

STM32U575/585 power optimization

Introduction

The STM32U575/585 microcontrollers (MCUs) are based on the high-performance Arm® 32-bit Cortex®-M33 CPU with Arm® TrustZone® and FPU. These MCUs use an innovative architecture to reach best-in-class, ultra-low power figures thanks to their high flexibility and advanced set of peripherals. The STM32U575/585 devices provide a very-high energy efficiency for applications.

The STM32U575/585 devices with a "Q" suffix (such as STM32U5xxxQ) support the use of an internal SMPS in Run and low-power modes, enabling the design of very efficient and low power application

With the integration of the ART accelerator 8-Kbyte instruction cache, the STM32U575/585 MCUS can operate at frequencies up to 160 MHz and achieve 240 DMIPS performance, while maintaining extremely low dynamic power consumption.

The STM32U575/585 devices embed a high number of smart high-performing peripherals, a large set of advanced and ultra-low-power analog features. Many peripherals (including communication, analog, timers and audio) can be functional and autonomous down to Stop mode with direct memory access, thanks to the LPBAM (low-power background autonomous mode).

The combination of ultra-low-power design and processing performance allows these devices to achieve an industry leading EEMBC® ULPBench™ score, up to 535 ULPMark™.

The STM32U575/585 MCUs embed several innovations which minimize the consumption in the different modes, while maintaining most of the existing peripherals and an excellent pin-to-pin compatibility to allow an easy migration from existing families.

Thanks to their built-in internal voltage regulator and voltage scaling, the consumption in active modes is kept at a minimum whatever the external supply voltage. This makes these devices particularly suited for portable battery-supplied products, down to 1.71 V.

In addition, their multi-voltage domains allow the product to be supplied at low voltage while the analog-to-digital and digital-to-analog converters can operate with a higher supply and reference voltage, up to 3.6 V.

The STM32U575/585 devices support a battery Backup domain to keep the RTC (real-time clock) running, and a set of 32 registers, each 32-bit wide, that can be retained in case of power loss. This optional backup battery can be charged when the main supply is present.

These devices support several main low-power modes, each of them with several submode options. This allows the designer to achieve the best compromise between low-power consumption figure, shorter startup time, available set of peripherals and maximum number of wakeup sources.

1 General information

This application note applies to the STM32U575/585 microcontrollers that are Arm® Cortex® core-based devices.

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



Reference documents

- [1] Reference manual *STM32U575xx and STM32U585xx advanced Arm®-based 32-bit MCUs* (RM0456)
- [2] Datasheets for STM32U575xx (DS13737) and STM32U585xx (DS13086)
- [3] Application note *STM32 microcontroller GPIO configuration for hardware settings and low-power consumption* (AN4899)
- [4] Application note *STM32U575/585 power optimization using LPBAM* (AN5645)
- [5] EEMBC organization on <http://www.eembc.org>

2 Energy efficiency processing

The high processing performance in Run mode (expressed in DMIPS/MHz) is achieved thanks to the use of a Cortex-M33 core associated with the interfaces of its memories. To ensure full performance operation at maximum operating frequency, the STM32U575/585 devices embed the ART accelerator instruction cache (ICACHE), which masks the Flash memory access wait state. The processing performance can then achieve 1.5 DMIPS/MHz, whatever the system clock frequency.

The STM32U575/585 devices 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 system maximum operating frequency. Refer to the document [1] for more details.

The main regulator operates in the following ranges:

- Range 1 ($V_{CORE} = 1.2$ V) with CPU and peripherals running at up to 160 MHz
- Range 2 ($V_{CORE} = 1.1$ V) with CPU and peripherals running at up to 110 MHz
- Range 3 ($V_{CORE} = 1.0$ V) with CPU and peripherals running at up to 55 MHz
- Range 4 ($V_{CORE} = 0.9$ V) with CPU and peripherals running at up to 25 MHz

All consumption data used in this application note are based on typical specifications extracted from the document [2] at $T_A = 25^\circ\text{C}$ and $V_{DD} = 3.0$ V, unless otherwise specified.

Note: When the low-power regulator is used, the system clock can be either MSIS up to 24 MHz (4 MHz as default), or HSI16, depending on software configuration.

2.1 Internal regulator efficiency

The STM32U575/585 devices embed two internal regulators, that can be selected when the application runs, depending on the application requirements:

- a SMPS step-down converter
- a linear voltage regulator (LDO)

The LDO and SMPS regulators have two modes:

- main regulator (used when performance is needed)
- low power regulator

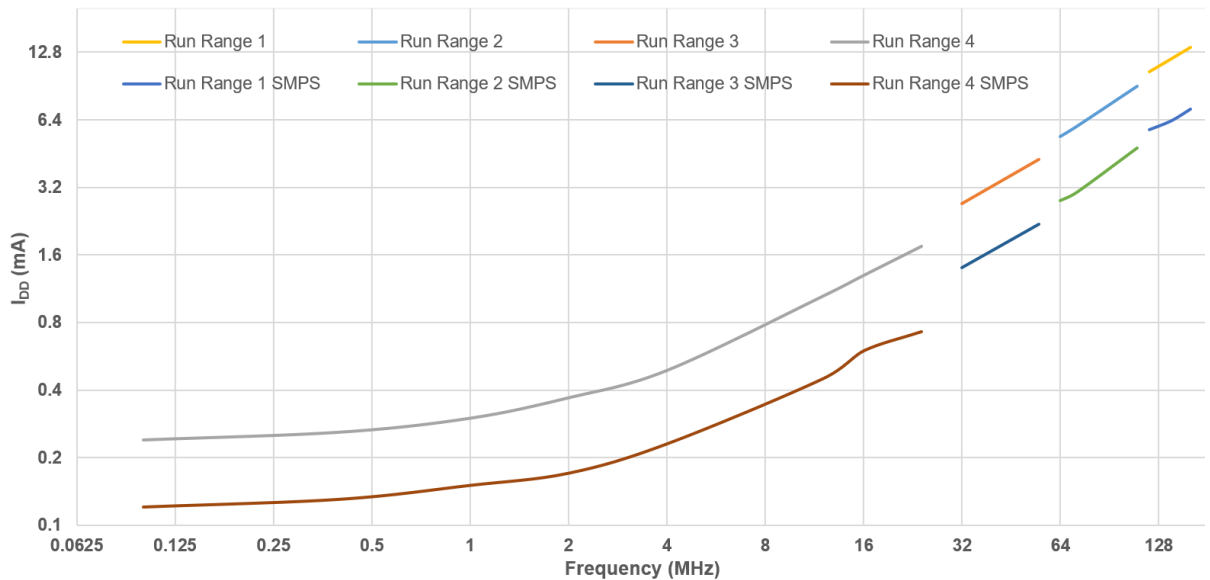
Both regulators can provide four different voltages (voltage scaling) and can operate in Stop mode.

Table 1. Power distribution of the internal regulators

Voltage regulator mode	Device mode
Main regulator (Range 1, 2, 3, 4)	Run, Sleep, Stop 0
Low-power regulator	Stop 1, Stop 2, Stop 3, (Standby with 8-Kbyte or full SRAM2)

The figure below shows the typical current consumption of an STM32U575/585 microcontroller in Run mode, as a function of system frequency, for both SMPS and LDO configurations.

Figure 1. Current consumption in Run with ICACHE ON, 1-way, prefetch ON, SMPS versus LDO

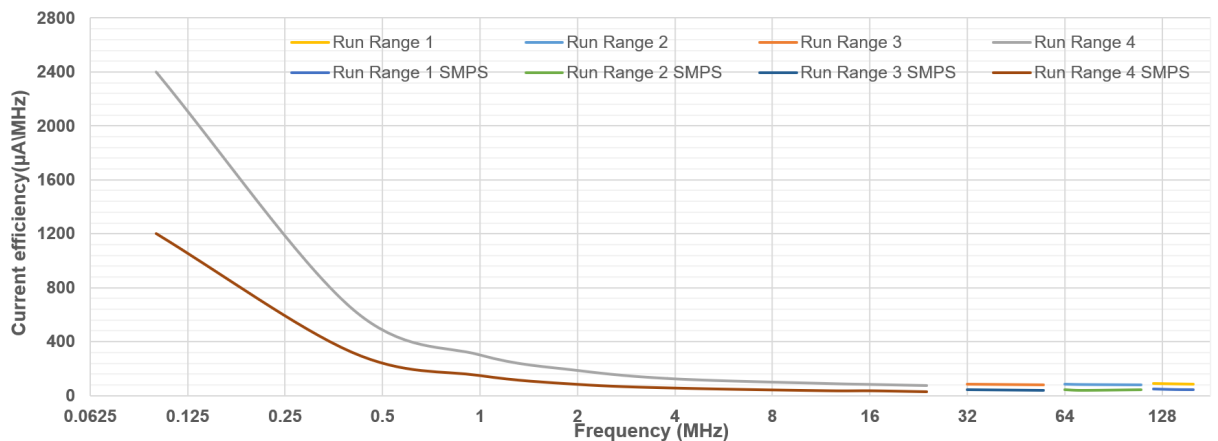


Note:

The lowest power consumption is achieved when running from the internal Flash memory with cache 1-way. The instruction cache tends to reduce the number of accesses to the memory thus reducing the overall current consumption. The current consumption from the internal SRAM is similar to the Flash memory when the ICACHE is used.

