RH850/U2A-EVA Group

A/D Conversion Operation Example using ADCJ Module

Summary

This application notes explains the operation example of RH850/U2A series by using the 12 bits A/D convertor of successive approximation method (hereinafter called ADCJ).

Aim of this document and software is to provide supplemental information for the function on RH850/U2A. It is not intended to implement in the design for mass production.

There is no guarantee to update in this document and software to reflect the latest manual, errata, technical update and development environment. You are fully responsible for the incorporation or any other use of the information of this document in the design of your product or system, and please refer to latest manual, errata, technical update and development environment.

Target Device

• RH850/U2A-EVA Group

Target Integrated Development Environment

CS+(from RENESAS Electronics)

Version	:V8.07.00
Device File	:DR7F702300.DVF
	:DR7F702301.DVF
	:DR7F702302.DVF

Reference Document

RH850/U2A-EVA User's Manual: Hardware

For function details and electrical characteristics, please refer to "User's Manual: Hardware".

This application note is based on the following manual.

- RH850/U2A-EVA User's Manual (Rev.1.20): R01UH0864EJ0120

Operation Confirmation Condition

Operation Frequency	External Oscillator: 40MHz CPU clock: 400MHz CLK_HSB: 80MHz CLK_LSB: 40MHz
Operation Voltage	AnVREF: 5.0V AnVCC: 5.0V AnVSS: GND

Contents

1. Introduction

This application note explains the ADCJ usage of the RH850/U2A and the software operation example.

1.1 Use Function

The RH850/U2A hardware functions used in this application note are shown below.

- Successive approximation method 12 Bits A/D Convertor（ADCJ）*1
- General purpose input/output pin（PORT）
- DMA(sDMAC)
- OS Timer（OSTM）*1
- Motor Control Timer（TSG3）*1

*1 The module standby register setting (the setting of the clock supply to the use function) is necessary to perform for enabling ADCJ, OSTM, and TSG3.

2. ADCJ Overview

2.1 ADCJ Operation

2.1.1 Virtual Channel and Scan Group

RH850/U2A mounts the 3 ADCJ modules (ADCJ0, ADCJ1, ADCJ2). The each ADCJ modules has the 64 virtual channels, and Analog channel for which A/D conversion is to be made and accompanying information like a conversion mode, interrupt are configured for each virtual channel. In addition, ADCJ has 5 scan groups, and each scan group can independently set the scan contents such as scan mode and input trigger. It is possible to execute the scan that A/D converts any analog channel in any order to execute the virtual channels indicated by the start virtual channel pointer and the end virtual channel pointer in order in each scan group.

The following show the allocation example of the physical channel, virtual channel, and scan group.

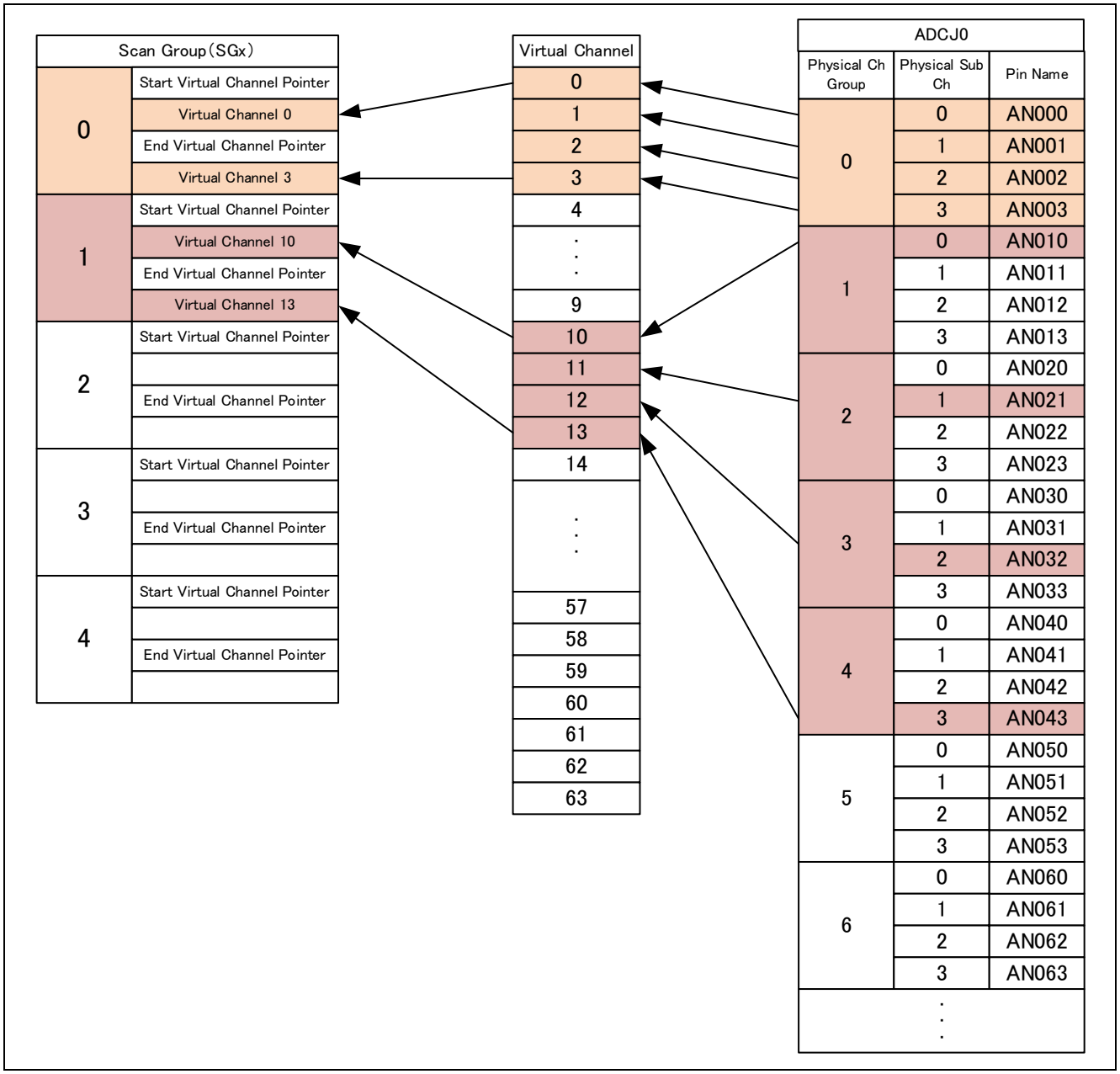

Figure 2-1 Virtual Channel and Scan Group Allocation (ADCJ0)

2.1.2 Scan End Interrupt Request

The Scan group (SGx) can generate the scan end interrupt request (INTADCJnlx) to the INTC. There are two sources of this scan end interrupt request: generated “when the scan of the scan group (SGx) is ended” or “when the A/D conversion of the virtual channel j allocated to the scan group (SGx) is ended”.

The scan end interrupt request enable/disable setting “when the scan of the scan group (SGx) is ended” is performed by the ADIE (0=disable, 1=enable) in ADCJnSGCRx. The scan end interrupt request enable/disable setting “when the A/D conversion of the virtual channel j allocated to the scan group (SGx) is ended” is performed by the ADIE (0=disable, 1=enable) in ADCJnVCRj. The setting of ADIE in ADCJnSGCRx and ADIE in ADCJnVCRj are irrelevant.

The following shows the scan end interrupt generation timing when execute the scans of virtual channel 0 and virtual channel 1 in the scan group 0 (SG0).

Ex.1) Output INTADCJnI0 at the A/D conversion end of virtual channel 0

ADCJnSGCR0.ADIE = 0 (Not output INTADCJnI0 at the SG0 scan end.)

ADCJnVCR0.ADIE = 1 (Output INTADCJnI0 at the end of virtual channel 0 end on SG0.)

ADCJnVCR1.ADIE = 0 (Not output INTADCJnI0 at the end of virtual channel 1 end on SG0.)

Ex.2) Output INTADCJnI0 at the A/D conversion end of virtual channel 0 and virtual channel 1

ADCJnSGCR0.ADIE = 0 (Not output INTADCJnI0 at the SG0 scan end.)

ADCJnVCR0.ADIE = 1 (Output INTADCJnI0 at the end of virtual channel 0 end on SG0.)

ADCJnVCR1.ADIE = 1 (Output INTADCJnI0 at the end of virtual channel 1 end on SG0.)

Ex.3) Output INTADCJnI0 at the SG0 scan end

ADCJnSGCR0.ADIE = 1 (Not output INTADCJnI0 at the SG0 scan end.)

ADCJnVCR0.ADIE = 0 (Not output INTADCJnI0 at the end of virtual channel 0 end on SG0.)

ADCJnVCR1.ADIE = 0 (Not output INTADCJnI0 at the end of virtual channel 1 end on SG0.)

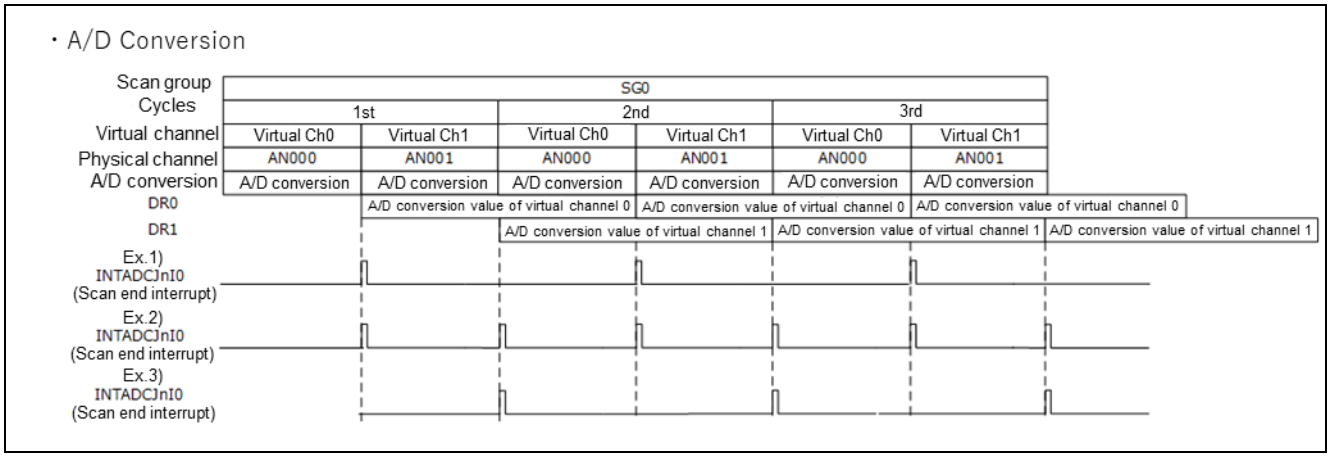

Figure 2-2 Scan End Interrupt Generation Timing

3. Operation Overview

3.1 Sequential Scan Conversion of Arbitrary Channel

3.1.1 Specification Overview

This section explains the normal A/D conversion method not using T/H (Track & Hold).

Allocates the 2 virtual channels (AN000, AN001) to the scan group 0 (SG0), and performs 1 scan in Multicycle scan Mode. Stores the AN000 and AN001 conversion values to variable in the scan group end and ends operation.

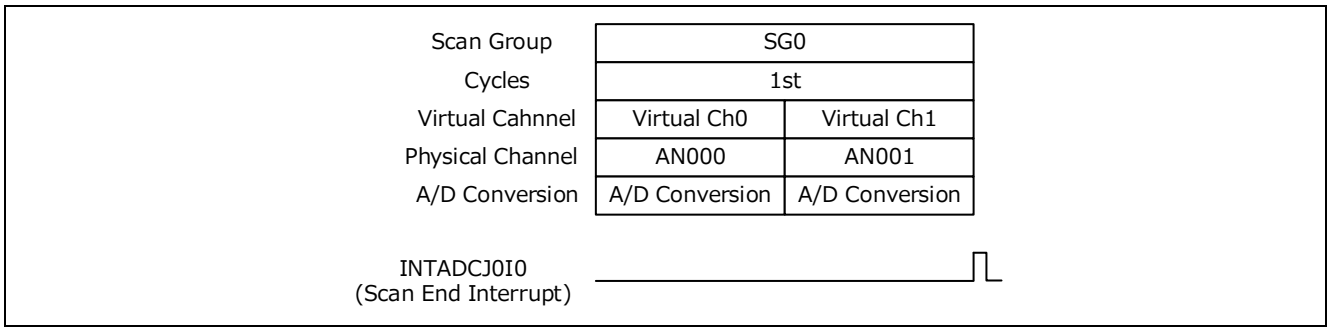

Figure 3.1.1 Normal A/D Conversion (Multicycle scan Operation)

3.1.2 Use Function

The used hardware function in this operation example is shown below.

・A/D convertor（ADCJ0）

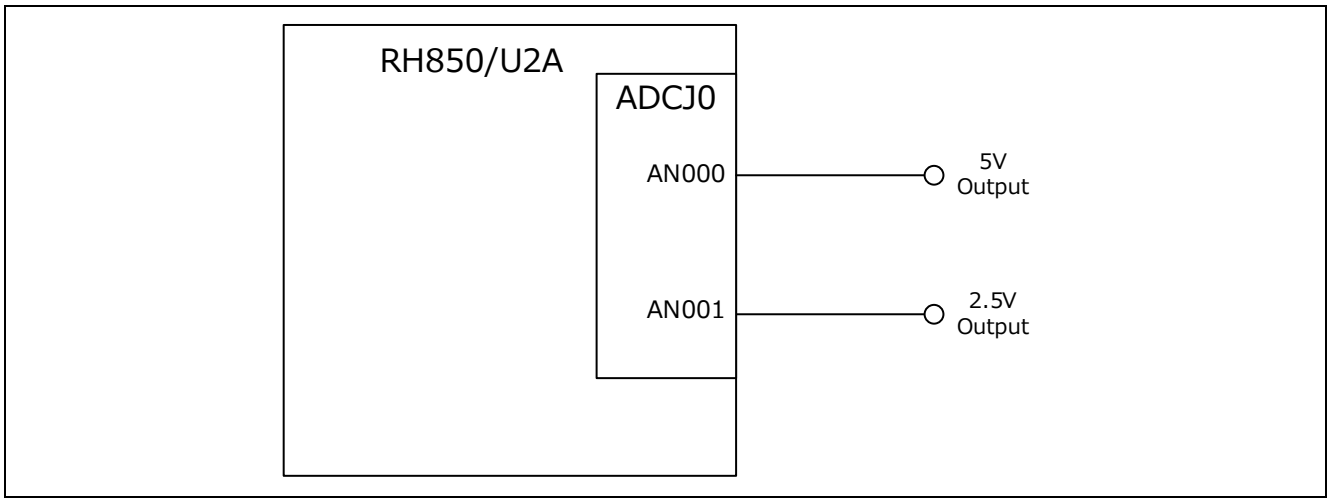

Figure 3.1.2 System Configuration

3.1.3 Explanation of Operation Example

In this operation example, perform the normal A/D conversion by the 1 scan of the multicycle scan mode that uses the AN000 and AN001 of the ADCJ0 module.

Allocate the virtual channel 0 (AN000) and virtual channel 1 (AN001) to the scan group 0 (SG0).

The analog signal inputs 5.0V to AN000 and 2.5V to AN001.

Start by the software trigger SGST, and AN001 is A/D converted after AN000 is A/D converted.

Enable the scan end interrupt INTADCJ0I0 (when scan group ends), store each A/D conversion result to the variable in the interrupt processing, and end the operation.

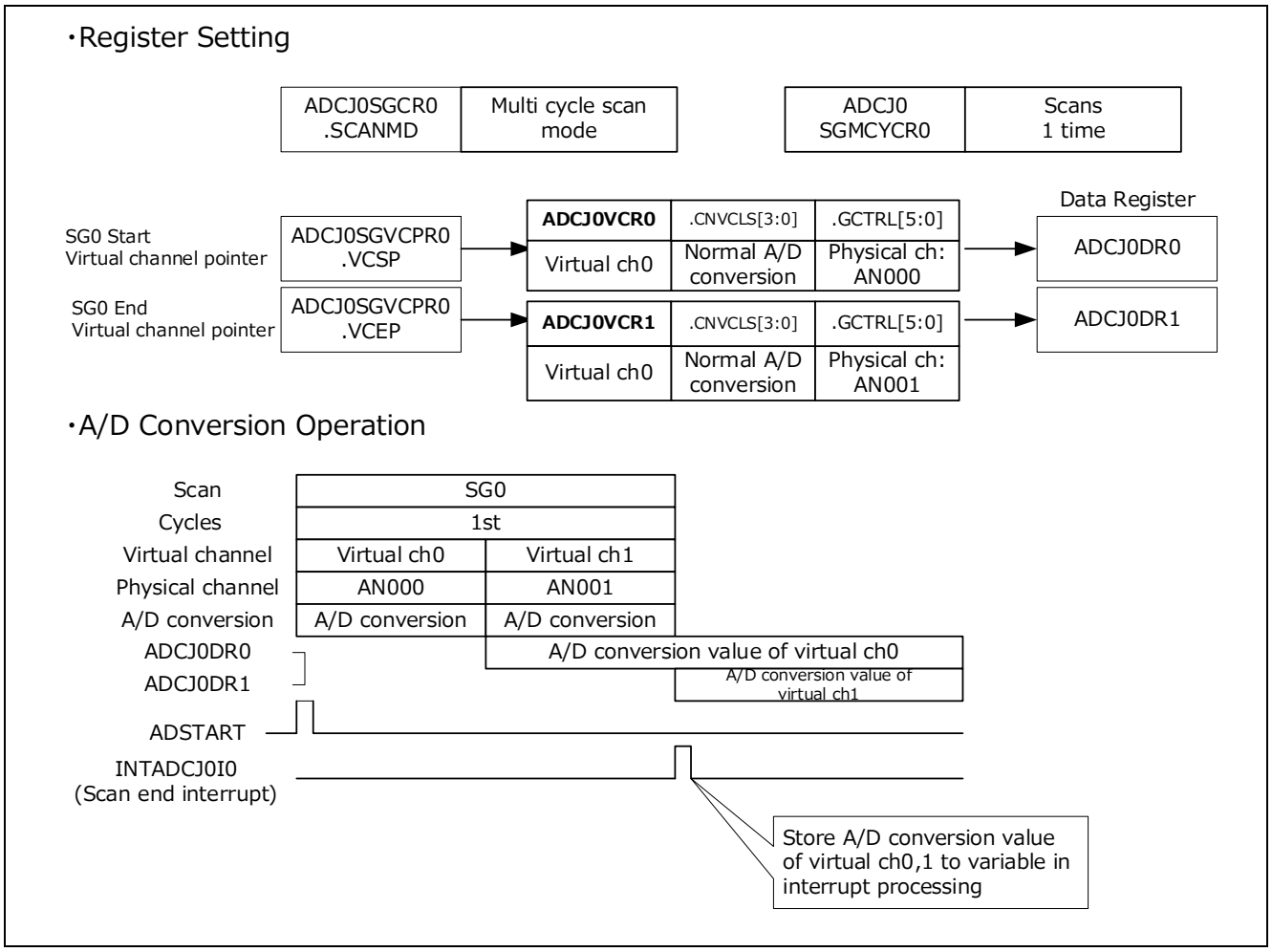

Figure 3.1.3 Example of Normal A/D Conversion (Multicycle scan Mode) Operation

3.1.4 Software Explanation

・Module Explanation

The module list in this operation example is shown below.

Table 3.1.1 Module List

Module Name	Label Name	Function
Maine routine	main_pm0	Perform various setting and application start.
ADCJ initialize routine	ADCJ_init	Perform ADCJ initialization.
ADCJ interrupt processing routine	INTADCJ0I0	Store A/D conversion result to variable in virtual scan group end interrupt
Interrupt initialize routine	intc2_init	Perform ADCJ interrupt initialization.

・Register Setting

The Register setting of each function in this operation example is shown below.

Table 3.1.2 ADCJ Register Setting

Register Name	Setting Value	Function
ADCJ0ADCR1	0x00	Synchronous suspend
ADCJ0ADCR2	0x10	Signed integer format
		Disable since CNVCL = 0
ADCJ0SFTCR	0x00	Disable variable setting
ADCJ0VCLMINTERx	0x00000000	Not use upper/lower limit check interrupt table
ADCJ0VMONVDCR1	0x00	Not use divide voltage resistance
ADCJ0VMONVDCR2	0x00	Not use divide voltage resistance
ADCJ0SMPCR	0x00000000	Not use buffer amp filter
ADCJ0TDCR	0x00	Not perform pin level self-diagnosis
ADCJ0ODCR	0x0000	Not perform disconnection detection
ADCJ0TOCCR	0x00	Not perform trigger overlap check.
ADCJ0GTMENTSGER	0x0000	Not GTM entry
ADCJ0ADENDP0	0x00	Not use A/D conversion monitor
ADCJ0THCR	0x00	Not use T&H
ADCJ0THER	0x00	Not use T&H
ADCJ0THACR	0x00	Not use T&H
ADCJ0PWDCR	0x00	Disable PWM-Diag
ADCJ0SGVCPR0	0x0100	End virtual channel: 1 Start virtual channel: 0
ADCJ0SGCR0	0x50	Enable ADSTART Multicycle scan mode Output INTADCJ0I0 when SG0 end Disable H/W trigger input to SG0
ADCJ0SGMCYCR0	0x00	Scan 1 time in multicycle scan mode
ADCJ0VCR0	0x00000000	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion Not use multiplexer Not output VCRj end interrupt Physical CH0 (AN000)

Register Name	Setting Value	Function
ADCJ0VCR1	0x00000001	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion Not use multiplexer Not output VCRj end interrupt Physical CH1（AN001）

Table 3.1.3 Interrupt Register Setting

Register Name	Setting Value	Function
EIBD227	0x00000000	Bind interrupt to PE0
EIC227	0x0040	Table reference/priority level 0

・Operation Flow

The flowchart of the operation example in this operation example is shown below.

