An Introduction to OpenVG™

FTF-AUT-F0465

Oliver Tian | Auto FAE

MAY. 2014
Agenda

• Trend of Graphics in Vehicle
• Roadmap of Cluster
• Introduction of Rainbow/Vybrid
• OpenVG Scenario
• Development Ecosystem
• Conclusion
Trend of Graphics in Vehicle
The Connected Vehicle
Infotainment + Communication + Security

- **Consumer electronics trends** are dictating features in the car
- Always **connected, applications driven, advanced graphics**
- **Infotainment systems** becoming battleground for Auto differentiation
- As more connected systems get introduced into the vehicle, the need for **security is critical**
  - Increasing external communication features (Bluetooth, TPMS, Ethernet, Wi-Fi, etc).
  - Future interface for vehicle-to-vehicle and vehicle-to-infrastructure.
Affordable Solutions for Emerging Markets

- **100M vehicles** annually forecasted before 2020, on top of motorcycle & e-bike growth

- 80% of quantity growth after 2015 happening in **emerging markets**

- **Safety** and **emissions** reduction are key for a sustainable development

Source: IHS Automotive, February 2014
More, More, More for Less, Less, Less

**More** performance, **more** embedded memory, **more** safety for **less** cost, **less** power and **less** development effort

### More

- Electronic complexity
- ECUs per car (50+)
- MCUs per car (100+)
- In-car Wi-Fi® (7.2Mbps and 3.7Bpcs by 2017) iSuppli

### Less Reuse

- Other markets have less critical applications
- Some automotive specific challenges

Source: Strategy Analytics and Freescale analysis
Today’s Car

• Complex computerized control
  – Millions of lines of code, from multiple vendors
  – Dozens of distinct ECUs, from multiple vendors
• Shared internal networking (e.g., CAN, FlexRay)
  – Increasing external communications features
    ▪ Telematics, Bluetooth, TPMS, RDS, XM radio, GPS, keyless start/entry, USB ports, Wi-Fi, etc.
Tomorrow’s car -> much more of everything

• The Infrastructure
  - The Intelligent Transportation System (ITS)
    ▪ V2V/ACAS, V2I, traffic control, autonomous driving, real-time data fusion, pervasive sensing
Driver Information Systems Roadmap

Reflects Lead Customer Timing

### Applications

**High-end**
- Best Graphics, Performance & Integration

**Mid-range**
- Optimized Graphics, High Integration
- Cost Effective Performance

**Low-end**
- Lowest System Cost, High Integration

---

**Color Graphics**

### High-end

- **I.MX53**
  - Jan 12
  - ARM Cortex-Dual A7/M4, VSPA, 2D-ACE, GC355

### Mid-range

- **MPC5645S** (Rainbow)
  - May 12
  - ARM Cortex-Dual A7/M4, VSPA, 2D-ACE, GC355

- **MPC5606S** (Spectrum)
  - Jan 13
  - ARM Cortex-Dual as/M4, 2D-ACE, GC355

### Low-end

- **S12XHY** (Sea Wasp)
  - Aug 12
  - ARM Cortex-M0+, 2xCANPhy

- **S12HY** (Jellyfish)
  - Nov 12
  - ARM Cortex-M0+, 2xCANPhy

- **S08LG32**

---

**Proposals**

- **S08 Core**
- **S12 Core**
- **S12z Core** (MagniV)

**Planning**

- **Production**

**Execution**

- **2012**
  - **I.MX6 DUAL**
    - Jun 12
    - ARM Cortex-Dual A7/M4, VSPA, 2D-ACE, GC355
  - **I.MX6 SOLO**
    - May 12
    - ARM Cortex-M4/M0+, 2D-ACE, 2D-GPU, GC255

- **2013**
  - **SVF5/3R** (Faraday/Vybrid)
    - Jan 13
    - ARM Cortex-Dual A5/M4, 2D-ACE, GC355
  - **MACCxx** (Corona)
    - Q2 ’16
    - ARM Cortex-M0+, 2xCANPhy, MagniV

