Embedded Graphics Possibilities Using STM32

Mike Hartmann
Staff FAE, Microcontrollers
• In this presentation we will cover:
  • Overview of Graphics on Microcontrollers
  • Graphic peripherals available within the STM32 Family
  • Common display types and resolutions supported
  • Performance features available to fully optimize and improve STM32 CPU performance enhancing your next embedded design
  • An overview of the STM32 ecosystem showing available hardware, software and documentation necessary to realize your next graphics based embedded design
Graphics Overview
Graphics on a Microcontroller

- Microcontrollers bring low cost, low power, and relatively low complexity

- Steps to display graphics:

Ex: 480x272x16bit ~256kB

One shot refresh

Continuous refresh (usually 30 – 60 fps)
Hardware Configurations

1 chip STM32 (embedded GRAM and Controller) + Display (LTDC – DSI)

2 chips SDRAM & STM32 (embedded Controller) + Display (LTDC – DSI)

1 chip STM32 + Display Module (SPI – FMC)

2 chips STM32 & GFX Controller + Display (SPI – FMC)
RAM Usage in Graphics

- Image must be put together in an area of RAM called a Framebuffer.
  - Size is width x height x bytes_per_pixel
  - Bytes per pixel, also known as color depth, usually 2 (16 bit) or 3 (24 bit), but may also be 1 or 4.
  - Size is constrained by available SRAM, more often requires external SDRAM.

- Can be single or double buffered
  - Double buffering avoids a problem called tearing, where parts of two frames are displayed
RAM Usage in Graphics

- Framebuffers are also known as GRAM

- Layers can be split between SDRAM or internal SRAM and GRAM on display device.
  - At least one framebuffer is required on STM32 side

![Diagram showing RAM usage in graphics]

- STM32 SRAM/SDRAM
- Framebuffer
- Display device
- DISPLAY GRAM
• Touch uses an external controller

• Use Discovery kits (STM32F746, for example) as models
STM32 Graphics Hardware
## STM32 Graphics Capability

<table>
<thead>
<tr>
<th>Series</th>
<th>Core</th>
<th>Frequency</th>
<th>Graphic Acceleration</th>
<th>Display IF</th>
<th>Resolutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>STM32F1</td>
<td>Cortex-M3</td>
<td>72 MHz</td>
<td>-</td>
<td>8080/6800 parallel IF</td>
<td>CGA/QVGA 320 x 200/320 x 240</td>
</tr>
<tr>
<td>STM32F2</td>
<td>Cortex-M3</td>
<td>120 MHz</td>
<td>-</td>
<td>8080/6800 parallel IF</td>
<td>CGA/QVGA 320 x 200</td>
</tr>
<tr>
<td>STM32L4</td>
<td>Cortex-M4</td>
<td>80 MHz</td>
<td>Chrom-ART</td>
<td>8080/6800 parallel IF</td>
<td>QVGA/WQVGA 320 x 240/480 x 272</td>
</tr>
<tr>
<td>STM32F4 Access and Foundation lines</td>
<td>Cortex-M4</td>
<td>100 to 180 MHz</td>
<td>-</td>
<td>8080/6800 parallel IF</td>
<td>QVGA/WQVGA 320 x 240/480 x 272</td>
</tr>
<tr>
<td>STM32F4 Advanced lines</td>
<td>Cortex-M4</td>
<td>180 MHz</td>
<td>Chrom-ART</td>
<td>8080/6800 parallel IF LCD TFT controller MIPI-DSI</td>
<td>Up to XGA</td>
</tr>
<tr>
<td>STM32F7</td>
<td>Cortex-M7</td>
<td>216 MHz</td>
<td>Chrom-ART HW JPEG</td>
<td>8080/6800 parallel IF LCD TFT controller MIPI-DSI</td>
<td>Up to XGA 1024 x 768</td>
</tr>
</tbody>
</table>
Most Common Supported Resolutions

Targeted Resolutions (survey)