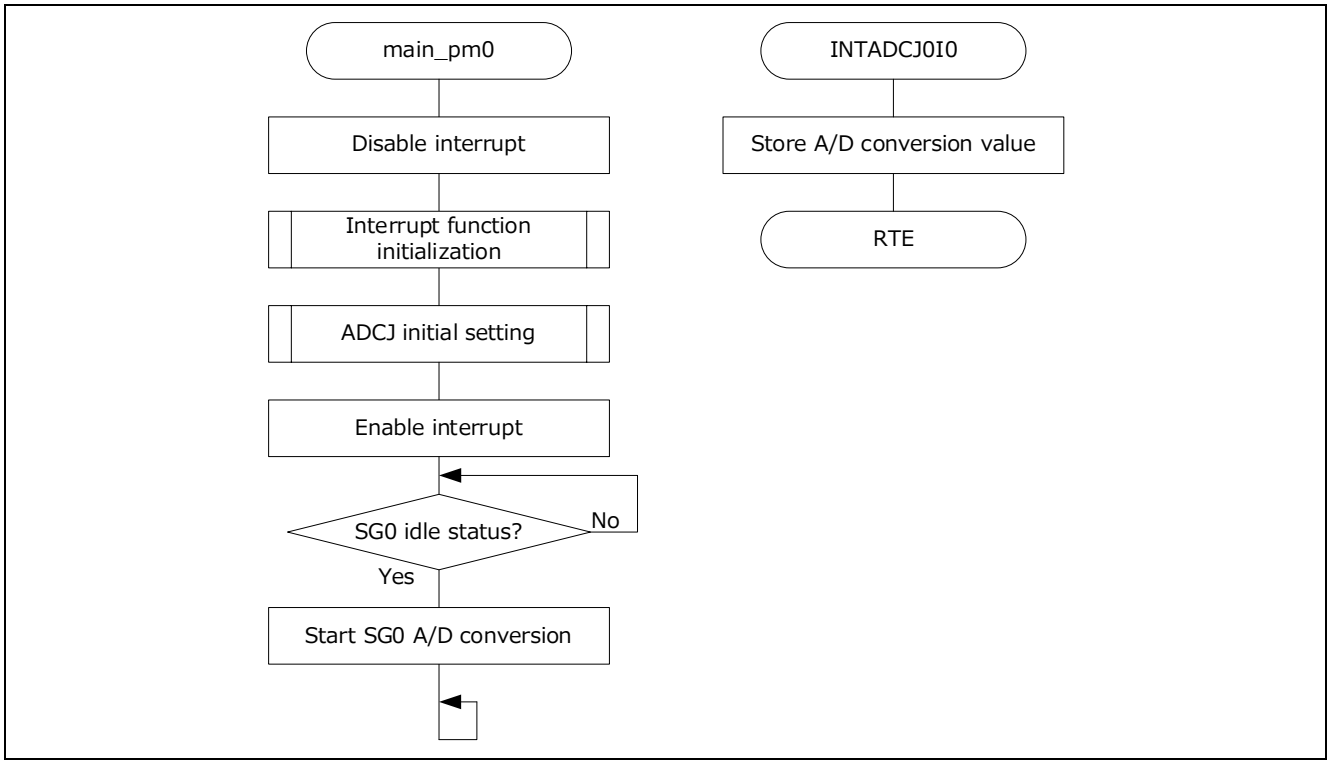

Figure 3.1.4 Flowchart

3.2 Synchronous suspend & Resume Operation

3.2.1 Specification Overview

This section explains the synchronous suspend & resume operation.

Allocate the 3 virtual channels (AN001, AN002, AN000) to the scan group 0 (SG0) and the virtual channel (AN010) to the scan group 1 (SG1). (Priority: SG0<SG1<SG2<SG3<SG4)

Start SG0 operation, and start SG1 during SG0 A/D conversion. Suspend the SG0 operation after end of the SG0 virtual channel in processing, and start the SG1 A/D conversion. After end of the SG1 conversion , restart from the suspended virtual channel of SG0.

Store the conversion value to the variable at SG0 scan group end.

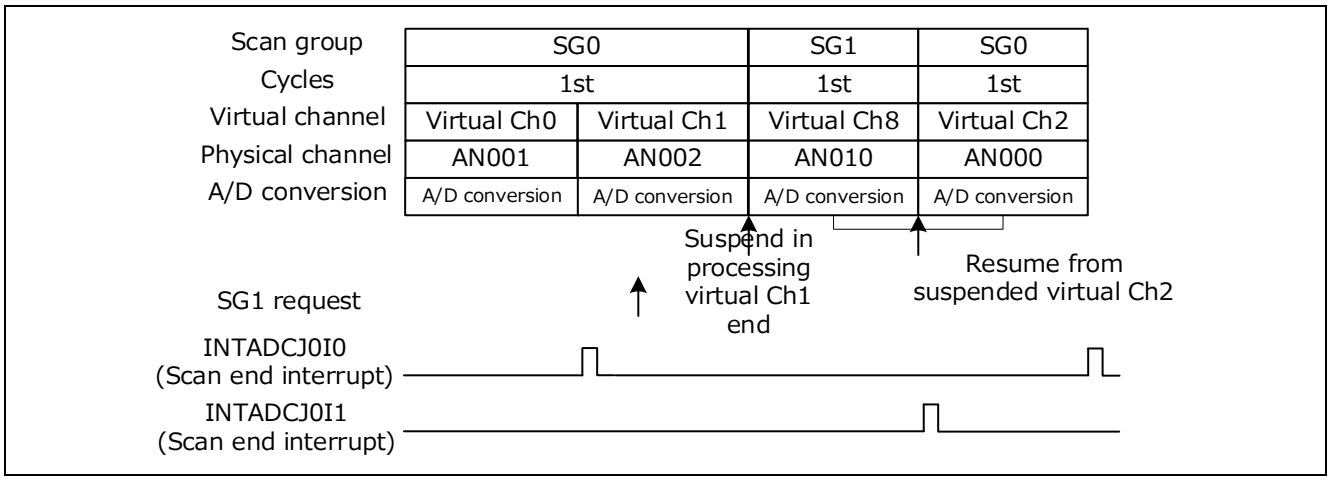

Figure 3.2.1 Synchronous suspend & Resume Operation

3.2.2 Use Function

The used hardware function in this operation example is shown below.

・A/D convertor（ADCJ0）

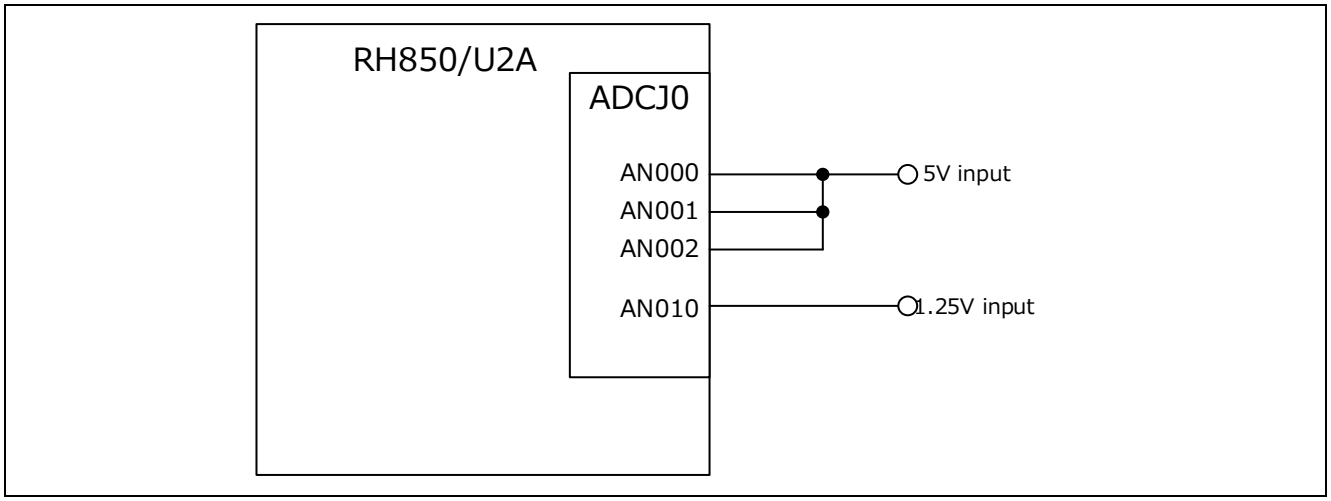

Figure 3.2.2 System Configuration

3.2.3 Explanation of Operation Example

In this operation example, perform the synchronous suspend & resume using AN000, AN001, AN002, and AN010 of ADCJ0 module.

Allocate the virtual channel 0 (AN001), virtual channel1 (AN002), and virtual channel 2 (AN000) to the scan group (SG0), and allocate the virtual channel8 (AN010) to the scan group 1 (SG1).

Input 5.0V to AN001, AN002, and AN000. Input 1.25 V to AN010.

Scan end interrupt INTADCJ0I0 of SG0 is generated at the end of virtual channel 0 conversion and the end of SG0. Scan end interrupt INTADCJ0I1 of SG1 is generated at the end of SG1.

At first, set ADCJ0SGSTCR0.SGST, and start SG0. When the virtual channel 0 processing of SG0 ends, INTADCJ0I0 is generated, ADCJ0SGSTCR1.SGST is set in the interrupt processing, and SG1 is started.

Since the high priority scan group processing is started after waiting for the converting virtual channel end in the synchronous suspend & resume, in this operation example, suspends SG0 operation and starts the virtual channel 8 A/D conversion of SG1 after the virtual channel 1 of SG0 ends.

After SG1 ends, resumes the suspended virtual channel 2 operation of SG0.

Stores each A/D conversion value to the variable by the interrupt processing of the scan end interrupt INTADCJ0I0 generated at end of SG0.

Also, in this operation example, it monitors the Port2_0 and Port2_14 output in order to confirm the synchronous suspend and resume operations. Port2_0 is for SG0 status and Port2_14 is for SG1 status.

Port2_0 output sets High in INTADCJ0I0 interrupt processing generated at end of the virtual channel 0 of SG0, and sets Low in INTADCJ0I0 interrupt processing generated at end of the SG0.

Port2_14 sets High in INTADCJ0I1 interrupt processing generated at end of the SG1

The synchronous suspend and resume can be confirmed to be performed since the Port2_14 becomes High (SG1 operation ends) during the Port2_0 High period (SG0 operation/suspension).

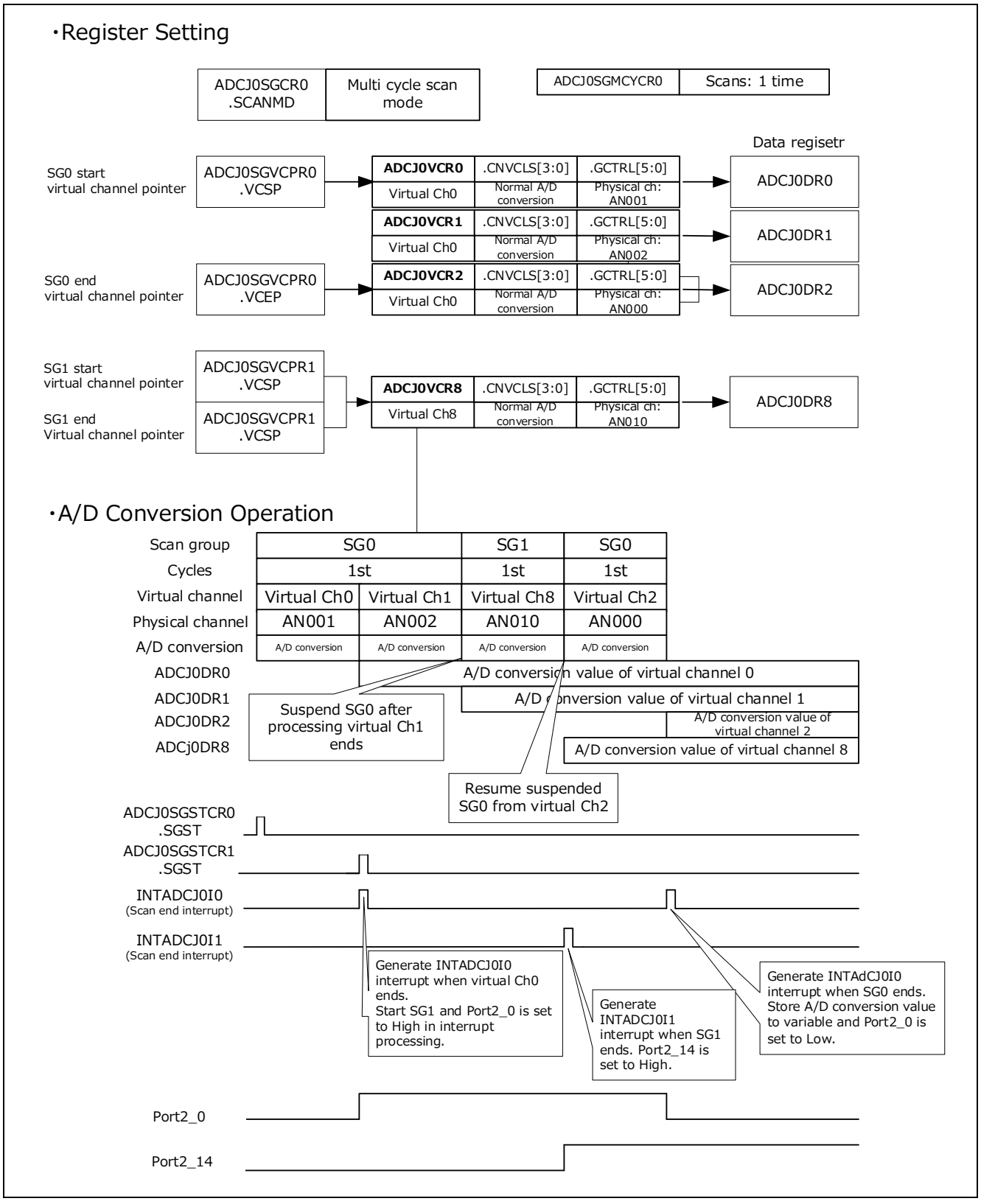

Figure 3.2.3 Operation Example of Synchronous Suspend & Resume

3.2.4 Software Explanation

・Module Explanation

The module list in this operation example is shows below.

Table 3.2.1 Module List

Module Name	Label Name	Function
Maine routine	main_pm0	Perform various setting and application start.
Port initialize routine	port_init	Perform port initialization.
ADCJ initialize routine	ADCJ_init	Perform ADCJ initialization.
ADCJ/SG0 interrupt processing routine	INTADCJ0I0	・Start SG1 and invert Port 0_0 output after virtual channel of SG0 ends. ・Store A/D conversion result to variable and invert Port 0_0 output after virtual channel of SG0 ends.
ADCJ/SG1 interrupt processing routine	INTADCJ0I1	Invert Port 0_1 output in scan group end interrupt processing.
Interrupt initialize routine	intc2_init	Perform ADCJ interrupt initialization.

・Register Setting

The Register setting of each function in this operation example is shown below.

Table 3.2.2 ADCJ Register

Register Name	Setting Value	Function
ADCJ0ADCR1	0x00	Synchronous suspend
ADCJ0ADCR2	0x10	Signed integer format
		Disable since CNVCL = 0
ADCJ0SFTCR	0x00	Disable variable setting
ADCJ0VCLMINTERx	0x00000000	Not use upper/lower limit check interrupt table
ADCJ0VMONVDCR1	0x00	Not use divide voltage resistance
ADCJ0VMONVDCR2	0x00	Not use divide voltage resistance
ADCJ0SMPCR	0x00000000	Not use buffer amp filter
ADCJ0TDCR	0x00	Not perform pin level self-diagnosis
ADCJ0ODCR	0x0000	Not perform disconnection detection
ADCJ0TOCCR	0x00	Not perform trigger overlap check.
ADCJ0ADENDP0	0x00	Not use A/D conversion monitor
ADCJ0THCR	0x00	Not use T&H
ADCJ0THER	0x00	Not use T&H
ADCJ0WAITTR0	0x0000	Unuse since WTTS[3:0]=0
ADCJ0THACR	0x00	Not use T&H
ADCJ0PWDCR	0x00	Disable PWM-Diag
ADCJ0SGVCPR0	0x0200	End virtual channel: 2 Start virtual channel: 0
ADCJ0SGCR0	0x50	Enable ADSTART Multicycle scan mode Output INTADCJ0I0 when SG0 end Disable H/W trigger input to SG0
ADCJ0SGMCYCR0	0x00	Scan 1 time in multicycle scan mode

Register Name	Setting Value	Function
ADCJ0VCR0	0x00000081	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion Not use multiplexer Not output VCRj end interrupt Physical CH0 (AN000)
ADCJ0VCR1	0x00000002	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion Not use multiplexer Not output VCRj end interrupt Physical CH2 (AN002)
ADCJ0VCR2	0x00000000	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion Not use multiplexer Not output VCRj end interrupt Physical CH0 (AN000)
ADCJ0SGVCPR1	0x0808	End virtual channel :8 Start virtual channel :8
ADCJ0SGCR1	0x50	Enable ADSTART Multicycle scan mode Output INTADCJ0I0 when SG0 ends Disable HW trigger input to SG0
ADCJ0SGMCYCR1	0x00	Scan 1 time in multicycle scan mode
ADCJ0VCR8	0x00000004	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion Not use multiplexer Not output VCRj end interrupt Physical CH10 (AN010)

Table 3.2.3 Interrupt Register Setting

Register Name	Setting Value	Function
EIBD227	0x00000000	Bind interrupt to PE0
EIC227	0x0041	Table reference/priority level 1
EIBD228	0x00000000	Bind interrupt to PE0
EIC228	0x0040	Table reference/priority level 0

・Operation Flow

The flowchart of the operation example in this operation example is shown below.

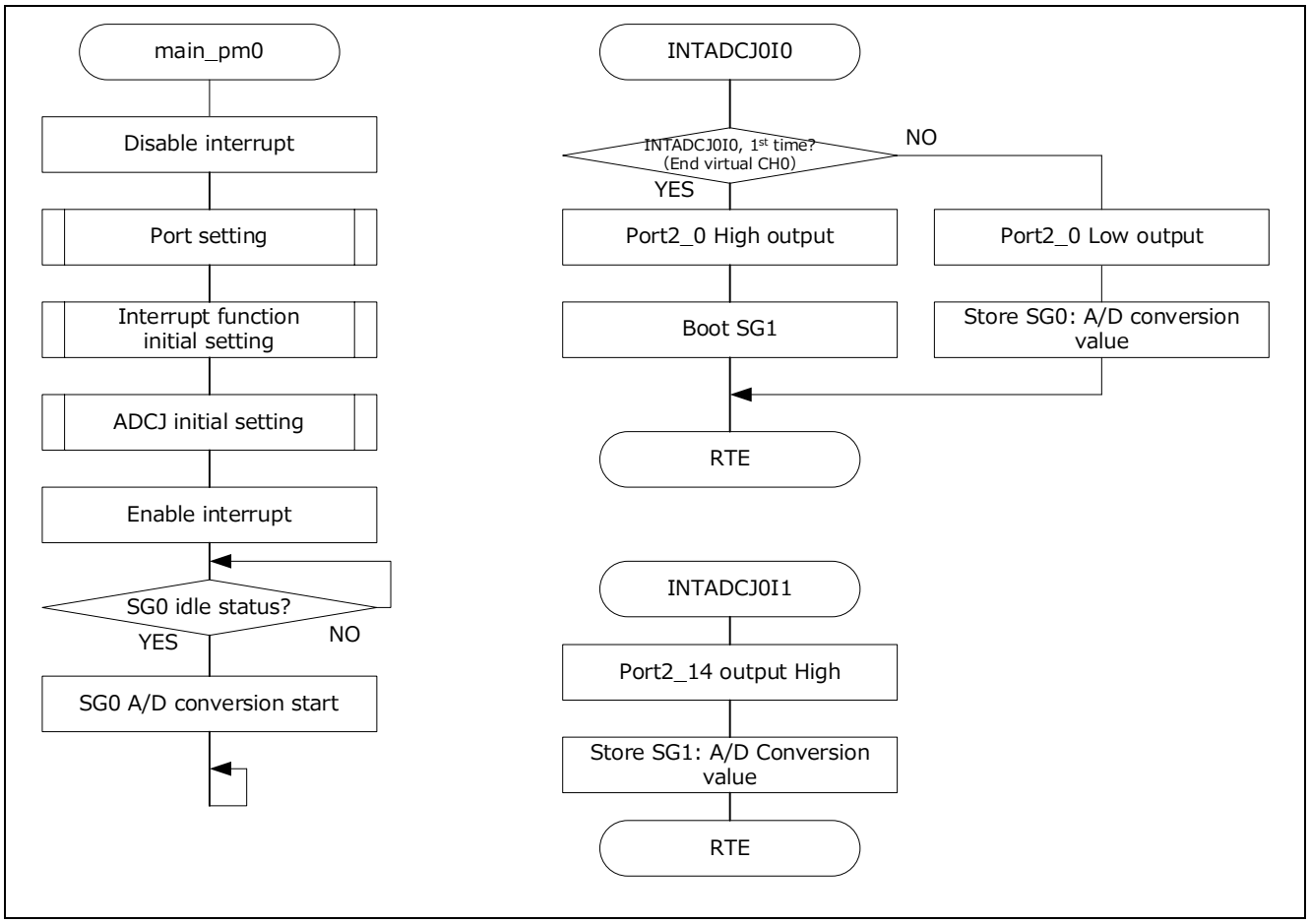

Figure 3.2.4 Flowchart

3.3 Interval Conversion using A/D timer

3.3.1 Specification Overview

This section explains the A/D conversion performing method in arbitrary period to use A/D timer.

