



---

## How to optimize power and performance on STM32L4 and STM32L4+ MCUs

### Introduction

The **STM32L4** and **STM32L4+ Series** microcontrollers are designed using an innovative architecture to reach best-in-class, ultra-low-power figures thanks to their high flexibility and advanced set of peripherals.

The **STM32L4 series** devices are based on the Cortex<sup>®</sup>-M4 with FPU core. They can operate at up to 80 MHz and achieve 100 DMIPS performance at 80 MHz, thanks to the integration of the ART Accelerator, while maintaining the smallest possible dynamic power consumption. The **STM32L4+ series** stepped up the performance by 50%, going to 120 MHz and 150 DMIPS.

Both **STM32L4** and **STM32L4+ Series** feature the FlexPowerControl function, which increases flexibility in power mode management, while at the same time reducing the overall application consumption.

In order to maximize the battery life and/or reduce its cost, the selection of the processor working mode is critical. In addition to power consumption considerations, the application constraints also have to be taken into account. Consequently the microcontroller needs to offer a wide range of working modes in order to support all applications, while always providing close-to-optimum power performance.

In many ultra-low-power applications the microcontroller is used with long periods of sleep time followed by very short periods of dense processing.

This application note provides qualitative and quantitative information in order to be able to configure different parameters (such as frequency, range, and low-power mode), before starting the implementation and optimization. All the computations are performed with typical data from the product datasheet and at ambient temperature, unless otherwise specified.

This application note uses the industry-standard ULPMark<sup>™</sup> benchmark from EEMBC as a reference case to correlate the computations and simulations with the measurements.

## 1 General information

STM32L4 and STM32L4+ Series microcontrollers are built around an Arm® Cortex®-M4 with FPU and DSP instruction set.

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



### Related documents

STM32L4/L4+ Series core documentation and in particular:

- [1] *STM32L47xxx, STM32L48xxx, STM32L49xxx and STM32L4Axxx advanced Arm®-based 32-bit MCUs reference manual (RM0351)*
- [2] *STM32L41xxx/42xxx/43xxx/44xxx/45xxx/46xxx advanced Arm®-based 32-bit MCUs reference manual (RM0394)*
- [3] *STM32L4 and STM32L4+ ultra-low-power features overview application note (AN4621)*
- [4] STM32L4 datasheets
- [5] *Design recommendations for STM32L4xxxx with external SMPS, for ultra-low-power applications with high performance application note (AN4978)*
- [6] EEMBC organization, <http://www.eembc.org>
- [7] *STM32L4+ Series advanced Arm®-based 32-bit MCUs reference manual (RM0432)*
- [8] *STM32 cross-series timer overview application note (AN4013)*

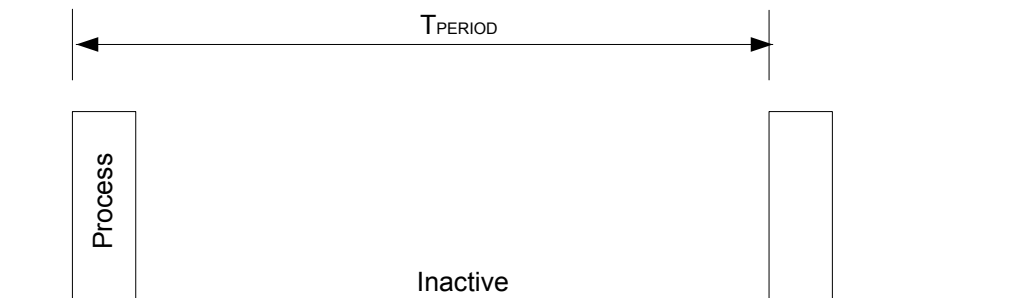
## 2 Low-power application profile

In applications where the battery lifetime is a concern, the system must be optimized to provide maximum performance and reactivity at the minimal power consumption.

This type of application contains generally two different phases:

- Process phase, in which some sensor or radio information needs to be processed at either regular time intervals (RTC) or on external events (GPIO, interrupts..).
- Inactive phase, in which the system is sleeping and waiting for either RTC or GPIO wakeup.

Figure 1. Application profile



### 2.1 Key parameters to consider

Both phases have different performance criteria (in addition to power consumption):

- Process phase requires a certain number of instructions to be executed in a given time frame.
- Inactive phase requires a certain minimal context to be kept (data RAM retention) and wakeup capability (peripheral, GPIO, RTC...) while keeping the power consumption as low as possible.

Furthermore, the transition between both phases must support different constraints:

- Sleep-to-Run transition requires a fast wakeup time with the smallest peak current (also called inrush current) at the time of transition, in order to allow the external supply to settle.
- Run-to-Sleep transition has less timing concerns, but needs to be optimized for energy saving.

When operating in such burst mode, four parameters need to be considered:

- Average power consumption. This parameter ( $P_{avg}$  expressed in  $\mu W$ ) determines the battery size in order to support the expected autonomy.
- Maximum peak current. This parameter ( $I_{peak}$  expressed in  $\mu A$ ) determines the type of battery that can be used but also the number and size of external components (decoupling capacitors) that need to be added on the board.
- Processing performance in Process phase. This parameter is expressed in DMIPS which is proportional to the CPU frequency (Freq).
- Reaction time. This is the time between the activation of the wakeup signal source and the execution of the first instruction, generally within the interrupt service routine (ISR), once the context has been restored (power, data space context, clock).

## 2.2 ULPMark™ description

The EEMBC ULPMark™ benchmarks are well suited for evaluating ultra-low-power performance. It is a standardized test that allows the characterization of any 8-, 16- and 32-bit microcontroller.

The ULPMark™ CoreProfile (ULPMarc-CP) allows the evaluation of the following resources energy efficiency:

- CPU, RAM and Flash memory
- RTC timer and wakeup mechanism
- 32 kHz LSE
- Power management circuitry

During the Process phase, the EEMBC ULPMark™ CoreProfile performs:

- Data array manipulation including permutations and sorting
- Simple filtering using 8 and 16-bit mathematics
- Simple state machine
- Simplified RTOS called TES, that allows to test real time events.

ULPMark™ PeripheralProfile (ULPMark-PP) is focused on typical peripherals found in every microcontroller:

- SPI (serial communication)
- ADC (data acquisition, measurement)
- Timer (PWM, actuators)
- RTC (time reference)

The MCU is supposed to perform a series of actions involving the mentioned peripherals with minimal energy budget.

ULPMark™ CoreMark builds on existing CoreMark test and evaluates number of iterations executed for a unit of energy.

For comparing different microcontrollers, the ULPMark™ result is expressed as a score. For more information on ULPMark™, refer to [6].

## 3 STM32L4/L4+ Series low-power feature

### 3.1 Numerous low-power modes

The STM32L4/L4+ Series microcontrollers implement many different power modes, seven of them are low-power. On top of these modes, the power consumption can be moderated by selecting different clock sources and frequencies, as well as gating clock for off peripherals not in use.

In all these modes, except Shutdown, the safe power monitoring Brown out reset (BOR) and the IWDG can stay active to guarantee safe execution. More details are provided in AN4621 [3].

The usage of an external SMPS is described in AN4978 [5].

All consumption data used in this application note are based on typical specifications extracted from the STM32L476xx datasheet [4] at  $T_A = 25^\circ\text{C}$  and  $V_{DD} = 3.0\text{V}$ , unless otherwise specified.

The STM32L4R5 figures are used where STM32L4+ series specific features are presented. While the numbers are different for each particular product, their ratios remain largely consistent and general principles stand unchanged.

#### 3.1.1 Low-power run and Low-power sleep modes

Two low-power active modes are available on the STM32L4/L4+ Series in addition to those already used on the STM32Fx series (Sleep, Stop and Standby): the Low-power run and Low-power sleep modes.

They offer Run and Sleep mode functionality for applications with extremely low current consumption where some peripherals cannot be switched off, or where the CPU is processing continuously at low speed to minimize current variations.

Several features have been put in place to reduce the current consumption:

- The core logic is supplied by the low-power voltage regulator to reduce the quiescent current.
- The Flash memory can be switched off (power-down mode and clock gating) in Low-power sleep mode. It can also be switched off in Low-power run when the processor is executing from SRAM1 or SRAM2.
- The system clock is limited to 2 MHz maximum. The MSI internal RC oscillator can be selected as it supports several frequency ranges, with a small MCU total consumption down to 14  $\mu\text{A}$  (on 255 Kbytes devices) in Low-power sleep Flash memory off at 100 kHz.

#### Batch acquisition sub-mode (BAM)

The STM32L4/L4+ Series microcontrollers support the power efficient batch acquisition sub-mode (BAM), in which data are transferred with communication peripherals, while the rest of the device is in low-power mode.

This is achieved by entering Sleep or Low-power sleep mode with this configuration:

- Only the DMA, the communication peripheral(s) and the SRAM1 or SRAM2 clocks are enabled in Sleep (or Low-power sleep) mode.
- The Flash memory is off in Sleep (or Low-power sleep) mode: the Flash memory is in power-down and the Flash memory clock is gated off.
- If the system clock can be limited to 2 MHz, the main regulator is switched off (to enter Low-power sleep).

In Low-power sleep mode, the I2C and USART/LPUART peripherals can still be clocked with HSI at 16 MHz. This allows supporting BAM with I2C or USART at up to 1 Mbps speed.

#### 3.1.2 Stop mode

The STM32L4/L4+ Series implements three Stop modes with full SRAM and peripheral retention capability and capacity to wakeup in 1  $\mu\text{s}$  thanks to the use of the MSI up to 48 MHz.

In these Stop modes all the high speed oscillators (HSE, MSI, HSI) are stopped, while the low speed ones (LSE, LSI) can be kept active. The peripherals can be set active, using the HSI clock when needed, to be able to wakeup the device on some specific events (such as UART character reception or I<sup>2</sup>C address recognition).

The Stop 2 mode implements a dedicated mechanism to keep the retention current as low as possible while allowing a very fast wakeup of 5  $\mu\text{s}$  from SRAM or 8  $\mu\text{s}$  from Flash memory.

#### 3.1.3 Standby mode

In Standby mode the BOR is always enabled, ensuring that the device is under reset when the supply voltage is below the selected functional threshold.

By default the SRAMs content is lost in Standby mode. However, it is possible to preserve the content of the SRAM2 (see table below).

**Table 1. SRAM2 retained content**

Device	SRAM2 retained (Kbytes)	Consumption (nA)
STM32L49xxx/L4Axx	64	374
STM32L47xxx/L48xxx	32	230
STM32L45xxx/L46xxx	32	250
STM32L43xxx/L44xxx	16	200
STM32L4P5xx/L4Q5xx <sup>(1)</sup>	64	615
	4	230
STM32L41xxx/L42xxx	8	100
STM32L4Rxxx/L4Sxxx	64	385

1. The amount of retained memory is configurable for these devices.

Pull-up and pull-down can individually be applied on each I/O during the Standby mode, allowing the external device configuration to be kept.

