

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 in 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 DAC8 HAL module is a high-level API for digital-to-analog conversion applications implemented on r_dac8. The DAC8 HAL module supports an 8-bit D/A converter (DAC8) peripheral on Synergy S128 MCUs.

Contents

1. DAC8 HAL Module Features	2
2. DAC8 HAL Module APIs Overview	2
3. DAC8 HAL Module Operational Overview	3
3.1 DAC8 HAL Module Important Operational Notes and Limitations	3
3.1.1 DAC8 HAL Module Operational Notes	3
3.1.2 DAC8 HAL Module Limitations	3
4. Including the DAC8 HAL Module in an Application	3
5. Configuring the DAC8 HAL Module	4
5.1 DAC8 HAL Module Clock Configuration	4
5.2 DAC8 HAL Module Pin Configuration	4
6. Using the DAC8 HAL Module in an Application	5
7. The DAC8 HAL Module Application Project	5
8. Customizing the DAC8 HAL Module for a Target Application	8
9. Running the DAC8 HAL Module Application Project	8
10. DAC8 HAL Module Conclusion	9
11. DAC8 HAL Module Next Steps	9
12. DAC8 HAL Module Reference Information	10
Revision History	12

1. DAC8 HAL Module Features

- Available on the Synergy S128 MCU
- 8-Bit D/A Converter with three channels
- Left-Justified or Right-Justified Input Data Format
- Synchronization with the Analog-to-Digital Converter (ADC) module
- Multiple Operation Modes
 - Normal
 - Real-Time (Event Link)
- Charge Pump Control

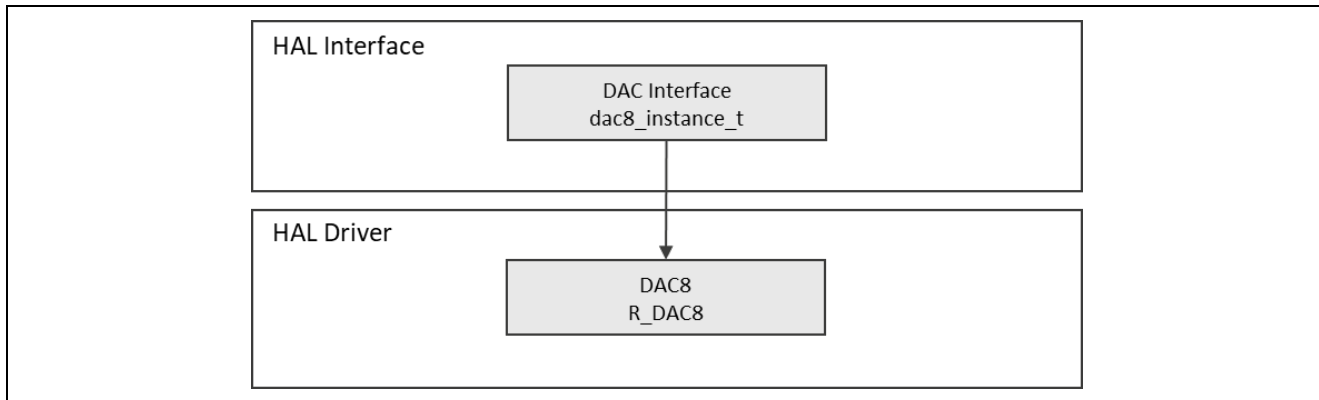


Figure 1 DAC8 HAL Module Block Diagram

2. DAC8 HAL Module APIs Overview

The DAC8 HAL module defines APIs for opening, closing, starting, stopping and writing to the DAC. The following table lists each of the APIs available with an example call and a short description. A listing of status return values follows the API summary table.

Table 1 DAC8 HAL Module API Summary

Function Name	Example API Call and Description
.open	g_dac8.p_api->open(g_dac8.p_ctrl, g_dac8.p_cfg) Initial Configuration.
.close	g_dac8.p_api->close(g_dac8.p_ctrl) Close the D/A Converter.
.write	g_dac8.p_api->write(g_dac8.p_ctrl, val) Write Sample value to the D/A Converter.
.start	g_dac8.p_api->start(g_dac8.p_ctrl) Start the D/A Converter if it has not been started yet.
.stop	g_dac8.p_api->stop(g_dac8.p_ctrl) Stop the D/A Converter if the converter is running.
.versionGet	g_dac8.p_api->versionGet(&version) Retrieve the API version with the version pointer.