Allocate 1 virtual channel (AN000) to scan group 3 (SG3), and set A/D timer as the trigger of A/D conversion start. A/D timer initial phase is 0.05us and the timer cycle is 200us. Start SG3 every cycle, and perform 1 scan by multicycle scan mode. Store the AN000 conversion value to the variable at scan group end.

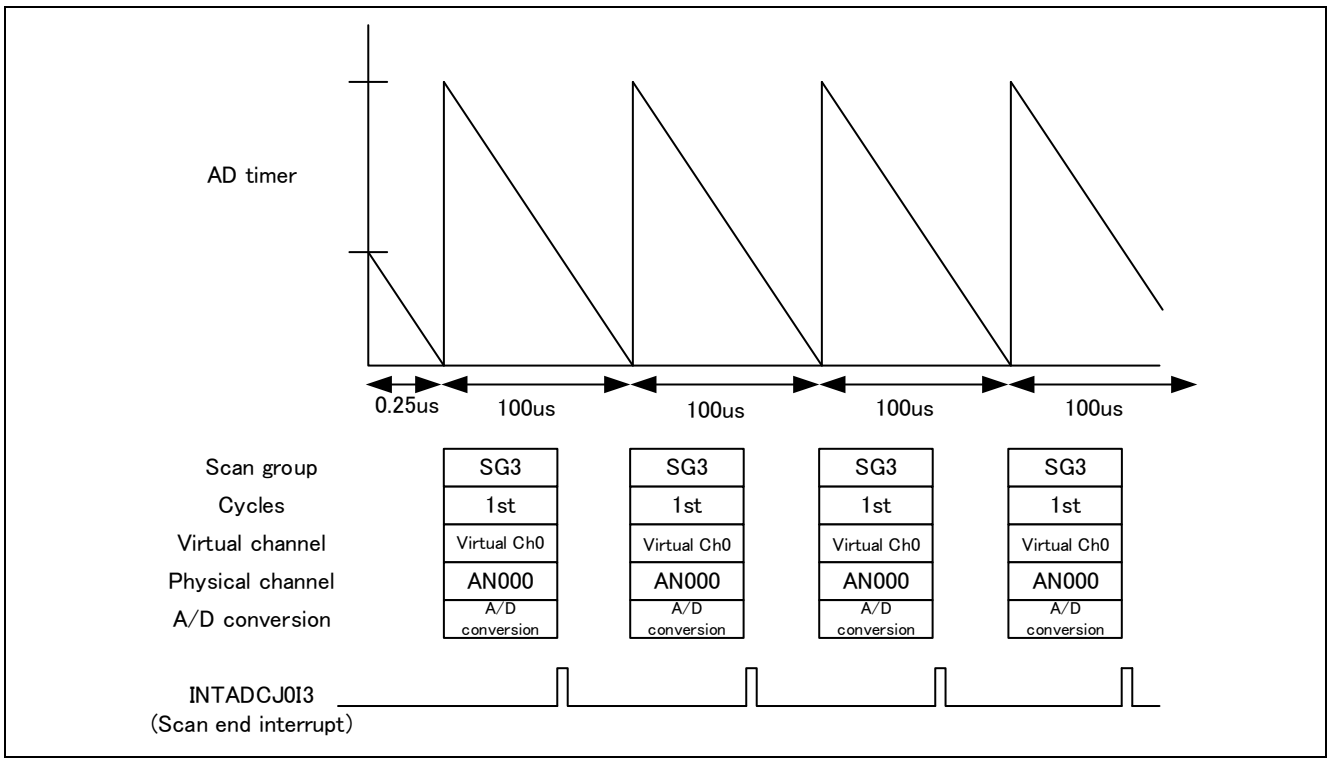

Figure 3-1 A/D Conversion Operation by A/D timer Trigger

3.3.2 Use Function

The used hardware functions in this operation example are shown below.

Input to AN000 generates the sine wave by the pulse generator.

- A/D convertor（ADCJ0）
- Pulse generator

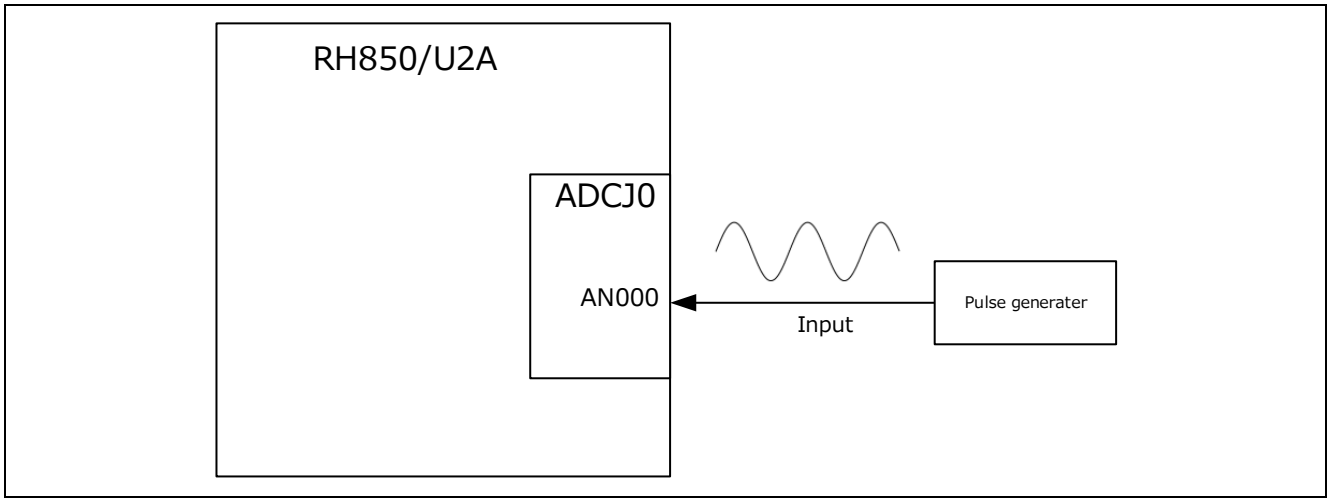

Figure 3-2 System Configuration

3.3.3 Explanation for Operation Example

In this operation example, select A/D timer as the trigger and start the scan group in constant period to use AN000 of ADCJ module.

Allocate scan group 3 (SG3) incorporating A/D timer to the virtual channel 0 (AN000).

Set 0.05us to the initial phase (ADCJ0ADTIPR3) and 200us to the A/D timer cycle (ADCJ0ADTPRR3).

When start A/D timer, the initial phase (ADCJ0ADTIPR3) is loaded to the A/D timer and down-counted. When A/D timer becomes 0, the A/D timer trigger is outputted and SG3 A/D conversion is started. Also, the A/D timer cycle (ADCJ0ADTPRR3) is loaded to the A/D timer and start the down-count, then the A/D conversion is started when the counter become 0. After that, the A/D timer cycle is loaded, and the A/D conversion operation is repeated.

Enable the scan end interrupt INTADCJ0I3, and store the A/D conversion value to the variable in the interrupt processing after A/D conversion ends.

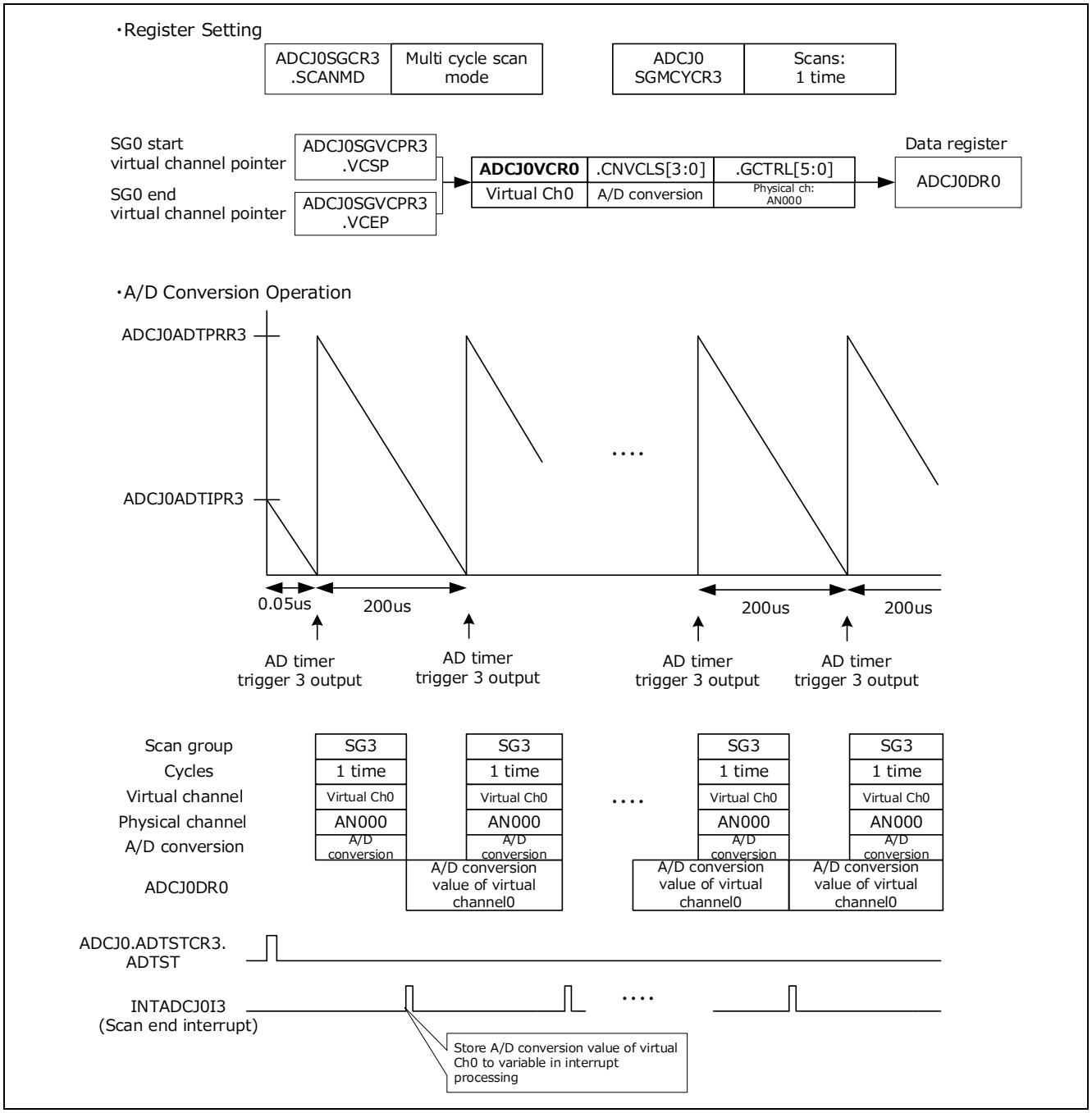

Figure 3-3 A/D timer Operation Value

3.3.4 Software Explanation

・Module Explanation

The module list in this operation example is shows below.

Table 3.3.1 Module List

Module Name	Label Name	Function
Maine routine	main_pm0	Perform various setting and application start.
ADCJ initialize routine	ADCJ_init	Perform ADCJ initialization.
ADCJ interrupt processing routine	INTADCJ0I3	Store A/D conversion result to variable in virtual scan group end interrupt
Interrupt initialize routine	intc2_init	Perform ADCJ interrupt initialization.

・Register Setting

The Register setting of each function in this operation example is shown below.

Table 3.3.2 ADCJ Register Setting

Register Name	Setting Value	Function
ADCJ0ADCR1	0x00	Synchronous suspend
ADCJ0ADCR2	0x10	Signed integer format Disable since CNVCL = 0
ADCJ0SFTCR	0x00	Disable variable setting
ADCJ0VCLMINTERx	0x00000000	Not use upper/lower limit check interrupt table
ADCJ0VMONVDCR1	0x00	Not use divide voltage resistance
ADCJ0VMONVDCR2	0x00	Not use divide voltage resistance
ADCJ0SMPCR	0x00000000	Not use buffer amp filter
ADCJ0TDCR	0x00	Not perform pin level self-diagnosis
ADCJ0ODCR	0x0000	Not perform disconnection detection
ADCJ0TOCCR	0x00	Not perform trigger overlap check.
ADCJ0GTMENTSGER	0x0000	Not GTM entry
ADCJ0ADENDP0	0x00	Not use A/D conversion monitor
ADCJ0THCR	0x00	Not use T&H
ADCJ0THER	0x00	Not use T&H
ADCJ0WAITTR0	0x0000	Unuse since WTTS[3:0]=0
ADCJ0THACR	0x00	Not use T&H
ADCJ0PWDCR	0x00	Disable PWM-Diag
ADCJ0SGVCPR3	0x0000	End virtual channel: 0 Start virtual channel: 0
ADCJ0SGCR3	0x92	ADTSTART enable Enable ADSTART Multicycle scan mode Output INTADCJ0I0 when SG0 end Disable H/W trigger input to SG0
ADCJ0SGMCYCR3	0x00	Scan 1 time in multicycle scan mode

Register Name	Setting Value	Function
ADCJ0VCR0	0x00000000	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion Not use multi plexor Not output VCRj end interrupt Physical CH0 (AN000)
ADCJ0ADTIPR3	0x00000001	A/D timer 3 initial phase: 0.05us
ADCJ0ADTPRR3	0x00000FA0	A/D timer cycle: 200us

Table 3.3.3 Interrupt Register Setting

Register Name	Setting Value	Function
EIBD230	0x00000000	Bind interrupt to PE0
EIC230	0x0040	Table reference/priority level 1

・Operation Flow

The used hardware function in this operation example is shown below.

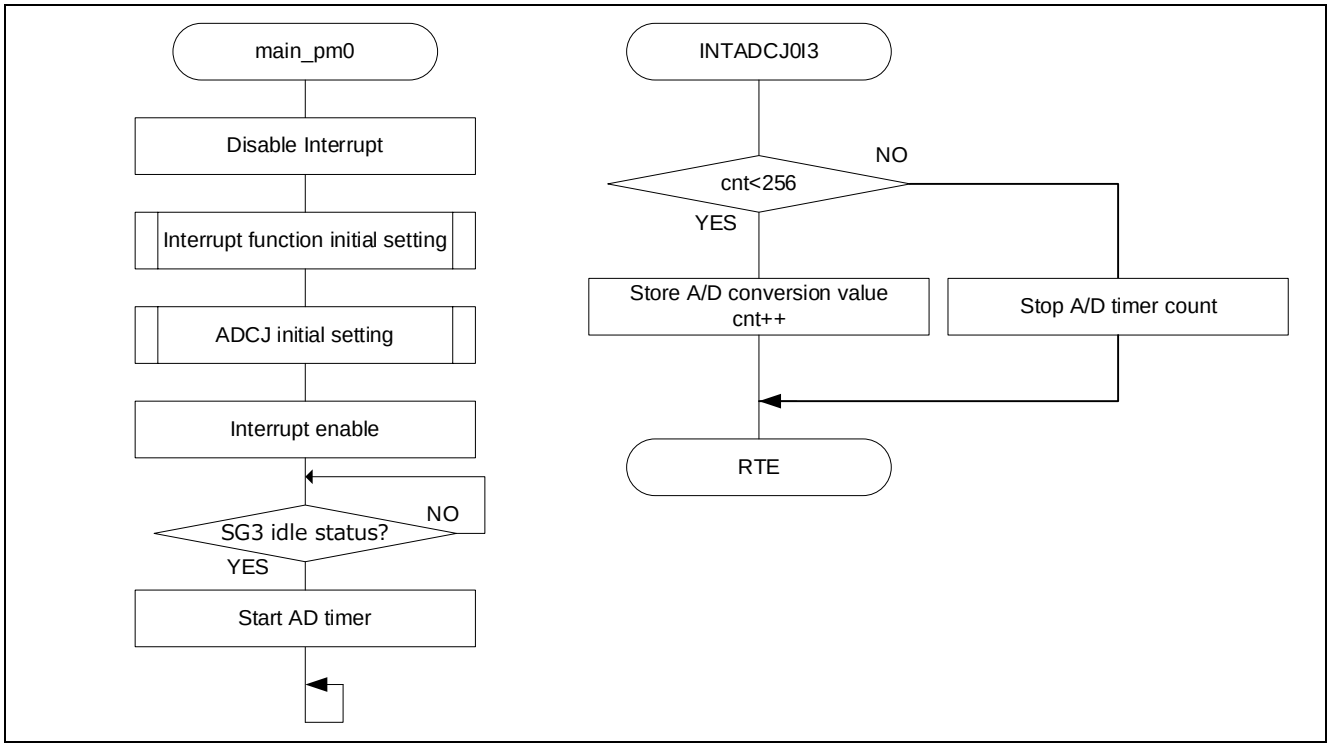

Figure 3-4 Flowchart

3.4.1 Specification Overview

This section explains the normal A/D conversion performing method in the addition mode.

Allocate two virtual channels (AN000, AN001) to the scan group 0 (SG0), scan 1 time in the multicycle scan mode, and set addition 4 times for each virtual channel. Store the conversion value of AN000 and AN001 to the variable at end of the operation and end the operation.

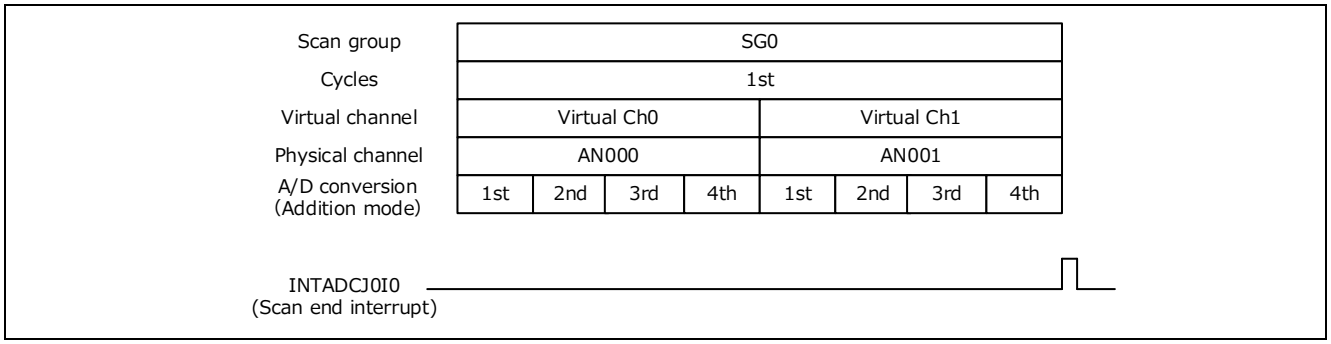

Figure 3-5 Normal A/D conversion Operation using Addition Mode

3.4.2 Use Function

The used hardware function in this operation example is shown below.

- A/D convertor (ADCJ0)

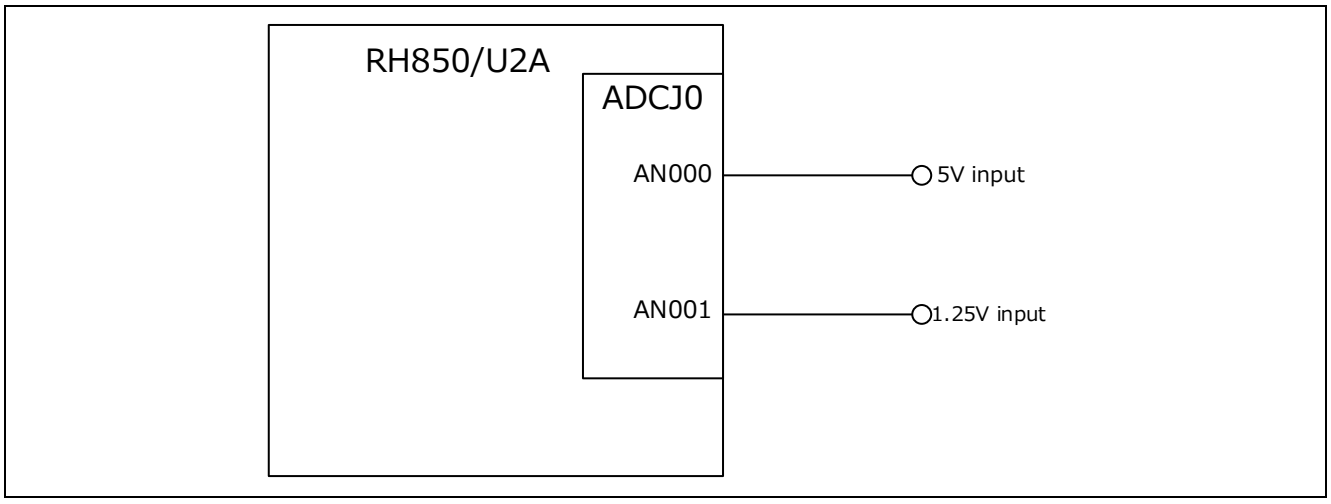

Figure 3-6 System Configuration Diagram

3.4.3 Operation Explanation

In this operation example, perform the normal A/D conversion of the addition mode using AN000 and AN001 of ADCJ module.

