How to wake up an STM32 microcontroller from low-power mode with the USART or the LPUART

Introduction

The universal synchronous/asynchronous receiver transmitter (USART) and the low-power universal asynchronous receive transmitter (LPUART) feature advanced low-power mode functions. These functions allow receiving data properly even when an STM32 microcontroller (MCU) of the series listed below is in low-power mode, and the APB clock is disabled.

Table 1. Applicable products

Type	Products
Microcontroller	• STM32C0 series • STM32F0 series, STM32F3 series • STM32G0 series, STM32G4 series • STM32H5 series, STM32H7 series • STM32L0 series, STM32L4 series, STM32L4+ series, STM32L5 series • STM32N6 series • STM32U0 series, STM32U3 series, STM32U5 series • STM32WB series, STM32WB0 series, STM32WBA series

1 General information

This application note applies to the STM32 Series 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.

2 STM32 low-power modes

The table below summarizes the low-power modes, from which the STM32 MCU can be woken up with the USART or the LPUART.

Table 2. STM32 low-power modes

STM32 series	Mode from which the STM32 can be woken up by	
	USART	LPUART
STM32C0	Stop mode	N/A
STM32F0, STM32F3	Stop mode (with the main regulator in Run mode or in low-power mode)	
STM32G0, STM32G4, STM32WBA	Stop 0 and Stop 1 modes	
STM32H5, STM32H7, STM32N6	Stop mode	
STM32L0	Stop mode (with the main regulator in Run mode or in low-power mode, range 1/2/3)	
STM32L4, STM32L4+, STM32L5, STM32U5, STM32WB, STM32U0, STM32U3	Stop 0 and Stop 1 modes	Stop 0, Stop 1, and Stop 2 modes
STM32WB0	DEEPSTOP mode	

3 USART/LPUART wake-up features

3.1 Dual-clock domain

The USART or LPUART is able to wake up the MCU from a low-power mode only when the peripheral supports the dual-clock domain. The USART or LPUART can be clocked by a clock independent from the APB clock. This clock can be either the HSI, MSI, or LSE clock depending on the device.

The USART or LPUART can then receive data even if its clock is disabled and the MCU is in low-power mode.

3.2 USART/LPUART wake-up sources

Some transmit, receive, and error interrupts are used to wake up from low power modes. For the full list of these interrupts, refer to the USART and LPUART interrupts section in the product reference manual.

To enable the USART or LPUART to wake up the MCU from low-power mode, the UESM bit must be set in USART/LPUART_CR1 before entering a low-power mode.

4 How the USART/PLUART wakes up the STM32

4.1 Wake-up from a low-power mode when the internal oscillator is off

If the STM32 MCU is in a low-power mode, and the internal oscillator clock used as USART/LPUART kernel clock is switched off, when a falling edge on the USART/LPUART receive line is detected, the USART or LPUART requests the kernel clock to be switched on again. The internal oscillator clock is then used for frame reception:

- If the wake-up event is verified, the STM32 MCU wakes up from low-power mode, and the data reception goes on normally.
- If the wake-up event is not verified, the internal oscillator clock is switched off again, the STM32 MCU is not woken up and remains in low-power mode, and the kernel clock request is released.

Table 3. STM32 internal oscillators

STM32 series	Internal clock supported			LPBAM supported(1)
	MSI	HSI	CSI	
STM32C0, STM32F0, STM32F3, STM32G0, STM32G4, STM32L4, STM32L4+, STM32L5, STM32WBA	No	Yes	No	No
STM32WB0				
STM32H5, STM32H7	Yes		Yes	
STM32L0, STM32N6, STM32WB, STM32U0, STM32U3			No	
STM32U5				Yes

1. *LPBAM means low-power background autonomous mode.*

The figures below show an example of a wake-up event programmed to an “address match detection" (see Section 5.2 for details on $t_{WUUSART}$ and $t_{WULPUART}$).

