UNIVERSITEIT
iYUNIVESITHI
STELLENBOSCH
UNIVERSITY

100
1918 · 2018

forward together · saam vorentoe · masiye phambili

Computer Systems / Rekenaarstelsels 245

Lecture 26

Displays/
Vertooneenhede

Dr Rensu Theart & Dr Lourens Visagie

LEDs
Lig diodes

Simple low-power LED

- So far we've used low-power LEDs driven directly from microcontroller for "display"

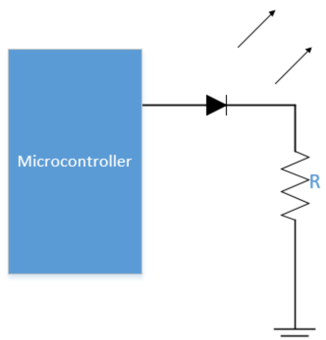


Table 12. Current characteristics

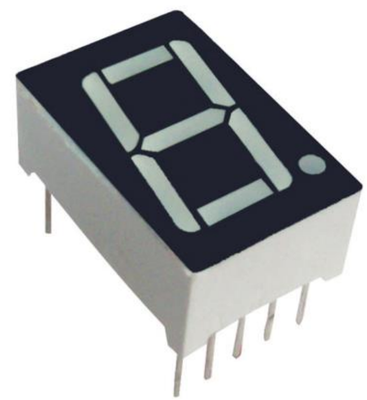
Symbol	Ratings	Max.	Unit
ΣI_{VDD}	Total current into sum of all V_{DD_x} power lines (source)(1)	160	mA
ΣI_{VSS}	Total current out of sum of all V_{SS_x} ground lines (sink)(1)	-160	
I_{VDD}	Maximum current into each V_{DD_x} power line (source)(1)	100	
I_{VSS}	Maximum current out of each V_{SS_x} ground line (sink)(1)	-100	
I_{IO}	Output current sunk by any I/O and control pin	25	
	Output current sourced by any I/O and control pin	-25	
ΣI_{IO}	Total output current sunk by sum of all I/O and control pins (2)	120	
	Total output current sourced by sum of all I/Os and control pins(2)	-120	
$I_{INJ(PIN)}$ (3)	Injected current on FT and TC pins (4)	–5/+0	
	Injected current on NRST and B pins (4)		
$\Sigma I_{INJ(PIN)}$	Total injected current (sum of all I/O and control pins)(5)	±25	

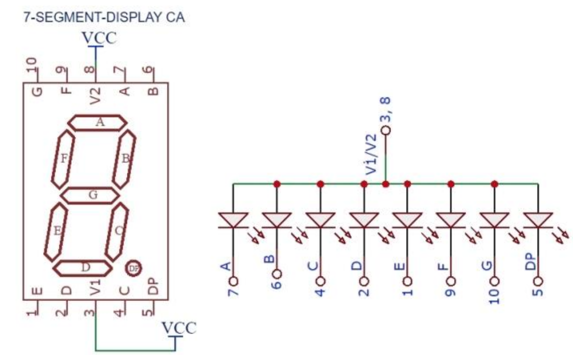
- GPIO pin, configured for output (Push-pull)
- Resistor needed to limit current supplied to LED (and current supplied by MCU pin)
- Single colour only

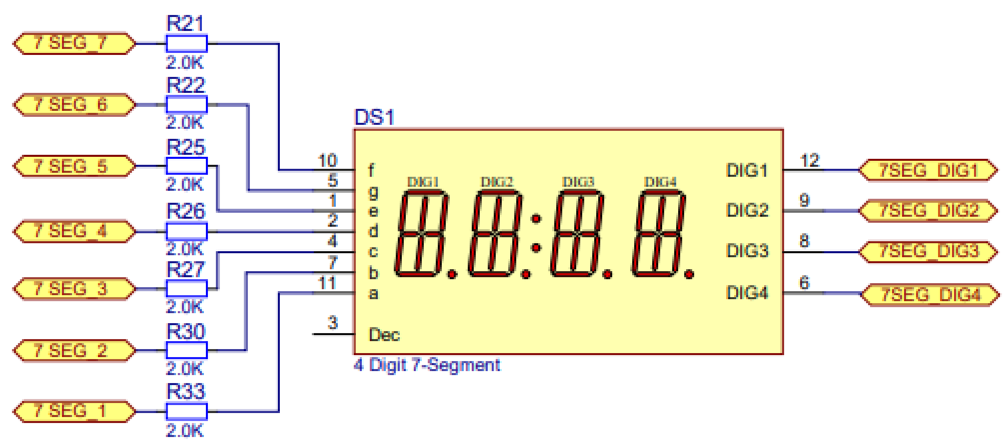
7-Segment LED
7-Segment vertooneenheid

7-segment LED

- Same principle as low-power LED, but diodes are arranged so that you can display a character or digit by selectively switching segments
  - Need 7x GPIO signals, one for each segment
- There are such displays with multiple digits (think about oven timer display)
  - Uses binary selection signals to reduce the number of GPIOs– selects each 7-segment display in turn
  - MCU implementation: 7x GPIO output signals for each segment, plus one enable GPIO output signal for each digit enable
  - In code, update each digit at a time (set relevant digit's enable signal)



