

RL78/G24

Interleaved CrM PFC and LLC Control with RL78/G24 (FAA Software)

Introduction

This application note describes the digital power converter control library using the Flexible Application Accelerator (FAA).

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

Target Device

RL78/G24

Board

RL78/G24 Interleaved PFC + LLC Board 400W Kit (RTK0EL0006D00000BJ)

Note:

When applying this sample application to a different microcontroller, please modify it according to the specifications of that microcontroller and perform sufficient evaluation.

Related Documents

- Interleaved CrM PFC and LLC Control Using RL78/G24(Hardware & Software Basics) (RA01AN8175)
- Interleaved CrM PFC and LLC Control Using RL78/G24(CPU Software) (RA01AN8177)
- RL78/G24 User's Manual: Hardware (R01UH0961)
- CS+ Integrated Development Environment User's Manual: Project Operation (R20UT4691)
- e2 studio Integrated Development Environment User's Manual: Quick Start Guide (R20UT5293)
- RL78 Smart Configurator User's Guide: CS+ (R20AN0580)
- RL78 Smart Configurator User's Guide: IAREW (R20AN0581)
- RL78 Smart Configurator User's Guide: e2 studio (R20AN0579)

Contents

1.	Digital Power Converter Control Library Functions	6
1.1	PFC + LLC Converter Control	6
1.2	Flexible Application Accelerator (FAA).....	6
2.	Tools that can use the digital power converter control library	7
3.	Using Resources	8
3.1	Peripheral Functions	8
3.2	Peripheral Functions Restrictions.....	8
3.3	ROM / RAM Size	9
4.	Digital Power Converter Control Library Usage Procedure	10
4.1	Environment Setup.....	10
4.1.1	CS+ for CC.....	10
4.1.2	IAR Embedded Workbench for Renesas RL78.....	10
4.1.3	e2 studio.....	10
4.2	Smart Configurator	11
4.3	Digital Power Converter Control Library.....	11
4.3.1	Starting the Smart Configurator	11
4.3.1.1	CS+	11
4.3.1.2	IAR.....	11
4.3.1.3	e2 studio.....	11
4.3.2	Clock Settings	12
4.3.3	Adding FAA Components.....	13
4.3.4	Downloading FAA Modules.....	14
4.3.5	FAA Module Configuration	15
4.3.6	Code Generation	15
5.	Library Structure	16
5.1	Architecture	16
5.2	Folder Structure.....	17
6.	Library Control Specifications	18
6.1	Clock Operation Modes.....	18
6.1.1	Low-Speed Mode	18
6.1.2	High-Speed Mode	18
6.2	PFC Converter Control.....	19
6.2.1	Fixed On-Width Output Control.....	19
6.2.2	Feedback Control	21
6.2.2.1	Output Voltage Monitoring	21
6.2.2.2	PI (Proportional-Integral) Control	22
6.2.3	Switching Between Single and Interleaved Modes	23
6.2.3.1	Fade Processing	23
6.2.3.2	Output Energy Adjustment.....	24
6.2.4	Maximum Frequency Limit Control	25

6.2.5	Protection Functions.....	25
6.2.5.1	D-OVP (Dynamic Overvoltage Protection).....	25
6.2.5.2	OVP (Overvoltage Protection).....	25
6.2.5.3	OCP (Overcurrent Protection).....	25
6.3	LLC Converter Control	26
6.3.1	Feedback Control	26
6.3.1.1	Output Voltage Monitoring	26
6.3.1.2	Soft Start Control.....	28
6.3.1.3	PI (Proportional-Integral) Control	29
6.3.2	Protection Functions.....	30
6.3.2.1	D-OVP (Dynamic Overvoltage Protection).....	30
6.3.2.2	OCP (Overcurrent Protection).....	30
6.4	A/D Conversion Control for User Applications	31
6.4.1	A/D Conversion Control in Low-Speed Mode	31
6.4.2	A/D Conversion Control in High-Speed Mode.....	31
6.5	Data Read Function for User Applications	31
6.6	Control Flow	32
6.6.1	Feedback Control Sequence.....	32
6.6.2	Task Processing Flow (FAA Program).....	33
6.6.3	A/D Conversion Complete Interrupt Processing Flow (FAA Program)	34
7.	Configuration Specifications	35
7.1	Basic Settings Items	35
7.2	PFC Settings	36
7.3	LLC Settings	38
7.4	User Application Settings	40
8.	API Information.....	41
8.1	API Typedef Definitions.....	41
8.1.1	e_faa_peripheral_t	41
8.1.2	e_faa_clock_mode_t.....	41
8.1.3	e_faa_pfc_status_t.....	41
8.1.4	e_faa_pfc_output_t.....	42
8.1.5	e_faa_llc_status_t	42
8.1.6	e_faa_ad_channel_t.....	42
8.1.7	e_faa_result_adc_t.....	43
8.1.8	e_faa_request_id_t.....	43
8.1.9	faa_pfc_callback_t.....	43
8.1.10	faa_llc_callback_t.....	44
8.2	API Function Specifications.....	45
8.2.1	R_{ConfigName}_DPConv_Create	45
8.2.2	R_{ConfigName}_DPConv_PowerOn.....	46
8.2.3	R_{ConfigName}_DPConv_PowerOff.....	47
8.2.4	R_{ConfigName}_DPConv_ChangeClockMode	48
8.2.5	R_{ConfigName}_DPConv_PFC_Init.....	49
8.2.6	R_{ConfigName}_DPConv_PFC_StartFixedOnWidth	50
8.2.7	R_{ConfigName}_DPConv_PFC_Suspend	51

8.2.8	R_{ConfigName}_DPConv_PFC_Resume	52
8.2.9	R_{ConfigName}_DPConv_PFC_StartFeedbackControl	53
8.2.10	R_{ConfigName}_DPConv_PFC_Stop	54
8.2.11	R_{ConfigName}_DPConv_PFC_TaskForFeedbackControl	55
8.2.12	R_{ConfigName}_DPConv_PFC_GetStatus	56
8.2.13	R_{ConfigName}_DPConv_PFC_ChangeOutputMode	57
8.2.14	R_{ConfigName}_DPConv_PFC_SetMaxFreqLimit	58
8.2.15	R_{ConfigName}_DPConv_PFC_GetMaxFreqLimit	59
8.2.16	R_{ConfigName}_DPConv_PFC_SetFz	60
8.2.17	R_{ConfigName}_DPConv_PFC_SetKp	61
8.2.18	R_{ConfigName}_DPConv_PFC_SetAdjustParamForInterleaved	62
8.2.19	R_{ConfigName}_DPConv_PFC_GetOnWidthCount	63
8.2.20	R_{ConfigName}_DPConv_LLCn_Init	64
8.2.21	R_{ConfigName}_DPConv_LLCn_StartFeedbackControl	65
8.2.22	R_{ConfigName}_DPConv_LLCn_Stop	66
8.2.23	R_{ConfigName}_DPConv_LLCn_EnableOCP	67
8.2.24	R_{ConfigName}_DPConv_LLCn_DisableOCP	68
8.2.25	R_{ConfigName}_DPConv_LLCn_GetStatus	69
8.2.26	R_{ConfigName}_DPConv_GetAdValue	70
8.2.27	R_{ConfigName}_DPConv_RequestAdValue	71
8.2.28	R_{ConfigName}_DPConv_GetRequestedAdValue	72
8.2.29	R_{ConfigName}_DPConv_RequestData	73
8.2.30	R_{ConfigName}_DPConv_GetRequestedData	74
9.	Sample Program	75
9.1	Hardware Configuration	75
9.2	Operational Confirmation Conditions	75
9.3	Software Specifications	76
9.3.1	Software Control	76
9.3.2	Software Configuration	76
9.3.3	Folder Structure	77
9.3.4	Option Byte Settings	78
9.3.5	Configuration Settings	78
9.3.6	How to build the environment	84
9.3.6.1	CC-RL compiler (CS+ for CC)	84
9.3.6.2	CC-RL compiler (e2 studio)	85
9.3.6.3	IAR compiler	86
9.3.7	API Functions	87
10.	Operation using “QE for Lighting & Power”	88
10.1	Downloading QE for Lighting & Power	88
10.2	Preparing a Workspace	88
10.2.1	Selecting a Board	88
10.2.2	Selecting a Workspace	88
10.3	Configure Lighting Communication	89
10.4	Configure Power Control	90
10.4.1	Configure Power Control Mode	91

10.4.1.1 Power Configuration View	91
10.4.2 Simulate Power Control.....	97
10.4.2.1 Power Simulation Dialog	98
10.4.3 Evaluate Power Control.....	102
10.4.3.1 Build Settings	103
10.4.3.2 Prepare Evaluation Program.....	103
10.4.3.3 Write to Board	103
10.4.3.4 Connect to Board	104
10.4.3.5 Evaluate Power Control.....	104
10.5 Generate and Program.....	108
10.5.1 Generate Source Code	108
10.5.2 Download to Board.....	108
10.6 Notes on PFC Control Parameters A1 and A2.....	109
10.6.1 Calculation Method of Kp and fz	109
10.6.2 How to Reflect Kp and fz.....	109
11. Notes	110
12. Reference Documents.....	110

1. Digital Power Converter Control Library Functions

This chapter describes the functions of the digital power converter control library.

1.1 PFC + LLC Converter Control

This library implements PFC (Power Factor Correction) converter control and LLC converter control using the timer KB PWM output function of the RL78/G24 microcontroller. Constant voltage control for each converter is achieved through feedback processing based on PI (Proportional-Integral) control. This eliminates the need for dedicated external control ICs, reducing design costs.

1.2 Flexible Application Accelerator (FAA)

This library utilizes the Flexible Application Accelerator (FAA) to implement digital power converter control. The FAA is integrated into the RL78/G24 microcontroller and operates as a processor independent of the CPU. This enables high-speed, efficient voltage control processing without occupying CPU resources.

For details of FAA, refer to RL78/G24 User's Manual: Hardware (R01UH0961).

2. Tools that can use the digital power converter 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 shows a list of tools for using this library. For details on the usage procedure, refer to 4 Digital Power Converter Control Library Usage Procedure.

Table 2.1 Tools that can use the digital power converter control library

Environment	Item	Description
CS+ for CC	Integrated Development Environment	Renesas Electronics CS+ for CC V8.14.00 or later
	Compiler	CC-RL V1.15.01 or later
IAR	Integrated Development Environment	IAR Systems IAR Embedded Workbench for Renesas RL78 V5.10.3 or later
	Compiler	IAR C/C++ Compiler for Renesas RL78 V5.10.3.2716 or later
e2 studio	Integrated Development Environment	Renesas Electronics e2 studio 2025-10 or later
	Compiler	CC-RL V1.15.01 or later
common	Configurator (SC)	Renesas Smart Configurator for RL78 V1.15.0 or later

3. Using Resources

3.1 Peripheral Functions

This library uses the following peripheral functions.

- Flexible application accelerator
- A/D converter
- D/A converter
- Comparator
- Timer Array Unit
- 16-bit timer KB
- Event Link Controller

3.2 Peripheral Functions Restrictions

When using this library, please note the following points:

- ELCAD3 events in the Event Link Controller are used by this library.
Do not change the setting value of Event Output Destination Selection Register 33 (ELSER33 register).
- Timer Array Unit 0 Channel 1 is used by this library.
Do not change the CK00 clock division ratio, as CK00 is used as the operating clock.
(Do not change the setting value of bits 3-0 in the TPS0 register.)
- When using an A/D converter, use the A/D conversion control functions provided by this library.
For details, see Section 6.4 A/D Conversion Control for User Applications.

3.3 ROM / RAM Size

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

Compiler

CC-RL

Compiler Options

-cpu=S3 -memory_model=medium -Odefault

Linking Options

-NOOptimize

Table 3.1 ROM / RAM Size

Item	ROM	RAM	FAACODE	FAADATA
PFC	5,309 [byte]	12 [byte]	1,432 [byte]	556 [byte]
LLC1	3,630 [byte]	6 [byte]	1,080 [byte]	440 [byte]
LLC1 + LLC2	4,775 [byte]	11 [byte]	1,628 [byte]	616 [byte]
PFC + LLC1	5,856 [byte]	17 [byte]	2,016 [byte]	768 [byte]
PFC + LLC1 + LLC2	6,997 [byte]	22 [byte]	2,564 [byte]	944 [byte]

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

The ROM size includes the FAACODER section and the FAADATAR section.

4. Digital Power Converter Control Library Usage Procedure

The procedure for using this library is shown below.

4.1 Environment Setup

This section describes how to set up the development environment for each IDE.

4.1.1 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)

4.1.2 IAR Embedded Workbench for Renesas RL78

For installation instructions, refer to the video below.

[Using Smart Configurator with IAR Embedded Workbench for RL78 \(1/2\) - Installation | Renesas](#)

4.1.3 e2 studio

Download e2 studio 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 e2 studio operations.

e2 studio Integrated Development Environment User's Manual: Quick Start Guide (R20UT5293)

4.2 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.

4.3 Digital Power Converter Control Library

This section describes the procedure for using the Digital Power Converter Control Library on the Smart Configurator.

For basic operation of the Smart Configurator, refer to the following user guides:

- RL78 Smart Configurator User's Guide: CS+ (R20AN0580)
- RL78 Smart Configurator User's Guide: IAREW (R20AN0581)
- RL78 Smart Configurator User's Guide: e2 studio (R20AN0579)

4.3.1 Starting the Smart Configurator

4.3.1.1 CS+

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

4.3.1.2 IAR

Start Smart Configurator and create a new project or open an existing project's .scfg file.

4.3.1.3 e2 studio

Create a new project or import an existing project, then double-click the .scfg file within the e2 studio project workspace to launch it.

4.3.2 Clock Settings

Select the "Clocks" page of the "Smart Configurator View" and set the "Timer clock" and "f_{CLK}" outputs as follows. If settings other than this clock configuration are applied, peripheral functions configured by the user may cease to function properly.

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

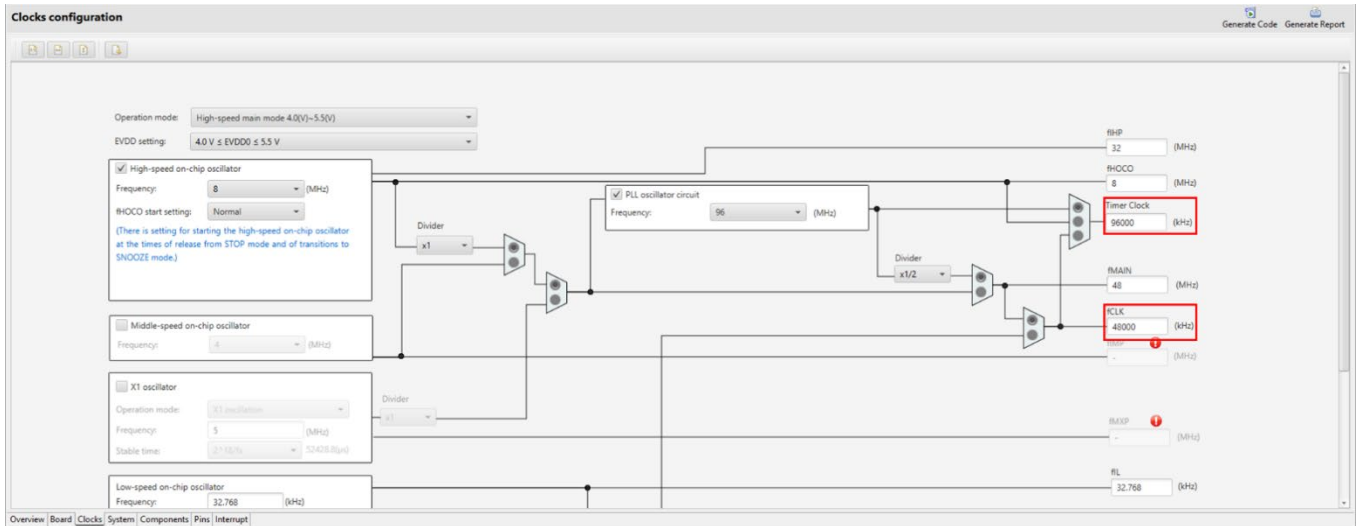


Figure 4.1 Clock Settings

4.3.3 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.

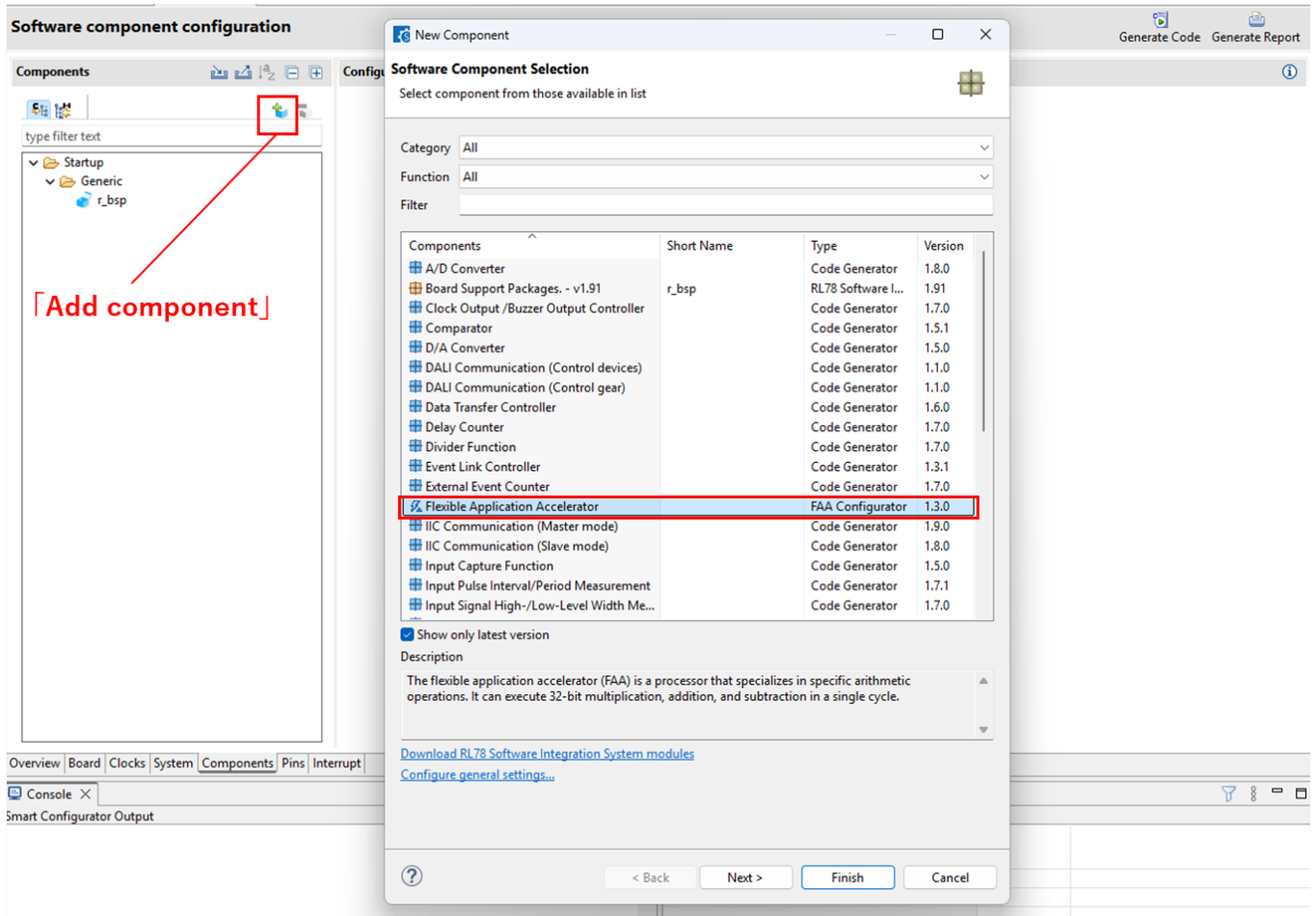



Figure 4.2 Adding FAA Components

4.3.4 Downloading FAA Modules

Click the “” icon displayed on the screen to view the downloadable FAA modules. Select “Digital Power Converter” and download it.

