Developing a TFT-based Cluster Solution
AMF-AUT-T0500

Luis Daniel Olea Mendivil
Introduction to Cluster Systems
Market Trends....
....driving a new generation of Instrument Cluster MCUs

Security
- Publicity around “hacking” of today's cars
- OEM’s want to prevent software from being copied

Hi-Res Color Graphics
- WVGA LCD is becoming more economical
- Fully reconfigurable clusters moving from luxury cars to mainstream
- Even small cars will have color TFT

Head-Up Displays
- HUD attach rates set to grow significantly
- Current generation systems use mechanical warping
- Next generation HUDs will require electronic warping in the MCU
Market Trends (continued)....
....driving a new generation of Instrument Cluster MCUs

Safety
- Rear-view camera being legislated in US
- Growth of panoramic camera and vision applications
- Internal cameras for drowsiness detection
- More instrument clusters with camera inputs

Graphics Tools & Software
- OpenVG1.1 becoming widely used in Cluster applications for needle animation, fonts, textures
- OpenGL ES2.0 and beyond for high-end 3D effects
- Sophisticated tools and graphic design capabilities required
# Instrument Cluster Market Segments

<table>
<thead>
<tr>
<th>Motocycle/Basic Cluster</th>
<th>Low-line Cluster</th>
<th>Mid-/High-line Cluster</th>
<th>Premium Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>• One or two gauges</td>
<td>• Two to Four Gauges</td>
<td>• Up to six gauges</td>
<td>• Few or no gauges</td>
</tr>
<tr>
<td>• Basic segment LCD only</td>
<td>• Large segment LCD, or</td>
<td>• Medium resolution color LCD</td>
<td>• Very high-resolution color LCD</td>
</tr>
<tr>
<td>• Lowest component count and system cost</td>
<td>• Low resolution dot-matrix LCD</td>
<td>• Video input from rear-view camera or from infotainment system</td>
<td>• Dual video inputs</td>
</tr>
<tr>
<td></td>
<td>• Low component count and system cost</td>
<td>• Some sophisticated graphics</td>
<td>• Very demanding animation requirements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Strong focus on system cost</td>
<td>• State-of-the-Art Graphics</td>
</tr>
</tbody>
</table>

Motorcycles | Small Cars | Compact Cars | Medium Cars | Large Cars | Luxury & SUV
MPC560xS Solutions for 32-bit Instrument Cluster Applications

- **Family Concept** – MPC560xS is the latest generation of 32-bit Power Architecture® microcontrollers that address color thin-film transistor (TFT) displays in automotive instrument cluster applications. It offers a cost-effective, entry-level instrument cluster solution with the ability to scale designs to fit performance needs.
- **Performance** - Single core MCUs – 64MHz up to 1MB Flash
  - Powerful crossbar switch for parallel access
  - Memory Protection Unit to avoid different task to access the same memory address
- **DCU** – Display Control Unit
  - Direct drive of TFT displays up to WQVGA resolution
- **Advanced Peripherals**
  - CAN, LIN
  - Stepper motor, Stall detect
  - LCD Driver

**Healthy Ecosystem:**

- i.MX6x Quad, Dual or Single Core Options
  - Up to 1.2 GHz
  - Up to 1 MB L2 Cache

**MPC560xS**
- Single Core
  - 256KB-1MB Flash
  - 1 DCU
  - 144/176 LQFP

**MPC564xS**
- Dual issue
  - 2MB Flash
  - 1-2 DCUs
  - 176/208 LQFP, 416 TEPBGA
MPC5606S – Mid-Line Cluster Solution

- Delivers all system components in a single device
  - No external DRAM => low-cost PCB

- Integrated stepper motor drivers
  - Patented stall detect capabilities

- Display Control Unit
  - Dramatically reduces VRAM requirements for large color displays
  - Delivers 60fps animation with very low CPU load

- Ideal for cluster with single color display and segment LCD

- Multiple stepper motors for true hybrid solution
MPC5606S: Single Chip Cluster Solution