A wakeup from this mode is done thanks to one of the five wakeup pins, the reset pin or the independent watchdog. The RTC clocked by the low-speed oscillators (LSE or LSI) is also functional in this mode, with wakeup capability.

### 3.1.4 Shutdown mode

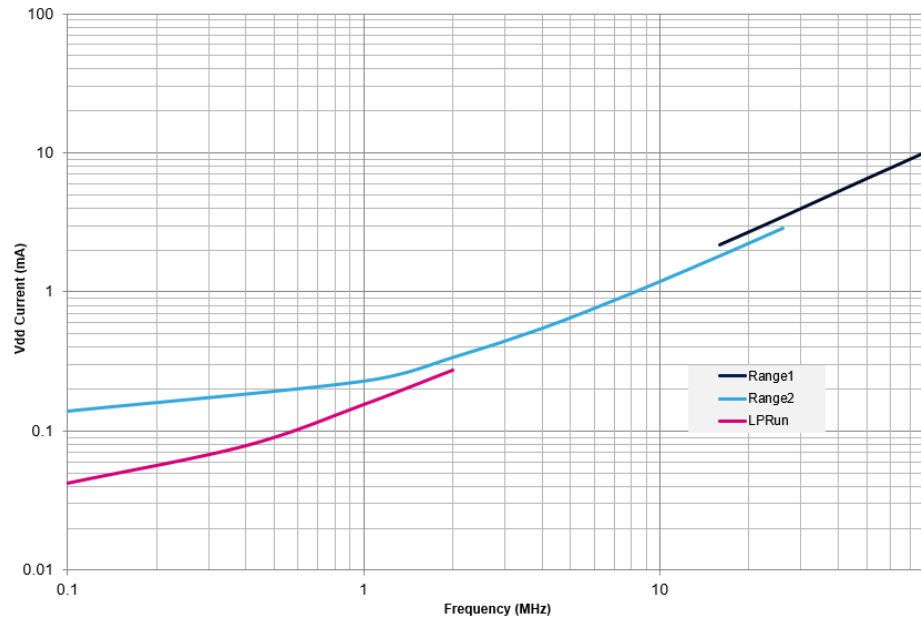
A Shutdown mode is implemented in the STM32L4/L4+ Series devices in order to lengthen even more the battery life of battery-powered applications.

This mode allows the lowest consumption (8 nA at 1.8 V for 256 Kbytes devices), by switching off the internal voltage regulators, and by disabling the voltage power monitoring. A wakeup from this mode is done by 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.

### 3.2 Run mode power consumption

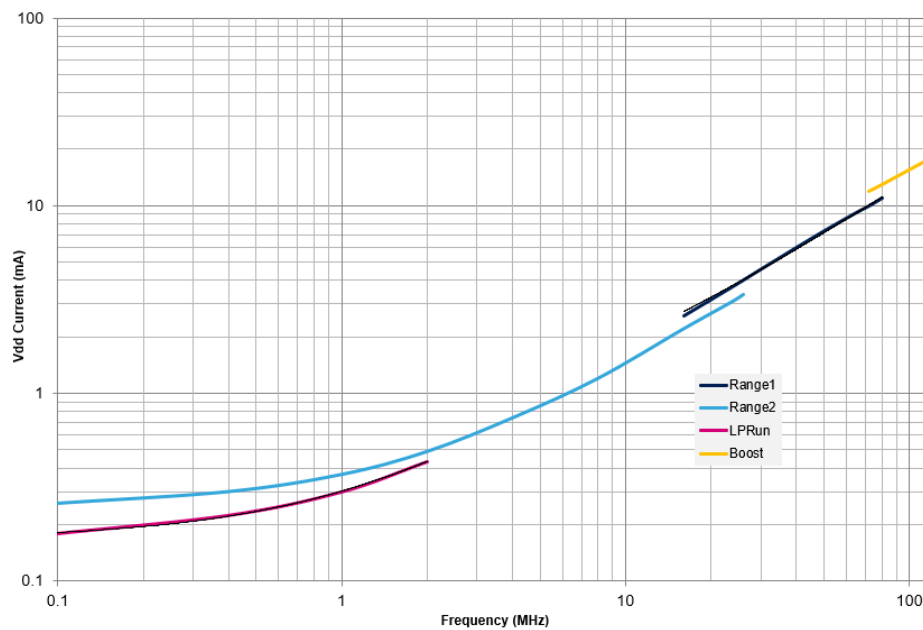
The following figure, extracted from [2], gives the typical current consumption values for the STM32L476 at 25 °C for different CPU frequencies while running the Fibonacci algorithm at 3.0 V supply:

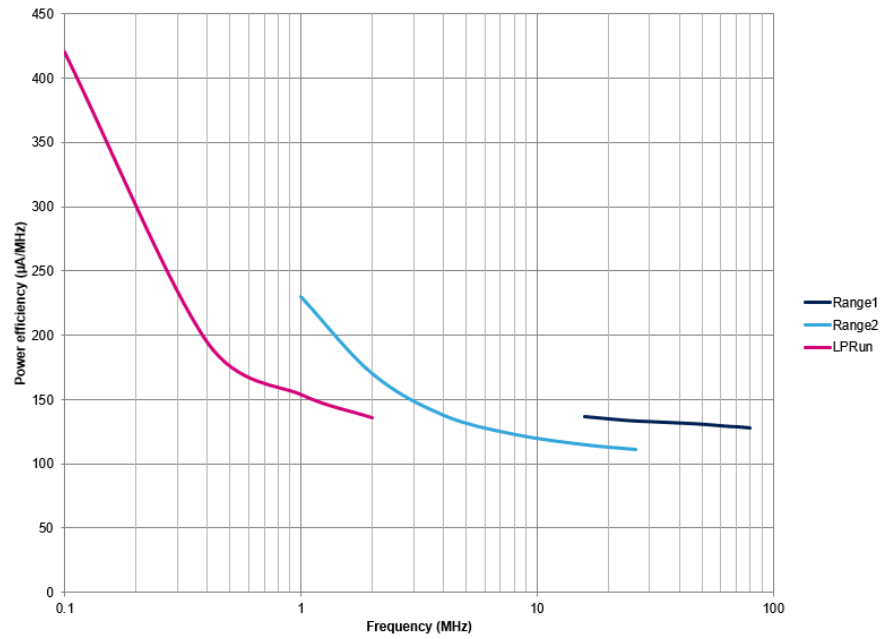
**Figure 2. STM32L476 Run mode current consumption**



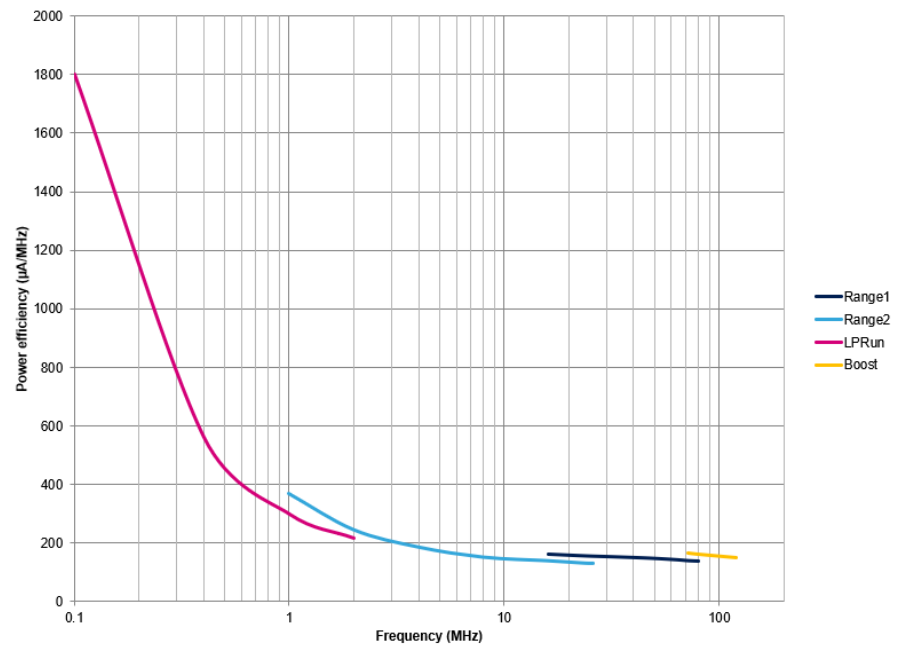
This curve can be translated into power efficiency by dividing the current consumption, multiplied by supply voltage to get power figure, by the CPU frequency, see the following figure.

**Figure 3. STM32L4R5 Run mode current consumption**



**Figure 4. STM32L476 Run mode power efficiency**


The optimum efficiency is obtained in Range 2 at the maximum frequency of 26 MHz.

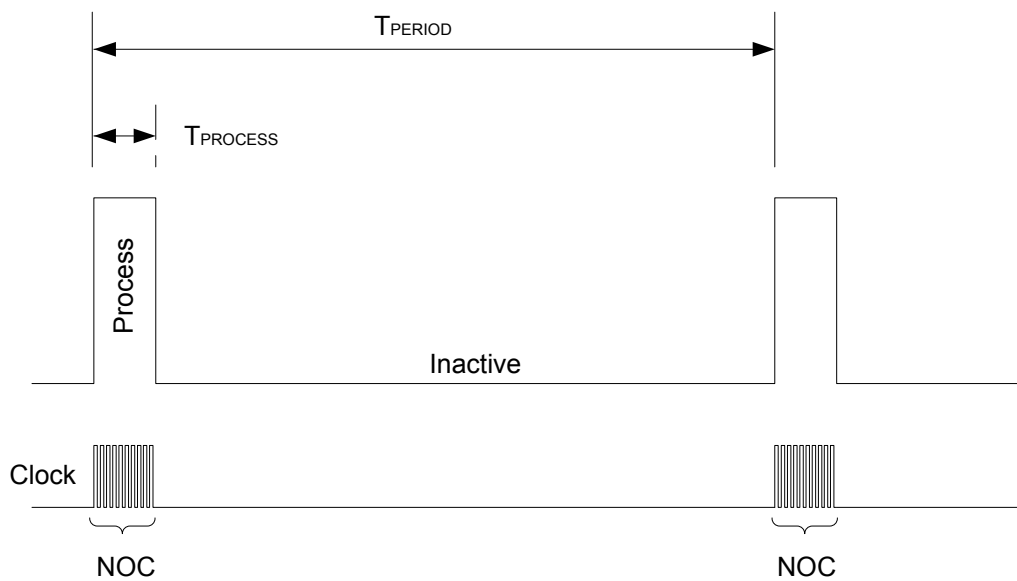
**Figure 5. STM32L4R5 Run mode power efficiency**


## 4 Low-power mode selection methodology

A simplified application model has been considered, where the application wakes up every  $T_{\text{PERIOD}}$  and performs some processing that is always the same and seen as a constant number of instructions (no waiting loops, no data dependencies). This allows to make the following approximations:

- The Process phase duration ( $T_{\text{PROCESS}}$ ) can be defined in terms of the number of cycles (NOC) to be executed at each period. Defining  $F_{\text{CLK}}$  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  $I_{\text{PROCESS}}$ .
- The Inactive phase duration is  $T_{\text{PERIOD}} - T_{\text{PROCESS}}$ , and its average current consumption is  $I_{\text{INACTIVE}}$ .

