RL78/F23, F24

DC/DC control using Timer RDe and AAU

R01AN6838EJ0100
Rev.1.00
2023.07.31

Introduction

This application note describes the example of DC/DC power control operation for the RL78/F23, F24 using Timer RDe, and AAU.

Depending on the usage conditions, the operation shown in the example may differ from the actual operation of the microcomputer. Therefore, please perform a thorough evaluation after porting. For details, be sure to refer to User's Manual: Hardware.

Target Devices

- RL78/F23, F24

Contents

1. Overall DC/DC Control

This chapter describes an example of DC/DC control of a digital power supply system using the RL78/F2x.

1.1 Feature of RL78/F2x for the Digital Power Supply Control

The RL78/F2x microcontroller is equipped with the following functions to efficiently control the digital power supply system.

(1) As an extended PWM mode function of timer RDe, it supports simultaneous rewriting of compare registers and can output up to 4 PWM waveforms of the same cycle (2 signal each counter).

(2) It supports the dithering function as an extension function of the PWM waveform output of the timer RDe.

(3) The gate function of the timer RDe supports control.

(4) PWM output can be interrupted and resumed without software by inputting a signal from the outside of the microcontroller to the timer restart pin (TRDiRES).

(5) Improved responsiveness by improving the performance of the A/D converter.

(6) By installing arithmetic assist hardware (AAU), the calculation performance in PI control such as digital power supply is improved.

1.2 RL78/F2x Peripherals for the Digital Power Supply Control

1.2.1 Timer RDe

(1) Dithering Function

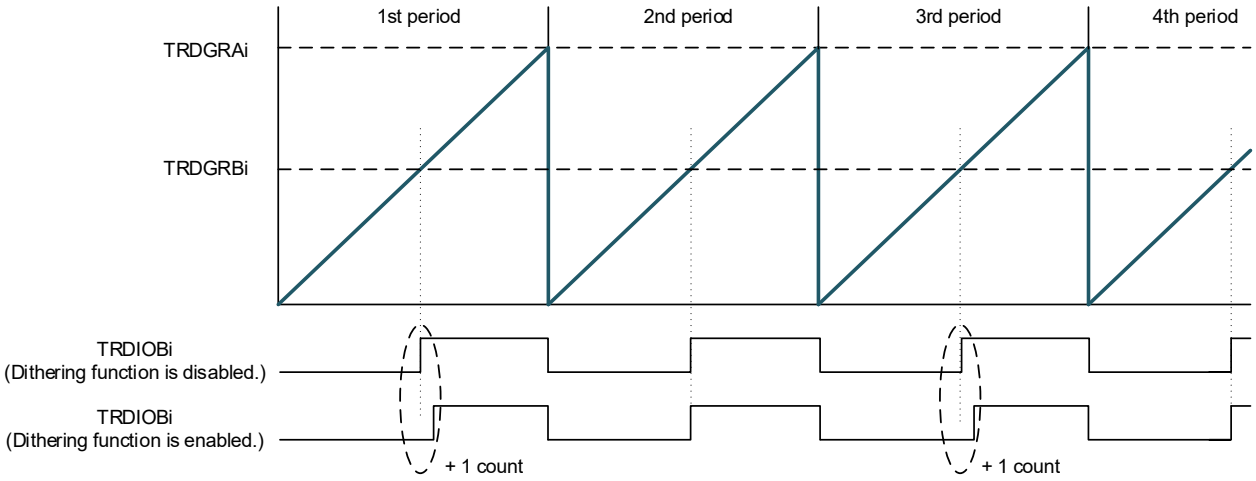
A dithering function is available for the outputs of TRDIOBi and TRDIODi (i = 0, 1) when using the extended PWM mode. The average resolution of the PWM can be improved by extending the active pulse width of the selected period by one count relative to the 16 periods of the PWM waveform.

The period for extending the active pulse width by 1 count is selected by setting bits DNR3 to DNR0 in the TRDDNRi register. The figure below shows the PWM output operation with the dithering function.

DTEN	DNR[3:0]	PWM Output Period															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0000B																
	0001B	■															
	0010B	■								■							
	0011B	■				■				■							
	0100B	■				■				■				■			
	0101B	■		■		■				■				■			
	0110B	■		■		■				■		■		■			
	0111B	■		■		■		■		■		■		■			
	1000B	■		■		■		■		■		■		■		■	
	1001B	■	■	■		■		■		■		■		■		■	
	1010B	■	■	■		■		■		■	■	■		■		■	
	1011B	■	■	■		■	■	■		■	■	■		■		■	
	1100B	■	■	■		■	■	■		■	■	■		■	■	■	
	1101B	■	■	■	■	■	■	■		■	■	■		■	■	■	
	1110B	■	■	■	■	■	■	■		■	■	■	■	■	■	■	
	1111B	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
0	-																

■ The PWM period during which one clock is added to the active pulse.
□ The PWM output set in register.

Figure 1-1 The PWM Output Operation of Dithering Function



Conditions: TRDDNRi.DNR[3:0] = 1000B (1 clock is added in the 1st , 3rd , 5th, 7th, 9th, 11th, 13th, and 15th periods.)
TRDPOCRi.POLB = 1 (Active level: High)

Figure 1-2 The PWM Output Waveform Example of Dithering Function

(2) Gate Function

When using extended PWM mode, the gate function can be used for the output of TRDIOBi and TRDIODi (i = 0, 1). Based on 16 PWM cycles, it is masked (gate) controlled for the selected n cycles. The output mask period is selected by setting the bits GPR3 to GPR0 of the TRDGPRi register. The figure below shows the PWM output operation of the gate function.

TRDGPRi.GPAT = 0 (Pattern.1 select):

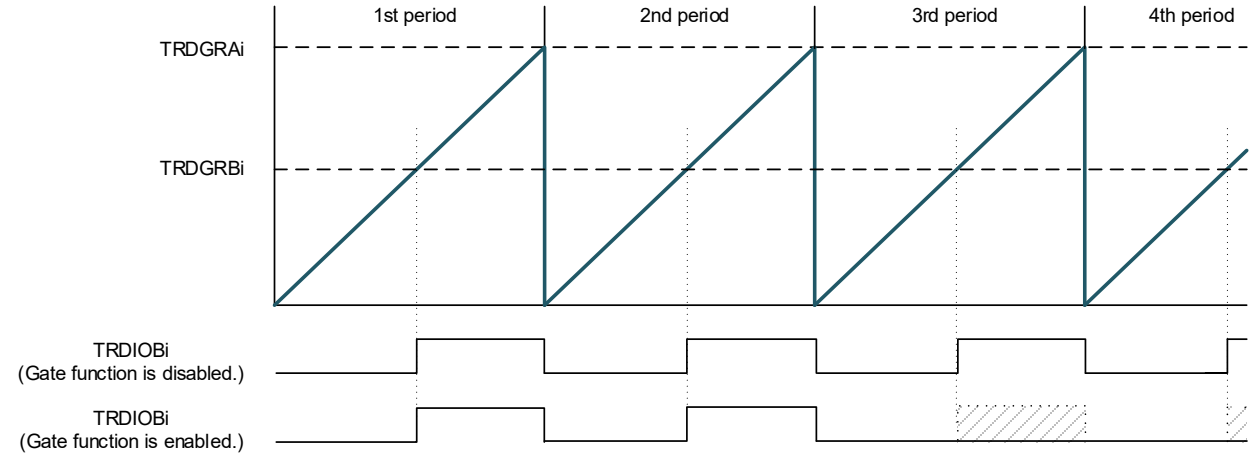
GTEN	GPR[3:0]	PWM Output Period															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0000B																
	0001B	■															
	0010B	■	■														
	0011B	■	■	■													
	0100B	■	■	■	■												
	0101B	■	■	■	■	■											
	0110B	■	■	■	■	■	■										
	0111B	■	■	■	■	■	■	■									
	1000B	■	■	■	■	■	■	■	■								
	1001B	■	■	■	■	■	■	■	■	■							
	1010B	■	■	■	■	■	■	■	■	■	■						
	1011B	■	■	■	■	■	■	■	■	■	■	■					
	1100B	■	■	■	■	■	■	■	■	■	■	■	■				
	1101B	■	■	■	■	■	■	■	■	■	■	■	■	■			
	1110B	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
	1111B	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
0	-	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

TRDGPRi.GPAT = 1 (Pattern.2 select):

GTEN	GPR[3:0]	PWM Output Period															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0000B		■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
	0001B			■	■	■	■	■	■	■	■	■	■	■	■	■	■
	0010B				■	■	■	■	■	■	■	■	■	■	■	■	■
	0011B					■	■	■	■	■	■	■	■	■	■	■	■
	0100B						■	■	■	■	■	■	■	■	■	■	■
	0101B							■	■	■	■	■	■	■	■	■	■
	0110B								■	■	■	■	■	■	■	■	■
	0111B									■	■	■	■	■	■	■	■
	1000B										■	■	■	■	■	■	■
	1001B											■	■	■	■	■	■
	1010B												■	■	■	■	■
	1011B													■	■	■	■
	1100B														■	■	■
	1101B															■	■
	1110B																■
	1111B																
0	-	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

■ The PWM output of this cycle is not masked.
The PWM output of this cycle is masked. (Gate function enables)

Figure 1-3 The PWM Output Operation of Gate Function

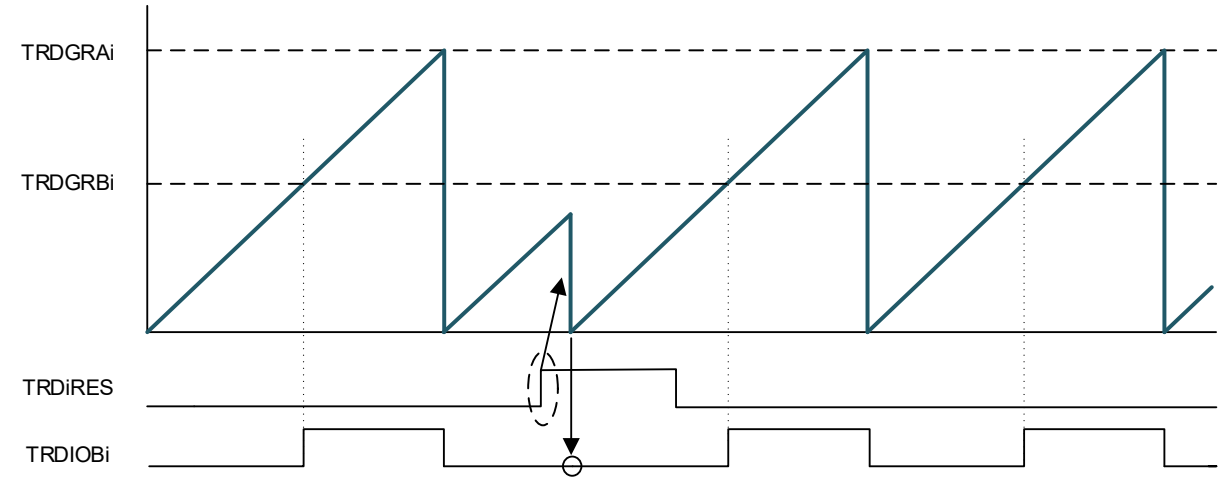


Conditions: TRDGPRi.GPAT = 0 (When Patter.1 is selected.)
TRDGPRi.GPR[3:0] = 0010B (Only 1st and 2nd periods are PWM output. PWM output is masked after 3rd periods.
TRDPOCRi.POLB = 1 (Active level: High)

Figure 1-4 The PWM Output Waveform Example of Gate Function

(3) Counter Restart Function

In extended PWM mode, the timer RDi counter (TRDi) can be initialized by TRDiRES (i = 0, 1) pin input and restarted counting by using the counter restart function. The figure below shows an example of PWM output operation of the counter restart function.



Conditions: TRDEMRi.CCLV[1:0] = 10B (Counter restart at the rising edge of TRDiRES input.)
TRDOCRi.TOBi = 0 (Initial output level: Low)
TRDPOCRi.POLB = 1 (Active level: High)

Figure 1-5 The PWM Output Operation Example of Counter Restart Function

1.2.2 Application Accelerator Unit (AAU)

AAU math functions can be used to implement PI control algorithms for DC/DC control. This function can be used by selecting 1- to 3-channel of processing.
The PI calculation algorithm for DC/DC control is shown below.

