Section 8. LCD Display

Freescale Embedded GUI and TWR-LCD

Module Objectives

- This session will introduce you to the...
  - Features of the new TWR-LCD Module
  - Freescale Graphic Library Solution for Embedded MCUs
  - Documentation and Demo Software available.

Module Objectives

- This session will introduce you to the...
  - Features of the new TWR-LCD Module
  - Freescale Graphic Library Solution for Embedded MCUs
  - Documentation and Demo Software available.

Introduction

- Driving graphics LCD panels normally requires a dedicated MPU
- Solutions such as the Coldfire MCF532x, MCF5227x or i.MX devices have been specifically designed for this purpose
- Recently, a number of new LCD displays have been released to the market which have LCD controllers integrated onto the LCD glass
- A simple SPI or 8/16 bit parallel interface can be driven by any microcontroller

Smart vs Dumb LCD Panels & Bus requirements

- Dumb LCD TFT panels are very much like DRAM’s
  - No on-board frame buffer or graphics engine
  - Cells’ need to be refreshed periodically
  - Refreshing uses a lot of bus bandwidth
  - Cheaper but require LCD controller
- Smart LCD TFT panels – Like Static RAM
  - LCD is refreshed from on-board RAM
  - CPU bus bandwidth is only used when data must be changed
  - Requires significantly less CPU bus bandwidth
  - More expensive but does not require LCD controller

Display Spectrum

- Display Technologies
  - Monochrome LCD
  - Segment-based
  - Dot-Matrix/Graphical
  - Color LCD/TFT
  - Graphical
  - Vacuum Fluorescent Display (VFD)
  - Organic Light Emitting Diode Displays (OLED)
  - LED Matrix
- Small-size Displays
- Med- to Large-sized Displays
- Monitors, Terminals, TVs, etc.
- Passive
- Active

LCD Driving Methods

- Traditional LCD Driver
  - Requires a dedicated LCD MPU
- SPI/Parallel LCD Driven
  - Can be driven from any MCU
Peripheral Module: TWR-LCD Graphical LCD Module

**TWR-LCD**

- **Launch Date:** April 26, 2010
- **Features:**
  - 3.2" TFT display
  - SPI and CPU interface
  - Touch screen interface
  - 6-position navigation switch
  - Piezo buzzer
  - Temperature sensor
  - Micro SD card slot
  - Measurement port with current sensing circuitry
  - Standalone mode

**Resale:** TWR-LCD = $99

TWR-LCD Graphical LCD Module Block Diagram

- Truly Semiconductor 3.2" TFT LCD with an analog resistive touch overlay
- The switch will allow user interaction with the TWR-LCD providing a method to indicate Up, Down, Right, Left and Select.

**What's the Freescale Embedded GUI?**

- High level graphic driver for color LCD
- Structure of driver brings complete SW solution for applications with LCD screen
- Touch screen capability
- Organization done by screen with objects
- We can find the_alias references to D4D
  - (Driver for Display) in function software

**Block scheme of FGUI implementation**

- User application – layer of user application code
- Application programmable interface - interface layer between user application and FGUI
- D4D high level – layer that manages all screens and objects (redraws, input keys, touch screen events etc.)
- LCD low level – manage communication with LCD and provides some basic function (Draw line, bitmap, circle etc.)
- HW – Freescale MCU board + LCD

**Features of Freescale Embedded GUI**

- Multiple platform support. Tested on HCS08/MCF51– Flexis series 8-bit, 32-bit MCUs and MCF52xx
- Objects:
  - Button
  - Check Box / User handled Radio Button
  - Gauge
  - Icon
  - Label
  - Menu
  - Picture
  - Slider
  - Graph
Freescale Embedded GUI Screen

- The basic item of FGUI structure is SCREEN
- The screen representing the real screen showed on LCD
- The screen definition contains:
  - list of all used objects on screen
  - function bodies:
    - OnInit
    - OnActivate
    - OnDeactivate
    - OnMain
    - OnObjectMsg

Freescale Embedded GUI Button Object

- Button object is targeted for control of user functions
- Check Box object is targeted for control and set up of user application

Freescale Embedded GUI Gauge Object

- Gauge object is targeted as a graphic information of numerical values
- Icon object is targeted as a graphic information

Freescale Embedded GUI Label object

- Label object is targeted as a text information
- Menu object is targeted as a control item

Freescale Embedded GUI Picture object

- Picture object is used only to draw bitmap on screen

Freescale Embedded GUI Slider object

- Slider object is targeted as a graphic information of numerical values and for setting this numerical values
- The D4D_GRAPH object is prepared to create a simple graph that provides a lot of features
Freescale Embedded GUI + TWR-LCD Demo Labs "Codewarrior for MCU V10"

Freescale Embedded GUI example application Screen Shot

Freescale Embedded GUI example application Window

Agenda
- Introduction
- TWR-LCD System
- Freescale Embedded GUI (D4D)
- Demo Projects
- Third Party Software Solutions
- Demonstration Labs with Codewarrior for MCU V10
- Additional Resources
- Review

Demo Labs
- TWR-LCD projects for CodeWarrior for MCU v10:
  - Demo_MCP5101_32K
  - Demo_MCP5101_64K
  - Demo_MCP6001_64K
  - Network_MCP5101_32K
  - HelloWorld_MCP5101_32K
  - Network_MCP6001_64K

