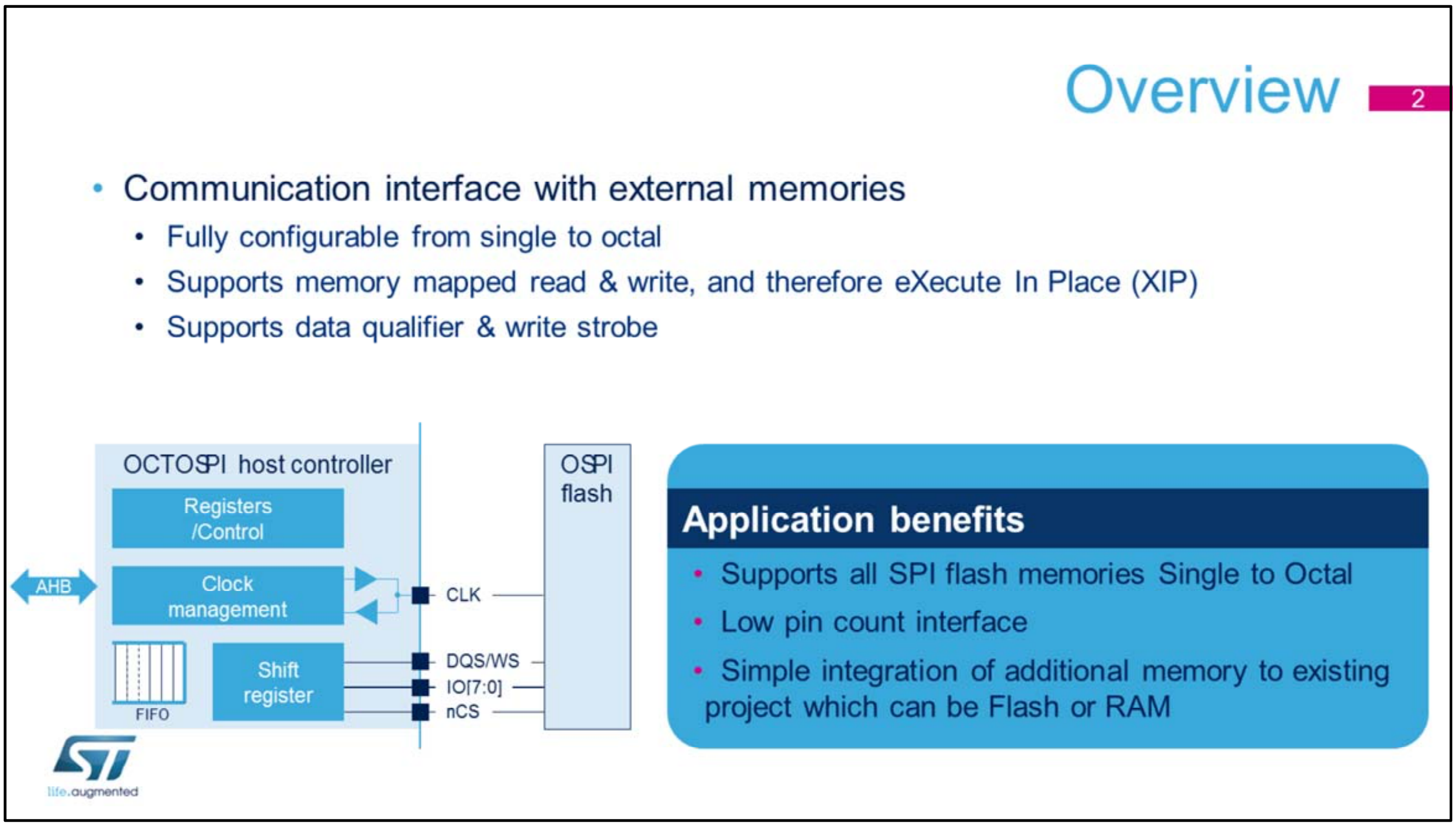
STM32L5 - OCTOSPI

OctoSPI interface

Revision 1.0

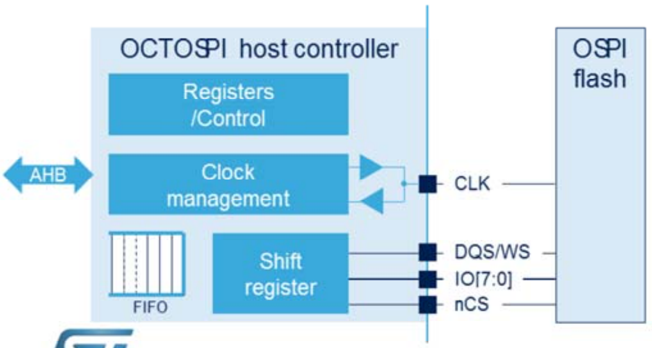
life.augmented

Hello, and welcome to this presentation of the STM32 OctoSPI interface that will present the features of this interface, which is widely used to connect external memories to the microcontroller.
OctoSPI was first implemented in the STM32L4+. The version present in the STM32L5 supports additional features.



Overview

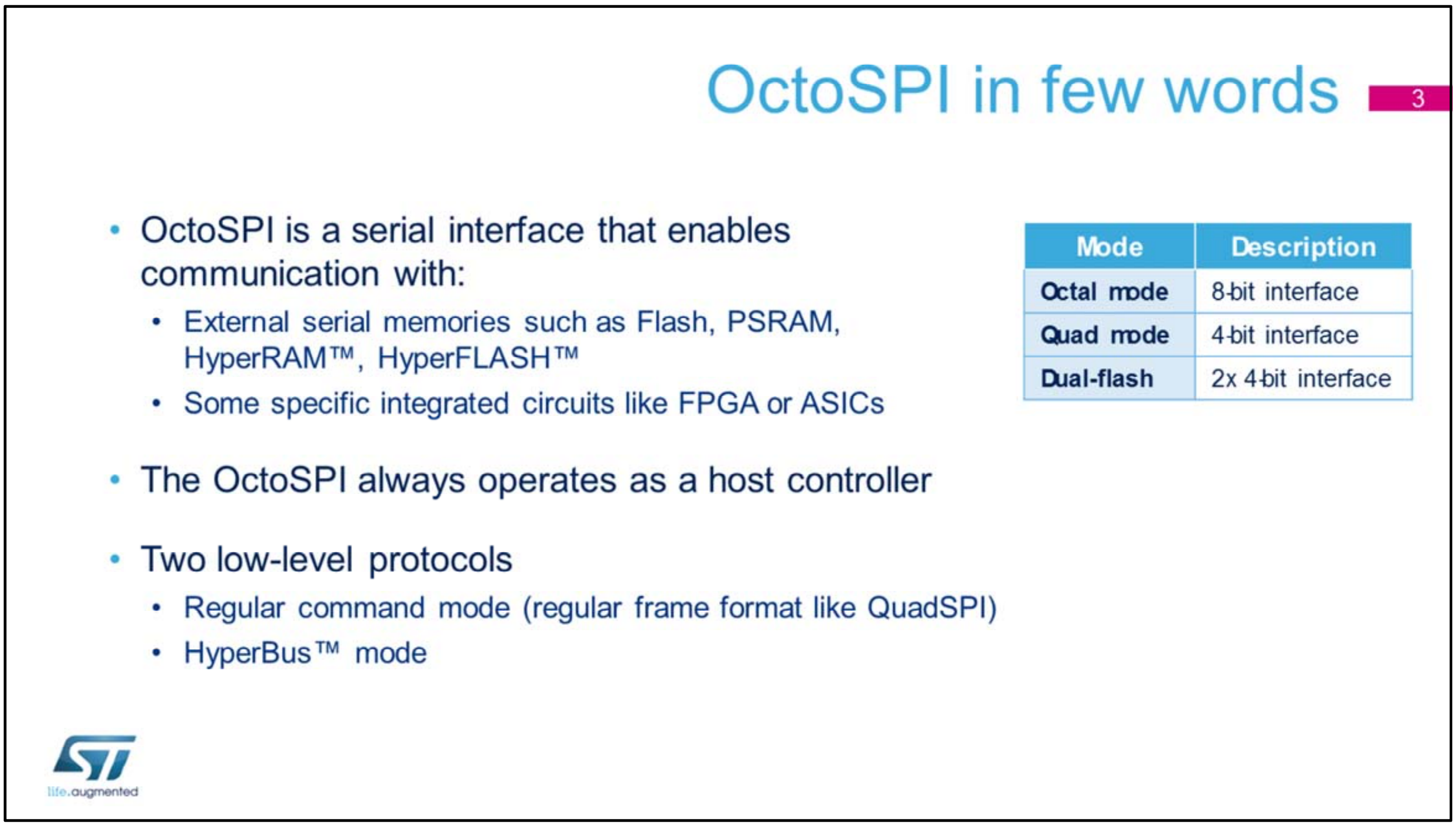
- Communication interface with external memories
  - Fully configurable from single to octal
  - Supports memory mapped read & write, and therefore eXecute In Place (XIP)
  - Supports data qualifier & write strobe



Application benefits

- Supports all SPI flash memories Single to Octal
- Low pin count interface
- Simple integration of additional memory to existing project which can be Flash or RAM

The OctoSPI interface integrated inside STM32 products provides a communication interface allowing the microcontroller to communicate with external single, dual, quad or octal SPI memories. This interface is fully configurable, allowing easy connection of any existing serial memories available today.
The external device can be memory mapped which allows any system master to access it, like internal memories, for read and write operations.
Applications will benefit from the easy connection of serial external memory, with only a few pins required. Thanks to the memory mapping feature, external memory could be simply accommodated in the existing project when more memory is needed whether it be Flash or RAM.



