APPLICATION NOTE

Renesas Synergy™ Project

Simple PMOD Display Example for DK-S124 Getting Started Guide

R12AN0044EU0100
Rev.1.00
Jul 7, 2016

Introduction

This application note describes how to install, build, and run an example PMOD display application on a Renesas Synergy development board using the Renesas e^2 studio ISDE and Renesas Synergy Software Package (SSP).

Goals and Objectives

The goal of this application note is to help you install, build, and run the example application.

The example application shows how to create a simple graphics-enabled application using the Okaya PMOD LCD and Drivers within the SSP. When the application is running, you can control the displayed content using two on-board pushbuttons and a potentiometer.

The example can be adapted to add visual effects to more complex applications.

Prerequisites

The reader of this application note is assumed to have some experience with the Renesas e^2 studio ISDE and SSP. For example, before you perform the procedure in this application note, you should follow the procedure in your board's Quick Start Guide to build and run the Blinky project. By doing so, you will become familiar with e^2 studio and the SSP and ensure that the debug connection to your board is functioning properly.

Required Resources

The example application targets Renesas Synergy S124 devices. To build and run the application, you will need:

- A Renesas Synergy DK-S124 board
- An Okaya PMOD Display (bundled with every Renesas Synergy DK-S124 kit)
- A PC running Microsoft® Windows® 7 with the following Renesas software installed:
  - e^2 studio ISDE 5.0.0.043 or newer
  - Synergy Software Package (SSP) 1.1.0

You can download the required Renesas software from the Renesas Synergy Gallery: (https://synergygallery.renesas.com).

Time Required

You can install, build, and run the example application in under 30 minutes. The high-level steps involved are:

1. Configure the DK-S124 board for the simple PMOD™ display application example.
2. Import, build, and debug the project.
3. Control and observe the display application program output.

Contents

1. Configure the DK-S124 for Simple PMOD Display Example

Configure the DK-S124 board to run the example PMOD display application:

1. Connect the JLink-OB on J14 of the DK-S124 to the PC using a micro USB cable (Figure 1).
2. Verify that J17 (BOOT MODE) is not installed across the pins (Figure 2).
3. Connect Okaya PMOD™ display to J7 (Figure 3).
4. Set the voltage for the PMOD™ interface to 3.3v using J8

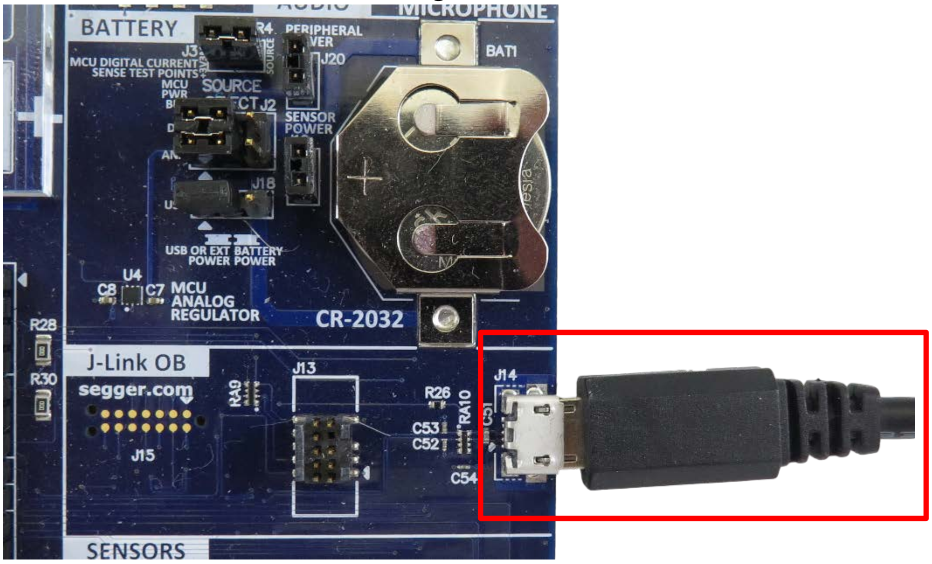

Figure 1. Power setup (top) and J-Link OB connection (bottom)