D (n) = D (n-1) + A1 × E (n) + A2 × E (n-1)

D (n): Calculated Duty value
D (n-1): Previous Duty value
E (n): Current error value = Reference current value − Measurement current value
E (n-1): Last current error value = Reference current value − Last measurement current value
A1, A2: Coefficient

n: Sampling count number

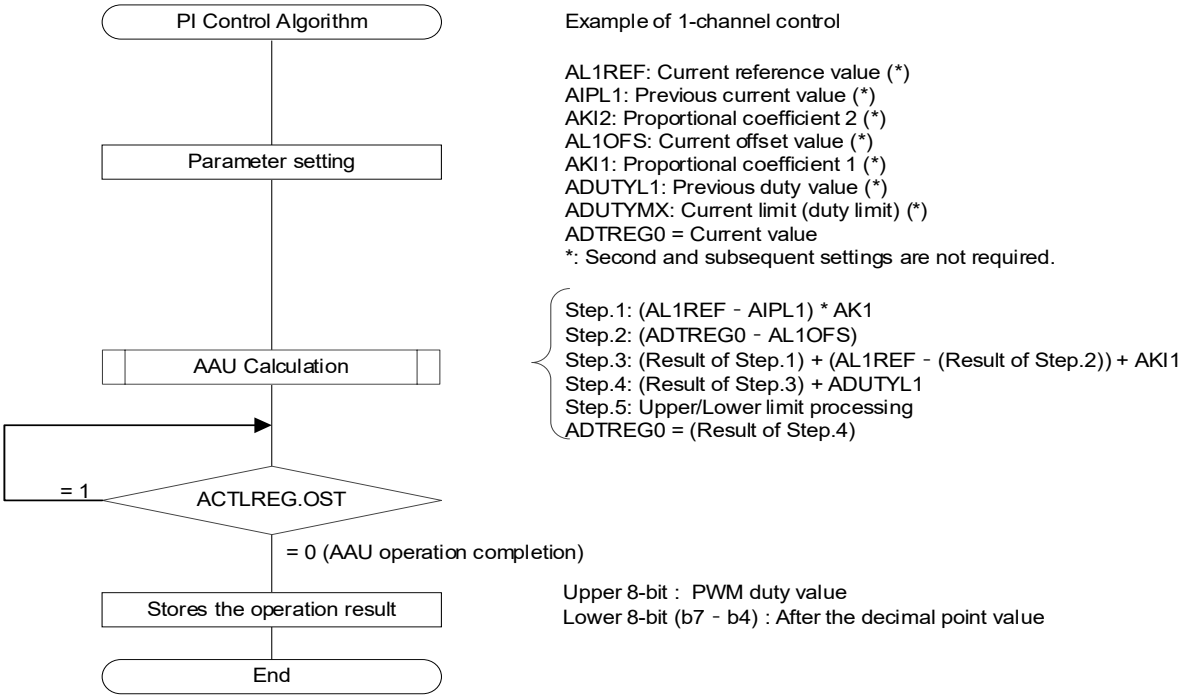


Figure 1-6 Calculation Algorithm of PI Control for DC/DC

1.3 Hardware Structure for the Digital Power Supply Control

Figure 1-7 shows the hardware configuration for digital power control using DC/DC control by RL78/F2x. The hardware specifications are also shown in Table 1-1.

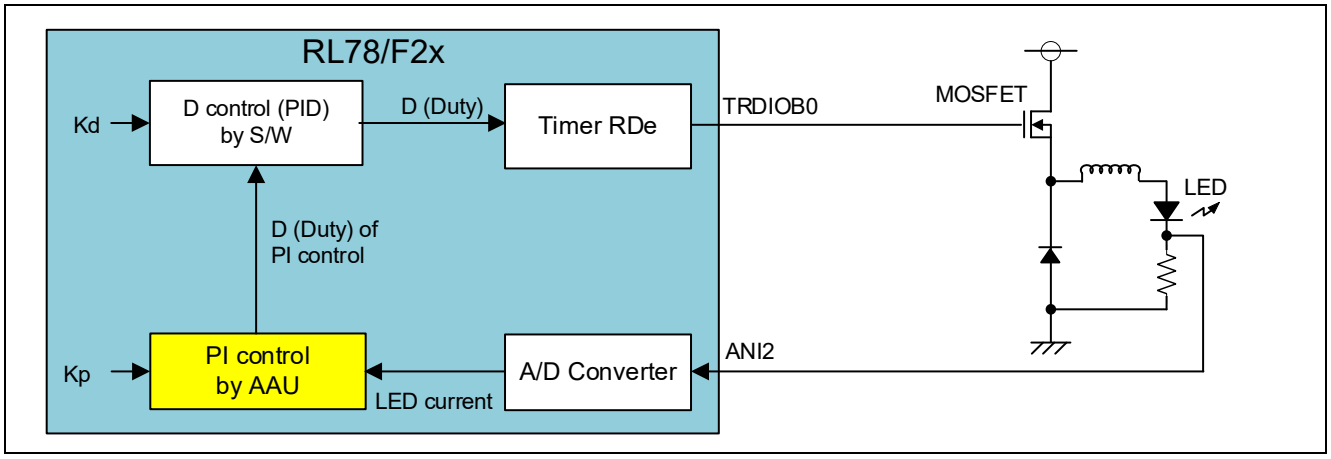


Figure 1-7 System Configuration

Table 1-1 Used Hardware Configuration

Function	Description
CPU	f_{CLK} = 40 MHz
Timer RDe	• The f_{TRD} (count clock) use the f_{MP} = 80 MHz • Extended PWM Mode use • PWM output (Use dithering function)
A/D Converter	• LED Current measurement use • Use INTTRD0_IFA of timer RDe for A/D conversion start event
DTC	• High-speed DTC use • Use A/D end interrupt for DTC transfer start request event
Application Accelerator Unit (AAU)	• Use PI calculation for DC/DC control

1.4 List of Used Pin Function

Table 1-2 shows the pins used and the functions in the digital power control system.

Table 1-2 List of Used Pin

Pin Function	Input / Output	Contents
TRDIOB0/P11	Output	Timer RDe PWM output
ANI2/P82	Input	Feedback current input from the digital power control system

2. Control Software

The DC/DC control software in this application note controls the gate of the step-up DC/DC converter by the extended PWM of the timer RDe of the RL78/F2x microcontroller. In addition, the feedback current value from the digital power supply circuit is monitored by the built-in A/D converter to adjust the gate opening/ closing timing.

2.1 Initialize the RL78/F2x Internal Functions

This control software uses the following peripheral functions built into the RL78/F2x microcontroller.

2.1.1 Internal Clock

Control software uses the internal clock with the following settings.

- Use PLL clock (80MHz) as the timer RDe count clock
- Use PLL clock (40MHz) as the main system clock

2.1.2 Timer RDe

Control software uses the timer RDe with the following settings.

Table 2-1 Timer RDe Settings

Items	Contents	Usage
Count clock (f_{TRD})	f_{MP} (PLL clock) = 80 [MHz]	PWM output Dithering function
Operation mode	Extended PWM mode	
Output pin	TRDIOB0	
PWM frequency	5 [µs]	
Output level	Initial output: low level, Active: High level	
A/D trigger output	Output INTTRD0_IFA as A/D start trigger	
Other functions	• Gate function does not use • Counter-restart function does not use	

2.1.3 A/D Converter

Control software uses the A/D converter with the following settings.

Table 2-2 A/D Converter Settings

Items	Contents	Usage
Used channel	ANI2 (analog input that supports high-speed conversion) use	Measure the current value
A/D Conversion clock (ADCLK)	f_{CLK}/1 = 40 [MHz]	
Operation mode	Single scan mode	
Synchronous trigger	Use timer RDe interrupt (INTTRD0_IFA) as a synchronous trigger to start A/D conversion	
Reference voltage	Uses AV_{REFP} and AV_{REFM}	
Interrupt output	Generates INTAD (end of A/D conversion) each time scanning of the ANI2 pin ends	

2.1.4 DTC

Control software uses the DTC with the following settings.

Table 2-3 DTC Settings

Items	Contents	Usage
Used DTC channel	High-speed DTC channel 0	Transfer A/D conversion result to RAM address
Operation mode	Repeat mode	
Data size	16-bit data size, ANI2 conversion result transferred 8 times	
DTC vector base address	FFD00H	
Synchronous trigger	A/D conversion end interrupt	

2.1.5 Application Accelerator Unit (AAU)

Control software uses the AAU with the following settings.

Table 2-4 AAU Settings

Items	Contents	Usage
Calculation algorithm	PI control for DC/DC converter control operation (total channel 1)	PI control for DC/DC converter control operation
Start trigger	Use software trigger (Start operation by writing to OST bit)	

2.2 Control Software DC/DC control

DC/DC control of the control software uses the extended PWM mode function of timer RDe, and outputs a PWM waveform from the TRDIOB0 pin of RL78/F2x as shown in Figure 2-1. High output and low output of the PWM waveform can be controlled by compare match between the count value set in the timer RDe registers TRDGRA0 and TRDGRB0 and the TRD0 counter.

In addition, the PWM signal waveform is input to the MOSFET of the digital power supply and used for gate ON/OFF control of the booster circuit (see Figure 1-7. In the DC/DC control of the booster circuit, setting a long MOSFET gate ON time can lengthen the charging period of the digital power supply. The voltage of the digital power supply is consumed by the LEDs connected to the circuit. In order to stably light the LED with a stable voltage supply, the control software obtains the current value from the digital power supply by A/D conversion and feeds it back to the PWM signal by PID control.

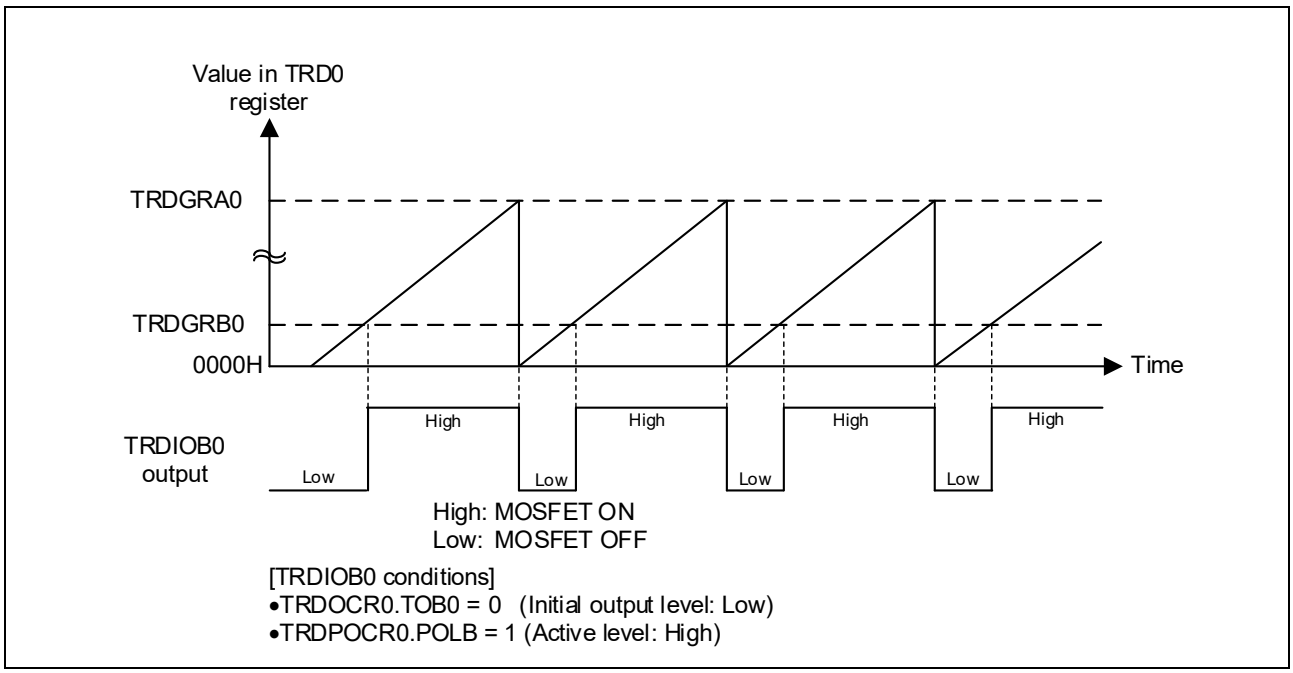


Figure 2-1 DC/DC Gate Control Example

2.2.1 A/D Value Acquisition Control

The control software periodically performs A/D conversion of the LED current value in synchronization with timer RD0. The converted A/D value is transferred to the RAM area by the DTC. The control software treats 8 RAM transfers as one set depending on the DTC setting. After 8-time transfers are completed, an A/D conversion end interrupt is notified to the CPU and the A/D interrupt handler is executed.

