
RX ファミリ

R01AN3610JJ0100

Rev.1.00

静電容量計測精度向上のためのセンサ補正

2017.1.18

要旨

本アプリケーションノートは、内部電流制御発振器（以下 **ICO**）の変動を補正することにより、静電容量式タッチセンサユニット（以下 **CTSU**）の静電容量検出精度を向上させる手順を説明します。本補正は、補正係数を計測データから算出し、**CTSU** のセンサカウンタ値に乗算することで精度向上を図ります。

動作確認デバイス

RX113、RX130、RX231、RX230

目次

1. 概要.....	3
2. ICOゲイン補正.....	6
2.1 理論値.....	7
2.2 ICO補正レジスタ.....	8
2.3 ICO補正計測.....	9
2.3.1 第1補正計測.....	9
2.3.2 第2補正計測.....	10
2.3.3 補正係数演算.....	12
3. ICO補正ソフトウェア説明.....	15
3.1 ファイル構成.....	15
3.2 オプション設定メモリ.....	15
3.3 定数一覧.....	16
3.4 関数一覧.....	17
3.5 関数仕様.....	18
3.5.1 main.....	18
3.5.2 R_Set_Cap_Touch_Create.....	18
3.5.3 correcton_CTSU_sensor_ico.....	19
3.5.4 correction_CTSU_sensor_selection.....	19
3.5.5 correction_CTSU_register_parameter_set.....	20
3.5.6 correction_CTSU_register_txd_set.....	21
3.5.7 DTC_Set_Initial_of_CTSU.....	21
3.5.8 correction_CTSU_register_set.....	22
3.5.9 R_Set_CTSU_Correction_Mode.....	22
3.5.10 correction_CTSU_measurement_start.....	23
3.5.11 correction_CTSU_1st_coefficient_create.....	23
3.5.12 correction_CTSU_2nd_standard_val_create.....	24
3.5.13 correction_CTSU_2nd_coefficient_create.....	24
3.5.14 correction_CTSU_16point_coefficient_create.....	25
3.5.15 correction_sensor_magnification_set.....	25
3.5.16 correction_sensor_cnt_create.....	25
4. フローチャート.....	26
4.1 Main関数 : main().....	26
4.2 静電容量タッチ生成関数 : R_Set_Cap_Touch_Create().....	27
4.3 センサICO補正関数 : correcton_CTSU_sensor_ico().....	28

1. 概要

CTSU を使用した静電容量測定は、ICO の動作に依存します。CTSU 回路の動作は、R30AN0218 に詳述されています。CTSU は、静電容量値を示すカウンタ値を出力します。しかし、CTSU 内の ICO 特性変動により、同じ条件下で動作する 2 つの MCU は、同一の出力を生成しません。計測される変化誤差は、最大約 20% 存在します。このアプリケーションノートでは、これらのバリエーションについて説明し、補正する手順についても説明します。静電容量を測定カウント値に変換する概要を図 1.1 に示します。

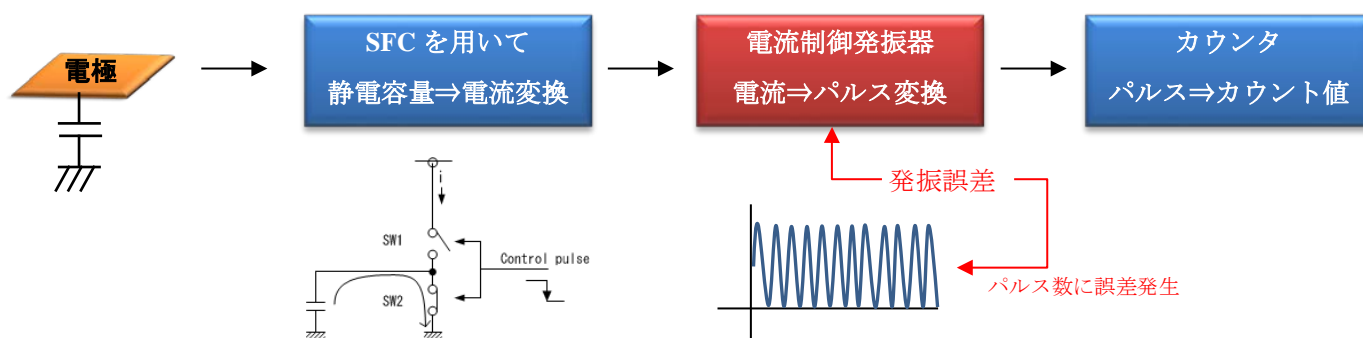


図 1.1 静電容量変換概要図

図 1.1 に示した ICO の発振誤差補正は、異なる 2 つの電流値から 2 値を求め補正係数を算出し、計測カウント値を補正します。図 1.2 と図 1.3 に補正データ図を示します。

横軸：電流値 縦軸：カウント値

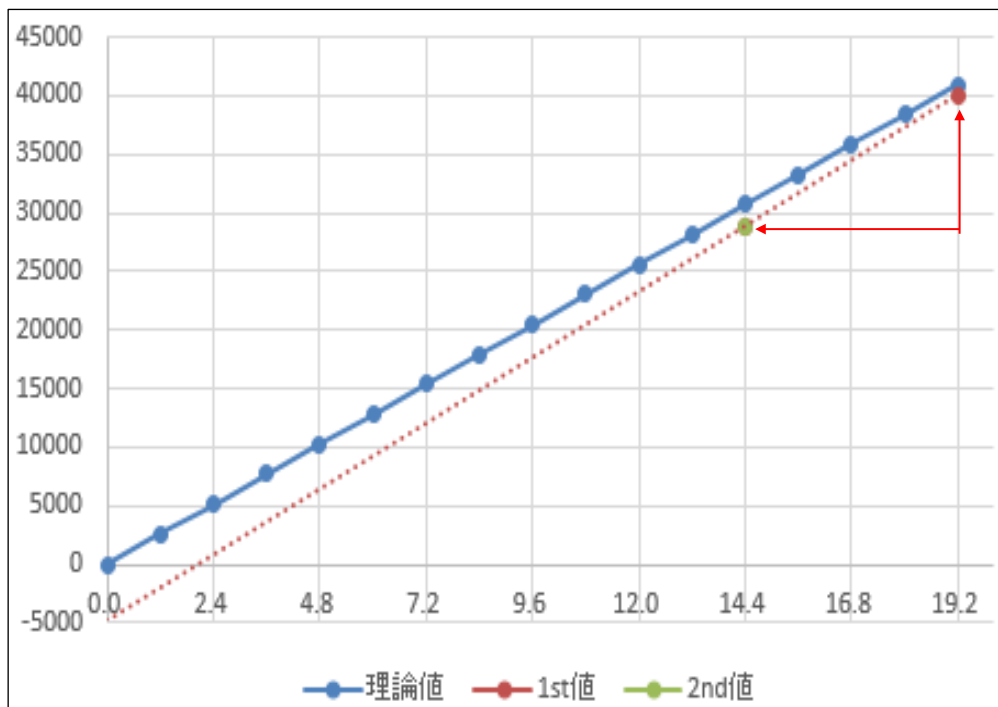


図 1.2 補正係数

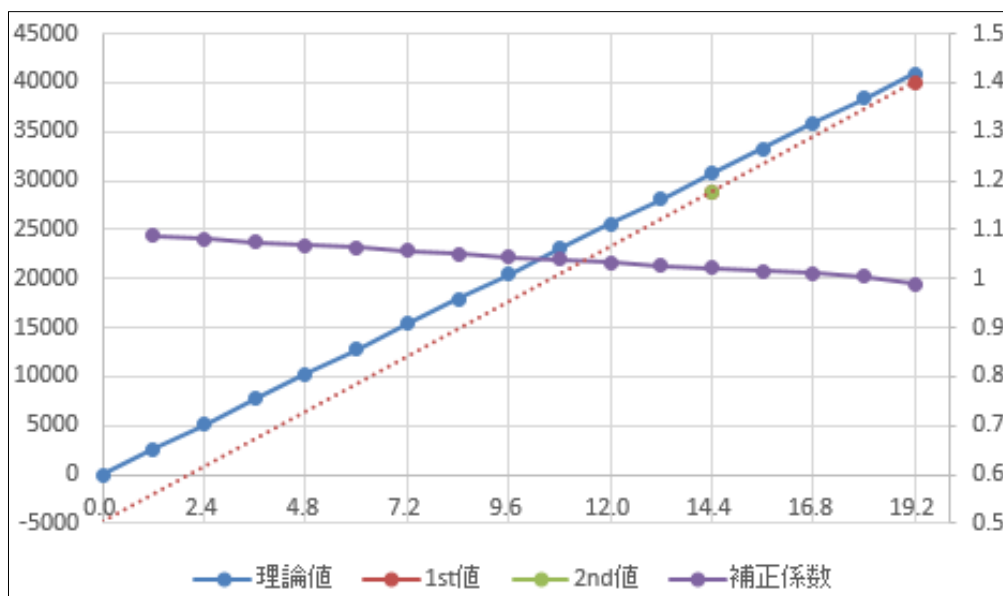


図 1.3 補正係数

得られた補正係数を計測カウント値に掛けることで、補正を行います。

従来の ICO 補正処理は ICO に供給する電流量を電流オフセット機能によりリファレンス ICO カウンタと合致させていました。この方法では計測される静電容量のオフセット量は補正する事が出来ませんが、MCU 毎に異なる ICO のゲイン特性（傾き）は補正する事が出来ません。1 次関数として表現しますと $y = ax + b$ におけるオフセット値 b は補正出来ますが、ゲイン特性の係数 a は補正出来ません。

