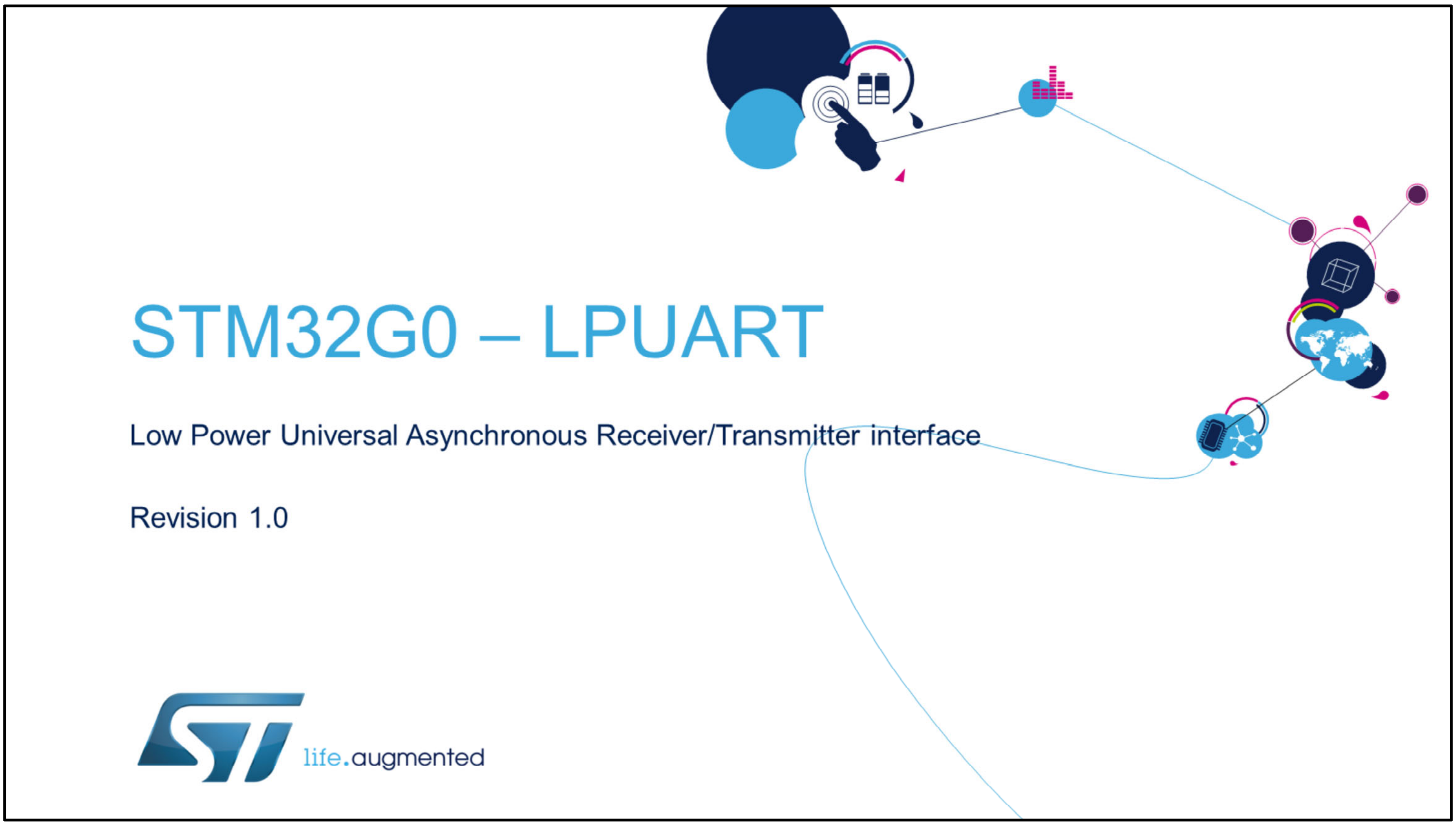


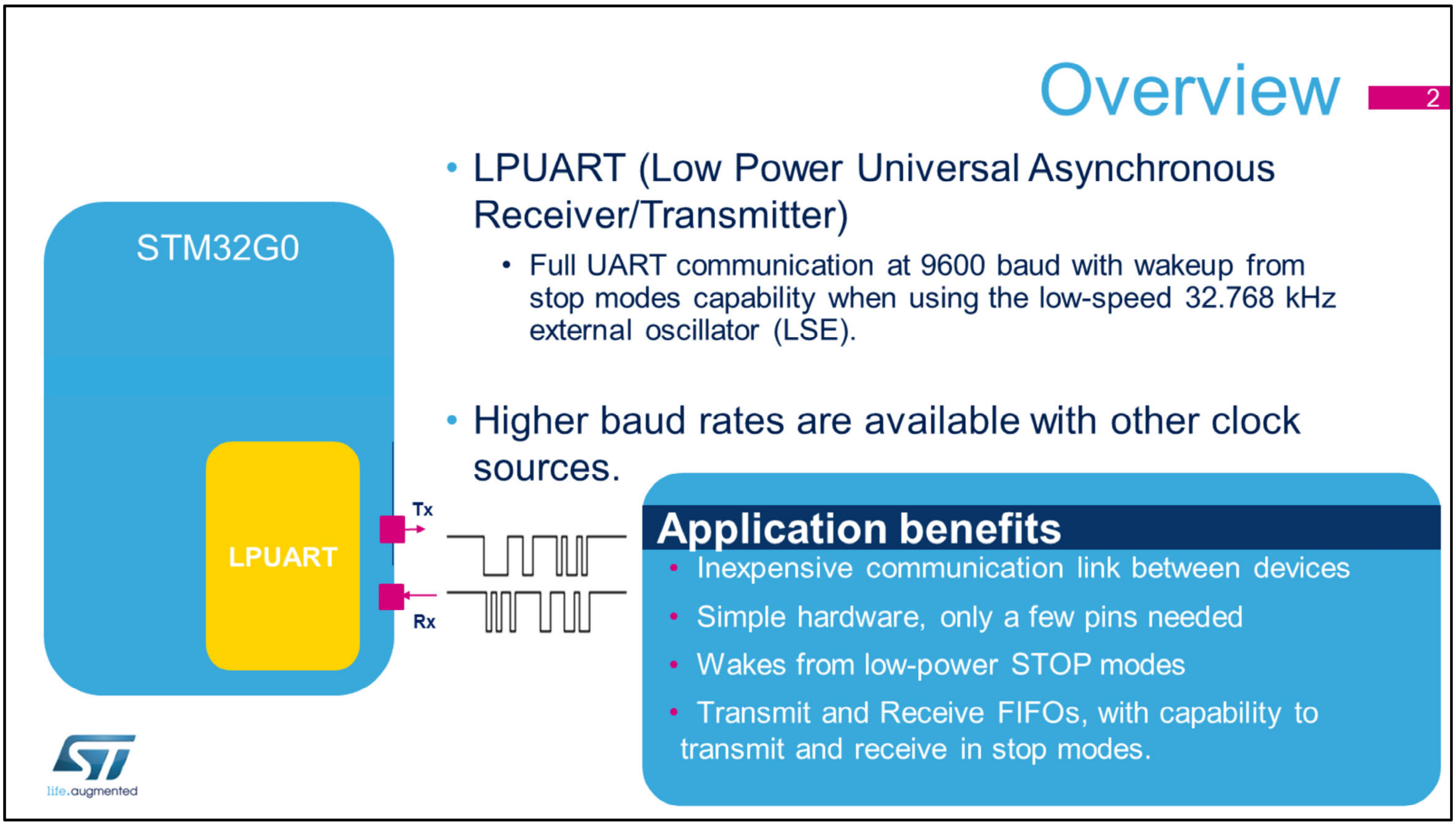
STM32G0 – LPUART

Low Power Universal Asynchronous Receiver/Transmitter interface

Revision 1.0

life.augmented

Hello, and welcome to this presentation of the STM32 Low Power Universal Asynchronous Receiver/Transmitter interface. It covers the main features of this interface, which is widely used for serial communications.



Overview

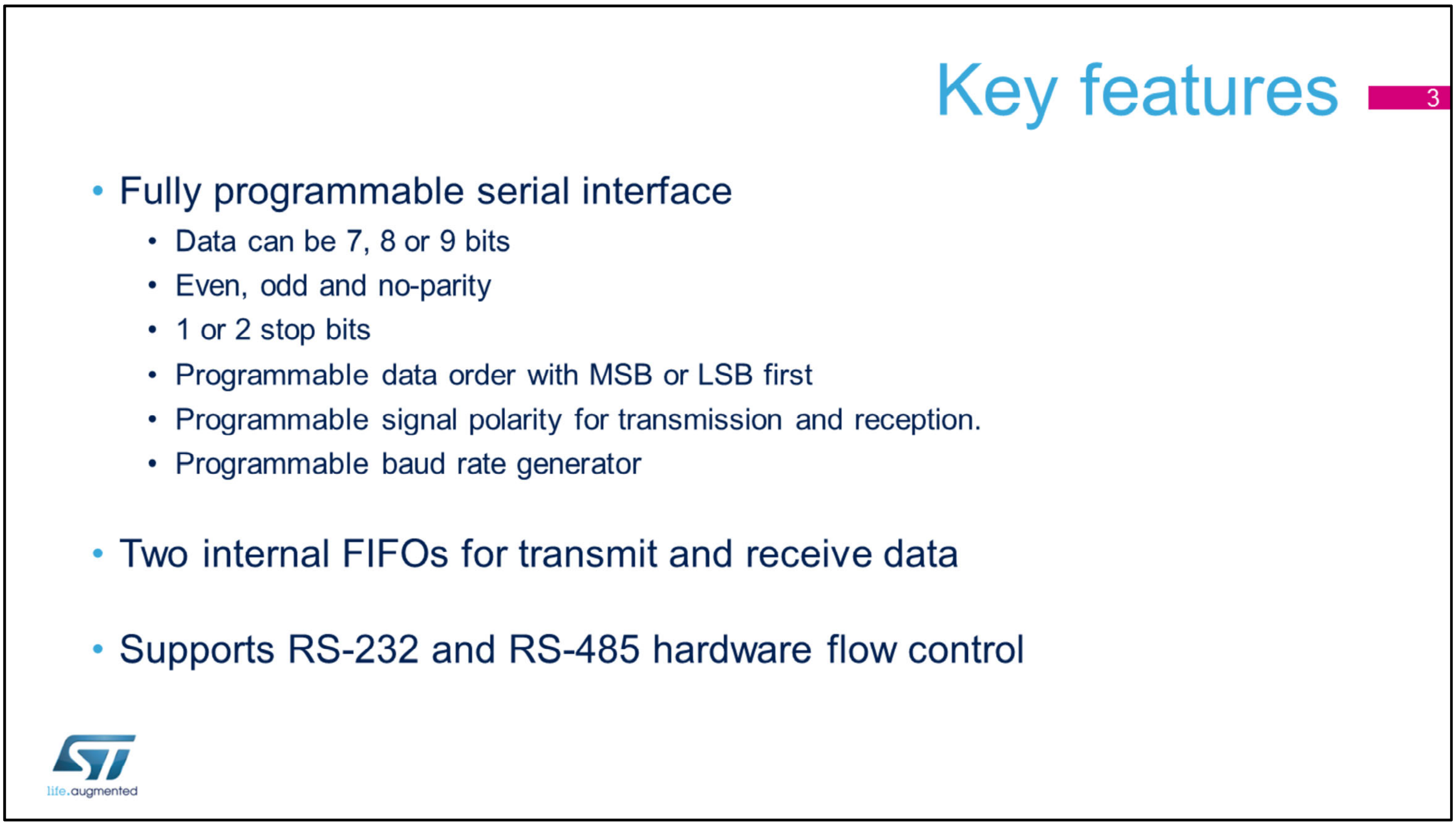
- LPUART (Low Power Universal Asynchronous Receiver/Transmitter)
  - Full UART communication at 9600 baud with wakeup from stop modes capability when using the low-speed 32.768 kHz external oscillator (LSE).
- Higher baud rates are available with other clock sources.

