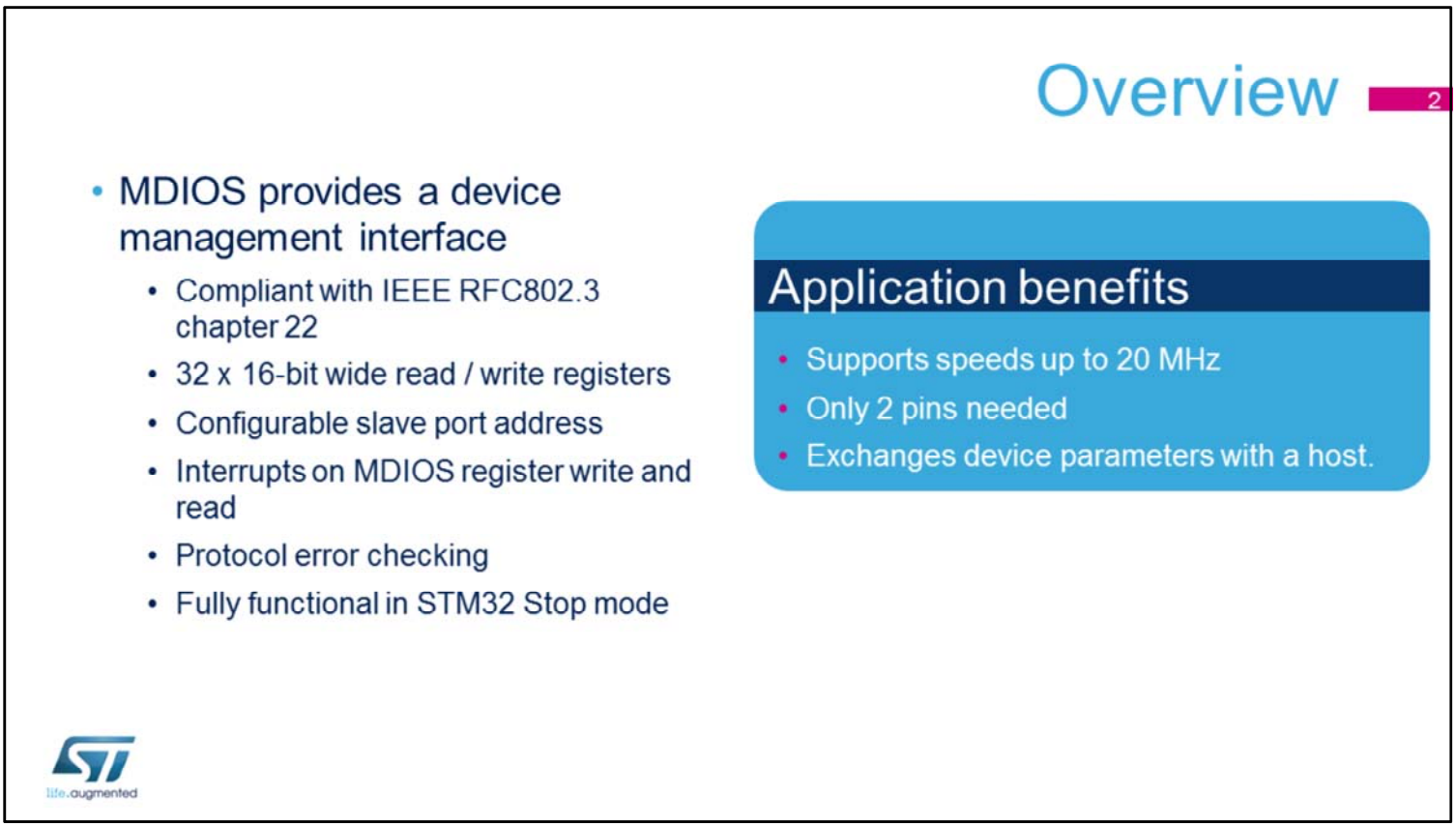
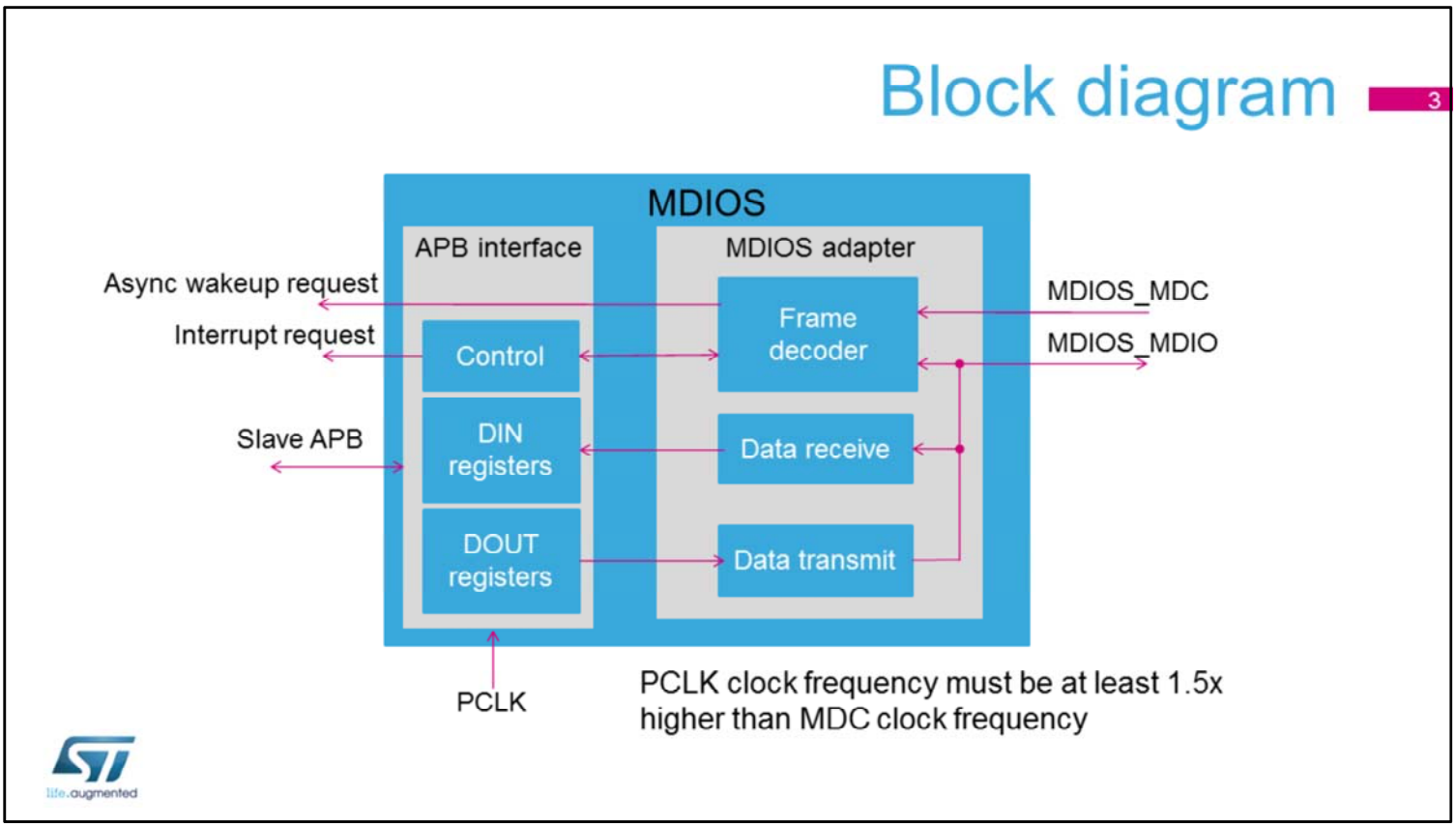
Hello, and welcome to this presentation of the STM32 management data input/output slave controller or MDIOS controller module. It covers the main features of the controller which is used to exchange management data with a host device.



