

RX21A グループ

R01AN1923JJ0100

Rev.1.00

2014.09.01

温度センサを用いた周囲温度の算出例

要旨

本アプリケーションノートでは、RX21A グループの温度センサを使用して、周囲温度を算出する方法について説明します。

対象デバイス

- ・ RX21A グループ 100 ピン版 ROM 容量 : 256KB~512KB
- ・ RX21A グループ 80 ピン版 ROM 容量 : 256KB~512KB
- ・ RX21A グループ 64 ピン版 ROM 容量 : 256KB~512KB

注. 対象デバイスは G バージョン(動作周囲温度 : -40°C ~ $+105^{\circ}\text{C}$)のみです。

目次

1. 仕様.....	3
2. 動作確認条件	5
3. 関連アプリケーションノート	5
4. ハードウェア説明	6
4.1 ハードウェア構成例	6
4.2 使用端子一覧	6
5. ソフトウェア説明	7
5.1 動作概要	7
5.1.1 温度特性の計算式	9
5.2 ファイル構成	11
5.3 オプション設定メモリ	12
5.4 定数一覧	12
5.5 変数一覧	14
5.6 関数一覧	15
5.7 関数仕様	16
5.8 フローチャート	20
5.8.1 メイン処理	20
5.8.2 ポート初期設定	21
5.8.3 周辺機能初期設定	21
5.8.4 CMT 初期設定	22
5.8.5 IRQ 初期設定	23
5.8.6 7SEG 表示データの更新処理	24
5.8.7 7SEG セレクト出力切り替え処理	25
5.8.8 7SEG バー表示処理	25
5.8.9 コンペアマッチ割り込み処理	26
5.8.10 AD、温度センサ初期設定	27
5.8.11 A/D 変換状態取得	28
5.8.12 温度センサ測定結果取得	28
5.8.13 現在温度取得	28
5.8.14 温度センサキャリブレーション処理	29
5.8.15 温度センサ測定処理	29
5.8.16 現在温度算出処理	30
5.8.17 A/D 変換完了割り込み処理	31
6. サンプルコード	32
7. 参考ドキュメント	32

1. 仕様

温度センサを使用して、MCU の周囲温度を測定します。周囲温度を測定し、7 セグメント LED(以下、7SEG) に表示します。

MCU の周囲温度を測定するために、測定前に温度センサのキャリブレーションを実行します。本アプリケーションノートにおけるキャリブレーションとは、温度特性の計算式に必要な温度傾斜を算出することです。

RX21A の G バージョンには、工場出荷時にチップごとに測定された温度センサ校正データが格納されています。このデータと、ユーザが試行測定した温度から、温度傾斜を算出することができます。

本サンプルコードでは、試行測定する温度として、周囲温度 25°C(以下、常温基準温度)の結果を計算に使用します。キャリブレーションの内容については 5.1.1 状態遷移と 7SEG の表示のパターンで説明します。

表 1.1 に使用する周辺機能と用途を、図 1.1 に状態遷移と 7SEG の表示を示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
10 ビット A/D コンバータ(以下、AD)	温度センサ出力測定用
温度センサ	MCU の周囲温度測定
コンペアマッチタイマ (CMT0。以下、CMT)	温度測定周期計測
外部端子割り込み(IRQ2。以下、IRQ)	常温基準温度(25°C)でキャリブレーションを実行するための SW 入力
I/O ポート	7SEG 表示(温度測定結果の表示用)

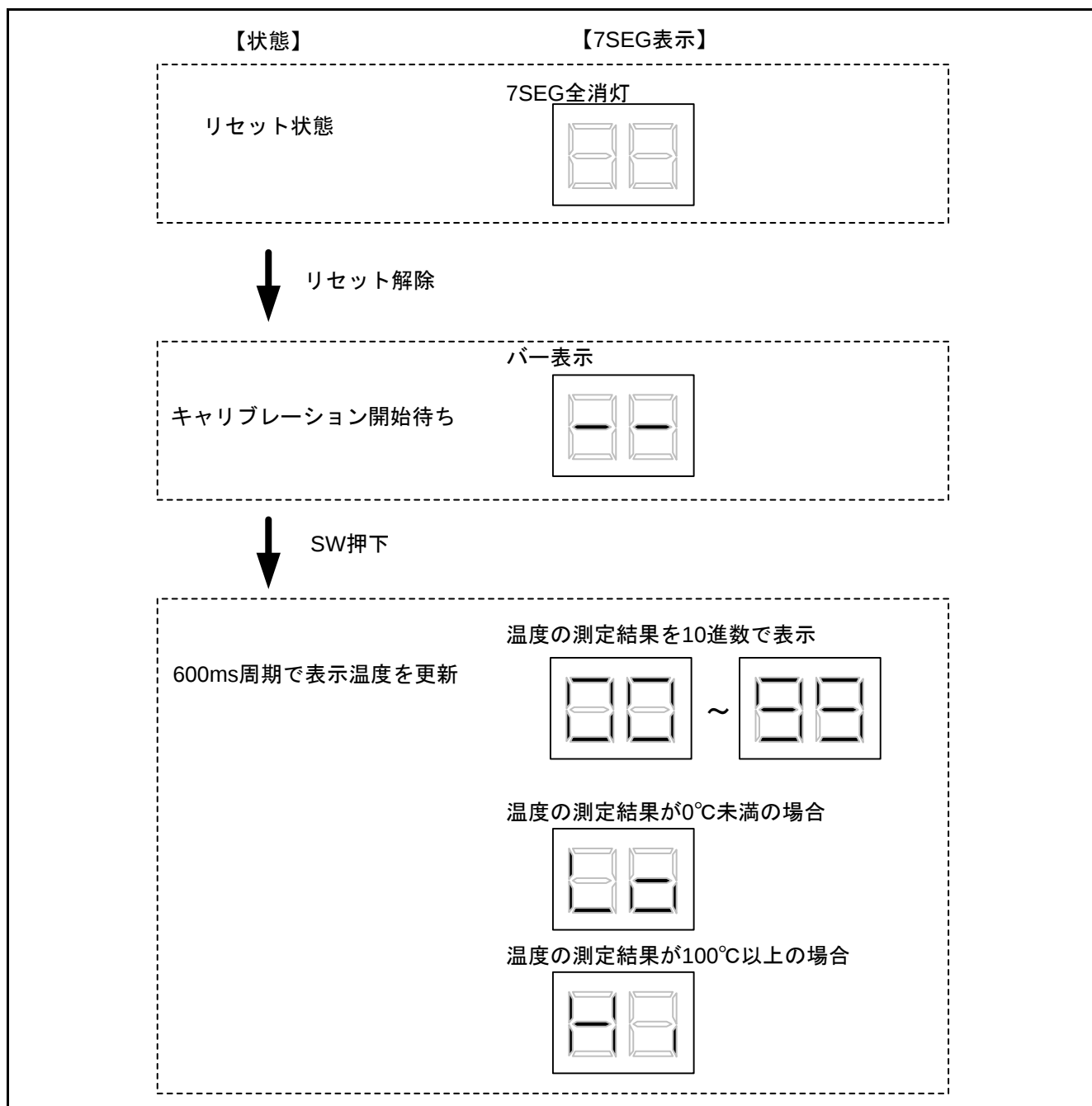


図 1.1 状態遷移と 7SEG の表示のパターン

2. 動作確認条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表 2.1 動作確認条件

項目	内容
使用マイコン	R5F521A8BDFP (RX21A グループ)
動作周波数	メインクロック: 20MHz システムクロック (ICLK): 20MHz 周辺モジュールクロック B (PCLKB): 20MHz 周辺モジュールクロック D (PCLKD): 2.5MHz
動作電圧	3.3V
統合開発環境	ルネサスエレクトロニクス製 High-performance Embedded Workshop Version 4.09.01
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler Package for RX Family V.1.02 Release 01 コンパイルオプション -cpu=rx200 -output=obj="\$(CONFIGDIR)¥\$(FILELEAF).obj" -debug -nologo (統合開発環境のデフォルト設定を使用しています)
iodefine.h のバージョン	Version 1.1A
エンディアン	リトルエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
サンプルコードのバージョン	Version 1.00

3. 関連アプリケーションノート

本アプリケーションノートに関連するアプリケーションノートを以下に示します。併せて参照してください。

- RX21A グループ 初期設定例 Rev.1.10 (R01AN1486JJ)
- RX ファミリ ソフトウェアによるウェイト処理のコーディング例 Rev.1.00 (R01AN1852JJ)

上記アプリケーションノートの初期設定関数とソフトウェアによるウェイト処理を、本アプリケーションノートのサンプルコードで使用しています。Rev は本アプリケーションノート作成時点のものです。

最新版がある場合、最新版に差し替えて使用してください。最新版はルネサスエレクトロニクスホームページで確認および入手してください。

4. ハードウェア説明

4.1 ハードウェア構成例

図 4.1 に接続例を示します。

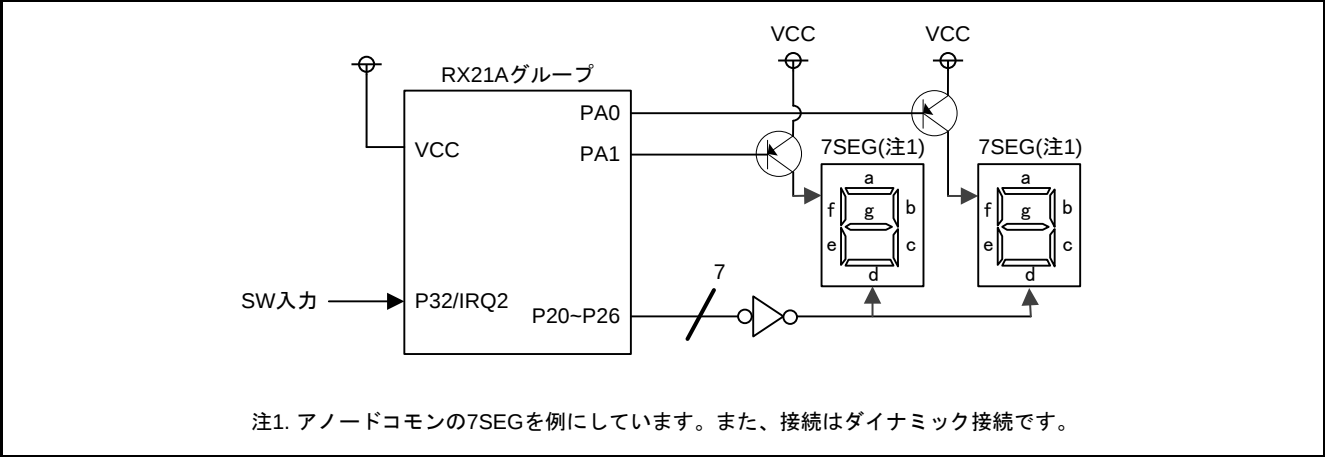


図 4.1 接続例

4.2 使用端子一覧

表 4.1 に使用端子と機能を示します。

使用端子は 100 ピン版の製品を想定しています。100 ピン版未満の製品を使用する場合は、使用する製品に合わせて端子を選択してください。

表 4.1 使用端子と機能

端子名	入出力	内容
P32/IRQ2	入力	キャリブレーション実行 SW 入力
P20	出力	7SEG セグメント a 出力
P21	出力	7SEG セグメント b 出力
P22	出力	7SEG セグメント c 出力
P23	出力	7SEG セグメント d 出力
P24	出力	7SEG セグメント e 出力
P25	出力	7SEG セグメント f 出力
P26	出力	7SEG セグメント g 出力
PA0	出力	7SEG 1 桁目セレクト出力
PA1	出力	7SEG 2 桁目セレクト出力

