RL78/I1A

PFC 制御搭載 LED 照明制御

R01AN2138JJ0100
Rev.1.00
2014.07.11

要旨

本アプリケーション・ノートは、RL78/I1A マイクロコントローラの機能を使用して PFC 制御搭載 LED 照明システムを制御する方法について説明します。

動作確認デバイス

本書は、LED 照明システムおよび電源システムを設計し、開発するシステム・エンジニアを対象にしています。

対象製品は以下のとおりです。

- 20 ピン：R5F1076C
- 30 ピン：R5F107AE、R5F107AC
- 38 ピン：R5F107DE

目次

1. はじめに

本アプリケーション・ノートは、RL78/I1A マイクロコントローラを使用した LED 照明を制御するサンプル･プログラムについて説明しています。本サンプル･プログラムは定電流制御による 3 チャネルの LED の独立制御、スイッチを使用した個別調光制御、および PFC 出力電圧制御を行っています。また LED の定電流制御、PFC 出力電圧制御のフィードバック処理は、PI（比例積分）制御に基づいているものです。

本プログラムは、テセラテクノロジ製 RL78/I1A AC/DC LED 制御評価ボード（TPW-RL78I1A）を使用して評価することができます。RL78/I1A AC/DC LED 制御ボードの端子構成は回路図にて確認ください。

2. RL78/I1A を使用した LED 制御の概要

2.1 LED システム制御に関する RL78/I1A の特徴

RL78/I1A マイクロコントローラには、LED 照明システムを効率的に制御するための以下のような様々な機能が組み込まれています。

— RL78/I1A マイクロコントローラでは、16 ビット・タイマ KB0、KB1、KB2 と 16 ビット・タイマ KC0 のタイマ出力機能を使用して、最大 6 チャネルの LED 定電流制御と PWM 調光制御をすることができます。そのため、LED 定電流制御専用の外部 IC が不要になり、設計コストを削減することができます。16 ビット・タイマ KBn は強力な機能を有し、その中にはサンプル・プログラムで使用するディザリング機能があります。この機能を使用すると、平均 PWM 分解能を 0.98ns に上げることができます。

— また、RL78/I1A では、臨界導通モード（CRM）において、16 ビット・タイマ KBn と連動するコンパレータおよび外部割り込みによるタイマ・リスタート機能を使用して力率改善（PFC）制御が可能です。PFC 制御のための専用 IC も不要になるので、設計コストがさらに削減されます。

— RL78/I1A には、LED または PFC 制御回路内で過電流または過電圧が検出された場合に CPU を介在しないで PWM 出力を停止することができる保護機能も内蔵されています。この機能は、16 ビット・タイマ KBn と連動するコンパレータおよび外部割り込みをトリガとした強出力停止機能を使用することによって実現します。
さらに、緊急停止後の動作再開はソフトウェアで制御できるため、システム要件に応じてフレキシブルな保護機能が実現されます。

— Rl78/I1A マイクロコントローラは、DALI 通信機能をサポートするシリアル・アレイ・ユニット（UART4/DALI）を搭載し、DALI 通信規格に準拠したマンチェスタコードの送受信（8、16、17 または 24 ビット）をすることができます。
これにより、データ送信および受信時の CPU 負荷を軽減します。

— RL78/I1A は、UART0 シリアルインタフェースを使用した DMX512 通信もサポートしています。ここで、タイマ・アレイ・ユニットのチャネル 7 の入力信号パルス幅測定機能を使用して、RxD0 端子のブレーク時間の立ち下りエッジを検出しその幅（88μs 以上のロー・レベル）を測定することができます。また、16 ビットタイマ・アレイ・ユニットのインターバル・タイマ機能を使用して、
Mark After Break 信号の幅（8μs～1s のハイ・レベル）を計算し信号を受け付けることができます。
Mark Time BetweenSlots を計測することもできます。

— 赤外線（IR）リモート・コントロール信号受信に 16 ビット・タイマ・アレイ・ユニットのパルス間隔測定機能を使用することができます。これにより、データ受信時の CPU 負荷を軽減することができます。

【注】16 ビット・タイマ KC0 ゲート制御機能を使用した PWM 調光は、本アプリケーション・ノートで説明するサンプル・プログラムでは使用しません。タイマ KB0 および KB1 チャネルの PWM 値を調整することによる DC 調光のみを使用します。

2.2 システム・ブロック図

図 1 に、RL78/I1A AC/DC LED 制御評価ボード（TPW-RL78I1A）のシステム・ブロック図を示します。
本 LED 照明システムは、スイッチ入力に応じて，PFC 制御および LED×3ch 制御を行います。
RL78/I1A マイクロコントローラのみで完全制御を実現するため、追加の外部 IC が必要ありません。

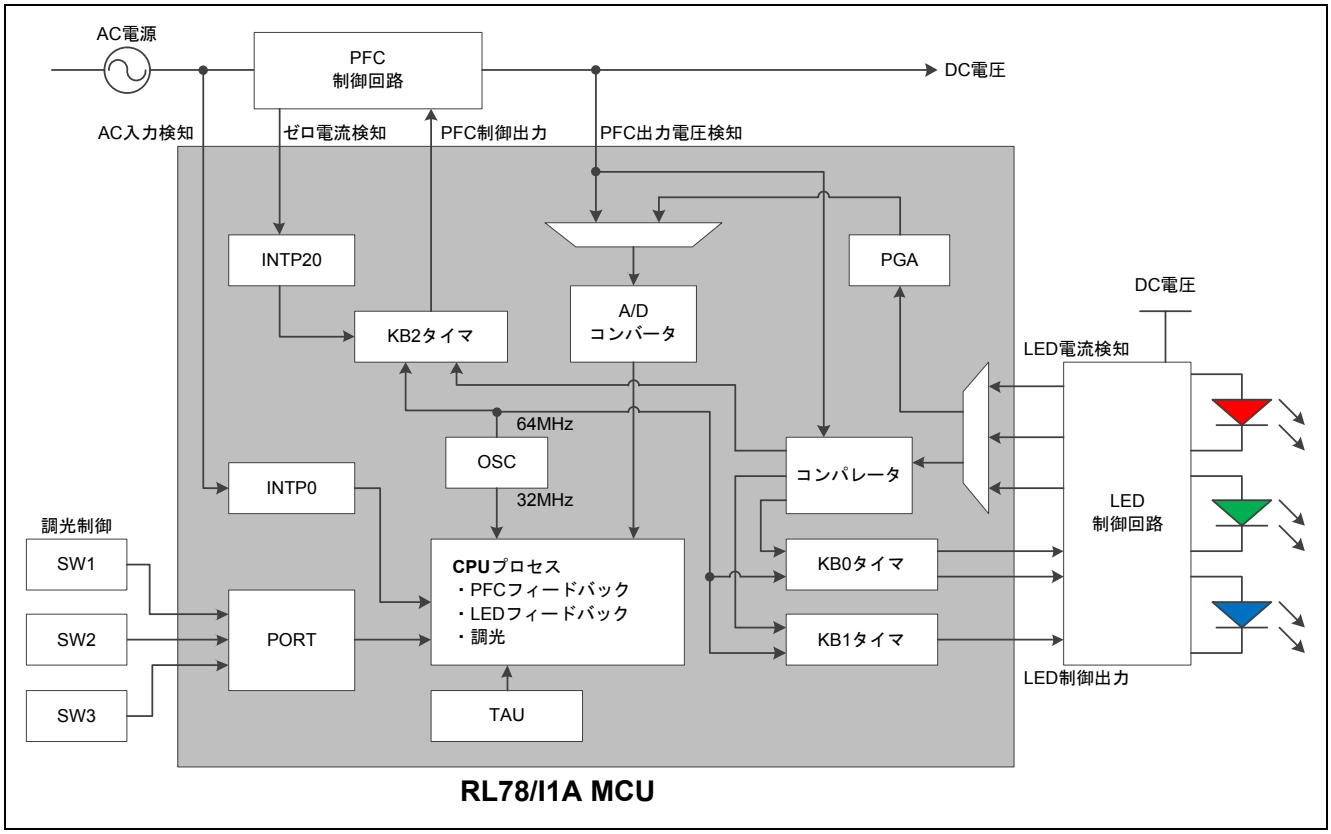


図 1　RL78/I1A AC/DC LED 制御評価ボードのブロック図

2.3 RL78/I1A の端子機能

表 1 に、使用端子と LED 制御システムにおけるそれぞれの機能を簡単に説明します。

表 1 端子機能

機能	機能名	端子名	I/O	説明
LED 制御	TKBO00	P200	O	LED1 PWM 出力
	TKBO01	P201	O	LED2 PWM 出力
	TKBO10	P202	O	LED3 PWM 出力
	ANI4/CMP1P	P24	I	LED1 電流監視用アナログ入力
	ANI5/CMP2P	P25	I	LED2 電流監視用アナログ入力
	ANI6/CMP3P	P26	I	LED3 電流監視用アナログ入力
PFC 制御	TKBO21	P205	O	PFC 出力
	INTP0	P137	I	AC ゼロクロス検出用割り込み入力
	INTP20	P203	I	ゼロ電流検出用割り込み入力
	ANI7/CMP4P	P27	I	PFC 出力電圧監視用アナログ入力
その他	P75	P75	I	スイッチ 1 入力
	P76	P76	I	スイッチ 2 入力
	P77	P77	I	スイッチ 3 入力

3. 制御ソフトウェア

本章では、本サンプル・プログラムのファイル構成、使用する RL78/I1A の内蔵周辺機能およびそれらの初期設定について説明します。また、全体的な動作（定電流と調光制御）概要と、フィードバック制御用に実装した PI 制御の説明とフロー・チャートを示します。

本サンプル・プログラムは、ルネサス エレクトロニクスホームページから Applilet EZ for HCD をダウンロードして、Switch モード 1 を選択していただくことで入手できます。

3.1 ファイル構成

本サンプル・プログラムは、以下のファイルで構成します。

表 2 ファイル一覧

File Name	Contents
Init.asm	オプション・バイトの設定
r_systeminit.c	システム初期化関数
r_userinit.c	周辺初期化設定の最上位関数
r_main.c	メイン関数
r_usermain.c	LED 調光関数
r_swmode.c	SW 入力判定用初期化関数と実行関数
r_swmode1.c	SW 入力による状態遷移判定関数
r_ac.c	AC 電源入力判定用関数
r_led_autotuning.c	オートチューニング関数
r_led_dimcontrol.c	PFC 制御、LED 調光制御関数
r_led_dimrequest.c	LED 定電流制御関数
r_led.c	LED 制御用初期化関数
r_timer.c	AC 入力、SW 入力サンプリングのタイマ関数（インターバル･モード）
r_wdt.c	ウォッチドッグ初期化関数
r_user.h	クロックおよび ADC の設定に使用するパラメータ定義
r_led_dimming.h	LED 制御に使用するパラメータ定義
r_led_user.h	制御対象 LED の定義
r_system.h	r_systeminit.c 用定義
r_ac.h	r_ac.c 用定義
r_led.h	r_led_dimcontrol.c 用定義
r_swmode.h	r_swmode.c 用定義
r_wdt.h	r_wdt.c 用定義
r_timer.h	r_timer.c 用定義
r_macrodriver.h	ドライバ用マクロ定義

3.2 グローバル変数一覧

表 3 に本プログラムのグローバル変数を示します。

表 3 グローバル変数一覧(1)

データ型	変数名	概要	使用関数
unsigned char	ucConnectLed	LED 接続状態 b1: LED1 接続状態（1:接続、0:非接続） b2: LED2 接続状態（1:接続、0:非接続） b3: LED3 接続状態（1:接続、0:非接続）	auto_tuning() at_check() at_finalize() dim_led1() dim_led2() dim_led3()
unsigned char	ucReqLed	LED 動作要求 b1: LED1 動作要求（1:点灯、0:消灯） b2: LED2 動作要求（1:点灯、0:消灯） b3: LED3 動作要求（1:点灯、0:消灯）	int_tm00() boost_pfc() stop_led1() stop_led2() stop_led3() dim_led1() dim_led2() dim_led3() dim_trig() dim_ini() at_check() at_run() at_finalize()
unsigned char	ucStateLight	LED 動作状態 0＝All OFF b0: PFC 昇圧（1:ON、0:OFF） b1: LED1 動作状態（1:ON、0:OFF） b2: LED2 動作状態（1:ON、0:OFF） b3: LED3 動作状態（1:ON、0:OFF）	int_tm00() start_pfc() boost_pfc() feedback_led1() feedback_led2() feedback_led3() stop_led1() stop_led2() stop_led3() dim_trig() dim_ini()
unsigned short	usErrStatus	エラー情報（1:あり、0:なし） 0＝ エラーなし b0: LED 未検出 b1: PFC 出力過電圧（昇圧前） b2: PFC 出力過電圧（昇圧中） b3: 昇圧タイムアウト b4: PFC 出力過電圧（点灯中） b5: LED1 過電流 b6: LED2 過電流 b7: LED3 過電流 b8: PFC 出力過電圧(コンパレータ検知)	main() start_pfc() boost_pfc() dim_trig() feedback_led1() feedback_led2() feedback_led3() feedback_pfc() at_check() at_run() at_finalize()

表 4　グローバル変数一覧(2)

データ型	変数名	概要	使用関数
unsigned char	ucFeedbackCnt	フィードバック処理フェーズ 0x01: LED1 フィードバック処理 0x02: LED2 フィードバック処理 0x03: LED3 フィードバック処理 0x04: PFC フィードバック処理 0x05: その他処理	int_tm00() boost_pfc() at_finalize()
unsigned char	g_ucSwInputMode[3]	動作モード 0x00 : SWMODE_LED_OFF LED 消灯状態 0x01 : SWMODE_LED_ON_MIN LED 調光値最小状態(SW ON) 0x02 : SWMODE_LED_ON_MAX LED 調光値最大状態(SW ON) 0x03 : SWMODE_LED_MAXFADE LED ステップ調光状態(最大目標) 0x04 : SWMODE_LED_MINFADE LED ステップ調光状態(最小目標) 0x05 : SWMODE_LED_ON_UP LED 調光停止状態(最小目標) 0x06 : SWMODE_LED_ON_DN LED 調光停止状態(最大目標) 0x07 : SWMODE_LED_ON_MAX_REL LED 調光値最大状態(SW OFF) 0x08 : SWMODE_LED_ON_MIN_REL LED 調光値最小状態(SW OFF)	SwMode_init() SwMode1_get Value()
unsigned short	g_usACInputOn Counter	AC 入力検出カウンタ 1ms ごとにカウント	ACInput_init() AC_pulse_check() AC_on_check()
unsigned char	g_ucACInputOff Counter	AC 入力未検出カウンタ 1ms ごとにカウント	ACInput_init() ACInput_Timer Interrupt() AC_pulse_check() AC_on_check()
unsigned short	usCntInterval	オートチューニング処理用インターバルカウンタ 64μs ごとにカウント	at_check() at_run() at_finalize()
unsigned long	ulCntBoost	PFC 昇圧時間計測カウンタ 64μs ごとにカウント	start_pfc() boost_pfc()
unsigned short	ushOffsetPhase	LED オフセットフェーズ 1: LED1 オフセット処理 2: LED2 オフセット処理 3: LED3 オフセット処理	boost_pfc()
unsigned short	ushOffsetCount	LED オフセット回数カウンタ 3LED オフセット完了ごとにカウント	boost_pfc()

表 5　グローバル変数一覧(3)

データ型	変数名	概要	使用関数
signed short	shAdOffsetLed1	LED1 オフセット AD 値	start_pfc() boost_pfc() feedback_led1() at_finalize()
signed short	shAdOffsetLed2	LED2 オフセット AD 値	start_pfc() boost_pfc() feedback_led2() at_finalize()
signed short	shAdOffsetLed3	LED3 オフセット AD 値	start_pfc() boost_pfc() feedback_led3() at_finalize()
unsigned char	g_ucSWInputCounter	SW 入力時間計測カウンタ 1ms ごとにカウント	SwMode_init() SwMode_TimerInterrupt()
unsigned char	g_ucSwCheckTimePass[3]	SW 判定時間経過情報 0: 経過なし 1: 経過あり	SwMode_init() SwMode_TimerInterrupt() SWInput_check()
unsigned char	g_ucSwPushTime[3]	SW 押下継続時間 10ms ごとにカウント	SwMode_init() SwMode_TimerInterrupt() SWInput_check()
unsigned char	g_ucSwPushFlag[3]	SW 長押しフラグ 0: 長押し判定中 1: 長押し確定	SwMode_init() SWInput_check()
unsigned char	g_ucSwFix_Data[3]	SW 動作確定データ 0x00: OFF 確定 0x1F: ON 確定	SwMode_init() SWInput_check()
unsigned char	g_ucSwIn_Data[3]	SW 入力データ	SwMode_init() SWInput_check()
unsigned short	g_usSwValue[3]	SW 動作による調光要求値	SwMode_init() SwMode1_getValue()
signed short	shReqLed1	LED1 調光目標要求値	feedback_led1() dim_led1() dim_ini() at_run()
signed short	shReqLed2	LED2 調光目標要求値	feedback_led2() dim_led2() dim_ini() at_run()
signed short	shReqLed3	LED3 調光目標要求値	feedback_led3() dim_led3() dim_ini() at_run()

表 6　グローバル変数一覧(4)

データ型	変数名	概要	使用関数
signed short	shAdCled1Tgt	LED1 調光目標値	feedback_led1() stop_led1() dim_led1() dim_ini()
signed short	shAdCled2Tgt	LED2 調光目標値	feedback_led2() stop_led2() dim_led2() dim_ini()
signed short	shAdCled3Tgt	LED3 調光目標値	feedback_led3() stop_led3() dim_led3() dim_ini()
signed short	shAdLed1	LED1 フィードバック AD 値	feedback_led1() boost_pfc() at_finalize()
signed short	shAdLed2	LED2 フィードバック AD 値	feedback_led2() boost_pfc() at_finalize()
signed short	shAdLed3	LED3 フィードバック AD 値	feedback_led3() boost_pfc() at_finalize()
unsigned short	ushPowLed1	LED1 負荷値	at_finalize()
unsigned short	ushPowLed2	LED2 負荷値	at_finalize()
unsigned short	ushPowLed3	LED3 負荷値	at_finalize()
unsigned short	ushPowTotal	LED 総負荷値	at_finalize()
unsigned long	ulDpfcLed1	総負荷に対する LED1 負荷の比率	dim_led1() at_check() at_finalize()
unsigned long	ulDpfcLed2	総負荷に対する LED2 負荷の比率	dim_led2() at_check() at_finalize()
unsigned long	ulDpfcLed3	総負荷に対する LED3 負荷の比率	dim_led3() at_check() at_finalize()
signed long	slDpfcLed1	PFC フィードフォワード用補正値 PFC PWM デューティ値 ±= PFC フィードフォワード用補正値	feedback_led1() stop_led1() dim_led1()
signed long	slDpfcLed2	PFC フィードフォワード用補正値 PFC PWM デューティ値 ±= PFC フィードフォワード用補正値	feedback_led2() stop_led2() dim_led2()
signed long	slDpfcLed3	PFC フィードフォワード用補正値 PFC PWM デューティ値 ±= PFC フィードフォワード用補正値	feedback_led3() stop_led3() dim_led3()

表 7　グローバル変数一覧(5)

データ型	変数名	概要	使用関数
unsigned long	ulSumDpfco	PFC PWM デューティ積算値	at_finalize()
unsigned long	ulSumCled1	LED1 電流 AD 積算値	at_finalize()
unsigned long	ulSumCled2	LED2 電流 AD 積算値	at_finalize()
unsigned long	ulSumCled3	LED3 電流 AD 積算値	at_finalize()
unsigned long	ulSumDled1	LED1 PWM デューティ積算値	at_finalize()
unsigned long	ulSumDled2	LED2 PWM デューティ積算値	at_finalize()
unsigned long	ulSumDled3	LED3PWM デューティ積算値	at_finalize()
unsigned long	ulSumCled1Count	LED1 PWM デューティ積算回数カウンタ	at_finalize()
unsigned long	ulSumCled2Count	LED2 PWM デューティ積算回数カウンタ	at_finalize()
unsigned long	ulSumCled3Count	LED3 PWM デューティ積算回数カウンタ	at_finalize()
signed long	slLedA1	LED PI 制御係数 A1	feedback_led1() feedback_led2() feedback_led3()
signed long	slLedA2	LED PI 制御係数 A2	feedback_led1() feedback_led2() feedback_led3()
signed long	slPfcA1	PFC PI 制御係数 A1	feedback_pfc()
signed long	slPfcA2	PFC PI 制御係数 A2	feedback_pfc()
union long short	uData1 slErrLED1 sErrLED1[2]	LED1 PI 制御計算結果	boost_pfc() feedback_led1() stop_led1()
union long short	uData2 slErrLED2 sErrLED2[2]	LED2 PI 制御計算結果	boost_pfc() feedback_led2() stop_led2()
union long short	uData3 slErrLED3 sErrLED3[2]	LED3 PI 制御計算結果	boost_pfc() feedback_led3() stop_led3()
union long short	uData4 slErrPfc sErrPfc[2]	PFC PI 制御計算結果	start_pfc() feedback_led1() feedback_led2() feedback_led3() feedback_pfc() stop_led1() stop_led2() stop_led3()

表 8　グローバル変数一覧(6)

データ型	変数名	概要	使用関数
signed short	shAdTempLed1	PI 制御計算用変数 LED1 の誤差値を保持	feedback_led1() stop_led1() dim_ini()
signed short	shAdTempLed2	PI 制御計算用変数 LED2 の誤差値を保持	feedback_led2() stop_led2() dim_ini()
signed short	shAdTempLed3	PI 制御計算用変数 LED3 の誤差値を保持	feedback_led3() stop_led3() dim_ini()
signed long	slErrTemp	PI 制御計算用変数 (前回誤差値×係数 A2)を保持	feedback_led1() feedback_led2() feedback_led3()
signed long	slErrTemp1	PI 制御計算用変数 (最新誤差値×係数 A1)を保持	feedback_led1() feedback_led2() feedback_led3()
unsigned short	ushADtemp	PFC 出力 AD 値	start_pfc() boost_pfc() feedback_pfc()
signed short	shADtempPFC	PI 制御計算用変数 PFC の最新誤差値を保持	feedback_pfc()
signed short	shAdOldVout	PI 制御計算用変数 PFC の前回誤差値を保持	feedback_pfc()
unsigned char	ucIntP00	AC 入力割り込みフラグ ・INTP0	int_p00() AC_pulse_check() at_run() at_finalize()
unsigned char	ucIntTm00	TM00 タイマ割り込みフラグ ・INTTM00	int_tm00() at_check() at_run() at_finalize()

