Renesas RA Family

16-Bit Analog-to-Digital Converter Operation for RA6T2

Introduction

This application note describes the operation of the Analog-to-Digital Converter (ADC) on the RA6T2, with a focus on the conversion methods that enable 16-bit depth resolution. The note begins with a brief background on oversampling techniques to increase A/D resolution, then dives into the specifics of the oversampling features built into the ADC on the RA6T2. The application note covers the key configurations for capturing data in 16-bit depth mode and details the important functions for ensuring proper operation.

The sample code folder provided with this application note contains 2 projects: one project demonstrates operating the ADC with 16-bit depth in Oversampling Mode, and the other project demonstrates operating the ADC with 12-bit depth in SAR Mode for a performance comparison.

Target Device

RA6T2

Required Resources

To build and run the example projects accompanying this application note, you will need the following:

Development tools and software:

- e² studio IDE, version 2024-07 (24.7.0)
- RA Family Flexible Software package (FSP) v5.5.0
- J-Link RTT Viewer, version 7.98b

The FSP and e² studio are bundled in a downloadable platform installer available on Renesas' website at: renesas.com/ra/fsp

Hardware:

- RA6T2 (240MHz Arm® Cortex®-M33 TrustZone®, High Real-time Engine for Motor Control)
- MCK-RA6T2 (RTK0EMA270S00020BJ - Renesas Flexible Motor Control Kit for RA6T2 MCU Group)
- USB-C to USB-A cable
- Host PC running Windows® 10

Prerequisites and Intended Audience

This application note assumes you have some experience with the Renesas e² studio IDE and RA Family Flexible Software Package (FSP). Before you perform the procedures in this application note, follow the procedure in the *FSP User Manual* to build and run the Blinky project. Doing so enables you to become familiar with e² studio and the FSP and validates that the debug connection to your board functions properly. Additionally, this application note assumes that you have some theoretical background on analog topics relating to A/D conversion. The intended audience are users who want to develop applications with 16-bit ADC module using the RA6T2.

Contents

1. Overview of A/D Converters

Analog-to-digital converters (A/D converters, ADCs) are an integral part of data acquisition systems (DAQs) that function by capturing analog signals and converting them into discrete digital signals. ADCs convert analog voltages into numbers for a processor to operate on the values as needed: to store, display or further analyze the captured digital signal.

1.1 Characterizing an A/D Converter

A/D converters are generally characterized by three inherent qualities: the **input voltage range**, the **resolution** of the discrete values, and the **conversion rate** .

The **input voltage range** defines the range of acceptable analog input voltages that the ADC can convert to digital values. The input voltage range's maximum value is dependent on the reference voltage that the ADC system uses; typically the upper bound on the input voltage is equal to the value of the internal reference voltage. In the RA6T2, VREFH0 is the input analog reference voltage supply and is defined to lie in the range [2.7 V, 3.6 V] according to Electrical Characteristics.

The **resolution** of an A/D converter refers to the smallest incremental voltage measure detected, which causes a change in the value of the converted digital output. The resolution of an ADC is determined by the number of bits used to store the digital converted value. For an n-bit resolution ADC, 2^n values can be represented digitally.

The **conversion rate** (also referred to as **sampling rate**) describes the amount of time, recorded by the number of clock cycles it takes to convert the analog input to its digital representation. Typically, this value is expressed in Hertz as the number of A/D readings that can be completed each second. The conversion rate is especially important for analyzing acceptable AC signal input frequency rates according to Nyquist rules, to avoid unwanted artifacts like aliasing.

1.2 Analog Input Types

Most analog signals are either transmitted in one wire as a single analog voltage or in two wires as a difference between two analog voltages. The ADC on the RA6T2 supports both **single-ended input** and **differential input**.

1.2.1 Single-Ended Input

Single-ended input ADCs convert the difference between the voltage of the analog signal source and the analog reference ground voltage. Single-ended input is the most cost efficient in implementation, but the signals are sensitive to noise from electromagnetic interference.

1.2.2 Differential Input

Differential input ADCs convert the difference voltage between two complementary signals: a non-inverting and inverting input. Differential input is more costly to implement but has a higher performance and robustness against noise in the signals.

1.3 Types of ADCs

There are multiple hardware implementations capable of converting an input voltage to a digital representation. The variations in ADC implementation result in variations of the conversion's characteristics, so your application's requirements guide and influence which ADC type is optimal for that particular use case.

In the current industry, there are five major types of A/D converters: **successive approximation**, **delta-sigma**, **dual slope**, **pipelined**, and **flash** ADCs. When looking for an ADC to function as part of a data acquisition system, the two relevant types are the successive approximation ADC and delta-sigma ADC. The RA6T2 A/D converters have a **hybrid architecture** with features of both the successive approximation type and delta-sigma modulation type.

The following sections provide an overview of the main characteristics and differences of the ADC types, with a greater focus on the successive approximation and delta-sigma modulation ADCs.

1.3.1 Successive Approximation Register ADC

The **Successive Approximation Register** (**SAR**) type ADC converts a continuous analog voltage signal to a discrete digital representation by utilizing a binary search through all possible quantization levels before converging on the digital output. The input analog signal is sampled and held, and the value of each bit of the digital representation is approximated successively, by comparing the input voltage to a reference voltage.

A basic SAR type A/D converter is shown in the following schematic:

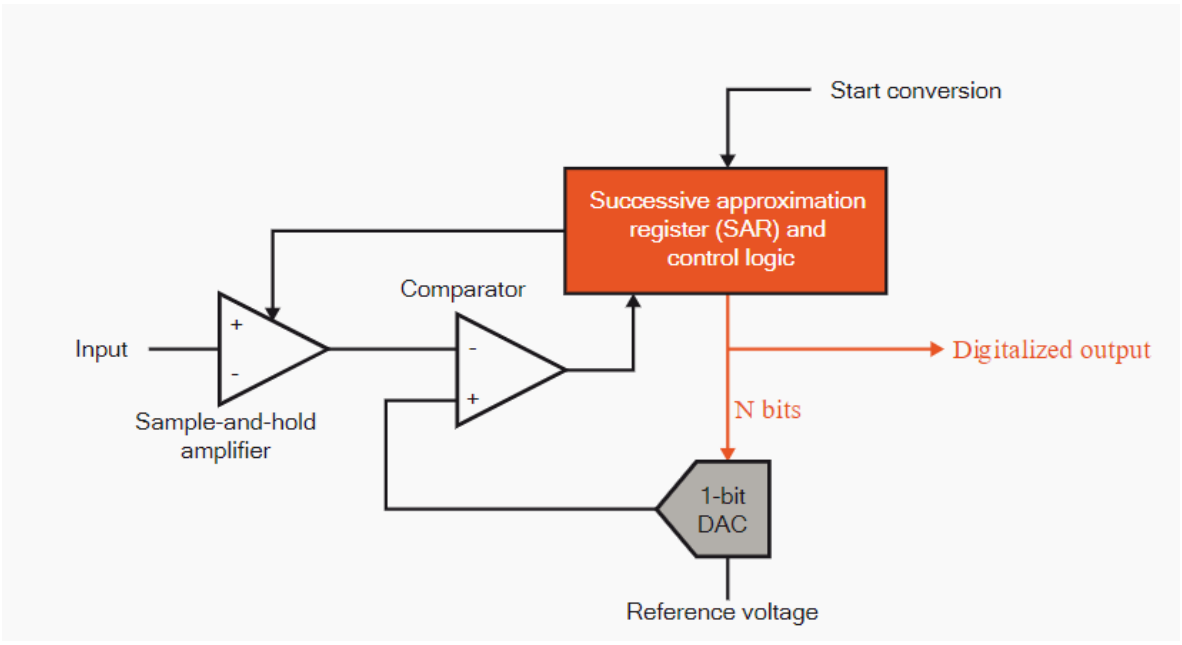


Figure 1. Circuit Block Diagram of SAR ADC

1.3.1.1 Key Features and Limitations

- Require N clock periods for N-bit SAR
- Low power and physically small
- Lower sampling rates for higher resolutions
- Limited resolution due to limits of comparator
- Circuit size increases as resolution increases

1.3.2 Delta-Sigma Modulation ADC

The **Delta-Sigma Modulation** (**DSM**) type ADC first encodes a continuous analog voltage signal to a stream of pulses that represent the change (delta) of the signal. The accuracy of the modulation is improved by then passing the digital output to a 1-bit DAC and summing (sigma) the resulting analog signal to the input signal. The delta-sigma modulation ADC type takes advantage of filtering techniques that improve amplitude axis resolution and reduce the high-frequency noise inherent in SAR designs.

The design of a delta-sigma modulation type A/D converter is shown in the following schematic:

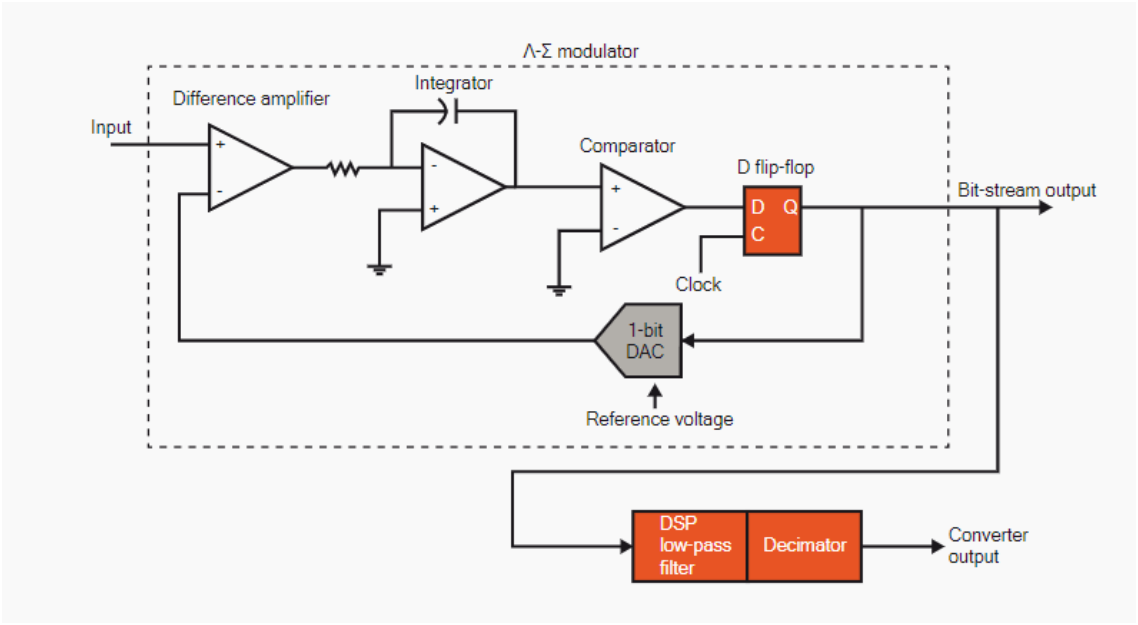


Figure 2. Circuit Block Diagram of Delta-Sigma ADC

1.3.2.1 Key Features and Limitations

- Oversampling in the design reduces quantization noise
- Resolution is highest amongst all types
- Sample rates are lower as compared to SAR type
- High clock speed is required to oversample the data
- Multiple samples must be converted per each ADC data value (oversampling) to achieve a higher signal quality
- Best for high-resolution and low-frequency applications

1.3.3 Additional ADCs

The other types of common ADCs are better suited for non-DAQ (non-Data Acquisition) applications and are briefly discussed below.

(1) Dual Slope ADC

Dual slope type ADCs provide very precise and accurate voltage readings but have a slow conversion time due to the iterative methods they employ. Typically, these are found in multimeters and other voltage reading applications where precise conversions are required, but not in a timely manner.

(2) Pipelined ADC

As indicated by the name, pipelined type ADCs have a pipeline architecture of flash comparators that convert a voltage to a digital representation. The pipeline introduces a latency in conversion of around 3 clock cycles, but this type of ADC is well-suited for applications that need higher sample rates than the SAR and delta-sigma type ADCs but not the ultra-high sample rate achievable by the flash type ADC. Typical applications include digital oscilloscopes, spectrum analyzers, RADAR, software radio, among others.

(3) Flash ADC

Flash type ADCs operate without any latency so they can achieve the highest sample rate among all ADC types. A/D conversion occurs when an incoming voltage is compared to all possible voltages as a reference. To attain N-bit resolution, the flash ADC requires 2^N reference voltages, which means high resolutions require larger circuits with higher power consumptions. For that reason, resolution is typically no larger than 8 bits.

Flash ADCs are meant for applications that require extremely fast conversions but not large resolution, and as such, they are found in the fastest digital oscilloscopes, microwave measuring, fiber optics, and more.

1.3.4 Noise-Shaping Successive Approximation Register ADC

The RA6T2 contains two units of **Noise-Shaping SAR**-type A/D converters (**NS-SAR**). NS-SAR ADCs are an emerging hybrid A/D architecture that have features of the SAR type ADC and the delta-sigma modulation type ADC.

The NS-SAR's hybrid architecture combines the benefits of the SAR and DSM ADC architectures to achieve both high resolution and power efficiency simultaneously, characteristics which typically exist in a tradeoff relationship. NS-SAR type A/D converters benefit from SAR ADC qualities that make them low cost and energy efficient, and they benefit from DSM ADC qualities that provide a high signal-to-noise.

The hybrid architecture of the A/D converter module in the RA6T2 allows for new conversion methods for the peripheral that utilize oversampling and noise-shaping techniques to increase the resolution from 12-bits to 16-bits.

1.4 Increasing Resolution

This section gives an overview of the algorithmic process of oversampling and noise-shaping to increase resolution and explains when this method is most effective.

1.4.1 Oversampling and Noise-Shaping of NS-SAR ADCs

The Noise-Shaping Successive Approximation Register ADC units in the RA6T2 contain hardware circuits that implement the algorithmic processes of oversampling and noise-shaping to convert the analog input into a digital representation and obtain 16-bit data resolution. These methods are briefly explained as follows, but the detailed theory is outside the scope of this application note.

Oversampling achieves higher resolution through decimation. Decimation is a process of oversampling an input signal and averaging the samples to increase the number of bits from the A/D conversion that contain valid measurement information. First, the input signal is oversampled, meaning the A/D converter samples the input signal at a higher rate than the Nyquist sampling rate.

After enough samples are collected, their values are accumulated, and the sum is right shifted by n. This results in a final conversion value that has increased bits of useful data.

Noise-shaping is a method that uses the natural residue, or quantization noise, that occurs during A/D conversion. Noise-shaping mitigates the effects of this quantization noise through implementing a negative feedback loop filter.

1.4.2 Use Cases

There are many applications that require measuring a wide dynamic range of input values while also detecting slight changes to the input. Such systems need a measurement resolution larger than the typical 12-bit SAR ADCs and might benefit from using a 16-bit NS-SAR ADC instead.

The advantages and disadvantages of oversampling and noise-shaping provide helpful information when determining if oversampling to achieve 16-bit resolution is the right choice for your application:

Advantages:

- Implementing a 16-bit DAQ system is cheaper and easier with the RA6T2. The RA6T2 natively supports 16-bit resolution data so there is no need for an external ADC.
- The RA6T2 NS-SAR ADCs increase resolution without affecting CPU usage. Decimation can be implemented in software but that requires the CPU to perform necessary calculations. The dedicated hardware performs oversampling and averaging without the cost of reduced throughput and increased CPU bandwidth.

Disadvantages:

- Oversampling requires multiple samples to be captured before a single A/D value is converted. Your system should account for this initial delay period and subsequent group delay periods.
- NS-SAR ADCs produce more precise measurements at the cost of a lower maximum conversion frequency than their SAR counterpart.

Operating the RA6T2 ADC peripheral in its 16-bit operation methods is fitting for applications that require more precise measurements than 12-bit A/D data, but it might not be appropriate for extremely rapidly changing input signals. Carefully determine your data acquisition system's sampling rate requirements and use the RA6T2 Hardware Users' Manual chapter on Electrical Characteristics to determine if the conversion rate is acceptable.

1.4.3 Additional Renesas ADCs

If you determine that your system requires an ADC with higher resolution measurements than 16-bits, Renesas offers several options of 24-bit delta-sigma modulation ADCs that you can find on our website at: https://www.renesas.com/us/en/products/analog-products/data-converters/analog-digital-converters-adc-precision/delta-sigma-ad-converters

2. Configuring the RA6T2 ADC Module

The RA6T2 MCU contains two units of noise-shaping SAR-type A/D converters that have a hybrid architecture with features of both SAR type and delta-sigma modulation type.

The RA6T2 ADCs differ from other RA MCU ADC modules for the larger number of available features, including the extended 16-bit resolution that takes advantage of oversampling and noise-shaping successive approximation techniques to achieve. The wide variety of settings, features and conversion methods allow a user to tailor the configurations of the converter for their desired application.

2.1 ADC Specifications

This section reviews the major available settings and features of the ADC peripheral that are specified in the *RA6T2 Hardware Users' Manual* (HW UM) and/or are configurable with the Flexible Software Package (FSP) APIs. Each setting will be thoroughly explained to understand their impact on A/D operation and conversion.

2.1.1 Analog Channels

Analog channels are the physical source of the analog signal targeted for the A/D conversion. The analog channels consist of the following:

Analog input channels: A/D conversion channels for the analog input from MCU's I/O pin as the signal source.

Extended analog channels: A/D conversion channels for the source of the analog signal inside the MCU, for example, the temperature sensor, internal reference voltage and D/A converters.

Analog channels support both single-ended inputs and differential inputs. In total there are up to 29 analog input channels available, calculated based on the following conditions:

- A/D converter unit 0 (ADC0) can select up to 21 analog channels that support single-ended inputs and up to 12 analog channels that support differential input (capturing up to 6 differential signals).
- A/D converter unit 1 (ADC1) can select up to 17 analog channels that support single-ended inputs and up to 8 analog channels (4 pairs) that support differential inputs.
- ADC0 and ADC1 share 9 analog input channels.

The available configurations of the analog input channels are shown in the following table:

Table 1. Analog Channel List of Available Configurations

Analog channel number	Source of analog signal			A/D converter unit 0		A/D converter unit 1	
	Analog input	Programmable Gain Amplifier (PGA)	Channel-dedicated Sample-and-hold Circuit (S&H)*5	Single-ended input	Differential input*6	Single-ended input	Differential input*6
0	AN000 (AN000P)*3	PGAIN0	SHIN0P*3	✓	✓	—	—
1	AN001 (AN000N)*2	PGAVSS0	SHIN0N*4	✓		—	
2	AN002 (AN002P)*3	PGAIN1	SHIN1P*3	✓	✓	—	—
3	AN003 (AN002N)*2	PGAVSS1	SHIN1N*4	✓		—	
4	AN004 (AN004P)*3	PGAIN2	SHIN2P*3	✓	✓	—	—
5	AN005 (AN004N)*2	PGAVSS2	SHIN2N*4	✓		—	
6	AN006 (AN006P)	—	SHIN4P	—	—	✓	✓
7	AN007 (AN006N)	—	SHIN4N	—		✓	
8	AN008 (AN008P)	—	SHIN5P	—	—	✓	✓
9	AN009 (AN008N)	—	SHIN5N	—		✓	
10	AN010 (AN010P)	—	SHIN6P	—	—	✓	✓
11	AN011 (AN010N)	—	SHIN6N	—		✓	
12	AN012 (AN012P)	—	—	✓	✓	—	—
13	AN013 (AN012N)	—	—	✓		—	—
14	AN014 (AN014P)	—	—	✓	✓	—	—
15	AN015 (AN014N)	—	—	✓		—	—
16	AN016 (AN016P)	—	—	✓	✓	—	—
17	AN017 (AN016N)	—	—	✓		—	—
18	AN018 (AN018P)*3	PGAIN3*3	—	—	—	✓	✓
19	AN019 (AN018N)*2	PGAVSS3	—	—		✓*4	
20	AN020*1	—	—	✓	—	✓	—
21	AN021*1	—	—	✓		✓	
22	AN022*1	—	—	✓	—	✓	—
23	AN023*1	—	—	✓		✓	
24	AN024*1	—	—	✓	—	✓	—
25	AN025*1	—	—	✓		✓	
26	AN026*1	—	—	✓	—	✓	—
27	AN027*1	—	—	✓		✓	
28	AN028*1	—	—	✓	—	✓	—

Notes:

1. Do not perform A/D conversions of the same signal source from both A/D converter unit 0 and 1.
2. Only when the Programmable Gain Amplifier (PGA) is disabled and the PGA is set to single-ended input, the ANx can be used for A/D conversion as an analog input channel. When the PGA is enabled, Anx functions as the PGAVSS pin. Do not perform A/D conversion of Anx. Regardless of the PGA enable/disable setting, when the PGA is set to pseudo-differential input, A/D conversion of Anx is not possible.
3. When the PGA is used in single-ended or pseudo-differential input mode, the A/D is converting the output of the PGA, the A/D conversion channel corresponding to PGAINn (n = 0 to 3) should be set to single-ended input.
4. When using the PGA or when the PGA is set to pseudo-differential mode, the SHINxN (inverting input (-)) of the channel-dedicated sample-and-hold circuit cannot be used.

5. When the channel-dedicated sample-and-hold circuit is used in single-ended input the SHINxP (non-inverting input (+)) channel and the SHINxN (inverting input (-)) channel must not be assigned to the same scan group (x = 0 to 2, 4 to 6).
6. When performing A/D conversion with differential input, set the even-numbered channel as the A/D conversion target. For differential input the even-numbered channels function as non-inverting inputs and the odd-numbered channels function as inverting inputs (-).