y : 補正カウント値

a : MCU 毎のゲイン特性

x : 計測カウント値

b : MCU 毎の計測オフセット値

今回の補正は、寄生容量の違いによる電流オフセット、カウント誤差の補正を行う前に MCU 毎の ICO が持っている発振誤差約±20%を補正（ a を補正）し精度向上を行います。静電容量と計測カウント値の関係を4つのサンプルにて計測した例を図 1.4 に示します。このようにサンプル毎にゲイン特性とオフセット値が異なります。

本アプリケーションノートでは MCU 毎の ICO 特性の違いをキャリブレーション機能とソフトウェア補正により精度向上する方法について説明します。

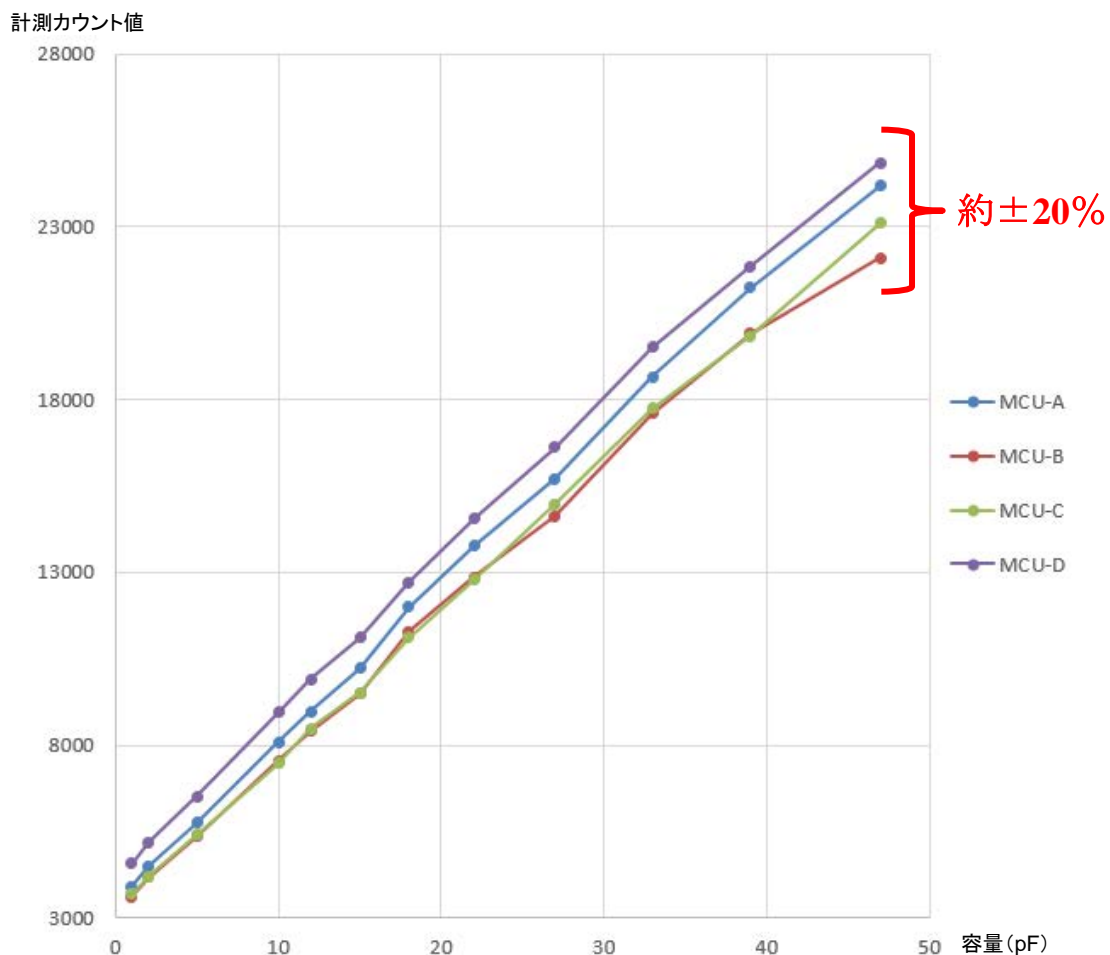


図 1.4 MCU 間誤差

2. ICO ゲイン補正

ICO ゲイン補正を実行するために必要なアクションを図 2.1 に示します。ICO ゲイン補正は、CTSUS が初期化中に「キャリブレーションモード」で動作することを必要とします。補正係数が計算されると、CTSUS は通常動作モードに遷移し、測定されたすべてのカウンタ値に補正係数が乗算します。

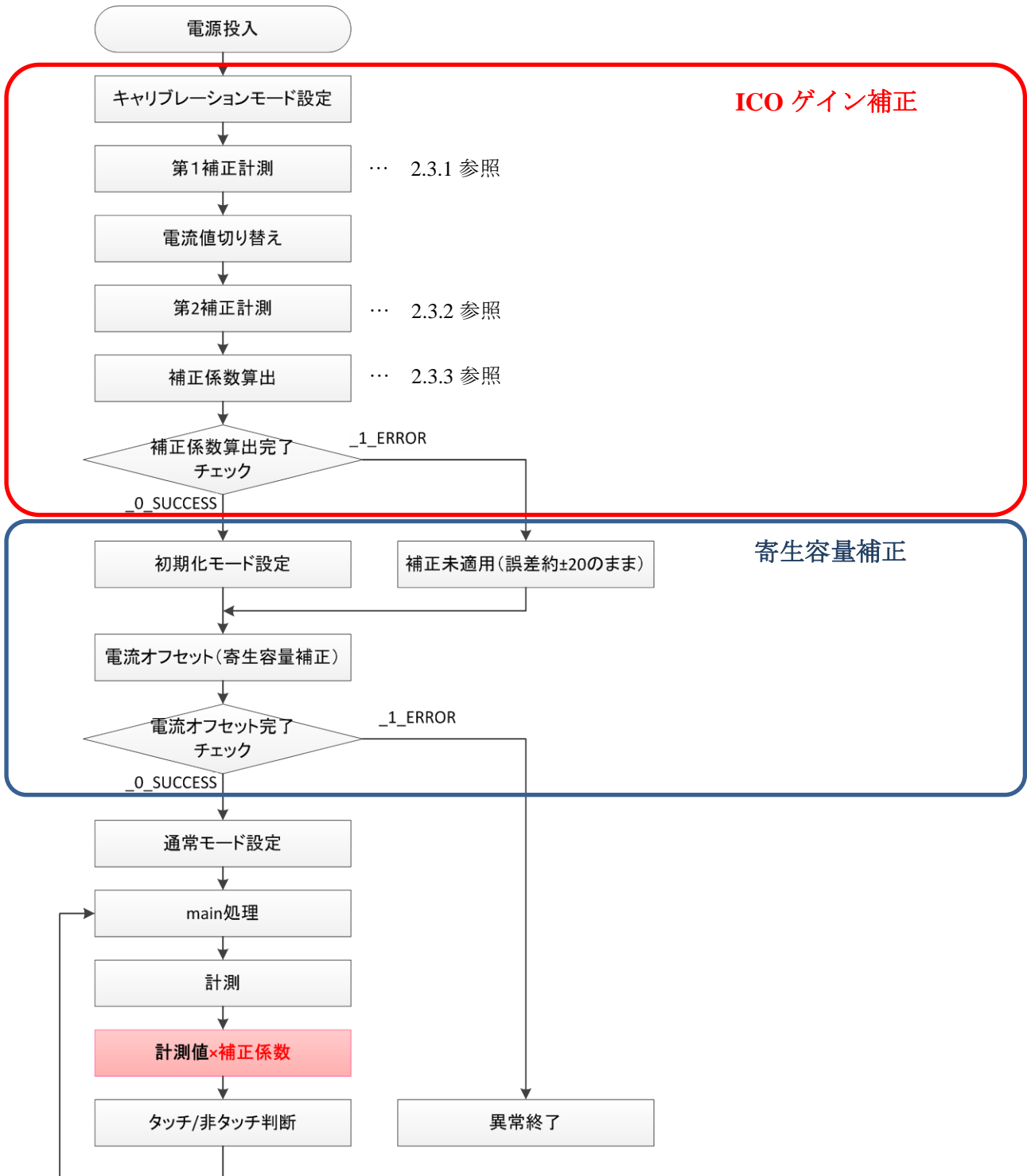


図 2.1 補正概略フローチャート

2.1 理論値

ICO ゲイン補正は、計測時間 0.526ms の時間単位あたりに、供給される電流値から出力するパルス数=カウント値の理論値を基準に補正を行います。その理論値を表 2.1 と図 2.2 に示します。