OctoSPI in few words

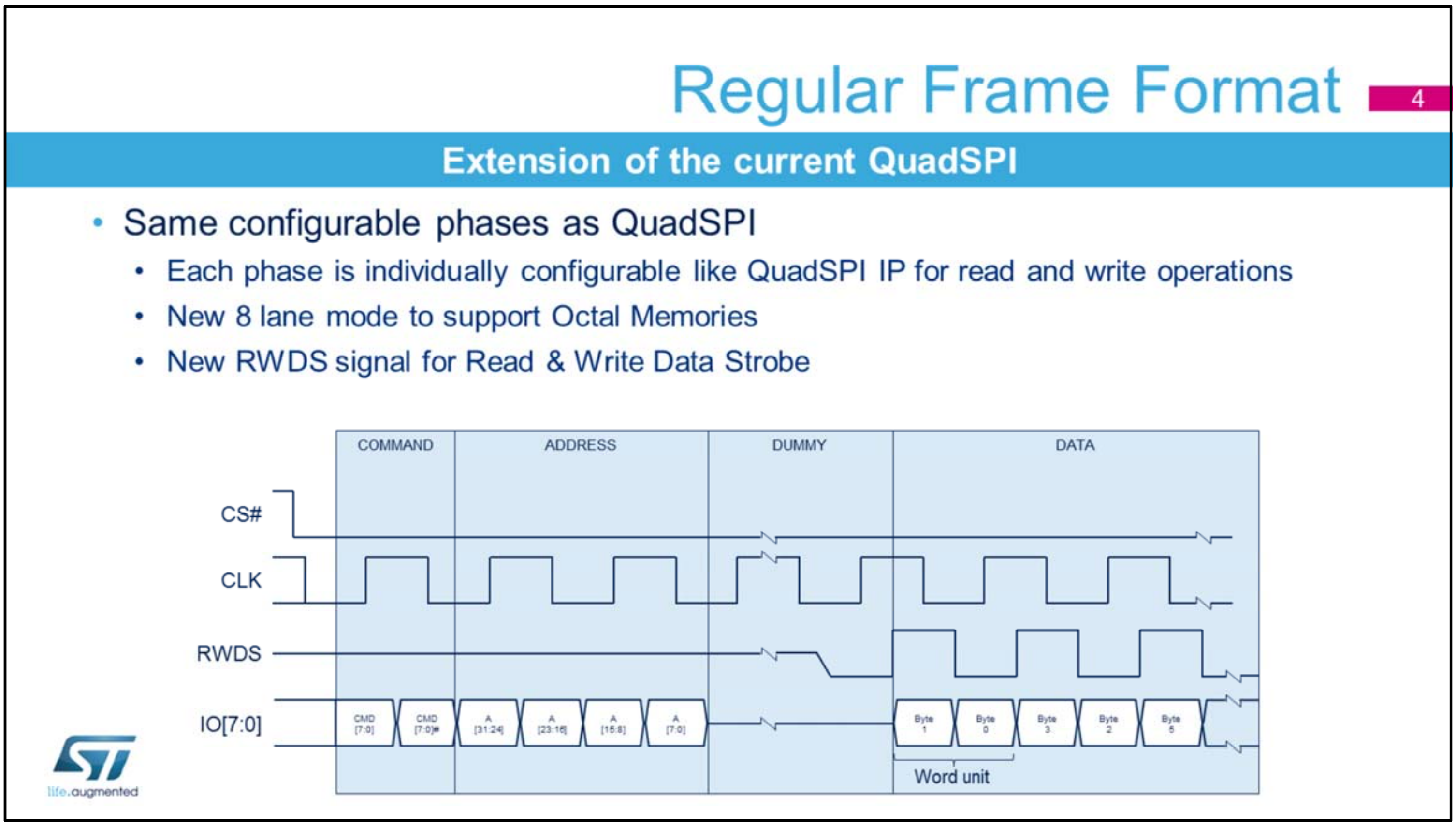
3

- OctoSPI is a serial interface that enables communication with:
  - External serial memories such as Flash, PSRAM, HyperRAM™, HyperFLASH™
  - Some specific integrated circuits like FPGA or ASICs
- The OctoSPI always operates as a host controller
- Two low-level protocols
  - Regular command mode (regular frame format like QuadSPI)
  - HyperBus™ mode

Mode	Description
Octal mode	8-bit interface
Quad mode	4-bit interface
Dual-flash	2x 4-bit interface

life.augmented

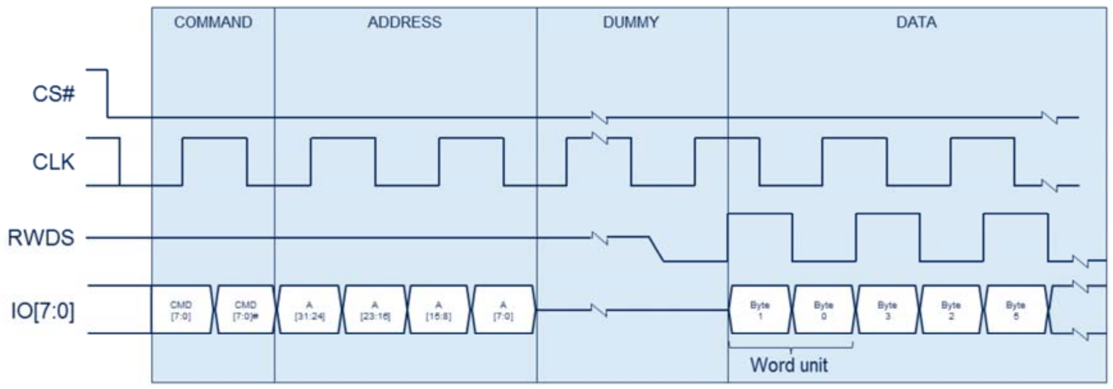
The OctoSPI is a serial interface that enables communication with serial memories.
The following protocols are supported, serial flash, psram, HyperRAM and HyperFlash.
The OctoSPI always operates as a host controller, it initiates the data transfer to the memory.
Two low level protocols are implemented: regular command mode, which is an extension of QuadSPI, and HyperBus protocol.
Three types of interfaces are supported: 8-bit, 4-bit and 2 times 4-bit.



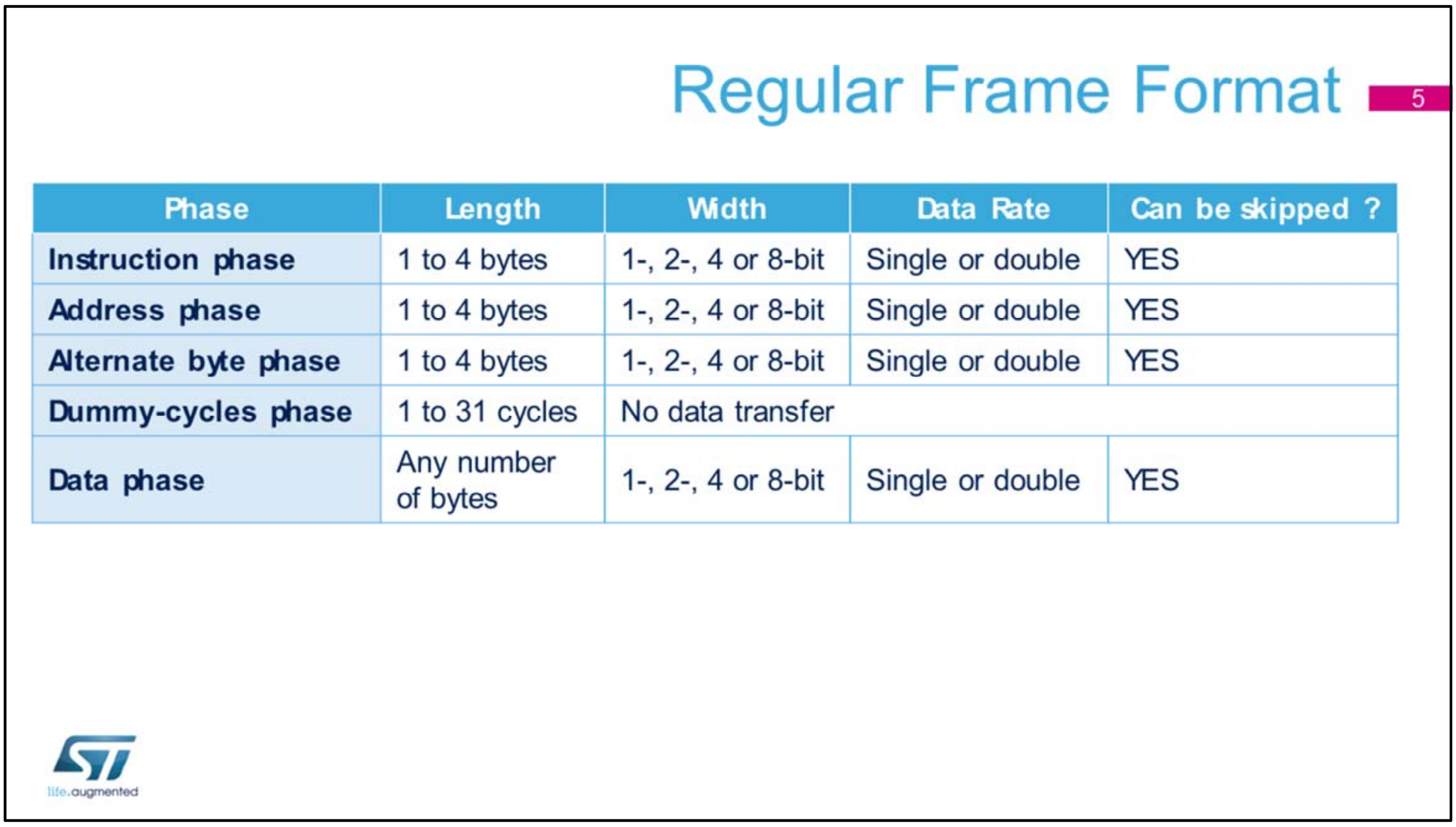
Regular Frame Format

Extension of the current QuadSPI

- Same configurable phases as QuadSPI
  - Each phase is individually configurable like QuadSPI IP for read and write operations
  - New 8 lane mode to support Octal Memories
  - New RWDS signal for Read & Write Data Strobe



The OctoSPI interface offers high flexibility for frame format configuration to address any serial Flash from single data lane up to 8 data lines.
As with regular QuadSPI, the user can enable or disable each of the phases, configure the length of each phase and configure the number of lines used for each phase from 1 to 8.
A new signal RWDS acts as either a write strobe during write operations or a read qualifier during read operations.