3.3 関数一覧

表 9 に本プログラムの関数を示します。

表 9 関数一覧

関数名	概要
int_p00	INTP0 割り込み処理
int_tm00	INTTM00 割り込み処理
start_pfc	PFC 動作開始処理
boost_pfc	PFC 昇圧処理
feedback_led1	LED1 フィードバック処理
feedback_led2	LED2 フィードバック処理
feedback_led3	LED3 フィードバック処理
feedback_pfc	PFC フィードバック処理
stop_led1	LED1 停止処理
stop_led2	LED2 停止処理
stop_led3	LED3 停止処理
stop_pfcled	PFC、LED 停止処理
dim_led1	LED1 制御要求処理
dim_led2	LED2 制御要求処理
dim_led3	LED3 制御要求処理
dim_trig	調光制御用タイマ動作開始
SWInput_check	SW 入力判定
SwMode1_getValue	SW による調光状態遷移判定
SwValue_StepUp	調光値ステップアップ
SwValue_StepDown	調光値ステップダウン
SwMode_TimerInterrupt	SW 判定用タイマ周期処理
ACInput_check	AC 電源入力確認
AC_pulse_check	AC パルス入力判定処理
AC_on_check	AC 電源入力状態判定
ACInput_TimerInterrupt	AC 入力判定用タイマ周期処理
auto_tuning	オートチューニング処理
at_check	チューニング判定処理
at_run	チューニング実行処理
at_finalize	チューニング完了処理
user_init	ユーザ用初期化処理
hdwinit	レジスタ初期化処理
dim_ini	調光用初期化処理
ACInput_init	AC 入力判定用初期化処理
Timer_init	タイマ用初期化処理
LED_init	LED 制御用初期化処理
SwMode_init	SW 判定用初期化処理
user_main	ユーザ用メイン処理
main	メイン処理
Timer_Interrupt	1ms タイマ周期処理
WDT_Reset	ウォッチドッグタイマリセット処理

3.4 関数仕様

ここではソフトウェア関数の詳細仕様を説明します。

int_p00

概要	INTP0 割り込み処理
宣言	__interrupt void int_p00(void)
説明	AC 電源ゼロクロス検出割り込み
引数	なし
戻り値	なし
備考	割り込み周期= ½ fz

int_tm00

概要	INTTM00 割り込み処理
宣言	__interrupt void int_tm00(void)
説明	・LEDn フィードバック処理 ・PFC フィードバック処理 ・PFC 制御用 PWM タイマ動作開始 ・PFC 昇圧処理 ・その他処理
引数	なし
戻り値	なし
備考	インターバル・タイマ (64μs) LED、PFC フィードバック周期内に、ユーザ処理追加用としてその他処理を設けています。必要に応じて処理を追加してください。また、その場合は LED、PFC フィードバック周期に干渉する可能性があるため、処理時間が 32μs を超えないようにしてください。

start_pfc

概要	PFC 動作開始処理
宣言	void start_pfc(void)
説明	PFC 制御用 PWM タイマの動作を開始する
引数	なし
戻り値	なし
備考	なし

boost_pfc

概要	PFC 昇圧処理
宣言	void boost_pfc(void)
説明	PFC の昇圧処理を行う ・LEDn 制御用 PWM タイマ動作開始 ・LEDn オフセット値取得 PFC 出力昇圧タイムアウト判定を行う PFC 出力過電圧判定を行う
引数	なし
戻り値	なし
備考	なし

feedback_led1

概要	LED1 フィードバック処理
宣言	void feedback_led1(void)
説明	PI 制御式による計算結果を LED1 制御 PWM デューティ周期に反映する LED1 過電流判定を行う LED1 制御に伴い、PFC フィードフォワード制御を行う
引数	なし
戻り値	なし
備考	なし

feedback_led2

概要	LED2 フィードバック処理
宣言	void feedback_led2(void)
説明	PI 制御式による計算結果を LED2 制御 PWM デューティ周期に反映する LED2 過電流判定を行う LED2 制御に伴い、PFC フィードフォワード制御を行う
引数	なし
戻り値	なし
備考	なし

feedback_led3

概要	LED3 フィードバック処理
宣言	void feedback_led3(void)
説明	PI 制御式による計算結果を LED3 制御 PWM デューティ周期に反映する LED3 過電流判定を行う LED3 制御に伴い、PFC フィードフォワード制御を行う
引数	なし
戻り値	なし
備考	なし

feedback_pfc

概要	PFC フィードバック処理
宣言	void feedback_pfc(void)
説明	PI 制御式による計算結果を PFC 制御 PWM デューティ周期に反映する PFC 出力過電圧判定を行う
引数	なし
戻り値	なし
備考	なし

stop_led1

概要	LED1 停止処理
宣言	void stop_led1(void)
説明	LED1 制御用 PWM タイマの停止処理を行う LED1 制御に伴い、PFC フィードフォワード制御を行う
引数	なし
戻り値	なし
備考	なし

stop_led2

概要	LED2 停止処理
宣言	void stop_led2(void)
説明	LED2 制御用 PWM タイマの停止処理を行う LED2 制御に伴い、PFC フィードフォワード制御を行う
引数	なし
戻り値	なし
備考	なし

stop_led3

概要	LED3 停止処理
宣言	void stop_led3(void)
説明	LED3 制御用 PWM タイマの停止処理を行う LED3 制御に伴い、PFC フィードフォワード制御を行う
引数	なし
戻り値	なし
備考	なし

stop_pfcled

概要	PFC & LED 停止処理
宣言	void stop_pfcled(void)
説明	PFC、LED 制御用 PWM タイマをすべて停止する
引数	なし
戻り値	なし
備考	なし

dim_led1

概要	LED1 制御要求処理
宣言	void dim_led1(unsigned short ushInLed)
説明	LED1 調光目標要求値および LED 動作要求の設定を行う
引数	• unsigned short 調光目標値
戻り値	なし
備考	なし

dim_led2

概要	LED2 制御要求処理
宣言	void dim_led2(unsigned short ushInLed)
説明	LED2 調光目標要求値および LED 動作要求の設定を行う
引数	• unsigned short 調光目標値
戻り値	なし
備考	なし

dim_led3

概要	LED3 制御要求処理
宣言	void dim_led3(unsigned short ushInLed)
説明	LED3 調光目標要求値および LED 動作要求の設定を行う
引数	• unsigned short 調光目標値
戻り値	なし
備考	なし

dim_trig

概要	調光制御用タイマ動作開始
宣言	void dim_trig(void)
説明	PFC、LED 制御に同期するタイマ TM00 の動作を開始する
引数	なし
戻り値	なし
備考	なし

SWInput_check

概要	SW 入力判定	
宣言	unsigned char SWInput_check(unsigned char ucSwNum)	
説明	各 SW の ON/OFF 判定を行う ・単押し判定 ・長押し判定	
引数	• unsigned char	SW No. (0~2)
戻り値	• 0: SW なし • 1: SW 単押し • 2: SW 長押し • 3: SW OFF • 4: SW ON (立ち上がりエッジ)	
備考	なし	

SwMode1_getValue

概要	SW による調光状態遷移判定	
宣言	unsigned short SwMode1_getValue(unsigned char ucChannel)	
説明	SW 状態と動作モードにより調光目標値を決定する 動作モードの定義は下記を参照	
	・SWMODE_LED_OFF	LED 消灯状態
	・SWMODE_LED_ON_MIN	LED 調光値最小状態(SW ON)
	・SWMODE_LED_ON_MIN_REL	LED 調光値最小状態(SW OFF)
	・SWMODE_LED_MINFADE	LED ステップ調光状態(最小目標)
	・SWMODE_LED_ON_UP	LED 調光停止状態(最小目標)
	・SWMODE_LED_ON_MAX	LED 調光値最大状態(SW ON)
	・SWMODE_LED_ON_MAX_REL	LED 調光値最大状態(SW OFF)
	・SWMODE_LED_MAXFADE	LED ステップ調光状態(最大目標)
	・SWMODE_LED_ON_DN	LED 調光停止状態(最大目標)
引数	• unsigned char	LED チャネル No. (1~3)
戻り値	• 調光目標値	
備考	なし	

SwValue_StepUp

概要	調光値ステップアップ	
宣言	unsigned short SwValue_StepUp(unsigned short usStep)	
説明	LED 調光要求値を 1 ステップ増加する 本プログラムでは、1 ステップ＝1	
引数	• unsigned short	現在の調光要求値
戻り値	処理後の調光要求値	
備考	なし	

SwValue_StepDown

概要	調光値ステップダウン
宣言	unsigned short SwValue_StepDown(unsigned short usStep)
説明	LED 調光要求値を 1 ステップ減少する 本プログラムでは、1 ステップ＝1
引数	• unsigned short 現在の調光要求値
戻り値	処理後の調光要求値
備考	なし

SwMode_TimerInterrupt

概要	SW 判定用タイマ周期処理
宣言	void SwMode_TimerInterrupt(void)
説明	1ms タイマ周期で下記処理を行う ・SW 入力時間判定 ・SW 押下継続時間判定
引数	なし
戻り値	なし
備考	なし

ACInput_check

概要	AC 電源入力確認
宣言	void ACInput_check(void)
説明	AC パルス入力判定を行う 23ms 間 AC パルスを検出できない場合は PFC、LED 制御を停止し、AC パルス入力待ちとなる
引数	なし
戻り値	なし
備考	なし

AC_pulse_check

概要	AC パルス入力判定処理
宣言	void AC_pulse_check(void)
説明	AC パルス入力割り込み(INTP0)状態を判定し、AC 入力検出カウンタのカウント、および AC 未検出カウンタのクリアを行う
引数	なし
戻り値	なし
備考	なし

AC_on_check

概要	AC 電源入力状態判定
宣言	void AC_on_check(void)
説明	AC 電源入力割り込み(INTP0)待ちを行う
引数	なし
戻り値	なし
備考	なし

ACInput_TimerInterrupt

概要	AC 入力判定用タイマ周期処理
宣言	void ACInput_TimerInterrupt(void)
説明	1ms タイマ周期で下記処理を行う ・AC 入力未検出カウンタをカウントする
引数	なし
戻り値	なし
備考	なし

auto_tuning

概要	オートチューニング処理
宣言	void auto_tuning(void)
説明	オートチューニングを行う
引数	なし
戻り値	なし
備考	なし

at_check

概要	チューニング判定処理
宣言	void at_check(void)
説明	SW1 の入力待ちを行う SW1 入力を 51ms 間連続検出した時、LED1、LED2、LED3 の点灯を開始する
引数	なし
戻り値	なし
備考	なし

at_run

概要	チューニング実行処理
宣言	void at_run(void)
説明	64μs ごとに LED1、LED2、LED3 の順番で、各 LED 最大調光値まで調光要求値を増加する 3LED すべてが最大調光要求値まで増加し 16ms 経過した時、チューニング完了処理を行う
引数	なし
戻り値	なし
備考	なし

at_finalize

概要	チューニング完了処理
宣言	void at_finalize(void)
説明	PFC フィードフォワード制御に使用するための各 LED の総負荷比率を算出する LED 未検出判定を行う
引数	なし
戻り値	なし
備考	なし

user_init

概要	ユーザ用初期化処理
宣言	void user_init(void)
説明	各処理に必要な初期化処理を行う
引数	なし
戻り値	なし
備考	必要に応じて初期化処理を追加してください

hdwinit

概要	レジスタ初期化処理
宣言	void user_init(void)
説明	各レジスタの初期化処理を行う 初期化内容の詳細は 3.5 節を参照
引数	なし
戻り値	なし
備考	必要に応じて初期化処理を追加してください

dim_ini

概要	調光用初期化処理
宣言	void dim_ini(void)
説明	LED 調光に必要な変数の初期化処理を行う
引数	なし
戻り値	なし
備考	必要に応じて初期化処理を追加してください

ACInput_init

概要	AC 入力判定用初期化処理
宣言	void ACInput_init(void)
説明	AC 入力判定に必要な変数の初期化処理を行う
引数	なし
戻り値	なし
備考	必要に応じて初期化処理を追加してください

Timer_init

概要	タイマ用初期化処理
宣言	void Timer_init(void)
説明	タイマ処理に必要な変数の初期化処理を行う
引数	なし
戻り値	なし
備考	必要に応じて初期化処理を追加してください

LED_init

概要	LED 制御用初期化処理
宣言	void LED_init(void)
説明	LED 制御に必要な変数の初期化処理を行う
引数	なし
戻り値	なし
備考	必要に応じて初期化処理を追加してください

SwMode_init

概要	SW 判定用初期化処理
宣言	void SwMode_init(void)
説明	SW 判定に必要な変数の初期化処理を行う
引数	なし
戻り値	なし
備考	必要に応じて初期化処理を追加してください

user_main

概要	ユーザメイン処理
宣言	void user_main(void)
説明	TM01 タイマ(1ms)周期監視 AC 電源入力状態判定 SW1、SW2、SW3 入力読み込み ・調光値変更 ・LED ON/OFF LED1、LED2、LED3 調光処理
引数	なし
戻り値	なし
備考	なし

main

概要	メイン処理
宣言	void main(void)
説明	起動時初期化処理 ・AC 電源入力状態判定 ・ユーザ初期化処理 ・LED オフセット値取得 ユーザメイン処理
引数	なし
戻り値	なし
備考	なし

Timer_Interrupt

概要	1ms タイマ周期処理
宣言	void Timer_Interrupt (void)
説明	TM01 タイマ(1ms)割り込み待ちを行う 割り込み検出後、下記のタイマ周期処理を行う ・AC 入力判定用タイマ周期処理 ・SW 判定用タイマ周期処理
引数	なし
戻り値	なし
備考	なし

WDT_Reset

概要	ウォッチドッグタイマリセット処理
宣言	void WDT_Reset(void)
説明	ウォッチドッグ・タイマ・イネーブル・レジスタ(WDTE)のリセットを行い、ウォッチドッグタイマを再スタートする
引数	なし
戻り値	なし
備考	なし

3.5 内蔵周辺機能の初期化

本サンプル・プログラムでは、以下の RL78/I1A マイクロコントローラ内蔵周辺機能を使用します。

- 調光処理間隔制御： 16 ビット TAU チャネル 0
- フィードバック処理間隔制御： 16 ビット TAU チャネル 1
- ディザリング機能による PWM 出力： 16 ビット・タイマ KB0、KB1 および KB2
- フィードバック入力増幅： PGA チャネル n（n＝4、5、6）
- PFC 電圧フィードバック入力： 10 ビット A/D コンバータ・チャネル n（n＝7）
- LED 電流フィードバック入力（PGA 出力から）： 10 ビット A/D コンバータ・チャネル n（n＝3）

表 10 A/D コンバータ・チャネル割り当て

アナログ入力チャネル	機能
ANI0	AVREFP
ANI1	AVREFM
ANI4	LED1 フィードバック電流入力
ANI5	LED2 フィードバック電流入力
ANI6	LED3 フィードバック電流入力
ANI7	PFC 出力電圧入力

ハードウェア初期化には以下の設定が含まれます。

1. オプション・バイトの設定
   — ウォッチドッグ・タイマ動作の停止
   — LVD（低電圧検出器）動作モードおよび検出レベルの設定
     - VLVI（低電圧検波電圧）を 4.06 V に設定
     - LVD をリセット・モードに設定（VDD が VLVI 未満のとき、内部リセットを生成する）
   — システム・クロック・ソースとして高速内蔵発振器（4 MHz）を選択
   — オンチップ・デバッグの有効化

2. 周辺設定
   — PLL を使用して CPU クロック周波数を 32 MHz に設定（内蔵高速発振クロック f_{IH} x 1/2 の 16 倍）
   — 周辺機能クロック供給の設定
   — I/O ポートの設定
   — 16 ビット TAU チャネル 0 の設定
     - カウント・クロックを f_{CLK}（32 MHz）に設定
     - 間隔時間を 1ms（(TDR00 + 1) / f_{CLK}）に設定
     - 割り込み INTTM00 のマスクを解除
   — 16 ビット TAU チャネル 1 の設定
     - カウント・クロックを f_{CLK}（32MHz）に設定
     - 間隔時間を 64 μs（(TDR01 + 1) / f_{CLK}）に設定
     - 割り込み INTTM01 のマスクを解除
   — A/D コンバータの設定
     - A/D 変換時間を 2.125 μs に設定
     - 割り込み INTAD をマスク
   — プログラマブル利得増幅器（PGA）の設定
     - PGA 増幅率を 8 倍に設定
     - 入力チャネルを ANI2 に設定

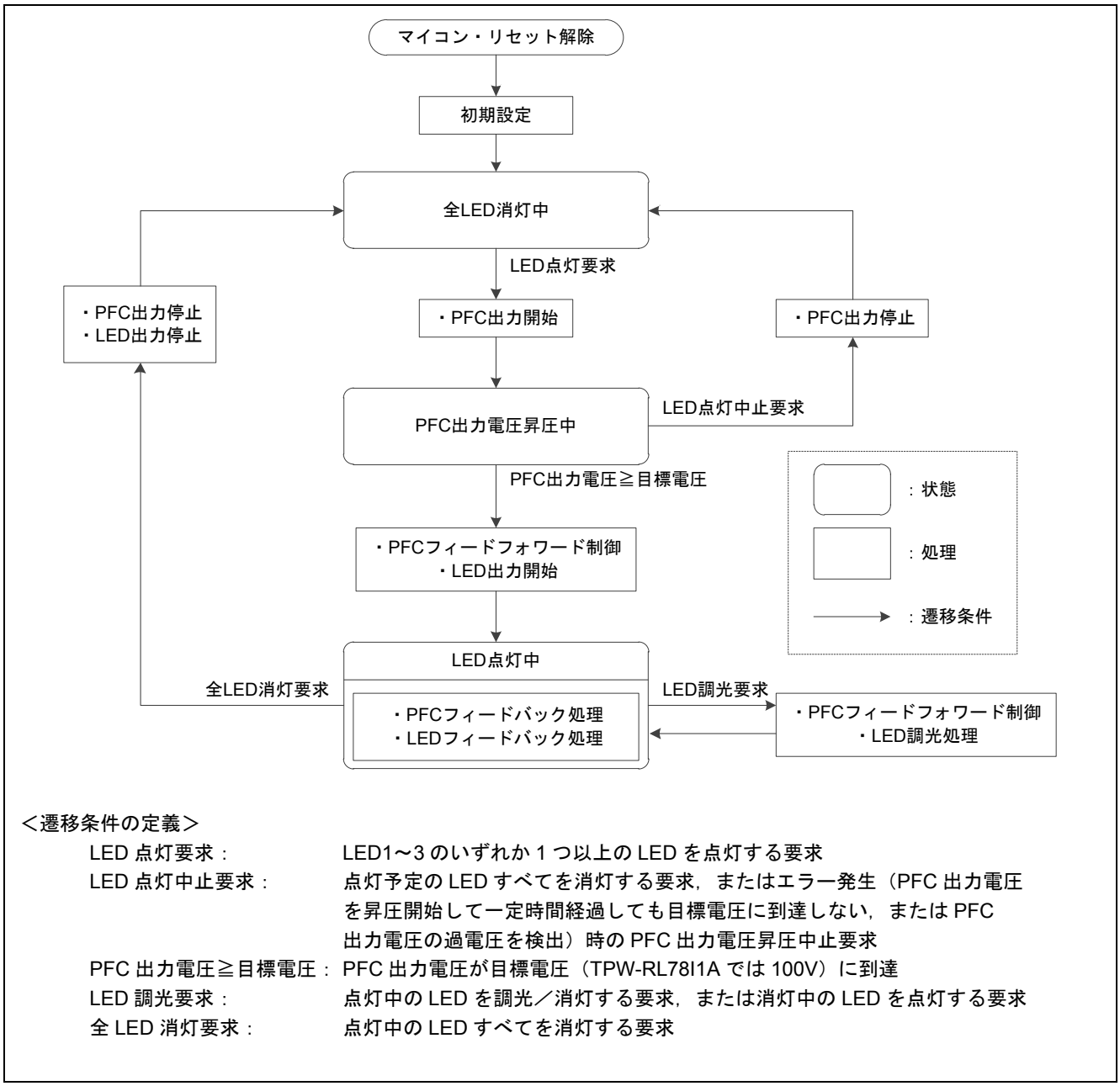
— コンパレータの設定
- コンパレータ 1、コンパレータ 2、コンパレータ 3 を LED 過電圧検出に設定
- コンパレータ 4、コンパレータ 5 を PFC 過電圧検出に設定
- コンパレータ 1～4 の出力を許可

— 16 ビット・タイマ KB の設定
- カウント・クロックを f_{PLL}＝64 MHz に設定
- TKBO00、TKBO01、TKBO10 PWM 出力ディザリング機能を有効化
- TKBO および TKB1 の動作モードをスタンドアロン・モードに設定
- TKBO00、TKBO01、TKBO10 の出力のデフォルト・レベルをロー・レベルに、アクティブ・レベルをアクティブ・ハイに設定
- PWM 出力の周波数を 250 kHz（f/(TKBCRn0 + 1)、n＝0、1）に設定
  - 分解能 8 ビットの 64 MHz カウント・クロック・ソース（64 MHz/28）
- 割り込み INTTMKB0、INTTMKB1 をマスク
- TKBO00 強制出力停止機能のトリガとしてコンパレータ 1 を設定
- TKBO01 強制出力停止機能のトリガとしてコンパレータ 2 を設定
- TKBO10 強制出力停止機能のトリガとしてコンパレータ 3 を設定
- TKBO21 強制出力停止機能のトリガとしてコンパレータ 4、コンパレータ 5 を設定

初期化後、LED を駆動するために 16 ビット・タイマ KB0 と KB1 から 250 kHz PWM 信号が出力されます。A/D コンバータは、フィードバック入力端子からセンス電圧を検出し、それらを ADC 目標レベルと比較し、定電流を維持するように PWM 出力のデューティを調整します。

3.6 システム動作概要

図 2 に RL78/I1A マイクロコントローラによる PFC 制御搭載 LED 照明制御の状態遷移図を示します。状態は大きく分けて「全 LED 消灯中」、「PFC 出力電圧昇圧中」、「LED 点灯中」の 3 つあります。



＜遷移条件の定義＞

LED 点灯要求：	LED1～3 のいずれか 1 つ以上の LED を点灯する要求
LED 点灯中止要求：	点灯予定の LED すべてを消灯する要求，またはエラー発生（PFC 出力電圧を昇圧開始して一定時間経過しても目標電圧に到達しない，または PFC 出力電圧の過電圧を検出）時の PFC 出力電圧昇圧中止要求
PFC 出力電圧≧目標電圧：	PFC 出力電圧が目標電圧（TPW-RL78I1A では 100V）に到達
LED 調光要求：	点灯中の LED を調光／消灯する要求，または消灯中の LED を点灯する要求
全 LED 消灯要求：	点灯中の LED すべてを消灯する要求

図 2　RL78/I1A マイクロコントローラによる PFC 制御搭載 LED 照明制御の状態遷移図

3.7 LED 制御

3.7.1 SW 入力による調光制御

本プログラムでは、SWn 注1 入力（単押し、長押し）によって LED の輝度を調節することができます。SW1 が LED1、SW2 が LED2、SW3 が LED3 にそれぞれ対応しています。

SW 入力は、メイン処理周期に同期して 10ms ごとにサンプリングし、5 回連続一致で ON/OFF 判定します。

本プログラムにおける SW 動作の判定は以下のように行います。

- ON : 5 回連続で Low レベル検出
- OFF : 5 回連続で High レベル検出
- 単押し : ON→OFF エッジの検出
- 長押し : 500ms 連続 ON 検出、以降は 50ms 毎連続 ON 検出で長押しと判定

判定した SW 動作によって異なる状態に遷移し、LED の輝度を表す調光レベル（目標レベル：‘shAdCllednTgt’ 注2）を決定します。図 3 に SW による調光レベルの遷移図を示します。

