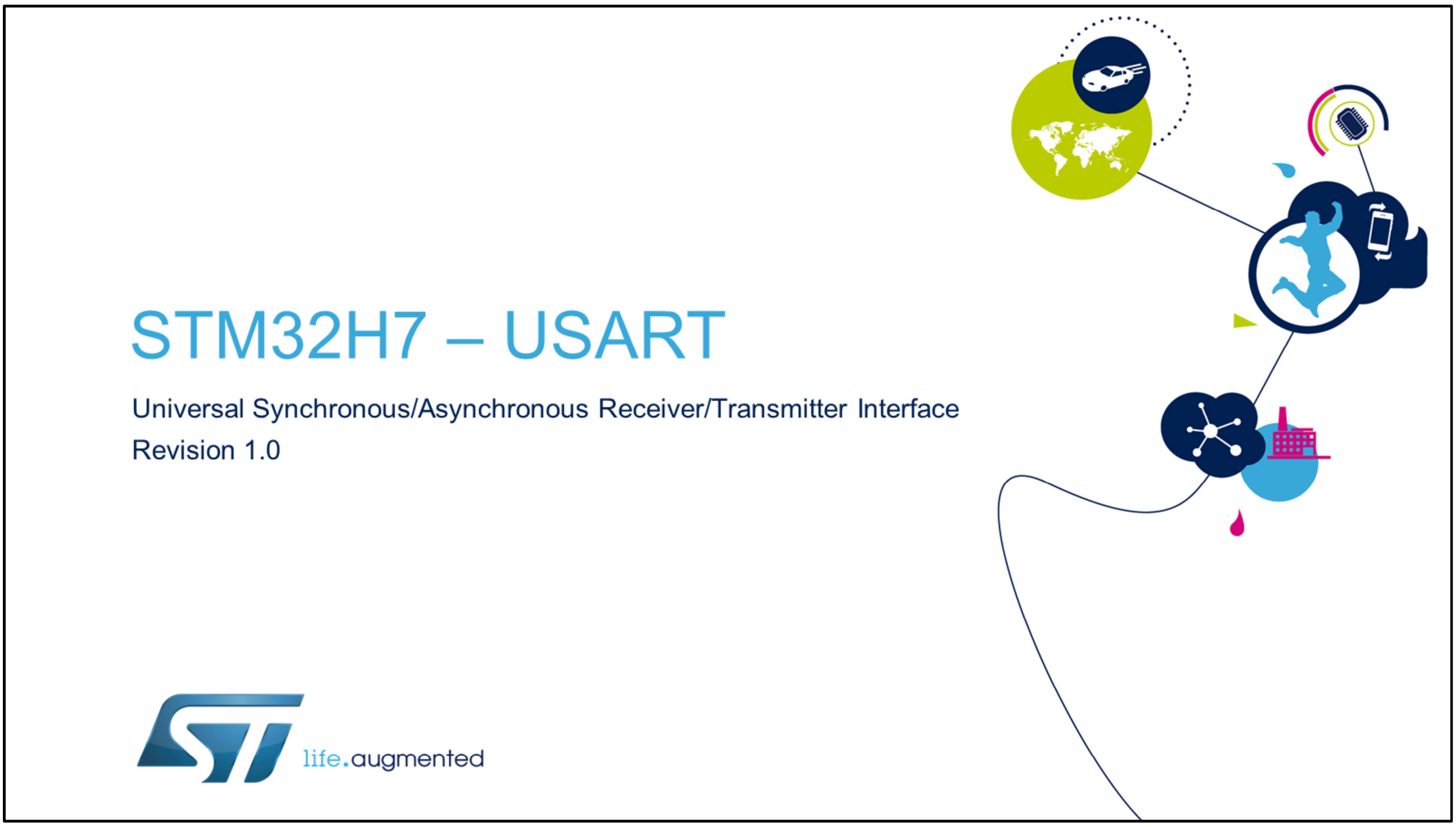
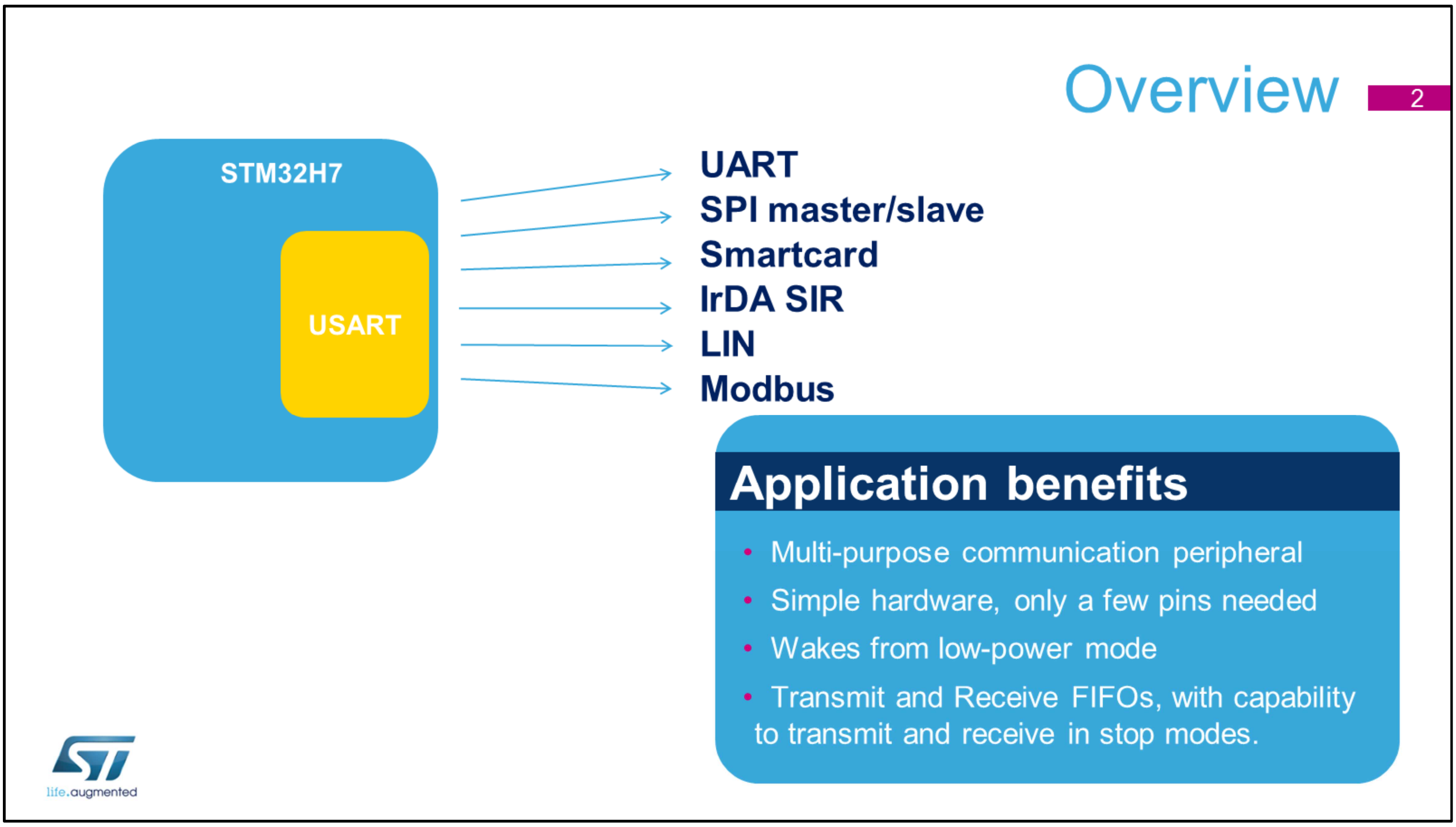


