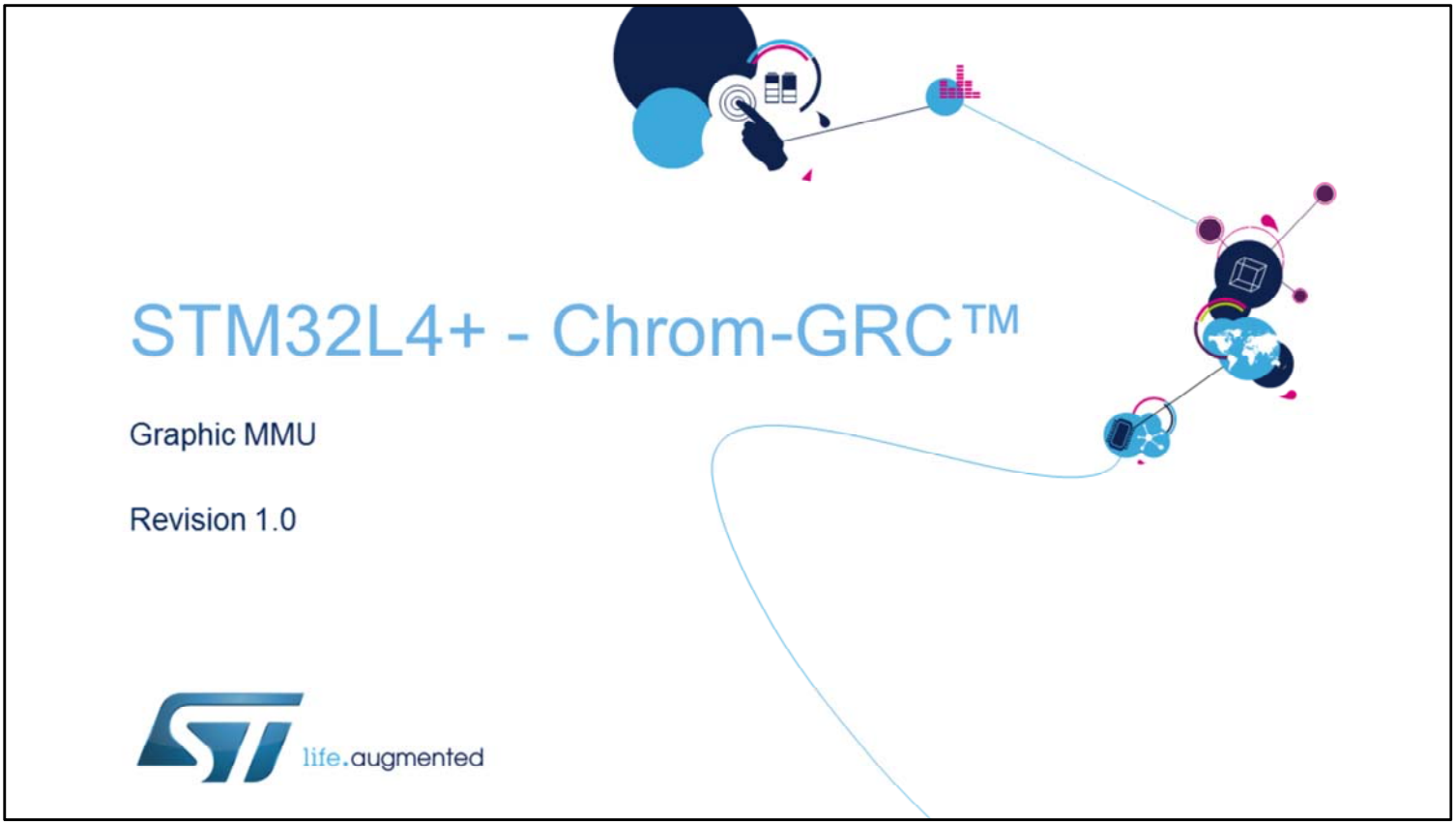
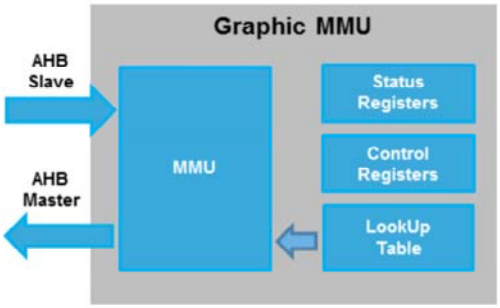


Hello, and welcome to this presentation of the STM32 Chrom-GRC™.
This chapter presents the features of the Memory Management Unit used in graphical-oriented applications.

