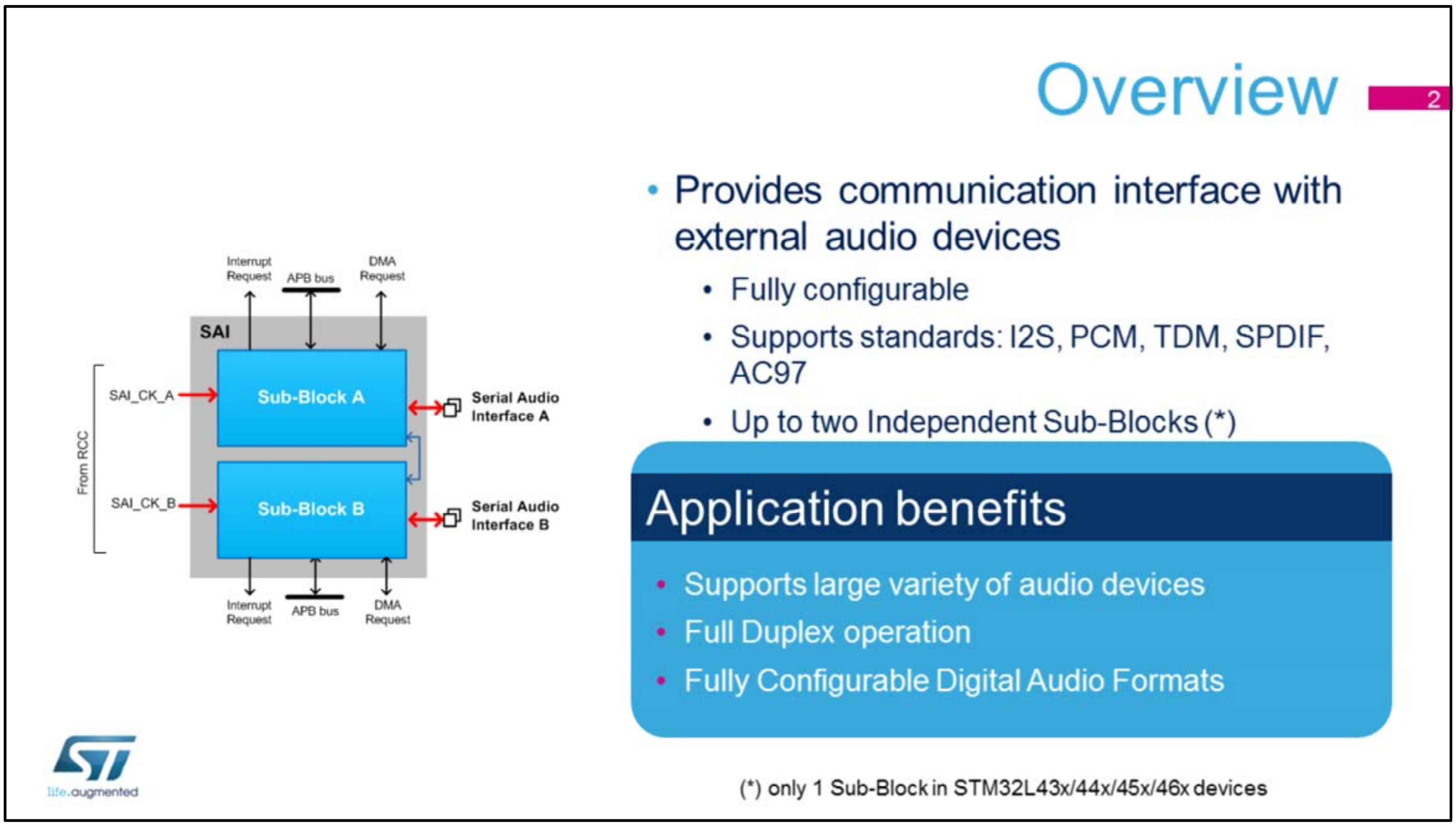
STM32L4 - SAI

Serial Audio Interface

Revision 3.1

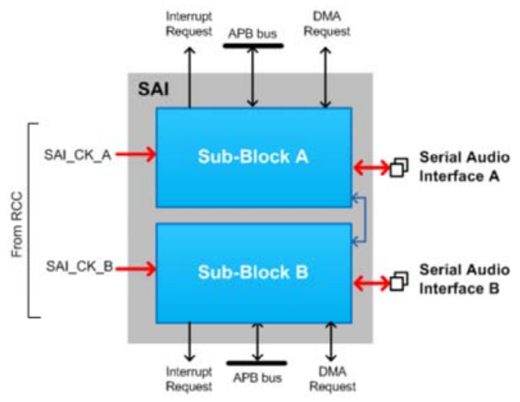
life.augmented

Hello, and welcome to this presentation of the STM32 Serial Audio Interface.
I will present the features of this interface, which is used to connect external audio devices



Overview

- Provides communication interface with external audio devices
  - Fully configurable
  - Supports standards: I2S, PCM, TDM, SPDIF, AC97
  - Up to two Independent Sub-Blocks (*)

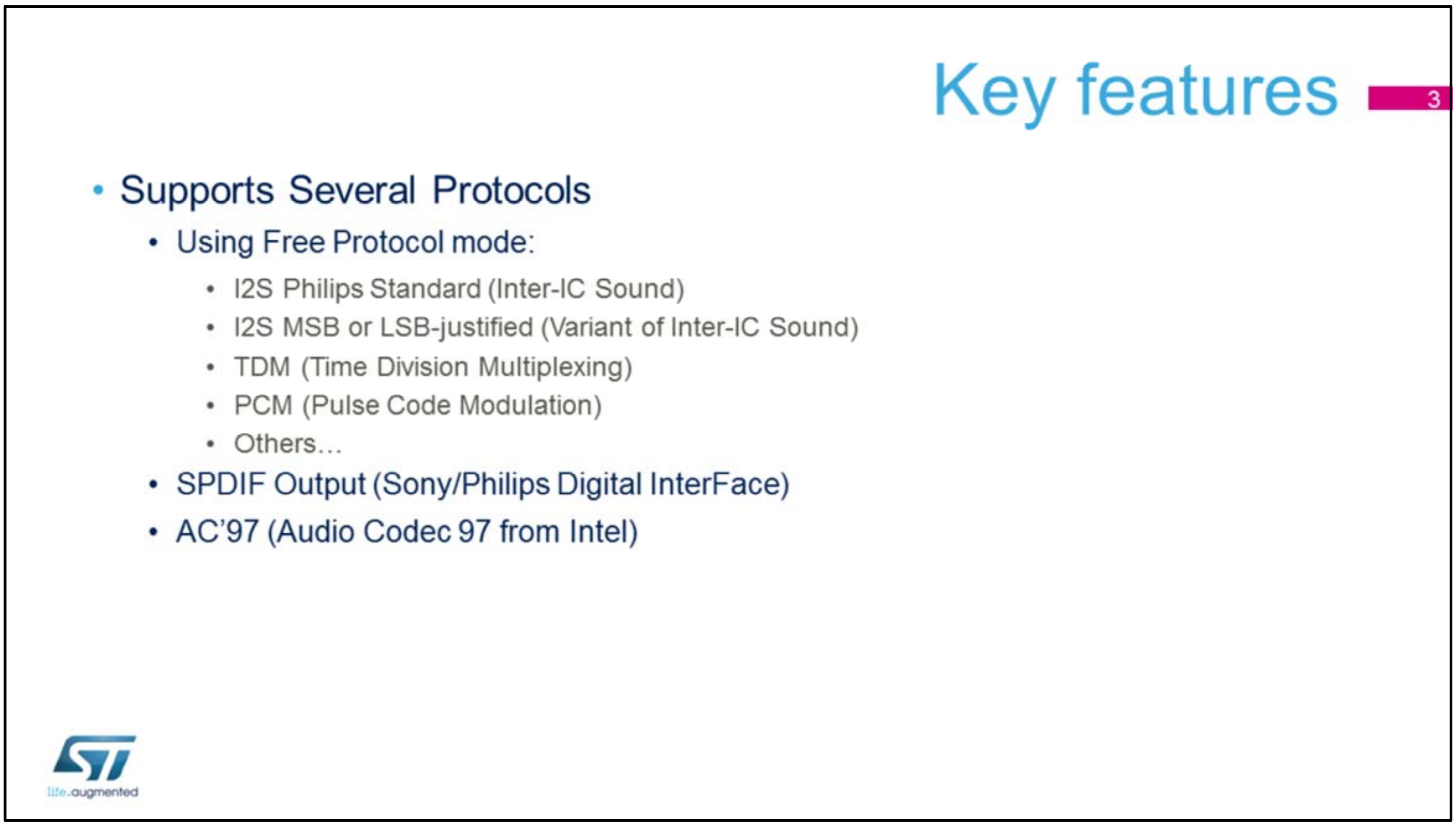


Application benefits

- Supports large variety of audio devices
- Full Duplex operation
- Fully Configurable Digital Audio Formats

(*) only 1 Sub-Block in STM32L43x/44x/45x/46x devices

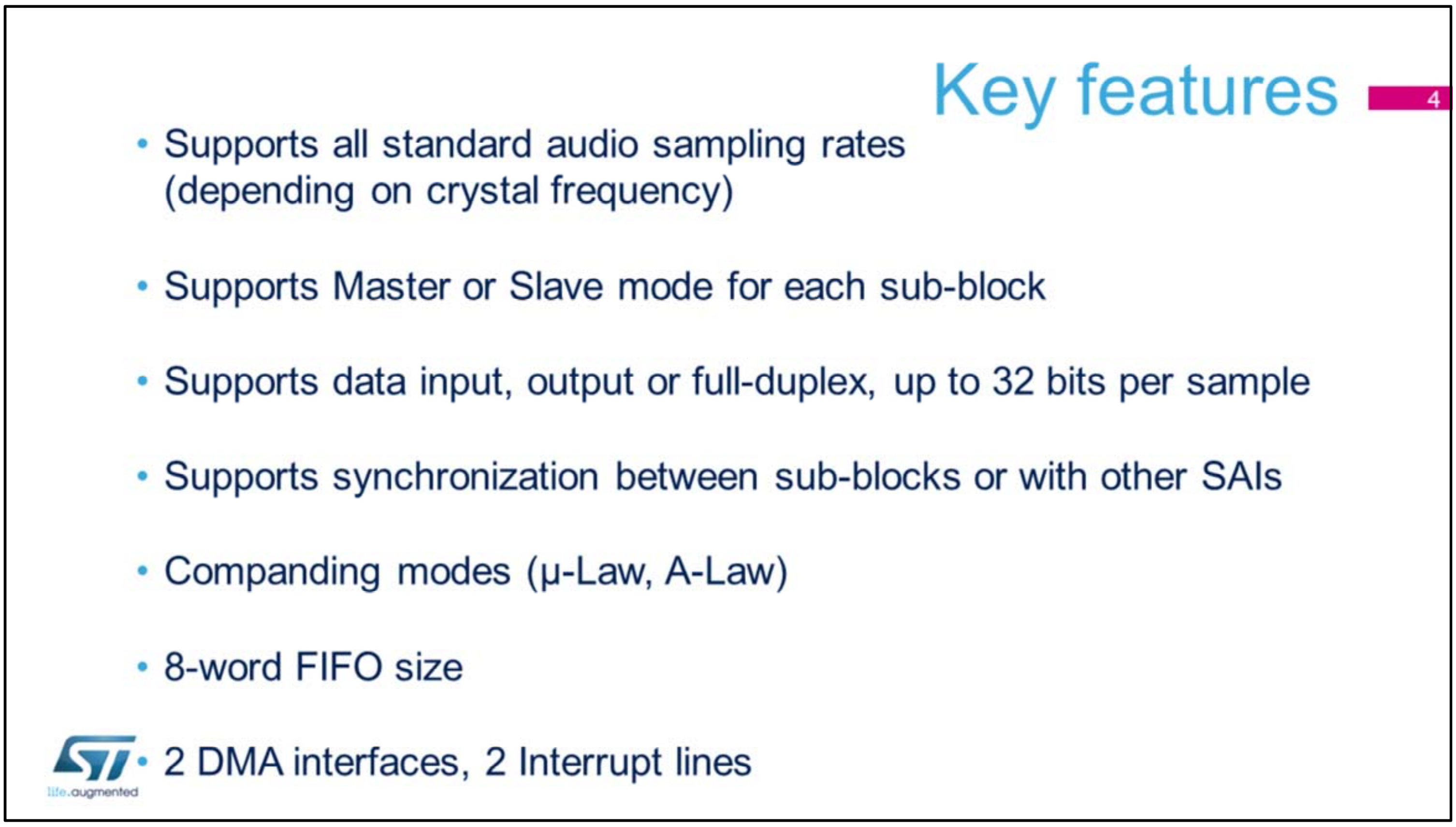
The Serial Audio Interface (SAI), is integrated in STM32 products to provide an interface, for communicating with external audio devices such as amplifiers, ADCs, DACs, audio codecs, and audio processors.
This interface is fully configurable, supporting the most digital audio standards, allowing easy connection to any existing audio devices.
Due to the internal synchronization features of the SAI , the required IO pins are reduced to a minimum.