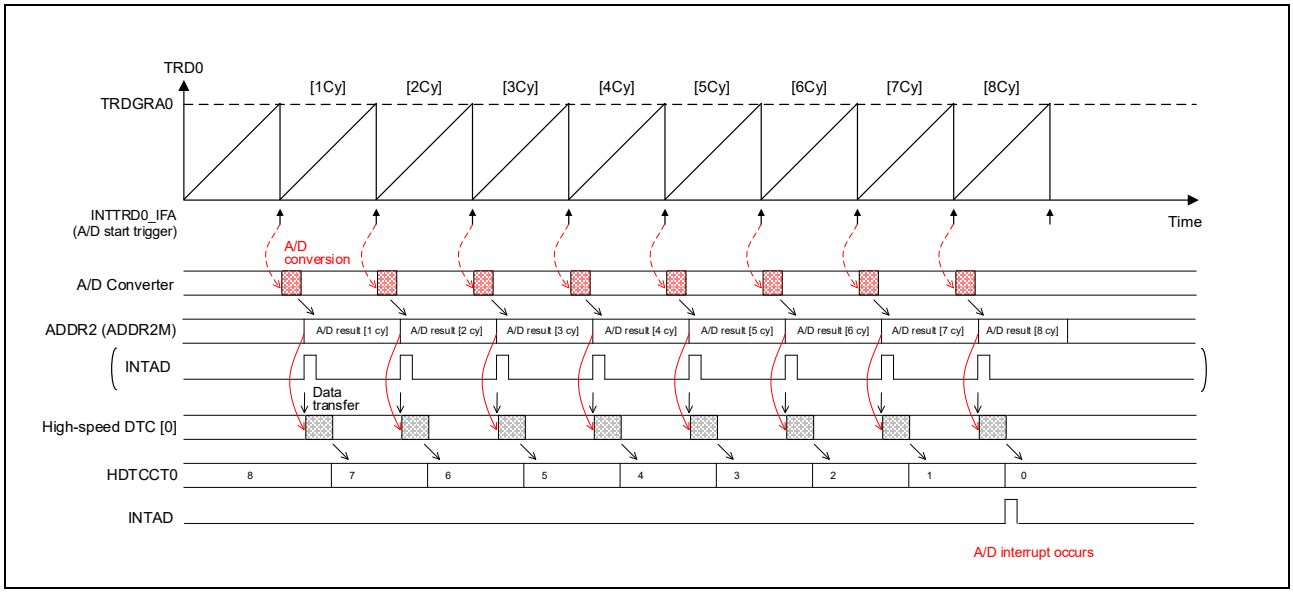


Figure 2-2 A/D Acquisition Image of Control Software

The control software averages the A/D values 8 times and uses the result as the LED current value (measured value).

2.2.2 PI Calculation

In the control software, AAU of RL78/F2x is used for calculation of P control (proportional) and I control (integral) of PID control.

The PI calculation of the control software is performed in the following procedure.

(1) Set each parameter used in PI operation to AAU register

- Set each parameter in AAU
  ADUTYMX (current limit duty), AKI1 (proportional coefficient Kp1),
  AKI2 (proportional coefficient Kp2), AL1REF (current reference), AL1OFS (current offset),
  ADUTYL1 (previous duty), AIPL1 (previous current)

(2) Execute calculation by AAU

- Set the acquired current value to the AAU register
  ADTREG0 = current value
- Set previous PWM duty value after PID calculation to AAU register
  ADUTYL1 = previous PWM duty (including dithering count value)
- Start calculation
  ACTLREG.OST ← 1

(3) Confirm completion of calculation by AAU

- Confirm ACTLREG.OST bit is 0

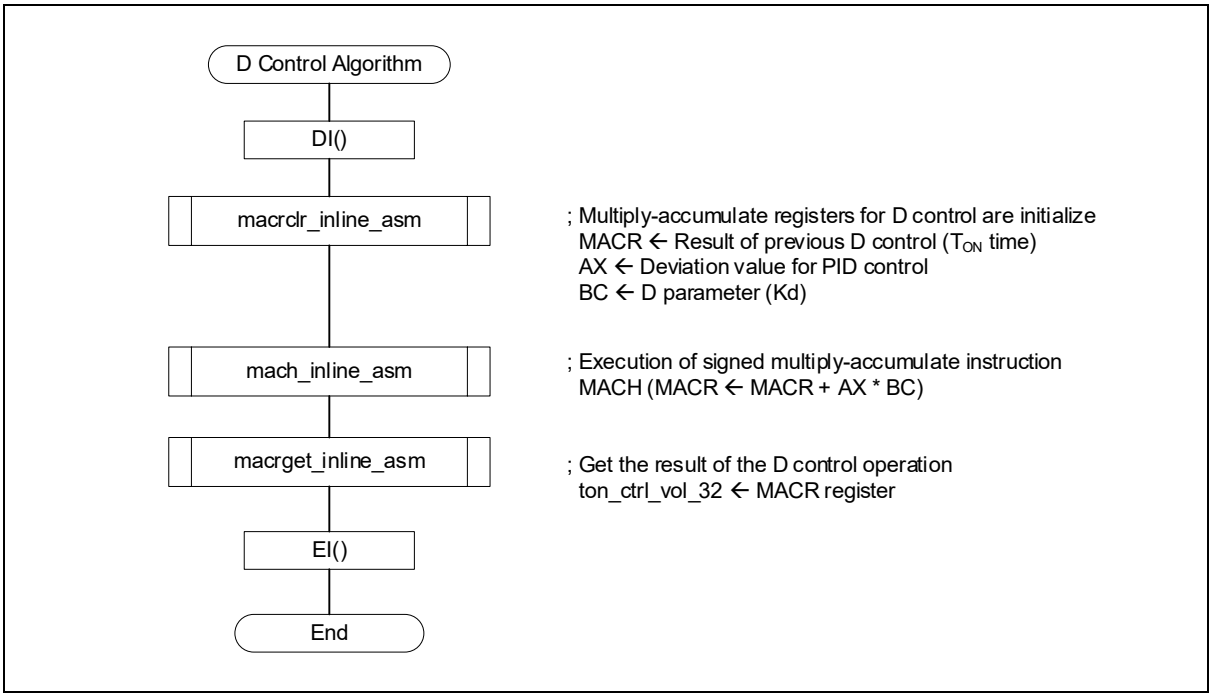
(4) Store the result of PI calculation

- Store AAU calculation result (ADTREG0)

2.2.3 D Calculation

In the control software, D control (differentiation) of PID control uses RL78/F2x CPU instructions to perform software calculations. As an execution condition for D control, control is executed when there is a deviation between the A/D value (measured value) and the reference current value (target value).

(1) Perform D control calculations by microcomputer instructions



(2) Extract the next duty value and dithering count setting from the calculation result of D control

- Get next PWM duty value
  led_ch1_pid_param.dutys[1] = (uint16_t)(ton_ctrl_vol_32 >> 16);
- Get next dithering count
  led_ch1_pid_param.remainder[1] = (16U) - (uint8_t)(ton_ctrl_vol_32 & 0x0000FFFF) >> 12);

2.2.4 Feedback to PWM

The duty value and dithering count are set in the buffer register of timer RDe based on the calculation result obtained by the PID control of the control software. The control software uses the simultaneous rewrite function of the timer register when updating the set value of timer RDe.

(1) Reflecting the calculation result of PID control in the timer value

- Write PWM duty value to TRDGRB0 buffer register TRDGRD0
  trd0_update->trdgrdn = led_ch1_pid_param.dutys[1]; // Timer RDe duty value
- Write dithering count to TRDDNR0 register
  trd0_update->trddnrn = led_ch1_pid_param.rimainder[1]; // Dithering number

(2) Simultaneous rewrite request for timer registers

- Write 1 to the RDT bit of the TRDRDT0 register
  R_Config_TRD0_Set_TRD0_ReloadTrigger(trd0_update);

2.3 DC/DC Control Flow

Figure 2-3 shows the overall flow of DC/DC control.

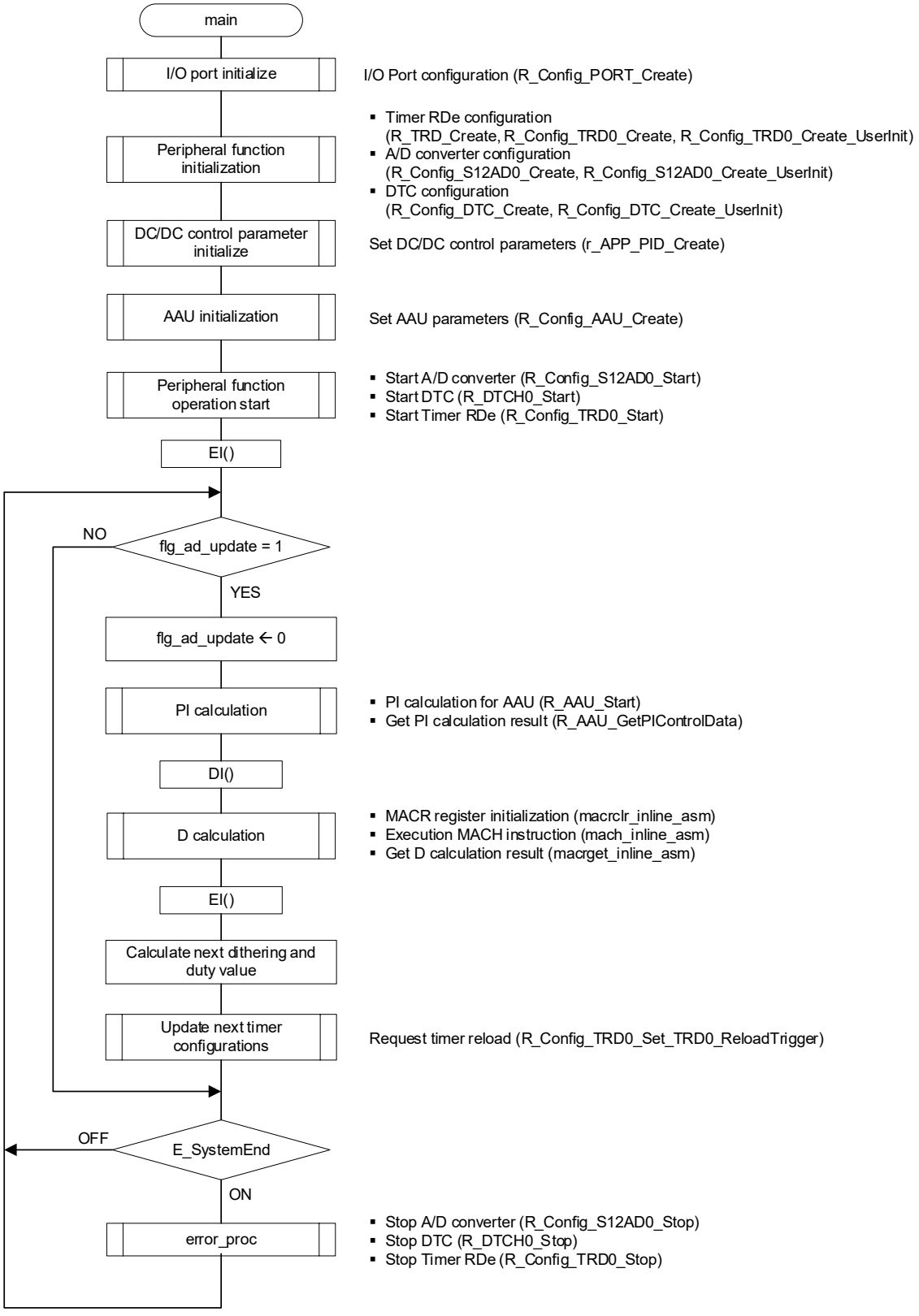


Figure 2-3 DC/DC Control Flowchart (1/2)

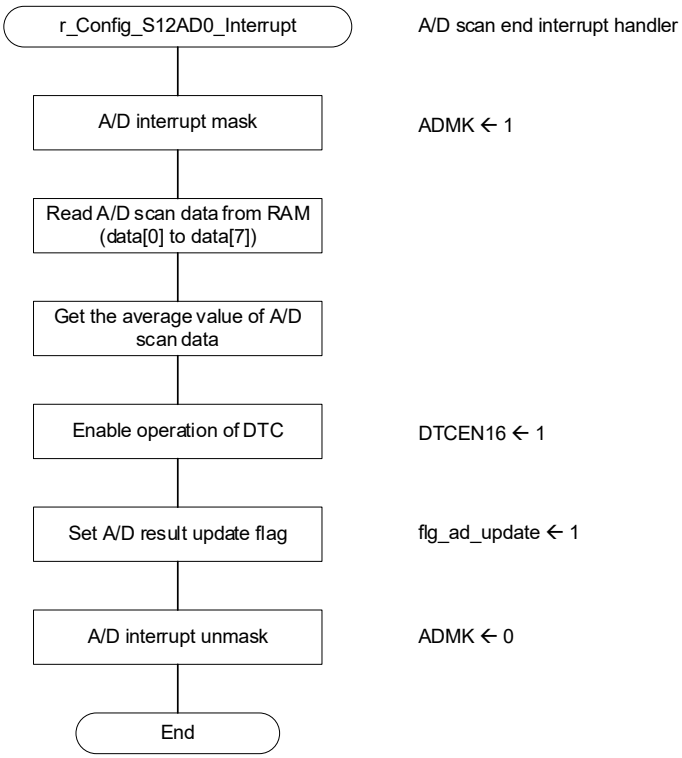


Figure 2-3 DC/DC Control Flowchart (2/2)

2.4 Variable List

Table 2-5 shows the variable type definitions used by the control software.

Table 2-5 PID_PARAM Type Definition

Member	Type	Usage	Content
paramKp1	int16_t	PI control	Constants for PI operations
paramKp2	int16_t	PI control	Constants for PI operations
duty_limit	uint8_t	PI control	Maximum current duty
current_offset	uint16_t	PI control	Current value offset
currents[2]	uint16_t	PI control	Current measurement value ([0]: previous value, [1]: latest value)
dutys[2]	uint16_t	PI control	Current duty ([0]: previous value, [1]: latest value)
remainder[2]	uint16_t	PI control	Dithering setting value ([0]: previous value, [1]: latest value)
paramKd	uint16_t	D control	Constants for D control calculation
devias[2]	int16_t	D control	Deviation (reference current value - measured current value); ([0]: previous value, [1]: latest value)
differs[2]	int16_t	D control	Deviation difference ([0]: previous value, [1]: latest value)
current_ad	uint16_t	D control	Current A/D value
current_target	int16_t	PI and D common	Target current value
dither_count	uint8_t	PI and D common	Dithering times

A list of variables used by the control software is shown below.

