

Renesas RA0 シリーズ

低速オンチップオシレータ (LOCO)、中速オンチップオシレータ (MOCO) クロック周波数補正

要旨

本アプリケーションノートでは、RA0 シリーズ (RA0) に搭載されている低速オンチップオシレータ (LOCO) および中速オンチップオシレータ (MOCO) の発振クロック周波数を補正する方法について説明します。補正レジスタとして、低速オンチップオシレータトリミングレジスタ (LIOTRM) を LOCO 用、中速オンチップトリミングレジスタ (MIOTRM) を MOCO 用として使用します。

LOCO または MOCO の発振周波数の誤差は、高速オンチップ発振器 (HOCO) を使用して検出します。その後、LOCO トリミングレジスタ (LIOTRM) および MOCO トリミングレジスタ (MIOTRM) を調整し、LOCO の発振周波数を約 32.768 kHz、MOCO の発振周波数を約 4 MHz にそれぞれ近づけます。

動作確認デバイスとして RA0E1 グループ (RA0E1) を使用していますが、他の RA0 製品にも流用できません。

必要なリソース

このアプリケーションノートでは、以下の資料を参照しています。

開発ツールとソフトウェア

- e²studio version: 2025-04 (25.4.0)
- LLVM Embedded Toolchain for Arm v18.1.3
- Renesas Flexible Software Package (FSP) v5.9.0 or later.

ターゲットデバイス

RA0E1 MCU

本アプリケーションノートのサンプルプログラムを別のマイクロコントローラ (MCU) に流用する際は、対象となるマイクロコントローラの仕様に合わせて修正してください。変更を加えた後は、更新されたプログラムの評価を行ってください。

ハードウェア要件

- このアプリケーションノートでは、Windows® 10 オペレーティングシステムを搭載した PC を例として使用しています。対応オペレーティングシステムの一覧については、開発ツールおよびソフトウェアの各ユーザーマニュアルをご参照ください。
- FPB-RA0E1.

参考マニュアル

- RA Flexible Software Package Documentation Release v5.9.0
- Renesas RA0E1 Group User's Manual Rev.1.0
- FPB-RA0E1-v1.0 Schematics

Contents

1. 仕様	3
1.1 仕様概要	3
1.2 動作概要	5
1.3 キャリブレーション方法の説明	8
2. ハードウェアの説明	11
2.1 ハードウェア構成の例	11
3. ソフトウェア説明	12
3.1 定数一覧	14
3.2 関数一覧	14
3.3 関数のフローチャート	14
3.3.1 Calibration Process	14
4. サンプルプロジェクト	19
4.1 プロジェクトのインポート	19
4.2 プロジェクトのビルド	19
4.3 プロジェクトのダウンロードと実行	20
4.4 周波数補正結果の確認	20
Revision History	23

1. 仕様

1.1 仕様概要

本アプリケーションノートでは、HOCO を使用して LOCO および MOCO のクロック発振周波数のずれを検出します。LIOTRM レジスタと MIOTRIM レジスタを調整し、LOCO の発振周波数を 32.768 kHz に、MOCO の発振周波数を 4 MHz にそれぞれ近づけます。

スタートスイッチが押されると、32 ビットインターバルタイマ (TML32) が 16 ビットキャプチャモードで動作し、LOCO または MOCO の周期を測定します。TML32 のカウントクロックとして HOCO を使用します。TML32 からのカウント値が目標範囲外の場合、LIOTRM レジスタまたは MIOTRIM レジスタを調整して、LOCO の周波数を 32.768 kHz に、MOCO の周波数を 4 MHz にそれぞれ近づけるように調整します。LOCO の発振周波数の目標範囲は $32.768 \text{ kHz} \pm 1.6\%$ (32.571 kHz~32.965 kHz) であり、MOCO の発振周波数の目標範囲は $4 \text{ MHz} \pm 1.3\%$ (3.948 MHz~4.052 MHz) です。ただし、HOCO の周波数が $\pm 1.0\%$ の範囲内にあることが条件となります。

TML32 は、LOCO と MOCO の両方の周期を測定します。精度を高めるため、パルス間隔を 4 回測定し、LOCO と MOCO の発振周波数の誤差を検出します。

注意：本サンプルコードでの設定時間、キャリブレーションの方法は一例です。本サンプルコードでは、フローの簡略化、理解の容易さを考慮し、スタートスイッチ入力でキャリブレーションを開始しています。システムに応じて、キャリブレーション処理の開始タイミングや開始間隔の調整を行ってください。

表 1-1 に使用する周辺機能と用途を示します。

表 1-1 使用する周辺機能と用途

周辺機能	用途
P200 (S1), 端子入力エッジ検出割り込み (R_ICU)	補正開始スイッチに使用します。
TML32 を 16 ビットキャプチャモードで使用：チャンネル 0 と 1、チャンネル 2 と 3 を接続し、それぞれ 16 ビットタイマとして使用します。	LOCO/MOCO のキャリブレーション処理に使用します。
タイマアレイユニット チャンネル 0 (R_TAU)	補正開始スイッチのチャタリング除去に使用します。