The MDIOS controller integrated inside STM32 products provides a device management interface allowing a host to manage the STM32 configuration. It offers 32 x 16-bit wide registers. Applications benefit from a low pin count standard interface to manage the device configuration. The communication speed can go as high as 20 MHz.



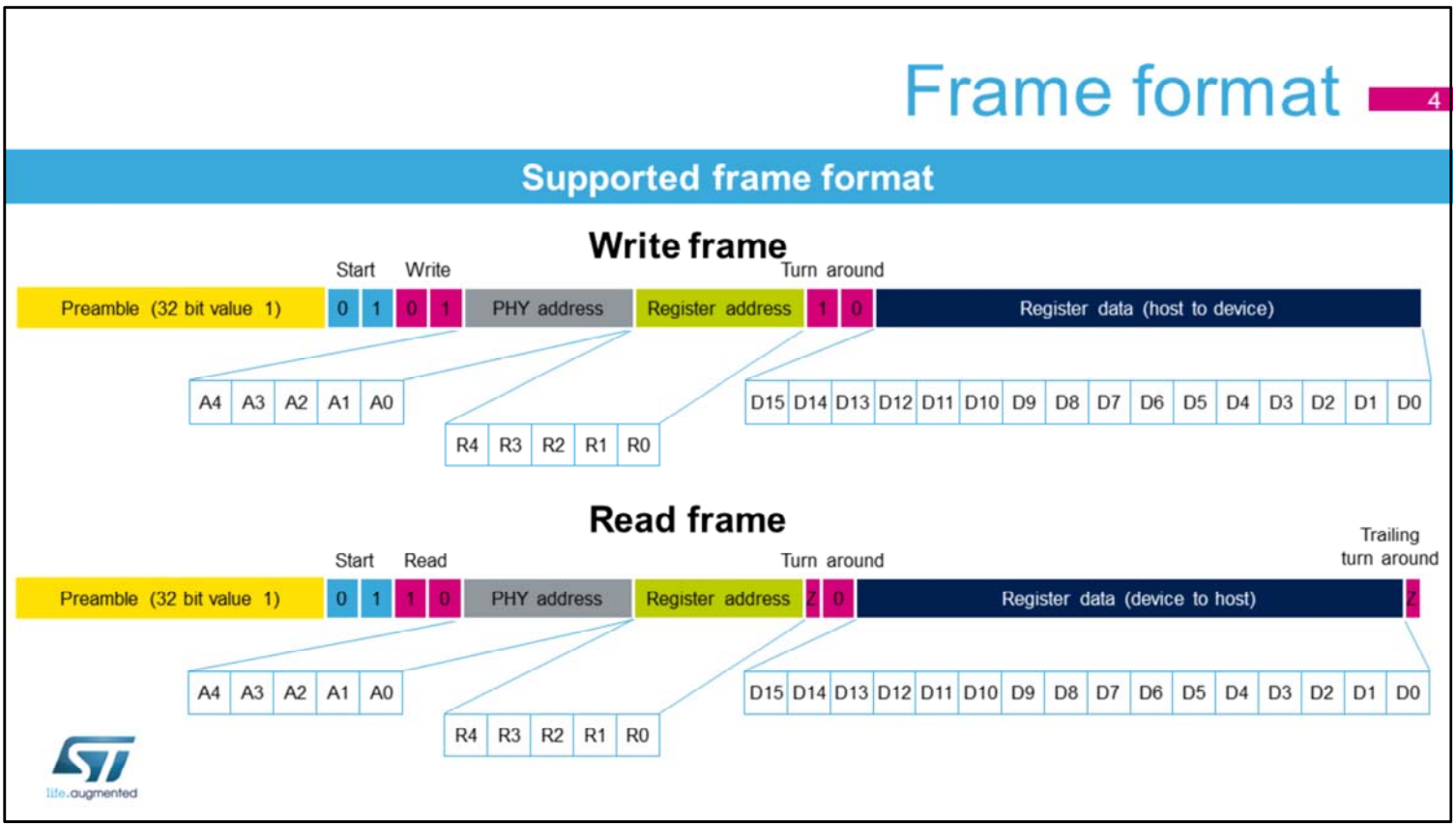
The MDIOS controller provides all the functions specified in IEEE RFC 802.3 chapter 22 needed to interface with a host.

It consists of an “MDIOS Adapter" and an "Advanced Peripheral Bus or APB interface".

The “MDIOS adapter” provides functions such as frame decoding and checking, asynchronous wakeup and interrupt generation while the “APB interface” manages the control and status registers, data in registers and data out registers and synchronous interrupt requests.

Two clocks are available for the MDIOS controller, the APB clock (PCLK) for the “APB interface” and the MDIOS_MDC bus clock for the “MDIOS adapter”.