**Figure 6. 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.

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

The Process phase duration is proportional to the number of cycles to execute, which is a constant in our case:

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

The value  $\text{NOC}/(F_{\text{CLK}} \times T_{\text{PERIOD}})$  is usually called duty cycle.

This model does not take into account the consumption during the transition from Inactive to Process phase which is addressed in [Section 4.5: Impact of transition](#).

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

- The Run mode and clock frequency to be used during the Process phase.
- The 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:

- Run1 boost: Run mode available on STM32L4+ Series only, with a maximum of 150 DMIPS when run at 120 MHz.
- Run1: Run mode with Range 1 voltage, with a maximum of 100 DMIPS when run at 80 MHz.
- Run2: Run mode with Range 2 voltage, with a maximum of 32.5 DMIPS when run at 26 MHz.
- LP run: Low-power run mode, with a maximum of 2.5 DMIPS when run at 2 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 STM32L4/L4+ Series provides different low-power modes that can be used for the Inactive phase:

- Sleep or Low-power (LP) sleep
- Stop 0 (with main regulator on)
- Stop 1 (with regulator in low-power mode)
- Stop 2
- 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 or LP sleep: if the reactivity is the key parameter (in this case the wakeup time is only 6 system clock cycles).
- Stop 0: when the wakeup timing is critical (<1  $\mu$ s if the program is in RAM).
- Stop 1: if a lot of peripherals have to stay awake and the system has multiple sources of wakeup, including USB Suspend (in this case the application must tolerate a wakeup time from Flash memory of about 6.3  $\mu$ s).
- Stop 2: if few peripherals have to stay awake and can generate wakeup event as LPUART reception, LPTIM1 or I<sup>2</sup>C slave address recognition (in this case the application must tolerate a wakeup time from Flash memory of about 8.2  $\mu$ s).
- Standby: if no other peripherals than RTC need to stay awake, and up to 16/32/64 Kbytes of data retention is required (in this case the application must tolerate a wakeup time from Flash memory of about 14  $\mu$ s). Note that in this mode the LPUART and the I2C can wakeup the system quickly enough so that no loss of information occurs.
- 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 256  $\mu$ s).

*Note:* The wakeup timing depends on the code location (Flash memory or SRAM1), the system clock sources and the frequencies, refer to [4] for detailed conditions. The above wakeup from Stop figures correspond to MSI at 24 MHz with the code in Flash memory.

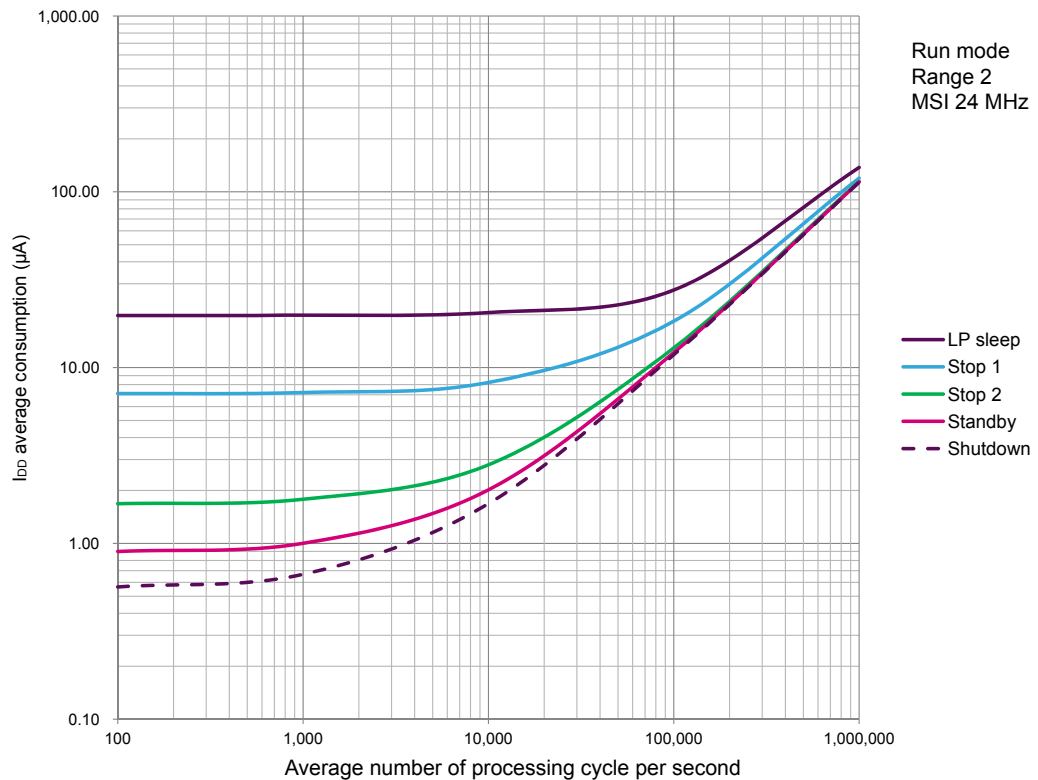
### 4.3 Sleep mode selection

The following numerical results are based on typical specifications extracted from the STM32L476xx datasheet [4] at 25 °C (typical value) with 32.768 kHz crystal LSE oscillator and RTC active all the time.

As shown in the previous section, the optimum power efficiency is obtained when using the Range 2 mode at 24 MHz (almost maximum frequency) using the MSI as source of clock.

The next figure gives the theoretical consumption, for different selections of low-power mode (Low-power sleep, Stop 1, Stop 2, Standby with RTC and SRAM2 retention, or Shutdown) with RTC in the Inactive phase, as a function of the average number of processing cycles needed in the Process phase assuming a one-second wakeup cycle period.

**Figure 7. STM32L476 influence of low-power mode on average current consumption**



The previous figure shows that, for applications requiring some data retention, the Standby low-power mode gives the best performance, whatever the number of cycles to be executed. It shows also that when the duty cycle increases (> 1 Mega cycles per second), Stop 2 gives almost the same results as Standby.

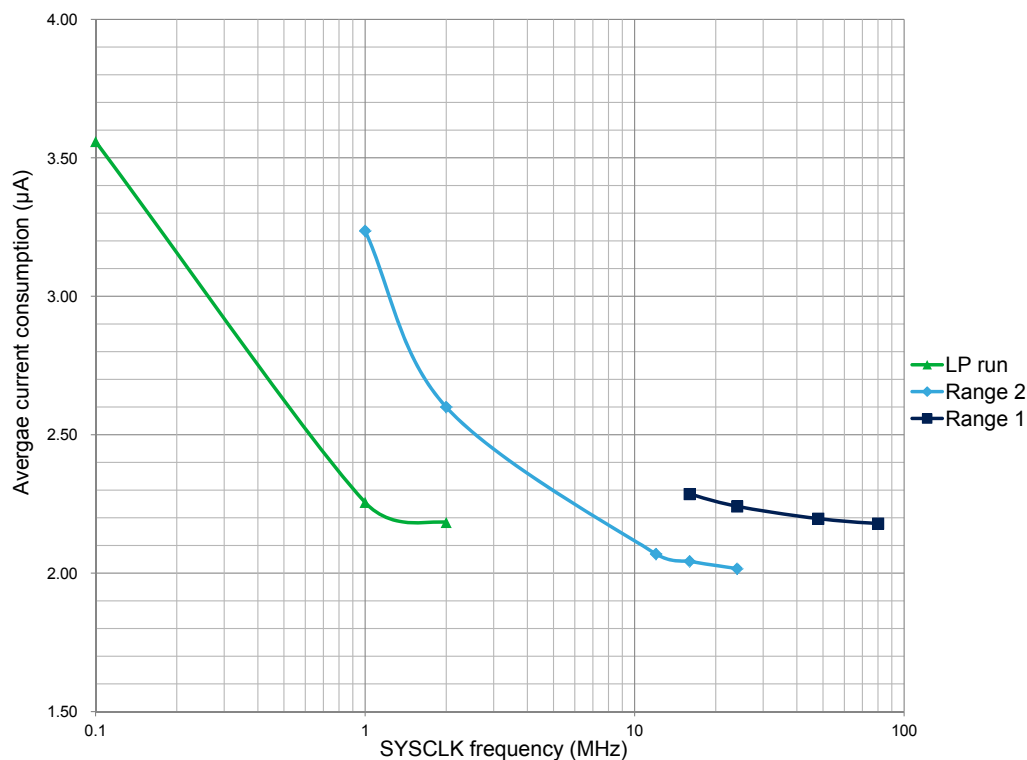
The Shutdown mode has been discarded (dotted line) because it does not present enough data retention capability for this type of application.

However the choice of low-power mode is not only dictated by the overall power consumption figure, but also by other wakeup considerations linked to the application. The previous figure also shows that the STM32L4/L4+ Series Stop 2 mode is very close to the Standby low-power mode while presenting a much more powerful and simpler setup for waking-up.

## 4.4 Run mode selection impact

When using Standby mode in the Inactive phase, the figure below shows the influence of the Run mode used during the Process phase on the average power consumption.

**Figure 8. STM32L476 influence of Run mode and frequency on average current consumption**



This figure corresponds to 10 K cycles of processing during the Process phase. The three different segments correspond to the LP run (left part of the graph), the Range 2 (middle part) and the Range 1 (right part).

The figure above confirms that the Range 2 at 26 MHz gives the best performance. However, the STM32L4/L4+ Series present very similar performances from 1 MHz up to 80 MHz, thanks to the three (or four) run modes (up to 120 MHz in STM32L4+ Series Boost mode). Simulations have been performed with other number of cycles (from 100 to 10 M) giving the same optimum point.

## 4.5 Impact of transition

This study would be incomplete if it did not take into account the energy spent during the transition phases (wake-up and deactivation phases).

Thanks to its built-in capless regulators, the STM32L4/L4+ Series needs a very low current to reload its internal nodes, once the device needs to wake up.

According to [4], at  $V_{DD} = 3.0$  V, the energy spent when leaving low-power mode is roughly:

- 32 nJ for transition from Stop 1 (with LPR) to Run mode Range 2.
- 50 nJ for a transition from Stop 2 to Run mode Range 2.
- 100 nJ for a transition from Standby to Run mode Range 1, with MSI at 4 MHz.
- 460 nJ for a transition from Shutdown to Run mode Range 1.