Table 2-6 Variable List of Control Software

Variable name	Type	Content
led_ch1_pid_param	PID_PARAM	PID control internal management variable
E_SystemEnd	uint8_t	System termination flag
dtc_vectortable[]	uint8_t	DTC vector management table
flg_ad_update	uint8_t	A/D data updated flag

2.5 List of Function

The function list of this software is shown below.

Table 2-7 Function List of Control Software

Function Name	Overview
main	Main processing for the digital power supply system
error_proc	System error processing
r_APP_PID_Create	Initialize for PID control
r_APP_PID_Start	Start PID control
r_APP_PID_Stop	Stop PID control
r_APP_PID_UpdateParam	Update settings for PID control
R_Config_AAU_Create	Initialize AAU
R_AAU_Start	Start AAU calculation
R_AAU_GetPIControlData	Get result from AAU (PI calculation for DC/DC control).
R_Config_PORT_Create	Initialize PORT
R_Config_DTC_Create	Initialize DTC
R_Config_DTC_Create_UserInit	Initialize DTC (Control software specific settings)
R_DTCH0_Start	Start high-speed DTC
R_DTCH0_Stop	Stop high-speed DTC
R_Config_S12AD0_Create	Initialize A/D Converter
R_Config_S12AD0_Create_UserInit	Initialize A/D Converter (Control software specific settings)
r_Config_S12AD0_Interrupt	A/D Converter interrupt handler
R_Config_S12AD0_Start	Start A/D converter control
R_Config_S12AD0_Stop	Stop A/D converter control
R_TRD_Create	Initialize timer RDe (Initialization before timer start)
R_Config_TRD0_Create	Initialize Timer RDe
R_Config_TRD0_Create_UserInit	Initialize Timer RDe (Control software specific settings)
R_Config_TRD0_Start	Start Timer RDe
R_Config_TRD0_Stop	Stop Timer RDe
R_Config_TRD0_Set_TRD0_ReloadTrigger	Reload processing for Timer RDe settings

2.6 Function Specifications

The function specifications of software are shown below.

2.6.1 main

declaration	void main(void)
overview	Main processing for the digital power supply system.
explanation	This is the main process of this control software. Initializes each peripheral function at the start of main process. • R_Config_PORT_Create : Initialize the I/O port settings. • R_TRD_Create : Initialize the Timer RDe. • R_Config_S12AD0_Create : Initialize the 12-bit A/D converter. • R_Config_DTC_Create : Initialize the high-speed DTC. • r_APP_PID_Create : Set parameters for digital power control. • R_DTCH0_Start : Start high-speed DTC. • R_Config_S12AD0_Start : Start A/D converter. • R_Config_TRD0_Start : Start timer count of Timer RDe. • EI() : Enable interrupt. After that, when the current value of the DC/DC section is fixed (flg_ad_update = 1), the digital power control is processed according to the following procedure. • R_AAU_Start : Set the current value to AAU and start calculation. • R_AAU_GetPIControlData : Gets the calculation result of AAU. • D control calculation PWM : Perform D control using multiply-accumulate instructions, and set the next duty and dithering count in the timer RDe buffer register. • R_Config_TRD0_Set_TRD0_ReloadTrigger : Requests a transfer from the current RDe's buffer register to the compare register. If an error is detected (E_SystemEnd = 1), call the error_proc function.
parameter	no
return value	no

2.6.2 error_proc

declaration	void error_proc(void)
overview	System error processing.
explanation	Stops timer, A/D conversion, and DTC transfer, when a system error occurs. • R_Config_TRD0_Stop : Stop Timer RD0 count. • R_Config_S12AD0_Stop : Stop A/D conversion. • R_DTCH0_Stop : Stop High-speed DTC transfer.
parameter	no
return value	no

2.6.3 r_APP_PID_Create

declaration	void r_APP_PID_Create(void)
overview	Initialize for PID control
explanation	Sets the initial value to the internal management variable led_ch1_pid_param for PID control. [Member for PI control] • led_ch1_pid_param.paramKp1 = (int16_t) PI_CONTROL_PARAM_A1; • led_ch1_pid_param.paramKp2 = (int16_t) PI_CONTROL_PARAM_A2; • led_ch1_pid_param.duty_limit = (uint8_t)PI_CONTROL_CURRENT_DUTY_LIMIT; • led_ch1_pid_param.current_offset = (uint16_t)PI_CONTROL_CURRENT_AD_OFFSET; • led_ch1_pid_param.current_target = (int16_t) PI_CONTROL_CURRENT_AD_TARGET; • led_ch1_pid_param.currents[0] = 0; • led_ch1_pid_param.currents[1] = 0; • led_ch1_pid_param.dutys[0] = 0; • led_ch1_pid_param.dutys[1] = (uint16_t)PI_CONTROL_CURRENT_DUTY_INIT; [Member for D control] • led_ch1_pid_param.paramKd = (uint16_t)D_CONTROL_PARAM_KD; • led_ch1_pid_param.devias[0] = 0; • led_ch1_pid_param.devias[1] = 0; [Common members for PI and D control] • led_ch1_pid_param.dither_count = 8; Finally, Call the R_Config_AAU_Create() for AAU initialize.
parameter	no
return value	no

2.6.4 r_APP_PID_UpdateParam

declaration	void r_APP_PID_UpdateParam(void)
overview	Update settings for PID control.
explanation	At the end of PID control, copy the value used in the current operation to the previous value of the led_ch1_pid_param member. The led_ch1_pid_param member to replace is as follows. • led_ch1_pid_param.currents[0] ← led_ch1_pid_param.currents[1]; • led_ch1_pid_param.dutys[0] ← led_ch1_pid_param.dutys[1]; • led_ch1_pid_param.devias[0] ← led_ch1_pid_param.devias[1]; • led_ch1_pid_param.differs[0] ← led_ch1_pid_param.differs[1];
parameter	no
return value	no

2.6.5 R_Config_AAU_Create

<table>
<tr><td>declaration</td><td>void R_Config_AAU_Create(void)</td></tr>
<tr><td>overview</td><td>Initialize AAU.</td></tr>
<tr><td>explanation</td><td>Set the following registers for the AAU component.
<table>
<tr><th>Register</th><th>AAUWINR</th><th>Set Value</th><th>Contents</th></tr>
<tr><td>PER2</td><td>–</td><td>0x40</td><td>Supply clock to AAU.</td></tr>
<tr><td>ACTLREG</td><td>0x00</td><td>0x82</td><td>Calculates algorithm "PI control for DC/DC control 1ch" is select.</td></tr>
<tr><td>ADUTYMX</td><td>0x00</td><td>user defined</td><td>led_ch1_pid_param.duty_limit:Current limit duty</td></tr>
<tr><td>AL1REF</td><td>0x00</td><td>user defined</td><td>led_ch1_pid_param.current_target: Current reference</td></tr>
<tr><td>AKI1</td><td>0x01</td><td>user defined</td><td>led_ch1_pid_param.paramKp1: Proportional coefficient for PI</td></tr>
<tr><td>AKI2</td><td>0x01</td><td>user defined</td><td>led_ch1_pid_param.paramKp2: Proportional coefficient for PI</td></tr>
<tr><td>AL1OFS</td><td>0x01</td><td>user defined</td><td>led_ch1_pid_param.current_offset：Current value offset</td></tr>
<tr><td>AIPL1</td><td>0x02</td><td>user defined</td><td>led_ch1_pid_param.currents[0]: Current value (previous)</td></tr>
<tr><td>ADUTYL1</td><td>0x02</td><td>user defined</td><td>led_ch1_pid_param.dutys[0]: Current duty (previous)</td></tr>
</table></td></tr>
<tr><td>parameter</td><td>no</td></tr>
<tr><td>return value</td><td>no</td></tr>
</table>

2.6.6 R_AAU_Start

<table>
<tr><td>declaration</td><td>void R_AAU_Start(void)</td></tr>
<tr><td>overview</td><td>Start AAU calculation.</td></tr>
<tr><td>explanation</td><td>Set the following registers for the AAU.
<table>
<tr><th>Register</th><th>AAUWINR</th><th>Set Value</th><th>Contents</th></tr>
<tr><td>ADTREG0</td><td>0x00</td><td>Current value</td><td>led_ch1_pid_param.currents[1]: Current value (latest)</td></tr>
<tr><td>ADUTYL1</td><td>0x02</td><td>Current duty (previous)</td><td>((led_ch1_pid_param.duty[1] << 7) & 0xFF80) | ((led_ch1_pid_param.remainder[1] << 5) & 0x0070) : Current duty (previous)</td></tr>
</table>
Sets the latest current measurement value in the ADTREG0 register and starts PI calculation.
Calculation is completed after about 6 clocks after the start of calculation.</td></tr>
<tr><td>parameter</td><td>no</td></tr>
<tr><td>return value</td><td>no</td></tr>
</table>

2.6.7 R_AAU_GetPIControlData

declaration	void R_AAU_GetPIControlData(void)
overview	Get result data of AAU calculation. (PI calculation for DC/DC control)
explanation	Check the end of AAU calculation and get the result. Completion of operation can be confirmed by reading the status of the ACTLREG.OST bit. This function gets the value of the ADTREG0 register, which stores the calculation result when the calculation is completed. After getting it, store the result in the led_ch1_pif_param member below. • led_ch1_pid_param.current_ad ← ADTREG0; PI Calculation result
parameter	no
return value	no

2.6.8 R_Config_DTC_Create

declaration	void R_Config_DTC_Create(void)
overview	Initialize DTC
explanation	Set the following registers for the DTC.

Register	Set Value	Contents
PER1	0x80	Supply clock to DTC
DTCEN0	0x00	Set all DTC activation to disable.
DTCEN1	0x00	
DTCEN2	0x00	
DTCEN3	0x00	
DTCEN4	0x00	
DTCEN5	0x00	
DTCBAR	0xFD	Set the base address of the DTC vector table area to FFD00H (RAM area).
SELHS0	0x09	Set activation factor number 9 (INTAD) to high-speed DTC channel 0.
HDTCCR0	0x61	"HSZ0=1 (Size: 16-bit), HRPTINT0=1 (Enable interrupt), HMODE0=1 (Repeat mode)"
HDTCCT0	0x08	Set DTC transfer count to 8 times.
HDTRLD0	0x08	Set DTC transfer count to 8 times. (for reload setting)
HDTSAR0	0x06A4	Set ADDR2M register (address: 0x06A4) of the A/D converter as the source address of the high-speed DTC.
HDTDAR0	user defined	Set destination address of high-speed DTC.

Finally, Call the R_Config_DTC_Create_UserInit() for control software specific settings.

parameter	no
return value	no

2.6.9 R_Config_DTC_Create_UserInit

declaration	void R_Config_DTC_Create_UserInit(void)
overview	Initialize DTC. (Control software specific settings)
explanation	Initializes the DTC vector table area and sets the DTC activation factor. Set the activation factor number 9 (A/D conversion end) for DTC transfer start trigger. dtc_vectortable[9U] = 0x40U; /* A/D conversion end */
parameter	no
return value	no

2.6.10 R_DTCH0_Start

declaration	void R_DTCH0_Start(void)
overview	Start DTC.
explanation	Enables input of "A/D conversion end" interrupt as a trigger for DTC transfer. DTCEN1 |= 0x40; /* A/D conversion end */
parameter	no
return value	no

2.6.11 R_DTCH0_Stop

declaration	void R_DTCH0_Stop(void)
overview	Stop DTC.
explanation	Prohibit activation of DTC transfer. DTCEN1 &= ~0x40; /* A/D conversion prohibit */
parameter	no
return value	no

2.6.12 R_Config_S12AD0_Create

<table>
<tr><td>declaration</td><td>void R_Config_S12AD0_Create(void)</td></tr>
<tr><td>overview</td><td>Initialize A/D Converter.</td></tr>
<tr><td>explanation</td><td>Set the following registers for the ADC.
<table>
<tr><th>Register</th><th>ADWINR</th><th>Set value</th><th>Contents</th></tr>
<tr><td>PER0</td><td>–</td><td>0x20</td><td>A/D converter enabled.</td></tr>
<tr><td>ADCSR</td><td>0x00</td><td>0x1200</td><td>ADCS[1:0]=00B (Single scan mode),
ADIE=1(Enable INTAD interrupt),
TRGE=1 (Enable A/D conversion to be started by trigger)</td></tr>
<tr><td>ADANSA0</td><td>0x00</td><td>0x0004</td><td>Set ANI2 as conversion target.</td></tr>
<tr><td>ADSTRGR</td><td>0x01</td><td>0x0400</td><td>TRSA[5:0]=000100B (A/D conversion start Trigger:
INTTRD0_IFA)</td></tr>
<tr><td>ADHVREFCNT</td><td>0x08</td><td>0x11</td><td>Select reference voltage. (AVREFM and AVREFP)</td></tr>
<tr><td>ADCER</td><td>0x00</td><td>0x0000</td><td>ADRFMT=0 (Right aligned). DIAGM=0 (Disable self-diagnosis)</td></tr>
<tr><td>ADDISCR</td><td>0x07</td><td>0x00</td><td>Disable disconnection detect function.</td></tr>
<tr><td>ADSSTR2</td><td>0x0E</td><td>0x0D</td><td>Sampling state setting for ANI2. (13.5 state)</td></tr>
</table>
After setting the above registers, set the ANI2, AVREFP, and AVREFM pins.
ANI2:
• PMC8.PMC82 set to the "analog input" (PMC82 = 1)
• PM8.PM82 set to the "input mode" (PM82 = 1)

