

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 DAC HAL module is a high-level API for digital-to-analog conversion applications and is implemented on r_dac. The DAC HAL module supports a dual-channel 12-bit D/A converter (DAC12) peripheral on Synergy MCUs.

Contents

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

1. DAC HAL Module Features

This module configures the dual-channel 12-bit D/A Converter (DAC12) to output one of 4096 voltage levels between the positive and negative reference voltages. The module includes configuration settings to:

- Set either a left-justified or right-justified 12-bit value format for the 16-bit input data registers
- Enable or disable output amplifiers
- Operate in synchronous anti-interference mode with the Analog-to-Digital Converter (ADC) module.

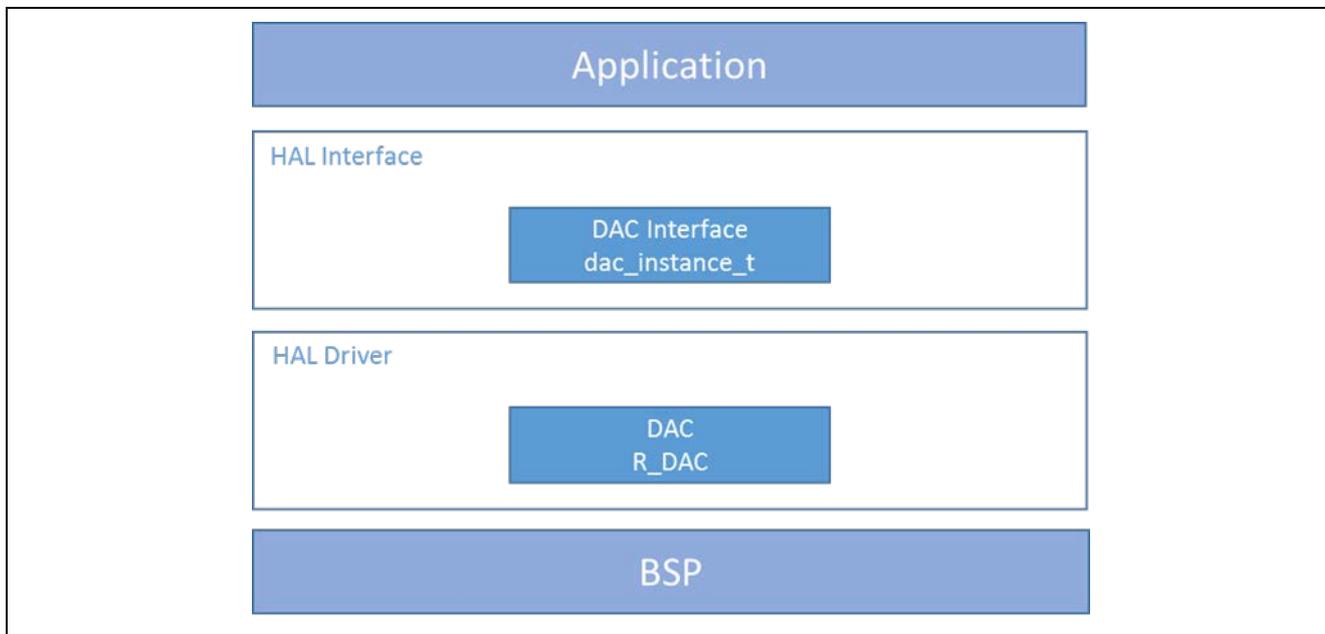


Figure 1 DAC HAL Module Block Diagram

2. DAC HAL Module APIs Overview

The DAC HAL module defines APIs to open, close, start, stop and write 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 DAC HAL Module API Summary

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

Table 2 Status Return Values

Name	Description
SSP_SUCCESS	API Call Successful.
SSP_ERR_NOT_OPEN	Unit is not open.
SSP_ERR_ASSERTION	Wrong parameter.
SSP_ERR_IN_USE	DAC resource is locked.

