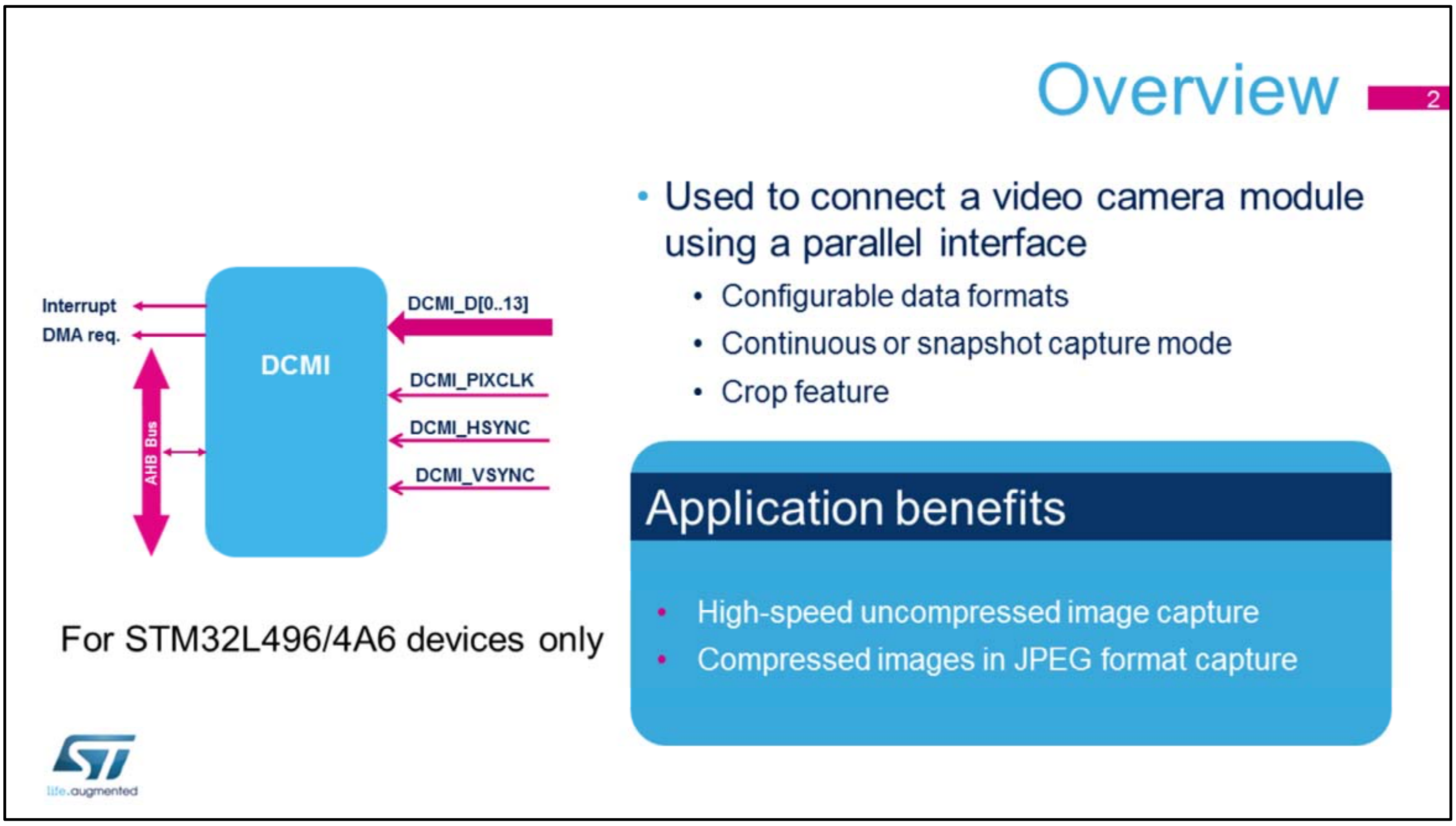
STM32L4 - DCMI

Digital Camera Interface

Revision 3.1

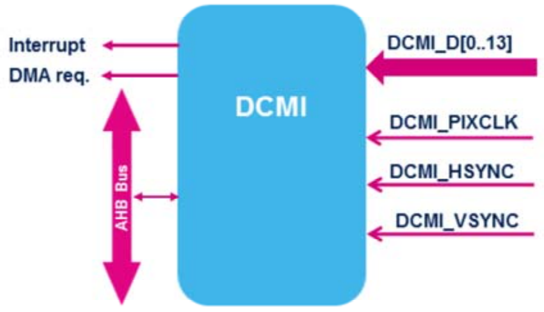
life.augmented

Hello, and welcome to this presentation of the STM32 digital camera interface controller. It covers all the features of this interface.



Overview

- Used to connect a video camera module using a parallel interface
  - Configurable data formats
  - Continuous or snapshot capture mode
  - Crop feature