計測時間は、計測周波数と計測回数によって決定されます。

表 2.1 理論値

電流値 (uA)	カウント値	電流値 (uA)	カウント値
0.0	0	10.8	23040
1.2	2560	12.0	25600
2.4	5120	13.2	28160
3.6	7680	14.4	30720
4.8	10240	15.6	33280
6.0	12800	16.8	35840
7.2	15360	18.0	38400
8.4	17920	19.2	40960
9.6	20480	-	-

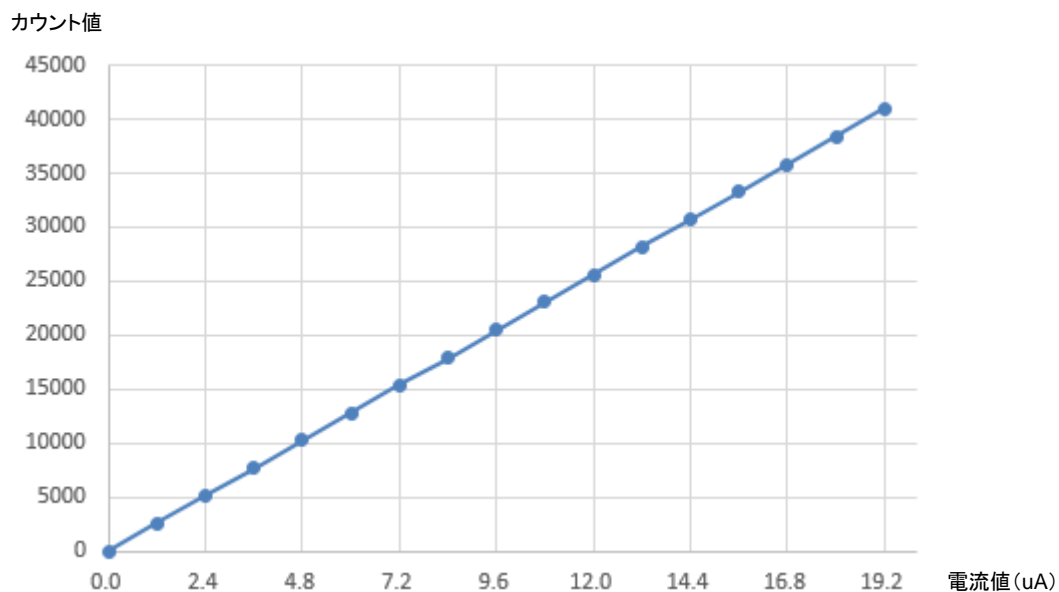


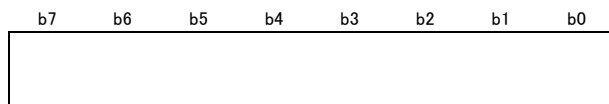
図 2.2 理論値

2.2 ICO 補正レジスタ

ICO 補正計測は、CTSU 基準電流調整レジスタ（以下：CTSUTRMR）と CTSU エラーステータスレジスタ（以下：CTSUERRS）を用いて行います。CTSUTRMR と CTSUERRS を下記に示します。

CTSU 基準電流調整レジスタ (CTSUTRMR)

アドレス 007F FFBEh



リセット値

チップごと固有値

CTSUTRMRレジスタには、工場出荷時に個々のチップごとに一定の条件で調整した基準電流値が書き込まれています。

CTSU エラーステータスレジスタ (CTSUERRS)

アドレス 000A 091Ch

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CTSUICOMP	-	-	-	-	-	-	-	CTSUTSOC	-	-	-	CTSUDRV	CTSUTSOD	CTSUSPMD	
リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ビット	シンボル	ビット名	機能	R/W
b1-b0	CTSUSPMD	キャリブレーションモードビット	b1 b0 00 : 静電容量計測モード 01 : 設定禁止 10 : キャリブレーションモード 11 : 設定禁止	R/W
b2	CTSUTSOD	TS 端子固定出力ビット	0 : 静電容量計測モード 1 : TS 端子固定出力 (High/Low 出力)	R/W
b3	CTSUDRV	キャリブレーションモードビット設定 3	0 : 静電容量計測モード 1 : キャリブレーション設定 3	R/W
b6-b4	-	予約ビット	読むと“0”が読めます。書く場合、“0”としてください	R/W
b7	CTSUTSOC	キャリブレーションモードビット設定 7	0 : 静電容量計測モード 1 : キャリブレーション設定 7	R/W
b14-b8	-	予約ビット	読むと“0”が読めます。書く場合、“0”としてください	R/W
b15	CTSUICOMP	TSCAP 電圧異常監視ビット	0 : TSCAP 電圧正常 1 : TSCAP 電圧異常	R

ICO 補正は、CTSUERRS の CTSUSPMD (b1:b0) に'10'を設定し、キャリブレーションモード設定を行います。その後、CTSUTRMR をデフォルト値 (19.2uA) と 0xFF の異なる 2 つの電流値にて計測を行い、補正係数を算出します。

2.3 ICO 補正計測

2.3.1 第1補正計測

第1補正計測は、CTSUTRMR をデフォルト値のまま計測を開始します。デフォルト設定は、ICO へ必ず 19.2uA の電流を供給します。この計測で得られた計測値を、1st 値とします。また、**図 2.3** に示します。

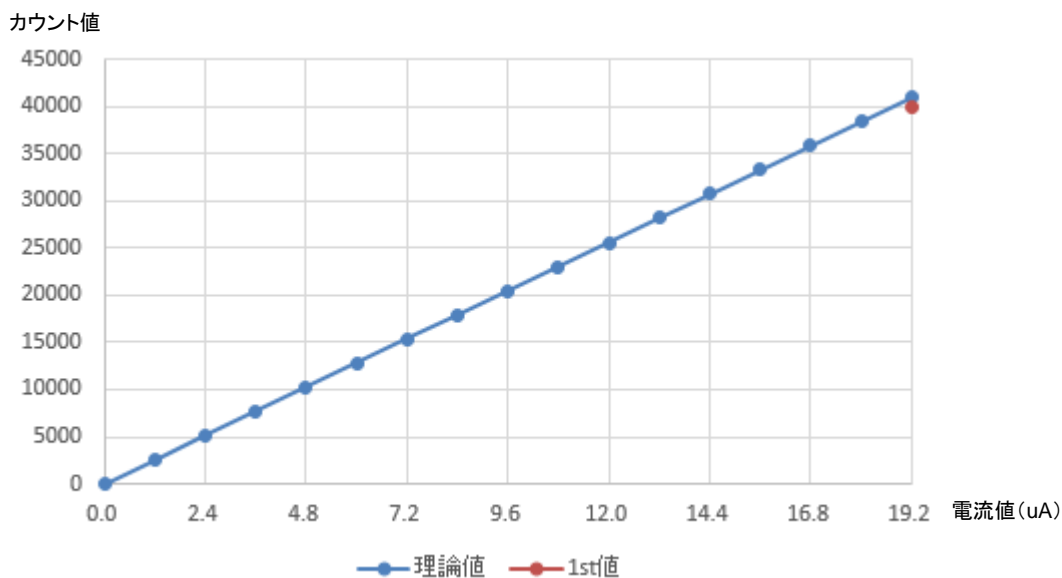


図 2.3 1st 値

図 2.3 に示した 1st 値は、19.2uA の理論値に対して、発振数が低い ICO の例となります。

また、発振数が高い ICO も存在しますが、今回は発振数が低い ICO を例として説明します。

2.3.2 第2補正計測

第2補正計測は、ICOへ供給電流を変更し計測を行います。そのため、CTSUTRMRに0xFFを書き込み、供給電流を変更します。

CTSUTRMRは、工場出荷時にCTSUSへ供給する電流を19.2uAとするために、トリミングされMCUごとに、異なります。

今回の補正は、異なる2点間の電流値から傾きを求め補正を行います。その際に、第1補正計測時の19.2uAから、大きく電流差をとり第2補正計測をすることで、補正係数の精度を向上させます。そのため、CTSUTRMRに設定可能な最大値0xFFの値を入力し、最大の電流差にて計測を行います。

