RH850/U2B Group

R01AN6563EJ0100
Rev.1.00

ADCK Application Note

Summary

This application note describes about the operation example using sequential comparison method 12bit A/D converter (hereinafter, ADCK) of RH850/U2Bx.

The operation examples described in this application note have been confirmed to operate, be sure to confirm the operation before using them.

Contents

1. Introduction

This application note shows the usage of sequential comparison method 12bit A/D converter of RH850/U2Bx and the software create example.

1.1 Use Functions

The hardware function of RH850/U2Bx used in this application note is shown below.

- Sequential comparison method 12bit A/D converter (ADCK）
- ASF (ADC accumulation function)
- ADC VMON Secondary Error Generator (AVSEG)
- ADC Boundary Flag Generator (ABFG)
- General-purpose pin (PORT)
- DMA (DTS)
- DMA (sDMAC)

2. ADCK Overview

2.1 ADCK Operation

2.1.1 Virtual Channel and Scan Group

The RH850/U2Bx is equipped with a maximum of 5 modules (ADCK0 to 4) of 12bit A/D converters that is the sequential comparison method, and can perform A/D conversion for a total of 96 channels. Each module has 64 channels of virtual channels, and each virtual channel is set with accompanying information such as analog channels for A / D conversion, conversion mode, and interrupts. In addition, ADCK 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 allocation example of the physical channel, virtual channel, and scan group are shown below.

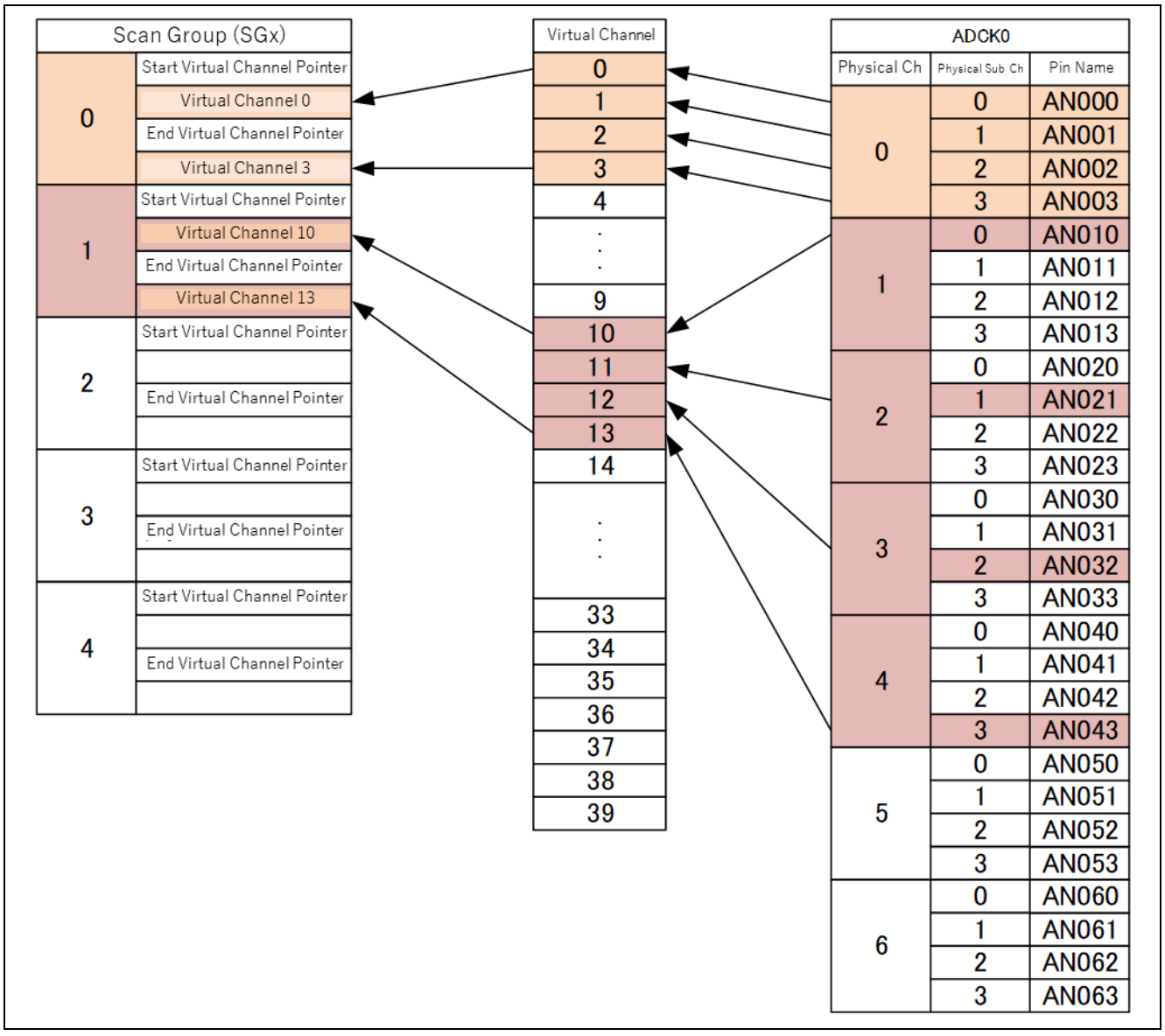

Figure 2-1 Allocation of Virtual Channel and Scan Group (ADCK0)

2.1.2 Scan End Interrupt Request

The scan group (SGx) can generate the scan end interrupt request (ADImx) to the INTC. There are two output factors for this this scan interrupt request, which “generate at the end of the scan group (SGX)” or “generate at the end of the A/D conversion of the virtual channel n allocated to the scan group (SGx)”.

The enable/disable setting of the scan end interrupt request “at the end of scan group (SGx) scanning” is performed by the ADIE (0 = disabled, 1 = enabled) of ADCKmSGCRx. The enable/disable setting of the scan end interrupt request “at the end of the A/D conversion for virtual channel allocated to the scan group (SGx)” is performed by the ADIE (0 = disabled, 1 = enabled) of ADCKmVCRn. The ADIE setting of ADCKmSGCRx and the ADIE setting of ADCKmVCRn are irrelevant.

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

Ex.1) ADIm0 output at the end of A/D conversion of virtual channel 0

ADCKmSGCR0.ADIE = 0 （Not output at the end of SG0 Scanning.）

ADCKmVCR0.ADIE = 1 (Output at the end of virtual channel 0 on SG0.)

ADCKmVCR1.ADIE = 0 （Not output at the end of virtual channel 0 on SG0.）

Ex.2) ADIm0 output at the end of A/D conversion of virtual channel 0 and virtual channel 1

ADCKmSGCR0.ADIE = 0 （Not output at the end of SG0 Scanning.）

ADCKmVCR0.ADIE = 1 (Output at the end of virtual channel 0 on SG0.)

ADCKmVCR1.ADIE = 1 (Output at the end of virtual channel 1 on SG0.)

Ex.3) ADIm0 output at the end of SG0 scanning

ADCKmSGCR0.ADIE = 1 (Output at the end of SG0 Scanning.）

ADCKmVCR0.ADIE = 0 （Not output at the end of virtual channel 0 on SG0.）

ADCKmVCR1.ADIE = 0 （Not output at the end of virtual channel 1 on SG0.）

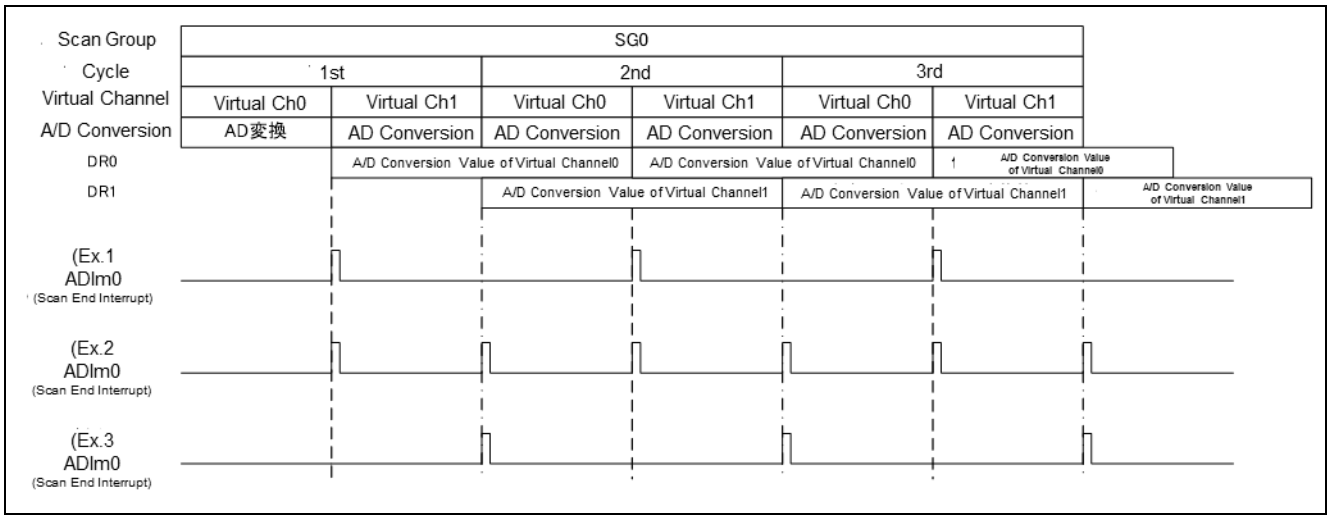

Figure 2-2 Generation Timing of Scan End Interrupt

3. Operation Overview

3.1 Sequential Scan Conversion of Arbitrary Channel

3.1.1 Specification Overview

Describes how to perform normal A/D conversion.

Assign two channels (AN000, AN 001) to the scam group 0 (SG0) and perform one scan in multi-scan mode. At the end of the scan group, the converted values of AN000 and AN001 are stored in variables, and the operation ends.

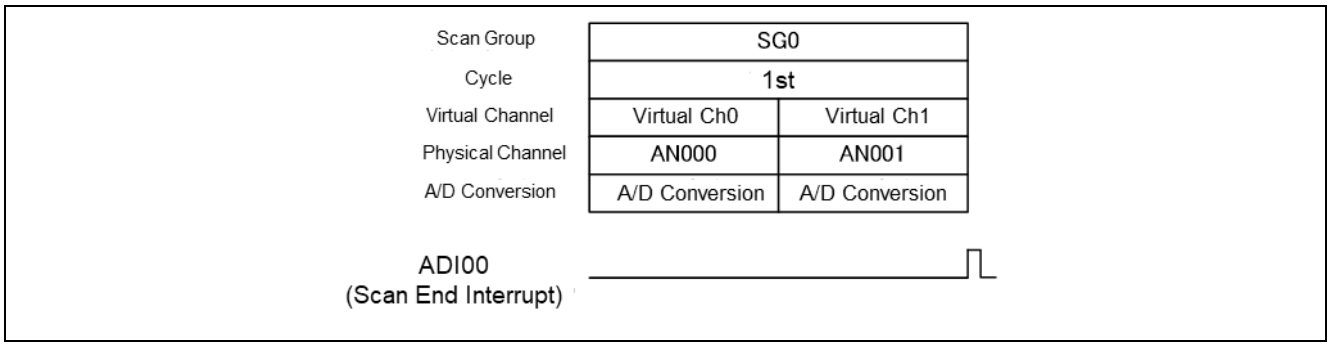

Figure 3-1 Normal A/D Conversion (Multi-scan) Operation

3.1.2 Use Function

Hardware function used in this operation example is shown below.

- A/D Converter（ADCK0）

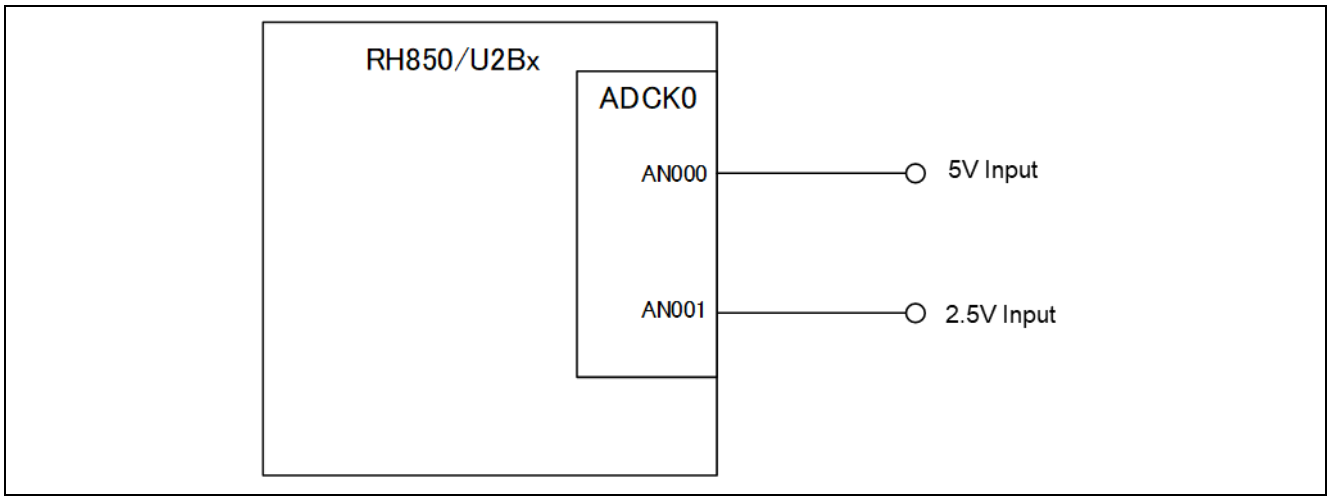

Figure 3-2 System Configuration

3.1.3 Explanation for Operation Example

In this operation example, A / D conversion is normally performed by one scan in the multi-scan mode using AN000 and AN001 of the ADCK0 module.

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

For analog signals, input 5.0V to AN000 and 2.5v to AN001.

Start the soft trigger ADSTART, and perform A/D conversion for AN001 after performed A/D conversion for AN000.

Enable the scan end interrupt ADI00 (at the end of the scan group), store each A/D conversion result to the variable in the interrupt processing, and the process ends.

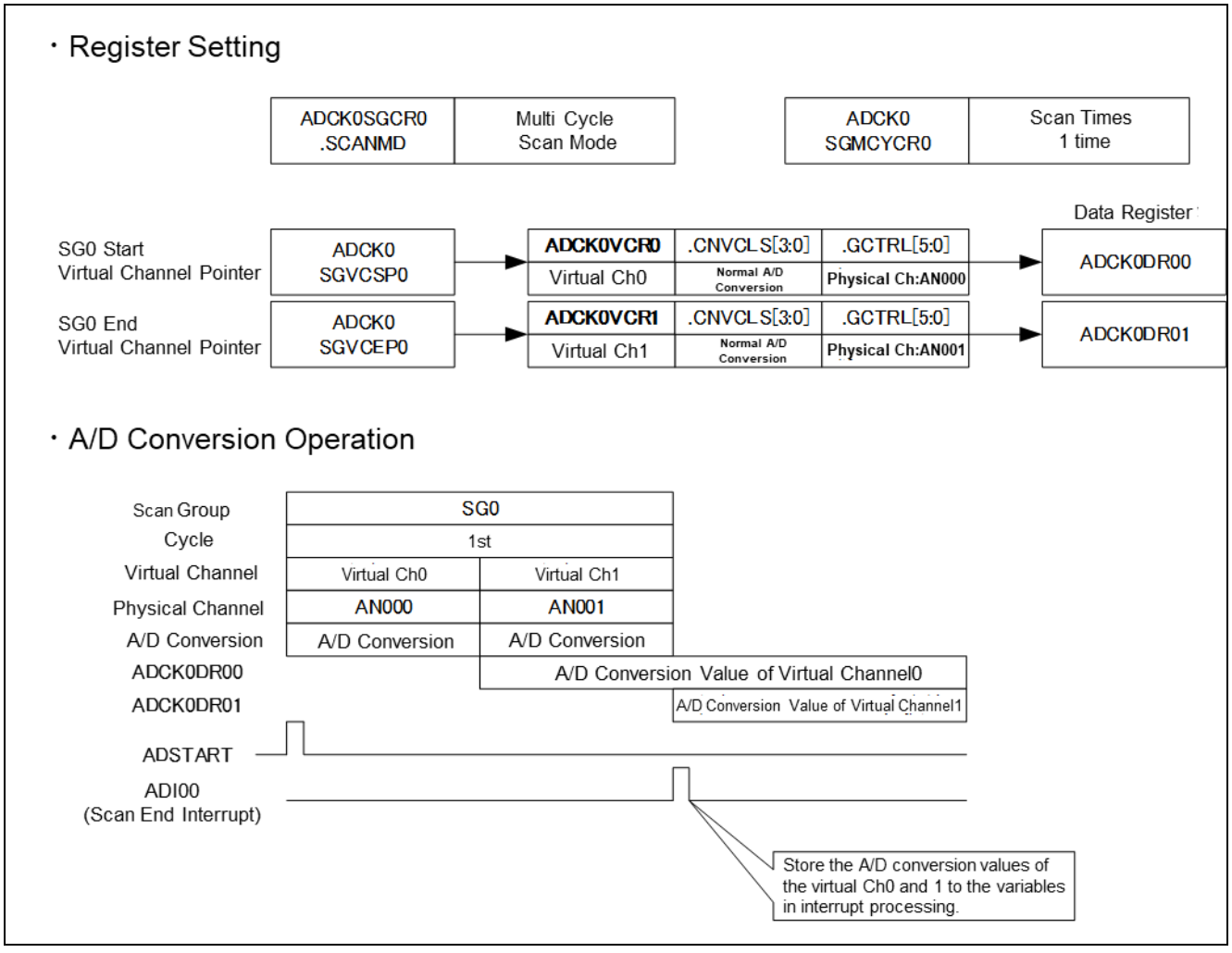

Figure 3-3 Normal A/D Conversion (Multi-scan Mode) Operation Example

3.1.4 Software Explanation

- Module Explanation

The module list in this operation example is shown below.

Table 3-1 Module List

Module Name	Label Name	Function
Main routine	main_pe0	Perform the various settings and application start.
ADCK initialize routine	ADCK_init	Initialize the ADCK.
Interrupt initialize routine	intc_init	Initialize the ADCK interrupt.
ADCK interrupt process routine	eiint441	Store the A/D conversion result in variable by the virtual scan group end interrupt processing.

- Register Setting

The register setting for various function in this operation example is shown below.

Table 3-2 ADCK Register Setting

Register Name	Setting Value	Function
ADCK0VCR00	0x00000000	Conversion type: normal A/D conversion
		Not select the wait time table.
		No DFE entry.
		Not output the virtual channel end interrupt.
		Physical Ch0/Sub CH0 (AN000)
ADCK0VCR01	0x00000001	Conversion type: normal A/D conversion
		Not select the wait time table.
		Not DFE entry.
		Not output the virtual channel end interrupt.
		Physical Ch0/Sub CH1 (AN001)
ADCK0ADCR1	0x00	Synchronous suspend
ADCK0ADCR2	0x10	Signed 12bit integer format
ADCK0SGCR0	0x50	Enable ADSTART.
		Multi-cycle scan mode
		Output “ADI00” in the end of “SG0”.
		Prohibit the trigger input to “SG0”.
ADCK0SGVCPR0	0x0100	Start virtual Ch0, End virtual Ch1
ADCK0SGMCYCR0	0x00	1 scan in multi-cycle scan mode

Table 3-3 Interrupt Register Setting

Register Name	Setting Value	Function
EIBD441	0x00000000	Bind the interrupt to PE0 (CPU0).
EIC441	0x0040	Refer to table/ Priority level 0

- Operation Flow
  The flow chart in this operation example is shown below.

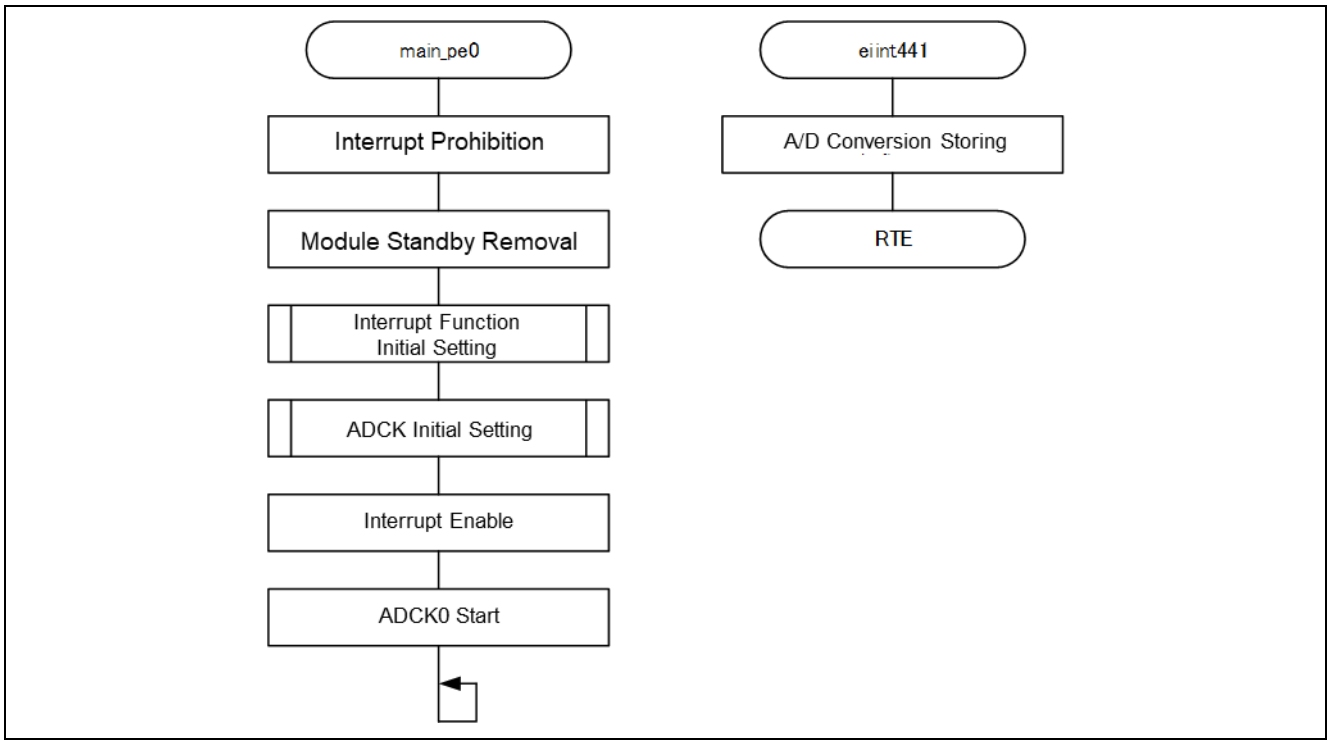

Figure 3-4 Flow Chart

3.2 Synchronous Suspend & Resume Operation

3.2.1 Specification Overview

Explain for synchronous suspend & resume operation.

Assign the three virtual channel (AN020, AN030, AN040) to scan group 0 (SG0), and one virtual channel (AN021) to scan group 1.

Starts SG0 operation and starts SG1 during SG0 A/D conversion. After the in-process SG0 virtual channel ends, SG0 operation is suspended and SG1 A/D conversion is started. After the conversion operation of SG1 is completed, it resumes from the suspended virtual channel of SG0.

The converted value is stored in the variable at the end of the SG0 scan group.

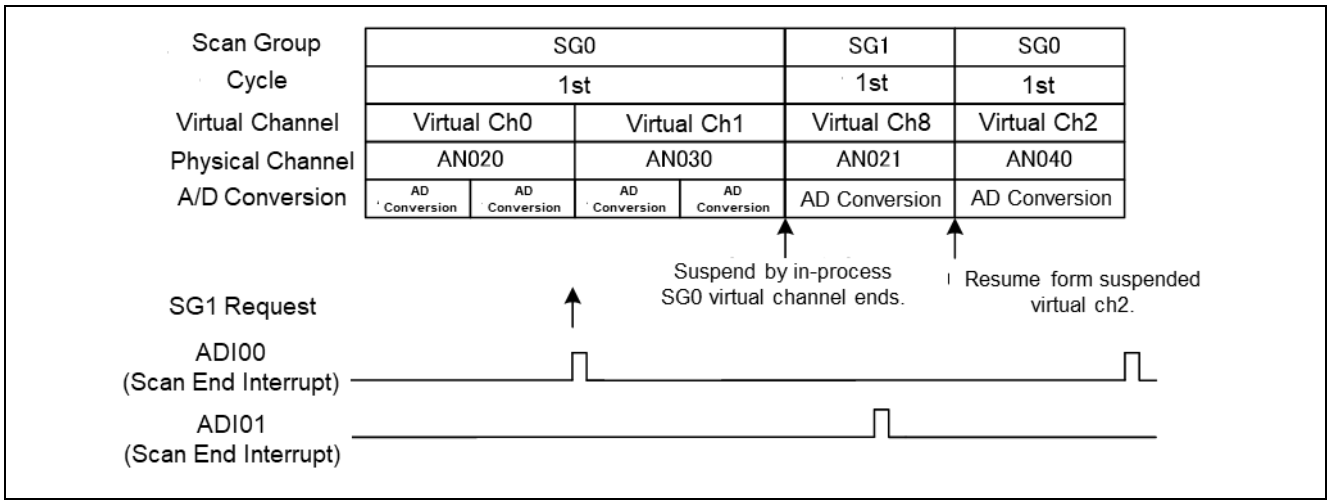

Figure 3-5 Synchronous Suspend & Resume Operation

3.2.2 Use Function

The hardware function in this operation example is shown below.

- A/D Converter (ADCK0)

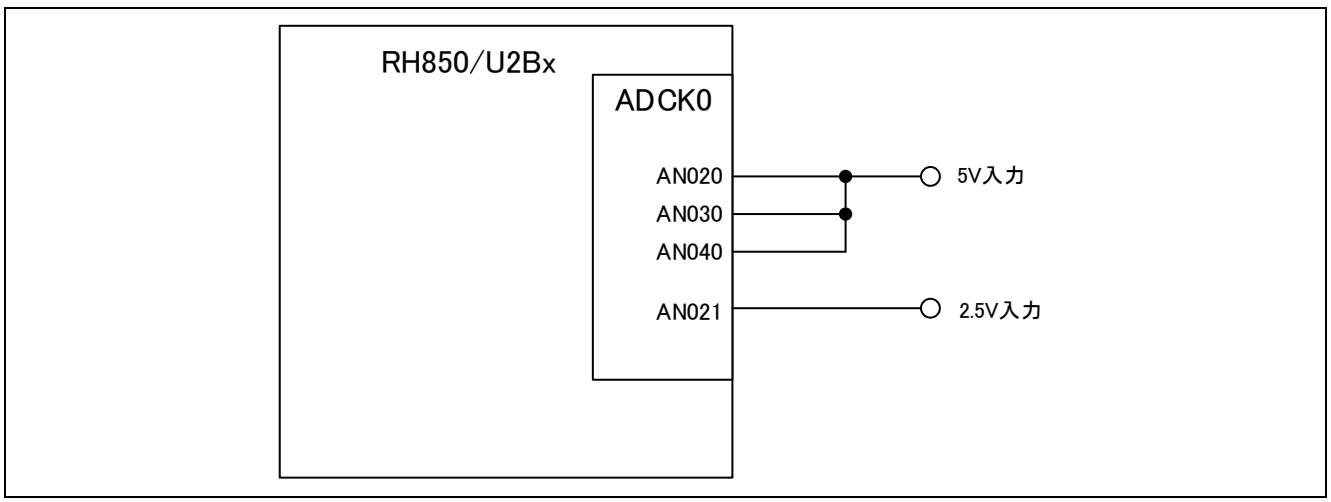

Figure 3-6 System Configuration

3.2.3 Explanation for Operation Example

In this operation example, perform the synchronous suspend and resume to use AN020, AN021, AN 030, and AN 040 of the ADCK0 module.

Allocate the virtual channel 0 (AN020), the virtual channel 1 (AN030), and the virtual channel 2 (AN040) to the scan group 0 (SG0). Also, allocate the virtual channel 8 (AN021) to the scan group (SG1).

For analog signals, input 5.0V to AN020, AN030, and AN0404. Input 2.5V to AN021.

ADI00, the scan end interrupt of the SG0 is generated in the end of the virtual channel 0 and SG0. ADI01, the scan end interrupt of the SG1 is generated in the end of the SG1.

At first, set “ADCK0SGSTCR0.SGST”, and start the SG0. ADI00 generates when the processing of virtual channel 0 of SG0 is completed, “ADCK0SGSTCR1.SGST” is set in the interrupt processing, and SG1 is started.