The figure below shows the same curve translated into power efficiency (by dividing the current consumption, multiplied by supply voltage to get power figure), by the CPU frequency.

Figure 2. Run mode power efficiency with ICACHE ON, 1-way, prefetch ON, SMPS versus LDO



2.2 ICACHE efficiency

The purpose of the instruction cache (ICACHE) is to cache instruction fetches or instruction memory loads coming from the processor. The ICACHE only manages read transactions and does not manage write transactions. The ICACHE reduces the power consumption by fetching instructions from the internal ICACHE most of the time, rather than from the bigger and more power consuming main memories.

The default ICACHE configuration (at reset) is a 2-way set associative cache. For applications needing very-low-power consumption profile, the ICACHE can be configured as 1-way, means direct-mapped cache.

Table 2. Current consumption in Run mode with ICACHE ON (1-way) versus prefetch

Fibonacci benchmark with SMPS and $V_{DD} = 1.8\text{ V}$

Frequency (MHz)	Current consumption (mA)	
	Prefetch ON	Prefetch OFF
160	8.25	8.35
110	5	5.06
55	2.4	2.41
24	0.97	0.97

Note: Prefetch tends to increase the code execution performance at the cost extra Flash memory accesses.

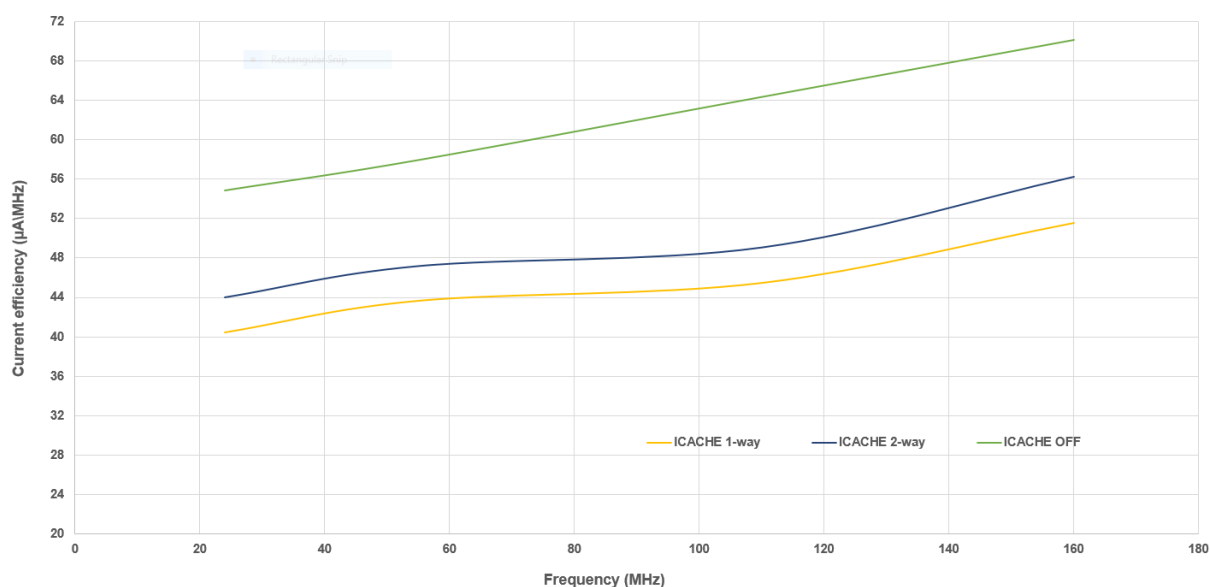
Table 3. Current consumption in Run mode with prefetch ON versus ICACHE configurations

Fibonacci benchmark with SMPS and $V_{DD} = 1.8\text{ V}$

Frequency (MHz)	Current consumption (mA)		
	ICACHE ON (1 way)	ICACHE ON (2 ways)	ICACHE OFF
160	8.25	8.99	11.21
110	5	5.39	7.075
55	2.4	2.59	3.186
24	0.97	1.05	1.316

The figure below shows the power efficiency of an STM32U575/585 microcontroller in Run mode on Fibonacci benchmark for different ICACHE configurations, as a function of system frequency, with SMPS and $V_{DD} = 1.8\text{ V}$.

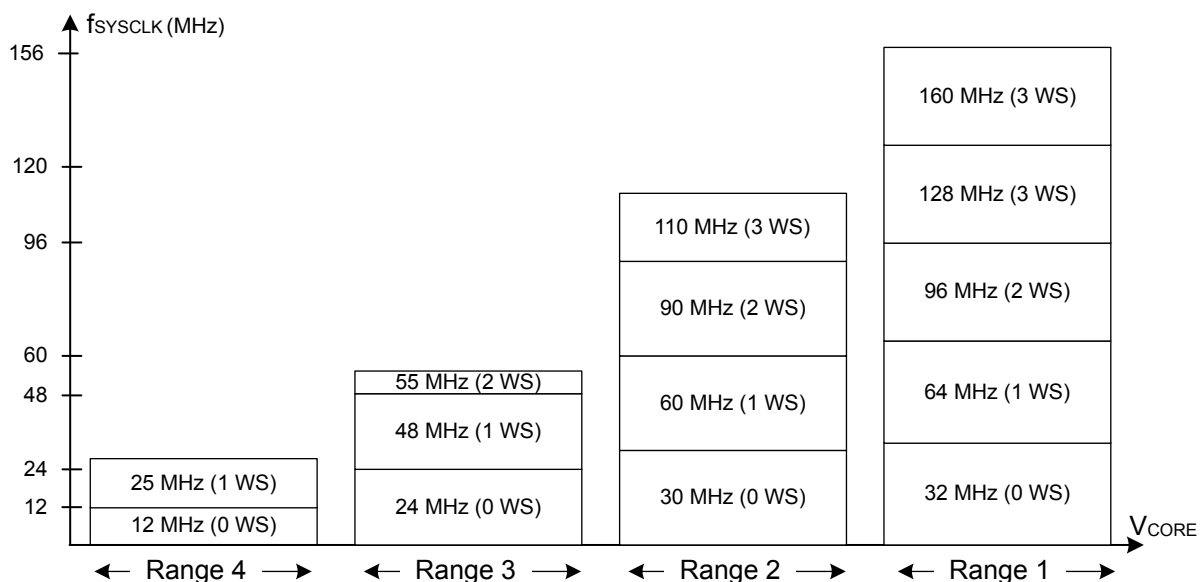
Figure 3. Power efficiency for various ICACHE configurations (SMPS, prefetch ON, $V_{DD} = 1.8\text{ V}$)



2.3 Flash memory efficiency

The figure below shows the Flash memory latency (number of wait states to be programmed in the Flash memory access control register), depending on the STM32U575/585 regulator voltage scaling range and system clock frequency.

Figure 4. Flash memory latency vs V_{CORE} range ($V_{DD} = 1.71$ to 3.6 V, LPM = 0)



The Flash memory supports a low-power read mode (LPM). The number of wait states (WS) depends on LPM:

- LPM = 0: up to 4 WS depending on supply voltage and frequency
- LPM = 1 (reduced consumption and increased latency): up to 15 WS depending on supply voltage and frequency

Table 4. Current consumption in Run mode versus LPM

Symbol	Conditions	Current consumption (mA)	
		LPM = 1	LPM = 0
$I_{DD(RUN)}$	SMPS, prefetch ON, Fibonacci code, $V_{DD} = 3.0$ V, Range 4	0.59	0.65

It is particularly interesting in Range 4 to reduce the dynamic consumption by enabling LPM in FLASH_ACR.

3 STM32U575/585 ultra-low-power feature

3.1 Low-power modes

The STM32U575/585 microcontrollers implement many different power modes. By default, the MCU is in Run mode after a system or a power reset. Several low-power modes are available to save power when the CPU does not need to be kept running, for example when waiting for an external event. It is up to the user to select the mode that gives the best compromise between low-power consumption, short startup time and available wakeup sources. Refer to document [1] for more details.

3.1.1 Sleep mode

CPU clock off, all peripherals including Cortex-M33 core such as NVIC and SysTick can run and wake up the CPU when an interrupt or an event occurs.

