Renesas Synergy™ Platform

Low Voltage Detection HAL Module Guide

R11AN0097EU0100
Rev.1.00
Aug 2, 2017

Introduction

This module guide will enable you to effectively use a module in your own design. Upon completion of this guide, you will be able to add this module to your own design, configure it correctly for the target application and write code, using the included application project code as a reference and efficient starting point. References to more detailed API descriptions and suggestions of other application projects that illustrate more advanced uses of the module are available on the Renesas Synergy Knowledge Base (as described in the References section at the end of this document), and should be valuable resources for creating more complex designs.

The Low Voltage Detection (LVD) HAL module is a high-level API for voltage-detection applications and is implemented on `r_lvd`. The LVD HAL module uses the LVD peripheral on the Synergy MCU. A user-defined callback can be created to notify the CPU when a voltage-detection event is triggered. The VCC is the source for all voltage-detection functions.

Contents

1. LVD HAL Module Features

The LVD HAL module supports the following functions:

- V_{CC} as the voltage-detection input
- One build-time configurable low-voltage detector (through OFS1 register)
- Two run-time configurable low-voltage detectors
- Two result flags; one for a threshold check and one for the current state
- Support for both interrupt or polling-event checking

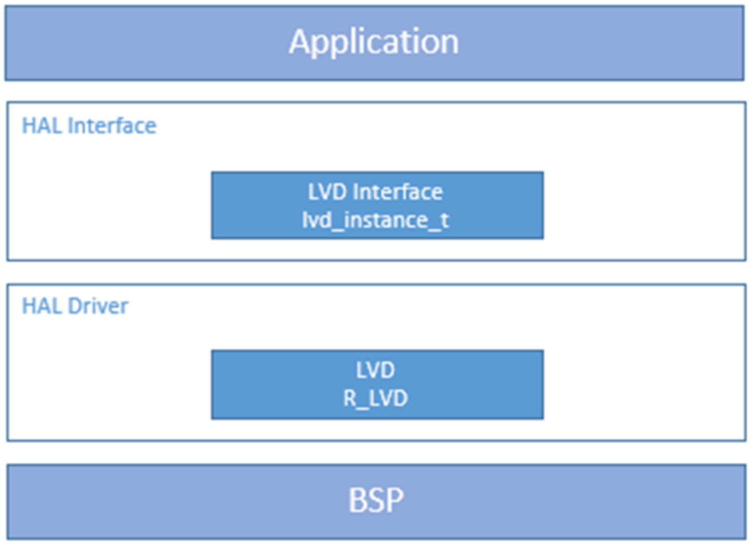

Figure 1 LVD HAL Module Block Diagram

2. LVD HAL Module APIs Overview

The LVD HAL module defines APIs for `opening`, `closing`, `statusGet` and `statusClear`. The following table includes a complete list of the available APIs, an example API call, and a short description of each API. A table of status return values follows the API summary table.

Table 1 LVD HAL Module API Summary

Function Name	Example API Call and Description
.open	`g_lvd.p_api->open(g_lvd.p_ctrl, g_lvd.p_cfg;` Initializes a low voltage detection driver according to the passed in configuration structure. Enables an LVD peripheral based on configuration structure.
.statusGet	`g_lvd.p_api->statusGet(g_lvd.p_ctrl, &monitor_status);` Get the current state of the monitor, (threshold crossing detected, voltage currently within range) Can be used to poll the state of the LVD monitor at any time. Must be used if the peripheral was initialized with the lvd_response_t set to LVD_RESPONSE_NONE.
.statusClear	`g_lvd.p_api->statusClear(g_lvd.p_ctrl);` Clears the latched status of the monitor. Must be used if the peripheral was initialized with lvd_response_t set to LVD_RESPONSE_NONE.
.close	`g_lvd.p_api->close(g_lvd.p_ctrl, g_lvd.p_cfg);` Disables the LVD peripheral. Closes the driver instance.
.versionGet	`g_lvd.p_api->versionGet(&version);` Retrieve the API version with the version pointer.