Allocate the virtual channel 0 (AN000) and virtual channel 1 (AN001) to the scan group 0 (SG0).

For the analog signal, input 5.0V to AN000 and 1.25V to AN001 of ADCJ0 module.

Start by the software trigger SGST, and perform the AN000 and AN001 A/D conversion for the number of additions (4 times). Enable the scan end interrupt (when scan group ends), store the result of adding each A/D conversion 4 times in interrupt processing to the variable, and end the operation.

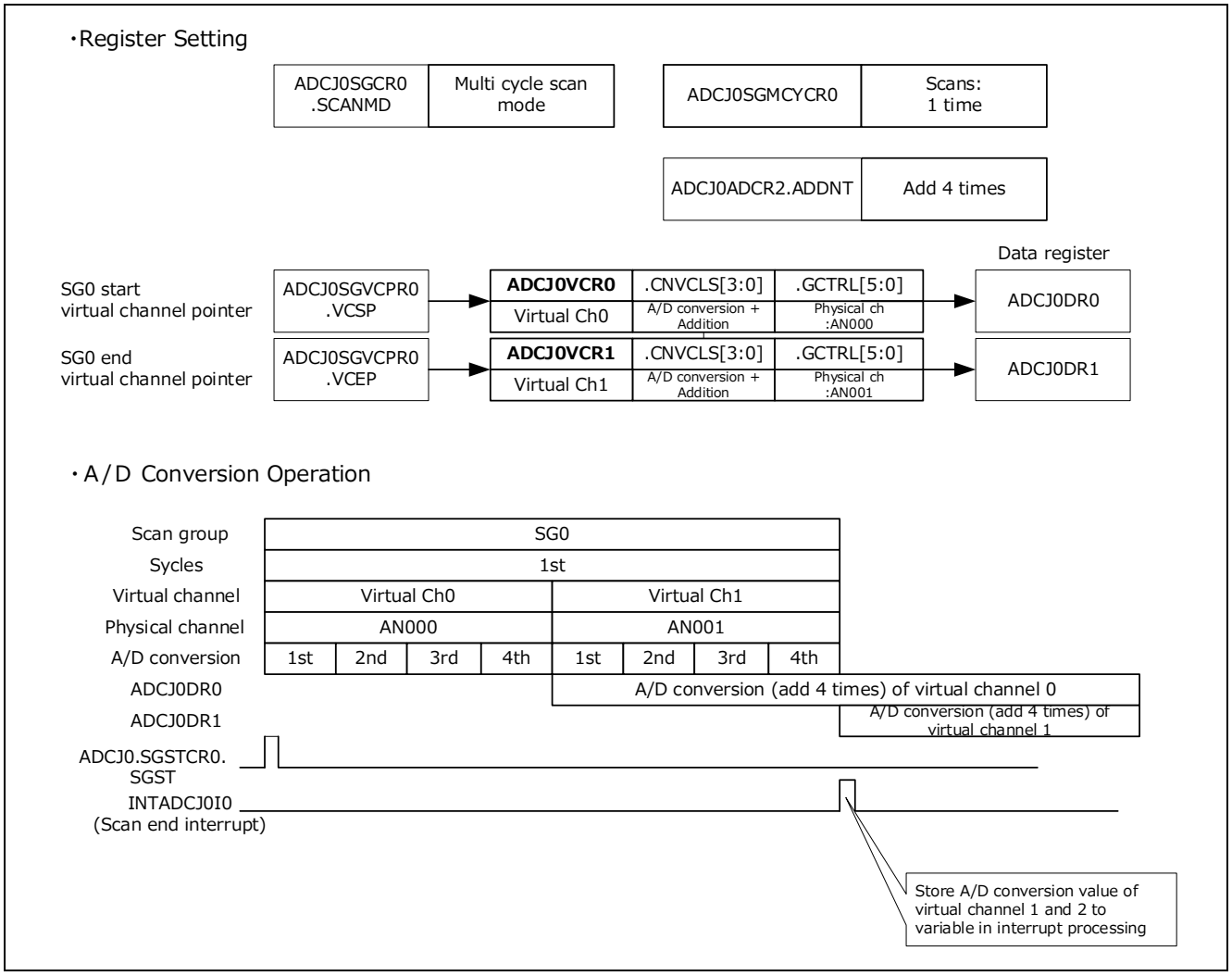

Figure 3-7 Addition Function Operation Example

3.4.4 Software Explanation

・Module Explanation

The module list in this operation example is shows below.

Table 3.4.1 Module List

Module Name	Label Name	Function
Maine routine	main_pm0	Perform various setting and application start.
ADCJ initialize routine	ADCJ_init	Perform ADCJ initialization.
ADCJ interrupt processing routine	INTADCJ0I0	Store A/D conversion result to variable in virtual scan group end interrupt
Interrupt initialize routine	intc2_init	Perform ADCJ interrupt initialization.

・Register Setting

The Register setting of each function in this operation example is shown below.

Table 3.4.2 ADCJ Register Setting

Register Name	Setting Value	Function
ADCJ0ADCR1	0x00	Synchronous suspend
ADCJ0ADCR2	0x11	Signed integer format Additions: 4 times
ADCJ0SFTCR	0x00	Disable variable setting
ADCJ0VCLMINTERx	0x00000000	Not use upper/lower limit check interrupt table
ADCJ0VMONVDCR1	0x00	Not use divide voltage resistance
ADCJ0VMONVDCR2	0x00	Not use divide voltage resistance
ADCJ0SMPCR	0x00000000	Not use buffer amp filter
ADCJ0TDCR	0x00	Not perform pin level self-diagnosis
ADCJ0ODCR	0x0000	Not perform disconnection detection
ADCJ0TOCCR	0x00	Not perform trigger overlap check
ADCJ0GTMENTSGER	0x0000	Not GTM entry
ADCJ0ADENDP0	0x00	Not use A/D conversion monitor
ADCJ0THCR	0x00	Not use T&H
ADCJ0THER	0x00	Not use T&H
ADCJ0WAITTR0	0x0000	Unuse since WTTS[3:0]=0
ADCJ0THACR	0x00	Not use T&H
ADCJ0PWDCR	0x00	Disable PWM-Diag
ADCJ0SGVCPR0	0x0100	End virtual channel: 1 Start virtual channel: 0
ADCJ0SGCR0	0x50	Enable ADSTART MulticycleMulticycle scan mode Output INTADCJ0I0 when SG0 end Disable H/W trigger input to SG0
ADCJ0SGMCYCR0	0x00	Scan 1 time in multicyclemulticycle scan mode

Register Name	Setting Value	Function
ADCJ0VCR0	0x00002000	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion Not use multiplexer Not output VCRj end interrupt Physical CH0 (AN000)
ADCJ0VCR1	0x00002001	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion Not use multiplexer Not output VCRj end interrupt Physical CH1 (AN001)

Table 3.4.3 Interrupt Register

Register Nama	Setting Value	Function
EIBD227	0x00000000	Bind interrupt to PE0
EIC227	0x0040	Table reference/priority level 0

・Operation Flow

The flowchart of the operation example in this operation example is shown below.

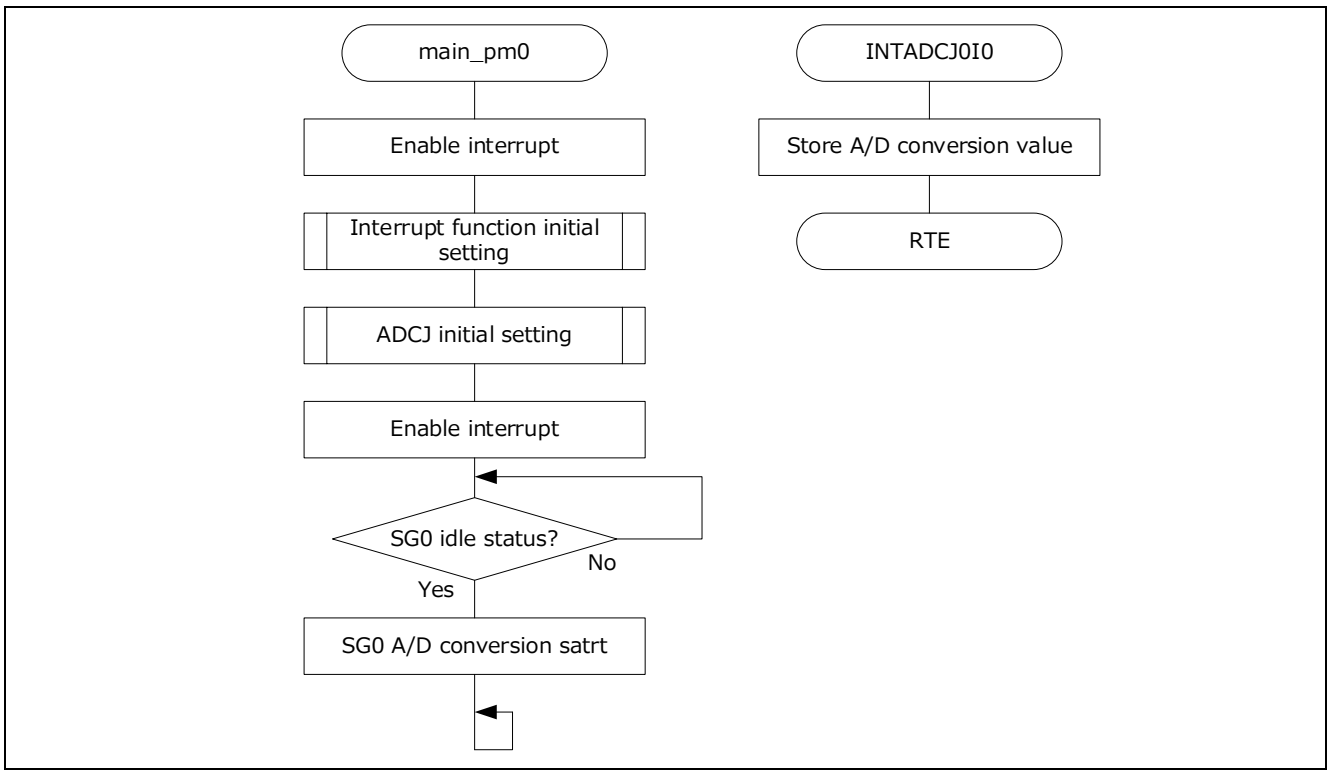

Figure 3-8 Flowchart

3.5 A/D Conversion using External Analog Multiplexer

3.5.1 Specification Overview

This section explains the A/D conversion method using the external analog multiplexer and the normal A/D conversion w/MPX mode.

Allocate the six virtual channels to the scan group (SG0). Use the external multiplexer input pin for the physical channel.

Start DMA (sDMAC) for each virtual channel starts, transfer the MPX value for the external analog multiplexer control to the I/O port, and control the analog multiplexer. Store the conversion value to the variable in the SG0 scan group end.

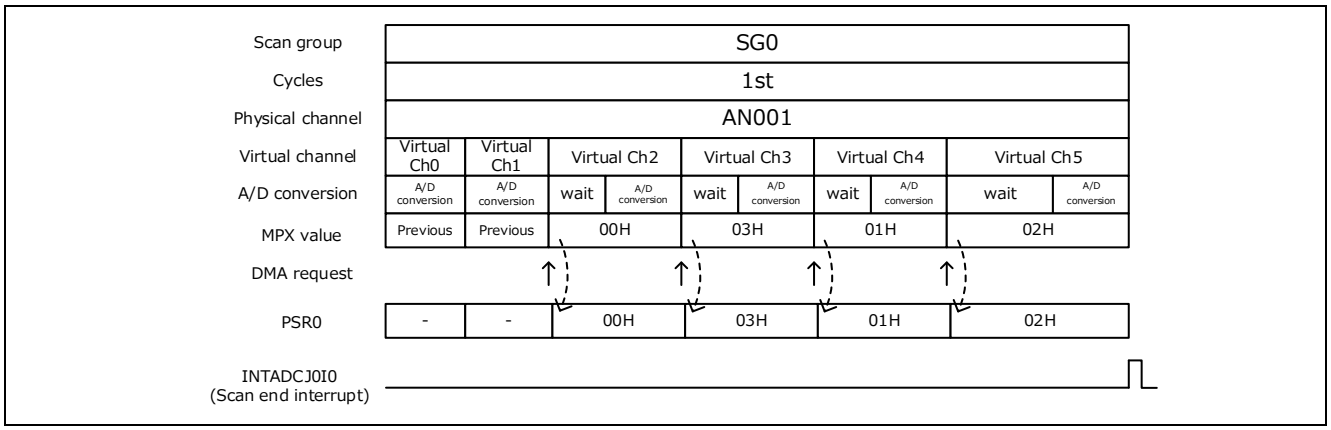

Figure 3-9 Operation Example of A/D Conversion using External Analog Multiplexer

3.5.2 Use Function

The used hardware functions in this operation example are shown below.

The input to ADCJ can be switched to control the external multiplexer by the MPX value transferred to I/O port. In this operation example, after normal 2 times conversions, A/D convert 4 out of 6 external multiplexers

- A/D convertor (ADCJ0)
- Pin (PORT0)
- DMA (sDMAC)

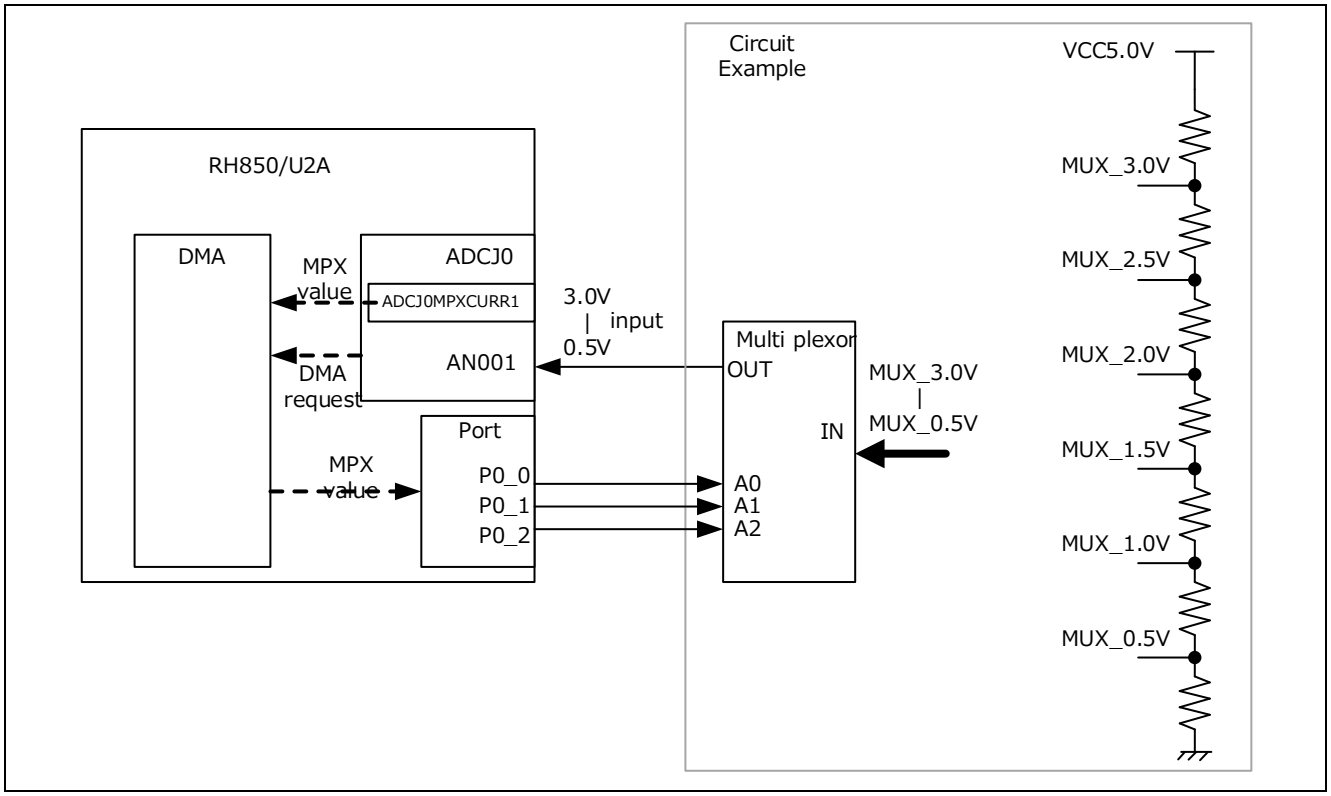

Figure 3-10 System Configuration

3.5.3 Explanation of Operation Example

In this operation example, use AN001 of ADCJ0 module and perform the A/D conversion using the external analog multiplexer.

ADCJ supports the interrupt for the external analog multiplexer and the DMA request, and it can be generated the interrupt request generation for INTC and started DMA (sDMAC) when the specified virtual channel starts.

In this operation example, transfer the MPX value to the external analog multiplexer with the I/O port.

Allocate the virtual channel 0 to 5 to the scan group (SG0).

For analog signal, input AN001 from the external analog multiplexer.

Set the MPX value transferred to the external analog multiplexer to ADCJ0VCRj.MPXV[2:0]of each virtual channel.

When ADCJ0VCRj.CNVCLS[3:0] is 0x5 (normal A/D conversion w/MPX) or 0x6 (addition mode A/D conversion w/MPX), transfer the MPX value (ADCJ0VCRj.MPXV[2:0]) to the MPX current register (ADCJ0MPXCURR1) in the virtual channel starts, issue the DMA request (ADMPXI0), and start DMA(sDMAC).

For DMA(sDMAC), set ADCJ0MPXCURR1 as the transfer source and PSR0 register as the transfer destination. Start DMA(sDMAC) for each SGO virtual channel start, and transfer the set MPX value from P0_0, P0_1, and P0_2 to the external analog multiplexer. Control the external multiplexer by the transferred MPX value.

Enable the scan end interrupt INTADCJ0I0, and store the A/D conversion value to the variable in interrupt processing after A/D conversion.

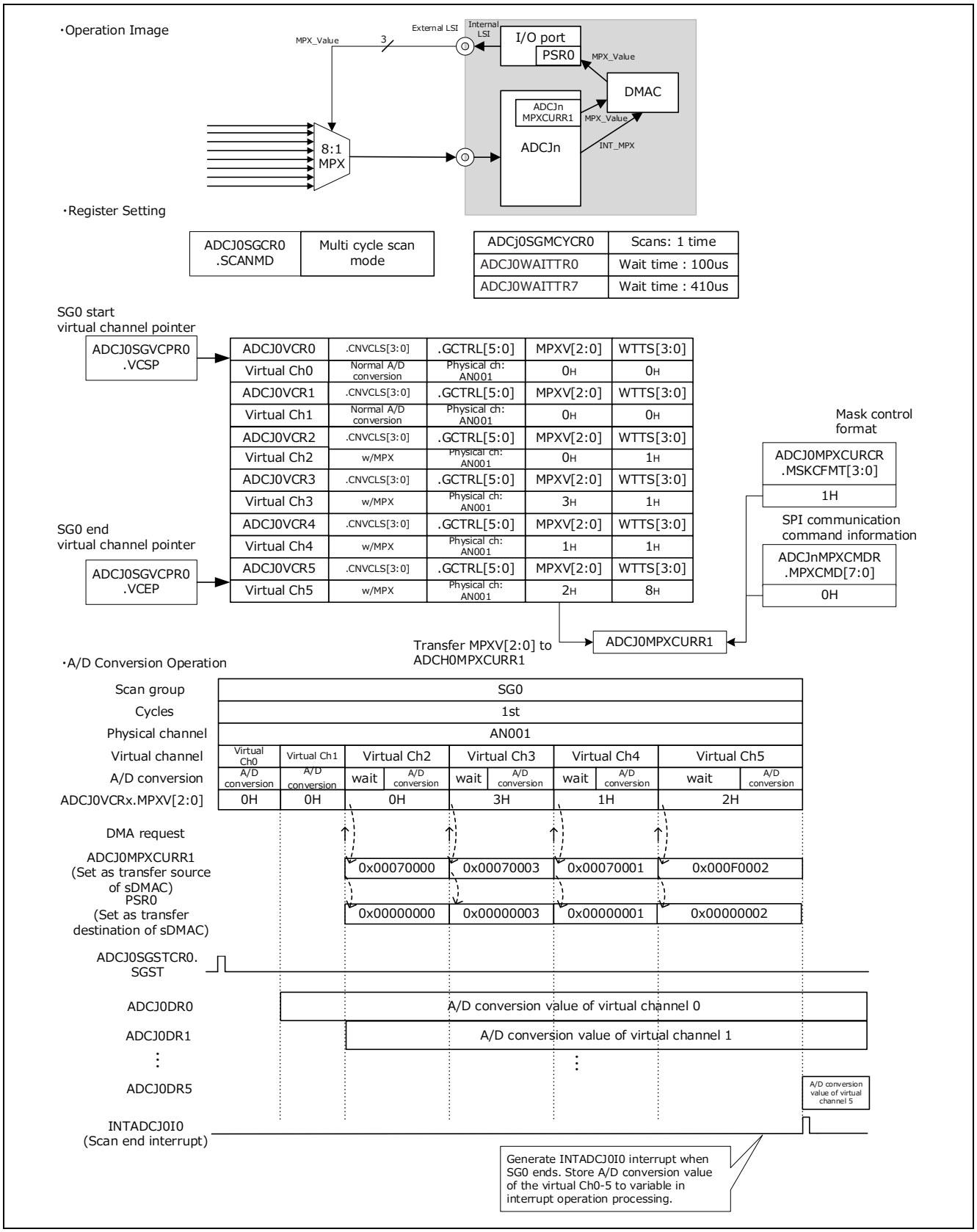

Figure 3-11 Operation Example using External Analog Multiplexer and DMA

3.5.4 Software Explanation

・Module Explanation

The module list in this operation example is shows below.

