

RL78/G24

RL78/G24 による LED 制御

要旨

本アプリケーションノートは、RL78/G24 マイクロコントローラの機能を使用して LED 照明システムを制御する方法について説明します

対象読者

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。

動作確認デバイス

- 64 ピン : R7F101GLG

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

目次

1. はじめに	3
2. RL78/G24 を使用した LED 制御の概要	4
2.1 LED システム制御に関する RL78/G24 の特徴	4
2.2 システム・ブロック図	5
2.3 RL78/G24 の端子機能	6
3. LED 制御ソフトウェア	7
3.1 ファイル構成	7
3.2 内蔵周辺機能の初期化	10
3.3 定電流および調光制御	11
3.4 PI 制御によるフィードバック方法	14
3.5 PI 制御式の係数の計算	17
3.6 ソフトウェア・フロー・チャート	20
4. RL78/G24 DC/DC LED 制御評価ボードの起動方法	23
付録 A 写真	26
付録 B 回路図	27
改訂記録	29

1. はじめに

本アプリケーションノートは、RL78/G24 マイクロコントローラを使用した高輝度 LED 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 チャンネルの LED の独立制御、DALI 通信を使用した個別調光制御をおこなっています。また定電流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、RL78/G24 DC/DC LED 制御評価ボード（RTK7RLG240P00000BJ）を使用して評価することができます。

RL78/G24 DC/DC LED 制御評価ボードの端子構成は回路図にて確認ください。

2. RL78/G24 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/G24 の特徴

RL78/G24 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/G24 マイクロコントローラでは、16 ビット・タイマ KB0、KB1、KB2 とタイマ RD2 のタイマ出力機能を使用して、最大 6 チャンネルの 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 0.65 ns に上げることができます。
- また、RL78/G24 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/G24 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制的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/G24 マイクロコントローラは、DALI 通信機能をサポートし、DALI 通信規格(IEC 62386-101 Edition 1.0/2.0/2.1)に準拠した送受信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 負荷が軽減されています。
- RL78/G24 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・ユニットのチャンネル 7 の入力信号パルス幅測定機能を使用して、RxD0 端子のブレイク時間の立ち下りエッジを検出しその幅（88 μ s 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（8 μ s～1 s 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 Between Slots 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 タイマ RD2 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 チャンネルの 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 2-1 に、RL78/G24 DC/DC LED 制御評価ボード（RTK7RLG240P00000BJ ボード）のシステム・ブロック図を示します。本 LED 照明システムは、16 ビット・タイマ KB0 および KB1 を使用して 3 チャンネル LED 制御を実行し、以下を使用して調光制御を行います。

- ・ オンボード可変抵抗器（12 ビット A/D コンバータを使用）によるボリューム制御。
- ・ DALI 通信（DALI インターフェースを使用）。この調光についてのみ、本アプリケーションノートで説明します。
- ・ DMX512 通信（UART0 インターフェースを使用）
- ・ IR 通信（パルス間隔測定モードでタイマ・アレイ・ユニット・チャンネルを使用）

RL78/G24 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。

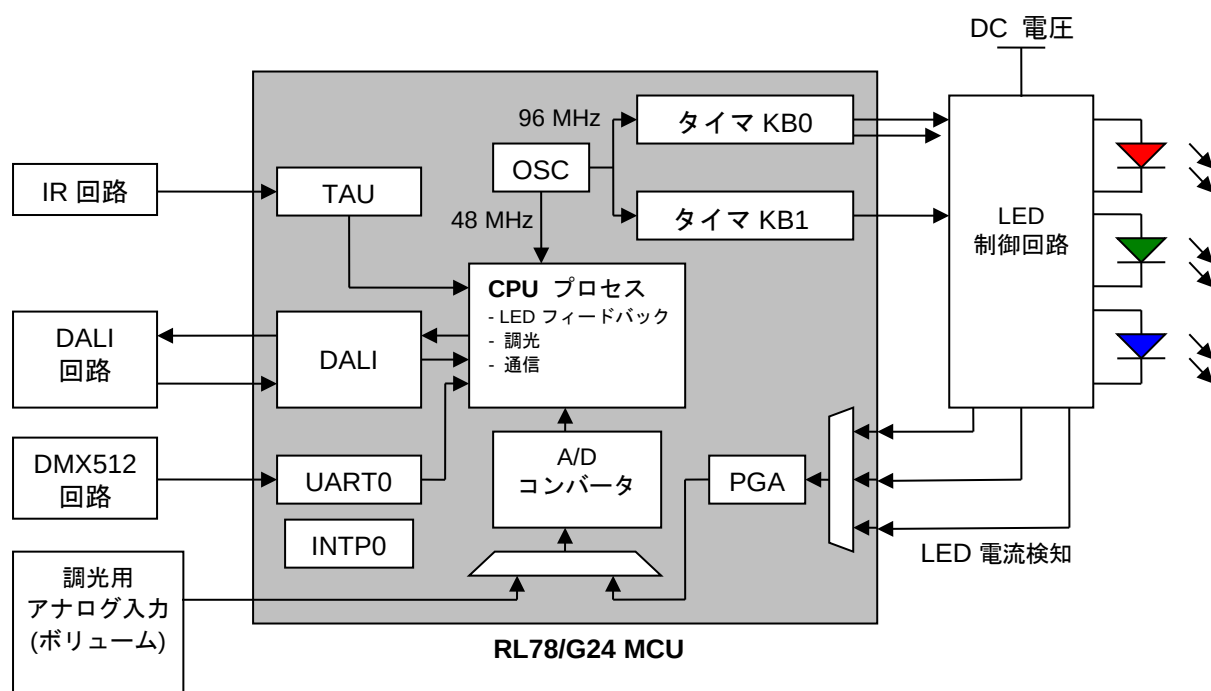


図 2-1 RL78/G24DC/DC LED 制御評価ボードのブロック図

2.3 RL78/G24 の端子機能

以下の表に、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。

機能	機能名	端子名	I/O	説明
LED 制御	TKBO00	P12	O	LED1 の PWM 出力
	TKBO01	P13	O	LED2 の PWM 出力
	TKBO10	P14	O	LED3 の PWM 出力
	ANI19	P120	I	LED1 のフィードバック電流入力
	ANI29	P00	I	LED2 のフィードバック電流入力
	ANI30	P01	I	LED3 のフィードバック電流入力
	ANI2	P22	I	LED2 調光用ボリューム入力
	ANI18	P147	I	LED3 調光用ボリューム入力
	ANI28	P146	I	LED1 調光用ボリューム入力
通信	DALITxD0	P02	O	DALI送信出力
	DALIRxD0	P03	I	DALI受信入力
	TxD2	P77	O	DMX512送信出力
	RxD2	P76	I	DMX512受信入力
	TI02	P17	I	IR 受信入力