Application benefits

- Inexpensive communication link between devices
- Simple hardware, only a few pins needed
- Wakes from low-power STOP modes
- Transmit and Receive FIFOs, with capability to transmit and receive in stop modes.

life.augmented

The Low Power Universal Synchronous Asynchronous Receiver provides full UART communications at 9600 baud when the LPUART is clocked using a low-speed external 32.768 kHz oscillator (LSE).
Higher baudrates can be reached when it is clocked by clock sources different from the LSE clock.
Applications can benefit from the easy and inexpensive connection between devices, requiring only a few pins. In addition, the LPUART peripheral is functional in low-power modes. It comes with Transmit and Receive FIFOs with capability to transmit and receive in stop modes.

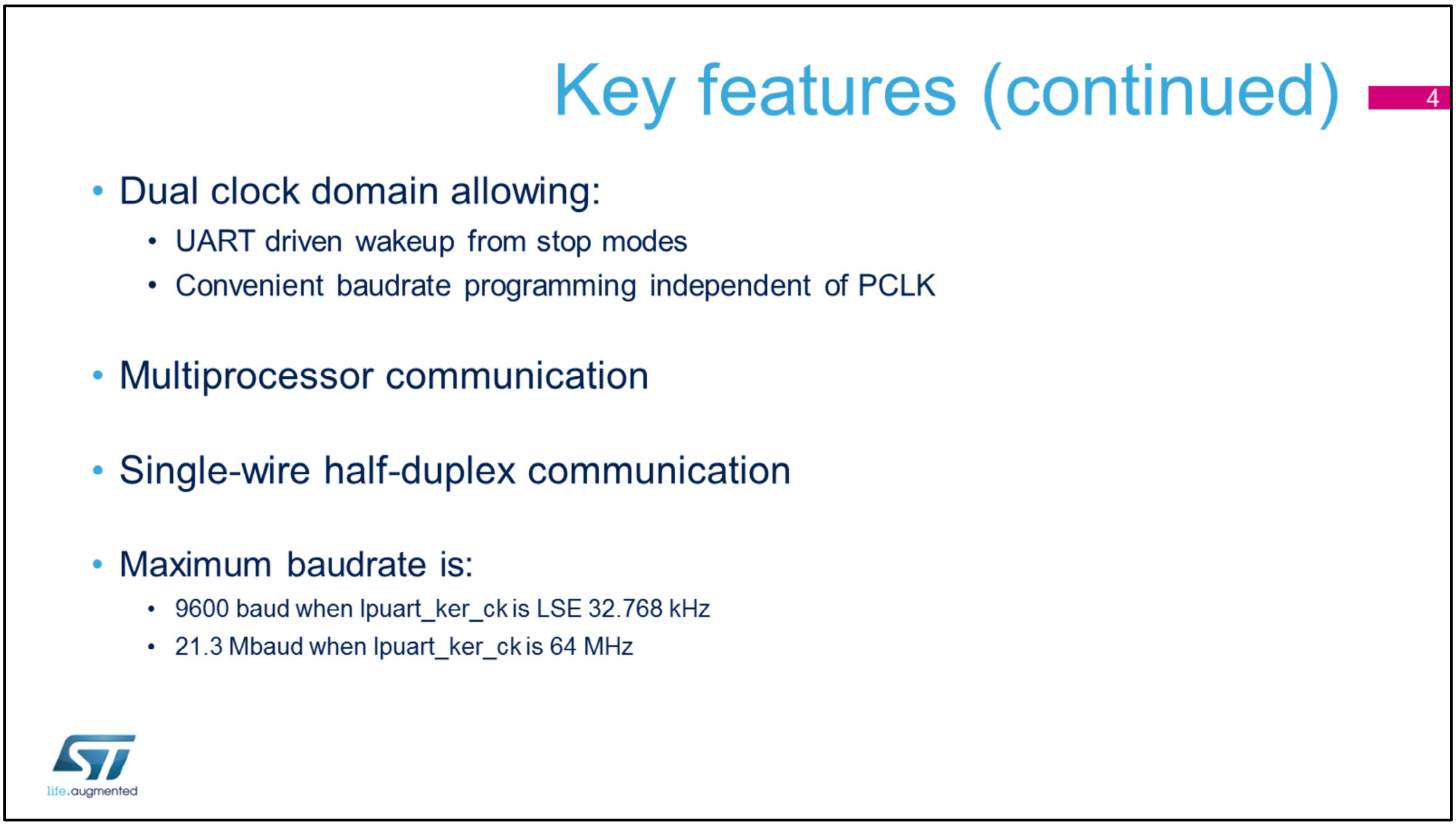


Key features

- Fully programmable serial interface
  - Data can be 7, 8 or 9 bits
  - Even, odd and no-parity
  - 1 or 2 stop bits
  - Programmable data order with MSB or LSB first
  - Programmable signal polarity for transmission and reception.
  - Programmable baud rate generator
- Two internal FIFOs for transmit and receive data
- Supports RS-232 and RS-485 hardware flow control

The LPUART is a fully programmable serial interface with configurable features such as data length, parity that is automatically generated and checked, number of stop bits, data order, signal polarity for transmission and reception, and baud rate generator.

The LPUART can operate in FIFO mode and it comes with Transmit and Receive FIFOs.
It supports RS-232 and RS-485 hardware flow control options.



Key features (continued)

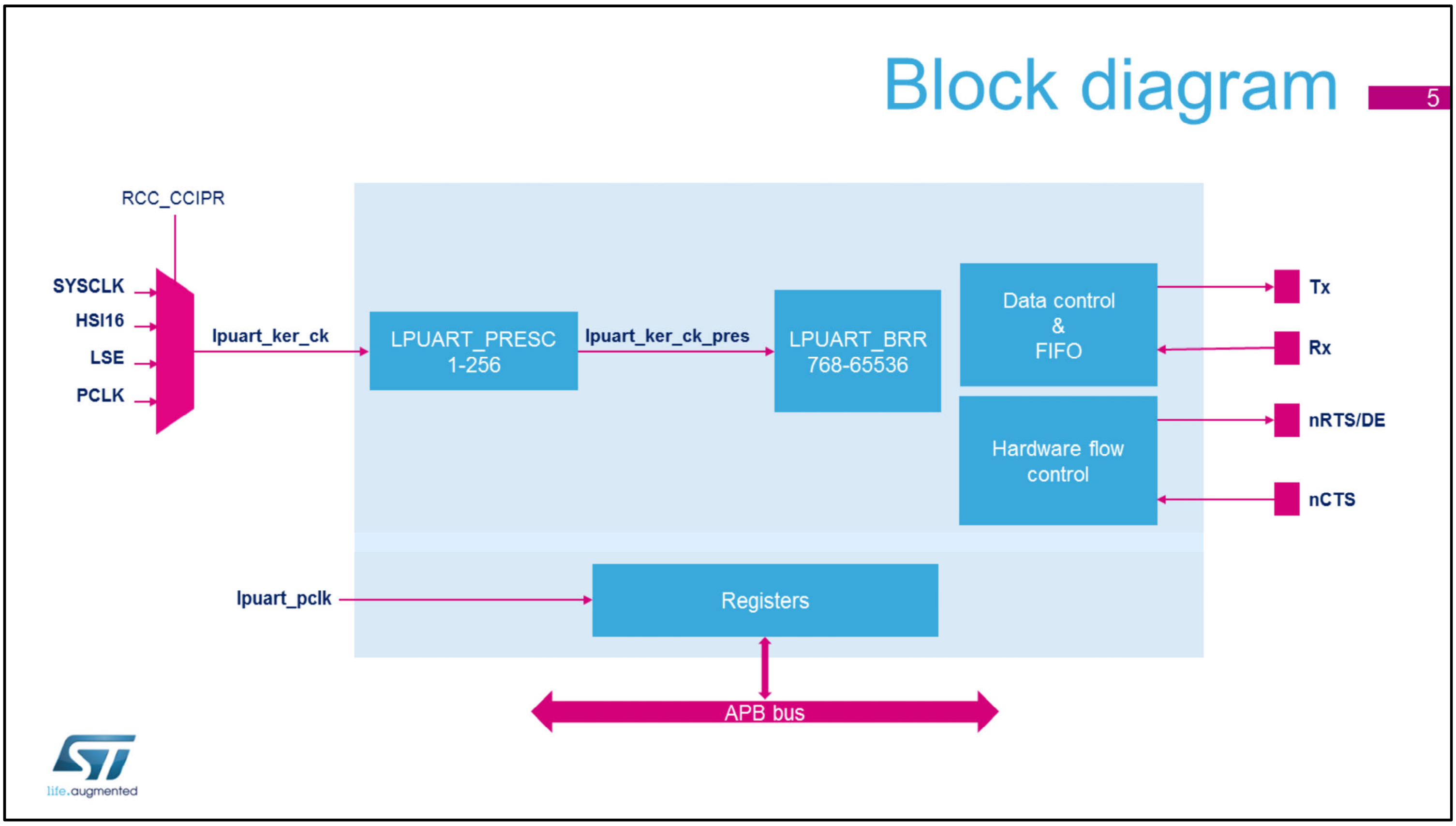
- Dual clock domain allowing:
  - UART driven wakeup from stop modes
  - Convenient baudrate programming independent of PCLK
- Multiprocessor communication
- Single-wire half-duplex communication
- Maximum baudrate is:
  - 9600 baud when lpuart_ker_ck is LSE 32.768 kHz
  - 21.3 Mbaud when lpuart_ker_ck is 64 MHz

The LPUART supports dual clock domains allowing for wake up from stop modes and baud rate programming independent of the peripheral clock.

The multi-processor mode allows the LPUART to remain idle when not addressed.

In addition to full duplex communication, it also supports single-wire half-duplex mode.

The maximum baudrate is 9600 baud when the clock source is the LSE, 21.3 Mbaud when it is the SYSCLK.