In order to integrate the energy consumed during transitions into the overall consumption, it is needed to take into account the system wakeup period.

The tables below provide an estimation of the overall current consumption for both Stop 2 and Standby with SRAM2 in retention modes according to the  $T_{PERIOD}$  and the number of average cycles in the Process phase.

**Table 2. Impact of wakeup energy on Standby**

-	T <sub>PERIOD</sub> <sup>(1)</sup>									Unit
	Cycles per second	1 ms	5 ms	10 ms	50 ms	100 ms	500 ms	1 s	5 s	
1000	-	-	4.40	1.68	1.34	1.07	1.03	1.01	1.00	μA
10k	36.02	8.82	5.42	2.70	2.36	2.08	2.05	2.02	2.02	
100k	46.17	18.97	15.57	12.85	12.51	12.24	12.21	12.18	12.17	
1M	148	121	117	114	114	114	114	114	114	
10M	1163	1135	1136	1133	1130	1129	1129	1129	1129	

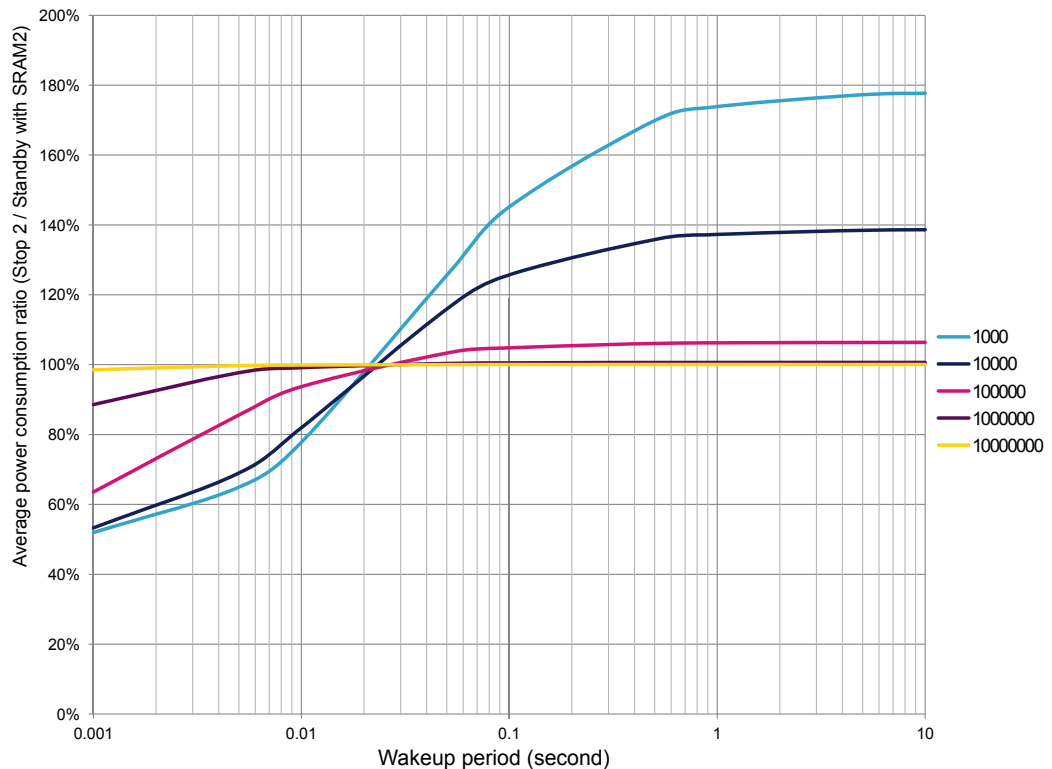
1. All consumption values are typical  $T_A = 25^\circ\text{C}$ ,  $V_{DD} = 3.0\text{ V}$  based on STM32L476 datasheet [4].

**Table 3. Impact of wakeup energy on Stop 2**

-	T <sub>PERIOD</sub> <sup>(1)</sup>									Unit
	Cycles per second	1 ms	5 ms	10 ms	50 ms	100 ms	500 ms	1 s	5 s	
1000	-	-	3.42	2.11	1.95	1.82	1.80	1.79	1.78	μA
10k	19.20	6.08	4.44	3.13	2.96	2.83	2.81	2.80	2.80	
100k	29.35	16.23	14.59	13.28	13.11	12.98	12.97	12.95	12.95	
1M	131	118	116	115	115	114	114	114	114	
10M	1146	1133	1131	1130	1130	1130	1130	1130	1130	

1. All consumption values are typical  $T_A = 25^\circ\text{C}$ ,  $V_{DD} = 3.0\text{ V}$  based on STM32L476 datasheet [4].

The next figure gives the ratio between the two tables presented above:

**Figure 9. STM32L476 comparison Stop 2 versus Standby with SRAM2**


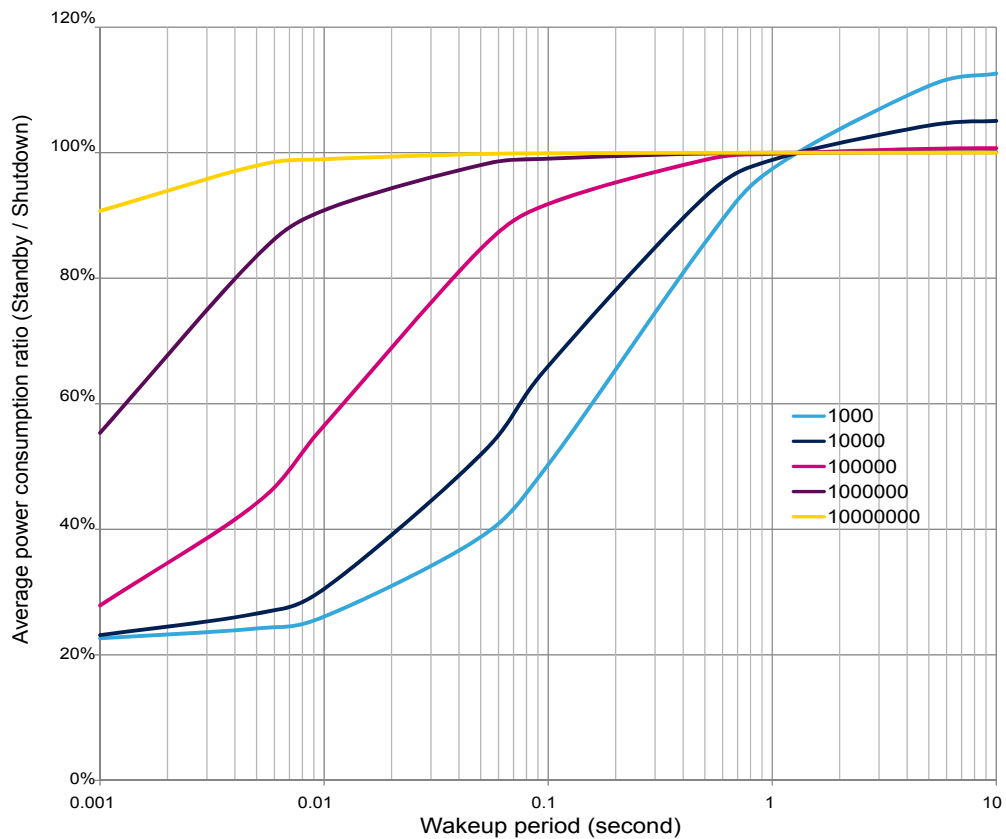
**Note:** *When the temperature increases the advantage provided by the Standby increases as well, for example the crossing point is 20 ms at 25 °C but become only 2 ms at 85 °C (and 70 ms at -20 °C).*

The horizontal axis corresponds to the system wakeup period, the vertical axis corresponds to the ratio between the average consumption of the Stop 2 case and that of the Standby with SRAM2 case. The different curves, correspond to the number of cycles for the Process phase.

If the wakeup period is below 20 ms Stop 2 is more interesting than Standby. If the number of cycles in the Process phase is high (for example 1 million and above) the interest of Standby decreases, as the consumption is then mainly driven by the Run mode used during the Process phase.

A similar computation has been performed for Shutdown versus Standby, the crossing point is in the range of 2 seconds.

**Figure 10. STM32L476 comparison Standby without SRAM2 retention versus Shutdown**



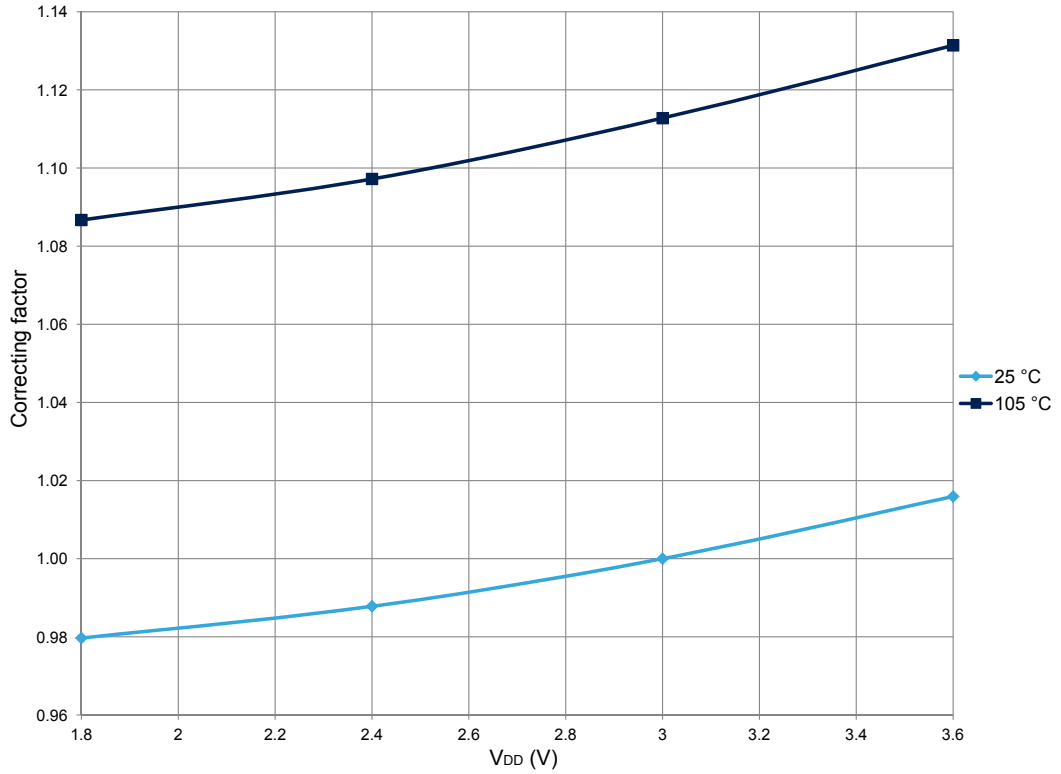
**Note:** *In this theoretical study the number of instructions to reload the context is assumed to be taken into account in the Process phase. In a real implementation it requires more instructions to restart from Shutdown and Standby than Stop modes.*

## 4.6 Impact of voltage and temperature

The previous data was provided for  $V_{DD} = 3.0\text{ V}$  and for a temperature of  $25\text{ }^{\circ}\text{C}$ .