General Characteristics:
- PPC e200z0h Core
- 1M FLASH with ECC
- 4x16k EEPROM Emulation block with ECC
- 48k SRAM with ECC
- 16 channel DMA
- Memory Protection Unit (12 regions)
- Voltage Regulator with external ballast transistor
- Real Time Counter + 32kHz crystal oscillator
- Watchdog, Periodic Interrupt Timer, System Timer
- 4-16MHz crystal oscillator
- Frequency Modulated PLL (x2)
- Nexus 2+ / JTAG

Graphics Features:
- 160k Graphics SRAM (No ECC)
- Display Controller Unit – 18/24bit RGB
- Parallel Data Interface
- QuadSPI Serial Flash controller

General Characteristics:
- Up to 64MHz operation
- Low power modes
- -40 to +105C, 3.0V to 5.5V
- 144 LQFP, 176 LQFP package options

Peripherals and Communications:
- 6 Stepper Motor Drivers with Stall Detection
- Sound generation using eMIOS
- 40x4 LCD Segment Driver
- 2xCAN, 2xDSPI, 4xI2C, 2xLIN
- 24 channel eMIOS (PWM+Timer)
- 16 channel, 10bit ADC
Display Control Unit
Display Control Unit

- Internal Flash
- Internal RAM
- External Flash

DMA

DCU

DMA

DMA

Image: Display control unit diagram with connections to Internal Flash, Internal RAM, and External Flash, all connected to DCU via DMA.
DCU Operation

- DCU combines layers of “sprites” to create the final content
  - There are up to 19 different sources of content possible
    - 16 programmable layers that contain source graphics
    - A cursor layer
    - 1 layer as a default color for the background OR
    - 1 layer for an external video input
  - Layers are set in a fixed priority that determines order of pixels selected to blend
DCU Operation

- For each pixel position
  - The DCU fetches a pixel from the topmost layer placed there AND
  - A pixel from the next layer in the priority
  - And pixels from up to two further layers (dependent on user configuration)
- Indexed colors are converted to 24bpp colors for internal processing
- The fetched pixels are then blended to give the display content for that position
  - The blending attributes are determined per layer and the lowest priority pixel’s blending attributes are ignored
- Each resulting pixel can be gamma corrected
- The output format is 8-bits per color (24bpp)
The Layer Paradigm

• A layer is the DCU mechanism by which graphics are displayed on the panel

• Pick the paradigm that works for you:
  - A layer is a graphics-oriented interface to the DCU’s DMA function and contains commands for the blending engine
  - A layer is an object that encapsulates all of the information needed to display and blend a single image on the TFT panel
  - A layer is a set of registers that you program to display an image

• There are 16 layers available
  - Each has a fixed priority 0..15 (0 highest priority)
  - Each can be enabled individually
  - Each is buffered and can be changed at any time
    ▪ The DCU automatically synchronizes changes to the panel refresh cycle

• Not all graphic content is provided by the layers
The Anatomy of a Layer

- The DCU has a set of 7 registers to configure each layer
- The layer registers configure
  - Height & width of layer (pixels)
  - Position on panel (x,y) (pixels)
  - Pointer to the graphic (address)
    - and its encoding
    - and its palette (if needed)
  - Pixel selection
    - By color range
  - Transparency
  - Tile option
A Note on Graphic Encoding

- There are two main types of graphic encoding and one special type
  - RGB where each pixel in the image contains red, green and blue components
  - ARGB where each pixel in the image contains red, green and blue components and an alpha channel
  - Transparency where each pixel in the image contains only an alpha channel
A Note on RGB Encoding

- For RGB images, there are two ways of encoding the image:
  - Direct color where each pixel in memory contains its color components
  - Indexed color where each pixel in memory contains an index to a table that contains the color of the pixel
- Indexed colors use less memory than direct colors:
  - 8bit per pixel (bpp) stores 1 x 24bpp pixel as a single byte
  - There are 4bpp, 2bpp and 1bpp options that use progressively less memory
- Each indexed image references a palette of colors stored in a color look-up table (CLUT)
  - 1bpp CLUTs have two entries, 2bpp have up to 4 entries etc.
- However pixels are encoded, all blending occurs after conversion to 24bpp
DCU Layers & the Pixel-blend Stack