5. ソフトウェア説明

5.1 動作概要

リセット解除後、I/O ポートと周辺機能の初期設定を行ってから、キャリブレーション待ちの状態になります。この状態で IRQ2 割り込み要求が発生すると、キャリブレーションを実行します。キャリブレーションでは、常温基準温度の A/D 変換処理を行います。取得した A/D 変換値と温度センサ校正データの値から温度傾斜を算出します。

キャリブレーションが完了すると、引き続き A/D 変換が行われ、この A/D 変換値と温度傾斜から周囲温度を算出して 7SEG に表示します。

本アプリケーションノートでは 100ms ごとに 1 回 A/D 変換を行っています。また、A/D 変換値を平均化する手法として、6 回分の A/D 変換値を RAM に退避し、その合計値から最大値と最小値を引いた 4 回分の平均値を算出しています。

100ms ごとの A/D 変換開始動作には CMT の CMI0 割り込みを使用しています。CMT には 1ms 周期でコンペアマッチ割り込み要求が発生するように設定しており、コンペアマッチ割り込み要求が発生するごとに A/D 変換周期カウンタ変数(cnt_cycle)をカウントアップし、100ms を計測します。

図 5.1 に温度測定のタイミング図を示します。

CMT、AD、温度センサの設定を以下に示します。

<CMT0>

- ・ カウントクロック : PCLKB の 8 分周
- ・ コンペアマッチ割り込み周期 : 1ms

<AD>

- ・ 動作モード : シングルスキャンモード
- ・ A/D 変換開始条件 : 同期トリガ(温度センサからのトリガ)
- ・ サンプリングステート数 : 180 ステート(サンプリング時間:72 μ s)
- ・ アナログ入力断線検出アシスト機能 : 使用しない
- ・ A/D 変換値加算モード : 使用しない
- ・ 10 ビット A/D コンバータの自己診断機能 : 使用しない

<温度センサ>

- ・ PGA ゲイン(注 1) : $2.7V \leq AVCC0 \leq 3.6V$ (注 2)

注1. PGA : Programable Gain Amplifier

注2. 使用するシステムに応じて定数の設定を変更してください。

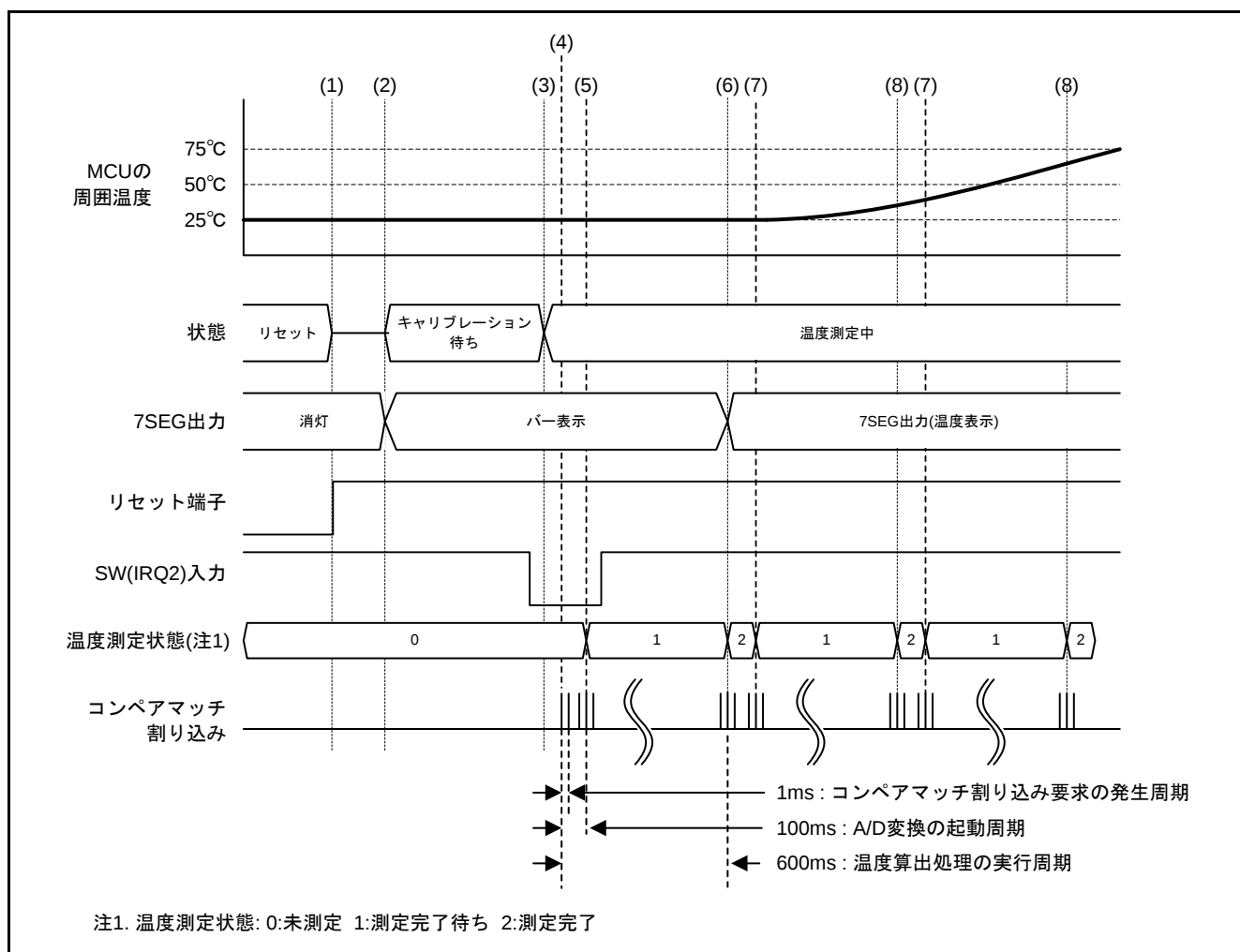


図 5.1 温度測定のタイミング図

- (1) リセット解除後、AD および温度センサの初期設定を行います。
- (2) AD のモジュールストップ状態を解除した後、1 μ s 待ってから(注 1)キャリブレーション待ち状態に移行します。このとき、7SEG をバー表示にします。
- (3) SW(IRQ2)に立ち下がりエッジを検出したときに、CMT のカウントを開始します。
- (4) コンペアマッチ割り込み要求が 1ms 周期で発生するごとに、A/D 変換周期カウンタ変数(cnt_cycle)をカウントアップします。
- (5) A/D 変換周期カウンタ変数が 100(100ms)になったとき、PGAEN ビットに“1”を設定し、A/D 変換を開始します。
- (6) A/D 変換を 6 回行い、それらの平均値を常温基準温度の A/D 変換値とし、温度傾斜を算出してキャリブレーションを終了します。
- (7) 引き続き、A/D 変換周期カウンタ変数が 100(100ms)になったとき、PGAEN ビットに“1”を設定し、A/D 変換を開始します。
- (8) 6 回 A/D 変換を行った後、平均値と温度傾斜から現在温度を算出し、7SEG に表示します。

注1. AD のモジュールストップ状態を解除した後は、1 μ s 待ってから A/D 変換を開始してください。

5.1.1 温度特性の計算式

本アプリケーションノートでは、温度センサ校正データレジスタ (TSCDRn (n=0,1,3))に格納されている周囲温度 125°C(以下、高温基準温度)の温度センサ出力の A/D 変換値と、リセット解除後に測定した常温基準温度の A/D 変換値の 2 点を用いて、温度特性の計算式に必要な温度傾斜を算出しています。

TSCDRn (n=0,1,3) レジスタの詳細は、RX21A グループ ユーザーズマニュアル ハードウェア編(以下、UMH)を参照してください。

表 5.1 に TSCDRn (n=0,1,3)レジスタに格納されている温度センサ出力の A/D 変換値の測定条件を示します。

表 5.1 TSCDRn (n=0,1,3)レジスタに格納されている温度センサ出力の A/D 変換値の測定条件

レジスタ名	測定条件		
	AVCC0 と VREFH0 への 印加電圧(V)	TSCR レジスタ PGAGAIN[1:0]ビット	測定温度(°C)
TSCDR0	1.8	00b	125
TSCDR1	2.7	01b	
TSCDR3	3.3	01b	

AVCC0 と VREFH0 に、表 5.1 に記載している条件以外の電圧を印加する場合は、印加電圧に応じた A/D 変換値を算出する必要があります。この算出する A/D 変換値を CAL_{125} と定義します。

AVCC0 が $1.8V \leq AVCC0 < 2.7V$ の場合は①の式で、 $2.7V \leq AVCC0 \leq 3.6V$ の場合は②の式で CAL_{125} を算出します。

$$\textcircled{1} \quad CAL_{125} = 1.8 / VREFH0 \times TSCDR(0)$$

$$\textcircled{2} \quad CAL_{125} = 2.7 / VREFH0 \times TSCDR(1) + \{ 3.3 / VREFH0 \times TSCDR(3) - 2.7 / VREFH0 \times TSCDR(1) \} \\ \times (AVCC0 - 2.7) / 0.6$$