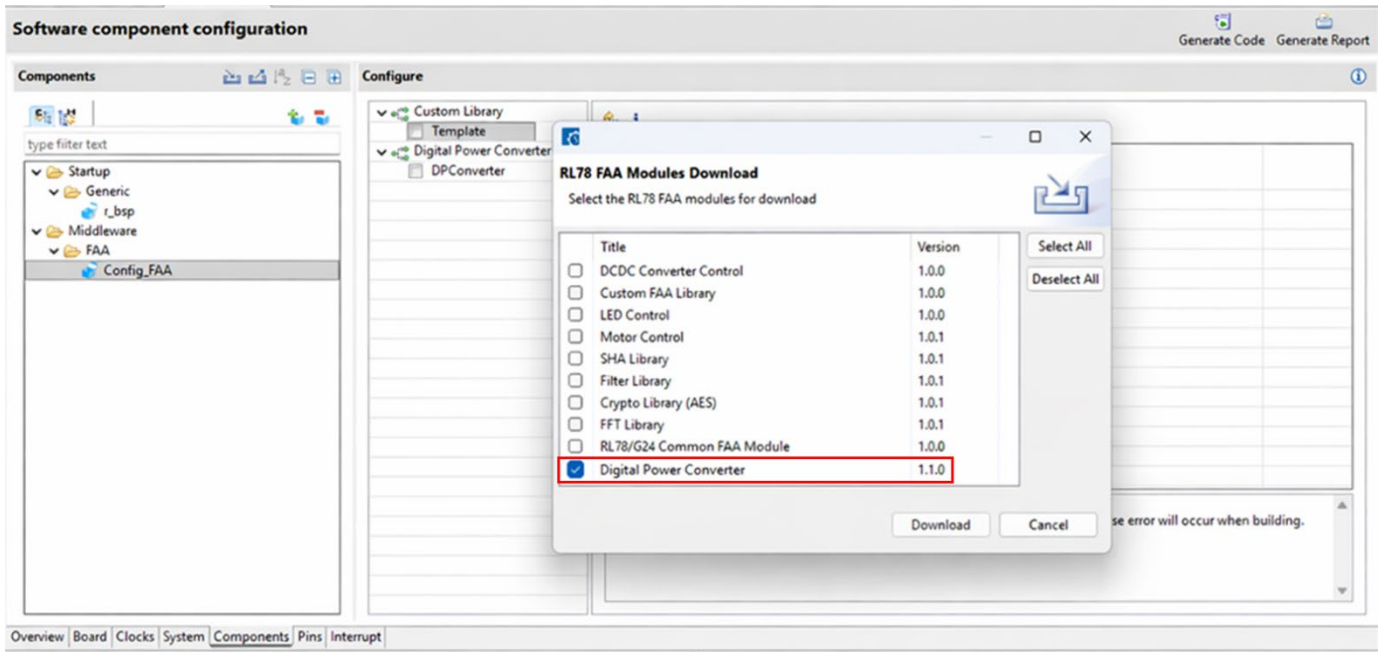


Figure 4.3 Downloading FAA Modules

4.3.5 FAA Module Configuration

Select the "Digital Power Converter" 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, refer to 7 Configuration Specifications.

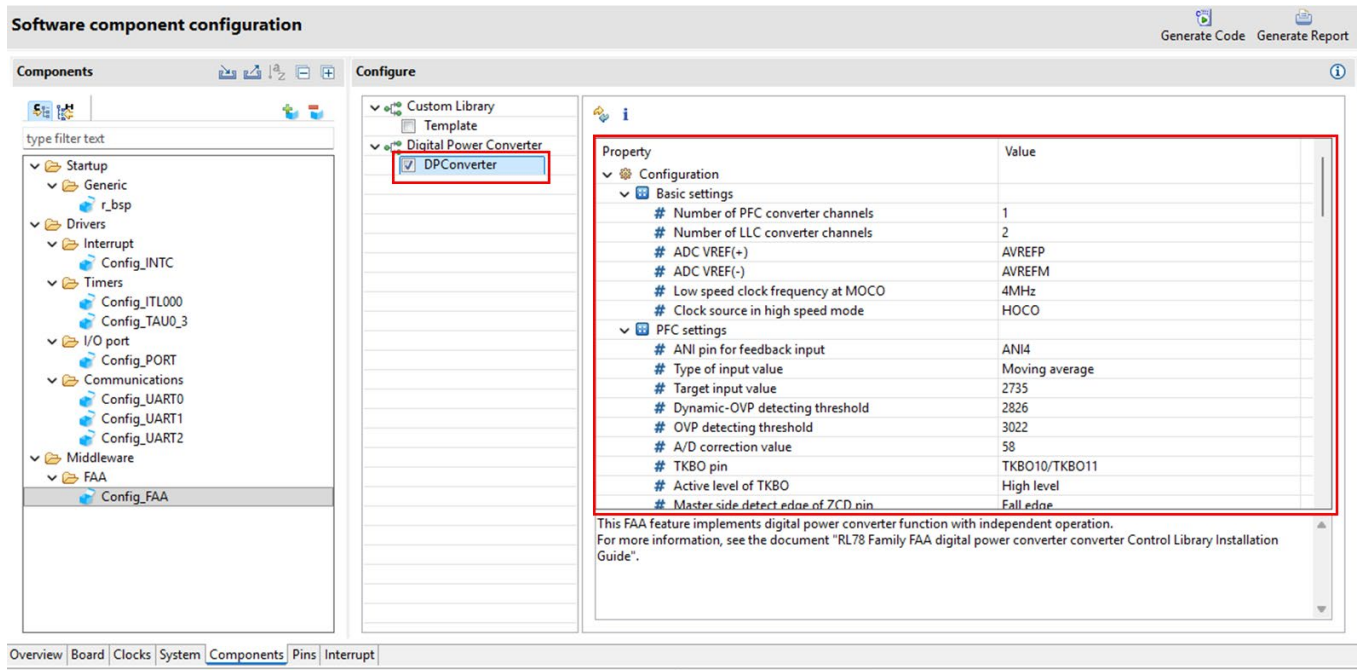


Figure 4.4 FAA Module Configuration

4.3.6 Code Generation

After setting the configuration, click the "Generate code" button to generate the source code for the Digital Power Converter Control library. The generated source code will be automatically registered in the user project on the development environment.

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. Library Structure

This section describes the structure of this library.

5.1 Architecture

This section describes the fundamental architecture of this library.

This library consists of a CPU program section and an FAA program section, with shared memory accessible from each program section. The CPU program and FAA program perform cooperative control via the shared memory.

By calling API functions from the user application, the CPU program of this library sets requests for the FAA program in shared memory. The FAA program periodically polls the shared memory and executes processing for the acquired requests. Subsequently, status information is set in the shared memory by the FAA program. Through this processing flow, the requested processing completes within 1 to 12 μ s from the API function call.

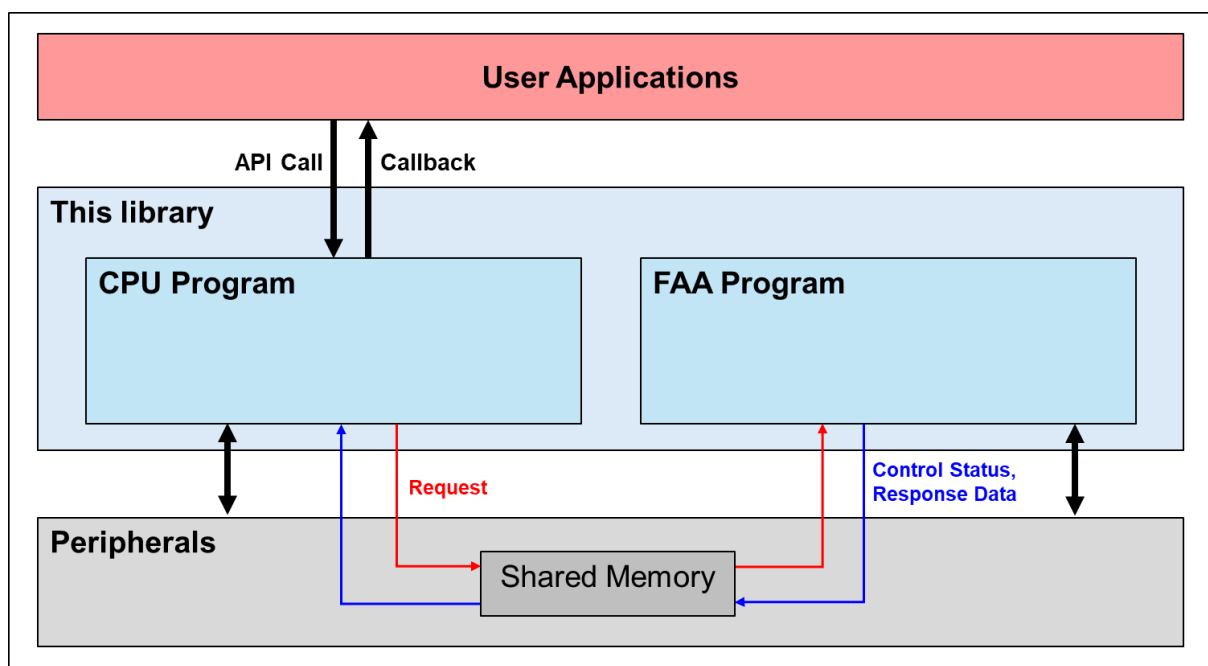


Figure 5.1 Library Architecture

5.2 Folder Structure

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

Table 5.1 Folder Structure

Folder, File Name	Description
smc_gen	SC generation folder
¥{ConfigName}	FAA component folder
{ConfigName}_common.c	FAA common function source file
{ConfigName}_common.h	FAA common function header file
{ConfigName}_common.inc	FAA common function include file
{ConfigName}_DPConv.c	Digital Power Converter Control Function source file
{ConfigName}_DPConv.h	Digital Power Converter Control Function header file
{ConfigName}_src.dsp	FAA DSP assembler source file

6. Library Control Specifications

This section describes the control specifications for the library.

6.1 Clock Operation Modes

This library supports two clock operation modes. The operation mode can be switched using the following API function.

API Function: R_{ConfigName}_DPConv_ChangeClockMode

6.1.1 Low-Speed Mode

In this mode, low-speed operation using the medium-speed on-chip oscillator (MOCO) is employed to achieve low power consumption. The MOCO clock frequency can be selected from 1/2/4 MHz. Additionally, FAA program operation is stopped, providing only the minimum operational functions via CPU programming.

When using peripheral functions in user applications during this mode, consider that the base clock becomes MOCO and configure settings appropriately.

6.1.2 High-Speed Mode

In this mode, the system's operating speed is maximized to perform feedback control for each converter. The CPU and FAA programs operate at $f_{CLK}=48$ MHz using an 8 MHz high-speed on-chip oscillator (HOCO) and PLL circuit. Additionally, the 16-bit timer KB used to implement converter control operates at 96 MHz.

6.2 PFC Converter Control

PFC converter control supports critical conduction mode (CCM) and accommodates both single-output and interleaved-output configurations. Details of the control functions are described below.

6.2.1 Fixed On-Width Output Control

This function is only active in Low-Speed Mode, providing PWM output with a fixed on-width. Operating in critical conduction mode during this state, it automatically restarts pulse output upon ZCD detection. Unlike the feedback control described later, it does not require short-cycle processing, enabling low-power operation with a slow clock. The PWM output period and on-width are determined as follows.

Period: Configuration parameter "Longest period [us]"
(Refer to Table 7.3 Smart Configurator Settings List (PFC settings) (2/2))

On width: Argument values for API function "R_{ConfigName}_DPConv_PFC_StartFixedOnWidth"

For example, in situations where the system requires a power-saving state, it is possible to achieve reduced power consumption by implementing intermittent operation. This involves initiating output via this control when the PFC output voltage level drops below a certain threshold, and stopping this control when the output voltage level rises back above a certain threshold.

A flowchart of this operation example is shown in Figure 6.1.

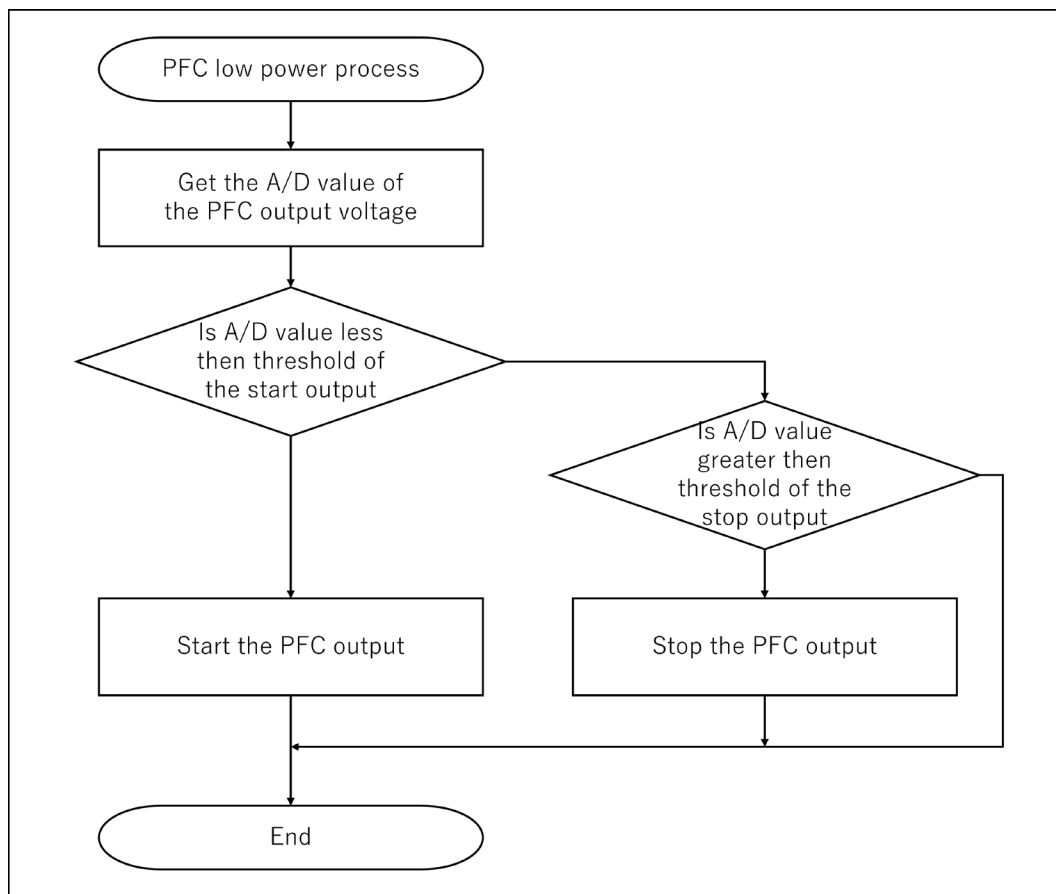


Figure 6.1 PFC Power Saving Control Example Flowchart

Figure 6.2 shows an example of operation when running based on this flowchart.

Periodically, the A/D value of the PFC output voltage is acquired. If it falls below the output start threshold (TH_{start}), the API function "R_{ConfigName}_DPConv_PFC_StartFixedOnWidth" is called to initiate PWM output with the specified On width. Each time the PWM output turns off, current flows due to the energy stored in the inductor, boosting the PFC output voltage. At this time, since the PWM output operates in Critical Conduction Mode (CRM), the PWM output restarts when the inductor current reaches 0A, triggered by the falling edge of the Zero Current Detection (ZCD) waveform.

When the boosted PWM output causes the PFC output voltage to exceed the output stop threshold (TH_{stop}), the API function "R_{ConfigName}_DPConv_PFC_Stop" is called to stop the PFC's PWM output.

This sequence of actions continues for the duration required for power-saving control.

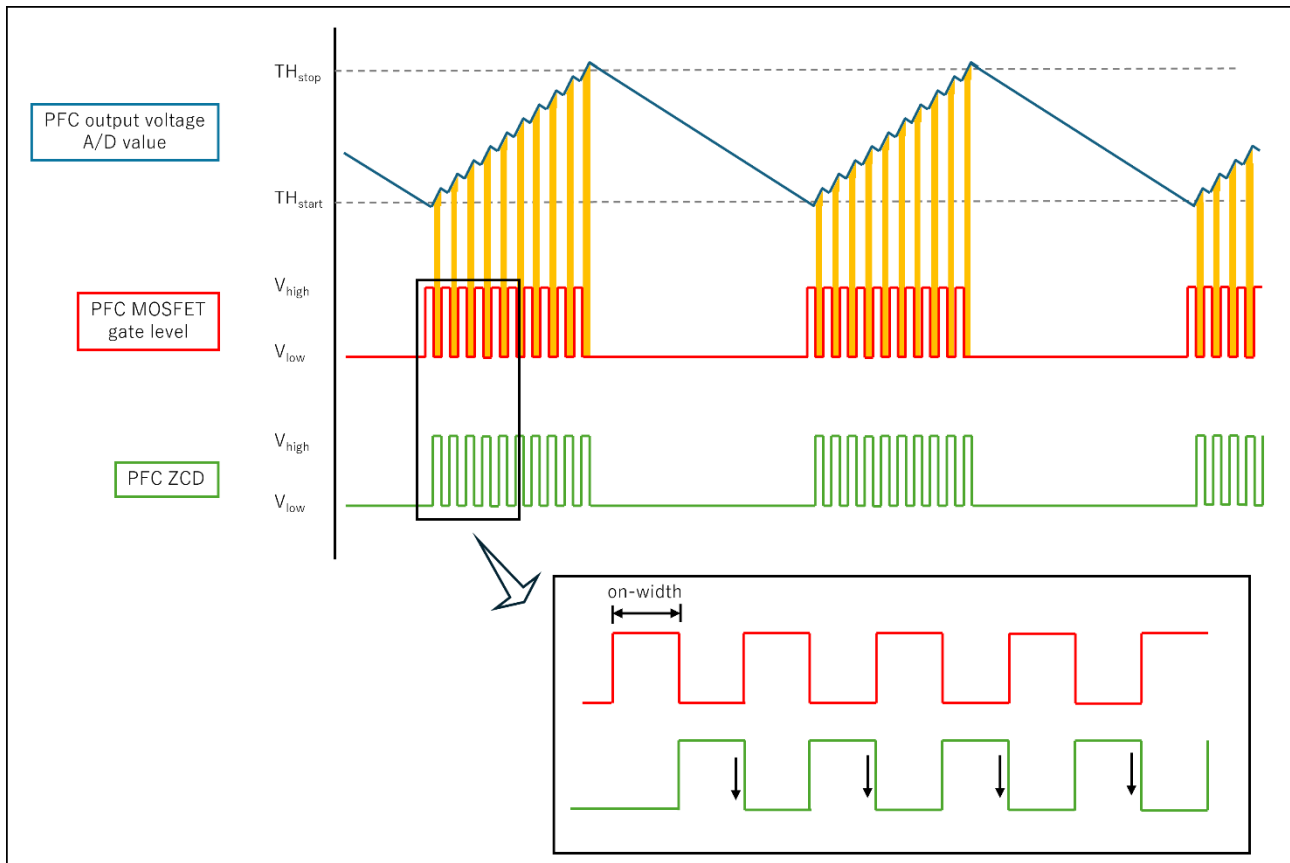


Figure 6.2 PFC Power Saving Control Operation Example

6.2.2 Feedback Control

This function is only enabled in High-Speed Mode. It monitors the PFC output voltage and performs feedback control (PI control) to maintain constant voltage.

6.2.2.1 Output Voltage Monitoring

In feedback control, the PFC output voltage is calculated using the moving average of A/D values as follows.

Moving Average Method:

A/D conversion values are sampled at 12.5 μ s intervals, and the average of the most recent N data points is calculated.

(N = PFC feedback control cycle \div 12.5 μ s)

6.2.2.2 PI (Proportional-Integral) Control

Constant voltage control in PFC converters is achieved through feedback processing based on PI (proportional-integral) control. The PI control equation is shown below.

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

D(n): Latest PWM output On-width

D(n-1): Previous PWM output On-width

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

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

A1, A2: Coefficients

Coefficients A1 and A2 are obtained from the following equation.

$$A1 = (\pi \times f_z \times T + 1) \times K_p$$

$$A2 = (\pi \times f_z \times T - 1) \times K_p$$

π : Pi (Circumference Ratio)
 f_z : Zero-Point Frequency
 T: Feedback Control Period
 K_p : Proportional Constant

Coefficients A1 and A2 are calculated based on the zero-point frequency (f_z), proportional constant (K_p), and feedback control period (T). Refer to Table 7.3 Smart Configurator Settings List (PFC settings) (2/2) for each parameter setting.

Additionally, the zero-point frequency (f_z) and proportional constant (K_p) can be dynamically changed using the following API functions.

API functions: R_{ConfigName}_DPConv_PFC_SetFz
 R_{ConfigName}_DPConv_PFC_SetKp

6.2.3 Switching Between Single and Interleaved Modes

The API function “R_{ConfigName}_DPConv_PFC_ChangeOutputMode” allows dynamic switching of the PFC output mode. Additionally, optional settings enable fade processing during mode switching and correction processing to maintain constant output energy during the switch.

6.2.3.1 Fade Processing

It is possible to specify whether to enable fade processing via the arguments when calling the API function “R_{ConfigName}_DPConv_PFC_ChangeOutputMode”.

When switching output modes without fade processing, the output mode will switch between Single (Master 100%) and Interleave (Master 51%, Slave 49%)^{Note} at the next feedback timing.

When switching output modes with fade processing enabled, the total on-width percentage is incrementally adjusted by 1% per feedback control cycle. The figure below illustrates the fade processing applied when switching from single to interleaved mode. Fade processing is applied such that the total on-width remains unchanged while increasing the slave's on-width percentage and decreasing the master's on-width percentage.

Additionally, the figure below shows the On-width ratio during fading for illustrative purposes, assuming the total On-width remains unchanged. Actually, the total On-width is recalculated for each feedback control cycle, and the calculated result is distributed between the master and slave in an appropriate ratio.

Note: If the master ON width is less than or equal to the slave ON width, the ZCD signal may not be detected within the specified time depending on circuit conditions, peripheral components, and input/output load conditions. As a result, Timer KB interleaving condition No. 8 may occur, causing pulse skipping.

To avoid this issue, this program intentionally introduces a difference between the master and slave ON widths so that the slave-side ZCD signal is detected slightly earlier.