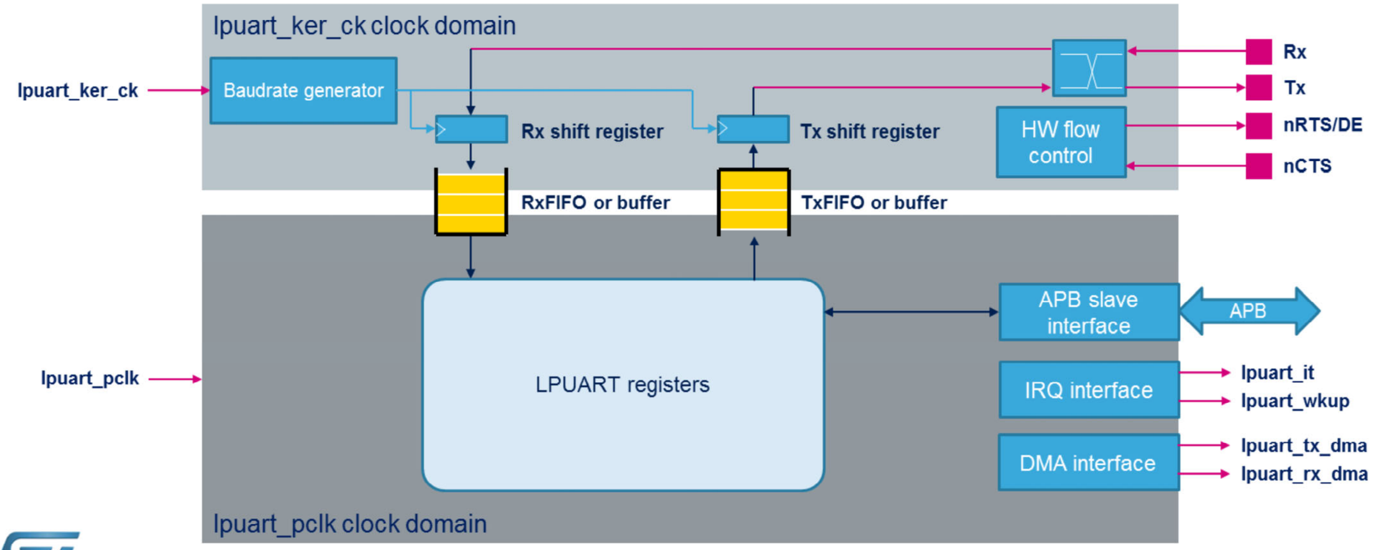


Block diagram

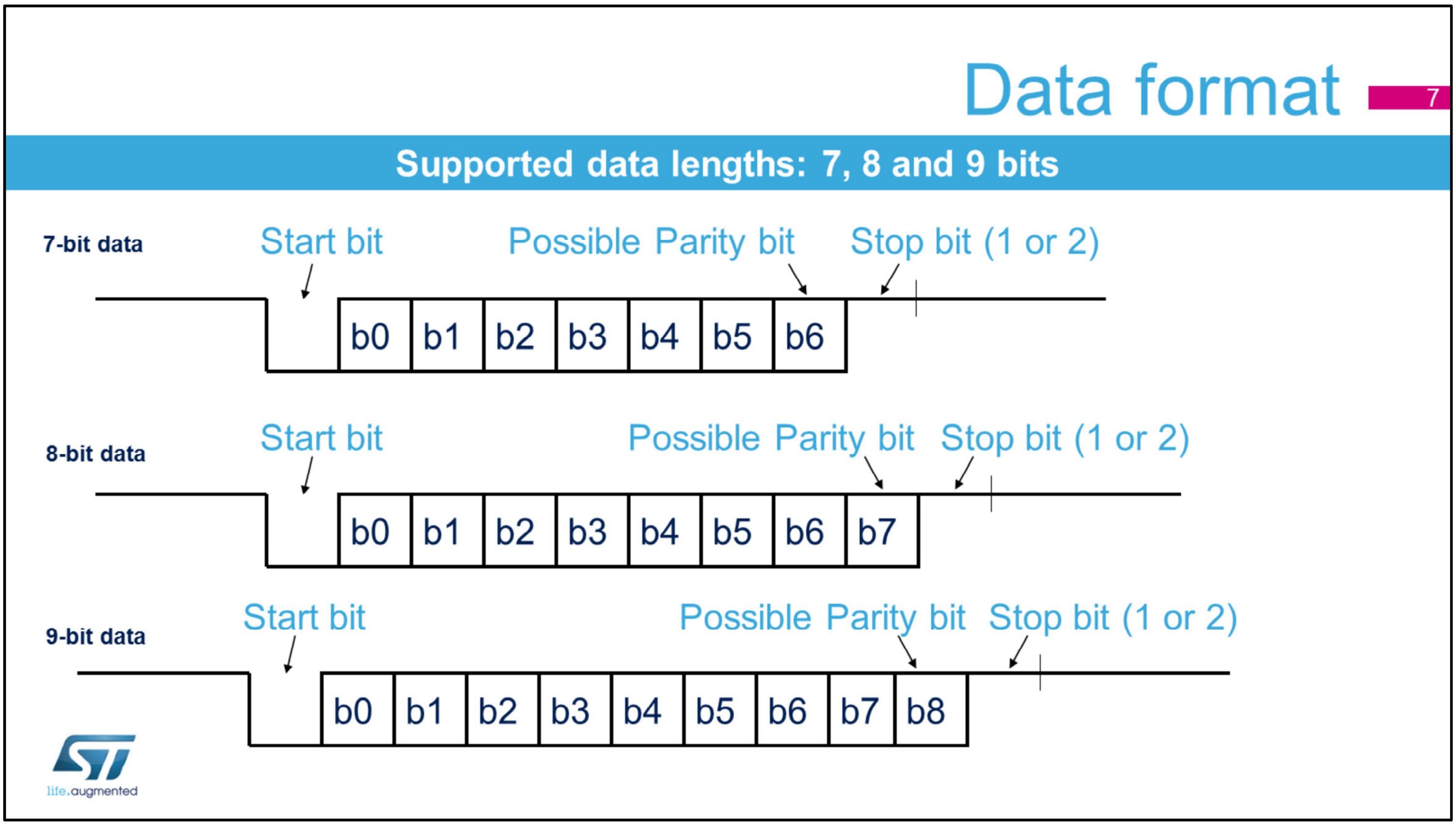
This is the LPUART block diagram.
The LPUART clock source (lpuart_ker_ck) can be selected from several sources: peripheral clock (APB clock PCK), SYSCLK, High Speed Internal 16-MHz oscillator (HSI16), Low Speed External Oscillator (LSE).
The LPUART clock source is divided by a programmable factor in the LPUART_PRESC register in range 1 to 256.
Tx and Rx pins are used for data transmission and reception.
nCTS and nRTS pins are used for RS-232 hardware flow control.
The Driver Enable pin (DE) which is available on the same I/O as nRTS is used in RS-485 mode.

Dual clock domain

Baudrate programming independent from PCLK reprogramming



The LPUART has a flexible clocking scheme.
The registers are accessed through the APB bus, and the kernel is clocked with lpuart_ker_ck (prescaled or not) which is independent from the APB clock.
In order to pass data from one clock domain to the other one, either 8-data FIFOs are used or single data buffers.
The LPUART block is an APB slave that can rely on DMA requests to transfer data to/from memory buffers.
The TX and RX pins functions can be swapped. This allows to work in the case of a cross-wired connection to another UART

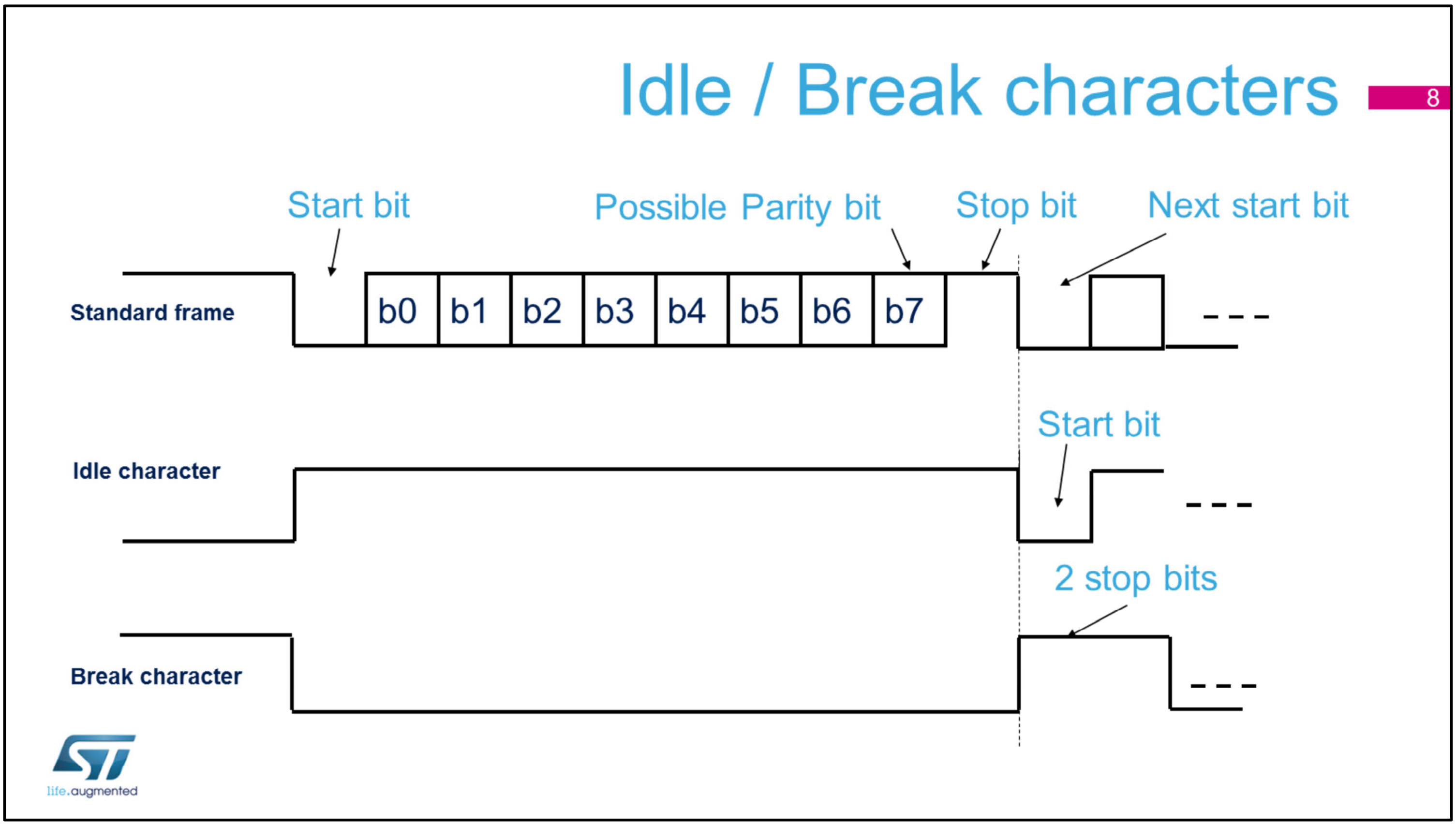


Data format

Supported data lengths: 7, 8 and 9 bits

The frame format consists of a set of data bits in addition to bits for synchronization and optionally a parity bit for error checking. A frame starts with one start-bit (S), where the line is driven low for one bit-period. This signals the start of a frame and is used for synchronization.

The data length can be 9, 8, or 7 bits with the parity bit counted. Finally, 1 or 2 stop bits, where the line is driven high, indicate the end of the frame.