各 LED の調光を開始する前に、SW1 を押してオートチューニングを行う必要があります。オートチューニングの詳細については 3.8.3 節で説明します。

【注】1. n＝1、2、3（ボード TPW-RL78I1A 上で、LED1 が赤、LED2 が緑、LED3 が青）
2. サンプル・プログラム中のグローバル変数をシングル・クォーテーション（‘ ’）で示します。

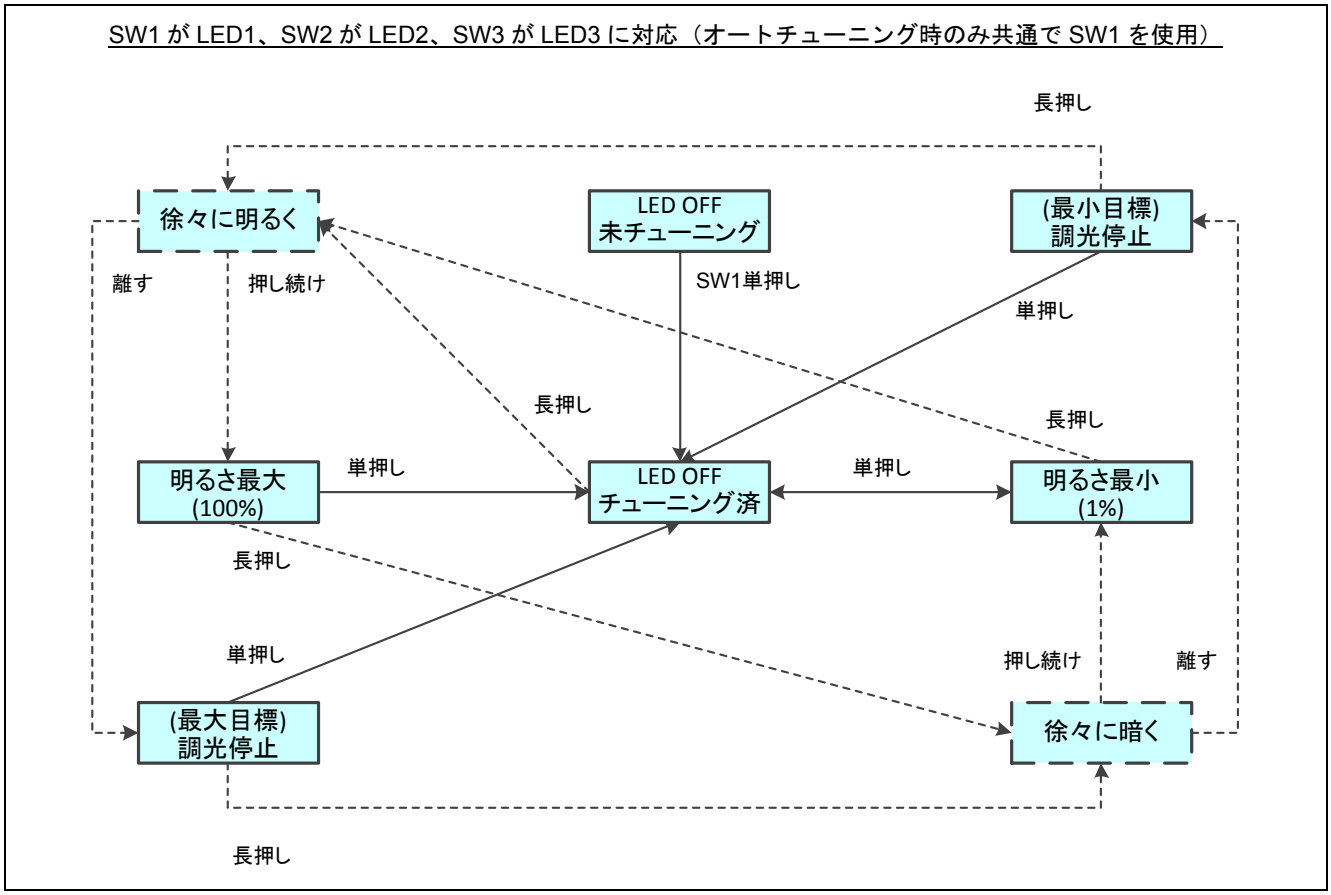


図 3 SW による LED 調光レベル状態遷移図

3.7.2 定電流制御

TAU チャネル 0、TAU チャネル 1、タイマ KB0、タイマ KB1、および A/D コンバータを初期化後に起動します。動作を開始すると、メイン・ループ・プログラムは SW によって決定された、各 LED チャネルの調光目標レベル（‘shAdClednTgt’ 注 2）を計算します。

フィードバック処理は、64μs ごとに発生する TAU チャネル 0（INTTM00）の割り込み処理ルーチン内で、PWM 出力のデューティを調整することによって LEDn 注 1 の定電流制御を行います。

A/D 変換‘shAdLedn’ 注 2 の結果は、PI 制御によるフィードバック処理で、最後の結果 ‘shAdTempLedn’ 注 2 および目標レベル‘shAdClednTgt’ 注 2 と比較されます。PI 制御によるフィードバック処理についての詳細は 3.9 節を参照してください。

このフィードバック・プロセスを使用することによって、センス電圧を目標レベル電圧に近づけることができます。目標レベルが変化した場合、センス電圧が目標に達する前にフィードバック・プロセスが 2 回以上実行されます。

TM00 割り込みが 64μs ごとに発生すると、アナログ入力チャネルはフィードバック・チャネルの 1 つに変更されます。さらに、割り込み処理ルーチンの次回の繰り返しで入力チャネルを次のフィードバック・チャネルにシフトするために、‘ucFeedbackCnt’ 変数を更新します。

【注】1. n＝1、2、3（ボード TPW-RL78I1A 上で、LED1 が赤、LED2 が緑、LED3 が青）
2. サンプル・プログラム中のグローバル変数をシングル・クォーテーション（‘ ’）で示します。

LED の定電流制御に使用する回路を図 4 に示します。RL78/I1A PWM 出力はプリドライバを介しバックコンバータの MOSFET を ON/OFF 制御し、次に、A/D 変換入力は LED フィードバック電流を測定し、RL78/I1A CPU は LED における定電流を維持するために PI 制御を実装します。A/D 変換目標値は前述 3.7.1 の通り SW 操作により決定します。

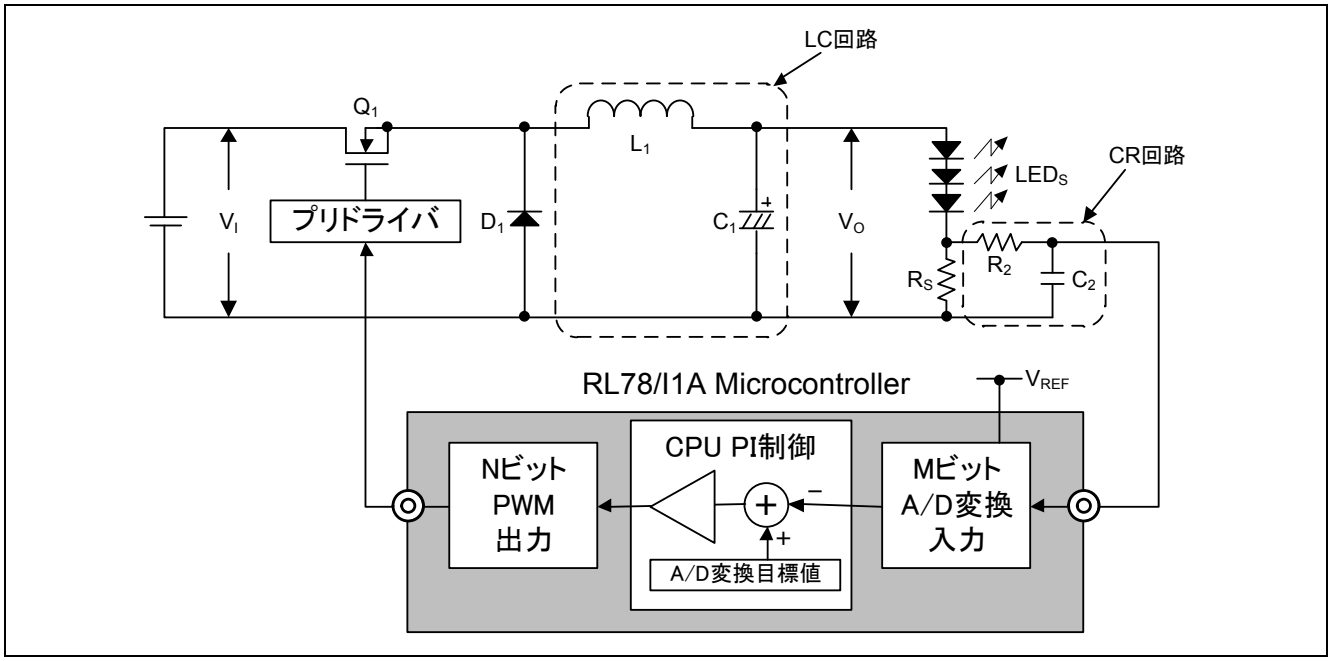


図 4 LED 定電流制御用バックコンバータ回路図

3.8 PFC 制御

3.8.1 PFC 制御の概要

(1) PFC 制御とは

図 5(a)に交流電源における理想的な電源電圧・電流波形を示します。このとき，電圧と電流は同位相かつ正弦波形であり，力率＝1 となります。図 5(b)は PFC 制御を行っていない電源における実際の電源電圧・電流波形を示しています。このとき，電流の導通時間は短く，そのピーク電流値も大きくなります。また，これにより電圧波形の頭もつぶれています。このように力率が低い場合，次のような問題が発生します。

- 高調波が発生し，規制に準拠できず製品を出荷できない可能性がある。
- ピーク電流が大きいため，電線に必要以上に太い線が必要となる。
- ブレーカーが落ちやすくなる。

そこで，力率を改善するため，PFC 制御が必要となります。この PFC 制御として，一般的な LED 照明では部品点数が比較的少なく，スイッチング・ノイズが小さいといった観点から臨界導通モード（CRM: Critical Conduction Mode）が用いられており，PFC 制御を用いた場合の交流電源の電源電圧・電流波形は図 5(c)に示すようになります。このように電源電流の ON/OFF を繰り返すことで電流値の分布を分散し，その平均値が電源電圧と同位相かつ正弦波形となるように制御しています。この臨界導通モードによる PFC 制御は，RL78/I1A マイクロコントローラを用いて実現することが可能です。

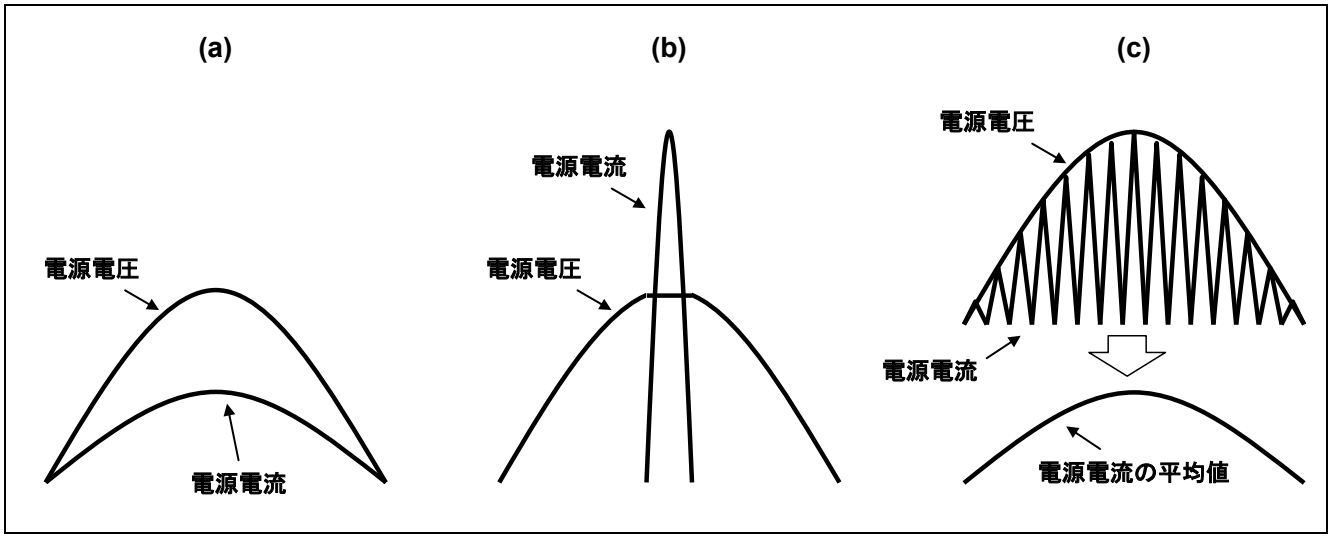


図 5 電源電圧・電流の波形と PFC 制御

(2) RL78/I1A による PFC 制御

RL78/I1A マイクロコントローラは，単体動作モードによるタイマ・リスタート機能と A/D コンバータを組み合わせることにより，臨界導通モードの PFC 制御を行うことが可能です。したがって，マイクロコントローラとは別に PFC 制御用の専用アナログ IC を使用する必要がありません。また，RL78/I1A マイクロコントローラは PFC 制御だけでなく，LED 制御すなわち PFC 出力の負荷制御も行うため，事前に負荷変動を予見制御することが可能です。したがって，負荷が変動してからフィードバックをかける方法に比べて，負荷変化時の電圧変動を小さく抑えることができます。

RL78/I1A マイクロコントローラによるフライバック・コンバータ型 PFC 回路の構成例を図 6 に示します。ここで，PFC 制御に必要な端子は PFC 出力（TKBO21 端子），ゼロ電流検出入力（INTP20 端子），PFC 出力電圧監視入力（ANI7/CMP4P 端子）の 3 端子です。

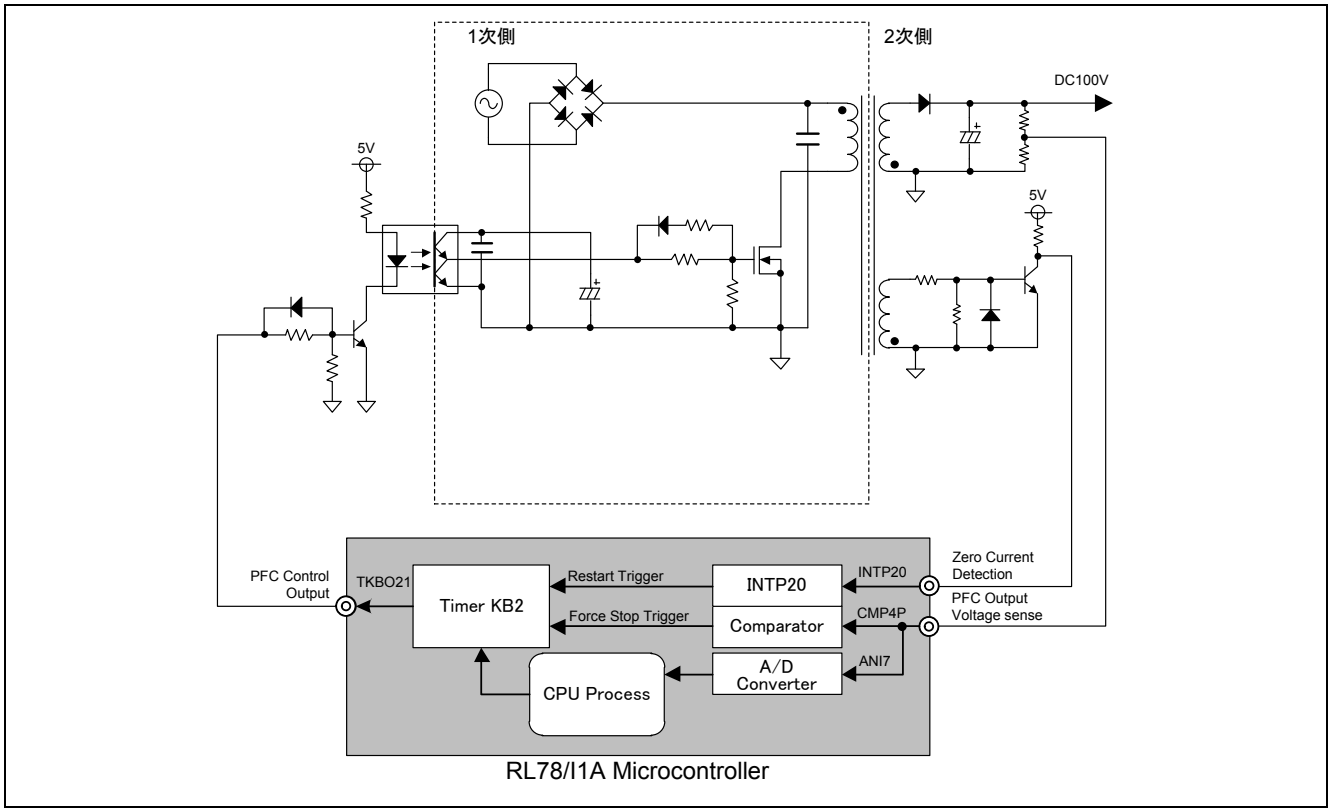


図 6 RL78/I1A マイクロコントローラによるフライバック・コンバータ型 PFC 回路構成例

図 7 にこの構成で PFC 制御を行った際の波形を示します。ここで，PFC 出力（TKBO21 出力）が ON の状態において，$I_{ON} = (V_{IN}／L)×t_{ON}$ となります。これにより，t_{ON} 出力の時間を一定にすることで，I_{ON} は V_{IN} に比例するので，I_{ON} のピーク電流 I_{PEAK} は V_{IN} の波形と同位相かつ正弦波形となります。また，電流波形が三角波であることから，平均電流 $I_{AVERAGE} = I_{PEAK} / 2$ となり，これも V_{IN} に比例します。したがって，平均電流波形が電源電圧波形と同位相かつ正弦波形となり，力率が 1 に近い波形を実現することができます。

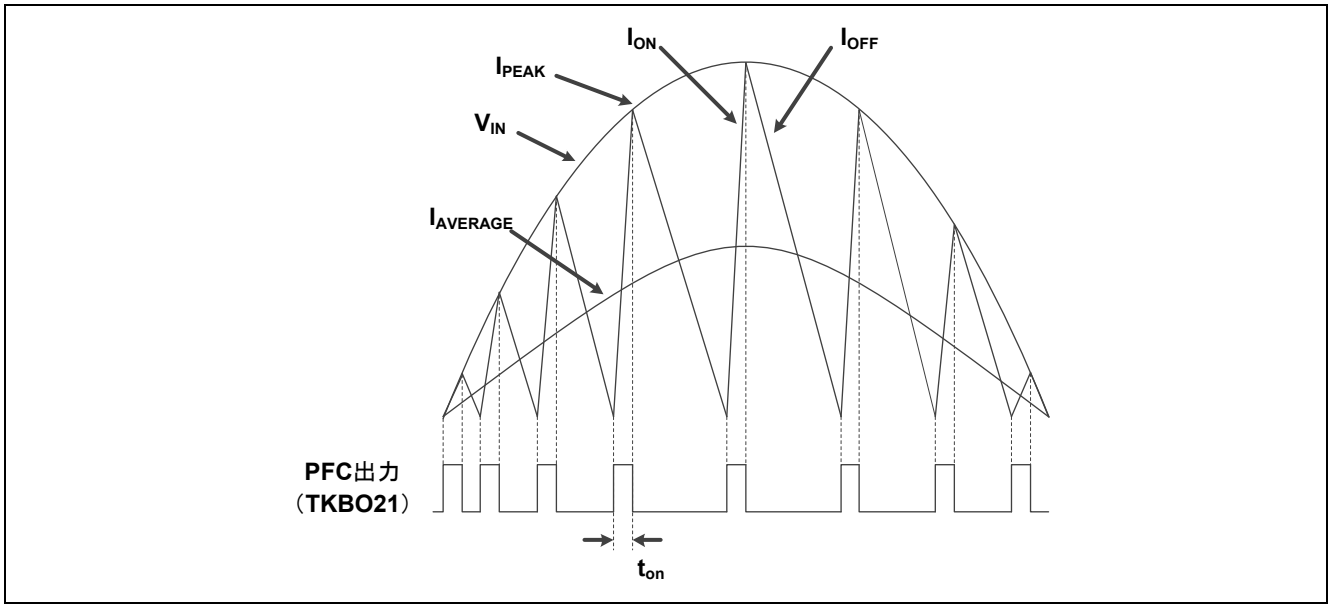


図 7　RL78/I1A マイクロコントローラによる PFC 制御波形

次に，PFC 制御に使用する RL78/I1A マイクロコントローラの周辺ハードウェアおよびその機能を示します。

- 16 ビット・タイマ KB2　　･･･PFC 出力
- A/D コンバータ　　･･･PFC 出力電圧監視

この周辺ハードウェアによる PFC 制御の特徴は次の通りです。

- 16 ビット・タイマ KB2 の単体動作モード機能により，ゼロ電流検出時に自動的に（ソフトウェア処理を介さずに）PFC 出力を ON することが可能。
- 16 ビット・タイマ KB2 のカウント・クロックに 64MHz を選択でき，PFC 出力の ON 時間を 15ns 単位で制御可能。また，ゼロ電流未検出時の PFC 出力リスタート周期を最大約 1.02ms まで 15ns 単位で柔軟に設定可能。
- 16 ビット・タイマ KB2 は，タイマ動作を停止することなく PFC 出力の ON 時間を変更することが可能。
- 最大 10 ビット分解能の A/D コンバータにより PFC 出力電圧を検出可能。

3.8.2 負荷が変化する場合の PFC 制御（フィードフォワード制御）

LED の調光／点灯／消灯時など，負荷が変化する際の PFC 制御波形を図 8 に示します。ここで，LED 照明は RL78/I1A マイクロコントローラにより制御するので, 負荷が変化するタイミングやどの程度負荷が変化するかを予測することが可能です。したがって，調光により大きく負荷が変化する場合においても，同時にその調光に合わせた PFC 出力を行うこと，すなわち予見制御が可能であり，PFC 出力電圧の揺れを抑えることができます。