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

3. DAC HAL Module Operational Overview

The DAC HAL module configures the dual-channel 12-bit D/A converter (DAC12) to output one of 4096 voltage levels between positive and negative reference voltages. The module can be used to configure the 12-bit output to left-or-right-justified format for 16-bit input data registers. The DAC HAL module can also enable or disable output amplifiers or operate in synchronous anti-interference mode with the ADC HAL module.

3.1 DAC HAL Module Important Operational Notes and Limitations

3.1.1 DAC HAL Module Operational Notes

The DAC HAL module requires the following initialization steps:

- DAC module stop-bit cleared to zero.
- DAC channel output-enable set to one.

The DAC module stop-bit is cleared to zero at the time of an open call when the driver's instance counter is zero. The driver's instance counter is initialized to zero, incremented when a channel open returns successfully, and decremented when a channel close is called. The DAC module stop-bit is set to one when the driver's instance counter is decremented to zero on a close call.

The DAC channel output-enable is set to one when a channel write is called the first time after open was called successfully. The open call writes a zero to the `dac_ctrl_t` structure element `channel_started`. When write is called with `channel_started` cleared to zero, the DAC output-enable bit for that channel is set to one. The DAC output-enable for the channel is cleared to zero on close and stop calls.

3.1.2 DAC HAL Module Limitations

- Pin configuration is not implemented for the DAC HAL module. Currently, DA0 and DA1 outputs are enabled by the reset values in the pin configuration control register's ASEL field.
- Voltage reference selection for the DAC HAL module is not implemented. Currently, no reference is selected by the reset values in the D/A VREF control register (DAVREFCR) which is a valid condition.
- Configuration of DAC input-event triggering for conversion is not currently implemented. The default reset value (zero) of the control register DAE bit allows individual triggering for each channel.
- Event signal-input for synchronization of the DAC HAL module conversions is not currently implemented.

Refer to the latest SSP Release Notes for operational limitations related to this module.

4. Including the DAC HAL Module in an Application

This section describes how to include the DAC HAL module in an application using the SSP configurator.

Note: This section assumes 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 DAC Driver to an application, use the stacks selection sequence in the following table. (The default name for the DAC is `g_dac0`. This name can be changed in the associated Properties window.)

Table 3 DAC HAL Module Selection Sequence

Resource	ISDE Tab	Stacks Selection Sequence
g_dac0 DAC Driver on r_dac	Threads	New Stack > Driver > Analog > DAC Driver on r_dac

When the DAC 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.

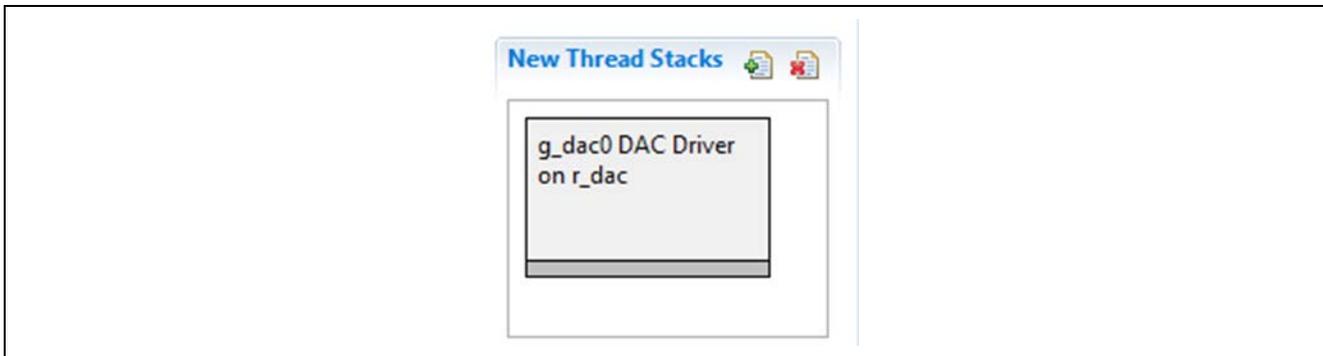


Figure 2 DAC HAL Module Stack

5. Configuring the DAC HAL Module

The DAC HAL module must be configured by the user for the desired operation. The SSP configuration window will automatically identify (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. Furthermore, 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 will include 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 4 Configuration Settings for the DAC HAL Module on r_dac

