

Simple PMOD Display Application Example for DK-S128 R12AN0071EU0101 Rev.1.01 Mar 16, 2018

Introduction

This application note describes a Pmod[™]-based display application on the DK-S128 kit using the Renesas e² studio ISDE and Renesas Synergy Software Package (SSP).

Also included in this document are how-to steps on importing the project, so that you can recreate the application with the e² studio Integrated Solutions Development Environment (ISDE/IAR SSC).

Minimum PC Requirements

- Microsoft[®] Windows[®] 7 or 10
- Intel[®] Core[™] family processor running at 2.0 GHz or higher (or equivalent processor)
- 8 GB memory
- 2 GB of free space hard disk or SSD
- USB 2.0.

Required Resources

The example application targets Renesas Synergy S128 MCU group. To build and run the application, you will need:

- Renesas DK-S128 kit
- e² studio ISDE v6.2.0 or greater or IAR EW for Synergy v8.21.1 or later
- Synergy Software Package (SSP) 1.4.0 or greater, Synergy Standalone configurator (SSC) 6.2.0 or later
- Segger J-link[®] USB driver
- Micro USB cables
- An Okaya PmodTM-based LED display (included with the DK-S128 kit)

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

Prerequisites and Intended Audience

As the user of this application note, you are 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 OOB application 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.



Contents

1.	Application Overview	3
1.1	Application Software Architecture	3
1.1.1	1 LCD Thread	3
1.1.2	2 Control Thread	4
2.	Procedure to Create Simple Pmod Display Example Project	6
3.	Running the Pre-existing DK-S128 Simple Pmod [™] Display Demonstration	12
3.1	Powering up the Board	12
3.2	Importing, Building, and Running the Project	12
3.3	Verifying the Demonstration	12
4.	Next Steps	14
5.	Limitations and Assumptions	14



1. Application Overview

This document shows how to create a simple graphics-enabled application using the Okaya Pmod[™]-based LCD display and drivers within the SSP. When the application is running, you can control the content displayed using two on-board pushbuttons and a potentiometer. The example can be adapted to add visual effects to more complex applications.

1.1 Application Software Architecture

The main components of this application are:

- Control Thread
- LCD Thread.



Figure 1 DK-S128 Simple Pmod[™] Display Application Architecture

1.1.1 LCD Thread

The Simple Pmod[™] display application is controlled by pushbutton switches S1 and S2 (found in the lower right corner of the DK-S128 board) and a potentiometer POT (located to the right of the two buttons).

Both pushbuttons are connected to hardware interrupt pins, which are controlled by the external IRQ framework. The framework allows you to control thread execution using hardware IRQs. In most cases, the application uses sf_irq. p_api->wait with the TX_WAIT_FOREVER timeout argument, to block processing in the thread until the given interrupt request is received. In the simple display example, however, the lcd thread scans through both button interrupts with a timeout value of zero, and, if neither button is pressed, it waits for 20 ms before repeating the process (IRQs are buffered by the framework before the next sf_irq. p_api->wait call). This gives enough processor ticks to lower-priority threads while still providing a responsive interface.

The simple display example uses an Okaya Pmod[™] LCD driver running on the SCI peripheral in Simple SPI mode. All driver files are contained inside the src/lcd_setup folder and can be easily copied into another application to enable the use of the PMOD display. The SPILCD_Init function accepts two arguments: a pointer to the SPI instance and an initial value for screen rotation. The SPI module needs to be configured for the right channel and bit rate (9 and

2.5 Mbps, 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 PmodTM LCD driver works with Simple SPI on SCI as well as RSPI, though you must be careful to remove the transfer drivers for the RSPI interface to allow 8-bit data width.