For STM32L496/4A6 devices only

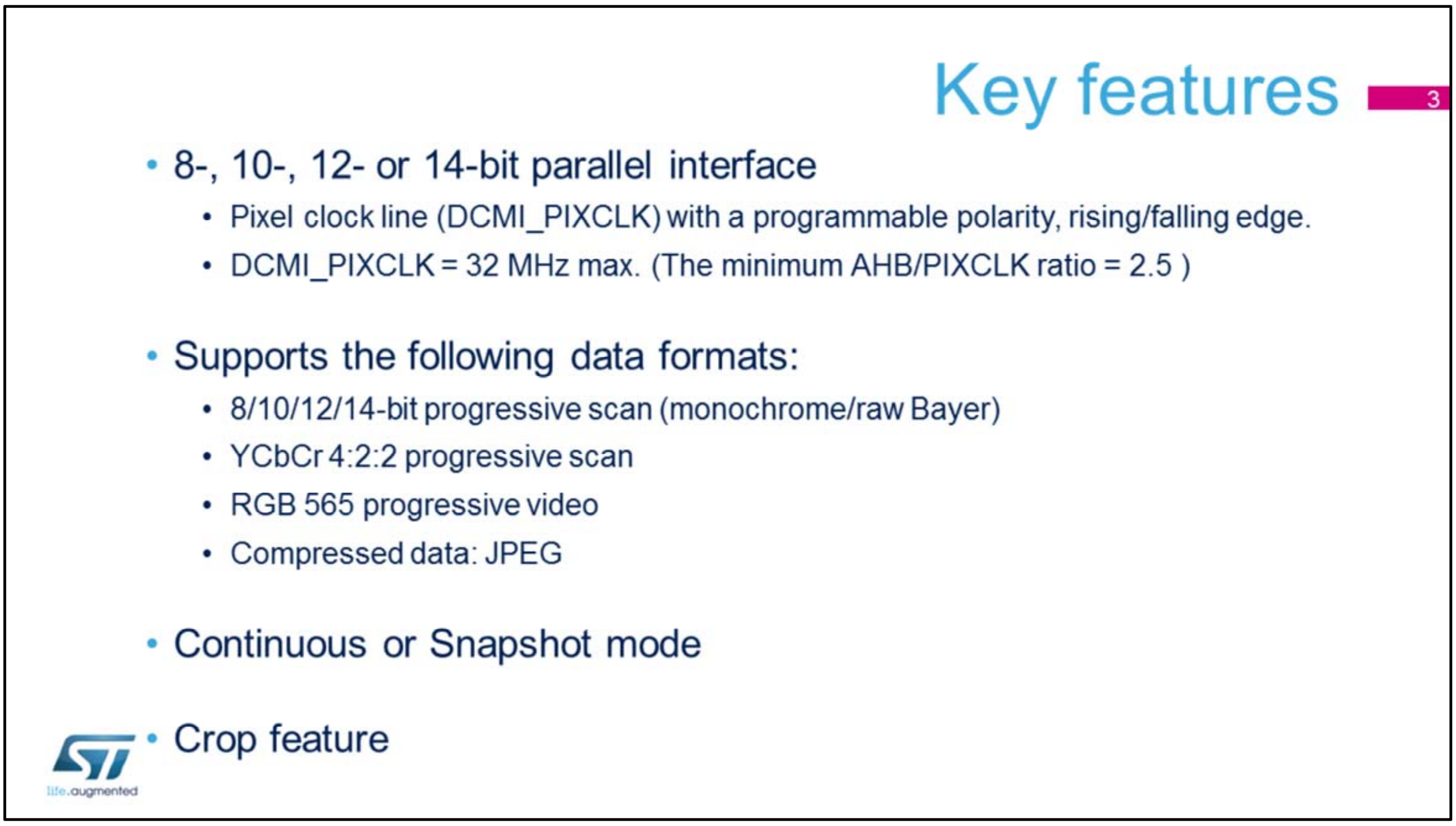
Application benefits

- High-speed uncompressed image capture
- Compressed images in JPEG format capture

DCMI stands for Digital Camera Interface. This interface is only available on STM32L496/4A6 devices.
The DCMI is used to connect a parallel camera module to the STM32.
The camera generates a parallel data flow together with a pixel clock signal (DCMI_PIXCLK) which allows the interface to capture the incoming data flow.
Two optional signals (HSYNC and VSYNC) may be used to synchronize the image frame between the camera and the STM32. The DCMI also supports embedded line/frame synchronization code in the data flow.

The DCMI is used to perform continuous grabbing. This process starts with an application request and continues until the CAPTURE bit is cleared. Alternatively, Snapshot mode is used to capture a single frame upon an application request.

With the crop feature, the camera interface can cut and store a rectangular portion of the received image.