AVREFP :
• PMC3.PMC33 set to the "analog input" (PMC33 = 1)
• PM3.PM33 set to the "input mode" (PM33 = 1)

AVREFM :
• PMC3.PMC34 set to the "analog input" (PMC34 = 1)
• PM3.PM34 set to the "input mode" (PM34 = 1)

Finally, Call the R_Config_S12AD0_Create_UserInit() for control software specific settings.</td></tr>
<tr><td>parameter</td><td>no</td></tr>
<tr><td>return value</td><td>no</td></tr>
</table>

2.6.13 R_Config_S12AD0_Create_UserInit

declaration	void R_Config_S12AD0_Create_UserInit(void)
overview	Initialize A/D Converter. (Control software specific settings)
explanation	Performs initialization processing (specific settings) for the A/D converter. This control software does not perform any processing.
parameter	no
return value	no

2.6.14 R_Config_S12AD0_Start

declaration	void R_Config_S12AD0_Start(void)
overview	A/D converter operation start.
explanation	Start A/D conversion. • Clear the interrupt request flag bit in IF1H.ADIF to 0. • MK1H. Clear the ADMK interrupt mask bit to 0 to enable interrupt generation. • Set ADWINR register to 0x00. • Set the ADCSR.TRGE bit to 1. (enable the setting to start A/D conversion by trigger input)
parameter	no
return value	no

2.6.15 R_Config_S12AD0_Stop

declaration	void R_Config_S12AD0_Stop(void)
overview	End of A/D conversion operation.
explanation	End of A/D conversion. • Set ADWINR register to 0x00. (Set window 0) • Clears ADCSR.TRGE setting. (Clears setting for starting A/D conversion by trigger input) • Clear ADCSR.ADST setting. (Stop A/D conversion by A/D converter) • Set the MK1H.ADMK interrupt mask bit to 1 to disable interrupt generation. • Clear the interrupt request flag bit in IF1H.ADIF to 0.
parameter	no
return value	no

2.6.16 r_Config_S12AD0_Interrupt

declaration	static void __near r_Config_S12AD0_Interrupt(void)
overview	A/D converter interrupt handler.
explanation	Performs interrupt processing when an A/D conversion end interrupt (INTAD) occurs. The INTAD signal is used as a trigger for the high-speed DTC. This interrupt handler is executed after 8 data transfers are completed. • Disable generation of A/D interrupts. • Calculate the average value of 8 A/D conversion results transferred to RAM. • Enable DTC. (Enable DTC activation triggered by A/D conversion end interrupt) • Set the management variable flg_ad_update to 1. (Update A/D conversion result) • Enable A/D interrupt.
parameter	no
return value	no

2.6.17 R_Config_TRD0_Create

<table>
<tr><td>declaration</td><td>void R_Config_TRD0_Create(void)</td></tr>
<tr><td>overview</td><td>Initialize Timer RDe.</td></tr>
<tr><td>explanation</td><td>Set the following registers for the Timer RDe.
<table>
<tr><th>Register</th><th>Set value</th><th>Contents</th></tr>
<tr><td>TRDMR</td><td>0x20</td><td>Set TRDGRD0 as TRDGRB0 buffer register.</td></tr>
<tr><td>TRDPMR</td><td>0x05</td><td>TRDIOB0 pin is used in PWM function.</td></tr>
<tr><td>TRDFCR</td><td>0xA0</td><td>Set the extended PWM mode.</td></tr>
<tr><td>TRDCR0</td><td>0x20</td><td>Set count source (fTRD) to no division of fCLK.</td></tr>
<tr><td>TRDEMR0</td><td>0x20</td><td>DTENB=1 (enable dithering function), GTENB=0 (disable gate function), CCLV[1:0]=00B (disable count restart function)</td></tr>
<tr><td>TRDGRA0</td><td>0x018F</td><td>PWM frequency = 5μs (200kHz)</td></tr>
<tr><td>TRDGRB0</td><td>0x0027</td><td>PWM duty = 92% (400 - 24)</td></tr>
<tr><td>TRDGRC0</td><td>0x018F</td><td>Set same value as TRDGRA0.</td></tr>
<tr><td>TRDGRD0</td><td>0x0027</td><td>Set same value as TRDGRB0. (buffer register)</td></tr>
<tr><td>PWMDLY0</td><td>0x0000</td><td>Disable PWM delay function.</td></tr>
<tr><td>TRDOER1</td><td>0xFD</td><td>EB0 = 0 (Enable TRDIOB0 pin output)</td></tr>
<tr><td>TRDOCR</td><td>0x00</td><td>TOB0 = 0 (TRDIOB0 initial output level setting)</td></tr>
<tr><td>TRDPOCR0</td><td>0x01</td><td>POLB = 1 (TRDIOB0 output level setting to High active)</td></tr>
<tr><td>TRDDNR0</td><td>0x00</td><td>Set the calculated dithering count value.</td></tr>
<tr><td>TRDOER2</td><td>0x00</td><td>Disable forced cut off function.</td></tr>
<tr><td>TRDELC</td><td>0x00</td><td>Do not use forced cut off function.</td></tr>
<tr><td>TRDIER0</td><td>0x00</td><td>Disable interrupt. (INTTRD0)</td></tr>
</table>
After setting the above, set the port for the TRDIOB0 pin.
• Set POM1.POM11 to normal output mode. (POM11 = 0)
• Set the initial output of P1.P11 to 0. (P11 = 0)
• Set PM1.PM11 to output mode. (PM11 = 0)

Finally, Call the R_Config_TRD0_Create_UserInit() for control software specific settings.</td></tr>
<tr><td>parameter</td><td>no</td></tr>
<tr><td>return value</td><td>no</td></tr>
</table>

2.6.18 R_Config_TRD0_Create_UserInit

declaration	void R_Config_TRD0_Create_UserInit(void)
overview	Initialize Timer RDe0. (Control software specific settings)
explanation	Set the start count value of the timer RD0 and the initial value of the dithering count. Register / Set value / Contents TRD0 / 0x0000 / Set initial value of TRD0. TRDDNR0 / user defined / Use the setting value of led_ch1_pid_param.dither_count.
parameter	no
return value	no

2.6.19 R_Config_TRD0_Start

declaration	void R_Config_TRD0_Start(void)
overview	Start Timer RDe0.
explanation	Start Timer RD0 counting. • Set the TRDSTR.CSEL0 bit to 1. (Continue timer RD0 counting even after a compare match with the TRDGRA0 register) • Set TRDSTR.TSTART0 bit to 1. (Start timer RD0 count)
parameter	no
return value	no

2.6.20 R_Config_TRD0_Stop

declaration	void R_Config_TRD0_Stop(void)
overview	End of timer RD0 count operation.
explanation	Stops timer RD0 counting. • Set the TRDSTR.CSEL0 bit to 0. (Stop the timer RD0 count on a compare match with the TRDGRA0 register) • Set TRDSTR.TSTART0 bit to 0. • Clear the interrupt mask bit in MK0H.TRDMK0. (Disable interrupt processing) • Clear the interrupt request flag bit of IF0H.TRDIF0. (Clear the interrupt flag) • Read the TRDSR0 register. • Write 00H to TRDSR0 register. (Clear interrupt notification)
parameter	no
return value	no

2.6.21 R_Config_PORT_Create

declaration	void R_Config_PORT_Create(void)
overview	Initialize I/O ports.
explanation	Initialize the I/O ports of the RL78/F2x. The ports controlled by the timer RDe and A/D converter are set using the functions below. • Timer RDe : 2.6.17 R_Config_TRD0_Create • A/D converter : 2.6.12 R_Config_S12AD0_Create
parameter	no
return value	no

2.6.22 R_Config_TRD0_Set_TRD0_ReloadTrigger

declaration	MD_STATUS R_Config_TRD0_Set_TRD0_ReloadTrigger(st_extpwm_buffer_registers_t * buffer)
overview	Update timer RDe setting.

explanation	Sets the timer RDe buffer register and requests a reload to the compare register.
	Also, set the TRDDNR0 register to request rewriting of the dithering count.
	• TRDGRD0 register (buffer register of TRDGRB0 register)
	• TRDDNR0 register (timer RD0 dithering number register)
	Set the TRDRDT0.RDT bit to 1 to request a register reload.
	
	[Examples of using functions]
	The type definition of st_extpwm_buffer_registers_t used by argument *buffer is as follows.
	`typedef struct {`
	`  uint16_t trdgrcn; // TRDGRCn setting value (n=0,1)`
	`  uint16_t trdgrdn; // TRDGRDn setting value (n=0,1)`
	`  uint16_t trdcmpdn; // TRDCMPDn setting value (n=0,1)`
	`  uint8_t trddnrn; // TRDDNRn setting value (n=0,1)`
	`} st_extpwm_buffer_registers_t;`
	
	Register the value to be set in the register when reloading in the argument *buffer.
	After registration, execute this function.
	`// set next Timer compare count`
	`buffer.trdgrcn = 0x0000;`
	`buffer.trdgrdn = 0x018F;`
	`buffer.trdcmpdn = 0x0000;`
	`// set next dithering count`
	`buffer->trddnrn = 0x0A;`
	
