# RL78/G24

## FAA LED Control Library Installation Guide

### Introduction

This application note describes the LED control library using the Flexible Application Accelerator (FAA) (hereinafter referred to as the LED control library).

The FAA is built into the RL78/G24 microcontroller and operates as a processor independent of the CPU.

### Target Device

RL78/G24

### Related Documents

- RL78/G24 User’s Manual: Hardware (R01UH0961)
- CS+ Integrated Development Environment User’s Manual: Project Operation (R20UT4691)
- RL78 Smart Configurator User's Guide: CS+ (R20AN0580)
- RL78 Smart Configurator User's Guide: IAR (R20AN0581)

Contents

# 1. LED Control Library Features

This chapter describes the features of the LED control library.

## 1.1 LED Constant Current Control

This library provides up to 4 channels of LED constant-current control using the timer KB PWM output feature, and LED constant-current control is realized by feedback processing based on PI (proportional-integral) control. This eliminates the need for an external IC dedicated to LED constant-current control and reduces design costs.

## 1.2 Flexible Application Accelerator (FAA)

Feedback processing to realize LED constant-current control is performed by the FAA, which is built into the RL78/G24 microcontroller and operates as a processor independent of the CPU. This enables high-speed LED dimming control processing without occupying the CPU.

For details of FAA, refer to RL78/G24 User's Manual: Hardware (R01UH0961J).

## 2. Tools that can use the LED Control Library

This library is provided as source code by the code generation function of the Smart Configurator. The generated source code is added to the user's project in the integrated development environment and built for integration.

Table 2.1, Table 2.2 shows a list of tools for using this library. For details on the usage procedure, refer to 5 LED Control Library Usage Procedure.

Table 2.1 CC-RL Environment (CS+ for CC)

| Item | Description |
|---|---|
| Integrated Development Environment (IDE) | CS+ for CC V8.09.00 (Renesas IDE) |
| Configurator (SC) | Renesas Smart Configurator for RL78 V1.4.0 |
| Compiler | CC-RL V1.12 |

Table 2.2 IAR Environment (IAR Embedded Workbench for Renesas RL78)

| Item | Description |
|---|---|
| Integrated Development Environment (IDE) | IAR Embedded Workbench for Renesas RL78 V5.10.3 |
| Compiler | IAR C/C++ Compiler for Renesas RL78 V5.10.3.2716 (5.10.3.2716) |
| Configurator (SC) | Renesas Smart Configurator for RL78 V1.11.0 |

## 3. Folder Structure of LED Control Library

The folder structure of this library is shown below. Each file is provided by the code generation function of the Smart Configurator.

Table 3.1 Folder Structure

| Folder, File Name | Description |
|---|---|
| smc_gen | SC generation folder |
| ¥{ConfigName} | FAA component folder |
| {ConfigName}_common.c | FAA common feature source file |
| {ConfigName}_common.h | FAA common feature header file |
| {ConfigName}_common.inc | FAA common feature include file |
| {ConfigName}_LEDControl.c | LED control feature source file |
| {ConfigName}_LEDControl.h | LED control feature header file |
| {ConfigName}_src.dsp | FAA DSP assembler source file |

# 4. Using Resources

## 4.1 Peripheral Features

This library uses the following peripheral features.

- Flexible application accelerator
- Programmable gain amplifier
- A/D converter
- D/A converter
- Comparator
- 16-bit timer KB

## 4.2 ROM / RAM Size

### 4.2.1 Integrated Development Environment CS+ (CS+ for CC)

The ROM/RAM size when built with the following options is given below for reference.

Compiler Options

-cpu=S3 -memory_model=medium –Odefault

Linking Options

-NOOPtimize

Table 4.1 ROM / RAM Size

| Number of LED channels | ROM | RAM | FAACODE | FAADATA |
|---|---|---|---|---|
| 1 (High-speed Operation) | 1,419 [byte] | 0 [byte] | 376 [byte] | 172 [byte] |
| 1 | 1,730 [byte] | 0 [byte] | 336 [byte] | 172 [byte] |
| 2 | 2,104 [byte] | 0 [byte] | 516 [byte] | 204 [byte] |
| 3 | 2,393 [byte] | 0 [byte] | 608 [byte] | 236 [byte] |
| 4 | 2,624 [byte] | 0 [byte] | 700 [byte] | 268 [byte] |

Note. Since the size varies depending on the settings of the smart configurator, the maximum values are listed in the table above.

### 4.2.2 Integrated Development Environment IAR

The ROM/RAM size when built with the following options is given below for reference.

Compiler Options

__core = s3

__code_model = far

__data_model = near

Table 4.2 ROM / RAM Size

| Number of LED channels | Optimization level | ROM | RAM | FAACODE | FAADATA |
|---|---|---|---|---|---|
| 1 | High (Balance) | 1,532 [byte] | 0 [byte] | 336 [byte] | 172 [byte] |
| 1 | Low | 1,612 [byte] | 0 [byte] | 336 [byte] | 172 [byte] |
| 2 | Low | 1,841 [byte] | 0 [byte] | 512 [byte] | 204 [byte] |
| 3 | Low | 2,034 [byte] | 0 [byte] | 600[byte] | 236 [byte] |
| 4 | Low | 2,179 [byte] | 0 [byte] | 692 [byte] | 268 [byte] |

Note. Since the size varies depending on the settings of the smart configurator, the maximum values are listed in the table above.

# 5. LED Control Library Usage Procedure

The procedure for using this library is shown below.

## 5.1 CC-RL Environment

### 5.1.1 Integrated Development Environment CS+ (CS+ for CC)

Download CS+ from the Renesas Electronics Website.

[Renesas Electronics Website]

https://www.renesas.com/us/en/products/software-tools/tools.html

Refer to the following User's Manual for basic CS+ operations.

- CS+ Integrated Development Environment User’s Manual: Project Operation (R20UT4691)

### 5.1.2 Smart Configurator

Download "RL78 Smart Configurator" and "CS+ RL78 Smart Configurator Communication Plug-in" from the following URL. CS+ RL78 Smart Configurator Communication Plug-in is required to register the source generated by the Smart Configurator to CS+.

https://www.renesas.com/rl78-smart-configurator

After launching the installer, follow the installer's instructions. Install with administrator privileges.

### 5.1.3 LED Control Library

This section describes the procedure for using the LED control library on the Smart Configurator.

For basic operation of the Smart Configurator, refer to the User's Guide for basic Smart Configurator operations.

- RL78 Smart Configurator User's Guide: CS+ (R20AN0580)

#### 5.1.3.1 Starting the Smart Configurator

Create a new project or load an existing project in CS+ and start the Smart Configurator from the CS+ screen.

#### 5.1.3.2 Clock Settings

Select the "Clocks" page of the "Smart Configurator View" and set the "Timer clock" and " $f_{CLK}$" outputs as follows.

- Timer clock : 96000[kHz]
- $f_{CLK}$ : 48000[kHz]

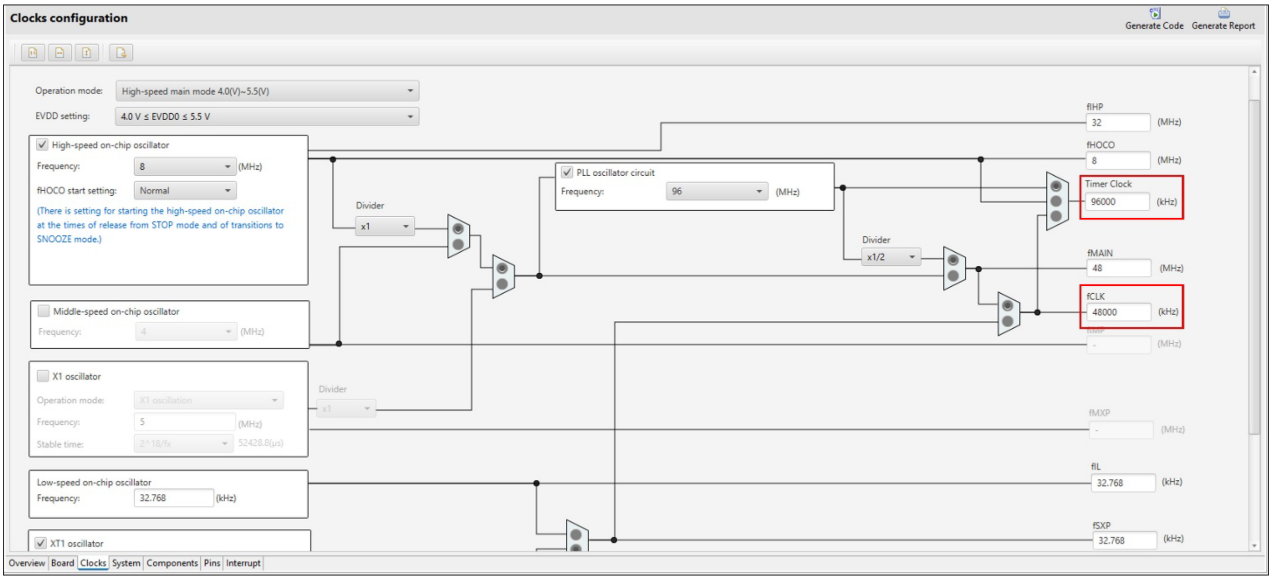



Figure 5-1 Clock Settings

### 5.1.3.3 Adding FAA Components

Select the "Components" page of the "Smart Configurator View" and click the "Add component" button.

Next, add the "Flexible Application Accelerator" component from the "Software Component Selection" page.