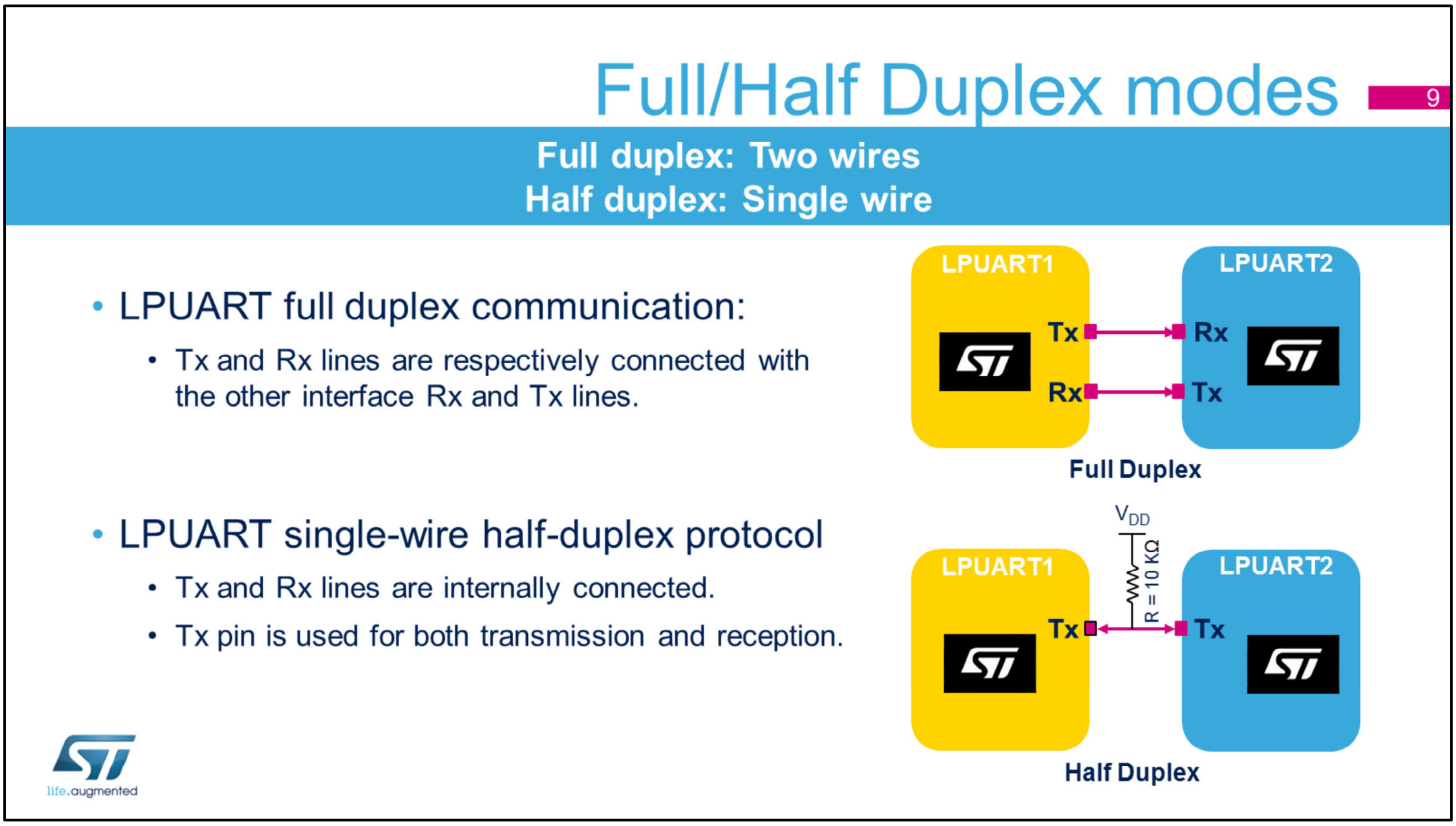


Idle / Break characters

The previous slide described a standard frame. This slide shows an example of an 8-bit data frame configured with 1 stop bit.

An Idle character is interpreted as an entire frame of “1”s. The number of “1”s will include the number of stop bits as well.

A Break character is interpreted as receiving all “0”s for a frame period. At the end of the break frame, 2 stop bits are inserted.



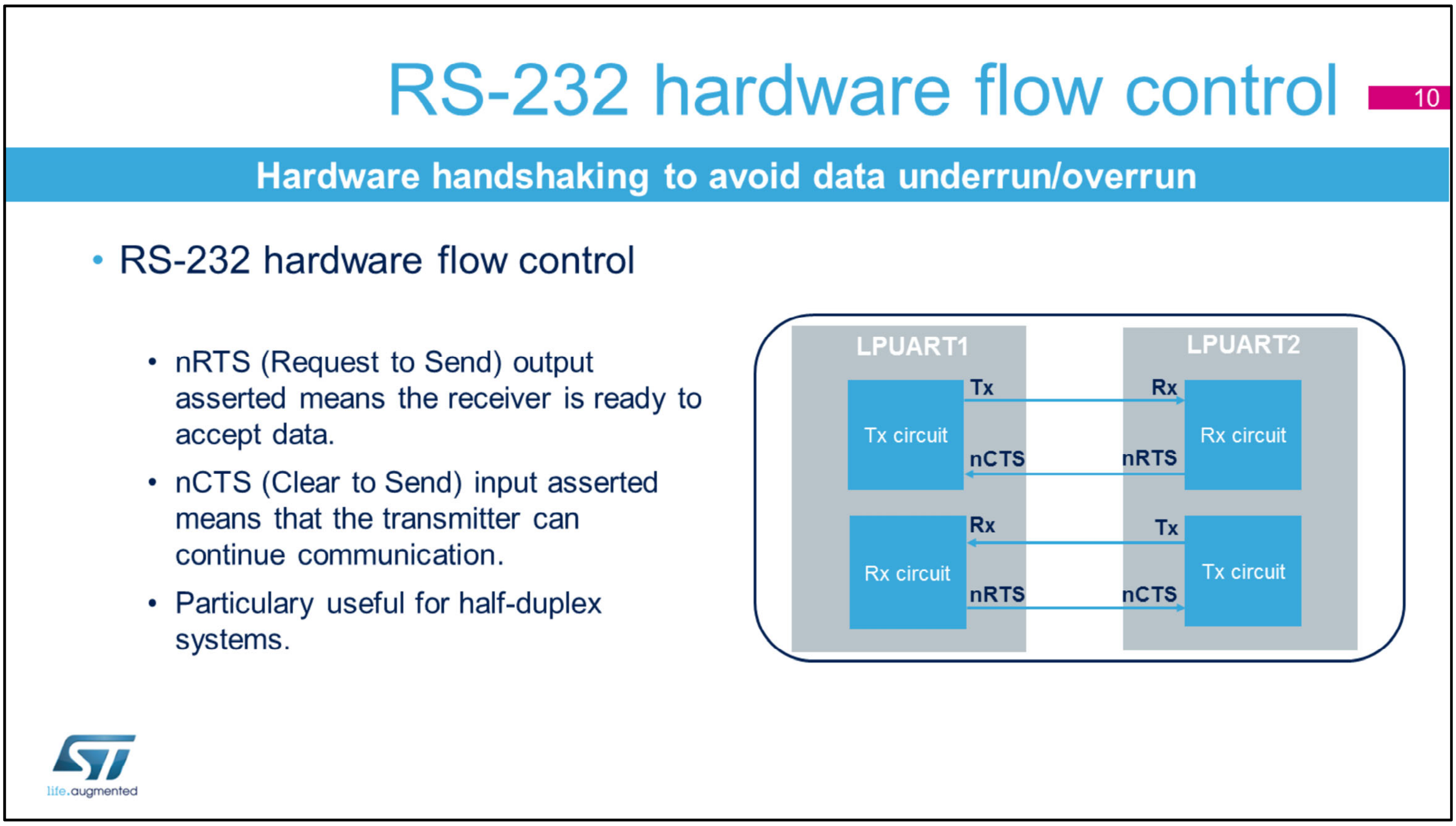
Full/Half Duplex modes

Full duplex: Two wires
Half duplex: Single wire

- LPUART full duplex communication:
  - Tx and Rx lines are respectively connected with the other interface Rx and Tx lines.
- LPUART single-wire half-duplex protocol
  - Tx and Rx lines are internally connected.
  - Tx pin is used for both transmission and reception.

The LPUART supports full-duplex communication where the Tx and Rx lines are respectively connected with the other interface’s Rx and Tx lines.

The LPUART can be also configured for single-wire half-duplex protocol where the Tx and Rx lines are internally connected. In this communication mode, only the Tx pin is used for both transmission and reception. The Tx pin is always released when no data is transmitted. Thus, it acts as a standard I/O in idle or reception states. For this usage, the I/O must be configured with the Tx pin in alternate function open-drain mode with an external pull-up resistor.



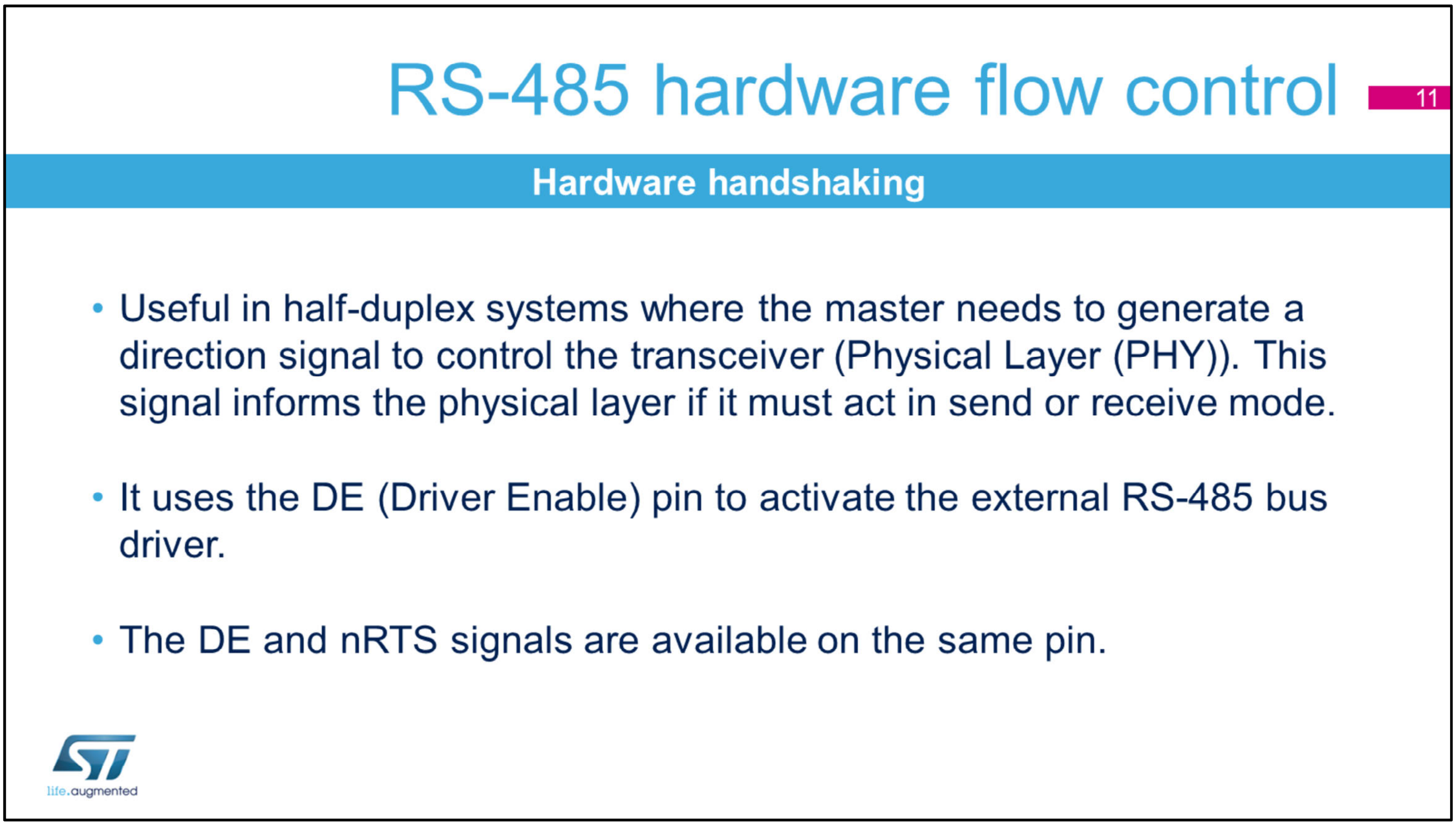
RS-232 hardware flow control

Hardware handshaking to avoid data underrun/overrun

- RS-232 hardware flow control
  - nRTS (Request to Send) output asserted means the receiver is ready to accept data.
  - nCTS (Clear to Send) input asserted means that the transmitter can continue communication.
  - Particulary useful for half-duplex systems.