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>WXGA 1080x768</td>
<td>1</td>
</tr>
<tr>
<td>XGA 1024x768</td>
<td>11</td>
</tr>
<tr>
<td>SVGA 800x600</td>
<td>13</td>
</tr>
<tr>
<td>WVGA 800x480</td>
<td>20</td>
</tr>
<tr>
<td>VGA 840x480</td>
<td>18</td>
</tr>
<tr>
<td>WQVGA 480x272</td>
<td>17</td>
</tr>
<tr>
<td>QVGA 320x240</td>
<td>12</td>
</tr>
<tr>
<td>CGA 320x200</td>
<td>12</td>
</tr>
</tbody>
</table>

Targeted Color Depths (survey)

- 16 bits per pixel: 47.73%
- 24 bits per pixel: 33.05%
- 8 bit per pixel: 18.18%
Graphics Hardware

- **Peripherals**
  - LTDC (STM32F7x6, STM32F7x9, STM32F4x9)
  - DSI (STM32F7x9, STM32F469)
  - FMC (for Motorola/8080)
  - Chrom-ART (F4/F7)
  - Touch done via i2c or spi (no internal touch controller)
    - Use whichever touch controller you like

- **Sizes**
  - Any up to 1024x768
  - Known examples: 320x200 up to 1024x768
    - Popular sizes are generally ones found on ST Evaluation hardware (EVAL, Discovery)
FMC/SPI Interfaces

- These require an external display controller
- Can be used with parts that don’t have LTDC controller
- Low end, we won’t cover these further, but are the best option for some cases
• LCD TFT Display Controller

• Parallel RGB interface

• 2 layers
  • For double buffering
• Display Serial Interface, part of MIPI specs

• Wraps LTDC

• Uses LTDC for timing info
Performance Features

• ChromART
  • Specialized DMA dedicated to graphics manipulation
  • Can copy or blend two sources
  • Can do pixel format conversion
  • Can do partial framebuffer updates
  • Supports formats from 4 bit up to 32 bit
    • 4 or 8 bit indexed
    • 4 or 8 bit alpha channels
    • 16 to 32 bit RGB/ARGB formats
  • Both Direct and indirect color modes, internal CLUT memory for indirect modes
• Also known as DMA2D
  • Specialized DMA controller for graphics handling

• Two inputs, one output

• Blender and color converter
Performance Tips

• Use DSI adapted command mode for static images
  • Requires part that supports DSI (F469/479, F769/779, H7)
  • Partial update feature for small animations or small updates can save bandwidth and cycles

• Other Tricks
  • Preload frequently used small images in SRAM or DTCM (F7, H7 only).
STM32 Graphics Ecosystem
What Can ST Provide to Get You Started?

• Boards
  • Ecosystem of many evaluation platforms such as Discovery and Eval boards

• STM32Cube HAL

• Middleware Libraries such as STemWin, FreeRTOS, and JPEG for F769
  • Other middleware libraries for the rest of your application (USB, network, FATFS, etc)

• Our graphics partners are also ready to help
Ecosystem – Discovery Boards

32F429IDISCOVERY
- STM32F429
- 320x240 QVGA LCD
- 64 MBits SDRAM

32F746GDISCOVERY
- STM32F746
- 480x272 WQVGA LCD
- 64 MBits SDRAM
- 128 Mbit QSPI Flash
- Arduino Uno

32F469IDISCOVERY
- STM32F469
- 800x480 WVGA LCD
- 128 MBits SDRAM
- 128 Mbit QSPI Flash
- Arduino Uno
Ecosystem – Evaluation Boards

STM32429I-EVAL
- STM32F429
- 480x272 WQVGA LCD
- 256 MBits SDRAM
- 128 Mbits NOR Flash

STM32439I-EVAL
- STM32F439
- 640x480 VGA LCD
- 256 MBits SDRAM
- 128 Mbits NOR Flash

STM32756G-EVAL
- STM32F756
- 640x480 VGA LCD
- 256 MBits SDRAM
- 128 Mbits NOR Flash
- 512 Mbits QSPI Flash