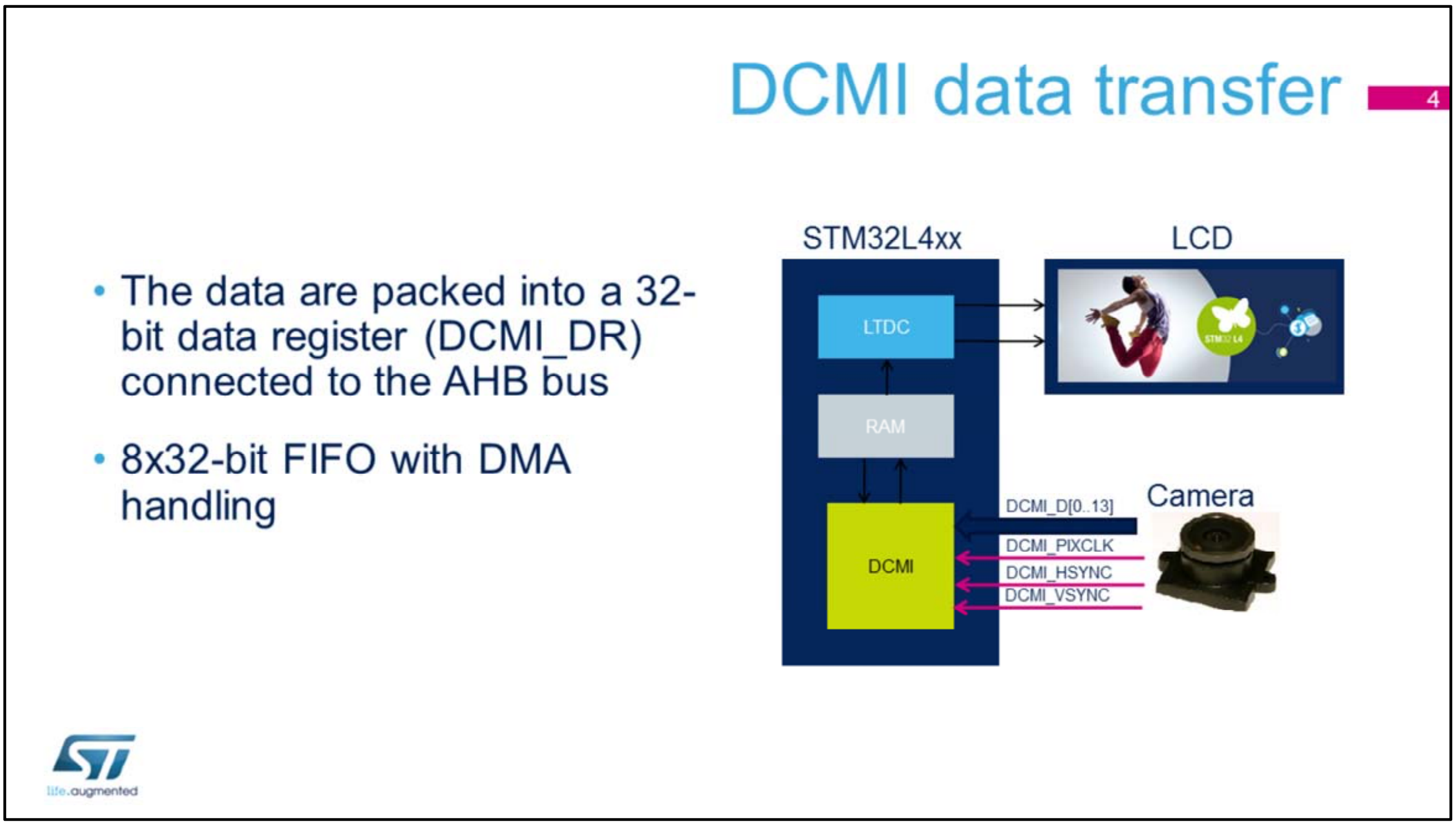


Key features

3

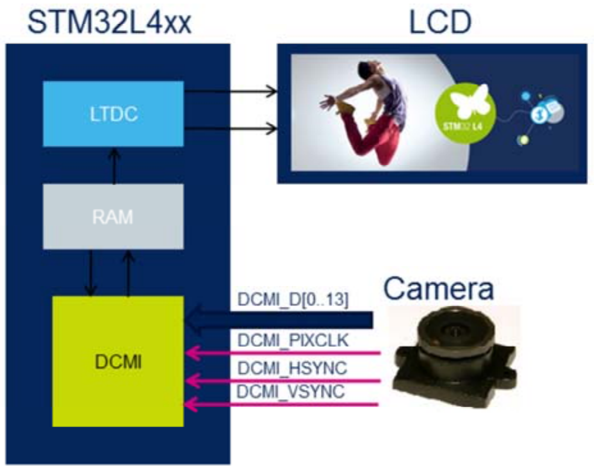
- 8-, 10-, 12- or 14-bit parallel interface
  - Pixel clock line (DCMI_PIXCLK) with a programmable polarity, rising/falling edge.
  - DCMI_PIXCLK = 32 MHz max. (The minimum AHB/PIXCLK ratio = 2.5)
- Supports the following data formats:
  - 8/10/12/14-bit progressive scan (monochrome/raw Bayer)
  - YCbCr 4:2:2 progressive scan
  - RGB 565 progressive video
  - Compressed data: JPEG
- Continuous or Snapshot mode
- Crop feature

The camera interface has a configurable parallel data interface with 8 to 14 data lines, together with a pixel clock line (DCMI_PIXCLK) with a programmable polarity, rising/falling edge configuration and a maximum DCMI_PIXCLK of 32 MHz.
The DCMI pixel clock and advanced high-performance bus (AHB) clock must respect the minimum AHB/PIXCLK ratio of 2.5.
The DCMI supports color or monochrome cameras using different data formats:
Uncoded parallel data, also known as progressive scan, which can be either monochrome or color (raw Bayer)
Luminance/color coded on 8 bits (4/2/2 progressive scan)
RGB 565, red-green-blue information coded on 16 bits
Some cameras also use this parallel interface to transmit compressed images in JPEG format.