Figure 1. Wake-up event verified (wake-up event = address match)

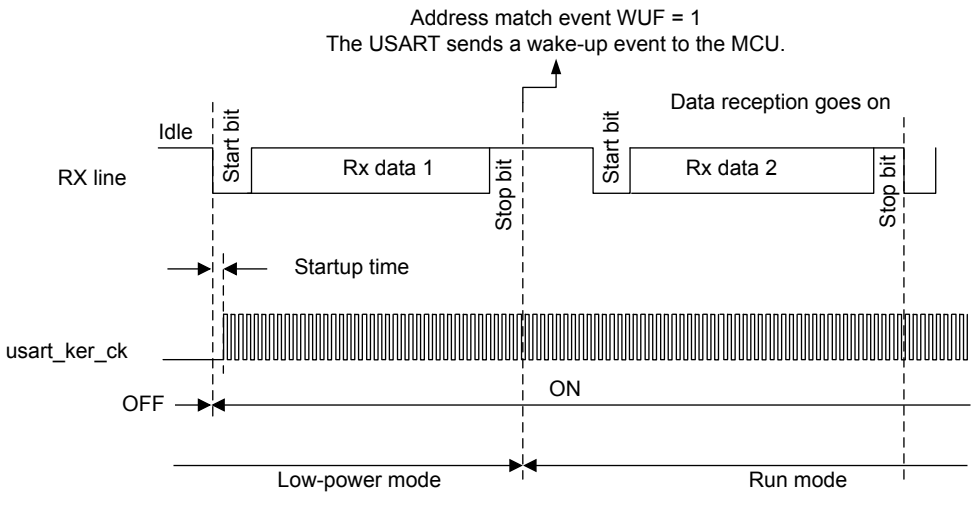
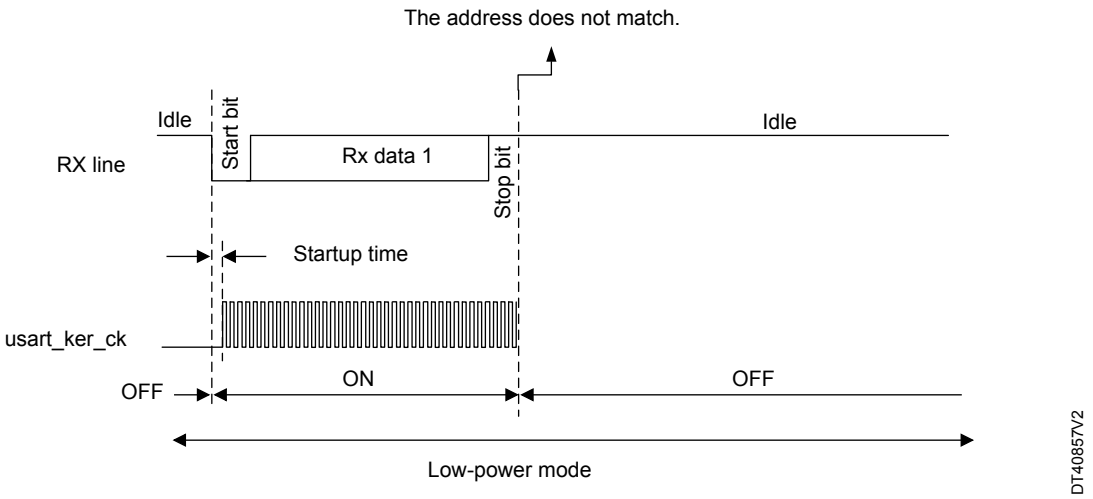


Figure 2. Wake-up event not verified (wake-up event = address match)