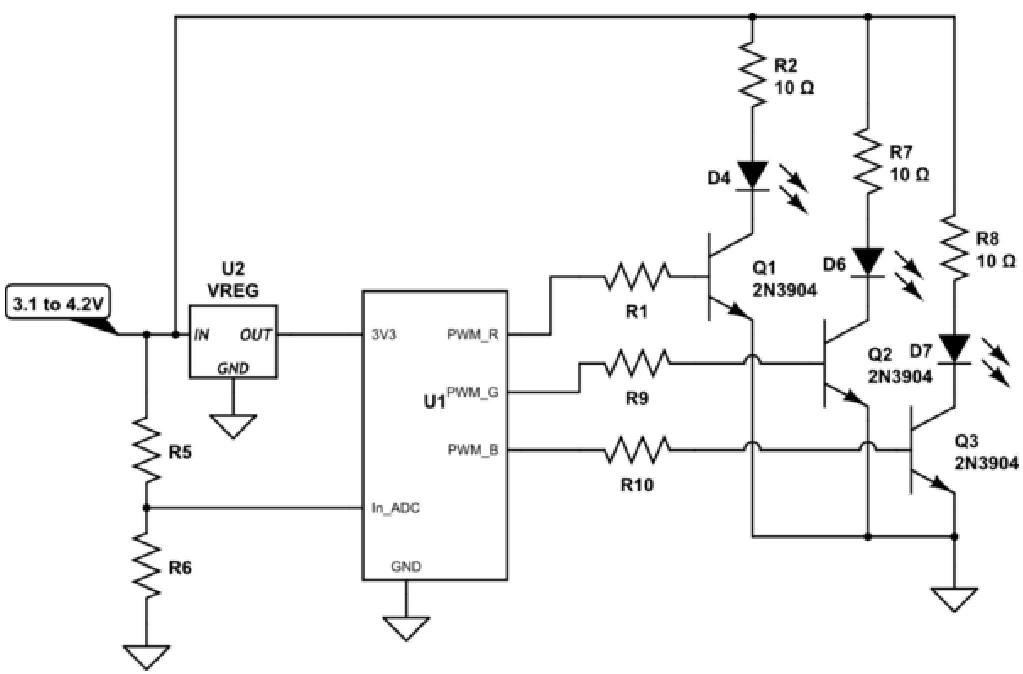




High-power RGB LED
Hoë-stroom rooi, groen en blou LED

High-power RGB LED

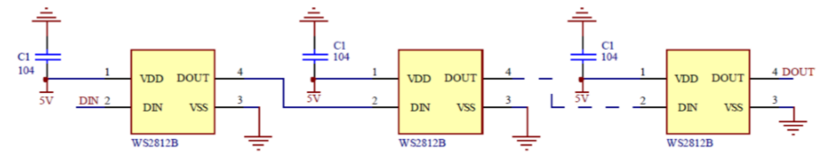
- Three LEDs in one package (red, green and blue)
- By varying the brightness of each, you can create arbitrary colours
- Microcontroller still outputs TTL digital signal
- Use PWM to control LED brightness (PWM period must be fast enough)
- Additional circuitry (transistors) is needed to provide enough current to LEDs

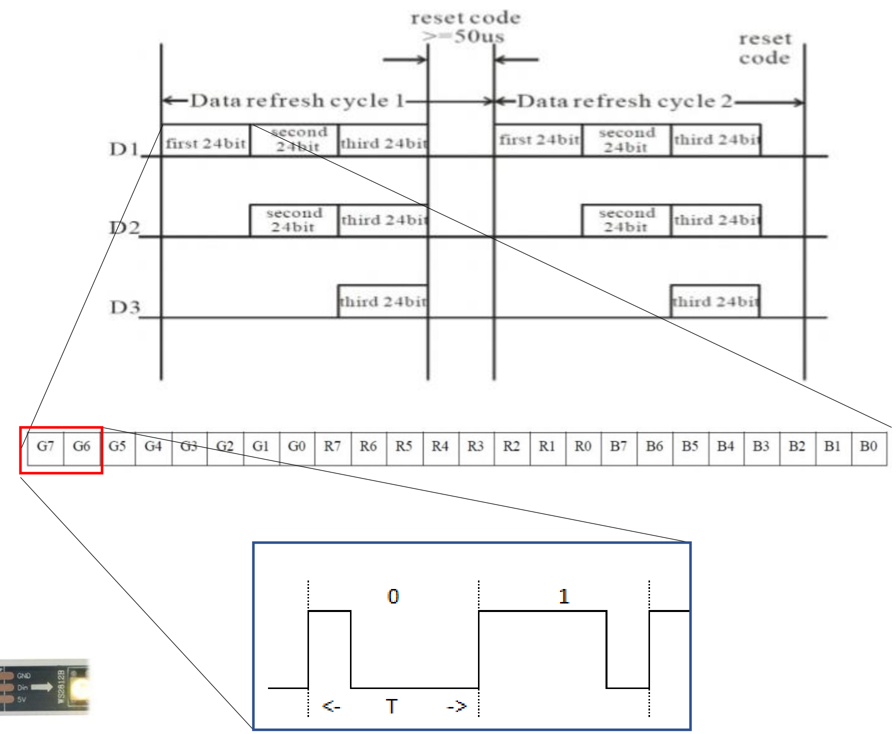


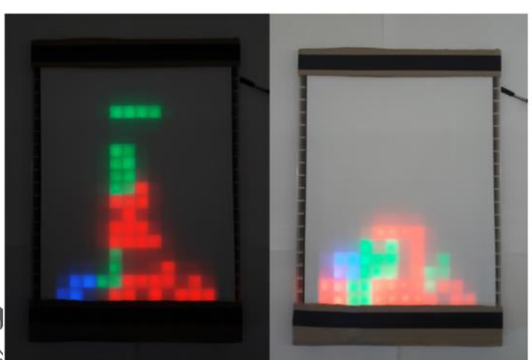
RGB Strip Lights
RGB Strook Ligte

RGB strip lights

- String of RGB LEDs, with integrated control electronics
- “PWM” serial interface for all LEDs in the strip
- Daisy chain configuration: DOUT of LED1 connected to DIN of LED2
- Per WS2818B LED, need 24-bits of information (8-bit R, G and B). Per bit of information, PWM duty cycle <50% = 0, PWM duty cycle>50% = 1
- Microcontroller implementation: Use PWM output (Timer, with PWM Output mode) and DMA (Direct Memory Access). DMA will automatically load the next value for the Compare (Pulse) register from memory array