Figure 2 Simplified view of LCD Thread processing

1.1.2 Control Thread

The Simple PmodTM 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 an ADC peripheral running inside the control thread. This task is launched before the LCD thread is initialized, to ensure that all setup is complete before displaying any data on the screen. The ADC peripheral running in continuous scan mode takes periodic readings of the channel 7 potential difference, through a connection to the on-board potentiometer, POT1. Despite the module being configured to read data with 12-bit resolution, software manipulates the data so that readings become an integer value contained within the 0-100 range. While the level of precision required for smooth backlight control is still sufficient,



reducing the effective resolution acts as a jitter-filter to prevent unnecessary PWM duty cycle updates when the ADC readings are rapidly oscillating. If the reading (post-processing) is different from the previous one, the 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 33 Hz by suspending the thread for 30 ms using tx_thread_sleep (3) after each reading.



Figure 3 Control Thread Processing



2. Procedure to Create Simple Pmod Display Example Project

The following steps are used to recreate the DK-S128 $Pmod^{TM}$ display example application project from scratch using the e² studio ISDE.

Step 1: Create a new project with RTOS included:

- 1. Create a new Synergy project by clicking File->New->Synergy C\C++ Project.
- 2. Select the 'Renesas Synergy C Executable Project'.
- 3. Enter the project name and setup the Synergy license file.
- 4. Choose the DK-S128 board by selecting **S128 DK** in the **Board** window.
- 5. Choose BSP option in the Project Template Selection window.

Device Selection	Project Template Selection
SSP version: 1.4.0 V Board: 5128 DK V Device: R7FS128783A01CFM	 BSP Base Board Support Package for the chosen Synergy family. [Renesas.Synergy.1.4.0.pack] Blinky Blinky project. [Renesas.Synergy.1.4.0.pack] Blinky with ThreadX Threaded version of Blinky project. [Renesas.Synergy.1.4.0.pack]

Figure 4 Synergy Project Creation window

Step 2: Create Control thread

- 1. Under the **Thread** tab, click '**New Thread**' to create a new thread.
- 2. Set the property of this new thread, as shown in Figure 5.
- 3. In the **Control Thread Stacks** window, click '**New Stack**'to add the ADC driver (ADC Driver on r_adc), GPT driver (Timer driver on r_gpt) modules, as shown in Figure 5.



Figure 5 Adding ADC and r_gpt driver module

4. Go to the **Properties** tab for the ADC driver and select the resolution and channels, as shown in Figure 6. For details on the ADC driver properties, see the ADC module guide. Use the keyword "r_adc" in this <u>link</u> to download the ADC module guide document.



g_adc ADC Driver on r_adc								
Settings	Property	Value						
Information	✓ Common							
mormation	Parameter Checking	Enabled						
	 Module g_adc ADC Driver on r_adc 							
	Name	g_adc						
	Unit	0						
	Resolution	12-Bit						
	Alignment	Right						
	Clear after read	On						
	Mode	Continuous Scan						
	Channel Scan Mask	Select channels below						
	Channel 0	Unused						
	Channel 1	Unused						
	Channel 2	Unused						
	Channel 3	Unused						
	Channel 4	Unused						
	Channel 5	Unused						
	Channel 6	Unused						
	Channel 7 (S3A7/S124 Only)	Unused						
	Channel 8 (S3A7/S124 Only)	Unused						
	Channel 9 (S3A7/S124 Only)	Unused						
	Channel 10 (S3A7/S124 Only)	Unused						
	Channel 11 (S3A7 Only)	Unused						
	Channel 12 (S3A7 Only)	Unused						
	Channel 13 (S3A7 Only)	Use in Normal/Group A						
	Channel 14 (S3A7 Only)	Unused						

Figure 6 ADC Module properties window

5. Go to the **Properties** tab for the r_gpt driver and set the properties, as shown in Figure 7. For details on GPT driver properties, see the GPT module guide. Use the keyword "r_gpt" in this <u>link</u> to download the GPT module guide document.