図 1-1 は、クロック設定の選択画面を示しています。FXSP ソースとして LOCO を選択し、TML32 FITL1 ソースとして選択してください。

図 1-1 クロック設定例

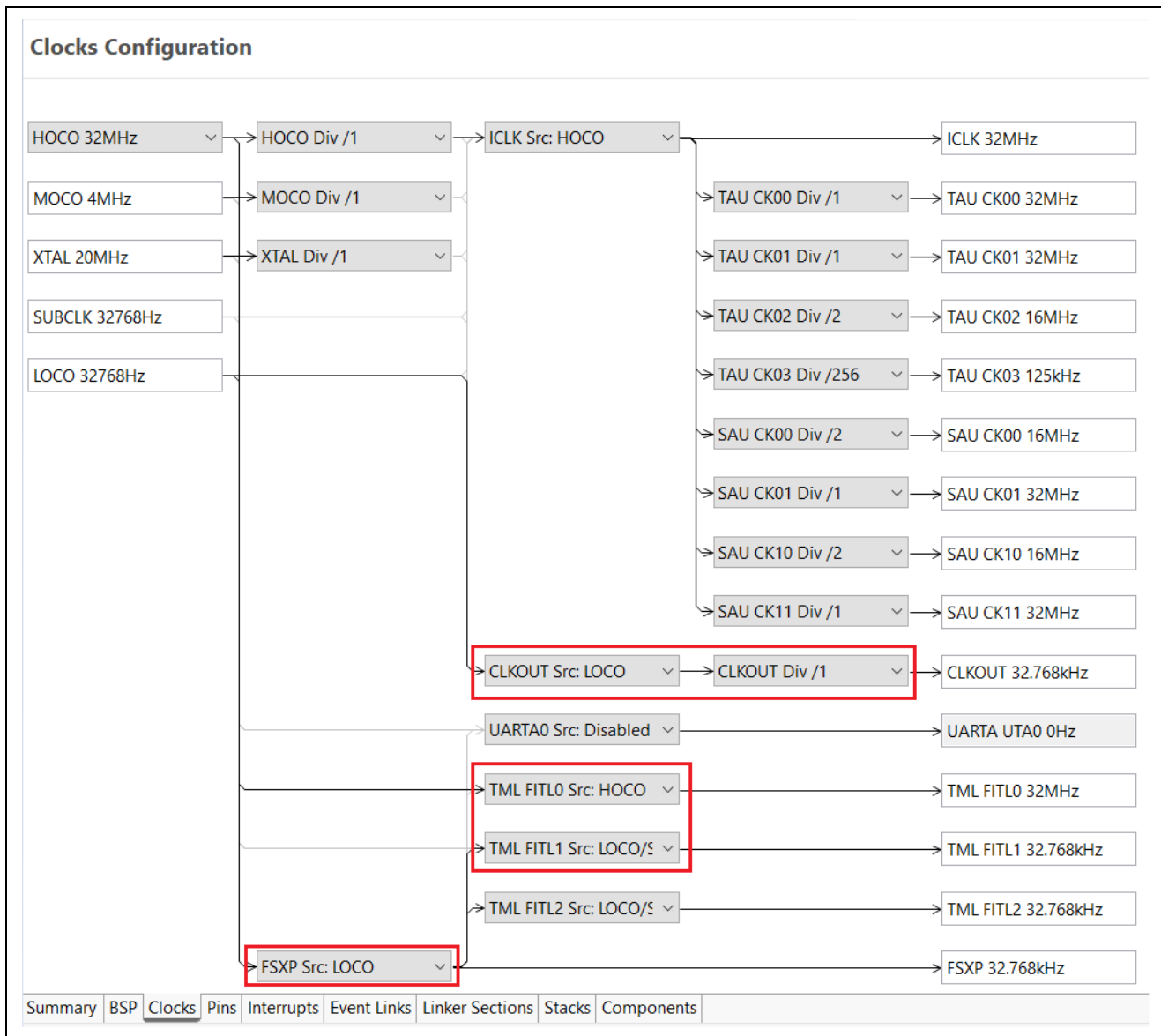
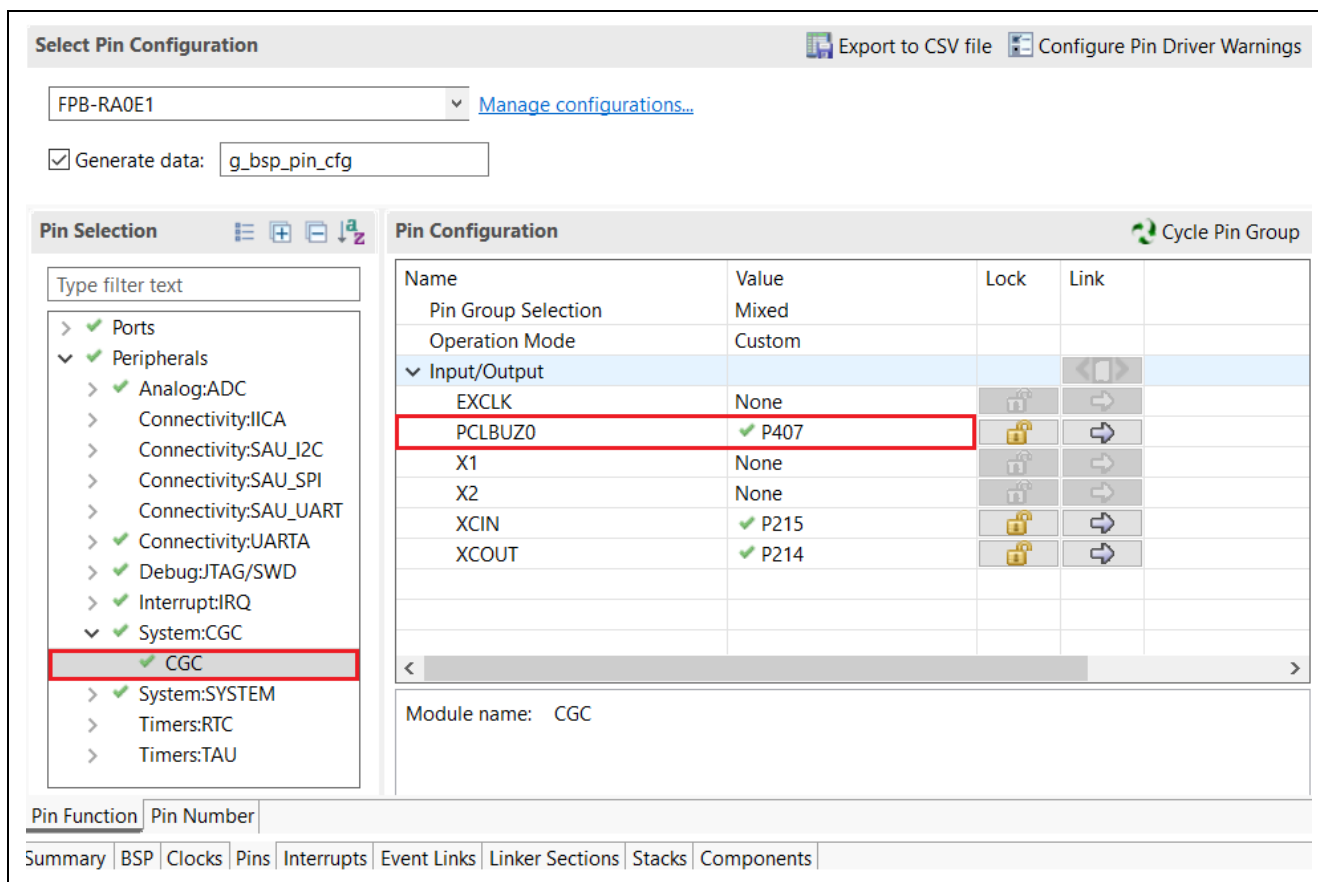


図 1-1 に示すように、CLKOUT のソースとして LOCO を設定し、図 1-2 に示すように P407 を PCLBUZ0 端子として設定します。この設定により、P407 からの LOCO 出力が有効になります。オシロスコープを使用して、LOCO クロックの補正を確認することができます。

図 1-2 PCLBUZ0 端子セットアップ例



1.2 動作概要

以下に、周辺機器機能の設定概要を示します。

(1) 外部割り込みの初期設定

表 1-2 に外部割り込みの設定条件を示します。

表 1-2 外部割り込み (IRQ0) の初期設定条件

設定項目	設定条件
有効エッジ	立ち下がりエッジ
端子	P200

(2) TML32 を 16 ビットキャプチャモードで使用するときの初期設定

チャンネル 0 および 1 で 16 ビットキャプチャモードを使用する場合、指定されたキャプチャトリガを検出すると、カウンタ値がインターバルタイムキャプチャレジスタ 00 (ITLCAP00) に格納されます。

16 ビットキャプチャモードのレジスタ設定については、図 1-3 を参照してください。

図 1-3 16 ビットキャプチャモードのレジスタ設定

レジスタ名 (シンボル)	ビット	設定
インターバルタイムコントロールレジスタ 0 (ITLCTL0)	EN0	チャンネル 0 とチャンネル 1 のカウンタの開始/停止を指定する。
	EN1	0 に設定する。
	EN2	チャンネル 2 とチャンネル 3 のカウンタの開始/停止を指定する。
	EN3	0 に設定する。
	MD[1:0]	01b に設定する。
インターバルタイム分周レジスタ 0 (ITLFDIV00)	FDIV0[2:0]	チャンネル 0 のカウンタ用クロックを選択する。
	FDIV1[2:0]	000b に設定する。
インターバルタイム分周レジスタ 1 (ITLFDIV01)	FDIV2[2:0]	000b に設定する。
	FDIV3[2:0]	000b に設定する。
インターバルタイムクロック選択レジスタ 0 (ITLCSEL0)	ISEL[2:0]	チャンネル 0 とチャンネル 1 で使用するインターバルタイムのカウンタ用クロックを選択する。
	CSEL[2:0]	チャンネル 2 とチャンネル 3 で使用するキャプチャ用インターバルタイムのカウンタ用クロックを選択する。
インターバルタイムキャプチャコントロールレジスタ 0 (ITLCC0)	CAPEN	1 に設定する。
	CAPCCR	キャプチャ完了後にチャンネル 0 とチャンネル 1 のカウンタ値をクリアするか保持するかを指定する。
	CTRS[1:0]	キャプチャトリガを選択する。

注 1. 「ITLCMP01 とのコンペアマッチ検出時の割り込み」をキャプチャトリガに使用しない場合に 16 ビットカウンタモードで使用できるのはチャンネル 2 とチャンネル 3 だけです。

サンプルコードの設定を表 1-3 に示します。

表 1-3 TML32 の初期設定 (チャンネル 0 および 1 で 16 ビットキャプチャモードに設定されている場合)

設定項目	設定条件
動作モード	16 ビットキャプチャモード
動作クロック	HOCO (32 MHz)
クロックソース	HOCO (分周なし)
キャプチャトリガ	ITLCMP01のコンペア一致割り込み
キャプチャクロック	FSXP/MOCO
キャプチャトリガのインターバル時間	FSXP(LOCO): 2カウント/MOCO: 128 カウント

図 1-4 は、FSP を使用した TML32 の設定例を示しています。

図 1-4 LOCO 補正のための TML32 設定例

g_tml_timer0 32-bit Interval Timer (r_tml)			
Settings	Property	Value	
API Info	16-bit Capture Mode Support	Enabled	
	Enable Single Channel	Disabled	
	Interrupt Support	Enabled	
	▼ Module g_tml_timer0 32-bit Interval Timer (r_tml)		
	▼ General		
	Name	g_tml_timer0	
	Mode	16-bit Capture Mode	
	Channel Selection	0	
	ELC event	Disabled	
	▼ Counter Mode Settings		
	Period	0x10000	
	Period Unit	Raw Counts	
	▼ Capture Mode Settings		
	▼ 16-Bit Counter Input Settings (when Capture source = Interr		
	Period	0x2	
	Period Unit	Raw Counts	
	Capture Trigger	Interrupt on compare match with ITLCMP01	
	Capture Clock Divider	fitL0	
	Counter Status	Cleared after the completion of capturing	
	> Interrupt		