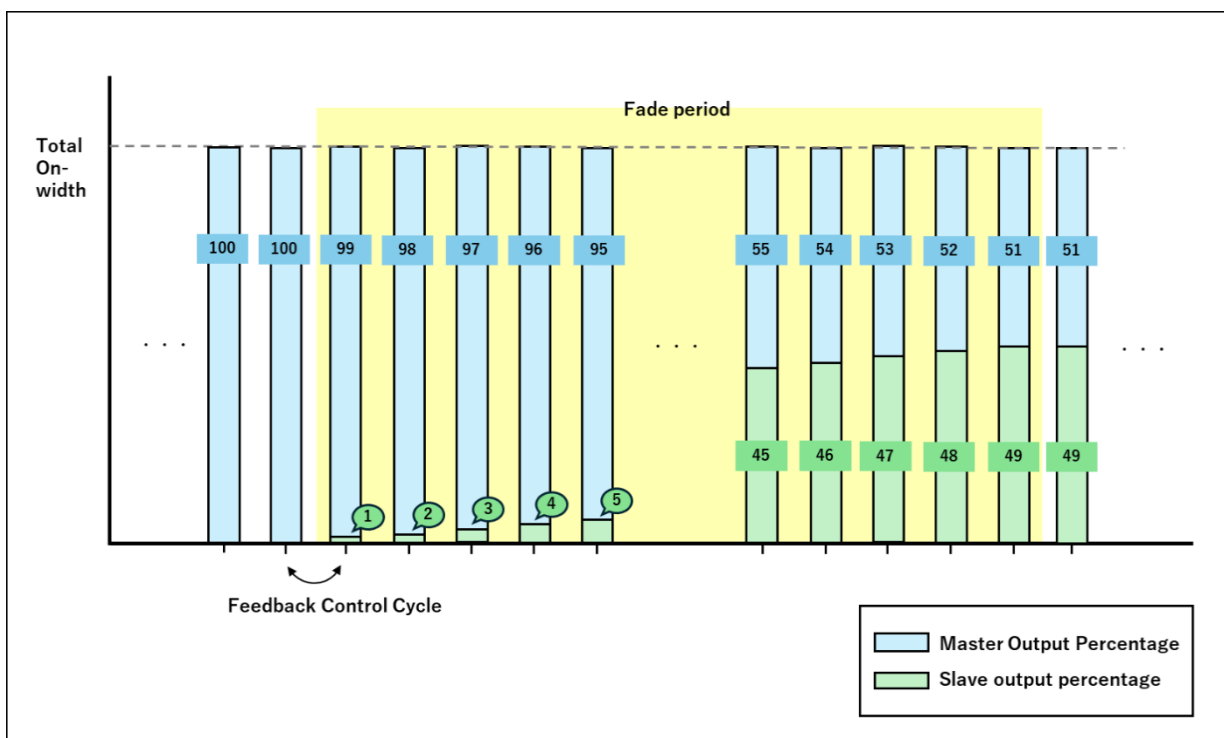


Figure 6.3 Fade processing

6.2.3.2 Output Energy Adjustment

When switching output modes, adjustments are made to ensure the total On-width value for both the master and slave remains identical. For example, when switching from single mode to interleave mode, the master's On-width in single mode serves as the reference. The On-width is appropriately distributed between the master and slave to prevent this total value from changing. When switching from interleave mode to single mode, the total On-width value for both the master and slave is set to the master On-width value.

Theoretically, output energy can be maintained at a constant level if the total On-time duration is identical. However, in practice, factors such as differences in switching rise/fall times and variations in master and slave circuitry can cause discrepancies in output energy between single-mode and interleaved- mode.

The aforementioned fade processing allows for adjustment of differences arising between single mode and interleave mode. However, to achieve more stable switching, adjustment processing can be applied.

Adjustment processing is performed using the following linear function. This method adjusts fluctuations in output energy occurring during mode switching through linear adjustment, enabling stable mode switching.

$$D_{interleave} = slope \cdot D_{single} + intercept$$

$D_{interleave}$: On -width in interleave mode

D_{single} : On -width in single mode

slope: Slope of the adjustment equation

intercept: Intercept of the adjustment equation

The slope and intercept in the above linear adjustment equation can be set to any value using the API function below.

By default, slope = 1 and intercept = 0 are applied, which effectively disables adjustment.

API function: R_{ConfigName}_DPConv_PFC_SetAdjustParamForInterleaved

6.2.4 Maximum Frequency Limit Control

The RL78/G24's 16-bit Timer KB incorporates a maximum frequency limit function. This function aims to suppress switching at excessively high frequencies in critical conduction mode (CRM) PFC control. If a ZCD (Zero Crossing Detection) trigger occurs before the maximum frequency is reached, the timer can delay restarting until the maximum frequency is reached. During this delay, the system operates in Discontinuous Conduction Mode (DCM). Suppressing switching at high frequencies is expected to reduce noise generation and switching losses.

In this library, the maximum frequency limit setting value can be changed via the following API function, allowing dynamic configuration adjustments based on system conditions.

API function: `R_{ConfigName}_DPConv_PFC_SetMaxFreqLimit`

6.2.5 Protection Functions

This section describes the protection functions in PFC converter control.

6.2.5.1 D-OVP (Dynamic Overvoltage Protection)

During PFC feedback control, if the PFC output voltage exceeds the D-OVP threshold, the PFC output is temporarily stopped via software control. This operation continues until the output voltage falls below the D-OVP threshold again, preventing overvoltage conditions.

6.2.5.2 OVP (Overvoltage Protection)

During PFC feedback control, if the PFC output voltage exceeds the OVP threshold, the PFC output is forcibly stopped via software control. This function protects the system from overvoltage conditions and prevents damage to the circuit and load.

Additionally, when OVP is detected, the user application is notified via a callback function.

Since the LLC function provided by this library operates independently, users must manually stop LLC operation if desired upon detection of PFC overvoltage (OVP).

6.2.5.3 OCP (Overcurrent Protection)

In PFC feedback control and fixed on-width output control, if the voltage-converted MOSFET source current value of the master and slave exceeds the OCP (overcurrent protection) threshold, the forced stop function of the 16-bit timer KB built into the RL78/G24 microcontroller activates, immediately forcing the PFC output to stop. This function protects the system from overcurrent conditions and prevents damage to the circuit and load.

Additionally, when OCP is detected, the user application is notified via a callback function.

Since the LLC function provided by this library operates independently, users must manually stop LLC operation if desired upon detection of OCP of PFC.

6.3 LLC Converter Control

This library provides LLC converter control functionality for up to two channels. Details of the control functions are described below.

6.3.1 Feedback Control

This function is only enabled in High-Speed Mode. It monitors the LLC output voltage and performs feedback control (PI control) to maintain a constant voltage.

6.3.1.1 Output Voltage Monitoring -- Pat. P.

There are two methods for monitoring the LLC output voltage in feedback control: moving average and binary decision.

1. Moving Average

A/D values are sampled at $12.5\mu\text{s}$ intervals, and the average of the most recent N times is calculated.

$$(N = \text{LLC feedback control cycle} \div 12.5\mu\text{s})$$

This monitoring method is effective in circuits that measure the LLC output voltage at an absolute voltage level.

For example, it applies to circuits that measure the secondary-side output voltage in an LLC circuit by dividing it and inputting it to the ANI pin of a microcontroller placed on the secondary side.

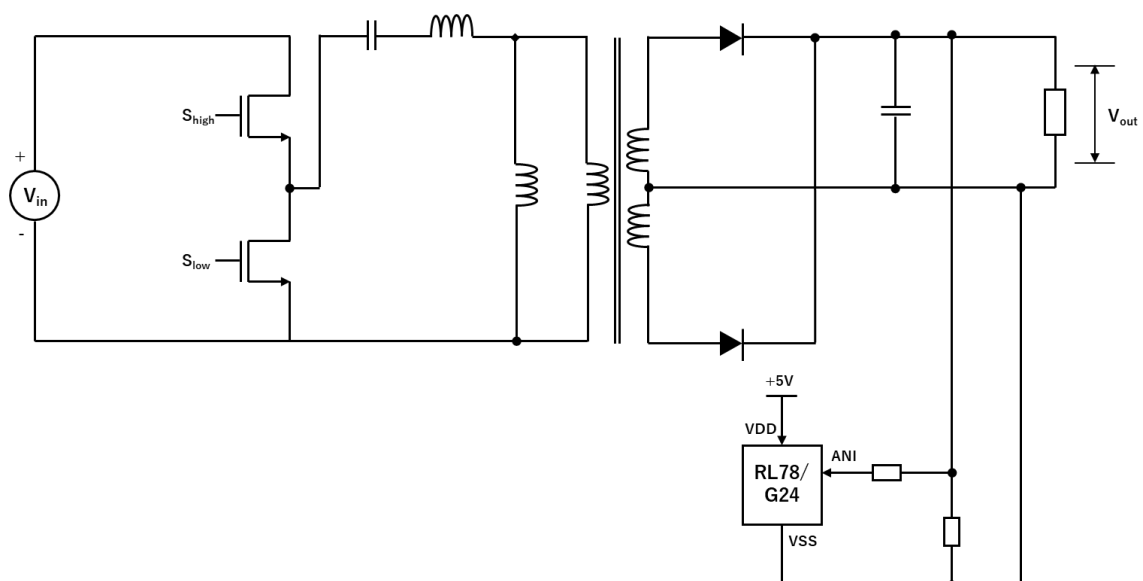


Figure 6.4 Example LLC Output Voltage Measurement Circuit on the Secondary Side

2. Binary Decision

The A/D value is sampled at $12.5\ \mu\text{s}$ intervals, and the number of times the target voltage was exceeded among the most recent N times is counted. ($N = \text{LLC feedback control cycle} \div 12.5\ \mu\text{s}$)

This monitoring method is effective in circuits that determine the LLC output level based on relative voltage levels (whether above or below the target voltage level).

For example, a circuit configuration where the secondary-side output voltage level in an LLC circuit is measured by a microcontroller placed on the primary side via a photocoupler.

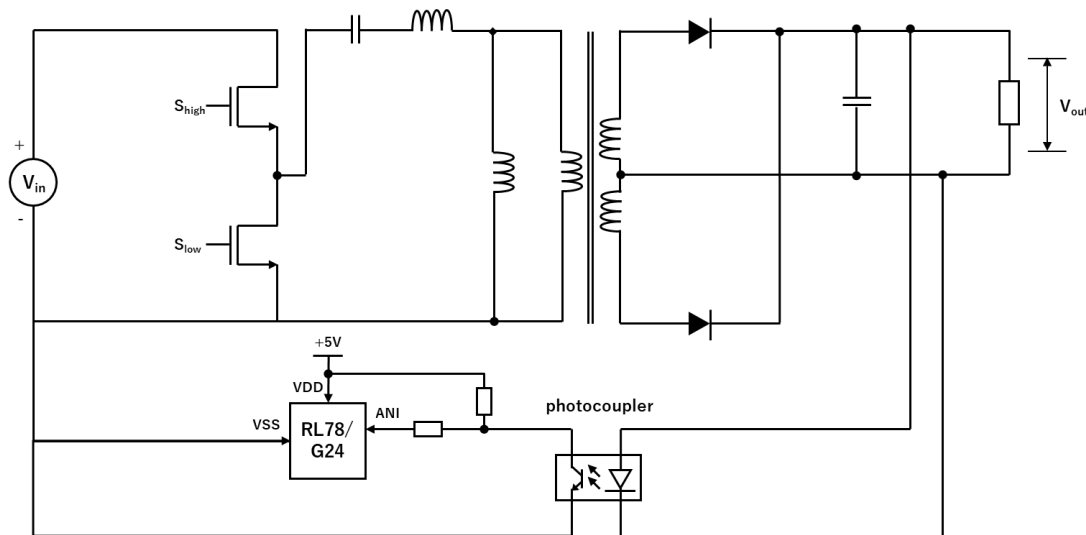


Figure 6.5 Example of a Simple LLC Output Voltage Measurement Circuit on the Primary Side Using a Photocoupler

In such circuits, the output voltage of the photocoupler undergoes a sudden level change near the reference voltage (target voltage), as shown in the figure below. Therefore, it is usually necessary to devise a method to measure the secondary-side output state using an A/D converter.

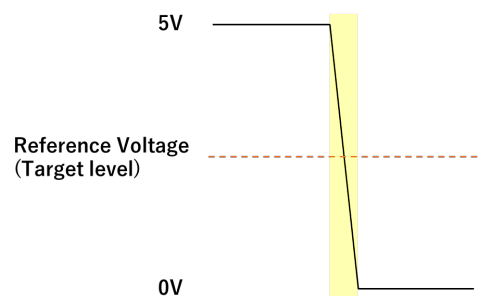


Figure 6.6 Voltage measurement using a photocoupler

This monitoring method counts the number of times the output voltage level exceeds the reference voltage within a fixed time period, using this result as an input value for feedback control. Feedback control adjusts the output value to balance the number of times the voltage exceeds the reference voltage with the number of times it falls below it, thereby converging the LLC output voltage to the reference voltage.

For example, when sampling 16 times for A/D values, the target value for feedback control is 8 counts.

6.3.1.2 Soft Start Control

Soft start control suppresses inrush current during startup by gradually reducing the LLC output frequency while starting the converter. This control also mitigates abrupt fluctuations in switching frequency, thereby suppressing audible noise generation.

This control starts when a feedback control request is generated via the API function "R_{ConfigName}_DPConv_LLCn_StartFeedbackControl" and ends when the LLC output voltage reaches the final target value or the maximum duration elapses. During the control period, the LLC output frequency is updated at 25 μ s intervals. Each parameter is determined based on the following:

Start Frequency: Configuration parameter "Frequency at the start [kHz]"

End Frequency: Configuration parameter "Frequency at the end [kHz]"

Max Duration: Configuration parameter "Max duration [ms]"

End Target Value: Configuration parameter "Analog input value for terminate"

Step width: $(\text{Start Frequency} - \text{End Frequency}) \div (\text{Max Duration} \div 25 \text{ us})$

For details on each configuration parameter, refer to Table 7.4 Smart Configurator Setting Item List (LLCn setting) (1/2).

6.3.1.3 PI (Proportional-Integral) Control

Constant voltage control in the LLC converter is achieved through feedback processing based on PI (proportional-integral) control. The PI control equation is shown below.

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

P(n): Latest PFM output cycle

P(n-1): Previous PFM output cycle

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

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

A1, A2: Coefficients

Coefficients A1 and A2 are obtained from the following equation.

$$A1 = (\pi \times f_z \times T + 1) \times K_p$$

$$A2 = (\pi \times f_z \times T - 1) \times K_p$$

π : Pi (Circumference Ratio)
 f_z : Zero-Point Frequency
 T: Feedback Control Period
 K_p : Proportional Constant

Coefficients A1 and A2 are calculated based on the zero-point frequency (f_z), proportional constant (K_p), and feedback control period (T). For parameter settings, refer to Table 7.5 Smart Configurator Setting Item List (LLCn setting) (2/2).

6.3.2 Protection Functions

This section describes the protection functions in LLC converter control.

6.3.2.1 D-OVP (Dynamic Overvoltage Protection)

During LLC feedback control, if the LLC output frequency exceeds the D-OVP threshold, the LLC output is temporarily halted via software control. This operation continues until the calculated LLC output frequency falls below the D-OVP threshold during feedback processing, preventing overvoltage conditions.

6.3.2.2 OCP (Overcurrent Protection)

When the LLC resonant current value exceeds the OCP (overcurrent protection) threshold in voltage equivalent, the forced shutdown function of the 16-bit timer KB built into the RL78/G24 microcontroller activates, immediately forcing the LLC output to stop. This function protects the system from overcurrent conditions and prevents damage to the circuit and load.

OCP monitoring is enabled when feedback control starts and disabled when control stops. Furthermore, it can be enabled or disabled at any time via the following API functions.

API function: R_{ConfigName}_DPConv_LLcN_EnableOCP
R_{ConfigName}_DPConv_LLcN_DisableOCP

Additionally, when OCP is detected, notification is sent to the user application via a callback function.

Since the PFC functionality provided by this library operates independently, users must manually stop PFC operation if they wish to halt it upon detection of OCP of LLC.

6.4 A/D Conversion Control for User Applications

This library provides the following A/D conversion functions. If A/D conversion is required in the CPU-side user application, use this conversion functionality.

6.4.1 A/D Conversion Control in Low-Speed Mode

In Low-Speed Mode, request A/D conversion processing using the API function below. This function can acquire A/D values for the specified number of times as an argument and obtain their average value.

API function: R_{ConfigName}_DPConv_GetAdValue

6.4.2 A/D Conversion Control in High-Speed Mode

In High-Speed Mode, use the following API functions to issue A/D conversion requests and retrieve conversion results.

API function: R_{ConfigName}_DPConv_RequestAdValue

R_{ConfigName}_DPConv_GetRequestedAdValue

6.5 Data Read Function for User Applications

This library provides data reading functions within the library itself. When a user application on the CPU side requires values from the FAA data memory, it can issue data read requests and obtain the read results using the API functions listed below.

API function: R_{ConfigName}_DPConv_RequestData

R_{ConfigName}_DPConv_GetRequestedData

6.6 Control Flow

6.6.1 Feedback Control Sequence

This library accepts requests from user applications to initiate feedback control and retrieve status information via API functions. Internally, the library coordinates control between CPU programs and FAA programs through SHDMEM (shared data memory).

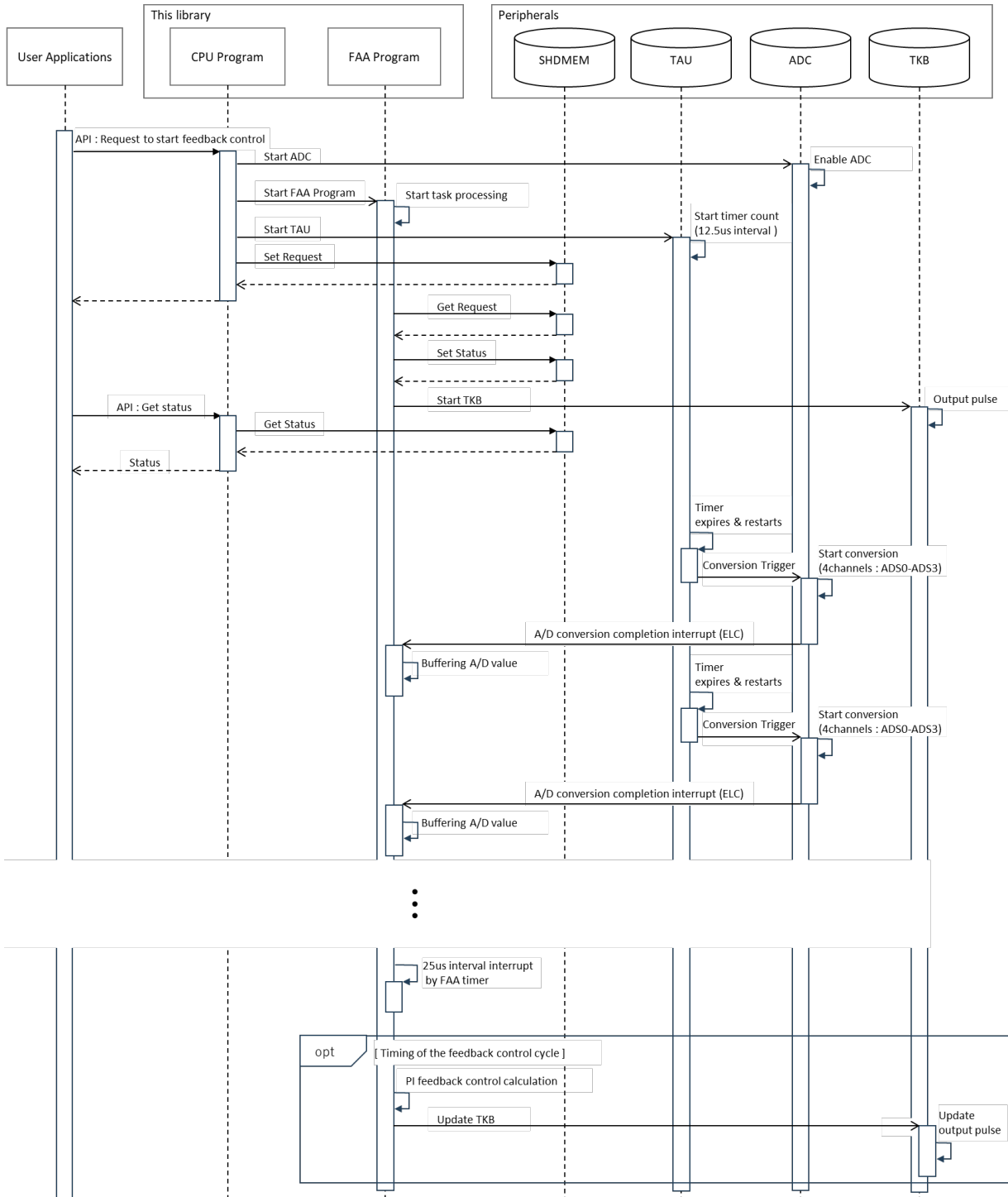


Figure 6.7 Feedback Control Sequence

6.6.2 Task Processing Flow (FAA Program)

This section describes task processing in the FAA program. After starting, the FAA program performs initialization processing and then executes the following processing within a loop.