Overview



- The **Graphic MMU** is a graphical oriented Memory Management Unit aimed to optimize memory usage according to the display shape (ex. round display).

Application benefits

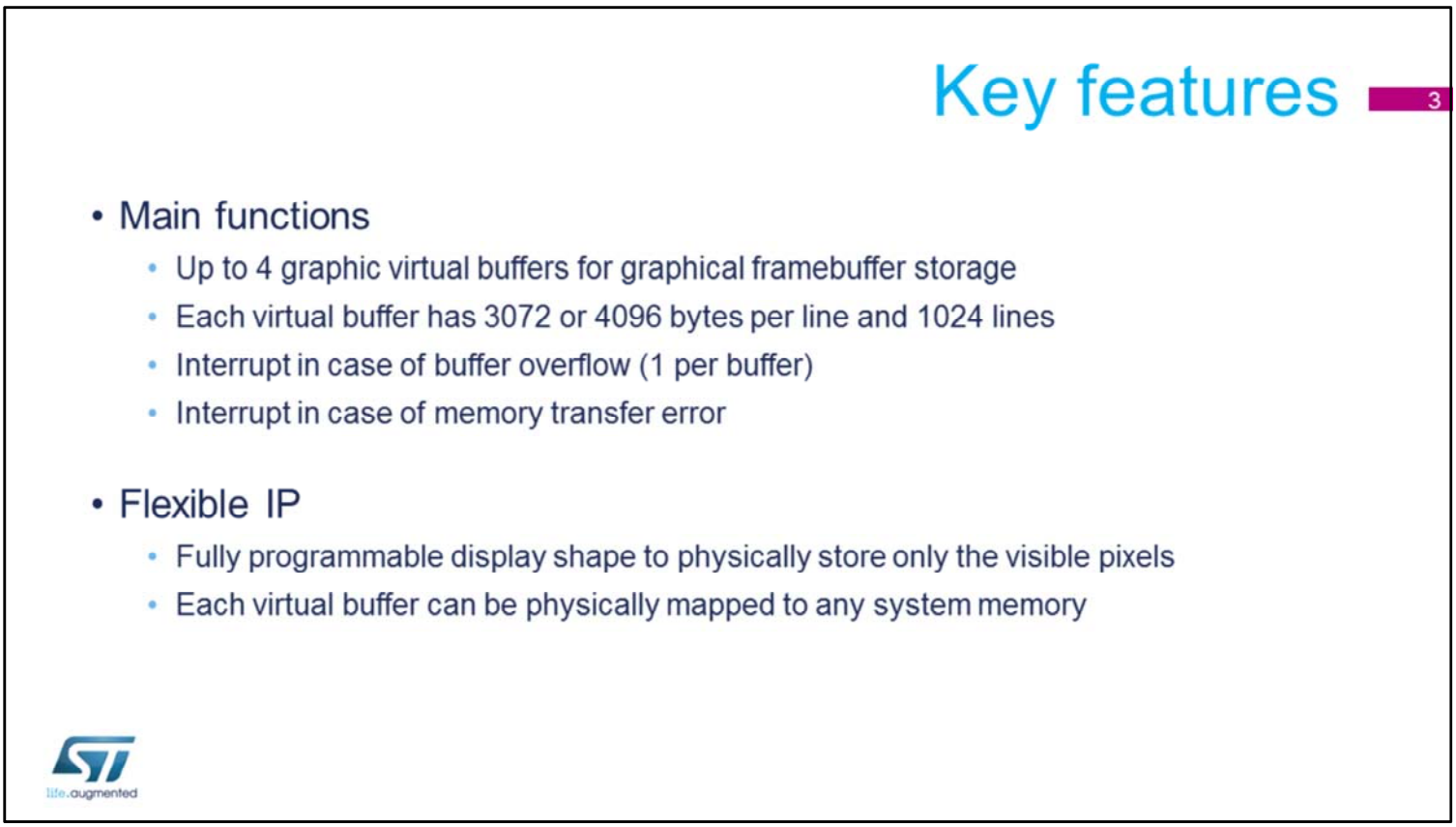
- Lower memory usage according to display shape
- Fully configurable display shape
- Transparent integration
- Works with any memory of the system

life.augmented

The Chrom-GRC™ is a graphic oriented memory management unit to optimize the memory usage according to the display shape.