	Duranta Milan									
Settings	Property	Value								
Information	✓ Common									
	Parameter Checking	Default (BSP)								
	 Module g_pwm_timer Timer Driver on r_gpt 									
	Name	g_pwm_timer								
	Channel									
	Mode	PWM								
	Period Value	10								
	Period Unit	Milliseconds								
	Duty Cycle Value	0								
	Duty Cycle Unit	Unit Percent								
	Auto Start	True								
	GTIOCA Output Enabled	True 📲								
	GTIOCA Stop Level	Pin Level Low								
	GTIOCB Output Enabled	False								
	GTIOCB Stop Level	Pin Level Low								
	Callback	NULL								
	Interrupt Priority	Disabled								

Figure 7 gpt module properties window

6. In the **Control Thread Stacks** window, click **New Stack** to add the **External IRQ framework on sf_external_irq** for user button S1 and S2, as shown in Figure 8.





Figure 8 Adding IRQ Framework

7. Go to **Properties** window for r_icu driver and set their properties, as shown in Figure 9.



	ternal IRQ Driver on r_icu				
Settings	Property	Value			
Information	✓ Common	l.			
mormation	Parameter Checking	Default (BSP)			
	 Module g_irq_s2 External IRQ Driver on r_icu 				
	Name	g_irq_s2			
	Channel	3			
	Trigger	Falling			
	Digital Filtering	Enabled			
	Digital Filtering Sample Clock (Only valid when Digital Filtering is Enabled)	PCLK / 64			
	Interrupt enabled after initialization	True			
	Callback NULL				
	Interrupt Priority	Priority 2			
	🛿 📳 Problems 🚡 Pin Conflicts 📮 Console 🔗 Search 🎲 Call Hierarchy 💁 Break	cpoints 🚺 Memory			
g_external	irq0 External IRQ Driver on r_icu				
g_external_ Settings	irq0 External IRQ Driver on r_icu Property	cpoints () Memory Value			
g_external	irq0 External IRQ Driver on r_icu Property Common	Value			
g_external_ Settings	Irq0 External IRQ Driver on r_icu Property Common Parameter Checking				
g_external_ Settings	irq0 External IRQ Driver on r_icu Property ✓ Common Parameter Checking ✓ Module g_external_irq0 External IRQ Driver on r_icu	Value Default (BSP)			
g_external_ Settings	irq0 External IRQ Driver on r_icu Property Common Parameter Checking Module g_external irq0 External IRQ Driver on r_icu Name	Value Default (BSP) g_irq_s1			
g_external_ Settings	irq0 External IRQ Driver on r_icu Property Common Parameter Checking Module g_external_irq0 External IRQ Driver on r_icu Name Channel	Value Default (BSP) g_irq_s1			
g_external_ Settings	irq0 External IRQ Driver on r_icu Property Common Parameter Checking Module g_external_irq0 External IRQ Driver on r_icu Name Channel Trigger	Value Default (BSP) g_irq_s1 0 Falling			
g_external_ Settings	irq0 External IRQ Driver on r_icu Property Common Parameter Checking Module g_external_irq0 External IRQ Driver on r_icu Name Channel Trigger Digital Filtering	Value Default (BSP) g_irq_s1			
g_external_ Settings	irq0 External IRQ Driver on r_icu Property Common Parameter Checking Module g_external_irq0 External IRQ Driver on r_icu Name Channel Trigger Digital Filtering Digital Filtering Sample Clock (Only valid when Digital Filtering is Enabled)	Value Default (BSP) g_irq_s1 0 Falling Enabled			
g_external_ Settings	irq0 External IRQ Driver on r_icu Property Common Parameter Checking Module g_external_irq0 External IRQ Driver on r_icu Name Channel Trigger Digital Filtering	Value Default (BSP) g_irq_s1 0 Falling Enabled PCLK / 64			

Figure 9 r_icu driver properties window

8. Go to **Properties** window for sf_external_irq framework and set the properties, as shown in Figure 10.

	Property	Value				
Settings		value				
nformation	Common Parameter Checking					
	Parameter Checking					
	Module g_sf_irq_s1 External IRQ Framework on sf_external_irq Name q_sf_					
	Name					
	Event	Semaphore Put				
Properties	업 😰 Problems 🎇 Pin Conflicts 📃 Console 🛷 Search 🍰 Call Hierarchy 🎭	Breakpoints 🚺 Memory				
g_sf_irq_s2	Image: Search	Breakpoints Memory				
g_sf_irq_s2 Settings	External IRQ Framework on sf_external_irq					
g_sf_irq_s2 Settings	External IRQ Framework on sf_external_irq Property Common					
g_sf_irq_s2 Settings	External IRQ Framework on sf_external_irq Property Common Parameter Checking	Value				
	External IRQ Framework on sf_external_irq Property Common	Value				