※CTSUS基準電流調整レジスタ値は、設定する値を大きくすると電流は減少します。

CTSUTRMRを変更後、第1補正計測と同様に計測を行い、カウント値を得ます。これを2nd値とします。また、図2.4に示します。

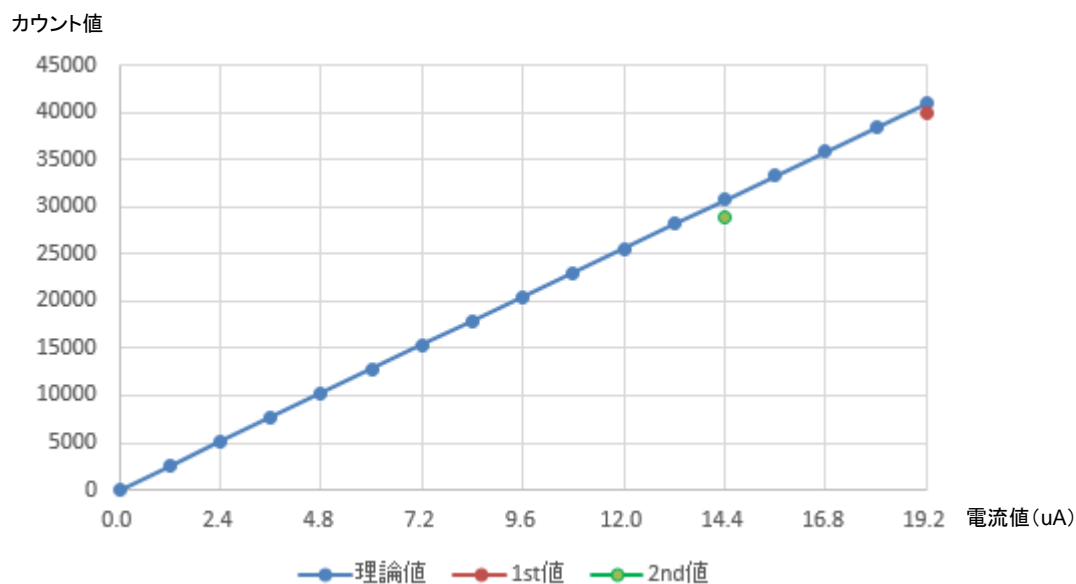


図 2.4 2nd 値

1st 値と 2nd 値のカウント結果から、1.2uA~19.2uA の 16 点補正係数を算出し、ICO の補正を行います。図 2.5 に補正係数を示します。

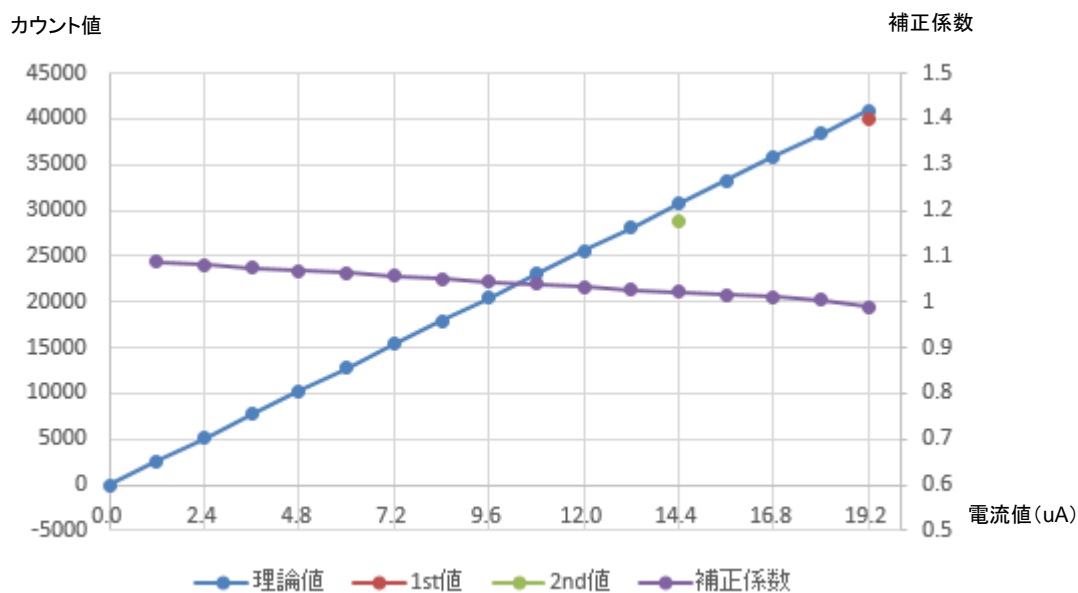


図 2.5 補正係数

2.3.3 補正係数演算

1.2uA～19.2uA の 16 点補正係数算出までの計算を下記に記載します。

■19.2uA 理論値に対する倍率演算

$$1\text{st 倍率} = \frac{1\text{st 値}}{19.2\text{uA 理論値}} = \frac{40040}{40960} = 0.978$$

表 2.1 を参照

■2nd の補正基準データ演算

$$2\text{nd の補正基準データ} = \frac{(1\text{stCTSUTRMR 値}+273)}{528} \times 19.2\text{uA 理論値} = \frac{(116+273) \times 40960}{528} = 30176$$

■2nd の補正基準データに対する倍率演算

$$2\text{nd 倍率} = \frac{2\text{nd 値}}{2\text{nd の補正基準データ}} = \frac{28807}{30176} = 0.954$$

■1.2uA～19.2uA までの 16 点係数算出

$$\begin{aligned} 1\text{st 倍率と } 2\text{nd 倍率の差} &= 1\text{st 倍率} - 2\text{nd 倍率} \quad (\text{または、} 2\text{nd 倍率} - 1\text{st 倍率}) \\ &= 1001 - 978 = 23 \end{aligned}$$

$$1\text{ 点あたりの変化率} = \frac{1\text{st 倍率と } 2\text{nd 倍率の差} \times 19.2\text{uA 理論値}}{19.2\text{uA 理論値} - 2\text{nd の補正基準データ}} = \frac{23 \times 40960}{40960 - 30176} = 87$$

$$\begin{aligned} 16\text{ 点倍率} &= (1\text{st 倍率} + \frac{1\text{ 点あたりの変化率} \times N}{16}) / 1024 \quad \text{※}N \dots \text{ポイント番号} \\ &= (1001 + \frac{87 \times N}{16}) / 1024 \end{aligned}$$

注) サンプルコードでは、MCU で割り算処理の整数演算を記述する際に、端数切り捨てによる演算誤差を減らすため、除算前に小数点第三位の考慮と処理不可軽減のため 10 ビット左シフトを行い、さらに小数点以下切り捨てによる量子化誤差の対応のため、分母/2 を加算する処理が実装されております。

図 2.6 に、 $1.2\mu\text{A}$ の 16 間隔でゲインが変化する要因を示します。この「変化の要因」は「倍率値」と呼ばれます。倍率値の逆数は、測定されたセンサカウント値を予想される理想センサカウント値に補正するための乗数として使用されます。倍率の逆数が補正係数となります。その結果を表 2.2 に示します。