The Chrom-GRC™ lowers the memory usage by storing only the visible pixels in the memory. It is fully configurable according to the display shape and is totally transparent in the system.

The Chrom-GRC™ works with any memories and any masters of the STM32.



The Chrom-GRC™ is seen by the masters of the STM32 system as four virtual buffers to store the graphical framebuffer.

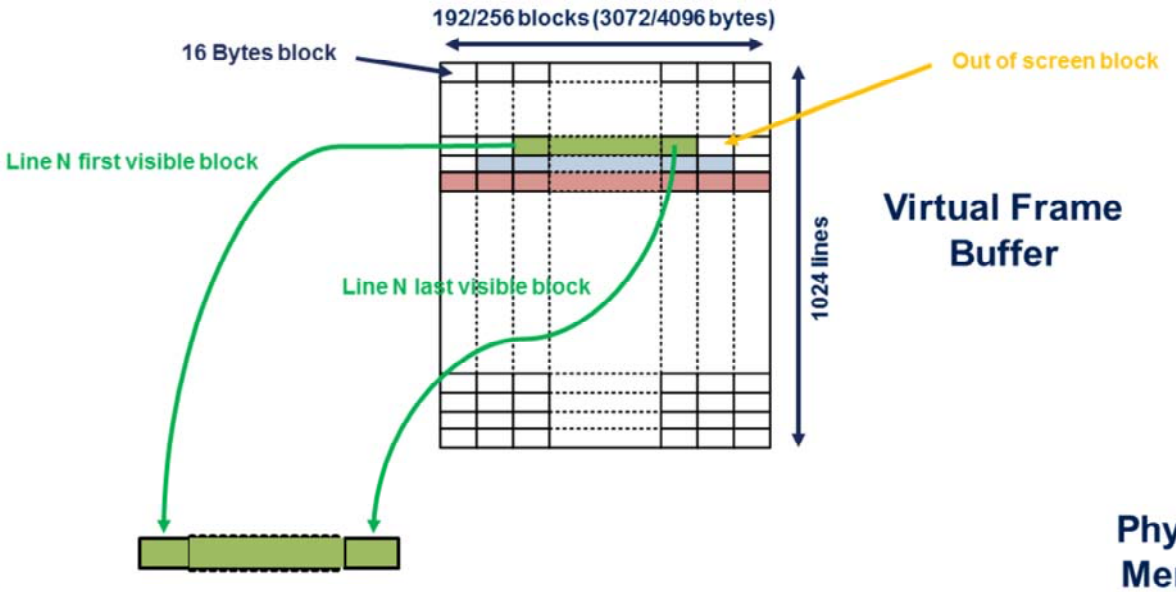
Each virtual buffer consists of 1024 lines with 3072 or 4096 bytes per line. The size of each line is configurable by software.

In case of accesses exceeding the buffer space, a buffer overflow interrupt can be generated, and in case of wrong physical memory access, a memory transfer error is generated.

The display shape is described in a lookup table to define which bytes of the virtual buffer have to be stored (or not) in a physical memory.

Thanks to this very flexible architecture, only the visible pixels are effectively stored, and each virtual buffer can be physically mapped to any memory of the system.

Functional overview



Each virtual buffer is seen as a continuous memory space of 3072 or 4096 bytes per 1024 lines.