ISDE Property	Value	Description
Parameter Checking	BSP, Enabled, Disabled (Default: BSP)	Enables or disables the parameter checking.
Name	g_dac0	Module Name.
Channel	0,1	Set to 0 for output DA0 or 1 for output DA1.
Synchronize with ADC	Enabled, Disabled (Default: Disabled)	Set to true for anti-interference synchronization with the Analog-to-Digital Converter (ADC) Module. Set to false if the power supply interference between the analog modules is not a problem, or if the asynchronous conversion by the DAC Module is desired.
Data Format	Right Justified, Left Justified (Default: Right Justified)	Set to zero if 12-bit data values are loaded in bits 11 through 0, or right justified. Set to one, if 12-bit data values are loaded in bits 15 through 4, or left justified.
Output Amplifier	Enable, Disable (Default: Disable)	Set to true, if output amplifier hardware function is desired. Set to false to bypass output amplifier hardware function.

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

In some cases, settings other than the defaults can be desirable. For example, it might be useful to select different events, depending on the operation desired.

5.1 DAC HAL Module Clock Configuration

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

5.2 DAC HAL Module Pin Configuration

To use the DAC 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 lists a method to select pins within the SSP configuration window. The subsequent table illustrates configuration settings for DAC pins.

Table 5 Pin Selection Sequence for the DAC Driver on r_dac

Resource	ISDE Tab	Pin Selection Sequence
DAC	Pins	Select Peripherals > Analog: DAC12 > DAC120

Table 6 Pin Configuration Settings for the DAC Driver on r_dac

Property	Value	Description
Module Name	DAC120	DAC Peripheral Module.
Operation Mode	Enabled, Disabled (Default: Enabled)	DAC Peripheral operation mode.
DA0	None, DA0 (Default: None)	DAC Output Pin.
DA1	None, DA1 (Default: None)	DAC Output Pin.

Note: The example settings illustrate a project using the Synergy S7G2 and SK-S7G2 Kits. Other Synergy Kits and Synergy MCUs may have pin configuration settings that differ in availability.

6. Using the DAC HAL Module in an Application

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

1. Initialize the DAC HAL module using the `open` API.
2. Write a data value using the `write` API.
3. Start writing data using the `start` API.
4. Continue writing data values as needed using the `write` API.
5. Stop writing data using the `stop` API.
6. Close the DAC HAL module using the `close` API.