Hello, and welcome to this presentation of the STM32 Universal Synchronous/Asynchronous Receiver/Transmitter Interface.
It covers the main features of this USART interface, which is widely used for serial communications in embedded systems.



The USART is a very flexible serial interface that supports:

- Asynchronous UART communication,
- SPI (serial peripheral interface) master mode, and
- LIN (local interconnect network) mode.

It can also interface with ISO/IEC 7816 smartcards and IrDA devices.

It also provides certain features that are useful when implementing Modbus communications.

Applications making use of the USART benefit from the easy and inexpensive connection between devices, which only requires a few pins.

In addition, the USART 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 (1/2)

- Fully programmable serial interface
  - Data can be 7, 8 or 9 bits
  - Even, odd and no-parity
  - 0.5, 1, 1.5 and 2 stop bits
  - Programmable data order with MSB or LSB first
  - Programmable baudrate generator
  - Configurable oversampling method by 16 or by 8
- Two internal FIFOs for transmit and receive data
- Supports RS-232 and RS-485 hardware flow control
- Dual clock domain allowing:
  - UART functionality and wakeup from low power mode
  - Baudrate programming independent of PCLK changes

life.augmented

The USART is a fully programmable serial interface featuring the following configurable parameters:

- data length,
- parity,
- number of stop bits,
- data order,
- baud rate generator
- and a configurable oversampling mode by 8 or by 16.

The USART can operate in FIFO mode and it comes with Transmit and Receive FIFOs.
You also have the option to use basic RS-232 flow control with CTS (Clear To Send) and RTS (Request To Send) signals.
The RS-485 DE (Driver Enable) signal is also supported.
The USART supports a dual clock domain allowing wakeup from Stop mode and baud rate programming

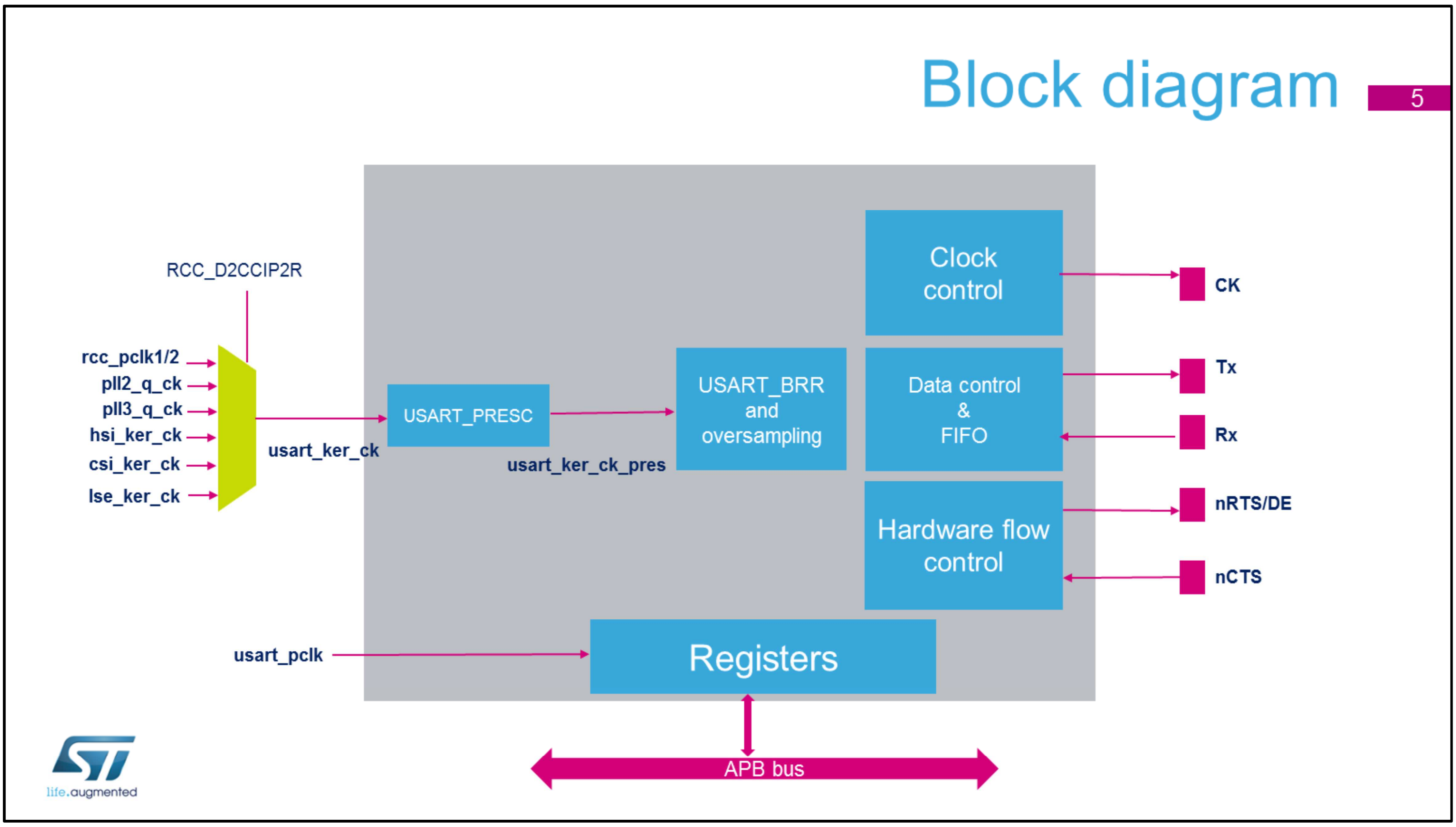
independent of the peripheral clock (PCLK).
This also allows the peripheral clock to be throttled along with the core clock without disrupting communications.