The available configurations of the extended analog input channels are shown in the following table:

Table 2. Extended Analog Channel List of Available Configurations

Analog channel number	Source of analog signal*1	A/D converter unit 0		A/D converter unit 1	
		Single-ended input*2	Differential input*3	Single-ended input*2	Differential input*3
96	Self-diagnosis	—	✓	—	✓
97	Temperature sensor	✓	—	✓	—
98	Internal reference voltage	✓	—	✓	—
101	D/A converter channel 0	✓	—	✓	—
102	D/A converter channel 1	✓	—	✓	—
103	D/A converter channel 2	✓	—	✓	—
104	D/A converter channel 3	✓	—	✓	—

Notes:

1. Do not perform A/D conversion of the same signal source from both A/D converter unit 0 and unit 1.
2. The extend analog function (except for the self-diagnosis function) is only supported with the single-ended input. Do not set to differential input.
3. The self-diagnosis function is only supported with differential input.

2.1.2 Virtual Channels

A virtual channel is a group of registers that stores the A/D conversion configurations for an analog channel, including, but not limited to the selection of the analog channels, the optional settings for A/D conversion, the data processing method of the A/D conversion data, the assignment to scan groups, and so forth.

Before performing the A/D conversion, the analog channel should be assigned to any of the virtual channels and the virtual channel should be assigned to a scan group. The relation between analog channels, virtual channels and scan groups is discussed in greater detail in section 2.1.3.

Notes:

- The virtual channel can only be assigned to one scan group.
- Assign the analog channel to several virtual channels if:
  - the same analog channel is to be converted in different scan groups, or
  - the same analog channel is to be converted several times in the same scan group
- FIFO function should be used in combination when performing A/D conversion more than once using more than one virtual channel within the same scan group for the same analog channel.

2.1.3 Scan Groups

A scan group is a group of analog channels that perform A/D conversion in a scanning operation. To use a scan group, do the following:

- Assign the analog channel for the A/D conversion to the desired virtual channel
- Assign the virtual channel to the scan group
- Assign the scan group to the specific A/D converter

The following image depicts an example of the relationship between analog channels, virtual channels, and scan groups. It shows the analog channel assignments and the resulting order of conversion during the scanning operation for each A/D converter.

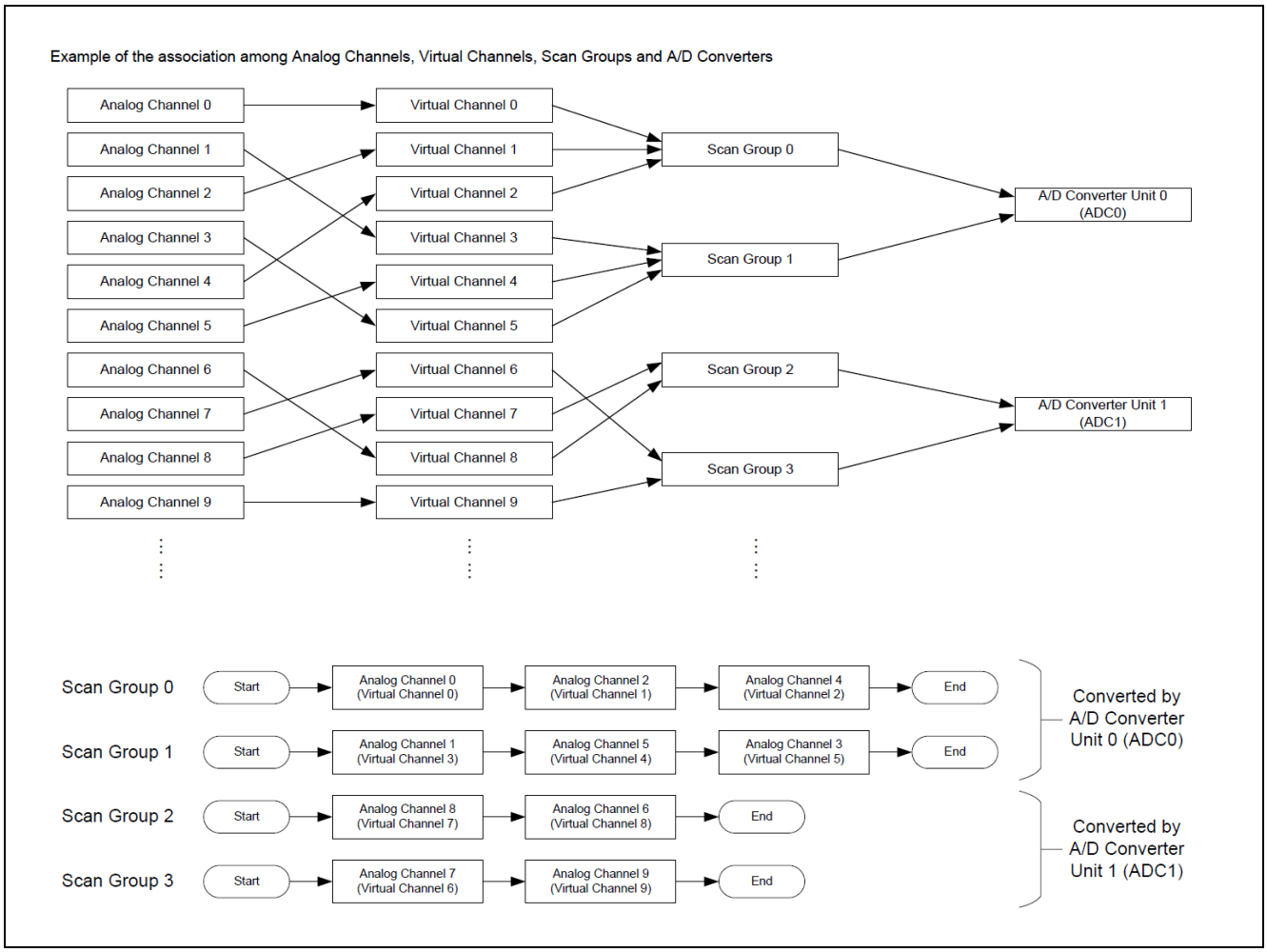


Figure 3. Example Configuration Depicting the Scan Group assignments

A single scan group can be assigned to multiple virtual channels. The conversion method determines the ranges of virtual channels as follows:

- SAR mode: up to 8 virtual channels per scan group
- Oversampling mode: up to 8 virtual channels per scan group
- Hybrid mode: 2 to 4 virtual channels per scan group

If more than the maximum number of virtual channels is assigned to one scan group, channels are converted in ascending order and any subsequent above the max will not be converted.

Note:

- The virtual channels assigned to a particular scan group should be assigned only the analog channels that can be converted by that ADC unit.
  - Any analog channels included that cannot be converted by the specified ADC unit will result in undefined conversions.
- Hybrid mode requires at least 2 virtual channels to be assigned to a scan group for operation to be guaranteed.
- If more than the maximum number of virtual channels is assigned to one scan group, channels are converted in ascending order and any subsequent above the max will not be converted.

2.1.4 Analog Input Mode

The ADC peripheral supports both single-ended inputs and differential inputs with the following counts:

- A/D converter unit 0 (ADC0) can select up to 21 analog channels that support single-ended inputs and up to 12 analog channels (6 pairs) that support differential inputs.
- A/D converter unit 1 (ADC1) can select up to 17 analog channels that support single-ended inputs and up to 8 analog channels (4 pairs) that support differential inputs.

The relationship between analog channels and the possible input modes is shown in greater detail in the tables of section 2.1.1 Analog Channels.

2.1.4.1 Single-Ended Input

In single-ended input mode, the difference between the voltage of the analog channel's signal source and the analog reference ground voltage (VREFL0) is converted.

For single-ended input, A/D conversion is formatted into unsigned data format based in the setting of the ADDOPCRCn.SIGNSEL (n = 0 to 36), where SIGNSEL should be 1.

2.1.4.2 Differential Input

In differential input mode, the even-numbered analog channel is used as the non-inverting input (+) (A_{INP}) and the odd-numbered channel is used as the inverting input (-) (A_{INN}). The difference between the voltage of the non-inverting input (+) and inverting input (-) (A_{INP} - A_{INN}) is converted.

For differential input A/D conversion is formatted into signed data format based in the setting of the ADDOPCRCn.SIGNSEL (n = 0 to 36), where SIGNSEL should be 0.

For A/D conversion with differential input mode, select the non-inverting input (+) channel of the target of differential pair in ADCHCRn.CNVCS[6:0] (n = 0 to 36) bits, and select the differential input mode in ADCHCRn.AINMD (n = 0 to 36) bits.

Note: The differential input pair is a combination of channels with analog channel number 2i and 2i + 1 (i = 0, 1, 2, 3, ...), and they are named ANxxxP and ANxxxN (xxx = 000, 002, 004, ...). The combination of discontinuous analog channel numbers or the combination of analog channel numbers 2i − 1 and 2i cannot be used as a differential input pair.

2.1.5 Conversion Methods

The A/D converter units of the RA6T2 have a hybrid architecture with combined features of a SAR type and delta-sigma type ADC. Because of the unique architecture, each ADC unit can convert in one of three different conversion methods: SAR mode, oversampling mode, and hybrid mode.

The key features, important restrictions, and major differences between the three operating modes are detailed in this section to help guide selection for your own application based on conversion requirements.

A/D conversion characteristics are determined by the combination of the conversion method and the scan mode selected. For more information on how these modes interact and affect processing for achieving 16-bit resolution, see section 2.2, Operating ADC with 16-bit Depth.

2.1.5.1 SAR Mode

In SAR mode, the A/D converter operates as a Nyquist rate A/D converter, sampling analog channels one at a time using the SAR method.

Key Features

- A/D converter samples the signal source once, and converts by SAR method
- Fast A/D conversion
- Supports up to 8 channels per one scan group
- Supports only single-ended input (excluding self-diagnosis function)
- Supports 10-bit and 12-bit resolution

Restrictions

(1) Resolution and Accuracy:

SAR mode is a 12-bit resolution A/D converter and as such, the accuracy of the data output is only guaranteed for 12 bits.

However, if 14-bit or 16-bit mode is selected, internally the data output from the A/D converter is extended to 16-bits for calculation processing. Select 14-bit or 16-bit length data format in SAR mode to increase the computational resolution, such as when using A/D-converted value addition/averaging function or User's Gain/User's Offset function.

(2) Differential Input

SAR mode only supports single-ended input except during the self-diagnosis operation. A/D conversion with differential input mode is only guaranteed during the self-diagnosis operation.

(3) Digital Filter Function

Operation is not guaranteed in SAR mode when the digital filter function is enabled.

2.1.5.2 Oversampling Mode

In Oversampling mode, the A/D converter operates by oversampling analog channels and converting the analog signal to digital data by the NS-SAR method.

Key Features

- A/D converter oversamples the signal source, and converts analog-to-digital by NS-SAR method
- High-accuracy A/D conversion
- Supports up to 8 channels per one scan group
- Supports single-ended input and differential input
- Supports 10,12,14 and 16-bit resolution

Restrictions

(1) Digital Filter Function

The digital filter function is mandatory in Oversampling mode. A/D conversion is not guaranteed without the use of the digital filter function.

2.1.5.3 Hybrid Mode

Hybrid mode has both features of SAR mode and Oversampling mode. In Hybrid mode, the A/D converter operates by oversampling and converting the analog signal to digital data by the NS-SAR method. In Hybrid mode, when switching channels for each sampling, multiple analog channels are oversampled simultaneously.

Key Features

- A/D converter oversamples the signal source, and converts analog-to-digital by NS-SAR method
- High-accuracy A/D conversion
- High data rate in continuous scanning operation
- Supports up to 4 channels per one scan group
- Supports single-ended input and differential input
- Supports 10,12,14 and 16-bit resolution

Restrictions

(1) Scan Group Assignment

In Hybrid mode, 2-4 virtual channels must be assigned to a scan group for operation to be guaranteed. Scanning operation with only one virtual channel is prohibited for ADC conversion results to be guaranteed.

(2) Virtual Channel Assignment

In Hybrid mode, do not assign the same analog channel to multiple virtual channels assigned to the same scan group for operation to be guaranteed.

(3) Digital Filter Function

The digital filter function is mandatory in Hybrid mode. A/D conversion is not guaranteed without the use of the digital filter function.

The digital filters selected by the ADDOPCRAn.DFSEL[2:0] (n = 0 to 36) bits must be set to be exclusive among virtual channels in the same scan group, since multiple digital filters are used simultaneously. If the same digital filter is selected from multiple virtual channels in the same scan group, operation is not guaranteed.

(4) A/D Converted Value Addition/Averaging Function

When using A/D converted value addition/averaging function in Hybrid mode, set the number of addition/average times for all virtual channels in the same scan group to the same value for operation to be guaranteed.

(5) Trigger Interval for Background Continuous Scan Mode

In Hybrid mode – Background Continuous Scan mode, AD conversion start trigger should be input no sooner than after the following intervals:

- When trigger delay function is not used, or software trigger: 8 ADCLK cycles or more
- When trigger delay function is used: (8 + (setting value of ADTRGDLRm.TRGDLYn[7:0])) × ADCLK cycles or more, for m = 0 to 4, n = 0 to 8

If this restriction is violated, the A/D conversion start trigger will not be accepted and will be ignored.

2.1.6 Scan Modes

The scanning operation is the operation of sequential A/D conversion for the analog channels. There are 3 scan modes of the scanning operation selectable: Single Scan mode, Continuous Scan mode, and Background Continuous Scan mode.

The available combinations of scan modes and conversion methods of the ADC units are show in the following table:

Table 3. Available Combinations of Scan Modes and Conversion Methods

Scan mode	Operation mode of A/D converter		
	SAR mode	Oversampling mode	Hybrid mode
Single scan mode	✓	✓	✓
Continuous scan mode	✓	✓	✓
Background continuous scan mode	—	—	✓

Note: ✓: available, —: not available

The scanning operations are performed based on the scan groups. When the scanning operation is started, A/D conversion is performed for each analog channel, based on the virtual channel settings. If the scanning operation starts for 2 or more scan groups assigned to the same A/D converter at the same time, the scanning operation of the group with the lowest scan group number is performed.

The A/D conversion order of analog channels is in ascending order based on the virtual channel numbers assigned to the scan group. The channels with the lowest virtual channel numbers are converted at the beginning of the scan group conversion.

2.1.6.1 Single Scan Mode

In single scan mode, one scan group is converted once for each A/D conversion start trigger input. Each time an A/D conversion start trigger occurs, A/D conversion is performed once for each analog channel assigned in the scan group.

2.1.6.2 Continuous Scan Mode

Continuous scan mode repeats the scanning operation of one scan group. When an A/D conversion start trigger occurs, A/D conversion of each analog channel assigned to that scan group is repeated until the A/D conversion stop processing occurs.

2.1.6.3 Background Continuous Scan Mode

Background continuous scan mode repeats the scanning operation of a single scan group. Scanning operation begins at the input of the first A/D conversion start trigger. Like in continuous scan mode, A/D conversion of each analog channel assigned to that scan group is repeated until the A/D conversion stop processing occurs.

When an A/D conversion start trigger occurs during the continuous scanning operation, the A/D conversion data at that point is output. Without the A/D conversion start trigger, the A/D conversion data are not output. and the values of the A/D data registers and the First In First Out (FIFO) data registers are not updated.

2.1.7 A/D Scan Start Conditions

A/D conversion can be triggered from multiple peripheral modules on the MCU. These include the Event Link Controller (ELC), the General Purpose PWM Timer (GPT), and external triggers from the I/O ports. The variety of sources provides flexibility for your signal acquisition system design. Each of these listed triggers is described in further detail in this section.

To perform A/D conversion by a trigger from a peripheral module, the trigger needs to be configured for each scan group and the trigger input needs to be enabled from the peripheral module in the ADTRGENR register.

In addition, there is a trigger delay function that delays the A/D conversion start trigger to adjust the A/D conversion start time for each scan group. The delay function is available for triggers generated by the ELC, GPT, or external source but not for software trigger sources. The delay value to be added is set for each scan group in ADTRGDLRi (i = 0 to 4) register. The delay time added is the register setting value × A/D conversion clock (ADCLK) cycle.

2.1.7.1 ELC Trigger Scan Start

A/D conversion can be initiated by an event from the Event Link Controller. To enable scan start by the ELC, configure the scan group using the ELC event in the ADTRGELCn (n = 0 to 8) register and enable the A/D conversion start trigger in the ADTRGENR register.

The following table shows the relationships between the ADTRGELCn register and its corresponding ELC event:

Table 4. ELC Event Names and Corresponding Register Bit Setting

Resister bit	Event Name
ADTRGELCn.TRGELC0	ELC_AD00
ADTRGELCn.TRGELC1	ELC_AD01
ADTRGELCn.TRGELC2	ELC_AD02
ADTRGELCn.TRGELC3	ELC_AD10
ADTRGELCn.TRGELC4	ELC_AD11
ADTRGELCn.TRGELC5	ELC_AD12

Note: n = 0 to 8

2.1.7.2 GPT Trigger Scan Start

A/D conversion can also be initiated by an interrupt from the general purpose PWM timer. To enable scan start by the GPT, configure the scan group using the GPT interrupt source in the ADTRGGPTn (n = 0 to 8) register and enable the A/D conversion start trigger in the ADTRGENR register.

The following table shows the relationships between the ADTRGGPTn register and GPT interrupt sources:

Table 5. GPT Interrupt Sources and Corresponding Register Bit Setting

Resister bit	Event Name
ADTRGGPTn.TRGGPTA0	GPT0_ADTRGA
ADTRGGPTn.TRGGPTA1	GPT1_ADTRGA
ADTRGGPTn.TRGGPTA2	GPT2_ADTRGA
ADTRGGPTn.TRGGPTA3	GPT3_ADTRGA
ADTRGGPTn.TRGGPTA4	GPT4_ADTRGA
ADTRGGPTn.TRGGPTA5	GPT5_ADTRGA
ADTRGGPTn.TRGGPTA6	GPT6_ADTRGA
ADTRGGPTn.TRGGPTA7	GPT7_ADTRGA
ADTRGGPTn.TRGGPTA8	GPT8_ADTRGA
ADTRGGPTn.TRGGPTA9	GPT9_ADTRGA
ADTRGGPTn.TRGGPTB0	GPT0_ADTRGB
ADTRGGPTn.TRGGPTB1	GPT1_ADTRGB
ADTRGGPTn.TRGGPTB2	GPT2_ADTRGB
ADTRGGPTn.TRGGPTB3	GPT3_ADTRGB
ADTRGGPTn.TRGGPTB4	GPT4_ADTRGB
ADTRGGPTn.TRGGPTB5	GPT5_ADTRGB
ADTRGGPTn.TRGGPTB6	GPT6_ADTRGB
ADTRGGPTn.TRGGPTB7	GPT7_ADTRGB
ADTRGGPTn.TRGGPTB8	GPT8_ADTRGB
ADTRGGPTn.TRGGPTB9	GPT9_ADTRGB

Note: n = 0 to 8

2.1.7.3 External Trigger Scan Start

A/D conversion can be initiated by input from an external trigger pin, ADTRG0 and ADTRG1. To enable scan start using an external trigger, configure the scan group using the external trigger source in the ADTRGENR register.

Note: External trigger is active low. Before enabling the external trigger (ADTRG0 and ADTRG1), set the external trigger pin to high.

2.1.8 Interrupt Sources

There are a variety of interrupt sources and ELC event requests that can be created by the A/D converter peripheral. This section lists the interrupt sources offered and describes the conditions that generate the requests in further detail.

The following table lists the interrupt sources and ELC events that can be created by the ADC module:

Table 6. List of Interrupt Sources and ELC Events the ADC Module Can Create

Interrupt request or ELC event	Symbol	Description	Status flag
A/D converter error interrupt	ADC_ERR0	Generated when abnormal operation of ADC0 is detected.	ADERSR.ADERF0
	ADC_ERR1	Generated when abnormal operation of ADC1 is detected.	ADERSR.ADERF1
A/D converter self-calibration end interrupt	ADC_CALEND0	Generated when self-calibration operation of ADC0 is finished.	ADCALENDSR.CALENDF0
	ADC_CALEND1	Generated when self-calibration operation of ADC1 is finished.	ADCALENDSR.CALENDF1
A/D scan end interrupt	ADC_ADI0	Generated at the scan end of scan group 0	ADSCANENDSR.SCENDF0
	ADC_ADI1	Generated at the scan end of scan group 1	ADSCANENDSR.SCENDF1
	ADC_ADI2	Generated at the scan end of scan group 2	ADSCANENDSR.SCENDF2
	ADC_ADI3	Generated at the scan end of scan group 3	ADSCANENDSR.SCENDF3
	ADC_ADI4	Generated at the scan end of scan group 4	ADSCANENDSR.SCENDF4
	ADC_ADI5678	Generated at the scan end of scan group 5 to 8	ADSCANENDSR.SCENDF5 ADSCANENDSR.SCENDF6 ADSCANENDSR.SCENDF7 ADSCANENDSR.SCENDF8
Limitter clip interrupt	ADC_LIMCLPI	Generated when a limiter clip using limiter table 0 to 7 is detected for A/D conversion.	ADLIMGRSR.LIMGRFn ADLIMCHSR0.LIMCHFi ADLIMEXSR.LIMEXFj
A/D conversion overflow interrupt	ADC_RESOVF0	Generated when the overflow is detected in A/D conversion result with ADC0.	ADOVFERSR.ADOVFEF0 ADOVFCHSR0.OVFCHFi ADOVFEXSR.OVFEXFj
	ADC_RESOVF1	Generated when the overflow is detected in A/D conversion result with ADC1.	ADOVFERSR.ADOVFEF1 ADOVFCHSR0.OVFCHFi ADOVFEXSR.OVFEXFj
Compare match interrupt	ADC_CMPI0	Generated when a compare match with compare match table 0 is detected.	ADCMPTBSR.CMPTBF0
	ADC_CMPI1	Generated when a compare match with compare match table 1 is detected.	ADCMPTBSR.CMPTBF1
	ADC_CMPI2	Generated when a compare match with compare match table 2 is detected.	ADCMPTBSR.CMPTBF2
	ADC_CMPI3	Generated when a compare match with compare match table 3 is detected.	ADCMPTBSR.CMPTBF3
Composite compare match interrupt	ADC_CCMPM0 ADC_CCMPM1	Generated when a composite compare match with combined conditions using compare match table 0 to 7 is detected.	ADCMPTBSR.CMPTBF0 to ADCMPTBSR.CMPTBF8

Interrupt request or ELC event	Symbol	Description	Status flag
FIFO data read request interrupt	ADC_FIFOREQ0	Generated when the number of vacant stages in FIFO for scan group 0 become less than or equal to the specified value.	ADFIFOSR0.FIFOST0[3:0]
	ADC_FIFOREQ1	Generated when the number of vacant stages in FIFO for scan group 1 become less than or equal to the specified value.	ADFIFOSR0.FIFOST1[3:0]
	ADC_FIFOREQ2	Generated when the number of vacant stages in FIFO for scan group 2 become less than or equal to the specified value.	ADFIFOSR1.FIFOST2[3:0]
	ADC_FIFOREQ3	Generated when the number of vacant stages in FIFO for scan group 3 become less than or equal to the specified value.	ADFIFOSR1.FIFOST3[3:0]
	ADC_FIFOREQ4	Generated when the number of vacant stages in FIFO for scan group 4 become less than or equal to the specified value.	ADFIFOSR2.FIFOST4[3:0]
	ADC_FIFOREQ5678	Generated when the number of vacant stages in any of FIFO for scan groups 5 to 8 become less than or equal to the specified value.	ADFIFOSR2.FIFOST5[3:0] ADFIFOSR3.FIFOST6[3:0] ADFIFOSR3.FIFOST7[3:0] ADFIFOSR4.FIFOST8[3:0]
FIFO data overflow interrupt	ADC_FIFOOVF	Generated when FIFO overflow is detected in any of FIFO for scan group 0 to 8.	ADFIFOERSR.FIFOOVF0 to ADFIFOERSR.FIFOOVF8

Note: n = 0 to 8.
i = 0 to 28.
j = 0 to 2, 5 to 8.