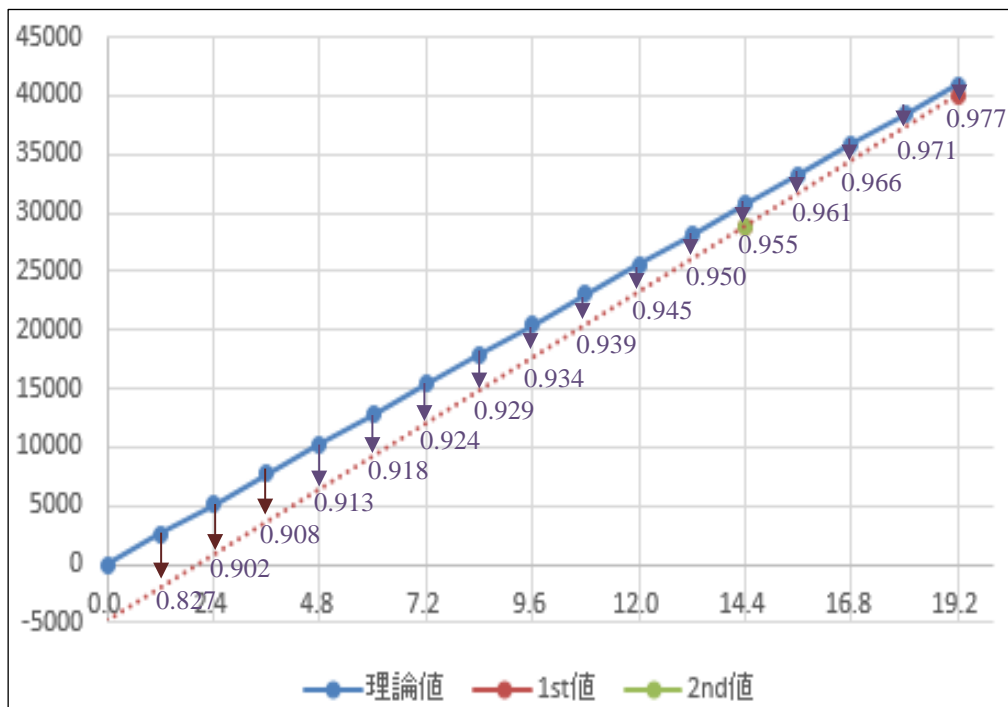


図 2.6 倍率値

表 2.2 補正係数

N	倍率	補正係数 (1/倍率)
0 (19.2uA)	0.977	1.023
1 (18.0uA)	0.971	1.029
2 (16.8uA)	0.966	1.034
3 (15.6uA)	0.961	1.040
4 (14.4uA)	0.955	1.046
5 (13.2uA)	0.950	1.052
6 (12.0uA)	0.945	1.058
7 (10.8uA)	0.939	1.064
8 (9.6uA)	0.934	1.070
9 (8.4uA)	0.929	1.076
10 (7.2uA)	0.924	1.082
11 (6.0uA)	0.918	1.088
12 (4.8uA)	0.913	1.094
13 (3.6uA)	0.908	1.101
14 (2.4uA)	0.902	1.107
15 (1.2uA)	0.897	1.114

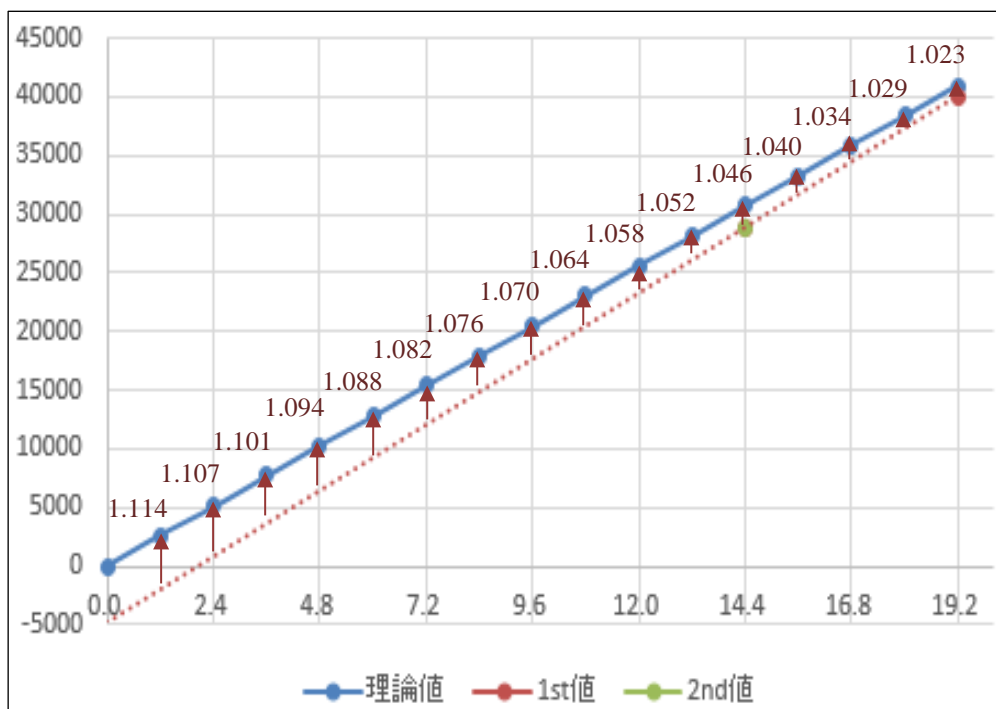


図 2.7 補正係数

3. ICO 補正ソフトウェア説明

3.1 ファイル構成

表 3.1 に ICO 補正処理のファイルを示します。なお、統合開発環境で自動生成されるファイルは除きます。

表 3.1 使用ファイル

ファイル名	概要	備考
r_main.c	メイン処理ソースファイル	
r_ctsu.c	CTSU 制御ソースファイル	
r_ctsu_physical_driver.c	CTSU フィジカルドライバソースファイル	
r_dtc.c	DTC 制御ソースファイル	
r_touch_API.c	タッチ API ソースファイル	
r_ctsu.h	CTSU 制御ヘッダファイル	
r_dtc.h	DTC 制御ヘッダファイル	
r_touch_API.h	タッチ API ヘッダファイル	

3.2 オプション設定メモリ

表 3.2 にオプション設定メモリの状態を示します。必要に応じて、お客様のシステムに最適な値を設定してください。

表 3.2 オプション設定メモリ

マイコン	シンボル	アドレス	設定値	内容
RX113	DTC.DTCVBR	0x00007C00	dtc_vector60 (0x00007CF0)	DTC 転送アドレス
RX231			dtc_vector61 (0x00007CF4)	
RX230				
RX130		0x00002400	dtc_vector60 (0x000024F0) dtc_vector61 (0x000024F4)	

3.3 定数一覧

表 3.3 に ICO 補正に用いる定数一覧を示します。

表 3.3 定数一覧 (r_ctsu.h)

r_ctus.h		
定数名	設定値	内容
CTSU_INPUT_FREQUENCY	32000000 (デフォルト)	PCLKB へ供給される動作クロック (Hz) 32000000 (=32MHz) 27000000 (=27MHz) 24000000 (=24MHz) 16000000 (=16MHz) 8000000 (=8MHz)
CTSU_INPUT_FREQUENCY_DIV	32 (デフォルト)	CTSU_INPUT_FREQUENCY/ 1000000
CORRECTION_AVERAGE	32	キャリブレーション計測値平均回数
_0_NORMAL	0	通常計測モード
_1_CORRECTION	1	キャリブレーション計測モード
_0_1ST	0	1st 計測フラグ
_1_2ND	1	2nd 計測フラグ
_19_2UA	40960	19.2uA 供給時の補正基準データ
_18_0UA	38400	18.0uA 供給時の補正基準データ
_16_8UA	35840	16.8uA 供給時の補正基準データ
_15_6UA	33280	15.6uA 供給時の補正基準データ
_14_4UA	30720	14.4uA 供給時の補正基準データ
_13_2UA	28160	13.2uA 供給時の補正基準データ
_12_0UA	25600	12.0uA 供給時の補正基準データ
_10_8UA	23040	10.8uA 供給時の補正基準データ
_09_6UA	20480	9.6uA 供給時の補正基準データ
_08_4UA	17920	8.4uA 供給時の補正基準データ
_07_2UA	15360	7.2uA 供給時の補正基準データ
_06_0UA	12800	6.0uA 供給時の補正基準データ
_04_8UA	10240	4.8uA 供給時の補正基準データ
_03_6UA	7680	3.6uA 供給時の補正基準データ
_02_4UA	5120	2.4uA 供給時の補正基準データ
_01_2UA	2560	1.2uA 供給時の補正基準データ

3.4 関数一覧

表 3.4 に関数一覧を示します。

表 3.4 関数一覧

関数名	概要
main()	システムメイン関数
correction_CTSU_sensor_ico()	ICO 補正関数
correction_CTSU_sensor_selection()	ICO 補正用のセンサ選択関数
correction_CTSU_register_parameter_set()	ICO 補正 CTSU レジスタ値設定関数
correction_CTSU_register_txd_set()	ICO 補正転送用計測パラメータ設定関数
DTC_Set_Initial_of_CTSU()	CTSU の DTC 初期化関数
correction_CTSU_register_set()	ICO 補正 CTSU レジスタ設定関数
correction_CTSU_measurement_start()	ICO 補正計測開始関数
correction_CTSU_1st_coefficient_create()	補正基準データに対する 1st 倍率を算出関数
correction_CTSU_2nd_standard_val_create()	2nd の補正基準データを算出関数
correction_CTSU_2nd_coefficient_create()	補正基準データに対する 2nd 倍率を算出関数
correction_CTSU_16point_coefficient_create()	1.2uA~19.2uA の補正係数算出関数
correction_sensor_magnification_set()	計測時間による補正係数の倍率設定関数
correction_sensor_cnt_create()	計測値補正生成関数
R_Set_Cap_Touch_Create()	静電容量タッチ生成関数
R_Set_CTSU_Correction_Mode()	ICO 補正計測モード設定関数