- Where layers overlap each other or the background the pixels may be blended
  - Up to four layers may be blended at each pixel location
- The position of each pixel in the blend stack determines how it is blended
  - Layers below the lowest priority pixel are not visible
  - The blending settings for the lowest priority pixel are ignored
Blending Modes

- There are two aspects to the blending of the pixels
  - Selection of pixels to blend
    - This is done by chroma keying
    - In other words, the DCU changes the way the blend works according to the color of the individual pixel
  - The amount of transparency applied to each selected pixel
- Depending on the settings and the content of the source graphic, there are 12 different blend settings
  - Images with RGB data have 5 options
  - Images with ARGB (alpha content) have 6 options
- When active, the alpha setting applied to the image is defined by an 8-bit value in the layer (TRANS bitfield)
Graphics Without Alpha Included

• The DCU can blend graphics containing only red, green and blue components (RGB)
• Three formats are supported
  - RGB888: 24-bit data containing red, green and blue all at 8 bits per color channel
  - Indexed color 1bpp, 2bpp, 4bpp, 8bpp where all pixels are blended as RGB888
  - RGB565: 16-bit data containing red and blue channels at 5 bits per color channel and green at 6 bits per channel
• RGB565 is converted automatically to RGB888 before blending takes place
• All blending takes place at 24bpp
Defining the Chroma Range

- The range is defined by giving maximum and minimum values for each color component
  - Red, green and blue have minimum and maximum values
- If the color of a pixel falls into the range in each of the components, then it is selected

<table>
<thead>
<tr>
<th>Red</th>
<th>Green</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 – 255</td>
<td>0 – 255</td>
<td>0 – 255</td>
</tr>
<tr>
<td>0 – 0</td>
<td>0 – 255</td>
<td>255 – 255</td>
</tr>
</tbody>
</table>
Graphics with Alpha Included

• The DCU can blend graphics with an embedded alpha channel (ARGB)
• Three formats are supported
  – ARGB8888: 32-bit data containing alpha and RGB channels all at 8-bits per color channel
  – ARGB4444: 16-bit data containing alpha and RGB channels all at 4-bits per color channel
  – ARGB1555: 16-bit data containing alpha channel of 1-bit (on/off) and RGB channels at 5-bits per color channel
• The DCU can blend these layers using the blend options already described
  – The behavior in certain modes is modified
• However an image is encoded, all blending takes place at 32bpp
<table>
<thead>
<tr>
<th>Alpha blending</th>
<th>BB (chroma blend)</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 0</td>
<td>0</td>
<td>No blend</td>
</tr>
<tr>
<td>B 0</td>
<td>1</td>
<td>Removes selected pixels</td>
</tr>
<tr>
<td>C 1</td>
<td>0</td>
<td>Alpha blend all</td>
</tr>
<tr>
<td>D 1</td>
<td>1</td>
<td>Alpha active for selected pixels</td>
</tr>
<tr>
<td>E 2</td>
<td>0</td>
<td>Same as C + TRANS alpha</td>
</tr>
<tr>
<td>F 2</td>
<td>1</td>
<td>Same as D + TRANS alpha</td>
</tr>
</tbody>
</table>

Original ARGB image over white background
Transparency Pre-blending

- The transparency mode graphic contains no color information
  - Only alpha values are stored
  - The front color of the graphic and the rear color of the background are fetched from dedicated foreground and background registers
- The anti-aliased edges are pre-blended by the DCU to give smooth edges between the graphic (front color) and its background (rear color)
Special Image Types

• The DCU supports further operating modes for the layer
  – Tile mode allows the filling of a large area using a small pattern that is repeated or “tiled” across the area
  – The luminance of an area in a graphic can be adjusted for those cases where it is desirable to highlight a graphic element such as an area on a map or a menu selection

• For user-input operations, there is on-screen cursor available separate from the rest of the graphic layers