In synchronous suspend and resume, the processing of the high priority scan group is started after waiting for the end of the virtual channel in progress. Therefore, in this operation example, suspend the operation of SG0 and start the AD conversion of the virtual channel 8 for SG1 after the end of virtual channel 1 of SG0 in progress

After the end of SG1, resume the suspended operation of virtual channel 2 of SG0.

Each AD conversion value is stored in the variable by the interrupt processing of the scan end interrupt ADI00 that is generated at the end of the SG0.

In this operation example, look at the output of Port11_0 and Port11_1 to check the operation of synchronous sus

Port11_0 sets the output to High by the interrupt processing of ADI00 that generates at the end of virtual channel 0 of SG0, and it sets the output to Low by the interrupt processing of ADI00 that generates at the end of SG0.

Port11_1 sets the output to High by the interrupt processing of ADI01 that generates at the end of SG1.

Since Port11_1 becomes High (SG1 operation end) during Port11_0 is High (SG0 operation/interrupt), it can be confirmed that synchronous suspend and resume operation has been performed.

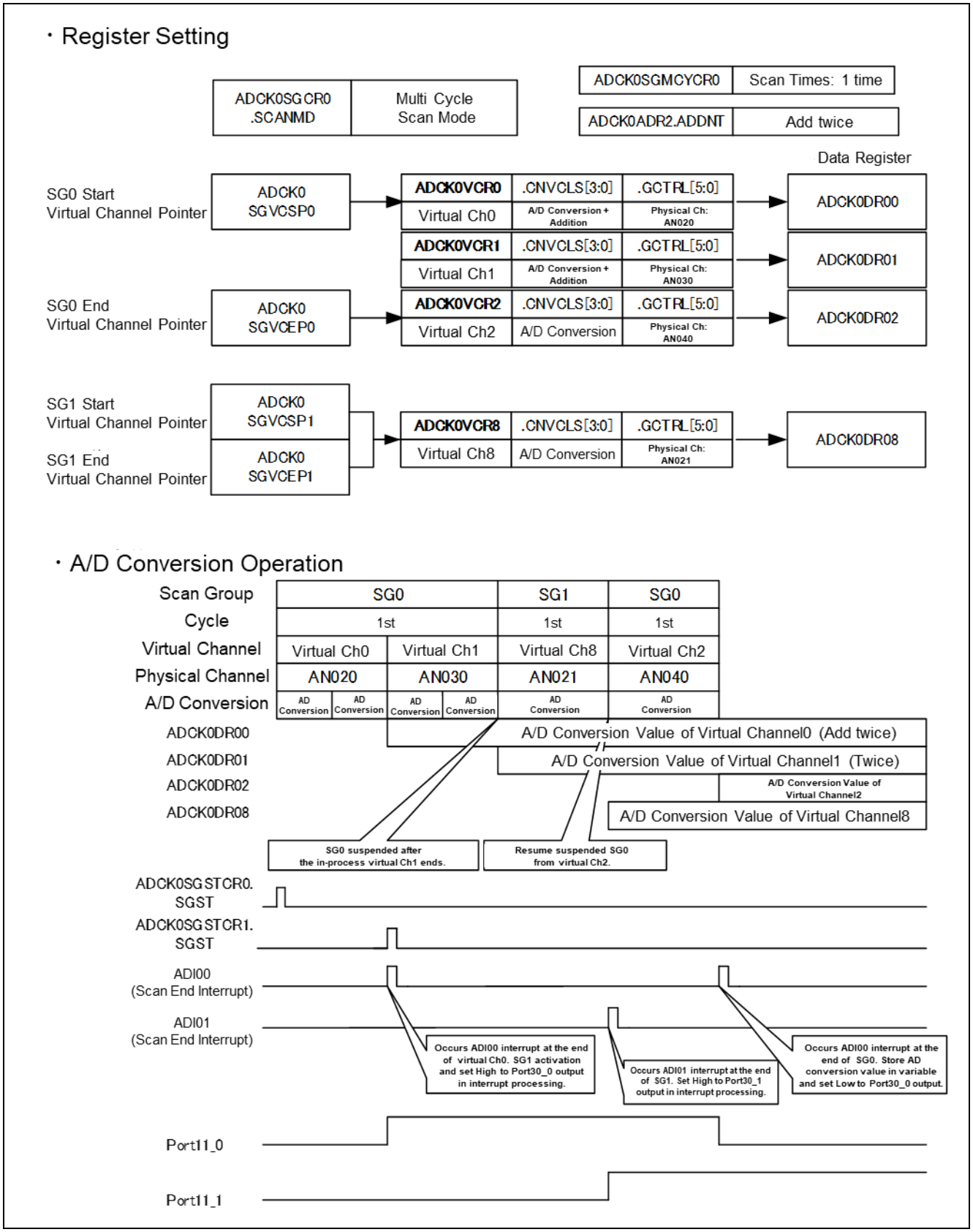

Figure 3-7 Synchronous Suspend & Resume Operation Example

3.2.4 Software Explanation

- Module Explanation

Module List in this operation example is shown below.

Table 3-4 Module List

Module Name	Label Name	Function
Main routine	main_pe0	Perform the various settings and application start.
Port initialize routine	port_init	Initialize the port.
ADCK initialize routine	ADCK_init	Initialize the ADCK.
Interrupt initialize routine	intc_init	Initialize the ADCK interrupt.
ADCK/SG0 interrupt process routine	eiint441	• Start SG1 and invert output of port 11_0 after virtual channel 0 of SG0 ends. • Store A/D conversion result in variable and invert output of port 11_0 after scan group of SG0 ends.
ADCK/SG1 interrupt process routine	eiint442	Invert output of port 11_1 by scan group and interrupt processing.

- Register Setting

The register setting for various function in this operation example is shown below.

Table 3-5 ADCK Register Setting

Register Name	Setting Value	Function
ADCK0VCR00	0x00002088	Conversion type: normal A/D conversion + Addition mode
		Not select the wait time table.
		No DFE entry.
		Output the virtual channel end interrupt
		Physical Ch2/Sub CH0 (AN020)
ADCK0VCR01	0x0000200C	Conversion type: normal A/D conversion + Addition mode
		Not select the wait time table.
		No DFE entry.
		Not output the virtual channel end interrupt.
		Physical Ch3/Sub CH0 (AN030)
ADCK0VCR02	0x00000010	Conversion type: normal A/D conversion
		Not select the wait time table.
		No DFE entry.
		Not output the virtual channel end interrupt.
		Physical Ch4/Sub CH0 (AN040)
ADCK0VCR08	0x00000009	Conversion type: normal A/D conversion
		Not select the wait time table.
		No DFE entry.
		Not output the virtual channel end interrupt.
		Physical Ch2/Sub CH1 (AN021)
ADCK0ADCR1	0x00	Synchronous suspend
ADCK0ADCR2	0x10	Signed 12bit integer format
		Add twice.

ADCK0SGCR0	0x10	Enable ADSTART.
		Multi-cycle scan mode
		Output “ADI00” in the end of “SG0”.
		Prohibit the trigger input to “SG0”.
ADCK0SGVCPR0	0x0200	SG0 start virtual channel0, SG0 end virtual channel2
ADCK0SGMCYCR0	0x00	One scan in multi-cycle scan mode.
ADCK0SGCR1	0x10	Prohibit ADSTART.
		Multi-cycle scan mode
		Output “ADI00” in the end of “SG0”.
		Prohibit trigger input to SG1.
ADCK0SGVCPR1	0x0808	SG1 start virtual channel8, SG1 end virtual channel8
ADCK0SGMCYCR1	0x00	One scan in multi-cycle scan mode.

Table 3-6 Interrupt Register Setting

Register Name	Setting Value	Function
EIBD441	0x00000000	Bind the interrupt to PE0 (CPU0).
EIC441	0x004F	Refer to table/ Priority level 15
EIBD442	0x00000000	Bind the interrupt to PE0 (CPU0).
EIC442	0x004F	Refer to table/ Priority level 15

- Operation Flow

Flow chart in this operation example is shown below.

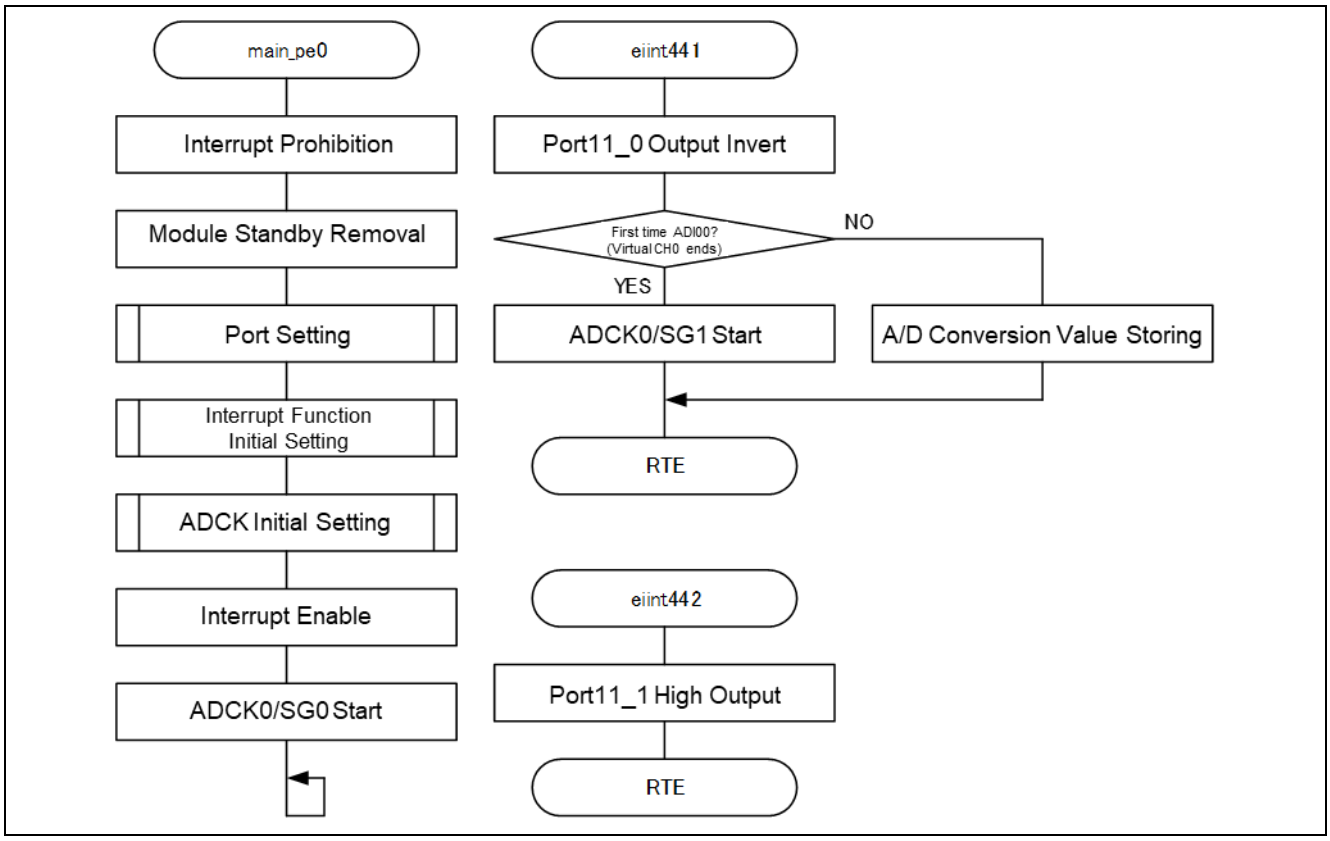

Figure 3-8 Flow Chart

3.3 Interval Conversion using AD Timer

3.3.1 Specification Overview

Explains how to perform A/D conversion at arbitrary cycle using AD timer.

Allocate a virtual channel (AN000) to scan group3 (SG3), and set the trigger for starting A/D conversion to the AD timer. The initial phase of the AD timer is 0.25us, the timer cycle is 100µs, SG3 is started every cycle. One scan is performed in the multi-scan mode. Store AN000 conversion value in the variable at the end of the scan group.

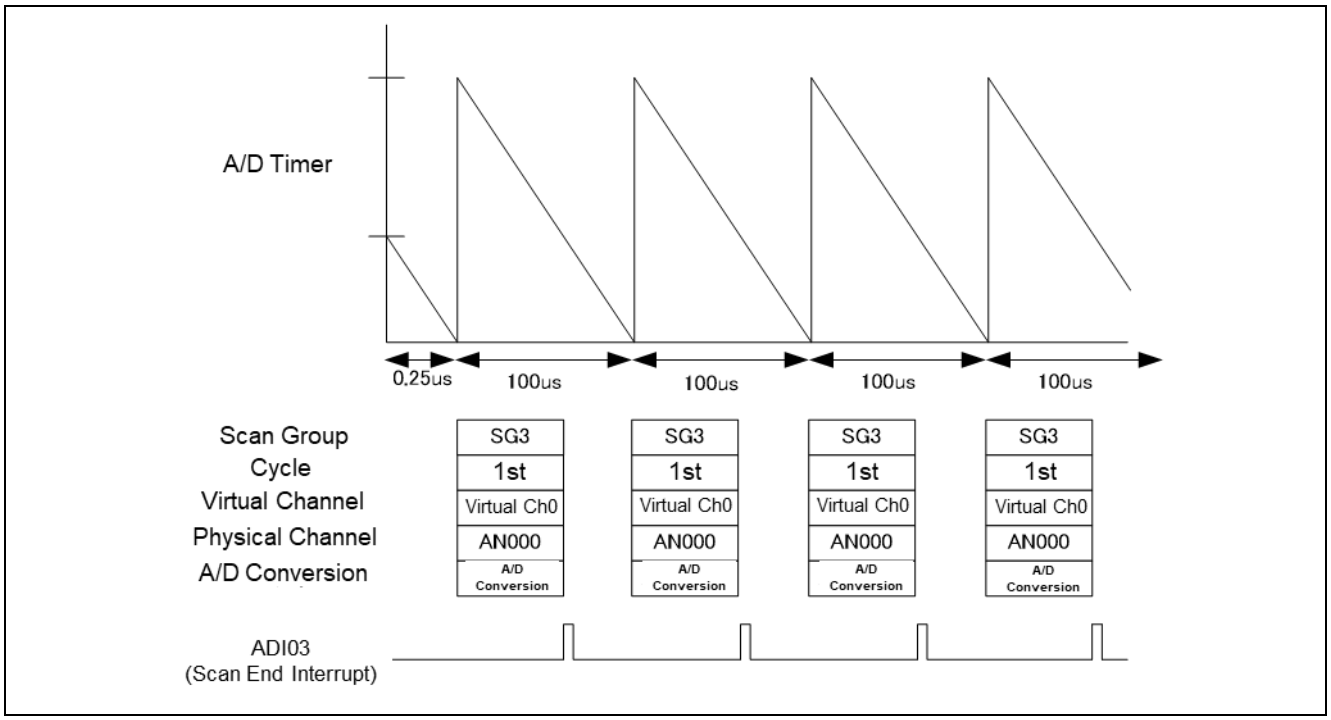

Figure 3-9 A/D Conversion Operation by AD Timer Trigger AD

3.3.2 Use Function

The hardware function used in this operation example is shown below. Input 100Hz Sin wave (amplitude 0 to 5V) to AN000 with the pulse generator, etc.

- A/D Converter（ADCK0）

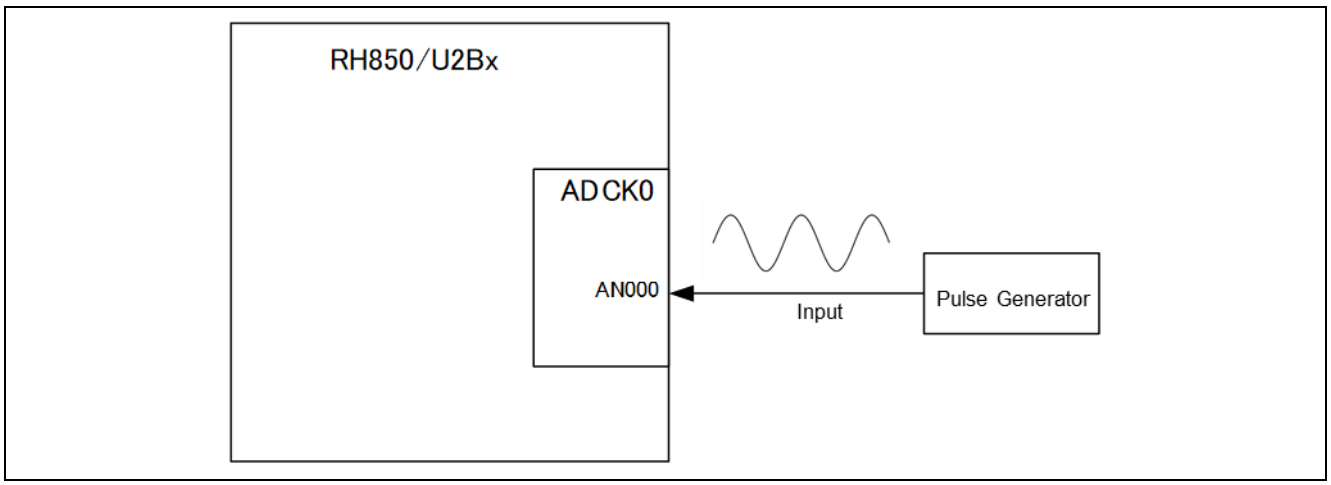

Figure 3-10 System Configuration Diagram

3.3.3 Explanation for Operation Example

In this operation example, AN000 of ADCK0 is used, AD timer is selected as a trigger, and the scan group is started in a constant period.

Allocate the virtual channel 0 (AN000) to the scan group 3 (SG3) with AD timer.

Set the initial phase (ADCK0ADTIPR3) to 0.25us and the AD timer period (ADCK0ADTPRR3) to 100 μ s.

When the AD timer is started, the initial phase (ADCK0ADTIPR3) is loaded into the AD timer and counted down. When the AD timer becomes 0, the AD timer trigger is output and SG3 A/D conversion is started. At the same time, the AD timer cycle (ADCK0ADTPRR3) is loaded into the AD timer, the down count is started again, and A/D conversion is started when it becomes 0. After that, the AD timer cycle is loaded, and the A/D conversion operation is repeated. The A / D conversion value is stored in the variable by the scan end interrupt.

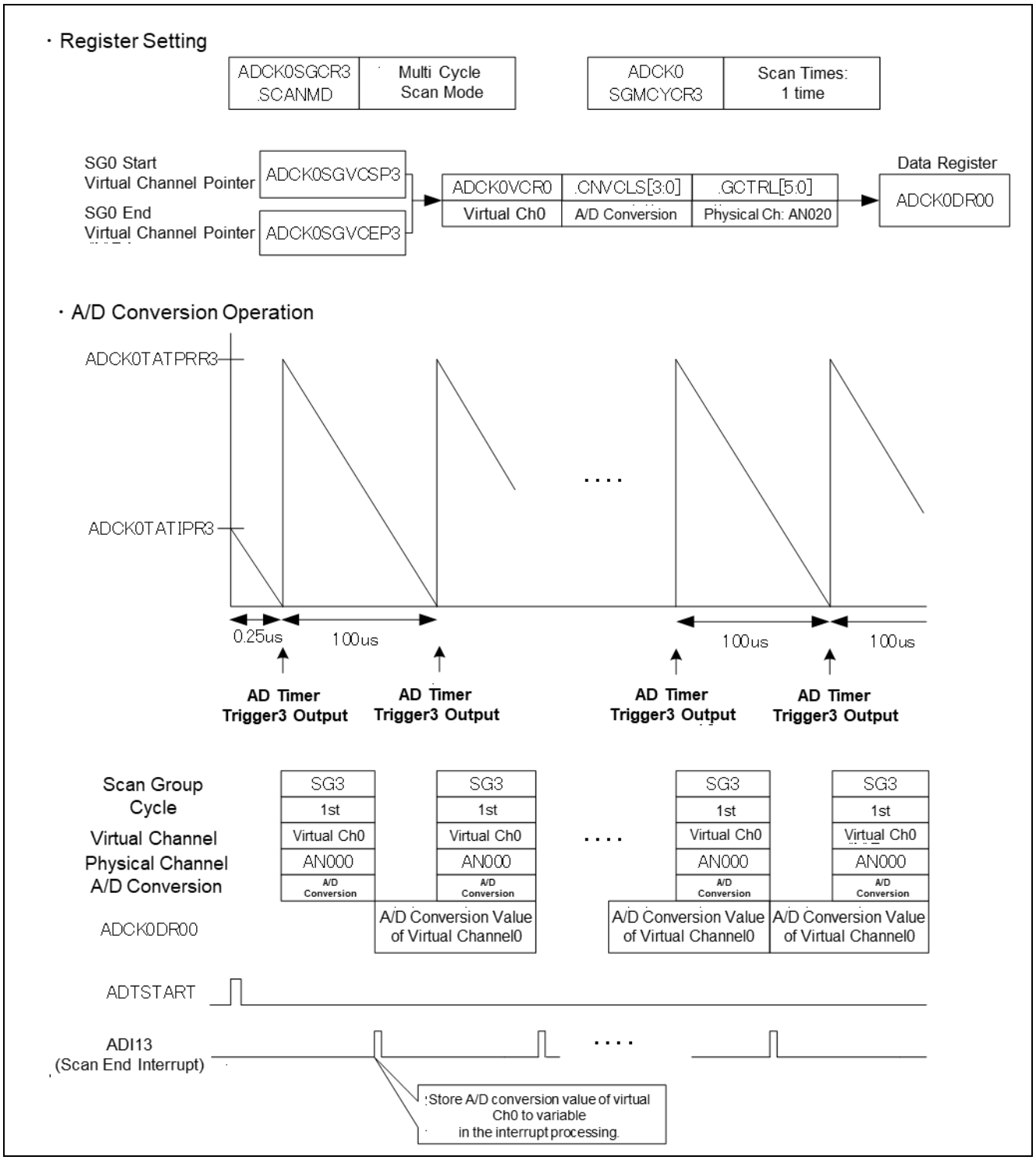

Figure 3-11 AD Timer Operation Example

3.3.4 Software Explanation

- Module Explanation

Module List in this operation example is shown below.

Table 3-4 Module List

Module Name	Label Name	Function
Main routine	main_pe0	Perform the various settings and application start.
ADCK initialize routine	ADCK_init	Initialize the ADCK.
Interrupt initialize routine	intc_init	Initialize the ADCK interrupt.
ADCK interrupt process routine	eiint444	Store the A/D conversion result in variable by the virtual scan group end interrupt processing.

- Register Setting

The register setting for various function in this operation example is shown below.

Table 3-7 ADCK Register Setting

Register Name	Setting Value	Function
ADCK0VCR00	0x00000000	Conversion type: normal A/D conversion
		Not select the wait time table.
		No DFE entry.
		Output the virtual channel end interrupt.
		Physical Ch0/Sub CH0（AN000）
ADCK0ADCR1	0x00	Synchronous suspend
ADCK0ADCR2	0x00	Signed 12bit integer format.
ADCK0SGCR3	0x52	Enable ADTSTART.
		Multi-cycle scan mode
		Output “ADI13” in the end of “SG3”.
		Select AD timer trigger3 as SG3 trigger.
ADCK0SGVCPR3	0x0000	Start virtual channel 0, End virtual channel 0
ADCK0SGMCYCR3	0x00	One scan in multi-cycle scan mode.
ADCK0ADTIPR3	1	AD timer initial phase：25ns
ADCK0ADTPRR3	4000	AD timer initial phase：100μs

Table 3-8 Interrupt Register Setting

Register Name	Setting Value	Function
EIBD444	0x00000000	Bind the interrupt to PE0 (CPU0).
EIC444	0x004F	Refer to table/ Priority level 15

- Operation Flow

Flow chart in this operation example is shown below.

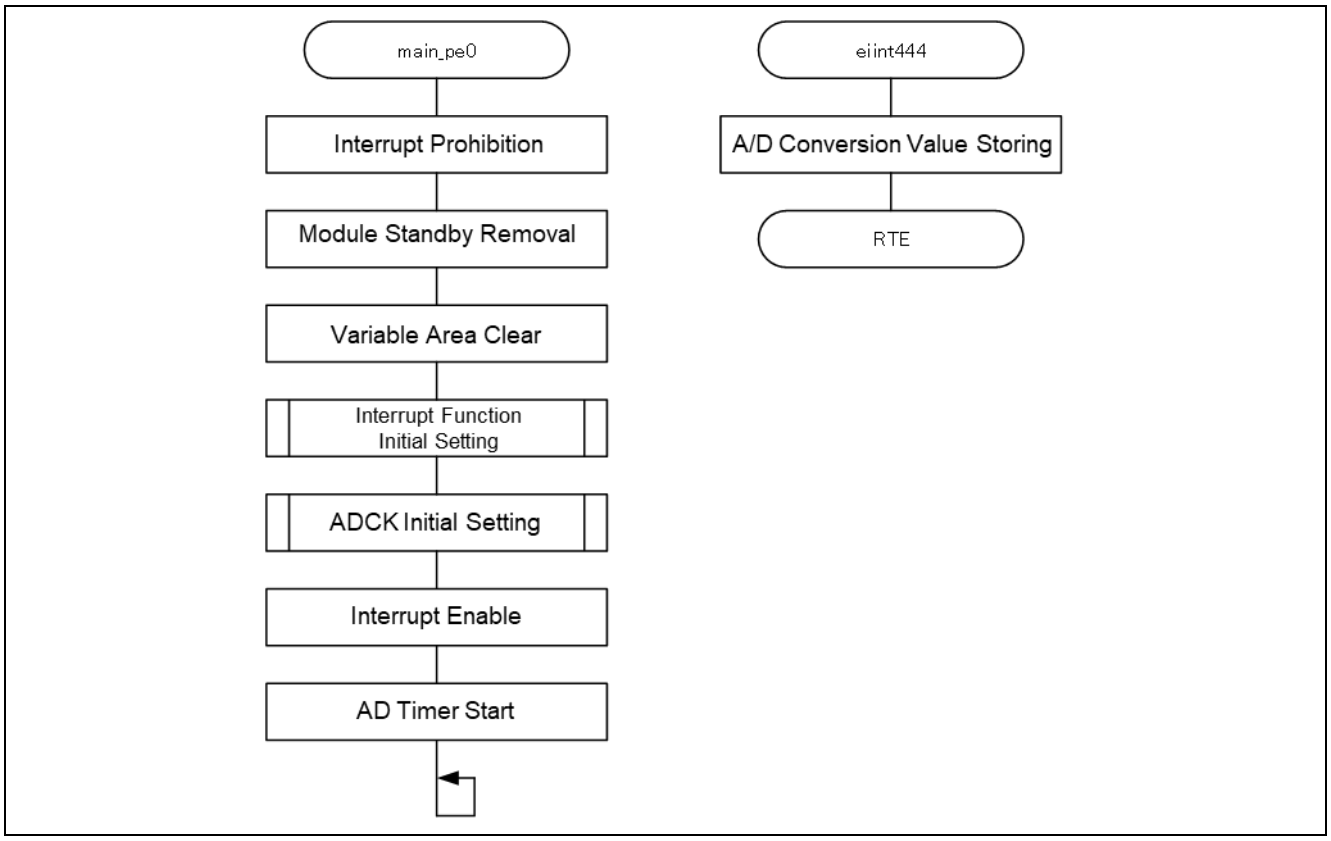

Figure 3-12 Flow Chart

3.4 Reading via IFC using DMA Transfer

3.4.1 Specification Overview

Describes how to use DAM transfer to store the A/D conversion value of the floating-point data in the array variable via IFC.

Treat AD timer as the trigger, perform the A/D conversion using scan group 3 (SG3), and convert the A/D value to the floating-point format by reading it via IFC module.