Table 3.5.1 Module List

Module Name	Label Name	Function
Maine routine	main_pm0	Perform various setting and application start.
Port initialize routine	port_init	Perform port initialization.
ADCJ initialize routine	ADCJ_init	Perform ADCJ initialization.
ADCJ/SG0 interrupt processing routine	INTADCJ0I0	Store A/D conversion result to variable in virtual scan group end interrupt processing.
DMA(sDMAC) end interrupt routine	INTSDMAC0CH0	Perform DMA flag clear in virtual scan group end interrupt processing.
DMA initialize routine	sdmac_init	Perform DMA initialization.
Interrupt initialize routine	intc2_init	Perform ADCJ interrupt initialization.

・Register Setting

The Register setting of each function in this operation example is shown below.

Table 3.5.2 ADCJ Register Setting

Register Name	Setting Value	Function
ADCJ0ADCR1	0x00	Synchronous suspend
ADCJ0ADCR2	0x10	Signed integer format Disable since CNVCL = 0
ADCJ0SFTCR	0x00	Disable variable setting
ADCJ0VCLMINTERx	0x00000000	Not use upper/lower limit check interrupt table
ADCJ0VMONVDCR1	0x00	Not use divide voltage resistance
ADCJ0VMONVDCR2	0x00	Not use divide voltage resistance
ADCJ0SMPCR	0x00000000	Not use buffer amp filter
ADCJ0TDCR	0x00	Not perform pin level self-diagnosis
ADCJ0ODCR	0x0000	Not perform disconnection detection
ADCJ0TOCCR	0x00	Not perform trigger overlap check.
ADCJ0GTMENTSGER	0x0000	Not GTM entry
ADCJ0ADENDP0	0x00	Not use A/D conversion monitor
ADCJ0THCR	0x00	Not use T&H
ADCJ0THER	0x00	Not use T&H
ADCJ0THACR	0x00	Not use T&H
ADCJ0PWDCR	0x00	Disable PWM-Diag
ADCJ0MPXCURCR	0x01	Mask control format: MPXCURR1.MSCK[3:0]=0111
ADCJ0MPXINTER	0x01	Enable MPX interrupt
ADCJ0MPXCMDR	0x00	Unuse SPI
ADCJ0SGVCPR0	0x0500	End virtual channel: 5 Start virtual channel: 0

Register Name	Setting Value	Function
ADCJ0SGCR0	0x50	Enable ADSTART MulticycleMulticycle scan mode Output INTADCJ0I0 when SG0 end Disable H/W trigger input to SG0
ADCJ0SGMCYCR0	0x00	Scan 1 time in multicyclemulticycle scan mode
ADCJ0VCR0	0x00000001	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion Not use multiplexer Not output VCRj end interrupt Physical CH1 (AN001)
ADCJ0VCR1	0x00000001	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion Not use multiplexer Not output VCRj end interrupt Physical CH1 (AN001)
ADCJ0VCR2	0x01002801	Not use upper/lower limit check Wait time table: ADCJ0WAITTR0 Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion w/MPX MPX value: 0 Not output VCRj end interrupt Physical CH1 (AN001)
ADCJ0VCR3	0x01002B01	Not use upper/lower limit check Wait time table: :ADCJ0WAITTR0 Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion w/MPX MPX value: 3 Not output VCRj end interrupt Physical CH1 (AN001)
ADCJ0VCR4	0x01002901	Not use upper/lower limit check Wait time table: :ADCJ0WAITTR0 Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion w/MPX MPX value: 1 Not output VCRj end interrupt Physical CH1 (AN001)

Register Name	Setting value	Function
ADCJ0VCR5	0x08002A01	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion MPX value: 2 Not output VCRj end interrupt Physical CH1 (AN001)
ADCJ0WAITTR0	0x0FA0	Wait time: 100μs
ADCJ0WAITTR7	0x3FFF	Wait time: 410 μs.

Table 3.5.3 DMA Register Setting

Register Name	Setting Value	Function
DMA0CM_0	0x00001C00	Channel master SPID setting SPID=0x1C (initial value) Supervisor mode
DMA0SAR_0	Written value on right	Transfer source：ADCJ0MPXCURR1 register
DMA0DAR_0	Written value on right	Transfer destination：PSR0 register
DMA0TSR_0	0x00000004	Transfer size：4 bytes
DMA0TMR_0	0x00001022	DMA transfer mode: normal mode Channel priority DMA transfer request selection interrupt: hardware DMA transfer request Destination address/count direction: fixed Source address/count destination: fixed DMA transfer transaction size: 4 bytes DMA source transaction size: 4 bytes
DMA0RS_0	0x00010005	Transfers per hardware request: 1 time Transfer upper limit per hardware request Not use since PLE=0 Disable preload Disable DRQ initialization Hardware DMA transfer source selection: (MPX DMA trigger request (ADMPXI0))
DMA0CHFCR_0	0x0000320F	Clear each flag
DMA0OR	0x0001	Priority: CH0 > CH1 > ... > CH14 > CH15 Enable DMA transfer
DMA0CHCR_0	0x0003	Disable descriptor Disable descriptor Disable address error notification Disable channel address error notification Disable descriptor step end interrupt Enable transfer completion interrupt Enable channel operation

Table 3.5.4 Port Register Setting

Register Name	Setting Value	Function
PSR0	0x00000000	Port set reset: P0_m initial value
PKCPROT	0xA5A5A501	Port key code
PWE	0xFFFFFFFE	Enable write
PM0	0x00000000	P0_m output mode

Table 3.5.5 Interrupt Register

Register Name	Setting Value	Function
EIBD47	0x00000000	Bind interrupt to PE0
EIC47	0x0040	Table reference/priority level 0
EIBD227	0x00000000	Bind interrupt to PE0
EIC227	0x0040	Table reference/priority level 0

Table 3.5.6 PBG Register

Register Name	Setting Value	Function
PBGERRSLV20PBGKCPROT	0xA5A5A501	PBG key code
PBG21PBGPROT0_4	0x00000143	Port group21,ch4 register: Enable write Port group21,ch4 register: Enable write
PBG21PBGPROT1_4	0x10000000	SPID28: Enable access

・Operation Flow

The flowchart of the operation example in this operation example is shown below.

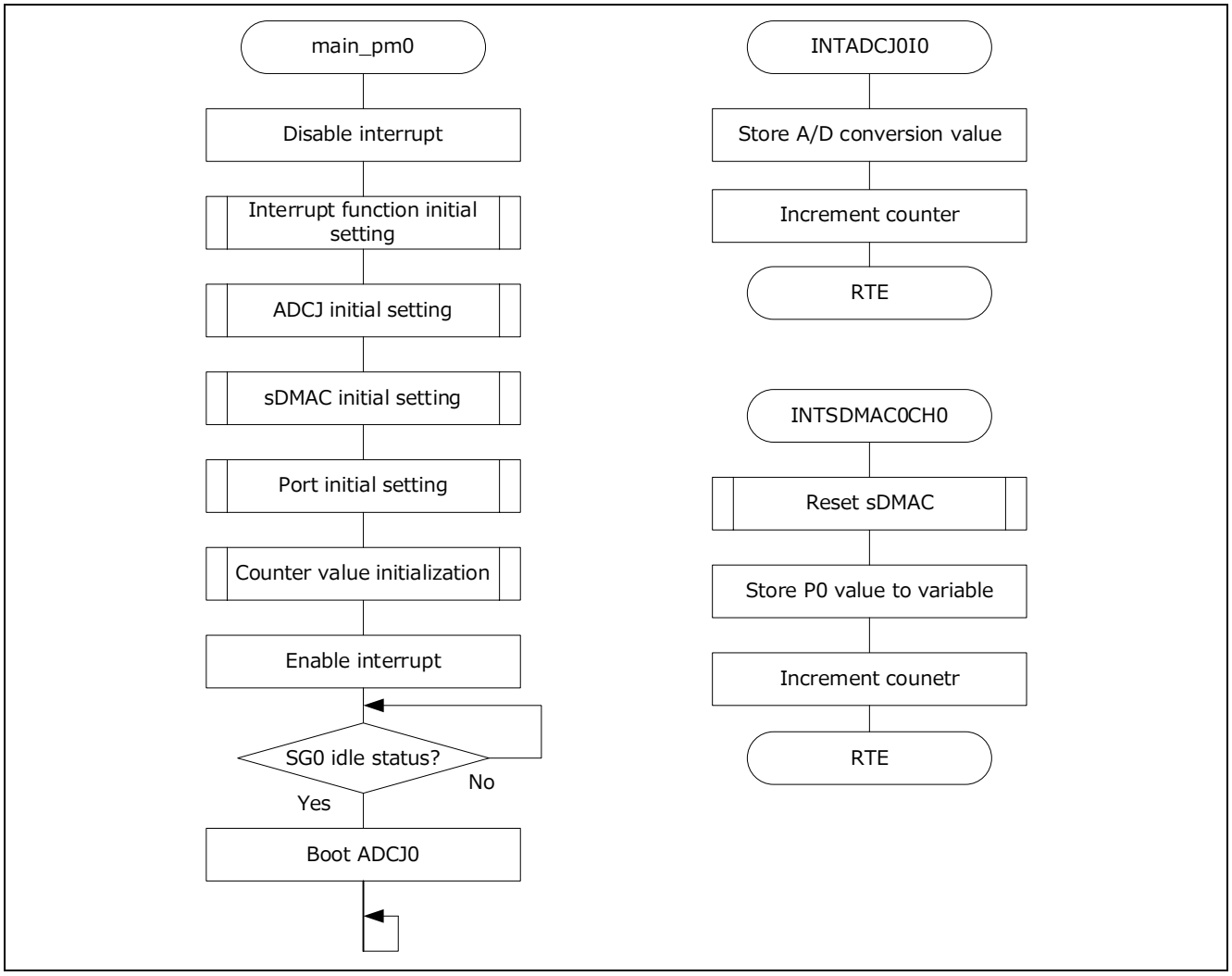

Figure 3-12 Flowchart

3.6 Concurrent Track & Hold Operation

3.6.1 Specification Overview

This section explains A/D conversion performing method using T&H(Concurrent Track & Hold).

Allocate 3 virtual channels (AN000, AN001, AN002) to the scan group 1 (SG1), and scan in multicycle scan mode. As A/D conversion start trigger, it is used TSG30ADTRG0 of TSG30. Store the AN000, AN001, and AN002 conversion value to the variable at end of each scan group, and end operation.

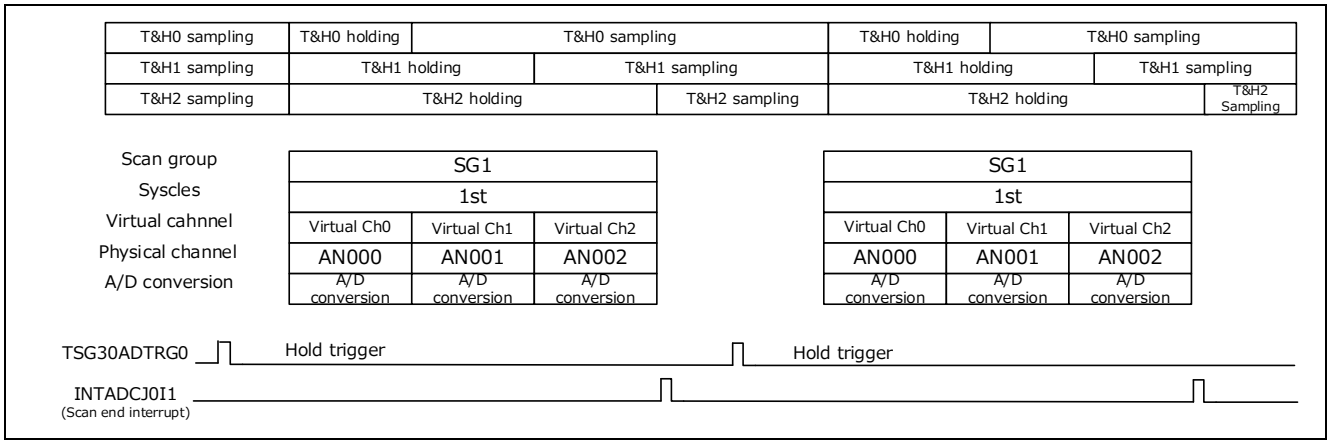

Figure 3-13 A/D Conversion (Multicycle scan) Operation Using T&H

3.6.2 Use Function

The used functions in this operation example are shown below.

- A/D convertor（ADCJ0）
- Motor control timer（TSG30）*1
- OS timer（OSTM）

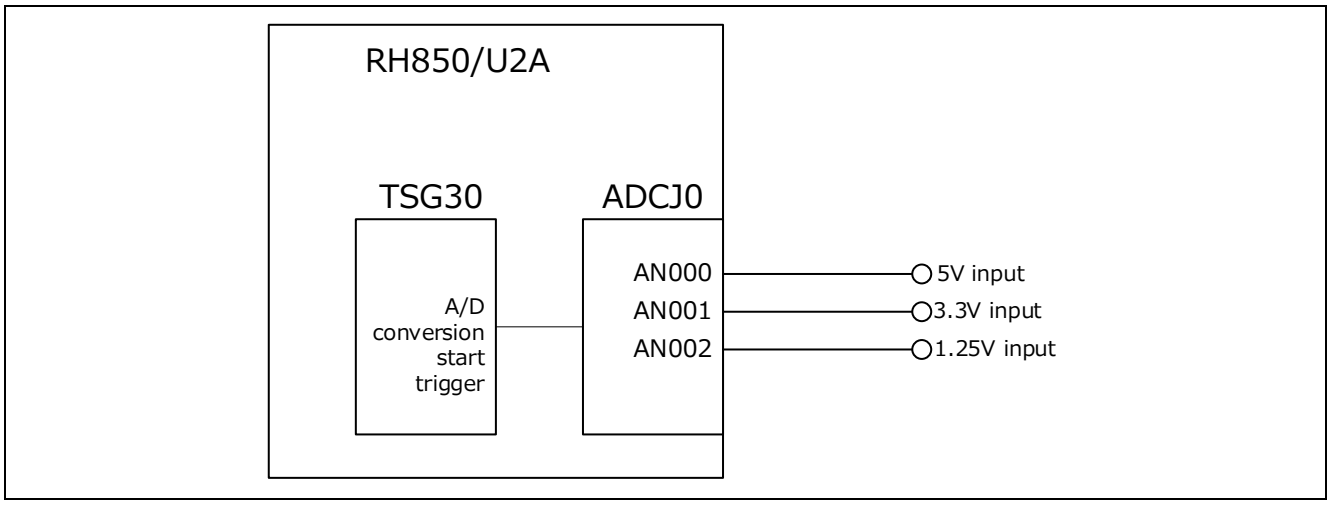

Figure 3-14 System Configuration

*1 When operate the sample soft by U2A6 144 pins or 156 pins, use TSG31.

3.6.3 Explanation of Operation Example

In this operation example, use AN000, AN001, and AN002 of ADCJ0 module, and perform the A/D conversion using T&H (concurrent track & hold).

ADCJ contains the track & hold circuit for 4 channels. It supports the concurrent track & hold up to 4 channels.

Allocate the virtual channel 0, 1, and 2 to the scan group 1 (SG1). Input analog signal to AN000, AN001, and AN002 . Insert the sampling cycle wait time of the track & hold circuit using OS timer.

Generate A/D conversion start trigger using TSG30.

Start TSG30 count, and A/D convert the hold value. Enable the scan end interrupt INTADCJ0I1 (when scan group end), store the A/D conversion result to the variable in each interrupt processing, and end operation.

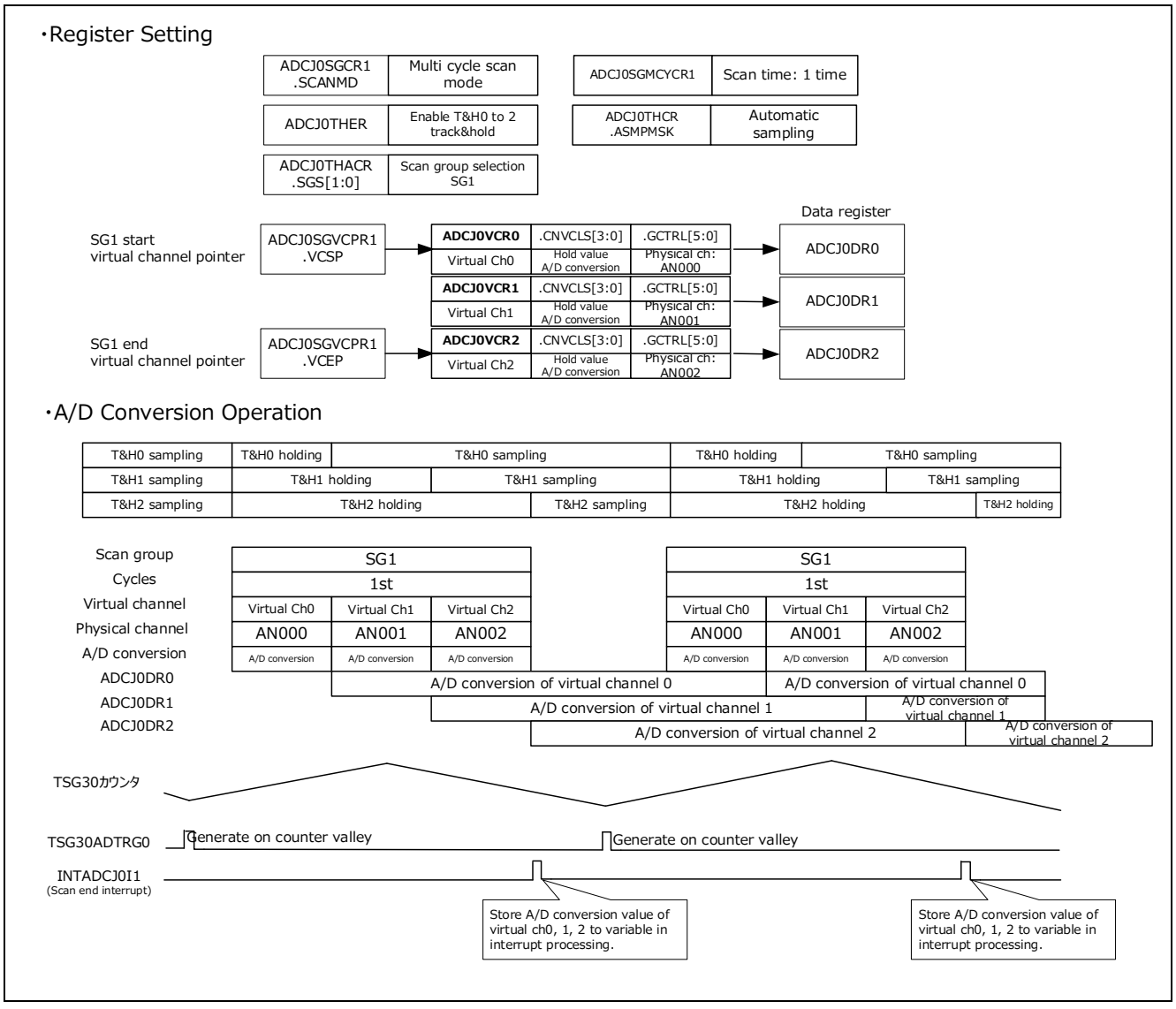

Figure 3-15 A/D Conversion Operation Example using T&H Function

3.6.4 Software Explanation

・Module Explanation

The module list in this operation example is shows below.

Table 3.6.1 Module List

Module Name	Label Name	Function
Maine routine	main_pm0	Perform various setting and application start.
Port initialize routine	port_init	Perform port initialization.
ADCJ initialize routine	ADCJ_init	Perform ADCJ initialization.
OSTM initialize routine	ostm_init	Perform OSTM initialization.
TSG3 initialize routine	tsg3_init	Perform TSG3 initialization.
Wait processing routine	wait	Perform wait of 30CLK_ADC.
ADCJ/SG1 interrupt processing routine	INTADCJ0I1	Store A/D conversion result to variable in virtual scan group end interrupt

・Register Setting

The Register setting of each function in this operation example is shown below.