- **2014**
  - **MACHxx** (Halo)
    - Jun 14
    - ARM Cortex-M4/M0+, 2D-ACE, 2D-GPU, GC355

- **2015**
  - **MACCxx** (Corona)
    - Q2 ’15
    - Pin Compatible QFP’s

- **2016**
  - **LumenNG512**
    - Q1 ’16
    - ARM Cortex-M0+, 2xCANPhy, MagniV

---

**2017**
MPC5645S: Rainbow 2M

**General Characteristics:**
- PPC e200z4d Dual Issue core, 5 stage pipeline, 4k I-Cache
- 16 entry Memory Management Unit
- 2M FLASH with ECC
- 64k SRAM with ECC
- 16 channel DMA
- Memory Protection Unit (16 regions)
- QuadSPI Serial Flash Interface
- Voltage Regulator with external ballast transistor
- Real Time Counter + 32kHz crystal oscillator
- Watchdog, Periodic Interrupt Timer, System Timer
- 4-16MHz XOSC
- Frequency Modulated PLL (x2)
- Nexus 3+ / JTAG

**Graphics Features:**
- 2D Graphics Accelerator: AMD z160 OpenVG
- 1M Graphics SRAM
- Display Control Unit: 4 planes / 16 layers
- Display Control Unit –Lite: 2 planes / 4 layers
- DDR DRAM interface (324BGA only)
- Video input Unit (VIU)
- RLE Decoder

**General Characteristics:**
- Up to 120MHz operation
- Low power modes
- -40 to +105°C, 3.0V to 5.5V
- 176LQFP, 208LQFP, 324BGA package options

**Peripherals and Communications:**
- 6 Stepper Motor Drivers with Stall Detection
- Sound Generator Module
- 3xCAN, 3xDSPI, 4xI2C, 4xLIN
- 32 channel eMIOS (PWM+Timer)
- 20 channel, 10bit ADC
## Key Functional Characteristics:
- Cortex-A5 “value” processor for best MIPs/mW
- On-chip SRAM, 2D-ACE, Quad-SPI and RTOS result in low system cost (no DRAM)
- Flexible memory solution configurable based on application needs (1.5MB SRAM or 1MB SRAM + 512K L2)
- DDR3 and OpenVG support for performance critical applications
- Synchronous Audio Interface (SAI) supporting independent I2S, TDM, AC97 and Codec/DSP interfaces
- Enhanced Serial Audio Interface (ESAI) with I2S and AC97 modes

## Key Electrical Characteristics:
- A5 at up to 400MHz, and DDR3-800
- -40 to +85C (ambient)
- 3.0V to 3.6V supply (3.3V nominal)

## Package:
- 144/176 LQFP; 364MAPBGA

## Initial Samples:
- Ready

## Enablement:
- Production Software including CODECs, Stacks, RTOS
- UI development Tools for 2D-ACE
- Radio Reference Design – HW and SW
ARM Core Architecture:
- Cortex M4 vehicle processor
- Cortex A5 application processor
- Cortex M0+ I/O processor

4MByte ECC flash
- 2x 512KB ECC SRAM
- 1.3MB non-ECC SRAM

Supports 2 x WVGA displays:
- OpenVG 1.1 GPU
- 2 x 2D-ACE display interfaces
  - DigitalRGB, RSDS, LVDS i/f
- Hardware HUD warping engine
- Digital camera input

Extensive connectivity:
- Ethernet AVB, MLB50, CAN-FD

I/O Processor (Cortex M0+)
- Supports autonomous operation
- Stepper Motor Drivers
- Peripheral control and Low power operation

Security (CSE2)
- Meets SHE specification
- Meets GM’s Global B Cybersecurity requirements

Functional Safety
- Built in support for ASIL-B

Software Support
- AutoSAR
- GC355
- I/O Stepper motor driver

BGA and QFP package options:
- 176/208LQFP + 516MAPBGA
- -40 to +105°C $T_A$
# MPC56xxS vs MACHxx vs Vybird