Key features

- Supports Several Protocols
  - Using Free Protocol mode:
    - I2S Philips Standard (Inter-IC Sound)
    - I2S MSB or LSB-justified (Variant of Inter-IC Sound)
    - TDM (Time Division Multiplexing)
    - PCM (Pulse Code Modulation)
    - Others…
  - SPDIF Output (Sony/Philips Digital InterFace)
  - AC'97 (Audio Codec 97 from Intel)

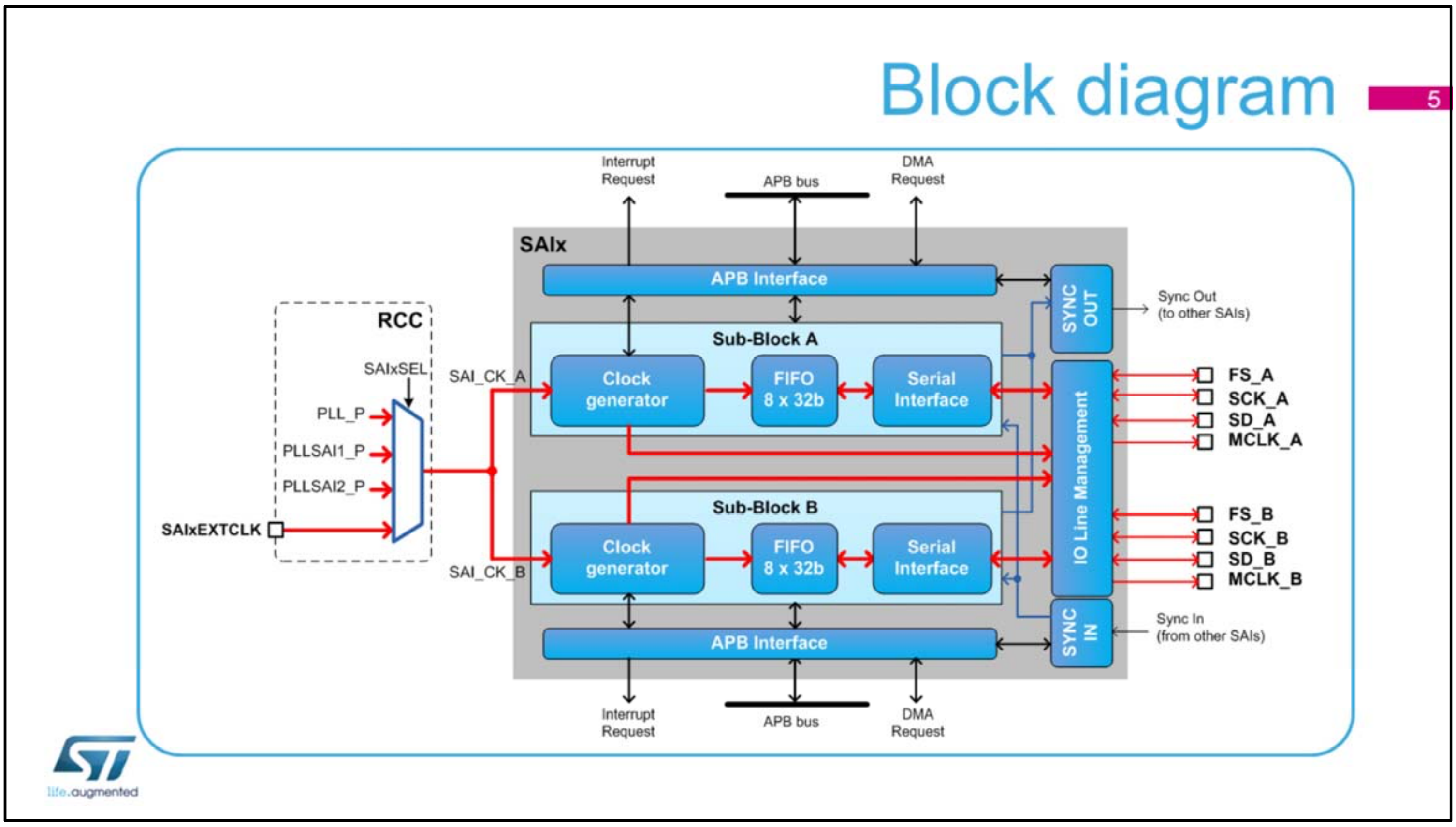
The SAI can be programmed in three different modes:
The Free protocol mode allows the SAI to support standards such as I2S, PCM, TDM… Due to its flexibility, it is possible to customize the serial interface if needed.
The SPDIF protocol mode, allows the SAI to transmit audio samples using the IEC60958 standard.
The AC97 protocol is also supported by the SAI.



Key features

- Supports all standard audio sampling rates (depending on crystal frequency)
- Supports Master or Slave mode for each sub-block
- Supports data input, output or full-duplex, up to 32 bits per sample
- Supports synchronization between sub-blocks or with other SAIs
- Companding modes (μ-Law, A-Law)
- 8-word FIFO size
- 2 DMA interfaces, 2 Interrupt lines

The SAI supports all the standard audio sampling rates, depending on the crystal frequency used for the application.
In addition, the SAI supports MASTER and SLAVE mode, half-duplex or full-duplex communication.
It is also possible to synchronize several SAIs together.
The SAI also provides a FIFO buffer of 8 samples, and up to two interrupt and DMA interfaces.



Block diagram

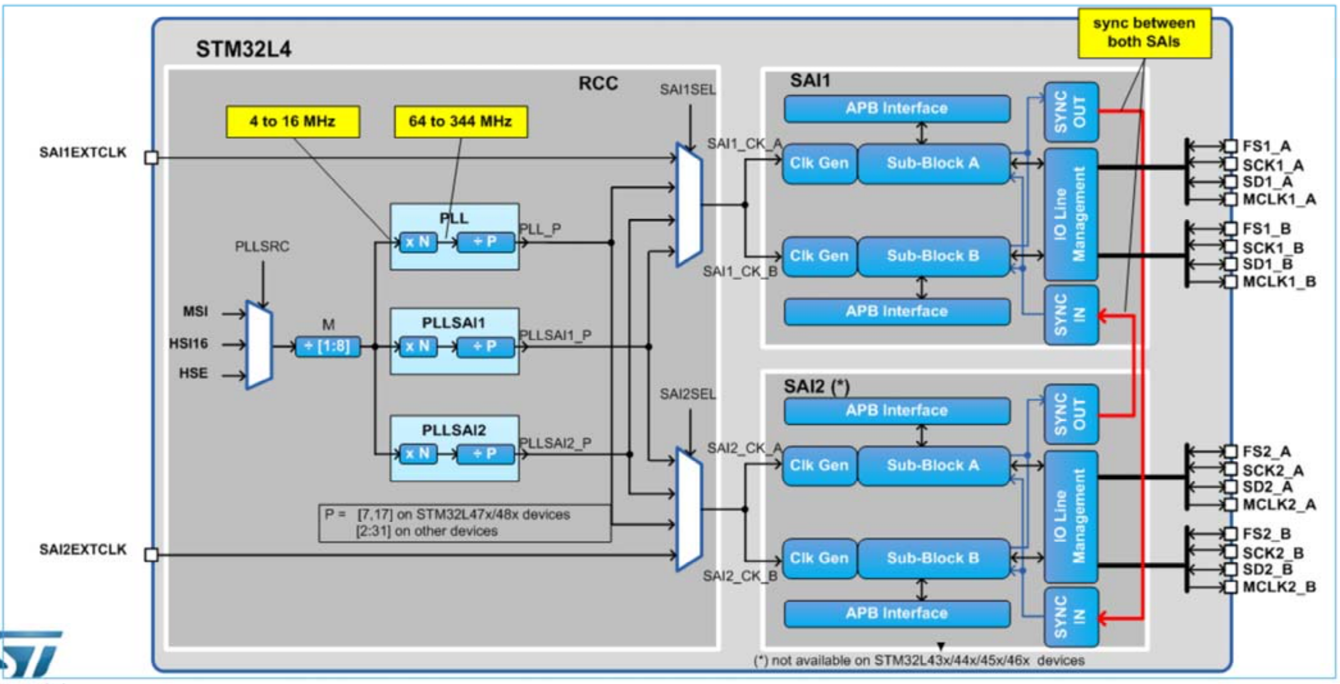
The SAI is composed of two independent sub-blocks (A and B).
Each sub-block has:
Its own APB interface , clock generator , FIFO buffer , DMA interface , and Interrupt interface.

Each sub-block can be configured in receive or transmit, master or slave, with its own protocol.
Internal and external synchronization allows two sub-blocks to be synchronized, or two SAIs to be synchronized.

Each sub-block can handle up to four IOs:
FS_x is the frame synchronization signal
SCK_x is the bitclock
SD_x is the serial data line
MCLK_x is the MASTER clock.

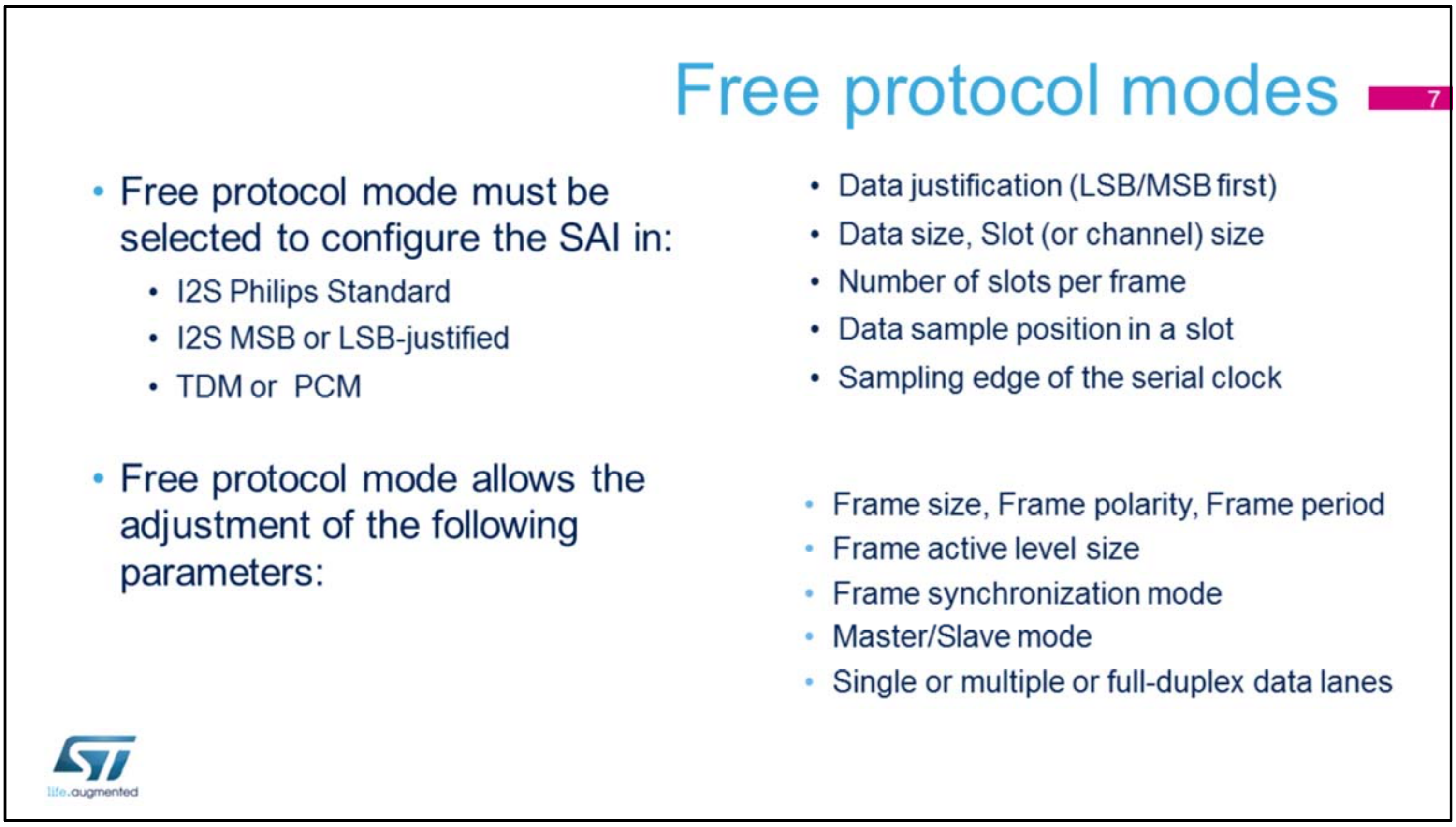
SAI In the circuit

6



life.augmented

The STM32L4xx, embeds 2 SAIs.
Each SAI can receive a kernel clock (SAIn_CK_x) from one of the three internal PLL or from the PADs (SAInEXTCLK).
The kernel clock is used by the SAI in order to generate the timing of the serial audio interface when programmed in MASTER mode.

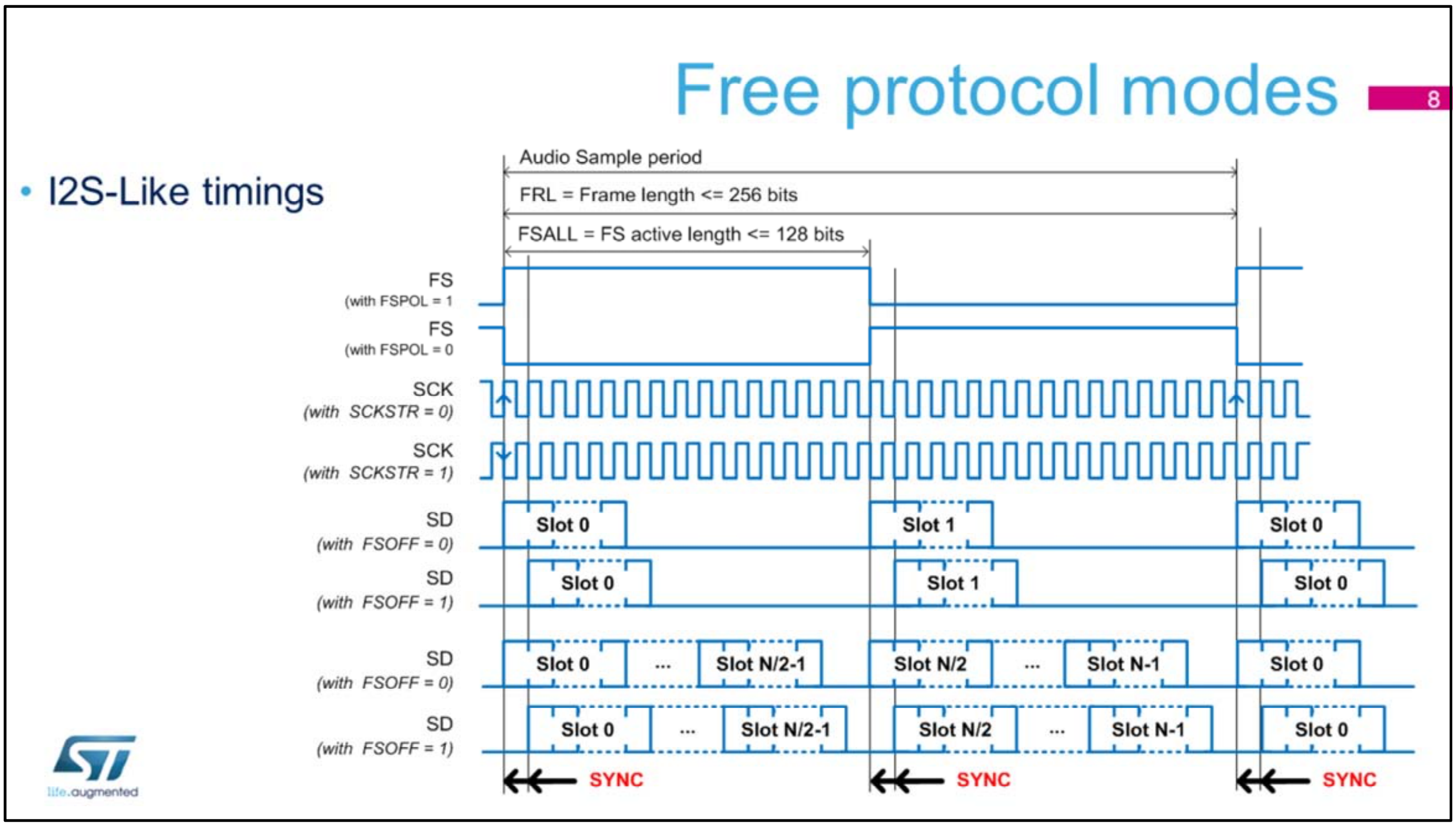


Free protocol modes

- Free protocol mode must be selected to configure the SAI in:
  - I2S Philips Standard
  - I2S MSB or LSB-justified
  - TDM or PCM
- Free protocol mode allows the adjustment of the following parameters:
  - Data justification (LSB/MSB first)
  - Data size, Slot (or channel) size
  - Number of slots per frame
  - Data sample position in a slot
  - Sampling edge of the serial clock
  - Frame size, Frame polarity, Frame period
  - Frame active level size
  - Frame synchronization mode
  - Master/Slave mode
  - Single or multiple or full-duplex data lanes