Figure 10 sf_external_irq properties window



Step 3: Create LCD Thread

- 1. Under the **Thread** tab, click '**New Thread**' to create a new thread.
- 2. Set the property of this new thread, as shown in Figure 11.
- 3. In the **LCD Thread Stacks** window, click 'New Stack' to add the SPI driver module, r_sci_spi, as shown in Figure 11.
- 4. Go to the **Properties** tab for SPI driver, select the channel and set the interrupt priorities, as shown in Figure 11.

Settings Property Value Ymbol Coll Thread Steck size (bytes) 512 Settings Property Value Value Mame Steck size (bytes) 512 Settings Property Value Property Value Value Property Value Value Or Driver on r_can Analog Property Value Parameter Checking Default (BSP) Value V Matter Driver on r_ric Graphics X-Ware Value V EX Master Driver on r_ric Graphics X-Ware Power V LART Driver on r_sci_uart Storage Storage Data sampling on odd edge, data variation on et Clock Phase V LART Driver on r_sci_uart System Storage Bit rate S000000 Bit rate S000000 Bit rate S000000 Editade Bit rate S000000 Editade Calback Null Receive Interrupt Priority Priority 2 Transmit End Interrupt Priority Priority 2 Transmit End Interrupt Priority Priority 2 Priority 2 Prior	[hrea		read 👔 Remove	Properti	ies 🛿 🖹 Problems	- 🏠 F	Pin Conflicts 🛛 📮 Conso	le 🔗 Search	
Setting Thread Symbol Connectivity Cannectivity Cannectivity Framework X-Ware Connectivity Framework X-Ware Property Value Cannectivity Property Value Cannectivity Framework X-Ware Property Value Connectivity Framework X-Ware Property Value Connectivity Framework X-Ware Set Driver on r_sci_spi SPI Driver on r_sci_spi Dever Storage System Transfer Nover Transfer Nover Transfer Nover Transfer Nover Transfer Nover Driver Driver Driver Driver Driver Driver Prover Driver Driver Driver Driver Driver Driver Driver Prover Driver <th>⇔ L</th> <th>CD Thread</th> <th>^</th> <th>New Thr</th> <th>read</th> <th></th> <th></th> <th></th> <th></th>	⇔ L	CD Thread	^	New Thr	read				
Symbol Icd_thread LCD Thread Name Size Priority 1 Auto start Disabled Time slicing interval (tick) 1 Priority 1 Auto start Disabled Time slicing interval (tick) 1 Priority 1 Auto start Property Value CAN Driver on r_can Connectivity Framework V Crypto X-Ware Property Value Graphics Name g.spi9 Iput Spi Driver on r_sci_spi Operating Mode Master VAUR Driver on r_sci_spi Power Clock Phase Data sampling on odd edge, data variation on exit of the row hen idle Spi Driver on r_sci_uart System Bit Order MSB First Bitrate S000000 Bit Rate Modulation Enable Disable Disable Disable Disable Callback NULL Transfer Transfer Transfer Transfer Transfer				Settings				Value	
Stack size (bytes) 512 Priority 1 Auto stat Disabled Time slicing interval (ticks) 1 Property Value COnnectivity Property Property Value CAN Driver on r_can Onnectivity Crypto Framework X-Ware Model g_spi9 SPI Driver on r_sci_spi V Adduster Graphics Input Input SPI Driver on r_sci_spi Power Power Sorage V UART Driver on r_sci_gart System System System Timers Timers Timers Timers Tansfer Transfer							~	lcd_thread	
Journey (view) Jour					Name		-	LCD Thread	