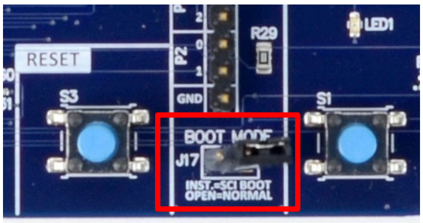

Figure 2. Boot mode configuration

Note: For the purposes of this example application, ensure J17 is not installed across the pins.

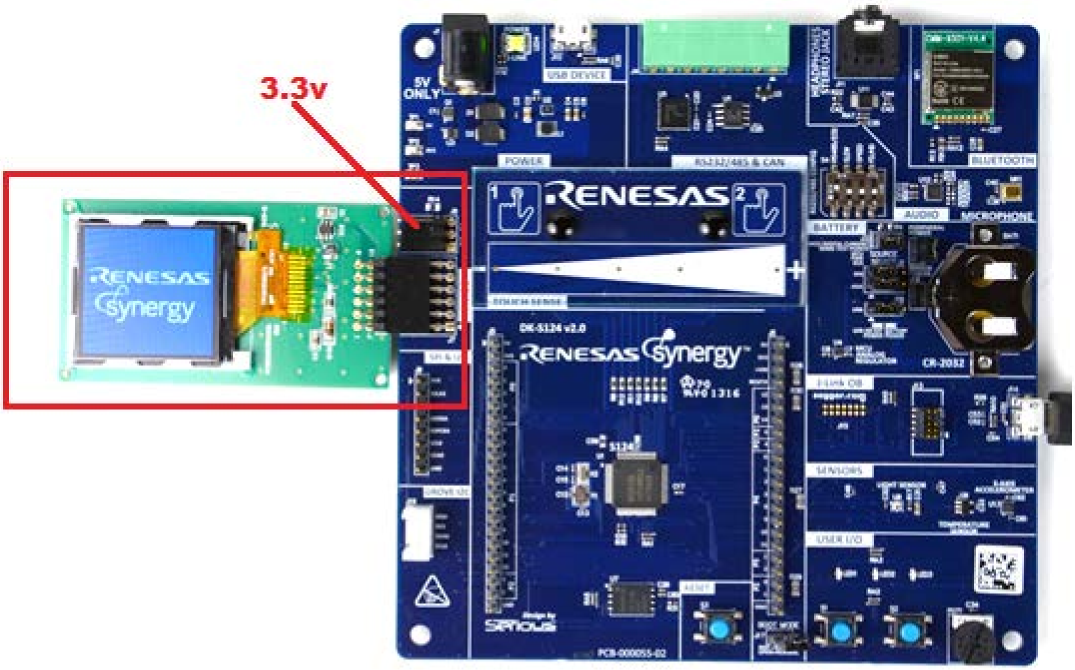

Figure 3. Okaya PMOD display connected to J7 PMOD™ socket

2. Importing and Building the Project

Follow the procedure in the *Synergy Project Import Guide* (r11an0023eu0110_synergy_ssp.pdf) to import the project into the e[2] studio ISDE, and to build and debug the project. For the debug configuration, select S1_DK_pmod_display_example Debug (under Renesas GDB Hardware Debugging).

3. Observing the PMOD Display Application Output

When you first press the key F8 or the Resume button to start the application, the application stops at `main()`. Press F8 or the resume button again to run the code.

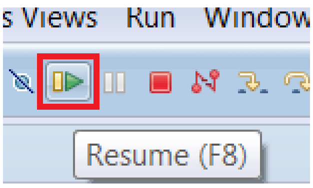

Figure 4. Resume Button

Once application is resumed, you will see a welcome screen (Figure 5, left). After few seconds (or when S1 is pressed), the program will proceed to the screen with instructions (Figure 5, middle and right). From this moment, you can use S1 pushbutton to go to the next step (provided application has finished drawing the current screen) and S2 pushbutton to change screen orientation in the clockwise direction. Changing screen orientation will restart the sequence of drawings, but the Welcome and Instruction screen will be skipped.