Note: For details on operation and definitions for the function data structures, typedefs, defines, API data, API structures, and function variables, review the *SSP User's Manual* API References for the associated module.

Table 2 Status Return Values

Name	Description
SSP_SUCCESS	API Call Successful
SSP_ERR_IN_USE	Driver already open or unable to acquire hardware lock
SSP_ERR_NOT_OPEN	Unit is not open
SSP_ERR_ASSERTION	Invalid configuration value
SSP_ERR_INVALID_MODE	If the attempted mode is invalid for this configuration

Note: Lower-level drivers may return common error codes. Refer to the *SSP User's Manual* API References for the associated module for a definition of all relevant status return values.

3. LVD HAL Module Operational Overview

The LVD HAL module supports the configuration and operation of LVD monitors in the Synergy MCUs. The LVD HAL module provides configuration structures with all the information needed to fully configure a single LVD monitor. There is one instance of the LVD HAL module per LVD monitor instance, with the exception of the LVD0 monitor. The LVD0 monitor is not configurable at runtime; it must be configured at compile time via the OFS1 register.

Both the LVD1 and LVD2 monitors are configurable at runtime by this module. The `open` function allows the user to configure and enable an LVD monitor with a single function call; the `close` function disables the LVD monitor. The `statusGet` function returns the current status of the LVD monitor. The `statusGet` function should be used if the module is in polling mode, i.e. without the LVD monitor interrupt enabled. The monitor status consists of two flags. The first flag is a latched flag called `crossing_detected` which indicates whether or not the voltage being monitored has crossed the voltage threshold. In polling mode, this flag must be cleared via a call to `statusClear`. The flag does not need to be cleared explicitly if the LVD interrupt is in use; it is cleared in the LVD interrupt by the driver code after the user-callback function is called. The second flag, `current_state`, is the instantaneous status of the monitored voltage with respect to the voltage threshold; this flag is not latched and its status changes with the monitored voltage changes.

The LVD HAL module can be configured to enable one or several of the LVD peripheral interrupts. If using an interrupt, the user should provide a callback function for that monitor. Separate callback routines should be provided for each LVD monitor.

The LVD HAL module requires functionality provided by the BSP; this driver makes use of the BSP hardware locks for locking registers, as well as enabling and clearing interrupts.

3.1 LVD HAL Module Important Operational Notes and Limitations

3.1.1 LVD HAL Module Operational Notes

- Once the appropriate values are chosen for these LVD settings, you should add the code to call the LVD HAL module `open` API function for your project. This function should be called once early in the application.
- The module can be closed and opened whenever the configuration of an LVD monitor needs to be changed. Calling the LVD `open` API function configures and enables the LVD hardware peripheral for the specified LVD monitor.
- The `close` function disables the LVD monitor and closes the driver.
- Using this module to configure the LVD peripheral to generate an interrupt requires the corresponding interrupt to be enabled in the module's Properties tab.
- When using the LVD interrupts, a callback function is not required, but is recommended.
- A unique callback function for each LVD interrupt is not required, but is recommended.
- Clock system initialization, configuration and runtime modification are handled outside this module. This driver makes changes to the digital filter sample clock based on the user's choice of a sample clock divisor. The digital filter sample clock is derived from the low-speed on-chip oscillator (LOCO) system clock.
- Not all voltage thresholds are available on all MCUs.
- Digital filtering of the V_{CC} input to the LVD monitor is not available on all Synergy MCU devices.
- The LVD driver requires functionality provided by the BSP; it makes use of hardware locks provided by the BSP for register locks, as well as enabling and clearing interrupts.

3.1.2 LVD HAL Module Limitations

- Configuring and enabling a LVD monitor requires specific timing constraints and register write ordering. Due to the constraints, the entire process of configuring and enabling a voltage monitor is most effective when performed by a single function. The `open` API function performs configuration and enables the monitor and properly enforces the timing and register write ordering constraints.

- All the Synergy MCU Series include option-setting memory that can be used to set the operating state of peripherals after a reset. The (option function select) OFS can be used to set the state of the IWDT, WDT, LVD and CGC high-speed on-chip oscillator (HOCO).
- See the latest SSP Release Notes for any additional operational limitations for this module.