(3) タイマアレイユニットの初期化

表 1-4 にタイマアレイユニットの初期設定を示します。

表 1-4 タイマアレイユニット チャンネル 0 の初期設定条件

設定項目	設定条件
動作モード	インターバルタイマ
動作クロック	CK00
クロックソースの分周比	分周なし
インターバル時間	10ms

図 1-5 は、FSP を使用したタイマレイユニットの設定例を示しています。

図 1-5 タイマレイユニットの設定例

g_tau_debounce_timer0 Timer, Independent Channel, 16-bit and 8-bit Timer Operation (r_tau)		
Settings	Property	Value
API Info	▼ Common	
	Parameter Checking	Default (BSP)
	Interrupt Support	Enabled
	Pin Output Support	Disabled
	Pin Input Support	Disabled
	Extra Input Mode Support	Disabled
	8-Bit Mode Support	Disabled
	▼ Module g_tau_debounce_timer0 Timer, Independent Channel, 16-	
	▼ General	
	Name	g_tau_debounce_timer0
	Channel	0
	Function	Interval Timer
	Bit Timer Mode	16-bit timer
	Operation Clock	CK00
Period	10	
Period Unit	Milliseconds	

1.3 キャリブレーション方法の説明

このセクションでは、本アプリケーションノートで使用される 2 つのキャリブレーション方法について説明します。

まず、LOCO および MOCO の周期を測定するために、TML32 を 16 ビットキャプチャモードで動作するように設定します。キャプチャトリガに ITLCMP01 のコンペア一致割り込みを選択します。このトリガは、LOCO (32.768 kHz) のクロックを使用する場合、2 カウントごとに、MOCO (4 MHz) のクロックを使用する場合、128 カウントごとにイベントを生成します。

次に、カウントクロックとして HOCO (32 MHz) を選択します。FXSP ソースとして LOCO を設定します。その後、TML32 FITL1 ソースとして MOCO または LOCO のいずれかを選択できます。精度を高めるために、LOCO/MOCO の周期を連続して 4 回測定します。キャプチャされた 4 つの値を合計し、LOCO または MOCO の発振クロックの周波数に誤差がないかを確認します。

目標周波数範囲は ± 2 LSB 以内とします。これは、LIOTRM レジスタでは $\pm 0.6\%$ 、MIOTRM レジスタでは $\pm 0.3\%$ に相当します。

表 1-5 は、LOCO の 4 回分のカウント値の目標範囲を示しており、その範囲は 7766 から 7859 です。

表 1-5 LOCO の 4 回分のカウント値の目標範囲

LOCO クロック周波数		キャプチャ 4 回分のカウント値 (計算値)
32.768 kHz		7812.5
32.768 kHz - 0.6%	32.571 kHz	7859.658
32.768 kHz + 0.6%	32.965 kHz	7765.905

HOCO の周波数精度が±1.0%以内である場合、LOCO の周波数精度は±1.6%以内となります。

カウント値が 7765 以下の場合、LOCO クロックが目標周波数よりも遅いことを示します。一方、カウント値が 7860 以上の場合、LOCO クロックが目標周波数よりも早いことを示します。このカウント値に基づいて、クロックの速度を上げるか下げるかを判断し、LIOTRM レジスタの値を調整します。キャリブレーションは、LIOTRM レジスタの値を±1 ずつ増減させることで行います。カウント値が目標範囲内に入れば、キャリブレーションは完了です。

は、LOCO の補正プロセスを示します。

図 1-6 は、LOCO の補正プロセスを示します。

図 1-6 LOCO の補正プロセス

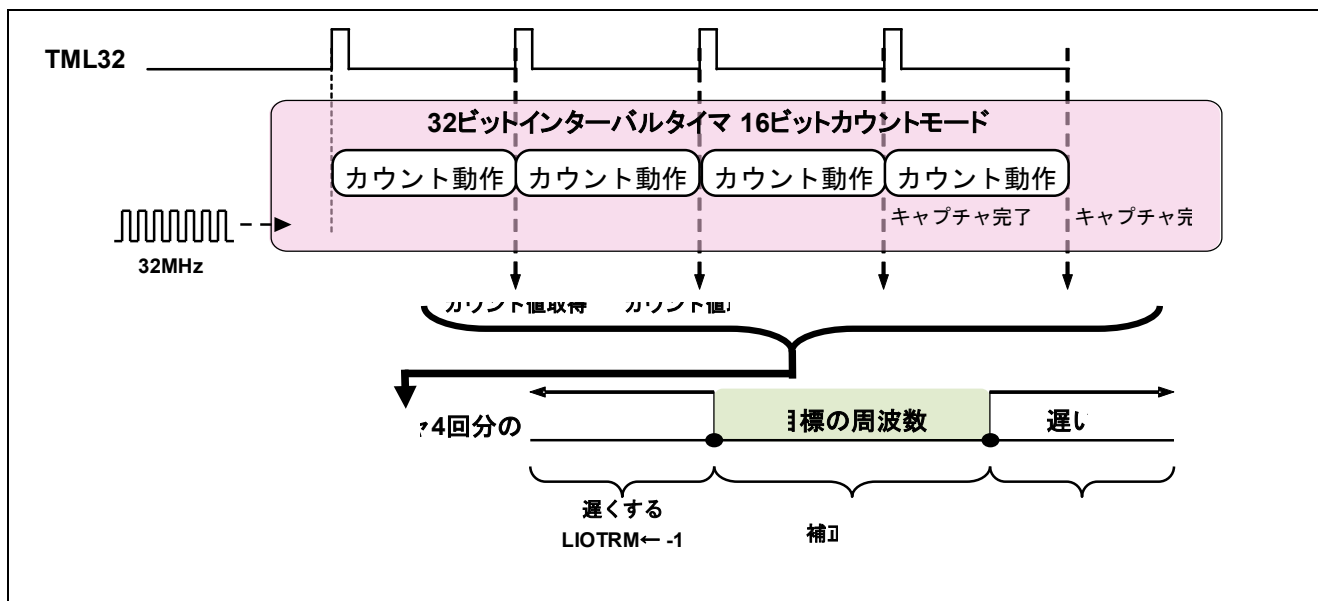


表 1-6 は、MOCO の 4 回分のカウント値の目標範囲を示しており、その範囲は 4084 から 4108 です。

表 1-6 MOCO の 4 回分のカウント値の目標範囲

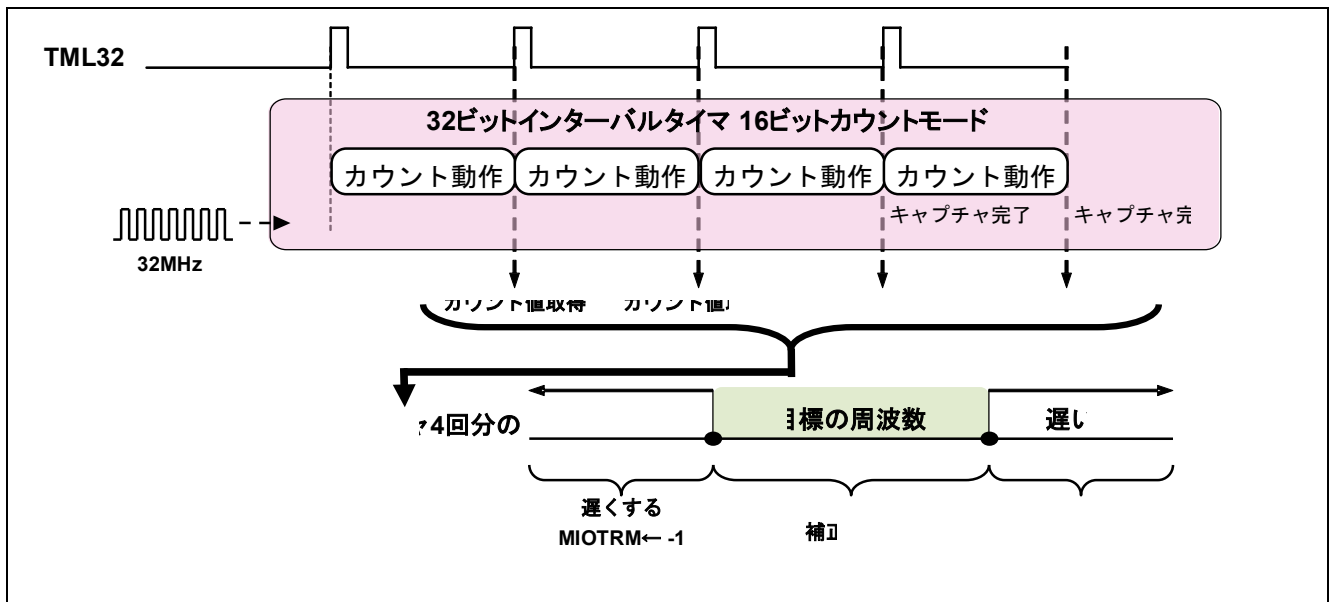
MOCO クロック周波数		キャプチャ 4 回分のカウント値 (計算値)
4MHz		4096
4MHz - 0.3%	3.988MHz	4108.325
4MHz + 0.3%	4.012MHz	4083.749

