



Application Note

Writing a Video driver on Windows CE 6.0

© Digi International Inc. 2007. All Rights Reserved.

The Digi logo is a registered trademark of Digi International, Inc.

All other trademarks mentioned in this document are the property of their respective owners.

Information in this document is subject to change without notice and does not represent a commitment on the part of Digi International.

Digi provides this document “as is,” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Digi may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time.

This product could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes may be incorporated in new editions of the publication.

Digi International Inc.

11001 Bren Road East

Minnetonka, MN 55343 (USA)

☎ +1 877 912-3444 or +1 952 912-3444

<http://www.digi.com>

Table of contents

History	4
References	4
Acronyms	4
4 Overview	5
4.1 Introduction.....	5
5 LCD Driver Modification	7
5.1 LCD Driver structure	7
5.2 Creating new header file	8
5.3 Modify Driver Sources.....	8
5.4 Updating Catalog File.....	9
5.5 Building the Driver.....	10
5.6 Building the OS	10

History

Date	Version	Author	Description
03/27/08	0.1	ME	First version

References

Number	Name	Description
1		
2		

Acronyms

	Description
OS	Operating System
HFP	Horizontal Front Porch
HBP	Horizontal Back Porch
VFP	Vertical Front Porch
VBP	Vertical Back Porch
PB	Platform Builder
BSP	Board Support Package
OEM	Original Equipment Manufacturer
TFT	Thin file transistor
LCD	Liquid Crystal Display

4 Overview

4.1 Introduction

This document will describe what is necessary to adapt the video driver of Windows CE 6 for a new display on the Digi ConnectCore 9 module family. All the examples used in the application will make use of the Digi ConnectCore 9P module.

The current Windows CE 6 display driver is made for 16bpp color TFTs. The purpose of the application note is to show how the driver needs to be changed to support a new LCD available under Windows CE 6.

Under Windows CE, only one LCD can be supported during run time. That means that during build time the supported display need to be selected. This is done over the catalog of the Platform Builder in the *Catalog View*.

Theory

The main configuration that is necessary to adjust a color TFT LCD are the three main clocks Pixel clock, Horizontal sync and Vertical sync. The LCD controller must be configured to achieve those timings to function correctly.

The timing consist basically of the following parts

Horizontal clock calculation:

$$X\text{-Size} + HFP + HBP = HSYNC$$

Vertical clock calculation:

$$Y\text{-Size} + VFP + VBP = VSYNC$$

Some manufacturers are including not only the timings also the necessary configuration for the back and front porch values that can be added directly to the controller registers.

$$\text{Blanking period} = VFP + VBP \text{ or } HFP + HBP$$

Calculate pixel clock frequency

$$PCD = (\text{CPU Clk}/2/\text{Pixel Clk}/\text{Divisor})-2$$

5 LCD Driver Modification

5.1 LCD Driver structure

First, it is important to get an overview of the source code structure of the LCD driver under Windows CE 6.

The source code of the LCD driver for the ConnectCore 9P is located at:

```
%(_WINCEROOT)\platform\common\src\soc\NS9XXX_DIGI_V1\Display\NS9XXX\Xlcd
```

The following figure shows the complete structure of the LCD driver sources.

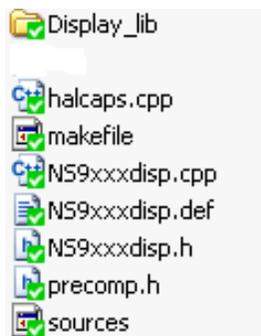


Figure 5.1: Display driver structure

In the folder *Display_lib* are several header files located. Each of the header file represents a supported color LCD for Windows CE. To support a new color LCD a new header file need to be created and added to the *display.h* file with a condition to add that header file.

```
#ifndef DISPLAY_H
#define DISPLAY_H

#if defined (CRT_VGA)
#include "CRT_VGA.h"

#elif defined (LQ57Q3DC2)
#include "LQ57Q3DC2.h"
```

```
#elif defined (LQ64V3DG01)
#include "LQ64V3DG01.h"

#else
# error "no platform specific display type selected"
#endif

#endif /* DISPLAY_H */
```

5.2 *Creating new header file*

After creating or copying and renaming an existing header file, several macros in that file have to be updated to meet the new LCD specification. Some LCD manufacturers are including already the correct values for the HFB, HBP, VFP and VBP. If that information is not directly available, it can be extracted by the timing configuration of the display.

Once the correct settings are done the new header, file need to add to the *display.h* file located in the same directory.

5.3 *Modify Driver Sources*

After modifying and including the new header file with the configuration of the display, it might be necessary to make some modifications to the video driver itself. The display driver configuration is made for 16 bit color LCDs. If the new display also support 16 bit color deeps no additional modifications are necessary.

In case the display has a lower or even higher color deeps or only mono support these differences have to added to the video driver. This modification is out of scope of that documentation for more information on the necessary implementation please refers to the Platform Builder online help.

5.4 Updating Catalog File

To be able to build the new video driver and to improve the usability a Platform Builder component need to be added to the current catalog. The catalog of the BSP can be found at:

```
%(_TARGETROOT)\catalog
```

The catalog file is a XML file that can be edited with the Visual Studio 2005. Following an example of such a modification in the catalog file

```
..
<BspItemId>Item:Digi:MY_LCD:BSPXX</BspItemId>
..
<Item Id="Item:Digi:MY_LCD:BSPXX">
  <Title>TFT My LCD (X,Y)</Title>
  <Description>NS9XXX LCD Driver for My LCD
(X,Y)</Description>
  <Type>BspSpecific</Type>
  <SysgenVariable>SYSGEN_DISPLAY</SysgenVariable>
  <Variable>Variable used in display.h</Variable>
  <SupportedCpuId>Cpu:ARMV4I</SupportedCpuId>
  <Module>ns9xxx_disp.dll</Module>
  <ChooseOneGroup>>true</ChooseOneGroup>
  <Location>Device Drivers\Display</Location>
  <SourceCode>

<Title>$(_WINCEROOT)\PLATFORM\COMMON\SRC\SOC\NS9XXX_DIGI_V1\
DISPLAY</Title>

<Path>$(_WINCEROOT)\PLATFORM\COMMON\SRC\SOC\NS9XXX_DIGI_V1\
DISPLAY</Path>
  </SourceCode>
</Item>
```

After adding the new display to the catalog, it might ne necessary to refresh the catalog view to make the new component visible.

5.5 Building the Driver

Now that the new component is added to the catalog, select the new supported display. When the component is select it will automatically set the variable specified by the node `<Variable></Variable>` by the catalog.

The build of the video driver can be done from the *Solution Explorer* tab moving to the directory

```
%( _WINCEROOT )\platform\common\src\soc\NS9XXX_DIGI_V1\Display
```

and selecting *Build* from the menu.

5.6 Building the OS

After successfully building the video driver, the new driver needs to be included into the Windows CE image. Depending on your Platform Builder build configuration the PB might build directly the Windows CE image.

In case that is not happening, the command *Make Run Time Image* from the *Build* menu needs to be selected. This will build a new Windows CE image that after building can be deployed to the target.