Note: The LVD0 can only be configured via the OFS registers. The LVD0 does not support Register Start mode.

Table 3 OFS Settings for Low Voltage Detection

Control	Description	
LVD0 detection level	S7 & S5 Series 2.94 V 2.87 V 2.80 V	S3 & S1 Series 3.84 V 2.82 V 2.51 V 1.90 V 1.70 V
LVD0 detection start	Automatically starts the LVD0 after a Reset, if enabled	

Set the OFS register values in the Synergy Configuration editor through the BSP tab Properties dialog box.

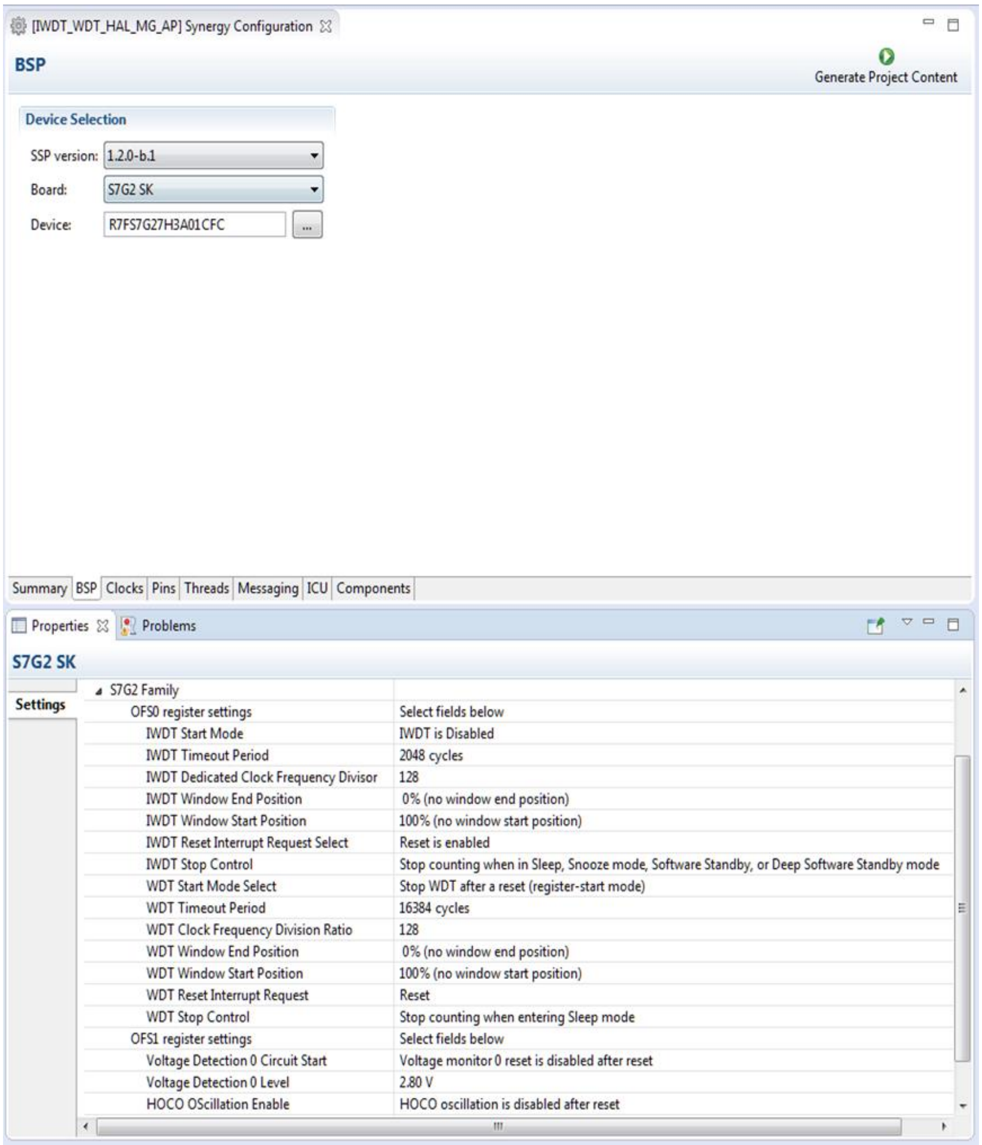

Figure 2 OFS Register Settings

4. Including the LVD HAL Module in an Application

This section describes how to include the LVD HAL Module in an application using the SSP configurator.