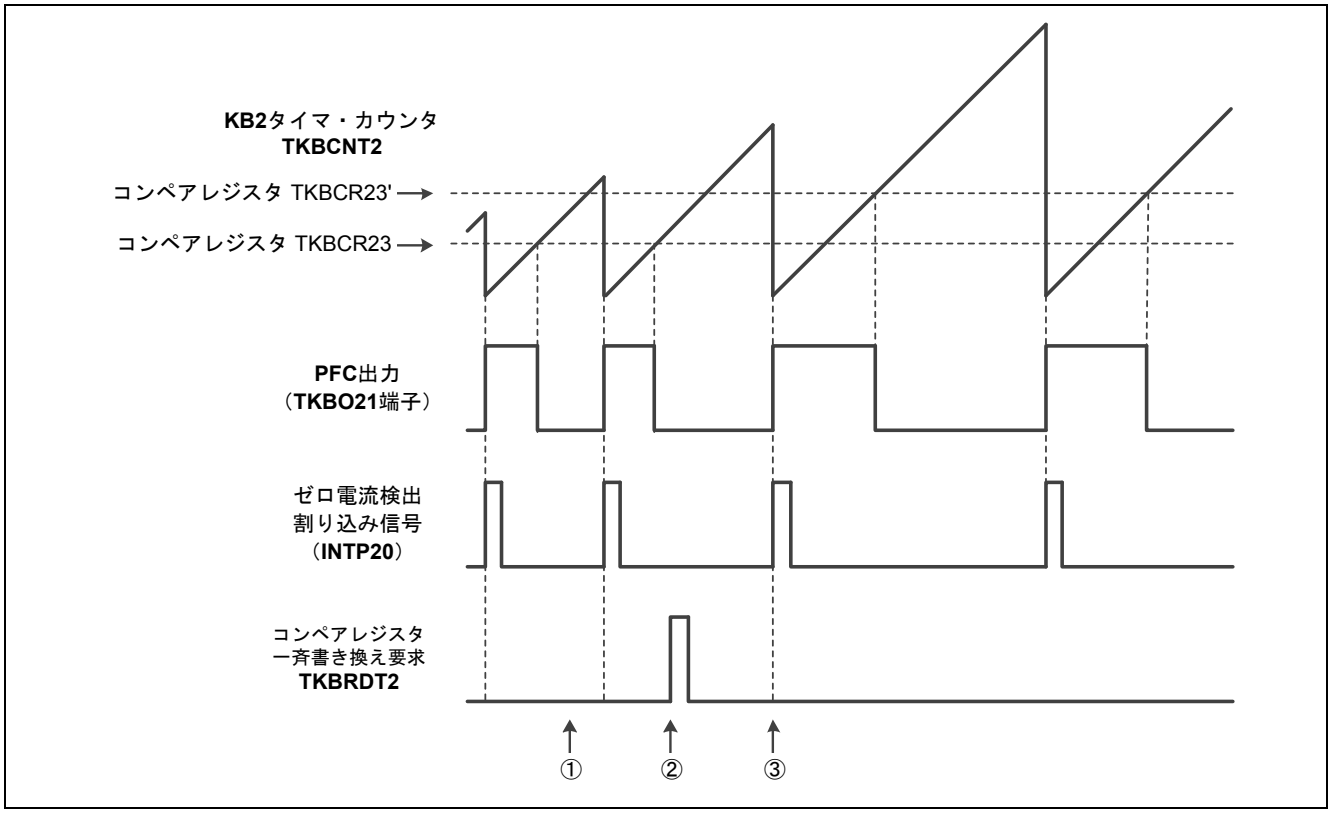


図 8　負荷が変化する場合の PFC 制御波形

動作概要は次の通りです。

① PFC 出力の ON 時間（図 8 の TKBCR23）を負荷に応じた値（図 8 の TKBCR23’）に増減させる。
② ①の更新を実際の出力に反映するため，トリガ・レジスタ 2（TKBTRG2）の bit0（TKBRDT2）に 1 を書き込み、コンペアレジスター一斉書き換え要求を行う。
③ 次のゼロ電流検出割り込み発生時に PFC 出力の ON 時間が更新される。

3.8.3 オートチューニング

各 LED の調光および PFC 制御を開始する前に、オートチューニングを行う必要があります。この処理では、PFC 制御の起動を行います。また、評価ボードに接続された LED を最大負荷状態にし、そこから各 LED の負荷比率を算出します。算出した負荷比率は通常調光時の PFC フィードフォワード制御に使用します。オートチューニング終了後、各 LED は全消灯状態となります。

以下に LED1 の負荷比率算出方法を示します。実際には 3 つの LED それぞれの負荷比率を算出します。

1. LED1 のフィードバック制御周期で、‘ulSumCled1’注 1 に LED1 のフィードバック A/D 値を加算し、‘ulSumDled1’注 1 に LED1 の PWM デューティ値を加算します。

```
; ulSumCled1 += (unsigned long)(shAdLed1 - shAdOffsetLed1);
; ulSumDled1 += TKBCR01;
```

2. INTP0 割り込み周期(ゼロクロス)で、‘ulSumDpfco’注 1 に PFC の PWM デューティ値を加算します。

```
; ulSumDpfco += TKBCR23;
```

3. INTP0 割り込みカウンタが 128 に達したとき、LED1 フィードバック A/D 最大値を 2 で割った値が LED1 フィードバック A/D 積算値を加算回数で割った値よりも小さいことを確認し、LED が接続されているかどうかを大まかに判断します。

```
; if ((AD_CLED1_TGT >> 1) < (ulSumCled1 / ulSumCled1Count))
```

4. LED が接続されていることを確認できたら、LED1 の PWM デューティ積算値に LED1 フィードバック A/D 最大値を乗算することで LED1 の負荷に比例した値を算出します。

```
; ushPowLed1 = (unsigned short)((ulSumDled1 * AD_CLED1_TGT) >> 24);
```

変数のデータ型によるオーバフローを考慮して、24 ビット右シフトし有効値のみを取り出します。

5. LED1 の負荷値を、3 つの LED の総負荷を保持する値に加算します。

```
; ushPowTotal += ushPowLed1;
```

6. 128 サイクルの間に加算された PFC PWM デューティ積算値から平均値を算出するため、128 で除算します。

```
; ulSumDpfco = ulSumDpfco >> 7;
```

7. 最後に、LED1 フィードバック A/D 最大値を 128 サイクルの PFC PWM デューティ平均値で割ることにより、総負荷に対する LED1 の負荷比率を算出します。

```
; ulDpfcLed1 = ((unsigned long)(ushPowLed1) * ulSumDpfco) / ushPowTotal;
; ulDpfcLed1 = (ulDpfcLed1 << 16) / AD_CLED1_TGT;
```

【注】1. サンプル・プログラム中のグローバル変数をシングル・クォーテーション（‘ ’）で示します。

3.8.4 PFC 出力電圧のフィードバック

PFC 出力電圧が常に一定値となるように，PFC 出力電圧を A/D コンバータで監視し，PFC 出力の ON 時間にフィードバック制御する必要があります。ここで，負荷が大きく変化するのは基本的に LED 負荷を意図的に調光／点灯／消灯した場合に限られるので，この際の PFC 出力電圧の変化は前述 3.8.2 の通りフィードフォワード制御により抑えることが可能です。したがって，フィードバック制御は LED が一定の明るさで点灯している状態，すなわち LED 負荷がほぼ一定の状態に対して行います。ここでは LED 制御同様に PI 制御によるフィードバック制御を行います。PI 制御によるフィードバック処理についての詳細は 3.9 節を参照してください。

3.9 PI 制御によるフィードバック制御

3.9.1 PI 制御について

LED の定電流制御と調光制御（オン／オフを含む）、PFC 出力電圧制御は、いずれも PI 制御によるフィードバック処理を使用することによって実現できます。

PI フィードバックの一般的な式を以下に示します。
係数 A1 および A2 の計算方法については、3.9.2 節「PI 制御式の係数の計算」を参照してください。

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

D（n）： 最新の PWM 出力デューティ
D（n－1）： 前回の PWM 出力デューティ
E（n）： 最新の誤差値＝（A/D 変換目標値）－（最新の A/D 変換測定値）
E（n－1）： 前回の誤差値＝（A/D 変換目標値）－（前回の A/D 変換測定値）
A1、A2： 係数

1) LED の定電流制御

LED 電流 I_{LED} の目標値は、A/D 変換目標値に基づいて判断されます。A/D 変換目標値が X_{TARGET} 注であるときの設定方法を以下に示します。

$$X_{TARGET} = \frac{(I_{LED} \times 8) \times R_S}{V_{REF}} \times 2^M$$

たとえば、LED 電流 I_{LED}＝350mA で定電流制御を実行するとき、センス抵抗 R_S＝1.3Ω、A/D コンバータ基準電圧 V_{REF}＝5V および A/D 変換解像度 M＝10 ビットと仮定して、A/D 変換目標値 X_{TARGET}＝744 を設定します。
LED 電流フィードバック測定値は、利得 8 のプログラマブル利得増幅器を使用して増幅されるので、A/D 変換目標値 X_{TARGET} に 8 を乗算する必要があります。

2) LED の調光制御

電流調光制御は、LED 定電流の目標値を変更することによって実行できます。つまり、調光に合わせて A/D 変換目標値 X_{TARGET} を変更することができます。その結果、PI 制御の目標値が変更され、RL78/I1A マイクロコントローラは X_{TARGET} の理想的な値に向けたフィードバック制御を実行します。たとえば、LED 電流を 350mA から 100mA に変更するには、X_{TARGET} 値を 744 から 216 に変更します。

3) PGA 入力オフセット電圧補正

PGA（プログラマブル・ゲイン・アンプ）を使用するとき、±5mV～±10mV の入力オフセット電圧を増幅してしまいます。その結果 PGA は正確な電圧をフィードバック・ループに供給できません。
そのためサンプル・プログラムでは、正のオフセット電圧を考慮し、補正する処理を行っています。
各チャネルの LED フィードバック処理の初回時、すなわち LED がまだオフであるときに、LED フィードバック電圧値を計算し、これらの値を'shAdOffsetLedn'変数として保存します。これらの値は、電流が LED 中を流れていないとき、PGA によって誘導されるオフセット電圧を表します。LED フィードバック処理では、LED が点灯するとき、オフセット電圧を消去するために次の LED フィードバック電圧値からこれらの値を減算します。

さらに、RL78/I1A AC/DC LED 制御評価ボードは、各チャネル・フィードバック回路上のプルアップ抵抗（R110、R210、R310）によって正のオフセット電圧からの影響を低減するように設計されています。

動作の概要を以下に示します。

① PGA 増幅を開始します。
② センス抵抗を使用して測定した LED フィードバック電圧の A/D 変換を開始します。
③ A/D 変換目標値を読み取ります。
④ オフセット電圧を考慮に入れて PI 制御の「A2×E（n－1）」を計算します。
⑤ A/D 変換の結果 LED で過電流が発生した場合は、LED 出力の処理を停止します。
（この場合は、以下の⑥から⑧までの PI 制御の処理を実行しません。）
⑥ A/D 変換の結果 LED で過電流が発生しない場合は、オフセット電圧を考慮に入れて PI 制御の「A1×E（n）＋A2×E（n－1）」を計算します。
⑦ 最後の PWM 出力デューティ D（n－1）を⑥の「A1×E（n）＋A2×E（n－1）」の結果と比較します。
⑧ 「D（n－1）＋A1×E（n）＋A2×E（n－1）」の計算結果が最小 PWM デューティ値から最大 PWM デューティ値までの範囲内にある場合は誤り計算の結果に応じてデューティ D（n）を設定し、それ以外の場合はデューティ D（n）を最大値または最小値に設定します。
⑨ PWM 出力のデューティ設定値を更新します。
⑩ LED フィードバック電圧の A/D 変換値を最後の値として保存します。

4) PFC 出力電圧制御

PFC 出力電圧制御での A/D 変換目標値 X_{TARGET} の設定方法を以下に示します。

$$X_{TARGET} = \frac{\left(\frac{V_{PFCO}}{33}\right) \times 2^{M}}{V_{REF}}$$

PFC 出力電圧 V_{PFCO}＝100V であり、PFC フィードバック電圧は 1/33 で分圧されています。A/D コンバータ基準電圧 V_{REF}＝5V および A/D 変換解像度 M＝10 ビットですので、A/D 変換目標値 X_{TARGET}＝620 を設定します。（但し、個体差により変動する可能性がありますので、実機に合わせて A/D 変換目標値を調整してください）

3.9.2 PI 制御式の係数の計算

ここでは、3.9.1 節に示す PI 制御式で係数を計算する方法について説明します。係数 A1 および A2 は次式から得られます。

$$A_1 = (\pi \times f_Z \times T + 1) \cdot K_P$$
$$A_2 = (\pi \times f_Z \times T - 1) \cdot K_P$$

π : 円周率
f_Z : ゼロ点周波数
T : フィードバック周期
K_P : 比例定数

つまり、係数 A1 および A2 は、3 つのパラメータ f_Z、T、K_P を判断することによって計算できます。これらのパラメータは、LED 制御回路、PFC 制御回路の利得から得られます。

1. LED 制御回路における PI 制御式係数の計算

1) 制御回路の極点周波数からのゼロ・ポイント周波数（f_Z）の計算

図 4 に示すように、LED 制御回路には 2 つの極点（LC 回路の極点と CR 回路の極点）があります。これらの極点周波数は、それぞれのカットオフ周波数に等しいと見なすことができます。前者を f_{C1}、後者を f_{C2} とすると、L_1＝2.2mH、C_1＝33 μF、C_2＝0.1 μF、R_2＝220 Ω の場合は次の値が得られます。

$$f_{C1} = \frac{1}{2\pi\sqrt{L_1 \cdot C_1}} = 0.6[kHz]$$
$$f_{C2} = \frac{1}{2\pi \cdot C_2 \cdot R_2} = 7.2[kHz]$$

次に、以下に示すように、これらの 2 つの周波数よりも低いゼロ・ポイント周波数を選択します。

$$f_Z = 500Hz$$

2) ゼロ・ポイント周波数（f_Z）からのフィードバック周期（T）の計算

サンプリング定理により、フィードバック周期 T の逆数に等しいサンプリング周波数はゼロ・ポイント周波数 f_Z の 2 倍以上でなければなりません。
つまり、フィードバック周期 T とゼロ・ポイント周波数 f_Z の関係は、次式とする必要があります。

$$T < \frac{1}{2f_Z}$$

したがって、f_Z＝500Hz とすると、フィードバック周期 T は 1ms 未満でなければなりません。

また、フィードバック処理専用のCPU負荷を考慮する必要があります。合計3つのLEDチャネルは
定電流フィードバック制御を必要とします。したがって、このサンプル・プログラムでは、CPU 負荷は以下の図 9 に示すように LED チャネルごとにフィードバック制御を実行するために 64 μ s の周期で分配されます。これにより、フィードバック周期 T は次のように設定されます。

$$T = 320\mu s$$

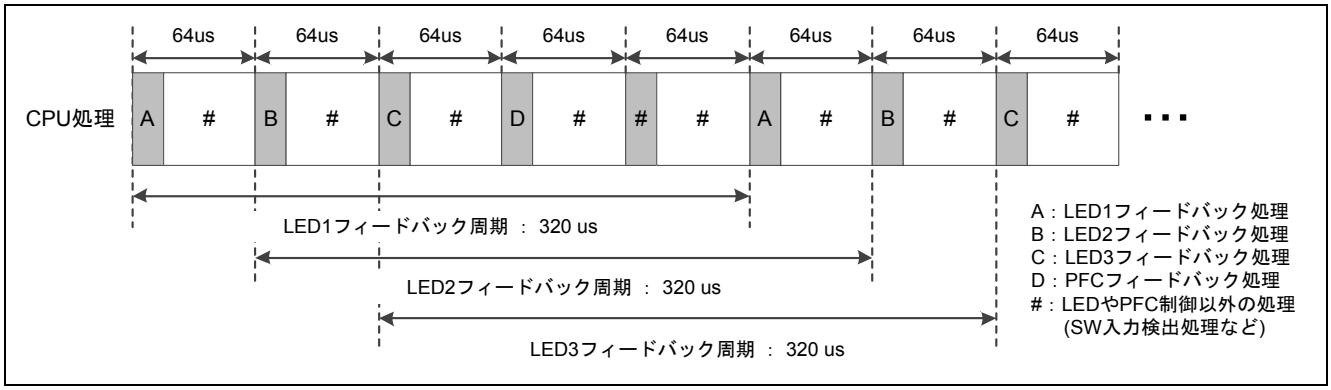


図 9　フィードバック処理のための CPU 負荷分散

3)　マイクロコントローラの利得（ADC 入力 / PWM 出力）からの比例定数 K_P の計算

マイクロコントローラの利得（A/D コンバータ入力／PWM 出力）は、特定の A/D 変換分解能と PWM 分解能に対する LED 電流の変化に注目することによって取得できます。
まず、固有の A/D 変換分解能に対する LED 電流の変化を判断する必要があります。LED 電流が I_{LED}、センス抵抗 R_S によるフィードバック電圧の A/D 変換結果が X、A/D 変換分解能が M ビット、ADC 基準電圧が V_{REF} であるとき、次式が成り立ちます。

$$I_{LED} \cdot R_S = \frac{V_{REF} \cdot X}{2^M}$$

ここで、1 に等しい A/D 変換値（X＝1）に対する LED 電流の変化を i_{AD} とすると、次の結果が得られます。

$$i_{AD} = \frac{V_{REF}}{R_S \cdot 2^M}$$

次に、固有の PWM 分解能に対する LED 電流の変化を判断する必要があります。LED 電流が I_{LED}、LED 順電圧の合計が V_{FT}、入力電圧が V_I、（PWM 出力のデューティ・レジスタ値＋1）が Y、PWM 出力分解能が N ビットであるとき、次式が成り立ちます。

$$I_{LED} \cdot R_S + V_{FT} = \frac{V_I \cdot Y}{2^N}$$

ここで、1 に等しい PWM デューティ値（Y＝1）に対する LED 電流の変化を i_{PWM} とすると、さらに LED 順電圧は一定のままなので、次の結果が得られます。

$$i_{PWM} = \frac{V_I}{R_S \cdot 2^N}$$

したがって、利得 i_{PWM}／i_{AD} は上式から以下と推定します。

$$\frac{i_{PWM}}{i_{AD}} = \frac{V_I}{V_{REF}} \cdot 2^{(M-N)}$$

A/D 変換分解能 M が 13 ビット（ADC の 10 ビット＋2^3＝8 増幅利得による PGA の 3 ビット）、PWM 出力分解能 N が 12 ビット（PWM の 8 ビット＋ディザリング機能のための 4 ビット）、入力電圧 V_I が 5V、A/D コンバータ基準電圧 V_{REF} が 5V であることを考慮すると、次の利得結果（A/D コンバータ入力／PWM 出力）が得られます。

$$\frac{i_{PWM}}{i_{AD}} = 2$$

比例定数 K_P は、この利得の逆数よりも小さい値に設定する必要があります。

$$K_P < \frac{1}{\left(\frac{i_{PWM}}{i_{AD}}\right)}$$

ここでは、K_P を次のように選択します。

$$K_P = 0.05$$

上記の結果から、LED 制御での PI 制御係数 A1 および A2 を計算することができます。

$$A_1 = 0.075132$$
$$A_2 = -0.024868$$

サンプル・プログラムでは、両方の係数は、PWM デューティおよび誤差値と同様に、整数変数を取得し、計算を容易にするために、2^{16}（＝65536）を乗算します。

$$A_1 = 4923$$
$$A_2 = -1629$$

2. PFC 制御回路における PI 制御式係数の計算

同様に、PFC 制御回路においても PI 制御係数を計算します。
PFC 制御回路において、f_Z、T、K_P は次の値になります。

$$\boxed{\begin{aligned} f_Z &= 1Hz \\ T &= 320\mu s \\ K_P &= 1.0 \end{aligned}}$$

上記の結果から、PFC 制御での PI 制御係数 A1 および A2 を計算することができます。

$$\boxed{\begin{aligned} A_1 &= 1.001 \\ A_2 &= -0.999 \end{aligned}}$$

サンプル・プログラムでは、両方の係数は、PWM デューティおよび誤差値と同様に、整数変数を取得し、計算を容易にするために、2^{16}（＝65536）を乗算します。