	`// Timer reload request`
	`R_Config_TRD0_Set_TRD0_ReloadTrigger(&buffer);`

parameter	st_extpwm_buffer_registers_t * buffer
return value	MD_STATUS code

2.7 Flowchart

2.7.1 Start-up Sequence

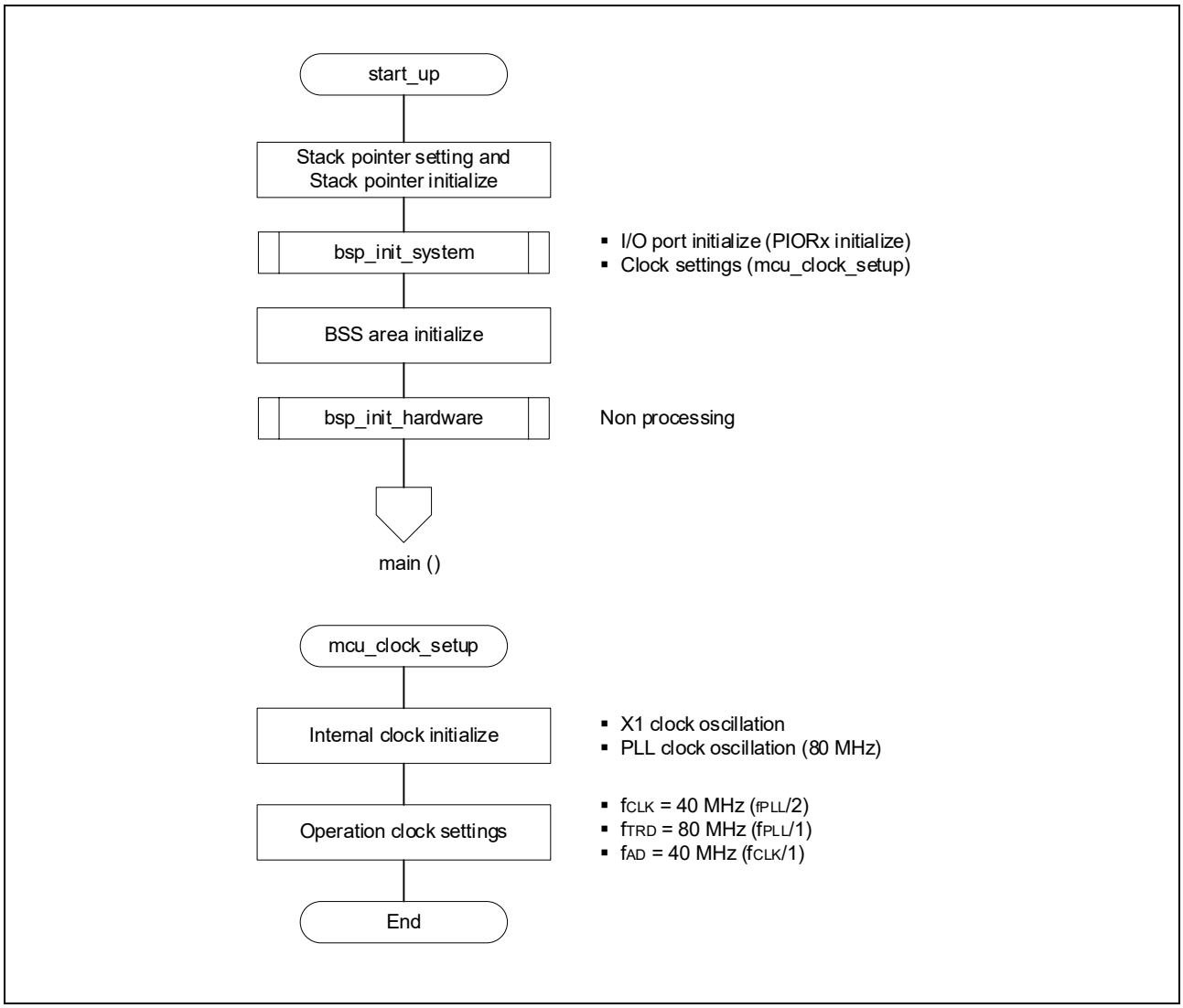


Figure 2-4 Start-up Sequence

2.7.2 System Control Flowchart

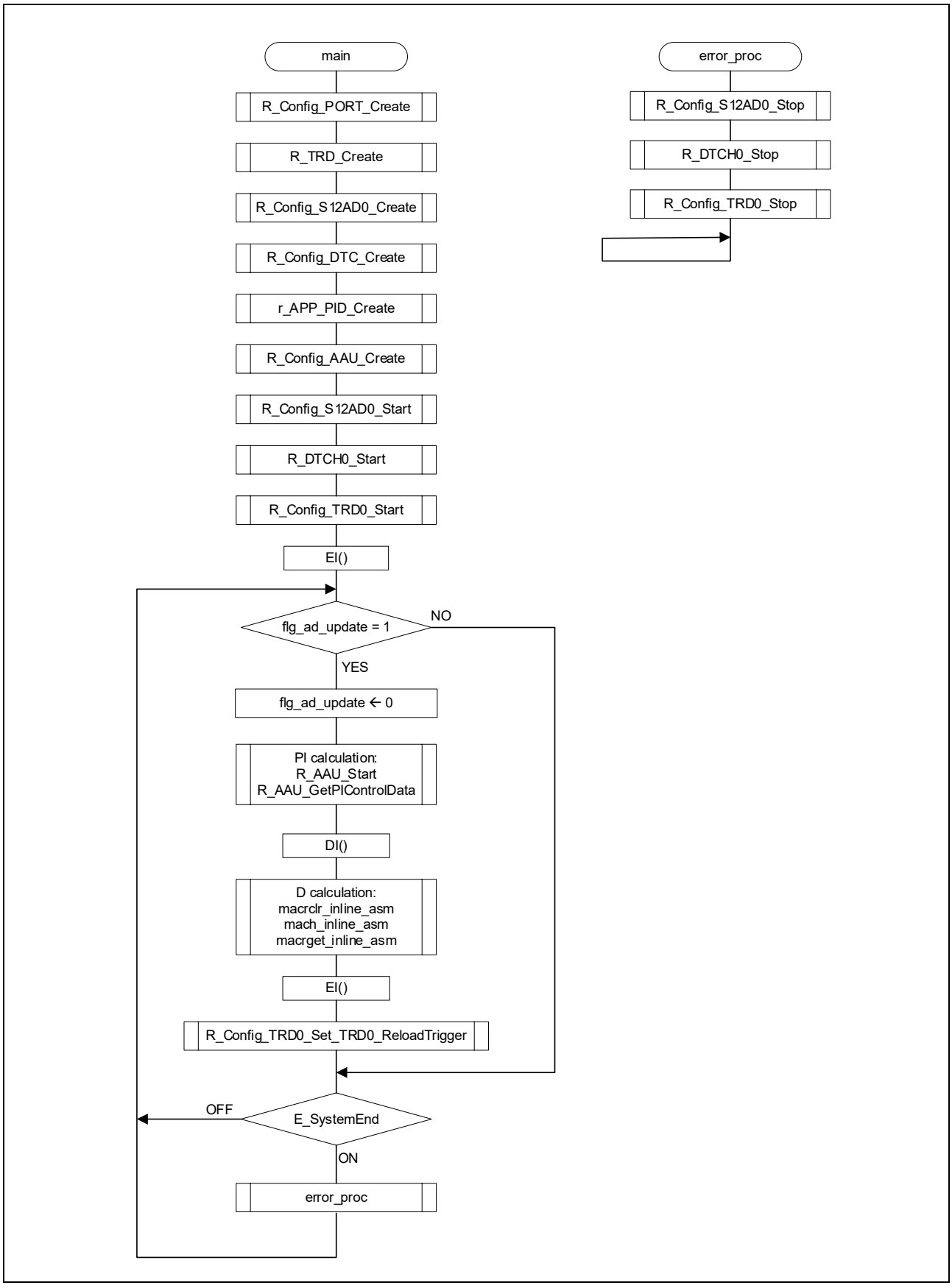


Figure 2-5 System Control Flowchart

2.7.3 Initialize PORT Flowchart

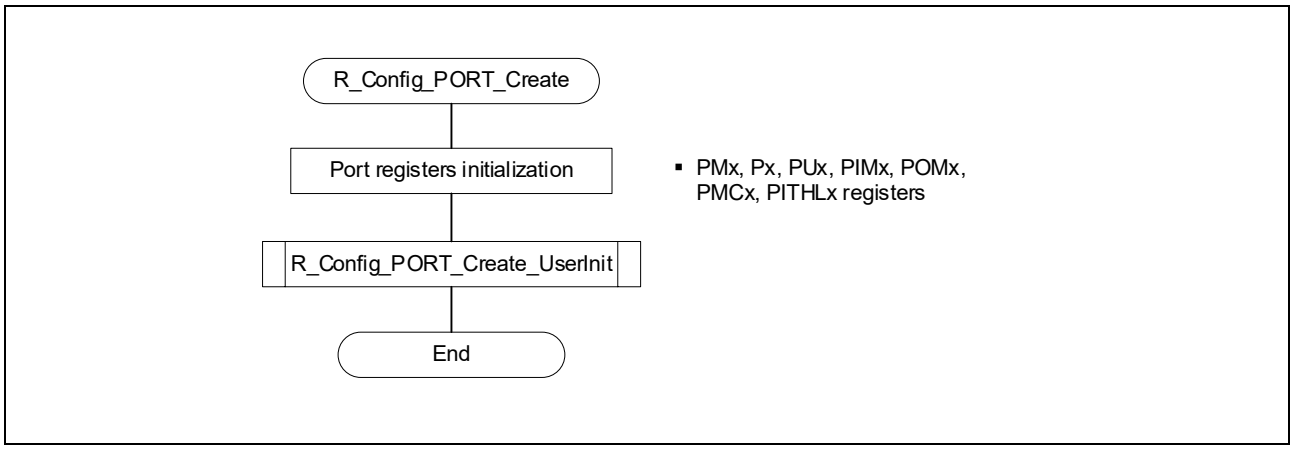


Figure 2-6 Initialize PORT Flowchart

For the initialization of the pins used by the timer RDe and A/D converter, refer to the timer RDe and A/D converter initialization flowchart.