3.1.2 Stop mode

The Stop mode is based on the Cortex-M33 Deepsleep mode combined with the peripheral clock gating. All clocks in the core domain are stopped. The PLL, MSIS, MSIK, HSI16 and HSE oscillators are disabled with the possibility to enable MSIS, MSIK, HSI16 in Stop 0, Stop 1, or Stop 2 mode if a peripheral requests it.

Stop 0 and Stop 1 modes offer the largest number of active peripherals and wakeup sources, a smaller wakeup time but a higher consumption than Stop 2. In Stop 2 and Stop 3 modes, most of the core domain is put in a lower leakage mode.

The system clock when exiting from Stop mode can be either MSIS up to 24 MHz or HSI16, depending on software configuration.

Low-power background autonomous mode (LPBAM)

Some peripherals are autonomous and can operate in Stop 0, Stop 1, or Stop 2 mode by requesting their kernel clock (MSIK up to 24 MHz, or HSI16) and their bus (APB or AHB) when needed, in order to transfer data with DMA (GPDMA1 or LPDMA1 depending on peripherals and power mode).

3.1.3 Standby mode

The Standby mode is used to achieve the lowest power consumption with a Brownout reset (BOR): the internal regulator is switched off, so that the core domain is powered off. The PLL, the MSI (MSIS and MSIK) RC, the HSI16 RC, and the HSE crystal oscillators are also switched off. The RTC can remain active. 8, 56, or 64 Kbytes of SRAM2 can be optionally retained. Internal pull-up or pull-down can be applied to keep I/O level.

The system clock after wakeup is MSIS up to 4 MHz.

3.1.4 Shutdown mode

The lowest power consumption is reached in Shutdown mode. It is based on the Deepsleep mode with the voltage regulator disabled. The core domain is consequently powered off. This mode provides the lowest consumption by switching off the internal voltage regulators, and by disabling the voltage power monitoring. A wakeup from this mode is done thanks to one of the five wakeup pins or to the reset pin. The RTC clocked by the low-speed external oscillator (LSE) is also functional in this mode, with wakeup capability.

The 'low-power mode summary' table in 'Power modes' section of the document [1] summarizes the features available for each mode and provides an indication of the current consumption.

3.2 Power consumption optimization

3.2.1 ICACHE in Run mode

The ICACHE reduces the power consumption by fetching instructions from the internal ICACHE. Most of the time, applications with a lower-performance profile and stringent low-power consumption constraints, may benefit from the lower power consumption of an ICACHE configured as direct mapped (ICACHE_1-way).

This single-way cache configuration is obtained by programming WAYSEL = 0 in ICACHE_CR.

3.2.2 FLASH in Run and Sleep modes

The STM32U575/585 Flash memory includes a main memory block organized as two banks of 1 Mbyte each, containing 128 pages of 8 Kbytes. This structure is optimized in terms of power consumption with dedicated modes when the MCU is in Run and Sleep modes.

In order to reduce power consumption in Run and Sleep modes, each bank can be independently put in power-down mode by setting PDREQ1 or PDREQ2 in FLASH_ACR, depending on which bank used.

The Flash memory supports a low-power read mode to reduce power consumption by programming LPM = 1 in FLASH_ACR.

Note: *Prefetch tends to increase the code execution performance at the cost extra Flash memory accesses. However, for most applications, the power efficiency is better with prefetch ON.*

3.2.3 Power control optimization

During Standby mode, the BOR can be configured in discontinuous mode (ultra-low-power mode) to further reduce the current consumption by setting to 1 the ULPMEN bit in PWR_CR.

It is recommended to switch from LDO to SMPS to reduce power consumption in Run, Sleep, or any low-power mode.

- Note:**
- *The SMPS power supply pins are available only on a specific package with SMPS step-down converter option.*
 - *The SMPS to LDO and LDO to SMPS switches can be done on the fly.*

3.2.4 RCC in low-power modes

By default, the MSI bias is in continuous mode in order to maintain the output clocks accuracy. Setting MSIBIAS in RCC_ICSCSR1 before entering low-power mode reduce the MSI consumption when the regulator is in Range 4, and when the device is in Stop 1 or Stop 2 mode.

Before entering a low-power mode, it is recommended to select either HSI16 or MSIS (16 or 24 MHz) as wakeup source, in order to speed up the wakeup time, and to reduce the energy lost during wakeup. In case 24 MHz is used, one wait-state latency must be configured for SRAM read access, which impacts code execution performance. This performance must then be evaluated between using 24 MHz with 1 SRAM wait-state, and using 16 MHz as system clock.

Note: *The MSIS frequency during Stop mode or when waking up from Stop mode is the same as before entering Stop mode. If the MSIS is used to feed the PLL in Run mode, the MSIS is between 4 and 16 MHz due to PLL input clock frequency constraint. In this case, it is recommended to switch off the PLL and to configure MSIS with high frequency before entering Stop mode (if MSIS is used as wakeup clock).*

All the peripherals that cannot be enabled in Stop 2 mode must be either disabled by clearing the enable bit in the peripheral itself, or put under reset state by configuring the RCC registers.

Each clock source can be switched on or off independently when it is not used, to optimize power consumption.

3.2.5 I/O states in low-power modes

Refer to document [3] for more details.

Configure unused GPIO

A GPIO always have an input channel, which can be either digital or analog. If it is not necessary to read the GPIO data, prefer the configuration as analog input. This saves the consumption of the input Schmitt trigger.

Disable GPIO register clock

If a GPIO bank does not need to be used for a long period, disable its clock by using the `HAL_RCC_GPIOx_CLK_DISABLE()` function.

Configure GPIO when entering low-power modes

In Sleep, Stop 0, Stop 1, or Stop 2 mode, all I/O pins keep the same state as in the Run mode. For outputs, drive the level that is suitable depending on the external component required level.

In Stop 3, Standby, or shutdown mode, the I/Os are by default in floating state. Apply a pull-up or a pull-down depending on external component required level. With the knowledge that this pull-up/pull-down is not applied when exiting Shutdown mode, until the firmware configures the GPIO.

Note: *In some application, an external pull-up is recommended instead of an internal pull-up to further reduce power consumption in Stop modes, and to obtain a higher accuracy in the resistance value.*

3.2.6 Internal SRAMs

STM32U575/585 devices embed five SRAMs, each with specific features: SRAM1, SRAM2, SRAM3 are the main SRAMs. SRAM4 is in the SRAM used for peripherals low-power background autonomous mode (LPBAM) in Stop 2 mode.

All SRAMs and register contents are preserved, but the SRAMs can be totally or partially switched off to further reduced consumption. These SRAMs are made of several blocks that can be powered down:

- **Run mode**
To reduce power consumption in Run mode, power off all SRAMs by programming SRAMxPD in PWR_CR1.
- **Stop mode**
This mode achieves the lowest power consumption while retaining the content of SRAM and registers supplied by the low-power regulator. Stop 3 is the lowest power mode with full retention. To further optimize power consumption, the following is recommended:
 - Power down unused SRAMs by programming SRAMxPDSy = 1 in PWR_CR2:
 - SRAM1, SRAM3, SRAM4
 - SRAMs of DCACHE1, DMA2D, FDCAN, FMAC, PKA (when available) and USB
 - Depending on the application data size, retain SRAM2 page1 and/or page2 by clearing SRAMPDS1 and/or SRAMPDS2 bits in PWR_CR2.
- **Standby mode**
After entering Standby mode, SRAMs and register contents are lost except for registers and backup SRAM (2 Kbytes) in the Backup domain and Standby circuitry. SRAM2 page1 (8 Kbytes), or page2 (56 Kbytes), or both can be retained in Standby mode by setting RRSB1 and/or RRSB2 in PWR_CR1, supplied by the low-power regulator (Standby with RAM2 retention mode).

3.2.7 Peripherals clock gating in Run and Stop modes

STM32U575/585 devices support the capability to gate off AHB/APB clock in order to further reduce power consumption.

When none of the AHB/APB peripherals are used, and when their clocks are disabled, the following happens:

- All clocks of the AHB1 peripherals are off (except for BKPSRAM, DCACHE1, FLASH, ICACHE, and SRAM1) when setting AHB1DIS = 1 in RCC_CFGR2.
- All clocks of the AHB2 peripherals from RCC_AHB2ENR1 are off (except for SRAM2 and SRAM3), when setting AHB2DIS = 1 in RCC_CFGR2.
- All clocks of the AHB3 peripherals are off (except for SRAM4), when setting AHB3DIS = 1 in RCC_CFGR3.
- All clocks of the APB1/2/3 peripherals are off when setting APB1DIS = APB2DIS = 1 in RCC_CFGR2 and APB3DIS = 1 in RCC_CFGR3.