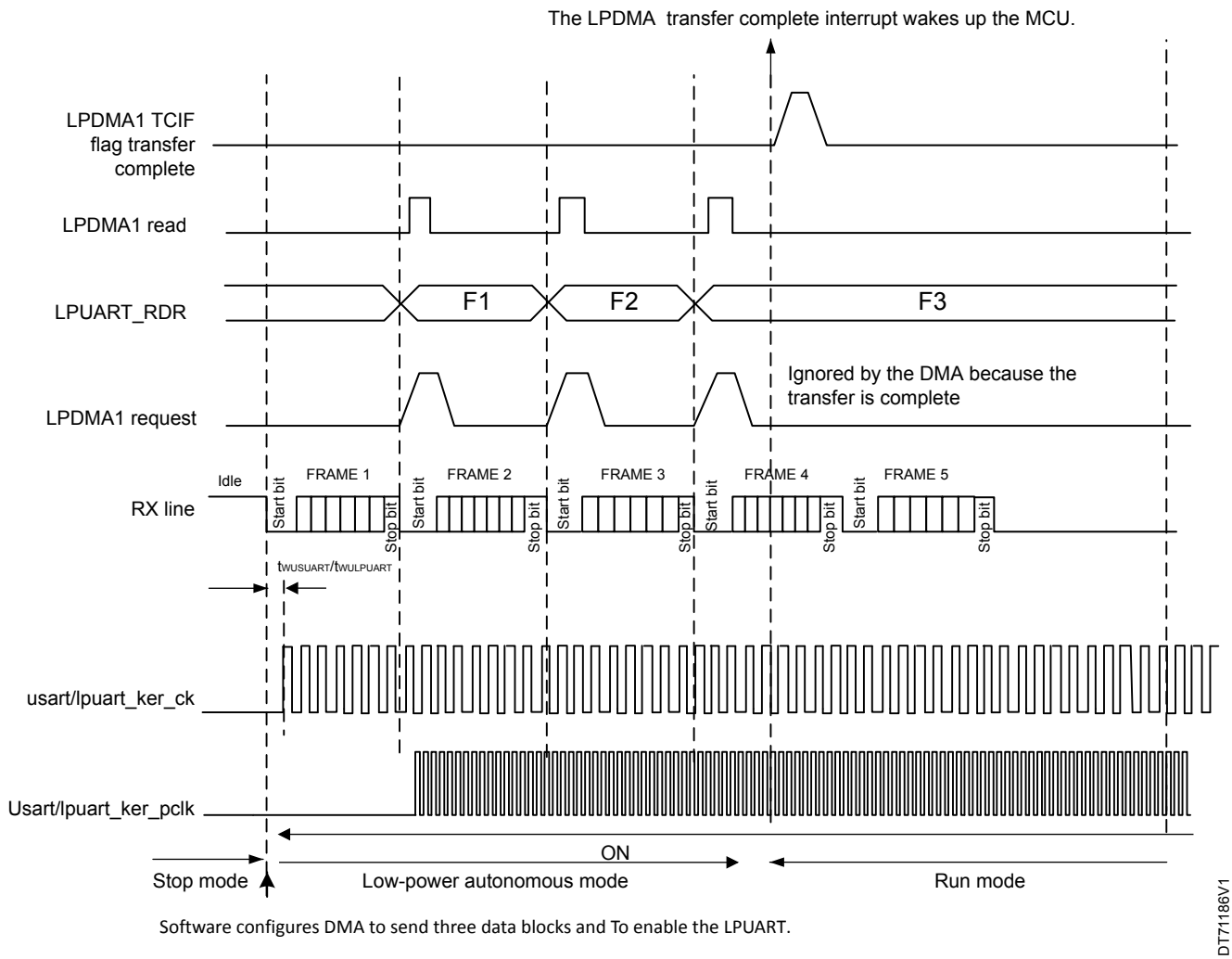
4.2 Wake-up from a low-power mode in SmartRun domain (SRD)

The STM32 MCUs that support the LPBAM, which allows the devices to be functional and autonomous in Stop 0, Stop 1, and Stop 2 modes (with no software running).

The USART/LPUART is also functional in Stop mode thanks to the autonomous mode. The APB clock is requested by the peripheral each time the USART/LPUART status needs to be updated. Once the USART or LPUART receives the APB clock, it generates either an interrupt or a DMA request, depending on the peripheral configuration.

If an interrupt is generated, the device wakes up from Stop mode. If no interrupt is generated, the device remains in Stop mode, but the kernel and APB clocks are still available for the USART/LPUART. All the autonomous peripherals are enabled in the RCC (reset and clock controller). If DMA requests are enabled, data are directly transferred to/from the SRAM thanks to the DMA, while the STM32 MCU remains in Stop mode.

Figure 3. Wake-up interrupt from SmartRun domain



Software configures DMA to send three data blocks and To enable the LPUART.

5 USART/LPUART baudrate for a correct wake-up

The maximum USART or LPUART baudrate to correctly wake up the STM32 MCU from a low-power mode depends on the USART/LPUART kernel clock status (switched on or off).

5.1 USART/LPUART kernel clock switched on in low-power mode

In this case, there is no constraint on the maximum baudrate to wake up the MCU from a low-power mode. It is the same as in Run mode.

5.1.1 Internal oscillator clock used as USART/LPUART clock source

For some devices like STM32L0/L4, there are two ways to keep the internal oscillator clock switched on during a low-power mode:

- Set the HSIKERON/MSIKERON bit in RCC_CR.
- Set the UCESM bit in USART/LPUART_CR3.
  This bit allows the USART/LPUART to request the clock all the time, and not only on START bit falling edge.