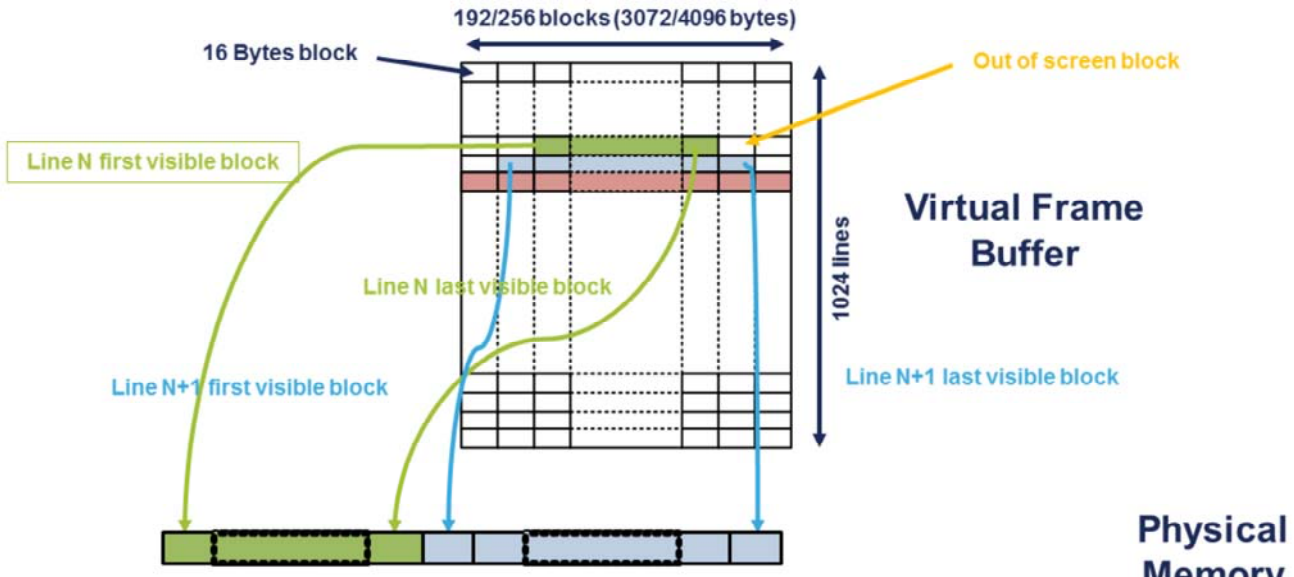
Each line is composed of 192 or 256 16-bytes blocks.

In this example, out of screen blocks (shown here in white) won't be mapped into the physical memory.

For each line, the first visible block and the last visible block are stored in the lookup table.

Only the visible blocks of each line will be stored in the physical memory.