表 2-1 端子機能

3. LED 制御ソフトウェア

本章では、ルネサス エレクトロニクスホームページからダウンロードするファイルのファイル構成、使用する RL78/G24 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、本サンプル・プログラムの全体的な動作（定電流と調光制御）概要と、フィードバック制御用に実装された PI 制御については包括的な説明とフロー・チャートを示します。

3.1 ファイル構成

本サンプル・プログラムは、以下のように複数のファイルに分割されています。

プロジェクト・フォルダの構成

./	
DALI102_207_209_sample.mtpj :	プロジェク・トファイル ^{注1}
DALI102_207_209_sample.scfg :	スマート・コンフィグレータ・ファイル ^{注1}
—App :	Application フォルダ
—DefaultBuild :	ビルド成果物フォルダ
DALI102_207_209_sample.mot :	ヘキサ・ファイル ^{注1}
—Driver :	Driver フォルダ
—Library :	Library フォルダ
—src :	スマート・コンフィグレータ生成フォルダ ^{注3}
—Utility :	Utility フォルダ

注1 102+207+209 規格のサンプル・プロジェクトの例です。対応する規格に応じてファイル名が異なります。

注2 コードを上書きしている箇所があるため再生成しないでください。

App フォルダの構成

ハードウェア依存の少ないコードを実装します。

App	
r_cg.c :	Control Gear の実装
r_cg.h :	Control Gear の実装
r_common.h :	共通ヘッダファイル
r_debug.c :	UART を使用したデバッグ機能
r_debug.h :	UART を使用したデバッグ機能
r_lamp.c or r_lamp_tc.c :	DALI 規格に従ったランプの調光/調色機能 ^{注1}
r_lamp.h or r_lamp_tc.h :	DALI 規格に従ったランプの調光/調色機能 ^{注1}
r_main.c :	main 関数
r_memory_bank.c :	Memory Bank 制御
r_memory_bank.h :	Memory Bank 制御
r_memory_banks.c :	複数 Memory Bank 制御
r_memory_banks.h :	複数 Memory Bank 制御
r_nvm.c :	RFD を使用した NVM 処理
r_nvm.h :	RFD を使用した NVM 処理
r_random.c :	Xorshift による疑似乱数生成
r_random.h :	Xorshift による疑似乱数生成
r_unit0_memory_bank.c :	Logical Unit 0 の Memory Bank 定義
r_unit0_memory_bank.h :	Logical Unit 0 の Memory Bank 定義
r_unit1_memory_bank.c :	Logical Unit 1 の Memory Bank 定義
r_unit1_memory_bank.h :	Logical Unit 1 の Memory Bank 定義
r_unit2_memory_bank.c :	Logical Unit 2 の Memory Bank 定義
r_unit2_memory_bank.h :	Logical Unit 2 の Memory Bank 定義

注1 209 規格(調色)に対応するサンプル・プロジェクトのみ r_lamp_tc、未対応のプロジェクトは r_lamp が該当。

Driver フォルダの構成

ハードウェア依存の大きいコードを実装します。

Driver	
r_dali101.c :	DALI ドライバのインターフェース
r_dali101.h :	DALI ドライバのインターフェース
r_dali101_bft.c :	DALI ドライバ/Backward Frame Transmitter の実装
r_dali101_bft.h :	DALI ドライバ/Backward Frame Transmitter の実装
r_dali101_common.h :	DALI ドライバ/共通ヘッダファイル
r_dali101_rx.c :	DALI ドライバ/Receiver の実装
r_dali101_rx.h :	DALI ドライバ/Receiver の実装
r_led.c :	LED ドライバのインターフェース、フィードバック処理
r_led.h :	LED ドライバのインターフェース、フィードバック処理
r_led1.c :	LED ドライバ/LED1 の実装
r_led1.h :	LED ドライバ/LED1 の実装
r_led2.c :	LED ドライバ/LED2 の実装
r_led2.h :	LED ドライバ/LED2 の実装
r_led3.c :	LED ドライバ/LED3 の実装
r_led3.h :	LED ドライバ/LED3 の実装
r_port.c :	GPIO の読み書き
r_port.h :	GPIO の読み書き
r_trng.c :	真性乱数生成器
r_trng.h :	真性乱数生成器

Library フォルダの構成

既存のライブラリやコードを配置します。

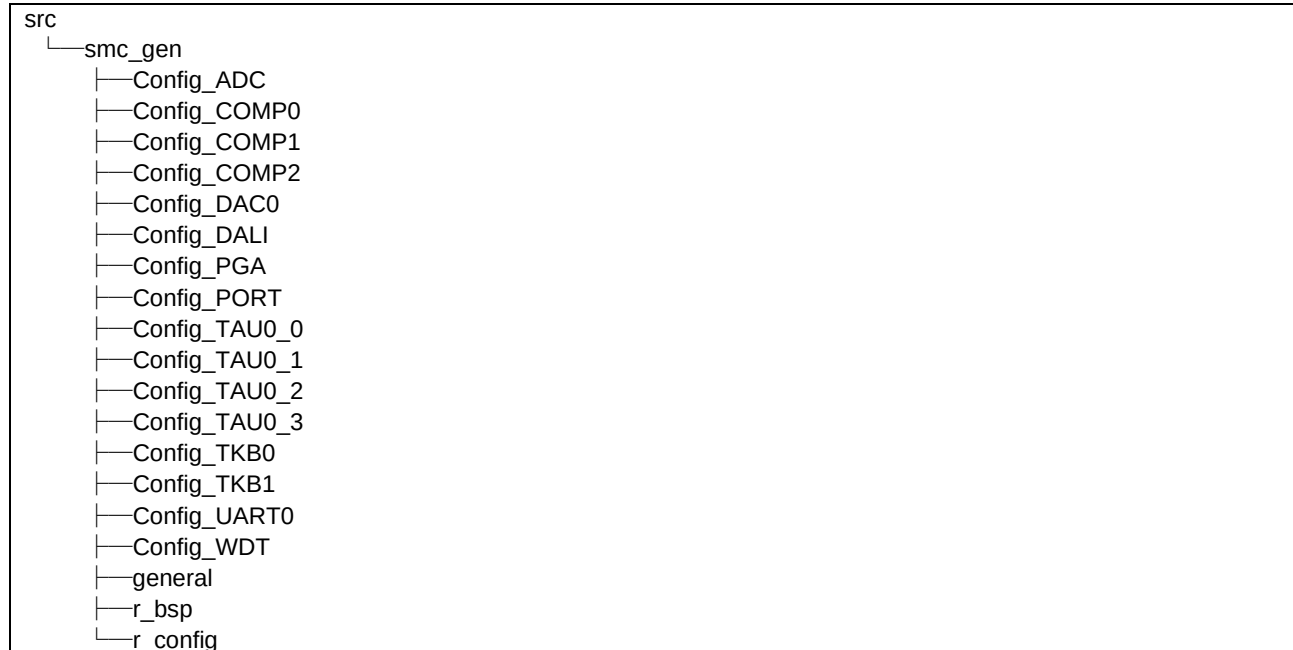
Library	
├─DALI102 :	DALI102 ライブラリを配置するフォルダ ^{注1}
├─DALI207 :	DALI207 ライブラリを配置するフォルダ ^{注1}
├─DALI209 :	DALI209 ライブラリを配置するフォルダ ^{注2}
└─RFD :	RFD を配置するフォルダ ^{注2}