LPUART1
LPUART2
Tx circuit
Tx
Rx
Rx circuit
nCTS
nRTS
Rx circuit
Rx
Tx
Tx circuit
nRTS
nCTS
life.augmented

In the RS-232 standard, it is possible to control the serial data flow between two devices by using the nCTS input and the nRTS output. These two lines allow the receiver and the transmitter to alert each other of their state. This slide shows how to connect two devices in this mode. The idea is to prevent dropped bytes or conflicts in case of half-duplex communication. Both signals are active low.

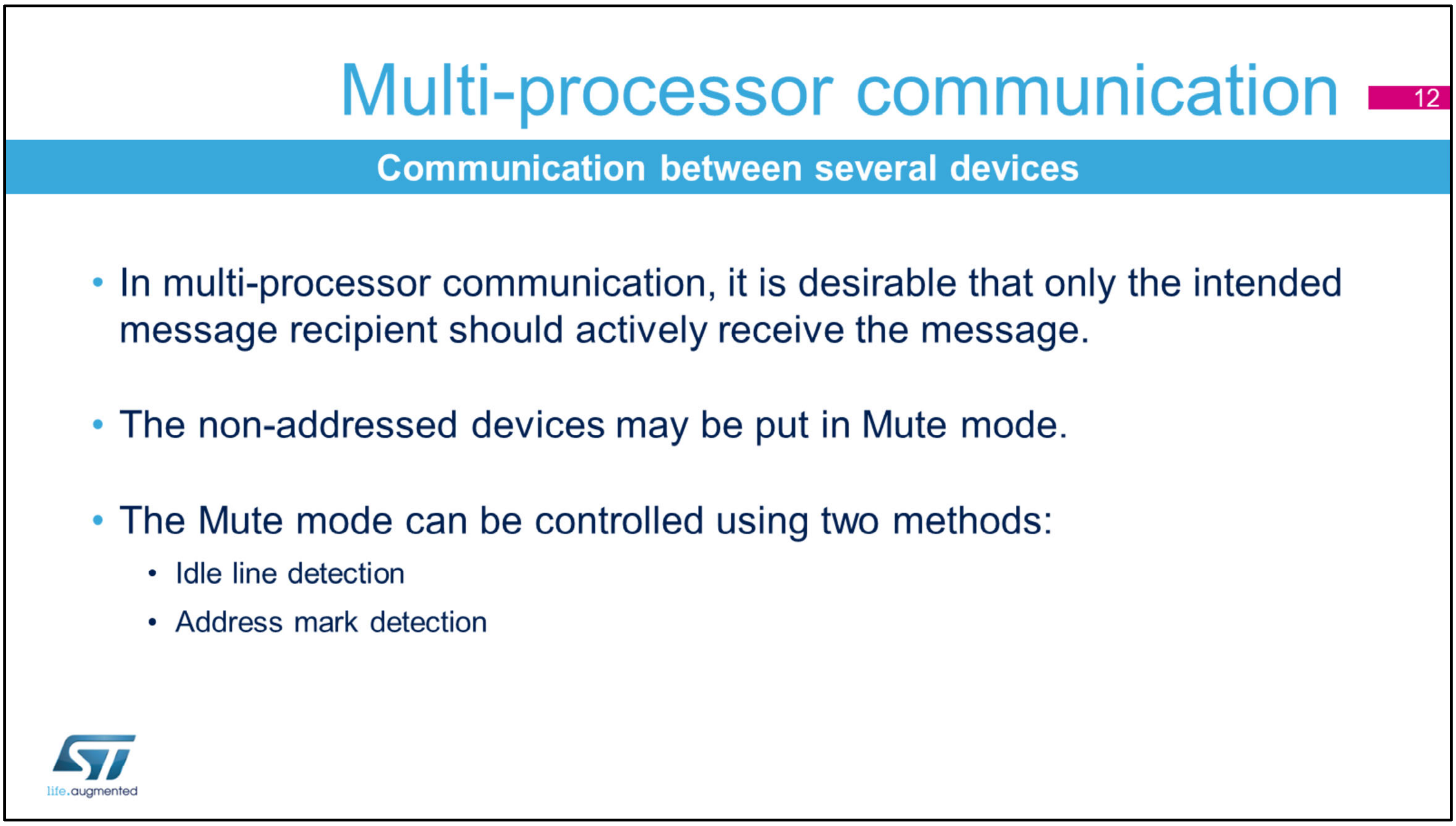


RS-485 hardware flow control

Hardware handshaking

- Useful in half-duplex systems where the master needs to generate a direction signal to control the transceiver (Physical Layer (PHY)). This signal informs the physical layer if it must act in send or receive mode.
- It uses the DE (Driver Enable) pin to activate the external RS-485 bus driver.
- The DE and nRTS signals are available on the same pin.

For serial half-duplex communication protocols like RS-485, the master needs to generate a direction signal to control the transceiver (Physical Layer). This signal informs the physical layer if it must act in send or receive mode.
In RS-485 mode, a control line “Driver enable” is used to activate the external transceiver control. The DE control line shares the pin with nRTS.



Multi-processor communication

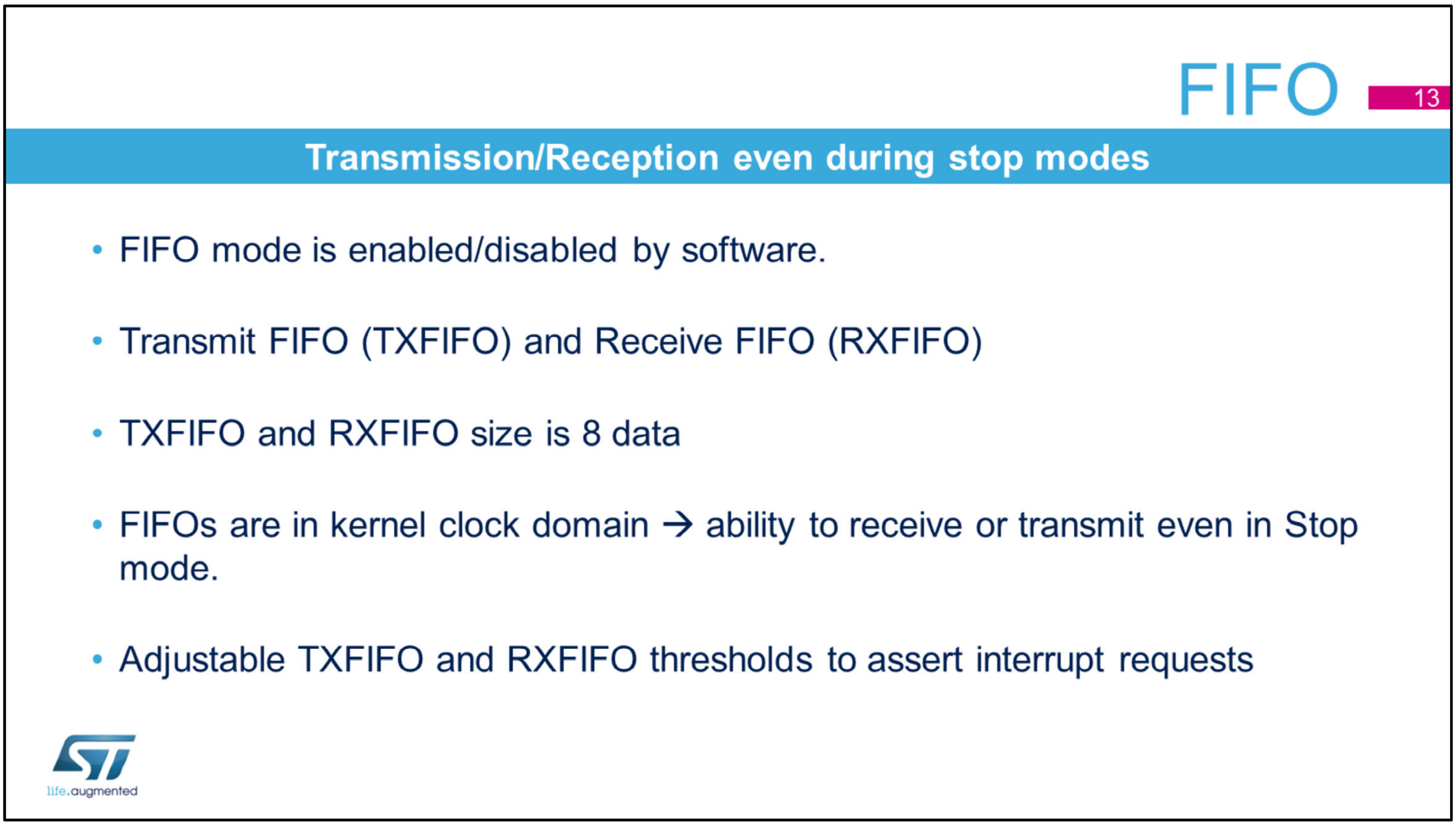
Communication between several devices

- In multi-processor communication, it is desirable that only the intended message recipient should actively receive the message.
- The non-addressed devices may be put in Mute mode.
- The Mute mode can be controlled using two methods:
  - Idle line detection
  - Address mark detection

life.augmented

To simplify communication between multiple processors, the LPUART supports a special multi-processor mode. In multi-processor communication, it is desirable that only the intended message recipient should actively receive the message. The non-addressed devices may be put in Mute mode using two methods: Idle line or address mark.
The LPUART can enter or exit from Mute mode using one of two methods:

- Idle line detection
- Address mark detection



FIFO

13

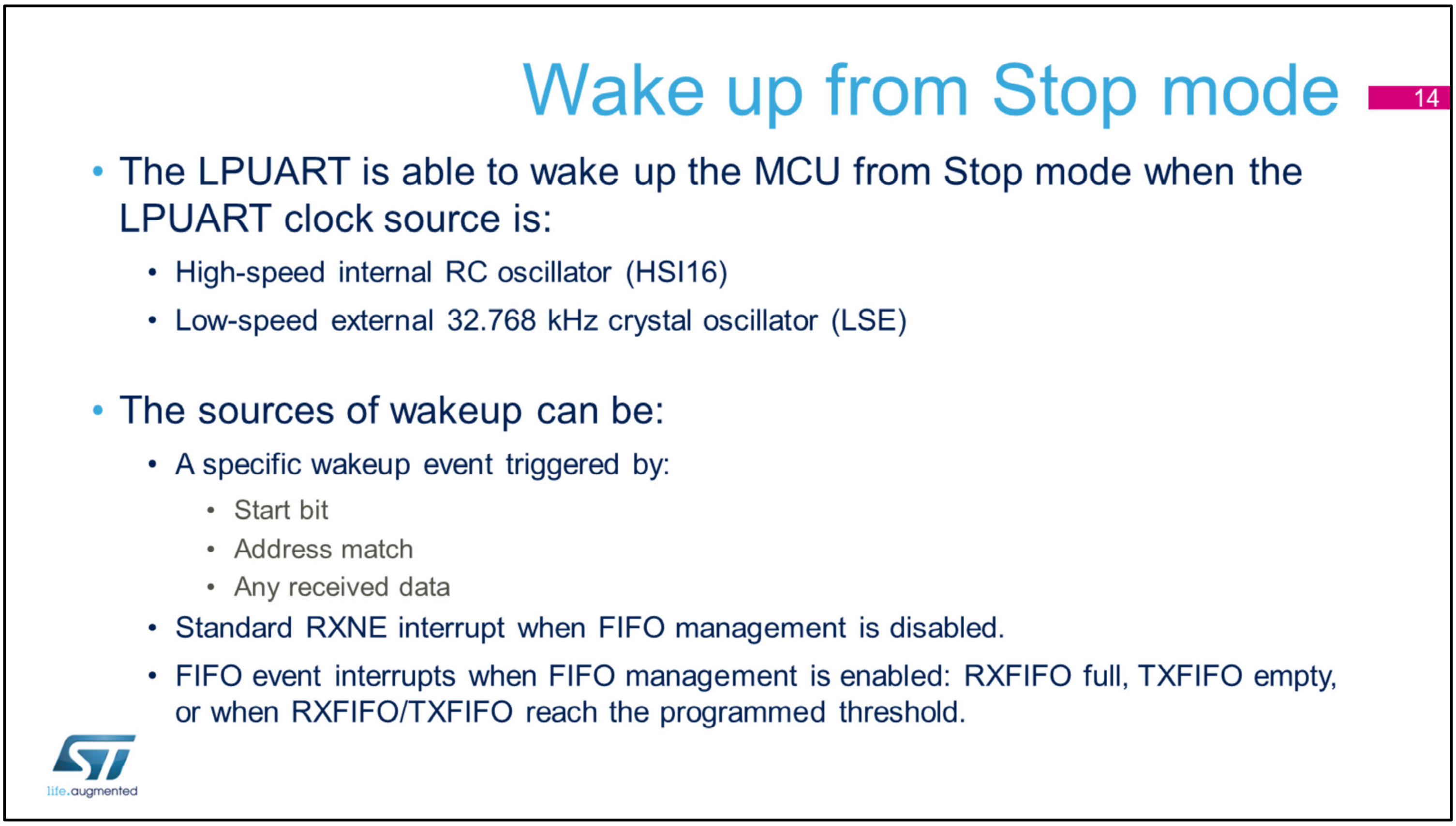
Transmission/Reception even during stop modes