- API request acquisition and execution processing
- PFC feedback control processing (executed per PFC feedback control cycle)
- LLC soft start control processing (executed every 25 μ s)
- LLC feedback control processing (executed per LLC feedback control cycle)

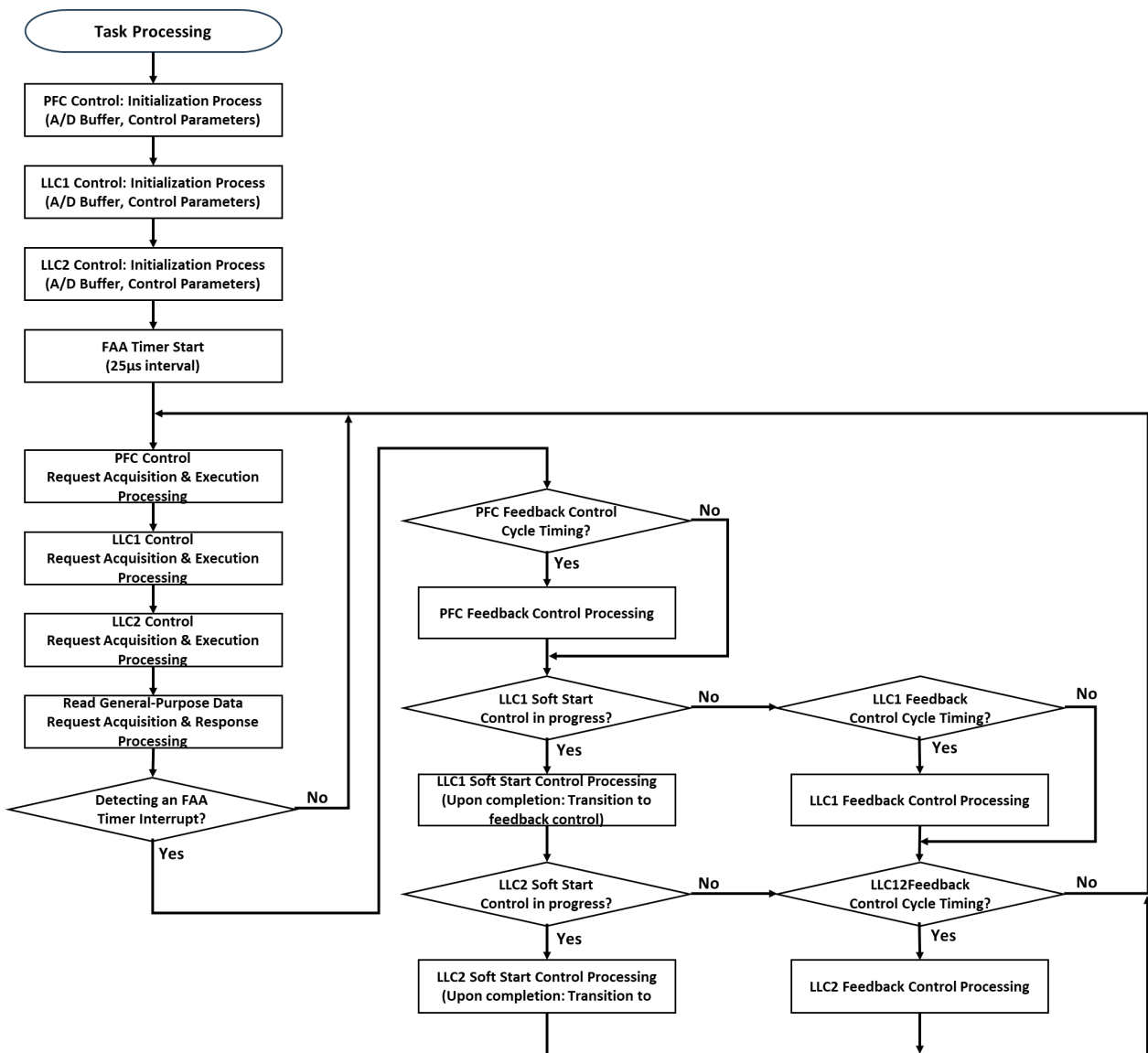


Figure 6.8 Task Processing Flow (FAA Program)

6.6.3 A/D Conversion Complete Interrupt Processing Flow (FAA Program)

This section describes the A/D conversion completion interrupt processing flow in the FAA program.

First, an interrupt occurs from the TAU at 12.5 μs intervals. This interrupt serves as the trigger to start the A/D conversion process. Using the ADC's advanced mode on the RL78/G24 microcontroller enables continuous A/D conversion across all four channels. Once conversion for all four channels is complete, the interrupt function defined within the FAA program is called via the Event Link Controller (ELC).

Within the FAA program's interrupt function, the converted A/D values are acquired, and buffer values and total values are updated for each feedback control. Additionally, if a general-purpose A/D conversion request is received from the user via the API, the target channel is updated and the conversion result is acquired.

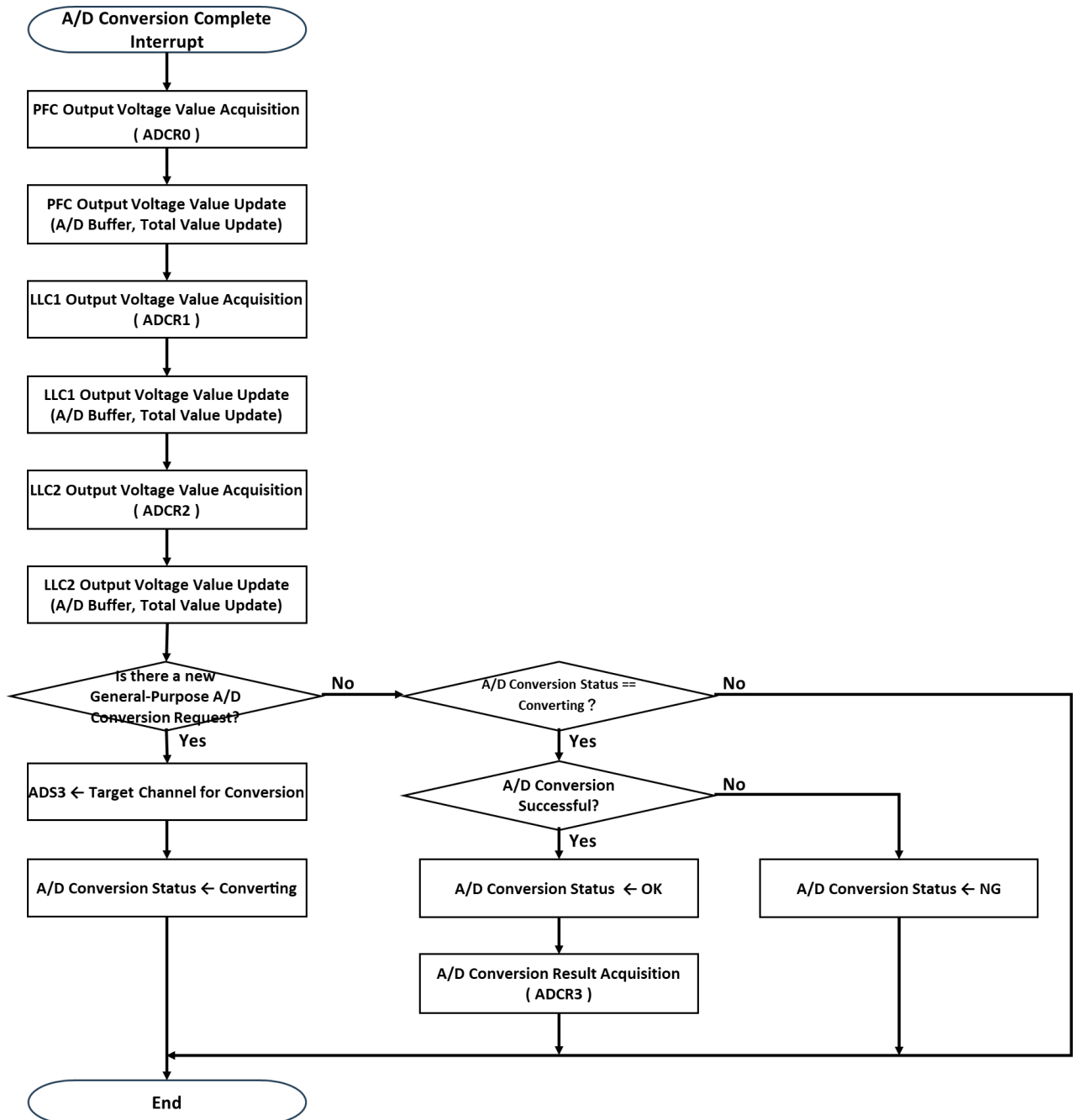


Figure 6.9 A/D Conversion Complete Interrupt Processing Flow (FAA Program)

7. Configuration Specifications

This section describes the configuration specifications for this library.

The following lists the configuration items that can be set using the Smart Configurator (SC).

7.1 Basic Settings Items

This section describes the Basic Settings items in SC Configuration.

Table 7.1 List of Smart Configurator setting items (Basic settings)

property	value	description
Number of PFC converter channels	0~1	Select the number of control channels for the PFC converter.
Number of LLC converter channels	0~2	Select the number of control channels for the LLC converter.
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.
Low speed clock frequency at MOCO	1MHz, 2MHz, 4MHz	Select the operating clock for Low-Speed Mode.
Clock source in high speed mode	HOCO, X1	Select the clock source for High-Speed Mode.

Note: Settings where the total number of PFC converters and LLC converters equals zero are not possible.

7.2 PFC Settings

This section describes the PFC settings items in SC Configuration.

Table 7.2 Smart Configurator Settings List (PFC settings) (1/2)

property	value	description
ANI pin for feedback input	ANI0, ANI1, ANI2, ANI3, ANI4, ANI5, ANI6, ANI7, ANI16, ANI17, ANI18, ANI19, ANI20, ANI21, ANI22, ANI23, ANI24, ANI25, ANI26, ANI27, ANI28, ANI29, ANI30	Select the ANI pin to use for feedback input.
Type of input value	Moving average	Select the input value type. For details, refer to Section 6.2.2.1.
Target input value	$0 \leq \text{value} \leq 4095$	Specify the target value for PI feedback control.
Dynamic-OVP detecting threshold	"Target input value" < value ≤ 4095	Specify the Dynamic-OVP threshold. (Protection function that temporarily halts output while the output voltage exceeds the threshold)
OVP detecting threshold	"Dynamic-OVP detecting threshold" < value ≤ 4095	Specify the input threshold for detecting OVP. (Protection function that permanently halts output when the output voltage exceeds the threshold)
A/D correction value	$0 < \text{value} \leq 4095$	Specify the correction value for the A/D acquisition value. Correction is necessary when the output impedance of the signal input to the target analog terminal is high.
TKBO pin	TKBO00/TKBO01, TKBO10/TKBO11, TKBO20/TKBO21	Select the TKBO pin for output.
Active level of TKBO	Low level, High level	Select the active level of TKBO.
Master side detect edge of ZCD pin	Rise edge, Fall edge	Select the detection edge (polarity) for the master-side ZCD pin.
Slave side detect edge of ZCD pin	Rise edge, Fall edge	Select the detection edge (polarity) for the slave-side ZCD pin.

Table 7.3 Smart Configurator Settings List (PFC settings) (2/2)

property		value	description
Longest period [us]		10 <= value <= 50	Specify the maximum period of the output pulse. (Output period setting when ZCD is not detected)
Minimum on-time [us]		0.5 <= value <= 20	Specify the minimum on-time of the output pulse.
Maximum on-time of each output pulse [us]		10 <= value <= 50	Specify the maximum on-time of the output pulse.
PI parameters	Zero point frequency [Hz]	1 <= value <= 65535	Specify the zero point frequency for PI feedback control.
	Feedback period [us]	100, 200, 300, 400	Select the PI feedback control period.
	Proportional Gain (Kp)	0 < value < 16	Specify the proportional gain (Kp) for PI feedback control.
Comparator combination of PFC OCP (Over Current Protection)		Disable, IVCMP0(M)+IVCMP1(S), IVCMP1(M)+IVCMP0(S), IVCMP1(M)+IVCMP3(S), IVCMP3(M)+IVCMP1(S), IVCMP0(M)+IVCMP3(S), IVCMP3(M)+IVCMP0(S)	Select the comparator input pin used for OCP (overcurrent protection). Select Disable if OCP is not to be applied.
OCP threshold voltage ratio to VDD		0 < value < 16	Specify the OCP threshold (voltage equivalent value) as a ratio relative to the VDD voltage value. e.g. When VDD = 5V and OCP threshold = 2.4V, the parameter value is calculated as follows: $2.4V / 5V = 0.48$

7.3 LLC Settings

This section describes the LLCn settings items in the SC configuration. (n = 1, 2)

Table 7.4 Smart Configurator Setting Item List (LLCn setting) (1/2)

property		value	description
ANI pin for feedback input		ANI0, ANI1, ANI2, ANI3, ANI4, ANI5, ANI6, ANI7, ANI16, ANI17, ANI18, ANI19, ANI20, ANI21, ANI22, ANI23, ANI24, ANI25, ANI26, ANI27, ANI28, ANI29, ANI30	Select the ANI pin to use for feedback input.
Type of input value		Moving average, Binary total	Select the input value type. For details, refer to Section 6.3.1.1.
Target input value		0 <= value <= 4095	Specify the target value for PI feedback control.
Active level at over target voltage		Low level, High level	Select the photocoupler output level when the input value exceeds the target value. *Valid only when the input type is Binary total.
TKBO pin		TKBO00/TKBO01, TKBO10/TKBO11, TKBO20/TKBO21	Select the TKBO pin for output.
Active level of TKBO		Low level, High level	Select the active level of TKBO.
Deadtime [ns]		50 <= value <= 500	Specify the dead time [ns].
Soft start parameters	Frequency at the start [kHz]	20 <= value <= 500	Specify the frequency [kHz] at the start of soft start.
	Frequency at the end [kHz]	20 < value <= "Frequency at the start"	Specify the frequency [kHz] at the end of soft start.
	Max duration [ms]	0 < value <= 500	Specify the maximum duration [ms] for soft start.
	Analog input value for terminate	0 <= value <= 4095	Specify the soft start end determination threshold (analog value).

Table 7.5 Smart Configurator Setting Item List (LLCn setting) (2/2)

property		value	description
PI parameters	Frequency at the start [kHz]	20 <= value <= 500	Specify the start frequency for PI feedback.
	Lowest frequency [kHz]	20 <= value <= 100	Specify the minimum frequency [kHz].
	Highest frequency [kHz]	Lowest frequency <= value <= 500	Specify the maximum frequency [kHz].
	Zero point frequency [Hz]	0 < value <= 10000	Specify the zero point frequency for PI feedback control.
	Feedback period [us]	100, 200, 300, 400	Select the PI feedback control cycle.
	Proportional Gain (Kp)	0 < value < 10	Specify the proportional gain (Kp) for PI feedback control.
OCP(Over Current Protection)		Disable, IVCMP0, IVCMP1, IVCMP2, IVCMP3	Select the comparator input pin to use for OCP (Over Current Protection). Select Disable if OCP is not applied.
OCP threshold voltage ratio to VDD		0 < value < 1	Specify the OCP threshold (voltage equivalent value) as a ratio relative to the VDD voltage value. e.g. When VDD = 5V and OCP threshold = 4V, the parameter value is calculated as follows: $4V / 5V = 0.8$

7.4 User Application Settings

This section describes the User Application Settings (User settings) items in SC Configuration.

Table 7.6 Smart Configurator Settings List (User setting)

property	value	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	
ANI5	Unused/Used	
ANI6	Unused/Used	
ANI7	Unused/Used	
ANI16	Unused/Used	
ANI17	Unused/Used	
ANI18	Unused/Used	
ANI19	Unused/Used	
ANI20	Unused/Used	
ANI21	Unused/Used	
ANI22	Unused/Used	
ANI23	Unused/Used	
ANI24	Unused/Used	
ANI25	Unused/Used	
ANI26	Unused/Used	
ANI27	Unused/Used	
ANI28	Unused/Used	
ANI29	Unused/Used	
ANI30	Unused/Used	

8. API Information

8.1 API Typedef Definitions

This section describes the Typedef definitions provided by this library.

8.1.1 e_faa_peripheral_t

This typedef defines peripheral functions used in this library.

```
typedef enum
{
    FAA_PERIPHERAL_ADC,
    FAA_PERIPHERAL_TKB,
    FAA_PERIPHERAL_TAU0,
    FAA_PERIPHERAL_PGACOMP,
    FAA_PERIPHERAL_DAC,
} e_faa_peripheral_t;
```

8.1.2 e_faa_clock_mode_t

This typedef defines the clock operating mode.

```
typedef enum
{
    CLOCK_MODE_HIGH_SPEED = 0,
    CLOCK_MODE_LOW_SPEED = 1,
} e_faa_clock_mode_t;
```

8.1.3 e_faa_pfc_status_t

This typedef defines status information for PFC converter control.

```
typedef enum
{
    FAA_PFC_STATUS_STOP = 0x00,
    FAA_PFC_STATUS_FIXED_WIDTH = 0x01,
    FAA_PFC_STATUS_FEEDBACK_SINGLE = 0x02,
    FAA_PFC_STATUS_FEEDBACK_INTERLEAVED = 0x04,
    FAA_PFC_STATUS_DETECT_OVP = 0x10,
    FAA_PFC_STATUS_DETECT_OCP = 0x20,
} e_faa_pfc_status_t;
```

8.1.4 e_faa_pfc_output_t

This typedef defines the output mode in PFC converter control.

```
typedef enum
{
    FAA_PFC_OUTPUT_SINGLE = 0,
    FAA_PFC_OUTPUT_INTERLEAVED = 1,
} e_faa_pfc_output_t;
```

8.1.5 e_faa_llc_status_t

This typedef defines status information for LLC converter control.

```
typedef enum
{
    FAA_LLC_STATUS_STOP = 0,
    FAA_LLC_STATUS_FEEDBACK = 1,
    FAA_LLC_STATUS_DETECT_OCP = 2,
    FAA_LLC_STATUS_DETECT_ERROR = 255,
} e_faa_llc_status_t;
```

8.1.6 e_faa_ad_channel_t

This typedef defines the analog input channels. It is used to specify the channels to be converted in the general-purpose A/D conversion control provided for user applications.

The target analog channel definitions are generated based on the settings in Table 7.6 Smart Configurator Settings List (User setting).

```
typedef enum
{
    ADCHANNEL0 = 0,
    ADCHANNEL1 = 1,
    ADCHANNEL2 = 2,
    . . .
    ADTEMPERSENSOR0 = 32,
    ADINTERREFVOLT = 33,
} e_faa_ad_channel_t;
```

8.1.7 e_faa_result_adc_t

This typedef defines the conversion results in the general-purpose A/D conversion control provided for user applications.

```
typedef enum
{
    FAA_ADC_NO_REQUEST = 0,
    FAA_ADC_CONVERTING = 1,
    FAA_ADC_CONVERT_OK = 2,
    FAA_ADC_CONVERT_NG = 3,
} e_faa_result_adc_t;
```

8.1.8 e_faa_request_id_t

This typedef defines the Request ID for the data read function.

```
typedef enum
{
    FAA_REQUEST_ID_NONE          = 0x0000,
    FAA_REQUEST_ID_PFC_V_FB      = 0x0001,
    FAA_REQUEST_ID_LLC1_V_FB     = 0x0002,
    FAA_REQUEST_ID_LLC2_V_FB     = 0x0003,
    FAA_REQUEST_ID_SPECIFIED_ADDR = 0xFFFF,
} e_faa_request_id_t;
```

8.1.9 faa_pfc_callback_t

This typedef defines the callback type for error detection in PFC converter control.

```
typedef struct
{
    void (*p_detect_ovp)(void);          /* callback for detecting PFC OVP */
    void (*p_detect_ocp)(void);          /* callback for detecting PFC OCP */
} faa_pfc_callback_t;
```

8.1.10 faa_llc_callback_t

This typedef defines the callback type for exception detection in LLC converter control.

```
typedef struct
{
    void (*p_detect_ocr)(void);          /* callback for detecting LLC OCP */
} faa_llc_callback_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.

8.2.1 R_{ConfigName}_DPConv_Create

This function performs the initialization processing for peripheral functions required in digital power converter control.

Format

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

Parameters

None

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function initializes the following peripheral functions used in digital power converter control.

- A/D converter
- D/A converter
- Comparator
- 16-bit timer KB
- Timer Array Unit 0
- External Interrupt

Special Notes:

Since the function call processing is included in the source code generated by the Smart Configurator, no function call processing is required from the user program.

8.2.2 R_{ConfigName}_DPConv_PowerOn

This function enables clock supply to peripheral functions required for digital power converter control.

Format

```
void R_{ConfigName}_DPConv_PowerOn (e_faa_peripheral_t peripheral)
```

Parameters

peripheral: Target peripheral functions

FAA_PERIPHERAL_ADC: A/D converter

FAA_PERIPHERAL_TKB: 16-bit timer TB

FAA_PERIPHERAL_TAU0: Timer array unit 0

FAA_PERIPHERAL_PGACOMP: PGA/Comparator

FAA_PERIPHERAL_DAC: D/A Converter

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function enables clock supply to the peripheral functions specified by the arguments. (It sets the corresponding bits in peripheral enable registers 0-2.)

When used in combination with the later-described R_{ConfigName}_DPConv_PowerOff function, clock supply can be enabled or disabled on a per-peripheral function basis for each peripheral function used within this library. Minimizing clock supply according to the system's state enables power consumption reduction.

Special Notes:

In High-Speed Mode, enable clock supply for all peripherals defined in e_faa_peripheral_t. In Low-Speed Mode, configure settings optimally based on user application requirements.

8.2.3 R_{ConfigName}_DPConv_PowerOff

This function disables clock supply to peripheral functions required for digital power converter control.

Format