life.augmented

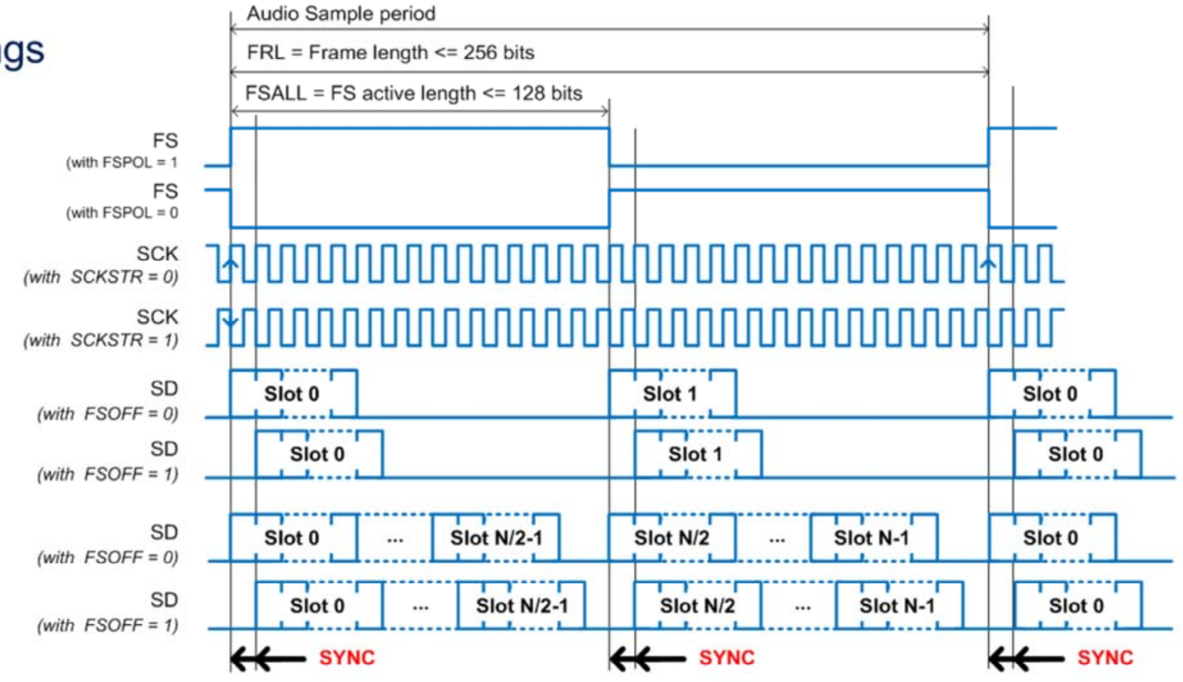
The free protocol mode, the flexible programming interface facilitates the configuration of most common audio standard interfaces.



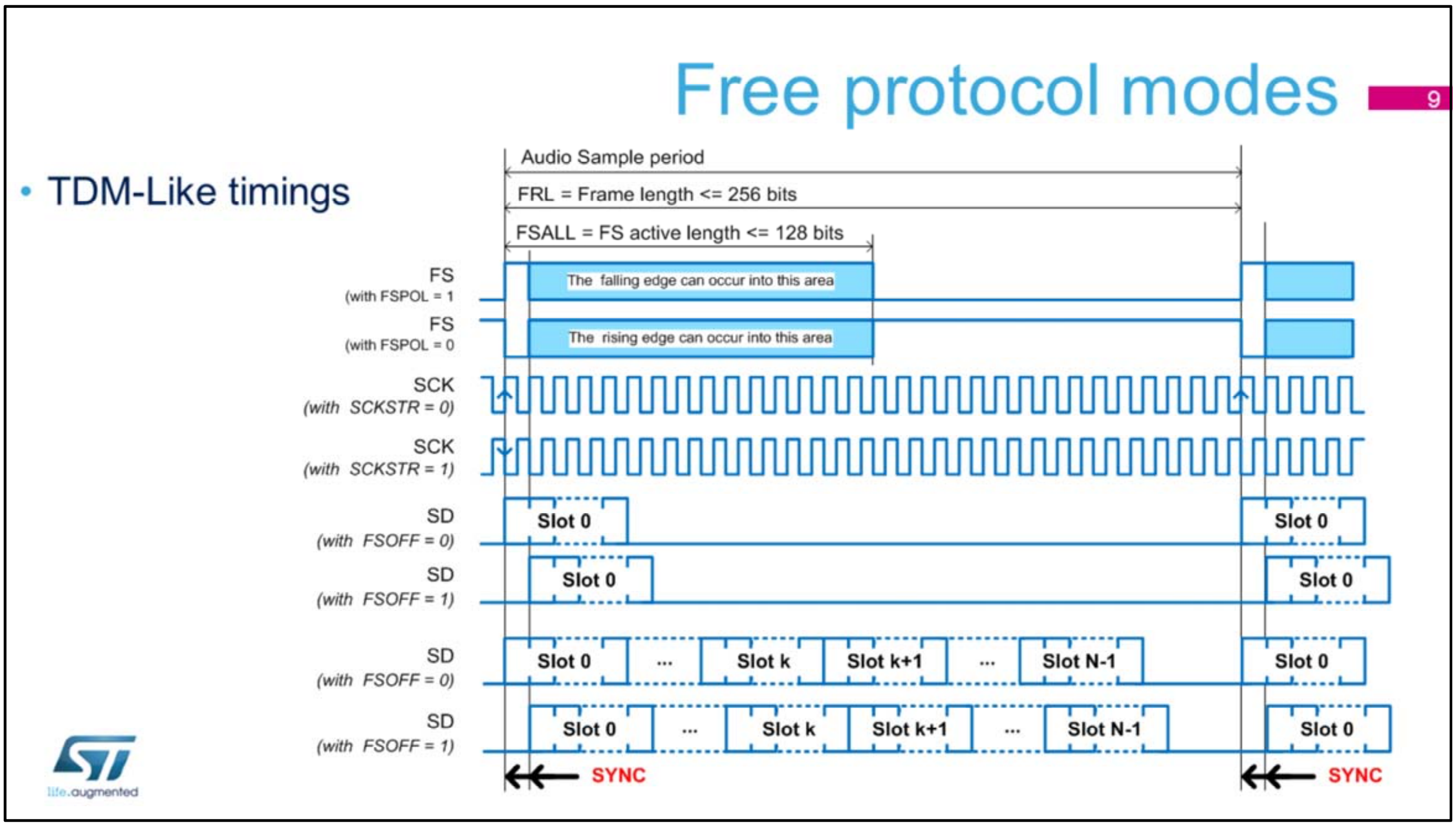
Free protocol modes

8

- I2S-Like timings

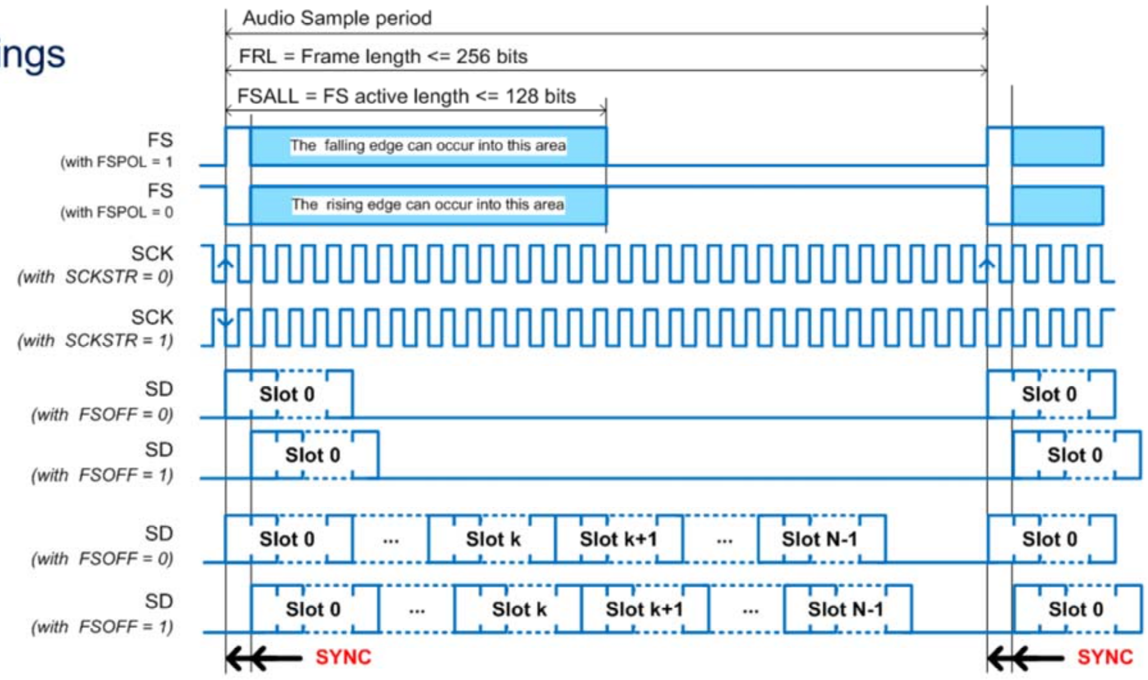


The following example shows some of the possibilities of the interface, for I2S-Like protocols:
In I2S-Like protocol, each edge of the frame synchronization signal (FS) is used to align the slots start position.
The frame length, the duty cycle, and polarity can be adjusted.
The clock strobing edge can be selected.
The position of the slots with respect to the frame edges can be selected.
The number of slots per frame: needs to be an even number in I2S-Like Protocol.



Free protocol modes

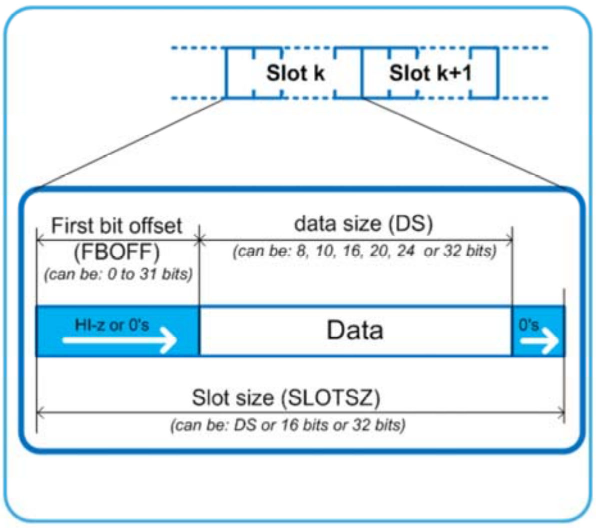
- TDM-Like timings



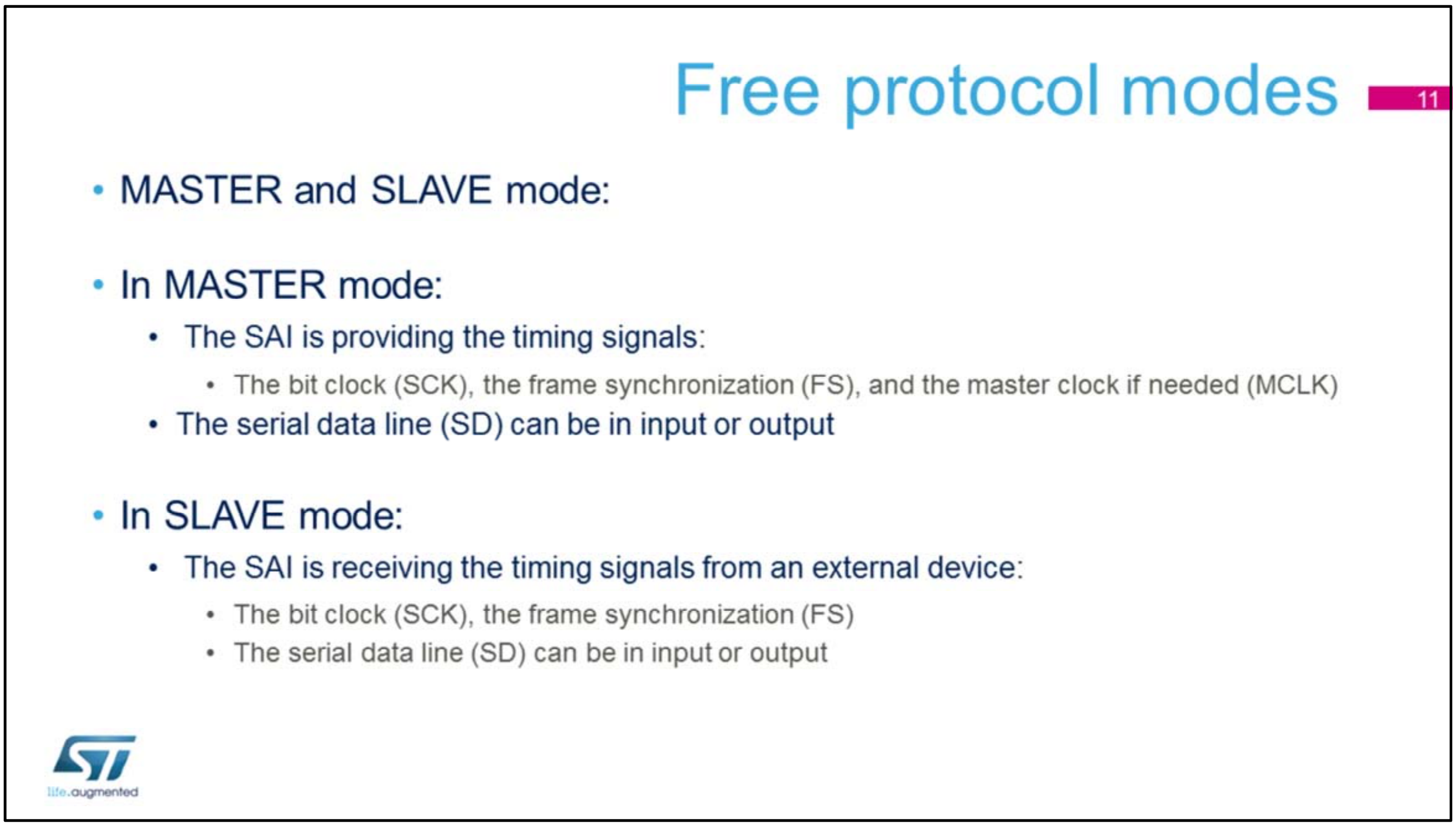
The following example shows some of the possibilities of the interface, for the TDM-Like protocols:
In TDM-Like protocol, only one edge of the frame synchronization (rising or falling) is used to align the slots position.
The frame length, the duty cycle, and polarity can be adjusted.
The clock strobing edge can be selected.
The position of the slots with respect to the frame active edge can be selected.
The number of slots per frame (up to 16) can be selected.