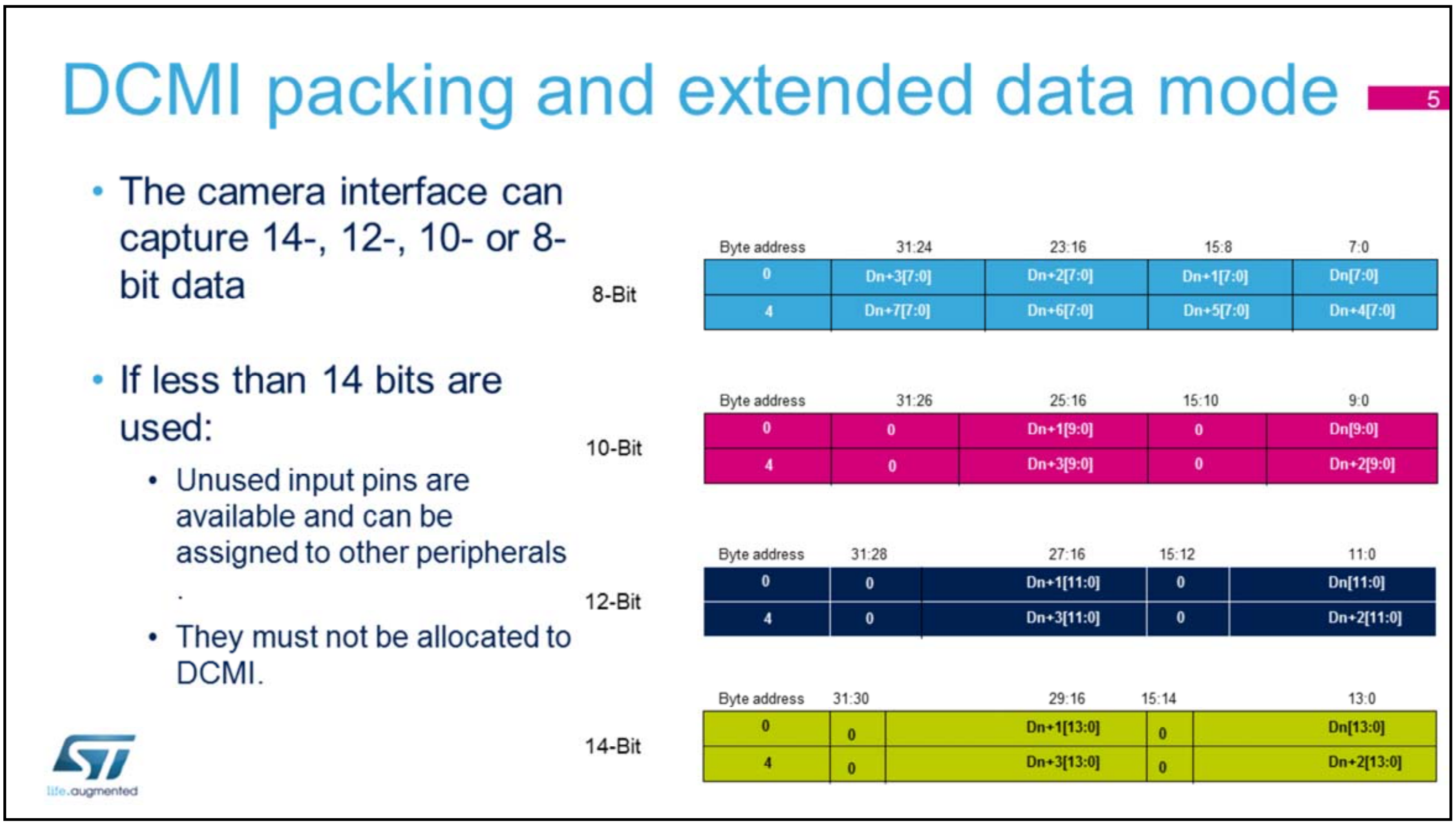
DCMI data transfer

- The data are packed into a 32-bit data register (DCMI_DR) connected to the AHB bus
- 8x32-bit FIFO with DMA handling



This is an example of a simple application used to transmit the camera image to the LCD display.

The standard way to use the camera interface is to store the received data in a frame buffer in RAM. The STM32 can then process this data or transmit it further through another interface (e.g. USB or Ethernet).
In order to limit the bus contention in the system and avoid missing data, despite the high data rate of this interface, the received data are packed in a FIFO buffer, as shown on the next slide.



DCMI packing and extended data mode

- The camera interface can capture 14-, 12-, 10- or 8-bit data
- If less than 14 bits are used:
  - Unused input pins are available and can be assigned to other peripherals
  .
  - They must not be allocated to DCMI.

8-Bit

Byte address	31:24	23:16	15:8	7:0
0	Dn+3[7:0]	Dn+2[7:0]	Dn+1[7:0]	Dn[7:0]
4	Dn+7[7:0]	Dn+6[7:0]	Dn+5[7:0]	Dn+4[7:0]

10-Bit

Byte address	31:26	25:16	15:10	9:0
0	0	Dn+1[9:0]	0	Dn[9:0]
4	0	Dn+3[9:0]	0	Dn+2[9:0]

12-Bit

Byte address	31:28	27:16	15:12	11:0
0	0	Dn+1[11:0]	0	Dn[11:0]
4	0	Dn+3[11:0]	0	Dn+2[11:0]

14-Bit

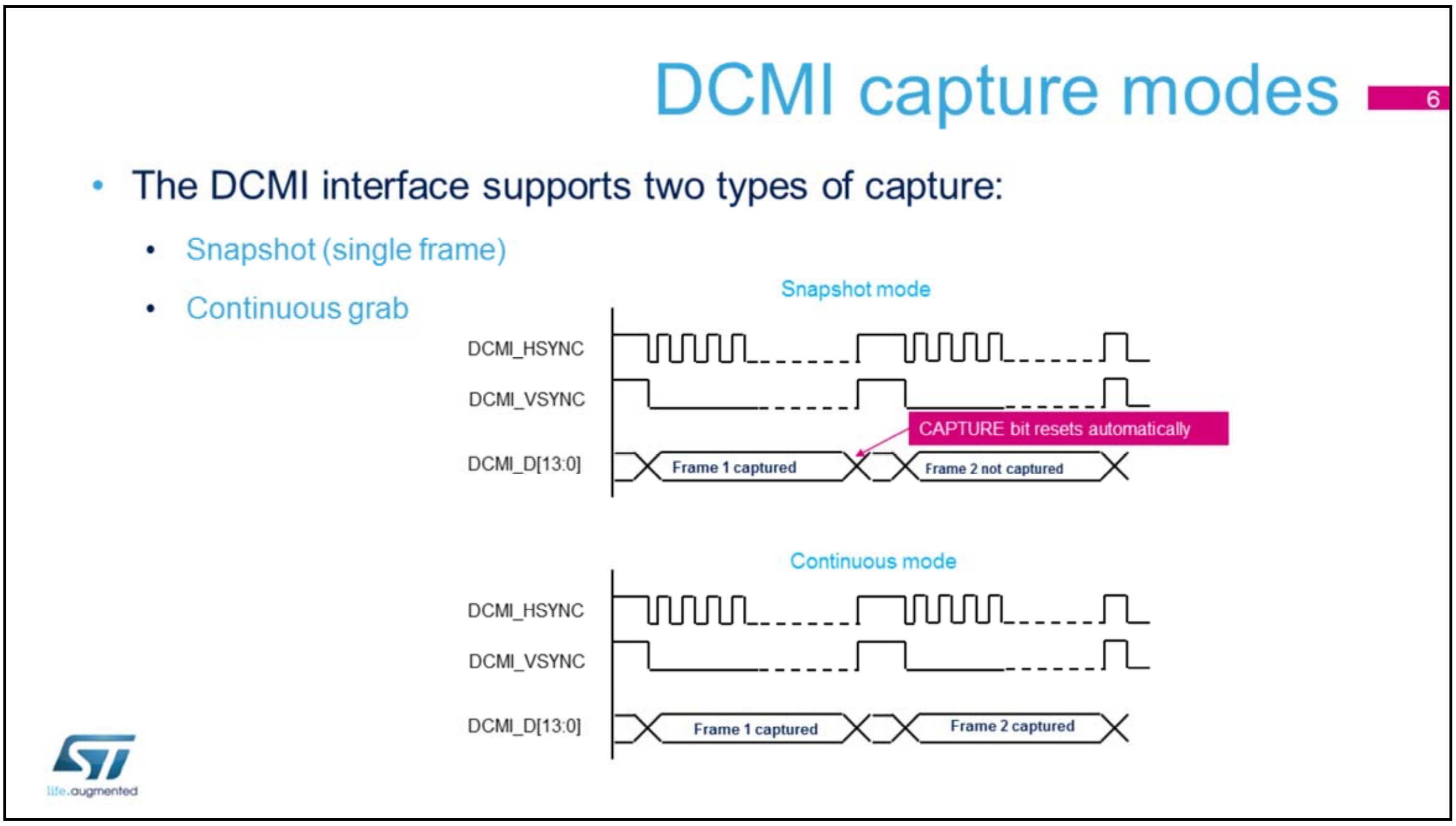
Byte address	31:30	29:16	15:14	13:0
0	0	Dn+1[13:0]	0	Dn[13:0]
4	0	Dn+3[13:0]	0	Dn+2[13:0]