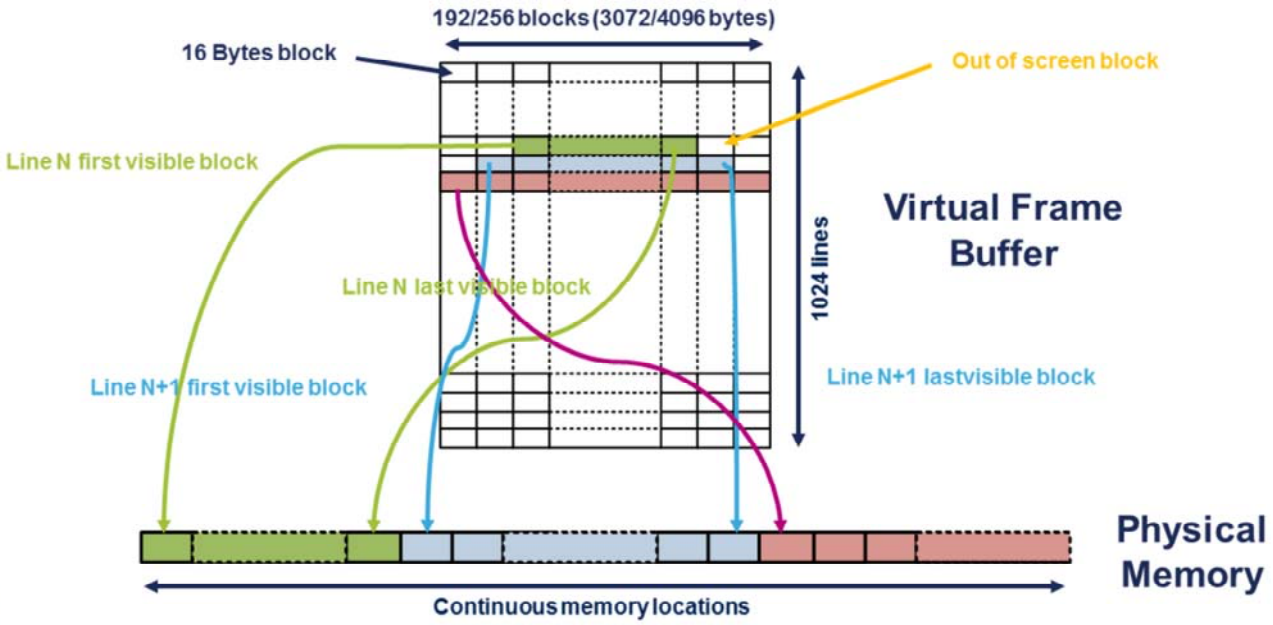
Functional overview



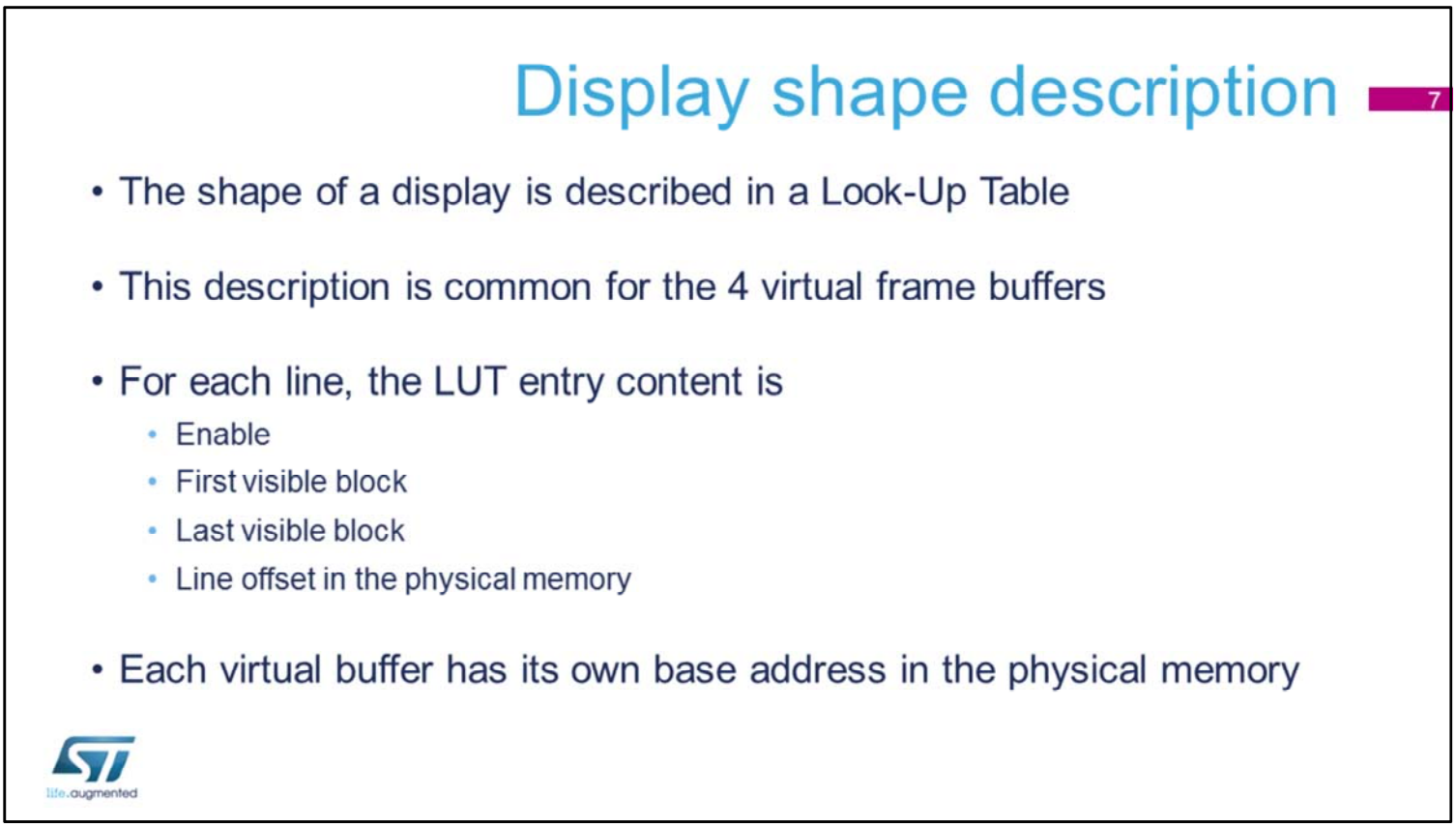
The operation is repeated for each line, indicating the first block and the last block.

All the first and last block numbers are stored in a lookup table.

Functional overview



And all the visible blocks are stored in the physical memory in a continuous way saving a huge amount of space for non-rectangular display shapes.



All the line parameters are stored in an internal lookup table.

The description is the same for all 4 virtual frame buffers.