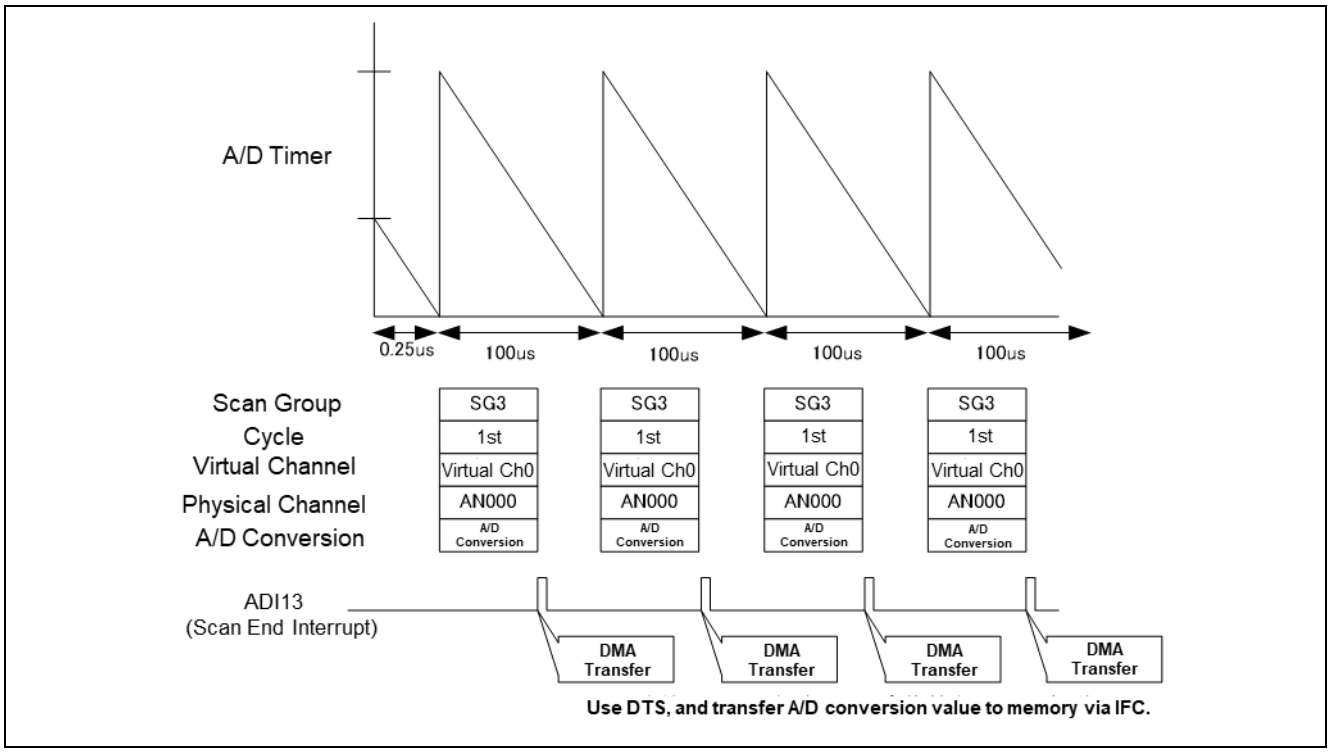

Figure 3-13 Operation Example using DTS

3.4.2 Use Function

The function used in this operation example is shown below.

- A/D Converter（ADCK0）
- IFC（Integer/Floating-point Conversion Module）
- DMA（DTS（Data Transfer Service））

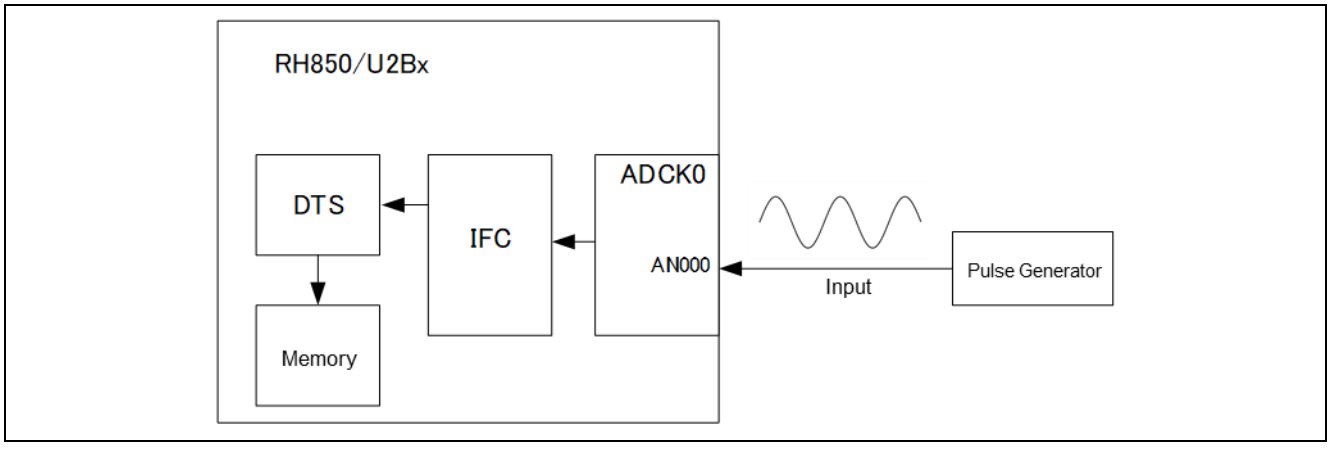

Figure 3-14 System Configuration

3.4.3 Explanation for Operation Example

In this operation example, A/D conversion is performed at regular intervals using the AD timer, and the A/D conversion value of floating-point data is stored in the array variable via IFC using DMA transfer.

Input 100Hz Sin wave (amplitude 0 to 5V) to AN000 of ADCK0 module using the pulse generator, etc. Treat AD timer as the trigger, and set the A/D conversion using scan group 3 (SG3). Convert the A/D value to the floating-point format by reading it via IFC module.

Set the interval of the AD timer is set to 100 μ s . Enable the scan end interrupt ADI03 for each A/D conversion. Store the A/D conversion value to the array variable by the DMA transfer using ADI03 as the DTS trigger factor.

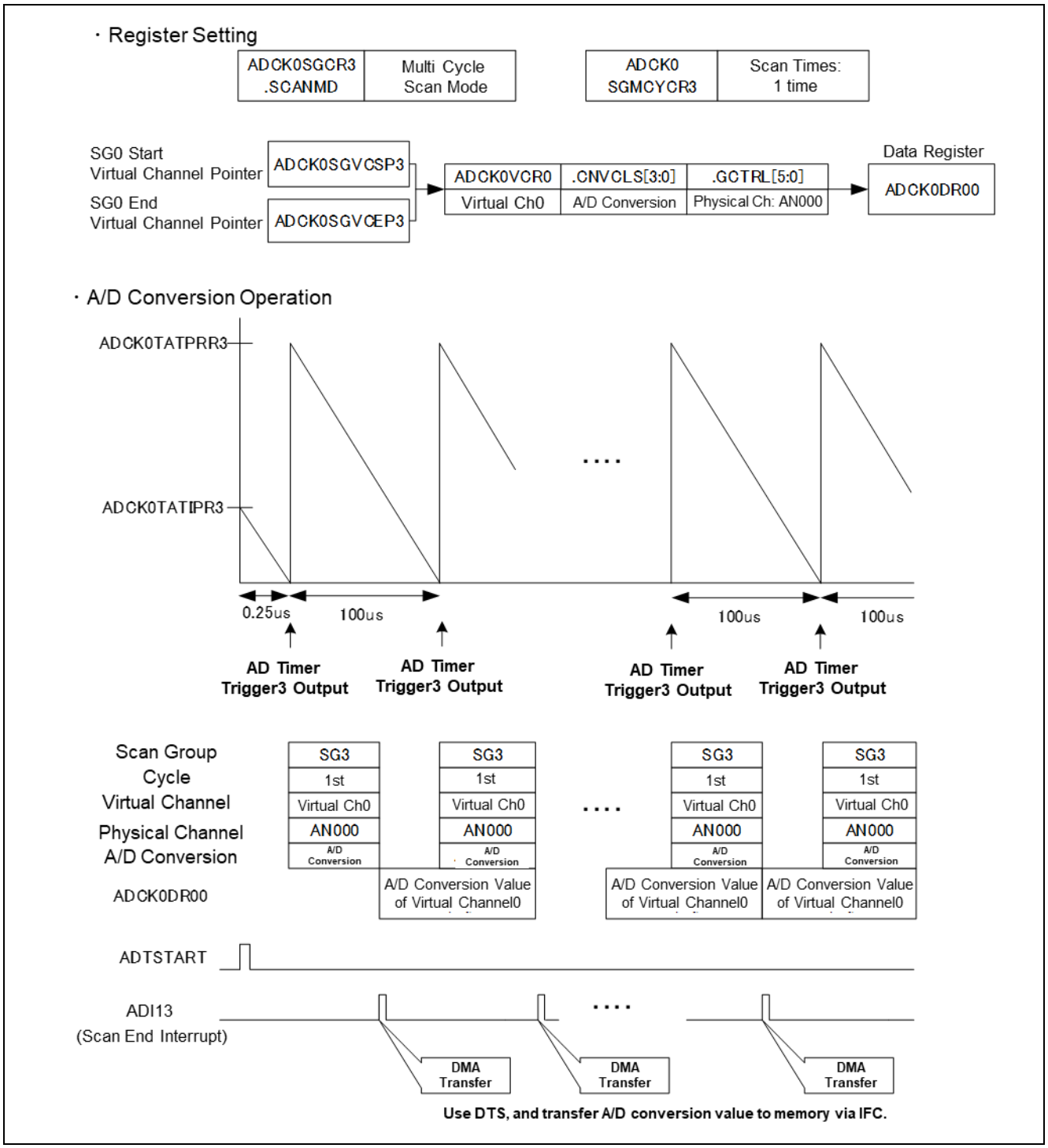

Figure 3-15 Example of Reading via IFC using DTS

3.4.4 Software Explanation

- Module Explanation

Module List in this operation example is shown below.

Table 3-9 Module List

Module Name	Label Name	Function
Main routine	main_pe0	Perform the various settings and application start.
ADCK initialize routine	adcb_init	Initialize the ADCK.
DTS initialize routine	dts_init	Initialize the DTS. Use DTS ch7 that operates ADCK0 Scan Group 3 End Interrupt ADI03 as a trigger.
Interrupt initialize routine	intc_init	Set the transfer completion interrupt including DTS ch7.
DTS transfer end interrupt process routine	eiint62	Confirm that the factor of the transfer end interrupt is DTS ch7, clear the interrupt factor, reset the DTS, and restart the DMA transfer.

- Register Setting

The register setting for various function in this operation example is shown below.

Table 3-10 ADCK Register Setting

Register Name	Setting Value	Function
ADCK0VCR00	0x00000000	Conversion type: normal A/D conversion
		Not select the wait time table.
		No DFE entry.
		Output the virtual channel end interrupt.
		Physical Ch0/Sub CH0（AN000）
ADCK0ADCR1	0x00	Synchronous suspend
ADCK0ADCR2	0x00	Signed 12bit integer format
ADCK0SGCR3	0x52	Enable ADTSTART.
		Multi-cycle scan mode
		Output “ADI13” in the end of “SG3”.
		Select AD timer trigger as SG3 trigger.
ADCK0SGVCPR3	0x0000	Start Virtual Channel 0
		End Virtual Channel 0
ADCK0SGMCYCR3	0x00	One scan in multi-cycle scan mode.
ADCK0ADTIPR3	1	AD timer initial phase ：25ns
ADCK0ADTPRR3	4000	AD timer period phase：100μs

Table 3-11 Interrupt Register

Register Name	Setting Value	Function
EIBD62	0x00000000	Bind the interrupt to PE0 (CPU0).
EIC62	0x004F	Refer to table/ Priority level 15

Table 3-12 DTS Register

Register Name	Setting Value	Function
DTSPR0	0x0FFF3FFF	Set DTS ch57 to highest priority (00B).
DTS057CM	0x00000000	Initialize the channel master register of DTS ch57.
DTSA057	FDR000 address	Set DMA transfer source address to FDR000 of IFC module.
DTDA057	fl_ADCK0_sg3 [0] address	Set DMA transfer source address to lead address of array variable fl_ADCK0_sg3 [0].
DTTC057	0x00000100	Set 256 data to number of transfer data.
DTTCT057	0x00004048	Set DMA transfer of DTS ch7 as below. ・Continue DMA transfer if DMA transfer error generates. ・Chain disable ・Transfer count match interrupt disable ・Transfer complete interrupt enable ・Reload function 1/2 disable ・Transfer source address count direction：Increment ・Transfer source address count direction：Fixed ・Transfer size：32bit ・Transfer mode：Single transfer
DTFSC057	0x000000B1	DTS status clear
DTFSL057	0x00000001	DTS transfer request set

- Operation Flow
  The flow chart in this operation example is shown below.

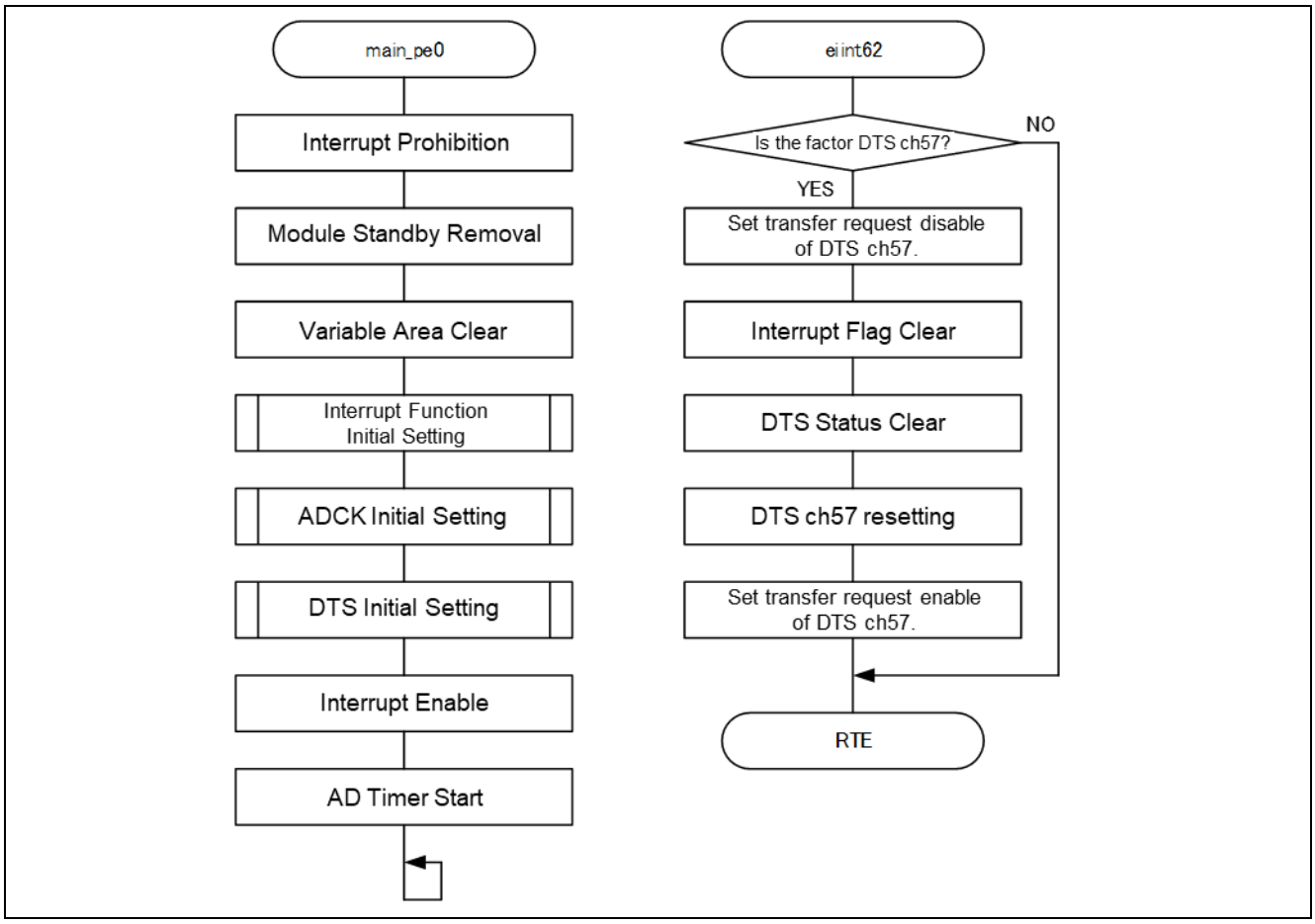

Figure 3-16 Flowchart

3.5 Addition Function

3.5.1 Specification Overview

This section describes how to perform normal A/D conversion in addition mode.

Allocate two virtual channels (AN000, AN001) to scan group 0 (SG0), scan once in multi-scan mode, and add four times for each virtual channel. At the end of the scan group, the converted values of AN000 and AN001 are stored in variables, and the operation ends.

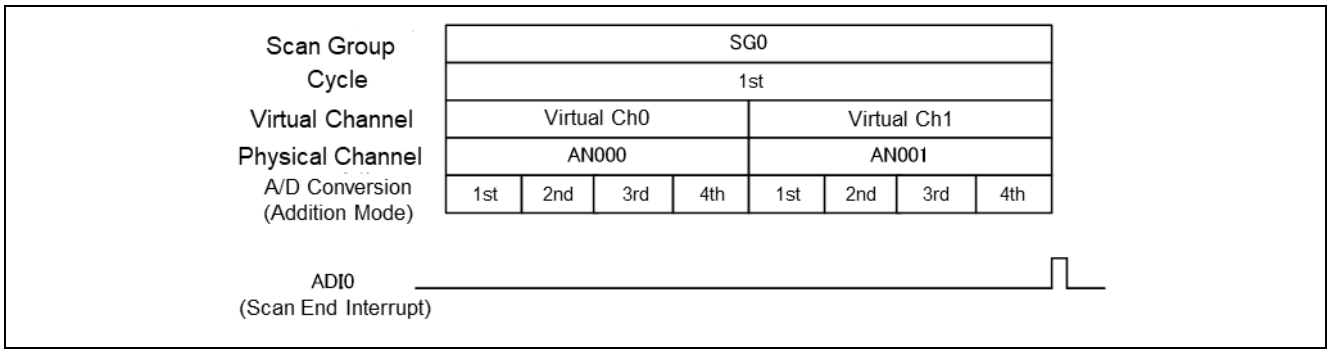

Figure 3-17 Normal AD Conversion Operation using Addition Mode

3.5.2 Use Function

The hardware function in this operation example is shown below.

- A/D Convertor（ADCK0）

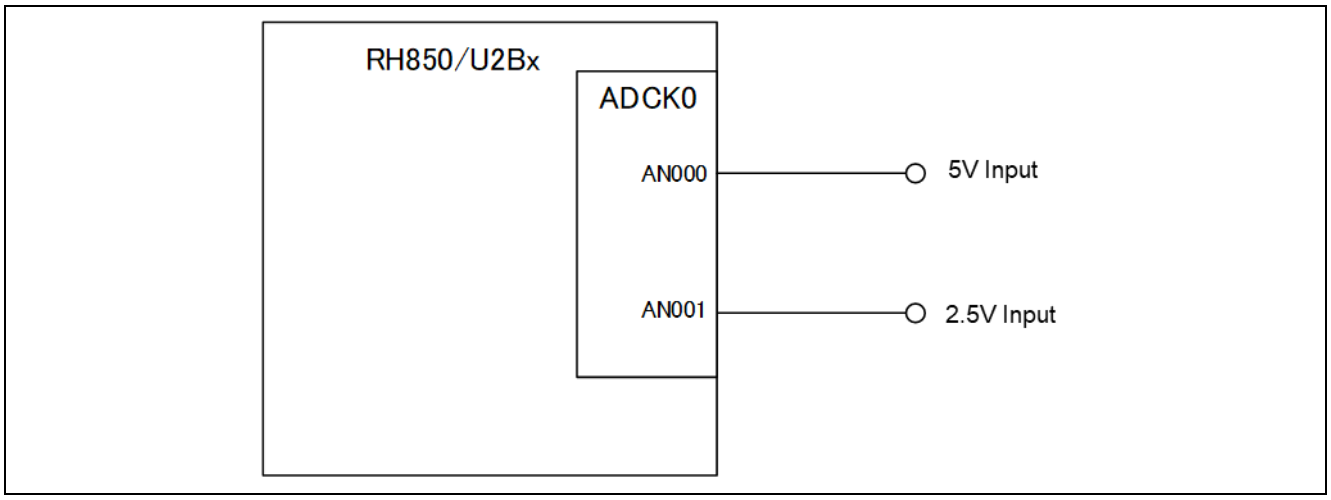

Figure 3-18 System Configuration Diagram

3.5.3 Explanation for Operation Example

In this operation example, perform normal A/D conversion in addition mode using AN000 and AN001 of ADCK0 module.

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

For analog signal, input 5.0V to AN000 of ADCK0 module, and input 2.5V to AN001 of ADCK0 module.

Starting with soft trigger ADSTART. Perform A/D conversion of AN000 and AN001 for the number of additions (four times). Enable scan end interrupt (at the end of the scan group), store the result of four times A/D conversion additions in the interrupt processing to the variable, and finish the processes.

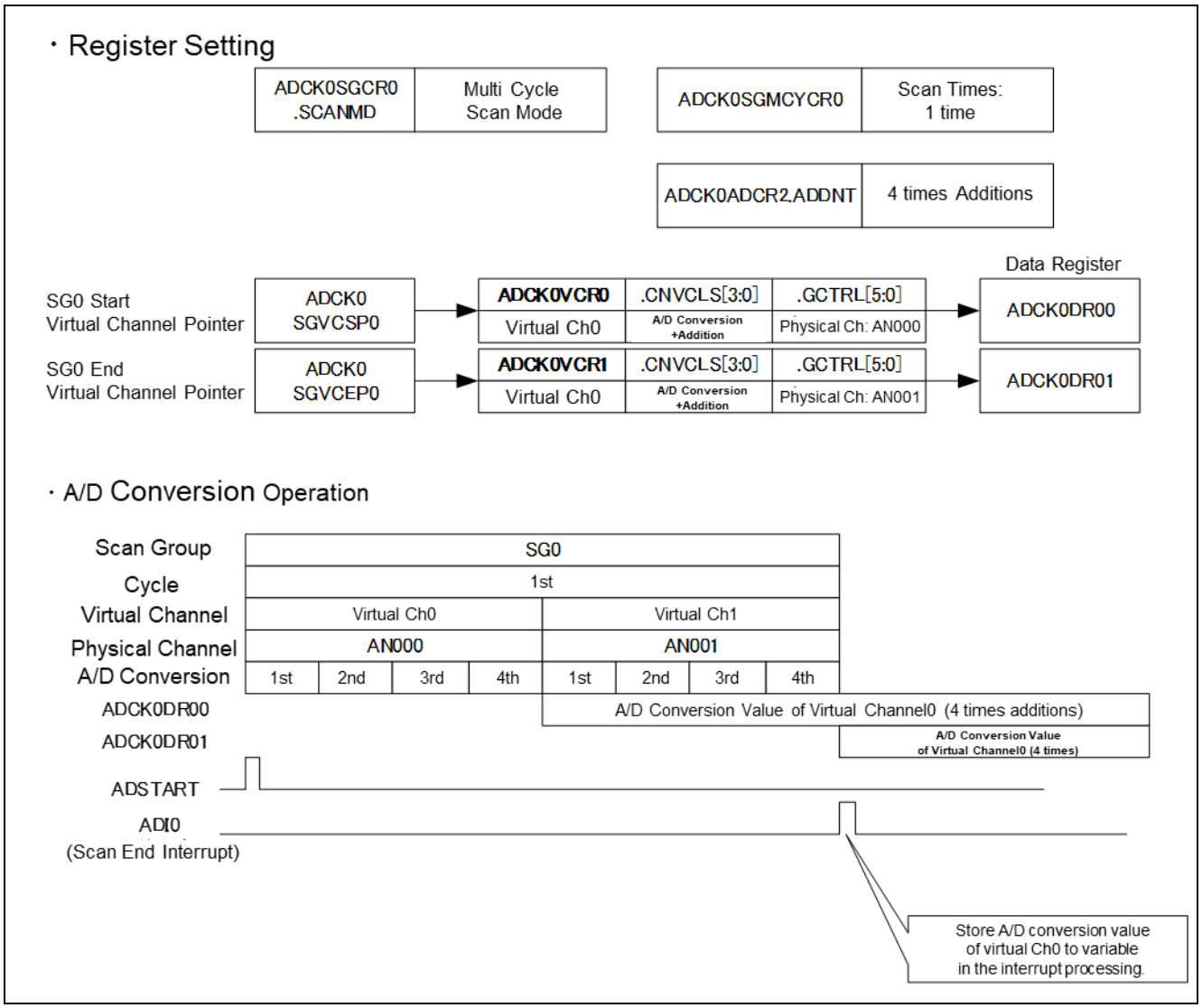

Figure 3-19 Operation Example of Addition Function

3.5.4 Software Explanation

- Module Explanation

Module List in this operation example is shown below.

Table 3-13 Module List

Module Name	Label Name	Function
Main routine	main_pe0	Perform the various settings and application start.
ADCK initialize routine	adcb_init	Initialize the ADCK.
Interrupt initialize routine	intc_init	Initialize the ADCK interrupt.
ADCK interrupt process routine	eiint441	Store the A/D conversion result in variable by the virtual scan group end interrupt processing.

- Register Setting

The register setting for various function in this operation example is shown below.

Table 3-14 ADCB Register Setting

Register Name	Setting Value	Function
ADCK0VCR00	0x00002000	Conversion type: normal A/D conversion + Addition mode
		Not select the wait time table.
		No DFE entry.
		Not output the virtual channel end interrupt
		Ch0/Sub CH0（AN000）
ADCK0VCR01	0x00002001	Conversion type: normal A/D conversion
		Not select the wait time table.
		No DFE entry.
		Not output the virtual channel end interrupt.
		Physical Ch0/Sub CH1 (AN001)
ADCK0ADCR1	0x00	Synchronous suspend
ADCK0ADCR2	0x11	Signed 12bit integer format
		Add 4 times.
ADCK0SGCR0	0x50	Enable ADSTART.
		Multi-cycle scan mode
		Output “ADI00” in the end of “SG0”.
		Prohibit the trigger input to “SG0”.
ADCK0SGVCPR0	0x0100	Start virtual channel0, end virtual channel 1.
ADCK0SGMCYCR0	0x00	One scan in multi-cycle scan mode.

Table 3-15 Interrupt Register

Register Name	Setting Value	Function
EIBD441	0x00000000	Bind the interrupt to PE0 (CPU0).
EIC441	0x004F	Refer to table/ Priority level 0

- Operation Flow
  Flow chart in this operation example is shown below.

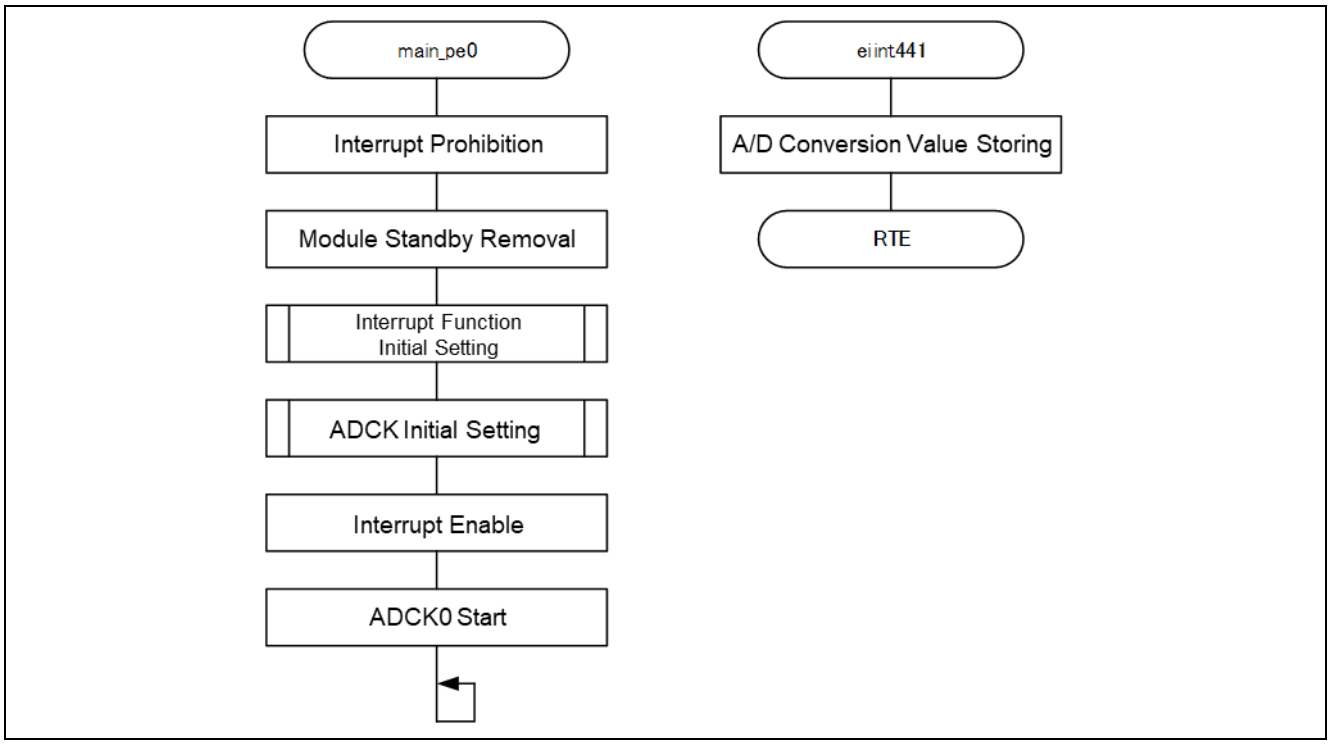

Figure 3-20 Flowchart

3.6 ADC Accumulation Function (ASF)

3.6.1 Specification Overview

This section describes how to perform A/D conversion using the ADC Accumulation Function (ASF).