```
void R_{ConfigName}_DPConv_PowerOff (e_faa_peripheral_t peripheral)
```

Parameters

peripheral: Target peripheral functions

FAA_PERIPHERAL_ADC: A/D converter

FAA_PERIPHERAL_TKB: 16-bit timer TB

FAA_PERIPHERAL_TAU0: Timer array unit 0

FAA_PERIPHERAL_PGACOMP: PGA/Comparator

FAA_PERIPHERAL_DAC: D/A Converter

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function disables clock supply to the peripheral function specified by the argument. (It clears the corresponding bit in peripheral enable registers 0-2.)

When used in combination with the previously mentioned R_{ConfigName}_DPConv_PowerOn, this function enables or disables clock supply for each peripheral used within this library on a per-peripheral basis. Minimizing clock supply according to the system's state allows for reduced power consumption.

Special Notes:

In High-Speed Mode, enable clock supply for all peripheral functions defined by e_faa_peripheral_t. In Low-Speed Mode, configure settings optimally based on user application requirements.

8.2.4 R_{ConfigName}_DPConv_ChangeClockMode

This function changes the clock operating mode.

Format

```
void R_{ConfigName}_DPConv_ChangeClockMode (e_faa_clock_mode_t clock)
```

Parameters

clock: Clock operation mode

CLOCK_MODE_HIGH_SPEED : High speed mode ($f_{CLK}=48\text{MHz}$, $f_{TIMER}=96\text{MHz}$)

CLOCK_MODE_LOW_SPEED : Low speed mode ($f_{CLK}=1\text{or}2\text{or}4\text{MHz}$) *Clock set via SC

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function switches the clock operation mode to the mode specified by the argument.

For details on the available clock operation modes, refer to Section 6.1 Clock Operation Modes.

Special Notes:

None

8.2.5 R_{ConfigName}_DPConv_PFC_Init

This function initializes the PFC converter control function.

Format

```
void R_{ConfigName}_DPConv_PFC_Init (faa_pfc_callback_t * p_callback)
```

Parameters

p_callback: Pointer to callback function

p_detect_ovp: Function pointer for registering callback function upon PFC overvoltage detection

p_detect_ocp: Function pointer for registering callback function upon PFC overcurrent detection

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function initializes parameters and registers user callback functions for the PFC converter control function.

Special Notes:

If there is no need for a user callback upon detecting OVP or OCP, set the member variable of the faa_pfc_callback_t type passed as an argument to NULL.

8.2.6 R_{ConfigName}_DPConv_PFC_StartFixedOnWidth

This function initiates fixed- On width output control.

Format

```
void R_{ConfigName}_DPConv_PFC_StartFixedOnWidth  
      (uint16_t on_width_time_ns)
```

Parameters

on_width_time_ns: Pulse On-width time [ns]

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function initiates output control with a fixed on-width based on the pulse on-width time specified as an argument.

Special Notes:

This function is only available in Low-Speed Mode.

8.2.7 R_{ConfigName}_DPConv_PFC_Suspend

This function temporarily suspends the PFC output.

Format

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

Parameters

None

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function temporarily suspends the PFC output.

Special Notes:

None

8.2.8 R_{ConfigName}_DPConv_PFC_Resume

This function resumes PFC output.

Format

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

Parameters

None

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function resumes PFC output.

Special Notes:

None

8.2.9 R_{ConfigName}_DPConv_PFC_StartFeedbackControl

This function initiates PFC feedback control.

Format

```
void R_{ConfigName}_DPConv_PFC_StartFeedbackControl  
      (uint16_t on_width_time_ns)
```

Parameters

on_width_time_ns: Pulse on-width time [ns]

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function initiates PFC feedback control. The initial output on-width is calculated based on the pulse on-width time specified by the argument. Additionally, OCP detection is enabled.

Special Notes:

This function is only available in High-Speed Mode.

8.2.10 R_{ConfigName}_DPConv_PFC_Stop

This function stops PFC converter control.

Format

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

Parameters

None

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function stops PFC converter control.

Special Notes:

None

8.2.11 R_{ConfigName}_DPConv_PFC_TaskForFeedbackControl

This function performs task processing for PFC feedback control.

Format

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

Parameters

None

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function performs task processing for PFC feedback control.

Special Notes:

This function should only be called periodically in High-Speed Mode.

8.2.12 R_{ConfigName}_DPConv_PFC_GetStatus

This function acquires status information for PFC converter control.

Format

```
e_faa_pfc_status_t R_{ConfigName}_DPConv_PFC_GetStatus (void)
```

Parameters

None

Return Values

PFC Status Information

FAA_PFC_STATUS_STOP: Stopped

FAA_PFC_STATUS_FIXED_WIDTH: Fixed On-width output operation in progress

FAA_PFC_STATUS_FEEDBACK_SINGLE: Feedback control (single output) operation in progress

FAA_PFC_STATUS_FEEDBACK_INTERLEAVED: Feedback control (interleaved output) in operation

FAA_PFC_STATUS_DETECT_OVP: Overvoltage detected

FAA_PFC_STATUS_DETECT_OCP: Overcurrent detected

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function acquires status information for PFC converter control.

Special Notes:

None

8.2.13 R_{ConfigName}_DPConv_PFC_ChangeOutputMode

This function changes the PFC output mode.

Format

```
void R_{ConfigName}_DPConv_PFC_ChangeOutputMode  
      (e_faa_pfc_output_t mode, bool is_fade)
```

Parameters

mode: PFC output mode

 FAA_PFC_OUTPUT_SINGLE: Single output mode

 FAA_PFC_OUTPUT_INTERLEAVED: Interleaved output mode

is_fade: Whether to apply fade processing

 true: Fade enabled

 false: Fade disabled

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function switches the PFC output mode to the mode specified by the argument. When fade processing is required, it gradually adjusts the output level at the start and stop of the slave output to achieve a smooth transition.

For details on fade processing, refer to 6.2.3.1 Fade Processing.

Special Notes:

None

8.2.14 R_{ConfigName}_DPConv_PFC_SetMaxFreqLimit

This function sets the maximum frequency for the PFC output.

Format

```
void R_{ConfigName}_DPConv_PFC_SetMaxFreqLimit (uint8_t freq_10khz)
```

Parameters

freq_10khz: Maximum frequency [10kHz]

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function sets the maximum frequency for the PFC output. Specify the desired maximum frequency in 10kHz increments as the argument. (Example: 200kHz → Set value 20)

Special Notes:

This function is optional. Call it according to user application requirements.

8.2.15 R_{ConfigName}_DPConv_PFC_GetMaxFreqLimit

This function acquires the maximum frequency setting value for the PFC output.

Format

```
uint8_t R_{ConfigName}_DPConv_PFC_GetMaxFreqLimit (void)
```

Parameters

None

Return Values

Maximum frequency [10kHz]

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function acquires the maximum frequency setting value for the PFC output. The acquired value represents the maximum frequency (unit: 10 kHz).

Special Notes:

This function is optional. Call it according to user application requirements.

8.2.16 R_{ConfigName}_DPConv_PFC_SetFz

This function sets the zero-point frequency (fZ) parameter in PFC feedback control.

Format

```
void R_{ConfigName}_DPConv_PFC_SetFz (uint16_t fz_hz)
```

Parameters

fz_hz: Zero-point frequency [Hz]

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function dynamically sets the zero-point frequency (fZ) parameter in PFC feedback control. The set parameter will be reflected in the next PFC feedback control processing.

Special Notes:

This function is optional. Call it according to user application requirements.

8.2.17 R_{ConfigName}_DPConv_PFC_SetKp

This function sets the proportional constant (KP) parameter in PFC feedback control.

Format

```
void R_{ConfigName}_DPConv_PFC_SetKp (uint16_t kp)
```

Parameters

KP: Proportional Constant (KP)

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function dynamically sets the proportional constant (KP) parameter in PFC feedback control. Enter the desired parameter value multiplied by 4096 as the argument. The set parameter will be reflected during the next PFC feedback control process.

Special Notes:

This function is optional. Call it according to user application requirements.

8.2.18 R_{ConfigName}_DPConv_PFC_SetAdjustParamForInterleaved

This function sets the parameter values in the output energy adjustment equation.

Format

```
void R_{ConfigName}_DPConv_PFC_SetAdjustParamForInterleaved  
    (double slope, double intercept)
```

Parameters

slope: Slope in the linear adjustment equation

intercept: Intercept in the linear adjustment equation

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function sets the parameter values to adjust for fluctuations in output energy that occur when switching between single-mode and interleaved-mode operation. For details on specifications, refer to 6.2.3.2 Output Energy Adjustment.

Special Notes:

This function is optional. Call it according to user application requirements.

In addition, when using this function, call it before starting PFC feedback control.

8.2.19 R_{ConfigName}_DPConv_PFC_GetOnWidthCount

This function gets the count value indicating the pulse on-width at the PFC master output.

Format

```
uint16_t R_{ConfigName}_DPConv_PFC_GetOnWidthCount (void)
```

Parameters

None

Return Values

Pulse on-width count value

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function gets the pulse on-width count value at the PFC master output.

Note that since the supply clock for the 16-bit timer KB implementing PFC operation is 96MHz, the time per count is 10.416[ns].

Special Notes:

This function is optional. Call it according to user application requirements.

8.2.20 R_{ConfigName}_DPConv_LLCn_Init

This function initializes the LLCn converter control function. (n = 1, 2)

Format

```
void R_{ConfigName}_DPConv_LLCn_Init (faa_llc_callback_t * p_callback)
```

Parameters

p_callback: Pointer to callback function

p_detect_ocp: Function pointer for registering callback function upon LLCn overcurrent detection

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function initializes parameters and registers user callback functions for the LLCn converter control function.

Special Notes:

If a user callback is not required upon detecting OCP, set the p_detect_ocp member variable of the faa_llc_callback_t type passed as an argument to NULL.

8.2.21 R_{ConfigName}_DPConv_LLCn_StartFeedbackControl

This function starts LLCn feedback control. (n = 1, 2)

Format

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

Parameters

None

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function starts LLCn feedback control. At this time, OCP detection becomes active.

Special Notes:

This function is available only in High-Speed Mode.

8.2.22 R_{ConfigName}_DPConv_LLCn_Stop

This function stops the LLCn converter control. (n = 1, 2)

Format

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

Parameters

None

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function stops the LLCn converter control.

Special Notes:

None

8.2.23 R_{ConfigName}_DPConv_LLCn_EnableOCP

This function enables the overcurrent protection function of the LLCn converter. (n = 1, 2)

Format

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

Parameters

None

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function enables the overcurrent protection function of the LLCn converter.

Special Notes:

This function is optional. Call it only if required for the desired user behavior.

Depending on user application requirements, OCP detection can be started at any time during Low-Speed Mode.

8.2.24 R_{ConfigName}_DPConv_LLCn_DisableOCP

This function disables the overcurrent protection function of the LLCn converter. (n = 1, 2)

Format

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

Parameters

None

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function disables the overcurrent protection function of the LLCn converter.

Special Notes:

None

8.2.25 R_{ConfigName}_DPConv_LLCn_GetStatus

This function gets status information for the LLCn converter control. (n = 1, 2)

Format

```
e_faa_llc_status_t R_{ConfigName}_DPConv_LLCn_GetStatus (void)
```

Parameters

None

Return Values

LLCn Status Information

FAA_LL_C_STATUS_STOP: Stopped

FAA_LL_C_STATUS_FEEDBACK: Feedback control in operation

FAA_LL_C_STATUS_DETECT_OCP: Overcurrent detected

FAA_LL_C_STATUS_DETECT_ERROR: Undefined error detected

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function gets status information for the LLCn converter control.

Special Notes:

None

8.2.26 R_{ConfigName}_DPConv_GetAdValue

This function performs an A/D conversion and gets the average value of the conversion result.

Format

```
uint16_t R_{ConfigName}_DPConv_GetAdValue  
        (e_faa_ad_channel_t channel, uint8_t number_of_times)
```

Parameters

channel: ANI Channel

number_of_times: Number of conversions

Return Values

A/D average value

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function performs A/D conversion a specified number of times on the ANI channel specified by the argument and gets the average value.

Special Notes:

This function is only available in Low-Speed Mode.

The A/D conversion settings when using this function are as follows:

- Input clock frequency (ADM1 register, ADLSP bit): 4 MHz or lower
- One-shot select mode
- Software trigger (triggered within this function)
- Conversion of only the specified channel in Advanced Mode (Standard 1)
- 12-bit resolution
- A/D conversion time: $43/f_{CLK}$ (When f_{CLK} is 4MHz in High Speed Mode, $10.75\mu s/ch$)
- Sampling clock count: $27f_{AD}$ ($f_{AD}=f_{CLK}$)

8.2.27 R_{ConfigName}_DPConv_RequestAdValue

This function requests an A/D conversion for the specified analog channel.

Format

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

Parameters

channel: ANI Channel

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function requests an A/D conversion for the specified analog channel.

Special Notes:

This function is only available in High-Speed Mode.

The A/D conversion settings when using this function are as follows:

- Input clock frequency (ADM1 register, ADLSP bit): over 4 MHz
- One-shot select mode
- Hardware trigger (triggered at 12.5us intervals by TAU01 in this library's CPU program)
- 4-channel conversion in Advanced Mode (default 1)
- 12-bit resolution
- A/D conversion time: $43/f_{CLK}$ (Since f_{CLK} in High-Speed Mode is 48MHz, this is 0.8958μs/channel)
- Number of sampling clock cycles: $27f_{AD}$ ($f_{AD}=f_{CLK}$)

8.2.28 R_{ConfigName}_DPConv_GetRequestedAdValue

This function gets the A/D conversion result.

Format

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

Parameters

buffer: Pointer to the buffer storing the AD conversion result

* Valid only when this function's return value is FAA_ADC_CONVERT_OK.

Return Values

A/D Conversion Status

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function gets the result of the requested A/D conversion.

Special Notes:

This function is only available in High-Speed Mode.

Before calling this function, request the A/D conversion using the R_{ConfigName}_DPConv_RequestAdValue function.

8.2.29 R_{ConfigName}_DPConv_RequestData

This function requests the readout of specified data.

Format

```
void R_{ConfigName}_DPConv_RequestData (e_faa_request_id_t id, uint16_t addr)
```

Parameters

id: Request ID

FAA_REQUEST_ID_NONE: No request

FAA_REQUEST_ID_PFC_V_FB: A/D value of the ANI used for the PFC feedback input^{Note1}

FAA_REQUEST_ID_LLC1_V_FB: A/D value of the ANI used for LLC1 feedback input^{Note1}

FAA_REQUEST_ID_LLC2_V_FB: A/D value of the ANI used for LLC2 feedback input^{Note1}

FAA_REQUEST_ID_SPECIFIED_ADDR: Specify a particular FAA data memory address

addr: Request FAA data memory address^{Note2}

The address of specific data within the FAA address map's data memory range (000H to 7FFH)

Note1 Values corresponding to the type of input value used for feedback can be gotten.

Note2 This setting is only effective when the request ID is FAA_REQUEST_SPECIFIED_ADDR.

Return Values

None

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function requests the readout of the data specified by the arguments.

Special Notes:

This function is only available in High-Speed Mode.

8.2.30 R_{ConfigName}_DPConv_GetRequestedData

This function gets the requested read data.

Format

