

Display Spectrum Quickstart Guide

IN THIS DOCUMENT

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

In this demonstration we use the following hardware and software:

- ▶ XP-SKC-U16 sliceKIT
- ▶ XA-SK-SCR480 Slice Card,
- ▶ XA-SK-SDRAM Slice Card,
- ▶ XA-SK-MIXED SIGNAL Slice Card,
- ▶ module_level_meter,
- ▶ module_fft_simple,
- ▶ module_display_controller,
- ▶ module_sdram,
- ▶ module_lcd,

together to create a level-meter kind of spectral display on an LCD for a simulated signal. This application demonstrates real-time rendering and display of spectrum by taking short-time fourier transform.

1 Hardware Setup

The XP-SKC-U16 sliceKIT Core board has four slots with edge connectors: DIAMOND, SQUARE, A and U.

To setup up the system:

1. Connect XA-SK-SDRAM Slice Card to the XP-SKC-U16 sliceKIT Core board using the connector marked with SQUARE.
2. Connect XA-SK-SCR480 Slice Card with LCD to the XP-SKC-U16 sliceKIT Core board using the connector marked with DIAMOND.
3. Connect the xTAG-2 to sliceKIT Core board.

4. Connect the xTAG-2 to host PC. Note that the USB cable is not provided with the sliceKIT starter kit.
5. Set the xCONNECT LINK to OFF on the sliceKIT Core board.
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.



Figure 1:
Hardware
Setup for
Display
Spectrum
Demo

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->XMOSEdit).
2. Locate the Display Spectrum 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. Include `#define LCD_USE_32_BIT_DATA_PORT 1` in `lcd.xc` located in `module_lcd`.
4. Click on the `app_display_spectrum` 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.
5. There will be quite a number of warnings that `bidirectional buffered port not supported`. These can be safely ignored for this component.

For help in using xTIMEcomposer, try the xTIMEcomposer tutorial, which you can find by selecting Help->Tutorials from the xTIMEcomposer menu.

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.

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 xTAG-2) into the xCORE multicore microcontroller.

1. Select `app_display_spectrum` project from the Project Explorer.
2. Click on the **Run** icon (the white arrow in the green circle).
3. At the **Select Device** dialog select **XMOS xTAG-2 connect to L1[0..1]** and click **OK**.
4. The spectra of segments of mixed signal of two simulated chirp waveforms are displayed on LCD.

4 Next Steps

1. `FFT_POINTS` gives the number of signal samples taken for FFT computation. This decides the number of level meter bands which can be varied by varying `FFT_POINTS`. `FFT_SINE` is also varied accordingly.
2. The colors of the level-meter display of spectrum can be changed in `level_meter_conf.h`.
3. Different choices of simulated waveforms are available in `main()` function. Uncomment to try them.



Copyright © 2013, 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.