The Process phase current consumption varies very little with the voltage and the temperature. The next figure gives the correction factor to apply for the Range 2 value at 24 MHz for varying  $V_{DD}$  voltage and temperature:

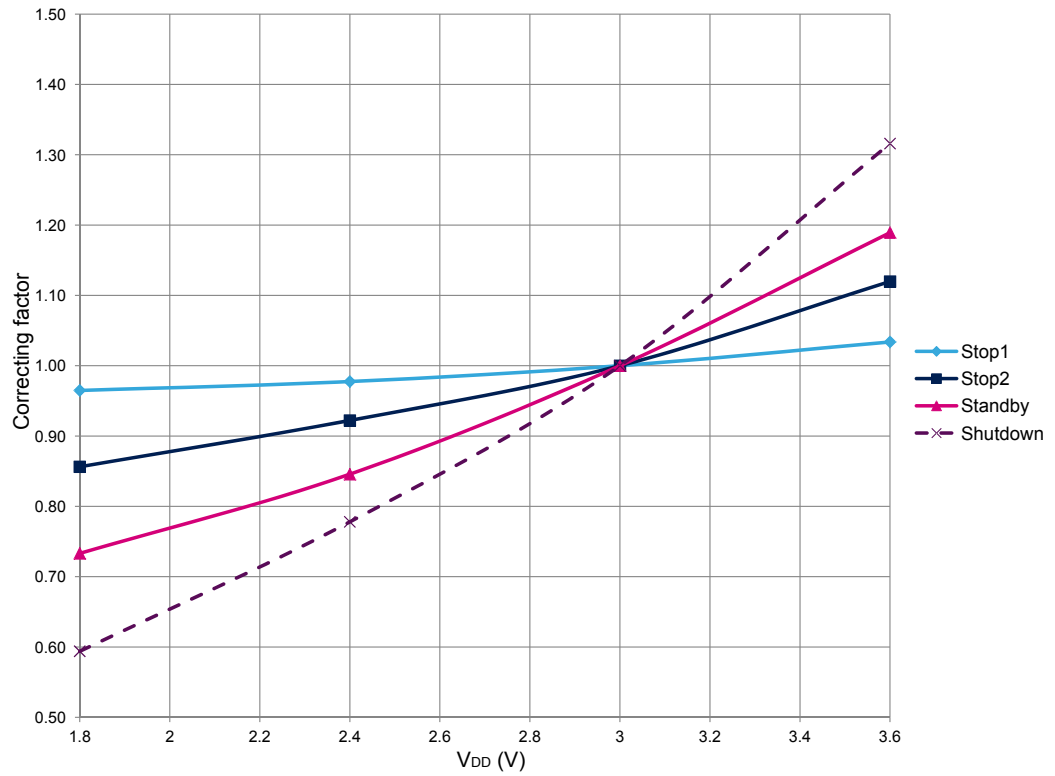
**Figure 11. STM32L476 Run mode correction with voltage and temperature**



The low-power modes have a stronger dependency on the voltage and temperature.

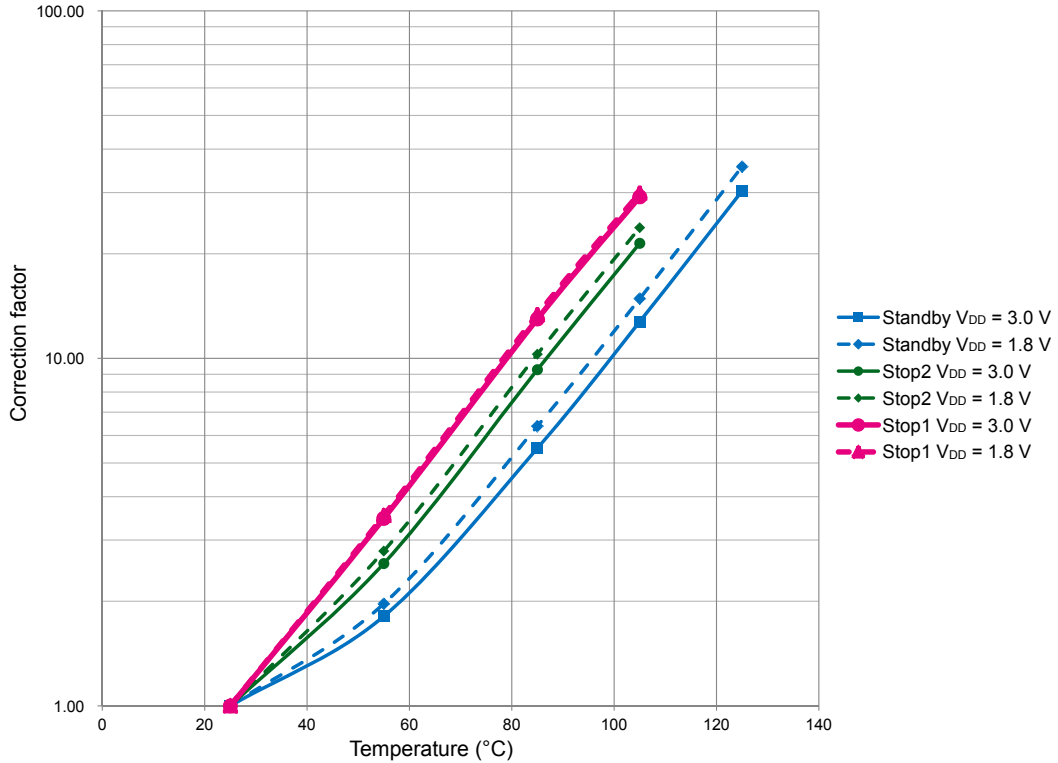
The figure below shows the correction factor to apply to the consumption for different low-power modes and supply voltage:

**Figure 12. STM32L476 low-power mode correction with  $V_{DD}$  voltage**



The figure hereafter gives the correction factor to apply to the consumption for different low-power modes and temperatures, for 1.8 V and 3.0 V supply:

**Figure 13. STM32L476 low-power mode correction with temperature**



Above 45 °C, a good approximation is to consider that the low-power mode consumption doubles every 17 °C. A similar theoretical optimization analysis has been performed over the whole range of temperature. The conclusion remains unchanged.

Reducing the voltage does not decrease significantly the current consumption in Process phase (2 % gain for 1.8 V compared to 3.0 V). Nevertheless voltage reduction should be considered for following reasons:

- The Inactive phase current is significantly lower (almost 30 % for Standby).
- As the power consumed by the system is the product of the current by the voltage, the power consumption is at least 40 % lower (1.8 V versus 3.0 V).

In the example above, the consumption in the Process phase is 60 % of the overall consumption. If the voltage is reduced from 3.0 V to 1.8 V:

- The current consumption is reduced by 12 %.
- The overall power consumption is reduced by 53 %.

## 4.7 Getting more accurate simulations

The approximation that the number of cycles (NOC) required to execute a task is constant with the CPU frequency is not exactly true. It tends to decrease when the CPU frequency decreases, because:

- The number of wait states required to access the Flash memory decreases when the frequency decreases.
- The number of cycles spent by the processor in waiting loops decreases.

The number of processing cycles depends also on the low-power mode selection. It takes more instruction cycles to restore the context when using Standby mode with SRAM2 retention compared to other low-power modes.

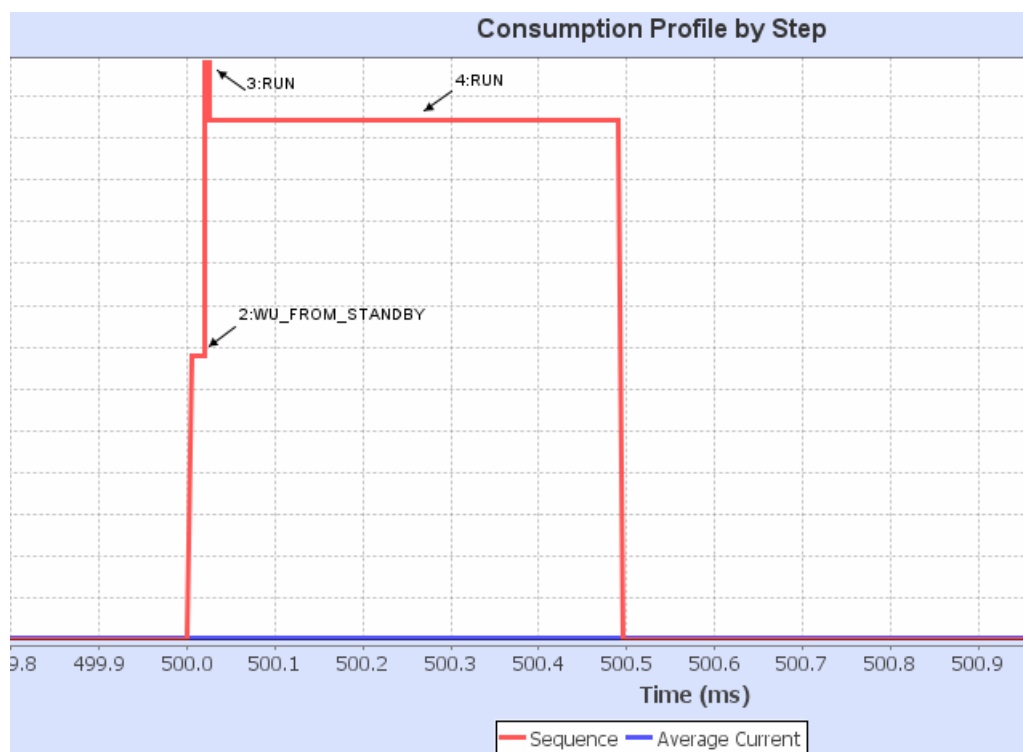
Finally the consumption required for the configuration of the peripherals during the active phase has also to be taken into consideration.

To help the customer to fine tune his application and make his own choices STMicroelectronics provides a Power Consumption Calculator (PCC) module within the **STM32CubeMX** that is available for free download from <http://www.st.com/stm32cube>.

PCC allows the average and peak power for a complex processing sequence to be computed accurately and helps with the selection of the appropriate modes according the different processing phases. Consumption of peripherals can also be added to the simulation.