When a peripheral is enabled, its clock can be automatically gated off when the device is in Sleep mode, by clearing the peripheral SMEN bit in RCC_AHBxSMENR and RCC_APBxSMENR. Both EN and SMEN bits of the peripheral must be set to 1 in order to keep the clock on in Sleep mode.

The SMEN bit of the peripheral is also used to allow peripheral clocking in Stop 0 and Stop 1 modes, upon peripheral request. When the clock is requested by a peripheral, this clock is distributed to all enabled peripherals. Therefore, the SMEN bit must be cleared before entering Stop mode, if the peripheral is not used in Stop mode.

Caution: The SMEN bit of the peripheral must be set to allow the generation of an interrupt capable to wake up the device from Stop mode.

3.3 Power optimization of peripherals using LPBAM

The STM32U575/585 devices support a low-power background autonomous mode (LPBAM), that allows peripherals to be functional and autonomous in Stop 0, Stop 1, and Stop 2 modes (without any software running).

This functionality in Stop mode is possible thanks to the peripheral own independent clock (named kernel clock) request capability. This kernel clock is automatically switched on when requested by a peripheral, and automatically switched off when no peripheral requests it. Refer to document [1] for more details.

3.3.1 Peripherals supporting LPBAM

The table below lists all STM32U575/585 peripherals that support LPBAM.

Table 5. Peripherals supporting LPBAM

Low-power mode	Peripherals
Stop 0 and Stop 1	ADC4, ADF1, DAC1, GPDMA1, LPDMA1, LPTIMx (x = 1 to 4), LPUART1, MDF1, I2Cx (x = 1 to 4), SPIx (x = 1 to 3), USARTx (x = 1 to 5)
Stop 2	ADC4, ADF1, DAC1, LPDMA1, LPTIM1, LPTIM3, LPUART1, I2C3, SPI3

3.3.2 Main use cases

Here below examples of use cases that can be done while remaining in Stop mode (see document [4] for more details):

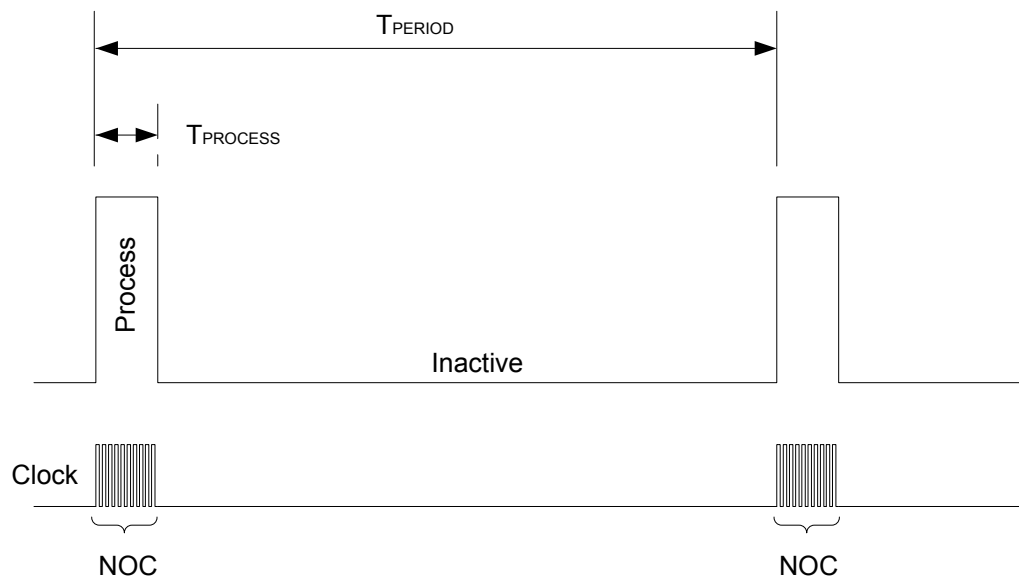
- ADC or DAC conversion triggered by a low-power timer (or any other trigger)
- Audio digital filter data transfer into SRAM: wakeup from Stop on sound activity detection
- I²C/SPI slave transfer, UART/LPUART reception: wakeup at the end of peripheral transfer or on DMA buffer event
- I²C/SPI master transfer, UART/LPUART transmission, triggered by LPTIM trigger (sensor periodic read for example): wakeup at the end of peripheral transfer or on DMA buffer event
- Bridges between peripherals. Example: ADC converted data transferred by communication peripherals
- Data transfer from a SRAM to another one

4 Low-power mode selection methodology

A simplified application model has been considered, where the application wakes up every T_{PERIOD} and performs some processing that is always the same and seen as a constant number of instructions (no waiting loops, no data dependencies). The following approximations can then be made:

- The process phase duration (T_{PROCESS}) can be defined in terms of the number of cycles (NOC) to be executed at each period. Defining F_{CLK} as the CPU system clock frequency, the duration of the process phase is equal to $T_{\text{PROCESS}} = \text{NOC} / F_{\text{CLK}}$. The average current consumption during this phase is equal to I_{PROCESS} .
- The inactive phase duration is $T_{\text{PERIOD}} - T_{\text{PROCESS}}$, and its average current consumption is I_{INACTIVE} .

Figure 5. Application sequence and parameters



In order to get the average current consumption, both consumptions during the process phase and during the inactive phase have to be summed up as follows:

$$I_{\text{AVERAGE}} = I_{\text{PROCESS}} \times \left(\frac{T_{\text{PROCESS}}}{T_{\text{PERIOD}}} \right) + I_{\text{INACTIVE}} \times \left(\frac{T_{\text{PERIOD}} - T_{\text{PROCESS}}}{T_{\text{PERIOD}}} \right)$$

The process phase duration is proportional to the number of cycles to execute, which is a constant in this use case:

$$I_{\text{AVERAGE}} = I_{\text{INACTIVE}} \times \left(\frac{(I_{\text{PROCESS}} - I_{\text{INACTIVE}}) \times \text{NOC}}{F_{\text{CLK}} \times T_{\text{PERIOD}}} \right)$$

This model does not take into account the consumption during the transition from inactive phase to process phase, which is addressed in [Section 4.5](#).

Two parameters need to be selected to optimize the average power consumption:

- Run mode and clock frequency to be used during the process phase
- low-power mode to be used during the inactive phase

4.1 Process phase

Depending on the processing requirement (DMIPS), one of the different Run modes can be used:

- Run mode with Range 1 voltage, with a maximum of 240 DMIPS when run at 160 MHz
- Run mode with Range 2 voltage, with a maximum of 165 DMIPS when run at 110 MHz
- Run mode with Range 3 voltage, with a maximum of 82.5 DMIPS when run at 55 MHz
- Run mode with Range 4 voltage, with a maximum of 37.5 DMIPS when run at 25 MHz

Another parameter that needs to be taken into consideration when selecting the frequency is the ability to support the constraints related to the peripherals (if any).

4.2 Inactive phase

The STM32U575/585 devices provides different low-power modes that can be used for the inactive phase:

- Sleep
- Stop 0 (with main regulator on)
- Stop 1 (with low-power regulator)
- Stop 2 (with low-power regulator)
- Stop 3 (with low-power regulator)
- Standby (with RTC and SRAM2 retention as options)
- Shutdown (with RTC as option)

Depending on the source of wakeup and the duration of the Sleep period, one of the following modes can be selected:

- Sleep: if the reactivity is the key parameter (in this case, the wakeup time is only eight system clock cycles)
- Stop 0: when the wakeup timing is critical ($< 2.5 \mu\text{s}$ if the program is in the Flash memory)
- Stop 1: if a lot of peripherals have to stay awake and the system has multiple sources of wakeup (in this case, the application must tolerate a wakeup time from Flash memory of about $13 \mu\text{s}$)
- Stop 2: if few peripherals have to stay awake and can generate wakeup event as DMA transfer complete, LPUART/SPI/I2C transfer event, voice detection through ADF, ADC analog watchdog (in this case, the application must tolerate a wakeup time from Flash memory of about $20 \mu\text{s}$)
- Stop 3: if no other peripherals than RTC need to stay awake with all SRAM retention, the wakeup time from Flash memory is $66.5 \mu\text{s}$ with $\text{FSTEN} = 0$
- Standby: if no other peripherals than RTC need to stay awake, and no more than 64-Kbyte SRAM2 retention is required (in this case, the application must tolerate a wakeup time from Flash memory of about $64.5 \mu\text{s}$ with $\text{MSIS} = 4 \text{ MHz}$ and $\text{FSTEN} = 0$)
- Shutdown: if only the RTC and the backup registers need to stay awake (in this case, the application must tolerate a wakeup time of about $610 \mu\text{s}$ with $\text{MSIS} = 4 \text{ MHz}$)

Note: *The wakeup timing depends on the code location (Flash memory), the system clock sources and the frequencies. Refer to document [2] for detailed conditions. The above wakeup from Stop figures correspond to MSIS at 24 MHz or 4 MHz with the code in Flash memory. To reduce wakeup time, it is recommended to set $\text{FLASHFWU} = 1$ and $\text{SRAM4FWU} = 1$ in PWR_CR2 .*

4.3 Sleep mode selection

The following numerical results are based on typical specifications extracted from the document [2] at 25 °C (typical value).