```
bool R_{ConfigName}_DPConv_GetRequestedData (uint32_t * p_data)
```

Parameters

p_data: Destination for read data

*This parameter is valid only when this function returns true (read completed).

Return Values

Get result

true : Read completed

false : Reading in progress

Properties

Prototype declared in {ConfigName}_DPConv.h.

Description

This function gets the requested read data.

Special Notes:

This function is only available in High-Speed Mode.

9. Sample Program

This section describes sample programs using the digital power converter control library.

9.1 Hardware Configuration

For detailed information on the target board and hardware, refer to Chapter 2 of the following documentation.

Interleaved CrM PFC and LLC Control with RL78/G24 (Hardware & Software Basics) (R01AN8175)

9.2 Operational Confirmation Conditions

The sample program is confirmed to operate in the following environment.

Table 9.1 Operation confirmation environments

Item		Description
Microcontroller used		RL78/G24 (R7F101GFGxFP)
Board Used		RL78/G24 Interleaved PFC + LLC Board 400W Kit (RTK0EL0006D00000BJ)
Operating Frequency		High-Speed On-Chip Oscillator Clock (fHOCO): 8 MHz PLL Clock (fPLL): 96 MHz CPU/Peripheral Hardware Clock (fCLK): 48 MHz
Operating Voltage		AC 100V, 230V
CS+ for CC	Integrated Development Environment	Renesas Electronics CS+ for CC V8.14.00
	Compiler	CC-RL V1.15.01
IAR	Integrated Development Environment	IAR Systems IAR Embedded Workbench for Renesas RL78 V5.10.3
	Compiler	IAR C/C++ Compiler for Renesas RL78 V5.10.3.2716
e2 studio	Integrated Development Environment	Renesas Electronics e2 studio 2025-10
	Compiler	CC-RL V1.15.01
Smart Configurator (SC)		V1.15.0 [Components Used] Board Support Packages V1.91 UART Communication V1.10.0 Interval Timer V1.8.0 Flexible Application Accelerator V1.3.0 Ports V1.8.0 Input Signal High/Low Width Measurement V1.7.0 Interrupt Controller V1.7.0 Watchdog Timer V1.7.0 Voltage Detector V1.6.1

9.3 Software Specifications

9.3.1 Software Control

For an overview of the software control for the sample program, refer to Chapter 2 of the following document.

Interleaved CrM PFC and LLC Control with RL78/G24 (Hardware & Software Basics) (R01AN8175)

9.3.2 Software Configuration

This section describes the software configuration of the sample program.

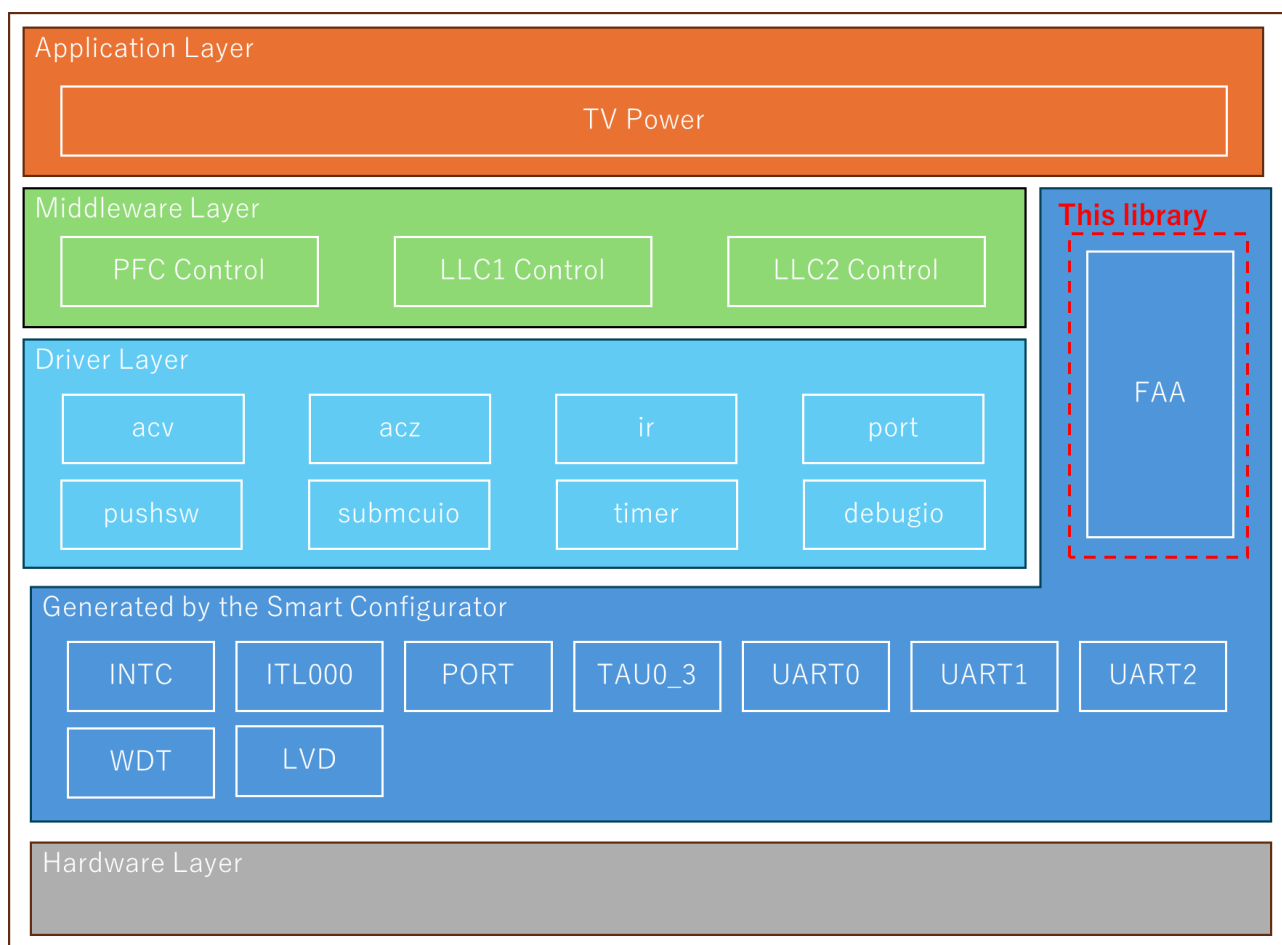


Figure 9.1 Software Configuration Diagram

9.3.3 Folder Structure

The folder structure of the sample program is described below.

Table 9.2 Folder Structure

Folder, File Name	Description
TvPower FAA	Sample program folder
main.c	Main file
src	Source code storage folder
r_common.h	Common definition file
application	Application program storage folder
r_tpower.c	TV power supply application source file
r_tpower.h	TV power supply application header file
middleware	Middleware program storage folder
r_pfc.c	PFC control middleware source file
r_pfc.h	PFC control middleware header file
r_llc1.c	LLC1 control middleware source file
r_llc1.h	LLC1 control middleware header file
r_llc2.c	LLC2 control middleware source file
r_llc2.h	LLC2 control middleware header file
driver	Driver program storage folder
r_acv.c	AC voltage detection driver source file
r_acv.h	AC voltage detection driver header file
r_acz.c	AC zero-cross detection driver source file
r_acz.h	AC zero-cross detection driver header file
r_debugio.c	Debug IO driver source file
r_debugio.h	Debug IO driver header file
r_ir.c	IR driver source file
r_ir.h	IR driver header file
r_port.c	Port driver source file
r_port.h	Port driver header file
r_pushsw.c	PushSW driver source file
r_pushsw.h	PushSW driver header file
r_submcuio.c	Sub MCU IO driver source file
r_submcuio.h	Sub MCU IO driver header file
r_timer.c	Timer driver source file
r_timer.h	Timer driver header file
smc_gen	Smart Configurator generation folder (omitted below) * The code for this library will be generated under this directory.

9.3.4 Option Byte Settings

Table 4.2 shows the option byte settings.

Table 9.3 Option Byte Settings

Address	Setting Value	Description
000C0H/040C0H	1111 0010 (F2H)	Watchdog timer operation permitted <ul style="list-style-type: none"> ▪ Overflow time: 2⁸/fIL (7.8125 ms) ▪ After reset is released, counting starts ▪ Counter operation stopped in the HALT or STOP mode
000C1H/040C1H	0011 1010 (3AH)	LVD reset mode (falling: 2.91 V, rising: 2.97 V)
000C2H/040C2H	1110 1010 (EAH)	Flash operation mode: High-speed main mode High-speed on-chip oscillator frequency: 8 MHz
000C3H/040C3H	1000 0100 (84H)	On-chip debug operation enabled

9.3.5 Configuration Settings

The configuration settings for the digital power converter control library in this sample program are as follows.

Table 9.4 Smart Configurator Settings (Basic settings)

property	value	description
Number of PFC converter channels	1	Select the number of control channels for the PFC converter.
Number of LLC converter channels	2	Select the number of control channels for the LLC converter.
ADC VREF(+)	AVREFP	Select the + side reference voltage for the A/D converter.
ADC VREF(-)	AVREFM	Select the - side reference voltage for the A/D converter.
Low speed clock frequency at MOCO	4MHz	Select the operating clock for Low-Speed Mode.
Clock source in high speed mode	HOCO	Select the clock source for High-Speed Mode.

Table 9.5 Smart Configurator Settings (PFC settings) (1/2)

property	value	description	
ANI pin for feedback input	ANI4	Select the ANI pin to use for feedback input.	
Type of input value	Moving average	Select the input value type.	
Target input value	2735	Specify the target value for PI feedback control.	
Dynamic-OVP detecting threshold	2826	Specify the Dynamic-OVP threshold. (Protection function that temporarily halts output while the output voltage exceeds the threshold)	
OVP detecting threshold	3022	Specify the input threshold for detecting OVP. (Protection function that permanently halts output when the output voltage exceeds the threshold)	
A/D correction value	58	Specify the correction value for the A/D get value. Correction is required when the output impedance of the signal input to the target analog pin is high.	
TKBO pin	TKBO10/TKBO11	Select the TKBO pin for output.	
Active level of TKBO	High level	Select the active level of TKBO.	
Master side detect edge of ZCD pin	Fall edge	Select the detection edge (polarity) for the master-side ZCD pin.	
Slave side detect edge of ZCD pin	Fall edge	Select the detection edge (polarity) for the slave-side ZCD pin.	
Longest period [us]	20	Specify the longest period for the output pulse. (Output period setting when ZCD is not detected)	
Minimum on-time [us]	0.5	Specify the minimum on-time of the output pulse.	
Maximum on-time of each output pulse [us]	15.625	Specify the maximum on-time of the output pulse.	
PI parameters	Zero point frequency [Hz]	2	Specify the zero point frequency for PI feedback control.
	Feedback period [us]	400	Select the PI feedback control period.
	Proportional Gain (Kp)	0.25	Specify the proportional gain (Kp) for PI feedback control.
Comparator combination of PFC OCP (Over Current Protection)	IVCMP1(M)+IVCMP3(S)	Select the comparator input pin used for OCP (Over Current Protection). Select Disable if OCP is not applied.	

Table 9.6 Smart Configurator Settings (PFC settings) (2/2)

property	value	description
OCP threshold voltage ratio to VDD	0.48	<p>Specify the OCP threshold (voltage equivalent value) as a ratio relative to the VDD voltage value.</p> <p>e.g. When VDD = 5V and OCP threshold = 2.4V, the parameter value is calculated as follows: $2.4V / 5V = 0.48$</p>

Table 9.7 Smart Configurator Settings (LLC1 setting)

property	Value	description	
ANI pin for feedback input	ANI5	Select the ANI pin to use for feedback input.	
Type of input value	Binary total	Select the type of input value.	
Target input value	-	Specify the target value for PI feedback control.	
Active level at over target voltage	Low level	Select the optocoupler output level when the input is over the target value. * Only valid when input type is Binary total.	
TKBO pin	TKBO00/TKBO01	Select the TKBO pin for output.	
Active level of TKBO	High level	Select the active level for TKBO.	
Deadtime [ns]	300	Specify the dead time [ns].	
Soft start parameters	Frequency at the start [kHz]	500	Specify the frequency [kHz] at soft start initiation.
	Frequency at the end [kHz]	120	Specify the frequency [kHz] at soft start termination.
	Max duration [ms]	100	Specify the maximum soft start duration [ms].
	Analog input value for terminate	9	Specify the soft start terminate threshold (analog value).
PI parameters	Frequency at the start [kHz]	100	Specify the start frequency for PI feedback.
	Lowest frequency [kHz]	65	Specify the lowest frequency [kHz].
	Highest frequency [kHz]	300	Specify the highest frequency [kHz].
	Zero point frequency [Hz]	1500	Specify the zero point frequency for PI feedback control.
	Feedback period [us]	200	Select the PI feedback control cycle.
	Proportional Gain (Kp)	0.015625	Specify the proportional gain (Kp) for PI feedback control.
OCP(Over Current Protection)	IVCMP0	Select the comparator input pin used for OCP (Over Current Protection). Select Disable if OCP is not applied.	
OCP threshold voltage ratio to VDD	0.84	Specify the OCP threshold (voltage equivalent value) as a ratio relative to the VDD voltage value. e.g. When VDD = 5V and OCP threshold = 4V, the parameter value is calculated as follows: $4V / 5V = 0.8$	

Table 9.8 Smart Configurator Settings (LLC2 setting)

property	Value	description
ANI pin for feedback input	ANI6	Select the ANI pin to use for feedback input.
Type of input value	Binary total	Select the type of input value.
Target input value	-	Specify the target value for PI feedback control.
Active level at over target voltage	Low level	Select the optocoupler output level when the input is over the target value. * Only valid when input type is Binary total.
TKBO pin	TKBO20/TKBO21	Select the TKBO pin for output.
Active level of TKBO	High level	Select the active level for TKBO.
Deadtime [ns]	300	Specify the dead time [ns].
Soft start parameters	Frequency at the start [kHz]	500 Specify the frequency [kHz] at soft start initiation.
	Frequency at the end [kHz]	120 Specify the frequency [kHz] at soft start termination.
	Max duration [ms]	100 Specify the maximum soft start duration [ms].
	Analog input value for terminate	9 Specify the soft start terminate threshold (analog value).
PI parameters	Frequency at the start [kHz]	100 Specify the start frequency for PI feedback.
	Lowest frequency [kHz]	65 Specify the lowest frequency [kHz].
	Highest frequency [kHz]	300 Specify the highest frequency [kHz].
	Zero point frequency [Hz]	1250 Specify the zero point frequency for PI feedback control.
	Feedback period [us]	200 Select the PI feedback control cycle.
	Proportional Gain (Kp)	0.059375 Specify the proportional gain (Kp) for PI feedback control.
OCP(Over Current Protection)	IVCMP2	Select the comparator input pin used for OCP (Over Current Protection). Select Disable if OCP is not applied.
OCP threshold voltage ratio to VDD	0.72	Specify the OCP threshold (voltage equivalent value) as a ratio relative to the VDD voltage value. e.g. When VDD = 5V and OCP threshold = 4V, the parameter value is calculated as follows: $4V / 5V = 0.8$

Table 9.9 Smart Configurator Settings (User setting)

property	Value	description
ANI0	-	Select whether to use analog input (ANI) in the user application.
ANI1	-	
ANI2	Used	
ANI3	Used	
ANI4	-	
ANI5	-	
ANI6	-	
ANI7	Used	
ANI16	Unused	
ANI17	Unused	
ANI18	-	
ANI19	-	
ANI20	Unused	
ANI21	Unused	
ANI22	-	
ANI23	-	
ANI24	-	
ANI25	-	
ANI26	-	
ANI27	-	
ANI28	Unused	
ANI29	-	
ANI30	-	

9.3.6 How to build the environment

This sample program provides projects that run on two types of compilers: CC-RL and IAR. The following describes how to set up the environment for each compiler.

9.3.6.1 CC-RL compiler (CS+ for CC)

The following describes how to set up the CC-RL compiler (CS+ for CC) environment.

(1) Download and install the CS+ for CC

To run the sample application, you will need the CS+ for CC: integrated development environment.

Please download and install it from the Renesas Electronics website.

(2) Launch the sample application

1. Extract the provided project files and place them in any location (folder).
2. Double-click TvPower_FAA.mtpj inside the extracted project folder.

(3) How to build the sample application

1. From the CS+ for CC menu bar, click "Build Project" in "Build".
2. The build will start, and the build status will appear in the "All Messages" section. Once the message "Build Complete" appears, the build is finished.

(4) How to debug the sample application

1. Connect the E2 Lite to the E2 Lite connector on the debug board mounted on the target board.
2. Open JP400 on the debug board.
3. Configure the system to supply 3.3V power from the E2 Lite to the debug board.
4. Apply AC input voltage between CN1 and CN2 on the target board.
5. From the CS+ for CC menu bar, click "Download to Debug Tool" in "Debug".
6. Download the program to the microcontroller.
7. Click "Run" in "Debug" to execute debugging of the sample application.

Note:

When using breakpoints, for safety reasons, please set "Stop timer peripheral emulation while stopped" to "Yes" in "RL78 E2 Lite Properties."

In addition, if the program stops at a breakpoint, reset the CPU before resuming execution.

9.3.6.2 CC-RL compiler (e2 studio)

The following describes how to set up the CC-RL compiler (e2 studio) environment.