life.augmented

Depending on the interface size (8, 10, 12 or 14 bits), 2 or 4 data items are stored in a single 32-bit word. Once a complete 32-bit word is available, it is transferred by DMA to the memory. This allows reduction of the bus bandwidth used by the DCMI, even for high-speed cases.

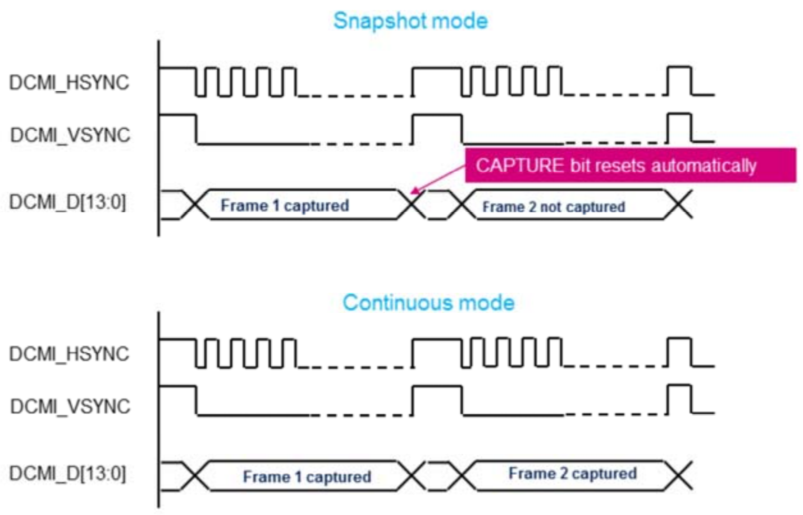
The DCMI has also a hardware feature allowing to select 1 byte out of 2 or 1 byte out of 4. This feature is used to convert of a color image to black and white and/or reduce the image size. In this latter case, in order to keep the form factor of the image, the DCMI may only store every other line, reducing the vertical resolution by a factor of two.

An 8-level FIFO is used in order to accommodate for any DMA response latency, without losing data.