注1 ユーザが各ライブラリを配置する必要があります。

注2 Renesas Flash Driver RL78 Type01 の略。ユーザが include、source、userown フォルダを配置する必要があります。(sample フォルダは不要)

src フォルダの構成

スマート・コンフィグレータで生成されたコードが配置されます。

**Utility フォルダの構成**

プロジェクト全体で 사용할 수 있는 보조적인 코드를 실장합니다.

Utility		
r_timer16.c :	16bit 소프트웨어 타이머	
r_timer16.h :	16bit 소프트웨어 타이머	
r_user_typedefs.h :	유저 정의의 typedef	

3.2 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/G24 マイクロコントローラ内蔵周辺機能を使用します。

- クロック設定：高速メイン・モード 4.0V~5.5V、 $4.0V < EVDD0 < 5.5V$ 、高速オンチップ・オシレータ 8MHz、PLL 発振回路 96MHz
- LVD0：リセット・モード($V_{LVD0}=2.91V$)
- TAU チャンネル 0：クロック周波数 48MHz、インターバル時間 1ms、割り込み無効
- TAU チャンネル 1：クロック周波数 48MHz、インターバル時間 100us、割り込み有効(レベル 0)
- TAU チャンネル 2：クロック周波数 48MHz、インターバル時間 100us、割り込み有効(レベル 1)
- TAU チャンネル 3：クロック周波数 24MHz、インターバル時間 1458us、割り込み有効(レベル 1)
- TKB0：クロック周波数 96MHz、PWM 周期 2.5us(400KHz)
 - TKBO00：出力許可、デフォルト・ロー、アクティブ・ハイ、ディザリング有効
強制出力停止機能 1 を有効、出力レベルロー、タイプ 4、コンパレータ 0 をトリガ
 - TKBO01：出力許可、デフォルト・ロー、アクティブ・ハイ、ディザリング有効
強制出力停止機能 1 を有効、出力レベルロー、タイプ 4、コンパレータ 1 をトリガ
- TKB1：クロック周波数 96MHz、PWM 周期 2.5us(400KHz)
 - TKBO10：出力許可、デフォルト・ロー、アクティブ・ハイ、ディザリング有効
強制出力停止機能 1 を有効、出力レベルロー、タイプ 4、コンパレータ 2 をトリガ
 - TKBO11：未使用
- WDT：HALT/STOP/SNOOZE モード時に使用する、オーバーフロー時間 $2^{17}/f_{IL}(3.90625ms)$ 、ウィンドウ・オープン期間 100%、割り込み無効
- DAC0：10 ビット、変換値 133(500mA の検出電圧)
- COMP0：DAC0 出力を基準、立ち上がりエッジ、割り込み無効
- COMP1：DAC0 出力を基準、立ち上がりエッジ、割り込み無効
- COMP2：DAC0 出力を基準、立ち上がりエッジ、割り込み無効
- ADC：12 ビット、AVREFP を+基準、AVREFM を-基準、ワンショット変換モード、標準 1、55/fCLK(1.14583us)
 - ADS0：ソフトウェア・トリガ、入力ソース=PGA 出力、変換優先順位=高、割り込み無効
 - ADS1：未使用
 - ADS2：未使用
 - ADS3：未使用
- PGA：PGAGND、入力チャンネル=PGA0、増幅率 8 倍
- DALI：クロック周波数 48MHz、非拡張モード、送信データ長 8 ビット、転送速度 1200bps、グレー領域のエッジ保存モード標準、ビットタイミング違反有効、ハーフビットと 2 ハーフビット間のグレー領域にあるエッジはビットタイミング違反として検出されない、BTV1=0(0us)、BTV2=96(624us)、BTV3=90(585us)、BTV4=92(598us)、BTV5=169(1098.5)、BTV6=292(1898us)、ストップ・ビット検出割り込み有効(レベル 3)、立ち下がりエッジ検出割り込み有効(レベル 3)、バスパワーダウン検出割り込み有効(レベル 3)
- UART：
 - 送信：クロック周波数 1500KHz、連続転送モード、8 ビット、LSB、パリティなし、1 ストップ・ビット
非反転、9600bps、割り込み有効(レベル 3)
 - 受信：クロック周波数 1500KHz、連続転送モード、8 ビット、LSB、パリティなし、1 ストップ・ビット
非反転、9600bps、割り込み無効

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 400 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。

3.3 定電流および調光制御

TAU チャンネル 0、TAU チャンネル 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 DALI 通信によって設定された LED の調光率^{注1}を取得し、各 LED チャンネルの調光目標レベル ('VRn'^{注1}) を計算します。本サンプル・プログラムでは LED1 と LED2 が制御対象であり LED3 は制御対象外です。LED3 の制御を行う場合は、"r_led.h"ファイルより提供される R_LED_SetLevel 関数を用いて調光レベルを設定してください。第一引数に ch = 2 を指定し、第二引数に目標値を設定することで LED3 の調光制御を行うことができます。

注 1 LED の調光率の設定方法については以下を参照してください。

(仮) DALI102,207,209 サンプル APN

フィードバック処理は、100 μ s ごとに発生する TAU0_1 (INTTM01) 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2} の定電流制御を行います。