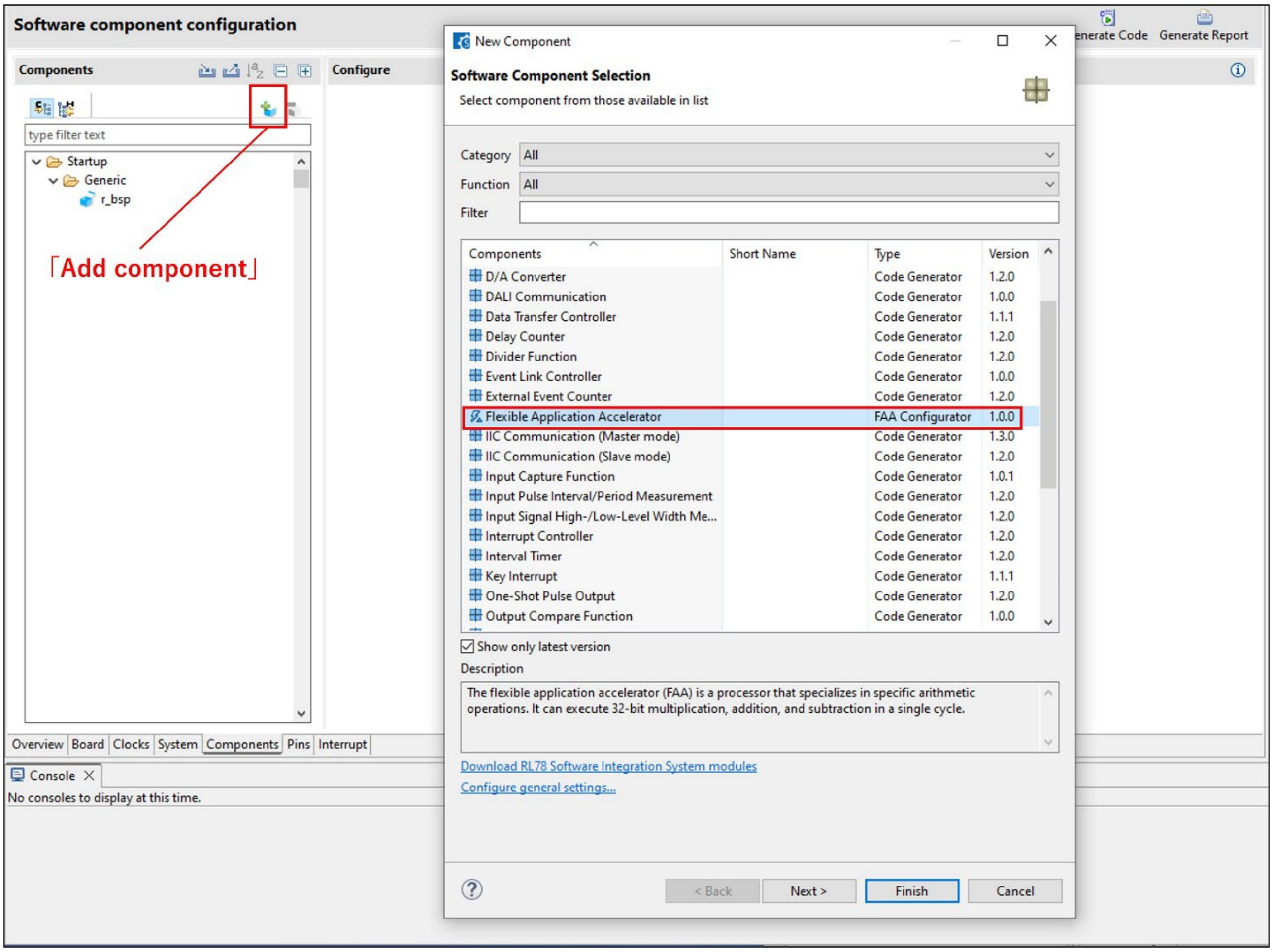
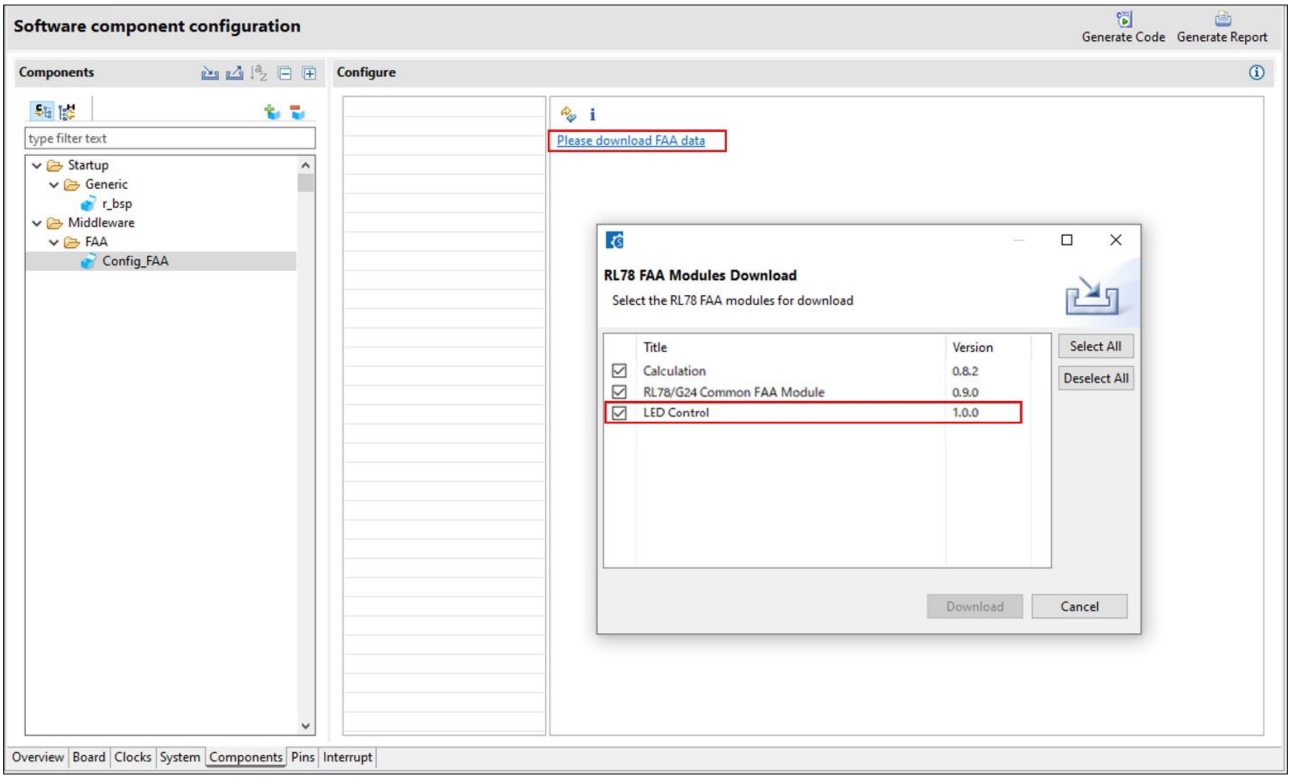



Figure 5-2 Adding FAA Components

### 5.1.3.4 Downloading FAA Modules

Click on "Please download FAA data" displayed on the screen to see the FAA modules available for download. Select "LED Control" to download.




Figrue 5-3 Downloading FAA Modules

#### 5.1.3.5 FAA Module Configuration

Select the "LEDControl" module from the list of downloaded FAA modules to display the configuration screen.

Configure the configuration settings according to the user environment. For details on each configuration item, see **7 Configuration Specifications**.

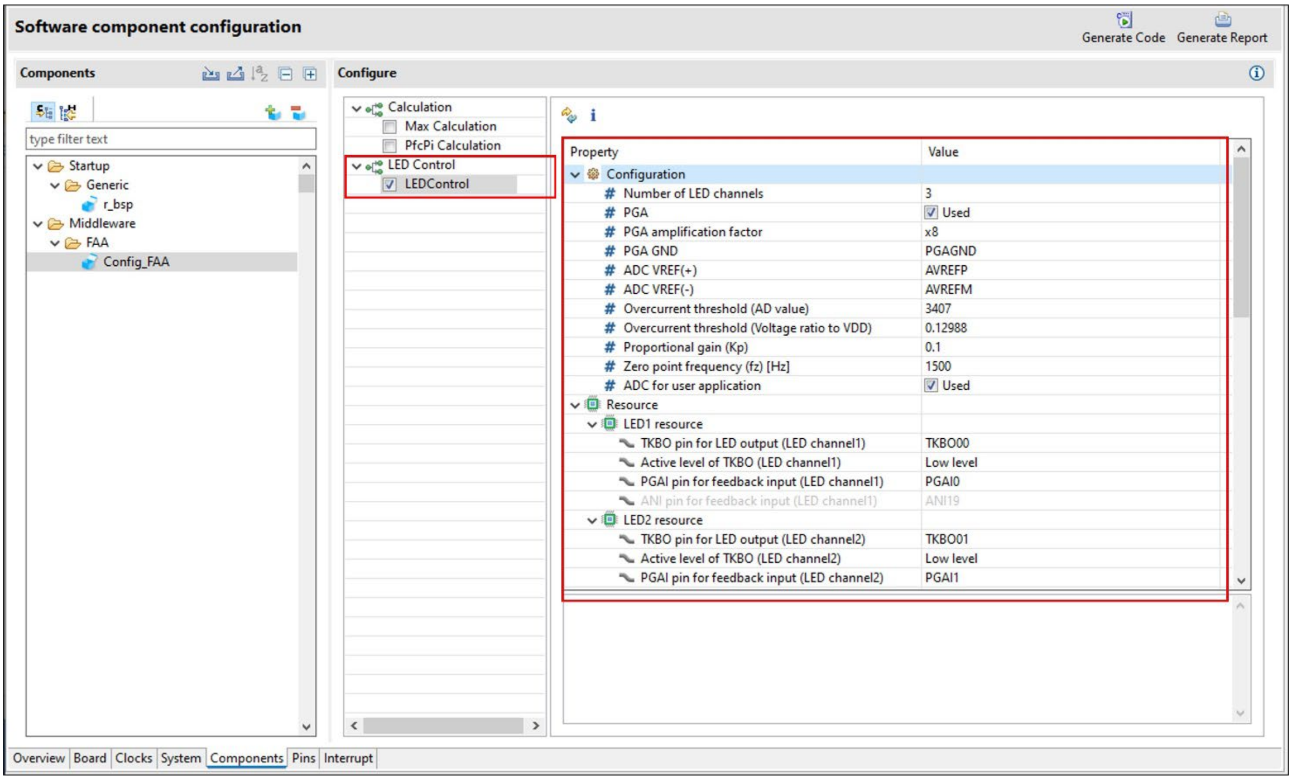



Figure 5-4 FAA Module Configuration

#### 5.1.3.6 Code Generation

After setting the configuration, click the "Generate code" button to generate the source code for the LED control library. The generated source code will be automatically registered in the user project on CS+.