$$\boxed{\begin{aligned} A_1 &= 65601 \\ A_2 &= -65470 \end{aligned}}$$

4. フローチャート

4.1 メイン処理

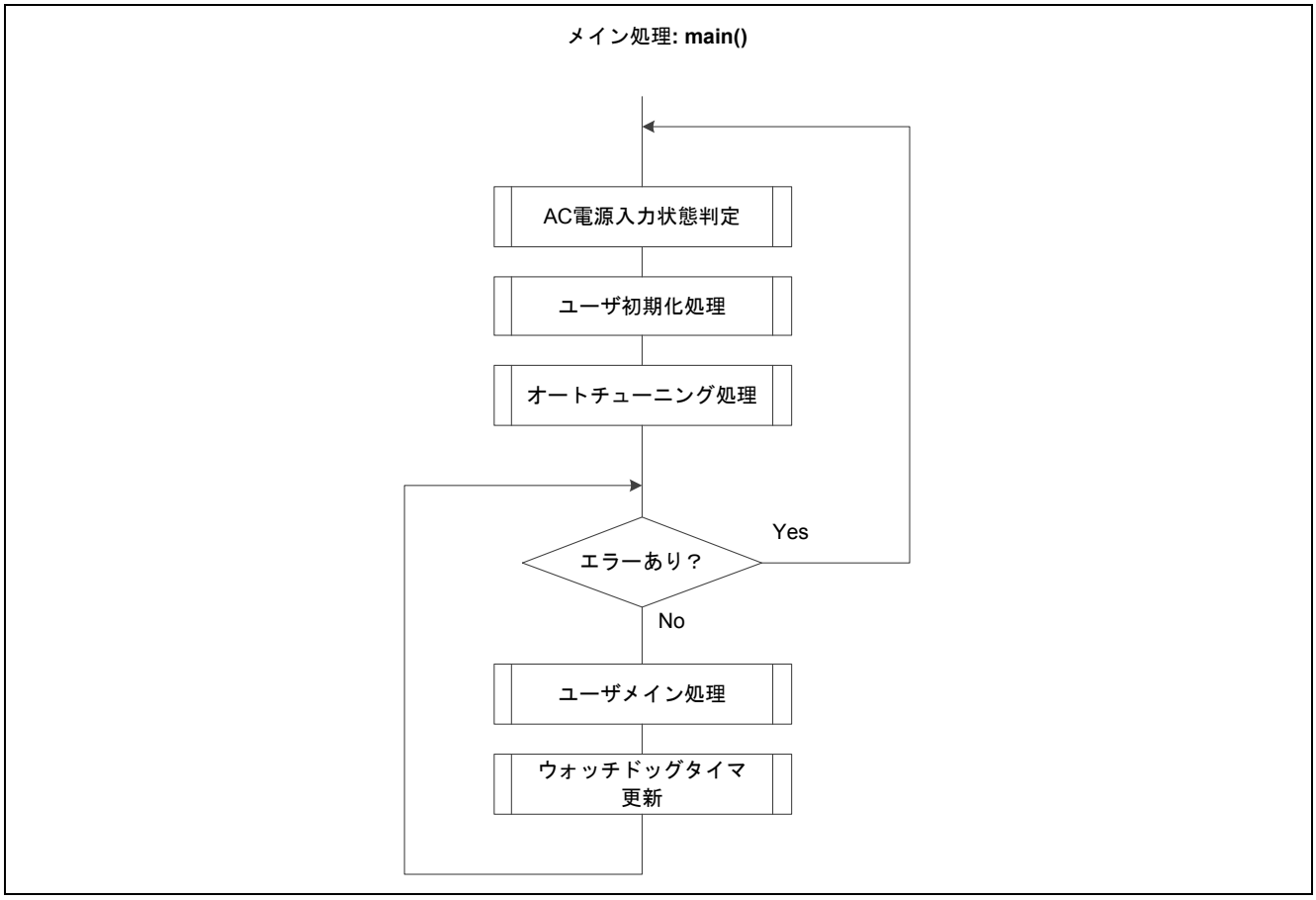


図 10 メイン処理フローチャート

4.2 ユーザメイン処理

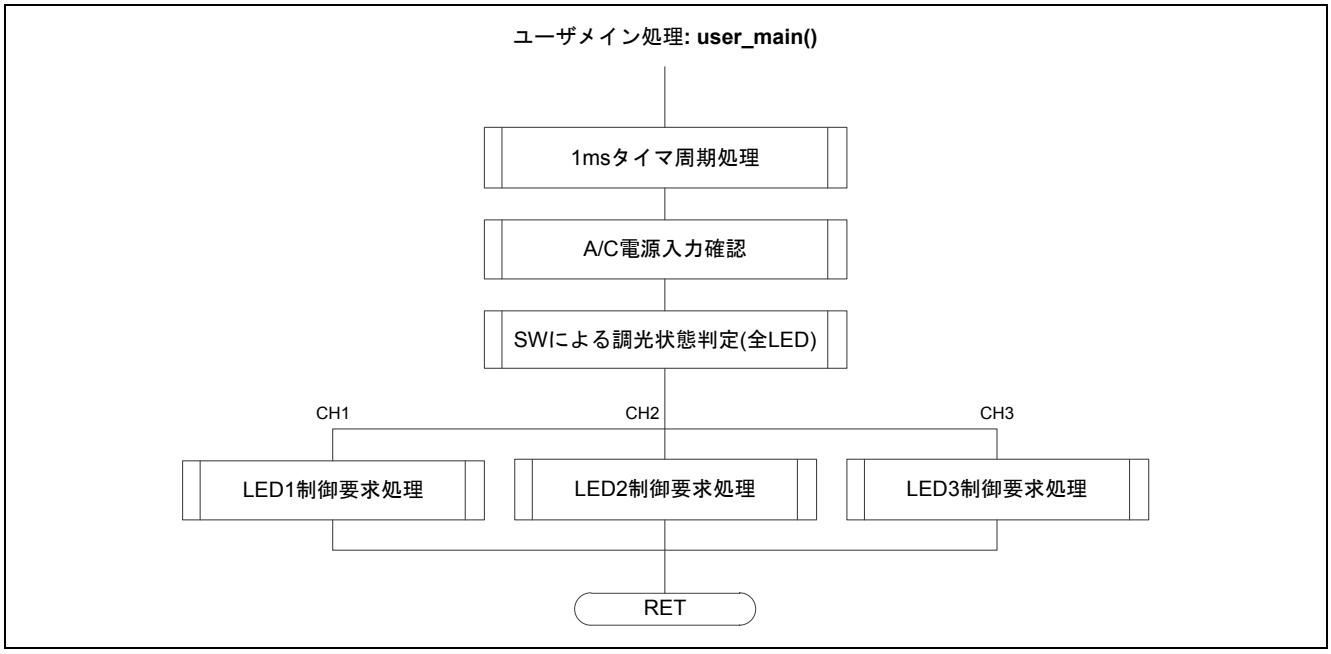


図 11 ユーザメイン処理フローチャート

4.3 ユーザ初期化処理

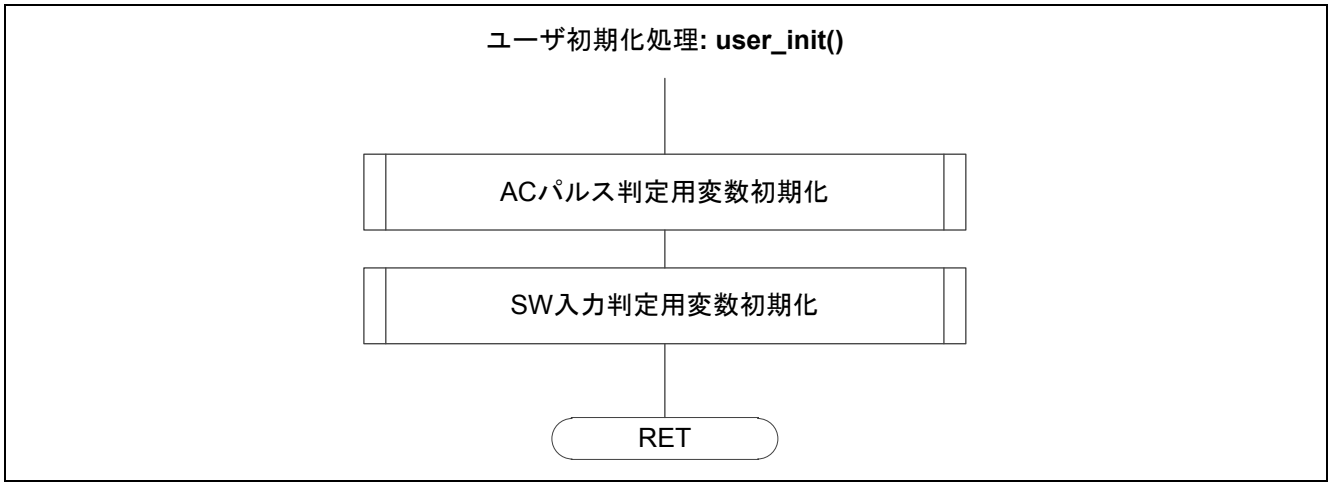


図 12 ユーザ初期化処理フローチャート

4.4 INTP00 割り込み

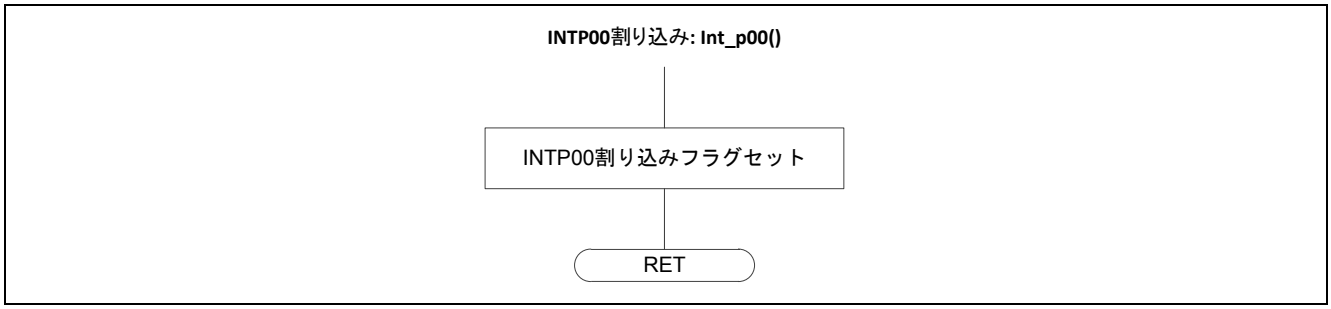


図 13 INTP00 割り込みフローチャート

4.5 INTTM00 割り込み

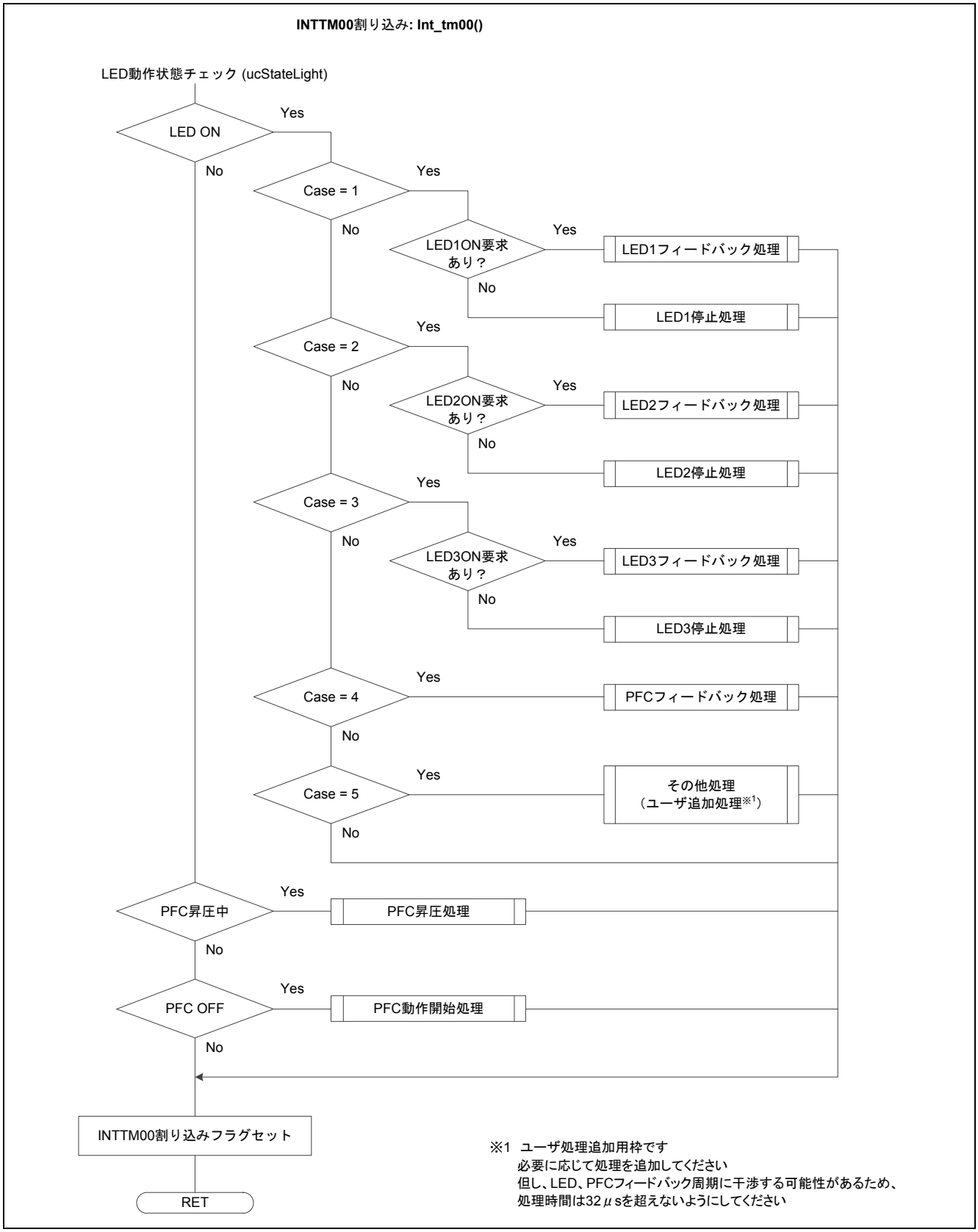


図 14 INTTM00 割り込みフローチャート

4.6 AC 電源入力状態判定

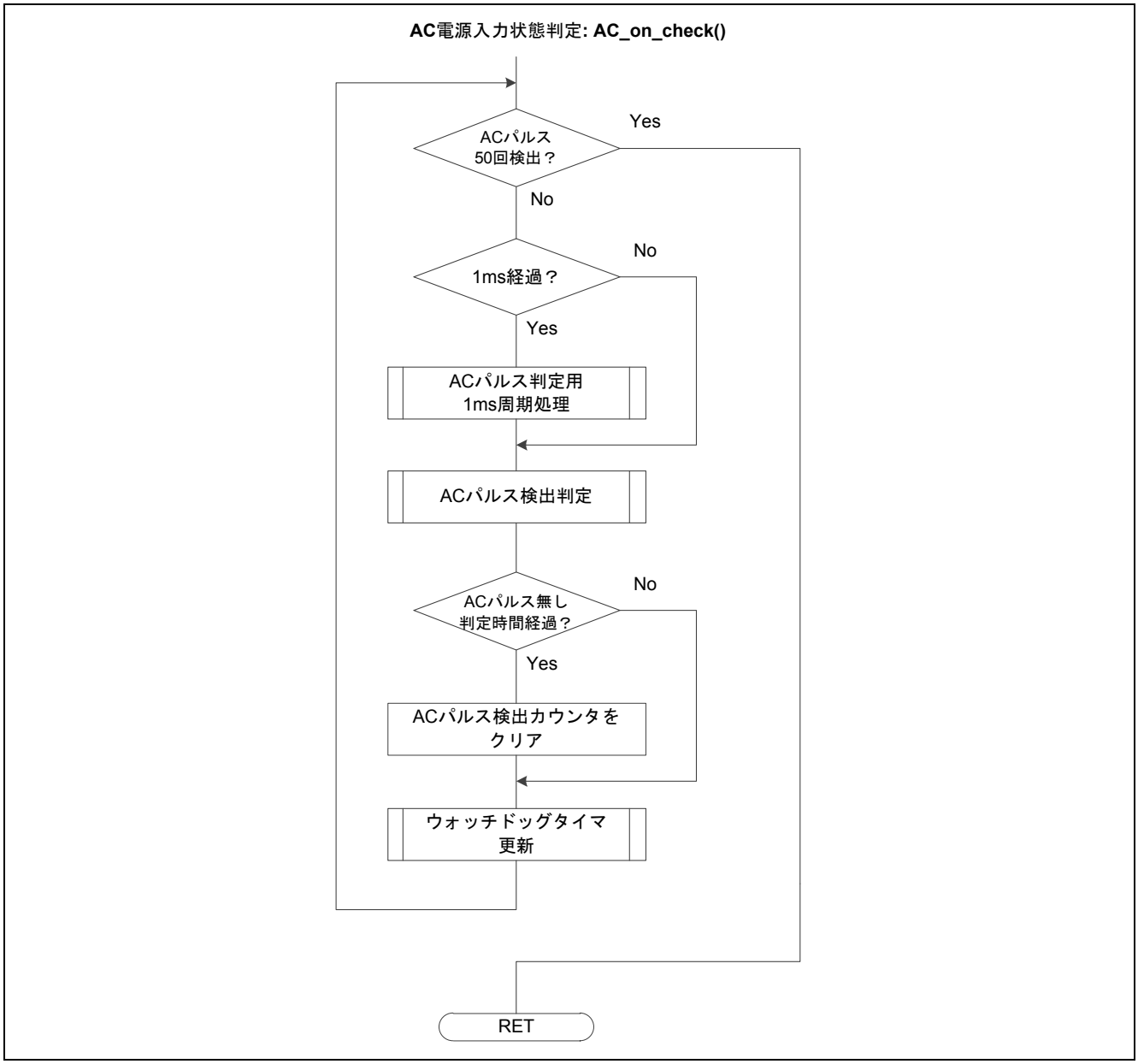


図 15　AC 電源入力状態判定フローチャート

4.7 オートチューニング処理

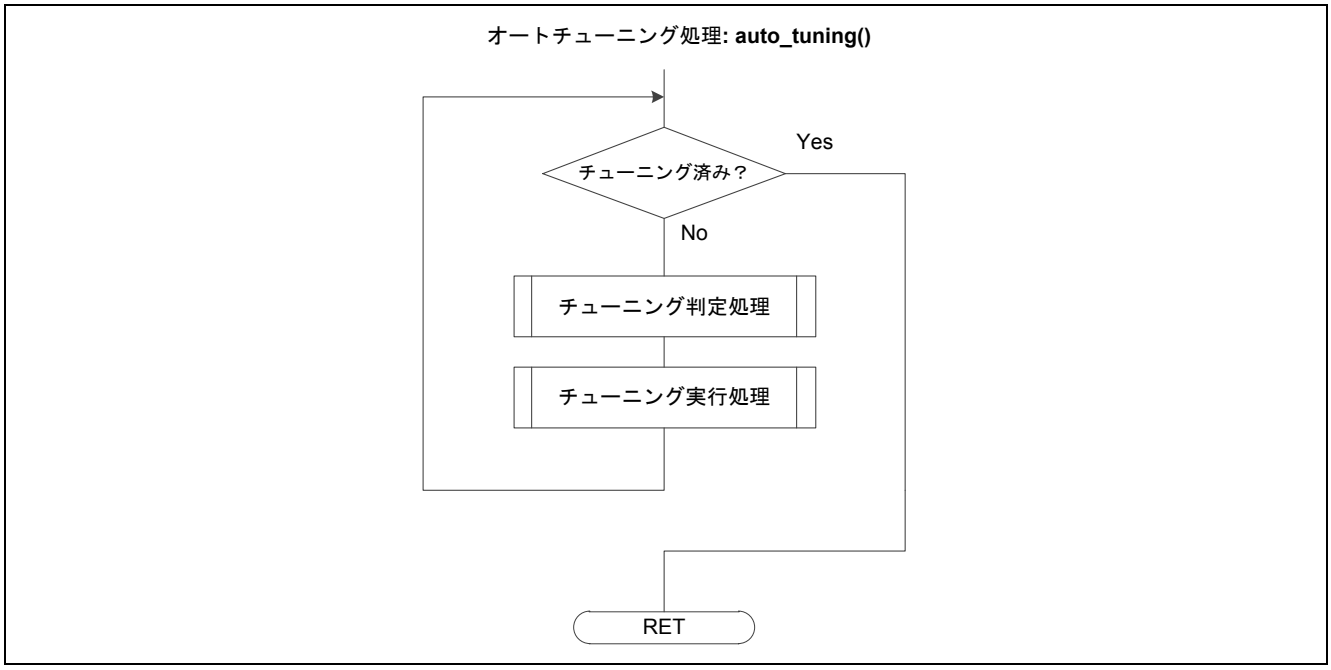


図 16 オートチューニング処理フローチャート

4.8 PFC 動作開始

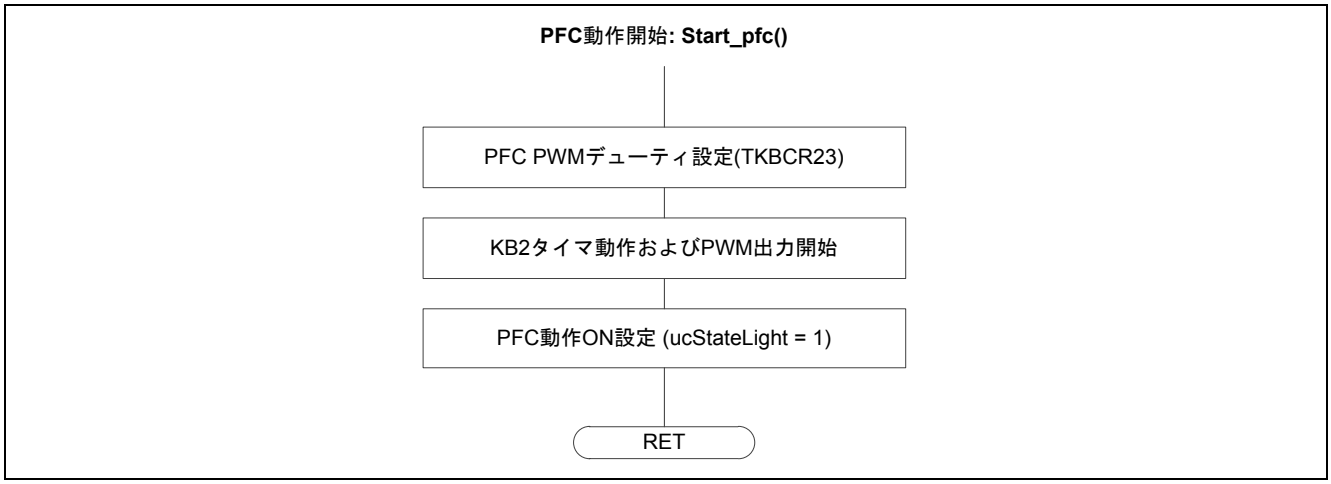


図 17 PFC 動作開始フローチャート

4.9 PFC 昇圧処理

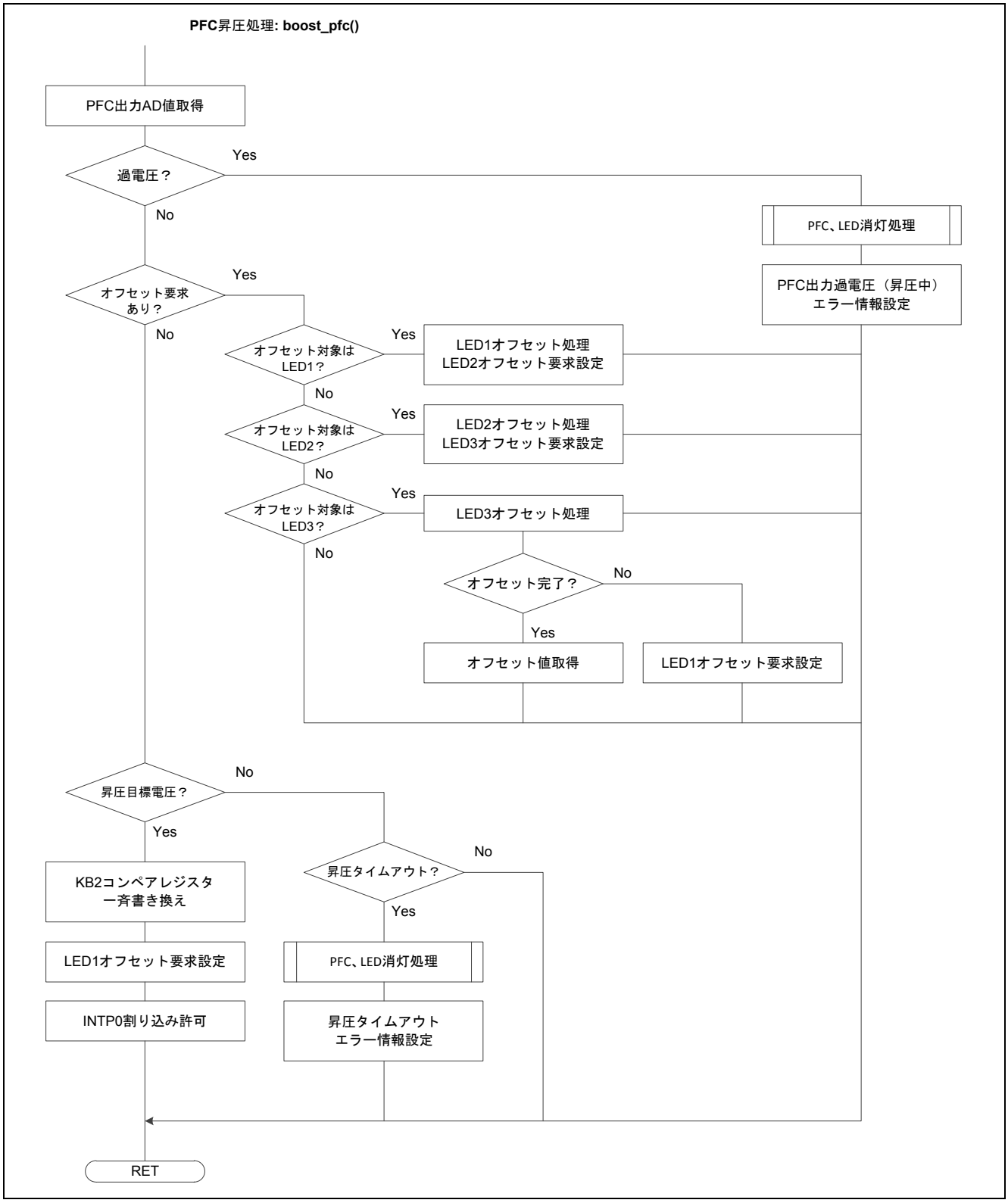


図 18　PFC 昇圧処理フローチャート

4.10 LED1 フィードバック処理

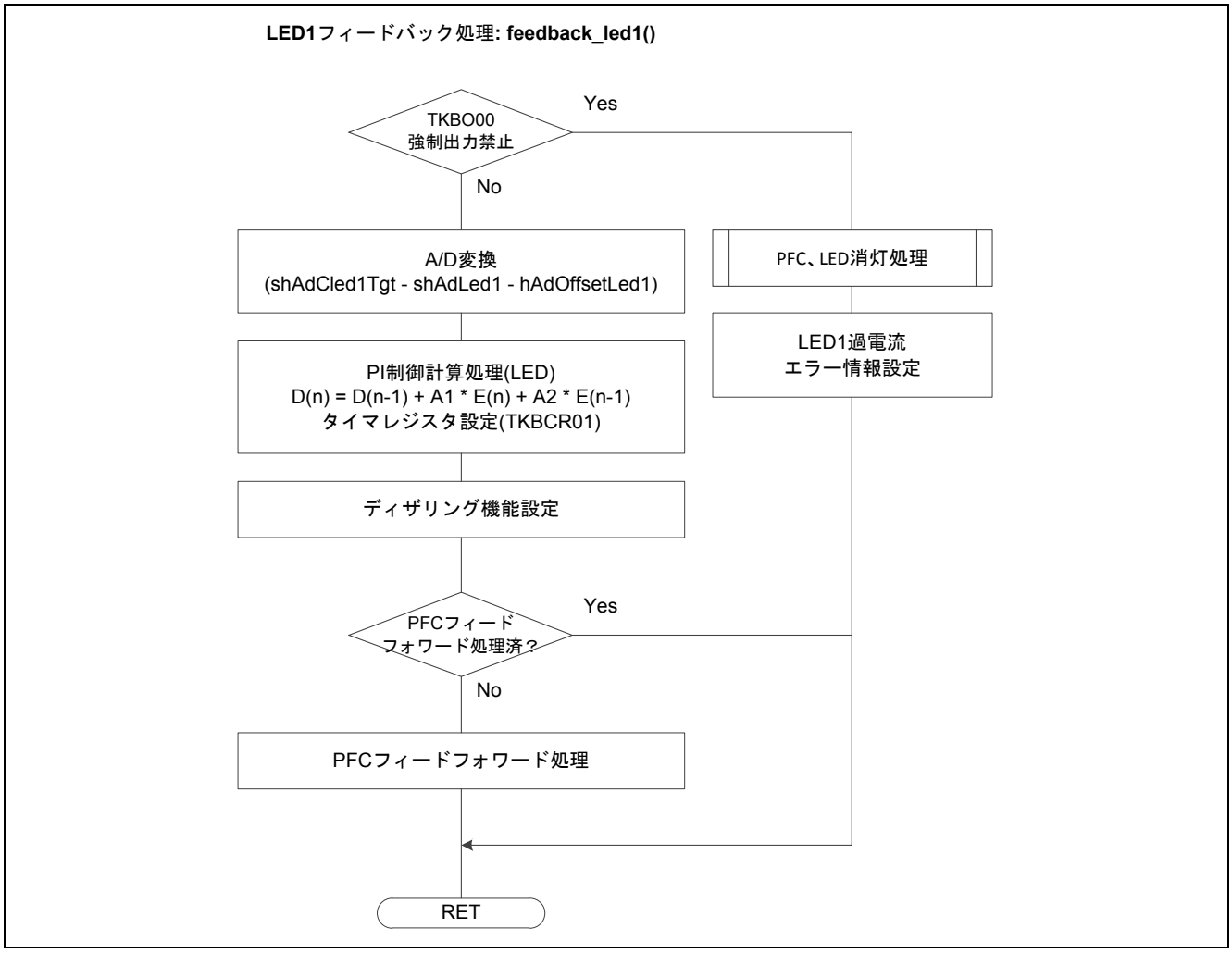


図 19 LED1 フィードバック処理フローチャート

4.11 LED2 フィードバック処理

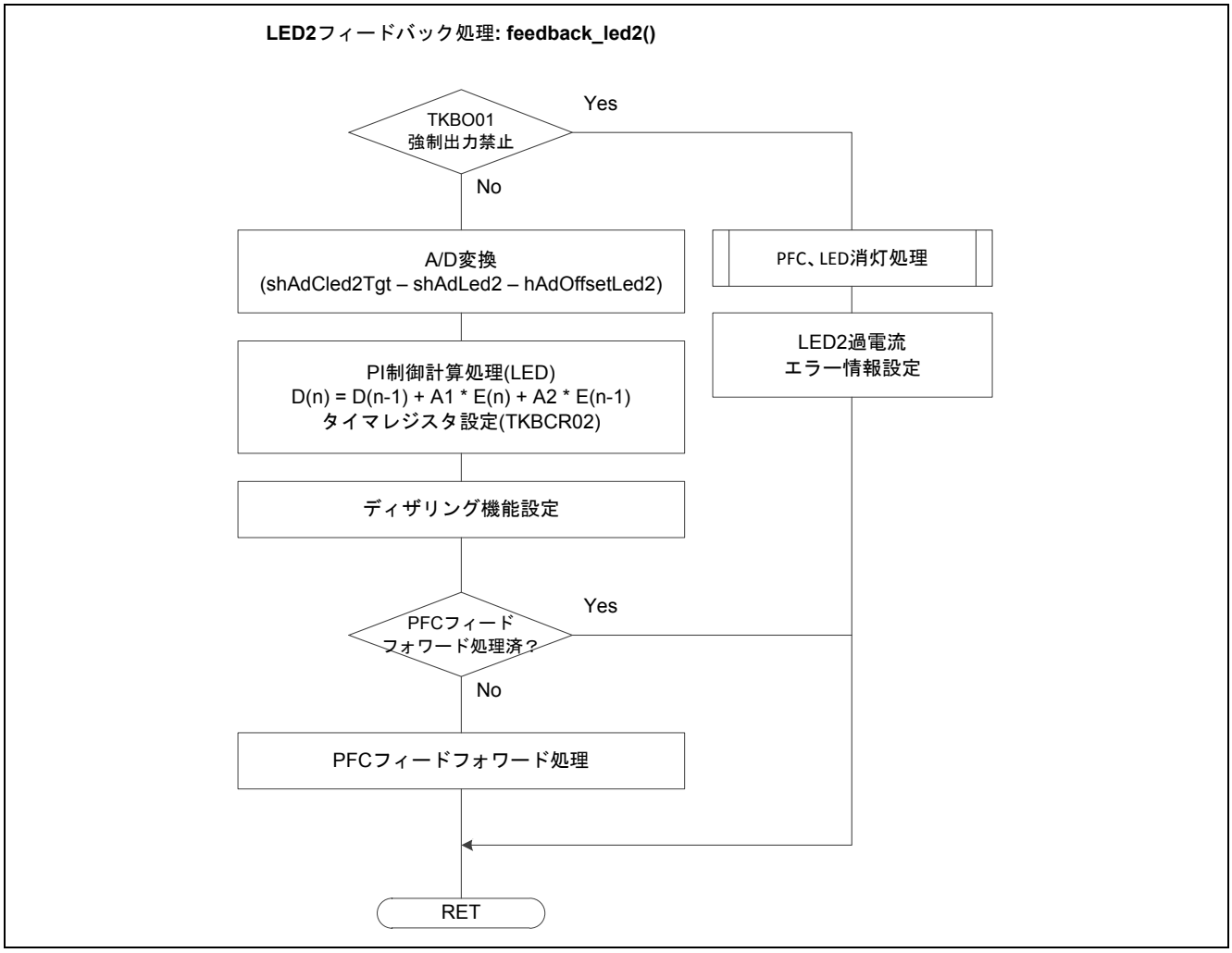