Free protocol modes

- Slot configuration:
- Up to 16 slots per audio frame
- Each slot can be defined as active or not
- The position of the data within a slot is adjustable .
- The data line can be set in HI-z
  - For inactive slots
  - During FBOFF area



The slot size is always bigger than or equal to the data size.
The SAI allows control of the position of the data inside each slot, and setting of the un-used slots to HiZ if needed.
This function can be useful when the data line is shared between several devices.

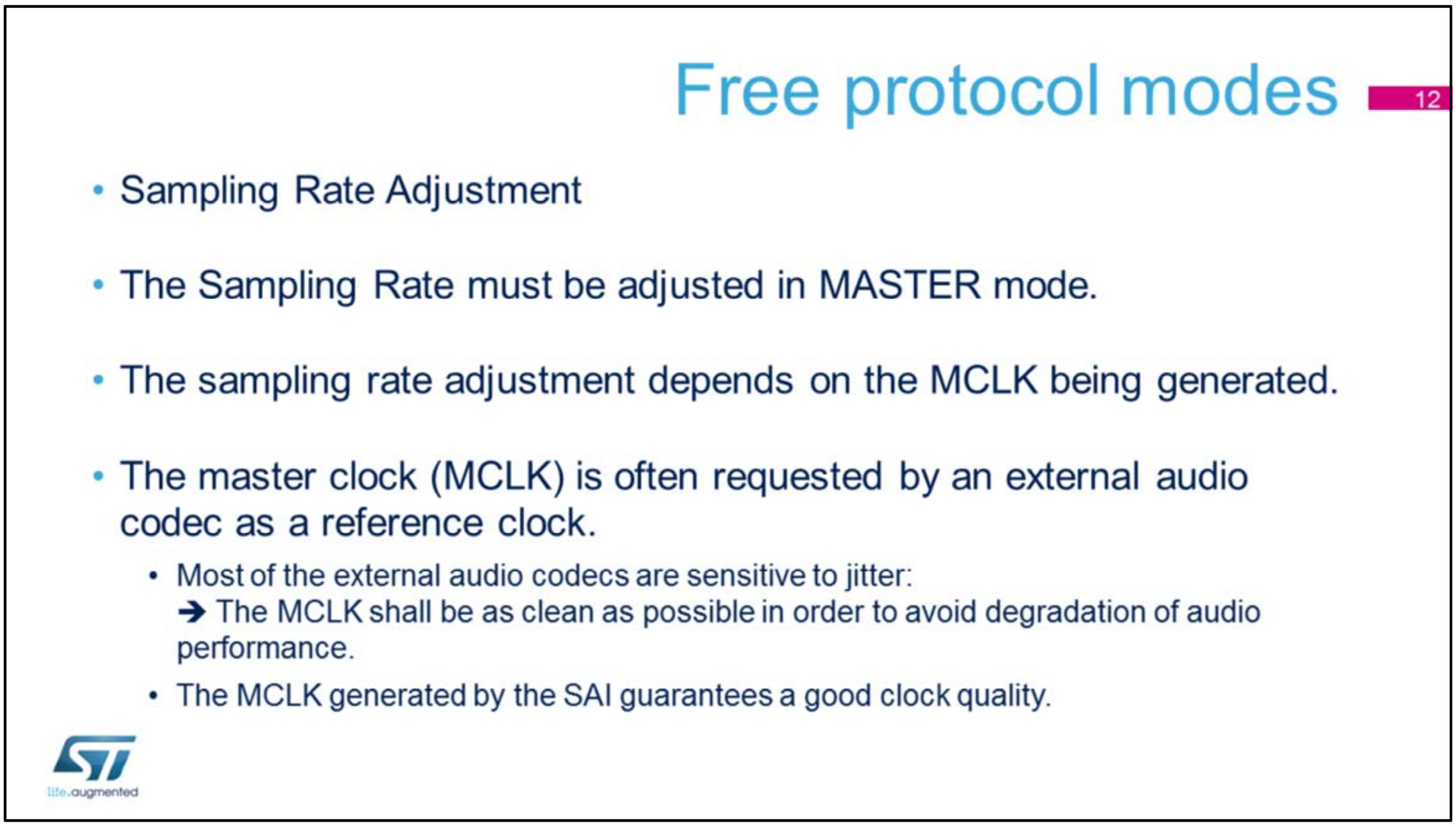


Free protocol modes

- MASTER and SLAVE mode:
- In MASTER mode:
  - The SAI is providing the timing signals:
    - The bit clock (SCK), the frame synchronization (FS), and the master clock if needed (MCLK)
  - The serial data line (SD) can be in input or output
- In SLAVE mode:
  - The SAI is receiving the timing signals from an external device:
    - The bit clock (SCK), the frame synchronization (FS)
    - The serial data line (SD) can be in input or output

In MASTER mode the SAI can generate the master clock (MCLK), or use an external master clock via SAIxEXTCLK PADs.
The master clock can be used to provide a reference clock to the external audio codecs.

In SLAVE mode, the MCLK signal is not used.

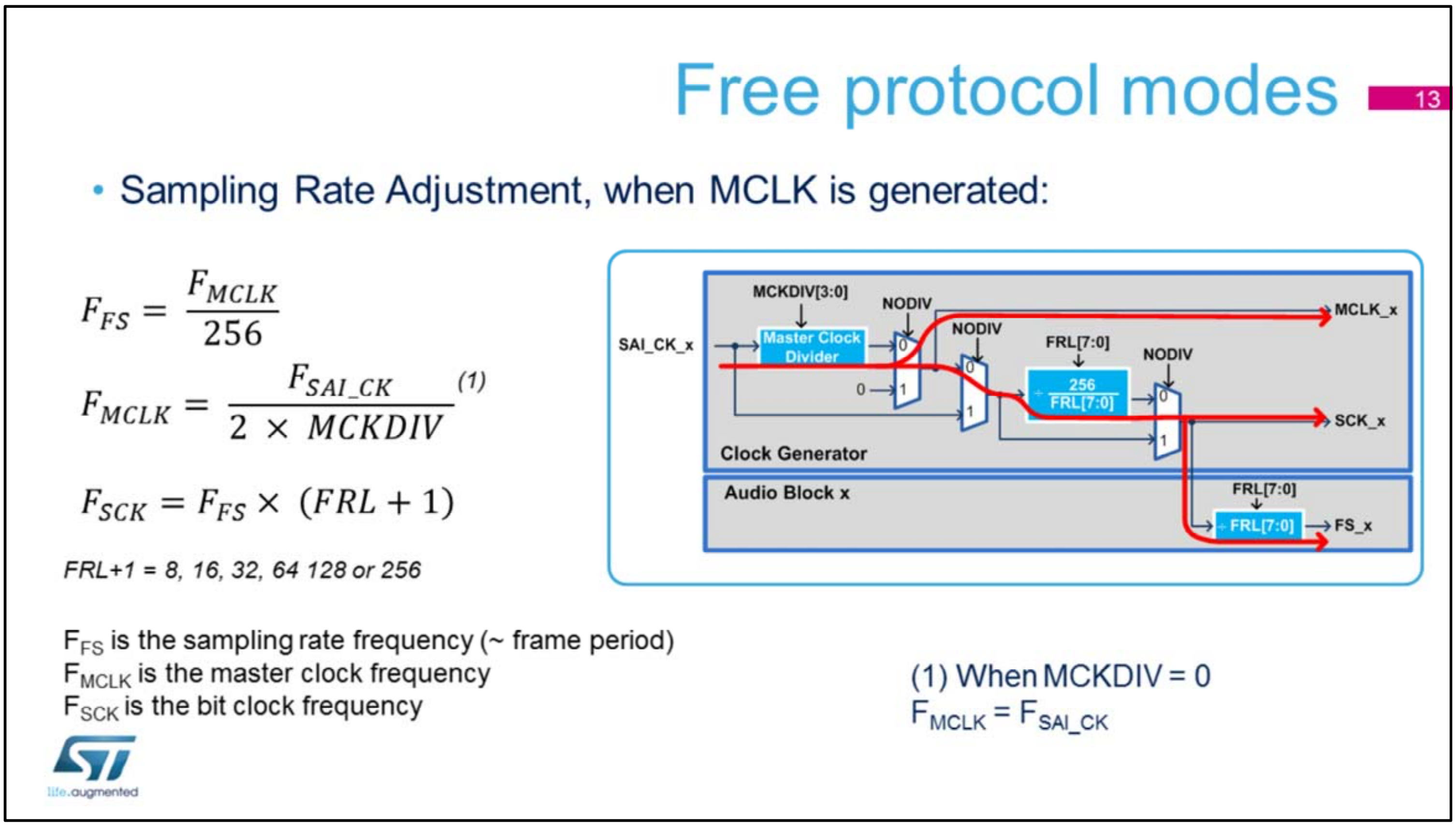


Free protocol modes

- Sampling Rate Adjustment
- The Sampling Rate must be adjusted in MASTER mode.
- The sampling rate adjustment depends on the MCLK being generated.
- The master clock (MCLK) is often requested by an external audio codec as a reference clock.
  - Most of the external audio codecs are sensitive to jitter:
    ➔ The MCLK shall be as clean as possible in order to avoid degradation of audio performance.
  - The MCLK generated by the SAI guarantees a good clock quality.

life.augmented

In MASTER mode it is up to the SAI to generate the timing, in order to provide the correct sampling rate.
In SLAVE mode, the sampling rate is provided by the external audio device.



Free protocol modes

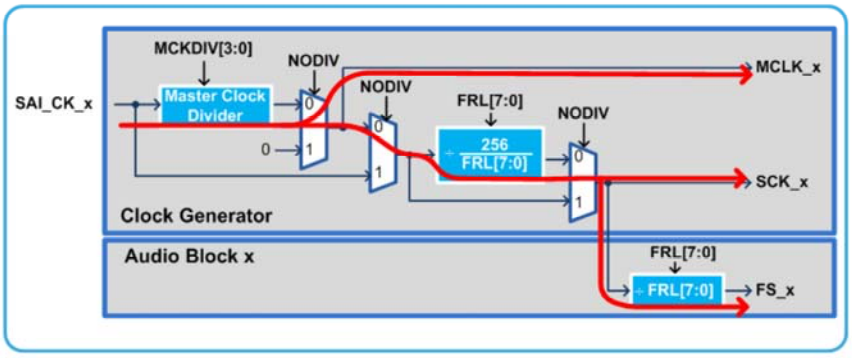
- Sampling Rate Adjustment, when MCLK is generated:

$$F_{FS} = \frac{F_{MCLK}}{256}$$

$$F_{MCLK} = \frac{F_{SAI\_CK}}{2 \times MCKDIV} \quad (1)$$

$$F_{SCK} = F_{FS} \times (FRL + 1)$$

FRL+1 = 8, 16, 32, 64 128 or 256

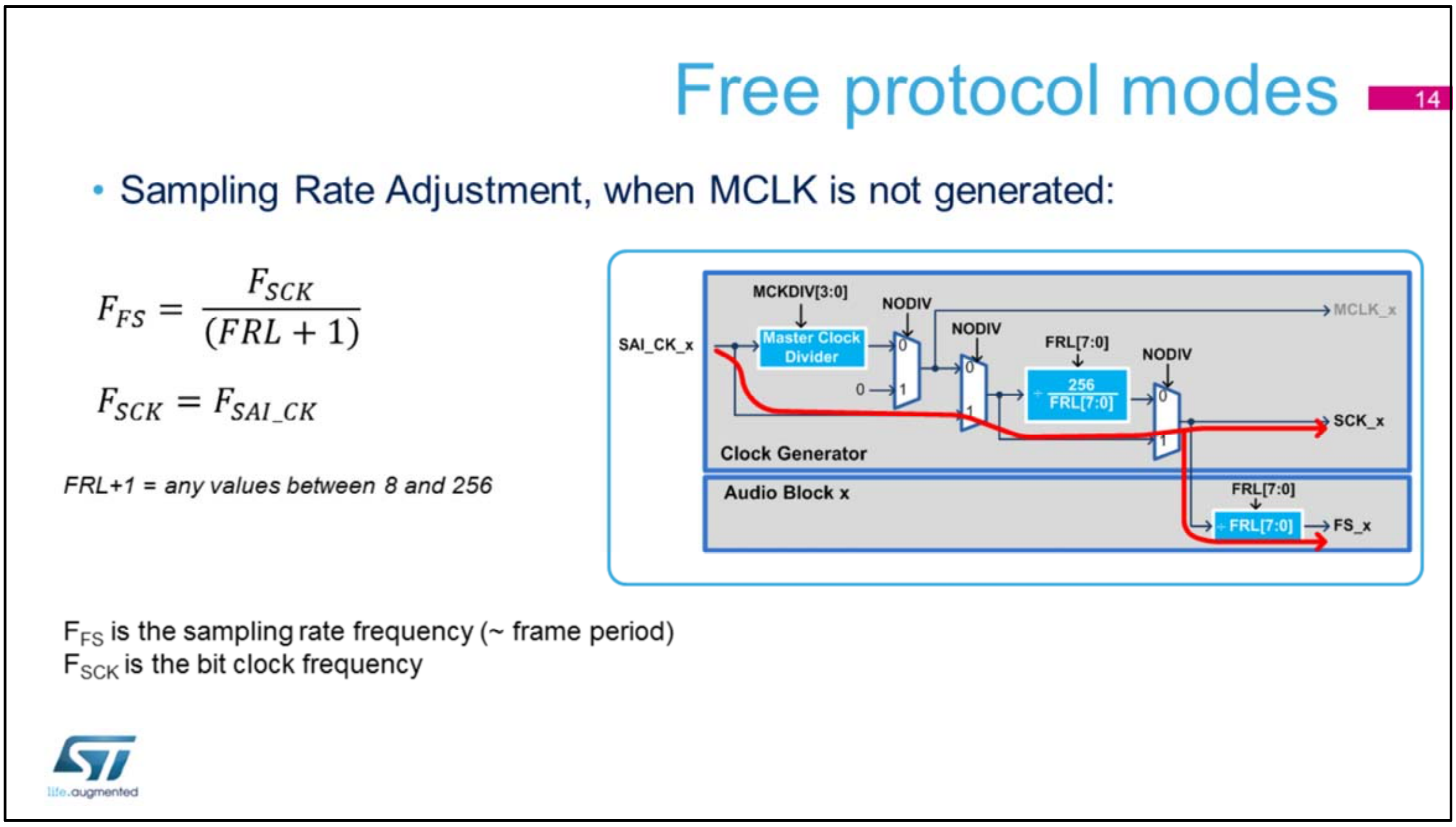


F_{FS} is the sampling rate frequency (~ frame period)
F_{MCLK} is the master clock frequency
F_{SCK} is the bit clock frequency

(1) When MCKDIV = 0
$F_{MCLK} = F_{SAI\_CK}$

The clock generator is needed for MASTER mode communications, it is used to adjust the sampling rate of the serial audio interface. The clock generator provides the root frequency for the MCLK_x, SCK_x and the FS_x.