(1) A/D Converter Error Interrupt

An A/D converter error interrupt can be generated when an error with the A/D converter is detected.

A/D converter error interrupt (ADC_ERRm (m = 0, 1)) is generated when the ADERINTCR.ADEIEm (m = 0, 1) bit is 1 and ADERSR.ADERFm (m = 0, 1) bit is 1.

(2) A/D Converter Self-Calibration End Interrupt

An A/D converter self-calibration end interrupt can be generated at the end of A/D converter self-calibration operation.

A/D converter self-calibration end interrupt (ADC_CALENDm (m = 0, 1)) is generated when the ADCALINTCR.CALENDIEm (m = 0, 1) bit is 1 and ADCALENDSR.CALENDFm (m = 0, 1) bit is 1.

(3) A/D Scan End Interrupt

A/D scan end interrupt can be generated at the end of scanning operation of scan group n (n = 0 to 8).

A/D scan end interrupt for scan group 0 to 4 (ADC_ADI0 to ADC_ADI4) are generated when ADINTCR.ADIEn (n = 0 to 4) bit is set to 1 and ADSCANENDSR.SCENDFn (n = 0 to 4) bit is set to 1.

The A/D scan end interrupt, for scan groups 5 through 8 (ADC_ADIn), is generated when the ADINTCR.ADIEn (n = 5 to 8) bit = 1 and the ADSCANENDSR.SCENDFn (n = 5 to 8) bit i= 1 for any of the scan groups 5 through 8.

However, A/D scan end interrupt is not generated when A/D conversion operation (scanning operation) is forcibly stopped by ADSTOPR register.

(4) Limiter Clip Interrupt

Limiter clip interrupt (ADC_LIMCLPI) can be generated when a limiter clip with limiter table i (i = 0 to 7) is detected.

Limiter clip interrupt is generated when either the ADLIMINTCR.LIMIEi (i = 0 to 8) bit is 1 or ADLIMGRSR.LIMGRFi (i = 0 to 8) bit is 1.

(5) A/D Conversion Overflow Interrupt

A/D conversion overflow interrupt can be generated when an A/D conversion result overflow is detected in either ADC unit.

A/D conversion overflow interrupt (ADC_RESOVFm (m = 0, 1)) is generated when the ADOVFINTCR.ADOVFIEm (m = 0, 1) bit is 1 and ADOVFERSR.ADOVFEFm (m = 0, 1) bit is 1.

(6) Compare Match Interrupt

Compare match interrupt is generated when a compare match is detected when using the composite compare match function.

Compare match interrupt (ADC_CMPIj (j = 0 to 3)) is generated when the ADCMPINTCR.CMPIEj (j = 0 to 3) bit is 1 and ADCMPTBSR.CMPTBFj (j = 0 to 3) bit is 1.

(7) Composite Compare Match Interrupt

Composite compare match interrupt (ADC_CCMPMk (k = 0, 1)) can be generated by combining the comparison results of two or more compare match tables.

(8) FIFO Data Read Request Interrupt

FIFO data read request interrupt can be generated when the number of vacant stages in FIFO becomes less than or equal to the specified value.

FIFO data read request interrupts for scan group 0 to 4 (ADC_FIFOREQ0 to ADC_FIFOREQ4) are generated when the ADFIFOINTCR.FIFOIEn (n = 0 to 4) is set to 1 and ADFIFOSRm.FIFOSTn[3:0] ≤ ADFIFOINTLRm.FIFOILVn[3:0] (m = 0 to 2, n = 0 to 4).

FIFO data read request interrupts for scan group 5 to 8 (ADC_FIFOREQ5 to ADC_FIFOREQ8) are generated when the ADFIFOINTCR.FIFOIEn (n = 5 to 8) is set to 1 and ADFIFOSRm.FIFOSTn[3:0] ≤ ADFIFOINTLRm.FIFOILVn[3:0] (m = 3 to 4, n = 5 to 8)

Note: FIFO read request interrupt is not generated while the ADFIFOERSR.FIFOFLFn (n = 0 to 8) bit corresponding to its interrupt source is set to 1.

(9) FIFO Data Overflow Interrupt

FIFO data overflow interrupt (ADC _FIFOOVF) can be generated when a FIFO overflow is detected in any of FIFO in scan groups 0 to 8.

FIFO data overflow interrupt is generated when ADFIFOINTCR.FIFOIEn (n = 0 to 8) bit is 1 and ADFIFOERSR.FIFOOVFn (n = 0 to 8) bit is 1.

2.1.9 Self-Calibration Function

The ADC has a built-in self-calibration function, which serves to normalize the variations of A/D converter characteristics that are caused by chip-to-chip variations.

2.1.9.1 Conditions When Self-Calibration is Required

Self-Calibration should be performed any time the ADC operating characteristics are modified, including after reset, releasing module-stop, when returning from Software Standby or Deep Software Standby mode, and each time the R_ADC_B_ScanCfg() function is called. A detailed list of conditions under which self-calibration is required is shown in the following table.

Table 7. Conditions When Self-Calibration is Required

Conditions under which self-calibration is required	Internal-circuit Calibration	Gain/Offset Calibration	Channel-dedicated sample-and-hold circuit Gain and Offset Calibration *1
After reset release	✓	✓	✓
After releasing the module stop	✓	✓	✓
When returning from Software Standby mode or Deep Software Standby mode	✓	✓	✓
When changed the ADCLK setting (When clock source or frequency is changed)	✓	✓	✓
When changed the operation mode or the scan mode of the A/D converter (When changed ADMDR.ADMDm bit (m = 0, 1))	✓	✓	✓
When changed A/D successive approximation time (When changed ADCNVSTR.CSTm bit (m = 0, 1))	✓	✓	✓
When changed the operation setting of the channel-dedicated sample-and-hold circuit *1	—	—	✓

Note: ✓: Self-calibration should be performed.
—: Self-calibration is not required.
Note 1. When any of ADSHCRm.SHENn bits are set to 1, or any of ADSHCRm.SHENn bits are changed, perform the channel-dedicated sample-and-hold circuits self-calibration (m = 0, 1, n = 0 to 2, 4 to 6). Not required if the channel-dedicated sample-and-hold circuits are not used.

If a condition shown in this table occurs, self-calibration needs to be performed before using the A/D converter for the conversion results to be guaranteed. If the A/D converter is still operating after a first attempt, stop all A/D converters and perform the self-calibration again.

2.1.9.2 Self-Calibration Operation and Procedure

The self-calibration function has the following purposes:

(1) Internal Circuit Calibration:

Self-calibration adjusts the operation of the A/D internal circuitry.

(2) Gain and Offset Calibration:

Self-calibration measures the A/D converter's gain error and offset error. The calibration processing of the A/D conversion result based on the measured error data is performed after the A/D conversion. This self-calibration should be performed after internal circuit self-calibration is completed.

(3) Channel-Dedicated Sample-and-Hold Circuit Gain and Offset Calibration:

Self-calibration measures the A/D converter's gain error and offset error when using the channel-dedicated sample-and-hold circuit. The calibration processing of the A/D conversion result based on the measured error data is performed after the A/D conversion. This self-calibration should be performed after A/D converter Gain/Offset self-calibration is completed.

2.1.9.3 Self-Calibration Restrictions

(1) Prohibition of Scanning Operation

When performing the scanning operation, start the scanning operation after self-calibration is completed. Operation is not guaranteed when the scanning operation is started during self-calibration.

(2) Prohibition of Additional Writes to ADCALSTR Register

After the self-calibration is started, writing to ADCALSTR register is prohibited until self-calibration is completed. Operation is not guaranteed if this restriction is violated.

(3) Prohibition of Forced Stop

Do not forcibly stop with the ADSTOPR register during self-calibration. Even if the A/D conversion operation needs to be forcibly stopped due to a system error or exception handing, be sure to wait to stop until self-calibration completes for guaranteed operation.

(4) Restrictions for Channel-Dedicated Sample-and-Hold Circuit

To perform the self-calibration for the channel-dedicated sample-and-hold circuit, all channel-dedicated sample-and-hold circuits connected to the A/D converter should be enabled as follows:

- For ADC0: set 1 for ADSHCR0.SHEN0 to SHEN2 bit
- For ADC1: set 1 for ADSHCR1.SHEN4 to SHEN6 bit
- If there is a channel-dedicated sample-and-hold circuit that is not used, set it to disabled (ADSHCRm.SHENn = 0 (m = 0, 1, n = 0 to 2, 4 to 6) after self-calibration completes.

Operation is not guaranteed when the self-calibration for the channel-dedicated sample-and-hold circuit is performed while any channel-dedicated sample-and-hold circuits are disabled.

(5) Restrictions on Self-Calibration Operation and Noise

Self-calibration should be performed for the A/D converters one by one. When one A/D converter is in the self-calibration operation, the other converters should be idle, that is, not in scan operation and not in self-calibration operation.

If this restriction is violated, the A/D conversion accuracy will be degraded due to noise during the self-calibration operation. In this case, the A/D converter characteristics are not guaranteed. For the best results, self-calibration operation should be performed under conditions with as little noise as possible.

(6) Restrictions on State Setting

The number of states to be set in the ADCALSTCR and ADCALSHCR registers should satisfy the values specified in the Electrical Characteristics section of the RA6T2 Hardware Users' Manual. Also, the number of states involved in self-calibration should be set to satisfy the following restrictions:

(a) Restrictions on setting ADCALSTCR register

- ADCALSTCR.CALADSST[9:0] bits
  - Depending on the type of self-calibration, set the values to satisfy those in the Electrical Characteristics chapter of the RA6T2's HW UM.
  - If the setting value differs according to the type of self-calibration, change the register setting value for each self-calibration.
- ADCALSTCR.CALADCST[5:0] bits
  - Set the same value as ADCNVSTR.CSTm[5:0] (m=0,1) bits.
    - (ADCALSTCR.CALADCST[5:0] = ADCNVSTR.CSTm[5:0])

(b) Restrictions on setting ADCALSHCR register

These only apply when a channel-dedicated sample-and-hold circuit is used.

- ADCALSHCR.CALSHSST[7:0] bits
  - Set the value equal to the value of the ADSHSTRm.SHSST[7:0] bits plus 1 (m = 0, 1).
    - (ADCALSHCR.CALSHSST[7:0] = ADSHSTRm.SHSST[7:0] + 1)
  - If the setting values of ADSHSTR0.SHSST[7:0] and ADSHSTR1.SHSST[7:0] are different, change the register setting values of CALSHSST[7:0] for each self-calibration of ADC0 (SH0 to SH2) and ADC1 (SH4 to SH6).
- ADCALSHCR.CALSHHST[2:0] bits
  - Set the same value as ADSHSTRm.SHHST[2:0] (m = 0, 1) bits.

2.2 Operating ADC with 16-bit Depth

The ADC peripheral on the RA6T2 MCU is unique in its ability to extend 12-bit resolution to 16-bit resolution by taking advantage of oversampling and averaging techniques. There are 5 different combinations of conversion method - scan mode pairs that can achieve 16-bit resolution, which are highlighted in the following table:

Table 8. Highlighted Conversion Method – Scan Mode Pairs Can Provide 16-bit Resolution

Scan mode	Operation mode of A/D converter		
	SAR mode	Oversampling mode	Hybrid mode
Single scan mode	✓	✓	✓
Continuous scan mode	✓	✓	✓
Background continuous scan mode	—	—	✓

Note: ✓: available, —: not available

Each pair’s scanning operation procedures are described in greater detail in the following sections.

2.2.1 Oversampling Mode – Single Scan Mode

The following steps detail the procedure of Oversampling Mode – Single-Scan Mode:

1. In oversampling mode - single scan mode, scanning operation begins when a software trigger or a peripheral module trigger is input. The scan group corresponding to the trigger is started.
2. Each analog channel is oversampled according to the number of taps in the digital filter and the number of A/D-converted value addition/averaging times. The oversampled data stored in the digital filter is discarded after the oversampling of each analog channel is completed.
3. A/D conversion data of each analog channel is output after oversampling. The A/D conversion data is stored in the data register (ADDRi (i = 0 to 28), ADEXDRj (j = 0 to 2, 5 to 8)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 8)).
4. If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
5. During the scanning operation, ADGRSR.ACTGRn (n = 0 to 8) bit corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0, 1) bit corresponding to the A/D converter performing the A/D conversion is also set to 1. When scanning operation is complete, each bit is cleared to 0 and A/D converter enters the idle state.

Following is an example timing diagram of the scanning operation in oversampling mode – single scan mode:

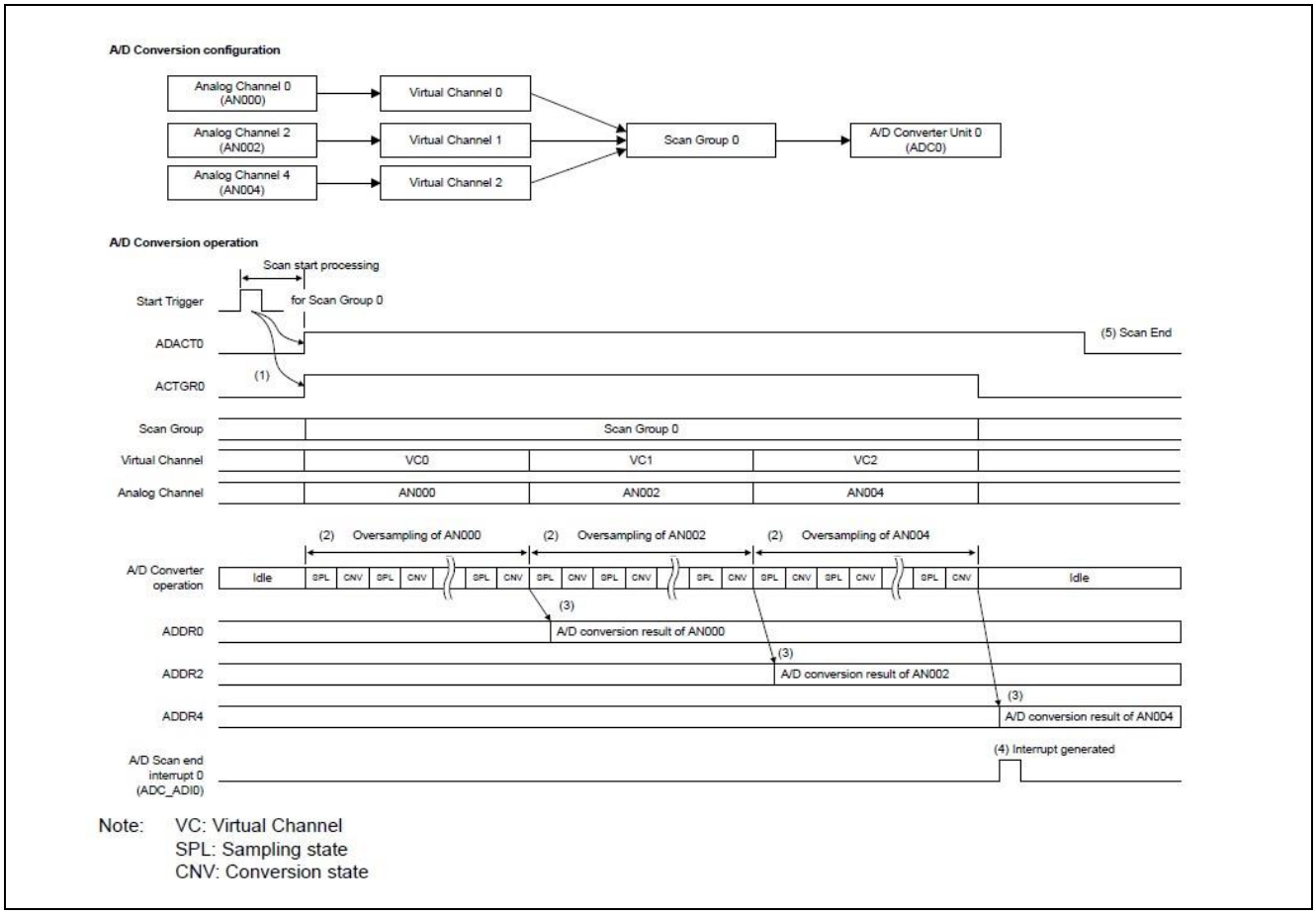


Figure 4. Timing diagram of Oversampling Mode – Single-Scan Mode

In this example ADC analog channels 0, 2, and 4 are assigned to virtual channels 0, 1, and 2, respectively. All the virtual channels are assigned to scan group 0 and are converted by ADC0. The order of scanning of the virtual and analog channels are shown. Each analog channel is oversampled to convert one ADC data sample.

2.2.2 Oversampling Mode – Continuous Scan Mode

The following steps detail the procedure of Oversampling Mode – Continuous-Scan Mode:

1. In oversampling mode – continuous scan mode, scanning operation begins when a software trigger or a peripheral module trigger is input. The scan group corresponding to the trigger is started.
2. When scanning operation starts, ADGRSR.ACTGRn (n = 0 to 8) corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0, 1) bit corresponding to the A/D converter performing the A/D conversion is also set to 1.
3. Each analog channel is oversampled according to the number of TAPs in the digital filter and the number of A/D-converted values' addition/averaging times. The oversampled data stored in the digital filter taps are discarded after the oversampling of each analog channel is completed.
4. A/D conversion data of each analog channel is output after oversampling. The A/D conversion data is stored in the data register (ADDRi (i = 0 to 28), ADEXDRj (j = 0 to 2, 5 to 8)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 8)).
5. If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
6. Until the A/D conversion stop process is performed, steps 3-5 are repeated, and the scanning operation continues.

Following is a timing diagram of the scanning operation in oversampling mode – continuous scan mode:

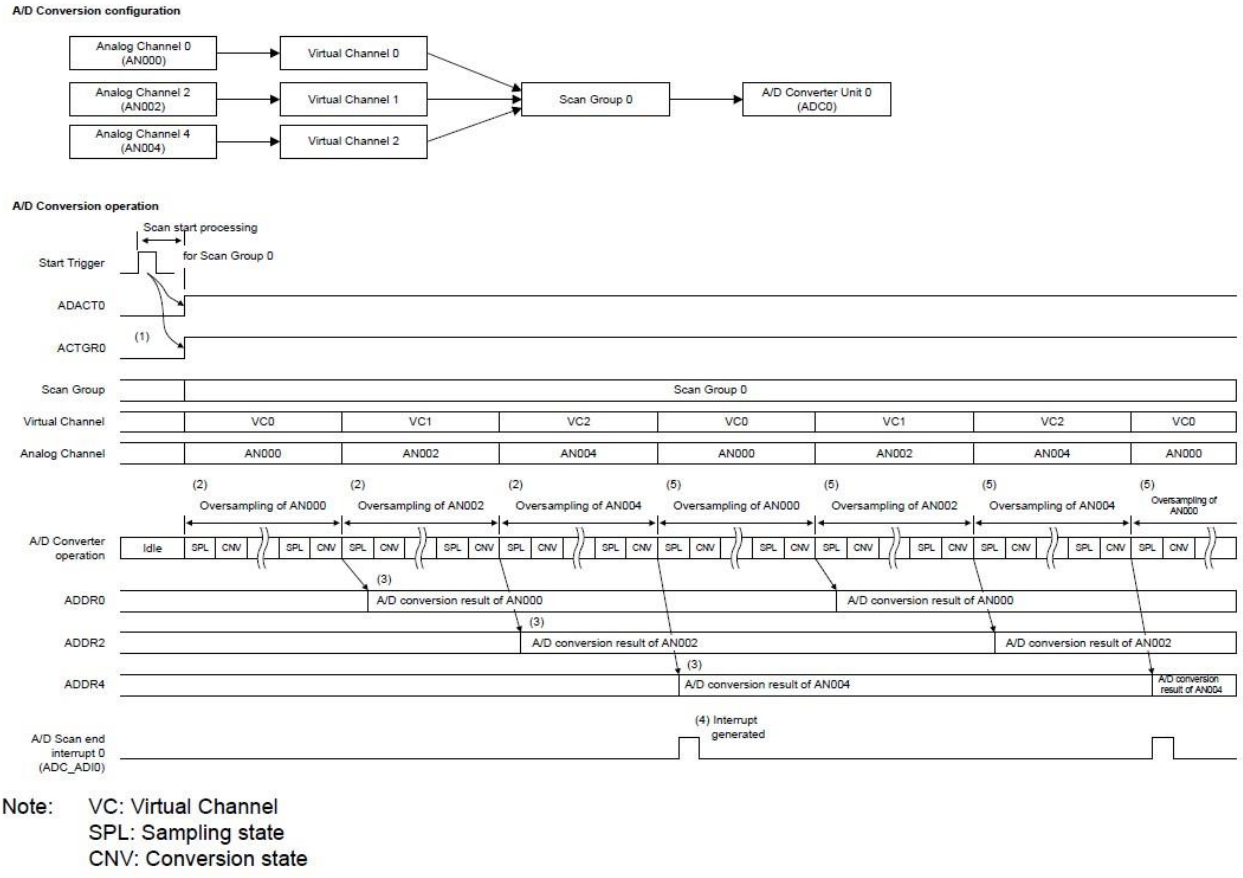


Figure 5. Timing diagram of Oversampling Mode – Continuous Scan Mode

In this example ADC analog channels 0, 2, and 4 are assigned to virtual channels 0, 1, and 2, respectively. All the virtual channels are assigned to scan group 0 and are converted by ADC0. The order of scanning of the virtual and analog channels are shown. Each analog channel is oversampled to convert one ADC data sample and then the scanning operation repeats continuously.