HOCO の周波数精度が±1.0%以内である場合、LOCO の周波数精度は±1.3%以内となります。

カウント値が 4084 以下の場合、MOCO クロックが目標周波数よりも遅いことを示します。一方、カウント値が 4108 以上の場合、MOCO クロックが目標周波数よりも早いことを示します。このカウント値に基づいて、クロックの速度を上げるか下げるかを判断し、MIOTRM レジスタの値を調整します。キャリブレーションは、MIOTRM レジスタの値を±1 ずつ増減させることで行います。カウント値が目標範囲内に入れば、キャリブレーションは完了です。

図 1-7 は、MOCO の補正プロセスを示します。

図 1-7 MOCO の補正プロセス

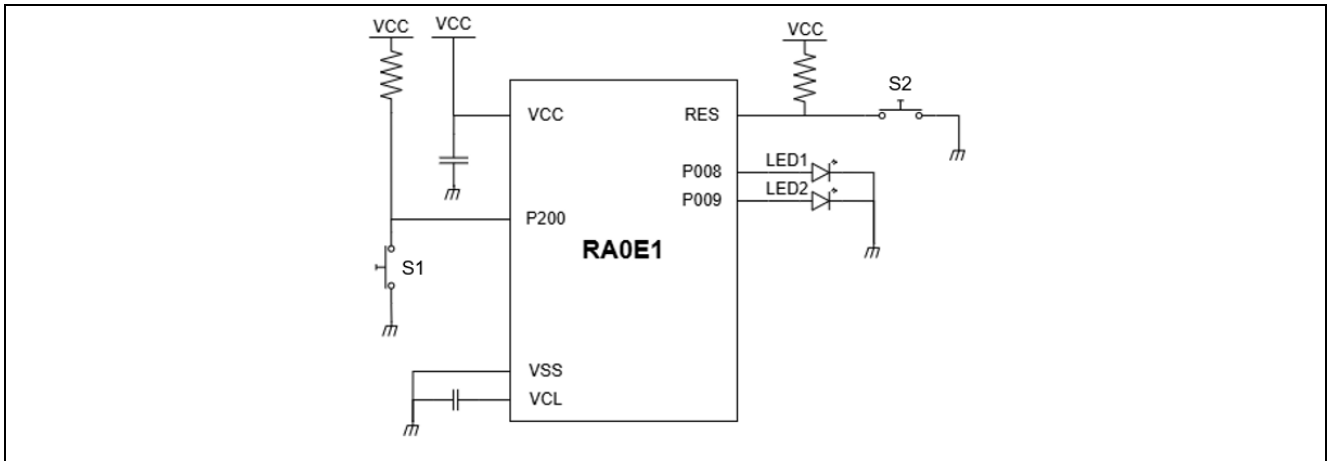


2. ハードウェアの説明

2.1 ハードウェア構成の例

図 2-1 は、本アプリケーションノートで使用されているハードウェア構成の例を示しています。

図 2-1. ハードウェア構成



注意 1. この回路イメージは接続の概要を示す為に簡略化しています。実際に回路を作成される場合は、端子処理などを適切に行い、電気的特性を満たすように設計してください(入力専用ポートは個別に抵抗を介して VCC 又は VSS に接続して下さい)。

注意 2. VCC は LVD にて設定したリセット解除電圧 (V_{det0}) 以上にしてください。

表 2-1 に、使用端子とその機能を示します。

表 2-1 使用端子と機能

端子名	入出力	内容
P200/S1	入力	ユーザースイッチ：2 秒以上押すと、補正モード（LOCO 補正⇔MOCO 補正）を変更し、短押しでキャリブレーションを開始します。
P008/LED1	出力	LOCO補正モードLED： LOCO 補正モード時の状態を表示します：補正開始待ち（点滅）、補正中（消灯）、補正完了（緑色の点灯）。
P009/LED2	出力	MOCO補正モードLED： MOCO 補正モード時の状態を表示します：補正開始待ち（点滅）、補正中（消灯）、補正完了（緑色の点灯）。

注意 本アプリケーションノートは、使用端子のみを端子処理しています。実際に回路を作成する場合は、端子処理などを適切に行い、電気的特性を満たすように設計してください。

3. ソフトウェア説明

図 3-1 に、デモシステムとサンプルコードの概要を示します。

1. 電源を入れると、LED1 が点滅し、LOCO 補正モードであることを示します。FPB スイッチ (S1) を 2 秒以上押し続けると、LED2 が点滅し、MOCO 補正モードであることを示します。S1 をもう一度 2 秒以上押し続けると、システムは LOCO 補正モードに戻ります。
2. スイッチを短押しすると、選択された OCO (LOCO/MOCO) の補正が開始されます。補正中は LED が消灯します。
3. 補正が完了すると、LED が点灯します。

図 3-1. デモシステムとサンプルコードの概要

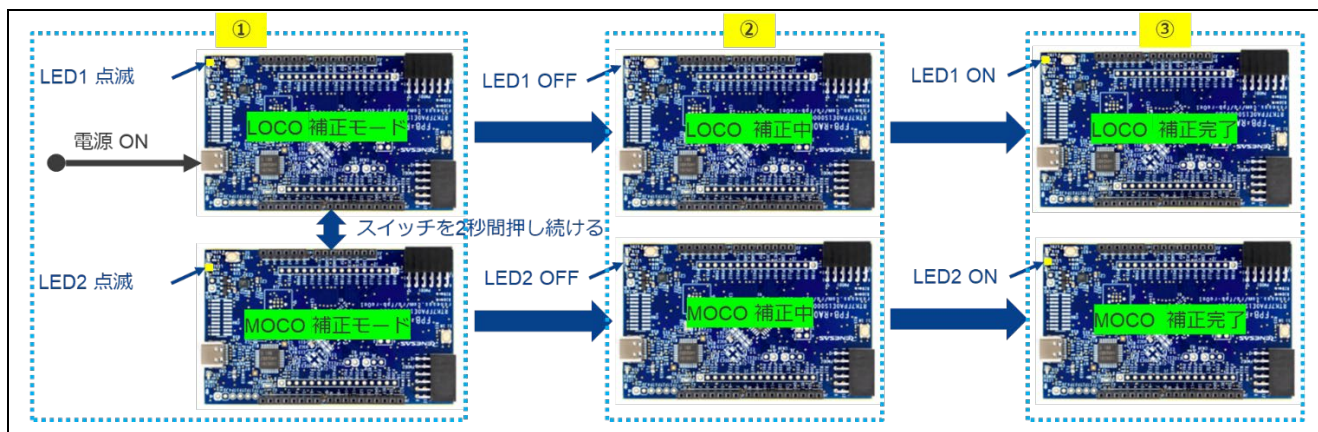


図 3-2 にサンプルプログラムの main ループを示します。

図 3-2. サンプルプログラムのメインループの概要

```

/* Main loop */
while (1)
{
    /* Toggle user LED1: LOCO correction mode */
    led_blinky (leds, oco_type, oco_sts);
    /* User select OCO trimming*/
    switch (g_sw_press)
    {
        case SHORT_PRESS: /* Start calibration process */
        {
            /* Set OCO process status */
            oco_sts = START;
            /* Start the trimming process*/
            R_Main_Calibrate_OCO(oco_type);
            /* Set OCO process status */
            oco_sts = COMPLETE;

            /* Clear user sw status */
            g_sw_press = NO_PRESS;
        }
        break;
        case LONG_PRESS: /* Change OCO correction mode*/
        {
            /* Turn off current user LED */
            err = R_IOPORT_PinWrite(&g_ioport_ctrl, (bsp_io_port_pin_t)leds.p_leds[oco_type], BSP_IO_LEVEL_LOW);
            /* Handle error */
            APP_ERR_TRAP(err);
            /* Change OCO trimming type */
            oco_type ^= MOCO;
            /* Set OCO process status */
            oco_sts = WAIT;

            /* Clear user sw status */
            g_sw_press = NO_PRESS;
        }
        break;
        case NO_PRESS:
        default:
            break;
    }
}

```

補正を実施する前に、MOCO を起動してください、デフォルトでは停止しています。図 3-3 に MOCO を起動するために必要なプログラムを示します。

図 3-3. MOCO 起動のためのサンプルプログラム

```

fsp_err_t MOCO_Start(void)
{
    fsp_err_t err = FSP_SUCCESS;

    /* Unlock CGC and LPM protection registers. */
    R_SYSTEM->PRCR = (uint16_t) BSP_PRV_PRCR_UNLOCK;

    /* Start MOCO */
    R_SYSTEM->MOCO_CR = (uint8_t) (0x00);

    /* Lock CGC and LPM protection registers. */
    R_SYSTEM->PRCR = (uint16_t) BSP_PRV_PRCR_LOCK;

    /* Give some delays fpr oscillation stabilization */
    R_BSP_SoftwareDelay(1U, BSP_DELAY_UNITS_MICROSECONDS);

    return err;
}