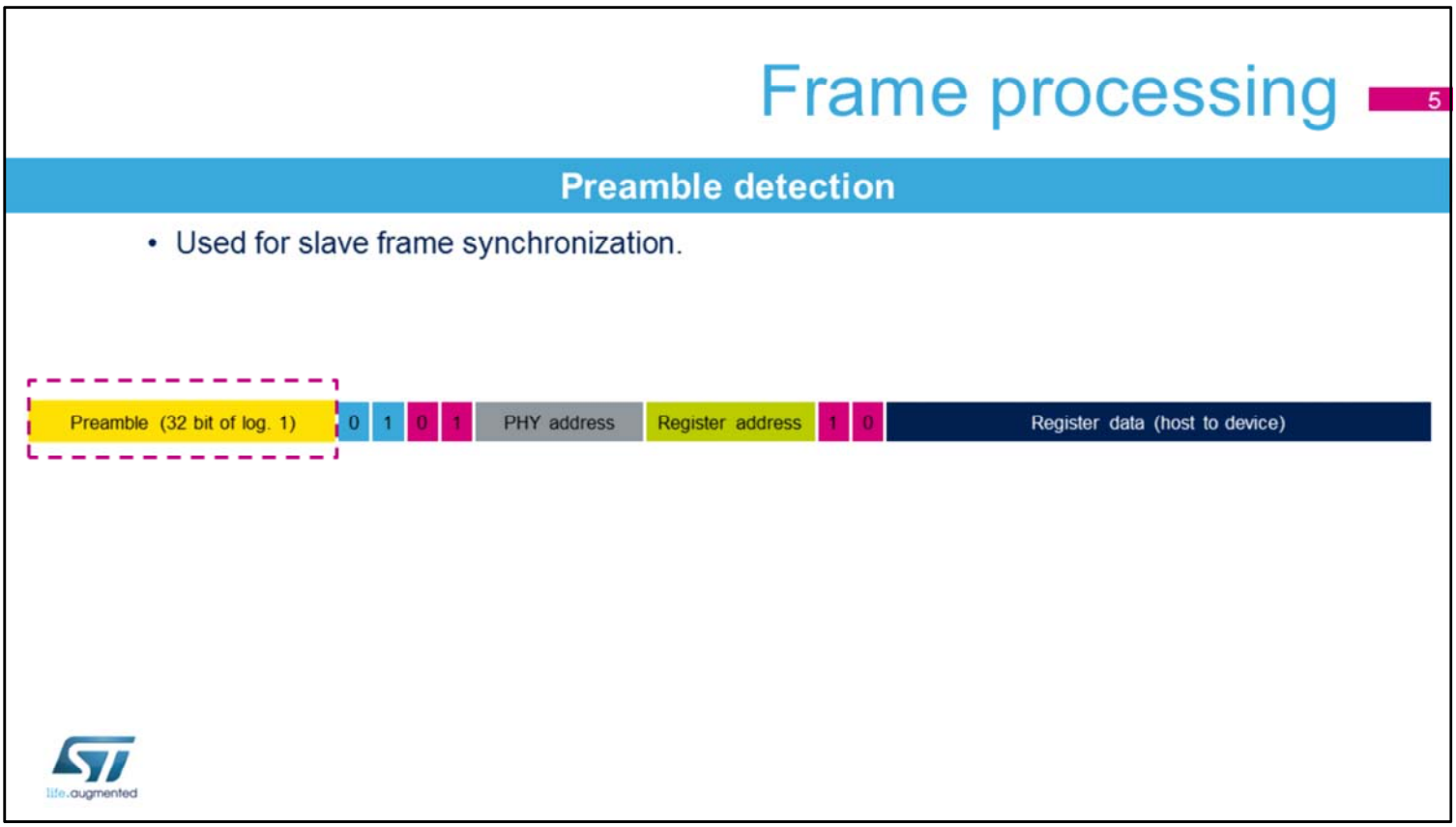
Note that PCLK clock frequency shall be at least 1.5x times higher than the MDC clock frequency. When the MDC bitrate is 20MHz, the PCLK shall run at least at 30MHz.



The MDIO frame format is sent over the MDIO pin with an active MDC clock. The clock may be stopped outside the frame format.
The host sends MDIO data on the falling edge of the MDC clock signal.

A preamble consisting of 32 bits with MDIO high is clocked in prior to the frame start.
The frame start is encoded as 2-bit value (0b01) (pronounced “zero” “one”- do not mention the “0b”).
The frame operation is selected by a 2-bit field: 0b01 for a register write and 0b10 for a register read.
Then MDIO slave device address and register address are sent. Up to 32 device registers can be addressed.
Subsequently for a write frame, the host sends a 2-bit turn-around code (0b10) followed by the 16-bit register data.

When the host issues read command, it changes the MDIO direction to input in the first turn-around cycle and the slave device drives the MDIO in the second cycle to ‘0’.
The slave outputs the data on the rising edge of the MDC clock signal.
Following the second turn-around bit, the slave device sends the register data and when finished, in the trailing turn-around period, it changes the MDIO direction to input again.



When enabled, the MDIOS monitors the MDIO interface for an incoming preamble.
At least 32 bits on MDIO shall be received with value ‘1’ to detect a valid preamble. This allows the slave device to synchronize with the MDIO bus.
Once synchronized, the 32-bit preamble is required after any received frame.
Only after the MDIOS is synchronized, the preamble errors between received frames are reported by the PERF bit.

Once the MDIOS is synchronized with the MDIO interface, the preamble check can be disabled by the DPC bit, allowing the host to send frames without preamble.

Frame processing

Start condition detection

- 4-bit frame start
  - Detection of frame start and frame operation (write or read).

A start condition is detected when a MDIO bit is set to ‘0’

Valid start conditions are 0b0101 for a write frame and ‘0b0110 for a read frame.
Start condition errors are reported by the SERF bit.