Auto start Time slicing interval (tick) Disabled 1 Nove LCD Thread Stacks New Stack > move Property Value CAN Driver on r_can Connectivity Framework X-Ware Poperty Value Ic2C Master Driver on r_sci_j2c Graphics Framework X-Ware Module g_spi9 SPI Driver on r_sci_spi Default (BSP) Ic2C Slave Driver on r_sci_j2c Graphics New Channel 9 Generation of the state of			¥			ytes)		512 🦛	
Image: LCD Thread Stacks Time slicing interval (ticks) 1 nove LCD Thread Stacks New Stack > (Time nove) Property Value							_	1	
ICD Thread Stacks New Stack > Property Value								Disabled	
Analog Driver CAN Driver on r_can Connectivity I2C Master Driver on r_sic_j2c Crypto I2C Master Driver on r_sic_j2c Input SPI Driver on r_sic_spi Monitoring VORT Driver on r_sci_ant SPI Driver on r_sci_ant V Cannel 9 Operating Mode Master Clock Plase Data sampling on odd edge, data variation on excitation excitation excitation on excitation excitation on excit					Time slicing	interv	al (ticks)	1	
 Analog Driver CAN Driver on r_can I2C Master Driver on r_nic I2C Slave Driver on r_nic SPI Driver on r_nic SPI Driver on r_nic Monitoring I2D Marter I2C Master Driver on r_nic Input SPI Driver on r_nic IDVE Monitoring I2C Master Driver on r_nic SPI Driver on r_nic SPI Driver on r_nic Spi Driver on r_nic IDVE Monitoring Spi Driver on r_nic IDVE Monitoring IDVE IDVE Monitoring IDVE IDVE Monitoring IDVE IDVE	nove	LCD Thread Stacks	🗐 N	ew Stack >	Remove	-	Property		Value
 I2C Master Driver on r_riic I2C Master Driver on r_sci_j2c I2C Master Driver on r_sci_j2c I2C Slave Driver on r_sci_j2c Input SPI Driver on r_sci_spi Monitoring System Timers Transfer 	ф	CAN Driver on a can	2						
 I2C Master Driver on r_sci_i2c Graphics Iput SPI Driver on r_sci_spi Monitoring SPI Driver on r_sci_spi Monitoring Power Inver on r_sci_spi VART Driver on r_sci_uart System Transfer Transfer 	Ψ.	-	-				2-1	SPI Driver on r_sci_spi	
 I2C Slave Driver on r_sric_slave Iput SPI Driver on r_sric_spi Monitoring SPI Driver on r_sci_spi Power Storage Timers Transfer 	Ŧ	-	Crypto	`	X-Ware				
 Input Input SPI Driver on r_ric_slave Monitoring SPI Driver on r_sci_spi Power VART Driver on r_sci_uart Storage System Timers Transfer 	Ð	I2C Master Driver on r_sci_i2c	Graphics	>		п			
 ♦ SPI Driver on r_rspi ♦ SPI Driver on r_rsci_spi Power VART Driver on r_sci_spi Power Storage System Timers Transfer 	+	I2C Slave Driver on r_riic_slave	Input	>					
 	#	SPI Driver on r rspi	Monitoring	>					
♦ UART Driver on r_sci_uart Storage > System > Timers > Transfer >		= 1	2			14		·	
System > Timers > Transfer > Transmit Interrupt Priority Priority 2 Transmit Interrupt Priority Priority 2								Error	
Timers > Transfer Bit Rate Modulation Enable Disable Callback NULL Receive Interrupt Priority Priority 2 Transmit Interrupt Priority Priority 2	Ψ	UARI Driver on r_sci_uart	Storage	>					
Imers > Transfer Callback NULL Receive Interrupt Priority Priority 2 Transmit Interrupt Priority Priority 2	_		System	>					
Transfer > Receive Interrupt Priority 2 Transmit Interrupt Priority 2 Priority 2			Timers	>				dulation Enable	
Transmit Interrupt Priority 2	nove		Transfer	>				Defender	
									Priority 2 Priority 2
Error Interrupt Priority 2		1 1							

Figure 11 Adding SPI driver module

5. Under the newly added r_sci_spi driver, disable the DTC driver for the transfer and reception module in the configurator, as shown in Figure 12.