A/D 変換の結果 'LEDn_FB_AD'^{注2} は、PI 制御によるフィードバック処理で、最後の結果 'LEDn_FB_AD_OLD'^{注2} および目標レベル 'VRn'^{注1} 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 3.4 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。

LEDn^{注2} の目標値計算では、DALI 通信によって設定された LED の調光率を取得し、LED の輝度を表す調光レベル (目標レベル: 'VRn'^{注2}) が計算されます。このサンプル・プログラムでは、"LEDn_FB_AD"^{注2} 範囲 (0~0.45 V) と等価な 'VRn'^{注1} 範囲 (当初は 0~5 V に対応する値の間) を得るように、変換します。

注 2. n=1,2,3 (ボード RTK7RLG24 上で、LED1 が赤、LED2 が緑、LED3 が青)

TM01 割り込みが 100 μ s ごとに発生すると、アナログ入力チャンネルはフィードバック・チャンネルの 1 つに変更されます。さらに、割り込み処理ルーチンの次の繰り返しで入力チャンネルを次のフィードバック・チャンネルにシフトするために、"gs_feedback"変数を更新します。

LED1 の定電流制御に使用する回路を下図に示します。RL78/G24 PWM 出力は Pre-Driver を介しバックコンバータの MOSFET を ON/OFF 制御し、次に ADC 入力は LED フィードバック電流を測定し、RL78/G24 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。目標値 VR_n は DALI 通信によって設定された LED の調光率に従って決定されます。

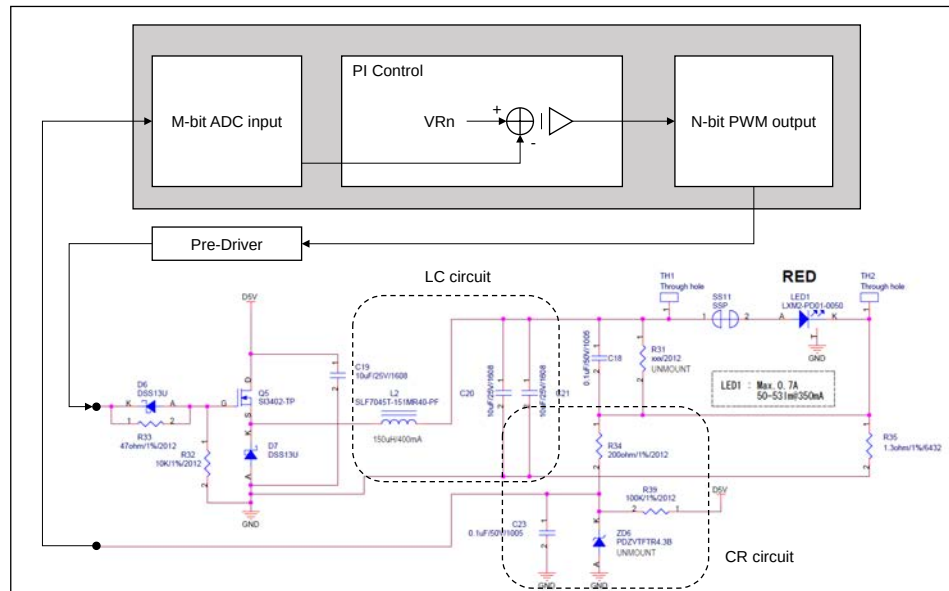


図 3-1 LED1 定電流制御のためのバック・コンバータ回路

アナログ入力チャネルのシフト・フローを下図に示します。

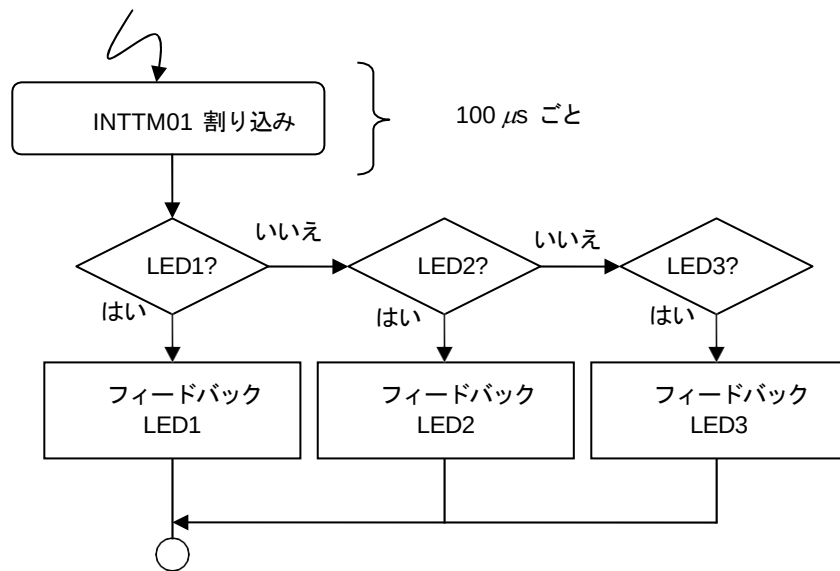


図 3-2 アナログ入力チャネルのシフト・フロー

3.4 PI 制御によるフィードバック方法

LED の定電流制御と調光制御（オン／オフを含む）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、3.5 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} であるときの設定方法を以下に示します。

$$X_{TARGET} = \frac{(I_{LED} \times 8) \times R_S}{V_{REF}} \times (2^M - 1)$$

注 この節のサンプル・プログラムでは、A/D 変換目標値 X_{TARGET} を 'VRn' と定義しています。

たとえば、LED 電流 $I_{LED}=350$ mA で定電流制御を実行するとき、センス抵抗 $R_S=1.3$ Ω、A/D コンバータ基準電圧 $V_{REF}=5$ V および A/D 変換解像度 $M=12$ ビットと仮定して、A/D 変換目標値 $X_{TARGET}=2981$ 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/G24 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 350 mA から 100 mA に変更するには、 X_{TARGET} 値を 2981 から 852 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、 ± 5 mV～ ± 10 mV 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 'LEDn_OFFSET' 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/G24 DC/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

(R39、R48、R57) 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「 $A2 \times E(n-1)$ 」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「 $A1 \times E(n) + A2 \times E(n-1)$ 」を計算します。
- <7> 最後の PWM 出力デューティ $D(n-1)$ を<6>の「 $A1 \times E(n) + A2 \times E(n-1)$ 」の結果と比較します。
「 $D(n-1) + A1 \times E(n) + A2 \times E(n-1)$ 」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。
- <8> PWM 出力のデューティ設定値を更新します。
- <9> 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。