Implement the API function calls provided by the library in the user application. For details on the various API functions, see **8.2 API Function Specifications**.

### 5.1.4 Project settings for using FAA (CS+)

The following project settings are required to use FAA in CS+.

#### 5.1.4.1 FAA assemble option setting

Set additional option [-macro_identify exact] in [CC-RL (Build Tool)] -> [FAA Assemble Options] -> [Other additional options].

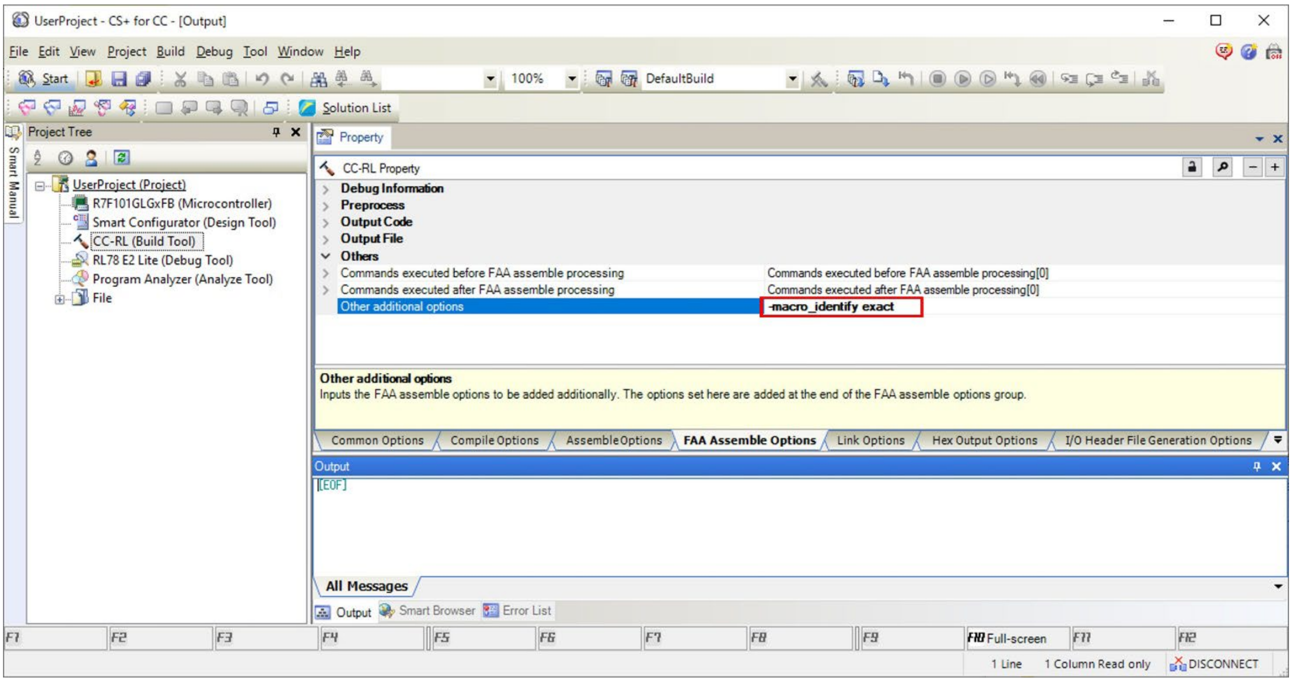



Figure 5-5 FAA assemble option setting

### 5.1.4.2 Setting of section start address

[CC-RL(Build Tool)]-> [Link Options] -> [Section start address] should be set as follows.

Table 5.1 Setting of section start address

| Address | Section |
|---|---|
| 0xFD800 | FAACODER |
| 0xFE800 | FAADATAR |

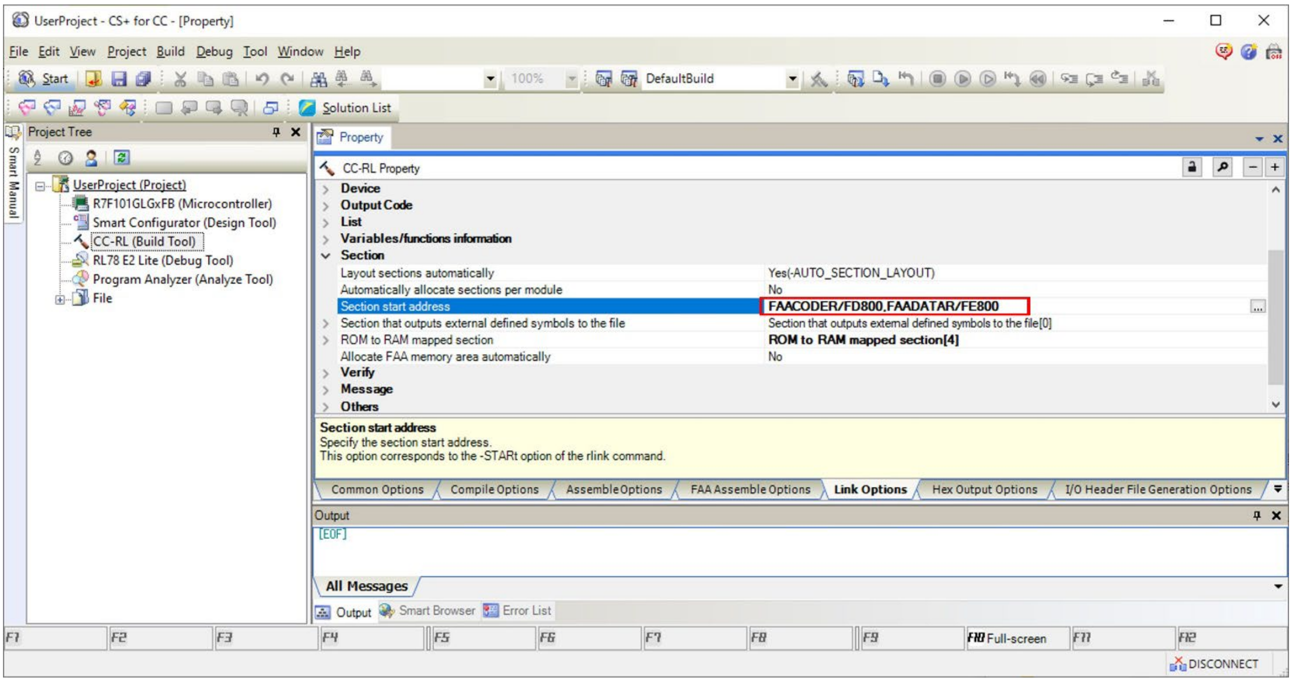



Figure 5-6 Setting of section start address

### 5.1.4.3 Settings for sections to be mapped from ROM to RAM

Add the following settings to [CC-RL (Build Tool)] -> [Link Options] -> [Section to map from ROM to RAM].

- "FAACODE=FAACODER"
- "FAADATA=FAADATAR"

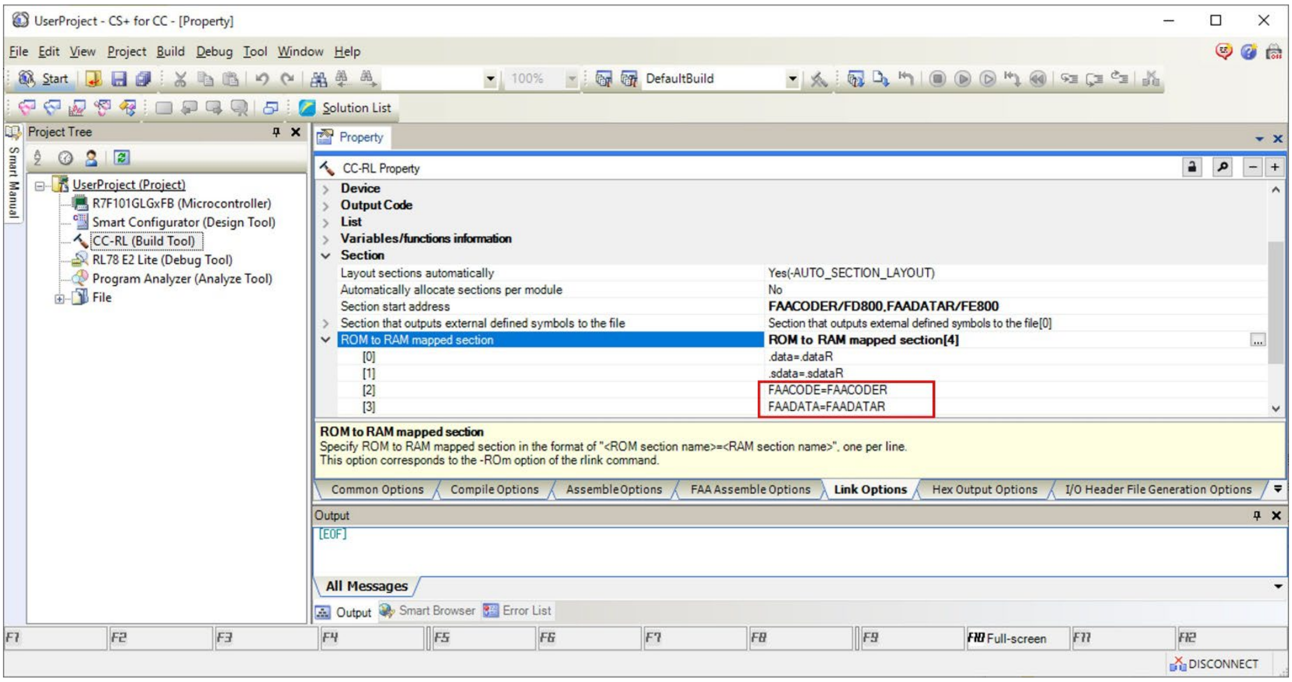



Figure 5-7 Settings for sections to be mapped from ROM to RAM

## 5.2 IAR Environment

### 5.2.1 Integrated Development Environment IAR Embedded Workbench

Download IAR Embedded Workbench from the IAR Website.