Note: It is assumed you are familiar with creating a project, adding threads, adding a stack to a thread and configuring a block within the stack. If you are unfamiliar with any of these items, refer to the first few chapters of the SSP User's Manual to learn how to manage each of these important steps in creating SSP-based applications.

To add the LVD Driver to an application, simply add it to a thread using the stacks selection sequence given in the following table. (The default name for the LVD HAL module is `g_lvd`. This name can be changed in the associated Properties window.)

Table 4 LVD Selection Sequence

Resource	ISDE Tab	Stacks Selection Sequence
g_lvd Low Voltage Detection Driver on r_lvd	Threads	New Stack> Driver> Power> Low Voltage Detection Driver on r_lvd

When the LVD Driver on `r_lvd` is added to the thread stack as shown in the following figure, the configurator automatically adds any needed lower-level modules. Any drivers that need additional configuration information are box text highlighted in Red. Modules with a Gray band are individual modules that stand alone.

Figure 3 LVD HAL Module Stack

5. Configuring the LVD HAL Module

The LVD HAL module must be configured by the user for the desired operation. The SSP configuration window automatically identifies (by highlighting the block in red) any required configuration selections, such as interrupts or operating modes, which must be configured for lower-level modules for successful operation. Only those properties that can be changed without causing conflicts are available for modification. Other properties are 'locked' and are not available for changes, and are identified with a lock icon for the 'locked' property in the Properties window in the ISDE. This approach simplifies the configuration process and makes it much less error-prone than previous 'manual' approaches to configuration. The available configuration settings and defaults for all the user-accessible properties are given in the properties tab within the SSP configurator, and are shown in the following tables for easy reference.

One of the properties most often identified as requiring a change is the interrupt priority; this configuration setting is available within the Properties window of the associated module. Simply select the indicated module and then view the Properties window; the interrupt settings are often toward the bottom of the properties list, so scroll down until they become available. Also note that the interrupt priorities listed in the Properties window in the ISDE includes an indication as to the validity of the setting based on the targeted MCU (CM4 or CM0+). This level of detail is not included in the following configuration properties tables, but is easily visible within the ISDE when configuring interrupt-priority levels.

Note: You may want to open your ISDE, create the module, and explore the property settings in parallel with looking over the following configuration table settings. This helps to orient you and can be a useful 'hands-on' approach to learning the ins and outs of developing with SSP.

Table 5 Configuration Settings for the LVD HAL Module on r_lvd

ISDE Property	Value	Description
Parameter Checking	Enabled, Disabled, BSP (Default: BSP)	Selects if code for parameter checking

ISDE Property	Value	Description
		is to be included in the build
Name	Default: g_lvd	LVD Driver module name
Monitor Number	1	Monitor number selection
Digital Filter, enable by selecting a valid sample clock rate (S7G2 only).	Enabled with clock LOCO/2/4/8/16, Disabled (Default: Disabled)	Digital filter selection
Voltage Threshold	2.85V (Vdet1_13) (S7G2 only).	Voltage threshold level selection
Detection Response, either reset, interrupt, non-maskable interrupt, or no response (polling mode).	Maskable interrupt is triggered when the voltage crosses the detection threshold. The non-maskable interrupt is triggered when voltage crosses the detection threshold. The MCU resets when the voltage falls below the detection threshold. No response: the driver is in polled mode (using statusGet and statusClear functions.) (Default: Maskable interrupt)	Detection response selection
Voltage Slope	Threshold crossing detected with decreasing voltage. Threshold crossing detected with increasing voltage. Threshold crossing detected with increasing or decreasing voltage. (Default: Threshold crossing detected with decreasing voltage.)	Direction of voltage slope for threshold detection
Negation of Monitor Signal	Negation of reset signal is based on delay from the reset. Negation of reset signal is based on delay from voltage returning to normal range. (Default: Negation of reset signal is based on delay from reset.)	Negation option selection
Monitor Interrupt Callback	NULL	Interrupt callback function name
LVD Monitor Interrupt Priority	Priority 0 (highest), 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 (lowest, not valid if using Thread X), Disabled (Default: Disabled)	Interrupt priority setting

Note: The example settings and defaults are for a project using the Synergy S7G2 MCU Group. Other MCUs may have different default values and available configuration settings.