STM32469I-EVAL
- STM32F469
- 800x480 WVGA LCD
- 256 MBits SDRAM
- 128 Mbits NOR Flash
- 512 Mbits QSPI Flash
STM32 Graphic Ecosystem
3 Recommended SW Solutions

- **Entry Solution**
  - STemWin
  - SEGGER

- **Advanced Solutions**
  - TouchGFX
  - Embedded Wizard
## STM32 Graphics Tool Options

<table>
<thead>
<tr>
<th></th>
<th>STemWin</th>
<th>TouchGFX</th>
<th>Embedded Wizard</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Provider</strong></td>
<td>ST (part of CUBE middleware)</td>
<td>Draupner Graphics</td>
<td>TARA systems</td>
</tr>
<tr>
<td><strong>Targets</strong></td>
<td>Simple GUI with limited animations</td>
<td>Advanced GUI</td>
<td>Advanced GUI</td>
</tr>
<tr>
<td><strong>Programming language</strong></td>
<td>C</td>
<td>C++</td>
<td>Chora (TARA's Object-oriented language)</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td>+</td>
<td>+++</td>
<td>+++</td>
</tr>
<tr>
<td><strong>Resource optimization</strong></td>
<td>++</td>
<td>+++</td>
<td>+++</td>
</tr>
<tr>
<td><strong>Supported color formats</strong></td>
<td>1, 8, 16 and 32 bpp</td>
<td>1, 16 and 24 bpp</td>
<td>1, 8, 16, 24 and 32 bpp</td>
</tr>
<tr>
<td><strong>Tools</strong></td>
<td>Bitmap converter Fonts converter GUI simulator GUI builder (generating widgets 'backbones' only)</td>
<td>Image converter Text converter Fonts converter GUI simulation GUI builder</td>
<td>Embedded Wizard studio: an integrated IDE with simulator and GUI builder with full code generation capability</td>
</tr>
<tr>
<td><strong>Licensing</strong></td>
<td>Free for STM32 devices</td>
<td>Free evaluation version (full features, watermarked) License fee for production</td>
<td>Free evaluation version (time and features limited) License fee for production</td>
</tr>
</tbody>
</table>
STemWin
STemWin

- Free ST branded toolkit (for STM32 parts only)
- Includes GUIBuilder, font converter, bitmap converter, VNC client/server, and an mjpeg creator.
- Limited functionality, but comes with basic widget set
- Already integrated with Cube HAL
STemWin

• No code generation except widget creation from GUIBuilder

• Simulator available (Windows only, requires Visual Studio)

• Programming language is C

• Provides moderate performance and reasonable resource usage
**Elements:**
- Widget bar

**Workspace showing**
- Basic Window with:
  - Text widget and menu
  - Button
  - slider
  - Graph Widget

- Widget Tree

- Properties
About TouchGFX

• TouchGFX, a C++ framework for modern Graphics on MCUs from Draupner Graphics A/S

• Object oriented design, with strong emphasis on quality, performance while minimizing footprint, through static memory allocations
  • Framework requires only 10-20 Kb of RAM, 20 Kb of Flash
  • Application’s widgets 1-15 Kb of RAM, 1-100 Kb of Flash
  • Framebuffer in internal SRAM or external SDRAM
Max UI Performance on STM32

- Current 2D hardware acceleration uses Chrom-ART for Fills (with alpha-blending) and Blits in RGB 16bit, 24bit, ARGB (32bit), A4 and A8 pixel formats

- Offers an easy platform integration for everything 2D hardware acceleration, display and touch sensing controllers
TouchGFX Technology

Advanced Rendering Algorithms
Optimized visible surface determination algorithm and customized invalidation techniques minimize the number of drawn pixels.

Model-View-Presenter software pattern, provides a clean split between application state and UI

Easy Creation of Custom Controls
Create custom controls by extending or modifying existing widgets or by combining existing controls with custom functionality.