Note: Review the *SSP User's Manual* API References for the associated module. The SSP defines operations, function data structures, typedefs, defines, API data, API structures, and function variables.

3. DAC8 HAL Module Operational Overview

The DAC8 HAL module configures the 8-bit D/A converter (DAC8) to output one of 256 voltage levels between the positive and negative reference voltages. The driver can be configured to accept the 8-bit output data in left-or-right-justified format in a 16-bit input data. The driver supports two modes for the DAC.

- Normal mode — D/A output is updated on writes to the data register.
- Real-Time (Event Link) — D/A output is updated on an Event Link event. While in this mode, the data register can be written at any time. An Event Link event triggers the start of conversion. For details, see the ELC Interface in the *SSP User’s Manual*.

To reduce the noise present in ADC readings, the driver’s synchronous anti-interference mode can be configured with the ADC module. This mode reduces conversion noise by disabling the DAC charge while the ADC is sampling. Check the applicable MCU hardware manual to determine whether this feature is supported.

For operation at a low AVCC voltage, the driver can enable or disable the hardware charge pump.

3.1 DAC8 HAL Module Important Operational Notes and Limitations

3.1.1 DAC8 HAL Module Operational Notes

The DAC8 channel output is enabled during `dac8_api_t::start()` and `dac8_api_t::write()`, and disabled during `dac8_api_t::stop()` and `dac8_api_t::close()`.

3.1.2 DAC8 HAL Module Limitations

- DAC8 uses the DAC8 peripheral available on the Synergy S128 MCU.
- DAC8 driver does not configure the output pins for analog output.
- DAC8 driver does not configure the ELC peripheral for real-time mode. You will need to configure the Event Link Controller in addition to enabling real-time mode in the DAC8 module.

See the latest *SSP Release Notes* for any additional limitations to operating this module.

4. Including the DAC8 HAL Module in an Application

This section describes how to include the DAC8 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 tasks, see the *SSP User’s Manual* to learn how to manage each of these important steps in creating SSP-based applications.

To add the DAC8 Driver to an application, use the stacks selection sequence in the following table. (The default name for the DAC8 is `g_dac8_0`. This name can be changed in the associated Properties window.)

Table 2 DAC8 HAL Module Selection Sequence

Resource	ISDE Tab	Stacks Selection Sequence
<code>g_dac8_0</code> DAC Driver on <code>r_dac8</code>	Threads	New Stack > Driver > Analog > DAC Driver on <code>r_dac8</code>

When the DAC8 HAL module on DAC is added to the thread stack as shown in the following figure, the configurator automatically adds any needed lower-level drivers. Modules with a Gray band are individual modules that stand alone.

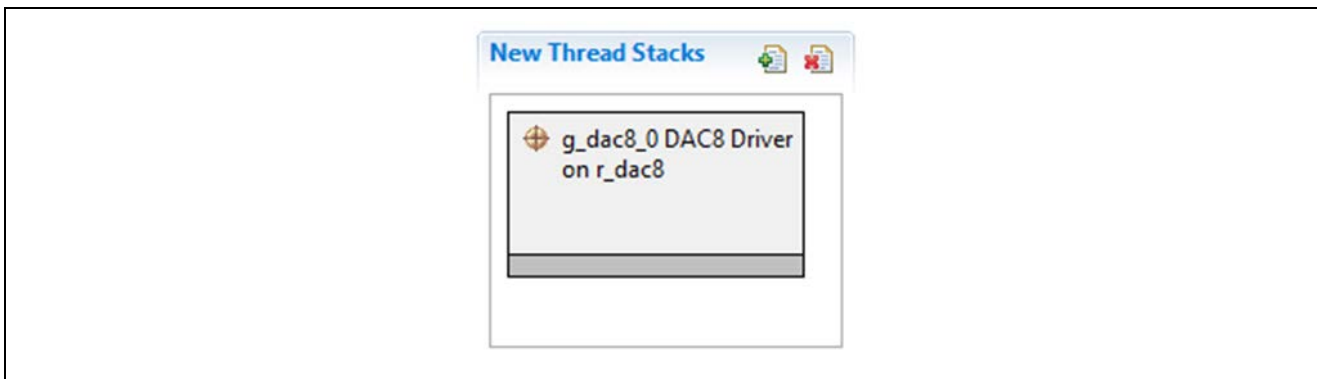


Figure 2 DAC8 HAL Module Stack

5. Configuring the DAC8 HAL Module

The DAC8 HAL module must be configured 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 with 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 3 Configuration Settings for the DAC8 HAL Module on r_dac