The figures below give the theoretical power consumption, for different selections of low-power mode (Sleep, Stop 1, Stop 2, Stop 3, Standby with RTC and SRAM2 retention, Standby and Shutdown) with RTC in the inactive phase, as a function of main supply (V_{DD}).

Figure 6. Low-power mode influence on average energy consumption (LDO)

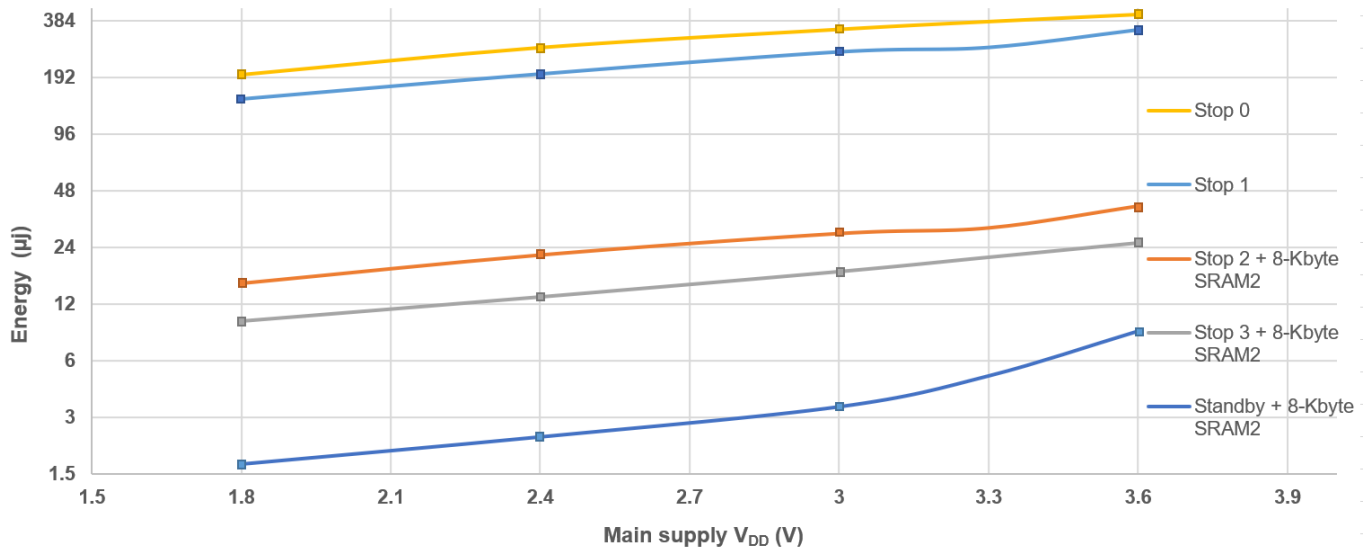
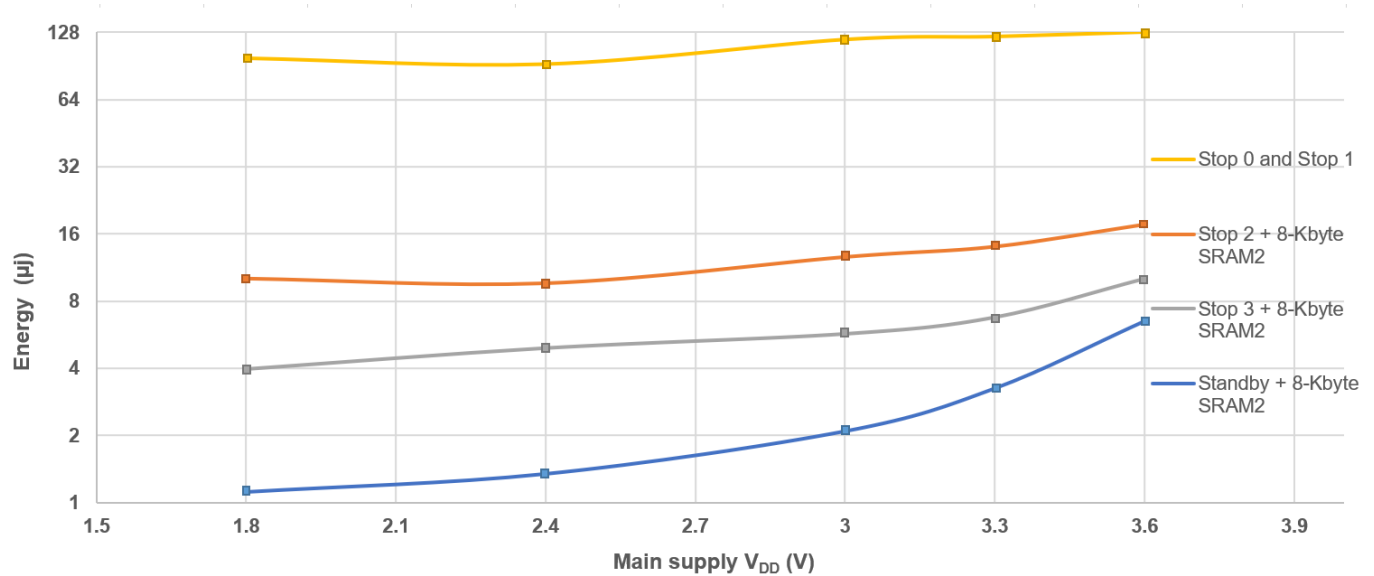


Figure 7. Low-power mode influence on average energy consumption (SMPS)



The previous figures show that, for applications requiring some data retention, the Standby mode gives the best energy consumption, and Stop 3 mode is the lowest power mode with all SRAM retention. Using SMPS reduces energy consumption to the half as showing [Figure 7](#).

The Shutdown mode has been discarded (dotted line) because it does not present enough data retention capability for this type of application.

The low-power mode selection is not only dictated by the overall power consumption figure, but also by other wakeup considerations linked to the application. Figure 6 and Figure 7 show that, for a small additional power consumption versus Standby, Stop 2 and Stop 3 modes offer a much powerful, simpler and faster setup for waking up (peripheral reinitialization not needed.)

4.4 Low-power mode selection

For a real application, the following rules apply, depending on the wakeup period:

- If the wakeup period is longer than some tens of ms, an implementation using Standby mode provides a better consumption.
- If the wakeup period is shorter, an implementation with Stop 2 or Stop 3 mode provides better results.
- If the temperature is high, the Standby mode is preferred.

The low-power mode selection depends not only on the power consumption but also on the wakeup time requirement (system reactivity) and the requirement for data retention.

If the processing can accommodate a frequency below 24 MHz, the optimum points are the following:

- Standby or Stop 3 mode
In Standby, the optimum points are a wakeup transition time longer than 66 μ s, and retention area lower than 64-Kbyte SRAM + 2-Kbyte BKPSRAM. In Stop 3 mode, all SRAMs can be retained.
- Stop 2 mode
The optimum points are a wakeup transition time longer than 20 μ s and retention area bigger than 66-Kbyte (64-Kbyte SRAM2 + 2-Kbyte BKPSRAM) retention.

If the wakeup period is longer than a few seconds and the retention memory is not needed, the Shutdown mode provides the best power performance. In this case the wakeup time is typically 610 μ s.

4.5 Impact of transition

To complete this study, the energy spent during the transition phases (wakeup and deactivation) must be considered.