```

3.1 定数一覧

表 3-1 にサンプルプログラムで使用されている、主要な定数一覧を示します。

表 3-1 サンプルプログラムの主要な定数

定数名	設定値	内容
LMIOTRM_MAX	11111111B	LIOTRM/ MIOOTRM レジスタの最大値
LMIOTRM_MIN	00000000B	MIOOTRM/ LIOTRM レジスタの最小値
CCNT_HOCO_LOCO_MAX	7860	LOCO 補正のためのクロックカウントの上限値
CCNT_HOCO_LOCO_MIN	7765	LOCO 補正のためのクロックカウントの下限值
CCNT_HOCO_MOCO_MAX	4108	MOCO 補正のためのクロックカウントの上限値
CCNT_HOCO_MOCO_MIN	4084	MOCO 補正のためのクロックカウントの下限值

3.2 関数一覧

表 3-2 にサンプルプログラムで使用されている、主要な関数一覧を示します。

表 3-2 サンプルプログラムの主要な関数

関数名	概要
R_Main_Calibrate_OCO	補正プロセス
R_Setup_OCO	FITL1（キャプチャ用インターバルタイマーカウントクロック）のクロックおよびトリミング範囲（最小、最大）を設定する
R_Trimming_OCO	LOCO/MOCO クロック補正を実行する

3.3 関数のフローチャート

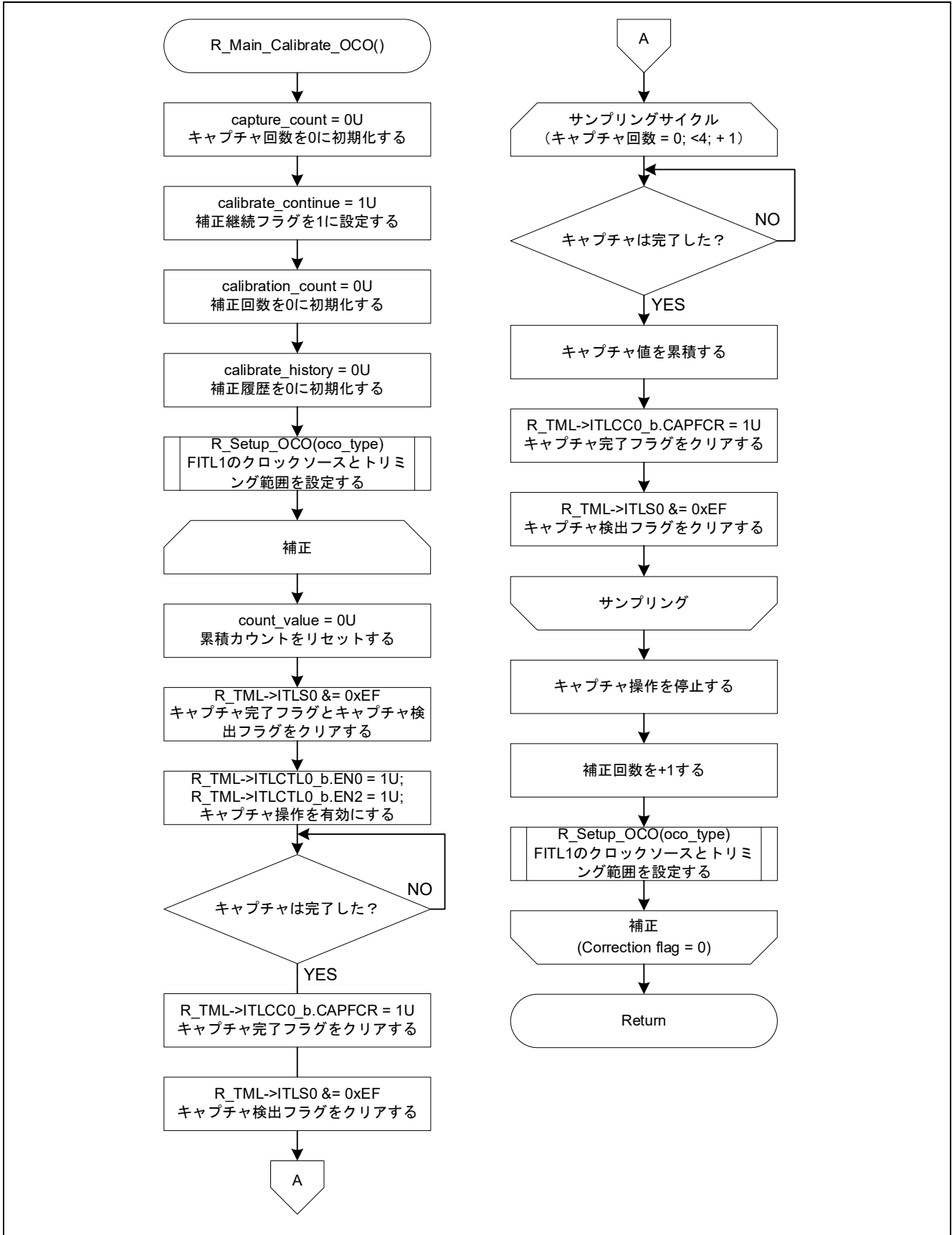
OCO 補正プロセスには、R_Trimming_OCO、R_Main_Calibrate_OCO、および R_Setup_OCO の 3 つの関数が含まれます。これらの関数の詳細は、後述のセクションで説明します。

このサンプルコードでは、FSP の設定と API を使用して TML32 モジュールを初期化します。しかし、LOCO および MOCO の周波数検出および補正手順の大部分は、レジスタへの直接アクセスに依存します。このアプローチにより、周波数補正機能はより効果的な結果を得ることができます。

3.3.1 Calibration Process

図 3-4 に LOCO および MOCO 補正プロセスのフローチャートを示します。

図 3-4. LOCO/MOCO 補正プロセス



LOCO および MOCO 補正関数

図 3-5 と図 3-6、図 3-7 に LOCO/MOCO 補正関数のフローチャートを示します。

図 3-5. LOCO/MOCO 補正関数 (1/3)