Advanced Graphical Objects
Draw lines, circles, custom shapes, and graphics, or apply scaling and 3D rotation to images at runtime with highly optimized and memory efficient algorithms.
TouchGFX Technology

- TrueType and OpenType, kerning, multi-script font support (e.g. Latin, Chinese, Arabic)

- Win32 PC simulator (Visual Studio) for fast UI design and prototyping

- TouchGFX provides support for 1bpp, 2bpp, 4bpp, 16bpp and 24bpp displays, DPI, DBI and DSI display interfaces

- Display framebuffer could be located in internal SRAM or in external SDRAM (when available)
Key Features

Easy Development

• Use the graphical WYSIWYG tool, TouchGFX Designer, and create your own prototype in minutes.

• Choose your preferred IDE for development

• Support for all major compilers: IAR, Keil, GCC.

• Run your application on any STM32L4, STM32F4 & F7 display board

• Try before buying: Get a free and fully functional evaluation version.

• Graphic designer not needed, but helpful
TouchGFX Key Features

• Easily ported to any STM32 that can handle a display

• Minimal footprint

• Selection of demo projects for all graphics oriented STM32 parts
  • touchgfx.com

• Already uses Cube HAL, build system can be configured to use any HAL tree

• Comes with simulator for prototyping
TouchGFX Designer

- Features include multi-screen applications, a large palette of ready-to-use widgets, skins (themes), interactions (trigger/actions callbacks), custom containers creation (code reuse), per-widget font and typography configuration

- Designer generates all the UI C++ codes with subclasses for the instantiated widgets, that can be manually extended
  - Also the text.xlsx Excel sheet for all text strings referenced by the application

- Designer uses MinGW/GCC to build a Win32 simulator and the target’s BSP to build and flash the UI prototype on the target MCU
What is Embedded Wizard?

• What is Embedded Wizard?
  • Product of TARA Systems, an ISV specialized for embedded systems based in Munich, Germany
  • GUI development and prototyping tool with code generation model – not “only” a pure graphics library
  • MCU and MPU type target hardware evolved over 20 years
  • Target Framework Memory Footprint: 32KB, 48KB (Index8 only)

• Target Markets
  • Consumer Electronics & Home Appliances
  • Industrial, Automation & Medical
  • Automotive
Architecture

Embedded Wizard Studio

Developer PC

Embedded Wizard Platform Package
- Code Generator
- Resource Converter

Target
- Runtime Environment
- Graphics Engine
Embedded Wizard Key Features

• Comfortable IDE with drag & drop

• Visual programming with WYSIWYG and instant prototyping of UI look and feel

• Simple programming model including object-oriented programming support, generating ANSI C

• Platform independent implementation of GUI logic

• No (RT)OS (i.e. tasks, semaphores, etc.) is required, GUIs can run on bare metal
Embedded Wizard Key Features

• Ready-to-use widgets as templates for state-of-the-art designs, including effects (rotation, scaling & perspective transformation each with Hi- and Low-Quality), animations, layout functions, etc.

• (Multi-)Touch, Gestures, Mouse, Remote Control support

• UNICODE based

• Supports various color depths/formats: RGBA8888, RGB888, RGBA4444, RGB565, Index8, LumA44
Embedded Wizard Key Features

• Native language is Chora
  • Invented by Tara Systems
  • Object oriented
  • Specifically intended for GUI building
  • Gets converted to C during build process

• Can interface to external C/C++ libraries

• Includes simulator for prototyping
Embedded Wizard Workspace
Third Party
What’s Coming

• Integration with Cube HAL and CubeMX
  • More examples for STemWin
  • Eval versions of TouchGFX and Embedded Wizard included, with examples
  • Configure any graphics project from within CubeMX
Summary

- Reasons to chose an STM32 for your next graphics project
  - Extensive ecosystem of hardware and software
  - Graphics partners give you a selection of entry level to advanced GUI tools
  - Graphics related peripherals provide outstanding graphics performance
  - ST has a range of STM32s to meet the complexity levels and costs you need
Thank you

www.st.com/stm32