上記の動作のフロー（<1>から<9>まで）に対応するサンプル・プログラムの LED1 チャンネルに関する一部分を次ページに記します。

```

PGAINS = FEEDBACK_LED1;
pga_start();                                     /* <1> */
gs_get_value = adc_ads0_execute();               /* <2> */

LED1_FB_AD = gs_get_value;                       /* <3> */
temp11 = LED1_VR - (LED1_FB_AD_OLD - LED1_OFFSET); /* <4> */
LED1_ERR = MACSW(0,A2,temp11);

If (LED1_FB_AD >= (CURRENT_MAX + LED1_OFFSET))    /* <5> */
{
    force_stop();
}
else
{
    temp12 = LED1_VR - (LED1_FB_AD - LED1_OFFSET); /* <6> */
    LED1_ERR = MACSW(LED1_ERR,A1,temp12);          /* <7> */
    if (LED1_ERR > 0)
    {
        if (DUTY_MAX - LED1_DUTY >= LED1_ERR)
        {
            LED1_DUTY += LED1_ERR;
        }
        else
        {
            LED1_DUTY = DUTY_MAX;
        }
    }
    else if (LED1_ERR < 0)
    {
        if (LED1_DUTY > - LED1_ERR)
        {
            LED1_DUTY += LED1_ERR;
        }
        else
        {
            LED1_DUTY = 0;
        }
    }
    else if ((LED1_ERR == 0) && (LED1_VR == 0))
    {
        LED1_DUTY = 0;
    }
    else
    {
    }
    LED1_TKBCRLD =
    (uint16_t)(LED1_DUTY >> 8) << TKBCRLD_BIT_SHIFT_SIZE; /* <8> */
}
LED1_FB_AD_OLD = LED1_FB_AD;                    /* <9> */
tkb0_set_batch_overwrite_request_on();
NOP();

```


3.5 PI 制御式の係数の計算

ここでは、3.4 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。

$$\begin{aligned} A1 &= (\pi \times f_Z \times T + 1) \times K_P \\ A2 &= (\pi \times f_Z \times T - 1) \times K_P \end{aligned}$$

π : パイ (円周率)
 f_Z : ゼロ・ポイント周波数
 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 制御回路の利得から得られます。

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 3-1 に示すように、この制御回路には 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、

$L1=150 \mu\text{H}$ 、 $C1=20 \mu\text{F}$ 、 $C2=0.1 \mu\text{F}$ 、 $R2=200 \Omega$ の場合は次の値が得られます。

$$f_{C1} = \frac{1}{2\pi\sqrt{L1 \cdot C1}} = 2.9\text{KHz}$$

$$f_{C2} = \frac{1}{2\pi \cdot C2 \cdot R2} = 7.9\text{KHz}$$

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。

$$f_Z = 1.5\text{KHz}$$

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。

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

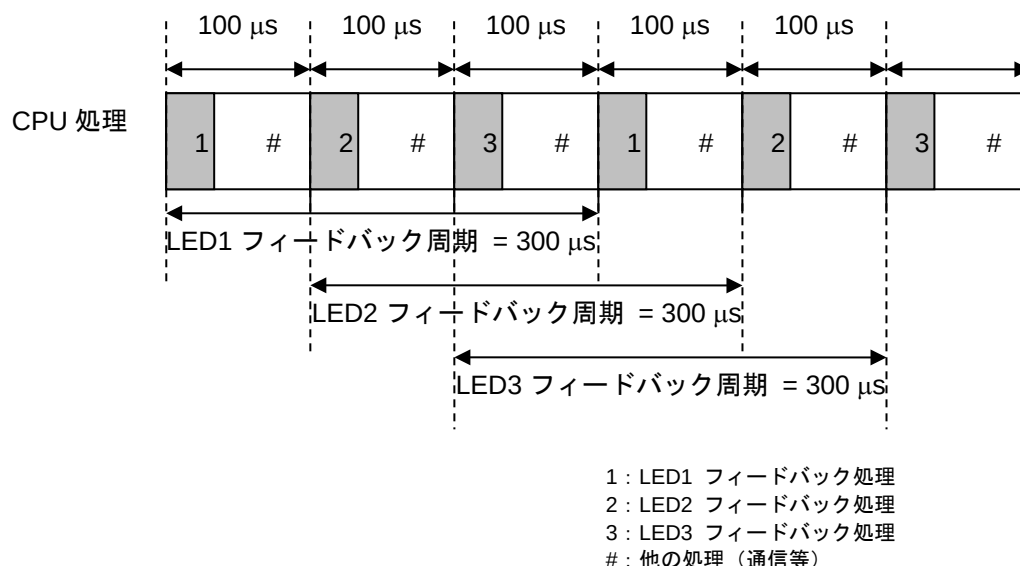
したがって、 $f_Z=1.5 \text{ kHz}$ とすると、フィードバック周期 T は $333 \mu\text{s}$ 未満でなければなりません。

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 3-3 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 $100 \mu\text{s}$ の周期で分配されません。

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。

$$T = 300\mu\text{s}$$

図 3-3 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 15 ビット (ADC の 12 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 5 V、A/D コンバータ基準電圧 V_{REF} が 5 V 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。