Key features (2/2)

- Multi-processor communication
- Single-wire half-duplex communication
- Auto baudrate detection
- Receiver timeout feature
- Supports also
  - LIN mode
  - Synchronous mode (Master mode)
  - IrDA SIR encoder decoder
  - Smartcard (ISO/IEC 7816 T=0 and T=1 protocols)
  - Basics to implement Modbus/RTU and Modbus/ASCII protocols

life.augmented

The USART features a multi-processor mode which allows the USART to remain idle when it is not addressed.
In addition to full-duplex communication, single-wire half-duplex mode is also supported.
The USART also offers many other features including auto baud rate detection, receiver timeout and supports several modes which will be described later in the presentation.



This is the USART block diagram.
The USART clock source (usart_ker_ck) can be selected from several sources: The peripheral clock (APB clock), the PLL2_q, the PLL3_q, the high-speed internal RC oscillator or the Low-power Internal oscillator CSI or the low-speed external 32.768 kHz crystal oscillator.
The USART clock source can be divided by a programmable factor in the USART_PRESC register.
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.
The clock output (SK) is dual purpose:

- When the USART is used in Synchronous Master/Slave mode, the clock provided to the slave device is output/input on the CK pin.

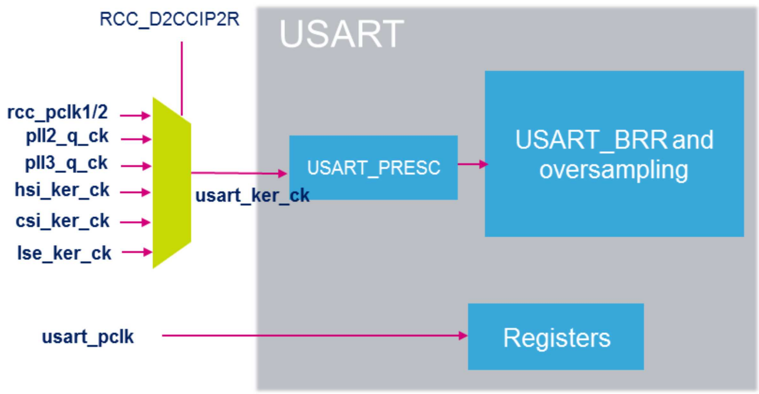
- When the USART is used in Smartcard mode, the clock provided to the card is output on the CK pin.

Dual clock domain

6

Baudrate programming independent from PCLK reprogramming

- Flexible clocking scheme through selectable clock sources
  - rcc_pclk1/2 (default)
  - pll2_q_ck
  - pll3_q_ck
  - hsi_ker_ck
  - csi_ker_ck
  - lse_ker_ck
- Access to registers is always at peripheral bus speed.



life.augmented

The USART has a flexible clocking scheme.
Its clock source can be selected in the RCC, and can be either the peripheral clock (APB clock), the PLL2_q, the PLL3_q, the HSI or CSI or the LSE clock.
The USART clock source can be divided by a programmable factor in the USART_PRESC register.
The registers are accessed through the APB bus, and the kernel is clocked with usart_ker_ck (prescaled or not) which is independent from the APB clock.

Oversampling

Different user configurable oversampling techniques

- Oversampling choice affects speed and framing tolerance:

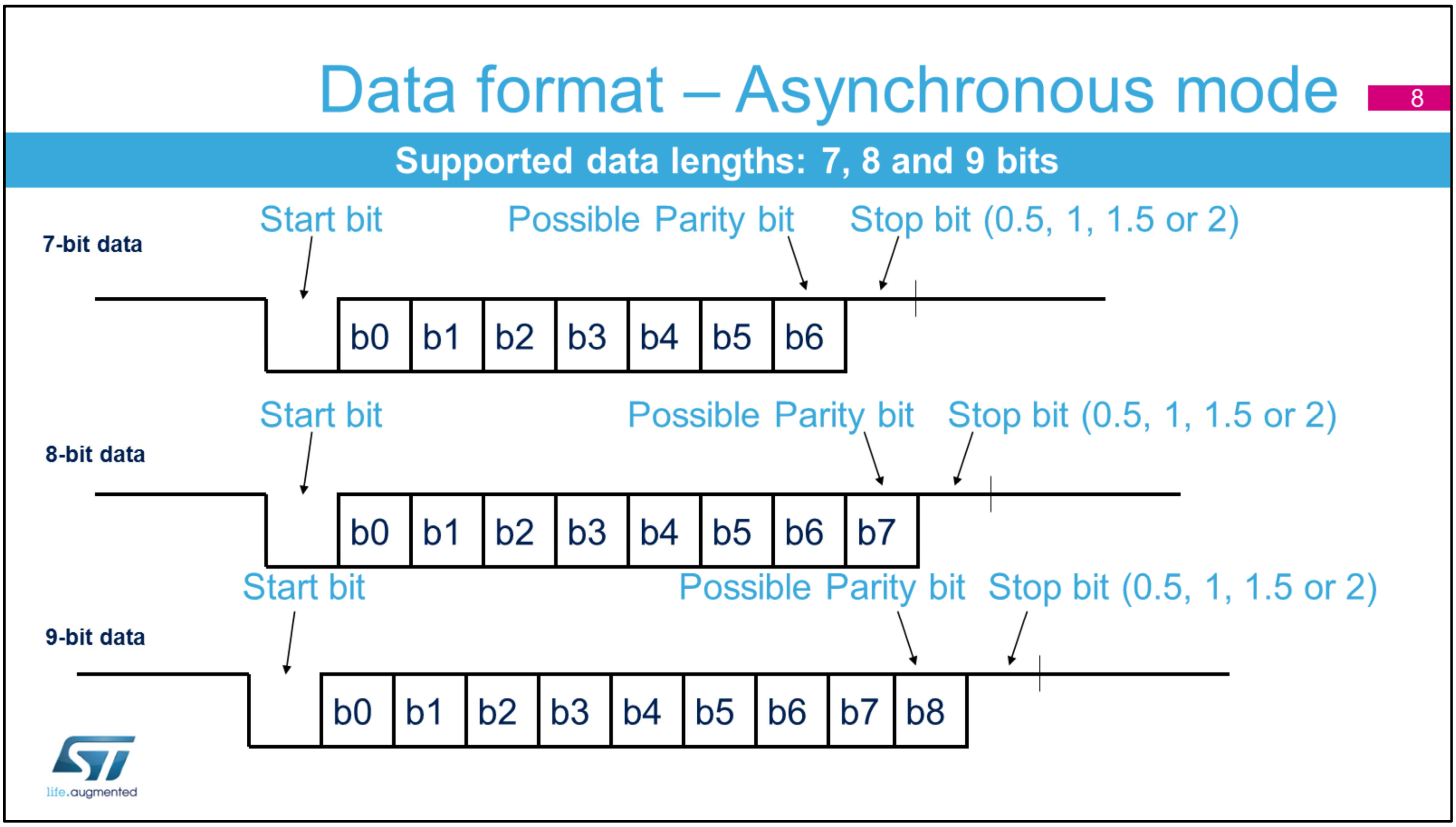
	Oversampling by 8	Oversampling by 16
Benefits	Achieve the maximum speed usart_ker_ck/8	Maximum receiver tolerance to clock deviations is increased.
Drawbacks	Maximum receiver tolerance to clock deviations is reduced.	Maximum speed is limited to usart_ker_ck/16.