ISDE Property	Value	Description
Parameter Checking	BSP, Enabled, Disabled (Default: BSP)	Enable or disable the parameter error checking.
Name	g_dac8_0	Module name
Channel	0	Channel selection- three channels are supported.
Synchronize with ADC	Enabled, Disabled (Default: Disabled)	Choose whether to sync with the ADC module
Data Format	Right Justified, Left Justified (Default: Right Justified)	Data format selection
DAC Mode	Normal Mode, Real-time (Event Link) Mode (Default: Normal Mode)	DAC mode selection
Charge Pump Enabled (Requires MOCO active)	Enabled, Disabled (Default: Enabled)	Enable or disable the charge pump

Note: The example values and defaults are for a project using the Synergy S128 MCU. Other MCUs may have different default values and available configuration settings.

In some cases, settings other than the defaults for module properties can be desirable. For example, it might be useful to enable the output amplifier in some applications.

5.1 DAC8 HAL Module Clock Configuration

The DAC8 HAL module does not require any specific clock configuration.

5.2 DAC8 HAL Module Pin Configuration

To use the DAC8 HAL module, the port pins for the channels receiving the analog input must be set as analog pins in the pin configurator. The following table illustrates the method for selecting the pins within the SSP configuration window and the subsequent table illustrates an example selection for the DAC pins.

Table 4 Pin Selection Sequence for the DAC8 Driver on r_dac8

Resource	ISDE Tab	Pin Selection Sequence
DAC122	Pins	Select Peripherals > Analog:DAC12 > DAC122

Table 5 Pin Configuration Settings for the DAC8 Driver on r_dac8

Property	Value	Description
Module Name	DAC122	DAC peripheral module
Operation Mode	Enabled, Disabled (Default: Enabled)	DAC peripheral operation mode
DA	None, P011, P004 (Default: None)	DAC output pin

Note: The example values illustrate a project using the Synergy S128 MCU. Other Synergy MCUs and Synergy Kits may have pin configuration settings that differ in availability.

6. Using the DAC8 HAL Module in an Application

The typical steps in using the DAC8 HAL module in an application are:

1. Initialize the DAC8 HAL module using the `open` API.
2. Write a value using the `write` API.
3. Start a conversion using the `start` API.
4. Continue writing values as needed using the `write` API.
5. Stop conversion using the `stop` API.
6. Use the `close` API to power down the peripheral.

The following figure shows common steps in a typical operational flow:

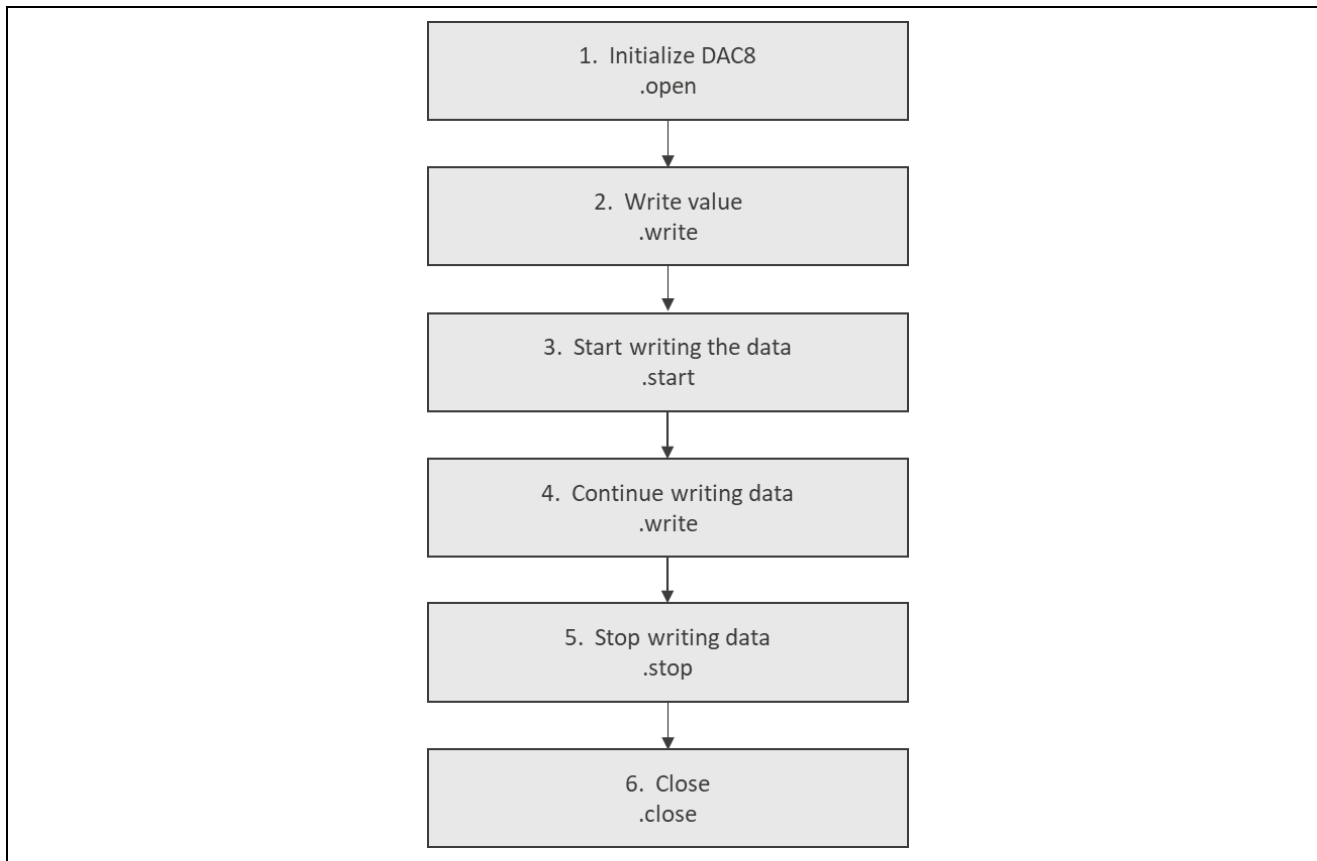


Figure 3 Typical DAC8 HAL Module Application Flow Diagram

7. The DAC8 HAL Module Application Project

The application project associated with this module guide demonstrates the typical steps the preceding figure shows in an example application. You may want to import and open the application project within the ISDE and view the configuration settings for the DAC8 HAL module. You can also read over the code in `dac_hal.c`, which is used to illustrate the DAC8 HAL module APIs in a complete design.

The application project demonstrates the typical use of the DAC8 HAL module APIs. The application project main part initializes the DAC and ADC HAL drivers and generates the square-like output with a constant value added. The result is continuously scanned by the ADC driver and these results are printed on the debug console using the common semi-hosting function. 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
e ² studio	6.2.1R20180620-1239 or later	Integrated Solution Development Environment
SSP	1.5.0 or later	Synergy Software Platform
IAR EW for Synergy	8.23.1 or later	IAR Embedded Workbench® for Renesas Synergy™
SSC	6.2.1R20180629	Synergy Standalone Configurator
DK-S128	v1.0	Starter Kit