$$\frac{i_{PWM}}{i_{AD}} = 2^3$$

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。

$$K_P = 0.1$$

上記の結果から、PI 制御係数 A_1 および A_2 を計算することができます。

$$\begin{aligned} A_1 &= 0.241 \\ A_2 &= 0.041 \end{aligned}$$

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、 $2^8 (=256)$ を乗算します。

$$\begin{aligned} A_1 &= 61 \\ A_2 &= 11 \end{aligned}$$

3.6 ソフトウェア・フロー・チャート

図 3-4 メイン・プログラム・ループのフロー・チャート

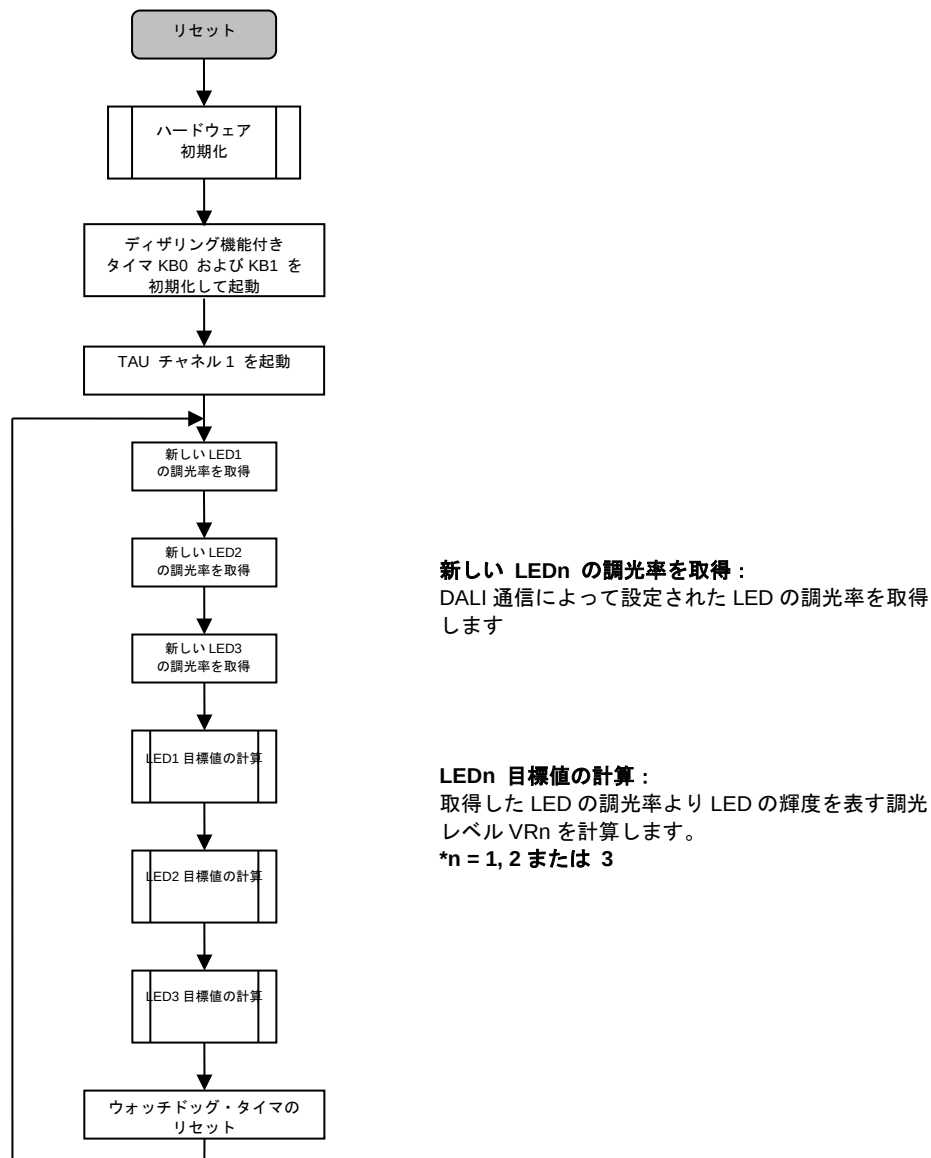


図 3-5 LEDn 目標値 (VRn) 計算処理のフロー・チャート (n=1、2、3)