Figure 12 Disabling DTC driver



Step 4: Update Pin Configurations

Go to the **Pins** tab and change the pin configurations for the following ports, as shown in Figure 11:

- 1. P205 to Output mode (PMOD_SS)
- 2. P302 to Peripheral mode (PMOD_EN)
- 3. P111 to Output mode (PMOD_RST)
- 4. P303 to Output mode (PMOD_DC)

P205 Configuration				
Mode:	Output mode (Initial High)	~	P302 Configuration	
Pull up:	None	~	Mode:	Peripheral mode
IRQ:	None	~	Pull up:	None
Drive Capacity:	Low	~	IRQ:	None
Output type:	CMOS	~	Drive Capacity:	Low
Chip input/output			Chip input/output	
P205:	✓ GPIO	\sim	P302:	✓ GPT4_GTIOCA
P111 Configuration				
Mode:	Output mode (Initial High)	\sim	P303 Configuration	
Pull up:	None	~	Mode:	Output mode (Initial Low) ~
			Pull up:	None
IRQ:	None	~	Drive Capacity:	Low
Drive Capacity:	Low	\sim		
Chip input/output			Chip input/output	
P111:	✔ GPIO	\sim	P303:	✓ GPIO ∨

Figure 13 Pin Configuration

Step 5: Generate Project Content

Click the Generate Project Content button. It generates the project files using the configuration option you selected.



Figure 14 Generate Project Content button



Step 6: Application Project files

- 1. After step 5, the e^2 studio ISDE generates the application project files with the configuration chosen.
- These files are a place holder for adding the user application code.
 You can either write your own application functions for these threads, or you can copy the existing DK_S128
 Simple Pmod Display demonstration application project source files to recreate this example demonstration.
- 3. To recreate this example demonstration, go to the existing dk_s128_simplePmodDisplay project src folder and copy the following files/folders contents to your newly created project.
 - A. Lcd_setup (folder)
 - B. control_thread_entry.c
 - $C. \ \texttt{lcd_thread_entry.c}$
 - D. util.h.

Step 7: Compiling the project

Build the application project by clicking the hammer icon as seen in the menu bar in Figure 15.



Figure 15 Build button

3. Running the Pre-existing DK-S128 Simple Pmod[™] Display Demonstration

3.1 Powering up the Board

This section describes how to connect the board to power, the J-Link debugger to the PC, the board to the PC USB port, and how to run the debug application to see it in action.

To connect to the board:

- 1. Connect the Micro USB end of the supplied USB cable to the DK-S128 board J12 connector (DEBUG_USB).
- Note: The kit contains a SEGGER J-Link[®] On-board (OB). The J-Link provides full debug and programming capabilities for the DK-S128 kit.
- 2. Connect the other end of the USB cable to the USB port on your workstation. Wait for LED4 to turn solid green, indicating a good connection.

3.2 Importing, Building, and Running the Project

See the *Renesas Synergy Project Import Guide* (r11an0023eu0119-synergy-ssp-import-guide.pdf) for instructions on importing the project into e² studio to build and run the project.

Note: You need to select the "SimplePmodLCD_DKS128 Debug" GDB Hardware Debugging configuration for debugging.

3.3 Verifying the Demonstration

Connect the PmodTM LCD display (included as part of the DK-S128 kit) to PmodTM connector J4. Verify that the J3 header near the battery is configured with two jumpers to make connections, as shown in Figure 16.





Figure 16 Jumper setting

Figure 17 shows the switch settings used to run this application project.



Figure 17 SW4/5/6/7 settings for Simple Pmod[™] Display demonstration

Once the DK-S128 kit is powered up and loaded with this application project, you should see a Welcome screen on the Pmod[™] LCD display, as shown in Figure 18.



Figure 18 Splash Screen

After a few seconds (or once S1 is pressed), the program goes to the screen with instructions, as shown in Figure 19.