<table>
<thead>
<tr>
<th>Feature</th>
<th>MPC56xxS</th>
<th>MACHxx</th>
<th>Vybird</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Target Use-Case</strong></td>
<td>Spectrum - 1 x WQVGA</td>
<td>MACCxx– 1 x WVGA</td>
<td>Up to 2 x WVGA</td>
</tr>
<tr>
<td></td>
<td>Rainbow – 2 x WQVGA</td>
<td>MACHxx- 2 x WVGA</td>
<td></td>
</tr>
<tr>
<td><strong>Core(s)</strong></td>
<td>Spectrum - e200z0h</td>
<td>Corona - ARM CM4</td>
<td>ARM CM4/CA5</td>
</tr>
<tr>
<td></td>
<td>Rainbow - e200z4d</td>
<td>Halo - ARM CM4/CA5</td>
<td></td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>Censorship only</td>
<td>CSE2 (encrypted protection)</td>
<td>CAAM</td>
</tr>
<tr>
<td><strong>Safety</strong></td>
<td>Limited features</td>
<td>ASIL-B</td>
<td>Limited Features</td>
</tr>
<tr>
<td><strong>Flash</strong></td>
<td>Spectrum-1M, Rainbow-2M</td>
<td>Corona-1-2M, Halo-2-4M</td>
<td>None – supports XIP QuadSPI Flash</td>
</tr>
<tr>
<td><strong>Graphics SRAM</strong></td>
<td>Spectrum-160kB, Rainbow-1MB</td>
<td>Corona-256kB</td>
<td>Up to 1.5MB on-chip SRAM in total</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Halo-1.3MB (+1MB ECC RAM)</td>
<td>(1MB no ECC, 512k ECC)</td>
</tr>
<tr>
<td><strong>Stepper Motors</strong></td>
<td>SMD/Stall Detect</td>
<td>'Intelligent’ SMD/SSD</td>
<td>None</td>
</tr>
<tr>
<td><strong>Head-Up Display</strong></td>
<td>None</td>
<td>Hardware warping engine on Halo</td>
<td>GC355 can support warping</td>
</tr>
<tr>
<td><strong>GPU</strong></td>
<td>Spectrum - None</td>
<td>Corona/Halo - 2D-GPU</td>
<td>GC355 OpenVG</td>
</tr>
<tr>
<td></td>
<td>Rainbow - Z160-OpenVG</td>
<td>Halo - GC355 OpenVG</td>
<td></td>
</tr>
<tr>
<td><strong>High Speed Serial</strong></td>
<td>None</td>
<td>MLB50, ENET-AVB, CAN-FD</td>
<td>MLB50, ENETx2, USB-HS x2</td>
</tr>
<tr>
<td><strong>GRAM Expandability</strong></td>
<td>Spectrum – None</td>
<td>Corona – SDR</td>
<td>LPDDR2 / DDR3</td>
</tr>
<tr>
<td></td>
<td>Rainbow – SDR/DDR2</td>
<td>Halo – (SDR)/DDR2</td>
<td></td>
</tr>
<tr>
<td><strong>Enablement</strong></td>
<td>EVB, Basic Compiler tools, “Lab Bench” Demos</td>
<td>EVBs, Complete High-Performance Demos, Optimized Graphics Tools</td>
<td>EVBs, Complete High-Performance Demos, Optimized Graphics Tools</td>
</tr>
</tbody>
</table>
MPC5645S working scenario

- 6 Stepper Motor Drivers with patented stall detection
- OpenVG1.1 GPU (Z160) natively renders:
  - Lines, curves, polygons, textures, etc.
  - Perspective transformation
  - True-type fonts
- DCU on MPC5645S can drive up to 800x480 24-bit LCD
- Optional 2nd Display (DCU-lite)
- Optional SD / DDR-DRAM
Vybrid working scenario