When the master clock (MCLK) needs to be generated, the frame length must be a power of two.
The ratio between the FS_x frequency and the MCLK_x frequency is fixed to 256.



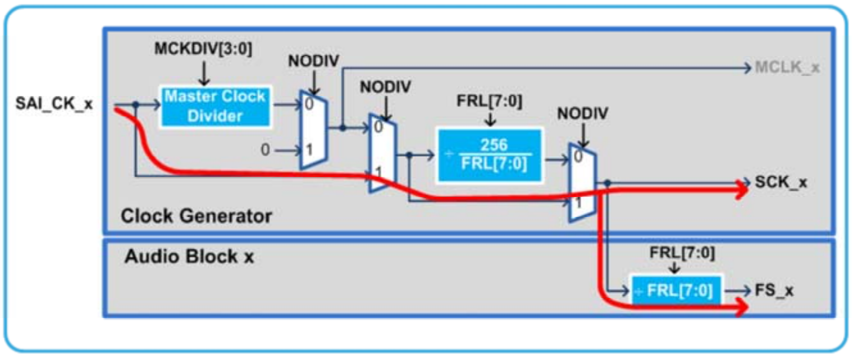
Free protocol modes

- Sampling Rate Adjustment, when MCLK is not generated:

$$F_{FS} = \frac{F_{SCK}}{(FRL+1)}$$

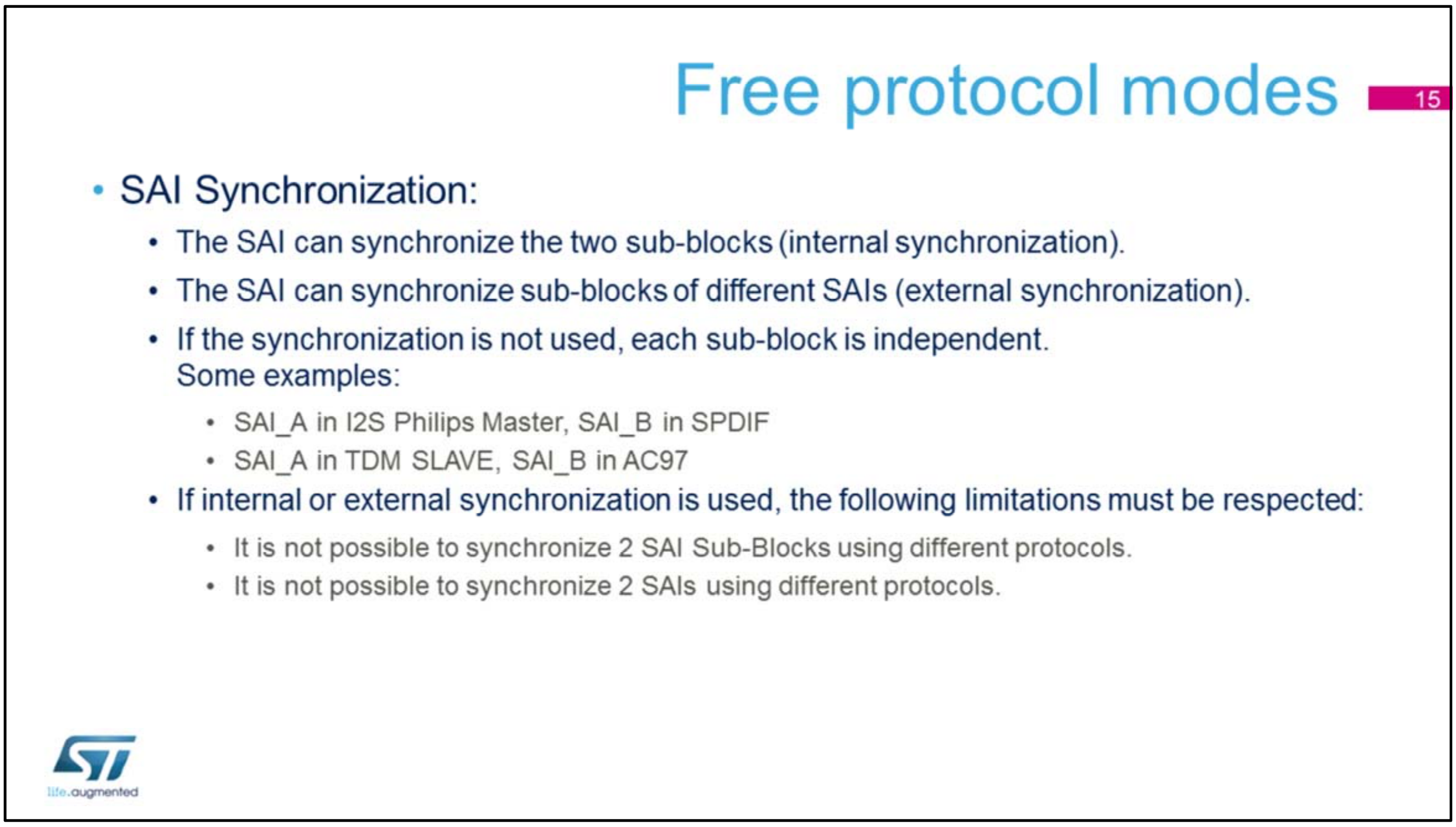
$$F_{SCK} = F_{SAI\_CK}$$

FRL+1 = any values between 8 and 256



F_{FS} is the sampling rate frequency (~ frame period)
F_{SCK} is the bit clock frequency

When the MCLK_x does not need to be generated, the frame length can take any value from 8 to 256.



Free protocol modes

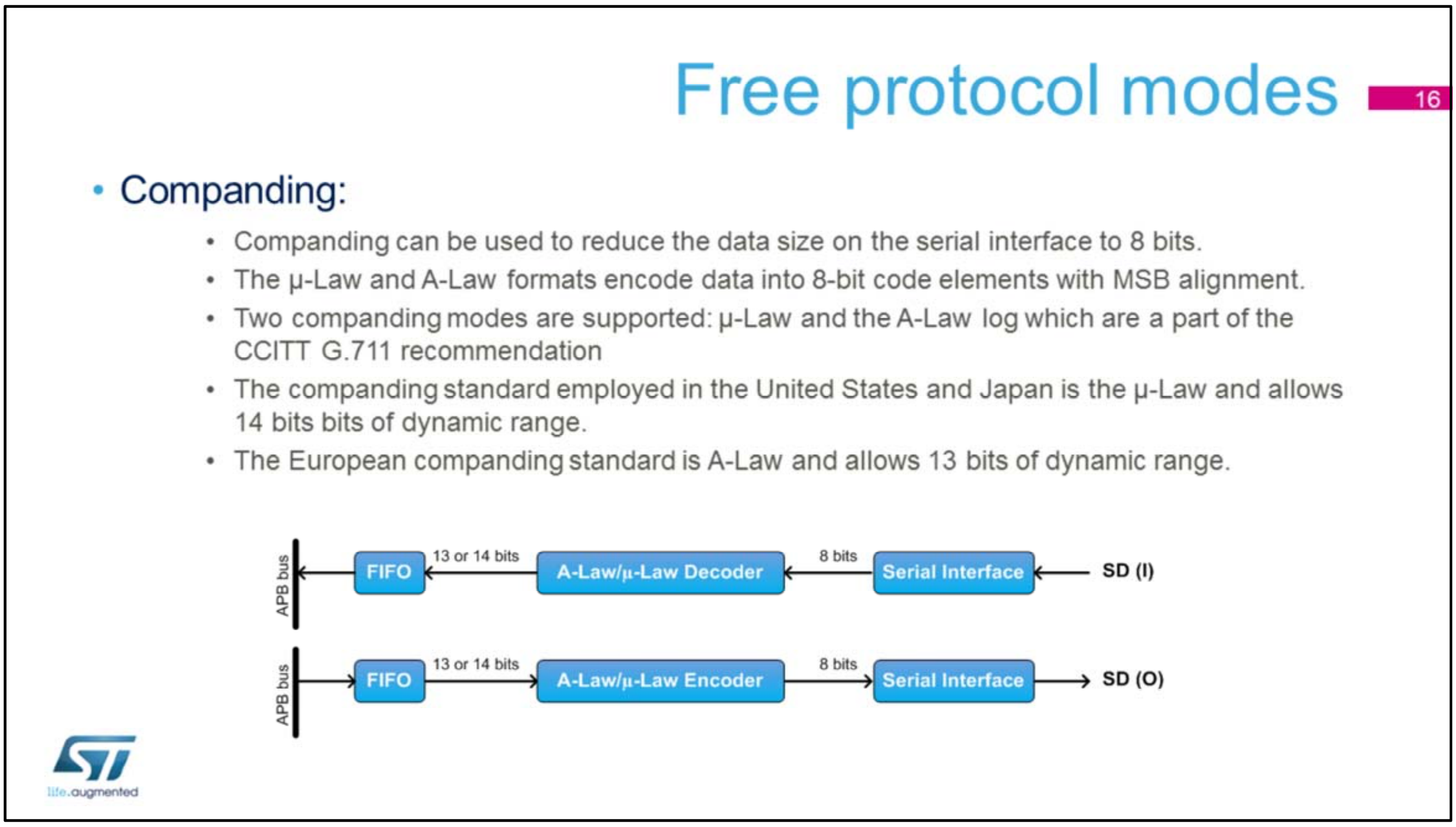
15

- SAI Synchronization:
  - The SAI can synchronize the two sub-blocks (internal synchronization).
  - The SAI can synchronize sub-blocks of different SAIs (external synchronization).
  - If the synchronization is not used, each sub-block is independent.
    Some examples:
    - SAI_A in I2S Philips Master, SAI_B in SPDIF
    - SAI_A in TDM SLAVE, SAI_B in AC97
  - If internal or external synchronization is used, the following limitations must be respected:
    - It is not possible to synchronize 2 SAI Sub-Blocks using different protocols.
    - It is not possible to synchronize 2 SAIs using different protocols.

life.augmented

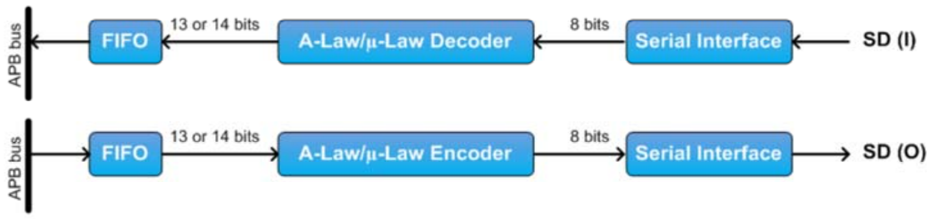
The internal synchronization can be used for communication requesting two data lanes, such as full-duplex I2S.
The external synchronization can be used for communication requesting more than 2 data lanes (up to 4). For example when interfacing HDMI ICS.

All the sub-blocks synchronized together must use the same protocol characteristics.