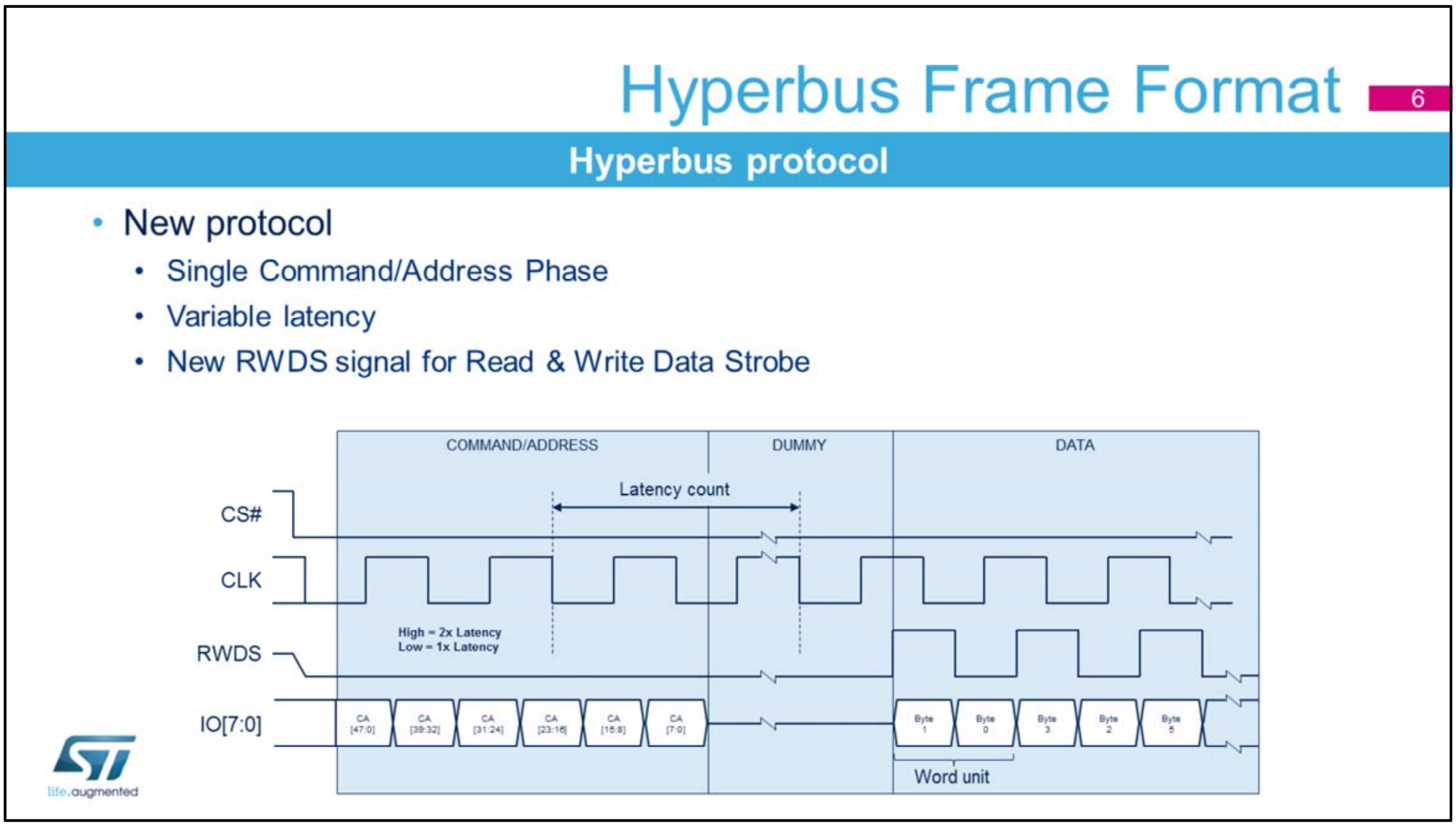
Regular Frame Format

Phase	Length	Width	Data Rate	Can be skipped ?
Instruction phase	1 to 4 bytes	1-, 2-, 4 or 8-bit	Single or double	YES
Address phase	1 to 4 bytes	1-, 2-, 4 or 8-bit	Single or double	YES
Alternate byte phase	1 to 4 bytes	1-, 2-, 4 or 8-bit	Single or double	YES
Dummy-cycles phase	1 to 31 cycles	No data transfer		
Data phase	Any number of bytes	1-, 2-, 4 or 8-bit	Single or double	YES

life.augmented

This table summarizes the features of each transfer phase.
The instruction phase transmits an instruction to the memory device, specifying the type of operation to be performed.
The address phase transmits the address of the operation.
In the alternate-bytes phase, 1 to 4 bytes are sent to the external device, generally to control the mode of operation.
In the dummy-cycles phase, 1 to 31 cycles are given without any data being sent or received, in order to give the external device the time to prepare for the data phase when the higher clock frequencies are used.
During the data phase, any number of bytes can be sent to or received from the external device
Any of these phases can be configured to be skipped,

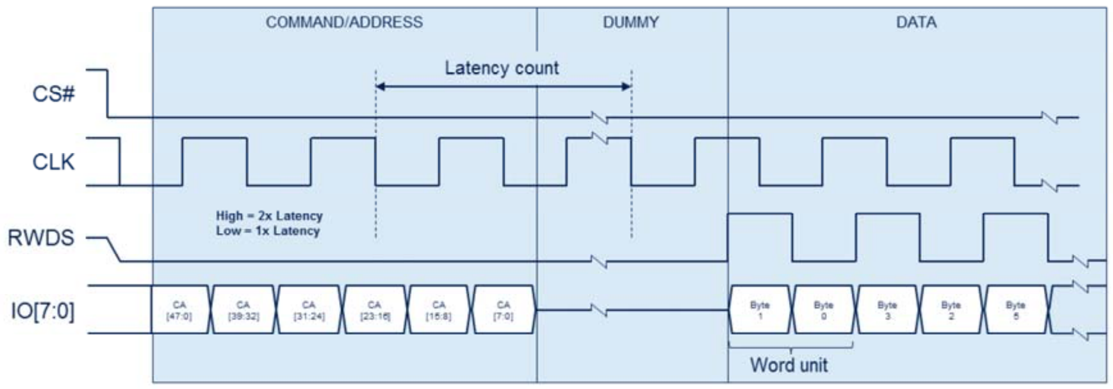
but at least one of the instruction, address, alternate byte, or data phases must be present