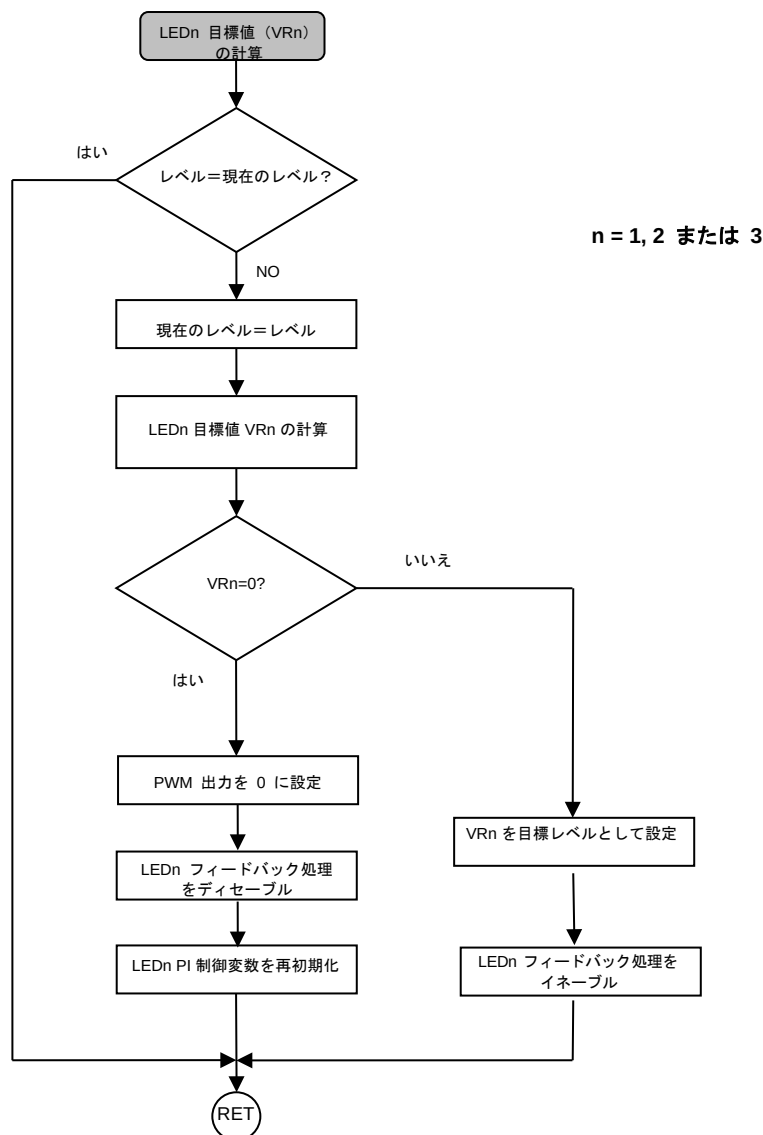
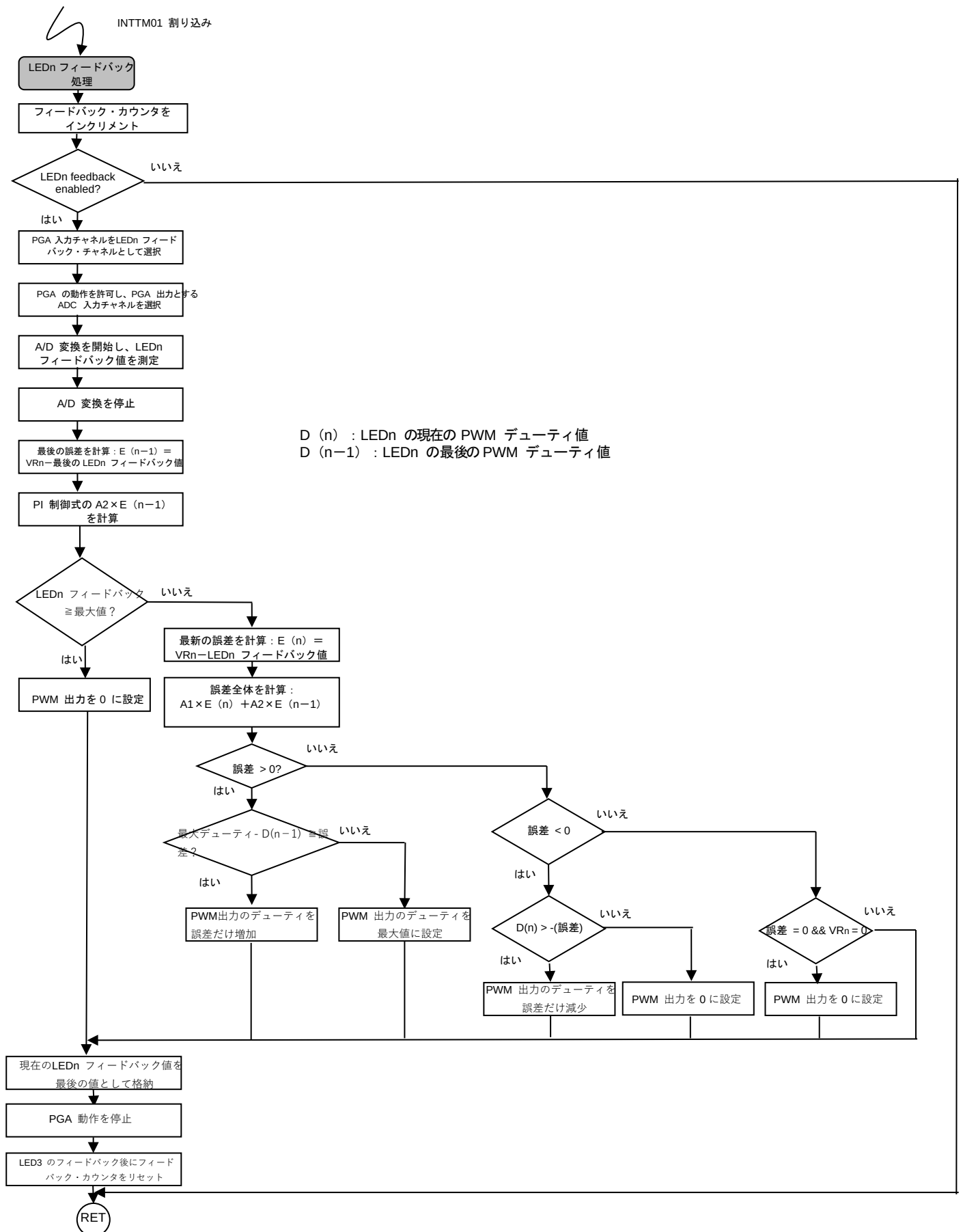


図 3-6 LEDn フィードバック処理のフロー・チャート (n=1、2、3)



4. RL78/G24 DC/DC LED 制御評価ボードの起動方法

与えられたサンプル・プログラムを使用して RL78/G24 DC/DC LED 制御評価ボードの評価を開始する前に、以下の手順を実行してください。

以下の手順は、RFP（Renesas Flash Programmer）ソフトウェア・ツールを使用してサンプル・プログラムを RL78/G24 フラッシュ・メモリにダウンロードするプロセスを示しています。

1. COM ポートによるフラッシュ・プログラミングを可能にするために、SW1 ビット 1-3-5 を ON 位置に設定（他のビットはすべて OFF 位置に設定）することによってフラッシュ・プログラミング・モードを選択します。
2. 5 V 電源を CN1 DC ジャック・コネクタに接続します。
3. タイプ A Mini-B USB ケーブルを使用して、PC USB ポートに RL78/G24 DC/DC LED 制御評価ボードを接続します。

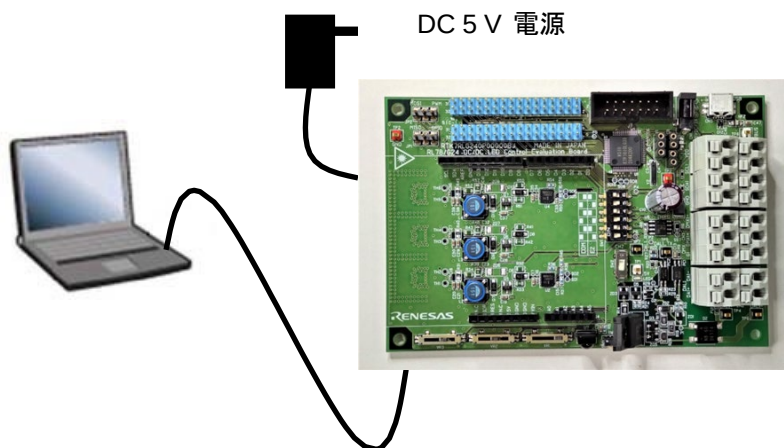


図 4-1 システム環境

4. 「RFP」フラッシュ・プログラマ・ツールを開きます。

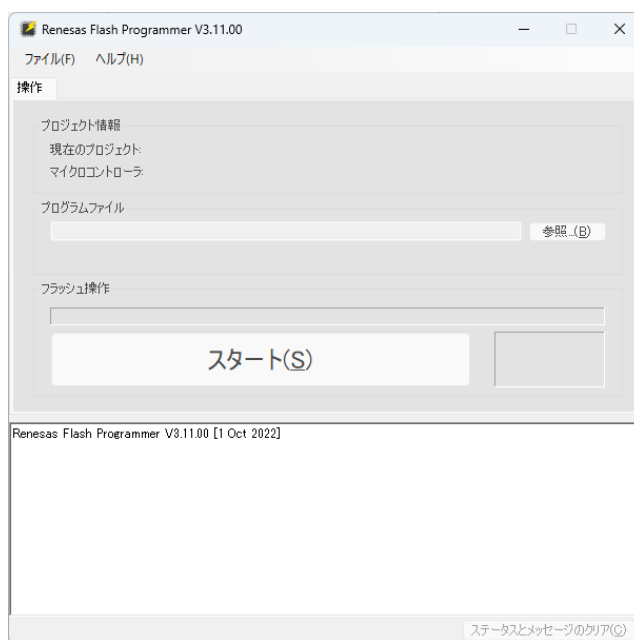


図 4-2 RFP 起動画面

5. 「ファイル」⇒「新しいプロジェクトを作成」をクリックします。「新しいプロジェクトの作成」ウィンドウが開きます。
マイクロコントローラに「RL78/G2x」を選択します。