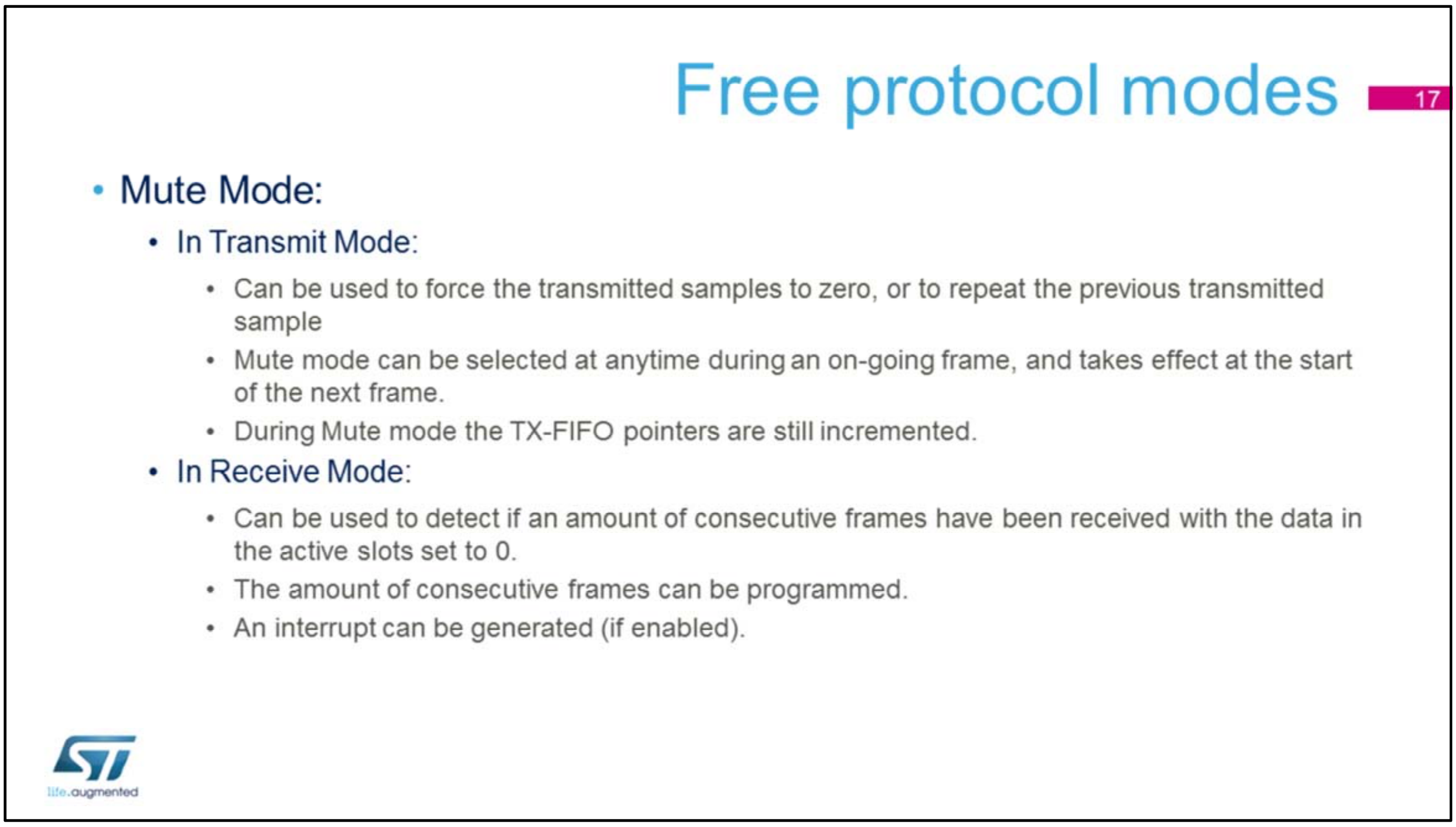


Free protocol modes

- Companding:
  - Companding can be used to reduce the data size on the serial interface to 8 bits.
  - The μ-Law and A-Law formats encode data into 8-bit code elements with MSB alignment.
  - Two companding modes are supported: μ-Law and the A-Law log which are a part of the CCITT G.711 recommendation
  - The companding standard employed in the United States and Japan is the μ-Law and allows 14 bits bits of dynamic range.
  - The European companding standard is A-Law and allows 13 bits of dynamic range.



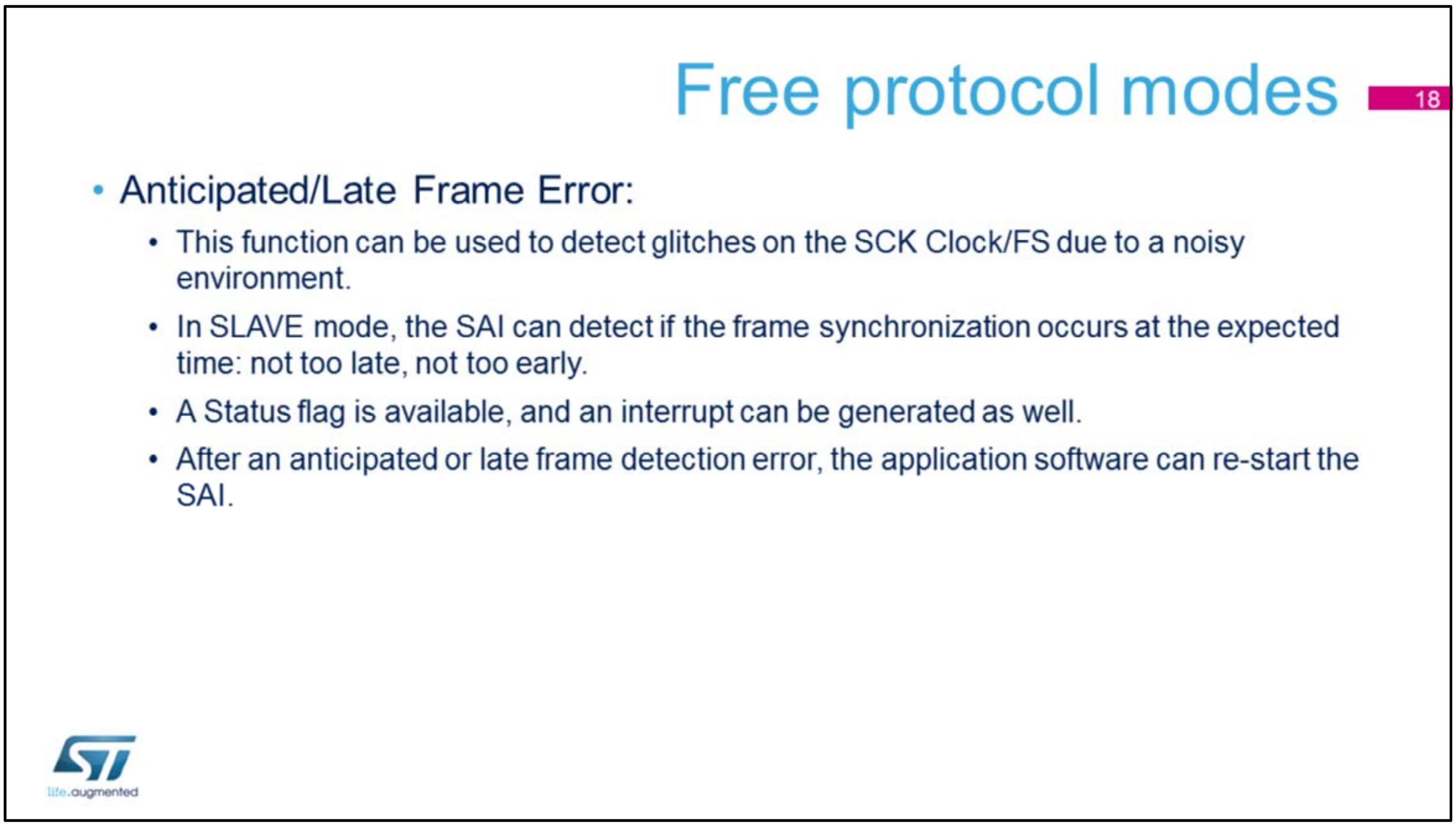
In order to reduce the data size, it is possible to insert in the data path, a A-law or u-law compander.
Note that A-law and u-law are not lossless compressors.
Companding modes are generally used in telephony:
The small data are amplified and the big data are attenuated .
The SNR tends to be identical for a strong and for a weak signal.



Free protocol modes

- Mute Mode:
  - In Transmit Mode:
    - Can be used to force the transmitted samples to zero, or to repeat the previous transmitted sample
    - Mute mode can be selected at anytime during an on-going frame, and takes effect at the start of the next frame.
    - During Mute mode the TX-FIFO pointers are still incremented.
  - In Receive Mode:
    - Can be used to detect if an amount of consecutive frames have been received with the data in the active slots set to 0.
    - The amount of consecutive frames can be programmed.
    - An interrupt can be generated (if enabled).

The SAI also provides a MUTE function.



Free protocol modes

18

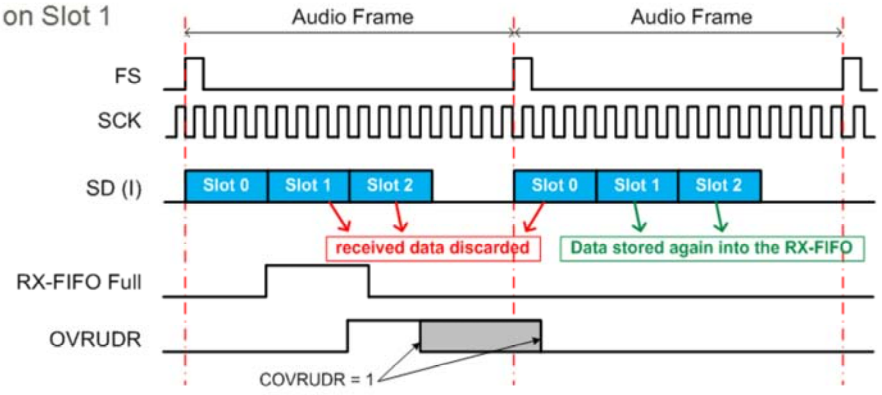
- Anticipated/Late Frame Error:
  - This function can be used to detect glitches on the SCK Clock/FS due to a noisy environment.
  - In SLAVE mode, the SAI can detect if the frame synchronization occurs at the expected time: not too late, not too early.
  - A Status flag is available, and an interrupt can be generated as well.
  - After an anticipated or late frame detection error, the application software can re-start the SAI.

The Anticipated/Late frame error detection increases the interface reliability by detecting unexpected frame synchronization misalignment.

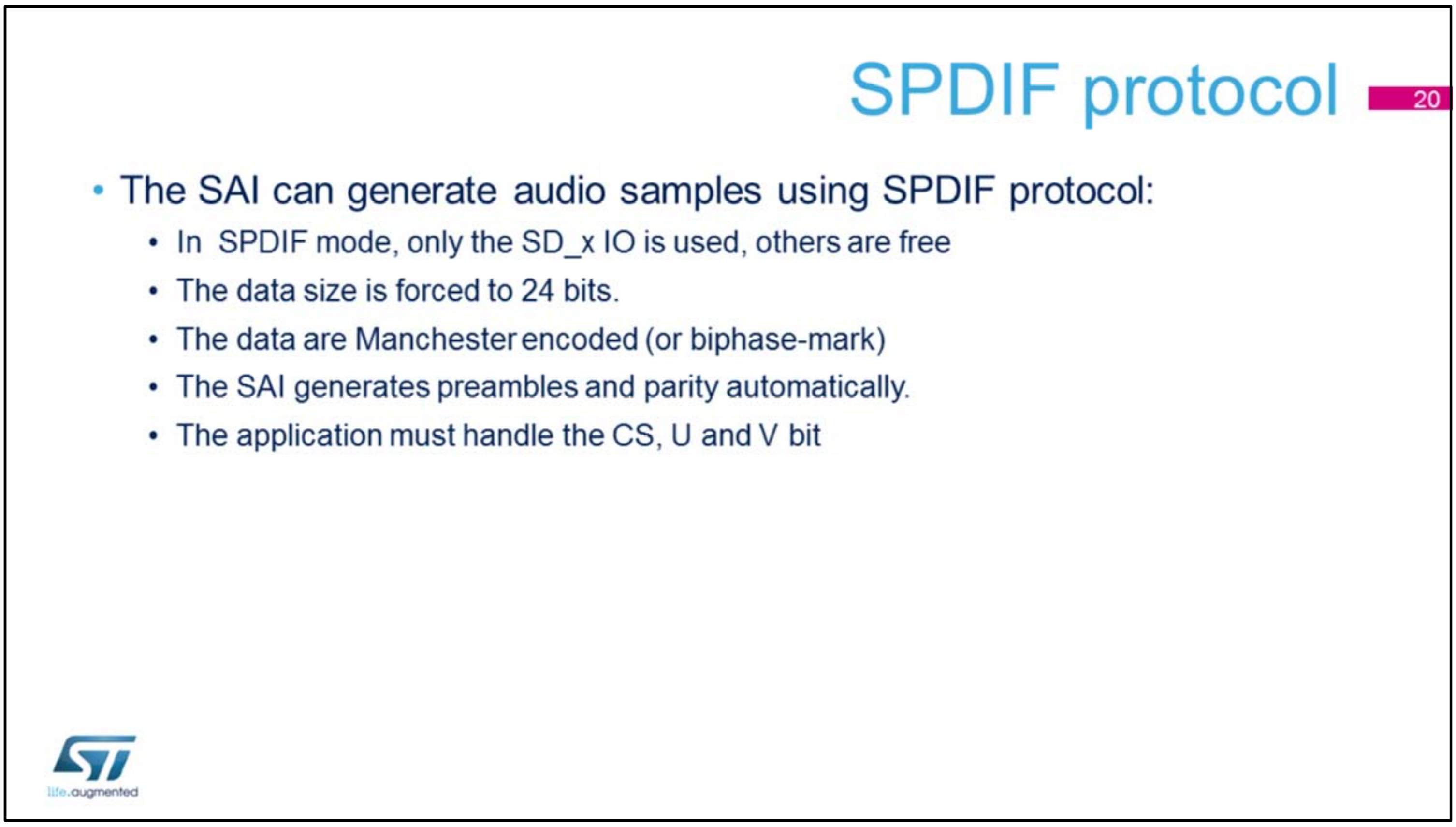
Free protocol modes

The SAI guarantees the data alignment even if underrun/overrun occurs

- Overrun/Underrun Handling:
  - Overrun occurs when the RX-FIFO is full, and new data coming from the serial interface has to be stored.
  - Underrun occurs when the TX-FIFO is empty, and new data is requested by the serial interface.
  - Example : FIFO Overrun on Slot 1