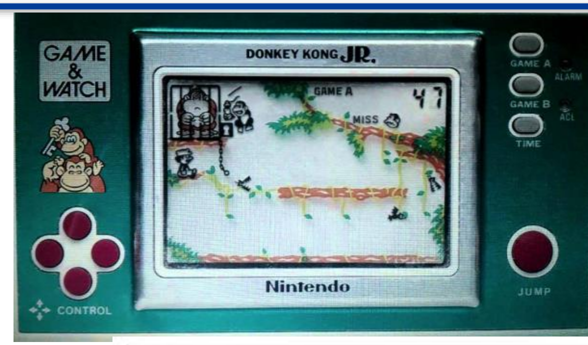


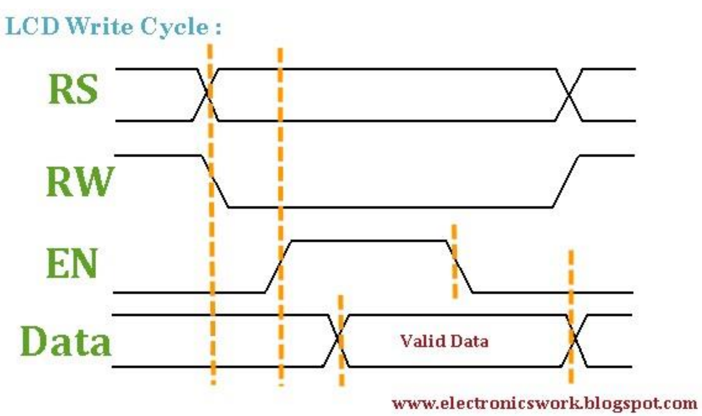


LCD Character Display
LCD Karakter vertooneenheid

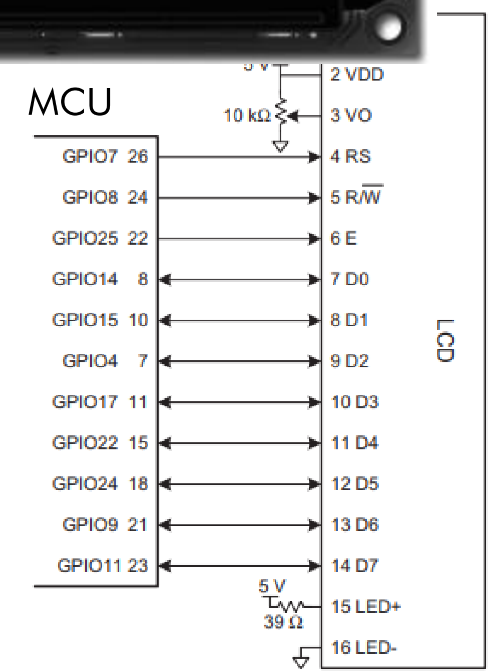
Liquid Crystal Display (LCD)

- Uses light-modulating properties of liquid crystals to change colour
- LCD does not emit light – usually accompanied by a backlight
- LCD character display: Has a matrix of LCD "pixels", but we (the microcontroller) does not control individual pixels. Rather the microcontroller tells the display which (ASCII) characters to display
  - Parallel GPIO interface
  - "Bit-banging" type control of signals
  - Review lecture 10





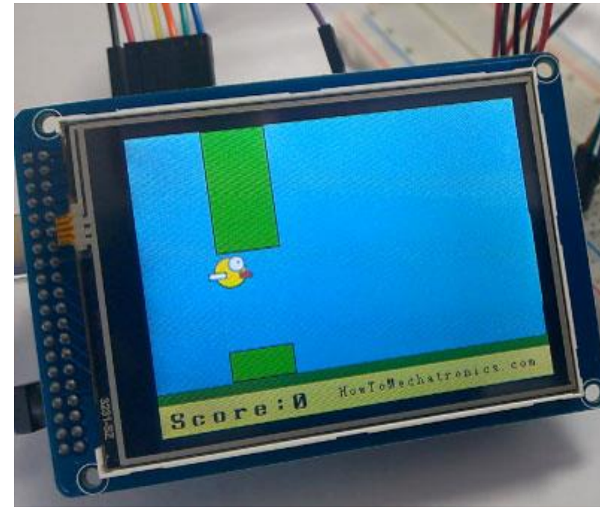
PIN ASSIGNMENT		
Pin no.	Symbol	Function
1	Vss	Power supply(GND)
2	Vdd	Power supply(+)
3	Vo	Contrast Adjust
4	RS	Register select signal
5	R/$\overline{W}$	Data read / write
6	E	Enable signal
7	DB0	Data bus line
8	DB1	Data bus line
9	DB2	Data bus line
10	DB3	Data bus line
11	DB4	Data bus line
12	DB5	Data bus line
13	DB6	Data bus line
14	DB7	Data bus line
15	A	Power supply for LED B/L (+)
16	K	Power supply for LED B/L (–)

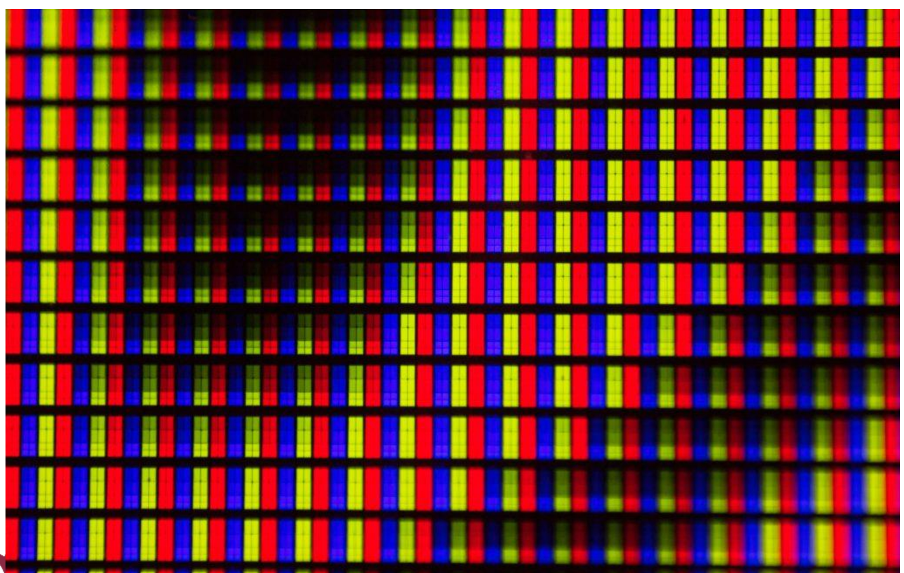


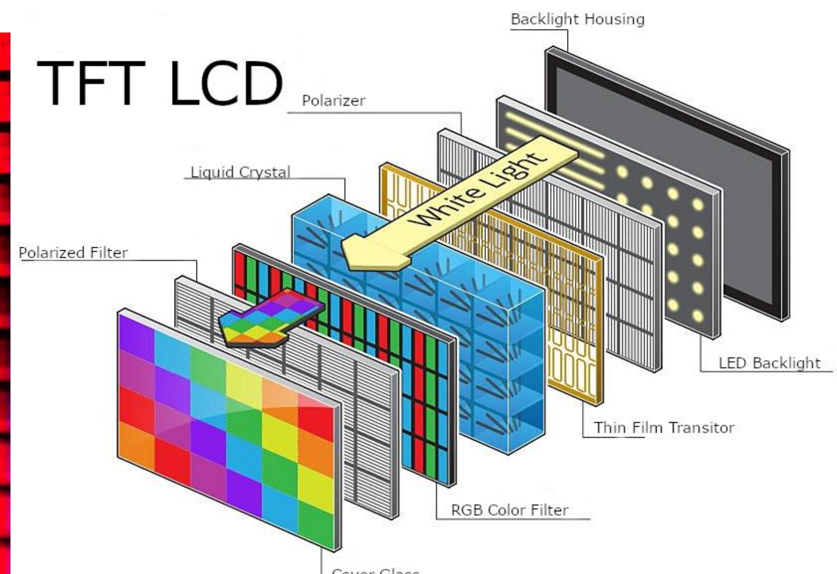
LCD-TFT Display
LCD dun film transistor vertooneenheid

LCD-TFT: Thin Film Transistor LCD display

- Modern display technology used in laptops, desktop display, TVs, projectors
- Matrix of many pixels, each with separate RGB intensity
- (Btw, LED TV is really still LCD, it just uses LEDs for the backlight. OLED = organic LED technology is different. With OLED there is no backlight and each pixel is organic LED emitting the light)
- Cannot control this with GPIO. Too many signals!