Vybrid F

- ARM Cortex-A5 400MHz
- 32K/32K Caches
- ASRC
- OpenVG GPU
- 1.5MByte SRAM
- 16-bit DRAM Interface
- Dual DDR Quad-SPI Interface
- 2D-ACE
- GPIO
- RTC
- Camera Interface with resizing
- 4-ch ADC
- USB OTG
- UART
- Rearview Camera input
- Optional Touch Screen
- Color LCD Up to WVGA 24bpp

Radio Chipset

- CAN PHY
- MOST

Power Management Unit

- IPS
- SPI

SD/MMC

- CAN
- MLB-50
- SD/MMC
- I2C
- I2C
- UART

I2S

- 4-ch ADC
- iPod Auth.

CD drive

- 2xSD Card or Managed NAND

Optional Touch Screen

- USB

Bluetooth

NXP

External Use | 14

Freescale
MACCxx/MACHxx working scenario

Common ARM Core Architecture:
- Cortex A5 application processor for high resolution graphics capability*
- Cortex M4 vehicle processor common platform for network communications
- Cortex M0+ I/O processor for stepper motor control, peripheral control, and low power performance

Common GPU Platform:
- Common 2D-ACE provides low memory footprint with advanced graphics capability
- Less requirement for external memory
- Scalable vector graphics & font support
- Vivante GC355 & GC255

External Components:
- 1x WVGA supported without external RAM
- 2x WVGA supported with external RAM
- High performance DDR Quad SPI interface**
- 16/32 Bit DDR2/SDR**

* Product dependent (Halo supports A5)
** Package dependent

Additional Use-cases possible with different package & memory options
What is Vector Graphic

Vector Graphic is size-independent!
Vector Graphic

- Vector graphics are drawn and stored as mathematical vector formulae.
- Each vector and fill is assigned color value, instead of assigning color to each separate pixel.
- A black circle can be represented as:
  \[ x = r \cos \theta \]
  \[ y = r \sin \theta \]
  or:
  \[ x^2 + y^2 = r^2 \]
  - With color value 0000 for black

- Benefits
  - Infinitely zoomable
  - Independent of screen resolution
  - Saves data memory
Vector Graphic

Spline with controls
Where is Vector Graphic used today?

<table>
<thead>
<tr>
<th>Fonts</th>
<th>Drawings</th>
<th>Page Layout</th>
<th>Animations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formats</td>
<td>Postscript</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reader</th>
<th>Design Apps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applications</td>
<td>Inkscape</td>
</tr>
<tr>
<td></td>
<td>- Illustrator</td>
</tr>
</tbody>
</table>
OpenVG Feature Overview

• Design Philosophy
  - Expands the OpenGL programming model to 2D vector graphics
  - Provides a low-level hardware acceleration abstraction layer
  - Uses OpenGL-style syntax where possible
  - Allows flexibility in the way acceleration can be provided
  - Enables hardware vendors to use their own preferred internal representations

• The VGU Utility Library
  - Higher-level Geometric Primitives
  - Image Warping
OpenVG Highlights

**Path** (Open path; Closed path)

The path is the basis for all vector objects. A path is made up of one or more line segments connected by two or more anchor points. Paths can be made from a combination of straight lines and curves, each of which may be made up of many connecting points. Paths can be open or closed. An open path is one with unconnected end points, while a closed path is one whose start and end points meet.

**Fill**

A fill can be applied to any area within a path. Fills can be single blocks of colour, gradients, patterns or images (raster or vector).

