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.
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 baud rates 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.
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

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.
Here is the LPUART block diagram. The LPUART clock “lpuart_ker_ck” can be selected from several sources: peripheral clock (APB clock), the system clock, the high-speed internal RC oscillator (HSI16) or the low-speed external 32.768 kHz crystal oscillator (LSE).

Tx and Rx pins are used for data transmission and reception.
Pins nCTS and nRTS are used for RS-232 hardware flow control.
The Driver Enable (DE) signal, which is available on the same I/O as nRTS, is used in RS-485 mode.
The LPUART has a flexible clocking scheme. Its clock source can be selected in the RCC, and can be either the peripheral clock (APB clock), the system clock, the high-speed internal RC oscillator (HSI16) or the low-speed external 32.768 kHz crystal oscillator (LSE). The LPUART clock source can be divided by a programmable factor in LPUART_PRESC register. The registers are accessed through the APB bus, and the kernel is clocked with lpuart_ker_ck (prescaled or not) and which is independent from the APB clock. The maximum baudrate that can be reached is 9600 baud when the clock source is LSE, and 21 Mbaud when the clock source is at 64 MHz.
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.
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.
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.
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.
For serial half-duplex communication protocols like RS-485, 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.
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
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 16 words 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.
The LPUART is able to wake up the MCU from STOP mode when the LPUART clock source is:

- High-speed internal RC oscillator (HSI)
- 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.
This table lists the LPUART events that can generate an interrupt.

<table>
<thead>
<tr>
<th>Interrupt event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmit Data register empty</td>
<td>Set when the Transmit Data register is empty and ready to be written.</td>
</tr>
<tr>
<td>Transmit complete</td>
<td>Set when the data transmission is complete and both data and shift registers are empty.</td>
</tr>
<tr>
<td>CTS</td>
<td>Set when the nCTS input toggles.</td>
</tr>
<tr>
<td>Receive data register Not Empty</td>
<td>Set when the Receive Data register contains data ready to be read.</td>
</tr>
<tr>
<td>Idle Line</td>
<td>Set when an idle line is detected.</td>
</tr>
<tr>
<td>Character match</td>
<td>Set when the received data corresponds to the programmed address.</td>
</tr>
<tr>
<td>Wakeup from stop mode</td>
<td>Set when a wakeup event (Start bit or address match or any received data) is verified.</td>
</tr>
<tr>
<td>Interrupt event</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>Transmit RFO not full</td>
<td>Set when the Transmit RFO is not full.</td>
</tr>
<tr>
<td>Transmit RFO empty</td>
<td>Set when the Transmit RFO is empty.</td>
</tr>
<tr>
<td>Transmit RFO threshold</td>
<td>Set when programmed threshold is reached.</td>
</tr>
<tr>
<td>Receive RFO not empty</td>
<td>Set when the Receive RFO is not empty.</td>
</tr>
<tr>
<td>Receive RFO full</td>
<td>Set when the Receive RFO is full.</td>
</tr>
<tr>
<td>Receive RFO threshold</td>
<td>Set when the programmed threshold is reached.</td>
</tr>
</tbody>
</table>

This table lists the FIFO events interrupts when the 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.
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.

<table>
<thead>
<tr>
<th>Interrupt event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overrun error</td>
<td>Set when an overrun error occurs.</td>
</tr>
<tr>
<td>Parity error</td>
<td>Set when a parity error occurs.</td>
</tr>
<tr>
<td>Framing error</td>
<td>Set when a de-synchronization or excessive noise is detected.</td>
</tr>
<tr>
<td>Noise error</td>
<td>Set when noise is detected on the received frame’s Start bit.</td>
</tr>
</tbody>
</table>
The LPUART peripheral is active in Run mode. The LPUART interrupts cause the device to exit Sleep mode. The LPUART is able to wake up the MCU from Stop mode when the LPUART clock is set to HSI or LSE clock.

In Standby mode, the peripheral is in power-down, and it must be reinitialized after exiting Standby or Shutdown mode.
The STM32WB 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.
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 (DMA)