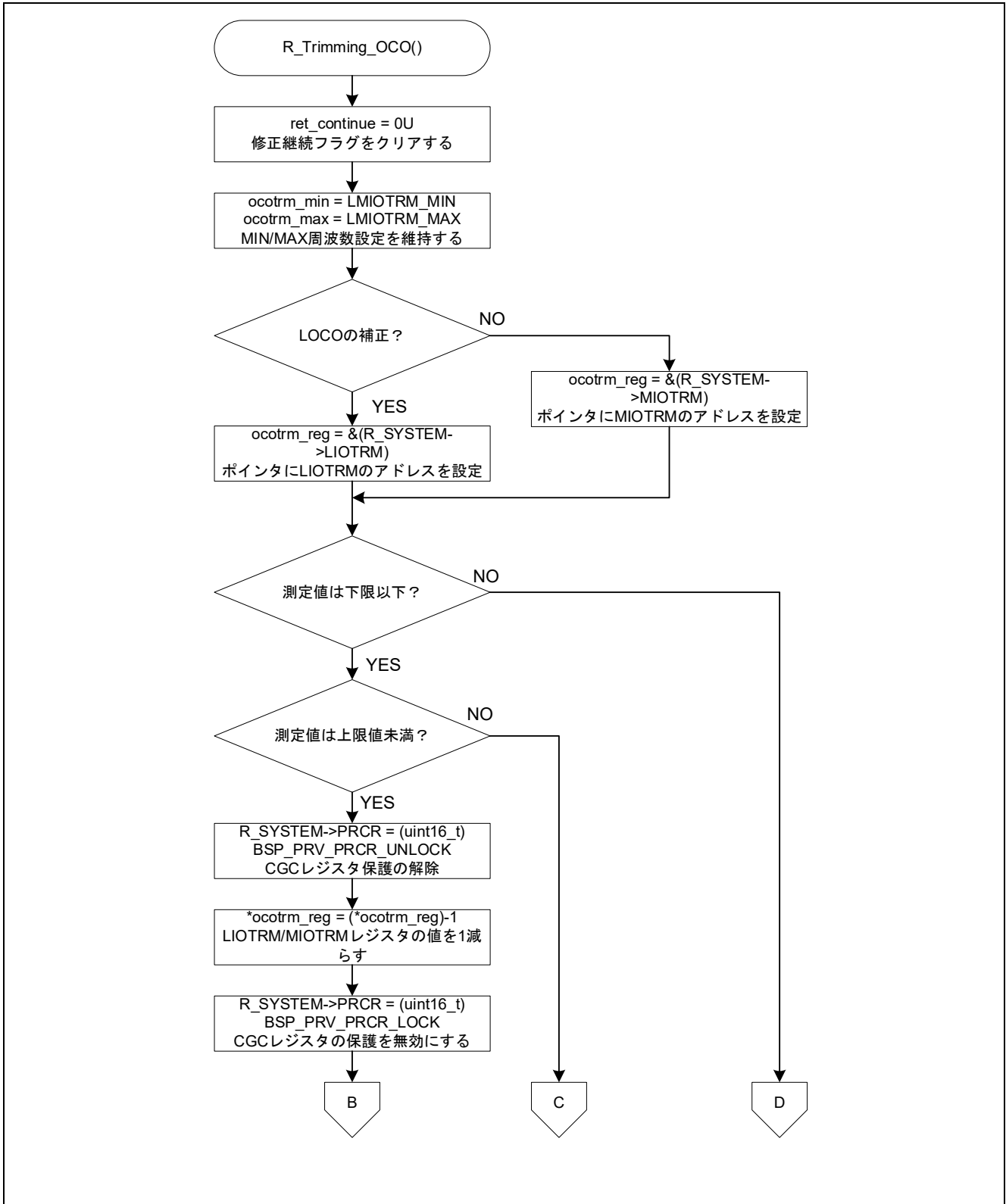


図 3-6. LOCO/MOCO 補正関数 (2/3)

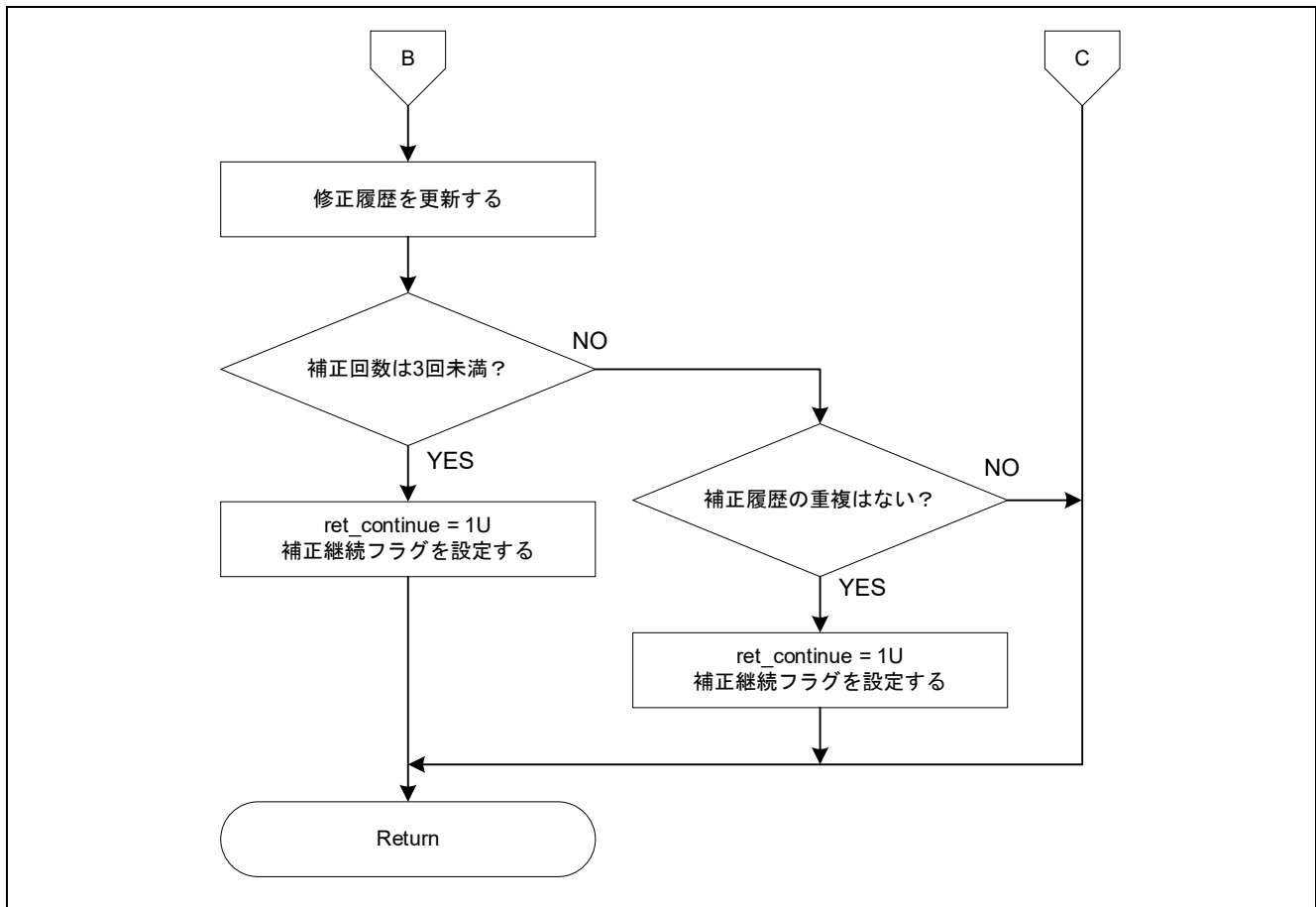
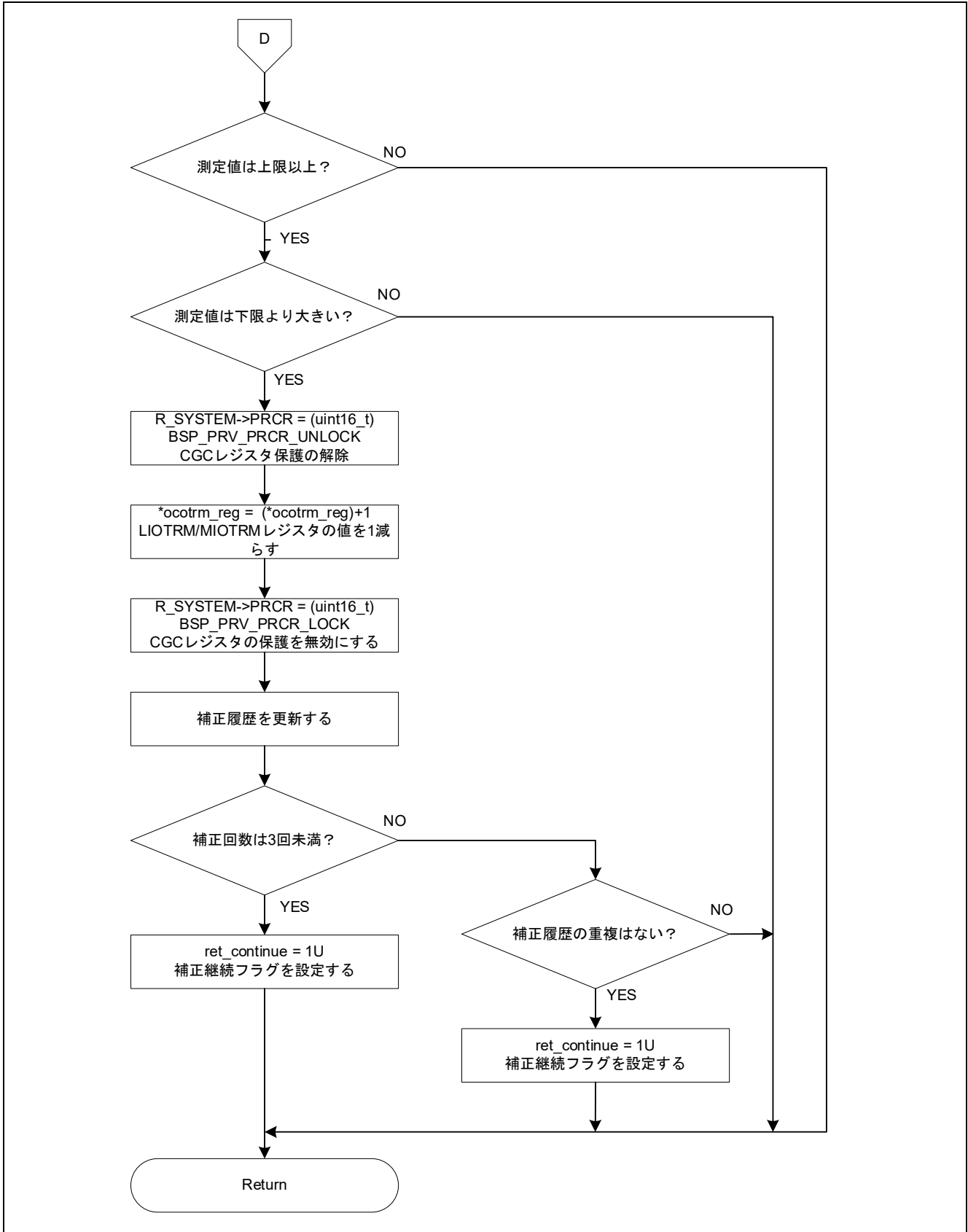