The Start condition will be processed by the MDIOS only when a valid preamble has been detected or when preamble detection has been disabled.

Frame processing

7

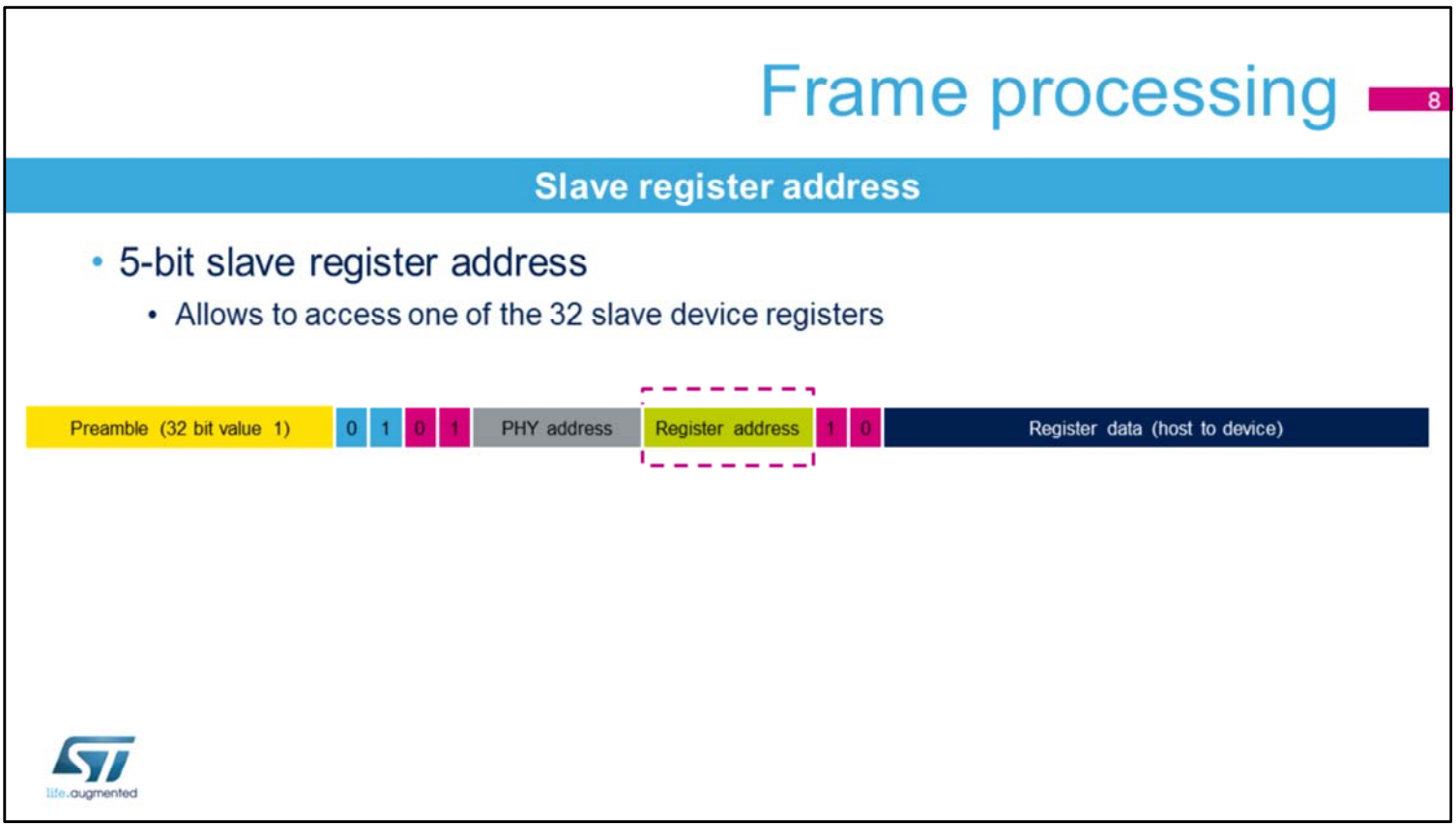
Slave physical address detection

- 5-bit slave physical address
  - Allows to determine if the frame addresses the correct device, because multiple devices may be present.

The physical address allows frames to be sent to different devices on the same MDIO bus.
The MDIO device slave address is programmed in the MDIOS PORT_ADDRESS register.

When a frame with a matching physical address is received, it will be further processed by the device.
Frames with a different physical address are discarded.