TSCDR(n) : TSCDRn.TSCD[9:0]ビットの値 (n=0,1,3)

周囲温度を算出するには、まず温度傾斜を算出します。ここでは、温度傾斜を温度に対する A/D 変換値の増分とします。なお、UMH では A/D 変換値を電圧に変換してから温度傾斜や温度を算出する説明となっていますが、本アプリケーションノートでは A/D 変換値のままで温度傾斜や温度を算出しています。

計算式は以下のとおりです。

温度傾斜 : Slope

高温基準温度(125°C) : T1

常温基準温度(25°C) : T2

高温基準温度(125°C)の A/D 変換値 : CAL_{125}

常温基準温度(25°C)の A/D 変換値 : CAL_{25} (リセット後の常温基準温度で測定した値)

$$\text{温度傾斜 Slope} = (CAL_{125} - CAL_{25}) / (T1 - T2)$$

T1=125(°C)、T2=25(°C)のため

$$\text{Slope} = (CAL_{125} - CAL_{25}) / (125 - 25) = (CAL_{125} - CAL_{25}) / 100$$

となります。

周囲温度は、以下の計算式で算出します。

測定温度：T(°C)

温度測定時の温度センサの A/D 変換値：CAL_S

$$\begin{aligned} T &= T_2 + (CAL_S - CAL_{25}) / \text{Slope} \\ &= T_2 + (CAL_S - CAL_{25}) / ((CAL_{125} - CAL_{25}) / (T_1 - T_2)) \\ &= T_2 + (T_1 - T_2) ((CAL_S - CAL_{25}) / (CAL_{125} - CAL_{25})) \\ &= 25 + 100((CAL_S - CAL_{25}) / (CAL_{125} - CAL_{25})) \end{aligned}$$

測定温度を小数点第一位まで測定したい場合は、温度データ(T1,T2)を 10 倍にして算出します。

測定温度：Ts (°C)

$$\begin{aligned} Ts &= T \times 10 = (25 + 100((CAL_S - CAL_{25}) / (CAL_{125} - CAL_{25}))) \times 10 \\ &= (25 \times 10) + (100((CAL_S - CAL_{25}) / (CAL_{125} - CAL_{25})) \times 10) \\ &= 250 + 1000((CAL_S - CAL_{25}) / (CAL_{125} - CAL_{25})) \end{aligned}$$

なお、基本的な内容は UMH に記載していますので参照してください。

5.2 ファイル構成

表 5.2 にサンプルコードで使用するファイル、表 5.3 に標準インクルードファイルを、表 5.4 に参照する関連アプリケーションノートの関数と設定値を示します。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 5.2 サンプルコードで使用するファイル

ファイル名	内容
main.c	メイン処理
temps.c	温度センサ処理
temps.h	temps.c のヘッダファイル

表 5.3 標準インクルードファイル

ファイル名	内容
stdbool.h	論理型、および論理値に関するマクロを定義します。
stdint.h	指定した幅の整数型を宣言してマクロを定義します。
machine.h	RX ファミリ用 組み込み関数の形式を定義します。

表 5.4 参照する関連アプリケーションノートの関数と設定値

(RX21A グループ 初期設定例、RX ファミリ ソフトウェアによるウェイト処理のコーディング例)

ファイル名	関数	設定値
r_init_stop_module.c	R_INIT_StopModule()	-
r_init_stop_module.h	-	-
r_init_non_existent_port.c	R_INIT_NonExistentPort()	-
r_init_non_existent_port.h	-	100 ピン版を指定
r_init_clock.c	R_INIT_Clock()	-
r_init_clock.h	-	クロックの選択例 : No.5 を指定。 PCLKD の分周比を 8 分周に変更。
r_delay.c	R_DELAY_Us(unsigned long us, unsigned long khz)	ウェイト時間を設定
r_delay.h	-	-

5.3 オプション設定メモリ

表 5.5 にサンプルコードで使用する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 5.5 サンプルコードで使用するオプション設定メモリ

シンボル	アドレス	設定値	内容
OFS0	FFFF FF8Fh~FFFF FF8Ch	FFFF FFFFh	リセット後、IWDG は停止 リセット後、WDT は停止
OFS1	FFFF FF8Bh~FFFF FF88h	FFFF FFFFh	リセット後、電圧監視 0 リセット無効 リセット後、HOCO 発振が無効
MDES	FFFF FF83h~FFFF FF80h	FFFF FFFFh	リトルエンディアン

5.4 定数一覧

表 5.6~表 5.9 にサンプルコードで使用する定数を示します。

表 5.6 サンプルコードで使用する定数(main.c)

定数名	設定値	内容
CMT_CYCLE_MS	100	A/D 変換周期(ms)
SEG_CYCLE_MS	8	7SEG セレクト出力切り替え周期(ms)
ONES_DIGIT	0	7SEG 出力フラグの値
SEG_TBL_DASH	10	7SEG 表示テーブルインデックス: “-”
SEG_TBL_H	11	7SEG 表示テーブルインデックス: “H”
SEG_TBL_i	12	7SEG 表示テーブルインデックス: “i”
SEG_TBL_L	13	7SEG 表示テーブルインデックス: “L”
SEG_TBL_o	14	7SEG 表示テーブルインデックス: “o”
SEG_TBL_BLANK	15	7SEG 表示テーブルインデックス: 空白

表 5.7 サンプルコードで使用する定数(temps.c)

定数名	設定値	内容
HIGH_REF_TEMP	125	高温基準温度(°C)
ADCONV_IN_OPERATION	0xFFFF	A/D 変換実行中の A/D 変換値(無効値)
SLOPE_COEFFICIENT_TEMP	(HIGH_REF_TEMP – ORDINARY_REF_TEMP) * TEMP_ACCURACY	温度傾斜
ORDINARY_REF_TEMP_IN_ACC	ORDINARY_REF_TEMP * TEMP_ACCURACY	常温基準温度(25°C)に温度計算精度を掛けた値

表 5.8 サンプルコードで使用する定数(temps.h)(ユーザ変更可)

定数名	設定値	内容
SEL_PGAGAIN	GAIN_RANGE1	PGA ゲイン選択 (注 1) GAIN_RANGE0 : $1.8V \leq AVCC0 < 2.7V$ GAIN_RANGE1 : $2.7V \leq AVCC0 \leq 3.6V$
AVCC_VOLTAGE	3.3	AVCC0 端子への印加電圧 (単位は[V]で指定。) (注 1)
VREF_VOLTAGE	3.3	VREFH0 端子への印加電圧 (単位は[V]で指定。)
ORDINARY_REF_TEMP	25	常温基準温度($^{\circ}C$) (設定値が 25 であれば、 $25^{\circ}C$ を常温基準温度とする。)
TEMP_ACCURACY	10	温度計算精度(倍率を設定。設定値を“10”と設定したときは、小数点 1 位まで。設定値を“100”と設定したときは小数点 2 位までを計算対象とする。10 の乗数値以外や負数を設定しないでください。)
CNV_CNT_MAX	6	平均値サンプリング回数 (設定値が 6 であれば、6 回分の A/D 変換値が溜まったところで、その合計値から最大値と最小値を引いた 4 回分の平均値を A/D 変換値とする。)

注1. AVCC0 端子への印加電圧と PGA ゲインは、条件が一致するように設定してください。条件が一致しない場合、正しい計算結果が得られなくなります。

表 5.9 サンプルコードで使用する定数(temps.h)(ユーザ変更不可)

定数名	設定値	内容
GAIN_RANGE0	00h	PGA ゲイン : $1.8V \leq AVCC0 < 2.7V$
GAIN_RANGE1	01h	PGA ゲイン : $2.7V \leq AVCC0 \leq 3.6V$
STA_AD_IDLE	0	A/D 変換状態 : 未動作
STA_AD_WAIT	1	A/D 変換状態 : A/D 変換完了待ち
STA_AD_FINISH	2	A/D 変換状態 : A/D 変換完了
TSCDR0_VALUE	(TEMPSCONST.TSCDR0.BIT.TSCD)	TSCDR0 レジスタ値
TSCDR1_VALUE	(TEMPSCONST.TSCDR1.BIT.TSCD)	TSCDR1 レジスタ値
TSCDR3_VALUE	(TEMPSCONST.TSCDR3.BIT.TSCD)	TSCDR3 レジスタ値
HIGH_REF_POTENTIAL_VAL	(注 1)	高温基準温度($125^{\circ}C$)の A/D 変換値

注1. 選択した PGA ゲインにより、設定値が異なります。以下に、選択した PGA ゲイン毎の設定値を示します。

GAIN_RANGE0 を選択 :

$(uint16_t) (1.8 / VREF_VOLTAGE * TSCDR0_VALUE)$

GAIN_RANGE1 を選択 :

$(uint16_t) ((2.7 / VREF_VOLTAGE * TSCDR1_VALUE) + ((3.3 / VREF_VOLTAGE * TSCDR3_VALUE) - (2.7 / VREF_VOLTAGE * TSCDR1_VALUE))) * (AVCC_VOLTAGE - 2.7) / 0.6)$

5.5 変数一覧

表 5.10、表 5.11 に static 型変数を、表 5.12 に const 型変数を示します。

表 5.10 static 型変数(main.c)

型	変数名	内容	使用関数
static volatile uint16_t	cnt_cycle	A/D 変換周期カウンタ	Excep_CMT0_CMIO
static volatile uint16_t	cnt_led_cycle	7SEG セレクト出力切り替え周期カウンタ	Excep_CMT0_CMIO
static uint8_t	digit_10	7SEG 2 桁目の表示データ	disp_7seg disp_comswitch_7seg disp_bar_7seg
static uint8_t	digit_1	7SEG 1 桁目の表示データ	disp_7seg disp_comswitch_7seg disp_bar_7seg

表 5.11 static 型変数(temps.c)