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

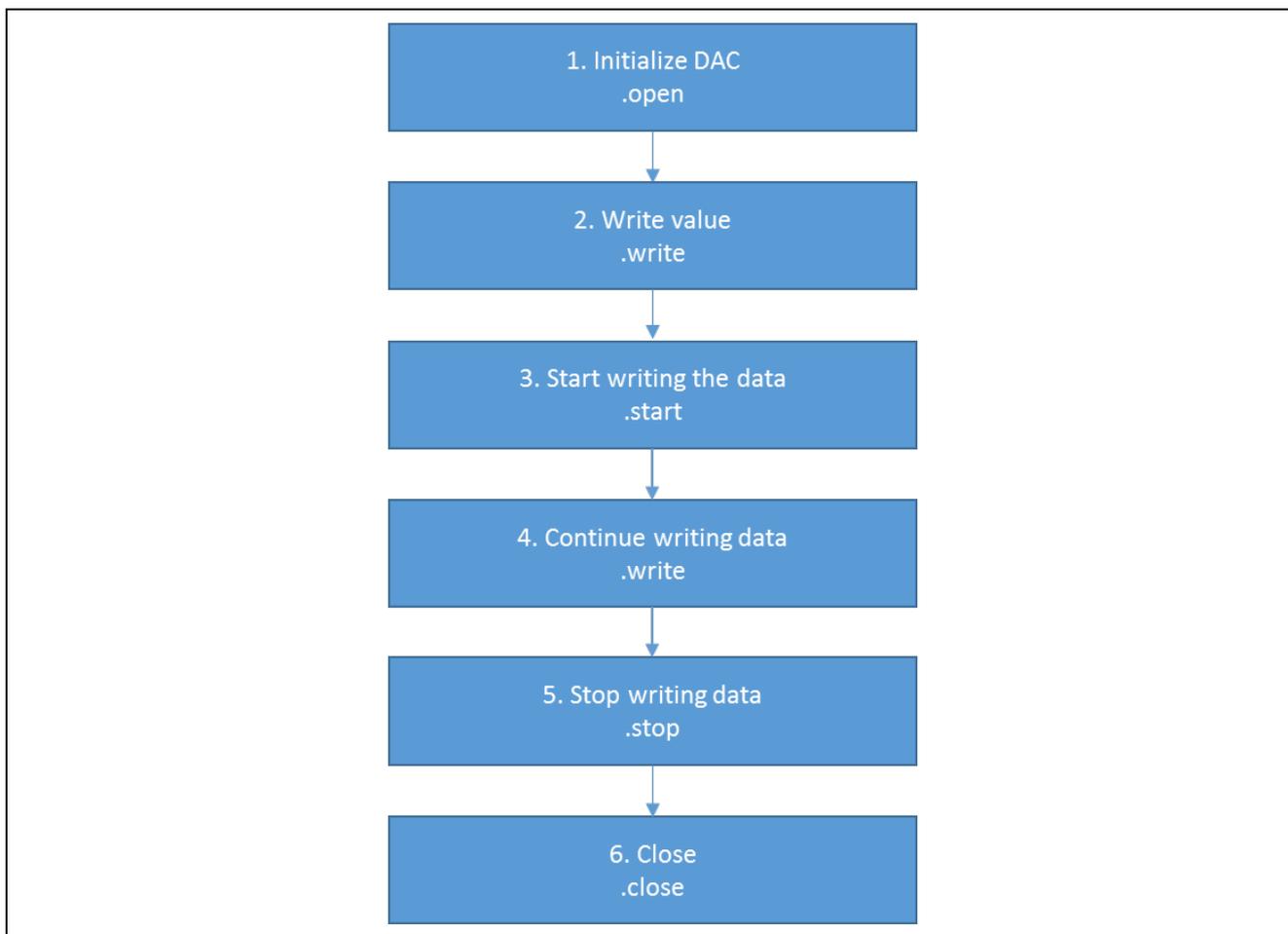


Figure 3 Typical DAC HAL Module Application Flow Diagram

7. The DAC HAL Module Application Project

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

The application project demonstrates the typical use of the DAC HAL module APIs. The application project initializes the DAC, AGT and ADC HAL modules, and periodically generates a sine-like output with a constant value added. The result is continuously scanned by the ADC HAL module and is 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 7 Software and Hardware Resources Used by the Application Project

Resource	Revision	Description
e ² 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 illustrates a simple application project flow diagram.