Table 3.6.2 ADCJ Register Setting

Register Name	Setting Value	Function
ADCJ0ADCR1	0x02	Not Synchronous suspend
ADCJ0ADCR2	0x10	Signed integer format Disable since CNVCL = 0
ADCJ0SFTCR	0x00	Disable variable setting
ADCJ0VCLMINTERx	0x00000000	Not use upper/lower limit check interrupt table
ADCJ0VMONVDCR1	0x00	Not use divide voltage resistance
ADCJ0VMONVDCR2	0x00	Not use divide voltage resistance
ADCJ0SMPCR	0x00000000	Not use buffer amp filter
ADCJ0TDCR	0x00	Not perform pin level self-diagnosis
ADCJ0ODCR	0x0000	Not perform disconnection detection
ADCJ0TOCCR	0x00	Not perform trigger overlap check.
ADCJ0GTMENTSGER	0x0000	Not GTM entry
ADCJ0ADENDP0	0x00	Not use A/D conversion monitor
ADCJ0THCR	0x00	Automatic sampling
ADCJ0THER	0x07	T&H0 to 2 enable
ADCJ0WAITTR0	0x0000	Unuse since WTTS[3:0]=0
ADCJ0THACR	0x30	Hold control Hold trigger enable SG1 select
ADCJ0PWDCR	0x00	Disable PWM-Diag
ADCJ0SGVCPR1	0x0200	End virtual channel: 2 Start virtual channel: 0
ADCJ0SGCR1	0x51	ADTSTART enable Multicycle scan mode Output INTADCJ0I0 when SG0 end Disable H/W trigger input to SG0
ADCJ0SGMCYCR1	0x00	Scan 1 time in multicycle scan mode

Register Name	Setting Value	Function
ADCJ0VCR0	0x00000800	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: hold value A/D conversion Not use multiplexer Not output VCRj end interrupt Physical CH0 (AN000)
ADCJ0VCR1	0x00000801	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: hold value A/D conversion Not use multiplexer Not output VCRj end interrupt Physical CH1 (AN001)
ADCJ0VCR2	0x00000802	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: hold value A/D conversion Not use multiplexer Not output VCRj end interrupt Physical CH2 (AN002)

Table 3.6.3 Port Register Setting

Register Name	Setting Value	Function
PCR2_0	0x00000000	P2_0 initial value
PCR2_14	0x00000000	P2_14 initial value
PKCPROT	0xA5A5A501	Port key code
PWE	0xFFFFFFFE	Port write interrupt

Table 3.6.4 Interrupt Register Setting

Register Name	Setting Value	Function
EIBD228	0x00000000	Bind interrupt to PE0
EIC228	0x0040	Table reference/priority level 0

Table 3.6.5 OSTM Register Setting

Register Name	Setting Value	Function
OSTM0CMP	0x00000041	Free running mode: compare value
OSTM0T0	0x00	Output level: low
OSTM0CTL	0x02	Disable OSTM interrupt Counter start value: 0x00000000 Free running compare mode Disable interrupt when count start

Table 3.6.6 TSG3 Register Setting

Register Name	Setting Value	Function
TSG30CTL0	0x01	Output pulse by 8 clock width Timer mode: HT-PWM mode
TSG30CTL1	0x0000	Not detect various error
TSG30CTL3	0x00	Reload mode
TSG30CTL4	0x00C0	Enable reload operation in 18 bit counter valley timing Enable peak interrupt generation in 18 bit counter valley timing
TSG30CTL5	0x0100	Treat 18 bit sub-counter valley timing as A/D conversion trigger
TSG30CTL6	0x0000	Not generate A/D conversion trigger
TSG30IOC0	0x7E	Disable TSG3nO6-1 control by TSnIOC2 rewrite
TSG30IOC1	0x00	Output up/down count flag of 18 bit counter
TSG30IOC2	0x7E7E	Active level: low level
TSG30CMP0	0x1F40	TSG30 compare register 0: PWM cycle 100us
TSG30CMPU	0x0FA0	Compare register for TSG30 HT-PWM U phase: U phase duty
TSG30CMPV	0x080C	Compare register for TSG30 HT-PWM V phase: V phase duty
TSG30CMPW	0x1732	Compare register for TSG30 HT-PWM W phase: W phase duty
TSG30DTC0W	0x00C8	Dead time value (between negative phase inactive to positive phase active)
TSG30DTC1W	0x00C8	Dead time value (between positive phase inactive to negative phase active)

Table 3.6.7 PIC Register Setting

Register Name	Setting Value	Function
PIC2ADCJ0TSEL1	0x00000020	Select TSG30ADTRG0 to SG1 trigger source of ADCJ0

・Operation Flow

The flowchart of the operation example in this operation example is shown below.

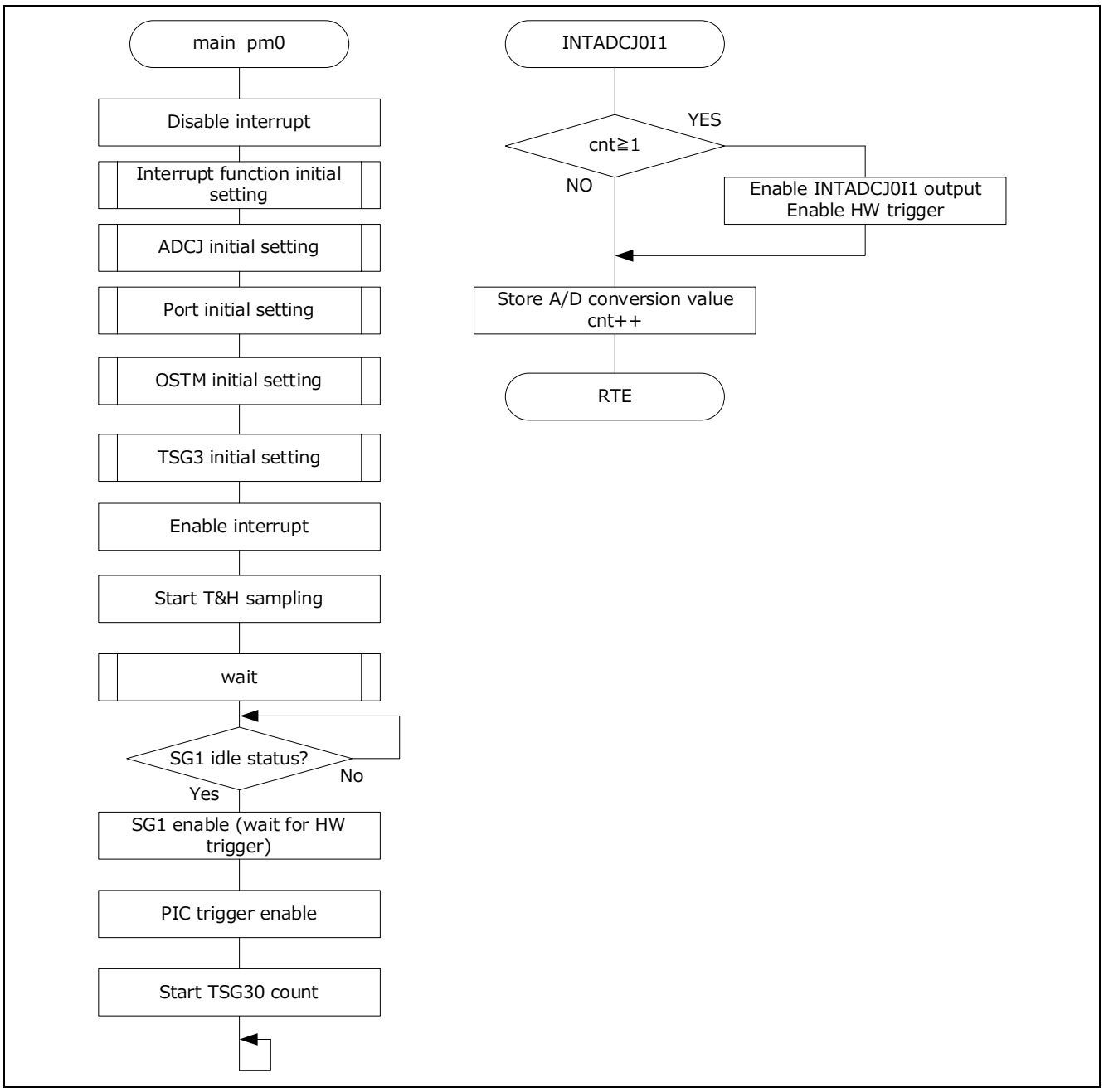

Figure 3-16 Flow Chart

3.7 Sequential Scan Conversion of Arbitrary Channel using DMA Transfer (Scatter Function)

3.7.1 Specification Overview

This section explains the A/D conversion value storing method to the array variable using DMA transfer (scatter function).

Allocate the 3 virtual channels (AN000, AN001, and AN002) to the scan group 0 (SG0), and perform the scan in the multicycle scan mode. Start DMA (sDMAC) by the scan end interrupt (INTADCJ0I0), and store AN000, AN001, and AN002 conversion value to the array variable.

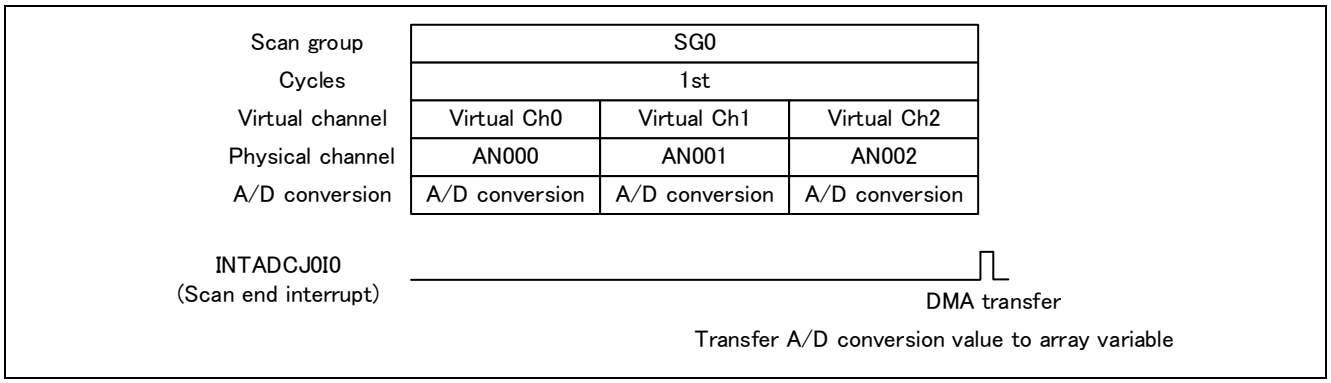

Figure 3.7.1 Operation Example using DMA Transfer

3.7.2 Use Function

The used hardware functions in this operation example are shown below.

- A/D convertor (ADCJ0)
- DMA (sDMAC0)

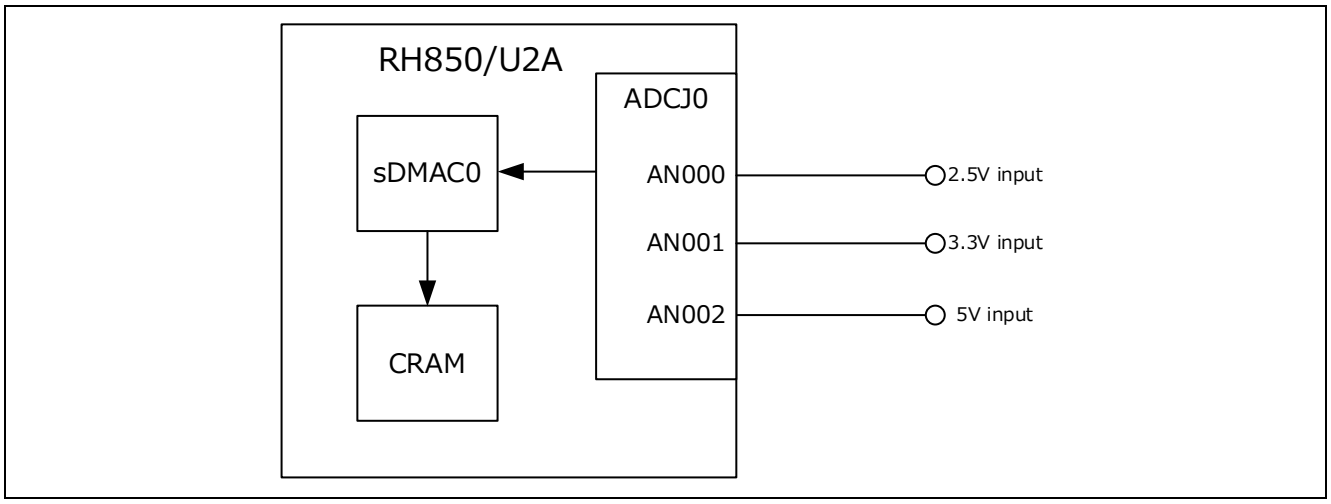

Figure 3.7.2 System Configuration

3.7.3 Explanation of Operation Example

In this operation example, perform the normal A/D conversion in the multicycle scan mode using AN000, AN001, and AN002 of ADCJ module, and store the A/D conversion value to the array variable using the DMA conversion (scatter function).

Allocate the virtual channel 0 (AN000), virtual channel 1 (AN001), and virtual channel 2 (AN002) to the scan group (SG0). For analog signal, input 2.5V to AN000, 3.3V to AN001, and 5V to AN002.

As the start trigger of sDMAC, it is used the scan end interrupt (INTADCJ0I0), and set enable to the scatter function. Figure 3.7.3 shows the scatter function overview. In the scatter function, the A/D conversion value is possible to transfer to each array variable in the sDMAC1 channel.

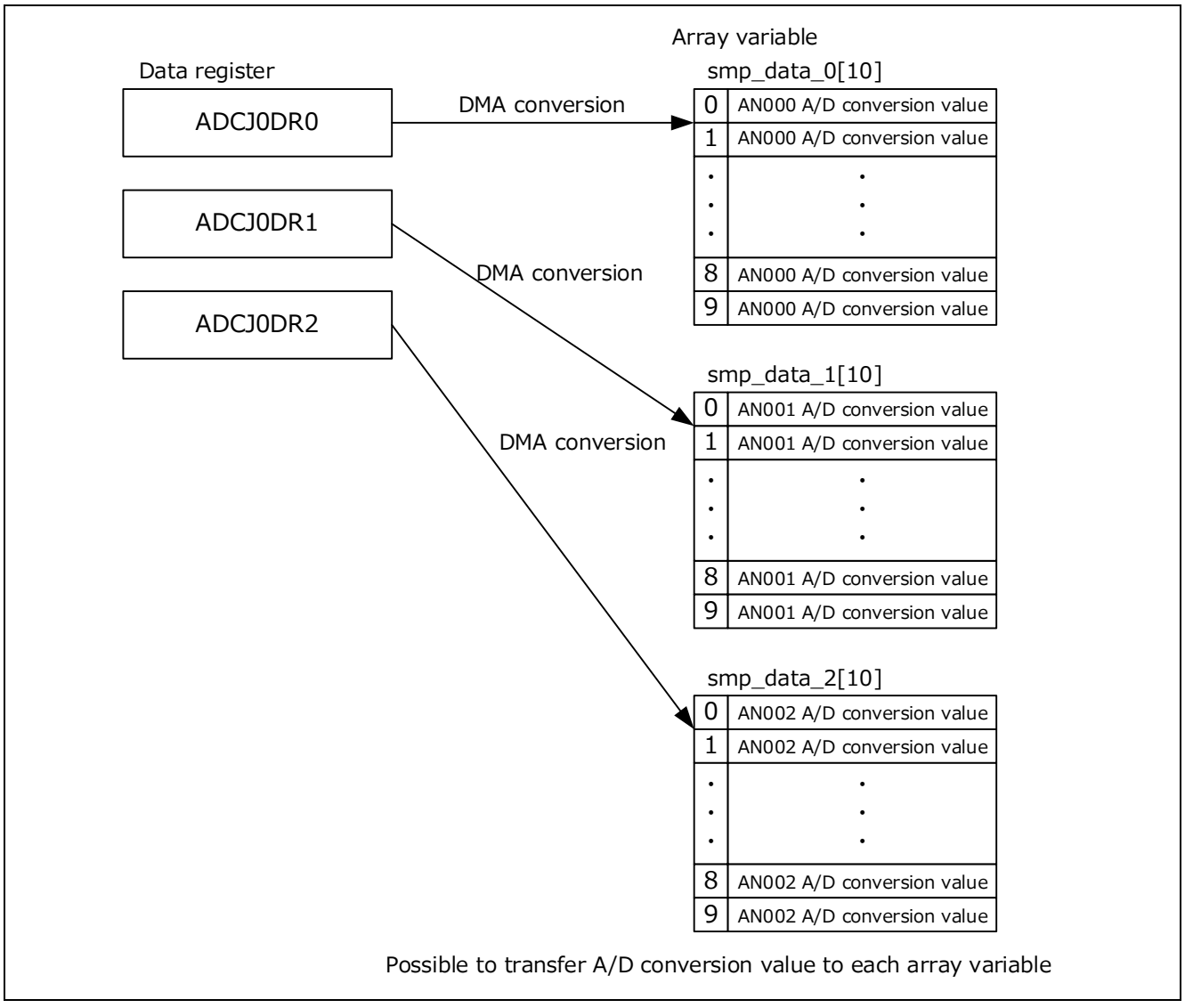

Figure 3.7.3 Scatter Function Overview

Start A/D conversion by the software trigger ADSTART. Started DMA by the scan end interrupt (INTADCJ0I0), and store the A/D conversion value to the array variable.

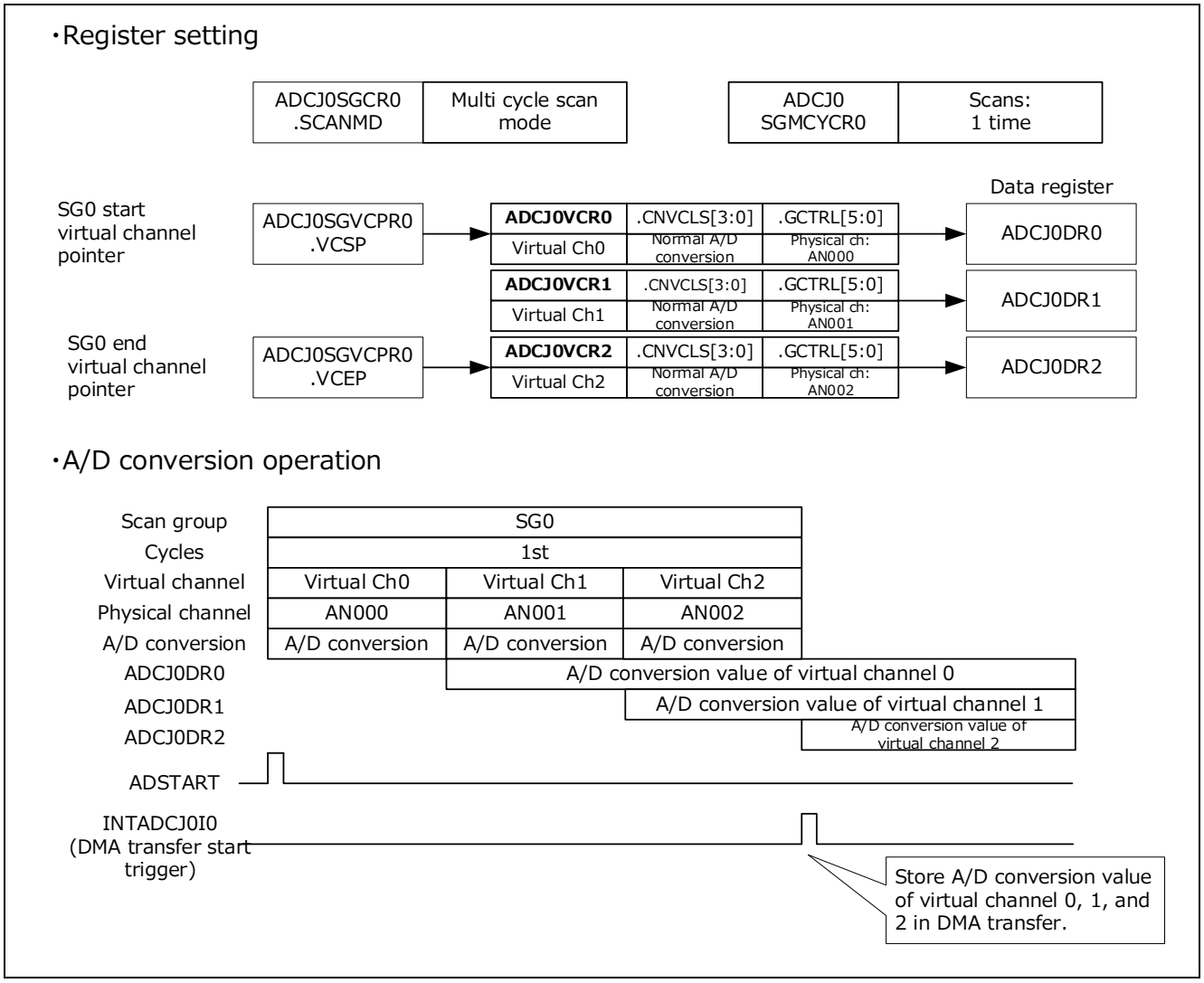

Figure 3.7.4 Operation Example using DMA Transfer

3.7.4 Software Explanation

• Module Explanation

The module list in this operation example is shown below.