Demo Labs
- Lab1 : Demo Application programming using Bootloader
- Lab2 : HelloWorld Application
Lab2 : HelloWorld Application

► Select LCD Project
► Right click -> Open Project

Lab2 : HelloWorld Application

► Expand Source Tree
► Open main.c (Double click)

Lab2 : HelloWorld Application

Init HelloWorld Screen
Library Polling Routine

Lab2 : HelloWorld Application

► Open screen_hello_world.c

Lab2 : HelloWorld Application

We have to declare the components in our screen (write code or use copy&Paste.txt file)
Lab2 : HelloWorld Application

Write code under "INCLUDE YOUR DECLARATION CODE HERE"

- Label Declaration:
  D4D_DECLARE_STD_LABEL_AUTOSIZE(lbl_HelloWorld, "Hello World", 70, 10, FONT_8x14_BIG)

- Text Button Declaration:
  D4D_DECLARE_TXT_BUTTON_AUTOSIZE(btn_Example, "Change Icon", 70, 100, FONT_8x14_BIG, OnClick_BtnExample)

- Bitmaps:
  D4D_DECLARE_BMP(bmpIconGraph, icon_graph24x24, NULL)
  D4D_DECLARE_BMP(bmpIconBulb, icon_bulb24x24, NULL)
  D4D_DECLARE_BMP(bmpIconPaint, icon_paint24x24, NULL)
  D4D_DECLARE_BMP(bmpIconTchScr, icon_Touchscreen24x24, NULL)

- Icon with 4 bitmaps Declaration:
  D4D_DECLARE_STD_ICON_BEGIN_AUTOSIZE(icon_Example, "Icon", 120, 150, FONT_8x14, NULL)
  D4D_DECLARE_ICON_BMP(&bmpIconGraph)
  D4D_DECLARE_ICON_BMP(&bmpIconBulb)
  D4D_DECLARE_ICON_BMP(&bmpIconPaint)
  D4D_DECLARE_ICON_BMP(&bmpIconTchScr)
  D4D_DECLARE_STD_ICON_END()

- Screen with 3 objects Declaration:
  D4D_DECLARE_STD_SCREEN_BEGIN(screen_HelloWorld, ScreenHelloWorld_)
  D4D_DECLARE_SCREEN_OBJECT(lbl_HelloWorld)
  D4D_DECLARE_SCREEN_OBJECT(btn_Example)
  D4D_DECLARE_SCREEN_OBJECT(icon_Example)
  D4D_DECLARE_STD_SCREEN_END()

Lab2 : HelloWorld Application

We have to add the callback function code for Button Touch/Click event under "INCLUDE YOUR BUTTON CALLBACK CODE HERE"

```c
D4D_UNUSED(pThis);
D4D_IconChangeIndex(&icon_Example, 1);

static void OnClick_BtnExample(D4D_OBJECT* pThis)
{
  D4D_UNUSED(pThis);
  D4D_IconChangeIndex(&icon_Example, 1);
}
```

Lab2 : HelloWorld Application

- Connect USB cable to TWR-LCD

Lab2 : HelloWorld Application

- Find the S19 application output file:
  C:\CW10\workspace\icf\MCF51J1M128_Internal_Flash\HelloWorld_MCF51J1M_SPI_BL.elf.S19

© Freescale Semiconductor 2005
Lab2 : HelloWorld Application

- Drag&Drop S19 output file to bootloader Drive

- The bootloader will load the file and flash the application

Bootloader Drive

Lab2 : HelloWorld Application

- Additionally the MSD (Mass storage device will show 'SUCCESS.TXT')

- Now you can reset the board (pressing JMRST), and this will launch your HelloWorld Application

Lab2 : HelloWorld Application

- Touch the Button to change the icon image.

Lab2 : HelloWorld Application

TWR-LCD Documentation

- TWR-LCD User Manual
- TWR-LCD Lab Guide
- Freescale_Embedded_GUI_User_Manual
- TWR-LCD-RevA-SCH : TWR-LCD Schematic
- TWR_LCD_QSG : TWR-LCD Quick Star Guide
- SSD1289 : Display Controller Datasheet
- TRULY-TFT2N0369-E : Truly Display Datasheet

Lab1 : Demo Application programming using Bootloader

- Connect USB cable to TWR-LCD
Lab1: Demo Application programming using Bootloader

- The TWR-LCD features a bootloader to facilitate the loading of applications without the need for an external debugger.
- Bootloader will enumerate the TWR-LCD JM128 as a MSD (Mass Storage Device).
- To enter the bootloader hold the 'BTLD' button while momentarily pressing the 'JMRST' button and finally releasing ‘BTLD’.
- In bootloader mode, you will hear a beep from the sounder and the screen will write a welcome message.

Precompiled Applications:

- Now you can drag&drop / copy S19 (Motorola S-Records) precompiled applications files to the bootloader.

Lab1: Demo Applications Programming

- Drag&Drop S19 Demo file to bootloader

- The bootloader will load the file and flash the application

Boostrap Drive

- Additionally the MSD (Mass storage device will show ‘SUCCESS.TXT’)

- Now you can reset the board (pressing JMRST), and this will launch your new application.

Lab1: Demo Applications Programming

- Flash and Explore the Applications:

- JM128_BL_EGUI_SPI.S19

- JM128_BL_ECUI_SPI.S19

If an application is already loaded, then you need to reset the board (press the JMRST button) while holding down the BTLD button.