In some cases, settings (other than the defaults) for a module can be desirable. For example, it might be useful to select a voltage level appropriate for the target system.

5.1 LVD HAL Module Clock Configuration

Clock system clock initialization, configuration and runtime modification are handled outside this module. This module only makes changes to the digital filter sample clock based on the user's choice of sample clock divisor. The digital filter sample clock is derived from the LOCO system clock.

5.2 LVD HAL Module Pin Configuration

The LVD HAL module measures the voltage on the VCC pin only and doesn't need to be configured.

6. Using the LVD HAL Module in an Application

The typical steps in using the LVD HAL module in an application are:

1. Initialize the LVD HAL module using the `open` API.
2. If using software polling, monitor the LVD status flags with the `statusGet` API and process accordingly. If using the interrupt mode, process accordingly within the callback function which will return both the monitor number as well as the status.

3. Process as needed
4. Close the LVD Instance with the `close` API

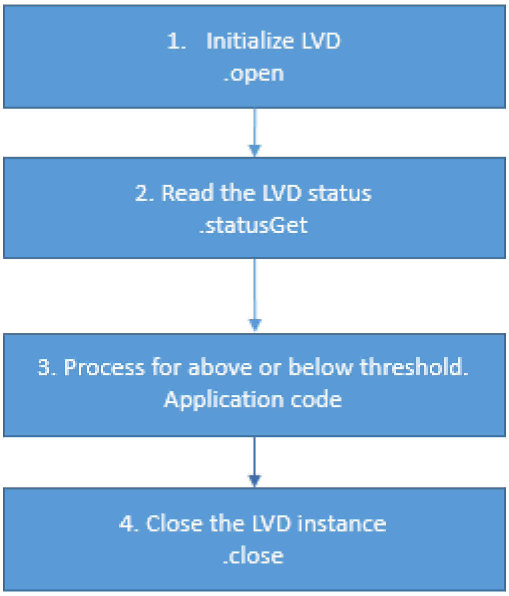

Figure 4 Flow Diagram of a Typical LVD HAL Module in software polling mode

7. The LVD HAL Module Application Project

The application project associated with this module guide demonstrates the aforementioned steps in a full design. You may want to import and open the application project within the ISDE and view the configuration settings for the LVD HAL module. You can also read over the code (in `hal_entry.c`) used to illustrate the LVD APIs in a complete design.

The application project demonstrates the typical use of the LVD APIs, using LVD1 in software-polling mode and LVD2 in interrupt-driven mode to demonstrate both modes of operation. The application project main thread entry initializes both LVDs, then continuously scans the status of LVD1 and performs the application process if a voltage-threshold crossing is detected. A user-callback function is entered when an LVD2 voltage-threshold crossing is detected.

The following table identifies the target versions for the associated software and hardware used by the application project:

Table 6 Software and Hardware Resources Used by the Application Project

Resource	Revision	Description
e² studio	5.3.1 or later	Integrated Solution Development Environment
SSP	1.2.0 or later	Synergy Software Platform
IAR EW for Synergy	7.71.2 or later	IAR Embedded Workbench® for Renesas Synergy™
SSC	5.3.1 or later	Synergy Standalone Configurator
SK-S7G2	v3.0 to v3.1	Starter Kit

The following figure shows a simple flow diagram of the application project.