図 20 LED2 フィードバック処理フローチャート

4.12 LED3 フィードバック処理

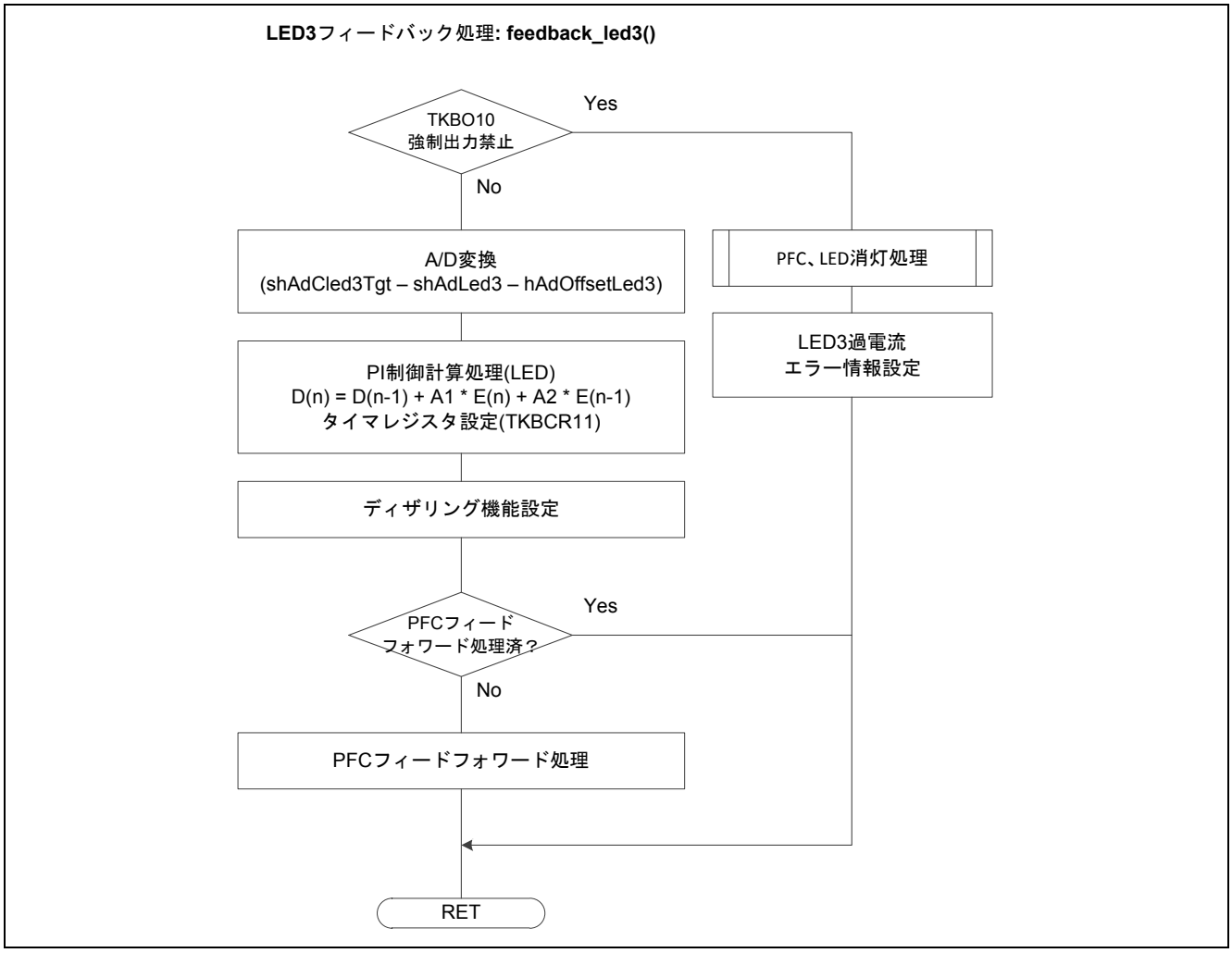


図 21 LED3 フィードバック処理フローチャート

4.13 PFC フィードバック処理

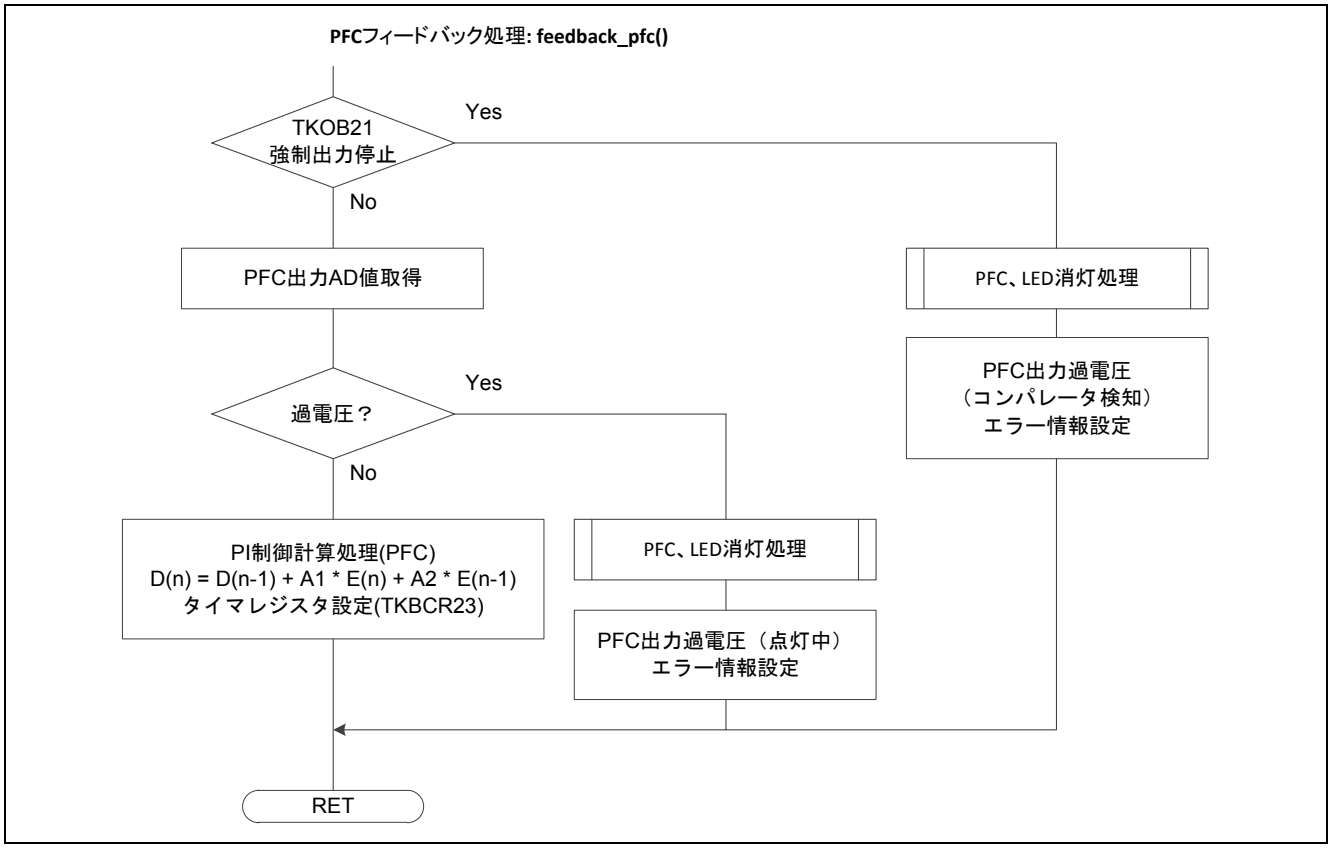


図 22 PFC フィードバック処理フローチャート

4.14 LED1 停止処理

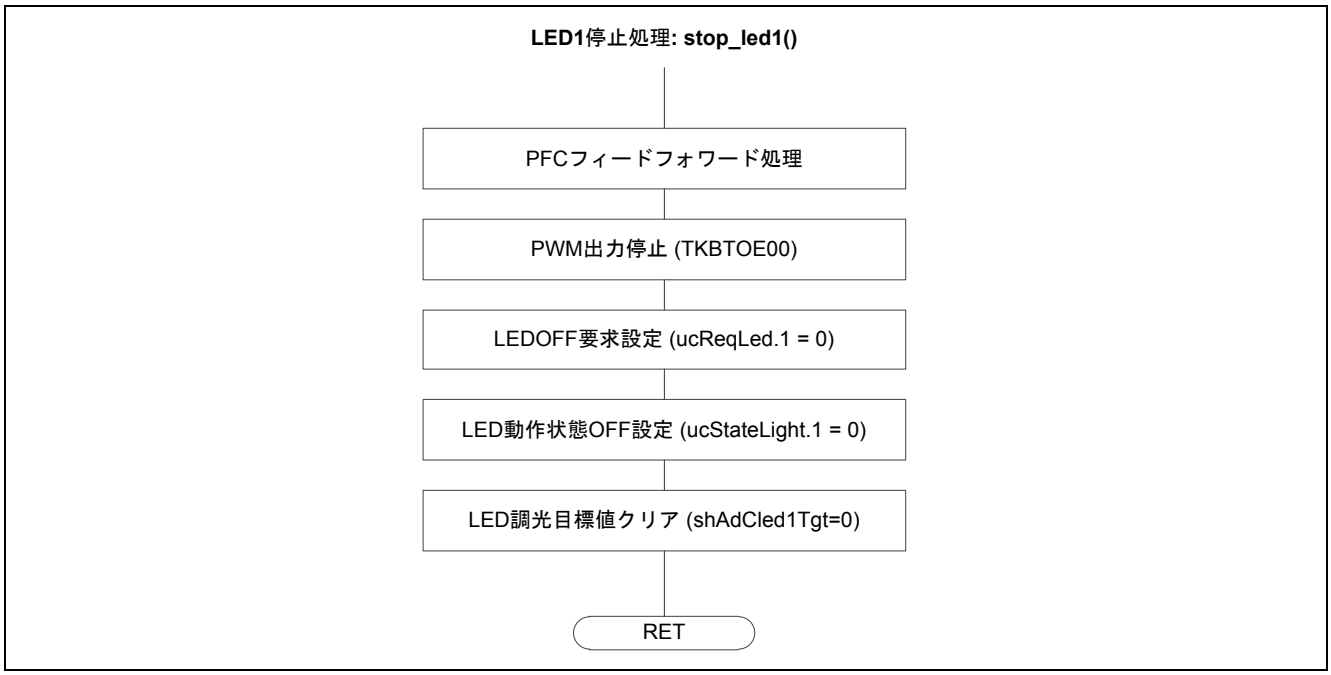


図 23 LED1 停止処理フローチャート

4.15 LED2 停止処理

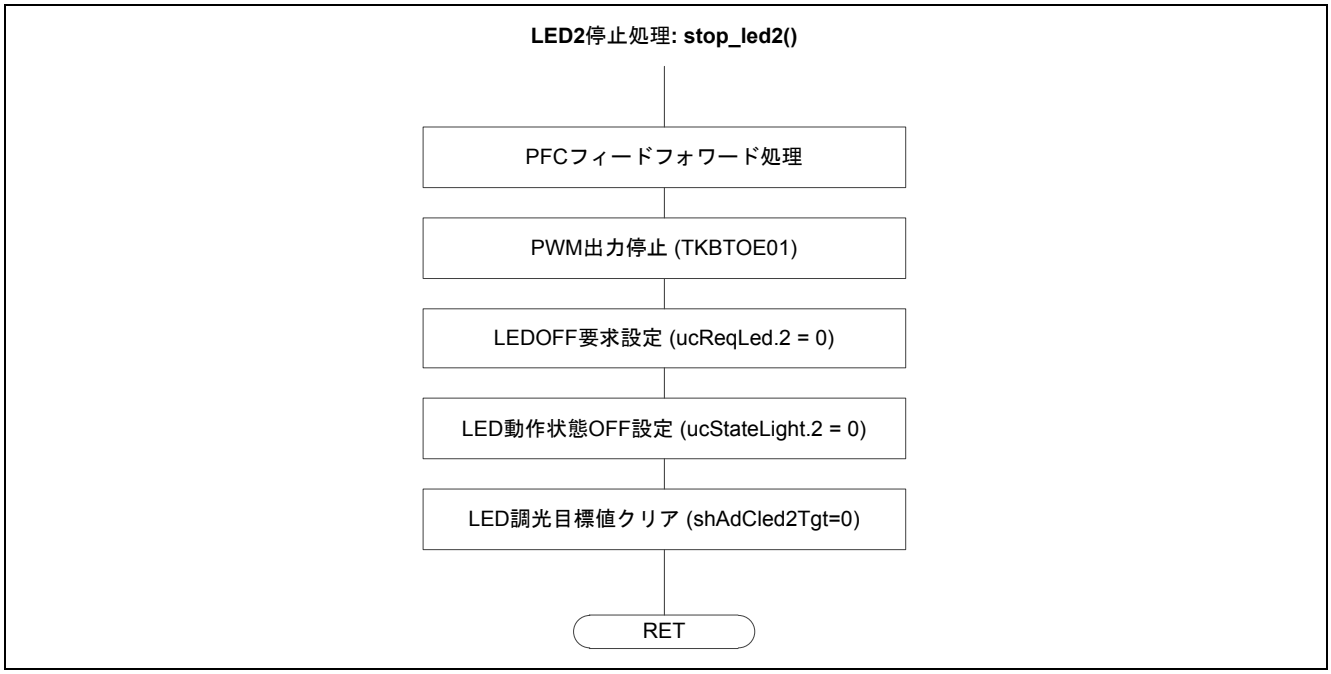


図 24 LED2 停止処理フローチャート

4.16 LED3 停止処理

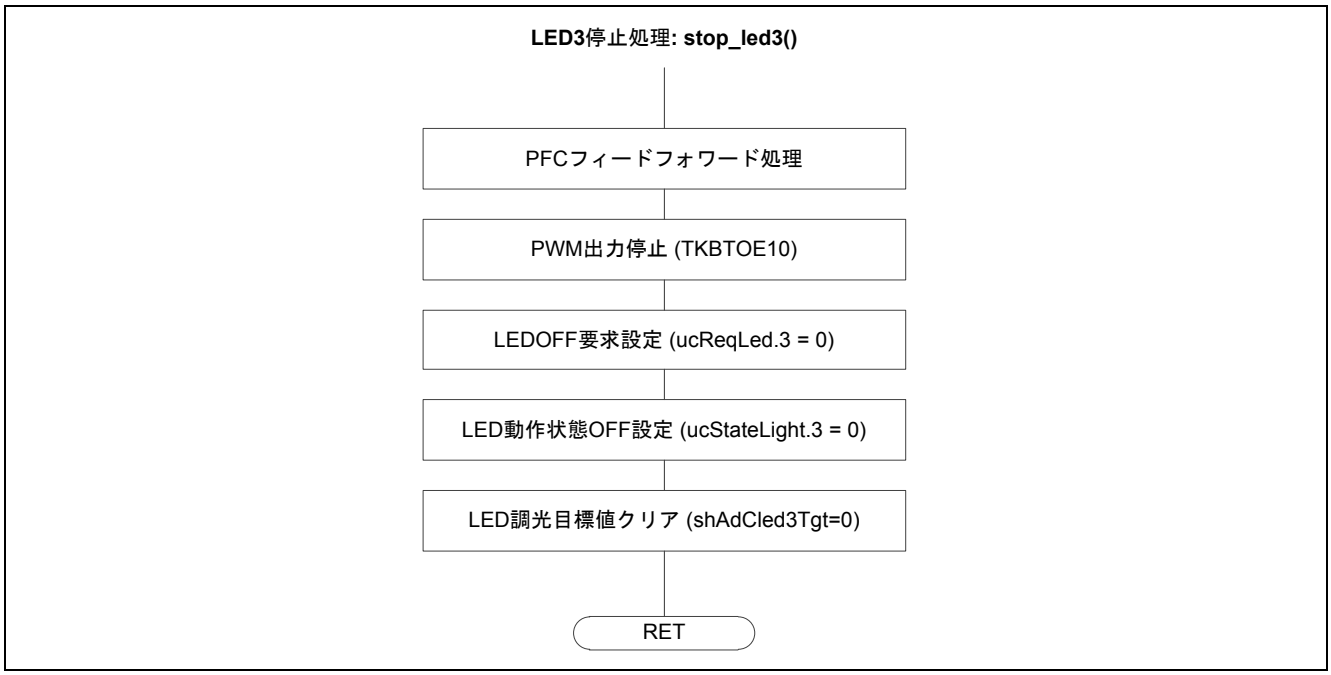


図 25 LED3 停止処理フローチャート

4.17 PFC、LED 消灯処理

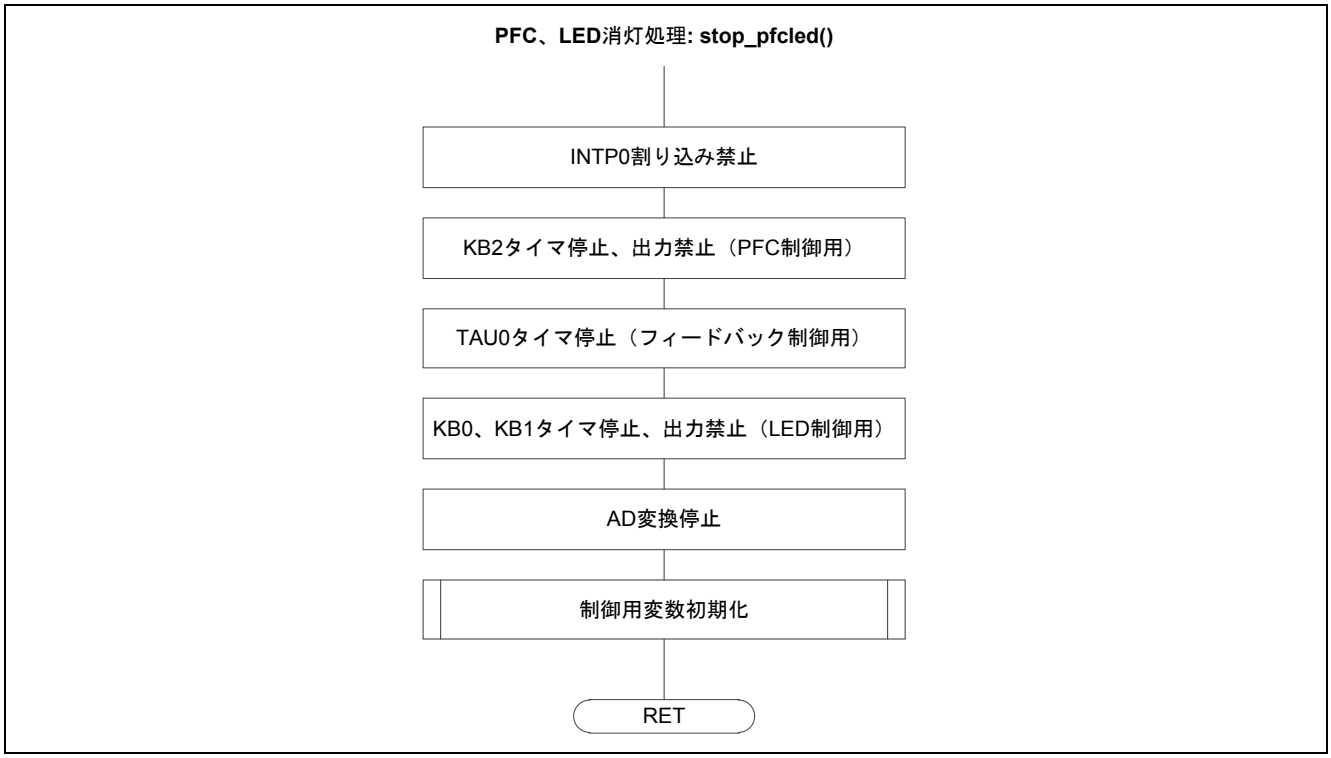


図 26 PFC、LED 消灯処理フローチャート

4.18 LED1 制御要求処理

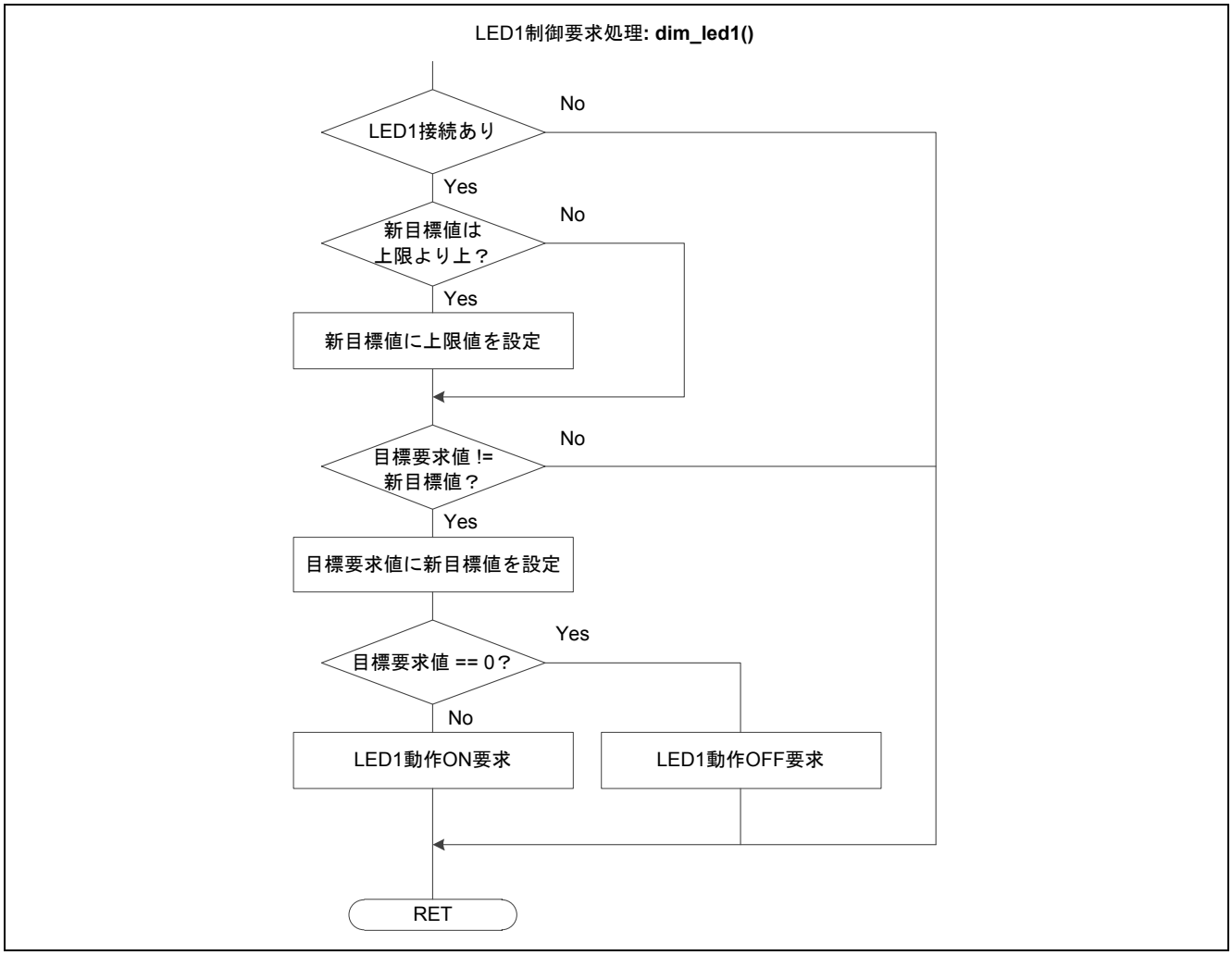


図 27 LED1 制御要求処理フローチャート

4.19 LED2 制御要求処理

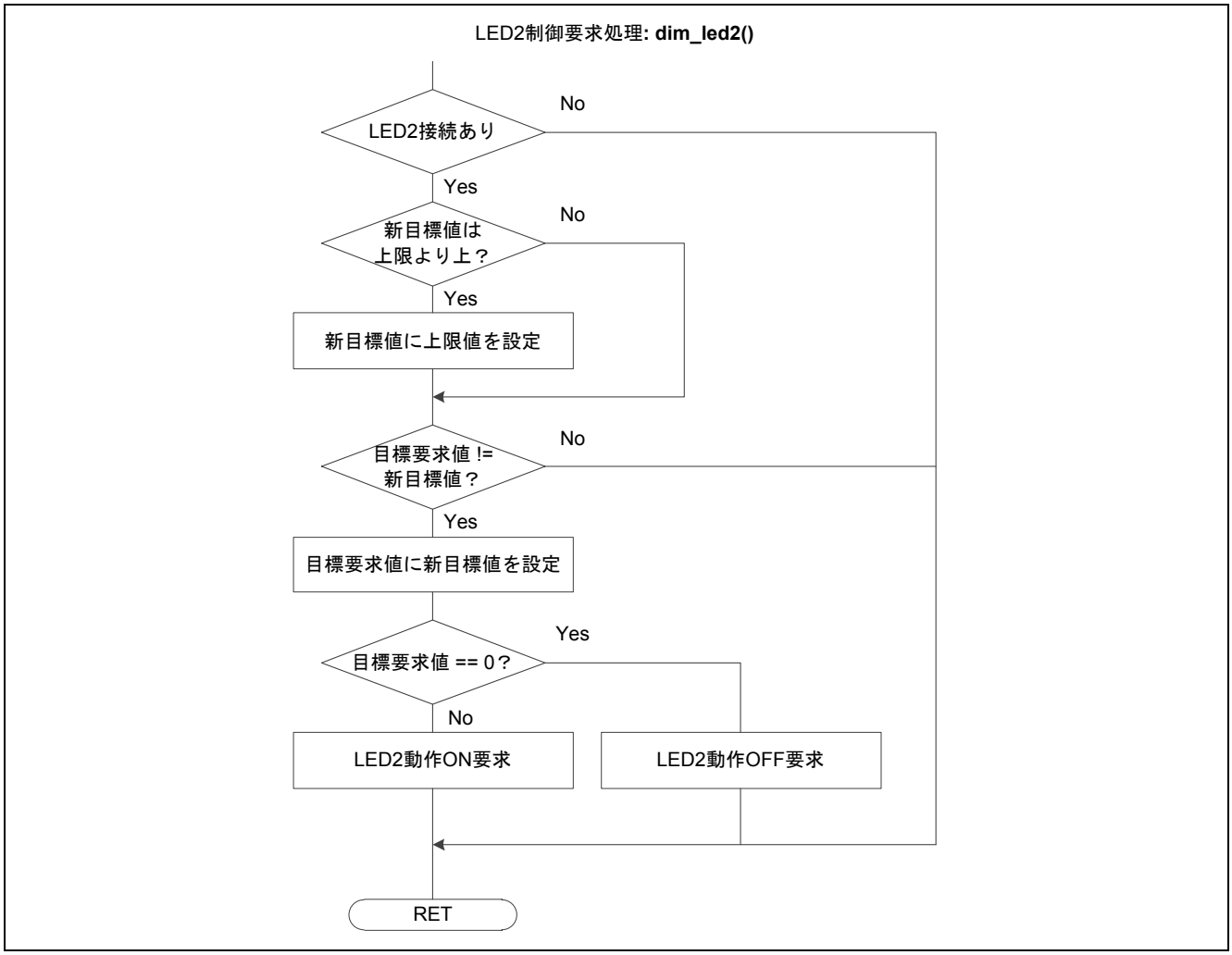


図 28 LED2 制御要求処理フローチャート

4.20 LED3 制御要求処理

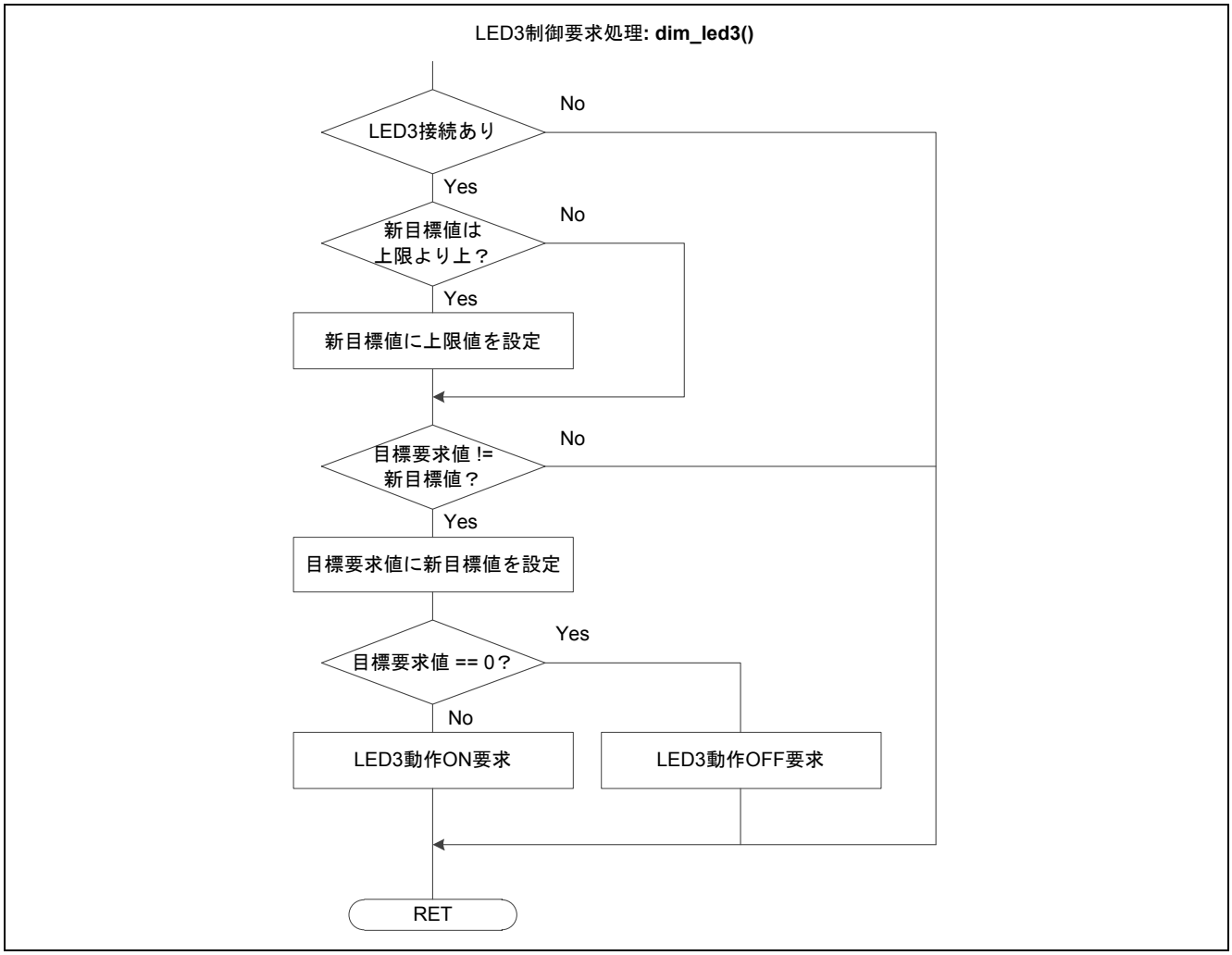


図 29 LED3 制御要求処理フローチャート

4.21 調光制御用タイマ動作開始

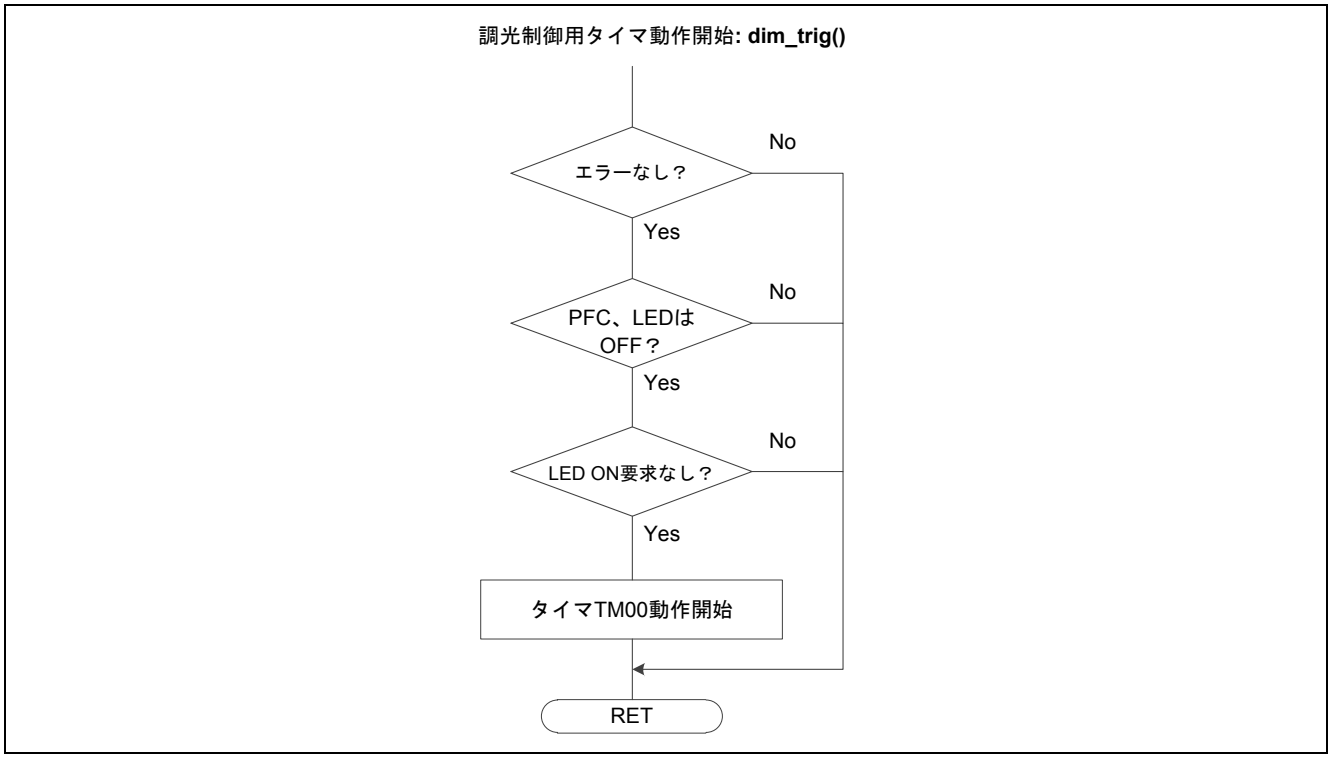


図 30 調光制御用タイマ動作開始フローチャート

4.22 SW 入力判定

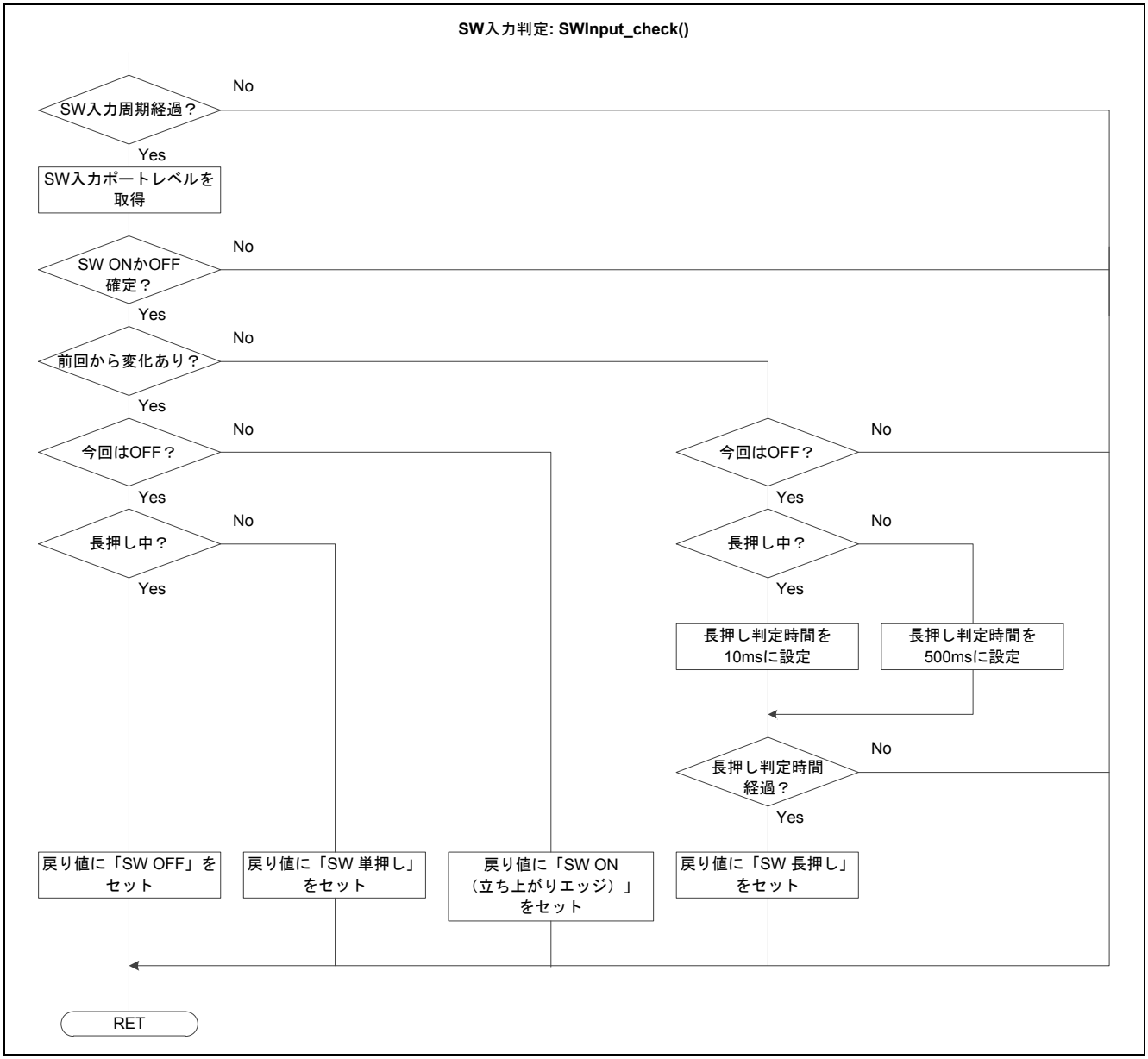


図 31 SW 入力判定フローチャート