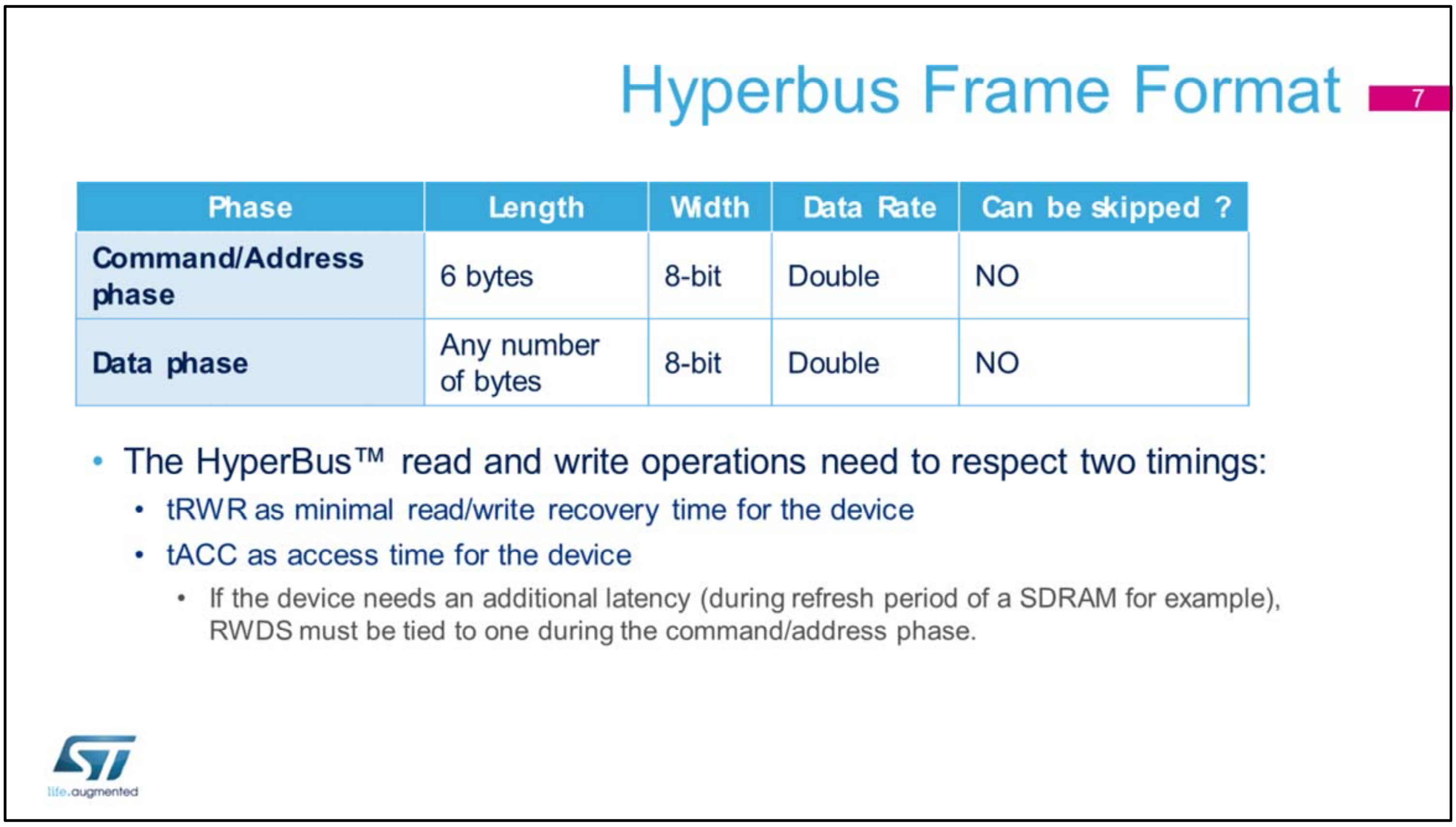
Hyperbus Frame Format

Hyperbus protocol

- New protocol
  - Single Command/Address Phase
  - Variable latency
  - New RWDS signal for Read & Write Data Strobe



The OctoSPI supports the new Hyperbus mode which combines the command and the addresses in a single initial phase.
The HyperBus does not require any command specification nor any alternate bytes.
As with the regular frame format, Hyperbus mode also uses a read qualifier and a write strobe during the data operations.
The OctoSPI supports variable or fixed external memory latency as defined by the Hyperbus protocol specification.



Hyperbus Frame Format

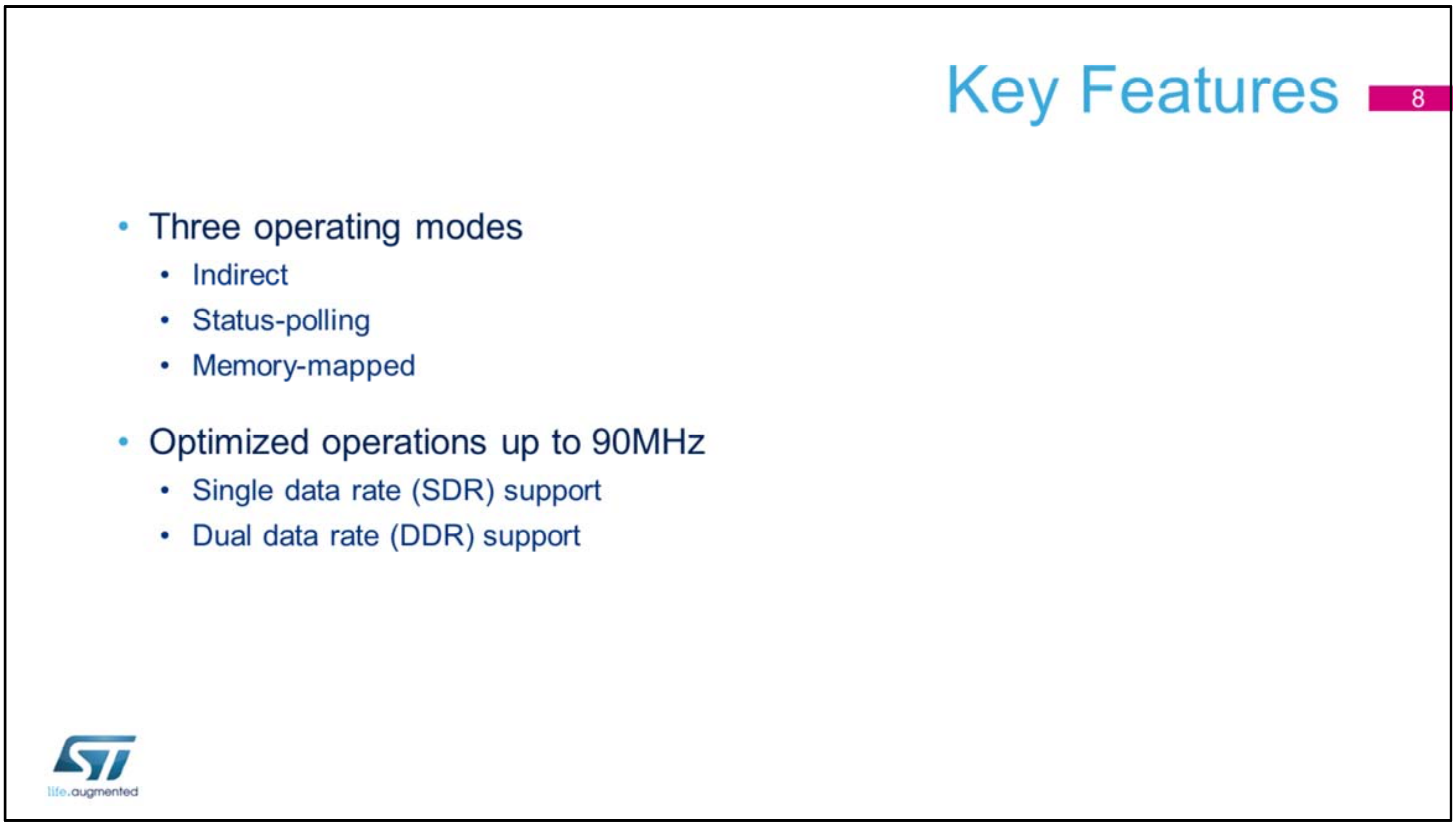
Phase	Length	Width	Data Rate	Can be skipped ?
Command/Address phase	6 bytes	8-bit	Double	NO
Data phase	Any number of bytes	8-bit	Double	NO

- The HyperBus™ read and write operations need to respect two timings:
  - tRWR as minimal read/write recovery time for the device
  - tACC as access time for the device
    - If the device needs an additional latency (during refresh period of a SDRAM for example), RWDS must be tied to one during the command/address phase.

The HyperBus™ frame is composed of two phases:
- Command/address phase
- Data phase.

This table summarizes the features of each of these phases.
The command and address phase transmits both a command and an address to the memory device.
48 bits are transferred, selecting the data transfer direction, the burst type, linear or wrap, the address space, memory or registers and the address.
During the data phase, any number of bytes can be sent to or received from the external device.
Two timings have to be configured according to the characteristics of the Hyperbus memory device:
tRWR which is the device read write recovery time and tACC which is the Device access time.
If the device needs an additional latency, RWDS must be

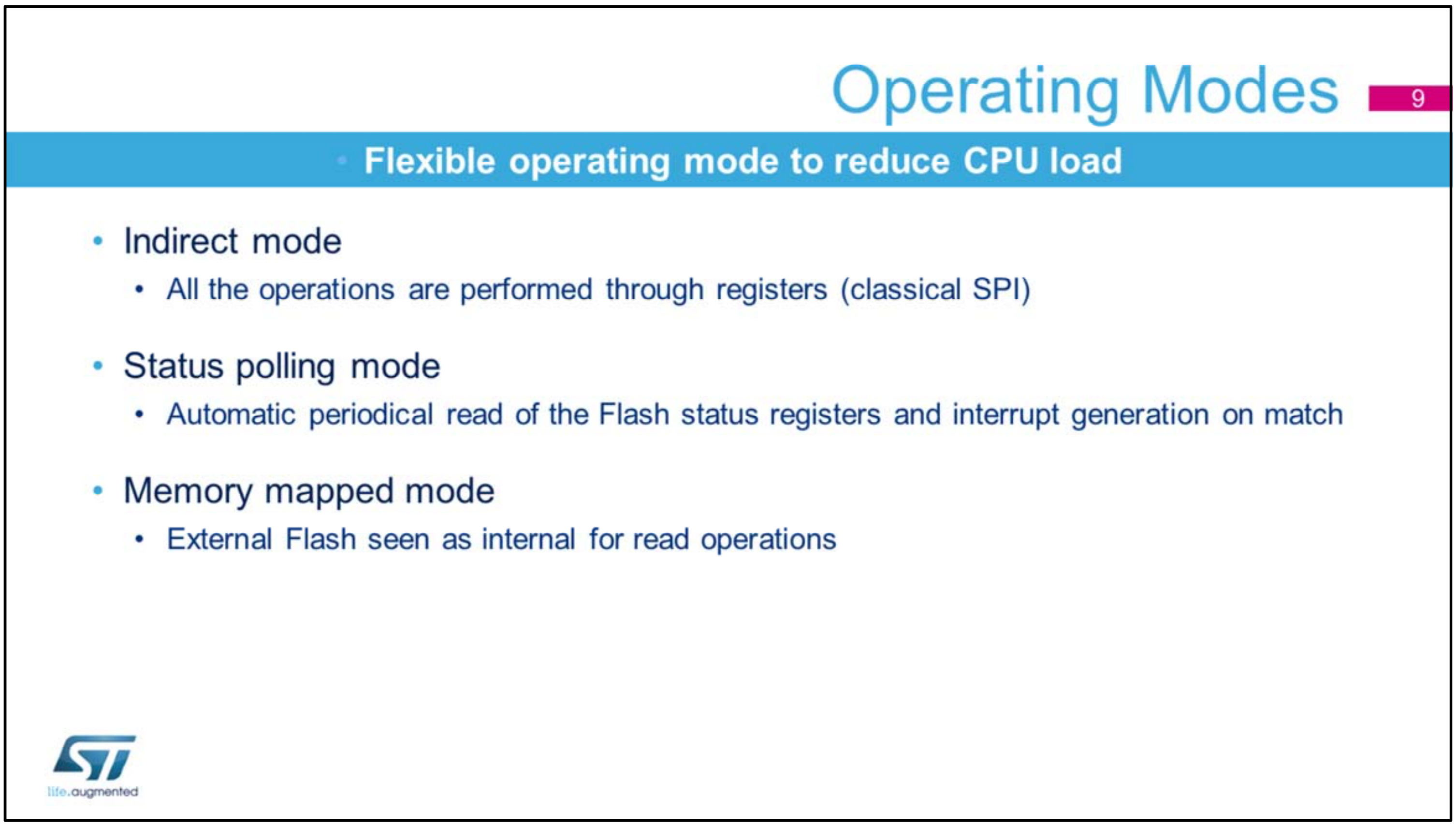
tied to one during the command/address phase.