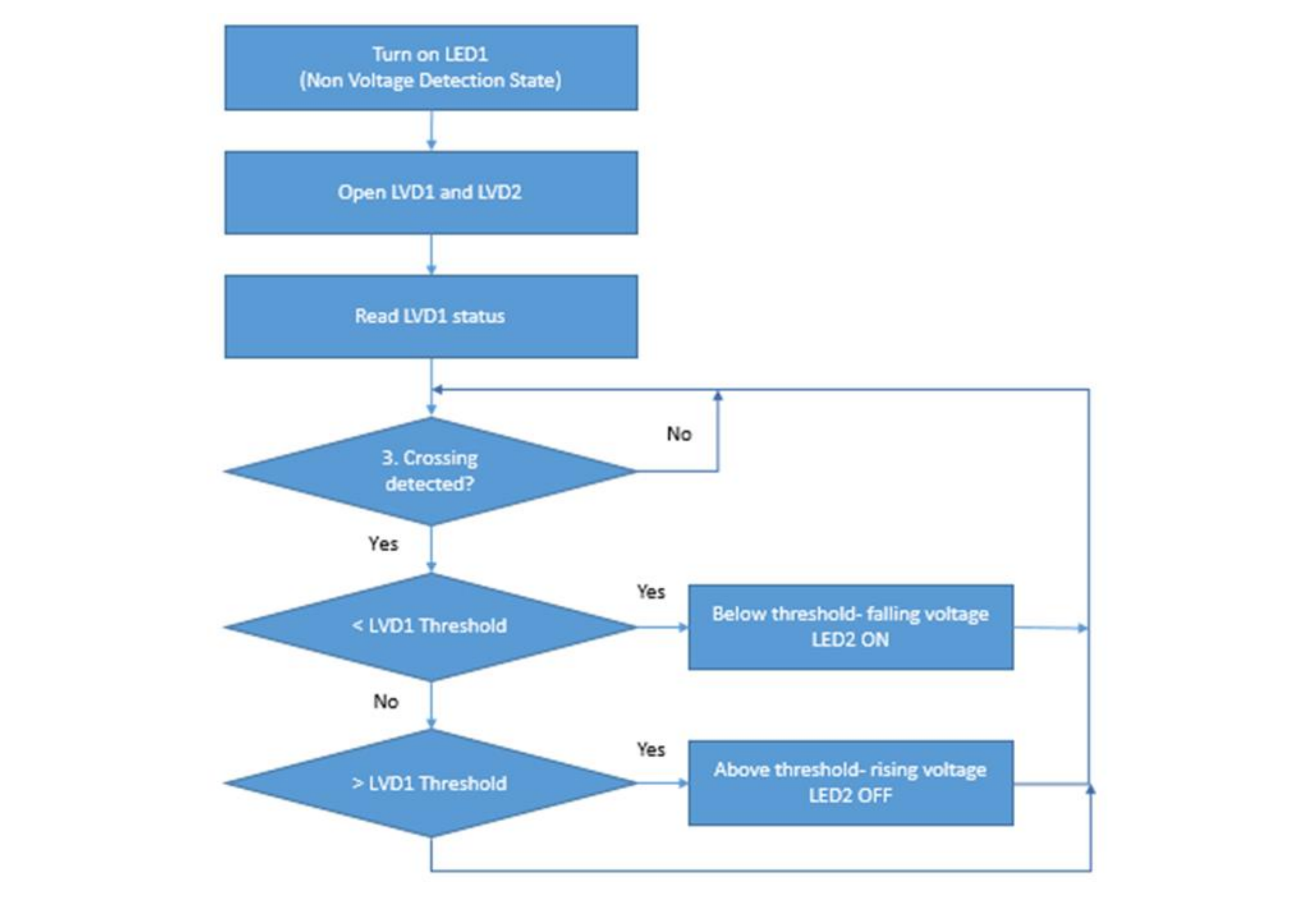

Figure 5 LVD HAL Module Application Project Flow Diagram with Callback

The `hal_entry.c` file is located in the project once it has been imported into the ISDE. You can open this file within the ISDE and follow along with the description provided to help identify key uses of APIs.

The first section of `hal_enty.c` has the header files that reference the LVD instance structures, and a function prototype to an error handler that is called if any of the LVD APIs do not return `SSP_SUCCESS`. This reference to instance structures is required; a low-voltage condition cannot be created with the debugger attached and a visual indication has to be given to notify whether an API error occurred.