図 3-7. LOCO/MOCO 補正関数 (3/3)

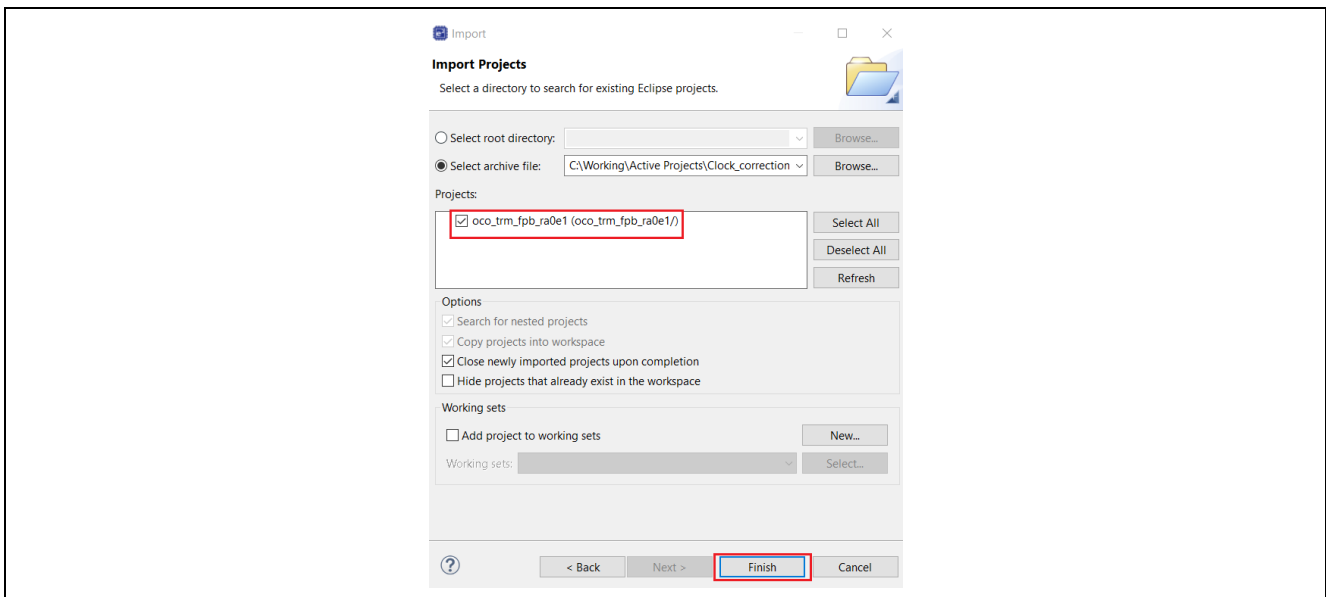


4. サンプルプロジェクト

4.1 プロジェクトのインポート

1. e²studio を起動します。
2. 任意のワークスペースを選択します。
3. 「ようこそ」ウィンドウを閉じます。
4. ファイル > インポートを選択します。
5. インポートダイアログボックスから「既存のプロジェクトをワークスペースに追加」を選択します。
6. zip ファイル「oco_trm_fpb_ra0e1.zip」を選択してください”
7. 下図のように、開発済みプロジェクトサンプルを選択し、「Finish」をクリックします。

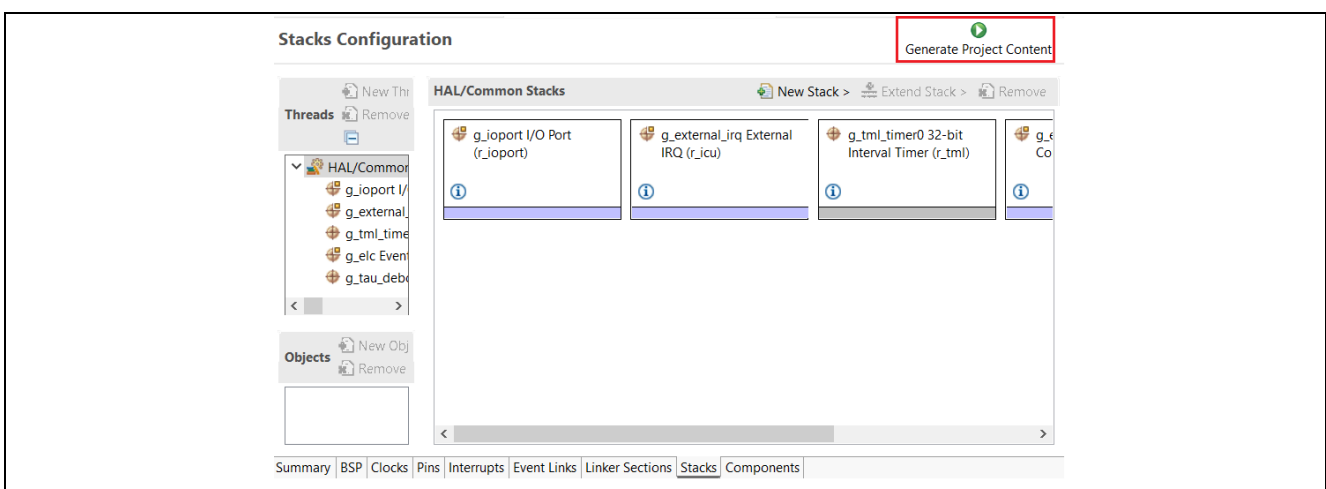
図 4-1. アプリケーションプロジェクトをワークスペースへインポート



4.2 プロジェクトのビルド

プロジェクト内の configuration.xml を開き、「Generate Project Content」をクリックし、プロジェクトを生成します。

図 4-2. プロジェクトの生成

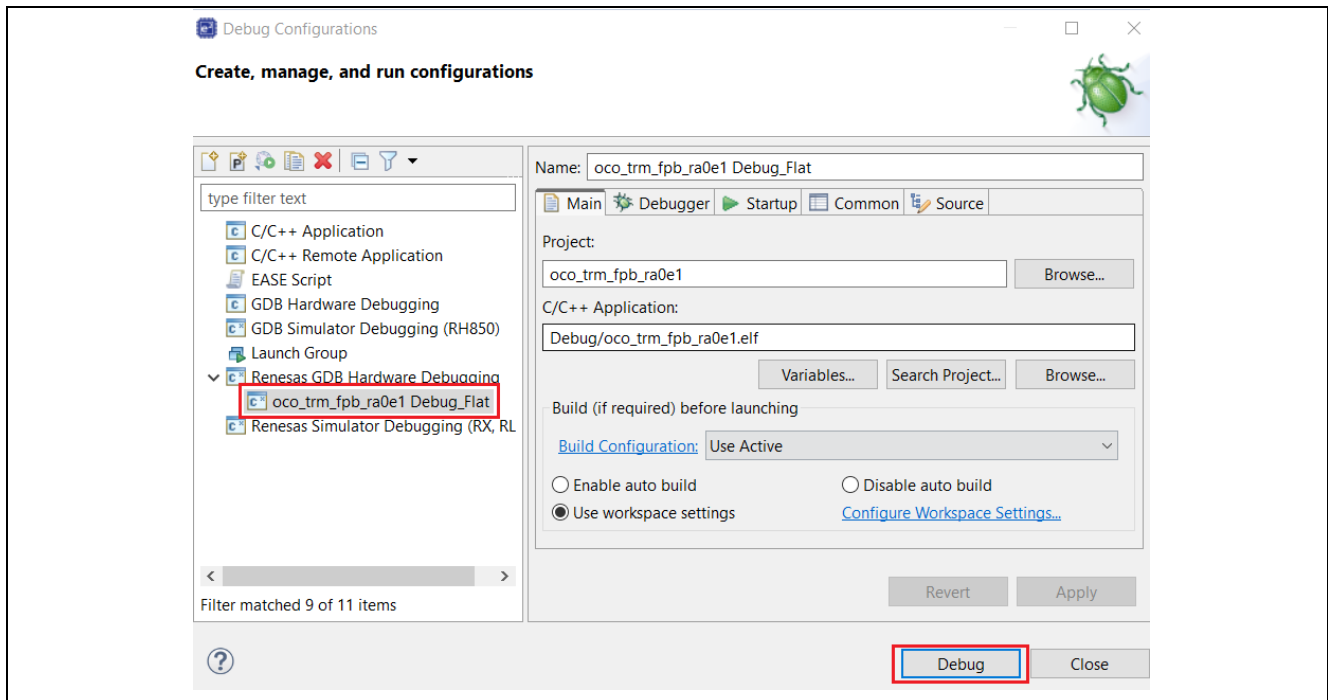


プロジェクトを生成したら、プロジェクトを右クリックし、「プロジェクトをビルド」を選択します。ビルドログコンソールを確認し、ビルドが正常に完了したことを確認してください。