Key Features

- Three operating modes
  - Indirect
  - Status-polling
  - Memory-mapped
- Optimized operations up to 90MHz
  - Single data rate (SDR) support
  - Dual data rate (DDR) support

The OctoSPI integrated inside STM32 products offers three operating modes which will be explained later in this presentation.
Communication with external memories supports single or dual data rate operation.



Operating Modes

9

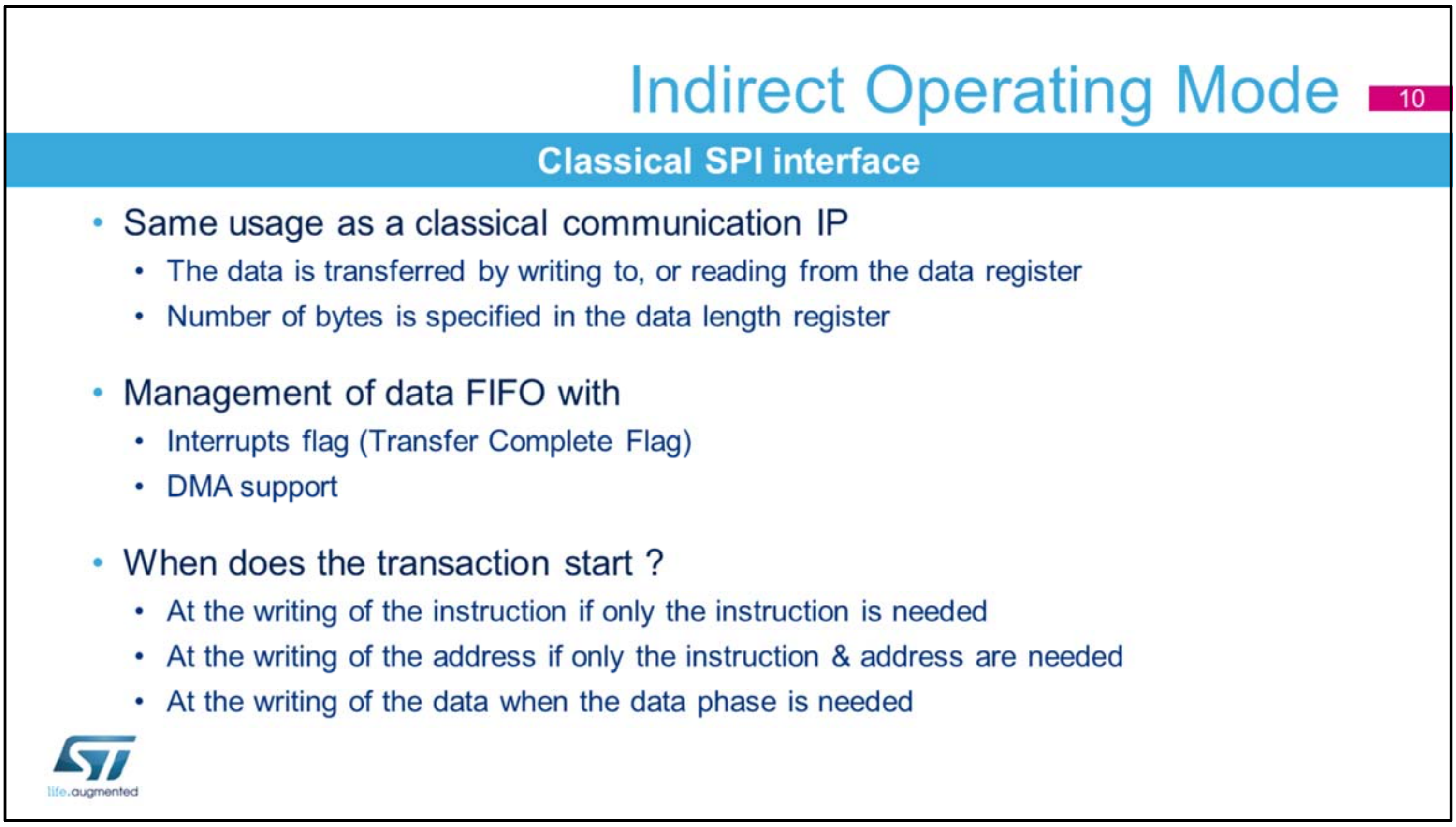
- Flexible operating mode to reduce CPU load

- Indirect mode
  - All the operations are performed through registers (classical SPI)
- Status polling mode
  - Automatic periodical read of the Flash status registers and interrupt generation on match
- Memory mapped mode
  - External Flash seen as internal for read operations

life.augmented

The OctoSPI supports three different modes of operation :

- Indirect mode, where it behaves as a classical SPI interface and all operations are performed through registers,
- Status polling mode, where the Flash status registers are read periodically with interrupt generation
- Memory mapped mode, where external memory is seen as if it is internal memory for read operations.



Indirect Operating Mode

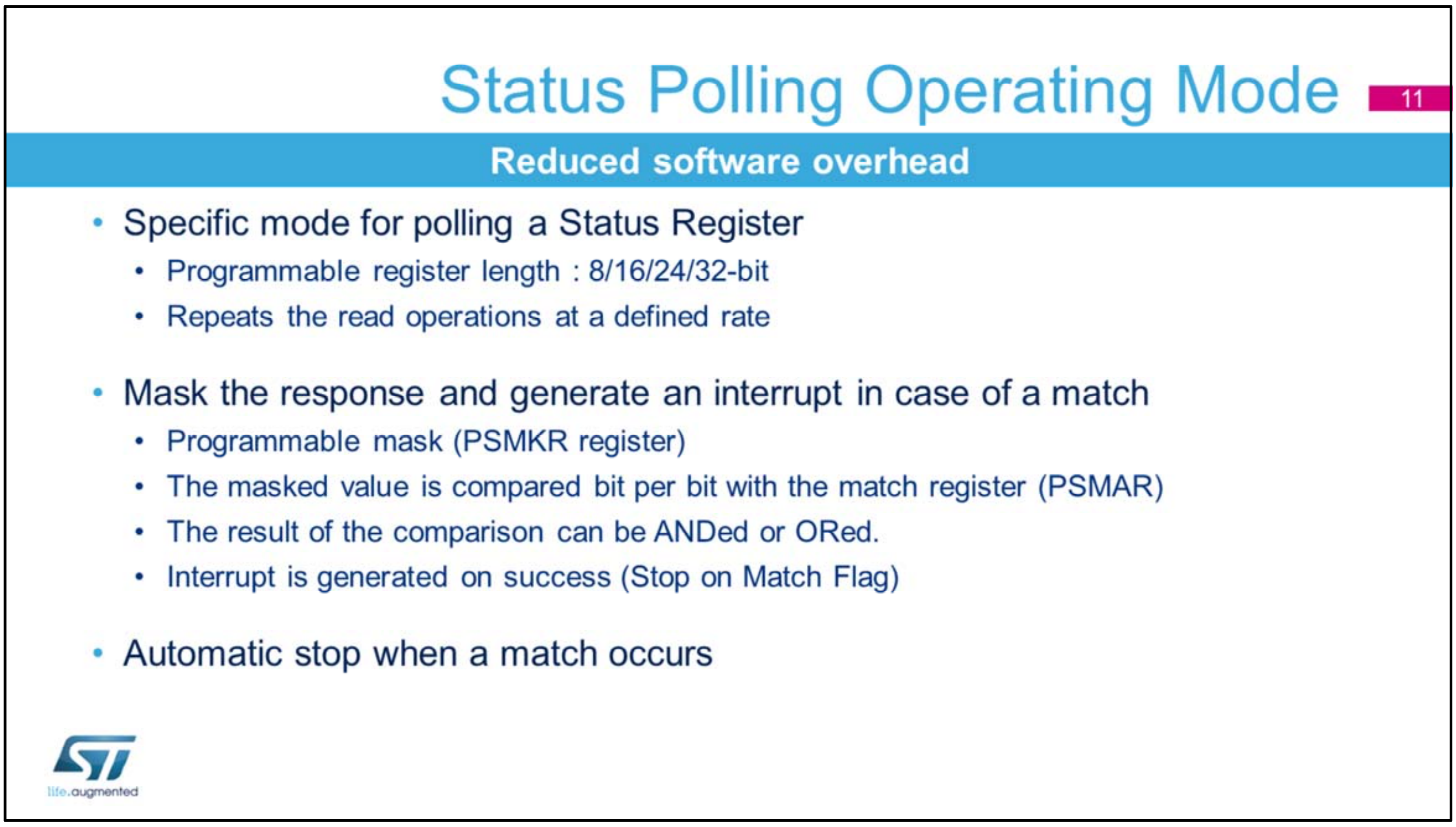
10

Classical SPI interface

- Same usage as a classical communication IP
  - The data is transferred by writing to, or reading from the data register
  - Number of bytes is specified in the data length register
- Management of data FIFO with
  - Interrupts flag (Transfer Complete Flag)
  - DMA support
- When does the transaction start ?
  - At the writing of the instruction if only the instruction is needed
  - At the writing of the address if only the instruction & address are needed
  - At the writing of the data when the data phase is needed

life.augmented

In indirect operating mode, the OctoSPI behaves like a classical SPI interface. Transferred data goes through the data register via a FIFO. Data exchange is driven by software or by the DMA, using related interrupt flags in the OctoSPI status registers.
Each command is launched by the writing of an instruction, address or data depending on the instruction context