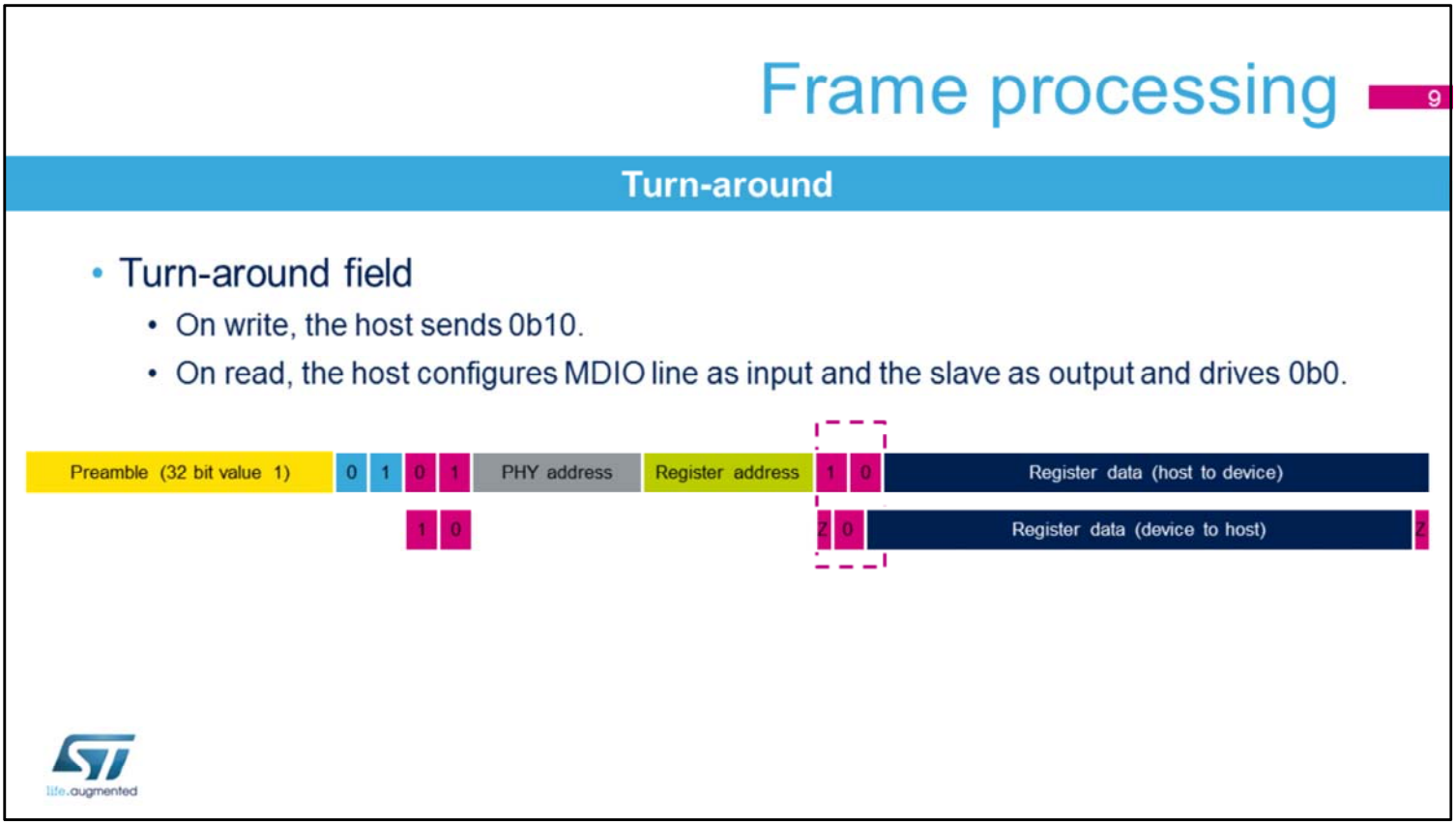
The physical address will only be processed by the MDIOS when a valid start condition has been detected.



The register address allows the host to access one of the 32 slave registers for write or read.

The register address will only be processed by the device MDIOS when the frame's physical address matches.

Note on STM32 devices where the MDIOS implements less than 32 registers, if the host writes to a register which is not implemented, data will be lost and on read, '0' will be returned.



When the host reads the data from slave:

- The turn-around field is used in read frames to hand over the MDIO line from the host to the slave.
- The turn-around time is only 1 ½ cycle long, where ½ a cycle is used by the host to switch its MDIO pin to input and 1 cycle by the slave to drive MDIO with ‘0’. There is no turn-around error generated for read frames.