- Maximum baud rate depends on selected clock and oversampling: It is 12.5 Mbaud when clock source is at 100 MHz and Oversampling by 8 is configured.

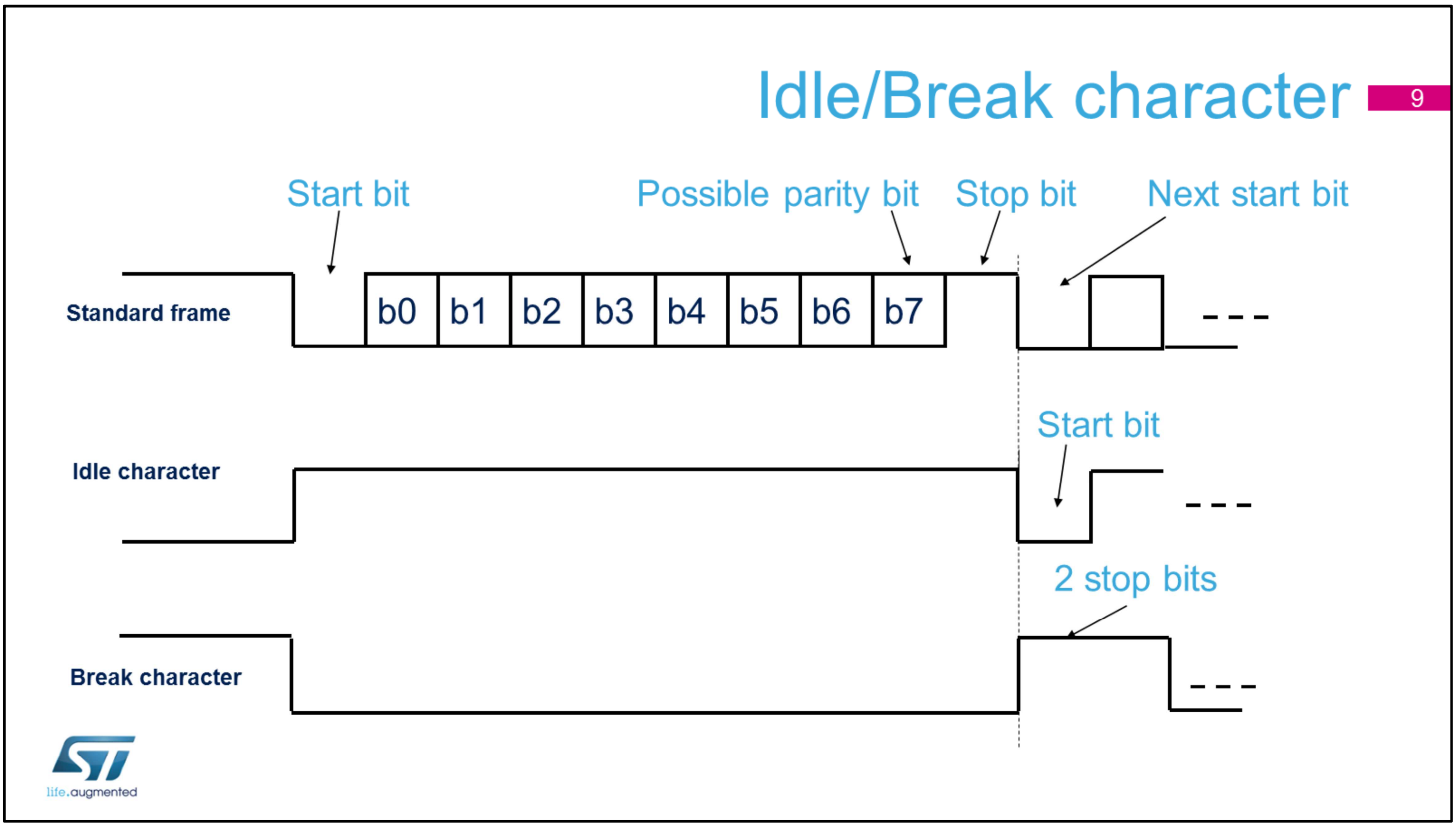
life.augmented

The USART receiver implements different user-configurable oversampling techniques for data recovery by discriminating between valid incoming data and noise.
This allows a trade-off between the maximum communication speed and noise/clock inaccuracy immunity.
Select oversampling by 8 to achieve higher speed (up to usart_ker_ck_pres/8) where usart_ker_ck_pres is the USART clock source frequency. In this case the maximum receiver tolerance to clock deviations is reduced.
Select Oversampling by 16 (OVER8 = 0) to increase the tolerance of the receiver to clock deviations. In this case, the maximum speed is limited to usart_ker_ck_pres/16.