Figure 5. Splash, Welcome and Instruction screens

While the application is running, you can use potentiometer to adjust backlight intensity. If you cannot see any difference in the strength of the backlight, please verify that the PMOD™ is outputting 3.3V on Vcc pin (set on J8 header). To end the debug session, press Ctrl + F2 or the Stop button.

Figure 6. Stop button

4. How the Application Works

The simple PMOD display example application is controlled by pushbutton switches S1 and S2 (found in the lower right corner of the DK-S124 board) and potentiometer POT1 (located to the right of the two buttons).

Both pushbuttons are connected to hardware interrupt pins, which are controlled by External IRQ Framework. The framework allows you to control thread execution using hardware IRQs. In most cases, the application would use `sf_irq.p_api->wait` with `TX_WAIT_FOREVER` timeout argument, to block processing in the thread until given interrupt request is received. In the Simple Display example, the LCD thread scans through both button interrupts with timeout value of zero and, if neither button is pressed, waits for 20 milliseconds before repeating the process (IRQs are buffered by the framework before next `sf_irq.p_api->wait` call). This provides enough processor ticks to lower-priority threads while still providing a responsive interface. Figure 7 shows a simplified view of the processing flow.

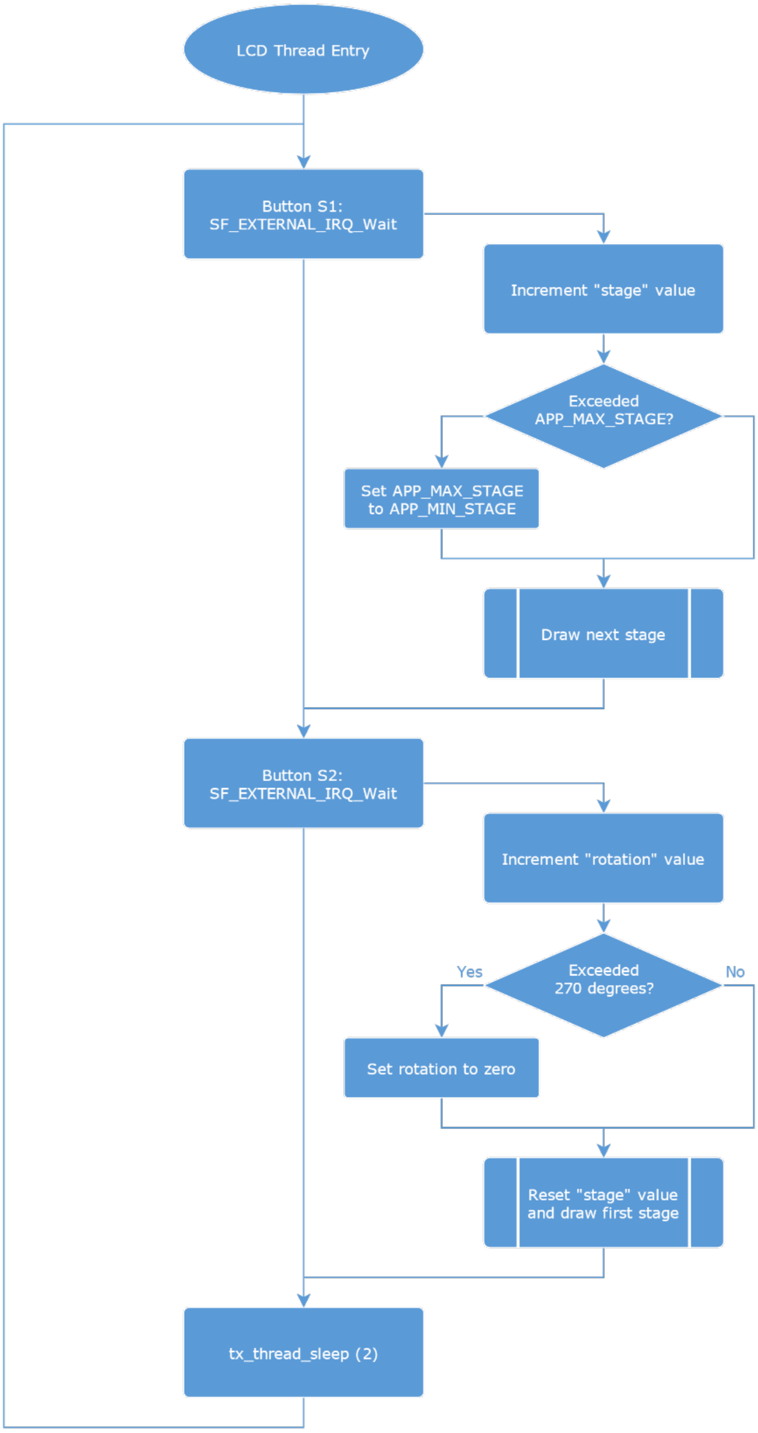

Figure 7. Simplified View of Example Display Application Processing

A detailed view of External IRQ Framework processing can be found in the Synergy Software Package User's Manual, chapter 4.1.8.

The simple display example uses an Okaya PMOD LCD driver running on SCI peripheral in Simple SPI mode. All driver files are contained inside `src/lcd_setup` folder and can easily be copied into another application to enable use of PMOD display. The `SPILCD_Init` function accepts two arguments: pointer to the SPI instance and initial value for screen rotation. The SPI module needs to be configured for the right channel and bitrate (9 and 2.5mbit/s, respectively), however the callback function input in the configurator