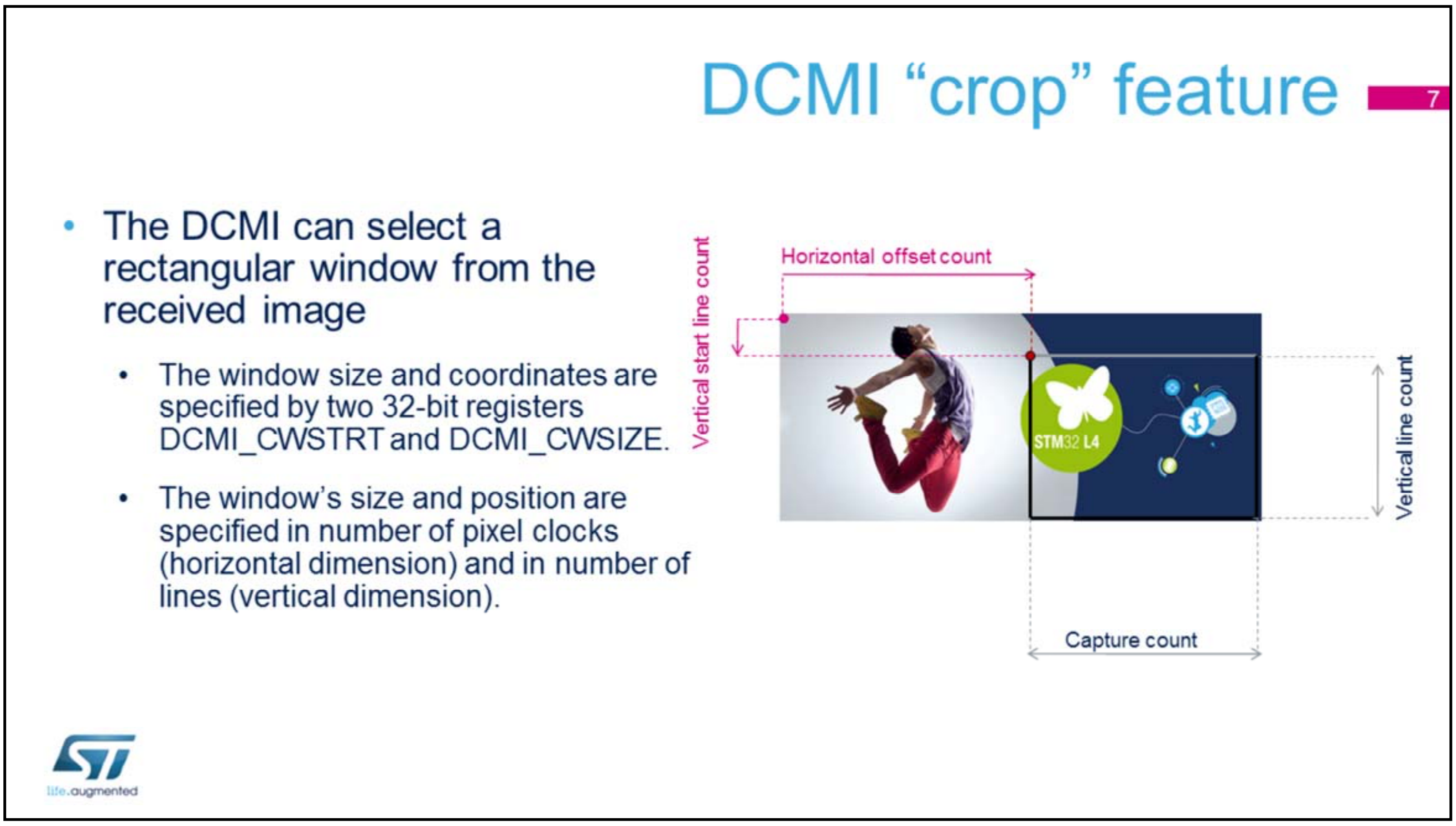


DCMI capture modes

- The DCMI interface supports two types of capture:
  - Snapshot (single frame)
  - Continuous grab

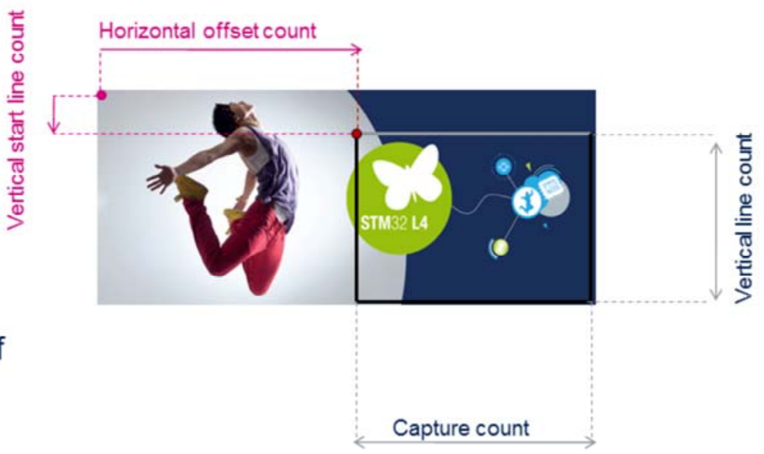


The camera interface allows to capture a single frame (synchronized on after a software request) or to continuously receive the video flow.
In Capture mode, the capture is requested by setting the Capture bit in software and starts with the beginning of the next incoming frame and the DCMI clears the Capture bit when the single frame has been received.