According to the document [2], the energy spent at $V_{DD} = 3.0$ V, with SMPS, when leaving a low-power mode is roughly:

- 1.1 nAs for a transition from Stop 1 to Run mode (MSI = 24 MHz)
- 0.57 nAs for a transition from Stop 2 to Run mode (MSI = 24 MHz)
- 4.54 nAs for a transition from Stop 3 to Run mode (MSI = 24 MHz)
- 21.22 nAs for a transition from Standby to Run mode (MSI = 4 MHz)
- 457.29 nAs for a transition from Shutdown to Run mode (MSI = 4 MHz)

5 ULPMark-CP use case optimizations

5.1 ULPMark-CP description

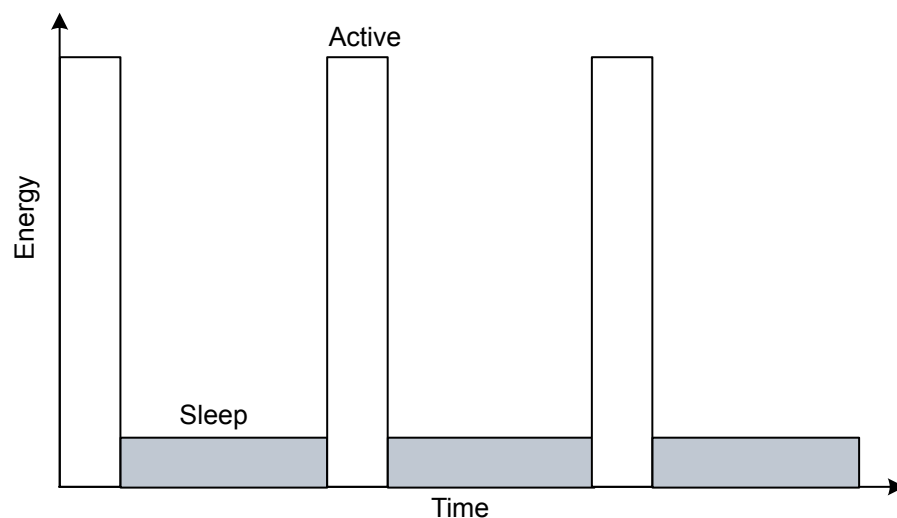
The ULPMark-Core Profile benchmark focuses on the MCU core, specifically the energy cost in sleep, and the transition to and from active mode. This benchmark uses a common set of workloads that are portable across 8-, 16-, and 32-bit microcontrollers. The Core Profile runs on a one-second duty cycle combining these workloads with an extended period of inactivity to enable the use of microcontroller low-power modes.

During the active portion of the test, the benchmark does the following:

- Generate 20 GPIO pulses.
- Perform an 8-bit linear interpolation.
- Perform a 16-bit integration (filter).
- Compute a 7-segment LCD binary conversion (saving state).
- Search for a substring in a string.
- Perform a small bubble-sort.
- Permute the bits of a string based on input and previous state.

The figure below shows the ULPMark-CP benchmark operating in long periods of sleep, followed by a brief wakeup to perform minimal processing, mimicking a sleepy-edge node conserving energy (see [5] for more details).

Figure 8. ULPMark-CP use case description



5.2 Use-case constraints

The ULPMark-CP imposes the following constraints:

- The application wakes up every second, based on an accurate RTC to perform some operations on a set of data.
- The data need to be maintained from one processing period to the next. This requires data retention during the inactive phase.
- The amount of processing cycles required by the application, once compiled using the compiler strongest optimization option, is about 10000 cycles every second.

There are no particular constraints regarding reactivity of the system, in order to serve external events or the RTC periodic interrupts.

5.3 Process phase optimization

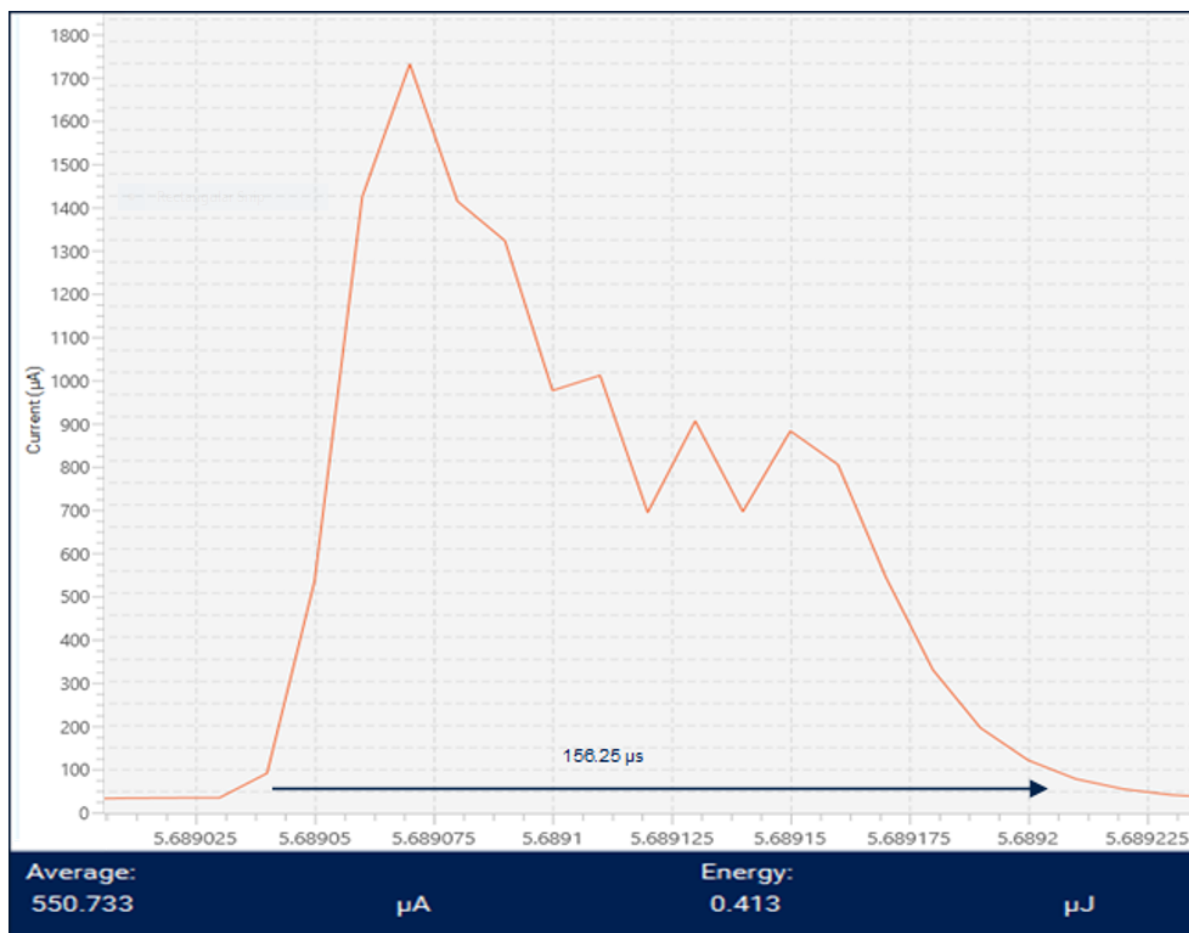
The active portion of the benchmark is only running for ~3% of the total runtime. The user can therefore use the power optimum point corresponding to the voltage scaling range 4 at 16 MHz, using the internal MSIS. When using the MSIS at 16 MHz nominal frequency, it consumes less than any other solution using the PLL.

To further optimize power consumption in active phase, the following steps can be applied:

1. Power down SRAM1, SRAM2, SRAM3, SRAM4 in Run mode.
2. Configure system clock as MSIS voltage scaling range 4 at 16 MHz.
3. Put data in BKPSRAM.
4. Disable unused AHB/APB clock.
5. Put all GPIO pins in analog mode.
6. Power down Flash Bank 2.
7. Enable Flash memory low-power mode.
8. Configure compiler optimization (high speed, no size constraints).

The figure below shows ULPMark-CP in active phase. It takes ~156 μ s between wakeup and entering in low-power mode with 550 μ A current consumption in Run mode at system clock MSIS 16 MHz.

Figure 9. STM32U575/585 ULPMark-CP in active phase



5.4 Inactive phase optimization

The ULPMark-CP benchmark requires to keep the data variables unchanged in low-power mode from one run to the next. Therefore, the Shutdown mode cannot be used because it does not provide enough data retention capability for this application.