- FIFO mode is enabled/disabled by software.
- Transmit FIFO (TXFIFO) and Receive FIFO (RXFIFO)
- TXFIFO and RXFIFO size is 8 data
- FIFOs are in kernel clock domain → ability to receive or transmit even in Stop mode.
- Adjustable TXFIFO and RXFIFO thresholds to assert interrupt requests

The LPUART can operate in FIFO mode which is enabled/disabled by software. It is disabled by default. The LPUART comes with a Transmit FIFO (TXFIFO) and a Receive FIFO (RXFIFO), each being 8 data deep.

The TXFIFO is 9-bits wide. The RXFIFO default width is 12 bits. This is due to the fact that the receiver does not only store the data in the FIFO, but also the error flags associated to each character (Parity error, Noise error and Framing error flags).

Provided that the TXFIFO and RXFIFO are clocked by the kernel clock, it is possible to transmit and receive data even in Stop mode.
It is possible to configure TXFIFO and RXFIFO thresholds, -used mainly to avoid underrun/overrun issue while waking up from Stop mode.



Wake up from Stop mode

14

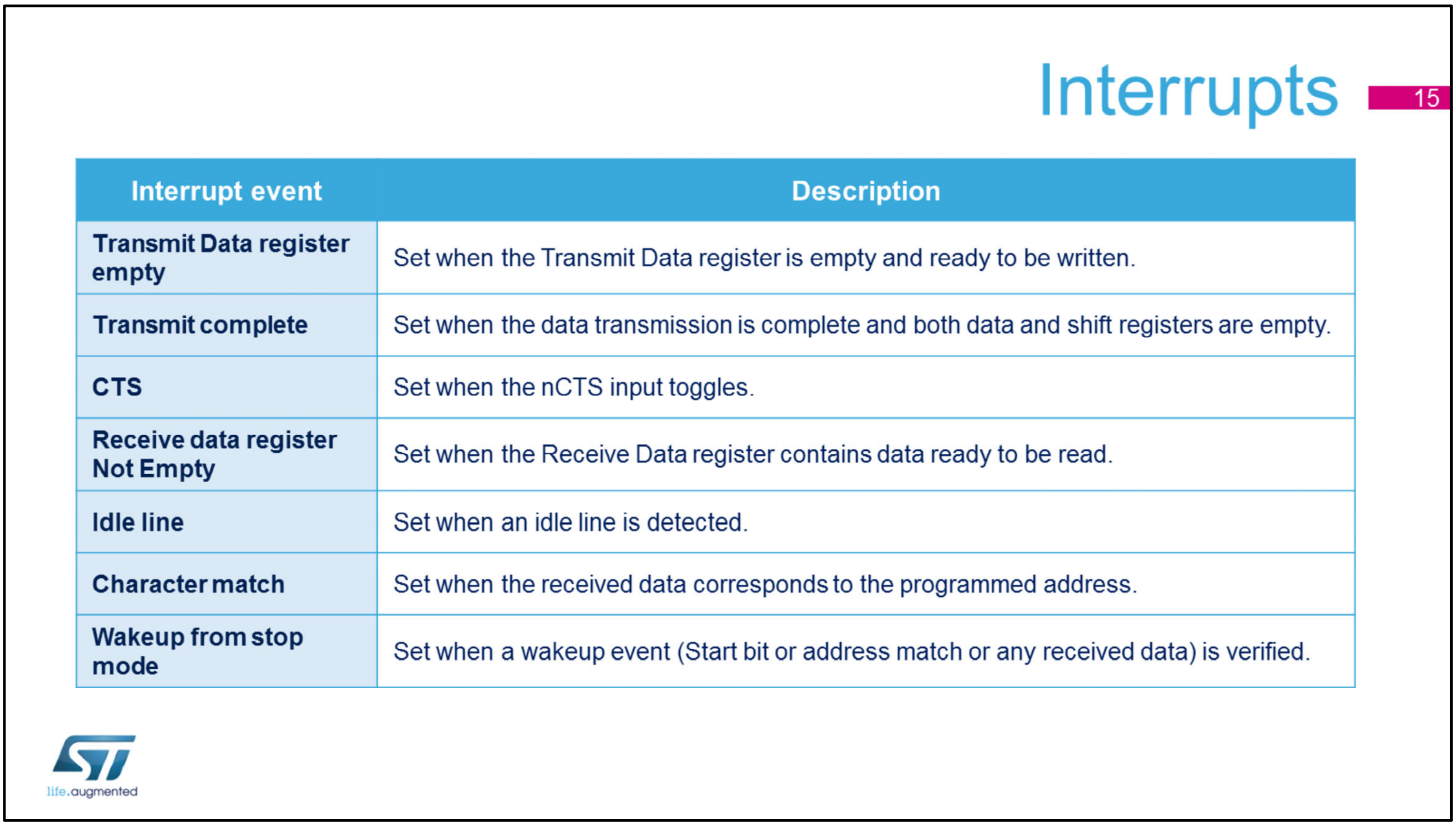
- The LPUART is able to wake up the MCU from Stop mode when the LPUART clock source is:
  - High-speed internal RC oscillator (HSI16)
  - Low-speed external 32.768 kHz crystal oscillator (LSE)

- The sources of wakeup can be:
  - A specific wakeup event triggered by:
    - Start bit
    - Address match
    - Any received data
  - Standard RXNE interrupt when FIFO management is disabled.
  - FIFO event interrupts when FIFO management is enabled: RXFIFO full, TXFIFO empty, or when RXFIFO/TXFIFO reach the programmed threshold.

The LPUART is able to wake up the MCU from STOP mode when the LPUART clock source is the HSI or LSE clock.

The sources of wakeup can be:

- A specific wakeup event which is triggered by either a start bit or an address match or any received data.
- An RXNE interrupt when FIFO management is disabled or
- FIFO event interrupts when FIFO management is enabled.

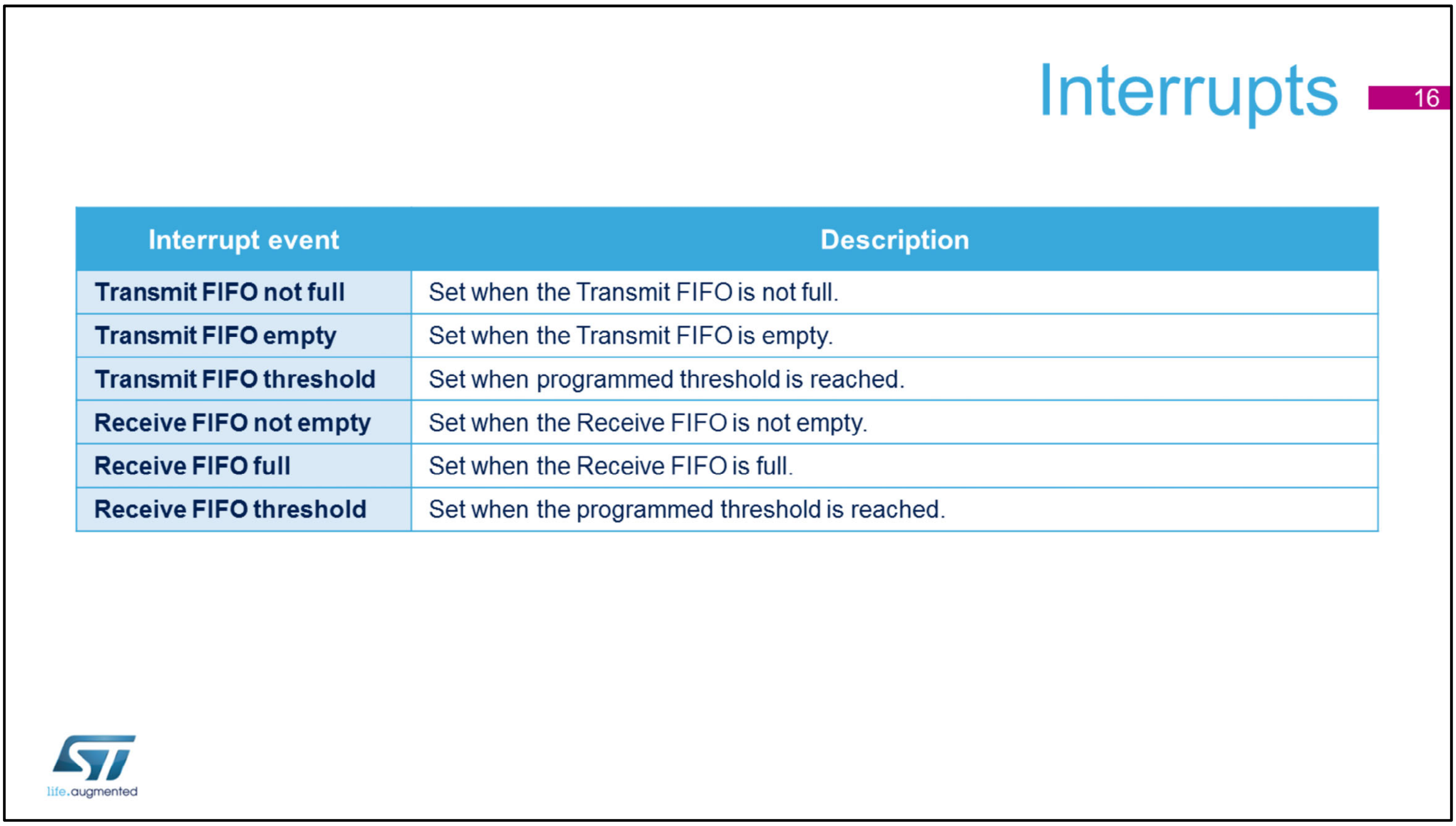


Interrupts

Interrupt event	Description
Transmit Data register empty	Set when the Transmit Data register is empty and ready to be written.
Transmit complete	Set when the data transmission is complete and both data and shift registers are empty.
CTS	Set when the nCTS input toggles.
Receive data register Not Empty	Set when the Receive Data register contains data ready to be read.
Idle line	Set when an idle line is detected.
Character match	Set when the received data corresponds to the programmed address.
Wakeup from stop mode	Set when a wakeup event (Start bit or address match or any received data) is verified.