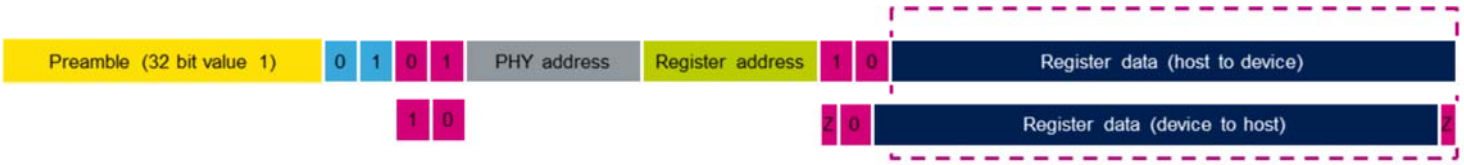
When host writes the data to slave:

- The host sends the 0b10 code instead. The turn-around code errors are reported by the TERF bit.
- The turn-around field will only be processed by the MDIOS device when the frame’s physical address matches.

Frame processing

Register data

- 16-bit register data
  - On write, the data is sent from host to slave.
  - On read, the data is sent from slave to host.



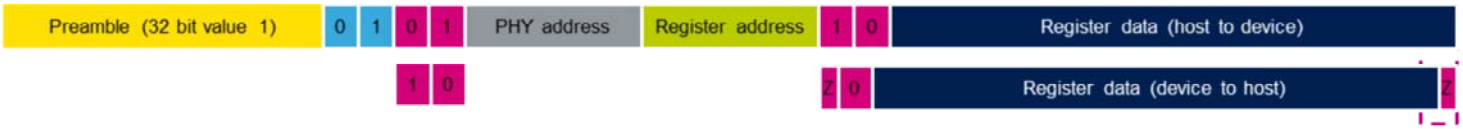
MDIO data received from the host is written in the addressed MDIOS register.
When enabled, the MDIOS generates an WRF(n) interrupt, that's also able to wake up the device from Stop mode.
The received data will only be processed by the MDIOS device when the write frame turn-around code is valid.

MDIO data requested by the host will be read from the addressed MDIOS register.
When enabled, the MDIOS will generate an RDF(n) interrupt that is able to wake up the device from Stop mode.