Figure 19 Welcome and Instruction screens

Then, you can use the S1 pushbutton to go to the next step (provided that the application has finished drawing the current screen) and the S2 pushbutton to change the screen orientation in the clockwise direction. Changing the screen orientation restarts the sequence of drawings, but the instruction screen is now skipped.

While the application is running, you can use the potentiometer to adjust the backlight intensity. If you cannot see any difference in the strength of the backlight, verify that the Pmod is outputting 3.3V on the Vcc pin.

4. Next Steps

- 1. Visit http://renesassynergy.com/kits/dk-s128 for more information about the DK-S128 kit.
- 2. Visit <u>renesassynergy.com/tools</u> to learn more about development tools and utilities.
- 3. Visit <u>http://www.renesassynergy.com/gallery</u> to download development tools and utilities.
- 4. To learn more about:
 - A. Synergy kits, see http://www.renesassynergy.com/kits
 - B. Synergy Microcontrollers, see http://www.renesassynergy.com/microcontrollers
 - C. Synergy Software, see <u>http://www.renesassynergy.com/software</u>
 - D. Synergy Solutions, see http://www.renesassynergy.com/solution

5. Limitations and Assumptions

In this application project, we set the DIP switch 1 (LED_DIS) in SW6 to OFF. You cannot use Reset button (SW3) to restart the kit if the DIP switch 1 (LED_DIS) in SW6 is set to OFF since the P2_1 is shared between LED2 and boot mode, in DK-S128 schematics.



Website and Support

Support: <u>https://synergygallery.renesas.com/support</u>

Technical Contact Details:

- America: <u>https://www.renesas.com/en-us/support/contact.html</u>
- Europe: <u>https://www.renesas.com/en-eu/support/contact.html</u>
- Japan: https://www.renesas.com/ja-jp/support/contact.html

All trademarks and registered trademarks are the property of their respective owners.



Revision History

		Descript	tion
Rev.	Date	Page	Summary
1.00	Aug 29, 2017		Initial version
1.01	Mar 16, 2018	_	Updated for 1.4.0

Notice

- 1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
- Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
- 3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
- 4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
- 5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
 - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc. Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

- 6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
- 7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and malfunction of measures to the final products or systems manufactured by you.
- 8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
- 9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
- 10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
- 11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics
- 12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.
- (Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
- (Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

http://www.renesas.com



SALES OFFICES

Renesas Electronics Corporation

Refer to "http://www.renesas.com/" for the latest and detailed information. Renesas Electronics America Inc. 1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A. Tet: +1-408-432-8888 Licknonics Eanada Limited 2051 Yong Street, Suite 8309 Richmond Hill, Ontario Canada L4C 913 Tet:: +1-905-237-2004 Renesas Electronics Europe Limited Dukes Meadow, Milboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K Tet: +44-1052-851-700, Fax: +44-128-851-300 Renesas Electronics Furope CmbH Arcadiastrasse 10, 40472 Disselforf, Germany Tet: +49-211-6503-0, Fax: +44-128-651-700, Fax: +44-128-651-300 Renesas Electronics (China) Co., Ltd. Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China Tet: +80-10-8235-1155, Fax: +80-10-8235-7679 Renesas Electronics (Shanghai) Co., Ltd. Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China Tet: +80-2226-5688, Fax: +852-1226-0399 Renesas Electronics (Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong Tet: +805-2265-6688, Fax: +862-21226-0399 Renesas Electronics Singapore Pte. Ltd. 80 Bendemeer Road, Unit #06-02 Hyllux Innovation Centre, Singapore 339949 Tet: +605-213-2020, Fax: +456-213-000 Renesas Electronics Malaysis Sch.Brd. Mont 7107, 100 Feet Road, Hal2, 2nd Stage, Indiranagar, Bangalore 560 038, India Tet: +805-4730500, Fax: +456-220877 Renesas Electronics Malaysis Sch.Brd. Mon 777C, 100 Feet Road, Hal2, and Captury Teace, Gangam-gu, Seoul, 06265 Korea Tet: +90-873795-9390, Fax: +452-2558-5338