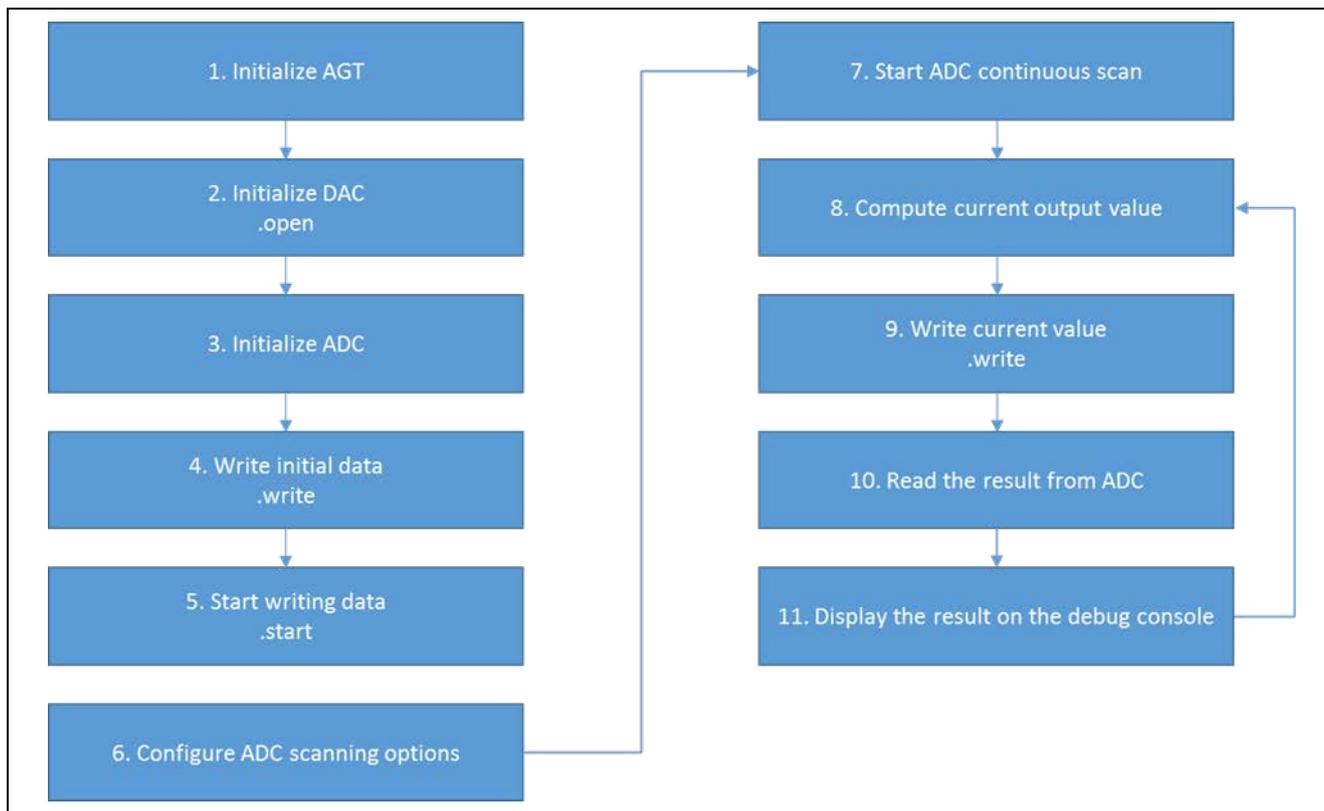


Figure 4 DAC HAL Module Application Project Flow Diagram

The `dac_hal_demo.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_demo.c` has the header files which reference the DAC instance structure, CMSIS DSP library, and macro-constant definitions. There is a code section which allows semi-hosting to display results using `printf()`. The next section is the function for getting the current output value. It computes the value of the sine function for the current timer’s counter value. The returned value is scaled. The sinusoidal waveform varies between the minimum (0) and the maximum (4095) in DAC units.

The last section is the entry function for the main program-control section. The AGT, DAC and ADC drivers are initialized using the `open` API. The DAC HAL module is then set to output the lowest level possible using the `write` API and is started using the `start` API. Secondly, the ADC HAL module is configured for continuous scanning and is started. Inside the infinite while loop, the current output value is computed and written using the `write` API. If it succeeds, after a delay of one hundred milliseconds, the value from the ADC is read. When the read operation is successfully completed, the result is printed using semi-hosting. If the status return code is non-zero, an error has occurred and the interior while loop traps on the error and the while loop functions are repeated.

Note: This description assumes you are familiar with using `printf()` with the debug console in the Synergy Software Package. If you are unfamiliar with this, refer to, *How do I Use Printf() with the Debug Console in the Synergy Software Package*, Knowledge Base article, available as described in the References section at the end of this document. Alternatively, the user can see results via the watch variables in the debug mode.