ASF accumulates the A/D conversion results outputted from the ADC, stores the accumulation results in a register each time the accumulation counter matches the specified compare value, and generates the accumulation end interrupt (ASI00).

Allocate one virtual channel (AN000) to scan group 0 (SG0) and set continuous scan mode. The number of accumulations is set to 32, the accumulation counter and the compare value match, and the accumulation result is stored in a variable in the generated accumulation end interrupt processing.

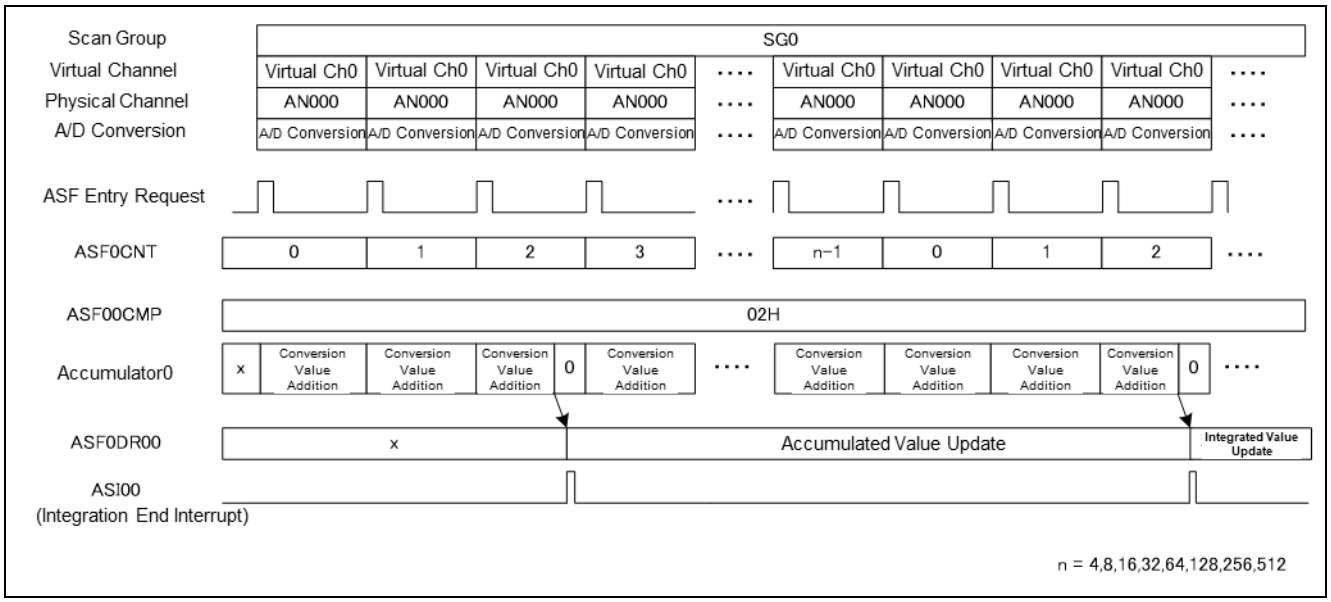

Figure 3-21 A/D Conversion Operation using ASF

3.6.2 Use Function

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

- A/D Convertor（ADCK0）
- ASF（ADC Accumulation Function）

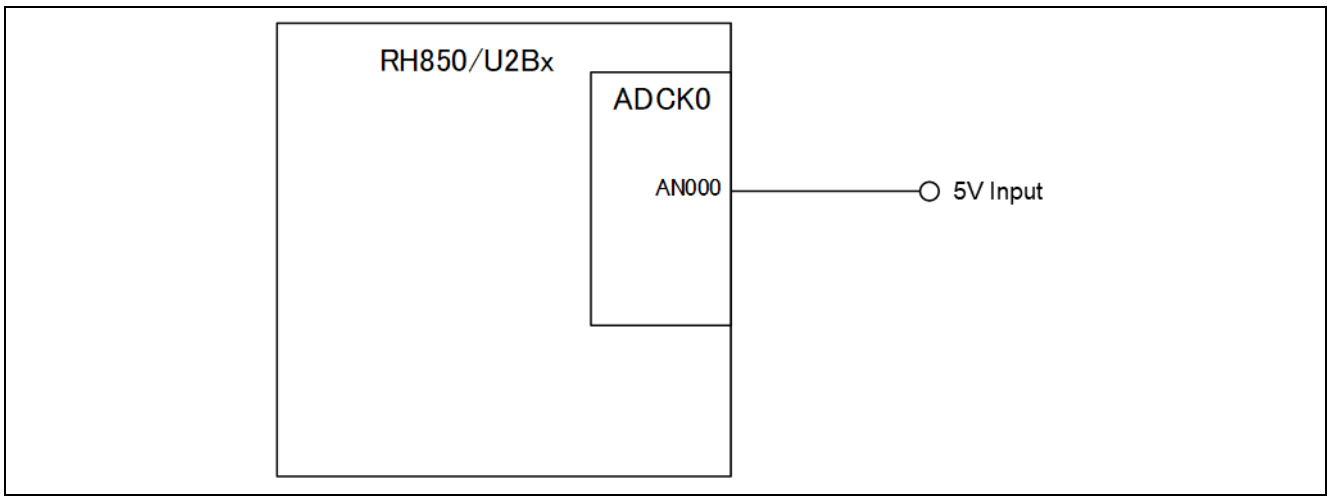

Figure 3-22 System Configuration Diagram

3.6.3 Explanation for Operation Example

In this operation example, perform A/D conversion using the ASF function for using AN000 of the ADCK0 module.

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

For analog signal, input 5.0V to AN000 of ADCK0 module.

Starting with soft trigger ADSTART. Perform A/D conversion of AN000 with continues scan mode.

The ASF function uses accumulation channel 00, and the number of accumulations is 32. The accumulation compares (ASF0CMP00) of the accumulation channel 00 is set to 6, and when it matches the accumulation counter (ASF0CNT), the accumulation data (ASF0DR00) is updated. The accumulated data is stored in the variable during the accumulation end interrupt (ASI00) processing that generates at the same time as this accumulated data update.

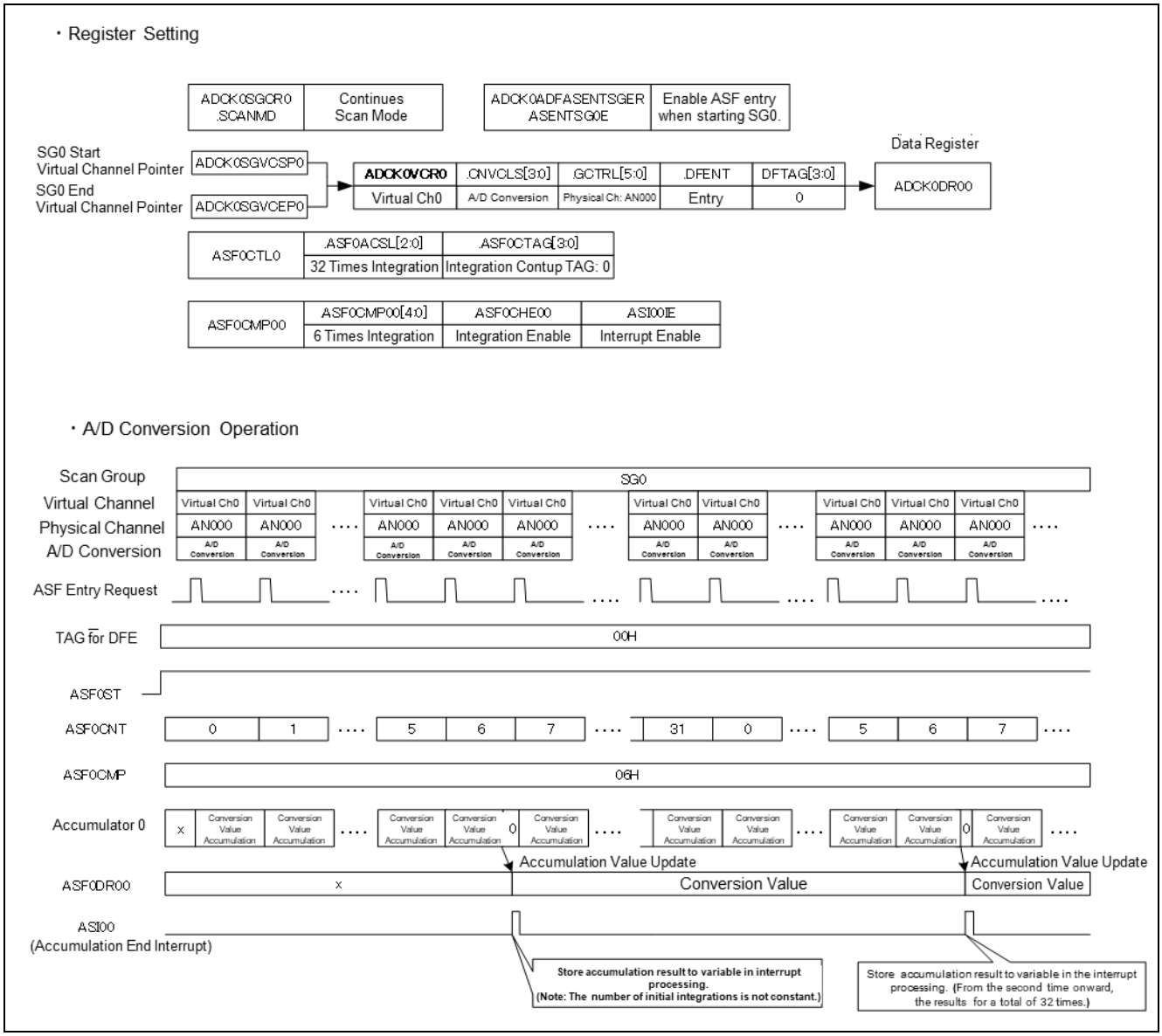

Figure 3-23 ASF Operation Example

3.6.4 Software Explanation

- Module Explanation

Module list in this operation example is shown below.

Table 3-16 Module List

Module Name	Label Name	Function
Main routin	main_pe0	Perform the various settings and application start.
ADCK initialize routin	ADCK_init	Initialize ADCK, ASF.
Interrupt initialize routin	intc_init	Initialize ADCK interrupt.
ASF interrupt processing routine	eiint476	Store the accumulate result in variable by the accumulate end interrupt processing.

- Register Setting

The register setting for various function in this operation example is shown below.

Table 3-17 ADCK Register Setting

Register Name	Setting Value	Function
ADCK0VCR00	0x00001000	Conversion type: normal A/D conversion
		Not select the wait time table.
		DFE Entry
		Not output the virtual channel end interrupt
		Physical Ch0/Sub CH0 (AN000)
ADCK0ADCR1	0x00	Synchronous suspend
ADCK0ADCR2	0x00	Signed 12bit integer format
ADCK0SGCR0	0x60	Enable ADSTART.
		Continuous scan mode
		Not output “ADI00” in the end of “SG0”.
		Prohibit the trigger input to “SG0”.
ADCK0SGVCPR0	0x0000	Start virtual channel0, end virtual channel 0.
ADCK0DFASENTSGER	0x0001	Enable ASF entry in SG0 starting.

Table 3-18 ASF Register Setting

Register Name	Setting Value	Function
ASF0CTL0	0x00000300	Accumulation times: 32
		Accumulation countup TAG 00
ASF0CMP00	0x8086	Enable ASI00 interrupt.
		Enable accumulation.
		Accumulation comparing match 6
ASF0CTL1	0x01	Enable accumulation counter operation and accumulation processing.

Table 3-19 Interrupt Register

Register Name	Setting Value	Function
EIBD476	0x00000000	Bind the interrupt to PE0 (CPU0).
EIC476	0x0040	Refer to table/ Priority level 0

- Operation Flow
  Flowchart in this operation example is shown below.

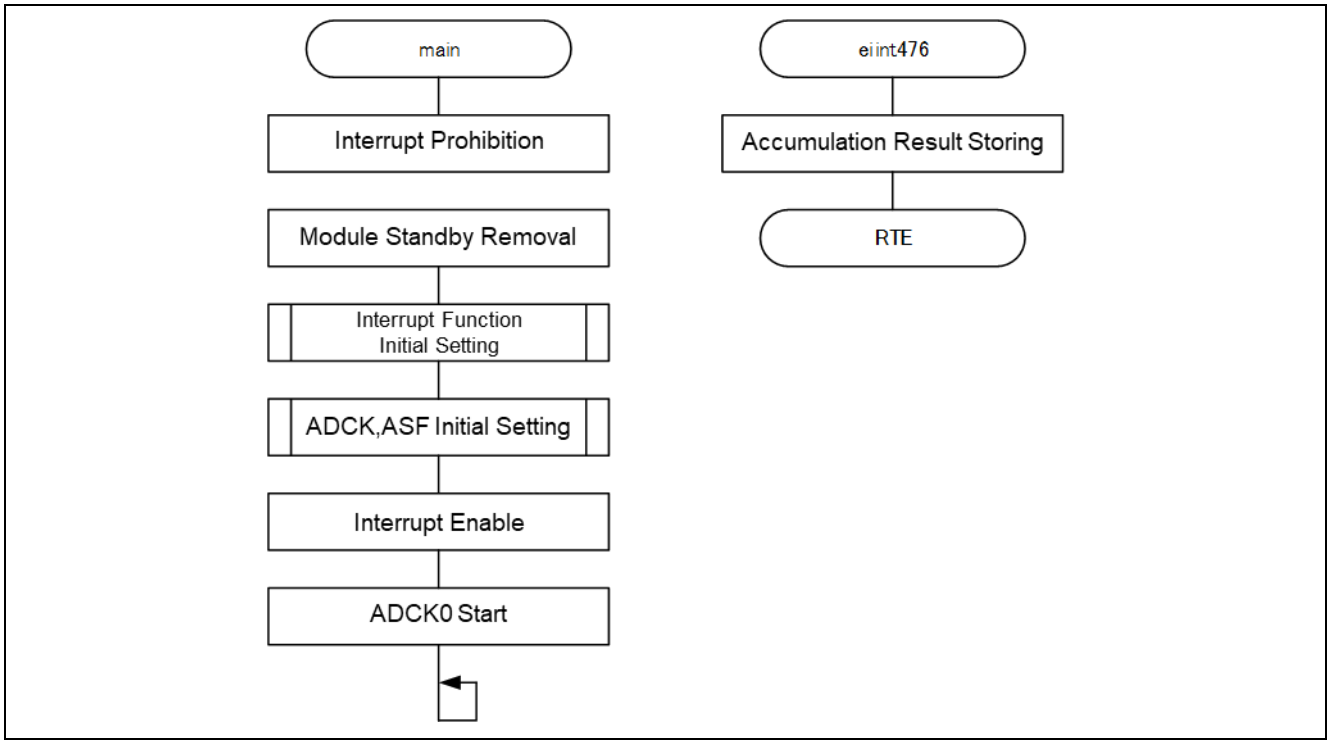

Figure 3-24 Flowchart

3.7 Operation Example using External Analog Multiplexer

3.7.1 Specification Overview

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

Allocate 7 virtual channels to scan group 0 (SG0). Use external multiplexer input pin as physical channel. (ADCK:AN000)

Start DMA (DTS) every time a virtual channel starting, transfer MPX value for external analog multiplexer control to I/O port, and control the analog multiplexer.

Store conversion value to variable in SG0 scan group end.

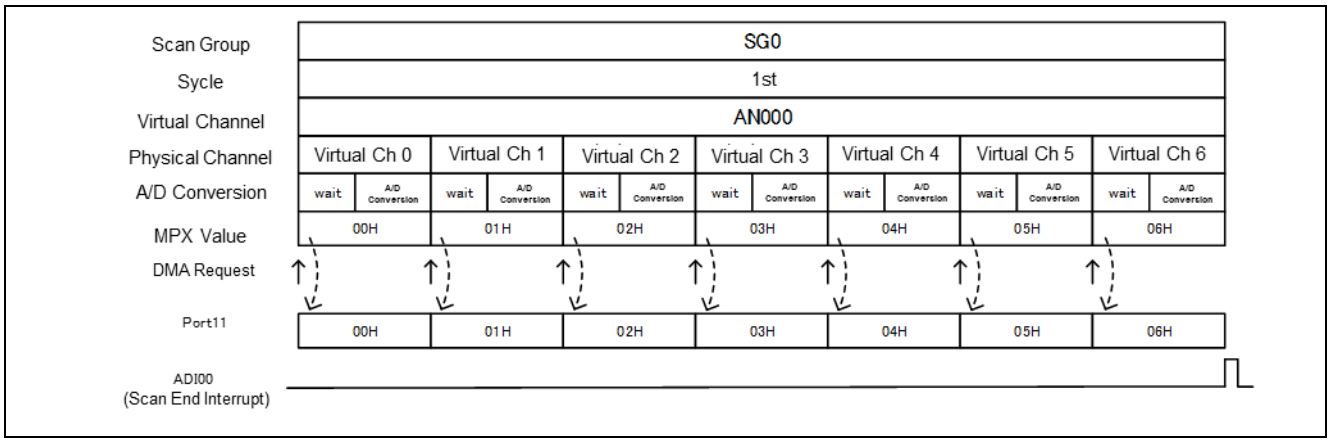

Figure 3-25 A/D Conversion Operation using External Analog Multiplexer

3.7.2 Use Function

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

Input to AN000 is selected by switching 0.5 to 3.5V from external multiplexer by port control.

- A/D Converter (ADCK0)
- Pin (PORT11)
- DMA (DTS)

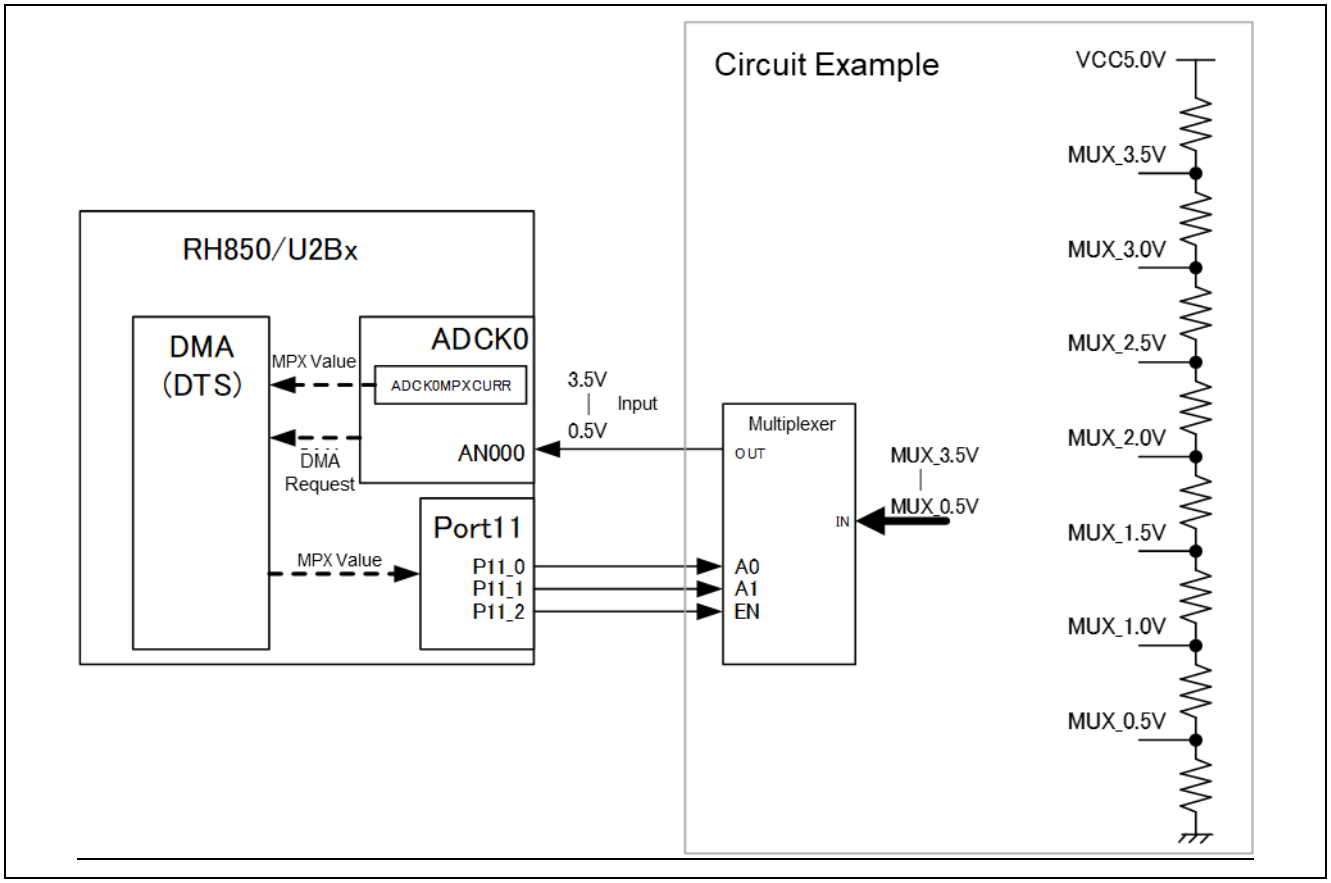

Figure 3-26 System Configuration

3.7.3 Explanation for Operation Example

In this operation example, perform the A/D conversion using AN000 of ADCK0 module and external analog multiplexer.

The ADCK supports the interrupt and DMA transfer for external analog multiplexer. It is possible to generate the interrupt request and start the DMA at the start of specified virtual channel. In this operation example, transfer the MPX value to external multiplexer in cooperation with the I/O port.

Allocate virtual channel 0 to 6 virtual channel to scan group 0 (SG0).

Analog signal is inputted from external multiplexer to AN000 corresponding to the external multiplexer input terminal.

Ser the MPX value transferred to the external analog multiplexer in ADCK0VCRn.GCTRL[5:0] of each channel. Transfer ADCK0VCRn.GCTRL[5:0] to ADCK0MPXCURR at the start of each virtual channel, issue the interrupt request (ASMPXI0) or DMA request at the same time.

In DMA (DTS), Set ADCK0MPXCURR as the transfer source and P11 of I/O port as the transfer destination. Therefore, the DMA request is issued at each start of each virtual channel of SG0, and the start set MPX value is transferred to the external analog multiplexer.

Control the external analog multiplexer by transferred MPX value, change the input voltage to AN000 in 0.5 to 3.5V (in O.5V increments).

Enable the scan end interrupt ADI00, and store the A/D conversion value to the variable in interrupt processing after A/D conversion is completed.

Also, in this operation example, ADCK0MPXCURR and P11 value are stored to the variable in variables in the interrupt (ASMPXI0) processing that generates every time the virtual channel is started to confirm whether the MPX value is transferred as expected. Therefore, it is possible to confirm the MPX value transferred to ADCK0MPXCURR at the start of the current virtual channel and the MPX value transferred to P11 by DTS.

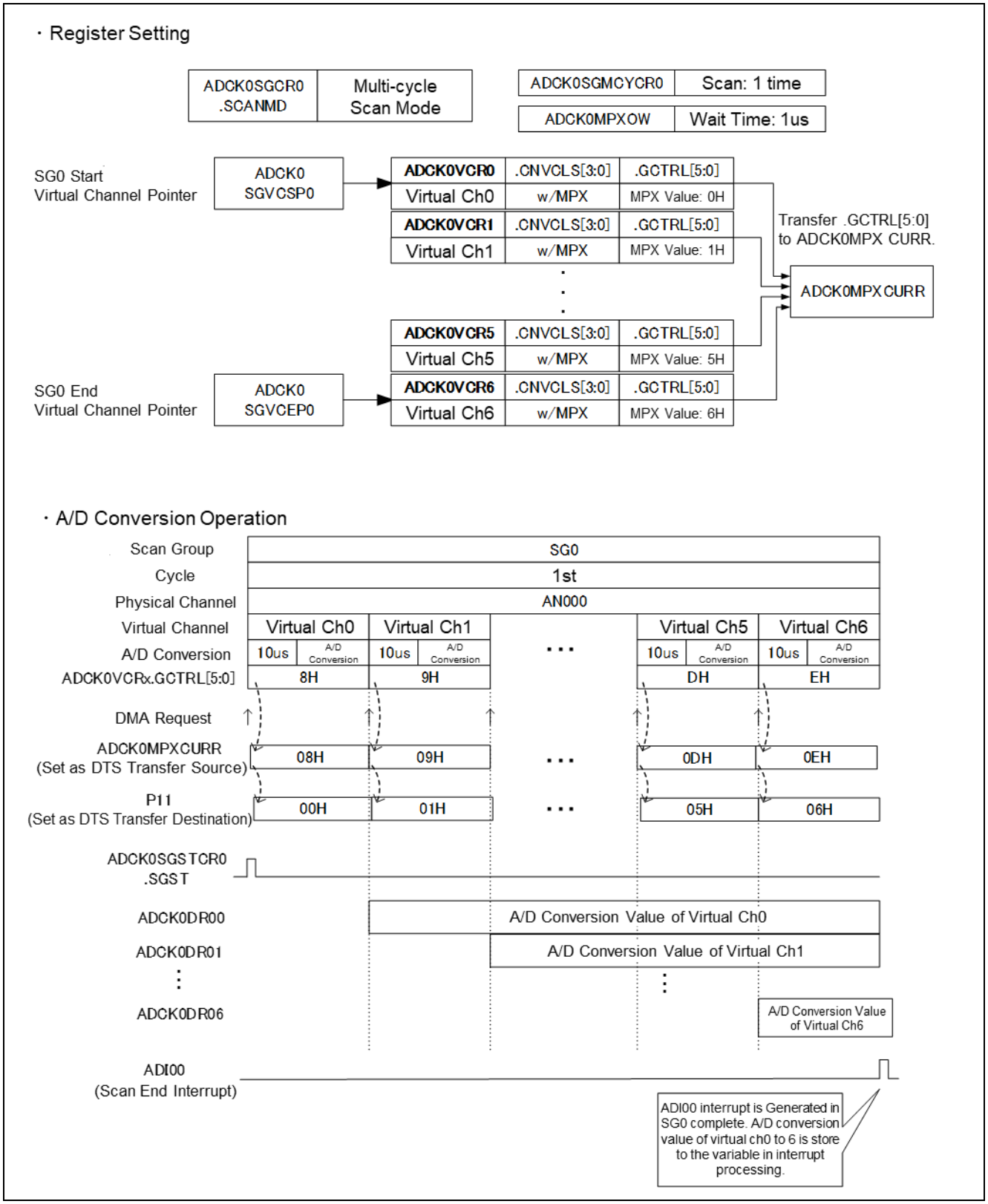

Figure 3-27 Operation Example using External Analog Multiplexer and DTS

3.7.4 Software Explanation

- Module Explanation

Module list in this operation example is shown below.

Table 3-20 Module List

Module Name	Label Name	Function
Main routine	main_pe0	Perform the various settings and application start.
Port initialize routine	port_init	Initialize the port.
ADCK initialize routine	ADCK_init	Initialize the ADCK.
DTS initialize routine	dts_init	Initialize the DTS.
Interrupt initialize routine	intc_init	Initialize the ADCK interrupt.
ADCK0/SG0 interrupt processing routine	eiint441	Store the A/D conversion result to the variable in scan group complete interrupt processing.
ADCK0/MPX interrupt processing routine	eiint437	Confirm the transfer status of the set MPX value. ①Store the value of ADCK0MPXCURR in the variable to confirm the MPX value transferred from ADCK0VCRn.GCTRL [5:0] to ADCK0MPXCURR at the start of the virtual channel. ②Store the P11 value in the variable to confirm the MPX value transferred from ADCK0MPXCURR to P11 by the DTS triggered by the virtual channel start trigger.
DTS transfer complete interrupt processing routine	eiint62	Clear the DTS transfer complete interrupt request (ch0).