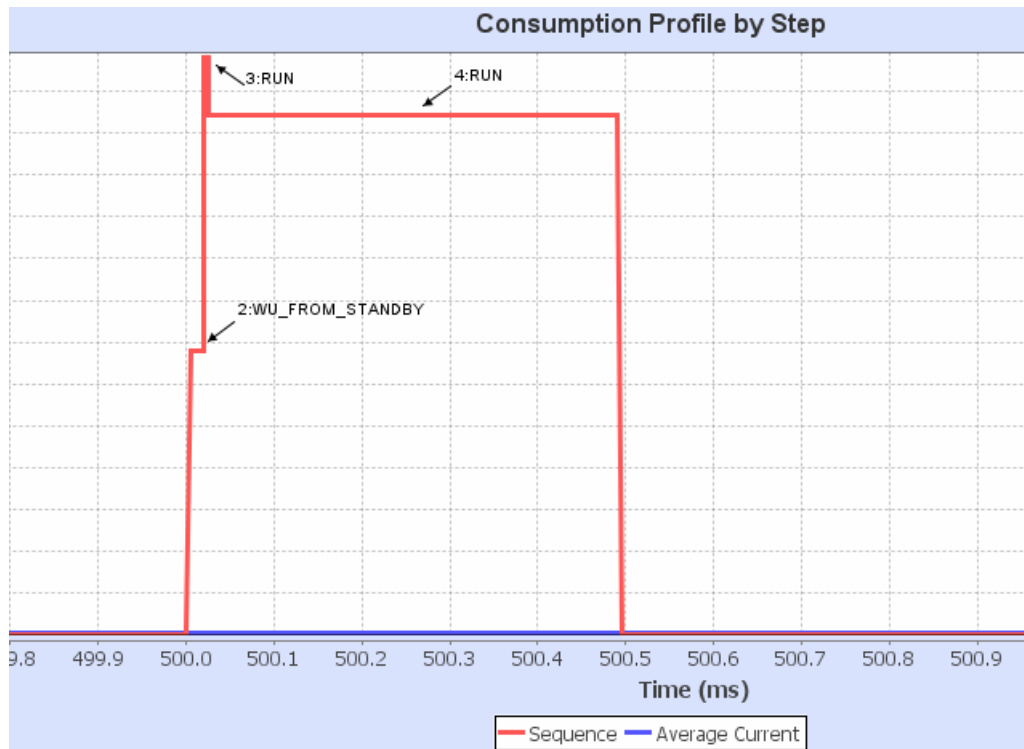
The following snapshots of the PCC tool screen correspond to a Process phase of 10 K instructions, similar to the ULPBench™ sequence execution using Standby and Range 2 at 24 MHz. It allows to take into account the transition from Standby to Range 1 then Range 2.

**Figure 14. Zoom on ULPBench™ sequence showing transitions**



The PCC tool interface provides:

- A table showing the configuration parameters for each processing step
- A chart showing the contribution of each phase to the overall power consumption.

**Figure 15. PCC simulation of ULPBench™ sequence**


The PCC tool allows to simulate the consumption of the STM32L4/L4+ Series device using an external SMPS. The user can enter its SMPS main characteristics such as:

- Quiescent current
- Efficiency
- Off current
- Output voltage

## 5 ULPMark-CP use case optimization

---

### 5.1 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.

### 5.2 Process phase optimization

Since the computation load is quite small (in the range of 10 Kcycles every second), any of the three Run modes can be used to execute the Process phase within the one second time limit, down to a frequency of 10 kHz. In the same way, there is no particular limit on the maximum peak current that can be drawn and hence the maximum clock frequency that can be used.

Since there are no real-time aspect constraints (minimum latency to serve the ISR), it is not mandatory to select a very fast clock or wakeup time.

The user can therefore use the power optimum point corresponding to the Range 2 at 24 MHz, using the internal MSI. When using the MSI at 24 MHz nominal frequency, it consumes less than any other solution using the PLL that would allow to reach the 26 MHz limit of the Range 2.

*Note:* In the Range 2 mode, the efficiency figure is expected to be almost the same at 24 MHz, 16 MHz or even 12 MHz.

### 5.3 Inactive phase optimization

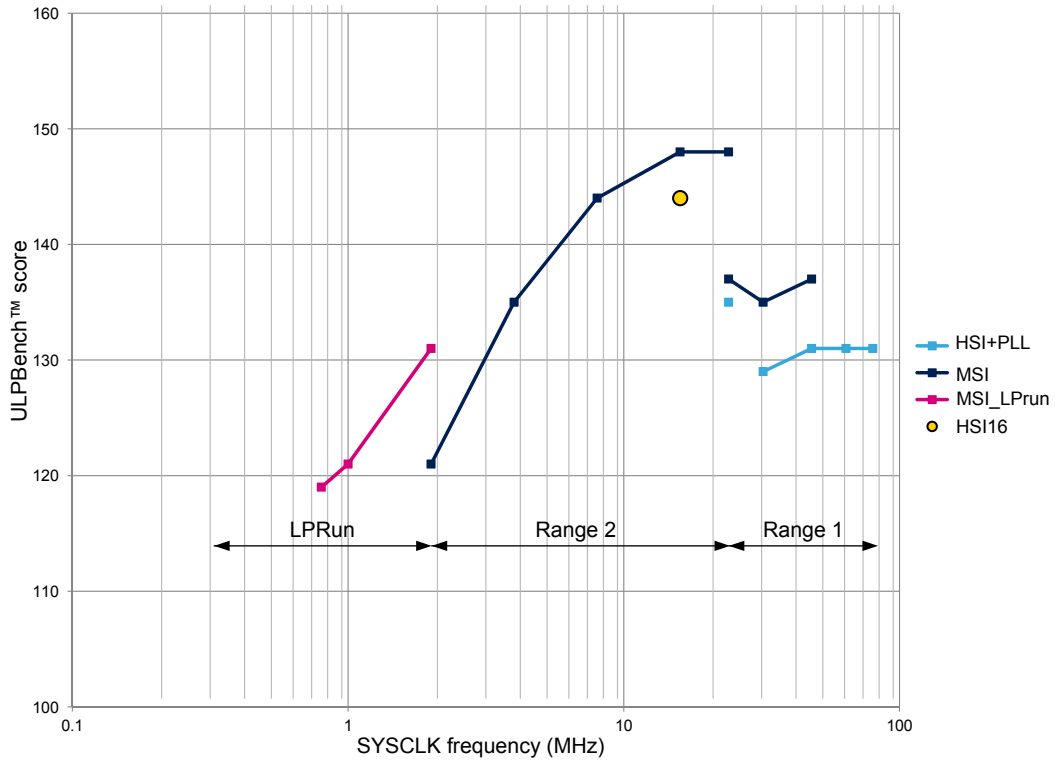
The ULPMark-CP benchmark, requires to keep the data variables unchanged from one run to the next. Therefore the Shutdown mode cannot be used because it does not provide enough data retention capability for this application.

The most power efficient solution uses the Standby mode with RTC and retention in SRAM2.

## 5.4 STM32L476xx measurements results

The next figure gives the ULPMark-CP score measurements depending on the Run mode and the system clock frequency used during the Process phase. It uses the Standby mode with SRAM2 retention and RTC for the Inactive phase.

**Figure 16. ULPMark-CP STM32L476 measurements versus Run mode and frequency**



The results are provided with different configurations of the clocks:

- For frequencies below 48 MHz, the MSI is used. This is because the MSI is automatically started when waking-up from Standby.
- For frequencies above 48 MHz, the HSI (at 16 MHz) and the PLL are used. Since the user needs to use the PLL and wait until it is locked, it is more interesting (better accuracy) to use the HSI at 16 MHz. In this case the MSI is stopped once the PLL has started.
- A special point using the HSI at 16 MHz is provided, without any PLL.

*Note:* For frequencies below or equal to 2 MHz, the LP run mode is used while for the range 2 MHz to 24 MHz, the Range 2 is used and for frequencies above 24 MHz the Range 1 is used.

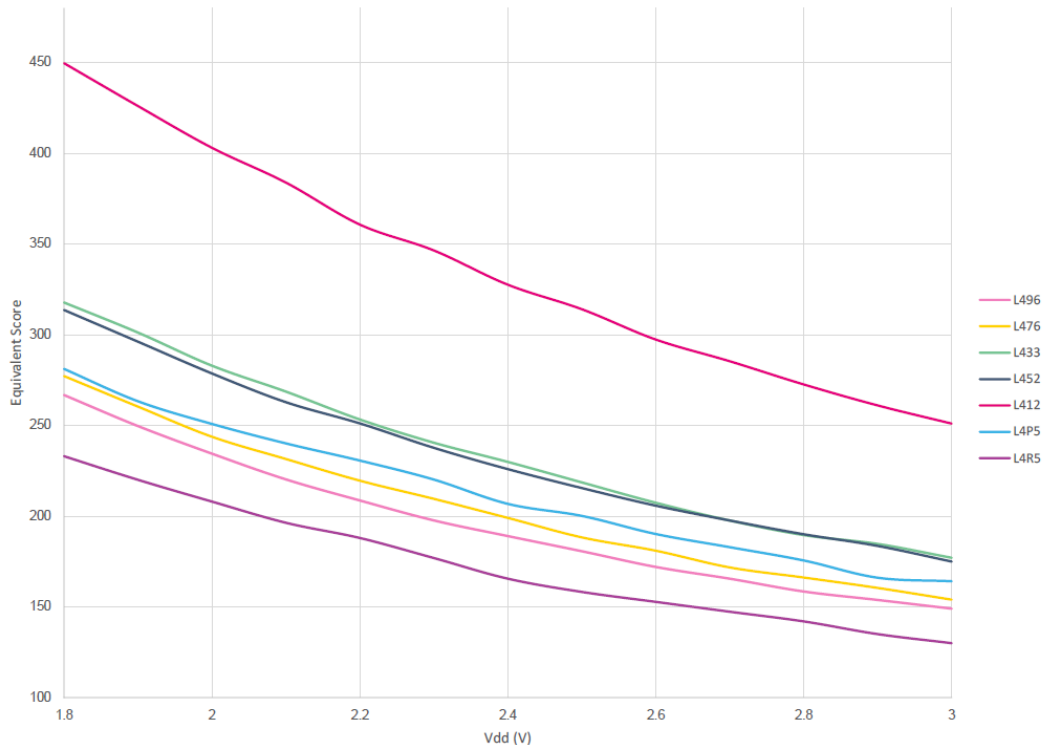
As already seen in the theoretical study, the Range 2 at 24 MHz and 16 MHz, using the MSI provides the best results.