**Bézier curve** (Curve)

A curved segment of a path is known as a Bézier curve (after French mathematician Pierre Bézier). Bézier curves are defined by mathematical equations - essentially, the coordinates of a curve can be calculated and drawn by knowing the position of two end points and two control points.
OpenVG Rendering Pipeline

- Path Definition
- Stroking
  Line width, joins & caps, dashing, etc.
- Transformation
  2 x 3 (paths) and 3 x 3 (images) transformations
- Rasterization
- Clipping & Masking
  - Scissor rectangles
- Paint Generation
  Flat color, gradient, or pattern paint
- Blending
  Multiple blend modes
- Dithering
- Image Filters
Path Definition

- MOVE_TO, LINE_TO, QUAD_TO, CUBIC_TO, CLOSE_PATH
  - Elliptical Arcs
  - Absolute / Relative Coordinates
  - Smooth Curves
  - Path Interpolation
  - Path queries:
    - Bounding boxes
    - Point along path
    - Tangent along path
  - Non-Zero and Even-Odd fill rules
Stroking takes a path and defines an outline around it based on:

- Line width
- End cap style (butt, round or square)
- Line join style (bevel, round or miter)
- Miter limit (to convert long miters to bevels)
- Dash array and offset

Dash array = { 10, 20, 30, 40} / Dash Phase = 35
Stroking – Dash Array

Dash array = \{ 10, 20, 30, 40 \} / Dash Phase = 35

Dash Pattern:

Resulting Line:
Transformation

- Paths use 2 x 3 affine transformations
- Images use 3 x 3 perspective transformations
- Transformation functions are similar to OpenGL:
  - vgLoadIdentity
  - vgLoadMatrix
  - vgGetMatrix
  - vgMultMatrix
  - vgScale
  - vgRotate
  - vgTranslate

- NOTE: If you want to transform a VG path / object into a perspective, you must do that before sending the points to OpenVG

\[
\begin{bmatrix}
1.080 & 0.101 & 0 \\
0.209 & 0.691 & 0 \\
1.28 \times 10^3 & -1.19 \times 10^3 & 1
\end{bmatrix}
\]
• The goal of rasterization is to determine a filtered alpha value for each pixel, based on the geometry around that pixel
• Filters may be up to 3 pixels in diameter
• OpenVG handles rasterizing arbitrary shapes
Clipping

- Pixels outside a rectangular viewport are not drawn.
- Only pixels inside a set of scissor rectangles are drawn.
- Clip-rects help to cut down the amount of pixels rendered in cases that have only a portion of the screen being rendered. This is essential for GUIs and can help with both maximizing performance and minimizing power.
Masking

- In addition to clipping, a per-pixel mask may be applied
- The mask has an alpha value at each pixel that is multiplied by the alpha from the rendering stage
- May be used to “cut out” an area, create a transition between areas
- Mask values may be modified using image data
  - Fill, Clear, Set, Add, Subtract, Intersect

Generating a Mask is a ‘draw’ action and relatively expensive. Masks should be generated infrequently to minimize performance impact.
Paint Generation

- Paint is generated pixel-by-pixel and applied on top of geometry
- The alpha from the previous stage (rendering + masking) is used to determine how much paint to apply
- Separate paint objects for stroking, filling
- Paint is applied through an affine transformation
- Four types of paint are supported:
  - Flat color paint
  - Linear Gradient paint: Points \((x_1, y_1)\) and \((x_2, y_2)\), color ramp
  - Radial Gradient paint: center \((x, y)\), focus \((x, y)\), radius, color ramp
  - Pattern paint based on an image, tiling mode
Blending

• Combine masked alpha from path with paint alpha
• Blend the result onto the drawing surface
• Blending is a function of:
  – The paint (R, G, B) color
  – The masked alpha value (path alpha mask alpha paint alpha)
  – The destination (R, G, B) color
  – The destination alpha value (1 if no stored alpha)
• There are 8 blending functions:
  – Porter-Duff rules (3)
  – Lighten (choose lighter of source and destination), Darken
  – Multiply (black source pixel forces black, white leaves unchanged)
  – “Screen” (white source pixel forces white, black leaves unchanged)
  – Additive (add pixel values, add alpha up to 1)
Images are defined using one of 15 pixel formats
- Linear or non-linear (sRGB) color spaces (8/8/8, 5/6/5 or 5/5/5 bit depths)
- Linear or non-linear grayscale
- Pre-multiplied or non-premultiplied alpha
- 1-bit Black & White (e.g., for Fax applications)
Images may be stored in accelerated memory
Image filters may be applied:
- Color Matrix
- Convolve, Separable Convolve, Gaussian Blur
- Lookup, LookupSingle
- Dither
Images may be drawn in perspective
Image may be used as a stencil to apply paint
OpenVG Programming Model

- OpenVG is designed around a State Machine based client-server model
  - User ‘sets’ and ‘gets’ variables in the machine (enable/disable, bind, etc.)