is discarded, as the display driver overrides it automatically with its own implementation, allowing for better data flow control when outputting data through the PMOD. The Okaya PMOD LCD driver works with Simple SPI on SCI as well as RSPI, however care must be taken to remove the transfer drivers for the RSPI interface to allow 8-bit data width.

The driver provides following features:

- `SPILCD_Init` opens the SPI channel and initializes the PMOD display.
- `SPILCD_Rotate` changes screen orientation to one of the enumerations defined in `spilcd_rotation_t`.
- `SPILCD_Clear` fills the screen with provided color in RGB565 format.
- `SPILCD_DrawPixel` draws a single pixel at a given position and with set color.
- `SPILCD_DrawRectangle` draws a rectangle with the position, dimensions and color provided.
- `SPILCD_DrawBitmap` renders a bitmap image stored at address provided. The bitmap must be in RGB565 color format and should be read top-to-bottom with increasing address.
- `SPILCD_DrawText` renders a null-terminated string provided as an argument. Text can be formatted by calling following commands prior to `DrawText` function call:
  - `SPILCD_SetTextColor` sets color values to be used for text and its background. If both values are equal, the background is rendered transparent (i.e. last screen contents are still visible from underneath the text). Text with transparent background is rendered using slower drawing method and background color should be provided, when possible.
  - `SPILCD_SetTextCursor` sets top-left boundary and a starting point for the `DrawText` function. In case when text is wrapped onto the next line, it will stay vertically aligned to the starting point and continue directly below it.
  - `SPILCD_SetTextMargin` sets rectangular boundary for the text box. This function should be used to wrap text before it reaches the end of a screen. If current text cursor position is outside of the boundary, cursor is automatically set to the top-left corner of the rectangle defined by this function.

The simple PMOD display application uses Okaya LCD driver to draw 15 different screens, each implementing different features of the driver.