型	変数名	内容	使用関数
static int16_t	high_ref_potential	高温基準温度(125°C)の A/D 変換値 (= CAL ₁₂₅)	temps_init temps_calibration
static volatile int16_t	slope_potential	A/D 変換値傾斜	temps_calibration temps_calc
static volatile int16_t	ordinary_potential	常温基準温度(25°C)の A/D 変換値 (= CAL ₂₅)	temps_calibration temps_calc
static volatile uint8_t	ad_status	A/D 変換ステータス	main temps_get_ad_status temps_calibration temps_measurement Excep_AD_ADI
static volatile int16_t	now_temp	算出した現在温度	temps_get_now_temp Excep_AD_ADI
static volatile uint16_t	now_potential	現在の A/D 変換値	temps_calibration Excep_AD_ADI
static volatile uint16_t	buf_ad_value[CN T_CNT_MAX]	A/D 変換値バッファ	Excep_AD_ADI
static volatile uint16_t	ad_max_value	A/D 変換値最大値	Excep_AD_ADI
static volatile uint16_t	ad_min_value	A/D 変換値最小値	Excep_AD_ADI
static volatile uint8_t	ad_smp_cnt	A/D 変換値バッファのライトポインタ	Excep_AD_ADI

表 5.12 const 型変数(main.c)

型	変数名	内容	使用関数
static const uint8_t	seg_pattern_table	7SEG 表示テーブル	disp_comswitch_7seg

5.6 関数一覧

表 5.13 に関数を示します。

表 5.13 関数

関数名	概要	記載ファイル
main	メイン処理	main.c
port_init	ポート初期設定	main.c
peripheral_init	周辺機能初期設定	main.c
cmt_init	CMT 初期設定	main.c
irq_init	IRQ 初期設定	main.c
disp_7seg	7SEG 表示データの更新処理	main.c
disp_comswitch_7seg	7SEG セレクト出力切り替え処理	main.c
disp_bar_7seg	7SEG バー表示処理	main.c
Excep_CMT0_CMI0	コンペアマッチ割り込み処理	main.c
temps_init	AD、温度センサ初期設定	temps.c
temps_get_ad_status	A/D 変換状態取得	temps.c
temps_get_potential	温度センサ測定結果取得	temps.c
temps_get_now_temp	現在温度取得	temps.c
temps_calibration	温度センサキャリブレーション処理	temps.c
temps_measurement	温度センサ測定処理	temps.c
temps_calc	現在温度算出処理	temps.c
Excep_AD_ADI	A/D 変換完了割り込み処理	temps.c

5.7 関数仕様

サンプルコードの関数仕様を示します。

main	
概 要	メイン処理
ヘッダ	なし
宣 言	void main(void)
説 明	初期設定後、100ms ごとに温度センサ出力を A/D 変換し、算出した温度を 7SEG に表示します。
引 数	なし
リターン値	なし

port_init	
概 要	ポート初期設定
ヘッダ	なし
宣 言	static void port_init(void)
説 明	ポートの初期設定を行います。
引 数	なし
リターン値	なし

peripheral_init	
概 要	周辺機能初期設定
ヘッダ	なし
宣 言	static void peripheral_init(void)
説 明	使用する周辺機能の初期設定を行います。
引 数	なし
リターン値	なし

cmt_init	
概 要	CMT 初期設定
ヘッダ	なし
宣 言	static void cmt_init(void)
説 明	CMT0 の初期設定を行います。
引 数	なし
リターン値	なし

irq_init	
概 要	IRQ 初期設定
ヘッダ	なし
宣 言	static void irq_init(void)
説 明	IRQ2 の初期設定を行います。
引 数	なし
リターン値	なし

disp_7seg

概 要	7SEG 表示データの更新処理
ヘッダ	なし
宣 言	static void disp_7seg(int16_t disp_data)
説 明	引数で指定された値を 7SEG に表示するデータとして設定します。
引 数	int16_t disp_data : 7SEG 表示
	0 未満(マイナス値) : “Lo” 表示
	100 以上 : “Hi” 表示
	上記以外 : 温度表示
リターン値	なし

disp_comswitch_7seg

概 要	7SEG セレクト出力切り替え処理
ヘッダ	なし
宣 言	static void disp_comswitch_7seg(void)
説 明	出力する 7SEG のセレクト信号を切り替えます。
引 数	なし
リターン値	なし

disp_bar_7seg

概 要	7SEG バー表示処理
ヘッダ	なし
宣 言	static void disp_bar_7seg(void)
説 明	バー(ダッシュ)を 7SEG に表示します。
引 数	なし
リターン値	なし

Excep_CMT0_CMI0

概 要	コンペアマッチ割り込み処理
ヘッダ	なし
宣 言	static void Excep_CMT0_CMI0(void)
説 明	1ms 周期の割り込み処理を行います。割り込み要求が発生するたびにカウンタをアップします。100 回(100ms)経過すると温度の測定を開始します。また、8 回(8ms)経過すると出力する 7SEG のセレクト信号を切り替えます。
引 数	なし
リターン値	なし

temps_init	
概 要	AD、温度センサ初期設定
ヘッダ	temps.h
宣 言	void temps_init(void)
説 明	AD と温度センサの初期設定を行います。
引 数	なし
リターン値	なし

temps_get_ad_status	
概 要	A/D 変換状態取得
ヘッダ	temps.h
宣 言	uint8_t temps_get_ad_status(void)
説 明	現在の A/D 変換状態を取得します。
引 数	なし
リターン値	uint8_t
	: A/D 変換状態
	STA_AD_IDLE : 未動作
	STA_AD_WAIT : A/D 変換完了待ち
	STA_AD_FINISH : A/D 変換完了

temps_get_potential	
概 要	温度センサ測定結果取得
ヘッダ	なし
宣 言	static uint16_t temps_get_potential (void)
説 明	測定した A/D 変換値を取得します。
引 数	なし
リターン値	uint16_t
	: 温度センサの A/D 変換値
	:ADCONV_IN_OPERATION :A/D 変換動作中
	:ADCONV_IN_OPERATION 以外 :A/D 変換値

temps_get_now_temp	
概 要	現在温度取得
ヘッダ	temps.h
宣 言	int16_t temps_get_now_temp (void)
説 明	現在の温度を取得します。
引 数	なし
リターン値	int16_t
	: 現在温度

temps_calibration	
概 要	温度センサキャリブレーション処理
ヘッダ	temps.h
宣 言	void temps_calibration(void)
説 明	常温基準温度の A/D 変換値を取得し、RAM に退避します。
引 数	なし
リターン値	なし
temps_measurement	
概 要	温度センサ測定処理
ヘッダ	temps.h
宣 言	void temps_measurement(void)
説 明	現在の温度の測定を開始します。
引 数	なし
リターン値	なし
temps_calc	
概 要	現在温度算出処理
ヘッダ	なし
宣 言	static int16_t temps_calc(uint16_t w_now_potential)
説 明	引数の A/D 変換値から温度を算出します。
引 数	uint16_t w_now_potential : A/D 変換値
リターン値	int16_t : 現在温度(°C)
Excep_AD_ADI	
概 要	A/D 変換完了割り込み処理
ヘッダ	なし
宣 言	static void Excep_AD_ADI(void)
説 明	A/D 変換完了時、A/D 変換値を RAM に退避します。6 回目の A/D 変換完了時には、その 6 回分の合計値から最大値と最小値を引いた 4 回分の平均を算出し、その平均値から温度を算出します。
引 数	なし
リターン値	なし