Status Polling Operating Mode

Reduced software overhead

- Specific mode for polling a Status Register
  - Programmable register length : 8/16/24/32-bit
  - Repeats the read operations at a defined rate
- Mask the response and generate an interrupt in case of a match
  - Programmable mask (PSMKR register)
  - The masked value is compared bit per bit with the match register (PSMAR)
  - The result of the comparison can be ANDed or ORed.
  - Interrupt is generated on success (Stop on Match Flag)
- Automatic stop when a match occurs

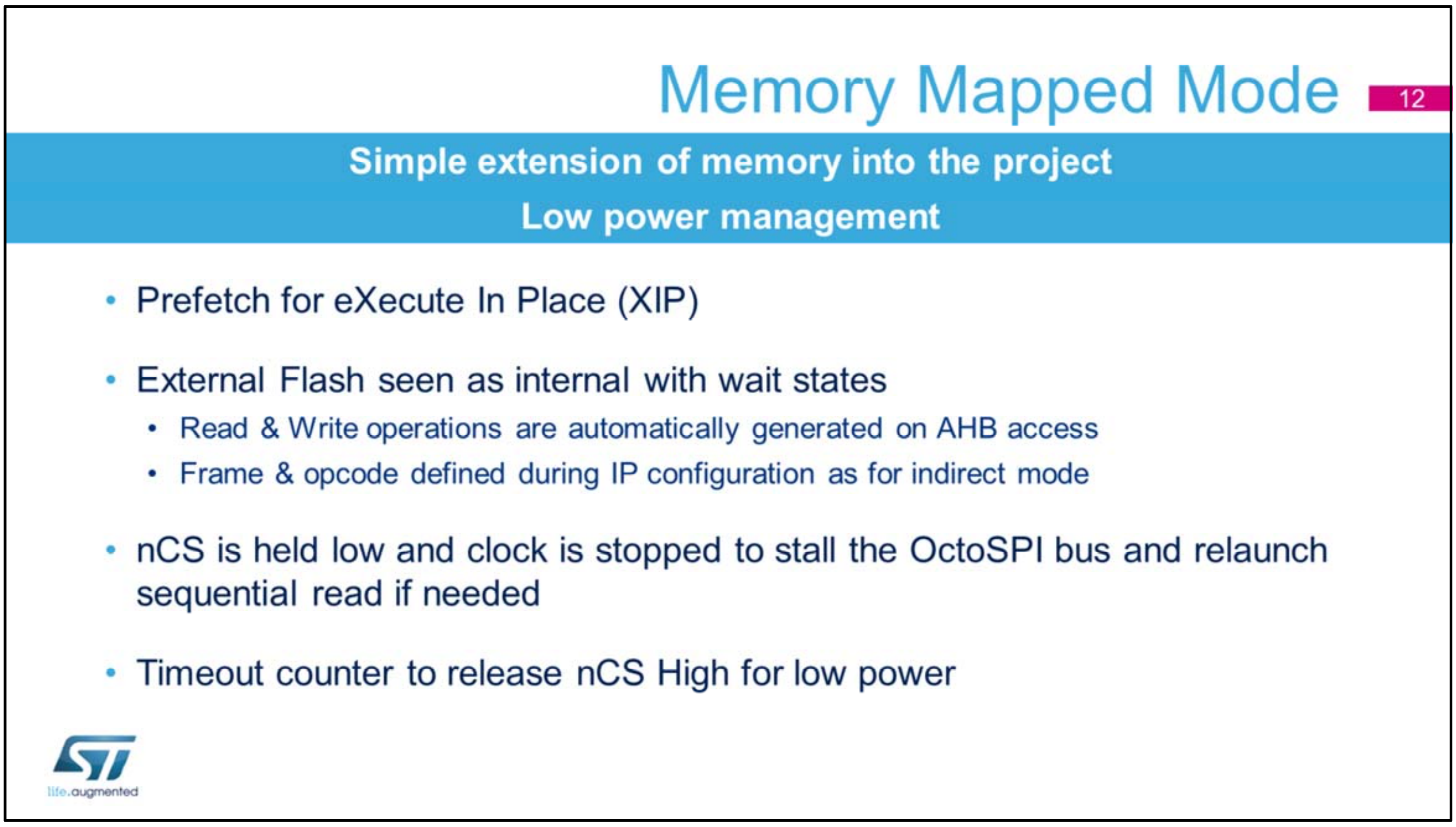
A specific mode as been implemented in the OctoSPI interface to autonomously poll status registers in the external Flash.

The OctoSPI interface can be configured to periodically read a register in the external Flash.

The returned data can be masked to select the bits to be evaluated. The selected bits are compared with their required values stored in the match register. The result of the comparison can be treated in two ways:

- In ANDed mode, if all the selected bits match, an interrupt is generated.
- In Ored mode, if one of the selected bits matches, an interrupt is generated.

When a match occurs, the OctoSPI interface can stop automatically.


Memory Mapped Mode

12

Simple extension of memory into the project

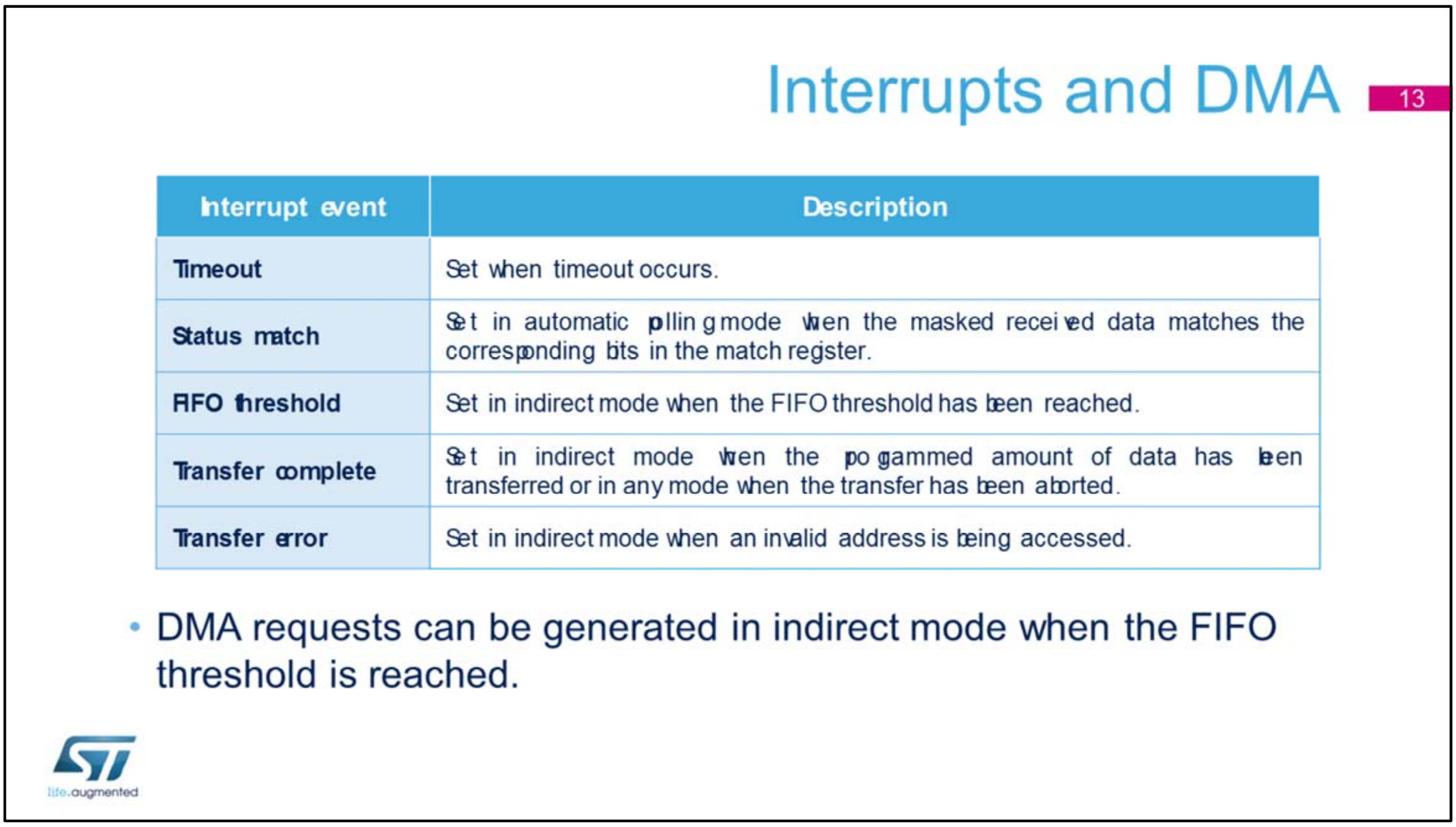
Low power management

- Prefetch for eXecute In Place (XIP)
- External Flash seen as internal with wait states
  - Read & Write operations are automatically generated on AHB access
  - Frame & opcode defined during IP configuration as for indirect mode
- nCS is held low and clock is stopped to stall the OctoSPI bus and relaunch sequential read if needed
- Timeout counter to release nCS High for low power

life.augmented

The OctoSPI also provides a memory mapped mode.
The main application benefit introduced by this mode is the simple integration of an external memory extension with no difference between read or write accesses of internal or externally connected memory, except for the number of wait states.
This mode is suitable for both read and write operations and external memories, whether it be RAM or Flash, are seen as internal memory with wait states included to compensate for the lower speed of the external memory.
The maximum size supported by this mode is limited to 256 MB.
A prefetch buffer supports the local execution, therefore the code could be executed directly from the external memory without the need to download it into the internal RAM.
This mode supports also a mode called Send Instruction

Only Once mode, known as SIOO, which is supported by some Flash memories.
It allows the controller to send instructions once only and remove the instruction phase for the following accesses.