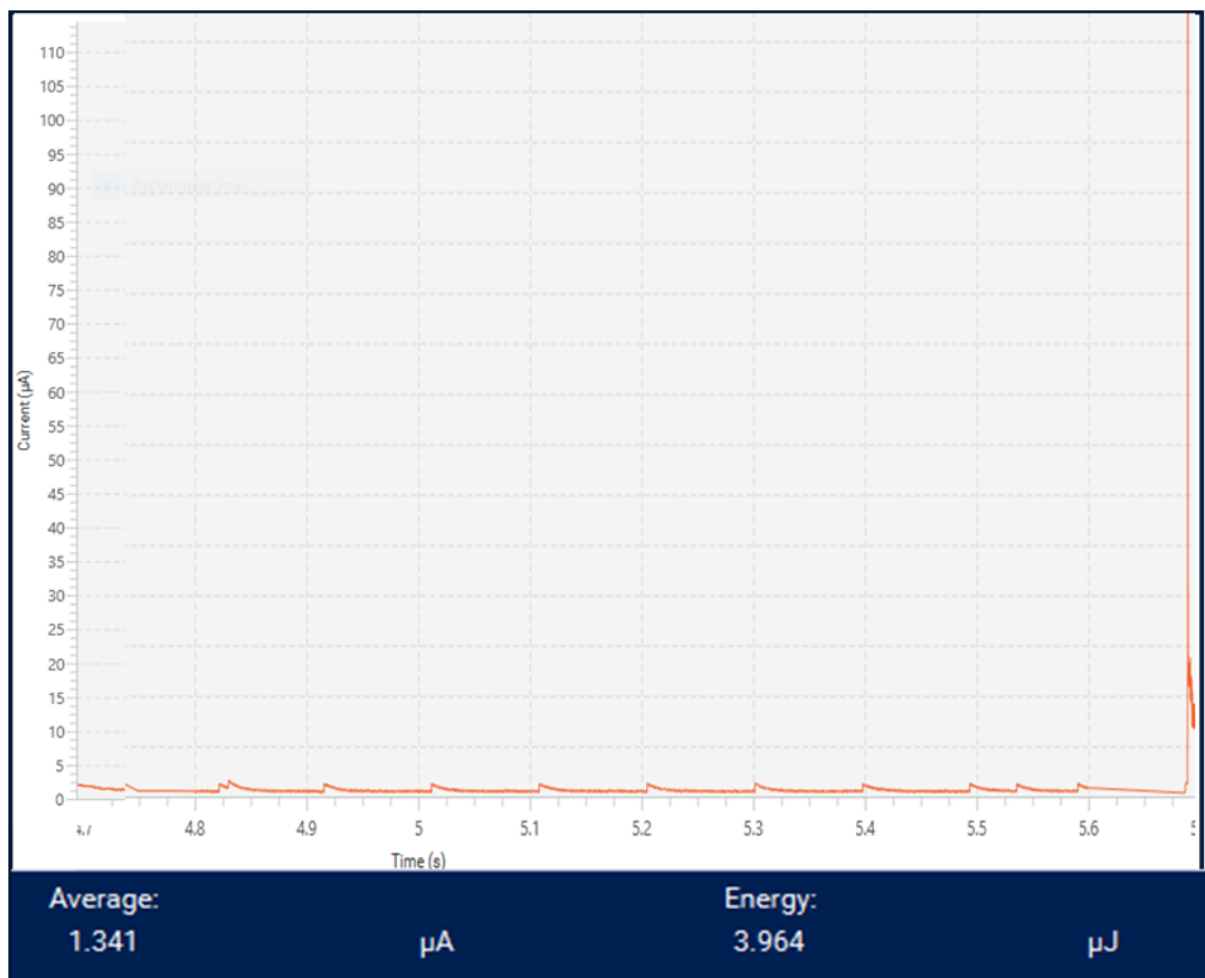
The Stop3 mode can be used with 8-Kbyte SRAM2 or 2-Kbyte BKPSRAM retention and with a lowest wakeup time. However the most power efficient solution uses the Standby mode with RTC and 8-Kbyte SRAM2 or 2-Kbyte BKPSRAM retention.

To further optimize power consumption in inactive phase, the following steps can be applied:

1. Enable SMPS regulator.
2. Enable ULPMEN bit.
3. Power down unused SRAM in Stop 3 mode.
4. Disable all clocks of AHB/APB peripherals (except AHB3/APB3 for RTC).

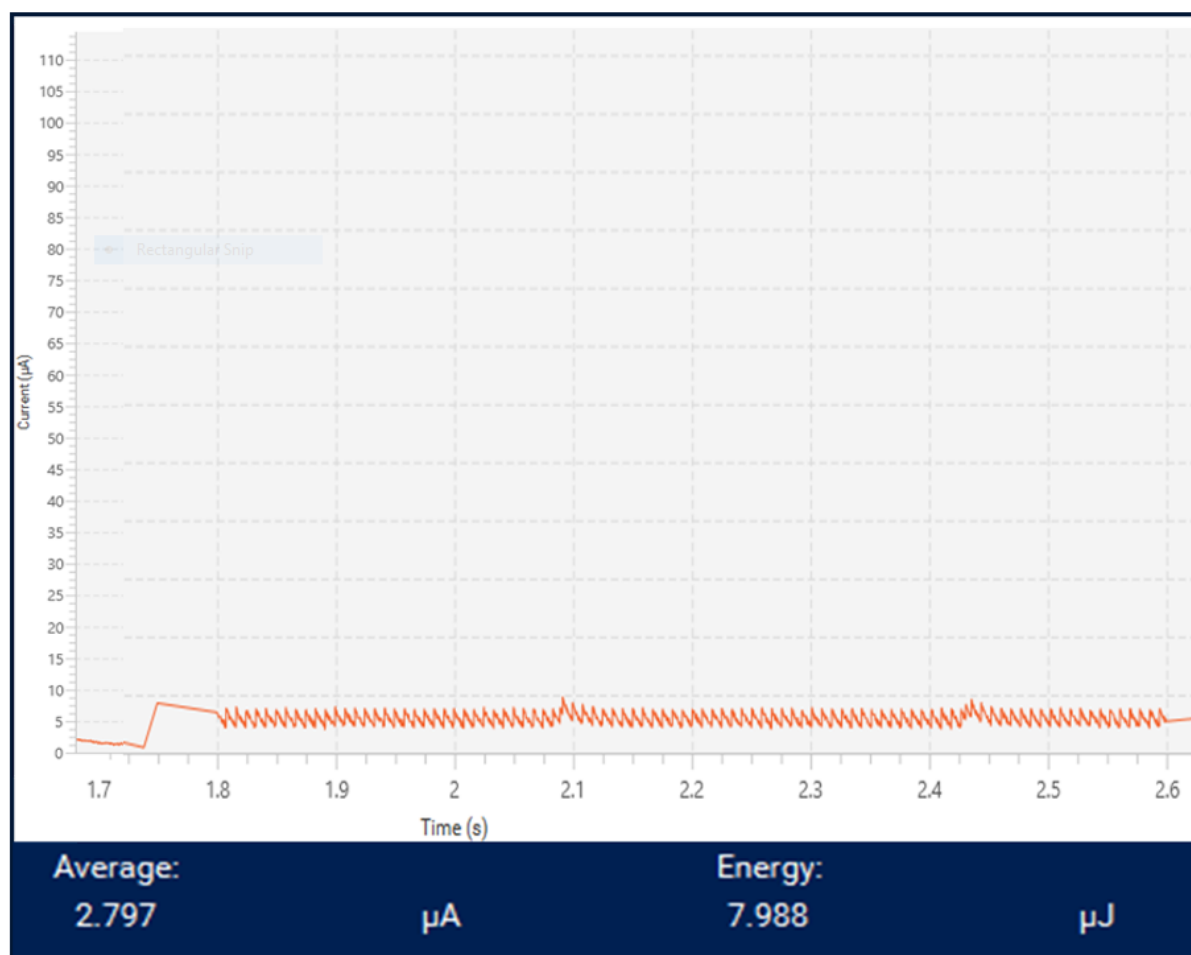
The figure below shows ULPMark-CP in inactive phase in Standby mode with an average current consumption of 1.34 μA .

Figure 10. STM32U575/585 ULPMark-CP in inactive phase with Standby mode



The figure below shows ULPMark-CP in inactive phase in Stop 3 mode with an average current consumption of 2.97 μA .