- Register Setting

The register setting in this operation example is shown below.

Table 3-21 ADCK Register Setting

Register Name	Setting Value	Function
ADCK0VCR00	0x01002801	Conversion type: normal A/D conversion w / MPX
		Select wait time table.
		Not entry DFE.
		Not output virtual channel complete interrupt.
		MPX value: 0x00
ADCK0VCR01	0x01002901	Conversion type: normal A/D conversion w/MPX
		Select wait time table.
		Not entry DFE.
		Not output virtual channel complete interrupt.
		MPX value: 0x01
ADCK0VCR02	0x01002A01	Conversion type: normal A/D conversion w/MPX
		Select wait time table.
		Not entry DFE.
		Not output virtual channel complete interrupt.
		MPX value: 0x02
ADCK0VCR03	0x01002B01	Conversion type: normal A/D conversion w/MPX
		Select wait time table.
		Not entry DFE.
		Not output virtual channel complete interrupt.
		MPX value: 0x03
ADCK0VCR04	0x01002B01	Conversion type: normal A/D conversion w/MPX
		Select wait time table.
		Not entry DFE.
		Not output virtual channel complete interrupt.
		MPX value: 0x04
ADCK0VCR05	0x01002D01	Conversion type: normal A/D conversion w/MPX
		Select wait time table.
		Not entry DFE.
		Not output virtual channel complete interrupt.
		MPX value: 0x05
ADCK0VCR06	0x01002E01	Conversion type: normal A/D conversion w/MPX
		Select wait time table.
		Not entry DFE.
		Not output virtual channel complete interrupt.
		MPX VALUE: : 0x06
ADCK0ADCR1	0x00	Synchronous suspend
ADCK0ADCR2	0x10	Signed 12bit integer format
ADCK0SGCR0	0x50	Enable ADSTART.
		Multi-Sycle scan mode
		Output “ADI00” in the end of “SG0”.
		Prohibit the trigger input to “SG0”.
ADCK0SGVCPR0	0x0600	Start virtual channel0, end virtual channel 6.
ADCK0SGMCYCR0	0x00	One scan in multi-cycle scan mode.
ADCK0MPXINTER	0x01	Enable ADMPXI0interrupt.
ADCK0MPXCURCR	0x00	MSKC specified format: 0
ADCK0WAITTR0	0x0190	MPX wait time: 10μsec

Table 3-22 Interrupt Register Setting

Register Name	Setting Value	Function
EIBD441	0x00000000	Bind the interrupt to PE0 (CPU0).
EIC441	0x004F	Refer to table/ Priority level 15
EIBD437	0x00000000	Bind the interrupt to PE0 (CPU0).
EIC437	0x004F	Refer to table/ Priority level 15
EIBD63	0x00000000	Bind the interrupt to PE0 (CPU0).
EIC63	0x004F	Refer to table/ Priority level 15

Table 3-23 DTS Register Setting

Register Name	Setting Value	Function
DTSPR0	0xFFFFFFFC	ch0 Highest Priority
DTS050CM	0x00000000	Initialize channel master register of DTS ch0.
DTSA050	ADCK0MPXCURR Address	Transfer source: ADCK0MPXCURR
DTDA050	P11 Address	Transfer destination: P11
DTTC050	0x00000007	Transfer time：7 times
DTTCT050	0x00004144	Transfer complete interrupt: enable
		Reload function: unavailable
		Chain function: unavailable
		Destination address count: fixed
		Source address count: fixed
		Transfer data size: 16bit
		Single transfer
DTFSC050	0x000000B1	Clear DTS status.
DTFSL050	0x00000001	Set DTS transfer request.

- Operation Flow
  The flowchart in this operation example is shown below.

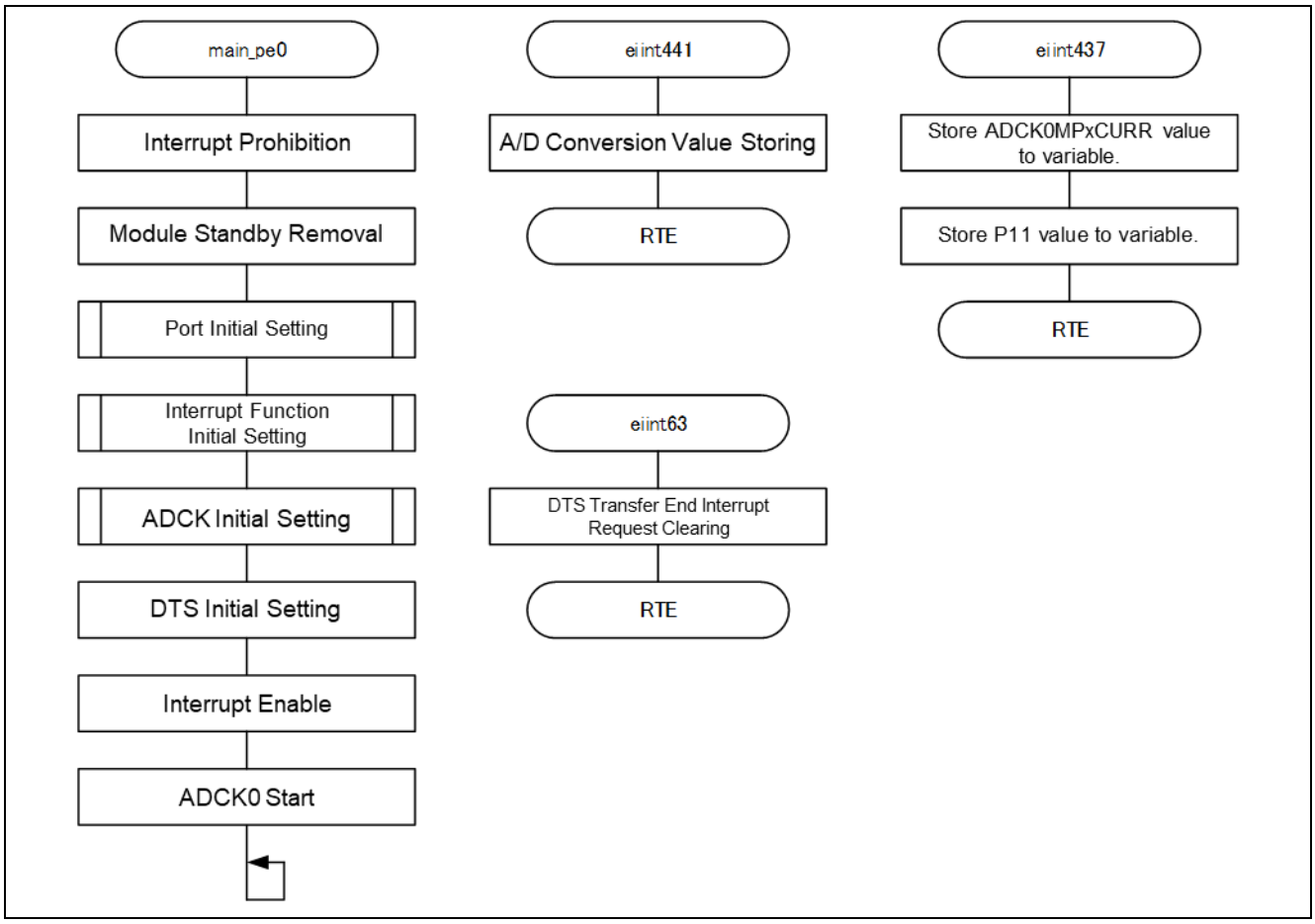

Figure 3-28 Flowchart

3.8 Sequential Scan of Optional Channel using DMA Transfer (Scatter Function)

3.8.1 Specification Overview

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

Allocate 3 virtual channels to scan group 0 (SG0), and scan by multi-scan mode. DMA is started in scan end interrupt (ADI00), and the conversion values of AN000, AN001, and AN002 are stored to the array variable.

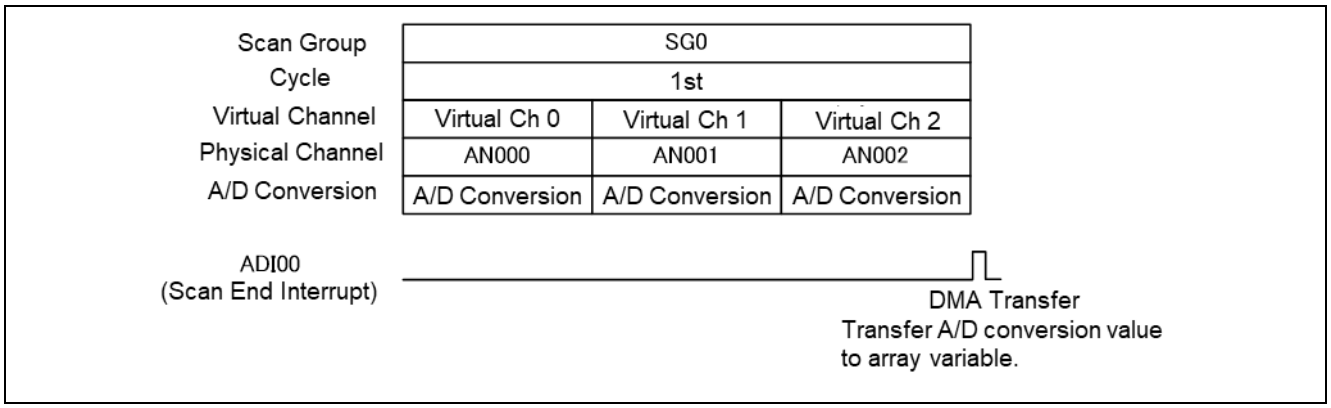

Figure 3-29 Operation Example using DMA Transfer

3.8.2 Use Function

The functions used in this operation example are shown below.

- A/D convertor（ADCK0）
- DMA（sDMAC0）

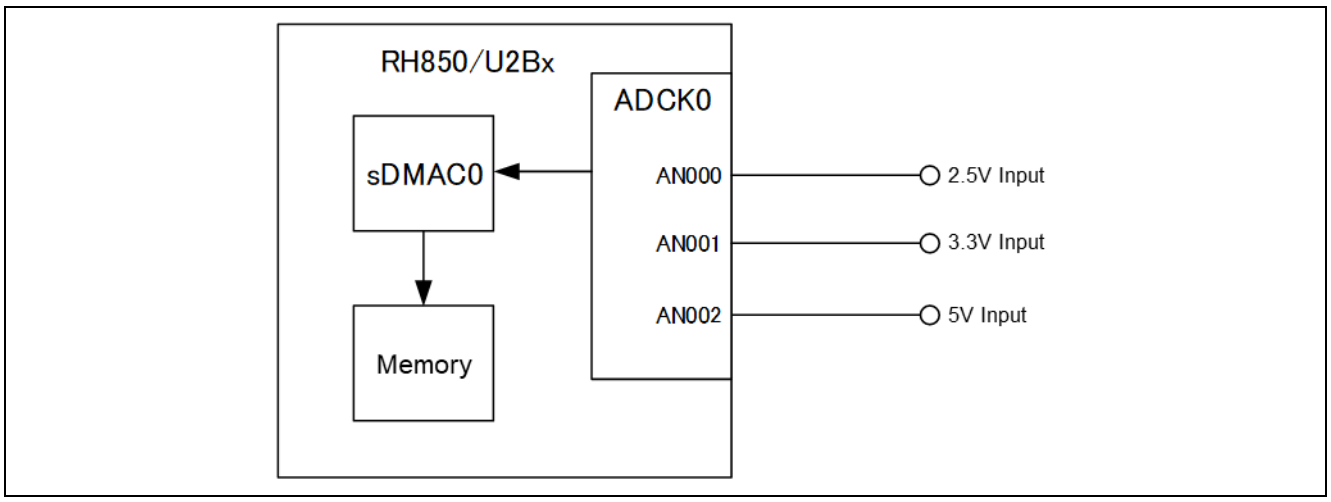

Figure 3-30 System Configuration

3.8.3 Explanation for Operation Example

In this operation example, perform normal A/D conversion in multi-scan mode using AN000, AN001, and AN002 of ADCK0 module. Also, store the A/D conversion value to the array variable using DAM conversion (scatter function).

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

For sDMAC, set the scan end interrupt (ADI00) to start trigger, enable the scatter function. Figure 3-31 shows the overview of scatter function. In scatter function, A/D conversion value can be transferred to each array variable with one channel of sDMAC.

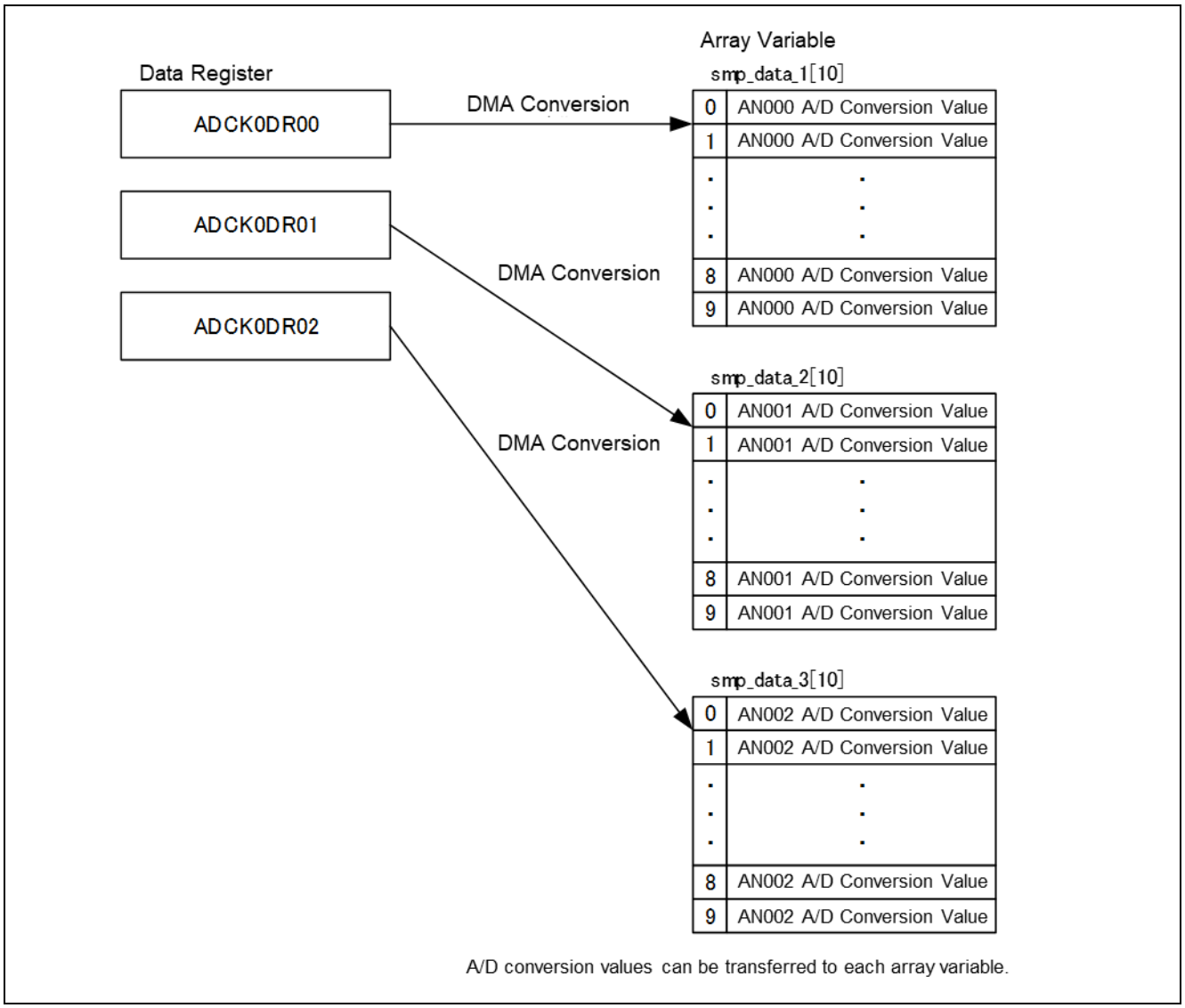

Figure 3-31 Overview of Scatter Function

Start A/D conversion with software trigger ADSTART. DMA is started by the scan end interrupt (ADI00), and A/D conversion value is stored to the array variable.

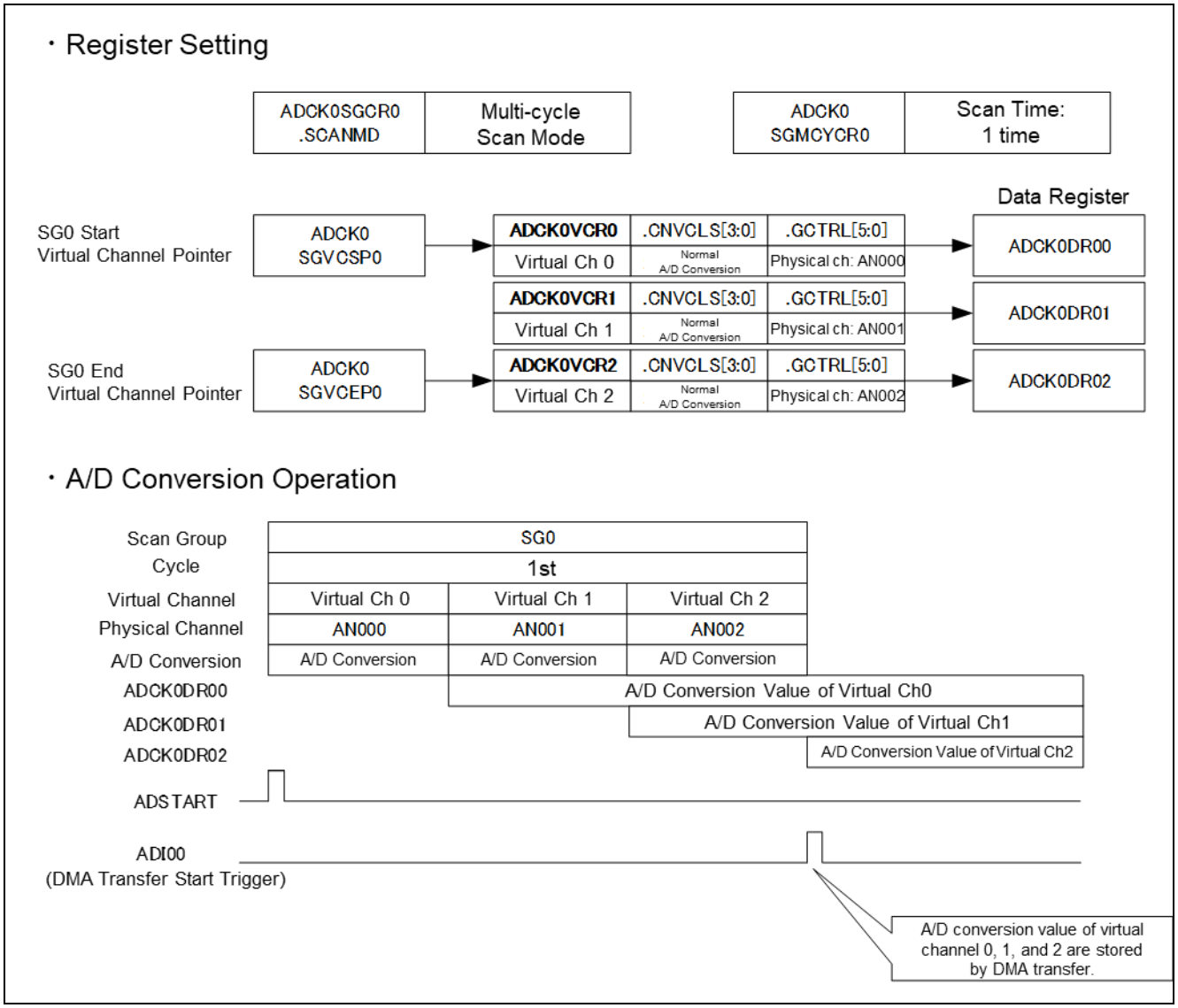

Figure 3-32 Operation Example using DMA Transfer

3.8.4 Software Explanation

- Module Explanation

Module list in this operation example is shown below.

Table 3-24 Module List

Module Name	Label Name	Function
Main routine	main_pe0	Perform the various settings and application start.
ADCK initialize routine	ADCK_init	Initialize the ADCK.
DTS initialize routine	sdmac_init	Initialize the sDMAC.
Interrupt initialize routine	intc_init	Initialize the sDMAC interrupt.
sDMAC transfer complete interrupt	eiint70	Perform sDMAC resetting and A/D conversion restarting by transfer complete interrupt processing.

- Register Setting

The register setting in this operation example is shown below.

Table 3-25 ADCK Register Setting

Register Name	Setting Value	Function
ADCK0VCR00	0x00000000	Conversion type: normal A/D conversion
		Not select wait time table.
		Not entry DFE.
		Not output virtual channel complete interrupt.
		Physical Ch0/Sub CH0 (AN000)
ADCK0VCR01	0x00000001	Conversion type: normal A/D conversion
		Not select wait time table.
		Not entry DFE.
		Not output virtual channel complete interrupt.
		Physical Ch0/Sub CH1 (AN001)
ADCK0VCR02	0x00000002	Conversion type: normal A/D conversion
		Not select wait time table.
		Not entry DFE.
		Not output virtual channel complete interrupt.
		Physical Ch0/Sub CH2(AN002)
ADCK0ADCR1	0x00	Synchronous suspend
ADCK0ADCR2	0x10	Signed 12bit integer format
ADCK0SGCR0	0x50	Enable ADSTART.
		Multi-Sycle scan mode
		Output “ADI00” in the end of “SG0”.
		Prohibit the trigger input to “SG0”.
ADCK0SGVCPR0	0x0200	Start virtual channel0, end virtual channel 2.
ADCK0SGMCYCR0	0x00	One scan in multi-cycle scan mode.
AIR.DSELR0	0x00000010	DMA Request：INTADCK0I0

Table 3-26 Interrupt Register Setting

Register Name	Setting Value	Function
EIBD70	0x00000000	Bind the interrupt to PE0 (CPU0).
EIC70	0x0040	Refer to table/ Priority level 0

Table 3-27 sDMAC Register

Register Name	Setting Value	Function
DMA0CM_0	0x00001C00	Channel Master SPID Setting SPID=0x1C (Initial Value)
		Supervisor mode
DMA0SAR_0	ADCK0DR00 register address	Transfer source：ADCK0DR00 register
DMA0DAR_0	smp_data_1[0] address	Transfer destination：Array variable for storing A/D conversion value data
DMA0TSR_0	0x00000006	Transfer size：6byte (2byte × 3 times)
DMA0TMR_0	0x00001111	DMA transfer request select allocation：hardware DMA transfer request
		Destination address count direction: fixed
		Source address count direction: increment
		DMA transfer transaction size: 2byte
		DMA source transaction size: 2byte
DMA0RS_0	0x000300B2	Number of transfers per hardware request: 3 times
		Hardware DMA transfer factor selection: group 1-178 (A/D conversion complete interrupt (ADI00))
DMACSEL0_11	0x00000010	DMA transfer request group: ADCK0I0(group 1-178)
DMA0SIAI_0	Value on right	Inner address increment value: smp_data_2[0] address - smp_data_1[0] address
DMA0SGCR_0	0x80020000	Scatter function: enable
		Number of repetitions of internal loop: 2 times
DMA0CHFCR_0	0x0000320F	Clear each flag.
DMA0OR	0x0001	Enable DMA transfer.
DMA0CHCR_0	0x0003	Enable transfer complete interrupt.
		Enable channel operation.

- Operation Flow
  Flowchart in this operation example is shown below.

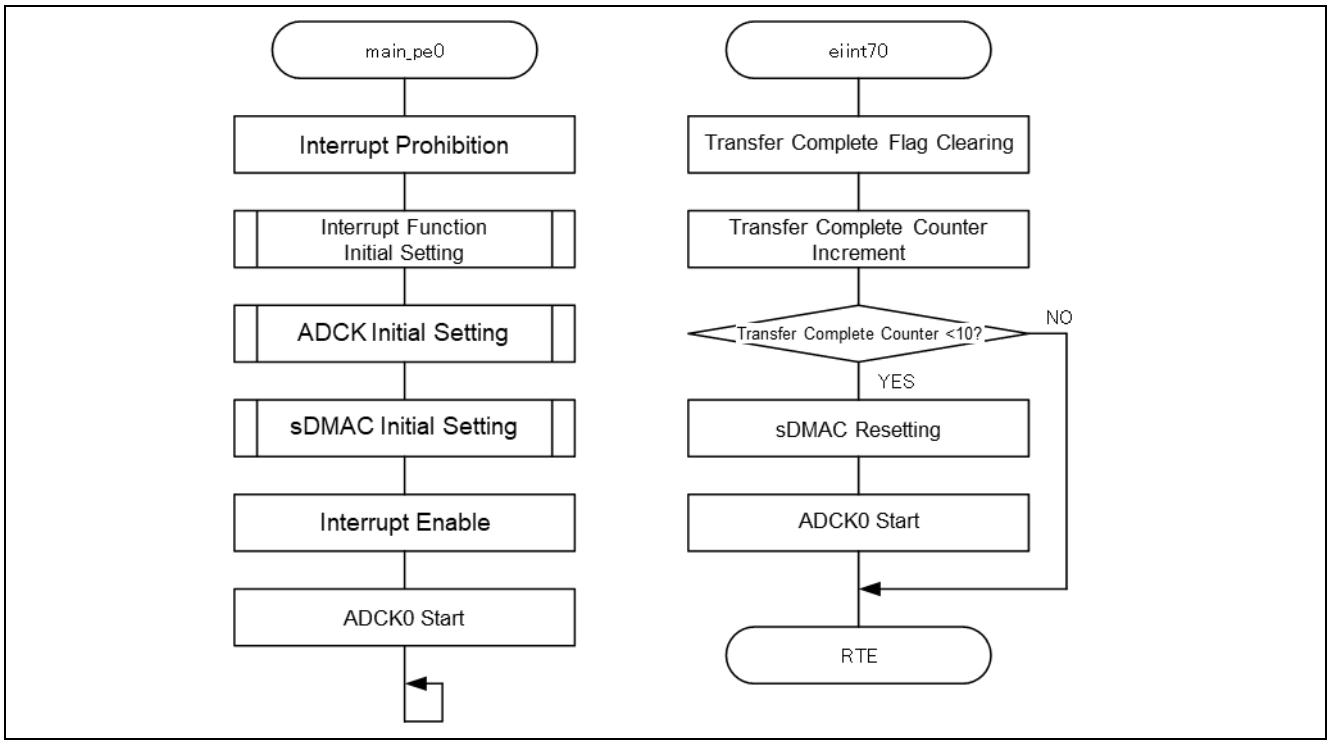

Figure 3-33 Flowchart

3.9 Sequential Scan of Optional Channel using DMA Transfer (Gather Function)

3.9.1 Specification Overview

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

Allocate 3 virtual channels to scan group 0 (SG0), and scan by multi-scan mode. DMA is started in scan end interrupt (ADI00), and the conversion values of AN000, AN001, and AN002 are stored to the array variable.

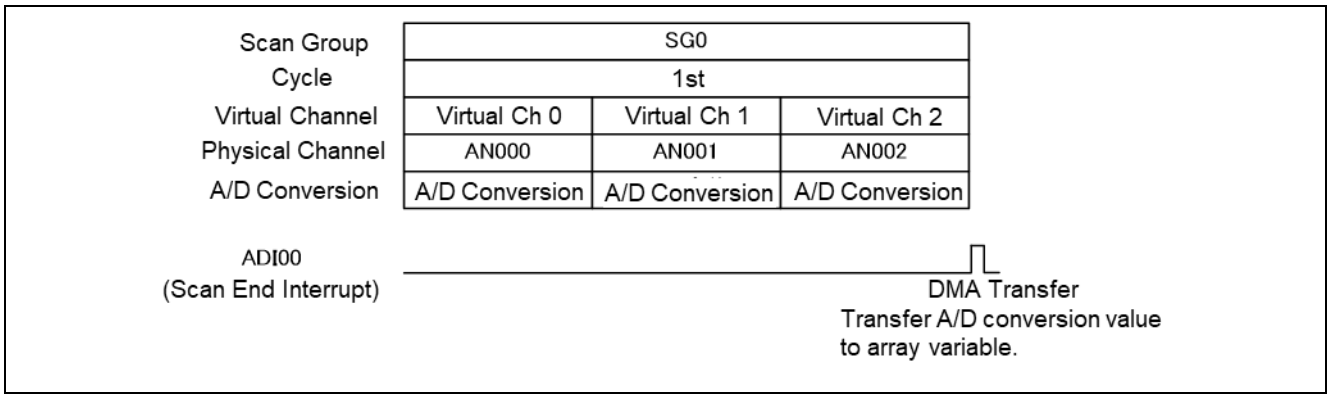

Figure 3-34 Operation Example using DMA Transfer

3.9.2 Use Function

The functions used in this operation example are shown below.