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

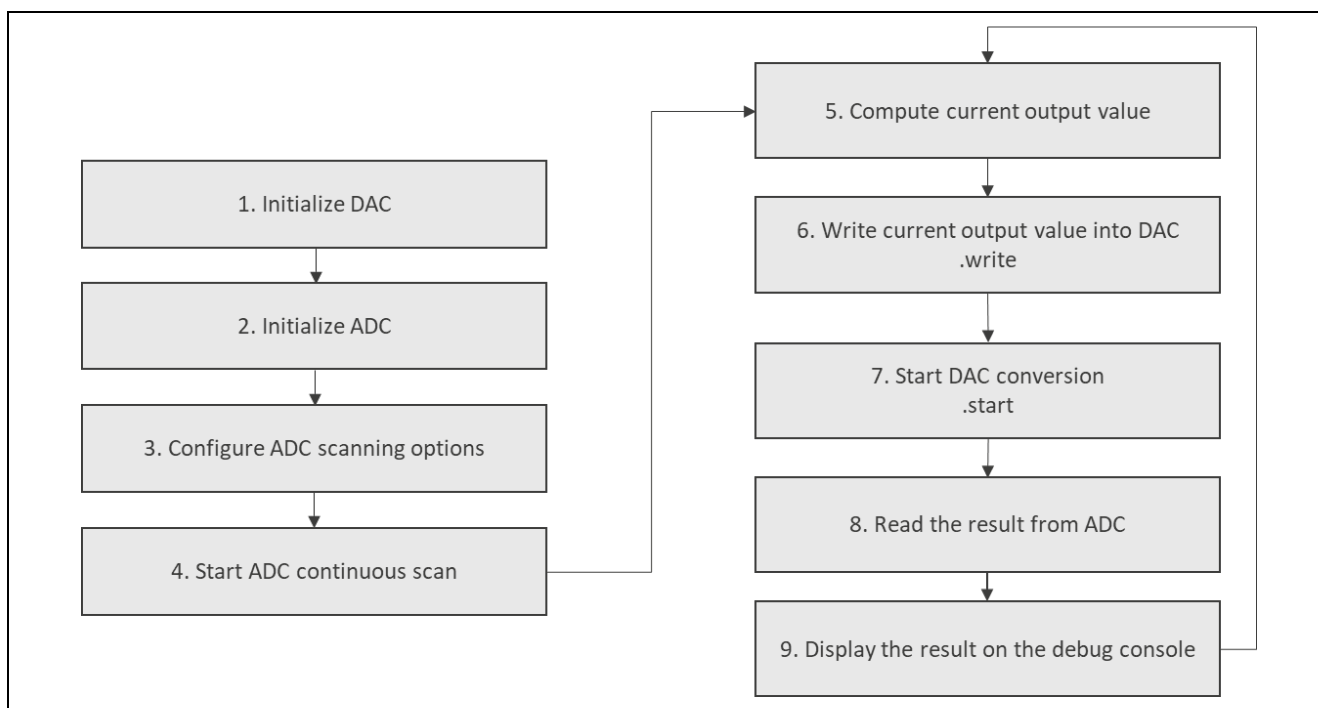


Figure 4 DAC8 HAL Module Application Project Flow Diagram

The `dac_hal.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 `dac_hal.c` has the header files which reference the DAC instance structure, CMSIS DSP library, and macro constant definitions. Then there is a code section which allows semi-hosting to display results using `printf()`. The next section is the function for getting current output value. It computes value of the square function. The returned value varies between the minimum (0) and the maximum (255) in DAC unit steps to generate square waveform.

The last section is the entry function for the main program control section. The DAC and ADC drivers are initialized using the open API. Next, the ADC driver is configured for continuous scanning and the scanning process is started. Inside the infinite while loop, the current output value to generate a square wave is computed and written using the write API. If it succeeds, the DAC8 conversion is started. After a one-hundred milliseconds delay, the value from the ADC is read. When the read operation completes successfully, the result is printed using semi-hosting. During any API calls, if the status return code is non-zero, it indicates an error has occurred and the interior while loop traps on the error, else the while loop functions are repeated.

Note: It is assumed that you are familiar with using `printf()` with the debug console in the Synergy Software Package. If you are unfamiliar with the debug console, See “*How do I Use Printf() with the Debug Console in the Synergy Software Package*” Knowledge Base article, available in the References section at the end of this document. Alternatively, you can see results using the watch variables in the debug mode.

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

Table 7 DAC8 HAL Module Configuration Settings for the Application Project

ISDE Property	Value Set
Parameter Checking	Default (BSP)
Name	g_dac8_0
Channel	2
Synchronize with ADC	Disabled
Data Format	Right Justified
DAC Mode	Normal Mode
Charge Pump Enabled (Requires MOCO active)	Disabled