2.2.3 Hybrid Mode – Single Scan Mode

The following steps detail the procedure of Hybrid Mode – Single-Scan Mode:

1. In hybrid mode – single scan mode, scanning operation begins when a software trigger or a peripheral module trigger is input. The scan group corresponding to the trigger is started.
2. In Hybrid mode, the scanning operation is performed while switching analog channels every time oversampling is performed.
3. After the initial delay time (time to oversample enough values to fill the taps of the digital filter) and the adding/averaging time for each analog channel, the A/D conversion data is stored in the data register (ADDRi (i = 0 to 28), ADEXDRj (j = 0 to 2, 5 to 8)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 8)).
4. If the scan end interrupt is enabled, when the A/D conversion of all virtual channels assigned to that scan group is completed, the scan end interrupt corresponding to that scan group is generated.
5. During the scanning operation, ADGRSR.ACTGRn (n = 0 to 8) bit corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0, 1) bit corresponding to the A/D converter performing the A/D conversion is also set to 1. When scanning operation is complete, each bit is cleared to 0 and A/D converter enters the idle state. And the oversampled data in the digital filter is also discarded at the end of the scanning operation.

Following is a timing diagram of the scanning operation in hybrid mode – single scan mode:

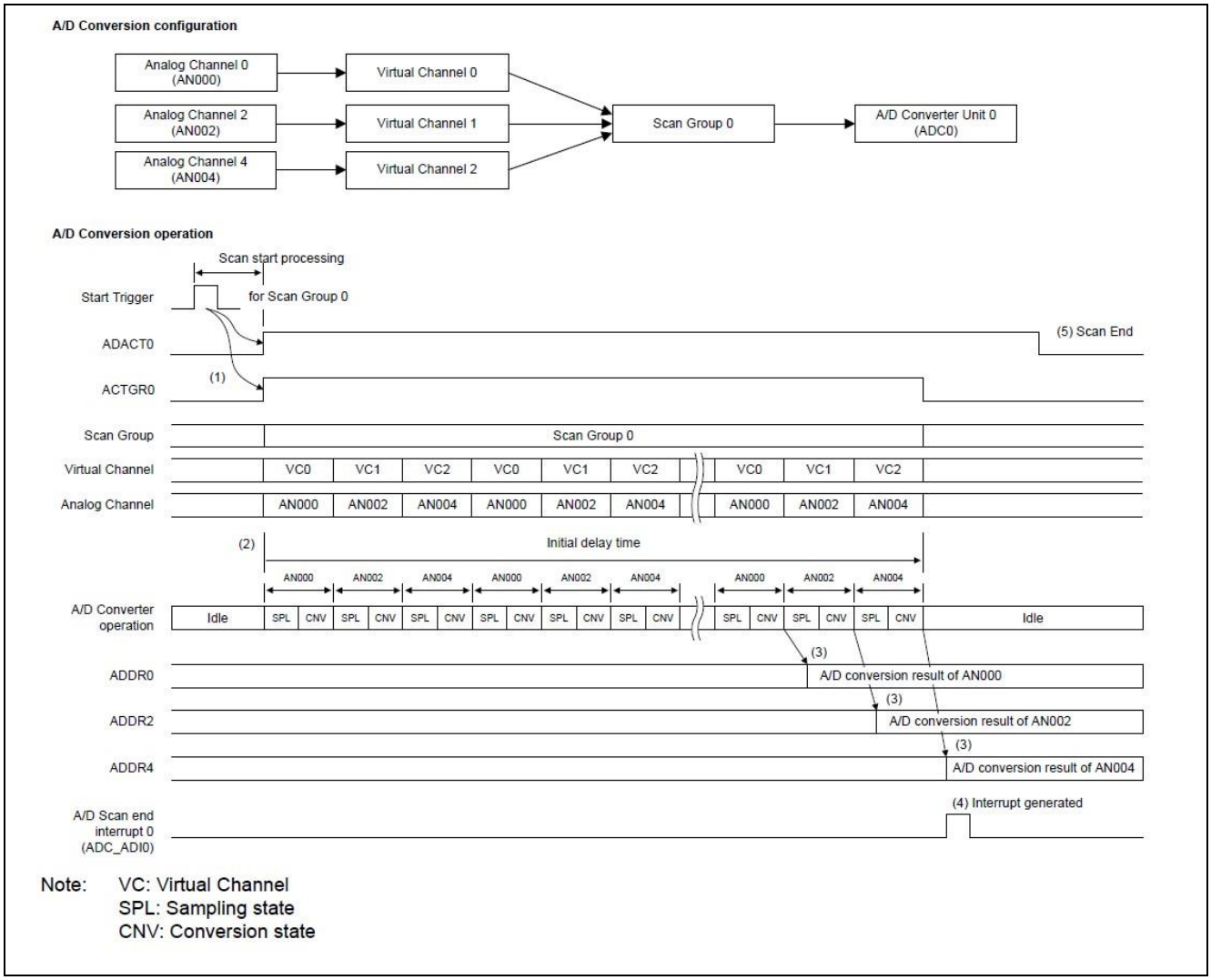


Figure 6. Timing diagram of Hybrid Mode – Single-Scan Mode

In this example ADC analog channels 0, 2, and 4 are assigned to virtual channels 0, 1, and 2, respectively. All the virtual channels are assigned to scan group 0 and are converted by ADC0. The order of scanning of the virtual and analog channels are shown. Each analog channel is oversampled in a hybrid fashion to convert one ADC data sample.

2.2.4 Hybrid Mode – Continuous Scan Mode

The following steps detail the procedure of Hybrid Mode – Continuous Scan Mode:

1. In hybrid mode – continuous scan mode, scanning operation begins when a software trigger or a peripheral module trigger is input. The scan group corresponding to the trigger is started.
2. When scanning operation starts, ADGRSR.ACTGRn (n = 0 to 8) corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0, 1) bit corresponding to the A/D converter performing the A/D conversion is also set to 1.
3. In Hybrid mode, the scanning operation is performed while switching analog channels every time oversampling is performed.
4. After the initial delay time (time to oversample enough values to fill the taps of the digital filter) and the adding/averaging time for each analog channel, the A/D conversion data is stored in the data register (ADDRi (i = 0 to 28), ADEXDRj (j = 0 to 2, 5 to 8)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 8)).
5. If the scan end interrupt is enabled, when the A/D conversion of all virtual channels assigned to that scan group is completed, the scan end interrupt corresponding to that scan group is generated.
6. The subsequent rounds of scanning operations are performed while the oversampling data stored in the digital filter is retained. After the group delay time (time to oversample another value to obtain subsequent A/D conversion data in continuous scan operation) and the adding/averaging times for each analog channel, the A/D conversion data is stored in the data register (ADDRi (i = 0 to 28), ADEXDRj (j = 0 to 2, 5 to 8)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 8)).
7. If the scan end interrupt is enabled, when the A/D conversion of all virtual channels assigned to that scan group is completed, the scan end interrupt corresponding to that scan group is generated.
8. Until the A/D conversion stop process is performed, steps 6 and 7 are repeated, and the scanning operation continues.

Following is a timing diagram of the scanning operation in hybrid mode – continuous scan mode:

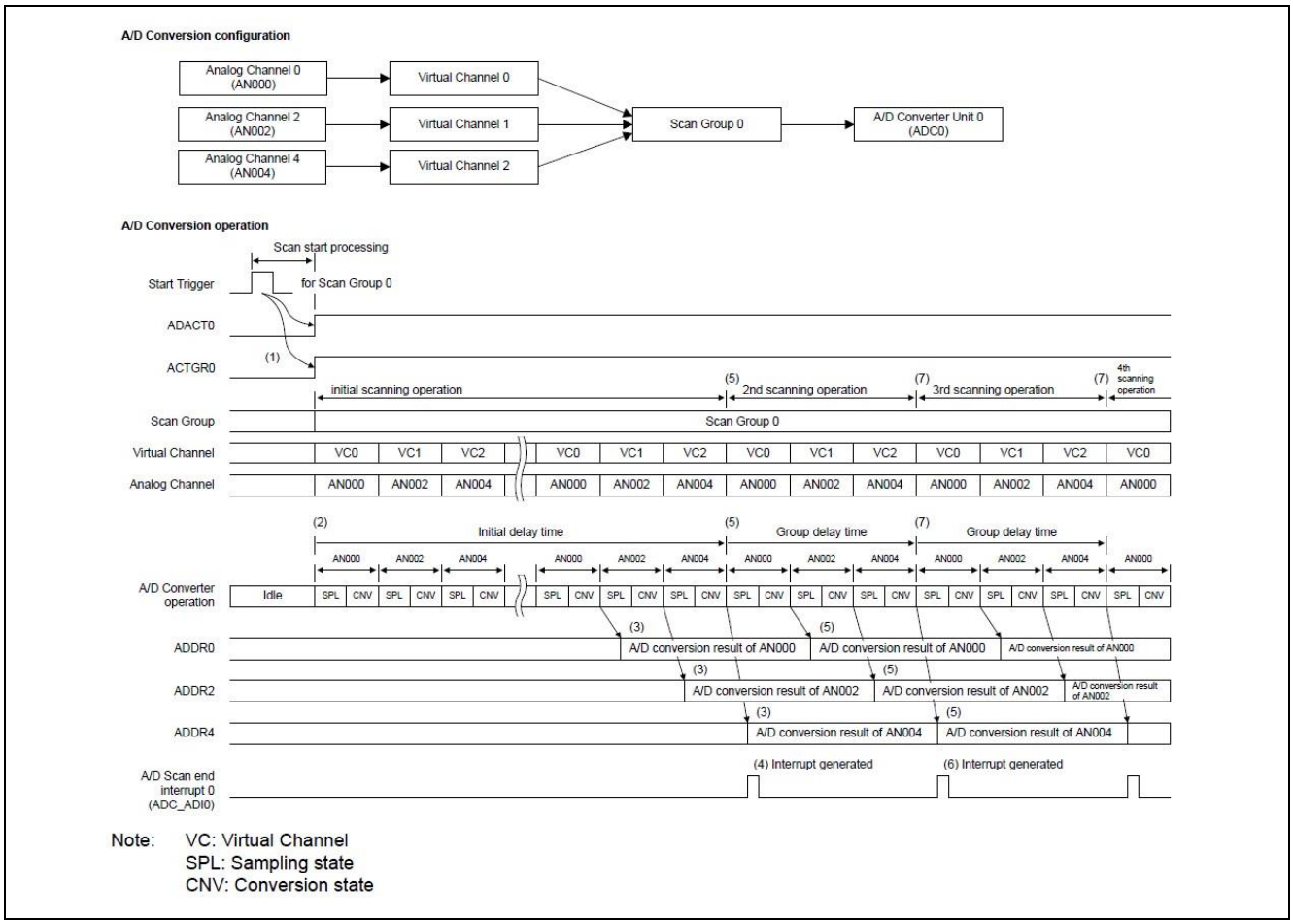


Figure 7. Timing diagram of Hybrid Mode – Continuous Scan Mode

In the above example ADC analog channels 0, 2, and 4 are assigned to virtual channels 0, 1, and 2, respectively. All the virtual channels are assigned to scan group 0 and are converted by ADC0. The order of scanning of the virtual and analog channels are shown. Each analog channel is oversampled in a hybrid fashion to convert one ADC data sample and then that scanning operation repeats continuously.

2.2.5 Hybrid Mode – Background Continuous Scan Mode

Steps detailing the procedure of Hybrid Mode – Background Continuous Scan Mode:

1. In hybrid mode – background continuous scan mode, scanning operation begins when a software trigger or a peripheral module trigger is input. The scan group corresponding to the trigger is started.
2. When scanning operation starts, ADGRSR.ACTGRn (n = 0 to 8) corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0, 1) bit corresponding to the A/D converter performing the A/D conversion is also set to 1.
3. In Hybrid mode, the scanning operation is performed while switching analog channels every time oversampling is performed.
4. After the initial delay time (time to oversample enough values to fill the taps of the digital filter) and the adding/averaging time for each analog channel, the A/D conversion data is ready to be output. In the background continuous scan mode, A/D conversion data can be obtained after the initial delay time has elapsed from the start of the scanning operation.
5. The subsequent rounds of scanning operations are performed while the oversampling data stored in the digital filter is retained. After the group delay time (time to oversample another value to obtain subsequent A/D conversion data in continuous scan operation) and the adding/averaging times for each analog channel, the A/D conversion data is ready to be output.
6. When an A/D conversion start trigger is input during background continuous scanning operation, the most recent A/D conversion data at that time is stored in the data register (ADDRi (i = 0 to 28), ADEXDRj (j = 0 to 2, 5 to 8)). If FIFO function is set to enabled, A/D conversion data is also stored in the FIFO data register (ADFIFODRk (k = 0 to 8)).
7. If scan end interrupt is enabled the scan end interrupt is generated.
8. Until A/D conversion stop operation is performed, the background continuous scanning operation (step 5) is repeated. Whenever the A/D conversion start trigger is input during background continuous scanning mode, the A/D conversion data is output (steps 6 and 7).

Following is a timing diagram of the scanning operation in hybrid mode – continuous scan mode:

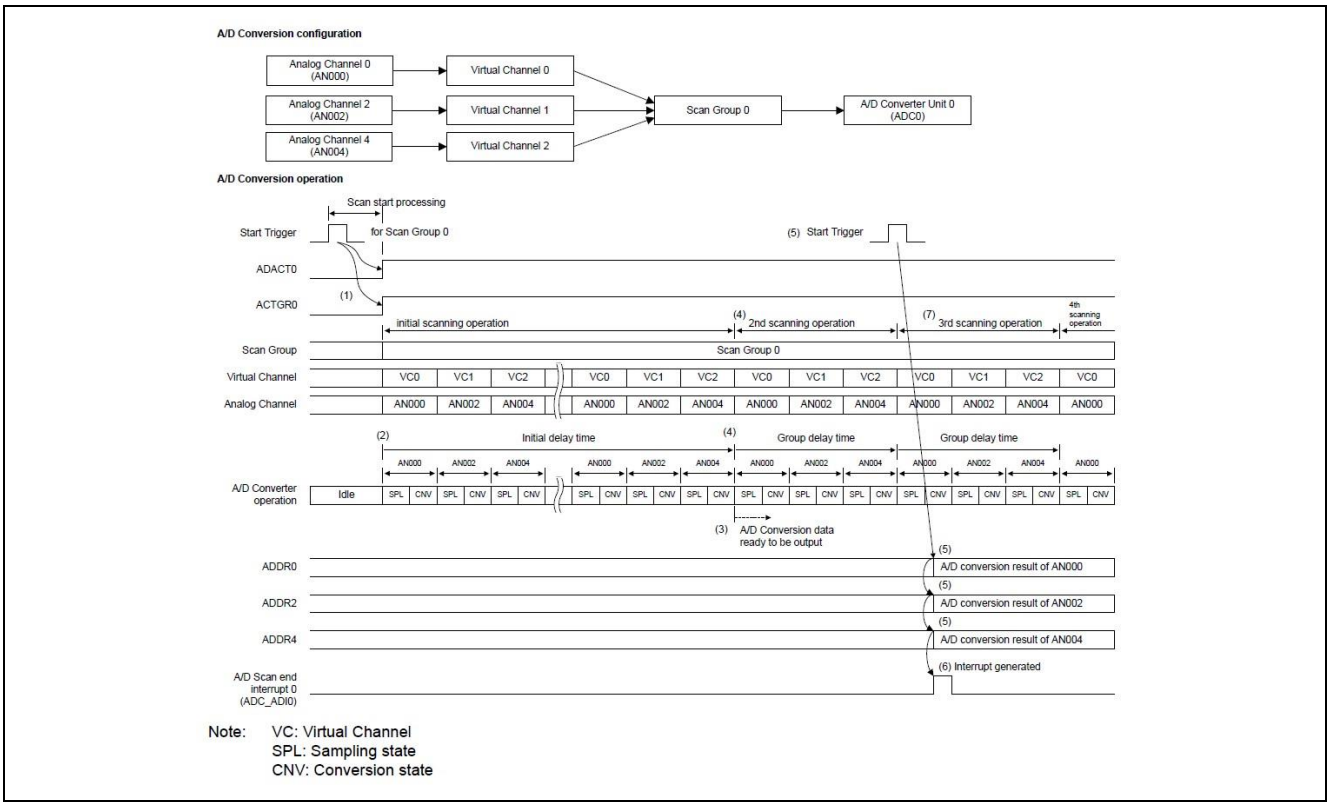


Figure 8. Timing diagram in Hybrid Mode – Continuous Scan Mode

In this example ADC analog channels 0, 2, and 4 are assigned to virtual channels 0, 1, and 2, respectively. All the virtual channels are assigned to scan group 0 and are converted by ADC0. The order of scanning of the virtual and analog channels are shown. Each analog channel is oversampled in a hybrid fashion and when the start trigger is generated then one ADC data sample is converted and this scanning operation repeats continuously.

2.2.6 Digital Filter Function

The digital filter function is an integral part of obtaining high-precision and high-resolution 16-bit depth A/D conversion results. Because of this, its use is required in Oversampling mode and Hybrid mode. The use of the digital filter in SAR mode is prohibited.

This section describes the characteristics and configurations of the digital filter function and explains how to operate the filter.

2.2.6.1 Configuration and Characteristics

Each ADC unit has 4 digital filters. The digital filter is an FIR type filter with 22 taps. Before operation of the ADC, the digital filter must be configured. There are two preset filters to choose from: the sinc filter (referred to as sync3 filter in the FSP configuration) or the minimum phase filter.

The frequency response of the sinc3 filter is shown in the following image:

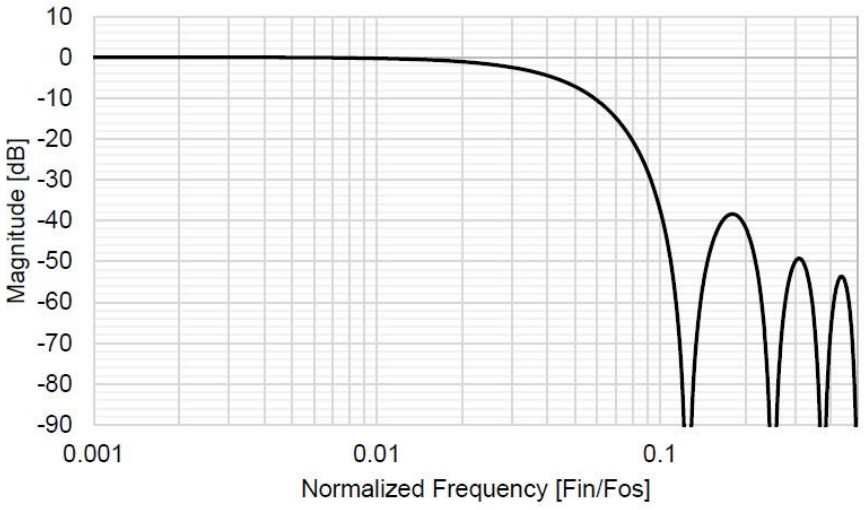


Figure 9. Normalized Frequency Response of the Sinc3 Filter

The frequency response of the minimum phase filter is shown in the following image:

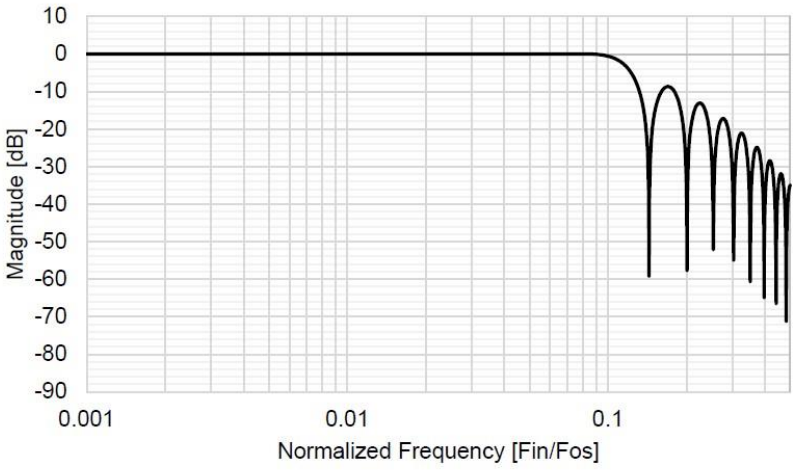


Figure 10. Normalized Frequency Response of the Minimum Phase Filter

For more information regarding the characteristics of the sinc3 filter and the minimum phase filter, please refer to the Electrical Characteristics chapter of the RA6T2 HW UM.