A few key properties are configured in this application project to support the required operations and 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 8 DAC HAL Module Configuration Settings for the Application Project

ISDE Property	Value Set
Parameter Checking	Default (BSP)
Name	g_dac
Channel	0
Synchronize with ADC	Disabled
Data Format	Right Justified
Output Amplifier	Disable

Table 9 ADC HAL Module Configuration Settings for the Application Project

ISDE Property	Value Set
Name	g_doc_overflow
Event	Addition overflow
Callback	doc_overflow_callback
DOC Interrupt Priority	Disabled

Table 10 AGT HAL Module Configuration Settings for the Application Project

ISDE Property	Value Set
Parameter Checking	Default (BSP)
Name	g_timer
Mode	Periodic
Period Value	1
Period Unit	Hertz
Auto Start	True
Count Source	LOCO
AGTO Output Enabled	False
AGTIO Output Enabled	False
Output Inverted	False
Callback	NULL
Interrupt Priority	Disabled

In addition, a hardware connection between ADC and DAC pins is needed. The ADC HAL module is configured to use channel 0 (P00 pin) and the DAC HAL module outputs write values to channel 0 (P014).

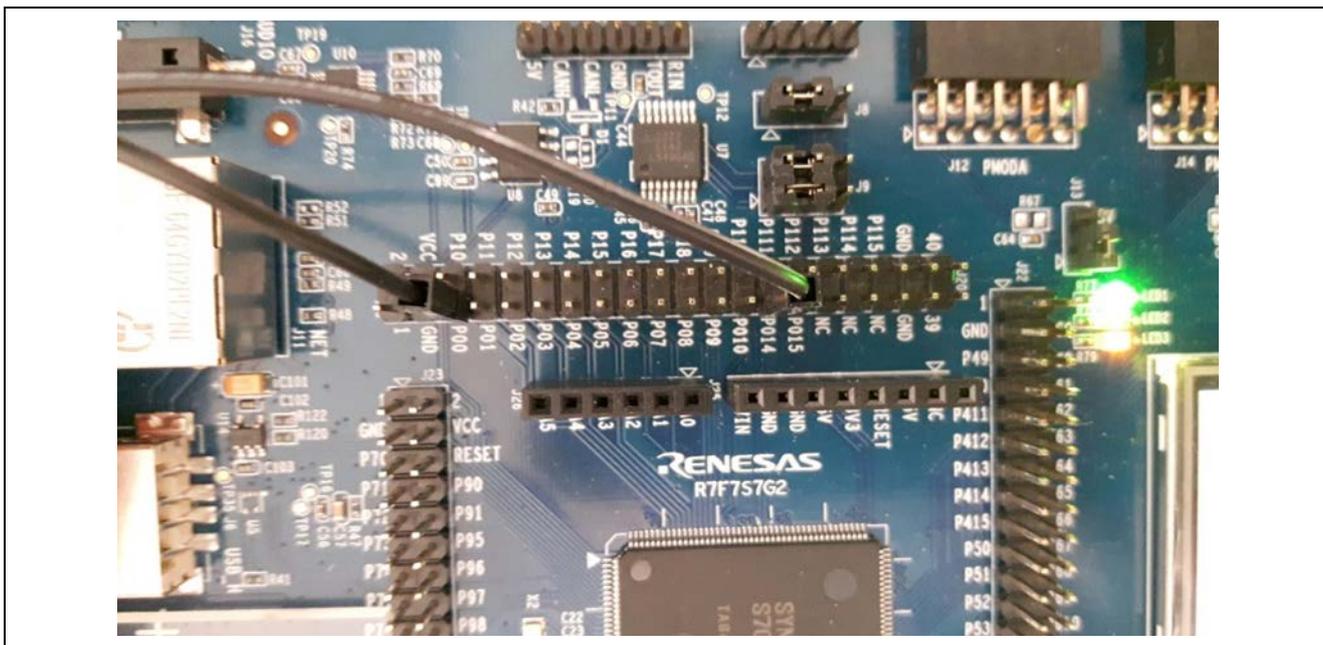


Figure 5 Connecting ADC input pin (P00) and DAC output pin (P014)

8. Customizing the DAC HAL Module for a Target Application

Some configuration settings are changed by the developer from those shown in the application project. For example, the user can easily change the channel or enable the output amplifier for the DAC. The DAC application project uses channel 0 (P401), a channel that needs to be connected with the ADC input pin (P000 for channel 0), with the associated DAC pin configured as an analog output.