The next section is the entry function for the main program-control section. The user LEDs are set to LED1 ON, LED2, and LED3 OFF to indicate a V_{CC} operating voltage above the LVD1 and LVD2 thresholds.

Use the `open` API to initialize the LVD1 and LVD2 instances. Inside the 'forever' while loop, the status of LVD1 is read. If the `status.crossing_detected` is set to `LVD_THRESHOLD_CROSSING_DETECTED`, the application uses `status.current_state` to check whether the crossing was due to a rising or falling voltage. If the current state is below the LVD1 threshold, it is assumed that a falling voltage has occurred; the RED LED is then turned on and the GREEN LED is turned off. If the current state is above the LVD1 threshold, it is assumed that a rising voltage has occurred; the RED LED is then turned off and the GREEN LED is turned on.

The user-callback function for LVD2 is in the next section; this function is called when LVD2 detects a voltage crossing over its threshold. The function uses `status.current_state` to check whether the crossing was due to a rising or falling voltage. If the current state is below the LVD2 threshold, it is assumed a falling voltage has occurred and the ORANGE LED is turned on. If the current state is above the LVD2 threshold, it is assumed a rising voltage has occurred and the ORANGE LED is turned off.

The following figure shows the relationship between the V_{CC}, the LVD thresholds, and the LEDs.

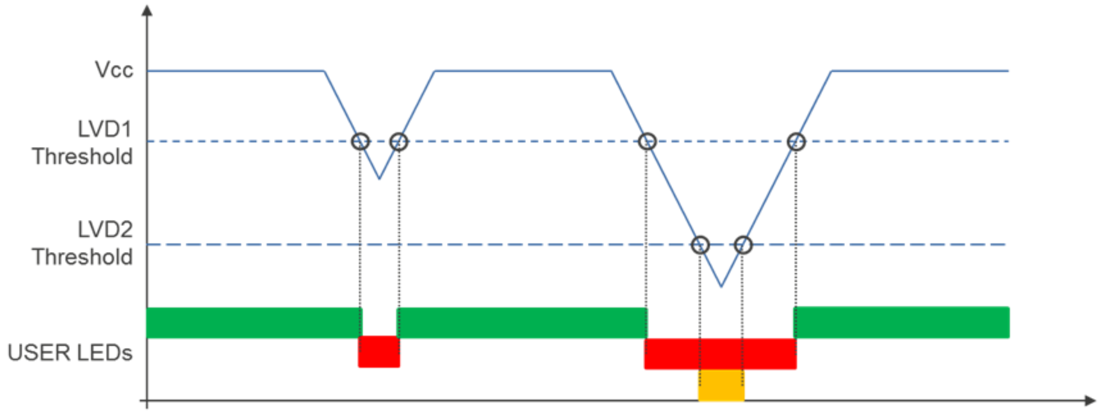

Figure 6 VCC/LVD/LED Relationship

A few key properties are configured in this application project. These properties support the required operations and the physical properties of the target board and MCU device. The following table lists properties with the values set for this specific project. You can also open the application project and view these settings in the Properties window as a hands-on exercise.

Table 7 LVD1 and LVD2 Configuration Settings for the Application Project

ISDE Property	Value Set
LVD1	
Name	g_lvd1
Monitor Number	1
Digital filter	Disabled
Voltage Threshold	2.92V (Vdet1_12)
Detection Response	No response, driver will be in polled mode
Monitor Interrupt Callback	NULL
LVD monitor Interrupt Priority	Disabled
LVD2	
Name	g_lvd2
Monitor Number	2
Digital filter	Disabled
Voltage Threshold	2.85V (Vdet2_7)
Detection Response	Maskable interrupt triggered when voltage crosses the detection threshold.
Voltage Slope	Threshold crossing detected with increasing or decreasing voltage
Monitor Interrupt Callback	cb_lvd2
LVD monitor Interrupt Priority	Priority 2

8. Customizing the LVD HAL Module for a Target Application

Some configuration settings are normally be changed by the developer from those shown in the application project. For example, the user can easily change the configuration settings for the voltage-detection levels, interrupt of polling mode, or whether the device asserts a reset.