Frame processing

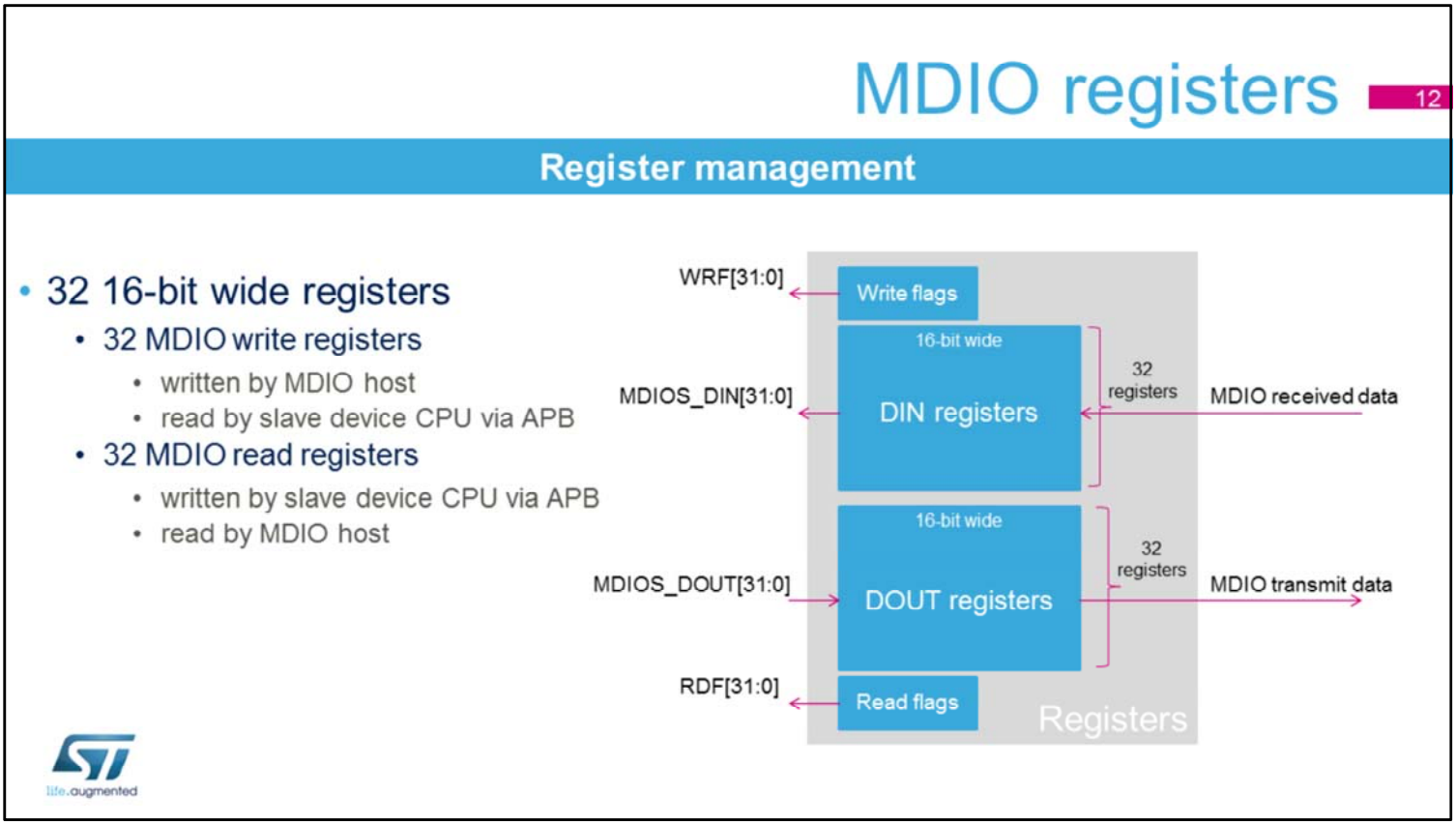
Trailing turn-around

- Trailing turn around field
  - On read, the host configures the MDIO line as output and the slave as input.



The trailing turn-around field is only present in read frames to hand over the MDIO line from the slave back to the host.

In detail, the trailing turn-around field is only ½ cycle long and is used by the slave to switch its MDIO pin to Input mode.



The MDIOS provides separate write and read registers.
The write registers are written by the MDIO host and read by the slave device CPU via the APB bus.
The read registers are written by the slave device CPU via APB and read by the MDIO host.

Each write and read register has an associated interrupt flag WRF[31:0] and RDF[31:0] able to generate an interrupt and wake up the slave device from Stop mode when the MDIO host accesses the register.

For the DOUT[n] register to reflect the DIN[n] data, the device CPU has to copy the data via the APB bus. The WRF[31:0] flags may be used to detect write register update by the MDIO host.
The device CPU has a time of (32-bit preamble + 4-bit start condition + 5-bit physical address + 5-bit register address) MDC clocks to copy the data before the MDIO host can read

the DOUT register again.

MDIOS interrupts

Interrupt event	Description
WRF[31:0]	MDIO register write event
RDF[31:0]	MDIO register read event
PERF	Preamble error detected
SERF	Start code error detected
TERF	Write frame turn around code error detected

Here is an overview of the MDIOS interrupt events.
Register accesses by the MDIO host are signaled by bits WRF[31:0] for write registers and bits RDF[31:0] for read registers.
Frame transfer errors are signaled by PERF, SERF and TERF interrupt events.

Low-power modes

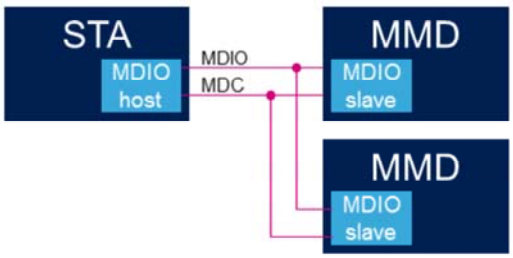
Mode	Description
Run	Active
Sleep	Active, peripheral interrupts cause the CPU to exit Sleep mode
Stop	Active, peripheral is able to wake up the device
Standby	Powered-down, the peripheral must be reinitialized after exiting domain and system Standby mode

life.augmented

Here is an overview of the peripheral status at specific low-power configuration modes. The device is not able to perform any communication in Standby mode. It is important to ensure that all transmissions are completed before the MDIOS controller is disabled or the system is switched down to Standby mode.

Application example

- The MDIO bus is used to transfer control and status information between the Station Management device (STA) and MDIO Manageable devices (MMD).
- Typical usage:
  - STA containing the MAC layer
  - MMD containing PHY layer



life.augmented

Here is an example of the MDIO interface used for communication between the Station Management device (STA) and MDIO Manageable devices (MMD).
Typically an MDIO bus is used between the Ethernet MAC and the physical layer (PHY) in parallel with the physical layer (PHY) bus and is used to detect cable (dis)connection, speed of the Ethernet link, usage of CSMA/CD or full-duplex links, auto-crossover, etc…

Related peripherals and software

16

- This is a list of peripherals related to the MDIOS controller. Please refer to these trainings for more information if needed:
  - Reset and clock control (RCC)
  - Interrupts (NVIC)
  - Extended interrupts and events controller (EXTI)
  - General-purpose inputs/outputs (GPIO)

Here is a list of peripherals related to the MDIO interface. Users should be familiar with all the relationships between these peripherals to correctly configure and use the MDIOS controller.