## 5.5 Performance evolution with voltage range

EEMBC has defined base test defaults on 3.0 V, however this voltage is not very representative of battery supplied low-power applications. As explained in the theoretical study, and unlike systems with DC/DC converters, the gain in power when moving to a lower voltage is very significant compared to staying at 3.0 V. 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 17. Equivalent score as a function of supply voltage**



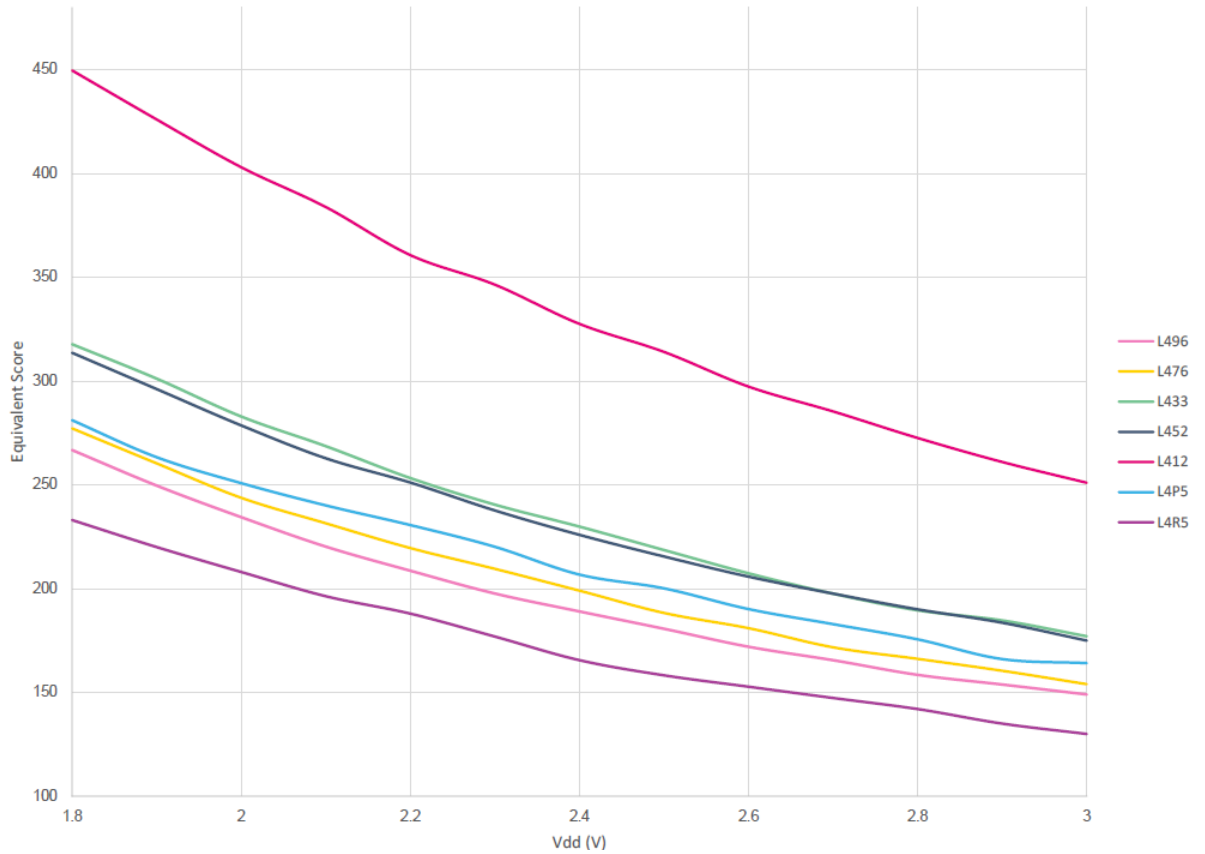
The above curves correspond to the Range 2 at 24 MHz MSI mode, it clearly shows the advantages of a voltage reduction.

## 5.6 Performance evolution with the temperature

At a temperature increase, the chip leakage becomes the most dominant part of the overall consumption.

The next figure gives the variation of the score at two different temperatures for the STM32L4/L4+ Series devices.

**Figure 18. ULPMark-CP score versus temperature**

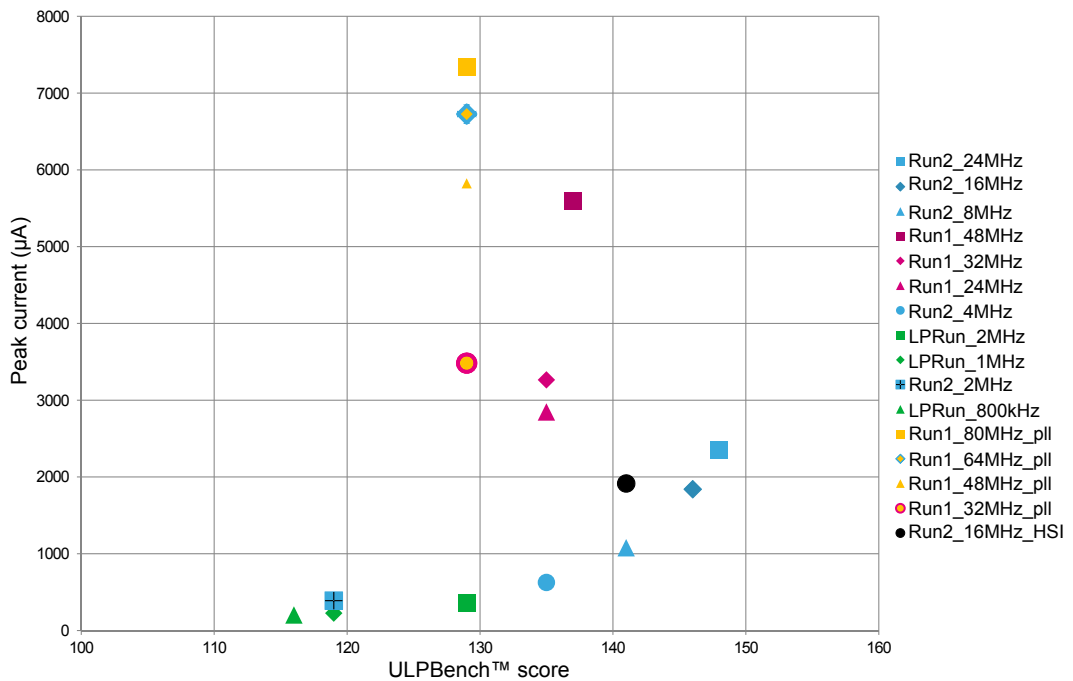


## 5.7 Peak current performance

This type of application, with very low average consumption, could present a very high peak current. The higher the frequency, the higher the current peak. It can become the predominant parameter for the selection of the frequency and the operating mode used in the Process phase.

The figure below shows the measurements of the peak current as a function of the ULPMark-CP score.

**Figure 19. ULPMark-CP score versus STM32L476 peak current measurement**



The above measurements have been performed on Nucleo-L476RG board assuming an impedance source of 200  $\Omega$  for frequencies below 48 MHz and 100  $\Omega$  for frequencies above. If the source impedance is lower, the peak current increases up to the corresponding Run mode maximum.

**Note:**

*The peak current drawn from the battery system depends a lot on the battery internal resistance as well as the decoupling capacitors placed on the PCB. Aged batteries or extreme temperature conditions could significantly increase this resistance, making the peak factor a decisive choice for the selection of the optimum mode.*

An easy way to reduce the peak current is to increase the input decoupling capacitance. On the Nucleo-L476RG board a maximum of 4.7  $\mu\text{F}$  of tank capacitor can be used.

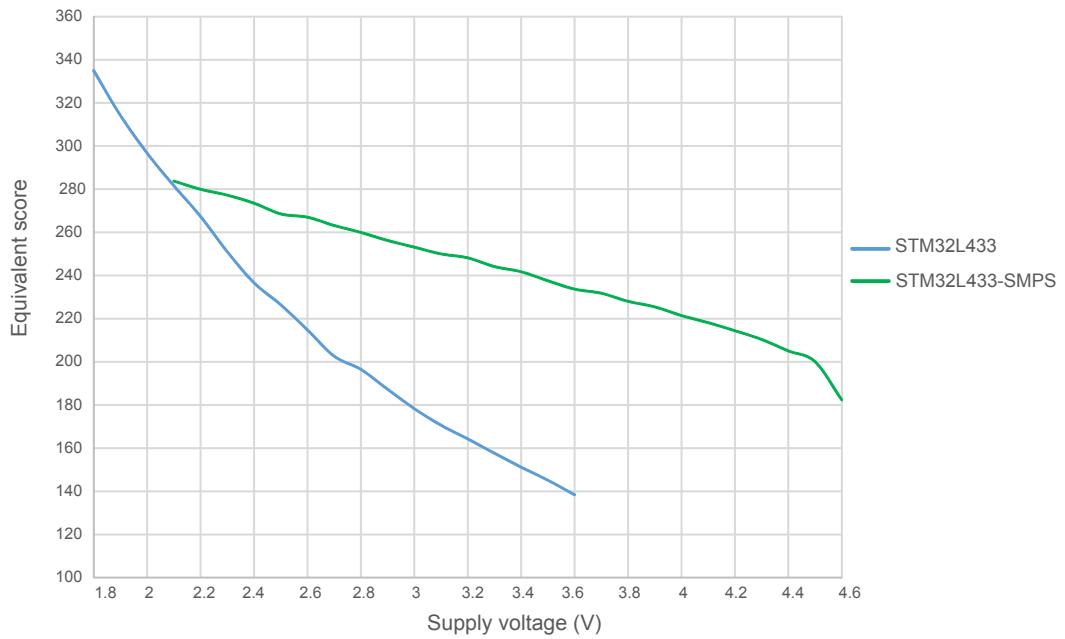
## 5.8 Using STM32L4/L4+ Series with external SMPS support

For detailed reference how to use an external SMPS, refer to AN4978 [5].

The external SMPS capability allows to supply the STM32L4/L4+ digital logic with an efficient conversion factor. The gain efficiency, versus non external SMPS solution, increases when the supply voltage increases. This solution is well suited for higher operating voltage (>2.5 V) and allows to overcome the STM32L4/L4+ maximum voltage operation (3.6 V) while keeping the low-power performance.