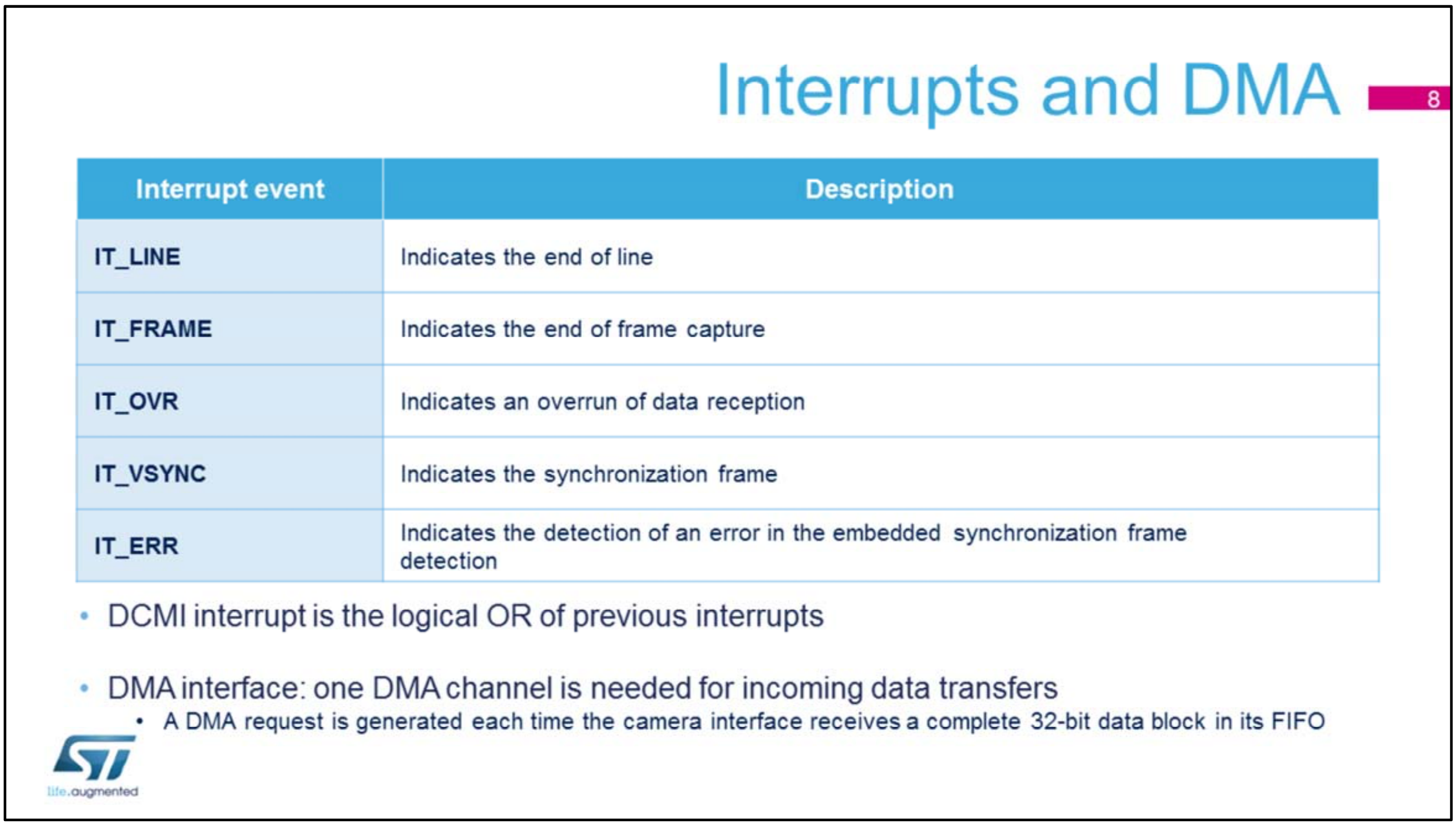


DCMI "crop" feature

- The DCMI can select a rectangular window from the received image
  - The window size and coordinates are specified by two 32-bit registers DCMI_CWSTRT and DCMI_CWSIZE.
  - The window's size and position are specified in number of pixel clocks (horizontal dimension) and in number of lines (vertical dimension).



Cropping is another way to reduce the image size, in addition to reducing the pixel resolution as mentioned previously.
This option is valid for both single frame capture and in Continuous mode, but it does not support JPEG format.



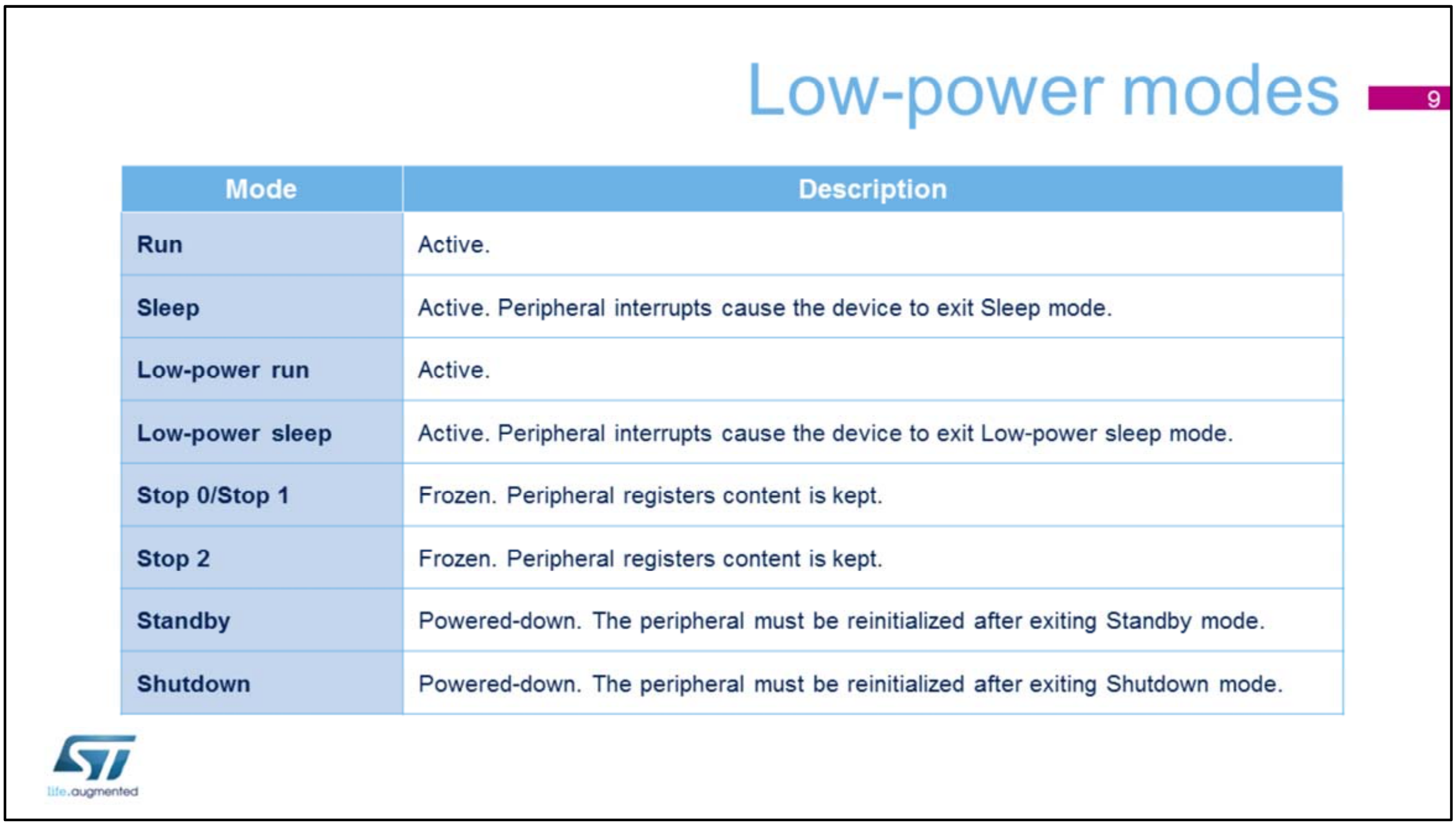
Interrupts and DMA

Interrupt event	Description
IT_LINE	Indicates the end of line
IT_FRAME	Indicates the end of frame capture
IT_OVR	Indicates an overrun of data reception
IT_VSYNC	Indicates the synchronization frame
IT_ERR	Indicates the detection of an error in the embedded synchronization frame detection

- DCMI interrupt is the logical OR of previous interrupts
- DMA interface: one DMA channel is needed for incoming data transfers
  - A DMA request is generated each time the camera interface receives a complete 32-bit data block in its FIFO

Five interrupts can be generated. All interrupts are maskable by software. The global interrupt is the OR of all single interrupts.