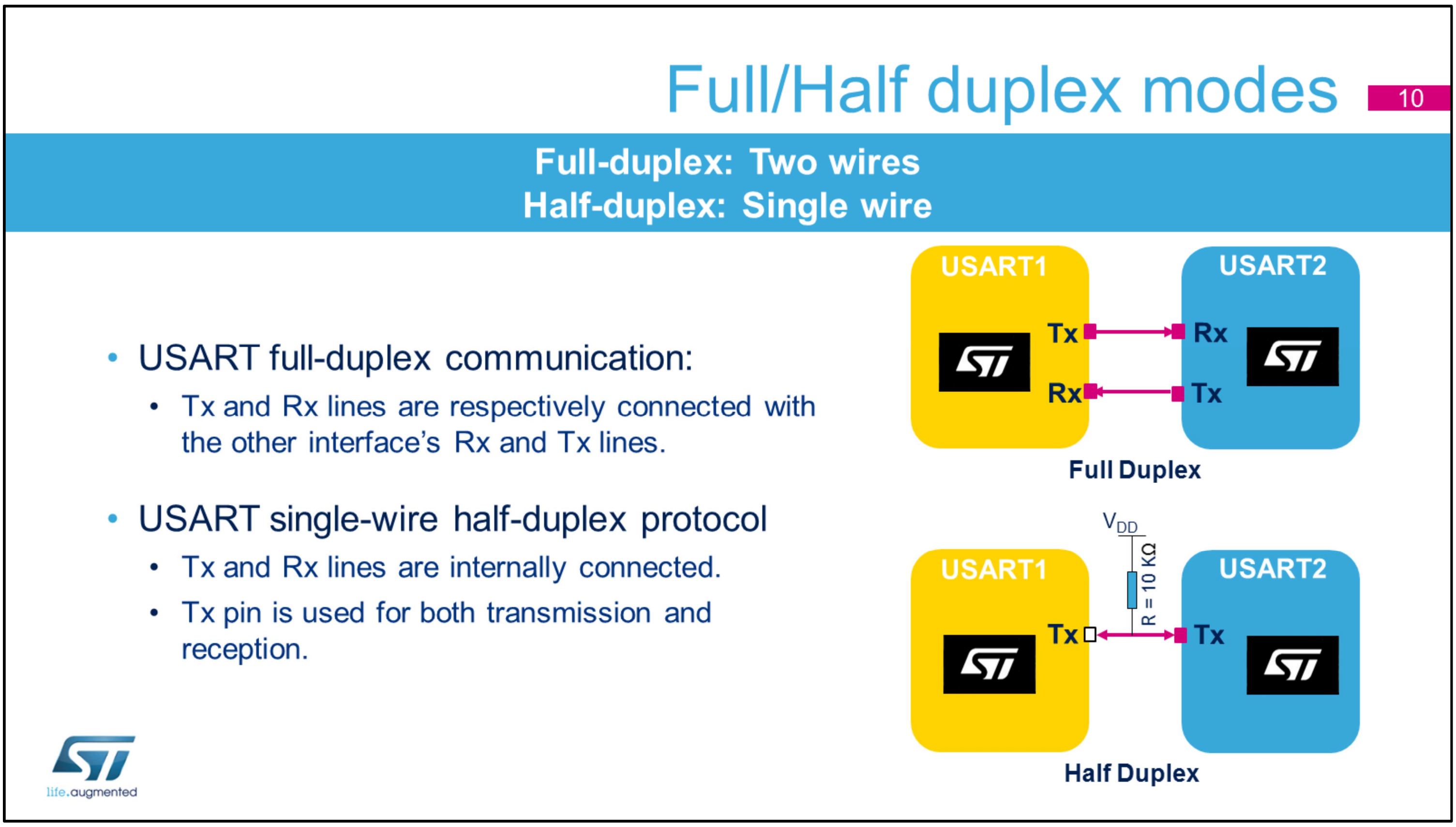
The maximum baud rate that can be reached is 12.5 Mbaud when the clock source is at 100 MHz and Oversampling by 8 is configured.
With other clock sources, and/or higher oversampling ratio, the maximum speed is limited.



The frame format used in Asynchronous mode consists of a set of data bits in addition to bits for synchronization and optionally a parity bit for error checking.
The USART supports 7-, 8- or 9-bit data lengths.
A frame starts with one start-bit, where the line is driven low for a one-bit period. This signals the start of a frame, and is used for synchronization.
The start bit is followed by 7, 8 or 9 data bits. If Parity Control is enabled, the parity bit is transmitted as the last data bit and is included in the data length count.
Finally, a number of stop-bits (0, 1, 1.5 or 2), where the line is driven high, ends the frame.



The standard frame was described in the previous slide. This slide shows an example of 8-bit data frames 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).
A Break character is interpreted on receiving “0”s for a frame period. At the end of the break frame, 2 stop bits are inserted.



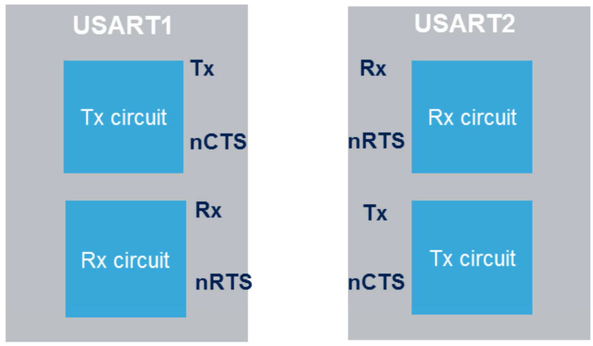
The USART supports full-duplex communication where Tx and Rx lines are respectively connected with the other interface's Rx and Tx lines.
The USART can be configured to follow a single-wire half-duplex protocol where the Tx and Rx lines are internally connected.
In this communication mode, only the Tx 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 modes.
This means that the I/O must be configured so that the Tx pin is configured as an alternate function open-drain with an external pull-up.

RS-232 hardware flow control

11

Hardware handshaking to avoid data underrun/overrun

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



life.augmented

In RS-232 communication, it is possible to control the serial data flow between 2 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. The following figure shows how to connect 2 devices in this mode. The idea is to prevent dropped bytes or conflicts in case of half-duplex communication. Both signals are active low.