Table 8 ADC HAL Module Configuration Settings for the Application Project

ISDE Property	Value Set
Parameter Checking	BSP
Name	g_adc0
Unit	0
Resolution	14-Bit (S3A7/S124 only) S128 supports 14-bit ADC
Alignment	Right
Clear after read	On
Mode	Continuous Scan
Channel Scan Mask: Channel 0	Used in Normal/Group A
Channel Scan Mask: Channel 1-27	Unused
Channel Scan Mask: Temperature Sensor	Unused
Channel Scan Mask: Voltage Sensor	Unused
Normal/Group A Trigger	Software
Group B Trigger (Valid only in Group Scan Mode)	ELC Event
Group Priority (Valid only in Group Scan Mode)	Group A cannot interrupt Group B
Add/Average Count	Disabled
Addition/Averaging Mask: Channel 0-27	Disabled
Addition/Averaging Mask: Temperature Sensor	Disabled
Addition/Averaging Mask: Voltage Sensor	Disabled
Sample and Hold Mask: Channel 0-2	Disabled
Sample Hold States (Applies only to the 3 channels selected)	24
Callback	NULL
Scan End Interrupt Priority	Disabled
Scan End Group B Interrupt Priority	Disabled

Use the following procedure to configure the analog pins for DAC and ADC modules:

1. Click on the **Pins** tab in Synergy Configurator.
2. Delete `g_bsp_pin_cfg` in the Generate data window.
3. Uncheck the **Generate data** box. The Generate data field becomes grayed out.
4. Hit **Generate Project Content**.
5. Click the drop-down arrow for the pin configuration.
6. Change the pin configuration setting from `S128-DK.pincfg` to `R7FS128783A01CFM.pincfg`.
7. Check Generate data and the input `g_bsp_pin_cfg` in the empty field.
8. Hit **Generate Project Content**.
9. Browse to Peripherals -> Analog DAC12 -> DAC122.
10. Change operation mode to **Enable** and select pin **P004**.

11. Browse to Peripherals -> Analog ADC -> ADC0.
12. Change operation mode to **Custom** and select pin **P000** for AN00.

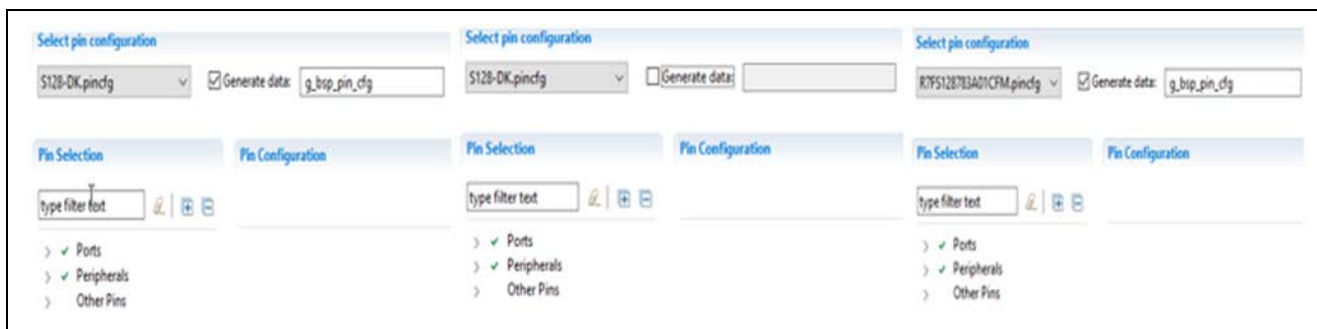


Figure 5 Steps to Configure Analog Pins for DAC8 and ADC

In addition, a hardware connection between ADC and DAC pins is needed. The ADC driver is configured to use channel 0 (P000 pin). The DAC driver outputs the written values to channel 2 (P004). Switch SW2 is connected to P004: Disable switch SW2 by turning OFF pin 2 (SW2) in the DIP switch (SW5).