- A/D Convertor（ADCK0）
- DMA（sDMAC0）

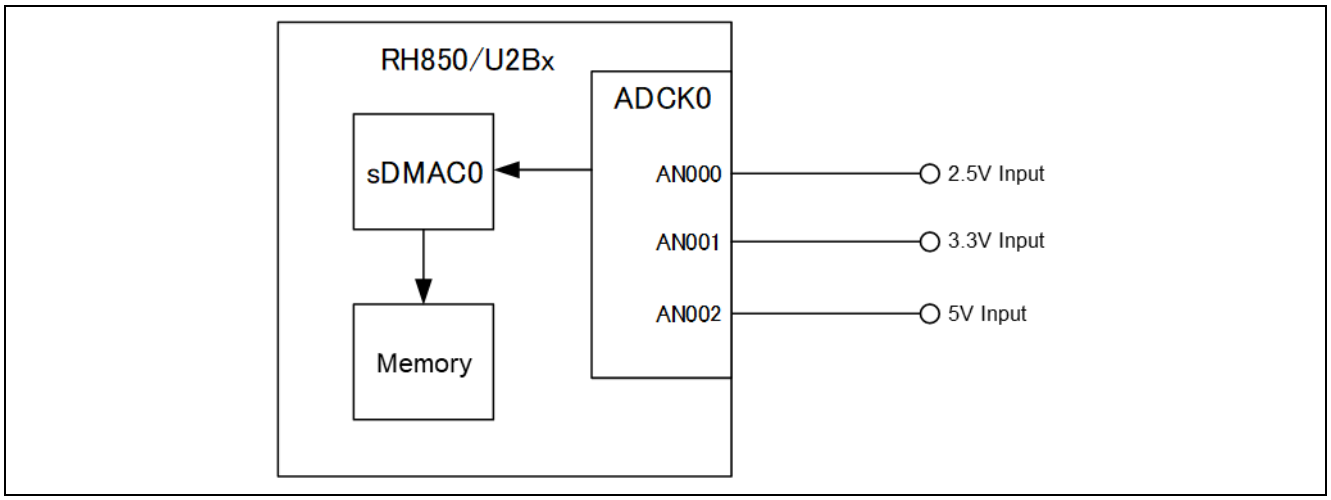

Figure 3-35 System Configuration

3.9.3 Explanation for Operation Example

In this operation example, perform normal A/D conversion in multi-scan mode using AN000, AN001, and AN002 of ADCK0 module. Also, store the A/D conversion value of AN000 and AN002 to the array variable using DAM conversion (gather function).

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

For sDMAC, set the scan end interrupt (ADI00) to start trigger, enable the scatter function. Figure 3-31 shows the overview of gather function. In gather function, A/D conversion values of 2 channels can be transferred to each array variable with one channel of sDMAC.

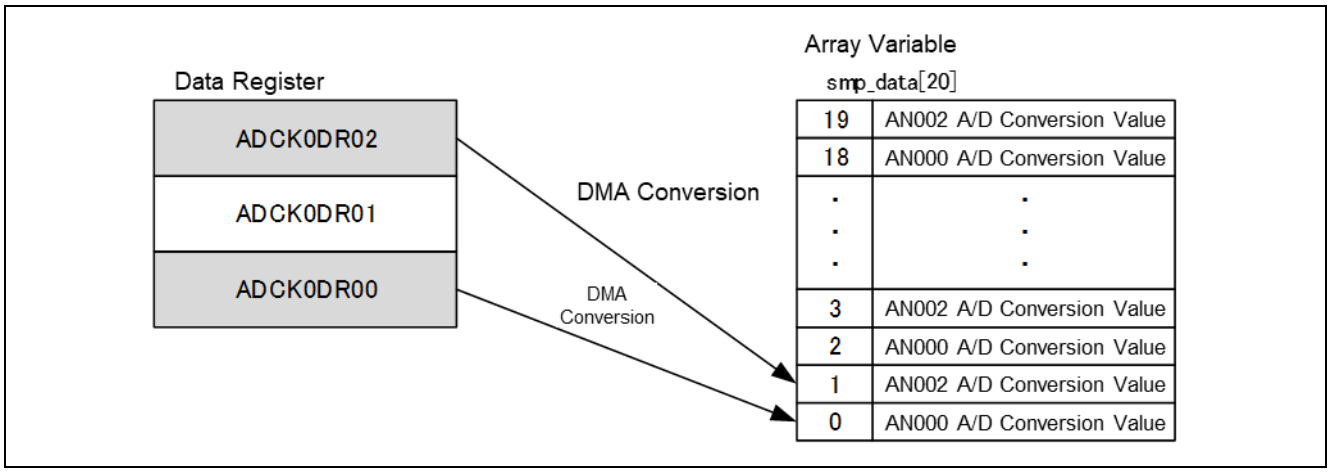

Figure 3-36 Gather Function Overview

Start A/D conversion with software trigger ADSTART. DMA is started by the scan end interrupt (ADI00), and A/D conversion value is stored to the array variable.

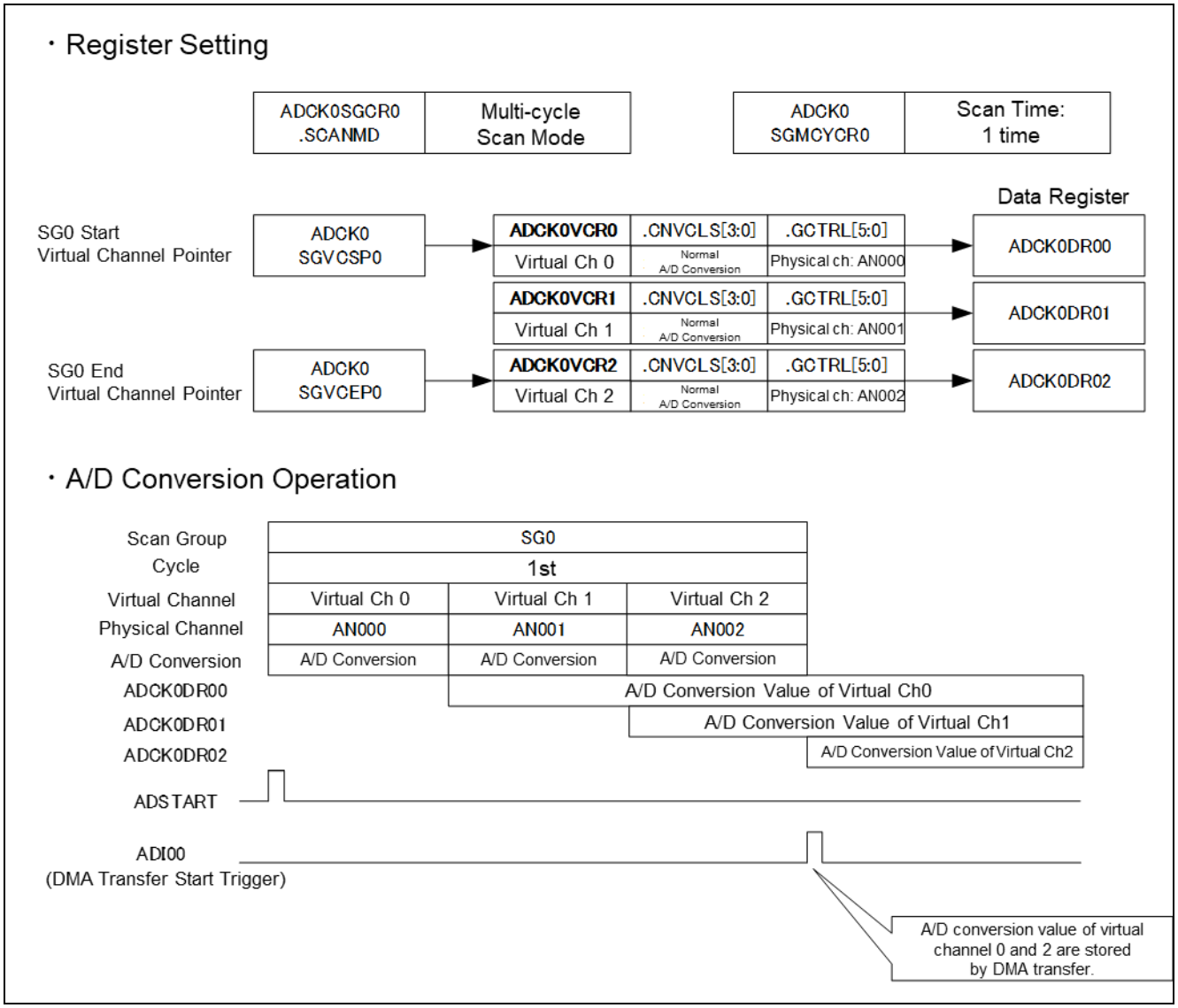

Figure 3-37 Operation Example using DMA Transfer

3.9.4 Software Explanation

- Module Explanation

Module list in this operation example is shown below.

Table 3-28 Module List

Module Name	Label Name	Function
Main routine	main_pe0	Perform the various settings and application start.
ADCK initialize routine	ADCK_init	Initialize the ADCK.
DTS initialize routine	sdmac_init	Initialize the sDMAC.
Interrupt initialize routine	intc_init	Initialize the sDMAC interrupt.
sDMAC transfer complete interrupt	eiint70	Perform sDMAC resetting and A/D conversion restarting by transfer complete interrupt processing.

- Register Setting

The register setting in this operation example is shown below.

Table 3-29 ADCK Register Setting

Register Name	Setting Value	Function
ADCK0VCR00	0x00000000	Conversion type: normal A/D conversion
		Not select wait time table.
		Not entry DFE.
		Not output virtual channel complete interrupt.
		Physical Ch0/Sub CH0 (AN000)
ADCK0VCR01	0x00000001	Conversion type: normal A/D conversion
		Not select wait time table.
		Not entry DFE.
		Not output virtual channel complete interrupt.
		Physical Ch0/Sub CH1 (AN001)
ADCK0VCR02	0x00000002	Conversion type: normal A/D conversion
		Not select wait time table.
		Not entry DFE.
		Not output virtual channel complete interrupt.
		Physical Ch0/Sub CH2(AN002)
ADCK0ADCR1	0x00	Synchronous suspend
ADCK0ADCR2	0x10	Signed 12bit integer format
ADCK0SGCR0	0x50	Enable ADSTART.
		Multi-Sycle scan mode
		Output “ADI00” in the end of “SG0”.
		Prohibit the trigger input to “SG0”.
ADCK0SGVCPR0	0x0200	Start virtual channel0, end virtual channel 2.
ADCK0SGMCYCR0	0x00	One scan in multi-cycle scan mode.
AIR.DSELR0	0x00000010	DMA Request：INTADCK0I0

Table 3-30 Interrupt Register Setting

Register Name	Setting Value	Function
EIBD70	0x00000000	Bind the interrupt to PE0 (CPU0).
EIC70	0x0040	Refer to table/ Priority level 0

Table 3-31 sDMAC Register

Register Name	Setting Value	Function
DMA0CM_0	0x00001C00	Channel Master SPID Setting SPID=0x1C (Initial Value)
		Supervisor mode
DMA0SAR_0	ADCK0DR00 register Address	Transfer source：ADCK0DR00 register
DMA0DAR_0	smp_data[0] address	Transfer destination：Array variable for storing A/D conversion value data
DMA0TSR_0	0x00000004	Transfer size：4byte (2byte × 2 times)
DMA0TMR_0	0x00001411	DMA transfer request select allocation：hardware DMA transfer request
		Destination address count direction: increment
		Source address count direction: fixed
		DMA transfer transaction size: 2byte
		DMA source transaction size: 2byte
DMA0RS_0	0x000200B2	Number of transfers per hardware request: 2 times
		Hardware DMA transfer factor selection: group 1-178 (A/D conversion complete interrupt (ADI00))
DMACSEL0_11	0x00000010	DMA transfer request group: ADCK0I0(group 1-178)
DMA0GIAI_0	Value on right	Inner address increment value: ADCK0DR02 address – ADCK0DR00 address
DMA0SGCR_0	0x00008001	Gather function: enable
		Number of repetitions of internal loop: 1 time
DMA0CHFCR_0	0x0000320F	Clear each flag.
DMA0OR	0x0001	Enable DMA transfer.
DMA0CHCR_0	0x0003	Enable transfer complete interrupt.
		Enable channel operation.

- Operation Flow
  Flowchart in this operation example is shown below.

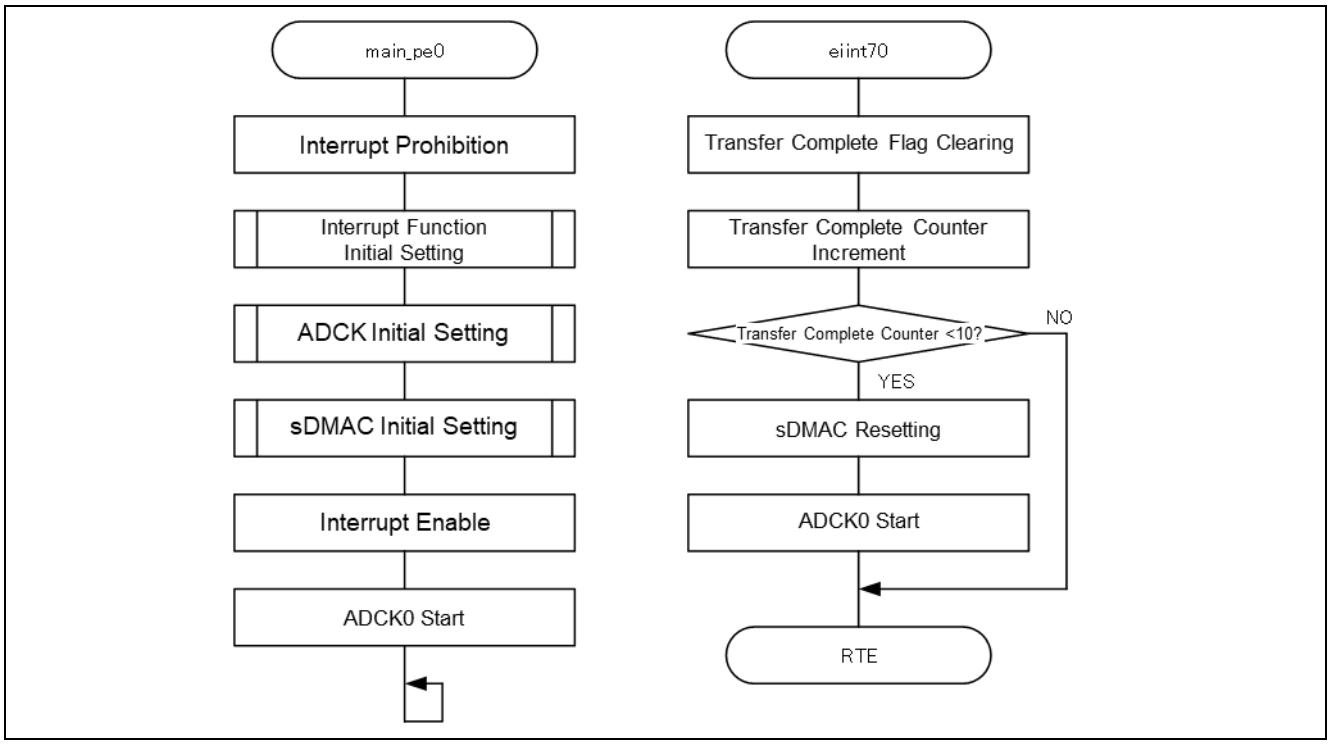

Figure 3-38 Flowchart

3.10 Wait Time Insertion Function between Channels

3.10.1 Specification Overview

This section explains the wait time incretion function between channels.

Allocate the two virtual channels (AN000 and An001) to the scan group 0 (SG0), and perform a scanning in the multi-scan mode. At the end of the scan group, the converted values of AN000 and AN001 are stored in variables, and the operation ends. Then, the wait time can be inserted between virtual channels in normal A/D conversion.

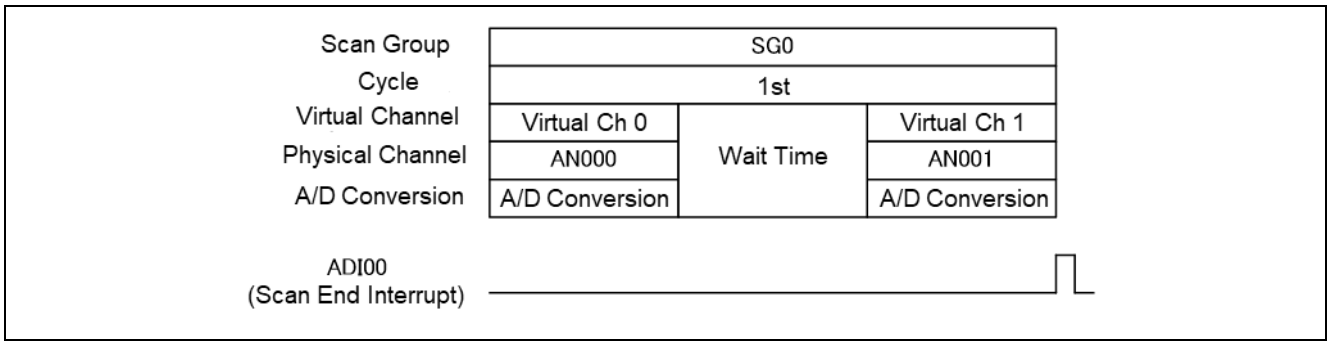

Figure 3-39 Normal A/D Conversion (multi-scan) Operation

3.10.2 Use Function

The functions used in this operation example are shown below.

- A/D Converter（ADCK0）
- Port

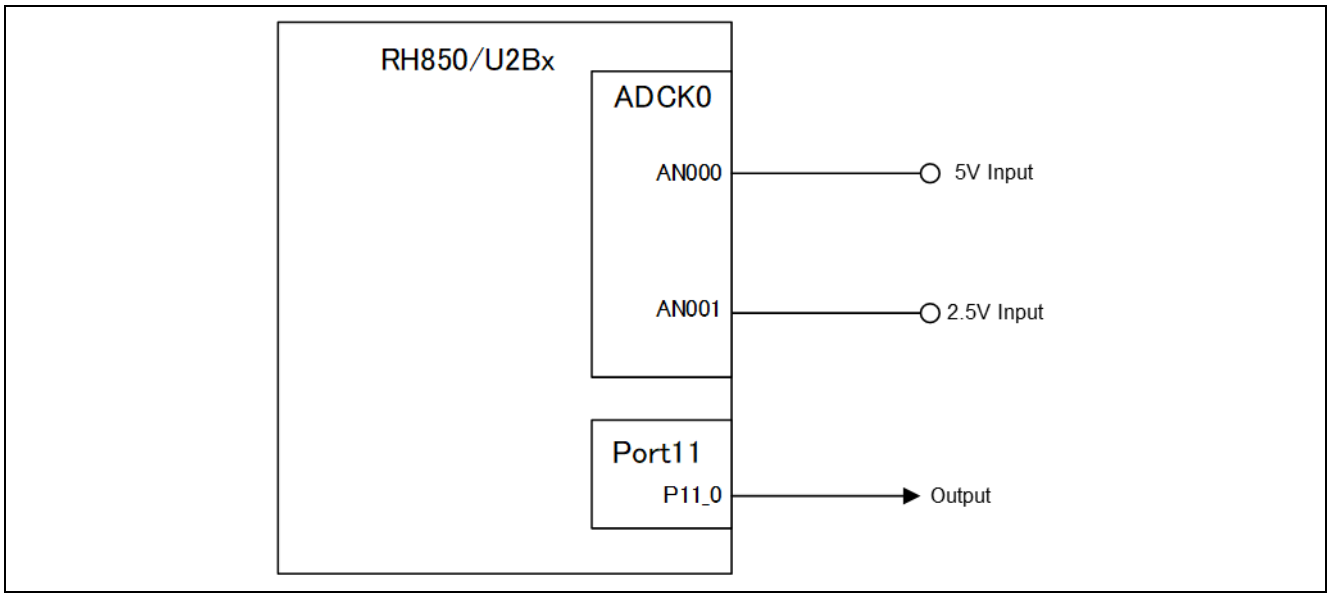

Figure 3-40 System Configuration

3.10.3 Explanation for Operation Example

In this operation example, A / D conversion is normally performed by one scan in the multi-scan mode using AN000 and AN001 of the ADCK0 module.

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

For analog signals, input 5.0V to AN000 and 2.5v to AN001.

Set 100 us to the wait time between virtual channels.

Start the soft trigger ADSTART, and perform A/D conversion for AN001 after performed A/D conversion for AN000. Enable the scan end interrupt ADI00 (at the end of the scan group), store each A/D conversion result to the variable in the interrupt processing, and the process ends.

In this operation example, use the port (P11_0) as the output port to check that weights are inserted between virtual channels.

At the start of A/D conversion, set High to the output of P11_0, and set Low by the scan end interrupt ADI00.

It is possible to confirm that the weights is inserted between the virtual channels in the high period of P11_0.

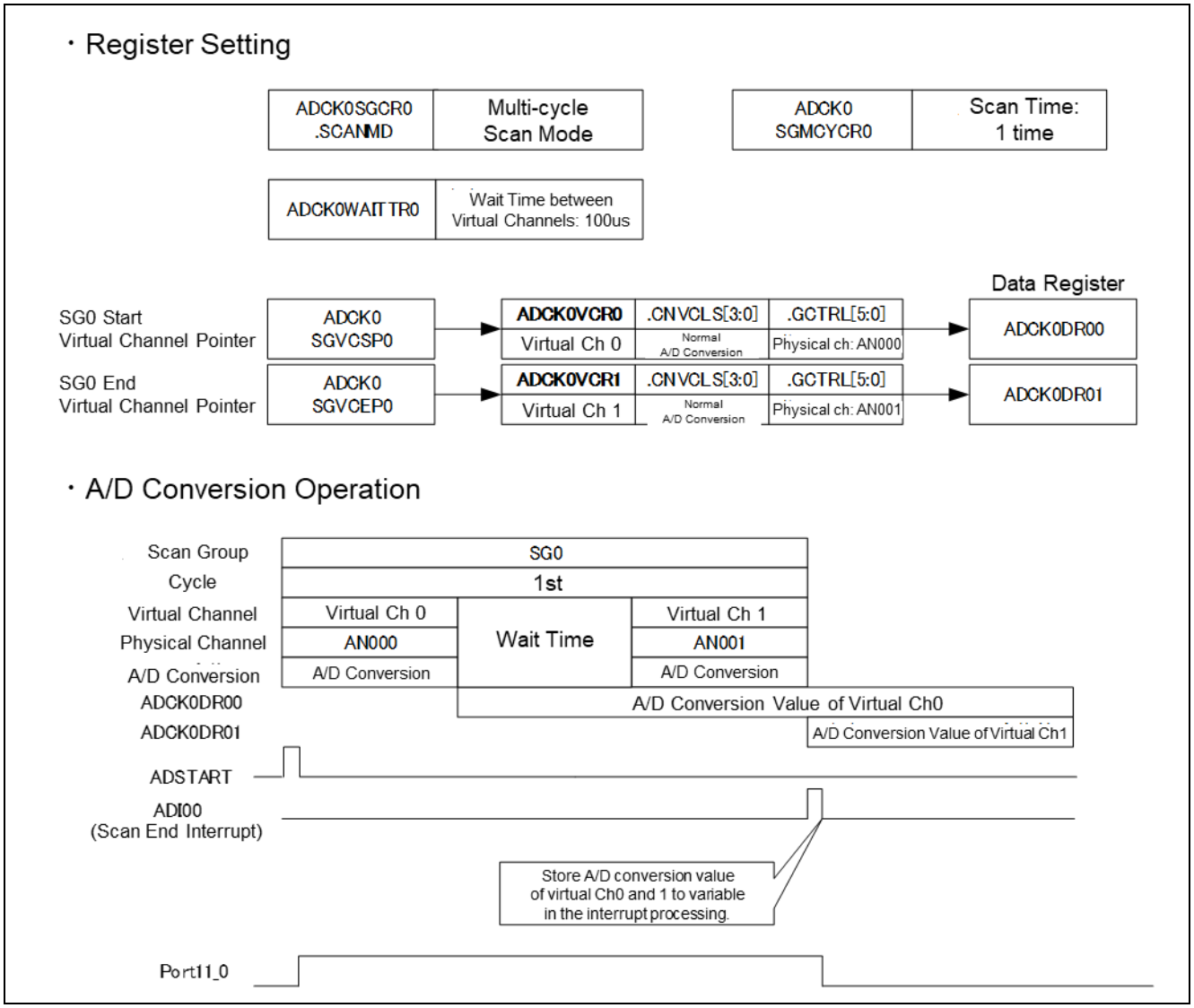

Figure 3-41 Normal A/D Conversion (Wait Insertion between Virtual Channels) Operation Example

3.10.4 Software Explanation

- Register Setting

Module Explanation

The module list in this operation example is shown below.

Table 3-32 Module List

Module Name	Label Name	Function
Main routine	main_pe0	Perform the various settings and application start.
Port initialize routine	port_init	Initialize the port.
ADCK initialize routine	ADCK_init	Initialize the ADCK.
Interrupt initialize routine	intc_init	Initialize the ADCK interrupt.
ADCK interrupt process routine	eiint441	Store the A/D conversion result in variable by the virtual scan group end interrupt processing.

- Register Setting

The register setting for various function in this operation example is shown below.

Table 3-33 ADCK Register Setting

Register Name	Setting Value	Function
ADCK0VCR00	0x00000000	Conversion type: normal A/D conversion
		Not select the wait time table.
		No DFE entry.
		Not output the virtual channel end interrupt.
		Physical Ch0/Sub CH0 (AN000)
ADCK0VCR01	0x01000001	Conversion type: normal A/D conversion
		Not select the wait time table.
		Not DFE entry.
		Not output the virtual channel end interrupt.
		Physical Ch0/Sub CH1 (AN001)
ADCK0ADCR1	0x00	Synchronous suspend
ADCK0ADCR2	0x10	Signed 12bit integer format
ADCK0SGCR0	0x50	Enable ADSTART.
		Multi-cycle scan mode
		Output “ADI00” in the end of “SG0”.
		Prohibit the trigger input to “SG0”.
ADCK0SGVCPR0	0x0100	Start virtual Ch0, End virtual Ch1
ADCK0SGMCYCR0	0x00	Conversion type: normal A/D conversion
ADCK0WAITTR0	0x0FA0	100us (Wait time = Setting value×4 ADCLK)

Table 3-34 Interrupt Register Setting

Register Name	Setting Value	Function
EIBD441	0x00000000	Bind the interrupt to PE0 (CPU0).
EIC441	0x0040	Refer to table/ Priority level 0

- Operation Flow
  The flow chart in this operation example is shown below.