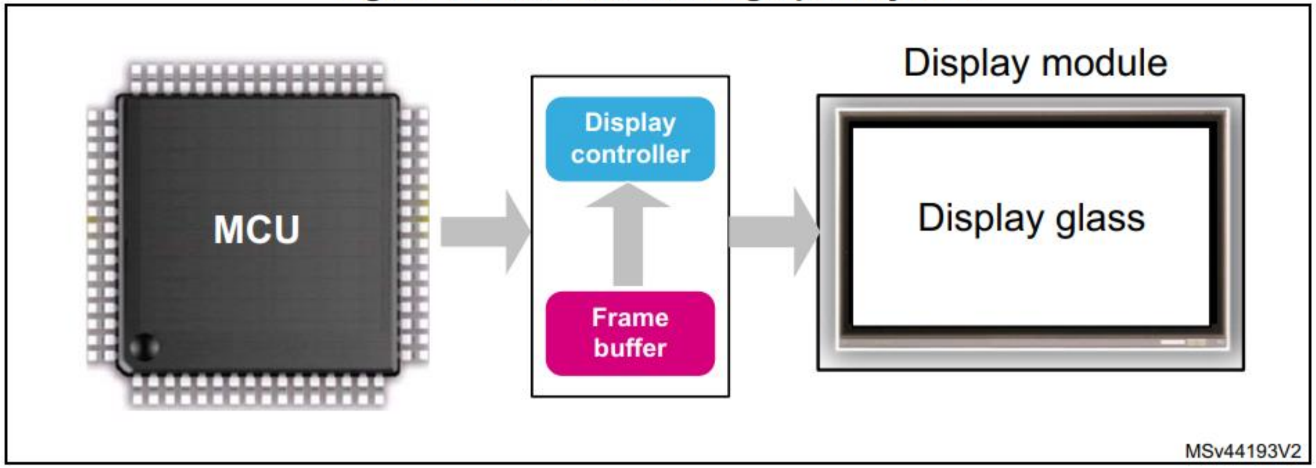


https://aoi-display.com/AOI/tft-technology/

LCD-TFT Display
Vloeibare kristal dun film transistor vertooneenheid

- Microcontroller with display basic block diagram

Figure 1. Basic embedded graphic system

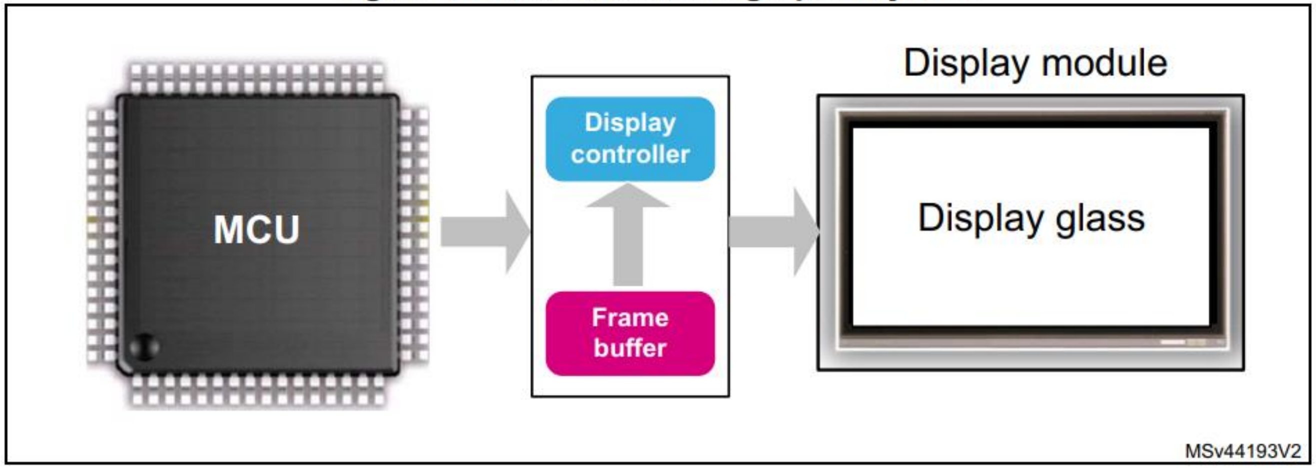


- MCU computes what to display at pixel level (text, sprites, background image, etc.) by writing data into the frame buffer
- Frame buffer: volatile memory used to store pixel data to be displayed. Also called Graphics RAM or GRAM
- Display controller: control electronics that continuously refreshes the display by transferring frame buffer content to the display

LCD-TFT Display
Vloeibare kristal dun film transistor vertooneenheid

- Microcontroller with display basic block diagram

Figure 1. Basic embedded graphic system



- Display characteristics:
  - Display size/resolution: number of vertical and horizontal pixels
  - Color depth: number of colours in which a pixel can be drawn. Usually represented in bits-per-pixel (bpp) units. For instance, colour depth of 24 bpp (usually 8-bits each for red, green and blue), there are 16777216 colour combinations
    - Frame buffer size has to be at least width * height * bpp/8 bytes
  - Refresh rate (in Hz): the number of times per second that the display controller refreshes the display. Usually 60Hz or higher, otherwise we would see flickering effects

LCD-TFT Display

Vloeibare kristal dun film transistor vertooneenheid

- Different display architectures
  - Framebuffer and display controller embedded with display
  - Framebuffer (RAM) and display controller embedded in microcontroller
  - External RAM used for framebuffer, and display controller embedded in MCU
- It gets more complicated when there is a dedicated GPU (Graphics Processing Unit)