2.7.4 Initialize Timer Flowchart

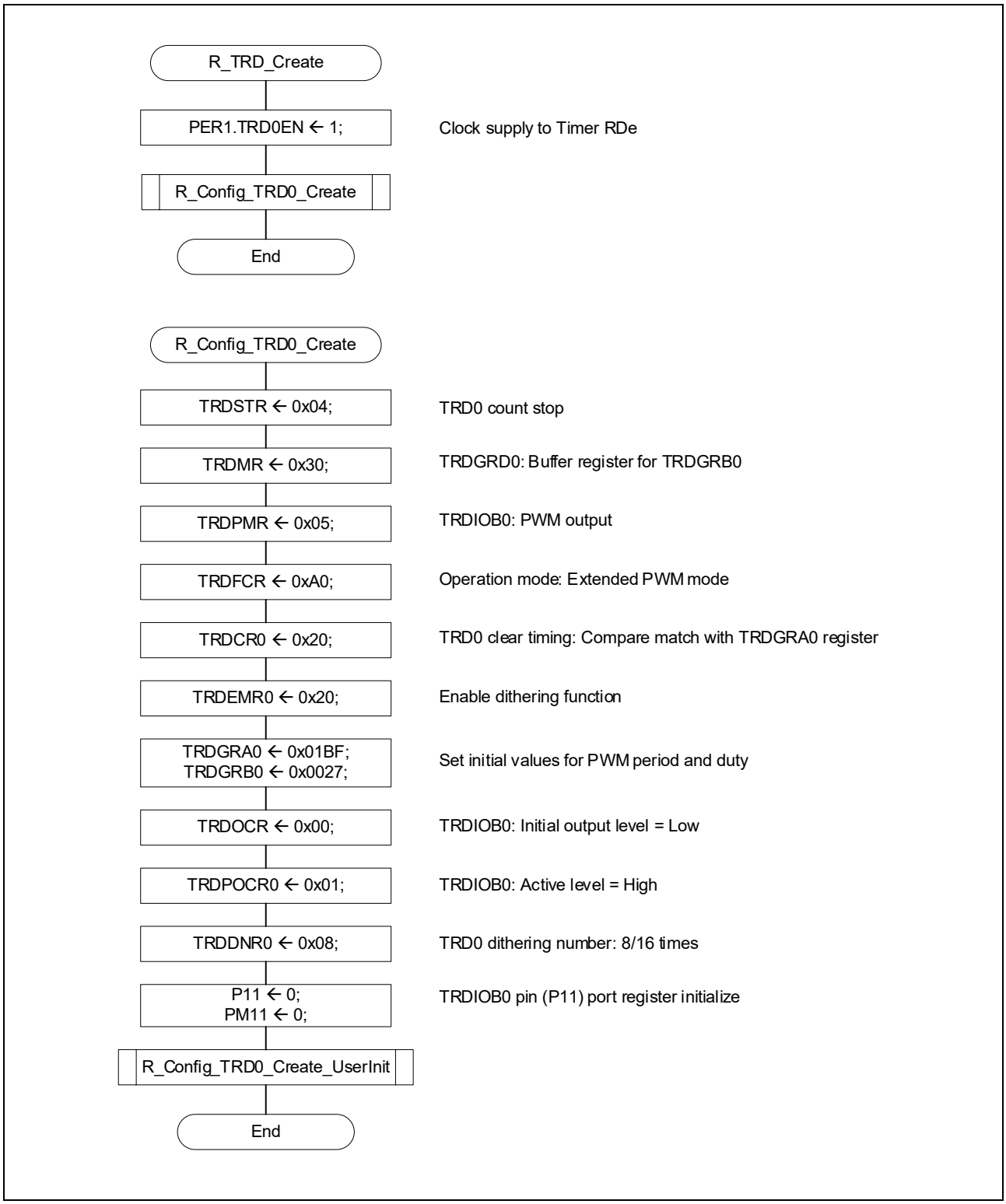


Figure 2-7 Initialize Timer Flowchart

2.7.5 Initialize A/D Converter Flowchart

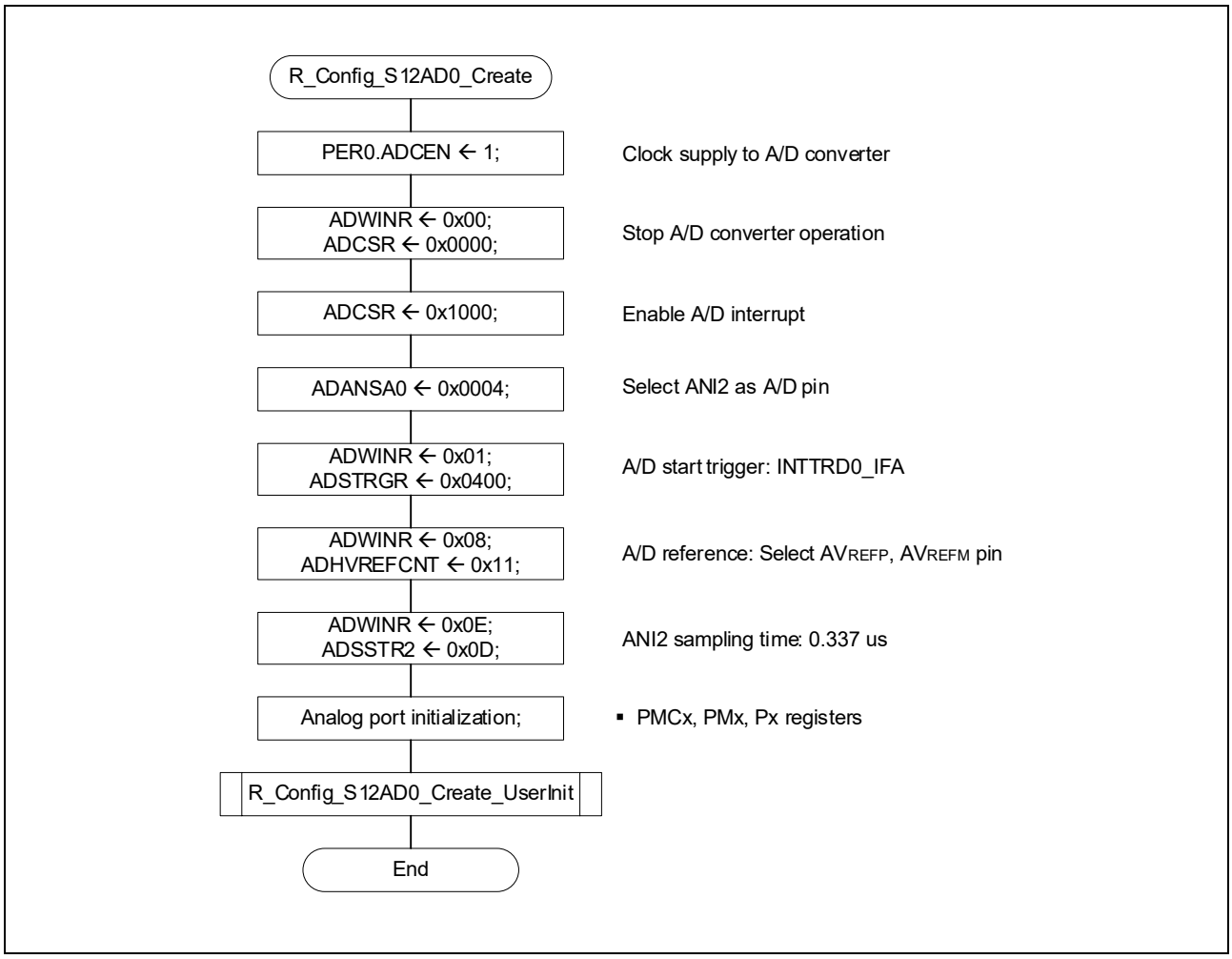


Figure 2-8 Initialize A/D Converter Flowchart

2.7.6 Initialize DTC Flowchart

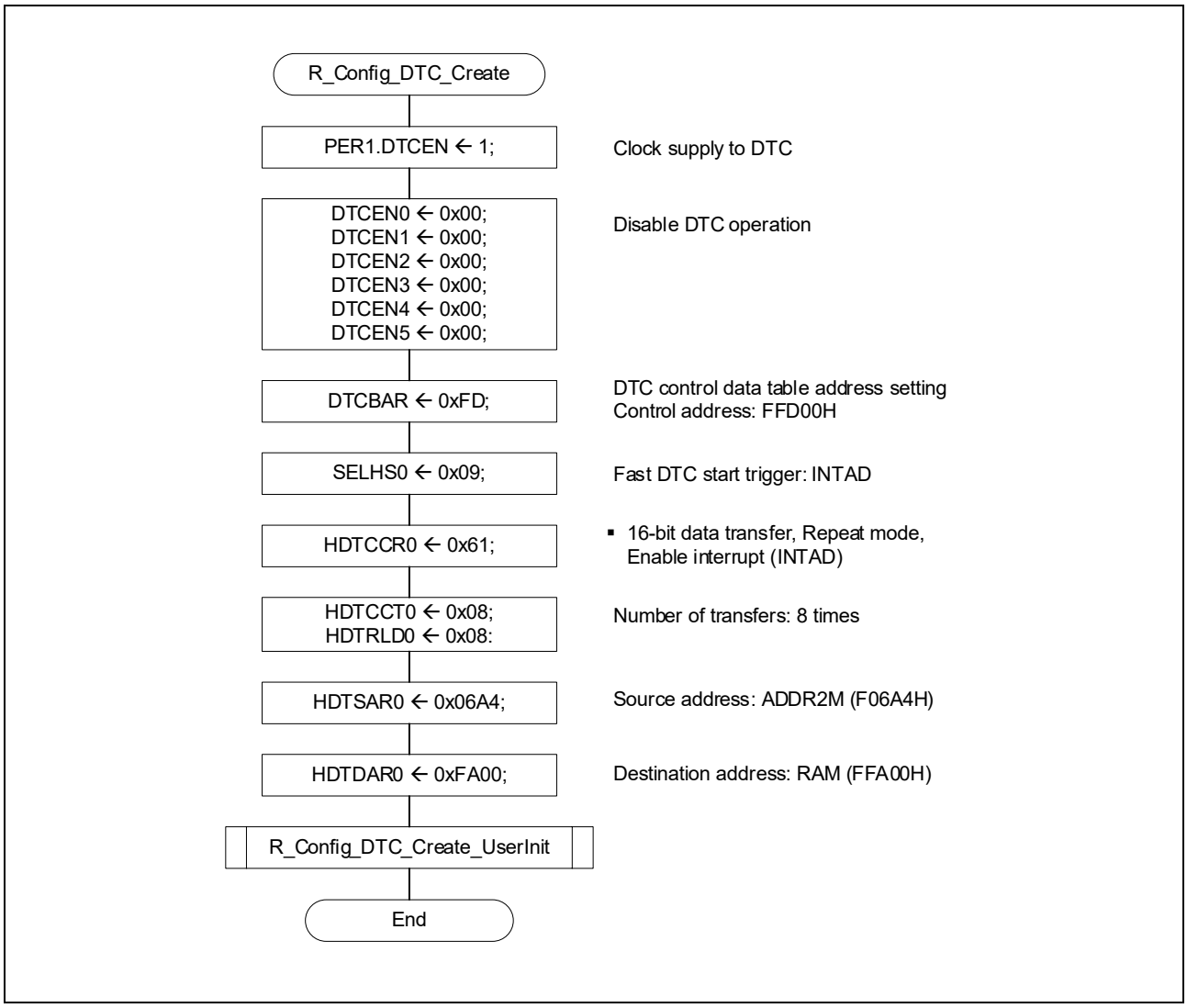


Figure 2-9 Initialize DTC Flowchart

2.7.7 Initialize AAU Flowchart

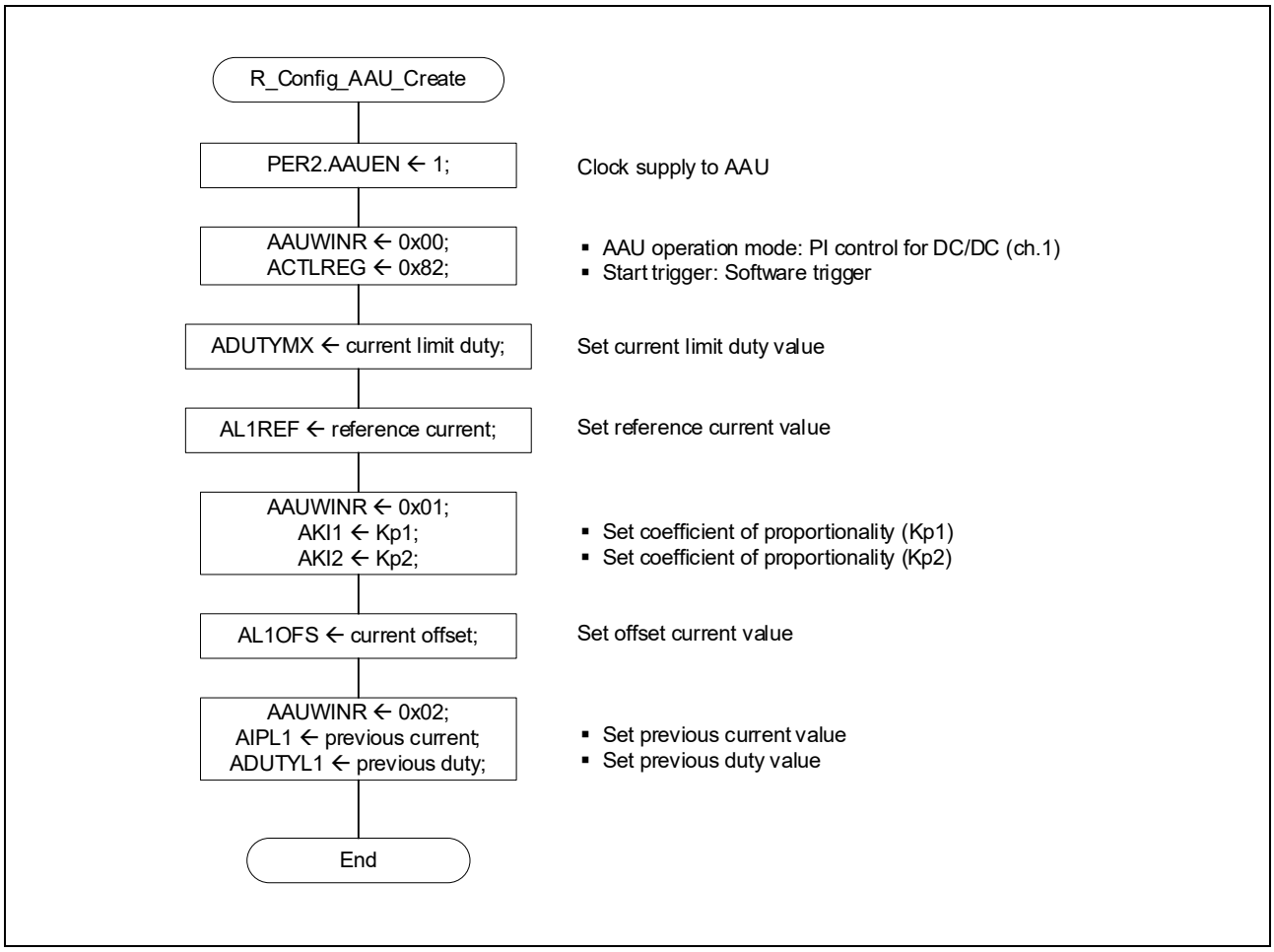


Figure 2-10 Initialize AAU Flowchart

2.7.8 Timer Control Flowchart

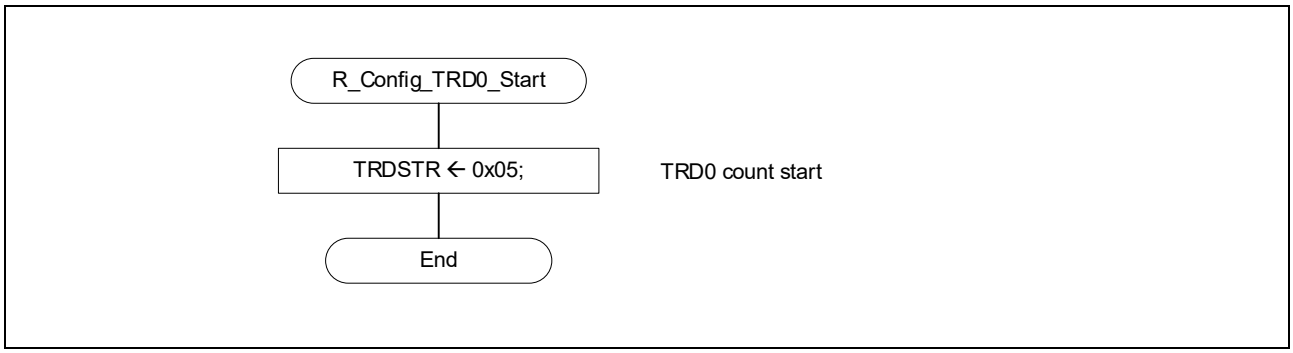


Figure 2-11 Timer Start Flowchart

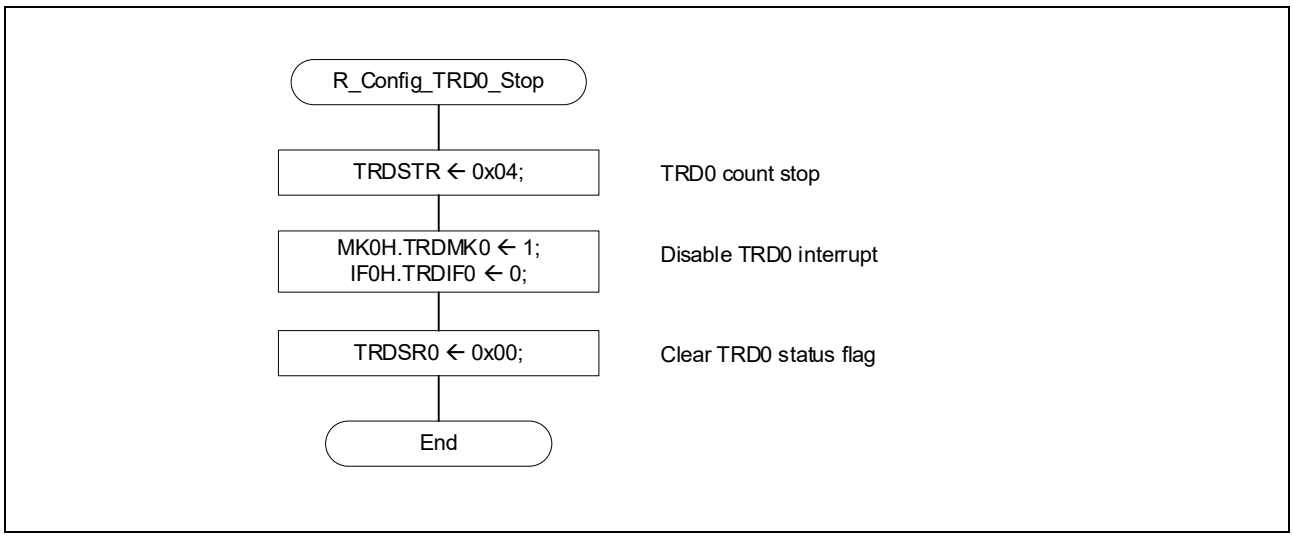


Figure 2-12 Timer Stop Flowchart

2.7.9 A/D Converter Control Flowchart

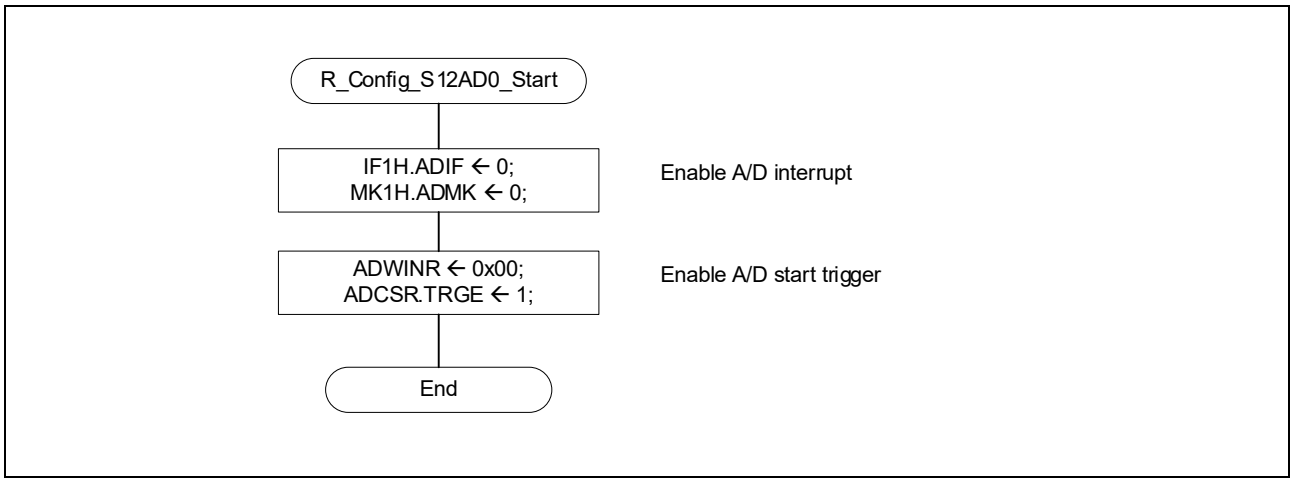


Figure 2-13 A/D Converter Start Flowchart

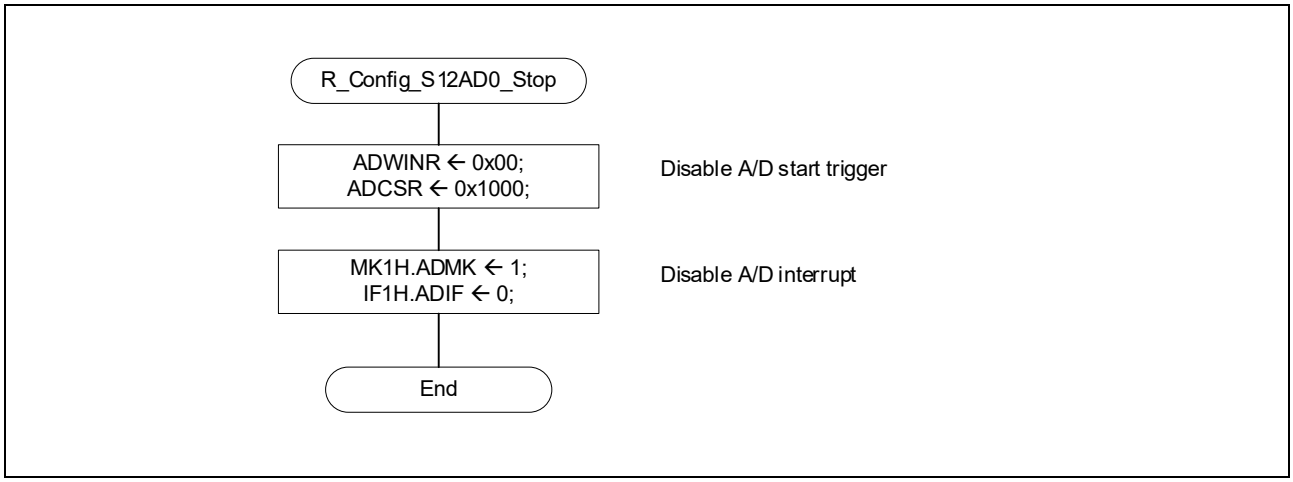


Figure 2-14 A/D Converter Stop Flowchart