4.23 SW による調光状態遷移判定

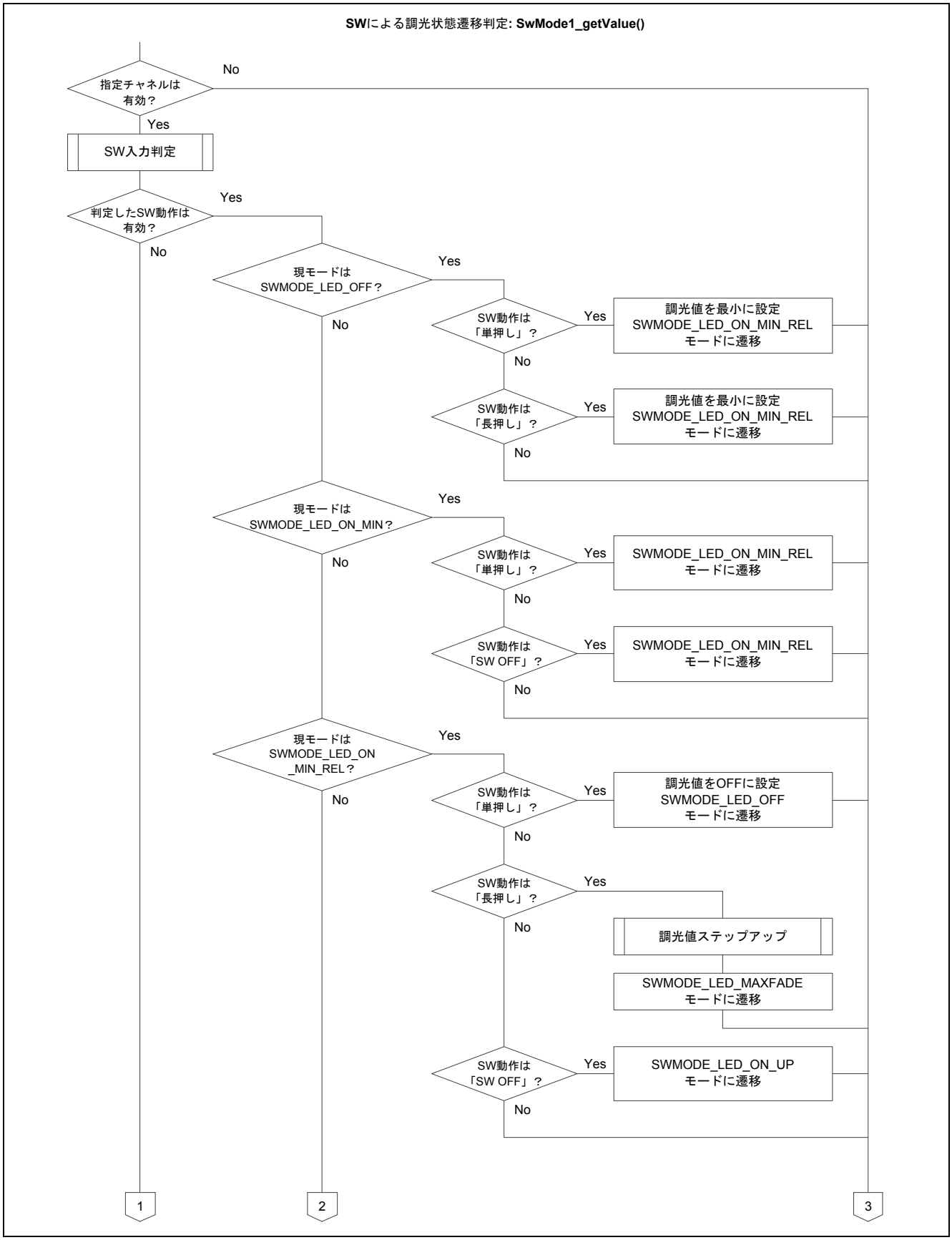


図 32 SW による調光状態遷移判定フローチャート(1)

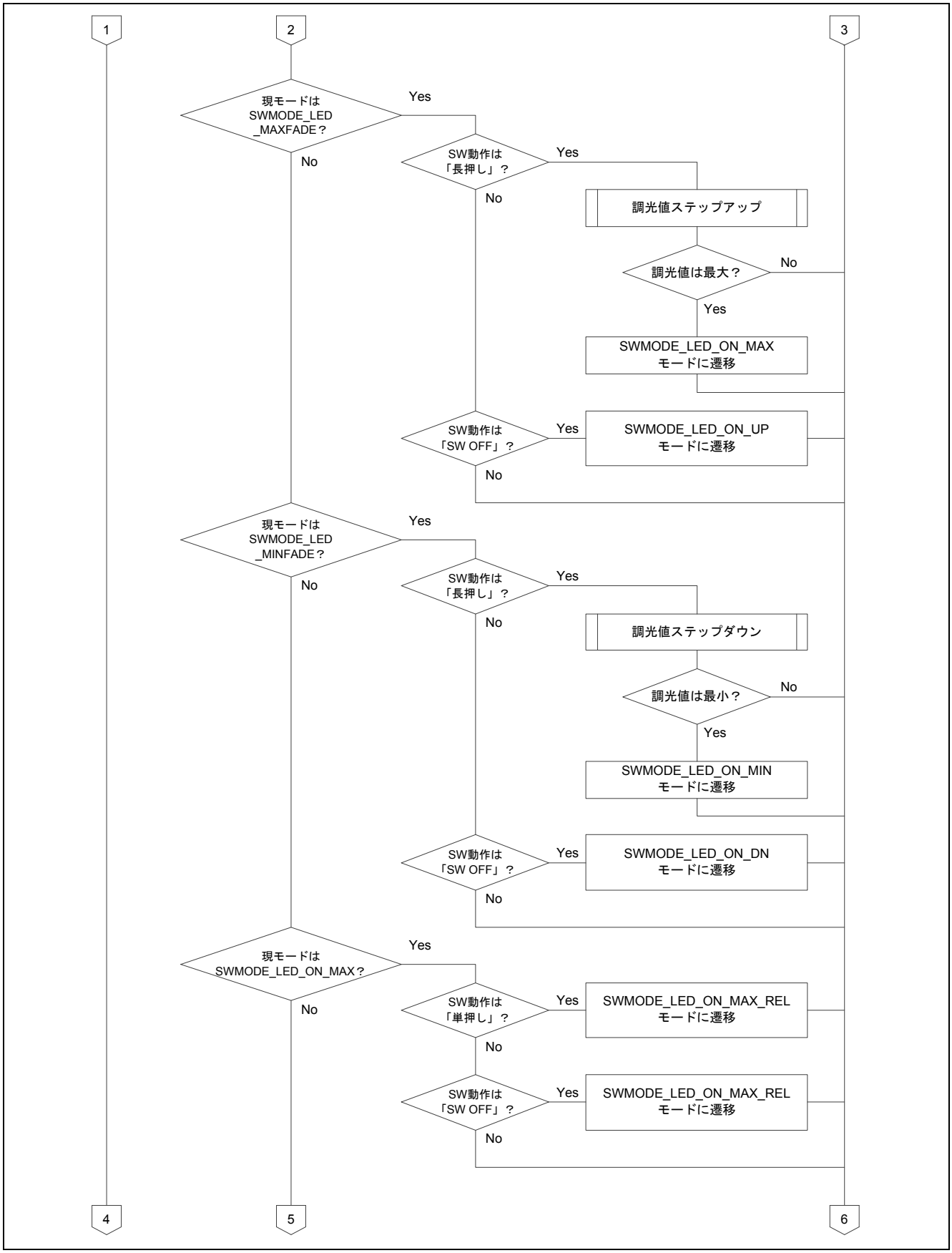


図 33　SW による調光状態遷移判定フローチャート(2)

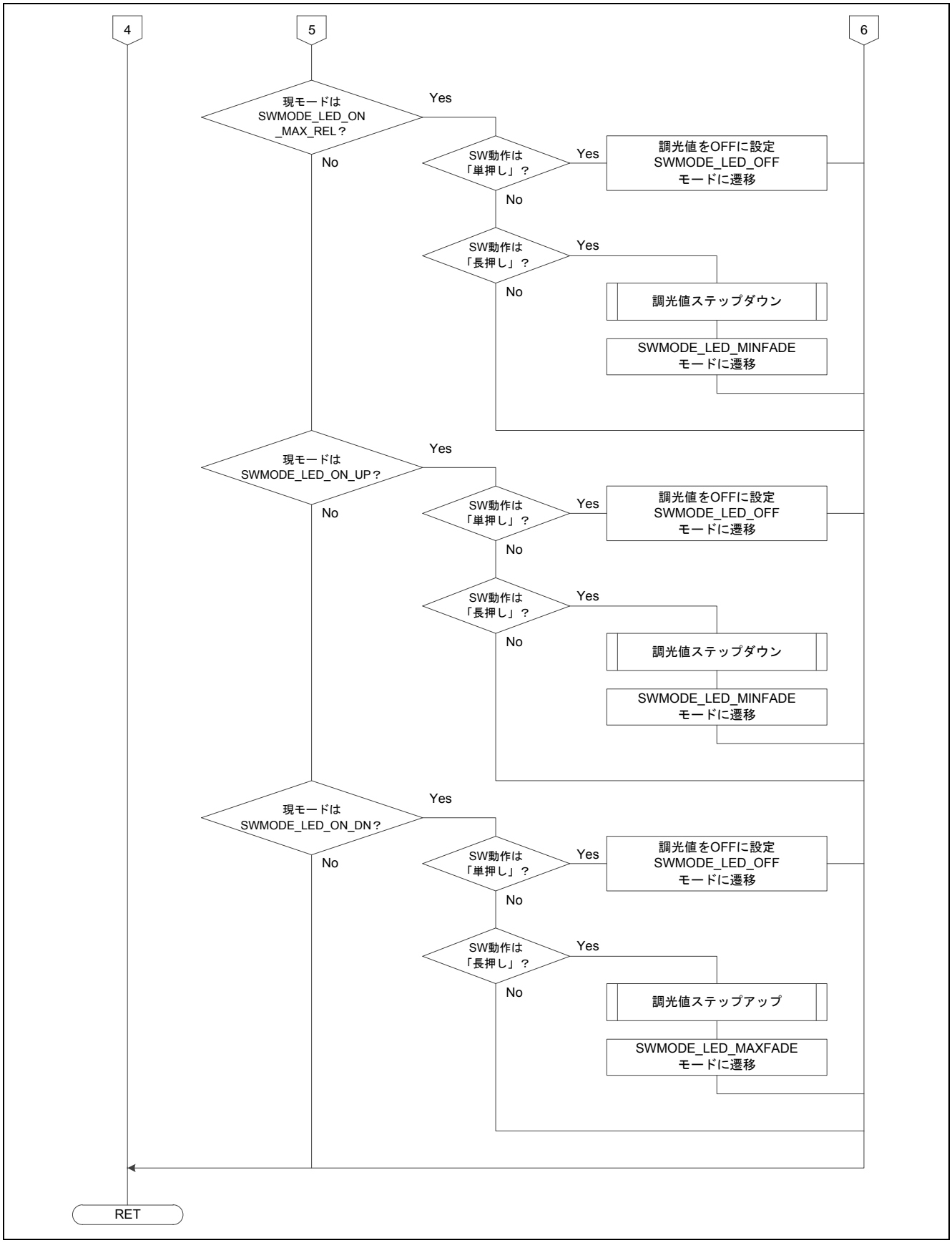


図 34　SW による調光状態遷移判定フローチャート(3)

5. 特性

- 周波数特性
  PFC 制御および LED 制御における周波数特性を示します。

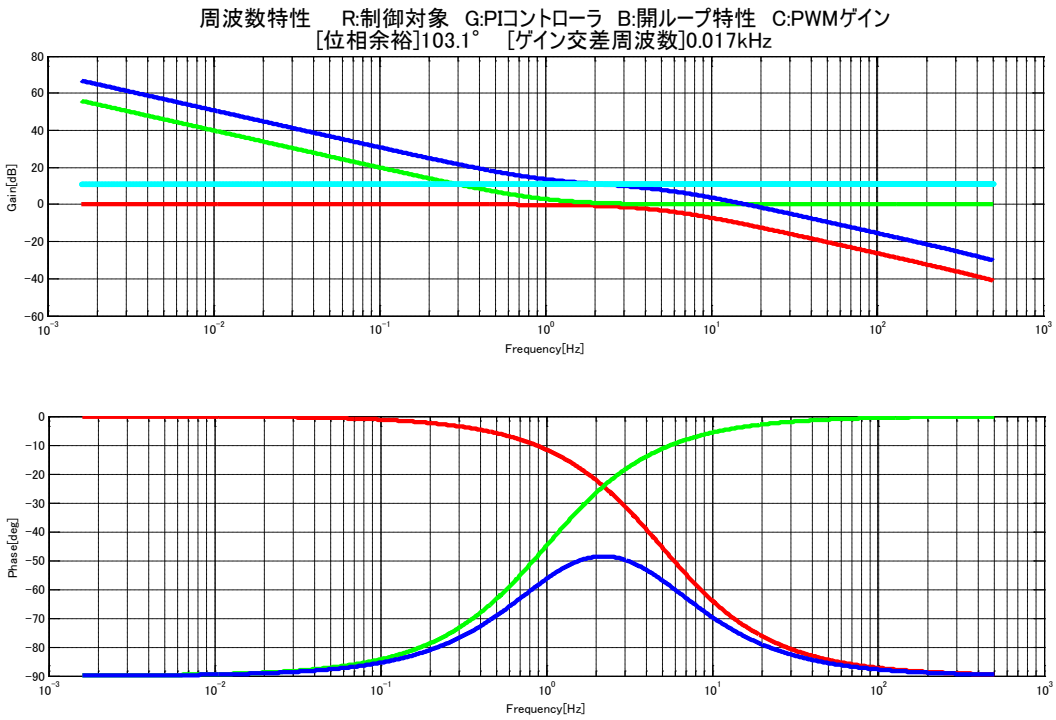


図 35　周波数特性（PFC 制御）

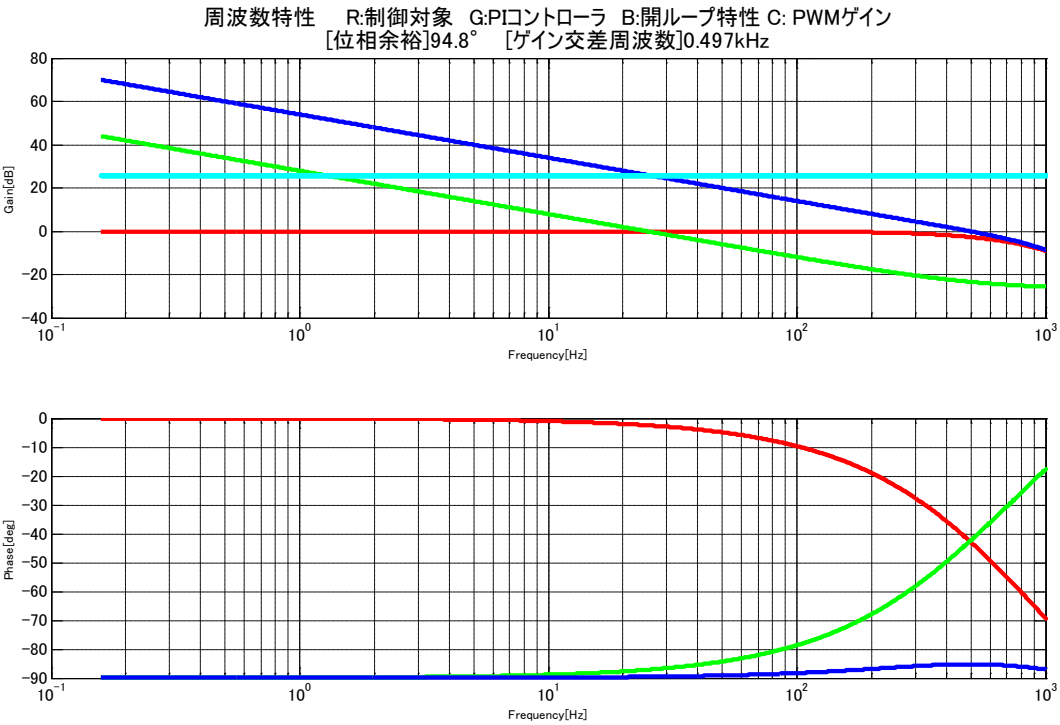


図 36　周波数特性（LED 制御）

- 高調波含有率
  図 37、図 38 に本サンプル・プログラムを使用して 100V、240VAC 入力定格 90W 出力を行った時の高調波含有率を示します。

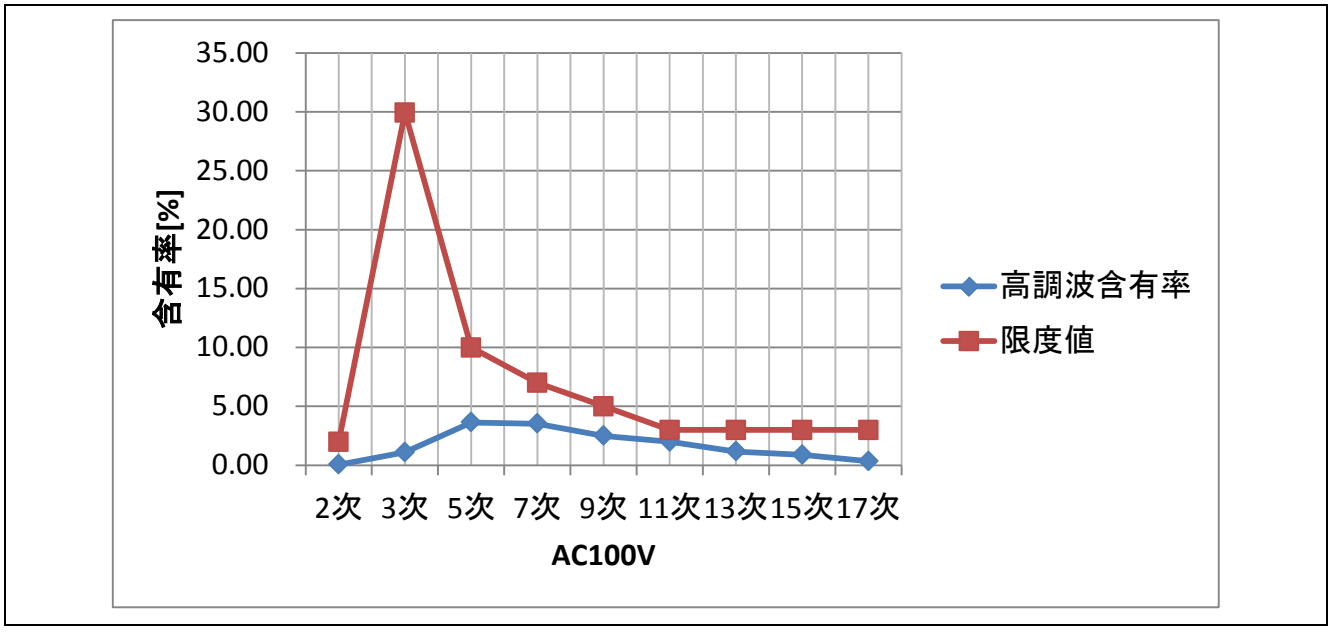


図 37　高調波含有率（AC100V 入力、90W）

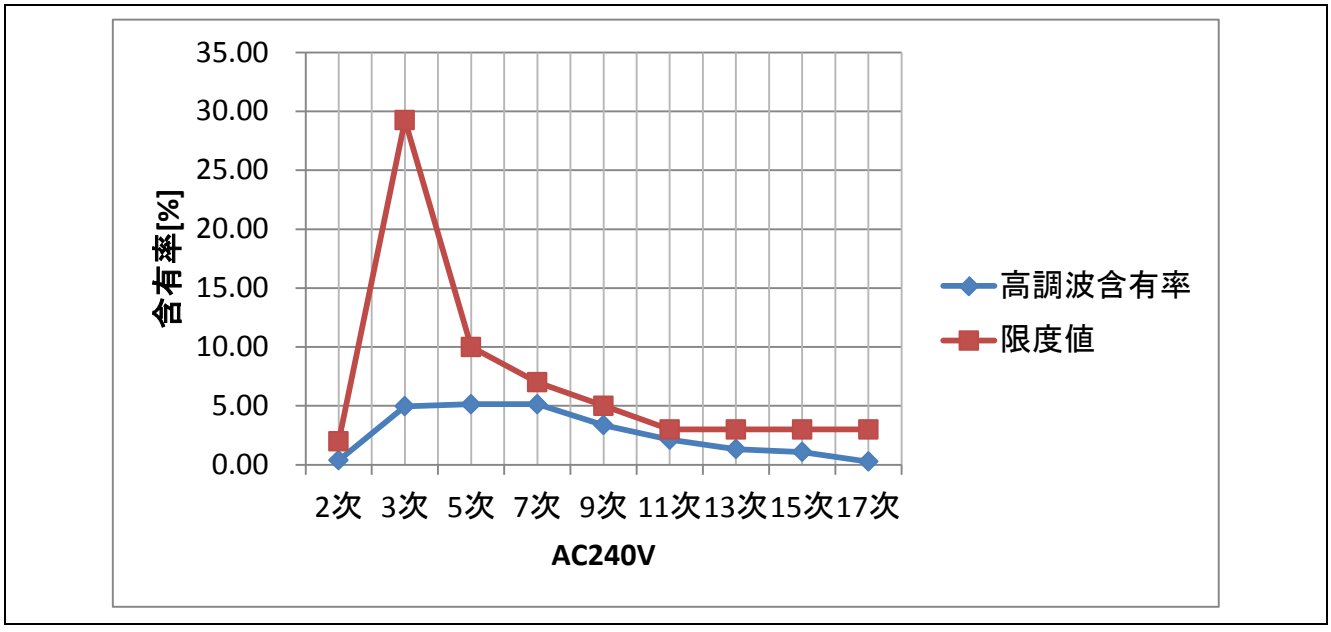


図 38　高調波含有率（AC240V 入力、90W）

- **待機電力**

  測定条件

  LED 制御ボード：SW5-2, 4, 6, SW7-1, 3, 5, 6 を ON
  RL78/I1A：ポート処理を行い、STOP モードに遷移
  入力電圧：AC 100V

  待機電力

  0.394W

- **雑音端子**

  CISPR15 の雑音端子規制値をクリア

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ
http://japan.renesas.com/

お問合せ先
http://japan.renesas.com/contact/

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2014.07.11	ー	初版発行

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】電源投入時は，製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレス（予約領域）のアクセス禁止

【注意】リザーブアドレス（予約領域）のアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。
プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。

リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違うと、内部ROM、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、
   各品質水準は、以下に示す用途に製品が使用されることを意図しております。
   標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、
   家電、工作機械、パーソナル機器、産業用ロボット等
   高品質水準：輸送機器（自動車、電車、船舶等）、交通用信号機器、
   防災・防犯装置、各種安全装置等
   当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。
注2. 本資料において使用されている「当社製品」とは、注１において定義された当社の開発、製造製品をいいます。

RENESAS

ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

http://www.renesas.com

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社　〒100-0004　千代田区大手町2-6-2（日本ビル）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：http://japan.renesas.com/contact/

Colophon 3.0