[IAR Website]

IAR Embedded Workbench for Renesas RL78 | IAR

Refer to the following IAR C/C++ Development Guide Compiling and Linking for the Renesas RL78 Microcontroller Family.

### 5.2.2 Smart Configurator

Download the Smart Configurator from the URL below.

https://www.renesas.com/rl78-smart-configurator

After activating the installer, install the Smart Configurator by following the procedure of the installer. User will require administrator privileges to do this.

### 5.2.3 LED Control Library

This section describes the procedure for using the LED control library on the Smart Configurator.

For basic operation of the Smart Configurator, refer to the User's Guide for basic Smart Configurator operations.

- RL78 Smart Configurator User's Guide: IAR (R20AN0581)

#### 5.2.3.1 Starting the Smart Configurator

Select [Smart Configurator for RL78 Vx.x.x] of [Renesas Electronics Smart Configurator] from the Windows start menu. The main window of the Smart Configurator will be starting.

Note: Please replace Vx.x.x with user’s version.

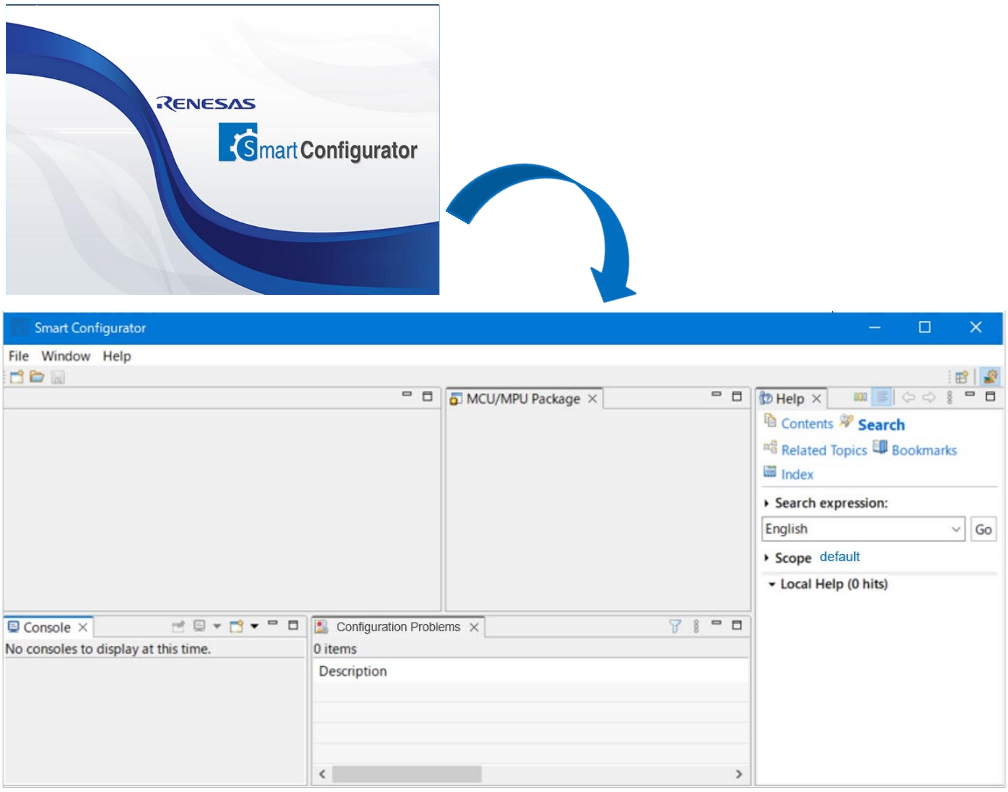



Figure 5-8 Starting of Smart Configurator

#### 5.2.3.2 Creating a New Smart Configurator Configuration File

On the main window, click the  [New Configuration File] button to display the [New Smart Configuration File] dialog box.

(1) In [Platform:], panel, select the device.

(2) In [Toolchain:], select [IAR RL78 Toolchain].

(3) In [File name:], enter the file name.

(4) Confirm [Location:], To change the location, please click [Browse] and select the save destination.

Note: The *.eww, *.ewp, *.ewd, main.c and buildinfo.ipcf files will be generated to this location after clicking “Generate Code” button.

(5) Click [Finish] to create the configuration file.

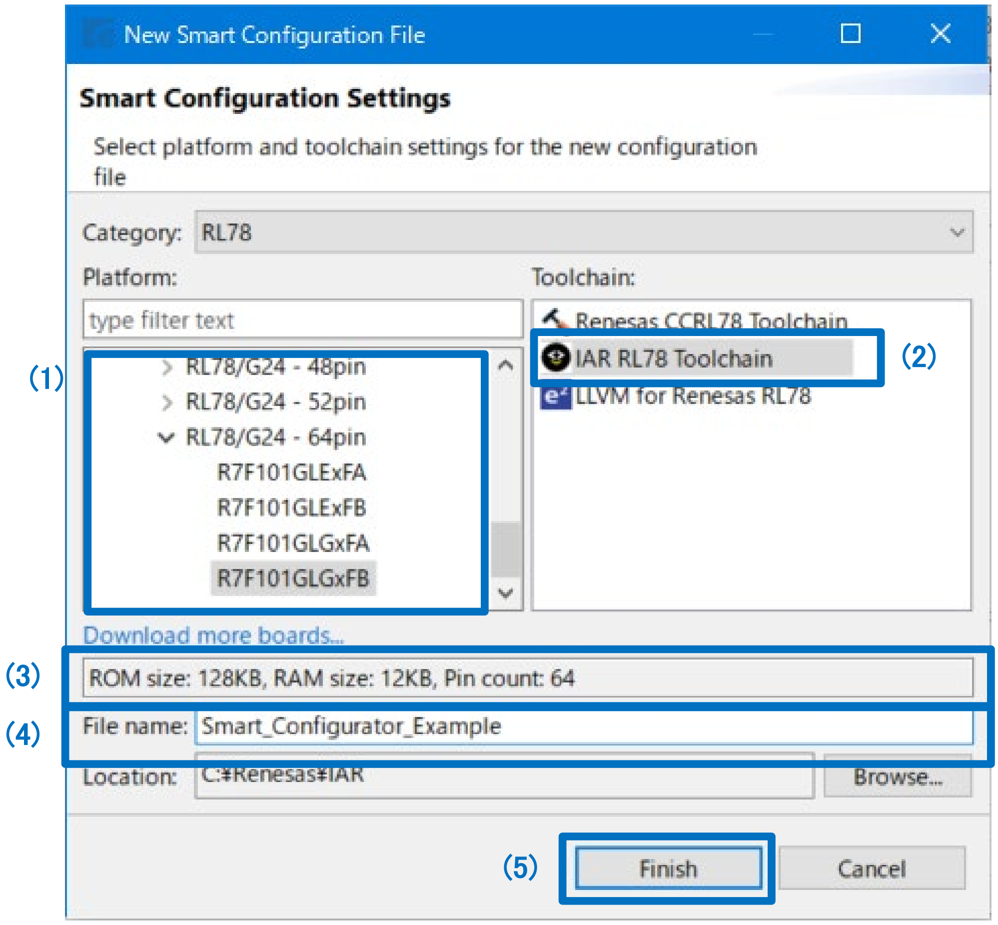



Figure 5-9 Create a Configuration File

(6) Add driver component, configure the setting, generate code, and save the project.

Note: The *.eww, *.ewp, *.ewd and main.c files will be generated only for the first-time code generation, while the buildinfo.ipcf file will be generated always for each time code generation.

#### 5.2.3.3 Clock Setting

For details, please refer to section **5.1.3.2 Clock Settings**.

#### 5.2.3.4 Adding FAA Components

For details, please refer to section **5.1.3.3 Adding FAA Components**.

#### 5.2.3.5 Downloading FAA Modules

For details, please refer to section **5.1.3.4 Downloading FAA Modules**.

#### 5.2.3.6 FAA Module Configuration

For details, please refer to section **5.1.3.5 FAA Module Configuration**.

#### 5.2.3.7 Code Generation

Output a source file for the configured details by clicking on the 

"Generate Code" button in the Smart Configurator view.

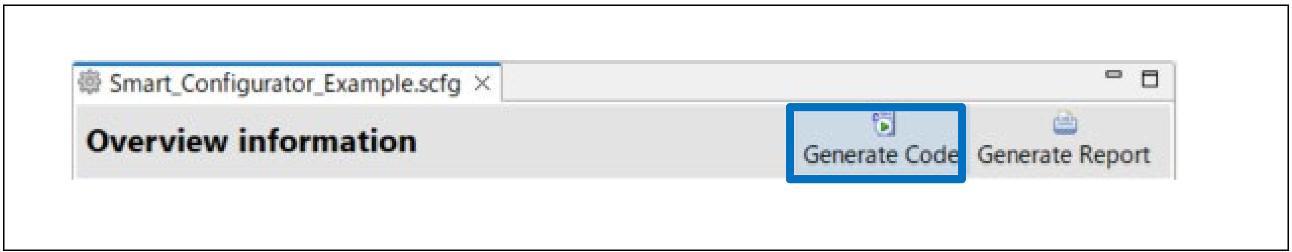



Figure 5-10 Generating a Source File

The Smart Configurator generates a source file in ¥src¥smc_gen and IAR related files in save location (refer to 5.2.3.2Creating a New Smart Configurator Configuration File).

#### 5.2.3.8 Loading in IAR Embedded Workbench

When IAR environment is selected for the compiler to be used, Smart Configurator outputs the related files (.eww/.ewp/.ewd/main.c) together with the source file. It is not necessary for the user to create project files in IAR Embedded Workbench.

The usage procedure is as follows.

(1) Select "Open Workspace…" from the "File" menu of IAR Embedded Workbench.

(2) In the "Open Workspace" dialog box, browse to the folder where the project file is saved, select the project file (.eww), and click the "Open" button.