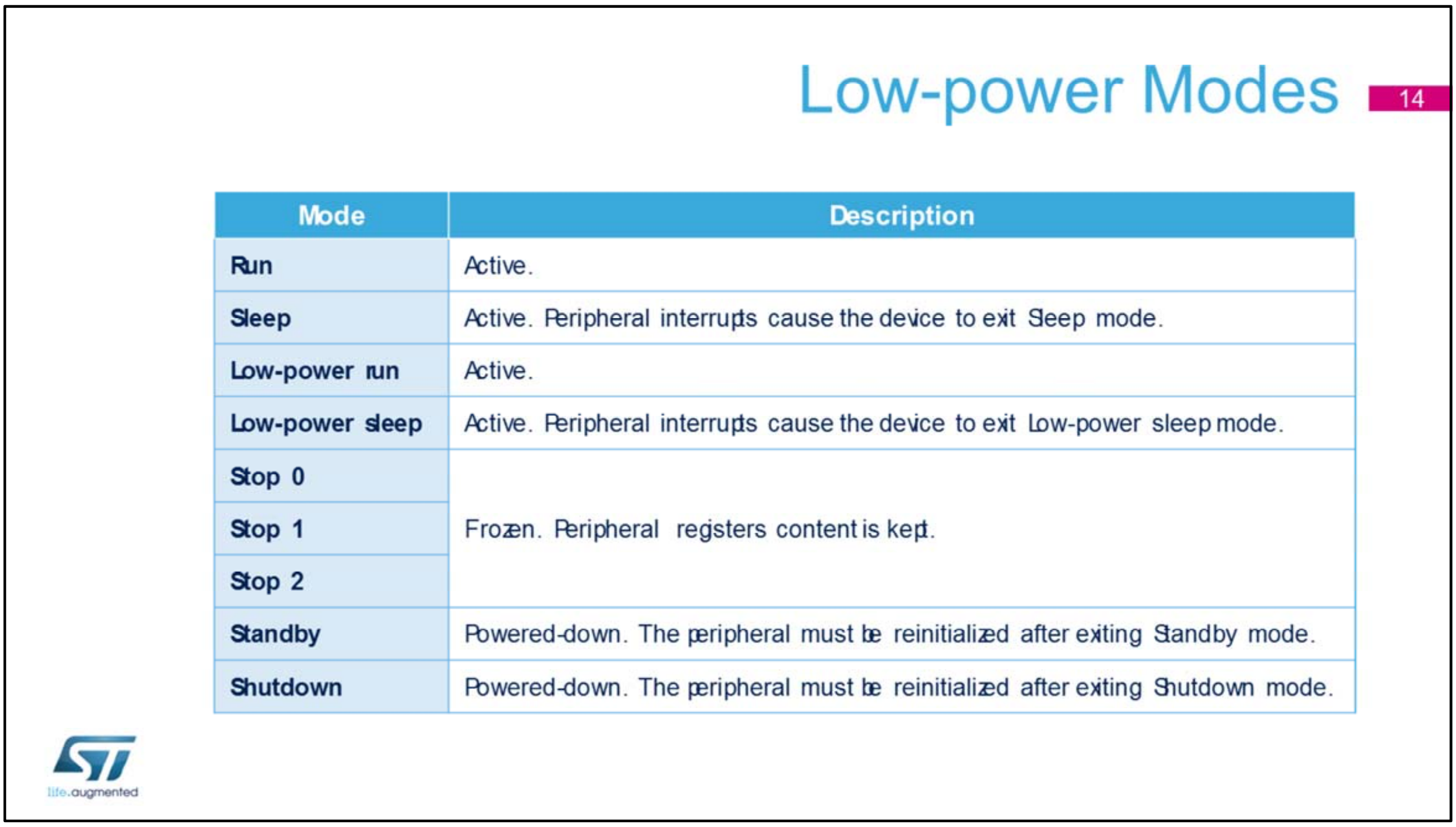


Interrupts and DMA

Interrupt event	Description
Timeout	Set when timeout occurs.
Status match	Set in automatic polling mode when the masked received data matches the corresponding bits in the match register.
FIFO threshold	Set in indirect mode when the FIFO threshold has been reached.
Transfer complete	Set in indirect mode when the programmed amount of data has been transferred or in any mode when the transfer has been aborted.
Transfer error	Set in indirect mode when an invalid address is being accessed.

- DMA requests can be generated in indirect mode when the FIFO threshold is reached.

The OctoSPI has 5 interrupt sources: Timeout, Status match when the masked received data matches the corresponding bits in the match register in automatic polling mode, FIFO Threshold, Transfer complete and Transfer error.
DMA requests can be generated in indirect mode when the FIFO threshold has been reached.

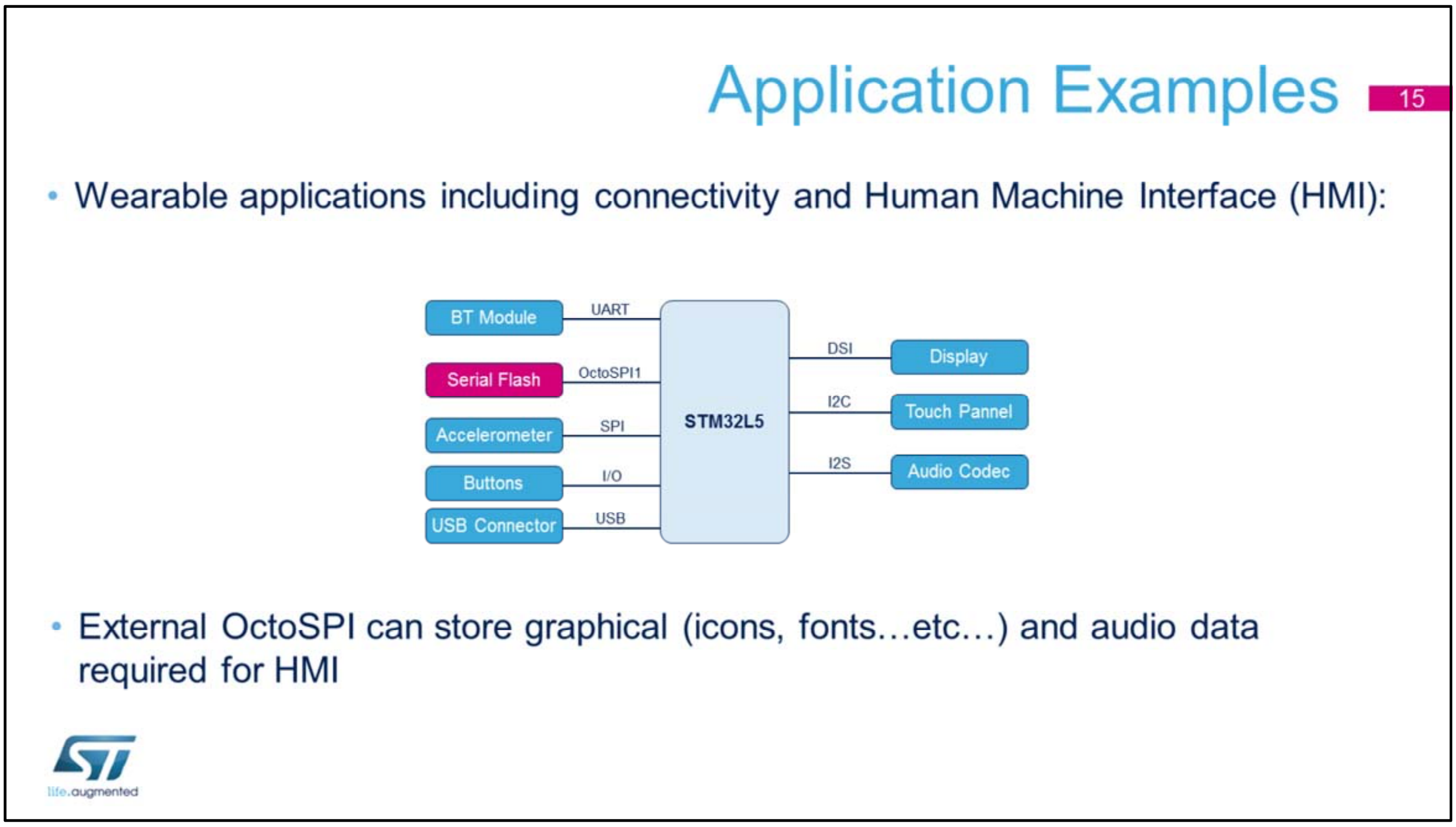


Low-power Modes

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	Frozen. Peripheral registers content is kept.
Stop 1	
Stop 2	
Standby	Powered-down. The peripheral must be reinitialized after exiting Standby mode.
Shutdown	Powered-down. The peripheral must be reinitialized after exiting Shutdown mode.