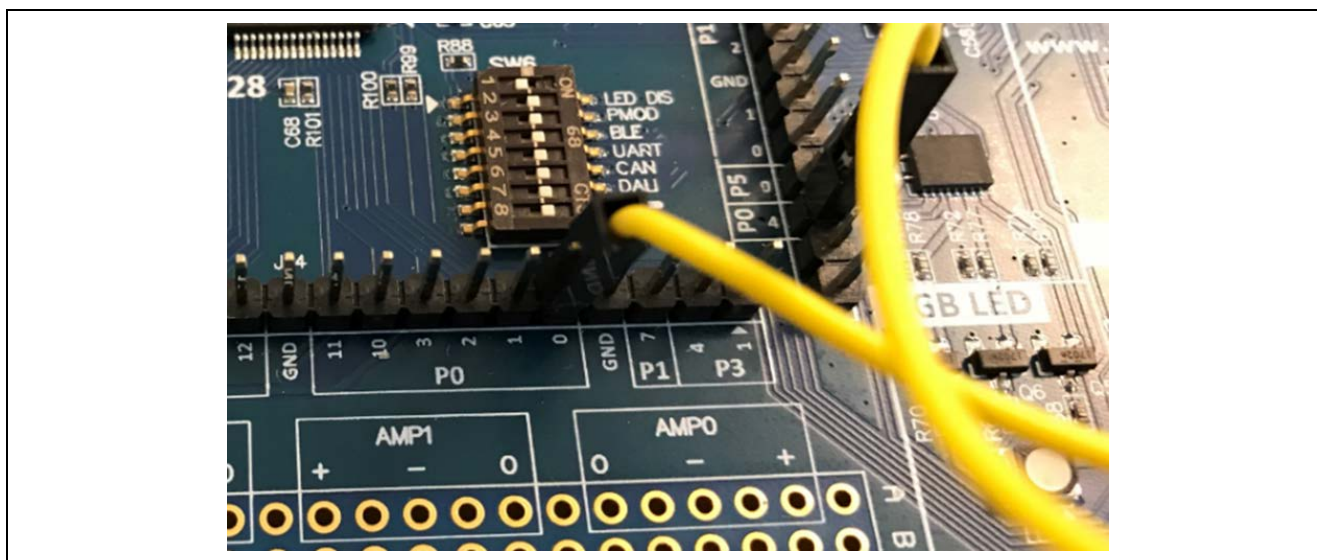


Figure 6 Connecting ADC Input Pin (P00) and DAC output pin (P004)

8. Customizing the DAC8 HAL Module for a Target Application

Some configuration settings are normally changed by the developer from those shown in the application project. You can easily change the channel, or enable the output amplifier for the DAC, and configure the DAC to operate in Normal mode or ELC mode. For example, The DAC application project uses channel 2 (P004), which needs to be connected to the ADC input pin (P000 for channel 0), and you can change this setting.

9. Running the DAC8 HAL Module Application Project

To run the DAC8 HAL Module application project and to see it executed on a target kit, you can import it into your ISDE, compile and run debug. Refer to the *Renesas Synergy™ Project Import Guide* for instructions on importing the project into e² studio or IAR embedded workbench and building/running the application.

To implement the DAC8 HAL Module application in a new project, follow the 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.

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

To create and run the DAC8 application project, use the following steps:

1. Create a new Renesas Synergy project for the S7G2-SK called DAC_HAL_MG_AP.
2. Select the **Threads** tab.

3. Add the DAC driver to the HAL/Common thread.
4. Add the ADC driver to the HAL/Common thread.
5. Configure the blocks according to the preceding tables.
6. Click **Generate Project Content**.
7. Add the code from the supplied project file "hal_entry.c", "dac_hal.h" and "dac_hal.c" or copy over generated "hal_entry.c", "dac_hal.h" and "dac_hal.c" files.
8. Connect ADC input pin (P000) with DAC output pin (P004).
9. Connect to the host PC using the USB cable (use J12 DEBUG_USB connector).
10. Start to debug the application.
11. Probe at point P004 to get a square wave waveform.