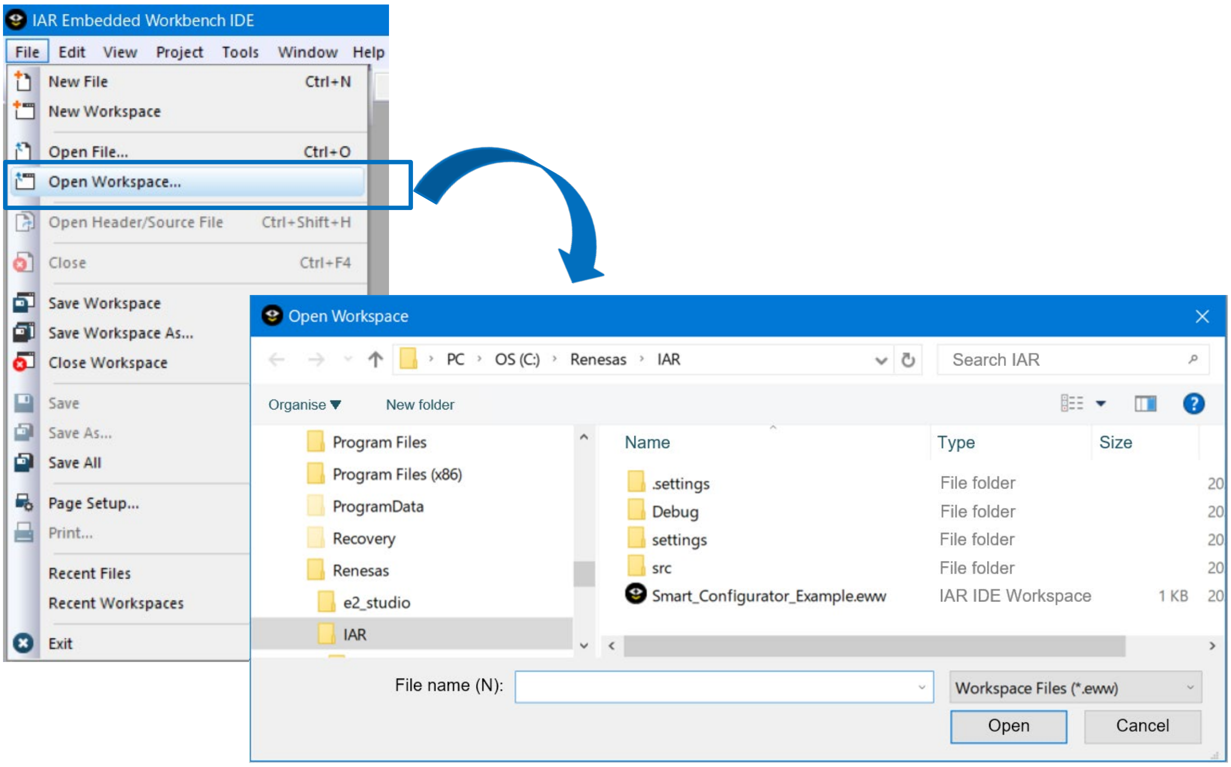



Figure 5-11 Load a *.eww File

(3) The source file output by the Smart Configurator is added to the IAR C project workspace.

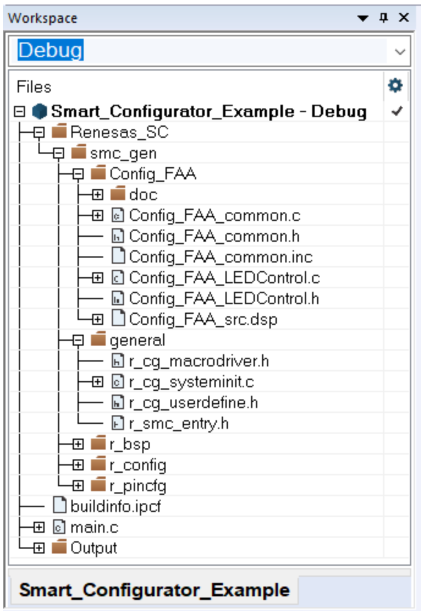



Figure 5-12 New Files Added to IAR Workspace

(4) Select “Options…” from the “Project” menu of IAR Embedded Workbench.

(5) In the “Options for node “ProjectName”” dialog box, change the target device to match with the target device selected when creating Smart Configurator’s configuration file.

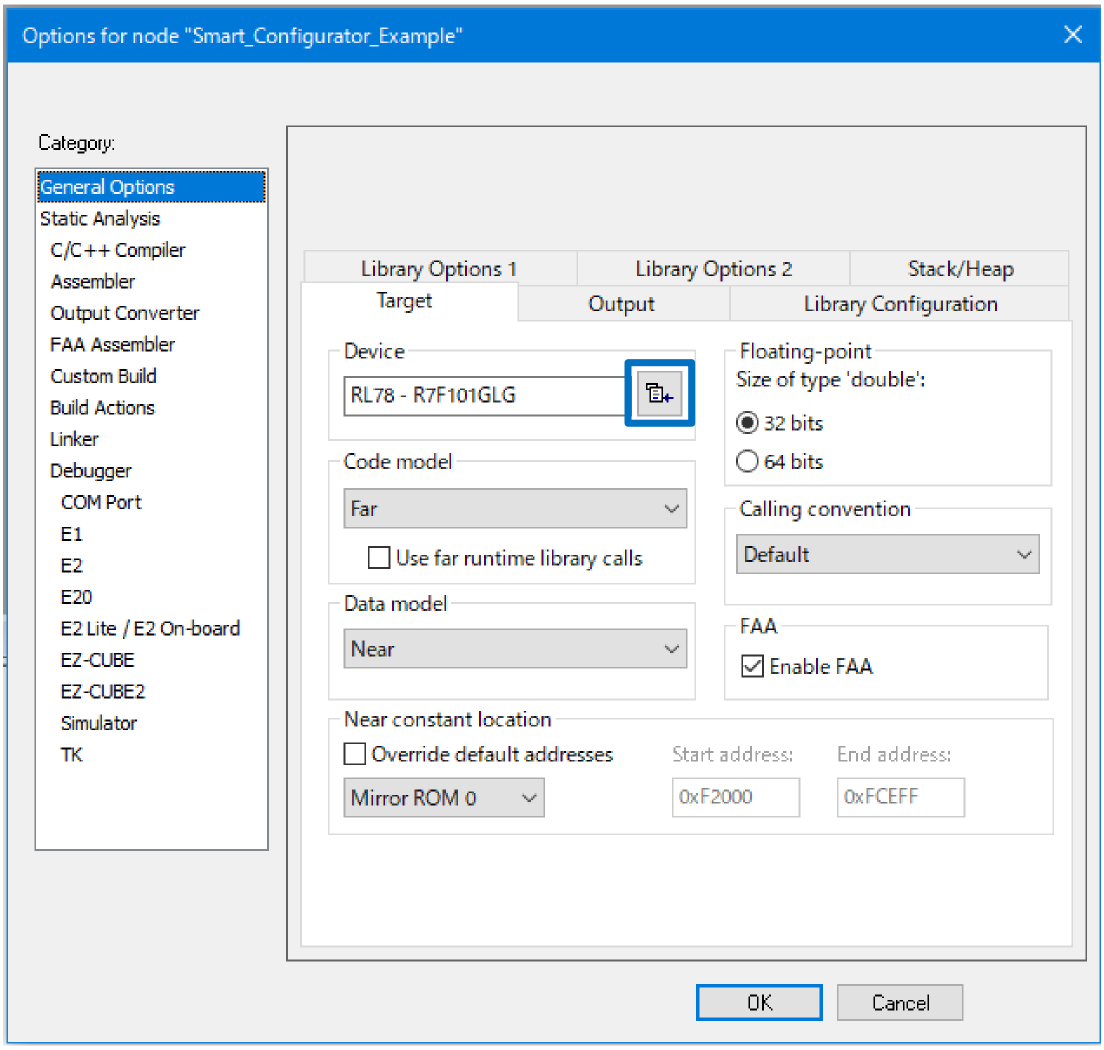



Figure 5-13 New Files Added to IAR Workspace

Implement the API function calls provided by the library in the user application. For details on the various API functions, see **8.2 API Function Specifications**.

# 6. Control Specifications

This chapter describes the LED constant current control of this library.

## 6.1 PI (Proportional-Integral) Control

LED constant current control is achieved by feedback processing based on PI (Proportional-Integral) control.

$$D(n) = D(n-1) + A_1 \cdot E(n) + A_2 \cdot E(n-1)$$

D (n): Latest PWM output duty cycle

D (n-1): Previous PWM output duty

E (n): Latest error value = (A/D conversion target value) - (Latest measured A/D conversion value)

E (n-1): Previous error value = (A/D conversion target value) - (Previous A/D conversion measured value)

A1, A2: Coefficients

The coefficients A1 and A2 are obtained from the following equations

$$A1 = (\pi \times f_Z \times T + 1) \times K_P$$
$$A2 = (\pi \times f_Z \times T - 1) \times K_P$$

π: Pi (pi)
$f_Z$: Zero point frequency
T: Feedback control period
$K_P$: Proportional constant

As shown above, coefficients A1 and A2 are calculated based on the three parameters fZ, T, and KP.

The zero point frequency $f_Z$ and proportionality constant $K_P$ can be set to any value according to the user environment as shown in **Table 7.1 List of Smart Configurator setting items (1/3)**.

The feedback control cycle T is automatically calculated according to the smart configurator settings. For details, see **6.2 Feedback Control Period**.

## 6.2 Feedback Control Period

Feedback control processing is performed at interval FAA timer interrupt timing. If the number of LED channels to be controlled is 2 or more, feedback control processing is performed for one channel per timer interrupt.