life.augmented

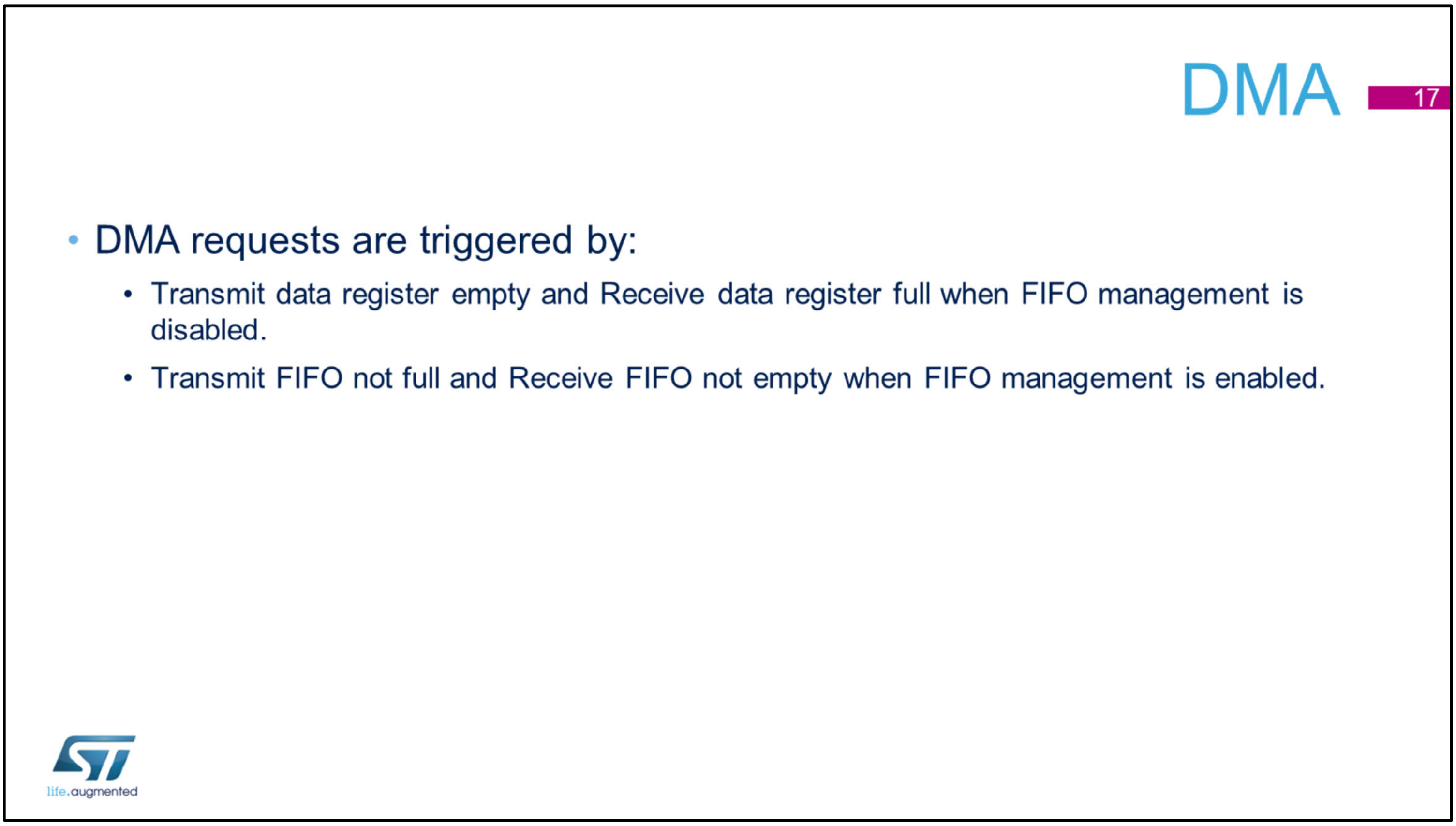
This table lists the LPUART events that can generate an interrupt.



Interrupts

Interrupt event	Description
Transmit FIFO not full	Set when the Transmit FIFO is not full.
Transmit FIFO empty	Set when the Transmit FIFO is empty.
Transmit FIFO threshold	Set when programmed threshold is reached.
Receive FIFO not empty	Set when the Receive FIFO is not empty.
Receive FIFO full	Set when the Receive FIFO is full.
Receive FIFO threshold	Set when the programmed threshold is reached.

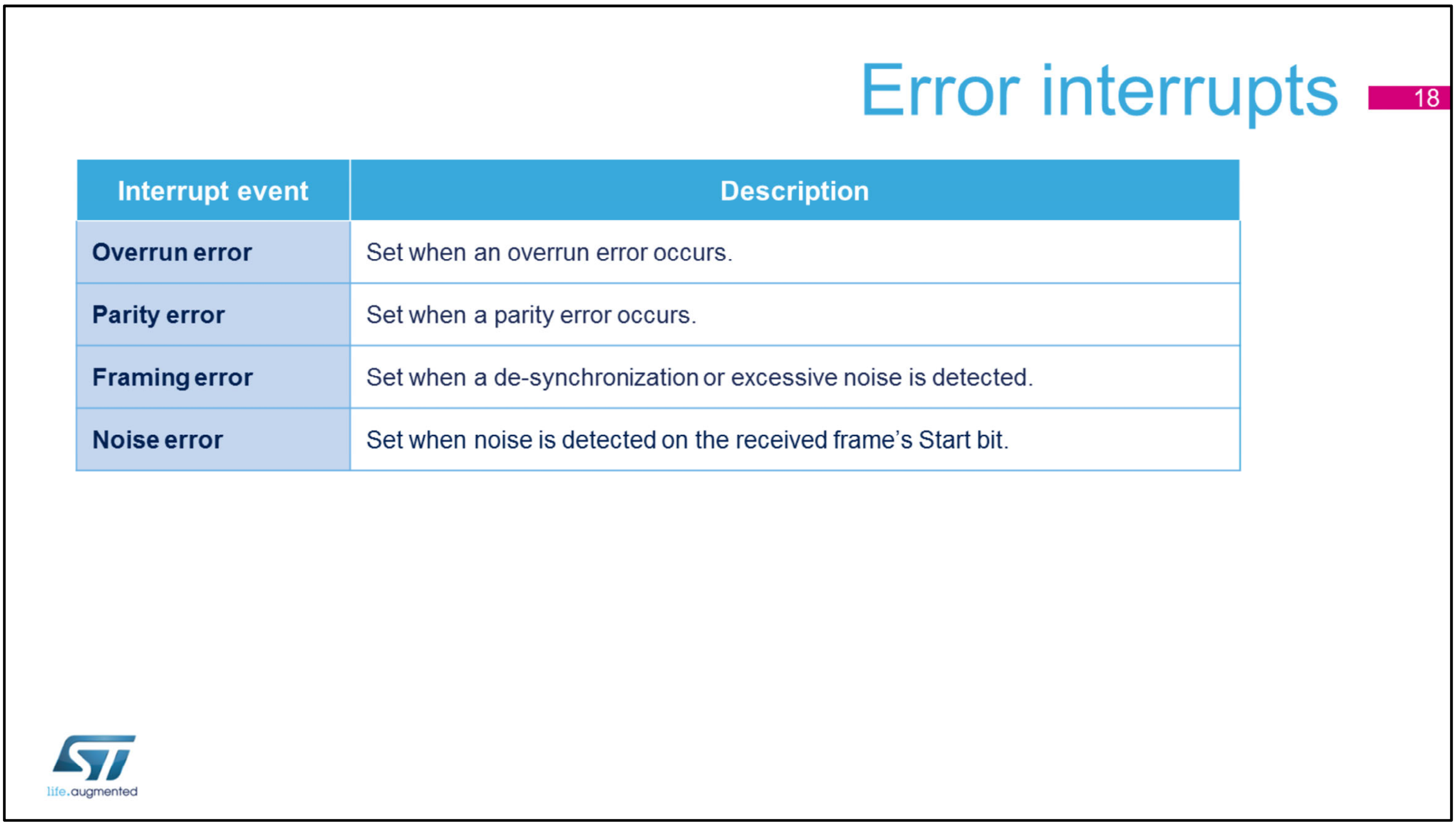
This table lists the FIFO events interrupts when the FIFO management is enabled.



DMA

- DMA requests are triggered by:
  - Transmit data register empty and Receive data register full when FIFO management is disabled.
  - Transmit FIFO not full and Receive FIFO not empty when FIFO management is enabled.

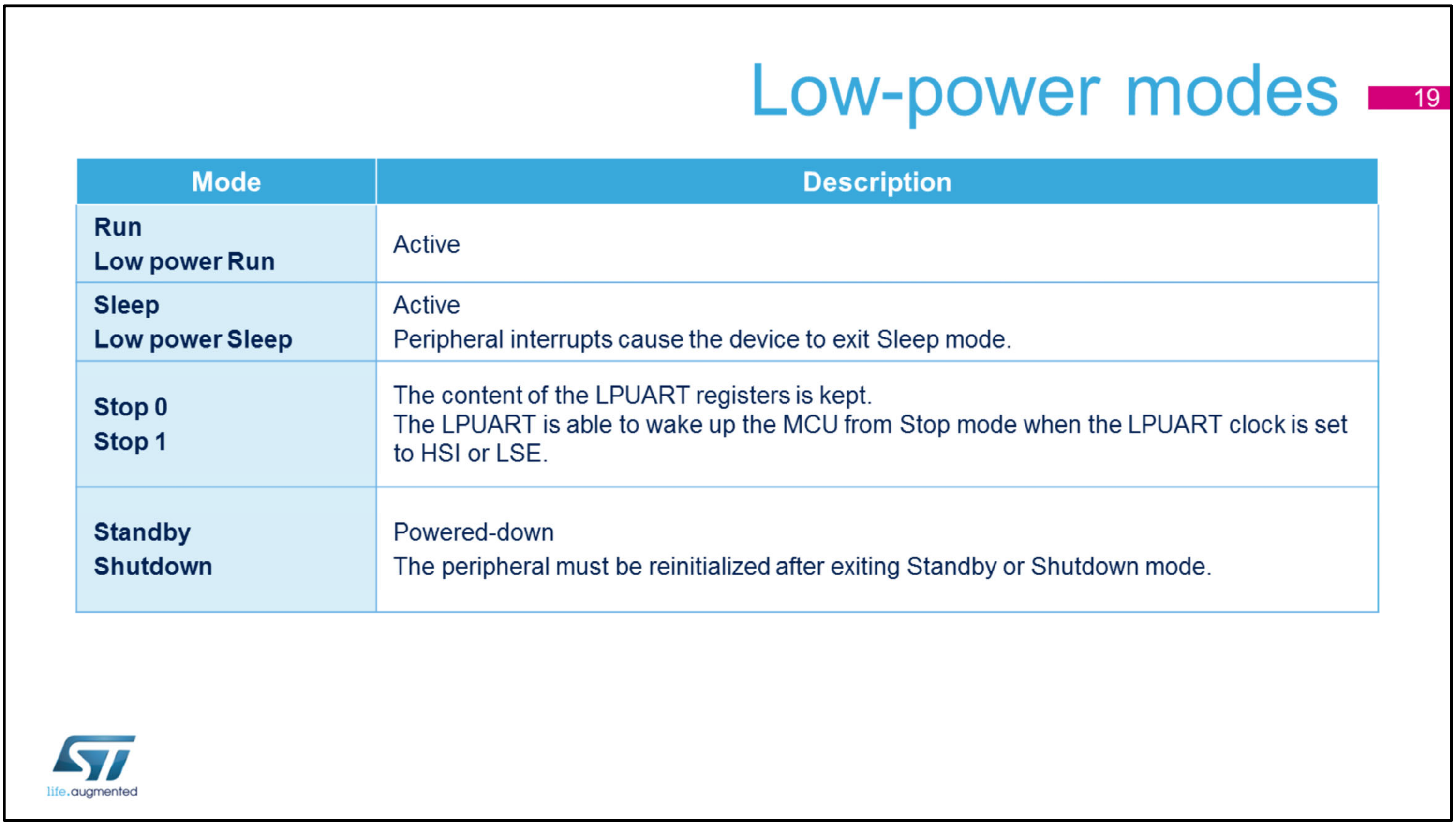
The DMA requests can be generated when Receive Buffer Not Empty or Transmit Buffer Empty flags are set when FIFO management is disabled.
The DMA requests can be generated when the Transmit FIFO not full and Receive FIFO not empty flags are set when FIFO management is enabled.