The digital filter is set by ADDOPCRAn.DFSEL[2:0] (n = 0 to 36) and ADDFSRm (m = 0, 1) registers. Following is a block diagram illustrating the digital filter configuration:

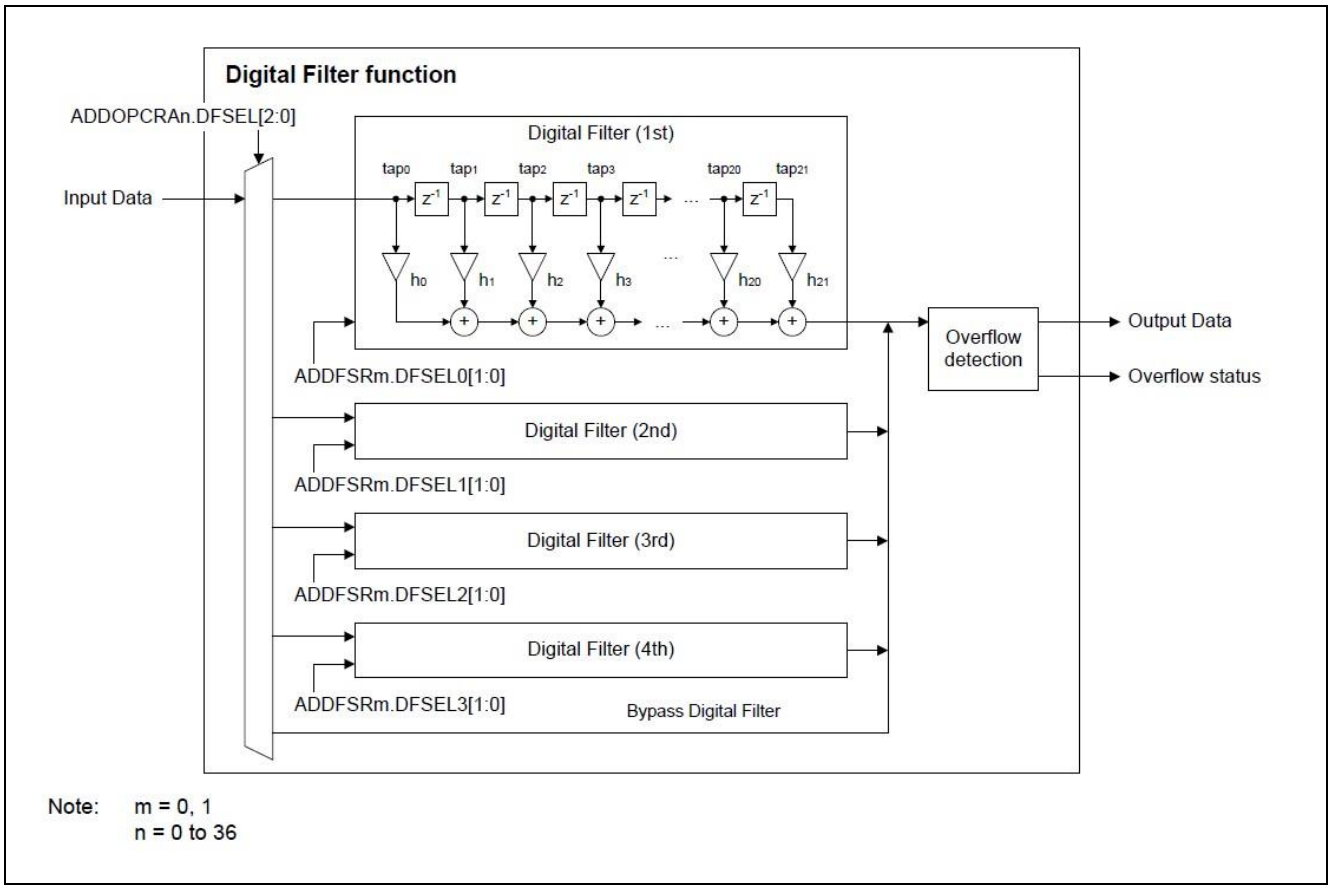


Figure 11. Block Diagram of the Digital Filter

2.2.6.2 Operating the Digital Filter

When the digital filter function is enabled, A/D conversion data is input sequentially into the digital filter. When all the taps in the digital filter are filled with data, the calculation result is output and sent to the next data processing.

The operation of the digital filter function varies slightly between Oversampling mode and Hybrid mode.

(1) Digital Filter Operation – Oversampling Mode

In oversampling mode, the A/D converter continuously oversamples one analog channel. Each time oversampling is performed, the A/D conversion data is input sequentially into the digital filter. Once all the taps of the filter are populated with conversion data, the calculation result is output from the filter and sent to the next data processing.

The data of the taps in the digital filter is discarded when the A/D conversion data is output to the next data processing step. However, when A/D-converted value addition/averaging function is used, the data of taps in the digital filter is kept until the data required to calculate the addition value or average value of the A/D converted value is collected. When the A/D-converted value addition/averaging value is calculated, the tap data in the digital filter is discarded.

(2) Digital Filter Operation – Hybrid Mode

Hybrid mode can process oversampled data (A/D converted data) of up to four analog channels in parallel by using several digital filter circuits at the same time. Each time oversampling is performed, the A/D conversion data is input sequentially into the digital filter. Once all the taps of the filter are populated with conversion data, the calculation result is output from the filter and sent to the next data processing.

In Hybrid mode – Single scan mode, taps in the digital filter are discarded at the end of scanning operation.

In Hybrid mode – Continuous scan mode, the data of the taps in the digital filter continues to be updated as long as the continuous scanning operation is continued. Therefore, after the data is filled in all taps in the digital filter, a new calculation result is output for each oversampling. When scanning operation is aborted due to a forced stop of A/D conversion, the tap data in the filter is discarded.

The operation of the digital filter in Hybrid mode – Background continuous scan mode is the same as in Hybrid mode – Continuous scan mode. During the background continuous scanning operation, the digital filters and other data processing continue to take place in the background. When scanning operation is aborted due to a forced stop of A/D conversion, the tap data in the filter is discarded.

2.3 ADC Timing Considerations

This section highlights a couple of timing considerations for operating the ADC units.

2.3.1 Sampling Rate Calculation

The sampling time required in A/D conversion for externally input analog signal is determined by the charging time to the sampling capacitance in A/D converter. The simplified model of the circuit is shown in Figure 12. The sampling time can be roughly estimated by the following equation:

$$t_{SPL} = (R_{EXT} + R_{AD}) \times (C_{EXT} + k_{CAD}) \times \ln [k_{CAD} / (C_{EXT} + k_{CAD}) \times (2^N / M)]$$

t_{SPL}: Estimated sampling time

C_{EXT}: External capacitance (pin capacitance + PCB parasitic capacitance)

C_{AD}: Internal sampling capacitance

R_{EXT}: External input signal source Impedance

R_{AD}: Internal resistance

k: Correction coefficients according to operation mode

N: Target conversion resolution (16, 14, 12 or 10)

M: Sampling error based on 1 LSB in N-bits A/D converter (1/4, 1/2, 1, 2, or 4 LSB etc.)

The typical values for each parameter are as follows as a reference:

- Pin capacitance at analog input pin: 5 pF
- Internal sampling capacitance (CAD): 5 pF
- R_{AD} at High-speed channels: 0.7 kΩ
- R_{AD} at High-precision channels: 1.2 kΩ
- R_{AD} at Normal-precision channels: 3.0 kΩ
- k at SAR mode: 1.2
- k at Oversampling mode: 1.0
- k at Hybrid mode: 1.2

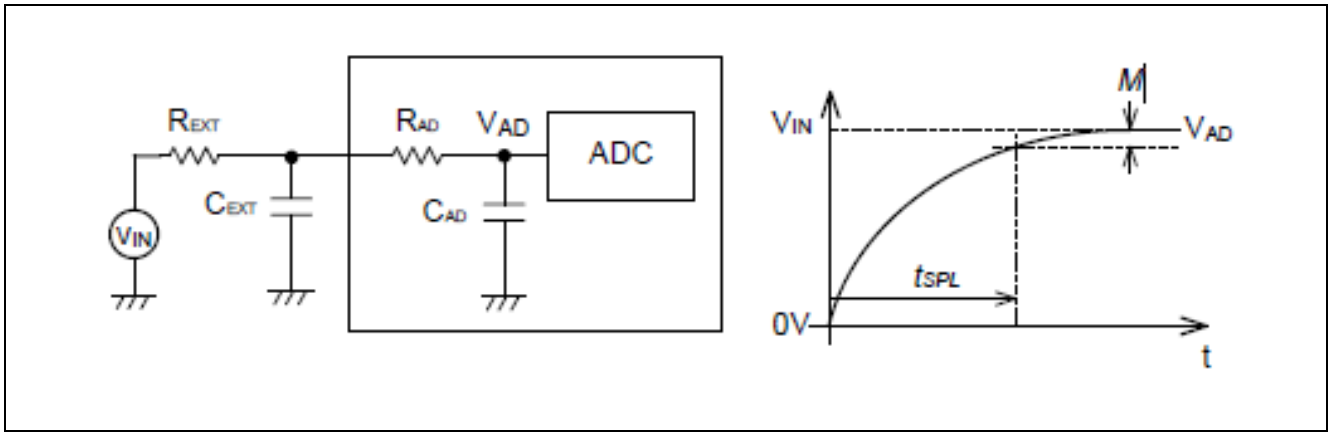


Figure 12. Simplified Circuit Model and the Sampling Time Curve to Charge Capacitance

In the given equation, time is estimated by the time it takes for the difference between the analog input voltage (V_{IN}) and the sampling capacitance voltage (V_{AD}) to be less than or equal to sampling error based on N-bits A/D converter.

This equation is a simplified view of general use case. It should only be used to roughly estimate the sampling time and does not guarantee the accurate sampling time. Especially in the case of normal-precision channels, the accuracy for the estimate for sampling time deteriorates if (2^N / M > 16384).

2.3.1.1 Note on 16-bit Conversion Methods

Calculating the maximum range of analog input signal frequency for 16-bit conversion methods depends on the maximum sampling frequency as well as the normalized cutoff frequency of the digital filter that is selected.

The reciprocal of the sampling time gives the sampling frequency. Multiplying by the normalized cutoff frequency for the corresponding digital filter selected gives the upper limit of the input frequency. The normalized cutoff frequencies are highlighted in the following table:

Table 9. Normalized Cutoff Frequencies of the Digital Filters

Parameter				Min	Typ	Max	Unit	Test conditions
Oversampling mode and Hybrid mode	Analog input voltage range	Single-ended input voltage		VREFL0	—	VREFH0	V	—
		Differential input voltage*1		-VREFH0	—	+VREFH0	V	—
	Resolution			—	—	16	bit	—
	Oversampling period	Oversampling mode		0.16	—	—	µs	• ADCLK: 50 MHz • Sampling time: 3 ADCLK • Successive approximation time: 5 ADCLK • Without disconnection detection assist function
		Hybrid mode		0.18	—	—	µs	• ADCLK: 50 MHz • Sampling time: 4 ADCLK • Successive approximation time: 5 ADCLK • Without disconnection detection assist function
	Digital filter characteristics *2	Sinc filter	Initial delay	—	22	—	Fos	—
			Group delay	—	11	—		—
			Normalized Cutoff Frequency	—	0.033	—	Fin/Fos	—
		Minimum phase filter	Initial delay	—	22	—	Fos	—
			Group delay	—	2	—		—
			Normalized Cutoff Frequency	—	0.116	—	Fin/Fos	—
			Passband ripple	—	<± 0.01	—	dB	—

Note: Fos is oversampling frequency.
Note 1. Differential input voltage is (A_{INP} - A_{INN})
- A_{INP} is input voltage of ANx, and VREFL0 ≤ A_{INP} ≤ VREFH0.
- A_{INN} is input voltage of ANy, and VREFL0 ≤ A_{INN} ≤ VREFH0.
  (x = 2i, y = 2i +1, i = 0, 1, 2… (any integer))

Example Calculation:

For a conversion time of 1460 ns, at least 1 additional cycle of 20 ns is needed for resampling the sample and hold circuit. This gives a total of 1480 ns period = 675.7 KHz

Sinc filter: 675.7 KHz x 0.033 = 23.0 KHz maximum input frequency

Minimum phase filter: 675.7 KHz x 0.116 = 78.4 KHz maximum input frequency

2.3.2 Clock Rate

A/D conversion clock (ADCLK) is the operation clock of ADC. The A/D converters (ADC0 and ADC1) are operated and controlled by ADCLK as the basic clock. The following figure shows the clock structure of ADC:

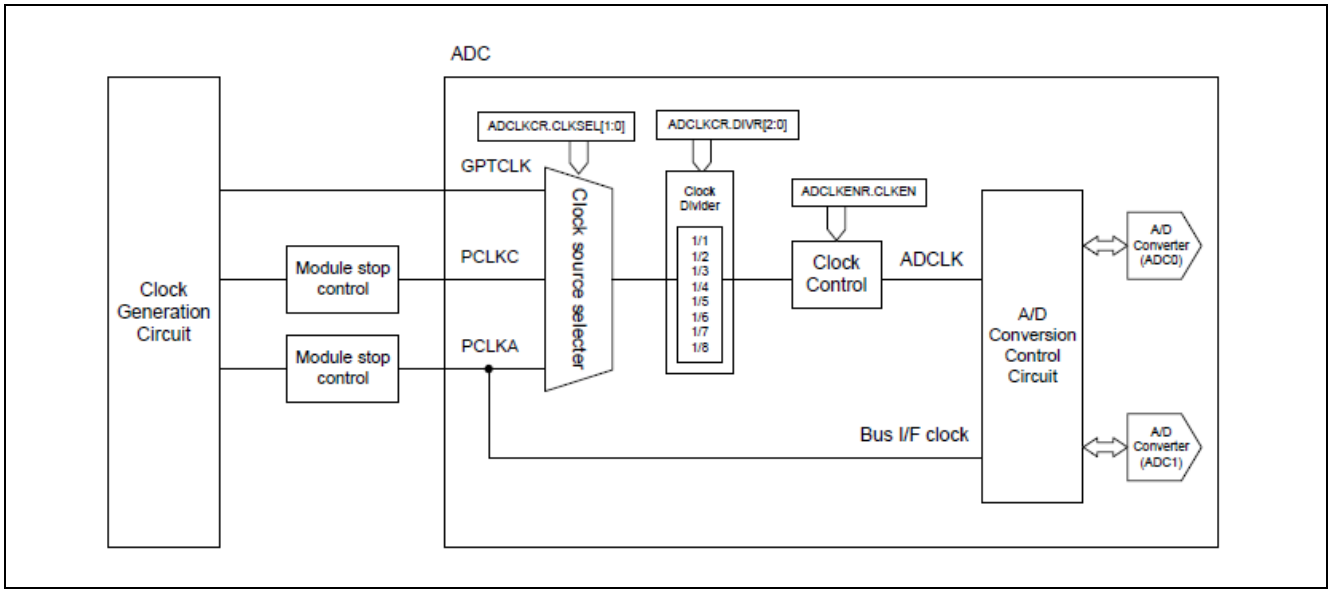


Figure 13. Simplified Clock Structure of the ADC Peripheral

ADCLK is generated from the clock source and the division ratio selected in ADCLKCR register. The frequency of ADCLK should be set so that PCLKA ≥ ADCLK. Also, the frequency of ADCLK should be set so that it is within the guaranteed operating range of 25 MHz minimum to 60 MHz maximum, as specified in the electrical characteristics chapter of the HW UM.

3. Introduction to the Application Project

It is important to understand the practical implementation of configuring the ADC peripheral through the FSP settings in e^2 studio and using the API functions to create your own application. This application note is supplemented with two application projects that demonstrate operating the ADC module in both 12-bit and 16-bit data conversion modes.

The primary project is called “adc_16bit_oversampling_ra6t2” and is intended to demonstrate the proper FSP configurations and self-calibration operation needed specifically for 16-bit conversion to be guaranteed using Oversampling Mode – Continuous Scan Mode. The steps to run this project begin in section 4.

The secondary project is called “adc_12bit_sar_ra6t2” and is the same data acquisition system, but with the ADC configured for 12-bit conversion using SAR Mode – Continuous Scan Mode. This project is intended to be used for your own comparison purposes. You can run the secondary project with the same input as the primary project to see the performance difference between the 12-bit and 16-bit ADC modes. Steps to run the secondary project are not detailed in this application note as they are the same as the primary project.

The projects differ in their configuration, which determines the ADC conversion method; the code is essentially the same. After running the primary project, you should understand how to set important configurations in the Stacks tab of e^2 studio for 16-bit operation. Additionally, you should understand the importance of self-calibration and the scenarios when it is necessary to use the self-calibration function.

3.1 Project Statement

In the “adc_16bit_oversampling_ra6t2” project the A/D converter will operate in Oversampling Mode – Continuous Scan Mode to obtain 16-bit converted values from the user’s choice of an external analog input, for example, a signal generator, sensor, and so forth.

A block diagram of the application project can be found as follows:

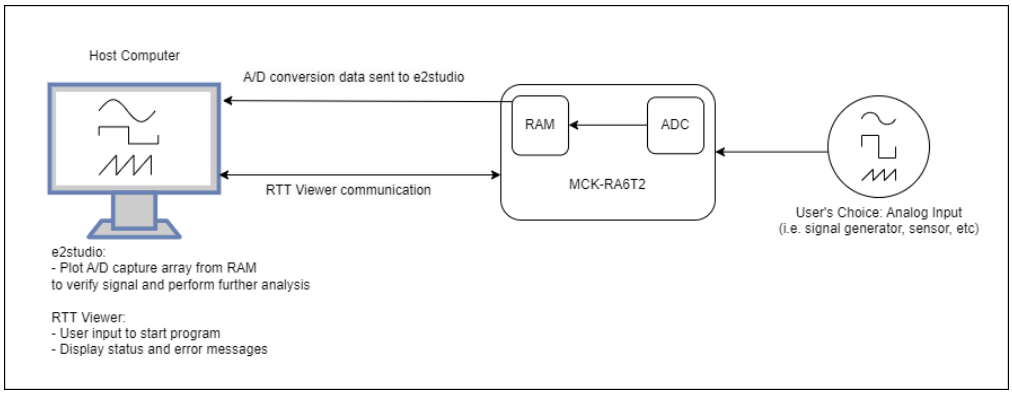


Figure 14. Block Diagram of the Application Project

The host computer runs the two software: e^2 studio and J-Link RTT Viewer. The USB connection from the host computer communicates between both software and the RA6T2 MCU. e^2 studio communicates debug session information with the MCU, and J-Link RTT Viewer sends project commands to the MCU and receives any module errors that occur from the MCU.

The following steps describe the project’s functionality:

1. User will initiate the ADC peripheral’s continuous scan.
2. Analog input is continuously converted until a set number of samples (defaulted to 1024 samples) have been stored in an array in RAM.
3. After 1024 samples are captured, the ADC scanning will halt, and the program will pause.
4. If desired, users can examine and/or export the data for further analysis.
5. User will continue the program when ready to restart the scan procedure.
6. If the conditions require it, the user will choose to run the self-calibration function.

User input, application status print messages, and errors are all handled through the J-Link RTT Viewer software installed to the host computer.

In the secondary project “adc_12bit_sar_ra6t2” the A/D Converter will operate in SAR mode – Continuous Scan mode and the same steps are performed.

3.2 FSP Configurations

This section reviews the important clock settings and stack settings of each application project. In both projects, the FSP configuration is stored in the file named configuration.xml. Double clicking on this file will bring up the FSP Configuration view for the project.

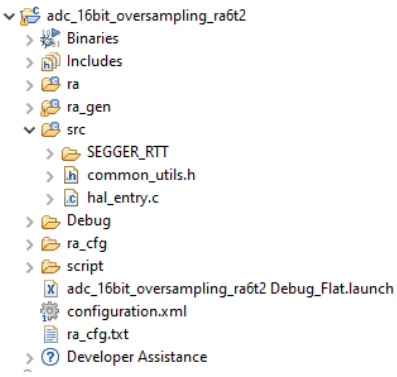


Figure 15. Folder Structure of Application Project

3.2.1 Clock/Pin Settings

The **Clocks** tab is where you can configure the various MCU clock speed settings for your project. For both projects, the A/D conversion clock (ADCLK) source is the peripheral module clock (PCLKC) which is set to 60MHz.

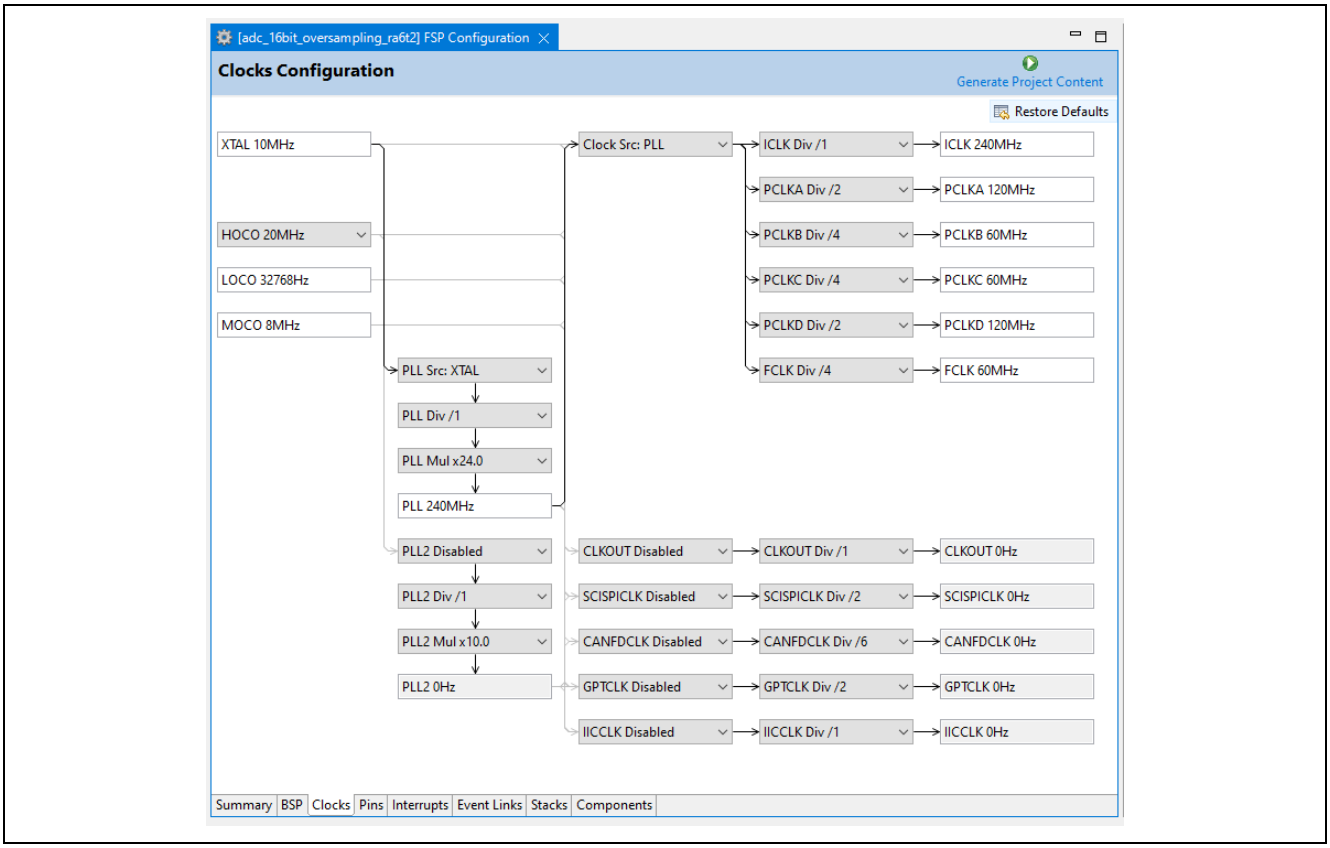


Figure 16. Clock tab of FSP Configuration

The ADCLK can operate between 25 MHz as a minimum and 60 MHz as a maximum. The clock source and additional division ratio settings are configurable for each ADC module that is added to the **Stacks** tab.