9. Running the DAC HAL Module Application Project

To run the DAC 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 Synergy Project Import Guide for instructions on importing the project into e² studio or IAR EW for Synergy and building/running the application.

To implement the DAC 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 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 DAC 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 HAL module to the HAL/Common thread.
4. Add the ADC HAL module to the HAL/Common thread.
5. Add the AGT (timer) HAL module to the HAL/Common thread.
6. Configure the blocks according to the preceding tables.
7. Click the **Generate Project Content**.
8. Add the code from the supplied project file `hal_entry.c`, `dac_hal_demo.h` and `dac_hal_demo.c` or copy over generated `hal_entry.c`, `dac_hal_demo.h`, and `dac_hal_demo.c` files.
9. Connect the ADC input pin (P00) with the DAC output pin (P014).
10. Connect to the host PC using the USB cable (use J19 DEBUG_USB connector).
11. Start to debug the application.
12. The output can be viewed in the Renesas Debug Console.

```
Renesas Debug Virtual Console
DAC input: 3686 ADC output: 3651
DAC input: 3580 ADC output: 3552
DAC input: 1022 ADC output: 1024
DAC input: 103  ADC output: 129
DAC input: 2286 ADC output: 2273
```

Figure 6 Example Output from DAC HAL Module Application Project

10. DAC 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. DAC HAL Module Next Steps

After you have mastered a simple DAC module project, you may want to review a more complex example. Other application projects and application notes that demonstrate DAC 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. DAC 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_dac 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_dac_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>

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	May 19, 2017	—	Initial Release
1.01	Jul 31, 2017	7	Update to hardware and software Resources table

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 disputes 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, drawing, chart, program, algorithm, 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 otherwise misappropriate 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, copy or otherwise misappropriation of Renesas Electronics products.
 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; and 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.
Renesas Electronics products are neither intended nor 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 damages (space and 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 third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
 6. When using the 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 radiation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions or failure or accident arising out of the use of Renesas Electronics products beyond 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. Further, Renesas Electronics products are not subject to radiation resistance design. Please ensure to implement safety measures to guard them against the possibility of bodily injury, injury or damage caused by fire, and social damage in the event of 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 by your own responsibility as warranty for your products/system. Because the evaluation of microcomputer software alone is very difficult and not practical, please evaluate 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. Please investigate applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive carefully and sufficiently and use 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 not use Renesas Electronics products or technologies for (1) any purpose relating to the development, design, manufacture, use, stockpiling, etc., of weapons of mass destruction, such as nuclear weapons, chemical weapons, or biological weapons, or missiles (including unmanned aerial vehicles (UAVs)) for delivering such weapons, (2) any purpose relating to the development, design, manufacture, or use of conventional weapons, or (3) any other purpose of disturbing international peace and security, and you shall not sell, export, lease, transfer, or release Renesas Electronics products or technologies to any third party whether directly or indirectly with knowledge or reason to know that the third party or any other party will engage in the activities described above. When exporting, selling, transferring, etc., Renesas Electronics products or technologies, you shall comply with any applicable export control laws and regulations promulgated and administered by the governments of the countries asserting jurisdiction over the parties or transactions.
 10. Please acknowledge and agree that you shall bear all the losses and damages which are incurred from the misuse or violation of the terms and conditions described in this document, including this notice, and hold Renesas Electronics harmless, if such misuse or violation results from your resale or making Renesas Electronics products available any third party.
 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 majority-owned subsidiaries.
(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.3.0-1 November 2016)



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