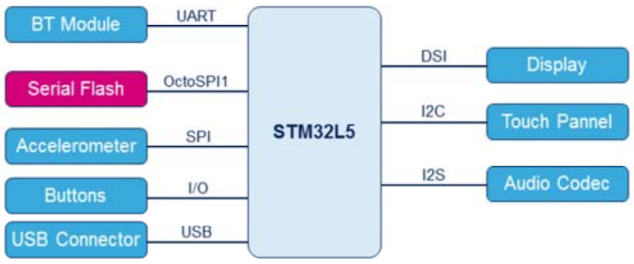
life.augmented

The OctoSPI is active in Run, Sleep, Low-power run and Low-power sleep mode. An OctoSPI interrupt can cause the device to exit Sleep or Low-power sleep mode.
In Stop0, Stop1 or Stop2 mode, the OctoSPI is frozen, and its registers content is maintained.
In Standby or Shutdown mode, the OctoSPI is powered-down and it must be reinitialized afterward.



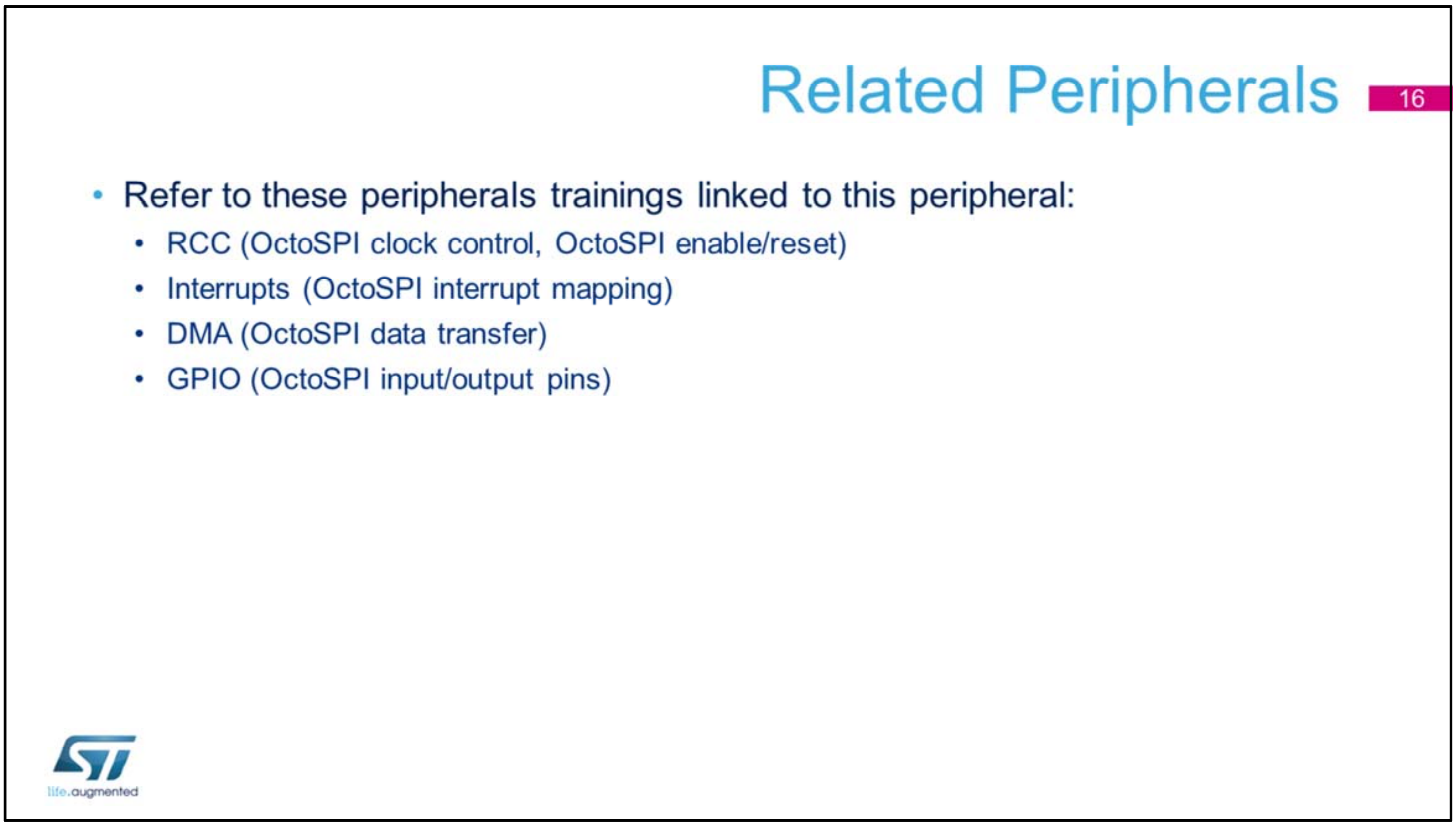
Application Examples

- Wearable applications including connectivity and Human Machine Interface (HMI):



- External OctoSPI can store graphical (icons, fonts…etc…) and audio data required for HMI

Wearable applications are requiring low-power management together with high quality HMI. This can be achieved using the STM32L5 OctoSPI interface to store in an external Flash all the graphical content needed like background images, high resolution icons, or fonts to support multiple languages. Additional audio data for ringtones can also benefit from the large space offered by the external Flash. The low pin count needed to drive such devices allows a very optimized system integration.

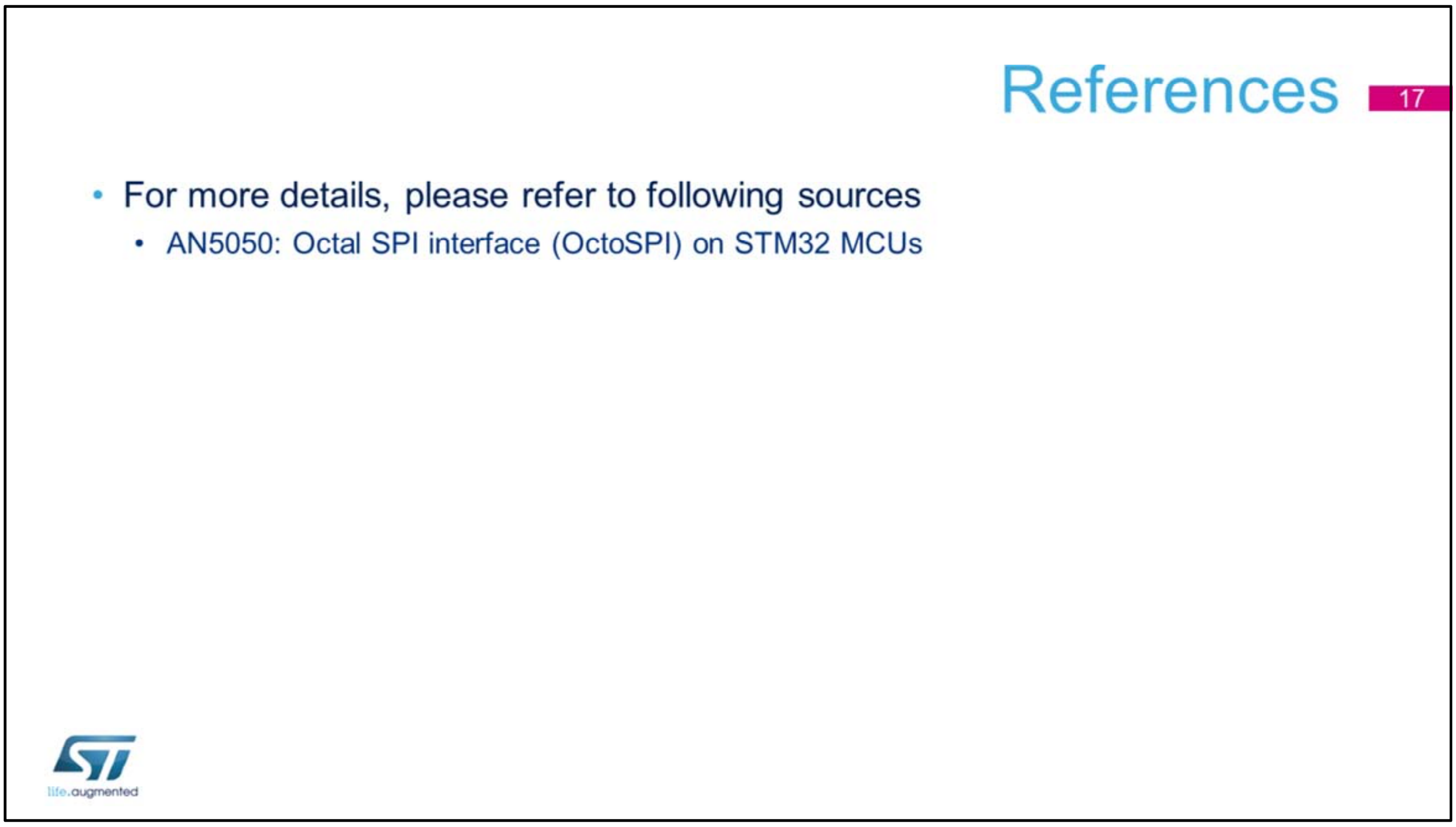


Related Peripherals

- Refer to these peripherals trainings linked to this peripheral:
  - RCC (OctoSPI clock control, OctoSPI enable/reset)
  - Interrupts (OctoSPI interrupt mapping)
  - DMA (OctoSPI data transfer)
  - GPIO (OctoSPI input/output pins)

life.augmented

You can refer to peripheral training slides related to RCC, interrupts, DMA and GPIO for additional information.



References

- For more details, please refer to following sources
  - AN5050: Octal SPI interface (OctoSPI) on STM32 MCUs

life.augmented

For more details, please have a look into the application note AN5050 about the Octal SPI interface.