3.2.2 Stack Settings

The **Stacks** tab is where you can add and configure peripheral modules for your project. To add a new module, use the **New Stack** button. To add the ADC16 module, use **New Stack** > **Analog** > **ADC Driver on r_adc_b** to add the module directly or use the search functionality by entering **New Stack** > **Search** > **r_adc_b**.

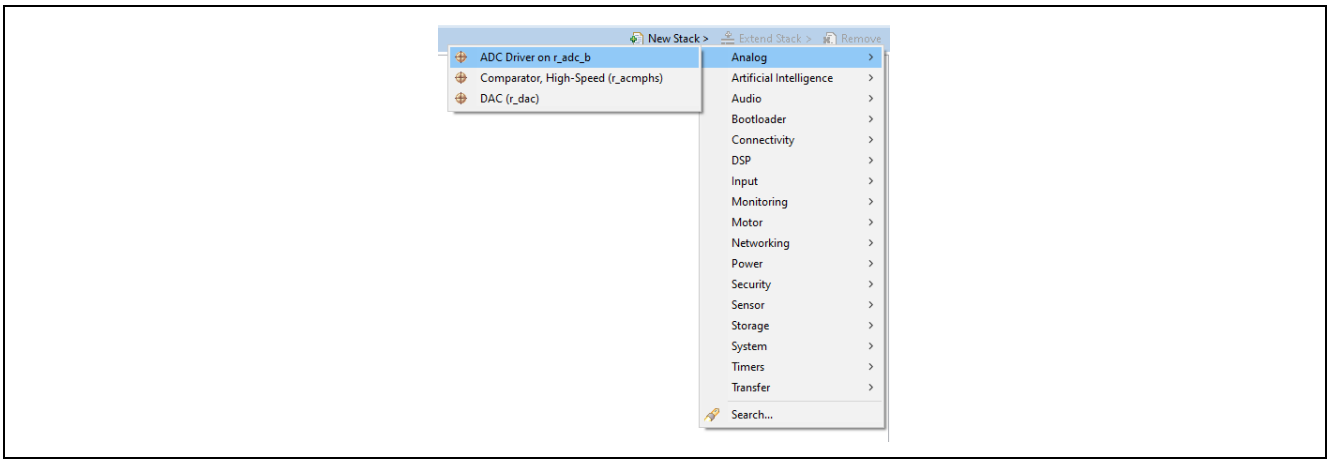


Figure 17. Adding a New r_adc_b Module to Stacks

Clicking on the **r_adc_b** module > **Properties** tab> **Settings** tab brings up the module settings. Each project's settings are explained further below.

3.2.2.1 16-bit Oversampling Mode Project ADC Module Properties

The ADC module settings that differ from the default values for the “adc_16bit_oversampling_ra6t2” project are listed in the following table:

Table 10. ADC module configurations that differ from the default for the adc_16bit_oversampling_ra6t2 project

Module Property Setting	Default Value	Used Value	Reason
ADC Module > General > Operation > ADC 0 > Conversion Method	SAR Mode	Oversampling Mode	Set the conversion method to Oversampling Mode
ADC Module > General > Operation > ADC 0 > Scan Mode	Single Scan	Continuous Scan	Set the scan mode to continuous capture
ADC Module > General > Name	g_adc0	g_adc_16bit	Set the name to indicate the 16-bit capture
ADC Module > Interrupts > Scan End Priority > Groups 1-8	Priority 12	Disabled	Disable interrupts for unused scan groups 1 through 8.
ADC Module > Interrupts > Callback	NULL	cb_scan_end	Create a callback for the scan end interrupt handling.
ADC Module > Virtual Channels > Virtual Channel 0 > Scan Group	None	Scan Group 0	Assigning virtual channel 0 to scan group 0
ADC Module > Virtual Channels > Virtual Channel 0 > Conversion Data Format Select	12-bit data format	16-bit data format	Enable 16-bit capture
ADC Module > Virtual Channels > Virtual Channel 0 > Digital Filter Selection	Disabled	Digital Filter Table Entry 1	Digital filter is required for 16-bit conversions to be guaranteed.
ADC Module > Scan Groups > Scan Group 0 > Enable	Disable	Enable	Enable scan group 0 to capture the virtual channels assigned to it when conversion begins.

Note: For ADC0 to read and convert the input signal from analog channel 0 (AN000): AN000 is assigned to virtual channel 0, virtual channel 0 is assigned to scan group 0, and scan group 0 is assigned to ADC0.

3.2.2.2 12-bit SAR Mode Project ADC Module Properties

The ADC module settings that differ from the default values for the “adc_12bit_sar_ra6t2” project are listed in the following table:

Table 11. ADC module configurations that differ from the default for the adc_12bit_sar_ra6t2 project

Module Property Setting	Default Value	Used Value	Reason
ADC Module > General > Operation > ADC 0 > Scan Mode	Single Scan	Continuous Scan	Set the scan mode to continuous capture
ADC Module > General > Name	g_adc0	g_adc_12bit	Set the name to indicate the 12-bit capture
ADC Module > Interrupts > Scan End Priority > Groups 1-8	Priority 12	Disabled	Disable interrupts for unused scan groups.
ADC Module > Interrupts > Callback	NULL	cb_scan_end	Create a callback for the scan end interrupt handling.
ADC Module > Virtual Channels > Virtual Channel 0 > Scan Group	None	Scan Group 0	Assigning virtual channel 0 to scan group 0
ADC Module > Scan Groups > Scan Group 0 > Enable	Disable	Enable	Enable scan group 0 to capture the virtual channels assigned to it when conversion begins.

3.3 FSP APIs Used

The FSP APIs that both application projects use and their explanations for use are listed in the following table:

Figure 18. List of FSP API Usage for Both Application Projects

API Name	Usage
R_ADC_B_Open	Initialize and open the ADC0 peripheral.
R_ADC_B_ScanCfg	Configure the ADC0 peripheral as specified in the module settings from the Stacks tab of the FSP configuration.
R_ADC_B_ScanGroupStart	Starts the scanning procedure for scan group 0 which is assigned to ADC0.
R_ADC_B_StatusGet	Obtain the ADC0 status. Used to determine when self-calibration is complete and determine if A/D conversion is complete for each sample.
R_ADC_B_Read	Read the sample from ADC0.
R_ADC_B_Calibrate	Perform the self-calibration function for ADC0.
R_ADC_B_ScanStop	Stop the scanning procedure for scan group 0.
R_BSP_SoftwareDelay	Used to wait for the calibration procedure to end.

4. Running the Application Project

This section provides detailed instructions for importing and running the “adc_16bit_oversampling_ra6t2” application project in e² studio. The same steps apply for the “adc_12bit_sar_ra6t2” project.

4.1 Importing and Building the Application Project

The following instructions will show you how to import the example project into your e² studio workspace.

Open e² studio on your host computer and select the option **File > Import… > Existing Projects into Workspace** and click **Next.**

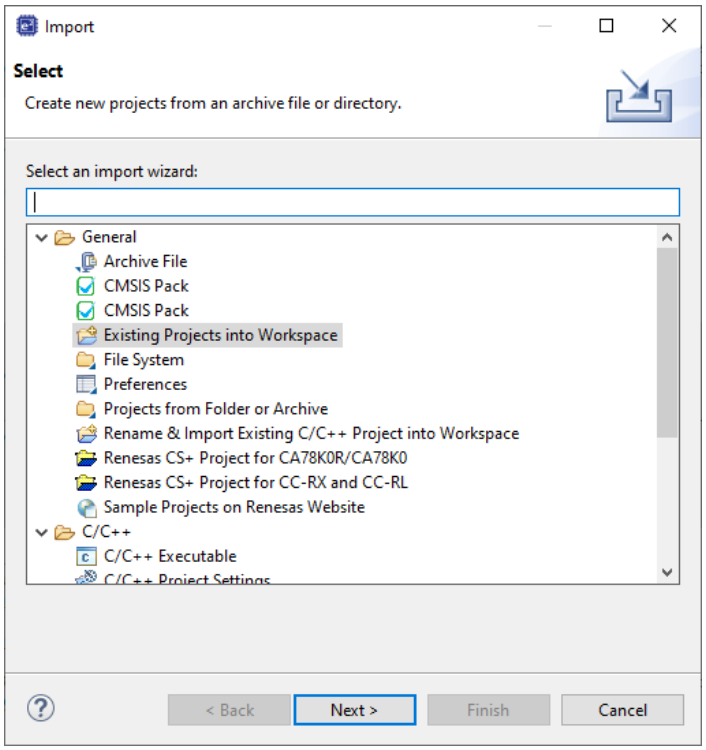


Figure 19. Import Existing Project into Workspace

Browse to the location of application project titled **adc_16bit_oversampling_ra6t2** and click **Finish.**

Once the project has finished importing, double click on the `configuration.xml` file in the workspace Project Explorer to open the FSP Configuration view.

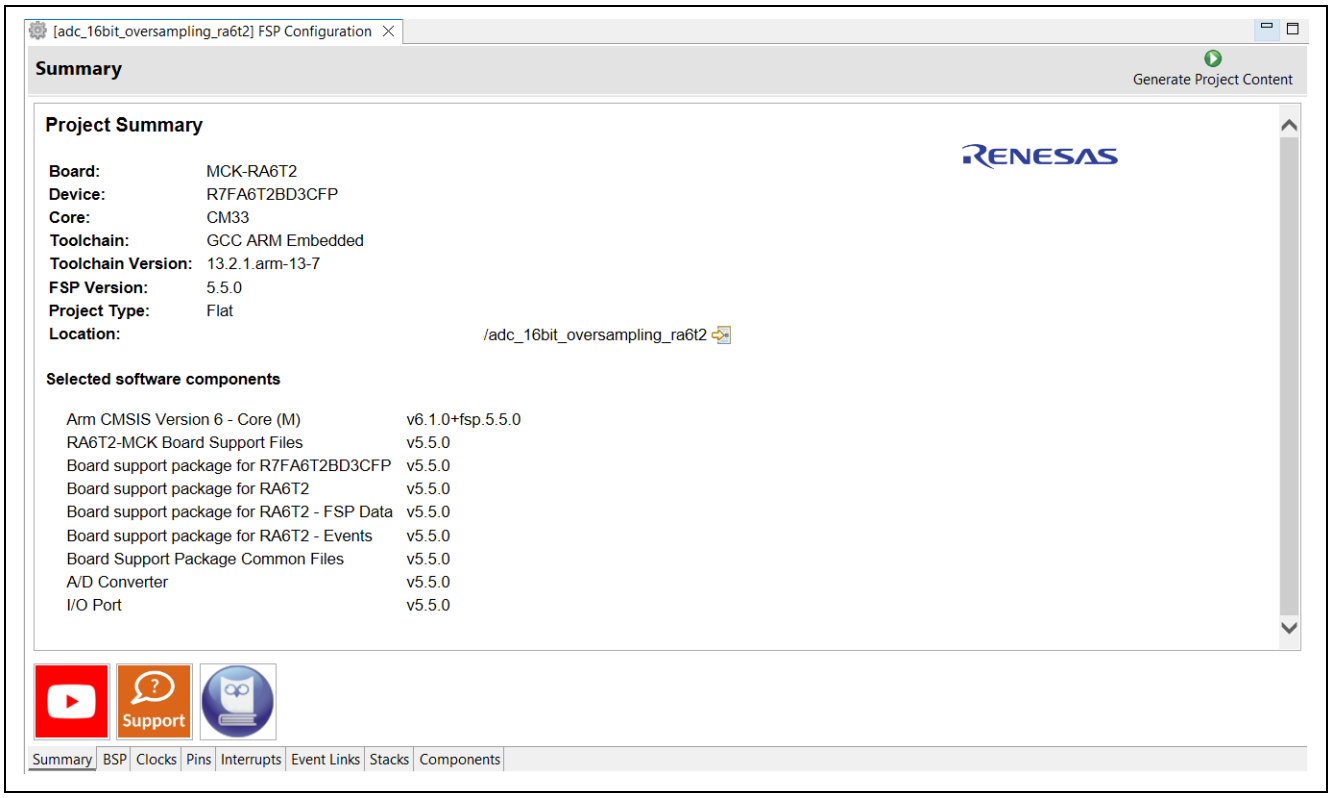


Figure 20. Overview of Project's Summary when FSP Configuration Opens

You can check the project's configurations by reviewing the Clocks, Pins, and Stacks tab. Once you are ready to proceed, click **Generate Project Content**.

4.2 Running the Application Project

Connect the USB cable to the MCK-RA6T2 board and to a port on your workstation.

Connect your analog input signal to AN000 and ground pins on the RA6T2. AN000 corresponds to the physical analog channel that is assigned to be scanned by ADC0 and pinout is at CN4 pin 9.

The file "hal_entry.c" in the project > src folder contains the code for the application project.

Click on the  button to build and compile the project.

Click on the  button to start a debug session for the project.

Open J-Link RTT Viewer and in the **Specify Target Device** section click **…** next to the drop down and browse for the corresponding target device. Click **OK.**

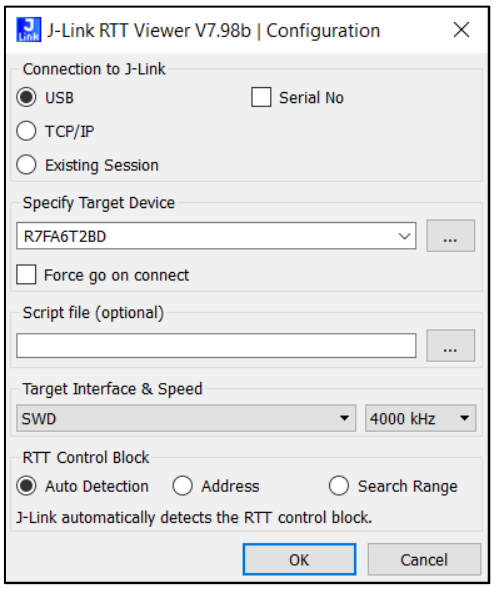


Figure 21. Setting up J-Link with the RA6T2

Note: On occasion, the default Auto Detection of the RTT Control Block does not successfully connect RTT Viewer to the MCU. If no communication is happening, you will need to reconnect and enter the address corresponding to the **'._SEGGER_RTT'** control block in the project's `.map` file. The .map file is located in the **Debug** folder in e^2 studio.

To manually connect RTT Viewer to the MCU perform the following: select **Address** in the RTT Control block and enter **0x200004a4 .** This address is the same for both projects.

In e^2 studio, click the **Resume** button  twice to run the project.

User input prompts, status messages and error messages are printed to J-Link RTT Viewer. Follow the prompts to operate the project. To begin the scanning operation, press '1' on your keyboard.

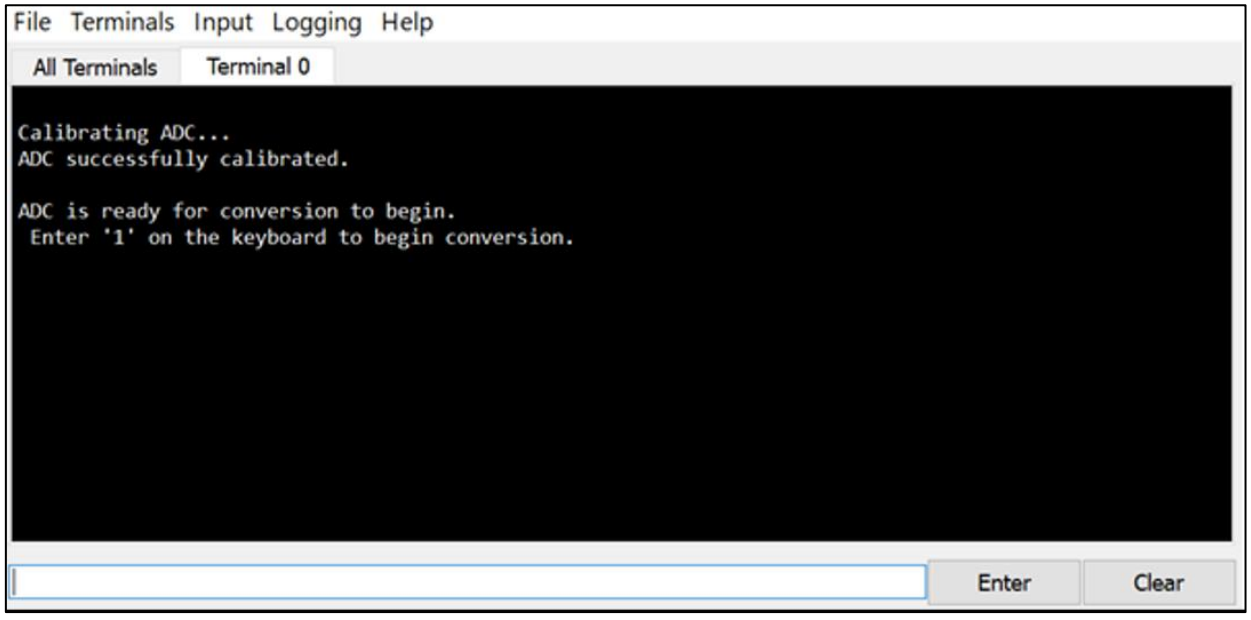


Figure 22. Prompt to Begin Scanning Operation

The scanning operation continuously captures the analog input until NUM_SAMPLES samples are captured, after which the program waits. At this point, you can pause the program and verify the ADC data.

Visit Section 5 Verifying the Application Project for the detailed steps on verifying the ADC data.

Once ready to continue, click the Resume button  if you had paused the program to verify ADC data. Enter ‘2’ on keyboard through J-Link RTT Viewer to restart the program.

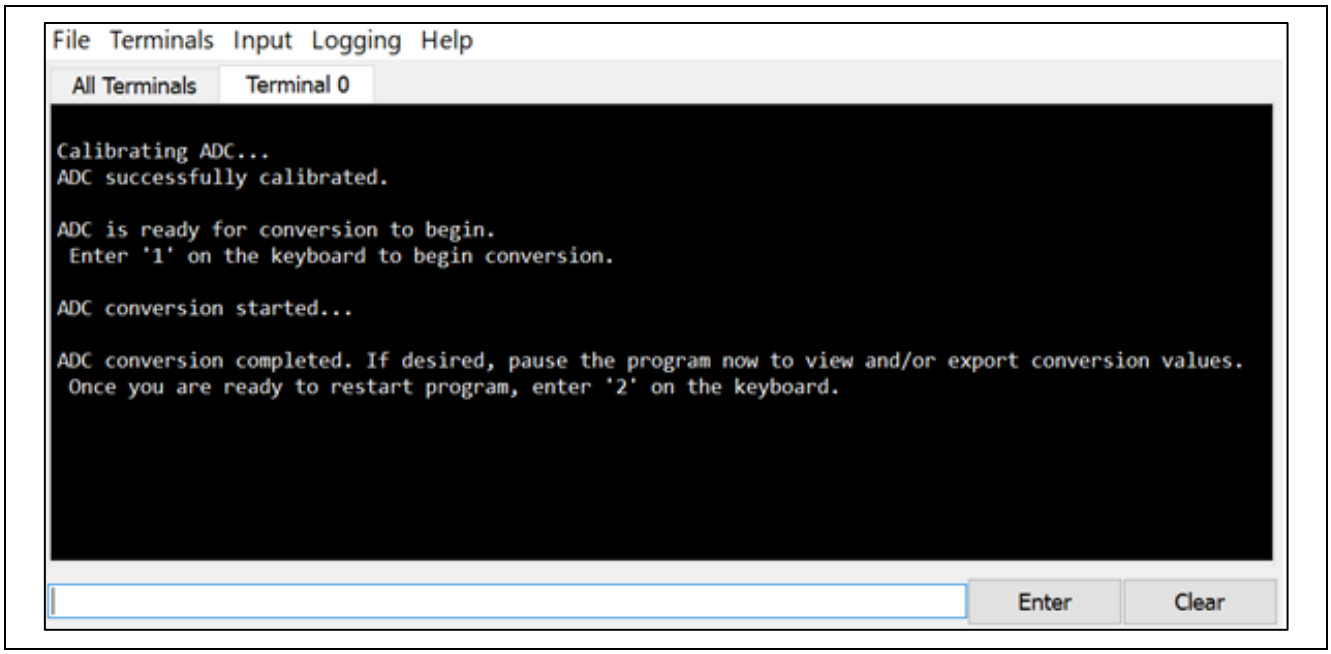


Figure 23. Prompt to Restart the Program

If conditions have changed that require ADC self-calibration between subsequent scanning operations, you can choose to re-run the calibration procedure. Enter ‘3’ on your keyboard through J-Link RTT Viewer to perform self-calibration. If self-calibration is not required enter ‘4’ on your keyboard to skip. See Section 2.1.9.1 for conditions when self-calibration is required.