• The DCU also allows users to specify the default background color that layers can blend with in the absence of other layers and that is visible where no layer is active

• The DCU also can adjust the value of any color on the panel by using its gamma correction transfer function
Creating Textures – Tile Mode

- The DCU provides a memory-efficient method of creating a textured layer by allowing a graphic to be “tiled” within a layer
  - Layer size defines the extent of the layer
  - Tile size defines the extent of the tile
  - The DCU automatically repeats the tile to fill the layer

![Layer + Tile graphic = Layer on panel (before blending)]
Background

• The DCU BGND register contains the RGB888 color used as a background color
• When the background pixel is involved in the blend stack, it is always as the lowest priority pixel
• The background can be replaced optionally with a video input signal from the Parallel Digital Interface (PDI)
Specialized Peripherals for Cluster applications
SSD – Stepper Stall Detect

- In automotive cluster applications the rotor of the stepper motor is fitted with a pointer.

- At startup the position should be zeroed out to maintain an accurate reference.

- As the gauge returns to zero, the SSD module detects the pointer collision with the stopper.
SSD – Stepper Stall Detect

- The polarity of the SSD pins can be switched
- Flexible preescaler
- Offset cancelation minimizes the internal accumulation error
- Drives the movement of the stepper motor in full steps.
- At the same time the motor is moved, the stall detection works to detect the zero
- Complete return to zero function supported for all 6 motors
SMC – Stepper Motor Control

- The Stepper Motor Control (SMC) module is designed to facilitate the simple creation of the waveforms required to drive stepper motors using micro-stepping. The module contains 12 Pulse Wave Modulation (PWM) channels, clocked by an 11-bit counter.

- Each channel is associated with two pins giving a total of 24 pins.

- The module also has the ability to detect a short circuit on any of these pins and can be configured to trigger an interrupt when one occurs. An optional interrupt can also be triggered upon a timer counter overflow.
SMC – Stepper Motor Control

• Integrated high current drivers for stepper motors.
• Up to 6 motors are supported.
• Direct connection from MCU to motors with no external components.
• Advanced PWM functions to create waveforms.
• Alignment, delay, dithering of PWM is supported.

  • Choice of type of step (full step, half step, micro step) is up to the user and is supported by hardware module.
SGL – Sound Generation Logic
SGL – Sound Generation Logic

- Two types of sounds supported: monophonic and polyphonic
- Selects any PWM channel for sound generation
- One PWM is used for monophonic, the anded output of two PWM’s is used for polyphonic.
- Configurable periodicity or continuous sound.
- Duration can be programmed.
- Interrupts can be configured.
LINFLEX Overview

• LIN Protocol Handler
• Master mode
• Slave mode
• LIN message buffer
• Filtering Unit (slave)
• Re-synchronization (slave)
• Enhanced error detection
• Standard UART/SCI mode
NFLEX Overview

Enhanced error detection circuit

- Bit error
  - Detection on all bits transmitted including header, delimiters
- Identifier Parity
- Break Delimiter
- Inconsistent Synch Field
- Framing error
- Checksum Error
  - Classic
  - Enhanced

- Slave response timeout
  - Dedicated timer programmable by application
- LINRX stuck dominant
- Error signalling
  - 9 error sources
  - Each error source can be independently enabled/disabled
 NFLEX Overview
UART mode

- **Mode**
  - Full Duplex
  - 8-bit / 9-bit
  - Even / Odd parity

- **Transmit Buffer**
  - Depth configurable from 1 to 4

- **Receive Buffer**
  - Depth configurable from 1 to 4

- **Error**
  - Parity
  - Overrun

---

**CONTROL STATUS REGISTERS**

**TRANSMIT BUFFER**
4 BYTES

**RECEIVE BUFFER**
4 BYTES

**UART/SCI CORE**

LIN TX  LIN RX
FlexCAN overview

- Full Implementation of the CAN protocol specification, Version 2.0A/B
  - Standard and Extended ID frames and Remote Frames
  - Zero to eight bytes data length
  - Programmable bit rate up to 1Mbit/s
  - content-related addressing
- 64 Message Buffers of zero to eight bytes data length
- Programmable loop-back mode supporting self-test operation
- Individual Rx Mask Registers per Message Buffer
- Powerful Rx FIFO ID filtering, capable of matching incoming IDs against either eight extended,
  - 16 standard, or 32 partial (8 bits) IDs, with individual masking capability
- individual Tx message buffers.Time Stamp based on 16-bit free-running timer
- Hardware cancellation on Tx message buffers
FlexCAN - Modes of Operation