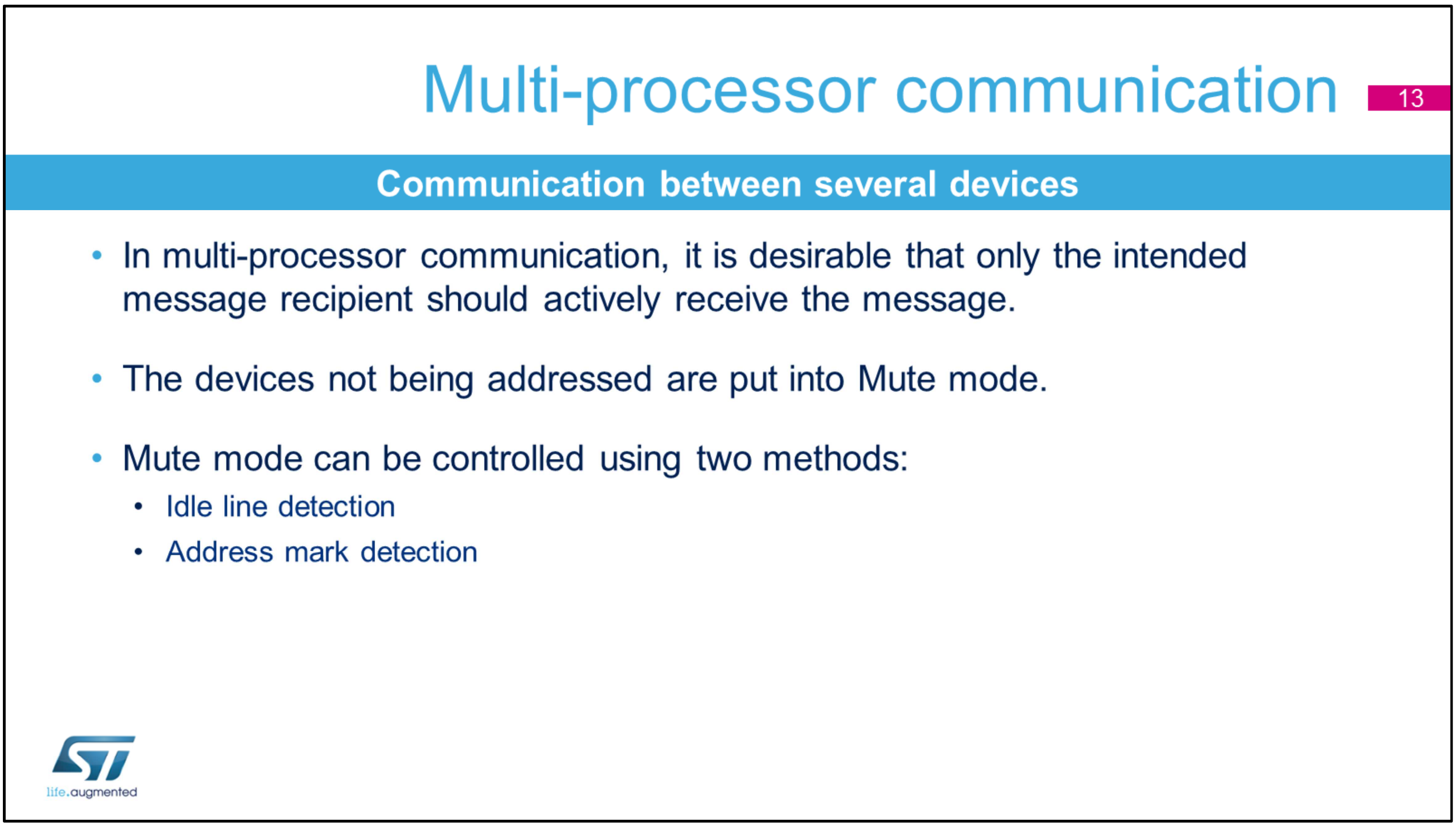
RS-485 hardware flow control

Hardware handshaking

- Useful in a half-duplex system where the master needs to generate a direction signal to control the transceiver (Physical Layer (PHY)). This signal informs the PHY if it must act in Transmit 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.

life.augmented

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 is used: the Driver Enable pin is used to activate the external transceiver control.
DE shares the pin with nRTS.



To simplify communication between multiple processors, the USART supports a multi-processor mode.

In multi-processor communication, it is desirable that only the intended message recipient should actively receive the message.

The devices not being addressed are put into Mute mode.

The USART can enter or exit from Mute mode using one of two methods:

- Idle line detection
- Address mark detection

Synchronous mode

14

USART used as SPI Master/Slave

- Full-duplex or simplex synchronous communication mode:
  - SPI master/slave modes
  - Programmable clock polarity (CPOL) and phase (CPHA)
  - Programmable data order with MSB or LSB first
  - Clock output/input on CK pin.
  - No clock pulses during the Start and Stop bits
  - Transmit underrun error (Only in SPI slave mode).
  - NSS management (Software or hardware management) (Only in SPI slave mode).

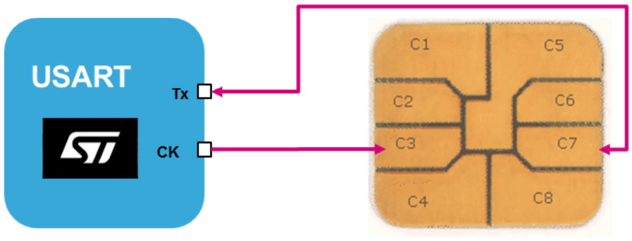
life.augmented

The USART can also communicate synchronously.
It can operate as a SPI in Master or Slave mode with programmable clock polarity (CPOL) and phase (CPHA) and programmable data order with MSB or LSB first.
The clock is output (in case of Master mode) or input (in case of Slave mode) on the CK pin. No clock pulses are provided during the start and stop bits.
When the USART is configured in SPI slave mode, it supports the Transmit underrun error and the NSS hardware or software management.

ISO/IEC 7816 mode

USART interface for Smartcards and Security Access Module

- Half-duplex mode
- Clock provided to Smartcard on CK pin.
- Programmable clock prescaler to guarantee a wide range of clock inputs
- ISO/IEC 7816 T=0 and T=1 protocols supported
- Both direct and inverse conventions are available



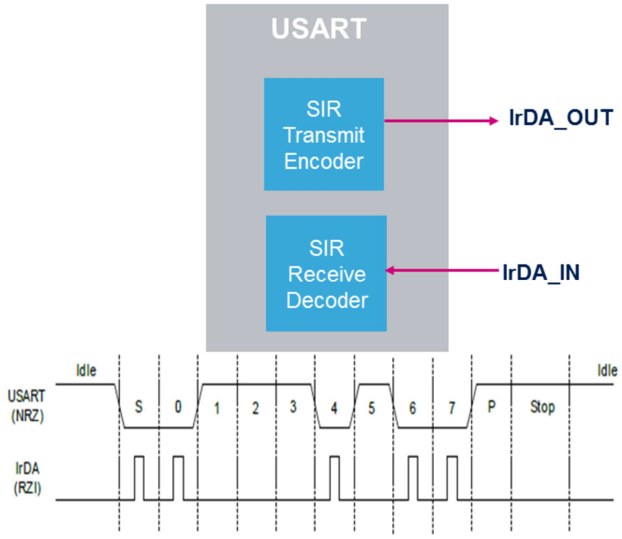
life.augmented

The USART can be used in Smartcard mode, based on a half-duplex communication.
The clock is output to the Smartcard on the CK pin.
It supports the T=0 protocol and provides many features allowing support for T=1.
Both direct and inverse conventions are supported directly by hardware.