Figure 2. Display module with embedded controller and GRAM

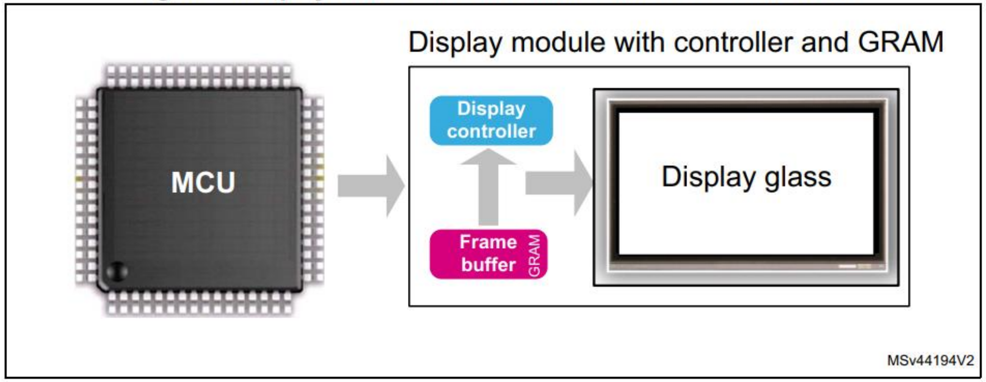


Figure 3. Display module without controller nor GRAM

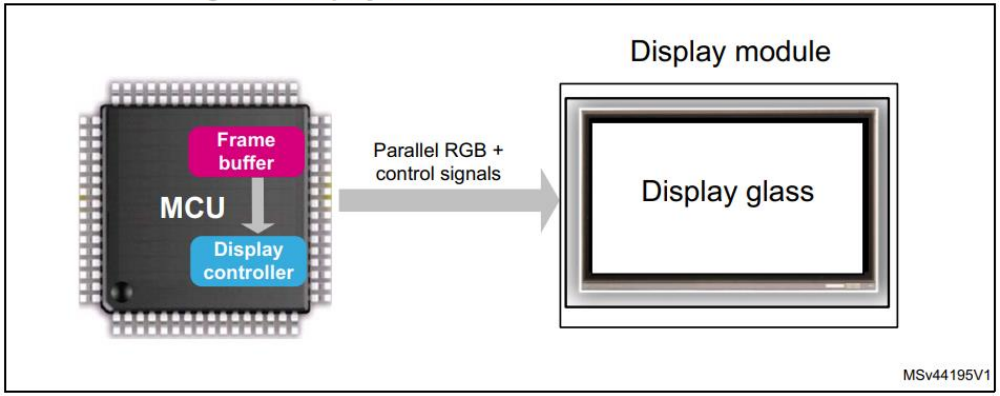
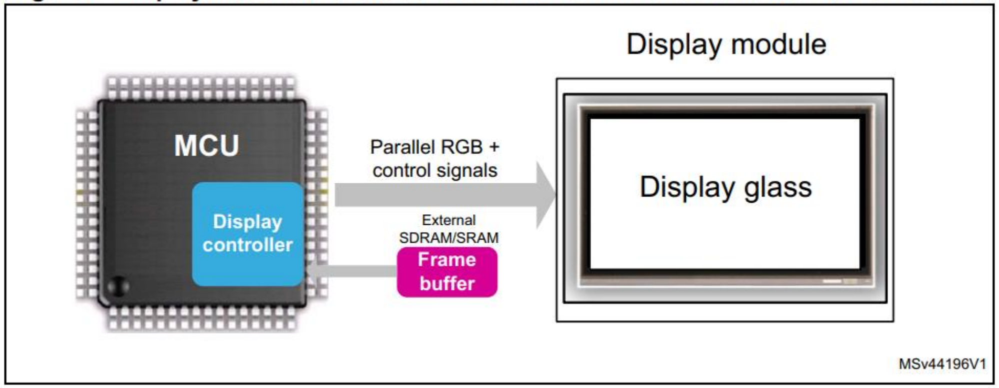


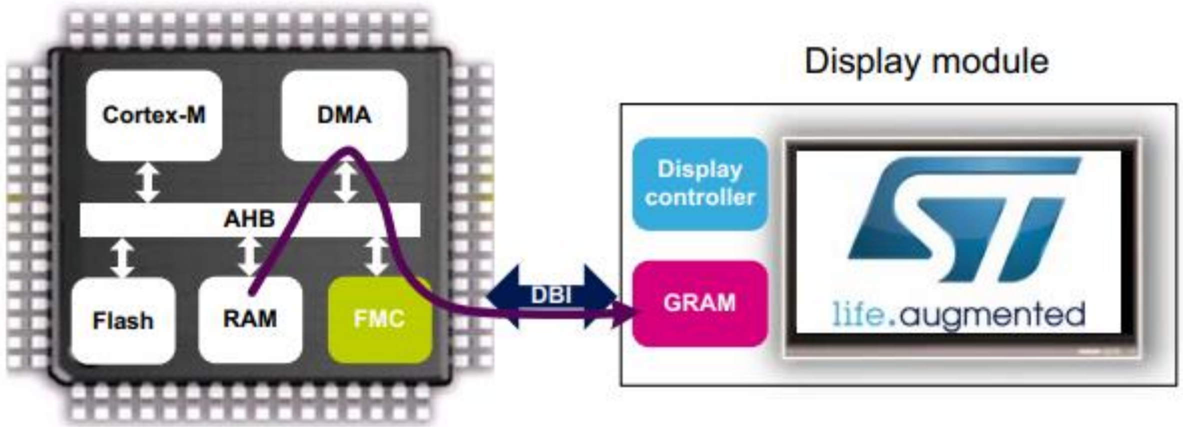
Figure 4. Display module without controller nor GRAM and with external framebuffer



LCD-TFT Display

Vloeibare kristal dun film transistor vertooneenheid

- There are different interfacing methods depending on architecture
- 1. If the Graphics RAM and Display controller are embedded in the display, and the display module has a parallel interface