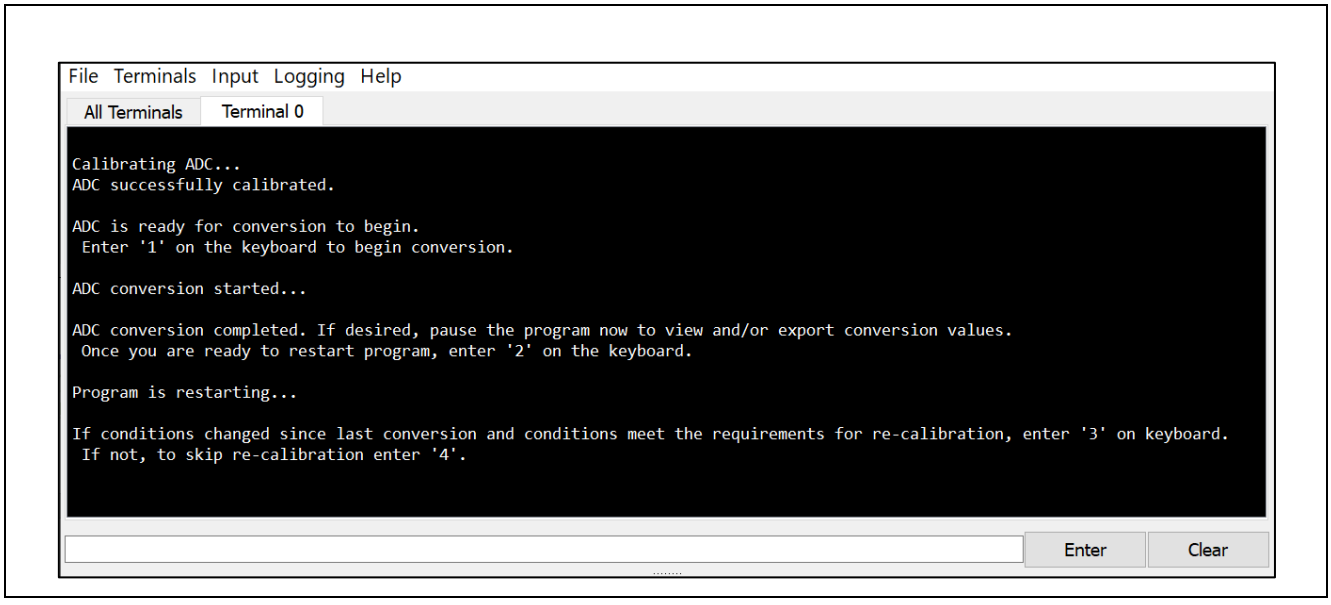


Figure 24. Prompt to Re-calibrate the ADC

Project restarts from the beginning, and you can run further conversions to observe the operation of the ADC. Follow the prompts through J-Link RTT Viewer to operate subsequent input signal scans and captures.

5. Verifying the Application Project

The project’s scanning operation continuously captures the analog input signal until NUM_SAMPLES samples have been converted. Then, the program waits until user input through J-Link RTT Viewer restarts the program, and the next scanning operation begins.

At the waiting point before restarting, you can pause the program to verify the converted ADC values. This section describes 3 different ways to verify the sample data while in the Debug View of e² studio:

1. Observe the raw converted values in e² studio.
2. Plot the converted values in e² studio.
3. Export the converted values from e² studio for another software to process.

Follow the steps in the next subsections at the point in the project when ADC conversion finishes and before the program restarts. At this point J-Link RTT Viewer will say "ADC conversion completed. If desired, pause the program now to view and/or export conversion values. Once you are ready to restart program, enter '2' on the keyboard."

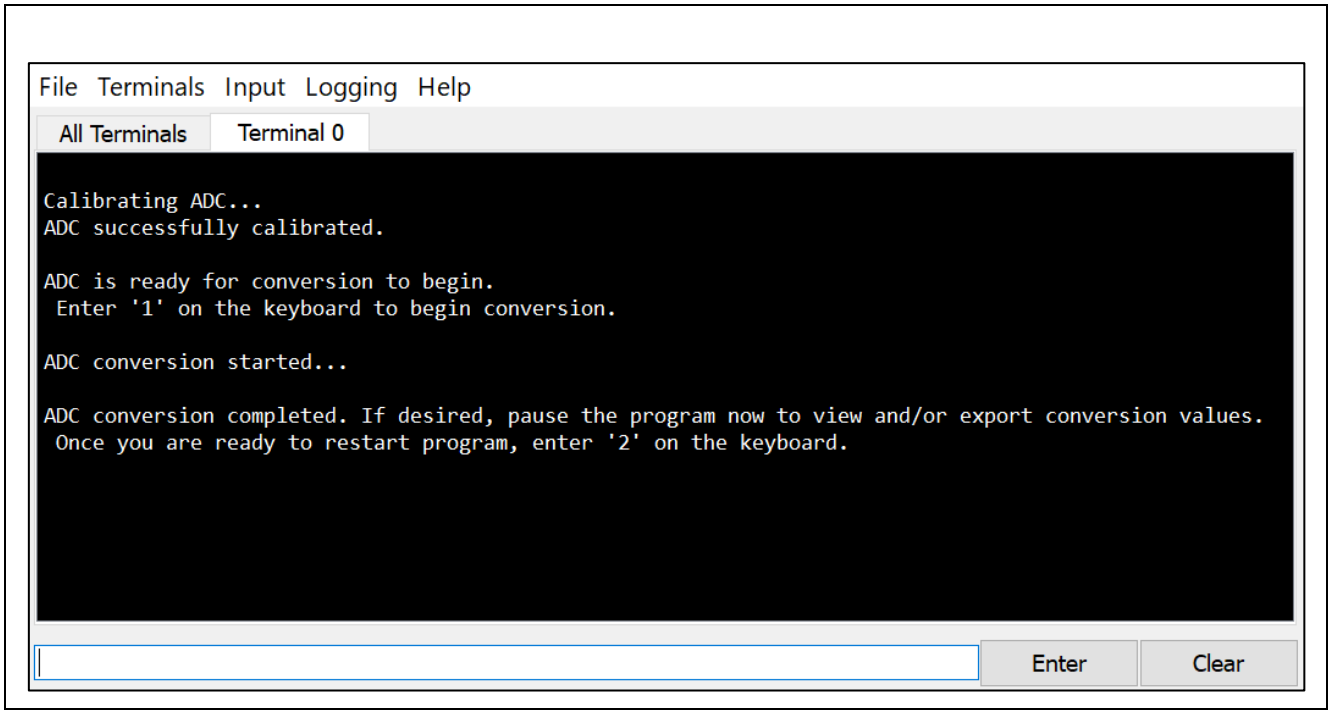


Figure 25. J-Link Prompt When the Verification can Begin

To generate the images for this section, the primary project was run with a signal generator as the analog input with the following settings:

- Waveform: Sine wave
- Frequency: 500 Hz
- Amplitude: 3.2 Vpp
- DC offset: 1.6 V
- Output load: HighZ

5.1 View Raw Values

While in the debug perspective, press the pause button to pause the debug session in e² studio.

Select the option **Window > Show View > Expressions** to open the **Expressions** tab.

Click the button to add a new expression.

If running the primary project "adc_16bit_oversampling_ra6t2" type **'adc16_vals'** in the expression field and click **OK.** If running the secondary project "adc_12bit_sar_ra6t2" type **'adc12_vals'** in the expression field and click **OK.**

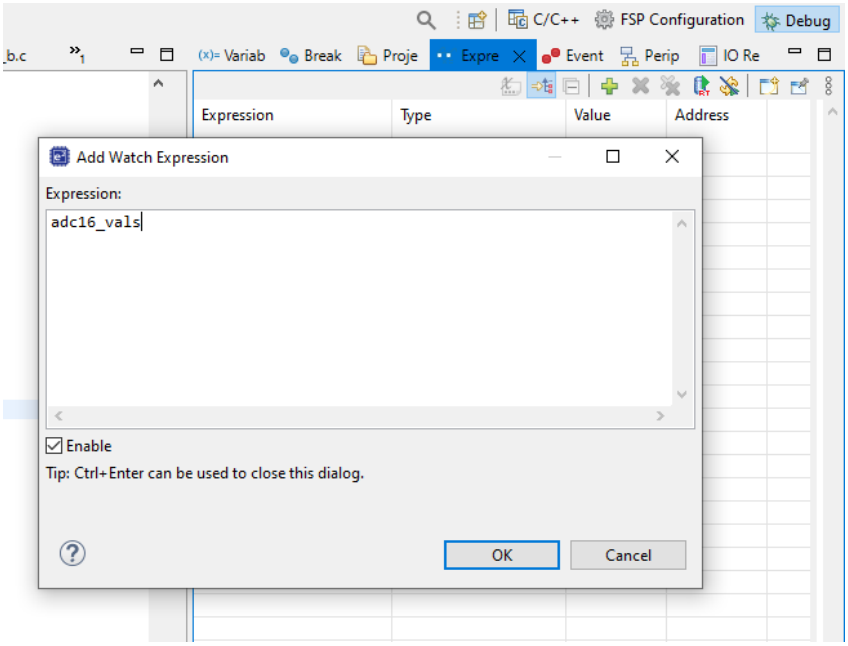
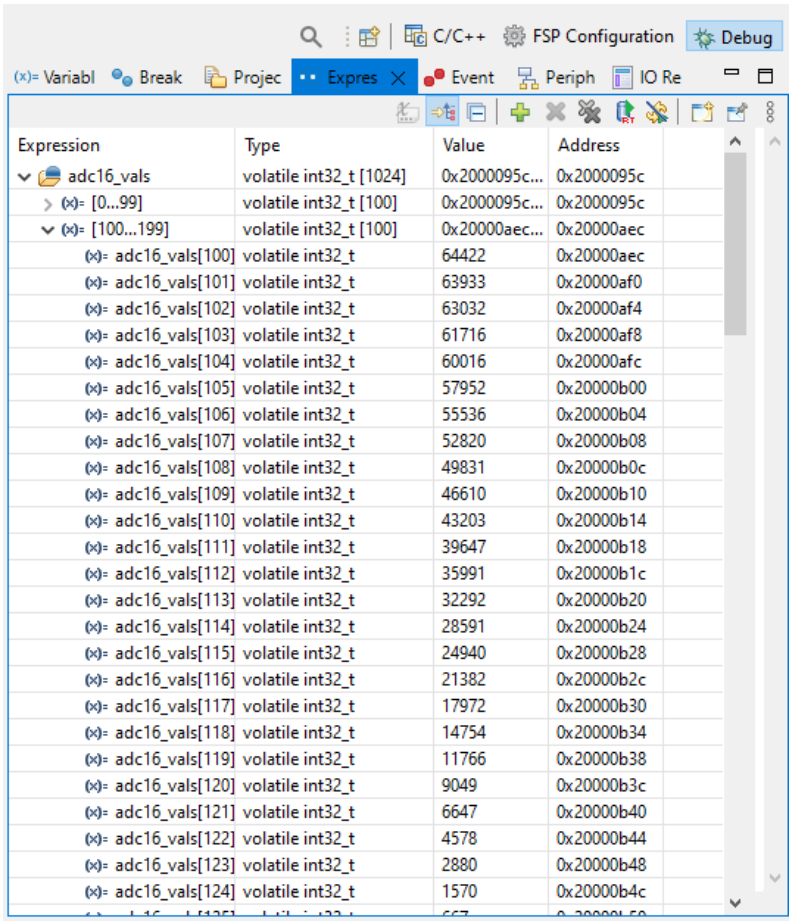


Figure 26. Adding adc16_vals as a Watch Expression

The ADC data array will appear in the Expressions tab. Click the arrows to the left of the array name to expand the data inside the array. The data is arranged in sections of 100 entries for each drop down section.



Expression	Type	Value	Address
adc16_vals	volatile int32_t [1024]	0x2000095c...	0x2000095c
[0...99]	volatile int32_t [100]	0x2000095c...	0x2000095c
[100...199]	volatile int32_t [100]	0x20000aec...	0x20000aec
adc16_vals[100]	volatile int32_t	64422	0x20000aec
adc16_vals[101]	volatile int32_t	63933	0x20000af0
adc16_vals[102]	volatile int32_t	63032	0x20000af4
adc16_vals[103]	volatile int32_t	61716	0x20000af8
adc16_vals[104]	volatile int32_t	60016	0x20000afc
adc16_vals[105]	volatile int32_t	57952	0x20000b00
adc16_vals[106]	volatile int32_t	55536	0x20000b04
adc16_vals[107]	volatile int32_t	52820	0x20000b08
adc16_vals[108]	volatile int32_t	49831	0x20000b0c
adc16_vals[109]	volatile int32_t	46610	0x20000b10
adc16_vals[110]	volatile int32_t	43203	0x20000b14
adc16_vals[111]	volatile int32_t	39647	0x20000b18
adc16_vals[112]	volatile int32_t	35991	0x20000b1c
adc16_vals[113]	volatile int32_t	32292	0x20000b20
adc16_vals[114]	volatile int32_t	28591	0x20000b24
adc16_vals[115]	volatile int32_t	24940	0x20000b28
adc16_vals[116]	volatile int32_t	21382	0x20000b2c
adc16_vals[117]	volatile int32_t	17972	0x20000b30
adc16_vals[118]	volatile int32_t	14754	0x20000b34
adc16_vals[119]	volatile int32_t	11766	0x20000b38
adc16_vals[120]	volatile int32_t	9049	0x20000b3c
adc16_vals[121]	volatile int32_t	6647	0x20000b40
adc16_vals[122]	volatile int32_t	4578	0x20000b44
adc16_vals[123]	volatile int32_t	2880	0x20000b48
adc16_vals[124]	volatile int32_t	1570	0x20000b4c

Figure 27. Viewing adc16_vals in the Expressions Tab

After viewing the data and once you are ready to continue the project, click the **Resume** button .

5.2 Plot Values in e² studio

Press the pause button to pause the debug session in e² studio.

Select the option **Window > Show View > Memory** to open a new Memory tab.

Click the button to add a new address or expression to monitor.

If running the primary project “adc_16bit_oversampling_ra6t2” type **‘&adc16_vals’** in the address/expression field and click **OK.** If running the secondary project “adc_12bit_sar_ra6t2” type **‘&adc12_vals’** in the address/expression field and click **OK.**

This will automatically open a hex integer view of memory:

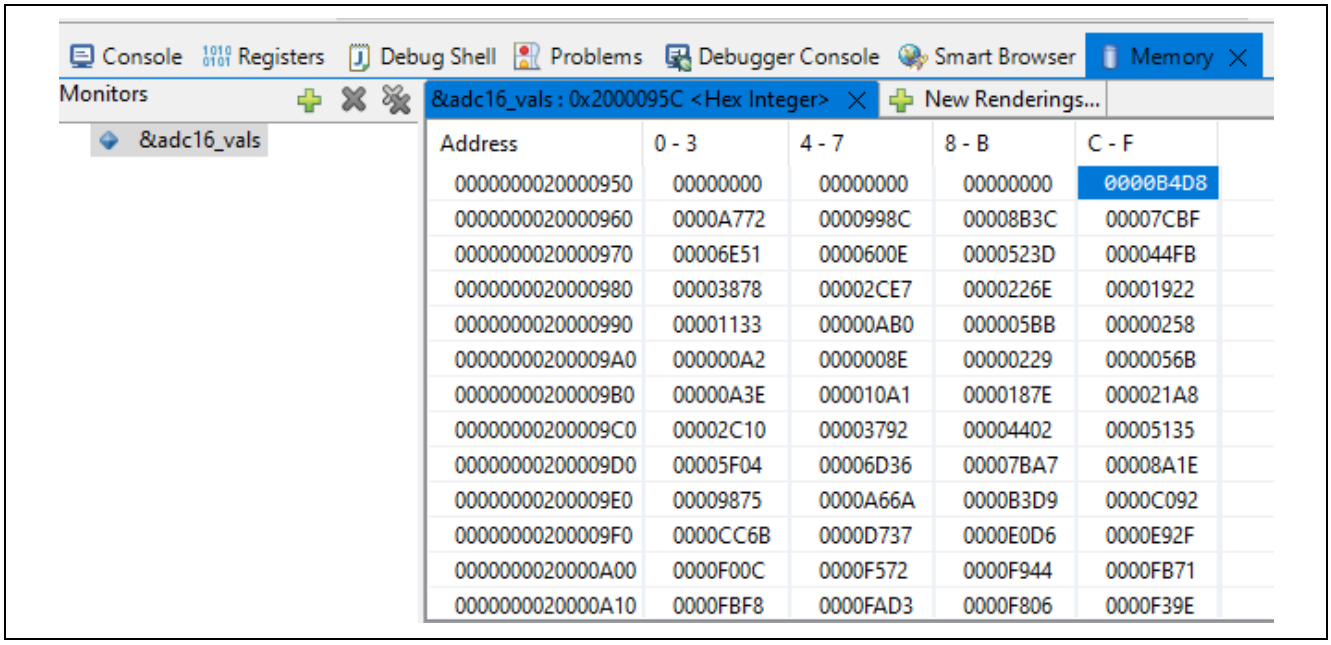


Figure 28. Memory View of adc16_vals Array

Next step is to switch the view to a waveform. Click on **New Renderings** and highlight the **Waveform** option. Press **Enter** or the **Add Rendering(s)** button. Click the **Waveform Properties** button to open the render settings.

In the menu, change the data size with the drop-down option to **32bit.** In the Y-axis settings, check the **User Specified** option. Set the **Minimum Value** to **0**.

If running the primary project “adc_16bit_oversampling_ra6t2” set the **Maximum value** to **65535**, corresponding to 2^{16} -1 which is the maximum possible reading for the 16-bit ADC data.

If running the secondary project “adc_12bit_sar_ra6t2” set the **Maximum value** to **4095**, corresponding to 2^{12}-1 which is the maximum possible reading for the 12-bit ADC data.

Set the **Buffer Size** to **4096** bytes, corresponding to the length of the array given by the value NUM_SAMPLES*4 bytes.

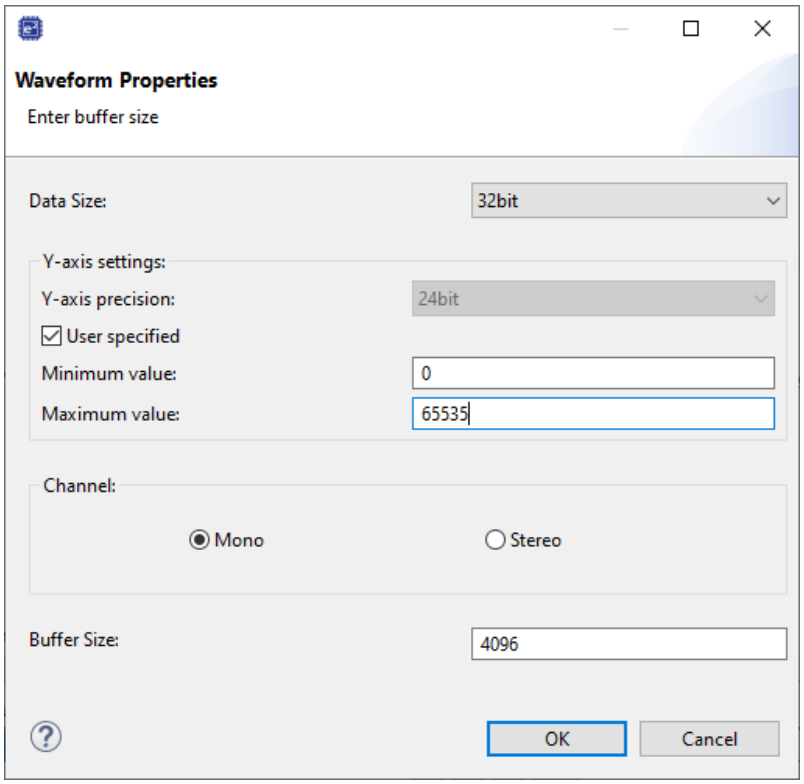


Figure 29. Proper Waveform Settings to View adc16_vals Array

Click **OK** to generate the waveform.

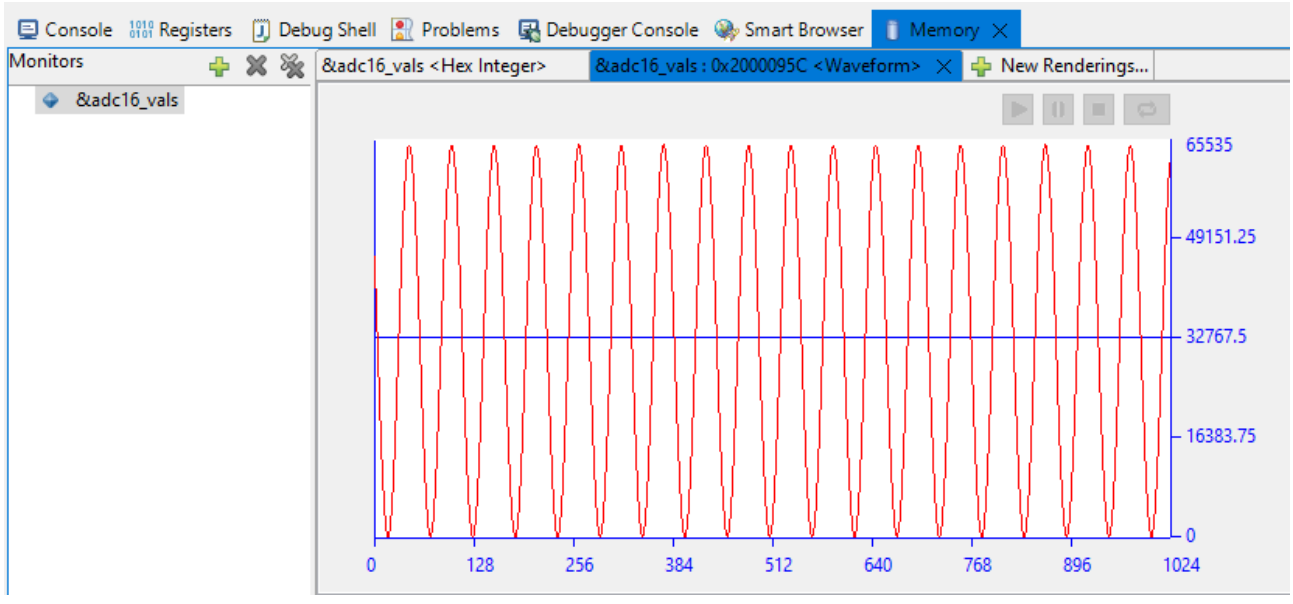


Figure 30. Waveform Plot of adc16_vals with a 500 Hz Input

In the waveform plot the x-axis corresponds to the array index value and the y-axis corresponds to the ADC sample value.