Note:

- *For consumption reasons, some products offer peripheral clock gating during Sleep and Stop modes. Consequently, for the USART and LPUART, the clock gating must be removed prior to entering in low-power mode. For more details, refer to the RCC section in the device reference manual,.*
- *For STM32F0/F3 devices, the internal oscillator clock is always switched off during Stop mode. It is switched on only when a falling edge is detected on the USART/LPUART receive line.*

5.1.2 LSE used as LPUART clock source

When the LSE clock is used as LPUART clock source, the maximum baudrate is 9600 bauds.

The LSE clock remains switched on in low-power mode, but, for STM32L0/L4 devices, it is not propagated to the LPUART if the LPUART does not request this kernel clock. To correctly receive data at 9600 bauds during a low-power mode, the UCESM bit must be set in LPUART_CR3: this bit allows the LPUART to request the clock at any time, not only on the START bit falling edge.

For the other STM32 devices, the LSE clock remains switched on in low-power mode, and is propagated to the LPUART. There is no constraint to receive data at 9600 bauds.

5.1.3 LSE used as USART clock source

When the LSE clock is used as USART clock source, the maximum baudrate is:

- 4096 bauds in case of oversampling by eight
- 2048 bauds in case of oversampling by 16

5.2 USART/LPUART internal oscillator clock off in low-power mode

If the internal oscillator clock is switched off during a low-power mode, the maximum baudrate to correctly wake up an STM32 MCU from a low-power mode depends on the following criteria:

- the wake-up time parameter ($t_{WUUSART}$ or $t_{WULPUART}$)
  - For STM32F0/F3/L0 devices, $t_{WUUSART}$ (or $t_{WULPUART}$) equals t_{WUSTOP} (as specified in the datasheet).
  - For the other STM32 devices, $t_{WUUSART}$ (or $t_{WULPUART}$) is specified in the datasheet.

- the USART receiver tolerance, which depends on the following parameters:
  - 9-, 10-, or 11-bit character length, configured through the M bits in USART_CR1
  - oversampling by eight or 16, configured through the OVER8 bit in USART_CR1
  - BRR[3:0] = or ≠ 0 in USART_BRR
  - one or three sample bits used to sample data, depending on ONEBIT in USART_CR3

  The tables below summarize the USART receiver tolerance according to the values of the above parameters.

Table 4. USART receiver tolerance when BRR[3:0] = 0x0

M bits	USART receiver tolerance (%) when OVER8 = 0		USART receiver tolerance (%) when OVER8 = 1	
	ONEBIT = 0	ONEBIT = 1	ONEBIT = 0	ONEBIT = 1
00	3.75	4.375	2.5	3.75
01	3.41	3.97	2.27	3.41
10	4.16	4.86	2.77	4.16

Table 5. USART receiver tolerance when BRR[3:0] ≠ 0x0

M bits	USART receiver tolerance (%) when OVER8 = 0		USART receiver tolerance (%) when OVER8 = 1	
	ONEBIT = 0	ONEBIT = 1	ONEBIT = 0	ONEBIT = 1
00	3.33	3.88	2	3
01	3.03	3.53	1.82	2.73
10	3.7	4.31	2.22	3.33

- the LPUART receiver tolerance, which depends on the following parameters:
  - number of Stop bits configured through STOP[1:0] bitfield in LPUART_CR2
  - value of LPUART_BRR

  The table below summarizes the LPUART receiver tolerance according to the values of the above parameters.

Table 6. LPUART receiver tolerance

M and Stop bits	LPUART receiver tolerance (%) when OVER8 = 0		LPUART receiver tolerance (%) when OVER8 = 1	
	ONEBIT = 0	ONEBIT = 1	ONEBIT = 0	ONEBIT = 1
8 bits (M = 00), 1 Stop bit	1.82	2.56	3.9	4.42
9 bits (M = 01), 1 Stop bit	1.69	2.33	2.53	4.14
7 bits (M = 10), 1 Stop bit	2.08	2.86	4.35	4.42
8 bits (M = 00), 2 Stop bits	2.08	2.86	4.35	4.42
9 bits (M = 01), 2 Stop bits	1.82	2.56	3.9	4.42
7 bits (M = 10), 2 Stop bits	2.34	3.23	4.92	4.42

The USART/LPUART asynchronous receiver works correctly only if the total clock system deviation is less than the USART/LPUART receiver tolerance. The following parameters contribute to the total deviation:

- DTRA: deviation due to the transmitter error (also includes the deviation of the transmitter local oscillator)
- DQUANT: error due to the baudrate quantization of the receiver
- DREC: deviation of the receiver local oscillator
- DTCL: deviation due to the transmission line (generally due to the transceivers, which introduce an asymmetry between the low-to-high and high-to-low transition timings)

The rule is then given by this formula

$$DTRA + DQUANT + DRAC + DTCL + DWU < USART\ or\ LPUART\ receiver\ tolerance$$

where DWU is the error due to sampling point deviation when the wake-up from a low-power mode is used.