5.8 フローチャート

5.8.1 メイン処理

図 5.2 にメイン処理のフローチャートを示します。

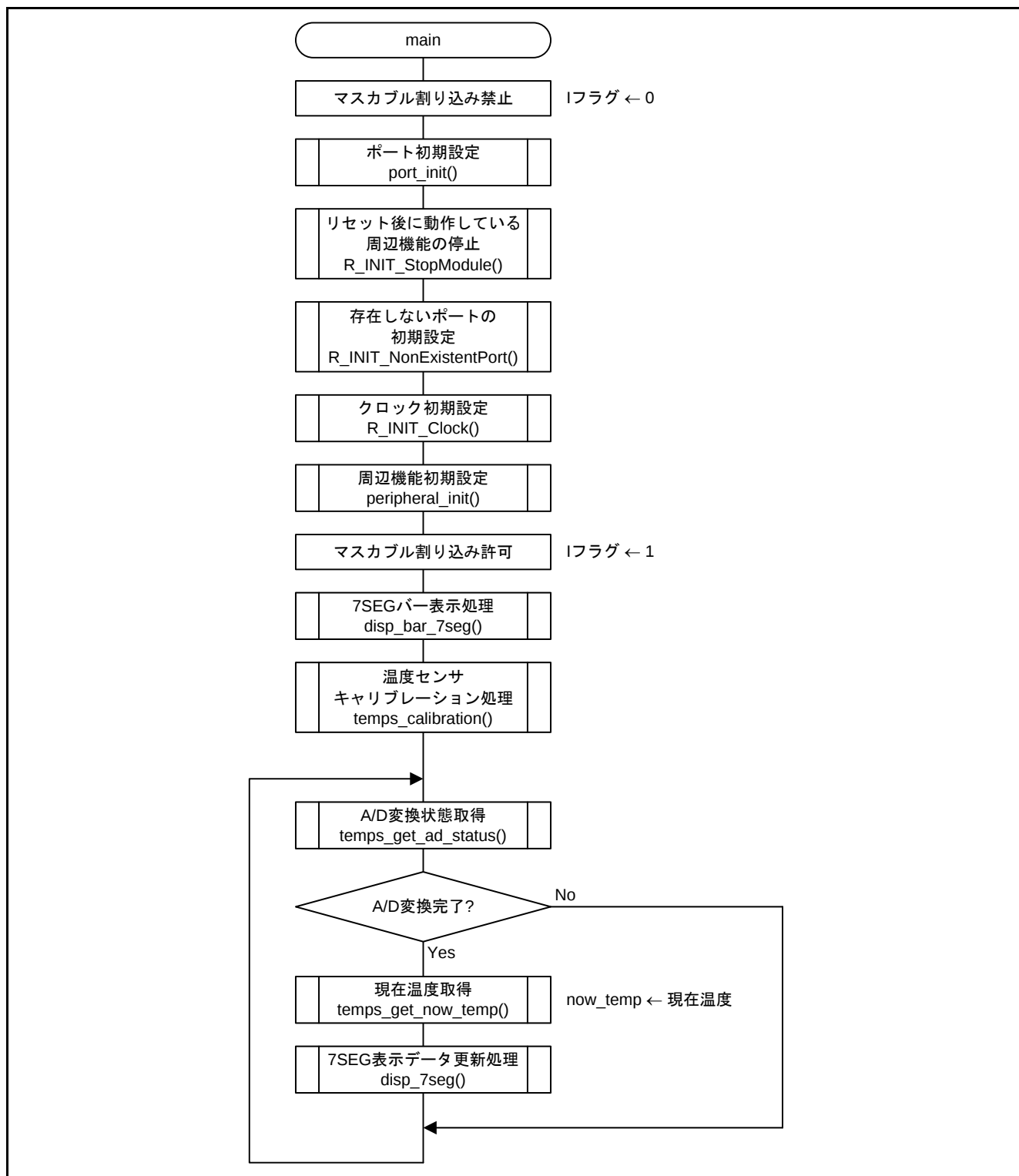


図 5.2 メイン処理

5.8.2 ポート初期設定

図 5.3 にポート初期設定のフローチャートを示します。

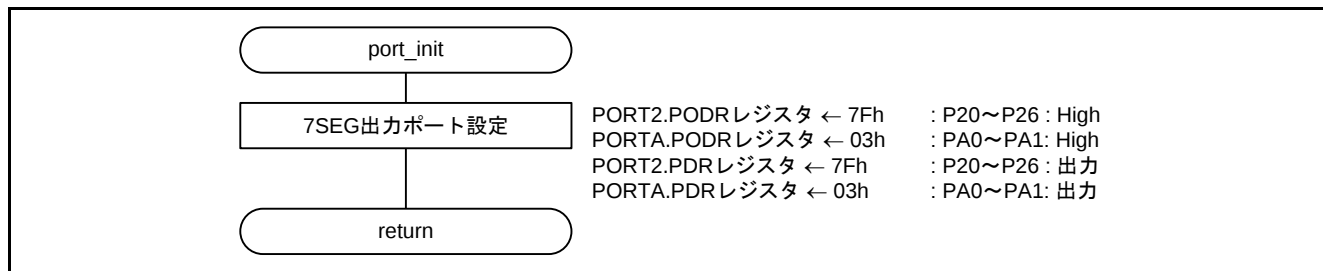


図 5.3 ポート初期設定

5.8.3 周辺機能初期設定

図 5.4 に周辺機能初期設定のフローチャートを示します。

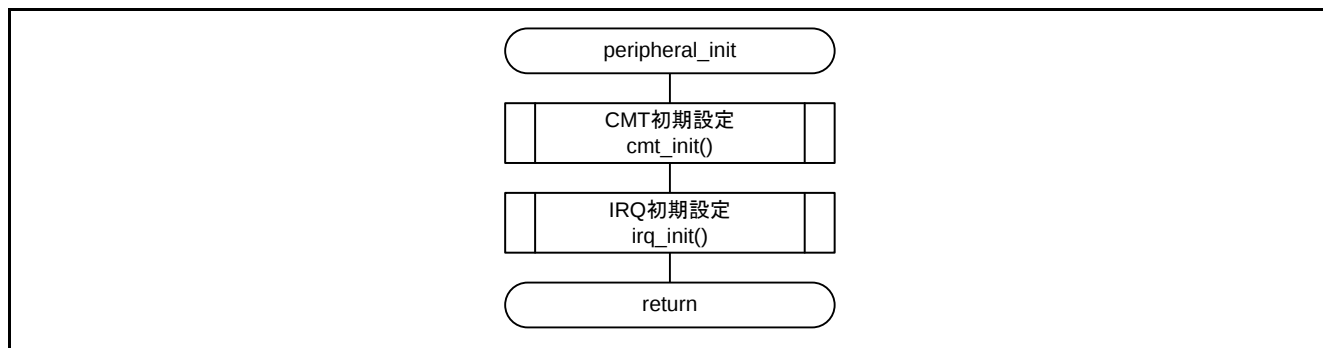


図 5.4 周辺機能初期設定

5.8.4 CMT 初期設定

図 5.5 に CMT 初期設定のフローチャートを示します。

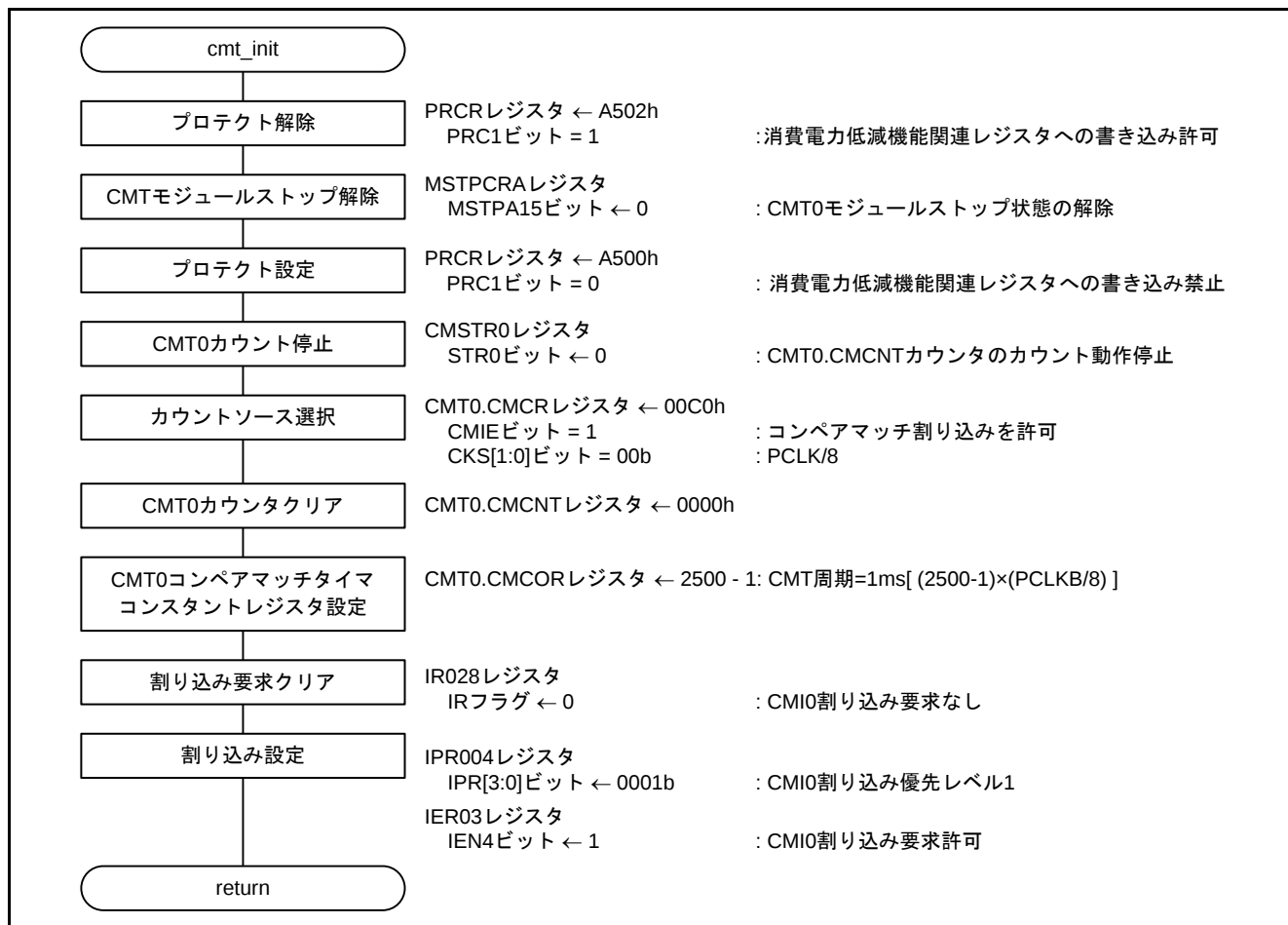


図 5.5 CMT 初期設定

5.8.5 IRQ 初期設定

図 5.6 に IRQ 初期設定のフローチャートを示します。

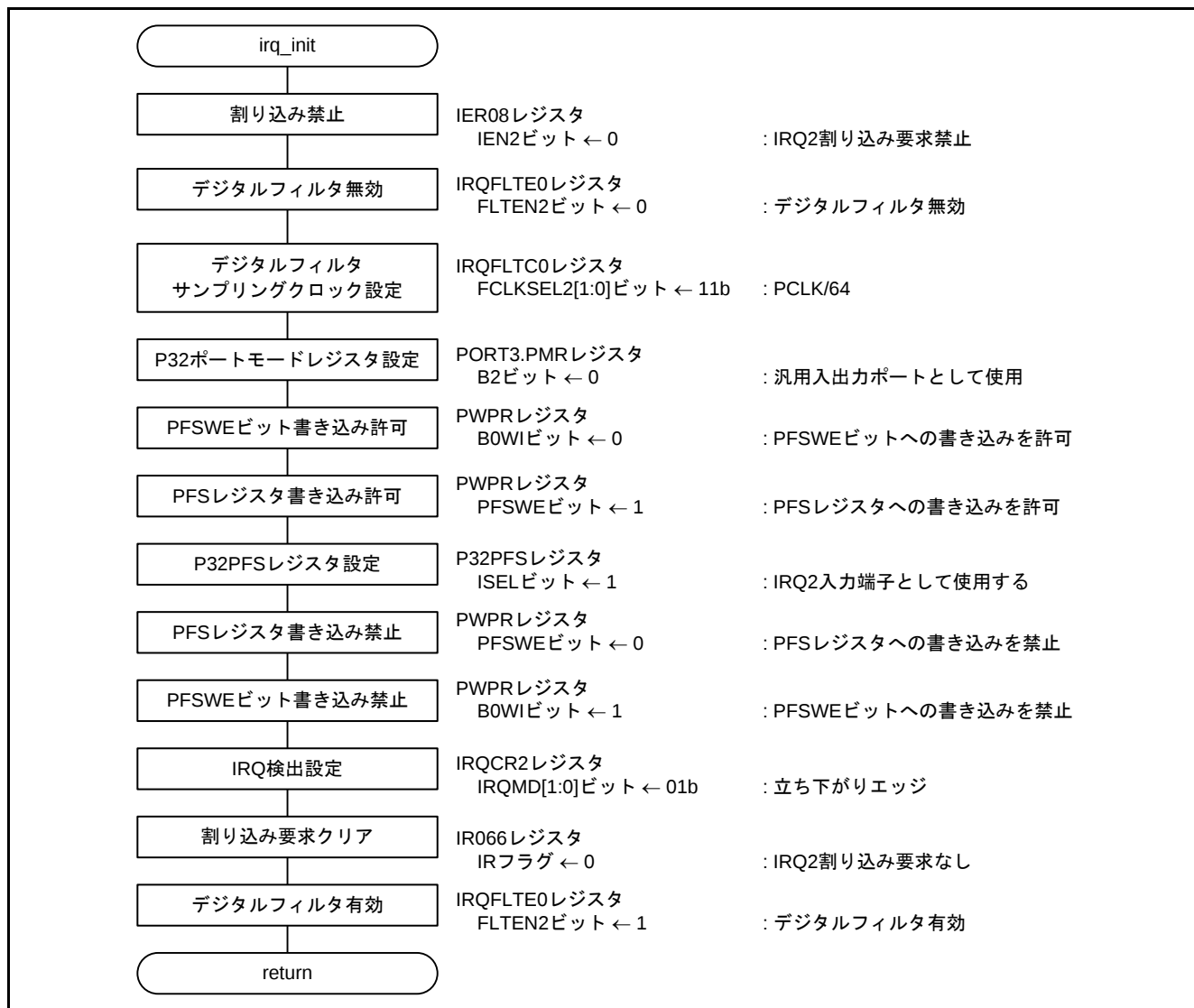


図 5.6 IRQ 初期設定

5.8.6 7SEG 表示データの更新処理

図 5.7 に 7SEG 表示データの更新処理のフローチャートを示します。

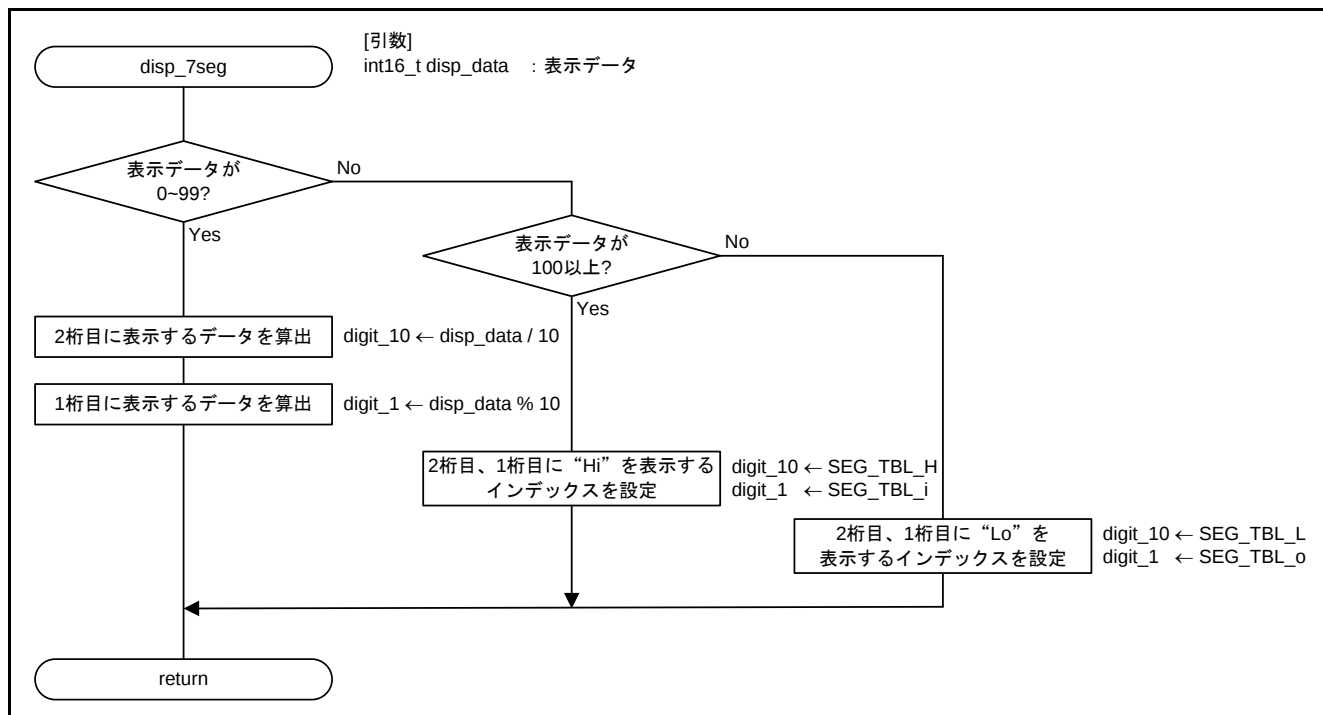


図 5.7 7SEG 表示データの更新処理

5.8.7 7SEG セレクト出力切り替え処理

図 5.8 に 7SEG セレクト出力切り替え処理のフローチャートを示します。

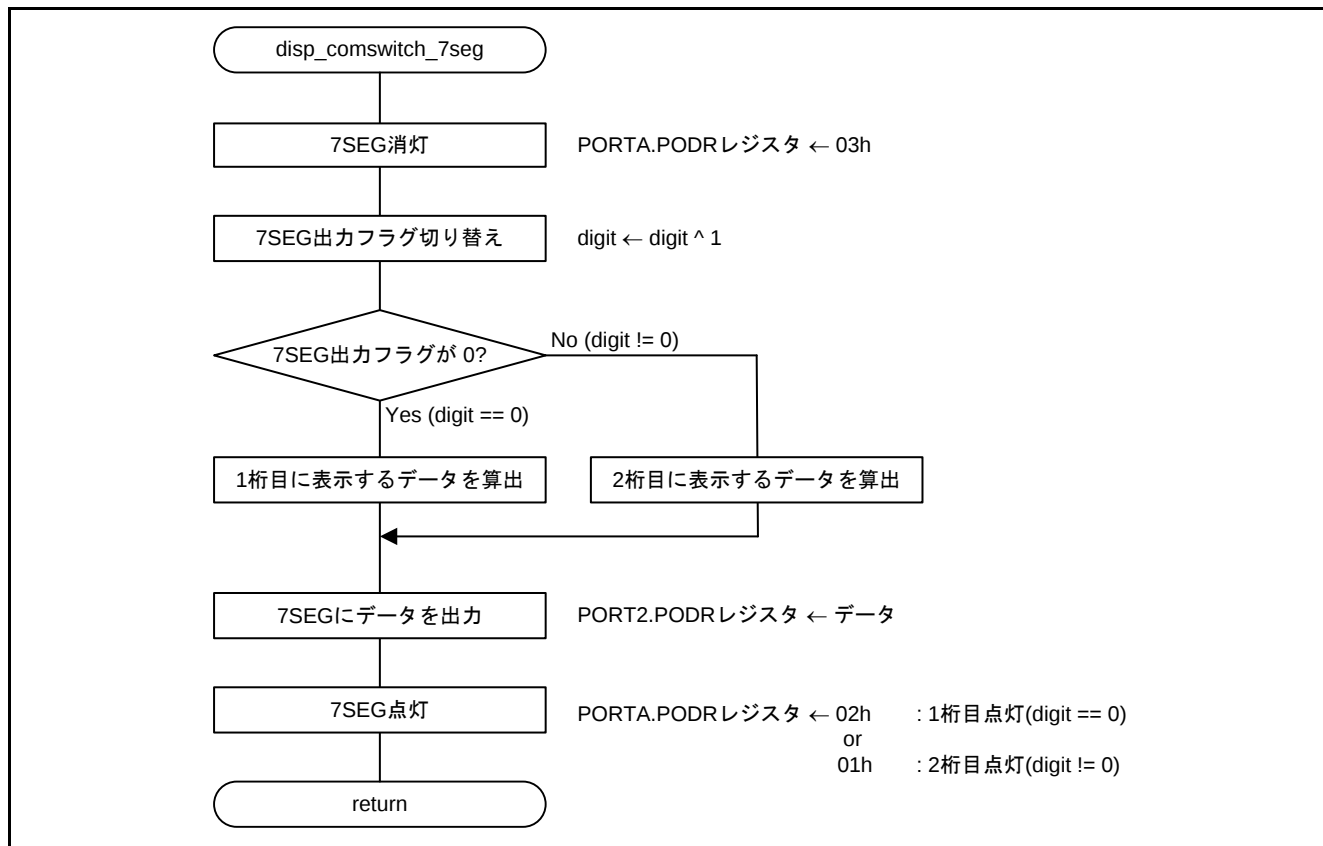