After viewing the data and once you are ready to continue the project, click the **Resume** button .

5.3 Export Values for Further Analysis in External Software

Press the pause button to pause the debug session in e² studio.

Select the option **Window > Show View > Memory** to open a new Memory tab.

Click the button to add a new address or expression to monitor.

If running the primary project “adc_16bit_oversampling_ra6t2” type **‘&adc16_vals’** in the address/expression field and click **OK.** If running the secondary project “adc_12bit_sar_ra6t2” type **‘&adc12_vals’** in the address/expression field and click **OK.**

This will automatically open a hex integer view of memory:

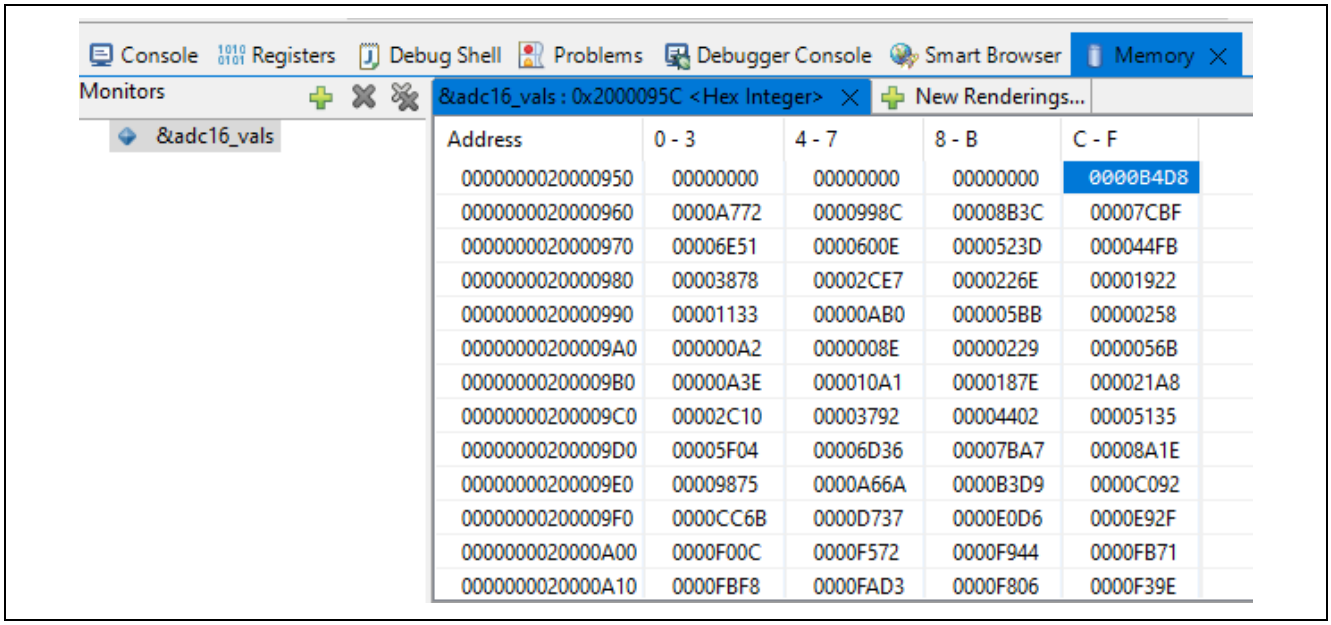


Figure 31. Hex Integer View of adc16_vals in Memory

Click the **Export Memory** button located in the Memory View toolbar.

Figure 32. Export Memory Button

Choose the export format you would like from the drop down and enter the appropriate corresponding data into the available fields. For example, to export the memory to a raw binary file: set **Format** to **RAW Binary**, set **Length** to **4096** (NUM_SAMPLES * 4 bytes), and choose your desired **File Name** and location.

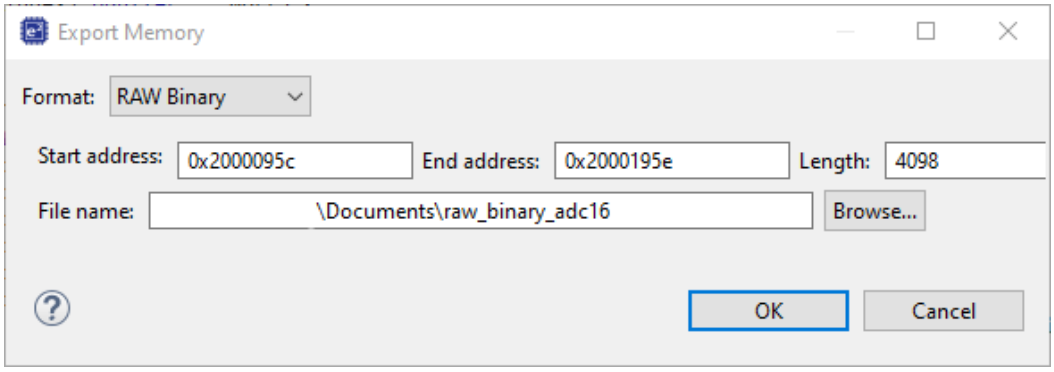


Figure 33. Example of Exporting adc16_vals Array as RAW Binary File

Click **OK** to export the memory data as the format of your choice and save it to your workstation.

Once ready to continue, click the **Resume** button **.**

6. Performance Analysis

This section provides a brief performance analysis comparison between 12-bit SAR and 16-bit Oversampling conversion methods with a couple of applied examples. You are encouraged to use the application projects to perform your own system comparisons.

In the example project comparison section, the results of conversion in the primary project and the secondary project are compared when using the same analog input signal. In the voltage conversion comparison section, conversion calculations demonstrate the difference in resolution precision.

6.1 Example Project Comparison

Both example projects were run with a signal generator as the analog input with the following settings:

- Waveform: Sine wave
- Frequency: 250 Hz
- Amplitude: 3.2 Vpp
- DC offset: 1.6 V
- Output load: HighZ

Below are waveform plots generated by following steps in section 5.2, Plot Values in e^2 studio. The first photo is the waveform captured running the primary project with the ADC configured for 16-bit oversampling conversion. The second photo is the waveform captured running the secondary project with the ADC sampled for 12-bit SAR conversion.

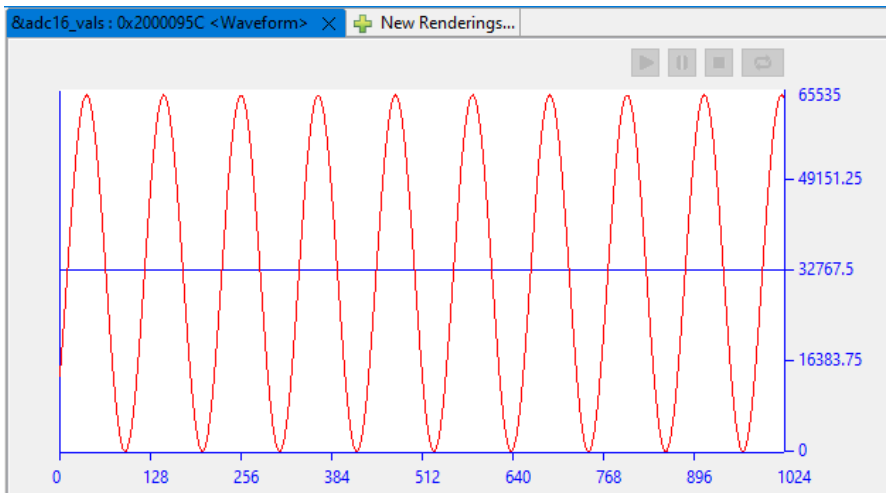


Figure 34. Plot From Running adc_16bit_oversampling_ra6t2 with 250 Hz Input Signal

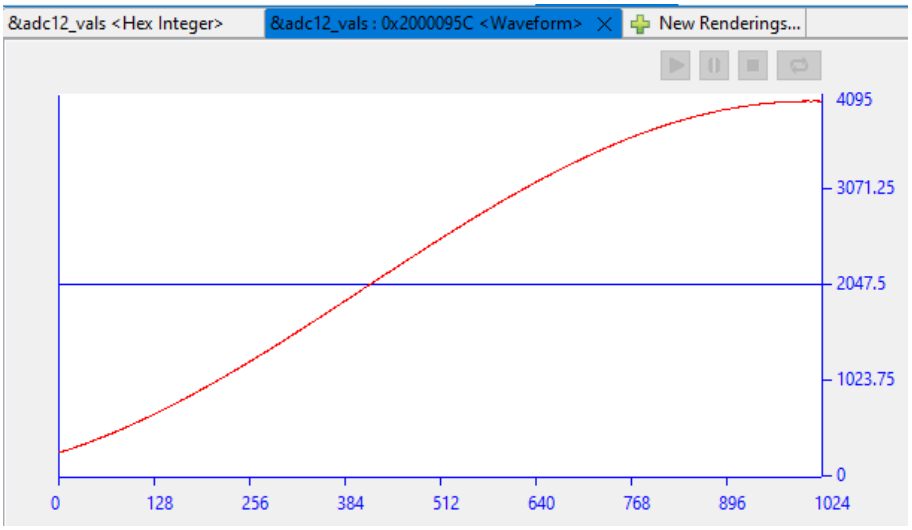


Figure 35. Plot From Running adc_12bit_sar_ra6t2 with 250 Hz Input Signal

Key Observations:

1. The range of values the ADC converts from the input signal voltage change for the 16-bit waveform is much greater than the 12-bit waveform. The difference in ADC data ranges gives a visual observation of the effect of increasing resolution with 4 extra bits of information. See section 6.2, Voltage Conversion Precision Comparison for a continued analysis of this effect.
2. The number of 250 Hz sinusoidal cycles captured is greater for the 16-bit project than the 12-bit project. Since the projects were identical except for the conversion method, this difference indicates that the conversion time of the oversampling conversion method is longer than the conversion time of the SAR conversion method. With these specific project settings, it was about 18x longer for 16-bit mode conversion. This is expected behavior intrinsic to the theory of ADC oversampling. With oversampling, multiple ADC samples need to be converted and then filtered to convert one ADC data value. The oversampling and processing require extra clock cycles. The number of clock cycles is dependent on the specific settings for your project and can be roughly calculated using the equations described in section 2.3.1.

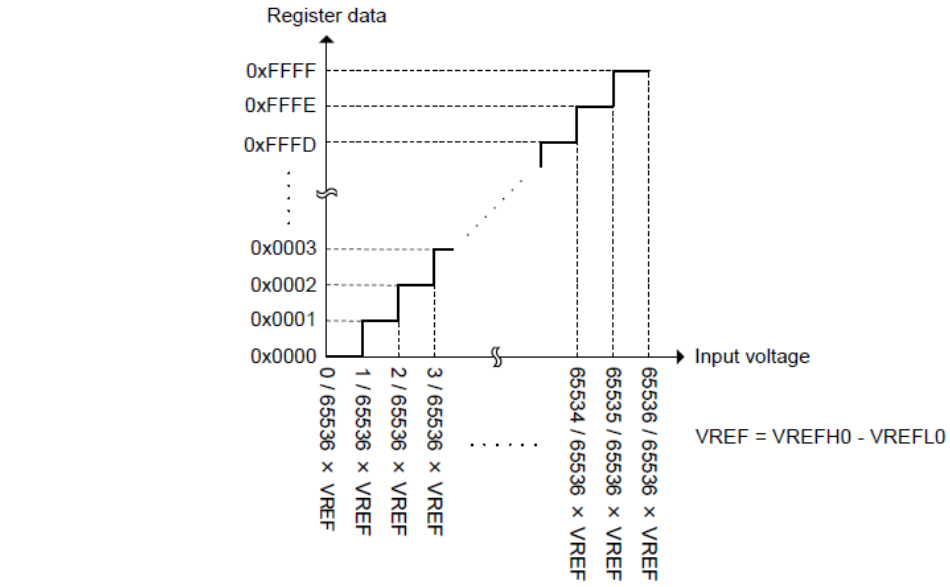
6.2 Voltage Conversion Precision Comparison

6.2.1 16-bit Voltage Conversion

(1) Single-Ended Input

With single-ended 16-bit conversion, the sampled data is in the 16-bit length unsigned data format, with a range of 0x0000 (VREFL0) to 0xFFFF (VREFH0).

The data range of A/D conversion is shown in the following image:



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 36. Data Range of A/D Conversion Result (16-bit, unsigned, single-ended input)

In single-ended input mode for Oversampling and Hybrid conversion methods, one LSB is calculated by:

$1LSB = (VREFH0 – VREFL0) / 2^{15}$

(2) Differential Input

With differential 16-bit conversion, the sampled data is in the 16-bit length signed data format, with a range of 0x8000 (-VREFH0) to 0x7FFF (+VREFH0).

The data range of A/D conversion is shown in the following image:

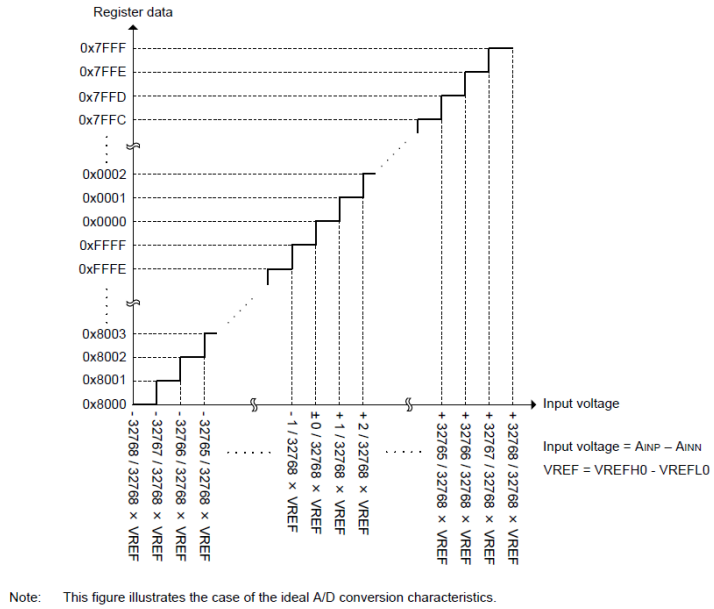


Figure 37. Data Range of A/D Conversion Result (16-bit, signed, differential input)

In differential input mode for Oversampling and Hybrid conversion methods, one LSB is calculated by:

$1LSB = [\ 2 \times (\ VREFH0 - VREFL0\ )\ ] / 2^{16}$

6.2.2 12-bit Voltage Conversion

(1) Single-Ended Input

With single-ended 12-bit conversion, the sampled data is in the 12-bit length unsigned data format, with a range of 0x0000 (VREFL0) to 0x0FFF (VREFH0).

The data range of A/D conversion is shown in the following image:

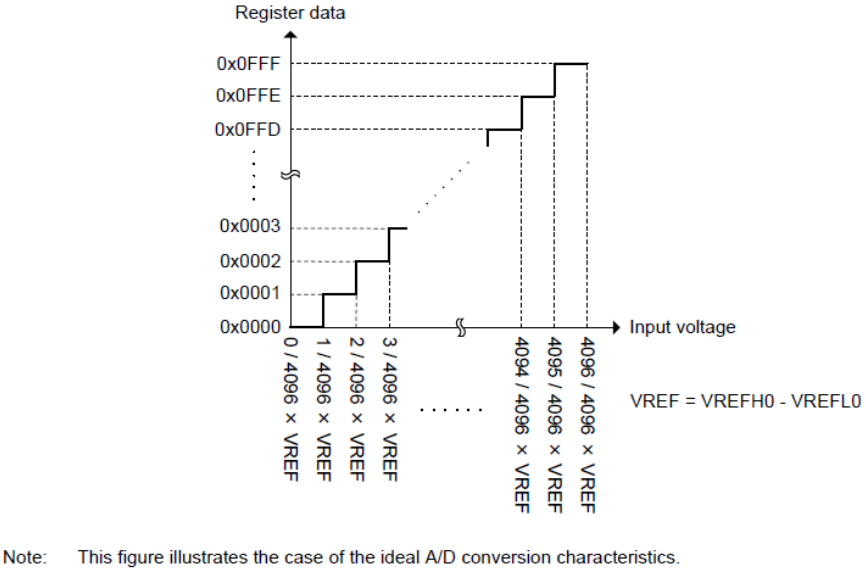


Figure 38. Data Range of A/D Conversion Result (12-bit, unsigned, single-ended input)

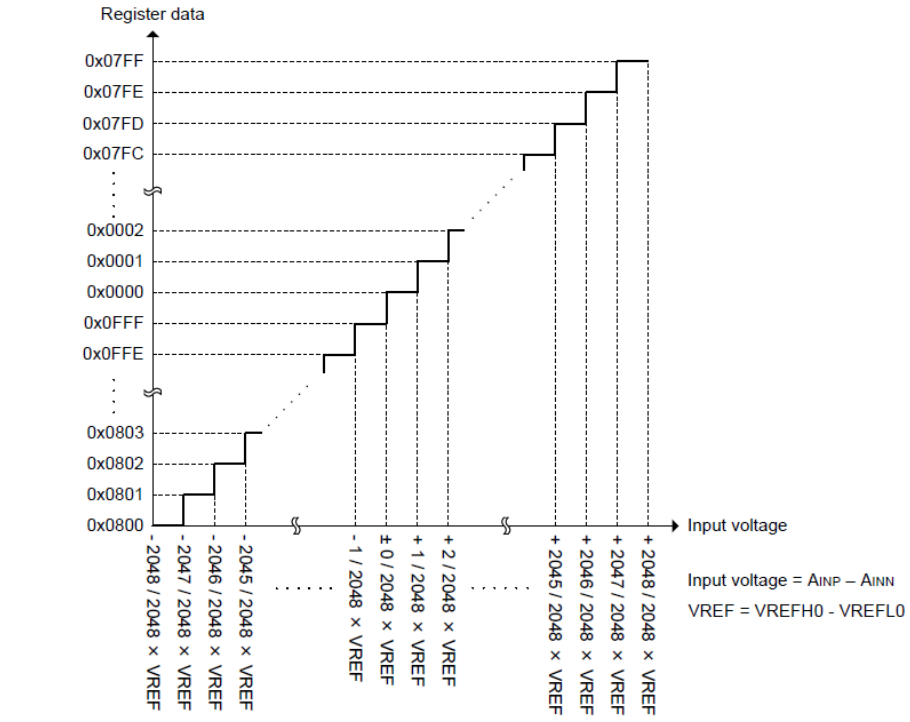
In single-ended input mode for the SAR conversion method, one LSB is calculated by:

$1LSB = (\ VREFH0 - VREFL0\ ) / 2^{11}$

(2) Differential Input

With differential 16-bit conversion, the sampled data is in the 16-bit length signed data format, with a range of 0x8000 (-VREFH0) to 0x7FFF (+VREFH0).

The data range of A/D conversion is shown in the following image:



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 39. Data Range of A/D Conversion Result (12-bit, signed, differential input)

In differential input mode for the SAR conversion method, one LSB is calculated by:

$1LSB = [2 \times (VREFH0 – VREFL0)] / 2^{12}$

6.2.3 Example LSB Calculation

For a sample calculation to demonstrate the rounding when the A/D converter samples a voltage, assume the following:

- VREFH0 = 3.20 V
- VREFL0 = 0.00 V
- Single-ended input

For 16-bit data conversion, one LSB is given by

$1 LSB = (VREFH0 – VREFL0) / 2^{15} = (3.20 V / 2^{15}) = 0.0977 mV$

And for 12-bit data conversion, one LSB is given by

$1 LSB = (VREFH0 – VREFL0) / 2^{11} = (3.20 V / 2^{11}) = 1.56 mV$

The LSB is a measurement of how much voltage change will occur before the converted digital value changes (either increases or decreases) by 1 bit value. Based on LSB calculation it is clear that the 16-bit conversion provides a more precise measurement than the 12-bit conversion.

7. References

- Renesas FSP User's Manual renesas.github.io/fsp
- Renesas RA MCU datasheet Select the relevant MCUs from www.renesas.com/ra
- Example Projects github.com/renesas/ra-fsp-examples
- An Overview of Noise-Shaping SAR ADC: From Fundamentals to the Frontier: https://ieeexplore.ieee.org/document/9569768

Website and Support

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

RA Product Information	www.renesas.com/ra
RA Product Support Forum	www.renesas.com/ra/forum
RA Flexible Software Package	www.renesas.com/FSP
Renesas Support	www.renesas.com/support

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Mar.23.23	—	First release document
1.10	Feb.28.24	—	Update to FSP v5.0.0
1.20	Oct.16.24	—	Update to FSP v5.5.0

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

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

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

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 be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. 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.
6. 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.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. 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.
9. 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.
10. 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.
11. 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.
12. 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.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.