3.5 関数仕様

関数仕様を示します。

3.5.1 main

main()

概要	システムメイン関数
宣言	void main(void)
説明	この関数は、システム全体の制御を行います。
コール関数	R_Set_Cap_Touch_Create() R_CMT0_Create() R_CMT0_Start() R_Set_Cap_Touch_Measurement_Start() ※1 R_Get_Cap_Touch_Data_Check() ※1 R_Get_Cap_Touch_Initial_Status() ※1 R_Set_Cap_Touch_Result_Create() ※1 R_Get_Cap_Touch_Result() ※1 R_Set_Cap_Touch_Initial_Tuning() ※1 R_Set_Cap_Touch_Next_Method_Change() ※1

引数 - -

リターン値 -

※1 タッチ API リファレンスガイドを参照下さい。

3.5.2 R_Set_Cap_Touch_Create

R_Set_Cap_Touch_Create()

概要	静電容量タッチ生成関数
宣言	void R_Set_Cap_Touch_Create(void)
説明	この関数は、main 関数よりコールされ、静電容量タッチを計測するための設定を行います。
コール関数	correction_CTSU_sensor_ico() touch_parameter_address_set() CTSU_register_initial_value_ram_set() DTC_Set_Initial_of_CTSU() initial_port_set() R_Set_CTSU_All_Register() DTC_transmit_data_set() touch_parameter_set() R_Set_Cap_Touch_Tuning_Cntrol() R_Set_Cap_Touch_Offset_Timing()

引数 - -

リターン値 -

3.5.3 correcton_CTSU_sensor_ico

correcton_CTSU_sensor_ico()

概要	センサ ICO 補正関数
宣言	uint8_t correcton_CTSU_sensor_ico(void)
説明	この関数は、R_Set_Cap_Touch_Create 関数よりコールされ、センサ ICO の補正を実行します。
コール関数	correction_CTSU_sensor_selection() correction_CTSU_register_parameter_set() correction_CTSU_register_txd_set() DTC_Set_Initial_of_CTSU() correction_CTSU_register_set() R_Set_CTSU_Correction_Mode() correction_CTSU_measurement_start() correction_CTSU_1st_coefficient_create() correction_CTSU_2nd_standard_val_create() correction_CTSU_2nd_coefficient_create() correction_CTSU_16point_coefficient_create()
引数	-
リターン値	0x00 : 正常終了 0x01 : 異常終了

3.5.4 correction_CTSU_sensor_selection

correction_CTSU_sensor_selection()

概要	センサ補正用のセンサ選択関数
宣言	uint8_t correction_CTSU_sensor_selection(void)
説明	この関数は、correcton_CTSU_sensor_ico 関数よりコールされ、補正用のセンサを選択します。 補正用のセンサは、実際に使用されるセンサの中で一番小さい番号のセンサが選択されます。
コール関数	-
引数	-
リターン値	0x00 : 正常終了 (補正用センサ番号選択完了) 0x01 : 異常終了 (補正用センサ番号選択不能)

3.5.5 correction_CTSU_register_parameter_set

correction_CTSU_register_parameter_set()

概要 ICO 補正 CTSU レジスタ値設定関数
 宣言 **void** correction_CTSU_register_parameter_set(**void**)
 説明 この関数は、correcton_CTSU_sensor_ico 関数よりコールされ、補正用の CTSU レジスタ設定を行います。また、設定内容を以下に示します。

CTSU 制御レジスタ 0 (CTSUCR0)		
シンボル名	設定値	設定内容
CTSUSNZ	0	待機時省電力機能無効
CTSUTXVSEL	0	Vcc を選択

CTSU 制御レジスタ 1 (CTSUCR1)		
シンボル名	設定値	設定内容
CTSUPON	1	電源 ON
CTSUCSW	1	容量スイッチ ON
CTSUATUNE0	0	通常動作モード
CTSUATUNE1	0	通常出力
CTSUCLK	0	PCLK (分周無し)
CTSUMD	1	自己容量マルチスキャンモード

CTSU 同期ノイズ低減設定レジスタ (CTSUSDPRS)		
シンボル名	設定値	設定内容
CTSUPRRATIO	3	推奨値
CTSUPRMODE	2	62 パルス (推奨値)
CTSUOFF	0	高域ノイズ低減機能 ON

CTSU センサ安定待ち時間レジスタ (CTSUSST)		
シンボル名	設定値	設定内容
CTSUSST	16	固定値

CTSU 同期ノイズ低減設定レジスタ (CTSUSDPRS)		
シンボル名	設定値	設定内容
CTSUSSMOD	0	固定値
CTSUSSCNT	3	固定値

コール関数 -
 引数 -
 リターン値 -

3.5.6 correction_CTSU_register_txd_set

correction_CTSU_register_txd_set()

概要 ICO 補正転送用計測パラメータ設定関数
 宣言 **void** correction_CTSU_register_txd_set(**void**)
 説明 この関数は、correcton_CTSU_sensor_ico 関数よりコールされ、動作クロックに従って、静電容量計測周波数（スイッチドキャパシタフィルタ周波数）を 0.5MHz、計測時間 0.5ms の設定を行います。また、電流オフセットの設定を 0 にします。

CTSU 高域ノイズ低減スペクトラム拡散制御レジスタ (CTSUSSC) = 0x0700
 CTSU センサオフセットレジスタ 0 (CTSUSO0) = 0x0000

動作クロック	CTSU センサオフセットレジスタ 1 (CTSUSO1)		
	CTSURICOA	CTSUSDPA	CTSUICOG
32MHz	15	31 (64 分周)	1
27MHz		26 (54 分周)	
24MHz		23 (48 分周)	
16MHz		15 (32 分周)	
8MHz		7 (16 分周)	

コール関数 -
 引数 -
 リターン値 -

3.5.7 DTC_Set_Initial_of_CTSU

DTC_Set_Initial_of_CTSU()

概要 CTSU の DTC 初期化関数
 宣言 **void** DTC_Set_Initial_of_CTSU(**void**)
 説明 この関数は、correcton_CTSU_sensor_ico 関数よりコールされ、CTSU データ転送初期化設定を行います。

コール関数 -
 引数 -
 リターン値 -

3.5.8 correction_CTSU_register_set

correction_CTSU_register_set()

概要 ICO 補正 CTSU レジスタ設定関数
 宣言 **uint16_t** correction_CTSU_register_set(**void**)
 説明 この関数は、correcton_CTSU_sensor_ico 関数よりコールされ、CTSU レジスタに補正用設定を行います。それに伴い、以下の各レジスタ設定関数をコールします。

コール関数 R_Set_CTSU_TSCAP_Discharge()
 MPC_CTSU_set()
 R_Set_CTSU_Module_Operation()
 R_Set_CTSU_Transmit_Power_Supply()
 R_Set_CTSU_Power_Operation_Mode()
 R_Set_CTSU_Power_Capacity_Adjustment()
 R_Set_CTSU_Operation_Clock()
 R_Set_CTSU_Sensor_Stabilization_Wait_Time()
 R_Set_CTSU_Power_Supply()
 R_Set_CTSU_Measurement_Mode()
 R_Set_CTSU_Synchronous_Noise_Reduction()
 R_Set_CTSU_Measurement_Channel()
 R_Set_CTSU_Channel_Enable0()
 R_Set_CTSU_Channel_Transmit_Receive0()
 R_Set_CTSU_High_Pass_Noise_Reduction()

引数 - -

リターン値 0x0000 : 正常終了
 0x0001-0xFFFF : 異常終了

3.5.9 R_Set_CTSU_Correction_Mode

R_Set_CTSU_Correction_Mode()

概要 ICO 補正用計測モード設定関数
 宣言 **void** R_Set_CTSU_Correction_Mode(**void**)
 説明 この関数は、correcton_CTSU_sensor_ico 関数よりコールされ、以下の設定を行うことで補正用計測モード設定を行います。

CTSU エラーステータスレジスタ (CTSUERRS)		
シンボル名	設定値	設定内容
CTSUTSOC	1	キャリブレーション設定
CTSUSPMD	2	キャリブレーションモード

コール関数 -

引数 - -

リターン値 -

3.5.10 correction_CTSU_measurement_start

correction_CTSU_measurement_start()