Table 3.7.1 Module List

Module Name	Label Name	Function
Maine routine	main_pm0	Perform various setting and application start.
ADCJ initialize routine	ADCJ_init	Perform ADCJ initialization.
sDMAC initialize routine	sdmac_init	Perform sDMAC initialization.
sDMAC transfer completion interrupt processing routine	INTSDMAC0CH0	Perform A/D conversion restart by sDMAC transfer completion interrupt processing.

・Register Setting

The Register setting of each function in this operation example is showing below.

Table 3.7.2 ADCJ Register Setting

Register Name	Setting Value	Function
ADCJ0ADCR1	0x00	Not Synchronous suspend
ADCJ0ADCR2	0x10	Signed integer format Disable since CNVCL = 0
ADCJ0SFTCR	0x00	Disable variable setting
ADCJ0VCLMINTERx	0x00000000	Not use upper/lower limit check interrupt table
ADCJ0VMONVDCR1	0x00	Not use divide voltage resistance
ADCJ0VMONVDCR2	0x00	Not use divide voltage resistance
ADCJ0SMPCR	0x00000000	Not use buffer amp filter
ADCJ0TDCR	0x00	Not perform pin level self-diagnosis
ADCJ0ODCR	0x0000	Not perform disconnection detection
ADCJ0TOCCR	0x00	Not perform trigger overlap check.
ADCJ0GTMENTSGER	0x0000	Not GTM entry
ADCJ0ADENDP0	0x00	Not use A/D conversion monitor
ADCJ0WAITTR0	0x0000	Unuse since WTTS[3:0]=0
ADCJ0THACR	0x00	Not use T&H
ADCJ0PWDCR	0x00	Disable PWM-Diag
ADCJ0SGVCPR0	0x0200	End virtual channel: 2 Start virtual channel: 0
ADCJ0SGCR0	0x50	ADTSTART enable Multicycle scan mode Output INTADCJ0I0 when SG0 end Disable H/W trigger input to SG0
ADCJ0SGMCYCR0	0x00	Scan 1 time in multicycle scan mode

Register Name	Setting Value	Function
ADCJ0VCR0	0x00000000	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion Not use multiplexer Not output VCRj end interrupt Physical CH0 (AN000)
ADCJ0VCR1	0x00000001	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion Not use multiplexer Not output VCRj end interrupt Physical CH1 (AN001)
ADCJ0VCR2	0x00000002	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion Not use multiplexer Not output VCRj end interrupt Physical CH2 (AN002)

Table 3.7.3 sDMAC Register Setting

Register Name	Setting Value	Function
DMA0CM_0	0x00001C00	Channel master SPID setting SPID=0x1C (initial value) Supervisor mode
DMA0SAR_0	Written value on right	Transfer source: ADCJ0DR0 register
DMA0DAR_0	Written value on right	Transfer destination: Array variable for A/D conversion value data storing
DMA0TSR_0	0x00000006	Transfer size: 6 bytes (2 bytes × 3 times)
DMA0TMR_0	0x00001111	DMA transfer mode: normal mode Channel priority DMA transfer request selection interrupt: hardware DMA transfer request Destination address/count direction: fixed Source address/count destination: increment DMA transfer transaction size: 4 bytes DMA source transaction size: 4 bytes

Register Name	Setting Value	Function
DMA0RS_0	0x00030000	Transfers per hardware request: 3 times Transfer upper limit per hardware request Not use since PLE=0 Disable preload Disable DRQ initialization Hardware DMA transfer source selection: (A/D conversion completion interrupt (INTADCJ0I0))
DMA0SIAI_0	Written value on right	Inner address increment value: smp_data_1[0] address - smp_data_0[0] address
DMA0SGCR_0	0x80020000	Scatter function: enable Disable Zero fill Inner loop repetition times: 2 times Gather function: disable Unuse since GEN=0
DMA0CHFCR_0	0x0000320F	Clear each flag
DMA0OR	0x0001	Priority: CH0 > CH1 > ... > CH14 > CH15 DMA transfer enable
DMA0CHCR_0	0x0003	Disable descriptor Disable descriptor Disable address error notification Disable channel address error notification Disable descriptor step end interrupt Enable transfer completion interrupt Enable channel operation

Table 3.7.4 Interrupt Register Setting

Register Name	Setting Value	Function
EIBD47	0x00000000	Bind interrupt to PE0
EIC47	0x0040	Table reference/priority level 0

• Operation Flow

The flowchart of the operation example in this operation example is shown below.

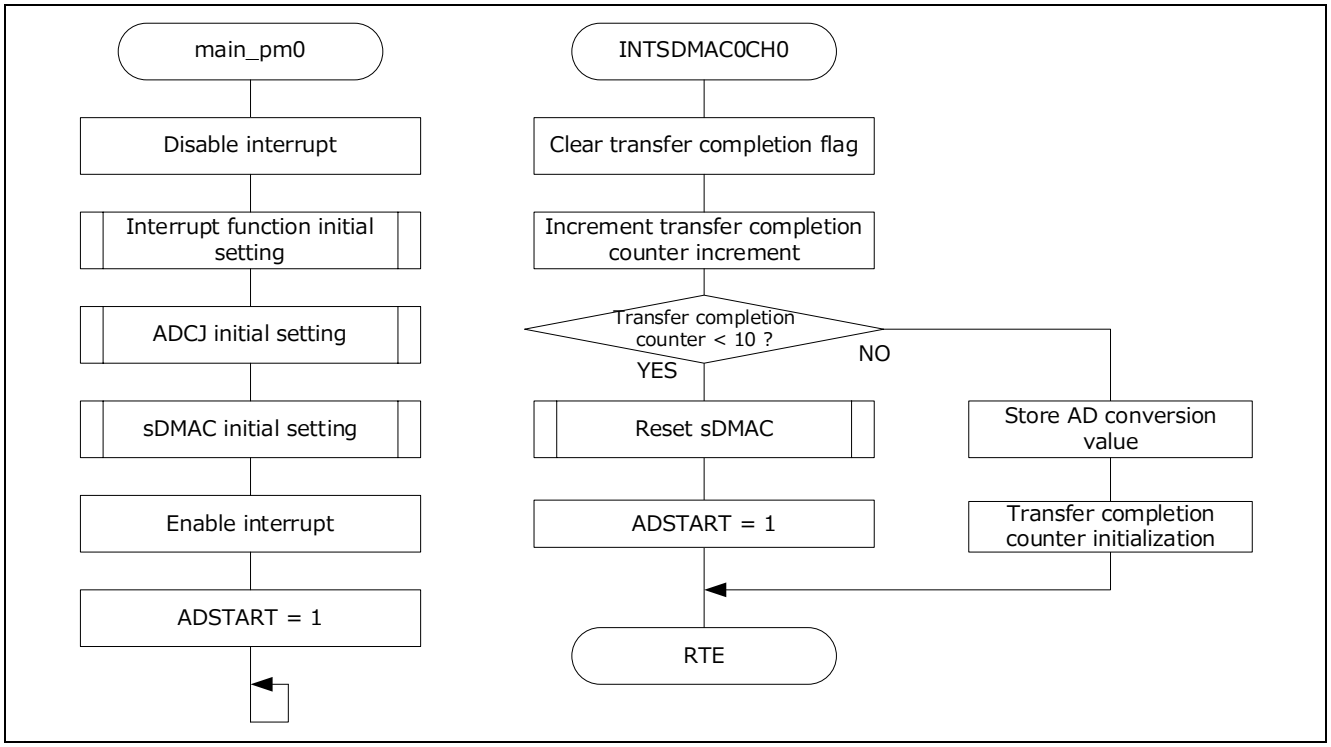

Figure 3.7.5 Flowchart

3.8 Sequential Scan Conversion of Arbitrary Channel using DMA Transfer (Gather Function)

3.8.1 Specification Overview

This section explains the A/D conversion value storing method to the array variable using DMA transfer (gather function).

Allocate the 3 virtual channels (AN000, AN001, AN002) to the scan group 0 (SG0), and perform the scan in the multicycle scan mode. Start DMA (sDMAC) by the scan end interrupt (INTADCJ0I0), and store AN000 and AN002 conversion value to the array variable.

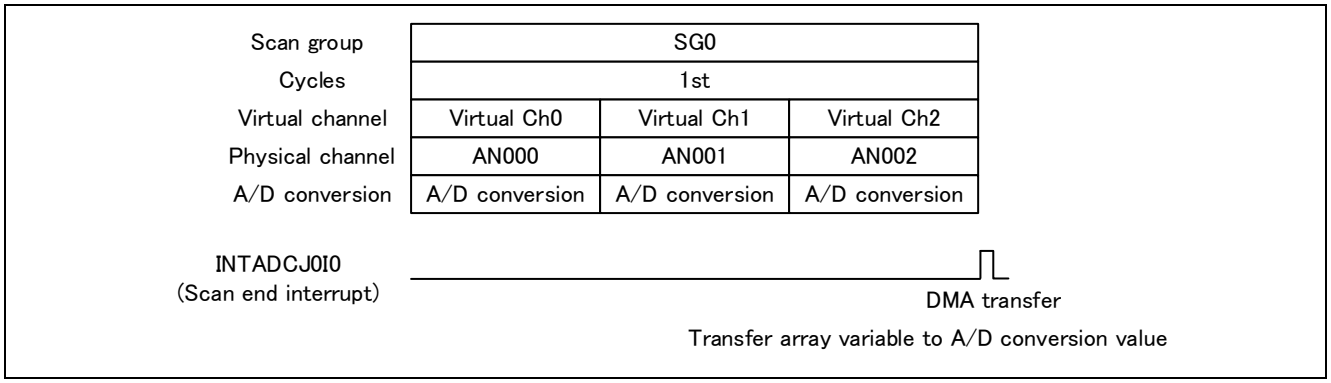

Figure 3.8.1 Operation Example using DMA Transfer

3.8.2 Use Function

The used functions in this operation are shown below.

- A/D convertor (ADCJ0)
- DMA（sDMAC）

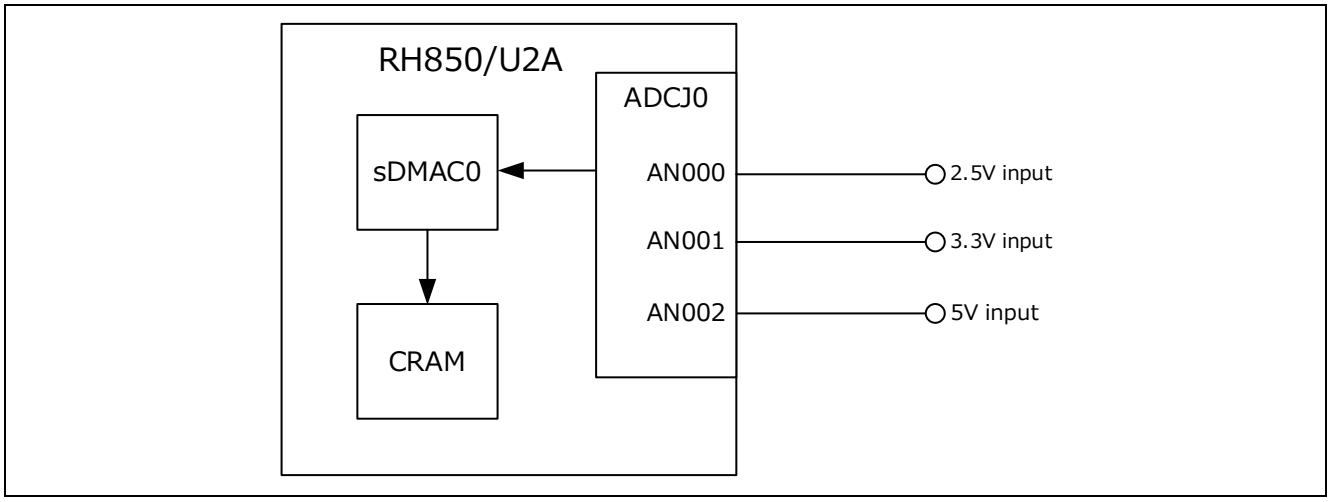

Figure 3.8.2 System Configuration

3.8.3 Explanation of Operation Example

In this operation example, perform the normal A/D conversion in the multicycle scan mode using AN000, AN001, and AN002 of ADCJ0 module, and store the A/D conversion value of AN000 and AN002 to the array variable using the DMA conversion (gather function).

Allocate the virtual channel 0 (AN000), virtual channel 1 (AN001), and virtual channel 2 (AN002) to the scan group (SG0). For analog signal, input 2.5V to AN000, 3.3V to AN001, and 5V to AN002.

As the start trigger of sDMAC, it is used the scan end interrupt (INTADCJ0I0), and set enable to the gather function. Figure 3.8.3 shows the scatter function overview. In the gather function, the A/D conversion value of the 2 channels is possible to transfer to each array variable by 1 channel of sDMAC.

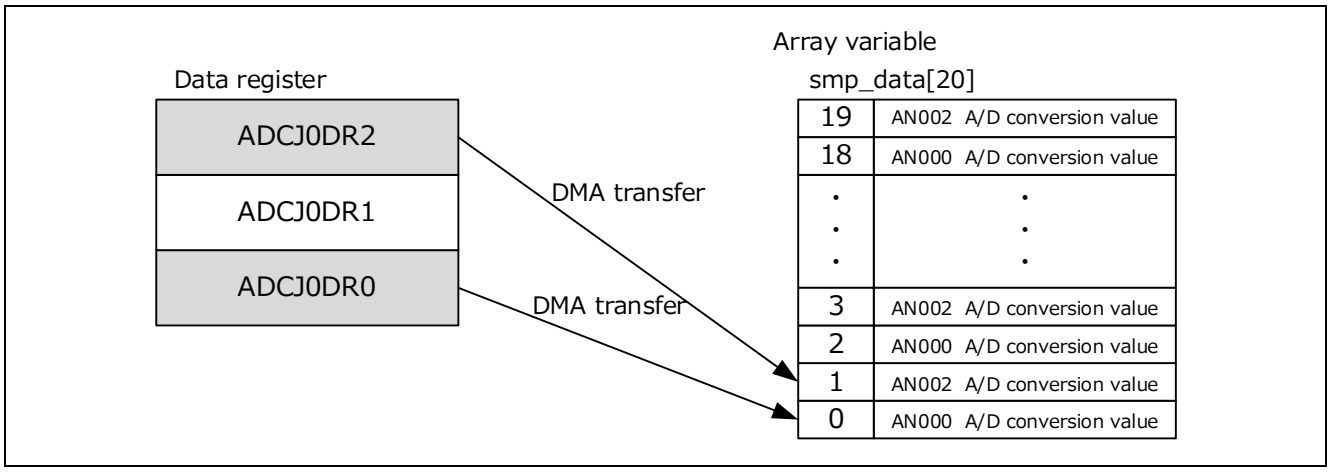

Figure 3.8.3 Overview of Gather Function

Start A/D conversion by the software trigger ADSTART. Started DMA by the scan end interrupt (INTADCJ0I0), and store the A/D conversion value to the array variable.

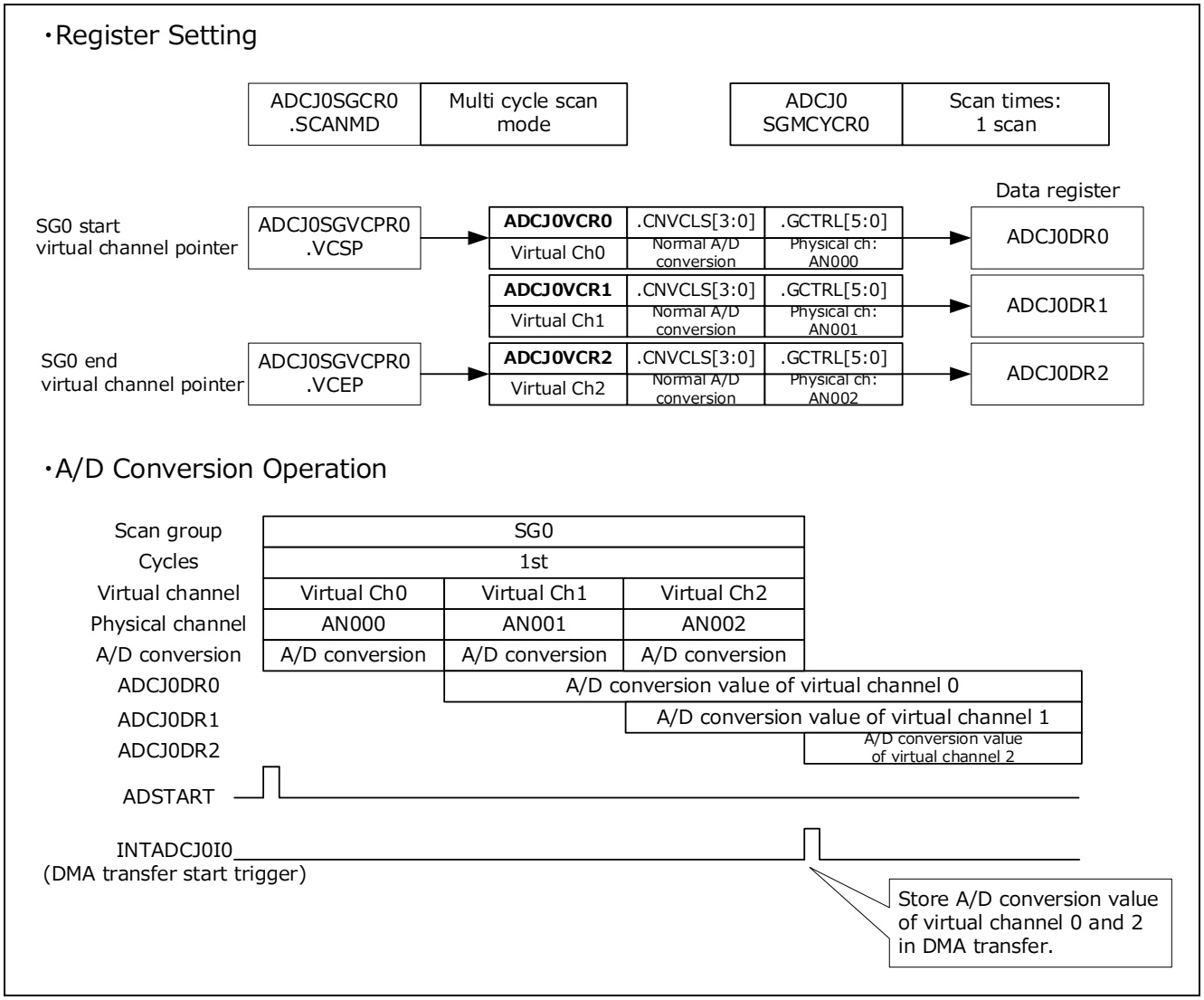

Figure 3.8.4 Operation Example using DMA Transfer

3.8.4 Software Explanation

• Module Explanation

The module list in this operation example is shows below.

Table 3.8.1 Module List

Module Name	Label Name	Fucntion
Maine routine	main_pm0	Perform various setting and application start.
ADCJ initialize routine	ADCJ_init	Perform ADCJ initialization.
sDMAC initialize routine	sdmac_init	Perform sDMAC initialization.
sDMAC transfer completion interrupt processing routine	INTSDMAC0CH0	Perform A/D conversion restart by sDMAC transfer completion interrupt processing.

・Register Setting

The Register setting of each function in this operation example is shown below.

Table 3.8.2 ADCJ Register Setting

Register Name	Setting Value	Function
ADCJ0ADCR1	0x00	Not Synchronous suspend
ADCJ0ADCR2	0x10	Signed integer format Disable since CNVCL = 0
ADCJ0SFTCR	0x00	Disable variable setting
ADCJ0VCLMINTERx	0x00000000	Not use upper/lower limit check interrupt table
ADCJ0VMONVDCR1	0x00	Not use divide voltage resistance
ADCJ0VMONVDCR2	0x00	Not use divide voltage resistance
ADCJ0SMPCR	0x00000000	Not use buffer amp filter
ADCJ0TDCR	0x00	Not perform pin level self-diagnosis
ADCJ0ODCR	0x0000	Not perform disconnection detection
ADCJ0TOCCR	0x00	Not perform trigger overlap check.
ADCJ0GTMENTSGER	0x0000	Not GTM entry
ADCJ0ADENDP0	0x00	Not use A/D conversion monitor
ADCJ0WAITTR0	0x0000	Unuse since WTTS[3:0]=0
ADCJ0THACR	0x00	Not use T&H
ADCJ0PWDCR	0x00	Disable PWM-Diag
ADCJ0SGVCPR0	0x0200	End virtual channel: 2 Start virtual channel: 0
ADCJ0SGCR0	0x50	ADTSTART enable Multicycle scan mode Output INTADCJ0I0 when SG0 end Disable H/W trigger input to SG0
ADCJ0SGMCYCR0	0x00	Scan 1 time in multicycle scan mode

Register Name	Setting Value	Function
ADCJ0VCR0	0x00000000	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion Not use multiplexer Not output VCRj end interrupt Physical CH0 (AN000)
ADCJ0VCR1	0x00000001	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion Not use multiplexer Not output VCRj end interrupt Physical CH1 (AN001)
ADCJ0VCR2	0x00000002	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion Not use multiplexer Not output VCRj end interrupt Physical CH2 (AN002)

Table 3.8.3 sDMAC Register Setting

Register Name	Setting Value	Function
DMA0CM_0	0x00001C00	Channel master SPID setting SPID=0x1C (initial value) Supervisor mode
DMA0SAR_0	Written value on right	Transfer source: ADCJ0DR0 register
DMA0DAR_0	Written value on right	Transfer destination: Array variable for A/D conversion value data storing
DMA0TSR_0	0x00000004	Transfer size: 4 bytes (2 bytes × 2 times)
DMA0TMR_0	0x00001411	DMA transfer mode: normal mode Channel priority DMA transfer request selection interrupt: hardware DMA transfer request Destination address/count direction: fixed Source address/count destination: increment DMA transfer transaction size: 2 bytes DMA source transaction size: 2 bytes

Register Name	Setting Value	Function
DMA0RS_0	0x00020000	Transfers per hardware request: 3 times Transfer upper limit per hardware request Not use since PLE=0 Disable preload Disable DRQ initialization Hardware DMA transfer source selection: (A/D conversion completion interrupt (INTADCJ0I0))
DMA0GIAI_0	Written value on right	Inner address increment value: Address of ADCJ0DR2 register - ADCJDR0 address
DMA0SGCR_0	0x00008001	Scatter function: enable Disable Zero fill Unuse since SEN=0 Gather function: enable Inner loop repetition times: 1 time
DMA0CHFCR_0	0x0000320F	Clear each flag
DMA0OR	0x0001	Priority: CH0 > CH1 > ... > CH14 > CH15 Enable DMA transfer
DMA0CHCR_0	0x0003	Disable descriptor Disable descriptor Disable address error notification Disable channel address error notification Disable descriptor step end interrupt Enable transfer completion interrupt Enable channel operation

Table 3.8.4 Interrupt Register Setting

Register Name	Setting Value	Function
EIBD47	0x00000000	Bind interrupt to PE0
EIC47	0x0040	Table reference/priority level 0

• Operation Flow

The flowchart of the operation example in this operation example is shown below.