IrDA SIR Encoder Decoder

USART interface for infrared wireless connection

- Half-duplex communication
- The data from and to the USART is represented in a NRZ (Non Return to Zero) format
- For IrDA, the required format is RZI (Return to Zero Inverted)
- The SIR Tx Encoder modulates the signal before it leaves the USART. Similarly, the input signal is demodulated in the SIR Rx Decoder
- Maximum bit rate is 115.2 Kbits/s
- The pulse width is 3/16 bit duration in normal mode



life.augmented

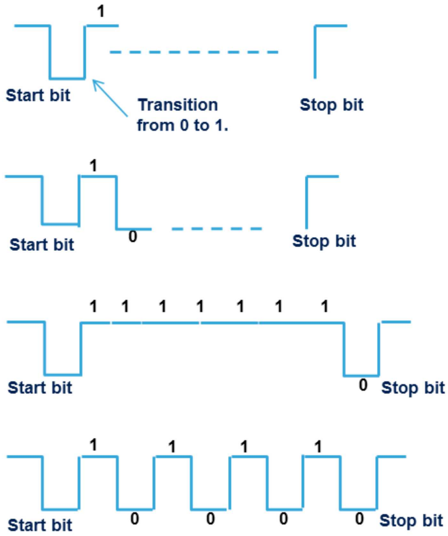
The USART supports IrDA specifications which is a half-duplex communication protocol.
The data from and to the USART is represented in a NRZ (Non Return to Zero) format where the signal value is at the same level through the entire bit period. For IrDA, the required format is RZI (Return to Zero Inverted) where a “1” is signaled by holding the line low, and a “0” is signaled by a short high pulse.
The SIR Transmit encoder modulates the Non Return to Zero (NRZ) transmit bit stream output from the USART. The SIR receive decoder demodulates the return-to-zero bit stream from the infrared detector and outputs the received NRZ serial bit stream to the USART.

The USART only supports bit rates up to 115.2 Kbits/s for the SIR ENDEC.
In normal mode, the transmitted pulse width is specified as 3/16 of a bit period.

Auto baudrate detection

Automatic baudrate configuration - UART receiver

- The USART is able to automatically determine the baudrate based on the reception of one character
- The received character can be:
  - Any character starting with a bit at ‘1’
  - Any character starting with a 10xx pattern
  - 0x7F
  - 0x55



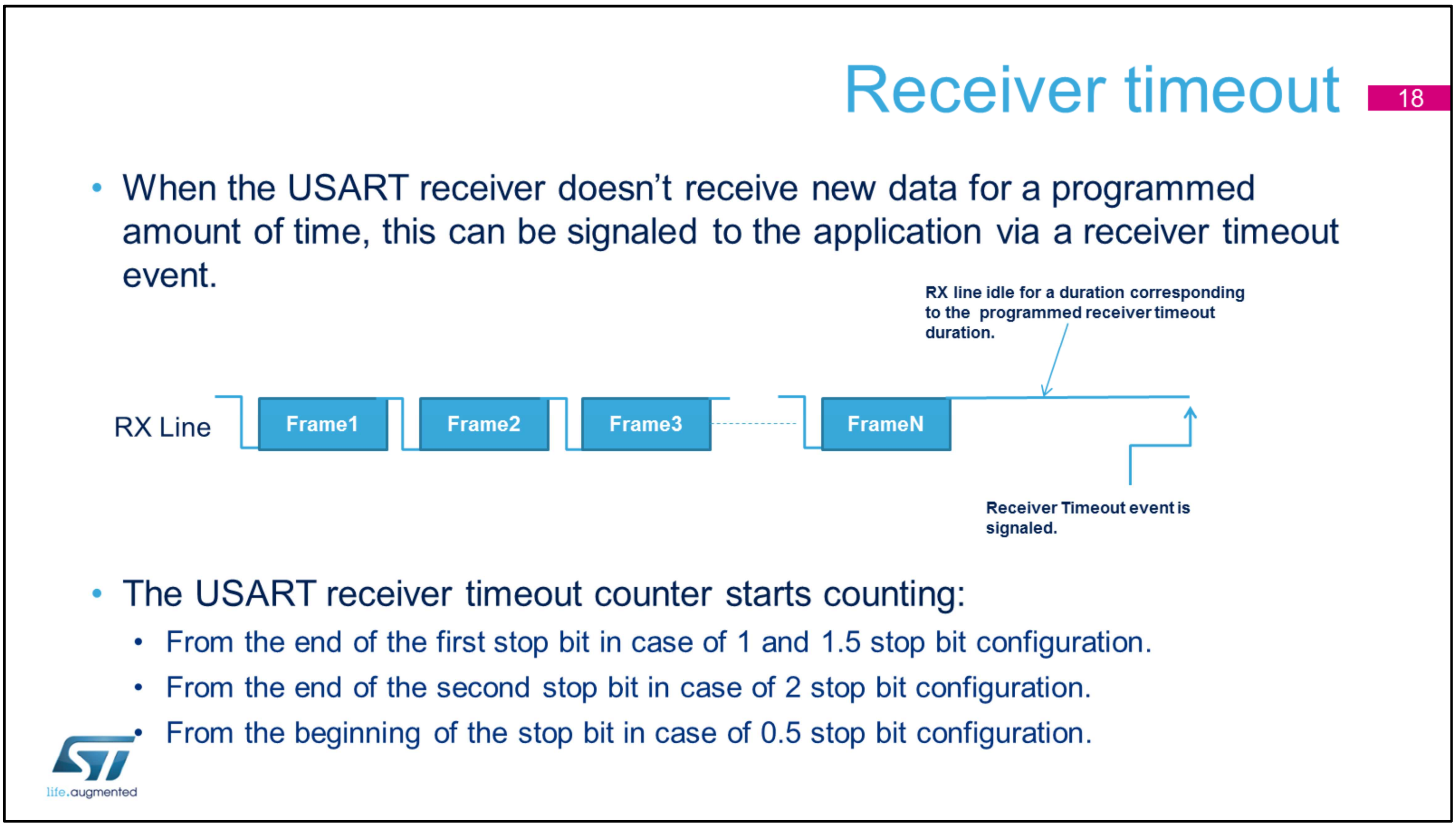
life.augmented

The USART receiver is able to detect and automatically configure the baud rate based on the reception of one character.