life.augmented

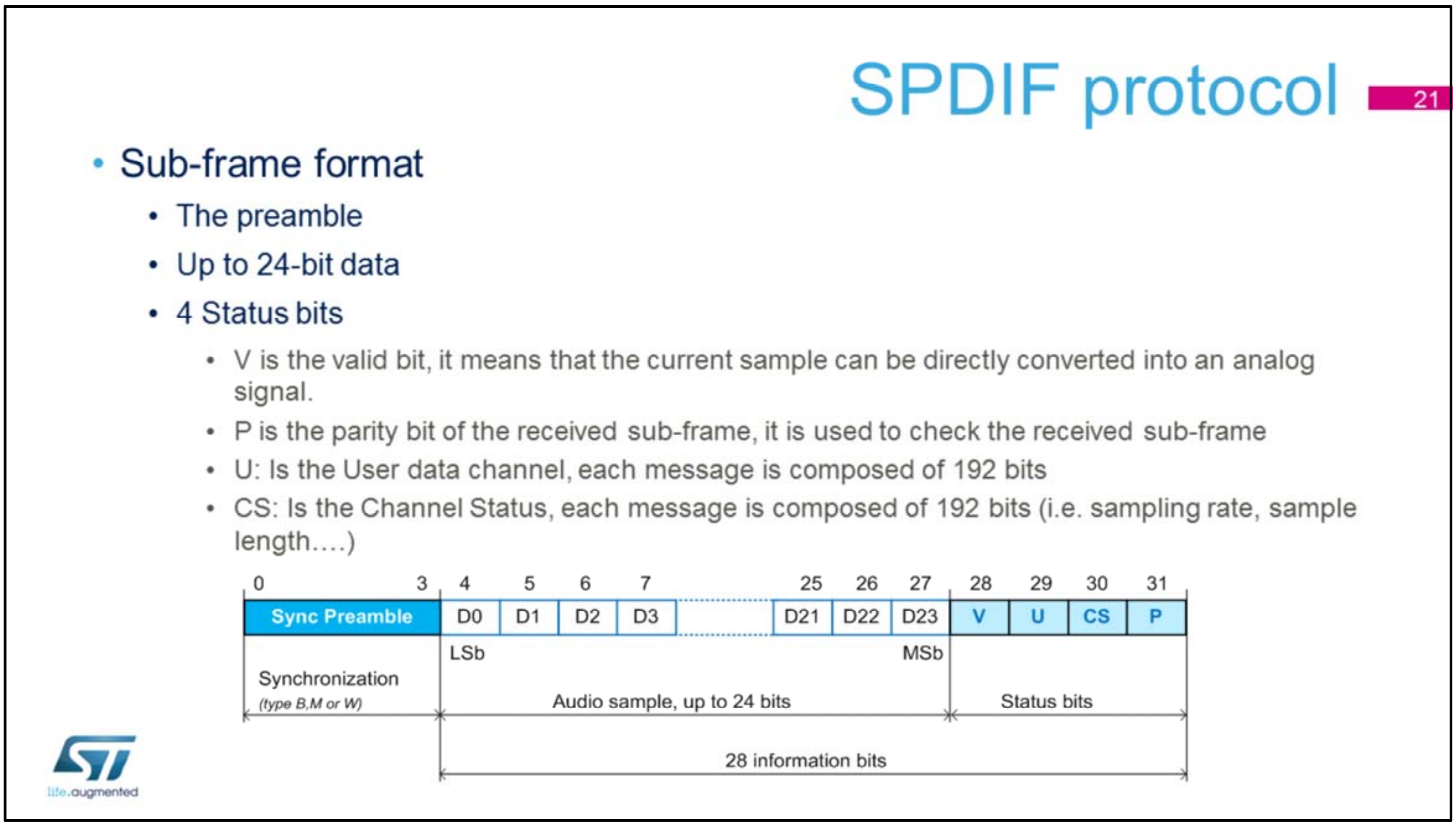


SPDIF protocol

- The SAI can generate audio samples using SPDIF protocol:
  - In SPDIF mode, only the SD_x IO is used, others are free
  - The data size is forced to 24 bits.
  - The data are Manchester encoded (or biphase-mark)
  - The SAI generates preambles and parity automatically.
  - The application must handle the CS, U and V bit

life.augmented

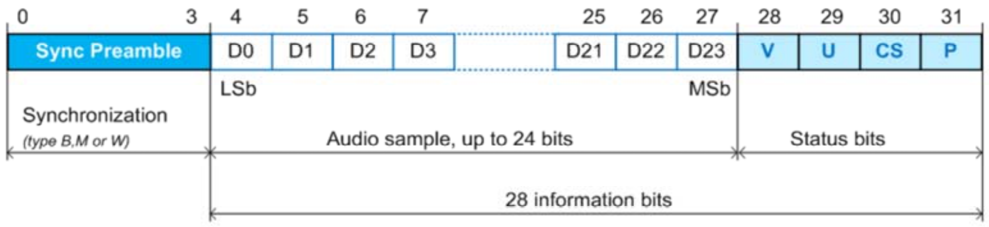
The SAI supports the audio IEC60958 standard in transmit mode when configured in SPDIF mode.
The software has to handle the CS, U and V bits.
The SAI generates the Parity bit according to the transmitted data.



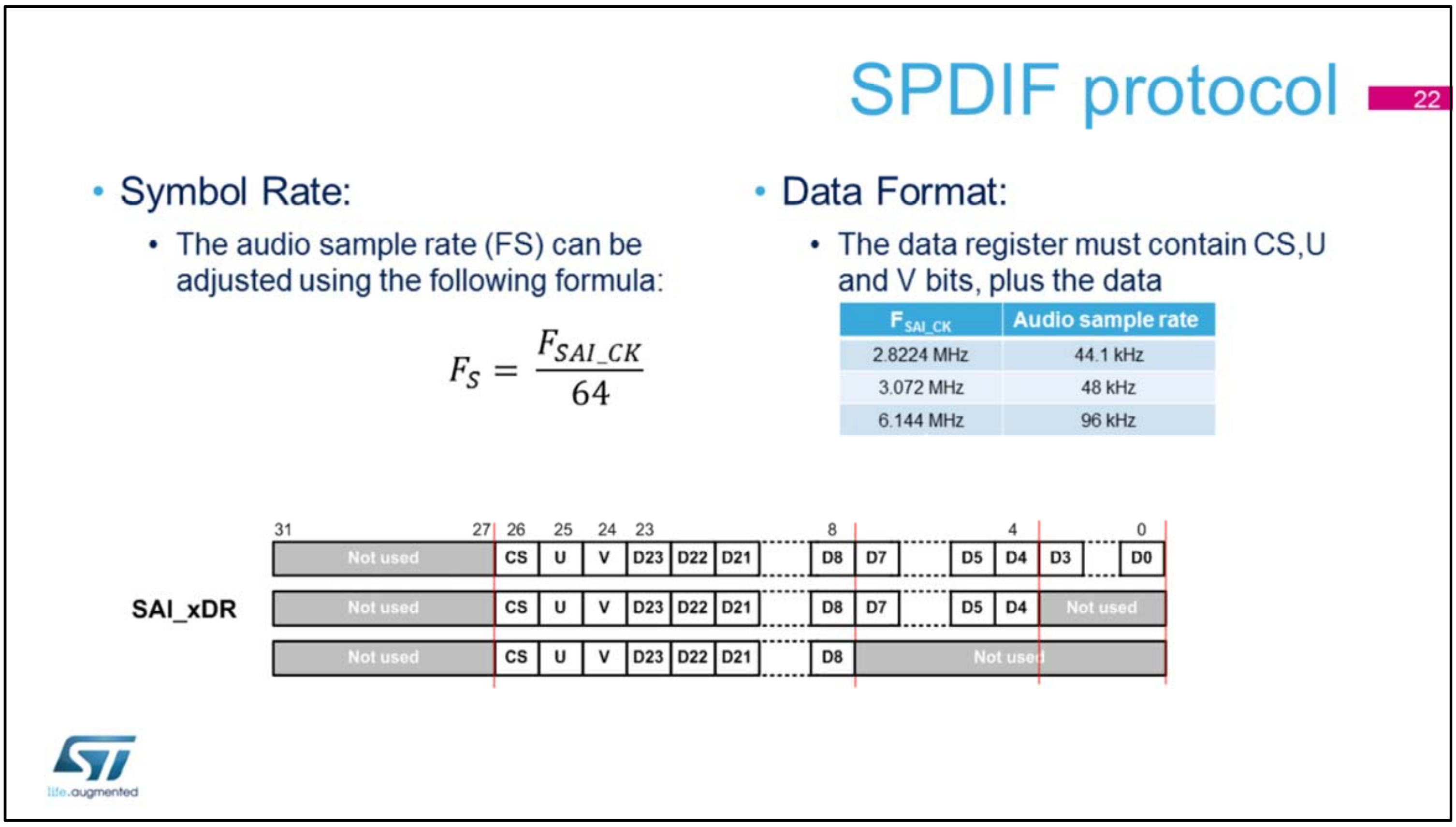
SPDIF protocol

21

- Sub-frame format
  - The preamble
  - Up to 24-bit data
  - 4 Status bits
    - V is the valid bit, it means that the current sample can be directly converted into an analog signal.
    - P is the parity bit of the received sub-frame, it is used to check the received sub-frame
    - U: Is the User data channel, each message is composed of 192 bits
    - CS: Is the Channel Status, each message is composed of 192 bits (i.e. sampling rate, sample length….)



In the IEC60958, the block structure is used to decode the Channel Status (CS), and User information (U).
Each block contains 192 frames.
Each frame contains 2 sub-frames.
Each sub-frame contains 32 bits.
A synchronization preamble allows the detection of the block and sub-frame boundaries.



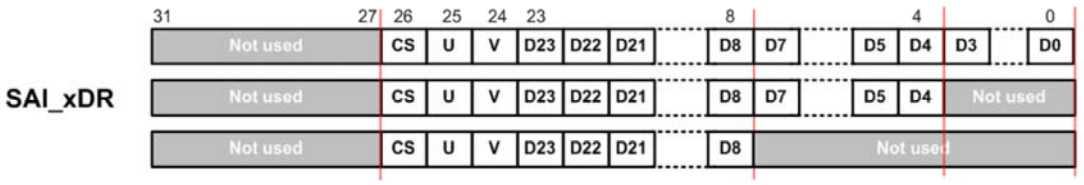
SPDIF protocol

- Symbol Rate:
  - The audio sample rate (FS) can be adjusted using the following formula:

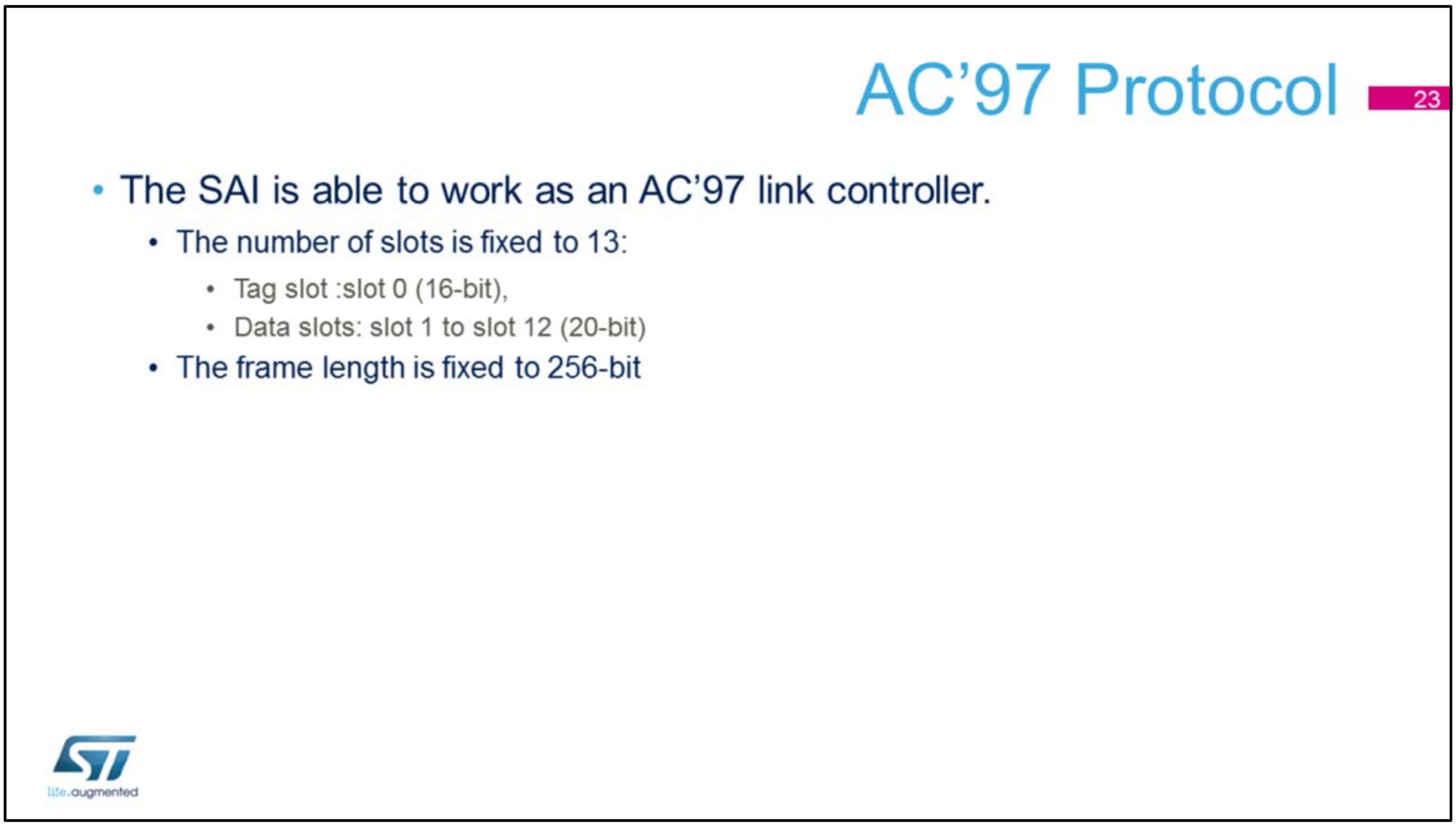
$$F_S = \frac{F_{SAI\_CK}}{64}$$

- Data Format:
  - The data register must contain CS,U and V bits, plus the data

F_{SAI_CK}	Audio sample rate
2.8224 MHz	44.1 kHz
3.072 MHz	48 kHz
6.144 MHz	96 kHz



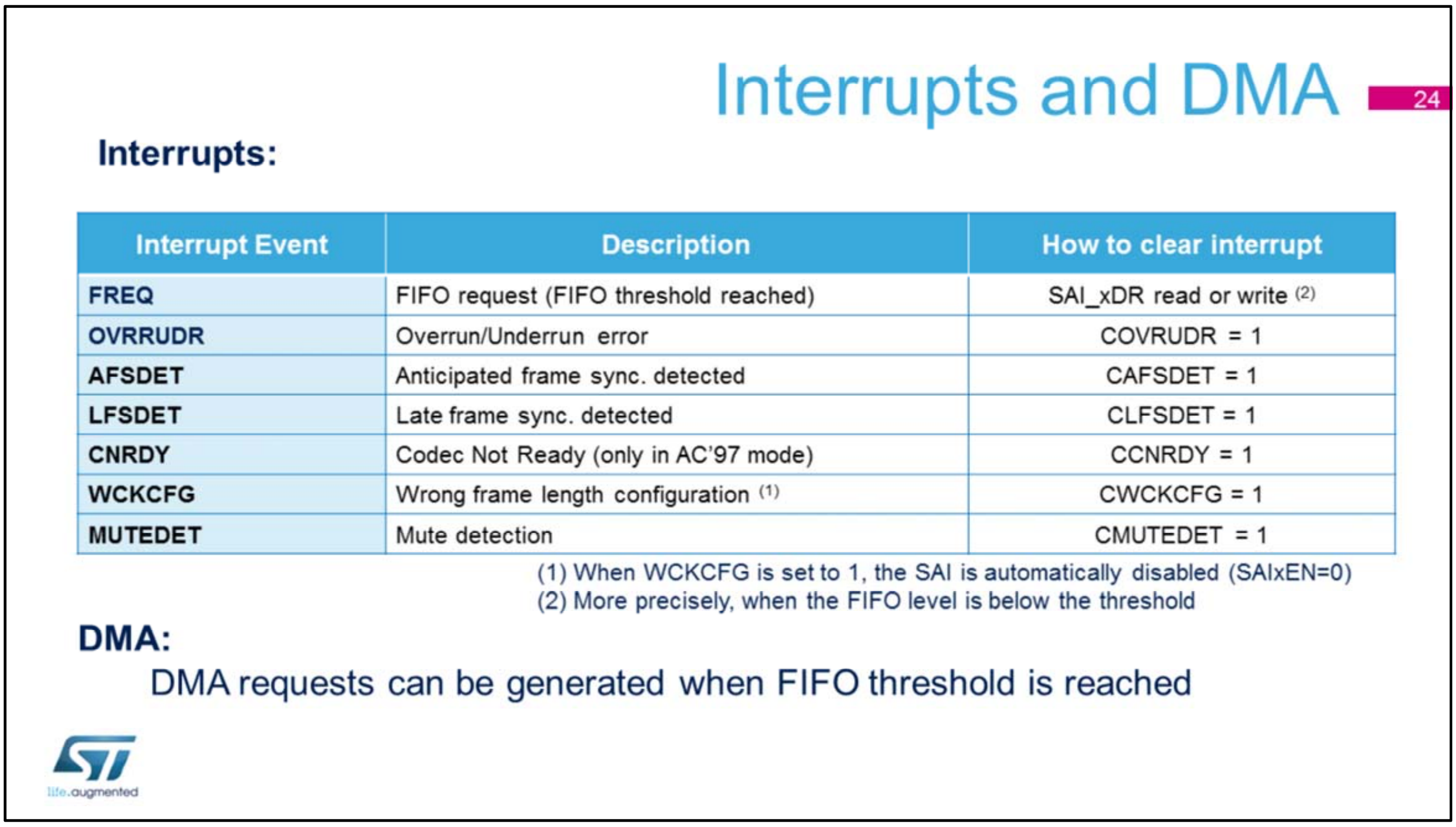
The Fsai_ck_x frequency must be adjusted in order to generate the proper audio sample rate.
The data inside the transmit FIFO must be adjusted as shown in the slide: the MSB must always be at position 23.



