Logitech G-series LCD SDK

LCD UI Framework Overview

Overview

The LCDUI classes are a set of C++ foundation classes to allow easier programming of the LCD. The classes reside on top of the Core API layer represented in the header file lglcd.h. They provide such services as:

- Plug and Play of G-series LCD devices
- Sending data to the LCD device
- Reading button data on the LCD device
- Creation of commonly used controls
- Support for monochrome and color displays

Core Classes

CLCDConnection

This is the top-level class used to connect to the LCD Manager. It handles device arrival and removal. Applications should periodically call the Update() function. Use the "HasMonochromeDevice()" or "HasColorDevice()" to determine which type of device support is available. You will need to create one, and only one, instance of this class.

CLCDPage

This class is known as the "Page" class. A Page contains LCD UI objects such as a Text object or Icon object. If your applet support both monochrome and color, you will want to derive two sets of pages. For monochrome pages, use CLCDConnection::MonoOutput to get the monochrome device output and then CLCDOutput::AddPage to add the page. For color pages, use CLCDConnection::ColorOutput to get the color device output and then CLCDOutput::AddPage to add the page.

CLCDGfxBase

CLCDGfxMono

CLCDGfxColor

These classes handle details related to the generation of device-specific bitmaps. In the case of monochrome devices, CLCDGfxMono is used. In the case of color devices, CLCDGfxColor is used.

Typically, you will never have to deal with this class.

CLCDOutput

This class handles device specific LCD output details. At most there will be two instantiations of this class: one for monochrome LCD device output, and one for color LCD device output. The CLCDConnection class will automatically handle creation and destruction of this class, as devices are attached and removed.

Use the CLCDConnection::MonoOutput and CLCDConnection::ColorOutput functions to access the relevant LCD output object.

Common Control Classes

CLCDBitmap

Class to draw a bitmap (BMP) file onto the LCD.

CLCDAnimatedBitmap

Class to draw a tiled sequence of bitmaps onto the LCD resulting in an animation effect.

CLCDIcon

Class to draw an icon (ICO) file onto the LCD.

CLCDProgressBar

Class to draw/control a progress bar onto the LCD.

CLCDColorProgressBar

Color LCD only. Class to draw/control a plain, color progress bar onto the LCD.

CLCDSkinnedProgressBar

Color LCD only. Class to draw/control a skinned progress bar onto the color LCD.

CLCDText

Generic class to draw standard text onto the LCD. You can set the font, point-size, color, etc.

CLCDScrollingText

Class to draw scrolling text onto the LCD. Scrolling text is scrolled when the text does not fit onto the LCD. It is scrolled to the end of the text and then reset back to the beginning of the text.

CLCDStreamingText

Class to draw streaming text onto the LCD. Unlike scrolling text, streaming text will not reset to the beginning of the text but will rather start displaying the text over again in the same animation. So the text appears to flow infinitely. This is useful for displaying a long string that needs to be repeated such as a song title and artist, or an RSS text stream.

CLCDColorText

Color LCD only. Class that merges all of the above text classes into a single class.

Sample Usage

- Create 1 instance of **CLCDConnection**.

- Create your pages, each derived from **CLCDPage.** Override the **CLCDPage::Initialize()** function and add your controls. Refer to the sample on how to do this.

- If you wish to deal with buttons, have your page class override the **OnLCDButtonDown()** and **OnLCDButtonUp()** methods.

- Use **CLCDOutput::AddPage()** to add the page and **CLCDOutput::ShowPage()** to show the page. Use **CLCDConnection::ColorOutput()** and **CLCDConnection::MonoOutput()** to get the respective device output class you wish to use.

- In your application's initialization procedure invoke **CLCDConnection::Initialize()**

- In your application timer procedure, periodically invoke the **CLCDConnection::Update()**.

- In your application's termination procedure, invoke **CLCDConnection::Shutdown()**.