- OpenVG: handles to paints, paths, gradients, etc., to avoid the re-preprocessing of each frame

- Data types in a function are determined by a i, f, or v postfix
  - Integer, float, or vector

- Execution decoupled
  - Execution is only guaranteed when user blocks (vgFinish)
  - Or flips the drawing surfaces (eglSwapBuffers)
Coordinate System

User Coordinates

Screen Coordinates

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

\[
\begin{bmatrix}
319 & 0 & 0 \\
0 & 239 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]
Graphic Processor, OpenVG1.1

- Full fixed function hardware vector graphics GPU
- Hardware Tessellation
  - Minimum CPU involvement
- 16x FSAA
  - Photorealistic quality
  - No performance degradation
- Multiformat rendering
  - sRGB color transformation
- High quality vector font rendering
- Standard API OpenVG1.1
GC355 Vector Graphics Engine

GC355 VGMark Performance @ 320MHz

<table>
<thead>
<tr>
<th>GC355</th>
<th>VGA resolution 16X AA (Frames/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>320MHz (projected)</td>
</tr>
<tr>
<td>Tiger (rotation)</td>
<td>85</td>
</tr>
<tr>
<td>UI</td>
<td>96</td>
</tr>
<tr>
<td>Navigation</td>
<td>28</td>
</tr>
<tr>
<td>Flash</td>
<td>42</td>
</tr>
</tbody>
</table>

- Independent 2D Vector GPU Use-cases:
  - Instrument cluster: 2D engine accelerates needles at 60fps; 2D-ACE renders the rest of the scene
- Infotainment: UI acceleration
- Native rendering of true-type fonts, with 16x Anti-Aliasing
- Additional graphics acceleration for dual display systems
Vector Graphic Processor: MACCxx

- Vector graphics
  - Lines / Bezier curves
  - Support for vector fonts
- Raster graphics
  - Raster operations (copy, blend, fill)
- Hardware Tessellation
  - Limited CPU involvement
- Arbitrary Rotation
- 16x FSAA
  - Photorealistic quality
- Optimize low footprint driver
  - Also runs on GC355 GPU on MACHxx
## OpenVG Resources

<table>
<thead>
<tr>
<th>Model</th>
<th>GPU Type</th>
<th>Performance</th>
<th>OpenVG Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC5645s</td>
<td>Z160 GFX</td>
<td>166 Mpxl/s, OpenVG 1.1</td>
<td></td>
</tr>
<tr>
<td>Vybrid</td>
<td>GC355 GPU</td>
<td>300 Mpxl/s, OpenVG 1.1</td>
<td></td>
</tr>
<tr>
<td>MACHxx</td>
<td>GC355 GPU</td>
<td>300 Mpxl/s, OpenVG 1.1</td>
<td></td>
</tr>
<tr>
<td>MACCxx</td>
<td>GC255 GPU</td>
<td>300 Mpxl/s, OpenVG 1.1</td>
<td></td>
</tr>
</tbody>
</table>
Development Ecosystem

- CodeWarrior
- iAR
- GCC
- PVG illustrator plug-in
- PVG Converter tool
- Adobe flash
- Photoshop
- Actia
- iKivo
- Tiny UI
Conclusion

OpenVG in Embedded Graphics System

- Elastic
- Efficient
- Easier
Designing with Freescale

Tailored live, hands-on training in a city near you

2014 seminar topics include

• QorIQ product family update
• Kinetis K, L, E, V series MCU product training

freescale.com/DwF