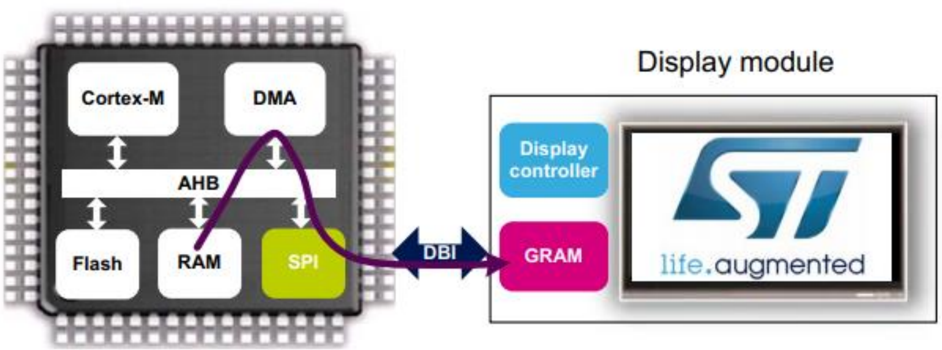


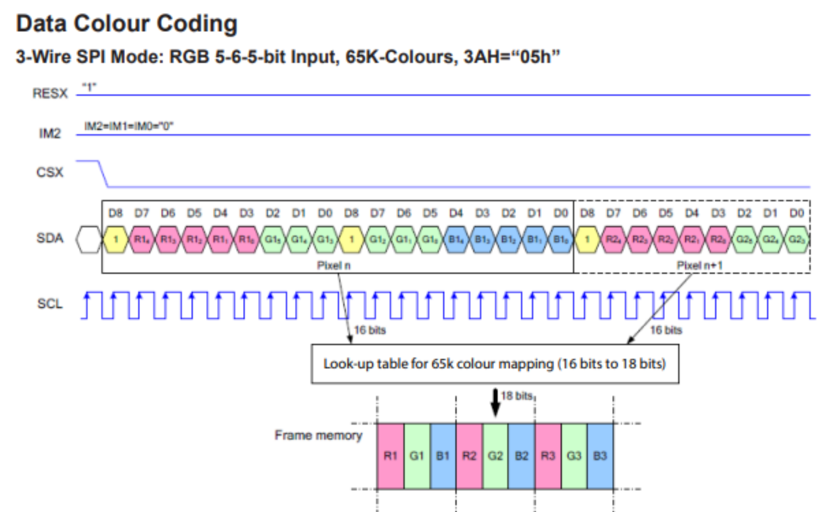
- Graphics RAM is accessed by the MCU as if it is external memory
- The MCU needs an external memory interface for this (FMC = Flexible Memory Controller = the peripheral for external memory access in STM MCUs)
- Some LCDs have Intel 8080 or Motorola 6800 memory bus interface: 8-bit parallel bi-directional data bus, with enable, R/W (read/write), command/data selection signal
- Use DMA to transfer the data from internal MCU RAM to external memory controller (to GRAM)

LCD-TFT Display

Vloeibare kristal dun film transistor vertooneenheid

- There are different interfacing methods depending on architecture
- 2. If the Graphics RAM and Display controller are embedded in the display and the display has a serial interface

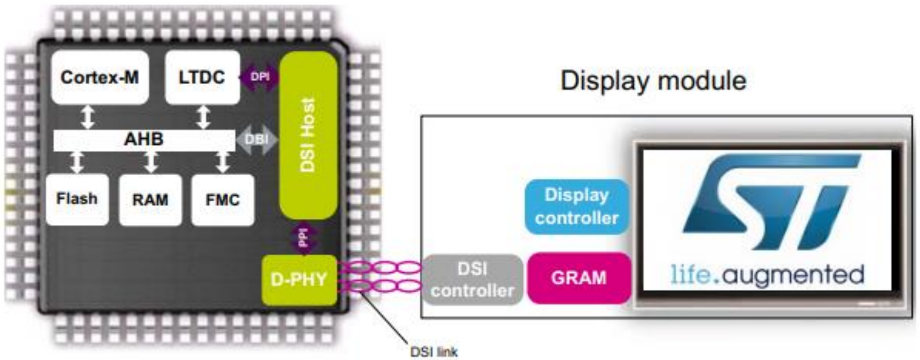




- Graphics data is transferred to display module through SPI peripheral
- Also use DMA to transfer data from internal RAM to SPI peripheral

LCD-TFT Display
Vloeibare kristal dun film transistor vertooneenheid

- There are different interfacing methods depending on architecture
- 2. If the Graphics RAM and Display controller are embedded in the display and the display has a serial interface

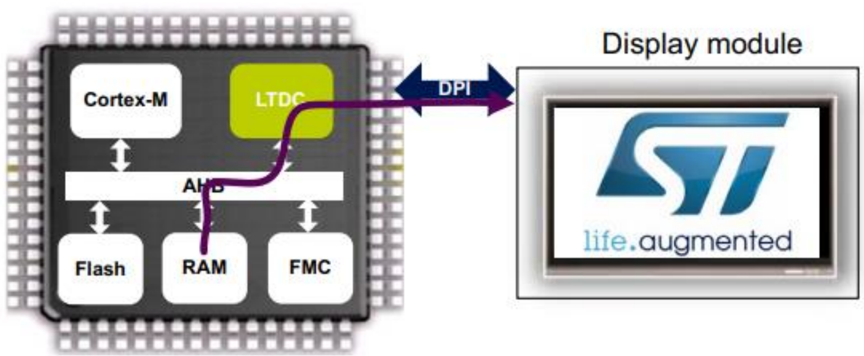


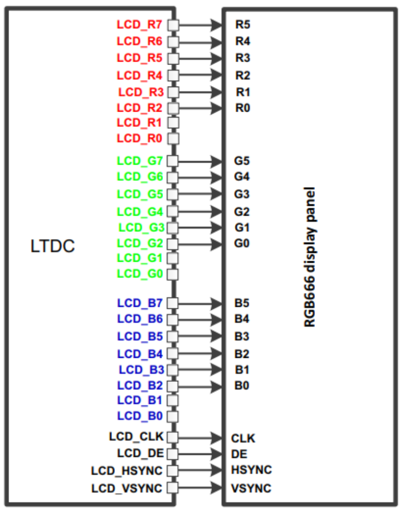
- There is also something called a “Display Serial Interface” – DSI.
  - Differential signaling
  - One clock and at least one serial data signal
  - High data rates
  - (Raspberry Pi has such an interface)
  - The MCU needs a DSI Host controller peripheral embedded

LCD-TFT Display

Vloeibare kristal dun film transistor vertooneenheid

- There are different interfacing methods depending on architecture
- **3. If the display does not have embedded GRAM or Display Controller**





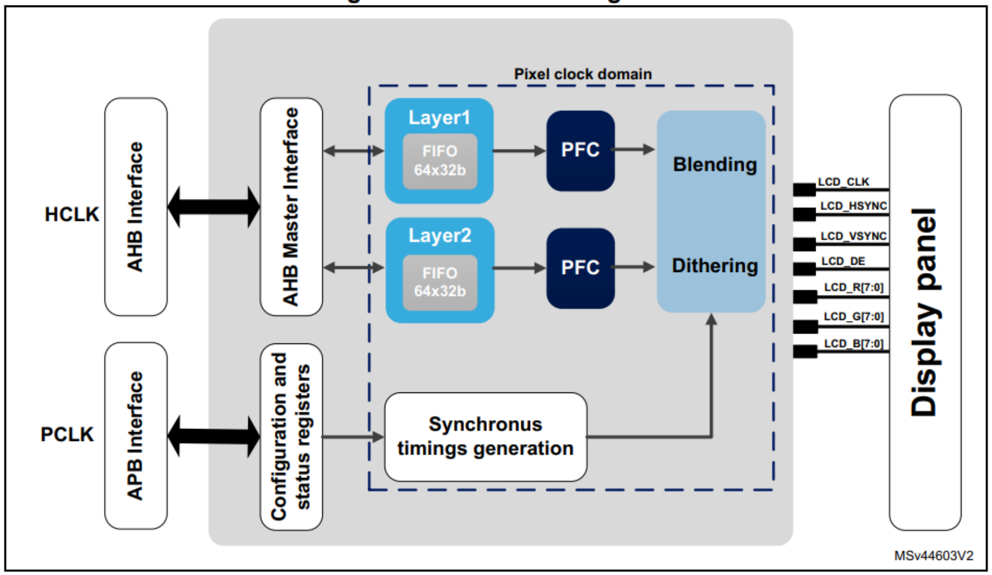
- Frame data is transferred using parallel interface (DPI = Display Parallel Interface)
- Data is clocked out, pixel-for-pixel starting at top-left screen coordinate, line-by-line
- Synchronisation signals, to signal start of a line (HSYNC), and start of a frame (VSYNC)
- Pixel clock (PCLK) indicates when following pixel data is available
- Pixel data is output as parallel red, green and blue values
- "GRAM" is embedded in MCU (or it could be a separate, external memory device connected to the MCU and accessed through an external memory interface)
- MCU needs a Display Controller peripheral. STM microcontrollers have a LTDC – LCD-TFT Display Controller peripheral