4.3 プロジェクトのダウンロードと実行

FPB-R0E1 ボードを PC の USB ポートに接続します。メニューから「実行」>「デバッグ構成」を選択し、「oco_trm_fpb_ra0e1 Debug_Flat」を選んでプロジェクトを起動します。

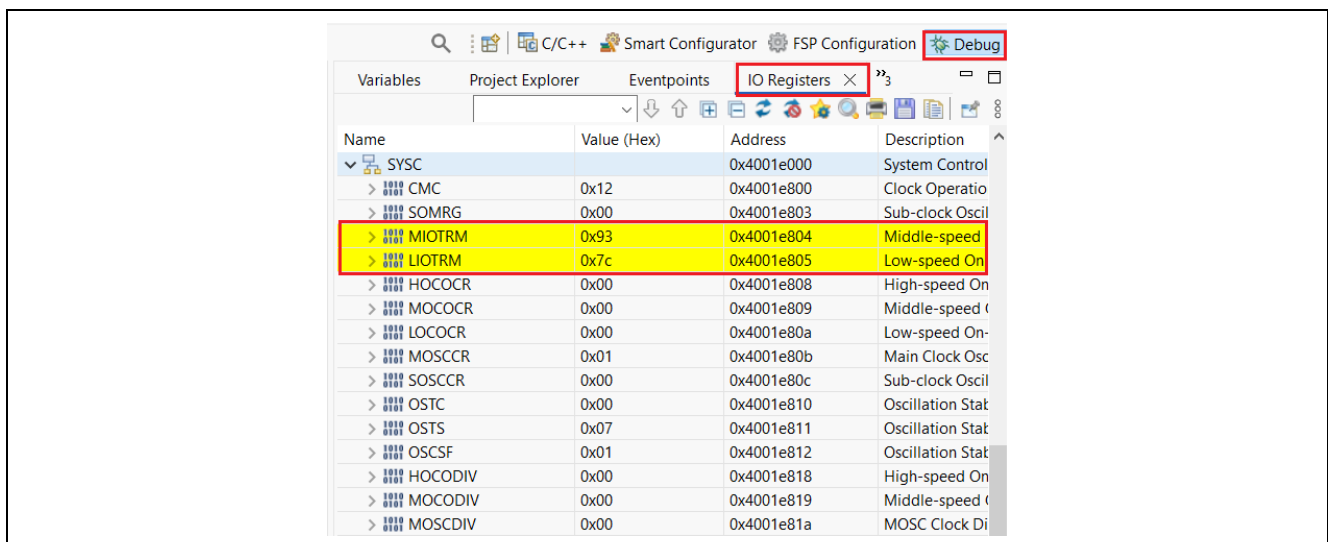
図 4-3. デバッグ開始



4.4 周波数補正結果の確認

サンプルプロジェクトをデバッグする際は、「IO Registers」タブを使用して、LIOTRM および MIOTRM のレジスタ値を確認してください。レジスタ値を、MCU ユーザーマニュアルに記載されているデフォルト設定と比較してください。

図 4-4. 周波数補正後の LIOTRM および MIOTRM レジスタの例



P407 で LOCO をできます。図 4-5 および図 4-6 に示すように、オシロスコープで信号を測定し、補正後の周波数を補正前の周波数と比較して確認することができます。

図 4-5. 周波数補正後の P407 における LOCO 出力の例

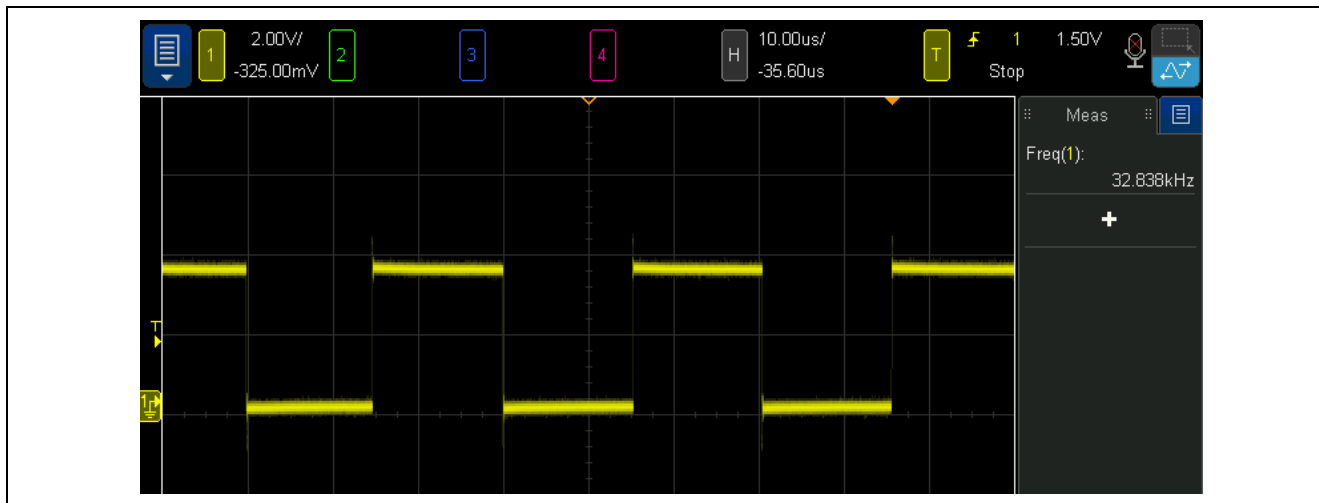
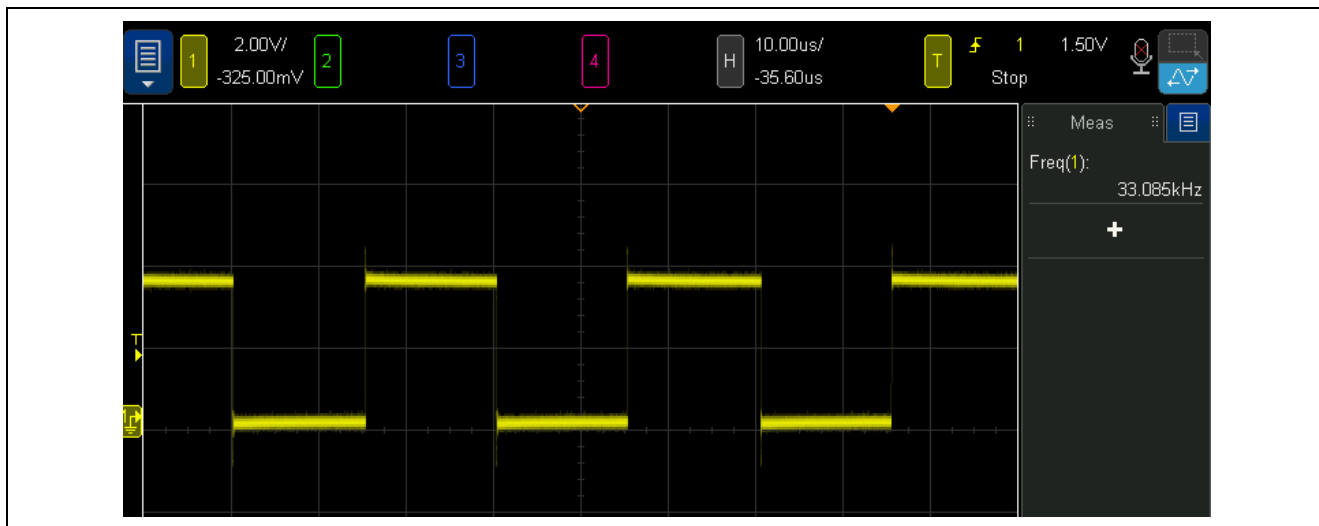


図 4-6. 周波数補正前の P407 における LOCO 出力の例



Website and Support

Visit the following vanity URLs to learn about key elements of the RA family, download components and related documentation, and get support.

RA Product Information	www.renesas.com/ra
RA Product Support Forum	www.renesas.com/ra/forum
RA Flexible Software Package	www.renesas.com/FSP
Renesas Support	www.renesas.com/support

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	2026.4.2	-	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. 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ーやマガジンケース、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しないでください。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、リセットを解除してください。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、 $V_{IL}(\text{Max.})$ から $V_{IH}(\text{Min.})$ 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、 $V_{IL}(\text{Max.})$ から $V_{IH}(\text{Min.})$ 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、ノイズ放射量などが異なる場合があります。型名が異なる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まれます。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）、交通管制（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、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いたします。
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（豊洲フォレストシア）

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。