• Normal Mode
  – Module Rx and Tx frames, error handling, module fully operational

• Freeze Mode
  – No Transmission/Reception of frames and synchronicity to the CAN bus lost. Some registers are only accessible in this mode

• Listen-Only Mode
  – Transmit disabled, all error counters are frozen. Only messages acknowledged from other nodes will be received

• Loop-Back Mode
  – Module performs internal loop back. Bit steam output from the transmitter is fed back to the receiver. Used for self-test operation.
Each buffer has its own receive ID mask
- For backwards compatibility, Global mask, Mask 14 & 15 register are used out of reset.
- Buffers 0-7 can be used to implement an 8 frame Rx FIFO
Debugging, Tools
Freescale Example Applications

• MPC5606S
  http://www.youtube.com/watch?v=V-rFl1oglkw

• MPC5645S
  http://www.youtube.com/watch?v=Vkqr6EsqcIM
Freescale MPC5606S Development Tools

• Try all the features in the MPC5606S. This board includes:
  - MPC5606S Microcontroller in a 176 LQFP package
  - On-board JTAG connection
  - 4.3" 480x272 24bit Touchscreen LCD Display
  - MC34845 Backlight controller
  - High Speed QSPI 8MB Flash Memory
  - Loudspeaker with sound amplifier
  - CAN & LIN Interface
Freescale MPC5645S development tool

Try out all the features of the MPC5645S
- 2x connections for LCD TFT panels
- 2x connections for touchscreen panels
- 2x DVI outputs
- Headphone amplifier
- Video in port
- 64 MB of serial flash
- 64 MB of mobile LPDDR
- CAN and LIN
- USB port for debug and communication
XPC560SKIT: Evaluation system supporting Freescale MPC56xxS-Family

- **XPC56xxMB2**
  - 12 V DC power supply input barrel connector
  - Two CAN channels with jumper enables
    - One CAN channel with high-speed transceiver
    - One CAN channel with low-speed fault tolerant and high-speed transceiver (selectable with jumpers)
  - Two LIN channels with jumper enables
    - One channel with footprints only
    - One channel with transceiver
  - One SCI channel with jumper enables
  - Two FlexRay™ channels with jumper enables
    - One channel with transceiver
    - One channel with footprint only
  - Four user push buttons with jumper enables and polarity selection
  - Four user LEDs with jumper enables

- **XPC560SKIT**: Adapter Mini-Module for the XPC56xxMB2 board. Can be used as a stand-alone board by providing external 5-volt power supply input
Cluster HMI Tool Select Partner

- Pre-integrated key technologies from industry-leading partners help the customer get to market faster and leverage the differentiated advantage of Freescale’s technologies – particularly the DCU

- Proven, tested, production-ready code ready for evaluation and integration

- Specifically optimized for Freescale processor performance and memory footprint

- Altia has been publically announced as our Select HMI tool partner (FTF 2012)

- Non-exclusive partnership
Recommended Application notes

- **AN3330**: Introduction to the stepper stall detector module
- **AN4037**: Driving a stepper motor using the MPC56xxS SMC module
- **AN4187**: Configuring and using the DCU2 on the MPC5606S MCU
- **AN4444**: Configuring and using the DCU3 and DCUlite on the MPC5645S MCU
- **AN4719**: MPC5606S Graphical cluster hardware design
- **AN4186**: Using the QuadSPI Module on MPC56xxS
- **AN4435** Sound Generation Logic (SGL) Module
For technical support go to www.freescale.com/support

or send an email to support@freescale.com