AC’97 Protocol

- The SAI is able to work as an AC’97 link controller.
  - The number of slots is fixed to 13:
    - Tag slot :slot 0 (16-bit),
    - Data slots: slot 1 to slot 12 (20-bit)
  - The frame length is fixed to 256-bit

The SAI is able to work as an AC’97 link controller.
When this protocol is used, the frame length, the slot number, and slot length are fixed by the hardware.



Interrupts and DMA

Interrupts:

Interrupt Event	Description	How to clear interrupt
FREQ	FIFO request (FIFO threshold reached)	SAI_xDR read or write (2)
OVRRUDR	Overrun/Underrun error	COVRUDR = 1
AFSDET	Anticipated frame sync. detected	CAFSDET = 1
LFSDET	Late frame sync. detected	CLFSDET = 1
CNRDY	Codec Not Ready (only in AC'97 mode)	CCNRDY = 1
WCKCFG	Wrong frame length configuration (1)	CWCKCFG = 1
MUTEDET	Mute detection	CMUTEDET = 1

(1) When WCKCFG is set to 1, the SAI is automatically disabled (SAIxEN=0)
(2) More precisely, when the FIFO level is below the threshold

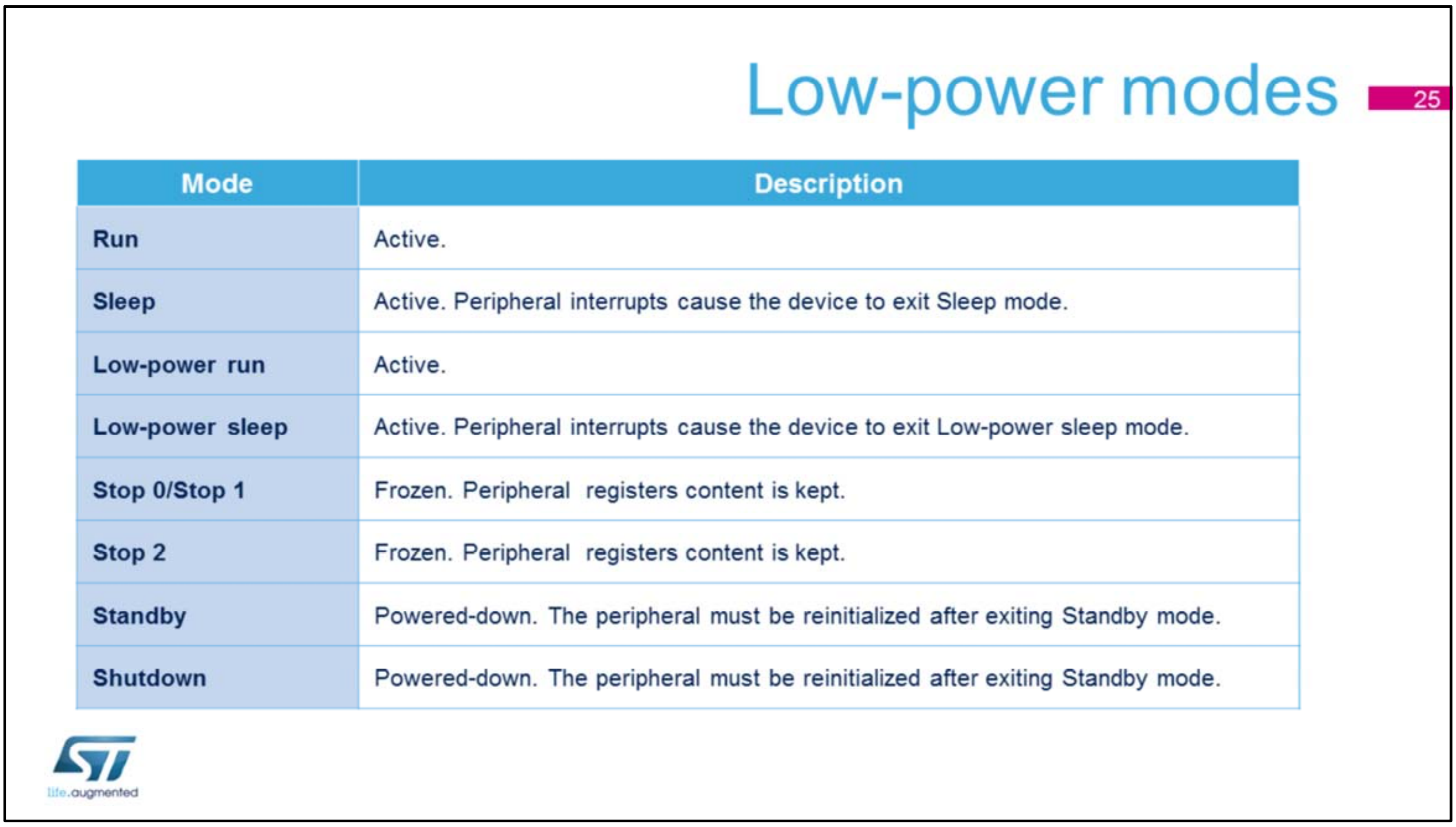
DMA:

DMA requests can be generated when FIFO threshold is reached

Several events can be enabled in order to generate interrupts.

- The FIFO request event, the overrun/underrun event,
- the anticipated or late frame synchronization event,
- the codec not ready event (only in AC'97),
- the mute detection event.

The WCKCFG event can be used in order to inform the user that the frame length of the SAI has been improperly programmed. This feature is only available in MASTER mode.



Low-power modes

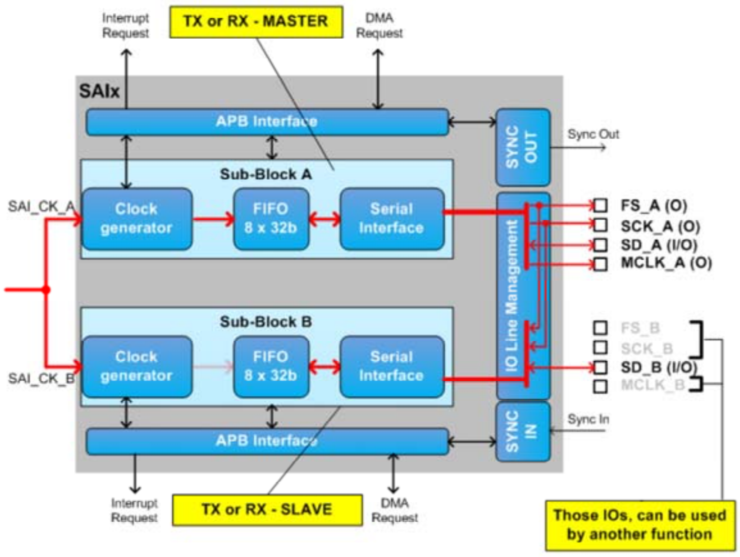
Mode	Description
Run	Active.
Sleep	Active. Peripheral interrupts cause the device to exit Sleep mode.
Low-power run	Active.
Low-power sleep	Active. Peripheral interrupts cause the device to exit Low-power sleep mode.
Stop 0/Stop 1	Frozen. Peripheral registers content is kept.
Stop 2	Frozen. Peripheral registers content is kept.
Standby	Powered-down. The peripheral must be reinitialized after exiting Standby mode.
Shutdown	Powered-down. The peripheral must be reinitialized after exiting Standby mode.

life.augmented

The SAI needs the bus interface clock (APB clock) and the kernel clock (SAIn_CK_x) to work properly.

Application examples

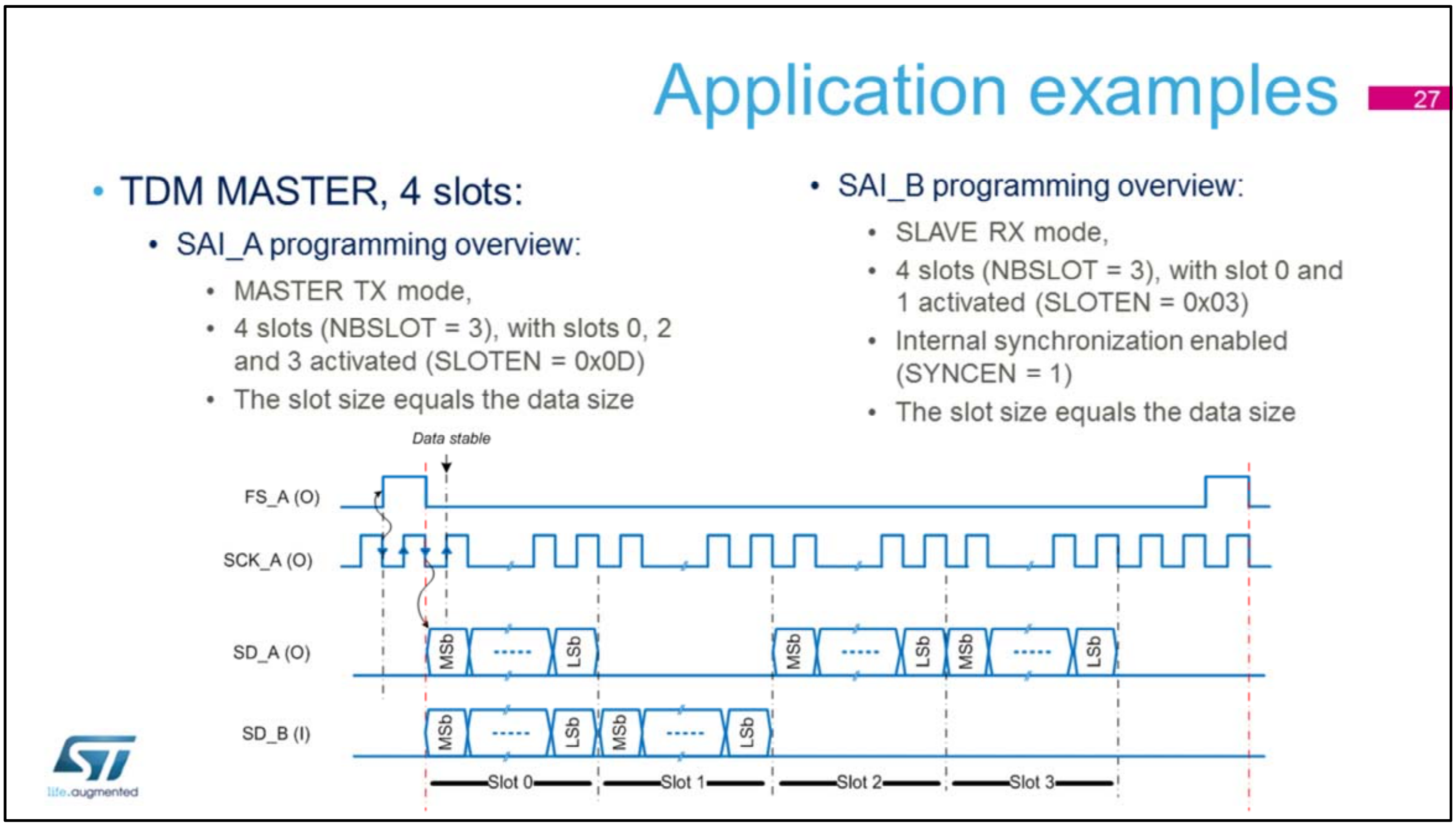
- MASTER full-duplex or dual lane:
  - Sub-block A is MASTER
  - Sub-block B is SLAVE
  - Sub-block B is synchronized to sub-block A



life.augmented

For a full-duplex MASTER mode, two data lanes are needed, so two sub-blocks need to be used.
The MASTER sub-block A, provides the synchronization to SLAVE sub-block B, using the internal synchronization feature (IO Line Management).

Using internal synchronization, the number Using internal synchronization, the number of IOs required is reduced to its minimum.



Application examples

- TDM MASTER, 4 slots:
  - SAI_A programming overview:
    - MASTER TX mode,
    - 4 slots (NBSLOT = 3), with slots 0, 2 and 3 activated (SLOTEN = 0x0D)
    - The slot size equals the data size
- SAI_B programming overview:
  - SLAVE RX mode,
  - 4 slots (NBSLOT = 3), with slot 0 and 1 activated (SLOTEN = 0x03)
  - Internal synchronization enabled (SYNCEN = 1)
  - The slot size equals the data size

Another example of full-duplex mode use case with the TDM protocol usage.
The slot 1 is inactive (not used) for sub-block A, the slots 2 and 3 are inactive for sub-block B.
For both sub-blocks, the frame structure has 4 slots.
The sub-block A will generate 3 samples per frame.
The sub-block B will receive 2 samples per frame.