概要 ICO 補正計測開始関数
 宣言 **void** correction_CTSU_measurement_start(**void**)
 説明 この関数は、correcton_CTSU_sensor_ico 関数よりコールされ、32 回の補正計測を行います。

32 回の計測値は、グローバル変数 g_correction_1st_val と g_correction_2nd_val に格納します。

コール関数 -
 引数 -
 リターン値 -

3.5.11 correction_CTSU_1st_coefficient_create

correction_CTSU_1st_coefficient_create()

概要 補正基準データに対する 1st 倍率を算出関数
 宣言 **void** correction_CTSU_1st_coefficient_create(**void**)
 説明 この関数は、correcton_CTSU_sensor_ico 関数よりコールされ、補正基準データに対する 1st (19.2uA) の倍率を算出します。

$$1st \text{ 倍率 (g_correction_1st_coefficient)} = \frac{1st \text{ 値 (g_correction_1st_val)}}{19.2\mu A \text{ 補正基準データ}}$$

演算された結果は、グローバル変数 g_correction_1st_coefficient に格納します。

コール関数 -
 引数 -
 リターン値 -

3.5.12 correction_CTSU_2nd_standard_val_create

correction_CTSU_2nd_standard_val_create()

概要	2nd の補正基準データを算出関数
宣言	void correction_CTSU_2nd_standard_val_create(void)
説明	この関数は、correcton_CTSU_sensor_ico 関数よりコールされ、2nd の補正基準データを算出します。

$$\text{2nd の補正基準データ} = \frac{(\text{1stCTSUTRMR 値} + 273)}{528} \times \text{19.2uA 補正基準データ}$$

演算された結果は、グローバル変数 g_correction_2nd_std_val に格納します。

コール関数	-
引数	-
リターン値	-

3.5.13 correction_CTSU_2nd_coefficient_create

correction_CTSU_2nd_coefficient_create()

概要	補正基準データに対する 2nd 倍率を算出関数
宣言	void correction_CTSU_2nd_coefficient_create(void)
説明	この関数は、correcton_CTSU_sensor_ico 関数よりコールされ、2nd の補正基準データに対する 2nd の倍率を算出します。

$$\text{2nd 倍率 (g_correction_2nd_coefficient)} = \frac{\text{2nd 値}}{\text{g_correction_2nd_std_val}}$$

演算された結果は、グローバル変数 g_correction_2nd_coefficient に格納します。

コール関数	-
引数	-
リターン値	-

3.5.14 correction_CTSU_16point_coefficient_create

correction_CTSU_16point_coefficient_create()

概要	1.2uA～19.2uA の補正係数算出関数
宣言	void correction_CTSU_16point_coefficient_create(void)
説明	この関数は、correcton_CTSU_sensor_ico 関数よりコールされ、1.2uA～19.2uA の補正係数を算出します。

演算された結果は、グローバル変数 g_correction_16coefficient[]に格納します。

コール関数	-
引数	-
リターン値	-

3.5.15 correction_sensor_magnification_set

correction_sensor_magnification_set()

概要	計測時間による補正係数の倍率設定関数
宣言	void correction_sensor_magnification_set(void)
説明	この関数は、R_Set_CTSU_Measurement_Start 関数よりコールされ、タッチ計測時間 0.5ms の補正係数を現タッチ計測時間に対する倍率を設定します。

現タッチ計測時間は、PCLKBの供給クロックと計測周波数、計測回数から判断し、倍率を設定します。

コール関数	-
引数	-
リターン値	-

3.5.16 correction_sensor_cnt_create

correction_sensor_cnt_create()

概要	計測値補正関数
宣言	uint16_t correction_sensor_cnt_create(uint8_t method, uint8_t ts_num, uint8_t number)
説明	この関数は、touch_data_moving_average 関数よりコールされ、計測された計測値に補正係数を掛け、補正計測値を設定します。

コール関数	-
引数	uint8_t method 0-9 : デフォルト計測方式番号
	uint8_t ts_num 計測TS番号
	uint8_t number 計測ポインタ番号
リターン値	補正計測値 ... 0-65535

4. フローチャート

4.1 Main 関数 : main()

図 4.1 にフローチャートを示します。

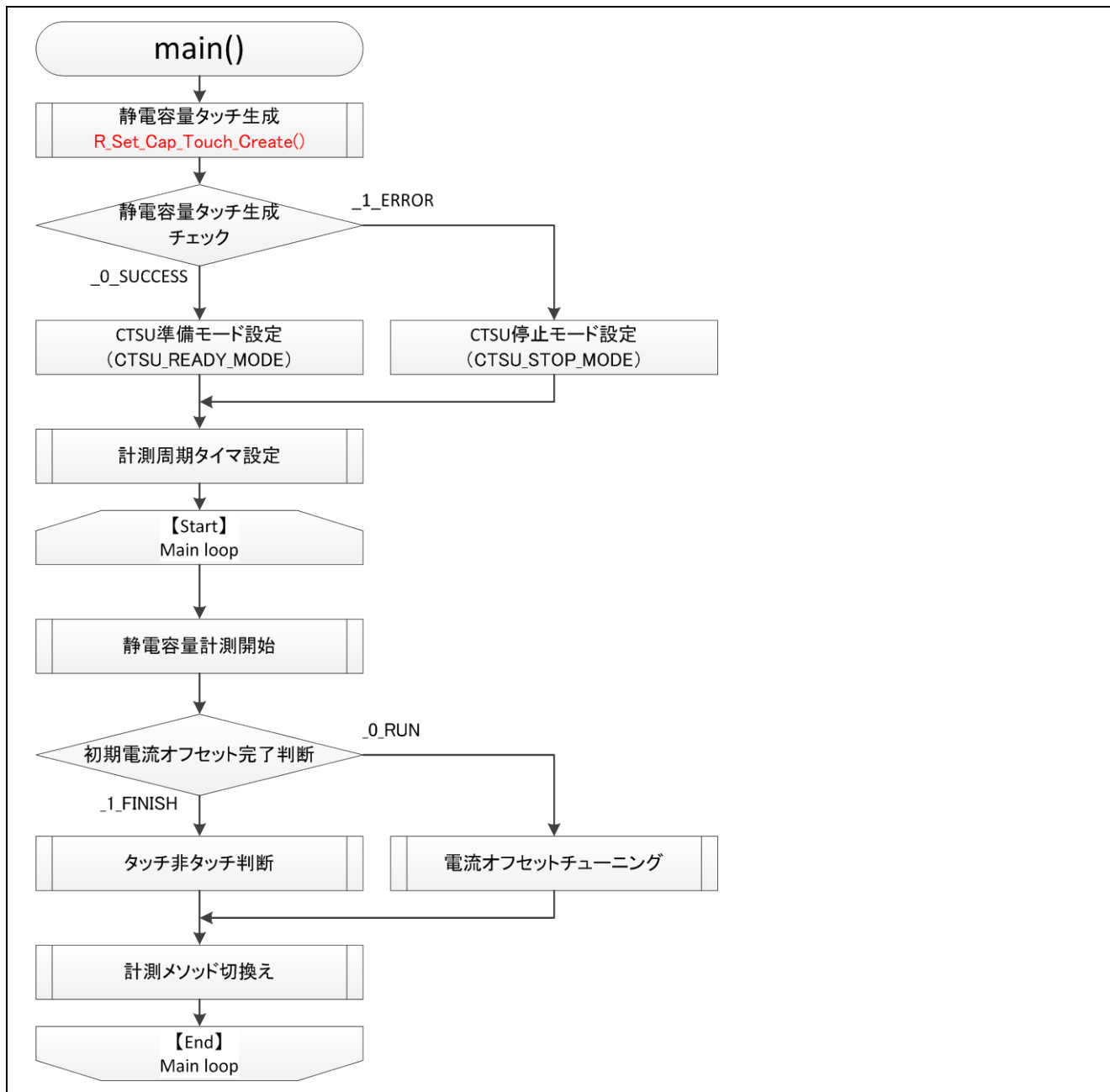


図 4.1 main()

4.2 静電容量タッチ生成関数 : R_Set_Cap_Touch_Create()