図 5.8 7SEG セレクト出力切り替え処理

5.8.8 7SEG バー表示処理

図 5.9 に 7SEG バー表示処理のフローチャートを示します。

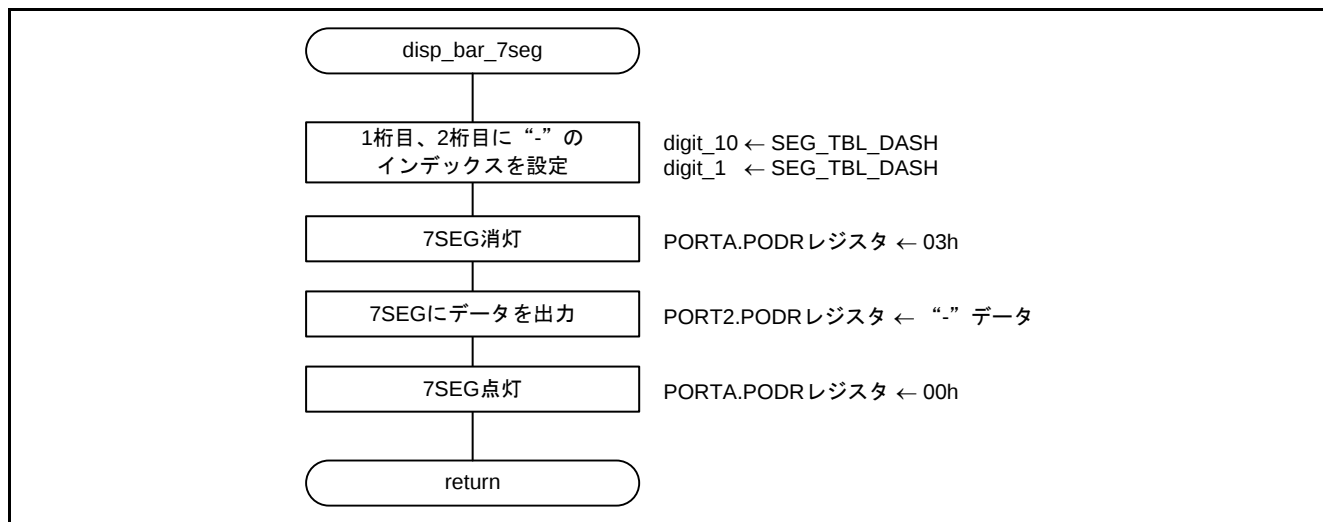


図 5.9 7SEG バー表示処理

5.8.9 コンペアマッチ割り込み処理

図 5.10 にコンペアマッチ割り込み処理のフローチャートを示します。

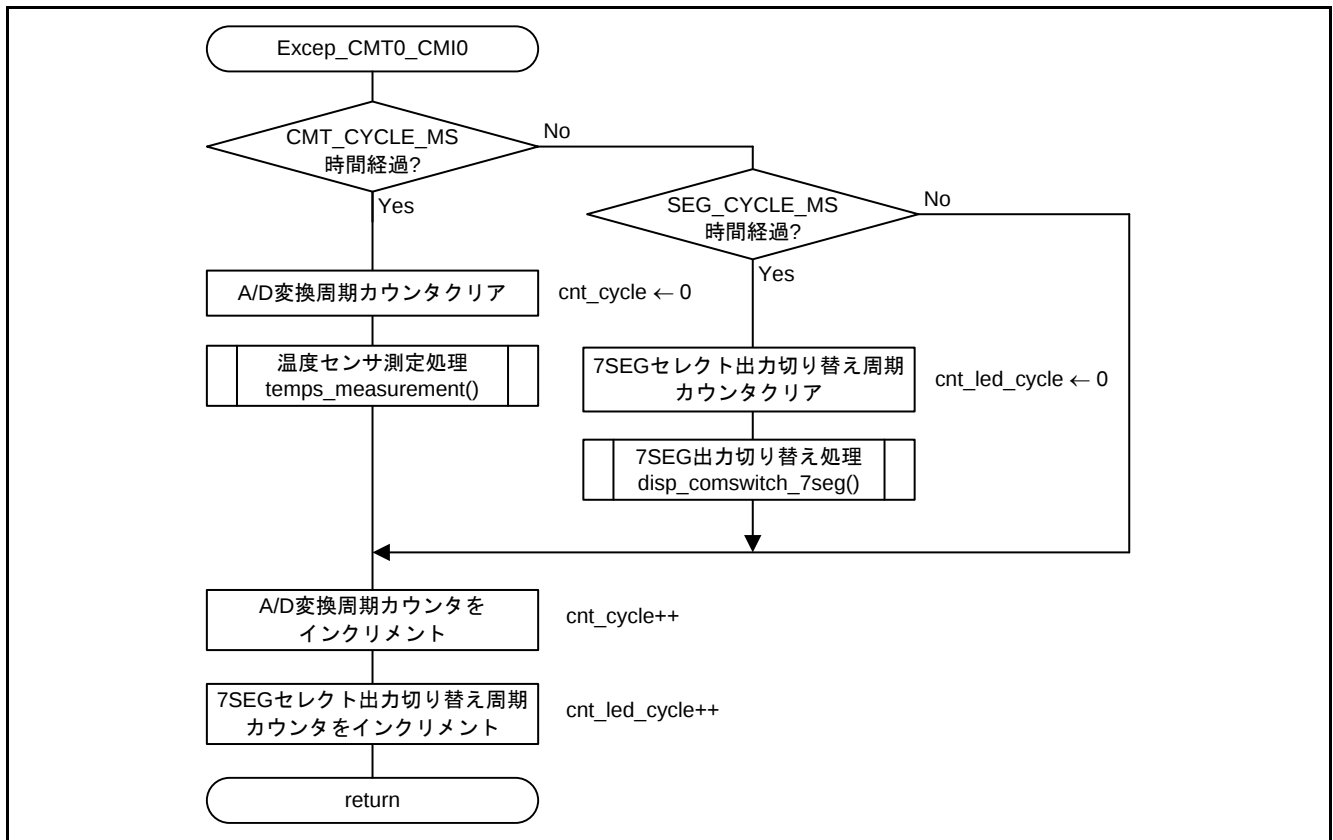


図 5.10 コンペアマッチ割り込み処理

5.8.10 AD、温度センサ初期設定

図 5.11 に AD、温度センサ初期のフローチャートを示します。



図 5.11 AD、温度センサ初期設定

5.8.11 A/D 変換状態取得

図 5.12 に A/D 変換状態取得のフローチャートを示します。

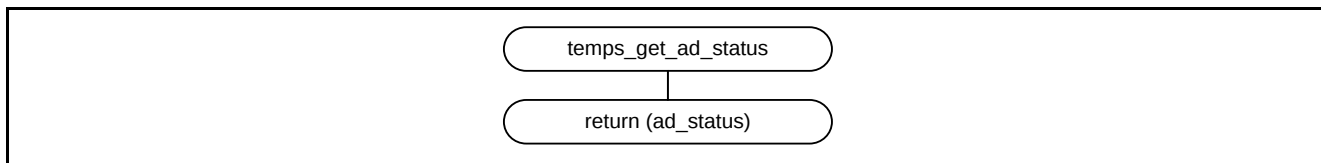


図 5.12 A/D 変換状態取得

5.8.12 温度センサ測定結果取得

図 5.13 に温度センサ測定結果取得のフローチャートを示します。

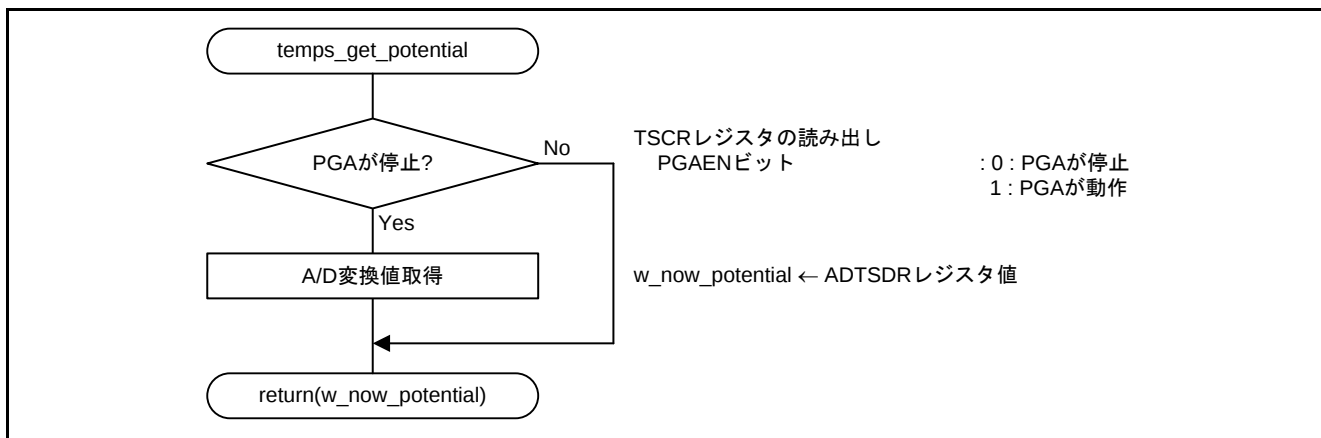


図 5.13 温度センサ測定結果取得

5.8.13 現在温度取得

図 5.14 に現在温度取得のフローチャートを示します。

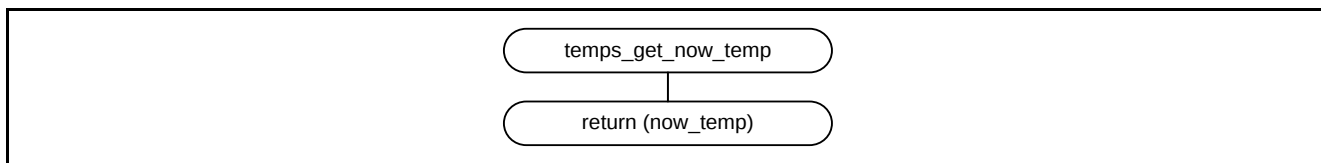


図 5.14 現在温度取得

5.8.14 温度センサキャリブレーション処理

図 5.15 に温度センサキャリブレーション処理のフローチャートを示します。

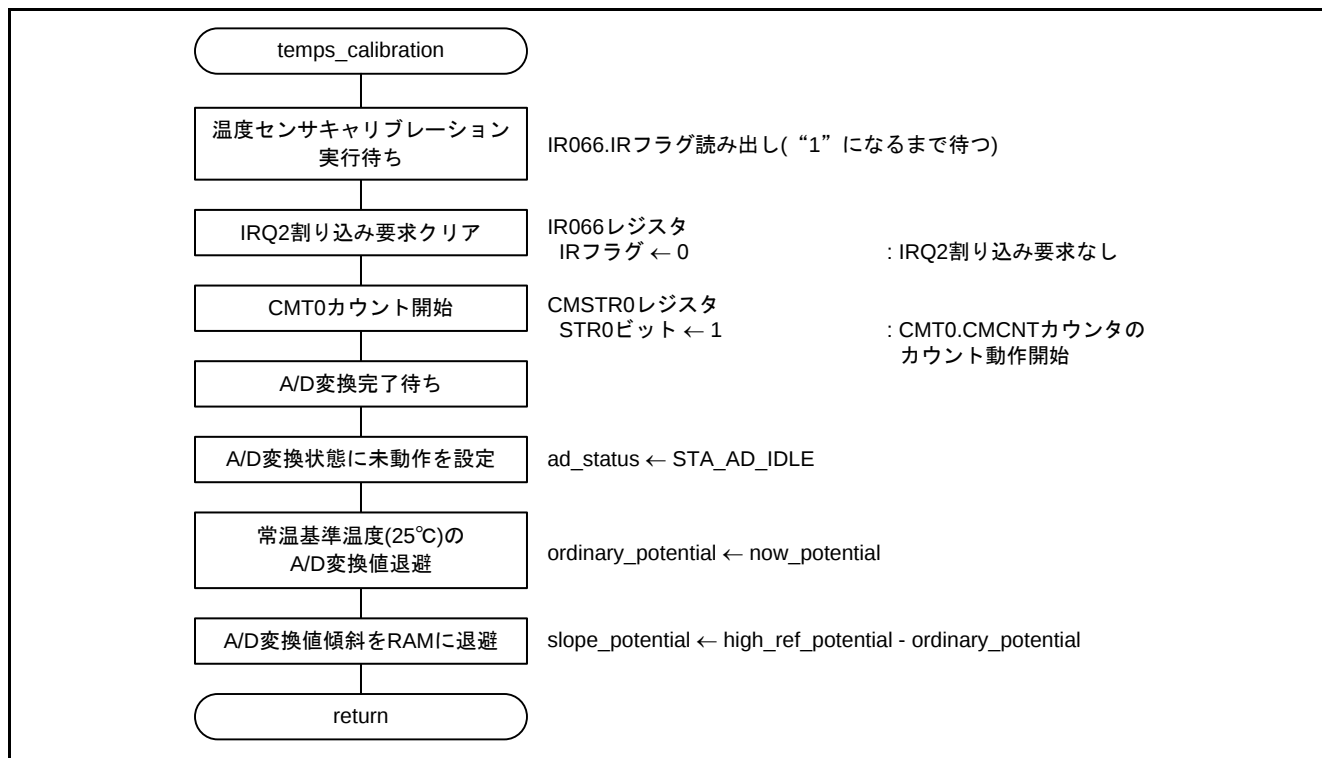


