

Display Controller Demo Quickstart Guide

REV A

Publication Date: 2012/10/10
XMOS © 2012, All Rights Reserved.



Table of Contents

1	sw_display_controller demo : Quick Start Guide	3
1.1	Hardware Setup	3
1.2	Import and Build the Application	4
1.3	Run the Application	4
1.4	Next Steps	4

1 sw_display_controller demo : Quick Start Guide

IN THIS CHAPTER

- ▶ Hardware Setup
 - ▶ Import and Build the Application
 - ▶ Run the Application
 - ▶ Next Steps
-

- ▶ XA-SK-SCR480 Slice Card,
- ▶ XA-SK-SDRAM Slice Card,
- ▶ module_sdram,
- ▶ module_lcd,
- ▶ module_display_controller,
- ▶ module_slicekit_support,

together to create a double buffered LCD controller. This application showcases some of the key software features and serves as an example on how to use an LCD without the real-time constraint of having to update the LCD line buffer.

1.1 Hardware Setup

The XP-SKC-L2 Slicekit Core board has four slots with edge connectors: SQUARE, CIRCLE, TRIANGLE and STAR.

To setup up the system:

1. Connect XA-SK-SDRAM Slice Card to the XP-SKC-L2 Slicekit Core board using the connector marked with the TRIANGLE.
2. Connect XA-SK-SCR480 Slice Card to the XP-SKC-L2 Slicekit Core board using the connector marked with the STAR.
3. Connect the XTAG Adapter to Slicekit Core board, and connect XTAG-2 to the adapter.
4. Connect the XTAG-2 to host PC. Note that the USB cable is not provided with the Slicekit starter kit.
5. Set the XMOS LINK on the to OFF on the XTAG Adapter(XA-SK-XTAG2).
6. Ensure the jumper on the XA-SK-SCR480 is bridged if the back light is required.
7. Switch on the power supply to the Slicekit Core board.

1.2 Import and Build the Application

1. Open xTIMEcomposer and check that it is operating in online mode. Open the edit perspective (Window->Open Perspective->XMOS Edit).
2. Locate the 'Display Controller Demo' item in the xSOFTip pane on the bottom left of the window and drag it into the Project Explorer window in the xTIMEcomposer. This will also cause the modules on which this application depends to be imported as well.
3. Click on the app_display_controller_demo item in the Explorer pane then click on the build icon (hammer) in xTIMEcomposer. Check the console window to verify that the application has built successfully.

For help in using xTIMEcomposer, try the xTIMEcomposer tutorial. [FIXME add link.](#)

Note that the Developer Column in the xTIMEcomposer on the right hand side of your screen provides information on the xSOFTip components you are using. Select the module_display_controller component in the Project Explorer, and you will see its description together with API documentation. Having done this, click the *back* icon until you return to this quickstart guide within the Developer Column.

