Hello, and welcome to this presentation of the STM32 Chrom-ART Accelerator™. It covers the features of this adaptive real-time accelerator block, which is widely used for graphic computing in the microcontroller.
The Chrom-ART™ accelerator offers true hardware acceleration for graphical operations.
The Chrom-ART™ accelerator is built around a 2D DMA engine for fast data copy with specific functions to support pixel format conversion as well as blending operations between two planes. It also provides specific modes for managing anti-aliased fonts.
The Chrom-ART™ accelerator will offload the CPU for most of the graphical operations with a one pixel per cycle throughput, integrated pixel format conversion and blending.
The Chrom-ART™ accelerator is fully integrated in graphical stacks making its software integration transparent to the user.

• Provides hardware acceleration for graphical operations
  • Graphics-oriented 2D DMA
  • Planes blending & pixel format conversion
  • Specific modes for anti-aliased fonts

Application benefits
• Offloads CPU for graphical operations
• One pixel per cycle calculation
• Integrated pixel format converter & blender
• Simple integration through graphical stack
The Chrom-ART™ accelerator has four operating modes.
• Register-to-memory for rectangle filling operations,
• Memory-to-memory for 2D memory copy operations,
• Memory-to-memory with pixel format conversion for bitmap drawing with format conversion,
• Memory-to-memory with pixel format conversion and blending for bitmap or text drawing with transparency.

The user can program independently all the parameters for the source and the destination:
- The address of the layer including its size and position
- The color format
- The way transparency is managed.
Register-to-memory mode is used to fill a part or whole destination image with a specific color. The color value is set in a register of the output PFC.
Memory-to-memory mode is used to copy a part or whole source image into a part or whole destination image without changing the color format.
Memory-to-memory mode with pixel format conversion is used to do the same type of copy as Memory-to-memory mode but with a pixel format conversion. It can copy any input format to any output format. For example, an ARGB565 image into an ARGB888 image without having to use the CPU.
Memory-to-memory mode with pixel format conversion and blending is used to blend a part or whole source image with a part or whole destination image with a different pixel format. This is widely used to draw bitmap icons having transparency or fonts.
For each foreground and background layer, the format can be programmed independently. Direct mode fetches the RGB or ARGB content directly from the memory. Indirect mode uses an intermediate color look-up table to determine the color to be used during the copy or blending operation. All the input color modes are transformed internally into ARGB8888 format to perform the blending operation.
The YCbCr mode enables a specific color space converter hardware to convert YCbCr data into RGB data. This block is available on the foreground plane. This color space converter is used to copy or blend uncompressed data from the JPEG decoder into the framebuffer. It allows to convert directly YCbCr data arranged in 8x8 pixel blocks into a linear RGB format. All the standard YCbCr chroma sub-sampling formats are supported from 4:4:4 down to 4:2:0. This new feature offloads the CPU from this costly operation and makes it 10 times faster.
Specific modes can be used to efficiently manage texts and fonts. Only the transparency value is stored in memory for rendering anti-aliased fonts. The color is added during the pixel format conversion process and can be programmed by the user. These modes are very efficient for storing high-quality bitmap fonts.
The output pixel format converter generates the color for the destination independently from the source. There is no indirect mode in output as this would imply to calculate a color look-up table (CLUT). Nevertheless, memory-to-memory operations without Pixel Format Conversion (PFC) can copy data independently of their formats.
The fully hardware blender allows to blend a foreground image and a background image with transparency. This can be used to draw bitmap images of any shape with a perfect rendering. 1 pixel is generated per cycle making this complex operation much more efficient than if it was done by the CPU. The resulting pixel can be coded independently from the source thanks to the output pixel format converter.
The output configuration defines the working area for the Chrom-ART™ operation. The address and the line offset parameters are used to select which sub-area of the output is concerned.
The background and foreground layers have their own configuration for address, line offset and color format. This defines which area of the foreground and background layers are targeted by the Chrom-ART™ operations.
The Chrom-ART™ accelerator has 6 interrupt sources to signal:

- Configuration errors
- CLUT transfer complete
- CLUT access error
- Watermark reached during a transfer
- Transfer complete
- Transfer error

No DMA trigger is used as the Chrom-ART™ accelerator embeds its own DMA.
The Chrom-ART™ accelerator is active in Run and Sleep modes. A Chrom-ART™ interrupt can cause the device to exit Sleep mode. In Stop mode, the Chrom-ART™ accelerator is frozen and its registers content is kept. In Standby mode, the Chrom-ART™ accelerator is powered-down and it must be reinitialized afterwards.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run</td>
<td>Active.</td>
</tr>
<tr>
<td>Sleep</td>
<td>Active. Peripheral interrupts cause the device to exit Sleep mode.</td>
</tr>
<tr>
<td>Stop</td>
<td>Frozen. Peripheral registers content is kept.</td>
</tr>
<tr>
<td>Standby</td>
<td>Powered-down. The peripheral must be reinitialized after exiting Standby mode.</td>
</tr>
</tbody>
</table>
The Chrom-ART™ accelerator is widely used in any graphical application to compute the frame buffer without any CPU load and with a very efficient throughput. It can compose the whole scene with transparency and facilitate the management of animations. Text rendering is also accelerated, making it easy and efficient to manage anti-aliased fonts.
You can refer to the trainings related to the RCC and interrupts for additional information.

- Refer to these trainings related to this peripheral:
  - RCC (Chrom-ART™ clock control and Chrom-ART™ enable/reset)
  - Interrupts (Chrom-ART™ interrupt mapping)
The Chrom-ART™ accelerator is implemented in the STM32F4 and STM32F7 series up to 216MHz with a 32-bit wide bus and up to 200MHz on the H7 series with a 64-bit wide bus.

<table>
<thead>
<tr>
<th>Chrom-ART™ features</th>
<th>STM32F0</th>
<th>STM32F1</th>
<th>STM32F2</th>
<th>STM32F3</th>
<th>STM32F4</th>
<th>STM32F7</th>
<th>STM32H7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of instances</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Maximum speed</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>180MHz</td>
<td>216MHz</td>
<td>200MHz(*)</td>
</tr>
</tbody>
</table>

(*) 64-bit wide data bus