Therefore, the feedback control period T is

T = Timer interrupt cycle × number of LED channels

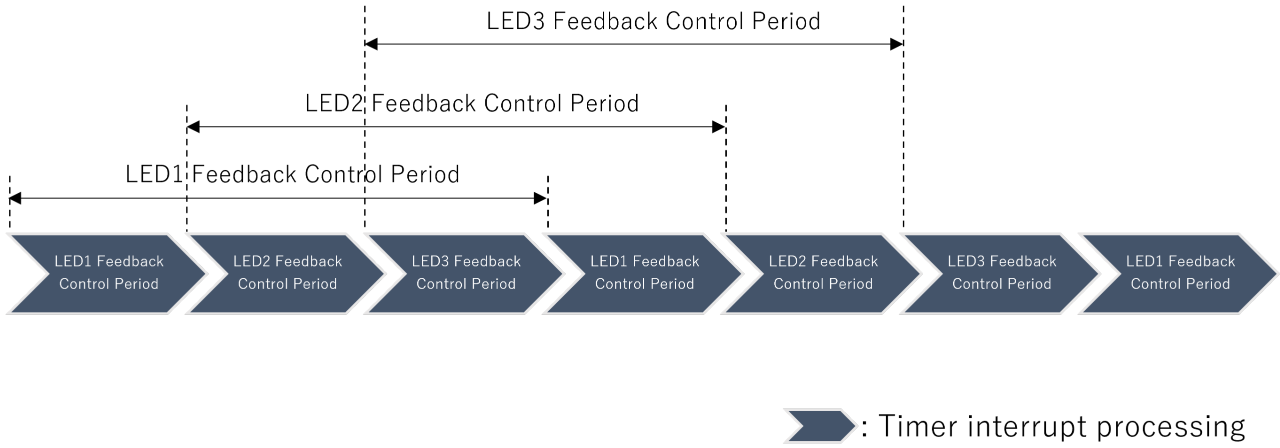



Figure 6-1 Feedback Control Period (e.g., number of LED channels = 3)

The following settings are applied to the timer interrupt cycle for the fastest operation according to the Smart Configurator settings. See **7 Configuration Specifications** for details on smart configurator setting items.

① When Number of LED channels = 1 (Number of LED channels = 1)

When ADC for user application = Unused: 2.5μs

ADC for user application = Used setting: 6μs

② When Number of LED channels = 2 to 4 (Number of LED channels = 2 to 4)

When PGA = Unused: 6.5μs

When PGA = Used, PGA amplification factor = x4/x8: 7μs

When PGA = Used, PGA amplification factor = x16/x32: 12μs

Table 6.1 Timer interrupt cycle

| Number of LEDs | Interrupt cycle | | | | |
|---|---|---|---|---|---|
| | ADC for user application=Unused | ADC for user application=Used | PGA=Unused | PGA=Used | |
| | | | | PGA amplification factor = x4,x8 | PGA amplification factor = x16,x32 |
| 1 | 2.5µs | 6µs | | | |
| 2 - 4 | | | 6.5µs | 7µs | 12µs |

Thus, the timer interrupt cycle is specified such that the minimum value is applied for each condition.

If the interrupt cycle can be changed to any desired value, manually modify the constant value (R_{ConfigName}_LEDControl_TMCP0) in the {ConfigName}_LEDControl.h file.

{ConfigName}" indicates the configuration name of the FAA component set in the Smart Configurator.

**Example**

```
/* Set 100µs @48MHz */
#define R_Config_FAA_LEDControl_TMCP0                    (0x000012C0)
```

## 6.3 Control Flow

The flow of the LED constant current control is shown below. The control process is implemented in the file "{ConfigName}_src.dsp" and executed by the FAA.

### 6.3.1 LED Control Start Processing (Normal Operation)

The API function "R_{ConfigName}_LEDControl_Start" call executes the LED control start process.

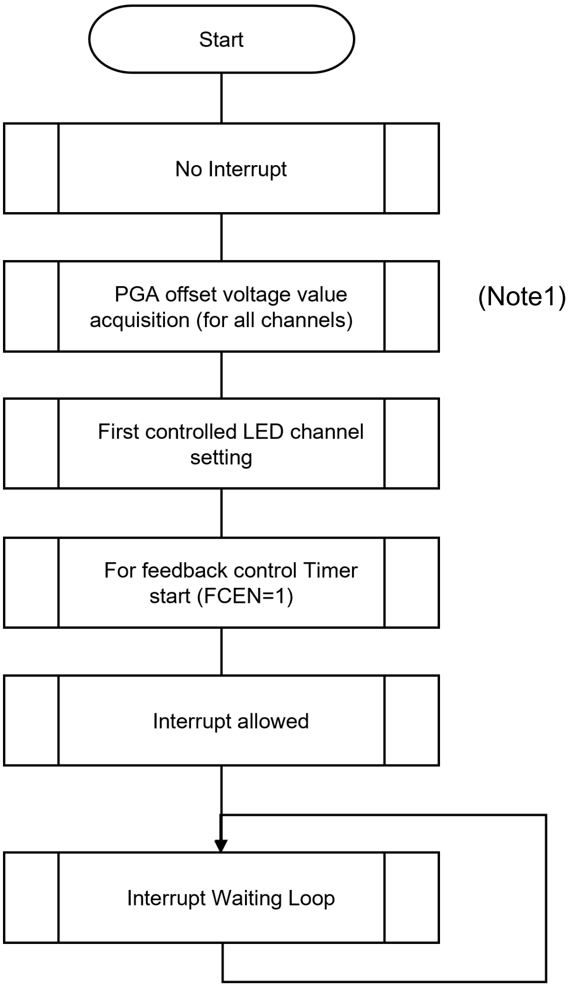



FIgure 6-2 LED Control Start Processing (Normal Operation)

Note1 Disabled when "PGA" = "Unused" is set.

### 6.3.2 LED Control Feedback Processing (Normal Operation)

After the LED control start process, the LED control feedback process is executed by an interval timer interrupt.

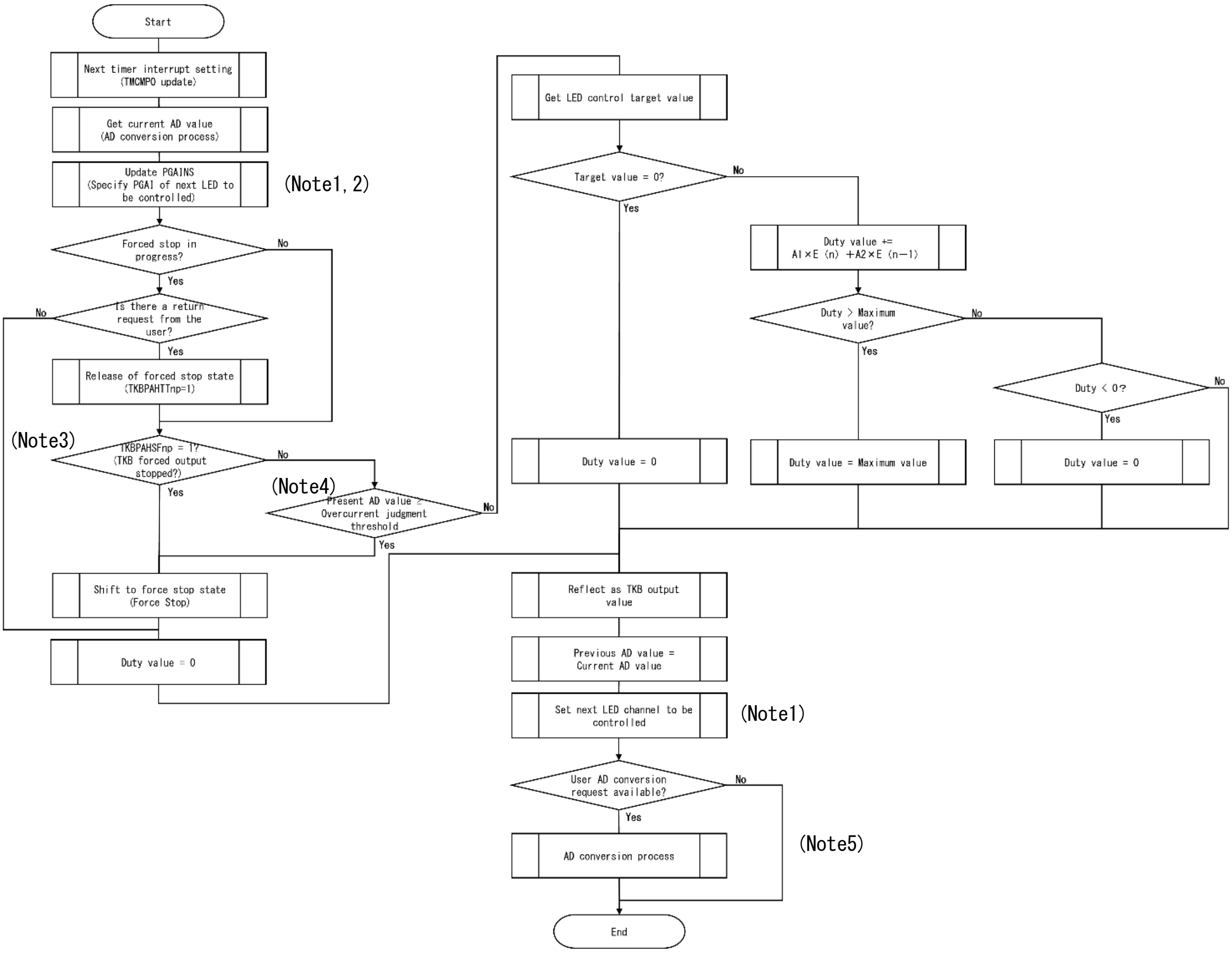



Figure 6-3 LED Control Feedback Processing (Normal Operation)

Note1. Disabled when "Number of LED channels" = "1".

Note2. Disabled when "PGA" = "Unused".

Note3. Disabled when "Overcurrent threshold (Voltage ratio to VDD)" = "0".

Note4. Disabled when "Overcurrent threshold (AD value)" = "0".

Note5. Disabled when "ADC for user application" = "Unused".

### 6.3.3 LED Control Start Processing (High-speed Operation)

If the following settings are specified in the Smart Configurator, faster feedback processing is applied.