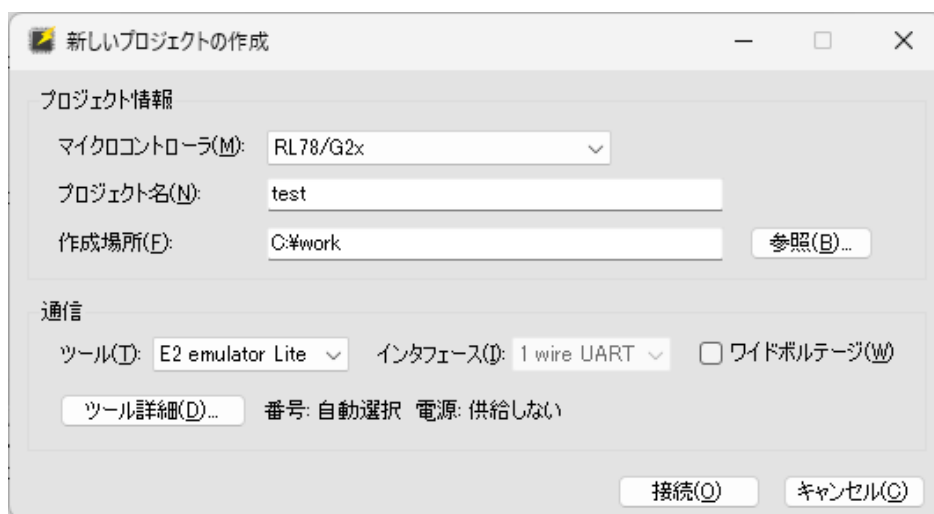


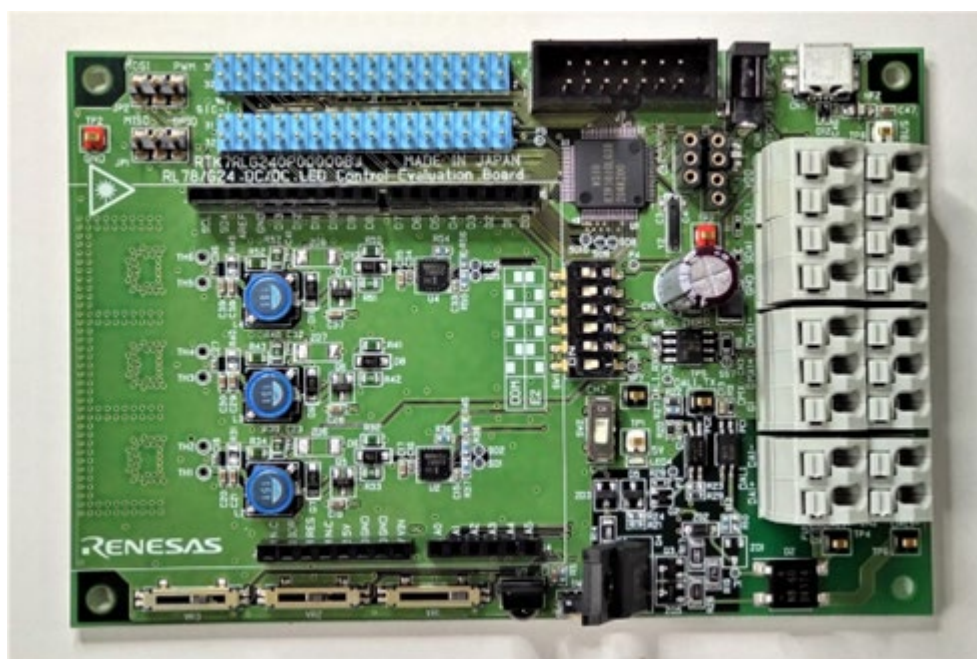
図 4-3 RFP フラッシュ・プログラミング・ツール - 「新しいプロジェクトの作成」ウィンドウ

6. プログラミングするサンプル・プログラムの「hex」ファイルを選択します。
7. 「スタート」ボタンをクリックして、サンプル・プログラムを RL78/G24 フラッシュ・メモリにフラッシュ・プログラミングします。
8. フラッシュ・プログラミングが完了したら、「RFP」を閉じます。



図 4-4 RFP フラッシュ・プログラミング・ツール - 「終了画面」ウィンドウ

9. 5 V 電源と USB ケーブルを取り外します。
10. 再び DC 5 V 電源を CN1 コネクタに接続します。
11. DALI 通信による LED の調光を確認します。
12. 評価が完了したら、CN1 コネクタから DC 5 V 電源を外します。



付録 B 回路図

回路図については以下を参照してください。

RTK7RLG240P00000BJ Schematics (R12TU0260)

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/>

お問い合わせ先

<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	Sep. 1 st , 23	—	初版

本製品は外国為替及び外国貿易法の規定により規制貨物等に該当しますので、日本国外に輸出する場合には、同法に基づき日本国政府の輸出許可が必要です。
すべての商標および登録商標は、それぞれの所有者に帰属します。

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. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレイやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

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. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れなくてください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「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。

5. クロックについて

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。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 V_{IL} (Max.) から V_{IH} (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 V_{IL} (Max.) から V_{IH} (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

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、将来の拡張機能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。

8. 製品間の相違について

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えば、フラッシュメモリ、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。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、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
 2. 当社製品または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またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
 3. 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. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
 5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング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。
 6. 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）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等
当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、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（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
 7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因したまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
 8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合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。
 9. 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、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他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。
 10. 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。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。
 11. 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。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
 12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものとなります。
 13. 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。
 14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 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直接的、間接的に支配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。

(Rev.5.0-1 2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。

すべての商標および登録商標は、それぞれの所有者に帰属します。