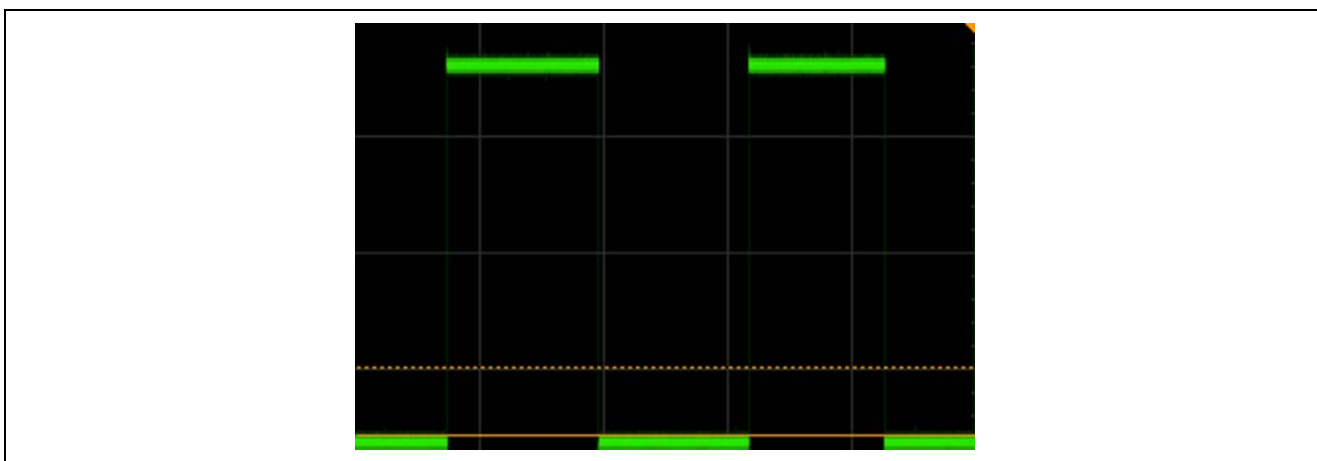


Figure 7 Square Wave at P004

```

DAC output(number of steps): 0 ADC input (number of steps): 62
DAC output(number of steps): 255 ADC input (number of steps): 16255
DAC output(number of steps): 0 ADC input (number of steps): 66
DAC output(number of steps): 255 ADC input (number of steps): 16250
DAC output(number of steps): 0 ADC input (number of steps): 74
DAC output(number of steps): 255 ADC input (number of steps): 16257
DAC output(number of steps): 0 ADC input (number of steps): 70
DAC output(number of steps): 255 ADC input (number of steps): 16254
DAC output(number of steps): 0 ADC input (number of steps): 66
DAC output(number of steps): 255 ADC input (number of steps): 16256
DAC output(number of steps): 0 ADC input (number of steps): 70
DAC output(number of steps): 255 ADC input (number of steps): 16251
DAC output(number of steps): 0 ADC input (number of steps): 65
DAC output(number of steps): 255 ADC input (number of steps): 16249
DAC output(number of steps): 0 ADC input (number of steps): 72
DAC output(number of steps): 255 ADC input (number of steps): 16251
    
```

Figure 8 Example Output from DAC HAL Module Application Project

10. DAC8 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 the 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. DAC8 HAL Module Next Steps

After you have mastered a simple DAC8 module project, you may want to review a more complex example. Other application projects and application notes that demonstrate DAC8 HAL use can be found as described in the References section at the end of this document.

You may find that the Audio Playback Framework is a better fit for your target application. The Audio Playback Framework is a higher level framework and can use DAC or I²S as an output.

12. DAC8 HAL Module Reference Information

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

Links to all the most up-to-date r_dac8 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_DAC8_Module_Guide_References](https://en-us.knowledgebase.renesas.com/English_Content/Renesas_Synergy%E2%84%A2_Platform/Renesas_Synergy_Knowledge_Base/R_DAC8_Module_Guide_References).

Website and Support

Visit the following vanity URLs to learn about key elements of the Synergy Platform, download components and related documentation, and get support.

Synergy Software	renesassynergy.com/software
Synergy Software Package	renesassynergy.com/ssp
Software add-ons	renesassynergy.com/addons
Software glossary	renesassynergy.com/softwareglossary
Development tools	renesassynergy.com/tools
Synergy Hardware	renesassynergy.com/hardware
Microcontrollers	renesassynergy.com/mcus
MCU glossary	renesassynergy.com/mcuglossary
Parametric search	renesassynergy.com/parametric
Kits	renesassynergy.com/kits
Synergy Solutions Gallery	renesassynergy.com/solutionsgallery
Partner projects	renesassynergy.com/partnerprojects
Application projects	renesassynergy.com/applicationprojects
Self-service support resources:	
Documentation	renesassynergy.com/docs
Knowledgebase	renesassynergy.com/knowledgebase
Forums	renesassynergy.com/forum
Training	renesassynergy.com/training
Videos	renesassynergy.com/videos
Chat and web ticket	renesassynergy.com/support

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Sep 12, 2018	-	Initial release
1.01	Nov 12, 2018	-	Updated for 1.5.0

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

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.
2. 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 impractical, you are responsible for evaluating the safety of 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)



SALES OFFICES

Renesas Electronics Corporation

<http://www.renesas.com>

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

Renesas Electronics Corporation

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

Renesas Electronics America Inc.

1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

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: +44-1628-651-700

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, 200333 P. R. China
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 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.

17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338