Smart Configurator Settings:

"Number of LED channels" = "1" and "ADC for user application" = "Unused”

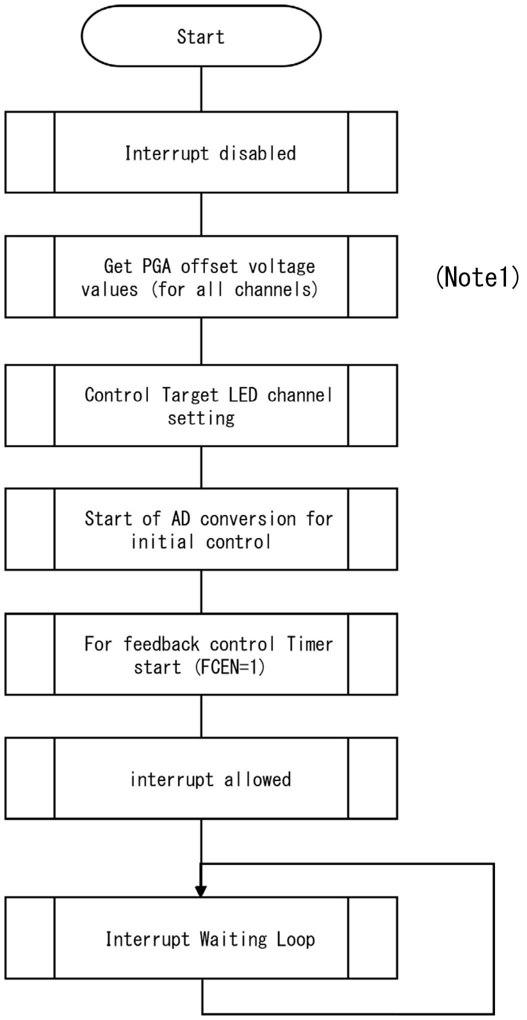



Figure 6-4 LED Control Start Processing (High-speed Operation)

Note1. Disabled when "PGA" = "Unused" is set.

### 6.3.4 LED Control Feedback Processing (High-speed Operation)

When high-speed operation is applied, LED control feedback processing is performed at 2.5-μs intervals.

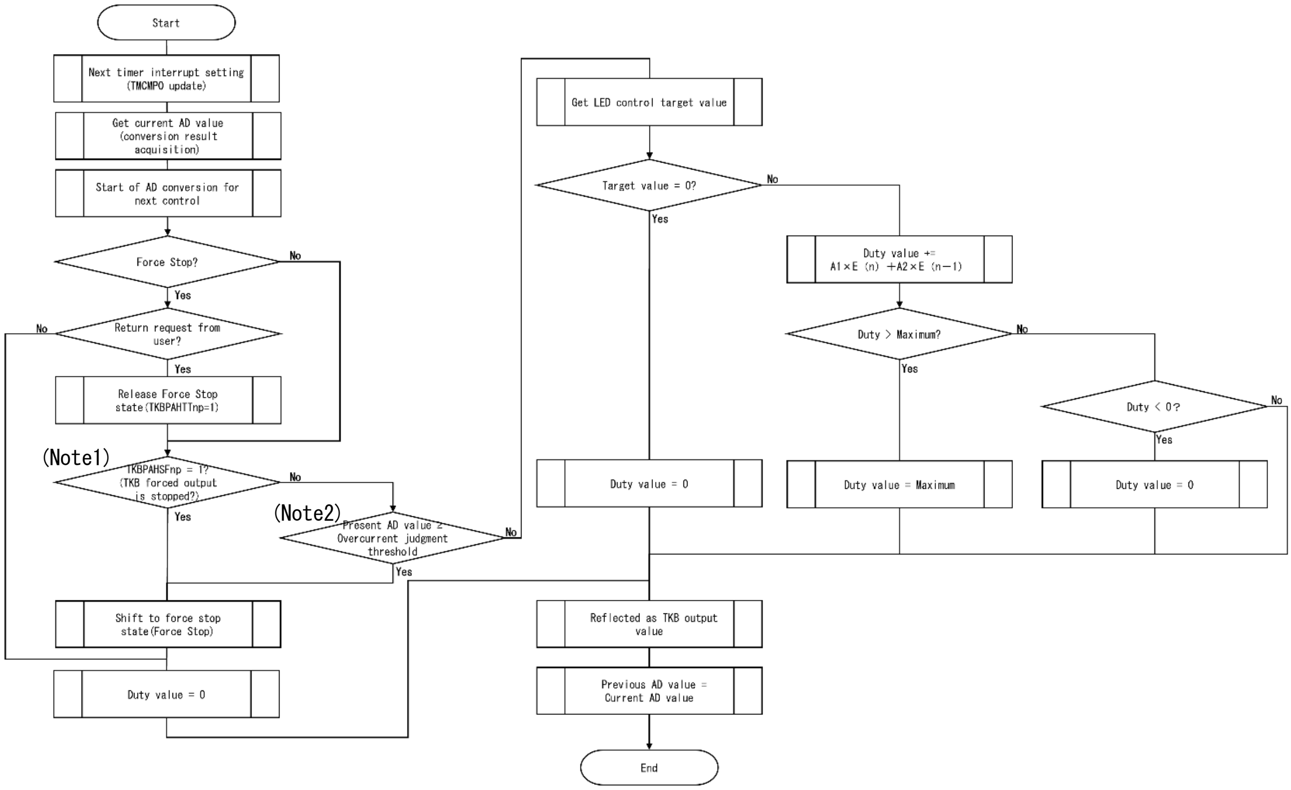



Figure 6-5 LED Control Feedback Processing (High-speed Operation)

Note1. Disabled when "Overcurrent threshold (Voltage ratio to VDD)" = "0".

Note2. Disabled when "Overcurrent threshold (AD value)" = "0".

# 7. Configuration Specifications

A list of configuration items that can be set in the Smart Configurator is shown below.

Table 7.1 List of Smart Configurator setting items (1/3)

| Item | Possible values | Description |
|---|---|---|
| Number of LED channels | 1～4 (Note1) | Select the number of LED channels to be controlled. |
| PGA | Unused/Used | Select whether to use PGA for LED current value detection. |
| PGA amplification factor | x4/x8/x16/x32 | Select the amplification factor of feedback current by PGA. |
| PGA GND | VSS/PGAGND | Select the GND of the feedback resistor of the PGA. |
| ADC VREF(+) | VDD/AVREFP | Select the reference voltage for the plus side of the A/D converter. |
| ADC VREF(-) | VSS/AVREFM | Select the reference voltage for the minus side of the A/D converter. |
| Overcurrent threshold (AD value) | 0～4095 | Select the overcurrent judgment threshold (AD value) for LEDs.<br>When using PGA, set the AD value considering the amplification factor.<br>If set to 0, this feature is disabled. |
| Overcurrent threshold (Voltage ratio to VDD) | 0～0.99999… | Set the LED current value (voltage value conversion value) to trigger forced output stop by TKB as a ratio to the VDD voltage value.<br>e.g.: If VDD = 5V and the overcurrent judgment threshold is 0.6V<br>0.6V / 5V = 0.12<br>If set to 0, this feature is disabled. |
| Proportional gain (Kp) | 0.0001～1000 | Set the proportional gain value in PI feedback control. |
| Zero point frequency (fz) [Hz] | 1～100000 | Set the zero point frequency in PI feedback control. |
| ADC for user application | Unused/Used | Select whether or not to use ADC in user application. |

Note1. For 20-pin products, it is "1 to 3".

Table 7.2 List of Smart Configurator setting items (2/3)

| Item | Possible values | Description |
|---|---|---|
| TKBO pin for LED output (LED chanel1) | TKBO00/TKBO01/TKBO10 /TKOB11/TKBO20/TKBO21 (Note1) | Select the TKBO pin used for output to LED channel 1. |
| TKBO pin for LED output (LED chanel2) | TKBO00/TKBO01/TKBO10 /TKOB11/TKBO20/TKBO21 (Note1) | Select the TKBO pin used for output to LED channel 2. |
| TKBO pin for LED output (LED chanel3) | TKBO00/TKBO01/TKBO10 /TKOB11/TKBO20/TKBO21 (Note1) | Select the TKBO pin used for output to LED channel 3. |
| TKBO pin for LED output (LED chanel4) | TKBO00/TKBO01/TKBO10 /TKOB11/TKBO20/TKBO21 (Note1) | Select the TKBO pin used for output to LED channel 4. |
| Active level of TKBO (LED channel1) | Low level/High level | Select the active level of TKBO for LED channel 1. |
| Active level of TKBO (LED channel2) | Low level/High level | Select the active level of TKBO for LED channel 2. |
| Active level of TKBO (LED channel3) | Low level/High level | Select the active level of TKBO for LED channel 3. |
| Active level of TKBO (LED channel4) | Low level/High level | Select the active level of TKBO for LED channel 4. |
| PGAI pin for feedback input (LED chanel1) | PGAI0/ PGAI1/PGAI2/PGAI3 (Note2) | Select the PGAI pin used for the feedback input of LED channel 1. |
| PGAI pin for feedback input (LED chanel2) | PGAI0/ PGAI1/PGAI2/PGAI3 (Note2) | Select the PGAI pin used for the feedback input of LED channel 2. |
| PGAI pin for feedback input (LED chanel3) | PGAI0/ PGAI1/PGAI2/PGAI3 (Note2) | Select the PGAI pin used for the feedback input of LED channel 3. |
| PGAI pin for feedback input (LED chanel4) | PGAI0/ PGAI1/PGAI2/PGAI3 (Note2) | Select the PGAI pin used for the feedback input of LED channel 4. |
| ANI pin for feedback input (LED chanel1) | ANI18/ANI19/ANI29/ANI30 (Note3) | Select the ANI pin used for the feedback input of LED channel 1. |
| ANI pin for feedback input (LED chanel2) | ANI18/ANI19/ANI29/ANI30 (Note3) | Select the ANI pin used for the feedback input of LED channel 2. |
| ANI pin for feedback input (LED chanel3) | ANI18/ANI19/ANI29/ANI30 (Note3) | Select the ANI pin used for the feedback input of LED channel 3. |
| ANI pin for feedback input (LED chanel4) | ANI18/ANI19/ANI29/ANI30 (Note3) | Select the ANI pin used for the feedback input of LED channel 4. |

Note1. "TKBO20" and "TKBO21" are not supported for 20-pin products.

Note2. "PGAI3" is not supported for 20-pin products.

Note3. "ANI18" is not supported for 20-pin products.