図 5.15 温度センサキャリブレーション処理

5.8.15 温度センサ測定処理

図 5.16 に温度センサ測定処理のフローチャートを示します。

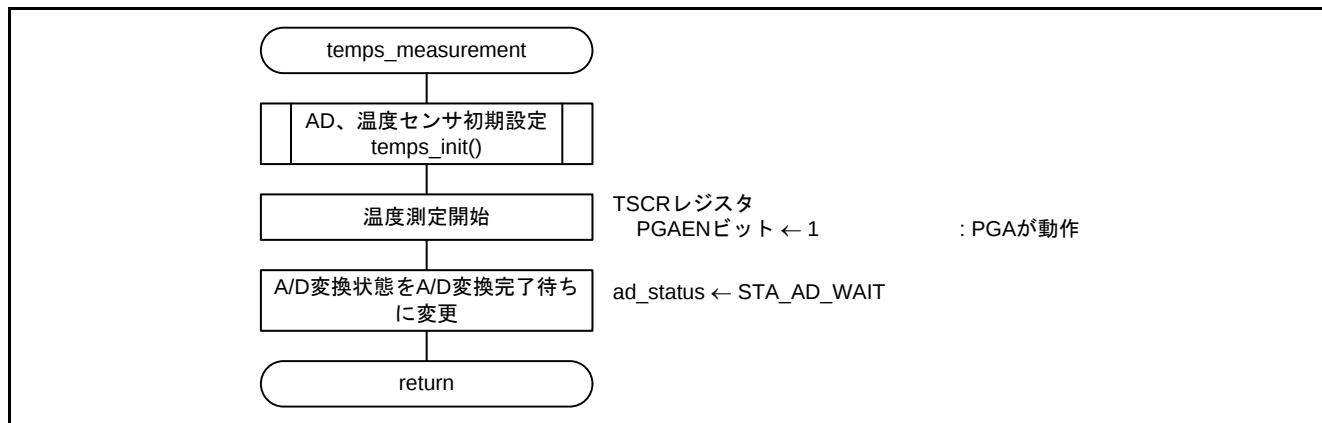


図 5.16 温度センサ測定処理

5.8.16 現在温度算出処理

図 5.17 に現在温度算出処理のフローチャートを示します。

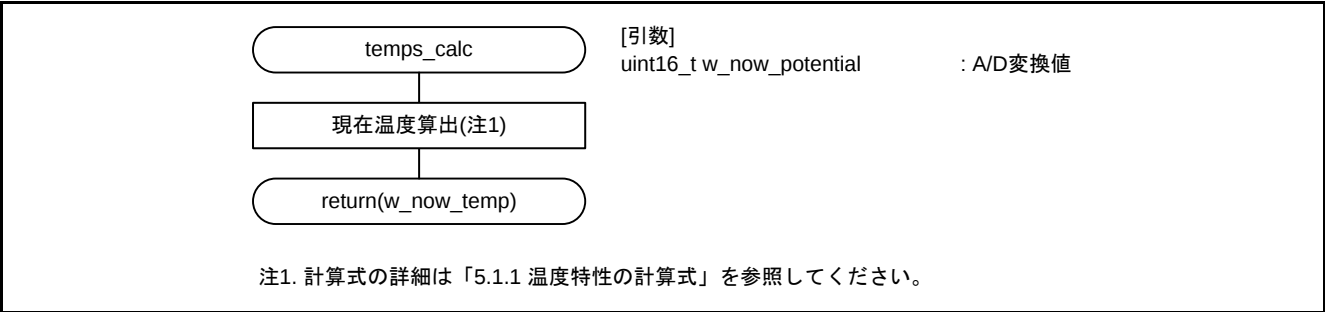


図 5.17 現在温度算出処理

5.8.17 A/D 変換完了割り込み処理

図 5.18 に A/D 変換完了割り込み処理のフローチャートを示します。

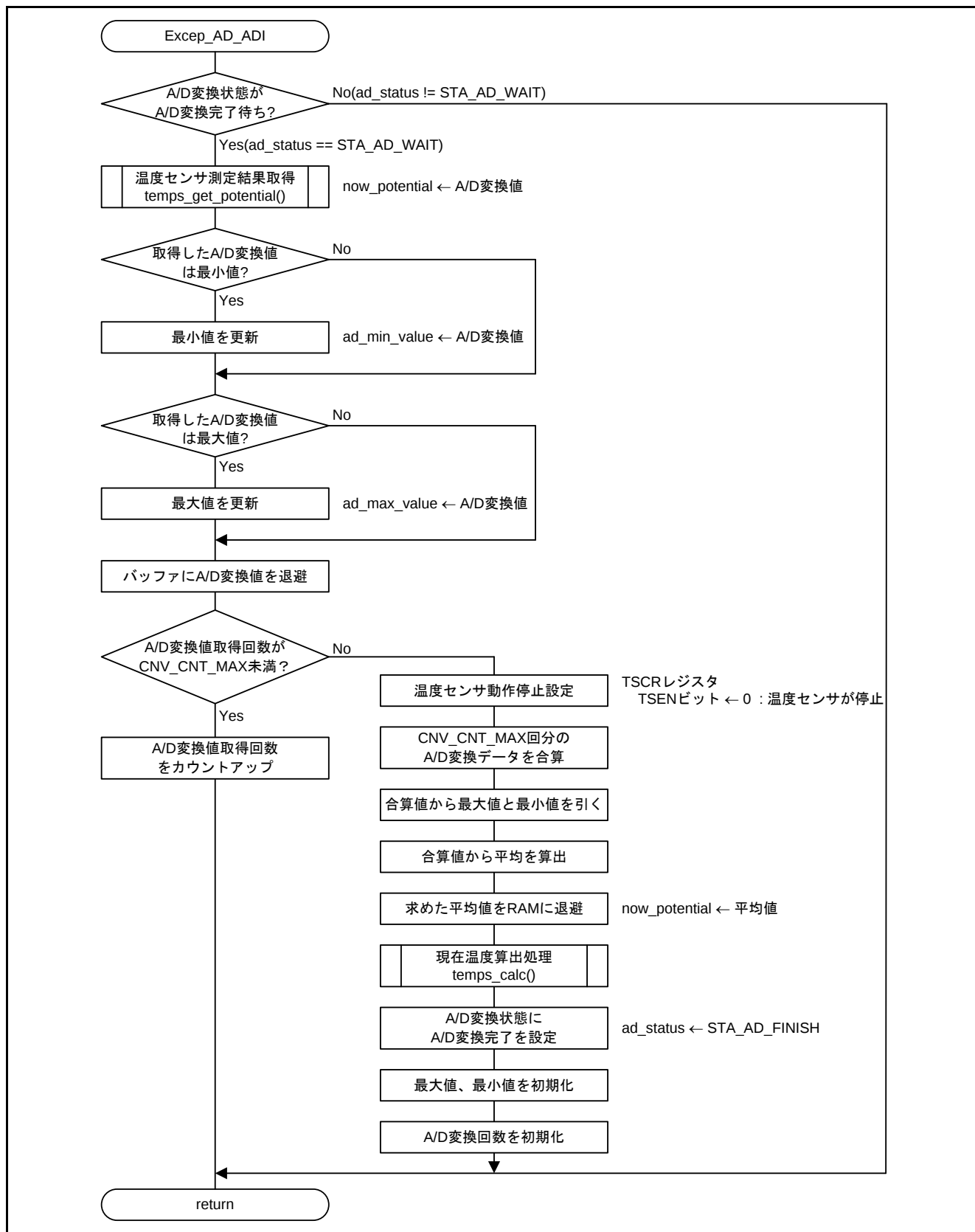


図 5.18 A/D 変換完了割り込み処理

6. サンプルコード

サンプルコードは、ルネサス エレクトロニクスホームページから入手してください。

7. 参考ドキュメント

ユーザーズマニュアル：ハードウェア

RX21A グループ ユーザーズマニュアル ハードウェア編 Rev.1.00 (R01UH0025JJ)
(最新版をルネサス エレクトロニクスホームページから入手してください。)

テクニカルアップデート／テクニカルニュース

(最新の情報をルネサス エレクトロニクスホームページから入手してください。)

ユーザーズマニュアル：開発環境

RX ファミリ C/C++コンパイラパッケージ V.1.01 ユーザーズマニュアル Rev.1.00 (R20UT0570JJ)
(最新版をルネサス エレクトロニクスホームページから入手してください。)

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	RX21A グループ アプリケーションノート 温度センサを用いた周囲温度の算出例
------	--

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2014.09.01	—	初版発行

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. 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. リザーブアドレスのアクセス禁止

【注意】リザーブアドレス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、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違うと、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。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。

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、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、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情報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。



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/>