Figure 11. STM32U575/585 ULPMark-CP in inactive phase with Stop 3 mode

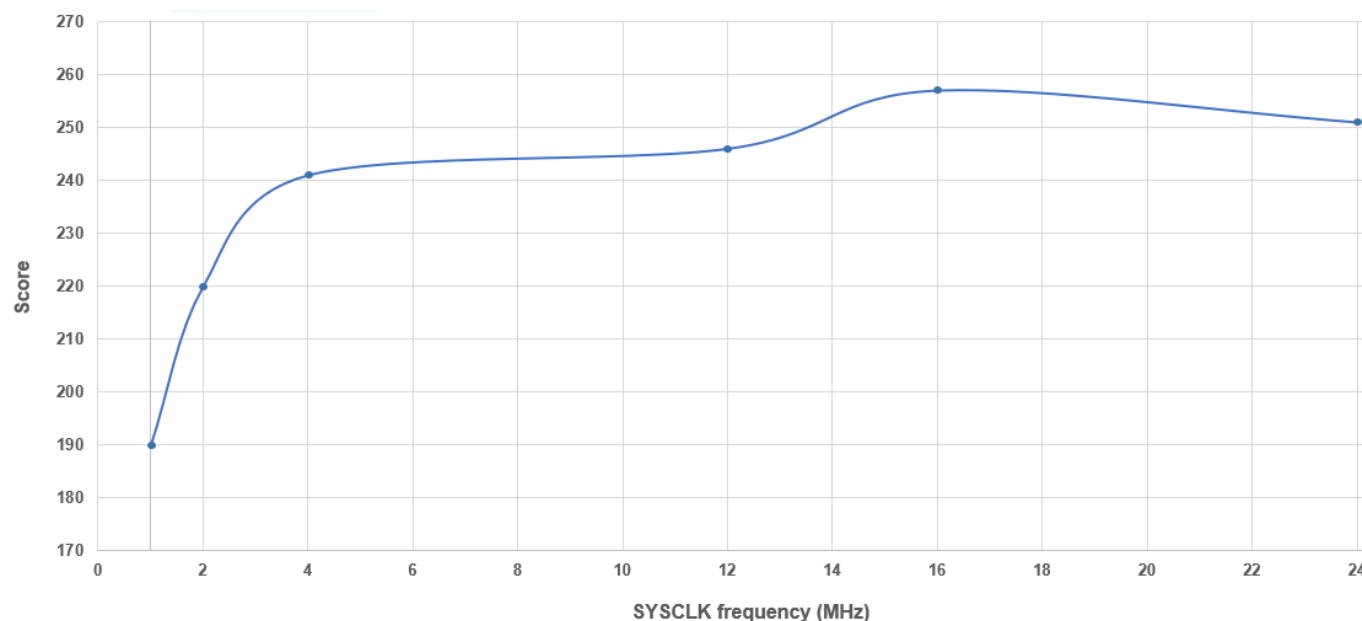


In the case of the ULPBench benchmark with a wakeup period of 1 second and a processing load of about 10 Kcycles per period, the optimum choice is to use the Standby mode for the inactive phase, associated with the voltage scaling range 4 mode at 16 MHz (from the MSIS) for the process phase.

5.5 STM32U575/585 measurements results

The figure below shows the ULPMark-CP score measurement at different system clock frequency used during the process phase. It uses the Standby mode with BKPSRAM retention and RTC for the inactive phase.

Figure 12. STM32U575/585 ULPMark-CP measurements versus frequency



The results are provided with different configurations of the clocks: for frequencies below 24 MHz, the MSI is used (because automatically started when waking up from Standby mode).

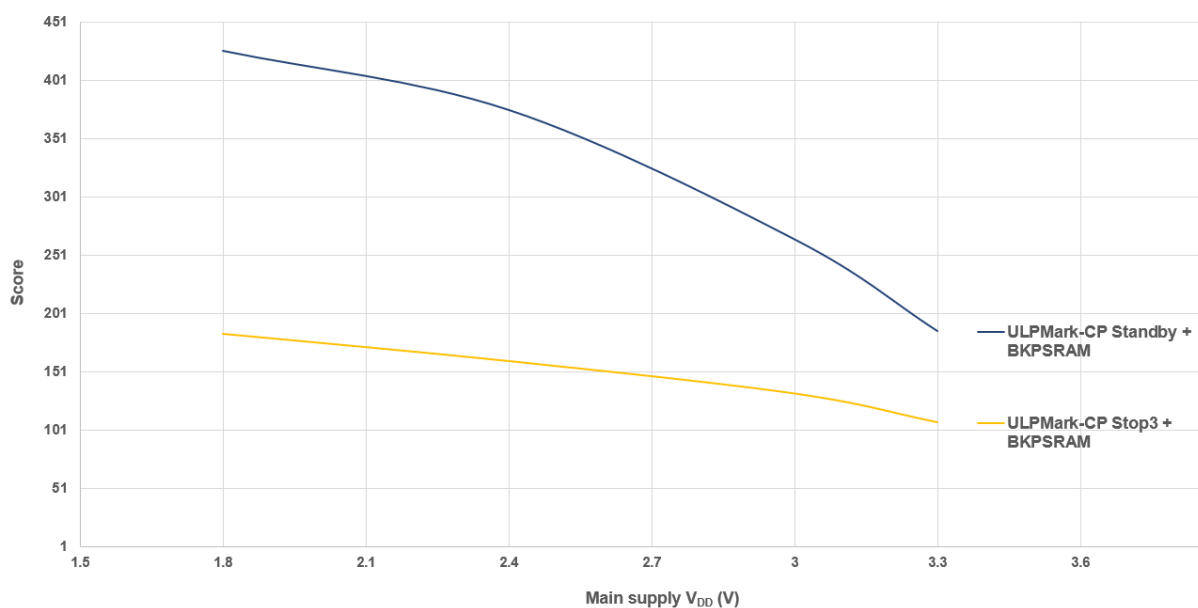
As already seen in the theoretical study, the voltage scaling range 4 at 16 MHz, using the MSIS provides the best results.

5.6 Performance evolution with voltage range

EEMBC defined base-test defaults on 3.0 V, but this voltage is not representative of battery supplied low-power applications. When the LDO regulator is used, the gain in power when moving to a lower voltage is significant compared to staying at 3.0 V. On the contrary, the power consumption decreases when increasing V_{DD} with SMPS regulator. This is why EEMBC introduced a comparison on user-defined voltage, letting manufacturers choose ideal operating conditions.

The figure below shows the ULPMark-CP score measurement at different voltages.

Figure 13. Equivalent score versus supply voltage



The above curves correspond to the voltage scaling range 4 at 16 MHz MSIS mode. It clearly shows the advantages of a voltage reduction.

6 Conclusion

The main ultra-low-power features of the STM32U575/585 microcontrollers are presented in this application note. They show the benefits offered by this microcontroller family to reduce the current consumption in embedded systems. These devices offer a large choice of options for optimizing both performance and power consumption, whatever the application.

This document provides guidelines based on experiments and quantitative results in order to help quickly selecting the best running and low-power modes, according to the characteristics and constraints of the end-user application.

Revision history

Table 6. Document revision history

Date	Version	Changes
6-Oct-2021	1	Initial release.

Contents

1	General information	2
2	Energy efficiency processing	3
2.1	Internal regulator efficiency	3
2.2	ICACHE efficiency	4
2.3	Flash memory efficiency	6
3	STM32U575/585 ultra-low-power feature	7
3.1	Low-power modes	7
3.1.1	Sleep mode	7
3.1.2	Stop mode	7
3.1.3	Standby mode	7
3.1.4	Shutdown mode	7
3.2	Power consumption optimization	7
3.2.1	ICACHE in Run mode	7
3.2.2	FLASH in Run and Sleep modes	8
3.2.3	Power control optimization	8
3.2.4	RCC in low-power modes	8
3.2.5	I/O states in low-power modes	8
3.2.6	Internal SRAMs	9
3.2.7	Peripherals clock gating in Run and Stop modes	9
3.3	Power optimization of peripherals using LPBAM	9
3.3.1	Peripherals supporting LPBAM	10
3.3.2	Main use cases	10
4	Low-power mode selection methodology	11
4.1	Process phase	12
4.2	Inactive phase	12
4.3	Sleep mode selection	13
4.4	Low-power mode selection	14
4.5	Impact of transition	14
5	ULPMark-CP use case optimizations	15
5.1	ULPMark-CP description	15
5.2	Use-case constraints	15
5.3	Process phase optimization	16
5.4	Inactive phase optimization	17
5.5	STM32U575/585 measurements results	19
5.6	Performance evolution with voltage range	20

6	Conclusion21
	Revision history22
	List of tables25
	List of figures.....	.26

List of tables

Table 1.	Power distribution of the internal regulators	3
Table 2.	Current consumption in Run mode with ICACHE ON (1-way) versus prefetch	5
Table 3.	Current consumption in Run mode with prefetch ON versus ICACHE configurations.	5
Table 4.	Current consumption in Run mode versus LPM	6
Table 5.	Peripherals supporting LPBAM	10
Table 6.	Document revision history	22

List of figures

Figure 1.	Current consumption in Run with ICACHE ON, 1-way, prefetch ON, SMPS versus LDO	4
Figure 2.	Run mode power efficiency with ICACHE ON, 1-way, prefetch ON, SMPS versus LDO	4
Figure 3.	Power efficiency for various ICACHE configurations (SMPS, prefetch ON, $V_{DD} = 1.8\text{ V}$).	5
Figure 4.	Flash memory latency vs V_{CORE} range ($V_{DD} = 1.71$ to 3.6 V , LPM = 0)	6
Figure 5.	Application sequence and parameters	11
Figure 6.	Low-power mode influence on average energy consumption (LDO).	13
Figure 7.	Low-power mode influence on average energy consumption (SMPS)	13
Figure 8.	ULPMark-CP use case description	15
Figure 9.	STM32U575/585 ULPMark-CP in active phase.	16
Figure 10.	STM32U575/585 ULPMark-CP in inactive phase with Standby mode	17
Figure 11.	STM32U575/585 ULPMark-CP in inactive phase with Stop 3 mode	18
Figure 12.	STM32U575/585 ULPMark-CP measurements versus frequency	19
Figure 13.	Equivalent score versus supply voltage	20

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2021 STMicroelectronics – All rights reserved