Table 7.3 List of Smart Configurator setting items (3/3)

| Item | Possible values | Description |
|---|---|---|
| ANI0 | Unused/Used | Select whether to use analog input (ANI) in the user application. |
| ANI1 | Unused/Used | |
| ANI2 | Unused/Used | |
| ANI3 | Unused/Used | |
| ANI4 | Unused/Used (Note1,2,3) | |
| ANI5 | Unused/Used (Note1,2,3) | |
| ANI6 | Unused/Used (Note1,2,3) | |
| ANI7 | Unused/Used (Note1,2,3,4) | |
| ANI16 | Unused/Used (Note1,2,3,4,5) | |
| ANI17 | Unused/Used (Note1,2,3,4,5) | |
| ANI18 | Unused/Used (Note1) | |
| ANI19 | Unused/Used | |
| ANI20 | Unused/Used | |
| ANI21 | Unused/Used | |
| ANI22 | Unused/Used | |
| ANI23 | Unused/Used | |
| ANI24 | Unused/Used (Note2) | |
| ANI25 | Unused/Used (Note1,2) | |
| ANI26 | Unused/Used (Note1) | |
| ANI27 | Unused/Used (Note1,2) | |
| ANI28 | Unused/Used (Note1,2,3,4) | |
| ANI29 | Unused/Used | |
| ANI30 | Unused/Used | |

Note1. "Used" cannot be selected for 20-pin products because they are not supported.

Note2. "Used" cannot be selected for 24-pin products because they are not supported.

Note3. "Used" cannot be selected for 25-32 pin products because they are not supported.

Note4. “Used" cannot be selected for 40-pin products because they are not supported.

Note5. “Used" cannot be selected for 44-48 pin products because they are not supported.

# 8. API Information

## 8.1 API Typedef Definitions

This section describes the Typedef definitions provided by this library.

### 8.1.1 e_faa_ad_channel_t

This typedef defines the analog input channel. It is used to specify the channel to be converted in the ADC functions provided for user applications.

The target analog channel definitions are generated according to the settings in Table 7.3 List of Smart Configurator setting items (3/3).

```
typedef enum
{
    ADCHANNEL0 = 0,
    ADCHANNEL1 = 1,
    ADCHANNEL2 = 2,
    ADCHANNEL2 = 3,
    . . .

} e_faa_ad_channel_t;
```

### 8.1.2 e_faa_result_adc_t

This typedef defines the conversion result in the ADC function for the user application.

```
typedef enum
{
    FAA_ADC_NOTCOMPLETED,
    FAA_ADC_COMPLETED,
    FAA_ADC_FAILED
} e_faa_result_adc_t;
```

## 8.2 API Function Specifications

This section describes the API functions provided by this library.

The "{ConfigName}" in the API function name indicates the configuration name of the FAA component set by the smart configurator. "{n}" indicates the LED channel number.

### 8.2.1 R_{ConfigName}_LEDControl_Create

This function performs the initialization process of peripheral features required for LED control.

**Format**

```
void R_{ConfigName}_LEDControl_Create (void)
```

**Parameters**

None

**Return Values**

None

**Properties**

Prototype declared in {ConfigName}_LEDControl.h.

**Description**

This function initializes the following peripheral functions used in LED control.

- Programmable gain amplifier
- A/D converter
- D/A converter
- Comparator
- 16-bit timer KB

**Example**

```
/* Set FAA settings */
R_Config_FAA_Create();
R_Config_FAA_LEDControl_Create();
```

**Special Notes:**

Since this function call processing is included in the source code generated by the smart configurator, function call processing from the user program is not required.

### 8.2.2 R_{ConfigName}_LEDControl_Start

This function initiates the peripheral functions and FAA processing required by the LED control.

**Format**

```
FAA_Status_t R_{ConfigName}_LEDControl_Start (void)
```

**Parameters**

None

**Return Values**

| | |
|---|---|
| FAA_SUCCESS | FAA processing success |
| FAA_ALREADY_RUNNING | FAA in progress |

**Properties**

Prototype declared in {ConfigName}_LEDControl.h.

**Description**

This function starts the following peripheral functions used in LED control and FAA processing.

- Programmable gain amplifier
- A/D converter
- D/A converter
- Comparator
- 16-bit timer KB

**Example**

```
/* Start LED Control */
FAA_Status_t status;
status = R_Config_FAA_Start();

if (status != FAA_SUCCESS)
{
  . . .
}
```

**Special Notes:**

Be sure to call R_{ConfigName}_LEDControl_Create() before calling this function.

### 8.2.3 R_{ConfigName}_LEDControl_Stop

This function stops peripheral features and FAA processing required by the LED control.

**Format**

```
void R_{ConfigName}_LEDControl_Stop (void)
```

**Parameters**

None

**Return Values**

None

**Properties**

Prototype declared in {ConfigName}_LEDControl.h.

**Description**

This function stops the following peripheral functions used in LED control and FAA processing.

- Programmable gain amplifier
- A/D converter
- D/A converter
- Comparator
- 16-bit timer KB

**Example**

```
/* Stop LED Control */
R_Config_FAA_Stop();
```

### 8.2.4 R_{ConfigName}_LEDControl_SetTargetLevelCh{n}

This function sets the target value in LED constant current control.

**Format**

```
void R_{ConfigName}_LEDControl_ SetTargetLevelCh{n} (uint16_t target_level)
```

**Parameters**

target_level

LED constant current control target value (valid range: 0 - 4095)

**Return Values**

None

**Properties**

Prototype declared in {ConfigName}_LEDControl.h

**Description**

This function sets the target value (12bit) for the LED constant current control.

**Example**

```
/* Set LED target level */
uint16_t level = 150;
R_Config_FAA_LEDControl_SetTargetLevelCh1(level);
```

### 8.2.5 R_{ConfigName}_LEDControl_GetCurrentLevelCh{n}

This function gets the current value in the LED constant current control.

**Format**

```
uint16_t R_{ConfigName}_LEDControl_GetCurrentLevelCh{n} (void)
```

**Parameters**

None

**Return Values**

LED constant current control Current value (valid range: 0 - 4095)

**Properties**

Prototype declared in {ConfigName}_LEDControl.h.

**Description**

This function gets the current value (12 bits) in the LED constant current control.

**Example**

```
/* Get LED current level */
uint16_t level;
level = R_Config_FAA_LEDControl_GetCurrentLevelCh1();

if (level >= 20)
{
  . . .
}
```

### 8.2.6 R_{ConfigName}_LEDControl_IsForceStopCh{n}

This function gets whether or not an LED forced stop has occurred.

**Format**

```
bool R_{ConfigName}_LEDControl_IsForceStopCh{n} (void)
```

**Parameters**

None

**Return Values**

Forced stop status (true : occurs / false : does not occur)

**Properties**

Prototype declared in {ConfigName}_LEDControl.h.

**Description**

This function gets the occurrence of forced stop by LED overcurrent detection.

**Example**

```
/* Get LED force stop status */
bool is_force_stop;
is_force_stop = R_Config_FAA_LEDControl_IsForceStopCh1();

if (is_force_stop == true)
{
  . . .
}
```

### 8.2.7 R_{ConfigName}_LEDControl_ClearForceStopCh{n}

This function clears the LED forced stop status.

**Format**

```
void R_{ConfigName}_LEDControl_ClearForceStopCh{n} (void)
```

**Parameters**

None

**Return Values**

None

**Properties**

Prototype declared in {ConfigName}_LEDControl.h.

**Description**

This function clears the forced stop status caused by LED overcurrent detection.

**Example**

```
/* Get LED current level */
uint16_t level;
level = R_Config_FAA_LEDControl_GetCurrentLevelCh1();

if (level <= 15)
{
  R_Config_FAA_LEDControl_ClearForceStopCh1();
}
```

### 8.2.8 R_{ConfigName}_LEDControl_RequestADC

This function requests an AD conversion for the specified analog channel.

**Format**

```
void R_{ConfigName}_LEDControl_RequestADC (e_faa_ad_channel_t channel)
```

**Parameters**

channel

Analog channel to be converted

**Return Values**

None

**Properties**

Prototype declared in {ConfigName}_LEDControl.h.

**Description**

This function requests the FAA function to perform an AD conversion on the specified analog channel.

**Example**

```
/* Request AD conversion */
R_Config_FAA_LEDControl_RequestADC(ADCHANNEL2);
```

**Special Notes:**

If "ADC for user application" = "Unused" in the smart configurator, this function is not provided.

### 8.2.9 R_{ConfigName}_LEDControl_GetAD

This function gets the result of an AD conversion.

**Format**

```
e_faa_result_adc_t R_{ConfigName}_LEDControl_GetAD (uint16_t * const buffer)
```

**Parameters**

buffer

Pointer to the buffer for storing the result of AD conversion

* Valid only when the return value of this function is FAA_ADC_COMPLETED (AD conversion complete)

**Return Values**

| | |
|---|---|
| FAA_ADC_COMPLETED | AD conversion completed |
| FAA_ADC_FAILED | AD conversion failed |
| FAA_ADC_NOTCOMPLETED | AD conversion not completed |

**Properties**

Prototype declared in {ConfigName}_LEDControl.h.

**Description**

This function gets the result of the AD conversion requested by R_{ConfigName}_LEDControl_RequestADC().

**Example**

```
uint16_t buff;
e_faa_result_adc_t adc_result;

/* Request AD conversion */
R_Config_FAA_LEDControl_RequestADC(ADCHANNEL2);
/* Wait for completion of AD conversion by FAA */
do
{
  adc_result = R_Config_FAA_LEDControl_GetAD(&buff);
}
while (adc_result == FAA_ADC_NOTCOMPLETED);
```

**Special Notes:**

If "ADC for user application" = "Unused" in the smart configurator, this function is not provided.

# 9. Website and Support

Renesas Electronics Website

http://www.renesas.com/

Contact information

http://www.renesas.com/contact/

# Revision History

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| 1.00 | Sep. 1st, 2023 | — | First edition |
| 1.01 | Jan. 23th,2026 | — | Add LED control library for IAR EW for Renesas RL78 |

# 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

## 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/.

## Trademarks

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