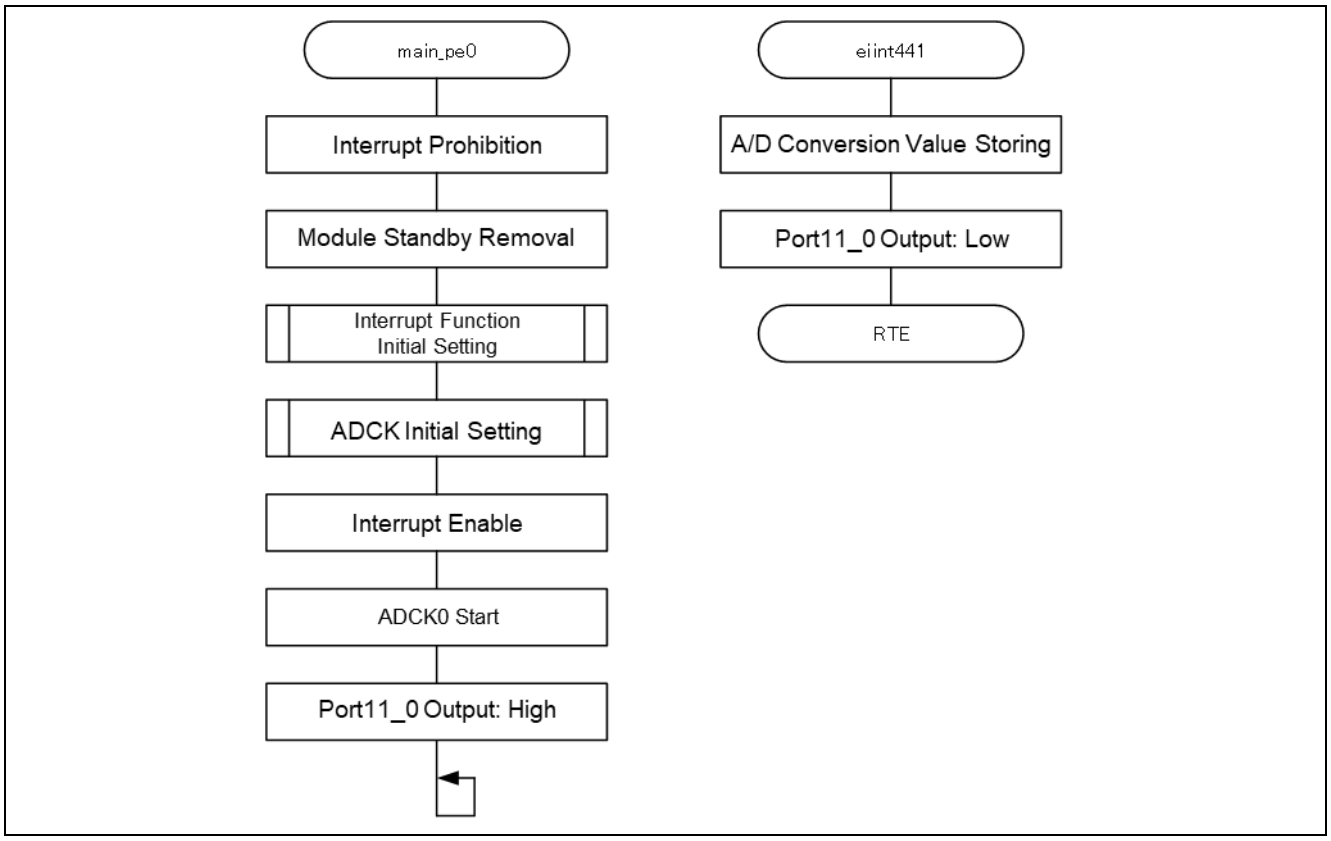

Figure 3-42 Flowchart

3.11 Operation Example using ADC VMON Secondary Error Generator (AVSEG)

3.11.1 Specification Overview

This section describes how to detect an error in the power supply voltage (VCC, E0VCC, VDD) and notify the ECM. This operation example describes how to monitor the E0VCC voltage.

Allocate a virtual channel (VMON_E0VCC) to the scan mode 0 (SG0), and perform scanning in the multi-scan mode. Store VMON_E0VCC conversion value in the variable at the end of the scan group. After that, the same operation is repeated continuously.

Then, if the A/D conversion value exceeds the set upper and lower limit values, an error is notified to ECM and the EMC interrupt is generated.

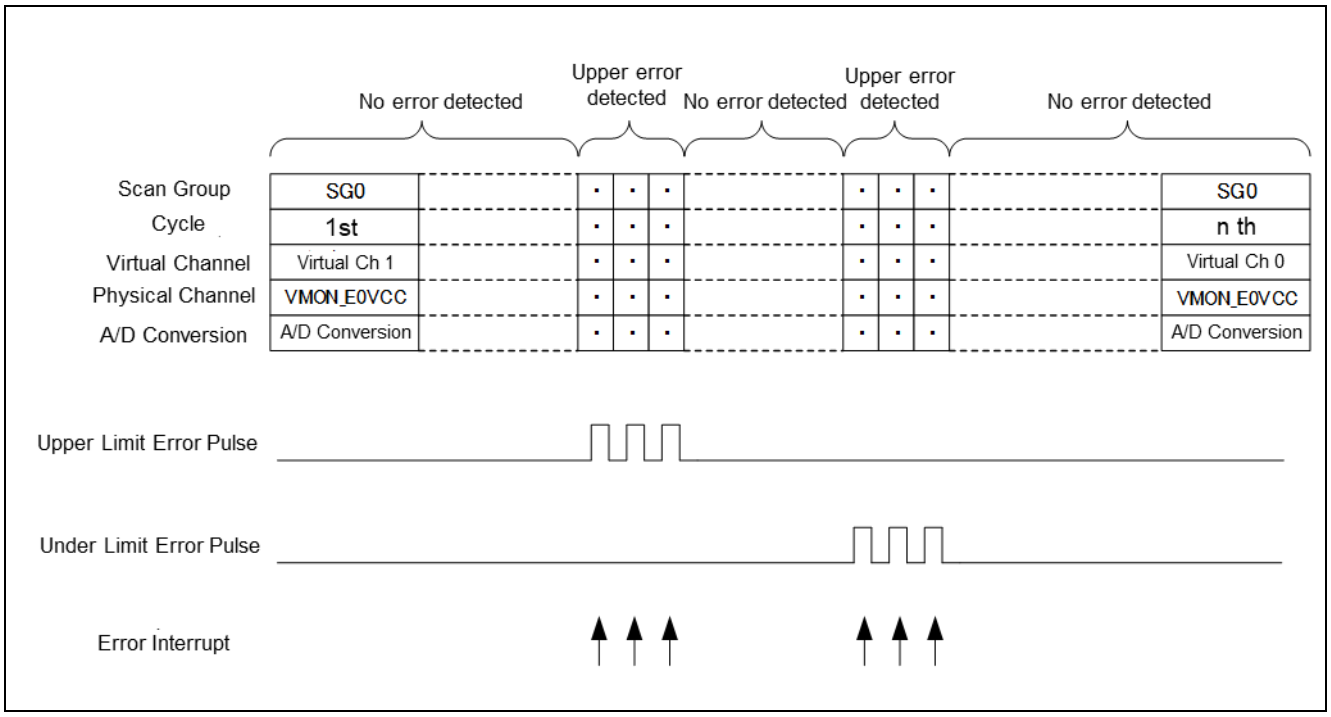

Figure 3-43 Operation Example of Power Supply Voltage Monitoring

3.11.2 Use Function

The functions used in this operation example are shown below.

- A/D convertor（ADCK0）
- ADC VMON secondary error generator（AVSEG）

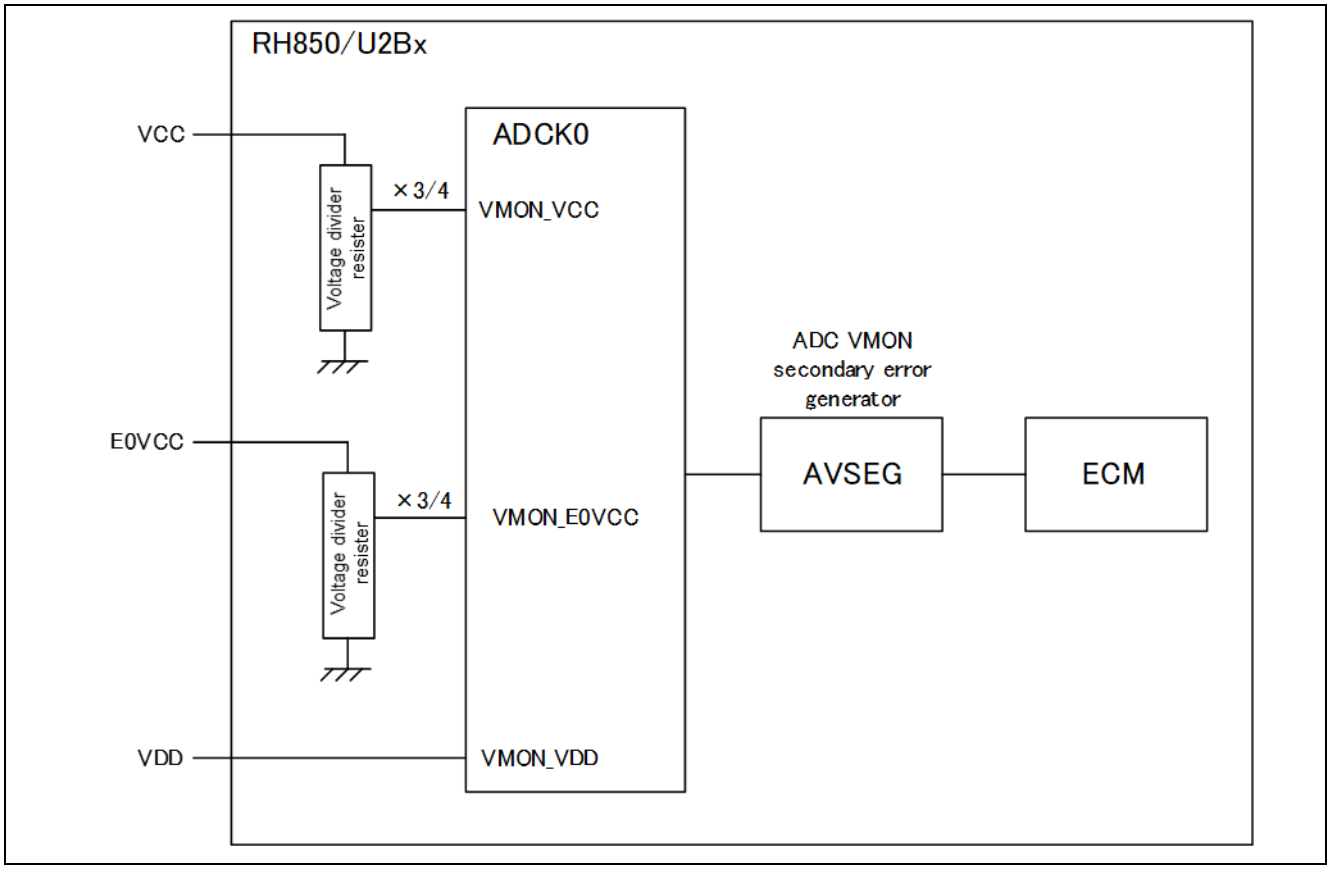

Figure 3-44 System Configuration

3.11.3 Explanation for Operation Example

In this operation example, A/D conversion is normally performed by the multi-scan mode using VMON_EVCC of the ADCK0 module.

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

Enable the scan end interrupt ADI00 (in the end of the scan group).

Set 3.673V to the upper limit and 3.485V to the lower limit. * This setting is a reference value.

The E0VCC power supply is divided into 3/4 inside the RH850 / U2Bx and input to the ADCK.

Please set the upper and lower limits in consideration of the above.

Enable the AVSEG noise filter and set the recovery counter to “1” and the error counter to “3”. In this setting, if the A / D conversion value exceeds the upper and lower limit values three or more times, an upper and lower limit error is outputted.

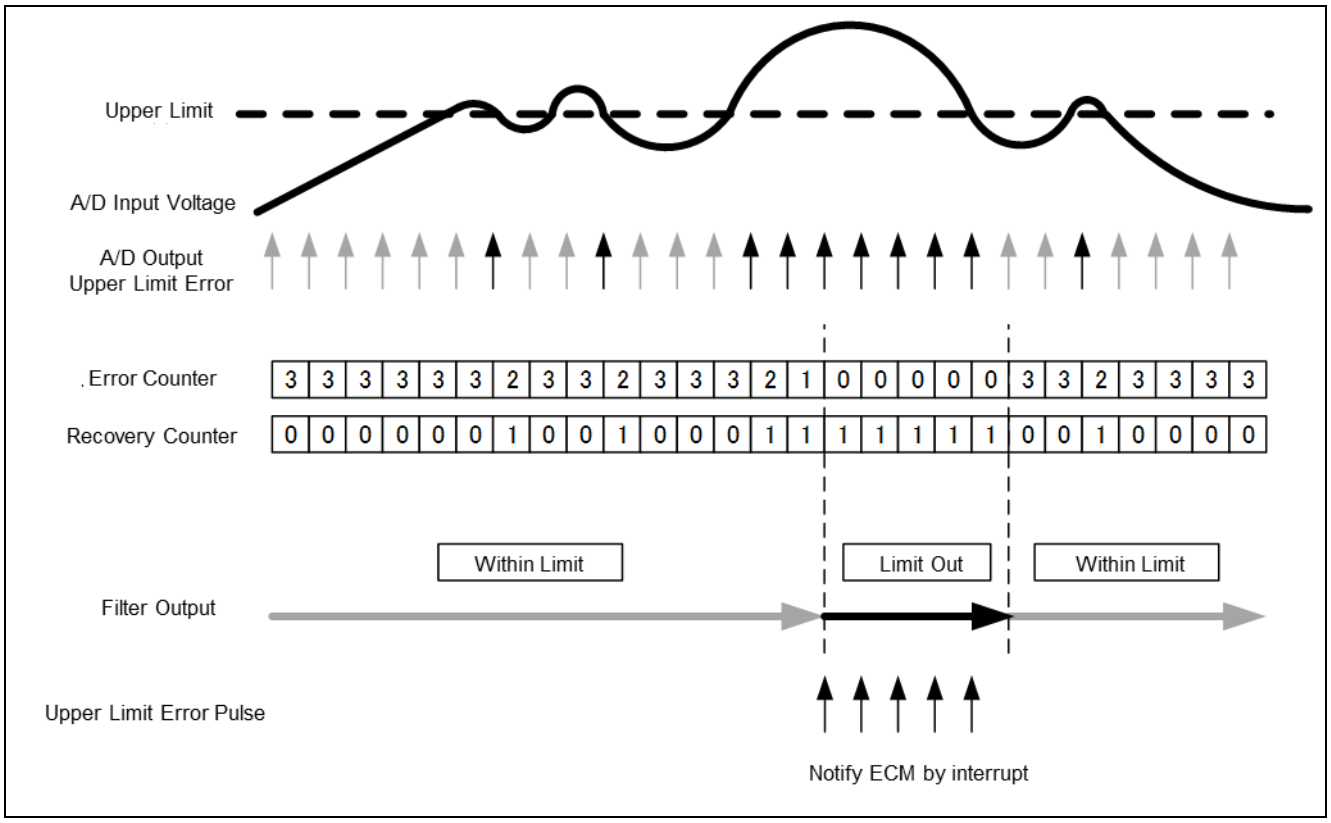

Figure 3-45 Noize Filter Operation

Start with the soft trigger ADSTART and perform A/D conversion of VMON_E0VCC.

The scan end interrupt ADI00 stores the A/D conversion result in the variable.

At this time, if the A / D conversion value exceeds the set upper and lower limit values, an error is notified to ECM and an ECM interrupt is generated.

3.11.4 Software Explanation

- Register Setting

Module Explanation

Module list in this operation example is shown below.

Table 3-35 Module List

Module Name	Label Name	Function
Main routine	main_pe0	Perform the various settings and application start.
Interrupt initialize routine	intc_init	Initialize the ADCK and ECM interrupt.
ECM initialize routine	ecm_init	Initialize the ECM.
ADCK initialize routine	ADCK_init	Initialize the ADCK.
AVSEG initialize routine	avseg_init	Perform the noise filter enable and filter setting.
ADCK interrupt processing routine	eiint441	Store the A/D conversion result to the variable in virtual scan group end interrupt processing.
Voltage monitors error interrupt routine	eiint8	Increment the error counter in ECM interrupt processing.

- Register Setting

The register settings for each function in this operation example are shown below.

Table 3-36 Interrupt Register Setting

Register Name	Setting Value	Function
EIBD441	0x00000000	Bind the interrupt to PE0 (CPU0).
EIC441	0x0040	Refer to table/ Priority level 0
EIBD8	0x00000000	Bind the interrupt to PE0 (CPU0).
EIC8	0x0040	Refer to table/ Priority level 0

Table 3-37 ECM Register Setting

Register Name	Setting Value	Function
ECMINCFG0_3	0x00000008	EMMIE 99 : E0VCC upper/lower limit error interrupt enable

Table 3-38 ADCK Register Setting

Register Name	Setting Value	Function
ADCK0VCR00	0x1000001D	Enable the virtual channel upper/lower limit exceeded notification.
		Select the virtual channel upper/lower limit table register 0.
		Conversion type: normal A/D conversion
		Not select wait time table.
		Not entry DFE.
		Not output virtual channel complete interrupt.
		Physical Ch7/Sub CH1 (VMON_E0VCC)
ADCK0VCULLMTBR0	0x5E005932	Virtual channel upper/lower limit table register 0 Upper limit: 3.673V (0x5E00) Lower limit: 3.485V（0x5932）
ADCK0SGCR0	0x70	Enable ADSTART.
		Multiple-cycle Scan Mode
		Output “ADI00” in the end of “SG0”.
		Prohibit the trigger input to “SG0”.
ADCK0SGVCPR0	0x0000	Start virtual channel0, end virtual channel 0.
ADCK0ADCR2	0x00	Signed 12bit integer format
ADCK0ADCR1	0x00	Synchronous suspend
ADCK0VMONVDCR1	0x01	Turn-on the voltage divider register, Turn-off the pulldown.
ADCK0VMONVDCR2	0x01	

Table 3-39 AVSEG Register Setting

Register Name	Setting Value	Function
AVSEGEVCCCHSCR	0x00000000	Select the virtual channel 0.
AVSEGEVCCCNTCR	0x10000103	Enable the filter.
		Recovery Counter Setting (NRMCNT)：1
		Error Counter Setting(ERRCNT)：3

- Operation Flow
  The flow chart in this operation example is shown below.

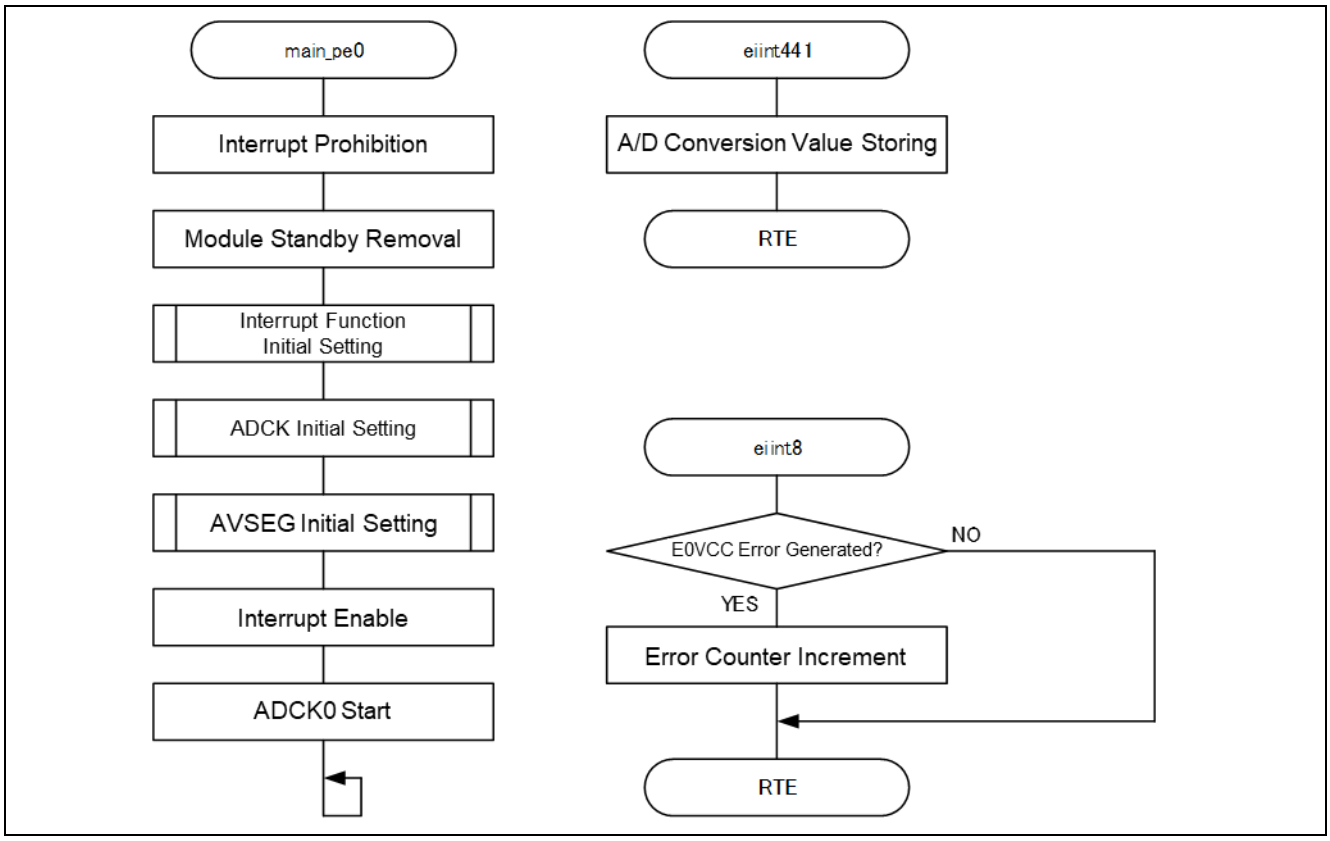

Figure 3-46 Flowchart

3.12 Operation Example using ADC Boundary Flag Generator (ABFG)

3.12.1 Specification Overview

This section describes the usage of boundary flag

Allocate one virtual channel (AN000) to scan group 0 (SG0) and set continuous scan mode. The conversion value of AN000 is stored to the variable in the scan group end.

At this time, if the A / D conversion value exceeds the set upper and lower limit values, the boundary flag changes and a boundary flag pulse is generated.

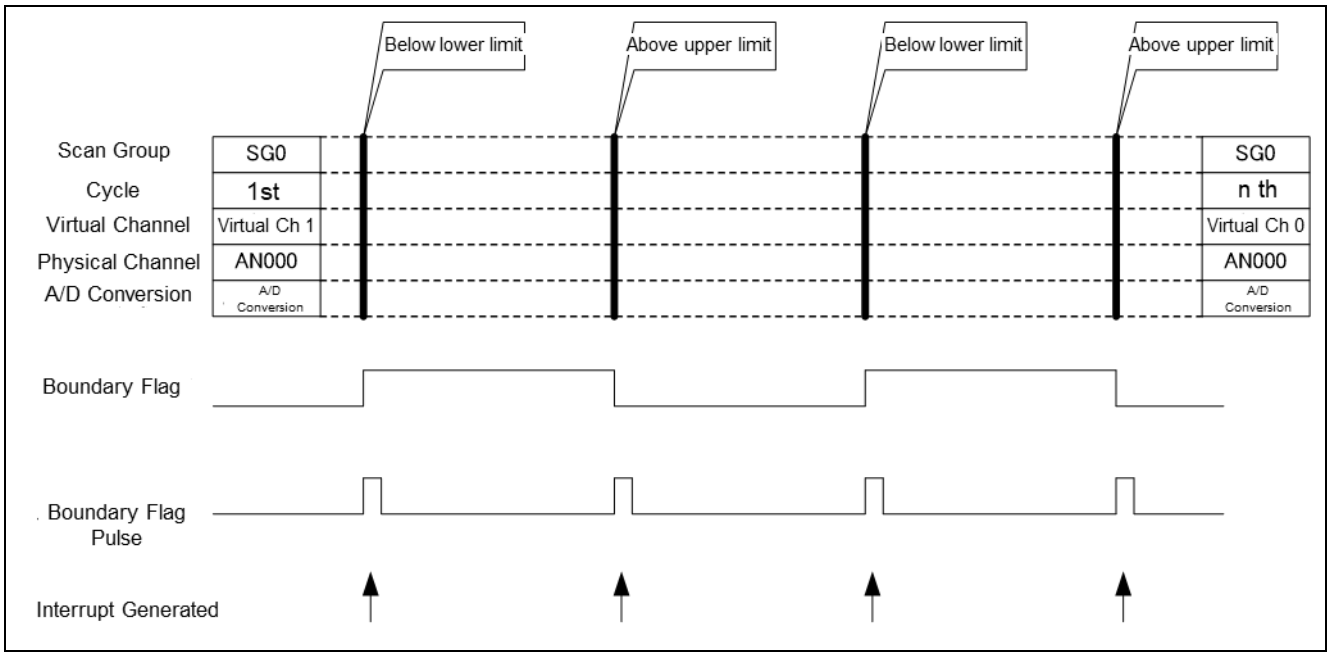

Figure 3-47 Operation Example of Boundary Flag Change and Boundary Flag Pulse Generation

3.12.2 Use Function

The functions used in this operation example are shown below.

- A/D Convertor（ADCK0）
- ADC Boundary Flag Generator（ABFG）

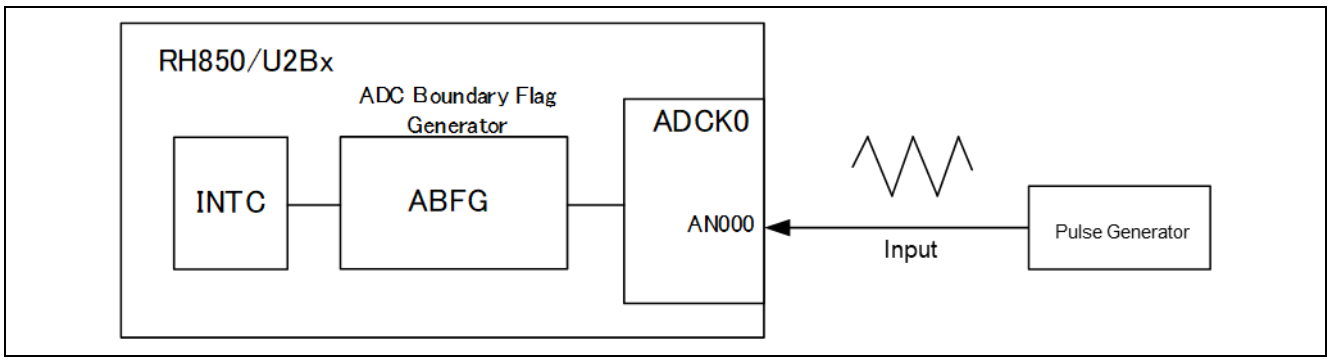

Figure 3-48 System Configuration

3.12.3 Explanation for Operation Example

In this operation example, A/D conversion is normally performed by the multi-scan mode using AN000 of the ADCK0 module.

Allocate a virtual channel 0 (AN000) to the scan group 0 (SG0).

Input the triangle wave with the amplitude of 0 to 5[V] and the period of 100Hz to AN000.

Set 4V to the upper limit and 1V to the lower limit. * This setting is a reference value.

For the boundary flag of ABFG, Set the A/D conversion value to High when it is below the lower limit and Low when it is above the upper limit. Also, the boundary flag pulse is set to be generated when the boundary flag changes (rising / falling).

Enable the AVSEG noise filter and set the recovery counter to “1” and the error counter to “3”.

In this setting, if the A / D conversion value exceeds the upper and lower limit values three or more times, an upper and lower limit error is outputted.

(About the noise filter operation, refer to “3.11.3 Explanation for Operation Example”)

Enables the boundary flag pulse interrupt caused by the boundary flag pulse.

Start A/D conversion of AN000 with the soft trigger ADSTART and the scan end interrupt (ADI00) stores the conversion value of AN000 in the variable.

At this time, if the A / D conversion value exceeds the upper and lower limit values, the boundary flag changes. The boundary flag pulse interrupt is generated by a boundary flag edge change, and the edge counter is incremented.