2.7.10 DTC Control Flowchart

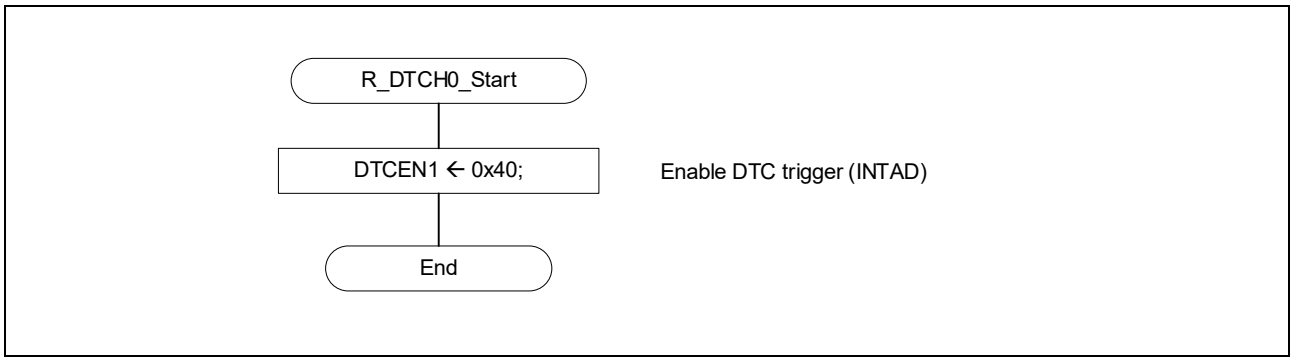


Figure 2-15 DTC Start Flowchart

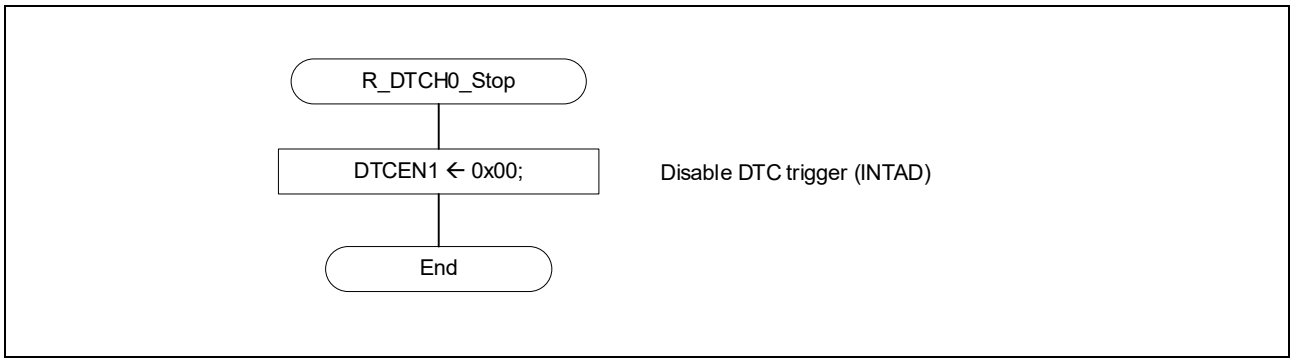


Figure 2-16 DTC Stop Flowchart

2.7.11 AAU Control Flowchart

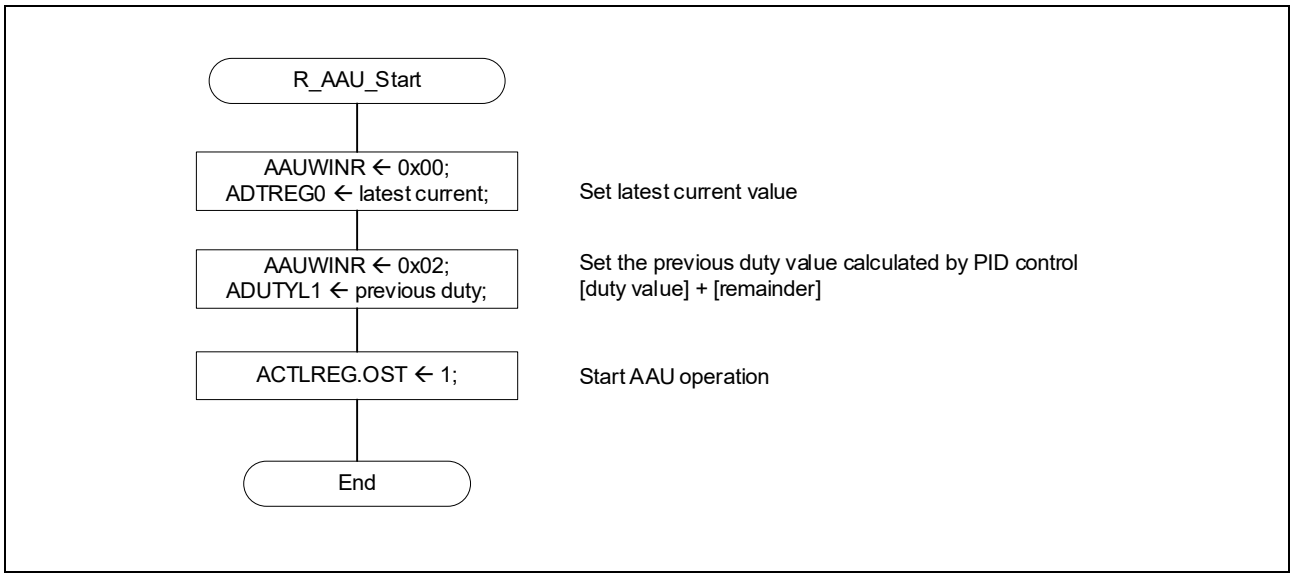


Figure 2-17 AAU Calculation Start Flowchart

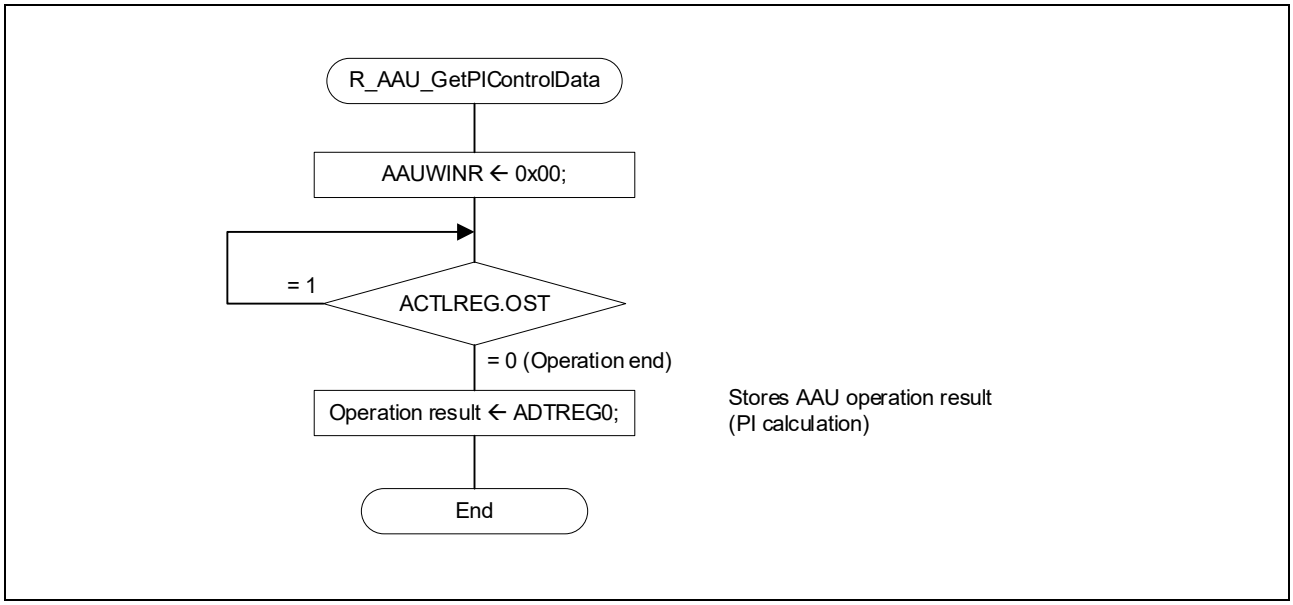


Figure 2-18 Read AAU Calculation Result Flowchart

3. References

References documents for this application note are shown below. Be sure to obtain the latest version of each document from the website of Renesas Electronics Corporation when designing.

- RL78/ F23, F24 User's Manual: Hardware Rev. 1.00
- RL78 family User's Manual: Software Rev. 2.30

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	2023.07.31	-	First edition issued

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/.