Error interrupts

Interrupt event	Description
Overrun error	Set when an overrun error occurs.
Parity error	Set when a parity error occurs.
Framing error	Set when a de-synchronization or excessive noise is detected.
Noise error	Set when noise is detected on the received frame’s Start bit.

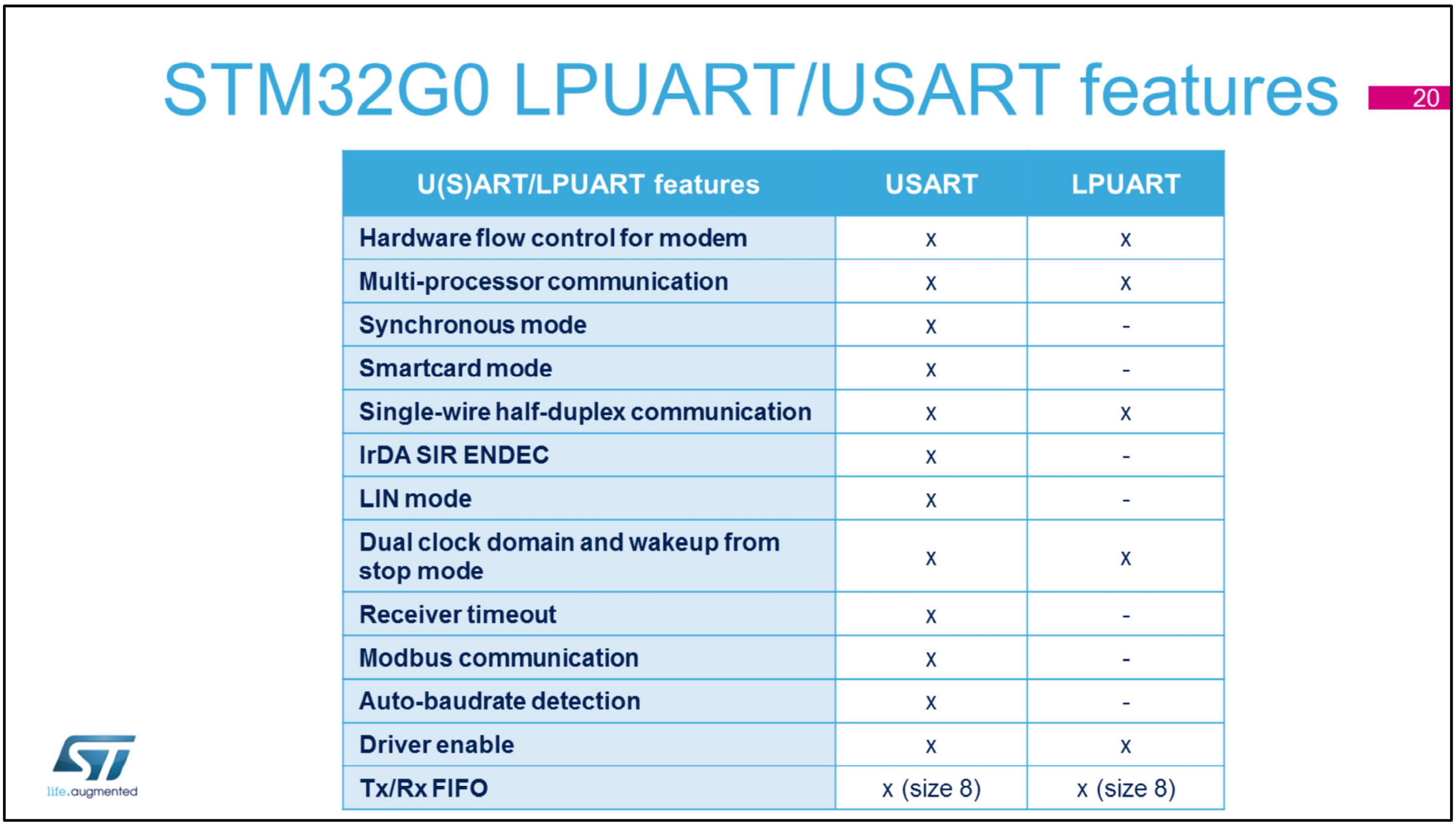
Several errors flags can also be generated by the LPUART as shown in the table.
The Overrun, Parity and Framing error flags are each set when the corresponding error occurs.
The Noise error flag is set when a noise is detected on the received frame’s Start bit.



Low-power modes

Mode	Description
Run **Low power Run**	Active
Sleep **Low power Sleep**	Active Peripheral interrupts cause the device to exit Sleep mode.
Stop 0 **Stop 1**	The content of the LPUART registers is kept. The LPUART is able to wake up the MCU from Stop mode when the LPUART clock is set to HSI or LSE.
Standby **Shutdown**	Powered-down The peripheral must be reinitialized after exiting Standby or Shutdown mode.

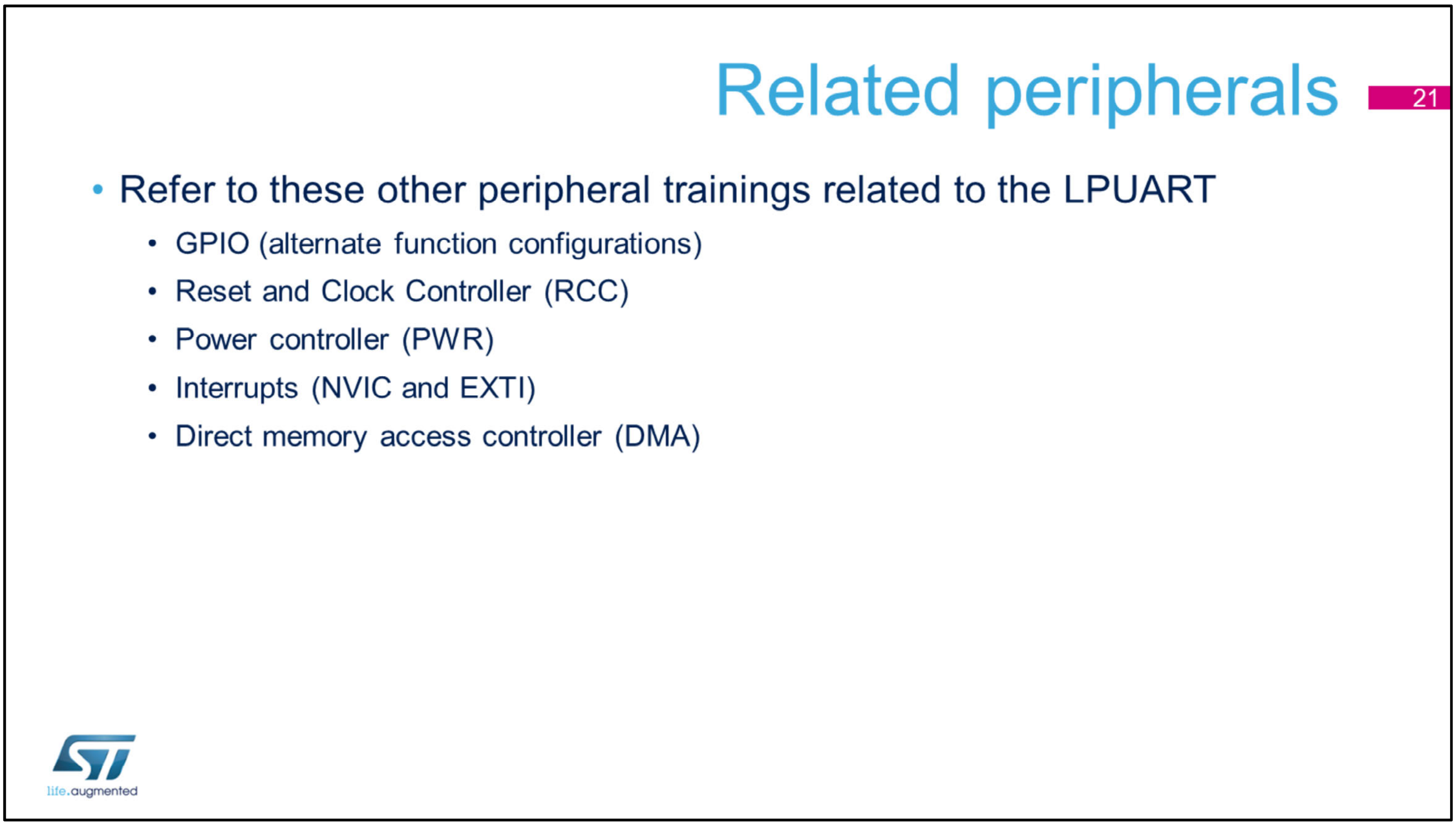
The LPUART peripheral is active in Run, Sleep and low-power modes. The LPUART interrupts cause the device to exit Sleep and Low-power Sleep modes.
The LPUART is able to wake up the MCU from Stop 0 and Stop 1 modes when the LPUART clock is set to HSI or LSE.
In Standby and shutdown modes, the peripheral is in power-down, and it must be reinitialized after exiting Standby or Shutdown mode.



STM32G0 LPUART/USART features

U(S)ART/LPUART features	USART	LPUART
Hardware flow control for modem	x	x
Multi-processor communication	x	x
Synchronous mode	x	-
Smartcard mode	x	-
Single-wire half-duplex communication	x	x
IrDA SIR ENDEC	x	-
LIN mode	x	-
Dual clock domain and wakeup from stop mode	x	x
Receiver timeout	x	-
Modbus communication	x	-
Auto-baudrate detection	x	-
Driver enable	x	x
Tx/Rx FIFO	x (size 8)	x (size 8)

The STM32G0 devices embed a single LPUART instance. Compared to the USART, the LPUART doesn’t support Synchronous, Smartcard, IrDA and LIN modes. It does not support the Receiver timeout, modbus communication and the auto-baudrate detection features as well.



Related peripherals

- Refer to these other peripheral trainings related to the LPUART
  - GPIO (alternate function configurations)
  - Reset and Clock Controller (RCC)
  - Power controller (PWR)
  - Interrupts (NVIC and EXTI)
  - Direct memory access controller (DMA)

This is a list of peripherals related to the LPUART. Please refer to these peripheral trainings for more information if needed.

- General-purpose input/output
- Reset and clock controller
- Power controller
- Interrupts controller
- Direct memory access controller.