9. Running the LVD HAL Module Application Project

To run the LVD HAL module application project and see it executed on a target kit, import the application code into your ISDE, compile and download it to the target.

To observe the LVD circuits operating, a variable voltage has to be applied to the SK-S7G2 Kit.

Typically, the SK-S7G2 target kit is powered through a USB cable attached to J19. To apply a variable voltage, a variable voltage power supply unit (PSU) is required, along with the following hardware modifications to the SK_S7G2 Kit.

Note: Hardware modifications to the SK-S7G2 Kit are done entirely at your own risk. Renesas accepts no liability for damage that may occur to the SK-S7G2 Kit as a result.

1. Remove R113
2. Remove R114
3. Connect TP29 to TP27 and solder a wire to connect.
4. Connect the DEBUG USB cable to J19 as normal to program the application project into the device.
5. To execute the application project, disconnect the DEBUG USB cable from J19, and inject a variable voltage across both pins of J31and ground to the GND pin of J23.

To implement the LVD application in a new project, use the following steps for defining, configuring, auto-generating files, adding code, compiling, and debugging on the target kit. Following these steps is a hands-on approach that can help make the development process with SSP more practical, while just reading over this guide tends to be more theoretical.

Note: The following steps are described in sufficient detail for someone experienced with the basic flow through the Synergy development process. If these steps are not familiar, refer to the first few chapters of the *SSP User's Manual* for a description of how to accomplish these steps.

To create and run the LVD HAL module application project, simply follow these steps:

1. Create a new Renesas Synergy project for the SK-S7G2 called LVD_HAL.
2. Select the **Threads** tab
3. Add two instances of the LVD HAL module to the HAL/Common thread.
4. Configure each instance of `r_lvd`.
5. Click on the **Generate Project Content** button.
6. Add the code from the supplied project file `hal_entry.c` or copy over the generated `hal_enrtry.c` file.
7. Build the code.
8. Connect to the host PC.
9. Start to debug the application.
10. Disconnect from the host PC and connect the variable voltage PSU.
11. Observe the LEDs as the voltage is varied between 3.3V > 2.7V.

10. LVD HAL Module Conclusion

This module guide has provided all the background information needed to select, add, configure, and use the module in an example project. Many of these steps were time consuming and error-prone activities in previous generations of embedded systems. The Renesas Synergy Platform makes these steps much less time consuming and removes the common errors, like conflicting configuration settings or incorrect selection of lower-level drivers. The use of high-level APIs (as demonstrated in the application project) illustrates additional development time savings by allowing work to begin at a high level and avoiding the time required in older development environments to use or, in some cases, create, lower-level drivers.

11. LVD HAL Module Next Steps

After you have mastered a simple LVD HAL module project, you may want to review a complex example. You may find that the configuring the LVD to assert a reset is a better fit for your target application, rather than using software polling and notification via an interrupt callback.

Additionally, the LVD application project detailed in this module guide illustrates the use of the LVD within a HAL-only application. ThreadX® RTOS-based implementations are typically more complex.

Other application projects and application notes that demonstrate LVD HAL use are listed in the References section at the end of this document.

12. LVD HAL Module Reference Information

SSP User Manual: Available in html format in the SSP distribution package and also as pdf from the Synergy Gallery.

Links to all the most up-to-date `r_lvd` module reference materials and resources are available on the Synergy Knowledge Base: https://en-us.knowledgebase.renesas.com/English_Content/Renesas_Synergy%E2%84%A2_Platform/Renesas_Synergy_Knowledge_Base/r_lvd_Module_Guide_Resources.

Website and Support

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

Technical Contact Details:

- America: https://renesas.zendesk.com/anonymous_requests/new
- Europe: https://www.renesas.com/en-eu/support/contact.html
- Japan: https://www.renesas.com/ja-jp/support/contact.html

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Aug 2, 2017	—	Initial Release

RENESAS

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, HAL II 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 6.0