The DMA interface is active when Capture mode is enabled. A DMA request is generated each time the camera interface receives a complete 32-bit data block in its FIFO. For DMA channels available to DCMI, please refer to the DMA section in the STM32L4 reference manual.

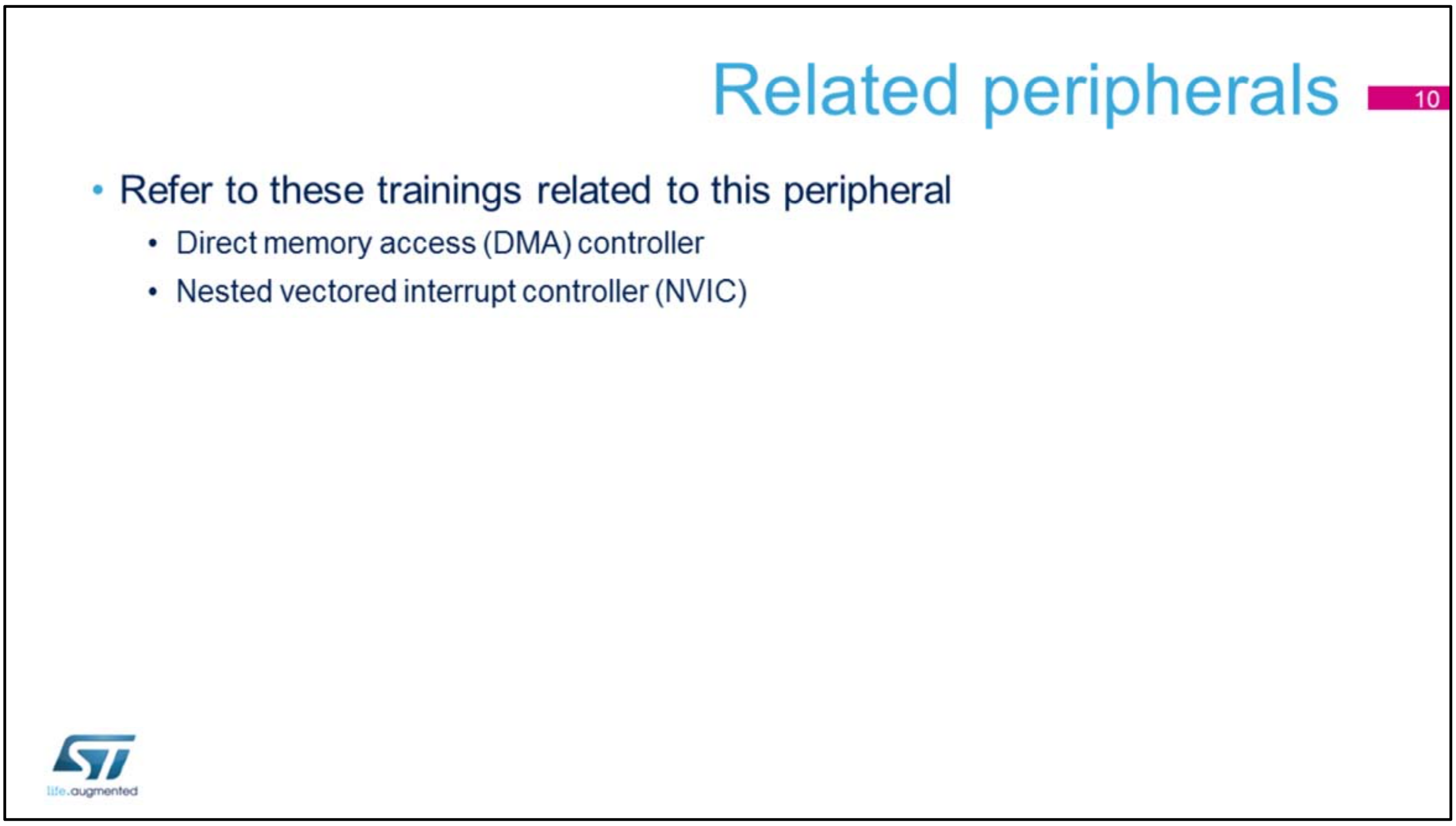


Low-power modes

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 Shutdown mode.

life.augmented

The DCMI module is active in Run, Sleep, Low-power run and Low-power sleep modes. In Stop 0, Stop 1 or Stop 2 modes, DCMI operations are not possible but the contents of its registers are kept. In Standby or Shutdown modes, the DCMI is powered-down and must be reinitialized when returning to a higher power state.



Related peripherals

10

- Refer to these trainings related to this peripheral
  - Direct memory access (DMA) controller
  - Nested vectored interrupt controller (NVIC)

life.augmented

This is a list of peripherals related to the DCMI module. Refer to DMA and NVIC trainings for more information about the DCMI channel and interrupt configuration and to GPIO chapter for setting up the alternate function pins used by the DCMI.

Examples of configuration and use of camera interface are available in the STM32CubeL4 package available at www.st.com .

The camera additionally need to be set up through the I2C bus before the images can be transferred via DCMI. An example of two different camera setups is available in the Board Support Package driver files.
Other camera types need different setups that users have to extract from their documentation.