The received character can be:

- Any character starting with a bit at 1. In this case, the USART measures the duration of the start bit (from falling edge to rising edge).
- Any character starting with a 10xx pattern. In this case, the USART measures the duration of the start and of the 1st data bit. The duration is measured from falling edge to falling edge, ensuring better accuracy in the case of slow signal slopes.
- 0x7F character frame. In this case, the baud rate is updated first at the end of the start bit then at the end of bit 6.

- A 0x55 character frame. In this case, the baud rate is updated first at the end of the start bit, then at the end of bit 0 bit and finally at the end of bit 6. In parallel, another check is performed for each intermediate transition of the RX line.



The USART supports a receiver timeout feature. When the USART doesn't receive new data for a programmed amount of time, a receiver timeout event is signaled and an interrupt is generated if enabled.

The USART receiver timeout counter starts counting:

- From the end of the first stop bit in case of 1 and 1.5 stop bit configuration.
- From the end of the second stop bit in case of 2 stop bits configuration.
- From the beginning of the stop bit in case of 0.5 stop bit configuration.

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 16 data bits
- FIFO mode is not available in IrDA and LIN modes.
- FIFOs are in kernel clock domain → ability to receive or transmit even in Stop mode.
- TXFIFO and RXFIFO thresholds

life.augmented

The USART can operate in FIFO mode which is enabled/disabled by software. It is disabled by default.
The USART comes with a Transmit FIFO (TXFIFO) and a Receive FIFO (RXFIFO), each being 16 words deep.
When the IrDA and LIN modes are used, the FIFO mode is not supported.
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, mainly to avoid underrun/overrun issues while waking up from Stop mode.

Wake up from Stop mode

20

- The USART is able to wake up the MCU from Stop mode when the USART clock source is:
  - HSI
  - LSE
  - CSI
- 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, RXFIFO/TXFIFO reach the programmed threshold.

life.augmented

The USART is able to wake up the MCU from Stop mode when the USART clock source is the HSI, LSE or CSI 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:
  - o Receive FIFO full interrupt
  - o Transmit FIFO empty interrupt
  - o Receive FIFO threshold interrupt
  - o Transmit FIFO threshold interrupt

Interrupts (1/2)

Interrupt event	Description
Transmit data register empty	Set when the Transmit Data register is empty.
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.
Idle line	Set when an idle line is detected.
Character match	Set when the received data corresponds to the programmed address.
Receiver timeout	Set when there is no activity on the Rx line for a duration equal to the programmed timeout.

life.augmented

Several events can provide an interrupt:

- The Transmit Data Register Empty flag is set when the Transmit Data register is empty and ready to be written.
- The Transmit Complete flag is set when the data transmission is complete and both data and shift registers are empty.
- The CTS flag is set when the nCTS input toggles.
- The Receive Data Register Not Empty flag is set when the Receive Data register contains data ready to be read.
- The Idle Line flag is set when an idle line is detected.
- The Character Match flag is set when the received data corresponds to the programmed address.
- The Receiver Timeout flag is set when there is no activity on the Rx line for a programmed duration.

Interrupts (2/2)

Interrupt event	Description
End of block	Set when a complete block was received.
Wakeup from stop mode	Set when the wakeup event is verified.
LIN break	Set when a LIN break frame is detected.
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.

life.augmented

The End of Block flag is set when a complete block is received.
The Wakeup from Stop Mode flag is set when the wakeup event is verified.
The LIN break flag is set when a LIN break frame is detected.
The Transmit FIFO not full flag is set when the Transmit FIFO is not full.
The Transmit FIFO empty flag is set when the Transmit FIFO is empty.
The Transmit FIFO threshold flag is set when programmed threshold is reached.
The Receive FIFO not empty flag is set when the Receive FIFO is not empty.
The Receive FIFO full flag is set when the Receive FIFO is full.
The Receive FIFO threshold flag is set when the programmed threshold is reached.

- 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.

life.augmented

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 also 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 framing error occurs.
Noise error	Set when a noise is detected on the received frame.
Auto-baudrate error	Set when the baudrate measurement failed.
Underrun error	Set when an underrun error occurs in synchronous slave mode.

life.augmented

Several errors flags can be generated:

- The Overrun error flag is set when an overrun error occurs.
- The Parity error flag is set when a parity error occurs.
- The Framing error flag is set when a framing error occurs.
- The Noise error flag is set when a noise is detected on the received frame.
- The Auto-baudrate error flag is set when the baudrate measurement failed.
- The Underrun error flag is set when an underrun error occurs in synchronous slave mode.

Low-power modes

Mode	escription
Run	Active.
Sleep	Active. Peripheral interrupts cause the device to exit Sleep mode.
Stop	The content of the USART registers is kept. The USART is able to wake up the MCU from Stop mode when the USART clock is set to HSI, LSE or CSI.
Standby	Powered-down. The peripheral must be reinitialized after exiting Standby mode.

life.augmented

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

STM32H7 USART instances features

USART features	USART1/2/3/6	UART4/5/7/8
Hardware flow control for modem	x	x
Multiprocessor communication	x	x
Synchronous mode	x	-
Smartcard mode	x	-
Single-wire half-duplex communication	x	x
IrDA SIR Encoder/Decoder	x	x
LIN mode	x	x
Dual clock domain and wakeup from Stop mode	x	x
Receiver timeout	x	x
Modbus communication	x	x
Auto baudrate detection	x	x
Driver enable	x	x
Tx/Rx FIFO	X (size 16)	X (size 16)

life.augmented

The STM32H7 devices embed eight USART instances:

- USART 1, 2, 3 and 6 have a full set of features.
- Instances 4, 5, 7 and 8 do not support Synchronous and Smartcard modes

Related peripherals

- Refer to these trainings linked to this peripheral for more information:
  - GPIO (Alternate function configuration)
  - Reset and clock controller (RCC)
  - Power controller (PWR)
  - Interrupts (NVIC and EXTI)
  - Direct memory access controller (DMA)

life.augmented

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

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