1.3 Run the Application

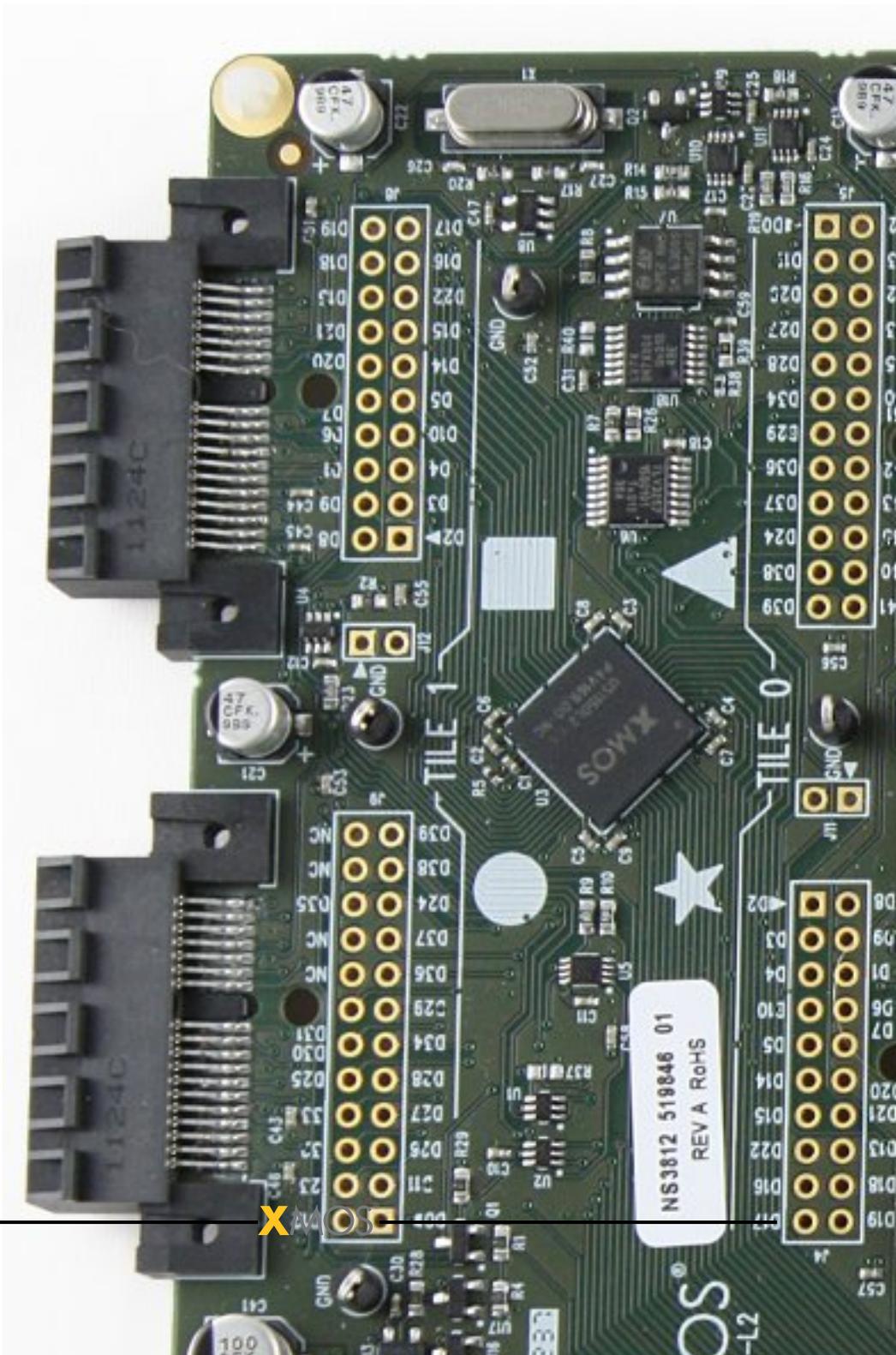
Now that the application has been compiled, the next step is to run it on the SliceKit Core Board using the tools to load the application over JTAG (via the XTAG2 and Xtag Adaptor card) into the xCORE multicore microcontroller.

1. Click on the Run icon (the white arrow in the green circle).
2. Wait until the images have loaded over the XTAG connector from the host.
3. There should be a series of 6 images the transition from one to another.

1.4 Next Steps

1. Trying changing the files that are loaded from the host. To do this, produce an image of 480 by 272 pixels, save it in tga format uncompressed in "top left" format ("bottom left" will also work but the image will have to be upside-down). Save the file(s) into the app_display_controller_demo directory within your workspace.
2. Each transition has a frame count that depicts the speed of the transition, try adjusting them and observe the results.
3. Try writing an exciting transition effect. To do this begin with the template shown below, and refer to the Display Controller API documentation at [sec_display_controller_api](#):

```
static void transition_exciting_impl(chanend server, unsigned
next_image_fb,
    unsigned image_from, unsigned image_to, unsigned line) { //insert code
    here
} unsigned transition_exciting(chanend server, unsigned frame_buf[2],
    unsigned from, unsigned to, unsigned frames, unsigned cur_fb_index)
{ unsigned next_fb_index; for (unsigned frame = 0; frame < frames;
frame++) {
    next_fb_index = (cur_fb_index + 1) & 1; for (unsigned line = 0; line <
    LCD_HEIGHT; line++)
        transition_exciting_impl(server, frame_buf[next_fb_index], from,
        to, line);
    frame_buffer_commit(server, frame_buf[next_fb_index]);
    cur_fb_index = next_fb_index;
} return cur_fb_index;
}
```





Copyright © 2012, All Rights Reserved.

Xmos Ltd. is the owner or licensee of this design, code, or Information (collectively, the "Information") and is providing it to you "AS IS" with no warranty of any kind, express or implied and shall have no liability in relation to its use. Xmos Ltd. makes no representation that the Information, or any particular implementation thereof, is or will be free from any claims of infringement and again, shall have no liability in relation to any such claims.

XMOS and the XMOS logo are registered trademarks of Xmos Ltd. in the United Kingdom and other countries, and may not be used without written permission. All other trademarks are property of their respective owners. Where those designations appear in this book, and XMOS was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.