The following figure plots the equivalent ULPMark-CP score obtained with the same device using or not the external SMPS capability:

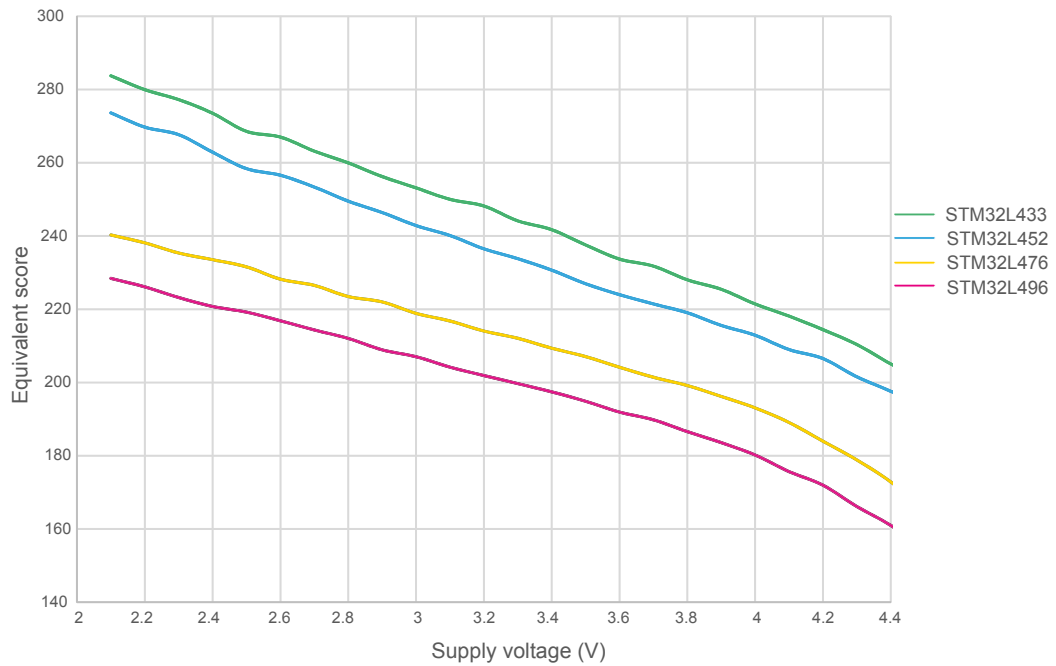
**Figure 20. Equivalent score with and without external SMPS**



It clearly shows the advantages of using an external SMPS when the supply voltage is high.

All of the STM32L4/L4+ Series devices have the capability to support an external SMPS, the following figure plots the obtained performance for each of them.

**Figure 21. STM32L4 Series devices comparison using external SMPS**



Note that the STM32L476 and STM32L496 measurements have been performed on a Nucleo144 board, while the STM32L433 and STM32L452 have been performed on a Nucleo 64 board. Each board having the same SMPS devices allowing the operation down to 2.1 V and also a dedicated SMPS to produce the 1.8 V needed for the  $V_{DD}$  and IO.

## 6 ULPMark-PP use-case optimization

---

### 6.1 ULPMark-PP benchmark description

The peripheral profile has been added to the ULPMark™ suite to reflect on extensive use of embedded peripherals in the typical MCU application. The peripheral profile includes ubiquitous MCU peripherals: ADC, timers and SPI. RTC is again used to wake the core every second and perform scheduled tasks.

Detailed description is available on the EEMBC website.

### 6.2 Optimization for highest efficiency with peripherals

Decisive factor in this test is the degree of autonomous task execution by the peripheral. If the task can be executed with processor core being in a low-power mode, it is a significant advantage. This requires using a DMA. Configuring the interrupts to wake the CPU core only when the peripheral needs to be reconfigured is also important. For example instead of waking the CPU after each byte received or transmitted by the SPI communication interface, the CPU only handles the beginning and the end, while being left in Stop mode during communication.

The DMA is used in similar way for a batch conversion using ADC.

For the PWM tasks there is an array of different available timers. Specifically for the ULPMark-PP, the LPTimer is most suited. See AN4013 [8] for a high level overview of timer features. Remember that the RTC (alarm) or SysTick can perform some simple timing jobs, for which a timer would otherwise be used.

Rules and principles described in the ULPMark-CP chapter apply generally also for the ULPMark-PP case.

## 7 Conclusion

The STM32L4/L4+ Series offers 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 modes and low-power modes, according to the characteristics and constraints of the end-user application.

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 low-power mode for the Inactive phase associated with the Range 2 mode at 24 MHz (from the MSI) for the Process phase. Other frequencies in the range 1 to 80 MHz give also very good results. Power efficiency of STM32L4+ Series boost mode is again slightly lower, but still a viable option, for example for short burst-data processing.

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 provides better consumption.
- If the wakeup period is shorter, an implementation with Stop 2 provides better results.
- If the temperature is high, the Standby mode should be preferred.

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

For STM32L4/L4+ Series, if the processing can accommodate a frequency below 24 MHz, the optimum points are (in decreasing power efficiency order):

- Standby mode: if wakeup transition time can be longer than 20  $\mu$ s and retention area can be as small as 16/32/64 Kbytes.
- Stop 2: if wakeup transition time can be longer than 8  $\mu$ s and the application needs more than 16/32/64 Kbytes of retention.

In the second case, the consumption penalty is less than 1.8 times the first case.

In any case, decreasing the  $V_{DD}$  voltage improves the power consumption.

The usage of an external SMPS, when the  $V_{DD}$  voltage is higher than 2 V, can significantly improve the power performance of the overall application.

Ultimately 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 256  $\mu$ s.

To help the customer to fine tune his application and make his own choice STMicroelectronics provides a power consumption calculator (PCC) module within the STM32CubeMX that is available for free download from <http://www.st.com/stm32cube>.

*Note: This document is using ULPBench™ results that are updated regularly, refer to the official score stored at <http://www.eembc.org> (search for ULPBench™).*

## Revision history

Table 4. Document revision history

Date	Version	Changes
16-Sep-2015	1	Initial release.
30-May-2017	2	<p>Updated cover adding related documents.</p> <p>Updated <i>Section 2.1.1: Low-power run and Low-power sleep modes</i>.</p> <p>Added <i>Table 1: SRAM2 retained content</i>.</p> <p>Updated <i>Section 3.2: INACTIVE phase adding stop 0 mode</i>.</p> <p>Updated <i>Figure 14: ULPBench™ STM32L476 measurements versus Run mode and frequency</i>.</p> <p>Updated <i>Figure 15: Equivalent score as a function of supply voltage</i>.</p> <p>Added <i>Section 4.6: Performance evolution with the temperature and Figure 16: ULPBench™ score versus temperature</i>.</p> <p>Added STM32L49xxx, STM32L45xxx, STM32L43xxx performance.</p> <p>Added <i>Section 4.8: Using STM32L4 Series with external SMPS support</i>.</p>
16-Dec-2019	3	<p>Added STM32L4P5xx, STM32L4Q5xx and STM32L412 products, hence updated:</p> <ul style="list-style-type: none"> <li>• <i>Table 1: SRAM2 retained content</i></li> <li>• <i>Figure 15: Equivalent score as a function of supply voltage</i></li> <li>• <i>Figure 16: ULPBench™ score versus temperature</i>.</li> </ul>
18-Jan-2021	4	Document's scope updated to include STM32L4+ Series
28-Jan-2026	5	Updated document title.

## Contents

<b>1</b>	<b>General information</b>	<b>2</b>
<b>2</b>	<b>Low-power application profile</b>	<b>3</b>
2.1	Key parameters to consider	3
2.2	ULPMark™ description	4
<b>3</b>	<b>STM32L4/L4+ Series low-power feature</b>	<b>5</b>
3.1	Numerous low-power modes	5
3.1.1	Low-power run and Low-power sleep modes	5
3.1.2	Stop mode	5
3.1.3	Standby mode	5
3.1.4	Shutdown mode	6
3.2	Run mode power consumption	7
<b>4</b>	<b>Low-power mode selection methodology</b>	<b>9</b>
4.1	Process phase	10
4.2	Inactive phase	10
4.3	Sleep mode selection	11
4.4	Run mode selection impact	12
4.5	Impact of transition	12
4.6	Impact of voltage and temperature	15
4.7	Getting more accurate simulations	18
<b>5</b>	<b>ULPMark-CP use case optimization</b>	<b>20</b>
5.1	Use case constraints	20
5.2	Process phase optimization	20
5.3	Inactive phase optimization	20
5.4	STM32L476xx measurements results	21
5.5	Performance evolution with voltage range	22
5.6	Performance evolution with the temperature	23
5.7	Peak current performance	24
5.8	Using STM32L4/L4+ Series with external SMPS support	25
<b>6</b>	<b>ULPMark-PP use-case optimization</b>	<b>27</b>
6.1	ULPMark-PP benchmark description	27
6.2	Optimization for highest efficiency with peripherals	27
<b>7</b>	<b>Conclusion</b>	<b>28</b>
	<b>Revision history</b>	<b>29</b>
	<b>List of tables</b>	<b>32</b>

---

List of figures.....33

## List of tables

<b>Table 1.</b>	SRAM2 retained content . . . . .	6
<b>Table 2.</b>	Impact of wakeup energy on Standby . . . . .	13
<b>Table 3.</b>	Impact of wakeup energy on Stop 2 . . . . .	13
<b>Table 4.</b>	Document revision history . . . . .	29

## List of figures

<b>Figure 1.</b>	Application profile . . . . .	3
<b>Figure 2.</b>	STM32L476 Run mode current consumption . . . . .	7
<b>Figure 3.</b>	STM32L4R5 Run mode current consumption . . . . .	7
<b>Figure 4.</b>	STM32L476 Run mode power efficiency . . . . .	8
<b>Figure 5.</b>	STM32L4R5 Run mode power efficiency . . . . .	8
<b>Figure 6.</b>	Application sequence and parameters . . . . .	9
<b>Figure 7.</b>	STM32L476 influence of low-power mode on average current consumption . . . . .	11
<b>Figure 8.</b>	STM32L476 influence of Run mode and frequency on average current consumption . . . . .	12
<b>Figure 9.</b>	STM32L476 comparison Stop 2 versus Standby with SRAM2 . . . . .	13
<b>Figure 10.</b>	STM32L476 comparison Standby without SRAM2 retention versus Shutdown . . . . .	14
<b>Figure 11.</b>	STM32L476 Run mode correction with voltage and temperature . . . . .	15
<b>Figure 12.</b>	STM32L476 low-power mode correction with $V_{DD}$ voltage . . . . .	16
<b>Figure 13.</b>	STM32L476 low-power mode correction with temperature . . . . .	17
<b>Figure 14.</b>	Zoom on ULPBench™ sequence showing transitions . . . . .	18
<b>Figure 15.</b>	PCC simulation of ULPBench™ sequence . . . . .	19
<b>Figure 16.</b>	ULPMark-CP STM32L476 measurements versus Run mode and frequency . . . . .	21
<b>Figure 17.</b>	Equivalent score as a function of supply voltage . . . . .	22
<b>Figure 18.</b>	ULPMark-CP score versus temperature . . . . .	23
<b>Figure 19.</b>	ULPMark-CP score versus STM32L476 peak current measurement . . . . .	24
<b>Figure 20.</b>	Equivalent score with and without external SMPS . . . . .	25
<b>Figure 21.</b>	STM32L4 Series devices comparison using external SMPS . . . . .	26

**IMPORTANT NOTICE – 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.

In the event of any conflict between the provisions of this document and the provisions of any contractual arrangement in force between the purchasers and ST, the provisions of such contractual arrangement shall prevail.

The 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 acknowledgment.

The 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 the 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.

If the purchasers identify an ST product that meets their functional and performance requirements but that is not designated for the purchasers’ market segment, the purchasers shall contact ST for more information.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to [www.st.com/trademarks](http://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.

© 2026 STMicroelectronics – All rights reserved