# Freescale MQX RTOS Example Guide

DCU4 example

This document explains the DCU4 example, what to expect from the example and a brief introduction to the API used.

The example

DCU4 example can run on svf522revb_a5 and twrvf65gs10_a5 board with DDR support. Since it requires big memory block to hold display buffer, IntRam version will fail in running. The example will show rectangles on screen for several stages, every stage will hold on about 5 seconds. Firstly a red rectangle appears at the up-left corner, this is the highest layer numbered 0. After that layers with number 1 to 5 are created, and you can find red (ARGB8888), orange (RGB565), yellow (RGB888), green (ARGB1555), blue (ARGB4444) and purple (UYVY) rectangles from top to bottom. After that, three rectangles disappear and only red, green and blue layers are shown, this means chroma keying blending engine works. Later on all six rectangles are on the screen with transparency, which means the alpha blending engine take effects. Thereafter only the bottom rectangle exists. Finally all rectangles are gone.
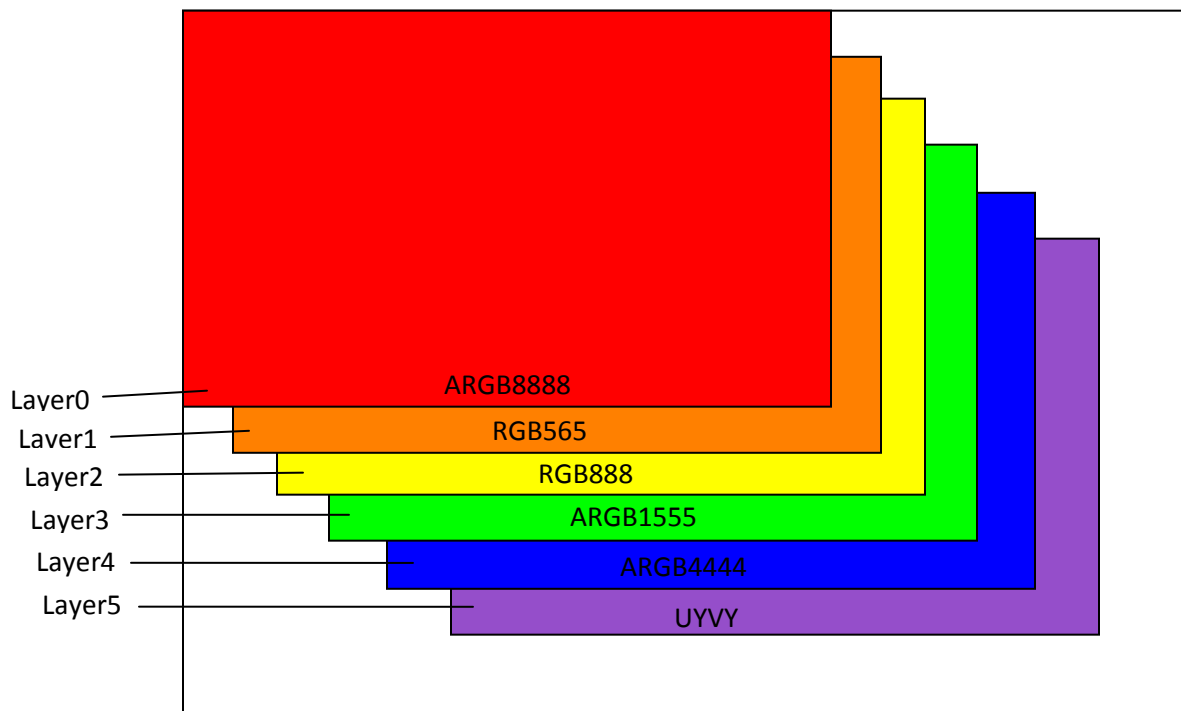
Running the example

The BSPCFG_ENABLE_DCU4 macro must be set to non-zero in the user_config.h file prior to compilation of MQX libraries and the example itself.

To run the example the corresponding IDE, compiler, debugger and a terminal program are needed.

Explaining the example

This example demonstrates the usage of DCU4 driver to create simple graphic patterns – rectangles – in memory and to display them over the LCD hardware. The example employs only one task called main_task and it does following jobs.

- Open the connection to the DCU4 instance 0 peripheral module to configure the DCU4 block in software.
- Install the event handler for the interrupt happens when a new frame begins. The event handler which is function *vsync_handler()* updates the number of new frames generated and displays the statistics of number of frames generated per second or frame rate in frames/second.
- Call function *setup_layer()* to set up the property of layer 0 where the width, height, vertical and horizontal coordinates as well as the encoding format and color of the layer are initialized. Delay for 5 seconds to allow user to be able to see the rectangle.
- Display the timing configuration of the DCU4 module which is associated with the resolution of the LCD panel.
- In the case that the display can support multiple resolutions, the timing configuration could be changed to adapt the new resolution (1024 x 768) of the display.
- Testing multiple layers without blending method.
    - Set up the property of layer 1, 2, 3, 4 and 5 with layer width and height equal to layer 0's width and height but the encoding formats and color of these layers are different.
    - Set up the blending mechanism in DCU4 module by calling function *ioctl()* with input parameter IO_IOCTL_DCU4_LAYER_SET_BLEND, with blend type DCU_BLEND_NO_ALPHA. The chroma keying blending and alpha blending are disabled.
    - Verify the process of configuring blending above and in case everything is as expected the application is delayed for 5 seconds to allow user to be able see 6 rectangles on LCD screen.
- Testing multiple layers with chroma keying blending method.
    - Set up the blending mechanism in DCU4 module by calling function *ioctl()* with input parameter IO_IOCTL_DCU4_LAYER_SET_BLEND. The blending is set for all pixels of six layers from layer 0 to layer 5. Alpha value is 0.
    - Verify the process of configuring blending above and in case everything is as expected the application is delayed for 5 seconds to allow user to be able see the rectangles on LCD screen.
- Testing multiple layers with alpha blending method.
    - The layers' properties are similar to the previous case but this time the blending method is changed to alpha blending with the transparency level set to a half of the maximum opaque level in the transparency scale (128/255).

- Verify the process of configuring blending above and in case everything is as expected the application is delayed for 5 seconds to allow user to be able see the rectangles on LCD screen.
- Disable layers 0 to 4 and delay for 5 seconds to allow user to see only one rectangle on screen, unregister all the event handlers installed previously for layers and free memory of layers' data before closing the connection to DCU4 peripheral module.
- Terminate the application.