retention

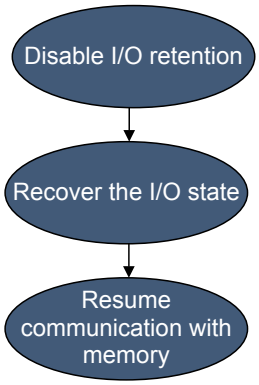


DT71657V2

- Memory interface init/config:
  It contains the IO and external device interface configuration.
- Check the wake-up process from Standby mode:
  Use the HAL API to check whether the system wakes up from Standby mode or not. This API __HAL_PWR_GET_FLAG enables checking the status and PWR_FLAG_SBF is the Standby mode flag check.
- Memory configuration:
  The STM32 interface (external device interface) initializes and configures the memory depending on the user use case.
- Run the application:
  Check the standby flag if zero then the application is kept running.
- Put the memory into low-power mode:
  Putting the memory into low-power mode enables the reduction of power consumption.

- Activate the I/O retention:
  In the Standby mode, the I/Os are by default in a floating state. If the IORETEN bit in the PWR_IORETR register is set, the GPIO output state is retained. The used API to activate it is the following one: HAL_PWREx_EnableStandbyIORetention
- Enter the Standby mode:
  When the CPU is not used or is in a polling mode, it is recommended to enter to the low-power mode to reduce the power consumption. The API used to enter this mode is HAL_PWR_EnterSTANDBYMode.

Figure 5. I/O retention process



DT71658V1

- Disable I/O retention: In PWR_IORETR write 0 to the IORETEN bit.
  HAL_PWREx_DisableStandbyIORetention is the used HAL API.
- Recover the I/O state: Read the last I/O state before entering the Standby mode.
- Resume communication with memory
- Run application: run again the application from where it stopped previously.

Revision history

Table 5. **Document revision history**

Date	Version	Changes
06-Mar-2023	1	Initial release.
27-Mar-2024	2	Updated: • Figure 1. STM32H5 series system architecture • Section 4.1: Static mode overview • Section 5.1: Tips for saving power consumption • Figure 4. Wake up from Standby mode with I/O retention Added Section 3.1: Bypass mode and LDO constraints

Contents

List of tables

List of figures