(1) Download and install the e2 studio

To run the sample application, you will need the e2 studio: integrated development environment.

Please download and install it from the Renesas Electronics website.


(2) Import the sample application

1. Unzip the provided project files and place them in any location (folder).
2. Launch e2 studio.
3. From the e2 studio menu bar, select "Import" in "Files".
4. Select "Import Existing Project into Workspace" and click "Next".
5. Click "Browse" under "Select Root Directory" and choose the project's root directory you placed in step
6. Check the box next to TvPower_FAA displayed in the project list and click "Finish".

(3) How to build the sample application

1. Right-click the sample project imported from Project Explorer and click "Build Project."
2. The build will start, and its status will appear in the "Console." Once the message "Build Finished." is displayed, the build is complete.

(4) How to debug the sample application

1. Connect the E2 Lite to the E2 Lite connector on the debug board mounted on the target board.
2. Open JP400 on the debug board.
3. Configure the system to supply 3.3V power from the E2 Lite to the debug board.
4. Apply AC input voltage between CN1 and CN2 on the target board.
5. Click the  button in e2 studio.
6. Download the program to the microcontroller and perform debugging.

Note:

When using breakpoints, for safety reasons, please set "Debug Configuration" → "Debug Tool Settings" → "Stop timer group emulation while stopped" to "Yes", so that the timer outputs are stopped (Hi-z) while execution is halted at a breakpoint.

In addition, if the program stops at a breakpoint, do not resume execution from that point. Please reset the CPU and restart the program from the beginning.

9.3.6.3 IAR compiler

The following describes how to set up the IAR compiler environment.

(1) Download and install the IAR Embedded Workbench

To run the sample application, you will need the IAR Embedded Workbench for Renesas RL78: integrated development environment.

Please download and install it from the IAR Systems website.


(2) Launch the sample application

1. Launch IAR Embedded Workbench for Renesas RL78.
2. Select "New Workspace" in "Files" menu.
3. Select "Save Workspace As" in "Files" menu and place the workspace in a folder of your choice.
4. Extract the provided project files and place them in the created workspace (folder).
5. Select "Add Existing Project" in "Project" menu and choose the project file (EWP file) placed in step3.
6. Click "Finish" in "Files".

(3) How to build the sample application

1. Right-click the project in the workspace and select "Make"
2. The build will start, and the build status will appear in the "Console". Once the message "Total errors: 0" is displayed, the build is complete.

(4) How to debug the sample application

1. Connect the E2 Lite to the E2 Lite connector on the debug board mounted on the target board.
2. Open JP400 on the debug board.
3. Configure the system to supply 3.3V power from the E2 Lite to the debug board.
4. Apply AC input voltage between CN1 and CN2 on the target board.
5. Click the  button in IAR Embedded Workbench for Renesas RL78.
6. Download the program to the microcontroller and perform debugging.

Note:

For safety, when breakpoints are used, enable "A (Timer)" under "Peripheral Break" in the debugger "Hardware Settings" to ensure that timer outputs enter the Hi-Z state during break.

If the program stops at a breakpoint, do not continue execution from that point. Reset the CPU and execute the program from the beginning.

9.3.7 API Functions

The API functions for the middleware layer in this sample program are described here for reference.

Table 9.10 PFC Control (Middleware Layer) API Function List

Function Name	Description
R_PFC_Init	Initialize the PFC control function.
R_PFC_StartBoost	Start PFC boost control.
R_PFC_StartBurst	Start PFC burst control.
R_PFC_StartNormalSingle	Start PFC normal control. (Single output operation)
R_PFC_SwitchToInterleaved	Switch PFC output to interleave mode.
R_PFC_SwitchToSingle	Switch PFC output to single mode.
R_PFC_Stop	Stop PFC control.
R_PFC_TickMs	Notify the Tick timing in PFC control.
R_PFC_TaskNormal	Execute Task processing in PFC normal control.
R_PFC_GetStatus	Get PFC control status.
R_PFC_UpdateMaxFreqLimit	Update PFC maximum frequency limit setting.
R_PFC_GetLoadPower100V	Estimate and get load power. (AC100V)
R_PFC_GetLoadPower200V	Estimate and get load power. (AC200V)

Table 9.11 LLCn Control (Middleware Layer) API Function List (n = 1, 2)

Function Name	Description
R_LLCn_Init	Initialize the LLCn control function.
R_LLCn_StartBurst	Start LLCn burst control.
R_LLCn_StartNormal	Start LLCn normal control.
R_LLCn_Stop	Notify LLCn control.
R_LLCn_TickMs	Notify the Tick timing in LLCn control.
R_LLCn_GetStatus	Get LLCn control status.

10. Operation using “QE for Lighting & Power”

QE for Lighting & Power (V2.0.0 or later) enables end-to-end management of power control parameter configuration, simulation, evaluation, and code generation.

This chapter describes the sequence of operations for verifying and evaluating the power control functions of a target board using QE for Lighting & Power.

10.1 Downloading QE for Lighting & Power

Download and install it from the Renesas Electronics website.

10.2 Preparing a Workspace

Configure the workspace, which serves as the folder for selecting the target board and storing the application program.

10.2.1 Selecting a Board

1. Open “2. Preparing a Workspace” and select “Renesas Board”.
2. From the target project selection, choose “[R01AN8175EJ0100] Interleaved CrM PFC and LLC Control with RL78/G24 (Hardware & Software Basics)”.

10.2.2 Selecting a Workspace

From “Select a Workspace,” choose any folder to store the application program.

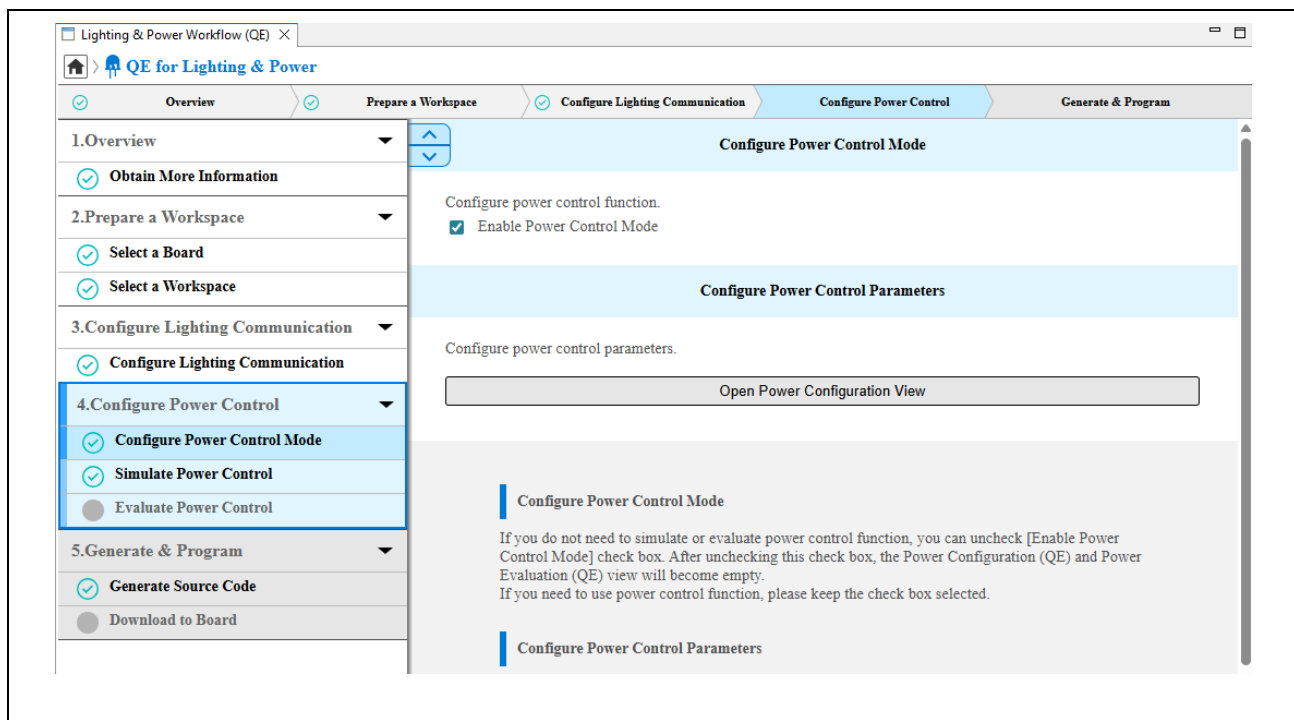


Figure 10.1 Preparing a Workspace

10.3 Configure Lighting Communication

This project does not support lighting communication, so “3. Configure Lighting Communication” cannot be used.

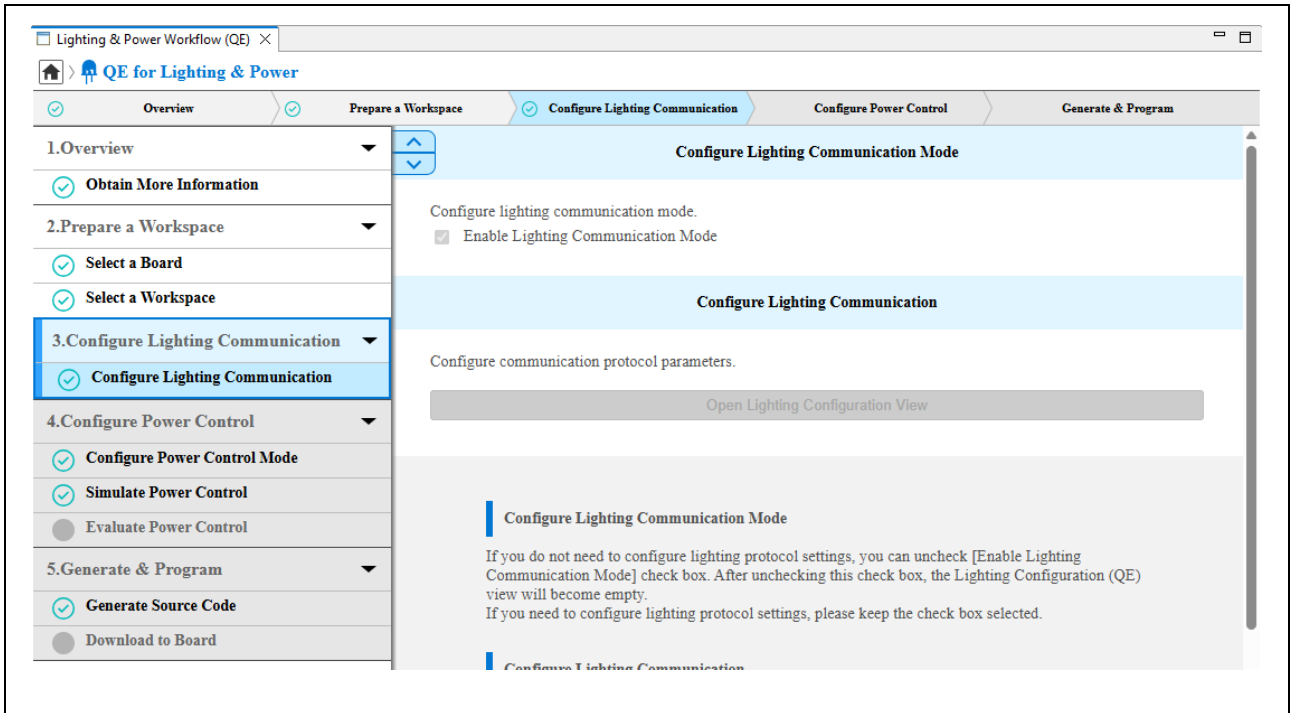


Figure 10.2 Configure Lighting Communication

10.4 Configure Power Control

“4. Configure Power Control” allows you to modify circuit characteristics and power control parameters using the Power Configuration view.

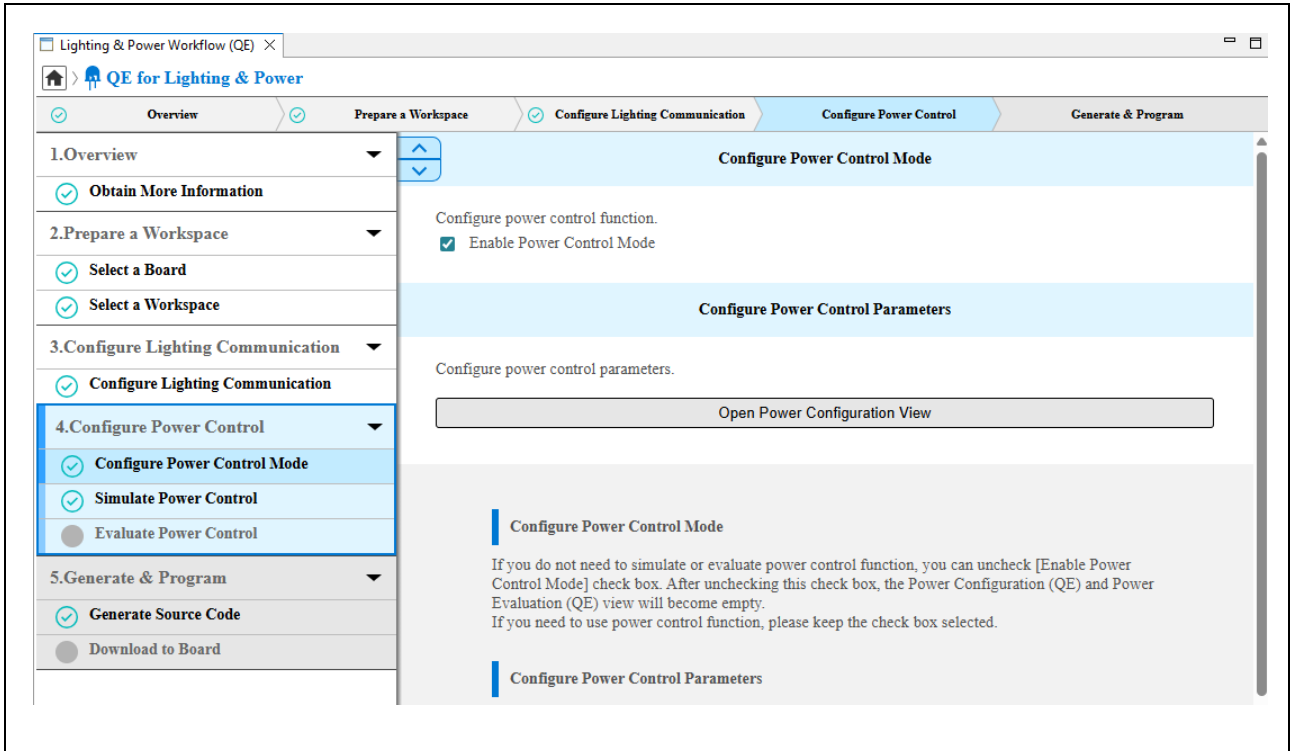


Figure 10.3 Configure Power Control

10.4.1 Configure Power Control Mode

Enabling “Configure Power Control Mode” makes power control features available, including setting power control parameters, simulation, and code generation.

10.4.1.1 Power Configuration View

Pressing the “Open Power Configuration View” button allows you to reference the schematic and modify circuit characteristics and power control parameters.

The Power Configuration View consists of the target board schematic (top) and configurable parameters (bottom).

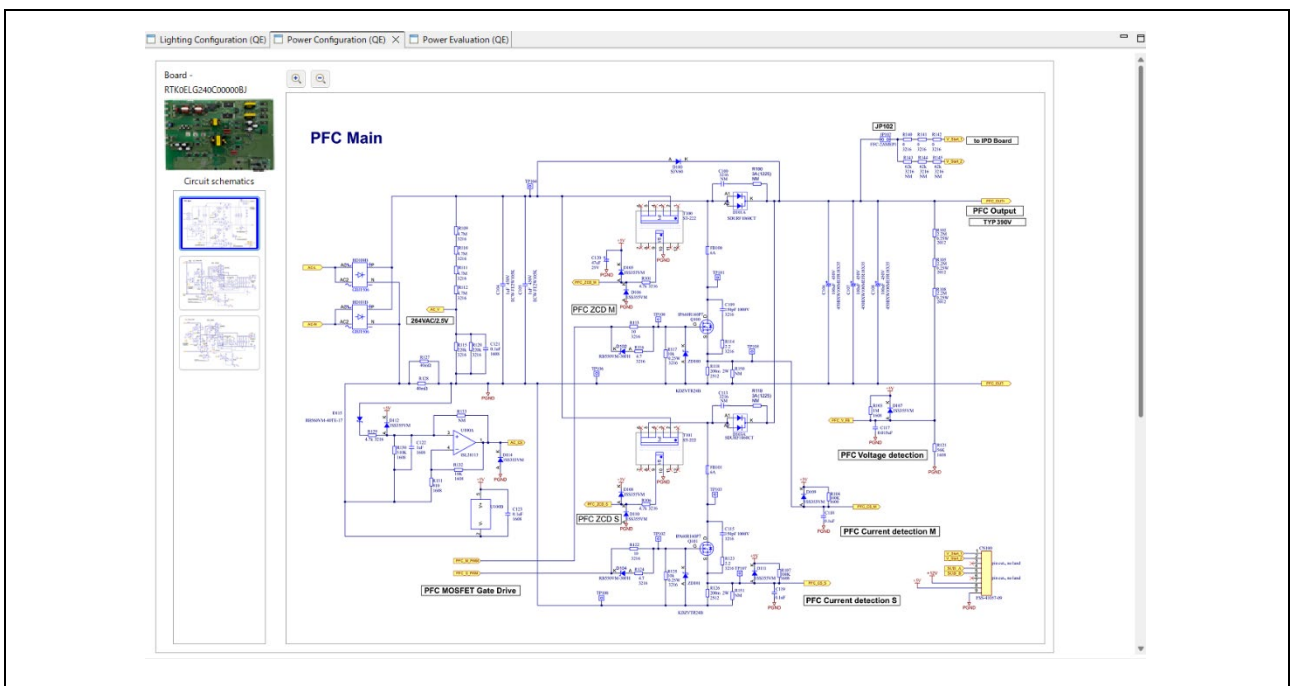


Figure 10.4 Power Configuration (the target board schematic)

Circuit Characteristics	Value	Power Control Parameters	Value
Common		CPU	
AC freq (Hz)	60	PFC	
Synchronous Rectification	<input type="checkbox"/>	DAC value for PFC OCP detection	122
PFC		PFC output voltage target	2735
PFC_M		PFC OVP detection threshold	3022
Power MOSFET		PFC Dynamic-OVP detection threshold	2826
Ron (Ω)	0.281	PFC feedback parameter A1 (only for simulative)	16425
Ids (A)	6.3	PFC feedback parameter A2 (only for simulative)	-16343
Vth (V)	3.5	LLC1	
Vgs (V)	10	DAC value for LLC1 OCP detection	860
Ciss (pF)	1317	LLC1 output voltage target	8
Coss (pF)	520	LLC1 feedback parameter A1	1989
Crss (pF)	80	LLC1 feedback parameter A2	-59
Inductor L1 (μH)	180	LLC2	
Inductor L2 (μH)	18	DAC value for LLC2 OCP detection	184
PFC_S		LLC2 output voltage target	8
Power MOSFET		LLC2 feedback parameter A1	6947
Ron (Ω)	0.281	LLC2 feedback parameter A2	-835
Ids (A)	6.3	FAA	
Vth (V)	3.5	PFC	

Figure 10.5 Power Configuration (configurable parameters)

Configurable parameters consist of “Circuit Characteristics” and “Power Control Parameters”.

Circuit Characteristics allows you to set parameters for circuit characteristics. The parameters set here will be reflected in the power control simulator described later.

Table 10.1 List of Circuit Characteristics (1/3)

Item		Default value	Description	
Common	AC freq (Hz)	60	AC voltage frequency	
	Synchronous Rectification	Not checked	Switch LLC Synchronous/Asynchronous rectification circuit.	
PFC_M	Power MOSFET	Ron (Ω)	0.281	PFC Master side MOSFET Ron [Ω]
		Ids (A)	6.3	PFC Master side MOSFET Ids [A]
		Vth (V)	3.5	PFC Master side MOSFET Vth [V]
		Vgs (V)	10	PFC Master side MOSFET Vgs [V]
		Ciss (pF)	1317	PFC Master side MOSFET Ciss [pF]
		Coss (pF)	520	PFC Master side MOSFET Coss [pF]
		Crss (pF)	80	PFC Master side MOSFET Crss [pF]
	Inductor L1 (μ H)	180	PFC Master side transformer primary inductance L1 [μ H]	
	Inductor L2 (μ H)	18	PFC transformer secondary inductance L2 [μ H]	
PFC_S	Power MOSFET	Ron (Ω)	0.5	PFC Slave side MOSFET Ron [Ω]
		Ids (A)	6.3	PFC Slave side MOSFET Ids [A]
		Vth (V)	3.5	PFC Slave side MOSFET Vth [V]
		Vgs (V)	1317	PFC Slave side MOSFET Vgs [V]
		Ciss (pF)	520	PFC Slave side MOSFET Ciss [pF]
		Coss (pF)	520	PFC Slave side MOSFET Coss [pF]
		Crss (pF)	80	PFC Slave side MOSFET Crss [pF]
	Inductor L1 (μ H)	180	PFC Slave side transformer primary inductance L1 [μ H]	
	Inductor L2 (μ H)	18	PFC Slave side transformer secondary inductance L2 [μ H]	

Table 10.2 List of Circuit Characteristics (2/3)

Item			Default value	Description
LLC1	Power MOSFET High Side	Ron (Ω)	0.237	LLC1 high side MOSFET Ron [Ω]
		Ids (A)	3.6	LLC1 high side MOSFET Ids [A]
		Vth (V)	4	LLC1 high side MOSFET Vth [V]
		Vgs (V)	10	LLC1 high side MOSFET Vgs [V]
		Ciss (pF)	807	LLC1 high side MOSFET Ciss [pF]
		Coss (pF)	249	LLC1 high side MOSFET Coss [pF]
		Crss (pF)	80	LLC1 high side MOSFET Crss [pF]
	Power MOSFET Low Side	Ron (Ω)	0.237	LLC1 low side MOSFET Ron [Ω]
		Ids (A)	3.6	LLC1 low side MOSFET Ids [A]
		Vth (V)	4	LLC1 low side MOSFET Vth [V]
		Vgs (V)	10	LLC1 low side MOSFET Vgs [V]
		Ciss (pF)	807	LLC1 low side MOSFET Ciss [pF]
		Coss (pF)	249	LLC1 low side MOSFET Coss [pF]
		Crss (pF)	80	LLC1 low side MOSFET Crss [pF]
	Inductor L1 (μ H)		525	LLC1 transformer primary inductance L1 [μ H]
	Inductor L2 (μ H)		2.5	LLC1 transformer secondary inductance L2 [μ H]
	Inductor L3 (μ H)		2.5	LLC1 L3 [μ H]
Inductor LR (μ H)		75	LLC1 Load Resistance [μ H]	
Capacitor CR (μ F)		0.044	LLC1 Resonant Capacitor Capacitance [μ F]	

Table 10.3 List of Circuit Characteristics (3/3)

Item			Default value	Description
LLC2	Power MOSFET High Side	Ron (Ω)	0.237	LLC2 high side MOSFET Ron [Ω]
		Ids (A)	3.6	LLC2 high side MOSFET Ids [A]
		Vth (V)	4	LLC2 high side MOSFET Vth [V]
		Vgs (V)	10	LLC2 high side MOSFET Vgs [V]
		Ciss (pF)	807	LLC2 high side MOSFET Ciss [pF]
		Coss (pF)	249	LLC2 high side MOSFET Coss [pF]
		Crss (pF)	80	LLC2 high side MOSFET Crss [pF]
	Power MOSFET Low Side	Ron (Ω)	0.237	LLC2 low side MOSFET Ron [Ω]
		Ids (A)	3.6	LLC2 low side MOSFET Ids [A]
		Vth (V)	4	LLC2 low side MOSFET Vth [V]
		Vgs (V)	10	LLC2 low side MOSFET Vgs [V]
		Ciss (pF)	807	LLC2 low side MOSFET Ciss [pF]
		Coss (pF)	249	LLC2 low side MOSFET Coss [pF]
		Crss (pF)	80	LLC2 low side MOSFET Crss [pF]
	Inductor L1 (μ H)		425	LLC2 transformer primary inductance L1 [μ H]
	Inductor L2 (μ H)		28	LLC2 transformer secondary inductance L2 [μ H]
	Inductor L3 (μ H)		28	LLC2 L3 [μ H]
Inductor LR (μ H)		75	LLC2 Load Resistance [μ H]	
Capacitor CR (μ F)		0.044	LLC2 Resonant Capacitor Capacitance [μ H]	

Power Control Parameters allow you to configure parameters related to power control. The parameters set here will be reflected in the power control simulator and generation program described later. This document lists the configuration items for the FAA version evaluation program below.

Table 10.4 List of Power Control Parameters (1/2)

Item	Default value	Description
PFC		
PFC output voltage target	2735	Set the target A/D value of PFC_V_FB pin. Calculate the temporary formula based on actual measurements $_N_DPConv_PFC_TARGET_VALUE = 6.5 * Voltage + 207.5$
PFC OVP detection threshold	3022	Set the threshold for PFC OVP detection (the detection object is the PFC_V_FB pin). Calculate the temporary formula based on actual measurements $_N_DPConv_PFC_OVP_THRESHOLD = 6.5 * Voltage + 207.5$
PFC Dynamic-OVP detection threshold	2826	Set the threshold for PFC Dynamic-OVP detection (the detection object is the PFC_V_FB pin). Calculate the temporary formula based on actual measurements $_N_DPConv_PFC_DOVP_THRESHOLD = 6.5 * Voltage + 207.5$
PFC output voltage target correction	58	The correction value of A/D value in PFC Output Voltage Target.
PFC feedback parameter A1 (only for simulation)	16425	Set the coefficients A1 of the PI control. $A1 = (\pi * fz * T + 1) * Kp$ And A1 needs to be multiplied by 65536 to facilitate calculations and improve the precision of the fractional part. e.g. $fz = 2Hz, T = 400us, Kp = 0.25$ $A1 = (\pi * 2 * 400 * 10^{-6} + 1) * 0.25 * 65536 = 16425$
PFC feedback parameter A2 (only for simulation)	-16433	Set the coefficients A2 of the PI control. $A2 = (\pi * fz * T - 1) * Kp$ And A2 needs to be multiplied by 65536 to facilitate calculations and improve the precision of the fractional part. e.g. $fz = 2Hz, T = 400us, Kp = 0.25$ $A2 = (\pi * 2 * 400 * 10^{-6} - 1) * 0.25 * 65536 = -16343$

Table 10.5 List of Power Control Parameters (2/2)

Item		Default value	Description