LCD-TFT Display

Vloeibare kristal dun film transistor vertooneenheid

- STM LCD-TFT Display Controller (LTDC) peripheral – functional description

Figure 11. LTDC block digram

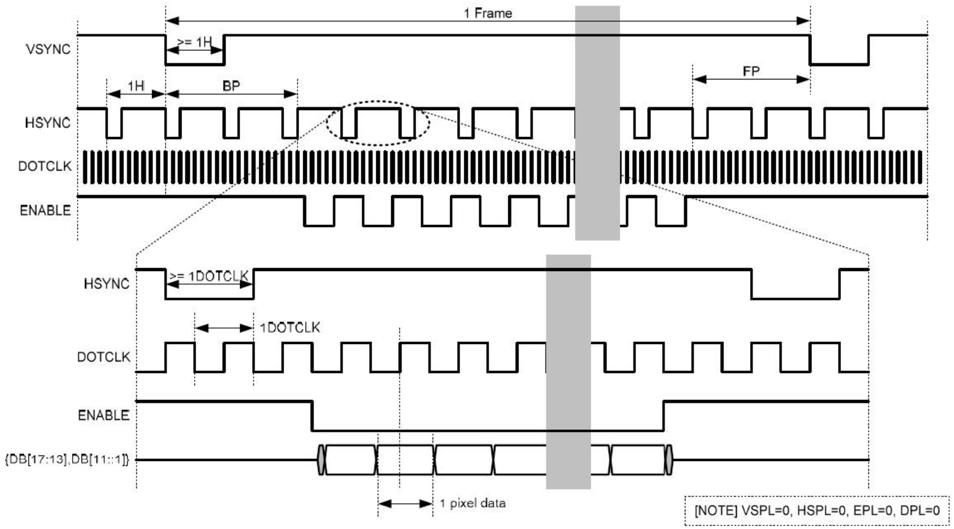


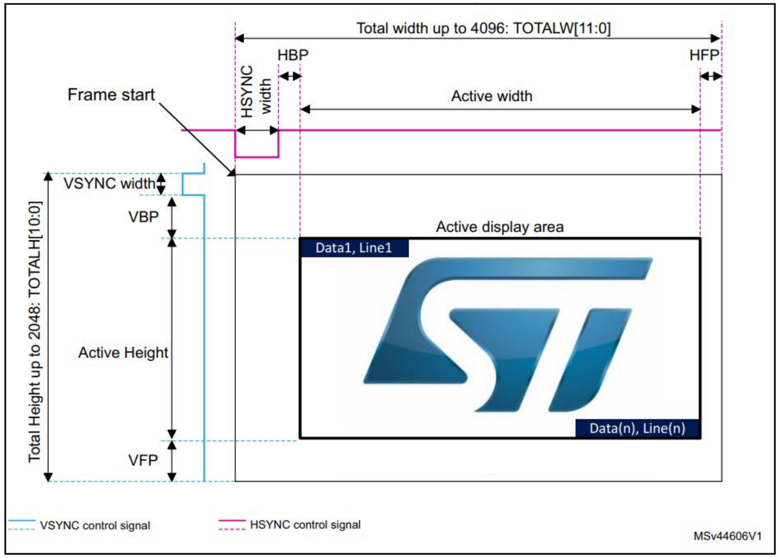
- LTDC peripheral acts a bit like DMA – retrieving data from memory through the AHB interface in the background. But then writes it out to parallel display interface
- The LDTC peripheral has the ability to retrieve data for two rectangular regions, and blend them before outputting to display
- Control and setup of the LDTC peripheral is through memory mapped registers (same as with all other peripherals)

LCD-TFT Display

Vloeibare kristal dun film transistor vertooneenheid

- STM LCD-TFT Display Controller (LTDC) peripheral – <u>**signals**</u>



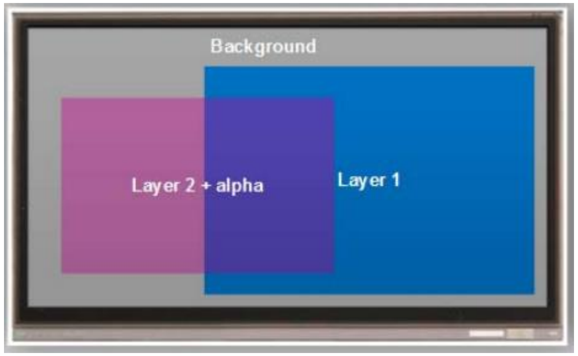


- VSYNC (vertical sync) pulses once per frame
- HSYNC (horizontal sync) pulses once per line
- Pixel clock (PCLK or DOTCLK) cycles for every pixel
- Data Enable (DE) signal indicates when RGB data is valid. Pixel clock might be running but Data Enable signed is not enabled – this is called the Back Porch and Front Porch regions (BP and FP)

LCD-TFT Display

Vloeibare kristal dun film transistor vertooneenheid

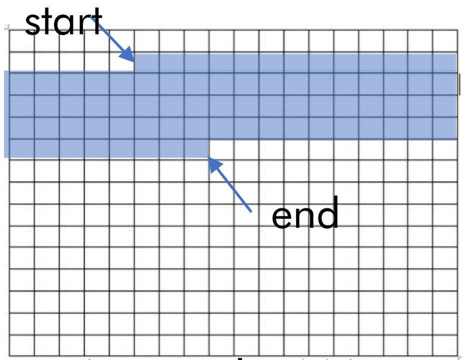
- STM LCD-TFT Display Controller (LTDC) peripheral – <u>**programming**</u>
- Use the LTDC peripheral registers to setup up:
  - Display resolution and colour depth/pixel format
  - Timings for VSYNC, HSYNC, PCLK
  - Duration for Front Porch and Back Porch
  - Polarity for VSYNC, HSYNC, PCLK
  - Memory address and size of frame buffer (also for second layer if you want to use it)
  - (Conveniently done using the device configuration tool and HAL framework)
- To display stuff, write to memory at the framebuffer address