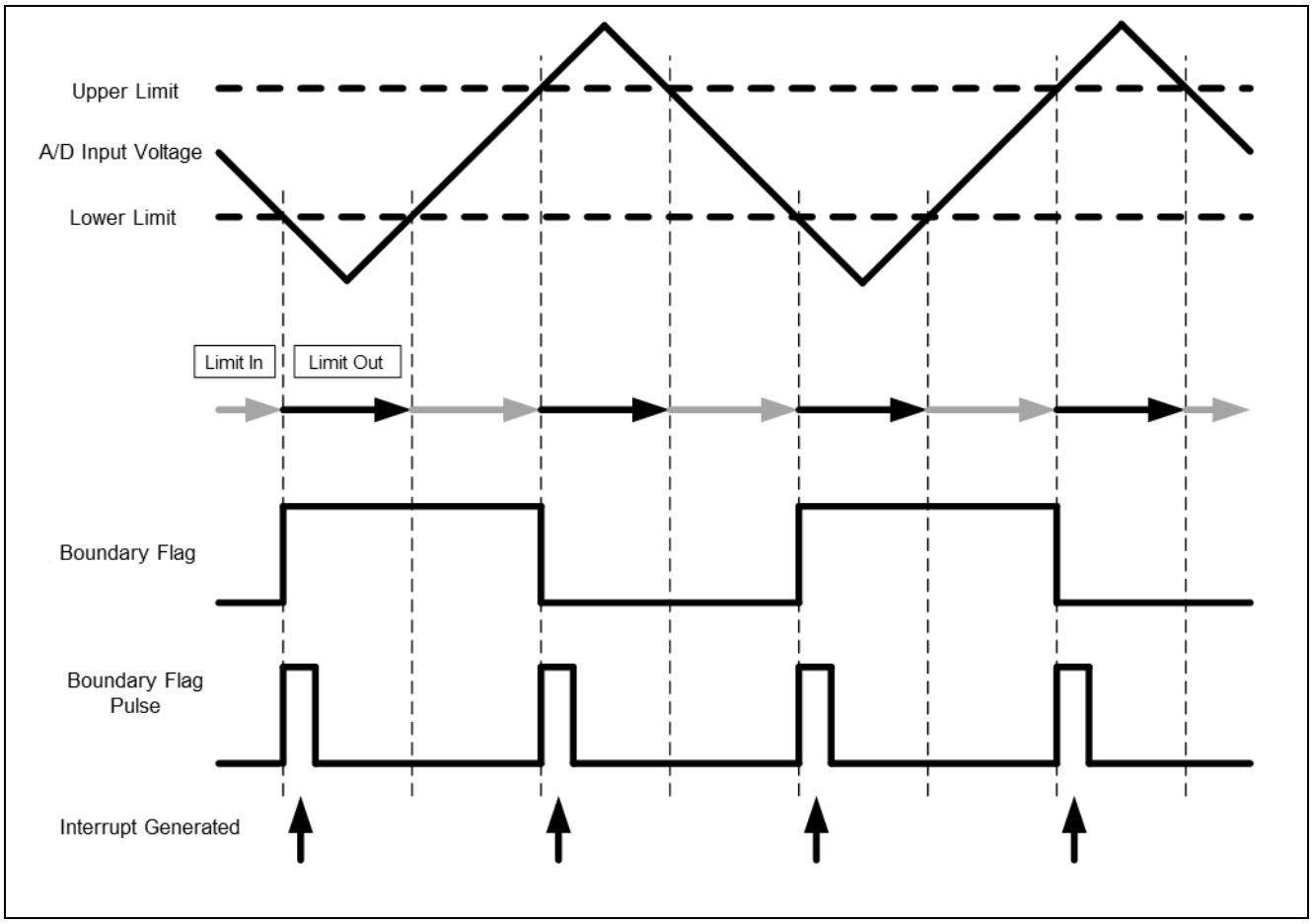

Figure 3-49 Operation Example of Boundary Flag

3.12.4 Software Explanation

- Register Setting

Module Explanation

Module list in this operation example is shown below.

Table 3-40 Module List

Module Name	Label Name	Function
Main routine	main_pe0	Perform the various settings and application start.
Interrupt initialize routine	intc_init	Initialize the ADCK and boundary flag pulse interrupt.
ADCK initialize routine	ADCK_init	Initialize the ADCK.
ABFG initialize routine	abfg_init	Initialize the ABFG.
ADCK interrupt processing routine	eiint441	Store the A/D conversion result to the variable in virtual scan group end interrupt processing.
Boundary flag pulse interrupt processing routine	eiint492	Increment the edge counter in boundary flag pulse interrupt processing.

- Register Setting
- The register settings for each function in this operation example are shown below.

Table 3-41 Interrupt Register Setting

Register Name	Setting Value	Function
EIBD441	0x00000000	Bind the interrupt to PE0 (CPU0).
EIC441	0x0040	Refer to table/ Priority level 0
EIBD492	0x00000000	Bind the interrupt to PE0 (CPU0).
EIC492	0x0040	Refer to table/ Priority level 0

Table 3-42 ADCK Register Setting

Register Name	Setting Value	Function
ADCK0VCR00	0x10000000	Enable the virtual channel upper/lower limit exceeded notification.
		Select the virtual channel upper/lower limit table register 0.
		Conversion type: normal A/D conversion
		Not select wait time table.
		Not entry DFE.
		Not output virtual channel complete interrupt.
		Physical Ch0/Sub CH0 (AN000)
ADCK0SGCR0	0x70	Enable ADSTART.
		Multiple-cycle Scan Mode
		Output “ADI00” in the end of “SG0”.
		Prohibit the trigger input to “SG0”.
ADCK0SGVCPR0	0x0000	Start virtual channel0, end virtual channel 0.
ADCK0ADCR2	0x00	Signed 12bit integer format
ADCK0ADCR1	0x00	Synchronous suspend
ADCK0VCULLMTBR0	0x66661999	Virtual channel upper/lower limit table register 0 Upper limit：4V (0x6666), Lower limit：1V (0x1999)

Table 3-43 ABFG Register Name

Register Name	Setting Value	Function
ABFG0 BFGCR	0x01030000	Boundary flag toggle setting (BTGC) : Low when the upper limit is exceeded, high when the lower limit is exceeded.
		Boundary flag pulse generation (BPGC) : Boundary flag rising/falling
		Input channel : virtual channel 0
ABFG0CNTCR	0x10000103	Enable the filter.
		Recovery counter setting (NRMCNT) : 1
		Error counter setting (ERRCNT) : 3

- Operation Flow

  The flow chart in this operation example is shown below.

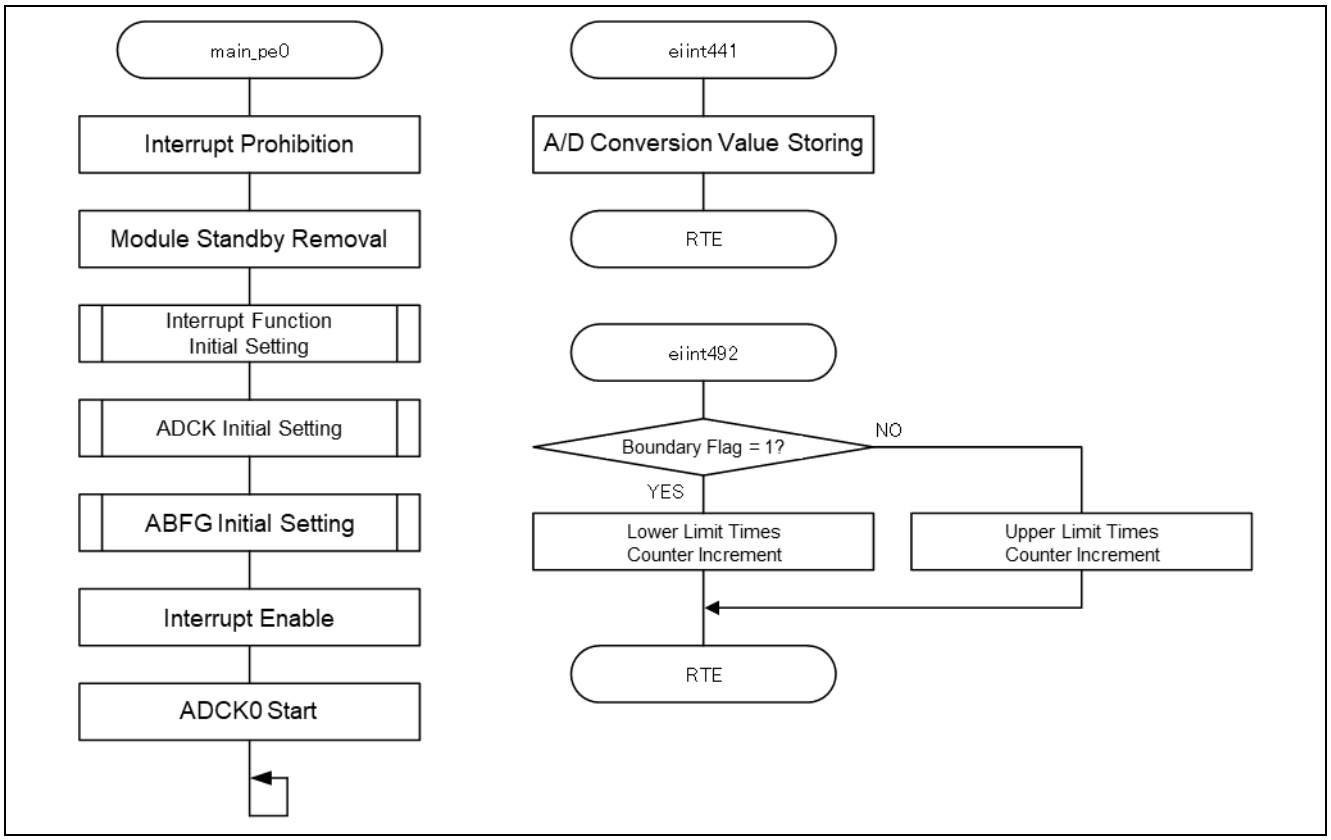

Figure 3-50 Flowchart

3.13 Operation Example using Virtual Port

3.13.1 Specification Overview

This section describes the usage of the virtual port.

Allocate a virtual channel (AN000) to the scan group 0 (SG0) and scan in the multi-scan mode. The conversion value of AN000 is stored to the variable in the scan group end.

At this time, if the A / D conversion value exceeds the set upper and lower limit values, the boundary flag is changed. This boundary flag is input to timer A of ATU via the virtual port to generate the input capture interrupt.

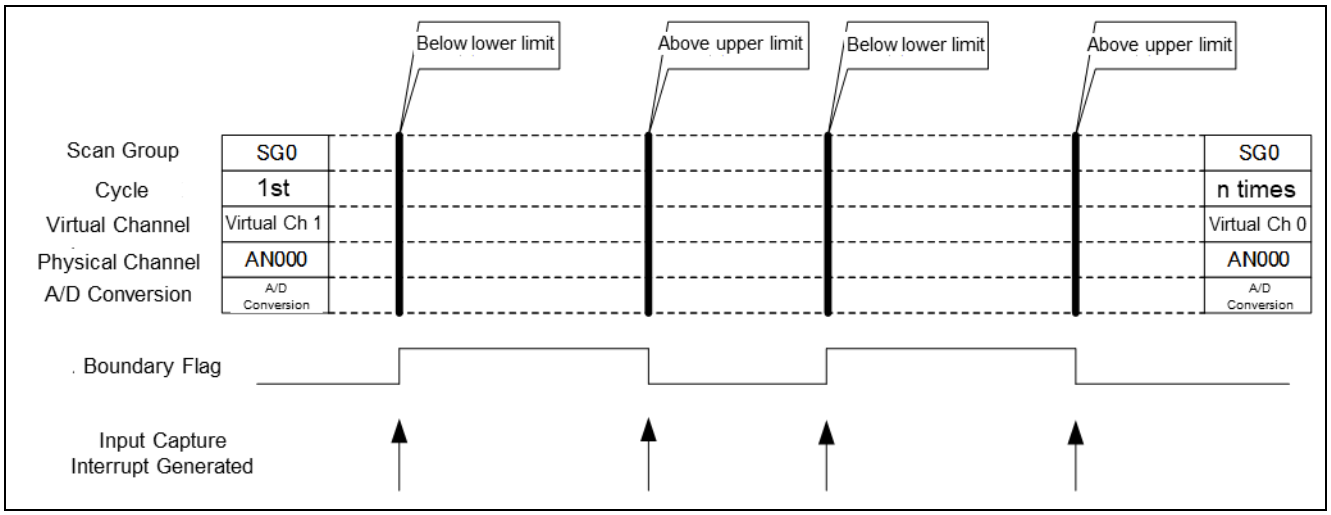

Figure 3-51 Operation Example of Boundary Flag Change and Input Capture Interrupt Generation

3.13.2 Use Function

The functions used in this operation example are shown below.

- A/D Convertor（ADCK0）
- ADC Boundary Flag Generator（ABFG）
- Virtual Port
- Timer A
- Port

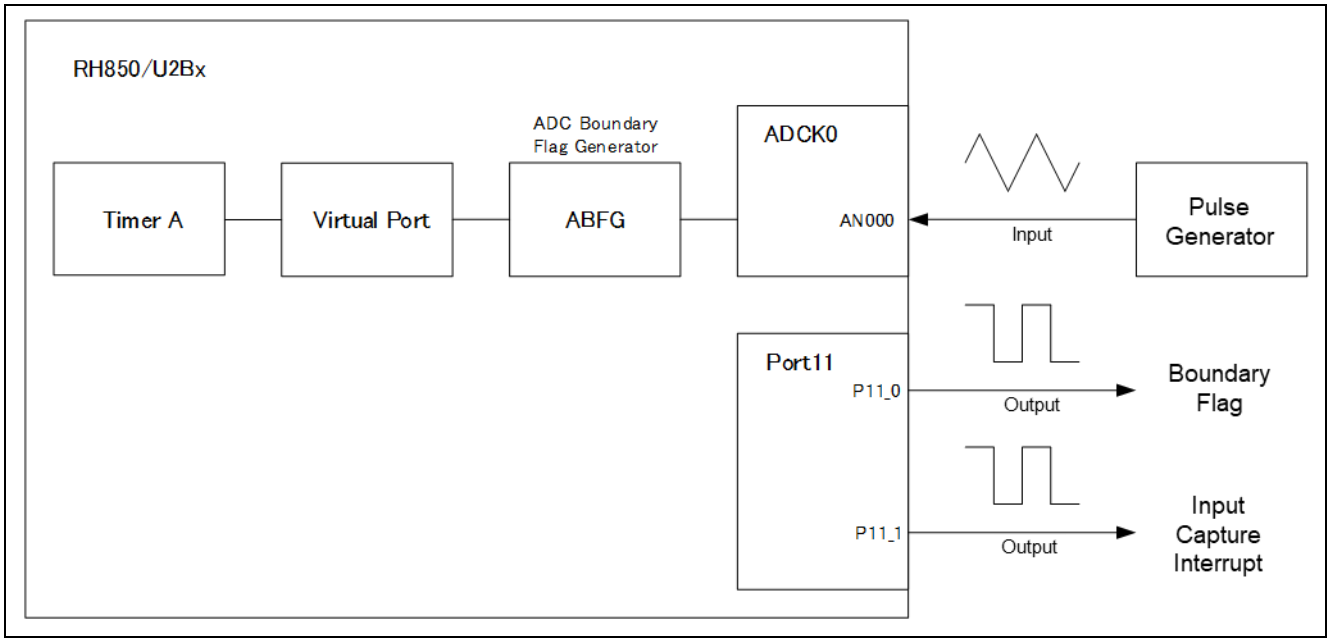

Figure 3-52 System Configuration

3.13.3 Explanation for Operation Example

In this operation example, normal A / D conversion is performed in the multi-scan mode using AN000 of the ADCK0 module.

Allocate a virtual channel (AN000) to the scan group 0 (SG0).

Input the triangle wave with the amplitude of 0 to 5[V] and the period of 100Hz to AN000.

Set 4V to the upper limit and 2V to the lower limit. * This setting is a reference value.

For the boundary flag of ABFG, Set the A/D conversion value to High when it is below the lower limit and Low when it is above the upper limit. Enable the AVSEG noise filter and set the recovery counter to “1” and the error counter to “3”. In this setting, if the A / D conversion value exceeds the upper and lower limit values three or more times, an upper and lower limit error is outputted.

(About the noise filter operation, refer to “3.11.3 Explanation for Operation Example”)

Sets Timer A of ATU as the input capture at both the rising and falling edge of TIA00, and sets the input capture interrupt to enable.

Set the virtual port (P37_0) to TIA00.

Start A/D conversion of AN000 with the soft trigger ADSTART and the scan end interrupt (ADI00) stores the conversion value of AN000 in the variable.

At this time, if the A / D conversion value exceeds the upper and lower limit values, the boundary flag changes. Also, the input capture interrupt of the timer A is generated.

In this operation example, uses port P11_0 and P11_1 as the output port to confirm the generation of the input capture interrupt of timer A.

P11_0 polls the boundary flag status register and outputs the value of the boundary flag status register.

P11_1 performs the toggle output at the input capture interrupt of timer A.

By changing the outputs of P11_0 and P11_1 at the same time, it can be confirmed that the input capture interrupt of timer A is generated by the boundary flag.

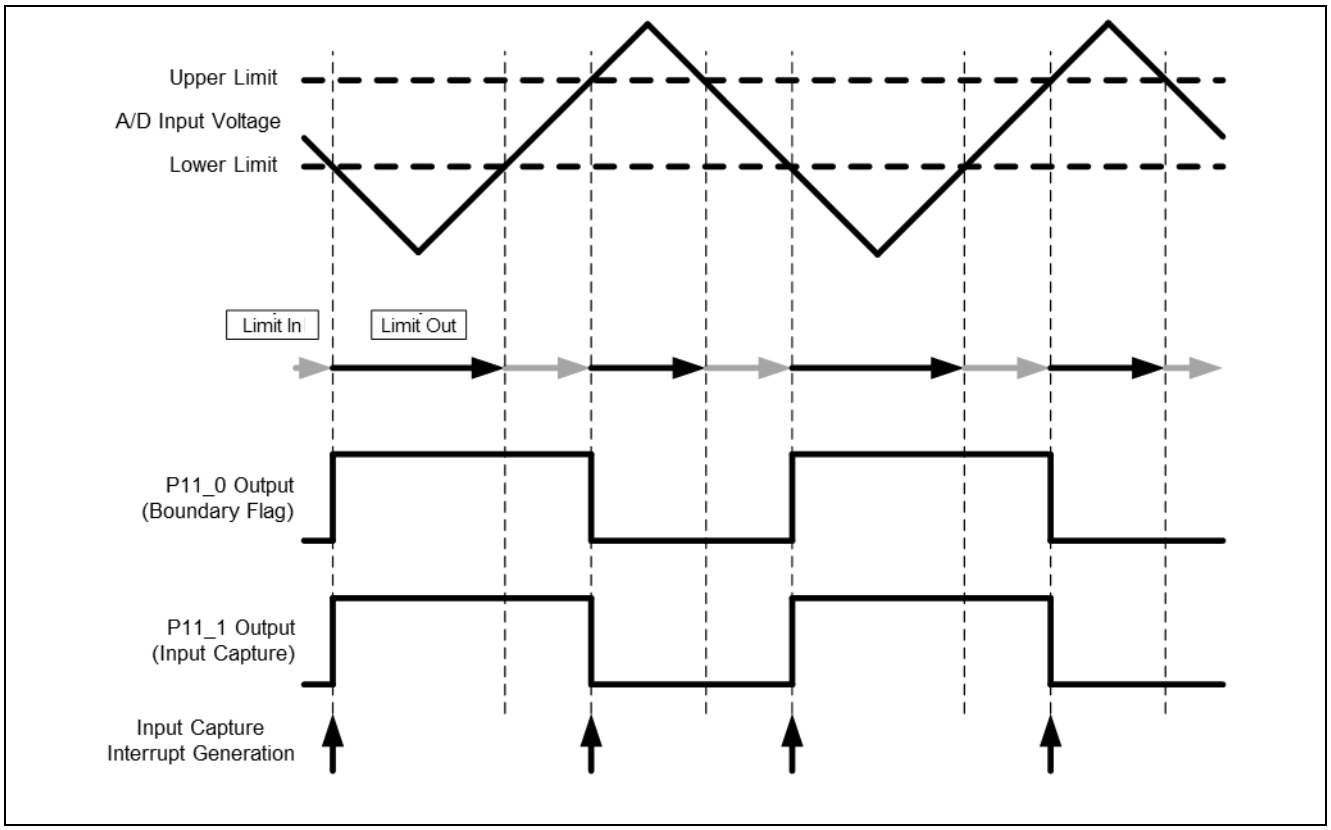

Figure 3-53 Boundary Flag Change and Port Output

3.13.4 Software Explanation

- Register Setting

Module Explanation

Module list in this operation example is shown below.

Table 3-44 Module List

Module Name	Label Name	Function
Main routine	main_pe0	Perform the various settings and application start.
Port initialize routine	port_init	Initialize the port.
Interrupt initialize routine	intc_init	Initialize the ADCK and the timer A interrupt.
ADCK initialize routine	ADCK_init	Initialize the ADCK.
ABFG initialize routine	abfg_init	Initialize the ABFG.
Timer A initialize routine	atu_init	Initialize the timer A.
Timer A interrupt processing routine	eiint86	Clear the input capture of TIA00 and invert the port P11_0 output.
ADCK interrupt processing routine	eiint441	Store the A/D conversion result to the variable in virtual scan group end interrupt processing.

The register settings for each function in this operation example are shown below.

Table 3-45 Interrupt Register Setting

Register Name	Setting Value	Function
EIBD86	0x00000000	Bind the interrupt to PE0 (CPU0).
EIC86	0x0040	Refer to table/ Priority level 0
EIBD441	0x00000000	Bind the interrupt to PE0 (CPU0).
EIC441	0x0040	Refer to table/ Priority level 0

Table 3-46 ABFG Register Setting

Register Name	Setting Value	Function
ABFG0 BFGCR	0x01030000	Boundary flag toggle setting (BTGC)： Low when the upper limit is exceeded, high when the lower limit is exceeded.
		Boundary flag pulse generation (BPGC)： Boundary flag rising/falling
		Input channel：virtual channel 0
ABFG0CNTCR	0x10000101	Enable the filter.
		Recovery counter setting (NRMCNT)：1
		Error counter setting (ERRCNT)：1

Table 3-47 ADCK Register Setting

Register Name	Setting Value	Function
ADCK0VCR00	0x10000000	Enable the virtual channel upper/lower limit exceeded notification.
		Select the virtual channel upper/lower limit table register 0.
		Conversion type: normal A/D conversion
		Not select wait time table.
		Not entry DFE.
		Not output virtual channel complete interrupt.
		Physical Ch0/Sub CH0 (AN000)
ADCK0SGCR0	0x50	Enable ADSTART.
		Multiple-cycle Scan Mode
		Output “ADI00” in the end of “SG0”.
		Prohibit the trigger input to “SG0”.
ADCK0SGVCPR0	0x0000	Start virtual channel0, end virtual channel 0.
ADCK0SGMCYCR0	0x00	Signed 12bit integer format
ADCK0ADCR2	0x00	Synchronous suspend
ADCK0ADCR1	0x00	Output “ADI00” in the end of “SG0”.
ADCK0VCULLMTBR0	0x66663332	Virtual channel upper/lower limit table register 0 Upper limit：4V (0x6666), Lower limit：2V (0x3332)

Table 3-48 ATU Register Setting

Register Name	Setting Value	Function
PSCR0	0x0001	Set the prescaler 0 division ration to 2.
TIOR1A	0x0003	Set to input capture on both rising and falling edges of TIA00.
TSCRA	0x80FF	Clear the flag of the timer A.
TIERA	0x0001	Set the TIA00 input capture interrupt to enable.
ATUENR	0x03	Enable the count operation of the timer A and the prescaler.

- Operation Flow

  The flow chart in this operation example is shown below.

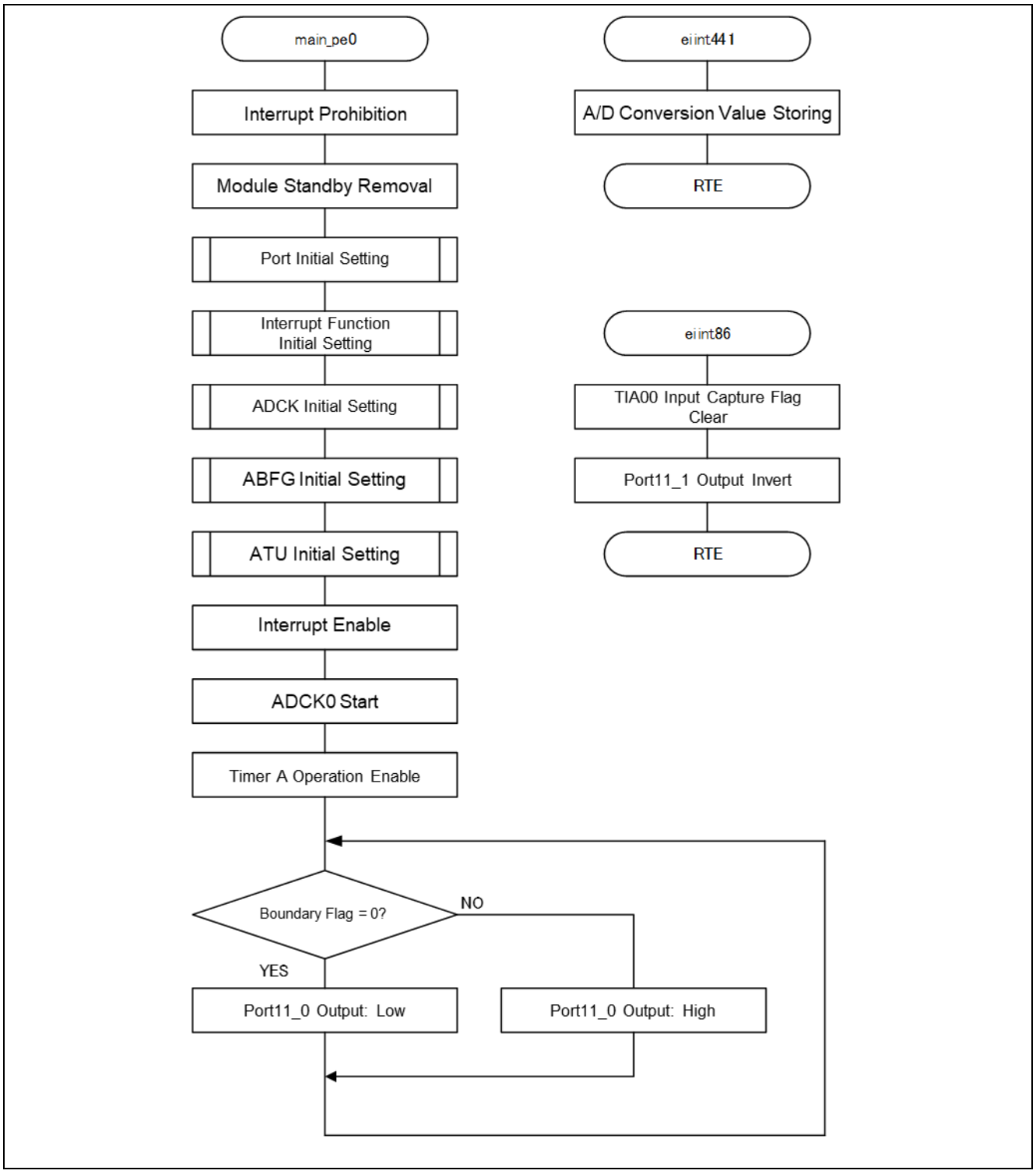

Figure 3-54 Flowchart

Our Company's Website and Inquiry

Website
http://japan.renesas.com/

Inquiry
http://japan.renesas.com/contact/

Revision History

Rev.	Data	Description	
		Page	Summary
1.00	2023.10.5	-	Initial edition

Precautions for use of the product

This section describes the "Precautions" that apply to all microcontroller products. Please refer to this document and the Technical Update for precautions on individual products.

1. Treatment of unused pins

   [Caution] Please dispose of unused pins according to "Handling of unused pins" in the text.

   The impedance of the input pins of CMOS products is generally high impedance. If the unused pins are operated in an open state, noise around the LSI may be applied due to the induction phenomenon, a through current may flow inside the LSI, or it may be recognized as an input signal and malfunction may occur. Dispose of unused pins according to the instructions given in "Disposal of unused pins" in the text.

2. Treatment at power-on

   [Caution] The state of the product is undefined when the power is turned on.

   When the power is turned on, the state of the internal circuits of the LSI is indeterminate and the state of register settings and pins is undefined.

   For products that are reset using the external reset pin, the pin state cannot be guaranteed from the time the power is supplied until the reset becomes valid.

   Similarly, in the case of products that are reset using the built-in power-on reset function, the pin states cannot be guaranteed from the time the power is turned on until the voltage reaches a certain level.

3. Prohibition of Access to Reserved Addresses

   [Caution] Access to reserved addresses is prohibited.

   The address area has a reserved address allocated for future function expansion. The operation when these addresses are accessed cannot be guaranteed, so do not access them.

4. About clock

   [Caution] When resetting, release the reset after the clock has stabilized.

   When switching the clock during program execution, switch the clock after the switching destination clock is stable.

   In a system that starts operating with a clock that uses an external oscillator (or external oscillator circuit) at reset, release the reset after the clock is sufficiently stable. Also, when switching to a clock that uses an external oscillator (or external oscillator circuit) in the middle of a program, make sure that the clock to be switched to is sufficiently stable before switching.

5. Differences between products

   [Caution] When changing to a product with a different model name, perform a system evaluation test for each product model name.

   Even if the MCUs in the same group have different model numbers, the characteristic values, operating margins, noise immunity, noise radiation, etc. may differ within the range of electrical characteristics due to differences in internal ROM and layout patterns. When changing to a product with a different model name, perform a system evaluation test for each individual 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 not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

RENESAS

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

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

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

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

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

Colophon 7.0