LLC1	LLC1 feedback parameter A1	1989	<p>Set the coefficients A1 of the PI control.</p> $A1 = (\pi * fz * T + 1) * Kp$ <p>And A1 needs to be multiplied by 65536 to facilitate calculations and improve the precision of the fractional part.</p> <p>e.g. $fz = 1500\text{Hz}$, $T = 200\mu\text{s}$, $Kp = 0.0156$</p> $_N_DPConv_LLC1_A1_PARAM = (\pi * 1500 * 200 * 10^{-6} + 1) * 0.0156 * 65536 = 1989$
	LLC1 feedback parameter A2	-59	<p>Set the coefficients A2 of the PI control.</p> $A2 = (\pi * fz * T - 1) * Kp$ <p>And A2 needs to be multiplied by 65536 to facilitate calculations and improve the precision of the fractional part.</p> <p>e.g. $fz = 1500\text{Hz}$, $T = 200\mu\text{s}$, $Kp = 0.0156$</p> $_N_DPConv_LLC1_A2_PARAM = (\pi * 1500 * 200 * 10^{-6} - 1) * 0.0156 * 65536 = -59$
LLC2	LLC2 feedback parameter A1	6947	<p>Set the coefficients A1 of the PI control.</p> $A1 = (\pi * fz * T + 1) * Kp$ <p>And A1 needs to be multiplied by 65536 to facilitate calculations and improve the precision of the fractional part.</p> <p>e.g. $fz = 1250\text{Hz}$, $T = 200\mu\text{s}$, $Kp = 0.05937$</p> $_N_DPConv_LLC2_A1_PARAM = (\pi * 1250 * 200 * 10^{-6} + 1) * 0.05937 * 65536 = 6947$
	LLC2 feedback parameter A2	-835	<p>Set the coefficients A2 of the PI control.</p> $A2 = (\pi * fz * T - 1) * Kp$ <p>And A2 needs to be multiplied by 65536 to facilitate calculations and improve the precision of the fractional part.</p> <p>e.g. $fz = 1250\text{Hz}$, $T = 200\mu\text{s}$, $Kp = 0.05937$</p> $_N_DPConv_LLC2_A2_PARAM = (\pi * 1250 * 200 * 10^{-6} - 1) * 0.05937 * 65536 = -835$

10.4.2 Simulate Power Control

The Power Simulator is a feature that simulates the target board's operation in software using parameters set in the Power Configuration view.

Using this simulator allows you to preview the time-varying voltage and current of PFC and LLC, as well as control behavior trends, via graphs and other visualizations before verifying on the actual hardware.

10.4.2.1 Power Simulation Dialog

Press the “Open Power Simulation Dialog” button to launch the Power Simulator. In the simulator, configure the following parameters and press the “Start Simulation” button to begin. The simulator visualizes the transition of each parameter value during simulation via graphs.

Table 10.6 List of Configurable Parameters

Item	Time [ms] (Default)	Default Value		Description
AC Voltage Gain	0.000	282.000 [V]		Input voltage. The configured value is applied after the specified time [ms] has elapsed.
LLC1 Load Resistance	0.000	100 [Ω]		LLC1 load resistance. The configured value is applied after the specified time [ms] has elapsed.
LLC2 Load Resistance	0.000	100 [Ω]		LLC2 load resistance. The configured value is applied after the specified time [ms] has elapsed.
Operating modes	0.000	SW1	SW2	Operation Mode Switch Basic Operation: After the set time [ms] elapses, the configured switch state takes effect. <ul style="list-style-type: none"> · Checked: Switch pressed · Unchecked: Switch not pressed The switch press duration determines short press/long press. Determination Conditions: Short press: 1999 ms or less Long press: 2000 ms or more Operation: SW1 Short Press: LLC2 Start/Stop SW1 Long Press: Maximum Frequency Limit Function ON/OFF SW2 Short Press: Standby/Normal Mode Switch SW2 Long Press: Standby/Normal Mode Switch
		unchecked	unchecked	
Simulation time [ms]	-	5		Simulation duration.

Table 10.7 Output Waveform List

Item	Description	
Common	Input AC Power	AC power [W]
PFC	PFC Output Current	PFC output current [A]
	PFC Output Voltage	PFC output voltage [V]
	PFC M PWM On Duty	PFC master-side PWM duty ratio [%]
	PFC S PWM On Duty	PFC slave-side PWM duty ratio [%]
	PFC PWM Frequency	PFC PWM operating frequency [Hz]
LLC1	LLC1 Output Current	LLC1 output current [A]
	LLC1 Output Voltage	LLC1 output voltage [V]
	LLC1 LC Resonant Current	LLC1 LC resonant current [A]
	LLC1 Secondary Current	LLC1 secondary-side current [A]
	LLC1 PWM Frequency	LLC1 PWM operating frequency [Hz]
LLC2	LLC2 Output Current	LLC2 output current [A]
	LLC2 Output Voltage	LLC2 output voltage [V]
	LLC2 LC Resonant Current	LLC2 LC resonant current [A]
	LLC2 Secondary Current	LLC2 secondary-side current [A]
	LLC2 PWM Frequency	LLC2 PWM operating frequency [Hz]

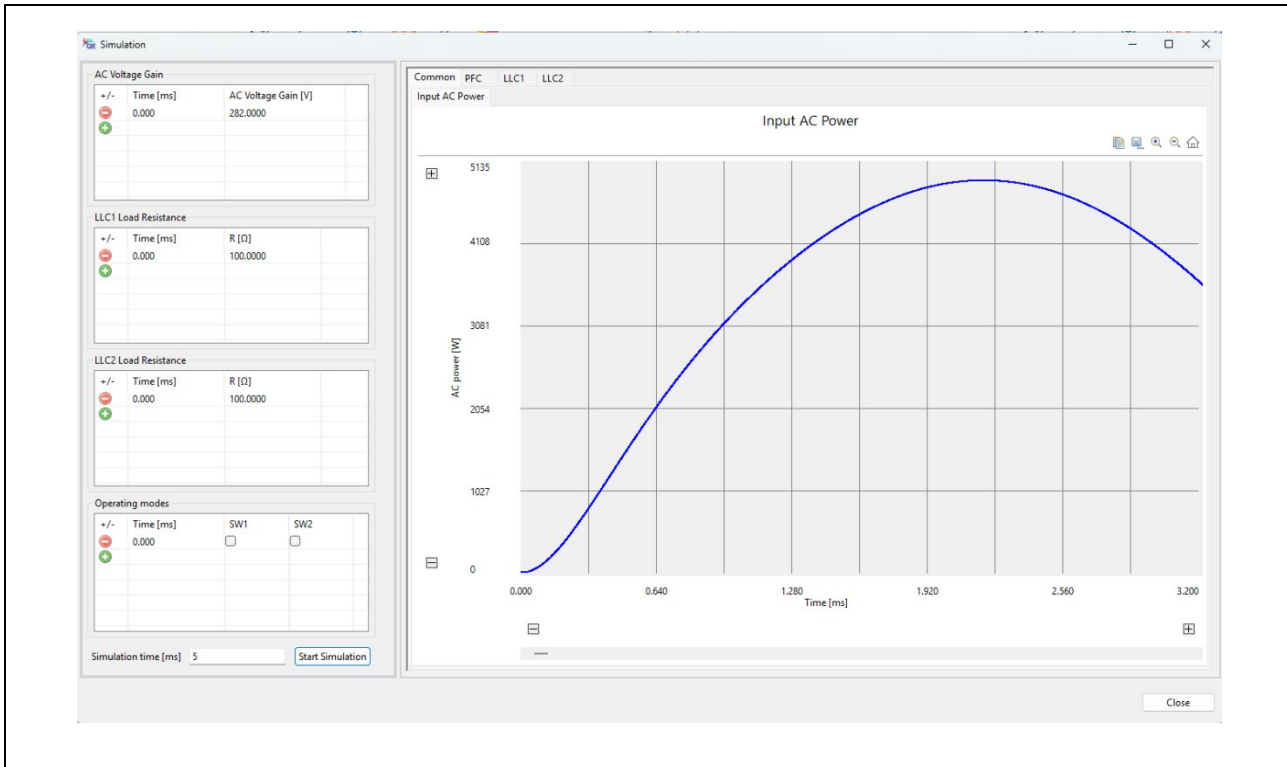


Figure 10.6 Power simulation dialog

Operation Procedure (Basic Operations)

1. Setting the simulation conditions

Configure the simulation conditions as needed, such as AC Voltage Gain, load resistance values, and operating-mode switching (SW operation).

Each parameter can have multiple conditions defined, and the specified value will be dynamically applied during the simulation once the configured time [ms] has elapsed.

2. Setting the simulation time

Enter the simulation duration in “Simulation time [ms]”.

3. Starting and stopping the simulation

Click “Start Simulation” to begin the simulation.

While the simulation is running, the button changes to “Stop Simulation”, which you can click to stop the simulation.

After stopping, you can click “Start Simulation” again to resume the simulation.

4. Checking Output Waveforms

During the simulation, you can observe waveform trends of the power-control operation, including time-varying voltages and currents of the PFC and LLC stages, as well as PWM duty ratios.

Operation Procedure (How to Switch to Normal Mode)

To simulate the transition from Standby mode to Normal mode, set “Simulation time [ms]” to 600 ms or longer, configure SW2 in “Operating modes” so that it performs a short-press operation at 600 ms or later, and then start the simulation.

After 600 ms has elapsed from the start of the simulation, the short-press operation of SW2 is applied, and the system switches from Standby mode to Normal mode.

The above procedure is an example based on the default conditions.

Depending on the circuit characteristics and power-control parameters modified in the Power Configuration view, adjust the timing values for “Simulation time [ms]” and “Operating modes” as appropriate.

Estimated Simulation Time

The computation performed on the PC depends on the simulation duration specified in “Simulation time [ms]”.

Although the actual execution time varies depending on the performance of the PC, the following can be used as a general guideline.

Table 10.8 Estimated Simulation Time

Simulation time [ms] (Setting)	Simulation Execution Time	Execution Environment
1000	Approx. 90 minutes	CPU: Intel Core i7-13700 Memory: 32 GB OS: Windows 11

Notes

Setting a large value for "Simulation time" may increase execution time and memory usage, which could lead to errors.

For details, refer to the release notes.

10.4.3 Evaluate Power Control

You can generate evaluation programs that incorporate parameters set in the Power Configuration view. Additionally, using “Variable Read/Write” and “Variable Monitoring,” you can observe changes in various parameter values in real time on the target board where the evaluation program is running.

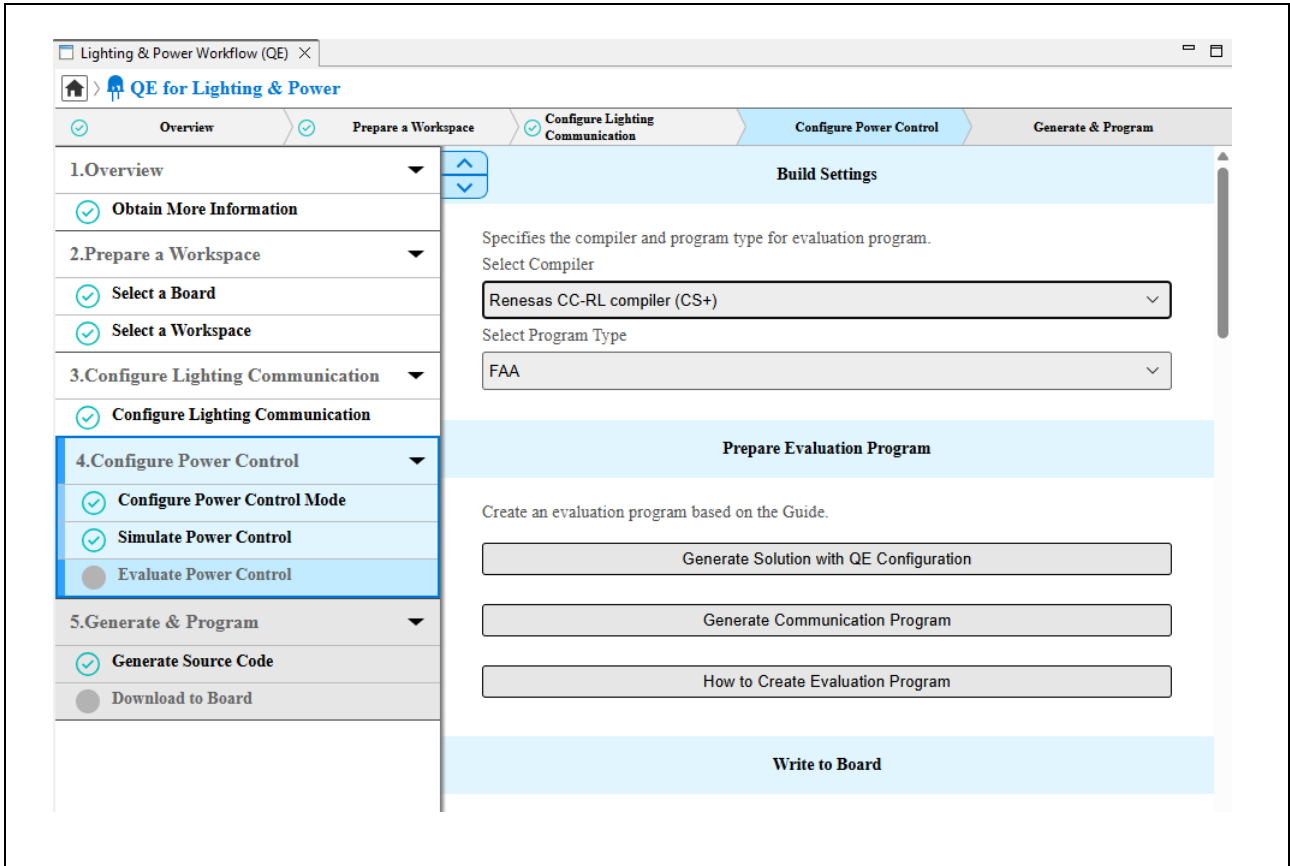


Figure 10.7 Evaluate Power Control

10.4.3.1 Build Settings

Configure the build settings for the evaluation project, incorporating parameters set in the Power Configuration view.

(1) Select Compiler

The generated project supports the following compilers:

- CC-RL Compiler (e² studio)
- CC-RL Compiler (CS+)
- IAR RL78 Compiler

(2) Select Program Type

You can also select either CPU or FAA as the core type to execute the power control processing.

This document assumes FAA is selected and explains the following accordingly.

10.4.3.2 Prepare Evaluation Program

Generate an evaluation program tailored to the selected compiler and program type.

(1) Generate Solution with QE Configuration

Clicking “Generate Solution with QE Configuration” creates an evaluation project in the following folder:

<workspace>\qeLighting_gen\tuning_program

(2) Generate Communication Program

Clicking the “Generate Communication Program” button adds the communication programs used for Variable Read/Write and Variable Monitoring to the project.

Note: After adding communication programs, integrate the generated application program according to the help documentation (click the “How to Create Evaluation Program” button).

10.4.3.3 Write to Board

Write the generated project to the target board.

Connect the E2 Lite to the E2 Lite connector on the debug board mounted on the target board, then click the “Write the Evaluation Program on Board” button.

10.4.3.4 Connect to Board

Use a USB cable to connect the target board to the host PC.

This connection is used for communication during power control evaluation via Variable Read/Write and Variable Monitoring.

Note: Before starting communicating, press SW2 manually to set the target board to Normal mode.

1. Connect the USB cable to the debug board.
2. Set the "Bitrate" to 500000.
3. Specify the COM Oort connected to the PC in "COM Port".
4. Press the "Connect" button.

10.4.3.5 Evaluate Power Control

Evaluate power control parameters on the target board in real time.

It possesses two evaluation functions.

(1) Variable Read/Write

Variable Read/Write enables real-time reading and monitoring of application program internal variables and register values.

All internal variables available for monitoring in this application note are read-only, and writing to these variables is not supported.

While writing to shared memory registers is technically possible, it may lead to unintended operation. Therefore, writing to shared memory registers is not recommended.

Pressing the "[Open Variable Read/Write View" button opens the [Variable Read/Write] tab in the [Power Evaluation (QE)] view

How to operate

1. Selecting Evaluation Variables

Select the power control variables to be evaluated from the variable selection list within the tab.

2. Reading Variable Values

Press the “Read” button to display the current variable values.

(2) Variable Monitoring

Variable Monitoring enables continuous acquisition of internal variables and register values from the evaluation program at fixed intervals, displaying their temporal changes as graphs.

Clicking the “[Open Variable Monitoring View]” button opens the [Variable Monitoring] tab in the [Power Evaluation (QE)] view.

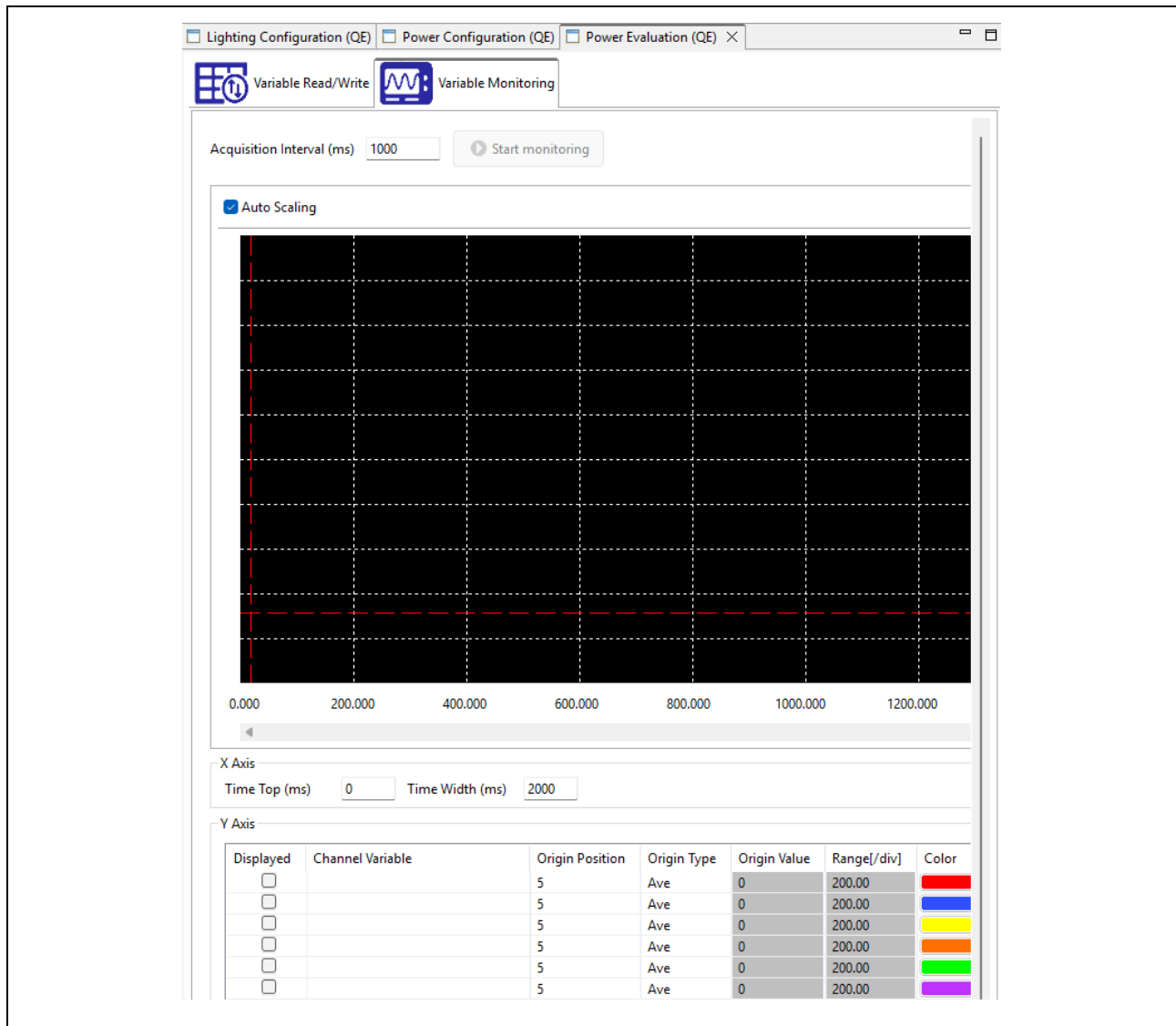


Figure 10.9 Variable Monitoring

How to operate

1. Selecting Monitoring Variables

Select the variable whose waveform you wish to display from the dropdown list (Channel) within the tab. Multiple variables can be selected simultaneously.

2. Configuring Acquisition Conditions

You can set the monitoring time and the range for the output waveform.

3. Starting Monitoring

Pressing “Start Monitoring” initiates variable value acquisition and waveform display.

10.5 Generate and Program

You can generate a project that reflects the parameters configured during power-control evaluation and write it to the target board.

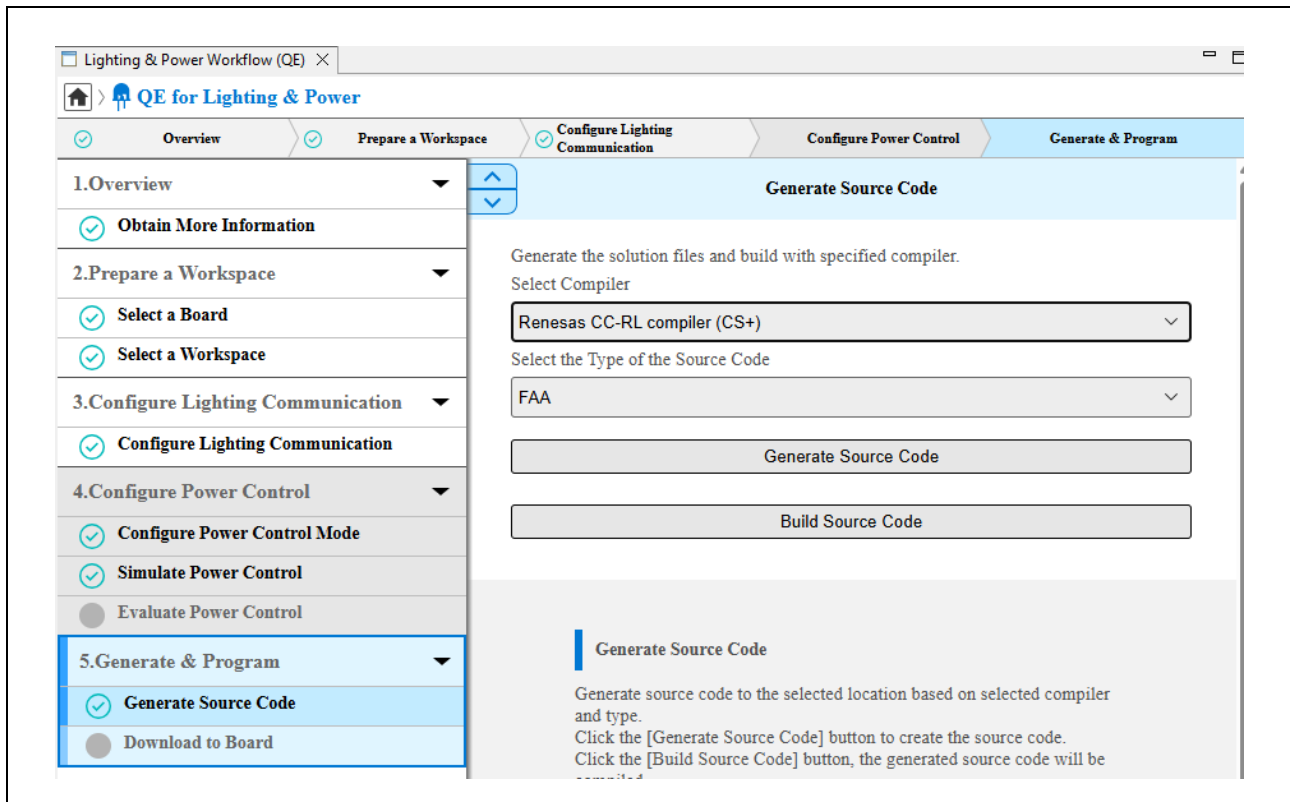


Figure 10.10 Generate and Program

10.5.1 Generate Source Code

Clicking the “Generate Source Code” button will create the project in the following folder.

“<workspace>%qeLighting_gen%solution”

10.5.2 Download to Board

Pressing the “Download to Board” button writes the project to the target board via E2 Lite.

10.6 Notes on PFC Control Parameters A1 and A2

The coefficients A1 and A2 for PFC PI control are calculated within the evaluation program based on specifications such as the zero-point frequency (fz) and proportional gain (Kp). Therefore, values for A1 and A2 calculated or set in the Power Configuration view or simulator will not be reflected in the generated application program.

To change these parameters, you must directly modify the values of the zero-point frequency (fz) and proportional gain (Kp) defined in the application program.

10.6.1 Calculation Method of Kp and fz

The zero-point frequency (fz) and proportional gain (Kp) can be calculated from the A1 and A2 values calculated and set in the Power Configuration view or simulator using the following formula.

However, since the values of A1 and A2 are multiplied by 65536 for application program convenience, divide the calculated values by 65536 before substituting them.

Also, the PFC feedback cycle T in the target program is fixed at 400us.

$$Kp = \frac{A1 - A2}{2}$$

$$fz = \frac{A1 + A2}{\pi T(A1 - A2)}$$

10.6.2 How to Reflect Kp and fz

The zero-point frequency (fz) and proportional gain (Kp) values calculated from the formula can be changed by directly editing the following macro values within the application program. Set each as an integer value.

Table 10.9 fz, Kp program modification points

Target file	Macro name	Default value	Note
Config_FAA_DPConv.c	PFC_DEFAULT_FZ	2	Unit is Hz.
	PFC_DEFAULT_KP	1024	Due to the application program's requirements, set the value multiplied by 4096. ex). When Kp is 0.25 0.25 * 4096 = 1024

11. Notes

The sample program operations provided in this document are just examples. The application layer must be modified and evaluated according to the intended use.

12. Reference Documents

RL78/G24 User's Manual: Hardware (R01UH0961)

RL78 Family User's Manual: Software (R01US0015)

Obtain the latest version of the manual from the website of Renesas Electronics.

Technical Update/Technical News

Obtain the latest information from the website of Renesas Electronics.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	2026.03.19	—	First edition

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/.