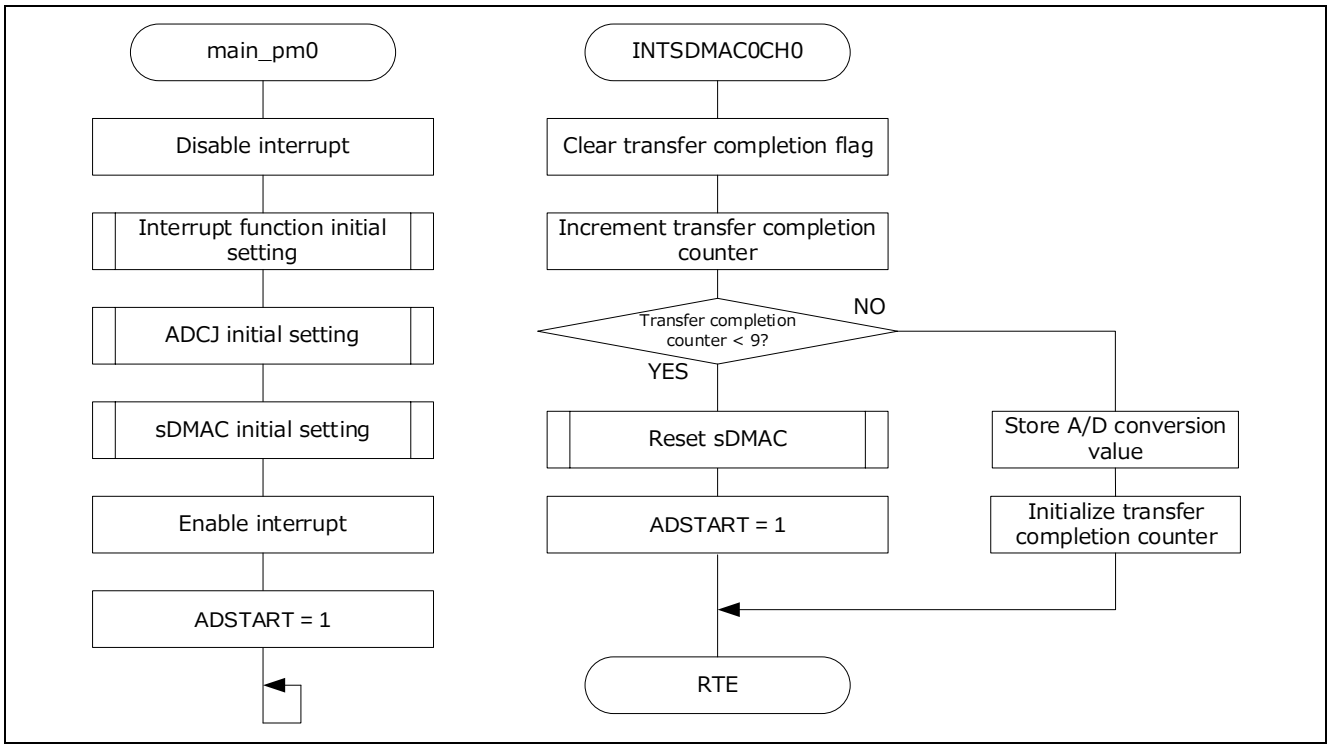

Figure 3.8.5 Flowchart

3.9 Wait Time Insert Function between Virtual Channels

3.9.1 Specification Overview

This section explains for the wait time insert function between virtual channels.

Allocate 2 virtual channels (AN000, AN001) to the scan group 0 (SG0), scan 1 time in the multicycle scan mode. Store the conversion value of AN000 and AN001 to the variable and end the operation. In this time, for the normal A/D conversion, the wait time can be inserted between the virtual channels

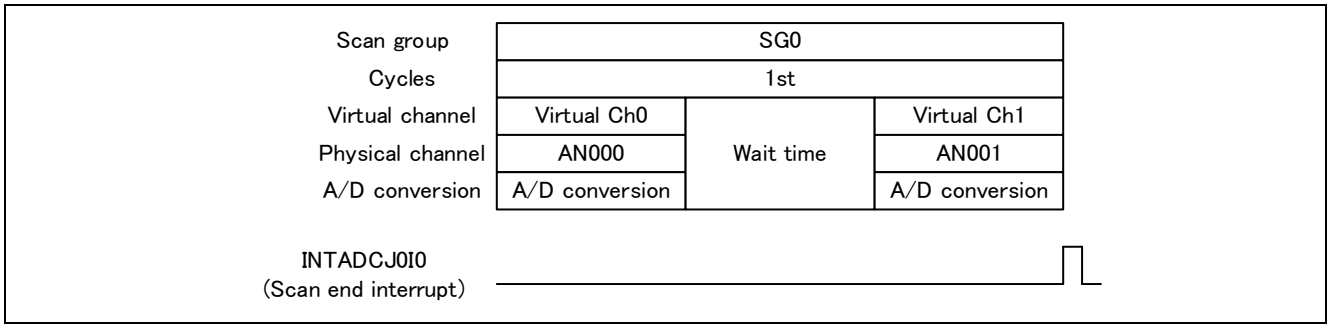

Figure 3.9.1 Normal A/D Conversion (Multicycle scan) Operation

3.9.2 Use Function

The used hardware functions in this operation example are shown below.

- A/D convertor (ADCJ0)
- Port

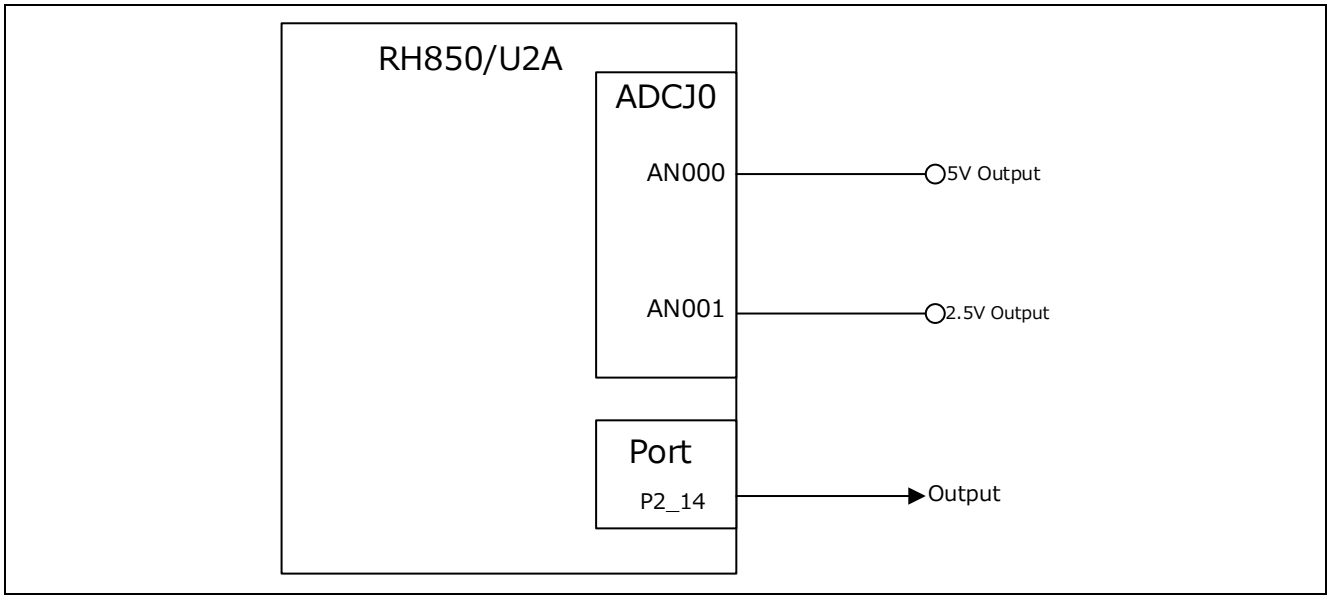

Figure 3.9.2 System Configuration

3.9.3 Explanation of Operation Example

In this operation example, perform the normal A/D conversion by the 1 scan of the multicycle scan mode that uses the AN000 and AN001 of the ADCJ0 module

Allocate the virtual channel 0 (AN000) and virtual channel 1 (AN001) to the scan group 0 (SG0).

The analog signal inputs 5.0V to AN000 and 2.5V to AN001.

Set the wait time between the virtual channels as 400us to ADCJ0WAITTR0, and set ADCJ0VCR1.WTTS[3:0] to select the ADCJ0WAITTR0.

Start by the software trigger SGST, and A/D convert AN001 after the AN000 A/D conversion.

Enable the scan end interrupt INTADCJ0I0 (when scan group end), store each A/D conversion result to the variable in the interrupt processing, and end the operation.

Also, in this operation example, use the port (P2_14) as the output port for checking that the wait time is inserted between the virtual channels.

Set high to P2_14 output when starting the A/D conversion, and set the output to low in the scan end interrupt INTADCJ0I0. It is possible to check that the wait time is inserted between the virtual channels by the high period of P2_14.

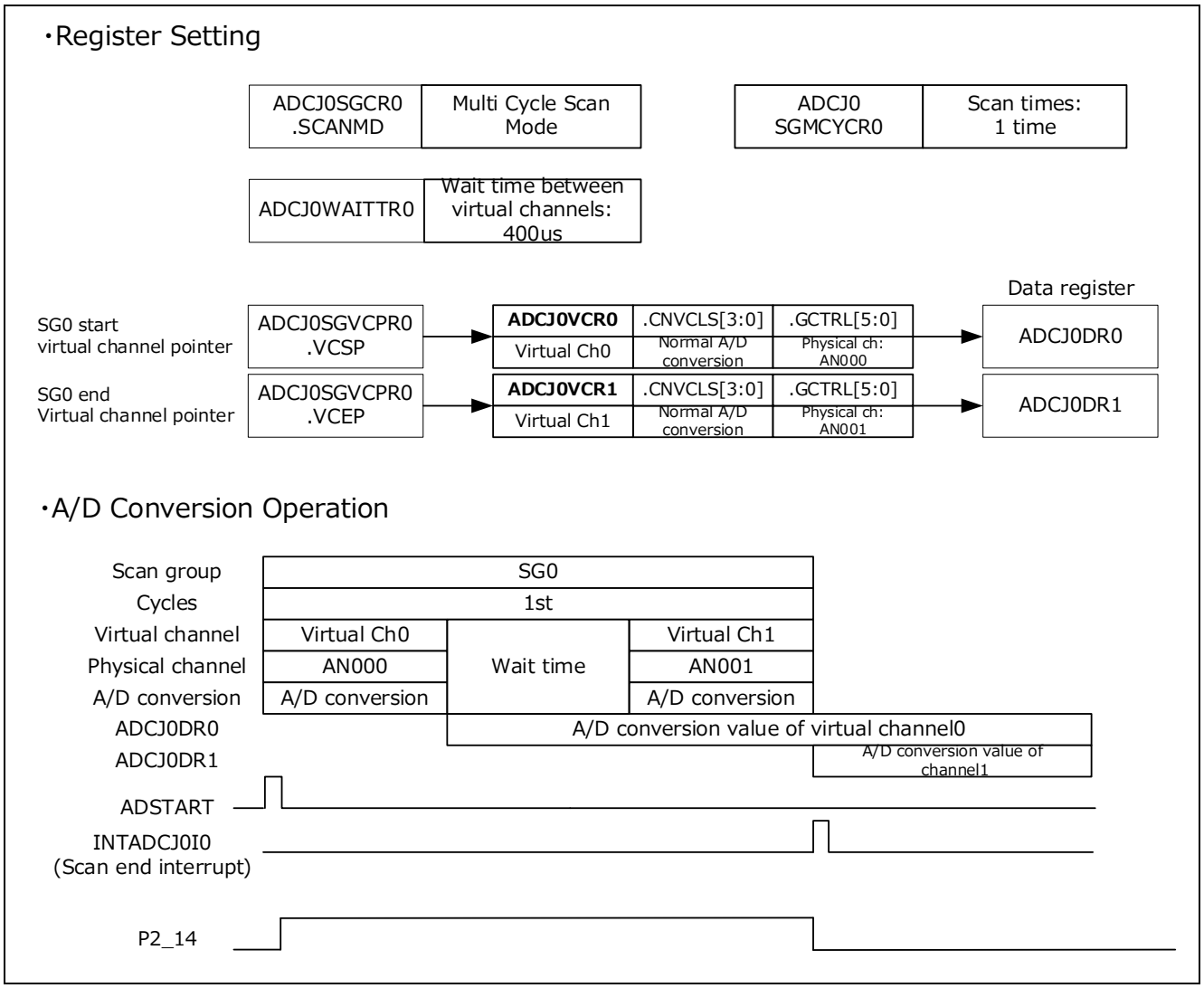

Figure 3.9.3 Normal A/D Conversion (Wait Insert between Virtual Channels) Operation Example

3.9.4 Software Explanation

• Module Explanation

The module list in this operation example is shows below.

Table 3.9.1 Module List

Module Name	Label Name	Function
Maine routine	main_pm0	Perform various setting and application start.
ADCJ initialize routine	ADCJ_init	Perform ADCJ initialization.
ADCJ interrupt processing routine	INTADCJ0I0	Store A/D conversion result to variable in virtual scan group end interrupt
Interrupt initialize routine	intc2_init	Perform ADCJ interrupt initialization.

・Register Setting

The Register setting of each function in this operation example is shown below.

Table 3.9.2 ADCJ Register Setting

Register Name	Setting Value	Function
ADCJ0ADCR1	0x00	Synchronous suspend
ADCJ0ADCR2	0x10	Signed integer format Disable since CNVCL = 0
ADCJ0SFTCR	0x00	Disable variable setting
ADCJ0VCLMINTERx	0x00000000	Not use upper/lower limit check interrupt table
ADCJ0VMONVDCR1	0x00	Not use divide voltage resistance
ADCJ0VMONVDCR2	0x00	Not use divide voltage resistance
ADCJ0SMPCR	0x00000000	Not use buffer amp filter
ADCJ0TDCR	0x00	Not perform pin level self-diagnosis
ADCJ0ODCR	0x0000	Not perform disconnection detection
ADCJ0TOCCR	0x00	Not perform trigger overlap check.
ADCJ0GTMENTSGER	0x0000	Not GTM entry
ADCJ0ADENDP0	0x00	Not use A/D conversion monitor
ADCJ0THACR	0x00	Disable hold control Disable hold trigger Disable T&H
ADCJ0PWDCR	0x00	Disable PWM-Diag
ADCJ0SGVCPR0	0x0100	End virtual channel: 1 Start virtual channel: 0
ADCJ0SGCR0	0x50	Enable ADSTART Multicycle scan mode Output INTADCJ0I0 when SG0 end Disable H/W trigger input to SG0
ADCJ0SGMCYCR0	0x00	Scan 1 time in multicycle scan mode

Register Name	Setting Value	Function
ADCJ0VCR0	0x00000000	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion Not use multiplexer Not output VCRj end interrupt Physical CH0 (AN000)
ADCJ0VCR1	0x01000001	Not use upper/lower limit check Not use wait time table Not use GTM entry Not use since GTMENT=0 Conversion type: normal A/D conversion Not use multiplexer Not output VCRj end interrupt Physical CH1（AN001）
ADCJ0WAITTR0	0x1F40	Waite time: 400us

Table 3.9.3 Interrupt Register Setting

Register Name	Setting Value	Function
EIBD227	0x00000000	Bind interrupt to PE0
EIC227	0x0040	Table reference/priority level 0

• Operation Flow

The flowchart of the operation example in this operation example is shown below.

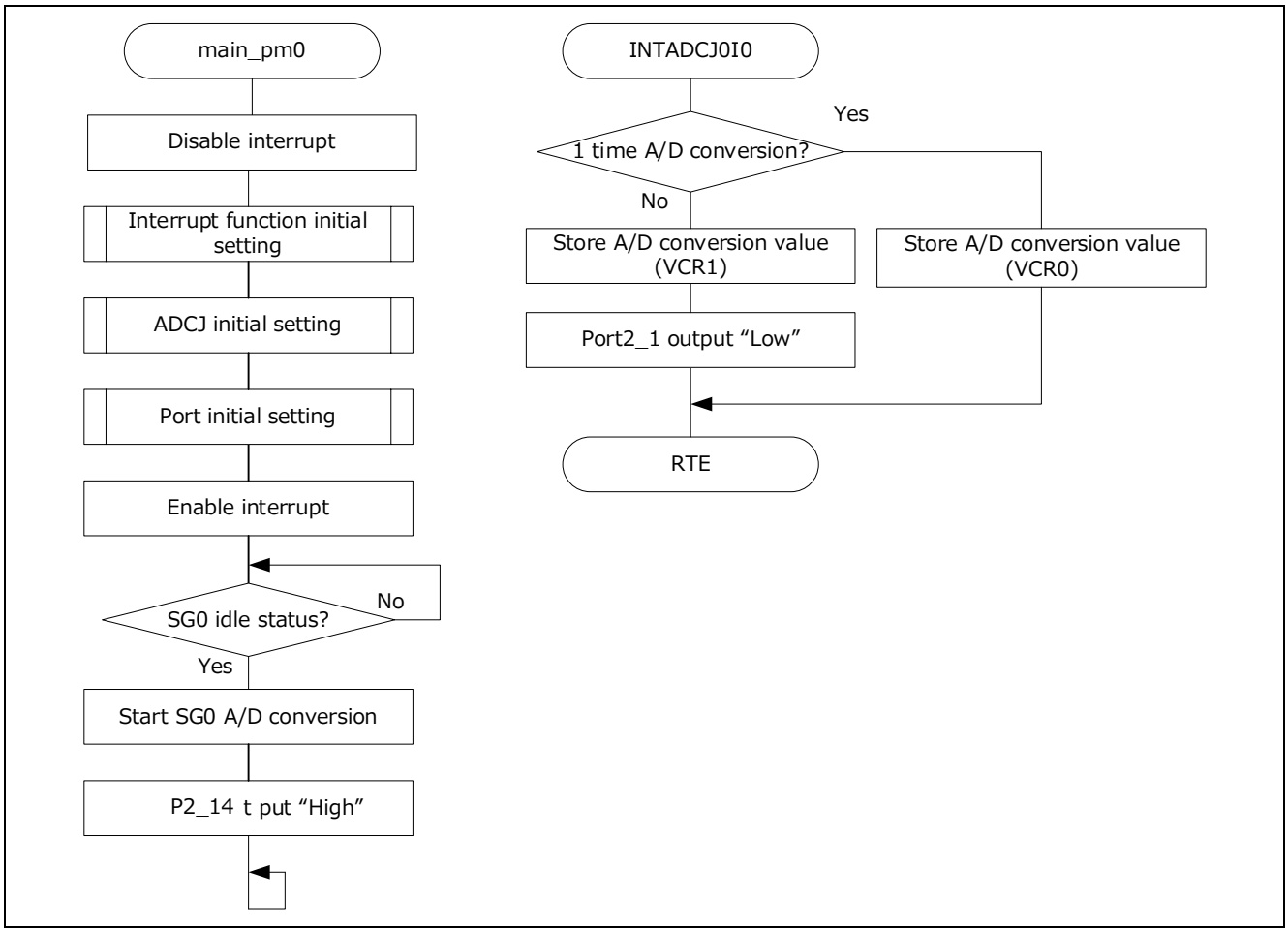

Figure 3.9.4 Flowchart

Revision History

Rev.	Date	Description	
		Page	Summary
1.1	2023.01.13	-	Released English version of *r01an4754jj0110*

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/.