For each line, the following parameters are stored:

- The enable bit of the line
- The number of the first visible block
- The number of the last visible block
- And the line offset in the physical memory.

The line offset is the offset between the physical buffer base address programmed by the user, and the address where the block 0 of the line has to be stored.

Each virtual buffer has its own base address in the physical memory.

Operating modes

8

- Two operating modes
  - 192 Block Mode: each frame buffer line is composed of 192 blocks or 16 bytes
  - 256 Block Mode: each frame buffer line is composed of 256 blocks or 16 bytes
- Line size in pixel of the framebuffers

Mode	16bpp	24bpp	32bpp
192 BM	1536 pixels/line	1024 pixels/line	768 pixels/line
256 BM	2048 pixels/line	1365,3 pixels/line	1024 pixels/line

  - 192 Block Mode allows to have an integer number of pixels in 24bpp mode

life.augmented

The Chrom-GRC™ runs with two operating modes:

- 192 blocks per line mode
- 256 blocks per line mode

The 192 blocks per line mode is useful as we can have an integer number of pixels per line when the frame buffer color mode is in 24 bits per pixel.

Gain

9

- The gain in size on the graphical frame buffer on round display is ~20% depending on the number of bit per pixels
  - Example for the 390x390 round display of the STM32L4+ kit

Mode	Square size	Chrom-GRC™ optimized size	Size reduction
16bpp	297,1 KBytes	238,3 KBytes	**-19,8 %**
24bpp	445,6 KBytes	355,5 KBytes	**-20,2 %**
32bpp	594,1 KBytes	471,0 KBytes	**-20,7 %**

life.augmented

Thanks to the smart mapping of visible pixel in the memory, the Chrom-GRC™ reduces the RAM usage significantly for the graphical frame buffer.

For a round display, the gain is above 20%. There may be slight variation depending on the frame buffer color mode.

Interrupts

Interrupt event	Description
AHB Master Error	An error occurred on the AHB master port
Buffer 0 Overflow	An overflow occurred during Buffer 0 address calculation
Buffer 1 Overflow	An overflow occurred during Buffer 1 address calculation
Buffer 2 Overflow	An overflow occurred during Buffer 2 address calculation
Buffer 3 Overflow	An overflow occurred during Buffer 3 address calculation

The Chrom-GRC™ manages 5 interrupt sources:

- AHB Master error when an error occurred during an AHB transaction to the physical memory
- 4 Buffer Overflows, one per buffer, when an overflow occurs during the offset calculation of a buffer.

Low-power modes

Mode	Description
Run	Active.
Sleep	Active. Peripheral interrupts cause the device to exit Sleep mode.
Stop	Frozen. Peripheral registers content is kept.
Standby	Powered-down. The peripheral must be reinitialized after exiting Standby mode.

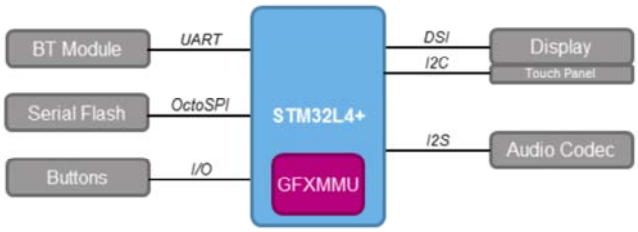
The Chrom-GRC™ is active in Run and Sleep modes. A Chrom-GRC™ interrupt can cause the device to exit.

In Stop mode, the Chrom-GRC™ is frozen and its registers' content are kept.

In Standby mode, the Chrom-GRC™ is powered-down and it must be reinitialized afterwards.

Application examples

- Embedded applications with connectivity and graphics:



- The Graphic MMU optimizes internal RAM usage for graphic applications
  - No need for external SRAM/SDRAM
  - Faster graphical operations on internal RAM (0-WS)

life.augmented

Wearable applications require low-power management functions together with a high-quality user interface. In this context, the Chrom-GRC™ optimizes the internal RAM usage, and it's no longer necessary to add an external component for SRAM/SDRAM. The graphical content creation is much faster thanks to the internal 0 Wait State RAM.

Related peripherals

- Refer to these peripherals trainings linked to this peripheral:
  - RCC (GFXMMU clock control, GFXMMU enable/reset)
  - Interrupts (GFXMMU interrupt mapping)

You can refer to the peripheral training slides related to RCC and interrupts.