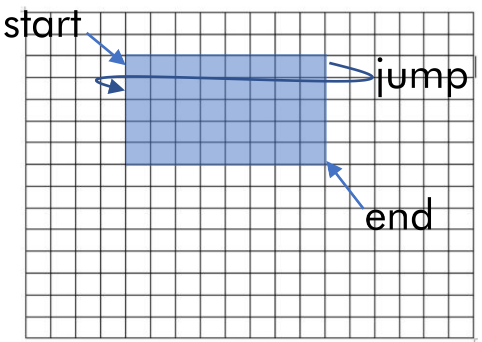

```c
static void MX_LTDC_Init(void)
{

  /* USER CODE BEGIN LTDC_Init 0 */

  /* USER CODE END LTDC_Init 0 */

  LTDC_LayerCfgTypeDef pLayerCfg = {0};
  LTDC_LayerCfgTypeDef pLayerCfg1 = {0};

  /* USER CODE BEGIN LTDC_Init 1 */

  /* USER CODE END LTDC_Init 1 */
  hltdc.Instance = LTDC;
  hltdc.Init.HSPolarity = LTDC_HSPOLARITY_AL;
  hltdc.Init.VSPolarity = LTDC_VSPOLARITY_AL;
  hltdc.Init.DEPolarity = LTDC_DEPOLARITY_AL;
  hltdc.Init.PCPolarity = LTDC_PCPOLARITY_IPC;
  hltdc.Init.HorizontalSync = 40;
  hltdc.Init.VerticalSync = 9;
  hltdc.Init.AccumulatedHBP = 53;
  hltdc.Init.AccumulatedVBP = 11;
  hltdc.Init.AccumulatedActiveW = 533;
  hltdc.Init.AccumulatedActiveH = 283;
  hltdc.Init.TotalWidth = 565;
  hltdc.Init.TotalHeigh = 285;
  hltdc.Init.Backcolor.Blue = 0;
  hltdc.Init.Backcolor.Green = 0;
  hltdc.Init.Backcolor.Red = 0;
  if (HAL_LTDC_Init(&hltdc) != HAL_OK)
  {
    Error_Handler();
  }
  pLayerCfg.WindowX0 = 0;
  pLayerCfg.WindowX1 = 480;
  pLayerCfg.WindowY0 = 0;
  pLayerCfg.WindowY1 = 272;
  pLayerCfg.PixelFormat = LTDC_PIXEL_FORMAT_RGB565;
  pLayerCfg.Alpha = 255;
  pLayerCfg.Alpha0 = 0;
  pLayerCfg.BlendingFactor1 = LTDC_BLENDING_FACTOR1_PAxCA;
  pLayerCfg.BlendingFactor2 = LTDC_BLENDING_FACTOR2_PAxCA;
  pLayerCfg.FBStartAdress = 0xC0000000;
  pLayerCfg.ImageWidth = 480;
  pLayerCfg.ImageHeight = 272;
  pLayerCfg.Backcolor.Blue = 0;
  pLayerCfg.Backcolor.Green = 0;
  pLayerCfg.Backcolor.Red = 0;
  if (HAL_LTDC_ConfigLayer(&hltdc, &pLayerCfg, 0) != HAL_OK)
  {
    Error_Handler();
  }
```

LCD-TFT Display
Vloeibare kristal dun film transistor vertooneenheid

- STM LCD-TFT Display Controller (LTDC) peripheral – DMA2D
- The DMA2D controller behaves similar to normal DMA controller, but intended for memory-to-memory transfers (not to/from peripherals)
- The DMA2D controller operates on rectangular regions of memory



Normal DMA



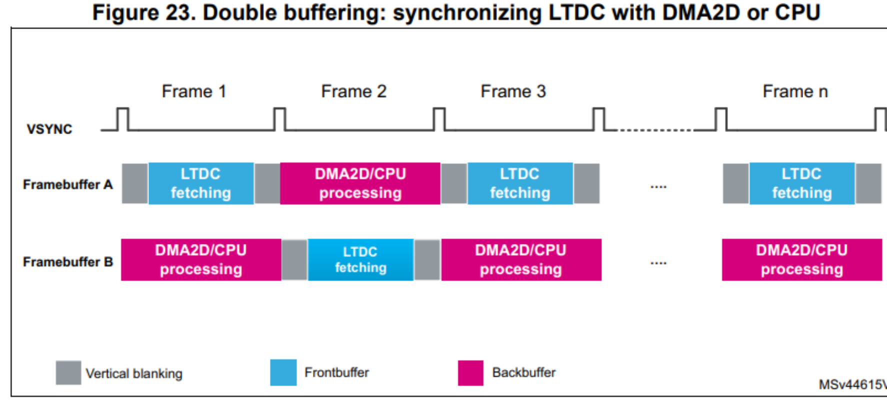
DMA2D

- The DMA2D implements four basic tasks:
  - Fill a rectangular shape with a unique colour.
  - Copy a frame or a rectangular part of a frame from a memory to another.
  - Convert the pixel format of a frame or a rectangular part of a frame while transferring it from one memory to another memory.
  - Blend two images with different sizes and pixel format and store the resulting image in one resulting memory

LCD-TFT Display
Vloeibare kristal dun film transistor vertooneenheid

- STM LCD-TFT Display Controller (LTDC) peripheral – <u>Double buffering</u>
- Screen tearing, flickering, “strange sprite movement” effects happen because the CPU is drawing to the same memory, at the same time that the Display Controller is refreshing the display
- One way to fix it, is to wait for VSYNC and then draw on the framebuffer parts that have already been refreshed
- A better (and more common way) is to use double-buffering:
  - Two frame buffers
  - While the display controller is refreshing from frame A (the “front buffer”), draw to frame B (the “back buffer”) using CPU or DMA2D
  - At the end of the refresh cycle, flip the buffers. New back buffer = frame A, new front buffer = frame B.
  - Now display controller refreshes from frame B, while drawing happens on Frame A

Figure 23. Double buffering: synchronizing LTDC with DMA2D or CPU



LCD-TFT Display
Vloeibare kristal dun film transistor vertooneenheid

- LCD display: Other signals
  - Reset signal – Use GPIO
  - I2C or SPI – used to control other aspects of integrated display, for instance touch screen position sensing, brightness, contrast
  - Backlight: GPIO signal from MCU to enable

- Reference for images in these slides:
  - ST Application Note AN4861: LCD-TFT display controller (LTDC) on STM32 MCUs