In addition to the displayed content, control of backlight intensity is provided through ADC peripheral running inside control thread. This task is launched before LCD thread is initialized, to ensure all setup is complete before displaying any data on the screen. ADC peripheral running in continuous scan mode takes periodic readings of the channel 7 (connected to the on-board potentiometer POT1) potential difference. Despite module being configured to read data with 12-bit resolution, software manipulates the data so that readings become an integer value contained within 0-100 range. While the level of precision required for smooth backlight control is still sufficient, reducing effective resolution acts as a jitter-filter, that prevent unnecessary PWM duty cycle updates when the ADC readings are rapidly oscillating. If the reading (post-processing) is different to the previous one, duty cycle of the GPT (connected to the PMOD display backlight-enable pin) is updated using R_GPT_DutyCycleSet. ADC sampling frequency is limited to 33Hz by suspending the thread for 30ms using tx_thread_sleep (3) after each reading.

Figure 8 illustrating the processing inside the control thread can be found on the next page.

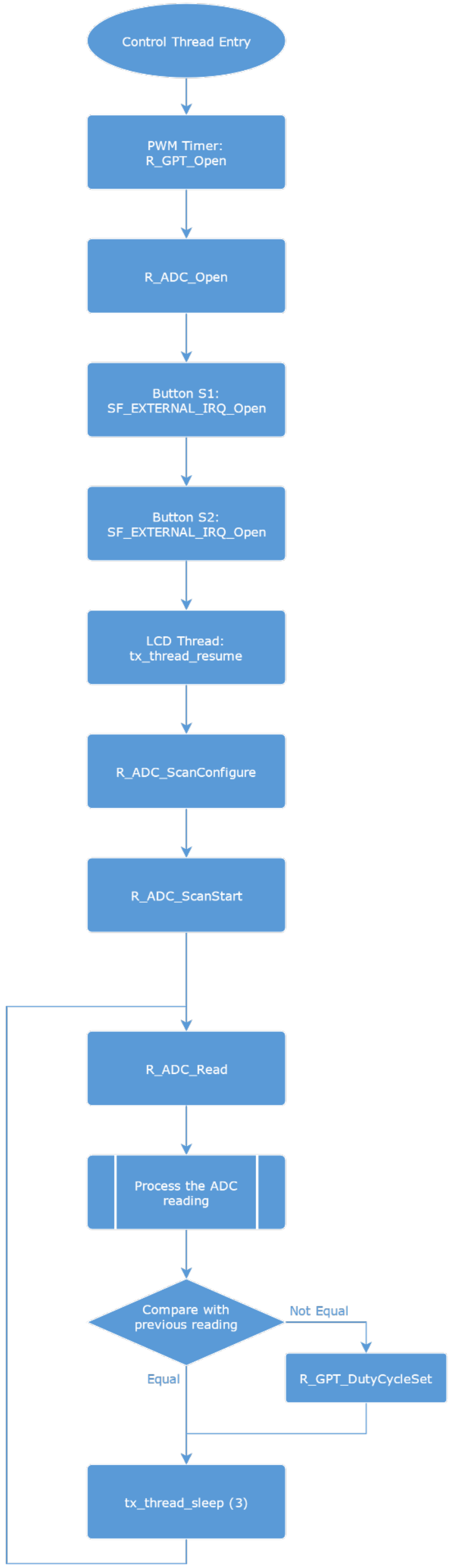

Figure 8. Simplified View of Control Thread Processing

5. Next Steps

After you run the example application, you can learn more about how the application works and the API calls involved by examining the application source code.

You can also download additional Synergy example applications from the following URL:
http://am.renesas.com/request?SCREEN_ID=ViewGRSDownloadSearch&EXECUTE_ACTION=search&LAYER_ID=188821

Website and Support

Support https://synergygallery.renesas.com/support

Technical Contact Details

America: https://renesas.zendesk.com/anonymous_requests/new

Europe: http://www.renesas.eu/support/index.jsp

Japan: http://japan.renesas.com/contact/index.jsp

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jul 7, 2016	-	Initial Version

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins
   Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.
   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.
   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.
5. Differences between Products
   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.
   — The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HALII Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 5.0