The maximum baudrate to wake up the MCU from a low-power mode can be computed as follows:

- case 1: USART/LPUART receiver with 9-bit data length and M = 01

$$DWU_{\max} = \frac{t_{WUUSART/WULPUART}}{(11 \times T_{bit\ \min})}$$

where $T_{bit\ min}$ is the minimum bit duration

$$Maximum\ \ baudrate = \frac{(11 \times DWU_{\max})}{t_{WUUSART/WULPUART}}$$

- case 2: USART/LPUART receiver with 8-bit data length and M = 00

$$DWU_{\max} = \frac{t_{WUUSART/WULPUART}}{(10 \times T_{bit\ \min})}$$

$$Maximum\ \ baudrate = \frac{(10 \times DWU_{\max})}{t_{WUUSART/WULPUART}}$$

- case 3: USART/LPUART receiver with 7-bit data length and M = 10

$$DWU_{\max} = \frac{t_{WUUSART/WULPUART}}{(9 \times T_{bit\ \min})}$$

$$Maximum\ \ baudrate = \frac{(9 \times DWU_{\max})}{t_{WUUSART/WULPUART}}$$

STM32L4 example

Conditions: USART receiver with OVER8 = 0, M = 10, ONEBIT = 1, and BRR[3:0] = 0x0.

In this case, the USART receiver tolerance is 4.86 % (refer to Table 4).

In the ideal case where DTRA = DQUANT = DREC = DTCL = 0%, DWU max = 4.86%. In reality, the HSI inaccuracy must be taken into account.

For an HSI inaccuracy of 1 %:

- $t_{WUUSART}$ = 8.5 µs (for Stop 1 or Stop 2 modes)
- DWU_{max} = 4.86 - 1 % = 3.86 %
- $T_{bit\ min}$ = 8.5 / (9 x 3.86 %) = 24.4 µs

The maximum baudrate to correctly wake up the STM32 MCU from a low-power mode is then:

1 / 24.4 µs = ~ 40 kbauds.

6 Conclusion

This application note explains how the USART or the LPUART wakes up an STM32 MCU from a low-power mode. It also provides guidelines to approximately determine the associated USART/LPUART maximum baudrate.

Revision history

Table 7. **Document revision history**

Date	Version	Changes
8-Mar-2017	1	Initial release.
17-Dec-2019	2	Updated microcontroller list in: • Title of the document • Table 1: Applicable products • Section 5.2: USART/LPUART HSI kernel clock is OFF in low-power mode Added Section 1: General information (Arm logo added) Updated: • Table 2: Low-power modes versus STM32xx Series • Section 5.1.1: HSI clock used as USART/LPUART clock source • Section 5.1.2: LSE clock used as LPUART clock source
8-Dec-2022	3	STM32U5 Series inserted in this document Updated • Document title • Table 1. Applicable products • Section 2 STM32 low-power modes • Section 3.2 USART/LPUART wake-up sources • Section 4 How the USART/PLUART wakes up the STM32 with new Section 4.2 • Section 5.1.1 Internal oscillator clock used as USART/LPUART clock source
14-Dec-2022	4	Updated: • Table 1. Applicable products to include the STM32C0 Series • Table 2. STM32 low-power modes • Table 3. STM32 internal oscillators
16-Jan-2023	5	Updated: • Table 1. Applicable products to include the STM32WBA series • Table 2. STM32 low-power modes • Table 3. STM32 internal oscillators
05-Sep-2024	6	Added support for STM32H5 series, STM32H7 series, STM32N6 series, and STM32WB0 series.
14-Feb-2025	7	Added STM32U0 series and STM32U3 series. Updated: • Section 2: STM32 low-power modes. • Section 4.1: Wake-up from a low-power mode when the internal oscillator is off

Contents

List of tables

List of figures