図 4.2 にフローチャートを示します。

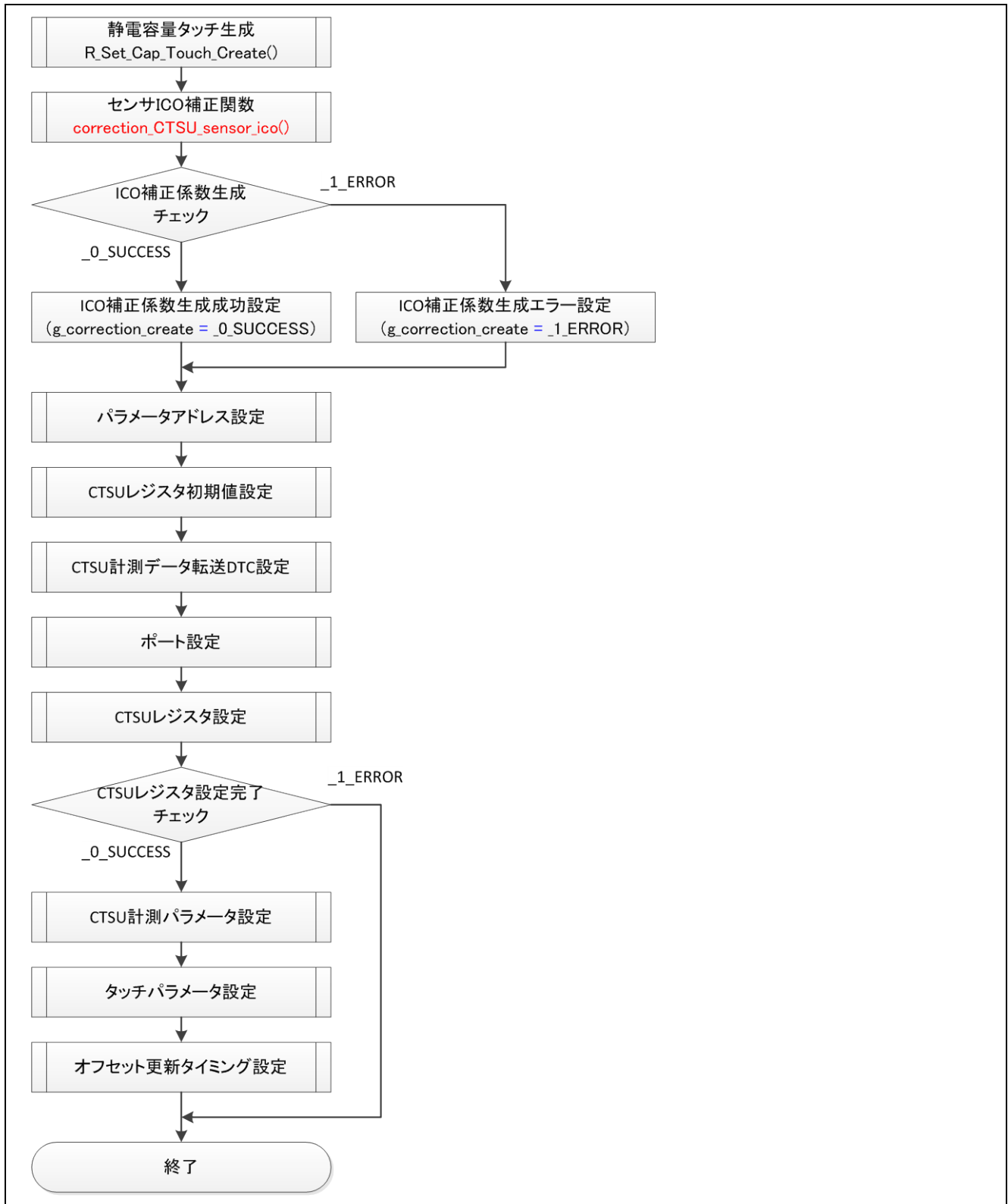


図 4.2 R_Set_Cap_Touch_Create()

4.3 センサ ICO 補正関数 : correcton_CTSU_sensor_ico()

図 4.3 にフローチャートを示します。

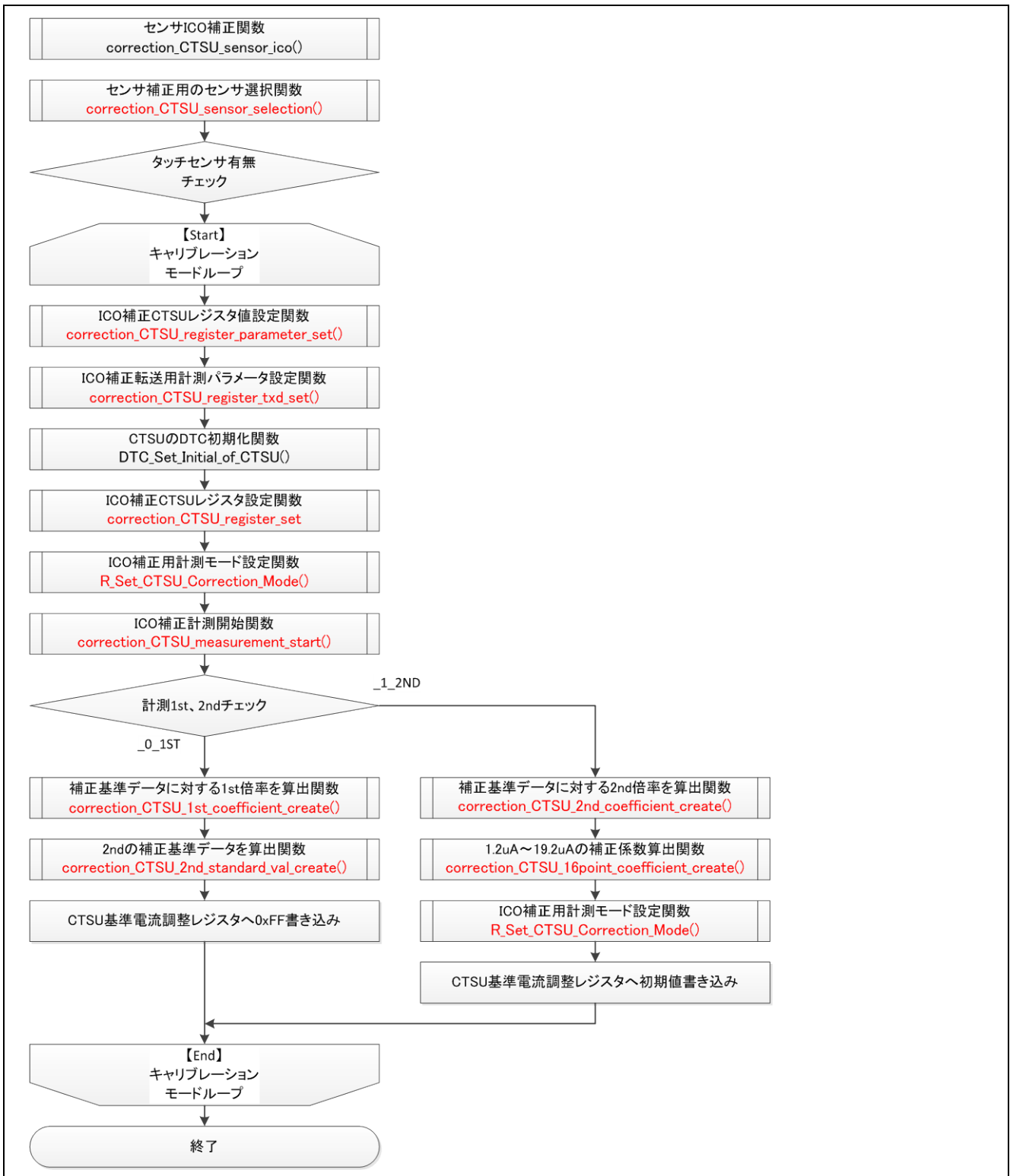


図 4.3 correcton_CTSU_sensor_ico()

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/>

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。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2017.1.18	-	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っていると、内部ROM、レイアウトパターンの相違などにより、電氣的特性の範囲で、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、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、全部または一部を問わず、改造、改変、複製、その他の不適切に使用しないでください。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、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、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損害が生じても、当社は一切その責任を負いません。
 6. 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。
 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。
 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。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. 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、(1)核兵器、化学兵器、生物兵器等の大量破壊兵器およびこれらを運搬することができるミサイル（無人航空機を含みます。）の開発、設計、製造、使用もしくは貯蔵等の目的、(2)通常兵器の開発、設計、製造または使用の目的、または(3)その他の国際的な平和および安全の維持の妨げとなる目的で、自ら使用せず、かつ、第三者に使用、販売、譲渡、輸出、賃貸もしくは使用許諾しないでください。
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。
 10. お客様の転売、貸与等により、本書（本ご注意書きを含みます。）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、当社は一切その責任を負わず、お客様にかかる使用に基づく当社への請求につき当社を免責いただきます。
 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。
 12. 本資料に記載された情報または当社製品に関し、ご不明点がある場合には、当社営業にお問い合わせください。
- 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と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。

(Rev.3.0-1 2016.11)



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 〒135-0061 東京都江東区豊洲3-2-24 (豊洲フォレシア)

■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：<https://www.renesas.com/contact/>